Attention-Driven Graph Convolution Network for Remote Sensing Image Retrieval

Ushasi Chaudhuri, *Member, IEEE*, Biplab Banerjee, *Member, IEEE*, Avik Bhattacharya, *Senior Member, IEEE*, and Mihai Datcu, *Fellow, IEEE*

***Abstract*— Graph convolution networks (GCNs) are useful in remote sensing (RS) image retrieval. It is found to be effective because, in a graph representation, the relative geometrical interactions between different regions (or segments) are appropriately captured, along with their region-wise features in their region adjacency graphs. Also, the attention mechanism has often been applied to the nodes to highlight the essential features in each node. In this regard, a significant amount of high-frequency information is missed since each image segment is effectively summarized within a single node. To account for this and increase the learning capacity, we propose to attend over the edge/adjacency matrix to highlight the interactions among meaningful regions that contribute to supervised learning from images. We exploit this novel edge attention mechanism together with node attention to highlight essential image context by allowing more importance to the meaningful neighboring regions that highlight a relevant node. We implement the proposed context-attended GCN framework for image retrieval on the benchmarked UC-Merced and the PatternNet datasets. We observe a notable improvement in the results compared to the state of the art.**

***Index Terms*— Attention network, graph convolution networks (GCNs), image retrieval, remote sensing (RS), Siamese architecture.**

I. Introduction

DATA pruning and retrieval have received paramount interest in the remote sensing (RS) community due to the advancement in sensor technology and the accumulation of vast amounts of data. A conventional context-based image retrieval (CBIR) task uses a target query image as the context and uses it to prune the database to search for its nearest matches. Hence, it is essential that the designed framework provides an adequate representation of the extracted features of the images and is also efficient at the same time [1].

Convolutional neural networks (CNNs) mainly comprise data-driven feature extractors [1] which describe the global level features from the images. Such a global descriptor can often yield inferior results in RS as they do not include the local neighborhood constructs within the images. To this end, researchers have found Graph convolution networks (GCNs) [2] to be more effective in RS image retrieval [3]–[5]. It helps in highlighting the local scene constructs in addition to describing the global scenes. GCNs can be employed either in the spatial or spectral domain. They are usually applied in the spatial domain as it provides the notion of part learning and part interactions in objects. It works on a local neighborhood of nodes and learns from the properties of a node based on its local neighbors. Until recently, attention mechanism has received a lot of consideration in learning sequence-based tasks [6] by using CNNs and recurrent neural networks. It primarily helps the network to focus more on the most relevant part of the input information. RS scene images comprise multiple subclasses, which together constitute a single-labeled scene. Thus, using attention networks in conjugation with CNNs is often less effective as it captures the global scene, leaving out these subtle important local constructs. In addition, CNN's with attention requires many hyperparameters to be tuned, making the training task challenging. Despite this, Veličković *et al.* [6] applied attention mechanism on a graph-theoretic framework and performed attention within each subregions of the image. This helped to highlight each node's most essential features that contribute to the overall classification accuracy.

While node attention helps us highlight the vital node features, it is also essential to realize the contribution of the interaction of each neighborhood region in describing the overall scene. For instance, in an RS application, to describe an airport scene, the presence of airplanes near a runway would be more relevant than the possibility of finding a water body or a bush near the runway. In such a case, we would like to highlight the former adjacency more than the latter one. It is commonly referred to as a link prediction problem in network theory [7]. Although link prediction is frequently used in network theory, it has not been used in conjunction with GCNs for RS and computer vision (CV) applications to the best of our knowledge.

Inspired by several sources of knowledge reviewed above, we propose a *context-attended* GCN, wherein we attend over nodes and the edges. The distinct context embedding vector is a nonlinear combination of all node-attended features and their corresponding nearest most important neighboring regions. We enforce the model to learn from the most crucial features within each node and highlight the critical neighbors for describing the particular scene. Since we exploit the regional interactions to highlight the actual node features, we refer to this as context-attention. We establish its efficacy in RS aerial image retrieval from our studies and experiments.

Our present contributions are as follows.

