GACNet: Defect Detection with Graph Attention Mechanism Convolutional Network Model

Yilong Guo [1, a, *], Yiming Yao [1, b], and Luyang Jie [1, c]

[1] School of Microelectronics, Shanghai University, Shanghai, 201899, China

[a, *] wdz_gyl@shu.edu.cn, [b] yaoym@shu.edu.cn, [c] jly@she.edu.cn

Abstract. Convolutional Neural Networks (CNNs) are classic models for image classification. In modern industrial and manufacturing fields, automated defect detection and classification often rely on CNNs and their variant networks. However, due to the diversity and complexity of defects, traditional image processing methods often struggle to perform this task effectively. Graph Neural Networks (GNN), as effective tools for handling graph data, can capture relationships and local structures among nodes in a graph. This is valuable for describing the distribution and interconnections of defects in images. This paper introduces a deep learning framework that combines the advantages of both CNNs and GNNs, along with a method for transforming 2D images into graph data. In defect classification tasks, this framework outperforms ResNet-50 with pre-trained weights, achieving 4.07% higher precision.

Keywords: GAT; CNN; Defect Classification; Image to Graph.

1. Introduction

Convolutional Neural Networks (CNNs) have achieved significant breakthroughs in the field of image classification [1]. CNNs are adept at capturing local features within 2D grids, recognizing patterns, and distinguishing intricate textures [2], making them a cornerstone of image classification tasks. The limitation of CNNs in feature extraction is confined to Euclidean spaces and grid-based partitions [3]. This restricts their applicability in scenarios where data is inherently non-Euclidean, as they are primarily designed to operate on regularly gridded data such as images. In the context of defect detection and classification tasks, Defect shapes exhibit a high degree of similarity, while the sizes of defects of the same type vary significantly, fuzzy boundaries for defining different defect categories, and cluttered image backgrounds pose formidable challenges. Relying solely on 2D image features proves inadequate in capturing the overall characteristics of defects.

However, a graph-based representation offers a new perspective for solving problems in this field. Graph Neural Networks (GNNs) offer distinct advantages for information extraction in non-Euclidean spaces, as they are capable of processing graph data, which could bridge the gap between different domains [3]. GNNs can effectively capture relationships and dependencies among data points in a more flexible manner, making them suitable for applications like social networks, molecular chemistry, and recommendation systems, where data often exhibits graph-like structures [5]. Moreover, GNN also exhibits strong performance on the MNIST dataset [6], making it entirely plausible to contemplate a novel model architecture that combines the strengths of GNN and CNN.

This paper considers to use of GNNs to capture global feature information from images through graph-based message passing, transcending the limitations of Euclidean space for 2D image feature learning. We propose the Graph Attention Mechanism Convolutional Network (GACNet), which amalgamates graph attention mechanisms with the well-performing ResNet architecture from CNNs. The primary objective is to comprehensively learn image features from multiple perspectives, thereby enhancing defect detection accuracy. Comparative analyses between GACNet and traditional CNNs, specifically the ResNet model, demonstrate a remarkable precision rate of 80.95%, surpassing the pre-trained ResNet50 model by 4.07%.

The primary contributions of this paper are as follows:

We propose a novel model that combines Graph Attention Mechanisms (GAT) [10] with traditional CNNs, leading to an enhancement in classification accuracy for defect categorization in comparison to conventional CNNs.

We use a data transformation approach that employs superpixel segmentation to divide images into nodes and establish a graph based on inter-node correlations. This method effectively converts images from Euclidean space into a non-Euclidean graph structure.

The model simultaneously uses the results of segmentation and the original image for defect category determination.

The first section of this paper reviews the current research status and introduces the model's innovations. The second section provides a detailed description of the model architecture. The third section conducts experimental comparisons, and the fourth section summarizes the paper.

2. Model

2.1 Image to Graph

The process of transforming an image into a graph-based representation involves several stages, and each stage is visualized as shown in Fig 1.

