Classification of Magnetic Tile Surface Defects Based on Efficientnet Network with Attention

Chengshuai Fan\textsuperscript{1*}

\textsuperscript{1}School of Information Engineering, Zhengzhou University, Zhengzhou, Henan, 452300, China
*Corresponding author’s e-mail: fcs56123@gs.zzu.edu.cn

Abstract. The magnetic tile image has the characteristics of uneven illumination, complex surface texture, and low contrast. Aiming at the problem that the traditional defect detection algorithm is difficult to accurately identify the defects, and the deep learning algorithm is difficult to balance the classification accuracy and the size of the speed model, a defect classification algorithm based on attention-based EfficientNet is proposed. The algorithm first enhances the network's spatial and location information for image features by integrating the Convolutional Block Attention Module, and improves the network's ability to identify defects. Then, on this basis, Criss-Cross Attention is added to the network, so that the network can better the context information of the horizontal and vertical cross of image features, so that each pixel can finally capture the full image dependency of all pixels. Experimental results show that the algorithm has higher classification accuracy than EfficientNet-B0, reached 99.11%, and has a better balance between accuracy, speed and model size than other classification models.

1. Introduction
Magnetic tile is a tile-shaped magnet mainly used in permanent magnet motors. Common defects are shown in Figure 1[1]. Its quality will directly affect the performance and service life of permanent magnetic motors. In the production process of magnet tiles, affected by factors such as raw material composition, processing technology, equipment conditions, some defects, such as collapse, and missing wear, will inevitably appear on the surface of the magnet tiles. These defects have a great impact on the magnetic flux, residual magnetic induction, anti-overload demagnetization and anti-aging performance of the magnetic tile. Therefore, it is necessary to evaluate the quality of the magnetic tiles after they are produced.
Figure 1. Examples of magnetic tile surface defects, labeled with pixel-level ground truths (GTs).

At present, most of the magnetic tile production factories still use manual methods to detect the surface defects of the magnetic tiles. The efficiency of manual visual inspection is low, and the accuracy of detection and classification cannot be guaranteed, so it is important to improve the efficiency of defect recognition on the production line. In recent years, automatic recognition technology has been applied to the identification of magnetic tile defects[2]-[4], but there are also many problems. Due to the many types of magnetic tile defects, the complex surface texture and low contrast, it is difficult for traditional visual inspection and image processing technologies to detect the surface of the magnetic tile.

With the development of deep learning, convolutional neural networks (CNN) have achieved the best performance in visual tasks such as image classification, object detection, and semantic segmentation. In the field of image classification, deep neural networks such as VGGNet[5], ResNet[6] and Densenet[7] have been proposed. These methods have been used as backbone networks in various fields and have achieved good results.

The current classification algorithm has achieved good results in the classification of magnetic tile surface defects, but the accuracy, speed, and model size cannot achieve a good balance. The network speed with high accuracy is relatively slow, the opposite is the same. Therefore, this article proposes an EfficientNet[8] network with attention to this problem, which achieves a balance of accuracy and speed, and the model is small. In the end, our algorithm has an accuracy of 99.11% for magnetic tile surface defects, 117 FPS(frames per second) , and 31.9MB model-size . It is suitable for deployment to the production line site.

The main contributions of this article:
1. Apply the EfficientNet network to the classification task of magnetic tile surface defects. The detection speed and model size are faster than other models and have a smaller memory footprint.
2. In order to improve accuracy, integrate the attention mechanism into the EfficientNet network.(1) Integrate the Convolutional Block Attention Module into EfficientNet, so that the network can better learn the space and channel information, and improve the detection accuracy. (2) Integrate Criss-Cross Attention into the EfficientNet network to help the network capture the horizontal and vertical context information and improve the detection accuracy.

2. Formatting the title, authors and affiliations

2.1. Algorithm overview
Based on EfficientNet-B0, this paper adds Convolutional Block Attention Module (CBAM)[9] and Criss-Cross Attention[10](CCA) modules to fully improve the network's feature recognition ability for magnetic tile surface defects. The overall network structure is shown in Figure 2.
2.2. Convolutional Block Attention Module

Convolutional Block Attention Module (CBAM) represents the attention mechanism module of the convolution module. It is an attention mechanism module that combines spatial and channel.

