applied sciences

MDPI

Article

Steel Surface Defect Classification Based on Small Sample Learning

Shiqing Wu *, Shiyu Zhao, Qianqian Zhang, Long Chen and Chenrui Wu

School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; zsy_4992@163.com (S.Z.); Zqq6183@163.com (Q.Z.); cl@usst.edu.cn (L.C.); wuchenrui@usst.edu.cn (C.W.)
* Correspondence: wsq19870612@163.com

Abstract: The classification of steel surface defects plays a very important role in analyzing their causes to improve manufacturing process and eliminate defects. However, defective samples are very scarce in actual production, so using very few samples to construct a good classifier is a challenge to be addressed. If the layer number of the model with proper depth is increased, the model accuracy will decrease (not caused by overfit), and the training error as well as the test error will be very high. This is called the degradation problem. In this paper, we propose to use feature extraction + feature transformation + nearest neighbors to classify steel surface defects. In order to solve the degradation problem caused by network deepening, the three feature extraction networks of Residual Net, Mobile Net and Dense Net are designed and analyzed. Experiment results show that in the case of a small sample number, Dense block can better solve the degradation problem caused by network deepening than Residual block. Moreover, if Dense Net is used as the feature extraction network, and the nearest neighbor classification algorithm based on Euclidean metric is used in the new feature space, the defect classification accuracy can reach 92.33% when only five labeled images of each category are used as the training set. This paper is of some guiding significance for surface defect classification when the sample number is small.

Keywords: surface defect classification; feature extraction; few samples

Citation: Wu, S.; Zhao, S.; Zhang, Q.; Chen, L.; Wu, C. Steel Surface Defect Classification Based on Small Sample Learning. *Appl. Sci.* **2021**, *11*, 11459. https://doi.org/10.3390/app112311459

Academic Editors: Xinyue Zhao, Zheng Chen and Ming Fang

Received: 10 November 2021
Accepted: 30 November 2021
Published: 3 December 2021

1. Introduction

In the hot rolling manufacturing process of strip steel, defects may occur on its surface due to processing technology, mechanical equipment and human errors. These defects will greatly change the mechanical properties of steel and weaken its quality. The classification of steel surface defects plays a very important role in analyzing their causes to improve manufacturing process and eliminate defects [1]. For example, rolled in scale is probably caused by severe peeling of the oxide film of the stand roll before rolling is finished, while the scratches are caused by the protrusions of the strip in the area of the rolling line, or the friction between the fixed roll and the strip surface. Traditionally, the surface quality of steel is detected by human observation. However, it depends on the worker's skill and experience, and continuous work will reduce the inspection accuracy and cause great harm to the worker's health. Meanwhile, defective samples are very scarce in the hot rolling process, which poses a challenge to the classification of steel surface defects.

Traditional machine learning algorithms, such as Support Vector Machine (SVM) and Decision Tree, can only learn some low-level features rather than detailed and abstract features. When two kinds of defects are similar, traditional machine learning algorithms may fail to classify them.

The core of deep learning is an artificial neural network. Similar to human nerves, artificial neural networks are composed of thousands of artificial neurons, with numerous nodes and tens of thousands of parameters that need to be learned. Generally speaking, when the number of samples is small, the model will not be able to learn the corresponding

features, and even overfitting will occur. In recent years, more and more researchers have been devoted to the field of small sample learning. Generally, there are four methods in the field of small sample learning, as described below.

Method based on meta-learning. It trains a meta model to learn the knowledge of different tasks, so that the trained model can be quickly generalized on new tasks, such as the Model-Agnostic Meta-Learning algorithm (MAML) proposed by Finn [2] and the Long Short Term Memory network (LSTM) proposed by Ravi [3]. The existing meta-learning methods mostly use an LSTM or Recurrent Neural Network (RNN) structure in the model, but the disadvantages are high time complexity and slow running speed. Therefore, it is not suitable for industrial application.

Method based on data enhancement. Since the small sample problem is caused by lack of samples, it can be solved by expanding the sample set. Inspired by this idea, models such as Generative Adversarial Networks (GAN) [4] have been developed. However, because the expanded samples are transformations of the original images, they are similar to the original images.

