Journal of Engineering Science and Technology
Vol. 18, No. 1 (2023) 554 - 562

PERFORMANCE OF DIFFERENT CNN-BASED MODELS ON CLASSIFICATION OF STEEL SHEET SURFACE DEFECTS

VAN THAN TRAN[1], BA-PHU NGUYEN[2],
NHAT-PHI DOAN[2], THANH DANH TRAN[1,*]

[1]Faculty of Civil Engineering, Ho Chi Minh City Open University,
Ho Chi Minh City, Vietnam
[2] Department of Civil Engineering, Industrial University of Ho Chi Minh City,
Ho Chi Minh City, Vietnam
*Corresponding Author: danh.tt@ou.edu.vn

Abstract

Today, hot rolled steel strip has been widely used in the automobile industry and construction. However, the manufacturing process of this steel often causes various defects on the surface, thereby causing great damage to enterprises. The detection of steel surface defects becomes an indispensable part of iron and steel production facilities. Recently, deep learning has been broadly applied in the domain of image recognition. However, training a deep learning model from scratch creates many problems because the state-of-the-art CNN requires considerable training size and computational resources. For small data scale, transfer learning is an option. This paper focuses on cross-comparing the performance of different CNN-based models on the NEU database of 1800 images which consists of six different typical surface defects. In addition, the performance of optimization algorithms is also cross-compared by incorporating these algorithms in the model with the best performance or highest accuracy. The results show that the DenseNet121 network using the Adam optimizer performs the most effectively with an accuracy of 99.26% for testing set.

Keywords: Convolutional neural network (CNN), Deep learning, Steel sheet, Surface defect classification, Transfer learning.

1. Introduction

Nowadays, hot rolled steel strip has been widely used in the automobile industry and construction. The quality of steel not only affects the product cost, but also significantly affects the accuracy of the machining process later. However, the manufacturing process of this product often causes various defects on the surface [1, 2], thereby causing great damage to enterprises. Thus, detection of steel surface defects becomes an indispensable part of iron and steel production facilities.

Although the surface defect testing for hot rolled steel strip was carried out by traditional methods such as artificial visual inspection, magnetic flux leakage testing [3], Gabor filter [4], support vector machine (SVM) [5], etc., these methods cannot detect surface defects of steel trip in real time due to the influence of human factors and the low rate of defect recognition [6].

In order to simultaneously increase the accuracy of defect recognition and reduce calculation costs and implementation time, deep learning has recently been broadly applied in the image recognition area, in which the Convolutional neural network (CNN) is preferred to apply.

The CNN was early presented by LeCun et al. [7] and has then become one of the most dominant architectures and algorithms for visual object recognition [8, 9]. Thanks to the CNN-based deep learning, the efficiency of image classification has been significantly improved [10-13].

Deep learning is a data-driven method, the output performance of the method depends on the quality and variety of the input dataset, which is a drawback of the deep learning method. In addition, training a deep learning model from scratch creates many problems because state-of-the-art CNN requires considerable training size and computational resources. For small data scales, transfer learning is an option [14]. The application of transfer learning with feature extraction parameters fixed can effectively reduce the training difficulty and improve the accuracy, particularly in the industrial field with small training set. Specially, transfer learning can overcome the phenomenon of overfitting, which is a challenging case for small data.

Currently, there are many classification models of deep learning model built on ImageNet dataset for image feature extraction and connected to a convolutional neural network, such as Xception, NASNetMobile, VGG16, DenseNet121, MobileNet, InceptionV3. These models with pre-trained weights are being viewed as promising models because of their high predictive accuracy in image classification and recognition [14].

On the other hand, the optimization algorithms in these CNN models such as RMSprop, SGD, Adam, AdaGrad, Adamax, Ftrl, Nadam have been incorporated in the model for learned network parameters updating and optimization. Although there are many advantages, the application of transfer learning by the above CNN architectures in the field of surface defect testing for hot rolled steel strip has not been paid too much attention.

This paper focuses on cross-comparing the performance of different CNN-based models on a dataset of 1800 images of steel sheet surface defects. In addition, the performance of optimization algorithms is also cross-compared by incorporating these algorithms in the model with the best performance or highest accuracy.

