A CNN-GCN FRAMEWORK FOR MULTI-LABEL AERIAL IMAGE SCENE CLASSIFICATION

Yansheng Li [1], Ruixian Chen [1], Yongjun Zhang [1], and Hang Li [2]

[1] School of Remote Sensing and Information Engineering, Wuhan University, China
[2] Beijing Aerospace System Engineering Research Institute, China

IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium | 978-1-7281-6374-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/IGARSS39084.2020.9323487

ABSTRACT

As one of the fundamental tasks in aerial image understanding, multi-label aerial image scene classification attracts increasing research interest. In general, the semantic category of a scene is reflected by the object information and the topological relations among objects. Most of existing deep learning-based aerial image scene classification methods (e.g., convolutional neural network (CNN)) classify the image scene by perceiving object information, while how to learn spatial relationships from image scene is still a challenging problem. In literature, graph convolutional network (GCN) has been successfully used for learning spatial characteristics of topological data, but it is rarely adopted in aerial image scene classification. To simultaneously mine both the object visual information and spatial relationships among multiple objects, this paper proposes a novel framework combining CNN and GCN to address multi-label aerial image scene classification. Extensive experimental results on two public datasets show that our proposed method can achieve better performance than the state-of-the-art methods.

Index Terms—Graph convolutional network (GCN), convolutional neural network (CNN), multi-label aerial image classification.

1. INTRODUCTION

Aerial images scene classification takes the image scene as basic interpretation unit and aims at assigning semantic categories to the image scene (i.e., one image block) according to its visual and contextual content [1, 2]. Due to its wide applications in object detection [3], image retrieval [4, 5], and so forth, aerial image scene classification attracts increasing research interest. Compared with single-label aerial image scene classification, multi-label aerial image scene classification is a more realistic and complex task, which assigns multiple semantic labels to represent scenes. As a whole, multi-label aerial image scene classification is still an open problem and deserves much more exploration.

Benefiting from the hierarchy abstract ability of deep learning, convolutional neural network (CNN) has been widely used for multi-label aerial image scene classification and shows significant improvement. In [6], the authors adopt CNN and design multi-labeling layer to address multi-label aerial image scene classification. To fully exploit the co-occurrence dependencies among multiple labels, the authors in [7] propose a class attention-based convolutional and bidirectional LSTM network (CA-CNN-BiLSTM) and the authors in [8] propose an attention-aware label relational reasoning network (AL-RN-CNN). In [9], the authors propose a CNN-RNN framework and adopt multi-attention mechanism for classification. However, these CNN-based methods only perceive the objects in the scene but ignore the spatial distribution relationships among many separate objects in the image scene.

In fact, when judging the categories of an aerial image scene, people not only recognize what objects are in the scene, but also consider the spatial relationships of objects. Motivated by the fact that CNN is effective in representing the visual content of local regions in the image scene and the spatial relationships of topological data can be effectively learned by graph convolutional network (GCN) [10], we propose an aerial image scene classification framework by combining CNN and GCN. Specifically, we encode the spatial structure of image scene by constructing region adjacency graph for each image using unsupervised segmentation algorithm. The CNN is used to extract deep features vector of regions to semantically represent the visual content of regions. And the GCN is used to synthesize high-level visual features and spatial relationships of local regions and learn abstract representation for classification.

We use the proposed CNN-GCN framework to comprehensively address multi-label aerial image scene classification task. The extensive experimental results on the UCM and AID multi-label datasets show that our proposed method can achieve better classification performance compared with the existing methods.

2. METHODOLOGY

To effectively represent the spatial structure of aerial image scene, we construct region adjacency graph for each image scene where the image is over-segmented into non-overlapping regions and the feature representation of each region is semantically represented by CNN. Taking the

