Superpixel-based Graph Convolutional Network for Semantic Segmentation

Hoin Jung
Seoul National University
hoyin@snu.ac.kr

Seong Yeon Park
Seoul National University
psyeon990@snu.ac.kr

Su Yang
Seoul National University
s8431@snu.ac.kr

Jin Kim
Seoul National University
kimjin116@snu.ac.kr

Abstract

The encoder-decoder structured Convolutional Networks(CNNs) are a general approach for semantic segmentation tasks. However, it is hard to capture precise boundaries of objects, and the boundary information loss is inevitable since the input image is contracted to small-sized features through the encoder and then extended as the original size through the decoder. To tackle this problem, we propose a whole new approach, Superpixel-based Graph Convolutional Network, not containing any pooling layer thus, preserving the shape of a target object. At first, the superpixel algorithm segments an image into plausible clusters with RGB values of pixels. Then, Graph Convolutional Networks(GCNs) predict an assigned label of each superpixel, regarding them as a node of a graph. In other words, our GCN framework conducts a node prediction for each image converted as a superpixel graph. We utilize two graph convolutions to capture the semantics of nodes, spectral convolutions with topology adaptiveness and spatial convolutions with weighted node sampling. Also, we propose a novel loss function, Superpixel Penalty Loss, to address imbalance problems of the classes and the size of superpixels. Experiments are performed on the UAVid dataset, with has ambiguous boundaries in their target objects. Although the proposed method does not reach the state-of-the-art performance, it shows comparable ability to classify each pixel's label and expands the concept of the GCN combined with superpixel into semantic segmentation. GitHub Code

1. Introduction

Deep Neural Networks have achieved significant advancements in semantic segmentation on account of recent improvements. Many vision-based applications, including autonomous driving, remote sensing, and medical image analysis, benefit from semantic segmentation. FCN[11] proposed an end-to-end and pixels-to-pixels method on semantic segmentation, using deconvolution[14] and skip-connection. However, due to their shallow upsampling layers of FCN, the ability to classify pixels was insufficient, and fragmentation occurred. U-Net[16] solved this problem by adopting deep encoder-decoder architecture. Especially, scene segmentation[25, 26, 20] is a difficult but important endeavor to divide the categories to each pixel in scene pictures. It's vital to enhance feature similarity between objects while maintaining feature differentiation amongst them. Due to its huge resolution, precise segmentation is a prominent issue. It is hard to capture precise boundaries of objects, and the boundary information loss is inevitable since the input image is contracted to small-sized features through the encoder and then extended as the original size through the decoder

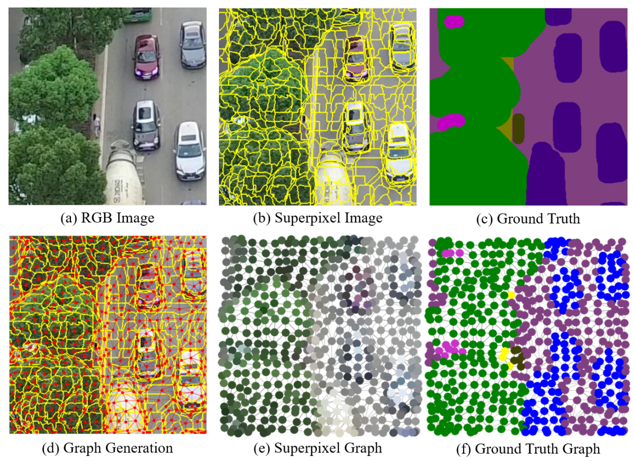

Figure 1. Example of graph generation via superpixel. Each superpixel cluster is treated as a node of a graph. Each node represent a diminutive region containing color and spatial information and its class labels.