2. Methods

The dataset used in this study is obtained from the Northeastern University (NEU) hot-rolled steel strip surface defect database which contains 1,800 images of six different types of typical surface defects (patches, rolled-in scale, pitted surface, crazing, scratches and inclusion) [15]. The representative images of six classes are shown in Fig. 1. Each column in the figure shows one example image from each of 300 samples of a class.

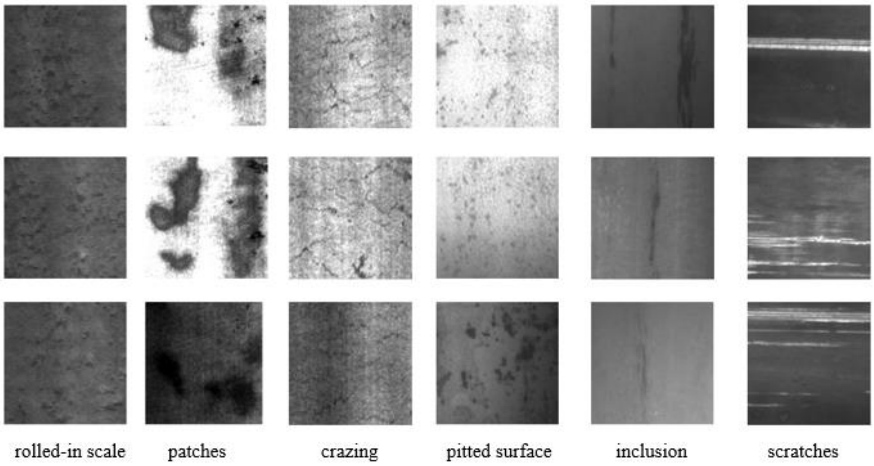

Fig. 1. Image samples (NEU surface defect database).

In this study, to generate different datasets from the above NEU surface defect database, 70% of randomly selected images from the database were used for training, 15% for validation during training, and the remaining 15% was used for testing. Table 1 shows a selection of images used for training, validation, and testing.

Table 1. The quantity of images in different datasets.

Image class	Rolled-in scale	Patches	Crazing	Pitted surface	Inclusion	Scratches
Training	210	210	210	210	210	210
Validation	45	45	45	45	45	45
Testing	45	45	45	45	45	45
Total	**300**	**300**	**300**	**300**	**300**	**300**

In order to collect feature maps of images for classification, six pre-trained networks, including: Xception, NASNetMobile, VGG16, DenseNet121, MobileNet and InceptionV3, were used as image feature extractors. All these networks were pre-trained on ImageNet data using Python code, and the deep learning models were formulated based on the TensorFlow library. Here, the Adam optimization algorithm is used which replaces the classical stochastic gradient descent procedure to interactively update the network weights based on the training data. Data augmentation techniques artificially create different versions of the real data set in order to increase its size. Computer vision models use data augmentation strategy to deal with data scarcity and insufficient data diversity. The object classification models were trained by a desktop computer with access to a Google Colab machine, which allowed calculations to be performed on a Tesla K80 GPU with 12 GB of memory. Each of the pre-trained networks was trained for 30 epochs using the training datasets listed in Table 1. It should be noted that only the classifier was trained, and the weights of the feature extractors were kept unchanged. Table 2 shows the number of trainable and non-trainable parameters of the six pre-trained networks used in this study.

Table 2. Number of trainable and non-trainable parameters of pre-trained networks.

Pre-trained networks	Total parameters	Trainable parameters	Non-trainable parameters
Xception	22,090,286	1,228,806	20,861,480
NASNetMobile	4,580,186	310,470	4,269,716
VGG16	14,865,222	150,534	14,714,688
DenseNet121	7,338,566	301,062	7,037,504
MobileNet	3,529,926	301,062	3,228,864
InceptionV3	22,589,222	786,438	21,802,784

After cross-comparing the performance results of the six models above, the model with the highest accuracy will be optimized with six other optimization algorithms, including: RMSprop, SGD, AdaGrad, Adamax, Ftrl, Nadam to compare the performance of these optimizers with that of Adam optimizer.