Method based on fine-tuning. This model is usually pre-trained on a large-scale dataset, after which the main layers of the entire network are frozen [5,6] and, finally, the parameters of the full connection layers or the top layers of the neural network model are fine-tuned on the target dataset. The limitation of this method is that it requires a certain similarity between the target dataset and the pre-trained source dataset. Because the characteristics of the samples in the existing large-scale datasets are very different from the characteristics of the steel surface defects, it is difficult to deal with the classification of steel surface defects in this method.

Method based on metric learning. Metric Learning is also called similarity learning or distance metric learning. It maps the features of the image to a new feature space. In this new feature space, a special measurement method is used to make the distance between similar samples as short as possible and the distance between heterogeneous samples as long as possible. Compared with the above three methods, its advantage is that the main goal of learning is the similarity between samples. Therefore, it is particularly suitable for classification. Some applications are the Prototypical Networks proposed by Snell [7], the Relation Network proposed by Sung [8] and Siamese neural networks proposed by Koch [9]. These studies adopt a four convolutional layer network as a feature extractor [7,10]; the network structure is relatively simple and the training speed is fast, but it is very dependent on a good feature space.

As to specific applications of small sample learning, the main focus is on hyperspectral images [11–13] and biological signals [14], while steel surface defects are relatively few. Min Su Kim [1] used a twin neural network based on L1 distance to classify steel surface defect samples, but the performance of the model was not good under small datasets. Guizhong Fu [15] used image enhancement to expand the dataset, and pre-trained Squeeze Net to realize the classification of steel surface defect, but the sample number is big. In this paper, the model of feature extraction + feature transformation + nearest neighbor was used to classify the steel surface defects in a small dataset. A neural network was used to extract the image features of steel surface defects, after which the extracted image features were transformed to a new feature space and, finally, the nearest neighbor algorithm was used for classification.

Section 2 introduces the core idea and advantages of three feature extraction networks, two effective feature transformation methods and the rules of nearest neighbor classification. Section 3 gives the settings and results of the experiment. The conclusions are given in Section 4.

2. Principles and Methodology

Traditional feature extraction methods, such as Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradient (HOG), greatly rely on manual design. The quality of the extracted image features often depends on the experience of technical per-

sonnel. It shows great uncertainty and greatly reduces the classification accuracy. With the continuous development of convolutional neural networks, many efficient and accurate networks emerged in feature extraction, such as AlexNet [16], VGG [17], GoogLeNet [18] and ResNet [19]. According to the findings of related research [16,20], the depth of the model plays a vital role. In this paper, we conducted theoretical analyses on three feature extraction networks, namely Residual Net, Mobile Net and Dense Net, after which the two feature transformation methods of mean subtraction and L2-normalization are demonstrated and, finally, the rules of nearest neighbor classification are illustrated theoretically.

2.1. Feature Extraction Network (FEN)

2.1.1. Residual Net

Deep Convolutional Neural Networks have made a series of breakthroughs in the field of image classification. According to the findings of related researches [16,20], the depth of the model plays a vital role, because the deep layer can learn the abstract features of the image. However, when the network is deepened, there will be gradient disappearance and serious degradation problems, which reduces the accuracy of the model. In order to solve this problem, Kaiming He [19] proposed the famous residual network model, as shown in Figure 1.

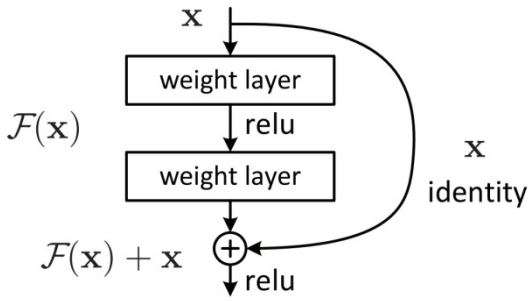

Figure 1. Residual block.

Assume that the desired underlying mapping is $H(x)$. We let the stacked nonlinear layer fit another mapping, that is $F(x) = H(x) - x$. Then, the original mapping is transformed into $F(x) + x = H(x)$. Compared with $H(x)$, $F(x)$ is easier to be fit. This can be illustrated by the extreme case, namely identity mapping. For an identity mapping $H(x) = x$, then $F(x) = 0$, obviously, fitting 0 is easier than fitting a stack of nonlinear layers x.

