Journal of Nondestructive Evaluation (2022) 41:74
https://doi.org/10.1007/s10921-022-00907-9

Infrared Thermal Imaging-Based Turbine Blade Crack Classification Using Deep Learning

Benedict E. Jaeger[1] · Simon Schmid[1] · Christian U. Grosse[1] · Anian Gögelein[2] · Frederik Elischberger[2]

Received: 28 July 2021 / Accepted: 9 October 2022 / Published online: 19 October 2022

Abstract
Non-destructive testing is widely applied for the detection and identification of defects in turbine blades of modern aircraft engines. Cracks in turbine blades can affect the turbine performance and pose a risk to safety and service life. For Original Equipment Manufacturers it is, therefore, essential to be able to identify all defects. Heat flow thermography offers, compared to the often used penetrant testing, the potential to improve the detection of defects in turbine blades and is contact-free, reproducible, quick to apply, and can be automated. With induction (heat flow) thermography, it is even possible to detect cracks that lie below the surface and therefore are not externally visible. However, manual inspection of thermography images is very time-consuming. By automating the image classification procedure with a deep learning technique, the speed and accuracy of the classification can be improved over a manually performed classification. The development objective of this AI application is expected to support and assist the highly skilled and experienced inspection specialists in the medium term. Our solution is based on convolutional neural networks. Several challenges of the AI training process, including data imbalance, a small dataset, and extremely small cracks in large images are addressed.

Keywords Artifical neural networks · Deep learning · Convolutional neural networks · Turbine blade cracks · Crack/defect classification · Infrared thermal imaging · Non-destructive testing · Data imbalance

1 Introduction

Non-destructive testing (NDT) techniques are of high importance for aircraft engines manufacturers, to ensure an aircraft engine's safety and optimal operational performance [1,2]. Of particular interest are the turbine blades, as those complex and very thin components are exposed to high loads, extreme conditions [3] and are chemically highly impure [2]. Elevated requirements imposed on these high performance components make it increasingly necessary to introduce new and improved NDT techniques into the process of turbine blade production [3]. Especially at the trailing edge of a turbine blade, which is significantly thinner than the leading edge, a possible incipient crack can lead to a faster crack propagation. Inspection specialists further showed that conventional eddy current testing sometimes leads to incorrect test results due to lift-off signals at that particular turbine blade area.

The MTU Aero Engines AG already performs, supplementary to the standard (fluorescent) penetrant inspection, an additional examination based on the manual analysis of infrared induction thermography images for an extended identification of cracks after production and before assembly. Besides the potential of improving the detection of cracks overall, a key advantage of induction thermography over penetrant testing is that, in addition to surface cracks, also sub-surface cracks can be detected [4,5].

However, as the manual identification of cracks in the images is time-consuming and requires skilled and experienced specialists, this work presents an approach for an automated solution to significantly increase the efficiency of the operational classification. Induction thermography is further also very well suited for a process-integrated quality control of series production, making its application even more desirable [6]. The major advances in the field of deep learning made in recent years, especially in the area of image

✉ Benedict E. Jaeger
benedict.jaeger@tum.de

[1] Chair of Non-destructive Testing, Technical University of Munich, Franz-Langinger-Strasse 10, 81245 Munich, Germany

[2] MTU Aero Engines AG, Dachauer Strasse 665, 80995 Munich, Germany

recognition, offer great potential when used in combination with NDT techniques.

In this work, a ResNet-18 convolutional neural network (CNN) is used in the two models developed, which are either trained on large images (Large Image Model—LIM) or small image patches (Small Image Model—SIM) to classify turbine blade images either crack or crack-free. The application of deep learning for turbine blade defect identification is still an area of research. To our best knowledge, this is the first time where deep learning is utilized to classify defects in thermographic images of aircraft engine turbine blades. The main challenges can be summarized in the following points:

- Image dimensions are quite large (512 × 640 pixels)
- Cracks in the turbine blade trailing edges are extremely small (approx. between 8 × 6 to 8 × 34 pixels), resulting in the number of pixels representing a crack being significantly less than 0.1% (max.: 0.08%; min.: 0.01%) of the total number of pixels of an image (512 × 640 pixels).
- Cracks are difficult or not even possible to identify for untrained people.
- The necessity of converting the 16-bit grayscale images into a false-color representation to make the cracks visible and analyzable before model development. However, the actual models were trained on the 16-bit grayscale images.
- Only a relatively small dataset, according to deep learning standards is available. Due to the protection of OEMs' (original equipment manufacturer) intellectual property, no publicly available datasets are available, regarding cracks in turbine blades.
- The dataset is characterized by a data imbalance of 11.5. In combination with the particularly low amount of crack samples, this property negatively affects the model training.
- Of the total crack images, the number of cracks and their positions is only known for 40% of the images. This is particularly disadvantageous when cropping smaller partial image patches from the original 512 × 640 pixel images.

2 Related Work

Especially from civil engineering, quite a few deep learning solutions and approaches for detecting cracks in concrete tunnels, pavements, or other concrete structures are available [7–11]. However, these cracks are fundamentally different from cracks in turbine blades. Cracks in construction materials, such as concrete and asphalt, are usually significantly longer and wider. Asphalt cracks normally range in lengths of up to several meters and widths in the range of a few millimeters up to a few centimeters [10]. Compared to asphalt cracks, concrete cracks are generally shorter and less wide but still much longer and wider than turbine blade cracks. The concrete cracks range from thin hairline cracks to large cracking of partial building and tunnel structures in ranges of several meters [7,9]. Turbine blade cracks that occur during production (like those in the present image dataset) have smaller aspect ratios and spatial expansions than cracks in concrete and asphalt.

Panizza et al. [1] deployed a RetinaNet for object detection of drilling defects in cooling holes of high-pressure gas turbine blades. The dataset consists of 560 defect-free and 134 defective high-resolution X-ray images including large uninformative background. The RetinaNet was pre-trained on the MS COCO dataset and further utilized data augmentation. The initial approach of cropping the original images (8496 × 6960 pixels) to the aerofoil (1900 × 1500 pixels) did not provide a good result, as the defects are extremely small compared to the aerofoil. In a further step, the aerofoil image was split up into 5 × 5 overlapping image patches of 500 × 600 pixels. By down-sampling the over-represented class the data imbalance was reduced to 1.1. Since the image patches partly overlap, patches of each image are either assigned to the training or validation set. During training, the image patches are further scaled up by a factor of 2 in height and width as the defects were smaller than the smallest bounding box (anchor) of the detection algorithm. With further anchor optimization finally a mAP (mean average precision) of 0.90 is reached.

In Khani et al. [12] a surface crack detection approach for gas turbine structures (e.g. turbine housing) is presented. For automated visual crack detection, various conventional digital image processing techniques were used with differing success. Therefore, Khani et al. [12] proposed a novel (surface) crack detection architecture that is especially characterized by the combination of digital image processing (median and bilateral filtering) and deep learning. The dataset consisted of 250.000 labeled image patches (40 × 40 pixels) from 700 gas turbine surface images [12]. The results showed that, by applying filters to the image data before training the model, the classification performance of the CNN model could be significantly improved. The final model yielded an accuracy of 96.26% on the cracked surface dataset [12]. However, the cracks detected and classified in [12] are much more comparable to fine medium-length concrete cracks than to the characteristics of the cracks in turbine blades presented in this work.

Yang et al. [13] investigated cracks in steel plates using induction thermography and a deep learning approach. The crack characteristics in [13] are only to a very small extent comparable with the cracks analyzed in this work. However, the method of infrared thermography is used for image acquisition, and it is shown that this particular NDT technique, in combination with a CNN, is suitable for crack detection. The input to the proposed Faster R-CNN architecture was 3000

inductive thermographic images for training and 125 for testing. The model was trained to detect and classify into three different crack classes (penetrating cracks, non-penetrating cracks, and shallow surface scratches) while achieving an accuracy of 95.54%. For feature extraction, a VGG-16 architecture, pre-trained on the ImageNet dataset, was used [13].

