Technische Universität München
TUM School of Computation, Information and Technology

Deep Representation Learning for Object Perception Based on Attention Mechanisms

Hu Cao

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

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Martin Bichler

Prüfer der Dissertation:

1. Prof. Dr.-Ing. habil. Alois C. Knoll
2. Prof. Dr. Guang Chen

Die Dissertation wurde am 12. 04. 2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 09. 11. 2023 angenommen.

Abstract

Deep representation learning has shown remarkable capability for learning feature representations from data. Recently, attention mechanisms have been introduced into deep representation learning to mimic the working principle of the human visual system. Attention-based methods can adaptively adjust the weights based on the input features, allowing the model to focus on object features while suppressing noisy features. For object perception, current methods are required to achieve high accuracy, efficiency, and robustness. The high performance of deep representation models heavily relies on a large model size and a large-scale dataset. It is difficult to design a model that strikes a balance between accuracy and speed.

The goal of this thesis is to improve the performance of deep representation models by using attention mechanisms in applications of object detection and grasp detection. With effective network design, the proposed approach can achieve better performance with a small increase in computational cost. First, channel attention and spatial attention are used to form a simultaneous attention refinement module (SARM) for people detection; second, channel- and pixel-based attention is applied to construct a multidimensional attention fusion network (MDAFN) to fuse valuable semantic information for grasp detection; and finally, multimodal learning with spatial attention is employed for vehicle detection and grasp detection based on event cameras. Extensive experiments demonstrate the effectiveness of the proposed methods.

All the implementations presented in this thesis have been peer-reviewed and published in international conferences and journals, confirming the originality and reliability of the work.

Zusammenfassung

Tiefes Repräsentationslernen hat eine bemerkenswerte Fähigkeit zum Lernen von Merkmalsrepräsentationen aus Daten gezeigt. Kürzlich wurden Aufmerksamkeitsmechanismen in das Deep-Representation-Learning eingeführt, um das Funktionsprinzip des menschlichen Sehsystems nachzuahmen. Aufmerksamkeitsbasierte Methoden können die Gewichte auf der Grundlage der eingegebenen Merkmale adaptiv anpassen, so dass sich das Modell auf die Objektmerkmale konzentrieren kann, während verrauschte Merkmale unterdrückt werden. Für die Objektwahrnehmung müssen die aktuellen Methoden eine hohe Genauigkeit, Effizienz und Robustheit aufweisen. Die hohe Leistung von Deep-Representation-Modellen hängt stark von einer großen Modellgröße und einem großen Datensatz ab. Es ist schwierig, ein Modell zu entwickeln, das ein Gleichgewicht zwischen Genauigkeit und Geschwindigkeit herstellt.

Das Ziel dieser Arbeit ist es, die Leistung von Deep-Representation-Modellen durch die Verwendung von Aufmerksamkeitsmechanismen in Anwendungen der Objekterkennung und der Greiferkennung zu verbessern. Mit einem effektiven Netzwerkdesign kann der vorgeschlagene Ansatz eine bessere Leistung bei einem geringen Anstieg der Rechenkosten erreichen. Erstens werden Kanalaufmerksamkeit und räumliche Aufmerksamkeit verwendet, um ein simultanes Aufmerksamkeitsverfeinerungsmodul (SARM) für die Personendetektion zu bilden; zweitens wird kanal- und pixelbasierte Aufmerksamkeit verwendet, um ein multidimensionales Aufmerksamkeitsfusionsnetzwerk (MDAFN) zu konstruieren, um wertvolle semantische Informationen für die Greiferfassung zu fusionieren; und schließlich wird multimodales Lernen mit räumlicher Aufmerksamkeit für die Fahrzeugdetektion und die Greiferfassung auf der Grundlage von Ereigniskameras eingesetzt. Ausführliche Experimente demonstrieren die Effektivität der vorgeschlagenen Methoden.

Alle in dieser Arbeit vorgestellten Implementierungen wurden von Experten begutachtet und in internationalen Konferenzen und Fachzeitschriften veröffentlicht, was die Originalität und Zuverlässigkeit der Arbeit bestätigt.

Acknowledgement

I had a wonderful time in Munich, Germany. I experienced different cultures, met people from different countries, and tasted a lot of delicious food and impressive beer. I am fortunate to be pursuing my Ph.D. degree in the *Chair of Robotics, AI, and Real-time Systems* at TU Munich. First and foremost, I am very grateful to my supervisor, Prof. Alois Knoll, for giving me this opportunity to conduct academic research in your lab and for your valuable advice, support, and assistance.

It is worth mentioning that the favorable environment provided by the *Chair of Robotics, AI, and Real-time Systems* faculty enabled my research to proceed smoothly and successfully. I am especially thankful to Amy Bücherl, who has helped me a lot in both life and administrative matters. Additionally, I am very grateful to Dr. Alexander Lenz and Ute Lomp for their dedicated work.

I would like to thank my colleagues, Xinyi Li and Genghang Zhuang. The times we spent together doing research, talking, and partying were unforgettable. I hope we can all finish our doctoral studies successfully and then move on to the next stage of our lives. I am also grateful for the help provided by my senior colleagues, Yinlong Liu, Yingbai Hu, Wenjun Liu, Liguo Zhou, Zhuangyi Jiang, and Dr. Zhengshan Bing, during my study in the lab. Furthermore, I am really appreciated by my mentor, Prof. Guangchen, for supporting my Ph.D. research.

Further, I would like to thank Dai Liu, Dongyi Sun, Yanpeng Li, Xingzhuo Yan, Yuanhao Zhong, Encheng Su, Mengyu Li, Liangwei Zhou, and Zehua Zhang, who contributed to the realization of my ideas with their respective semester or master theses.

Another awesome memory is the three months of living in Switzerland as an academic guest of ETH Zurich. I would like to sincerely thank Prof. Lothar Thiele for his supervision and guidance during my visit to the *Computer Engineering and Network Group*. I still remember how you worked with us step by step to derive mathematical formulas and how you shared birthday cake with us on your birthday. I am also thankful to Dr. Zhongnan Qu for the constructive suggestions you gave me in discussing academic problems together. During this period, I was fascinated by the beautiful landscape of Switzerland. I think I will definitely travel to Switzerland again in the future.

Finally, I am very thankful for my family. First of all, my parents, thank you for your constant love and support through all the ups and downs. I know that you will always be there for me whenever I need you. I am also very appreciative of my sisters for taking care of my parents and being there for them when I am not at home. Last but not least, I am very grateful to my girlfriend. Without your constant encouragement, tolerance, and companionship, I could not complete this long journey.

Munich, 03. 21, 2023 *Hu Cao*

Contents

List of Figures

List of Tables

List of Publications

This thesis is mainly based on the works presented in the following papers:

- **Hu Cao**, Boyang Peng, Linxuan Jia, Bin Li, Alois Knoll, and Guang Chen. Orientation-aware People Detection and Counting Method based on Overhead Fisheye Camera. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2022.

- **Hu Cao**, Guang Chen, Zhijun Li, Jianjie Lin, Alois Knoll. Residual Squeeze-and-Excitation Network with Multi-scale Spatial Pyramid Module for Fast Robotic Grasping Detection. IEEE International Conference on Robotics and Automation (ICRA), 2021.

- **Hu Cao**, Guang Chen, Zhijun Li, Qian Feng, Jianjie Lin, Alois Knoll. Efficient Grasp Detection Network with Gaussian-based Grasp Representation for Robotic Manipulation. IEEE/ASME Transactions on Mechatronics, 2022.

- Guang Chen, **Hu Cao**, Jorg Conradt, Huajin Tang, Florian Rohrbein, Alois Knoll. Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. IEEE Signal Processing Magazine, 2020.

- **Hu Cao**, Guang Chen, Jiahao Xia, Genghang Zhuang, Alois Knoll. Fusion-based Feature Attention Gate Component for Vehicle Detection based on Event Camera. IEEE Sensors Journal, 2021.

- **Hu Cao**, Guang Chen, Zhijun Li, Yingbai Hu, Alois Knoll. NeuroGrasp: Multi-modal Neural Network with Euler Region Regression for Neuromorphic Vision-based Grasp Pose Estimation. IEEE Transactions on Instrumentation and Measurement, 2022.

Other publications during doctoral study:

- Wei Li, **Hu Cao***, Jiacai Liao, Jiahao Xia, Libo Cao, Alois Knoll. Parking Slot Detection on Around-View Images Using DCNN. Frontiers in Neurorobotics, 2020.

- Genghang Zhuang, Carlo Cagnetta, Zhenshan Bing, **Hu Cao**, Xinyi Li, Kai Huang, Alois Knoll. A Biologically-Inspired Global Localization System for Mobile Robots Using LiDAR Sensor. IEEE Intelligent Vehicles Symposium (IV), 2022.

- Xinyi Li, Yinlong Liu, Venkat, **Hu Cao**, Feihu Zhang, and Alois Knoll. Globally Optimal Robust Radar Calibration in Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, 2023.

1

Introduction

1.1 Overview

Intelligent systems are expected to sense and interact with their surroundings autonomously. Excellent object perception is a prerequisite to achieving this goal. As the cornerstone of vision understanding, object perception form the basis for a wide range of applications such as intelligent video surveillance [Cao+22c], robot vision [Cao+21a; Cao+22a], autonomous driving [Cao+21b], and human-computer interaction [MCL20].

Recently, deep representation learning methods have shown an excellent ability to automatically learn feature representations from data. Deep representation learning-based methods have made significant advances in popular visual recognition competition benchmarks. Ever since the deep convolutional neural network, AlexNet [KSH17], achieved record-breaking image classification accuracy in the Large Scale Visual Recognition Challenge (ILSVRC), the research community has mainly used deep representation learning methods to solve object perception problems. Currently, deep representation learning methods have achieved great success in the fields of image classification [KSH17; He+16; HSS18], object detection [RF16; Lin+20; Ren+15], semantic segmentation [Xie+21; Cao+22d], and robotic grasp detection [Cao+21a; Cao+22a].

Inspired by the fact that humans can effectively detect salient regions in complex scenes, attention mechanisms are introduced into deep representation learning to mimic the human visual system [Guo+22]. Attention mechanisms dynamically adjust the weights based on the features of the input image. Based on such processing, the model focuses mainly on the important areas of the image and suppresses the irrelevant parts. Researchers usually apply attention mechanisms to deep representation models to improve their performance. Attention-based models have the potential to become a more powerful and general architecture for object perception.

The goal of this thesis is to improve the performance of deep representation models for object perception using attention mechanisms. Extensive experiments are conducted to analyze the effectiveness of the proposed methods. The limitations of the proposed methods are also pointed out with a critical eye. In addition, potential future directions for improvement are discussed. This provides a deeper understanding of the new methods and helps us to evaluate their effectiveness more fairly and accurately.

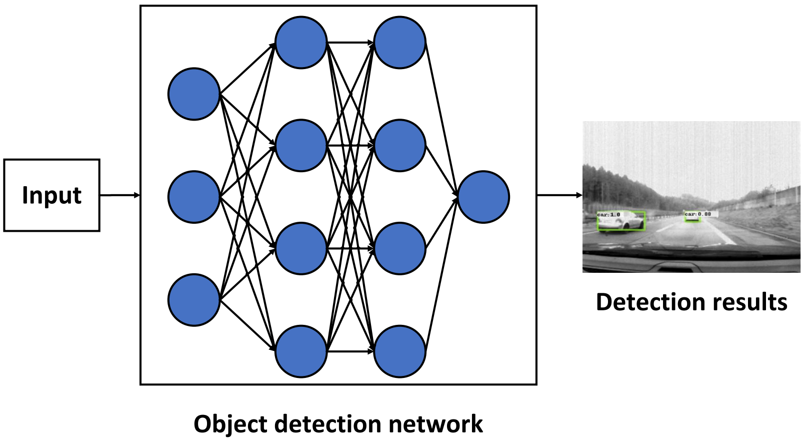


Figure 1.1: The general architecture of object detection. Image selected from the DDD 17 dataset [Bin+17]

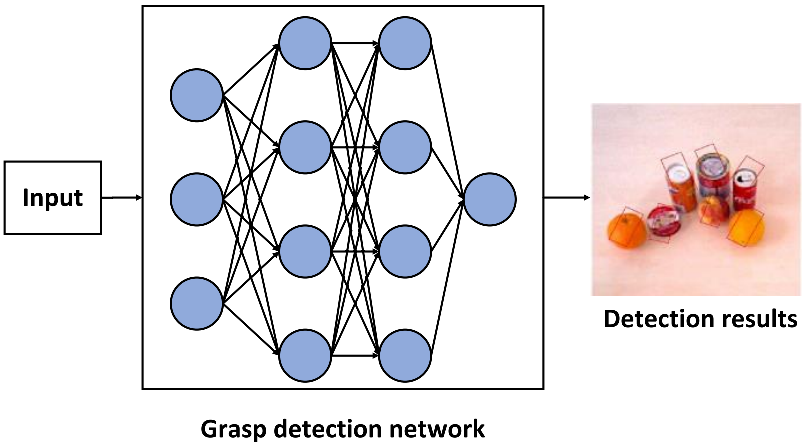


Figure 1.2: The general architecture of 2D grasp detection. Image selected from the Cornel grasp dataset [YMS11]

1.2 The Problem

The object perception problem represents the general problem of identifying or localizing the objects in an image. In this thesis, I mainly focus on object detection and robotic grasp detection.

1.2.1 Object Detection

As shown in Fig. 1.1, objects can be identified by using bounding boxes. Given an image, object detection is involved in the simultaneous estimation of the class and location of object instances.

1.2.2 Robotic Grasp Detection

In order to grasp an object, we need to obtain the direction vector between the parallel plate gripper (PPG) and the object so that the robot can approach it. In this thesis, I consider the direction normal to the surface of the workplace to be the direction vector, i.e., the gripper moves strictly perpendicular to the workplace (2D). With these settings, robotic grasp pose estimation can be considered as an detection task (robotic grasp detection). Compared to creating 3D point cloud data, the entire 2D grasping system can reduce the cost of storage and calculation. As shown in Fig. 1.2, the grasp pose of a flat object can be regarded as a grasp rectangle.

1.3 Main Challenges

Object perception algorithms are required to achieve high accuracy, efficiency, and robustness. However, current methods hardly strike a balance between accuracy and speed. Additionally, there are many imbalance problems that need to be addressed. In the following sections, the challenges of object perception algorithms are described in detail.

1.3.1 Prediction Accuracy Challenges

Intra-class variations. Each object has many instances of objects with different colors, materials, shapes, and sizes. People, for example, are a very complex object with many different postures, different types of clothing, and non-rigid deformations. Until now, accurate people detection is still a challenging problem.

Imaging condition variations. Image conditions have a significant impact on the appearance of an object in a natural environment. For instance, weather conditions, background, indoor, and viewing distance can produce a variety of object appearances such as illumination, occlusion, blur, and scale. Of these, the recognition challenges of occluded objects and small objects are particularly common.

Inter-class variations. The current category of objects owned by the world is in the order of $10^4 - 10^5$. However, current object recognition algorithms are not able to handle such a large number of object categories. Therefore, the recognition ability of current deep representation models is not yet comparable to that of humans.

1.3.2 Efficiency Challenges

In order to enhance the capability of the model, the size of the deep representation network becomes large, which leads to the demand for high computational power. However, mobile devices have limited computing power and storage space, so it is critical to design a model that directly balances efficiency and performance.

1.3.3 Imbalance Problems

Scale imbalance. In the field of object perception, the problem of scale imbalance usually arises because object instances have various sizes. In particular, at the feature extraction level, the methods using pyramid features should solve the problem of feature imbalance that exists in them. As shown in Fig. 1.3, the features from the pyramid layer have different scales of semantics. High-resolution features with shallow semantics are beneficial for detecting small objects, and low-resolution features with deep semantics are beneficial for detecting large objects. Both high-level and low-level features are complementary for object perception. It is valuable to explore how these features can be integrated together to improve the performance of object perception.

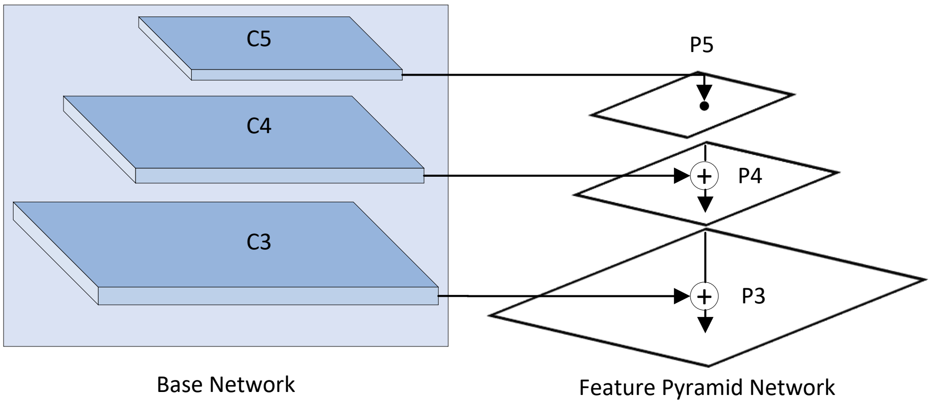


Figure 1.3: Feature imbalance is presented in the feature pyramid network (FPN) architecture.

Modality imbalance. Multi-modal data, such as RGB+depth, RGB+LiDAR, RGB+thermal, and RGB+events, is used to improve the performance of the object perception algorithms. Compared to taking unimodal data as input, multimodal data can provide more useful information for object perception algorithms in complex scenes (e.g., low light, complex backgrounds, adverse weather conditions, etc.). However, how to effectively fuse multi-modal data still needs to be explored. The modality imbalance problem in multi-modal fusion has a significant impact on the performance of the object perception algorithm. Misaligned and inadequate integration of data from different modalities does not provide benefits but rather degrades performance. It is essential to fully incorporate cross-modal complementarities to improve performance.

1.4 Key Contributions

This thesis adopts deep representation learning with attention mechanisms to address the challenge of object perception. Specifically, the thesis focuses on orientation-aware people detection, robotic grasp detection, and multimodal learning for object perception (vehicle detection and grasp detection) based on event cameras. With attention mechanisms, deep representation learning models can suppress noise features and highlight important object features, resulting in better performance.

The main contributions are concluded as follows:

- For intelligent surveillance, this thesis proposes an orientation-aware people detection and counting method based on an overhead fisheye camera. Specifically, an orientation-aware deep convolutional neural network with simultaneous attention refinement module (SARM) is introduced for people detection in arbitrary directions. Based on the attention mechanism, SARM can suppress the noise feature and highlight the object feature to improve the context-focusing ability of the network on people in different poses and orientations. Following the collection of detection results, an Internet of Things (IoT) system based on Real Time Streaming Protocol (RTSP) is constructed to output results to different devices.

- In the field of robot manipulation tasks, this thesis proposes an efficient grasp detection network with n-channel images as inputs for robotic grasp. The proposed network is a simple generative structure for grasp detection. Specifically, a Gaussian kernel-based grasp representation is introduced to encode the training samples, embodying the maximum center that possesses the highest grasp confidence. A receptive field block (RFB) is plugged into the bottleneck to improve the model's feature discriminability. In addition, pixel-based and channel-based attention mechanisms are used to construct a multi-dimensional attention fusion network (MDAFN) to fuse valuable semantic information.

- A neuromorphic vision sensor (event camera) is introduced to the field of vehicle detection and robotic grasp detection. The strengths of an event camera are that it can provide high temporal resolution, high dynamic range, low power consumption, and no motion blur. Frame-based data and event-based data are complementary. This thesis employs multimodal learning for vehicle detection and robotic grasp detection based on event cameras. The detailed contributions include: (a) This thesis introduces a fully convolutional neural network with a feature attention gate component (FAGC) for vehicle detection by combining frame-based and event-based vision. Both grayscale features and event features are fed into the feature attention gate component (FAGC) to generate the pixel-level attention feature coefficients to improve the feature discrimination ability of the network. Moreover, this thesis explores the influence of different fusion strategies on the detection capability of the network. (b) For robotic grasp detection, this thesis constructs a neuromorphic vision-based robotic grasp dataset with 154 moving objects, named *NeuroGrasp*, which is the first RGB-Event multi-modality grasp dataset (to the best of my knowledge). This dataset records both RGB frames and the corresponding event streams, providing frame data with rich color and texture information and event streams with high temporal resolution and high dynamic range. Based on the *NeuroGrasp* dataset, this thesis further develops a multi-modal neural network with a specific Euler-Region-Regression sub-network (ERRN) to perform grasping object detection. Combining frame-based and event-based vision, the proposed method achieves better performance than the method that only takes RGB frames or event streams as input.

1.5 Thesis Outline

As shown in Fig. 1.4, this thesis is structured as follows:

- In Chapter 1, this part introduces the related background of object perception. Concretly, the problem, challenges, and contributions are summarized.

- Chapter 2 illustrates the theoretical foundations and methods. It mainly includes deep representation learning, attention mechanisms and evaluation metrics.

- After that, Chapters 3, 4, and 5 are applications of deep representation learning with attention mechanisms. In particular, Chapter 3 introduces an orientation-aware deep convolutional neural network with simultaneous attention refinement module (SARM) for people detection.

  This chapter is based on [Cao+22c], published at the 2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI).

- Chapter 4 presents an efficient grasp detection network for robot manipulation.

  This chapter is based on [Cao+21a] and [Cao+22a], published at the 2021 IEEE International Conference on Robotics and Automation (ICRA) and IEEE/ASME Transactions on Mechatronics (2022).

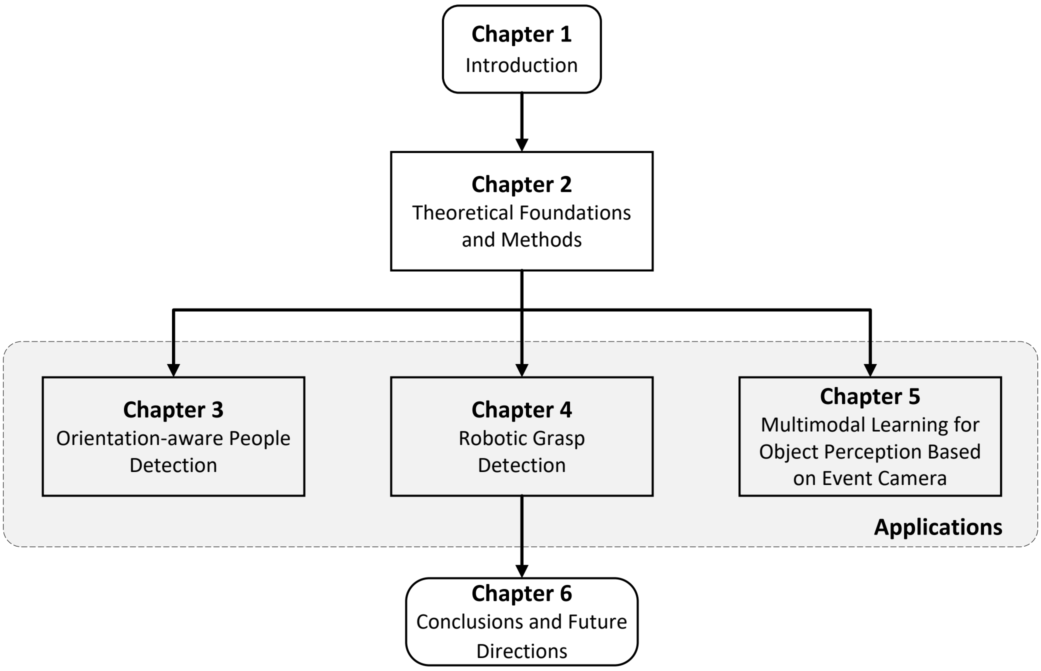

Chapter 1
Introduction
Chapter 2
Theoretical Foundations and Methods
Chapter 3
Orientation-aware People Detection
Chapter 4
Robotic Grasp Detection
Chapter 5
Multimodal Learning for Object Perception Based on Event Camera
Applications
Chapter 6
Conclusions and Future Directions

Figure 1.4: The framework of this thesis.

- Chapter 5 shows multimodal learning for object perception (including vehicle detection and grasp detection).

  This chapter is based on [Che+20a], [Cao+21b], and [Cao+22b], published at the IEEE Signal Processing Magazine (2020), IEEE Sensors Journal (2021), and IEEE Transactions on Instrumentation and Measurement (2022).

- Chapter 6 summarizes the work completed during the thesis and discusses future directions.

2

Theoretical Foundations and Methods

The purpose of this chapter is to present the theoretical foundations of deep representation learning and attention mechanisms relevant to this thesis. In Section 2.1, deep representation learning is reviewed, and attention mechanisms are illustrated in Section 2.2. Finally, evaluation metrics are summarized in Section 2.3.

2.1 Deep Representation Learning

Deep representation learning has achieved great success in perception tasks, including image classification, object detection, semantic segmentation, and grasp detection. Convolutional neural networks (CNNs) are the most representative models, and this thesis focuses on the use of CNNs to perform object detection and grasp detection.

2.1.1 Convolutional Neural Network

Convolutional neural network (CNN) is a popular feature extraction architecture that is composed of three types of layers, including convolutional layers, pooling layers, and fully connected layers. It uses spatially localized convolutional filtering to capture local features of the input image. Basic visual features, such as lines, edges, and corners, are learned in the first few layers, while more abstract features are learned in deeper layers. For an input image matrix I, the correspondence activation map M is computed in the nth neuron of the CNN as follows:

$$M[i,j] = \sigma(\sum_{x=-2k-1}^{2k+1} \sum_{y=-2k-1}^{2k+1} W[x,y]I[i-x,j-y]+b) \tag{2.1}$$

where, the image size is $2k+1$, W is the nth convolutional filter, and σ is the nonlinear activation function. Generally, a max pooling layer follows each convolutional layer, in which the local maximum is used to reduce the dimension of the matrix and prevent over-fitting. Moreover, fully connected layers are usually added to learn the nonlinear combination of extracted features from previous layers. Over the decades, many variants of CNNs such as fully convolutional neural network, and encoder-decoder network have emerged. These networks have different structures from traditional CNNs, such as removing the full connection layer. The performance of CNN has surpassed traditional machine learning methods in many vision tasks, relying on successful training algorithms and large amounts of data.

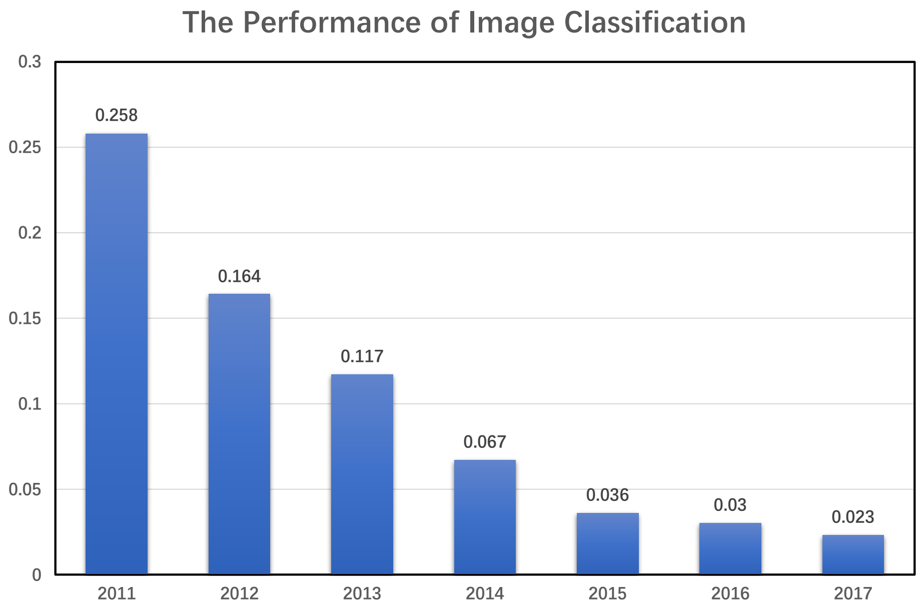


Figure 2.1: The performance of representative CNN frameworks in the ILSVRC competitions.

2.1.2 Object Representation

Good feature representation is a prerequisite for high performance object perception. Previously, researchers usually designed various descriptors as high-level feature representations, such as SIFT [Low99], HOG [DT05b] and Fisher Vector [PSM10]. However, all these representation methods require specialized engineering experience and domain knowledge. Instead, deep representation learning methods can learn powerful feature representations directly from the raw data. Therefore, how to design a better network architecture is the key issue. The following parts review the leading CNN frameworks and methods for improving object representation.

CNN frameworks. The CNN frameworks are used as the backbone of the object perception networks. The most popular frameworks include AlexNet [KSH17], VGGNet [SZ14], GoogLeNet [Sze+15], the Inception family of networks [IS15; Sze+16; Sze+17], ResNet [He+16], DenseNet [Hua+17], and SENet [HSS18]. As shown in Fig. 2.1, CNN-based approaches have achieved better image recognition performance than humans. The architecture of CNN is becoming increasingly complex: AlexNet has only 8 layers, VGGNet has 16, and ResNet and DenseNet have even surpassed the 100-layer mark. As the depth of the network deepens, the representation learning capability becomes more powerful. In addition, the Inception module proposed in Inception series [IS15; Sze+16; Sze+17] demonstrates that the width of the network can also be increased to improve representation learning. However, the researchers found that the performance of the model decreased rather than improved as the network continued to deepen. In order to solve the degradation problem of network deepening, ResNet proposes skipping connections to make it possible to learn hundreds of layers of network models. In [Hua+17], the authors further introduce a dense connection to build the DenseNets, leading to better performance. Moreover, the Squeeze and Excitation (SE) block was developed based on the channel attention mechanism in SENets. This work achieves excellent performance with minimal additional computational cost. More attention mechanisms are illustrated in Section 2.2.

Methods for improving object representation. The features from the last layer of the CNN backbone are usually used as the final representation. High-level features have a large receptive field and strong semantics, but the resolution is low and structural detail information is lost. In comparison, low-level features have high resolution and rich details, but small receptive fields and weak semantics. High-resolution features with shallow semantics are beneficial for detecting small objects, and low-resolution features with deep semantics are beneficial for detecting large objects. Nowadays, many methods have been proposed to improve the representation of objects by exploring multi-scale features. They can be classified into three categories as follows:

Predicting based on the combined multi-scale features. HyperNet [Kon+16] and ION [Bel+16] combine multi-scale features from hierarchical layers as the final representation for prediction. ION extracts ROI features using ROI pooling from hierarchical layers. Then, the predictions are performed on the concatenated features. In Hypernet, deep, medium, and shallow features are integrated to predict objectness and classification.

Predicting based on the hierarchical CNN layers. To capture objects of different sizes, recent methods perform prediction at different CNN layers. In SSD [Liu+16], the authors deploy default boxes of different scales to multiple CNN layers and then predict objects of a certain scale at each layer. Based on the SSD, RFBNet [LHW18] further proposes the Receptive Field Block (RFB) to improve the discriminability and robustness of features. RFB consists of multiple branches with different kernels and convolution layers. RFBNet can effectively improve performance by combining RFB and SSD architecture.

Combination of the above two methods. Features from different CNN layers are complementary. Feature Pyramid Network (FPN) [Lin+17], Path Aggregation Network (PAN) [Liu+18], DSSD [Fu+17] and RetinaNet [Lin+20] were developed to exploit multi-scale features to improve performance. As shown in Fig. 1.3, the architecture of FPN is a top-down network with lateral connections. High-level semantic features and low-level semantic features are combined by this top-down network with lateral processing. The fused features are used for prediction at each layer. Based on the FPN, the authors of [Liu+18] proposed PANet by adding an additional bottom-up path. FPN-like structures have been shown to be effective as generic feature extractors in a variety of applications, including object detection [Lin+20], instance segmentation [Liu+18], and grasp detection [Cao+21a; Cao+22a].

2.2 Attention Mechanisms

In this part, the general form of attention mechanisms and several representative works are illustrated.

2.2.1 General Form

In [Guo+22], the formulation of the attention mechanism can be expressed as follows:

$$Attention = f(g(x), x) \tag{2.2}$$

where x represents the input vector, $g(x)$ denotes the attention function corresponding to the process of attending to the discriminative regions, and $f(g(x), x)$ means the computation of using the attention $g(x)$ on the input x.

With this definition, almost all existing attention mechanisms can be represented by this formulation. For squeeze-and-excitation (SE) attention, $f(g(x), x)$ can be written as follows:

$$\begin{aligned} g(x) &= Sigmoid(MLP(GAP(x))) \\ f(g(x),x) &= g(x)x \end{aligned} \tag{2.3}$$

In the following, various attention mechanisms will be illustrated.

2.2.2 Channel Attention

The feature maps extracted by deep neural networks with different channels usually represent different objects [Che+17]. Channel attention was first proposed in SENet [HSS18] for adaptively calibrating each channel. This process can be considered as object selection: *what to pay attention to*.

SENet. In SENet [HSS18], a squeeze-and-excitation (SE) block is developed to capture the channel-wise relationships. The SE block consists of a squeeze module and an excitation module. Specifically, global average pooling (GAP) is used as a squeeze module to collect global information. The excitation module, composed of fully connected layers (W_1, W_2) and non-linear layers (ReLU (δ) and sigmoid (σ)), is used to capture channel-wise relationships (attention vector). Then, the input features (x) are scaled by multiplying the attention vector. The SE block can be formulated as follows:

$$\begin{aligned} SE(x) &= \sigma(W_2\delta(W_1 GAP(x))) \\ Y &= SE(x)x \end{aligned} \tag{2.4}$$

ECANet. Since fully-connected layers are used in SE blocks, the complexity of SENet is high. To reduce the complexity of SENet, ECANet [Wan+20] introduced a lightweight excitation module. The key module of ECANet is the efficient channel attention (ECA) block. The ECA block can be formulated as follows:

$$\begin{aligned} ECA(x) &= \sigma(Conv1D(GAP(x))) \\ Y &= ECA(x)x \end{aligned} \tag{2.5}$$

where Conv1D represents 1D convolution with a kernel size of k. The parameter k is adaptively determined by the channel dimension C. The computation can be expressed as follows:

$$k = \psi(C) = \left| \frac{log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{2.6}$$

where γ and b denote hyperparameters. $||_{odd}$ represents the odd function.

2.2.3 Spatial Attention

Unlike channel attention, spatial attention can be seen as an adaptive process of spatial region selection: *where to pay attention*.

Attention gate. The attention gate is proposed in Attention Unet [Okt+18] for focusing on object regions while suppressing irrelevant regions. Specifically, additive attention is used to obtain the gating coefficient. The input feature map x and the gating signal g are first mapped to $\mathbb{R}^{F\times H\times W}$ dimensional space, and then the mapped features are fused to generate the spatial attention weight map $S^{1\times H\times W}$. The whole computation can be written as follows:

$$\begin{aligned} AG(x) &= \sigma(\varphi(\delta(\phi_x(x)+\phi_g(g)))) \\ Y &= AG(x)x \end{aligned} \tag{2.7}$$

where φ, ϕ_x and ϕ_g are transformation functions, which are usually implemented by using convolutions.

2.2.4 Channel & Spatial Attention

Channel & spatial attention can adaptively select important objects and regions by combing the strengths of channel attention and spatial attention, thus determining *what to pay attention to and where to pay attention*.

