The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-025-16539-y

ORIGINAL ARTICLE

# Steel surface defects analysis with machine vision and deep learning

Karol Frydrych[1,2] · Maciej Tomczak[1] · Jarosław Jasiński[1] · Stefanos Papanikolaou[1]

Received: 8 May 2025 / Accepted: 5 September 2025


**Abstract**
Steel surface defects in both flat and long products are undesired not only from an aesthetic point of view, but also can lead to severe deterioration of material performance. Manual defect inspection is slow and costly, and thus, automatization of such processes is of interest. Several steel surface defect datasets have been made publicly available so far, and the most famous of them is the Northeastern University (NEU) surface defect database. Many research on surface defect inspection has already been conducted using this dataset, and excellent prediction capabilities were demonstrated in the open literature. More recently, this dataset was extended to account for effects that are expected to occur in real industrial scenarios, such as motion blur, non-uniform illumination, and noise. The extended dataset containing images with those modifications was also made publicly available (E-NEU). In previous papers on the subject, it was shown that using deep learning models trained on the NEU dataset to the E-NEU dataset does not necessarily lead to correct predictions. In this paper, based on the steel surface defects analysis, it is demonstrated that the performance of deep learning architectures can be effectively improved by applying image preprocessing techniques.

**Keywords** Surface defects classification · Quality control · Steel surface · Long products · Flat products

## Highlights

- Classification performance of different convolutional neural network architectures as applied to steel defect datasets is assessed.
- An effect of image perturbations on classification is studied.
- It is demonstrated that image preprocessing can considerably improve the classification results.

## 1 Introduction

Thanks to its availability and properties, various types and grades of steel constitute a major metallic material used by various industries such as civil engineering, energy (including nuclear), shipbuilding, automotive, or household appliances, to give only the most obvious examples. Steel surface defects in both flat [1] and long products [2] are an important manufacturing problem. On the one hand, they are not only detrimental for surface appearance, but also decrease resistance to corrosion and fatigue strength [3, 4]. On the other hand, surface defects cause more than 50% of total hot rolling scraps and cost hundreds of millions of dollars each year [2]. Therefore, major steel companies are involved in research devoted to their mitigation [5].

Traditionally, surface quality control was done manually by humans [6], but machine vision methods started to be used for that purpose already in the 1990s [4]. The topic of surface defects inspection (and *steel* surface defects inspection in particular) received considerable attention (cf., e.g., [7]), and was also reviewed from several points of view. Neogi, Mohanta, and Dutta [6] reviewed the literature related to steel surface inspection based on vision. This paper is very important from the point of view of practical applications, as the authors identified major challenges related to applications of vision systems. These include high speed of the moving surface (20 m/s for flat products and up to 100 m/s for long products), issues related to proper choice of illumination, camera type, time required for processing, etc. A similar topic was reviewed in [8]. The defect detection methods were divided into statistical, spectral, model, and machine learning

✉ Karol Frydrych
Karol.Frydrych@ncbj.gov.pl

1 NOMATEN Centre of Excellence, National Centre for Nuclear Research, Sołtana 7, Otwock 05-400, Poland

2 Institute of Fundamental Technological Research, Polish Academy of Sciences, Pawińskiego 5B, Warsaw 02-106, Poland

Published online: 13 September 2025

(ML) methods. Recently, vision-based steel defect detection problems have also been reviewed in [9].

Applications of ML in continuous casting of steel (a process where liquid steel is poured into the mold and then continuously withdrawn downwards) were recently summarized in [10]. The defects that can be present in such a process were divided into surface defects (transverse corner cracks, longitudinal corner cracks, transverse cracks, longitudinal facial cracks, star cracks, deep oscillation marks, pinholes and macro inclusions) and internal defects (internal corner cracks, side halfway cracks, center-line segregation, halfway cracks, nonmetallic inclusions, subsurface streaks, shrinkage cavity, diagonal cracks and porosity). The various applications of ML highlighted were: prediction of breakout, prediction of steel defects, prediction of steel quality, clogging detection, process parameter optimization, prediction of steel temperature in tundish, detection of ladle change, and detection of mold level fluctuation.

The topic of surface defects inspection (not limited to steel strips) was also recently reviewed in [11], especially from the real-time inspection applicability point of view. Deep learning based defects inspection methods were categorized into defect classification, detection, and segmentation (cf. Figure 2 in [11]). Typical challenges such as low image quality in real industrial scenarios, difficulties in obtaining enough data for training, etc., were also discussed in [11]. Concerning the required processing speed, in most of the research papers, it was assumed that the velocity of the steel strip is about 20 m/s. Liu et al. estimated that the required system processing speed at this velocity is 20 frames per second (FPS). Note, however, that in [4], it was reported that the velocity of the steel strip in modern cold rolling mill has reached 45 m/s, and wire rod velocity exceeds 130 m/s. As outlined in [12], there are several bottlenecks when it comes to providing the required processing speed when the rolling speed exceeds 20 m/s.

Another review focused on the entire machine vision system for steel surface defects detection rather than just the processing module [4]. However, even though the paper was published only recently (2023), the part describing the image processing algorithms focused on traditional methods, while paying less attention to deep learning based approaches. The review is, however, quite useful to see the big picture. For instance, Table 1 in [4] provides lists of defect types in various steel surfaces.

One of the results of defects presence is, of course, poor visual quality of the steel surface, which may be unacceptable, especially in the case of flat products. However, defects may be also detrimental from other points of view. For instance, crazing may cause rupture, inclusions affect the state of stress in steel structures, patches and pitted surface can have an effect upon the resistance to wear and corrosion [8, 13]. Surface defects can also severely decrease the fatigue resistance, cf. [14]. Defect-fatigue linkages are further discussed in Section 4.4.

As already mentioned above, currently, the defect inspection task is mostly done using machine learning (ML) approaches. Those include both "traditional" techniques like support vector machines (SVM), decision trees (DT), and artificial neural networks (ANNs), as well as deep convolutional neural networks (DCNNs). The application of the first type of approach was most often reported more than a decade ago, cf., e.g., [1, 15]. An example is the paper [5], where the efficiency of different ML tools in mill scale defects classification based on images taken in a hot rolling mill located in Cracow, Poland was investigated. The authors compared the efficacy of ANN, SVM, and DT in performing this task. Eight defect types, namely rolled-in primary scale, secondary scale, "V" scale, peeled roll scale, tiger/red scale, heavy scale, single strip scale, and bad descaling, were considered. Another paper [16] reported SVM classification of five strip steel surface defect types (roller mark, rust spot, emulsion spot, side mark, and scrape). An interesting aspect of the paper is the optimization of SVM hyperparameters using the genetic algorithm.

Despite the potential present in traditional ML techniques, their crucial disadvantage is the necessity of feature extraction and selection. The convolution operation that is inherent in DCNNs eliminates the feature extraction step. Thus, it is possible to feed the network directly with images. Both traditional ML and DCNNs have to be trained on datasets. One such dataset was published by researchers from Northeastern University (so-called NEU dataset) [17, 18] (a thorough review of available steel-related datasets can be found in [19]). The images were captured after laminar cooling using 4 CCD cameras. The gray projection algorithm was applied to remove the defect-free images. Defects were divided into 6 classes: patches (Pa), pitted surface (PS), inclusion (In), crazing (Cr), scratches (Sc), and rolled-in scale (RS), cf. Fig. 1. In the original paper reporting the dataset creation [17], the authors applied the adjacent evaluation completed local binary patterns (AECLBPs), which can be considered as an example of a traditional ML technique. However, in most of the more recent papers, the classification task was performed using DCNNs, cf, e.g., [13, 20, 21]. Note that [17] prepared two variants of their dataset [18] suited for both types of tasks, namely NEU-CLS (for classification) and NEU-DET (for defect detection). As the focus here is on classification, only the NEU-CLS variant is considered. Since the present paper does not concern defect detection or segmentation tasks, readers interested in steel surface defects detection and segmentation are referred to other articles, e.g., [4, 8, 22–25] and [26–29], respectively.

Even though some analysis of the influence of noise was performed already in [17], it was suggested in [30] that in

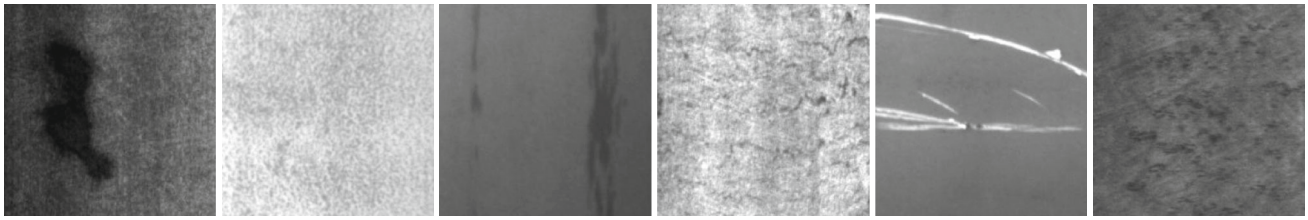



**Fig. 1** Randomly selected examples of six kinds of defects **a** patches, **b** pitted surface, **c** inclusions, **d** crazings, **e** scratches, and **f** rolled-in scale [17, 18]

real industrial conditions, the quality of classification of steel products by analytical imaging methods is influenced by various external conditions, such as motion blur, non-uniform illumination, and camera noise. In order to take into account the above-mentioned process conditions, the authors artificially modified the images from the previously mentioned NEU dataset [17] (and thus created the extended NEU dataset (E-NEU)). For each modification, they applied two intensity levels. In the case of motion blur, the length of camera motion $L_{cm}$ was set equal to 2 or 5. For non-uniform illumination, the luminance range $\alpha$ of bias fields was set to $\pm 0.4$ or $\pm 1$. The signal-to-noise ratio (SNR) in the case of camera noise was either 20 or $35db$. The results of modifications applied to images presented in Fig. 1 are shown in Fig. 2. Note that the E-NEU dataset for each modification and intensity level provides 5 random realizations and only one of them is shown in Fig. 2. Using various SqueezeNet-based CNN models, it was possible to achieve 100% accuracy of classification for the set without modification and 97.5% for the set with artificially introduced additional effects occurring in real production conditions. Therefore, especially the latter result gives very high hopes for the application of machine learning and artificial intelligence methods in steelmaking processes.

To the best of the authors' knowledge, in contrast to the vast research effort concerning the original NEU dataset, published works using the E-NEU dataset are scarce. It seems that only Nath, Chattopadhyay, and Desai in their papers [31, 32] paid attention to this challenging dataset. The approach used by these authors (histogram equalization plus adversarial training through neural structure learning) is indeed interesting and provided an accuracy of 92.4%, which is good but still worse than the accuracy reported in the paper introducing the E-NEU dataset [30]. It can be thus seen that: on the one hand, further improvement of accuracy over the E-NEU dataset is very challenging, while on the other, very limited research effort was taken to solve the problem.

Since low-quality images in real industrial settings due to noises, motion blur, or uneven illumination can make the solutions optimized for clean database images inapplicable [11, 30], the aim of this paper is to investigate ways to improve the applicability of deep learning techniques to low-quality images expected to be obtained in such conditions. In order to achieve this goal, the methodology reported in a preliminary study related to the classification of defects present in the NEU dataset [33] had to be improved. It was demonstrated that combining image preprocessing techniques with state-of-the-art deep learning methods can lead to satisfactory predictions even with the challenging E-NEU dataset.

The article is structured as follows. After this introductory section, Section 2 describes the machine vision and deep learning methods used in the present paper. Section 3 presents the results, while Section 4 provides a thorough discussion of the obtained results and puts them into a broader context. Finally, Section 5 shows the conclusions and outlines future research directions.

## 2 Methodology

### 2.1 Gray level histograms

Gray level histograms are an example of traditional or classical machine vision techniques [34, 35]. In the present paper, the histograms were generated in Wolfram Mathematica. The methodology is very simple. First, the images belonging to a given category were imported and assembled together. In the next step, the gray levels were normalized so that their values go from 0 to 1. Then, the histograms representing how many pixels corresponded to a given gray level were plotted.