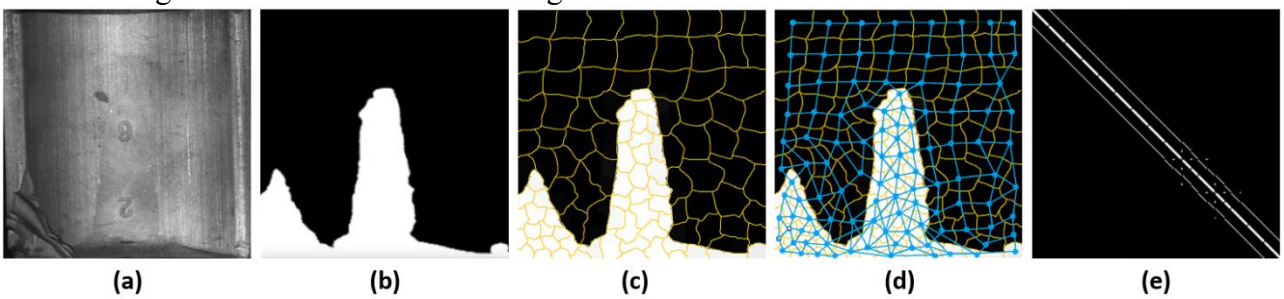

Fig. 1 The process of transformation from an image to a graph

Node Generation: To avoid using single pixels as a node and accurately distinguish relevant areas, we chose superpixel segmentation which has the advantage of reducing input size and representing different domains with a unified or similar graph structure [4]. Superpixels aggregate pixels that share similarities in color and other low-level attributes, such as their spatial coordinates, into perceptually meaningful units [8]. As shown in Fig 1, Fig 1(a) represents the original image, and Fig 1(b) shows the segmented mask. We use SLIC for superpixel segmentation on the mask image, resulting in Fig 1(c), which incorporates spatial components into its superpixel segmentation [9]. After segmentation, we obtain nodes feature vector (N_i) with five values, which consist of the RGB values for each segmented part in the three channels (R_i, G_i, B_i) and two coordinate values (X_i, Y_i).

$$N_i = (R_i, G_i, B_i, X_i, Y_i),\ (N_i \in N). \#(1)$$

Edge Generation: Edges (ε) are produced by utilizing the positional features within the node characteristics. Every neighboring pair of nodes generates a single edge, thus constituting the entire graph. The construction process is described by the following formula.

$$\varepsilon = \sum_{i \neq j} \left\{ \left(N_i, N_j\right) \middle| |X_i - X_i| \leq 1 \cap \left|Y_i - Y_j\right| \leq 1 \right\}. \#(2)$$

Graph Construction: A graph ($G = (N, \varepsilon)$) comprises nodes and edges, as shown in Fig 1(d), with each node possessing node feature representations, while edges are represented by an adjacency matrix, as shown in Fig 1(e). This transformation effectively converts 2D image matrices into graph-structured data.

2.2 Model framework

As illustrated in Fig 2, the overall framework of GACNet consists of three main components: Graph Message Passing (GMP) Module, Convolutional Network Module, and Multilayer Perceptron (MLP) [7]. Taking into consideration both the mask and the original image, we perform multiplication of the mask and the image to extract the detailed defect features while discarding the remaining information in the image. This enables the model to focus its feature learning on the

defect regions. Subsequently, defect features are segmented using a superpixel segmentation technique. After segmentation, each voxel serves as a node, and edges are created between nodes based on their color and spatial characteristics, forming a graph. The graph is then subject to information propagation using the GMP Module. To capture local structural information of defects, a dual-channel approach is employed, simultaneously applying CNN processing to the defects. Following eighteen layers of convolution and three layers of graph attention mechanisms, the obtained features are concatenated and fed into a MLP to obtain the final classification results.