To tackle this problem, we adopt both Superpixel and

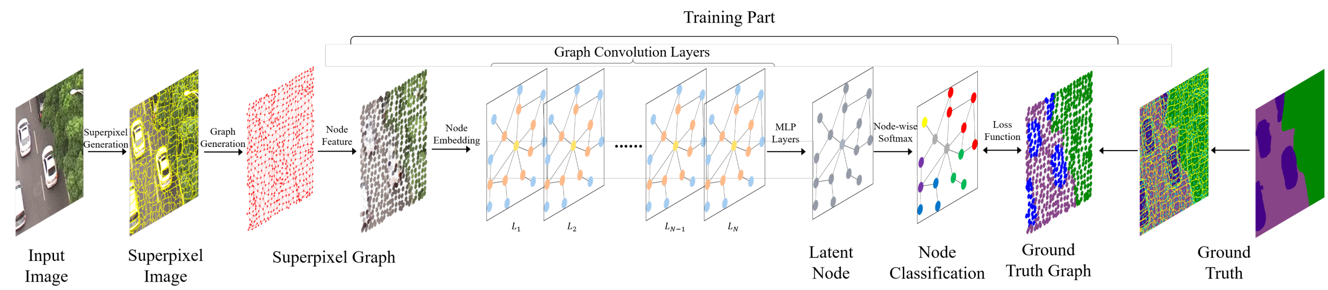

Figure 2. All the train process are conducted under the graph state. Graphs are generated as a preprocessing with superpixel map. Also, same superpixel mapping are applied for the ground truth image to generate ground truth nodes.

Graph Convolutional Networks(GCN) instead of 2D CNNs. Superpixel is one of the commonly used approach to segment an image into a number of clusters by grouping pixels into perceptually meaningful atomic regions. SLIC[1] adopts k-means clustering to group nearest pixels w.r.t both color and spatial distance, by converting CIELab color space. SSN[7] first suggested deep learning based superpixel method, defining soft-association map, also called differentiable SLIC. SFCN[23] extend the concept of SSN by adopting FCN[11] as prior step before obtaining superpixel association map. Also, LSN-Net[27] suggested non-iterative lifelong learning strategy with unsupervised CNN, while reducing computation complexity. Unlike semantic segmentation, superpixel does not require significant contraction of image. Therefore, superpixel is suitable to maintain a boundary information of relatively small object or ambiguous edges, which are easily disregarded in encoder-decoder structure. However, Superpixel cannot classify the label of each cluster though it can segment images successfully.

To compensate this problem, we also adopt GCN to capture relational information of objects, i.e. Superpixel. GCN integrates the local information of neighborhood node including their RGB values and geometrics. The objects and their surrounding information can be expressed as a graph whose nodes contain color and spatial information each Superpixel, whereas the edges express the spatial relationship between the Superpixel. In this paper, superpixel is used as a preprocessing of our framework to convert the grid-structured image into graph-structured image. In other words, our GCN framework conducts a node prediction for each image converted as a superpixel graph.

2. Preliminaries

2.1. Graph Convolutional Network

GCN is a network that has been applied to graph-structured data such as road networks, protein-protein interaction, and social networks. Within various kinds of social and physical phenomena that can be interpreted with the graph structure, GNN efficiently captures the relations between nodes and edges using their given attributes. To update the state of each node and to output the desired feature from a graph, GNN mainly adopts convolutional operation, which shares the same properties with CNN such as local connectivity, learnable filters, and use of multi-layer. GCN can be categorized by a spectral and spatial convolutional network.

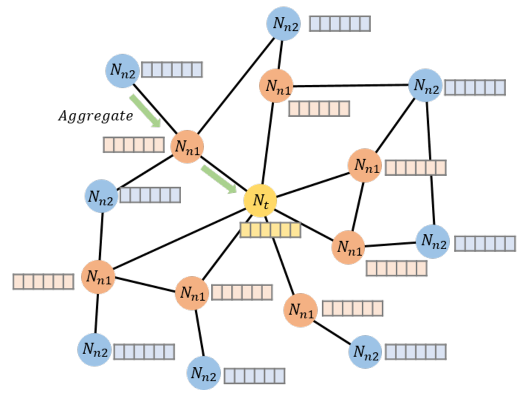

Figure 3. Each convolutional layer aggregate neighborhood node's embedding information N_n, k to the target node N_t.