Manuscript received April 21, 2021; revised July 18, 2021; accepted August 13, 2021. This work was supported by the Research Project under Grant RD/0518-IRCCSH0-008. *(Corresponding author: Ushasi Chaudhuri.)*

Ushasi Chaudhuri, Biplab Banerjee, and Avik Bhattacharya are with the Centre of Studies in Resources Engineering (CSRE), IIT Bombay, Mumbai 400076, India (e-mail: ushasi2cool@gmail.com; getbiplab@gmail.com; avikb@csre.iitb.ac.in).

Mihai Datcu is with the German Aerospace Center (DLR), 82234 Oberpfaffenhofen, Germany (e-mail: mihai.datcu@dlr.de).

Color versions of one or more figures in this letter are available at https://doi.org/10.1109/LGRS.2021.3105448.

Digital Object Identifier 10.1109/LGRS.2021.3105448

1) We perform a simple linear iterative clustering (SLIC) superpixel-based image segmentation. It produces high-quality superpixels at low computational complexity.
2) We construct a region-adjacency graph (RAG) from these images. Here, each segment acts like a graph node, and their neighborhood serves as the adjacency matrix.
3) We use discriminative learning of the embedding space features from an irregular and non-Euclidean spatial distribution of regions (i.e., graphs) by using GCNs.
4) Finally, we append the above framework using the context-attended attention framework and present its performance for RS aerial image retrieval. We perform our experiments on the benchmarked UC-Merced [8] and the PatternNet [9] datasets.

II. Methodology

Let us consider $\chi = \{x_i, y_i\}_{i=1}^{n}$ as a very high resolution (VHR) image training dataset comprising of n images from Y different categories. We aim to learn a retrieval framework from χ, for any given query image x_q from a test dataset $x_{j'}$ without any label information.

A. Region Adjacency Graphs

To convert the images into graphs in a non-Euclidean space, we first perform a SLIC superpixel-based segmentation of the images. Each segment acts as a node of a graph, and their nearest neighbors provide the adjacency information. Thus we construct RAG from each image by performing superpixel segmentation. We construct the graph representation for the ith image as $g_i = (E_i, A_i)$ where E_i represents the weighted graph adjacency matrix created from the RAG of x_i, and A_i represents the node level attributes.

1) Node Features A_i: To build node attributes, we compute features for each segmented region. As image segmentation leads to irregular regions, extracting CNN-based data-driven features is challenging. Also, since SLIC leads to homogeneous regions, simple handcrafted features can perform nearly as good as deep features. This essentially comprises shape features, color features, and texture features [10]. The shape features comprise Fourier descriptors and counter-based features, such as area, perimeter, solidity, eccentricity, orientation, bounding boxes, Euler number, and convex-area. The color features comprise normalized color histograms and color moments. Similarly, the texture features constitute spectral histograms, horizontal and vertical Sobel filter output, entropy, local binary patterns, and local phase quantization-based features. We concatenate all these features into a vector of dimensions 359. These node attributes are row-stacked to create the overall feature matrix $\mathbf{A}_i \in \mathbb{R}^{p_i \times 359}$, where p_i is the pth segment in the ith image.

2) Edge Adjacency E_i: In the segmented image, an edge $E_i(r_i^j, r_i^k)$ is said to exist between two segments, if they are spatial neighbors. To create the weighted adjacency matrix, we calculate the distance between the centroids of two segments $c_{r_i^j}$ and $c_{r_i^k}$. We also find the orientation angle $\theta_{r_i^j}$ between them by calculating the angle between the horizontal axis and the major axis of the best fit eclipse of the segment. The orientation angle is bounded between $\theta \in [-90°, 90°]$) as they can generalize well for any angle in the Cartesian space.

