Technische Universität München
TUM School of Computation, Information and Technology

Robust Face Recognition Under Adverse Conditions

Stefan Hörmann

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Priv.-Doz. Dr. rer. nat. habil. Gabriele Schrag

Prüfer der Dissertation:

1. Prof. Dr.-Ing. habil. Gerhard Rigoll
2. Prof. Dr.-Ing. Eckehard Steinbach

Die Dissertation wurde am 15.06.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 31.01.2023 angenommen.

Abstract

Face recognition (FR) describes the identification of a person in an image. It constitutes the most popular vision-based biometrics technology and has attracted tremendous research interest, accelerating the expansion of FR applications into everyday life. While current FR approaches accomplish near-perfect performance in straightforward scenarios, FR under adverse conditions, such as extreme head poses, occlusions, low image quality, and large age gaps, remains unsolved. This work presents three *robust* solutions to remedy the shortcomings of recent FR approaches under adverse conditions.

Firstly, a novel approach to blind face completion is presented, which reconstructs faces obstructed by small synthetic occlusions prior to the FR. The key innovation is the parallel dual attention structure embedded into the coarse-to-fine network, which leverages global information to reconstruct the occluded pixels, generating a realistic and semantically coherent image. The in-depth analysis demonstrated its applicability even for arbitrary shapes and multiple colors, which were not part of the training. Moreover, face completion substantially mitigates the drop in FR performance of occluded faces. Secondly, in order to cope with face patches, *i.e.*, faces suffering from large occlusions, a robust partial FR model is designed to precisely extract information only from relevant non-occluded pixels and aggregate it in a joint feature space. Experimental results show that satisfying performance is obtained even for tiny non-overlapping face patches. Besides, a performance improvement was observed even on natural occlusions despite exclusively training with synthetic occlusions. Lastly, a unique approach to robust video FR is presented, which constitutes the first permutation-invariant approach to face aggregation. The face aggregation network incorporates information from all video frames to synthesize a more discriminative face, which is then used for FR. Despite the challenging implementation, the presented face aggregation network outperforms various state-of-the-art models, including a permutation-variant face aggregation model. Besides, the robustness against widespread motion blur is improved since the network leverages information from the non-affected frames.

All three suggested FR methods are characterized by their improved robustness under adverse scenarios while maintaining the performance in standard conditions. Furthermore, the face completion and face aggregation approaches additionally provide synthesized images of higher quality, enabling new innovative FR applications.

Contents

List of Acronyms

Acc	Accuracy
AdaGrad	Adaptive gradient
AdaIN	Adaptive instance normalization
Adam	Adaptive moment estimation
ANN	Artificial neural network
BN	Batch normalization
BVMR	Blind visual motif removal
C2F-DAN	Coarse-to-fine dual attention network
CALFW	Cross-Age LFW
CE	Cross-entropy
CFP	Celebrities in Frontal-Profile
CMC	Cumulative match characteristic
CNN	Convolutional neural network
CPLFW	Cross-Pose LFW
DAN	Discriminative aggregation network
DMFB	Dense multi-scale fusion block
EER	Equal error rate
ELU	Exponential linear unit
FAP	Face alignment policy
FAR	False acceptance rate
FN	False negative
FP	False positive
$FPIR$	False positive identification rate
FR	Face recognition
FRR	False reject rate
GAN	Generative adversarial network
GAP	Global average pooling
GPU	Graphical processing unit
IEEE	Institute of Electrical and Electronics Engineers
IJB	IARPA Janus Benchmark

LFW	Labeled Faces in the Wild
LReLU	Leaky ReLU
LSTM	Long short-term memory
MAE	Mean absolute error
MSE	Mean squared error
MTCNN	Multi-task CNN
PLFW	Partial LFW
PFRN	Partial FR network
PIFAN	Permutation-invariant face aggregation network
$PSNR$	Peak signal-to-noise ratio
ReLU	Rectified linear unit
ResNet	Residual network
RMSProp	Root mean square propagation
ROC	Receiver operating characteristic
SIFT	Scale-invariant feature transform
SGD	Stochastic gradient descent
$SSIM$	Structural similarity
SURF	Speeded up robust features
TAR	True acceptance rate
TN	True negative
TP	True positive
$TPIR$	True positive identification rate
YTF	YouTubeFaces

List of Symbols

Miscellaneous

$\mathbb{1}_{a,b}$	All-ones matrix of size $a \times b$
$\boldsymbol{A}^\mathsf{T}$	Transpose of the matrix $\boldsymbol{A}$
$\|\boldsymbol{a}\|, \|\boldsymbol{A}\|$	L_2 norm of the vector $\boldsymbol{a}$ or Frobenius norm of the matrix, tensor $\boldsymbol{A}$
$[\boldsymbol{A}]_{i,j}$	Element in the ith row and jth column of the matrix $\boldsymbol{A}$
$\odot$	Hadamard product, *i.e.*, element-wise multiplication
$\otimes$	Matrix multiplication
$\oplus$	Addition
$\circledast$	Channel-wise 2D convolution
$\oplus\!\!\!\!\!\!\bigcirc$	Concatenation
$\times$	Cartesian product
$A^{[l]}$	Parameter A at depth l
$\boldsymbol{A}^{(n)}$	nth sample $\boldsymbol{A}$
$\tilde{A}$	Variation of the parameter A, *e.g.*, before normalization
$\hat{A}$	Prediction of the parameter A by the algorithm
$\boldsymbol{A}^*$	Optimal parameter A according to an optimization problem
$\frac{\partial}{\partial x}$	Partial derivative with respect to x
∇	Nabla operator

Functions

$\mathrm{A}(\cdot)$	Face aggregation network
$\mathrm{C}(\cdot)$	Crop a patch from an image
$d(\cdot, \cdot)$	Cosine distance between two features
$\mathrm{D}(\cdot)$	Discriminator
$\mathrm{F}(\cdot)$	Face feature extractor
$\mathrm{G}(\cdot)$	Generator

$\log(\cdot)$	Natural logorithm, *i.e.*, $\log(\cdot) = \log_{\mathrm{e}}(\cdot)$
$\mathscr{L}(\cdot, \cdot)$	Loss function
$\mathcal{N}_k(\mu, \sigma^2)$	k-dimensional uncorrelated Gaussian normal distribution with mean μ and variance σ^2
$\mathcal{U}_k(a, b)$	k-dimensional uncorrelated uniform distribution from a to b
$\Gamma(\cdot)$	Gram matrix
$\Theta_\theta(\cdot)$	Output vector or tensor of the network with weights θ
$\tilde{\Theta}_\theta(\cdot)$	Output vector or tensor of the network with weights θ without activation function
$\Xi(\cdot)$	Helper function for the decision boundary
$\Psi(\cdot)$	Activation function

Greek

α	Parameter of the leaky ReLU, PReLU or ELU activation function
α	Triplet loss margin for positive feature pairs
α_{BN}	BN: exponential decay rate
β	Parameter of the swish activation function
β	Triplet loss margin for negative feature pairs
β_1, β_2	Exponential decay rates for the moment estimates in Adam optimizer
$\boldsymbol{\beta}$	BN: target mean
γ_{lr}	Learning rate decay factor
$\gamma_{\mathrm{b}}, \gamma_{\mathrm{c}}, \gamma_{\mathrm{s}}$	Parameter adjusting the intensity of brightness, contrast, and saturation augmentation
$\boldsymbol{\gamma}$	BN: target variance
ϵ_{BN}	BN: small number
ζ	Similarity transformation: uniform scaling
η, η_{c}	Learning rate (of the feature centers)
θ	Set of all trainable weights and biases within a network
$\kappa_{\mathrm{a}}, \kappa_{\mathrm{h}}$	Area ratio, height ratio of an occlusion
λ	Scalar to balance the loss terms
$\mu, \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BN}}$	Arithmetic mean, mean of the minibatch (BN)
$\nu, \boldsymbol{\nu}$	Relevance of a feature (component) during aggregation
$\sigma, \boldsymbol{\sigma}, \boldsymbol{\sigma}^2_{\mathrm{BN}}$	Standard deviation, variance of the minibatch (BN)
$\tau_{\mathrm{x}}, \tau_{\mathrm{y}}$	Similarity transformation: horizontal and vertical shift
φ	Similarity transformation: rotation

Latin

$\tilde{\boldsymbol{A}}, \boldsymbol{A}$	Attention map before and after recalibration
$\boldsymbol{b}$	Bias vector
C	Channels of a tensor
$\boldsymbol{c}$	Color vector
$\boldsymbol{c}_\mathrm{o}$	Center position of a occlusion
$\boldsymbol{c}^i$	Center of the feature of the ith idenitiy
d_{ij}	Cosine distance between the ith probe feature and jth gallery feature
D	Dilation factor
$\boldsymbol{f}, \boldsymbol{F}$	Feature vector, map, tensor
$\tilde{\boldsymbol{f}}, \widetilde{\boldsymbol{F}}$	Feature vector, map, tensor before the activation function
$\boldsymbol{f}_\mathrm{A}, \boldsymbol{f}_\mathrm{P}, \boldsymbol{f}_\mathrm{N}$	Feature vector of the anchor, positive, negative face
$\mathcal{G}$	Gallery set for face identification
H, H_txt	Height of a matrix, tensor, and of a text in px
i, j, k, c, n	Indices; typically input i, output j, general k, numbers n, channels c
$\boldsymbol{I}$	Image, typically a face. Short for $\boldsymbol{I}_\mathrm{aug}$ during training and $\boldsymbol{I}_\mathrm{align}$ during evaluation
$\boldsymbol{I}_\mathrm{org}$	Unprocessed face from a dataset
$\boldsymbol{I}_\mathrm{align}$	Face after alignment
$\boldsymbol{I}_\mathrm{aug}$	Aligned face after augmentation
$\hat{\boldsymbol{I}}_\mathrm{a}$	Aggregated face
$\boldsymbol{I}_\mathrm{m}, \boldsymbol{I}_\mathrm{gt}, \hat{\boldsymbol{I}}_\mathrm{f}, \hat{\boldsymbol{I}}_\mathrm{f,r}$	Occluded face, ground-truth face, the prediction of the fine network and the prediction of the fine network after the reconstruction block
$K_\mathrm{H} \times K_\mathrm{W}$	Size of a convolutional filter
$\boldsymbol{K}$	Attention layer: key
L	Number of trainable layers, *i.e.*, depth, of a neural network
m_1, m_2, m_3	Multiplicative, (angular) additive margins
$M, M_\mathrm{f}, M_\mathrm{cls}$	Number of neurons of a fully connected layer, the feature embedding layer and the last layer
MSE_n	MSE normalized by the intraocular distance
$\boldsymbol{M}_\mathrm{gt}, \hat{\boldsymbol{M}}$	Ground-truth mask denoting occlusions and the mask predicted by the network
N	Number of samples in a dataset or number of frames in a video (face aggregation)
N_avg	Face aggregation: number of features to be averaged after face aggregation

N_{blur}	Face aggregation: number of frames affected with motion blur
N_{b}	Minibatch size
$N_{\text{train}}, N_{\text{test}}$	Number of samples in a training and test dataset
N_{LM}	Number of landmarks
$N_{\text{P}}, N_{\text{G}}$	Number of elements in the probe and gallery set
N_{d}	Number of distractors in MegaFace benchmark
$p_{\text{aug}}, p_{\text{d}}$	Augmentation probability and dropout probability
$P_{\text{H}}, P_{\text{W}}$	Number of pixels for horizontal and vertical zero padding at every side
$\mathcal{P}, \mathcal{P}_{\text{G}}, \mathcal{P}_{\text{N}}$	Probe set for face identification, exclusively with or without identities in the gallery
$\boldsymbol{Q}$	Attention layer: query
r	Image resolution
R	Rank of a match in face identification
$\boldsymbol{R}$	Attention layer: reconstructed tensor
s	Scale parameter indicating the hypersphere radius in feature space
S	Stride
$\tilde{\boldsymbol{s}}, \boldsymbol{s}$	Score vector before and after normalization
$\boldsymbol{S}$	Attention layer: similarity matrix
t	Discrimination threshold
$\boldsymbol{T}$	Transformation matrix
$\boldsymbol{V}$	Attention layer: value
$\mathcal{V}$	Video
W	Width of a matrix, tensor
$\boldsymbol{w}, \boldsymbol{W}$	Weight vector, matrix or kernel (for convolution)
$x, \boldsymbol{x}, \boldsymbol{X}$	Scalar, vector or tensor input
$\begin{pmatrix} x_{i,\text{src}} & y_{i,\text{src}} \end{pmatrix}^{\mathsf{T}}$	Source landmark coordinates
$\begin{pmatrix} x_{i,\text{tar}} & y_{i,\text{tar}} \end{pmatrix}^{\mathsf{T}}$	Target landmark coordinates
$\boldsymbol{X}_{\text{real}}, \boldsymbol{X}_{\text{fake}}$	Real, fake image
$\mathcal{X}, \mathcal{X}_{\text{b}}$	Dataset, minibatch
$\mathcal{X}_{\text{train}}, \mathcal{X}_{\text{val}}, \mathcal{X}_{\text{test}}$	Training, validation and test dataset
$y, \boldsymbol{y}$	Ground-truth value, typically the identity
$\hat{y}, \hat{\boldsymbol{y}}$	Prediction by the network

List of Figures

List of Tables

1

Introduction

Face recognition (FR) (or facial recognition) is the "technology that makes it possible for a computer to recognize a digital image of someone's face" [13]. Like gait, fingerprint, palm print, iris, and retina recognition, FR forms part of vision-based biometrics, *i.e.*, systems that leverage images of an individual's characteristic features for identification. Compared to other computer vision tasks, *e.g.*, action and gesture recognition, pose and face detection, object tracking, image synthesis, *etc.*, biometrics technology is unique as it requires exhaustive analysis due to its employment in security-sensitive areas.

FR research has made tremendous improvements in the last decades. The first breakthrough in FR was set in 1991 by Turk and Pentland [207]. They expressed every face as a combination of so-called eigenfaces, which are obtained by performing a principal component analysis on a face dataset. Shortly thereafter, many approaches achieved satisfying results under controlled conditions, *e.g.*, as covered by the Yale Face Database B [51]. Hence, new unconstrained datasets, such as the Labeled Faces in the Wild (LFW) [86] dataset, emerged, providing a new objective in FR research. Owing to the advances in *artificial neural network*s (ANNs), a new era in FR approaches required even more adverse evaluation datasets. Therefore, current challenges in FR become apparent by observing the evaluation datasets – more precisely in the face pairs, which the FR approach tries to classify as "same identity" or "different identity", which are depicted in Figure 1.1.

Even though the Similar-Looking LFW dataset [37] guaranteed that all face pairs of different identities in LFW look alike, current approaches managed to accomplish

Figure 1.1: Accuracy of state-of-the-art FR approaches on various evaluation datasets, which are represented by face pairs. Datasets are marked with ✓ if the accuracy exceeds 99.5% and saturation effects arise. Note that only the face pair from the Cross-Age LFW (CALFW) dataset is from the same identity.

near-perfect accuracy. Thus, the difficulty of pairs comprising faces of the same identity also needed to be increased. Cross-Age LFW (CALFW) [276] and Cross-Pose LFW (CPLFW) [275] outline the current limits of FR approaches. They created challenging face pairs by ensuring identical gender and ethnicity in pairs with different identities and only considering pairs with large age gaps or head pose variations.

Recent datasets for evaluating FR approaches demonstrated that FR still constitutes an unsolved problem under real-world conditions. Despite almost impeccable performance for frontal faces and the ability to distinguish even similar identities, adverse scenarios still pose a challenge, attracting massive research interest lately. Besides face pairs with an increased age gap or large head pose variations, various datasets were proposed to specifically evaluate adversarial attacks [280], partial faces [10†], cross-quality [16†], surveillance video [100], or racial bias [218]. Therefore, novel approaches are required to address the remaining challenges in FR.

1.1 Motivation

FR constitutes the most popular biometric modality for recognition even though it is typically less accurate than other biometric characteristics such as fingerprints or iris scans. Besides, faces can be manipulated with makeup and disguises to imitate other identities. On the other hand, FR benefits from being non-intrusive, *i.e.*, face images are obtained quickly with sufficiently high quality and without any action or physical contact of the person as opposed to other biometric modalities. Furthermore, faces constitute an abundant source of information as also information about the head pose, fiducial keypoints, gender, age, expression, and the presence of face attributes, *e.g.*, beard, hair color, piercing, *etc.*, can be extracted to build more powerful systems.

However, potentially missing consent to capture the data raises many ethical concerns. *E.g.*, given the ubiquitous surveillance in major cities [12], many people question data privacy or feel that their liberties are threatened [120]. Without understating the importance of ethical concerns provoked by this controversial technology, this work focuses entirely on the technical part of FR systems, emphasizing the algorithms and performance of the latest FR systems. Hence, an in-depth discussion about the ethical aspects of FR is out-of-scope. Consequently, consultation with an ethical expert is highly encouraged when employing any FR system.

Regardless of the ethical concerns, the FR market is growing and is projected to reach a worldwide value of 11.6 billion USD in 2026 (from 3.7 billion USD in 2020) [155]. China constitutes one of the major consumers and producers of FR technology, which is supported by the number of surveillance cameras in use. *E.g.*, in China, 54% of the global number of surveillance cameras, or $\approx$ 415M, were installed in 2021 [12]. Moreover, western cities are also highly surveilled; *e.g.*, on average, 1138 cameras per square mile are in use in London, which corresponds to a distance between two cameras of less than 50 meters if they were equally distributed in a grid [12]. Traditionally, the surveillance footage was manually screened for persons of interest by humans with a "significantly better than ordinary FR ability" [181] – so-called super-recognizers.

Nowadays, super-recognizers are still widely employed despite the success of automatic FR. This is supported by Phillips *et al.* [171], who showed that the performance of an FR algorithm could be further improved with human supervision.

Owing to the wide availability of low-cost embedded chips capable of running a real-time FR system with satisfying performance, nowadays, everyday life cannot be imagined without FR. Over the recent years, this development has become apparent even in the different options to unlock a smartphone. After multi-digit passcodes, fingerprint authentication became widely employed and reduced the effort of unlocking a smartphone to a single button press. Today, this already minimal effort is minimized using FR, which only requires a quick look at the phone and has even been enhanced to work when wearing a medical face mask [5]. The employment of FR technology across many operating systems and devices with sensitive private data indicates that sufficiently high accuracy can be guaranteed. Therefore, a detailed scan of the user's face encompassing varying head poses is required when setting up the authentication method. Moreover, popular products incorporate depth and infrared [4] or solely infrared [151] sensory information to ensure that spoofing attacks fail.

In addition to law enforcement or unlocking personal devices with on-device FR, there exist plenty of opportunities for employing FR to improve or speed-up daily life. *E.g.*, FR facilitates migration and passport control at airports. Retail stores that share a common database of convicted shoplifters can deny their entry even if they have never been to the store [161]. With the help of FR, not only fugitives or criminals but also missing children can be found [29]. People in photos uploaded to social media can be identified and photos can be clustered automatically into albums [47]. FR can further help in increasing safety at schools by automatically detecting unauthorized persons. Besides, the pupils' attendance can be tracked with FR, ensuring that no student signs for other students as in traditional attendance sheets [15]. Curiously, FR was even employed to limit the amount of toilet paper every individual could dispense within a certain amount of time [172]. Unsurprisingly, citizens were concerned about using cameras in a private space, resulting in its suspension.

To comply with this sheer unlimited creativity of employing FR in every inch of daily life, robust and reliable FR is essential for the success of any application. In order to find missing children, it is crucial to recognize faces despite a large age gap or develop methods for kinship recognition [8†]. Moreover, recognizing faces occluded by medical face masks or sunglasses is crucial to provide the user with a convenient way of unlocking a smartphone even amidst a pandemic or intense sunlight. When considering the recognition of faces in videos, new challenges arise due to the nature of the data since the approach must be capable of combining the information in an arbitrary number of frames. Besides, videos are typically captured in poor conditions and affected by motion blur due to camera movement. Consequently, current FR approaches suffer under adverse conditions, which is affirmed by the analyses on varying head poses, occlusions, noise, JPEG quality, and blur [60, 115, 145, 281, 9†, 10†, 13†]. Thus, algorithms with increased robustness are required to succeed in these unique scenarios and promote the advancement of FR in everyday life.

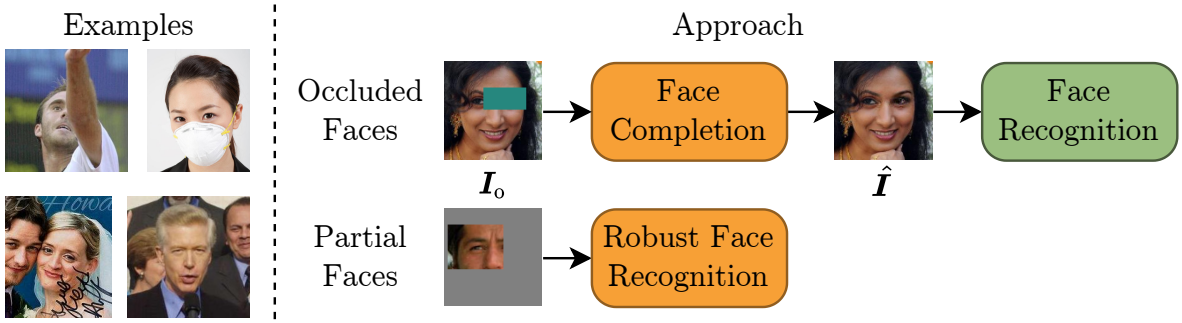

Figure 1.2: Left: Example of occluded faces. Right: Two approaches to cope with occluded faces $\boldsymbol{I}_o$. After reconstructing the occluded pixels in $\boldsymbol{I}_o$, an existing FR approach is leveraged. Alternatively, a typical FR approach is altered to better handle $\boldsymbol{I}_o$. The novel approaches presented in this work are highlighted in orange.

Apart from proposing robust algorithms for face recognition, recent progress in generative adversarial networks (GANs) [57] provides researchers with the tools to find new solutions, which are employed prior to the FR. *E.g.*, a face with a large head pose can be synthesized into a frontal face [164, 206] or the age gap can be reduced [76]. In order to cope with partial faces, missing or occluded areas can be reconstructed [138, 237, 284]. For videos, Rao *et al.* [178] proposed aggregating multiple video frames into a single more discriminative image. All approaches constitute popular alternatives compared to directly improving the robustness of an FR system since they provide the frontal, reconstructed, or aggregated face as an additional output. Ideally, this synthesized face contains the same or better identity information, whereas, in videos, it also provides a more compact representation than a large number of frames.

1.2 Objectives

This work considers two adverse FR scenarios and presents three solutions to remedy the vulnerability of state-of-the-art FR approaches.

Occluded and *partial* faces refer to images of a face where a part of the face, and therefore, part of their identity-defining characteristics, is occluded. Figure 1.2 (left) depicts examples of naturally occluded faces $\boldsymbol{I}_o$, *e.g.*, faces occluded by foreground objects such as arms, medical face masks, or text, and partial faces, *e.g.*, faces cut off at the image's border. Both are very similar; however, occluded faces comprise faces where most pixels are unobstructed, whereas only a small part of the face is visible for partial faces.

Typically, occluded and partial faces $\boldsymbol{I}_o$ represent a minority within large training datasets. Therefore, during the training of FR models, the network only learns a general representation of the face overlooking the characteristics of $\boldsymbol{I}_o$. To overcome this imbalance problem, augmenting the data by synthetically generating occluded and partial faces is a straightforward solution. However, the best performance is accomplished only by tailoring the entire FR algorithm to a given scenario.

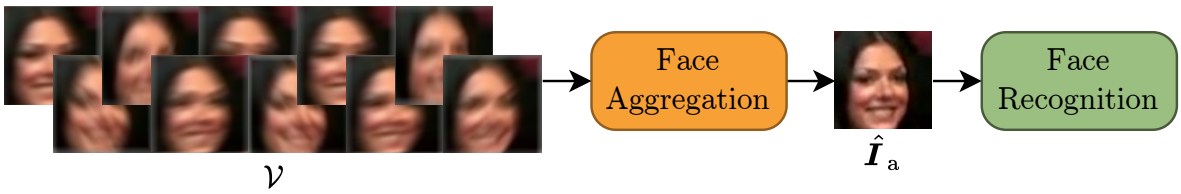

Figure 1.3: Approach for video face recognition. Frames of a video $\mathcal{V}$ are aggregated, yielding a more discriminative image $\hat{\boldsymbol{I}}_{\mathrm{a}}$, which is used for FR.

Depending on the size of the occlusions, two approaches are proposed: 1) predicting the pixels' values within the occluded areas, *i.e.*, face completion; or 2) increasing the robustness of FR.

By reconstructing the pixel values of the occluded areas in $\boldsymbol{I}_{\mathrm{o}}$, a new face $\hat{\boldsymbol{I}}$ is generated. Ideally, $\hat{\boldsymbol{I}}$ should be realistic, *i.e.*, only non-occluded areas of $\hat{\boldsymbol{I}}$ must be altered and $\hat{\boldsymbol{I}}$ should resemble the image before applying the synthetic occlusion. Besides, $\hat{\boldsymbol{I}}$ should mitigate the drop in FR performance when exposed to occlusions. In order to increase the applicability, the face completion should be *blind*, *i.e.*, no additional mask annotation is required. Furthermore, it should handle occlusions varying in form, position, size, and color.

Reconstructing tiny face patches, *e.g.*, faces where only 10% of the area is visible, constitutes a complicated task with few chances of satisfying results. Thus, a direct partial FR approach without prior reconstruction is more promising. Apart from increasing the robustness against partial faces compared to state-of-the-art FR approaches, this direct approach should still provide comparable results on non-partial faces. Moreover, it should improve the performance for synthetic and natural occlusions, as in Figure 1.2 (left), and permit the comparison of two non-overlapping face patches, *e.g.*, patches centered around the mouth matched with a patch containing an eye.

Besides occluded and partial faces, this work also addresses video FR. Here, motion blur constitutes an omnipresent issue as frequent head or camera movements lead to poor quality frames compared to the high quality still images. Additionally, video FR provides multiple video frames, which raises the question of how to efficiently combine the information from an arbitrary number of frames. Even though temporal information is available, it is rarely leveraged as it does not help extract richer identity information. In order to also consider set-based FR tasks, *i.e.*, a set of various images of the same identity, the video is described as a set $\mathcal{V}$ by dispensing with the frame order.

Unlike most state-of-the-art methods in video FR, which perform the aggregation of information of every frame in the feature space, this work presents a rather unorthodox approach by aggregating the face in the image space, as depicted in Figure 1.3. In fact, by considering $\mathcal{V}$ as a set, it constitutes the first approach for permutation-invariant face aggregation of video frames. While this requires advanced methods to ensure permutation invariance and allow information exchange between every frame, it has the benefit of providing the aggregated face as an additional output. Hence, it can be considered the first step towards face aggregation for sets comprising various still images. Besides being permutation invariant and capable of aggregating an arbitrary number of frames, the face

aggregation should provide a realistic image $\hat{\boldsymbol{I}}_{\mathrm{a}}$, which fuses the identity information of the entire video $\mathcal{V}$. In this way, $\hat{\boldsymbol{I}}_{\mathrm{a}}$ should ease distinguishing whether different identities occur in two videos – particularly if $\mathcal{V}$ suffers from motion blur.

In summary, this work contributes threefold to FR research. Besides an approach to partial FR, a blind face completion and a face aggregation method are presented. While the first contribution focuses on increasing the robustness of state-of-the-art algorithms, the latter two aim to increase robustness by generating an image of higher quality, which is then analyzed by state-of-the-art FR approaches. For occluded faces, the approach yields a reconstructed image $\hat{\boldsymbol{I}}$, where the viewer should be unable to recognize that the face was occluded in the first place. Moreover, the aggregated face $\hat{\boldsymbol{I}}_{\mathrm{a}}$ for video FR should constitute a compact representation of the all-embracing identity information within the input video $\mathcal{V}$. Even though the approach for partial FR does not generate any additional output, it should handle even tiny non-overlapping face patches.

1.3 Overview

The structure of this work is visually illustrated in Figure 1.4 and is summarized as follows:

Chapter 2 introduces the notation and lays the mathematical foundation essential for following and comprehending the subsequent chapters. This involves the components of an ANN for computer vision and ranges from elementary building blocks to optimization. Besides, a sophisticated framework to generate realistic images is described with the GAN.

Chapter 3 establishes the FR pipeline, *i.e.*, every step to build and evaluate an FR system. For that, a thorough discussion of the relevant literature is provided. Moreover, three base models, which play a vital role in the subsequent chapters, are evaluated in-depth and put into context with state-of-the-art approaches.

Chapter 4 presents the design and implementation of a novel approach for occlusion-robust FR by blind face completion. The chapter is concluded by a comprehensive evaluation encompassing quantitative and qualitative analyses.

Chapter 5 provides an alternative solution to the FR of highly occluded, *i.e.*, partial, faces and demonstrates its benefits for naturally occluded faces.

Chapter 6 introduces a face aggregation approach for recognizing faces in videos – particularly when affected by motion blur. It constitutes the first step towards permutation-invariant face aggregation of a set of still images.

Chapter 7 recapitulates the critical findings of this work and draws the main implications w.r.t. the research objectives stated in Section 1.2.

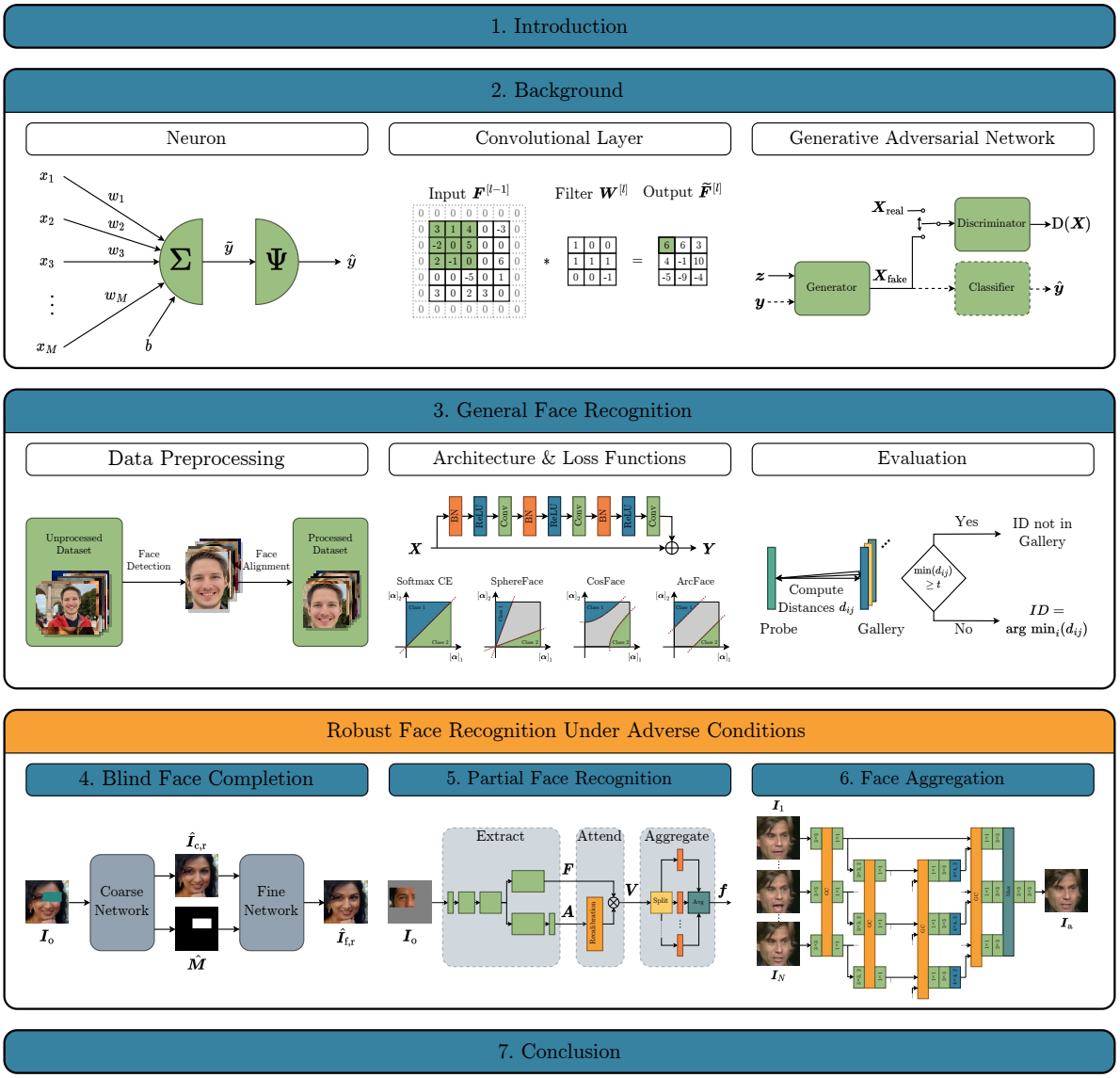

Figure 1.4: Graphical overview of the content of this work. After establishing a foundation in artificial neural networks (ANNs), general still image FR is explained in-depth, which forms a part of subsequent chapters. The core of this work form the novel approaches to robust FR under challenging conditions: blind face completion, partial face recognition, and face aggregation. The chapters are highlighted in blue.

2

Background in Artificial Neural Networks

The target of artificial neural networks (ANNs) is modeling how we – as humans – think as a mental process with logical rules. In its simplest realization, the ANN comprises a single processing unit – a *neuron*. While early models of a neuron were limited to predicting a binary output based on multiple binary inputs [147], more recent implementations of ANNs contain millions of neurons and have surpassed human performance on numerous tasks a long time ago, *e.g.*, in 2014 for *face recognition* (FR) [113, 198] or in 2015 for image classification [69].

To follow the remaining part of this work and the advances that led models to obtain performance superior to humans, this chapter briefly establishes the methodological and mathematical foundation. The mathematical notation utilized throughout this work is based on ISO 80000-2 [94] and is summarized in Appendix A.

2.1 Perceptron

The initial binary model of a neuron [147] was extended by Rosenblatt [180] to handle continuous real-valued inputs and outputs. In a similar form, the *perceptron* constitutes one of the core building blocks within recent ANNs. Figure 2.1 (left) depicts a graphical representation of a perceptron. For an input vector $\boldsymbol{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_M \end{pmatrix}^\mathsf{T}$, the output $\hat{y}$ is obtained by

$$\hat{y} = \Psi\left(\tilde{y}\right) = \Psi\left(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b\right), \tag{2.1}$$

with $\boldsymbol{w} = \begin{pmatrix} w_1 & w_2 & \dots & w_M \end{pmatrix}^\mathsf{T}$ and b being the weight vector and the bias, respectively. $\tilde{y}$ denotes the intermediate output before applying the activation function $\Psi\left(\cdot\right)$ (see also Section 2.3), which allows the perceptron to model non-linear behavior but can also be set to $\Psi\left(\tilde{y}\right) = \tilde{y}$ for a linear relationship.

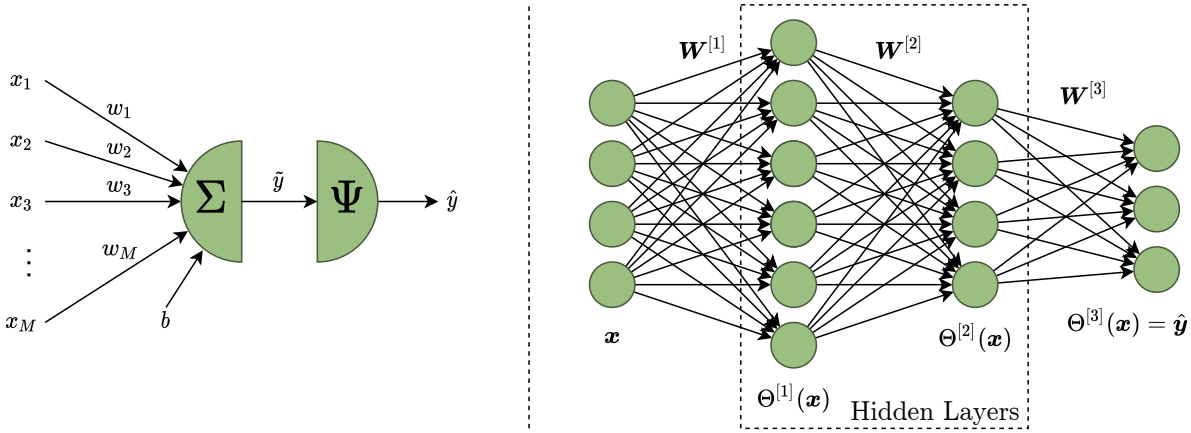

Figure 2.1: Left: Model of a perceptron. Right: Various perceptrons forming a multilayer fully connected neural network. For the sake of simplicity, the bias was omitted.

2.2 Fully Connected Neural Networks

Since the computational power of a single perceptron is very limited, multiple perceptrons are stacked layerwise to form a computational graph, in which all output neurons of the first layer are connected with all input neurons of the subsequent layer (see Figure 2.1 (right)). In this way, the information is processed in a feedforward manner. The output of a so-called hidden layer $\Theta^{[l]}(\boldsymbol{x}) \in \mathbb{R}^{M^{[l]}}$ at depth l with input $\boldsymbol{x}$ is calculated by

$$\begin{aligned} \Theta^{[l]}\big(\boldsymbol{x}\big) &= \Psi^{[l]}\big(\tilde{\Theta}^{[l]}(\boldsymbol{x})\big) = \Psi^{[l]}\big(\boldsymbol{W}^{[l]}\Theta^{[l-1]}\big(\boldsymbol{x}\big) + \boldsymbol{b}^{[l]}\big), \\ \Theta^{[0]}\big(\boldsymbol{x}\big) &= \boldsymbol{x}, \end{aligned} \tag{2.2}$$

with $\boldsymbol{W}^{[l]} \in \mathbb{R}^{M^{[l]} \times M^{[l-1]}}$, $\boldsymbol{b}^{[l]} \in \mathbb{R}^{M^{[l]}}$ and $\Psi^{[l]}(\cdot)$ denoting the weight matrix, bias vector and the element-wise activation function at depth l, respectively. $M^{[l]}$ is the number of neurons in a layer at depth l. Hence, the weight $\big[\boldsymbol{W}^{[l]}\big]_{i,j}$ indicates how the output $\big[\Theta^{[l-1]}(\boldsymbol{x})\big]_j$ of the jth neuron in layer $l-1$ is scaled when it is connected to the input of the ith neuron in layer l. The output vector $\hat{\boldsymbol{y}}$ of a network with depth L is then

$$\begin{aligned} \hat{\boldsymbol{y}} &= \Theta^{[L]}(\boldsymbol{x}) \\ &= \Psi^{[L]}\big(\boldsymbol{W}^{[L]}\Psi^{[L-1]}\big(\boldsymbol{W}^{[L-1]}\cdots\big(\Psi^{[1]}\big(\boldsymbol{W}^{[1]}\boldsymbol{x} + \boldsymbol{b}^{[1]}\big)\cdots\big) + \boldsymbol{b}^{[L-1]}\big) + \boldsymbol{b}^{[L]}\big). \end{aligned} \tag{2.3}$$

When assuming linear activation functions $\Psi^{[l]}\big(\tilde{\Theta}^{[l]}(\boldsymbol{x})\big) = \tilde{\Theta}^{[l]}(\boldsymbol{x})\ \forall l$, Equation (2.3) can be rewritten as

$$\hat{\boldsymbol{y}} = \left[\prod_{l=1}^{L} \boldsymbol{W}^{[l]}\right]\boldsymbol{x} + \boldsymbol{b}^{[L]} + \sum_{l=1}^{L-1}\left[\prod_{j=l+1}^{L} \boldsymbol{W}^{[j]}\right]\boldsymbol{b}^{[l]}. \tag{2.4}$$

Hence, without any non-linear activation function within the neural network, adding layers does not increase the modeling capability of a neural network since it can always be reduced to an affine linear mapping of the input onto the output space.

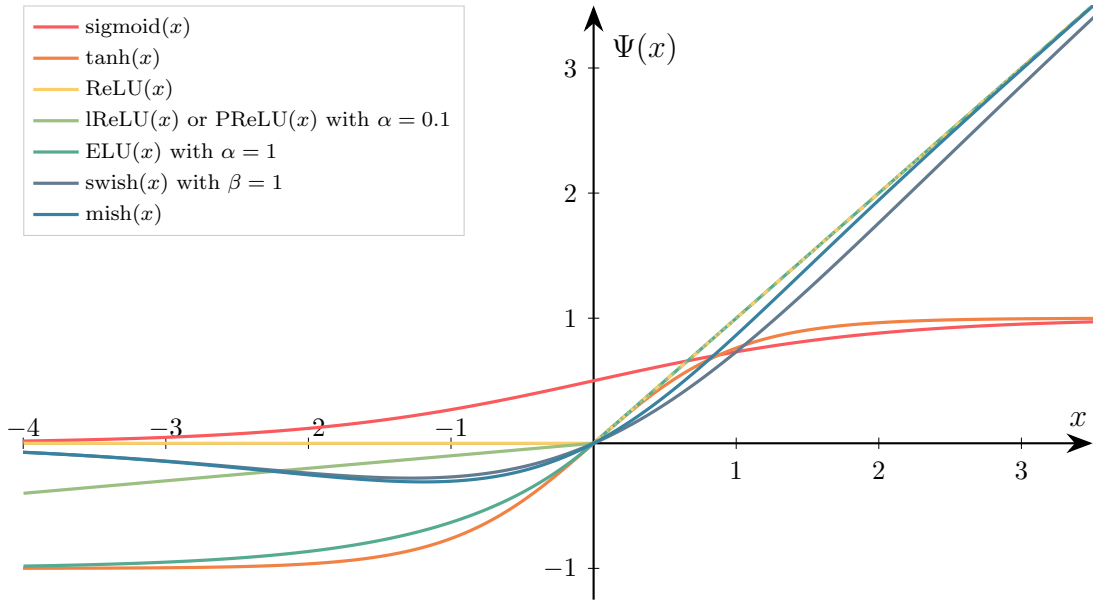

Figure 2.2: Popular activation functions $\Psi(x)$ with default parameters.

2.3 Activation Functions

As demonstrated by Equation (2.4), employing non-linear activation functions $\Psi(x)$ is decisive for modelling more complex relationships between input and output. While early works following LeNet [117, 118] mainly relied on $\text{sigmoid}(x) = 1/[1+\exp(-x)]$ or $\tanh(x) = 2\,\text{sigmoid}(2x) - 1$ as activation function, $\text{sigmoid}(x)$ is nowadays almost exclusively employed when transforming x into a probability distribution. With the *rectified linear unit* (ReLU) ($\text{ReLU}(x) = \max(0, x)$), Nair *et al.* [159] proposed a simple yet effective activation function, which solved the vanishing gradients for $x \gg 0$, in which gradients tend to converge to zero due to numerical instabilities (*cf.* Section 2.4). Besides, the ReLU is very resource-efficient compared to $\text{sigmoid}(x)$ or $\tanh(x)$. Consequently, the training of ANNs with ReLUs as activation functions is more efficient and yields better results. However, the zero gradient for $x < 0$ together with the shift of mean unit activation towards positive values constitutes a downside, which was tackled by multiple subsequent works. *E.g.*, leaky ReLU (LReLU) [141] $\text{lReLU}(x) = \max(\alpha x, x)$ uses a small predefined slope α for negative x, whereas parameterized ReLU (PReLU) considered α a trainable parameter [69]. The exponential linear unit (ELU) [27] is characterized by $\alpha\,(\exp(x) - 1)$ for $x < 0$, which further ensures a noise-robust deactivation state. Recently, the search for a more powerful activation function yielded $\text{swish}(x) = x\,\text{sigmoid}(\beta x)$ with trainable β [174] and $\text{mish}(x) = x \tanh(\ln(1+\exp(x)))$ [153].

As depicted in Figure 2.2, all activation functions except $\text{sigmoid}(x)$ and $\tanh(x)$ follow roughly the same principle: Small activations ≈ 0 for $x < 0$ and large activations for $x > 0$. In this way, the network can create sparse activations, which are suitable to encode the existence of certain features.

2.4 Optimization

During ANN training, the objective is to find the optimal weights $\theta^* = \left\{ \boldsymbol{W}^{[l]^*}, \boldsymbol{b}^{[l]^*} \right\}_{l=1}^{L}$ such that the network resembles the underlying problem $\boldsymbol{x} \mapsto y$ given a dataset $\mathcal{X} = \left\{ \left(\boldsymbol{x}^{(n)}, y^{(n)} \right) \right\}_{n=1}^{N}$ comprising N samples.[i] This can be reformulated into an optimization problem

$$\theta^* = \arg\min_{\theta} \mathscr{L}\left(\mathcal{X}, \theta\right), \tag{2.5}$$

where the loss function $\mathscr{L}\left(\mathcal{X}, \theta\right)$, which measures how well the network with the weights θ approximates the task $\boldsymbol{x} \mapsto y$, is minimized. For scalar regression tasks, the mean squared error (MSE) loss

$$\mathscr{L}\left(\mathcal{X}, \theta\right) = \frac{1}{|\mathcal{X}|} \sum_{n=1}^{|\mathcal{X}|} \left[\hat{y}^{(n)} - y^{(n)} \right]^2 = \frac{1}{N} \sum_{n=1}^{N} \left[\Theta_{\theta}^{[L]}(\boldsymbol{x}^{(n)}) - y^{(n)} \right]^2 \tag{2.6}$$

constitutes a popular choice with $\hat{y}^{(n)} = \Theta_{\theta}^{[L]}(\boldsymbol{x}^{(n)})$ denoting the network's output with the weights θ and the nth sample $\boldsymbol{x}^{(n)}$ at the input. While a system with only linear activation functions as in Equation (2.4) has – with the pseudo-inverse – a closed-form solution when minimizing Equation (2.6), arbitrary network structures require iterative methods.

Equation (2.5) defines a multi-dimensional optimization problem in the weight space defined by a loss function $\mathscr{L}\left(\mathcal{X}, \theta\right)$, which is chosen to be differentiable with respect to θ. This enables the employment of gradient descent to iteratively minimize Equation (2.5) by descending $\mathscr{L}\left(\mathcal{X}, \theta\right)$ in the direction of the gradient. Thus, $\boldsymbol{W}^{[l]}$ is updated by moving in the opposite direction of the current gradient $\nabla \boldsymbol{W}^{[l]}$ scaled by the learning rate η resulting in

$$\boldsymbol{W}^{[l]} \leftarrow \boldsymbol{W}^{[l]} - \eta \nabla \boldsymbol{W}^{[l]}. \tag{2.7}$$

In addition, the biases $\boldsymbol{b}^{[l]}$ are updated analogously.

There are multiple ways to initialize the weights before the training. A popular choice is using a Gaussian normal distribution with zero mean and a small standard deviation. More sophisticated methods involve considering the number of neurons in the previous layer [53] and combining it with the nonlinearity of the activation functions [69]. However, their detailed discussion is out of scope for this work.

The weight update in Equation (2.7) is prone to stagnate due to vanishing gradients or diverge if the gradients oscillate. Both behaviors can be addressed with a *momentum* term, which uses the exponential moving average over all previous gradients to increase the effective learning rate η if all gradients point in the same direction and decrease η in case of oscillating gradients. Multiple extensions (adaptive gradient (AdaGrad) [43], root mean square propagation (RMSProp) [78], Adadelta [259], adaptive moment estimation (Adam) [110]) have been proposed, among which Adam is still considered the most popular optimizer despite its age. In addition to the momentum term, Adam

[i] For the sake of simplicity, a scalar output is assumed.

employs the exponential moving average of the second moment of the gradient and utilizes bias correction to speed up the early training stages.

Regardless of the modifications on Equation (2.7) proposed by [43, 78, 110, 259], the gradient computation of the weight matrix $\nabla \boldsymbol{W}^{[l]}$ is always required. In case of a single sample $\mathcal{X} = \{(\boldsymbol{x}, y)\}$, $\nabla \boldsymbol{W}^{[l]}$ is computed by leveraging the chain rule of calculus and Equation (2.2) via

$$\begin{aligned} \nabla \boldsymbol{W}^{[l]} &= \frac{\partial \mathscr{L}\left(\mathcal{X}, \theta\right)}{\partial \boldsymbol{W}^{[l]}} = \frac{\partial \mathscr{L}\left(\left\{(\boldsymbol{x}, y)\right\}, \theta\right)}{\partial \Theta^{[l]}\left(\boldsymbol{x}\right)} \cdot \frac{\partial \Theta^{[l]}\left(\boldsymbol{x}\right)}{\partial \boldsymbol{W}^{[l]}} \\ &= \frac{\partial \mathscr{L}\left(\left\{(\boldsymbol{x}, y)\right\}, \theta\right)}{\partial \Theta^{[l]}\left(\boldsymbol{x}\right)} \cdot \Psi^{[l]\prime}\left(\boldsymbol{W}^{[l]} \Theta^{[l-1]}\left(\boldsymbol{x}\right) + \boldsymbol{b}^{[l]}\right) \cdot \Theta^{[l-1]}\left(\boldsymbol{x}\right)^{\mathsf{T}}. \end{aligned} \tag{2.8}$$

Hence, to calculate $\nabla \boldsymbol{W}^{[l]}$, the input $\boldsymbol{x}$ is propagated forwards through the network yielding the output $\hat{\boldsymbol{y}}$ and then backwards to obtain $\partial \mathscr{L}\left(\left\{(\boldsymbol{x}, y)\right\}, \theta\right) / \partial \Theta^{[l]}\left(\boldsymbol{x}\right)$, which explains why this technique is named *backpropagation*.

Equation (2.8) also reveals a limitation of backpropagation as the gradient is composed of two terms: 1) the backwards propagated gradient from the loss function $\mathscr{L}\left(\left\{(\boldsymbol{x}, y)\right\}, \theta\right)$; and 2) the output of the previous layer $\Theta^{[l-1]}\left(\boldsymbol{x}\right)$. Thus, to compute the gradient for all θ, the outputs of all layers $\Theta^{[l]}\left(\boldsymbol{x}\right)$ need to be retained in memory. In practice, computing the gradients for the entire dataset $\mathcal{X}$ is only feasible for small datasets or low-dimensional data due to hardware limitations. Contrarily, utilizing a single sample $(\boldsymbol{x}, y)$ results in a noisy and biased gradient. As a trade-off, *minibatch stochastic gradient descent* (SGD) is employed to calculate the gradient for a representative minibatch $\mathcal{X}_\mathrm{b} \subset \mathcal{X}$ of N_b samples. In this case, all gradients computed per sample from Equation (2.8) are averaged as in Equation (2.6) to obtain a more robust gradient.

The selection of a representative minibatch is crucial for efficient training. If the samples within a minibatch are too similar, minibatch SGD behaves similarly as if a single sample was used to calculate the gradient for the weight update step. To ensure variety within a minibatch, the entire dataset is always shuffled before traversing it. Moreover, a sample is only used once in every dataset traverse. The counter *epoch* indicates how often the whole dataset $\mathcal{X}$ was iterated through, while *step* refers to the weight update step executed for every minibatch $\mathcal{X}_\mathrm{b}$. Typically, the batch size N_b is chosen as large as possible such that the optimization step does not exceed the memory limitations. In this way, N_b gradients are averaged, which speeds up convergence. Besides, the calculations are more efficient due to the parallel computation of N_b samples.

One issue when training ANNs are *vanishing gradients*, *i.e.*, $\nabla \boldsymbol{W}^{[l]} \approx \boldsymbol{0}$. There are multiple apparent reasons in Equation (2.8) that can lead to a vanishing gradient: 1) the backpropagated gradient $\mathscr{L}\left(\left\{(\boldsymbol{x}, y)\right\}, \theta\right)$; 2) the derivative of the activation function $\Psi^{[l]\prime}\left(\cdot\right)$; and 3) the output of the previous layer $\Theta^{[l-1]}\left(\boldsymbol{x}\right)$. All three highly depend on the activation function $\Psi^{[l]}\left(\cdot\right)$. While the second term directly contains $\Psi^{[l]\prime}\left(\cdot\right)$, the first term encompasses all $\Psi^{[k]\prime}\left(\cdot\right)$ for $k = l+1, \ldots, L$. Thus, to compute the gradient of the weights in early layers, all gradients from deeper layers are multiplied, resulting in barely any weight update if various gradients are close to zero. Since these early layers typically

recognize basic patterns in the input data, the entire network training is impaired. This constitutes the main reason for dispensing with $\Psi(x) = \tanh(x)$ or $\Psi(x) = \text{sigmoid}(x)$ after ReLUs were introduced as ReLUs provide a constant gradient $\text{ReLU}'(x) = 1 \,\forall\, x > 0$. However, the ReLU's zero gradients for $x < 0$ can lead to entirely deactivated neurons as there is no gradient to reactivate them. Moreover, the term $\Theta^{[l-1]}(\boldsymbol{x})$ directly depends on $\Psi^{[l-1]}(\cdots)$, which can also lead to zero gradients. Despite the apparent disadvantage of only updating very few weights at every update step and parts of the network not contributing towards the prediction due to dead neurons, the advantages of zero gradients are fast computations and high numerical stability. To solve the issue of zero gradients in ReLUs, one of the multiple extensions discussed in Section 2.3 can be utilized.

Even though all optimizers with a momentum term account for oscillating or small gradients, the training typically reaches a stationary state after several epochs. By reducing the learning rate η after reaching such a state, the optimizer can find the minimum more efficiently in the current loss surface resulting in further improvement. Besides detecting a stationary state by considering the loss on a separate dataset and triggering a decrease, learning rates η are updated every epoch or step. While a learning rate update every epoch is independent of the batch size N_b, updating η every step results in independence from the number of samples of the training dataset N_train. One option is to multiply η with a factor at every update $\eta(i+1) \leftarrow \frac{1}{\gamma_\text{lr}}\eta(i)$ with $\gamma_\text{lr} > 1$. Alternatively, η is decreased by a factor γ_lr only for predefined values of i, *e.g.*, every five epochs. Loshchilov and Hutter [136] proposed a cosine-like learning rate decay together with resetting the learning rate schedule multiple times during the training to speed up the convergence. Since random initial weights can cause unstable training with a large η, slowly increasing η warms up the network such that training with a larger η is stable. The interested reader is referred to the analysis by Gotmare *et al.* [58] as a detailed analysis is out of scope for this work.

2.5 Convolutional Neural Networks

Fully connected neural networks with multiple layers, as introduced in Section 2.2, are powerful tools to solve various problems; however, when applied to images, their inherent properties raise several issues: 1) Fully connected layers do not leverage the spatial information within images, *i.e.*, the relationship of neighboring pixels is not considered. If the pixels of all images in a dataset were permuted in the same way, the training outcome of a multilayer fully connected neural network would be similar even though the permuted images would be irrecognizable to us humans. Besides, the fully connected layer is not shift-equivariant, *i.e.*, shifting the position of an object within the input image does lead to the same shift in the output. This property is crucial for, *e.g.*, object detection or image-to-image translation tasks. 2) Using fully connected layers on images causes the weight matrices to become huge, which requires more data to obtain a well-generalizing network (see also Section 2.6). *E.g.*, an RGB image with a resolution of $100 \times 100\,\text{px}$ passed into a single fully connected layer with 1000 neurons requires $3 \cdot 10^7$ weights - more than most of the networks discussed in this work.

Similar to the perceptron (see Section 2.1), which tries to model neural connections in the human brain, the first neural model leveraging spatial dependency of nearby pixels – the neocognitron [49] – is also inspired by biology since it is based on experiments performed on cat's visual cortex [88]. Already in 1983, the neocognitron showed promising results when recognizing handwritten numbers. Nevertheless, only six years later, the first *convolutional neural network*s (CNNs) demonstrated an even greater potential when exploiting the spatial information in images [117]. Even decades after the first CNN, the convolutional layer is still considered an essential layer for computer vision tasks.

2.5.1 Convolutional Layer

In order to obtain a shift-equivariant operation, the weight is assigned solely based on the relation of the input pixel with respect to the output pixel, *i.e.*, the input pixel to the left of the target position always has the same weight regardless of the target position. By only considering pixels within a $K_\mathrm{H} \times K_\mathrm{W}$ vicinity of the target position, the network is forced to learn local patterns, resulting in a substantial reduction of the number of weights. Thus, this operation can be interpreted as multiple fully connected layers with shared weights and restricted to the vicinity of a pixel.

Given the input tensor $\boldsymbol{F}^{[l-1]} \in \mathbb{R}^{H^{[l-1]} \times W^{[l-1]} \times C^{[l-1]}}$ with $H^{[l-1]}$, $W^{[l-1]}$, and $C^{[l-1]}$ denoting the height, width, and the number of channels of $\boldsymbol{F}^{[l-1]}$, respectively, the output $\big[\widetilde{\boldsymbol{F}}^{[l]}\big]_{i,j,k}$ at every possible output position $i = 1, \ldots, H^{[l]}$, $j = 1, \ldots, W^{[l]}$, and the output channel $k = 1, \ldots, C^{[l]}$ is computed as

$$\big[\widetilde{\boldsymbol{F}}^{[l]}\big]_{i,j,k} = \big[\boldsymbol{b}^{[l]}\big]_k + \sum_{h=1}^{K_\mathrm{H}^{[l]}} \sum_{w=1}^{K_\mathrm{W}^{[l]}} \sum_{c=1}^{C^{[l-1]}} \big[\boldsymbol{W}^{[l]}\big]_{h,w,c,k} \big[\boldsymbol{F}^{[l-1]}\big]_{i+h-\Delta_\mathrm{H}^{[l]}-1, j+w-\Delta_\mathrm{W}^{[l]}-1, c}, \tag{2.9}$$

with $\boldsymbol{W}^{[l]} \in \mathbb{R}^{K_\mathrm{H}^{[l]} \times K_\mathrm{W}^{[l]} \times C^{[l-1]} \times C^{[l]}}$ and $\boldsymbol{b}^{[l]} \in \mathbb{R}^{C^{[l]}}$ denoting the weight matrix and bias vector, respectively, and $2\Delta_i^{[l]} + 1 = K_i^{[l]}$ for $i \in \{H, W\}$.[ii] In the context of convolutional layers, the weight matrix is also referred to as the filter or kernel. The filter size $K_\mathrm{H}^{[l]} \times K_\mathrm{W}^{[l]}$ determines the vicinity and is often chosen to be an odd number such that it is not required to employ symmetric padding [18] to cope with the inherent spatial shift of a convolution with an even filter size.

As depicted in Figure 2.3 (left), the convolution moves the filter $\boldsymbol{W}^{[l]}$ to every possible position in the input $\boldsymbol{F}^{[l-1]}$ and computes the dot product of $\big[\boldsymbol{W}^{[l]}\big]_{:,:,:,k}$ with the input volume in the $K_\mathrm{H}^{[l]} \times K_\mathrm{W}^{[l]}$ vicinity for $k = 1, \ldots, C^{[l]}$. Along the width $W^{[l-1]}$, the filter can be aligned to $W^{[l-1]} - K_\mathrm{W}^{[l]} + 1$ positions resulting in an underrepresentation of borders and a shrinking output size. Due to its convenience of maintaining the width and height during the operation, *zero-padding* is frequently applied to every border with $P_\mathrm{W} = \Delta_\mathrm{W}^{[l]}$

[ii] In contrast to Equation (2.2), the activation function $\Psi^{[l]}(\cdot)$ is considered an independent layer as it does not always follow directly after the convolution (see also Figure 3.4).

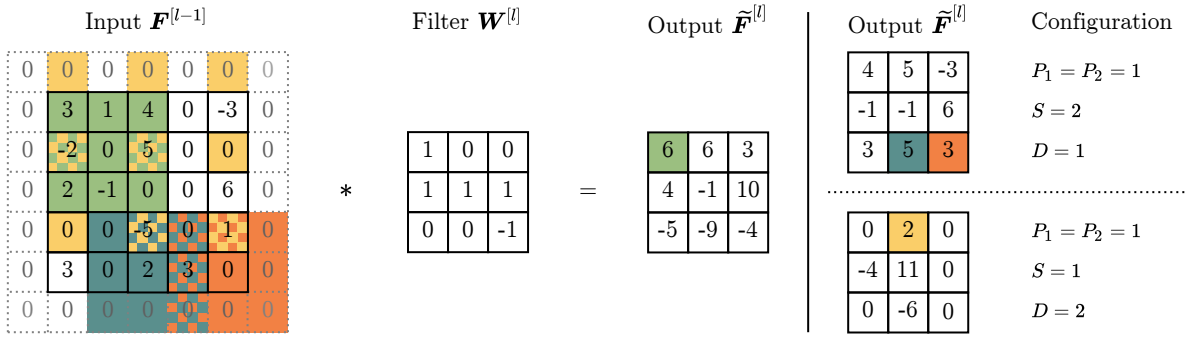

Figure 2.3: Left: Convolution of the input $\boldsymbol{F}^{[l-1]} \in \mathbb{R}^{5\times5}$ with a 3×3 filter $\boldsymbol{W}^{[l]}$ and default configuration. Right: The output $\widetilde{\boldsymbol{F}}^{[l]}$ for two configurations with zero-padding along both axes $P_\mathrm{H} = P_\mathrm{W} = 1$ and different strides S and dilation factors D. All configurations result in an output $\widetilde{\boldsymbol{F}}^{[l]}$ of size 3×3, however, with different values. The colors indicate which input pixels were used to compute the corresponding output pixel. For the sake of simplicity, the bias is omitted.

pixels along the width and $P_\mathrm{H} = \Delta_\mathrm{H}^{[l]}$ pixels along the height.[iii] Since zero-padding only adds zeros, *i.e.*, no artificial activations in $\widetilde{\boldsymbol{F}}^{[l]}$, no unwanted artifacts are introduced. If not stated otherwise, zero-padding is employed throughout this work for all convolutional layers such that $W^{[l]} = W^{[l-1]}$ and $H^{[l]} = H^{[l-1]}$.

However, in some scenarios, it is desired to reduce the spatial footprint after a convolution. By introducing a *stride* S, the convolution is only evaluated every S pixels, which leads to an output width $W^{[l]} = \frac{W^{[l-1]} - K_\mathrm{W}^{[l]} + 2P_\mathrm{W}}{S} + 1$ with the parameters $K_\mathrm{W}^{[l]}$ and $2P_\mathrm{W}$ chosen such that $(W^{[l-1]} - K_\mathrm{W}^{[l]} + 2P_\mathrm{W}) \mod S = 0$. The top parameter configuration in Figure 2.3 (right) illustrates an example incorporating padding and stride.

Despite its name, the convolutional layer, according to Equation (2.9), is actually not performing a convolution, as known from signal theory, but a cross-correlation between $\left[\boldsymbol{W}^{[l]}\right]_{:,:,:,k}$ and $\boldsymbol{F}^{[l-1]}$ in a $K_\mathrm{H}^{[l]} \times K_\mathrm{W}^{[l]}$ vicinity for multiple positions. Nevertheless, both operations are closely related since Equation (2.9) also represents a convolution of $\boldsymbol{F}^{[l-1]}$ yet with a horizontally and vertically flipped kernel $\boldsymbol{W}^{[l]}$. While the channels of an RGB image correspond to the red, green, and blue colors, every channel of the output $\widetilde{\boldsymbol{F}}^{[l]}$ of a convolutional layer, a so-called *feature map*, indicates the presence of particular shapes in the input $\boldsymbol{F}^{[l-1]}$. In earlier layers, these shapes are relatively simple and often just involve edges, whereas more complex shapes, *e.g.*, circles or squares, are detected in deeper layers.

Even though every convolution only considers pixels within a $K_\mathrm{H} \times K_\mathrm{W}$ vicinity, this restriction is lifted by stacking multiple convolutional layers, which results in an increasing *receptive field*, *i.e.*, the region of the input image $\boldsymbol{X}$ affecting a particular position in $\widetilde{\boldsymbol{F}}^{[l]}$.

[iii] Even though parameters may vary for every convolutional layer, the dependency on the layer $^{[l]}$ is omitted to improve readability.

In fact, by stacking l convolutions, which each considers a 3×3 vicinity, the receptive field of the lth convolution is $l(K_i - 1) + 1 = 2l + 1$ along both axes. When employing a layer with $S > 1$, the receptive field increases further. Thus, in order to encompass all pixels of the input $\boldsymbol{X}$, the network's receptive field must be substantially more extensive than the input size, as otherwise, border effects lead to an underrepresentation of pixels in the corners.

Another technique to increase the receptive field is by *dilating* the kernel with a dilation factor $D > 1$ [253], *i.e.*, the kernel is spread out by introducing $D - 1$ spaces between every kernel element, which is indicated by the bottom parameter configuration in Figure 2.3 (right). While the receptive field of a single dilated convolution is extended to $D(K_i - 1) + 1$, the sparse kernel creates gridding artifacts, which need to be accounted for with custom degridding techniques [227] or prior smoothing [149].