2.2. Spectral Graph Convolution

Spectral graph convolution[8, 2, 3] uses spectral filters based on a Fourier transform of graph signal, an eigen-decomposition of graph Laplacian matrix. However, it requires an entire and fixed graph since the graph Laplacian depends on the overall graph structure. Thus the model cannot be adapted on newly generated nodes which means the change of the original graph structure. To overcome its limitation, modification of previous networks such as TAGCN[4], SGCN[22], and APPNP[9] have been proposed, which are adaptive to the topology of arbitrary graph and have lower computational complexity.

2.3. Spatial Graph Convolution

As opposed to this transductive learning, spatial graph convolution is inductive learning which can be generalized to previously unseen data. Spatial graph convolution[13] achieves its inductiveness by convolving graph with spatial filters while aggregating information of locally connected neighborhoods. Spatial convolutional networks learn a node embedding function only reflecting the node's local neighborhood instead of referring entire graph, the model successfully works on unseen graphs or continuous changes in the graph. GraphSAGE[5] randomly samples target nodes and their fixed number of neighborhoods. Then these sampled subgraphs go through a learnable aggregator sharing the same weights. Attention based spatial convolutions, such as GAT[18], AGNN[17], have been also proposed to dynamically adjust weights of neighbor nodes.

3. Proposed Method

3.1. Superpixel Graph Generation

Prior to the graph generation, superpixel segmentation is conducted as a preprocessing. Each superpixel represents a group of pixels containing similar spatial and color information. Superpixel is very efficient method to segments region sensitively, retaining boundary well, while each cluster includes information about original image, C_{mean} and P_{mean},

$$C_{j,mean} = \frac{\sum_i^{N_j}(R,G,B)_i}{N_{j,pixel}} \tag{1}$$

$$P_{j,mean} = \frac{\sum_i^{N_j}(x,y)_i}{N_{j,pixel}}, \tag{2}$$

where $N_{j,pixel}$ is the number of pixels in j-th cluster.

Each superpixel cluster is allocated as a node V of a graph $G = (V, E)$, which have 5-dimension features in every nodes $\mathbf{h}_{i,j} = [C_{mean}|P_{mean}]$. The undirected edges E in a graph are simply generated as adjacent relations between neighborhood nodes.

In this paper, we adopts SFCN[23] rather than SLIC [1] to obtain more precise boundaries to distinguish adhering objects.

3.2. Superpixel Penalty Loss Function

In this paper, we propose a novel *Superpixel penalty loss* which is designed to address two main problems derived from Superpixel graph generation and node classification. The first is the extreme imbalance between node classes (e.g., Background versus Person) in a graph. In this case, directly training a GNN classifier with a graph would under-represent samples from those minority classes and result in sub-optimal performance. The second is that Superpixels have a different number of pixels in each Superpixel and they do not carry the information about the amount of pixels in each node. To mitigate these problems, we introduce *Superpixel penalty loss* that adds the class balanced and Superpixel weights to cross-entropy loss (CE) for node classification. The class balanced CE in *Superpixel penalty loss* is defined as:

$$l_k = -w_k y_k \cdot \log \frac{exp(x_{k,y_k})}{\sum_{c=1}^{C} exp(x_{k,c})} \tag{3}$$

where x is the input, y is the target, w is a class balanced weight, and C is the number of class. w_k can be calculate as

$$w_k = \frac{N - n_k}{N} \tag{4}$$

where N is total samples and n is the number of samples in each class. Following the above equation, the class balanced weight gives a more penalty to rare samples than others. After calculating losses of class balanced CE in each node, we apply the Superpixel weights to the losses:

$$SPL_k = s_k \cdot l_k \tag{5}$$

where SPL_k is *Superpixel penalty loss* in each node k. s_k is a superpixel weight in each node k, as follow

$$s_k = -\frac{1+\epsilon}{\log p_k + \epsilon} \tag{6}$$

where p_k is the normalized number of pixels in a Superpixel generated into each node k. ϵ is the constant value for numerical stability to avoid zero division error, set as 10^{-5}. We impose a greater penalty for nodes generated by Superpixels containing more pixels than other nodes. Finally, *Superpixel penalty loss* is calculated by

