metals

MDPI

Article

Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network

Ihor Konovalenko [1], Pavlo Maruschak [1], Vitaly Brevus [2] and Olegas Prentkovskis [3,*]

[1] Department of Industrial Automation, Ternopil National Ivan Puluj Technical University, Rus'ka Str. 56, 46001 Ternopil, Ukraine; ic@tu.edu.te.ua (I.K.); maruschak.tu.edu@gmail.com (P.M.)
[2] DataEngi LLC, Vienuolio g. 4A, LT-01104 Vilnius, Lithuania; v_brevus@tntu.edu.ua
[3] Department of Mobile Machinery and Railway Transport, Vilnius Gediminas Technical University, Plytinės g. 27, LT-10105 Vilnius, Lithuania
* Correspondence: olegas.prentkovskis@vilniustech.lt

Abstract: Classification of steel surface defects in steel industry is essential for their detection and also fundamental for the analysis of causes that lead to damages. Timely detection of defects allows to reduce the frequency of their appearance in the final product. This paper considers the classifiers for the recognition of scratches, scrapes and abrasions on metal surfaces. Classifiers are based on the ResNet50 and ResNet152 deep residual neural network architecture. The proposed technique supports the recognition of defects in images and does this with high accuracy. The binary accuracy of the classification based on the test data is 97.14%. The influence of a number of training conditions on the accuracy metrics of the model have been studied. The augmentation conditions have been figured out to make the greatest contribution to improving the accuracy during training. The peculiarities of damages that cause difficulties in their recognition have been studied. The fields of neuron activation have been investigated in the convolutional layers of the model. Feature maps which developed in this case have been found to correspond to the location of the objects of interest. Erroneous cases of the classifier application have been considered. The peculiarities of damages that cause difficulties in their recognition have been studied.

Keywords: steel sheet; surface defects; visual inspection technology; classification; neural network

Citation: Konovalenko, I.; Maruschak, P.; Brevus, V.; Prentkovskis, O. Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network. *Metals* **2021**, *11*, 549. https://doi.org/10.3390/met11040549

Academic Editor: Luis Norberto López De Lacalle

Received: 25 February 2021
Accepted: 25 March 2021
Published: 28 March 2021

1. Introduction

Rolled metal is one of the main products of ferrous metallurgy. It is widely used for the manufacture of metal structures. However, since a significant part of the finished products is characterized by surface defects, the production technology needs to be improved [1]. Some defects result from the original workpiece. Some are related to the chemical composition of steels. And some are associated with the shortcomings of the rolling equipment, its adjustment, calibration, wear, etc. When combined, violations of the casting and rolling technology contribute to the impairment of the rolled metal [2–4].

In this regard, establishing a quantitative relationship between the surface defect of the workpiece and technological factors of the rolling process is of scientific and practical interest. This will allow us to develop rational technological conditions [5].

The condition of the rolled surface is monitored using the automated optical-digital control systems, which make it possible to identify defects in real time, as well as recognize and classify them [6,7]. Such systems are based on a previous study of defect groups using materials from databases, technological and morphological analysis [8]. In previous articles, the types and causes of the main defects are analyzed, and their characteristics are described [9]. However, even within one class, defects may have significant differences in shape. This applies to a large group of defects, such as scratches, abrasions, striations. These defects differ not only in shape, size and orientation towards rolling direction, but also in origin. They are caused by friction (scratching) of the strip against the surface

of mechanical equipment. This is especially true for defects formed in the longitudinal direction [2,10,11].

Surface defects obtained by scratching have a small depth and clear edges. However, within this class, there are defects that are more similar to abrasions. It should be noted that in contrast to cracks, the bottom of scratches of both types can be seen clearly, and the top radius of the scratch is much larger than that of cracks. Scratches in the form of abrasions have a small depth, greater width, and "smoothed-out" profile. Such properties are additional features of defects of this class.

Differences in morphology require additional capabilities in terms of detection and recognition of defects in order to split scratches and striations into subclasses and generalize the most significant features of such defects. In our opinion, they indicate malfunctions of the technological equipment. Summarizing the results of the well-known studies, we can conclude that a system of methods, models and means of detecting technological defects of the rolled metal is in place to-date, and methodological principles of their use have been developed, which make it possible to solve a wide range of problems. However, since scratches, striations, and abrasions become more numerous on the rolled surface, we should develop fundamentally different approaches to their automated identification in the process of rolling, and take into account the diversity of their morphology.

In recent years, neural networks have been used for image analysis. Mikołajczyk et al. [12] developed a system for estimating the tool wear based on an artificial neural network. The neural network realized a classifier based on one category. The method for determining the wear rate of the tool based on image analysis was also presented. Visual Basic was used for the software realization of the proposed approach. A number of pixels belonging to the worn area were found as a result of the analysis. The method was investigated on an example of images showing the wear of the cutting tool edge at different working hours.

Ferreira and Giraldi [13] developed a convolutional neural network to classify granite tiles. The architecture of the neural network trained on the basis of CIFAR and MNIST was taken as a basis. An image database containing 1000 full-color RGB images with a size of 1500 × 1500 pixels divided into 25 classes of 40 images per class was used for training. The first 100 images were obtained by scanning the surface of the tiles, the remaining 900 by rotating the original images at different angles. The authors emphasized the effectiveness of their approach as applied to the analysis of high-resolution images.

In [14], a surface defect recognition system for checking steel sheets was proposed. It is based on a symmetrical map of the surface area and deep convolutional neural networks, which receive an image of the defect at the input and output of a label of one of the seven classes of defects. The proposed approach allowed achieving an overall accuracy of 99.05%.

The authors [15] developed few-shot learning with Siamese Neural Network using CNN structure with contrastive loss to classify defects with a small number of steel surface defect images. Typically, the training of a neural network requires a large amount of training data. The proposed approach facilitated the training of neural network models while using a smaller number of samples. The developed models allowed us to classify defects of steel surfaces with an accuracy of 86.5%.