For a feature map \( F \in \mathbb{R}^{C \times H \times W} \) imported from the upper layer of CBAM, where \( C \) is the channel of the feature map, \( H \) is the height of the feature map, and \( W \) is the width of the feature map, first obtain the one-dimensional channel attention through the channel attention module of CBAM Feature map \( M_{\mathcal{C}} \in \mathbb{R}^{C \times 1 \times 1} \), and then obtain a two-dimensional spatial attention feature map \( M_{\mathcal{S}} \in \mathbb{R}^{1 \times H \times W} \) through the spatial attention module. The overall process is as formula (1), (2):

\[
F' = M_{\mathcal{C}}(F) \otimes F \\
F'' = M_{\mathcal{S}}(F') \otimes F'
\]

Among them, \( \otimes \) is multiplication element by element. First, multiply the channel attention feature map \( M_{\mathcal{C}}(F) \) and the input feature map \( F \) to get \( F' \), then calculate the spatial attention feature map \( M_{\mathcal{S}}(F') \), and finally multiply the two to get the final output \( F'' \).

The calculation process of \( M_{\mathcal{C}}(F) \) and \( M_{\mathcal{S}}(F') \) are as formula (3), (4):

\[
M_{\mathcal{C}}(F) = \sigma(W_1 \left(W_0(F_{avg})\right) + W_1(W_0(F_{max})) \in \mathbb{R}^{C \times 1 \times 1} \\
M_{\mathcal{S}}(F') = \sigma(f_7 \times 7 (F_{avg}^c, F_{max}^c)) \in \mathbb{R}^{H \times W}
\]

where, \( \sigma \) is the Sigmoid activation function, \( W_1 \) is the second layer of the shared fully connected layer, and the output vector length is \( C \). \( F_{avg}^c \) and \( F_{max}^c \) are the two different channel background descriptions obtained by MaxPool and AvgPool, same for \( F_{max}^s \) and \( F_{avg}^s \).

2.3. Criss-Cross Attention

The Criss-Cross Attention (CCA) module collects contextual information in horizontal and vertical directions to enhance the function of pixel representation. The structure is shown in Figure 3.

![Figure 3. Flow chart of CCA.](image)

In the above Figure 3, the space size of \( H \) is denoted as \( C \times H \times W \), and the CCA module first uses two \( 1 \times 1 \) convolutions to generate \( Q \) and \( K \). The space size is \( C' \times H \times W \), which plays a role in dimensionality reduction. We use \( Q \) and \( K \) to generate \( A \) through the Affinity operation, and its spatial size is \((H + W - 1) \times W \times H\). For any position \( \mu \) on the feature map, the features of \( C' \) channels (at
this pixel position) will be extracted to form $Q_{\mu}$, the size is $1 \times C'$, and then a similar operation is performed to extract the feature vector at the position of the cross and mark it as $\Omega_{\mu}$. The Affinity operation is defined as follows.

$$d_{i,u} = Q_{i,u} \Omega_{i,u}^T$$  \hspace{1cm} (5)$$

where $d_{i,u} \in D$ is the degree of correlation between features $Q_{i,u}$ and $\Omega_{i,u}$, $i = [1, \ldots, H + W - 1]$, and $D \in \mathbb{R}^{(H+W-1) \times (W \times H)}$. Then, a softmax layer on $D$ is applied to calculate the attention map $A$.

Another $1 \times 1$ convolution layer is applied on $H$ to generate $V \in \mathbb{R}^{C \times W \times H}$ for feature adaptation. We can obtain a vector $V_{i,u} \in \mathbb{R}^C$ and a set $\Phi_{i,u} \in \mathbb{R}^{(H+W-1) \times C}$. The set $\Phi_{i,u}$ is a collection of feature vectors in $V$ which are in the same row or column with position $u$. The contextual information is collected by an Aggregation operation defined as follows.

$$H_{i,u}' = \sum_{i=0}^{H+W-1} A_{i,u} \Phi_{i,u} + H_{i,u}$$  \hspace{1cm} (6)$$

where $H_{i,u}'$ is a feature vector in $H' \in \mathbb{R}^{C \times W \times H}$ at position $u$ and $A_{i,u}$ is a scalar value at channel $i$ and position $u$ in $A$. The contextual information is added to local feature $H$ to augment the pixel-wise representation. Therefore, it has a wide contextual view and selectively aggregates contexts according to the spatial attention map.