CBAM. To enhance the information at channel-level and spatial-level, the convolutional block attention module (CBAM) is proposed in [Woo+18]. CBAM is composed of channel attention and spatial attention in series. The channel attention can be expressed as follows:

$$\begin{aligned} F^c_{avg} &= GAP^s(x) \\ F^c_{max} &= GMP^s(x) \\ A_c &= \sigma(W_2\delta(W_1F^c_{avg}) + W_2\delta(W_1F^c_{max})) \\ Y_c &= A_c x \end{aligned} \tag{2.8}$$

where GAP^s and GMP^s are global average pooling and global max pooling in the spatial dimension, respectively. The structure of the channel attention sub-module is similar to that of the SE block. The spatial attention sub-module captures spatial relationships. Both channel attention and spatial attention are complementary. The spatial attention is computed as follows:

$$\begin{aligned} F^s_{avg} &= GAP^c(x) \\ F^s_{max} &= GMP^c(x) \\ A_s &= \sigma(Conv([F^s_{avg}; F^s_{max}])) \\ Y_s &= A_s x \end{aligned} \tag{2.9}$$

where GAP^c and GMP^c are global average pooling and global max pooling in the channel dimension, respectively. [] represents the concatenation operation over channels. The overall process of CBAM can be summarized as follows:

$$\begin{aligned} x_c &= Y_c(x) \\ Y &= Y_s(x_c) \end{aligned} \tag{2.10}$$

BAM. Bottleneck attention module (BAM) [Par+18] is emerging at the same time as CBAM. Unlike CBAM, the dilated convolution is used to enlarge the receptive field of the spatial attention submodule of BAM. Moreover, the channel attention submodule and the spatial attention submodule of BAM are formed in a parallel manner. The process of BAM can be written as follows:

$$\begin{aligned} Y_c &= BN(W_2(W_1 GAP(x) + b_1) + b_2) \\ Y_s &= BN(Conv_2^{1\times1}(DC_2^{3\times3}(DC_1^{3\times3}(Conv_1^{1\times1}(x))))) \\ A &= \sigma(Expand(Y_c) + Expand(Y_s)) \\ Y &= Ax + x \end{aligned} \tag{2.11}$$

where W_i and b_i are weights and bias of fully connected layers, respectively. BN indicates batch normalizer and $DC_i^{3\times3}$ represents a dilated convolution with a kernel size of 3 × 3.

2.3 Evaluation Metrics

In this thesis, the average precision (AP) and the grasp rectangle metric are used to evaluate the performance of object detection and grasp detection, respectively.

2.3.1 Average Precision (AP)

The common metric, AP, is used to evaluate the performance of the different object detectors. The value of AP denotes the area under the Precision-Recall curve. Recall, Precision and IOU (Intersection over Union) are expressed as the following:

$$\begin{aligned} Recall &= \frac{tp}{tp + fn}, \\ Precision &= \frac{tp}{tp + fp} \\ IOU &= \frac{detections \cap groundtruth}{detections \cup groundtruth} \\ &= \frac{tp}{tp + fp + fn} \end{aligned} \tag{2.12}$$

where tp represents the true positive samples, meaning the correctly predicted vehicles. Similarly, fp and fn denote the false positive samples and false negative samples, respectively. After Recall, Precision, and IOU are calculated, the area under the Precision-Recall curve (AP) can be used to summarize the performance of the detector. Different from traditional AP, MS COCO [Lin+14] introduces AP_{coco} by averaging over all object classes and multiple IOU values from 0.5 to 0.95.

In particular, AP_{coco} and AP at $IOU = 0.5$ [Eve+10] are used as evaluation metrics for people detection and vehicle detection, respectively.

2.3.2 Grasp Rectangle

Similar to previous works [CXV18; KK17b; Cao+21a], the metric used to evaluate grasp model is the grasp rectangle. Specifically, a predicted grasp is regarded as a correct grasp when it meets the following two conditions:

- **Angle difference:** the difference of orientation angle between the predicted grasp and corresponding grasp label is less than 30° .
- **Jaccard index:** the Jaccard index of the predicted grasp and corresponding grasp label is greater than 25%, which can be formulated as Eq. 2.13.

$$J(g_p, g_l) = \frac{|g_p \cap g_l|}{|g_p \cup g_l|} \tag{2.13}$$

where g_p and g_l denote the predicted grasp rectangle and the area of the corresponding grasp label, respectively. $g_p \cap g_t$ represents the intersection of the predicted grasp and the corresponding grasp label. The union of predicted grasp and the corresponding grasp label is represented as $g_p \cup g_t$.

3

Orientation-aware People Detection

This chapter is about the application of spatial and channel attention in people detection. The key idea is that a simultaneous attention refinement module (SARM) is introduced for people detection in arbitrary directions. Based on SARM, the proposed method can suppress the noise feature and highlight the object feature.

The contents of this chapter are based mainly on the paper "Orientation-aware People Detection and Counting Method Based on Overhead Fisheye Camera," that is published at the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2022 [Cao+22c].

3.1 Background

People detection and counting are becoming increasingly critical tasks for intelligent buildings, which are currently based on wifi signal monitoring, door tripwires, and video cameras. Systems combining video cameras and computer vision algorithms are particularly effective for people detection and counting [EG09; NLO16]. Compared with standard cameras, fisheye cameras have a larger field of view (FOV) of 360°, making them better suited for people detection and counting tasks in the overhead view. The object detector based on deep learning [Che+19] and IoT technology [Lou+20] can provide the basic support for the people detection and counting system.

The overhead view object detection and tracking have been well applied in the IoT scenario[Ahm+20]. In [Ahm+20], the authors explored the application of deep learning models, FasterRCNN [Ren+15] and Mask-RCNN [He+20], in the overhead view multiple object detection. For a standard image object detection task, many deep learning-based methods, such as YOLO [RF16], SSD [Liu+16], and FasterRCNN [Ren+15], have achieved excellent performance. However, these algorithms do not perform well in fisheye images with radial geometry and barrel distortion. To address this issue, several works, such as [THM19; Li+19b; Dua+20], have been proposed to perform fisheye image people detection. In the case of a non-standing pose, the method in [Li+19b] has better performance than [THM19], but it runs slowly. Recently, the authors of [Dua+20] collected a challenging fisheye image dataset and proposed a people detection algorithm with state-of-the-art performance under normal lighting conditions. However, all of the methods mentioned above do not work well in low-light scenarios[THM19; Li+19b; Dua+20].

In this work, I introduce an orientation-aware people detection and counting method based on an overhead fisheye camera. Specifically, an end-to-end deep convolutional neural network is used to detect and count people in fisheye images captured by an overhead fisheye camera. Since the existing fisheye image detection methods do not work well in low-light scenarios, a simultaneous attention refinement module (SARM) is developed to improve the generalization ability of the neural network model under low-light conditions. SARM is composed of two subnetworks: spatial attention and channel attention. To effectively improve the context-focusing ability of the model, I simultaneously apply spatial and channel attention mechanisms to the feature map in a one-step manner. Based on the attention mechanism, the performance of the people detection and counting model in low-light conditions is significantly improved. After the people detection results are available, a datastream server based on RTSP is constructed to send data to the client devices for further analysis. Concretely, my main contribution can be summarized as follows:

- For intelligent infrastructure, an orientation-aware people detection and counting method based on an overhead fisheye camera is presented.

- To improve the performance of the people detection and counting model, an SARM is embedded in the backbone of the network to suppress the noise feature and highlight the object feature.

- The results of the experiments on the three public fisheye image datasets demonstrate that the proposed method achieves better performance, with particularly enhanced generalization under low-light conditions.

3.2 Related Work

In the early object detection methods, algorithms based on handcrafted features were widely used. Histogram of oriented gradients (HOG) [DT05a] and aggregate channel features (ACF) [Dol+14] are the most popular approaches for the standard camera people detection task. Many works combined handcrafted features with classification methods to achieve better performance for people detection. In recent years, deep learning-based methods have achieved great progress [Che+20a; Cao+21b]. Two stage algorithms, such as [Ren+15; He+20], achieve high detection accuracy, but have a slow running speed due to the high computational cost. On the contrary, one-stage algorithms [RF16; Liu+16] achieve high real-time performance, but the detection accuracy decreases. Recently, the methods [Dua+19; Tia+20] based on anchor-free mechanism have been developed to balance detection accuracy and running speed. However, all the above-mentioned methods were developed for standard camera images. The authors of [Ahm+20; Ahm+18] use an overhead view standard camera to do object detection and tracking tasks in IoT applications.

Presently, few works are developed for people detection based on the overhead fisheye camera. [THM19] tries to do fisheye image people detection by training a convolutional neural network on a rotated version of the COCO dataset [Lin+14]. Another work proposed in [Li+19b] achieves high detection accuracy on fisheye images. The disadvantage of [Li+19b] is that its running speed is slow due to high computational complexity. In [Dua+20], a people detection algorithm with state-of-the-art performance is proposed. The authors collect a challenging fisheye image dataset and develop a periodic loss function for rotation bounding box regression. However, the method does not perform well in low-light situations.

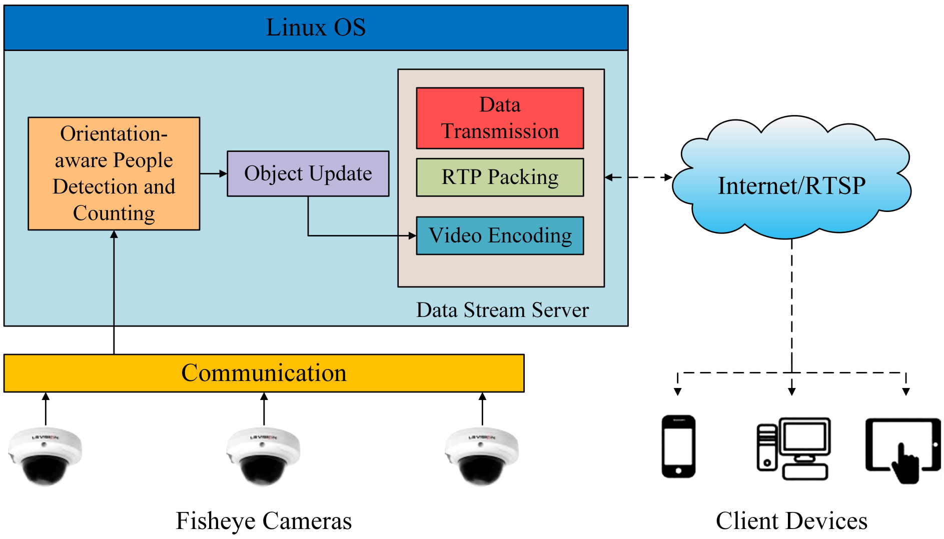


Figure 3.1: Overview of the proposed IoT system architecture. The IoT system is composed of fisheye cameras, an orientation-aware people detection and counting algorithm, a datastream server, and client devices. Adapted from Fig. 1 in [Cao+22c] ©IEEE.

3.3 System Architecture

The proposed IOT system architecture consists mainly of four parts, including fisheye cameras, an orientation-aware people detection and counting algorithm, a datastream server based on RTSP, and client devices, as illustrated in Fig. 3.1. The fisheye cameras are mounted in the overhead position of the monitored indoor spaces. In comparison with standard cameras, fisheye cameras can provide a 360° field of view (FOV) and effectively reduce occlusion between people. By collecting the fisheye images of the top view, it can obtain a variety of human body poses and orientations, such as standing, walking, diagonal, and corner positions. The obtained fisheye image data is fed into an end-to-end deep convolutional neural network to perform people detection and counting. The constantly updated people detection and counting results are encoded and packaged by the datastream server. The packaged data is then sent to the network flow. The entire sender is implemented in the Linux system on the edge device. Client devices, such as smartphones, PCs, and tablets, can obtain data directly from the network flow for decoding and further analysis.

3.4 Orientation-aware People Detection

In this section, I introduce an orientation-aware people detection and counting algorithm. The overall structure of the proposed people detection model is presented in Fig. 3.2. The proposed algorithm's architecture is based on YOLO [RF16; RF18; Dua+20], which achieves excellent performance on standard camera images. I developed the SARM resblock as the backbone to extract rich context features to predict the bounding box with rotation angle, which enables my algorithm to detect arbitrarily-oriented people and get better results on fisheye images. Since I only detect people, my model needs a regression task of rotating bounding boxes. After obtaining the results, an orientation-aware nonmaximum suppression (NMS) algorithm is used to remove redundant detections and obtain the final number of people counting.

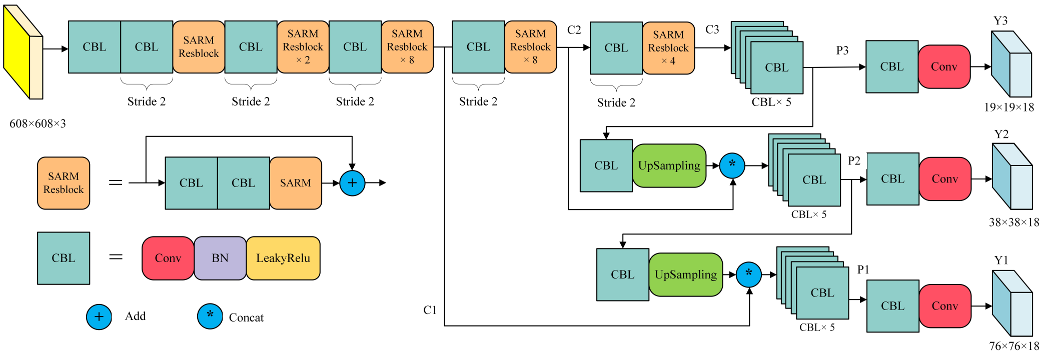


Figure 3.2: The overall structure of the orientation-aware people detection algorithm. Adapted from Fig. 2 in [Cao+22c] ©IEEE.

3.4.1 Basic Networks

The orientation-aware people detection network is composed of a backbone, multilevel prediction based on feature fusion, and detection outputs. The network can be represented as follows:

$$
\begin{aligned}
C_k &= F_{\text{backbone}}\,(I) \\
P_k &= F_{\text{fusion}}\,(C_k) \\
Y_k &= F_{\text{output}}\,(P_k)\,, k=1,2,3
\end{aligned}
\tag{1}
$$

where I is the input fisheye image, and $\{C_k\}_{k=1}^{3}$ is the extracted multilevel features. $\{P_k\}_{k=1}^{3}$ and $\{Y_k\}_{k=1}^{3}$ denote the fused multiresolution features and prediction outputs respectively. F_{backbone} , F_{fusion} and F_{output} denote the backbone function, multilevel feature fusion function, and detection output function, respectively.

Backbone. The resolution of 608 × 608 fisheye images is fed into the backbone network to extract deep features. CBL units and SARM resblocks constitute the backbone network. Specifically, the CBL unit consists of a convolutional filter, batch normalization, and leaky relu activation function. SARM resblock is composed of a CBL unit with a kernel size of 1 × 1, a CBL unit with a kernel size of 3 × 3, and an SARM. Attention mechanisms are inspired by the human visual system, which has been extensively studied in computer vision[HSS18; Woo+18; Wan+18]. In this work, I develop an SARM to improve the discrimination ability of the network. Similar to [Woo+18], SARM consists of spatial attention subnetwork and channel attention subnetwork, as shown in Fig. 3.3. However, SARM uses fewer convolution filters to reduce the number of parameters and simultaneously applies spatial and channel attention enhancement on the feature map in a one-step manner. For the spatial attention subnetwork, average pooling and maximum pooling are used to get representative features and concatenate them into a convolutional filter with a kernel size of 5 × 5. In the channel attention subnetwork, average pooling and a convolution filter with a kernel size of 1×1 are used to generate channel feature distributions. The outputs of the spatial and channel attention subnetworks are fused by adding them together. After the sigmoid operation is performed on the fused feature, it is multiplied by the original input to obtain the novel feature weights. The combination of SARM and residual networks can extract more valuable features to improve the detection performance of the network.

Multilevel prediction based on feature fusion. Since the high-resolution feature map of shallow semantic context information is conducive to the detection of small objects, the low-resolution feature map of deep semantic context information is conducive to the detection

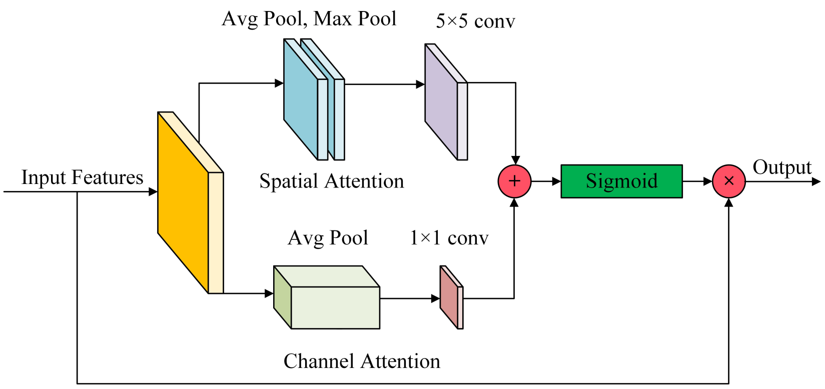


Figure 3.3: SARM work flow diagram. The top branch is the spatial attention subnetwork, and the bottom branch is the channel attention subnetwork. Adapted from Fig. 3 in [Cao+22c] ©IEEE.

of large objects, I use multilevel prediction based on feature fusion to detect objects of different sizes. Specifically, $\{P_k\}_{k=1}^3$ are generated by the feature maps $\{C_k\}_{k=1}^3$ with top-down pathway and lateral connections [Lin+17]. More robust object semantic information can be obtained by combining the shallow and deep features. In this work, P_1, P_2, P_3 are responsible for detecting people of small, medium, and large sizes, respectively.

Detection outputs. The network outputs four coordinates, a rotation angle, and confidence for each rotation bounding box, $t_x, t_y, t_w, t_h, t_\theta, t_{\text{conf}}$. For grid cell $\left(c_x, c_y\right)$ and anchor box with width w_a and height h_a. A rotation bounding box prediction can be formulated as follows:

$$\begin{aligned}
b_x &= \sigma\left(t_x\right) + c_x, \\
b_y &= \sigma\left(t_y\right) + c_y, \\
b_w &= w_a e^{t_w}, \\
b_h &= w_h e^{t_h}, \\
b_\theta &= \sigma\left(t_\theta\right) * r - \frac{r}{2}, \\
b_{\text{conf}} &= \sigma\left(t_{\text{conf}}\right)
\end{aligned} \tag{2}$$

where $\sigma()$ and e denote the sigmoid activation function and exponential function, respectively. r is the angle range of rotation bounding box, which is set to 2π in this work.

3.4.2 Loss Function

For given fisheye image datasets that contain people objects $T = \{T_1 \ldots T_n\}$, fisheye images $M = \{M_1 \ldots M_n\}$, and corresponding ground truth $G = \{G_1 \ldots G_n\}$, I use a neural network model to match the function $F : M \longmapsto Y$. The prediction Y is obtained by applying a deep learning-based training method to learn function F. The loss function is demployed to optimize the minimum error between the network model and the ideal function F. In my model, the loss function L is defined as follows:

$$L = l_{\text{loc}} + l_{\text{angle}} + l_{\text{conf}} \tag{3}$$

Algorithm 1 Orientation-aware nonmaximum suppression (ONMS)

```
Require:
    IOU threshold ($IOU_{th}$), detection outputs $B = \{b_1, b_2, \dots, b_n\}$
Ensure:
    Detection results $S$
S ← {}
B ← argsort $B$; m ← argmax $B$
M ← $b_m$; S ←$S \cup M$; B ← B - M
for $b_i$ in $B$ do
    compute $RotationIOU(b_i, S)$
    if $RotationIOU(b_i, S) \geq (IOU_{th})$ then
        continue
    end if
    S ← S ∪ $b_i$
end for
return S
```

where, the total loss L is composed of localization loss l_{loc} . angle regression loss l_{angle} and object confidence loss l_{conf} . Specifically, $l_{loc}, l_{\text{angle}}, l_{\text{conf}}$ are formulated as follows:

$$
\begin{aligned}
l_{\text{loc}} &= \sum_{t \in Y^{\text{pos}}} \left[BCE\left(\sigma\left(t_x\right), g_x\right) + BCE\left(\sigma\left(t_y\right), g_y\right) \right] \\
&\quad + \sum_{t \in Y^{pos}} \left[\left(\sigma\left(t_w\right) - g_w\right)^2 + \left(\sigma\left(t_h\right) - g_h\right)^2 \right] \\
l_{\text{angle}} &= \sum_{t \in Y^{\text{pos}}} f\left(\bmod\left(b_\theta - g_\theta - \frac{\pi}{2}, \pi \right) - \frac{\pi}{2} \right) \\
l_{\text{conf}} &= \sum_{t \in Y^{\text{pos}}} BCE\left(\sigma\left(t_{\text{conf}}\right), 1\right)
\end{aligned}
\tag{4}
$$

where Y^{pos} denotes the positive samples from predictions. g_x, g_y, g_w, g_h are the corresponding transformed ground truth. b_θ is computed in Eq.2, and g_θ is the corresponding angle ground truth. BCE represents binary cross-entropy function, f is any symmetric function, such as L_1, L_2 norm, and mod denotes the modulo operation. Using periodic loss function l_{angle} [Dua+20], ambiguity issues can be solved in the rotation angle regression process.

3.4.3 Orientation-aware Nonmaximum Suppression

For the people counting and inference processes, an orientation-aware nonmaximum suppression (ONMS) method is used to remove duplicate detection outputs and get more accurate people counting results. First, the detection output is removed when the confidence score is lower than the confidence threshold, which can improve the quality of the detection outputs. The remaining detection proposals are then computed by orientation-aware nonmaximum suppression (ONMS) to remove the overlap proposals. The detailed steps are presented in Algorithm I. In ONMS, when the value of the rotation IOU of the detection proposal is greater than the IOU threshold (IOU_{th}), it is removed, thus ensuring that each person has only one corresponding detection bounding box. Finally, the number of people counting is equal to the total number of bounding boxes in each image.

Table 3.1: Summary of the public fisheye image datasets. Adapted from Table I in [Cao+22c] ©IEEE.

Datasets	Videos	Objects	Images	Resolutions	FPS
MW-R	19	6	8752	1065-1488	15
HABBOF	4	5	5837	2048	30
CEPDOF	8	13	25504	1080-2048	1-10

3.5 Experiments

To validate the effectiveness of the algorithm, I performed experiments on three public fisheye image datasets. Furthermore, I discuss the impact of the design of different attention mechanisms on network performance as well as the shortcomings of the network.

3.5.1 Dataset

For orientation-aware fisheye image detection, the available dataset is insufficient. Recently, Mirror Worlds (MW) datset[1], Human-Aligned Bounding Boxes from Overhead Fisheye Cameras (HABBOF) dataset [Li+19b] and Challenging Events for Person Detection from Overhead Fisheye Images (CEPDOF) dataset [Dua+20] 14) were collected, whose details are summarized in Table.3.1.

MW-R Dataset. Mirror Worlds (MW) is a multi-object, multi-camera tracking dataset that was built for an infrastructure research project funded by the National Science Foundation (NSF) at Virginia Tech. Since the dataset is not labeled with rotated bounding boxes, the authors of [Dua+20] manually annotated a subset of the MW dataset with rotated bounding boxes named MW-R.

HABBOF Dataset. The HABBOF dataset contains four videos with a resolution of 2048x2048. The videos were recorded using a fisheye camera in challenging scenes such as occlusion and brightness changes. All fisheye images in HABBOF are labeled with a rotated bounding box.

CEPDOF Dataset. CEPDOF is a large fisheye image dataset created in [Dua+20]. More challenging scenes, such as varied people poses, crowded rooms, and low light, were recorded in the CEPDOF dataset, which will further facilitate the development of robust algorithms.

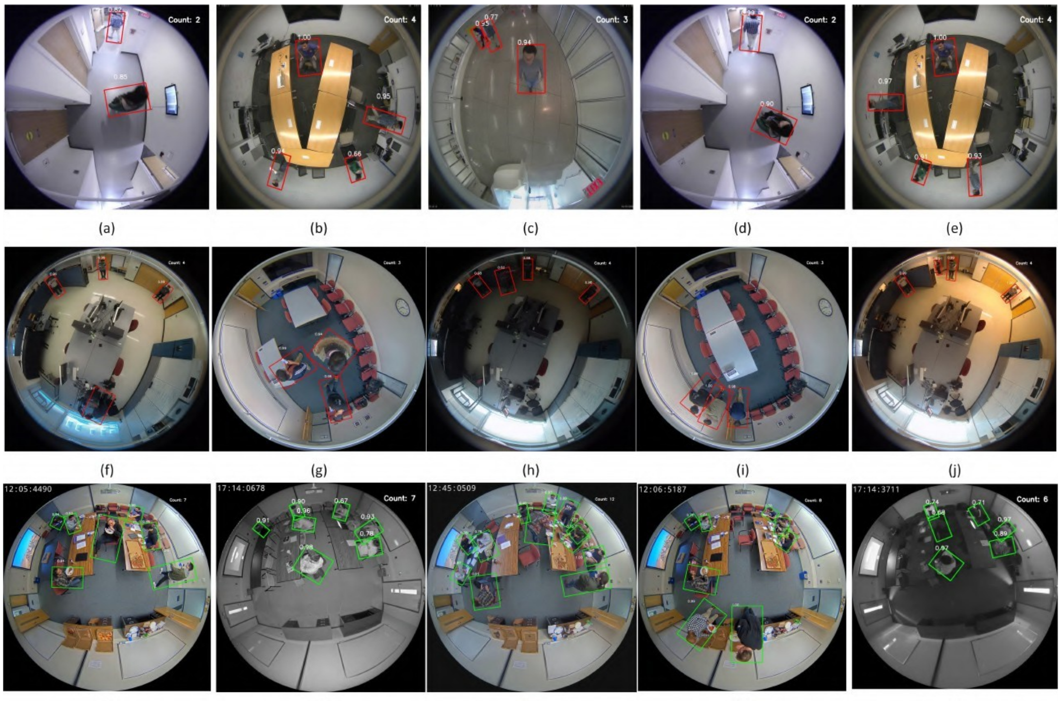
3.5.2 Implementation Details

The people detection and counting algorithm is implemented based on Python 3.7 and Pytorch 1.3 with CudNN 7.5 and Cuda-10.0 packages, and the datastream server is completed based on the libraries of live555[2] and x264[3]. The people detection model is pretrained on an Nvidia GTX 2080 Ti GPU with 11GB of memory. During training, I use data augmentation methods such as random rotation, scaling, flipping, and color enhancement to improve the generalization ability of the model. Furthermore, stochastic gradient descent (SGD) with parameters of a learning rate of 0.0001, momentum of 0.9, and weight decay of 0.0005 is used

[1] http://www2.icat.vt.edu/mirrorworlds/challenge/index.html

[2] http://www.live555.com/

[3] https://www.videolan.org/developers/x264.html


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

Figure 3.4: Detection results on MW-R (a–e), HABBOF (f–j), and CEPDOF (k–o). Adapted from Fig. 4 in [Cao+22c] ©IEEE.

Table 3.2: Comparison results of detection accuracy (%) of different methods on MW-R, HABBOF, and CEPDOF dataset. Adapted from Table II in [Cao+22c] ©IEEE.

Methods	MW-R	HABBOF	CEPDOF	FPS
Tamura[THM19]	78.2	87.3	61.0	6.8
Li[Li+19b]	88.4	87.7	73.9	0.3
Duan[Dua+20]	96.6	97.3	82.4	7.0
This work	**97.1**	**97.8**	**83.6**	6.5

for network optimization. Due to the limitation of computing resources, the batch size is set to 4 in this work with the fisheye images with a resolution of 608 × 608 as the input.

3.5.3 Comparisons under Three Fisheye Image Datasets

Following the previous works [THM19; Li+19b; Dua+20], cross-validation is used to evaluate the performance of the model on the fisheye image datasets. The experiment results on three public fisheye image datasets indicate that my method outperforms other algorithms presented in Table. 3.2. The neural network model based on SARM that takes the image resolution of 608 × 608 as input achieves better performance on MW-R, HABBOF and CEPDOF with detection accuracy of 97.1%, 97.8% and 83.6% respectively. In Fig. 3.4, the selected detection results from three datasets, MW-R (a–e), HABBOF (f–j), and CEPDOF (k–o), are presented.

Table 3.3: The detection performance of different methods on individual sub-datasets in CEPDOF. Adapted from Table III in [Cao+22c] ©IEEE.

Scenarios	Sequence	Method	AP(%)	Precision	Recall
Normal light	Lunch Meeting 1	Duan	96.7	0.95	0.92
		This work	96.8	0.96	0.95
	Lunch Meeting 2	Duan	95.7	0.93	0.87
		This work	92.2	0.95	0.91
	Lunch Meeting 3	Duan	91.3	0.91	0.79
		This work	91.9	0.92	0.82
	Edge cases	Duan	89.2	0.97	0.79
		This work	89.4	0.98	0.82
	High activity	Duan	93.2	0.97	0.88
		This work	93.2	0.97	0.90
Low light	IRill	Duan	88.6	0.93	0.71
		This work	90.0	0.96	0.77
Extremely low light	All-off	Duan	52.8	0.85	0.43
		This work	58.9	0.86	0.50
	IRfilter	Duan	51.4	0.88	0.35
		This work	56.3	0.92	0.42

3.5.4 Performance under Different Illuminations

The current fisheye image detection methods based on deep learning have achieved high detection accuracy under normal lighting conditions, but the accuracy has been significantly reduced in low-light scenarios.[Li+19b; Dua+20]. I compare my model with the state-of-the-art method [Dua+20] under the conditions of various brightness changes. Specifically, the challenging CEPDOF dataset, composed of 8 sequences, is used as a benchmark to explore the impact of illumination. I explored three main light conditions, namely normal light, low light (overhead lights off, with IR illumination) and extremely low light (overhead lights off, no IR illumination) presented in Table. 3.3.

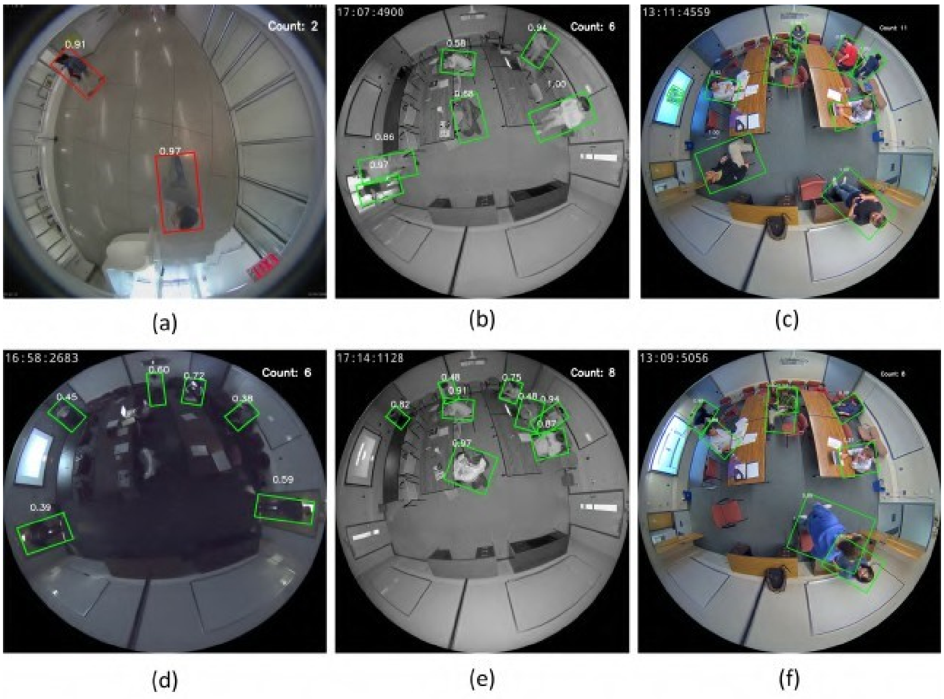
I use the resolution of 608×608 as the input. Using AP (average precision), precision, recall, and F-measure as evaluation metrics, both [Dua+20] and my method achieve excellent performance under normal lighting conditions. However, in the low-light scene, my method achieves the accuracy improvement of 6.1% and 5.9% on All-off and IRfilter respectively, indicating that my method can alleviate the problem that the detection ability of the model decreases under low light.

3.5.5 Ablation Study

In order to improve the discrimination ability of the model, a simultaneous attention refinement module (SARM) is embedded into the backbone network. Based on the attention mechanism, the generalization ability of the network under low-light conditions is significantly improved. In Table. 3.4, I further explored the influence of different attention methods on the performance of fisheye image detection. Experimental results on CEPDOF demonstrate that SE block [HSS18] and CBAM [Woo+18], which perform well in the standard camera tasks, do not work well in the fisheye image dataset, but SARM achieves the best accuracy.

Table 3.4: The impact of different attention methods on detection performance. Adapted from Table IV in [Cao+22c] ©IEEE.

Datasets	+ SE block	+ CBAM	+ SARM
Lunch Meeting1	95.5	93.1	**96.8**
Lunch Meeting2	90.2	88.9	**92.2**
Lunch Meeting3	86.1	83.2	**91.9**
Edge cases	87.2	85.0	**89.4**
High activity	92.6	91.1	**93.2**
All-off	53.0	52.6	**58.9**
IRfilter	49.7	46.4	**56.3**
IRill	86.1	69.5	**90.0**
Average Accuracy (%)	80.1	76.2	**83.6**



(a) (b) (c)

(d) (e) (f)

Figure 3.5: Failed detection cases: only one person was detected due to severe occlusion between two people in (a). The clothes on the chair were wrongly detected as people (b, e) and missed detection under low light conditions (d). Three people on the projector screen were mistakenly detected as real people in (c). Two detection boxes appeared for the same person in (f). Adapted from Fig. 5 in [Cao+22c] ©IEEE.

3.5.6 Failure Cases Analysis

Although my method achieves high accuracy, there are still some problems to be solved, such as occlusion, low light, and false detection. Some typical cases are presented in Fig. 3.5. However, all of these problems can be alleviated by increasing the diversity of datasets.

3.6 Summary

In this work, I propose an orientation-aware people detection and counting system based on IoT technology and an overhead fisheye camera. First, an orientation-aware deep convolutional neural network is developed for arbitrarily oriented people detection and counting. In order to effectively improve the context-focusing ability of the model, a simultaneous attention refinement module (SARM) is embedded in the backbone of the network to extract more discriminative features. Based on attention mechanisms, the performance of the people detection and counting model under low-light conditions is significantly improved. The model takes the image resolution of 608×608 as input to achieve detection accuracy of 97.1%, 97.8% and 83.6% on MW-R, HABBOF, and CEPDOF, respectively. Finally, after obtaining the detection results, a datastream server is constructed to send data to the different client devices for further analysis. By combining deep learning algorithms and IoT technology, the capabilities of intelligent infrastructures can be improved effectively.

4

Robotic Grasp Detection

This chapter is about the application of spatial and channel attention in grasp detection. The key idea is that a multi-dimensional attention fusion network (MDAFN) is introduced to fuse valuable semantic information.

The contents of this chapter are based mainly on the papers "Residual Squeeze-and-Excitation Network with Multi-scale Spatial Pyramid Module for Fast Robotic Grasping Detection," that is published at the IEEE International Conference on Robotics and Automation (ICRA), 2021 [Cao+21a] and "Efficient Grasp Detection Network with Gaussian-based Grasp Representation for Robotic Manipulation" that is published at the IEEE/ASME Transactions on Mechatronics, 2022 [Cao+22a].

4.1 Background

Intelligent robots are crucial in human-robot cooperation, robot assembly, and robot welding [DWL21]. The robots need an effective automated manipulation system to complete the task of picking and placing [Liu+20; WG21]. However, grasping is a straightforward action for humans but challenging for robots because it involves perception, planning, and execution. Grasp detection is a crucial procedure for robots to perform grasp and manipulation tasks in the real world. Therefore, it is necessary to develop a robust perception algorithm to improve the performance of the robotic grasp.

Early grasp detection algorithms were mainly based on search algorithms. Unfortunately, these algorithms are inefficient in complex real-world scenarios [PBK14]. Recently, deep learning-based approaches have achieved excellent results in robotic grasp detection [LLS15; Che+20e; Li+20a; KK17b]. A five-dimensional grasp configuration is proposed to represent a grasp rectangle based on two-dimensional space projected into three-dimensional space to guide the robot to grasp [LLS15]. Due to the simplification of the grasp object dimension, the deep convolutional neural network can learn to extract more suitable features for specific tasks than hand-engineered features by taking 2-D images as inputs. According to the literature, training neural networks to predict grasp with the highest probability score from multiple grasp candidates is the best grasping result [RA15; ATH19; KK17b]. Currently, excellent general object detection models have been introduced in the grasp detection task, such as one-stage and two-stage deep learning methods [RF16; Liu+16; Ren+15]. Similarly, the idea of Faster RCNN is to perform robotic grasp detection by taking RGB-D images as inputs [CXV18]. While in [Wu+19] and [PSC18a], achieving excellent grasp detection accuracy is based on single-stage object detection methods, YOLO [RF16], and SSD [Liu+16].