2.5.2 Transposed Convolution

Any convolution layer can also be rewritten as a sparsely connected layer with shared weights following the notation of Equation (2.3). *E.g.*, considering Figure 2.3 (left) with the input $\boldsymbol{F}^{[l-1]}$ and the output $\widetilde{\boldsymbol{F}}^{[l]}$ flattened into a column vector $\boldsymbol{f}^{[l-1]} \in \mathbb{R}^{25\times 1}$ and $\tilde{\boldsymbol{f}}^{[l]} \in \mathbb{R}^{9\times 1}$, respectively. Then, the doubly block-circulant matrix – a special case of a Toeplitz matrix *cf.* [59] – $\widetilde{\boldsymbol{W}}^{[l]} \in \mathbb{R}^{9\times 25}$ of $\boldsymbol{W}^{[l]}$ can be used to describe the convolution

$$\tilde{\boldsymbol{f}}^{[l]} = \widetilde{\boldsymbol{W}}^{[l]} \boldsymbol{f}^{[l-1]}. \tag{2.10}$$

Despite its large size, $\widetilde{\boldsymbol{W}}^{[l]}$ is typically very sparse and contains redundant information as every row $\left[\widetilde{\boldsymbol{W}}^{[l]}\right]_{k,:}$ is filled with the same coefficients $\left[\boldsymbol{W}^{[l]}\right]_{i,j}$, yet distributed at different positions such that the matrix-vector multiplication represents the convolution. Thus, only $K_{\mathrm{H}}^{[l]} K_{\mathrm{W}}^{[l]}$ out of $H^{[l-1]} W^{[l-1]}$ elements in $\left[\widetilde{\boldsymbol{W}}^{[l]}\right]_{k,:}$ are unequal zero. For the general case, as in Equation (2.9), $\widetilde{\boldsymbol{W}}^{[l]}$ is of size $H^{[l]} W^{[l]} C^{[l]} \times H^{[l-1]} W^{[l-1]} C^{[l-1]}$ with a bias vector $\widetilde{\boldsymbol{b}} \in \mathbb{R}^{H^{[l]} W^{[l]} C^{[l]}}$. Moreover, by altering the position of the $K_{\mathrm{H}}^{[l]} K_{\mathrm{W}}^{[l]}$ elements in $\left[\widetilde{\boldsymbol{W}}^{[l]}\right]_{k,:}$, even special cases involving strides, and dilated convolutions can be represented as in Equation (2.10).

As analyzed in Section 2.5, a convolution typically maintains or reduces the spatial size of a feature map. However, some tasks, such as super-resolution, image-to-image translation, or semantic segmentation, require the spatial resolution to be increased,[iv] which is accomplished by transposing the weight matrix $\widetilde{\boldsymbol{W}}^{[l]}$ in Equation (2.10) yielding

$$\tilde{\boldsymbol{f}}^{[l]} = \widetilde{\boldsymbol{W}}^{[l]\mathsf{T}} \boldsymbol{f}^{[l-1]}. \tag{2.11}$$

[iv] In image-to-image translation and semantic segmentation, the input resolution is typically reduced and increased forming an hourglass structure. Thus, increasing spatial resolution is necessary despite identical input and output resolutions.

This so-called *transposed convolution* is illustrated by Figure 2.4 (left) with the original weight matrix $\boldsymbol{W}^{[l]} \in \mathbb{R}^{K_{\mathrm{H}}^{[l]} \times K_{\mathrm{W}}^{[l]} \times C^{[l-1]} \times C^{[l]}}$. While for the standard convolution (*cf.* Equation (2.9)), a $K_{\mathrm{H}}^{[l]} \times K_{\mathrm{W}}^{[l]}$ vicinity in the input is considered for computing the value of a single output pixel, transposed convolutions operate fundamentally differently. Still, both are related since the transposed convolution corresponds to the derivation of a regular convolution. Thus, it is only necessary to switch forward and backward operations of a convolutional layer to obtain a transposed convolutional layer.

Considering the kth output channel of a transposed convolution, the filter $\left[\boldsymbol{W}^{[l]}\right]_{:,:,c,k}$ is scaled by every input value $\left[\boldsymbol{F}^{[l-1]}\right]_{i,j,c}$ and summed over all $C^{[l-1]}$ input channels to obtain the intermediate output

$$\left[\widetilde{\boldsymbol{G}}^{[l]}\right]_{i,j,k,:,:} = \sum_{c=1}^{C^{[l-1]}} \left[\boldsymbol{F}^{[l-1]}\right]_{i,j,c} \left[\boldsymbol{W}^{[l]}\right]_{:,:,c,k}. \tag{2.12}$$

Then, the final output $\widetilde{\boldsymbol{F}}^{[l]}$ is obtained by placing neighboring $\left[\widetilde{\boldsymbol{G}}^{[l]}\right]_{i,j,k,:,:}$ separated by the stride S that is followed by a summation of overlapping pixels. The overlap also allows a different point of view onto the transposed convolution since – unlike for the regular convolution – more than a single position i, j of the kernel $\boldsymbol{W}^{[l]}$ can influence an output pixel. For $S \geq K_{\mathrm{H}}^{[l]}$ and $S \geq K_{\mathrm{W}}^{[l]}$, there is no overlap between neighboring $\left[\widetilde{\boldsymbol{G}}^{[l]}\right]_{i,j,k,:,:}$ in the output $\widetilde{\boldsymbol{F}}^{[l]}$ (*cf.* Figure 2.4 (left) for $S = 2$). Typically, a doubling of the spatial resolution and overlap is desired as only then values of multiple input pixels are considered to obtain an output pixel's value. Hence, zero-padding with P_{H} and P_{W} can be employed similarly to the convolutional layer in order to obtain an exact doubling of the spatial resolution computed by $W^{[l]} = (W^{[l-1]} - 1) \cdot S + K_{\mathrm{W}}^{[l]} - 2P_{\mathrm{W}}$.

The overlap also reveals a downside when dealing with transposed convolutions with a filter size not being a multiple of the stride $S > 1$. *E.g.*, considering $S = 2$ and $K_{\mathrm{H}}^{[l]} = K_{\mathrm{W}}^{[l]} = 3$, the 3×3 dimensional intermediate outputs $\left[\widetilde{\boldsymbol{G}}^{[l]}\right]_{i,j,k,:,:}$ are placed $S = 2$ pixels apart in $\widetilde{\boldsymbol{F}}^{[l]}$. Thus, the center element $\left[\widetilde{\boldsymbol{G}}^{[l]}\right]_{i,j,k,2,2}$ is not affected by overlapping, whereas all surrounding elements are altered by the neighboring $\left[\widetilde{\boldsymbol{G}}^{[l]}\right]_{i,j,k,:,:}$.[v] Due to this nonuniform overlap, checkerboards artifacts occur [165], which are even more prominent if multiple transposed convolutions are employed in the CNN. Odena *et al.* [165] further found that subtle checkerboard artifacts are present even if $K_{\mathrm{H}}^{[l]}$ and $K_{\mathrm{W}}^{[l]}$ are multiples of S. Therefore, when employing transposed convolutions, it is very difficult to avoid checkerboard artifacts entirely as they are easily created during training.

Transposed convolutions can also be interpreted as an upsampling with a trainable kernel. Hence, they provide more flexibility compared to the nearest neighbor, bi-linear or bi-cubic interpolations. However, when combining the upsampling alternatives with a convolution layer, similar complexity is achieved and checkerboard artifacts are prevented [165].

[v] This is only valid for elements not being affected by border effects; hence, $i \notin \{1, H^{[l-1]}\}$ and $j \notin \{1, W^{[l-1]}\}$.

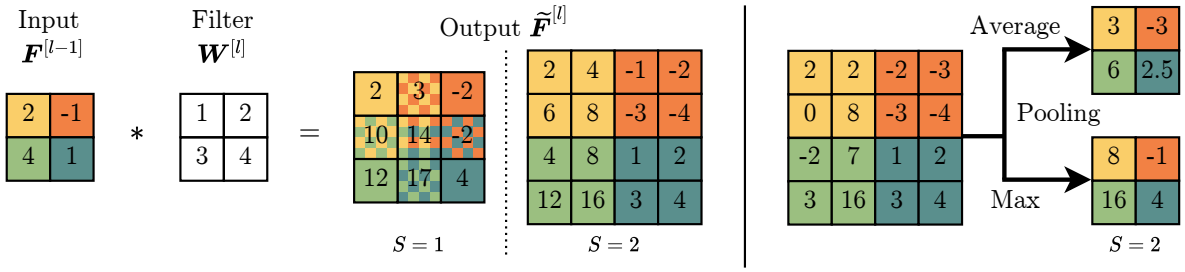

Figure 2.4: Transposed convolution of the input $\boldsymbol{F}^{[l-1]} \in \mathbb{R}^{2\times2}$ with a 2×2 filter $\boldsymbol{W}^{[l]}$ and stride $S \in \{1, 2\}$ (left). For the sake of simplicity, the bias is omitted. Average and maximum pooling layer (right). The colors indicate which input pixels were used to compute the corresponding output pixel.

2.5.3 Pooling

The pooling layer constitutes a parameter-free operation within a CNN and serves two purposes: 1) increasing the receptive field; and 2) decreasing the feature map size, which makes the network less prone to overfitting and lowers its computational cost. The *average pooling* layer can be interpreted as a convolutional layer with a constant weight matrix $\boldsymbol{W} = \frac{1}{K_\mathrm{H} K_\mathrm{W}} \mathbb{1}_{K_\mathrm{H}, K_\mathrm{W}}$, which is applied – unlike in convolutional layers – to every input feature map separately. In this way, the values within a $K_\mathrm{H} \times K_\mathrm{W}$ vicinity are averaged and the number of channels is maintained. Another popular option is the *maximum pooling* layer, which outputs the maximum in a $K_\mathrm{H} \times K_\mathrm{W}$ vicinity. The difference between both pooling operations is illustrated in Figure 2.4 (right).

Typically, pooling layers are used with stride $S = K_\mathrm{H} = K_\mathrm{W}$ to obtain non-overlapping pooling and a dimensionality reduction. Besides, pooling allows the model to become (slightly) shift-invariant as a slight spatial shift only results in minor changes in feature maps, particularly after multiple pooling layers. A special case of pooling is *global pooling*, which calculates the average or maximum over the entire spatial dimensions and thereby dispenses with any spatial information present in the input. Usually, it is employed after all convolutional layers to obtain a feature vector indicating the average or maximum activation within every feature map, *i.e.*, the presence of specific shapes within the input regardless of their position.

With the first CNN [117], a variant of average pooling was utilized. Krizhevsky *et al.* [112] used overlapping maximum pooling with $K_\mathrm{H} = K_\mathrm{W} = 3$ and $S = 2$. Nowadays, average and maximum pooling are equally popular due to their negligible influence on the results. Nevertheless, multiple more complex pooling methods have been proposed, among which mixed pooling combines average and maximum pooling [252], and spatial pyramid pooling employs a spatial pyramid instead of global pooling to obtain a fixed-size feature representation [70]. Region of interest pooling is typically used in object detection tasks and involves pooling region proposals of an arbitrary size into a fixed-size feature map for further processing [52]. More recent region pooling methods fuse the information based on the predicted object's corners [114] or the object's corners and center [41]; however, their discussion is out of scope for this work.

2.6 Improving Generalization

When training an ANN with a dataset $\mathcal{X}_{\text{train}}$, it is desired to obtain the best performance on a separate dataset $\mathcal{X}_{\text{test}}$ comprising different samples ($\mathcal{X}_{\text{train}} \cap \mathcal{X}_{\text{test}} = \emptyset$). In this way, the measured performance resembles the actual performance in real-world applications.

However, when optimizing the network to obtain the best results on $\mathcal{X}_{\text{train}}$, the results on $\mathcal{X}_{\text{test}}$ might be better when choosing different weights θ, which leads to suboptimal performance on $\mathcal{X}_{\text{train}}$. In such a scenario, the model *overfits* on $\mathcal{X}_{\text{train}}$ and does not *generalize* well on unknown data $\mathcal{X}_{\text{test}}$, which can be verified by comparing the respective losses

$$\mathscr{L}\left(\mathcal{X}_{\text{test}}, \theta^*\right) \gg \mathscr{L}\left(\mathcal{X}_{\text{train}}, \theta^*\right) \approx 0. \tag{2.13}$$

Overfitting occurs if $\mathcal{X}_{\text{train}}$ is not sufficiently diverse for the network to learn the underlying patterns or – seen from the network's perspective – if the network is too complex and thus memorizes a direct input-output mapping instead of learning the underlying relationship. If the amount of data does not match the network's complexity, the network will try to memorize the input and thereby the training results in a loss $\mathscr{L}\left(\mathcal{X}, \theta\right) \approx 0$. *E.g.*, in contrast to learning what makes a cat differ from a dog, a sufficiently complex network directly learns the images of a cat.

In fact, overfitting is ubiquitous in ANN, as demonstrated by Zhang *et al.* [262]. The authors showed that even simple networks trained with "images" containing shuffled pixels, *i.e.*, with no apparent spatial dependencies, or shuffled labels still manage to obtain 100% accuracy on $\mathcal{X}_{\text{train}}$ and an accuracy corresponding to random guessing on $\mathcal{X}_{\text{test}}$. So, the network can predict the labels perfectly despite the absence of any relationship between images and their labels. This vast generalization gap demonstrates that every network is sufficiently complex and thus prone to overfit. Hence, it is crucial to present more obvious links between images and labels, *i.e.*, shared patterns in images for a given label, forcing the networks to focus on generalizing patterns rather than sample-specific nuances.

There are two straightforward approaches to cope with overfitting: increasing the dataset size or reducing the complexity of the neural network. The former can be achieved through data augmentation, *i.e.*, the amount of data is artificially increased by flipping, randomly cropping, or changing brightness, contrast, saturation *etc.* (see also Section 3.5.1). Network complexity encompasses multiple aspects and is hard to grasp. *E.g.*, a single layer with a huge number of neurons is typically not sufficient to solve any task, and using multiple layers only improves complexity if non-linear activation functions are employed (*cf.* Equation (2.4)). Thus, network complexity involves – among others – the number of trainable parameters θ, values taken by the parameters, activation functions, and depth.

Increasing complexity can also lead to less overfitting if the predictions from multiple networks are averaged, even when they are trained on the same dataset. Since every network is optimized to a different set of optimal weights θ^* due to random weight initialization and shuffling of $\mathcal{X}_{\text{train}}$, the *ensemble* of networks is more robust and generalizes better. Despite their popularity when CNNs were adapted to many new computer

vision tasks, nowadays, their usage is mainly limited to challenges, in which the overall performance is slightly boosted by simply employing an ensemble.

Besides the amount of training data and network complexity, there are multiple techniques to manipulate the network or the training to improve generalization. Among these, the most popular are introduced in the following.

2.6.1 Early Stopping

Early stopping is a simple albeit effective method to mitigate overfitting, which utilizes an additional disjoint validation dataset $\mathcal{X}_{\text{val}}$ with $\mathcal{X}_{\text{train}} \cap \mathcal{X}_{\text{val}} = \mathcal{X}_{\text{test}} \cap \mathcal{X}_{\text{val}} = \emptyset$. Then $\mathcal{X}_{\text{val}}$ is employed to terminate the training on $\mathcal{X}_{\text{train}}$ if $\mathscr{L}(\mathcal{X}_{\text{val}}, \theta)$ begins to rise.[vi] In this way, the loss is not optimal on $\mathcal{X}_{\text{train}}$ yet yields better performance on $\mathcal{X}_{\text{test}}$ since the network's weights θ are not too specialized on $\mathcal{X}_{\text{train}}$, which improves generalization on unseen data $\mathcal{X}_{\text{test}}$.

2.6.2 Weight Decay

One aspect contributing to model complexity is the value range taken by the trainable parameters θ. If a network can choose θ freely within $\mathbb{R}$, the weights after training θ^* optimally resemble the relationship in $\mathcal{X}_{\text{train}}$ given the network structure. However, it also increases susceptibility to slightly varying input data as in unseen data $\mathcal{X}_{\text{test}}$ (*cf.* Equation (2.13)). By constraining the value space of θ, the network cannot optimally approximate the task given by $\mathcal{X}_{\text{train}}$, resulting in inferior performance, which is recognizable by the increased training loss $\mathscr{L}(\mathcal{X}_{\text{train}}, \theta)$. However, the network is less specialized on $\mathcal{X}_{\text{train}}$, which leads to improved generalization and ultimately to better performance on $\mathcal{X}_{\text{test}}$.

This concept is implemented under the term *weight decay*. Formally, weight decay adds the L_2 norm of all weights, denoted by $\mathscr{L}_{\text{reg}}$, as a penalty term to the total loss $\mathscr{L}_{\text{tot}}$, which is used to optimize the network

$$\mathscr{L}_{\text{tot}}(\mathcal{X}, \theta) = \mathscr{L}(\mathcal{X}, \theta) + \lambda_{\text{reg}} \underbrace{\left[\sum_{\boldsymbol{W}, \boldsymbol{b} \in \theta} \|\boldsymbol{W}\|^2 + \|\boldsymbol{b}\|^2 \right]}_{\mathscr{L}_{\text{reg}}}, \tag{2.14}$$

with λ_{reg} denoting a regularization constant to balance the losses. In this way, the network needs to find an optimal trade-off between minimizing the objective defined by $\mathscr{L}(\mathcal{X}, \theta)$ and utilizing large weights to achieve it. In contrast to clipping all values exceeding a given threshold, this soft regularization also affects small weights and allows the network to balance the absolute value among all weights.

[vi] Besides the loss $\mathscr{L}(\mathcal{X}_{\text{val}}, \theta)$, any other stopping criteria based on $\mathcal{X}_{\text{val}}$, such as accuracy, can be used to measure the generalization gap.

Besides the L_2 norm, L_1 norm can also be employed to allow sparser weight distributions, which is particularly useful in the presence of noisy data. With the L_2 norm, outliers are penalized, resulting in more uniformly distributed weights. Thus, weight decay with the L_2 norm mitigates the creation of multiple spurious associations during training, which ultimately alleviates the dependency on $\mathcal{X}_{\text{train}}$ and improves generalization on $\mathcal{X}_{\text{test}}$.

2.6.3 Dropout

An important objective in ANN training is robustness to small perturbations in the data, *i.e.*, a small change in the input pixels' value should not lead to different predictions. To improve such robustness, Srivastava *et al.* [197] proposed to inject noise into different layers of the neural network. Their technique is named *dropout* since neurons – including all incoming and outgoing connections – are dropped during training with a probability p_{d}. This behavior is implemented by adding a new dropout layer,[vii] which modifies the ith neuron's output value $\Theta^{[l]}(\boldsymbol{x})$ of a preceding hidden layer at depth l

$$\left[\Theta_{\text{D}}^{[l]}(\boldsymbol{x})\right]_i = \begin{cases} 0 & \text{with probability } p_{\text{d}} \\ \frac{[\Theta^{[l]}(\boldsymbol{x})]_i}{1-p_{\text{d}}} & \text{otherwise} \end{cases} \qquad \forall\, i. \tag{2.15}$$

If the value is retained, it is scaled with $1/(1-p_{\text{d}})$ to guarantee that the expected value after adding the dropout layer remains unchanged $\mathbb{E}\left[\Theta_{\text{D}}^{[l]}(\boldsymbol{x})\right] = \Theta^{[l]}(\boldsymbol{x})$. This is equivalent to scaling the weights at test time with $1-p_{\text{d}}$ as proposed by the authors [197]. However, compensating the dropout during the training as in Equation (2.15) allows a more convenient use of identical weights during training and testing.

Dropout can also be interpreted as sampling different thinned-out networks from the original network for every gradient update. Since the network does not know which neuron is dropped, it is forced to distribute detections among various neurons and base its decision on multiple neurons. Hence, a certain degree of redundancy is induced, which ultimately leads to improved generalization. Despite their effectiveness in fully connected networks, dropout is hardly incorporated in convolutional layers due to the spatial dependency of different pixels. While the gradient is zeroed for all weights associated with the dropped out units in fully connected layers, filter weights are still updated through backpropagation when dropping out pixels in feature maps as they are influenced by multiple pixels. This is also apparent in Equation (2.10) since the trainable weights occur in various rows of $\widetilde{\boldsymbol{W}}^{[l]}$. Thus, a variation of the original dropout, such as spatial dropout [205], must be incorporated to obtain similar behavior.

[vii] Even though an activation function, dropout, and batch normalization (see Section 2.6.4) are layers, both are not counted towards the overall depth of a network L as they are considered as part of the preceding layer.

2.6.4 Batch Normalization

One essential technique for efficient ANN training – or in fact, when training any classifier – is input normalization [119]. The idea is to provide input data with zero mean and unit variance to the network in order to avoid over-prioritization of individual inputs. However, the initially normalized data is not maintained after several layers of processing as not every feature (map) outputs values in the same range. Moreover, most activation functions, including the popular ReLU and its variations (see Section 2.3), do have an expected value $\mathbb{E}[\Psi(x)] \neq 0$ and thus provoke a drift of the layers' output values towards the positive value range, which is accumulated throughout the network. This drift also leads to unequal weight updates since the output of every layer directly influences its gradient (*cf.* Equation (2.8)). Generally, the gradient for a given weight $\nabla \boldsymbol{W}^{[l]}$ is computed during backpropagation under the assumption that all remaining weights remain constant. However, in practice, all weights are updated simultaneously, which requires lower learning rates η or maintaining the output distribution of every layer roughly constant to ensure a stable training.

Thus, it is no surprise that a drift of the distribution of the layers' output values – the so-called *internal covariate shift* – throughout the network can hamper convergence, as postulated by Ioffe and Szegedy [96]. To alleviate this issue, they proposed the *batch normalization* (BN) layers, which inserts normalization within the network based on the statistics of every minibatch $\mathcal{X}_\mathrm{b}$. Instead of normalizing the outputs of an intermediate layer $\tilde{\Theta}^{[l]}(\boldsymbol{x})$ to zero mean and unit variance, BN introduces two trainable parameters, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, denoting the target mean and variance after normalization. Then, the BN layer can be mathematically written as

$$\left[\tilde{\Theta}^{[l]}_\mathrm{BN}(\boldsymbol{x})\right]_i = [\boldsymbol{\gamma}]_i \frac{\left[\tilde{\Theta}^{[l]}(\boldsymbol{x})\right]_i - [\boldsymbol{\mu}_\mathrm{BN}]_i}{\sqrt{[\boldsymbol{\sigma}^2_\mathrm{BN}]_i + \epsilon_\mathrm{BN}}} + [\boldsymbol{\beta}]_i\,, \tag{2.16}$$

with $\boldsymbol{\mu}_\mathrm{BN}$ and $\boldsymbol{\sigma}^2_\mathrm{BN}$ denoting the mean and variance of the minibatch $\mathcal{X}_\mathrm{b}$, and ϵ_BN being a constant to account for numerical instabilities due to small variances $\boldsymbol{\sigma}^2_\mathrm{BN}$, which is typically chosen in the orders of 10^{-3}. Formally, $\boldsymbol{\mu}_\mathrm{BN}$ and $\boldsymbol{\sigma}^2_\mathrm{BN}$ are calculated as

$$\boldsymbol{\mu}_\mathrm{BN} = \frac{1}{N_\mathrm{b}} \sum_{\boldsymbol{x}\in\mathcal{X}_\mathrm{b}} \tilde{\Theta}^{[l]}(\boldsymbol{x}) \quad \text{and} \tag{2.17}$$

$$\boldsymbol{\sigma}^2_\mathrm{BN} = \frac{1}{N_\mathrm{b}} \sum_{\boldsymbol{x}\in\mathcal{X}_\mathrm{b}} \left(\tilde{\Theta}^{[l]}(\boldsymbol{x}) - \boldsymbol{\mu}_\mathrm{BN}\right) \odot \left(\tilde{\Theta}^{[l]}(\boldsymbol{x}) - \boldsymbol{\mu}_\mathrm{BN}\right). \tag{2.18}$$

For a fully connected layer with $M^{[l]}$ neurons, BN introduces $2M^{[l]}$ new trainable weights since every neuron is normalized separately. When employing BN in convolutional layers, Equations (2.16) to (2.18) are adapted such that every feature map $\left[\widetilde{\boldsymbol{F}}^{[l]}\right]_{:,:,i}$ is normalized separately. In this way, its computational footprint is relatively low compared to the large trainable weight matrices $\boldsymbol{W}$.

Typically, BN is utilized in every layer $\tilde{\Theta}^{[l]}(\boldsymbol{x})$ of the ANN and positioned before the activation function $\Psi(\cdot)$. Thus, $\boldsymbol{\beta}$ acts as a bias $\boldsymbol{b}$, making $\boldsymbol{b}$ obsolete when BN

is employed together with a fully connected or a convolutional layer. Also, $\boldsymbol{\gamma}$ is often omitted if a ReLU follows the BN since the scaling can be applied in the next layer. In order to leverage meaningful minibatch statistics, the minibatch size N_b must be sufficiently large. In fact, for the extreme case of $N_\mathrm{b} = 1$, training is completely stopped in fully connected layers as $\boldsymbol{\mu}_\mathrm{BN} = \tilde{\Theta}^{[l]}(\boldsymbol{x})$. Multiple extensions to cope with small N_b proposed normalizing across all feature maps for every sample [7], across all feature maps and all samples [208], or across a group of feature maps for every sample [236].

Equation (2.16) also reveals the dependency of the output of every training sample $\boldsymbol{x}$ on the remaining samples in $\mathcal{X}_\mathrm{b}$. This has a positive effect during training as it induces noise and makes the BN further act as regularization. However, to obtain deterministic values independent of N_b at inference, *i.e.*, when not training the weights as during evaluation, exponential moving averages of $\boldsymbol{\mu}_\mathrm{BN}$ and $\boldsymbol{\sigma}^2_\mathrm{BN}$ are stored during training with momentum α_BN, and used for inference [95].

BN fundamentally impacts ANN training in multiple ways. It mitigates the network's internal covariate shift addressing vanishing and exploding gradients, which enables the training of deeper networks. Santurkar *et al.* [185] attributed the success of BN to the smoother landscape of the optimization problem it creates, making the gradients more predictive. Regardless the reasons for its success, the introduction of BN ultimately enables training of deeper networks with larger learning rates and increases robustness to weight initialization and thus constitutes a key ingredient in most ANNs.

2.7 Generative Adversarial Networks

To this point, this work has only considered *discriminative* tasks, in which a mapping of a sample $\boldsymbol{X}$ onto the target class (classification) or value y (regression) is approximated by the network. While such tasks can achieve remarkable performance by straightforwardly stacking the layers introduced throughout this chapter and employing a suitable loss function, this approach fails to deliver satisfying results when the generation of new data is involved. The biggest challenge of such *generative* tasks, *e.g.*, creating synthetic data resembling a given distribution, image manipulation, *etc.*, is that the generated image $\boldsymbol{X}_\mathrm{fake}$ must be photo-realistic and indistinguishable from real images $\boldsymbol{X}_\mathrm{real}$. Hence, these tasks require to cautiously design a loss function that is capable of measuring the images' realism and imitating the judgment of humans. Even if pairwise ground-truth data is available during training, *e.g.*, for image inpainting tasks in which a mask is typically generated synthetically,[viii] obvious loss functions like the pixel-wise mean absolute error (MAE) or MSE turn out to be impractical as a slight spatial shift of high-frequency patterns leads to significant losses despite them being realistic. Moreover, it is difficult to grasp the ambiguity of $\boldsymbol{X}_\mathrm{fake}$ since generative tasks do not have a single acceptable solution. Thus, generative models minimize such losses by providing a blurry output $\boldsymbol{X}_\mathrm{fake}$ that lacks details and thereby is close to all possible realistic solutions,[ix] making $\boldsymbol{X}_\mathrm{fake}$ easy to identify as fake.

[viii] Image inpainting refers to the tasks of filling missing pixels of an image.

[ix] in terms of pixel-wise MAE or MSE

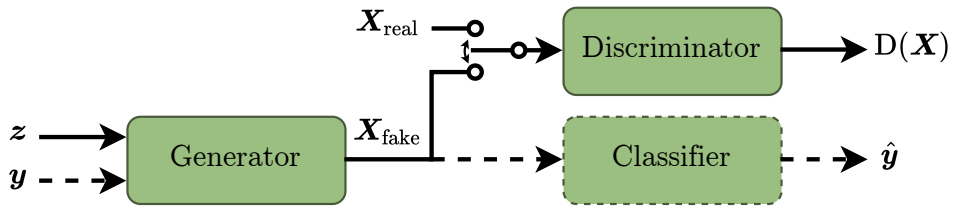

Figure 2.5: Architecture of a generative adversarial network (GAN): While the generator creates an image $\boldsymbol{X}_{\text{fake}}$ based on a noise vector $\boldsymbol{z}$, the discriminator tries to discern $\boldsymbol{X}_{\text{fake}}$ from a real image $\boldsymbol{X}_{\text{real}}$. The dashed components illustrate the addition of information $\boldsymbol{y}$ to create $\boldsymbol{X}_{\text{fake}}$ such that it satisfied $\boldsymbol{y}$ (conditional GAN).

In 2014, Goodfellow *et al.* [57] proposed *generative adversarial network*s (GANs) – a new concept of designing ANNs to generate realistic data. Their clever idea was to circumvent the cumbersome design of a loss function that imitates human judgment of realism by employing a separate ANN – the discriminator – that is trained with this task. Hence, the requirement of a loss function, which has to be meticulously designed to measure realism, is substituted by training a separate network with the sole task of discerning fake images $\boldsymbol{X}_{\text{fake}}$ from real images $\boldsymbol{X}_{\text{real}}$.

In this way, Goodfellow *et al.* [57] effectively expressed the generative task as two discriminative tasks: 1) the training of a *generator* to generate an image $\boldsymbol{X}_{\text{fake}}$, which is classified as real by the discriminator; and 2) the training of a *discriminator*, which can distinguish fake images $\boldsymbol{X}_{\text{fake}}$ from real images $\boldsymbol{X}_{\text{real}}$. This unique configuration of two components with contrary objectives creates a competitive dynamic, making them improve each other. If the generator has learned to deceive the discriminator, the discriminator is presented with harder samples making it learn to differentiate them better, which in turn encourages the generator to create even more realistic samples. This competitive relationship between generator and discriminator constitutes the *adversarial* part of this approach.

2.7.1 Architecture

In their paper [57], Goodfellow *et al.* investigated the task of training the generator $\mathrm{G}(\cdot)$ to create images $\boldsymbol{X}_{\text{fake}}$, which resemble the data distribution of real images $\boldsymbol{X}_{\text{real}}$ in a training dataset $\mathcal{X}_{\text{train}}$, from a noise vector $\boldsymbol{z}$. The discriminator $\mathrm{D}(\boldsymbol{X})$ yields a scalar denoting the probability of $\boldsymbol{X}$ being real $\mathrm{D}(\boldsymbol{X}) = P(\boldsymbol{X} \in \mathcal{X}_{\text{train}})$. This structure is denoted by the solid arrows in Figure 2.5.

While the first publication related to GANs [57] was mainly a – mostly theoretical – proof of concept confirmed by a few experiments, architectures have evolved substantially. In their first approach, Goodfellow *et al.* transformed the noise vector $\boldsymbol{z}$ into an image $\boldsymbol{X}_{\text{fake}}$ employing solely fully connected layers and thereby without leveraging the spatial dependencies within an image (see also Section 2.5). Mirza and Osindero [152] proposed a conditional GAN, which creates the image $\boldsymbol{X}_{\text{fake}} = \mathrm{G}(\boldsymbol{z} \mid \boldsymbol{y})$ based on an additional condition given by a vector $\boldsymbol{y}$. Their work added the condition to the discriminator

$\mathrm{D}(\boldsymbol{X} \mid \boldsymbol{y})$; however, a separate classifier, as depicted by the dashed lines in Figure 2.5, is a valid alternative to better separate both objectives: generating realistic data that also fulfills the condition $\boldsymbol{y}$. By substituting the fully connected layers in the generator with four transposed convolutions (see Section 2.5.2) to transform $\boldsymbol{z}$ into an image of resolution 64×64, Radford *et al.* [173] managed to better leverage the spatial properties of images. With StyleGAN [104] and StyleGAN2 [105], Karras *et al.* employed the technique of progressively increasing the resolution of a GAN from ProGAN [103]. They combined it with adaptive instance normalization (AdaIN), which inserts the image's style into the network by scaling the normalized feature maps with $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ computed from the style vector. Together with the noise, both are introduced into the network at different depths resulting in highly realistic but still random images, which are also controllable by the style vector.

In contrast to tasks involving the generation of data from noise, image manipulation tasks, *e.g.*, face attribute manipulation, face swapping, image coloring, style transfer, *etc.*, perform a transformation on an input image $\boldsymbol{X} \mapsto \boldsymbol{X}_{\text{fake}}$ and do typically not include any noise vector $\boldsymbol{z}$. Thus, the requirement of generative models to also match the training data distribution in terms of variety is lifted. While some approaches [170, 251] opted for encoder-decoder structure, which downsampled the feature maps to a lower resolution and upsampled again, Isola *et al.* [97] employed skip connections to allow low-level information to propagate directly towards the last layers. For tasks where the output has the same underlying structure as the input, *e.g.*, edges or corners are at similar positions, this so-called *U-Net*[179] has become the default architecture for image manipulation tasks. Besides, Zhu *et al.* [285] proposed to use cycle-consistent losses by training an additional generator $\widetilde{\mathrm{G}}(\boldsymbol{X})$ to perform the inverse task $\boldsymbol{X}_{\text{fake}} \mapsto \boldsymbol{X}$. By ensuring that $\widetilde{\mathrm{G}}(\mathrm{G}(\boldsymbol{X})) \approx \boldsymbol{X}$, the generator can learn a realistic transformation even for frequent under-constrained scenarios in which pair-wise data is unavailable. This concept was further extended by StarGAN [23, 24] to multiple domains or by DiscGAN [107] by employing separate cycle-consistency losses and separate discriminators for each generator. Another major step in improving the structure of the generator (and the discriminator) was proposed by Zhang *et al.* [264] by incorporating the self-attention mechanism, which enables the generator to grasp long-distance relationships between pixels.

Even though improving the architecture of the generator seems to be the best approach to generate more realistic data, the discriminator $\mathrm{D}(\boldsymbol{X})$ also has a significant influence on the outcome. Overall, there are various fundamentally different discriminator architectures to consider. Traditionally, the image $\boldsymbol{X}$ is propagated through multiple convolutional layers with stride $S = 2$ and concluded by a fully connected layer with sigmoid$(\cdot)$ activation to output the probability of $\boldsymbol{X}$ being real $\mathrm{D}(\boldsymbol{X}) \in [0, 1]$ [103, 104, 173]. Isola *et al.* [97] assumed that low-level realism is already captured by additional L_1 or L_2 losses and proposed restricting the receptive field of the convolutions such that it does not cover the entire image $\boldsymbol{X}$ and employing global average pooling to obtain a single output probability. In this way, local patches are evaluated separately, and thus the focus is shifted towards evaluating whether every patch also contains high-frequent

information, *i.e.*, crisp details. To reduce the overfitting onto a single discriminator, multiple discriminators [44, 154, 219] can be employed, which can fulfill a different purpose if provided with a different input, *e.g.*, multiple scales of $\boldsymbol{X}$ as in Wang *et al.* [219]. Besides, other discriminator variants also include intermediate layers from the generator in judging realism [102] or output the probability of $\boldsymbol{X}$ being real for every class separately if images are translated between multiple classes [129].

2.7.2 Training

The discriminator $\mathrm{D}(\cdot)$ is trained in order to maximize the probability of correctly classifying $\boldsymbol{X}$, which can be formulated as minimizing the negative log-likelihood

$$\min_{\mathrm{D}} \left(-y \log(\mathrm{D}(\boldsymbol{X}_{\text{real}}) - (1-y) \log(1 - \mathrm{D}(\boldsymbol{X}_{\text{fake}})\right) \tag{2.19}$$

assuming that the label y is 1 for $\boldsymbol{X}_{\text{real}}$ and 0 for $\boldsymbol{X}_{\text{fake}}$.[x] The generator $\mathrm{G}(\cdot)$ tries to deceive the discriminator $\mathrm{D}(\cdot)$ with $\boldsymbol{X}_{\text{fake}}$, *i.e.*, it maximizes Equation (2.19) for $y = 0$

$$\max_{\mathrm{G}} \left(-\log(1 - \mathrm{D}(\mathrm{G}(\boldsymbol{z}))\right). \tag{2.20}$$

To address vanishing gradient problems, Equation (2.20) is expressed as a minimization problem

$$\min_{\mathrm{G}} \left(-\log(\mathrm{D}(\mathrm{G}(\boldsymbol{z}))\right), \tag{2.21}$$

from which a training loss can be deduced.

Overall, the adversarial min-max game between generator $\mathrm{G}(\cdot)$ and discriminator $\mathrm{D}(\cdot)$ is described by the following comprehensive objective function:

$$\min_{\mathrm{D}} \max_{\mathrm{G}} \left(-y \log(\mathrm{D}(\boldsymbol{X}_{\text{real}}) - (1-y) \log(1 - \mathrm{D}(\mathrm{G}(\boldsymbol{z}))\right). \tag{2.22}$$

Generator $\mathrm{G}(\cdot)$ and discriminator $\mathrm{D}(\cdot)$ are typically optimized in an alternating manner, *i.e.*, the weights of $\mathrm{D}(\cdot)$ are fixed when optimizing $\mathrm{G}(\cdot)$ and vice versa. However, since their objectives are contrary, their respective losses do not behave similarly to the training of a single ANN. Every weight update of $\mathrm{G}(\cdot)$ makes $\boldsymbol{X}_{\text{fake}}$ being perceived as more realistic by $\mathrm{D}(\cdot)$, which increases the difficulty of distinguishing $\boldsymbol{X}_{\text{fake}}$ from $\boldsymbol{X}_{\text{real}}$, setting back the loss of $\mathrm{D}(\cdot)$ and vice versa. Thus, GAN training can be interpreted as training the generator $\mathrm{G}(\cdot)$ with the discriminator $\mathrm{D}(\cdot)$ as a loss function, which also learns to better judge realism while training progresses.

Despite this apparent standstill in terms of losses, the tasks of both networks adapt, *i.e.*, the weight updates in $\mathrm{G}(\cdot)$ lead to a more realistic $\boldsymbol{X}_{\text{fake}}$, which is now necessary to deceive $\mathrm{D}(\cdot)$. If the loss of either $\mathrm{G}(\cdot)$ or $\mathrm{D}(\cdot)$ approaches zero, the GAN training fails as $\mathrm{G}(\cdot)$ found an easy way of deceiving $\mathrm{D}(\cdot)$ or $\mathrm{D}(\cdot)$ always correctly identifies $\boldsymbol{X}_{\text{fake}}$, *e.g.*, due to artifacts, overrepresentation of a single color, *etc.* This behavior is also

[x] This binary version of the cross-entropy (CE) loss is discussed more in-depth in Section 3.4 on the example of face recognition (FR).

referred to as (partial) *mode collapse*, which makes the generator converge towards always generating a very similar face despite depending on random noise $\boldsymbol{z}$. In such scenarios, the training is stuck in local minima and both networks cannot benefit from one another anymore. Hence, it is crucial to balance weight updates of $\mathrm{G}(\cdot)$ and $\mathrm{D}(\cdot)$ to maintain a steady loss, even though it seems counterintuitive. To mitigate mode collapse, Metz *et al.* [150] proposed to unroll several optimization steps of the discriminator. In this way, $\mathrm{G}(\cdot)$ is less likely to overfit to local minima, which could be exploited by $\mathrm{D}(\cdot)$, resulting in a stabilization of the training. Apart from the meticulous hyperparameter selection to maintain steady losses throughout the training, the inability of losses to measure training success contributes to the overall cumbersome GAN training.

Goodfellow *et al.* [57] showed that at convergence, which is also referred to as the *Nash equilibrium*, $\mathrm{G}(\boldsymbol{z})$ is indistinguishable from $\mathcal{X}$, causing the discriminator to always guess with $\mathrm{D}(\boldsymbol{X}_{\text{real}}) = \mathrm{D}(\boldsymbol{X}_{\text{fake}}) = 0.5$. Moreover, they illustrate that optimizing the generator corresponds to minimizing the Jensen-Shannon divergence between $\mathrm{G}(\boldsymbol{z})$ and $\mathcal{X}$.[xi] However, according to Farnia and Ozdaglar [46], not all GANs have Nash equilibria. Since the initial GAN, multiple new loss functions, such as least-squares GAN [144], Wasserstein GAN [6] with gradient penalty [61], or boundary equilibrium GAN [11], have emerged with their benefits being questioned by Lucic *et al.* [137], in which a large-scale study showed that all investigated loss functions can yield comparable results given sufficient hyperparameter optimization. Thus, the vanilla GAN by Goodfellow *et al.* [57] remains viable.

[xi] In contrast to the Kullback-Leibler divergence, the Jensen-Shannon divergence symmetrically measures the dissimilarity between two probability distributions.

3

General Face Recognition

Any face recognition (FR) system can be divided into *subject-dependent* and *subject-independent* FR as depicted in Figure 3.1. Since subject-dependent FR systems are trained and evaluated on the same limited group of identities, their generalization to unknown identities – while theoretically possible – is rather limited. Hence, they are inflexible as they require costly retraining if the pool of identities is extended. Nowadays, the focus lies entirely on subject-independent architectures as well-established powerful deep learning architectures are capable of learning to generalize intra-subject variations. While subject-dependent architectures are considered *classifiers*, which directly predict the identity y of a face $\boldsymbol{I}$,[i] subject-independent approaches are *feature extractors* as they map $\boldsymbol{I}$ into a discriminative deep feature space $\boldsymbol{f}$, in which the distance between two features corresponds to the faces' dissimilarity. Thus, subject-independent architectures can also cover the subject-dependent evaluation scenarios as long as at least one reference image is provided. This work follows recent advances in FR and exclusively considers subject-independent approaches.

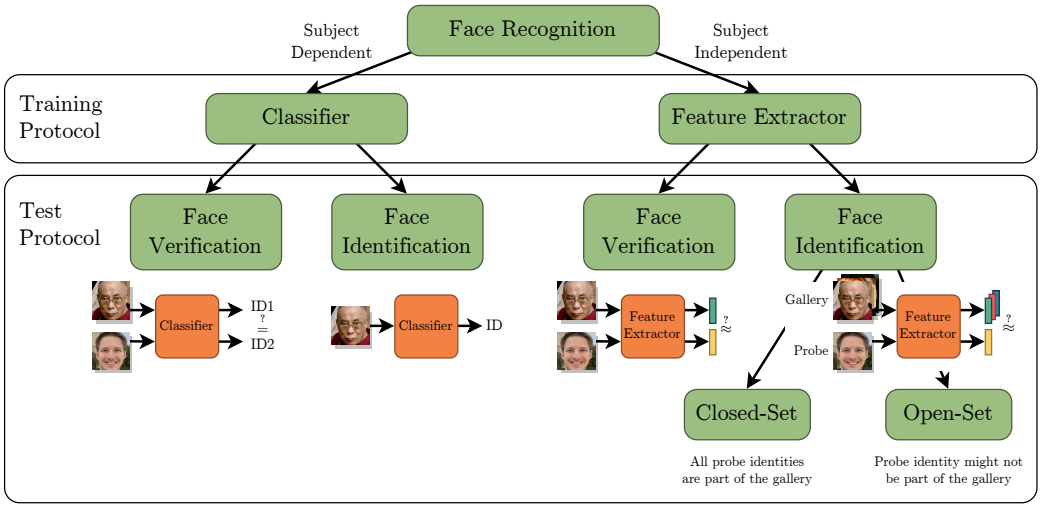

Figure 3.1: Different approaches to face recognition (FR).

[i]with "face" referring to "image of a face"

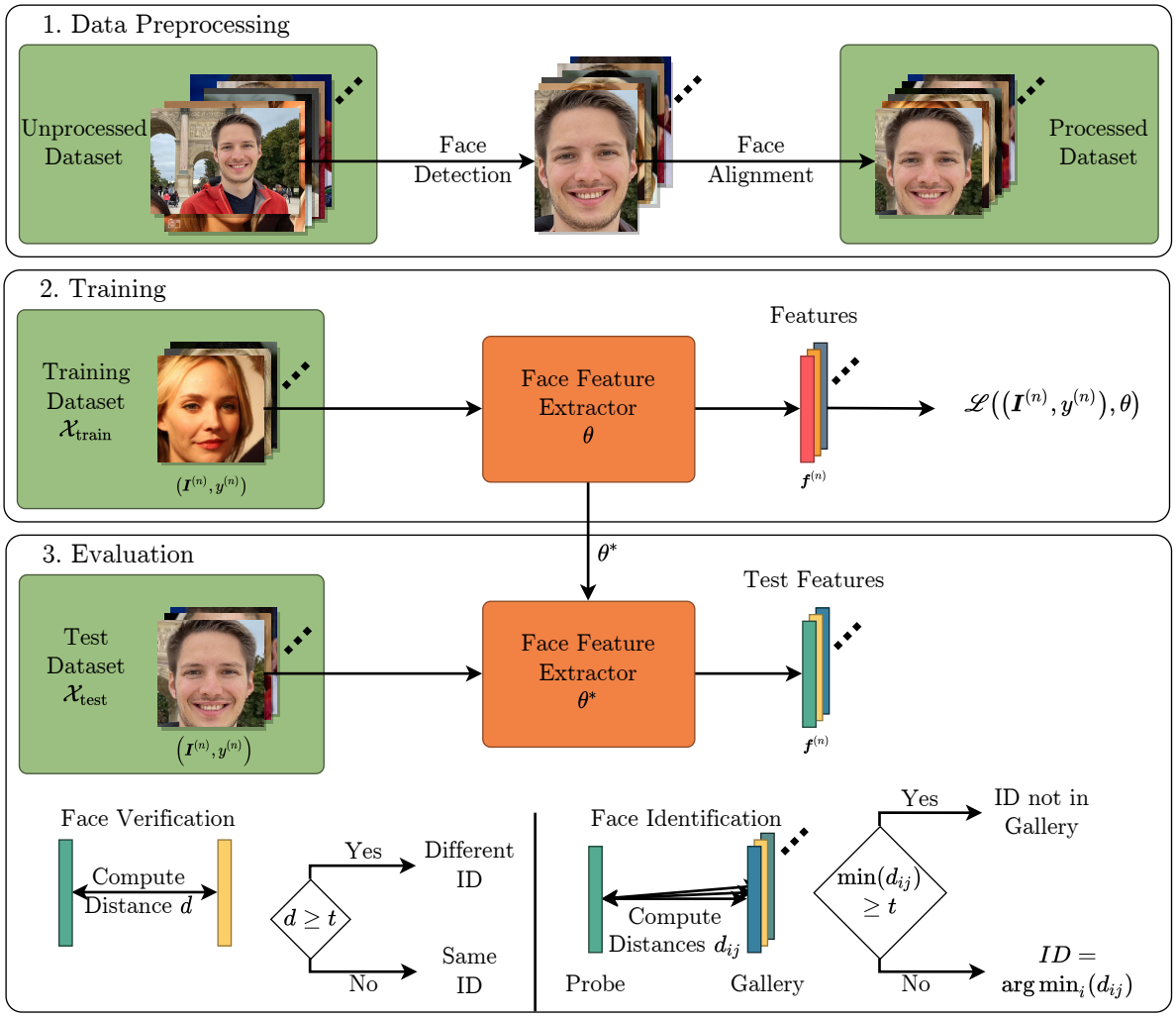

Figure 3.2: Training and evaluation pipeline of a subject-independent FR system.

Regardless of subject dependency, both architectures can be evaluated in terms of face verification or face identification. Face verification (one-to-one comparison) investigates whether a pair comprising two faces is either *genuine*, *i.e.*, both faces' identity is identical, or *imposter*, *i.e.*, both faces belong to the distinct identities. Face identification determines the identity of a probe face by comparing it with a gallery of faces (one-to-many comparison). The latter is further split into *closed-set* and *open-set* dependent on whether the identity of the probe face is represented within the gallery or not. All three evaluation protocols are closely related, yet they provide different metrics corresponding to everyday real-world tasks, *e.g.*, unlocking the smartphone and automatic passport control (face verification), or law enforcement and large-scale image retrieval (face identification).

Building and evaluating an FR system usually involves three steps: 1) data preprocessing; 2) network training; and 3) performance evaluation. Figure 3.2 illustrates this process for a subject-independent system.

First, all unprocessed face datasets (for training and evaluation) are preprocessed, which involves detecting the face and spatially normalizing it by *face alignment*. Then, the resulting preprocessed training dataset is utilized to optimize the weights of a face feature extractor by minimizing a loss function. In order to evaluate the performance of the FR system, features are extracted from the faces within the test dataset. Ultimately, the test features are used to evaluate the performance of the FR system in terms of face verification or face identification.

The first part of this chapter introduces the reader to different components of a conventional FR system, as indicated by Figure 3.2. Section 3.1 provides an overview of widely-used datasets for training and testing, and emphasizes the criteria for a suitable dataset selection. In Section 3.2, the data preprocessing steps, including face detection and alignment, are explained. Popular architectures and loss functions for FR are presented in Section 3.3 and Section 3.4.

After providing a general and exhaustive overview of every component necessary to perform FR in Sections 3.1 to 3.4, the second part of this chapter (Section 3.5) presents training strategies and defines metrics to evaluate the models in terms of FR. Then, various FR models are trained and their parameters are compared in an ablation study in Section 3.6.1. In addition, Section 3.6.2 provides an in-depth analysis of three models used throughout this work and Section 3.6.3 compares the three models with the state of the art.

3.1 Datasets

3.1.1 Training Datasets

As introduced in Section 2.4, an artificial neural network (ANN) is trained for a specific task using a dataset $\mathcal{X}_{\text{train}} = \left\{ \left(\boldsymbol{I}^{(n)}, y^{(n)} \right) \right\}_{n=1}^{N_{\text{train}}}$ comprising N_{train} tuples of face images $\boldsymbol{I}$ with their corresponding identities as a label y. To ensure that any FR model is capable of generalizing well to unknown identities, it is decisive to train it on a sufficiently large dataset (see Section 2.6). Large in this context can signify two different things: 1) a large number of identities with a relatively low number of images per identity, which is referred to as *wide*; or 2) a limited number of identities with a large number of images per identity, which is denoted as *deep*.

As depicted in Table 3.1, the average number of images per identity of popular datasets ranges from 20 to 1000, allowing the selection of relatively wide or deep datasets. Intuitively, it is not evident whether training with a wider or a deeper dataset leads to better FR performance. Wide datasets comprise many identities and thereby cover more inter-identity variations, which allows the model to learn even subtle differences between similar-looking identities. On the other hand, a deep dataset encompasses a high amount of intra-identity variations.

In her analysis, Fees [7$^+$] showed that face identity representations highly depend on face attributes such as *grey hair*, *bald*, and *eyeglasses*, requiring the dataset to contain many examples of both cases such that the trained model becomes invariant to them.

Moreover, Sun [32+] evaluated how the variety of head poses affects face verification accuracy. She found that when only utilizing half of the images for every identity, equally distributing the head poses among the range of head poses clearly outperforms using only frontal faces or faces with extreme head poses. This effect was observed when training with 1000 to 6000 number of identities on the VGGFace2 dataset and evaluating on the Labeled Faces in the Wild (LFW) [86] dataset (see Section 3.1.2). Hence, it is crucial to ensure that a dataset also exhibits a vast variety of head poses to obtain the best possible performance given a limited number of images. In addition, different illuminations, occlusions, and expressions of one identity increase the variety of the data and allow the model to conclude what truly represents an identity.

To evaluate whether deeper or wider datasets are superior, Bansal *et al.* [9] divided several datasets into deep and wide halves. They found that it is better to use deeper datasets for deeper networks, while wider datasets are preferred for shallower networks. In addition, the analysis from Zhou *et al.* [283] shows that adding more identities to a given dataset boosts performance, but only if the added identities contain a certain number of images. While this rule of thumb holds for the investigated cases, additional factors likely influence the overall quality of a dataset.

Zhang *et al.* [270] showed that a uniformly distributed dataset outperforms so-called *long-tailed* datasets, comprising many identities with few samples and few identities with many samples. Hence, it is vital that a dataset contains at least a few images for every identity and it is uniformly distributed in terms of the number of images per identity. With Range Loss, a specific loss to cope with long-tailed datasets was proposed [268].

Due to the large number of images in most datasets, manual annotations are too time-consuming. Moreover, the large number of identities and the thereby larger number of similarly looking identities together with poor quality (*e.g.*, low resolution, extreme head poses, *etc.*) causes labeling errors from manual annotators – especially if they are unfamiliar with the identity to be labeled. But also automatically generated datasets are noisy if they are not supervised by any (manual) cleaning. Wang *et al.* [211] showed that large datasets are especially susceptible to label noise. They estimated that the MS-Celeb1-1M dataset contains $\approx 50\%$ label noise rendering the training particularly cumbersome. Moreover, Bansal *et al.* [9] further demonstrated that adding noise to a dataset worsens the performance, which was reproduced by Wang *et al.* [211]. While the authors MS-Celeb-1M are aware of the high level of noise present in their dataset and see it as a challenge to develop noise-robust training strategies, the analysis from Wang *et al.* [221] suggests that the noise level in the VGGFace2 dataset is comparably low.

To cope with noisy datasets, some approaches [17, 83, 189, 235] directly addressed the uncertainty in face datasets, whereas other researchers [98, 221] showed that cleaning training datasets (especially MS-Celeb-1M [65]) yields superior results. Deng *et al.* [33] even employed annotators familiar with the ethnicity to improve the label quality of hard samples in MS-Celeb-1M, which resulted in the MS1MV2 dataset. Other approaches utilized different cleaning techniques resulting in a different number of identities and images, *e.g.*, 59k and 3.7M [175], 73k and 3.3M [223], 82k and 4M [36], 82k and 4.5M [34, 35], 100k and 5M [268], respectively. Still MS1MV2 by Deng *et al.* [33], remains the most popular training dataset for FR [87, 108, 109, 148, 199, 281].

Table 3.1: Popular datasets used for training FR models. † denotes that only the training subset is considered.

				# Images/Videos per Identity		
Dataset	# Identities	# Images	# Videos	Min	Avg	Max
CelebFaces+ [201]	10 177	202 599		1	20	35
CASIA-WebFace [248]	10 575	494 414		2	47	804
UMDFaces [10]	8 277	367 888			44	
VGGFace [169]	2 622	2 622 000		1 000	1 000	1 000
VGGFace2† [14]	8 631	3 138 924		87	364	843
MS-Celeb-1M [65]	100 000	10 000 000			100	
MS1MV2 [33]	85 742	5 822 653		2	68	602
Asian-Celeb [30]	93 979	2 830 146			30	
Glint360k [3]	360 232	17 091 657			47	
UMDFaces-Videos [9]	3 107		22 075		7	
VoxCeleb2 [26]	6 112		150 480		25	

Besides noisy data, it is crucial to minimize bias and ensure that the distribution of the training dataset resembles the distribution of the test dataset, *e.g.*, when testing the recognition of children, the training dataset must contain juvenile faces (see also Section 2.6). While gender is distributed relatively equally in VGGFace2 (59.3% male versus 40.7% female [91]),[ii] the ethnicity distribution is substantially biased towards Caucasians (74.2%) with Asians (6.0%) Indians (4.0%) and Africans (15.8%) being underrepresented [218]. For MS-Celeb-1M, this inequality is even more pronounced. To compensate the resulting inferior performance when evaluating Asian faces, extending the training dataset to encompass the Asian-Celeb dataset [30] constitutes a viable option as performed by Deng *et al.* [33].

It can be concluded that a training dataset should ideally be: 1) uniformly distributed; 2) contain a low amount of noise and bias; and 3) deep, as the models used in this work are considered deep when compared to [9]. This leaves two choices for the training of FR models, namely the VGGFace2 dataset – a deep dataset with particularly low noise and a focus on high intra-identity variations in head pose and age – and the MS1MV2 dataset, which comprises a very high number of identities and was cleaned to reduce the amount of noise. CASIA-WebFace [248] is also quite popular in FR [87, 130, 131, 148, 189, 215, 268, 281], but its single benefit compared to VGGFace2 lies in a time-efficient training with less images. Thus, VGGFace2 constitutes a viable and powerful training dataset despite being less popular [14, 33, 101, 239, 240, 273]. Other large datasets are either very specific for one ethnicity (Asian-Celeb [30]) or are difficult to train as they require additional hardware due to the huge number of identities (Glint360k [3]). In terms of video FR datasets, VoxCeleb2 [26] is considered superior to UMDFaces-Videos [9] due to its larger number of identities and videos.

[ii] assuming binary genders

3.1.2 Benchmark Datasets

In contrast to datasets used for training, benchmark datasets $\mathcal{X}_{\text{test}}$ are typically wide as they try to mimic the general population.[iii] In addition, a wide dataset is crucial in order to generate pairs of similar-looking identities and to obtain a huge number of imposter pairs, which is necessary to evaluate security-sensitive applications. Since only subject-independent approaches are considered, the identities in benchmark datasets do not have to be part of the training dataset. In fact, there should be no identity overlap between both datasets as otherwise, the performance would not correspond to the actual performance when exposed to unknown identities. However, minor overlaps exist between various training and benchmark datasets. Ideally, overlapping identities should be removed during training. Since this would inevitably increase the clutter of the training datasets, most approaches ignore the identity overlap. In addition, the overlap is considered minor and since every approach is affected similarly, the comparison of various approaches is still meaningful.

Typically, every benchmark dataset was created to investigate a particular task. Table 3.2 sums up the most important properties of popular datasets used to evaluate FR performance. The LFW dataset [86], initially released in 2007, was the first dataset to comprise images taken in uncontrolled (in the wild) environments and was long the gold standard for measuring FR performance. Nowadays, the accuracy is saturated as achieving a face verification accuracy above 99.3% is considered relatively easy. One reason contributing to this is that imposter pairs often have different gender and ethnicity, whereas both are the same for genuine pairs. Moreover, the age gap between two faces in genuine pairs is usually substantially lower than between two faces in imposter pairs.

Therefore, new datasets, such as Cross-Pose LFW (CPLFW) [275] and Celebrities in Frontal-Profile (CFP) [188],[iv] emerged evaluating face verification performance under varying head poses. Another direction was pursued by the Cross-Age LFW (CALFW) [276] dataset and AgeDB [156],[v] which focus on ensuring a similar age gap among genuine and imposter pairs. While all datasets mentioned above fulfill their purpose and are still widely employed, their relatively low number of pairs restricts the performance analysis. In particular, more practical use cases cannot be evaluated, including the performance when only allowing a very low (1 in 10^6) number of falsely classified imposter pairs.

To overcome these limitations, the MegaFace dataset [106] provides $\approx 4 \cdot 10^9$ face pairs enabling a more comprehensive face verification analysis. In addition, MegaFace allows the evaluation of closed-set face identification performance, where all probe identities form part of the gallery. Due to its unique protocols with up to $N_{\text{d}} = 10^6$ images of distractors, the influence of an extensive gallery can be investigated exhaustively. Following the latest analyses of FR algorithms, this work utilizes the refined MegaFace benchmark provided by Deng *et al.* [33] to minimize noise if not stated otherwise.

[iii] The terms "test dataset" (*cf.* Section 2.6) and "benchmark dataset" are used as synonyms.

[iv] In this work, only the frontal-profile protocol is utilized as the frontal-frontal protocol does not yield any additional insights compared to LFW.

[v] In this work, only the AgeDB-30 protocol, *i.e.*, an age gap of 30 for genuine and imposter pairs, is employed.

Table 3.2: Popular datasets used to evaluate FR models. † denotes that FaceScrub [163] is used as a gallery and ◇ indicates the mixed media protocol.

Dataset	# Identities	# Images	# Videos	# Pairs	Gallery	Probe	Description
LFW [86]	4 281	7 701		6 000			Saturated
CPLFW [275]	2 296	5 984		6 000			Head pose
CFP [188]	500	5 901		7 000			Head pose
CALFW [276]	2 996	7 156		6 000			Age
AgeDB [156]	$\leq$ 568	$\leq$ 12 000		6 000			Age
MegaFace† [106]	690 572	1 027 060		$4 \cdot 10^9$	1 000 001	3 530	Distractors
YTF [232]	1 447	3 226		5 000			Frames
IJB-A [111]	500	5 712	2 085	11 748	113	1 764	Image & Frames
IJB-B◇ [231]	1 845	21 798	7 011	8 010 270	931 & 914	10 270	Images & Frames
IJB-C◇ [146]	3 531	31 334	11 779	15 658 489	1 772 & 1 759	19 593	Images & Frames

For video FR, the YouTubeFaces (YTF) dataset [232] represents a popular choice as it contains on average 181.3 frames per video. The IARPA Janus Benchmarks (IJBs) [111, 146, 231] contain protocols for face sets comprising still face images together with video frames. Thus, they evaluate information fusion for an arbitrary number of images within a set. Moreover, in IJB-B and IJB-C, the gallery is split into two disjoint galleries. This allows the investigation of open-set face identification performance, *i.e.*, the probe identity is not always part of the gallery, which is particularly relevant for security access applications.

Besides the datasets mentioned earlier, benchmark datasets focusing on similarly looking faces [37], adversarial attacks [280], racial bias [218], cross-quality [16†], or comprising trillion pairs [30] are occasionally employed. However, the analysis of FR performance on the latter datasets is out of scope for this work.

3.2 Data Preprocessing

Typically, the images $\boldsymbol{I}_{\text{org}}$ provided by FR datasets $\mathcal{X}$ were taken in unconstrained environments and are rather loosely cropped, *i.e.*, they contain not only information about the person but also background or even other persons (see Figure 3.2). Hence, it is necessary to either develop an algorithm capable of identifying multiple persons within an image or employ face detection prior to the FR. Joint face detection and recognition is feasible [22], yet its performance is far from optimal. Thus, most FR algorithms incorporate face detection to mitigate the influence of background noise and thereby allow the FR network to focus on the recognition task.

After face detection, faces are usually aligned, *i.e.*, spatially normalized, to further speed up the training and improve performance. This is possible since all human faces – and even animal faces to some extent – are always composed in the same way with roughly the same proportions, and every face part (*e.g.*, eyes, mouth, nose) is located at a specific position in relation to one another. By enforcing a fixed image resolution, it is not required to employ specific algorithms capable of handling arbitrary input resolutions. Besides, sizes and distances within the faces remain relatively constant for all faces. Therefore, the network does not have to learn multiple filter combinations with

distinct receptive fields to extract identity features at different scales. This lowers the redundancy within the network, which ultimately allows the network to focus on more decisive features.

Cropping the face to the bounding boxes obtained by face detection and resizing the face to a fixed resolution constitutes a straightforward method to achieve spatial normalization. In this way, all faces have roughly the same size. However, due to the resizing, faces are not scaled uniformly, causing unwanted deformations, which ultimately results in worse performance [9]. To obtain a more accurate spatial normalization to a canonical position without deforming the face, a similarity transformation (rotation φ, uniform scaling ζ,[vi] and translation τ_x and τ_y) of N_LM facial landmarks is incorporated.

Thus, the objective of the face alignment is to map N_LM facial landmarks of the original image $\begin{pmatrix} x_{i,\mathrm{src}} & y_{i,\mathrm{src}} \end{pmatrix}^\mathsf{T}$ onto predefined dataset-wide target landmark positions $\begin{pmatrix} x_{i,\mathrm{tar}} & y_{i,\mathrm{tar}} \end{pmatrix}^\mathsf{T}$

$$\begin{pmatrix} x_{i,\mathrm{src}} \\ y_{i,\mathrm{src}} \end{pmatrix} \mapsto \begin{pmatrix} x_{i,\mathrm{tar}} \\ y_{i,\mathrm{tar}} \end{pmatrix}, \qquad i = 1, \ldots, N_\mathrm{LM}. \tag{3.1}$$

After face alignment, the recognition model knows where information about a specific face part can be extracted without first localizing this specific face part within the input image space. Aligning a face involves multiple steps, which are illustrated in Figure 3.3, and listed as follows:

1. Extract facial landmark positions of the source image.
2. Compute the transformation parameters to map the source onto the target positions.
3. Transform the image using the transformation parameters.
4. Interpolate and crop the aligned image.

Facial landmark detection is the localization of fiducial keypoints within the face. Even though face alignment is an essential preprocessing step in any state-of-the-art FR algorithm, the accuracy of facial landmarks plays a less important role compared to the remaining FR system. The main reason for this is that optimal alignment is impeded when some face parts do not exist within the image due to occlusions or extreme head poses. Moreover, due to different face proportions (narrow versus wide faces), the facial landmarks of the aligned face only lie as close as possible (in a least-squares sense) to the target landmarks. Hence, imperfect alignment is unavoidable, requiring the FR model to become robust to slight variances.

The number of detected facial landmarks N_LM varies depending on the dataset used for training the facial landmark detector. While older facial landmark detectors have focused primarily on 4 or 5 landmarks [265], nowadays, 68 landmarks are mainly investigated [16, 63, 149], and the HELEN dataset [116] allows the training of models predicting even 192 landmarks. Despite the analysis from Guo *et al.* [63], in which they demonstrated that FR performance is superior when aligning faces with 68 3D facial landmarks instead of utilizing the multi-task CNN (MTCNN) (five 2D landmarks) from Zhang *et al.* [265], the MTCNN is still widely used [33, 87, 108, 109] as a facial landmark detector for FR.