### 2.2 Convolutional neural networks

In the present article, two CNN architectures are studied. The first one is the EfficientNet-v2-s [36] (as implemented in the PyTorch library [37]) that was already used in our previous paper [33] to classify the defects present in the NEU-CLS dataset. The scheme of EfficientNet-v2-s (EN2s) is shown in Fig. 3. The EfficientNet-v2-s is a part of the EfficientNet model family optimized for floating point operations per second (FLOPs) and parameter efficiency [38]. The aim of developing EN2s was to improve the speed of training with a

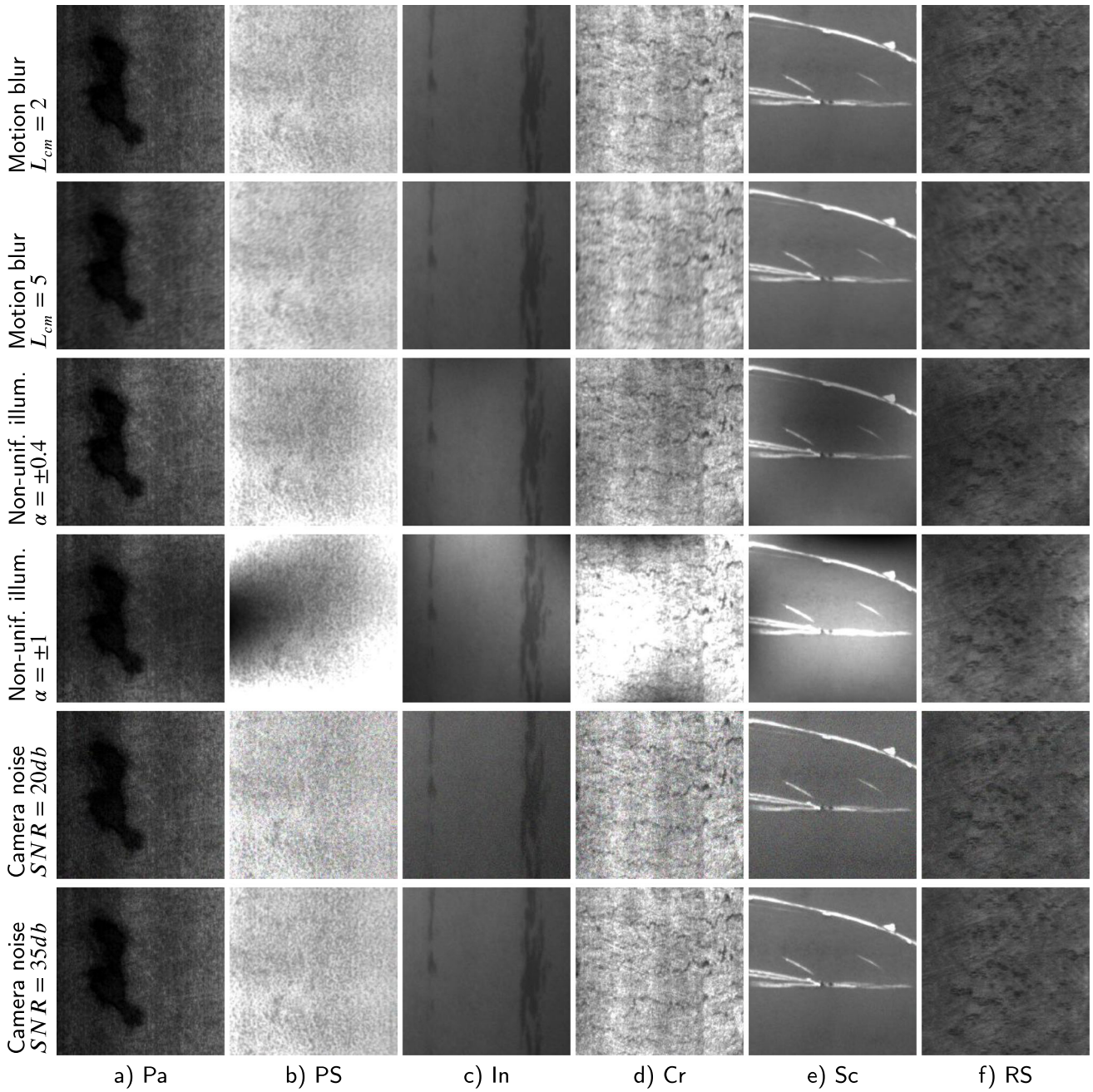



**Fig. 2** The images from the E-NEU dataset corresponding to images of defects **a** patches, **b** pitted surface, **c** inclusions, **d** crazings, **e** scratches, and **f** rolled-in scale) presented in Fig. 1. Fu et al. [30] obtained these images by applying additional modifications (motion blur, non-uniform illumination, camera noise) to the original defect images collected in [17]

similar efficiency of model parameters. EfficientNet was chosen for its state-of-the-art performance in image classification tasks with significantly fewer parameters and FLOPs compared to traditional architectures such as ResNet or VGG. It utilizes a compound scaling method that uniformly scales the network's depth, width, and resolution, enabling efficient use of model capacity. Given the fine-grained nature of steel defect detection, EfficientNet's capacity to capture rich visual features makes it highly suitable. Moreover, using EfficientNet enabled us to take advantage of the pre-trained model. As can be seen in Fig. 3, the applied architecture consists of a $3 \times 3$ convolution layer, 3 layers of $3 \times 3$ fused mobile inverted bottleneck convolution (MBConv) [39], 3 layers of $3 \times 3$ MBConv layers, $1 \times 1$ convolution layer, and finally pooling and fully connected layers. Since the weights pretrained on the ImageNet dataset were used, additional learning on

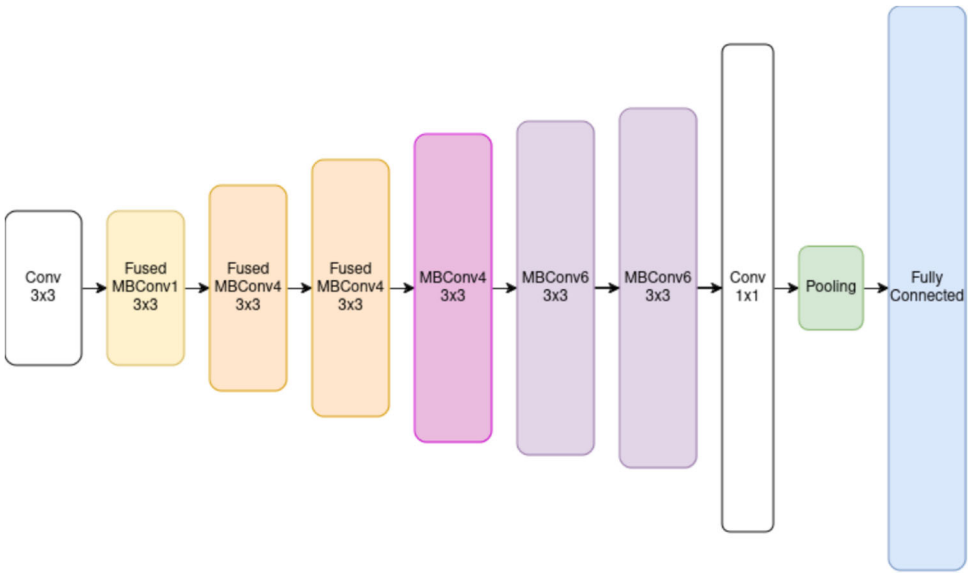



**Fig. 3** Scheme of the EfficientNet-v2-S (EN2s) [36] convolutional neural network architecture

the datasets examined in this contribution was limited to 20 epochs in order to avoid overfitting and limit training time. The effect of the number of epochs is further discussed in the Discussion Section 4.

The second architecture was based on the description provided in [30]. It was constructed by joining SqueezeNet [40, 41] with Multi-Respective Field Learning Module (SN-MRFLM) [30], cf. Fig. 4. SqueezeNet was selected for its ability to deliver AlexNet-level accuracy with a model size of less than 5 MB. Its architecture is based on fire modules, which consist of squeeze ($1 \times 1$) and expand ($1 \times 1$ and $3 \times 3$) convolutions, effectively reducing the number of parameters without sacrificing accuracy. In industrial applications—such as real-time quality control on production lines—low latency and minimal memory footprint are critical. Therefore, SqueezeNet seemed to be the best choice for our application. As the architecture was described in detail in [30], here we only summarize the key information. The pre-trained SqueezeNet 1.0 model with 8 so-called fire modules serves as a backbone network. A fire module consists of squeeze and expand layers and was described in detail elsewhere [30, 40]. The Multi-Respective Field Learning Module (MRFLM) was proposed in [30] in order to generate scale-dependent high-level features for accurate steel surface defect classification. It consists of parallel $1 \times 1$ and $3 \times 3$ convolution layers, each followed by a rectified linear unit (ReLU) activation function. The outputs are concatenated, and then average pooling and convolution are applied.

In the learning process for both architectures, the adaptive moment estimation (Adam) optimizer with learning rate equal to $10^{-5}$, weight decay equal to $10^{-6}$, and $L2$ regularization was used. Note that the hyperparameters applied in the present paper are standard and consistent with those used in the scientific literature. A detailed analysis of the hyperparameters influence was not performed. Readers interested in hyperparameters influence studies should consult other reports. Data splits were done automatically using the `train_test_split` function from scikit-learn, with a seed equal to 42. This was done both for NEU and E-NEU datasets. For network training, we did not set a random seed to any specific value; thus, they were by default taken from system randomness. The code was implemented using Pytorch

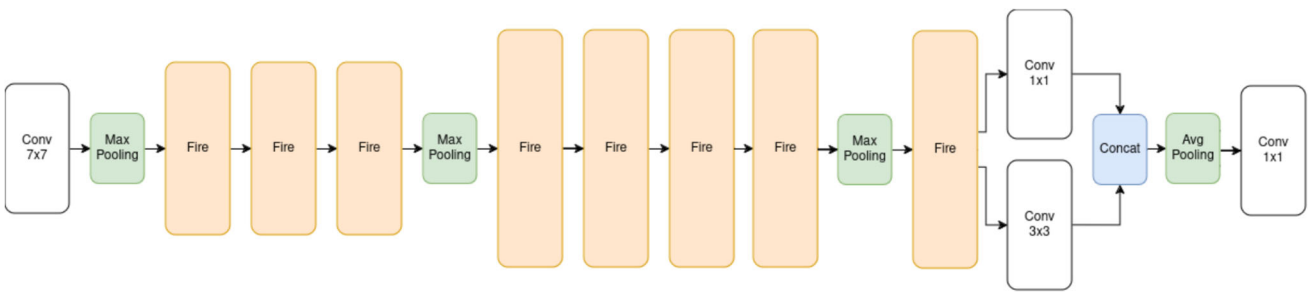



**Fig. 4** Scheme of the SN-MRFLM network obtained by joining the SqueezeNet network [40] with the Multi-Respective Field Learning Module [30]

and performed on a desktop workstation supplied with Dual Nvidia Quadro RTX4000 8GB GPU. The simulations were performed on the Ubuntu 20.04 operating system.

Standard metrics were used for the evaluation of the network's performance. Accuracy measures the proportion of correct predictions made by the model across the entire dataset. It is calculated as the ratio of true positives (TP) and true negatives (TN) to the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

Precision measures the proportion of true positive predictions among all positive predictions made by the model. It is calculated as the ratio of TP to the sum of TP and false positives (FP):

$$Precision = \frac{TP}{TP + FP}. \tag{2}$$

Recall, also known as sensitivity or true positive rate, measures the proportion of true positive predictions among all actual positive instances. It is calculated as the ratio of TP to the sum of TP and false negatives (FN):

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$F_1$ score is a metric that balances precision and recall. It is calculated as the harmonic mean of precision and recall. $F_1$ score is useful when seeking a balance between high precision and high recall, as it penalizes extreme negative values of either component:

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall}. \tag{4}$$

In order to study classification performance separately for each class, a confusion matrix (CM) can be applied. In such a matrix, each row corresponds to the actual images and each column to predictions. In the ideal case, there is always 100% of TPs on the diagonal and 0% everywhere else. However, if the classification performance is worse, the CM can give more insight into class-specific performance. For instance, Fig. 5 shows the results of image classification applied to 3 dog breeds. One can see that only half of the images showing a husky were classified correctly. Another 30% of images with husky were classified as german shepherd and 20% as labrador. On the other hand, 95% of labradors were correctly classified by the hypothetical network, and 5% of labradors were classified as german shepherds.

| | Prediction | | | |
|---|---|---|---|---|
| Ground truth | Class | husky | german shepherd | labrador |
| | husky | 50% | 30% | 20% |
| | german shepherd | 10% | 80% | 10% |
| | labrador | 0% | 5% | 95% |

**Fig. 5** Illustration of the confusion matrix concept

### 2.3 Image pre-processing

Rather than modifying the network architecture, one can study classification improvement potential with image preprocessing. To this aim, one method of image contrast enhancement and 4 noise filtering methods were studied. All the methods were imported from OpenCV (cf., e.g., [42]) Python library.

Image histograms themselves were already described in Section 2.1. Histogram equalization (HE) on the other hand, is a method to improve the image contrast by stretching out the intensity range. It proceeds by mapping a histogram of a given image to another histogram that has a wider and more uniform distribution of intensity values. The idea is schematically shown in Fig. 6.