In Soukup & Huber-Moerk [14] a CNN model for classifying cracks of rail surface images is presented. The approach outperforms the currently used model-based approach with handcrafted parameter adjustment. The cracks considered are, to some extent, comparable to those in turbine blades. The photometric stereo image dataset includes a total of 2532 cavity and non-cavity images. Image patches with a size of 16×16 pixels were cropped out from the color images. It could be shown that a CNN model with unsupervised layer-wise pre-trained initialization (auto-encoder as a regularization method) resulted in a better performance than a CNN model with (standard) random initialization. With further applied data augmentation the error rate could be reduced from 0.67 to 0.556%. The CNN architecture used consisted of three convolutional layers, three max-pooling layers, and a final fully-connected layer.

In Xian et al. [15] a two-stage automatic detection of defects on metallic surfaces of industrial products is presented. The model accurately localizes and classifies defects in three-channel color images with a size of 2720×2040 pixels. A novel cascaded autoencoder (CASAE) module is first used to segment defective regions with semantic segmentation. Each defect in the resulted segmented image is further localized via a defect region detector and is cropped. The cropped defect patches are converted to grayscale images and are used as the input for the classification module. The patches are resized to 227×227 pixels and trained on a CNN architecture with five convolutional layers, three max-pooling layers, two batch normalization layers (after the first two convolutional layers), two fully connected layers, and a softmax layer at the end. The softmax layer converts the output values into probabilities for the three output classes (damage spot, glue mark, and dust/fiber). The developed model achieves an IoU (Intersection over Union) score of 89.60% using the industrial dataset DAGM 2007.

3 Approach for Capturing and Classification of the Thermographic Images

In this section, we first discuss in Sect. 3.1 the image acquisition process and its pre-processing, followed in Sect. 3.2 by the used neural network architecture including the investigated hyperparameters. The metrics to over-watch and evaluate the training process are described in Sect. 3.3, followed by the applied loss functions in Sect. 3.4. Finally, in Sect. 3.5 the dataset is discussed.

3.1 Infrared Induction Thermography Acquisition Setup

The images for the crack classification were acquired semi-automatically, using an experimental setup utilizing pulsed induction infrared thermography. Figure 1 shows the schematic test setup. Each turbine blade was first manually placed in a mounting bracket. Three images are sequentially captured along the trailing edge of the turbine blade. Each image, therefore, represents one third of a turbine blade trailing edge.

First, the turbine blade is heated by a pulse-shaped voltage induction through an induction wire positioned near the trailing edge. The heating duration for turbine blades typically ranges between 50 and 100 ms depending on the blade type (e.g. compressor blade, turbine blade, turbine vane, …), blade geometry, and the area inspected. Induction frequencies ranging from 200 to 500 kHz have proven to enable the detection of surface and sub-surface cracks in modern turbine blades, which are mainly made out of nickel-based or titanium-based alloys. The induction frequency f was provided by an Huettinger Axio high-frequency generator with a capacity of 10 kW and is the main control variable of the penetration depth δ of the material induced eddy currents [16]. The higher the induction frequency, the lower the penetration depth into the material [16]:

$$\delta = \sqrt{\left(\frac{2 * \rho}{\omega * \mu}\right)} \tag{1}$$

whereas μ is the permeability, ρ the specific resistance and ω the angular frequency with:

$$\omega = 2 * \pi * f \tag{2}$$

An industrial robot, equipped with an infrared camera (thermographic camera Radiance HS InSb, Ti=2 ms) at the

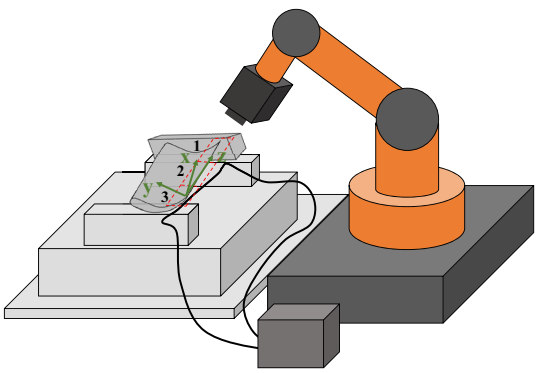

Fig. 1 Experimental induction thermography setup with an industrial robot arm equipped with an infrared camera. From each turbine blade trailing edge three images along the longitudinal blade z-axis are captured. The image acquisition plane corresponds to the y–z plane

end-effector, then detects and records a time series of the surface temperature distribution over the focused area of the turbine blade trailing edge. With a phase algorithm based on fast Fourier transformation (FFT), described in detail in [16], then, a phase image from each stack of thermal images, is calculated. For the calculation of each phase image only the most interesting part, i.e. a finite window of the function holding all relevant information, is considered. The resulting phase images are further normalized and exported as TIFF-images [16].

The use of phase images, compared to amplitude images, is particularly beneficial, as it also offers the possibility to visualize only weakly detectable cracks by reducing disturbing influences [17,18].

3.2 Neural Network Architecture and Hyperparameters

In this work, the deep learning library fastai was used. It provides a layered API (application programming interface) based on Python and the PyTorch library [19]. Two different models were trained using small and large images sizes (see Sect. 4). For the choice of the CNN architecture, different ImageNet pre-trained networks were compared in a two-staged evaluation process in Sect. 4. The results are listed in Tables 3 (first stage) and 4 (second stage). It was found that the ResNet-18 neural network architecture works best for the classification task at hand and is used in the further. For the stochastic gradient-based optimization, in all experiments, the Adam optimizer is utilized. Major hyperparameters chosen in this work are the learning rate, batch size, and weight decay (L2-regularization). For choosing a reasonable learning rate, the library fastai provides a so-called learning rate finder based on [20] and [21]. In addition, fastai uses the concept of a one-cycle policy, enabling a model to be trained on a learning rate range instead of a fixed or decreasing value [22,23]. To prevent overfitting the L2-regularization is introduced in both models. To determine a reasonable weight decay, a grid-search with different values is performed. As the batch size heavily depends on the strategies used to address the data imbalance and the fact that only very few samples, representing the class crack, are available, the choice of the batch size is discussed model-dependently in more detail in Sect. 4. To reduce the time needed for model training and to reduce the GPU memory usage, the concept of mixed precision training, introduced by [24], was further also implemented in both models. Hardware-wise a NVIDIA GeForce RTX 2080 Ti with a GPU-Memory of 11 GB and an Intel Xeon CPU E5-2620 v2 with 256 GB of RAM were used in all experiments.

3.3 Classification Metric

The confusion matrix represents a fundamental concept for evaluating binary classification models. This tabulated visualization contrasts the model predictions and its ground-truth labels (see Fig. 2). The confusion matrix rows represent the occurrence in an actual class, whereas the columns represent a predicted class occurrence [25].

As the decisive classification metric in this work we have chosen a F_β-score with an β value of 2. With an β of 2, this metric is also referred as F_2-score. The reason for this choice is the fact that the dataset at hand is imbalanced and contains significantly more crack-free images than crack images. Before computing a F_β-score, the value of precision and recall of the model have to be calculated using the following two equations [25]:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

While a high value of precision corresponds to having fewer false positives, a high recall value corresponds to having fewer false negatives. In the context of turbine blade cracks, a false negative (also known as type-2 error) is equivalent to an "overlooked crack" and a false positive (also known as type-1 error) is equivalent to a "false alarm". While an overlooked crack can lead in the worst case to a malfunctioning aircraft engine, an false alarm leads to a improperly rejected turbine blade from manufacturing thus reducing the profitability.

By choosing a β value of 2 for the F_β-score more emphasis is placed on finding a model with a high recall value instead of taking the harmonic mean of both, precision and recall. The F_β-score is calculated as follows [26]:

$$F_\beta = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall} \tag{5}$$

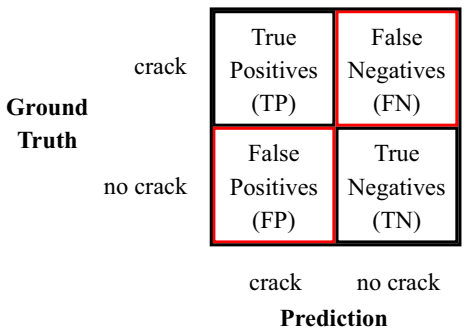

Fig. 2 Binary confusion matrix

The choice of more common classification metrics such as accuracy, ROC-curve (receiver operating characteristic), and AUROC (area under the receiver operating characteristic) do not provide any useful or representative results and partly lead to misleading results. The reason for this is, that the dataset is not only imbalanced but also involves a class with a very low amount of samples. This issue is known as class rarity and leads to the unreliable values of those metrics [27]. Another reason for using an F_2-score metric is, that it gives back a single value, with which the model performance can be easily compared with other models. In contrast, a model comparison is more difficult for metrics that return a curve (e.g. precision-recall-curve).