[vi] $\zeta > 0$, thus no mirroring.

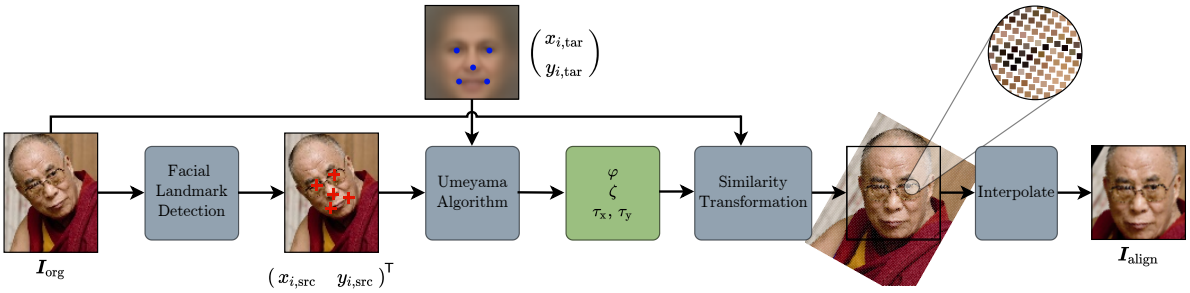

Figure 3.3: Face alignment pipeline with $N_\text{LM} = 5$ landmarks extracted with the MTCNN using the face alignment policy (FAP) from ArcFace [33].

Since only rotation φ, uniform scaling ζ, and translations τ_x and τ_y are allowed when transforming the facial landmarks, the mapping from Equation (3.1) can be rewritten

$$\begin{pmatrix} x_{i,\text{tar}} \\ y_{i,\text{tar}} \end{pmatrix} = \underbrace{\begin{pmatrix} \zeta\cos\varphi & -\zeta\sin\varphi & \tau_\text{x} \\ \zeta\sin\varphi & \zeta\cos\varphi & \tau_\text{y} \end{pmatrix}}_{\boldsymbol{T}} \cdot \begin{pmatrix} x_{i,\text{src}} \\ y_{i,\text{src}} \\ 1 \end{pmatrix}, \qquad i = 1, \ldots, N_\text{LM}, \tag{3.2}$$

where $\boldsymbol{T}$ denotes the transformation matrix.

For $N_\text{LM} > 2$, the system of equations induced by Equation (3.2) is overdetermined, and therefore, except for unrealistic cases of linear dependencies, does not have a solution. Hence, the objective is to find a solution that minimizes the mean squared error (MSE) between source and target landmarks

$$\varphi^*, \zeta^*, \tau_\text{x}^*, \tau_\text{y}^* = \underset{\varphi,\zeta,\tau_\text{x},\tau_\text{y}}{\arg\min} \left(\frac{1}{N_\text{LM}} \sum_{i=1}^{N_\text{LM}} \left\| \begin{pmatrix} x_{i,\text{tar}} \\ y_{i,\text{tar}} \end{pmatrix} - \boldsymbol{T} \cdot \begin{pmatrix} x_{i,\text{src}} \\ y_{i,\text{src}} \\ 1 \end{pmatrix} \right\|^2 \right). \tag{3.3}$$

In 1991, Umeyama derived the least-squares solution for Equation (3.3) [209]. The interested reader is referred to Appendix B for the solution for two-dimensional facial landmarks.

In order to align the face, new coordinates $\tilde{x}$ and $\tilde{y}$ of the original pixel values $[\boldsymbol{I}_\text{org}]_{y,x,:}$ in the aligned image $\boldsymbol{I}_\text{align}$ must be computed. This is achieved by applying Equation (3.2) with the solution of Equation (3.3) to every position of the original image $\boldsymbol{I}_\text{org}$ with size $W \times H \times C$

$$[\boldsymbol{I}_\text{align}]_{\tilde{y},\tilde{x},:} = [\boldsymbol{I}_\text{org}]_{y,x,:} \qquad \forall \begin{pmatrix} y & x \end{pmatrix} \in \{1, \ldots, H\} \times \{1, \ldots, W\} \tag{3.4}$$

with

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \zeta^*\cos\varphi^* & -\zeta^*\sin\varphi^* & \tau_\text{x}^* \\ \zeta^*\sin\varphi^* & \zeta^*\cos\varphi^* & \tau_\text{y}^* \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{3.5}$$

The new coordinates $\tilde{x}$ and $\tilde{y}$, computed by Equation (3.5), are real-valued and thus do not typically lie on the grid $\begin{pmatrix}\tilde{y} & \tilde{x}\end{pmatrix} \in \{1, \dots, \widetilde{H}\} \times \{1, \dots, \widetilde{W}\}$, where $\widetilde{H} \times \widetilde{W}$ is the desired spatial resolution after face alignment. To obtain the final pixel values $[\boldsymbol{I}_{\text{align}}]_{\tilde{y},\tilde{x},:}$ with $\tilde{x}$ and $\tilde{y}$ being integer, the pixel values are bilinearly interpolated $\forall \begin{pmatrix}\tilde{y} & \tilde{x}\end{pmatrix} \in \{1, \dots, \widetilde{H}\} \times \{1, \dots, \widetilde{W}\}$. If the transformed image does not contain information to fill all pixels in the target image space, the pixels' values are set to zero (see also Figure 3.3). To mitigate the influence of such hard borders within the image onto the activations within the neural network, these values are sometimes set to grey (127 for input value range $[0, 255]$). After the face alignment pipeline, all images $\boldsymbol{I}_{\text{align}}$ of a dataset have exactly the same resolution $\widetilde{H} \times \widetilde{W}$, with their facial landmarks lying roughly at the same position.

In addition to the model used to extract facial landmarks, the facial alignment depends on the image resolution and the landmarks' target positions $\begin{pmatrix}x_{i,\text{tar}} & y_{i,\text{tar}}\end{pmatrix}^{\mathsf{T}}$. With the emerging cleaned versions [33, 98, 221] of the MS-Celeb-1M dataset [65], aligning faces to 112×112 px has become the standard. This is mainly due to the authors of the MS-Celeb-1M dataset [65] revoking database access and the authors of MS1MV2 [33] only providing access to the dataset after face alignment to 112×112 px. Besides, using faces aligned to 160×160 px or 224×224 px is not uncommon in the domain of face feature aggregation [240, 245]. In this case, a training dataset with high-resolution images such as VGGFace2 [14] needs to be used. Consequently, the model can leverage identity features of higher frequency, which are only present in higher resolutions, enabling superior performance at the cost of higher memory requirements and longer training time.

Concerning the face alignment policy (FAP), *i.e.*, the landmarks' target positions, there is no mutual agreement among the research community. Earlier, researchers did not reveal their FAP, which is most likely since their work's attention was set on the FR algorithm and, as a result, the rare code releases did not include the face alignment. Until the analysis by Xu *et al.* [242] in 2021, the influence of different FAPs on the FR performance was hardly investigated. By utilizing their proposed FAP search, they found that the optimal FAP involves a looser crop and positive vertical shift (*i.e.*, including less forehead and more chin information) compared to the FAP used in ArcFace [33]. By changing the FAP, the face verification accuracy of ArcFace on CALFW and CPLFW improved by 0.42% and 0.97%, respectively, while the effect on LFW and AgeDB is rather negligible. Similar to the preference of 112×112 px as image resolution, the FAP by Deng *et al.* [33] is widely employed since the refined MS1MV2 dataset is only available as an aligned dataset following their FAP.

While the choice of FAP and facial landmark detection algorithms vary, there is mutual agreement on the necessity of facial alignment. Face alignment not only allows faster training but also dispenses with the need for non-uniform scaling to obtain a fixed resolution or a resolution-independent architecture. Moreover, not aligning faces results in a substantial drop in FR performance if the model is not specifically designed to cope with unaligned faces [9, 63]. Multiple approaches have been proposed to allow end-to-end FR. While the prediction of facial landmarks is required for standalone face alignment, Zhong *et al.* [279] and Wu *et al.* [234] directly predicted the transformation

parameters (rotation φ, uniform scaling ζ, and translations τ_x and τ_y) to align the face in a separate module within their networks. In contrast to the similarity transformation of face alignment, face frontalization normalizes the head pose by generating an entirely new image of the same identity. Thus, unlike previous face alignment methods, Zhou *et al.* [282] incorporated multiple local homography transformations to frontalize the face. Typically, generative adversarial networks (GANs) (see Section 2.7) are frequently employed for face frontalization. *E.g.*, Zhao *et al.* [272] directly normalized the person's head pose and Na *et al.* [157] completed the UV map for accurate face synthesis. Even though utilizing an end-to-end approach seems beneficial, the additional supervision of the face alignment (or frontalization) together with a more complex network architecture make the training unnecessarily cumbersome. Moreover, their additional computational cost compared to efficient facial landmark algorithms limits their application in the real world. This is also indicated by the underrepresentation of end-to-end FR approaches among the published FR papers despite the increased access to more powerful hardware in recent years.

This work follows most recent FR approaches [33, 87, 108, 109] and employs the MTCNN [265] as the facial landmark detector. An additional face detector is not required since the MTCNN handles inputs of arbitrary resolution without previous face detection due to its image pyramid structure. The MTCNN was able to predict landmarks for all test sets, which are used throughout this work. However, extreme head poses or low resolutions in the VGGFace2 training dataset impeded the facial landmark prediction for 2944 images (0.09%), whose influence on the training outcome is negligible. Typically, faces in all datasets are centered and loosely cropped. Thus, in case facial landmarks of multiple faces were detected, the face size together with the offset from the center of the image are considered to find the most prominent face within the image. Besides, two different FAPs are utilized depending on the training dataset: 1) the FAP proposed by Deng *et al.* [33] when training on the MS1MV2 dataset; and 2) a handcrafted FAP of the same tightness but shifted $\approx 15\%$ towards the bottom (*i.e.*, more similar to the optimal FAP according to [242]) when training on the VGGFace2 dataset.

3.3 Architectures

The architecture constitutes the component of an FR system, which transforms the input image $\boldsymbol{I}$ into a deep feature vector $\boldsymbol{f}$. Despite its central part in the system, recent advances in FR are majorly attributed to advances concerning the loss function and training strategy. Nevertheless, FR approaches directly benefit from developments in classical computer vision tasks, such as image classification.

Early deep learning models for FR incorporated variants of the *AlexNet* [112], which comprise 4-5 convolutional and one fully connected layer [198, 204]. In contrast to the large convolutions (up to 11×11) in [112, 204], Simonyan and Zisserman [191] opted for smaller 3×3 convolutions in their 16-layer deep *VGGNet*.[vii] Moreover, VGGNet is the

[vii] In contrast to the layers introduced in Chapter 2, the depth of a network only considers trainable layers, *i.e.*, dropout, activation function, batch normalization, and pooling are not included.

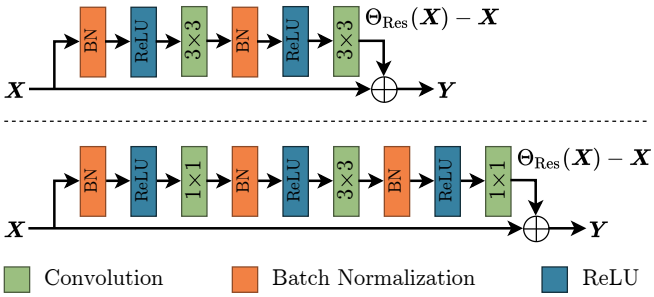

Figure 3.4: Residual units [72] for shallow (top) and deep (bottom) ResNets.

Table 3.3: Utilization of ResNets of different depths in recent FR approaches.

Depth	Methods
27	[36]
34	[108, 199]
50	[14, 33, 34, 87]
64	[130, 131, 189, 215]
101	[33, 87, 101, 108, 109, 148, 199]

first network that proved the effectiveness of doubling the number of feature maps when their spatial dimensions are halved. For FR, VGGNet was utilized in combination with a new dataset [169], supervisory signals [200], or a new loss function [268]. The 22-layer deep *GoogleNet* [203] employed inception modules, which comprise parallel 1×1, 3×3, 5×5 convolutions and a 3×3 maximum pooling, to obtain multi-resolution feature maps and was applied to FR by Schroff *et al.* [187].

The architectures mentioned above demonstrate that increasing the network depth results in a substantial performance boost. However, deeper networks are generally more cumbersome to train as they suffer from vanishing and exploding gradients and thus are prone to diverge. This limitation was lifted by the *residual network* (ResNet) [71, 72], which is still the most popular architecture in FR even five years after its publication (*cf.* Table 3.3). The key innovation constitutes the residual unit, as depicted in Figure 3.4, which allows the training of networks with even 1000 layers [72].

The detailed architecture of ResNets with varying depths is depicted in Table 3.4.[viii] Similar to GoogleNet [203], the dimension of the input image is reduced by a factor of 4 using a 7×7 convolution followed by maximum pooling – both with stride 2. Then, to obtain a network of depth L, the residual units (see Figure 3.4) are stacked from 8-times for $L = 18$ to 33-times for $L = 101$, where a group of residual units with the same output dimension forms a residual block (*cf.* Table 3.4). For shallower architectures ($L \leq 34$), the residual unit in Figure 3.4 (top) is used. This *bottleneck* structure, *i.e.*, the number of feature maps is reduced and restored afterwards, of the residual unit in Figure 3.4 (bottom) enables an economical implementation of deeper networks.

Instead of learning the desired mapping of a unit $\boldsymbol{Y} = \Theta_{\text{Res}}(\boldsymbol{X})$ directly, the residual unit only needs to learn the difference $\Theta_{\text{Res}}(\boldsymbol{X}) - \boldsymbol{X}$. This has multiple advantages: Low-level features extracted from early layers are forwarded to the output via the skip connections. In this way, the network can decide to skip particular residual units if the complexity of the input does not require a high amount of nonlinearity. This added flexibility further unburdens the selection of the network depth L specifically for a given task, as the network is capable of not incorporating all residual units in its prediction. Moreover, skip connections allow an unimpeded gradient flow, which stabilizes the

[viii] Note that the depth L of the ResNet is the number of trainable layers, which are used at inference. Since the last fully connected layer is dropped after training, it is not counted towards L.

Table 3.4: Architectures of ResNets for varying depths L with an input resolution of 112×112. Residual units are shown in brackets (*cf.* Figure 3.4), with the numbers of units stacked. † denotes that the first 3×3 convolution in the first unit operates with stride $S = 2$. Note that the depth L does not include the last layer since it is omitted after training. Adapted from [71].

		Depth L			
Layer Name	Output Size	$L = 18$	$L = 34$	$L = 50$	$L = 101$
conv1	56×56	7×7, 64, $S = 2$			
conv2_x	28×28	3×3 maximum pooling, $S = 2$			
		$\begin{bmatrix} 3\times3,\ 64 \\ 3\times3,\ 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3,\ 64 \\ 3\times3,\ 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1,\ 64 \\ 3\times3,\ 64 \\ 1\times1,\ 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1,\ 64 \\ 3\times3,\ 64 \\ 1\times1,\ 256 \end{bmatrix}\times3$
conv3_x	14×14	$\begin{bmatrix} 3\times3,\ 128^{\dagger} \\ 3\times3,\ 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3,\ 128^{\dagger} \\ 3\times3,\ 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128^{\dagger} \\ 1\times1,\ 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128^{\dagger} \\ 1\times1,\ 512 \end{bmatrix}\times4$
conv4_x	7×7	$\begin{bmatrix} 3\times3,\ 256^{\dagger} \\ 3\times3,\ 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3,\ 256^{\dagger} \\ 3\times3,\ 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256^{\dagger} \\ 1\times1,\ 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256^{\dagger} \\ 1\times1,\ 1024 \end{bmatrix}\times23$
conv5_x	4×4	$\begin{bmatrix} 3\times3,\ 512^{\dagger} \\ 3\times3,\ 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3,\ 512^{\dagger} \\ 3\times3,\ 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1,\ 512 \\ 3\times3,\ 512^{\dagger} \\ 1\times1,\ 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1,\ 512 \\ 3\times3,\ 512^{\dagger} \\ 1\times1,\ 2048 \end{bmatrix}\times3$
	1×1	GAP Dropout M_{f}-dimensional fc M_{cls}-dimensional fc, softmax			

training. Together with batch normalization (see Section 2.6.4), the employment of skip connections is a crucial component in training deep models.

In contrast to the initial publication [71], the convolutional layers in the residual units of ResNet-v2 [72] are preactivated, *i.e.*, the activation function is applied before the convolution, as in Figure 3.4. Besides, ResNet-v2 does not employ the activation function in the skip connection, which allows unrestricted propagation of information to deeper layers.

To conclude the stacked residual units, multiple FR approaches [33, 131] dispense with the global average pooling (GAP) in the original ResNet [71, 72]. Still, the GAP outputs the average activation in every feature map independent of its location and reduces the number of parameters and thus remains a viable option. Regardless of the usage of GAP, dropout is employed to improve the generalization of the subsequent layer. Specifically for FR, another fully connected layer – the so-called *bottleneck layer* – with $M^{[L-1]} = M_{\text{f}}$ neurons is introduced before the final fully connected layer with $M^{[L]} = M_{\text{cls}}$ neurons. Then every activation of the last layer $\Theta^{[L]}(\boldsymbol{I})$ is associated with an identity in $\mathcal{X}$. The output of the bottleneck layer constitutes the feature space $\boldsymbol{f} = \Theta^{[L-1]}(\boldsymbol{I})$, into which the faces $\boldsymbol{I}$ are embedded. By choosing $M_{\text{f}} \ll M_{\text{cls}}$, this bottleneck structure ensures that the relevant information to discern all M_{cls} identities must be encoded in the M_{f}-dimensional vector $\boldsymbol{f}$. This is crucial to allow subject-independent FR evaluation and improve the generalization to unknown identities. While *rectified linear unit* (ReLU) activation is employed throughout the network, it is crucial not to apply it to $\boldsymbol{f}$ as it would restrict

the features after the activation function $\boldsymbol{f}$ to the non-negative value space resulting in inferior performance. Thus, $\boldsymbol{f} = \tilde{\boldsymbol{f}}$. Due to its unique purpose when combined with the loss function, the activation function of the last layer $\Psi^{[L]}(\cdot)$ is discussed in Section 3.4.

Multiple extensions to the original ResNets have been proposed, which involve grouped convolutions [238] or so-called squeeze-and-excitation units to recalibrate the feature maps [82] in every residual unit. In addition, Wang *et al.* [213] incorporated attention models into the ResNet and Duta *et al.* [45] proposed the improved ResNet, which enhances the information flow and the skip connections for the resolution reduction, and adds grouped convolutions in the residual layer. Other architectures have shown the potential of adding more skip connections [84]. However, all extensions to the ResNet are hardly utilized by FR approaches. Lately, a promising approach was postulated by Zhong *et al.* with the vision transformer [281], which is likely to shape future FR research.

For the sake of comparability, this work follows the majority of related works and utilizes the ResNet architecture. To balance the increased training time and memory requirements of a deeper model with the undeniable inferior performance of shallower models, the ResNet-50 was selected as it offers the best trade-off.

3.4 Loss Functions

FR research is driven mainly by novel loss functions specifically designed to increase the discriminability of the feature space, *i.e.*, decrease intra-class distance and increase inter-class distances.

Early FR approaches [14, 169, 198, 204] opted for the *softmax cross-entropy* (CE) loss, which is popular in image classification. First, the softmax function is applied to the network's last layer $\tilde{\boldsymbol{y}} = \tilde{\Theta}^{[L]}(\boldsymbol{I})$ with M_{cls} neurons to obtain the final prediction

$$[\hat{\boldsymbol{y}}]_i = \left[\Psi^{[L]}(\tilde{\boldsymbol{y}})\right]_i = [\text{softmax}(\tilde{\boldsymbol{y}})]_i = \frac{\mathrm{e}^{[\tilde{\boldsymbol{y}}]_i}}{\sum_{m=1}^{M_{\text{cls}}} \mathrm{e}^{[\tilde{\boldsymbol{y}}]_m}}. \tag{3.6}$$

Unlike the activation functions introduced in Section 2.3, the softmax is not applied to every element of the vector of unnormalized class scores $\tilde{\boldsymbol{y}}$ separately. In this way, the softmax ensures that the output of the neural network $\hat{\boldsymbol{y}}$ fulfills both requirements of a probability distribution: $[\hat{\boldsymbol{y}}]_i \in [0, 1]\ \forall\, i$ and $\sum_{m=1}^{M_{\text{cls}}} [\hat{\boldsymbol{y}}]_m = 1$. Thus, the activation of the ith neuron $[\hat{\boldsymbol{y}}]_i$ represents the probability $P\left(i = y \mid \boldsymbol{I}\right)$ of i being the index of the ground-truth identity y of the face $\boldsymbol{I}$. According to maximum likelihood estimation, this probability is maximized by minimizing the negative log-likelihood, which results in the CE loss

$$\mathscr{L}_{\text{CE}} = -\sum_{m=1}^{M_{\text{cls}}} [\boldsymbol{y}]_m \log\left([\hat{\boldsymbol{y}}]_m\right) = -\log\left([\hat{\boldsymbol{y}}]_y\right), \tag{3.7}$$

with $\boldsymbol{y}$ denoting the one-hot encoded vector of the index of the ground-truth identity

y.[ix] Since the CE can also be expressed as the entropy of the label distribution $\boldsymbol{y}$ and the Kullback-Leibler divergence between $\boldsymbol{y}$ and the distribution of the predictions $\hat{\boldsymbol{y}}$, minimizing Equation (3.7) directly leads to the desired minimization of the difference of $\hat{\boldsymbol{y}}$ from $\boldsymbol{y}$.

Despite its simplicity, $\mathscr{L}_{\text{CE}}$ provides satisfying results for FR, which is why it is still viable in FR-related tasks when only a face feature extractor is required. However, $\mathscr{L}_{\text{CE}}$ is not optimal, which motivated a considerable amount of publications to gradually improve the discriminability of the features $\boldsymbol{f}$. Among these publications, two paradigms can be identified: 1) Using pairwise labels to directly minimize feature distances; and 2) utilizing class-level labels to minimize feature distances indirectly.

3.4.1 Pairwise Losses

Softmax CE loss was combined with feature-distance-based losses leveraging pairwise labels to improve the discriminability of the feature vector $\boldsymbol{f}$ directly. Sun *et al.* [198, 200] employed the *contrastive loss* [25], which minimizes the pairwise distances between two faces of the same identity and maximizes the distances for imposter pairs. Since contrastive loss only considers pairs, the feature distance is absolute. This causes issues as intra-class variance differs for every identity. Therefore, Schroff *et al.* [187] utilized the *triplet loss* $\mathscr{L}_{\text{Triplet}}$ [79, 217], which jointly minimizes the distance between a feature of an anchor face $\boldsymbol{f}_{\text{A}}$ and a positive face $\boldsymbol{f}_{\text{P}}$, *i.e.*, from the same identity, and maximizes the distance between $\boldsymbol{f}_{\text{A}}$ and the feature of a negative face $\boldsymbol{f}_{\text{N}}$, *i.e.*, from a different identity. Hence, by considering triplets

$$\mathscr{L}_{\text{Triplet}} = \left[\|\boldsymbol{f}_{\text{A}} - \boldsymbol{f}_{\text{P}}\|^2 - \|\boldsymbol{f}_{\text{A}} - \boldsymbol{f}_{\text{N}}\|^2 + \alpha\right]_+, \tag{3.8}$$

where α denotes a margin between positive and negative pairs and $[\cdot]_+ = \max(0, \cdot)$, the feature distance of a genuine pair is minimized relative to the feature distance to a sample from a different identity. Like contrastive loss, $\mathscr{L}_{\text{Triplet}}$ is also used in combination with softmax CE loss $\mathscr{L}_{\text{CE}}$ [169, 184], which shows a good trade-off between learning for face verification ($\mathscr{L}_{\text{Triplet}}$) and face identification ($\mathscr{L}_{\text{CE}}$). However, the effectiveness of $\mathscr{L}_{\text{Triplet}}$ highly depends on the triplet generation as the network does not learn much from easy samples, *i.e.*, from negative pairs, which are easy to differentiate due to, *e.g.*, opposite gender. To alleviate this dependency, Sohn [193] and Deng *et al.* [36] computed pairwise distances within the entire batch. Another solution was proposed with the *center loss* by Wen *et al.* [229], which minimizes the distance between the feature and the center of its identity in the feature space. By continuously updating all centers, all previous features are leveraged. In this way, the center loss is less susceptible to pair generation. The center loss was extended by Zhang *et al.* to jointly maximize the distance between two class centers [268].

With *circle loss*, Sun *et al.* [199] proposed a unified framework encompassing pairwise and class-level losses. In contrast to previous approaches, they re-weigh the pairwise distances to focus on those that deviate far from their optima.

[ix] To improve readability and following the notation in literature, losses are defined for a single sample $\mathscr{L} = \mathscr{L}(\{(\boldsymbol{X}, y), \theta)$ and the dependency on θ is omitted if not relevant to avoid ambiguity.

3.4.2 Class-Level Losses

Another trend was motivated by the analysis of Parde *et al.* [167], who identified a correlation between feature quality and its norm $\|\boldsymbol{f}\|_2$, which was later confirmed by Meng *et al.* [148] in their recent FR approach. Thus, by normalizing the features to lie on a hypersphere with a fixed radius s before passing them onwards to the last fully connected layer, Ranjan *et al.* [175] forced the network to shift focus onto the hard, low-quality samples with smaller feature distances. They prove that the scaling parameter s to map the normalized features to a hypersphere with a given radius depends on the number of classes M_{cls} and is crucial to provide "sufficient space" on the hypersphere. This concept was further extended by Hasnat *et al.* [68], who also normalized the variance by effectively employing a batch normalization layer (with $\boldsymbol{\gamma} = \mathbf{1}$ and $\boldsymbol{\beta} = \mathbf{0}$). In contrast to enforcing a fixed radius s of the hypersphere by normalizing and scaling the features [175], the *ring loss* [277] uses an additional loss to minimize the feature distance to a trainable radius.

In 2016, Liu *et al.* [132] initiated a new direction in FR by analyzing the decision boundaries of the softmax CE loss. With the help of the definition of a fully connected layer and the softmax function (see Equations (2.2) and (3.6), respectively),[x] Equation (3.7) can be rewritten as follows:

$$\mathscr{L}_{\text{CE}} = -\log\left(\frac{\mathrm{e}^{[\tilde{\boldsymbol{y}}]_y}}{\sum_{m=1}^{M_{\text{cls}}} \mathrm{e}^{[\tilde{\boldsymbol{y}}]_m}}\right) = -\log\left(\frac{\mathrm{e}^{\left[\boldsymbol{W}^\mathsf{T}\boldsymbol{f}\right]_y}}{\sum_{m=1}^{M_{\text{cls}}} \mathrm{e}^{\left[\boldsymbol{W}^\mathsf{T}\boldsymbol{f}\right]_m}}\right). \tag{3.9}$$

To compute the yth entry of the vector $\boldsymbol{W}^\mathsf{T}\boldsymbol{f}$ only the yth column of $\boldsymbol{W}$ is relevant, which results in the scalar product

$$[\boldsymbol{W}]_{:,y} \cdot \boldsymbol{f} = \|[\boldsymbol{W}]_{:,y}\| \|\boldsymbol{f}\| \cos\left([\boldsymbol{\alpha}]_y\right) \tag{3.10}$$

with $[\boldsymbol{\alpha}]_y$ denoting the angle between the vectors $[\boldsymbol{W}]_{:,y}$ and $\boldsymbol{f}$. Since the denominator in Equation (3.9) is independent of y, a feature $\boldsymbol{f}$ is classified as class y if

$$\|[\boldsymbol{W}]_{:,y}\| \|\boldsymbol{f}\| \cos([\boldsymbol{\alpha}]_y) > \|[\boldsymbol{W}]_{:,j}\| \|\boldsymbol{f}\| \cos([\boldsymbol{\alpha}]_j) \qquad \forall\, j \neq y. \tag{3.11}$$

Thus, to obtain a correct classification, the angle $[\boldsymbol{\alpha}]_y$ must be close to 0. Hence, $[\boldsymbol{W}]_{:,y}$ constitutes a trainable vector representing the yth identity. In this case, the training is referred to as proxy-based learning since the similarity between samples and a set of proxies representing each class is optimized.

With their *large-margin softmax loss*, Liu *et al.* [132] proposed a more robust decision boundary than the original $\mathscr{L}_{\text{CE}}$. To enforce an angular margin m_1 they substitute $\cos([\boldsymbol{\alpha}]_y)$ with a generalization of the form $\Xi([\boldsymbol{\alpha}]_y) = \cos(m_1\,[\boldsymbol{\alpha}]_y)$ to ensure a monotonically decreasing function in $[\boldsymbol{\alpha}]_y$.

With *SphereFace*, Liu *et al.* [131] further investigated this concept by normalizing the weight matrix $\|\boldsymbol{W}\| = 1$. Hence, the decision for a class is solely dependent on the angle

[x] For the sake of simplicity, the bias $\boldsymbol{b}$ and the depth L information in $\boldsymbol{W} = \boldsymbol{W}^{[L]}$ are omitted.

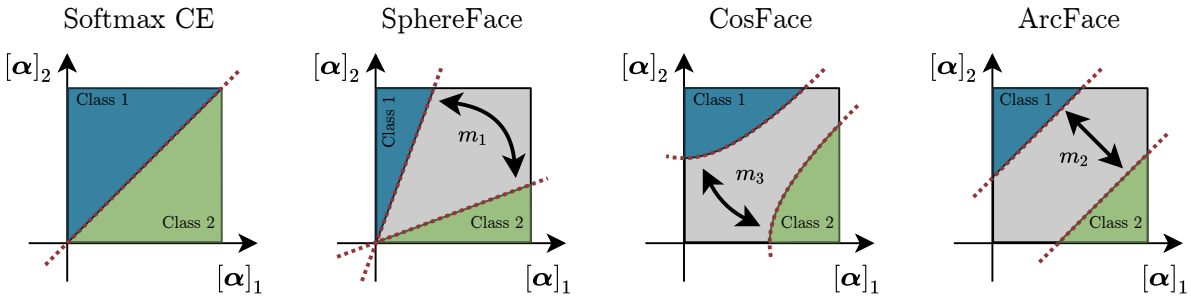

Figure 3.5: Decision boundaries (dashed) for binary classification with different loss functions and their respective margins (grey). Adapted from [33].

$[\boldsymbol{\alpha}]_y$ on the hypersphere (*cf.* Equation (3.11)), resulting in the angular softmax loss. By also minimizing the minimum hyperspherical energy of the output layer, the classes are more evenly distributed on the hypersphere as demonstrated by Liu *et al.* [130]. This issue of unbalanced distributed features in SphereFace [131] is also addressed by additionally maximizing the distance between class centers of dissimilar classes [42, 273].

A slightly different approach was proposed by Wang *et al.* with *NormFace* [214]. They used the normalization of features to a hypersphere with a fixed radius s from Ranjan *et al.* [175] and combined it with weight normalization $\|\boldsymbol{W}\| = 1$. In this way, NormFace minimizes the cosine distance directly without any influence from the feature quality $\|\boldsymbol{f}\|$ as in SphereFace [131]. The normalization and feature rescaling to $\|\boldsymbol{f}\| = s$ constitutes the first step towards implementing more sophisticated losses [33, 34, 87, 108, 109, 148, 212, 215, 223].

To ease the cumbersome training caused by the multiplicative margin m_1 in SphereFace [131], an additive margin was proposed [33, 212, 215]. With *CosFace* [215] and *additive margin loss* [212], both groups independently proposed to leverage the normalization and scaling of $\boldsymbol{f}$ and $\boldsymbol{W}$ as in [214], and further employ an additive margin $\Xi([\boldsymbol{\alpha}]_y) = \cos([\boldsymbol{\alpha}]_y) - m_3$. In this way, the cosine distance between two features is minimized directly [214] yet with an additive margin m_3. As opposed to [212, 215], *ArcFace* applies the margin in an angular manner $\Xi([\boldsymbol{\alpha}]_y) = \cos([\boldsymbol{\alpha}]_y + m_2)$, which is considered a breakthrough in FR research and is widely addressed in recent FR research [17, 34, 42, 87, 108, 109, 148, 223, 273].

The multiplicative margin m_1 in SphereFace [131, 132], the additive margin m_3 in CosFace [212, 215], and the additive angular margin m_2 in ArcFace [33] all improve discriminability in the feature space by enforcing a margin. Despite their similarity, the margins result in substantially different decision boundaries as depicted in Figure 3.5. In contrast to the non-linear margins of SphereFace and CosFace, the additive angular margin from ArcFace is constant $\forall\ [\boldsymbol{\alpha}]_i$. Hence, only ArcFace ensures that lookalike $[\boldsymbol{\alpha}]_i \approx [\boldsymbol{\alpha}]_j$ and dissimilar classes $[\boldsymbol{\alpha}]_i \gg [\boldsymbol{\alpha}]_j$ are equally separated by a uniform margin in the angular feature space. Combining all margins into a unified framework $\Xi([\boldsymbol{\alpha}]_y) = \cos(m_1 [\boldsymbol{\alpha}]_y + m_2) - m_3$ was analyzed by Deng *et al.* [33]. However, their ablation study on suitable margins m_i revealed that such losses are upper-bounded by ArcFace ($m_1 = m_3 = 0$).

After providing a new baseline with ArcFace, special attention was given to the hard, *i.e.*, difficult, samples. Wang *et al.* [223] extended the concept of ArcFace [33] by introducing an extra margin on the misclassified vectors since well-separated features do not contribute substantially to the learning task. By incorporating curricular learning, *i.e.*, easy samples at the beginning of the training and hard samples in later stages, Huang *et al.* [87] adaptively adjusted the importance of easy and hard samples during the training (*CurricularFace*). To obtain a universal face representation for all faces, including hard samples, Shi *et al.* [190] applied occlusion, low resolution, and head pose data augmentation to generate hard training samples, whose features are divided into multiple sub-features with confidence scores.

A distinct approach to cope with hard or noisy data constitutes extending the representation of a face, which is typically a point $\boldsymbol{f}$ in the feature space. To improve robustness to noisy data, Shi *et al.* [189] modeled the uncertainty of every feature of a given pretrained model by adding a variance term. This approach of representing the face as a Gaussian distribution in the feature space was trained end-to-end by Chang *et al.* [17]. Deng *et al.* [32] proposed to use multiple class representatives instead of a single one. Then, Deng *et al.* [34] extended their previous approach by incorporating the uncertainty modeling as in [17, 189], which resulted in variational class-wise prototypes. Revisiting the findings of Prade *et al.* [167], Meng *et al.* proposed with *MagFace* a magnitude-aware additive angular margin [148] $\Xi([\boldsymbol{\alpha}]_y) = \cos([\boldsymbol{\alpha}]_y + m_2(\|\boldsymbol{f}\|))$, which allows to adaptively select a suitable margin based on the feature quality $\|\boldsymbol{f}\|$. Class dependent margins to address dataset imbalances were also proposed by Liu *et al.* [126], who predicted an adaptive margin for CosFace and ArcFace utilizing reinforcement learning, and by Liu *et al.* [128] by employing a trainable additive margin m_3.

Besides the extensions mentioned above, Kim *et al.* proposed two different approaches based on ArcFace: *GroupFace* [108] combined the default instance-based representation of a face with group-aware representations to leverage specific group-dependent features. Moreover, they proposed *BroadFace* [109], which overcomes the restriction of only considering a limited number out of tens of thousands of identities per batch by buffering feature vectors of previous batches.

3.5 Experiments

After providing a general overview of every component required for FR in Sections 3.1 to 3.4, the remaining part of this chapter describes training strategies in Section 3.5.1 and the evaluation metrics in Section 3.5.2 in order to train and compare multiple datasets, FAPs, architectures, and loss functions. Moreover, these general FR models are used later in this work to extract identity features for face completion (see Chapter 4) or face aggregation (see Chapter 6), or to provide a baseline for partial FR (see Chapter 5).

3.5.1 Training Details

In order to determine how datasets, FAPs, architectures, and loss functions influence the FR performance, fixed training parameters are required. As elaborated in Section 3.1.1, there is no mutual agreement on a single training dataset in the related works. Nevertheless, MS1MV2 [33] and VGGFace2 [14] offer two viable options focusing on a considerable number of images and low noise. From all faces, five facial landmarks are extracted using the MTCNN [265]. Then the faces are aligned following two different FAPs as described in Section 3.2, yielding aligned faces $\boldsymbol{I}_{\text{align}}$ with resolutions $r \times r$ px with $r \in \{112, 160, 224\}$.

To increase the variance of the samples during training, data augmentation is employed. However, the face alignment renders some conventional augmentations, *e.g.*, random cropping, vertical flipping, rotating, and translation, rather unpopular. Still, multiple data augmentations do not interfere with the facial alignment and thus are widely used in FR. Most approaches [148, 199, 281] utilize horizontal flipping as it does not change the face alignment due to the symmetrical target facial landmark positions. Even though faces are not perfectly symmetric, this simple augmentation is particularly beneficial for faces with extreme head poses as it allows the network to become more invariant to such cases. Besides horizontal flipping, slightly changing the brightness, contrast, and saturation allows the face to remain realistic while still providing different pixel values to the network. Before applying these operations, the aligned image $\boldsymbol{I}_{\text{align}} \in [0, 255]$ is transformed to float $\boldsymbol{I}_{\text{align}} \in [0, 1]$.

- To change the brightness, a scalar γ_{b}, drawn from a uniform distribution $\gamma_{\text{b}} \sim \mathcal{U}_1(-\gamma_{\text{b},\Delta}, \gamma_{\text{b},\Delta})$, is added to the image by

$$\boldsymbol{I}_{\text{aug}} = \boldsymbol{I}_{\text{align}} + \gamma_{\text{b}}. \tag{3.12}$$

- The contrast is altered independently for every channel by scaling every pixel's distance to the channel-wise mean $\left[\boldsymbol{\mu}_{\text{I}_{\text{align}}}\right]_c$ with a scalar γ_{c} drawn from a uniform distribution $\gamma_{\text{c}} \sim \mathcal{U}_1(\gamma_{\text{c,min}}, \gamma_{\text{b,max}})$ as follows:

$$\left[\boldsymbol{I}_{\text{aug}}\right]_{:,:,c} = \gamma_{\text{c}} \left(\left[\boldsymbol{I}_{\text{align}}\right]_{:,:,c} - \left[\boldsymbol{\mu}_{\text{I}_{\text{align}}}\right]_c \right) + \left[\boldsymbol{\mu}_{\text{I}_{\text{align}}}\right]_c. \tag{3.13}$$

- As opposed to the previous augmentations, saturation augmentation is applied in the HSV color space. There, only the saturation channel is multiplied with a scalar γ_{s} drawn from a uniform distribution $\gamma_{\text{s}} \sim \mathcal{U}_1(\gamma_{\text{s,min}}, \gamma_{\text{s,max}})$ by

$$\left[\boldsymbol{I}_{\text{aug}}\right]_{:,:,2} = \gamma_{\text{s}} \left[\boldsymbol{I}_{\text{align}}\right]_{:,:,2}. \tag{3.14}$$

To ensure that the augmented image $\boldsymbol{I}_{\text{aug}}$ fulfills the properties of every float image, it is clipped to $[0, 1]$ by

$$\boldsymbol{I}_{\text{aug}} \leftarrow \min\left(\max\left(\boldsymbol{I}_{\text{aug}}, 0\right), 1\right). \tag{3.15}$$

To ease convergence, the image is normalized to a symmetrical value range $[-1, 1]$ with a mean value close to zero

$$\boldsymbol{I}_{\text{aug}} \leftarrow \boldsymbol{2} I_{\text{aug}} - 1. \tag{3.16}$$

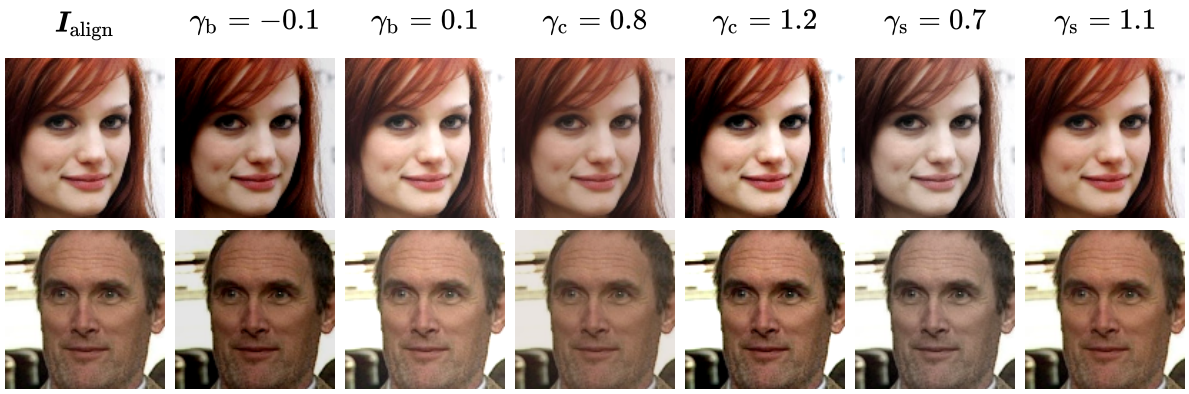

Figure 3.6: Impact of different values of brightness γ_{b}, contrast γ_{c}, and saturation γ_{s} parameters when augmenting the aligned image $\boldsymbol{I}_{\text{align}}$.

While the objective of data augmentation is to increase variety, it is vital to guarantee the realism of a face and match the distribution of the benchmark datasets. *E.g.*, one could train an FR system specifically for low-light conditions by reducing the brightness of an otherwise normal training dataset. Hence, suitable limits $\gamma_{\text{b},\Delta}, \gamma_{\text{c,min}}, \gamma_{\text{c,max}}, \gamma_{\text{s,min}}, \gamma_{\text{s,max}}$ must be determined.

Figure 3.6 depicts the augmentations for the limits $\gamma_{\text{b},\Delta} = 0.1, \gamma_{\text{c,min}} = 0.8, \gamma_{\text{c,max}} = 1.2, \gamma_{\text{s,min}} = 0.7, \gamma_{\text{s,max}} = 1.1$, which are employed when training convolutional neural networks (CNNs) throughout this work. All augmented images are still very realistic and force the network to become robust against minor variations in brightness, contrast, and saturation, which frequently occur in benchmark datasets. In her analysis, Sun [7^{+}] demonstrated that these limits provide a natural augmentation, and a reduction or an increment of these augmentation intensities leads to slightly inferior FR performance.

All previous data augmentations are applied independently in succession with a probability of $p_{\text{aug}} = 50\%$ each, *i.e.*, multiple augmentations may be applied to an image. To ensure that a combination of all augmentations does not produce any unwanted results, the limits are chosen rather conservative. Moreover, the uniform probability distribution of the augmentation parameters causes the augmentation to be often subtle and barely noticeable.

Other augmentations like Gaussian blur, motion blur, and JPEG quality can also be helpful; however, they depend on the purpose of the FR system, *e.g.*, motion blur is often employed for video FR. As a more advanced data augmentation technique, *cutout* was proposed by Devries *et al.* [38], in which square areas within the image are masked to make the model consider less distinctive parts of an object and recognize it from partial views. Hence, cutout can be interpreted as a special localized form of dropout applied to the input layer. *Mixup* augmentation [263] creates a weighted combination of two images and their labels. Instead of only removing pixels as in cutout, Yun *et al.* [258] proposed *cutmix*, which substitutes the masked pixels with patches from another image and mixes the labels according to the number of pixels. Despite their benefits, these advanced augmentation techniques are barely used in FR.

Along with data augmentation and normalization, additional training parameters, which were introduced in Sections 2.4 and 2.6, are required to train various FR models. Since the spatial resolution is substantially reduced at deeper layers of the network, the memory footprint of a standard FR model is comparatively low. Thus, batch sizes of $N_\mathrm{b} = 100$ are manageable even on consumer graphical processing units (GPUs), which is sufficient to obtain a high amount of variety and a steady gradient from the samples within the batch. To improve generalization, the methods presented in Section 2.6 are widely used. Specifically, batch normalization ($\alpha_\mathrm{BN} = 0.995$ and $\epsilon_\mathrm{BN} = 0.001$), L_2-regularization on all weights with a factor $\lambda_\mathrm{reg} = 5 \cdot 10^{-5}$ and dropout with a probability of $p_\mathrm{d} = 40\%$ before the feature embedding layer (*cf.* Table 3.4) are employed. The network is then optimized with ADAM [110] ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) for 20 epochs with an initial learning rate of $\eta = 0.05$, which is reduced by a factor of $\gamma_\mathrm{lr} = 4$ every 4 epochs.

In various training runs, multiple parameters are varied, resulting in slightly different training parameters:

- Dataset: VGGFace2 [14] and MS1MV2 [33] with different FAPs (custom and following Deng *et al.* [33]) and image resolutions $r \in \{112, 160, 224\}$.
- Architecture: ResNet-v2 with ReLU activation function and different depths $L \in \{18, 34, 50, 101\}$. The incorporation of GAP before the feature embedding layer and the number of hidden neurons M_f in the feature layer $\boldsymbol{f}$ are varied. Following Hasnat *et al.* [68], batch normalization is utilized after $\boldsymbol{f}$.
- Loss: Softmax CE is investigated with and without additive angular margin [33]. For training with additive angular margin, the margin is set to $m_2 = 0.3$ when training on VGGFace2, as proposed by the authors. For MS1MV2, the default parameters as in the paper are utilized ($s = 64$ and $m_2 = 0.5$).

3.5.2 Evaluation Details

Whereas the model is trained with $\boldsymbol{I}_\mathrm{aug}$, only aligned faces $\boldsymbol{I}_\mathrm{align}$ are utilized to ensure a deterministic evaluation.[xi] Nevertheless, some FR approaches [215, 229] perform horizontal flipping and concatenate the feature vectors of the flipped and original image to slightly improve the performance. However, in this work, no data augmentation is employed at test time.

In general, FR models are evaluated following the protocols published with the benchmark datasets enumerated in Table 3.2. Thus, every face is aligned with the same FAP as the training dataset and then processed by the CNN to obtain the respective features $\boldsymbol{f}$. Independent of the evaluation protocol (see Figure 3.1), it is necessary to classify whether two face images, $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, belong to the same identity. Typically, this is achieved by computing the cosine distance between their features $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$

$$d\left(\boldsymbol{f}_1, \boldsymbol{f}_2\right) = 1 - \frac{\boldsymbol{f}_1 \cdot \boldsymbol{f}_2}{\|\boldsymbol{f}_1\| \|\boldsymbol{f}_2\|}. \tag{3.17}$$

[xi] For the remaining part of this work, the indices $_\mathrm{aug}$ and $_\mathrm{align}$ are omitted if not necessary to avoid ambiguity.

The cosine distance $d(\boldsymbol{f}_1, \boldsymbol{f}_2) \in [0, 2]$ measures the angular distance in feature space, which is indirectly optimized by many loss functions introduced in Section 3.4.2. Hence, $d(\boldsymbol{f}_1, \boldsymbol{f}_2)$ corresponds to the dissimilarity of two faces, $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. By definition, genuine pairs will ideally yield a small cosine distance ≈ 0. However, cosine distances ≈ 1 are more prominent in imposter pairs as ≈ 2 would imply that $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are entirely contrary. Thus, all identity features must be contrary, *i.e.*, dark eye color versus bright eye color, *etc.*, making it highly unlikely to encounter a face fulfilling all requirements. Moreover, a cosine distance ≈ 1 is typically observed when comparing any face with an image not containing a face, such as an image of a single color. Since an image without a face should ideally provoke no activation within the network, the network outputs a feature vector $\boldsymbol{f} \approx \mathbf{0}$. Hence, in practice, cosine distances rarely occupy the entire range $[0, 2]$ as most distances of imposter pairs are centered around 1.

3.5.2.1 Face Verification

For face verification protocols, a list of N triplets is given $\left[\left(\boldsymbol{I}_1^{(n)}, \boldsymbol{I}_2^{(n)}, y^{(n)}\right)\right]_{n=1}^{N}$, which comprise an image pair $\boldsymbol{I}_1, \boldsymbol{I}_2$, and a binary ground-truth label $y \in \{0, 1\}$ indicating the image pair is imposter (0) or genuine (1). To obtain a binary prediction $\hat{y}$ from the network, a dataset-wide discrimination threshold t is applied to the continuous distance $d \in [0, 2]$ by

$$\hat{y}(\boldsymbol{f}_1, \boldsymbol{f}_2) = \begin{cases} 0 & \text{if } d(\boldsymbol{f}_1, \boldsymbol{f}_2) \geq t, \\ 1 & \text{if } d(\boldsymbol{f}_1, \boldsymbol{f}_2) < t. \end{cases} \tag{3.18}$$

By comparing the prediction $\hat{y}$ with the ground truth y, the prediction is classified as *true positive* (TP) ($y = \hat{y} = 1$), *true negative* (TN) ($y = \hat{y} = 0$), *false positive* (FP) ($y = 0$ and $\hat{y} = 1$) and *false negative* (FN) ($y = 1$ and $\hat{y} = 0$). In this way, the face verification *accuracy* (Acc) of a FR system is calculated as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.19}$$

where Acc depends on the choice of the threshold t. However, by choosing t such that Acc is maximized, information from the benchmark dataset is leveraged resulting in a bias. Therefore, computing Acc as in Equation (3.19) renders its value meaningless when determining Acc of an FR system that is exposed to *unknown* data. To alleviate this issue, 10-fold *cross-validation* with deterministic, predefined folds is employed for most benchmarks (LFW, CALFW, CPLFW, CFP, AgeDB, IJB-A, and YTF).[xii] In this way, t is computed to maximize Acc_{val} when using nine out of ten validation folds. Then, the test accuracy Acc_{test} is calculated on the left-out fold with t maximizing Acc_{val}. After repeating this process ten times, the final actual accuracy on the dataset is obtained by averaging all ten test accuracies Acc_{test}.

Even though Acc is a popular metric for face verification performance, it is inappropriate when evaluating typical scenarios in biometrics. The issue of Acc lies in its equal

[xii] No predefined data splits into folds for the IJB-B and IJB-C face verification benchmarks are provided.

weighting of genuine and imposter pairs. Thus, *e.g.*, in security-sensitive applications, FPs need to be minimized at the cost of obtaining fewer TPs. The *receiver operating characteristic* (ROC),[xiii] *i.e.*, the *true acceptance rate* (TAR) as a function of the *false acceptance rate* (FAR),[xiv] focuses on providing more insights into this regard. Formally, TAR and FAR are defined as

$$TAR = \frac{TP}{TP + FN}, \tag{3.20}$$

$$FAR = \frac{FP}{FP + TN}. \tag{3.21}$$

Thus, a large discrimination threshold t ensures that FP is small, resulting in $FAR \approx 0$, as desired for security-sensitive applications. On the downside, a large t also misses the classification of many positives, causing a large FN. To capture this trade-off, TAR is computed for the threshold t, which results in a predetermined FAR. *E.g.*, $TAR@FAR = 0.01$ denotes TAR calculated for a threshold, which led to $FAR = 0.01$.

Besides computing TAR for different FAR, the *equal error rate* (EER) denotes the error rate when both error rates (FAR and false reject rate (FRR), where $FRR = 1 - TAR$) are equal $FAR = FRR$. Thus, a lower EER is desirable. When the impacts of FPs and FNs are considered equally harmful, EER constitutes the most popular metric to describe a biometric system.

3.5.2.2 Closed-Set Face Identification

In contrast to face verification, face identification tries to determine the identity of a face. In real-world applications, one has a gallery of, *e.g.*, mugshot images with their identity labels and wants to determine the identity of a face by comparing it with every image in the gallery. This is described by a gallery $\mathcal{G} = \left\{ \left(\boldsymbol{I}_{\mathrm{G}}^{(n)}, y_{\mathrm{G}}^{(n)} \right) \right\}_{n=1}^{N_{\mathrm{G}}}$ and a probe set $\mathcal{P} = \left\{ \left(\boldsymbol{I}_{\mathrm{P}}^{(n)}, y_{\mathrm{P}}^{(n)} \right) \right\}_{n=1}^{N_{\mathrm{P}}}$ with y_{G} and y_{P} denoting the identity of the gallery image $\boldsymbol{I}_{\mathrm{G}}$ and probe image $\boldsymbol{I}_{\mathrm{P}}$, respectively.

For closed-set face identification, all identities in the probe set $\mathcal{P}$ also must be present in the gallery $\mathcal{G}$. Thus, $\forall \left(\boldsymbol{I}_{\mathrm{P}}^{(i)}, y_{\mathrm{P}}^{(i)} \right) \in \mathcal{P} \, \exists \left(\boldsymbol{I}_{\mathrm{G}}^{(j)}, y_{\mathrm{G}}^{(j)} \right) \in \mathcal{G} \mid y_{\mathrm{G}}^{(j)} = y_{\mathrm{P}}^{(i)}$. Therefore, closed-set FR simplifies the identification task by leveraging the prior knowledge that y_{P} is also part of $\mathcal{G}$. First, the pairwise feature distances d_{ij} between the ith probe image $\boldsymbol{I}_{\mathrm{P}}^{(i)}$ and all gallery images $\boldsymbol{I}_{\mathrm{G}}^{(j)} \in \mathcal{G}$ are calculated according to Equation (3.17). The smallest distance between the probe and all gallery images of the same identity is calculated by

$$d_{i*} = \min_{j} \left\{ d_{ij} \mid y_{\mathrm{G}}^{(j)} = y_{\mathrm{P}}^{(i)} \right\}. \tag{3.22}$$

Then, the match of the ith probe image $\boldsymbol{I}_{\mathrm{P}}^{(i)}$ with $\mathcal{G}$ is said to have *rank* R if d_{i*} is the Rth smallest feature distance. This is denoted by $\mathrm{rank}(\boldsymbol{I}_{\mathrm{P}}^{(i)}) = R$. In other words, R

[xiii] also referred to as detection error trade-off (DET)

[xiv] In biometrics, the true and false positive rates are typically referred to as true and false acceptance rates.

is the index of d_{i*} in a list of distances $[d_{i1}, d_{i2}, \ldots, d_{iN_\mathrm{G}}]$, which is sorted in ascending manner.

With the help of the rank of a match, it is possible to investigate more sophisticated face identification applications, *e.g.*, in law enforcement when displaying a list of R suspects for a given probe face. Thus, it is crucial to evaluate how often the correct suspect appears on this list. This application is grasped by the *cumulative match characteristic* (CMC), which displays the *true positive identification rate* ($TPIR$) at rank R and is computed by

$$TPIR(R) = \frac{\left|\left\{\boldsymbol{I}_\mathrm{P}^{(i)} \mid \mathrm{rank}\left(\boldsymbol{I}_\mathrm{P}^{(i)}\right) \leq R\right\}\right|}{|\mathcal{P}|}. \tag{3.23}$$

Typically, $TPIR(1)$ is reported and referred to as the rank 1 identification accuracy. $TPIR$ is highly dependent on the gallery size N_G as $TPIR(N_\mathrm{G}) = 1$. Hence, it is vital to either normalize the rank by the gallery size or only compare face identification benchmarks with similar gallery sizes.

3.5.2.3 Open-Set Face Identification

In contrast to closed-set face identification, it cannot be relied upon that the identity of a probe face is also part of the gallery in open-set face identification. To analyze this scenario, the probe set is split into two disjoint subsets $\mathcal{P} = \mathcal{P}_\mathrm{G} \cup \mathcal{P}_\mathrm{N}$ ($\mathcal{P}_\mathrm{G} \cap \mathcal{P}_\mathrm{N} = \emptyset$) with $\mathcal{P}_\mathrm{G}$ containing images of identities in $\mathcal{G}$ and $\mathcal{P}_\mathrm{N}$ images of identities, which do not form part of $\mathcal{G}$.

Therefore, a discrimination threshold t is employed as in face verification, which labels the probe face as "not in $\mathcal{G}$" if all pairwise feature distances d_{ij} exceed t. This means that for open-set face identification, Equation (3.23) needs to be adapted as follows:

$$TPIR(R, t) = \frac{\left|\left\{\boldsymbol{I}_\mathrm{P}^{(i)} \in \mathcal{P}_\mathrm{G} \mid \mathrm{rank}\left(\boldsymbol{I}_\mathrm{P}^{(i)}\right) \leq R \text{ and } d_{i*} < t\right\}\right|}{|\mathcal{P}_\mathrm{G}|} \tag{3.24}$$

With the help of the additional subset $\mathcal{P}_\mathrm{N}$, it is also possible to evaluate how often probe images $\boldsymbol{I}_\mathrm{P} \in \mathcal{P}_\mathrm{N}$ are erroneously assigned an identity. This statistic measures the *false positive identification rate* ($FPIR$), which is computed by

$$FPIR(t) = \frac{\left|\left\{\boldsymbol{I}_\mathrm{P}^{(i)} \in \mathcal{P}_\mathrm{N} \mid \min_j \{d_{ij}\} < t\right\}\right|}{|\mathcal{P}_\mathrm{N}|}. \tag{3.25}$$

Generally, the open-set face identification performance of an FR system can be described by a surface in a three-dimensional parameter space, spanned by $TPIR$, $FPIR$, and R. However, most information is only relevant in very specific scenarios such that the analysis is typically reduced to two-dimensional slices to analyze the most prominent scenarios: 1) the ROC, which displays $TPIR$ in terms of $FPIR$ for $R = 1$; and 2) the CMC, which illustrates $TPIR$ as a function of R for $FPIR = 1$. In this way, open-set face identification not only allows the analysis of how often the correct identity

Table 3.5: Ablation study on datasets and preprocessing. Verification accuracy *Acc* and $TPIR$ at rank $R = 1$ in % for a fixed architecture (ResNet-v2 [72] with depth $L = 50$, size of the feature layer $M_\mathrm{f} = 256$, GAP after the feature layer and softmax cross-entropy loss without additive angular margin [33]). $^\uparrow$ denotes that the results were obtained by upscaling the aligned faces from the original resolution $r = 112$ since unaligned faces with higher resolutions were unavailable. The highlighted models are analyzed in detail in the next section.

Training Dataset	Alignment Policy		Image Resolution r	Verification					ID
	Training	Test		LFW	CPLFW	CFP	CALFW	AgeDB	MegaFace
MS1MV2	ArcFace		112	99.18	80.30	90.77	**90.23**	92.52	66.14
VGGFace2	ArcFace		112	99.40	85.87	94.97	88.35	90.15	61.24
VGGFace2	Custom		112	99.43	85.70	94.76	87.95	89.63	61.25
VGGFace2	Custom		160	99.42	87.40	96.37	88.97	91.70$^\uparrow$	66.27$^\uparrow$
VGGFace2	Custom		224	**99.65**	**87.73**	**96.97**	89.87	**92.62**$^\uparrow$	71.21$^\uparrow$
VGGFace2	ArcFace	Custom	112	97.63	78.05	91.93	78.48	82.17	27.89
VGGFace2	Custom	ArcFace	112	97.68	79.88	91.60	80.88	85.90	24.55

is within the top R identity proposals (CMC), but also how often the correct identity is identified when a fixed number of erroneous identifications of identities, which do not form part of the gallery, are permitted (ROC).

3.6 Results

3.6.1 Ablation Study

The ablation study is divided into Tables 3.5 and 3.6. While Table 3.5 depicts the impact of different datasets and preprocessing on the FR performance, Table 3.6 focuses on the architecture.

In terms of the training dataset, the results do not provide a clear picture. Table 3.5 suggests that VGGFace2 outperforms MS1MV2 on half (LFW, CPLFW, and CFP) of the evaluated benchmark datasets, whereas the results favor MS1MV2 (see Table 3.6) after dispensing with GAP and incorporating additive angular margin. However, only on VGGFace2 the maximum accuracy on the CFP benchmark is obtained. This affirms the claim of Cao *et al.* [14] that VGGFace2 contains faces with more significant pose variations. Another aspect is the high amount of noise present in MS1MV2. Even though Deng *et al.* [33] employed annotators familiar with the identities' ethnicity, detecting falsely labeled data is particularly difficult in case of extreme head poses. Thus, it would be no surprise if more mistakes are made under these circumstances or if more faces are removed due to high uncertainty. Both lead to comparably low performance for frontal to pose comparisons in the CFP benchmark. Hence, both datasets are viable. However, when training with additive angular margin, it is important to use MS1MV2 as it benefits from the high number of identities.

As indicated in Section 3.2, alignment is essential for FR. Nevertheless, there is no mutual agreement on the alignment policy. Table 3.5 illustrates the FR performance for models trained on the VGGFace2 dataset with different alignment policies introduced in Section 3.2. Both policies yield comparable results with a slight advantage of the policy proposed in ArcFace [33]. This indicates that both FAPs cover all relevant information necessary to distinguish two faces despite their vertical offset of 15%. Therefore, all identity information relevant for the FR network must lie at the center of the face and cropping part of the chin – as when using the ArcFace FAP (*cf*. Figure 3.3) – does not mitigate the FR performance. When comparing the results with the analysis by Xu *et al*. [242], it seems like the looser crop in their FAP is decisive for obtaining the best FR performance.

When the FAPs during training and testing are not identical, a substantial decline in FR performance can be identified. This demonstrates the dependency of the FR models on a consistent face alignment. In terms of robustness, both FAPs are relatively equal. While the FAP from ArcFace shows advantages on CFP and MegaFace, the custom FAP is favored for CPLFW, CALFW, and AgeDB. Hence, the choice of FAP has only a minor effect on the FR performance as long as training and benchmark datasets are aligned based on the same FAP.

As expected, image resolution r clearly impacts the performance, improving non-saturated benchmarks by over 1% compared to the default resolution $r = 112$. However, the improvement from $r = 160$ to $r = 224$ is relatively small on CPLFW and CFP, which suggests saturation due to the limited available resolution before alignment. The inherent higher amount of details in the input face aids the model in making better decisions and is decisive for very similarly looking faces that differ only by nuances. This also demonstrates that despite the increased resolution r the receptive field still suffices to capture global features, which are then encoded in deeper layers. The increased resolution does not impact the overall number of parameters due to the GAP before the bottleneck layer providing the features $\boldsymbol{f}$. However, it leads to higher memory load as feature maps of higher spatial resolutions need to be stored in memory to compute the gradients during backpropagation. For even larger image resolutions, one must consider the network's receptive field and eventually compensate it by incorporating dilated convolutions. Moreover, when training without GAP, the quadratic dependency of the number of parameters of the feature layer on r results in a shift of the network's focus from the convolutional layers to the bottleneck layer, which needs to be accounted for. Overall, the employment of higher resolutions is also limited by the poor availability of high-resolution training datasets. *E.g.*, MS1MV2 is only available as an aligned version cropped to $r = 112$.

Table 3.6 provides FR results for different architectures. As expected, training a ResNet with more layers L consistently boosts the performance as the network can extract more sophisticated features. Furthermore, the skip connections in the ResNet allow more combinations of feature maps at different depths for higher L. Doubling the layers also roughly doubles the number of parameters.[xv] Moreover, the memory

[xv] Note that for ResNets up to $L = 34$ a residual unit comprising two 3×3 convolutions is used

Table 3.6: Ablation study on the architecture. Verification accuracy *Acc* and $TPIR$ at rank $R = 1$ in %, and the number of parameters at inference for ResNet-v2 [72] at different depths L, varying size of the feature layer M_f, GAP after the feature layer, and additive angular margin [33]. All models are trained with a resolution $r = 112$ and the FAP as proposed in ArcFace [33]. The highlighted model is analyzed in detail in the next section.

Training					Verification					ID	#
Dataset	L	GAP	M_f	ArcFace	LFW	CPLFW	CFP	CALFW	AgeDB	MegaFace	Params
MS1MV2	18	✓	512	✓	99.38	82.17	91.06	92.38	93.75	74.13	11.4
MS1MV2	34	✓	512	✓	99.52	83.82	92.19	93.08	94.72	79.41	21.6
MS1MV2	50	✓	512	✓	99.53	86.50	92.93	94.22	96.13	87.85	24.6
MS1MV2	101	✓	512	✓	99.55	86.92	92.97	94.27	96.55	90.02	43.6
MS1MV2	50	✗	512	✓	99.60	87.62	92.99	**94.87**	97.03	**93.52**	40.3
MS1MV2	50	✗	256	✓	**99.67**	**87.92**	**93.17**	94.83	**97.05**	93.28	31.9
MS1MV2	50	✗	256	✗	99.38	83.57	92.04	92.07	93.83	75.84	31.9
MS1MV2	50	✓	256	✗	99.18	80.30	90.74	90.23	92.52	66.14	24.6
VGGFace2	50	✗	256	✓	**99.43**	86.43	94.84	89.67	**91.68**	72.35	31.9
VGGFace2	50	✗	256	✗	**99.43**	**86.47**	94.71	**89.75**	91.33	**73.04**	31.9
VGGFace2	50	✓	256	✗	99.40	85.87	**94.99**	88.35	90.15	61.24	24.0

requirement is substantially higher and training time is longer, resulting in a cumbersome training on consumer GPUs. Hence, utilizing $L = 50$ offers a good trade-off between performance, memory efficiency, and training time.