$$SPL = \frac{1}{N}([l_1, \cdots, l_N]^T \cdot [s_1, \cdots, s_N]) \tag{7}$$

3.3. Spectral Approach

3.3.1 Topology Adaptive Graph Convoloution Layer

Topology Adaptive Graph Convoloution Network(TAGCN)[4] is one of the simplest convolutional layer for the graph-structured data. Based on the GCN[8], TAGCN can adapt higher-order relations between K-hops nodes. Each $k \in \{1, 2, \cdots, K\}$ means a k-size learnable graph convolution filter, likewise a squared convolution filter of grid structured data. An output embedding of a vertex is the weighted sum of these filter's outputs.

$$X' = \sum_{k=0}^{K}(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^k X\Theta_k, \tag{8}$$

where A denotes the adjacency matrix, $D_{ii} = \sum_{j=0} A_{ij}$ is diagonal degree matrix, Θ_k denotes the linear weights to

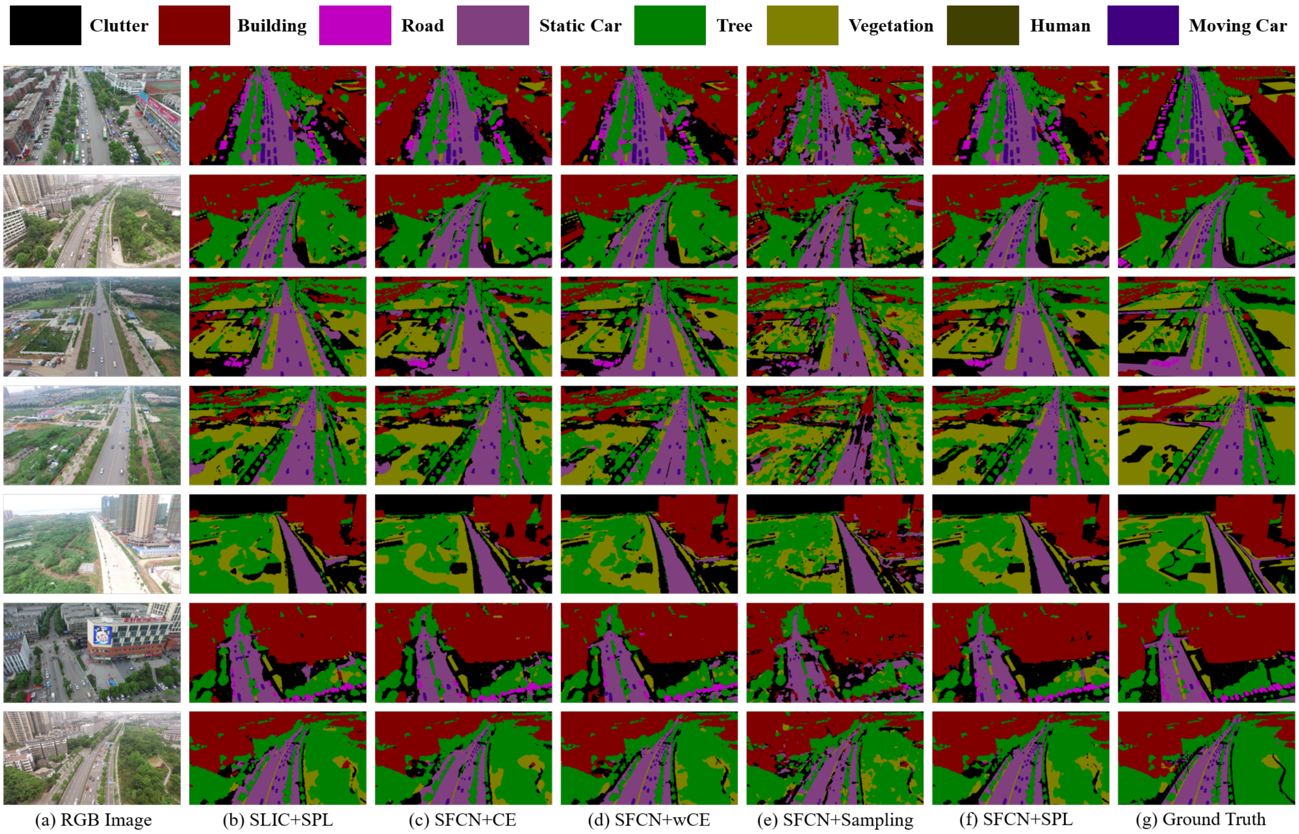

Figure 4. The example results of our method. (a) and (g) are the RGB images from UaVid dataset and the corresponding ground truth annotation. (b), (c), (d), (e), and (f) are the predicted segmentation maps of GCN with SLIC+SPL, SFCN+CE, SFCN+wCE, SFCN+Sampling, and SFCN+SPL, respectively.

Method	Clutter	Building	Road	Static Car	Tree	Vegetation	Human	Moving Car	mIoU
U-Net[16]	40.3	70.7	63.5	11.9	67.2	35.5	00.0	47.5	40.9
BiSeNet*[24]	64.7	85.7	61.1	63.4	78.3	77.3	17.5	48.6	61.5
BANet*[21]	66.6	85.4	80.7	52.8	78.9	62.1	21.0	69.3	64.6
Ours (SPL)	50.5	79.9	64.9	35.1	67.4	48.4	8.4	40.9	49.4

Table 1. The experimental results on the UAVid dataset. Asterisks of BANet[21] and BiSeNet[24] means the result mentioned in each paper.

sum the results of different hops together. TAGCN layer extracts both vertex features and correlation strength between vertices.

3.3.2 Multi-layer Loss

As the model gets deeper, GCN suffers from over-smoothing probelm[10, 15], which is an main obstacle for GCN to have richer representations. Since GCN aggregates the features of adjacent nodes inherently, stacking more layers lead to aggregating more information through further hops. Thus, it results in convergence of node representation, which is called over-smoothing, and it is why many research on GCN have shallow networks. However, more layers still achieve better performance, we apply the multi-layer loss to TAGCN to handle the over-smoothing problem and to make deep GCN.

