Journal of Physics: Conference Series

A Graph Neural Network for superpixel image classification

A Graph Neural Network for superpixel image classification

Jianwu Long[1], Zeran yan[1*] and Hongfa chen[1]

[1] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

[*]Corresponding author's e-mail: yan3142537@gmail.com

Abstract. The classification of superpixel images by graph neural networks has gradually become a research hotspot. It is a crucial issue to embed super-pixel images from low-dimensional to high-dimensional so as to turn complex image information into graph signals. This paper proposes a method for image classification using a graph neural network (GNN) model. We convert the input image into a region adjacency graph (RAG) composed of superpixels as nodes, and use residual and concat structure to extract deep features. Finally, the loss function that increases the distance between classes and compactness within classes is used as supervision. Experiments have been tested with different numbers of superpixels on multiple datasets, and the results show that our method has a great performance in superpixel images classification.

1. Introduction

In the past few years, the Graph Neural Network (GNN) has received great attention from academia and industry, which has led to rapid development in many fields. Superpixel is a collection of pixels with similar colors and other low-level attributes (such as location) into a perceptually meaningful representation unit (region or segment). These over-segmented and simplified images can be applied to many common tasks in computer vision, which include image classification. Monti et al. [1] used the GNN to classify the superpixel images for the first time, as well as extend their framework MoNET to a universal framework for graph learning. Their framework weights the neighborhood aggregation by learning the scale factor of the geometric distance.

Later, some researchers have regarded the superpixel image classification as a graph convolution method and one of the evaluation criteria of GNN performance. Matthias et al. [2] proposed a graph convolution operator based on the B-spline kernel, called SplineCNN, which can be used as a convolution operator for extracting graph features to build an end-to-end deep GNN. Ushasi et al. [3] used multi-scale superpixels as image preprocessing methods combined with GCN [4] algorithm to propose a siamese graph convolutional network for image retrieval. Boris et al. [5] proposed a method of graph attention pooling, which is compared with GIN [6], GCN on Mnist-75 dataset, and this paper explored the anti-noise ability and robustness of different pooling methods by adding random Gaussian noise to the Mnist-75. After that, Boris et al. [7] constructed a hierarchical multigraph network (Hierarchical Multigraph Networks), which takes superpixels with different scale as input and combines different granularities information to complete image classification.

Our method HGNN uses GAT [8] as the graph convolution operator, and combines the residual and concat structure to build a GNN to improve the feature extraction ability. Finally, we use a loss function that increases the distance between classes and compactness within classes to supervise model learning.

2. Method

2.1. Superpixel graph

Nowadays, there are many technics that can generate superpixels from images, such as SLIC [9], SEEDS [10] and etc. Like other related works, we use the SLIC to segment superpixels because of its efficiency. After segmentation, we define nodes from superpixels and connect them from K nearest neighbors to generate region adjacency graph (RAG), Figure 1 describes the entire process from the original image to the RAG generation.

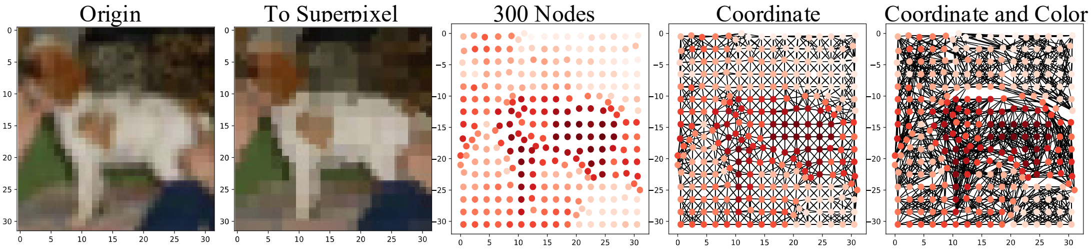

Figure 1. Image to graph.

There are many ways to generate nodes features, such as coordinate information (superpixel's center of gravity), the mean and variance of each superpixel RGB channels. It can be seen from Figure 1 that after the original image is segmented by the SLIC, the target features have been initially extracted from the obtained superpixel image, and then the superpixels are defined as nodes according to the coordinates. Because the features are not particularly obvious when the nodes only use coordinate information for connection, we use both coordinates and color for connection, and the simple outline of the target can be seen. Therefore, we use coordinates and mean RGB color for initialization. For each superpixel, where the coordinates are its center of gravity, the feature X of the superpixel image can be represented by the equation (1) and equation (2).