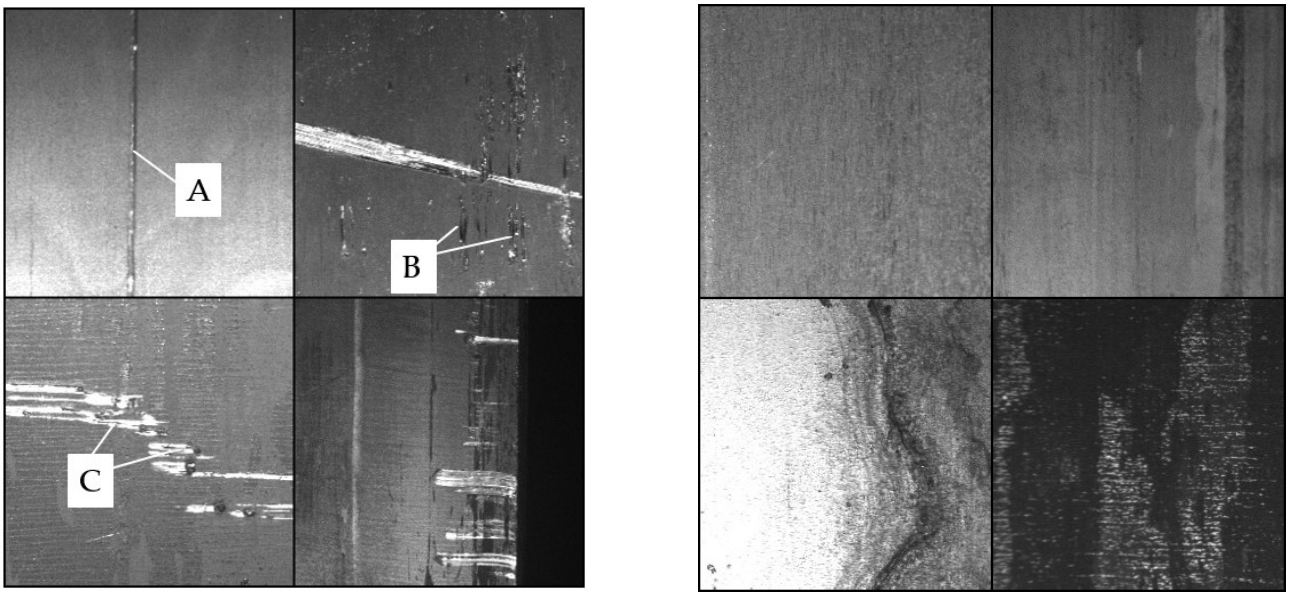
It is promising to create technological approaches that reduce defects in the area of plastic deformation of materials, in particular known solutions [16–18]. This article is also one of the steps to such technology, because the exact establishment of the nature of defects, their classification is the first step to diagnose the causes of their occurrence and further optimize the process of rolling strips.

Thus, the classification of defects of steel surfaces is an important task both for the recognition of defects and for the study of their causes. The understanding and the study of the features of defects facilitates their identification in a timely manner and allows to eliminate the causes of their occurrence, which, in turn, helps us reduce the number of production shortages and avoid cost reduction.

The aim of this research is to develop the means of identification of a surface damage found on the rolled metal strips with a view to a more accurate optical-digital defectoscopy of defects such as "scratches, striations, abrasions".

2. Defects

Defects (Figure 1a) are represented by scratches, scrapes and abrasions in the images. A scratch is a shallow, long defect made by something thin and sharp. A scrape is a scratch, with a torn profile. An abrasion is a surface defect caused by mechanical stress (shear) with an uneven in width profile. Abrasion can be both single and multiple. Scratches are usually oriented in a certain direction. Defects of this type are caused by friction of a hard surface against the sheet surface.



(**a**) (**b**)

Figure 1. Examples of scratches (A), scrapes (B) and abrasions (C) (**a**) and undamaged surfaces (**b**).

They are very common, have different shapes, directions and colors. Depending on the lighting direction and surface texture, they can be lighter or darker than the surface. The color range of such damage often changes within one area of the image. Defects can be both very thin and thick, occupy a significant surface area. The shape and appearance of defects can be quite different, which makes it difficult to identify them, even at the expert level. Damage is often found on a structurally inhomogeneous surface and cannot be found easily. Defects can end up in a smooth gradient, which also makes it difficult to detect them in the end zones. If the image contains only the disappearing side of a defect, it may not be recognized also.

3. Methodology

3.1. Training Dataset

The dataset for training the neural network classifier was formed on the basis of photo images of steel surfaces with damage such as scratches and abrasions. We have collected images of different sizes from open sources, which show metal surfaces with and without such damage. Most of the images were taken from the data provided by the Severstal Russian Public Company for the Kaggle International Competition entitled "Severstal: Steel Defect Detection" [19]. The database contains grayscale images measuring 1600 × 256 pixels.

In addition, images from the database of surface defects of the Northeastern University (NEU) were used [20]. This database contains six types of typical surface defects of

the hot-rolled steel strip, such as rolled scales, stains, cracks, dimples, inclusions, and scratches. The database includes 1800 images in grayscale, which show 300 specimens of each of the six different types of typical surface damage. The resolution of each image is 200 × 200 pixels. All the images of this database were resized to 256 × 256 pixels using the bicubic interpolation algorithm.

A total of 3938 images, which show defects of different shapes and directions, and 5447 images of surfaces without defects were used for training. The training, validation and test datasets were formed in such a way that the ratio between images with and without defects was the same. The training images were divided into three parts: test (it accounted for 10% of the total number of images), validation (15%) and training (75%). Training and validation dataset were used in the training of the neural network, and test was used to assess its quality on data unknown to it. The images show flat metal surfaces of two types: with and without scratches and abrasions (Figure 1). All images were reviewed and marked by experts.

3.2. Neural Network Classifier

The residual convolutional neural network (ResNet152) was chosen as the basis for the classifier. Since 2015, residual neural networks have shown excellent results in image processing [21]. Such networks choose low-, medium- and high-level features of the object of interest in a multilayer way. By increasing the number of stacked layers, we obtain more informative feature maps.

The ResNet model consists of residual blocks connected in series. Each block contains three layers, which perform 1 × 1 and 3 × 3 convolution. However, shortcut connections are the main feature of ResNet models. They pass one or more layers and connect the input of one layer with the output of another. Shortcut connections significantly reduce the problem of vanishing gradient in deeper layers of the network and significantly increase the depth of the model. This advantage makes it relatively easy to optimize ResNet networks, because the training error does not increase so drastically when the model depth is increased. Due to its structural features, even the 152-layer ResNet model is less complex than the 16-layer VGG-16 network [22] (11.3 vs. 15.3 billion of FLOP).