Using GAP between the last convolutional and the bottleneck layer $\boldsymbol{f}$ averages the activations across the spatial dimensions for all 2048 feature maps. Thus, the network becomes more robust against the activations' spatial shifts frequently occurring under varying head poses. However, without GAP, the fully connected layer leverages spatial variations in activations for given feature maps at the cost of an increased parameter count. For an input resolution $r = 112$ and $M_\mathrm{f} = 512$ neurons in the bottleneck layer, dispensing with GAP adds $2048 \cdot 512 \cdot (4^2 - 1) = 15.7\mathrm{M}$ parameters. When training with faces cropped to $r = 224$, the fully connected layer comprises 51.4M parameters, more than doubling the number of parameters in all previous convolutional layers and heavily shifting the network's focus towards a single fully connected layer. The FR performance depicted in Table 3.6 illustrates that for CFP, which focuses on extreme head pose differences within the pairs, no significant difference between models trained with and without GAP can be identified. Thus, the GAP can cope with spatial shifts in feature maps of faces with large pose variations similar to a larger bottleneck layer while at the same time maintaining a low parameter count. On all remaining benchmarks, it is evident that many additional parameters in the bottleneck layer improve the FR performance substantially, which proves that the model is not prone to overfitting despite the huge number of parameters in the bottleneck layer.

The number of neurons M_f in the bottleneck layer only has a negligible influence on the FR performance. Nevertheless, it is decisive that M_f fits the training dataset and that $M_\mathrm{f} \ll M_\mathrm{cls}$ is fulfilled to ensure good generalization (see also Section 3.3). The

(*cf.* Figure 3.4), whereas a more parameter-efficient structure (1×1, 3×3, 1×1) is employed for deeper ResNets.

analysis in Table 3.6 further shows that a 256-dimensional feature vector $\boldsymbol{f}$ suffices to encode enough identity information. In contrast, incrementing the dimensionality of $\boldsymbol{f}$ to 512, accompanied by a doubling of the number of parameters of $\boldsymbol{f}$, does not lead to better performance. Overall, both choices ($M_\mathrm{f} \in \{256, 512\}$) are viable when using MS1MV2 as a training dataset. Since VGGFace2 only comprises $\approx 10\%$ of the identities (*cf.* Table 3.1), $M_\mathrm{f} = 256$ constitutes the better choice to ensure good generalization.

In accordance with the findings in the original publication [33], incorporating additive angular margin (ArcFace) leads to superior FR results when training on MS1MV2. Especially for networks without GAP, the additive angular margin can better leverage the additional parameters in the bottleneck layer to increase the discriminability of the feature vector $\boldsymbol{f}$. Nevertheless, a performance boost after training with additive angular margin can also be observed with GAP. On VGGFace2, there are no apparent benefits from adding ArcFace. This is probably due to the lower number of identities in the dataset and the necessary reduction of the margin $m_2 = 0.3$ (compared to $m_2 = 0.5$ on MS1MV2) to ensure convergence. Since the margin is enforced during training between the ground-truth identity and all remaining identities, it is conclusive that a lower margin together with a reduced number of identities leads to less impact on the performance. Thus, despite the substantial improvement by training with additive angular margins on MS1MV2, it is decisive to carefully choose the parameters according to the dataset in order to obtain a noticeable improvement.

3.6.2 Detailed Analysis

To obtain more insights on the FR performance, additional metrics as introduced in Section 3.5.2 are reported for selected methods, which form the feature extractor in Chapters 4 and 6 or a baseline in Chapter 5.[xvi] In particular, three ResNets with $L = 50$ layers, which are highlighted in Tables 3.5 and 3.6, are considered: While *MS-112-Arc* is trained with additive angular margin loss [33] on MS1MV2 with a resolution of $r = 112$, both other models do not incorporate the additive angular margin loss and utilize the VGGFace2 dataset with faces cropped to $r = 112$ (*VGG-112*) and $r = 160$ (*VGG-160*).

3.6.2.1 Face Verification

Figure 3.7 and Table 3.7 depict the ROC and important metrics describing the ROC, respectively. Both illustrations affirm the assumptions made in the previous ablation study. Utilizing additive angular margin in combination with the MS1MV2 dataset (MS-112-Arc) offers the best performance even against VGG-160, which is trained with a higher input resolution. When focusing solely on benchmarks with high head pose variances (CPLFW and CFP), VGG-112 and VGG-160 are more robust. Besides, higher input resolution always yields better performance.

[xvi] While the models utilized later in this work are trained with the same parameters according to Table 3.5 and Table 3.6, they use slightly different learning rates or batch sizes to maintain consistency in the respective task, which explains the minor deviations.

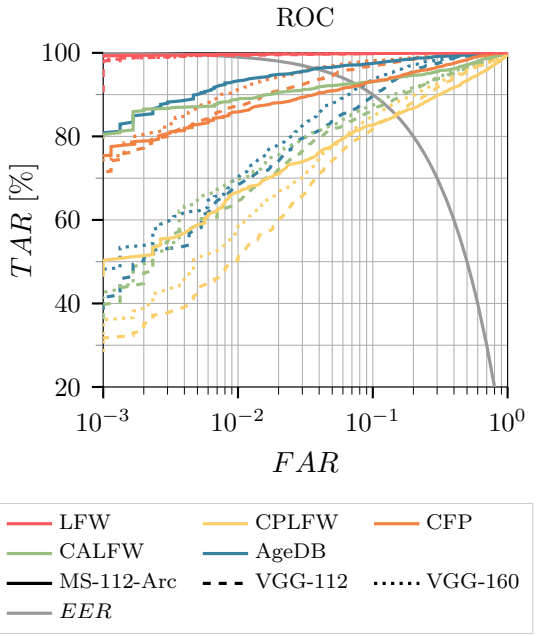

Figure 3.7: ROC for selected methods (line style) on various benchmarks (line color).

Table 3.7: Detailed evaluation of selected methods on multiple face verification benchmarks. $^{\uparrow}$ denotes that aligned faces with $r = 112$ were upscaled. All results are given in % and .

			$TAR@FAR =$		
Dataset	Method	EER	10^{-3}	10^{-2}	10^{-1}
LFW	MS-112-Arc	**0.46**	**99.30**	**99.63**	99.83
LFW	VGG-112	0.60	93.33	99.43	99.87
LFW	VGG-160	0.60	98.00	99.47	**99.90**
CPLFW	MS-112-Arc	14.96	**48.57**	**66.67**	82.73
CPLFW	VGG-112	14.75	31.73	51.03	81.83
CPLFW	VGG-160	**12.97**	31.80	59.00	**84.70**
CFP	MS-112-Arc	7.86	**76.26**	85.86	93.11
CFP	VGG-112	5.19	72.51	87.26	96.91
CFP	VGG-160	**3.71**	74.37	**91.17**	**98.14**
CALFW	MS-112-Arc	**7.36**	**80.47**	**88.87**	**93.10**
CALFW	VGG-112	12.21	39.73	64.53	86.17
CALFW	VGG-160	11.43	42.63	69.73	87.47
AgeDB	MS-112-Arc	**3.78**	**80.87**	**93.30**	**97.47**
AgeDB	VGG-112	10.15	39.53	68.40	89.73
AgeDB$^{\uparrow}$	VGG-160	8.05	50.60	70.27	93.70

From Figure 3.7 can be deduced that MS-112-Arc performs exceptionally well at very low $FAR = 10^{-3}$ despite its susceptibility to extreme head poses. Moreover, it reveals the limitations of the LFW benchmark as a difference between the methods is noticeable only for $FAR \approx 10^{-3}$. Generally, TARs for $FAR \approx 10^{-3}$ are afflicted with substantial uncertainty since they correspond to TAR when falsely predicting solely three imposter pairs as genuine. Hence, for a more detailed analysis at lower FAR, benchmark datasets comprising more pairs need to be employed.

The IJB-B and IJB-C benchmarks comprise 8.0M and 15.7M pairs. While the number of genuine pairs is relatively low (10k and 19k, respectively), the huge number of imposter pairs allows a detailed analysis of more practical use cases, *i.e.*, FAR as low as 10^{-6}. In contrast to previous face verification benchmarks, which considered a pair of two face images, both IJB benchmarks use pairs of so-called templates – a set containing an arbitrary number of still images and frames. Thus, to obtain a single feature vector representing a template, the features within the template are extracted separately from every face, L_2-normalized, and then averaged. In addition to the IJB-B/C benchmarks, the MegaFace verification benchmark allows computing meaningful TARs even for $FAR \approx 10^{-8}$ by encompassing $\approx 4 \cdot 10^9$ pairs.

Figure 3.8 and Table 3.8 depict the ROC and important metrics describing the ROC for the verification benchmarks mentioned above, which comprise millions of pairs. Both confirm the analysis obtained from Figure 3.7 and Table 3.7 in which MS-112-Arc obtains the best results for low FAR. Generally, an inferior performance on IJB-B/C compared to MegaFace is observed. This result can be considered unexpected as the templates in

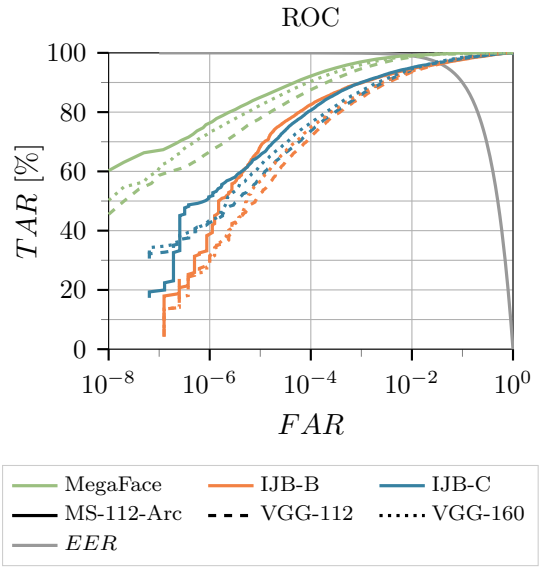

Figure 3.8: ROC for selected methods (line style) on IJB-B/C mixed media and the MegaFace (line color) face verification protocols.

Table 3.8: Detailed evaluation of selected methods on IJB-B/C mixed media and the MegaFace face verification benchmarks. $^{\uparrow}$ denotes that aligned faces with $r = 112$ were upscaled. All results are given in %.

			$TAR@FAR =$		
Dataset	Method	EER	10^{-6}	10^{-5}	10^{-4}
IJB-B	MS-112-Arc	3.62	**38.84**	**68.24**	**82.52**
IJB-B	VGG-112	3.66	30.51	52.43	71.76
IJB-B	VGG-160	**3.52**	30.58	56.63	74.87
IJB-C	MS-112-Arc	3.32	**51.26**	**65.49**	**80.59**
IJB-C	VGG-112	3.33	42.74	57.70	73.65
IJB-C	VGG-160	**3.13**	43.42	62.10	76.46
MegaFace	MS-112-Arc	**0.84**	**76.32**	**85.01**	**92.14**
MegaFace	VGG-112	1.20	66.66	77.93	87.55
MegaFace$^{\uparrow}$	VGG-160	0.99	73.09	82.28	90.33

IJB-B/C contain more information than a single image of a face. However, it is quite certain that frames almost always contain less information due to the inherent motion blur or poor quality compared to single images. By naively taking the average of all L_2-normalized features within a template, all features are considered equally valuable in terms of information quality. Thus, poor features extracted from video frames mitigate the performance. Additionally, noise also influences the performance. Regarding the MegaFace benchmark, the refined protocol guarantees that it can be considered relatively clean and thus less noisy than both IJB benchmarks. Hörmann *et al.* proposed two methods to cope with noisy identity labels in templates [7†] and low-quality video frames [6†]. The latter is presented in Chapter 6 together with an overview of other sophisticated methods to better leverage the additional information present in templates.

While the superiority of MS-112-Arc is evident for most FARs, EER indicates no clear preference when minimizing FNs and FPs is deemed equally important. Moreover, correctly verifying 76.32% of all genuine pairs while falsely predicting only a single imposter pair out of 10^6 as genuine demonstrates that FR systems can be employed even in security-sensitive scenarios.

3.6.2.2 Face Identification

Due to their disjoint galleries, IJB-B and IJB-C allow the simultaneous analysis of closed-set and open-set face identification. In both benchmarks, every identity in both galleries is represented by a single template containing solely still images. Like during face verification, the probe sets contain templates comprising still images and video frames.

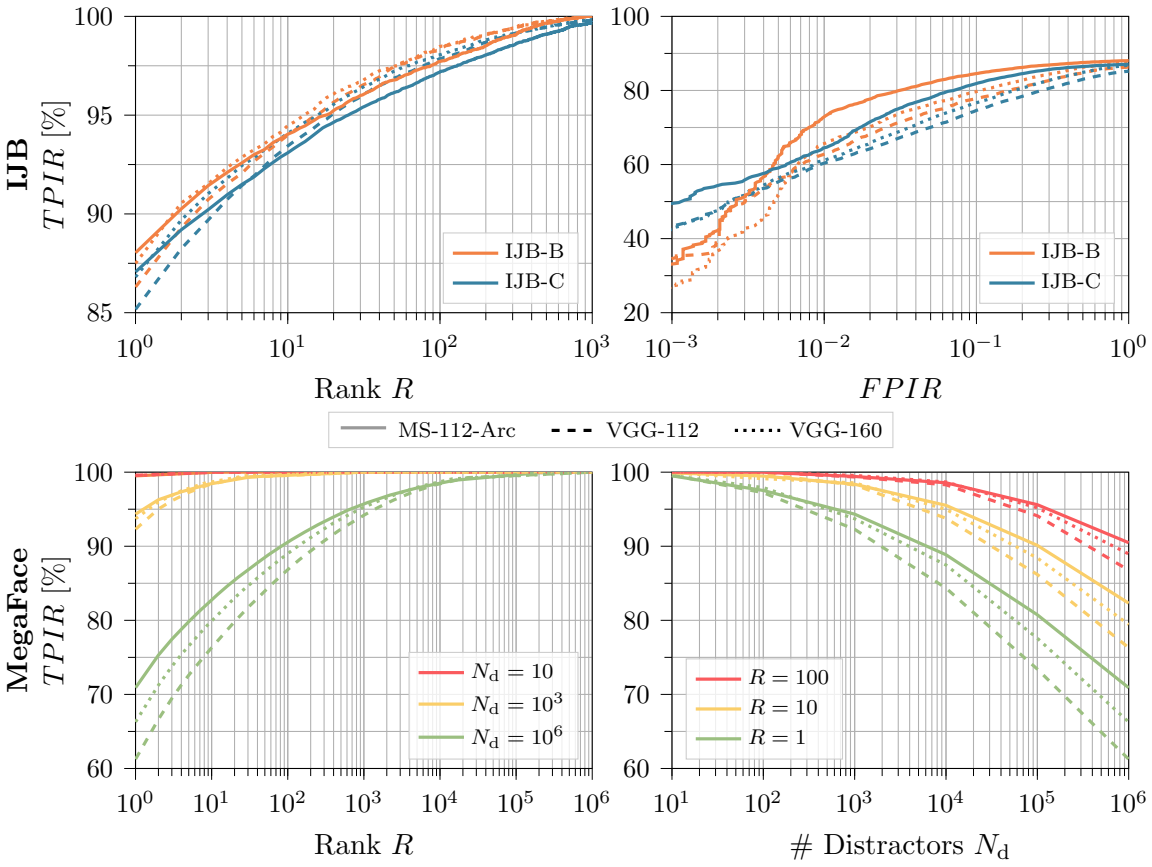

Figure 3.9: CMC and ROC at rank $R = 1$ for selected methods (line style) on IJB-B/C mixed media and MegaFace identification protocols (line color).

Hence, the feature representative for every template is obtained by averaging the L_2-normalized features of every face within the template. The MegaFace benchmark evaluates only closed-set face identification. However, the protocol allows the analysis for varying gallery sizes $|\mathcal{G}| = N_\mathrm{d} + 1$ by adding N_d distractors to the single image representing the identity that is altered throughout the analysis. Then, the face identification performance is evaluated by comparing all remaining probe images of the same identity with the gallery $\mathcal{G}$.

Figure 3.9 and Table 3.9 illustrate the face identification performance on the three datasets. Both CMCs indicate an explicit dependency of $TPIR$ on the rank R. While MS-112-Arc outperforms both methods on MegaFace with $N_\mathrm{d} = 10^6$ distractors, it only achieves a better $TPIR$ for lower ranks R on IJB-B/C. The MegaFace benchmark reveals the apparent dependency of the CMC on the gallery size, as introduced in Section 3.5.2.2. Particularly when considering $TPIR$ for $R = 1$, a considerable difference for distinct N_d can be identified. At the same time, for relatively small galleries of $\approx$ 1k to 2k identities (IJB-B/C and MegaFace with $N_\mathrm{d} = 10^3$), $TPIR$s between the models do not differ a lot – particularly for higher R. For larger galleries $N_\mathrm{d} = 10^6$, the superior performance of MS-112-Arc is more apparent even for higher R. Thus, it is crucial to consider the

Table 3.9: Detailed evaluation of selected methods on IJB-B/C mixed media and MegaFace identification benchmarks. $^{\uparrow}$ denotes that aligned faces with $r = 112$ were upscaled. For MegaFace, $TPIR@R$ was computed for $N_\mathrm{d} = 10^6$ and $TPIR@N_\mathrm{d}$ for $R = 1$. All results are given in %.

		$TPIR@R =$			$TPIR@FPIR =$			$TPIR@N_\mathrm{d} =$		
Dataset	Method	1	10	100	10^{-3}	10^{-2}	10^{-1}	10	10^3	10^6
IJB-B	MS-112-Arc	**88.03**	94.03	97.72	**35.98**	**72.88**	**85.35**			
IJB-B	VGG-112	86.30	93.99	98.31	33.42	62.87	78.53			
IJB-B	VGG-160	87.49	**94.47**	**98.48**	26.93	66.21	80.56			
IJB-C	MS-112-Arc	**87.06**	93.09	97.19	**47.98**	**64.70**	**82.37**			
IJB-C	VGG-112	85.17	93.44	97.85	43.46	60.27	74.95			
IJB-C	VGG-160	86.81	**94.09**	**98.06**	44.20	60.66	77.16			
MegaFace	MS-112-Arc	**70.90**	**82.31**	**90.48**				99.51	**94.33**	**70.90**
MegaFace	VGG-112	61.25	76.31	86.76				**99.59**	92.29	61.25
MegaFace$^{\uparrow}$	VGG-160	66.27	79.45	88.96				99.55	93.71	66.27

gallery size $|\mathcal{G}|$ when estimating the closed-set face identification performance of an FR system.

In terms of open-set FR, the ROC reveals a clear advantage of MS-112-Arc of mostly 5-10% over both other methods. For all methods, a substantial drop in $TPIR$ at $R = 1$ for $FPIR < 1\%$ is noted. When applying this trend to a larger gallery, most FR models could be deemed unusable for open-set face identification tasks. Hence, despite exceptional performance in face verification, open-set face identification offers much room for improvements and is considered one of the hardest challenges in biometrics.

3.6.3 Comparison with the State of the Art

Table 3.10 provides an overview of the FR performance of the most recent methods (*cf.* Section 3.4) on various datasets. It is evident that the algorithms' capabilities have surpassed humans on LFW by a substantial margin for many years.[xvii] Besides the better efficiency in processing images, the clear advantage of algorithms is proven with these results. Furthermore, the results demonstrate that relying solely on humans to label FR datasets is far from optimal, which justifies the employment of either ethnicity-specific annotators familiar with the identities [33] or super-recognizers [171] to minimize label noise.

Table 3.10 clearly illustrates the limitations of LFW as saturation was reached in 2019. Moreover, the authors published a list of seven incorrectly labeled pairs [85], which account for most errors in recent methods. Thus, despite the popularity of the LFW benchmark, one must consider additional benchmarks to reveal the differences in FR performance of multiple models.

[xvii] For a fair comparison, the human performance of tightly cropped images was chosen as it is similar to the input of deep learning algorithms. Kumar *et al.* [113] showed that humans achieve 99.20% accuracy on LFW if the background is also considered.

Table 3.10: Performance comparison of various state-of-the-art approaches and the methods used throughout this work. For the verification datasets, accuracy *Acc* is reported. The IJB mixed-media face verification protocols were employed and $TAR@FAR = 10^{-4}$ is reported. On the MegaFace, ID denotes the $TPIR$ for rank $R = 1$ with $N_\mathrm{d} = 10^6$ distractors and Ver refers to face verification performance described by $TAR@FAR = 10^{-6}$. $^\uparrow$ denotes that aligned faces with $r = 112$ were upscaled. All results are given in %.

		Verification					IJB		MegaFace	
Method	Year	LFW	CPLFW	CFP	CALFW	AgeDB	IJB-B	IJB-C	ID	Ver
Human Performance [113]	2009	97.53								
CenterLoss [229]	2016	99.28	77.48		85.48	90.72			65.23	76.51
SphereFace [131]	2017	99.42	81.40		90.30				72.73	85.56
VGGFace2 [14]	2018	99.43	84		90.57		80	84		
CosFace [215]	2018	99.73							82.72	96.65
ArcFace [33]	2019	99.83	92.08	98.27	95.45	98.15	94.20	95.60	98.35	98.48
CircleLoss [199]	2020	99.73		96.02				93.95	98.50	98.73
GroupFace [108]	2020	**99.85**	93.17	98.63	**96.20**	98.28	94.93	96.26	98.74	98.79
CurricularFace [87]	2020	99.80	93.13	98.37	**96.20**	98.32	94.80	96.10	98.71	98.64
BroadFace [109]	2020	**99.85**	93.17	98.63	**96.20**	98.38	94.97	96.38	98.70	98.95
MagFace [148]	2021	99.83	92.87	98.46	96.15	98.17	94.51	95.97		
Face Transformer [281]	2021	99.80	93.08	96.77	96.18	98.05		96.31		
ArcFace-VPL [34]	2021	99.83	**93.45**	**99.11**	96.12	**98.60**	**95.56**	**96.76**	**98.80**	**98.97**
MS-112-Arc		99.53	86.50	92.93	94.22	96.13	82.52	80.59	87.85	76.32
VGG-112		99.43	85.70	94.76	87.95	89.63	71.76	73.65	61.25	66.66
VGG-160		99.42	87.40	96.37	88.97	91.70$^\uparrow$	74.87	76.46	66.27$^\uparrow$	73.09$^\uparrow$

On datasets besides LFW, the differences between methods and the substantial improvements in the last years become apparent. Multiple methods demonstrate different directions of improving the baseline provided by ArcFace [33]. Most notably, buffering previous feature vectors to allow the model to consider identities outside the batch as in BroadFace [109] and using variational class-wise prototypes instead of a single prototype vector as in ArcFace-VPL [34] shows the best performance boost compared to ArcFace. Overall, ArcFace-VPL provides the best results and demonstrates that it can be confidentially employed in adverse scenarios with large head pose variations (CPLFW and CFP), age gaps (CALFW and AgeDB) or even security-sensitive applications since the reported $TAR \approx 99\%$ for $FAR = 10^{-6}$ on MegaFace indicates that it would still yield a satisfying TAR even for a lower FAR.

The analysis further reveals that the selected methods (VGG-112, VGG-160 and MS-112-Arc) used in Chapters 4 to 6 cannot match the performance of current methods. This can be partially attributed to the employment of deeper ResNets (see Table 3.3) and the usage of more powerful albeit costly GPUs [33, 34, 87, 108, 109], allowing the training without GAP in a reasonable time. Nevertheless, these selected methods form a valid baseline for Chapter 5 and suffice as a feature extractor for Chapters 4 and 6.

4

A Coarse-to-Fine Dual Attention Network for Blind Face Completion

This chapter introduces a novel approach for blind face completion. Face completion constitutes a subdomain of image inpainting, in which occluded pixels are reconstructed. Formally, this can be written as $\boldsymbol{I}_{\mathrm{o}} \mapsto \hat{\boldsymbol{I}}$ with $\boldsymbol{I}_{\mathrm{o}}$ denoting the occluded image and $\hat{\boldsymbol{I}}$ the reconstructed image, which is optimized to resemble the non-occluded ground-truth image $\boldsymbol{I}_{\mathrm{gt}}$.[i] While the focus in classical image inpainting lies exclusively on achieving a realistic prediction of the occluded pixels, face completion must also ensure that identity features are semantically coherent within the reconstructed face $\hat{\boldsymbol{I}}$. If there is a mismatch between identity features extracted from the occluded and non-occluded areas, the feature extractor does not know which areas to rely on, resulting in an ambiguous feature vector $\boldsymbol{f}$. This can even lead to worse performance than if identity features are extracted from the occluded image $\boldsymbol{I}_{\mathrm{o}}$. Moreover, special attention must be given to guarantee realism in terms of faces, such as consistent eye colors and makeup, as even recent methods struggle to deliver satisfying results [260, 284]. Hence, face completion has the extra complexity of considering identity features and matching face-specific realism criteria compared to image inpainting.

A selection of occluded faces from various face recognition (FR) datasets are depicted in Figure 4.1. Some occlusions, *e.g.*, sunglasses, medical face masks, microphones, body

Figure 4.1: Examples of occluded faces ranging from natural to synthetic occlusions.

[i]The ground-truth image corresponds to the aligned and augmented image $\boldsymbol{I}_{\mathrm{aug}}$.

parts, *etc.*, occur naturally, whereas others, such as copyright protection, subtitles, album or magazine covers, *etc.*, are introduced after the photo was taken. Besides, a part of a face can be cut off if encountered at the image's borders. In the case of natural occlusion, no pairwise data ($\boldsymbol{I}_{\mathrm{o}}$ and $\boldsymbol{I}_{\mathrm{gt}}$), *i.e.*, the exact face with and without occlusion, is available. Therefore, standard approaches based on the availability of pairwise data cannot be employed.

One approach to compensate the lack of natural pairwise data constitutes synthetically generating them. Multiple datasets [228, 257] have been published, which motivated more comprehensive works [138, 250] on medical face mask removal. *E.g.*, Yuan *et al.* [257] synthesized the faces by adding medical face masks, sunglasses, scarves, *etc.*, according to the facial landmarks. Even though Yin and Di [250] only considered medical face masks, they showed that their approach can handle a wide range of medical face masks. Thus, the crucial aspect when coping with natural occlusions is recreating them synthetically to be as realistic as possible, which is very challenging.

Synthetic occlusions, as depicted in Figure 4.1, are reproduced relatively effortlessly via data augmentation (see Section 3.5.1). Furthermore, an exact implementation of specific occlusion patterns is not required since the network can generalize well as long as it is provided with occlusions varying in shape, color, size, and position. Formally, the synthetically occluded image $\boldsymbol{I}_{\mathrm{o}}$ with a spatial resolution $r \times r$ can be written as

$$\boldsymbol{I}_{\mathrm{o}} = (\mathbb{1}_{r,r} - \boldsymbol{M}_{\mathrm{gt}}) \odot \boldsymbol{I}_{\mathrm{gt}} + \boldsymbol{M}_{\mathrm{gt}} \odot \boldsymbol{c}, \tag{4.1}$$

where $\odot$ denotes the Hadamard product, and $\boldsymbol{M}_{\mathrm{gt}}$ and $\boldsymbol{c}$ are the ground-truth mask and the mask's color vector, respectively. The mask $\boldsymbol{M}_{\mathrm{gt}}$ is a binary $r \times r$ matrix with a value of 1 describes an occluded area. In case of synthetically occluded images $\boldsymbol{I}_{\mathrm{o}}$ as created with Equation (4.1), $\boldsymbol{I}_{\mathrm{o}}$ is also referred to as masked image.

While the availability of pairwise data eases training, automatic processing of synthetically occluded faces is only possible if $\boldsymbol{M}_{\mathrm{gt}}$ is not an input of the network but rather predicted by it. This case is referred to as *blind* face completion. By definition, blind face completion approaches also include the case where the mask $\boldsymbol{M}_{\mathrm{gt}}$ is provided. Hence, it can also cope with natural occlusions given a manually annotated mask encompassing the unwanted regions. On the other hand, non-blind face completion approaches require the meticulous annotation of a mask if $\boldsymbol{M}_{\mathrm{gt}}$ is unavailable. This is not only time-consuming but also prone to cause errors. Both cases, too small or too large masks, yield undesired results due to either untouched occluded areas or alterations of non-occluded areas. Thus, blind face completion constitutes a clear advantage as it dispenses with the need for tedious mask annotations, given that the network can reliably detect the mask $\boldsymbol{M}_{\mathrm{gt}}$. Nevertheless, special attention must be given to avoid errors in the mask detection as the thereby introduced unwanted artifacts may distort the reconstructed image $\hat{\boldsymbol{I}}$.

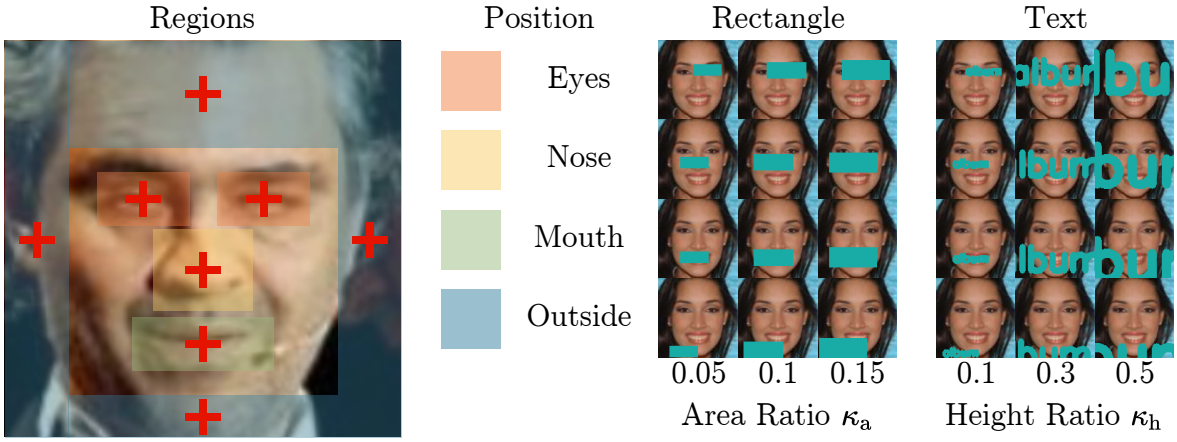

Figure 4.2: Position of the occlusions defined by regions and their respective centers (left) and the influence of shape, position and size on the occlusion (right).

From the FR systems' point of view, prior face completion is not essential as long as the FR system is robust against occlusions. As analyzed by multiple researchers [60, 145, 281, 13†], FR systems are vulnerable to occlusions – mainly if both eyes are affected [60]. To alleviate the effect of occlusions on the FR performance, Mathai *et al.* [145] and Hörmann *et al.* [13†] demonstrated that prior face completion boosts FR performance substantially. Considering this result, the main objectives of the blind face competition approach are: 1) a realistic reconstruction; and 2) mitigating the drop in FR performance. Since both objectives are not always aligned, this chapter aims at a balanced approach for blind face completion for synthetic occlusions with the option to shift the focus to either objective. Besides, the blind FR approach should be able to handle occlusions varying in form, position, size and color.

The approach, experiments, and results presented in this work are, in part, pre-published in [13†] and are referenced throughout the chapter. After describing how to generate synthetic occlusions in Section 4.1, Section 4.2 analyzes the impact of occlusion on the FR performance and thereby motivates the consideration of occluded faces. In Section 4.3, the related work on image inpainting with emphasis on face completion is presented. The architecture and the proposed loss functions are described in Sections 4.4 and 4.5, respectively, followed by the training strategy and evaluation metrics in Section 4.6. Section 4.7 reports the quantitative and qualitative results of the blind face completion approach.

4.1 Generating Synthetic Occlusions

The approach presented in this chapter aims to remove synthetic occlusions similar to the occlusions in Figure 4.1. Therefore, it is necessary to encompass occlusions, varying in shape, size, position, and color, to ensure the generalization to unknown occlusions. Generating suitable occlusions is crucial for satisfying results despite its apparent simplicity compared to the complex neural network design and training.

Table 4.1: Parameters used to define occlusions.

	Shape	
Parameter	Rectangle	Text
Form	aspect ratio $\sim \mathcal{U}_1(0.5, 2)$	2048 words
Size	area ratio κ_a	height ratio κ_h
Position	eyes, nose, mouth, and outside with the occlusions' center $\boldsymbol{c}_\mathrm{o} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$	
Color	uniform color $\boldsymbol{c}$	

There are multiple options to generate masks. Previous works in image inpainting removed random rectangular regions [89, 160, 202, 254, 284]. More complex free-form masks were created by Liu *et al.* [127] based on point trajectories in video frames and published as a mask dataset, which is widely used in image inpainting works [89, 160, 256, 260, 261, 286]. Various works [20, 170, 249] manipulated real object shape templates obtained from object segmentation masks. Besides, Yu *et al.* [255] created free-form masks by simulating random drawing accompanied by repeatedly changing the angle and Suvorov *et al.* [202] used polygonal chains, which were dilated with random width.

This work employs a more straightforward approach following [13†], in which two geometric shapes are considered: 1) rectangular occlusions representing uniform occlusions concentrated in a small area; and 2) occlusions generated from words, which cover the entire face and contain holes. Thus, text occlusions resemble the free-form masks in related works [20, 89, 160, 170, 202, 249, 255, 256, 260, 261, 286]. Utilizing two different shapes not only aids the network in generalizing but also enables a more differentiated and meaningful evaluation.

For rectangular occlusions, their form is determined randomly by selecting an aspect ratio $\sim \mathcal{U}_1(0.5, 2)$. To generate text occlusion, words are chosen from a list of 2048 mnemonic English words [192], with a mean length of 5.4 letters (minimum 3 and maximum 8). In this way, the forms of the occlusions cover a wide range of variations.

To create a synthetic occlusion of a specific size, which is defined by its mask $\boldsymbol{M}_\mathrm{gt}$, an area ratio κ_a is introduced as

$$\kappa_\mathrm{a} = \frac{1}{r^2} \sum_{\forall\, x,y} \left[\boldsymbol{M}_\mathrm{gt}\right]_{x,y}. \tag{4.2}$$

While κ_a is suitable to define the size of a rectangular occlusion, it is rather difficult to generate a text occlusion for a given κ_a since κ_a depends on the number of letters and the letters themselves. Hence, the height ratio $\kappa_\mathrm{h} = H_\mathrm{txt}/r$ is utilized for text occlusions, where H_txt is the height of the text, since κ_h is independent of the word length.

To investigate the influence of occlusions covering different face parts, the center of the occlusion $\boldsymbol{c}_\mathrm{o} = \begin{pmatrix} x_\mathrm{o} & y_\mathrm{o} \end{pmatrix}^\mathsf{T}$ is placed within one of the following four regions: 1) eyes; 2) mouth; 3) nose; and 4) outside, where two regions, eyes and outside, are further divided to encompass all cases. All regions with their respective centers $\boldsymbol{\mu}_i$ are depicted in Figure 4.2 (left). Then, the center of every occlusion $\boldsymbol{c}_\mathrm{o}$ is sampled from a Gaussian normal distribution $\boldsymbol{c}_\mathrm{o} \sim \mathcal{N}_2(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ with a region-specific center coordinate $\boldsymbol{\mu}_i$ and

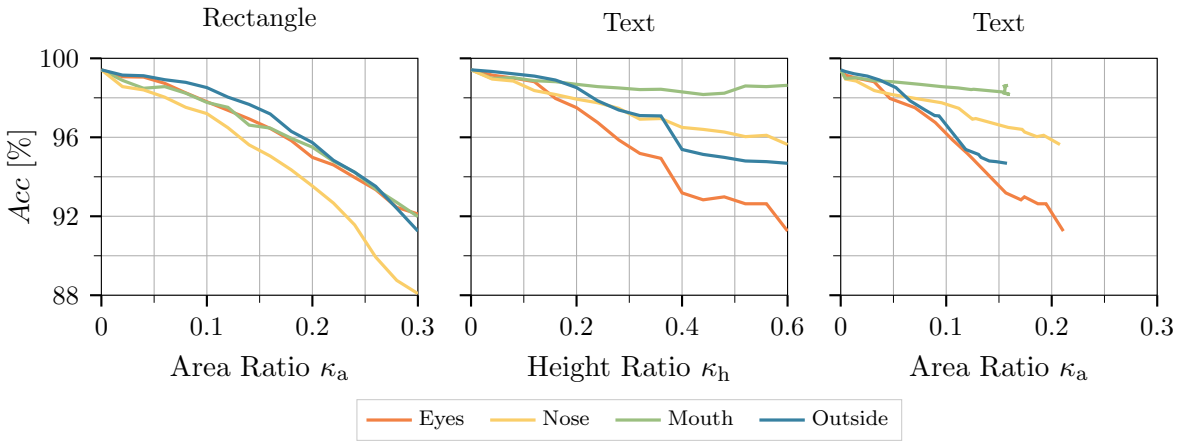

Figure 4.3: Face verification accuracy *Acc* of the VGG-112 model on the LFW dataset dependent on the occlusions' size. Different shapes and positions are considered.

standard deviation $\boldsymbol{\sigma}_i$ to recreate the form of the region. By truncating $\boldsymbol{c}_o$, the center point always remains within its respective region. Besides, since all images $\boldsymbol{I}_{gt}$ are aligned (see Section 3.2), eyes, mouth, and nose lie roughly at the same position for all images, which justifies the definition of dataset-wide regions.

Lastly, the occlusions' colors are chosen randomly within the color space and encoded in the color vector $\boldsymbol{c}$ (see Equation (4.1)). All masks are colored uniformly, *i.e.*, no color gradient or color variations within the masks are permitted. While this restricts the generalization, allowing non-uniform masks complicates the distinction of valid occlusions from the background.

Table 4.1 summarizes all parameters describing an occlusion. During training, every parameter is selected randomly within certain constraints. In this way, the network is exposed to a large variation of occlusions, which improves generalization. For evaluation, the randomness is constrained yet ensured to be deterministic by using a seed. Thus, if the size or region is changed for rectangular occlusions, all other parameters are untouched, guaranteeing a meaningful analysis. Figure 4.2 (right) depicts an excerpt of the vast variations possible utilizing this augmentation scheme.

4.2 Preliminary Analysis: The Impact of Occlusions on Face Recognition

Intuitively, restricting the information within a face mitigates the performance of FR systems, which was also shown by [60, 145, 281, 13†]. Thus, even subtle occlusions, as in Figure 4.1, are expected to introduce classification errors and require an in-depth analysis. Figure 4.3 illustrates the vulnerability of FR systems to occlusions on the example of the FR model *VGG-112*, which is a ResNet-v2 of depth $L = 50$ trained on the VGGFace2 dataset with the softmax cross-entropy (CE) loss as introduced in

Section 3.6.2. The analysis is performed on the Labeled Faces in the Wild (LFW) face verification dataset. Both images of every face verification pair are occluded similarly following Section 4.1, *i.e.*, shape, size, and region are identical for both faces, whereas form and color are distinct. Note that an identical region means that the position within the region still varies between both images, which may lead to an occluded left eye for one image and an occluded right eye for the other image of a pair.

For rectangular occlusions covering 30% of the image area, a drop in face verification accuracy (*Acc*) of $8-12\%$ is observed. The most significant drop occurs when occluding the nose region of a face. This is expected as the nose constitutes the most central region, which often involves the occlusion of the critical eye and mouth regions if the occlusion is sufficiently large. Moreover, occluding areas outside the face only slightly affects the *Acc*.

The *Acc* of faces tampered with text occlusion also clearly depends on the occlusions' size. However, the effect is less explicit than for rectangular occlusions, which is mainly due to the letters' arbitrary form. The eyes region is particularly susceptible to text occlusion as text occlusions are wider than tall, resulting in a large probability of both eyes being occluded. On the other hand, rectangular occlusions indicate that the occlusion of only one eye is compensated well by extracting the information from the other eye. This finding is in accordance with the analysis by Grm *et al.* [60].

Computing the area ratio κ_a for text occlusions (*cf.* Figure 4.3 (right)) reveals that κ_a highly depends on the region and barely exceeds 0.15 even for $\kappa_\mathrm{h} = 0.6$, which clearly hints toward border effects due to wide occlusions. Thus, text occlusions centered in the mouth region only slightly affect *Acc* even for large height ratios κ_h as large parts of the occlusion lie outside the image boundaries allowing the network to still extract sufficient information from eye or nose regions. For $\kappa_\mathrm{a} = 0.1$, rectangular occlusions yield superior *Acc* except for the mouth region. This suggests that feature vectors are particularly distorted if occlusions contain holes and thereby cover a larger part of the face. Since κ_a for text occlusions is afflicted by large standard deviations and varies largely among the regions, text occlusion is evaluated dependent on κ_h for the remaining part of this chapter.

In conclusion, this analysis highlights the impact of the occlusions' position, shape, and size on the FR performance. To reduce this apparent dependency, it is necessary to develop methods to cope with occlusions.

4.3 Related Work in Image Inpainting

Even though there exist some image inpainting methods [28, 77, 123, 127, 133, 139, 267], which utilize a convolutional neural network (CNN) without incorporating it in a generative adversarial network (GAN), the vast majority [20, 31, 66, 80, 89, 90, 93, 121, 140, 160, 196, 202, 216, 224, 225, 244, 247, 249, 254, 255, 260, 261, 269, 274, 284] use a GAN. This is mainly due to the GANs' unique ability to create astonishingly realistic reconstructions of the occluded areas and to obtain an overall semantically coherent image.

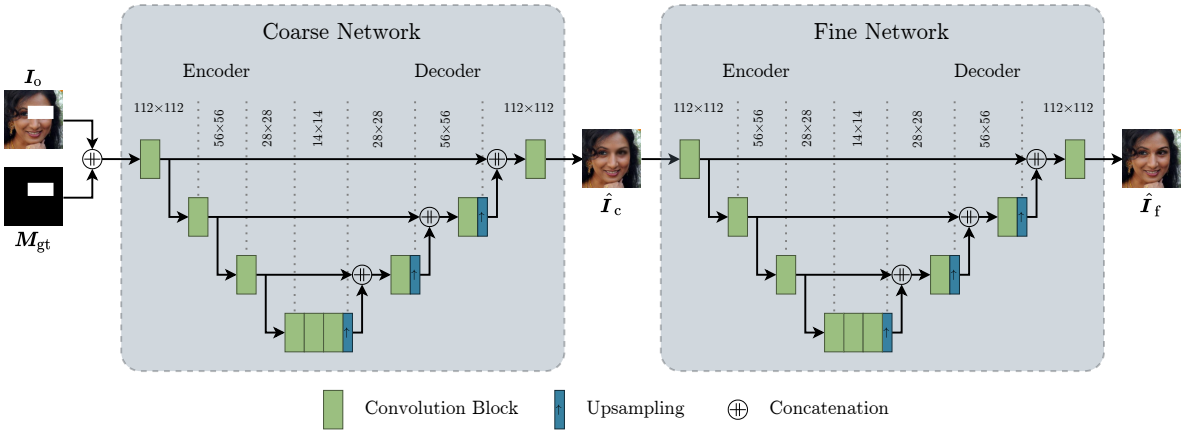

Figure 4.4: Example of a typical coarse-to-fine architecture used for image inpainting. Each network comprises a U-Net [179], which is characterized by downsampling the input to the desired resolution followed by an upsampling, resulting in the U-like shape. Additional skip connections allow the incorporation of low-level features from the encoder in the decoder.

Multiple approaches [28, 123, 139, 182, 249, 254, 255, 286] employ a so-called coarse-to-fine structure, in which the first generator outputs a rough estimation of the input, which is then refined by the subsequent generator (*cf*. Figure 4.4). In their analysis, Sagong *et al*. [182] demonstrated that using a coarse network as a prior provides superior results compared to exclusively training the refinement network. A similar way of providing the network with a prior is proposed by Nazeri *et al*. [160], in which they first predict a reconstructed edge map, which is then used together with the masked face for refinement. Likewise, Xiong *et al*. [241] proposed to include the reconstructed contours as the input of the coarse and the fine reconstruction networks. As opposed to all approaches above, Guo *et al*. [64] proposed a coupled network to process two inputs parallelly yet collaboratively, predicting a structure and a texture, which are then fused to obtain a compelling reconstruction.

A variant of the U-Net architecture [179], depicted in Figure 4.4, constitutes the most popular architecture in image inpainting [28, 77, 80, 90, 127, 133, 139, 196, 244, 249, 269, 284] even if no coarse-to-fine structure is employed. While most approaches utilize skip connections to alleviate learning identity transformations for the non-occluded regions some researchers [145, 202, 254, 255] proposed to dispense with them or employ ResNet-like units to mimic their behavior on a more local level [202].

Besides the adversarial loss, the network is typically guided by pixel-wise similarity distance losses between the prediction $\hat{\boldsymbol{I}}$ and the ground-truth image $\boldsymbol{I}_{\text{gt}}$ [20, 28, 39, 64, 77, 80, 121–123, 127, 133, 138, 139, 160, 216, 237, 246, 249, 254, 255, 261, 267, 286] to obtain a rough reconstruction by the coarse network. Johnson *et al*. [99] proposed to measure similarity in the feature space by computing the distance between feature maps extracted by a network pretrained on image classification. In this way, this so-called perceptual loss compares high-level features instead of pixel values. Similarly, the style loss [99] punishes the difference in the correlation between activations in feature maps.

Both losses were successfully adapted to image inpainting and are frequently utilized in conjunction [64, 80, 127, 138–140, 160, 216, 237, 246, 261, 286].

One research direction in image inpainting addresses the unreliable information within the input pixels in every convolutional layer. This direction was initiated by Liu *et al.* [127], who proposed partial convolutions, in which the input of a convolutional layer is masked such that the convolution only considers input values originating from the non-occluded areas. The binary mask is then expanded successively after every convolution according to the convolution's receptive field. This concept was enhanced by Yu *et al.* [255] with gated convolutions. Unlike the binary masks in partial convolutions, every gated convolution learns its own soft mask for every channel with values between 0 and 1 by performing a separate convolution on the input followed by a sigmoid function. This soft attention mask is applied to the output via an element-wise multiplication. Yi *et al.* [249] proposed lightweight gated convolutions with multiple modifications to substantially reduce the number of parameters during the mask prediction.

In contrast, region-wise convolutions [139] operate with two distinct weights (and biases) depending on whether the input lies within the occluded or non-occluded area. Hukkelås *et al.* [90] proposed imputed convolutions, which substitute the value of uncertain input pixels by a weighted average over spatially close features. With mask-aware dynamic filtering, Zhu *et al.* [286] created kernels of a convolutional layer adaptively based on a mask. In contrast to the previous methods, which focus on altering the convolution operation, Yu *et al.* [256] introduced a region normalization layer, which normalizes occluded and non-occluded regions with different means and variances. Hong *et al.* [80] utilized a fusion block on multiple feature maps of the decoder to generate an attention map, which makes the network focus on the occluded pixels during reconstruction.

4.3.1 Attention Blocks for Image Inpainting

While manipulating the convolutional layers to focus on reliable information [90, 127, 249, 255, 286] via an attention map is considered a form of attention mechanism, the attention blocks introduced in this section aim to exchange information at larger scales. This is crucial as a realistic reconstruction is obtained only if the network successfully captures the environment in which the picture was taken. Thus, the network needs to extract the underlying context to reconstruct the area with fitting elements. For face completion, the network can only maintain a coherent makeup if it is aware of global information. Hence, it is essential to provide the network with the tools to leverage information, even from more distant pixels, *i.e.*, a large receptive field is required.

One straightforward method to increase the receptive field is by employing convolutional layers with large 7×7 kernels [225]. However, it is necessary to employ them in deeper layers with a large number of feature maps C, which inevitably leads to a massive increase of trainable parameters. As introduced in Section 2.5.1, the receptive field can also be increased substantially by dilated convolutions with a dilation factor $D > 1$ [31, 93, 160, 254, 260, 261]. Despite the substantial increase of the receptive field without additional parameters, dilated convolutions only provide sparse activations and are prone to generate gridding artifacts [165].

To cope with sparse activations, Hui *et al.* [89] proposed the dense multi-scale fusion block (DMFB), which combines multi-scale information extracted from four parallel convolutional layers with different dilation factors D. Later, Zeng *et al.* [261] proposed the aggregated contextual-transformation block, which is a simplification of the DMFB as it does not employ the hierarchical feature fusion. Even though the DMFB provides a considerably large receptive field of 21 and accounts for the inherent sparsity of dilated convolutions, it is biased towards the center. In contrast, attention blocks are desired to truly incorporate global information while considering the activation of every pixel equally and thus independently of its distance.

The patch-swap block proposed by Song *et al.* [196] substitutes feature patches within the occluded area with similar patches from the non-occluded area. The authors use feature maps extracted from a coarse reconstruction by a pretrained image classification network as the input of the patch-swap block. With Shift-Net, Yan *et al.* [244] incorporated a shift-connection layer, which enhances the skip connections in the U-Net by concatenating the encoder and decoder feature maps with a pixel-wise replacement of all features in the decoder with their nearest neighbors in the encoder. Similarly, Zheng *et al.* [274] restricted the pixel-wise attention from Yan *et al.* [244] to only allow the replacement of occluded features.

A comparison with patches of size 3×3 is performed by Yu *et al.* [254]. Analogous to the patch-swap block [196], the authors replaced patches from the occluded area with patches from the non-occluded area. Moreover, they used attention propagation to encourage coherency by smoothing the patch similarity scores before patch replacement. In order to not distort the patch features during the normalization when computing their similarity, Sagong *et al.* [182] proposed to use euclidean distance instead of cosine similarity as in [254]. In accordance with Yu *et al.* [254], Zeng *et al.* [260] calculated patch-wise similarity between occluded and non-occluded decoder feature patches; however, they utilized features patches from the encoder for reconstruction. In addition, the output of the attention block is further refined by four parallel dilated convolutions and the attention block is employed at multiple resolutions. Likewise, Guo *et al.* [64] utilized global patch-wise attention followed by parallel dilated convolutions. Then, the output of every branch was fused after multiplying it with an attention map generated from the output of the attention layer. Yi *et al.* [249] extended image inpainting to ultra-high resolution images of 8k pixels per side. While the patch similarity is computed only once, they substitute patches at multiple resolutions in the decoder. To account for the varying resolutions at which the attention transfer is performed, the resolution of the patches for reconstruction is modified to cover the whole feature map.

The dual spatial attention module, proposed by Zhou *et al.* [284], comprises two pixel-wise attention mechanisms: 1) self-attention restricted to the occluded area; and 2) cross-attention to replace features from the occluded areas with features from the non-occluded area. In contrast to all aforementioned methods, the authors adapted the structure from the pioneering work in self-attention [264] by utilizing three 1×1 convolutional layers to split the input into three different branches for matching and reconstruction.

Recently, Suvorov *et al.* [202] proposed a very different approach by incorporating fast Fourier convolution [21], *i.e.*, transforming the feature maps into the frequency domain by a fast Fourier transform and applying a convolution before transforming back into the feature domain. In this way, global information is leveraged without adding a huge number of parameters. The authors show impressive results particularly – and unsurprisingly – on periodic structures, *e.g.*, bricks, fences, windows, *etc.*

4.3.2 Face Completion

Most approaches introduced in Section 4.3 were trained and evaluated on faces, thus performing face completion. However, since face completion (or face inpainting) requires the method to guarantee coherent face identity features apart from realism, face completion is considered one of the most challenging image inpainting tasks. This involves maintaining the rich information and implicit relationships between multiple face parts also for the reconstructed areas. Therefore, face-specific algorithms are proposed focusing on improving image inpainting tasks.

One research direction is characterized by leveraging face-specific priors during the reconstruction. Li *et al.* [122] supervised the reconstruction with an additional loss determined by a semantic parsing network. In contrast, Song *et al.* [194] predicted the semantic parsing map jointly with facial landmarks prior to the reconstruction. Similar to Song *et al.* [194], Yang *et al.* [246] guided the inpainting with facial landmarks, whereas Yin and Di [250] first predicted the mask and the 3D reconstructed face prior to the reconstruction of medical face masks. Dey and Boddeti [39] disentangled the face into geometric and photometric factors followed by an iterative inpainting algorithm.

While previous approaches provided the output of other face-specific algorithms at the input of the reconstruction network, Zhang *et al.* [269] proposed to embed information about the face regions and facial landmarks into latent variables. These latent variables are concatenated at the input of the decoder and thus provide valuable guidance in the reconstruction. Using a siamese network structure, Ma *et al.* [138] leveraged features from dense field estimation and employed a dual attention fusion module for medical face mask removal. A more complex framework is proposed by Wu *et al.* [237], which extracts features from a coarse reconstruction and stores them in a memory grouped according to a semantic parse map. Then, the most relevant features are injected into the refinement network to produce a highly realistic reconstruction.

In contrast to approaches mentioned earlier, Chen *et al.* [20] do not incorporate any additional face-specific algorithms. They proposed to increase the resolution of the network progressively in order to consolidate information across multiple scales and thereby reconstruct even high-quality faces. Li *et al.* [121] leveraged the faces' symmetry by a symmetry-consistent CNN, which transfers brightness-adjusted information from one half of the face to the other. If pixels are occluded in both halves, the remaining pixels are reconstructed in a separate network. Besides the DMFB, Hui *et al.* [89] proposed a self-guided regression loss, which focuses on the pixels in the feature map loss that cause a discrepancy in the image space, and a loss punishing spatial misalignment of the feature maps. In addition to the dual attention block, Zhou *et al.* [284] proposed

an oracle supervision signal to the attention blocks to ensure that the attention scores are reasonable. Moreover, seven discriminators focus on various face regions, ensuring a realistic reconstruction.

Entirely distinct from all previous approaches, image inpainting can also be performed in the latent feature space of a generative model (see also Figure 2.5). In this case, a randomly initialized latent vector is updated iteratively until the generated face matches the synthetically occluded face [1, 247].

4.3.3 Blind Image Inpainting

While the mask $\boldsymbol{M}_{\text{gt}}$ is provided in default image inpainting tasks, the occluded areas are unknown in blind image inpainting, increasing the task's difficulty substantially. Thus, blind image inpainting approaches differ from previous methods on image inpainting or face completion (see Sections 4.3 and 4.3.2) by how the network is trained to detect occlusions.

Zhang *et al.* [267] investigated faces that were affected by random meshes for identity protection. Due to the relatively thin periodic patterns, they obtained satisfying reconstructions without utilizing a GAN nor additional mask prediction. Liu *et al.* [133] proposed a residual learning approach, *i.e.*, the network estimates only the pixels affected by occlusion. Moreover, the gradients of the image are used to predict the structure in the occluded regions. With the blind visual motif removal (BVMR) model, Hertz *et al.* [77] employed a U-Net architecture with three decoder branches, which predict the reconstructed image, mask, and motif. They found that learning the motif and partially sharing weights between the decoder branches for image and motif improves the reconstruction quality. Cun *et al.* [28] extended the BVMR model by adding a subsequent refinement network and perceptual loss.

In contrast to [28, 77], Liang *et al.* [123] employed two decoding branches in the coarse network, in which mask predictions at different levels are leveraged in the reconstruction branch. Besides, additional skip connections between the coarse network's decoder and the refinement network's encoder are used. The visual consistency network by Wang *et al.* [224] disentangles mask prediction and inpainting and trains both in an adversarial manner. The mask prediction network predicts visually inconsistent areas within the face, while the reconstruction network uses the mask at multiple depths to reconstruct only the inconsistent areas. In order to cope with masks of different patterns and colors, Wang *et al.* [216] proposed a frequency-guided transformer together with a top-down refinement network. First, they detect the mask with a vision transformer [281] and include high-frequency information of the masked image, which was extracted leveraging a discrete cosine transform. Then, the refinement network utilizes the predicted mask and facial landmarks as a prior for hierarchically restoring semantically consistent features. Additionally, Yin and Di [250] predicted medical face masks and the 3D reconstructed face of the masked face. Then the mask is substituted with noise before passing it to the reconstruction network. In contrast to Yin and Di [250], Ma *et al.* [138] proposed an approach for the blind removal of medical face masks without any intermediate mask prediction.

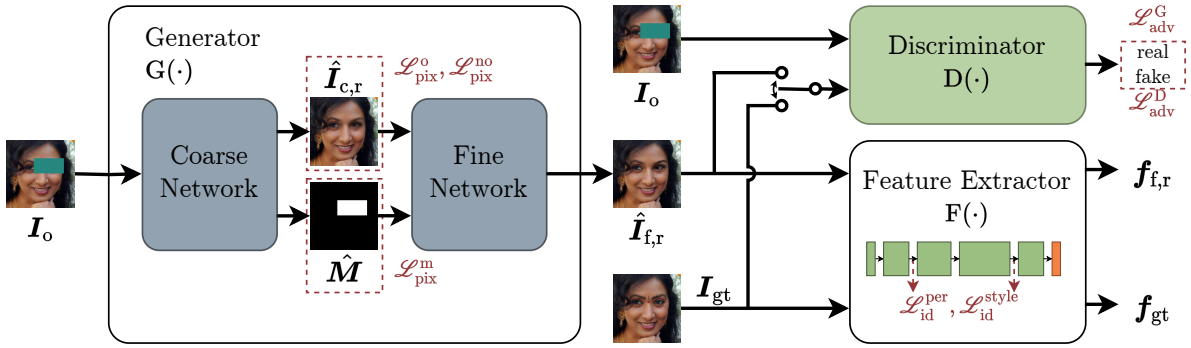

Figure 4.5: Overview of the blind face completion network. The occluded face $\boldsymbol{I}_{o}$ is reconstructed by a coarse-to-fine generator G(·), which additional outputs the mask $\hat{\boldsymbol{M}}$. First, a rough reconstructed image $\hat{\boldsymbol{I}}_{c,r}$ is obtained by the coarse network, guided by pixel-wise similarity losses $\mathscr{L}_{pix}$. Then, $\hat{\boldsymbol{I}}_{c,r}$ is further refined by the fine network yielding $\hat{\boldsymbol{I}}_{f,r}$. A highly realistic reconstruction is ensured by leveraging a discriminator D(·) and adversarial loss $\mathscr{L}_{adv}$. Furthermore, identity losses $\mathscr{L}_{id}$ guarantee similarity in the identity feature space using a pretrained face feature extractor F(·). Adapted from [13†].

4.4 Architecture

Reconstructing faces occluded by masks, which vary in shape, form, size, position, and color, is a highly complex task requiring the synergy of multiple components supervised by precisely designed loss functions as depicted in Figure 4.5. Following the majority of image inpainting works [28, 123, 139, 182, 249, 254, 255, 286], the central component of the proposed approach for blind face completion is the coarse-to-fine generator G(·), which divides the complex reconstruction into two steps with distinct objectives. First, the coarse generator predicts the mask $\hat{\boldsymbol{M}}$ and outputs a rough reconstructed image $\hat{\boldsymbol{I}}_{c,r}$. By only incorporating pixel-wise similarity losses for the generation of $\hat{\boldsymbol{I}}_{c,r}$, the focus lies on a coarse estimation without high-frequency details. In this way, the subsequent fine network solely focuses on refining the coarse reconstruction $\hat{\boldsymbol{I}}_{c,r}$ yielding $\hat{\boldsymbol{I}}_{f,r}$. With the help of two parallel attention blocks, the fine network leverages long-range pixel-wise relationships within a face and substitutes patches of occluded areas with similar patches from non-occluded areas. Distinct loss functions guide the reconstruction by focusing on two criteria: 1) a highly realistic reconstruction $\hat{\boldsymbol{I}}_{f,r}$, which is guaranteed by the discriminator D(·) and by training the coarse-to-fine generator G(·) as a GAN; and 2) coherent identity features within $\hat{\boldsymbol{I}}_{f,r}$, which are obtained by incorporating identity losses $\mathscr{L}_{id}$ based on feature maps extracted from a pretrained face feature extractor F(·).

The following subsections first introduce four advanced blocks, among which the latter three blocks play an essential role in accomplishing large receptive fields and exchanging global information during the reconstruction. Then, every component in Figure 4.5 is described in-depth, followed by the definition of the loss functions.

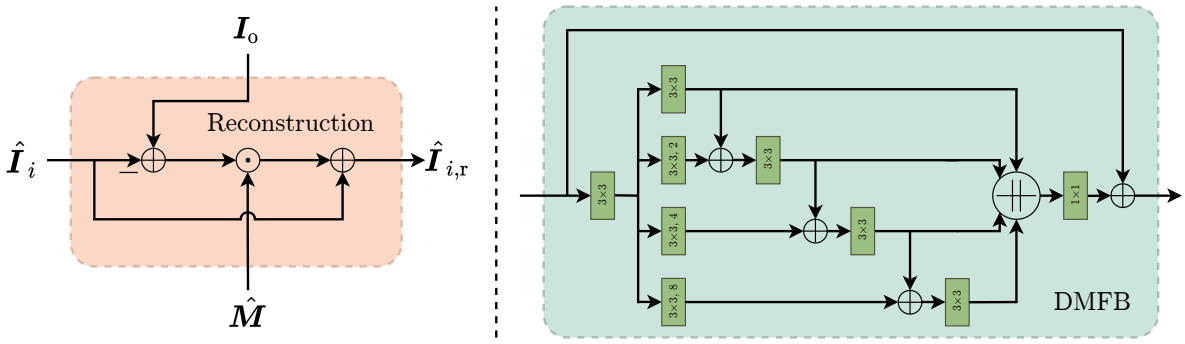

Figure 4.6: The reconstruction block (left) ensures that in the output $\hat{\boldsymbol{I}}_{i,\mathrm{r}}$, only occluded pixels indicated by $\hat{\boldsymbol{M}}$ are altered. With the dense multi-scale fusion block (DMFB) (right), multi-scale features are extracted using parallel 3×3 convolutional layers (green) with different dilation factors $D \in \{1, 2, 4, 8\}$ and hierarchically fused by a concatenation $\oplus$ together with a 1×1 convolutional layer. Adapted from [89].

4.4.1 Advanced Blocks

4.4.1.1 Reconstruction Block

Ideally, the generator $\mathrm{G}(\boldsymbol{I}_\mathrm{o})$ only alters the value of pixels within the occluded area of $\boldsymbol{I}_\mathrm{o}$, whereas non-occluded pixels are left untouched. Thus, similar to [77, 13†], the predicted mask $\hat{\boldsymbol{M}}$ can be used to limit the reconstruction to the area specified by $\hat{\boldsymbol{M}}$. Then, the final reconstructed image $\left[\hat{\boldsymbol{I}}_{i,\mathrm{r}}\right]_{x,y,:}$ of the coarse and the fine network is obtained from $\left[\boldsymbol{I}_\mathrm{o}\right]_{x,y,:}$ if $\left[\hat{\boldsymbol{M}}\right]_{x,y} = 0$ or from the network's output $\left[\hat{\boldsymbol{I}}_i\right]_{x,y,:}$ if $\left[\hat{\boldsymbol{M}}\right]_{x,y} = 1$. Unlike the binary ground-truth mask $\boldsymbol{M}_\mathrm{gt}$, $\hat{\boldsymbol{M}}$ is continuous $\in (0, 1)$, resulting in the following relationship of the reconstruction block

$$\hat{\boldsymbol{I}}_{i,\mathrm{r}} = (\mathbb{1}_{r,r} - \hat{\boldsymbol{M}}) \odot \boldsymbol{I}_\mathrm{o} + \hat{\boldsymbol{M}} \odot \hat{\boldsymbol{I}}_i \qquad \forall\, i \in \{\mathrm{c}, \mathrm{f}\}. \tag{4.3}$$

After a minor transformation of Equation (4.3), the reconstruction block can be drawn as a signal flow graph, which is depicted in Figure 4.6 (left).

By using Equation (4.1) and assuming reliable mask detection $\hat{\boldsymbol{M}} = \boldsymbol{M}_\mathrm{gt}$, Equation (4.3) can be rewritten to

$$\hat{\boldsymbol{I}}_{i,\mathrm{r}} = (\mathbb{1}_{r,r} - \boldsymbol{M}_\mathrm{gt}) \odot \boldsymbol{I}_\mathrm{gt} + \underbrace{(\mathbb{1}_{r,r} - \boldsymbol{M}_\mathrm{gt}) \odot \boldsymbol{M}_\mathrm{gt}}_{=\mathbf{0}} \odot \boldsymbol{c} + \boldsymbol{M}_\mathrm{gt} \odot \hat{\boldsymbol{I}}_i \tag{4.4}$$

$$= (\mathbb{1}_{r,r} - \boldsymbol{M}_\mathrm{gt}) \odot \boldsymbol{I}_\mathrm{gt} + \boldsymbol{M}_\mathrm{gt} \odot \hat{\boldsymbol{I}}_i \qquad \forall\, i \in \{\mathrm{c}, \mathrm{f}\}. \tag{4.5}$$

Equation (4.5) demonstrates that by concluding the coarse and the fine network with the reconstruction block, only pixels indicated by $\hat{\boldsymbol{M}}$ are modified, whereas ground-truth pixel values are used for non-occluded pixels mitigating unwanted artifacts. Still, reliable mask detection is essential.