Concerning image smoothing, whose influence on improving the classification accuracy was also studied, 4 methods were used. For each method, windows of $3 \times 3$, $5 \times 5$, and $7 \times 7$ pixels were considered. The blur (B) filter is applied by convolving an image with a normalized box filter. This operation replaces the central pixel with the mean of all the pixels under the kernel. Median blur (MB) differs from the previous one in that the median of pixels is taken instead of a mean. The next method is the Gaussian blur (GB), which differs from B in that it takes a Gaussian kernel instead of a box filter. The last method to be used is the bilateral filter BF. It is similar to GB in that it also works as a weighted average of pixels. However, in BF, also pixel intensity variation is accounted for so that edges can be preserved. For more info about BF, see [43].

## 3 Results

### 3.1 Conventional image analysis

The purpose of generating image histograms was twofold. First, to have more insight into the differences between defect images. Second, in order to see how various distortions introduced in [30] lead to information loss. Gray level histograms of each defect class are shown in Fig. 7. Looking on the histograms, one can certainly point out some key

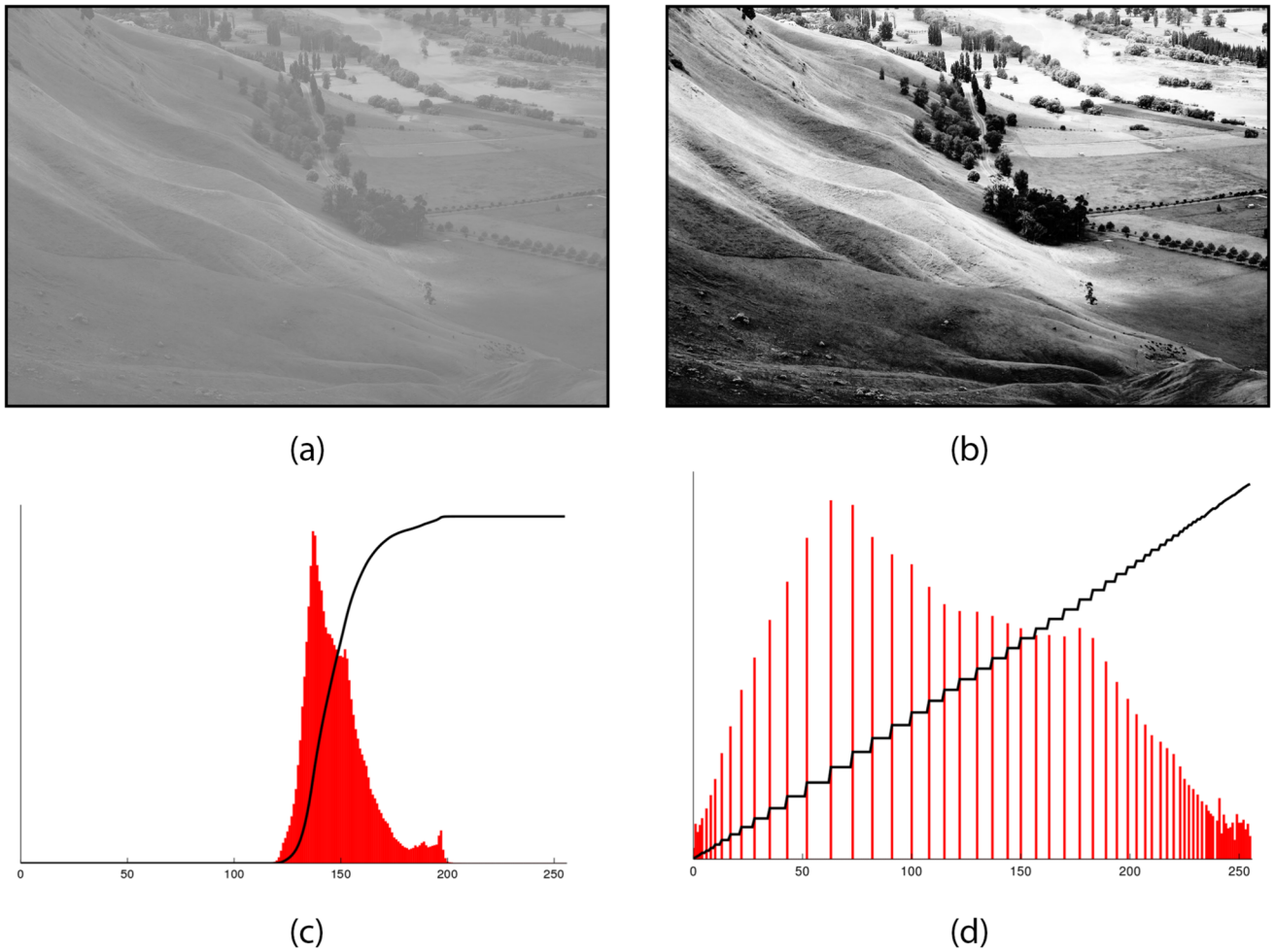



**Fig. 6** A scheme explaining the histogram equalization idea (reprinted from https://en.wikipedia.org/wiki/Histogram_equalization under CC BY 2.0 license)

differences between defect classes. For example, the histogram for rolled-in scale (Fig. 7f) is clearly different from the histogram for patches (Fig. 7a). On the other hand, the histogram for patches (Fig. 7a) is somewhat similar to the histogram for pitted surface (Fig. 7b). Namely, both have more or less flat distribution of gray levels with a sharp peak near one. Note, however, that the origin of the peak is clearly different: in the case of patches, the peak is related to big,

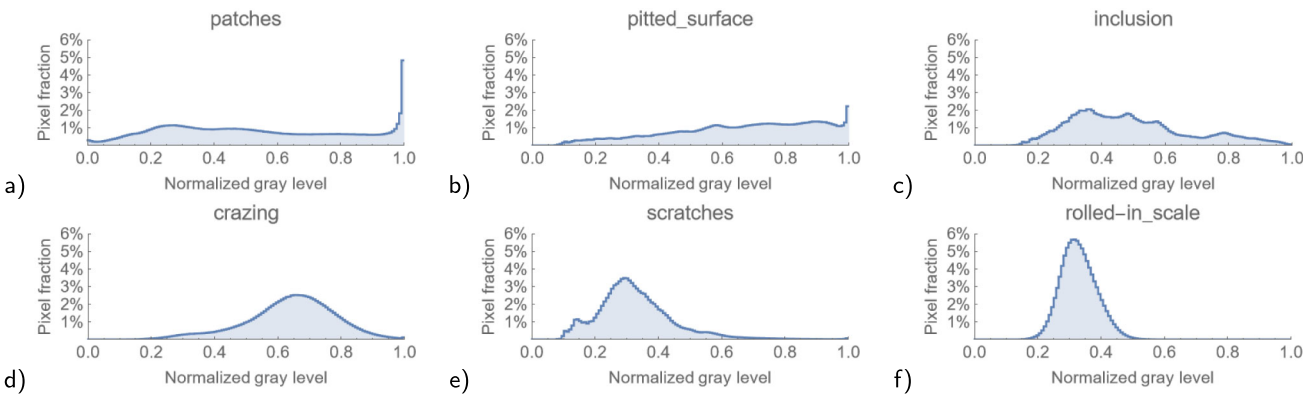



**Fig. 7** The histograms for six kinds of defects: **a** patches, **b** pitted surface, **c** inclusions, **d** crazings, **e** scratches, and **f** rolled-in scale

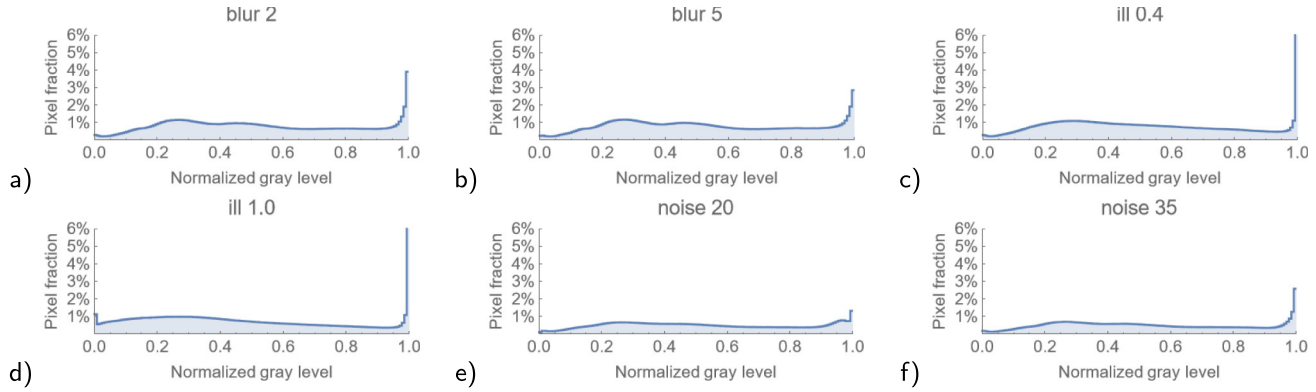



**Fig. 8** The histograms for patches—influence of additional effects: motion blur with $L_{cm}$ equal to **a** 2 and **b** 5; non-uniform illumination with $\alpha$ equal to **c** $\pm 0.4$ and **d** $\pm 1$; camera noise with SNR equal to **e** $20db$ and **f** $35db$

**Fig. 9** The histograms for rolled-in scale—influence of non-uniform illumination with $\alpha$ equal to $\pm 1$: **a** original image histogram and **b** after modification

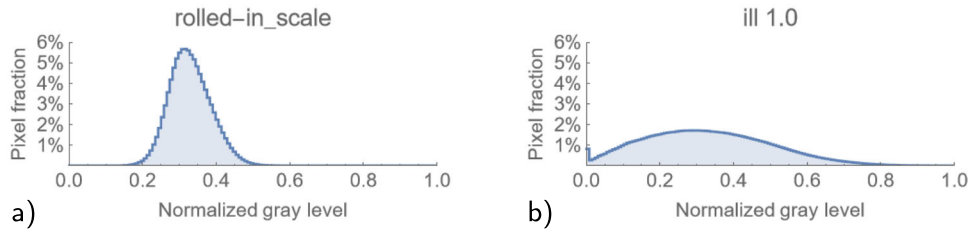
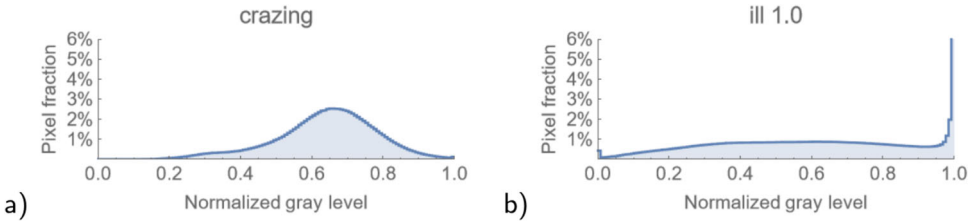



**Fig. 10** The histograms for crazings—influence of non-uniform illumination with $\alpha$ equal to $\pm 1$: **a** original image histogram and **b** after modification



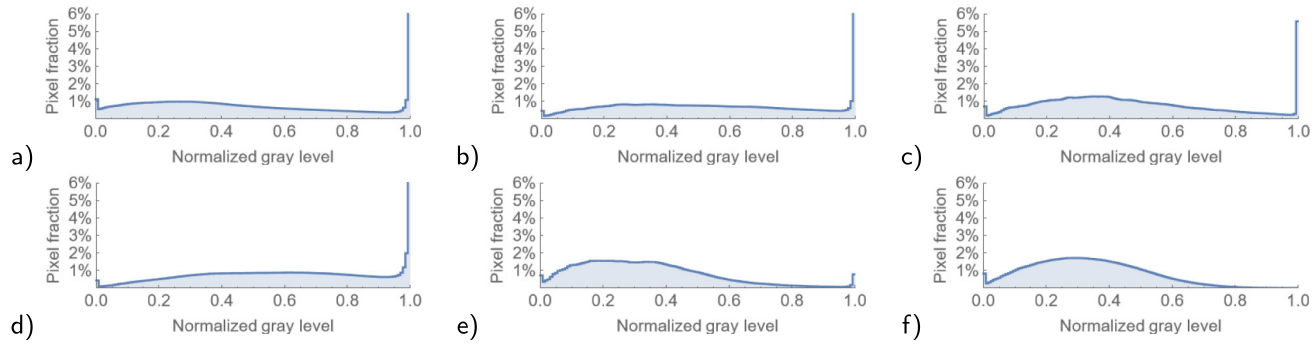




**Fig. 11** The influence of non-uniform illumination with $\alpha$ equal to $\pm 1$ on image histogram of each defect type: **a** patches, **b** pitted surface, **c** inclusions, **d** crazings, **e** scratches, and **f** rolled-in scale

**Table 1** Results of case 1—both training and testing on NEU

| Model | $F_1$ score | Accuracy | Precision | Recall |
|---|---|---|---|---|
| EN2s | 1.0 | 1.0 | 1.0 | 1.0 |
| SN-MRFLM | 0.9917 | 0.9917 | 0.9917 | 0.9917 |

**Table 3** Results of case 3—training on E-NEU, testing on NEU

| Model | $F_1$ score | Accuracy | Precision | Recall |
|---|---|---|---|---|
| EN2s | 0.9693 | 0.9694 | 0.9713 | 0.9494 |
| SN-MRFLM | 0.9668 | 0.9667 | 0.9684 | 0.9667 |

nearly black stains (cf. Fig. 1a) and in the case of pitted surface, it is related to small black dots (cf. Fig. 1b). Yang and Liu [28] complained that crazing and rolled-in-scale are very similar and thus their unambiguous discrimination is very challenging. While some similarity can be seen when looking on Fig. 1d and f, the histograms are qualitatively similar (showing one wide peak) but the position of the maximum is shifted to the right for pitted surface (Fig. 7d), and to the left for rolled-in scale (Fig. 7f).