We used the transfer learning technique to avoid getting the neural network to detect signs at random. This approach is often used to improve the results and hasten the training process itself [23–25]. Therefore, prior to training, the neural network was initialized by weights obtained during training using the ImageNet dataset, which in total contains more than 1.4 million images divided into 1000 different classes. The general structure of the classifier is shown in Figure 2. The input layer of the classifier has a size of 256 × 256 neurons. It is followed by blocks of layers of the ResNet model. However, the last layer of the ResNet model was removed, and a 1-neuron layer was attached instead, which performs the classification. Thus, the model is a binary classifier. The sigmoid activation function is used in the output layer. Therefore, the value is in the range of 0–1 at the output of each neuron. The presence of a defect in the input image causes a value close to 1 to appear on the output neuron (or a value close to 0 in the opposite case).

Since the training dataset is not exhaustive, the augmentation technique was used to expand it. This contributes to the development of the best generalizing properties of the model [26–28]. In the context of solving our problem, previous research has also shown that the augmentation allows us to achieve better performance of the model [29]. Therefore, we used the random crop technique to train the neural network classifier [30]. In this case, plots of 256 × 256 pixels (according to the input layer shape) were selected randomly on the input images. These plots formed the tensor fed to the input of the neural network. In addition, each cropped frame of the image was transformed at random (horizontal and vertical flip, rotation by a multiple of 90°, random zoom in and out). This approach allowed us to significantly diversify training data and provided conditions, under which training batches are never repeated in practice.

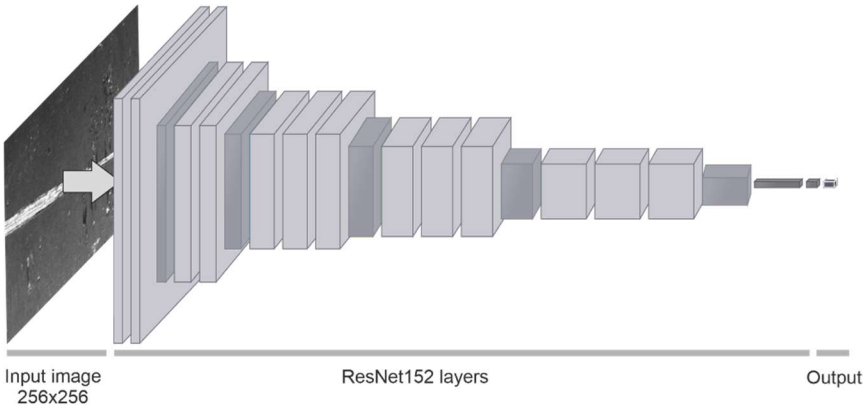


Figure 2. Generalized structure of the defect classifier based on ResNet.

Some defects shown in the image have quite small linear dimensions (for example, narrow scratches and striations). If an object of this type is cut during cropping, it will be difficult to recognize it even for an expert. An algorithm was developed that forms a set of valid frame positions for each image to avoid a very small area of the object (s) of interest getting into the frame during random framing. This algorithm allows us to reduce the number of false negatives of the model without interfering with the recognition of uncut small objects. Note that this approach was not applied to objects with linear dimensions greater than 15 pixels: they could partially fall into the frame during random framing.

Neural network classifiers were realized in Python 3.6 using the Keras 2.2.4-tf and TensorFlow 1.14.0 libraries. We used a workstation based on Intel Core i7-2600 CPU, 32 GiB RAM, and two NVIDIA GeForce GTX 1060 GPUs with 6 GiB of video memory for training and testing.

In the previous investigations [29] studied the work of residual neural networks with optimizers Stochastic Gradient Descent (SGD), Adam and RMSProp. It was found that the best results were achieved with the SGD optimizer in combination with the focal loss function [31]. Therefore, further training was performed with SGD optimizer and focal loss. Focal loss function defined as:

$$\begin{gathered} FL(p_t) = -(1-p_t)^{\gamma} log(p_t), \\ p_t = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise \end{cases}, \end{gathered} \tag{1}$$

where y specifies the ground-truth class, $p \in [0,1]$ is the model's estimated probability for the class with label $y = 1$ and $\gamma \geq 0$ is a tunable focusing parameter that smoothly adjusts the rate at which easy examples are downweighted.

Focal loss (1) applies a modulating term to the cross-entropy loss to be able to focus training on hard detective examples. It down-weights the well-classified examples and puts more training emphasis on the data that is hard to classify.

A number of models based on ResNet50 and ResNet152 architectures with different hyperparameters were trained. Models were trained at a learning rate of 0.01 and 0.001. Three different types of augmentation were used:

(a) Minimal: random crop only,
(b) Basic: random crop and for each crop random flip horizontal, flip vertical or flip both,
(c) Extended: random crop and for each crop random flip horizontal, flip vertical, flip both or random zooming in/out.

In total, 12 classifiers were trained (Table 1).

Table 1. Training conditions of models.

Model No	Architecture	Initial Learning Rate	Augmentation
1	ResNet50	0.01	a
2	ResNet50	0.01	b
3	ResNet50	0.01	c
4	ResNet50	0.001	a
5	ResNet50	0.001	b
6	ResNet50	0.001	c
7	ResNet152	0.01	a
8	ResNet152	0.01	b
8	ResNet152	0.01	c
10	ResNet152	0.001	a
11	ResNet152	0.001	b
12	ResNet152	0.001	c

During training and validation, the batch size was 10, the number of steps per epoch was 3000, and the number of validation steps was 1000. Training began at some initial learning rate. When the focal loss function (1) did not decrease over 10 epochs, the learning rate was reduced by 25%. The training of the model was stopped when the learning rate became less than 0.0001. According to the chosen learning strategy, different neural networks could learn over a different number of epochs.

3.3. Model Evaluation Metrics

At the end of each epoch, the model was preserved, along with the quality metrics of binary accuracy, precision, recall, and f1-score.

