https://doi.org/10.7840/kics.2023.48.10.1304

Fusion of CNN and ORB Detection with Graph Convolutional Network for Enhancing Feature Discrimination

Ryan Febriansyah◆, Soo Young Shin°

ABSTRACT

A novel image classification scheme called ORB-GCN, which combines convolutional neural networks (CNNs) with features from Oriented and Rotated BRIEF (ORB) detection for a graph convolutional network (GCN). CNN models often encounter challenges in differentiating between similar features, leading to reduced interpretability and lower accuracy. Enhancing feature discrimination in local and global information is the goal of the ORB algorithm fusion for graph construction in GCN. By training CNN and ORB-GCN simultaneously and performing end-to-end classification, the proposed method effectively improves the discriminative ability of features compared to state-of-the-art methods. According to experiments on the MIT Indoor CVPR09 and Intel Image Scene datasets, the proposed ORB-GCN approach has the best accuracy.

Key Words : Oriented and Rotated BRIEF, Discriminative feature, Graph convolutional network, Graph construction

Ⅰ. Introduction

Computer-vision tasks such as object identification, location detection, and image categorization have gained significant importance owing to the widespread availability of mobile devices equipped with high-quality cameras[1]. Convolutional neural networks (CNNs) have emerged as the preferred choice for image classification because of their exceptional performance[2]. In a typical CNN model, feature extraction captures characteristics such as colors, textures, and edges from the images. Subsequently, multiple convolutional layers process these features by combining and abstracting them to generate sophisticated representations[3]. Despite the success of traditional CNN models, they have certain limitations that affect their interpretability and accuracy, particularly when learning discriminative features[4,5]. To address this challenge, researchers have explored techniques to boost model capacity while maintaining acceptable training behavior. However, developing models with an unlimited number of parameters is computationally expensive. To mitigate this, methods such as depthwise separable convolutions (DSCs) and blueprint separable convolutions (BSCs) have been proposed to improve the model efficiency and performance[6-8]. These techniques leverage redundant filter weights to minimize computational costs. Whereas DSCs perform spatial and channel-wise filtering separately, BSCs employ a hierarchical structure to deepen the

※ Enter acknowledgment here: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2023-2020-0-01612) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation) and by Priority Research Centers Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003).

◆ First Author : Kumoh National Institute of Technology, Department of Electronic Engineering, ryaan354@kumoh.ac.kr, 학생회원

° Corresponding Author : Kumoh National Institute of Technology, Department of IT Convergence Engineering, wdragon@kumoh.ac.kr, 종신회원

논문번호 : 202305-096-C-SE, Received May 4, 2023; Revised July 3, 2023; Accepted July 18, 2023

model and reduce the parameter counts. Moreover, current CNNs primarily focus on global visual features and often ignore spatial information or local location features[9]. To overcome this limitation, researchers have turned to graph neural networks, which effectively capture spatial information. Notably, graph convolutional networks (GCNs) have demonstrated the ability to learn spatial relations[10]. However, incorporating graph structures and GCNs into image processing models remains challenging.

A novel image classification scheme called ORB-GCN combines a CNN with features from Oriented and Rotated BRIEF (ORB) detection and a GCN. Keypoints with high repeatability and distinctiveness can be detected using the ORB feature-detection algorithm. By integrating the ORB feature algorithm for graph construction in a GCN, the proposed approach aims to enhance the discriminative ability of features in both local and global information through the joint learning of unique visual information and spatial location relationships. The differences between the proposed and existing methods are listed in Table 1. The effectiveness of the proposed scheme was demonstrated through experiments on the MITindoorCVPR09 and Intel Image Scene datasets, where the highest accuracy was achieved by ORB-GCN.

Table 1. Related studies

Related studies	Scope		GCN batch Normalization	Graph construction methods
	Global feature	Local feature		
[19, 20, 21,22]	-	✓	-	SLIC Superpixel
[23, 24]	-	✓	-	K-nn
[25]	-	✓	✓	Co-occurance
[26]	-	✓	✓	RBF
[27]	✓	✓	-	R-CNN
This study	✓	✓	✓	ORB

Ⅱ. System Model

The proposed scheme includes four primary modules, as depicted in Fig. 1: CNN for feature extraction, ORB-based graph construction, GCN, and feature fusion. First, Resnet18 is utilized in the Feature Extraction module to extract global features from the input image. In the ORB-based graph construction module, differentiable mapping from the pixel representation of the image to the ORB feature space is employed to construct a graph and update its structure to model the relationships between unique contents in the image. The constructed graph is then fed into the GCN layer. Finally, the Feature Fusion module combines the global features from the CNN and local features from the GCN before they are sent to the classifier for prediction. The parameters for this approach were optimized using a cross-entropy loss function to minimize the difference between the predicted and ground-truth labels of the training samples. The objective of the proposed method is to improve the discriminatory ability of features in image classification. Each of these four modules is discussed in detail below.