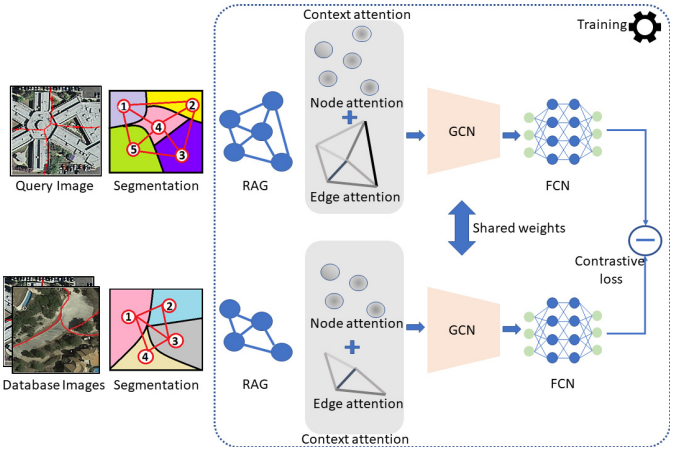


Fig. 1. SGCN framework exploiting the proposed context-attended attention.

The following equation helps us assigning the weights of the adjacency matrix E_{ij}:

$$E_{ij}\left(r_i^j, r_i^k\right) = \left|\left|c_{r_i^j} - c_{r_i^k}\right|\right|_2 + \lambda\left|\theta_{r_i^j} - \theta_{r_i^k}\right| \tag{1}$$

if $r, k \in \mathcal{N}(r, j)$, where $\mathcal{N}$ represents neighbor and λ is a hyperparameter. Depending on the superpixel segmentation algorithm, different images will result in a variable number of segments. Therefore, it requires a mapping of all these irregular-sized graphs into one common comparable latent space. For this purpose, we use the spatial graph convolution approach proposed in [2]. Previous studies have established the effectiveness of this approach [3], [4]. Here, we instead propose a context-based attention framework for spatial GCN.

B. Siamese Graph Convolution Network

We use a Siamese graph convolution framework to train our overall framework, as shown in Fig. 1. A Siamese architecture requires a pair of graphs as input to the framework. We use a node attention layer for each of the Siamese branches, which brings the features to a 256-dimensional output ($F = 359$, and $F' = 256$). An edge attention layer then follows this process. The remaining framework prevails the same as given in [4], wherein we use a graph convolution of 128 dimensions, followed by a graph embedding layer to have 64 nodes. Finally, we use a dropout layer with probability 0.5, followed by two fully-connected (FC) layers of 256 and C dimensions, where C is the number of classes in the concerning dataset. A target variable t_{ij} is defined here that learns either 0 or 1 if either pair of the input graphs belong to the same target class or from different classes. We minimize a contrastive loss to train the Siamese network as defined in the following equation:

$$\begin{aligned}\mathcal{L}^k = &\sum_{n=1}^{N} \\ &\times \Big(t_{nij}\left\{\max\left(0, m - (\widehat{x}_{ni} - \widehat{x}_{nj})^2\right)\right\} \\ &+ (1 - t_{nij})(\widehat{x}_{ni} - \widehat{x}_{nj})^2 + \lambda_1\left(|\widehat{x}_{ni}|_2 + |\widehat{x}_{nj}|_2\right) + \lambda_2|\phi|_2\Big).\end{aligned} \tag{2}$$

Here, m defines the margin across which we want to push a different class graph embedding in the latent space. The contrastive loss is a ranking loss function. Our system uses

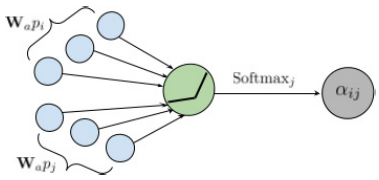


Fig. 2. Illustration of the attention mechanism $a(\mathbf{W}_a p_i, \mathbf{W}_a p_j)$ employed by our model, applying a LeakyReLU activation.

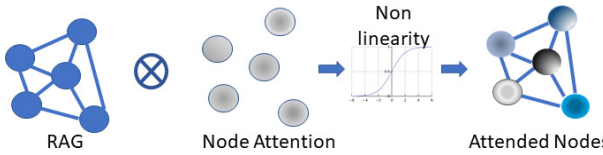


Fig. 3. Illustration of the overall working of node attention. The attention weights learns the most essential features in each node.

pairs to form an inherent ranking order. The training batches take advantage of the distance between the graph embedding vectors to mine complex negative samples for a given anchor class. This implies that graphs from different classes with smaller distances in the feature space will have a higher probability of being used during training.