The recall metric shows the ability of a model to find all the relevant cases within a dataset. The recall equals the number of true positives divided by the number of true positives plus the number of false negatives. True positives are cases classified as positive by the model that actually is correct, and false negatives are cases the model identifies as negative that actually are incorrect:

$$Recall = \frac{TP}{(TP + FN)}. \tag{2}$$

The precision metric expresses the proportion of the cases that model predicts as positive actually were correct:

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

The f1 score is the harmonic mean of precision and recall taking both metrics into account in the following equation. The contributions of both precision and recall metrics to the f1 score are equal:

$$F1 = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + Recall)}. \tag{4}$$

The accuracy is the fraction of predictions our model got right:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5}$$

Accuracy is a common evaluation metric for classification problems. This metric shows the proportion of the total number of predictions that were correct.

3.4. Models Training and Quality Estimation

The variation dynamics of the accuracy metric (5) and loss function (1) when training models 11 and 12 (Table 1) is shown in Figure 3. Over several epochs, models have attained the maximum degree of generalization, and the validation loss has begun increasing gradually, while the training loss keeps decreasing. The validation binary accuracy during the training of both models varied from approximately 0.945 to 0.975, with model 12 achieving better accuracy on validation data.

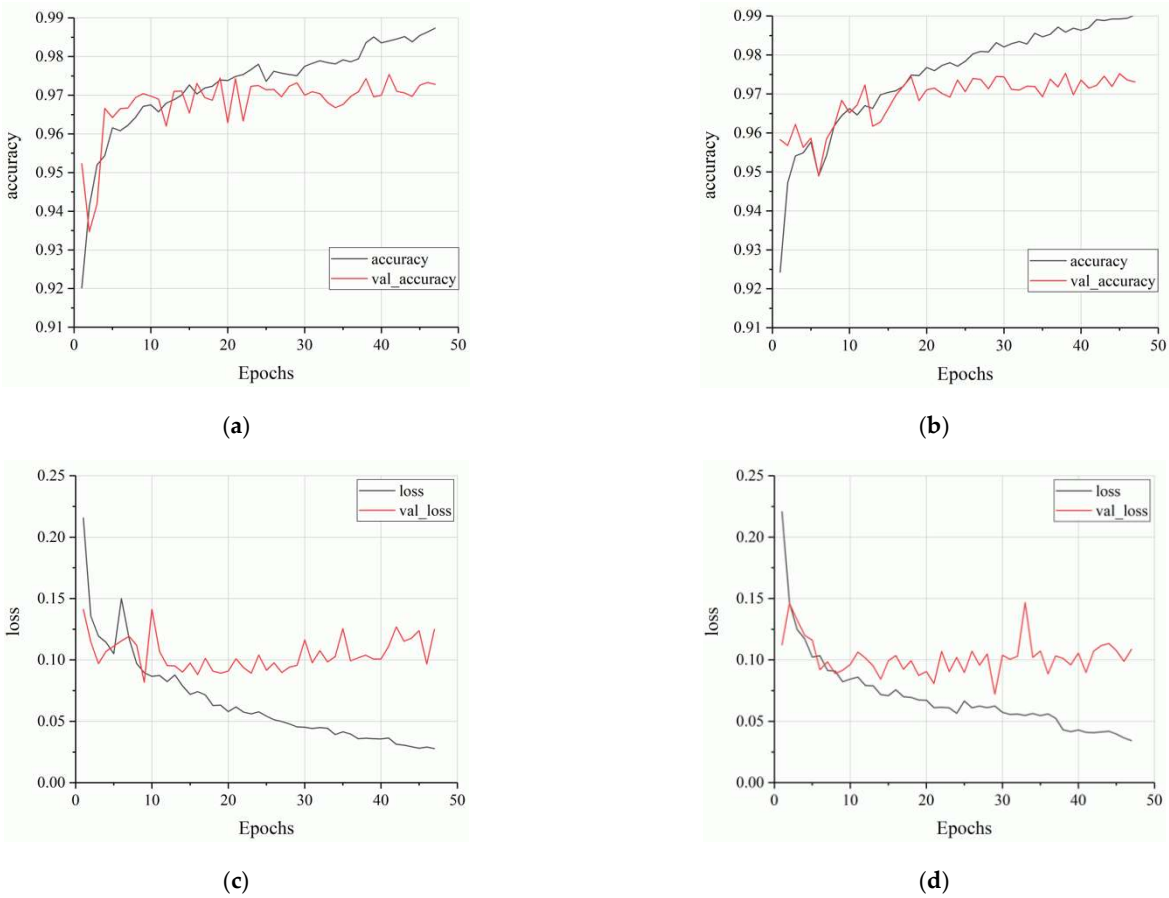


Figure 3. Variation dynamics of training and validation accuracy (**a**,**b**) and training and validation loss (**c**,**d**) during the training of model 11 (**a**,**c**) and model 12 (**b**,**d**).

However, the accuracy metric (5) does not fully reflect the quality of the model. The distribution of images is uneven, and the training dataset is dominated by images without defects (58%). In addition, since the random crop technique is used in the training process, areas without defects can get into random frames when images are processed. In this case, the overall accuracy metric will indicate the successful detection of defect-free images. Therefore, other metrics, such as recall (2), precision (3) and f1-score (4) metrics, were taken into account when studying the quality of the model.

Since the sigmoid activation function is used in the output layer of the classifier, its value is in the range of [0 . . . 1] at the output. This value can be interpreted as the degree of the model confidence in the presence of a defect in the image. The final decision about the presence of damage can be made if the output neuron value exceeds a certain threshold. The value of this limit affects the quality metrics of the model. To select the optimal threshold value, f1-score model metrics was calculated for all models for thresholds from 0.05 to 0.95 with a step of 0.01. The f1-score metric is the harmonic mean value of the precision and recall metrics, and characterizes the model's ability to recognize class defects. Graphs showing the changes in the f1-score metric (4) are provided in Figure 4.

Based on the obtained data, the optimal threshold values for the obtained models were selected (0.36 and 0.41, respectively). Further research was performed on models based on selected thresholds.