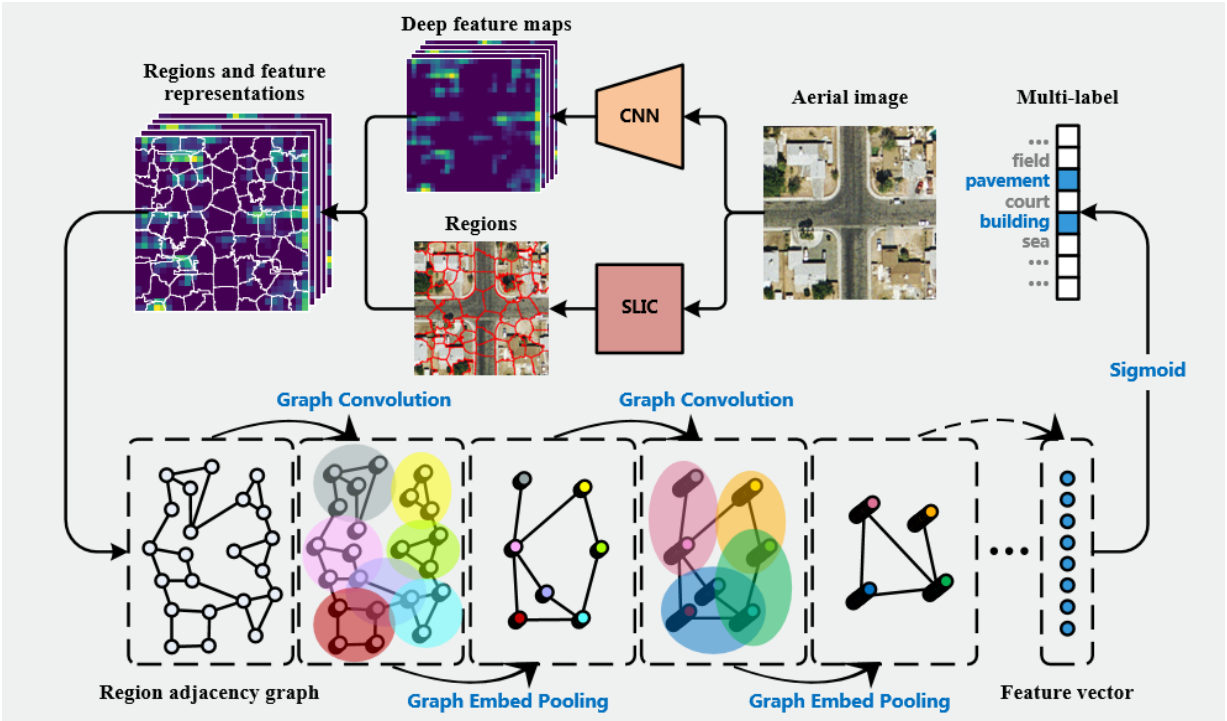

Figure 1. An overview of the proposed CNN-GCN framework.

region adjacency graph as the input, a GCN model is trained to mine the spatial relationship among regions and complete multi-label classification.

2.1. Constructing CNN-based region adjacency graph

The region adjacency graph can encode the image as a graph structure consisting of adjacency matrix $A \in \mathbb{R}^{N\times N}$ and vertex feature matrix $X \in \mathbb{R}^{N\times D}$, where N is the number of vertices and D is the number of features.

We use SLIC super-pixel algorithm [11] to segment the image and get multiple non-overlapping regions as vertices of graph. Note that the region is made up of homogeneous pixels, so it can be assumed that it is an approximate representation of local objects.

Similar to [10], we quantify the spatial relationship between regions to obtain the adjacency matrix A. When regions i and j have a common boundary, the adjacency weight a_{ij} is calculated by Eq. (1),

$$a_{ij} = \beta_1 \|c_i - c_j\|_2 + \beta_2 |o_i - o_j| \quad (1)$$

where c_i represents the centroid pixel of region and o_i represents the orientation angle. β_1 and β_2 are empirically set in 0.8 and 0.2 to assign weights for distance and direction relations between adjacent regions.

We use the high-level visual features as the vertex features to represent visual elements in the scene. Particularly, we feed images into the pre-trained CNN and obtain a series of feature maps from convolutional layers. It is worth mentioning that the CNN can also be trained from scratch using the addressed multi-label dataset. Then we combine the feature maps and segmentation results by up-sampling the features to the size of original image. According to the region boundary of the segmentation, we calculate the max value of each feature map slice as the corresponding vertex feature of the region. Therefore, for each region, we can get multidimensional features from multiple channels of feature maps. In this way, we can get the vertex feature matrix X of graph.

2.2. Learning graph convolutional network

To explore the spatial distribution relationships of different regions, we adopt the spatial GCN model proposed in [12], which can take A and X of graph as input directly. The convolution, pooling and fully connected operations in GCN are briefly introduced as follow.

2.2.1. Graph convolution operation

Generally, a convolutional layer uses a parameter-shared convolution kernel as a filter to extract features by calculating the weighted sum of adjacent pixels. In the case of graphs, the receptive field of convolution needs to be provided by A. Taking the first order neighborhoods into account, we use a simplified spatial-domain convolution filter $F \in \mathbb{R}^{N\times N\times D}$, shown in Eq. (2),

$$F = w_0 E + w_1 A \quad (2)$$

where E is the 0-th order adjacency matrices, and w_0 and w_1 are learnable weights. Graph convolution is the matrix multiplication between F and X, and the graph convolution operation is given by Eq. (3),

$$X_{out} = \sum_{d=1}^{D} F^{(d)} X_{in}^{(d)} + b \quad (3)$$

where X_{in} indicates the input vertex feature, X_{out}

indicates the output and b is a bias. The size of X_{out} can be adjusted by adding another dimension to F. For example, we can set F in $\mathbb{R}^{N\times N\times D\times D'}$ and get $X_{out}\in\mathbb{R}^{N\times D'}$ by repeating the Eq. (3) D' times.