However, these methods are challenging to balance in terms of accuracy and inference speed due to their complex network structures. The authors in [Zho+18; Son+20] improved the performance of grasp detection by employing an oriented anchor box mechanism to match the grasp rectangles. These methods have improved the detection accuracy, but the computational loads are still too large to be suitable for real-time applications.

A new grasp representation was proposed to solve these problems using the method of sampling grasp candidate rectangles and applying convolutional neural networks to regress grasp points directly [MCL20]. This approach simplifies the definition of grasp representation and achieves high real-time performance based on a lightweight architecture. Inspired by [MCL20], the authors of [Wan+19a; Wan20] use the key idea of algorithms in vision segmentation tasks to predict robotic grasp poses from extracted pixel-wise features. Recently, the residual structure was introduced into the generated neural network model [KK17b], achieving better grasp detection accuracy on the common grasp datasets. However, the shortcoming is the failure to highlight the importance of the largest grasp probability at the center point.

This work uses a 2-D Gaussian kernel to encode training samples to emphasize the highest grasp confidence score at the center point position. Based on Gaussian-based grasp representation, I developed a lightweight generative architecture for robotic grasp pose estimation. Referring to the human visual system's receptive field structure, the combination of residual and receptive field blocks (RFBs) in the bottleneck layer can enhance the feature's discriminability and robustness. Furthermore, low-level features and deep features in the decoder are fused to reduce the information loss caused in the sampling process. Specifically, a multi-dimensional attention network composed of pixel and channel attention networks is used to suppress redundant features and highlight significant object features in the fusion process. Experimental results demonstrate that the proposed algorithm achieves excellent performance in balancing accuracy and inference speed. The main contributions are summarized as follows:

- I propose a Gaussian-based grasp representation, reflecting the maximum grasp score at the center point location.
- I developed an efficient generative architecture for robotic grasp detection.
- A receptive field block is embedded in the network's bottleneck to enhance its feature discriminability and robustness. A multi-dimensional attention fusion network has been developed to suppress redundant features and improve object features in the fusion process.
- Experimental results demonstrate that the proposed method performs well on the public Cornell [YMS11], Jacquard [DDC18], and extended OCID [AF21] grasp datasets.

4.2 Related Work

For 2D planar robotic grasping where the grasp is constrained in one direction, the methods can be divided into oriented rectangle-based grasp representation methods and contact point-based grasp representation methods. The comparison of the two grasp representations is presented in Fig. 4.1. I will review the relevant works below.

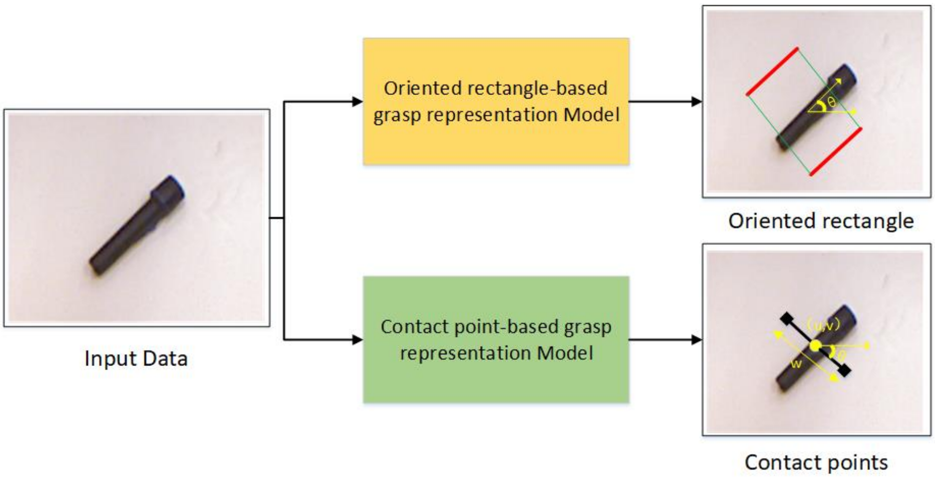

Input Data
Oriented rectangle-based grasp representation Model
Contact point-based grasp representation Model
θ
(u,v)
w
Oriented rectangle
Contact points

Figure 4.1: A comparison between the methods of oriented rectangle-based grasp representation and the methods of contact point-based grasp representation. The top branch is the workflow of the model using the oriented rectangle as grasp representation, and the bottom branch is the workflow of the model using the contact point grasp representation. Adapted from Fig. 1 in [Cao+21a] ©IEEE.

4.2.1 Methods of Oriented Rectangle-based Grasp Representation

The goal of grasping detection is to find the appropriate grasp pose for the robot through the grasping object's visual information to provide reliable perception information for the subsequent planning and control processes and achieve a successful grasp. Grasp is a widely studied topic in the field of robotics, and the approaches used can be summarized as analytic methods and empirical methods. The analytical methods use mathematical and physical models in geometry, motion, and dynamics to carry out the calculations for grasping [DWL21]. Its theoretical foundation is solid, but the deficiency lies in the fact that the model between the robot manipulator and the grasping object in the real 3-dimensional world is complex. It is difficult to realize the model with high precision. In contrast, empirical methods do not strictly rely on real-world modeling methods, and some works utilize data information from known objects to build models to predict the grasping pose of new objects [Ina+19; Gar+19; Zha+19a]. A new grasp representation is proposed in [YMS11], where a simplified five-dimensional oriented rectangle grasp representation is used to replace the seven-dimensional grasp pose consisting of 3D location, 3D orientation, and the opening and closing distance of the plate gripper. Based on the oriented rectangles grasp configuration, the deep learning approaches can be successfully applied to the grasping detection task, which mainly includes classification-based methods, regression-based methods, and detection-based methods [DWL21].

Classification-based methods. A first deep learning-based robotic grasping detection method is presented in [LLS15], the authors achieve excellent results by using a two-step cascaded structure with two deep networks. In [PG16], grasping proposals are estimated by sampling grasping locations and adjacent image patches. The grasp orientation is predicted by dividing the angle into 18 discrete angles. Since the grasp dataset is scant, a large simulation database called Dex-Net 2.0 is built in [Mah+17]. On the basis of Dex-Net 2.0, a Grasp-Quality Covolutional Neural Network (GQ-CNN) is developed to classify the potential grasps. Although the network is trained on synthetic data, the proposed method still works well in the real world. Moreover, a classification-based robotic grasping detection method with a spatial transformer network (STN) is proposed in [PC18]. The results of evaluating the Cornell grasping dataset indicate that their multi-stage STN algorithm performs well. The grasping detection method based on classification is a more direct and reasonable method, many aspects of which are worth further study.

Regression-based methods. Regression-based methods are used to directly predict the grasp parameters of location and orientation by training a model. A first regression-based single shot grasping detection approach is proposed in [RA15], in which the authors use AlexNet to extract features and achieve real-time performance by removing the process of searching for potential grasps. Combing RGB and depth data, a multi-modal fusion method is introduced in [Zha+17]. With the fusion of RGB and depth features, the proposed method directly regresses the grasp parameters and improves the grasp detection accuracy on the Cornell grasping dataset. Similar to [Zha+17], the authors of [KK17b] use ResNet as backbone to integrate RGB and depth information and further improves the performance of grasping detection. In addition, a grasping detection method based on ROI (region of interest) is proposed in [Zha+19a]. In this work, the authors regress grasp pose on ROI features and achieve better performance in an object-overlapping challenge scene. The regression-based method is effective, but its disadvantage is that it is more difficult to learn the mean value of the ground truth.

Detection-based methods. Many detection-based methods refer to some key ideas from object detection, such as the anchor box. Based on the prior knowledge of these anchor boxes, the regression problem of grasping parameters is simplified. In [Guo+17], vision and tactile sensing are fused to build a hybrid architecture for robotic grasping. The authors use an anchor box to do axis alignment, and grasp orientation is predicted by considering grasp angle estimation as a classification problem. The grasp angle estimation method used in [Guo+17] is extended by [CXV18]. By transforming the angel estimation problem into a classification problem, the method of [CXV18] achieves high grasping detection accuracy on the Cornell dataset based on FasterRCNN [Ren+15]. Different from the horizontal anchor box used in object detection, the authors of [Zho+18] specially design an oriented anchor box mechanism for grasping task and improve the performance of model by combing end-to-end fully convolutional neural network. Morever, [DDC18] further extends the method of [Zho+18] and proposes a deep neural network architecture that performs better on the Jacquard dataset.

4.2.2 Methods of Contact Point-based Grasp Representation

The grasping representation based on an oriented rectangle is widely used in robotic grasping detection tasks. In terms of the real plate grasping task, the gripper does not need so much information to perform the grasping action. A new simplified contact point-based grasping representation is introduced in [MCL20], which consists of grasp quality, center point, oriented angle, and grasp width. Based on this grasping representation, GGCNN and GGCNN2 are developed to predict the grasping pose, and their methods achieve excellent performance in both detection accuracy and inference speed. Referring to [MCL20], the grasping detection performance is improved by a fully convolutional neural network with a pixel-wise approach in [Wan+19a]. Both [MCL20] and [Wan+19a] take depth data as input, a generative residual convolutional neural network is proposed in [KK17b] to generate grasps, which take n-channel images as input. Recently, the authors of [Wan20] take some ideas from image segmentation to perform three-finger robotic grasping detection. Similar to [Wan20], an orientation attentive grasp synthesis (ORANGE) framework is developed in [Gka+20], which

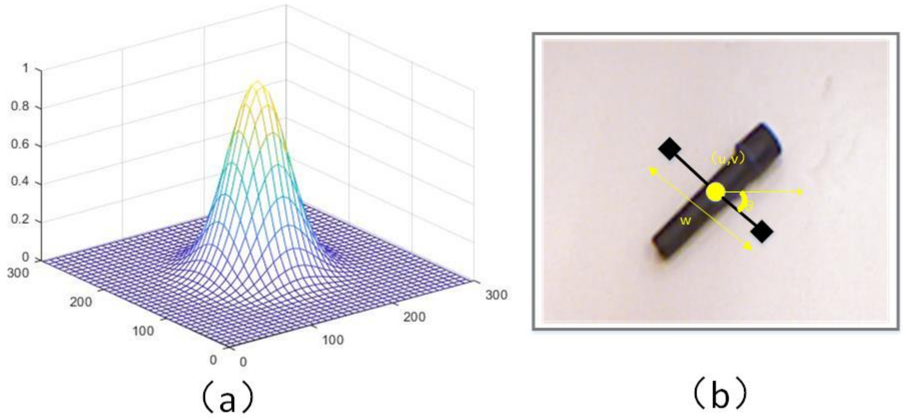


Figure 4.2: Gaussian-based grasp representation: The 2-D Gaussian kernel is applied to the grasp quality map to highlight the maximum grasp quality of its central point position. (a) the schematic diagram of grasp quality weight distribution after 2-D Gaussian function deployment, and (b) the schematic diagram of grasp representation. Adapted from Fig. 1 in [Cao+22a] ©IEEE.

achieves better results on the Jacquard dataset based on the GGCNN and Unet models. In this paper, based on the contact point-based grasp representation, I further develop a lightweight generative architecture for robotic grasping detection that performs well in inference speed and accuracy on two public datasets, Cornell and Jacquard.

4.3 Robotic Grasp System

This part gives an overview of the robotic grasp system settings and illustrates the principles of Gaussian-based grasp representation.

4.3.1 System Setting

A robotic grasp system consists of a robot arm, perception sensors, grasping objects, and workspace. To complete the grasping task successfully, the subsystem of planning and control is involved along with the grasping pose of objects. In grasp detection, I consider limiting the manipulator to the normal direction of the workspace to become a goal for perception in 2D space. Most graspable objects are flat through these settings by placing them reasonably on the workbench. As opposed to building 3D point cloud data, the whole grasp system can reduce the cost of storage and calculation and improve its operational capacity. The grasp pose of flat objects can be treated as a rectangle. Since the size of each plate gripper is fixed, I use a simplified grasp representation to perform grasp pose estimation.

4.3.2 Gaussian-based Grasp Representation

The grasp detection model should take RGB or depth images as inputs to generate grasp candidates for subsequent manipulation tasks. Works from literature built their grasp detection model for grasp pose prediction based on 5-D grasp representation [Guo+17; CXV18; Zho+18].

$$g = \{x, y, \theta, w, h\} \tag{4.1}$$

where the center point is denoted as (x, y). θ is the grasp angle and (w, h) is the weight and height of the grasp rectangle, respectively. The five-dimensional grasp representation is borrowed from conventional object detection, but is not perfectly suited for robotic grasp detection. The simplified grasp representation introduced in [MCL20] for fast robotic grasp detection can be formulated as in Eq. 4.2:

$$g = \{\mathbf{p}, \varphi, w, q\} \tag{4.2}$$

where $\mathbf{p}$ is the position of the center point expressed in Cartesian coordinates as $\mathbf{p} = (x, y, z)$. The φ and w denote the grasp angle and grasp width, respectively. And q is a scale factor for measuring the grasp quality. Furthermore, the new grasp representation in two-dimensional space is represented in Eq. 4.3:

$$\hat{g} = \{\hat{p}, \hat{\varphi}, \hat{w}, \hat{q}\} \tag{4.3}$$

where $\hat{p}$ is the center point in the image coordinates denoted as $\hat{p} = (u, v)$. $\hat{\varphi}$ represents the grasp angle in the camera frame. $\hat{w}$ and $\hat{q}$ denote the grasp width and the grasp quality, respectively. After obtaining the grasp system calibration results, matrix operations transform the grasp pose $\hat{g}$ into world coordinates g, as shown in Eq. 4.4.

$$g = T_{\mathrm{RC}}(T_{\mathrm{CI}}(\hat{g})) \tag{4.4}$$

where T_{RC} is the transformation matrix from camera frames to world frames and T_{CI} is the transformation matrix from two-dimensional image space to camera frames. The grasp map in image space is denoted in Eq. 4.5:

$$\mathbf{G} = \{\Phi, W, Q\} \in \mathbb{R}^{3 \times W \times H} \tag{4.5}$$

where the pixels of grasp maps, Φ, W, Q, are filled with the corresponding values of $\hat{\varphi}, \hat{w}, \hat{q}$. The central location can be found by searching the pixel coordinate with the maximum grasp quality $\hat{g^*} = \max_{\hat{Q}} \hat{G}$. The authors in [MCL20] filled a rectangular area around the center with 1, indicating the highest grasp quality and other pixels as 0. The training model learns the maximum grasp quality of the center. Because all pixels in the rectangular area have the best grasping quality, it leads to a limitation that the importance of the center point is not highlighted, resulting in ambiguity in the model. In this work, I use a 2-D Gaussian kernel to regularize the grasp representation to indicate where the object center might exist, as shown in Fig. 4.2. The novel Gaussian-based grasp representation is represented as g_k. The corresponding Gaussian-based grasp map is defined as the Eq. 4.6:

$$\begin{aligned} &G_K = \{\Phi, W, Q_K\} \in \mathbb{R}^{3 \times W \times H} \\ &\text{where,} \\ &Q_K = K(x, y) = \exp\Big(-\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2}\Big) \\ &\text{where,} \\ &\sigma_x = T_x, \sigma_y = T_y \end{aligned} \tag{4.6}$$

In Eq. 4.6, the generated grasp quality map is decided by the center point location (x_0, y_0), the parameter σ_x and σ_y, and the corresponding scale factor T_x and T_y. In this method, the peak of the Gaussian distribution is the center coordinate of the grasp rectangle. This work discusses the detailed effects of parameter settings in the Section 5.2.4.

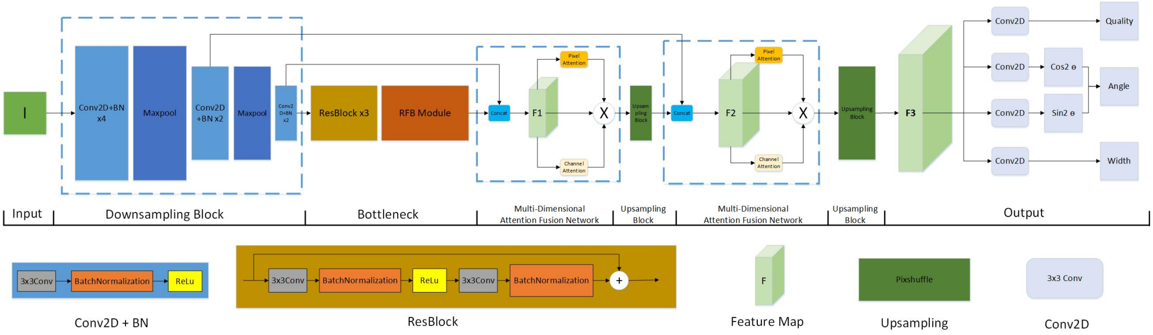


Figure 4.3: The architecture of my generative grasping detection model. I and Conv denote the input data and convolution filter, respectively. The proposed method consists of the down-sampling block, the bottleneck layer, the multi-dimensional attention fusion network (MDAFN), and the up-sampling block. Adapted from Fig. 2 in [Cao+22a] ©IEEE.

4.4 Method

In this section, I introduce a lightweight generative architecture for robotic grasp detection. Fig. 5.12 presents the overall structure of my grasp detection model. The input data is down-sampled into feature maps with smaller sizes, more channels, and richer semantic information. Resnet [He+16] and the multi-scale RFB are combined in the bottleneck to extract more discriminability and robustness features. Furthermore, a MDAFN consisting of pixel-based and channel-based attention subnetworks is used to fuse shallow and deep semantic features. The proposed model suppresses redundant features and enhances the object features during the fusion process based on the attention mechanism. Finally, based on the extracted features, four task-specific sub-networks are added to predict grasp quality, angle (in the form of $sin(2\theta)$ and $cos(2\theta)$), and width, respectively. A detailed illustration of each component of the proposed grasp network is depicted in the following subsections.

4.4.1 Basic Network Architecture

The proposed generative grasp architecture comprises of the down-sampling block, bottleneck layer, multi-dimensional attention fusion network, and up-sampling block, as shown in Fig. 5.12. The down-sampling block consists of a convolutional layer with a kernel size of 3×3 and a maximum pooling layer with a kernel size of 2×2, which can be represented as Eq. 4.7.

$$x_d = f_{\text{maxpool}}(f^n_{\text{conv}}(f^{n-1}_{\text{conv}}(\ldots f^0_{\text{conv}}(I)\ldots))) \tag{4.7}$$

In this work, I use two down-sampling blocks and two convolutional layers in the down-sampling process. The first down-sampling block comprises four convolutional layers (n = 3) and one maximum pooling layer. The second down-sampling layer comprises two convolutional layers (n = 1) and one maximum pooling layer. After the downsampled data passes through two convolutional layers, it is fed into a bottleneck layer consisting of three residual blocks (k = 2) and one receptive field block (RFB) to extract features. Since RFB comprises various scale convolutional filters, it is possible to acquire richer image details. More illustrations of RFB are presented in Section 4.4.2. The output of the bottleneck can be formulated as Eq. 4.8.

$$x_b = f_{\text{RFBM}}(f^k_{\text{res}}(f^{k-1}_{\text{res}}(\ldots f^0_{\text{res}}(f^1_{\text{conv}}(f^0_{\text{conv}}(x_d)))\ldots))) \tag{4.8}$$

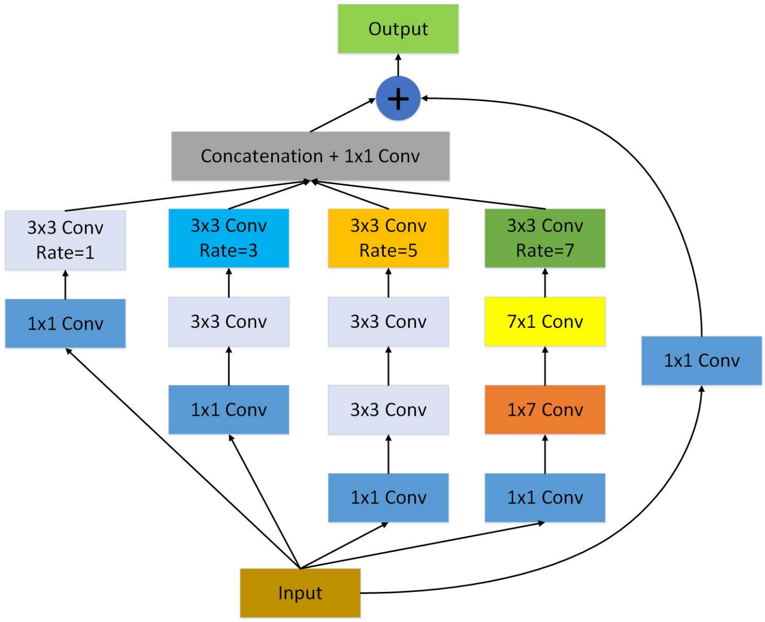


Figure 4.4: Multi-scale receptive field block. The RFB consists of multi-branch convolutional layers with different kernels corresponding to the receptive fields of various sizes. Adapted from Fig. 3 in [Cao+22a] ©IEEE.

The output x_b of the bottleneck is fed into a MDAFN and upsampling block. The MDAFN composed of pixel attention and channel attention subnetworks can suppress the noise features and enhance the valuable features during the fusion of shallow features and deep features. The detailed illustration of the MDAFN is presented in Section 4.4.3. In the upsampling block, the pixshuffle layer [Shi+16] increases feature resolution with the scale factor set to 2. In this work, the number of MDAFN and upsampling blocks is two, and the output is represented as Eq. 4.9.

$$x_u = f^1_{\text{pixshuffle}}(f^1_{\text{MDAFN}}(f^0_{\text{pixshuffle}}(f^0_{\text{MDAFN}}(x_b)))) \tag{4.9}$$

The final layer consists of four convolutional filters with a kernel size of 3×3. The corresponding outputs can be expressed as Eq. 4.10.

$$\begin{aligned} g_q &= \max_q(f^0_{\text{conv}}(x_u)), \\ g_{\cos(2\theta)} &= \max_q(f^1_{\text{conv}}(x_u)), \\ g_{\sin(2\theta)} &= \max_q(f^2_{\text{conv}}(x_u)), \\ g_w &= \max_q(f^3_{\text{conv}}(x_u)). \end{aligned} \tag{4.10}$$

where the center point is located by searching the pixel coordinate with the highest grasp quality g_q. g_w denotes the grasp width, and the grasp angle is calculated by $g_{\text{angle}} = \arctan(\frac{g_{\sin(2\theta)}}{g_{\cos(2\theta)}})/2$.

4.4.2 Multi-scale Receptive Field Block

In neuroscience, researchers have discovered a particular function in the human visible cortex that regulates the size of the visible receptive area [BJ15; Che+20a]. This mechanism can help to emphasize the importance of the area near the center. For robotic grasping tasks, multi-scale receptive fields can enhance the neural network's deep features. I hope to enhance

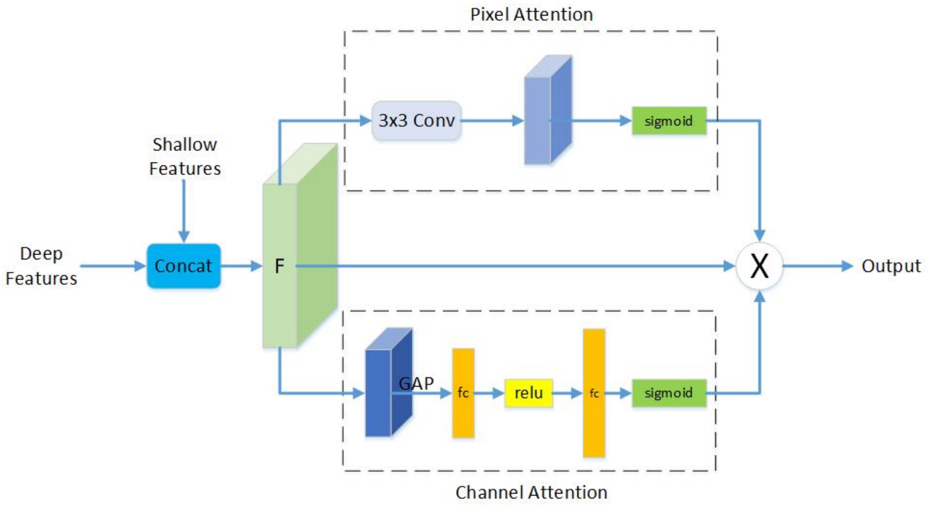


Figure 4.5: Multi-dimensional attention fusion network (MDAFN). The top branch is the pixel-level attention subnetwork, and the bottom is the channel-level attention subnetwork. Adapted from Fig. 4 in [Cao+22a] ©IEEE.

the model's receptive field to improve its feature extraction capability for multi-grasp objects. In this work, I introduce a multi-scale RFB [LHW18] to assemble the bottleneck layer to improve the model's receptive field capability. The RFB comprises multi-branch convolutional layers with different kernels corresponding to the receptive fields of various sizes. The dilated convolution layer controls the eccentricity, and the features extracted by the branches of the different receptive fields are recombined to form the final representation, as shown in Fig 4.4. In each branch, the convolutional layer follows a dilated convolutional layer. The kernel sizes are a combination of (1×1, 3×3, 7×1, 1×7). The features extracted from the four branches are concatenated and then added to the input data to obtain the final multi-scale feature output.

4.4.3 Multi-dimensional Attention Fusion Network

When humans look at an image, not much attention is paid to everything; instead, more focus is paid to what is interesting. In computer vision, attention mechanisms with few parameters, fast speed, and excellent effects have been developed [Wan+18; HSS18; Woo+18; Jen+20; Cao+21b]. The motivation for MDAFN is to effectively perceive grasping objects against a complex background. This attention mechanism can suppress the noise features and highlight the object features. As shown in Fig. 4.5, the shallow and deep features are concatenated together. The concatenated features are fed into MDAFN to perform representation learning at pixel-level and channel-level. The feature map F passes through a 3×3 convolution layer in the pixel attention subnetwork to generate an attention map by a convolution operation. The attention map is computed with a sigmoid to obtain the corresponding pixel-wise weight score. SENet [HSS18] is then used as the channel attention subnetwork, which accepts 1×1×C features through global average pooling. It then uses two feedforward layers and the corresponding activation function Relu to build the correlation between channels and finally outputs the weight score of the feature channel through the sigmoid operation. Both pixel-wise and channel-wise weight maps are multiplied with the feature map F to obtain a novel output with reduced noise and enhanced object information.

4.4.4 Loss Function

The neural network model can be considered as a method to approximate the complex function $F : I \longmapsto \hat{G}$ for input images $I = \{I_1...I_n\}$ and corresponding grasp labels $L = \{L_1...L_n\}$. F is the proposed grasp model and I is the input image. $\hat{G}$ denotes the grasp prediction. Specifically, the model is trained on the dataset to learn the grasp detection function F by optimizing minimum errors between grasp predictions $\hat{G}$ and the corresponding labels L. This task is a regression problem, so the Smooth L1 loss is deployed as the loss function to optimize my model. The loss function L_r is formulated as Eq. 4.11:

$$L_r(\hat{G}, L) = \sum_{i}^{N} \sum_{m \in \{q, \cos 2\theta, \sin 2\theta, w\}} \text{Smooth}_{L1}(\hat{G}_i^m - L_i^m) \tag{4.11}$$

where Smooth_{L1} is defined as:

$$\text{Smooth}_{L1}(x) = \begin{cases} (\sigma x)^2/2, & if\ |x| < 1; \\ |x| - 0.5/\sigma^2, & otherwise. \end{cases}$$

where N represents the count of grasp candidates, the grasp angle is defined as the form of $(\cos(2\theta), \sin(2\theta))$. And q, w denote the grasp quality and grasp width, respectively. σ is the hyperparameter in the Smooth_{L1} function, which controls the smooth area.

4.5 Dataset Analysis

Since deep learning has become popular, large public datasets, such as ImageNet [Den+09], COCO [Lin+14], KITTI [GLU12], etc., have been driving the progress of algorithms. However, in the field of robotic grasping detection, the number of available grasping datasets is insufficient. Dexnet, Cornell, Jacquard, and OCID are famous common grasping datasets that serve as a platform to compare the performance of the state-of-the-art grasping detection algorithms. In Tab. 4.1, it presents a summary of the different grasping datasets.

4.5.1 Dexnet Grasping Dataset

The Dexterity Network (Dex-Net) is a research project established by the UC Berkeley Automation Lab that provides code, a dataset, and algorithms for grasping tasks. At present, the project has released four versions of the dataset, namely Dex-Net 1.0, Dex-Net 2.0, Dex-Net 3.0, and Dex-Net 4.0. Dex-Net 1.0 is a synthetic dataset with over 10,000 unique 3D object models and 2.5 million corresponding grasp labels. Based on Dex-Net 1.0, thousands of 3D objects with arbitrary poses are used to generate more than 6.7 million point clouds and grasps, which constitute the Dex-Net 2.0 dataset. Dex-Net 3.0 is built to study the grasp using suction-based end effectors. Recently, an extension of previous versions, Dex-Net 4.0, has been developed, which can perform training for parallel-jaw and suction grippers. Since the Dex-Net dataset includes only synthetic point cloud data and no RGB information about the grasped objects, the experimentation in this work is mainly carried out on the Cornell, Jacquard, and extended OCID grasping datasets.

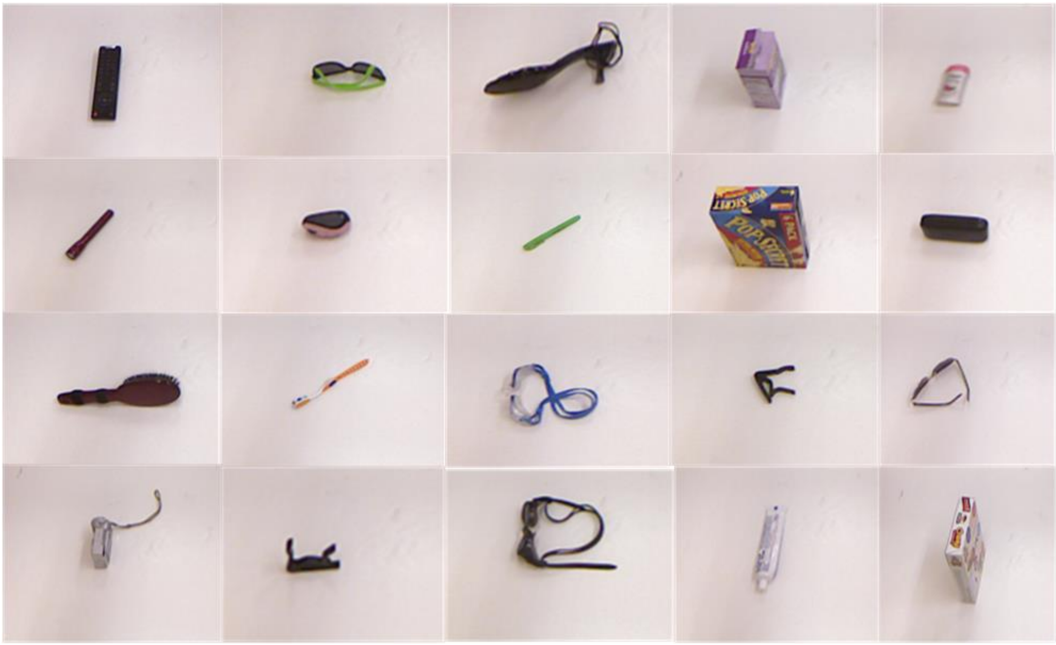


Figure 4.6: Qualitative images from the Cornell grasping dataset.

4.5.2 Cornell Grasping Dataset

The Cornell dataset, which is widely used as a benchmark evaluation platform, was collected in the real world with the RGB-D camera. Some example images are shown in Fig 4.6. The dataset is composed of 885 images with a resolution of 640×480 pixels of 240 different objects with positive grasps (5110) and negative grasps (2909). RGB images and corresponding point cloud data of each object in various poses are provided. However, the scale of the Cornell dataset is too small for training my convolutional neural network model. In this work, I use online data augmentation methods, including random cropping, zooms, and rotation, to extend the dataset and avoid overfitting during training.

4.5.3 Jacquard Grasping Dataset

Jacquard is a large grasping dataset created through simulation based on CAD models. Because no manual collection and annotation is required, the Jacquard dataset is larger than the Cornell dataset, containing 50k images of 11k objects and over 1 million grasp labels. In Fig. 4.7, it presents some images from the Jacquard datset. Furthermore, the dataset also provides a standard simulation environment to perform simulated grasp trials (SGTs) under a consistent conditions for different algorithms. In this work, I use SGTs as a benchmark to fairly compare the performance of various algorithms in the robot arm grasp. Since the Jacquard dataset is large enough, I do not apply any data augmentation methods to it.

4.5.4 OCID Grasping Dataset

$OCID_{grasp}$ is an extension dataset of the OCID dataset [Suc+19]. The original OCID dataset was collected to evaluate semantic segmentation methods; it contains RGB-D data with segmentation labels. The authors of [AF21] manually annotated the ARID10 and ARID20 subsets of the original OCID dataset with grasp labels. And each object's class information is

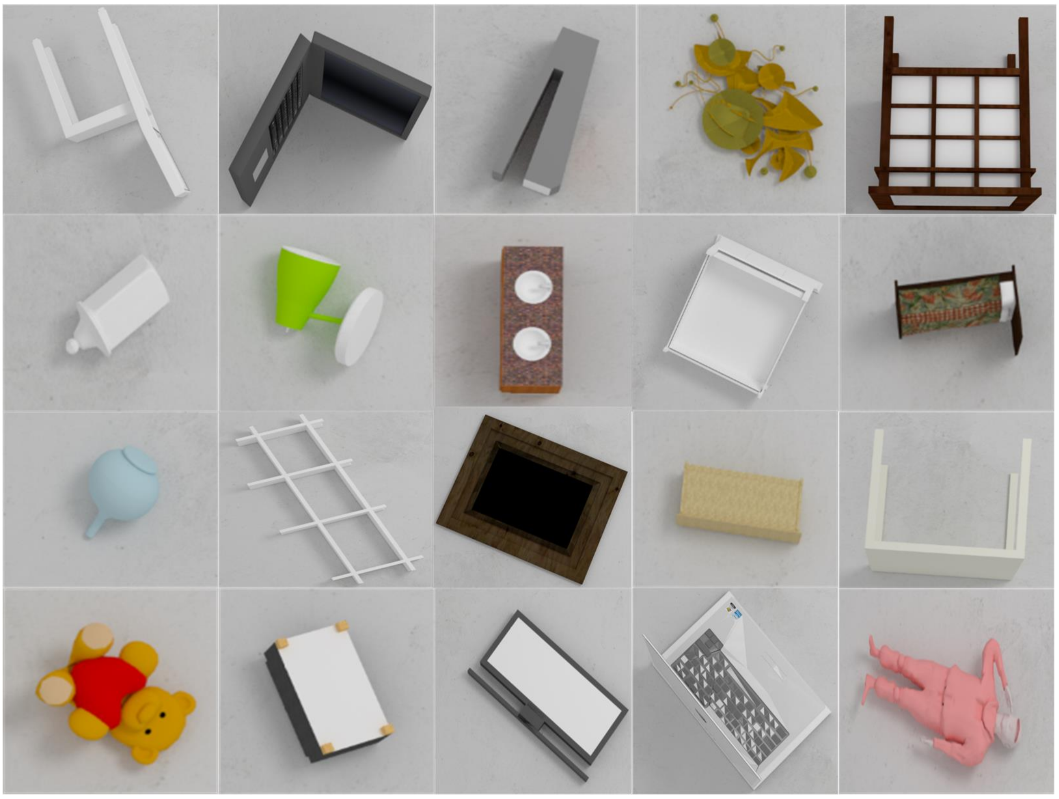


Figure 4.7: Qualitative images from the Jacquard grasping dataset.

Table 4.1: Description of the public Grasping Datasets. Adapted from Table I in [Cao+21a] ©IEEE.