Now, let us see how image modifications introduced in [30] affect the image histograms. In Fig. 8, one can see the effect of various modifications on images of patches. The influence of motion blur (with $L_{cm}$ either 2 or 5) is very weak (see Fig. 8a and b). Non-uniform illumination mostly acts by smoothing out the distribution and increasing the value of the sharp maximum at 1 (see Fig. 8c and d). Adding noise has the contrary effect—the height of the sharp peak on the right is decreased. The analogous figures for other defect types are shown in the Appendix (Figs. 17–21).

In the case of rolled-in scale, one can see a very clear effect of non-uniform illumination, especially with $\alpha \pm 1$ (cf. Figs. 9 and 21). Namely, the height of the maximum is greatly reduced, and its width considerably increases. A quite interesting example is the effect of illumination on the crazing defect image histogram (cf. Figs. 10 and 19). As can be seen in Fig. 10b, a new peak corresponding to black points comes into existence that was absent in Fig. 10a. The new histogram is thus more similar to the histogram for patches (cf. Fig. 7a) than for crazing.

As can be seen above, the non-uniform illumination, especially with the higher luminance range value, introduces the biggest change in image histograms. Therefore, a new Fig. 11 was prepared that shows image histograms for each defect type after modification by non-uniform illumination with $\alpha = \pm 1$. The figure can be compared with Fig. 7. One can see that even though the image histograms for clean images of defects could be discriminated from each other, the image histogram for defect images modified with non-uniform illumination presented in Fig. 11 cannot be distinguished anymore. In particular, histograms for patches, pitted surface, inclusion, and crazing look very similar. Also, the histograms for scratches and rolled-in scale are very similar to each other.

To conclude, one can notice that the histograms can be helpful for distinguishing some defect categories, but for others, the histograms are very similar, even though the defects are not. The modifications present in the E-NEU dataset have a considerable impact on their shape. The most prominent example is the severe non-uniform illumination (Fig. 11) which makes seeing differences between defect categories in histograms almost impossible. Considering the above, it is clear that histograms cannot serve as a reliable tool for defect classification, especially for the E-NEU dataset. Usage of other tools, such as CNNs, should be thus considered.

### 3.2 Convolutional neural networks

In most of the papers based on the NEU dataset published so far, both training and testing were performed on the NEU dataset. The training dataset was typically obtained by taking 80% of the images, and the remaining ones served as a testing dataset. In [30–32], the training was performed also on the NEU dataset, but the trained model was tested also on the diversity-enhanced E-NEU dataset. Here, we exercised a more elaborate approach. Namely, 4 cases were investigated:

1. training on NEU, testing on NEU,
2. training on NEU, testing on E-NEU,
3. training on E-NEU, testing on NEU,
4. training on E-NEU, testing on E-NEU.

The purpose of such combinations is to thoroughly examine what is the effect of the diversity enhancement present in the E-NEU dataset on the classification performance. Each case was tested using both EN2s and SN-MRFLM. The metrics corresponding to the results are summarized in Tables 1, 2,

**Table 2** Results of case 2—training on NEU, testing on E-NEU

| Model | $F_1$ score | Accuracy | Precision | Recall |
|---|---|---|---|---|
| EN2s | 0.7029 | 0.6832 | 0.8461 | 0.6832 |
| SN-MRFLM | 0.8835 | 0.8822 | 0.8956 | 0.8822 |

**Table 4** Results of case 4—both training and testing on E-NEU

| Model | $F_1$ score | Accuracy | Precision | Recall |
|---|---|---|---|---|
| EN2s | 0.9897 | 0.9897 | 0.9902 | 0.9897 |
| SN-MRFLM | 0.9996 | 0.9996 | 0.9996 | 0.9996 |

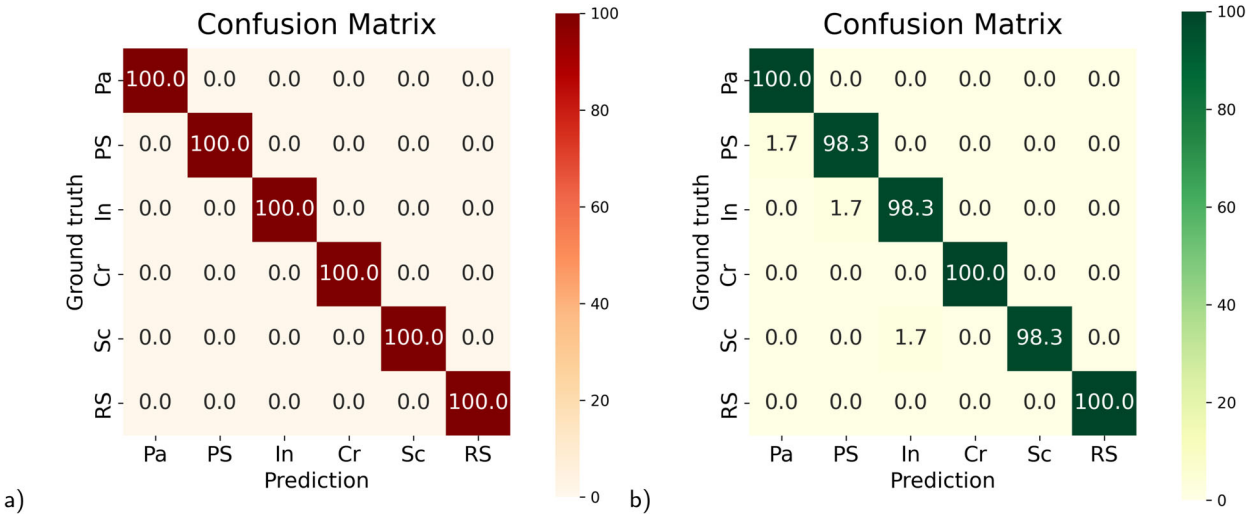



**Fig. 12** Confusion matrices for case 1—both training and testing on NEU: **a** EfficientNet-v2-s and **b** Multi-Respective Field Learning Module (SN-MRFLM). Numerical values available for copying are additionally presented in Supplementary Table 14

3, and 4. The corresponding confusion matrices (CM) are shown in Figs. 12, 13, 14, and 15.

Table 1 shows the metrics obtained for the models trained and tested on the NEU dataset (without enhancements). EN2s provided perfect classification, while SN-MRFLM achieved over 99% accuracy. In order to see the classification performance with respect to each defect, it is useful to investigate the confusion matrices (Fig. 12). EfficientNet-v2-s led to 100% correctly classified images in each category. The SN-MRFLM had difficulties with pitted surface, inclusions, and scratches: 1.7% of such images were classified incorrectly. For the remaining 3 defect categories, 100% of images were correctly classified.

Now, let us look on the case where the models were trained using the NEU dataset and tested using the diversity-enhanced images (case 2). This corresponds to the situation examined also in [30–32], where the system is supposed to be trained on clean data and then applied in the industrial conditions, where various disturbances occur. This time, the SN-MRFLM, specifically developed for such a case, shows its superiority in tackling this task over EfficientNet-v2-s. Table 2 shows that the overall accuracy of EfficientNet-v2-s is only 68%, while SN-MRFLM scored over 88%. It is thus seen that although the performance of both architectures on clean images (case 1) is similar, and even slightly better in the case of EN2s, SN-MRFLM is visibly better on perturbed

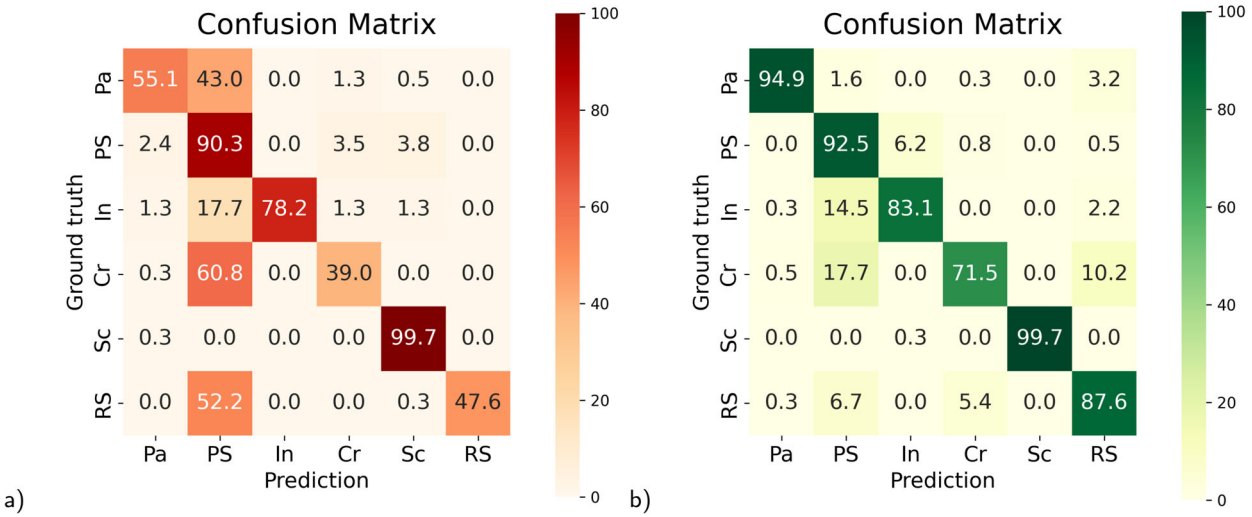



**Fig. 13** Confusion matrix for case 2—training on NEU, testing on E-NEU: **a** EfficientNet-v2-s and **b** Multi-Respective Field Learning Module (SN-MRFLM). Numerical values available for copying are additionally presented in Supplementary Table 15

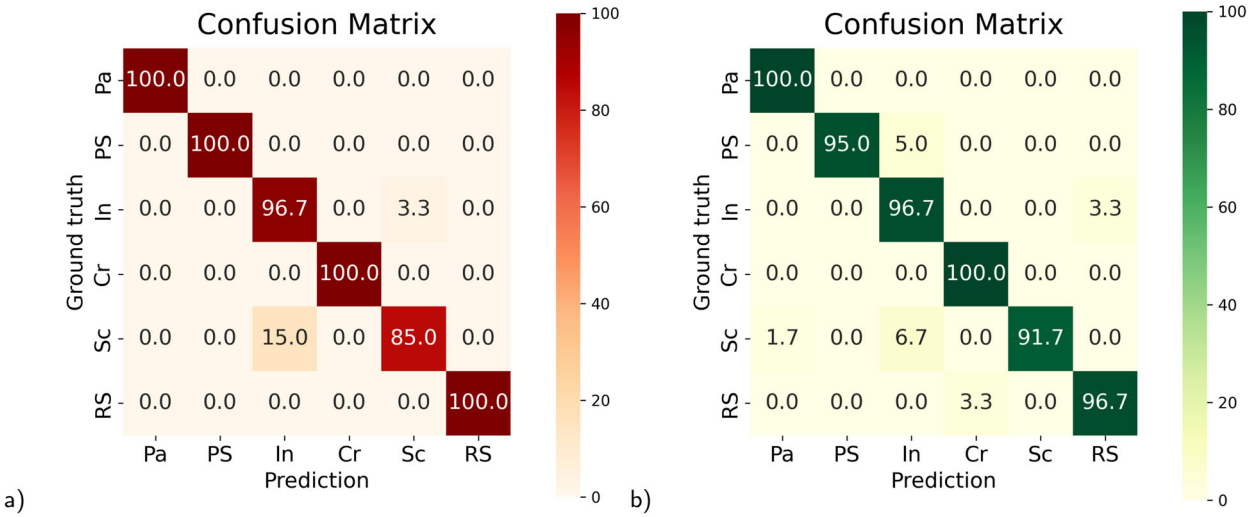



**Fig. 14** Confusion matrix for case 3—training on E-NEU, testing on NEU: **a** EfficientNet-v2-s and **b** Multi-Respective Field Learning Module (SN-MRFLM). Numerical values available for copying are additionally presented in Supplementary Table 16

images. On the other hand, one has to admit that the classification performance of SN-MRFLM also deteriorated.