We add a l_2 norm on the final features and the final layer learnable weights (ϕ) to help stabilize the loss function [4]. When we get two same class graphs as the input, the target t_{ij} is assigned 0. This procedure makes the second term of the loss $\mathcal{L}^k$ to zero, and hence, we only minimize the distance between the embeddings of the two graphs in the latent space. Therefore, the intraclass distances are reduced in the embedding space. Similarly, when we have two different class graph pairs as the input, the target t_{ij} is assigned 1. This procedure makes the first term of the loss $\mathcal{L}^k$ zero. Therefore, we push the two indifferent class features further apart by at least a margin of m in the embedding space. In effect, this increases the interclass embedding feature distances. Here, we experimentally choose the model hyperparameters λ_1 and λ_2. We append this Siamese GCN (SGCN) framework with node attention [6] and the novel edge attention mechanism to highlight the important constructs of the image. The following sub-sections elaborate on the formulation of these attentions.

C. Node Attention

Let us consider an image having $p_i \times F$ node features, where $F = 359$ is the input feature dimension in this case. The set of node features can then be presented as $\mathbf{p} = \{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_R}\}$, where R is the number of nodes. When this input is fed to the attention layer, we obtain a new learned node feature matrix with different cardinality F'. This yields an output node matrix of $\mathbf{p}' = \{\mathbf{p_1}', \mathbf{p_2}', \ldots, \mathbf{p_R}'\}$. To learn an attention on this node matrix $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \mapsto \mathbb{R}$, we train a learnable weight matrix $w_a \in \mathbb{R}^{F' \times F}$, for every node. The importance [6] of a node j's features to node i is then given by

$$e_{ij} = a(\mathbf{W}_a p_i, \mathbf{W}_a p_j). \tag{3}$$

This helps in allowing every node to attend on every other node features. Let us represent the nodes in the neighborhood of node i as $\mathcal{Z}_i$. Further, to get the attention heads comparable across different nodes, Veličković *et al.* [6] propose

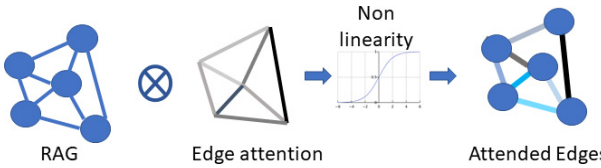


Fig. 4. Illustration of the overall working of the proposed edge attention. The attention weights learn the essential edges around each node. The overall mechanism then combines both node and edge attentions.

normalizing the learned coefficients using a softmax layer as $(\exp(e_{ij}))/(\sum_{k \in \mathcal{Z}_i} \exp(e_{ik}))$.

We consider the attention layer as a FC feed-forward neural network, activated with a nonlinearity LeakyReLU($\cdot$). The learned layer features are concatenated with the input features and back-propagated to learn the attention network. The LeakyReLU function provides a soft attention mechanism on the input node features and prevents learning up a binary 0/1 matrix. The overall node attention module is illustrated in Fig. 2. Expanding out the above softmax layer, the general effective attention layer looks like

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(a[\mathbf{W}_a p_i \parallel \mathbf{W}_a p_j]))}{\sum_{k \in \mathcal{Z}_i} \exp(\text{LeakyRelu}(a[\mathbf{W}_a p_i \parallel \mathbf{W}_a p_j]))}. \tag{4}$$

Effectively, this layer assigns higher importance to the most important features within each node and a lower priority on the less important node features. When concatenated with the original node feature matrix, we obtain highlighted important constructs within the input node features, as shown in Fig. 3.