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)^T \tag{1}$$

$$\boldsymbol{x}_i = \left[\frac{1}{N_i}\sum_{j=1}^{N_i}(R_j, G_j, B_j, a_j, b_j)\right]^T \tag{2}$$

where $\boldsymbol{x_i}$ is the node feature with superpixel label is i, and N_i is the number of pixels which label is i, a_j and b_j is pixel's position in the image.

2.2. Hierarchical Graph Neural Network

The nodes in graph convolutional neural network usually tend to over-smooth (OS) as the increasing iteration and deeper layers, that is the nodes of the same subgraph have the same values or features. We use two aspects to solve OS. First, residual and concat structure are used for the node graph neural network to alleviate OS situation. Second, GAT is the operater that pays attention to the information of adjacent nodes, which can suppress the over-smooth through aggregating neibourhood features.

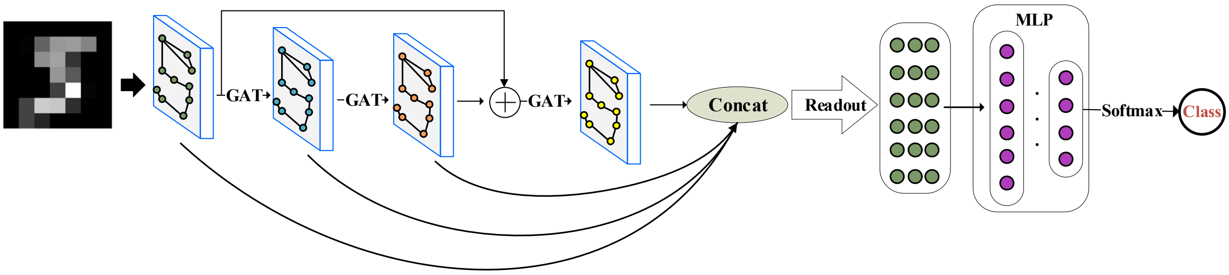

Figure 2. Hierarchical Graph Neural Network.

Therefore, HGNN is composed of GAT, Residual connection and Concat, and the structure is shown in Figure 2, where the Readout operation of the graph is also equal to Graph Coarsening or Graph Pooling. In the graph classification task, Readout is used to reduce the number of graph nodes, reduce the interference of redundant information on classification, and reduce the amounts of parameters that need to be calculated, which is convenient for the subsequent MLP layer to perform calculations. The Readout function can be a simple permutation invariant function, such as Sum, Mean, Max and etc., or a multilayer perceptron can be used as the decoder structure, our Readout is Mean for each graph.

2.3. Loss function

The loss function of HGNN is ArcFace loss (Additive Angular Margin Loss, Additive Angular Margin Loss) [11] and cross-entropy (CE) loss. Among them, the CE loss is as the equation (3).

$$L_{CE} = -\sum_{i=1}^{n} p_i \log\left(q_i\right) \tag{3}$$

where p_i is ground truth and q_i is prediction.

ArcFace loss used for face recognition at the beginning, which improves the additive angle interval on the basis of Softmax loss (equation (4)) to increases the distance between classes and compactness within classes.

$$L_{Softmax} = -\frac{1}{m}\sum_{i=1}^{m} \log\left(\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}\right) \tag{4}$$

where $\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j$ is the output of the fully connection (FC). When calculating $L_{softmax}$, as the $\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}$ increases, more ith samples can be within the decision boundary. However, $L_{softmax}$ ignores the intra-class and inter-class constraint, so ArcFace loss adds an angle interval t to the angle θ between $\boldsymbol{x}_i$ and $\boldsymbol{W}_{ji}$, and penalizes the angle between the features and weights in an additive manner, so that GNN can learn information more efficiently. $L_{ArcFace}$ such as equation (5).

$$L_{ArcFace} = -\frac{1}{m}\sum_{i=1}^{m} \log\left(\frac{e^{s\times(\cos(\theta_{y_i}+t))}}{e^{s\times(\cos(\theta_{y_i}+t))} + \sum_{j=1,j\neq y_i}^{n} e^{s\times\cos\theta_j}}\right) \tag{5}$$

where θ_j is the angle of the weight $\boldsymbol{W}_j$ and the feature $\boldsymbol{x}_i$, is calculated by $\boldsymbol{W}_j^T \boldsymbol{x}_i = ||\boldsymbol{W}_j||\ ||\boldsymbol{x}_i||\cos\ \theta_j$, so is θ_{y_i}.In summary, the final loss function L used in this paper is as follows equation (6):

$$L = L_{CE} + L_{ArcFace} \tag{6}$$

3. Experiment

Our experiments on the Mnist dataset, Cifar10 dataset and Euro_SAT dataset. The size of Mnist is 28×28×1, the train set has 60,000 pictures, and the test set has 10,000 pictures. The picture size of Cifar10 is 32×32×3, the train set has 50,000 pictures, and the test set has 10,000 pictures. The picture size of Euro_SAT is 64×64×3, the train set has 24300 pictures, and the test set has 2700 pictures.