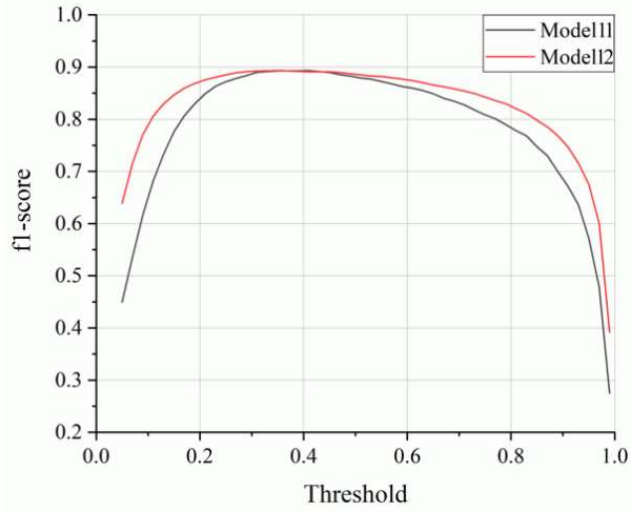


Figure 4. Changing the f1 score metric for different thresholds of model 11 and model 12.

The quality metrics of the models based on the test data are given in Table 2.

Table 2. Quality metrics of classifiers based on the test data.

Model No	Accuracy	Precision	Recall	f1 Score
1	0.9595	0.9894	0.7692	0.8655
2	0.9654	0.9780	0.8139	0.8885
3	0.9664	0.9764	0.8212	0.8921
4	0.9641	0.9826	0.8020	0.8831
5	0.9674	0.9532	0.8488	0.8980
6	0.96757	0.9713	0.8328	0.8967
7	0.9666	0.9608	0.8362	0.8942
8	0.9662	0.9644	0.8307	0.8926
9	0.9686	0.9685	0.8419	0.9008
10	0.9647	0.9134	0.8733	0.8929
11	0.9682	0.9381	0.8696	0.9025
12	0.9714	0.9397	0.8869	0.9125

Studies have shown that the augmentation method has the greatest impact on the final accuracy of defect recognition during training. For models based on ResNet50, the best result was achieved with basic augmentation. Using the variety of training samples to extended augmentation led to a slight decrease in quality metrics. For ResNet152-based models, the best result was achieved with extended augmentation. Obviously, a larger volume of neurons allows this model to process and generalize a larger number of features of defects that are formed during extended augmentation. According to the f1-score metric, the best models are 11 and 12 based on ResNet152.

The recall metric (2) shows the proportion of defective images recognized by the model as containing defects. The highest recall value for the Model 12 was 0.8869. This means that the model detects almost 89% of defects among the test data. The precision metric (3) shows what proportion of images recognized as containing defects actually holds defects. The metric value of 0.9397 indicates that more than 93% of the images marked

as defective are such in fact. The overall accuracy metric (5) value 0.9714 shows that the model carries out a correct recognition of more than 97% of all images (both with and without defects).

The receiver operating characteristic (ROC-curve) of model 12 is given in Figure 5. It shows the ability of the binary classifier to recognize the input signal at different thresholds of the output signal. This curve represents the dependence of the true positive rate ($TPR = TP/(TP + FN)$) on the false positive rate ($FPR = FP/(FP + TN)$) at different thresholds at the output of the binary classifier. It is important that ROC curves are insensitive to the uneven class distribution: if the proportion between the number of defective and defect-free images changes, the ROC curve remains unchanged. However, the area under this curve (AUC-ROC with a maximum value of 1) is an integral indicator of the model quality, which summarizes its ability to distinguish a particular class. For the Model 2 classifier, the AUC-ROC area is 0.991.

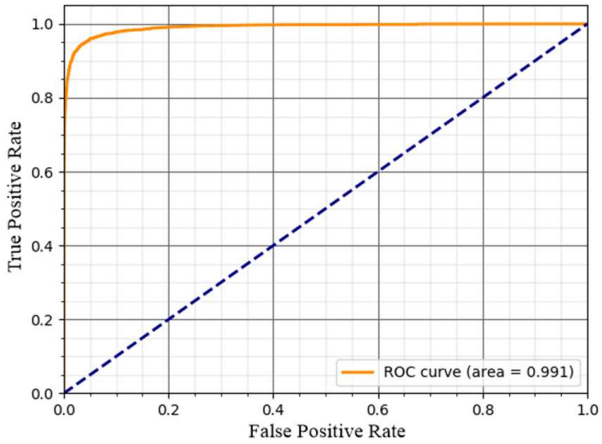


Figure 5. ROC-curve of the classifier of model 12.

To analyze false negatives and false positives, we constructed a confusion matrix for the classifier based on model 2 (Figure 6). For this purpose, a set of frames measuring 256 × 256 pixels (according to the size of the model input layer) was prepared on the basis of test images. A confusion matrix is a summary of prediction results on a classification task. This matrix allows us to see what difficulties arise when applying the model, and in what cases it can produce a wrong result. A binary classifier can produce two types of errors: it can either predict a defect when it does not actually exist (false positives), or predict the absence of a defect when it actually exists (false negatives).