3. Results

3.1. Performance of the six CNN models

Figures 2 and 3 present the loss and accuracy results of the six models on the training dataset and validation dataset during training, respectively. It is evident that most of the networks are found to converge within 10 epochs with fairly high accuracy. This proves the effectiveness of the applied CNN models.

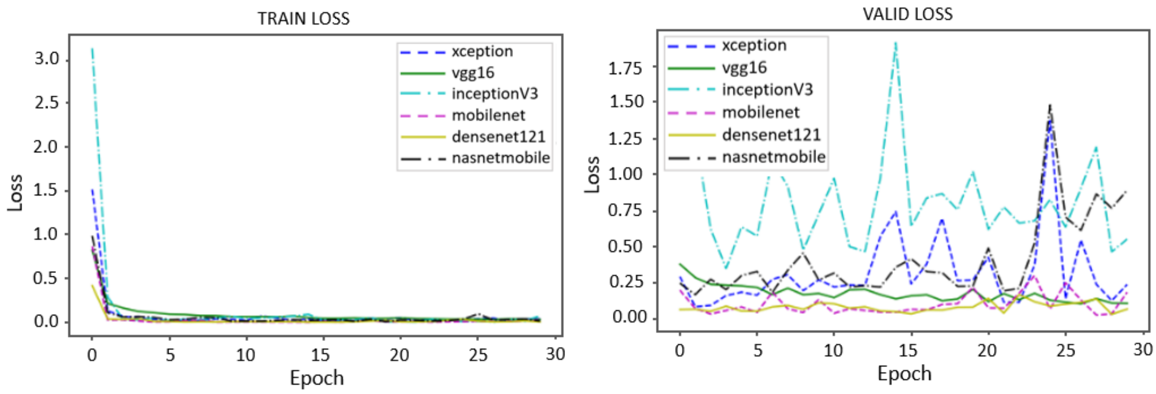

Fig. 2. Loss of the six models.

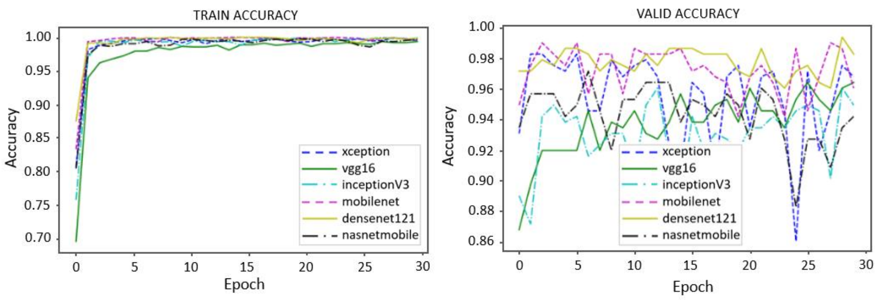

Fig. 3. Accuracy of the six models.

The performance of the six models is last evaluated on the testing data sets. The results of the accuracy of these models are shown in Table 3. It is shown that all six evaluated models exhibit accuracy values above 96%, out of which the DenseNet121 network performs the highest one with an accuracy of up to 99.26%.

Table 3. Accuracy of six models on testing dataset.

Networks	Accuracy (%)
Xception	96.67
NASNetMobile	97.41
VGG16	98.15
DenseNet121	**99.26**
MobileNet	98.89
InceptionV3	97.78

3.2. Performance of the optimizers

After obtaining the accuracy results of the DenseNet121 optimized with Adam optimizer, this network is optimized with six other optimization algorithms, including: RMSprop, SGD, AdaGrad, Adamax, Ftrl, Nadam. The results of loss and accuracy on the training dataset and validation dataset are shown in Figs. 4 and 5, respectively. It can be seen that the DenseNet121 using different optimizers converges very quickly, stabilizing after about 10 epochs, with not much difference in accuracy.