$F(x)$, mentioned above, is called a residual, and the residual learning algorithm is adopted on the stacked layer. A residual block is shown in Figure 1 and defined as:

$$y = F(x, \{W_i\}) + x \tag{1}$$

where x and y represent the input and output of the stacked layer, respectively, the function $F(x, \{W_i\})$ represents the learned residual mapping [10] and W_i depends on the operation of the non-linear layer.

In this paper, the standard 10/18/34 layer structure is adopted, and the size of the convolution kernel is 3×3. Residual Net10 contains four residual blocks, Residual Net18 contains eight residual blocks and Residual Net34 contains 16 residual blocks.

2.1.2. Mobile Net

In order to solve the problem of a sharp increase in the number of parameters caused by the deepening of the model, Howard [21] proposed a lightweight network with low latency and high response, named MobileNet. It mainly reduces the amount of model parameters by converting the standard convolution to depthwise separable convolution.

Depthwise separable convolution can be divided into two smaller convolutions, namely, depthwise convolution and pointwise convolution.

As shown in Figure 2, the convolution kernel of standard convolution acts on all input channels. However, every convolution kernel of the depthwise convolution corresponds to one input channel. There is little difference between pointwise convolution and standard convolution, except for a 1×1 convolution kernel of pointwise convolution.

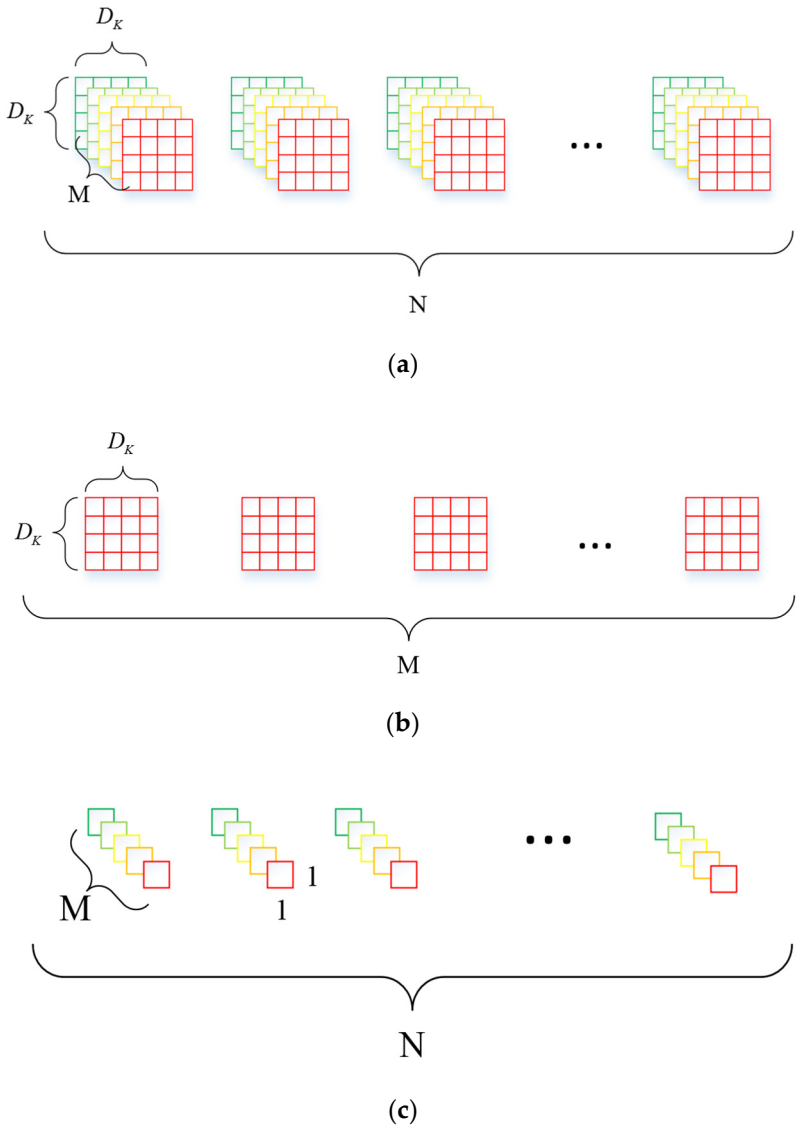

Figure 2. (**a**) Standard convolution kernel; (**b**) Depthwise Convolutional Filters; (**c**) Pointwise Convolutional Filters.

