Image classification method based on superpixels and Graph Neural Network

Yonghong Tang*

School of mathematics, Harbin Institute of Technology, 91944, Harbin, Heilongjiang, China

Abstract. In image classification tasks, commonly used deep learning network architectures include Convolutional Neural Network (CNN) and Vision Transformer (ViT). Still, both are relatively mature, while Graph Neural Network (GNN) network architecture has been attempted less in image classification. In addition, the architecture and design of GNN are still rapidly developing and have enormous potential. This study aims to create an image classification method based on superpixels and graph neural networks. This project attempted three different approaches to preserve superpixel node features with varying numbers of features, designed two training network architectures with different complexities and conducted six sets of tests on image classification tasks on the fashion-mnist dataset. As a result, the use of superpixel and GNN methods achieved good accuracy in image classification tasks, demonstrating the potential of this method in image recognition tasks. It was also found that adding the boundary coordinates of the region to the training in the selection of superpixel node features can improve the accuracy of the final image classification task.

1 Introduction

In image classification tasks, the commonly used deep learning network architectures currently include Convolutional Neural Network (CNN) and Vision Transformer (ViT) [1]. CNN is suitable for use on small datasets with high computational efficiency, especially for tasks with prominent local features. ViT is suitable for use on large datasets and performs better when dealing with complex global features. However, both methods have their shortcomings. CNN relies on the size of the convolutional kernel and requires multiple layers of stacking to obtain global information, making it weak in handling long-distance dependencies. ViT lacks spatial structure prior knowledge and requires a large amount of training data to leverage its advantages. It is inefficient in processing local features and difficult to learn local details and edge information in images without sufficient data support. As the image size increases, the computational complexity of the self-attention mechanism will significantly increase, making the computational cost of ViT higher [2].

Graph Neural Network (GNN) is a deep learning architecture designed to process non-Euclidean data such as graph data, which can effectively model the relationships between nodes and edges [3]. It is widely used in fields such as social network analysis, recommendation systems, and chemical molecular structure analysis. GNN has not been

* Corresponding author: 2021112754@stu.hit.edu.cn

previously applied in the field of computer vision in Euclidean space, but in recent years, some literature has pointed out that by superpixel processing of image data, the image is segmented into a set of small, interrelated regions as graph nodes. GNN can also be used in tasks such as image classification, image semantic segmentation, etc. Compared to CNN, GNN is adept at effectively capturing global information and establishing long-range dependencies across multiple nodes. Compared to ViT, GNN can capture edge information through superpixel algorithm processing, which has advantages in performance on small datasets and training costs on large datasets. As of now, the architecture and design of GNN are still rapidly developing, with great potential. The research on image classification methods based on superpixels and GNN has a certain engineering value for every field that requires image classification.

There are few studies related to image classification methods based on superpixels and GNN, but there is also some literature that proves their feasibility and potential. In a paper published in 2022, a wavelet-based superpixel algorithm called WaveMesh was proposed for image classification, and its feasibility was demonstrated on three benchmark datasets [4]. In a paper published in February 2024, graph neural networks were used to classify grayscale image datasets, demonstrating their powerful performance and low complexity, and emphasizing their hardware friendliness. In a paper published in 2024, GNN was applied to a skin lesion dataset for testing classification models and outperformed architectures such as ResNet18, ResNet34, and ViT-Base in accuracy, demonstrating the potential of GNN in image classification [5, 6].

This research focuses on an image classification method based on superpixels and GNN, which requires three steps. The first step is to choose a suitable superpixel segmentation method. Because the dataset selected for this project is a grayscale image with pixel values of 28*28, it is not meaningful to replace different segmentation methods. This project only tested and used one superpixel segmentation method. The second step is to select the node features of each superpixel region after segmentation for training. This project tested three methods of preserving node features from low to high feature counts. The third step is to select the GNN network for training. This project tested two types of networks with different complexities. Combining three steps, a total of six sets of data were tested.

2 Dataset and Model

2.1 Datasets

This project performs experiments on the Fashion-MNIST dataset [7]. The Fashion-MNIST dataset, as a modern alternative to the classic MNIST dataset, is a clothing classification dataset published by Zalando, a German online fashion retailer. This dataset contains 10 categories and a total of 70000 grayscale images. It contains 60,000 training samples and 10,000 testing samples. Each image is provided at a resolution of 28*28 pixels. The proposal of Fashion-MNIST is intended to replace the classical MNIST dataset, as the latter is already too simple in contemporary machine learning algorithms to effectively distinguish algorithm performance.