3.4 Applied Loss Functions

To evaluate the optimization progress during training, a loss function has to be defined. The quality measure compares the predicted output with the ground truth (label). In the following, the two applied loss functions are described. While the (binary) cross-entropy loss is the commonly used standard loss function for convolutional neural networks [28], the focal loss is explicitly developed for datasets with severe data class imbalances [29].

Binary Cross-Entropy Loss (CE-Loss): The cross-entropy loss measures the difference between two probability distributions for a given random example (here: images). The CE-loss is, for the binary classification case, defined as [30]:

$$L(\mathbf{y}, \hat{\mathbf{y}}; \theta) = -\sum_{i=1}^{m} (y_\mathrm{i}\, log(\hat{y}_\mathrm{i}) + (1 - y_\mathrm{i}) \cdot log(1 - \hat{y}_\mathrm{i})) \\ \begin{cases} -log(1 - \hat{y}_\mathrm{i}), & \text{if } y_\mathrm{i} = 0 \\ -log(\hat{y}_\mathrm{i}), & \text{if } y_\mathrm{i} = 1 \end{cases} \tag{6}$$

where the vector θ consists of the model parameters θ_i (weights and biases), the vector y denotes the ground truth and $\hat{y}$ the predicted output. The index m refers to the number of samples.

Focal Loss (FL): The focal loss adds a modulating factor to the standard CE-loss. This allows the model to focus more on learning difficult examples and reducing the influence on the loss of easy classifiable examples [29]. Equation 7 represents a rewritten version of the (binary) cross-entropy loss defined in Eq. 6.

$$\mathrm{CE}(p_\mathrm{t}) = -log(p_\mathrm{t}) \tag{7}$$

where p_t is, for notational reasons, defined in Eq. 8:

$$p_\mathrm{t} = \begin{cases} p, & \text{if y= 1} \\ 1 - p, & \text{otherwise} \end{cases} \tag{8}$$

Equation 9 shows the novel focal loss (FL), including two new parameters, $\alpha \in [0, 1]$ and γ. This is a classic weighting factor to address the data imbalance, i.e. the importance of positive or negative samples. Due to notational reasons, α_t is defined as analogues to p_t. α_t can be used as a hyperparameter or is, for example, set to the inverse class frequency. The focusing parameter γ is used to define how easily classifiable samples are down-weighted. Those samples then have a smaller influence on the loss and more focus is set on difficult, misclassified samples. With a γ of 0, the focal loss is equivalent to the CE-loss [29].

$$\mathrm{FL}(p_t) = -\alpha_\mathrm{t}(1 - p_\mathrm{t})^\gamma log(p_\mathrm{t}) \tag{9}$$

3.5 Dataset of the Thermographic Images

The dataset was acquired utilizing pulsed induction thermography and an infrared camera. The images were labeled by NDT inspection specialists into two classes, crack-free and crack-containing images. A crack-containing image can contain one or several defective spots. The 16-bit grayscale images with dimensions of 512×640 pixels represent phase images. Such an image is shown in Fig. 3.

The datasets of both models are split into 3 parts, namely the training set, validation set, and test dataset (see Fig. 4). While the training set is used to train the model, the validation set is used to improve the model's performance during train-

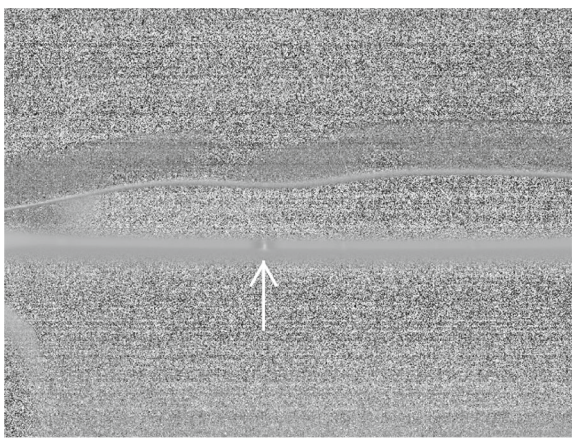

Fig. 3 A grayscale image representing one-third of a turbine blade trailing edge. This particular turbine blade edge image contains seven cracks, but only one crack of the total seven is to some extent recognizable (bright spot on the turbine blade edge). Therefore, a false-color representation is used for visualization. The thin curved line above the turbine blade edge is the induction wire

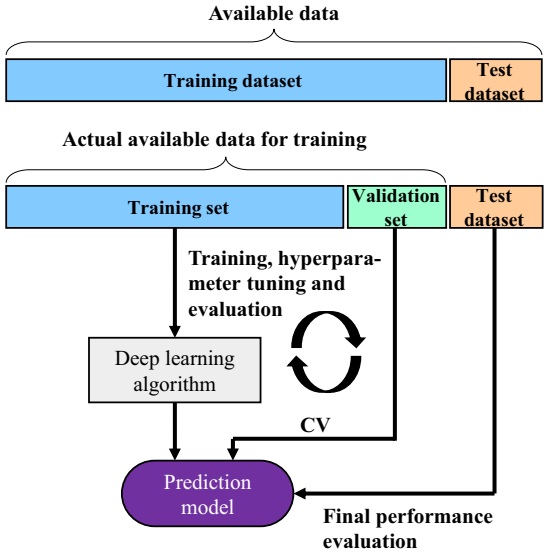

Fig. 4 Dataset splitting

Table 1 Large image dataset (512 × 640)

Set	Crack-free	Crack
Training dataset	540	47
Test dataset	60	5

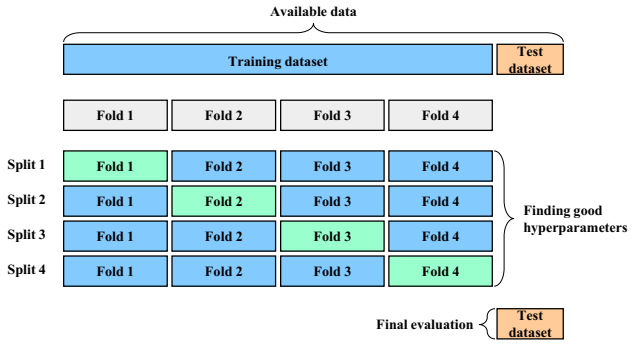

Fig. 5 K-fold cross-validation: Here, the training dataset is split in $k = 4$ folds. In each iteration step a different fold is used as a validation set (green fold), while the other three folds represent the training set (blue folds)

ing. This statistical method for evaluating machine learning models is known as cross-validation (CV). Based on the CV results (i.e. metric values and visualized learning curves), one can adjust the model's hyperparameters or make changes to the model architecture. After the model development, a final model evaluation is performed on the remaining test dataset, which was never part of the training process [25].

The following two paragraphs describe the dataset for each of the two developed models, which namely are the Large Image Model (LIM) and Small Image Model (SIM), in detail. The main difference of both models is that the SIM dataset consists of significantly smaller images. Those images are cropped by a self-developed automatic image patch cropping algorithm in combination with the known crack positions. The SIM dataset is built-up from dataset of the LIM.

Large Image Model (LIM) Dataset

The images with a size of 512 × 640 pixels were used as the input images to the large image model (LIM). The thermographic dataset consisted of 600 crack-free images and 52 crack images. In 21 of the total 52 crack-containing images, the horizontal position of each crack along the trailing edge of the turbine blade is further labeled. Approximately 90% of the data is used for model training (training dataset) and 10% of the images are held out for final testing (test dataset) (see Table 1). For the weight decay grid search and the learning rate finder, the training dataset is further split by a 75/25% ratio into a training and validation set.