2.2.2. Graph embed pooling operation

Using pooling operation can reduce dimensions of the input and improve the computing performance. Since graphs are often heterogeneous structures, pooling should be done by embedding, which can map the input graph of any size to a fixed-size output [12]. Graph embed pooling is implemented through an embedded matrix $X_{emb}\in\mathbb{R}^{N\times N'}$, which can be learned as Eq. (4), where N' is the new number of vertices. And Eq. (5) shows a softmax operation to normalize X_{emb}.

$$X_{emb}^{(n')}=\sum_{d=1}^{D}F_{emb}^{(d,n')}X_{in}^{(d)}+b \quad (4)$$

$$X_{emb}^{*}=\sigma(X_{emb}) \quad (5)$$

Then the vertex features output X_{out} and the adjacency matrix output A_{out} can be calculated by Eq. (6) and Eq. (7), respectively.

$$X_{out}=X_{emb}^{*T}X_{in} \quad (6)$$

$$A_{out}=X_{emb}^{*T}A_{in}X_{emb}^{*} \quad (7)$$

Because embed pooling method is learnable, the output structure is an optimized result representing the reduced-dimension input structure. So, it has the potential to optimize the distribution of vertices.

2.2.3. Fully connected operation

After a series of operation of graph convolution and pooling, the vertex feature matrix is projected into the one-dimensional vector space by the fully connected layer, which is a case of graph embed pooling layer to produce graph with only one vertex. And we use sigmoid as the activation function of the end of the network. Eq. (8) shows the formula to transform the features vectors x_i into the classification probabilities of multi-hot label.

$$\sigma(x_i)=\frac{1}{1+\exp(-x_i)} \quad (8)$$

Furthermore, we use the binary cross-entropy as the loss function for multi-label classification, which can be computed by Eq. (9),

$$loss=-\sum_i\big(y_i\log(\sigma(x_i))+(1-y_i)\log(1-\sigma(x_i))\big) \quad (9)$$

where y_i indicates the ground truth label of class i.

Through backward propagation, parameters of the GCN can be optimized based on the gradient of $loss$, and we use GCN to learning abstract features and complete classification in an end-to-end manner.

3. EXPERIMENTAL RESULTS

In this section, data description and the details of experimental settings are presented at first. The experimental results and analysis are given after that.

3.1. Data description

We performed experiments on two public multi-label aerial image scene classification datasets, which are described below. UCM multi-label dataset, which contains 2100 aerial images with 0.3 m/pixel spatial resolution and 256×256 pixels image size, is labeled into 17 categories base on DLRSD dataset [13]. In addition, we use AID multi-label dataset [8], which contains 3000 aerial images from the AID dataset [14] and is assigned with 17 object labels. The spatial resolutions of images vary from 0.5 to 0.8 m/pixel, and the size of each image is 600×600 pixels.

3.2. Experimental settings

For experiment, the UCM and AID multi-label datasets are split into 80% for training and 20% for testing.

We use pre-trained VGG16 [15] to extract visual features and our model using the 28×28×512 feature maps output from the third convolutional layer of block4 in VGG16 as the visual features.

Our GCN architecture contains a graph convolution layer with 512 filters and a graph embed pooling layer outputting a 64-vertex graph. At the end of the GCN, we set up two fully connected layers with 256 outputs and 17 (number of classes) outputs, respectively.

Moreover, the dropout layer is set in the middle of each layer, and ReLU activation function and batch normalization are used for all but the last layer. We train the GCN with Adagrad optimizer. The learning rate is initially set as 0.01 and decayed during training process.

3.3. Comparison with the state-of-the-art methods

We compare our proposed CNN-GCN method with several recent methods, including the standard CNN method, CNN-RBFNN [6], CA-CNN-BiLSTM [7] and AL-RN-CNN [8]. For a fair comparison, all compared methods adopt the same VGG16 structure as the CNN backbone. We calculate Precision (P), Recall (R), F1-Score (F1) and F2-Score (F2) to evaluate the multi-label classification performance of each method. As the compared methods also adopt the same training/testing data, we take the reported evaluation results from their separate publications as the reference in this paper.

Table 1 shows the experimental results of our proposed CNN-GCN method and other methods on the UCM multi-label dataset. We can observe that our proposed method achieve the highest scores of Recall, F1-Score and F2-Score. In general, our proposed method achieves the best performance. In comparison with AL-RN-CNN, our method increases F1-Score and F2-Score by 0.11% and 0.73%, respectively. We can also observe that our methods with GCN have significant improvement compared with the methods only use CNN. Compared to the method of CNN, our method gains an improvement of 7.27% in F1-Score and 6.37% in F2-Score, which demonstrates that learning spatial relationships via GCN plays an important role in advancing classification performances.