Dataset	Modality	Objects	Images	Grasps
Dexnet	Depth	1500	6.7M	6.7M
Cornell	RGB-D	240	885	8019
Jacquard	RGB-D	11K	54K	1.1M
$OCID_{grasp}$	RGB-D	31	1763	75K

added to the $OCID_{grasp}$ dataset. $OCID_{grasp}$ consists of 31 object classes. Specifically, it consists of 1763 images with 11.4K segmentation masks and 75K grasp labels. In Fig. 4.8, several selected images are shown.

4.6 Experiments

To verify the generalization capability of the proposed lightweight generative model, I conducted experiments on three public grasp datasets, Cornell [YMS11], Jacquard [DDC18], and extended OCID [Suc+19; AF21]. Experimental results indicate that the proposed algorithm has high inference speed while achieving high grasp detection accuracy. In addition, I further explore the impact of different network designs on algorithm performance and discuss the shortcomings of the proposed method.

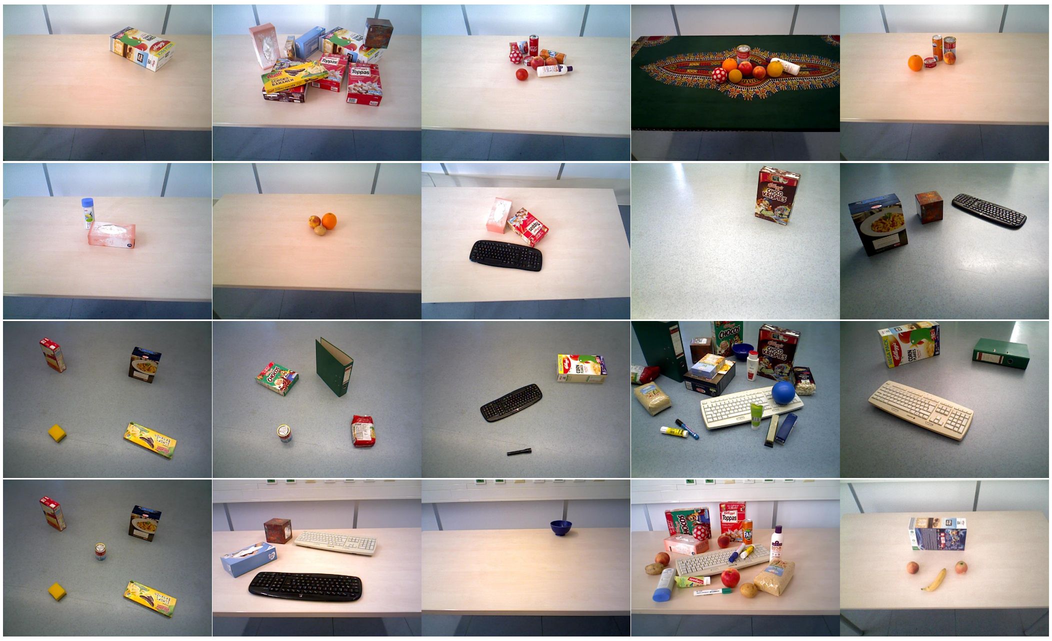


Figure 4.8: Qualitative images from the $OCID_{grasp}$ grasping dataset.

4.6.1 Implementation Details

Data Preprocessing. The experiments for this work were performed on the Cornell [YMS11], Jacquard [DDC18], and extended OCID [AF21] grasp datasets. Due to the small data size of Cornell and OCID, online data augmentation is conducted to train the network. Specifically, random crops, zooms, and rotations are used to improve the diversity of the Cornell and OCID grasp datasets. Meanwhile, the Jacquard dataset has sufficient data and the network is trained directly without any data augmentation. In addition, the data labels are encoded for training. A 2D Gaussian kernel is used to encode each ground-truth positive grasp so that the corresponding region satisfies the Gaussian distribution, where the peak of the Gaussian distribution is the coordinate of the center point. I also use $sin(2\theta)$ and $cos(2\theta)$ to encode the grasp angle, where $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The resulting corresponding values range from -1 to 1. Using this method, ambiguity can be avoided in the angle learning process, which is beneficial to the convergence of the network. Similarly, the grasp width is scaled to a range of 0–1 during the training.

Training Configuration. The grasp network is achieved using Pytorch 1.7.0 with Cudnn-7.5 and Cuda-10.0 packages. During the training period, the model is trained end-to-end on an Nvidia RTX2080Ti GPU with 11GB of memory.

4.6.2 Experiments on the Cornell Grasp Dataset.

The images of the Cornell dataset are resized to 224×224 to feed into the network. Following [MCL20], an image-wise split method is used to test my network, where the images of the dataset are randomly divided and the images of each object in the training set and test set are different. The average of the 5-fold cross-validation is used as the final results.

Training schedule. The famous Adam optimizer [KB15] is used to optimize the network for backpropagation during the training process. The initial learning rate is defined as 0.001

Table 4.2: Evaluation Results (%) of Different Methods on the Cornell Dataset. Runtime results for the methods †are referred to in [Wan+19a], the runtime results for the methods ‡are referred to in [KK17b], and the runtime results for the methods * are tested by ourselves. Adapted from Table I in [Cao+22a] ©IEEE.

Author	Method	Input Modality	Input Size	Accuracy(%)	Time (ms)
Jiang‡ [YMS11]	Fast Search	RGB-D	227 × 227	60.5	5000
Lenz† [LLS15]	SAE	RGB-D	227 × 227	73.9	1350
Chu† [CXV18]	FasterRcnn	RGD	227 × 227	96.0	120
Zhang† [Zha+17]	Multimodal Fusion	RGB-D	224 × 224	88.9	117
Zhou‡ [Zho+18]	FCGN	RGB	320 × 320	97.7	117
Redmon† [RA15]	AlexNet, MultiGrasp	RGB-D	224 × 224	88.0	76
Kumra‡ [KK17b]	ResNet-50	RGB-D	224 × 224	89.2	103
Kumra* [KK17b]	GR-ConvNet	RGB-D	300 × 300	97.7	7
Asif† [ATH18b]	GraspNet	RGB-D	224 × 224	90.6	24
Morrison* [MCL20]	GGCNN	D	300 × 300	73.0	4
Wang† [Wan+19a]	GPWRG	D	400 × 400	94.4	8
This work	Efficient Grasp	D	224 × 224	94.6	6
		RGB		95.3	6
		RGB-D		**97.8**	6

and the batch size is set as 8. The network is trained for a total of 50 epochs to get the final training weights.

Results. The comparison of the grasp detection accuracy of my model and other methods on the Cornell dataset [YMS11] is presented in Table 4.2. Since the grasping scene in the Cornell dataset is simple (single object grasping scene), the proposed grasp detection model achieves high detection accuracy of 97.8% with an inference time of $6ms$. The model maintains better accuracy and running speed performance than other state-of-the-art algorithms. By changing the mode of input data, the generated grasp detection architecture achieves excellent performance with the input of depth data. The results demonstrate that the combination of depth data and RGB data with rich color and texture information enables the model to have a more robust generalization ability to unseen objects. Fig. 4.9 shows the plot of the grasp detection results of some objects for display. Only the grasp prediction with the highest quality score is selected as the final output, and the top-1 grasp is visualized in the last row. The first three rows are the grasp quality, angle, and width maps. It can be seen that the proposed algorithm provides reliable grasp candidates for objects with different shapes and poses.

4.6.3 Experiments on the Jacquard Grasp Dataset.

The images of the Jacquard dataset are resized to 300×300 to feed into the network. I use an image-wise split to test my network, where 90% of the data is used as a training set, and the remaining data is used as a test set.

Training schedule. Similar to training the network on the Cornell dataset, I train the model end-to-end on the Jacquard dataset with a learning rate of 0.001 and a batch size of 8. Adam [KB15] is used as the default optimizer. Since the data size of the Jacquard dataset

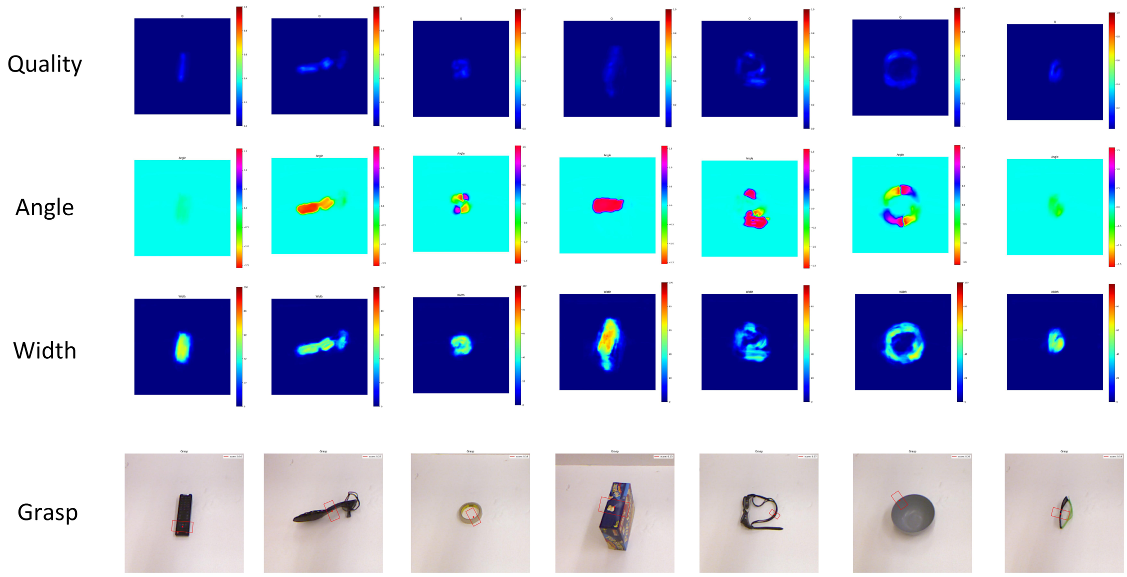


Figure 4.9: The detection results of the grasp network on the Cornell dataset. The first three rows are the grasp quality, angle, and width maps. And the last row is the best grasp output for several objects. Adapted from Fig. 5 in [Cao+22a] ©IEEE.

Table 4.3: Evaluation Results (%) of Different Methods on the Jacquard Dataset. The runtime results for the methods * are tested by ourselves. Adapted from Table II in [Cao+22a] ©IEEE.

Author	Method	Accuracy(%)	Time(*ms*)
Depierre [DDC18]	Jacquard	74.2	-
Morrison* [MCL20]	GG-CNN2	84.0	4
Kumra* [KK17b]	GR-ConvNet	94.6	7
This work	Efficient Grasp-D	**95.6**	6
	Efficient Grasp-RGB	91.6	6
	Efficient Grasp-RGB-D	93.6	6

is larger than the Cornell dataset, the network is trained for a total of 150 epochs to get the final training weights.

Results. Similarly, the network is trained on the Jacquard dataset [DDC18] to perform grasp pose estimation. The results are summarized in Table 4.3. Taking depth data as input, the proposed approach obtains excellent performance with a detection accuracy of 95.6%, which exceeds the existing methods and reaches the best result on the Jacquard dataset. The experimental results in Table 4.2 and Table 4.3 demonstrate that my algorithm achieves excellent performance on the Cornell grasp dataset and outperforms other methods on the Jacquard grasp dataset. Detection examples are displayed in Fig. 4.10. Specifically, grasp quality, angle, width, and the best detection results are presented in the figure.

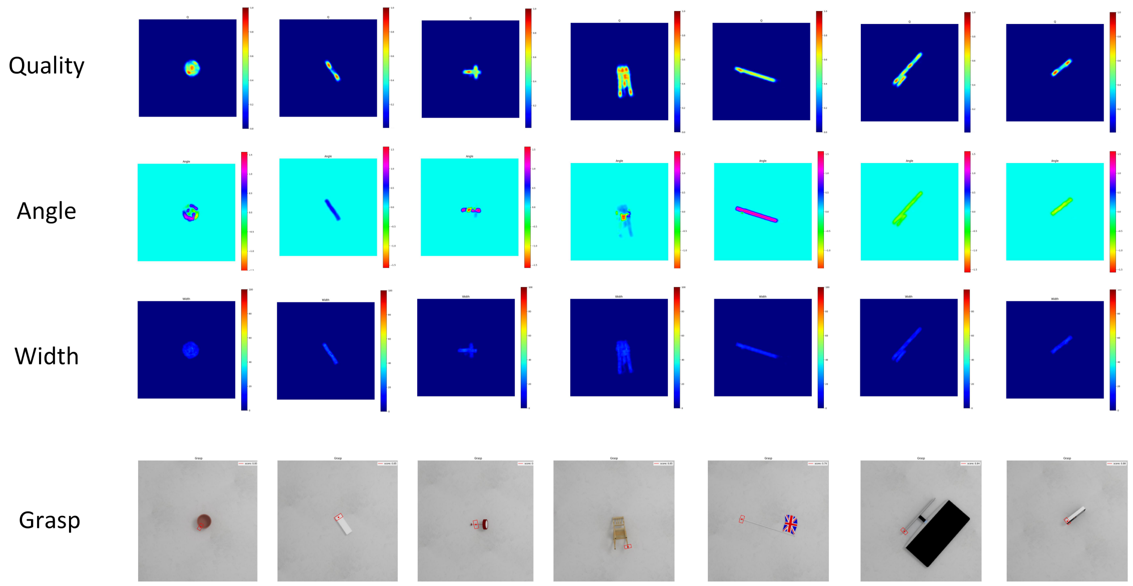


Figure 4.10: The detection results of the grasp network on the Jacquard dataset. The first three rows are the grasp quality, angle, and width maps. And, the last row is the best grasp output for several objects. Adapted from Fig. 6 in [Cao+22a] ©IEEE.

4.6.4 Experiments on the OCID Grasp Dataset.

The images of the extended OCID dataset are resized to 224×224 to pass through the network. The image-wise method is used to split the dataset. Specifically, 1411 selected images are divided into training set and 352 selected images are used as test set. I report the average of the 5-fold cross-validation as the final results.

Training schedule. The network is trained end-to-end on the extended OCID dataset with a learning rate of 0.001 and a batch size of 8. Adam [KB15] is used as the default optimizer, and the network is trained for a total of 400 epochs to get the final training weights.

Results. To verify the effectiveness of the proposed method on the complexity scenes, I test my method on the extended OCID [AF21] grasp dataset. The experimental results are shown in Table 4.4. The grasp detection accuracy of my method is better than contact point-based methods [MCL20; KK17b] and the running speed of my method is faster than detection-based method [AF21]. My method provides an excellent balance between accuracy and speed.

Objects in clutter. To validate the generalization ability of the proposed method in the cluttered scene, the model trained on the Cornell dataset is used to test it in a more realistic multi-object environment. The detection results are the first two rows presented in Fig. 4.11. The model is trained on a single object dataset but can still predict the grasp pose of multiple objects. Moreover, the last two rows presented in Fig. 4.11 are the test results of my model trained on the extended OCID dataset. The results show that the proposed method can simultaneously output grasp poses of various objects in complex scenarios.

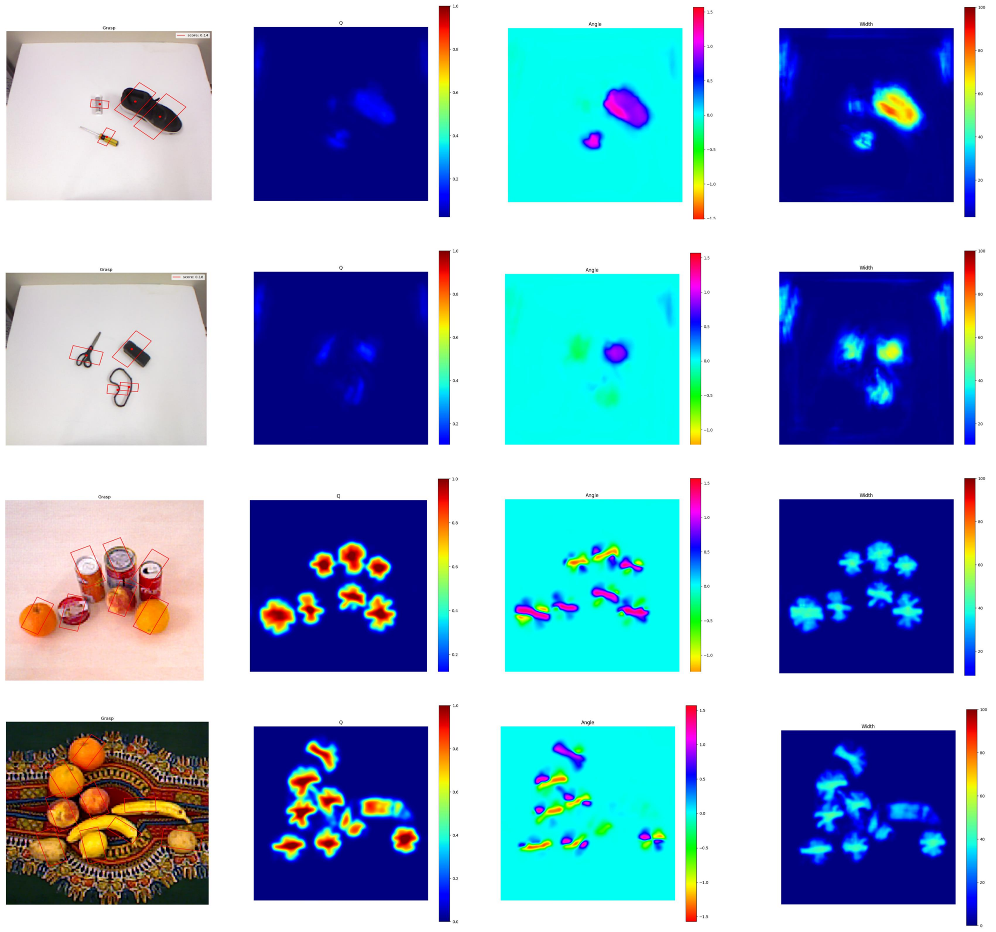


Figure 4.11: Multiple grasped object detection results The first column is the grasp outputs of corresponding RGB images for several objects. The last three columns are the maps for grasp quality, angle, and width. Adapted from Fig. 7 in [Cao+22a] ©IEEE.

Table 4.4: Evaluation Results (%) of Different Methods on the extended OCID Dataset. The runtime results for the methods * are tested by ourselves. Adapted from Table III in [Cao+22a] ©IEEE.

Author	Method	Accuracy(%)	Time(ms)
Stefan* [AF21]	Det_Seg	89.0	22
Morrison* [MCL20]	GG-CNN2	63.4	4
Kumra* [KK17b]	GR-ConvNet	74.1	7
This work	Efficient Grasp-D	72.7	6
	Efficient Grasp-RGB	74.7	6
	Efficient Grasp-RGB-D	76.4	6

4.6.5 Ablation Study

Influence of the different components. To further explore the impact of different components on grasp pose learning, I trained my models with varying network settings on the Cornell dataset [YMS11] with RGB-D data as input. The experimental results are summa-

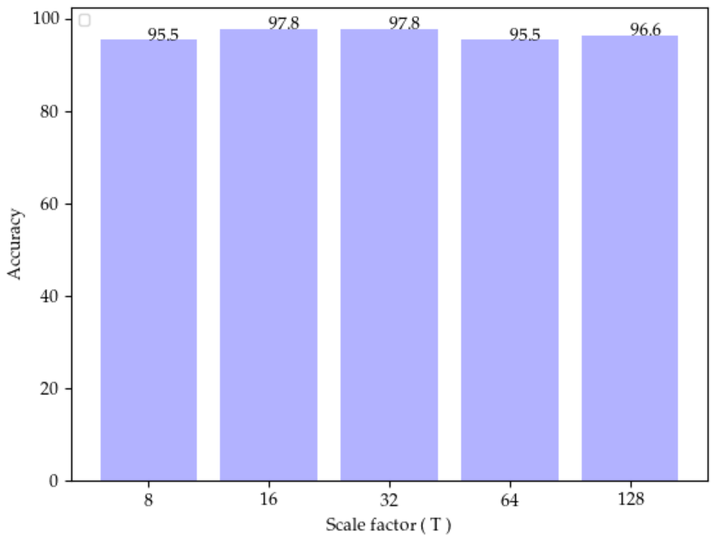


Figure 4.12: The grasp detection accuracy when using different scale factors of the Gaussian kernel. Adapted from Fig. 8 in [Cao+22a] ©IEEE.

Table 4.5: The impact of different network settings on detection performance. Adapted from Table IV in [Cao+22a] ©IEEE.

+ GGR + RFB + MDAFN	✓	 ✓	 ✓	 ✓ ✓	✓ ✓	✓ ✓	✓ ✓ ✓
Acurracy (%)	94.4	95.5	95.5	96.6	95.5	96.6	**97.8**

rized in Table 4.5. It can be obtained from the detection accuracy evaluation results in the Table 4.5 that Gaussian-based grasp representation (GGR), receptive field block (RFB), and multi-dimensional attention fusion network (MDAFN) can all bring performance improvement to the network, and all components combined can get the best grasp detection performance.

Effect of the scale factor. I also discuss the impact of different scale factor settings (T) on the model, as shown in Fig. 4.12. In this work, the scale factors T_x and T_y mentioned in Section 4.3.2 are set to $Tx = Ty = T$ with values ranging from $\{8, 16, 32, 64, 128\}$. When $T = 32$, the model training on the Cornell dataset reaches the best detection accuracy of 97.8. During the experiment, it is found that the different densities of the annotation for a particular dataset should be set to the size of the corresponding scale factor value, which can slow the instability of the network learning caused by labels' overlap.

Comparison of network efficiency. In Table 4.6, parameters, FLOPs, and the model's inference time (GPU and CPU) are used as efficiency evaluation metrics. To improve the real-time performance of the grasp algorithm, I developed a lightweight generative grasp detection architecture that achieves better detection accuracy and faster running speed. The experimental results show that the proposed method achieves excellent efficiency when executed

Table 4.6: Efficiency comparison of different methods (Approx).The results for the methods †are referred to in [MCL20]. The results for the methods * are tested by ourselves. Adapted from Table V in [Cao+22a] ©IEEE.

Methods	Params	FLOPs	Time (GPU)	Time (CPU)
Levine† [Lev+18]	1M	-	0.2-0.5s	-
Morrison* [MCL20]	70.6 k	1.0G	4ms	57ms
Kumra* [KK17b]	1.9M	10.9G	7ms	473ms
This work	1.2M	5.7G	6ms	86ms

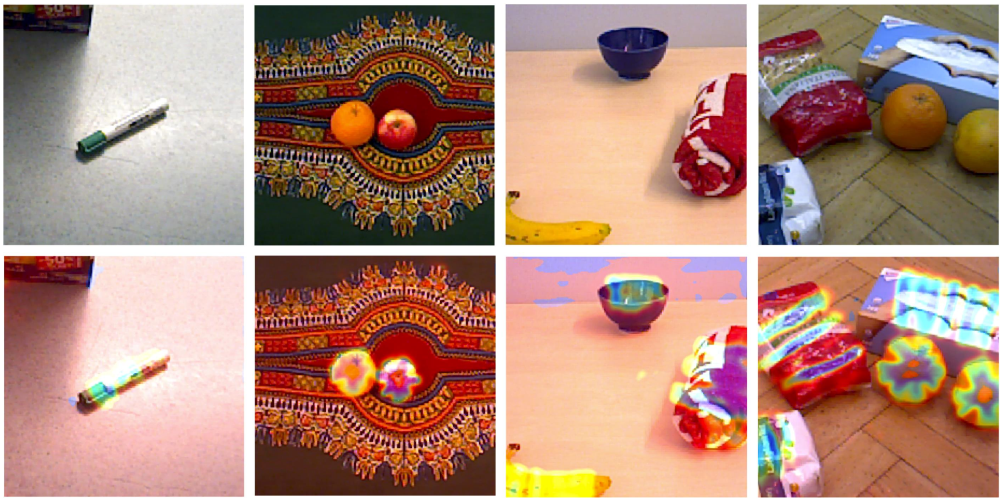


Figure 4.13: Visualization of feature heatmaps. Adapted from Fig. 9 in [Cao+22a] ©IEEE.

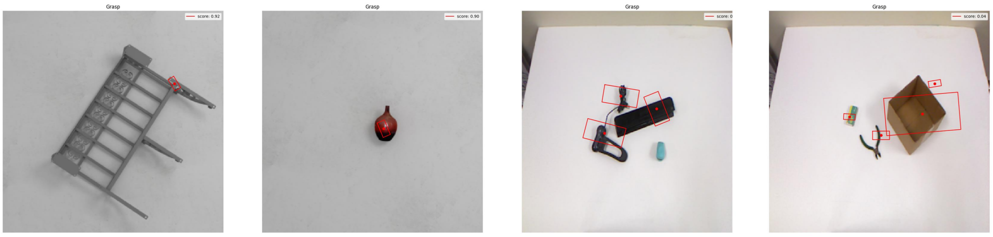


Figure 4.14: Failed detection cases with single and multiple objects. Adapted from Fig. 10 in [Cao+22a] ©IEEE.

on both GPU and CPU hardware.

Feature visualization. To help better understand the effectiveness of the proposed grasp model, I visualized the heatmaps of the feature maps, as shown in Fig. 4.13. The first row is the original images selected from the extended OCID dataset, and the second row is the corresponding heatmap visualization results of the feature maps. As can be seen from the figure, my grasp model can effectively focus on the object while suppressing unimportant background information.

Failure cases discussion. The experimental results show that the proposed method achieved excellent detection performance but still had some cases of detection failure, as shown in

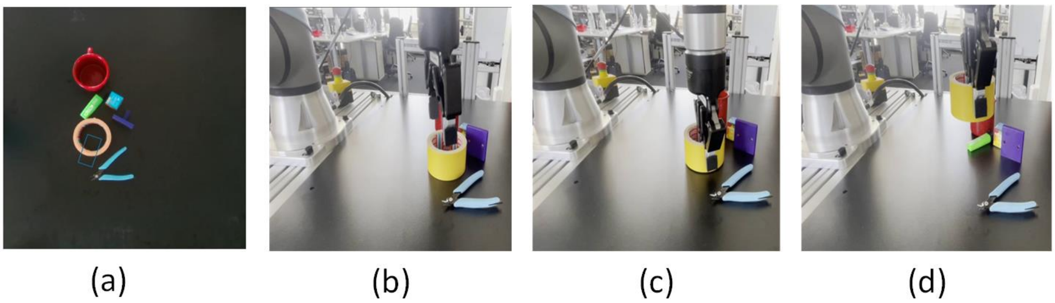


Figure 4.15: The process of physical grasp experiments. (a) Grasp detection output; (b) The robot approaches the object; (c) The robot grasps the object; and (d) The robot completes the successful grasp. Adapted from Fig. 11 in [Cao+22a] ©IEEE.

Table 4.7: The experimental results for the different grasp scenes. Adapted from Table VI in [Cao+22a] ©IEEE.

Scenes	Successes	Total Grasps	Grasp Success Rate(%)
Single object	94	100	94.0
Multiple object	142	156	91.0
Occluded object	114	128	89.1
Cluttered object	122	142	85.9

Fig. 4.14. The model does not work well for objects with complex shapes. Furthermore, in the clutter scenes, smaller objects among multiple objects are often missed by the model, and the detection quality of the model for large boxes is relatively insufficient. However, these shortcomings can be alleviated by adding more challenging data to the training set.

Verification on Real Robot. To evaluate the efficiency of the proposed model, a Universal Robot 5 (UR5) attached to a Robotic Gripper 2F-85 is chosen as my experimental instrument. The Universal Robot 5 offers a real-time data exchange interface with an update rate of 8ms, making it possible to achieve the real-time properties. I deploy the robotic library [RG17] as my primary platform to communicate with the robot. Furthermore, to build a compact system, the OPC-UA mechanism is integrated into the robotic library so that the camera can publish the images to the component, which can further utilize this information for object detection. Together with the OPC-UA, the robotic library shares a similar structure as ROS1, but much faster, since ROS1 lacks real-time properties. The whole experimental process is illustrated in Fig. 4.15.

I use the Intel Realsense camera to perceive the environment. The output of the Realsense camera will be fed to the proposed network, which can generate a bunch of grasp configurations, and a final grasp configuration will be selected based on my predefined criteria (the grasp candidate with the highest grasp quality score is selected as the final grasp configuration.) The coordinate transformation is necessary to apply the grasp configuration described in the image coordinate. After the transformation, the grasp configuration in the world coordinate is specified.

As a consequence, the robotic arm joints can be calculated using the analytical inverse kinematic approach. Therefore, a trajectory in joint space can be generated using the trajectory planning block from the robotic library. The novel objects are evaluated, with different and complex shapes. I summarize the results of single object grasp scene in Table 4.7 and indicate the effectiveness of my method with the 94% grasp success rate. To further test the performance of my method on more complex scenes, I performed real robot experiments on three multiple object grasp scenes: a multiple object scene, an occluded object scene, and a cluttered object scene. The robot attempts multiple grasps until all objects are grasped, and then grasped objects are removed. As shown in Table 4.7, my method has a grasping success rate of 91.0%, 89.1%, and 85.9% on the multiple object scene, occluded object scene, and cluttered object scene, respectively.

4.7 Summary

In this paper, I introduced a Gaussian-based grasp representation (GGR) to highlight the maximum grasp quality at the center position. Based on GGR, a lightweight generative architecture with a RFB and a MDAFN was developed for grasp pose estimation. Experiments on three common datasets, the Cornell [YMS11], Jacquard [DDC18], and extended OCID[AF21] datasets, demonstrate that the proposed method achieves a fast running speed of $6ms$ while having an excellent grasp detection accuracy of 97.8%, 95.6%, and 76.4%. In the physical grasp experiment, the proposed method achieves good performance with the application of the UR5 robot arm and robotic gripper.

5

Multimodal Learning for Object Perception Based on Event Camera

This chapter is about applications of multimodal learning with spatial attention for object perception (vehicle detection and grasp detection) based on event cameras.

The contents of this chapter are based mainly on the papers "Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception," that is published at the IEEE Signal Processing Magazine, 2020 [Che+20a], "Fusion-based Feature Attention Gate Component for Vehicle Detection based on Event Camera," that is published at the IEEE Sensors Journal, 2021 [Cao+21b], and "NeuroGrasp: Multi-modal Neural Network with Euler Region Regression for Neuromorphic Vision-based Grasp Pose Estimation" that is published at the IEEE Transactions on Instrumentation and Measurement, 2022 [Cao+22b].

5.1 Event Camera

Small insects such as bees outperform the most advanced artificial vision systems such as high-quality cameras nowadays in routine functions, including real-time sensing and processing, low-latency motion control, and so on. More importantly, such biological neural systems can well perform tasks with little energy consumption. In fact, biological neural systems usually consist of a large number of relatively simple elements. They operate on a massively parallel principle, which is different from the most common type of vision sensor, such as CMOS cameras. Thus, some researchers and engineers have tried to mimic the working principles of the biological visual systems and come up with a new artificial visual system.

Recently, the developments of material technologies, lithographic processes, very large scale integration (VLSI) design techniques, neuroscience, and neuromorphic technologies have enabled the novel conception and fabrication of bio-inspired visual sensors and processors. These new sensors and processors provide different methods to sense and perceive the world. The event-based neuromorphic vision sensor is a bio-inspired vision sensor that mimics the biological retina at both the system and element levels; it represents a paradigm shift in the acquisition, processing, and modeling of visual information. The Dynamic Vision Sensor (DVS) proposed by the group led by Tobi Delbruck is the first practicable event-based neuromorphic vision sensor based on biological principles [LPD08]. DVS captures the per-pixel brightness changes (called events) asynchronously instead of measuring the absolute brightness of all pixels at a constant rate, resulting in promising properties compared to standard frame-based cameras, such as low power consumption and low latency (in the order of microsecond), high dynamic range (120 dB), and high temporal resolution [Liu+19]. Thus, an alternative visual sensing and perception system for autonomous robots is provided in challenging scenarios that state-of-the-art standard frame-based cameras cannot well perform, such as high-speed scenes of autonomous highway driving, low-latency motion control, and low power consumption of the robot onboard system [Maq+18; Zhu+18a].

Frame-based vision sensors acquire the visual data as a sequence of snapshots recorded at discrete timestamps; therefore, the visual information is compressed and quantized at a pre-defined frame rate. Consequently, a problem (under-sampling) that is often known from the domain of signal processing arises due to the timescale of motions in the observed scenes and the frame rate of the recording camera. Things occurring between the adjacent frames would get lost. Generally, the advanced algorithms with multiple-sensor fusion are usually developed to compensate for single-sensor shortcomings in demanding applications such as highly piloted driving systems with low-latency motion control and visual feedback loops. Rather than solving this problem from an algorithm perspective, it is better to explore alternative methods from a novel sensing perspective, such as an event-based neuromorphic vision sensor, resulting in great value for promoting subsequent tasks to become more robust, accurate, and complementary together with the advanced algorithm development.

As bio-inspired and emerging sensors, event-based neuromorphic vision sensors have a different working principle compared to the standard frame-based cameras, which leads to promising properties such as low energy consumption, low latency, high dynamic range (HDR), and high temporal resolution. It poses a paradigm shift for sensing and perceiving the environment by capturing local pixel-level light intensity changes and producing asynchronous event streams.

5.1.1 Biological Retina

The retina of vertebrates, such as humans, is a highly developed multilayer neural system consisting of light-sensitive cells that contain millions of photoreceptors. It is the location where visual information is acquired and preprocessed. As shown in Fig. 5.1, the retina has three primary layers, including the photoreceptor layer, the outer plexiform layer, and the inner plexiform layer.

The photoreceptor layer consists of light-sensitive cells that convert incoming light into electrical signals and drive the horizontal cells and bipolar cells in the outer plexiform layer. There are two major types of bipolar cells, that is, the ON-bipolar cells and the OFF-bipolar cells. The ON and OFF bipolar cells are responsible for coding the bright and dark spatio-temporal contrast changes, respectively. If the illumination increases, the firing rate of the ON-bipolar cell increases, while the OFF-bipolar cell no longer generates spikes. This, in turn, increases the firing rate of OFF-bipolar cells in the case of illumination decreasing (such as getting darker). In the absence of a light stimulus, both cells generate a few random spikes. This phenomenon is achieved by comparing the photoreceptor signals with the spatio-temporal values, which are determined by the mean value of the horizontal cells, resulting in the facilitation of the connection between photoreceptors and bipolar cells laterally. In the outer plexiform layer, the ON- and OFF-bipolar cells synapse onto the amacrine cells, and ON- and OFF-ganglion cells are in the inner plexiform layer. The amacrine cells mediate signal transmission between bipolar cells and ganglion cells. The ganglion cells carry information along different parallel pathways in the retina, which is conveyed to the visual cortex. Thus, the retina is responsible for converting spatial-temporal illumination changes into pulses, which are transmitted to the visual cortex via the optical nerve.

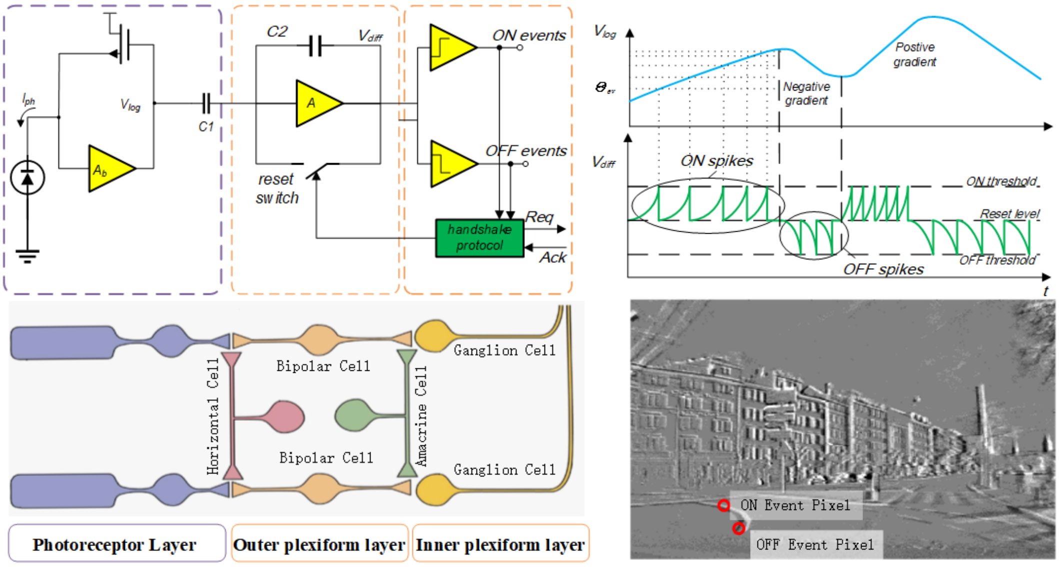


Figure 5.1: Three-layer model of a human retina (bottom-left) and corresponding dynamic vision sensor pixel circuitry (top-left). On the top-right, typical pixel circuit signals are depicted. The bottom-right image shows the accumulated events from a Dynamic Vision Sensor. The accumulated event map has ON (illumination increased) and OFF (illumination decreased) events drawn as white and black dots, respectively. Adapted from Fig. 2 in [Che+20a] ©IEEE.