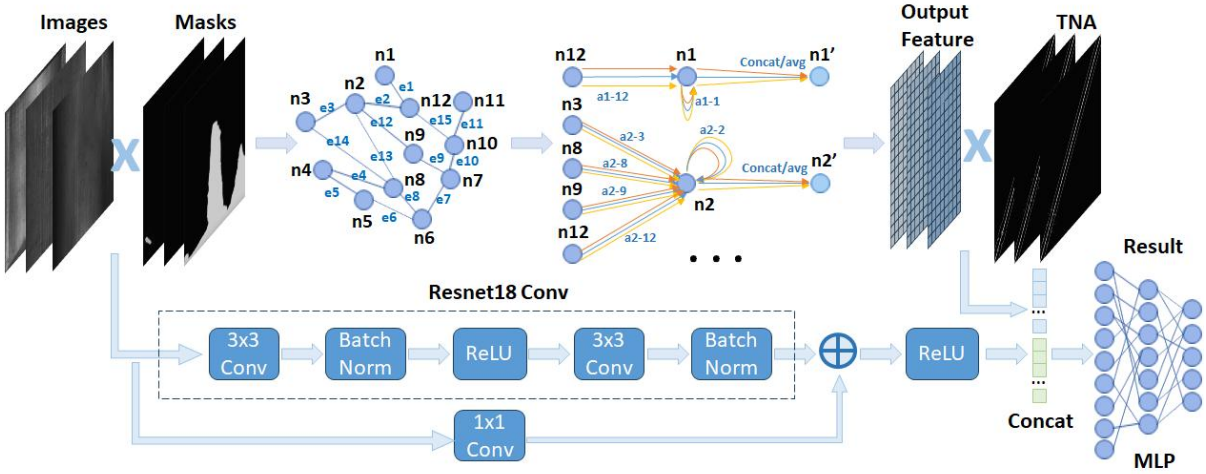

Fig. 2 The framework of the GACNet model

2.2.1 GMP Module

The GMP Module's backbone is GAT, whose ability to learn adaptive weights, as it incorporates learnable attention mechanisms for each edge, enabling nodes to selectively attend to their neighbors based on their individual significance. In contrast, Graph Convolutional Networks (GCN) employ fixed and uniform weights for aggregation [11], rendering GAT a more adaptable and expressive choice. GAT excels in capturing both local and selectively filtering global information, exhibiting strong scalability.

GMP Module applies attention by distributing the input features of surrounding nodes and the central node to their respective edges. For a node N_i, its neighboring nodes are represented as $N_i^j, (j \in N_{(i)})$, $N_{(i)}$ represents the set of all nodes adjacent to N_i. To enhance expressiveness and transform input features into higher-level ones, a learnable shared linear transformation is applied to each node, parameterized by a weight matrix $W \in R^{F^2}$, F is the number of nodes' feature dimension, which is 5 in this paper. Then, we perform self-attention using a shared attention mechanism denoted as $a: R^{2F} \to R$ to calculate attention coefficients $e_{ij} = a(WN_i, WN_j)$, indicating the importance of node j's features to node i.

Our model uses a 3-head attention mechanism (K=3). As shown in Fig 2, the three arrows in orange, blue, and yellow represent the three attention heads. Each head independent attention mechanism must normalize the attention coefficients ($e_{ij} \to a_{ij}$) and perform a linear combination with the corresponding node features, resulting in the final output features for each node. These features, each associated with attention weights, are then concatenated to obtain the following output feature representation:

$$a_{ij} = softmax_j[e_{ij}] = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}, \#(4)$$

$$N_i^{'} = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \epsilon N_i} a_{ij}^k W^k N_j\right), \#(5)$$

where || represents concatenation, a_{ij}^{k} are normalized attention coefficients computed by the k-th attention mechanism (a^{k}), and w^{k} is the corresponding input linear transformation's weight matrix. The final output $N_i^{'}$ will consist of $K{\cdot}F$ features for each node. Finally, as shown in Fig 2, all the updated nodes features and the adjacency matrix from target nodes to edges (TNA) are multiplied to obtain the output of the GMP Module.

2.2.2 Convolutional Network Module

The experimental section of part four shows relying solely on GNN for 2D image classification does not yield satisfactory results. Consequently, the model incorporates ResNet18 as the convolutional network module backbone to extract image textures and local feature information.