2.2 Superpixel Method

The superpixel method used in this project adopts the Felzenszwalb algorithm [8]. The Felzenszwalb algorithm is an image segmentation algorithm based on graph theory methods. The algorithm aims to maximize the similarity between elements within the same part and minimize the similarity between elements in different parts, to evaluate the segmentation

effect. Specifically, the edge weight between two points within the same part is low, while the edge weight between two points in different parts is high. The index of comparing the dissimilarity between pixels on the boundary of two parts and the dissimilarity between adjacent pixels within each part can be used to compare the difference values between parts and within parts and to describe the local characteristics of the data. Compared to some clustering-based segmentation methods such as K-means or mean shift, the Felzenszwalb method tends to preserve the true boundary information in the image, especially when there are significant gradient changes between regions. This applies to the fashion-mnist dataset used in this experiment. The characteristic of better preserving boundary information is helpful for the selection of superpixel region node information in the next step.

2.3 Superpixel Node Features

After segmentation, each superpixel node needs to select an appropriate amount of information and record it before entering the subsequent GNN network training steps. This project intuitively starts from the shape information of each superpixel node and attempts to better preserve the shape information of each node during model training by adding appropriate features. The project tested three methods of storing node information from low to high feature numbers. These methods are:

Feature 1: The average grayscale of pixels within a superpixel node, as well as the vertical and horizontal coordinates of the centroid of the superpixel node. There are a total of three characteristic values.

Feature 2: The average grayscale of pixels within a superpixel node, the vertical and horizontal coordinates of the centroid of the superpixel node, the maximum and minimum vertical and horizontal coordinates of the superpixel node, and the area of the superpixel node. There are a total of eight eigenvalues.

Feature 3: Average grayscale of pixels within superpixel nodes, vertical and horizontal coordinates of superpixel node centroid, maximum and minimum vertical and horizontal coordinates of superpixel nodes, area of superpixel points, and vertical and horizontal coordinates of randomly selected 12 superpixel boundary pixels (if insufficient, repeat pixel selection). There are a total of 32 feature values.

2.4 GNN Configurations

This project uses the implementation provided in PyTorch Geometric [9] to conduct experiments on two configurations. The two configurations are named SimpleGNN and complexGNN respectively, and the specific layer structures of the configurations are shown in Table 1.

Table 1. GNN models configures.

Model Name	Layer	Configuration
simpleGNN	GCNConv	Input: x, Output: 64
	ReLU	Activation function
	GCNConv	Input: 64, Output: 64
	ReLU	Activation function
	Dropout	Dropout rate: 0.5
	global_mean_pool	Pooling function (Global Mean Pooling)
	Linear	Input: 64, Output: 10
complexGNN	GCNConv	Input: x, Output: 128
	ReLU	Activation function
	GCNConv	Input: 128, Output: 128
	ReLU	Activation function
	GCNConv	Input: 128, Output: 128
	ReLU	Activation function
	Dropout	Dropout rate: 0.5
	global_mean_pool	Pooling function (Global Mean Pooling)
	Linear	Input: 128, Output: 128
	ReLU	Activation function
	Linear	Input: 128, Output: 128
	ReLU	Activation function
	Linear	Input: 128, Output: 10

3 Experimental Results and Analysis

3.1 Superpixel Node Features

The project randomly selected 48000 training set samples and 12000 testing set samples from 60000 fashion-mnist datasets. The project uses Adam optimizer [10]. Compared to other optimizers, the Adam optimizer can better navigate on the loss surface. In the model named simpleGNN, the initial learning rate is 0.005. In the model named complexGNN, the initial learning rate is 0.0005. The model is trained for 30 cycles. For each combination of superpixel node features and GNN configuration, the training and testing sets were shuffled five times, and record the average of the highest accuracy in each test.

3.2 Experimental Result

The specific experimental configuration and test results for each experimental group are shown in Table 2.

Table 2. Performance on fashion-mnist dataset.