We experiment the HGNN and other graph convolution methods under different numbers of nodes on each dataset. For data, as shown in Figure 3 to Figure 6, the more the number of superpixels, the more detailed information the node contains, and the more similar the superpixel image will be to the original image.

Our method compares the excellent graph convolution methods in recent years with different numbers of superpixels. Except for SplineCNN which has three layers, all graph convolution methods have a four-layer structure. The dimensions of each layer of GCN [4], GAT [8], GraphSage [12], MoNet [1], GIN [6] and SplineCNN [2] are 146, 152, 90, 270 and 220, 320 respectively. In our method, except for the first layer dimension is 3(5)×320, the others are 320×320.

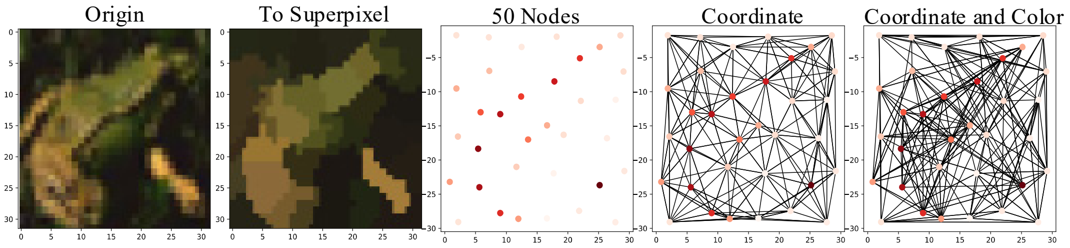

Figure 3. Cifar10 with 50 nodes.

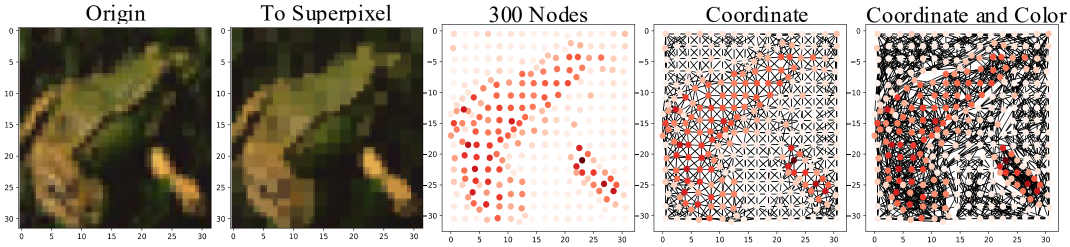

Figure 4. Cifar10 with 300 nodes.

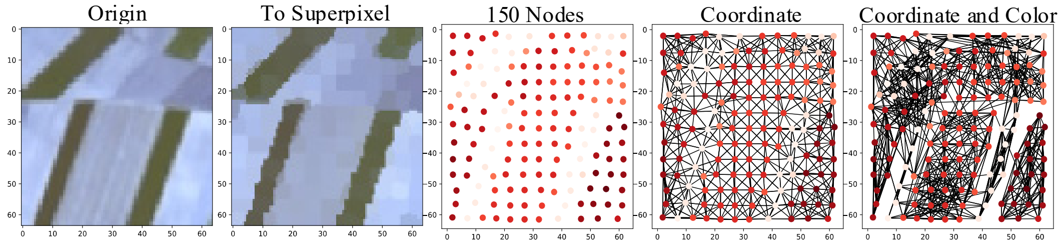

Figure 5. Euro_SAT with 150 nodes.

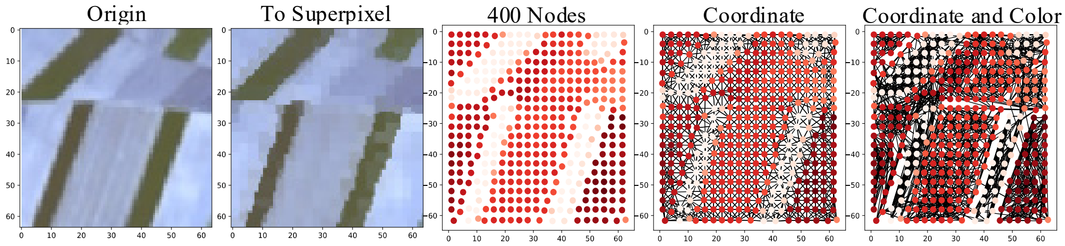

Figure 6. Euro_SAT with 400 nodes.