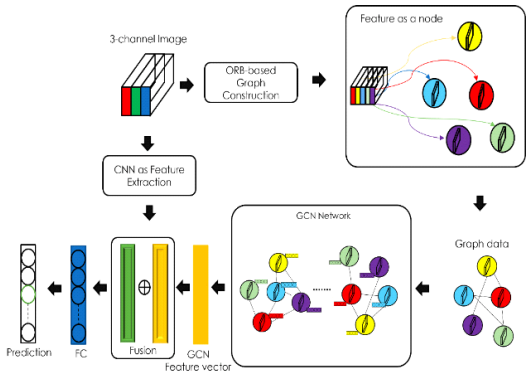

Fig. 1. Overall ORB-GCN system model

2.1 CNN as Feature Extraction

The CNN was the first module for feature extraction. CNNs are widely used in computer-vision tasks and are effective in learning powerful discriminative features from images. For feature extraction, the input image is passed through a series of convolutional layers that detect specific patterns and shapes in the image. These patterns were combined to create a set of features representing the most important parts of an image. To address the limitations of fully connected layers in feature

extraction, the proposed model uses convolutional layers that can handle inputs of varying sizes and detect features in different parts of an image. As shown in Fig. 2, a modified Resnet18 model was used for global-based feature extraction. Resnet18 consists of convolutional layers, followed by residual block layers. The convolutional layers detect specific features in the image, such as edges or textures, which help capture the most important information while reducing the computational complexity of the network.

We assumed an input image with dimensions $W \times H \times C$, where W and H represent the width and height of the image, respectively, and C represents the number of channels. Assume a filter kernel called F with dimensions of $K \times K \times C_{in}$, where K represents the size of the filter and C_{in} represents the number of input channels. Assume a stride value S that defines how the filter moves across the input image and an amount of zero padding P that determines the padding added to the edges of the input image. Using these values, the output of the convolutional layer can be computed as follows:

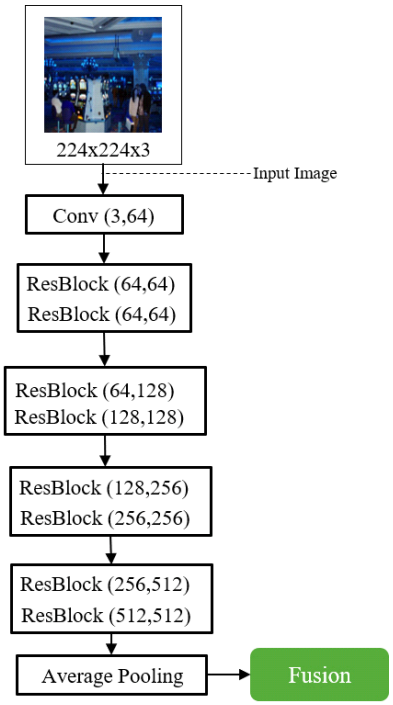

Fig. 2. Modified Resnet18 for Feature Extraction

$$O = \sigma(I * F + b) \tag{1}$$

The output, denoted as O, is a feature map with dimensions $W_{out} \times H_{out} \times C_{out}$. This output is generated through a convolution operation represented by the asterisk symbol *, which incorporates a vector of bias terms, denoted as b, and an activation function represented by the Greek letter sigma σ.

$$W_{out} = \left[\frac{W - K + 2P}{S}\right] + 1 \tag{2}$$

$$H_{out} = \left[\frac{H - K + 2P}{S}\right] + 1 \tag{3}$$

The value of C_{out} corresponds to the number of filters used. Typically, a pooling layer is added after the convolution layer to reduce the spatial dimensions of the output feature maps. The most frequently used pooling layer is the max-pooling layer, which selects the maximum value within each pooling region. Assume I represents the input feature map, which has dimensions $W \times H \times C_{in}$. K represents the pooling region size and S represents the stride used to move the pooling region across the input feature map. The output of the max-pooling layer is calculated as follows:

$$O_{i,j,k} = \max_{x=0}^{K-1} \max_{y=0}^{K-1} I_{(i+x)\times S,(j+y)\times S,k} \tag{4}$$

The output feature map is denoted by O after the pooling layer is applied. Although the output feature map maintains the same number of channels as the input feature map, its spatial dimensions are reduced because of the pooling operation. The output of the pooling layer is then sent to subsequent modules for further processing. The CNN feature extraction model has two modifications.