In the actual model training however an extended version of the CV, the so-called stratified k-fold CV, is used. This is because the LIM dataset is characterized by a relatively large data imbalance of 11.5 and contains only very few samples of the decisive class (crack class). While data imbalance with a sufficient number of samples in all classes is not much of an issue, the combination of data imbalance and a small number of samples of the important class poses a major challenge for model development and training [31].

In k-fold CV, the training dataset is split into k folds. The process of folding is shown exemplarily in Fig. 5. While *k-1* folds are used for model training, the one fold is used for validation. The process is then repeated k times, resulting in k models with corresponding performance. The final model, therefore, represents an accurate estimate of the model's average performance [25].

The key advantage of the k-fold CV compared to the standard CV is that the resulting averaged model is based on distinct and independent folds. This leads to a model that is much less sensitive to the training dataset's subdivision into the training and validation set. K-fold CV also ensures that each sample occurs in the training set and validation set. Therefore, it is no longer possible for samples containing important information or features to appear exclusively in the validation set but not in the training set [25]. It is further important to note that only the training set is to be sampled, not the validation set nor the test dataset.

An adaption of the standard k-fold CV approach, in the case of imbalanced datasets, is the so-called stratified k-fold CV. Besides yielding better bias and variance results, the stratified k-fold CV further ensures that each fold holds the same class proportion as the overall training dataset [30]. If the initial training dataset consists of, for example, 20% of class A and 80% of class B, then also all k folds hold a class ratio of 20 to 80%, respectively.

The issue of having data imbalance together with very few samples of the decisive class (class crack), is further addressed by an additional data-level strategy, namely over-sampling. Data over-sampling involves random duplication of samples of the underrepresented class, which can improve a model's performance effectively. The under- and over-represented class are also referred to as minority and majority classes [31]. The over-sampling of the LIM dataset is performed by a batch-weighted random sampler. Hereby class weights, based on the frequency of samples per class of a batch, are calculated. Then, the minority class samples are duplicated such that each batch afterwards contains an equal number of classes [32]. When combining (stratified) k-fold CV and over-sampling it is important, to first split the training dataset and then perform the over-sampling and not vice versa.

Despite the fact, that all models are trained on grayscale images, it is still helpful to visualize the grayscale images by means of a false-color representation. The false-color representation firstly enables one to identify all cracks along the edges and secondly allows one to see the cracks more clearly, as shown in Fig. 6 compared to the same image shown in Fig. 3. This visualization, on the one hand is used to investigate the impact of different data augmentation operations and is further especially used for the build-up of the SIM dataset, which will be described in detail in the next paragraph. In order to be able to determine, whether there is actually a crack in an attempted crack-crop, the approximate position in pixels of the crack along the edge must be obtained beforehand. This manual position measurement of the cracks is performed on the basis of false-color visualized images. The acquired approximate crack positions are later used for ensuring that each cropped crack image contains a complete and not only partially crack and additionally ensures that a crack is not too close to one of the image borders. As all gray values of the 16-bit images are only in the range from 0 to 3000, specific color limits have to be further chosen. For optimal visualization of all cracks in an image and particularly for almost not visible cracks, the specific color limits would have to be calculated for each image individually. For reasons of simplification, therefore, an approximate range of color limits was identified with which all cracks could be visualized sufficiently enough. The color limit range was identified by analyzing the 21 crack image samples with known crack positions in millimeters, provided by MTU Aero Engines AG NDT inspections specialists. These crack image samples incorporated a total of 50 different cracks.

Small Image Model (SIM) Dataset

Complementary to the dataset of images with the size of 512 × 640 pixels, a sub-dataset with significantly smaller image patches of 64 × 64 pixels was assembled for the second model. On one hand, this should increase the number of available crack images for model training and, on the other hand, maximize the percentage of pixels representing a crack in the crack images.

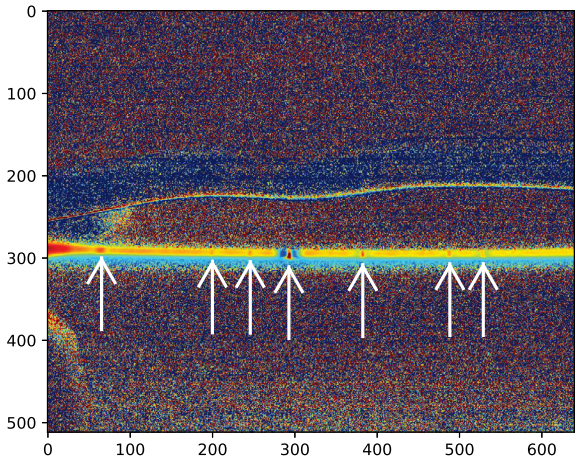

Fig. 6 False-color representation of the grayscale image from Fig. 3. Now, one can clearly see all (7) cracks in this turbine blade trailing edge. The red spots along the blade edge represent the cracks. The cracks range from very distinct appearances to only hardly recognizable ones. Furthermore, above the blade edge, the induction wire is now also more clearly visible

The SIM dataset was created by random cropping of the initial LIM dataset. All large crack-free images of the LIM dataset can be used for cropping of small crack-free image patches, while the cropping of crack image patches is constrained. For cropping small crack image patches, only the 21 crack images (of the total 52 crack images of the LIM dataset) with known crack positions can be utilized. The process of random cropping along the turbine blade trailing edge is shortly described in the following and is visualized in Fig. 7. After detecting the approximate vertical position of the turbine blade edge in each 512 × 640 pixel image, based on identifying the image row with the lowest pixel value variance, the image is then trimmed to the height of 64 pixels. A separately developed automatic patch cropping algorithm then, for crack-free images, *performs a defined number of crops* of crack-free image patches and for crack images *attempts a defined number of crops* of crack-containing image patches based on the known crack positions in the crack images. An example of the resulting 64 × 64 pixel image patches can be seen in Fig. 8.

The algorithm also ensures that when cropping crack image patches, the crack center is at least 15 pixels away from the left and right image border. This cross-checking guarantees, that the currently attempted crack to be cropped, is always completely captured and not only partially. The margin of 15 pixels corresponds to about half of the horizontal dimension of the largest existing crack, which is about 34 pixels wide.

By performing cropping of 90% smaller image patches, compared to the large images of the LIM dataset, the pixel

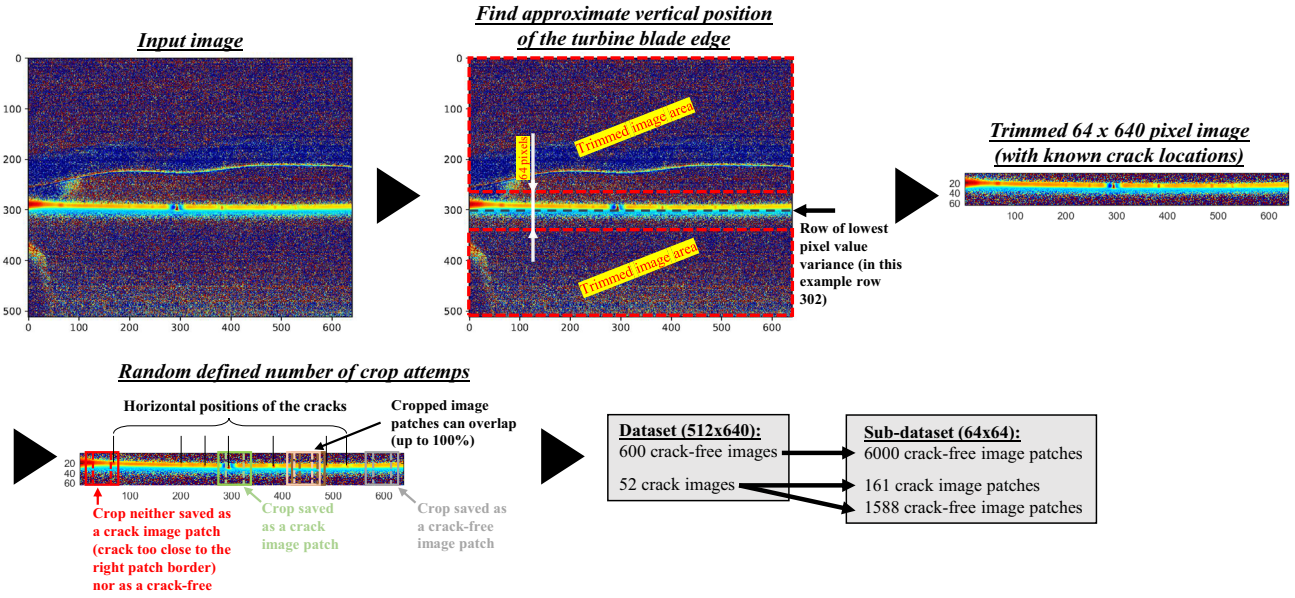

Fig. 7 Steps of the cropping procedure to build up a sub-dataset of significantly smaller image patches