$$L_{overall} = \frac{1}{3}(SPL_{h_4} + SPL_{h_8} + SPL_{h_{12}}) \tag{9}$$

where SPL_{h_i} denotes superpixel penalty loss at i^{th} hidden layer, which will be further explained later. Proposed multi-layer loss is the average of loss computed at intermediate convolutional layers after passing each MLP layers. In our experiment, we extract intermediate loss from 4th, 8th, 12th graph convolutional layers.

3.4. Spatial Approach

3.4.1 GraphSAGE with Weighted Node Sampling

To handle the class imbalance problem and to enhance the generality of networks, we adopt the GraphSAGE networks, which sample subgraphs and aggregate the node information. Unlike other node classification is conducted on a single graph, our model is applied to multiple graphs at the same time by constructing one batch graph from multiple input images while maintaining its inductiveness. Therefore, we first construct one large graph from multiple graphs without having any connection between each graph. Then we sample target nodes with different weight from this batch graph. Also, we sample the fixed number of neighbor nodes in each layer, not the entire neighbor nodes. With these approach, GraphSAGE networks have the ability to get more subgraphs on smaller class as well as encouraging generality of model by dropping some edges between nodes.

4. Experimental Result

4.1. Dataset

In this study, extensive experiments were conducted to evaluate the proposed method for UAVid dataset[12]. The UAVid dataset consists of 42 high-resolution sequential images in total capturing the urban scenes from an unmanned aerial vehicle(UAV), with 8 classes. Each sequence has 10 images. In our experiments, the sequential data would be considered as an individual data. Moreover, The sequence are split into 20 sequence for train, 7 sequence for validation, and 15 sequence for test. However, the test subset does not include ground truth data. So we use validation subset as test subset. Validation data for training will be obtained by randomly splitting from training data for every epoch. Therefore, the experiments are conducted on 200 images for training with randomly chosen 20% validation subset, and 70 images for test, each of size 4096×2160 or 3840×2160. Also, we modified train subset images into 2048×2048 cropped image, allowing overlapping.