2.2.3 MLP Module

The outputs from the GMP Module undergo normalization and are combined with the outputs from the Convolutional Network Module, which are then fed into the MLP Module. This module comprises three layers, with ReLU activation applied between the first and second layers. Sigmoid activation is utilized between the second and third layers to obtain the final classification results.

2.2.4 Loss Function

Considering the issue of imbalanced classification samples, the model employs the Focal Loss as its loss function. The Focal Loss is designed to address the problem of class imbalance by down-weighting well-classified examples, thereby emphasizing the importance of challenging or misclassified instances. This loss function helps the model focus on improving its performance for minority classes, where traditional cross-entropy loss may not be as effective due to the dominance of majority classes.

3. Experiment

3.1 Dataset

For defect detection, we utilized the "Magnetic-Tile-Defect" dataset [12], originally designed for segmentation tasks. However, in defect detection, we not only need to determine the position and shape of defects but also attribute them to specific categories. Currently, both object recognition and instance segmentation follow a two-stage approach, initially identifying the location and subsequently classifying the category. This paper primarily focuses on the second stage, which is classifying defect categories. The dataset is outlined in Table 1, which suffers from class imbalance and has a limited amount of data.

Table 1. Magnetic-Tile-Defect Dataset

Categories	Blowhole	Break	Crack	Fray	Uneven
Numbers	200	85	57	32	103

3.2 Model performance

3.2.1 Model Accuracy Comparison

To assess the effectiveness of GACNet, we compare it with several classical models, including a traditional CNN model (comprising three convolutional layers and two fully connected layers), ResNet18, ResNet50, ResNet50 -p (with pre-trained weight loading), and GAT (one-head attention) model applied to 2D image classification. The results are presented in Table 2.

Due to the imbalance in the training dataset, we have employed four comprehensive evaluation criteria to assess the model's performance on this dataset: accuracy, precision, recall, and F1 score. The dataset is divided into three groups: original images (Image), defect masks (Mask), and Defect Regions in Original Images (Image × Mask). From the Table 2, we have observed the following:

Table 2. Results Compared to Traditional Models

Dataset	Model	Accuracy	Precision	Recall	F1 score
	CNN	73.75%	65.33%	68.52%	65.21%
	GAT	31.52%	23.75%	22.92%	23.33%
Image	ResNet18	35.31%	27.65%	35.73%	28.38%
	ResNet50	85.94%	80.31%	80.40%	78.98%
	ResNet50-p	33.75%	24.82%	26.15%	23.50%
	GACNet	42.81%	37.04%	40.24%	38.57%
	CNN	60.31%	52.26%	56.86%	51.43%
	GAT	64.23%	56.32%	53.43%	54.84%
Mask	ResNet18	84.69%	80.38%	81.27%	79.22%
	ResNet50	80.62%	71.88%	71.70%	70.38%
	ResNet50-p	87.19%	83.80%	81.68%	81.44%
	GACNet	91.94%	90.01%	88.80%	89.40%
	CNN	64.56%	57.43%	61.27%	55.81%
Image	GAT	42.81%	39.93%	37.48%	38.67%
×	ResNet18	22.81%	7.71%	24.31%	11.32%
Mask	ResNet50	81.88%	76.88%	77.29%	75.71%
	ResNet50-p	26.25%	15.00%	27.08%	16.72%
	GACNet	83.04%	80.95%	78.27%	79.59%

(Image): Classifying the original images alone yielded poor results. This is due to variations in lighting and complex backgrounds, making classification challenging.

(Mask): Classifying the defect masks showed the best performance. However, the GACNet model achieved an F1 score of 89.40%, which is 7.04% higher than the ResNet50 model. Despite its effectiveness, relying solely on masks for defect classification leads to a loss of significant information, making it less interpretable. Therefore, it may not be the recommended approach.

(Image × Mask): Classifying only the defect regions within the original images resulted in an accuracy of 83.04%, which is 1.16% higher than the best-performing ResNet50 model. It remains the top-performing model in this dataset, striking a balance between accuracy and interpretability.