Table 1, Table 2 and Table 3 show the classification results, where Mnist-75 is the Mnist image with 75 superpixels, so are other datasets. It can be seen that HGNN has advantages in superpixel image classification. The evaluation is classification accuracy (Acc, proportion of correctly classified).

Table 1. Experiments in Mnist with different number of nodes.

Method	Acc		
	Mnist-75	Mnist-150	Mnist-200
GCN	89.99	91.12	93.24
GAT	95.62	97.01	98.23
GraphSage	97.09	98.07	-
MoNet	90.36	92.24	-
GIN	93.91	96.49	-
SplineCNN	95.22	97.48	98.11
Ours	**97.11**	**98.27**	**98.50**

Table 2. Experiments in Cifar10 with different number of nodes.

Method	Acc	
	Cifar10-150	Cifar10-200
GCN	54.46	58.28
GAT	65.40	69.09
GraphSage	65.93	-
MoNet	53.42	-
GIN	53.28	-
Ours	**66.24**	**70.61**

Table 3. Experiments in Euro_SAT with different number of nodes.

Method	Acc		
	Euro_SAT-75	Euro_SAT-150	Euro_SAT-200
GCN	59.54	62.33	62.25
GAT	66.12	66.74	67.66
SplineCNN	66.00	70.12	71.24
Ours	**75.22**	**78.13**	**77.88**

In Mnist and Cifar10, it can be seen that the number of superpixels is positively correlated with image characteristics, and more superpixels result in higher ACC. However, in the geographic remote sensing dataset Euro_SAT, because the image has a high degree of feature aggregation, even if it uses fewer superpixels, it can fit the boundary well and aggregate points with similar pixel values into one class. It can be seen from the classification accuracy that the classification results from 75 nodes to 200 nodes are not much different. In our method, the accuracy of 75 nodes is 75.22%, and the accuracy of 200 nodes is 77.88%. It is worth mentioning that, because the node retains more redundant information to participate in the calculation when the number of super pixels is large, the classification accuracy of Euro_SAT-150 is even higher than that of Euro_SAT-200.

4. Conclusion

This paper explores the impact of the number of superpixels on the results of image classification, and proposes a HGNN that uses the GAT as graph convolution. HGNN combines the residual structure and the concat structure, and uses ArcFace loss and cross entropy as the loss function. The initial guess is that the more superpixels, the higher the classification accuracy. As the experiment continues, in some cases, the number of superpixels may not necessarily be as many as possible. In order to verify the image recognition ability of the method proposed in this paper, experiments were carried out on the Mnist dataset, the Cifar10 dataset and the Euro_SAT dataset. The experimental results show that HGNN has advantages in image classification compared with the existing graph convolution methods.

Acknowledgments

This work was supported by the Humanities and Social Sciences Research Key Program of Chongqing Municipal Education Commission (Grant No. 17SKG136), the National Natural Science Foundation of China for Young Scientists (Grant No. 61502065) andChongqing University of Technology Postgraduate Innovation Project (clgycx20202097).

References

[1] Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5115-5124).

[2] Matthias F, Jan E.L., Frank W, and Heinrich M. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 869–877, 2018.

[3] Chaudhuri U, Banerjee B, Bhattacharya A. Siamese graph convolutional network for content based remote sensing image retrieval[J]. Computer Vision and Image Understanding, 2019, 184: 22-30.

[4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR), 2017.

[5] Knyazev B, Taylor G W, Amer M. Understanding Attention and Generalization in Graph Neural Networks[J]. Advances in Neural Information Processing Systems, 2019, 32: 4202-4212.

[6] Keyulu X, Weihua H, Jure L, and Stefanie J. How powerful are graph neural networks? In International Conference on Learning Representations (ICLR), 2019.

[7] Knyazev B, Lin X, Amer M R, et al. Image classification with hierarchical multigraph networks[J]. arXiv preprint arXiv:1907.09000, 2019.

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

[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] Radhakrishna A, Sabine S. Superpixels and polygons using simple noniterative clustering. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 4895–4904. IEEE Computer Society, 2017.

[11] Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(pp. 4690-4699).

[12] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs[J]. arXiv preprint arXiv:1706.02216, 2017.