First, the original image is directly forwarded to an ORB-based graph construction process. This was because the original images were used to retain as much detailed spatial information as possible. Therefore, this approach is passively used to preserve spatial information. Second, the last fully connected layers were removed, and an average-pooling layer was introduced. The last layer of the CNN model is

typically utilized because deeper layers can extract more complex and abstract features from the input data by combining lower-level features such as edges and corners. The average-pooling layer reduces the spatial dimensions of the feature map by calculating the mean height and width. The equations for average-pooling are expressed as follows:

$$g_{c,i} = \frac{\sum_{w=1}^{W}\sum_{h=1}^{H} x_{w,h,c,i}}{W \times H} \quad (5)$$

The input feature map, denoted by x, has width w, height h, and number of filters c. The global average-pooling layer reduces a 3-D tensor with dimensions of 7 × 7 × 512 to a 1-D feature vector with dimensions of 512.

2.2 ORB-based Graph Construction

Oriented and rotated BRIEF (ORB) is a feature-detection and description algorithm used in computer vision. This is a modification of the widely used Features from Accelerated Segment Test (FAST) algorithm, which is a corner detection algorithm that quickly detects points of interest in an image. The ORB was built by adding the orientation and descriptor computations. Specifically, ORB detects keypoints in an image and computes their orientation using the intensity centroid method. Then, Binary Robust Independent Elementary Features (BRIEF) are used to generate binary descriptors for the detected keypoints.

In Fig. 3, the ORB is used for local feature extraction and graph construction. Specifically, the ORB is used to detect keypoints and compute the binary descriptors for each pixel in the image. The binary descriptors for each pixel are then used to construct a graph, where each node represents a pixel, and the edges represent the similarity between the descriptors of the connected pixels. The graph is then processed using a GCN to learn spatial information relations and extract features that capture the dependencies among the unique contents in the image. The GCN uses a graph structure to propagate information between nodes and learn the relationships between them, which enhances the discriminative ability of features.

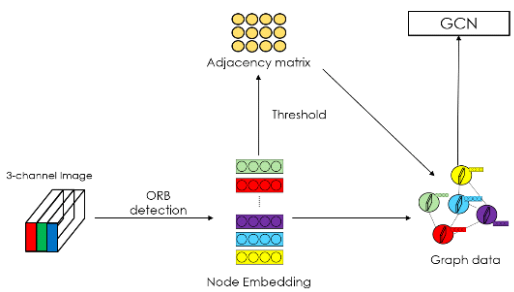

Fig. 3. Proposed ORB-based Graph Construction.

2.3 Graph Convolutional Network

The GCN utilizes adjacency matrix A to depict the relationships between nodes in the graph and propagate information between neighboring nodes during the graph convolution process. Each layer in the GCN is a sequence of graph convolutions that update the node features by aggregating the features of neighboring nodes via the adjacency matrix. This approach aims to learn representations that capture the structural information of a graph. The graph convolution operation involves applying the convolution operation to the graph using a weight matrix $W^{(l)} \in R^{F\times F}$, where F represents the number of input features. The resulting feature vector is updated for each layer of the network, and the process is represented by the following equation:

$$h^{(l+1)} = \sigma\left(\tilde{A}h^{(l)}W^{(l)}\right) \quad (6)$$

Here, $h^{(l)}$ represents the feature vector in layer l and $\tilde{A}$ is the normalized adjacency matrix. The σ function represents the activation function used to introduce non-linearity into the model. In the proposed ORB-GCN model, a GCN network was used to capture the local relationships between the nodes in the graph. The resulting feature vector is fused with the global feature vector obtained from the CNN module. This fusion process combines local and global features to provide a more complete representation of the input image, which can be used for classification.