Let us now take a look on classification results for each defect type separately (Fig. 13). In the case of EfficientNet-v2-s, the percentage of true positives for some defect types is really low. Only 39.0% of crazing images were classified correctly, while 60.8% of them were classified as pitted surface. 52.2% of rolled-in scale images, as well as 43.0% of patch images, were classified as pitted surface. For the rest of the classes, the fraction of true positives is at least 78%, and in the case of scratches, it is 99.7%. Note that there was a strong tendency of the model to classify other defect types as pitted surface. SN-MRFLM performed visibly better than EN2s in this case—the lowest fraction of TPs was 71.5% (for crazings). Also, inclusions and rolled-in scale defects were difficult for this architecture (83.1% and 87.6% of correct predictions, respectively).

Case 3 seems to be a rather academic example, where the model is trained on images containing diversity enhancements (E-NEU) and tested on clean images (NEU). Nevertheless, we have included this case for completeness. It is interesting to see that prediction performance in this case (Table 3) slightly deteriorated wrt. case 1 (Table 1). The accuracy of EN2s dropped to 96.9% (2.8% decline), and that of SN-MRFLM dropped to 96.6% (2.5% decline). The conclusion is thus that the model trained on distorted data achieves a lower score on clean data. When analyzing each class separately (Fig. 14), EN2s have the most difficulties with correct

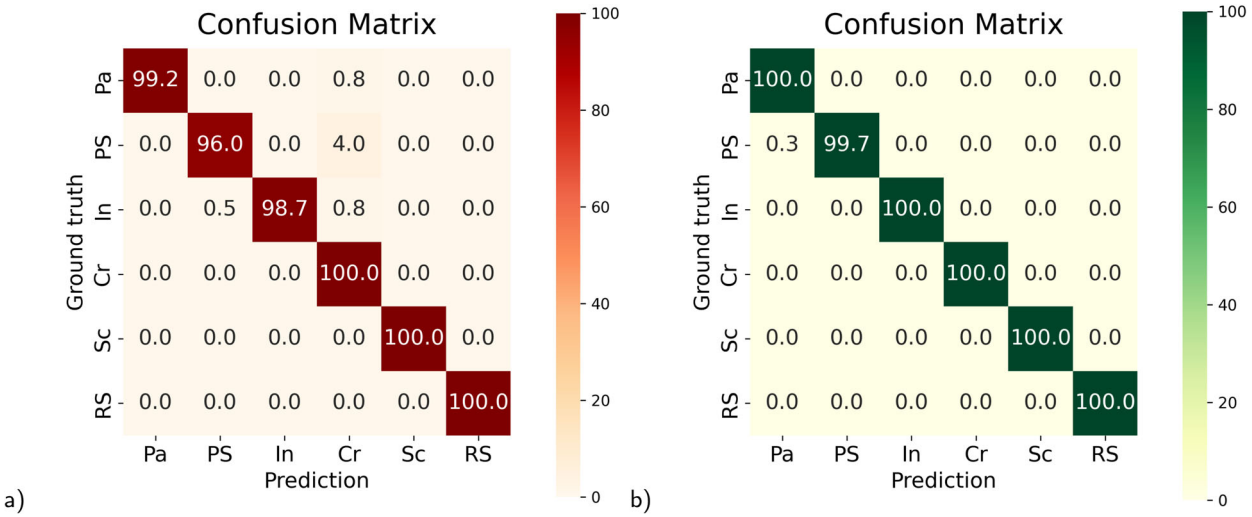



**Fig. 15** Confusion matrix for case 4—both training and testing on E-NEU: **a** EfficientNet-v2-s and **b** Multi-Respective Field Learning Module (SN-MRFLM). Numerical values available for copying are additionally presented in Supplementary Table 17

classification of scratches (similarly as in case 1). This time, this category is also the most difficult for SN-MRFLM.

The last combination (case 4) is a situation where the model was both trained and tested on the E-NEU dataset. This situation seems to be possible to occur in a real situation, e.g., when it was not possible to obtain sufficiently high-quality training data. The prediction performance was similar to case 1: it is 99% for EN2s and almost 100% for SN-MRFLM (cf. Table 4). Figure 15 shows the confusion matrices for both networks. The correct classification of the pitted surface was the most troublesome for both architectures.

### 3.3 CNNs on preprocessed data

Nath, Chattopadhyay, and Desai [31] reported improvement of classification results when preprocessing the perturbed images from the E-NEU dataset using histogram equalization (HE). Therefore, we have also checked if pre-processing the images with histogram equalization improves the classification. Table 5 reports the results (averaged over 10 trainings), which should be compared with Table 2. One can see that applying HE to all the images improved classification performance to a moderate level in the case of EN2s. However, in the case of SN-MRFLM, it led to worse classification accuracy.

Since HE did not provide any breakthrough improvement, other image preprocessing options were considered. Namely, the filters described in Section 2.3 were separately applied to all images (with windows $3 \times 3$, $5 \times 5$, and $7 \times 7$ pixels). The results are shown in Table 6. One can see that this time, the results were considerably improved. What is even more interesting, EN2s on preprocessed images achieved similar performance as SN-MRFLM. The best accuracy (highlighted in Table 6) was obtained with Gaussian blur in the case of EN2s and bilateral filter in the case of SN-MRFLM (using 7 pixel window in both cases). For EN2s, the best accuracy was equal to 97.9%, and for SN-MRFLM, it was equal to 98.6%. Note that the table shows mean values over 10 network trainings. Tables reporting each training result separately and the corresponding standard deviations are provided in the Supplementary Material. The confusion matrix corresponding to the best results obtained with EN2s and SN-MRFLM is shown in Fig. 16. This time, the crazing images were most difficult for classification in the case of EN2s, and inclusion images were the most troublesome for SN-MRFLM.

**Table 5** Results of case 2: training on NEU and testing on E-NEU when all images were preprocessed using histogram equalization (HE)

| Preprocessing | $F_1$ score | Accuracy | Precision | Recall |
|---|---|---|---|---|
| EN2s | 0.680 | 0.671 | 0.833 | 0.671 |
| SN-MRFLM | 0.764 | 0.768 | 0.778 | 0.768 |

Values averaged over 10 network trainings are shown

**Table 6** Results of case 2: training on NEU and testing on E-NEU when all images were preprocessed with one of the filters: blur (B), median blur (MB), Gaussian blur (GB), and bilateral filter (BF)

| Preprocessing | $F_1$ score | Accuracy | Precision | Recall |
|---|---|---|---|---|
| EN2s | | | | |
| No | 0.713 | 0.712 | 0.813 | 0.712 |
| B 3 | 0.884 | 0.885 | 0.910 | 0.885 |
| B 5 | 0.929 | 0.929 | 0.941 | 0.929 |
| B 7 | 0.926 | 0.925 | 0.939 | 0.925 |
| MB 3 | 0.906 | 0.903 | 0.920 | 0.903 |
| MB 5 | 0.934 | 0.933 | 0.944 | 0.933 |
| MB 7 | 0.894 | 0.892 | 0.924 | 0.892 |
| GB 3 | 0.905 | 0.903 | 0.921 | 0.903 |
| GB 5 | 0.955 | 0.955 | 0.962 | 0.955 |
| GB 7 | 0.971 | 0.971 | 0.973 | 0.971 |
| BF 3 | 0.881 | 0.879 | 0.912 | 0.879 |
| BF 5 | 0.920 | 0.918 | 0.939 | 0.918 |
| BF 7 | 0.877 | 0.880 | 0.908 | 0.880 |
| SN-MRFLM | | | | |
| No | 0.881 | 0.879 | 0.904 | 0.879 |
| B 3 | 0.961 | 0.961 | 0.964 | 0.961 |
| B 5 | 0.963 | 0.962 | 0.965 | 0.962 |
| B 7 | 0.921 | 0.923 | 0.941 | 0.923 |
| MB 3 | 0.949 | 0.949 | 0.953 | 0.949 |
| MB 5 | 0.964 | 0.963 | 0.966 | 0.963 |
| MB 7 | 0.965 | 0.965 | 0.968 | 0.965 |
| GB 3 | 0.965 | 0.965 | 0.966 | 0.965 |
| GB 5 | 0.955 | 0.955 | 0.958 | 0.955 |
| GB 7 | 0.966 | 0.966 | 0.968 | 0.966 |
| BF 3 | 0.948 | 0.948 | 0.953 | 0.948 |
| BF 5 | 0.963 | 0.963 | 0.965 | 0.963 |
| BF 7 | 0.972 | 0.971 | 0.972 | 0.971 |

The highest accuracies for each CNN architecture are highlighted. Values averaged over 10 network trainings are shown

To the best of the authors' knowledge, there are only 3 papers where the efficiency of the models was tested over the E-NEU dataset. In [30] (the original paper presenting the E-NEU dataset), the authors trained their model on the NEU dataset and tested on both NEU and E-NEU datasets, which in the nomenclature of the present paper corresponds to cases 1 and 2, respectively. Fu et al. proposed the SqueezeNet-based architecture (applied also here and denoted SN-MRFLM) to solve the classification problem and obtained an accuracy 100% when testing on NEU (analogous to our case 1) and 97.5% when testing on E-NEU (analogous to our case 2). They did not present other metrics though. Concerning accuracy in case1, 100% is the same as the result

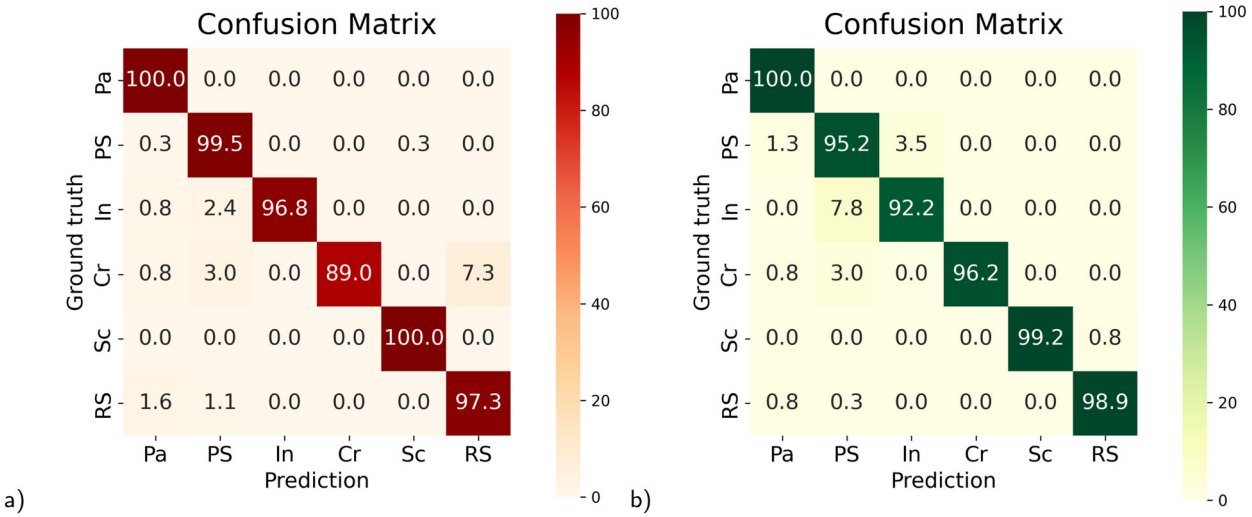



**Fig. 16** Confusion matrix for case 2: training on NEU and testing on E-NEU, with image preprocessing: **a** Gaussian blur (7 × 7 window) + EfficientNet-v2-s and **b** bilateral filter (7 × 7 window) + Multi-Respective Field Learning Module (SN-MRFLM). Numerical values available for copying are additionally presented in Supplementary Table 18

of EfficientNet-v2-s and slightly better than 99.2% obtained with SN-MRFLM in our case, cf. Table 1. Differences in results corresponding to the same architecture (SN-MRFLM) may result from different learning strategies or hyperparameter tuning. More interestingly, 97.5% accuracy obtained in [30] on the E-NEU dataset using the SN-MRFLM is much better than 53.7% obtained with EfficientNet-v2-s and 88.2% obtained with SN-MRFLM in our case, cf. Table 2. The large difference between the accuracy obtained using SN-MRFLM in [30] and us is difficult to explain at this stage.