D. Edge Attention

For each image consisting of p_i nodes, we have a corresponding weighted-adjacency matrix of dimension $p_i \times p_i$. To learn an attention on this edge matrix $b : \mathbb{R}^{p_i} \times \mathbb{R}^{p_i} \mapsto \mathbb{R}$, we propose a novel learnable weight matrix $w_e \in \mathbb{R}^{p_i \times p_i}$, applied to each image. The importance of jth edge to ith edge is then given by

$$h_{ij} = b(\mathbf{W}_e p_i, \mathbf{W}_e p_j). \tag{5}$$

This helps in allowing each edge to attend on every other edge weight. Further, to get the attention heads comparable across different edges, we perform a normalization operation on the learned coefficients similar to the node attention, using a softmax function. The edge attention layer is also employed as a feed-forward neural network, activated with a nonlinearity LeakyReLU($\cdot$). The learned layer weighted-adjacencies are concatenated with the input weighted-adjacency matrix and back-propagated to learn the edge-attention network. The overall attention layer is

$$\beta_{ij} = \frac{\exp(\text{LeakyRelu}(h[\mathbf{W}_e p_i \parallel \mathbf{W}_e p_j]))}{\sum_{i \in \mathrm{R}} \exp(\text{LeakyRelu}(h[\mathbf{W}_e p_i \parallel \mathbf{W}_e p_j]))}. \tag{6}$$

Effectively, this layer assigns higher importance on the most important edges within each image and a lower priority on the less important edges. When concatenated with the original input adjacency matrix, we obtain a highlighted important neighborhoods within the input images, as shown in Fig. 4.

We train the above framework by using a stochastic gradient descent optimizer, using mini-batches for minimizing $\mathcal{L}^k$. Posttraining this context-attended architecture, we use a simple

TABLE I

PERFORMANCE OF DIFFERENT MODELS ON UC-MERCED AND PATTERNNET DATASET. LOWER VALUES OF ANMRR AND HIGHER VALUES OF P@10 AND MAP DENOTES BETTER MODEL PERFORMANCE [9], [11]

Model	UC Merced			PatternNet		
	ANMRR	**mAP@all**	**P@10**	**ANMRR**	**mAP@all**	**P@10**
G-kNN [4]	0.92	07.50	10.12	0.88	12.35	13.24
RAG-kNN [4]	0.75	26.74	24.90	0.69	22.56	37.70
VGG-VD16 [11]	0.38	53.71	78.34	0.33	59.86	92.04
Attention + VGG-VD16	0.39	54.5	78.54	0.34	60.54	93.87
VGG-VD19 [11]	0.39	53.19	77.60	0.34	57.89	91.13
GoogLeNet [11]	0.39	53.13	80.96	0.29	63.11	93.31
GCN [4]	0.33	64.81	87.12	0.28	73.11	95.53
Context Attended-GCN (Ours)	**0.31**	**66.77**	**90.08**	**0.26**	**75.64**	**96.87**
SGCN [4]	0.30	69.89	93.63	0.21	81.79	97.14
Context Attended-SGCN (Ours)	**0.29**	**73.22**	**94.62**	**0.19**	**84.02**	**98.37**

k-nearest neighbor (k-NN) metric to find the top-k retrieved image samples for any given query.

III. EXPERIMENTS AND RESULTS

A. Datasets

To validate our performance of our framework, we experiment on the PatternNet [9] and the UC Merced [8] dataset. The PatternNet dataset consists of 38 classes, consisting of 800 images each, while the UC Merced consists of 21 classes and has 100 images per class.

B. Model Architecture

We keep the overall model architecture consistent with [4] to provide fairness in comparison. We perform the segmentation of the images using SLIC superpixels with a region size of 25 for the Merced dataset and 40 for the PatternNet dataset. Corresponding to each of these segments, we obtained a feature vector of 359 dimensions. We select the same set of features for creating the graphs to keep the backbone framework the same and not induce any backbone architecture-induced bias to the overall results. Initially, we trained the GCN network to optimize a cross-entropy loss function to yield feature embeddings of 256 dimensions. We then feed this optimized GCN framework to the SGCN network, wherein we minimize a contrastive loss function and obtain feature embeddings in 128 dimensions. The SGCN uses shared weights for both the input RAGs.

C. Training and Evaluation Protocol

We use the standard 75:25 to split the data in train:test. We trained the GCN network with a learning rate of 0.01 and a batch size of 32. The network converged in about 2500–3000 iteration for both datasets. The noise margin m in (2) is set to 1 heuristically. The λ_1 and the λ_2 parameters were both set to 0.001, following the sensitivity analysis of critical parameters in [4]. The same set of query images are used in the proposed work as the comparative frameworks to maintain fairness in the evaluation process and avoid bias. For the evaluation purpose, we use the precision@10 (P@10), mean average precision (mAP), and the average normalized modified retrieval rank (ANMRR) metrics for comparison that are consistent with the literature.