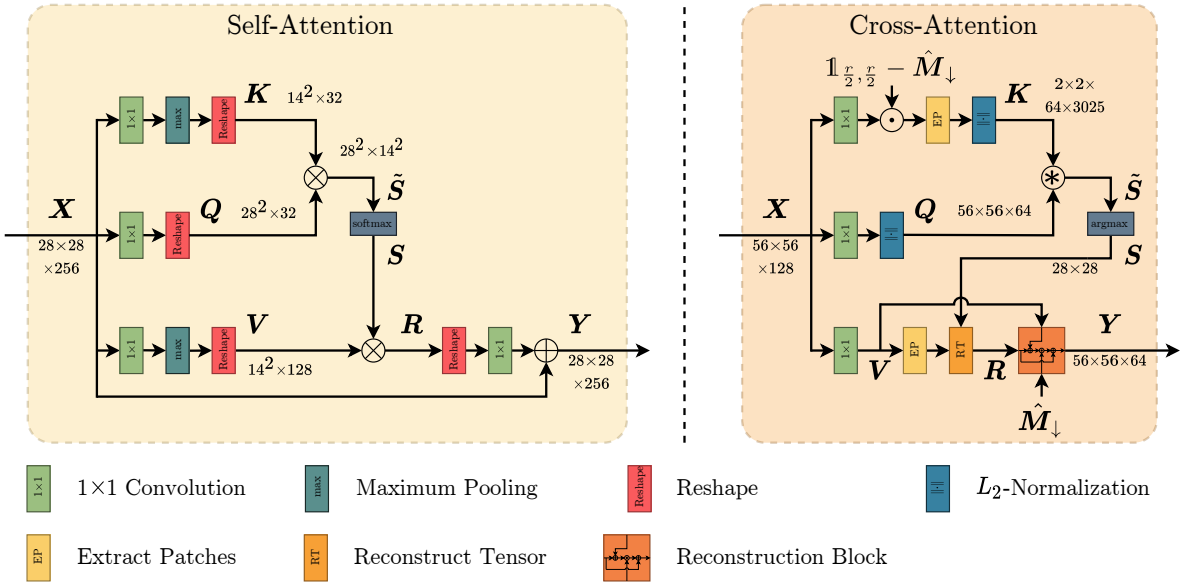

Figure 4.7: The global pixel-wise self-attention block (left) is designed to leverage global information without any restriction to the occluded area, whereas the patch-wise cross-attention block (right) searches and substitutes 2×2 patches of the occluded area with the most similar patches in the non-occluded area. Adapted from [13†].

4.4.1.2 Dense Multi-Scale Fusion Block

As motivated in Section 4.3.1, leveraging long-range pixel relationships is crucial in order to accomplish a realistic and semantically coherent reconstruction. Dilated convolutions provide large receptive fields with a low number of parameters. However, they only produce sparse activations and therefore require postprocessing. Hui *et al.* [89] proposed the DMFB to address the disadvantages of the dilated convolutions while maintaining a manageable parameter count. The implementation of the DMFB is illustrated in Figure 4.6 (right). First, a 3×3 convolutional layer reduces the number of feature maps by a factor of four. Then, four parallel branches extract information at multiple scales utilizing a single 3×3 convolution with a different dilation factor $D \in \{1, 2, 4, 8\}$. In order to obtain dense multi-scale features from the sparse activation for $D > 1$, all four outputs are combined in a cumulative manner, followed by another 3×3 convolution. After concatenating the four branches, a 1×1 convolution fuses the information.

Overall, the DMFB is designed as a residual block, which eases convergence during training and allows the network to focus on improving the input by leveraging the multi-scale information. Besides, all convolutional layers are followed by a leaky ReLU (LReLU) [141] activation function. With this unique structure, the DMFB provides the network with a parameter-efficient method to generate dense multi-scale features, which are essential in image inpainting. By enforcing this multi-scale structure, the network does not need to implicitly encode a similar relationship utilizing multiple convolutional layers, which speeds up convergence. Therefore, the DMFB is employed multiple times within the coarse-to-fine generator.

4.4.1.3 Pixel-Wise Self-Attention

Nowadays, attention blocks are vital in image inpainting, as listed in Section 4.3.1. This work incorporates a pixel-wise self-attention block, depicted in Figure 4.7 (left). A pixel-wise attention block is a modification of the non-local block of Wang *et al.* [220] by Zhang *et al.* [264], who successfully employed it in a self-attention GAN. Later, pixel-wise attention blocks demonstrated their benefits in image inpainting [244, 274, 284]. This work follows the implementation of Hörmann *et al.* [13†], which is based on the original implementation [220, 264].

The input of the self-attention block constitutes a feature tensor $\boldsymbol{X} \in \mathbb{R}^{28\times28\times256}$. First, $\boldsymbol{X}$ is split into three branches – $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ – by applying three 1×1 convolutions. The number of feature maps C is set to 32 for $\boldsymbol{Q}$ and $\boldsymbol{K}$, and to 128 for $\boldsymbol{V}$ since Zhang *et al.* [264] observed no decrease in performance when reducing C by a factor of 8. As in [220], maximum pooling is employed for $\boldsymbol{K}$ and $\boldsymbol{V}$ to further decrease the memory footprint. Even though this increases the sparseness of the computation, it does not alter the general behavior of the self-attention block. After reshaping every branch to perform a matrix multiplication, three tensors are obtained: query $\boldsymbol{Q}$, key $\boldsymbol{K}$, and value $\boldsymbol{V}$.

The network encodes in $\big[\boldsymbol{Q}\big]_{i,:}$ the reference to a key $\big[\boldsymbol{K}\big]_{j,:}$ which is associated with a value $\big[\boldsymbol{V}\big]_{j,:}$. Thus, the key-value pairs form a dictionary $\big\{\big(\big[\boldsymbol{K}\big]_{j,:}, \big[\boldsymbol{V}\big]_{j,:}\big)\big\}_{j=1}^{14^2}$, which is queried by $\big[\boldsymbol{Q}\big]_{i,:}$. In this way, $\big[\boldsymbol{Q}\big]_{i,:}$ defines a queried property for the ith pixel, $\big[\boldsymbol{K}\big]_{j,:}$ constitutes the property of the jth pixel, and $\big[\boldsymbol{V}\big]_{j,:}$ denotes the value of the property used for further computations.

Mathematically, the self-attention block is written as follows. First, the attention map $\boldsymbol{S} \in \mathbb{R}^{28^2\times14^2}$ containing the similarity scores is computed by

$$\boldsymbol{S} = \operatorname{softmax}\left(\boldsymbol{Q}\boldsymbol{K}^{\mathsf{T}}\right), \tag{4.6}$$

where the softmax$(\cdot)$ is applied to the columns of $\tilde{\boldsymbol{S}}$. Therefore, $\big[\boldsymbol{S}\big]_{i,j}$ denotes the normalized relevance of the jth key $\boldsymbol{K}$ for the ith pixel in $\boldsymbol{Q}$. Then, a feature for the ith pixel according to their query is selected from $\boldsymbol{V}$ by matrix multiplication

$$\boldsymbol{R} = \boldsymbol{S}\boldsymbol{V}. \tag{4.7}$$

In Equations (4.6) and (4.7), the two steps of a typical attention mechanism become apparent: 1) compare by computing a similarity; and 2) attend by moving the features to the desired pixels. After reshaping $\boldsymbol{R}$ and applying a 1×1 convolution to restore the initial number of feature maps, the output $\boldsymbol{Y}$ of the self-attention block is obtained by adding the input $\boldsymbol{X}$. Similar to the DMFB, this skip connection between input $\boldsymbol{X}$ and output $\boldsymbol{Y}$ facilitates convergence.

With this unique combination of relatively simple operations, the pixel-wise self-attention block can refine the input $\boldsymbol{X}$ by leveraging global information, which results in a receptive field equal to the size of the input feature map. However, this global influence comes with a rather large memory footprint. Therefore, the self-attention block is employed rather scarcely.

4.4.1.4 Patch-Wise Cross-Attention

As listed in Section 4.3.1, many image inpainting researchers [64, 196, 249, 254, 260] favor patch-wise attention over pixel-wise attention. The patch-wise cross-attention follows the same concept as the pixel-wise self-attention; however, two differences change the block's purpose fundamentally: 1) instead of substituting pixel-wise information, patches of 2×2 pixels allow the consideration of spatial dependency within a small vicinity; and 2) while self-attention performs unrestricted information exchange between all regions in the input, cross-attention only allows the replacement of patches from the occluded area with patches from the non-occluded area as in [64, 196, 249, 254, 260, 274, 284]. Multiple changes to the pixel-wise self-attention block are required to obtain such behavior. Nevertheless, the purpose of every branch $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ is identical.

Figure 4.7 (right) depicts the cross-attention block, which was adapted to blind face completion from [64, 196, 249, 254, 260] by Hörmann *et al.* [13†]. Similar to the self-attention block, the cross-attention block uses 1×1 convolutions to reduce the number of channels to 64 in every branch. After element-wise multiplication of the key branch with $\mathbb{1}_{\frac{r}{2},\frac{r}{2}} - \hat{\boldsymbol{M}}_{\downarrow}$, where $\hat{\boldsymbol{M}}_{\downarrow}$ denotes the mask predicted by the network downsized to 56×56, all activations within the occluded area are set to 0. Thus, only reliable information, *i.e.*, only keys $\boldsymbol{K}$ corresponding to values $\boldsymbol{V}$ extracted from the non-occluded areas, are considered. Unlike pixel-wise attention, every patch constitutes a key. Hence, $55^2 = 3025$ overlapping patches of size 2×2×64 are extracted and L_2-normalized, such that

$$\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{64}\left[\boldsymbol{K}\right]^2_{i,j,k,l} = 1 \qquad \forall\, l. \tag{4.8}$$

The query $\boldsymbol{Q}$ is also L_2-normalized, however, only along the channel dimension. Then, a convolution of the input $\boldsymbol{Q}$ with kernel $\boldsymbol{K}$ is performed with stride $S = 2$ according to Equation (2.9). Here, the fact that Equation (2.9) actually performs a cross-correlation is leveraged. Thus, the output $\tilde{\boldsymbol{S}} \in \mathbb{R}^{28\times28\times3025}$ of this convolution denotes the similarity of the 28^2 non-overlapping patches of size 2×2×64 of $\boldsymbol{Q}$ to the 3025 patches of $\boldsymbol{K}$. In fact, due to prior L_2-normalization $\tilde{\boldsymbol{S}}$ constitutes the cosine similarity, which corresponds to $1 - d(\boldsymbol{Q}, \boldsymbol{K})$ (*cf.* Equation (3.17)). Next, the matrix $\boldsymbol{S}$ containing the index of the most similar patch is computed by

$$\left[\boldsymbol{S}\right]_{x,y} = \arg\max_{c}\left[\tilde{\boldsymbol{S}}\right]_{x,y,c}. \tag{4.9}$$

As earlier for $\boldsymbol{K}$, $55^2 = 3025$ overlapping patches of size 2×2×64 are extracted from $\boldsymbol{V}$. Then, these patches are used to reconstruct the tensor $\boldsymbol{R}$ according to the similarity of $\boldsymbol{Q}$ with $\boldsymbol{K}$ measured by $\boldsymbol{S}$. While excluding the occluded region from $\boldsymbol{K}$ led to the corresponding patches in $\boldsymbol{K}$ not being queried by $\boldsymbol{Q}$, it is still necessary to ensure that the cross-attention block only alters the occluded region. This is accomplished by applying the reconstruction block (see Section 4.4.1.1) to $\boldsymbol{R}$, where $\boldsymbol{V}$ provides the values for the non-occluded areas and its output constitutes the output $\boldsymbol{Y}$ of the cross-attention block. Like in the self-attention block, employing the reconstruction block serves as a skip connection for the entire attention component, resulting in improved convergence.

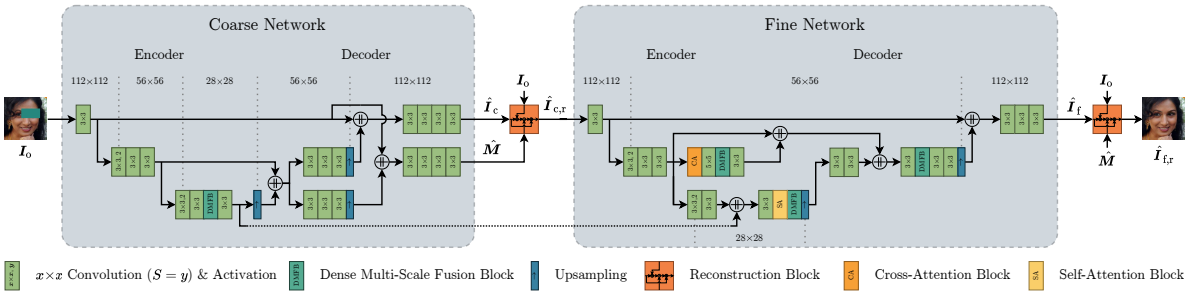

Figure 4.8: Architecture of the generator. First, the coarse network creates a rough reconstruction $\hat{\boldsymbol{I}}_{\mathrm{c}}$ of the occluded face $\boldsymbol{I}_{\mathrm{o}}$ and predicts the mask $\hat{\boldsymbol{M}}$. After passing $\hat{\boldsymbol{I}}_{\mathrm{c}}$ through the reconstruction block (see Section 4.4.1.1), the reconstructed face $\hat{\boldsymbol{I}}_{\mathrm{c,r}}$ is refined in the fine network yielding $\hat{\boldsymbol{I}}_{\mathrm{f}}$. While the coarse network only uses a single dense multi-scale fusion block (DMFB) for information exchange, multiple DMFBs together with a dual attention structure allow the fine network to extract and exchange global information, which is crucial for a realistic prediction of the occluded pixels. Adapted from [13†].

Through several modifications, the pixel-wise attention block is transformed to cope with 2×2 patches, which allows the network to transfer detailed and realistic textures. By restricting the information exchange to replace patches exclusively within the occluded area with patches exclusively from the non-occluded area, the cross-attention block further ensures that reliable information is exchanged and that the non-occluded area is left untouched.

4.4.2 Coarse-to-Fine Generator

As motivated by Section 4.4, the blind FR task is split into two steps: 1) The coarse network outputs a rough prediction of the occluded area, which is then 2) refined by the fine network. This separation is widely employed [28, 123, 139, 182, 249, 254, 255, 286] as it allows each network to focus on its respective task. The detailed architecture is depicted in Figure 4.8 and will be explained in detail in the following subsections. Overall, both networks are composed similarly following the popular U-Net architecture [179]. However, Hörmann *et al.* [13†] introduced multiple modifications, which are necessary to also predict the mask $\hat{\boldsymbol{M}}$ in the coarse network and consider global information exchange in the fine network in order to obtain a realistic reconstruction.

4.4.2.1 Coarse Network

The spatial resolution of the occluded face $\boldsymbol{I}_{\mathrm{o}}$ at the input is reduced from 112×112 to 28×28 utilizing convolutional layers with a 3×3 kernel and stride $S = 2$. At the same time, the number of feature maps C is doubled with every reduction starting from $C^{[1]} = 64$. After six convolutional layers with kernel size 3×3, the DMFB [89] (see Section 4.4.1.2) is employed to exchange information at multiple scales. In this way, the coarse network incorporates information from different regions to produce a rough

estimate of the occluded area. The DMFB is then followed by another 3×3 convolutional layer that concludes the encoder of the coarse network.

As typical for U-Net architectures, the latent feature map is decoded by concatenating the upsampled latent feature map with the respective feature map of the same resolution from the encoder. Before the concatenated feature maps are upsampled again, they are processed by three 3×3 convolutional layers per resolution. As discussed in Section 2.5.2, utilizing transposed convolutions causes checkerboard artifacts [165]. Thus, unlike [77] and following [80, 90], upsampling with nearest-neighbor interpolation is applied.

Since the coarse network predicts the reconstructed face $\hat{\boldsymbol{I}}_{\mathrm{c}}$ and the mask $\hat{\boldsymbol{M}}$, the decoder uses two parallel, almost identical branches. Both branches are concluded with a final convolutional layer, which sets the number of feature maps to $C^{[L]} = 3$ for $\hat{\boldsymbol{I}}_{\mathrm{c}}$ and to $C^{[L]} = 1$ for $\hat{\boldsymbol{M}}$. Besides, the value range of $\hat{\boldsymbol{I}}_{\mathrm{c}}$ is limited to $[-1, 1]$ via clipping, whereas the sigmoid activation function ensures a proper value range for $\hat{\boldsymbol{M}}$. Apart from the two convolutional layers, which conclude both branches, LReLU [141] is employed as an activation function in the encoder, whereas rectified linear unit (ReLU) [159] is used in the decoder following [244]. Due to its non-zero gradients for inputs < 0, LReLU is particularly helpful when training GANs as it can cope with sparse gradients.

4.4.2.2 Fine Network

The fine network implements the U-Net structure [179] similarly to the coarse network. While its single output $\hat{\boldsymbol{I}}_{\mathrm{f}}$ simplifies the structure, the dual attention structure requires additional modifications [13†].

First, the coarse reconstruction $\hat{\boldsymbol{I}}_{\mathrm{c,r}}$ is downsampled to a spatial resolution of 28×28 using convolutional layers with 3×3 kernels. The dual attention structure is utilized in the network at different resolutions. Since the cross-attention block operates on 2×2 patches, it considers the spatial dependency in the feature maps like in textures. Therefore, the cross-attention block benefits from prominent textures, which are more frequent at a relatively large spatial resolution of 56×56. In contrast, the pixel-wise self-attention block is applied at a resolution of 28×28. In order to allow the self-attention block to also leverage low-level features, the latent feature map extracted by the coarse network is fused at its input.

A DMFB follows both attention blocks to further refine the feature. A similar structure was proposed by Zeng *et al.* [260]; however, they only employed dilated convolutions without the additional hierarchical fusion. After upsampling the self-attention branch, the information is combined at a resolution of 56×56 by concatenation of: 1) the pixel-wise self-attention branch; 2) the patch-wise cross-attention branch; and 3) the feature map of the same resolution from the encoder. The fine network is concluded similarly to the coarse network with the expection of using another DMFB to promote the information exchange after fusing all branches. Following [254], exponential linear unit (ELU) [27] is employed as the activation function throughout the fine network.

The parallel employment of two attention blocks with distinct objectives similar to [48, 284] leads to complementing behavior even though both blocks are designed to handle long-range dependencies within feature maps. The cross-attention block excels

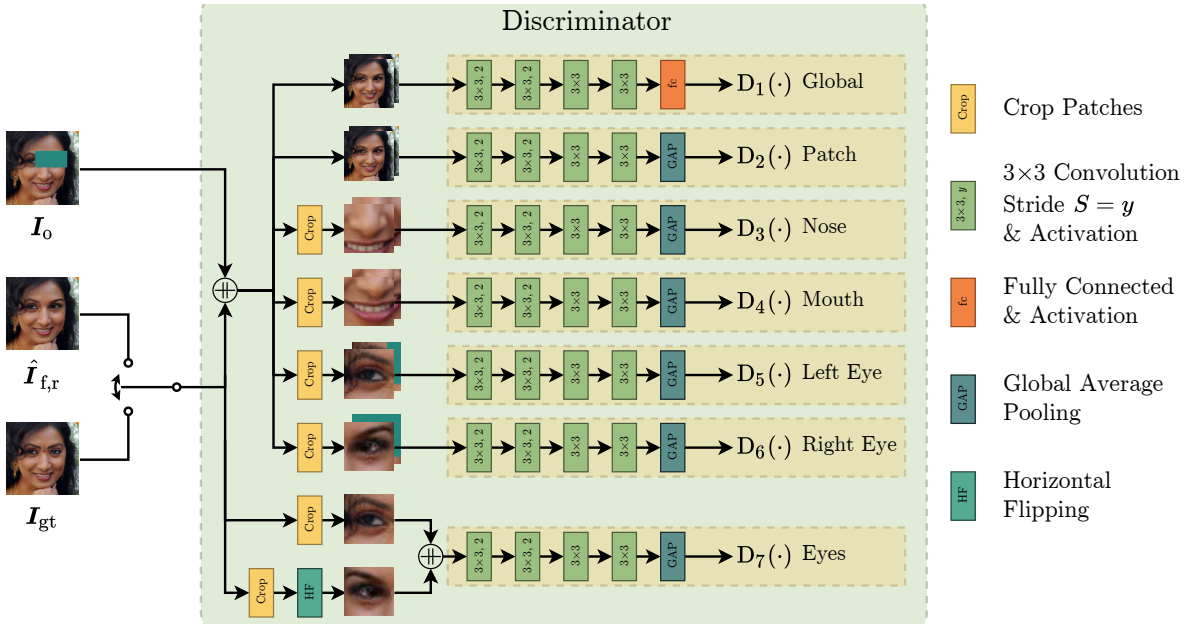

Figure 4.9: Architecture of the discriminator. In total, seven discriminators are employed: While $D_1(\cdot)$ is a global discriminator, all remaining discriminators are patch discriminators [97]. Except for $D_7(\cdot)$, all discriminators take as input the concatenation of the occluded face $\boldsymbol{I}_o$ with either the ground-truth face $\boldsymbol{I}_{gt}$ or the reconstruction by the generator $\hat{\boldsymbol{I}}_{f,r}$. All patch discriminators except $D_2(\cdot)$ focus on relevant face regions, which are cropped to 28×28 pixels around the corresponding facial landmark. $D_7(\cdot)$ considers both eyes from either $\boldsymbol{I}_{gt}$ or $\hat{\boldsymbol{I}}_{f,r}$ and therefore enforces consistency around the eye regions.

when the occlusion is not occurring symmetrically, *e.g.*, if only the right face half is occluded, since it finds and incorporates similar textures in the non-occluded parts of the face. This is particularly important when reconstructing eyes. Owing to the spatial dependency within the patches, the cross-attention block maintains textural consistency. The information exchange is restricted to avoid that the non-occluded area is altered by unreliable information originating from the occluded face area. In contrast, the self-attention block focuses on reconstructing unique face parts, *e.g.*, the nose or the mouth, by learning the relationship between all pixels within the entire feature map. Unlike the approach by Zhou *et al.* [284], the self-attention block is not limited to the occluded area.

4.4.3 Discriminator

As introduced in Section 2.7.1, the discriminator plays a vital role during GAN training. Here, it ensures that the reconstructed face $\hat{\boldsymbol{I}}_{f,r}$ is indistinguishable from the ground-truth face $\boldsymbol{I}_{gt}$, *i.e.*, before the synthetic occlusion was applied. In order to make the discriminator focus on the relevant regions and not base the final decision on a single output, multiple discriminators with different purposes are utilized [44, 219]. This technique is widespread in image inpainting tasks, in which two discriminators [31, 269],

one global and one patch discriminator [97], or even up to seven discriminators [284, 13†] are employed. This work follows the approach from Hörmann *et al.* [13†] by incorporating seven discriminators $\mathrm{D}_i(\cdot)$ – one global discriminator and six patch discriminators similar to [97], which are depicted in Figure 4.9.

Except for the discriminator $\mathrm{D}_7(\cdot)$, the input to all remaining discriminators $\mathrm{D}_i(\cdot)$ consists of the concatenation of the occluded face $\boldsymbol{I}_\mathrm{o}$ with either the ground-truth face $\boldsymbol{I}_\mathrm{gt}$ or the reconstruction by the generator $\hat{\boldsymbol{I}}_\mathrm{f,r}$. By providing the discriminators with $\boldsymbol{I}_\mathrm{o}$, the discriminators learn to pay attention to the occluded areas.

While the input of the global discriminator $\mathrm{D}_1(\cdot)$ and the patch discriminator $\mathrm{D}_2(\cdot)$ contains the entire face region 112×112, the inputs of all remaining discriminators are crops of size 28×28 around the respective facial landmark (see Section 3.2).[ii] Despite prior face alignment, a considerable difference between the target position and the actual position after alignment remains due to large yaw angles of the head poses and different face proportions. Hence, leveraging the facial landmarks after alignment ensures that all crops resemble each other as much as possible. The facial landmarks after alignment are obtained without any costly detection using Equation (3.2) with the transformation matrix $\boldsymbol{T}^{(n)}$, which was determined for the face alignment of the nth image. Moreover, storing and loading facial landmarks is only required for the training since all discriminators are discarded after training.

In order to cope with inconsistent eye colors, which caused unrealistic results upon closer inspection in [260, 284], Hörmann *et al.* [13†] proposed another discriminator. This discriminator, denoted as $\mathrm{D}_7(\cdot)$, processes the concatenation of the left eye with the right eye, which was horizontally flipped for improved spatial alignment.

Regardless of their purpose, all discriminators are built similarly and comprise four 3×3 convolutional layers with LReLU [141] as an activation function. The former two convolutions operate with strides $S = 2$ to reduce the spatial resolution to 28×28 or 7×7. The number of feature maps is doubled in the first three layers from initially $C^{[1]} = 64$ for $\mathrm{D}_1(\cdot)$ and $\mathrm{D}_2(\cdot)$, and $C^{[1]} = 32$ for all remaining discriminators. The fourth layer then outputs a single feature map, *i.e.*, $C^{[4]} = 1$. For all patch discriminators $\mathrm{D}_i(\cdot)$ with $i \in \{2, 3, \ldots, 7\}$, sigmoid is used as an activation function after the fourth layer $\Psi^{[4]}(\cdot)$ to obtain a patch-wise probability, which denotes the probability of the corresponding patch belonging to $\boldsymbol{I}_\mathrm{gt}$, *i.e.*, the patch is realistic. After global average pooling (GAP), a scalar probability is obtained for all patch discriminators. The global discriminator $\mathrm{D}_1(\cdot)$ obtains the probability utilizing a fully-connected layer followed by a sigmoid.

Unlike [244], Hörmann *et al.* [13†] dispensed with normalization layers. This is mainly due to the massive impact of the occluded area in $\boldsymbol{I}_\mathrm{o}$ on the mean and variance. Various related works, including image inpainting [254], super-resolution [125, 222], and deblurring [158], support the removal of the normalization layer and demonstrated that it leads to improved performance and reduced computational complexity.

[ii] The average of the two landmarks marking the mouth corners is used for the crop around the mouth region.

4.4.4 Face Feature Extractor

The objective of the face feature extractor is to guide the reconstruction with meaningful face identity features. The purpose of this supervision is two-fold:

1. Without consideration of any face identity features, the reconstruction is defined by the realism judged by the discriminators. In contrast to previous face completion methods, which used general feature maps extracted from image classification networks [64, 80, 121, 127, 139, 140, 160, 237, 246, 261, 284, 286], Hörmann *et al.* [13†] opted for a network explicitly trained on FR as in [138, 250], increasing the relevance of the extracted feature maps.

2. For an optimal reconstruction, consistency of the face features within the face is crucial, *i.e.*, features originating from occluded areas in $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ must not contradict features extracted from the non-occluded areas.

Thus, supervision by FR features is vital to ensure that $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ mitigates the drop in accuracy in faces manipulated by occlusions, as observed in Section 4.2. Besides guiding the training to consider face identity features, the face feature extractor is further employed to evaluate the FR performance of $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$.

Hörmann *et al.* [13†] employed the VGG-112 model, which was introduced and analyzed exhaustively in Section 3.6.2, as a face feature extractor. VGG-112 constitutes a ResNet-v2 [72] with depth $L = 50$ and a $M_{\mathrm{f}} = 256$-dimensional bottleneck layer, from which the face identity features are extracted. The network is trained on the VGGFace2 dataset [14] with softmax CE loss for 20 epochs. The faces were aligned following the custom face alignment policy (FAP) with facial landmarks extracted by the multi-task CNN (MTCNN) [265]. All training parameters are identical to those reported in Section 3.5.1 except for a smaller initial learning rate $\eta = 0.3$ to account for a reduced batch size $N_{\mathrm{b}} = 64$ due to memory limitations. This change ultimately resulted in slightly different values to those reported in Section 3.6.2. Since this work follows Hörmann *et al.* [13†], their model is used to maintain consistency and avoid costly retraining as all results depend on the choice of the feature extractor.

As discussed in Section 3.6.2, VGG-112 does not provide state-of-the-art results and can be improved by incorporating additive margin loss [33]. However, since this chapter focuses on mitigating the effect on FR performance caused by occlusion as analyzed in Section 4.2, a state-of-the-art FR model is not required as similar behavior is observed for many FR models [60] and thus is also expected to be observed in more recent approaches. Therefore, VGG-112 constitutes a viable choice for blind FR.

4.5 Loss Functions

Typically, GANs are trained in an alternating manner, *i.e.*, two total losses are required – one for the generator $\mathrm{G}(\cdot)$ and one for the discriminator $\mathrm{D}(\cdot)$. Figure 4.5 depicts an overview of all loss terms and their dependencies. A combination of multiple losses is required in order to obtain the desired behavior, *i.e.*, 1) a rough reconstruction $\hat{\boldsymbol{I}}_{\mathrm{c}}$ by the

coarse network; 2) a reliable mask prediction $\hat{\boldsymbol{M}}$; and 3) a photo-realistic reconstruction $\hat{\boldsymbol{I}}_{\mathrm{f}}$ by the fine network with 4) coherent face identity features matching the ground-truth image $\boldsymbol{I}_{\mathrm{gt}}$. Thus, at least one loss term is dedicated to one of the objectives mentioned above.

Pixel-wise similarity losses $\mathscr{L}_{\mathrm{pix}}$ cannot enforce complex spatial dependencies and thus suffice to ensure a rough estimate $\hat{\boldsymbol{I}}_{\mathrm{c}}$. Since no spatial dependencies occur in masks, a reliable prediction of the mask $\hat{\boldsymbol{M}}$ is also covered by a pixel-wise similarity loss $\mathscr{L}_{\mathrm{pix}}$. To improve realism and encourage crisp details in $\hat{\boldsymbol{I}}_{\mathrm{f}}$, adversarial losses $\mathscr{L}_{\mathrm{adv}}$ from multiple discriminators $D_i(\cdot)$ are employed. Coherent face identity features are obtained by incorporating identity losses $\mathscr{L}_{\mathrm{id}}$ based on meaningful features extracted by a face feature extractor $\mathrm{F}(\cdot)$.

As in [13†], all loss terms are merged into a single total loss function for the generator

$$\begin{aligned}\mathscr{L}_{\mathrm{tot}}^{\mathrm{G}} = {} & \lambda_{\mathrm{pix}}^{\mathrm{o}}\mathscr{L}_{\mathrm{pix}}^{\mathrm{o}} + \lambda_{\mathrm{pix}}^{\mathrm{no}}\mathscr{L}_{\mathrm{pix}}^{\mathrm{no}} + \lambda_{\mathrm{pix}}^{\mathrm{m}}\mathscr{L}_{\mathrm{pix}}^{\mathrm{m}} + \\ & \lambda_{\mathrm{adv}}^{\mathrm{G}}\mathscr{L}_{\mathrm{adv}}^{\mathrm{G}} + \lambda_{\mathrm{id}}^{\mathrm{per}}\mathscr{L}_{\mathrm{id}}^{\mathrm{per}} + \lambda_{\mathrm{id}}^{\mathrm{style}}\mathscr{L}_{\mathrm{id}}^{\mathrm{style}},\end{aligned} \tag{4.10}$$

where λ denote scalars to balance the losses. The discriminators are trained solely based on the adversarial loss $\mathscr{L}_{\mathrm{tot}}^{\mathrm{D}} = \mathscr{L}_{\mathrm{adv}}^{\mathrm{D}}$.

4.5.1 Pixel-wise Similarity Losses

A pixel-wise similarity loss, *e.g.*, L_1 or L_2 distance between $\hat{\boldsymbol{I}}_{\mathrm{c}}$ and $\boldsymbol{I}_{\mathrm{gt}}$, constitutes the most straightforward choice in image inpainting as the network is punished if it fails to model the exact pixel value. However, multiple problems arise in image inpainting tasks since $\boldsymbol{I}_{\mathrm{gt}}$ is typically not the single acceptable solution. This ambiguity is also widespread in face completion. *E.g.*, if the mouth is completely occluded, the network cannot know whether the mouth is open or closed unless a muscle contraction in the non-occluded region reveals its state. Thus, reconstructing an open or closed mouth is equally valid as long as the whole face with makeup, beard, *etc.*, is consistent. Even extreme scenarios are conceivable, where $\boldsymbol{I}_{\mathrm{gt}}$ is an "unacceptable" solution, *e.g.*, after manually covering acne or a birthmark with a mask, the reconstruction should not reconstruct it.

Since pixel-wise similarity losses punish the network if it makes the "wrong" decision, *i.e.*, $\hat{\boldsymbol{I}}_{\mathrm{c}} \neq \boldsymbol{I}_{\mathrm{gt}}$, it does not dare to make a decision that may result in ambiguity. Hence, the network takes only safe decisions, *e.g.*, reconstructing only a rough position of an eye, which ultimately results in a blurry reconstruction containing mainly low frequencies and entirely lacking high-frequent details. This also becomes evident since every pixel in $\hat{\boldsymbol{I}}_{\mathrm{c}}$ is compared independently with its corresponding pixel in $\boldsymbol{I}_{\mathrm{gt}}$. Therefore, shifting $\boldsymbol{I}_{\mathrm{gt}}$ by a single pixel leads to an increased loss with the original $\boldsymbol{I}_{\mathrm{gt}}$ even though the shift is invisible to the naked eye. To prevent an exploding loss, the network creates blurry outputs, which are less susceptible to such shifts. Still, pixel-wise losses perfectly fit the purpose of the coarse network.

Following the majority of image inpainting works [20, 28, 39, 64, 64, 77, 80, 123, 127, 133, 138, 139, 160, 216, 237, 246, 249, 255, 261, 286], L_1 distance is employed since outliers are punished less by the L_1 distance than by the L_2 distance [121, 122, 254, 267]. Then,

the L_1 distance losses are computed separately for the occluded $\mathscr{L}_{\text{pix}}^{\text{o}}$ and non-occluded area $\mathscr{L}_{\text{pix}}^{\text{no}}$ in order to weigh them individually. Formally, both losses are formulated as

$$\mathscr{L}_{\text{pix}}^{\text{o}} = \frac{\|\big(\hat{\boldsymbol{I}}_{\text{c}} - \boldsymbol{I}_{\text{gt}}\big) \odot \boldsymbol{M}_{\text{gt}}\|_1}{\|\boldsymbol{M}_{\text{gt}}\|_1}, \tag{4.11}$$

$$\mathscr{L}_{\text{pix}}^{\text{no}} = \frac{\|\big(\hat{\boldsymbol{I}}_{\text{c}} - \boldsymbol{I}_{\text{gt}}\big) \odot \big(\mathbb{1}_{r,r} - \boldsymbol{M}_{\text{gt}}\big)\|_1}{\|\mathbb{1}_{r,r} - \boldsymbol{M}_{\text{gt}}\|_1}. \tag{4.12}$$

Normalizing both losses with the (non-)occluded area is crucial to ensure that the losses do not depend on the size of occlusion and thus may vary largely between batches.

A reliable mask prediction is essential for the outcome of the blind FR approach as it determines whether a pixel in the occluded image $\boldsymbol{I}_{\text{o}}$ is substituted by the reconstruction $\hat{\boldsymbol{I}}_{\text{f}}$ in every reconstruction block (see Section 4.4.1.1). Due to the binary ground-truth mask $\boldsymbol{M}_{\text{gt}}$, the mask prediction task can be interpreted as a binary pixel-wise classification [28, 77, 123, 216, 224]. Thus, a binary CE loss is employed and formulated by

$$\mathscr{L}_{\text{pix}}^{\text{m}} = -\frac{1}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r}\big[\boldsymbol{M}_{\text{gt}}\big]_{i,j}\log\big(\big[\hat{\boldsymbol{M}}\big]_{i,j}\big) + \big(1 - \big[\boldsymbol{M}_{\text{gt}}\big]_{i,j}\big)\log\big(1 - \big[\hat{\boldsymbol{M}}\big]_{i,j}\big). \tag{4.13}$$

4.5.2 Adversarial Losses

As elaborated in Section 4.5.1, losses minimizing L_1 distances fail to provide realistic results as they promote blurry images lacking crisp details. Hence, additional losses are required to ensure a photo-realistic reconstruction, indistinguishable from real images such as $\boldsymbol{I}_{\text{gt}}$. Note that "indistinguishable" in this context does not mean that $\|\hat{\boldsymbol{I}}_{\text{f}} - \boldsymbol{I}_{\text{gt}}\|_1 = 0$ but rather that $\hat{\boldsymbol{I}}_{\text{f}}$ is perceived as similarly realistic as $\boldsymbol{I}_{\text{gt}}$. Embedding the blind face completion network inside a GAN allows a discriminator $\mathrm{D}(\cdot)$ to judge realism. Nowadays, this offers the best way to obtain realistic results since realism cannot be grasped mathematically and written as a derivable loss function. The discriminator is automatically trained to consider ambiguous solutions, *e.g.*, open and closed mouths, as equally viable as long as both options are realistic, which lifts the limitations imposed by the L_1 distance losses.

Despite the complex task of judging realism, this so-called adversarial loss can be expressed using a simple binary CE loss. Due to the discriminators' scalar output $D_i(\cdot) \in (0, 1)$, every discriminator yields the probability of its input being realistic. Then, the ground-truth label of the binary CE loss is set dependent on the origin of the input and as to whether the generator or the discriminator is trained. As introduced in Section 4.4.3, seven discriminators $D_i(\cdot)$ are employed. Thus, the adversarial loss of the generator $\mathscr{L}_{\text{adv}}^{\text{G}}$ is written as the average of all discriminators by

$$\begin{aligned}\mathscr{L}_{\text{adv}}^{\text{G}} = -\frac{1}{7}\Bigg[\sum_{i=1}^{2}\log\Big(\mathrm{D}_i(\hat{\boldsymbol{I}}_{\text{f,r}} \oplus \boldsymbol{I}_{\text{o}})\Big) + \sum_{i=3}^{6}\log\Big(\mathrm{D}_i(\mathrm{C}_i(\hat{\boldsymbol{I}}_{\text{f,r}} \oplus \boldsymbol{I}_{\text{o}}))\Big) + \\ \log\Big(\mathrm{D}_7\Big(\mathrm{C}_5(\hat{\boldsymbol{I}}_{\text{f,r}}) \oplus \tilde{\mathrm{C}}_6(\hat{\boldsymbol{I}}_{\text{f,r}})\Big)\Big)\Bigg],\end{aligned} \tag{4.14}$$

where $\mathrm{C}_i(\cdot)$ for $i \in \{3,4,5,6\}$ denotes a method to crop the input image to a 28×28 patch centered around the nose, mouth, left eye, and right eye, respectively, and $\tilde{\mathrm{C}}_6(\cdot)$ that horizontal flipping is applied after cropping. $\mathscr{L}_{\mathrm{adv}}^{\mathrm{G}}$ is minimized if $\mathrm{D}_i(\cdot) \approx 1$, meaning that $\mathrm{D}_i(\cdot)$ perceives the input as realistic. Hence, $\mathrm{G}(\cdot)$ successfully deceives the $\mathrm{D}_i(\cdot)$.

The discriminators $\mathrm{D}_i(\cdot)$ are trained to discern fake images $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ from real images $\boldsymbol{I}_{\mathrm{gt}}$ by promoting $\mathrm{D}_i(\cdot) \approx 0$ for the former and $\mathrm{D}_i(\cdot) \approx 1$ for the latter. Thus, the adversarial loss for the discriminators is defined as

$$\begin{aligned}\mathscr{L}_{\mathrm{adv}}^{\mathrm{D}} = -\frac{1}{7}\Bigg[&\sum_{i=1}^{2} \log\Big(1 - \mathrm{D}_i(\hat{\boldsymbol{I}}_{\mathrm{f,r}} \oplus \boldsymbol{I}_{\mathrm{o}})\Big) + \log\left(\mathrm{D}_i(\boldsymbol{I}_{\mathrm{gt}} \oplus \boldsymbol{I}_{\mathrm{o}})\right) + \\ &\sum_{i=3}^{6} \log\Big(1 - \mathrm{D}_i(\mathrm{C}_i(\hat{\boldsymbol{I}}_{\mathrm{f,r}} \oplus \boldsymbol{I}_{\mathrm{o}}))\Big) + \log\left(\mathrm{D}_i(\mathrm{C}_i(\boldsymbol{I}_{\mathrm{gt}} \oplus \boldsymbol{I}_{\mathrm{o}}))\right) + \\ &\log\Big(1 - \mathrm{D}_7\Big(\mathrm{C}_5(\hat{\boldsymbol{I}}_{\mathrm{f,r}}) \oplus \tilde{\mathrm{C}}_6(\hat{\boldsymbol{I}}_{\mathrm{f,r}})\Big)\Big) + \log\Big(\mathrm{D}_7\Big(\mathrm{C}_5(\hat{\boldsymbol{I}}_{\mathrm{gt}}) \oplus \tilde{\mathrm{C}}_6(\hat{\boldsymbol{I}}_{\mathrm{gt}})\Big)\Big)\Bigg].\end{aligned} \tag{4.15}$$

As derived for a general problem in Section 2.7.2, Equations (4.14) and (4.15) demonstrate that the generator and discriminator pursue contrary objectives, which ultimately leads to the desired realistic reconstruction $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$.

4.5.3 Identity Losses

While adversarial losses promote a realistic reconstructed face $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$, they do not guarantee that the face features are coherent within the face. Thus, it is crucial to incorporate losses, which ensure that the reconstructed faces resemble the underlying identity. Yin and Di [250] compared the face feature $\boldsymbol{f}_{\mathrm{f,r}}$ of $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ with the face feature $\boldsymbol{f}_{\mathrm{gt}}$ of the ground-truth face $\boldsymbol{I}_{\mathrm{gt}}$. However, the face feature $\boldsymbol{f}$ does not contain any spatial information, which is essential when forcing the network to maintain coherent face features within the entire face. Therefore, multiple approaches [20, 28, 64, 80, 89, 121, 123, 127, 138–140, 160, 196, 216, 237, 246, 261, 284, 286] favor comparing intermediate feature maps extracted from $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ and $\boldsymbol{I}_{\mathrm{gt}}$. By selecting feature maps at different depths, the network's focus can be adjusted to low-level or high-level feature map similarity. Unlike [64, 80, 121, 127, 139, 140, 160, 237, 246, 261, 284, 286] and following Ma *et al.* [138], Hörmann *et al.* [13†] extracted feature maps from a network trained on FR. Therefore, the feature maps contain more meaningful information for face completion than feature maps extracted by image classification networks.

In the following paragraphs, two different approaches for measuring feature map dissimilarity are introduced, which were proposed by Johnson *et al.* [99]: 1) The perceptual loss $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ directly computes the distances between two feature maps, whereas 2) the style loss $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$ compares the so-called Gram matrices of the feature maps.

Following [99], the perceptual loss is computed by

$$\mathscr{L}_{\mathrm{id}}^{\mathrm{per}} = \sum_{l \in \mathcal{L}} \frac{1}{H^{[l]} W^{[l]} C^{[l]}} \left[\left\| \tilde{\Theta}_{\mathrm{F}}^{[l]}(\hat{\boldsymbol{I}}_{\mathrm{f}}) - \tilde{\Theta}_{\mathrm{F}}^{[l]}(\boldsymbol{I}_{\mathrm{gt}}) \right\|_1 + \left\| \tilde{\Theta}_{\mathrm{F}}^{[l]}(\hat{\boldsymbol{I}}_{\mathrm{f,r}}) - \tilde{\Theta}_{\mathrm{F}}^{[l]}(\boldsymbol{I}_{\mathrm{gt}}) \right\|_1 \right], \tag{4.16}$$

where $\tilde{\Theta}_{\mathrm{F}}^{[l]}(\boldsymbol{I})$ denotes the feature map at depth l of the feature extractor $\mathrm{F}(\cdot)$ with the input $\boldsymbol{I}$ before applying the activation function, and $\mathcal{L}$ is a set of depths l. Following Liu *et al.* [127], $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ is computed for $\hat{\boldsymbol{I}}_{\mathrm{f}}$ and $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$. In this way, additional focus is set onto the occluded area, while the non-occluded area is still considered due to $\hat{\boldsymbol{I}}_{\mathrm{f}}$. Moreover, a mix of low-level and high-level feature information is compared by setting $\mathcal{L} = \{10, 40\}$, *i.e.*, after the ResNet-block "conv2_x" and "conv4_x" (*cf.* Table 3.4).

The perceptual loss $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ is often complemented by the style loss

$$\begin{aligned}\mathscr{L}_{\mathrm{id}}^{\mathrm{style}} = \sum_{l \in \mathcal{L}} \frac{1}{H^{[l]} W^{[l]} C^{[l]}} \Bigg[&\left\| \Gamma\left(\tilde{\Theta}_{\mathrm{F}}^{[l]}(\hat{\boldsymbol{I}}_{\mathrm{f}})\right) - \Gamma\left(\tilde{\Theta}_{\mathrm{F}}^{[l]}(\boldsymbol{I}_{\mathrm{gt}})\right) \right\|_1 + \\ &\left\| \Gamma\left(\tilde{\Theta}_{\mathrm{F}}^{[l]}(\hat{\boldsymbol{I}}_{\mathrm{f,r}})\right) - \Gamma\left(\tilde{\Theta}_{\mathrm{F}}^{[l]}(\boldsymbol{I}_{\mathrm{gt}})\right) \right\|_1 \Bigg],\end{aligned} \tag{4.17}$$

where the Gram matrix $\Gamma(\boldsymbol{F})$ of a feature tensor $\boldsymbol{F} \in \mathbb{R}^{H \times W \times C}$ is defined as

$$\Gamma(\boldsymbol{F}) = \boldsymbol{G}^{\mathsf{T}} \boldsymbol{G}. \tag{4.18}$$

Here, the matrix $\boldsymbol{G} \in \mathbb{R}^{HW \times C}$ corresponds to $\boldsymbol{F}$ reshaped to a matrix. Thus, the Gram matrix $\Gamma(\boldsymbol{F}) \in \mathbb{R}^{C \times C}$ of a feature tensor $\boldsymbol{F}$ describes the correlation between different feature maps, *i.e.*, whether different channels tend to activate together at similar positions. This becomes even more apparent when considering $\Gamma(\boldsymbol{F})$ at position i, j

$$\big[\Gamma(\boldsymbol{F})\big]_{i,j} = \sum_{x=1}^{H} \sum_{y=1}^{W} \big[\boldsymbol{F}\big]_{x,y,i} \big[\boldsymbol{F}\big]_{x,y,j}. \tag{4.19}$$

In this way, the style loss $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$ promotes similar feature map correlations, whereas $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ naively minimizes their difference. Moreover, $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$ can be employed to mitigate checkerboard artifacts caused in the decoder by transposed convolutions, as shown in image inpainting and style transfer tasks [127, 139]. Still, multiple image inpainting approaches [20, 28, 89, 121, 123, 196, 284] exclusively rely on $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ and do not incorporate $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$. This trend is particularly noticeable in face completion tasks.

4.6 Experiments

4.6.1 Training Details

As mentioned in Section 4.4.4, a pretrained face feature extractor $\mathrm{F}(\cdot)$ is employed, from which the feature maps for the identity losses $\mathscr{L}_{\mathrm{id}}$ are extracted. During the entire training of the face completion network, the weights of $\mathrm{F}(\cdot)$ are frozen.

Similar to the training of $\mathrm{F}(\cdot)$, the VGGFace2 dataset is utilized, which comprises 3.1M images of 8631 identities (see also Table 3.1). All faces are aligned according to the custom FAP introduced in Section 3.2 with the facial landmarks predicted by the MTCNN [265] and are then cropped to a resolution of 112×112 pixels. In order to

guarantee accurate crops for the inputs of the discriminators, the facial landmarks of the faces after alignment are also loaded.

While $\mathrm{F}(\cdot)$ was trained with the augmentations listed in Section 3.5.1, all color augmentations are discarded as the occlusions provide sufficient variations to avoid overfitting. Still, horizontal flipping is employed with a probability of $p_{\text{aug}} = 0.5$. Since occlusions constitute the central aspect of this task, faces are occluded with $p_{\text{aug}} = 0.9$. During training, the occlusion's size and center position $\boldsymbol{c}_{\text{o}}$ are selected more freely than introduced in Section 4.1. The area and height ratios are sampled from a uniform distribution $\kappa_{\text{a}} \sim \mathcal{U}_1(0.01, 0.15)$ and $\kappa_{\text{h}} \sim \mathcal{U}_1(0.05, 0.5)$. Besides, the occlusions' center $\boldsymbol{c}_{\text{o}}$ is not limited to any of the four regions but sampled from $\left[\boldsymbol{c}_{\text{o}}\right]_i \sim \mathcal{U}_1(0.1r, 0.9r)$. By excluding the borders, the occlusions primarily cover the face. All remaining parameters (shape, form, and color) are set as listed in Table 4.1.

The training of the blind face completion network is divided into two steps. First, the coarse network is pretrained for one epoch with a learning rate $\eta = 2 \cdot 10^{-4}$ and batch size $N_{\text{b}} = 8$. After pretraining, a rough estimate $\hat{\boldsymbol{I}}_{\text{c}}$ together with an accurate prediction of the mask $\hat{\boldsymbol{M}}$ is obtained, which eases the subsequent training of the fine network. Only pixel-wise losses focusing on the occluded area are considered for the pretraining with $\lambda_{\text{pix}}^{\text{o}} = 3$ and $\lambda_{\text{pix}}^{\text{no}} = \lambda_{\text{pix}}^{\text{m}} = 1$. Thus, the coarse network is not pretrained as part of a GAN.

For the GAN training, the weights in the coarse network are initialized with the values from the pretraining. To account for random weights in the fine network and the discriminators, their initial learning rates are set to $\eta_{\text{fine}} = \eta_{\text{D}} = 10^{-4}$, while the initial learning rate in the coarse network is substantially lower $\eta_{\text{coarse}} = 5 \cdot 10^{-5}$. The GAN is trained for five epochs in an alternating manner, *i.e.*, a batch is either used to train only the generator or only the discriminator. Furthermore, the discriminator's weights are fixed when training the generator and vice versa. After every epoch, the learning rate η is reduced by a factor of $\gamma_{\text{lr}} = 4$. The total loss of the generator $\mathscr{L}_{\text{tot}}^{\text{G}}$ (see Equation (4.10)) is balanced using the following factors $\lambda_{\text{pix}}^{\text{o}} = 3$, $\lambda_{\text{pix}}^{\text{no}} = 1$, $\lambda_{\text{pix}}^{\text{m}} = 10$, $\lambda_{\text{adv}}^{\text{G}} = 1$, $\lambda_{\text{id}}^{\text{per}} = 0.1$, and $\lambda_{\text{id}}^{\text{style}} = 240$. Note that by increasing the weight of the pixel-wise similarity loss for the mask $\lambda_{\text{pix}}^{\text{m}}$ by a factor of 10 compared to pretraining, the network is forced to ensure that the mask prediction remains reliable. $\lambda_{\text{id}}^{\text{per}}$ and $\lambda_{\text{id}}^{\text{style}}$ are chosen as in [127]. All remaining training parameters, including those related to the generation of occlusion, are identical to the pretraining of the coarse network.

4.6.2 Evaluation Details

The FR performance is evaluated using two benchmark datasets (see also Table 3.2): The LFW dataset [86] for face verification and the MegaFace dataset [106] for closed-set face identification. For both protocols, faces need to be synthetically occluded in order to measure the quality of the reconstruction. The occlusion parameters listed in Table 4.1 are controlled by a seed to obtain a random albeit deterministic augmentation. To cope with the vast amount of different settings, the influence of text occlusion is evaluated for height ratios $\kappa_{\text{h}} \in \{0.1, 0.3, 0.5\}$, whereas rectangular occlusions are investigated for area

ratios $\kappa_\mathrm{a} \in \{0.05, 0.1, 0.15\}$ if not stated otherwise. Furthermore, all four regions (eyes, nose, mouth, outside) are considered, resulting in overall 24 occlusion configurations. Xia [31$^+$] demonstrated that the blind face completion models are robust against color variations. Thus, color and form, are selected at random yet in a deterministic way.

The occlusion scheme for the LFW dataset was introduced for the preliminary analysis in Section 4.2, where the face pairs are always occluded similarly. Even though the position is fixed, it only refers to the region (eyes, nose, mouth, and outside) and permits a different center $\boldsymbol{c}_\mathrm{o}$ for every occlusion as long it is within the selected region. Thus, the cross-occlusion scenario of shape, size, and region varying between the two pairs is not considered.

The MegaFace benchmark comprises a probe set, which consists of 3530 images of 80 identities from the FaceScrub dataset [163], and up to $N_\mathrm{d} = 10^6$ distractor images from 690k identities within a gallery $\mathcal{G}$. By adding a single image from the probe set to the gallery set, all remaining images of the same identity are matched with the entire gallery $\mathcal{G}$ to evaluate the face identification performance for a varying gallery size $|\mathcal{G}| = N_\mathrm{d} + 1$. While for the LFW dataset, both faces from a pair are synthetically occluded, only the images from FaceScrub are occluded for the MegaFace benchmark. Thus, the probe set only contains occluded faces, whereas the gallery set $\mathcal{G}$ consists of N_d non-occluded distractors and a single occluded image. Therefore, the difficulty of the benchmark is increased substantially since the single feature of the occluded image enrolled in the gallery must prevail against N_d accurate features from non-occluded faces. All masks were published by Hörmann *et al.* [13†].

The metrics to evaluate face verification and face identification performance are computed as introduced in Sections 3.5.2.1 and 3.5.2.2.[iii] In order to evaluate the reconstruction quality, the *peak signal-to-noise ratio* ($PSNR$) is computed according to

$$PSNR = 10 \log_{10} \left(\frac{2^2}{\|\hat{\boldsymbol{I}}_\mathrm{f,r} - \boldsymbol{I}_\mathrm{gt}\|^2} \right), \tag{4.20}$$

where the images are assumed to have the value range $[-1, 1]$. In contrast to the pixel-wise $PSNR$, the *structural similarity* ($SSIM$) [226] offers a more comprehensive analysis of visual similarity as it comprises multiple factors accounting for luminance, contrast, and structure.

Moreover, the accuracy of the reconstruction is evaluated by extracting 196 facial landmarks [62] and computing the normalized mean squared error (MSE)

$$MSE_\mathrm{n} = \frac{\frac{1}{196} \sum_{i=1}^{196} \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2}{\|\boldsymbol{x}_{38} - \boldsymbol{x}_{88}\|_2}, \tag{4.21}$$

[iii] In contrast to the analysis in Section 3.6, the original MegaFace benchmark, *i.e.*, without the refinement introduced by Deng *et al.* [33], was used to match the results obtained by Hörmann *et al.* [13†]. Thus, the baseline results on MegaFace do not correspond to the results obtained in Section 3.6 by the VGG-112 model.

Table 4.2: Ablation study on FR performance using $TPIR$ [%] at rank $R = 1$ on the MegaFace benchmark [106] with $N_\mathrm{d} = 10^6$ and Acc [%] on the LFW benchmark [86]. All metrics are averaged over four positions and three sizes per shape (*cf.* Section 4.6.2). † denotes that $\mathscr{L}_\mathrm{pix}^\mathrm{o}$ was also computed on $\hat{\boldsymbol{I}}_\mathrm{f}$. The highlighted model is analyzed in detail in the following subsections.

Losses			Attention		MegaFace $N_\mathrm{d} = 10^6$				LFW		
$\mathscr{L}_\mathrm{id}^\mathrm{per}$	$\mathscr{L}_\mathrm{id}^\mathrm{style}$	$\mathscr{L}_\mathrm{pix}^\mathrm{(n)occ}$	Self	Cross	non-occluded	Rectangle	Text	Avg	Rectangle	Text	Avg
VGG-112					52.32	29.81	28.88	29.34	97.65	97.18	97.41
VGG-112-o					54.40	42.38	44.66	43.52	98.85	99.05	98.95
BVMR [77]					52.32	29.48	43.61	36.55	97.75	99.04	98.40
Coarse Network					52.32	31.63	44.89	38.26	97.95	99.08	98.52
✓	✓	✓	✓	✓	52.32	37.18	47.50	42.34	98.61	99.18	98.90
✓	✗	✓	✓	✓	52.32	37.43	47.53	42.48	98.60	99.17	98.88
✗	✓	✓	✓	✓	52.32	36.98	47.35	42.17	98.62	99.23	98.93
✗	✗	✓†	✓	✓	52.32	36.08	46.89	41.49	98.54	99.18	98.86
✓	✗	✗	✓	✓	52.32	36.53	47.24	41.89	98.56	99.20	98.88
✓	✗	✓†	✓	✓	52.32	37.09	47.43	42.26	98.64	99.22	98.93
✓	✗	✓	✗	✓	52.32	35.05	46.91	40.98	98.44	99.17	98.80
✓	✗	✓	✓	✗	52.32	37.19	47.42	42.31	98.63	99.21	98.92

where $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}_i$ denote the ith facial landmark of the ground-truth image $\boldsymbol{I}_\mathrm{gt}$ and the reconstructed image $\hat{\boldsymbol{I}}_\mathrm{f,r}$, respectively. The 38th and the 88th landmark correspond to the left and right pupil and are used to determine the intraocular distance. By normalizing MSE_n with the intraocular distance, the metric becomes invariant to different face sizes. This normalization is popular for evaluating facial landmark detection approaches.

4.7 Results

During evaluation, two primary baselines are considered. For a fair comparison, a blind face completion method, the BVMR model by Hertz *et al.* [77], was trained on the VGGFace2 dataset with the same augmentation scheme as all other models. The remaining parameters are identical to the parameters reported by the authors. Furthermore, a similar model to VGG-112 is trained, which follows the same occlusion data augmentation as when training the blind face completion models. This model is named *VGG-112-o*. Even though VGG-112-o does not perform any face completion, it still provides valuable insights into the performance of a straightforward method if face completion is not required.

4.7.1 Ablation Study

4.7.1.1 Quantitative Results

An ablation study is performed in terms of FR performance and reconstruction quality to obtain the best-performing model and analyze the impact of different training parameters. The results are reported in Tables 4.2 and 4.3, respectively.

Table 4.3: Ablation study on reconstruction quality using $PSNR$ and $SSIM$ on MegaFace. All metrics are averaged over four positions and three sizes per shape (*cf.* Section 4.6.2). † denotes that $\mathscr{L}_{\text{pix}}^{\text{o}}$ was also computed on $\hat{\boldsymbol{I}}_{\text{f}}$. The highlighted model is analyzed in detail in the following subsections.

Losses			Attention		$PSNR$			$SSIM$		
$\mathscr{L}_{\text{id}}^{\text{per}}$	$\mathscr{L}_{\text{id}}^{\text{style}}$	$\mathscr{L}_{\text{pix}}^{\text{(n)occ}}$	Self	Cross	Rectangle	Text	Avg	Rectangle	Text	Avg
Occluded					19.95	20.89	20.42	0.8665	0.8388	0.8526
BVMR [77]					31.29	39.07	35.18	0.9406	0.9721	0.9563
Coarse Network					33.11	39.48	36.30	0.9505	0.9739	0.9622
✓	✓	✓	✓	✓	33.44	39.94	36.69	0.9533	0.9743	0.9638
✓	✗	✓	✓	✓	33.42	39.94	36.68	0.9535	0.9743	0.9639
✗	✓	✓	✓	✓	33.43	39.93	36.68	0.9530	0.9739	0.9634
✗	✗	✓†	✓	✓	34.25	40.40	37.33	0.9566	0.9743	0.9655
✓	✗	✗	✓	✓	33.25	39.70	36.47	0.9523	0.9742	0.9632
✓	✗	✓†	✓	✓	34.32	40.47	37.39	0.9573	0.9751	0.9662
✓	✗	✓	✗	✓	33.25	39.81	36.53	0.9521	0.9735	0.9628
✓	✗	✓	✓	✗	33.36	39.87	36.61	0.9531	0.9739	0.9635

The drop in FR performance of VGG-112 for occluded faces observed in Section 4.2 is also apparent in Table 4.2 on the MegaFace benchmark from 52.32% to 29.34%. Besides, it is reflected by the $PSNR$ and $SSIM$ in Table 4.3. Note that on MegaFace, the true positive identification rate ($TPIR$), and $PSNR$ are very similar for rectangular and text occlusion. Thus, the evaluation parameters introduced in Section 4.6.2 create occlusions of different shapes, which are perceived as equally challenging for an FR model that is unfamiliar with occlusions. For non-occluded faces, no drop in $TPIR$ is observed. Hence, non-occluded faces are untouched by the blind face completion networks, and thus alterations are limited to the occluded areas.

Reconstructing the face clearly helps in mitigating this drop as the average $TPIR$ is restored to 36.55% and 38.26% by the BVMR model and the pretrained coarse network, respectively. The superior performance of the coarse network compared to the BVMR model is also apparent when considering reconstruction quality in Table 4.3. Thus, despite the lower parameter count (4.1M compared to 20.5M), the coarse network outperforms the BVMR model. Hörmann *et al.* [13†] conjectured that this is mainly due to the DMFB in the coarse network.

The addition of the fine network consistently provides superior results on all metrics regardless of the exact parameter configuration. Even though it is crucial to employ identity losses $\mathscr{L}_{\text{id}}$, the differences between employing $\mathscr{L}_{\text{id}}^{\text{style}}$ or $\mathscr{L}_{\text{id}}^{\text{per}}$ are only noticeable in the FR performance with minor advantages of solely using $\mathscr{L}_{\text{id}}^{\text{per}}$. This is in accordance with many image inpainting researchers [20, 28, 89, 121, 123, 196, 284] who employed $\mathscr{L}_{\text{id}}^{\text{per}}$ and discarded $\mathscr{L}_{\text{id}}^{\text{style}}$. Without any identity losses, the training only converges if an additional pixel-wise similarity loss is employed for the occluded area of $\hat{\boldsymbol{I}}_{\text{f,r}}$. Besides, the FR performance drops substantially, which demonstrates the necessity of using at least one identity loss.

Even though pixel-wise similarity losses $\mathscr{L}_{\text{pix}}^{(\text{n})\text{o}}$ are employed solely for the coarse reconstruction, they improve the FR performance and reconstruction quality of the refined reconstruction. Adding another loss $\mathscr{L}_{\text{pix}}^{\text{o}}$ on the prediction of the fine network $\hat{\boldsymbol{I}}_{\text{f,r}}$ slightly mitigates the FR performance yet increases reconstruction quality. However, one must note that $PSNR$ is computed similarly to $\mathscr{L}_{\text{pix}}^{\text{o}}$ without any spatial dependencies. Thus, a blurrier reconstruction would help in accomplishing a better $PSNR$. To fully evaluate this apparent trade-off between FR performance and reconstruction quality, a qualitative analysis is required.

Besides the influence of the loss function, it is apparent that the dual attention structure is vital for the best performance. When considering both attention blocks separately, it is clear that the self-attention block has a more significant impact on the FR performance and reconstruction quality than the cross-attention block. Still, a slight improvement is observed when adding the cross-attention block.

The FR performance for VGG-112-o, *i.e.*, the FR network trained on occluded faces, demonstrates that superior results are accomplished on average without prior face completion. This is no surprise since the face feature extractor $\text{F}(\cdot)$ of the blind face completion network was not optimized to cope with reconstructed faces $\hat{\boldsymbol{I}}_{\text{f,r}}$ and is still identical to VGG-112. Hence, $\text{F}(\cdot)$ was never exposed to occlusions and assumes that information originating from every pixel is equally reliable. In contrast to $\text{F}(\cdot)$, VGG-112-o only extracts information from non-occluded – *i.e.*, reliable – pixels. However, while VGG-112-o surpasses all face completion approaches for rectangular occlusions, face completion algorithms perform the best results for text occlusion, demonstrating that face completion excels for sparse activations. Moreover, face completion always provides the reconstructed face $\hat{\boldsymbol{I}}_{\text{f,r}}$ and a prediction of the mask $\hat{\boldsymbol{M}}$ as additional outputs.

This quantitative analysis demonstrated that face completion cuts in half the drop in FR performance when faces are tampered with occlusions. At the same time, no unwanted artifacts are introduced into the non-occluded regions, which shows that prior face completion can always be performed and does not create any downsides except for the additional processing time.

4.7.1.2 Qualitative Results

Figure 4.10 illustrates the qualitative results of the proposed coarse-to-fine network in comparison with the coarse network and the BVMR model [77]. It is apparent that the coarse network and the BVMR model struggle when exposed to rectangular occlusions. Since the results for text occlusions are acceptable, a dependency of the reconstruction quality on the distance to the closest reliable pixel is identified. Sparse text occlusions do not require long-distance information transfer, leading to overall superior performance compared to dense rectangular occlusions.

The results of the coarse-to-fine network show an increase in realism of the reconstructed areas, which is primarily attributed to the adversarial training and the dual attention structure. Even under adverse conditions, *e.g.*, glasses, head poses, and intense beards, the networks generated remarkable reconstructions without any unrealistic mismatch in eye colors as in [260, 284].

Figure 4.10: Qualitative results of selective models with the occluded input $\boldsymbol{I}_{\mathrm{o}}$ (first row) and the ground-truth image $\boldsymbol{I}_{\mathrm{gt}}$ (last row). The size set to as $\kappa_{\mathrm{a}} = 0.1$ and $\kappa_{\mathrm{h}} = 0.3$. The models using the fine network were trained with either $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$ or $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$. Hence, the model marked by $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ is highlighted in Tables 4.2 and 4.3. † denotes that $\mathscr{L}_{\mathrm{pix}}^{\mathrm{o}}$ was also computed on $\hat{\boldsymbol{I}}_{\mathrm{f}}$.