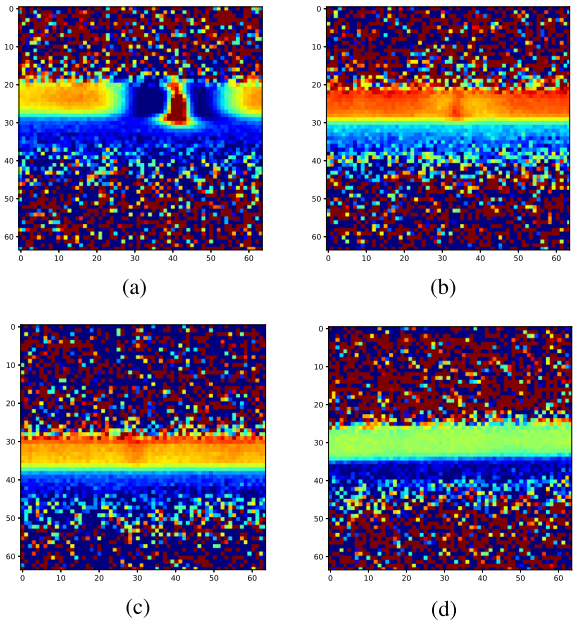

Fig. 8 Random cropped 64 × 64 pixel image patches along the turbine blade edge: **a–c** each contain a crack, while **d** is a crack-free image patch

percentage of cracks in crack images can be increased significantly, from much less than 0.1% up to 1.2 to 6.6%. In addition, the number of crack images is tripled. The SIM dataset finally contained 161 crack image patches and 7588 crack-free image patches of which 6000 images were cropped from the large crack-free images. The remaining 1588 crack-free image patches are cropped from crack-free areas in the large crack images. For crack-free images, the number of crop attempts was set to 10, while for cropping crack image patches the number of random attempts for each crack spot in each image was set to 60.

It must be further noted that to avoid overfitting, all retained cropped crack image patches from a given large image either belonged to the training set, validation set or test dataset. Random cropping along the turbine blade edge can result in crack image patches that are very similar to each other. Therefore, it is necessary to keep datasets strictly separate to evaluate the performance of the model during training and afterwards. Cropping to an image patch smaller than 64 × 64 pixels would not have any advantage, as the cracks has a certain spatial extent, which have to fit into the image patches.

The splitting of the SIM dataset into a training set, validation set, and test dataset turned out to be extremely challenging due to the small number of crack images where cropping is possible and the constraint that crack image patches of a given large crack image cannot be split between different datasets. At the same time, additional attention must be given to ensure that a disproportionately large number of crack image patches is not used for validation or final testing. It was possible to build up a representative validation set of approximately 10% of the training dataset (see Table 2) that included very distinct but also very weak cracks and most other crack shapes. However, the assembled test dataset was not representative and was therefore not used further.

Due to the constraint of keeping the dataset separated, the implementation of k-fold CV for the SIM is not possible. The data imbalance was instead addressed by using a focal loss function instead of the standard CE-loss. The focal loss

Table 2 Small image dataset (64 × 64)

Set	Crack-free	Crack
Train	6489	141
Validation	720	14
(Test)	(379)	(6)

is described in Sect. 3.4. Data over-sampling of the minority class was performed as well.

4 Models and Results

In Sects. 4.1 and 4.2, the two best-performing developed CNN models are presented. The main difference between the models is the image dimensions. While the first model is trained on the images of the dataset with dimensions of 512×640 pixels, the second model is trained on a sub-dataset of 64×64 pixel image patches. Due to the fluctuating behavior of the learning curves (loss curves) during training, often used techniques to prevent overfitting, such as early stopping, are not useful. Instead, the model saving is done by tracking the F_2-score over each epoch and if the value increases, the current best F_2-score model is overwritten.

4.1 Large Image Model (LIM)

In the first step, the 512 × 640 pixel images are stretched to squared dimensions of 640 × 640 pixels, as square images are preferable, and almost all state-of-the-art architectures are trained on such image dimension relations. Since many state-of-the-art neural network architectures are optimized for smaller dimensions (e.g. 224 × 224 pixels), squeezing the images to such a format was also attempted. However, by analyzing the squeezed images before training, it was already suspected that the loss of information through this squeezing would be too large. A test finally confirmed this assumption and squeezing the images was, therefore, not pursued further.

The LIM utilizes (stratified) k-fold CV, CE-loss, data augmentation, and over-sampling of the minority class. Data augmentation operations include random image rotation (up to $\pm 15°$), contrast adjustment (scale: 0.85 to 1.15), and vertical flipping, all applied with a probability of 10%, as well as horizontal image flipping with a probability of 50%. The model is further trained on a relatively small batch size of 8, which produced significantly better results than a batch size of 32 or 64. The reason for better results with a rather low batch size arises from the fact, that the weights per epoch are updated more frequently during the optimization process. By incorporating more noise than large batches, small batch sizes offer a regularizing effect which improves the generalization performance [33,34]. This noise, however, also prevents from fully converging to the minimum and instead causes a fluctuation around the minimum at one point. The magnitude of the fluctuations depends on the noisiness of the batches [33]. A further advantage of small batch sizes is the reduced amount of GPU memory that is needed for optimization.

To choose a reasonable architecture, different on ImageNet pre-trained CNNs were compared with each other. First a variety of models were trained for 8 epochs and 3 folds as shown in Table 3. As only the last layer (fully-connected layer), responsible for adjusting the number of output classes, of each pre-trained model is re-trained, the rather short training time of 8 epochs is also sufficient for more complex (deeper) networks to be compared [35].

Subsequently, the three best architectures with the highest achieved F_2-scores, were trained again, but with more epochs (30) and more folds (4). The results are given in Table 4. As the number of folds has been changed, the metric values of Tables 3 and 4 cannot be compared with each other.

The final model training was done using a learning rate range from 1e−4 to 1e−2, a weight decay of 1e−4, and a full re-training of all parameters of the ResNet-18 network. The training data was split into 8 stratified folds of which each was trained for 250 epochs (total training time of 38 h). Figure 9 shows the loss curves averaged over all folds.

As can be seen in Fig. 9, both loss curves are, after 250 epochs, still in a slightly decreasing trend. However, the averaged F_2-score begins after approximately 50–75 epochs to slowly decrease again, which results from a decreasing recall and vice versa an increasing precision. The average epoch from all 8 folds, where the model has its largest F_2-score, was found at 61 (see Fig. 10).

The confusion matrix in Fig. 11 shows the result achieved on the validation set. The final recall and precision value equal 0.60, resulting in an F_2-score of 0.60 (see Table 5). From a class-specific viewpoint, these values represent the values of the crack class. Looking at the class-specific value also reveals that the model works almost perfectly predicting on crack-free images (recall and precision equal both to 0.97).

As the turbine blade edges and possible cracks only take up a small proportion in the large images, the validation results are further checked with a gradient-weighted class activation mapping (Grad-CAM) implementation. In Fig. 12 the Grad-CAM heatmaps for some validation images can be seen. Clearly, the classifier focuses on horizontally short to medium-length narrow areas located about one half of the image height. The reddish highlighted areas, which are the image regions that are decisive for the model's classification decision, coarsely coincide with the turbine blade edges in these images. Therefore, it can be deduced that the correct image areas are considered for classification decision.