#	Superpixel node features	GNN model name	Acc(%)
1	Features1	simpleGNN	63.75
2	Features2	simpleGNN	67.55
3	Features3	simpleGNN	73.08
4	Features1	complexGNN	68.08
5	Features2	complexGNN	71.32
6	Features3	complexGNN	76.69

The experimental results show that the classification accuracy of the worst group is 63.75%, and the classification accuracy of the best group is 76.69%. Through simple design,

image classification methods based on superpixels and GNN can have good upper and lower limits, but there is still some gap in accuracy compared to CNN structures, which typically achieve the performance of over 80% on fashion-mnist datasets. Firstly, the accuracy of image classification tasks improves with an increase in the number of node features, indicating that incorporating information such as boundary nodes of superpixel regions into training can improve the final accuracy of image classification. The possible reason is that such operations can improve the utilization of information in superpixel node segmentation, and save more information that is helpful for image classification during the training process and in the final model. Then, the accuracy of image classification tasks improves with increasing network complexity. The possible reason is that the images segmented by superpixels have rich information, and the complexity of the network in the experiment is not sufficient to achieve the best fitting effect in training, so more complex networks are still needed. In fact, after 30 rounds of training in the experiment, the loss value still significantly decreased with the training performance of each cycle, indicating that the method has high fitting difficulty, and indirectly indicating that combining superpixels with graph neural networks can learn very rich information from images, reflecting the high upper limit that high methods may achieve.

4 Discussion

Due to hardware limitations, this experiment was only conducted on a dataset with a pixel size of 28 * 28. Obviously, research conducted only on small datasets has significant limitations. On small datasets, there is no significant difference in the performance of superpixel segmentation methods, and it is impossible to experimentally determine whether a method is excellent. The method of selecting node feature information is the same, and the differences cannot be reflected in nodes that are too simple in small datasets. Moreover, small datasets cannot test complex GNN training networks, and the performance of methods cannot be evaluated from the perspective of computational cost. Therefore, migrating to datasets with larger width and height images in the future can lead to a considerable number of research projects. Firstly, the segmentation method for superpixels can be improved. For images with larger width and length, their impact on the accuracy of image classification tasks is more significant. Therefore, further research can be conducted on superpixel segmentation methods on large datasets. Then, the improvement lies in the method of selecting node features. The method for selecting boundary node information in superpixel regions in this project is random selection, and a better method is thoroughly studied to select boundary node information in superpixel regions, in order to improve the accuracy of the final image classification task. It can also handle complex GNN networks. On large datasets, more complex and advanced networks are needed to fit a large amount of information, and network design skills are particularly important. In addition, the computational cost of image classification methods based on superpixels and GNN can be tested on large datasets and compared with the ViT model. Finally, we can try to extend this method to the field of semantic segmentation in computer vision, which also has great potential.

5 Conclusion

From this project, it can be seen that under simple preprocessing and training methods, the image classification method based on superpixels and GNN has shown good performance in image classification tasks on small datasets. It has been proven that starting from the shape information of superpixel nodes, adding node information such as superpixel region boundary coordinates during the training process can improve the accuracy of the final image

classification task. In addition, it has been confirmed that this method has high difficulty in training and fitting, requiring complex networks. However, this study was unable to test the training cost and efficiency on large datasets and compare it with ViT. The research methods in the study have not yet reached the performance limit of small dataset image classification methods based on superpixels and GNN. This reflects that graph neural networks can not only be applied to non-Euclidean data but also have great potential on Euclidean data such as images. In summary, this study focuses on a rarely researched method in the field of image classification, conducts preliminary research, proves its feasibility, and provides future improvement plans. In recent years, graph neural networks have gradually become popular and have been proven to be excellent networks in other studies. Or refer to the potential research directions mentioned in this study, or better research directions not mentioned, hoping that someone can continue to delve deeper into the use of this method in image tasks. ViT and CNN are both quite mature in image tasks, and the improvement of model performance mainly depends on the improvement of network parameters and computing power. Researching and developing their potential in new methods may bring new breakthroughs in the field of image classification.

References

1. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020)
2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.A. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017), 30, 5998-6008 (2017)
3. F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, and G. Monfardini, The graph neural network model, IEEE Trans. Neural Networks. 20(1), 61–80 (2009)
4. V. Vasudevan, M. Bassenne, M.T. Islam, and L. Xing, Image classification using graph neural network and multiscale wavelet superpixels, Pattern Recognit. Lett. 166, 89–96 (2023)
5. K.P. Santoso, R.V.H. Ginardi, R.A. Sastrowardoyo, and F.A. Madany, Leveraging spatial and semantic feature extraction for skin cancer diagnosis with capsule networks and graph neural networks, arXiv preprint arXiv:2403.12009 (2024)
6. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), 770-778
7. H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, arXiv preprint arXiv:1708.07747 (2017)
8. P.F. Felzenszwalb and D.P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59(2), 167–181 (2004)
9. T. Kipf, M. Welling, et al., PyTorch Geometric: Deep Learning on Graphs, arXiv preprint arXiv:1903.02428 (2019)
10. D.P. Kingma, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014)