The occlusion of the mouth region also illustrates the ambiguity of the task, which was introduced in Section 4.5.1. While the coarse network and the BVMR model produce blurry results, the coarse-to-fine networks provide realistic results even though they differ from the ground-truth image $\boldsymbol{I}_{\mathrm{gt}}$. *E.g.*, lipstick was not applied in the reconstruction; however, the reconstruction still fits and is realistic without lipstick. Moreover, two additional examples are depicted in which the mouth is open in $\boldsymbol{I}_{\mathrm{gt}}$ while it is closed in the reconstruction $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$, and vice versa. In the case of applying lipstick, the model seems to favor not reconstructing a mouth with applied lipstick – most likely due to the lack of training samples and gender bias. In contrast, the different reconstructions of the mouth indicate no mode collapse of the GAN. However, extreme facial expressions, as in the last column of Figure 4.10, provoke reconstruction artifacts.

Between the three models, minor differences become apparent only upon closer inspection. The reconstructions by the model, which was trained with $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$ and thus without $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$, contain more unrealistic textures and artifacts. Using $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$ instead of $\mathscr{L}_{\mathrm{id}}^{\mathrm{style}}$ reduces these artifacts. The additional L_1 loss on the occluded pixels in $\hat{\boldsymbol{I}}_{\mathrm{f}}$ suppresses the artifacts creating a smoother yet more realistic reconstruction. This finding confirms the analysis in the previous section, which found that this additional L_1 controls the trade-off between FR performance and reconstruction quality. The overall best reconstruction by the model denoted with $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}\dagger}$ is accompanied by slightly inferior FR performance compared to the model marked with $\mathscr{L}_{\mathrm{id}}^{\mathrm{per}}$.

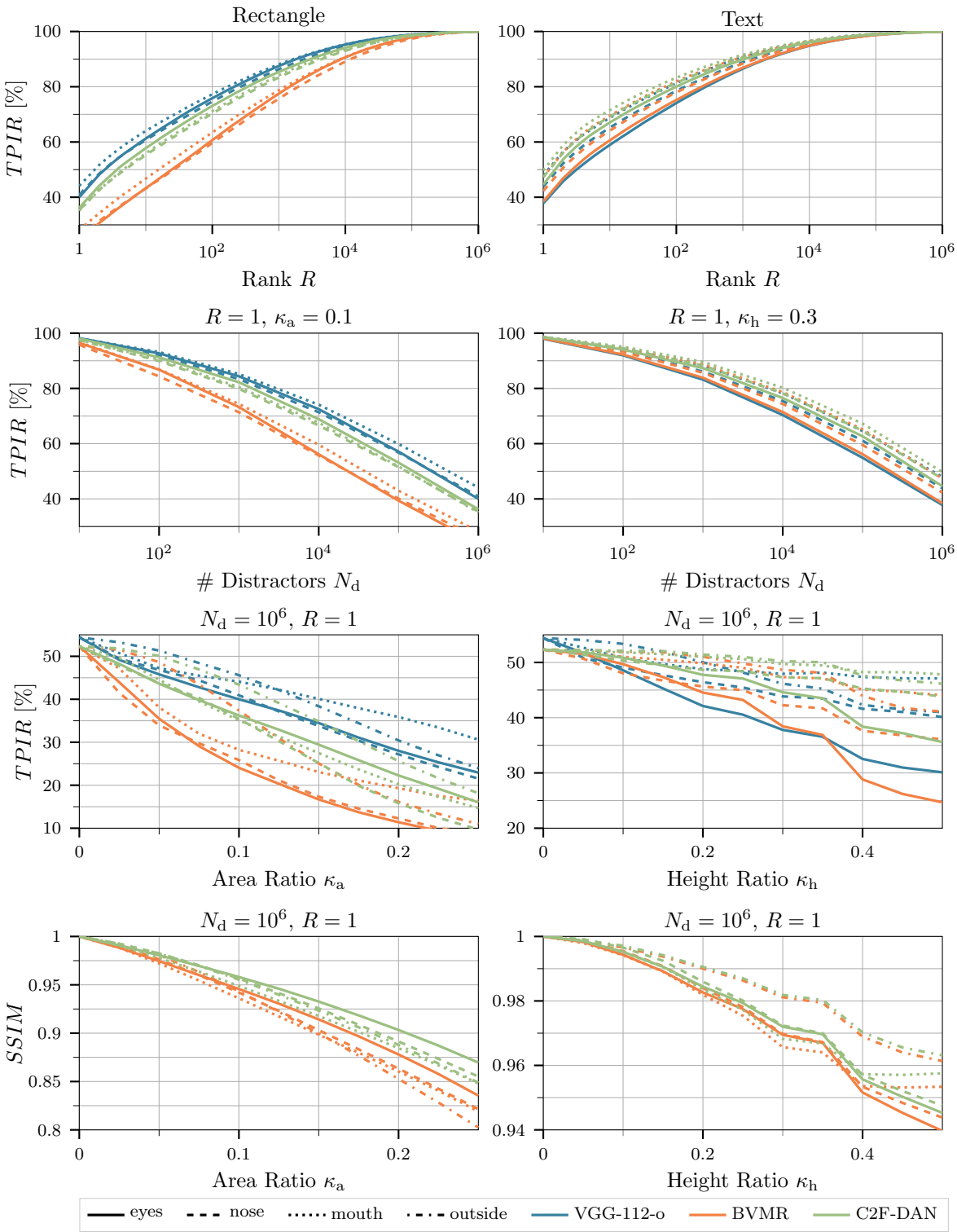

Figure 4.11: Overview of the robustness of the coarse-to-fine dual attention network (C2F-DAN) and the baselines in terms of rank R, number of distractors N_d, size κ_a and κ_h, and position on the MegaFace benchmark [106]. Adapted from [13†].

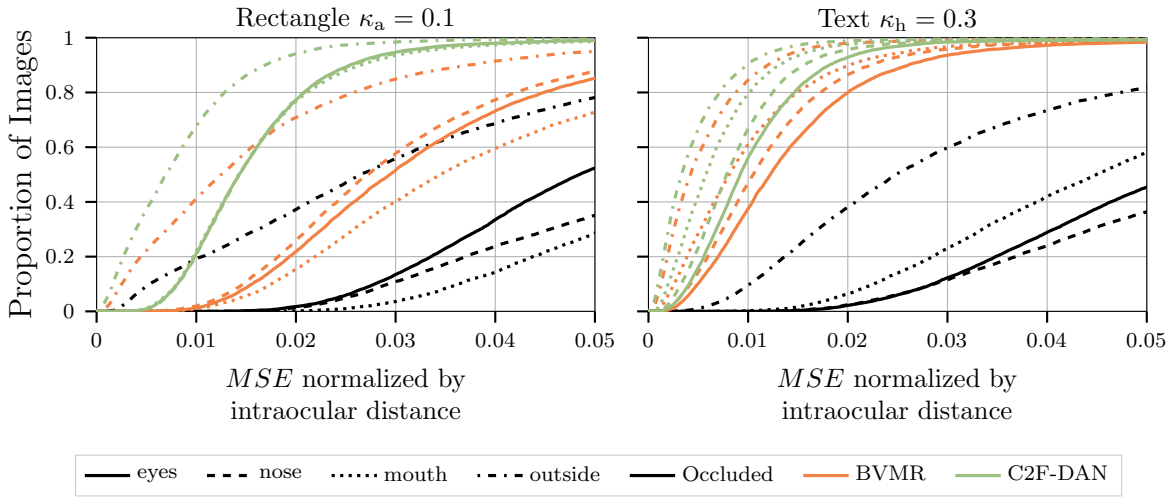

Figure 4.12: Effect of occlusion and reconstruction on the accuracy of facial landmark prediction on the LFW dataset [86].

4.7.2 Detailed Analysis

The ablation study in Section 4.7.1 identified a trade-off between FR performance and reconstruction quality. Since the focus of this work lies on FR performance, the model trained with $\mathscr{L}_{\text{id}}^{\text{per}}$ and without style loss $\mathscr{L}_{\text{id}}^{\text{style}}$, which is highlighted in Tables 4.2 and 4.3, is selected for the upcoming in-depth analysis and referred to as the *coarse-to-fine dual attention network* (C2F-DAN).

4.7.2.1 Rank R and Number of Distractors N_{d}

Figure 4.11 depicts an exhaustive analysis of the C2F-DAN in terms of FR dependent on shape, size, and position of the occlusions. The first two columns in Figure 4.11 show similar behavior for $TPIR$ as the general FR approaches in Section 3.6.2.2. Overall, the averaged results from Table 4.2 are confirmed. The C2F-DAN outperforms BVMR consistently regardless of the occlusions' shape, whereas the C2F-DAN obtains superior $TPIR$ compared to VGG-112-o only for text occlusions.

4.7.2.2 Influence of Position and Size κ

For both occlusion shapes, the dependency of $TPIR$ on the position and size follows the preliminary analysis in Section 4.2. Rectangular occlusions reveal a clear picture of the influence caused by the occlusions' position. As expected, occlusions outside the face only slightly harm FR performance and thus also yield the best $TPIR$ after reconstruction. This is confirmed by the reconstruction quality since worse $SSIM$ is observed for occlusion at the borders. Hence, the superior $TPIR$ for the borders is not due to impressive reconstruction quality but rather caused by the irrelevance of the background when identifying a face. Interestingly, VGG-112-o provides superior results if the rectangular occlusion is centered around the mouth. This is also observed for the

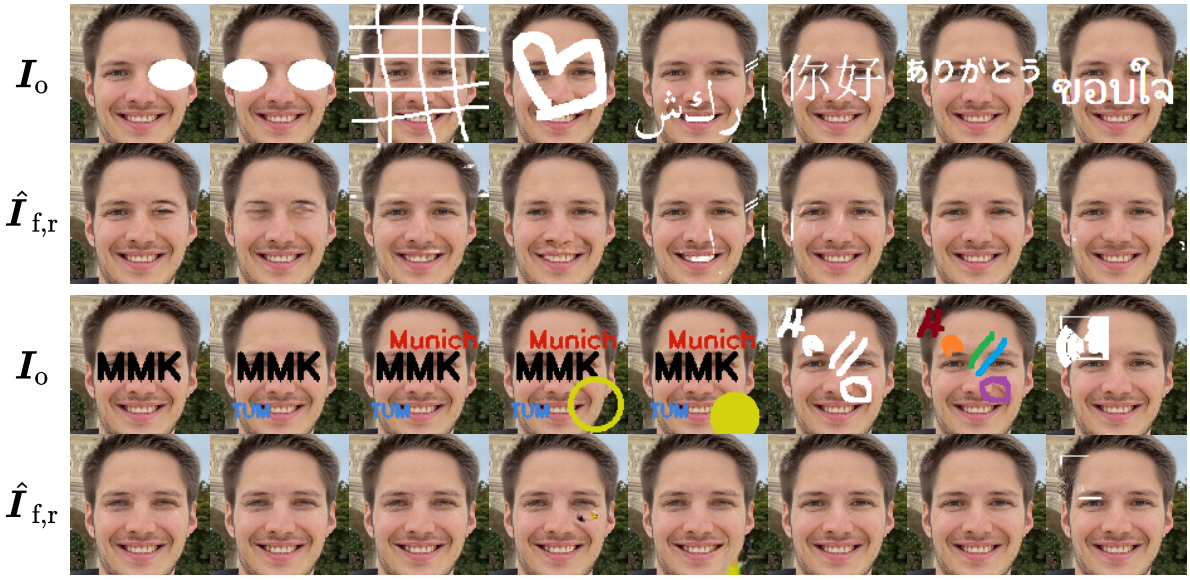

Figure 4.13: Qualitative results $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ of the C2F-DAN for faces tampered with occlusions $\boldsymbol{I}_{\mathrm{o}}$, which were not part during training.

BVMR, where a larger $TPIR$ is obtained from reconstructing the mouth regions. In contrast, the C2F-DAN considers the mouth and eyes as equally difficult, whereas the nose is perceived as the most challenging region – probably due to the central occlusion within the face. In contrast to BVMR and VGG-112-o, the slightly elevated $TPIR$ on eyes compared to mouth and nose suggests that the C2F-DAN successfully transfers information from the non-occluded eye to reconstruct the occluded eye.

Text occlusions reveal an impressive gap between the C2F-DAN and the remaining approaches. Besides, it is apparent that all approaches struggle with occlusions covering the eyes, which is mainly attributed to the wide form of the text. Thus, for large height ratios κ_{h} it is likely that both eyes are occluded, which inevitably leads to a drop in $TPIR$. The $SSIM$ provides a clear preference for the C2F-DAN, particularly for rectangular occlusions.

4.7.2.3 Reconstruction Accuracy

Hörmann *et al.* [13†] proposed to leverage facial landmarks to measure the accuracy of the reconstruction. By computing the MSE normalized by the intraocular distance according to Equation (4.21), they obtain a metric describing the accuracy of the position and the form of the reconstructed face parts. Ideally, the position of the facial landmarks of the reconstructed image $\hat{\boldsymbol{I}}_{\mathrm{f,r}}$ and the ground-truth image $\boldsymbol{I}_{\mathrm{gt}}$ should be identical. However, similar to $PSNR$ and $SSIM$, the MSE does not address the ambiguity problem.

Figure 4.12 depicts the cumulative distribution of the normalized MSE. The C2F-DAN provides a more accurate reconstruction, particularly for rectangles. Besides, this analysis further shows that prior face completion is vital to obtain accurate facial landmarks as facial landmark detectors are very susceptible to occlusions.

Figure 4.14: Attention map of the patch-wise cross-attention block. Colors illustrate the information transfer, *i.e.*, a red patch in the source $\boldsymbol{V}$ is used for the red position(s) in the target $\boldsymbol{R}$. The information transfer is visualized on $\boldsymbol{I}_{\text{gt}}$ even though takes place in feature maps. Still if the reconstruction looks visually appealing in the image space, it is likely that relevant information is considered in the feature space.

4.7.2.4 Extension to Arbitrary Shapes

Figure 4.13 depicts a wide range of occlusions which were never seen by the network during training. The reconstructed images show the impressive generalization capabilities and limitations of the C2F-DAN. It is evident that the network recognizes different forms (circles) or free-form shapes. However, covering both eyes with occlusion constitutes a challenging problem as no information can be transferred from a potentially non-occluded eye. Also, grids are reconstructed satisfyingly by the C2F-DAN. Occluding the face with text composed of non-Latin letters reveals that the C2F-DAN did not overfit on Latin letters during training. The artifacts created by Arabic and Chinese letters suggest that thin lines impede a reliable mask detection. Since thin lines were rare during training, extending the occlusion generation to incorporate lines could alleviate these limitations.

The second row of Figure 4.13 reveals that the C2F-DAN handles an arbitrary number of occlusions in random colors as long as they do not overlap or cover a large part of the face. This is particularly impressive since the C2F-DAN was only exposed to a single occlusion with a homogeneous color. Artifacts in the reconstruction occur if occlusions overlap. However, this can also be addressed by extending the augmentation scheme to encompass overlapping occlusions. Overall, Figure 4.13 highlights the astounding generalization capability of the C2F-DAN.

4.7.2.5 Attention Map of the Patch-Wise Cross-Attention Block

Figure 4.14 illustrates the most similar patches according to $\boldsymbol{S}$, which are used to reconstruct $\boldsymbol{R}$ from $\boldsymbol{V}$. The patch-wise cross attention block mostly leverages information from the non-occluded eye. This is particularly noticeable for the eye brows and the inner eye regions. Information for the pixels surrounding the eye are pooled from the entire face, which is reasonable. The swapped faces demonstrate that the cross-attention block leverages relevant information to rebuild the eye in the feature space. Hence. the patch-wise cross-attention block operates as intended.

4.8 Conclusion and Future Work

This chapter introduced a novel approach for blind face reconstruction utilizing a coarse-to-fine network with a dual attention structure (C2F-DAN). The coarse-to-fine network effectively splits the reconstruction task into two parts: 1) create a rough estimate of the occluded area and an accurate prediction of the mask; and 2) refine the rough estimate to obtain a realistic reconstruction. While the DMFB ensures that the coarse network has a large receptive field, the parallel structure of two attention modules that complement each other makes the network incorporate global information. The patch-wise cross-attention block captures the underlying structure and promotes the transfer of textures, whereas the pixel-wise self-attention block performs unrestricted information transfer.

Multiple loss functions were employed to obtain the desired results. Pixel-wise similarity losses $\mathscr{L}_{\text{pix}}$ ensure a rough estimate by the coarse network together with an accurate mask prediction. A realistic refinement is obtained by training the C2F-DAN embedded into a GAN to leverage the adversarial loss $\mathscr{L}_{\text{adv}}$. In total, seven discriminators focus on distinct parts of the face and ensure crisp details in the reconstruction. Identity losses $\mathscr{L}_{\text{id}}$ based on feature maps from a pretrained feature extractor guide the C2F-DAN to generate coherent identity features in the reconstructed areas.

The exhaustive ablation study revealed a trade-off between FR performance and reconstruction quality. By adding another pixel-wise similarity loss on the occluded area of the refined reconstructed image $\hat{\boldsymbol{I}}_{\text{f,r}}$, the reconstruction quality is improved at the cost of FR accuracy. Besides, the perceptual loss $\mathscr{L}_{\text{id}}^{\text{per}}$ is the preferred identity loss and the parallel attention structure is vital for the training success. Moreover, the analysis showed that the patch-wise cross-attention block operates as expected by replacing occluded patches with similar patches from the non-occluded region.

The in-depth analysis on the MegaFace benchmark demonstrated that the C2F-DAN excels when exposed to sparse occlusion outperforming the BVMR model and the baseline trained on occlusions (VGG-112-o). For rectangular occlusions, the VGG-112-o can better focus on the reliable pixels and ignore the occluded pixels. In contrast, the feature extractor in the C2F-DAN cannot tell whether the information originating from a single pixel is reliable since it was never trained with occlusions, resulting in overall inferior FR performance. Still, the C2F-DAN provides the reconstructed image $\hat{\boldsymbol{I}}_{\text{f,r}}$ and the mask $\hat{\boldsymbol{M}}$ as additional outputs. Regarding the occlusions' position, it can be stated that nose regions are most challenging for rectangular occlusions, whereas the eye regions constitute the most difficult position for text occlusions. Due the wide form of the text occlusion, both shapes have the highest probability that both eyes are occluded. Therefore, occluding both eyes is the most challenging scenario for face completion, whereas the occlusion of one eye is compensated by leveraging the information provided by the other eye. To conclude, the quantitative analysis confirms that the C2F-DAN successfully mitigates the drop in FR performance and thereby accomplishes the second objective of blind face completion.

The reconstruction accuracy indicated that the C2F-DAN closely approximates the underlying face parts and is very useful when extracting facial landmark positions of occluded faces. Furthermore, the abundance of distinct occlusions unseen during training

demonstrated that the C2F-DAN successfully adapts to unknown shapes and letters, and handles multiple occlusions of different colors. This analysis shows that a vast variety of occlusions are removed without tedious mask annotations since the C2F-DAN estimates the mask. Considering the highly realistic inpainting results and impressive generalization to unknown occlusions, the first objective of blind face completion is also achieved by the C2F-DAN.

The analysis also revealed several limitations of this work, which can be tackled in future works. The drop in FR performance due to occlusions can be further minimized by finetuning the face feature extractor $\mathrm{F}(\cdot)$ of the C2F-DAN to cope with the reconstructed faces. In this way, $\mathrm{F}(\cdot)$ learns to identify the pixels that contain reliable information and does not consider every pixel as equally important. This shortcoming can be remedied by incorporating modified partial convolutions [127]. The vanilla partial convolution sets the input pixel value to zero if it is encompassed by the occluded area. If partial convolutions were implemented in $\mathrm{F}(\cdot)$, this behavior would lead to similar performance as obtained by VGG-112-o as no information from the occluded area is considered and the face completion would become irrelevant. Subsequent works do not incorporate a binary mask and thus offer a more promising solution, *e.g.*, gated convolutions [249, 255], which learn to create layer-wise soft attention maps, or region-wise convolutions [139] with different weights for occluded and non-occluded pixels.

The C2F-DAN was trained with the VGG-112 model as a face feature extractor. However, the analysis in Section 3.6.3 illustrated that recent models outperform VGG-112 on FR. This is not a huge issue when analyzing the general behavior of FR models exposed to occluded faces since all FR models are expected to be affected similarly by occlusions. Nevertheless, incorporating richer features provided by recent FR models is expected to aid the reconstruction.

Another limitation becomes apparent in the qualitative results. Unrealistic reconstructions are obtained if both eyes are occluded, in extreme head poses or facial expressions. While the network would undoubtedly benefit from more training data encompassing such cases, increasing the focus of the adversarial loss $\mathscr{L}_{\mathrm{adv}}^{\mathrm{G}}$ on the occluded area may alleviate these issues. Besides increasing $\lambda_{\mathrm{adv}}^{\mathrm{G}}$, employing discriminators with only the occluded region as input would also shift the focus. Zhou *et al.* [284] demonstrated remarkable results with a similar discriminator structure.

Further extensions are conceivable in terms of the variety of occlusions. The generalization analysis illustrated that the C2F-DAN already handles a wide range of occlusions. Still, several limitations become apparent. The C2F-DAN often misses detections of thin occlusions and overlapping occlusions of distinct colors. Both can be coped with by diversifying the augmentation scheme to encompass such scenarios. Besides, masks are only detected for sharp edges. Thus, if antialiasing is applied afterward, the C2F-DAN struggles to detect the mask and thus entirely fails the reconstruction. Like overlapping masks, this scenario can be considered by extending the variety of the augmentations. As the next step, natural occlusions, *e.g.*, medical face masks or glasses, can be included. Recently, emerging works on face mask removal have shown astounding results [138, 250]; however, face mask removal with photos taken in the wild still leaves some artifacts due to the limited generalization of the model trained with synthetic occlusions [138].

5

Attentional Pooling for Partial Face Recognition

The previous chapter presented a blind face completion approach to cope with occluded faces, *i.e.*, part of the face $\boldsymbol{I}_{\text{gt}}$ is synthetically occluded by a mask $\boldsymbol{M}_{\text{gt}}$ of a random color $\boldsymbol{c}$. Partial face recognition (FR) is related to occluded FR since it also considers FR, where part of the face is occluded. However, partial FR constitutes a more challenging scenario since only a small part of the face is visible, whereas the occluded FR implies that only a (small) part of the face is occluded. Thus, partial FR is also referred to as the recognition of face patches.

As introduced for face completion in Equation (4.1), a partial face $\boldsymbol{I}_{\text{o}}$ is generated from a ground-truth face $\boldsymbol{I}_{\text{gt}}$ and a mask $\boldsymbol{M}_{\text{gt}}$ by

$$\boldsymbol{I}_{\text{o}} = (\mathbb{1}_{r,r} - \boldsymbol{M}_{\text{gt}}) \odot \boldsymbol{I}_{\text{gt}}, \tag{5.1}$$

where $\left[\boldsymbol{M}_{\text{gt}}\right]_{x,y} = 1$ denotes that the pixel position x, y is occluded.[i] In the domain of partial FR, non-occluded faces $\boldsymbol{I}_{\text{gt}}$ are typically referred to as *holistic* faces. Compared to Equation (4.1), Equation (5.1) does not consider distinct colors of the masked areas. Thus, the background is automatically set to gray to match the preprocessing, which provided an image $\boldsymbol{I}_{\text{gt}}$ with values $\in [-1, 1]$ (*cf.* Equation (3.16)). Besides, a gray background compared to a black background also softens the edges between the background and the non-occluded areas.

While occluded faces in Chapter 4 and partial faces are highly related, obtaining reliable FR results on both differ due to the varying size of occlusion. For synthetically occluded faces, only masks with an area ratio $\kappa_{\text{a}} \ll 0.5$ (*cf.* Equation (4.2)) are considered, whereas even faces with $\kappa_{\text{a}} > 0.9$, *e.g.*, only one eye of the face is visible, are addressed in partial FR. In such scenarios, reconstructing the faces as in Chapter 4 constitutes an incredibly challenging task. Moreover, after reconstruction, most pixels are artificially generated and thus unreliable. This needs to be considered during the FR as otherwise,

[i] The symbol $\boldsymbol{I}_{\text{o}}$ is used for partial faces to maintain consistency with Chapter 4 since occluded and partial faces are created similarly. Note that the ground-truth image $\boldsymbol{I}_{\text{gt}}$ corresponds to the aligned image $\boldsymbol{I}_{\text{aug}}$.

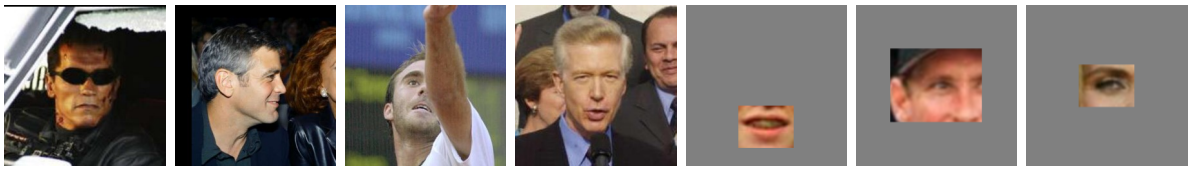

Figure 5.1: Examples of partial faces occurring in the Labeled Faces in the Wild (LFW) dataset [86] and synthetically generated partial faces.

the features become distorted by the overwhelming amount of unreliable information. Therefore, a direct FR approach, *i.e.*, recognition without reconstruction, is favored in partial FR. Still, advanced mechanisms are essential to aid the network in focusing on the non-occluded pixels, as straightforward data augmentation with partial faces does not suffice.

Figure 5.1 depicts partial faces in unconstrained conditions and the synthetic partial faces created with Equation (5.1). Thus, partial faces must be considered in unconstrained scenarios as information may be limited due to extreme head poses, occlusions of foreground objects, or if the face is cut off at the image's border. While natural and synthetic partial faces seem vastly different, both contain large areas with no identity information. Such faces complicate the recognition in multiple ways. On the one hand, information is scarce, and thus partial FR approaches must be able to extract meaningful identity information from a small patch. On the other hand, the non-occluded area varies, *i.e.*, the network must learn to extract information from distinct areas, *e.g.*, from the eye and from the mouth, and reach a decision whether two patches from different face regions belong to the same identity. Moreover, face alignment as introduced in Section 3.2 is often not feasible. Thus, the network must be able to handle face patches also at unusal positions in the input image, *e.g.*, the last image in Figure 5.1 shows an eye patch at the center of the image.

Therefore, the objective of partial FR is to obtain an algorithm that is robust to partial faces regardless of whether the partial faces are synthetically created or taken in the wild. Besides, the partial FR algorithm must be capable of comparing two non-overlapping face patches and should be invariant to the position of the patches within the image, while accomplishing comparable performance on holistic faces.

Robustness against partial faces is increased by encompassing them in training. However, by precisely designing the architecture and loss function to cope with the limited information inherent in partial faces the robustness further improves. Large parts of the work provided in this chapter were pre-published in [10†].

5.1 Related Work

As elaborated in Section 3.2, face detection and alignment are vital preprocessing steps in holistic FR. For partial FR, either the preprocessing must be modified to cope with partial faces, or the subsequent partial FR must be capable of handling images of varying scales and resolutions. The latter option requires highly sophisticated methods to learn

dependencies between face parts at various resolutions. Thus, most partial FR approaches rely on prior partial face detection [19, 142, 166] or manually create partial faces by synthetically masking holistic faces.

Partial FR has been investigated for decades, with the first approaches focusing on single face parts [67, 168, 186]. Sato *et al.* [186] proposed radial basis function networks to predict the identity given patches around the eye, ear, or nose, whereas Gutta *et al.* [67] considered face halves. Similarly, Park *et al.* [168] focused on the periocular region. However, they extracted features based on gradient orientation histogram and local binary patterns around multiple eye keypoints. Since the extraction of keypoints is feasible for partial faces of arbitrary size, various works [81, 124, 230] alleviated the limitation of fixed-size inputs of predefined face patches.

Liao *et al.* [124] represented every partial face with a set of scale-invariant feature transform (SIFT) keypoint descriptors and employed a sparse representation-based classification algorithm. SIFT descriptors were also extracted by Hu *et al.* [81]; however, they proposed a more efficient instance-to-class distance for matching. Weng et al. [230] combined SIFT and speeded up robust features (SURF) descriptors and utilized metric learning extended robust point matching method for recognition.

Owing to the emerging convolutional neural networks (CNNs), more powerful methods were proposed for partial FR. He *et al.* [75] designed a multiscale double supervision CNN, comprising 55 distinct CNNs. Every CNN is trained with different face patches at multiple scales for feature extraction by optimizing cross-entropy (CE) and triplet loss (see Equation (3.8)). For holistic faces, all features are extracted, whereas partial faces only allow the extraction of particular patches. Thus, partial faces can only be recognized if their patches overlap. Moreover, all faces are aligned using the eye corners, restricting this approach to faces where eyes are always visible.

With dynamic feature matching, He *et al.* [73] proposed a fully convolutional network capable of matching face patches of arbitrary size. After multiple convolutional layers, holistic faces are described by feature maps of size $7\times7\times512$, whereas partial faces are represented by feature maps of size $H\times W\times512$, where $H \leq 7$ and $W \leq 7$. Since identical feature map dimensions are required for matching, various sub-feature maps of size $H\times W\times512$ are extracted from the holistic feature maps. Then, every sub-feature map is compared to the feature map of the partial face. Sparse and similarity-guided constraints further supervise the matching. To cope with misalignment of feature maps caused by different resolutions, the authors further extended their work by incorporating multiscale feature maps [74].

Keypoint-based approaches [81, 124, 230] and both deep-learning methods [73, 75] accomplish matching of faces of arbitrary input resolutions. Nevertheless, they require overlapping face patches. Hence, matching non-overlapping partial faces, *e.g.*, a patch of the eye with the mouth, is not feasible with [73, 75, 81, 124, 230]. This also holds for earlier models, which solely focus on single face parts [67, 168, 186].

Despite the vigorous efforts to adapt CNNs to handle arbitrary input sizes by He *et al.* [73–75], recent methods shifted focus from face patches of arbitrary input resolutions to fixed-size inputs. In order to obtain an input image with a fixed resolution, two options are plausible: resizing or zero-padding every face. The former was considered by He *et*

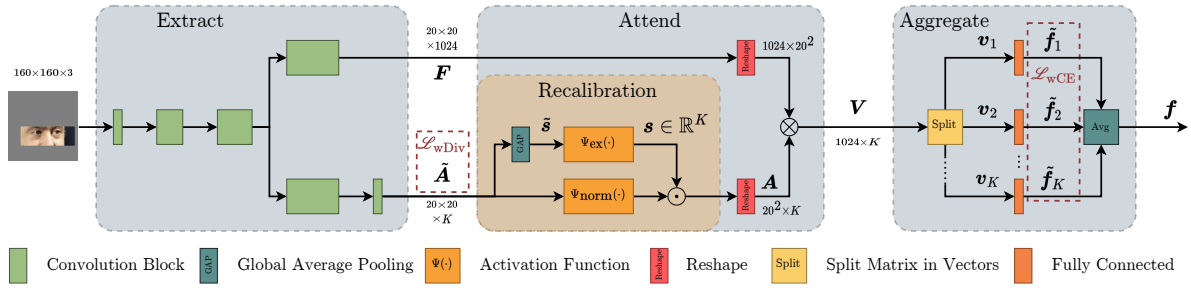

Figure 5.2: Architecture of the partial FR network. A truncated ResNet with $L = 40$ layers extracts feature maps $\boldsymbol{F}$ and attention maps $\tilde{\boldsymbol{A}}$ from a partial face $\boldsymbol{I}_{\mathrm{o}}$. By leveraging the recalibrated attention maps $\boldsymbol{A}$, information from $\boldsymbol{F}$ at the positions indicated by $\boldsymbol{A}$ is pooled into K intermediate feature vectors $\boldsymbol{v}_k$. The aggregate module maps $\boldsymbol{v}_k$ into a joint feature space $\tilde{\boldsymbol{f}}_k$ in which they are aggregated to obtain the final feature vector $\boldsymbol{f}$, which robustly encodes the information of a partial face. Adapted from [10†].

al. [73, 74], who employed a VGGNet [191] (see Section 3.3) as a baseline. Their analysis clearly showed that deforming the face patch to match the desired input resolution leads to inferior results. Hence, deforming the input should only be used if an FR network is available and a fast solution without additional training is necessary. Zero-padding provides a less invasive way of obtaining an image with a fixed resolution. Still, it must be noted that the information of an appropriate size is leveraged if face patches are zero-padded. Hence, this information needs to be available by the partial face detector.

Apart from the partial FR approaches mentioned earlier, occlusion-robust FR approaches can also handle partial faces. *E.g*., blind inpainting approaches (see Section 4.3.3) detect the occlusion and reconstruct the face [28, 77, 123, 133, 138, 216, 224, 250, 267, 281]. However, the reconstruction leaves many unreliable pixels, complicating the subsequent FR.

Thus, direct occlusion-robust FR approaches are proposed. Geng *et al*. [50] employed a generative adversarial network (GAN) to generate medical face masks for data augmentation and thereby increased the robustness of FR models when exposed to faces with medical face masks. Besides, Ding *et al*. [40] proposed to combine local features from the non-occluded area with global features originating from partly occluded pixels. Wan *et al*. [210] attenuated the activations associated with occluded face areas of a middle layer in a CNN. Song *et al*. [195] claimed that these activations in a middle layer are not discriminative enough and proposed to entirely discard activations originating from occluded pixels right before the bottleneck layer. In order to improve the robustness against occlusions, Xu *et al*. [243] incorporated an attention mechanism to extract local features, which are combined with global features to form an occlusion-robust feature.

5.2 Architecture

Extracting valuable information from faces, where only a small part of the face is visible, constitutes a challenging task. The network needs to extract relevant information from a

Table 5.1: Architecture of the extract module. Residual units are shown in brackets. † denotes that the 3×3 convolution in the first unit operates with stride $S = 2$.

Block	Size	Layer
1	80×80	7×7, 64, $S = 2$
2	40×40	3×3 maximum pooling, $S = 2$ $\begin{bmatrix} 1\times1, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix} \times 3$
3	20×20	$\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128^{\dagger} \\ 1\times1, & 512 \end{bmatrix} \times 4$
4	20×20	$\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix} \times 6$ $\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \\ 1\times1, & K \end{bmatrix} \times 6$

small patch and create general, comparable features regardless of the content and position of the patch. To be precise, the feature vector $\boldsymbol{f}$ describing the identity information of a face must be invariant to crops of varying sizes of the same face around different face parts. Besides, it must further be invariant against the position of the crop, *e.g.*, the network must learn to extract information from an eye since the patch contains an eye and not because the patch is positioned at the typical location of the eye.

To handle such scenarios and improve the invariance of the network against content and position, Hörmann *et al.* [10†] proposed to split the network into three distinct modules: 1) The extract module processes the input image $\boldsymbol{I}_{\mathrm{o}}$ and extracts feature maps $\boldsymbol{F} \in \mathbb{R}^{20\times20\times1024}$, which spatially encode identity information, and attention maps $\tilde{\boldsymbol{A}} \in \mathbb{R}^{20\times20\times K}$, which point towards relevant information in $\boldsymbol{F}$. 2) In the attend module, the information in $\boldsymbol{F}$ is pooled into K intermediate feature vectors $\boldsymbol{v}_k$ as indicated by the kth recalibrated attention map $\left[\boldsymbol{A}\right]_{:,:,k}$. 3) The aggregate module concludes the architecture, which maps all intermediate feature vectors $\boldsymbol{v}_k$ into a joint feature space $\tilde{\boldsymbol{f}}_k \in \mathbb{R}^{256}$ and outputs the final feature vector $\boldsymbol{f}$ by averaging all $\tilde{\boldsymbol{f}}_k$. The following subsections explain every module in detail.

5.2.1 Extract Module

As introduced in Section 3.3, state-of-the-art FR approaches leverage residual networks (ResNets) [72] to extract rich identity features from faces. In contrast to approaches designed to embed holistic faces into a discriminative feature space represented by a feature vector $\boldsymbol{f}$, the extract module provides feature maps $\boldsymbol{F}$ of size 20×20 in order to preserve spatial information. A larger resolution of $\boldsymbol{F}$ is obtained by considering larger resolutions of $r\times r = 160\times160$ of the input face $\boldsymbol{I}_{\mathrm{o}}$ instead of the widespread $r = 112$. Inspired by Xie *et al.* [239], Hörmann *et al.* [10†] based their extract module on a ResNet-v2 with depth $L = 50$, which is truncated after $l = 40$, *i.e.*, after the fourth block named "conv4_x" (*cf.* Table 3.4). By downsampling the input only three times, a resolution of $\boldsymbol{F} = \Theta^{[40]}(\boldsymbol{I}_{\mathrm{o}}) \in \mathbb{R}^{20\times20\times1024}$ is obtained. In this way, the position of the

features is still well-distinguishable. Unlike Xie *et al.* [239], Hörmann *et al.* [10†] split the ResNet after the third block such that each branch focuses on its respective task. To obtain K attention maps $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ an extra 1×1 convolutional layer is added. The rectified linear unit (ReLU) [159] activation function is employed throughout the extract module. Table 5.1 summarizes the detailed architecture of the extract module.

In order to extract meaningful information from the non-occluded areas, attention maps $\tilde{\boldsymbol{A}}$ should fulfill the following two properties: 1) They should be mutually exclusive, *i.e.*, $\left[\tilde{\boldsymbol{A}}\right]_{:,:,j} \not\approx \left[\tilde{\boldsymbol{A}}\right]_{:,:,k} \forall\, j \neq k$, such that every attention map $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ points at the location of distinct features. 2) The activation of the attention maps should correlate with the presence of its respective feature in $\boldsymbol{F}$, *i.e.*, if only the nose is visible, attention maps indicating the location of nose-related features should have only values ≈ 0. Note that $\tilde{\boldsymbol{A}}$ is implicitly defined, and thus the attention maps $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ do not necessarily correspond to the human-defined facial landmarks.

5.2.2 Attend Module

A recalibration is required before pooling the information from the feature maps $\boldsymbol{F}$ at the locations indicated by $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ to obtain a feature vector $\boldsymbol{v}_k$. Xie *et al.* [239] implemented attentional pooling for set-based FR. This allowed them to normalize $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ spatially over all pixels and across all images within a set. In this way, they guarantee that the information is pooled from the image with the most prominent features. In contrast to set-based FR with multiple holistic faces, partial FR only considers a single face or even a face region. Therefore, features corresponding to face regions defined by $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ may not be present in $\boldsymbol{F}$ due to the occlusion. This corresponds to attention maps $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ with values ≈ 0, which the recalibration must address.

Hörmann *et al.* [10†] based their recalibration on the squeeze-and-excitation block proposed by Hu *et al.* [82] and converted it into a parameter-free recalibration suitable for partial FR. First, every pixel in $\tilde{\boldsymbol{A}}$ is self-normalized independently to a value range between 0 and 1 by employing a sigmoid function $\Psi_{\text{norm}}(\cdot) = \text{sigmoid}(\cdot)$.

In parallel, a score vector $\tilde{\boldsymbol{s}} \in \mathbb{R}^K$ denoting the presence of the respective features in the kth feature map is obtained via global average pooling (GAP) by

$$[\tilde{\boldsymbol{s}}]_k = \frac{1}{20^2} \sum_{i=1}^{20} \sum_{j=1}^{20} \left[\tilde{\boldsymbol{A}}\right]_{i,j,k}, \qquad k = 1, \ldots, K. \tag{5.2}$$

After normalizing $\tilde{\boldsymbol{s}}$ with $\Psi_{\text{ex}}(\cdot) = \text{softmax}(\cdot)$ by

$$\boldsymbol{s} = \Psi_{\text{ex}}(\tilde{\boldsymbol{s}}), \tag{5.3}$$

the vector $\boldsymbol{s}$ indicates the relevance of every attention map $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ after comparing the average activation of all K attention maps $\left[\tilde{\boldsymbol{A}}\right]_{:,:,j}$. Then, the normalized scores $\boldsymbol{s}$ are leveraged to recalibrate the self-normalized attention maps yielding the final recalibrated

attention maps

$$\left[\boldsymbol{A}\right]_{:,:,k} = [\boldsymbol{s}]_k \Psi_{\text{norm}}\left(\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}\right), \qquad k = 1, \ldots, K. \tag{5.4}$$

Equation (5.4) illustrates that the attention maps are recalibrated in two ways: 1) Local self-normalization is performed by normalizing every pixel individually; and 2) global cross-normalization according to the relevance vector $\boldsymbol{s}$, which is calculated by comparing the average activation of all attention maps. Thus, the activations in $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ are recalibrated to correlate with the presence of the corresponding features in $\boldsymbol{F}$ with respect to all other features indicated by $\left[\tilde{\boldsymbol{A}}\right]_{:,:,j}$ for $j \neq k$.

After recalibration, $\boldsymbol{F}$ and $\boldsymbol{A}$ are reshaped to a resolution of 1024×20^2 and $20^2 \times K$, respectively. Then attentional pooling, as in [239], is performed by

$$\boldsymbol{V} = \boldsymbol{F}\boldsymbol{A}. \tag{5.5}$$

The attentional pooling, as described by Equations (5.2) to (5.5), describes a variation of a simplified self-attention block as introduced in Section 4.4.1.3. In self-attention, a normalized matrix of relevance scores $\boldsymbol{S}$ is used to pool information from the value matrix $\boldsymbol{V}$ at positions specified by $\boldsymbol{S}$. Here, $\tilde{\boldsymbol{A}}$ is normalized and used to pool information from $\boldsymbol{F}$ according to the positions indicated by $\boldsymbol{A}$.

The output of the attentional pooling $\boldsymbol{V} \in \mathbb{R}^{1024 \times K}$ is a matrix of intermediate feature vectors $\boldsymbol{V} = \begin{pmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \ldots \boldsymbol{v}_K\end{pmatrix}$, which contain the information in $\boldsymbol{F}$ determined by the corresponding attention map $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$. If the location where the information is encoded in $\boldsymbol{F}$ shifts, the corresponding attention maps pointing towards that information will also shift. Thus, this disentanglement of feature map $\boldsymbol{F}$ and attention maps $\tilde{\boldsymbol{A}}$ allows the network to become more invariant to the position of relevant information, which is vital for partial FR.

5.2.3 Aggregate Module

The aggregate module concludes the partial FR network with the objective of transforming K feature vectors $\boldsymbol{v}_k$ into a joint feature space for FR. Since $\boldsymbol{v}_k$ were pooled from different regions in $\boldsymbol{F}$ according to $\boldsymbol{A}$, they comprise different information and cannot be aggregated directly. Thus, every $\boldsymbol{v}_k$ needs to be mapped independently into a joint feature space $\tilde{\boldsymbol{f}}_k \in \mathbb{R}^{256}$ by a fully connected layer

$$\tilde{\boldsymbol{f}}_k = \boldsymbol{W}_k \boldsymbol{v}_k, \qquad k = 1, \ldots, K, \tag{5.6}$$

where $\boldsymbol{W}_k \in \mathbb{R}^{256 \times 1024}$ denotes the kth weight matrix of the aggregate module. As typical in FR, no activation function is applied to $\tilde{\boldsymbol{f}}_k$ in order to exploit the entire value space.

After transforming $\boldsymbol{v}_k$ into a joint feature space $\tilde{\boldsymbol{f}}_k$, identity information is encoded similarly in $\tilde{\boldsymbol{f}}_k$. Thus, the information from all K feature vectors is aggregated via averaging

$$\boldsymbol{f} = \frac{1}{K} \sum_{k=1}^{K} \tilde{\boldsymbol{f}}_k. \tag{5.7}$$

Owing to the previous attentional pooling followed by the mapping into a joint feature space, $\boldsymbol{f}$ is relatively invariant against the position and content of the partial face, which is crucial for a reliable partial FR.

5.3 Loss Functions

The main objective of the partial FR network constitutes a robust FR performance. Thus, any FR loss from Section 3.4 can be used for training. Besides, it is vital to ensure that the attention maps $\tilde{\boldsymbol{A}}$ are mutually exclusive. Hence, a diversity regularization loss is employed to guarantee diversity among $\tilde{\boldsymbol{A}}$. Both losses were adapted to partial FR by Hörmann *et al.* [10†].

Overall, both losses are combined with weight decay (*cf.* Section 2.6.2), which leads to the following description of the total loss

$$\mathscr{L}_{\text{tot}} = \lambda_{\text{wCE}}\mathscr{L}_{\text{wCE}} + \lambda_{\text{wDiv}}\mathscr{L}_{\text{wDiv}} + \lambda_{\text{reg}}\mathscr{L}_{\text{reg}}, \tag{5.8}$$

where λ denote scalars to balance the losses, and $\mathscr{L}_{\text{reg}}$ is the L_2 norm of all trainable weights following Equation (2.14).

5.3.1 Weighted Softmax Cross-Entropy Loss

The aggregate module comprises multiple fully connected layers that transform $\boldsymbol{v}_k$ into a joint feature space, in which an aggregation as in Equation (5.7) is possible. However, a joint feature space is not formed without any additional regularization.

In order to obtain a joint feature space, another fully connected layer is added to every $\tilde{\boldsymbol{f}}_k$; however, the weight matrices are shared. As typical when training FR losses, every neuron in the last fully connected layer represents one identity of the training dataset. Since the number of identities vastly outnumbers the feature dimensionality 256, $\tilde{\boldsymbol{f}}_k$ act as bottleneck layers, which improves the network's generalization. Furthermore, due to the shared weights of the fully connected layer, all $\tilde{\boldsymbol{f}}_k$ are transformed equally. In this way, all $\tilde{\boldsymbol{f}}_k$ must encode identity information likewise in order to be mapped onto the same identity.

Then, K softmax cross-entropy (CE) losses $\mathscr{L}_{\text{CE},k}$ are calculated for every $\tilde{\boldsymbol{f}}_k$ following Equation (3.7). However, not every $\mathscr{L}_{\text{CE},k}$ is equally relevant as they may originate from occluded areas. In order to consider the relevance of every $\mathscr{L}_{\text{CE},k}$, Hörmann *et al.* [10†] proposed to leverage the normalized relevance scores $\boldsymbol{s}$ during the aggregation $\mathscr{L}_{\text{CE},k}$. Then, the weighted softmax CE loss is calculated by

$$\mathscr{L}_{\text{wCE}} = \sum_{k=1}^{K} [\boldsymbol{s}]_k \mathscr{L}_{\text{CE},k}. \tag{5.9}$$

Scaling every $\mathscr{L}_{\text{CE},k}$ with its respective relevance $\mathscr{L}_{\text{wCE}}$ emphasizes the information from visible face parts, whereas the influence of unreliable features from occluded face parts is mitigated.

5.3.2 Weighted Diversity Regularizer

As motivated in Section 5.2.1, attention maps $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ should be mutually exclusive. Otherwise, the attention map will likely collapse into one of the two scenarios: All K attention maps focus on the same face regions or only a single attention map is established, whereas the remaining are discarded. Thus, diversity among all $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ must be promoted by a separate regularizer.

Xie *et al.* [239] proposed a diversity regularizer, which penalizes mutual overlap between $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ and $\left[\tilde{\boldsymbol{A}}\right]_{:,:,j}$ for $k \neq j$. First, every attention map $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ is self-normalized using a softmax. The resulting probability map $\left[\boldsymbol{P}\right]_{:,:,k}$ indicates the activations in $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ independent of the strength of the activations and are computed by

$$\left[\boldsymbol{P}\right]_{i,j,k} = \frac{\exp\left(\left[\tilde{\boldsymbol{A}}\right]_{i,j,k}\right)}{\sum_{i=1}^{20}\sum_{j=1}^{20}\exp\left(\left[\tilde{\boldsymbol{A}}\right]_{i,j,k}\right)} \tag{5.10}$$

If an attention map is responsible for a single face region and this region is occluded, the distribution of its activation should not be considered by the regularizer. Thus, Hörmann *et al.* [10†] modified the diversity regularizer from Xie *et al.* [239] to consider the relevance of the attention maps, which is indicated by $\boldsymbol{s}$. Then, the loss of weighted diversity regularizer is formulated as follows:

$$\mathscr{L}_{\text{wDiv}} = 1 - \sum_{i=1}^{20}\sum_{j=1}^{20}\max_{k}\left([\boldsymbol{s}]_k\left[\boldsymbol{P}\right]_{i,j,k}\right). \tag{5.11}$$

If the scaling with $[\boldsymbol{s}]_k$ is neglected, mutually overlapping $\left[\boldsymbol{P}\right]_{:,:,k}$ are absorbed by the maximum projection, as only the maximum value across all probability maps is considered for every pixel. K mutually exclusive probability maps $\left[\boldsymbol{P}\right]_{:,:,k}$ have their activations at different positions. This results in a max projection with the sum of all its pixels close to K. By normalizing $\left[\boldsymbol{P}\right]_{:,:,k}$ with $[\boldsymbol{s}]_k$, mutually exclusive probability maps yield a sum of all pixels close to 1.

5.4 Experiments

5.4.1 Training Details

Directly training the partial FR network with partial faces fails as the network cannot extract diverse and rich features from partial faces. One way to slowly adapt to this difficult task is to increase the occlusion area after every epoch. A more straightforward approach, which requires less tedious parameter tuning, was proposed by Hörmann *et al.* [10†] by splitting the training in a pretraining on solely holistic faces followed by a finetuning on partial faces.

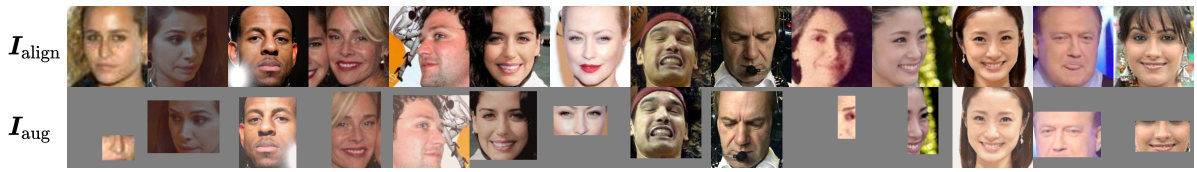

Figure 5.3: Faces before data augmentation $\boldsymbol{I}_{\text{align}}$ and after data augmentation $\boldsymbol{I}_{\text{aug}}$ for training.

The partial FR model is trained on the VGGFace2 [14] dataset, comprising 3.1M images of 8631 identities (see also Table 3.1). All faces are aligned using the custom face alignment policy (FAP) with the facial landmarks extracted by the multi-task CNN (MTCNN) [265] and cropped to a resolution of 160×160 pixels. Regarding data augmentation, brightness, contrast, and saturation are varied and horizontal flipping is performed with $p_{\text{aug}} = 0.5$. The remaining augmentation parameters are set to the default values introduced in Section 3.5.1.

For pretraining, the weighted softmax CE loss $\mathscr{L}_{\text{wCE}}$ is replaced by an average of all individual loss terms $\mathscr{L}_{\text{CE},k}$. In this way, the feature originating from every attention map is considered equally. This is necessary as all face areas are visible in holistic faces. Besides, this is crucial as Equation (5.11) is also minimized if the network focuses on a single attention map, whereas the remaining attention maps remain without activations regardless of the input. However, this would inevitably cause $K-1$ out of K individual CE loss terms $\mathscr{L}_{\text{CE},k}$ to predict wrong, ultimately leading to an even larger total loss $\mathscr{L}_{\text{tot}}$. Hence, the average CE loss ensures that every feature $\tilde{\boldsymbol{f}}_k$ extracted by its corresponding attention map $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ – and in this way, every $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ – is considered, while the weighted diversity regularizer $\mathscr{L}_{\text{wDiv}}$ guarantees that all $\left[\tilde{\boldsymbol{A}}\right]_{:,:,k}$ are mutually exclusive.

Similarly to the training of the FR models in Section 3.5.1, the pretraining is performed for 20 epochs with the Adam optimizer [110]. However, a batch size $N_{\text{b}} = 50$ is used due to memory limitations. The losses are balanced by setting $\lambda_{\text{wCE}} = \lambda_{\text{wDIV}} = 1$ and $\lambda_{\text{REG}} = 5 \cdot 10^{-5}$. An initial learning rate of $\eta = 0.05$ is used, which is reduced by a factor of $\gamma_{\text{lr}} = 4$ every 6 epochs. Furthermore, dropout with $p_{\text{d}} = 0.2$ is employed after $\boldsymbol{v}_k$ to improve generalization.

After the pretraining, the network provides satisfying results on holistic faces. To adapt the network to partial faces, the model is trained with partial faces with $p_{\text{aug}} = 0.8$. Figure 5.3 illustrates the partial faces, which are generated by applying a rectangular mask $\boldsymbol{M}_{\text{gt}}$ to the ground-truth image following Equation (5.1). In comparison to Chapter 4, the masks are inverted yielding a small face patch. The rectangular masks are generated with an aspect ratio sampled from $\mathcal{U}_1(0.5, 2)$ and an area ratio $\kappa_{\text{a}} \sim \mathcal{U}_1(0, 0.9)$ (*cf.* Equation (4.2)). Besides, the center coordinates of the face patch are drawn from $\mathcal{U}_1(0.15r, 0.85r)$ to ensure that a small patch always includes part of the face. Since pretraining provided well-initialized weights, training for five more epochs suffices. The learning rate η is reset to 0.002 and decayed by a factor of $\gamma_{\text{lr}} = 4$ every 2 epochs. For finetuning, the weighted softmax CE loss $\mathscr{L}_{\text{wCE}}$ as defined in Equation (5.9) is employed. All remaining training parameters are identical to the pretraining.

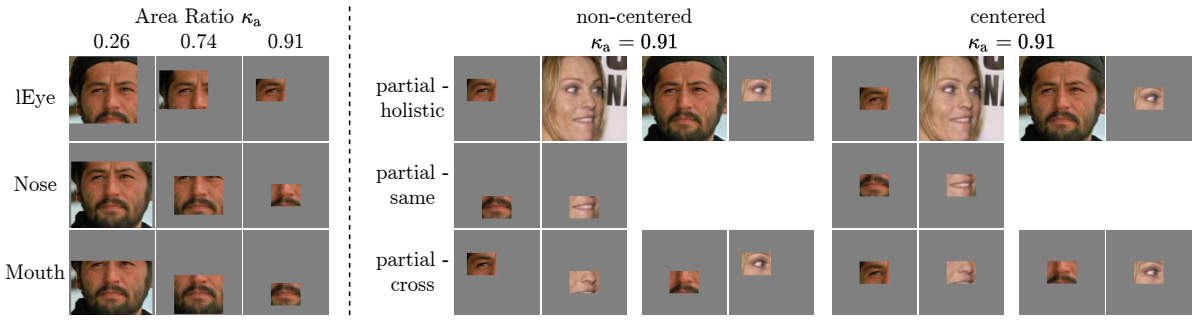

Figure 5.4: Position and size of the occluded area considered during benchmark (left), where the right eye is not listed to improve visibility. Definition of different protocols for partial FR with an excerpt of the included pairs (right).

5.4.2 Evaluation Details

Similar to Chapter 4, a systematic evaluation is desirable. In contrast to the random masking during training, the area ratio κ_a is varied from 0.26 to 0.91. Besides, the patches are centered around the left eye, right eye, nose, and mouth according to the dataset-wide target facial landmark coordinates. Figure 5.4 (left) illustrates an example of partial faces created using the script published by Hörmann *et al.* [10†].

Together with the holistic face, any face verification protocol offers 25 variations in terms of position, from which 7 are discarded due to the similarity of the left eye and right eye. As introduced in Section 3.5.2.1, face verification protocols are defined by triplets of two images, $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, and a binary ground-truth label. Then, Hörmann *et al.* [10†] divided the remaining 18 variations into the following protocols, where the variations are denoted by "Patch of $\boldsymbol{I}_1$"-"Patch of $\boldsymbol{I}_2$":

- The *holistic* protocol is the standard protocol as it compares the non-partial version of a pair, *i.e.*, Holistic-Holistic.
- The *partial-holistic* protocol matches a partial face with a holistic face. This includes the following variations: lEye-Holistic, Nose-Holistic, Mouth-Holistic, Holistic-lEye, Holistic-Nose, and Holistic-Mouth.
- The *partial-same* protocol considers two identical face patches, *i.e.*, lEye-lEye, Nose-Nose, and Mouth-Mouth.
- The *partial-cross* protocol evaluates face patches centered around different face regions, *i.e.*, lEye-rEye, lEye-Mouth, lEye-Nose, Mouth-Nose, rEye-lEye, Mouth-lEye, Nose-lEye, and Nose-Mouth.

To evaluate whether the network can extract features of a face patch regardless of its position within $\boldsymbol{I}_o$, every protocol is extended by a *centered* version. In this version, the face patch is moved to the center of $\boldsymbol{I}_o$ followed by zero padding. Examples for every partial protocol are depicted in Figure 5.4 (right).

For synthetic partial faces, the LFW dataset is utilized with the partial protocols as described in this section. Despite the saturation when evaluating on the LFW dataset with holistic faces, the accuracy (*Acc*) drops substantially when synthetically occluding the face. Therefore, LFW is a viable choice as it yields meaningful results. Hörmann *et al.* [10†] released this *Partial LFW* (PLFW) dataset. In order to evaluate whether training on synthetic partial faces also improves the performance for natural partial faces due to extreme head poses, as in Figure 5.1, the Cross-Pose LFW (CPLFW) dataset [275] is employed.

5.5 Results

The partial FR network is compared with the *VGG-160*, which was analyzed exhaustively in Section 3.6.2.[ii] Besides, a ResNet-v2 with depth $L = 50$ (identical to the VGG-160 model) and $L = 41$ were trained with the training protocol as in Section 5.4.1. The ResNet with $L = 41$ has the same number of layers as the partial FR network. The networks are denoted by *ResNet-41* and *ResNet-50*.

The partial FR network is also trained without the aggregate module. In this case, all K normalized attention $\boldsymbol{A}_k$ are averaged to obtain a single global attention map. With this single attention map, attentional pooling yields a single feature vector $\boldsymbol{v}$, which is processed by another fully connected layer denoting the feature space $\boldsymbol{f}$. Hence, the model without the aggregate module is trained with standard softmax CE loss.

5.5.1 Ablation Study

First, an ablation study is performed to confirm the design choices made in terms of architecture and loss functions. The *Acc* for various holistic and partial face verification protocols are reported in Table 5.2. As expected, the VGG-160 model struggles with partial faces. A substantial drop in *Acc* is observed for all partial protocols compared to the holistic protocol. While the partial-holistic and the partial-same protocols are still handled relatively well, the partial-cross protocol causes the most significant drop in *Acc*. Furthermore, when normalizing the position of the patches to the center, the *Acc* worsens further. Thus, by default, CNNs are very susceptible to spatial shifts. In FR, this effect is likely higher than usual since the network expects aligned faces and thus relies on specific information always being found at certain locations in the image space.

Finetuning VGG-160 on partial faces reduces the generalization gap inherent in any FR model, which was never exposed to partial faces during training. While the ResNet-50 obtains superior *Acc* than ResNet-41 on all non-centered protocols, ResNet-41 substantially outperforms ResNet-50 on all centered-protocol. Hörmann *et al.* [10†] conjecture that the large difference in trainable parameters (8.82M *vs.* 24.05M) causes the ResNet-50 to overfit on the default position of certain feature information as known

[ii] Note that the results vary slightly from those reported in Section 3.6.2 as a different batch size N_b was employed to maintain consistency.

Table 5.2: Ablation study on architecture and loss functions. The average *Acc* [%] is reported over nine occlusion sizes κ_a and all position pairs as listed in Section 5.4.2. The average *Acc* of all partial protocols is reported in the last column. The highlighted model is analyzed in detail in the following subsection.

					CPLFW	LFW							
							non-centered: partial-			centered: partial-			
K	Ψ_{ex}	Ψ_{norm}	Agg	$\mathscr{L}_{wCE}$	holistic	holistic	holistic	same	cross	holistic	same	cross	avg
VGG-160					88.20	99.58	94.77	94.93	88.85	92.05	92.47	83.92	91.16
ResNet-41					87.52	99.62	97.71	97.27	94.53	97.25	96.80	93.56	96.19
ResNet-50					87.80	99.60	97.75	97.36	94.80	95.48	94.72	89.60	94.95
5	no recalibration		✗	✗	87.80	99.47	97.60	97.18	94.14	97.01	96.64	92.96	95.92
5	softmax	softmax	✗	✗	88.42	99.45	97.76	97.30	94.23	97.25	96.77	93.00	96.05
5	softmax	sigmoid	✗	✗	88.87	99.62	**98.04**	97.60	94.58	97.62	**97.16**	93.45	96.41
5	softmax	sigmoid	✓	✗	89.10	99.47	98.02	97.56	94.79	97.61	97.12	93.74	96.48
5	softmax	sigmoid	✓	✓	**89.18**	99.67	97.99	97.54	94.79	97.58	97.08	93.73	96.45
12	no recalibration		✗	✗	88.03	99.63	97.74	97.28	94.38	97.17	96.63	93.06	96.04
12	softmax	softmax	✗	✗	88.10	99.50	97.61	97.11	94.43	96.77	96.24	92.68	95.81
12	softmax	sigmoid	✗	✗	89.13	99.62	97.99	97.61	94.62	97.54	97.03	93.44	96.37
12	softmax	sigmoid	✓	✗	89.08	99.60	98.02	97.56	94.85	97.60	97.08	93.86	96.49
12	softmax	sigmoid	✓	✓	88.97	**99.70**	98.03	**97.66**	**94.90**	**97.64**	**97.16**	**93.87**	**96.54**

from aligned faces. In contrast, the reduced number of trainable parameters of the ResNet-41 restricts the training by not learning such spatial dependencies.

The ablation study demonstrates the necessity of recalibration. No recalibration or recalibrating with Ψ_{norm} = softmax yields inferior *Acc* on all protocols compared to Ψ_{norm} = sigmoid. The employment of the aggregate module is only visible in an improvement on the partial-cross protocols for $K = 5$, whereas a slight but consistent improvement originating from the aggregate module is observed for all protocols if $K = 12$. The weighted softmax CE loss $\mathscr{L}_{wCE}$ only improves the *Acc* for $K = 12$. For $K = 5$, $\mathscr{L}_{wCE}$ has negligible influence. The reason for this behavior likely lies in the fact that every attention map is considered (equally) important for lower K, and thus $\mathscr{L}_{wCE}$ has no influence. Due to the slight boost in *Acc* after employing $\mathscr{L}_{wCE}$, the best performance on almost all protocols is accomplished by the model with $K = 12$ attention maps, which is highlighted in Table 5.2.

On naturally occurring occlusions due to extreme head poses, as frequent in CPLFW, all partial FR networks with Ψ_{ex} = softmax and Ψ_{norm} = sigmoid improve the performance. However, this improvement is not attributed to the data augmentation encompassing partial faces since *Acc* drops as indicated by the ResNet-41 and ResNet-50. Thus, only the sophisticated attend and aggregate modules adequately leverage the data augmentation and use it to extract identity features from partial faces due to extreme head poses.

The results on the holistic LFW benchmark indicate a slight improvement. However, due to the saturation, this improvement may be caused by noise. Overall, it is apparent that the network highlighted in Table 5.2 achieves the best performance. The presented approach surpasses all baselines with fewer parameters than the ResNet-50 (19.09M *vs.* 24.05M). For further analysis, this model is denoted as the *partial FR network* (PFRN).