Table 3 Used settings and implementation: a batch size of 8, stratified 3-fold CV + automatic oversampling, 8 epochs, CE loss, a LR of 1e−1, weight decay of 1e−8, ImageNet stats normalization, and re-training only the last layer group of the pre-trained architectures

Architecture	Confusion matrix [TP, FP; FN, TN]	Accuracy [%]	Recall	Precision	Fbeta-Score (β = 2)
Resnet18	[9, 4; 3, 95]	62.12	0.69	0.75	0.70
Resnet34	[10, 3; 17, 81]	76.90	0.77	0.37	0.63
Resnet50	[7, 6; 10, 88]	61.94	0.54	0.41	0.51
Resnet101	[8, 5; 8, 90]	83.57	0.61	0.50	0.58
Resnet152	[12, 1; 30, 68]	83.51	0.92	0.29	0.64
Squeezenet1_0	[6, 7; 15, 83]	83.52	0.46	0.29	0.41
Squeezenet1_1	[9, 4; 18, 80]	83.83	0.69	0.33	0.57
Densenet121	[11, 2; 9, 89]	89.22	0.85	0.55	0.77
Densenet161	[10, 3; 7, 91]	88.93	0.77	0.59	0.73
Densenet169	[9, 4; 9, 89]	84.12	0.69	0.50	0.64
Densenet201	[12, 1; 24, 74]	85.32	0.92	0.33	0.68
VGG16_bn	[8, 5; 4, 94]	89.54	0.62	0.67	0.63
VGG19_bn	[10, 3; 52, 46]	50.19	0.77	0.16	0.44
Alexnet	[9, 4; 38, 60]	72.46	0.69	0.19	0.45
InceptionResnetv2	[9, 4; 19, 79]	83.21	0.69	0.32	0.56
Inception v3	[12, 1; 29, 69]	82.01	0.92	0.29	0.64
Inception v4	[9, 4; 16, 82]	84.74	0.69	0.36	0.58
ResNeXt 101_32x4d	[9, 4; 14, 84]	79.31	0.69	0.39	0.60
ResNeXt 101_64x4d*	[7, 6; 3, 95]	44.95	0.54	0.70	0.57
Se_Resnet50	[8, 5; 8, 90]	77.89	0.62	0.50	0.59
Se_Resnet101	[11, 2; 42, 56]	56.40	0.85	0.21	0.53
Se_Resnext 50_32x4d	[10, 3; 50, 48]	77.22	0.77	0.17	0.45
Senet154	[9, 4; 18, 80]	58.31	0.69	0.33	0.57

* with batch size = 4, due to GPU memory limitations
Decisive metric: F_2-score (compare subsection with Sect. 3.3)

Table 4 Used settings and implementation: a batch size of 8, stratified 4-fold CV + automatic oversampling, 30 epochs, CE loss, a LR of 1e−1, weight decay of 1e−8, ImageNet stats initialization, and re-training only the last layer group of the pre-trained architectures

Architecture	Confusion matrix [TP, FP; FN, TN]	Accuracy [%]	Recall	Precision	Fbeta-Score (β = 2)
Resnet18	[9, 1; 10, 63]	73.53	0.90	0.47	0.76
Densenet121	[5, 5; 9, 64]	88.62	0.50	0.36	0.46
Densenet161	[4, 6; 14, 59]	78.12	0.40	0.22	0.34

Decisive metric: F_2-score (compare with Sect. 3.3)

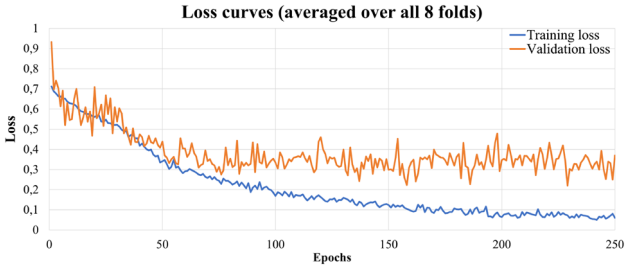

Fig. 9 Averaged loss curves over all 8 stratified folds of the LIM

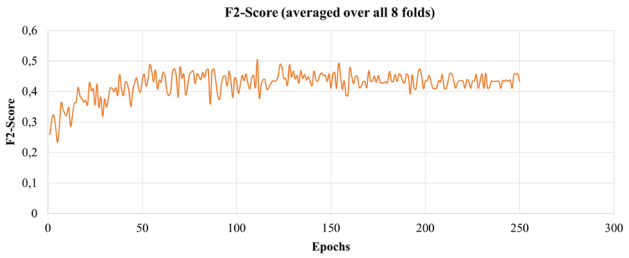

Fig. 10 F_2-score of the LIM

The performance achieved on the held-out test dataset, containing a total of 65 images (≈ 10% of the total dataset), of which 5 are crack images, is shown in Fig. 13. The confusion matrix equals to a recall of 0.60, precision of 0.20, and an

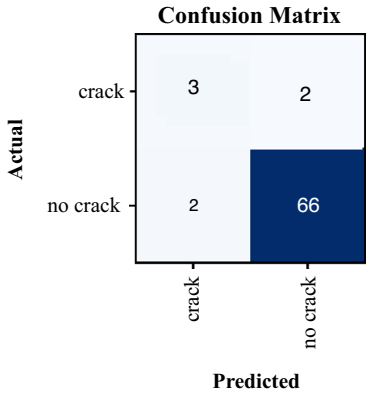

Fig. 11 Confusion matrix of the validation set images for the LIM model

Table 5 LIM validation set results (512 × 640)

Recall	Precision	F_2-score
0.60	0.60	0.60

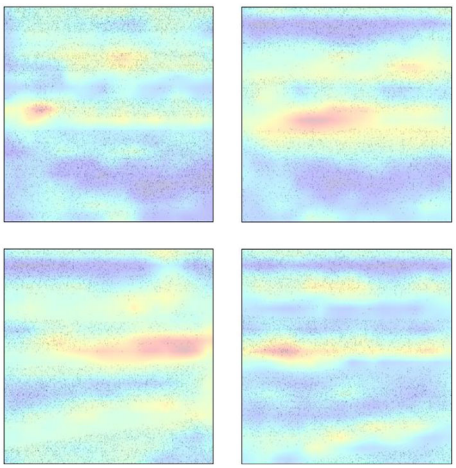

Fig. 12 Grad-CAM visualization of some LIM validation images

F_2-score of 0.43 (see Table 6). Considering the negative class (crack-free class) a recall of 0.80 and a precision of 0.96 is achieved.

When comparing the confusion matrix to the one achieved during training (validation set) an increase of false positives ($\widehat{=}$ increase of precision) can be identified. However, the most important metric value for the task at hand, the recall value of class crack, did not decrease. The performance reduction was mainly due to more crack-free images being classifed as crack images, i.e. false alarms. The degradation may result from the random choice of the only 5 crack images selected for final testing. For example, if all 5 crack images used in the test dataset only contain very weak cracks, distinguishing between crack-free and crack images is very difficult. Therefore, the results achieved on the validation set are significantly more meaningful.

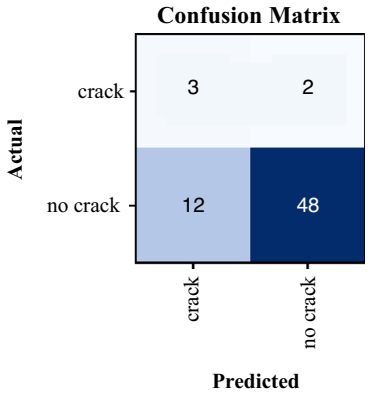

Fig. 13 Confusion matrix of the test set images for the LIM model

Table 6 LIM test set results (512 × 640)

Recall	Precision	F_2-score
0.60	0.20	0.43

The model's overall performance is on the one hand limited by the low amount of crack images and on the other hand by the fact that in the large images the cracks are extremely small (smaller than 0.1% of all image pixels) making it difficult for the classifier. This consideration was the reason for developing another model that used significantly smaller images patches to increase the number of pixels representing a crack in a crack image.