4.2. Implementation

We are conducting two types of experiment; Graph-wise and Node Sampling. Both methods use same graph dataset which is made by preprocessing containing node features, edge relations, node labels. As each image is converted to corresponding graph, graphs can allocated batch being regarded as images. A GNN model train the node features considering the edges around them, and final node embedding is obtained. This method is simply same with semantic segmentation, just replacing encoder-decoder architecture to graph network.

On the other hand, we have tried to adopt Node Sampling. Entire dataset is regarded as a single large graph, and target nodes are randomly sampled to train as same number as batch size. Although this might not a efficient way to train when it comes to the time cost, it allows model to oversample the scarce classes giving more opportunity to be trained them. The training procedure adopts early stopping strategy.

For Graph-Wise model, 12 TAGCN layers with 256 channels are used. For Node Sampling model, Graph Sage is adopted as a convolutional filter, with same number of layer and channel. Initial learning rate is $lr = 0.001$ with multi-step learning scheduler. Adam optimizer are used with decay 0.0001. All nodes are trained for the Graph-Wise model with Dropout rate 0.5, while only three neighbor node are sampled for all layers in the node sampler.

4.3. Evaluation Metric

Node accuracy[6] is used in Node Classification tasks of GNNs. However, although the proposed approach adopts GNNs, the node accuracy does not reflect perfectly the performance of semantic segmentation. Instead, mIoU is mainly used to evaluate the performance of given networks. The Jaccard Index(mIoU) is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth. To calculate the intersection and union, we invert graph to image again. In our experiments, even if a model could get further node accuracy, the mIoU score does not increased proportionally.

4.4. Result Analysis

To verify the contribution of *SPL*, SFCN[23], Node Sampling, and TAGCN[4], we have conducted several experiments. The results are shown in Table 2 and Table 3.

In terms of superpixel methods, SFCN outperforms SLIC[1]. Although node accuracy of SLIC was higher than SFCN in our SGCN, the overall mIoU score was poor since SLIC is not able to capture the precise boundaries of objects. The SPL also outweigh other loss function. Weighted cross entropy loss considers the imbalance in the number of class. In addition to that, SPL reflects the size of each superpixel. Also, we expected Node Sampling to obtain enhanced results especially for small objects such as human and car. However, the results show that Node Sampling is not helpful to segmentation, so Graph-wise method is selected.

On the other hand, we have adopted various convolutional filters such as APPNP[9], SGCN[22], CHEBConv[3], GraphSAGE[5], and TAGCN[4]. Among the various convolutional network and loss functions, our final frame work achieved 49.4% of mIoU score, as shown in Table 2 and Table 3. Moreover, the final was better than U-Net[16]. However, we couldn't reach the state-of-the-art performance

Method	Clutter	Building	Road	Static Car	Tree	Vegetation	Human	Moving Car	mIoU
SFCN + SAMPLING	33.9	64.6	45.0	5.7	58.2	37.4	0.0	10.3	31.9
SLIC + SPL	48.5	78.4	62.7	30.3	67.3	47.3	4.7	32.9	46.6
SFCN + CE	46.2	76.8	57.9	26.8	65.1	45.0	5.6	28.4	44.0
SFCN + WCE	50.1	79.4	63.9	38.8	67.3	47.9	8.5	35.3	48.9
SFCN + SPL	50.5	79.9	64.9	35.1	67.4	48.4	8.4	40.9	49.4

Table 2. Ablation study for various loss function and training strategy on the UAVid dataset.