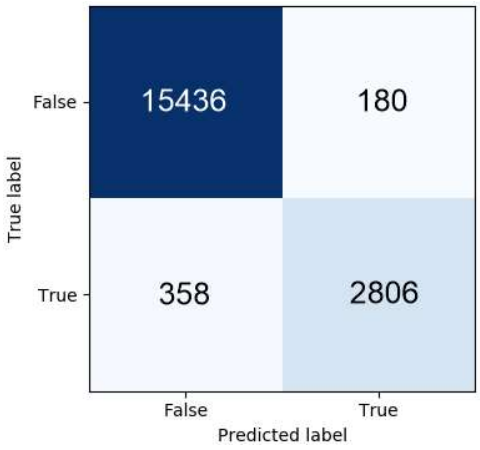
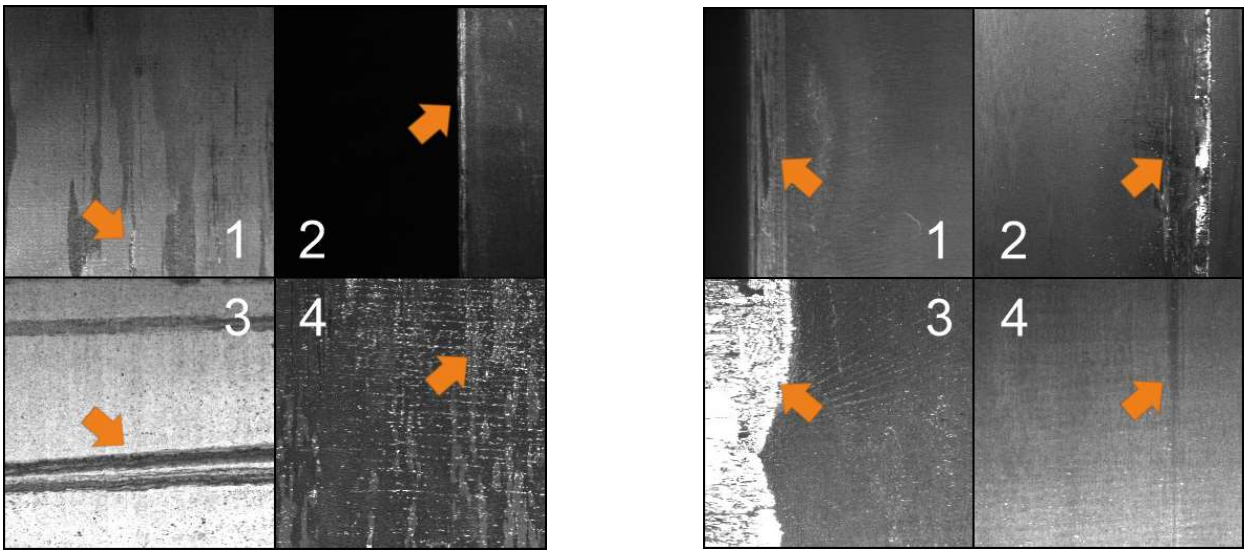


Figure 6. Confusion matrix of the classifier based on model 12.

False positives make up 1.1% of all images, which is only 1.4% of all images without defects. Thus, errors of this kind are quite rare. Figure 7a shows examples of images marked by the expert as not containing a defect, but recognized by the model as defective. The analysis of such images shows that they usually contain artifacts that resemble fragments of damage. Thus, image Figure 7a (1) contains an area resembling a faint scratch. In Figure 7a (2), the model perceived the glare at the sample edge as a defect, which is also similar to a scratch. Figure 7a (3) contains multi-colored drawings perceived as damage. The surface in Figure 7a (4) has characteristic formations, which resemble a small abrasion under certain lighting.



(**a**) (**b**)

Figure 7. False positives (**a**) and false negatives (**b**) classification cases for Model 12.

The main error rate is provided by false negatives (2.3% of all images, or 13.9% of images with defects). Examples of images with such errors for model 12 are shown in Figure 7b. Figure 7b (1,4) contains well-visible but low-gradient scratches. Defects in Figure 7b (1,2) are similar to the patterns that is sometimes present on defect-free surfaces. The defect in Figure 7b (3) is clearly visible and occupies a significant surface area, but the light makes the defect area highly contrastive, with no gradients of intensity typical to the damage.

3.5. Grad-CAM Class Activation Maps

When the model predicts the presence of a defect in the image, the corresponding fields of neurons are activated in the convolutional layers. We have investigated the appearance of feature maps, which are formed by convolutional layers of the neural network. The picture of the intermediate neuron activation shows how successfully the convolutional neural network converts the input signal and detects informative features [32,33]. The activation map makes it possible to see how the input image is decomposed by various filters generated during the neural network training. The first layers of the model contain the whole image, but with an emphasis on certain areas. At this stage, the model retains most of the information from the input image, but already focuses on its most interesting features. Deeper convolutional layers reveal increasingly more characteristic features of the image, so the picture of activation becomes more abstract.

In this case, the most interesting are feature maps from the last convolutional layer, which contains 2048 feature maps measuring 8×8. A more detailed study showed that damage of various types, sizes and shapes is quite fully represented in the feature maps. To make sure that the model predicts the presence of damage based on the objects of interest,

but not secondary elements of the image, we used Gradient-weighted Class Activation Mapping (Grad-CAM) [34]. Grad-CAM allows to get an interpretation of the model results by visualizing the heatmap of activation of neurons that were activated during the prediction of a particular class for a given sample. Using Grad-CAM, we can visually check which areas of the image the model focuses on, making sure that it is actually looking at the correct patterns in the image and activating around them.

Visualization of neuron activation maps of model 12 by the Grad-CAM method is shown in Figure 8. The heatmap is superimposed on the initial images (the larger value at the output of neurons is marked by the orange color, and the smaller one by the blue color). As seen from the given images, the sites of convolutional layers, which correspond to the initial image containing the defect, are activated. This confirms that the model focuses on their features when detecting defects.