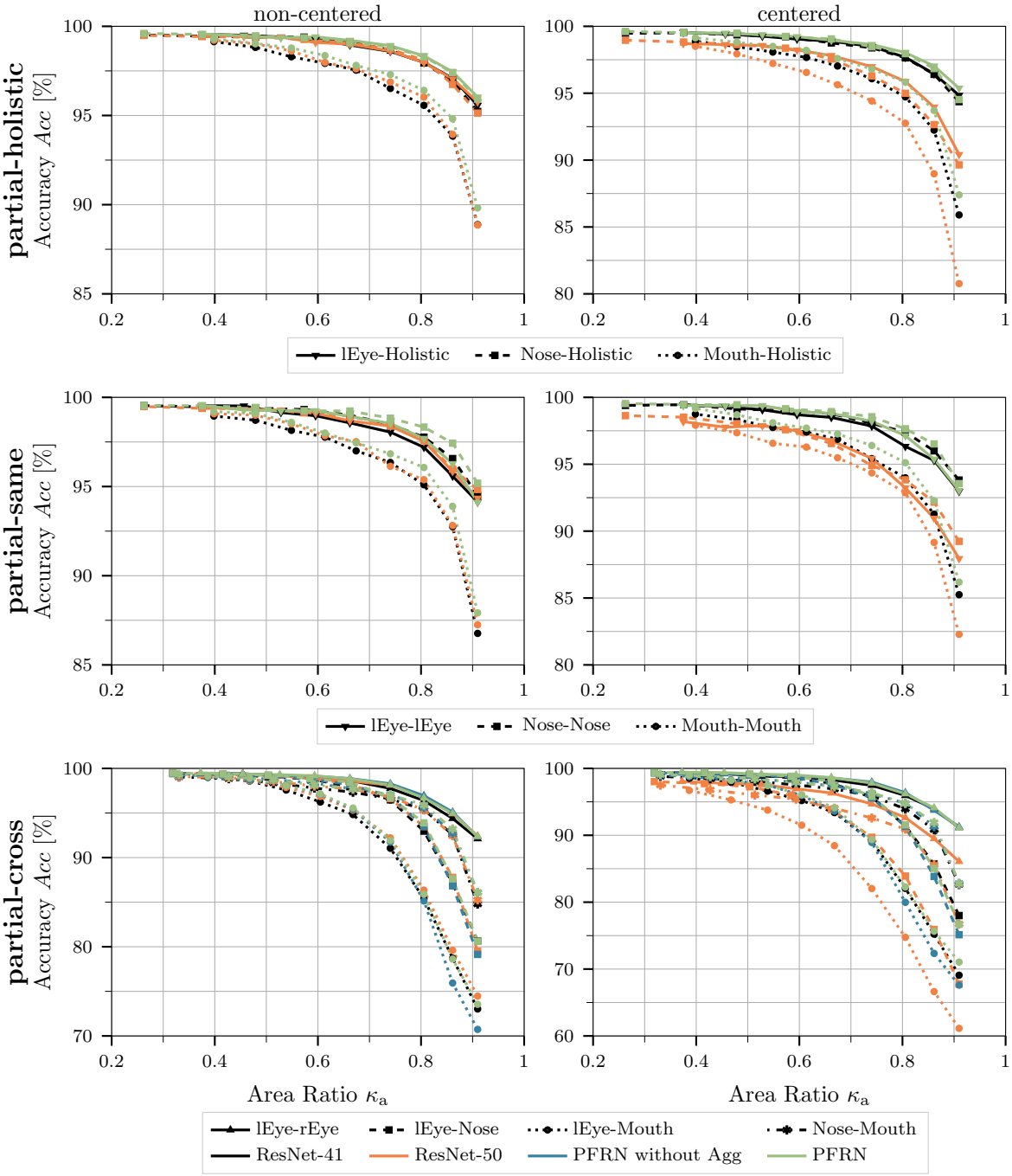

Figure 5.5: The accuracy (*Acc*) on the Partial LFW (PLFW) benchmark is split into all six protocols as introduced in Section 5.4.2. The partial FR network (PFRN) with and without the aggregate module are compared with the baselines (ResNet-41 and ResNet-50) for different area ratios κ_a. Note that all protocols are averaged with their inverse protocols, *i.e.*, lEye-nose also includes nose-lEye. Besides, partial FR network (PFRN) without the aggregate module is only shown for the partial-cross protocols due to its minor influence on the remaining protocols.

5.5.2 Detailed Analysis

Figure 5.5 illustrates the influence of the area ratio κ_a on the face verification *Acc*. As expected, increased occlusion sizes exacerbate the drop in *Acc*. The non-centered partial-holistic and partial-same protocols only indicate a clear difference between patches around the left eye, nose, and mouth. While *Acc* is relatively robust for the left eye and nose with $> 95\%$ for $\kappa_a \approx 0.9$, a clear drop below 90% is observed for the mouth region. Hence, the mouth region contains less meaningful information to distinguish between two faces. For the non-centered partial-cross protocol, the decrease in *Acc* correlates with the distance between both faces patches, *e.g.*, all networks struggle particularly with lEye-Mouth, followed by lEye-Nose and Nose-Mouth. lEye-rEye shows only slightly inferior results than lEye-lEye. Thus, comparing two distinct eyes is almost as good as comparing identical eyes.

The centered protocols follow the same trend as the non-centered protocols. Particularly noteworthy is the vast gap between ResNet-50 and the PFRN. Therefore, the most straightforward approach – finetuning a ResNet-50 with partial faces – overfits on the positions and clearly motivates the necessity of the attend and aggregate module as proposed in Section 5.2. Besides, a slight improvement between the PFRN with and without the aggregate module is apparent for lEye-Nose and lEye-Mouth, *i.e.*, the most challenging protocols, where it is necessary to map information originating from different patches into a joint feature space. Even when comparing centered patches containing $\approx 10\%$ of the images' area cropped around the left eye and mouth, the PFRN yields with $Acc > 70\%$ a satisfying performance when considering the difficulty of this protocol.

5.6 Conclusion and Future Work

In this chapter, a novel partial FR network (PFRN) network was presented, which comprises three modules: Firstly, the extract module employs a truncated ResNet-50 to predict feature maps and attention maps with a spatial resolution of 20×20. Secondly, the attend module pools information from the feature maps as indicated by recalibrated attention maps. Lastly, the aggregate module maps the information originating from different attention maps into a joint feature space and aggregates it into a single discriminative feature vector. The training is guided by a weighted classification loss, which considers the visibility of every attention map, together with a diversity regularizer to promote mutually exclusive attention maps.

The objectives of partial FR approaches, stated at the beginning of this chapter, were clearly met by the presented PFRN. The PFRN outperforms all baselines on all partial FR protocols and even improves the performance for holistic face pairs. Moreover, it allows the comparison of non-overlapping face patches. Most notably, an improvement was observed even for natural occurring occlusions despite training exclusively with synthetic occlusions. By considering centered face patches, it becomes apparent that the PFRN is more robust to spatial shifts than the traditional ResNet-50, which substantially overfits on the positions of the face patches. On the most challenging partial-cross

protocols, which have not been considered in most related works, the aggregate module of the PFRN is deemed vital to merge and compare feature information originating from non-overlapping face patches.

In order to improve the FR performance for partial faces in future works, more recent loss functions (see Section 3.4 can be employed. As analyzed by the centered protocols, the PFRN demonstrated robustness against the position of the partial face patch. However, due to the zero-padding of the face patch, the network leverages size information, *i.e.*, all face parts have roughly the same size as in the holistic face. Thus, future works should focus on lifting this restriction by extracting multiscale features. Besides, computing the attention maps similarly to Woo *et al.* [233], *i.e.*, without a parallel branch in the extract module, seems promising. First results by Tian [27$^+$] did not reveal any difference, suggesting that the number of trainable parameters can be reduced while maintaining comparable performance.

While this chapter has shown the viability of a direct partial FR approach, reconstructing partial faces prior to the FR offers a novel albeit very challenging research direction. Due to the large occlusions covering almost the entire face the task is only feasible with appropriate data augmentation and network structure. In fact, a model similar to the coarse-to-fine dual attention network (C2F-DAN) presented in Chapter 4 would not have a sufficiently large receptive field in the coarse network to reconstruct face parts, which are more distant from the non-occluded face patch. Therefore, guiding the model with identity information, *i.e.*, feature vectors extracted prior to the reconstruction, induced at various depths of the reconstruction, constitutes a promising option.

6

Towards Robust Permutation-Invariant Face Aggregation

In all still image face recognition (FR) approaches, previously discussed in Chapters 3 to 5, the objective constitutes recognizing the identity depicted in the input face $\boldsymbol{I}$. This chapter extends the input along the time axis by considering a video, *i.e.*, a sequence of various images. However, in FR, the additional temporal component does not serve as a valuable source of information to determine the identity. Except for scarce characteristic head movements, the vast amount of identity information is encoded in every frame instead of in the changes between frames. In contrast to other video-related tasks, such as gait recognition [20†, 21†] or action recognition [19†], where temporal information is vital, temporal information is rarely leveraged in video FR. Therefore, following most related works in video FR, the video sequence $\mathcal{V}$ comprising N frames $\boldsymbol{I}$ is written as an unordered set

$$\mathcal{V} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_N\}\,. \tag{6.1}$$

Hence, most video FR approaches can also be used for set-based FR tasks.

Compared to still image FR, recent video FR methods still face various challenges. Video frames are typically acquired under poor capturing conditions and are often affected by varying expressions, head poses, and motion blur. Due to these varying factors, not every frame is considered equally valuable. Thus, to leverage the theoretically vast increase of available information in N frames compared to a single image, it is essential to take into account the relevance of every frame. Sharp, frontal frames without motion blur must be favored compared to frames captured during head movements in order to optimally aggregate the information. Therefore, any video FR approach must carefully select which information is considered from which image. Besides, all video FR approaches must be capable of processing an arbitrary number of images N, and since the video is treated as an unordered set $\mathcal{V}$, the architecture must be entirely permutation-invariant.

Most video FR approaches [54–56, 134, 135, 148, 177, 183, 189, 240, 245, 266, 271, 278, 7†] first extract N feature vectors $\boldsymbol{f}_n \in \mathbb{R}^{M_\mathrm{f}}$. Then, all $\boldsymbol{f}_n$ are scaled by their relevance scores and aggregated to obtain a single feature vector $\boldsymbol{f}_\mathrm{a}$, which represents the entire video $\mathcal{V}$. In contrast, Rao *et al.* [178] proposed to aggregate N faces $\boldsymbol{I}_n \in \mathcal{V}$ into an aggregated face $\hat{\boldsymbol{I}}_\mathrm{a}$ followed by extracting a single feature vector $\boldsymbol{f}_\mathrm{a}$ from $\hat{\boldsymbol{I}}_\mathrm{a}$. This method provides multiple advantages: Firstly, the relevance of a frame is directly visible in the image space, whereas it is more challenging to extract this information in the feature space. Hence, the face aggregation network can directly extract the relevance from the image and predict it more accurately, mitigating the influence of outliers. Secondly, this disentanglement of face aggregation and FR allows every model to focus on its respective task. In this way, video FR performance can be further improved by employing a more recent FR network without the costly retraining of the aggregation network. Thirdly, with the aggregated image $\hat{\boldsymbol{I}}_\mathrm{a}$ an additional output is provided.

The objective of the face aggregation network $\mathrm{A}(\cdot)$ is formulated as

$$\underbrace{\{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_N\}}_{\mathcal{V}} \mapsto \hat{\boldsymbol{I}}_\mathrm{a} = \mathrm{A}(\mathcal{V}) \tag{6.2}$$

subject to:

$$d\left(\mathrm{F}(\mathcal{V}_i^{(n)}), \mathrm{F}(\mathcal{V}_j^{(m)})\right) \geq d\left(\mathrm{F}(\mathrm{A}(\mathcal{V}_i^{(n)})), \mathrm{F}(\mathrm{A}(\mathcal{V}_j^{(m)}))\right) \qquad \forall\, k, l, i = j, \tag{6.3}$$

$$d\left(\mathrm{F}(\mathcal{V}_i^{(n)}), \mathrm{F}(\mathcal{V}_j^{(m)})\right) < d\left(\mathrm{F}(\mathrm{A}(\mathcal{V}_i^{(n)})), \mathrm{F}(\mathrm{A}(\mathcal{V}_j^{(m)}))\right) \qquad \forall\, k, l, i \neq j, \tag{6.4}$$

where $d(\cdot, \cdot)$ is the cosine distance (see Equation (3.17)) and $\mathrm{F}(\cdot)$ denotes the feature extractor. The superscripts $^{(n)}$ and subscripts $_i$ indicate the nth video of the ith identity. In addition to the constraints in Equations (6.3) and (6.4), $\hat{\boldsymbol{I}}_\mathrm{a}$ should be realistic, *i.e.*, the viewer should not be able to discern $\hat{\boldsymbol{I}}_\mathrm{a}$ from any $\boldsymbol{I}_n \in \mathcal{V}$.

Hence, the main task constitutes the aggregation of N images $\mathcal{V}$ into a single aggregated image $\hat{\boldsymbol{I}}_\mathrm{a}$. The aggregation network should fuse the most valuable feature information from $\mathcal{V}$ in $\hat{\boldsymbol{I}}_\mathrm{a}$, such that its feature is more discriminative. *I.e.*, distances between genuine video pairs are smaller after aggregation (see Equation (6.3)), whereas distances between two imposter video pairs should become larger after the aggregation (see Equation (6.4)). Thereby, the aggregation network $\mathrm{A}(\cdot)$ becomes robust against varying capture conditions, *e.g.*, motion blur, illumination, or out-of-focus faces.

This chapter presents an approach for video face aggregation. Unlike Rao *et al.* [178], the video is considered an unordered set $\mathcal{V}$, requiring a complicated, permutation-invariant architecture since only then identical outputs independent on the image order are ensured. Large parts of the approach presented in this chapter were pre-published in [6†] and are referenced in the corresponding sections.

6.1 Related Work

The vast majority of approaches to video FR operate in the feature space, *i.e.*, every frame $\boldsymbol{I}_n \in \mathcal{V}$ is processed separately by a face feature extractor yielding N features $\boldsymbol{f}_n$. Next, a

aggregated feature vector $\boldsymbol{f}_{\mathrm{a}}$ representing the entire video $\mathcal{V}$ is obtained via an aggregation. Depending on whether temporal information is considered during this aggregation, every approach is either classified as sequence-based or set-based. While the order of the frames matters and is incorporated during the aggregation for sequence-based methods, set-based approaches are permutation-invariant, *i.e.*, the same $\boldsymbol{f}_{\mathrm{a}}$ must be obtained regardless of the frame order. Since the temporal information does not provide much insight into the faces' identity, most researchers [8, 56, 134, 135, 148, 183, 189, 239, 240, 245, 266, 271, 278, 7†] aim for set-based approaches as they can also be employed for sequence-based FR by discarding the temporal information. In contrast, sequence-based approaches [54, 55, 176–178] can typically not be employed in set-based FR, *e.g.*, in multiple still images.

Naive set-based approaches either compute the pairwise distance between all images within a set [187] or obtain $\boldsymbol{f}_{\mathrm{a}}$ by averaging frame-wise features by

$$\boldsymbol{f}_{\mathrm{a}} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{f}_n. \tag{6.5}$$

This simple technique is frequently employed by still image FR approaches, which also evaluate on videos or sets [33, 87, 109] (*cf.* Table 3.10). While this approach is fast and parameter-efficient, it disregards the varying relevance of every frame. Due to distinct quality, motion blur, occlusions, or changes in illumination, not every frame is considered equally valuable. Thus, more sophisticated approaches aim to adaptively aggregate every feature based on its respective relevance

$$\boldsymbol{f}_{\mathrm{a}} = \sum_{n=1}^{N} \nu_n \boldsymbol{f}_n, \tag{6.6}$$

where ν_n denotes the relevance of $\boldsymbol{f}_n$ and $\sum_{n=1}^{N} \nu_n = 1$. This general formulation of the feature aggregation motivated various researchers to propose ways of computing the relevance score ν_n.

Yang *et al.* [245] incorporated two cascaded attention blocks, which effectively predict a relevance score ν_n using the scalar product of $\boldsymbol{f}_n$ with a kernel. Hörmann *et al.* [7†] modified the neural aggregation network from Yang *et al.* [245] to improve its robustness against outliers, *i.e.*, faces with other identities in $\mathcal{V}$. Since the norm of the feature vectors $\|\boldsymbol{f}_n\|$ is a measure for feature quality [167], Meng *et al.* [148] provided a parameter-free solution of boosting the performance for set-based FR by scaling every feature with its norm, *i.e.*, $\nu_n = \|\boldsymbol{f}_n\|$. Shi *et al.* [189] represented every face as a Gaussian distribution in the feature space, in which aggregation is performed with $\nu_n = \frac{1}{\sigma_n^2}$. With multicolumn networks, Xie *et al.* [240] proposed to split ν_n into two factors: 1) visual quality, which is computed for every feature independently using a single fully connected layer; and 2) content-aware quality by concatenating every feature with the aggregation according to the visual quality and employing another fully connected layer. By combining visual and content-aware quality control, ν_n considers its internal quality compared to the remaining frames. Zhong *et al.* [278] proposed GhostVLAD, which soft-assigns every $\boldsymbol{f}_n$ to $K + G$ cluster centers. Then, the difference between every $\boldsymbol{f}_n$ and the cluster center is scaled

by the soft-assignment and aggregated. The final aggregated vector $\boldsymbol{f}_{\mathrm{a}}$ is obtained by concatenating all K cluster-wise aggregations.

While previous methods [148, 189, 240, 245, 278, 7†] directly leveraged the features $\boldsymbol{f}_n$ in order to compute their relevance scores ν_n, Liu *et al.* [134] computed a quality score ν_n based on intermediate feature maps of an FR network. Zhang *et al.* [266] employed a separate residual network (ResNet) with depth $L = 18$ to predict a discriminability score ν_n with an entirely separated branch. In contrast to previous approaches, Sankaran *et al.* [183] considered a pair of sets during matching in order to predict pair-specific ν_n. Guided by head pose information, a pixel-wise self-attention block (*cf.* Section 4.4.1.3 and [264]) computes ν_n specifically for the paired set such that both sets are aligned to each other, *i.e.*, information originating from similar head poses is compared. While this adaptive aggregation tailors two sets to each other, it creates a massive computation and storage overhead as the feature representing a set is not unique.

Besides estimating the relevance of every feature vector $\boldsymbol{f}_n$ as by [134, 148, 183, 189, 240, 245, 266, 278, 7†], another direction was pursued by predicting the relevance of every component of $\boldsymbol{f}_n$, *i.e.*, $[\boldsymbol{f}_n]_i$. Then, the aggregation is performed by

$$\boldsymbol{f}_{\mathrm{a}} = \sum_{n=1}^{N} \boldsymbol{\nu}_n \odot \boldsymbol{f}_n, \tag{6.7}$$

with $\boldsymbol{\nu} \in \mathbb{R}^{M_{\mathrm{f}}}$ and $\sum_{n=1}^{N} [\boldsymbol{\nu}_n]_i = 1 \,\forall\, i = 1, \ldots, M_{\mathrm{f}}$. Drawing from the success of the neural aggregation network [245], Liu *et al.* [135] modified the cascaded attention structure to perform a component-wise feature aggregation. Gong *et al.* [56] employed a separate, fully connected layer to predict the quality vectors $\boldsymbol{\nu}_n$, which is in parallel to the bottleneck layer representing $\boldsymbol{f}_n$. In order to avoid designing a permutation-invariant architecture, which can cope with an arbitrary number of inputs, all relevance scores ν_n are only compared with each other during a simple normalization to obtain $\sum_{n=1}^{N} \nu_n = 1$ [56, 134, 266]. To promote this information exchange between all N feature vectors $\boldsymbol{f}_n$ during the computation of the $\boldsymbol{\nu}_n$, Zhao *et al.* [271] incorporated graph convolutions, which allow leveraging more sophisticated contextual information.

With comparator networks, Xie *et al.* [239] proposed pooling information across all N inputs from feature maps at different spatial locations indicated by attention maps. Thus, the information of all N inputs at a specific region, *e.g.*, the eye, is combined before feature vectors are created. Similarly, Bai *et al.* [8] proposed a local feature enhancement network, which transfers local information across all images using pixel-wise self-attention (*cf.* Section 4.4.1.3 and [264]).

In contrast to all set-based approaches, which can also be applied to video sequences, sequence-based approaches are uniquely defined for videos and leverage the temporal information. Similar to [245, 7†], Gong *et al.* [54, 55] predicted feature quality ν_n; however, they employed long short-term memory networks (LSTMs) to leverage context information. Rao *et al.* [177] used attention-aware reinforcement learning to predict feature quality ν_n based on the feature vectors $\boldsymbol{f}_n$ and the frames $\boldsymbol{I}_n$.

Unlike all previous methods, which perform aggregation of feature vectors $\boldsymbol{f}_n$ according to equations Equations (6.5) to (6.7) [54–56, 134, 135, 148, 177, 183, 189, 240, 245, 266,

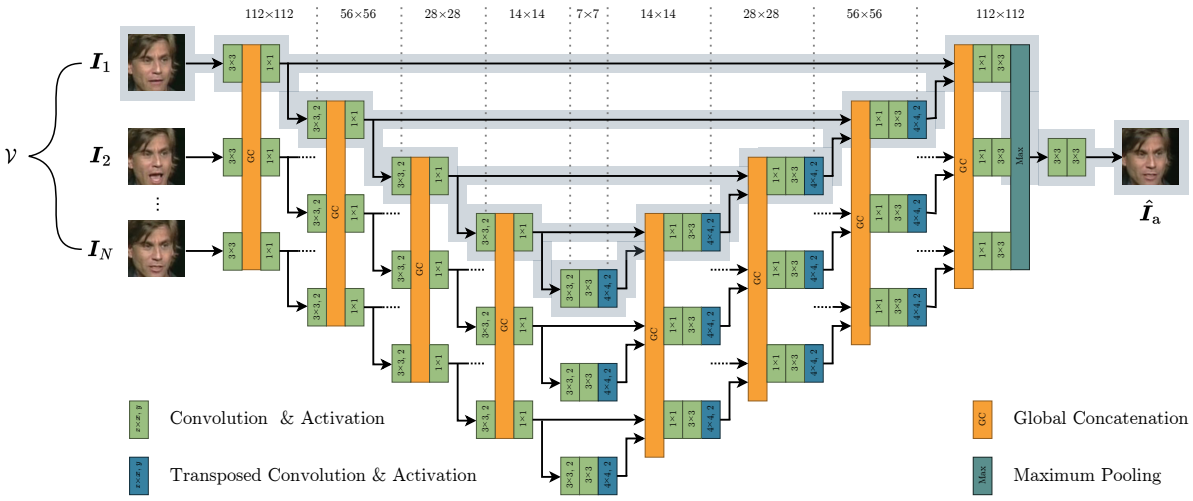

Figure 6.1: Architecture of the face aggregation network. N faces $\boldsymbol{I}_n$ are aggregated into a single face $\hat{\boldsymbol{I}}_\mathrm{a}$ by a permutation invariant U-Net. Every $\boldsymbol{I}_n$ is processed in a separate branch. Of these branches, the branch for $\boldsymbol{I}_1$ is highlighted. All weights between the branches are shared. Adapted from [6†].

271, 278, 7†], or fuse information of feature maps [8, 239], Rao *et al.* [176, 178] proposed the discriminative aggregation network (DAN), which aggregates images. Thus, N frames $\boldsymbol{I}_n$ are aggregated into a single image $\hat{\boldsymbol{I}}_\mathrm{a}$ followed by a feature extractor to obtain a single vector describing the video. However, due to the concatenation of $N = 20$ frames $\boldsymbol{I}_n$ at the input of the DAN, the approach becomes permutation-variant and allows the aggregation of exactly $N = 20$ frames.

6.2 Architecture

The DAN [178] always requires $N = 20$ frames at the input and creates a different $\hat{\boldsymbol{I}}_\mathrm{a}$ if the frame order changes. Since the video is written as a set $\mathcal{V}$, the DAN [178] is unsuitable for this task, as more sophisticated techniques are required to account for the permutation invariance and the arbitrary number of input frames N.

As introduced in Section 6.1, almost all related works perform the aggregation in the feature space. Unlike face aggregation, feature aggregation is a comparatively simple task since averaging all features, as in Equation (6.5), already provides a solid baseline. For face aggregation, performing such a naive aggregation with all frames $\boldsymbol{I}_n \in \mathcal{V}$ would not yield any satisfying result, even though all $\boldsymbol{I}_n$ depict the same identity. Therefore, it is necessary to provide the model with the ability to exchange information at multiple depths of the aggregation. Only then it is possible to generate a photo-realistic image $\hat{\boldsymbol{I}}_\mathrm{a}$.

In order to design a permutation-invariant face aggregation network that can handle an arbitrary number of frames N, Hörmann *et al.* [6†] modified a U-Net [179] by incorporating the global concatenation layer [2] for information exchange. The detailed architecture of the aggregation network $\mathrm{A}(\cdot)$ is illustrated in Figure 6.1.

Hörmann *et al.* [6†] proposed to process all N input frames $\boldsymbol{I}_n$ in parallel by a U-net with shared weights. In this way, the architecture can be adapted to any arbitrary number of input frames N. The global concatenation is employed to exchange information between branches [2]. For N input feature tensors $\boldsymbol{X}_n \in \mathbb{R}^{H \times W \times C}$, which are extracted from the corresponding input images $\boldsymbol{I}_n$, the output $\boldsymbol{Y}_n \in \mathbb{R}^{H \times W \times 2C}$ is computed by

$$\boldsymbol{Y}_n = \boldsymbol{X}_n \oplus \max_m (\boldsymbol{X}_m) \,. \tag{6.8}$$

Thus, the current feature map $\boldsymbol{X}_n$ is concatenated with a feature map of similar size, which contains the maximum activation of all N feature maps $\boldsymbol{X}_m$. In this way, local information is maintained in the first C feature maps, whereas the remaining C feature maps contain global information. In the decoder of the U-Net, the encoder feature maps form an additional input to the global concatenation. In this case, the encoder feature maps are concatenated such that $\boldsymbol{Y}_n \in \mathbb{R}^{H \times W \times 3C}$ and are not considered in the maximum pooling as it already took place in the encoder. Integrating global concatenation layers at multiple depths of the aggregation network enables various opportunities for back-and-forth information exchange between the branches. Therefore, every branch is always aware of the content of the remaining branches and can leverage this information to complement missing information.

With the help of the global concatenation, the U-Net is structured as follows (see Figure 6.1): In the encoder, four 3×3 convolutional layers with stride $S = 2$ downsample the input of resolution 112×112 to 7×7. Before every downsampling, the global concatenation pools global information from all branches and a subsequent 1×1 convolution merges local and global information. In the decoder, the initial resolution is restored by employing four 4×4 transposed convolutional layers (see Section 2.5.2). After eight global concatenations, all branches have equalized each other such that a final maximum pooling layer is used to consolidate all information into a single branch, which outputs the final aggregated image $\hat{\boldsymbol{I}}_\mathrm{a}$. As an activation function, exponential linear unit (ELU) [27] is employed throughout the aggregation network.

6.3 Loss Functions

Besides the objectives stated in Equations (6.3) and (6.4), the aggregated image $\hat{\boldsymbol{I}}_\mathrm{a}$ must be realistic. Therefore, Hörmann *et al.* [6†] trained the aggregation network as the generator of a generative adversarial network (GAN) (*cf.* Section 2.7). Overall, the total loss of the aggregation network $\mathscr{L}_\mathrm{tot}^\mathrm{G}$ is calculated by

$$\mathscr{L}_\mathrm{tot}^\mathrm{G} = \lambda_\mathrm{dis}\mathscr{L}_\mathrm{dis} + \ \lambda_\mathrm{rec}\mathscr{L}_\mathrm{rec} + \lambda_\mathrm{adv}\mathscr{L}_\mathrm{adv}^\mathrm{G} + \lambda_\mathrm{tv}\mathscr{L}_\mathrm{tv}, \tag{6.9}$$

where the scalars λ are used to balance the losses.

While the adversarial loss $\mathscr{L}_\mathrm{adv}^\mathrm{G}$ and the total variation loss $\mathscr{L}_\mathrm{tv}$ force the network to generate a realistic $\hat{\boldsymbol{I}}_\mathrm{a}$, the discriminative loss $\mathscr{L}_\mathrm{dis}$ and the reconstruction loss $\mathscr{L}_\mathrm{rec}$ ensure that the identity of $\mathcal{V}$ is maintained in $\hat{\boldsymbol{I}}_\mathrm{a}$. For $\mathscr{L}_\mathrm{dis}$ and $\mathscr{L}_\mathrm{rec}$, a feature extractor $\mathrm{F}(\cdot)$ is required. A ResNet [72] with depth $L = 50$ and trained with additive angular margin

loss [33] is employed to obtain meaningful features. This model is named *MS-112-Arc* and is analyzed exhaustively in Section 3.6.2.

6.3.1 Discriminative Loss

The discriminative loss was employed by Rao *et al.* [178] in the DAN to guarantee more discriminative features from $\hat{\boldsymbol{I}}_\mathrm{a}$ as desired in Equations (6.3) and (6.4). In addition to the feature of the aggregated image $\boldsymbol{f}_\mathrm{a}$, features are extracted from all frames $\boldsymbol{I}_\mathrm{m} \in \mathcal{V}_i^{(n)}$ of the ith identity. Moreover, the feature $\boldsymbol{f}_\mathrm{p}$ of a frame $\boldsymbol{I}_\mathrm{p} \in \mathcal{V}_i^{(\tilde{n})}$, *i.e.*, from a different video $n \neq \tilde{n}$ of the same identity, and the feature $\boldsymbol{f}_\mathrm{n}$ of a frame $\boldsymbol{I}_\mathrm{n} \in \mathcal{V}_j^{(\tilde{n})}$, *i.e.*, from a video of a different identity $i \neq j$ are considered. Then, the discriminative loss $\mathscr{L}_\mathrm{dis}$ is formulated as a triplet loss (*cf.* Equation (3.8) and [79, 187, 217]) by

$$\mathscr{L}_\mathrm{dis} = \Big[\big\|\boldsymbol{f}_\mathrm{a} - \boldsymbol{f}_\mathrm{p}\big\|^2 - \alpha\Big]_+ + \big[\beta - \|\boldsymbol{f}_\mathrm{a} - \boldsymbol{f}_\mathrm{n}\|^2\big]_+ , \tag{6.10}$$

where α and β denote margins, and $[\cdot]_+ = \max(0, \cdot)$. While the β is typically constant during training, α is calculated adaptively by

$$\alpha = \min_m \big\|\boldsymbol{f}_m - \boldsymbol{f}_\mathrm{p}\big\|^2 . \tag{6.11}$$

The first term in Equation (6.10) guarantees that $\hat{\boldsymbol{I}}_\mathrm{a}$ is closer to $\boldsymbol{I}_\mathrm{p}$ in the feature space than all N frames $\boldsymbol{I}_\mathrm{m} \in \mathcal{V}_i^{(n)}$ (*cf.* Equation (6.3)). At the same time, the second term ensures that the feature distance between $\hat{\boldsymbol{I}}_\mathrm{a}$ and $\boldsymbol{I}_\mathrm{n}$ is maximized (*cf.* Equation (6.4)).

6.3.2 Reconstruction Loss

As elaborated in Section 3.4.1, triplet loss alone does not suffice in obtaining a well-generalizing network since dataset-wide feature information representing an identity is missing. To consider this information during training, two distinct reconstruction losses are analyzed. The reconstruction loss proposed by Rao *et al.* [178] promotes intra-class compactness by minimizing the feature distance between $\boldsymbol{f}_\mathrm{a}$ and the feature center of $\mathcal{V}$. Mathematically, this can be formulated by

$$\mathscr{L}_\mathrm{rec}^\mathrm{avg} = \Big\|\boldsymbol{f}_\mathrm{a} - \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{f}_n\Big\|^2 . \tag{6.12}$$

While Equation (6.12) mitigates the influence of low-quality images $\boldsymbol{I}_m \in \mathcal{V}$, it only considers a single video. Despite including N features, the variance within a video is minimal since all images are captured under similar conditions regarding the distance to the subject, illumination, occlusions, and accessories. Hence, global features encompassing multiple videos are necessary to truly guide the network to embed information that represents the identity.

To overcome this limitation, Hörmann *et al.* [6†] adapted the center loss [229] to face aggregation. In contrast to considering the distance between the $\boldsymbol{f}_\mathrm{a}$ and the center of $\mathcal{V}$, the distance to a dataset-wide center $\boldsymbol{c}^i$ of the ith identity is minimized by

$$\mathscr{L}_\mathrm{rec}^\mathrm{cen} = \left\| \boldsymbol{f}_\mathrm{a}^i - \boldsymbol{c}^i \right\|^2 . \tag{6.13}$$

To obtain relevant centers for every identity, all $\boldsymbol{c}^i$ are updated during training by

$$\boldsymbol{c}^i \leftarrow (1-\eta_\mathrm{c})\boldsymbol{c}^i - \frac{\eta_\mathrm{c}}{N+1}\left[\boldsymbol{f}_\mathrm{p} + \sum_{n=1}^{N} \boldsymbol{f}_n\right], \tag{6.14}$$

with η_c denoting the learning rate of the centers. Since the center $\boldsymbol{c}^i$ contains all information of the ith identity, it can be considered a global memory and thus operates similarly to the trainable class representatives $[\boldsymbol{W}]_{:,i}$ of class-level losses (*cf.* Section 3.4.2).

While $\mathscr{L}_\mathrm{rec}^\mathrm{avg}$ is limited to the current video $\mathcal{V}$, $\mathscr{L}_\mathrm{rec}^\mathrm{cen}$ updates and leverages global features stored in $\boldsymbol{c}^i$. In this way, $\mathscr{L}_\mathrm{rec}^\mathrm{cen}$ reduces the influence of outliers in $\mathcal{V}$ and thus is more robust.

6.3.3 Adversarial Loss

In order to obtain a realistic aggregated image $\hat{\boldsymbol{I}}_\mathrm{a}$, the global discriminator $\mathrm{D}(\cdot)$, presented in Section 4.4.3, is employed and trained to judge whether its input $\hat{\boldsymbol{I}}_\mathrm{a}$ or $\boldsymbol{I}_\mathrm{p}$ is real or generated by the aggregation network. Following Equations (4.14) and (4.15), the adversarial losses are computed by

$$\mathscr{L}_\mathrm{adv}^\mathrm{G} = -\log(\mathrm{D}(\hat{\boldsymbol{I}}_\mathrm{a})). \tag{6.15}$$

$$\mathscr{L}_\mathrm{adv}^\mathrm{D} = -\log(1-\mathrm{D}(\hat{\boldsymbol{I}}_\mathrm{a})) - \log(\mathrm{D}(\boldsymbol{I}_\mathrm{p})). \tag{6.16}$$

6.3.4 Total Variation Loss

As discussed in Section 2.5.2, transposed convolutions are prone to cause checkerboard artifacts – especially if their kernel size $K_\mathrm{H} = K_\mathrm{W}$ is not a multiple of the stride [165]. Thus, Hörmann *et al.* [6†] employed transposed convolutions with $K_\mathrm{H} = K_\mathrm{W} = 4$ and $S = 2$ for the upsampling in order to mitigate checkerboard artifacts. In order to punish the creation of even subtle checkerboard artifacts, Hörmann *et al.* [6†] incorporated the total variation loss $\mathscr{L}_\mathrm{tv}$ from [99, 127, 143], which is calculated by

$$\mathscr{L}_\mathrm{tv} = \sum_{x=1}^{112}\sum_{y=1}^{111} \left\| [\hat{\boldsymbol{I}}_\mathrm{a}]_{x,y+1,:} - [\hat{\boldsymbol{I}}_\mathrm{a}]_{x,y,:} \right\|_1 + \sum_{x=1}^{111}\sum_{y=1}^{112} \left\| [\hat{\boldsymbol{I}}_\mathrm{a}]_{x+1,y,:} - [\hat{\boldsymbol{I}}_\mathrm{a}]_{x,y,:} \right\|_1 . \tag{6.17}$$

By punishing large gradients between neighboring pixels, which are ubiquitous if checkerboard artifacts arise, their generation by the transposed convolutions is hampered. However, the punishment of large gradients also smooths $\hat{\boldsymbol{I}}_\mathrm{a}$. Therefore, $\mathscr{L}_\mathrm{tv}$ can only be employed as a small regularizer, which should never dominate the total loss $\mathscr{L}_\mathrm{tot}^\mathrm{G}$.

6.4 Experiments

6.4.1 Training Details

VoxCeleb2 [26] is utilized as a video FR dataset (see also Table 3.1) and comprises 150k videos of 6112 identities with a total duration of 2442 hours. Since this dataset was initially proposed for speaker identification, it only contains frames of speaking persons, and thus the 150k videos are further split into 1.1M utterances. Five frames are extracted per utterance to obtain a roughly equal frame distribution, which results in 5.3M frames corresponding to 0.64 frames per second. Note that the frames per second vary depending on the length of the utterance. In this way, the data imbalance is reduced as otherwise, more frames would be extracted from longer utterances.

The frames are then aligned like during the training of the FR network, *i.e.*, the face alignment policy (FAP) following Deng *et al.* [33] is used with the landmarks extracted from the multi-task CNN (MTCNN) [265]. In addition to the contrast, brightness, and saturation augmentation and horizontal flipping (see Section 3.5.1), which are applied to the entire video, frame-wise motion blur with a filter size $\in \{7, 9, 11\}$ at a random angle is performed with a probability of $p_{\text{aug}} = 0.5$. Even though the architecture can process videos comprising an arbitrary number of frames N, all batches contain $N_{\text{b}} = 6$ videos of exactly $N = 10$ frames to make the training more efficient. All frames are selected from distinct utterances or repeated if there are not enough utterances in a video. Overall, every video is used once per epoch, *i.e.*, not every frame is exposed in every epoch to the network. Using $N = 10$ frames, also ensures comparable settings as when training the DAN. Compared to the DAN, no additional pretraining on mean squared error (MSE) loss is necessary to initialize the weights of the face aggregator, simplifying the overall training while still obtaining good convergence.

The face aggregation network is trained in an alternating manner with the discriminator using adaptive moment estimation (Adam) optimizer [110] for six epochs. The loss terms are balanced with $\lambda_{\text{dis}} = \lambda_{\text{adv}} = 1$, $\lambda_{\text{rec}} = 0.5$, and $\lambda_{\text{tv}} = 10^{-4}$ and the center loss $\mathscr{L}_{\text{rec}}^{\text{cen}}$ as reconstruction loss. The margin of $\mathscr{L}_{\text{disc}}$ is set to $\beta = 32$ and the learning rate of the center loss to $\eta = 0.5$. Initially, the learning rate η is set to $5 \cdot 10^{-5}$ and decreased to $\eta = 1.25 \cdot 10^{-5}$ after 3 epochs.

6.4.2 Evaluation Details

Three benchmark datasets from Table 3.2 are suitable to evaluate the face aggregation network.

First, the YouTubeFaces (YTF) dataset is used, which comprises 3425 videos of 1595 identities. Every video contains between 48 and 6070 frames. Even though the face aggregation network can handle any arbitrary number of faces N, every video is resampled to $N \cdot N_{\text{avg}}$ frames as proposed by Rao *et al.* [178]. In this way, a direct comparison between the presented face aggregation network and the DAN is possible. After resampling, N consecutive frames are aggregated into N_{avg} aggregated images $\hat{\boldsymbol{I}}_{\text{a},m}$. Then, features are extracted from every $\hat{\boldsymbol{I}}_{\text{a},m}$ using $\text{F}(\cdot)$ and the final feature vector

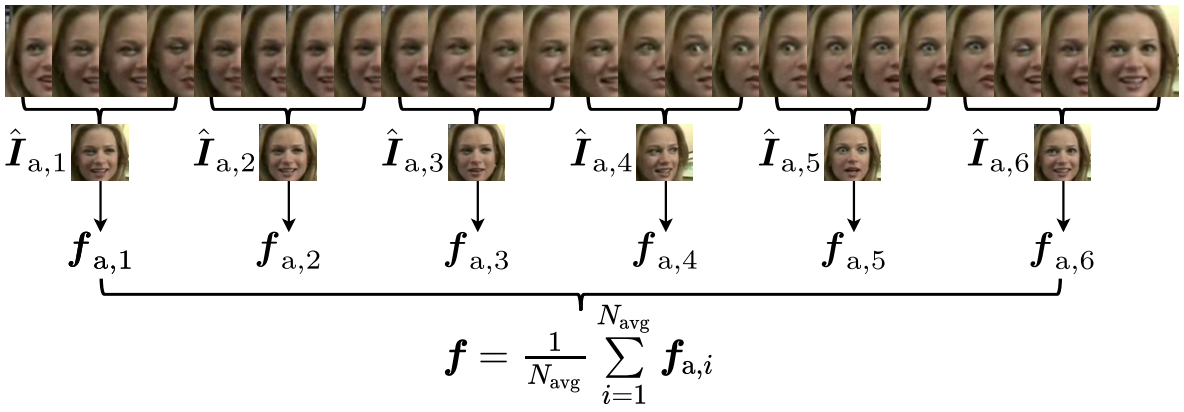

Figure 6.2: Evaluation protocol for video face aggregation for $N = 4$ and $N_{\text{avg}} = 6$.

representing the video is obtained by averaging all N_{avg} features $\boldsymbol{f}_{\text{a},m}$. This process in illustrated in Figure 6.2 for $N = 4$ and $N_{\text{avg}} = 6$.

Second, the IARPA Janus Benchmark (IJB)-B and IJB-C benchmarks are employed to evaluate face identification on the 1:N mixed media protocol. Both consider templates, *i.e.*, sets containing a mix of still images and video frames. While the general analysis in Section 3.6.2.2 assumed that every feature is equally relevant regardless of whether it is a still image or a video frame, multiple feature aggregation models employ hierarchical aggregation. *I.e.*, first, all frames are aggregated separately to obtain a representative feature vector for the entire video. Then, all features from the still images together with the aggregated feature vectors from the videos are aggregated. In this way, an entire video is considered equally relevant as a single still image, removing the previous bias towards the low-quality video frames due to their large number. Besides, their disjoint galleries allow the evaluation of open-set and closed-set face identification performance.

6.5 Results

The presented face aggregation network is compared to the average of all $N \cdot N_{\text{avg}}$ features denoted by MS-112-Arc. In addition, the DAN, a permutation-variant aggregation network, is trained on VoxCeleb2 using the MS1M-112-Arc model as the feature extractor. Thus, a fair comparison is guaranteed. For better differentiation with the original work [178], this version of the DAN is denoted as DAN*.

On both IJB benchmarks, averaging all features (MS-112-Arc) as in Section 3.6.2.2 and the hierarchical averaging (MS-112-Arc†) are considered. First, the DAN and the face aggregation network aggregate all video frames into a single frame. Then, all features are averaged to obtain the final feature for matching. Even though it is possible to aggregate different still images, the huge variance between still images and the exclusive training on videos results in unusable aggregate images.

Table 6.1: Ablation study on distinct loss functions. The verification *Acc* [%] is reported on the YTF dataset following the protocol introduced in Section 6.4.2. The last column denotes the average *Acc* over all protocols. To obtain the *Acc* marked with † input frames were duplicated to circumvent the limitation of the DAN's architecture. The highlighted model is analyzed in detail in the following subsections. Adapted from [6†].

Losses					$N =$	1	5	10				
$\mathscr{L}_{\text{adv}}$	$\mathscr{L}_{\text{dis}}$	$\mathscr{L}_{\text{rec}}^{\text{avg}}$	$\mathscr{L}_{\text{rec}}^{\text{cen}}$	$\mathscr{L}_{\text{tv}}$	$N_{\text{avg}} =$	2	2	2	4	6	8	Avg
MS-112-Arc						95.80	96.40	96.32	96.34	96.34	96.52	96.29
DAN*						95.44†	96.10†	96.24	96.48	96.20	96.56	96.17
✗	✓	✓	✗	✓		95.58	95.90	96.26	96.42	96.34	**96.62**	96.19
✗	✓	✗	✓	✓		95.52	96.08	96.24	96.44	96.28	96.42	96.16
✗	✓	✗	✗	✓		95.44	95.72	96.40	96.42	96.42	96.36	96.13
✓	✓	✓	✗	✓		95.58	96.44	96.36	96.44	96.28	96.44	96.26
✓	✓	✗	✓	✓		**95.90**	**96.52**	**96.56**	**96.60**	96.48	**96.62**	**96.45**
✓	✓	✗	✗	✓		95.58	96.26	96.40	96.58	**96.62**	96.52	96.33
✓	✓	✗	✓	✗		95.58	96.14	96.42	96.40	96.60	96.58	96.29

6.5.1 Ablation Study

Table 6.1 reports the impact of various losses on the video face verification performance measured by the accuracy (*Acc*) (see Section 3.5.2.1). The results follow the findings from Rao *et al.* [178] in that the adversarial Loss $\mathscr{L}_{\text{adv}}$ is vital to obtain the best *Acc*. Besides, slight improvements are identified by adding the total variation loss $\mathscr{L}_{\text{tv}}$ and using the global center loss $\mathscr{L}_{\text{rec}}^{\text{cen}}$ instead of the local $\mathscr{L}_{\text{rec}}^{\text{avg}}$, which was employed in the DAN [178].

As expected, increasing the number of aggregated frames N improves the *Acc* as more information is leveraged from more input data. For a single frame at the input $N = 1$, the face aggregation network yields roughly the same results, *i.e.*, no undesired artifacts are introduced during the "aggregation". When the face aggregation networks aggregate the information of $2 \cdot 10$ frames ($N = 10$, $N_{\text{avg}} = 2$) into barely two images whose features are averaged, the MS-112-Arc baseline averages all 20 features. Considering this, the remarkable result of the aggregation is apparent when comparing the results of the face aggregation networks (96.56%) with the MS-112-Arc when averaging two features, *i.e.*, 95.80%. Thus, the frames extracted from both aggregated images contain richer features for the following face verification. For $N_{\text{avg}} > 4$, no further increase is observed and the slight variance in *Acc* is likely attributed to different input frames due to different resampling.

Overall, the model with $\mathscr{L}_{\text{rec}}^{\text{cen}}$ yields the best results across all benchmark protocols and, from now on, is referred to as the *permutation-invariant face aggregation network* (PIFAN). While the DAN outputs a different aggregate image $\hat{\boldsymbol{I}}_{\text{a}}$ depending on the order of the input frames, the PIFAN creates identical $\hat{\boldsymbol{I}}_{\text{a}}$ regardless of the frame order. Despite the added complexity of the inherent permutation invariance, the face aggregation network yields superior results than the DAN and the MS-112-Arc baseline. Compared to MS-112-Arc, the face aggregation even provides the additional benefit of fusing the information of $N \cdot N_{\text{avg}}$ frames into N_{avg} aggregated images $\hat{\boldsymbol{I}}_{\text{a},m}$.

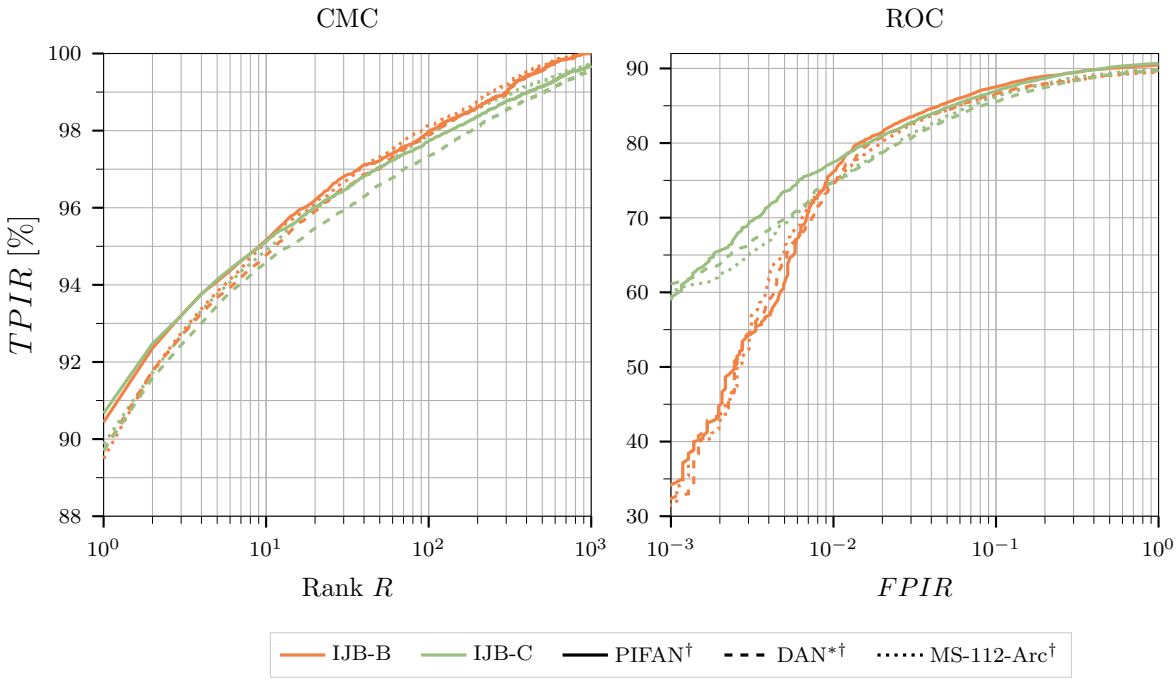

Figure 6.3: The cumulative match characteristic (CMC) and the receiver operating characteristic (ROC) at rank $R = 1$ for selective methods (line style) on the IJB-B/C mixed media identification protocols (line color). † denotes the use of hierarchical aggregation (*cf.* Section 6.4.2).

6.5.2 Detailed Analysis

6.5.2.1 Face Identification

Figure 6.1 and Table 6.2 report the face identification performance using the hierarchical aggregation as introduced in Section 6.4.2. The results demonstrate that hierarchical aggregation is a fast and straightforward way of improving the performance by leveraging the typically lower quality of video frames compared to still images.

It is apparent that the PIFAN outperforms all baselines on both datasets for the true positive identification rate ($TPIR$) at rank $R = 1$ by almost a 1% margin. Even though only the aggregated feature vectors of the videos vary between MS-112-Arc*, DAN*, and PIFAN, the results prove that it is decisive to intelligently aggregate the frames since outliers seem to considerably deteriorate the naive feature average. For larger ranks R, this advantage diminishes. The receiver operating characteristic (ROC) demonstrates that the PIFAN provides superior $TPIR$s for almost all false positive identification rates ($FPIR$s) on both datasets.

Besides the remarkable performance of the PIFAN, the face aggregation reduces the computational cost of feature extraction from 228k to 52.2k (−77.1%) on IJB-B and from 449k to 89.5k (−80.9%) on IJB-C.

Table 6.2: Detailed evaluation on the IJB-B/C mixed media identification benchmarks. All $TPIR$s are reported in % and both galleries are averaged. † denotes the use of hierarchical aggregation (*cf.* Section 6.4.2).

	IJB-B						IJB-C					
	$TPIR@R=$			$TPIR@FPIR=$			$TPIR@R=$			$TPIR@FPIR=$		
Method	1	10	100	10^{-3}	10^{-2}	10^{-1}	1	10	100	10^{-3}	10^{-2}	10^{-1}
MS-112-Arc	88.03	94.03	97.72	**35.98**	72.88	85.35	87.06	93.09	97.19	47.98	64.70	82.37
MS-112-Arc†	89.49	95.13	**98.14**	32.40	75.33	87.22	89.87	94.90	**97.88**	60.23	74.69	86.78
DAN*†	89.69	94.77	97.90	32.84	74.44	87.11	89.71	94.58	97.34	60.19	74.89	85.96
PIFAN†	**90.44**	**95.16**	97.98	30.02	**76.28**	**88.15**	**90.67**	**95.13**	97.74	**61.08**	**77.62**	**87.42**

Table 6.3: Verification *Acc* [%] for $N = 10$ and $N_{\text{avg}} = 4$ on the YTF dataset. N_{blur} denotes the number of frames affected by motion blur.

Filter Size	N_{blur}	✗	3	5	7	9	11	13	15	17	19
MS-112-Arc	9	96.34	96.42	96.34	96.28	95.90	95.40	94.78	93.76	92.66	91.64
DAN*	9	96.48	**96.44**	96.16	96.48	95.68	94.48	93.92	91.28	89.44	87.14
PIFAN	9	**96.60**	96.34	**96.40**	**96.50**	**96.48**	**96.40**	**96.18**	**95.82**	**95.92**	**95.84**
MS-112-Arc	10	96.34	**96.46**	**96.38**	95.84	**95.24**	**94.64**	**93.50**	**91.82**	**89.92**	**88.38**
DAN*	10	96.48	96.34	96.04	**95.96**	95.16	94.26	92.74	89.90	88.54	86.10
PIFAN	10	**96.60**	**96.46**	96.22	95.74	95.10	94.36	93.22	91.04	89.90	88.06

6.5.2.2 Robustness Analysis

In order to analyze the robustness, N_{blur} out of N frames are affected with motion blur. The *Acc* on the YTF dataset for two cases are analyzed in Table 6.3: All $N_{\text{blur}} = N = 10$ frames and $N_{\text{blur}} = 9$ out of $N = 10$ frames are tampered with motion blur. This analysis exposes the strengths and weaknesses of the PIFAN. While the PIFAN demonstrates astounding results if a single frame is left untouched, it struggles to match the MS-112-Arc baseline if all frames are affected by motion blur. Still, the *Acc* is only slightly worse than MS-112-Arc for $N_{\text{blur}} = N = 10$, and substantially higher robustness than by the DAN* is observed.

Overall, this analysis suggests that the PIFAN learned to successfully identify the single frame with reliable information, whereas it struggles to combine unreliable information to synthesize an aggregate face, which contains more discriminative features than any individual input frame.

6.5.2.3 Qualitative Results

The qualitative results in Figure 6.4 depict the aggregated face $\hat{\boldsymbol{I}}_{\text{a}}$ of the DAN* and the PIFAN given $N = 10$ input frames. It is apparent that the DAN* exclusively outputs $\boldsymbol{I}_3$ regardless of its relevance. This result is only possible for a permutation-variant architecture as in the DAN* since the permutation-invariance impedes this form of mode collapse (*cf.* Section 2.7.2) by design. Thus, the DAN clearly did not learn to aggregate the faces based on their relevance but instead always forwards $\boldsymbol{I}_3$ to the output.

Figure 6.4: Qualitative results for three videos with $N = 10$ frames. For every video, $N_{\text{blur}} \in \{0, 9, 10\}$ frames are tampered with motion blur of filter size 19.

The PIFAN cannot always forward the nth image to the output as there is no order in the input. Hence, by employing a permutation-invariant architecture, Hörmann *et al.* [6†] avoided this issue. For $N_{\text{blur}} < N$, the PIFAN typically identifies the single image without motion blur $\boldsymbol{I}_1$ as the most relevant image. However, the first two videos in Figure 6.4 illustrate two scenarios where the head pose or the occlusion in the untouched $\boldsymbol{I}_1$ are considered a less valuable information source than motion-blurred frontal face. On the other hand, in the last video, the PIFAN selected $\boldsymbol{I}_1$ despite the occlusion likely due to the sharp face. Even though the PIFAN correctly identifies the frame that most likely yields the best features, it does not aggregate information from multiple low-quality frames in order to form a high-quality aggregated face $\hat{\boldsymbol{I}}_{\text{a}}$ as illustrated by all videos with $N_{\text{blur}} = N = 10$. Therefore, while the PIFAN successfully avoids selecting always the same frame, it fails to fuse information from various frames, which explains the findings provided in Section 6.5.2.2.

In retrospect, this is no huge surprise since the easiest way to satisfy the discriminator is to provide it with a high-quality image, which can be found among the input frames. Moreover, by selecting a frame with meaningful features, the discriminative loss $\mathscr{L}_{\text{dis}}$ and the reconstruction loss $\mathscr{L}_{\text{rec}}$ do not punish the aggregation network.

Table 6.4: Comparisons with state-of-the-art methods on the YTF dataset. For better comparability, the improvements of the approaches with respect to average pooling (see Equation (6.5)) is provided. All values are reported in %.

Method	*Acc*	Improvement	Method	*Acc*	Improvement
FaceNet [187]	95.12		MS-112-Arc	96.52	
ArcFace [33]	98.02		Broadface [109]	98.0	
DAN[178]	95.01	+0.89	NAN [245]	95.72	+0.52
QAN [134]	96.17	+0.71	FAN [135]	96.21	+0.51
TADPool [183]	96.4	**+1.7**	MARN [54]	96.44	+0.20
C-FAN [56]	96.50	+0.14	ADRL [177]	96.52	
REAN [55]	96.60	+0.36	PFE [189]	97.36	+0.18
G-FAN [271]	97.98	+0.72	DDL [266]	**98.18**	+0.21
DAN*	96.56	+0.04	PIFAN	96.62	+0.10

6.5.2.4 Comparison with State of the Art

A comparison with state-of-the-art approaches in video FR is outlined in Table 6.4. The PIFAN outperforms a large part of the methods. As typical for most FR datasets, the comparison is not entirely fair as either a more powerful architecture – a ResNet with depth $L = 101$ – is used as a feature extractor [33, 54, 109, 183, 266] or a higher input resolution is considered [134, 245]. Besides, the PIFAN was not further finetuned on the YTF dataset as indicated in [177, 178]. Moreover, it is not always apparent if the related works used all frames or applied resampling, which impedes a conclusive comparison. The results become more comparable by considering the improvements compared to average pooling. Still, approaches evaluated with a high *Acc* of the average pooling baseline are subjected to saturation.

The PIFAN surpasses MS-112-Arc and DAN* and thus is considered the best approach for face aggregation. Besides, with its permutation-invariant architecture, a more flexible approach is pursued as the aggregated image $\hat{\boldsymbol{I}}_{\mathrm{a}}$ should not depend on the order of the input frames. Moreover, compared to the feature aggregation approaches [54–56, 134, 135, 177, 183, 189, 245, 266, 271], $\hat{\boldsymbol{I}}_{\mathrm{a}}$ is provided as an additional output and the computation cost of the feature extraction is reduced to $\frac{1}{N}$.

6.6 Conclusion and Future Work

This chapter introduced the first face aggregation approach, which handles an arbitrary number of faces in a permutation-invariant manner. The presented permutation-invariant face aggregation network (PIFAN) combines N frames of a video $\mathcal{V}$ into a single aggregated image $\hat{\boldsymbol{I}}_{\mathrm{a}}$ and thus considers every frame's quality where it is most visible. After face aggregation, identity features are only extracted from $\hat{\boldsymbol{I}}_{\mathrm{a}}$, reducing the computational cost of the feature extraction to $\frac{1}{N}$.

The PIFAN lifts the limitations of a previous face aggregation approach by employing a permutation-invariant U-Net. With various global concatenation layers, back-and-forth information exchange between the input faces is promoted. Besides, by incorporating

the center loss, the PIFAN further learns to approximate global dataset-wide identity features.

The thorough analysis revealed that the PIFAN outperforms the previous face aggregation approach while generating the same $\hat{\boldsymbol{I}}_{\mathrm{a}}$ regardless of the frame order. Besides, the PIFAN can match up to state-of-the-art feature aggregation approaches. While the qualitative analysis demonstrated that the PIFAN identifies the frame $\boldsymbol{I}_n \in \mathcal{V}$ that provides the best face identity features, it does not fuse the information from multiple frames in order to generate an $\hat{\boldsymbol{I}}_{\mathrm{a}}$ with higher quality than all $\boldsymbol{I}_n$. Still, the robustness analysis showed that the PIFAN is robust against motion blur despite the lack of details. Besides, compared to widespread feature aggregation methods, the permutation-invariant face aggregation network (PIFAN) additionally provides the aggregated face $\hat{\boldsymbol{I}}_{\mathrm{a}}$.

Regarding the objectives stated at the beginning of this chapter, the PIFAN is permutation-invariant and successfully aggregates an arbitrary number of frames N. Even though the PIFAN does not fuse information of all frames into $\hat{\boldsymbol{I}}_{\mathrm{a}}$, it forwards the most discriminative frame $\boldsymbol{I}_n \in \mathcal{V}$ even under aggravated conditions. Considering the tremendous limitations imposed by the requirements of a permutation-invariant architecture, this endeavor has delivered a satisfying solution and represents the first step towards permutation-invariant face aggregation of sets of multiple still images.

In future works, the shortcomings of the PIFAN can be remedied to truly fuse the information instead of selecting the most discriminative frame. Incorporating an additional loss, which punishes close pixel distances between $\boldsymbol{I}_n$ and $\hat{\boldsymbol{I}}_{\mathrm{a}}$, would not solve the issue as $\hat{\boldsymbol{I}}_{\mathrm{a}}$ should always resemble at least one $\boldsymbol{I}_n$. Therefore, modifying the architecture has a higher chance of success. For example, dispensing with skip connections at large spatial resolutions impedes the direct information flow and thereby limits the pixel-wise reconstruction of a selected image. Besides, even though the authors of the global concatenation layer [2] found no difference between maximum pooling and average pooling, average pooling forces the network to truly consider all branches instead of focusing on the branch with the most prominent activations. In addition, the activations of every branch can be scaled with with their respective relevance, *i.e.*, similar to feature aggregation approaches described by Equation (6.6), or information can be pooled from various feature maps as in [8, 239].

After solving the network's focus onto a single frame, the face aggregation network can be extended to also cope with a set of still images, which is even more challenging due to the vastly varying capture conditions, background, accessories, and even age of the identity. Drawing from the idea of Karras *et al.* [104, 105], who proposed to transfer style using adaptive instance normalization (AdaIN), a spatially independent approach constitutes the most promising direction.

7

Conclusion

This work presented various solutions to obtain satisfying face recognition (FR) performance under adverse conditions. Two adverse scenarios, with which state-of-the-art FR approaches still struggle, were selected and three robust solutions were designed, implemented, and evaluated. In particular, images of faces obstructed by occlusions of different sizes and videos affected by motion blur were considered. Every solution encompasses an architecture and several loss functions – both meticulously tailored to the given scenario. At the same time, special attention was given to ensure that the approaches also accomplished comparable results in non-adverse scenarios in order to avoid designing specialized systems, which are unable to perform simple tasks.

The following paragraphs recapitulate the most important findings and draw conclusions w.r.t. the objectives from Section 1.2. The three approaches presented in Chapters 4 to 6 were:

1. The *coarse-to-fine dual attention network* (C2F-DAN) for blind face completion reconstructs a synthetically occluded face.

2. The *partial FR network* (PFRN) constitutes a robust solution to the FR even of small face patches.

3. The *permutation-invariant face aggregation network* (PIFAN) represents the first approach of permutation-invariant face aggregation.

The C2F-DAN aims at retrieving information lost due to the occlusions and thus serves as a preprocessing step prior to the FR. Only synthetic occlusions were considered to facilitate the training and ensure that the occlusion is well defined. However, the approach dispenses with the tedious annotation of the mask, which describes the occlusions, since it is predicted by the network during the reconstruction. The architecture encompasses a parallel structure of two attention modules embedded into a coarse-to-fine network. By separating the architecture into two parts – a rough prediction by the coarse network followed by a refinement step – and supervising the training with carefully selected loss functions, a realistically reconstructed face with sharp details was obtained.

The exhaustive analysis revealed that the C2F-DAN handles occlusions with multiple colors, arbitrary forms, and letters, which were not part of the data augmentation.

Moreover, the C2F-DAN reconstructed challenging scenarios, such as glasses and large head poses, or underdefined face parts like the entire mouth. Only thin lines caused failure in detecting the mask. However, this issue can be resolved by incorporating such occlusions during training. Besides, the reconstruction of both eyes only rarely created satisfying results. Despite remaining limitations, the C2F-DAN mostly fulfills the qualitative objective posed in Section 1.2 as no further alterations of non-occluded pixels were observed. Additionally, utilizing the reconstructed face, which is generated by the C2F-DAN, yielded superior performance than the state of the art on all evaluation protocols. Despite the increased difficulty of the approach, the C2F-DAN even outperformed the FR baseline without reconstruction for sparse occlusions. Hence, this endeavor was successful as proven by the exhaustive qualitative and quantitative analyses on FR performance and reconstruction quality.

In order to recognize face patches covering less than 10% of the face's area, the PFRN pools information from relevant positions in the feature maps, which were pointed out by attention maps. In this way, the PFRN is to some degree invariant to the position of the face patch within the image. Moreover, mapping the local features extracted at various positions into a joint feature space enables the comparison of non-overlapping face patches. The thorough analysis demonstrated that the PFRN outperformed all baselines and successfully mitigated the drop in FR for partial faces. Despite training exclusively with synthetic occlusions, the PFRN improved the performance of face pairs with extreme head poses, *i.e.*, with natural occlusions. Besides, the performance is maintained on a high level for holistic, *i.e.*, non-occluded faces. Overall, the results demonstrated that the PFRN provided viable results even with tiny face patches, thereby accomplishing all objectives stated in Section 1.2.

For video face recognition, a unique approach was presented with the PIFAN by aggregating the frames in the image space. With global concatenation layers, the PIFAN enables back-and-forth information exchange at various depths during aggregation. In this way, only the most relevant information is leveraged to synthesize the aggregate image. By ensuring permutation invariance in the PIFAN, the aggregated image is identical regardless of the frame order. Quantitatively, the PIFAN outperformed the previous method in permutation-variant face aggregation on all benchmark protocols and accomplished satisfying results compared to state-of-the-art video FR approaches. Despite these remarkable quantitative results, the qualitative analysis revealed that the PIFAN selected the input frame with the best facial features instead of aggregating the information of all frames. In this way, the PIFAN is robust to some degree against motion blur as it correctly identifies the frame containing the most viable information. Nevertheless, if all frames are affected by motion blur, the PIFAN fails to provide a higher quality aggregated face. In conclusion, the PIFAN surpasses all requirements set for the FR performance yet fails to truly aggregate the information. Despite its flaws, the PIFAN constitutes the first approach for permutation-invariant face aggregation and paves the way to aggregate a set of faces without any temporal dependency.

In conclusion, all three approaches effectively remedy the imposed constraints by mitigating the drop in FR performance without exacerbating the performance under normal conditions. Hence, all approaches provide a clear benefit over state-of-the-art

approaches, which suffer under adverse conditions. Besides, two approaches even provide synthesized faces as additional outputs. Future research directions of all three methods were presented individually at the end of Chapters 4 to 6. Most notably, the generalization of the C2F-DAN can be improved quite effortlessly by extending the data augmentation during training. Additionally, the information about whether a pixel was reconstructed and thus contains less reliable identity information can help the subsequent FR network refine its features. Therefore, leveraging this information within the FR network would further boost the performance. Moreover, generating an entire face based on tiny face patches could pose an alternative solution for partial FR.