4.2 Small Image Model (SIM)

Based on the results of the LIM, a second model was developed that worked with significantly smaller images. Since the images of this sub-dataset were much smaller in their dimensions, the choice of larger batch sizes did not cause any problems regarding the GPU memory. Also, the issue of disproportionate over-sampling, when using larger batch sizes, was not as problematic as in the LIM. A batch size of 256 was finally chosen. As a data-level imbalance handling strategy again a batch weight random sampler was used for over-sampling. However, as already mentioned in Sect. 3.5, the use of k-fold CV was not possible as the image patches of the different datasets have to be strictly kept separate. Also, a focal loss instead of the standard CE-loss was implemented. As the data is already over-sampled, the weighting parameter α of the FL was set to 0.5 ($\widehat{=}$ no specific class weighting) and more focus was placed on difficult classification samples by setting the focal loss focusing factor γ to 2.

For the SIM a ResNet-18 architecture (no pre-training), data augmentation, a learning rate range of 3e−5 to 1e−3 and a weight decay value of 1e−8 was chosen. Data augmentation operations again include random image rotation (up to ±15°), contrast adjustment (scale: 0.85 to 1.15), and

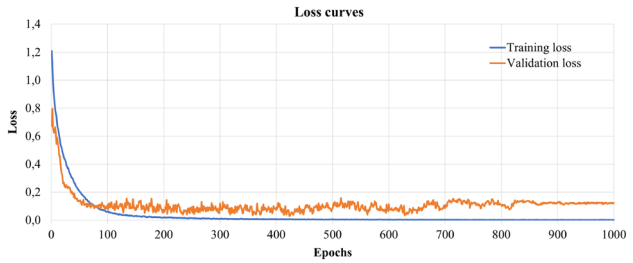

Fig. 14 Loss curves of the SIM

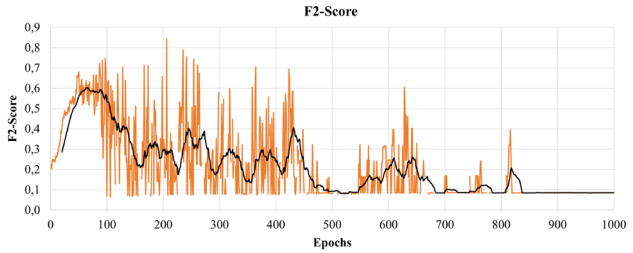

Fig. 15 F_2-score of the SIM with an additional moving average for an interval of 20 epochs for better visualization

Table 7 SIM validation set results (64 × 64)

Recall	Precision	F_2-score
0.93	0.62	0.85

vertical flipping, all applied with a probability of 10%, as well as horizontal image flipping with a probability of 50%. The model was trained for 1000 epochs and the model was saved at the epoch with the highest F_2-score (see Fig. 14).

The best-performing models (based on the F_2-score) are generally found within the first 100 epochs. However, due to the fluctuating behavior of the metrics, good models also can be found sporadically beyond 100 epochs, while the general overall performance is already in a decreasing trend due to overfitting. In this case, the best-performing model was found at epoch 208 (see Fig. 15).

The confusion matrix in Fig. 16 shows the results achieved on the validation set. The final recall value equals 0.93 and the precision 0.62, resulting in an F_2-score of 0.85 (see Table 7). From a class-specific viewpoint, these values represent the values of the crack class. Looking at those values also reveals that the model works almost perfectly predicting crack-free images (recall of 0.99 and precision of 1.00).

It is again noted, that due to the functioning of the image patch cropping procedure and the very limited amount of crack images of which crack image patches can be cropped, it was not possible to build up a representative test set.

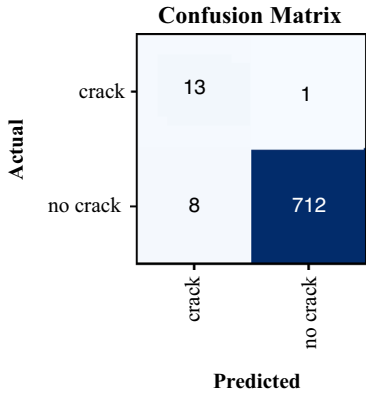

Fig. 16 Confusion matrix for the validation set images for the SIM model

5 Conclusion

After both the LIM and SIM models have been discussed, in this section a conclusion is drawn. The main issue of having large images with extremely small decisive details could be reduced by cropping smaller image patches of 64 × 64 pixels. Additionally, by performing random cropping the rather small number of crack images of the LIM dataset could be tripled in the SIM dataset, despite the fact that of only 21 original crack images (from a total of 52) the cracks positions were known and subsequently could be cropped from.

The resulting performance (on the validation set) of the LIM was only slightly better at correctly predicting crack images than random guessing (recall and precision of 0.60). In comparison, the SIM working with much smaller image patches achieves a recall of 0.93, which is significantly higher. The achieved precision value was 0.62, as more weighting was put on finding a good recall. Nevertheless, this rather low value is not too problematic, as it is much more important to have less false negative ($\widehat{=}$ high recall) than false positive ($\widehat{=}$ high precision) predictions. Furthermore, it must be remembered that the SIM cannot utilize k-fold CV , which generally (and especially for small datasets) can greatly improve a model's generalization performance. In addition, for both models, misclassified images of the validation set were analyzed for any labeling errors. However, no obvious mistakes could be found.

6 Outlook

In this section, an outlook for future work is given. This includes the potential for improvement of the developed AI pipeline, the data preparation/dataset creation, and also addresses further applications. It has must be noted that both models were optimized and trained on a very small dataset.

6.1 Improvements for Developed Classification Models

Image Dimension

The results achieved from both models indicate that the model working with small image patches should be the focus of further improvement and development efforts. However, this does not mean that the model, working with the images of 512 × 640 pixel, cannot be further improved with additional image data for training. Since the decisive details are extremely small in the large images, it can be assumed that a higher than the average number of training images would be required to improve performance, especially for the correct identification of crack images. Working with large images also results in significantly longer training times and higher GPU memory requirements.

Regarding the image labeling, the labeling of images used for the LIM is significantly less expensive, as one crack found in an image is already sufficient to label the image as a crack image. In comparison, the process of image labeling, from which the algorithm can crop, is much more time-consuming since all cracks in an image must be identified and labeled to ensure error-free image patch cropping. Additional labeling also posses an increased potential for error in the form of misclassifications. Further, the use k-fold CV is not possible when working with random cropped image patches.

Automatic Patch Cropping Algorithm

- The random cropping algorithm can be extended to provide a selection option whether multiple cracks are allowed in an crack image patch or just one. This can be the case if two cracks lie very close to each other. For a small dataset, such as the one at hand, the occurrence of two cracks in a single crack image patch can be confusing for the classification decision of the CNN model, since the vast majority of crack image patches contain only one defect. With an increasing dataset size, this becomes less of an issue, since there are then enough crack image patches that contain two (or more) cracks on which the model can be trained.
- If there is no extreme need to have more crack images for model training, it is preferable to implement and use an defined cropping approach of smaller image patches rather than a random approach. The reason for this is that during AI inference a turbine blade image is classified by sub-classifying 19 defined image patches of 64 × 64 pixels along the turbine blade trailing edge (see Fig. 17). For an optimal prediction model it is recommended to use images for training that are generated in the same way as in the later model deployment stage.

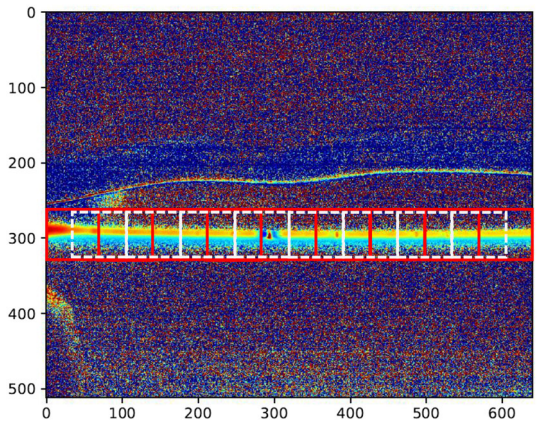

Fig. 17 Defined cropping with 19 partly overlapping small image patches which cover the entire turbine blade trailing edge of a large image

Multiclass Classification

A reasonable enhancement of the presented CNN models would be to extend the binary classification to more than the two classes. The prerequisite for this, however, is the availability of a significantly larger dataset than the one provided for this work. In the following, two possibilities for a multiclass classification are provided.