Method	Clutter	Building	Road	Static Car	Tree	Vegetation	Human	Moving Car	mIoU
APPNP[9]	23.6	48.8	35.8	1.2	47.7	25.7	0.0	7.0	23.7
SGCN[22]	26.5	54.2	32.6	5.8	50.8	30.1	0.0	5.1	25.6
CHEB[3]	12.4	48.4	24.0	6.3	51.2	31.0	0.0	12.8	23.3
GraphSAGE[5]	30.4	61.8	43.7	18.3	57.5	35.3	1.2	21.1	33.7
TAGCN[4] (Ours)	42.5	73.9	53.9	23.9	64.6	42.7	1.4	21.3	40.5

Table 3. Ablation study for various convolutional methods with same architecture on the UAVid dataset, with 10 layers and 256 channels.

such as BANet[21] and BiSeNet[24], as shown in Table 1. More experiments are necessary to improve the overall performance and comparison for other semantic segmentation methods such as HRNet[19] and ShelfNet[28].

5. Conclusion

Superpixel GCN can be applied for semantic segmentation successfully. However, its performance couldn't reach the state-of-the-art. We suppose that there might be two reasons for the limitation. At first, we adopt GCNs with simple structure, including only batch normalization and ReLU, without any functional module and blocks. Secondly, there are only few nodes for some classes. We are looking forward to developing improved structure to solve these problems in the future. Also, superpixel procedure is used as data preprocessing in the proposed method. We are planning to insert the superpixel training procedure to the entire architecture, achieving end-to-end Superpixel-based GCN model.

Although the performance of proposed approach has some limitations, our achievement is very meaningful in this field. It is a novel methodology expand the concept of Graph-based machine learning into semantic segmentation tasks. There are numerous possibilities to be improved with various auxiliary function. Both research area, Superpixel and GCNs, will contribute to Superpixel-based GCN, while state-of-the-art methods in each field can be adopted easily for our approach. In addition, the proposed method can be used on various dataset and field, including medical image, remote sensing, autonomous driving, and manufacturing.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 2, 3, 5

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016. 2, 5, 6

[4] Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017. 2, 3, 5, 6

[5] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017. 3, 5, 6

[6] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020. 5

[7] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018. 2

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2, 3

[9] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank, 2019. 2, 5, 6

[10] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning, 2018. 4

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Pro-*

ceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015. 1, 2

[12] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS journal of photogrammetry and remote sensing*, 165:108–119, 2020. 5

[13] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016. 3

[14] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 1

[15] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020. 4

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 4, 5

[17] Kiran K. Thekumparampil, Sewoong Oh, Chong Wang, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning, 2018. 3

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3

[19] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 6

[20] Libo Wang, Shenghui Fang, Ce Zhang, Rui Li, Chenxi Duan, Xiaoliang Meng, and Peter M Atkinson. Sanet: Scale-aware neural network for semantic labelling of multiple spatial resolution aerial images. *arXiv preprint arXiv:2103.07935*, 2021. 1

[21] Libo Wang, Rui Li, Dongzhi Wang, Chenxi Duan, Teng Wang, and Xiaoliang Meng. Transformer meets convolution: A bilateral awareness network for semantic segmentation of very fine resolution urban scene images. *Remote Sensing*, 13(16):3065, 2021. 4, 6

[22] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr. au2, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks, 2019. 2, 5, 6

[23] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13964–13973, 2020. 2, 3, 5

[24] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 4, 6

[25] Ce Zhang, Peter M Atkinson, Charles George, Zhaofei Wen, Mauricio Diazgranados, and France Gerard. Identifying and mapping individual plants in a highly diverse high-elevation ecosystem using uav imagery and deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:280–291, 2020. 1

[26] Ce Zhang, Paula A Harrison, Xin Pan, Huapeng Li, Isabel Sargent, and Peter M Atkinson. Scale sequence joint deep learning (ss-jdl) for land use and land cover classification. *Remote Sensing of Environment*, 237:111593, 2020. 1

[27] Lei Zhu, Qi She, Bin Zhang, Yanye Lu, Zhilin Lu, Duo Li, and Jie Hu. Learning the superpixel in a non-iterative and lifelong manner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1225–1234, 2021. 2

[28] Juntang Zhuang, Junlin Yang, Lin Gu, and Nicha Dvornek. Shelfnet for fast semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 6