On the other hand, to the best of the authors' knowledge, there are no reports of better accuracy on the E-NEU dataset than 97.5% as reported in [30]. Nath et al. [31] used histogram equalization and another network architecture and obtained only 92.4% accuracy. To sum up, 97.1% accuracy obtained in the present paper by applying EN2s with Gaussian blur and SN-MRFLM with bilateral filtering is similar to the best result reported in [30] and by far outperforms the second best result reported in [31], cf. Table 7. Note, however, that the values reported here were averaged over 10 network trainings, while it seems that in [30] this was just one result. Due to randomness, it could happen that the result reported in *op. cit.* was obtained by chance in a single network training. For example, in our case, in one of the training (see Supplementary Table 5), the highest training accuracy was even 98.7%.

**Table 7** Best accuracy when training on NEU dataset and testing on E-NEU dataset reported in the literature and in the present contribution

| Fu et al. [30] | Nath et al. [31] | This work (average) |
|---|---|---|
| 97.5% | 92.4% | 97.1% |

## 4 Discussion

### 4.1 Number of epochs for Efficient Net

In this paper, in the case of EN2s, we have chosen 20 epochs in order to avoid overfitting. Now, let us check what is the result of this choice. Table 8 presents the effect of the number of epochs on classification metrics in case 2 (case 1 is shown in Supplementary Table 13). Note that each result was obtained after a single training, and thus some statistical fluctuations are unavoidable. In general, no significant improvement in classification accuracy is seen when increasing the number of epochs to more than 20, and thus, it can be concluded that the selected number of epochs is optimal.

### 4.2 Time of classification

In order to assess the real-world applicability of the developed technique, we have computed time of image preprocessing and inference. These data are presented in Table 9. Parameters of the computer used for those computations were provided in Section 2.2. In general, the preprocessing time does not exceed 6 ms. The average inference time in the case of EN2s is around 42 ms, and in the case of SN-MRFLM, it is around 7 ms. The total time does not exceed 49 ms for EN2s and 14 ms for SN-MRFLM. As pointed out in Section 11 of [6], for a strip going at a speed of 20 m/s, the maximum total time is 12.5 ms. According to this criterion, EN2s

**Table 8** Classification metrics of case 2 (training using NEU, testing on E-NEU) with EN2s—effect of the number of epochs

| No. of epochs | F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| 1 | 0.6413 | 0.6595 | 0.7422 | 0.6595 |
| 2 | 0.6412 | 0.6573 | 0.7741 | 0.6573 |
| 3 | 0.5651 | 0.5748 | 0.7691 | 0.5748 |
| 4 | 0.7381 | 0.7464 | 0.8074 | 0.7464 |
| 5 | 0.6119 | 0.6120 | 0.7369 | 0.6120 |
| 6 | 0.6767 | 0.6770 | 0.8123 | 0.6770 |
| 7 | 0.6631 | 0.6676 | 0.8399 | 0.6676 |
| 8 | 0.6710 | 0.6640 | 0.8132 | 0.6640 |
| 9 | 0.7672 | 0.7666 | 0.8431 | 0.7666 |
| 10 | 0.5339 | 0.5488 | 0.8155 | 0.5488 |
| 12 | 0.6819 | 0.6873 | 0.8130 | 0.6873 |
| 14 | 0.6047 | 0.6165 | 0.7956 | 0.6165 |
| 16 | 0.7396 | 0.7231 | 0.8757 | 0.7231 |
| 18 | 0.6948 | 0.6859 | 0.8083 | 0.6859 |
| 20 | 0.7040 | 0.6958 | 0.7979 | 0.6958 |
| 40 | 0.6104 | 0.6089 | 0.7466 | 0.6089 |
| 60 | 0.6744 | 0.6644 | 0.7843 | 0.6644 |
| 80 | 0.5820 | 0.5950 | 0.7707 | 0.5950 |
| 100 | 0.6895 | 0.6810 | 0.8411 | 0.6810 |
| 120 | 0.7603 | 0.7433 | 0.8417 | 0.7433 |
| 140 | 0.7071 | 0.7030 | 0.8476 | 0.7030 |
| 160 | 0.6795 | 0.6676 | 0.7677 | 0.6676 |
| 180 | 0.7391 | 0.7321 | 0.8309 | 0.7321 |
| 200 | 0.6080 | 0.5892 | 0.8145 | 0.5892 |

is not suitable for image processing at this speed, while in the case of SN-MRFLM, some schemes are rather below the boundary (no preprocessing, histogram equalization and blur), some others are close to this boundary (median blur and bilateral filter, especially with $7 \times 7$ window), while Gaussian blur violates this boundary to some extent. Note that the best accuracy in the case of SN-MRFLM was observed with a bilateral filter with $7 \times 7$ window (cf. Table 6), whose total time is just below the limit estimated in [6]. This seems to be a good prognostic for future applications, but the real applicability should be thoroughly re-evaluated concerning specific manufacturing conditions, as well as the hardware applicable on-site.

### 4.3 Predicting defects

The present paper focuses on defects classification, while neglecting two other components of defects inspection, namely defect detection and defects segmentation. Note that defects inspection as a whole is only related to the existing defects and is blind to the process of their formation. However, analyzing the process of defects formation is also an interesting research direction. For example, in [44], finite element method (FEM) study was carried out in order to redesign the caliber rolling process, so that the creation of surface defects in AISI 4140 steel was limited. Note that the study was not purely theoretical; rather, the results were verified in the actual rod mill of SEAH BESTEEL Inc. at Kunsan, Korea. FEM study of initiation and growth of surface defects during hot rolling was reported in [45]. The main conclusion from this paper was that increase of friction increases the probability of defect formation.

### 4.4 Defects and fatigue

The linkages between material defects and fatigue are very significant, cf., e.g., [46–48]. These linkages are also recognized by means of so-called Kitagawa diagrams [49]. The impact of surface defects on very high cycle fatigue (VHCF) was analyzed in [50]. The case of *clean* spring steel was considered (which is important, since in high-strength steels, the fact that there are many *internal* crack initiation sites could make the study of surface defect influence irrelevant).

As pointed out in [47, 51], the important information from the fatigue point of view is the maximum defects size. [52] studied VHCF behavior of medium carbon structural steel and concluded that surface defects with dimensions more than $200\mu m$ were the primary crack initiation site. Note, however, that the defect size cannot be considered as the independent quantity as it depends on steel type and microstructure. For example, defects of about 50 $\mu m$ do not decrease the fatigue strength in the case of mild steel [48], but in the case of high-strength steels even defects below 10 $\mu m$ can be detrimental [53]. Based on these assumptions, [54] proposed the material-dependent critical defect size. Defects larger than this size should be considered detrimental for fatigue resistance. The critical size for the materials studied in *op. cit.* is on the order 100 $\mu m$.

Note that material defects should be considered in two ways in this case, i.e., as surface defects or defects occurring in the microstructure of the steel. Microstructural defects are mainly related to the effects of metallurgical processes, including the effects of deoxidation and segregation of steel, which lead to the formation of non-metallic inclusions or microcracks in the microstructure, as well as to thermal processes during hot forming. In turn, surface defects are considered in the paper, and unfortunately, there is no direct information on the sizes of defects included in the NEU dataset [17]. However, based on common metallurgical knowledge, we assume that such defects are macroscopic defects, with the order of magnitude of 0.5 mm upwards, visible on the surface and capable of being recorded by image capture and analysis devices, thus almost one order of magnitude higher than the critical microstructural defect size established in [54]. Such defects can be highly harmful from

**Table 9** Preprocessing and inference times for case 2 (training on NEU and testing on E-NEU) when all images were preprocessed with histogram equalization (HE) or one of the filters: blur (B), median blur (MB), Gaussian blur (GB) and bilateral filter (BF)

| Preprocessing | EN2s | | | SN-MRFLM | | |
|---|---|---|---|---|---|---|
| | Preprocessing time [ms] | Average inference time [ms] | Total time [ms] | Preprocessing time [ms] | Average inference time [ms] | Total time [ms] |
| No | 3.137 | 43.322 | 46.459 | 2.325 | 7.685 | 10.010 |
| HE | 2.726 | 42.348 | 45.074 | 1.996 | 7.292 | 9.288 |
| B 3 | 2.369 | 41.815 | 44.184 | 2.350 | 7.438 | 9.788 |
| B 5 | 2.543 | 42.198 | 44.741 | 2.675 | 7.269 | 9.944 |
| B 7 | 2.673 | 42.034 | 44.707 | 2.888 | 7.358 | 10.246 |
| MB 3 | 3.154 | 42.519 | 45.673 | 2.875 | 7.397 | 10.272 |
| MB 5 | 3.118 | 42.661 | 45.779 | 3.359 | 7.369 | 10.728 |
| MB 7 | 4.917 | 42.271 | 47.188 | 4.985 | 7.448 | 12.433 |
| GB 3 | 5.906 | 41.839 | 47.745 | 5.936 | 7.702 | 13.638 |
| GB 5 | 4.451 | 42.322 | 46.773 | 4.951 | 7.538 | 12.489 |
| GB 7 | 5.567 | 42.530 | 48.097 | 5.181 | 7.531 | 12.712 |
| BF 3 | 3.584 | 42.717 | 46.301 | 3.440 | 7.299 | 10.739 |
| BF 5 | 4.669 | 42.178 | 46.847 | 4.232 | 7.436 | 11.668 |
| BF 7 | 5.960 | 42.067 | 48.027 | 4.918 | 7.298 | 12.216 |

First line shows results where no filter was applied. Values were averaged over 10 network trainings

the point of view of fatigue, but in this case, in addition to the length, their depth, the nature of propagation or the defect area (in the case of rolled scales or clusters of macroscopic non-metallic inclusions) should also be taken into account. This issue requires further research.

### 4.5 Steel defects inspection in long products

Note that although it seems that steel surface defect inspection received more attention in the context of flat products, there are also some studies devoted to long products, e.g., [2, 15, 22, 23, 55]. In [15], a modified SVM approach taking into account process knowledge was applied to classify defects (seams, scales, and cracks) in hot-rolled bars. A further step forward was made in [2], where the occurrence of defects was predicted based on the process parameters using Bayesian modeling. In [22], a method of detecting three types of defects on steel bars (pit, scratch, and overfill) was described. In [23], a method calibrated for pit defects on steel bars was reported. In [55], an innovative semi-supervised anomaly detection model was used for defect detection on rail surfaces (both rails coming directly from the production line and in-service rails).

## 5 Conclusions

In the article, the performance of two deep learning architectures in classifying steel surface defects from the widely cited NEU dataset and the less-known diversity-enhanced E-NEU dataset was tested. The effectiveness of increasing the classification performance on the E-NEU dataset by image preprocessing was also studied. The main conclusion is that image preprocessing with well-known filters can effectively increase the classification accuracy, above the values achievable by applying even the most fined-tuned CNN architectures only. Accuracy improvement was demonstrated by comparing with available research reporting studies on the E-NEU dataset. In addition, the industrial applicability of the developed framework was studied by analyzing the preprocessing and inference time. Finally, gray level histograms for NEU and E-NEU datasets were reported for the first time.

Note that this paper focused on testing and improving the noise robustness of defect classification by using the publicly available E-NEU dataset. One should, however, bear in mind that image distortions in the E-NEU dataset have been themselves introduced artificially. The industrial applicability of the developed methodology should be further elaborated by testing on images that have disturbances related to real manufacturing conditions.

## Appendix

Figures 17, 18, 20, and 21 show the histograms modified by diversities included in the E-NEU dataset for pitted surface, inclusion, crazing, scratches, and rolled-in scale.