- Crack-specific classification:
  A subdivision of the class crack into a crack class of weak cracks and a crack class containing significant cracks is thinkable. NDT inspections specialists already distinguish between so-called SR-cracks and SX-cracks. While SR-cracks are characterized by a vertical (transverse to the blade edge) elongated extension and are often very distinct and tend to be easier to recognize, SX-cracks have a more circular shape and are more difficult to identify. SX-cracks are further also considered as conspicuities (a.o. scratches, surface damages).
- Crack number-specific classification:
  A subdivision of the class crack into classes with different numbers of cracks could also provide insightful information. When the number of defects of classified crack images is known, it would be possible to draw conclusions about particularly favorable (or unfavorable) manufacturing parameters or any other possible crack-causing circumstances.

6.2 Crack Detection and Localization

The developed approach provides the basis for an object detection algorithm or even an object segmentation approach for identifying cracks in turbine blade images. While the positions of cracks along the turbine blade trailing edge are predictable by an object detection approach, a further more

advanced crack segmentation model would even allow a quantification of the spatial extent of cracks. However, for such an approach, the training images need to be not only labeled but also segmented. Before that, a precise definition of what a crack is and which pixels in n thermographic crack image belong to a crack, is needed.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Panizza, A., Stefanek, S.T., Melacci, S., Veneri, G., Gori, M.: Learning to identify drilling defects in turbine blades with single stage detectors. Università degli Studi di Siena, In: Conference: Machine Learning for Engineering Modeling, Simulation, and Design Workshop at Neural Information Processing Systems (2020)
2. Schlobohm, J., Bruchwald, O., Frckowiak, W., Li, Y., Kaestner, M., Poesch, A.: Advanced characterization techniques for turbine blade wear and damage. Proc. CIRP **59**, 83–88 (2017). https://doi.org/10.1016/j.procir.2016.09.005
3. Reimche, W., Bach, F.-W., Boehm, V., Bruchwald, O., Frackowiak, W.: Nachweis von lokalen Schaedigungen an Hochleistungsbauteilen mit Hochfrequenz Wirbelstromtechniken und Induktions-Thermografie. In: DACH-Jahrestagung 2021 in Graz (2012)
4. Vrana, J., Goldammer, M.: Induction and conduction thermography: from the basics to application. In: DGZfP Thermographie-Kolloquium 2017 in Berlin (2017)
5. Vrana, J.: Grundlagen und Anwendungen der aktiven Thermographie mit elektromagnetischer Anregung. Induktions- und Konduktionsthermographie. Universitaet des Saarlandes, Saarbruecken, Dissertation (2008)
6. Sackewitz, M.: Leitfaden zur Waermefluss-Thermographie. Zerstoerungsfreie Pruefung mit Bildverarbeitung. Fraunhofer-Allianz Vision, Erlangen. Stuttgart (Vision-Leitfaden) (2011)
7. Makantasis, K., Protopapadakis, E., Doulamis, A., Doulamis, N., Loupos, C.: Deep convolutional neural networks for efficient vision based tunnel inspection. In: Potolea, R. (ed.) 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). 3–5 Sept. 2015, Cluj-Napoca, 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). Cluj-Napoca, Romania, 03.09.2015–05.09.2015. Institute of Electrical and Electronics Engineers; IEEE International Conference on Intelligent Computer Communication and Processing; ICCP. Piscataway: IEEE, S. 335–342 (2015)
8. Zhang, L., Yang, F., Zhang, Y., Zhu, Y.: Road crack detection using deep convolutional neural network. In: 2016 IEEE International Conference on Image Processing (ICIP), S. 3708–3712 (2016)
9. Cha, Y.-J., Choi, W., Bueyuekoeztuerk, O.: Deep learning-based crack damage detection using convolutional neural networks. Comput. Aided Civil Infrastruct. Eng. **32**(5), 361–378 (2017). https://doi.org/10.1111/mice.12263
10. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H.: Road damage detection and classification using deep neural networks with smartphone images. Comput. Aided Civil Infrastruct. Eng. **33**(12), 1127–1141 (2018). https://doi.org/10.1111/mice.12387
11. Ni, F.T., Zhang, J., Chen, Z.Q.: Pixel-level crack delineation in images with convolutional feature fusion. Struct. Control. Health Monit. **26**(1), e2286 (2019). https://doi.org/10.1002/stc.2286
12. Khani, M.M., Vahidnia, S., Ghasemzadeh, L., Ozturk, Y.E., Yuvalaklioglu, M., Akin, S., Ure, N.K.: Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines. Struct. Health Monitor. **10**, 5 (2020). https://doi.org/10.1177/1475921719883202
13. Yang, J., Wang, W., Lin, G., Li, Q., Sun, Y., Sun, Y.: Infrared thermal imaging-based crack detection using deep learning. IEEE Access **7**, 182060–182077 (2019). https://doi.org/10.1109/ACCESS.2019.2958264
14. Soukup, D., Huber-Moerk, R.: Convolutional neural networks for steel surface defect detection from photometric stereo images. Safety and Security Department and AIT Austrian Institute of Technology GmbH (2014). https://doi.org/10.1007/978-3-319-14249-4_64
15. Xian, T., Dapeng, Z., Wenzhi, M., Xilong, L., De, X.: Automatic metallic surface defect detection and recognition with convolutional neural networks. Appl. Sci. **8**, 1575 (2018). https://doi.org/10.3390/app8091575
16. Zenzinger, G., Bamberg, J., Satzgert, W., Carl, V.: Thermographic crack detection by eddy current excitation. https://doi.org/10.1080/10589750701447920 (2007)
17. Hasenstab, A., Langmeier, A., Schönitz, A., Zöcke, C.: Aktive Thermografie mit Phasenauswertung Praktische Anwendung im Bauwesen. www.ndt.net (2008)
18. Müller, J.P., Götschel, S., Weiser, M., Maierhofer, C. (2017) Thermografie mit optimierter Anregung für die quantitative Untersuchung von Delamination in kohlenstofffaserverstärkter Kunststoffe. www.ndt.net
19. Howard, J., Gugger, S.: fastai: a layered API for deep learning. www.fast.ai (2020)
20. Smith, L.N.: Cyclical learning rates for training neural networks. U.S. Naval Research Laboratory arXiv:1506.01186 (2017)
21. Smith, L.N.: No more Pesky learning rate guessing games. U.S. Naval Research Laboratory, arXiv: 1506.01186 (2015)
22. Smith, L.N., Topin, N.: Super-convergence: very fast training of neural networks using large learning rates. arXiv:1708.07120 (2017)
23. Gugger, S.: Another data science student's blog—the 1cycle policy. fastAI. https://sgugger.github.io/the-1cycle-policy.html. Accessed 02 March 2021 (2018)
24. NVIDIA (2020) Training with mixed precision. User Guide. https://docs.nvidia.com
25. Raschka, S.: Python machine learning. Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics. Packt Publishing, Birmingham (open source Community experience distilled) (2016)

26. Sasaki, Y.: The Truth of the F-Score. University of Manchester, Manchester (2007)
27. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data sets. https://doi.org/10.1007/978-3-319-98074-4 (2018)
28. Janocha, K., Czarnecki, W.M.: On loss functions for deep neural networks in classification. arXiv:1702.05659 (2017)
29. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. Facebook AI Research (FAIR). arXiv:1708.02002 (2018)
30. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
31. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. J. Big Data **6**, 27 (2019). https://doi.org/10.1186/s40537-019-0192-5
32. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: an imperative style. High-performance deep learning library. **1912**, 01703 (2019)
33. LeCun, Y.A., Bottou, L., Orr, G.B., Mueller, K.R.: Efficient BackProp. In: Montavon, G., Orr, G.B., Mueller, K.R. (eds.) Neural Networks: Tricks of the Trade Lecture Notes in Computer Science, vol. 7700. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-35289-8_3
34. Hoffer, E., Hubara, I., Soudry, D.: Train longer, generalize better: closing the generalization gap in large batch training of neural networks. arXiv:1705.08741 (2018)
35. Bressem, K.K., Adams, L.C., Erxleben, C., Hamm, B., Niehues, S.M., Vahldiek, J.L.: Comparing different deep learning architectures for classification of chest radiographs. Sci. Rep. **10**(1), 13–590 (2020). https://doi.org/10.1038/s41598-020-70479-z