2.4 Feature Fusion

In this section, we describe the enhancement of the feature discrimination ability for both global and local information by continuously training the CNN and ORB-GCN. This method is used in computer-vision tasks to record the global and local features of an image. Combining a CNN with an ORB-GCN is advantageous because it allows the model to utilize the capabilities of both techniques. The final classifier receives a combined set of feature descriptors retrieved from the CNN and ORB-GCN. The feature fusion procedure is described as follows.

$$H_{\mathrm{Fsn}} = W^{(f)} \cdot [g_i \ , \alpha h_i] \tag{7}$$

The features extracted by the two models, CNN and ORB-GCN, are represented by g_i and h_i, respectively. The weight matrix of the final fully connected layer that can be trained is denoted as $W^{(f)}$. The symbol $[\diamond,\diamond]$ is used to indicate the concatenation operation, and α is a variable that controls the ratio of fusion. During the training process, the parameters in the model were optimized using the cross-entropy loss function, as represented by the following formula:

$$L = -\sum_i \left(y_i \log\left(\sigma(H_{\mathrm{Fsn}})\right) + (1-y_i)\log\left(1-\sigma(H_{\mathrm{Fsn}})\right)\right) \tag{8}$$

The ground-truth label of class i is denoted as y_i in the formula. During the testing phase, a feature descriptor is used to predict the scene category.

Ⅲ. Experiments and Analysis

3.1 Data Preparation

MITindoorCVPR09. consists of 15,620 indoor scene images belonging to 67 categories, including bookstores, garages, gyms, libraries, restaurants, and offices. These images were annotated with 600 × 400 pixels and captured under diverse lighting conditions from different locations, making them suitable for various computer-vision tasks such as object recognition and scene classification.

Intel Image Scene, an outdoor image dataset from the Intel Corporation, was used to set a new standard for image classification assignments. The dataset consisted of 14,034 training images and 3,000 validation images divided into six categories: buildings, forests, glaciers, mountains, seas, and streets. The images in this dataset have a default resolution of 150 × 150 pixels.

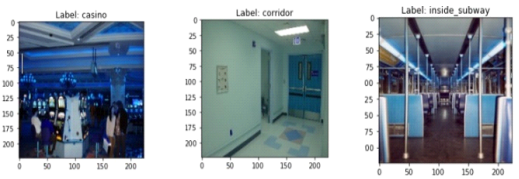

Fig. 4. MITIndoorCVPR09 dataset.

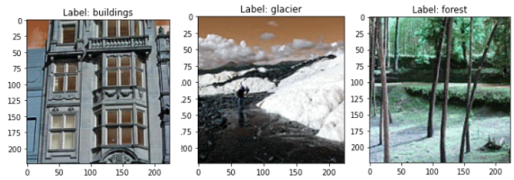

Fig. 5. Intel Image Scene dataset.

3.2 Experiment Setup

To implement the proposed ORB-GCN, the PyTorch framework was used in combination with the OpenCV and DGL graph libraries. An AMD Ryzen 7 2700X 16-Core CPU with an NVIDIA GeForce GTX 1070Ti GPU was used to accelerate the experiments. In fine-tuning the model on the image scene dataset, the Resnet18 model was used, and the initial input images for the model were 224 × 224 pixels. A stochastic gradient descent optimizer with an initial learning rate of 0.00001 was employed for optimization. The GCN layer was activated using the LeakyReLU activation function.

3.3 ORB Detection Result

The ORB detection algorithm is proficient in detecting corners in both datasets, precisely recognizing points of interest in images of indoor settings, such as corridors, bedrooms, and living spaces, as well as in outdoor environments, such as forests, beaches, and mountains. Fig. 6 illustrates examples of the ORB detection results for the MITindoorCVPR09 dataset. In the MIT indoor dataset, the ORB detection algorithm accurately

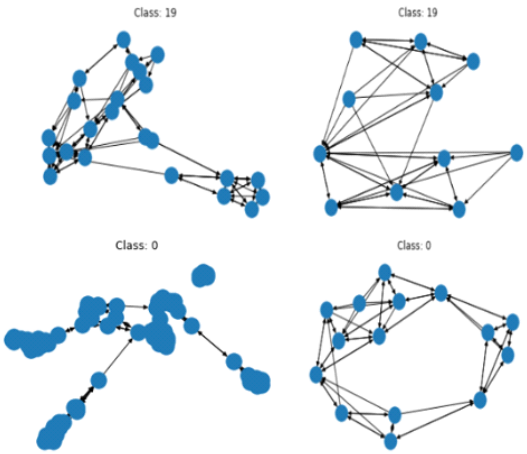

Fig. 6. ORB Detection result

detected points of interest in different settings, including those with intricate furniture and embellishments. In summary, the ORB corner detection algorithm demonstrated satisfactory performance, indicating its ability to identify corners precisely and effectively in digital images of varied indoor settings with complex lighting conditions and backgrounds.