Assume that the size of the input feature image is $D_F \cdot D_F \cdot M$ (where D_F represents the width and height of the feature image and M is the number of channels), the size of the output feature image is $D_F \cdot D_F \cdot N$ (where N is the number of convolution kernels) and the convolution kernel size is $D_K \cdot D_K$.

Then the computational cost of the standard convolution is as follows:

$$N_{std} = D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F, \tag{2}$$

The computational cost of the depthwise convolution is as follows:

$$N_{dw} = D_K \cdot D_K \cdot M \cdot D_F \cdot D_F, \tag{3}$$

The computational cost of pointwise convolution is as follows:

$$N_{pw} = M \cdot N \cdot D_F \cdot D_F, \tag{4}$$

The total computational cost of depthwise separable convolution is as follows:

$$N_{ds} = N_{dw} + N_{pw} = D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F, \tag{5}$$

The ratio of the computational cost of the depthwise separable convolution to the computational cost of the standard convolution is as follows:

$$\frac{N_{ds}}{N_{std}} \cdot \frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}, \tag{6}$$

In this paper, the standard Mobile Net structure is adopted, and the size of the convolution kernel is 3×3. Theoretically, this method can reduce the computational cost compared with standard convolution, which is beneficial to the small sample learning of the neural network.

2.1.3. Dense Net

Dense Net, proposed by Huang [22], is very similar to Residual Net. They both connect feature images across network layers to solve the problem of model degradation caused by deepening the network. The difference is that the Residual block in Figure 1 is a single-line connection, while the dense block in Figure 3 used by Dense Net is a multi-line connection, that is, the input of each layer comes from the output of all the previous layers.

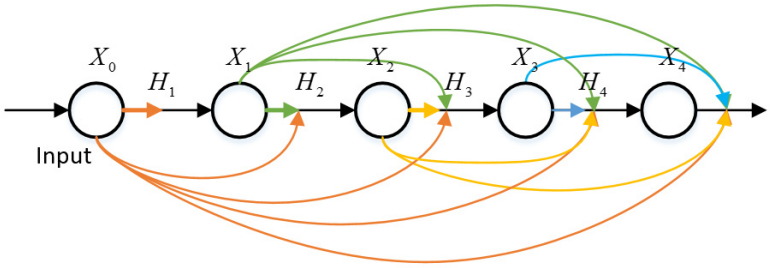

Figure 3. Dense block.

Assume that l represents the layer, X_l represents the output of the l layer and H_l represents the nonlinear transformation. For the traditional convolutional feedforward neural network, the output of the l layer is $X_l = H_l(X_{l-1})$. Because the residual block of Residual Net adds a shortcut, the output of its l layer should be $X_l = H_l(X_{l-1}) + X_{l-1}$. The dense block in Dense Net combines all the output according to the channels and stitches them, which is represented by the symbol []. Then, the output of its l layer is $X_l = H_l([X_0, X_1, \ldots, X_{l-1}])$. The features and gradients are transferred more effectively through the dense block of DenseNet. In this paper, the structure of DenseNet121 was adopted for the experiments.

2.2. Feature Transformation

Feature transformation is performed after the image features are extracted, so its input is the feature vector of the image. It plays a very important role as an intermediate bridge between feature extractor and classifier.

2.2.1. Mean Subtraction

Given a feature vector set X, it is composed of some multidimensional vectors x, then the mean value of the vector set is $\overline{x} = \frac{\sum x}{|X|}$. Then, mean subtraction operation $\hat{x} - \overline{x} \rightarrow \hat{x}$

is performed on the feature vector. It does not change the Euclidean distance between samples, but it can improve the classification accuracy [23].