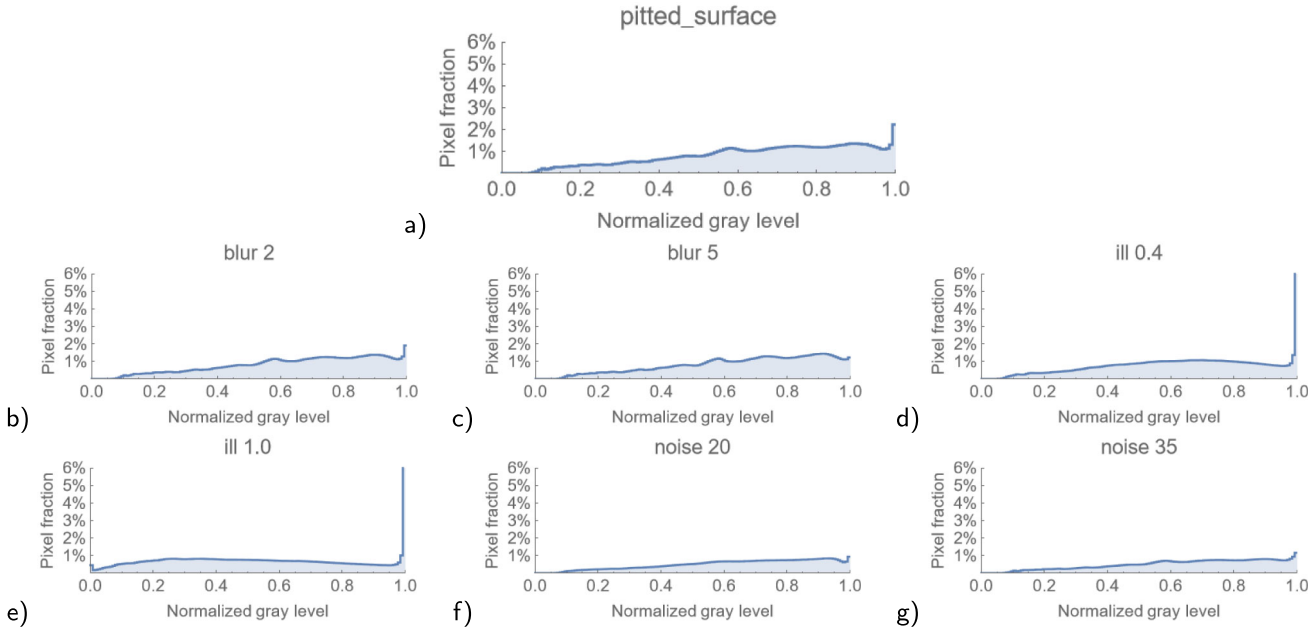



**Fig. 17** The histograms for pitted surface—influence of additional effects: **a** original images; motion blur with $L_{cm}$ equal to **b** 2 and **c** 5; non-uniform illumination with $\alpha$ equal to **d** $\pm 0.4$ and **e** $\pm 1$; camera noise with SNR equal to **f** $20db$ and **g** $35db$

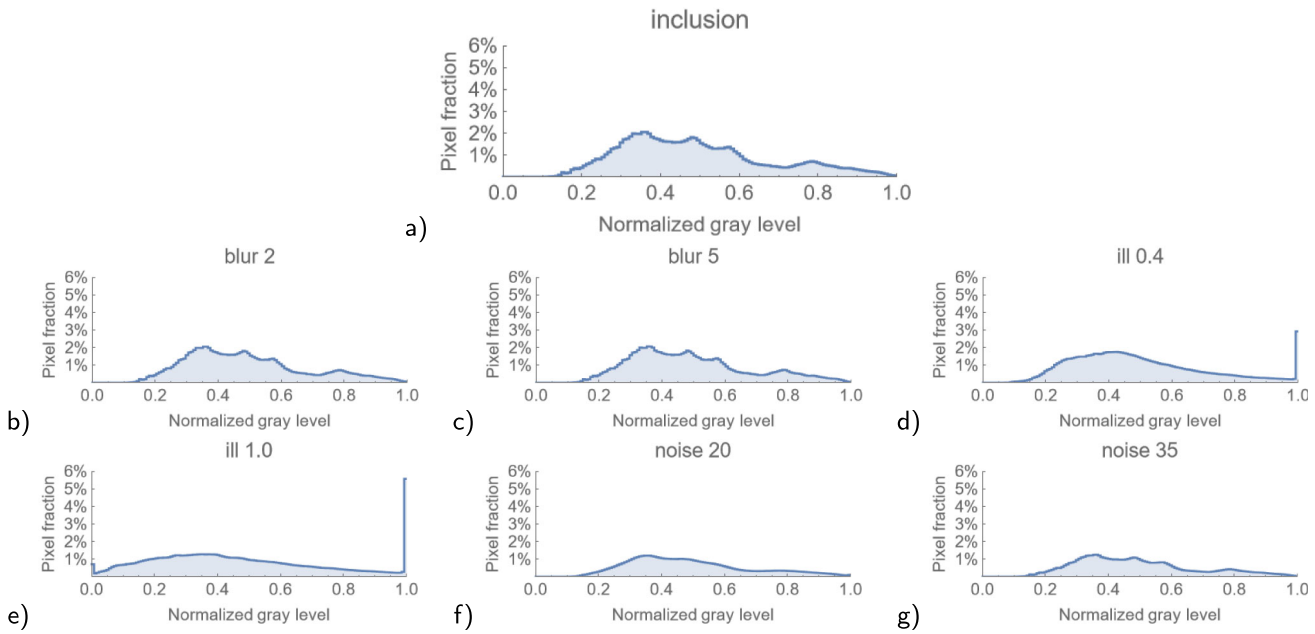



**Fig. 18** The histograms for inclusion—influence of additional effects: **a** original images; motion blur with $L_{cm}$ equal to **b** 2 and **c** 5; non-uniform illumination with $\alpha$ equal to **d** $\pm 0.4$ and **e** $\pm 1$; camera noise with SNR equal to **f** $20db$ and **g** $35db$

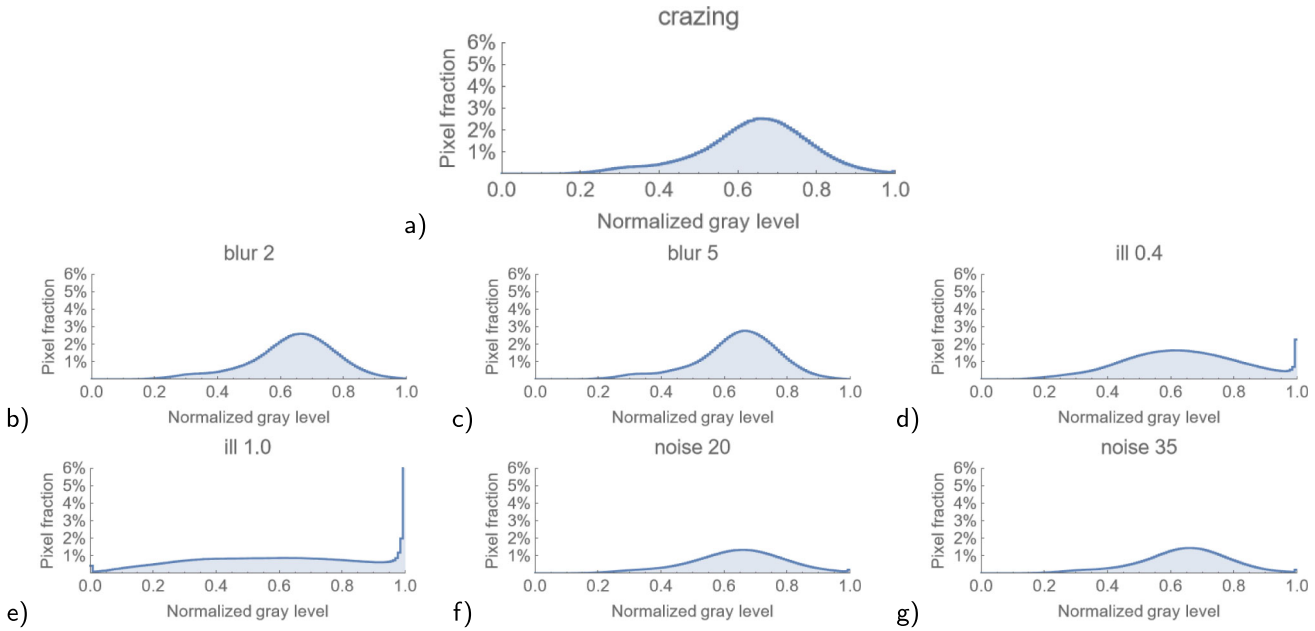



**Fig. 19** The histograms for crazing—influence of additional effects: **a** original images; motion blur with $L_{cm}$ equal to **b** 2 and **c** 5; non-uniform illumination with $\alpha$ equal to **d** $\pm 0.4$ and **e** $\pm 1$; camera noise with SNR equal to **f** $20db$ and **g** $35db$

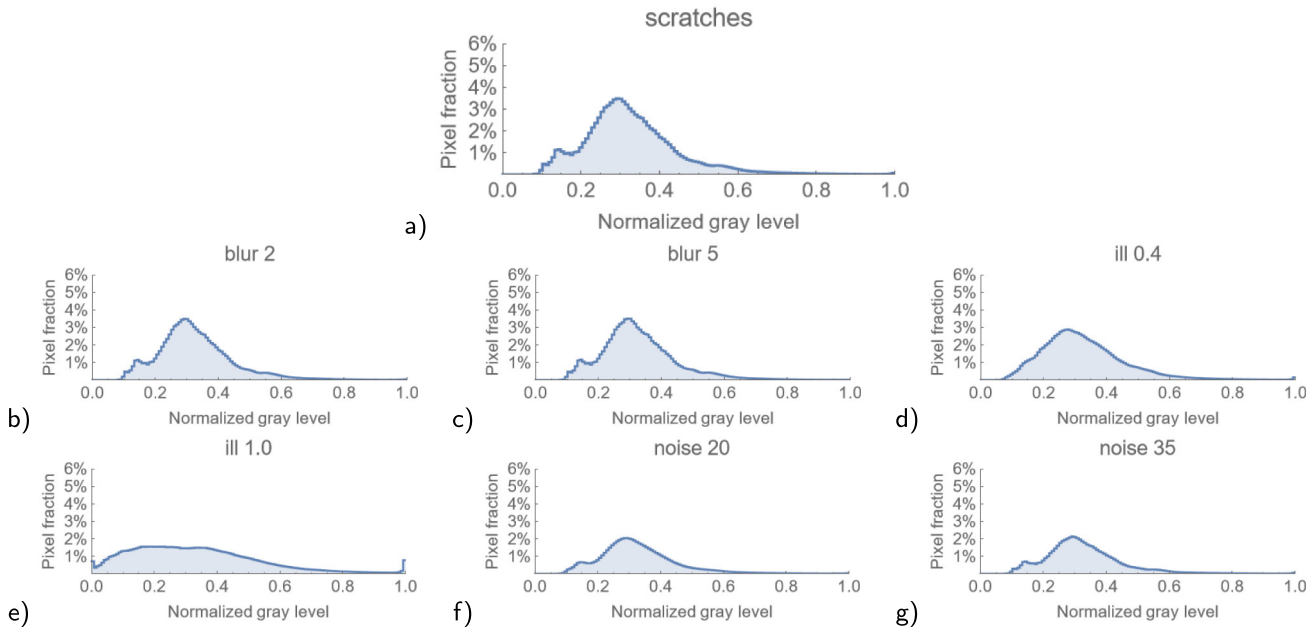



**Fig. 20** The histograms for scratches—influence of additional effects: **a** original images; motion blur with $L_{cm}$ equal to **b** 2 and **c** 5; non-uniform illumination with $\alpha$ equal to **d** $\pm 0.4$ and **e** $\pm 1$; camera noise with SNR equal to **f** $20db$ and **g** $35db$

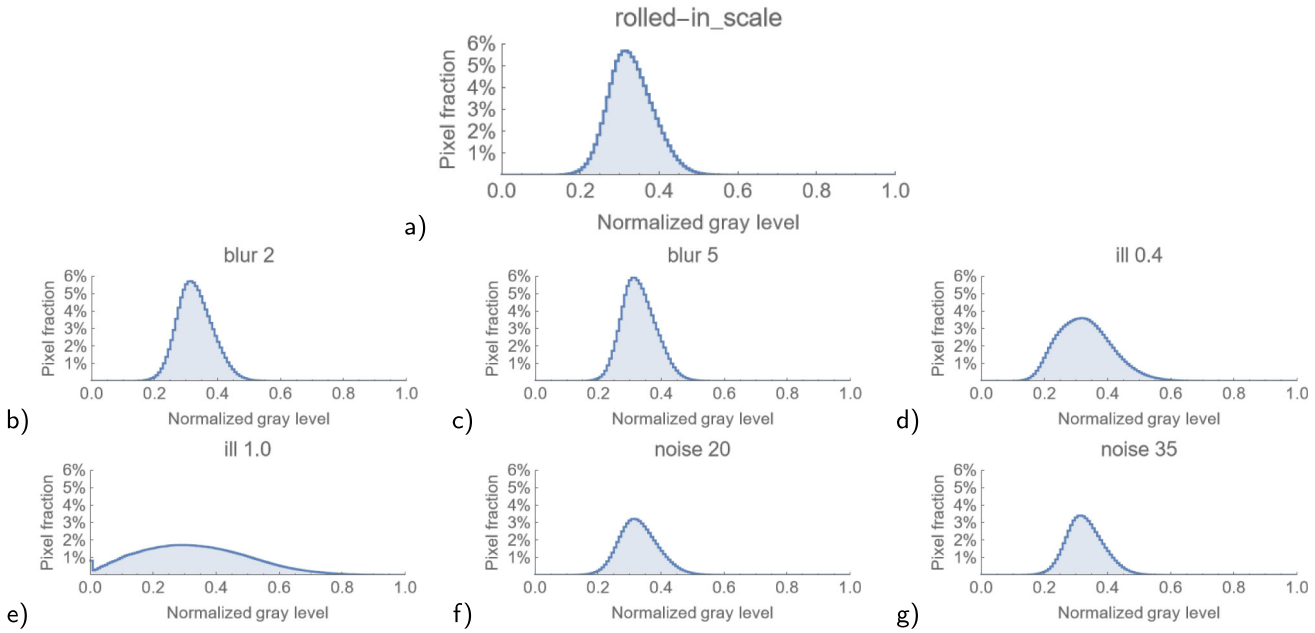



**Fig. 21** The histograms for rolled-in scale—influence of additional effects: **a** original images; motion blur with $L_{cm}$ equal to **b** 2 and **c** 5; non-uniform illumination with $\alpha$ equal to **d** $\pm 0.4$ and **e** $\pm 1$; camera noise with SNR equal to **f** $20db$ and **g** $35db$

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/s00170-025-16539-y.