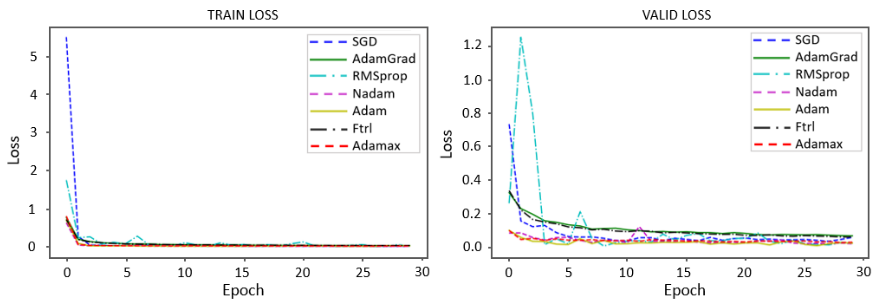

Fig. 4. Loss of the DenseNet121 optimized with different optimizers.

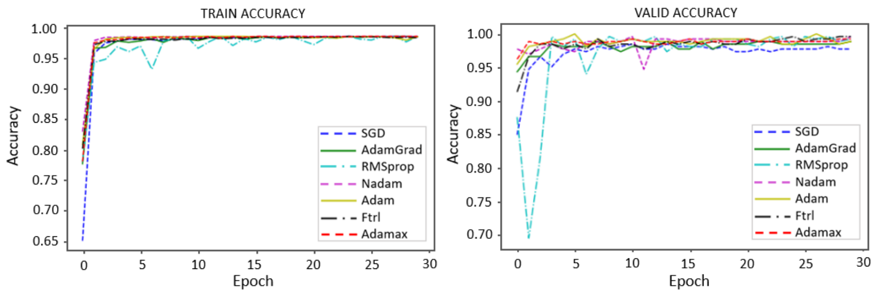

Fig. 5. Accuracy of the DenseNet121 optimized with different optimizers.

The accuracy of the DenseNet121 optimized with the seven optimization algorithms on testing dataset is shown in Table 4. It can be seen that the DenseNet121 network using the Adam optimizer performs the highest accuracy of 99.26%, which outperformed the results of previous studies using the same dataset [15-21].

Table 4. Accuracy of DenseNet121 optimized with seven optimizers on testing dataset.

Optimizers	Accuracy (%)
Adam	**99.26**
RMSprop	98.15
SGD	97.78
AdaGrad	98.52
Adamax	98.89
Ftrl	98.52
Nadam	98.89

3.3. Performance of DenseNet121 model

Figure 6 presents the change of loss and accuracy values versus epoch of the DenseNet121 optimized with Adam optimizer on the training dataset and validation dataset. It is shown obviously that the value of losses during the training is rapidly decreasing while the network is found to converge within 10 epochs.

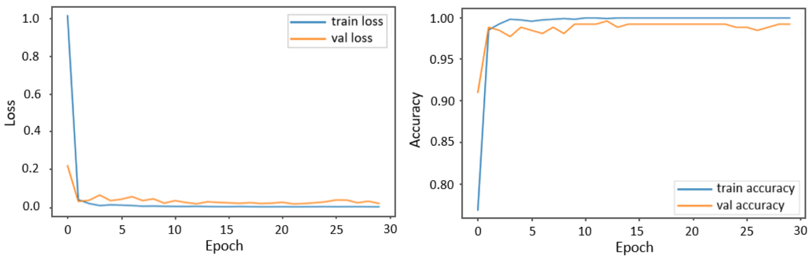

Fig. 6. Loss and Accuracy of the DenseNet121 optimized with Adam optimizer.

The confusion matrix and normalized confusion matrix of the DenseNet121 optimized with Adam optimizer on the testing dataset are shown in Fig. 7, where the colour saturation contours represent the accuracy of each class test. The confusion matrix basically shows how many data points actually belong to a class and are expected to fall into one. The results show that 2 out of 270 images were misclassified. Specifically, there are 2 images that belong in the category of "inclusion" but are predicted to belong to the category of "scratches", all the remaining images are correctly classified by the classification model.