3.4 Graph Construction Result

The spatial relationships between points of interest or nodes in an image are accurately represented by the resulting graph, which provides a useful depiction of the image structure. The effectiveness of the graph construction algorithm was evaluated on both datasets using the ORB detection algorithm and k-nearest neighbors. The results indicate that the method can construct precise graphs for a variety of images, even those with complex backgrounds and lighting conditions. Examples of the graph construction outcomes are illustrated in Fig. 7 for both datasets, where the nodes and edges precisely indicate the spatial connections between the interest points in the images. In summary, the number of nodes in the constructed graph was influenced by the threshold value of the ORB detection algorithm. When the threshold value was increased, the number of nodes decreased for both datasets. For example, in the MITIndoorCVPR09 dataset, raising the threshold value from 60 to 100 resulted in a decrease in the

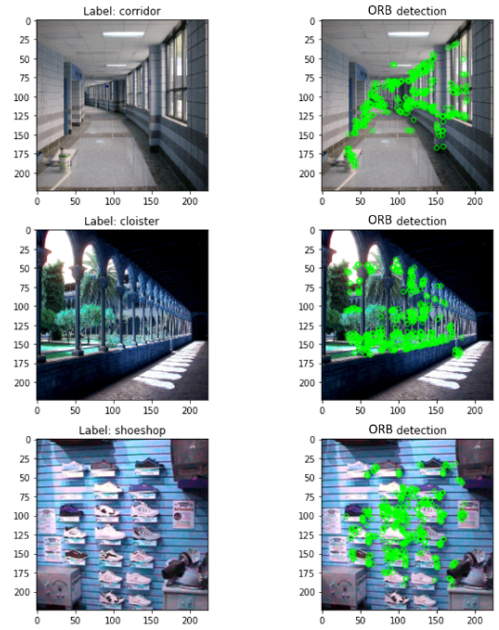

Fig. 7. Example result of graph construction on both datasets.

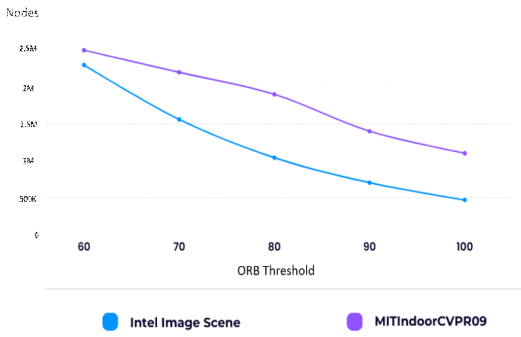

Fig. 8. Number of nodes according to ORB Threshold Parameter Evaluation

total number of nodes from 2.5 million to 1.1 million, as depicted in Fig. 8. Similarly, increasing the threshold value from 60 to 100 in the Intel Image Scene dataset caused a reduction in the number of nodes from 2.3 million to 465k.

3.5 Experiment Results

The proposed method is demonstrated through a comparative evaluation of state-of-the-art classification methods on the MITindoorCVPR09 and Intel Image Scene datasets. The performance of the proposed method, which incorporates a GCN, was

compared with that of a baseline CNN method that does not utilize a GCN, as well as several deep learning-based methods. The performance of the selected methods was compared in terms of image classification. The comparative evaluation results are presented in Tables 2 and 3, where the highest overall accuracies of 95.5% and 97.3% for both datasets were achieved using the proposed method. By incorporating the GCN into the image classification method, we observed a significant improvement in classification accuracy compared with the CNN baseline method by 17.5% on MITIndoorCVPR09 and 16.3% on the Intel Image Scene dataset. This improvement clearly demonstrates the usefulness of the GCN in capturing spatial dependencies among objects in the dataset. The GCN model effectively leverages the graph structure to propagate information and exploit the relationships between data elements, leading to more discriminative features and enhanced classification results.

Table 2. Overall accuracy of the proposed method and the comparison methods on MITIndoorCVPR09.