### 3. Results & Discussion

#### 3.1. Implementation details

Our experimental results are based on the following experimental environment: Ubuntu18.04, CUDNN7.6 for CUDA10.2, GPU: Tesla K80, the deep learning framework is Pytorch, and the program language was Python. Epoch=1000, LR=0.01. When the training accuracy reaches 99.9%, it will automatically stop training and save.

#### 3.2. Datasets and Evaluation Metrics

The training data set[1] is the magnetic tile defect data is shown in Figure 1. The original data contains the original image and the labeled image. We take the original image as the data set and expand it through operations such as flipping and translation. The final training set has 2000 images and the test set has 500 images. The evaluation indicators are classification accuracy, FPS and Model-Size.

#### 3.3. Ablation experiment

We conduct experiments on the EfficientNet-B0 network as the baseline, and conduct ablation experiments on CBAM and Criss-Cross Attention. The experimental results are shown in Table 1.

| Algorithm          | Accuracy(%) |
|--------------------|-------------|
| EfficientNet-B0[8] | 97.70       |
| EfficientNet-B0-CBAM | 98.67     |
| EfficientNet-B0-CCA | 98.10      |
| EfficientNet-B0-CBAM+CCA | **99.11** |

It can be seen from Table 1 that after adding CBAM and CCA respectively, the accuracy of the model has been improved to different degrees, which proves the effectiveness of the two modules, and proves that CBAM can improve the network's ability to use feature space and channel information. CCA can make the network better capture the full image dependency of all pixels. After integrating the two into the network, the accuracy is increased by 1.41% compared with EfficientNet-B0, which proves the
effectiveness of our algorithm design. Figure 4 is the training diagram of our final algorithm (EfficientNet-B0-CBAM+CCA).

![Training Diagram](image)

Figure 4. Training diagram of our final algorithm.

3.4. Comparative Experiment

For comparison, we conducted experiments on other general classification models (VGG, ResNet, Densenet). The experimental results are shown in Table 2.

| Algorithm               | Accuracy(%) | FPS  | Model-size(MB) |
|-------------------------|-------------|------|----------------|
| EfficientNet-B0-CBAM+CCA| 99.11       | 117  | 31.9           |
| VGG-16[5]               | 97.56       | 71   | 105            |
| ResNet-34[6]            | 97.52       | 150  | 83.2           |
| ResNet-50[6]            | 97.72       | 106  | 97.7           |
| Densenet-121[7]         | 96.5        | 96   | 30.9           |

It can be seen from Table 2 that our algorithm has the highest accuracy compared to other models, and model-size is also very small and suitable for deployment in the production site. FPS has also reached 117 to meet the needs of the production line. Through the analysis of Table 2, we can see that after evaluating the accuracy, model-size, and fps, our algorithm performs well in all three indicators, which proves that our algorithm is better than other classification models in the classification of magnetic tile surface defects.

4. Conclusion

In this paper, we propose a magnetic tile surface defect classification network based on EfficientNet with attention. In order to solve the current problem that the accuracy and speed model size of the magnetic tile surface defect classification network cannot have a good balance, we combine CBAM and CCA into EfficientNet-B0. Experimental results show that our algorithm can achieve 99.11% accuracy, and the detection speed and model size are excellent compared to other models. In the next step, we will design a detection algorithm that can accurately locate the defect location based on the proposed classification algorithm.

References

[1] Huang, Y., Qiu, C., & Yuan, K. (2020) Surface defect saliency of magnetic tile. The Visual Computer, 36(1), 85-96.
[2] Yang, C., Liu, P., Yin, G., & Wang, L. (2017) Crack detection in magnetic tile images using nonsubsampled shearlet transform and envelope gray level gradient. Optics & Laser Technology, 90, 7-17.
[3] Li, X., Liu, Z., Yin, G., & Jiang, H. (2020) Ferrite Magnetic Tile Defects Detection Based on Nonsubsampled Contourlet Transform and Texture Feature Measurement. Russian Journal of Nondestructive Testing, 56(4), 386-395.

[4] Ben Gharsallah Mohamed et al. (2021) Defect identification in magnetic tile images using an improved nonlinear diffusion method. Transactions of the Institute of Measurement and Control, 43(11): 2413-2424.

[5] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas. pp. 770-778.

[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Hawaii. pp. 4700-4708.

[8] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. Los Angeles. pp. 6105-6114.

[9] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). Munich. pp. 3-19.

[10] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019). Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul. pp. 603-612.