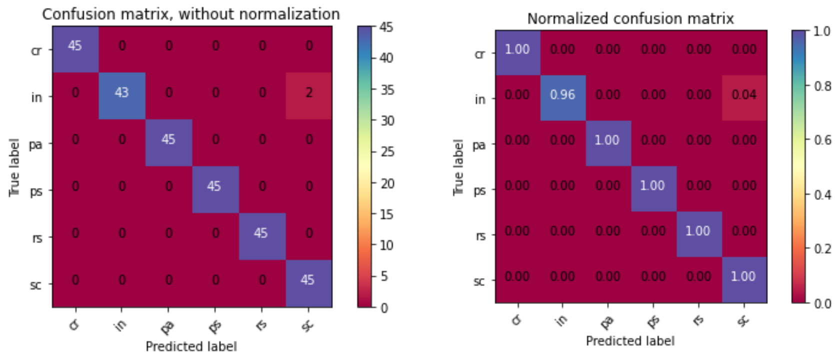

Fig. 7. Confusion and Normalized confusion matrix of the DenseNet121 optimized with Adam optimizer.

In addition, the receiver operating characteristic (ROC) is shown in Fig. 8. On the ROC curve, there exists a point close to the point with coordinates (0, 1), which proves that the classification model is very good. In other words, the more the curve goes along the left boundary and then along the upper boundary of the ROC space, the more accurate the test result is.

The area under the curve (AUC), which is limited to ROC space, is a measure of the accuracy of the test. This area is meant to be a measure of the ability to distinguish good or bad. The larger this value, the better the model. As shown in Fig. 8, the AUC values here are very large (1.00 and 0.98). This proves that the classification model is very good.

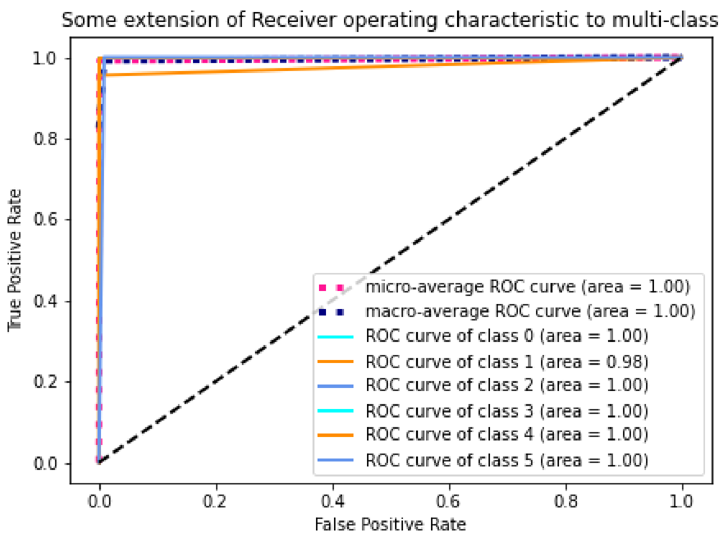

Fig. 8. ROC curve.

4. Conclusions

In this paper, the feasibility of applying transfer learning to identify surface defects on hot rolled steel strip is presented. Six pre-trained networks, including: Xception, NASNetMobile, VGG16, DenseNet121, MobileNet and InceptionV3 were cross-compared for accuracy.

The results show that all six pre-trained networks used in this study can be applied in the recognition of surface defects, even though these networks were trained on completely different datasets. In addition, the results also demonstrate that pre-trained networks have the potential to be implemented in a CNN architecture when the number of training samples is small. Out of the six selected CNN networks, the DenseNet121 model performed the most effective accuracy of 99.26% for the testing dataset. Moreover, seven different optimization algorithms were coupled on the DenseNet121 network, where the Adam optimizer performed the highest accuracy compared to the other optimizers.