Methods	Parameter	Accuracy (%)
GCN	77M	56
Adi-Red [11]	-	73.59
CNN	11M	78
VS-CNN [12]	86M	80.37
PatchNet [13]	40M	86.2
Semantic-Aware [14]	47M	87.1
FOSNet [15]	-	90.3
ORB-GCN	**165M**	**95.5**

Table 3. Overall accuracy of the proposed method and the comparison methods on Intel Image Scene.

Methods	Parameter	Accuracy (%)
GCN	77M	59
CNN	11M	81
WiseNet+DepthConv [16]	23M	87
Inception-V3 [17]	47M	91.81
RepConv [18]	160K	94.58
ORB-GCN	**145M**	**97.3**

Ⅳ. Conclusion

This study introduces a new image-based classification approach that combines a CNN with an ORB detection algorithm for GCN, called ORB-GCN. Integrating the ORB feature algorithm for graph construction in the GCN enhances the feature discrimination ability for both global and local information by simultaneously training the CNN and ORB-GCN. ORB-GCN was effective in capturing the overall visual characteristics of the image, as well as the spatial relationship between distinct features generated by ORB detection, which reduced visual confusion and improved the discriminative ability of the features. In future work, it will be possible to utilize ORB-GCNs in the field of robotics, specifically in applications such as simultaneous visual localization and mapping (VSLAM). Integrating the graph structure model into VSLAM algorithms can enhance the precision and efficiency of mapping and localization for robots, such as drones or mobile robots.

References

[1] L. Liu, "Deep learning for generic object detection: A survey," *Int. J. Computer Vision*, vol. 128, no. 2, pp. 261-318, Feb. 2020. (https://doi.org/10.48550/arXiv.1809.02165)

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in NIPS*, vol. 25, pp. 1097-1105, Dec. 2012. (https://doi.org/10.1145/3065386)

[3] R. Yamashita, et al., "Convolutional neural networks: An overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611-629, Aug. 2018. (https://doi.org/10.1148/radiol.2018180547)

[4] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Process.*, vol. 73, pp. 1-15, Jan. 2018. (https://doi.org/10.1016/j.dsp.2017.10.011)

[5] P. Linardatos, V. Papastefanopoulos, and S.

Kotsiantis, “Explainable AI: A review of machine learning interpretability methods,” *Entropy*, vol. 23, no. 1, p. 18, Jan. 2021. (https://doi.org/10.3390/e23010018)

[6] A. Khan, et al., “A survey of the recent architectures of deep convolutional neural networks,” *Artificial Intell. Rev.*, vol. 53, no. 8, pp. 5455-5516, Aug. 2020. (https://doi.org/10.1007/s10462-020-09825-6)

[7] A. G. Howard, et al., “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv: 1704.04861*, Apr. 2017. (https://doi.org/10.48550/arXiv.1704.04861)

[8] D. Haase and M. Amthor, “Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets,” in *Proc. IEEE/CVF Conf. CVPR*, pp. 1-10, Jun. 2020. (https://doi.org/10.48550/arXiv.2003.13549)

[9] J. Zhou, et al., “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57-81, Jan. 2020. (https://doi.org/10.48550/arXiv.1812.08434)

[10] S. Baroud, et al., “A brief review of graph convolutional neural network-based learning for classifying remote sensing images,” *Procedia Comput. Sci.*, vol. 191, pp. 349-354, Jan. 2021. (https://doi.org/10.1016/j.procs.2021.07.047)

[11] Z. Zhao and M. Larson, “From volcano to toyshop: Adaptive discriminative region discovery for scene recognition,” in *Proc. 26th ACM Int. Conf. Multimedia (ACM MM)*, pp. 1760-1768, 2018. (https://doi.org/10.1145/3240508.3240698)

[12] J. Shi, H. Zhu, S. Yu, W. Wu, and H. Shi, “Scene categorization model using deep visually sensitive features,” *IEEE Access*, 2019. (https://doi.org/10.1109/ACCESS.2019.2908448)

[13] Z. Wang, L. Wang, Y. Wang, B. Zhang, and Y. Qiao, “Weakly supervised patchnets: Describing and aggregating local patches for scene recognition,” *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2028-2041, 2017. (https://doi.org/10.1109/TIP.2017.2666739)

[14] H. Seong, J. Hyun, and E. Kim, “FOSNet: An end-to-end trainable deep neural network for scene recognition,” *IEEE Access*, vol. 8, pp. 82066-82077, 2020. (https://doi.org/10.1109/ACCESS.2020.2989863)