For training the SGCN network, we generate the train pairs during runtime. The closest negative class sample is preferred during training as it helps push them apart in the latent feature space. This process increases the interclass distancing between the different cluster centers of features of each class. The SGCN network is a series of three FC layers of 256, 256, and 128 dimensions. We trained the network using the Adam optimizer for about 20 epochs and mini-batches of 32 sample pairs. We set the learning rate to 0.001.

TABLE II

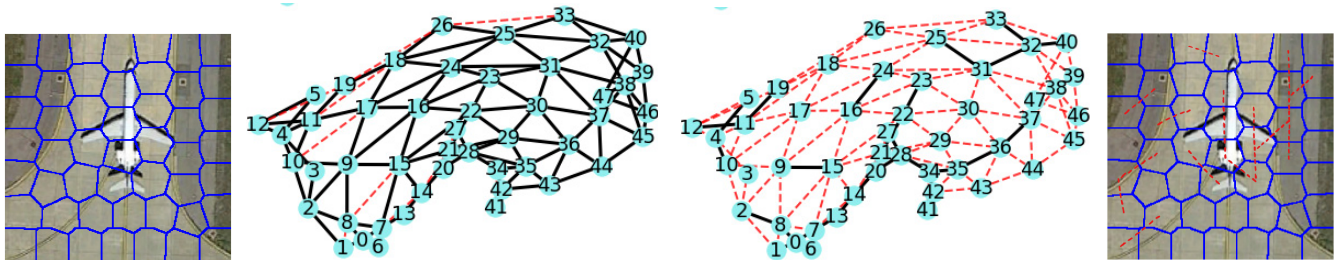
PERFORMANCE OF DIFFERENT MODELS ON UC-MERCED DATASET, IN TERMS OF MAP, ANMRR, AND P@10(%) VALUES

Model	UC Merced		
	ANMRR	**mAP@all**	**P@10**
GCN [4]	0.33	64.81	87.12
Node Attended-GCN	0.32	65.56	88.32
Edge Attended-GCN	0.32	65.85	88.09
Context Attended-GCN	**0.31**	**66.77**	**90.08**
SGCN [4]	0.30	69.89	93.63
Node Attended-SGCN	0.30	70.81	93.86
Edge Attended-SGCN	0.30	71.86	93.89
Context Attended-SGCN	**0.29**	**73.22**	**94.62**

We compare the performance of the proposed framework with that of the first principle component for RAG (G-k-NN), entire RAG (RAG-k-NN), standard CNN using pretrained VGG-net [12], attention network on CNN (VGG-16), and GoogLeNet [13]. Also, we compare the proposed method with a standard GCN and SGCN [4] in Table I. We use the same RAG to maintain consistency and fairness in the comparisons for all the graph-based comparisons. Since this method uses handcrafted features extracted from arbitrarily shaped segment regions, we stick to similar graph-based comparative frameworks. We apply our context-attended GCN on the simple GCN framework, as well as on the SGCN framework. We observed an increase of about 1%–2% in the overall P@10 and mAP values for both datasets in the two cases. This analysis supports us in establishing the efficacy of the proposed framework in RS images.

D. Ablation Studies

We perform a model ablation by considering each module of the overall framework individually. We chose the GCN and the SGCN models as the baseline networks. We report the ablation results on the UC Merced and the PatternNet datasets in Tables II and III. The baseline GCN model yields a P@10 value of 87.12% and 95.53% on the Merced and



(a) (b) (c) (d)

Fig. 5. Illustration of the original and the learned graph. We denote the edges weighing more than 0.1 in black for representation purpose, while the edges weighing less than 0.1 are represented with red dotted lines. (a) Segmented image. (b) Original edge weights of RAG. (c) Attended edge weights. (d) Attended edges.