5.1.2 Silicon Retina

The silicon retinas are visual sensors that model the biological retina and follow neurobiology principles. Pioneers of silicon retinas are Mahowald and Mead, who introduced their silicon VLSI retina in 1991 [MM91]. This kind of sensor is equipped with adaptable photoreceptors and a chip with a 2D hexagonal grid of pixels. It replicates parts of different cell types in biological retinas, including the photoreceptors, bipolar cells, and horizontal cells. Therefore, this kind of sensor represents merely the photoreceptor layer and the outer plexiform layer. Later, Zaghloul and Boahen built the Parvo-Magno retina, which is superior to the silicon VLSI retina by modeling five retinal layers.

Despite the promising structure, many of the early silicon retinas originated from the biological sciences community and were mainly used to demonstrate neurobiological models and theories without considering the real-world applications. Recently, an increasing amount of effort from Tobi Delbruck's team has been put into the development of a practicable silicon retina (the Dynamic Vision Sensor, or DVS) based on biological principles [LPD08]. In Fig. 5.1, the three-layer model of a human retina (bottom-left) and corresponding Dynamic Vision Sensor pixel circuitry (top-left) are presented. On the top-right, typical pixel circuit signals are displayed. The upper trace represents a voltage waveform at node v_{log} that tracks photocurrent through the photoreceptor layer circuit. The outer plexiform layer circuit responds with spike events (v_{diff}) of different polarities to positive and negative changes of the photocurrent. Spikes are transported to the next processing stage by the inner plexiform layer circuit. A large number of log-intensity changes are encoded in the events. The bottom-right image illustrates the accumulated events, including ON events (illumination increased) and OFF events (illumination decreased), that are drawn as white and black dots.Today's representatives of silicon retinas are mainly the pioneers Tobi Delbruck and Christoph Posch, representing a compromise between biological and technical aspects. In their development, one prominent challenge posed is usually regarded as a wiring problem, indicating that each pixel of the silicon retina needs its own cable, which is impossible for chip wiring. A key

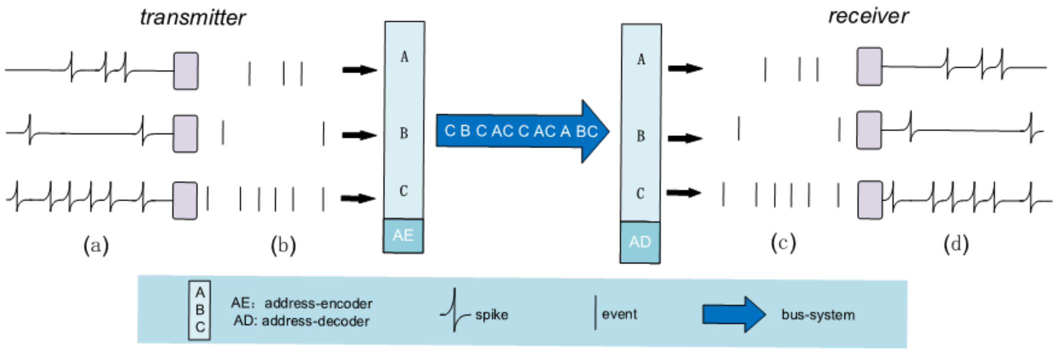


Figure 5.2: The AER communication protocol: (a) Three neurons on the sending chip generate spikes. (b) The spikes are interpreted as binary events. A binary address is generated by the address encoder (AE). This address is transmitted to the receiver chip by the bus line. (c) The binary address is decoded to the binary event by the address decoder (AD). (d) Spikes are emitted on the corresponding neurons of the receiver chip, where the positions of the neurons are determined by the address decoder. Adapted from Fig. 3 in [Che+20a] ©IEEE.

technique for the solution, named Address Event Representation (AER), was originally from the Caltech group of Carver Mead; it is used as an event-controlled and asynchronous point-to-point communication protocol for prototypes of the silicon retina.

As illustrated in Fig. 5.2, the basic functionality of AER is implemented by an address encoder, an address decoder, and a digital bus. All neurons and pixels could transmit the time-coded information on the same line because the digital bus implements a multiplex strategy. The address encoder of the sending chip generates a unique binary address for each neuron or pixel in the event of a change. The bus transmits the address at high speed to the receiver chip. Then, the address decoder determines the position and generates a spike on the receiver neuron. Event streams are employed in AER to communicate among chips. An event is a tuple (x, y, t, p): x and y are pixel addresses; t is the timestamp; and p represents the polarity. The polarity indicates an increase and decrease in the lighting intensity, corresponding to ON and OFF events, respectively.

My work mainly focuses on the first practically usable silicon retina, the Dynamic Vision Sensor (DVS)[1], which follows the natural, frame-free, and event-driven approach that triggers a plethora of research in event-based neuromorphic vision and robot. The DVS pixel models a simplified three-layer biological retina by mimicking the information flow of the photoreceptor-bipolar-ganglion cells (see Fig. 5.1). Pixels operate independently and attach special importance to the temporal development of the local lighting intensity. The DVS pixel would automatically trigger an event (either an ON event or an OFF event) when the relative change in intensity exceeded the threshold. Therefore, the working principle of the DVS is fundamentally different from that of the frame-based camera. There are three key properties of biological vision that are kept in this silicon retina, including the relative illumination change, the sparse event data, and the separate output channels (ON/OFF). The major consequence of the DVS is that the acquisition of visual information is no longer controlled by any form of external timing signals such as frame clocks or shutters, while the pixel itself controls its own visual information individually and autonomously.

[1] A recent approach by Tobi Delbruck is the so-called dynamic and active pixel vision sensor (DAVIS) that combines dynamic and static visual information into a single pixel.

5.1.3 Advantages of Event Cameras

Due to their fundamentally different working principle and mimicking of the biological retina, the event-based neuromorphic vision sensors have several advantages over standard frame-based cameras.

Energy Friendly. Since an event-based neuromorphic vision sensor transmits only events and autonomously filters redundant data, power is only used to process active pixels (e.g., the events triggered by illumination changes). Particularly, energy-friendly sensors are more important than advanced algorithms for the onboard computers and devices in autonomous vehicles.

Low Latency. There is no need for the global exposure of the frame because each pixel works independently. Ideally, the minimal latency is 10 μs. The low-latency control of the autonomous vehicle is highly dependent on the perception systems. A low-latency perception system such as an object detection system based on an event-based neuromorphic vision sensor would save lots of time to avoid obstacles for the control systems.

High Dynamic Range. The event-based neuromorphic vision sensor, such as DVS, has a high dynamic range (120 dB), which far exceeds that of the frame-based cameras (60 dB). Event-based neuromorphic vision sensors such as the DVS can simultaneously adapt to very dark and bright stimuli, ensuring a highly robust perception system even in a light-changing scene such as an autonomous vehicle driving through a tunnel.

Microsecond Resolution. It is fast to capture brightness changes in analog circuitry. With a 1 MHz clock, events can be detected and timestamped with microsecond resolution. Considering the fast response requirement of the controller in autonomous vehicles in emergency driving scenes, this property is quite useful in autonomous driving.

No Motion Blur. In the high-speed driving scenario, the motion blur problem occurs when the motion of the moving objects is beyond the sampling frequency of the frame-based camera; this may cause the failure of the perception system. An event-based neurmorphic vision sensor that can capture dynamic motion precisely with no motion blur is of great value to the autonomous driving community.

5.1.4 Event Noise Processing

The preprocessing of the raw data is essential for extracting meaningful information for sensor system. Event-based neuromorphic vision sensor not only captures the change in the light intensity caused by moving objects, it also generates some noise activities due to the movements of background objects and the sensor noise such as temporal noise and junction leakage currents [Liu+15; PBO18; KK17a]. As shown in Fig. 5.3, the event-noise processing technique is responsible for excluding the event noises from the event stream. Two commonly used methods in the literature, namely the spatial-temporal correlation filter and the motion consistency filter, are illustrated as follows.

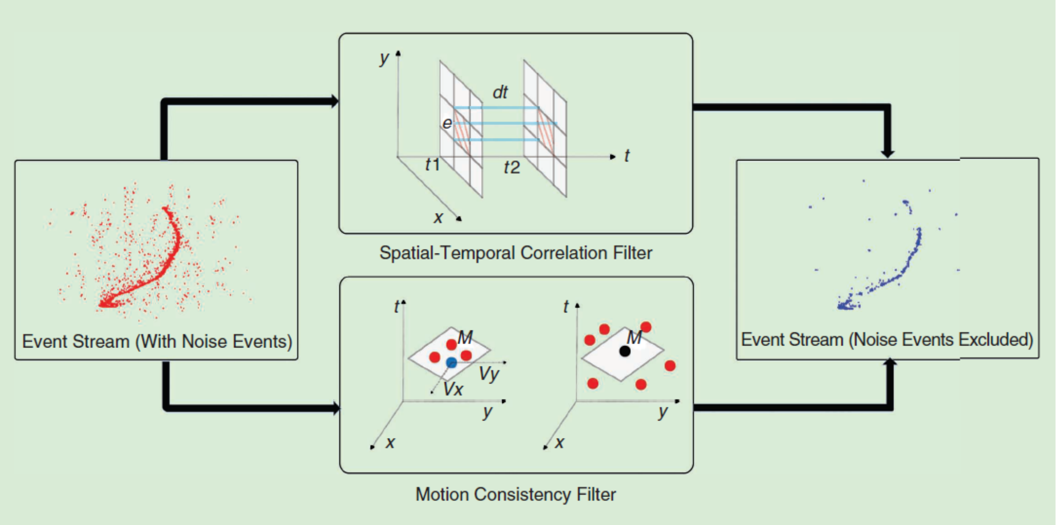

y
dt
e
t1
t2
t
x
Spatial-Temporal Correlation Filter
Event Stream (With Noise Events)
Event Stream (Noise Events Excluded)
t
M
Vy
Vx
y
x
t
M
y
x
Motion Consistency Filter

Figure 5.3: Event noise processing: the green dashed box shows the spatial-temporal correlation filter; the orange one is the motion consistency filter. Adapted from Fig. 4 in [Che+20a] ©IEEE.

Spatial-temporal correlation filter. For a newly incoming event $e_i = (x_i, y_i, t_i, p_i)$, the spatial-temporal filter searches the most recent neighborhood event around the current pixel location (x_i, y_i) within a distance D. The incoming event would be regarded as a nonnoise event if the time difference meets:

$$t_i - t_n < d_t \tag{5.1}$$

where t_i is the timestamp of the event; t_n is the timestamp of the most recent neighborhood event; and d_t is the predefined threshold. The search for the most recent event checks eight neighborhood pixels around (x_i, y_i), as shown in Fig. 5.3. It lacks temporal correlation with events in their spatial neighborhood because the event-noise occurs randomly. Hence, the spatial-temporal correlation filter can effectively filter out event noise.

Motion consistency filter. In Fig. 5.3, the principle of the motion consistency filter [Wan+19b] is depicted. The blue dot denotes an incoming event caused by the object motion and the black dot represents an event-noise. In the spatial-temporal domain, a newly incoming event should be consistent with the previous events (represented by red dots) caused by the same moving object. In a local region, the incoming event can be modeled as a consistent "moving plane" M. In this way, the velocity (v_x, v_y) can be used to assess the motion consistency, and the event-noise can be removed because the previous events (the red dots, signal) and the black dot are not on the same plane. Concretely, the motion consistency plane for each active event e_i can be formulated as

$$ax_i + by_i + ct_i + d = 0 \tag{5.2}$$

where $(a, b, c, d) \in \mathbb{R}^4$ defines the plane M; (x_i, y_i) is the coordinate of event e_i; and t_i is the timestamp of event e_i. The event noise processing is an essential step to extract useful information from unwanted noise data for bio-inspired visual sensing and perception tasks of autonomous driving; it can promote the accuracy and speed of subsequent algorithms.

Table 5.1: The comparison of different event data representations of spatial-temporal encoding. H and W represent the image height and width dimensions, respectively; B denotes the number of temporal bins. The polarity channel is 2 if the encoding method considers the polarities of events, otherwise is 1. Adapted from Table I in [Che+20a] ©IEEE.

Representation	Dimensions	Polarity Channel	Intensity	Weakness
Surface of Active Events (SAE)	H× W	2	timestamp of the most recent event	without temporal history
Leaky-Integrate and Fire (LIF)	H× W	1	event spikes	without polarity information
Voxel Grid	B× H× W	1	sum event polarities	without polarity information
Event Spike Tensor (EST)	B× H× W	2	sample event point-set into the grid	without the least amount of information

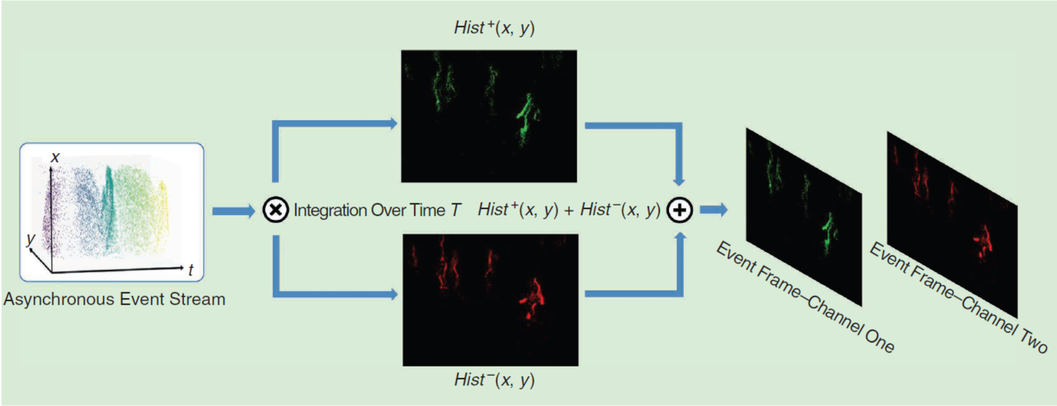


Figure 5.4: The process of converting asynchronous event data into an event-frame. An event-frame consists of two histograms from the positive events and negative events respectively. Adapted from Fig. 5 in [Che+20a] ©IEEE.

5.1.5 Event Representations

As an emerging sensing modality, event-based neuromorphic vision sensors only transmit local pixel-level changes caused by movement or a change in light intensity in a scene. The output data are sparse and asynchronous event streams that cannot be processed directly by standard vision pipelines such as CNN-based architecture. Therefore, encoding methods are utilized to convert asynchronous events into synchronous image- or grid-like representations for subsequent tasks such as object detection and tracking. According to whether or not the methods contain temporal information in the converted representations, we introduce two state-of-the-art encoding methods: **spatial encoding** and **spatial-temporal encoding** methods.

Spatial encoding. The spatial encoding methods convert event streams into event-frames by storing event data at pixel location (x_i, y_i) with either fixed-time interval (e.g., $30ms$, constant time frame) or fixed number of events (e.g., 500 events, constant count frame). For an event-frame, the value of the pixel is usually represented by the polarity of the last event (the positive event is 1 and the negative event is -1) or the statistical characteristics (such as the event count in a fixed-time interval, event count frame) of the events in the fixed interval. Assuming that $e_i(x_i, y_i, t_i, p_i)_{i\in[1,N]}$ represents event stream, typical approaches based on spatial encoding can be defined as follow:

Constant time frames:

$$F_j^t = \mathbf{card}(e_i | T \cdot (j-1) \leq t_i \leq T \cdot j) \tag{5.3}$$

where F_j^t represents the jth frame of time interval T; **card**() is the cardinality of a set; e_i is the ith event of the event stream.

Constant count frames:

$$F_j^e = \mathbf{card}(e_i | E \cdot (j-1) \leq i \leq E \cdot j) \tag{5.4}$$

The constant count frame is defined similarly to constant time frame. F_j^e is the jth frame that contains E events.

Event count frames:

$$Hist^+(x,y) = \sum_{p_i=+1, t_i \in T} \delta(x - x_i, y - y_i) \tag{5.5}$$

Two separate histograms for positive and negative events are generated in a fixed-time interval T. $Hist^+(x,y)$ denotes the histogram for positive events, where δ is the Kronecker delta function. The same goes for the negative-events histogram, which is represented by $Hist^-$ with $p_i = -1$. The final representation of the events in the fixed-time interval T is an event-frame, which consists of two histograms $Hist^+$ and $Hist^-$, as shown in Fig. 5.4. Since the principle of the spatial encoding method is to project the events onto the spatial plane ($x - y$ plane), it loses the temporal information of all of the events.

Spatial-temporal encoding. The microsecond temporal resolution of the event stream provides a highly precise recording and description of the scene dynamics, which is valuable in many perception tasks such as high-speed moving object detection (e.g., vehicles). Spatial-temporal encoding methods combine spatial and temporal information of the events and convert events into a compact representation. A comparison of spatial-temporal encoding methods is presented in Table. 5.1. A detailed description of these methods is displayed as follows.

Surface of active events: The surface of active events (SAE) uses timestamp values instead of intensity values to represent the pixel values. For each incoming event e_i:

$$SAE : t_i \longmapsto P(x_i, y_i) \tag{5.6}$$

where t_i is the timestamp of the most recent event at each pixel, the pixel value P at (x_i, y_i) is directly determined by the occurrence time of the events. The disadvantage of SAE method is that it completely ignores the information of previous events happening at (x_i, y_i) and only uses the timestamp of the most recent event.

Leaky integrate-and-fire: Leaky integrate-and-fire (LIF) is an artificial neuron inspired by biological perception principles and computation primitives. A neuron receives input spikes (events) generated from a DVS, which modifies its membrane potential. If the membrane potential exceeds a pre-defined threshold, a spike stimulus will be sent to the output. The LIF neuron can be modeled as

$$\tau \frac{dV}{dt} = -(V(t) - V_{reset}) + RI(t) \tag{5.7}$$

where, $V(t)$ is the membrane potential, which is a function across time; $I(t)$ is the total synaptic current; R is the membrane resistance; and τ is the membrane time constant. The neuron fires (produces a output spike) when the membrane potential reaches the threshold voltage (V_{th}) and then resets to reset voltage (V_{reset}). As shown in Fig. 5.5, the spatial-temporal events are encoded by LIF neuron, in which each event updates membrane potential of the neuron and the final converted representation is composed of the output spikes.

Voxel grid: Voxel Grid is a novel event representation aiming to improve the resolution of event stream in the temporal domain. Given a set of N events $(x_i, y_i, t_i, p_i)_{i \in [1,N]}$, B bins

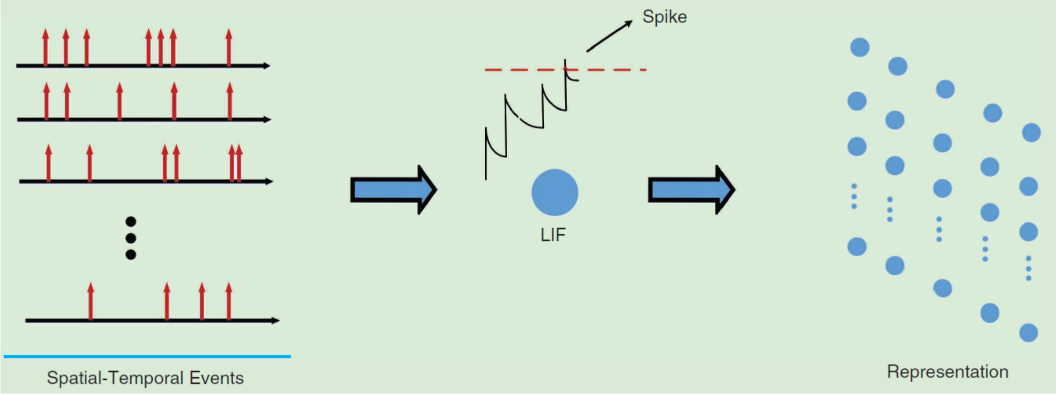


Figure 5.5: LIF representation: Asynchronous spatial-temporal events are converted into event data representation by LIF neurons. Adapted from Fig. 6 in [Che+20a] ©IEEE.

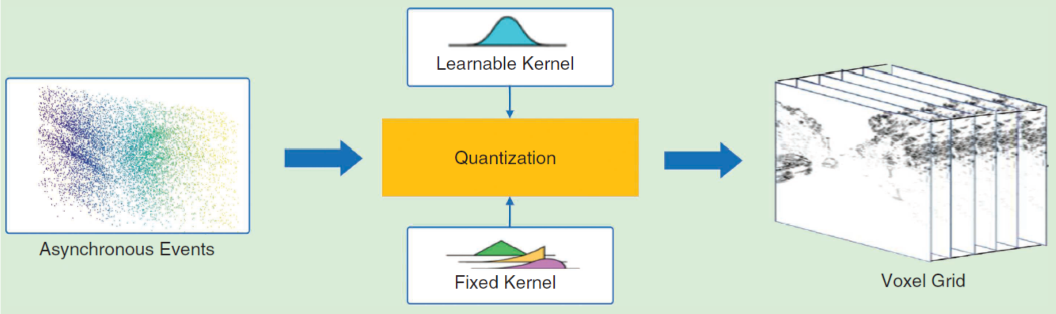


Figure 5.6: An illustration of converting asynchronous event data into Grid-based representation with fixed kernel [Jad+15] and learnable kernel [Geh+19]. Adapted from Fig. 7 in [Che+20a] ©IEEE.

are used to split the time dimension; then, the timestamps of events are scaled to the range of $[0, B-1]$. The event voxel grid is defined as

$$\hat{t} = (B-1)(t_i - t_1)/(t_N - t_1) \tag{5.8}$$

$$V(x, y, t) = \sum_{i}^{N} p_i k(x - x_i) k(y - y_i) k(t - \hat{t}) \tag{5.9}$$

$$k(z) = max(0, 1 - |z|) \tag{5.10}$$

where $k(z)$ is the trilinear voting kernel which is equivalent to the definition in [Jad+15]. As shown in Fig. 5.6, events are converted into voxel grid representation with the fixed kernel. This representation retains the distribution of the events across the spatial-temporal dimensions.

Event spike tensor: Event spike tensor (EST) is an end-to-end learned representation [Geh+19]. In a given time interval T, EST can be formed by sampling the convolved signal,

$$S_{\pm}[x, y, t] = \sum_{e_i \in p_{\pm}} f_{\pm}(x_i, y_i, t_i) k_c(x - x_i, y - y_i, t - t_i) \tag{5.11}$$

where $f_{\pm}(x_i, y_i, t_i)$ is a measurement assigned to each event to represent the corresponding intensity value at the pixel location. k_c is the kernel convolution function to derive meaningful signal from event stream. Generally, both measurement and kernel are hand-crafted functions in previous works, as illustrated in Fig. 5.6. Particularly, the EST deploys a multilayer perception (MLP) replacing the hand-crafted kernel function in Eq. 5.11 to fit the data

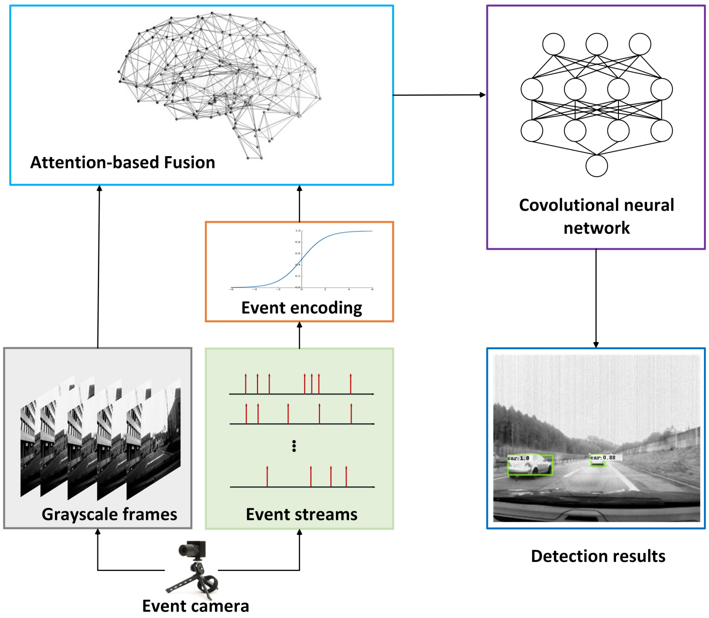
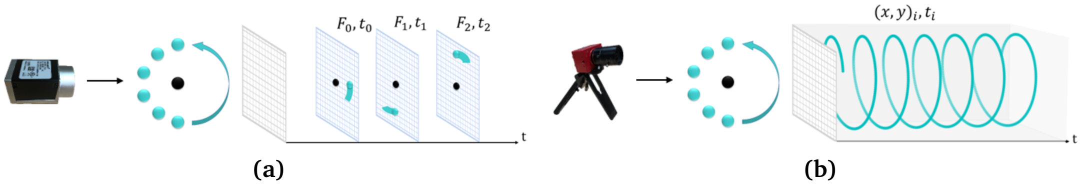


Figure 5.7: The framework of the fully convolutional neural network with a feature attention gate component (FAGC) for vehicle detection. Adapted from graph abstract in [Cao+21b] ©IEEE.

with the purpose of finding the best function for event streams. Simultaneously, the measurement function is chosen from a set of fixed functions. Examples of such function are the event polarity $f_{\pm} = \pm 1$; the event count $f_{\pm} = 1$; the timestamp $f_{\pm} = t$; and the normalized timestamp $f_{\pm} = \frac{t-t_0}{T}$.

5.2 Fusion-Based Feature Attention Gate Component for Vehicle Detection

In the field of autonomous vehicles, various heterogeneous sensors, such as LiDAR, radar, cameras, etc., are combined to improve the vehicle's ability to sense accurately and robustly. Multi-modal perception and learning has been demonstrated to be an effective method for assisting vehicles in comprehending the nature of complex environments. The event camera is a bio-inspired vision sensor that captures dynamic changes in the scene and filters out redundant information with high temporal resolution and high dynamic range. These characteristics of the event camera make it have a certain application potential in the field of autonomous vehicles. In this paper, I introduce a fully convolutional neural network with a feature attention gate component (FAGC) for vehicle detection by combining frame-based and event-based vision. The overall framework is shown in Fig. 5.7. Both grayscale features and event features are fed into the feature attention gate component (FAGC) to generate the pixel-level attention feature coefficients to improve the feature discrimination ability of the network. Moreover, I explore the influence of different fusion strategies on the detection capability of the network. Experimental results demonstrate that my fusion method achieves the best detection accuracy and exceeds the accuracy of the method that only takes a single-mode signal as input.



(a) (b)

Figure 5.8: Comparison of the output between a standard frame-based camera and an event camera [Che+20c]. (a) The frame-based camera captures images at a fixed frame rate. (b) The event camera emits events caused by the moving objects asynchronously. Adapted from Fig. 1 in [Cao+21b] ©IEEE.

5.2.1 Background

For autonomous vehicles, a reliable perception system can provide the state and pose of the objects. Vehicle detection plays an important role in the field of autonomous driving. For autonomous vehicles, it is equipped with various sensors, such as cameras, lidar, and radar, to sense the environment. By combining a variety of heterogeneous sensors, autonomous vehicles can sense obstacles and avoid accidents [Urm+08; Che+20a; Che+21a; Che+21b]. The frame-based camera acquires the visual data as a sequence of frames at a fixed frequency. Currently, thanks to the breakthrough in deep learning technology, frame-based object detection methods such as SSD [Liu+16], YOLO [RF16], and FasterRCNN [Ren+15] have achieved excellent performance. However, frame-based cameras still suffer from the challenges of overexposure and motion blur in high light and fast motion [Gal+20]. In this work, I try to introduce an event camera for vehicle detection tasks. The pixel-level changes caused by motion and brightness changes are captured by event cameras. Different from a frame-based camera, the event camera outputs high temporal resolution and high dynamic range (120 dB) event streams [Liu+19; Che+20b]. The comparison of the output between a frame-based camera and an event camera is presented in Fig. 5.8.

Some research into the application potential of event cameras has been proposed. In [Zhu+18b], the authors use event camera to predict optical flow by using the data from the MVSEC [Zhu+18a] dataset collected by themselves. The first event-based semantic segmentation is introduced in [AM19]. A Xception-based convolutional neural network (CNN) is trained on the Ev-Seg dataset to learn segmentation from events. The researchers also applied the event camera to perform the end-to-end steering angle prediction [Maq+18]. Recently, neuromorphic vision based safe driving system is built in [Che+20c; Che+20d]. In [Che+20c], the driver drowsiness detection is completed through facial motion analysis using an event camera. And a new database and baseline evaluation method are proposed in [Che+20d]. For event-based object detection, several works, such as [Che18; Che+19; Jia+19], have been done for vehicle or pedestrian detection. However, these methods focus on how to improve the detection accuracy of event-based detectors. Since the event streams lack appearance features such as texture and color information, it is difficult to achieve high object detection accuracy by using only the event streams as input. Now, there is still a lack of research on how to fuse frame-based and event-based multi-modal features. Hence, it is necessary to study the fusion of event information and other input signals.

In this work, I introduce a fusion-based feature attention gate component (FAGC) for vehicle detection based on event cameras. To take advantage of grayscale frames with texture features and events with high dynamic range, both grayscale frames and event streams are fed into the network to fuse together and complement each other. Based on this mechanism, the experimental results on the labeled DDD17 dataset [Bin+17; Li+19a] indicate that the detection accuracy of the vehicle detection network with FGAC is significantly improved, which is better than the method that only takes grayscale frames as input or only takes event streams as input. My detailed contributions are as follows:

- A vehicle detection method based on event cameras is introduced for autonomous vehicle perception.
- I develop a feature attention gate component (FAGC) to fuse grayscale-based features and event-based features to improve the performance of the vehicle detector. The impact of different fusion strategies and event representations is discussed.
- The experimental results on the labeled DDD17 dataset show that the detection accuracy of vehicle detectors can be significantly improved by combining frame-based vision and event-based vision.

5.2.2 Related Work

Methods of Frame-based Object Detection. Early frame-based object detection methods are based on handcrafted features such as the histogram of oriented gradients (HOG) [DT05a] and aggregate channel features (ACF) [Dol+14]. However, with the rise of deep learning, a large number of object detection algorithms based on deep learning have emerged. Current deep learning-based object detection algorithms are mainly divided into one-stage [RF16; Lin+20; Liu+16; Dua+19; Tia+20] and two-stage detectors [Ren+15; He+20], which have been applied in many fields [Li+20b; Lin+17; Li+20a]. For two-stage detectors, Faster-RCNN [Ren+15] and Mask-RCNN [He+20] achieve high detection accuracy based on the region proposal network (RPN). Compared with two-stage methods, the one-stage detectors achieve a better balance between accuracy and speed, such as YOLO [RF16], SSD [Liu+16] and Retinanet [Lin+20]. However, YOLO [RF16], SSD [Liu+16] and RetinaNet [Lin+20] are all anchor-based object detectors. Recently, anchor-free methods have been developed rapidly and achieved excellent performance, such as Centernet [Dua+19] and FCOS [Tia+20]. Compared with the frame-based object detection method, the event-based method is still in its preliminary stage.

Methods of Event-based Object Detection. For event-based vision, several works attempt to apply the event camera in various fields, such as intelligent transportation system [Che+18] and robotic grasping [Li+20a]. Compared with frame-based object detection, a small amount of research has been done on event-based object detection [Che18; Jia+19; Che+19; Zan+19; Li+19a; HDL20; Per+20]. More event-based-related works can be found in [Che+20a; Gal+20]. In [Che18], the authors use grayscale frames to pass through the state-of-the-art object detector to generate the pseudo-labels that are used for training the detector model, taking the events as input. And, a joint detection framework is introduced in [Li+19a] to combine the frame-based and event-based vision for autonomous driving. Different from focusing on vehicle detection under ego-motion in the work [Che18; Li+19a], the study in [Jia+19] concentrates on pedestrian detection in the field of intelligent transportation systems. The fusion method based on confidence maps is proposed in [Jia+19] to improve pedestrian detection accuracy. Moreover, in order to take full advantage of the event information, multi-cue event information fusion is being developed in [Che+19] for pedestrian detection. Recently, [Zan+19] and [HDL20] attempted to use an RGB-based detector to improve the performance of the event-based detector. And, the event-based detection method and a high-resolution large-scale dataset are introduced in [Per+20]. The results of the experiment demonstrate the effectiveness of their method.

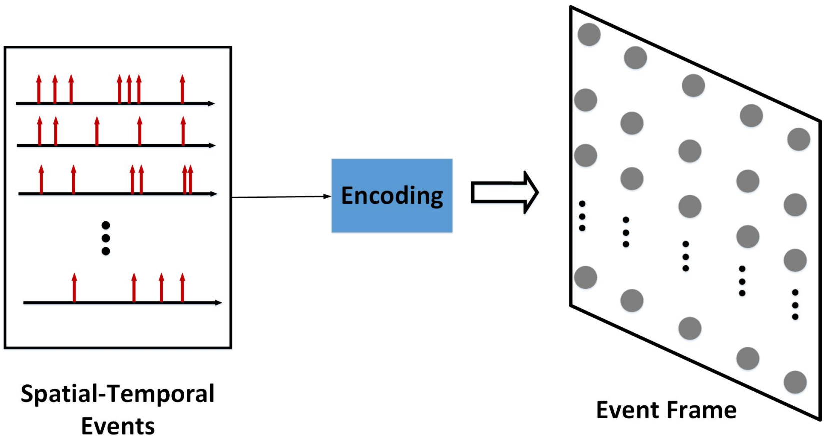


Figure 5.9: Event representation: the spatial-temporal events are processed by the encoding method to generate the event frame. Adapted from Fig. 2 in [Cao+21b] ©IEEE.

5.2.3 Method

Event Representation. The event camera is a bio-inspired vision sensor, also known as a neuromorphic vision sensor or a dynamic vision sensor, that works in ways that mimic the perception paradigm of the biological retina [Che+20a]. Conventional frame-based cameras output a series of frames at a fixed frequency. In contrast to frame-based cameras, event cameras produce data in microseconds and asynchronously, as illustrated in Fig. 5.8. An event is triggered only if the brightness change at the same pixel position exceeds a certain threshold. A sparse spatial-temporal event stream can be mathematically represented as:

$$E = \{e_i\}_{i \in [1,N]}, e_i = [x_i, y_i, t_i, p_i]^T \tag{5.12}$$

where, N represents the number of e_i contained in the event stream E. (x, y) is the coordinates of the triggered pixel position. t and p denote the corresponding triggering timestamp and event polarity, respectively. And, $p \in \{+1, -1\}$ represents the brightness change, $+1$ denotes increase and -1 denotes decrease.

In this work, the Dynamic and Active Pixel Vision Sensor (DAVIS) is used for sensing objects. DAVIS consists of a grayscale frame-based camera and an event camera, so that it can simultaneously output grayscale images and event streams. To take full advantage of the grayscale frames with texture features and event data with a high dynamic range, I combine the two data streams to improve the accuracy of vehicle detection. Since the asynchronous event stream cannot be directly processed by a convolutional neural network, I use the frequency-based [Che18; Che+19] encoding method to preprocess it into event frames before feeding it into the network. The frequency-based encoding method can be formulated as follows:

$$P(n) = 255 \cdot 2 \cdot (\frac{1}{1 + e^{-n}} - 0.5) \tag{5.13}$$

where, n represents the total number of the triggered events (*positive or negative*) at location (x, y), and $P(n)$ denotes the corresponding transformed pixel value. As presented in Fig. 5.9, the triggered spatial-temporal events are processed by a frequency-based encoding method to generate the event frame. Specifically, each pixel value in the event frame is obtained by using Eq. 5.13 to calculate the events generated within $20ms$.