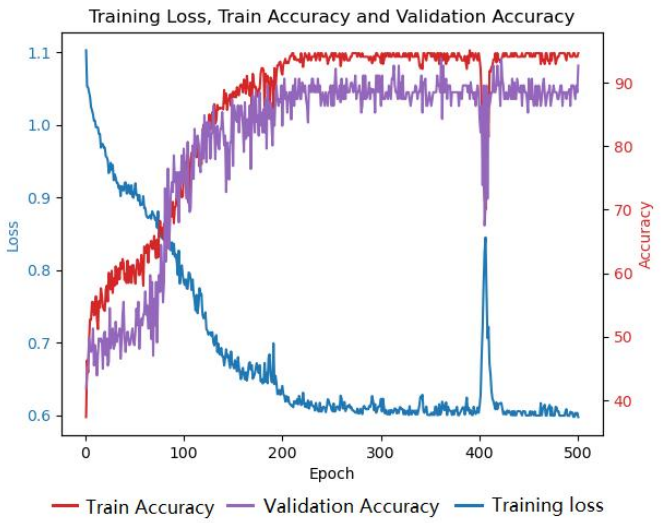

Fig. 3 Training loss, training accuracy and validation accuracy curve

In summary, the GACNet model has demonstrated its efficacy, particularly when focusing on defect regions within the original images, providing a valuable solution for defect classification. Training loss, training accuracy and verification accuracy during the training process, as shown in Fig 3.

3.2.2 Ablation Experiment

We conducted ablation experiments on some components of the model. These experiments involved varying the number of attention heads used in the GMP module (K=2, 3, 4) and testing

different loss functions, including Cross-Entropy Loss and Focal Loss. The comparative results for these experiments are presented in Table 3.

Table 3. The Results of Ablation Experiments for GACNet on The (Image × Mask) Dataset

Loss function	Attention heads	Accuracy	Precision	Recall	F1 score
Cross-	2-heads	68.25%	63.33%	58.52%	60.83%
Entropy	3-heads	71.32%	64.23%	60.22%	62.16%
Loss	4-heads	65.31%	62.65%	63.73%	63.18%
	2-heads	75.94%	72.31%	70.90%	71.60%
Focal Loss	3-heads	83.04%	80.95%	78.27%	79.59%
	4-heads	82.24%	76.73%	78.32%	77.51%

Therefore, the GMP module was configured with 3-heads attention, and the model training utilized Focal Loss as the loss function.

4. Summary

This paper introduces a novel model that demonstrates promising performance in defect detection and classification. Furthermore, it proposes an innovative approach to combine 2D images with graph structures, allowing for a more comprehensive capture of image information. By leveraging convolutional networks to learn texture features and incorporating the global information propagation characteristics of GNN, it enhances the accuracy of defect classification.

References

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[3] Liang F, Qian C, Yu W, et al. Survey of graph neural networks and applications[J]. Wireless Communications and Mobile Computing, 2022, 2022.

[4] Avelar P H C, Tavares A R, da Silveira T L T, et al. Superpixel image classification with graph attention networks[C]//2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). IEEE, 2020: 203-209.

[5] Guo Y, Chen Y, Zou X, et al. Algorithms and architecture support of degree-based quantization for graph neural networks[J]. Journal of Systems Architecture, 2022, 129: 102578.

[6] Monti F, Boscaini D, Masci J, et al. Geometric deep learning on graphs and manifolds using mixture model cnns[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5115-5124.

[7] Tolstikhin I O, Houlsby N, Kolesnikov A, et al. Mlp-mixer: An all-mlp architecture for vision[J]. Advances in neural information processing systems, 2021, 34: 24261-24272.

[8] Stutz D, Hermans A, Leibe B. Superpixels: An evaluation of the state-of-the-art[J]. Computer Vision and Image Understanding, 2018, 166: 1-27.

[9] Achanta R, Shaji A, Smith K, et al. SLIC superpixels compared to state-of-the-art superpixel methods[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(11): 2274-2282.

[10] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[11] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[12] Huang Y, Qiu C, Yuan K. Surface defect saliency of magnetic tile[J]. The Visual Computer, 2020, 36: 85-96.