Besides investigating the vulnerability of FR systems for faces with occlusion (see Chapters 4 and 5) and videos affected by motion blur (see Chapter 6), other domains can also be explored: faces with different image qualities [15†, 16†], face sets with outliers, *i.e.*, images of different identities within the set [7†], or combining auditory and visual information to compensate the degradation of one modality [9†]. While first approaches provide satisfying results, the latest advances in deep learning enable even more sophisticated methods to tailor the architecture and the loss functions more precisely to adverse conditions.

A

Appendix: Notation

This appendix briefly introduces the notation used throughout this work. The notation is based on the style manual published by the Institute of Electrical and Electronics Engineers (IEEE) [92].

References

References are divided into three groups to improve clarity:

1. Self-citations are denoted by a †, *e.g.*, [13†].
2. Citations of theses written by supervised students are marked with a $^{+}$, *e.g.*, [31^{+}].
3. Any other citations are not highlighted, *e.g.*, [14].

Multi-references are separated by a comma in alphanumerical order and thus independent of the order of appearance in the text, *e.g.*: The works of Xia [31^{+}], Kong [20^{+}], and Cao [5^{+}] employed a generative adversarial network (GAN). Or in other words, multiple works [5^{+}, 20^{+}, 31^{+}] employed a GAN. To avoid clutter from frequently recurring references, every reference is usually repeated only once per paragraph unless an attribution is ambiguous. Citing pages are listed after every reference. Besides, footnotes are marked by Roman letters.[i]

Acronyms

Acronyms are introduced (at least) once when they first appear on a per-chapter basis. Articles are assigned to the acronyms based on the pronunciation of acronym's first letter when used in the research field. *E.g.*, it is *an* face recognition (FR) system and *a* convolutional neural network (CNN), but *a* residual network (ResNet). To avoid confusion with plural usage, acronyms are suffixed by a plural "s", *e.g.*, CNNs is the plural of CNN.

[i] This is a footnote.

Mathematics

The mathematical notation follows the ISO 80000-2 standard where possible [94]. The most important aspects are summarized below:

- Scalars are written in italic lower-case letters, *e.g.*, l.
- Constant scalars are written in capital letters, *e.g.*, N.
- Vectors are written in bold lower-case letters, *e.g.*, $\boldsymbol{v}$.
- Matrices and tensors are written in bold capital letters, *e.g.*, $\boldsymbol{X}$.
- Distributions and sets are written in calligraphic capital letters, *e.g.*, $\mathcal{G}$.

In addition to ISO 80000-2, the following rules apply:

- Whenever partial and total derivatives are identical, partial notation is preferred. In other words, $\frac{\partial x}{\partial t}$ is preferred over $\frac{\mathrm{d}x}{\mathrm{d}t}$ unless the meaning changes.
- A single element x at the coordinate $\begin{pmatrix} a & b & c & d \end{pmatrix}$ of an $A{\times}B{\times}C{\times}D$ tensor $\boldsymbol{X}$ is obtained by $x = \left[\boldsymbol{X}\right]_{a,b,c,d}$. With the help of ":", all indices within are dimension are selected, *i.e.*, $\left[\boldsymbol{X}\right]_{a,:,c,d}$ denotes a vector since the ath element is chosen in the first dimension, the cth element in the third dimension, and the dth element in the fourth dimension. Besides, $\boldsymbol{X} = \left[\boldsymbol{X}\right]_{:,:,:,:}$ holds.
- A $M{\times}N$ matrix $\boldsymbol{W}$ can be written as

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_N \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,N} \\ w_{2,1} & w_{2,2} & \dots & w_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,1} & w_{M,2} & \dots & w_{M,N} \end{pmatrix}, \tag{A.1}$$

  where $\boldsymbol{w}_n = \left[\boldsymbol{W}\right]_{:,n}$ denotes the nth column vector and elements $w_{m,n} = \left[\boldsymbol{W}\right]_{m,n}$ the element at position $\begin{pmatrix} m & n \end{pmatrix}$.
- All mathematical operations are subject to automatic singleton expansion, also referred to by *broadcasting*. That is, if two tensors are included in a calculation that requires identical size, *e.g.*, a summation, their mismatching (singleton) dimensions are expanded via repetition until their dimensions match. *E.g.*, consider a scalar a and a vector $\boldsymbol{b} = \begin{pmatrix} b_1 & b_2 & \dots & b_N \end{pmatrix} \in \mathbb{R}^{1\times N}$, then $\boldsymbol{b} + a = \begin{pmatrix} b_1 + a & b_2 + a & \dots & b_N + a \end{pmatrix}$.
  Analogously, for a matrix $\boldsymbol{C} = \begin{pmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \dots & \boldsymbol{c}_N \end{pmatrix} \in \mathbb{R}^{M\times N}$,

$$\begin{aligned} \boldsymbol{C} + \boldsymbol{b} &= \begin{pmatrix} \boldsymbol{c}_1 + b_1 & \boldsymbol{c}_2 + b_2 & \dots & \boldsymbol{c}_N + b_N \end{pmatrix} \\ &= \begin{pmatrix} c_{1,1} + b_1 & c_{1,2} + b_2 & \dots & c_{1,N} + b_N \\ c_{2,1} + b_1 & c_{2,2} + b_2 & \dots & c_{2,N} + b_N \\ \vdots & \vdots & \ddots & \vdots \\ c_{M,1} + b_1 & c_{M,2} + b_2 & \dots & c_{M,N} + b_N \end{pmatrix}. \end{aligned} \tag{A.2}$$

- If not stated otherwise, the L_2 is used as the default vector norm, *i.e.*, $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{N}[\boldsymbol{x}]_i^2}$ for $\boldsymbol{x} \in \mathbb{R}^N$. However, for matrices (and tensors) the Frobenius norm is utilized, *i.e.*,

$$\|\boldsymbol{X}\| = \|\boldsymbol{X}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[\boldsymbol{X}]_{i,j}^2}, \tag{A.3}$$

  where $\boldsymbol{X} \in \mathbb{R}^{M\times N}$.

Artificial Neural Networks

The notation in artificial neural networks (ANNs) follows [162].

- The superscript $^{(n)}$ denotes the nth sample in a dataset, *e.g.*, $\mathcal{X} = \left\{\left(\boldsymbol{X}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$, where $\mathcal{X}$ is a dataset comprising N samples – here, an input matrix $\boldsymbol{X}$ together with a scalar label y.
- The superscript $^{[l]}$ describes the lth layer of an ANN.
- Predictions by the network are marked with $\hat{}$, *e.g.*, $\hat{y}$ is the prediction of a network and should ideally be close to the target label from a dataset y.
- To discern outputs before and after applying the activation function $\Psi(\cdot)$, the tilde $\tilde{}$ is used for the output without $\Psi(\cdot)$, *i.e.*, $\hat{y} = \Psi(\tilde{y})$.

In order to minimize clutter, all markers are omitted if they are not necessary to avoid ambiguity.

B

Appendix: Similarity Transformation for 2D Face Alignment

Umeyama [209] describes the least-squares estimation of the similarity transformation parameters (rotation φ, uniform scaling ζ, and translation τ_x and τ_y), which are used in this work to perform face alignment using N_LM 2D facial landmarks. Equation (3.3) can be rewritten as

$$\boldsymbol{R}^*,\, \zeta^*,\, \boldsymbol{\tau}^* = \underset{\boldsymbol{R},\,\zeta,\,\boldsymbol{\tau}}{\arg\min} \left(\frac{1}{N_\mathrm{LM}} \sum_{i=1}^{N_\mathrm{LM}} \left\| \boldsymbol{y}_i - (\zeta \boldsymbol{R} \boldsymbol{x}_i + \boldsymbol{\tau}) \right\|^2 \right), \tag{B.1}$$

with the rotation matrix $\boldsymbol{R} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$ and translation vector $\boldsymbol{\tau} = \begin{pmatrix} \tau_\mathrm{x} \\ \tau_\mathrm{y} \end{pmatrix}$, and $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ denoting the source and target vector, respectively. The mean vectors $\boldsymbol{\mu}_\mathrm{x}$ and $\boldsymbol{\mu}_\mathrm{y}$, the variance σ_x^2, and the covariance matrix $\boldsymbol{C}_\mathrm{x,y}$ are computed by

$$\boldsymbol{\mu}_\mathrm{x} = \frac{1}{N_\mathrm{LM}} \sum_{i=1}^{N_\mathrm{LM}} \boldsymbol{x}_i, \tag{B.2}$$

$$\boldsymbol{\mu}_\mathrm{y} = \frac{1}{N_\mathrm{LM}} \sum_{i=1}^{N_\mathrm{LM}} \boldsymbol{y}_i, \tag{B.3}$$

$$\sigma_\mathrm{x}^2 = \frac{1}{N_\mathrm{LM}} \sum_{i=1}^{N_\mathrm{LM}} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_\mathrm{x} \right\|^2, \tag{B.4}$$

$$\boldsymbol{C}_\mathrm{x,y} = \frac{1}{N_\mathrm{LM}} \sum_{i=1}^{N_\mathrm{LM}} (\boldsymbol{x}_i - \boldsymbol{\mu}_\mathrm{x}) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_\mathrm{y} \right)^T. \tag{B.5}$$

Moreover, the singular value decomposition of the covariance matrix $\boldsymbol{C}_\mathrm{x,y}$ is denoted as $\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, with the singular values in $\boldsymbol{D}$ being in descending order. In addition, $\boldsymbol{S} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Then, the optimal rotation matrix $\boldsymbol{R}^*$, the scaling ζ^*, and the translation

vector $\boldsymbol{\tau}^*$ minimizing Equation (B.1) are

$$\boldsymbol{R}^* = \begin{cases} \boldsymbol{U}\boldsymbol{V}^T & \text{if } (\operatorname{rank}(\boldsymbol{C}_{\mathrm{x,y}}) = 1 \;\&\; \det(\boldsymbol{C}_{\mathrm{x,y}}) \geq 0) \text{ or} \\ & \quad (\operatorname{rank}(\boldsymbol{C}_{\mathrm{x,y}}) = 2 \;\&\; \det(\boldsymbol{U})\det(\boldsymbol{V}) = 1) \\ \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T & \text{if } (\operatorname{rank}(\boldsymbol{C}_{\mathrm{x,y}}) = 1 \;\&\; \det(\boldsymbol{C}_{\mathrm{x,y}}) < 0) \text{ or} \\ & \quad (\operatorname{rank}(\boldsymbol{C}_{\mathrm{x,y}}) = 2 \;\&\; \det(\boldsymbol{U})\det(\boldsymbol{V}) = -1)\,, \end{cases} \tag{B.6}$$

$$\zeta^* = \begin{cases} \frac{1}{\sigma_{\mathrm{x}}^2}\operatorname{tr}(\boldsymbol{D}) & \text{if } \det(\boldsymbol{C}_{\mathrm{x,y}}) \geq 0 \\ \frac{1}{\sigma_{\mathrm{x}}^2}\operatorname{tr}(\boldsymbol{D}\boldsymbol{S}) & \text{if } \det(\boldsymbol{C}_{\mathrm{x,y}}) < 0, \end{cases} \tag{B.7}$$

$$\boldsymbol{\tau}^* = \boldsymbol{\mu}_{\mathrm{y}} - \zeta^* \boldsymbol{R}^* \boldsymbol{\mu}_{\mathrm{x}}. \tag{B.8}$$

References

[1] R. Abdal, Y. Qin, and P. Wonka. Image2StyleGAN++: How to Edit the Embedded Images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8296–8305, 2020. → p. 73

[2] M. Aittala and F. Durand. Burst Image Deblurring Using Permutation Invariant Convolutional Neural Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 731–747, 2018. → pp. 121, 122, 132

[3] X. An, X. Zhu, Y. Gao, Y. Xiao, Y. Zhao, Z. Feng, L. Wu, B. Qin, M. Zhang, D. Zhang, et al. Partial FC: Training 10 Million Identities on a Single Machine. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1445–1449, 2021. → p. 33

[4] Apple. About Face ID advanced technology, 2022. `https://support.apple.com/en-us/HT208108` (accessed May 25, 2022). → p. 3

[5] Apple. Use Face ID while wearing a mask with iPhone 12 and later, 2022. `https://support.apple.com/en-us/HT213062` (accessed May 25, 2022). → p. 3

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on International Conference on Machine Learning (ICML)*, pp. 214–223, 2017. → p. 28

[7] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. `arXiv:1607.06450`, 2016. → p. 24

[8] Z. Bai, R. Wang, S. Shan, and X. Chen. Local Feature Enhancement Network for Set-based Face Recognition. In *16th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8. 2021. → pp. 119, 120, 121, 132

[9] A. Bansal, C. Castillo, R. Ranjan, and R. Chellappa. The Do's and Don'ts for CNN-based Face Verification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 2545–2554, 2017. → pp. 32, 33, 36, 38

[10] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. UMDFaces: An Annotated Face Dataset for Training Deep Networks. In *IEEE International Joint Conference on Biometrics (IJCB)*, pp. 464–473, 2017. → p. 33

[11] D. Berthelot, T. Schumm, and L. Metz. BEGAN: Boundary Equilibrium Generative Adversarial Networks. `arXiv:1703.10717`, 2017. → p. 28

[12] P. Bischoff. Surveillance camera statistics: which cities have the most CCTV cameras? Comparitech, 2021. `https://www.comparitech.com/vpn-privacy/the-worlds-most-surveilled-cities/` (accessed May 25, 2022). → p. 2

[13] Cambridge Dictionary. Facial Recognition, 2022. `https://dictionary.cambridge.org/dictionary/english/facial-recognition` (accessed May 25, 2022). → p. 1

[14] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *13th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 67–74, 2018. → pp. xiii, xiv, 33, 38, 40, 42, 47, 49, 53, 61, 83, 110, 137

[15] T. F. Chan. A school in China is monitoring students with facial-recognition technology that scans the classroom every 30 seconds. Insider, 2018. `https://www.businessinsider.com/china-school-facial-recognition-technology-2018-5` (accessed May 25, 2022). → p. 3

[16] P. Chandran, D. Bradley, M. Gross, and T. Beeler. Attention-Driven Cropping for Very High Resolution Facial Landmark Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5861–5870, 2020. → p. 36

[17] J. Chang, Z. Lan, C. Cheng, and Y. Wei. Data Uncertainty Learning in Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5710–5719, 2020. → pp. 32, 45, 46

[18] J. Chen, Z. Lu, and Q. Liao. XSepConv: extremely separated convolution for efficient deep networks with large kernels. In *13th International Conference on Digital Image Processing (ICDIP)*, 2021. → p. 15

[19] Y. Chen, L. Song, Y. Hu, and R. He. Adversarial Occlusion-aware Face Detection. In *9th IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9. 2018. → p. 103

[20] Z. Chen, S. Nie, T. Wu, and C. G. Healey. High Resolution Face Completion with Multiple Controllable Attributes via Fully End-to-End Progressive Generative Adversarial Networks. `arXiv:1801.07632`, 2018. → pp. 66, 68, 69, 72, 84, 86, 87, 91

[21] L. Chi, B. Jiang, and Y. Mu. Fast Fourier Convolution. *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 4479–4488, 2020. → p. 72

[22] L. Chi, H. Zhang, and M. Chen. End-To-End Face Detection and Recognition. arXiv:1703.10818, 2017. → p. 35

[23] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8789–8797, 2018. → p. 26

[24] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8188–8197, 2020. → p. 26

[25] S. Chopra, R. Hadsell, and Y. LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 539–546, 2005. → p. 43

[26] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep Speaker Recognition. In *19th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1086–1090, 2018. → pp. 33, 125

[27] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv:1511.07289, 2015. → pp. 11, 80, 122

[28] X. Cun and C. Pun. Split then Refine: Stacked Attention-guided ResUNets for Blind Single Image Visible Watermark Removal. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 1184–1192, 2021. → pp. 68, 69, 73, 74, 79, 84, 85, 86, 87, 91, 104

[29] A. Cuthbertson. Indian police trace 3,000 missing children in just four days using facial recognition technology. INDEPENDENT, 2018. https://www.independent.co.uk/tech/india-police-missing-children-facial-recognition-tech-trace-find-reunite-a8320406.html (accessed May 25, 2022). → p. 3

[30] DeepGlint. Trillionpairs, 2019. http://trillionpairs.deepglint.com/ (accessed Aug. 23, 2021). → pp. 33, 35

[31] U. Demir and G. Unal. Patch-Based Image Inpainting with Generative Adversarial Networks. arXiv:1803.07422, 2018. → pp. 68, 70, 81

[32] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou. Sub-center ArcFace: Boosting Face Recognition by Large-scale Noisy Web Faces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 741–757, 2020. → p. 46

[33] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4690–4699, 2019. → pp. 32, 33, 34, 36, 37, 38, 39, 40, 41, 45, 46, 47, 49, 53, 54, 55, 56, 60, 61, 83, 89, 119, 123, 125, 131

[34] J. Deng, J. Guo, J. Yang, A. Lattas, and S. Zafeiriou. Variational Prototype Learning for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11906–11915, 2021. → pp. 32, 40, 45, 46, 61

[35] J. Deng, J. Guo, D. Zhang, Y. Deng, X. Lu, and S. Shi. Lightweight Face Recognition Challenge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2638–2646, 2019. → p. 32

[36] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal Loss for Deep Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 60–68, 2017. → pp. 32, 40, 43

[37] W. Deng, J. Hu, N. Zhang, B. Chen, and J. Guo. Fine-grained face verification: FGLFW database, baselines, and human-DCMN partnership. *Pattern Recognition*, 66:63–73. Elsevier, 2017. → pp. xiii, 1, 35

[38] T. DeVries and G. W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. `arXiv:1708.04552`, 2017. → p. 48

[39] R. Dey and V. N. Boddeti. 3DFaceFill: An Analysis-By-Synthesis Approach to Face Completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1586–1595, 2022. → pp. 69, 72, 84

[40] F. Ding, P. Peng, Y. Huang, M. Geng, and Y. Tian. Masked Face Recognition with Latent Part Detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2281–2289, 2020. → p. 104

[41] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. CenterNet: Keypoint Triplets for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6569–6578, 2019. → p. 19

[42] Y. Duan, J. Lu, and J. Zhou. UniformFace: Learning Deep Equidistributed Representation for Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3415–3424, 2019. → p. 45

[43] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(7), 2011. → pp. 12, 13

[44] I. Durugkar, I. Gemp, and S. Mahadevan. Generative Multi-Adversarial Networks. In *5th International Conference on Learning Representations (ICLR)*, 2017. → pp. 27, 81

[45] I. C. Duta, L. Liu, F. Zhu, and L. Shao. Improved Residual Networks for Image and Video Recognition. In *25th International Conference on Pattern Recognition (ICPR)*, pp. 9415–9422, 2021. → p. 42

[46] F. Farnia and A. Ozdaglar. Do GANs always have Nash equilibria? In *Proceedings of the 37th International Conference on International Conference on Machine Learning (ICML)*, pp. 3029–3039, 2020. → p. 28

[47] B. K. Floris Chabert, Jingwen Zhu and V. Sharma. Recognizing People in Photos Through Private On-Device Machine Learning. Apple, 2021. `https://machinelearning.apple.com/research/recognizing-people-photos` (accessed May 25, 2022). → p. 3

[48] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual Attention Network for Scene Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3146–3154, 2019. → p. 80

[49] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *Transactions on Systems, Man, and Cybernetics*, 13(5):826–834. IEEE, 1983. → p. 15

[50] M. Geng, P. Peng, Y. Huang, and Y. Tian. Masked Face Recognition with Generative Data Augmentation and Domain Constrained Ranking. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2246–2254, 2020. → p. 104

[51] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660. IEEE, 2001. → pp. xiii, 1

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, 2014. → p. 19

[53] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010. → p. 12

[54] S. Gong, Y. Shi, and A. Jain. Low Quality Video Face Recognition: Multi-Mode Aggregation Recurrent Network (MARN). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 1027–1035, 2019. → pp. 118, 119, 120, 131

[55] S. Gong, Y. Shi, and A. Jain. Recurrent Embedding Aggregation Network for Video Face Recognition. `arXiv:1904.12019`, 2019. → pp. 119, 120, 131

[56] S. Gong, Y. Shi, N. Kalka, and A. Jain. Video Face Recognition: Component-wise Feature Aggregation Network (C-FAN). In *IEEE International Conference on Biometrics (ICB)*, pp. 1–8, 2019. → pp. 118, 119, 120, 131

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, pp. 2672–2680, 2014. → pp. 4, 25, 28

[58] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher. A Closer Look at Deep Learning Heuristics: Learning rate restarts, Warmup and Distillation. In *7th International Conference on Learning Representations (ICLR)*, 2019. → p. 14

[59] R. M. Gray. Toeplitz and Circulant Matrices: A Review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239. Now Publishers, Inc., 2006. → p. 17

[60] K. Grm, V. Štruc, A. Artiges, M. Caron, and H. K. Ekenel. Strengths and weaknesses of deep learning models for face recognition against image degradations. *Biometrics*, 7(1):81–89. IET, 2017. → pp. 3, 65, 67, 68, 83

[61] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 5767–5777, 2017. → p. 28

[62] J. Guo. lightweight facial landmark models with fast coordinate regression. GitHub, 2021. `https://github.com/deepinsight/insightface/tree/master/alignment/coordinate_reg` (accessed Feb. 16, 2022). → p. 89

[63] J. Guo, J. Deng, N. Xue, and S. Zafeiriou. Stacked Dense U-Nets with Dual Transformers for Robust Face Alignment. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. → pp. 36, 38

[64] X. Guo, H. Yang, and D. Huang. Image Inpainting via Conditional Texture and Structure Dual Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14134–14143, 2021. → pp. 69, 70, 71, 78, 83, 84, 86

[65] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 87–102, 2016. → pp. 32, 33, 38

[66] Z. Guo, Z. Chen, T. Yu, J. Chen, and S. Liu. Progressive image inpainting with full-resolution residual network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2496–2504, 2019. → p. 68

[67] S. Gutta, V. Philomin, and M. Trajkovic. An Investigation into the Use of Partial-Faces for Face Recognition. In *5th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 33–38, 2002. → p. 103

[68] A. Hasnat, J. Bohné, J. Milgram, S. Gentric, and L. Chen. DeepVisage: Making face recognition simple yet with powerful generalization skills. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 1682–1691, 2017. → pp. 44, 49

[69] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1026–1034, 2015. → pp. 9, 11, 12

[70] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916. IEEE, 2015. → p. 19

[71] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. → pp. 40, 41

[72] K. He, X. Zhang, S. Ren, and J. Sun. Identity Mappings in Deep Residual Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 630–645, 2016. → pp. 40, 41, 53, 55, 83, 105, 122

[73] L. He, H. Li, Q. Zhang, and Z. Sun. Dynamic Feature Learning for Partial Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7054–7063, 2018. → pp. 103, 104

[74] L. He, H. Li, Q. Zhang, and Z. Sun. Dynamic Feature Matching for Partial Face Recognition. *Transactions on Image Processing*, 28(2):791–802. IEEE, 2019. → pp. 103, 104

[75] L. He, H. Li, Q. Zhang, Z. Sun, and Z. He. Multiscale Representation for Partial Face Recognition Under Near Infrared Illumination. In *8th IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–7, 2016. → p. 103

[76] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. AttGAN: Facial Attribute Editing by Only Changing What You Want. *Transactions on Image Processing*, 28(11):5464–5478. IEEE, 2019. → p. 4

[77] A. Hertz, S. Fogel, R. Hanocka, R. Giryes, and D. Cohen-Or. Blind Visual Motif Removal From a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6858–6867, 2019. → pp. 68, 69, 73, 75, 80, 84, 85, 90, 91, 92, 104

[78] G. Hinton, N. Srivastava, and K. Swersky. Neural Networks for Machine Learning Lecture 6a Overview of mini-batch gradient descent. Coursera - Neural Networks for Machine Learning, 2012. `http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf` (accessed Dec. 17, 2021). → pp. 12, 13

[79] E. Hoffer and N. Ailon. Deep metric learning using Triplet network. In *3rd International Conference on Learning Representations (ICLR)*, 2015. → pp. 43, 123

[80] X. Hong, P. Xiong, R. Ji, and H. Fan. Deep Fusion Network for Image Completion. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2033–2042, 2019. → pp. 68, 69, 70, 80, 83, 84, 86

[81] J. Hu, J. Lu, and Y. Tan. Robust partial face recognition using instance-to-class distance. In *IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–6, 2013. → p. 103

[82] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018. → pp. 42, 106

[83] W. Hu, Y. Huang, F. Zhang, and R. Li. Noise-Tolerant Paradigm for Training Face Recognition CNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11887–11896, 2019. → p. 32

[84] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017. → p. 42

[85] G. B. Huang. Labeled Faces in the Wild, 2018. `http://vis-www.cs.umass.edu/lfw/` (accessed Feb. 16, 2022). → p. 60

[86] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. → pp. xiii, xiv, 1, 32, 34, 35, 88, 90, 95, 102

[87] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang. CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5901–5910, 2020. → pp. 32, 33, 36, 39, 40, 45, 46, 61, 119

[88] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154. Wiley Online Library, 1962. → p. 15

[89] Z. Hui, J. Li, X. Wang, and X. Gao. Image Fine-grained Inpainting. `arXiv:2002.02609`, 2020. → pp. 66, 68, 71, 72, 75, 76, 79, 86, 87, 91

[90] H. Hukkelås, F. Lindseth, and R. Mester. Image Inpainting with Learnable Feature Imputation. In *DAGM German Conference on Pattern Recognition (GCPR)*, pp. 388–403, 2020. → pp. 68, 69, 70, 80

[91] I. Hupont and C. Fernández. DemogPairs: Quantifying the Impact of Demographic Imbalance in Deep Face Recognition. In *14th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–7, 2019. → p. 33

[92] IEEE. IEEE Editorial Style Manual for Authors, 2021. `http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Editorial-Style-Manual-for-Authors.pdf` (accessed Nov. 14, 2021). → p. 137

[93] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *Transactions on Graphics*, 36(4):1–14. ACM, 2017. → pp. 68, 70

[94] International Organization for Standardization. *ISO 80000-2: Quantities and Units: Part 2: Mathematical Signs and Symbols to be Used in the Natural Sciences and Technology*. ISO, 2009. → pp. 9, 138

[95] S. Ioffe. Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models. *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 1945–1953, 2017. → p. 24

[96] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, pp. 448–456, 2015. → p. 23

[97] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134, 2017. → pp. 26, 81, 82

[98] C. Jin, R. Jin, K. Chen, and Y. Dou. A Community Detection Approach to Cleaning Extremely Large Face Database. *Computational Intelligence and Neuroscience*, 2018:4512473:1–4512473:10. Hindawi Limited, 2018. → pp. 32, 38

[99] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 694–711, 2016. → pp. 69, 86, 124

[100] N. D. Kalka, B. Maze, J. A. Duncan, K. O'Connor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain. IJB–S: IARPA Janus Surveillance Video Benchmark. In *9th IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9, 2018. → p. 2

[101] B.-N. Kang, Y. Kim, B. Jun, and D. Kim. Attentional Feature-Pair Relation Networks for Accurate Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5472–5481, 2019. → pp. 33, 40

[102] A. Karnewar and O. Wang. MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7799–7808, 2020. → p. 27

[103] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *6th International Conference on Learning Representations (ICLR)*, 2018. → p. 26

[104] T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019. → pp. 26, 132

[105] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8110–8119, 2020. → pp. 26, 132

[106] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The MegaFace Benchmark: 1 Million Faces for Recognition at Scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4873–4882, 2016. → pp. 34, 35, 88, 90, 94

[107] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In *Proceedings of the 34th International Conference on International Conference on Machine Learning (ICML)*, pp. 1857–1865, 2017. → p. 26

[108] Y. Kim, W. Park, M.-C. Roh, and J. Shin. GroupFace: Learning Latent Groups and Constructing Group-Based Representations for Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5621–5630, 2020. → pp. 32, 36, 39, 40, 45, 46, 61

[109] Y. Kim, W. Park, and J. Shin. BroadFace: Looking at Tens of Thousands of People at Once for Face Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 536–552, 2020. → pp. 32, 36, 39, 40, 45, 46, 61, 119, 131

[110] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. → pp. 12, 13, 49, 110, 125

[111] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the Frontiers of Unconstrained Face Detection and Recognition: IARPA Janus Benchmark A. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1931–1939, 2015. → p. 35

[112] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NeurIPS)*, pp. 1097–1105, 2012. → pp. 19, 39

[113] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 365–372, 2009. → pp. 9, 60, 61

[114] H. Law and J. Deng. CornerNet: Detecting Objects as Paired Keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, 2018. → p. 19

[115] H. A. Le and I. A. Kakadiaris. UHDB31: A Dataset for Better Understanding Face Recognition across Pose and Illumination Variation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 2555–2563, 2017. → p. 3

[116] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive Facial Feature Localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 679–692, 2012. → p. 36

[117] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551. MIT Press, 1989. → pp. 11, 15, 19

[118] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. → p. 11

[119] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. In *Neural Networks: Tricks of the Trade*, pp. 9–50. 1998. → p. 23

[120] J. A. Lewis. Facial Recognition and Fear. CSIS, 2019. `https://www.csis.org/analysis/facial-recognition-and-fear` (accessed May 25, 2022). → p. 2

[121] X. Li, G. Hu, J. Zhu, W. Zuo, M. Wang, and L. Zhang. Learning Symmetry Consistent Deep CNNs for Face Completion. *Transactions on Image Processing*, 29:7641–7655. IEEE, 2020. → pp. 68, 69, 72, 83, 84, 86, 87, 91

[122] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative Face Completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3911–3919, 2017. → pp. 72, 84

[123] J. Liang, L. Niu, F. Guo, T. Long, and L. Zhang. Visible Watermark Removal via Self-calibrated Localization and Background Refinement. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 4426–4434, 2021. → pp. 68, 69, 73, 74, 79, 84, 85, 86, 87, 91, 104

[124] S. Liao, A. K. Jain, and S. Z. Li. Partial Face Recognition: An Alignment Free Approach. *Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1193–1205. IEEE, 2012. → p. 103

[125] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 136–144, 2017. → p. 82

[126] B. Liu, W. Deng, Y. Zhong, M. Wang, J. Hu, X. Tao, and Y. Huang. Fair Loss: Margin-Aware Reinforcement Learning for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10052–10061, 2019. → p. 46

[127] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image Inpainting for Irregular Holes Using Partial Convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 85–100, 2018. → pp. 66, 68, 69, 70, 83, 84, 86, 87, 88, 99, 124

[128] H. Liu, X. Zhu, Z. Lei, and S. Z. Li. AdaptiveFace: Adaptive Margin and Sampling for Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11947–11956, 2019. → p. 46

[129] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. Few-Shot Unsupervised Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10551–10560, 2019. → p. 27

[130] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song. Learning towards Minimum Hyperspherical Energy. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 6225–6236, 2018. → pp. 33, 40, 45

[131] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. SphereFace: Deep Hypersphere Embedding for Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6738–6746, 2017. → pp. 33, 40, 41, 44, 45, 61

[132] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-Margin Softmax Loss for Convolutional Neural Networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML)*, pp. 507–516, 2016. → pp. 44, 45

[133] Y. Liu, J. Pan, and Z. Su. Deep Blind Image Inpainting. In *International Conference on Intelligent Science and Big Data Engineering*, pp. 128–141, 2019. → pp. 68, 69, 73, 84, 104

[134] Y. Liu, J. Yan, and W. Ouyang. Quality Aware Network for Set to Set Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4703, 2017. → pp. 118, 119, 120, 131

[135] Z. Liu, H. Hu, J. Bai, S. Li, and S. Lian. Feature Aggregation Network for Video Face Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 990–998, 2019. → pp. 118, 119, 120, 131

[136] I. Loshchilov and F. Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *5th International Conference on Learning Representations (ICLR)*, 2017. → p. 14

[137] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are GANs Created Equal? A Large-Scale Study. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 698–707, 2018. → p. 28

[138] X. Ma, X. Zhou, H. Huang, G. Jia, Z. Chai, and X. Wei. Contrastive attention network with dense field estimation for face completion. *Pattern Recognition*, 124:108465. Elsevier, 2022. → pp. 4, 64, 69, 70, 72, 73, 83, 84, 86, 99, 104

[139] Y. Ma, X. Liu, S. Bai, L. Wang, D. He, and A. Liu. Coarse-to-Fine Image Inpainting via Region-wise Convolutions and Non-Local Correlation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3123–3129, 2019. → pp. 68, 69, 70, 74, 79, 83, 84, 86, 87, 99

[140] Y. Ma, X. Liu, S. Bai, L. Wang, A. Liu, D. Tao, and E. Hancock. Region-wise Generative Adversarial Image Inpainting for Large Missing Areas. `arXiv:1909.12507`, 2019. → pp. 68, 70, 83, 86

[141] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML)*, 2013. → pp. 11, 76, 80, 82

[142] U. Mahbub, V. Patel, D. Chandra, B. Barbello, and R. Chellappa. Partial Face Detection for Continuous Authentication. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2991–2995. 2016. → p. 103

[143] A. Mahendran and A. Vedaldi. Understanding Deep Image Representations by Inverting Them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5188–5196, 2015. → p. 124

[144] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least Squares Generative Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2794–2802, 2017. → p. 28

[145] J. Mathai, I. Masi, and W. AbdAlmageed. Does Generative Face Completion Help Face Recognition? In *IEEE International Conference on Biometrics (ICB)*, pp. 1–8, 2019. → pp. 3, 65, 67, 69

[146] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother. IARPA Janus Benchmark–C: Face Dataset and Protocol. In *IEEE International Conference on Biometrics (ICB)*, pp. 158–165, 2018. → p. 35

[147] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–137, 1943. → p. 9

[148] Q. Meng, S. Zhao, Z. Huang, and F. Zhou. MagFace: A Universal Representation for Face Recognition and Quality Assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14225–14234, 2021. → pp. 32, 33, 40, 44, 45, 46, 47, 61, 118, 119, 120

[149] D. Merget, M. Rock, and G. Rigoll. Robust Facial Landmark Detection via a Fully-Convolutional Local-Global Context Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 781–790, 2018. → pp. 17, 36

[150] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled Generative Adversarial Networks. In *5th International Conference on Learning Representations (ICLR)*, 2017. → p. 28

[151] Microsoft. Windows Hello face authentication, 2021. `https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/windows-hello-face-authentication` (accessed May 25, 2022). → p. 3

[152] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. `arXiv:1411.1784`, 2014. → p. 25

[153] D. Misra. Mish: A Self Regularized Non-Monotonic Activation Function. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020. → p. 11

[154] G. Mordido, H. Yang, and C. Meinel. Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators. arXiv:1807.11346, 2018. → p. 27

[155] Mordor Intelligence. Facial Recognition Market - Growth, Trends, COVID-19 Impact, and Forecasts (2022 - 2027), 2021. https://www.mordorintelligence.com/industry-reports/facial-recognition-market (accessed May 25, 2022). → p. 2

[156] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. AgeDB: The First Manually Collected, In-the-Wild Age Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1997–2005, 2017. → pp. 34, 35

[157] I. S. Na, C. Tran, D. Nguyen, and S. Dinh. Facial UV map completion for pose-invariant face recognition: a novel adversarial approach based on coupled attention residual UNets. *Human-centric Computing and Information Sciences*, 10(1):1–17. Springer, 2020. → p. 39

[158] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3883–3891, 2017. → p. 82

[159] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pp. 807–814, 2010. → pp. 11, 80, 106

[160] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi. EdgeConnect: Structure Guided Image Inpainting using Edge Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3265–3274, 2019. → pp. 66, 68, 69, 70, 83, 84, 86

[161] A. Ng. With facial recognition, shoplifting may get you banned in places you've never been. CNET, 2019. https://www.cnet.com/news/privacy/with-facial-recognition-shoplifting-may-get-you-banned-in-places-youve-never-been/ (accessed May 25, 2022). → p. 3

[162] A. Ng and K. Katanforoosh. Standard notations for Deep Learning. Standford Lecture: CS230 Deep Learning, 2021. https://cs230.stanford.edu/files/Notation.pdf (accessed Nov. 20, 2021). → p. 139

[163] H. W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *IEEE International Conference on Image Processing (ICIP)*, pp. 343–347, 2014. → pp. 35, 89

[164] L. M. Ngô, S. Karaoglu, and T. Gevers. Self-Supervised Face Image Manipulation by Conditioning GAN on Face Decomposition. *Transactions on Multimedia*, 24:377–385, 2022. → p. 4

[165] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and Checkerboard Artifacts. *Distill*. http://distill.pub/2016/deconv-checkerboard, 2016. → pp. 18, 70, 80, 124

[166] M. Opitz, G. Waltner, G. Poier, H. Possegger, and H. Bischof. Grid Loss: Detecting Occluded Faces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. → p. 103

[167] C. J. Parde, C. Castillo, M. Q. Hill, Y. I. Colon, S. Sankaranarayanan, J.-C. Chen, and A. J. O'Toole. Face and Image Representation in Deep CNN Features. In *12th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 673–680, 2017. → pp. 44, 46, 119

[168] U. Park, R. R. Jillela, A. Ross, and A. K. Jain. Periocular Biometrics in the Visible Spectrum. *Transactions on Information Forensics and Security*, 6(1):96–106. IEEE, 2010. → p. 103

[169] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015. → pp. 33, 40, 42, 43

[170] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016. → pp. 26, 66

[171] P. J. Phillips, A. N. Yates, Y. Hu, C. A. Hahn, E. Noyes, K. Jackson, J. G. Cavazos, G. Jeckeln, R. Ranjan, S. Sankaranarayanan, et al. Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *Proceedings of the National Academy of Sciences*, 115(24):6171–6176, 2018. → pp. 3, 60

[172] T. Qu. Facial recognition toilet paper dispensers in China put on hold as privacy concerns grow. South China Morning Post, 2020. `https://www.scmp.com/tech/article/3112836/facial-recognition-toilet-paper-dispensers-china-put-hold-privacy-concerns` (accessed May 25, 2022). → p. 3

[173] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations (ICLR)*, 2016. → p. 26

[174] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for Activation Functions. `arXiv:1710.05941`, 2017. → p. 11

[175] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained Softmax Loss for Discriminative Face Verification. `arXiv:1703.09507`, 2017. → pp. 32, 44, 45

[176] Y. Rao, J. Lin, J. Lu, and J. Zhou. Learning Discriminative Aggregation Network for Video-Based Face Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3781–3790, 2017. → pp. 119, 121

[177] Y. Rao, J. Lu, and J. Zhou. Attention-aware Deep Reinforcement Learning for Video Face Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3951–3960, 2017. → pp. 118, 120, 131

[178] Y. Rao, J. Lu, and J. Zhou. Learning Discriminative Aggregation Network for Video-Based Face Recognition and Person Re-identification. *International Journal of Computer Vision*, 127(6-7):701–718. Springer, 2019. → pp. 4, 118, 119, 121, 123, 125, 126, 127, 131

[179] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015. → pp. 26, 69, 79, 80, 121

[180] F. Rosenblatt. The perceptron. A probabilistic model for information storage and organization in the brain. *Psychological Reviews*, 65:386–408, 1958. → p. 9

[181] R. Russell, B. Duchaine, and K. Nakayama. Super-recognizers: People with extraordinary face recognition ability. *Psychonomic Bulletin & Review*, 16(2):252–257. Springer, 2009. → p. 2

[182] M.-c. Sagong, Y.-g. Shin, S.-w. Kim, S. Park, and S.-j. Ko. PEPSI: Fast Image Inpainting With Parallel Decoding Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11360–11368, 2019. → pp. 69, 71, 74, 79

[183] N. Sankaran, D. D. Mohan, S. Tulyakov, S. Setlur, and V. Govindaraju. TADPool: Target Adaptive Pooling for Set Based Face Recognition. In *16th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8. 2021. → pp. 118, 119, 120, 131

[184] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet Probabilistic Embedding for Face Verification and Clustering. In *8th IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8, 2016. → p. 43

[185] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry. How Does Batch Normalization Help Optimization? In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 2488–2498, 2018. → p. 24

[186] K. Sato, S. Shah, and J. Aggarwal. Partial Face Recognition Using Radial Basis Function Networks. In *3rd IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 288–293, 1998. → p. 103

[187] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015. → pp. 40, 43, 119, 123, 131

[188] S. Sengupta, J. C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to Profile Face Verification in the Wild. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016. → pp. 34, 35

[189] Y. Shi and A. K. Jain. Probabilistic Face Embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6902–6911, 2019. → pp. 32, 33, 40, 46, 118, 119, 120, 131

[190] Y. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain. Towards Universal Representation Learning for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6817–6826, 2020. → p. 46

[191] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations (ICLR)*, 2015. → pp. 39, 104

[192] sindresorhus. List of words for making random mnemonic sentences. GitHub, 2018. `https://raw.githubusercontent.com/sindresorhus/mnemonic-words/master/words.json` (accessed Jan. 21, 2022). → p. 66

[193] K. Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pp. 1849–1857, 2016. → p. 43

[194] L. Song, J. Cao, L. Song, Y. Hu, and R. He. Geometry-Aware Face Completion and Editing. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 2506–2513, 2019. → p. 72

[195] L. Song, D. Gong, Z. Li, C. Liu, and W. Liu. Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 773–782, 2019. → p. 104

[196] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C.-C. J. Kuo. Contextual-based Image Inpainting: Infer, Match, and Translate. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. → pp. 68, 69, 71, 78, 86, 87, 91

[197] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. → p. 22

[198] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep Learning Face Representation by Joint Identification-Verification. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, pp. 1988–1996, 2014. → pp. 9, 39, 42, 43

[199] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6398–6407, 2020. → pp. 32, 40, 43, 47, 61

[200] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face Recognition with Very Deep Neural Networks. `arXiv:1502.00873`, 2015. → pp. 40, 43

[201] Y. Sun, X. Wang, and X. Tang. Deep Learning Face Representation from Predicting 10, 000 Classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1891–1898, 2014. → p. 33

[202] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust Large Mask Inpainting with Fourier Convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2149–2159, 2022. → pp. 66, 68, 69, 72

[203] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015. → p. 40

[204] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1701–1708, 2014. → pp. 39, 42

[205] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 648–656, 2015. → p. 22

[206] L. Tran, X. Yin, and X. Liu. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1415–1424, 2017. → p. 4

[207] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86. MIT Press, 1991. → p. 1

[208] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. `arXiv:1607.08022`, 2016. → p. 24

[209] S. Umeyama. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *Transactions on Pattern Analysis and Machine Intelligence*, 13:376–380. IEEE, 1991. → pp. 37, 141

[210] W. Wan and J. Chen. Occlusion robust face recognition based on mask learning. In *IEEE International Conference on Image Processing (ICIP)*, pp. 3795–3799, 2017. → p. 104

[211] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy. The Devil of Face Recognition is in the Noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 765–780, 2018. → p. 32

[212] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive Margin Softmax for Face Verification. *Signal Processing Letters*, 25(7):926–930. IEEE, 2018. → p. 45

[213] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual Attention Network for Image Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164, 2017. → p. 42

[214] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. NormFace: L_2 Hypersphere Embedding for Face Verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 1041–1049, 2017. → p. 45

[215] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. CosFace: Large Margin Cosine Loss for Deep Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5265–5274, 2018. → pp. 33, 40, 45, 49, 61

[216] J. Wang, S. Chen, Z. Wu, and Y.-G. Jiang. FT-TDR: Frequency-guided Transformer and Top-Down Refinement Network for Blind Face Inpainting. *Transactions on Multimedia*. IEEE, 2022. → pp. 68, 69, 70, 73, 84, 85, 86, 104

[217] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning Fine-grained Image Similarity with Deep Ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1386–1393, 2014. → pp. 43, 123

[218] M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang. Racial Faces in-the-Wild: Reducing Racial Bias by Information Maximization Adaptation Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 692–702, 2019. → pp. 2, 33, 35

[219] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8798–8807, 2018. → pp. 27, 81

[220] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, 2018. → p. 77

[221] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei. Co-Mining: Deep Face Recognition With Noisy Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9358–9367, 2019. → pp. 32, 38

[222] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018. → p. 82

[223] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei. Mis-classified Vector Guided Softmax Loss for Face Recognition. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 12241–12248, 2020. → pp. 32, 45, 46

[224] Y. Wang, Y.-C. Chen, X. Tao, and J. Jia. VCNet: A Robust Approach to Blind Image Inpainting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 752–768, 2020. → pp. 68, 73, 85, 104

[225] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia. Image Inpainting via Generative Multi-column Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 331–340, 2018. → pp. 68, 70

[226] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *Transactions on Image Processing*, 13(4):600–612. IEEE, 2004. → p. 89

[227] Z. Wang and S. Ji. Smoothed dilated convolutions for improved dense prediction. *Data Mining and Knowledge Discovery*, 35(4):1470–1496. Springer, 2021. → p. 17

[228] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, et al. Masked Face Recognition Dataset and Application. `arXiv:2003.09093`, 2020. → pp. xiii, xiv, 64

[229] Y. Wen, K. Zhang, Z. Li, and Y-Qiao. A Discriminative Feature Learning Approach for Deep Face Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 499–515, 2016. → pp. 43, 49, 61, 124

[230] R. Weng, J. Lu, and Y. Tan. Robust Point Set Matching for Partial Face Recognition. *Transactions on Image Processing*, 25(3):1163–1176. IEEE, 2016. → p. 103

[231] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother. IARPA Janus Benchmark-B Face Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 90–98, 2017. → p. 35

[232] L. Wolf, T. Hassner, and I. Maoz. Face Recognition in Unconstrained Videos with Matched Background Similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 529–534, 2011. → pp. xiii, xiv, 35

[233] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. → p. 116

[234] W. Wu, M. Kan, X. Liu, Y. Yang, S. Shan, and X. Chen. Recursive Spatial Transformer (ReST) for Alignment-Free Face Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3772–3780, 2017. → p. 38

[235] X. Wu, R. He, Z. Sun, and T. Tan. A Light CNN for Deep Face Representation with Noisy Labels. *Transactions on Information Forensics and Security*, 13(11):2884–2896. IEEE, 2018. → p. 32

[236] Y. Wu and K. He. Group Normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. → p. 24

[237] Z. Wu, X. Qi, Z. Wang, W. Zhou, K. Yuan, M. Sun, and Z. Sun. ShowFace: Coordinated Face Inpainting with Memory-Disentangled Refinement Networks. `arXiv:2204.02824`, 2022. → pp. 4, 69, 70, 72, 83, 84, 86

[238] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1492–1500, 2017. → p. 42

[239] W. Xie, L. Shen, and A. Zisserman. Comparator Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 782–797, 2018. → pp. 33, 105, 106, 107, 109, 119, 120, 121, 132

[240] W. Xie and A. Zisserman. Multicolumn Networks for Face Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. → pp. 33, 38, 118, 119, 120

[241] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo. Foreground-Aware Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5840–5848, 2019. → p. 69

[242] X. Xu, Q. Meng, Y. Qin, J. Guo, C. Zhao, F. Zhou, and Z. Lei. Searching for alignment in face recognition. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 3065–3073, 2021. → pp. 38, 39, 54

[243] X. Xu, N. Sarafianos, and I. A. Kakadiaris. On Improving the Generalization of Face Recognition in the Presence of Occlusions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 798–799, 2020. → p. 104

[244] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan. Shift-Net: Image Inpainting via Deep Feature Rearrangement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–17, 2018. → pp. 68, 69, 71, 77, 80, 82

[245] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural Aggregation Network for Video Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5216–5225, 2017. → pp. 38, 118, 119, 120, 131

[246] Y. Yang, X. Guo, J. Ma, L. Ma, and H. Ling. LaFIn: Generative Landmark Guided Face Inpainting. `arXiv:1911.11394`, 2019. → pp. 69, 70, 72, 83, 84, 86

[247] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic Image Inpainting with Deep Generative Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5485–5493, 2017. → pp. 68, 73

[248] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning Face Representation from Scratch. `arXiv:1411.7923`, 2014. → p. 33

[249] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu. Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7508–7517, 2020. → pp. 66, 68, 69, 70, 71, 74, 78, 79, 84, 99

[250] X. Yin and L. Chen. Non-Deterministic Face Mask Removal Based On 3D Priors. `arXiv:2202.09856`, 2022. → pp. 64, 72, 73, 83, 86, 99, 104

[251] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-Level Domain Transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 517–532, 2016. → p. 26

[252] D. Yu, H. Wang, P. Chen, and Z. Wei. Mixed Pooling for Convolutional Neural Networks. In *9th International Conference on Rough Sets and Knowledge Technology*, pp. 364–375, 2014. → p. 19

[253] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *4th International Conference on Learning Representations (ICLR)*, 2016. → p. 17

[254] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative Image Inpainting with Contextual Attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5505–5514, 2018. → pp. 66, 68, 69, 70, 71, 74, 78, 79, 80, 82, 84

[255] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-Form Image Inpainting with Gated Convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4471–4480, 2019. → pp. 66, 68, 69, 70, 74, 79, 84, 99

[256] T. Yu, Z. Guo, X. Jin, S. Wu, Z. Chen, W. Li, Z. Zhang, and S. Liu. Region Normalization for Image Inpainting. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 12733–12740, 2020. → pp. 66, 70

[257] X. Yuan and I. K. Park. Face De-occlusion using 3D Morphable Model and Generative Adversarial Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10062–10071, 2019. → p. 64

[258] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6023–6032, 2019. → p. 48

[259] M. D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. `arXiv:1212.5701`, 2012. → pp. 12, 13

[260] Y. Zeng, J. Fu, H. Chao, and B. Guo. Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1486–1494, 2019. → pp. 63, 66, 68, 70, 71, 78, 80, 82, 92

[261] Y. Zeng, J. Fu, H. Chao, and B. Guo. Aggregated Contextual Transformations for High-Resolution Image Inpainting. *Transactions on Visualization and Computer Graphics*. IEEE, 2022. → pp. 66, 68, 69, 70, 71, 83, 84, 86

[262] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Communications of the ACM*, 64(3):107–115. ACM, 2021. → p. 20

[263] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations (ICLR)*, 2018. → p. 48

[264] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-Attention Generative Adversarial Networks. In *Proceedings of the 36th International Conference on International Conference on Machine Learning (ICML)*, pp. 7354–7363, 2019. → pp. 26, 71, 77, 120

[265] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *Signal Processing Letters*, 23:1499–1503. IEEE, 2016. → pp. 36, 39, 47, 83, 87, 110, 125

[266] M. Zhang, G. Song, H. Zhou, and Y. Liu. Discriminability Distillation in Group Representation Learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–19, 2020. → pp. 118, 119, 120, 131

[267] S. Zhang, R. He, Z. Sun, and T. Tan. DeMeshNet: Blind Face Inpainting for Deep MeshFace Verification. *Transactions on Information Forensics and Security*, 13(3):637–647. IEEE, 2017. → pp. 68, 69, 73, 84, 104

[268] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range Loss for Deep Face Recognition with Long-Tailed Training Data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5409–5418, 2017. → pp. 32, 33, 40, 43

[269] X. Zhang, X. Wang, B. Kong, Y. Yin, Q. Song, S. Lyu, J. Lv, C. Shi, and X. Li. Domain Embedded Multi-model Generative Adversarial Networks for Image-based Face Inpainting. `arXiv:2002.02909`, 2020. → pp. 68, 69, 72, 81

[270] Y. Zhang and W. Deng. Class-Balanced Training for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. → p. 32

[271] H. Zhao, Y. Shi, X. Tong, J. Wen, X. Ying, and H. Zha. G-FAN: Graph-Based Feature Aggregation Network for Video Face Recognition. In *25th International Conference on Pattern Recognition (ICPR)*, pp. 1672–1678, 2021. → pp. 118, 119, 120, 121, 131

[272] J. Zhao, Y. Cheng, Y. Xu, L. Xiong, J. Li, F. Zhao, K. Jayashree, S. Pranata, S. Shen, J. Xing, et al. Towards Pose Invariant Face Recognition in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2207–2216, 2018. → p. 39

[273] K. Zhao, J. Xu, and M.-M. Cheng. RegularFace: Deep Face Recognition via Exclusive Regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1136–1144, 2019. → pp. 33, 45

[274] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1438–1447, 2019. → pp. 68, 71, 77, 78

[275] T. Zheng and W. Deng. Cross-Pose LFW: A Database for Studying Cross-Pose Face Recognition in Unconstrained Environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, 2018. → pp. xiii, 2, 34, 35, 112

[276] T. Zheng, W. Deng, and J. Hu. Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments. `arXiv:1708.08197`, 2017. → pp. xiii, 2, 34, 35

[277] Y. Zheng, D. K. Pal, and M. Savvides. Ring Loss: Convex Feature Normalization for Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5089–5097, 2018. → p. 44

[278] Y. Zhong, R. Arandjelović, and A. Zisserman. GhostVLAD for Set-Based Face Recognition. In *14th Asian Conference on Computer Vision (ACCV)*, pp. 35–50, 2018. → pp. 118, 119, 120, 121

[279] Y. Zhong, J. Chen, and B. Huang. Toward End-to-End Face Recognition Through Alignment Learning. *Signal Processing Letters*, 24:1213–1217. IEEE, 2017. → p. 38

[280] Y. Zhong and W. Deng. Towards Transferable Adversarial Attack against Deep Face Recognition. *Transactions on Information Forensics and Security*, 16:1452–1466. IEEE, 2020. → pp. 2, 35

[281] Y. Zhong and W. Deng. Face Transformer for Recognition. `arXiv:2103.14803`, 2021. → pp. 3, 32, 33, 42, 47, 61, 65, 67, 73, 104

[282] E. Zhou, Z. Cao, and J. Sun. GridFace: Face Rectification via Learning Local Homography Transformations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. → p. 39

[283] E. Zhou, Z. Cao, and Q. Yin. Naive-Deep Face Recognition: Touching the Limit of LFW Benchmark or Not? `arXiv:1501.04690`, 2015. → p. 32

[284] T. Zhou, C. Ding, S. Lin, X. Wang, and D. Tao. Learning Oracle Attention for High-fidelity Face Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7680–7689, 2020. → pp. 4, 63, 66, 68, 69, 71, 72, 77, 78, 80, 81, 82, 83, 86, 87, 91, 92, 99

[285] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2223–2232, 2017. → p. 26

[286] M. Zhu, D. He, X. Li, C. Li, F. Li, X. Liu, E. Ding, and Z. Zhang. Image Inpainting by End-to-End Cascaded Refinement With Mask Awareness. *Transactions on Image Processing*, 30:4855–4866. IEEE, 2021. → pp. 66, 69, 70, 74, 79, 83, 84, 86

Publications

[1†] M. Babaee, Y. Zhu, O. Köpüklü, S. Hörmann, and G. Rigoll. Gait Energy Image Restoration Using Generative Adversarial Networks. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2596–2600, 2019.

[2†] E. Bosch, R. Le Houcq Corbí, K. Ihme, S. Hörmann, M. Jipp, and D. Käthner. Frustration Recognition Using Spatio Temporal Data: A Novel Dataset and GCN Model to Recognize In-Vehicle Frustration. *Transactions on Affective Computing*, pp. 1–12. IEEE, 2022.

[3†] F. Herzog, J. Chen, T. Teepe, J. Gilg, S. Hörmann, and G. Rigoll. Synthehicle: Multi-Vehicle Multi-Camera Tracking in Virtual Cities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 1–11, 2023.

[4†] F. Herzog, X. Ji, T. Teepe, S. Hörmann, J. Gilg, and G. Rigoll. Lightweight Multi-Branch Network for Person Re-Identification. In *IEEE International Conference on Image Processing (ICIP)*, pp. 1129–1133, 2021.

[5†] S. Hörmann, A. Bhowmick, M. Weiher, K. Leiss, and G. Rigoll. Face Texture Generation And Identity-Preserving Rectification. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2448–2452, 2021.

[6†] S. Hörmann, Z. Cao, M. Knoche, F. Herzog, and G. Rigoll. Face Aggregation Network For Video Face Recognition. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2973–2977, 2021. → pp. 58, 118, 121, 122, 124, 127, 130

[7†] S. Hörmann, M. Knoche, M. Babaee, O. Köpüklü, and G. Rigoll. Outlier-Robust Neural Aggregation Network for Video Face Identification. In *IEEE International Conference on Image Processing (ICIP)*, pp. 1675–1679, 2019. → pp. 58, 118, 119, 120, 121, 135

[8†] S. Hörmann, M. Knoche, and G. Rigoll. A Multi-Task Comparator Framework for Kinship Verification. In *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 665–669, 2020. → p. 3

[9†] S. Hörmann, A. Moiz, M. Knoche, and G. Rigoll. Attention Fusion for Audio-Visual Person Verification Using Multi-Scale Features. In *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 281–285, 2020. → pp. 3, 135

[10†] S. Hörmann, Z. Zhang, M. Knoche, T. Teepe, and G. Rigoll. Attention-based Partial Face Recognition. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2978–2982, 2021. → pp. 2, 3, 102, 104, 105, 106, 108, 109, 111, 112

[11†] S. Hörmann, E. Comulada Simpson, and M. Bahram. A generic Steering Wheel Torque Model using Neural Networks. In *Proceedings of the Driving Simulation Conference 2017 Europe VR*, pp. 43–50, 2017.

[12†] S. Hörmann, T. Kong, T. Teepe, F. Herzog, M. Knoche, and G. Rigoll. Face Morphing: Fooling a Face Recognition System Is Simple! `arXiv:2205.13796`, 2022.

[13†] S. Hörmann, Z. Xia, M. Knoche, and G. Rigoll. A Coarse-to-Fine Dual Attention Network for Blind Face Completion. In *16th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8, 2021. → pp. 3, 65, 66, 67, 74, 75, 76, 77, 78, 79, 80, 82, 83, 84, 86, 89, 91, 94, 96, 137

[14†] M. Knoche, M. Elkadeem, S. Hörmann, and G. Rigoll. Octuplet Loss: Make Face Recognition Robust to Image Resolution. In *17th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8, 2023.

[15†] M. Knoche, S. Hörmann, and G. Rigoll. Susceptibility to Image Resolution in Face Recognition and Training Strategies to Enhance Robustness. *Leibniz Transactions on Embedded Systems*, 8(1):01:1–01:20, 2022. → p. 135

[16†] M. Knoche, S. Hörmann, and G. Rigoll. Cross-Quality LFW: A Database for Analyzing Cross-Resolution Image Face Recognition in Unconstrained Environments. In *16th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–5, 2021. → pp. 2, 35, 135

[17†] M. Knoche, T. Teepe, S. Hörmann, and G. Rigoll. Explainable Model-Agnostic Similarity and Confidence in Face Verification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 711–718, 2023.

[18†] O. Köpüklü, M. Babaee, S. Hörmann, and G. Rigoll. Convolutional Neural Networks with Layer Reuse. In *IEEE International Conference on Image Processing (ICIP)*, pp. 345–349, 2019.

[19†] O. Köpüklü, S. Hörmann, F. Herzog, H. Cevikalp, and G. Rigoll. Dissected 3D CNNs: Temporal skip connections for efficient online video processing. *Computer Vision and Image Understanding*, 215:103318. Elsevier, 2022. → p. 117

[20†] T. Teepe, J. Gilg, F. Herzog, S. Hörmann, and G. Rigoll. Towards a Deeper Understanding of Skeleton-based Gait Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1569–1577, 2022. → p. 117

[21†] T. Teepe, A. Khan, J. Gilg, F. Herzog, S. Hörmann, and G. Rigoll. GaitGraph: Graph Convolutional Network for Skeleton-Based Gait Recognition. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2314–2318, 2021. → p. 117

Supervised Student Theses, Seminars, and Internships

[1+] T. Bartsch. Detecting Masked Faces in the Wild with LLE-CNNs. *Senior Seminar*. Technical University of Munich, 2017.

[2+] I. Belgacem. Vaccinate Your Images Against Deepfakes: Leveraging Adversarial Attacks to Fight Image-Based Abuse. *Master's Thesis*. Technical University of Munich, 2022.

[3+] A. Bhowmich. Conditional Face Image Generation and Its Application in 3d Pedestrian Modelling. *Master's Thesis*. Technical University of Munich, 2020.

[4+] Z. Cao. Comparison of Different GANs for Face Recognition. *Research Internship*. Technical University of Munich, 2019.

[5+] Z. Cao. An Image Aggregation Network for Video Face Recognition. *Master's Thesis*. Technical University of Munich, 2020. → p. 137

[6+] Y. Fang. Semi-Automatic Dataset Labeling Using Pretrained Neural Networks. *Research Internship*. Technical University of Munich, 2019.

[7+] M. Fees. Analysis of Face Attribute Information within Face Identity Features. *Engineering Internship*. Technical University of Munich, 2019. → pp. 31, 48

[8+] L. N. Gärtner. Exploring Disentangled Feature Representation Beyond Face Identification. *Senior Seminar*. Technical University of Munich, 2018.

[9+] J. Graefe. Multicolumn Networks for Face Recognition. *Senior Seminar*. Technical University of Munich, 2019.

[10+] L. Habermayr. Different Effects of Quantization Methods on Convolutional Neural Networks. *Research Internship*. Technical University of Munich, 2020.

[11+] Y. He. Performance Evaluation of GANs for Face Recognition Tasks against Image Degradations. *Master's Thesis*. Technical University of Munich, 2019.

[12+] S. Hillebrand. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. *Senior Seminar*. Technical University of Munich, 2018.

[13+] R. Hölzl. Finding Tiny Faces. *Senior Seminar*. Technical University of Munich, 2017.

[14+] A. Jesipow. Seeing Voices and Hearing Faces: Cross-modal biometric matching. *Senior Seminar*. Technical University of Munich, 2018.

[15+] P. Joppich. Burst Image Deblurring Using Permutation Invariant Convolutional Neural Networks. *Senior Seminar*. Technical University of Munich, 2019.

[16+] N. Kardileev. Enhanced Driver Gaze Classification using Video Data together with Car Bus Signals. *Master's Thesis*. Technical University of Munich, 2019.

[17+] S. J. Kim. Efficient Multi-Currency Banknotes Classification Using Binarized Neural Networks. *Bachelor's Thesis*. Technical University of Munich, 2020.

[18+] S. Kling. Facelet-Bank for Fast Portrait Manipulation. *Senior Seminar*. Technical University of Munich, 2018.

[19+] T. Kong. Face Morphing with Generative Adversarial Network. *Research Internship*. Technical University of Munich, 2021.

[20+] T. Kong. Robustness Study of Face Recognition Systems Subjected to Face Morphing Attacks. *Master's Thesis*. Technical University of Munich, 2022. → p. 137

[21+] R. S. le Houcq Corbí. Graph Convolutional Networks for Frustration Recognition of Drivers. *Master's Thesis*. Technical University of Munich, 2021.

[22+] Y. Lin. Region-based face detection with a fully-convolutional neural network. *Master's Thesis*. Technical University of Munich, 2018.

[23+] S. L. Loo. Generation of Thorax CT Images using a Generative Adversarial Neural Network. *Master's Thesis*. Technical University of Munich, 2020.

[24+] A. Moiz. Attention Based Fusion Using Feature Aggregation for Audio-Visual Person Verification. *Master's Thesis*. Technical University of Munich, 2019.

[25+] M. S. Saliya. Multi-Camera Landmark-based 3D Ear Localization. *Master's Thesis*. Technical University of Munich, 2021.

[26+] J. Sun. Influence of Data Augmentation and Pose Variations on Face Recognition. *Research Internship*. Technical University of Munich, 2020.

[27+] L. Tian. Investigating Attention Maps for Partial Face Recognition. *Research Internship*. Technical University of Munich, 2022. → p. 116

[28+] Z. Wang. Automotive Powertrain Health Prediction based on Diagnostic Trouble Code. *Master's Thesis*. Technical University of Munich, 2020.

[29+] A. Weber. Gesichtsbasierte Schauspielererkennung in Filmen. *Bachelor's Thesis*. Technical University of Munich, 2019.

[30+] S. Wolzmüller. Facial Emotion Recognition for Automotive Applications using RGB-D Information. *Master's Thesis*. Technical University of Munich, 2019.

[31+] Z. Xia. Blind Face Completion and Recognition Using Generative Adversarial Networks. *Master's Thesis*. Technical University of Munich, 2021. → pp. 89, 137

[32+] C. Yu. Evaluation of Different Face Alignment Strategies for Face Identification. *Research Internship*. Technical University of Munich, 2019. → p. 32

[33+] Y. Yuan. Joint Identification and Outlier Detection of Face Clusters Using an Aggregation Network. *Master's Thesis*. Technical University of Munich, 2018.

[34+] Z. Zhang. A Survey on Set-Based Face Recognition. *Research Internship*. Technical University of Munich, 2019.

[35+] Z. Zhang. Attention-based Partial Face Recognition. *Master's Thesis*. Technical University of Munich, 2020.

[36+] W. Zhao. Deep Aggregation Network for Set-Based Face Recognition. *Master's Thesis*. Technical University of Munich, 2021.

[37+] Y. Özşahin. Class Rectification Hard Mining for Imbalanced Deep Learning. *Senior Seminar*. Technical University of Munich, 2018.