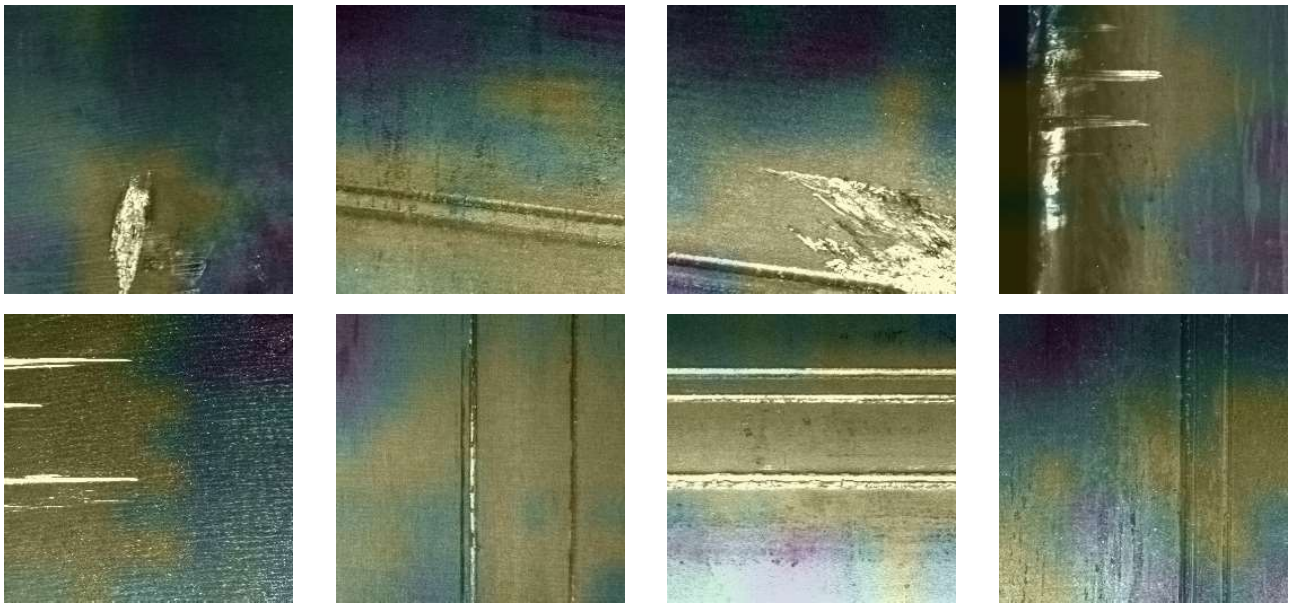


Figure 8. Visualization of activation maps of neurons when damage is detected.

3.6. Analysis of Images of Any Size

The size of the input layer of the neural network classifier is 256 × 256 pixels, but real images usually have other sizes. An algorithm with a sliding window was developed for the analysis of such images. In this case, frames measuring 256 × 256 pixels are cut from the initial image in steps of 3⁄4 of the image width and height (Figure 9). Based on the frame package, a tensor is formed, which is fed to the input of the classifier.

Fragments have to overlap so as to eliminate the influence of the edge effect on the defects found at the edge of a particular frame. Such defects can be cut and presented only in part. In another frame, such defects are closer to the middle and, therefore, are more suitable for the analysis.

A decision on the presence of a defect in the image is made taking into account the results of prediction for all frames obtained from the image. If at least one of them is found to be defective, the image as a whole is also marked as defective.

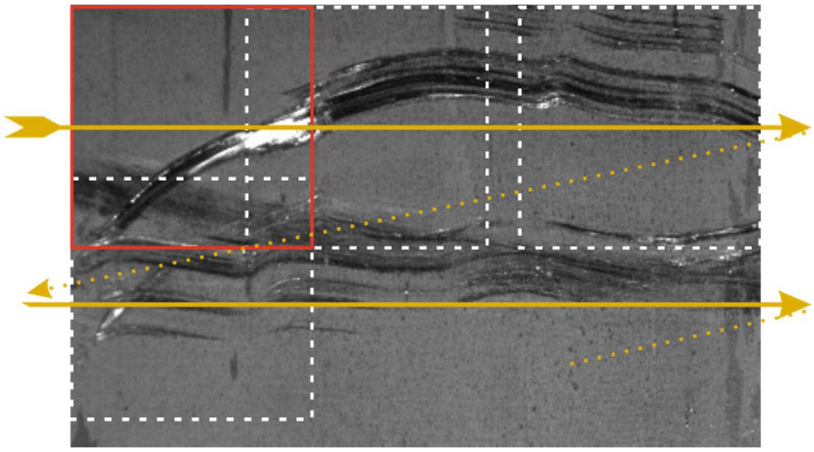


Figure 9. Sliding window for recognizing defects in the image of any size.

4. Conclusions

A classifier based on a residual neural network has been developed to recognize damage to metal surfaces such as scratches, scrapes and abrasions. It has been trained and researched on a set of "Severstal: Steel Defect Detection" images. The best model classifier is based on the ResNet152 deep convolutional neural network. It was found out that a significant contribution to improving the quality of education is made by the augmentation of training images. Models with depths of 50 and 152 layers have been considered in different conditions (in particular, with different optimizers, loss functions and augmentation conditions). The best hyperparameters of model have been chosen by comparison of recall, precision, f1-score and accuracy metrics.

The best recognition quality metrics are achieved using augmentation and focal loss function. The trained model makes it possible to recognize defects in the images and it is done with high accuracy. The accuracy of the classification based on the test data is 97.1% for all images. The model detects 88.7% of images with defects, with the precision reaching 94.0%.

The study has shown that most errors are due to false positives, which make 11.3% of images with defects. Moreover, the model often makes mistakes in case of significant visual similarity between surface artifacts and defects. Features of defects leading to their erroneous recognition are considered.

The fields of neuron activation were investigated in the convolutional layers of the model. Feature maps formed in this case were found to reflect the features of the position, size and shape of the objects of interest. The areas of the convolutional layers that corresponded to the original image containing the defect have been activated. Thus, the model has focused on the defect features when detecting defects. This suggests that a tool for semantic segmentation can be built based on the proposed models of neural networks.

The results obtained can be useful in improving the algorithms for analyzing the operation of rolling mills and the parameters of their adjustment. It is also the basis of the analytical assessment of the surface depending on the rolling parameters and other technological factors [35].