TABLE III

PERFORMANCE OF DIFFERENT MODELS ON PATTERNNET DATASET, IN TERMS OF MAP, ANMRR, AND P@10(%) VALUES

Model	PatternNet		
	ANMRR	**mAP@all**	**P@10**
GCN [4]	0.28	73.11	95.53
Node Attended-GCN	0.27	74.93	96.12
Edge Attended-GCN	0.28	74.68	96.09
Context Attended-GCN	**0.26**	**75.64**	**96.87**
SGCN [4]	0.21	81.79	97.14
Node Attended-SGCN	0.20	83.14	97.76
Edge Attended-SGCN	0.20	82.93	97.71
Context Attended-SGCN	**0.19**	**84.02**	**98.37**

PatternNet datasets, respectively. When we add the node attention [6] to this, we obtain an increase of ≈1% in the P@10 value in both the dataset. The addition of this prunes the essential features of each region and highlights them, thereby boosting overall performance. We also achieve ≈1% increase in the overall performance for all the evaluation metrics upon using the proposed edge attention. This module helps in pruning the important neighborhood information in a scene and assigns more weight to them. We achieve ≈2% increase in the overall performance using all the metrics for the attention in conjugation to the GCN framework. Likewise, we use SGCN as the baseline architecture and perform the above set of ablation studies on both datasets. We can note here that considering the baseline has already provided a high performance, even a ≈2% increase in the performance above 90% can be considered a noteworthy improvement.

Fig. 5(a) shows a sample image from the PatternNet dataset. We perform a rudimentary segmentation and display its corresponding segmentation mask. We enumerate the nodes from 1 to 49. Fig. 5(b) and (c) shows the corresponding initial edge adjacency matrix and the attended edge adjacency matrix, respectively. We denote the edges weighing more than 0.1 in black for representation purposes, while the remaining edges in red dotted lines. We choose a value of 0.1 on an *ad hoc* basis. In Fig. 5(d), we show the attended edges which receive the maximal importance in the embedding space by connecting their region centroids with red dotted lines. Please note that the overall retrieval performance has improved even though the attended edge graph comprises fewer edges with high weights. This result implies that emphasizing meaningful neighborhood interactions by implementing context attention assists in the overall representability of the image.

IV. CONCLUSION

We propose a novel context-attention network for GCNs, which comprises node and edge attentions. In addition to highlighting essential features within each node, edge attention enables the network to learn the most critical neighborhood constructs from the RAGs within each target class image. By learning the local interactions, it preserves the neighborhood information of the structures and stores the nonlinearities. Together with node attention, it comprises the context-attention network for GCNs. We demonstrate the effectiveness of attention on standard GCN architecture and the state-of-the-art SGCN architecture and demonstrate superior CBIR performance in both cases on VHR RS datasets.

REFERENCES

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, pp. 1–14, Sep. 2016.

[3] F. P. Such, "Robust spatial filtering with graph convolutional neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 884–896, Sep. 2017.

[4] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Comput. Vis. Image Understand.*, vol. 184, pp. 22–30, Jul. 2019.

[5] N. Khan, U. Chaudhuri, B. Banerjee, and S. Chaudhuri, "Graph convolutional network for multi-label VHR remote sensing scene recognition," *Neurocomputing*, vol. 357, pp. 36–46, Sep. 2019.

[6] P. Veliákoviä, G. Cucurull, A. Casanova, A. Romero, P. Liü, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903

[7] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," 2018, *arXiv:1802.09691*. [Online]. Available: http://arxiv.org/abs/1802.09691

[8] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.

[9] W. Zhou, S. Newsam, C. Li, and Z. Shao, "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 197–209, Nov. 2018.

[10] B. Chaudhuri, B. Demir, L. Bruzzone, and S. Chaudhuri, "Region-based retrieval of remote sensing images using an unsupervised graph-theoretic approach," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 7, pp. 987–991, Jul. 2016.

[11] P. Napoletano, "Visual descriptors for content-based retrieval of remote-sensing images," *Int. J. Remote Sens.*, vol. 39, no. 5, pp. 1343–1376, Mar. 2018.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[13] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.