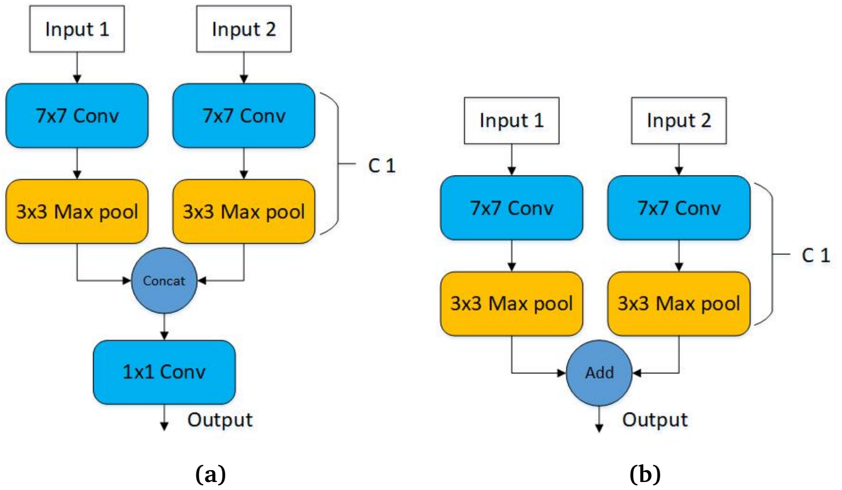


Figure 5.10: Comparison of fusion strategies between soft fusion and hard fusion. (a) Soft fusion, (b) Hard fusion. Adapted from Fig. 3 in [Cao+21b] ©IEEE.

Fusion Strategy. In this work, I explore the impact of different fusion strategies on the detection accuracy of the network. In Fig. 5.10, two fusion strategies are presented: soft fusion (Fig. 5.10a) and hard fusion (Fig. 5.10b). Instead of merging grayscale frames and events directly, I let the network learn which features need to be fused. Therefore, both the grayscale frames and the event frames are fed into the C_1 block to automatically learning to extract features, F_{gray} and F_{event}. $C1$ block consists of a convolutional filter with kernel size of 7 × 7 and a max pooling layer with kernel size of 3 × 3.

Hard fusion. For hard fusion, it denotes the element-wise sum of extracted feature maps, which can be defined as follows:

$$F_{hard} = f_{add}(F_{gray}, F_{event}) \tag{5.14}$$

where f_{add} represents the function of element-wise addition.

Soft fusion. For soft fusion, it represents feature fusion by using learned parameters. In particular, both F_{gray} and F_{event} are concatenated together, then, the convolutional filter with kernel size of 1 × 1 is applied to learn the weight parameters for feature fusion and unify dimension. The process can be expressed as follows:

$$F_{soft} = f_{concat}(F_{gray}, F_{event}) \otimes conv_{1\times1} \tag{5.15}$$

where f_{concat} and ⊗ denote concatenate and convolution operations, respectively. Different from hard fusion and soft fusion, the feature attention gate component (FAGC) combines hard fusion and attention mechanisms to fuse grayscale-based features and event-based features so as to significantly improve the vehicle detection accuracy.

Feature attention gate component (FAGC). Attention mechanism has been applied in computer vision and worked very well, such as [HSS18; Jen+20; Okt+18; Woo+18; Wan+18]. In this work, the extracted grayscale-based features F_{gray} and event-based features F_{event} are fed into the feature attention gate component (FAGC) to extract valuable features. The block diagram of the feature attention gate component (FAGC) is presented in Fig. 5.11. Both the grayscale-based features and the event-based features pass through a convolutional filter with kernel size of 3 × 3 and a ReLU activation function to get transformed contextual information. Then, the transformed features are fused by element-wise addition:

$$F_{fuse} = f_{add}((F_{gray}, F_{event}) \otimes conv_{3\times3}) \tag{5.16}$$

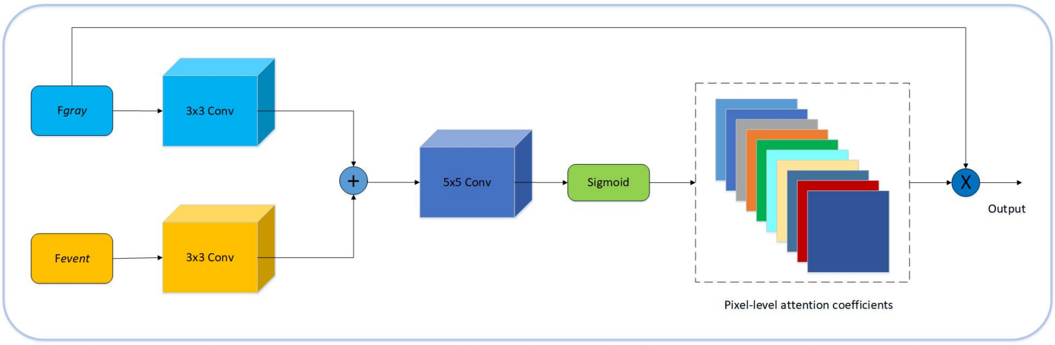


Figure 5.11: Feature attention gate component (FAGC): Both grayscale-based features and event-based features are fed into FAGC to generate the pixel-level attention coefficients. Adapted from Fig. 4 in [Cao+21b] ©IEEE.

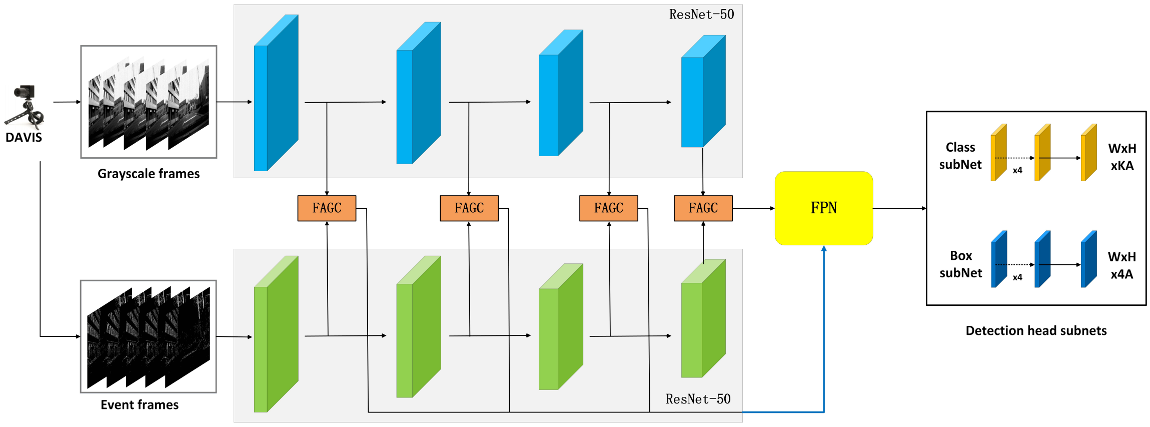


Figure 5.12: The architecture of my vehicle detection network. The network consists of ResNet [He+16], the feature attention gate component (FAGC), the feature pyramid network (FPN), and detection head subnets. Adapted from Fig. 5 in [Cao+21b] ©IEEE.

Furthermore, in order to identify salient feature regions and suppress unrelated background regions, event-based features are used as gate signals, and 5x5 convolution followed by a sigmoid activation function is used to generate pixel-level attention coefficients from the fused features. The output of the feature attention gate component (FAGC) is the element-wise multiplication of the input grayscale-based features and the pixel-level attention coefficients:

$$F_{output} = \sigma(F_{fuse} \otimes conv_{5\times5}) \cdot F_{gray} \tag{5.17}$$

Based on this mechanism, the object features will be enhanced to further improve the detection accuracy of the network. The impact of different fusion methods will be discussed in detail in section 5.2.4.

Network Architecture. The vehicle detection framework used in this work is built on the basis of [Lin+20], as shown in Fig. 5.12. The image size of 532 × 400 grayscale frames and event frames is fed into Resnet [He+16] to extract meaningful features. Both event-based features and grayscale-based features are fused by the feature attention gate component (FAGC). The fused features are collected to pass through the feature pyramid network (FPN) [Lin+17] to obtain deep features for detecting vehicles of different scales. The vehicle detection network is composed of ResNet, the feature attention gate component (FAGC), feature pyramid network (FPN), and detection head subnets. It can be formulated as follows:

$$\begin{aligned} L_c^k &= f_{Resnet}(x_{gray}, x_{event}) \\ F^k &= f_{FAGC}(L_c^k) \\ P^n &= f_{FPN}(F^k) \\ Y^n &= f_{Head}(P^n) \end{aligned} \tag{5.18}$$

where x_{gray} and x_{event} represent the grayscale frame input and event frame input, respectively. $\left\{L_c^k\right\}_{k=1,c\in[gray,event]}^4$ denotes the extracted grayscale-based features and event-based features. $\{F_k\}_{k=1}^4$ is the fused features generated by feature attention gate component (FAGC). $\{P_n\}_{n=1}^5$, and $\{Y_n\}_{n=1}^5$ denote the fused multi-resolution features and prediction outputs, respectively. And, the functions of the feature attention gate component (FAGC), Resnet, feature pyramid network (FPN), and detection head subnets are represented by f_{FAGC}, f_{Resnet}, f_{FPN}, and f_{Head}.

Resnet. In this work, I use Resnet-50 [He+16] as the backbone network. Resnet-50 is composed of four layers, represented as $\left\{L_1, L_2, L_3, L_4\right\}$, where the feature map resolution is continuously down-sampled from the L_1 to the L_4 layer and the feature resolution remains the same in each layer. The feature attention gate component (FAGC) mentioned above is inserted between two layers. By combining the residual learning and feature attention gate component (FAGC), the more strong semantic and valuable features can be extracted.

Feature pyramid network (FPN). Similar to the previous works [Lin+20; Lin+17], feature pyramid network (FPN) is used to fuse the features generated from $\{C_k\}_{k=1}^4$ to improve the detection robustness of vehicles of different sizes. The outputs $\{P_n\}_{n=1}^4$ are produced by top-down pathway and lateral connections. And, the last level feature map P_5 is produced by applying a 3×3 convolutional layer with stride 2 on the P_4. Multi-level feature maps $\{P_n\}_{n=1}^5$ will be fed into the detection head subnets for prediction.

Detection head subnets. After processing by the feature pyramid network (FPN), two separate subnets are applied for classification and box regression. Refer to [Lin+20], each subnet consists of four 3×3 convolutional layers with 256 filters. For classification subnet, followed by a 3×3 convolutional layers with KA filters, followed by sigmoid activations, it outputs KA binary predictions. For box regression subnet, followed by a 3×3 convolutional layers with $4A$ filters, it outputs $4A$ offset predictions. A is set as 9 in this work. Specific offset parameters of the bounding box can be represented as follows:

$$\begin{aligned} t_x^{'} &= \frac{(x^{'} - x_a)}{w_a}, \\ t_y^{'} &= \frac{(y^{'} - y_a)}{h_a}, \\ t_w^{'} &= log(\frac{w^{'}}{w_a}), \\ t_h^{'} &= log(\frac{h^{'}}{h_a}), \end{aligned} \tag{5.19}$$

where x, y, w, h represent the center coordinates, width, and height of the bounding box, respectively. Variables $t^{'}, x^{'}, x_a$ denote the prediction regression offsets, predicted bounding box and anchor box, respectively.

Loss function. The loss function of my vehicle detection network consists of a classification and regression loss function. The total loss function L can be represented as follows:

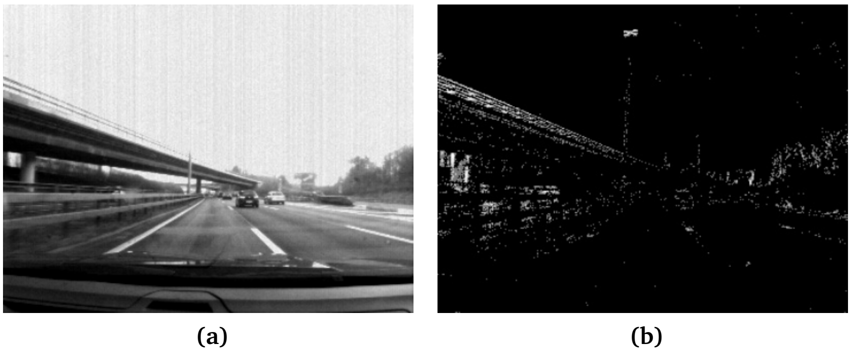


Figure 5.13: Comparison of grayscale frame and event frame. (a) grayscale frame; (b) event frame. Adapted from Fig. 6 in [Cao+21b] ©IEEE.

$$L = \frac{\lambda_1}{N}\sum_{i=1}^{N} l_{cls}(p_i, t_i) + \frac{\lambda_2}{N}\sum_{i=1}^{N} t_i^{'} \sum_{j\in\{x,y,w,h\}} l_{reg}(v_{ij}^{'}, v_{ij}) \tag{5.20}$$

where N denotes the number of anchors. Specifically, focal loss l_{cls} and giou loss l_{reg} are used in this work. The hyper-parameter λ_1 and λ_2 control the trade-off of classification and regression losses. $\lambda_1=\lambda_2=1$ are used in my experiments.

5.2.4 Experiments

The experiments of my vehicle detection network are performed on the labeled DDD17 dataset [Bin+17; Li+19a]. The results indicate that the fusion-based feature attention gate component (FAGC) can improve the detection accuracy of the vehicle detector. And I also discuss the influence of different fusion strategies and event representations on the detection performance of the network.

Dataset. In order to verify the effectiveness of my fusion method, the experiments are conducted on the DDD17 dataset. DDD17 [Bin+17] uses DAVIS to record both grayscale frames and event streams. The comparison of grayscale frame and the corresponding event frame is presented in Fig. 5.13. The dataset is collected from highway and city scenes from Switzerland to Germany. Since DDD17 is established for end-to-end learning, it does not contain the labels for object detection, while the authors of [Li+19a] manually labeled the vehicles of the dataset based on the original raw data. The detailed description is summarized in Tab. 5.2. On account of the fact that my model requires both event-based and frame-based data, and DDD17 is a challenging data set, I use the labeled DDD17 as the benchmark to compare the performance of the different fusion strategies on vehicle detection. The labeled DDD17 dataset contains 3154 frames. I used 2241 frames as the training set and 913 frames as the test set. In order to train more robust models, data augmentation methods such as flipping and color enhancement are used to increase the diversity of data samples.

Implementation Details. In the training period, I train the vehicle detection network end-to-end for 30 epochs on a Nvidia Tesla V100 GPU with 32GB memory. I define the initial learning rate as 0.01. Weight decay and momentum are set to 0.0001 and 0.9, respectively. The network is implemented using PyTorch 1.7.0 with CudNN 7.5 and CudA 10.0 packages.

Table 5.2: Detailed description of the recorded data in the labeled DDD17 dataset. Adapted from Table I in [Cao+21b] ©IEEE.

Recorded data	Condition	Length (s)	Type
1487339175	day	347	test
1487417411	day	2096	test
1487419513	day	1976	train
1487424147	day	3040	train
1487430438	day	3135	train
1487433587	night-fall	2335	train
1487593224	day	524	test
1487594667	day	2985	train
1487597945	night-fall	50	test
1487598202	day	1882	train
1487600962	day	2143	test
1487608147	night-fall	1208	train
1487609463	night-fall	101	test
1487781509	night-fall	127	test

Table 5.3: The detection results of different event representations on the labeled DDD17 dataset. Adapted from Table II in [Cao+21b] ©IEEE.

Methods	Input Modality	AP(%)	FPS
LIF	Events	13.9	14
SAE	Events	51.1	14
Frequency	Events	**52.3**	14

Quantitative Analysis. *Effect of event representation.* The vehicle detection network can take different image-like event representations as input. I compare the performance of three more representative event encoding methods, Frequency [Che18], LIF ((Leaky Integrate-and-Fire) [Bur06] and SAE (Surface of Active Events) [MBS17]. The results from the labeled DDD17 dataset are presented in Tab. 5.3. Compared with the other two encoding methods, frequency-based event representation achieves the best performance with an accuracy of 52.3%. Therefore, I use frequency as event preprocessing method in the subsequent experiments.

Impact of different fusion strategies. I explore the impact of MTC (Merged-Three-Channel) [Che+19], hard fusion, soft fusion, and FAGC. MTC is a channel-level fusion strategy. In this work, the three channels of MTC frames are consists of $[Frequency, SAE, LIF]$. The Retinanet based on resnet-50 with the grayscale frames and frequency-based event representation as input are the baselines. Specifically, I use the pre-trained weight of Resenet-50 on ImageNet to initialize the model parameters and train the vehicle detector with MTC, hard fusion, soft fusion, and FAGC, respectively, on the labeled DDD17 dataset. The experiment results are given in Tab. 5.4. As can be seen from the Tab. 5.4, the network gets 79.6% vehicle detection accuracy by taking grayscale frames as input. In order to enable the event streams to be processed by CNN, the frequency-based [Che18] encoding method is used to regularize the events into event frames. Using only event data as input, the network achieves a detection accuracy of 52.3%. Compared with a grayscale-based vehicle detector, the accuracy of an event-based

Table 5.4: The detection results of different fusion strategies on the labeled DDD17 dataset. Adapted from Table III in [Cao+21b] ©IEEE.

Methods	Input Modality	AP(%)	FPS
Baseline	Events	52.3	14
Baseline	Grayscale	79.6	14
MTC	Events	47.8	14
Hard fusion	Events & Grayscale	77.2	12
Soft fusion	Events & Grayscale	79.4	12
FAGC	Events & Grayscale	**81.6**	8

Table 5.5: The detection results of different methods on the labeled DDD17 dataset. Adapted from Table IV in [Cao+21b] ©IEEE.

Methods	Input Modality	AP(%)	FPS
FasterRCNN [Ren+15]	Grayscale	80.2	3
SSD [Liu+16]	Grayscale	73.1	12
Yolo [RF16]	Grayscale	70.2	15
Retinanet [Lin+20]	Grayscale	79.6	14
FAGC	Events & Grayscale	**81.6**	8

vehicle detector is significantly lower than that of a grayscale-based one because of the lack of appearance information such as texture in the event data. However, due to the advantages of high dynamic range and high temporal resolution of event data, it can alleviate the motion blur of grayscale frames under high illumination and high speed motion. Therefore, the feature attention gate component (FAGC) is developed to fuse event data with grayscale frames. The test results indicate that the performance of the vehicle detection network based on MTC, hard fusion, and soft fusion has basically not improved. However, based on FAGC, the network achieves the best detection accuracy of 81.6%, which outperforms the method that only takes grayscale frames or events as input.

Comparison with grayscale-based detectors. I compare the proposed model with several selected frame-based object detectors [Ren+15; Liu+16; RF16; Lin+20]. All experiments are conducted on the labeled DDD17 dataset, and the tested results are summarized in Tab. 5.5. Specifically, the results of [Ren+15; Liu+16; RF16] are referred from [Li+19a]. Compared with one-stage object detectors [Liu+16; RF16; Lin+20], the vehicle detection network with FAGC achieves a significant improvement in detection accuracy, while the running speed is reduced. Moreover, FAGC also has better performance over the two-stage object detector [Ren+15], which demonstrates the effectiveness of the proposed method.

Qualitative Analysis. The selected detection results are visualized in Fig. 5.14. The detection results for grayscale, events, and FAGC are presented in the first, second, and third rows, respectively.

Normal detections. It can be seen from Fig. 5.14 that the vehicle detection results of the first two columns demonstrate that the detection performance of the detector based on

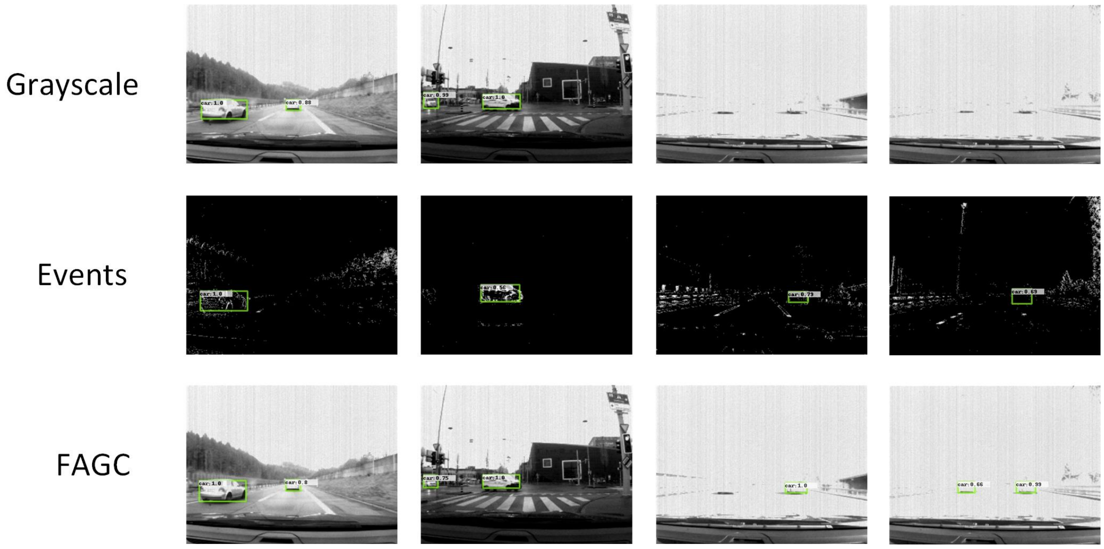


Figure 5.14: The selected detection results of the vehicle detection model on the labeled DDD17 dataset. The first, second, and third rows display the detection results of grayscale, events, and FAGC, respectively. Adapted from Fig. 7 in [Cao+21b] ©IEEE.

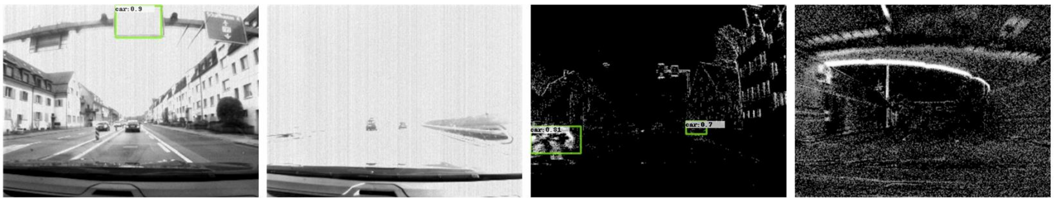


Figure 5.15: Failed detection cases: the first two cases are false detection results based on grayscale frames, and the last two cases are false detection results based on event frames. Adapted from Fig. 8 in [Cao+21b] ©IEEE.

Table 5.6: Evaluation on day and night-fall condition. Adapted from Table V in [Cao+21b] ©IEEE.

Methods	Input Modality	Day	Night-fall	All
Frequency	Events	49.4	67.1	52.3
Retinanet-Gray	Grayscale	77.9	**87**	79.6
FAGC	Events & Grayscale	**80.5**	86.2	**81.6**

grayscale is stronger than that based on event under the normal light condition. And the detection accuracy is similar between the grayscale-based vehicle detector and the fusion-based (FAGC) vehicle detector.

Overexposure detections. In Fig. 5.14, the vehicle detection results in the third column show that the grayscale-based detector is weaker than the event-based detector under high illumination conditions. Moreover, the vehicle detection results in the last column indicate that the FAGC-based fusion method can achieve better performance when the detector's detection performance is not good, either grayscale-based or event-based.

Evaluation on day and night-fall condition. In order to further explore the effectiveness of vehicle detectors, Frequency-based, grayscale-based (Retinanet-Gray) and FAGC-based vehicle detectors are tested respectively on day, night-fall, and all (day and night-fall) conditions. The test results are summarized in the Tab. 5.6. Both event-based and grayscale-based detectors achieve stable detection performance under day and night-fall conditions. And, the proposed FAGC-based detector can achieve more robust generalized ability through fusion of events and grayscale frames. The main reason for this result is that the high temporal resolution and high dynamic range of events can alleviate the challenge of grayscale frames due to overexposure, low light and high speed motion.

Failure cases analysis. Some failed detection cases are displayed in the Fig. 5.15. For grayscale-based vehicle detectors, the model incorrectly detects traffic signs as vehicles and performs poorly under high light conditions. Although the event data filtered most of the background, a large number of events were generated by some roadside obstacles in the process of perceiving the environment, leading to incorrect detection results output by the model. In addition, when passing a scene such as a bridge, a large number of events will be generated by the outline of the bridge, resulting in little information about the vehicle. Compared with traditional vision, event-based vision research is still in the preliminary stage, so further development of this technology is needed to make it mature gradually.

5.2.5 Summary

In this work, I introduce a fully convolutional neural network with a feature attention gate component (FAGC) to perform vehicle detection. Both grayscale frames and event streams are fused together to improve the detection accuracy of the network. To better fuse frame-based and event-based vision, hard fusion and soft fusion are discussed. Based on hard fusion and an attention mechanism, FAGC is developed to combine the grayscale frames with texture and events with a high dynamic range to improve the discrimination ability of the model. By integrating the FAGC into the model, the vehicle detector achieves better performance compared with the method that only takes grayscale frames or events as input. The experimental results on the labeled DDD17 dataset indicate that the proposed fusion method is effective. Compared with traditional frame-based vision, the dataset of event cameras is scarce. In the following work, I will collect a multimodal dataset to promote research on the fusion of the event signal and other modal information. Since event-based research is still in its infancy, I will try to explore the application of event cameras in more fields, such as object tracking, segmentation, etc.

5.3 Multimodal Neural Network for Robotic Grasp Detection

Grasping object detection is a crucial procedure in robotic manipulation. Most of the current robot grasp manipulation systems are built on frame-based cameras, like RGB-D cameras. However, the traditional frame-based grasping object detection methods have encountered challenges in scenarios such as low dynamic range and low power consumption. In this work, a neuromorphic vision sensor (DAVIS) is introduced to the field of robotic grasp. DAVIS is an event-based, bio-inspired vision sensor that records asynchronous streams of local pixel-level light intensity changes, called *events*. The strengths of DAVIS are that it can provide high temporal resolution, high dynamic range, low power consumption, and no motion blur. I constructed a neuromorphic vision-based robotic grasp dataset with 154 moving objects, named *NeuroGrasp*, which is the first RGB-Event multi-modality grasp dataset (to the best of my knowledge). This dataset records both RGB frames and the corresponding event streams, providing frame data with rich color and texture information and event streams with high temporal resolution and high dynamic range. Based on the *NeuroGrasp* dataset, I further develop a multi-modal neural network with a specific Euler-Region-Regression sub-network (ERRN) to perform grasping object detection. Combining frame-based and event-based vi-

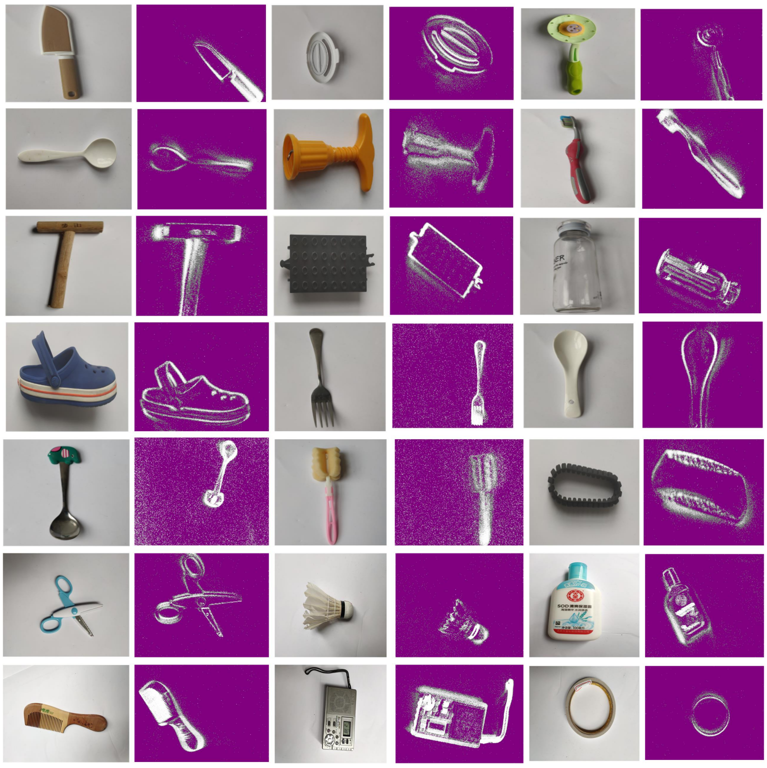


Figure 5.16: Samples from the NeuroGrasp dataset: a list of the selected RGB images and the corresponding event frames. Adapted from Fig. 1 in [Cao+22b] ©IEEE.

sion, the proposed method achieves better performance than the method that only takes RGB frames or event streams as input on the *NeuroGrasp* dataset.

5.3.1 Background

Grasping object detection plays an important role in robotic manipulation. The emergence of advanced sensors, such as *Microsoft Kinect*, has enriched robot perception systems. In recent years, deep learning-based methods have been widely applied in robotic manipulation [LLS15; Liu+21; Che+20f; Bag+20; Pay+05]. The success of deep learning has driven approaches that leverage large volumes of training data to perform complex tasks [Liu+21; Che+21a]. However, grasp datasets collected in the physical environment are relatively scarce. Dexnet [Mah+17] has explored the use of simulated data in grasping object detection to alleviate this problem. Another challenge is maintaining a balance between computational cost and the power available within embedded robot systems. Current state-of-the-art robotic grasp manipulation systems [KK17b; CXV18; Cao+21a] usually leverage frame-driven RGB-D cameras as the perception sensors. The traditional frame-driven cameras capture environmental information by generating a series of discrete frames at a fixed frequency, providing rich color and texture information. However, frame-based cameras suffer the challenges of high computing time and storage consumption [Gal+20]. In this paper, I build a dynamic sensing pipeline using a neuromorphic vision sensor, *Dynamic and Active-Pixel Vision Sensor* (DAVIS-346). DAVIS is a camera model that consists of a dynamic vision sensor (an event-based sensor) synchronized with an RGB frame-based sensor. DAVIS can synchronously record RGB data and the corresponding event streams. Specifically, it only transmits the local pixel-level changes caused by the change in lighting intensity within a scene *at the time they* *occur*, like a bio-inspired retina [Che+20a]. Concretely, the change in light intensity is very effective for detecting moving objects. Fig. 5.16 presents a comparison between conventional images and the corresponding event frames. Events are timestamped with the precision of around a microsecond. A single event is defined as the tuple $\{t, x, y, p\}$, where t is the timestamp of the event, x, y are the pixel coordinates of the event in 2D space, and $p = \pm 1$ is the polarity of the event which is the sign of the brightness change. Compared to frame-based cameras, the neuromorphic vision sensors have properties that are complementary to RGB sensors, including very high temporal resolution, high dynamic range (120 dB), and low power consumption [Che+22]. In my previous work [Li+20a], I introduced an event-based grasping dataset (E-Grasping) for robotic grasping object detection. However, the E-Grasping dataset only records event streams and contains fewer grasp objects. In this work, I use DAVIS as a perception sensor to construct a more challenging dataset in a practical environment. This dataset includes both RGB data and the corresponding event streams.

Early works on robotic grasp mainly relied on template matching to perform grasping object detection. In unstructured environments where objects vary in shape and appearance, template-matching algorithms cannot work effectively. Taking 2D images instead of the 3D model as input is more convenient for predicting grasp pose [LLS15; JMS11a]. Based on 2D images, many researchers have applied deep convolutional neural networks to robotic grasping object detection and achieved great success. In [LLS15], a sliding window detection framework is used for 2D robotic grasping object detection. Specifically, image sequences are fed into a convolutional neural network to extract features, and the candidate with the highest output confidence score of all grasp candidates is chosen as the final prediction result. The disadvantage of this method is its high computational cost. To speed up these algorithms, end-to-end methods are developed [CXV18; Zha+19b; ATH18a; Cao+22a]). Concurrently, the authors take RGB or RGB-D images as input to perform regression or classification on grasp rectangles and achieve significant improvements on the Cornell Grasping Dataset [JMS11b]. Compared to conventional frame-based grasping, neuromorphic vision-based grasping is still in its infancy.

For event-based robotic grasping object detection, it faces two main problems: a lack of data and effective algorithms. To cope with these challenges, I collected a manually labeled multi-modality (RGB-Event) robotic grasping dataset, *NeuroGrasp* dataset, and developed a multi-modal neural network to explore how to fuse the valuable feature context of RGB frames and events to improve performance. Specifically, with the use of the frequency-based encoding method [Che+19], events generated from DAVIS can be fed into convolutional neural networks for subsequent grasp pose prediction. To take advantage of DAVIS, I use convolutional filters to fuse the valuable feature context of events with RGB images to improve the prediction performance. Furthermore, an Euler-Region-Regression sub-network (ERRN) is introduced to predict the orientation of grasped objects by adding an imaginary and a real fraction to the regression network. This strategy builds a closed mathematical space to avoid singularities that may occur in single-angle estimation [Sim+18]. Experimental results show that the proposed method achieves better performance than the method that only takes a single-mode signal as input.

My main contributions can be summarized as follows:

- I collect an RGB-Event multi-modality grasp dataset named *NeuroGrasp* from a real-world experiment environment, which will promote the research on neuromorphic vision sensors for robotic grasping object detection.

- I develop a novel multi-modal neural network to fuse the valuable feature context of RGB images and events to improve performance. An Euler-Region-Regression sub-Network (ERRN) is also introduced for more accurate pose estimation.

- Extensive experiments on the *E-Grasping* and *NeuroGrasp* datasets demonstrate that the proposed method outperforms the method that only takes RGB frames or event streams as input.

5.3.2 Related Work

Datasets. At present, Cornell grasp dataset [JMS11b] and Dex-Net dataset [Mah+17] are collected for analyzing grasp quality with parallel plate gripper (PPG). The Cornell Grasp dataset recorded with an RGB-D camera consists of 885 images of 280 different objects. It is widely used by researchers and greatly contributes to the robotic grasp research field. The grasp dataset demonstrates 8019 labeled grasp rectangles, including several good grasp positions (5110) and bad grasp positions (2909) for each view of an object. The point cloud data and background image of each image are also provided. The Dex-Net dataset is collected by the UC Berkeley Automation Lab [Mah+17]. Dex-Net provides synthetic point clouds and grasp annotations based on 3D objects and has been extended to three versions: Dex-Net 1.0, Dex-Net 2.0, and Dex-Net 3.0. Dex-Net 1.0 includes over 10,000 different 3D object models and contains about 2.5 million grasp labels, and Dex-Net 2.0 is a dataset of more than 6.7 million synthetic point clouds and corresponding labels. Since Dex-Net 3.0 is built for studying suction grasp, I do not describe it in detail in this paper. Moreover, a simulated dataset, named the Jacquard Dataset [DDC18], is created from CAD models through simulation. In this dataset, more than 50k images of 11k objects are collected, and 1 million unique grasp rectangles are labeled. However, these datasets are all focused on RGB-D data. In my early work [Li+20a], I constructed an event stream-based grasping dataset (E-Grasping) using an event-based dynamic active vision sensor (DAVIS). By using an SMP filter to track LED markers, all objects are labeled automatically. The disadvantage of this dataset is that it only records the event streams for grasping objects. In this work, I build a more challenging multi-modality grasp dataset with more grasp objects.