[15] A. López-C, J. Martínez-G, A. García O, J. J. Pantrigo, and F. J. Madrid, “Semantic-aware scene recognition,” *Pattern Recognition*, vol. 102, no. 107256, Nov. 2020. (https://doi.org/10.1016/j.patcog.2020.107256)

[16] M. Rahimzadeh, et al., “Wise-SrNet: a novel architecture for enhancing image classification by learning spatial resolution of feature maps,” *arXiv preprint arXiv:2104.12294*, 2021. (https://doi.org/10.48550/arXiv.2104.12294)

[17] X. Wu, “An xception based convolutional neural network for scene image classification with transfer learning,” *2nd Int. Conf. ITCA, IEEE*, 2020. (https://doi.org/10.1109/ITCA52113.2020.00063)

[18] M. Soudy, Y. Afify, and N. Badr, “RepConv: A novel architecture for image scene classification on Intel scenes dataset,” *Int. J. Intell. Comput. and Inf. Sci.*, pp. 63-73, 2022. (https://doi.org/10.21608/ijicis.2022.118834.1163)

[19] L. V. Linh and C. H. Youn, “Dynamic graph neural network for superpixel image classification,” *2021 Int. Conf. ICTC*, pp. 1095-1099, 2021. (https://doi.org/10.1109/ICTC52510.2021.9621101)

[20] Y. Li, R. Chen, Y. Zhang, and H. Li,” A CNN-GCN framework for multilabel aerial image scene classification,” *IGARSS 2020*, pp. 1353-1356, 2020. (https://doi.org/10.1109/IGARSS39084.2020.9323487)

[21] H-S. Lin, J.-J. Ding, and J.-Y. Huang, “Graph saliency network: Using graph convolution

network on saliency detection," *2020 IEEE APCCAS*, pp. 177-180, 2020.
(https://doi.org/10.1109/APCCAS50809.2020.9301708)

[22] H. Zhang, J. Zou, and L. Zhang," EMS-GCN: An end-to-end mixhop superpixel-based graph convolutional network for hyperspectral image classification," in *IEEE Trans. Geosci. and Remote Sensing*, vol. 60, no. 5526116, pp. 1-16, 2022.
(https://doi.org/10.1109/TGRS.2022.3163326)

[23] Z. Gao, Z. Lu, J. Wang, S. Ying, and J. Shi, "A convolutional neural network and graph convolutional network based framework for classification of breast histopathological images," in *IEEE J. Biomed. and Health Informatics*, vol. 26, no. 7, pp. 3163-3173, Jul. 2022.
(https://doi.org/10.1109/jbhi.2022.3153671)

[24] A. Elmoogy, et al., "Pose-GNN: Camera pose estimation system using graph neural networks," *arXiv preprint arXiv:2103.09435*, 2021.
(https://doi.org/10.48550/arXiv.2103.09435)

[25] Z. -M. Chen, X.-S. Wei, P. Wang, and Y. Guo," Multi-label image recognition with graph convolutional networks," *2019 IEEE/CVF Conf. CVPR*, pp. 5172-5181, 2019.
(https://doi.org/10.48550/arXiv.1904.03582)

[26] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot," Graph convolutional networks for hyperspectral image classification," in *IEEE Trans. Geosci. and Remote Sensing*, vol. 59, no. 7, pp. 5966-5978, Jul. 2021.
(https://doi.org/10.1109/TGRS.2020.3015157)

[27] J. Liang, Y. Deng. and D. Zeng," A deep neural network combined CNN and GCN for remote sensing scene classification," in *IEEE J. Sel. Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4325-4338, 2020.
(https://doi.org/10.1109/JSTARS.2020.3011333)

Ryan Febriansyah

Feb. 2020 : B.S. degree, Telkom University
Feb. 2023 : M.S. degree, Kumoh National Institute of Technology
Kumoh National Institute of Technology
<Research Interests> Robotics, unmanned vehicle, computer vision and pattern recognition
[ORCID:0009-0004-4216-4211]

Soo Young Shin

Feb. 1999 : B.S. degree, Seoul National University
Feb. 2001 : M.S. degree, Seoul National University
Feb. 2006 : Ph.D, Seoul National University
Feb. 2010~current : Professor, Kumoh National Institute of Technology
<Research Interests> 5G/6G wireless communications, Internet of things, drone applications
[ORCID:0000-0002-2526-239]