2.2.2. L2-Normalization

Given a vector $x(x_1, x_2, \ldots, x_n)$, the L2 norm of the vector is $||x||_2 = (|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2)^{1/2}$. After obtaining its L2 norm, the vector x is normalize via $x \to \frac{x}{||x||_2}$.

2.3. Nearest Neighbor Algorithm

Once the feature extraction network f_α is trained, the subsequent operations are performed on the images in the feature space. $I = f_\alpha(input)$ is defined as the image after feature extraction. N-way K-shot settings are adopted, where N represents the number of classes in the training set and K represents the number of samples in each class. In the feature space, a certain distance metric, $d(I, I) \in R$, is used for nearest neighbor classification. For the one-shot setting, there is only one picture for each category in the training set D_{train}, $D_{train} = \{(I_1, L_1), (I_2, L_2), \ldots, (I_N, L_N)\}$, where L represents the category label. The nearest neighbor rule is to calculate the distance between the test image and the training image, and then assign the label of the training image with the smallest distance to the test image.

$$y(I) = argmin_{n \in \{1,\ldots,N\}} d\left(I, I'_n\right), \tag{7}$$

For the multi-shot setting, the prototype network is adopted. In the feature space, the average value of the sample feature vectors of each class in the training set is used as its class prototype, and then the distance between the test sample and class prototype is calculated according to Formula (7).

3. Experiments

In order to verify the influence of different feature extraction networks, feature transformation methods and network depth on the final classification results, the average accuracy of different feature extraction networks were measured through steel surface defect classification experiments. The running and testing environment of the algorithms in this paper is shown in Table 1.

Table 1. Running and testing environment.

OS	CPU	GPU	Pytorch	Python
Ubuntu 18.04.1	AMD R9-3900x	RTX 3090	1.8.0	3.7.0

3.1. Experiment Development

The publicly available NEU steel surface defects dataset was applied in this paper. The NEU dataset is divided into six categories, namely Crackle (Cr), Inclusion (In), Patches (PA), Pitted Surface (PS), Rolled in Scale (RS) and Scratch (Sc), as shown in Figure 4. And there are 300 samples (200 × 200 pixels) for each surface defect category. The image names and labels are saved in ImageNet format as a csv file.