Table 1. The performances of different methods on the UCM multi-label dataset (%).

Methods	P	R	F1	F2
CNN [15]	79.06	82.30	78.54	80.17
CNN-RBFNN [6]	78.18	83.91	78.80	81.14
CA-CNN-BiLSTM [7]	79.33	83.99	79.78	81.69
AL-RN-CNN [8]	**87.62**	86.41	85.70	85.81
Ours (CNN-GCN)	86.68	**87.59**	**85.81**	**86.54**

Table 2. The performances of different methods on the AID multi-label dataset (%).

Methods	P	R	F1	F2
CNN [15]	87.41	86.32	85.52	85.60
CNN-RBFNN [6]	84.56	87.85	84.58	85.99
CA-CNN-BiLSTM [7]	88.68	87.83	86.68	86.88
AL-RN-CNN [8]	**89.96**	89.27	88.09	88.31
Ours (CNN-GCN)	89.61	**89.55**	**88.26**	**88.68**

Table 2 shows the experimental results on the AID multi-label datasets. We can also observe that our proposed method has the best performance with the highest scores of Recall, F1-Score and F2-Score. Compared to standard CNN method, our method increases F1-Score and F2-Score by 2.74% and 3.08%, respectively. Compared to AL-RN-CNN, our method gains an improvement of 0.17% in F1-Score and 0.37% in F2-Score. The superior performances on both UCM and AID multi-label datasets can demonstrate the robustness and effectiveness of our method.

4. CONCLUSION

In this work, we propose a CNN-GCN framework for multi-label aerial image scene classification, where GCN is used to learn spatial relationships among regions in the region adjacency graph constructed based on CNN. The experimental results on two publicly open multi-label aerial image scene datasets show the robustness and effectiveness of our framework. In the future work, we will consider using a larger number of samples to explore the potential of our proposed framework.

5. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under grants 41971284 and 41601352; the China Postdoctoral Science Foundation under grants 2016M590716 and 2017T100581; the Hubei Provincial Natural Science Foundation of China under grant 2018CFB501.

6. REFERENCES

[1] Y. Li, and et al., "Unsupervised multilayer feature learning for satellite image scene classification," *IEEE Geoscience and RemoteSensing Letters*., vol. 13, no. 2, pp. 157-161, 2016.

[2] Y. Li, and et al., "Error-tolerant deep learning for remote sensing image scene classification," *IEEE Transactions on Cybernetics*., 2020.

[3] Y. Li, and et al., "Deep networks under scene-level supervision for multi-class geospatial object detection from remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*., vol. 146, pp. 182-196, 2018.

[4] Y. Li, and et al., "Large-scale remote sensing image retrieval by deep hashing neural networks," *IEEE Transactions on Geoscience and Remote Sensing*., vol. 56, no. 2, pp. 950-965, 2018.

[5] Y. Li, and et al., "Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*., vol. 56, no. 11, pp. 6521-6536, 2018.

[6] A. Zeggada, and et al., "A deep learning approach to UAV image multilabeling," *IEEE Geoscience and Remote Sensing Letters*., vol. 14, no. 5, pp. 694-698, 2017.

[7] Y. Hua, and et al., "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multilabel aerial image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*., vol. 149, pp. 188-199, 2019.

[8] Y. Hua, and et al., "Relation network for multi-label aerial image classification," *arXiv:1907.07274.*, 2019.

[9] G. Sumbul and et al., "A CNN-RNN framework with a novel patch-based multi-attention mechanism for multi-label image classification in remote sensing," *IEEE International Geoscience and Remote Sensing Symposium*., 2019.

[10] U. Chaudhuri, and et al., "Siamese graph convolutional network for content based remote sensing image retrieval," *Computer Vision and Image Understanding*., vol. 184, pp. 22-30, 2019.

[11] R. Achanta, and et al., "SLIC Superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*., vol. 34, no. 11, pp. 2274-2282, 2012.

[12] F. P. Such, and et al., "Robust spatial filtering with graph convolutional neural networks," *IEEE Journal of Selected Topics in Signal Processing*., vol. 11, no. 6, pp. 884-896, 2017.

[13] Z. F. Shao, and et al., "A benchmark dataset for performance evaluation of multi-Label remote sensing image retrieval," *Remote Sensing*., vol. 10, no. 6, pp. 964, 2018.

[14] G. S. Xia, and et al., "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*., vol. 55, no. 7, pp. 3965-3981, 2017.

[15] K. Simonyan, and et al., "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556.*, 2014.