**Author contributions** Karol Frydrych: conceptualization of this study, data curation, methodology, software, writing—original draft preparation. Maciej Tomczak: writing—original draft preparation, conceptualization of this study, methodology, software. Jarosław Jasiński: writing—review and editing. Stefanos Papanikolaou: writing—review and editing.

**Funding** We acknowledge the support from the European Union Horizon 2020 research and innovation program under NOMATEN Teaming grant agreement no. 857470 and from the European Regional Development Fund via the Foundation for Polish Science International Research Agenda Plus program grant no. MAB PLUS/2018/8, which partially covered the salary of Karol Frydrych, Maciej Tomczak, Jarosław Jasiński and Stefanos Papanikolaou.

The publication was created within the framework of the project of the Minister of Science and Higher Education "Support for the activities of Centres of Excellence established in Poland under Horizon 2020" under contract no. MEiN/2023/DIR/3795.

## Declarations

**Competing interests** The authors declare no competing interests.



## References

1. Ghorai S, Mukherjee A, Gangadaran M, Dutta PK (2012) Automatic defect detection on hot-rolled flat steel products. IEEE Trans Instrum Meas 62(3):612–621. https://doi.org/10.1109/TIM.2012.2218677
2. Agarwal K, Shivpuri R (2015) On line prediction of surface defects in hot bar rolling based on Bayesian hierarchical modeling. J Intell Manuf 26:785–800. https://doi.org/10.1007/s10845-013-0834-y
3. Radgolchin M, Anbarsooz M (2023) Fatigue failure of centrifugal compressor impellers: a comprehensive review. Eng Fail Anal 153:107592. https://doi.org/10.1016/j.engfailanal.2023.107592
4. Tang B, Chen L, Sun W, Lin Z-K (2023) Review of surface defect detection of steel products based on machine vision. IET Image Proc 17(2):303–322. https://doi.org/10.1049/ipr2.12647
5. Lechwar S, Rauch Ł, Pietrzyk M (2015) Use of artificial intelligence in classification of mill scale defects, steel research international. 86(3):266–277. https://doi.org/10.1002/srin.201400016
6. Neogi N, Mohanta DK, Dutta PK (2014) Review of vision-based steel surface inspection systems. EURASIP J Image Video Process 2014:1–19. https://doi.org/10.1186/1687-5281-2014-50
7. Ameri R, Hsu C-C, Band SS (2024) A systematic review of deep learning approaches for surface defect detection in industrial applications. Eng Appl Artif Intell 130:107717. https://doi.org/10.1016/j.engappai.2023.107717
8. Fang X, Luo Q, Zhou B, Li C, Tian L (2020) Research progress of automated visual surface defect detection for industrial metal planar materials. Sensors 20(18):5136. https://doi.org/10.3390/s20185136

9. Mordia R, Verma AK (2022) Visual techniques for defects detection in steel products: a comparative study. Eng Fail Anal 134:106047. https://doi.org/10.1016/j.engfailanal.2022.106047
10. Cemernek D, Cemernek S, Gursch H, Pandeshwar A, Leitner T, Berger M, Klösch G, Kern R (2022) Machine learning in continuous casting of steel: a state-of-the-art survey. J Intell Manuf 33:1–19. https://doi.org/10.1007/s10845-021-01754-7
11. Liu Y, Zhang C, Dong X (2023) A survey of real-time surface defect inspection methods based on deep learning. Artif Intell Rev 56(10):12131–12170. https://doi.org/10.1007/s10462-023-10475-7
12. Luo Q, He Y (2016) A cost-effective and automatic surface defect inspection system for hot-rolled flat steel. Robot Comput-Integr Manuf 38:16–30. https://doi.org/10.1016/j.rcim.2015.09.008
13. Hao R, Lu B, Cheng Y, Li X, Huang B (2021) A steel surface defect inspection approach towards smart industrial monitoring. J Intell Manuf 32:1833–1843. https://doi.org/10.1007/s10845-020-01670-2
14. Gu C, Bao Y, Prasad S, Lyu Z, Lian J (2023) Defect engineering of fatigue-resistant steels by data-driven models. Eng Appl Artif Intell 124:106517. https://doi.org/10.1016/j.engappai.2023.106517
15. Agarwal K, Shivpuri R, Zhu Y, Chang T-S, Huang H (2011) Process knowledge based multi-class support vector classification (PK-MSVM) approach for surface defects in hot rolling. Expert Syst Appl 38(6):7251–7262. https://doi.org/10.1016/j.eswa.2010.12.026
16. Hu H, Liu Y, Liu M, Nie L (2016) Surface defect classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm. Neurocomputing 181:86–95. https://doi.org/10.1016/j.neucom.2015.05.134
17. Song K, Yan Y (2013) A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 285:858–864. https://doi.org/10.1016/j.apsusc.2013.09.002
18. Song K (2025) Neu surface defect database. http://faculty.neu.edu.cn/songkc/en/zdylm/263265/list/index.htm
19. Wen X, Shan J, He Y, Song K (2022) Steel surface defect recognition: a survey. Coatings 13(1):17. https://doi.org/10.3390/coatings13010017
20. Ren R, Hung T, Tan KC (2017) A generic deep-learning-based approach for automated surface inspection. IEEE Trans Cybern 48(3):929–940. https://doi.org/10.1109/TCYB.2017.2668395
21. Yi L, Li G, Jiang M (2017) An end-to-end steel strip surface defects recognition system based on convolutional neural networks, steel research international. 88(2):1600068. https://doi.org/10.1002/srin.201600068
22. Li W-B, Lu C-H, Zhang J-C (2012) A local annular contrast based real-time inspection algorithm for steel bar surface defects. Appl Surf Sci 258(16):6080–6086. https://doi.org/10.1016/j.apsusc.2012.03.007
23. Li W-B, Lu C-H, Zhang J-C (2013) A lower envelope weber contrast detection algorithm for steel bar surface pit defects. Opt Laser Technol 45:654–659. https://doi.org/10.1016/j.optlastec.2012.05.016
24. Akhyar F, Liu Y, Hsu C-Y, Shih TK, Lin C-Y (2023) FDD: a deep learning-based steel defect detectors. Int J Adv Manuf Technol 126(3):1093–1107. https://doi.org/10.1007/s00170-023-11087-9
25. Chen S, Jiang S, Wang X, Sun P, Hua C, Sun J (2024) An efficient detector for detecting surface defects on cold-rolled steel strips. Eng Appl Artif Intell 138:109325. https://doi.org/10.1016/j.engappai.2024.109325
26. Zhang Z, Wang W, Tian X (2023) Semantic segmentation of metal surface defects and corresponding strategies. IEEE Trans Instrum Meas 72:1–13. https://doi.org/10.1109/TIM.2023.3282301
27. Xu M, Wei J, Feng X (2024) Two-stage encoder multi-decoder network with global-local up-sampling for defect segmentation of strip steel surface defects. Eng Appl Artif Intell 138:109469. https://doi.org/10.1016/j.engappai.2024.109469
28. Yang J, Liu Z (2024) A novel real-time steel surface defect detection method with enhanced feature extraction and adaptive fusion. Eng Appl Artif Intell 138:109289. https://doi.org/10.1016/j.engappai.2024.109289
29. Zhang R, Liu D, Bai Q, Fu L, Hu J, Song J (2024) Research on x-ray weld seam defect detection and size measurement method based on neural network self-optimization. Eng Appl Artif Intell 133:108045. https://doi.org/10.1016/j.engappai.2024.108045
30. Fu G, Sun P, Zhu W, Yang J, Cao Y, Yang MY, Cao Y (2019) A deep-learning-based approach for fast and robust steel surface defects classification. Opt Lasers Eng 121:397–405. https://doi.org/10.1016/j.optlaseng.2019.05.005
31. Nath V, Chattopadhyay C, Desai K (2023) On enhancing prediction abilities of vision-based metallic surface defect classification through adversarial training. Eng Appl Artif Intell 117:105553. https://doi.org/10.1016/j.engappai.2022.105553
32. Nath V, Chattopadhyay C, Desai K (2023) NSLNET: an improved deep learning model for steel surface defect classification utilizing small training datasets. Manuf Lett 35:39–42. https://doi.org/10.1016/j.mfglet.2022.10.001
33. Frydrych K, Tomczak M, Jasiński J, Papanikolaou S (2023) Zastosowanie metod sztucznej inteligencji (AI) w procesach produkcji stali. Stal, Metale & Nowe Technologie (in Polish)
34. Awcock GJ, Thomas R (1995) Applied image processing. Springer. https://doi.org/10.1007/978-1-349-13049-8
35. Rafajłowicz E, Rafajłowicz W (2010) Wstęp do przetwarzania obrazów przemysłowych. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej
36. Tan M, Le Q (2021) EfficientNetV2: smaller models and faster training. In: International conference on machine learning, PMLR. pp 10096–10106
37. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al (2019) Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32
38. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR. pp 6105–6114
39. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
40. Iandola FN (2016) Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv:1602.07360
41. Koonce B, Koonce B (2021) Squeezenet, Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization. pp 73–85. https://doi.org/10.1007/978-1-4842-6168-2
42. Bradski G, Kaehler A (2008) Learning OpenCV: computer vision with the OpenCV library. O'Reilly Media, Inc
43. Paris S (2007) A gentle introduction to bilateral filtering and its applications. In: ACM SIGGRAPH 2007 courses, pp. 3–es
44. Na D-H, Lee Y (2013) A study to predict the creation of surface defects on material and suppress them in caliber rolling process. Int J Precis Eng Manuf 14:1727–1734. https://doi.org/10.1007/s12541-013-0232-6
45. Yu H-L, Tieu K, Lu C, Deng G-Y, Liu X-H (2013) Occurrence of surface defects on strips during hot rolling process by FEM. Int J Adv Manuf Technol 67:1161–1170. https://doi.org/10.1007/s00170-012-4556-7
46. Segall A (2006) Manufacturing defects and the evidence of thermomechanical fatigue in a ceramic vacuum furnace tube. Eng

Fail Anal 13(7):1184–1190. https://doi.org/10.1016/j.engfailanal.2005.04.009
47. Murakami Y (2012) Material defects as the basis of fatigue design. Int J Fatigue 41:2–10. https://doi.org/10.1016/j.ijfatigue.2011.12.001
48. Murakami Y (2019) Metal fatigue: effects of small defects and nonmetallic inclusions. Academic Press
49. Vincent M, Nadot Y, Nadot-Martin C, Dragon A (2016) Interaction between a surface defect and grain size under high cycle fatigue loading: experimental approach for Armco iron. Int J Fatigue 87:81–90. https://doi.org/10.1016/j.ijfatigue.2016.01.013
50. Li W, Sakai T, Wakita M, Mimura S (2014) Influence of microstructure and surface defect on very high cycle fatigue properties of clean spring steel. Int J Fatigue 60:48–56. https://doi.org/10.1016/j.ijfatigue.2013.06.017
51. Shakeri I, Danielsen HK, Tribhou A, Fæster S, Mishin OV, Eder MA (2022) Effect of manufacturing defects on fatigue life of high strength steel bolts for wind turbines. Eng Fail Anal 141:106630. https://doi.org/10.1016/j.engfailanal.2022.106630
52. Jiang Q, Sun C, Liu X, Hong Y (2016) Very-high-cycle fatigue behavior of a structural steel with and without induced surface defects. Int J Fatigue 93:352–362. https://doi.org/10.1016/j.ijfatigue.2016.05.032
53. Karr U, Schuller R, Fitzka M, Schönbauer B, Tran D, Pennings B, Mayer H (2017) Influence of inclusion type on the very high cycle fatigue properties of 18Ni maraging steel. J Mater Sci 52:5954–5967. https://doi.org/10.1007/s10853-017-0831-1
54. Schönbauer BM, Mayer H (2019) Effect of small defects on the fatigue strength of martensitic stainless steels. Int J Fatigue 127:362–375. https://doi.org/10.1016/j.ijfatigue.2019.06.021
55. Niu M, Wang Y, Song K, Wang Q, Zhao Y, Yan Y (2021) An adaptive pyramid graph and variation residual-based anomaly detection network for rail surface defects. IEEE Trans Instrum Meas 70:1–13. https://doi.org/10.1109/TIM.2021.3125987