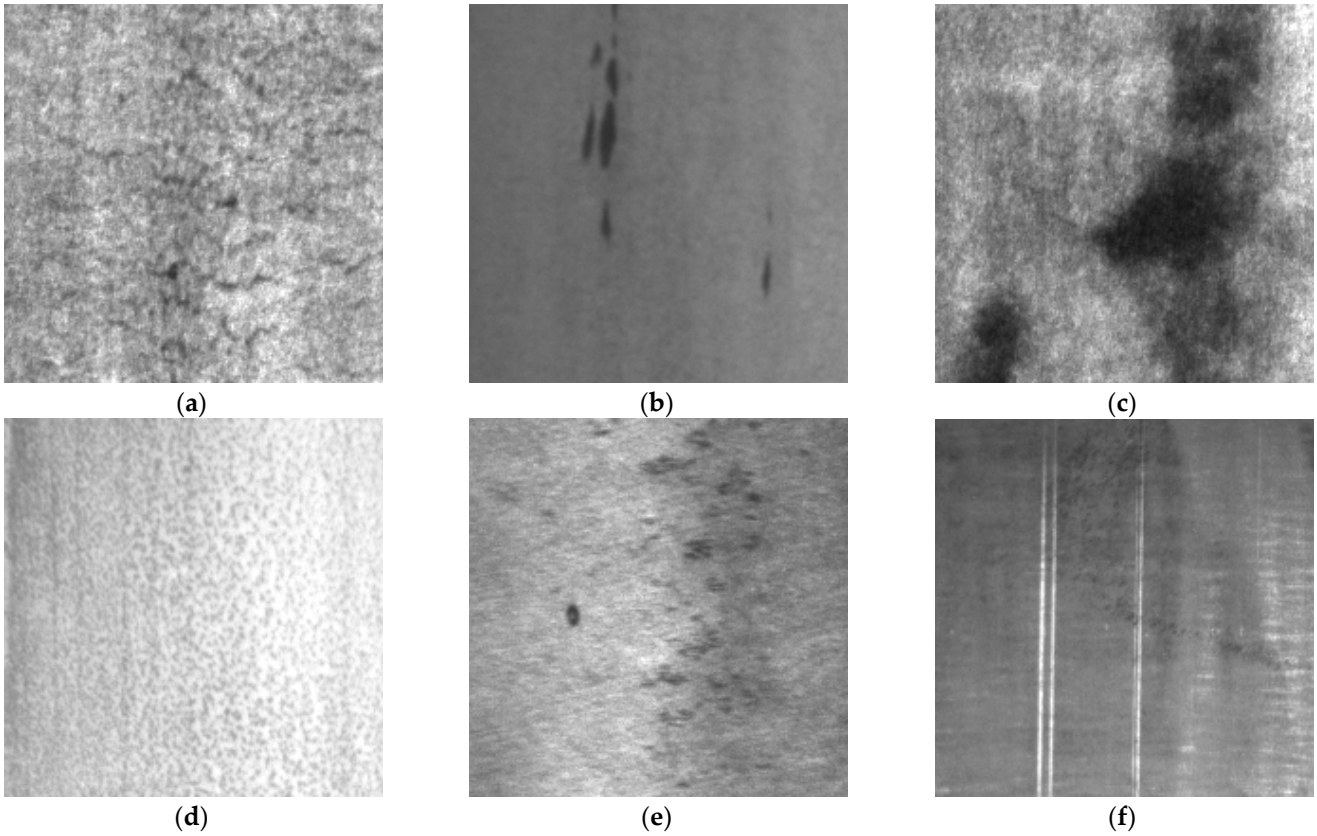

Figure 4. Categories of defects: (**a**) Cr, (**b**) In, (**c**) PA, (**d**) PS, (**e**) RS, (**f**) Sc.

The N-way K-shot setting was adopted to train and test the model. The classification networks were trained and tested according to the tactics of [10]. Compared with large datasets, the defect categories are much less. Therefore, the categories are no longer subdivided, and 1000 6-way K-shot tasks are constructed. In each task, there were six categories, and each category had K-labeled images to form the training set. At the same time, each category had 30 unlabeled images for testing. The average accuracy of the tests in all tasks (95% confidence level) was used as the criterion to evaluate different classification networks.

Stochastic gradient descent method, cross entropy loss and gradual decreasing learning rate [13] were adopted to train the network for 90 Epochs. The initial learning rate was 0.1, which was reduced to 1/10 of the original every 30 epochs, the default batch size was 124 and Euclidean distance was used as the metric of the nearest neighbor classifier.

3.2. Results and Analysis

During the training, 1000 1-shot tasks were collected every two epochs to calculate the average accuracy of different classification networks. Figure 5 shows that the performance of the feature extraction network using only four convolutional layer stacks was the worst. The reason is that it cannot learn the deep abstract features of the image. In the first half of the training phase, the classification accuracy of Residual Net10 and Dense Net fluctuated greatly. As a lightweight network, MobileNet had the smoothest accuracy curve and was relatively stable. From the perspective of training, the classification algorithm using MobileNet as the feature extraction network had the best performance in the six types of steel surface defect classification tasks.