Frame-based grasping object detection. Research on robotic grasp pose estimation has made important advances over the last 20 years. Early works [LLS15; SDN08] trained grasp detectors based on the sliding window, which is very time-consuming. In [JLD16; Dou20], the authors reduced inference time by learning their methods on a discrete set of grasp candidates. However, these approaches ignore some potential. Other methods, like [KK17b; Guo+17] used end-to-end CNN-based algorithms to regress a single grasp for an input image, but these approaches tend to estimate the average grasp pose of objects. In [CXV18], a grasp region proposal network is incorporated for grasp pose estimation based on the Faster RCNN [Ren+15]. Furthermore, the authors of [Zha+19b] proposed a single-stage real-time grasp network with the orientation anchor box mechanism, which achieves outstanding performance in both speed and accuracy. For object overlapping scenes, an ROI-based method is developed in [Zha+19a]. The experimental results showed that their algorithm can effectively deal with objects overlapping in scenes. Since the ground truths in the grasp pose are not exhaustive, [CHM19] introduced a grasp path to generate mapped grasp for convolutional multi-grasp prediction, which improved grasp accuracy in real-world scenarios. In [PSC18b], the authors presented a highly accurate and real-time grasp detection system with a rotation ensemble module (REM). Some ideas of this network design are inspired by YOLO9000 [RF16]. Another works [Wan+19a; Cao+21a; Cao+22a] deployed the neural network to generate grasps with high-resolution images. Their model solves the problem of pixel-wise robotic grasp pose estimation. Moreover, [ATH18a] and [Wu+19] used the fusion method to perform grasp prediction and achieved better performance. However, while the above methods usually take RGB or RGB-D images as input to perform regression or classification on grasp rectangles, I will explore the potential application of robotics by focusing on neuromorphic vision sensors (DAVIS 346).

Event-based grasping object detection. Recently, the development of event-based neuromorphic vision technology has provided an alternative sensing scheme for many vision fields. Some attempts have been made in the field of object detection [Che+19; Cao+21b]. For robotic grasp, a method including perception, reasoning, and control is proposed to solve the problem of picking and placing in mobile robots [Mir+18]. Based on an embedded Dynamic Vision Sensor (DVS), this method can pick up the object and move it to its correct position. In [Mut+20], the authors proposed a dynamic vision-based finger system for slip detection and suppression. This fingering system can detect objects better under illumination and vibration with a threshold algorithm. For vision-based measurement applications, a dynamic vision-based approach for tactile sensing is introduced in [Bag+20]. Furthermore, the authors of [Li+20a] constructed an event-based dataset and developed an event-based deep neural network to predict grasp pose. However, compared to the conventional frame-based vision, neuromorphic vision is still in its infancy and generally offers a lower spatial resolution.

In this work, I introduce a multi-modal neural network to perform robotic grasp pose estimation, which combines frame-based vision and event-based vision. I evaluate my model on two dynamic robotic grasping datasets, E-Grasping and NeuroGrasp. Experimental results demonstrate that my model is capable of predicting exactly grasping rectangular shapes.

5.3.3 Neuromorphic Grasping System

Neuromorphic vision sensor. A neuromorphic vision sensor (event camera) is a bio-inspired sensor that mimics the working principle of biological neurons found in the visual cortex of mammals [Che+20a]. Traditional frame-based vision cameras sense the environment by producing a series of frames that sample the light intensity at discrete time intervals. Neuromorphic vision sensors record asynchronous event streams of the change in light intensity of a given pixel. It allows the sensor to measure the per-pixel changes caused by motion in a scene at the time of occurrence. A stream of sparse spatial-temporal events can be represented by $e_i(x_i, y_i, t_i, p_i)_{i\in[1,N]}$, which means that an event is triggered at pixel location $l_i = (x_i, y_i)$ when the intensity change at a pixel occurs, i.e,

$$\Delta L(l_i, t_i) = L(l_i, t_i) - L(l_i, t_i - \Delta t_i) \tag{5.21}$$

where $L(\cdot)$ is the brightness log function and Δt_i is the time interval between the current event and the last event at the same pixel. Specifically, the temporal contrast threshold $\pm T$ $(T > 0)$ is set for the intensity change to be reached,

$$\Delta L(l_i, t_i) = p_i T \tag{5.22}$$

where $p \in \{+1, -1\}$ is the polarity of event, that represents the brightness change. $p = +1$ denotes the increase in brightness intensity and -1 denotes the decrease. As an emerging bio-inspired vision sensor, event-based neuromorphic vision sensors have several promising properties: low energy consumption, low latency, high dynamic range, and high temporal resolution. In this work, I will explore the potential of neuromorphic vision sensors in the field of robotic grasp pose estimation.

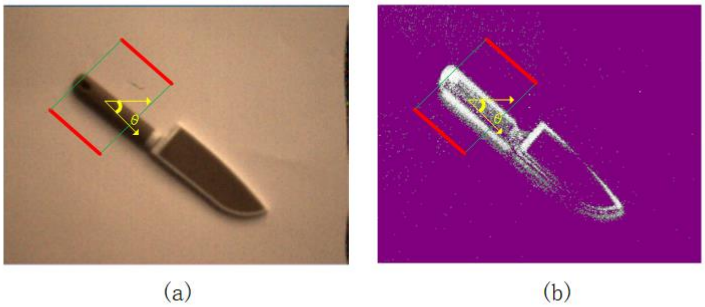


Figure 5.17: The 5-D grasp configuration. (a) Grasp configuration is presented in RGB images; (b) Grasp configuration is presented in the corresponding event frames. Adapted from Fig. 3 in [Cao+22b] ©IEEE.

System setting. A neuromorphic vision sensor (DAVIS 346) collects the event data through the lighting intensity changing, so the object needs to maintain movement within the field of view. The DAVIS 346 sensor is attached to the gripper of a robot arm (hand-eye system) to simulate the real trajectory during grasping. The parallel plate gripper (PPG) is widely mounted on the end of the robot arm, and my grasping dataset is built following the Cornell grasping dataset [JMS11b]. At first, I only consider flat objects as grasping objects. Moreover, most grasping objects can be considered flat objects when they are placed on the table in the proper direction. Compared with building a 3D grasping point cloud, this approach can reduce the cost of storage and calculation. The grasping information of flat objects with a PPG can be demonstrated as a rectangle. The width of the rectangles represents the distance between gripper plates, the height represents the range of compatible grasping, and the center is placed at a particular point on the table, which represents the grasping point. In addition, the rectangle must be rotated to a particular angle to increase the capability of grasping. This rectangle only provides the pose of PPG when it contacts the table and tries to grasp the object.

Problem definition. Given RGB images and event streams of different objects, the grasp pose estimation algorithm needs to learn how to find a successful grasp configuration G for each object. As described in [LLS15], a five-dimensional grasp representation can be mapped into the seven-dimensional configuration for robotic grasp execution on a real scene. In this work, I take RGB images and events generated by a neuromorphic vision sensor (DAVIS 346) as input to predict the five-dimensional grasp configuration of a robot with a parallel-plate gripper. As shown in Fig. 5.17, the grasp pose can be formulated as follows:

$$G = \{x, y, w, h, \theta\}^T \tag{5.23}$$

where (x, y) is the grasp rectangle's central coordination, w is the maximum distance between parallel plates, h is the height of the robot's parallel plates, and $theta$ is the grasp rectangle's angle with respect to the horizontal axis.

5.3.4 Robotic Grasp Detection

In this part, I present a multi-modal neural network architecture for grasping object detection. The overall framework of this method is shown in Fig. 5.18. The method consists of two branches: one branch extracts feature representations from RGB images, and another focuses

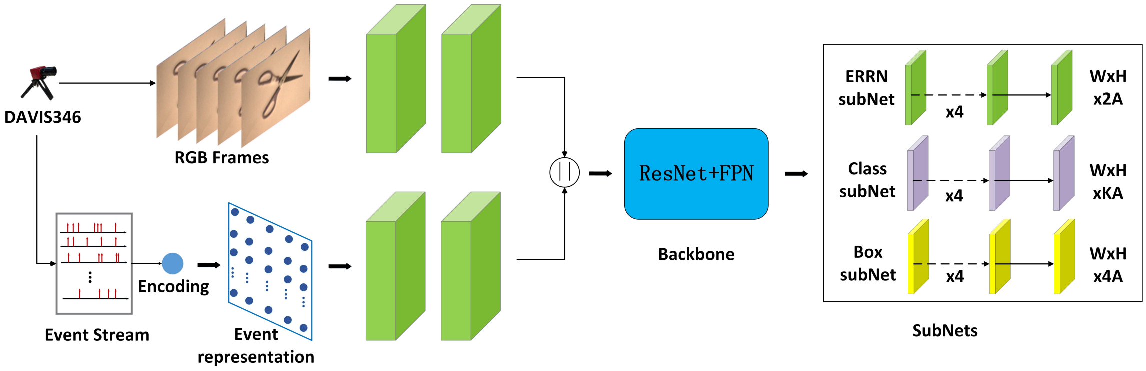


Figure 5.18: Multi-modal neural network architecture. The network includes two branches: one that extracts feature representations from RGB images, and another that focuses on extracting feature representations from event streams. The extracted features are fused and fed into the architecture formed by the Feature Pyramid Network (FPN) on top of a feedforward ResNet to generate multi-scale features. The outputs of the network are composed of the orientation angle, object classification, and corresponding grasp poses. Adapted from Fig. 4 in [Cao+22b] ©IEEE.

on extracting feature representations from event streams. The extracted features are fused and fed into the subsequent network. Furthermore, three task-specific subnetworks are added to perform grasp angle estimation, object classification, and bounding box regression on the feature outputs, respectively. I will describe the details of each component of the grasping network.

Event representation. Event streams generated by the neuromorphic vision sensor are sparse and asynchronous, which cannot be processed by the traditional computer vision method, such as CNN-based algorithms [Che+20a]. Therefore, I use a frequency-based encoding method to pre-process event sequences into an output matrix for the CNN to extract deep features.

Given that many more events would occur near an object's edges because the edges of the moving object tend to be the edges of the illumination in the image, I utilize the event frequency as the spike coding to strengthen the profile of the object. At the same time, noise caused by the sensor can be significantly filtered out due to its low occurrence frequency at a particular pixel within a given time interval [Che+19]. Concretely, I count the spike occurrence at each pixel (x, y), based on this, I calculate the spike coding value using the following activation function:

$$\sigma(n) = 255 \cdot 2 \cdot (\frac{1}{1+e^{-n}} - 0.5) \tag{5.24}$$

where n represents the total number of spikes (positive or negative) that occurred at pixel (x, y) within a given interval and $sigma(n)$ represents the spike coding value of this pixel in the event sequences.

Multi-modal fusion. After the event streams are processed by the frequency-based encoding method, I use a 7x7 convolution layer and a 3x3 max-pool layer to transfer the matrix of event sequences into unified scale feature maps. As shown in Fig 5.18, the features from event-based and RGB-based networks are fused to generate new feature maps. Since neuromorphic vision sensors can capture dynamic features of the object with high time resolution, combining event streams and RGB data can enhance the spatial-temporal context around

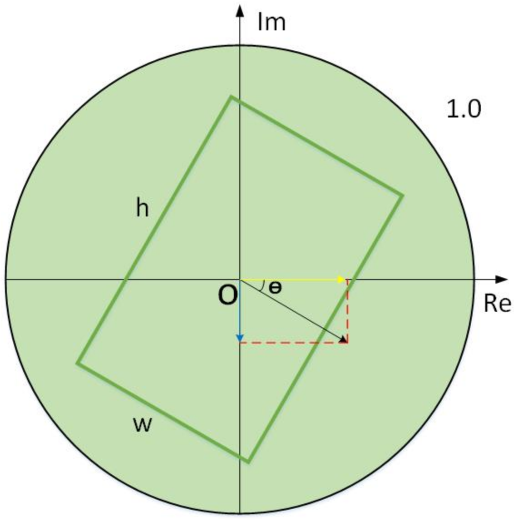


Figure 5.19: Grasp orientation angle regression. The complex angle represented by an imaginary and real fraction is used to predict the oriented grasp pose. Adapted from Fig. 5 in [Cao+22b] ©IEEE.

grasped objects, thus improving the detection performance. After that, I use ResNet as a feature extractor to learn deep feature representation for grasp pose estimation. The basic building block of ResNet is the residual block, which is designed to incorporate a skip connection with conventional CNN. Referring to [Lin+20], the "feature pyramid network" (FPN) is also utilized to get multi-scale features with a top-down pathway and lateral connections. I build a pyramid with $C = 256$ channels on each level from P_3 to P_7. To obtain a proper grasp pose, a 2-vector of grasp angle regression targets, a length K one-hot vector of classification targets, where K denotes the number of grasp object classes, and a 4-vector of box regression targets are assigned to each anchor. An imaginary and real fraction are directly embedded into the network to estimate the grasp angle. The grasping rectangle with an additional complex angle $arg(|r|e^{i\theta})$ is defined as follows:

$$\begin{aligned} t_x &= \frac{(x_g - x_a)}{w_a}, \\ t_y &= \frac{(y_g - y_a)}{h_a}, \\ t_w &= log(\frac{w_g}{w_a}), \\ t_h &= log(\frac{h_g}{h_a}), \\ t_\theta &= arg(|r|e^{i\theta}) = arctan_2(t_{Im}, t_{Re}) \end{aligned} \tag{5.25}$$

where (x, y) are the center coordinates of the grasping rectangle. The width and height are denoted by w and h, respectively. The orientation angle $theta$ is represented by an imaginary parameter $t_I m$ and a real fraction parameter $t_R e$. Variables x_g, x_a, and t_x are for the ground-truth box, anchor box, and regression offsets between the anchor box and the ground-truth box, respectively.

Euler-region-regression subNet. The Euler-region-regression subnet is responsible for predicting the orientation angle of each grasping object. A fully convolutional network (FCN) is applied to each feature pyramid level. FCN is made up of four 3x3 convolution layers with 256 filters each, followed by a 3x3 convolution layer with $2A$ filters ($A = 9$).Using $arctan_2(t_Im, t_Re)$, the orientation angle can be calculated from the regression parameters t_Im and t_Re. As shown in Fig. 5.19, instead of directly predicting the angle θ, I estimate the grasp pose of the object by adding an imaginary and a real fraction to the Euler-region-regression subnet. This strategy builds a closed mathematical space, resulting in a better generalization ability for the model.

Class and box subNet. In parallel with the Euler-region-regression subnet, two small FCNs are attached to each pyramid level for classification and bounding box regression, respectively. The structure of the two subnets is identical to that of the Euler-region-regression subnet, except that the classification subnet predicts KA and the box regression subnet predicts $4A$. The probability of grasping objects for each of the A anchors and K object classes is inferred in the classification subnet by finally passing sigmoid activations. Furthermore, the box regression subnet produces 4 outputs to regress the offsets between the anchor and the ground truth box.

Loss function. The multi-task loss function of my grasp pose estimation network is defined as follows:

$$L = L_{cls} + L_{reg} + L_{euler} \tag{5.26}$$

The loss function L consists of three parts, in which L_{cls} represents classification loss, L_{reg} denotes box regression loss, and L_{euler} is the Euler region regression loss. I refer to the design of the optimization loss functions L_{cls} and L_{reg} in [Lin+20] to improve the network's robustness. Furthermore, I extend the concepts of L_{reg} by including an Euler region regression part L_{euler} to make use of closed complex number space. The specific formulations are as follows:

$$\begin{aligned} L_{cls} &= \frac{\lambda_1}{N}\sum_{i=1}^{N} l_{cls}(p_i, t_i) \\ L_{reg} &= \frac{\lambda_2}{N}\sum_{i=1}^{N} t_i^{'} \sum_{j\in\{x,y,w,h\}} l_{reg}(v_{ij}^{'}, v_{ij}) \\ L_{euler} &= \frac{\lambda_3}{N}\sum_{i=1}^{N} t_i^{'} \sum_{k\in\{Im,Re\}} l_{reg}(\theta_{ik}^{'}, \theta_{ik}) \end{aligned} \tag{5.27}$$

Where l_{cls} is the focal loss, l_{reg} represents smooth L_1 loss. In addition, N is the number of anchors, p_i is computed by the sigmoid function to represent the probability distribution of various classes, and t_i is the corresponding label of the category. $v_{ij}^{'}$ and v_{ij} denote the predicted offset vector and the corresponding vector of ground-truth, respectively. For the Euler region regression loss, I assume that the difference between the predicted complex number and ground truth is always located on the unit circle with $|r| = 1$. Specifically, the orientation angle θ is regressed by the form of an imaginary Im and real fraction Re. The λ_1, λ_2, and λ_3 are hyper-parameters for controlling the trade-off of different losses.

5.3.5 Dynamic Robotic Grasping Dataset

For robotic grasping object detection, the number of available grasping datasets is limited. The most famous common RGB-D grasping datasets are Cornell, Dexnet, and Jacquard, which are used to compare state-of-the-art algorithms. To facilitate the application of event-based

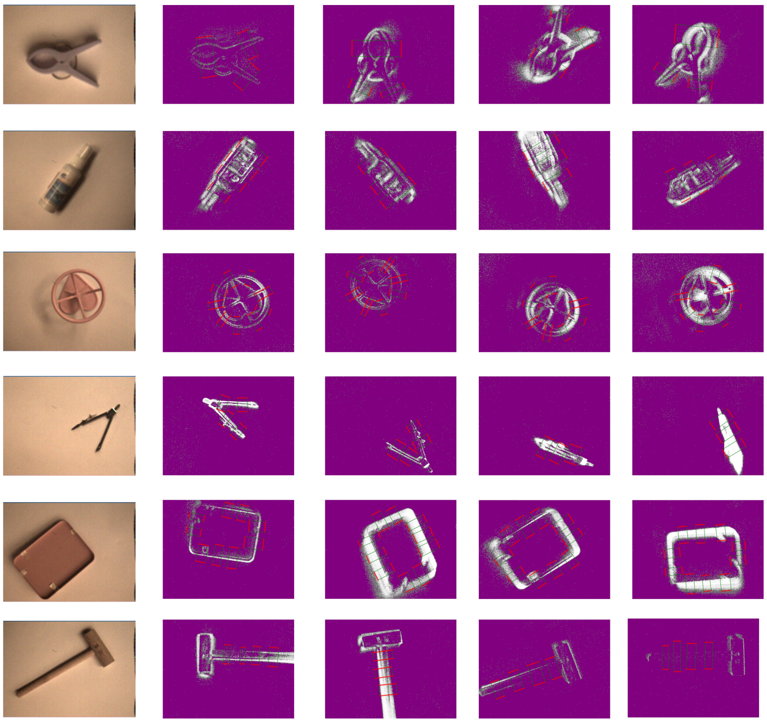


Figure 5.20: Grasp annotations: six grasping objects with different poses and views are selected for display. The first column contains RGB images, and the remaining columns contain labeled "grasping object" event data in various poses. Adapted from Fig. 6 in [Cao+22b] ©IEEE.

neuromorphic vision sensors in robotics, an automatically annotated event-based grasping dataset (*E-Grasping*) is proposed in my previous work [Li+20a]. However, compared with traditional vision, event-based research is still in its infancy. In this work, I present a manually labeled dynamic robotic grasping dataset named *NeuroGrasp*. Compared with the *E-Grasping* dataset proposed in [Li+20a], *NeuroGrasp* is the first event-based multi-modality dataset for grasp pose estimation. The dataset can be found at https://github.com/HuCaoFighting/DVS-GraspingDataSet.

Dataset recording. The dataset is collected using a neuromorphic vision sensor (DAVIS 346) with a 346×260-pixel resolution. The DAVIS 346, also known as a dynamic vision sensor or event-based camera, is a camera model consisting of a dynamic vision sensor synchronized with an RGB frame-based sensor. I use DAVIS346 to capture 154 grasp objects by recording event-based and RGB frame-based streams separately. The entire dataset is about 4620.42 s in length and contains 14141.7M events, making the dataset more diverse and challenging.

Dataset annotation. After manual filtering of unusable data, the *NeuroGrasp* dataset contains 8753 RGB images and corresponding event streams of 154 different objects with various scales, orientations, and locations. Each image is manually labeled with multiple ground truth grasp rectangles corresponding to possible grasp configurations, as shown in Fig. 5.20. However, the annotations are comprehensive and representative examples of good grasp candidates and do not cover all potential grasps. The rating score is affected by the density of each object's label. The standard file format in my benchmark is presented in Table 5.7. The dataset contains original binary data, raw event data, RGB images, timestamp files for each frame of RGB images, and labels. I also built a multi-object grasping dataset for testing the

Table 5.7: The introduction of a standard file format in my *NeuroGrasp* dataset. Adapted from Table I in [Cao+22b] ©IEEE.

File name	Description	Format
original data (.aedat)	original data	raw binary data
events (.txt)	One event per line	(timestamp, x, y, p)
RGB images (.png)	RGB frame-based data	PNG images
timestamp file for RGB frames (.txt)	One timecode per line	(frameNumber, timestamp)
labels (.txt)	One ground-truth measurement per line	(x1, y1, x2, y2, x3, y3, x4, y4)

Table 5.8: Summary of the public grasping datasets. Adapted from Table II in [Cao+22b] ©IEEE.

Dataset	**Modality**	**Objects**	**Images**
Cornell	RGB-D	240	885
Dexnet	Depth	1500	6.7M
Jacquard	RGB-D	11K	54K
E-Grasping	Event Stream	91	18.2k
NeuroGrasp	RGB+Event Stream	154	8753

generalization ability of my algorithm on a more realistic and cluttered scene. In a multi-object grasping dataset, a single image has 3–5 different objects with various orientations or poses.

Dataset analysis. In Table 5.8, I summarize the public datasets and my *NeuroGrasp* dataset. The most common grasping dataset is Cornell, which is collected in a real-world environment. The DexNet and Jacquard datasets are larger than Cornell's. However, both the DexNet and Jacquard datasets are generated by simulation, so large amounts of synthetic data and labels can be produced. The *E-Grasping* dataset is my previous work [Li+20a], which is labeled by tracking led markers. Since the size of the *E-Grasping* dataset is small, I extend the version of the event-based grasping dataset named *NeuroGrasp*, which is comprised of 8753 images with a resolution of 346×260 pixels of 154 different novel real objects. In the *NeuroGrasp* dataset, both RGB images and corresponding event streams are recorded, which is conducive to facilitating event-based robotic grasping research.

5.3.6 Experiments and Analysis

I present experimental results of the proposed multi-modal neural network on the *E-Grasping* dataset [Li+20a] and *NeuroGrasp* dataset.

Implementation details. In my experiment setup, the DAVIS 346 is attached to the end of the robot arm to ensure relative motion between the grasping object and the sensing sensor. The motion speed is controlled under $10-50mm/s$. The experimental dataset is randomly divided into training data and test data in a ratio of 8:2. In the training period, I train the

Table 5.9: The accuracy (%) of different methods on *E-Grasping* dataset proposed in [Li+20a]. Adapted from Table III in [Cao+22b] ©IEEE.

Method	Light Condition	Input	Accuracy(%)
[Li+20a]	Light Dark	Event Streams	97.8 96.2
This work	Light Dark	Event Streams	**98.9** **96.7**

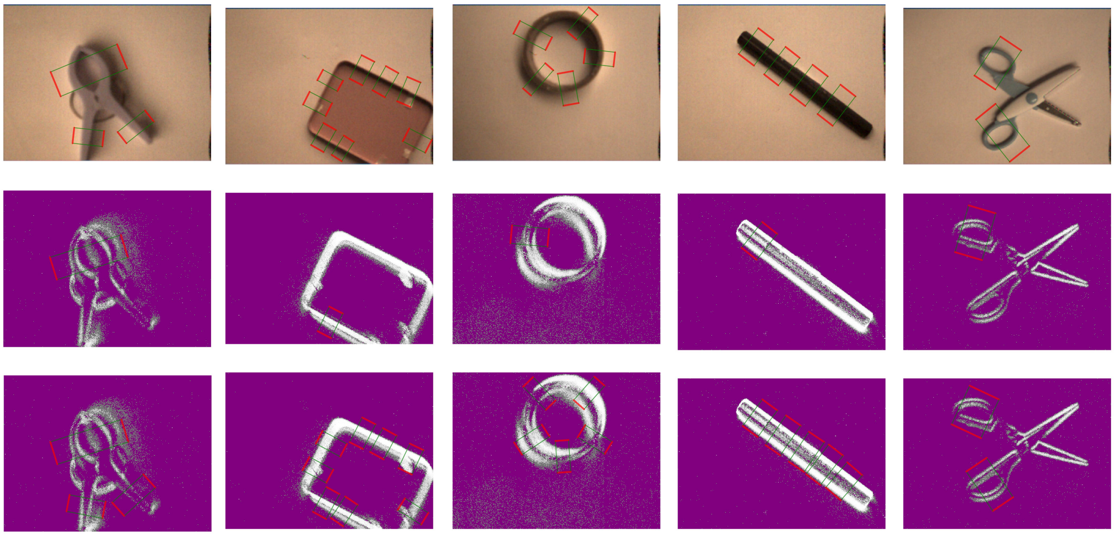


Figure 5.21: The prediction results of the proposed grasping network. The first row is the ground truth. The second row is the top-1 grasp output for several objects. The third row is the multi-grasp result (best viewed in color). Adapted from Fig. 7 in [Cao+22b] ©IEEE.

grasping network end-to-end for 30 epochs on two Nvidia GTX 2080 Ti GPUs with 22GB memory. I define the initial learning rate as 0.0005. Weight decay and momentum are set to 0.0001 and 0.9, respectively. The network is implemented using Tensorflow with Cudnn-7.5 and Cuda-10.0 packages.

Results. I explore the performance of the proposed multi-modal neural network with different input data and analyze the experimental results of different grasping object detection algorithms. The grasping performance are summarized in Table 5.9 and Table 5.10.

Experimental results on E-Grasping dataset. To facilitate comparison with [Li+20a], I train my model with the event streams as an input on the *E-Grasping* dataset. Compared with [Li+20a], the proposed grasping object detection method achieves better performance with an accuracy of 98.9%. For different lighting conditions, the proposed model can adapt well to the changes in brightness. Furthermore, both the proposed model and [Li+20a] have better performance in brighter conditions.

Experimental results on NeuroGrasp dataset. I compare my model with event-based method [Li+20a] and frame-based method [CXV18] on the *NeuroGrasp* dataset. Since DAVIS346 can simultaneously output two separate event streams and RGB images, I developed a multi-

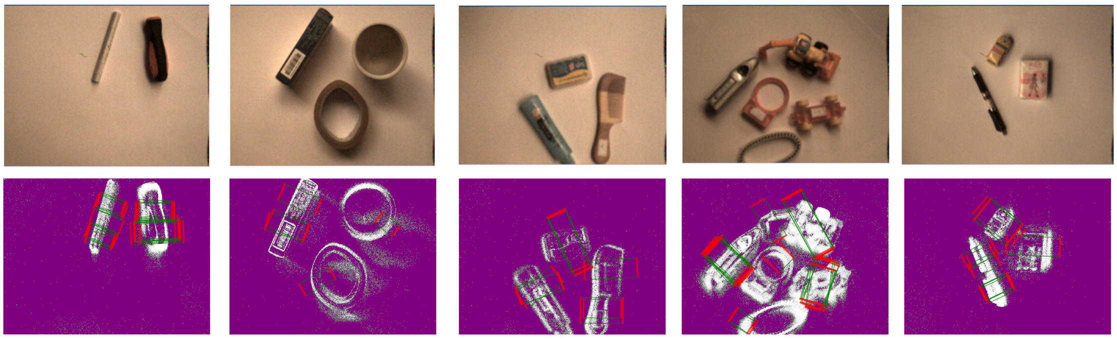


Figure 5.22: The prediction results in multiple grasping objects. The first row is the RGB images. The second row shows the grasp outputs of corresponding event streams for several objects (best viewed in color). Adapted from Fig. 8 in [Cao+22b] ©IEEE.

Table 5.10: The accuracy (%) of different methods on *NeuroGrasp* dataset. Adapted from Table IV in [Cao+22b] ©IEEE.

Method	Input	Backbone	Accuracy(%)
[Li+20a]	Event Streams	Vgg-16	41.2
[CXV18]	RGB Frames	ResNet-50	52.7
This work	Event Streams	ResNet-50	53.2
	RGB Frames		76.5
	Event + RGB		**80.6**

modal neural network to fuse the valuable feature context of event streams and RGB images. Experimental results demonstrate that the proposed multi-modal method has a better generalization ability and achieves the best performance with an accuracy of 80.6%.

In Fig. 5.21, the grasping object detection results are presented. The ground-truth grasping rectangles are in the first row, the top-1 prediction results are visualized in the second row, and the multi-grasp results are depicted in the third row. In the multi-grasp case, my grasping detection model can predict grasping poses from the features of different objects. The predicted results of these objects demonstrate that my grasping object detection method can predict grasp configuration effectively.

Single-modal vs multi-modal. In Table 5.10, grasp prediction results with different input data are presented. For each input, I use ResNet-50 as the backbone to explore the impact of input modality on algorithm performance. Due to the lack of rich appearance features such as color and texture, the grasping detection accuracy based on event streams is lower than that of RGB frames. However, event streams can provide valuable information with high temporal resolution and high dynamic range, which are complementary to RGB signals. In this work, I use a convolutional neural network to learn to fuse information from RGB frames and event streams. By combining the RGB frames and event streams, the prediction accuracy is improved by about 4%. The proposed fusion method outperforms the method that only takes RGB frames or event streams as input. In order to validate the generalization ability of my method, the model trained on the *NeuroGrasp* dataset is used to test in multi-grasp and multi-object environments. The prediction results are presented in Fig. 5.21 and Fig. 5.22. The model is trained on a single object dataset, but can still predict the grasp pose of multiple objects and multi-grasp with various orientations. The results demonstrate the

Table 5.11: The accuracy (%) of different backbones on *NeuroGrasp* dataset. Adapted from Table V in [Cao+22b] ©IEEE.

Method	Input	Backbone	Accuracy(%)
This work	Event Streams	ResNet-50	53.2
		ResNet-101	54.8
	RGB Frames	ResNet-50	76.5
		ResNet-101	83.0
	Event + RGB	ResNet-50	80.6
		ResNet-101	**83.8**

Table 5.12: The network parameter comparison of different methods. Adapted from Table VI in [Cao+22b] ©IEEE.

Model	Parameter size (Approx.)	Accuracy(%)	Speed(fps)
Single Input	113.95 million	76.5	15
Fusion Input	113.98 million	**80.6**	13

excellent generalization ability and robustness of the proposed method.

Effect of dataset. I train my grasping object detection algorithm on both the *E-Grasping* dataset and *NeuroGrasp* dataset. Because the annotation method and quantity of label data in the two datasets are different, this will affect the prediction accuracy. For the *E-Grasping* dataset, the same method achieves a higher precision on *E-Grasping* dataset than on the *NeuroGrasp* dataset as the size of the labeled ground-truth box is larger and the number of grasping objects is fewer. *NeuroGrasp* dataset is more challenging.

Effect of model scale. In Table 5.11, I discuss the effect of network deepening on model performance. It can be seen from the Table 5.11 that the performance of the model combined with ResNet-101 is better than that combined with ResNet-50. Furthermore, the proposed fusion method improves the prediction accuracy by about 4% on ResNet-50 but less on ResNet-101. The reasons for this issue can be summarized as follows: (1) Since the method used in this paper is early fusion (feature-level fusion at the early layers of the network), the early fused features become more abstracted as the network deepens, thus leading to less effective results. (2) The high detection accuracy achieved by the grasping model on ResNet-101 makes it difficult to further improve the performance. However, while the performance of the large model (ResNet-101) is higher, the model complexity is also higher. Therefore, the performance improvement of multi-modal fusion based on ResNet-50 is more promising for application.

Complexity analysis. The comparison of the network parameters between the method with single-modal input and the proposed algorithm is listed in Table 5.12. With the addition of 0.03M parameters, the proposed fusion method improves the prediction accuracy by about 4% and achieves a running speed of 13 fps. My fusion method has a good balance between accuracy and speed.

Ablation study. I provide an ablation study to discuss the impact of the Euler-region-regression subnet (ERRN), objects in clutter, and failure case analysis. All the results are based on the ResNet-50 backbone and trained on the *NeuroGrasp* dataset.

Table 5.13: The impact of ERRN subNet on the performance (%) on *NeuroGrasp* dataset. Adapted from Table VII in [Cao+22b] ©IEEE.

Input	Model	Accuracy(%)
Event Streams	Without ERRN	52.9
	With ERRN	53.2
RGB Frames	Without ERRN	73.3
	With ERRN	76.5
Event Streams + RGB Frames	Without ERRN	78.2
	With ERRN	**80.6**

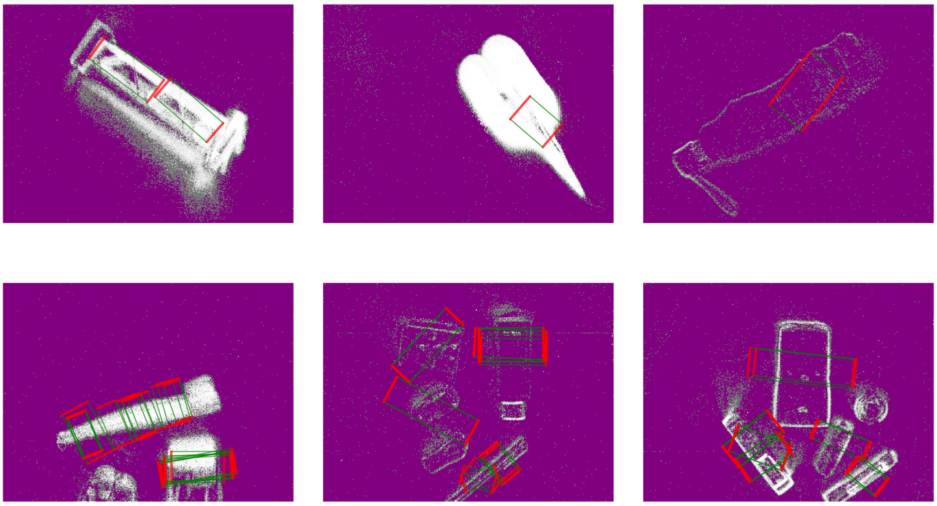


Figure 5.23: Failed detection cases. The first row is detection failure cases of a single grasped object; the second row is detection failure cases of objects in clutter (best viewed in color). Adapted from Fig. 9 in [Cao+22b] ©IEEE.

Effect of Euler-region-regression subNet. To explore the effect of the Euler-region-regression subnet (ERRN) for grasp pose learning, I use ResNet-50 as the backbone to train my model with and without ERRN on the *NeuroGrasp* dataset. The performances are presented in Table. 5.13. Experimental results illustrate that the prediction accuracy can be improved by about 3% in the best case (RGB input), which demonstrates the effectiveness of the proposed ERRN subnetwork.

Objects in clutter. For validating the generalization ability of my method, I use a ResNet-50-based model to test on a more realistic and cluttered scene, where a single view has 2–5 different objects with various orientations or poses. The test results are shown in Fig. 5.22. In complex scenarios, the proposed method can predict the grasp pose of multiple objects simultaneously and has a good generalization ability.

Failure cases analysis. Some failed prediction cases are selected to be shown in Fig. 5.23. It can be seen that the shadow of the grasping object also produces events and affects the prediction results. Some grasping objects with dense events may cause the model to fail to recognize their contour shapes, which leads to the failure of grasping prediction. At the same time, objects that fail to generate enough events can also not be predicted very well.

Discussion. Compared to traditional frame-based cameras, event-based neuromorphic vision sensors have several advantages.

Energy-friendly & *Low latency.* Event-based neuromorphic vision sensors consume less energy and have a lower latency because they only process triggered events and do not require global exposure of the frame. Such properties make it more suitable for real-time applications.

High temporal resolution. For event-based neuromorphic vision sensors, changes can be captured and time-stamped to the microsecond. This property meets the fast response requirements of the controller in robotics.

High dynamic range (HDR). The event-based neuromorphic vision sensors have an HDR $(120dB)$, which outperforms the frame-based cameras $(60dB)$. Under a light-changing scene, event-based sensors would perform better.

Capturing grasping object's edges. The event-based neuromorphic vision sensor can filter out redundant information and capture the grasping object's shapes and edges. The object's shapes and edges are beneficial for grasping and are complementary to frame-based sensors.