Author Contributions: Conceptualization, P.M.; formal analysis, I.K.; investigation, I.K., P.M., O.P. and V.B.; methodology, I.K.; project administration, I.K. and O.P.; validation, I.K., P.M. and V.B.; visualization, I.K.; writing—original draft, I.K. and P.M.; writing—review and editing, I.K., P.M. and O.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.kaggle.com/c/severstal-steel-defect-detection/overview (accessed on 27 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ke, X.; Chaolin, Y. On-line defect detection algorithms for surface inspection of hot rolled strips. In Proceedings of the 2010 International Conference on Mechanic Automation and Control Engineering, Wuhan, China, 26–28 June 2010; pp. 2350–2353.
2. Becker, D.; Bierwirth, J.; Brachthäuser, N.; Döpper, R.; Thülig, T. *Zero-Defect-Strategy in the Cold Rolling Industry. Possibilities and Limitations of Defect Avoidance and Defect Detection in the Production of Cold-Rolled Steel Strip*; Fachvereinigung Kaltwalzwerke e.V.; CIELFFA: Düsseldorf, Germany, 2019; p. 16.
3. Chu, M.-X.; Wang, A.-N.; Gong, R.-F.; Sha, M. Multi-class classification methods of enhanced LS-TWSVM for strip steel surface defects. *J. Iron Steel Res. Int.* **2014**, *21*, 174–180. [CrossRef]
4. Nioi, M.; Celotto, S.; Pinna, C.; Swart, E.; Ghadbeigi, H. Surface defect evolution in hot rolling of high-Si electrical steels. *J. Mater. Process. Technol.* **2017**, *249*, 302–312. [CrossRef]
5. Ren, Q.; Geng, J.; Li, J. Slighter faster R-CNN for real-time detection of steel strip surface defects. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 2173–2178.
6. Cong, J.-H.; Yan, Y.-H.; Zhang, H.-A.; Li, J. Real-time surface defects inspection of steel strip based on difference image. In Proceedings of the International Symposium on Photoelectronic Detection and Imaging: Related Technology and Applications 2007, Beijing, China, 9–12 September 2007; Volume 6625, p. 66250W.
7. Amin, D.; Akhter, S. Deep learning-based defect detection system in steel sheet surfaces. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; pp. 444–448.
8. Yun, J.P.; Shin, W.C.; Koo, G.; Kim, M.S.; Lee, C.; Lee, S.J. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *J. Manuf. Syst.* **2020**, *55*, 317–324. [CrossRef]
9. Konovalenko, I.; Maruschak, P.; Brezinová, J.; Viňáš, J.; Brezina, J. Steel surface defect classification using deep residual neural network. *Metals* **2020**, *10*, 846. [CrossRef]
10. Brezinová, J.; Viňáš, J.; Brezina, J.; Guzanová, A.; Maruschak, P. Possibilities for renovation of functional surfaces of backup rolls used during steel making. *Metals* **2020**, *10*, 164. [CrossRef]
11. Yu, H.-L.; Tieu, K.; Lu, C.; Deng, G.-Y.; Liu, X.-H. Occurrence of surface defects on strips during hot rolling process by FEM. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 1161–1170. [CrossRef]
12. Mikołajczyk, T.; Nowicki, K.; Kłodowski, A.; Pimenov, D. Neural network approach for automatic image analysis of cutting edge wear. *Mech. Syst. Signal Process.* **2017**, *88*, 100–110. [CrossRef]
13. Ferreira, A.; Giraldi, G. Convolutional neural network approaches to granite tiles classification. *Expert Syst. Appl.* **2017**, *84*, 1–11. [CrossRef]
14. Yi, L.; Li, G.; Jiang, M. An end-to-end steel strip surface defects recognition system based on convolutional neural networks. *Steel Res. Int.* **2017**, *88*, 87. [CrossRef]
15. Kim, M.S.; Park, T.; Park, P. Classification of steel surface defect using convolutional neural network with few images. In Proceedings of the 12th Asian Control Conference (ASCC), Kitakyusyu International Conference Center, Fukuoka, Japan, 9–12 June 2019; pp. 1398–1401.
16. Urbikain, G.; Alvarez, A.; De Lacalle, L.N.L.; Arsuaga, M.; Alonso, M.A.; Veiga, F. A reliable turning process by the early use of a deep simulation model at several manufacturing stages. *Machines* **2017**, *5*, 15. [CrossRef]
17. Bustillo, A.; Urbikain, G.; Perez, J.M.; Pereira, O.M.; de Lacalle, L.N.L. Smart optimization of a friction-drilling process based on boosting ensembles. *J. Manuf. Syst.* **2018**, *48*, 108–121. [CrossRef]
18. Zhao, W.; Chen, F.; Huang, H.; Li, D.; Cheng, W. A new steel defect detection algorithm based on deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*. [CrossRef]
19. Kaggle. Severstal: Steel Defect Detection. Can You Detect and Classify Defects in Steel? 2019. Available online: https://www.kaggle.com/c/severstal-steel-defect-detection (accessed on 27 March 2021).
20. Northeastern University. Available online: https://www.kaggle.com/kaustubhdikshit/neu-surface-defect-database (accessed on 27 March 2021).
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.15561556v6.
23. Taylor, M.E.; Stone, P. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.* **2009**, *10*, 1633–1685.
24. Khan, S.; Islam, N.; Jan, Z.; Din, I.U.; Rodrigues, J.J.P.C. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognit. Lett.* **2019**, *125*, 1–6. [CrossRef]
25. Maqsood, M.; Nazir, F.; Khan, U.; Aadil, F.; Jamal, H.; Mehmood, I.; Song, O.-Y. Transfer learning assisted classification and detection of Alzheimer's disease stages using 3D MRI scans. *Sensors* **2019**, *19*, 2645. [CrossRef]
26. Yu, X.; Wu, X.; Luo, C.; Ren, P. Deep learning in remote sensing scene classification: A data augmentation enhanced convolutional neural network framework. *GISci. Remote Sens.* **2017**, *54*, 741–758. [CrossRef]
27. Eyobu, O.S.; Han, D.S. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors* **2018**, *18*, 2892. [CrossRef]

28. Han, D.; Liu, Q.; Fan, W. A new image classification method using CNN transfer learning and web data augmentation. *Expert Syst. Appl.* **2018**, *95*, 43–56. [CrossRef]
29. Konovalenko, I.; Maruschak, P.; Prentkovskis, O.; Junevičius, R. Investigation of the rupture surface of the titanium alloy using convolutional neural networks. *Materials* **2018**, *11*, 2467. [CrossRef]
30. Takahashi, R.; Matsubara, T.; Uehara, K. Data augmentation using random image cropping and patching for deep CNNs. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 2917–2931. [CrossRef]
31. Lin, T.-Y.; Goyal, P.; Girshick, R.B.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
32. Chollet, F. *Deep Learning with Python*; Manning Publications: Shelter Island, NY, USA, 2017; p. 313.
33. Wang, S.; Xia, X.; Ye, L.; Yang, B. Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals* **2021**, *11*, 388. [CrossRef]
34. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]
35. Tishchenko, D.A. Development of Control Algorithms, Modes of Preparation and Operation of Work Rolls of the Finishing Group of a Continuous Wide-Strip Hot Rolling Mill to Ensure the Quality of Rolled Products. CSc. (Eng.) Dissertation, JSC, Institute Tsvetmetobrabotka, Moscow, Russia, 2006; p. 160. (In Russian).