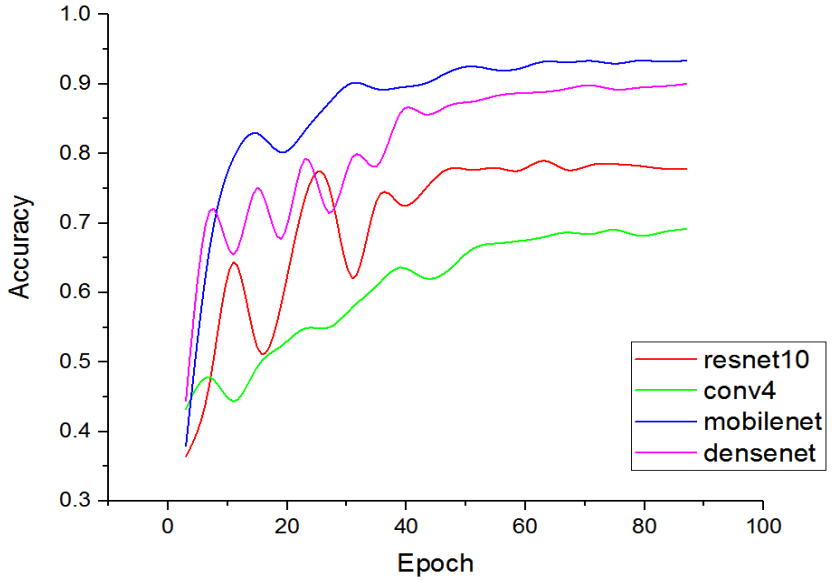

Figure 5. Average accuracy of different classification networks during training.

Tables 2 and 3 quantitatively show the results of multiple feature extraction networks and feature transformations. In Tables 2 and 3, Net indicates feature extraction networks, None indicates no transformation, MS indicates Mean Subtraction and L2 indicates L2-normalization. Moreover, the value before parentheses is the average accuracy and the value inside parentheses represents the confidence interval radius under the confidence level of 0.95. Take Table 2 for example: when MobileNet is used as the feature extraction network, and L2-normalization is adopted as the nearest neighbor classification method, the average classification accuracy of the steel surface defects is 87.30% and the confidence interval radius is 0.52%.

Table 2. 1-shot setting, the average accuracy rate of different classification networks, in %.

Net	None	MS	L2
Convolution4	43.67 (0.45)	49.54 (0.47)	49.27 (0.46)
Residual Net10	73.41 (0.40)	76.14 (0.37)	78.75 (0.33)
Residual Net18	56.34 (1.36)	59.34 (1.39)	59.33 (1.38)
Residual Net34	63.42 (1.48)	65.19 (1.32)	65.81 (1.33)
Residual Net50	41.32 (1.53)	42.68 (1.42)	45.14 (1.60)
Mobile Net	82.60 (0.57)	85.78 (0.53)	87.30 (0.52)
Dense Net121	82.28 (0.58)	82.58 (0.49)	85.16 (0.56)

Table 3. 5-shot Setting, the average accuracy rate of different classification networks, in %.

Net	None	MS	L2
Convolution4	62.32 (0.31)	66.19 (0.30)	66.24 (0.30)
Residual Net10	84.62 (0.24)	85.77 (0.22)	87.67 (0.20)
Residual Net18	70.39 (0.95)	74.54 (0.90)	74.94 (0.90)
Residual Net34	71.71 (0.89)	75.82 (0.86)	76.02 (0.91)
Residual Net50	49.60 (1.05)	54.93 (0.98)	54.03 (1.06)
Mobile Net	88.27 (0.32)	90.94 (0.24)	91.80 (0.23)
Dense Net121	89.47 (0.31)	88.97 (0.29)	92.33 (0.24)

It can be seen from Tables 2 and 3 that both mean subtraction and L2-normalization of the feature vectors in the feature space can effectively improve the defect classification accuracy, but L2-normalization was better than mean subtraction. The reason is that

the eigenvectors normalized by L2-normalization are more discrete under the Euclidean metric. In the 1-shot setting, these two feature transformations were the most effective for convolution 4, which can improve the classification accuracy by 5.87% and 5.6%, respectively. In the Residual Net series network, Residual Net10 had the best performance. The MobileNet + L2 normalization method had the highest accuracy, reaching 87.30%. In the 5-shot setting, mean subtraction was the most effective for Residual Net50, which was increased by 5.33%. L2 normalization was the most effective for Residual Net10, which was increased by 4.55%. In the Residual Net series network, Residual Net10 had the best performance. The Dense Net + L2 normalization method had the highest accuracy, reaching 92.33%. The performance of the Residual Net series network shows that in the case of small sample, the deepening of the network depth may not improve the accuracy. Comparing Residual Net50 and DenseNet121, it was found that the classification performance of DenseNet was better although it was deeper. This shows that in the case of a small sample, dense block can better solve the degradation problem caused by the deepening of the network than the residual block.