5.3.7 Summary

In this paper, I construct a dynamic robotic grasping dataset named *NeuroGrasp*. To the best of my knowledge, it is the first event-based multi-modality robotic grasping dataset. Based on this dataset, I introduce a multi-modal deep neural network for grasping object detection with a combination of frame-based vision and event-based vision. Furthermore, an Euler-Region-Regression sub-network (ERRN) is proposed to obtain more accurate orientation angle estimation. The proposed multi-modal method is evaluated on the *E-Grasping* and *NeuroGrasp* datasets. The experimental results indicate that the proposed method has better performance and generalization ability. I demonstrate that a neuromorphic sensor will improve both the versatility and the precision of robotic grasping object detection.

6

Conclusions and Future Directions

6.1 Conclusions

This thesis focuses on deep representation learning with attention mechanisms for object perception, including object detection and grasp detection. Concretely, channel attention, spatial attention, and channel & spatial attention are used to improve the representation capability of convolutional neural networks. The proposed methods have achieved excellent performance in applications such as people detection, vehicle detection, and robotic grasp detection.

In summary, the following tasks are explored:

Chapter 3. An orientation-aware people detection and counting method based on overhead fisheye cameras is developed. Specifically, the channel and spatial attention are used in this task. A simultaneous attention refinement module (SARM) is introduced to suppress the noise feature and highlight the object feature to improve the context-focusing ability on people in different poses and orientations. Following the collection of detection results, an Internet of Things (IoT) system based on Real Time Streaming Protocol (RTSP) is constructed to output results to different devices. Experiments on three common fisheye image datasets show that under low light conditions, the proposed method has high generalization ability and outperforms the state-of-the-art methods.

Chapter 4. An efficient grasp detection network with n-channel images as inputs is proposed for robotic grasp. The proposed network is a simple generative structure for grasp detection. In particular, a Gaussian kernel-based grasp representation is introduced to encode the training samples, embodying the maximum center that possesses the highest grasp confidence. A receptive field block (RFB) is plugged into the bottleneck to improve the model's feature discriminability. Furthermore, pixel-based and channel-based attention mechanisms are used to construct a multi-dimensional attention fusion network (MDAFN) to fuse valuable semantic information. The proposed method is evaluated on the Cornell, Jacquard, and extended OCID grasp datasets. The experimental results show that the proposed method achieves excellent balancing accuracy and running speed. The network gets a running speed of $6ms$, achieving better performance on the Cornell, Jacquard, and extended OCID grasp datasets with 97.8%, 95.6%, and 76.4% accuracy, respectively. Subsequently, an excellent grasp success rate in a physical environment is obtained using the UR5 robot arm.

Chapter 5. Event camera is a bi-inspired vision sensor that captures dynamic changes and filters out redundant information. Compared with standard cameras, event cameras provide event streams with high temporal resolution, high dynamic range, and low power consumption. However, standard cameras output frames with color and texture information. Therefore, event-based data and frame-based data are complementary. In this chapter, the working principle of event cameras is first introduced. Then, multimodal learning methods are developed for vehicle detection and robotic grasp detection. Experimental results show that the proposed methods achieve excellent performance over current methods.

6.2 Future Directions

Challenges and future directions closely related to object perception are pointed out in numerous opportunities, as described below.

Sensor fusion for object perception. Different kinds of sensors are complementary. For example, the event camera contains no color information, which is provided by a frame-based camera. The distance and speed information can be provided by LiDAR and radar. It remains to be seen whether the event camera output can be used to trigger frame captures from other sensors. If it is, the event camera and other sensors can operate together with a mix of conventional machine vision, bio-inspired, and event-based neuromorphic vision-based approaches. Therefore, some of the limitations of a traditional sensor-based perception system may be overcome.

Active vsion system for object perception. In robotics, the ability to directly fuse perception with its motoric ability is often referred to as "active perception". It is found that perception and action are often kept in separate spaces; this is a consequence of the state-of-the-art sensors equipped with the robotics being frame-based. The sensing and perception only exist in a discrete moment, while the motion is a continuous entity. A new method of encoding perceptions and actions could be meaningful to the active perception system of robotics. Moreover, this would create new opportunities for real-time navigation and obstacle avoidance if the visual perception could be bound with the system's dynamic to enable dynamic environment perception.

Large-scale benchmark for object perception. It is well known that standardized benchmarks promote the rapid development of deep representation learning. For example, the growing popularity of deep neural networks in intelligent vehicles and large-scale benchmarks such as KITTI, Cityscale, and ImageNet is interconnected and mutually reinforcing. There is an emerging need for high-quality benchmarks in the field of object perception.

Model robustness for object perception. Model robustness is crucial for object-perception systems. Occlusions, which occur when objects are partially hidden from view, pose a significant hurdle for current systems. Investigating innovative approaches to improve the detection and recognition of partially obscured objects can substantially enhance the practicality of these systems in everyday situations. Furthermore, variations in lighting conditions remain a persistent issue, affecting the accuracy of object perception. Researchers should explore techniques that adaptively adjust to diverse lighting scenarios, ensuring consistent performance across different environments. Additionally, the threat of adversarial attacks underscores the need for developing defenses that fortify object perception systems against intentional manipulations designed to mislead or compromise their functionality. Striking a balance between system complexity and computational efficiency is pivotal to implementing these advancements in real-world applications, ensuring the reliability and effectiveness of object perception systems across a spectrum of challenging scenarios.

Foundation models for open-world object perception. Open vocabulary object perception, dealing with a wide variety of objects without predefined categories, requires a combination of foundational models and vision-language models. Foundational models, like convolutional neural networks (CNNs), need to learn adaptable features and hierarchical representations for diverse objects. Vision-language models, such as transformer-based architectures, aid in understanding and describing objects by generating semantic embeddings and incorporating contextual reasoning. Systems should support dynamic adaptation through incremental learning and transfer learning to handle new objects over time. Integrating visual and linguistic modalities through fusion techniques ensures a comprehensive understanding of novel objects.

From simulated data to real-world object perception. Labeling the data is always a challenging problem. Additionally, there is no standard format for the annotations. From one perspective, developing an easy-to-use tool for recording and labeling data would make a significant contribution to the community. Particularly, the corresponding event streams, intensity frames, and depth information could be generated by a simulator based on the working principle of the sensor. Simultaneously, the basic facts of all recording data, including the trajectory of the sensor, the label of the object, and even the optical flow, are also generated without the need for annotation. With photorealistic virtual scenes and realistic sensor models, the future development of visual sensing and perception systems will be accelerated by prototyping on simulated data with transfer learning methods.

Limitations. Event-based neuromorphic vision is an emerging technique in the era of mature sensor hardware for object perception. Compared with LiDAR, radar and cameras are unfair because event-based sensors are not at the same maturity level as others. There are no appearance features such as color and texture because the event-based neuromorphic vision sensor only transmits local pixel-level changes, making it perform poorly in some applications with high requirements for appearance features. Although researchers have used the method of image reconstruction to reconstruct image frames from event streams, the quality of the reconstructed image frames is still not comparable to the output data produced by RGB cameras. The application of an event-based neuromorphic vision sensor is limited in some scenarios where energy, latency, and dynamic range are not important, especially in high-resolution complex scenarios.

Bibliography

[Ahm+18] Ahmed, I., Ahmad, A., Piccialli, F., Sangaiah, A. K., and Jeon, G. “A Robust Features-Based Person Tracker for Overhead Views in Industrial Environment”. In: *IEEE Internet of Things Journal* 5.3 (2018), pp. 1598–1605. DOI: 10.1109/JIOT.2017.2787779.

[Ahm+20] Ahmed, I., Din, S., Jeon, G., and Piccialli, F. “Exploring Deep Learning Models for Overhead View Multiple Object Detection”. In: *IEEE Internet of Things Journal* 7.7 (2020), pp. 5737–5744. DOI: 10.1109/JIOT.2019.2951365.

[AF21] Ainetter, S. and Fraundorfer, F. “End-to-end Trainable Deep Neural Network for Robotic Grasp Detection and Semantic Segmentation from RGB”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 13452–13458. DOI: 10.1109/ICRA48506.2021.9561398.

[AM19] Alonso, I. and Murillo, A. C. “EV-SegNet: Semantic Segmentation for Event-based Cameras”. In: *2019 IEEE/Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019).

[ATH18a] Asif, U., Tang, J., and Harrer, S. “Densely Supervised Grasp Detector (DSGD)”. In: *AAAI* (2018).

[ATH18b] Asif, U., Tang, J., and Harrer, S. “GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 4875–4882.

[ATH19] Asif, U., Tang, J., and Harrer, S. “Densely Supervised Grasp Detector (DSGD)”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (2019), pp. 8085–8093. DOI: 10.1609/aaai.v33i01.33018085. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4816.

[BJ15] BA, W. and J, W. “Field Block Net for AccurReceptiveate and Fast Object Detec-Computational neuroimaging and population recep-tive eldstion”. In: *Trends in Cognitive Sciences*. 2015.

[Bag+20] Baghaei Naeini, F., AlAli, A. M., Al-Husari, R., Rigi, A., Al-Sharman, M. K., Makris, D., and Zweiri, Y. “A Novel Dynamic-Vision-Based Approach for Tactile Sensing Applications”. In: *IEEE Transactions on Instrumentation and Measurement* 69.5 (2020), pp. 1881–1893. DOI: 10.1109/TIM.2019.2919354.

[Bel+16] Bell, S., Zitnick, C. L., Bala, K., and Girshick, R. “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2874–2883.

[Bin+17] Binas, J., Neil, D., Liu, S., and Delbrück, T. “DDD17: End-To-End DAVIS Driving Dataset”. In: *ICML* (2017).

[Bur06] Burkitt, A. N. “A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input”. In: *Biological cybernetics* 95.1 (2006), pp. 1–19.

[Cao+22a] Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. “Efficient Grasp Detection Network With Gaussian-Based Grasp Representation for Robotic Manipulation”. In: *IEEE/ASME Transactions on Mechatronics* (2022), pp. 1–11. DOI: 10.1109/TMECH.2022.3224314.

[Cao+22b] Cao, H., Chen, G., Li, Z., Hu, Y., and Knoll, A. “NeuroGrasp: Multimodal Neural Network With Euler Region Regression for Neuromorphic Vision-Based Grasp Pose Estimation”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–11. DOI: 10.1109/TIM.2022.3179469.

[Cao+21a] Cao, H., Chen, G., Li, Z., Lin, J., and Knoll, A. “Residual Squeeze-and-Excitation Network with Multi-scale Spatial Pyramid Module for Fast Robotic Grasping Detection”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 13445–13451. DOI: 10.1109/ICRA48506.2021.9561836.

[Cao+21b] Cao, H., Chen, G., Xia, J., Zhuang, G., and Knoll, A. “Fusion-Based Feature Attention Gate Component for Vehicle Detection Based on Event Camera”. In: *IEEE Sensors Journal* 21.21 (2021), pp. 24540–24548. DOI: 10.1109/JSEN.2021.3115016.

[Cao+22c] Cao, H., Peng, B., Jia, L., Li, B., Knoll, A., and Chen, G. “Orientation-aware People Detection and Counting Method based on Overhead Fisheye Camera”. In: *2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. 2022, pp. 1–7. DOI: 10.1109/MFI55806.2022.9913868.

[Cao+22d] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., and Wang, M. “Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation”. In: *Proceedings of the European Conference on Computer Vision Workshops(ECCVW)*. 2022.

[Che+20a] Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. “Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception”. In: *IEEE Signal Processing Magazine* 37.4 (2020), pp. 34–49. DOI: 10.1109/MSP.2020.2985815.

[Che+20b] Chen, G., Chen, W., Yang, Q., Xu, Z., Yang, L., Conradt, J., and Knoll, A. “A Novel Visible Light Positioning System With Event-Based Neuromorphic Vision Sensor”. In: *IEEE Sensors Journal* 20.17 (2020), pp. 10211–10219. DOI: 10.1109/JSEN.2020.2990752.

[Che+20c] Chen, G., Hong, L., Dong, J., Liu, P., Conradt, J., and Knoll, A. “EDDD: Event-Based Drowsiness Driving Detection Through Facial Motion Analysis With Neuromorphic Vision Sensor”. In: *IEEE Sensors Journal* 20.11 (2020), pp. 6170–6181. DOI: 10.1109/JSEN.2020.2973049.

[Che+20d] Chen, G., Wang, F., Li, W., Hong, L., Conradt, J., Chen, J., Zhang, Z., Lu, Y., and Knoll, A. “NeuroIV: Neuromorphic Vision Meets Intelligent Vehicle Towards Safe Driving With a New Database and Baseline Evaluations”. In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–13. DOI: 10.1109/TITS.2020.3022921.

[Che+18] Chen, G., Cao, H., Aafaque, M., Chen, J., Ye, C., Röhrbein, F., Conradt, J., Chen, K., Bing, Z., Liu, X., et al. "Neuromorphic vision based multivehicle detection and tracking for intelligent transportation system". In: *Journal of advanced transportation* 2018 (2018).

[Che+19] Chen, G., Cao, H., Ye, C., Zhang, Z., Liu, X., Mo, X., Qu, Z., Conradt, J., Röhrbein, F., and Knoll, A. "Multi-Cue Event Information Fusion for Pedestrian Detection With Neuromorphic Vision Sensors". In: *Frontiers in Neurorobotics* 13 (2019), p. 10. ISSN: 1662-5218. DOI: 10.3389/fnbot.2019.00010. URL: https://www.frontiersin.org/article/10.3389/fnbot.2019.00010.

[Che+21a] Chen, G., Kai, C., Lijun, Z., Liming, Z., and Knoll, A. "VCANet: Vanishing Point Guided Context-Aware Network for Small Road Hazards Detection". In: *Automotive Innovation* (2021). DOI: 10.1007/s42154-021-00157-x.

[Che+21b] Chen, G., Lu, F., Li, Z., Liu, Y., Dong, J., Zhao, J., Yu, J., and Knoll, A. "Pole-Curb Fusion based Robust and Efficient Autonomous Vehicle Localization System with Branch-and-Bound Global Optimization and Local Grid Map Method". In: *IEEE Transactions on Vehicular Technology (TVT)* (2021).

[Che+22] Chen, G., Qu, S., Li, Z., Zhu, H., Dong, J., Liu, M., and Conradt, J. "Neuromorphic Vision-Based Fall Localization in Event Streams With Temporal-Spatial Attention Weighted Network". In: *IEEE Transactions on Cybernetics* (2022), pp. 1–12.

[Che+17] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., and Chua, T.-S. "SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6298–6306. DOI: 10.1109/CVPR.2017.667.

[Che+20e] Chen, L., Huang, P., Li, Y., and Meng, Z. "Edge-dependent Efficient Grasp Rectangle Search in Robotic Grasp Detection". In: *IEEE/ASME Transactions on Mechatronics* (2020), pp. 1–1. DOI: 10.1109/TMECH.2020.3048441.

[CHM19] Chen, L., Huang, P., and Meng, Z. "Convolutional multi-grasp detection using grasp path for RGBD images". In: *Robotics and Autonomous Systems* 113 (2019), pp. 94–103.

[Che18] Chen, N. F. Y. "Pseudo-Labels for Supervised Learning on Dynamic Vision Sensor Data, Applied to Object Detection Under Ego-Motion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2018.

[Che+20f] Cheng, B., Wu, W., Tao, D., Mei, S., Mao, T., and Cheng, J. "Random Cropping Ensemble Neural Network for Image Classification in a Robotic Arm Grasping System". In: *IEEE Transactions on Instrumentation and Measurement* 69.9 (2020), pp. 6795–6806. DOI: 10.1109/TIM.2020.2976420.

[CXV18] Chu, F.-J., Xu, R., and Vela, P. A. "Real-World Multiobject, Multigrasp Detection". In: *IEEE Robotics and Automation Letters* 3 (2018), pp. 3355–3362.

[DT05a] Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[DT05b] Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[Den+09] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[DDC18] Depierre, A., Dellandréa, E., and Chen, L. “Jacquard: A Large Scale Dataset for Robotic Grasp Detection”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 3511–3516. DOI: 10.1109/IROS.2018.8593950.

[Dol+14] Dollár, P., Appel, R., Belongie, S., and Perona, P. “Fast Feature Pyramids for Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014), pp. 1532–1545. DOI: 10.1109/TPAMI.2014.2300479.

[Dou20] Douglas Morrison Peter Corke, J. L. “Learning robust, real-time, reactive robotic grasping”. In: *The International Journal of Robotics Research* 39.2-3 (2020), pp. 183–201.

[DWL21] Du, G., Wang, K., and Lian, S. “Vision-based Robotic Grasping from Object Localization, Pose Estimation, Grasp Detection to Motion Planning: A Review”. In: *Artificial Intelligence Review* 54 (2021).

[Dua+19] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. “CenterNet: Keypoint Triplets for Object Detection”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6568–6577. DOI: 10.1109/ICCV.2019.00667.

[Dua+20] Duan, Z., Ozan Tezcan, M., Nakamura, H., Ishwar, P., and Konrad, J. “RAPiD: Rotation-Aware People Detection in Overhead Fisheye Images”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2700–2709. DOI: 10.1109/CVPRW50498.2020.00326.

[EG09] Enzweiler, M. and Gavrila, D. M. “Monocular Pedestrian Detection: Survey and Experiments”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (2009), pp. 2179–2195. DOI: 10.1109/TPAMI.2008.260.

[Eve+10] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[Fu+17] Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. “Dssd: Deconvolutional single shot detector”. In: *arXiv preprint arXiv:1701.06659* (2017).

[Gal+20] Gallego, G., Delbruck, T., Orchard, G. M., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., and Scaramuzza, D. “Event-based Vision: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/TPAMI.2020.3008413.

[Gar+19] Gariépy, A., Ruel, J.-C., Chaib-draa, B., and Giguère, P. “GQ-STN: Optimizing One-Shot Grasp Detection based on Robustness Classifier”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019), pp. 3996–4003.

[Geh+19] Gehrig, D., Loquercio, A., Derpanis, K. G., and Scaramuzza, D. “End-to-End Learning of Representations for Asynchronous Event-Based Data”. In: *IEEE International Conference on Computer Vision (ICCV), Seoul* (2019).

[GLU12] Geiger, A., Lenz, P., and Urtasun, R. “Are we ready for autonomous driving? The KITTI vision benchmark suite”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[Gka+20] Gkanatsios, N., Chalvatzaki, G., Maragos, P., and Peters, J. *Orientation Attentive Robot Grasp Synthesis*. 2020. arXiv: 2006.05123 [cs.RO].

[Guo+17] Guo, D., Sun, F., Liu, H., Kong, T., Fang, B., and Xi, N. “A hybrid deep architecture for robotic grasp detection”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. May 2017. DOI: 10.1109/ICRA.2017.7989191.

[Guo+22] Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R. R., Cheng, M.-M., and Hu, S.-M. “Attention mechanisms in computer vision: A survey”. In: *Computational Visual Media* 8.3 (2022), pp. 331–368.

[He+20] He, K., Gkioxari, G., Dollár, P., and Girshick, R. “Mask R-CNN”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (2020), pp. 386–397. DOI: 10.1109/TPAMI.2018.2844175.

[He+16] He, K., Zhang, X., Ren, S., and Sun, J. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[HSS18] Hu, J., Shen, L., and Sun, G. “Squeeze-and-Excitation Networks”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141.

[HDL20] Hu, Y., Delbruck, T., and Liu, S.-C. “Learning to Exploit Multiple Vision Modalities by Using Grafted Networks”. In: *2020 European Conference on Computer Vision (ECCV)*. 2020.

[Hua+17] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[Ina+19] Inagaki, Y., Araki, R., Yamashita, T., and Fujiyoshi, H. “Detecting layered structures of partially occluded objects for bin picking”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 5786–5791.

[IS15] Ioffe, S. and Szegedy, C. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. pmlr. 2015, pp. 448–456.

[Jad+15] Jaderberg, M., Simonyan, K., Zisserman, A., and, k. kavukcuoglu koray. “Spatial Transformer Networks”. In: *Advances in Neural Information Processing Systems 28*. 2015, pp. 2017–2025.

[Jen+20] Jeny, A. A., Sakib, A. N. M., Junayed, M. S., Lima, K. A., Ahmed, I., and Islam, M. B. “SkNet: A Convolutional Neural Networks Based Classification Approach for Skin Cancer Classes”. In: *2020 23rd International Conference on Computer and Information Technology (ICCIT)*. 2020, pp. 1–6. DOI: 10.1109/ICCIT51783.2020.9392716.

[JMS11a] Jiang, Y., Moseson, S., and Saxena, A. “Efficient grasping from RGBD images: Learning using a new rectangle representation.” In: *ICRA*. IEEE, 2011, pp. 3304–3311.

[JMS11b] Jiang, Y., Moseson, S., and Saxena, A. “Efficient grasping from RGBD images: Learning using a new rectangle representation.” In: *ICRA*. IEEE, 2011, pp. 3304–3311.

[Jia+19] Jiang, Z., Xia, P., Huang, K., Stechele, W., Chen, G., Bing, Z., and Knoll, A. “Mixed Frame-/Event-Driven Fast Pedestrian Detection”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 8332–8338. DOI: 10.1109/ICRA.2019.8793924.

[JLD16] Johns, E., Leutenegger, S., and Davison, A. J. “Deep Learning a Grasp Function for Grasping under Gripper Pose Uncertainty”. In: *CoRR* (2016).

[KK17a] Khodamoradi, A. and Kastner, R. “O(N)-Space Spatiotemporal Filter for Reducing Noise in Neuromorphic Vision Sensors”. In: *IEEE Transactions on Emerging Topics in Computing* (2017), pp. 1–1. ISSN: 2168-6750. DOI: 10.1109/TETC.2017.2788865.

[KB15] Kingma, D. P. and Ba, J. “Adam: A Method for Stochastic Optimization”. In: *ICLR (Poster)*. 2015. URL: http://arxiv.org/abs/1412.6980.

[Kon+16] Kong, T., Yao, A., Chen, Y., and Sun, F. “Hypernet: Towards accurate region proposal generation and joint object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 845–853.

[KSH17] Krizhevsky, A., Sutskever, I., and Hinton, G. E. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[KK17b] Kumra, S. and Kanan, C. “Robotic grasp detection using deep convolutional neural networks”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2017, pp. 769–776. DOI: 10.1109/IROS.2017.8202237.

[LLS15] Lenz, I., Lee, H., and Saxena, A. “Deep learning for detecting robotic grasps”. In: *The International Journal of Robotics Research* 34.4-5 (2015), pp. 705–724. DOI: 10.1177/0278364914549607.

[Lev+18] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection”. In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436. DOI: 10.1177/0278364917710318.

[Li+20a] Li, B., Cao, H., Qu, Z., Hu, Y., Wang, Z., and Liang, Z. “Event-Based Robotic Grasping Detection With Neuromorphic Vision Sensor and Event-Grasping Dataset”. In: *Frontiers in Neurorobotics* 14 (2020), p. 51. ISSN: 1662-5218.

[Li+19a] Li, J., Dong, S., Yu, Z., Tian, Y., and Huang, T. “Event-Based Vision Enhanced: A Joint Detection Framework in Autonomous Driving”. In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. 2019, pp. 1396–1401. DOI: 10.1109/ICME.2019.00242.

[Li+19b] Li, S., Tezcan, M. O., Ishwar, P., and Konrad, J. “Supervised People Counting Using An Overhead Fisheye Camera”. In: *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2019, pp. 1–8. DOI: 10.1109/AVSS.2019.8909877.

[Li+20b] Li, W., Cao, H., Liao, J., Xia, J., Cao, L., and Knoll, A. “Parking Slot Detection on Around-View Images Using DCNN”. In: *Frontiers in Neurorobotics* 14 (2020), p. 46. ISSN: 1662-5218. DOI: 10.3389/fnbot.2020.00046. URL: https://www.frontiersin.org/article/10.3389/fnbot.2020.00046.

[LPD08] Lichtsteiner, P., Posch, C., and Delbruck, T. "A 128 × 128 120 dB 15 μ s Latency Asynchronous Temporal Contrast Vision Sensor". In: *IEEE journal of solid-state circuits* 43.2 (2008), pp. 566–576.

[Lin+17] Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[Lin+20] Lin, T., Goyal, P., Girshick, R., He, K., and Dollár, P. "Focal Loss for Dense Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (2020), pp. 318–327. DOI: 10.1109/TPAMI.2018.2858826.

[Lin+14] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[Liu+21] Liu, D., Tao, X., Yuan, L., Du, Y., and Cong, M. "Robotic Objects Detection and Grasping in Clutter based on Cascaded Deep Convolutional Neural Network". In: *IEEE Transactions on Instrumentation and Measurement* (2021), pp. 1–1. DOI: 10.1109/TIM.2021.3129875.

[Liu+15] Liu, H., Brandli, C., Li, C., Liu, S., and Delbruck, T. "Design of a spatiotemporal correlation filter for event-based sensors". In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2015, pp. 722–725. DOI: 10.1109/ISCAS.2015.7168735.

[Liu+19] Liu, S., Rueckauer, B., Ceolini, E., Huber, A., and Delbruck, T. "Event-Driven Sensing for Efficient Perception: Vision and Audition Algorithms". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 29–37. ISSN: 1558-0792. DOI: 10.1109/MSP.2019.2928127.

[Liu+20] Liu, S., Wang, F., Liu, Z., Zhang, W., Tian, Y., and Zhang, D. "A Two-Finger Soft-Robotic Gripper with Enveloping and Pinching Grasping Modes". In: *IEEE/ASME Transactions on Mechatronics* (2020), pp. 1–1.

[Liu+18] Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. "Path Aggregation Network for Instance Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8759–8768. DOI: 10.1109/CVPR.2018.00913.

[LHW18] Liu, S., Huang, D., and Wang, a. "Receptive Field Block Net for Accurate and Fast Object Detection". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[Liu+16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. "Ssd: Single shot multibox detector". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37.

[Lou+20] Lou, L., Li, Q., Zhang, Z., Yang, R., and He, W. "An IoT-Driven Vehicle Detection Method Based on Multisource Data Fusion Technology for Smart Parking Management System". In: *IEEE Internet of Things Journal* 7.11 (2020), pp. 11020–11029. DOI: 10.1109/JIOT.2020.2992431.

[Low99] Lowe, D. G. "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[Mah+17] Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J. A., and Goldberg, K. “Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics”. In: *Robotics: Science and Systems* (2017).

[MM91] Mahowald, M. A. and Mead, C. “The Silicon Retina”. In: *Scientific American* 264.5 (1991), pp. 76–83. ISSN: 00368733, 19467087.

[Maq+18] Maqueda, A. I., Loquercio, A., Gallego, G., Garca, N., and Scaramuzza, D. “Event-based vision meets deep learning on steering prediction for self-driving cars”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5419–5427.

[Mir+18] Mirus, F., Axenie, C., Stewart, T. C., and Conradt, J. “Neuromorphic sensorimotor adaptation for robotic mobile manipulation: From sensing to behaviour”. In: *Cognitive Systems Research* 50 (2018), pp. 52–66. ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2018.03.006.

[MCL20] Morrison, D., Corke, P., and Leitner, J. “Learning robust, real-time, reactive robotic grasping”. In: *The International Journal of Robotics Research* 39.2-3 (2020), pp. 183–201. DOI: 10.1177/0278364919859066.

[MBS17] Mueggler, E., Bartolozzi, C., and Scaramuzza, D. “Fast event-based corner detection”. In: *28th British Machine Vision Conference (BMVC)*. 2017.

[Mut+20] Muthusamy, R., Huang, X., Zweiri, Y., Seneviratne, L., and Gan, D. “Neuromorphic Event-Based Slip Detection and Suppression in Robotic Grasping and Manipulation”. In: *IEEE Access* 8 (2020), pp. 153364–153384. DOI: 10.1109/ACCESS.2020.3017738.

[NLO16] Nguyen, D. T., Li, W., and Ogunbona, P. O. “Human detection from images and videos: A survey”. In: *Pattern Recognition* 51 (2016), pp. 148–175. ISSN: 0031-3203.

[Okt+18] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D. “Attention U-Net: Learning Where to Look for the Pancreas”. In: *IMIDL Conference* (2018).

[PBO18] Padala, V., Basu, A., and Orchard, G. “A Noise Filtering Algorithm for Event-Based Asynchronous Change Detection Image Sensors on TrueNorth and Its Implementation on TrueNorth”. In: *Frontiers in Neuroscience* 12 (2018), p. 118. ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00118.

[PC18] Park, D. and Chun, S. Y. “Classification based Grasp Detection using Spatial Transformer Network”. In: *CoRR* abs/1803.01356 (2018).

[PSC18a] Park, D., Seo, Y., and Chun, S. Y. “Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images”. In: *CoRR* abs/1809.05828 (2018). Withdrawn. URL: http://arxiv.org/abs/1809.05828.

[PSC18b] Park, D., Seo, Y., and Chun, S. Y. “Rotation Ensemble Module for Detecting Rotation-Invariant Features”. In: *CoRR* (2018).

[Par+18] Park, J., Woo, S., Lee, J.-Y., and Kweon, I. S. “Bam: Bottleneck attention module”. In: *arXiv preprint arXiv:1807.06514* (2018).

[Pay+05] Payeur, P., Pasca, C., Cretu, A.-M., and Petriu, E. “Intelligent haptic sensor system for robotic manipulation”. In: *IEEE Transactions on Instrumentation and Measurement* 54.4 (2005), pp. 1583–1592. DOI: 10.1109/TIM.2005.851422.

[Per+20] Perot, E., Tournemire, P. de, Nitti, D., Masci, J., and Sironi, A. "Learning to Detect Objects with a 1 Megapixel Event Camera". In: *NeurIPS 2020 Conference*. 2020.

[PSM10] Perronnin, F., Sánchez, J., and Mensink, T. "Improving the fisher kernel for large-scale image classification". In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer. 2010, pp. 143–156.

[PG16] Pinto, L. and Gupta, A. "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 3406–3413.

[PBK14] Pokorny, F. T., Bekiroglu, Y., and Kragic, D. "Grasp moduli spaces and spherical harmonics". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 389–396.

[RA15] Redmon, J. and Angelova, A. "Real-time grasp detection using convolutional neural networks". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. Seattle: IEEE, 2015.

[RF16] Redmon, J. and Farhadi, A. "YOLO9000: Better, Faster, Stronger". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[RF18] Redmon, J. and Farhadi, A. "YOLOv3: An Incremental Improvement". In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

[Ren+15] Ren, S., He, K., Girshick, R., and Sun, J. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28*. Ed. by Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. Curran Associates, Inc., 2015, pp. 91–99.

[RG17] Rickert, M. and Gaschler, A. "Robotics library: An object-oriented approach to robot applications". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 733–740. DOI: 10.1109/IROS.2017.8202232.

[SDN08] Saxena, A., Driemeyer, J., and Ng, A. Y. "Robotic Grasping of Novel Objects Using Vision". In: *The International Journal of Robotics Research* 27.2 (Feb. 2008), pp. 157–173. ISSN: 0278-3649. DOI: 10.1177/0278364907087172.

[Shi+16] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1874–1883. DOI: 10.1109/CVPR.2016.207.

[Sim+18] Simon, M., Amende, K., Kraus, A., Honer, J., and Gross, H. M. "Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds". In: *IEEE Computer Vision and Pattern Recognition Workshop* (2018).

[SZ14] Simonyan, K. and Zisserman, A. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[Son+20] Song, Y., Gao, L., Li, X., and Shen, W. "A novel robotic grasp detection method based on region proposal networks". In: *Robotics and Computer-Integrated Manufacturing* 65 (2020), p. 101963. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2020.101963. URL: http://www.sciencedirect.com/science/article/pii/S0736584519308105.

[Suc+19] Suchi, M., Patten, T., Fischinger, D., and Vincze, M. "EasyLabel: A Semi-Automatic Pixel-wise Object Annotation Tool for Creating Robotic RGB-D Datasets". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 6678–6684. DOI: 10.1109/ICRA.2019.8793917.

[Sze+17] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

[Sze+15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[Sze+16] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[THM19] Tamura, M., Horiguchi, S., and Murakami, T. "Omnidirectional Pedestrian Detection by Rotation Invariant Training". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019, pp. 1989–1998. DOI: 10.1109/WACV.2019.00216.

[Tia+20] Tian, Z., Shen, C., Chen, H., and He, T. "FCOS: A Simple and Strong Anchor-free Object Detector". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/TPAMI.2020.3032166.

[Urm+08] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., and Geyer, C. "Autonomous Driving in Urban Environments: Boss and the Urban Challenge". In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466.

[Wan20] Wang, D. *SGDN: Segmentation-Based Grasp Detection Network For Unsymmetrical Three-Finger Gripper*. 2020. arXiv: 2005.08222 [cs.RO].

[Wan+20] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. "ECA-Net: Efficient channel attention for deep convolutional neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11534–11542.

[Wan+19a] Wang, S., Jiang, X., Zhao, J., Wang, X., Zhou, W., and Liu, Y. "Efficient Fully Convolution Neural Network for Generating Pixel Wise Robotic Grasps With High Resolution Images". In: *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2019, pp. 474–480. DOI: 10.1109/ROBIO49542.2019.8961711.

[Wan+18] Wang, X., Girshick, R., Gupta, A., and He, K. "Non-Local Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[Wan+19b] Wang, Y., Du, B., Shen, Y., Wu, K., Zhao, G., Sun, J., and Wen, H. "EV-Gait: Event-Based Robust Gait Recognition Using Dynamic Vision Sensors". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[WG21] Wen, K. and Gosselin, C. "Static Model Based Grasping Force Control of Parallel Grasping Robots with Partial Cartesian Force Measurement". In: *IEEE/ASME Transactions on Mechatronics* (2021), pp. 1–1. DOI: 10.1109/TMECH.2021.3077448.

[Woo+18] Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. “CBAM: Convolutional Block Attention Module”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[Wu+19] Wu, G., Chen, W., Cheng, H., Zuo, W., Zhang, D. Z., and You, J. “Multi-Object Grasping Detection With Hierarchical Feature Fusion”. In: *IEEE Access* 7 (2019), pp. 43884–43894.

[Xie+21] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers”. In: *Advances in Neural Information Processing Systems*. Ed. by Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. Vol. 34. Curran Associates, Inc., 2021, pp. 12077–12090. URL: https://proceedings.neurips.cc/paper/2021/file/64f1f27bf1b4ec22924fd0acb550c235-Paper.pdf.

[YMS11] Yun Jiang, Moseson, S., and Saxena, A. “Efficient grasping from RGBD images: Learning using a new rectangle representation”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3304–3311.

[Zan+19] Zanardi, A., Aumiller, A., Zilly, J., Censi, A., and Frazzoli, E. “Cross-Modal Learning Filters for RGB-Neuromorphic Wormhole Learning”. In: *Robotics: Science and System XV*. 15th Robotics: Science and Systems (RSS 2019); Conference Location: Freiburg im Breisgau, Germany; Conference Date: June 22-26, 2019. 2019, P45. DOI: 10.3929/ethz-b-000349414.

[Zha+19a] Zhang, H., Lan, X., Bai, S., Zhou, X., Tian, Z., and Zheng, N. “ROI-based Robotic Grasp Detection for Object Overlapping Scenes”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 4768–4775.

[Zha+19b] Zhang, H., Zhou, X., Lan, X., Li, J., Tian, Z., and Zheng, N. “A Real-Time Robotic Grasping Approach With Oriented Anchor Box”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2019), pp. 1–12.

[Zha+17] Zhang, Qiang, Qu, Daokui, Xu, Fang, and Zou, Fengshan. “Robust Robot Grasp Detection in Multimodal Fusion”. In: *MATEC Web Conf.* 139 (2017), p. 00060.

[Zho+18] Zhou, X., Lan, X., Zhang, H., Tian, Z., Zhang, Y., and Zheng, N. “Fully Convolutional Grasp Detection Network with Oriented Anchor Box”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 7223–7230. DOI: 10.1109/IROS.2018.8594116.

[Zhu+18a] Zhu, A. Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., and Daniilidis, K. “The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception”. In: *IEEE Robotics and Automation Letters* 3.3 (2018). ISSN: 2377-3766.

[Zhu+18b] Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. “EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras”. In: *Robotics:Science and System* (2018), pp. 1–9.