Abbreviations

AUC	Area Under the Curve
CNN	Convolutional Neural Network
NEU	Northeastern University (China)
ROC	Receiver Operating Characteristic

References

1. Li, F.; Spagnul, S.; Odedo, V.; and Soleimani, M. (2019). Monitoring surface defects deformations and displacements in hot steel using magnetic induction tomography. *Sensors*, 19(13), 3005.
2. Zhou, S.; Chen, Y.; Zhang, D.; Xie, J.; and Zhou, Y. (2017). Classification of surface defects on steel sheet using convolutional neural networks. *Materials and Technology*, 51(1), 123-131.
3. Maki, H.; Tsunozaki, Y.; and Matsufuji, Y. (1993). Magnetic online defect inspection system for strip steel. *Iron and Steel Engineer (USA)*, 70(1), 56-59.
4. Tikhe, C.; and Chitode, J.S. (2014). Metal surface inspection for defect detection and classification using Gabor filter. *International Journal of Innovative Research in Science, Engineering and Technology*, 3(6), 13702-9.
5. Hu, H.; Li, Y.; Liu, M.; and Liang, W. (2014). Classification of defects in steel strip surface based on multiclass support vector machine. *Multimedia Tools and Applications*, 69(1), 199-216.
6. Li, X.; Tso, S.K.; Guan, X.P.; and Huang, Q. (2006). Improving automatic detection of defects in castings by applying wavelet technique. *IEEE Transactions on Industrial Electronics*, 53(6), 1927-1934.
7. LeCun, Y.; Kavukcuoglu, K.; and Farabet, C. (2010). Convolutional networks and applications in vision. *Proceedings of the ISCAS 2010, IEEE International Symposium on Circuits and Systems, Nano-Bio Circuit Fabrics and Systems*, Paris, France, 253-256.
8. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; and Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.

9. Dinh, V.Q.; Munir, F.; Azam, S.; Yow, K.-C.; and Jeon, M. (2020). Transfer learning for vehicle detection using two cameras with different focal lengths. *Information Sciences*, 514, 71-87.
10. Amin, D.; and Akhter, S. (2020). Deep learning-based defect detection system in steel sheet surfaces. *Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 444-448.
11. Ho, T.T.; Kim, G.T.; Kim, T.; Choi, S.; and Park, E.K. (2022). Classification of rotator cuff tears in ultrasound images using deep learning models. *Medical & Biological Engineering & Computing*, 60, 1269-1278.
12. Ho, T.T.; Kim, T.; Kim, W.J.; Lee, C.H.; Chae, K.J.; Bak, S.H.; Kwon, S.O; Jin, G.Y; Park, E.K.; and Choi, S. (2021). A 3D-CNN model with CT-based parametric response mapping for classifying COPD subjects. *Scientific Reports*, 11(1), 1-12.
13. Nguyen, P.D.; Nguyen, T.Q.; Tao, Q.B.; Vogel, F.; and Nguyen-Xuan, H. (2022). A data-driven machine learning approach for the 3D printing process optimisation. *Virtual and Physical Prototyping*, 17(4), 768-786.
14. Park, S.S.; Tran, V.T.; Doan, N.P.; and Hwang, K.B. (2021). Evaluation of damage level for ground settlement using the convolutional neural network. *Proceedings of the CIGOS 2021, Emerging Technologies and Applications for Green Infrastructure*, Hanoi, Vietnam, 1261-1268.
15. Song, K.; and Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285, 858-864.
16. Song, K.; Hu, S.; and Yan, Y. (2014). Automatic recognition of surface defects on hot-rolled steel strip using scattering convolution network. *Journal of Computer Information Systems*, 10(7), 3049-3055.
17. Yi, L.; Li, G.; and Jiang, M. (2017). An end-to-end steel strip surface defects recognition system based on convolutional neural networks. *Steel Research International*, 88(2), 1600068.
18. Kholief, E.A.; Darwish, S.H.; and Fors, M.N. (2017). Detection of steel surface defect based on machine learning using deep auto-encoder network. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Rabat, Morocco, 218-229.
19. Luo, Q.; Sun, Y.; Li, P.; Simpson, O.; Tian, L.; and He, Y. (2018). Generalized completed local binary patterns for time-efficient steel surface defect classification. *IEEE Transactions on Instrumentation and Measurement*, 68(3), 667-679.
20. Jain, S.; Seth, G.; Paruthi, A.; Soni, U.; and Kumar, G. (2020). Synthetic data augmentation for surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*, 33(2), 1007-1020.
21. Feng, X.; Gao, X.; and Luo, L. (2021). X-SDD: a new benchmark for hot rolled steel strip surface defects detection. *Symmetry*, 13(4), 706.