4. Conclusions

Aiming at the classification of steel surface defects, this paper proposes a classification method based on small sample learning. In this paper, the feature extraction + feature transformation + nearest neighbor model was adopted. The classification results of the three feature extraction networks, namely Convolution 4, Residual Net, MobileNet and DenseNet, were compared. The effectiveness of the two feature transformations methods, including mean subtraction and L2 normalization, was verified. The experimental results show that in the case of only five samples in each category, the accuracy of DenseNet + L2 normalization + nearest neighbor classification can reach 92.33%. It was found that the classification accuracy was greatly dependent on the feature extraction network, feature transformation and network depth in solving small sample classification problems. The future research focus should be how to solve the degradation problem caused by deepening of the feature extraction network in case of few samples.

Author Contributions: S.W.: conceptualization and methodology. S.Z.: Writing—original draft, software and validation. Q.Z.: visualization. L.C.: formal analysis. C.W.: Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the financial supports by the National Natural Science Foundation of China (No. 52005338).

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm, accessed on 25 November 2021. The other data are available on request from the corresponding author.

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

References

1. Min, S.K.; Taesu, P.; PooGyeon, P. Classification of steel surface defect using convolutional neural network with few images. In Proceedings of the 12th Asian Control Conference (ASCC), Kitakyusyu, Japan, 9–12 June 2019; pp. 1398–1401.
2. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv* **2018**, arXiv:1703.03400.
3. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017; pp. 1–11.
4. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process Mag. J.* **2018**, *35*, 53–65. [CrossRef]
5. Wang, X.; Huang, T.E.; Darrell, T.; Gonzalez, J.E.; Yu, F. Frustratingly simple few-shot object detection. *arXiv* **2020**, arXiv:2003.06957.
6. Sun, B.; Li, B.; Cai, S.C.; Zhang, C. FSCE: Few-shot object detection via contrastive proposal encoding. *arXiv* **2021**, arXiv:2103.05950.
7. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *arXiv* **2017**, arXiv:1703.05175.
8. Sung, F.; Yang, Y.X.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation ntwork for few-shot learning. *arXiv* **2017**, arXiv:1711.06025.

9. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the Internation Conference on Machine Learning (ICML) Deep Learning Workshop, Lille Grande Palais, Lille, France, 6–11 July 2015; pp. 1–30.
10. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R. Meta-learning with latent embedding optimization. *arXiv* **2019**, arXiv:1807.05960.
11. Maryam, I.; Hassan, G. Band clustering-based feature extraction for classification of hyperspectral images using limited training samples. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1325–1329.
12. Maryam, I.; Hassan, G. Automatic defect recognition in x-ray testing using computer vision. In Proceedings of the 17th IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 1026–1035.
13. Maryam, I.; Hassan, G. Feature extraction using attraction points for classification of hyperspectral images in a small sample size situation. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1986–1990.
14. Zhang, Z.F.; Song, Y.; Cui, H.C.; Wu, J.; Schwartz, F.; Qi, H.R. Topological analysis and Gaussian decision tree: Effective representation and classification of biosignals of small sample size. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 2288–2299. [CrossRef] [PubMed]
15. Fu, G.Z.; Sun, P.; Zhu, W.B.; Yang, J.X.; Yang, M.L.; Cao, Y.P. A deep-learning-based approach for fast and robust steel surface defects classification. *Opt. Lasers Eng.* **2019**, *121*, 397–405. [CrossRef]
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
17. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556.
18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 1–9.
19. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 770–778.
20. Telgarsky, M. Benefits of depth in neural networks. In Proceedings of the Conference on Learning Theory, PMLR, New York, NY, USA, 23–26 June 2016; pp. 1517–1539.
21. Howard, A.G.; Zhu, M.; Chen, B. Mobilenets: Effificient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
22. Huang, G.; Liu, Z.; Van der Maaten, L.; Weinberger, K. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 4700–4708.
23. Wang, Y.; Chao, W.L.; Weinberger, K.Q.; Van der Maaten, L. Simpleshot: Revisiting nearest-neighbor classifification for few-shot learning. *arXiv* **2019**, arXiv:1911.04623.