Clustering and Graph Convolution of Sub-Regions for Unsupervised Image Segmentation

XUE JIAO
Henan Institute of Science and Technology, Henan 453003, China
e-mail: jiaoxue_jledu@126.com

This work was supported in part by the Key Scientific Research Project in Colleges and Universities of Henan Province of China under Grant 22B110004, and in part by the Key Science and Technology Research of Henan Province of China under Grant 222102110409.

Abstract
This paper focuses on the unsupervised segmentation of images, which is an essential topic in the field of computer vision. In the absence of prior knowledge, it is challenging to generate semantic segmentation regions based on image content automatically. In this paper, we consider unsupervised image segmentation from the perspective of sub-region clustering and graph convolution. We over-segment the source image into disjoint sub-regions and generate multiscale representative maps for each sub-region. To explore the potential contextual correlation between different sub-regions, we build a graph model that establishes the dependency relationship and use a graph convolution network to transfer long-range contextual information. The segmentation results are obtained by combining sub-regions clustering and graph convolution training. We conduct extensive experiments on three image datasets, and the results show that the proposed method can provide consistent and meaningful segmentation results.

Index Terms
Image segmentation, unsupervised learning, graph, sub-regions, clustering.

I. INTRODUCTION
As one of the essential computer vision tasks, image segmentation intends to partition images into disjoint regions by predicting each pixel’s category. The application scope of image segmentation includes scene understanding, image retrieval, medical image analysis, and video surveillance.

Since the success of deep learning in the computer vision field, segmentation methods based on convolutional networks have shown remarkable performance. Typically, most of these methods are trained using the models such as the fully convolutional network [1], and employ supervised learning methods to generate pixel-wise predictions. The success of these supervised segmentation methods depends strongly on the availability of pixel-level labels. Unfortunately, the annotations of these data are commonly time-consuming and laborious. By contrast, unsupervised methods are almost independent of training data and can be applied in many cases.

Considering the importance of segmentation in many fields and the lack of supervised data on novel domains, we devote to researching the image segmentation method in unsupervised learning settings. We approach the unsupervised segmentation problem from the perspective of sub-regions clustering and graph convolution. We segment the given image into meaningful, coherent regions according to the classification prediction generated by clustering and graph convolutional training. Different from the segmentation methods based on pixel-wise feature similarity clustering, the motivation of our method comes from the graphical reasoning of semantic information correlation in sub-regions.

As illustrated in Figure 1, the input of our method only needs one image without any additional training data. We start the whole process with two operations, one is over-segmentation, and the other is feature maps extraction. We over-segment the source image into several disjoint sub-regions and use a Convolutional Neural Networks (CNNs) model to generate the initial feature maps. Representative feature maps are extracted on the premise without changing the original semantic information and intrinsic context of initial feature maps. Then, to understand the potential contextual correlation between different sub-regions, we build a graph model to encode sub-regions’ semantic content. Furthermore, we use a graph convolution network to transfer long-range contextual information between the graph nodes. Finally, we
integrate clustering with graph convolutional training to achieve the final segmentation.

We evaluate our approach using three standard datasets, including the BSDS500 [2], Pascal VOC [3], and MS-COCO datasets [4]. We perform ablation experiments on the BSDS500 and Pascal VOC datasets to evaluate the performance of the key components of our method. On the MS-COCO dataset, we compare our method with different combination strategies by considering label initialization methods and clustering methods. Furthermore, we make statistics on the execution time by employing different computer configurations. The experimental results demonstrate that our method achieves a good trade-off between efficiency and effectiveness.

In summary, our work makes the following contributions:
1) We propose an unsupervised segmentation method based on semantic information relevance reasoning in sub-regions;
2) We propose a method to extract representative maps by reducing the number of original feature maps without altering the resolution and properties;
3) We construct a graph structure to encode the image content and achieve meaningful segmentation results by applying the combination of iterative clustering and graph convolution to transfer contextual information of the sub-regions.

The rest of the paper is organized as follows: Section 2 reviews related works in the literature. Section 3 describes the detailed procedures of the proposed method. Section 4 presents the experimental results, and the conclusions are presented in Section 5.

II. RELATED WORK
This section introduces the references related to our work. We start by providing an overview of the representative work on unsupervised image segmentation, followed by a brief introduction of graph convolution network applications in image segmentation.

A. UNSUPERVISED SEGMENTATION
Many unsupervised segmentation methods use features such as color, brightness, gradient, or texture to obtain pixel-level segment results of the input image. The most widely-used methods include the K-means method [5], Mean-Shift method [6], Fuzzy C-means [7], and classical graph-based segmentation methods [8], [9]. In recent years, some segmentation methods based on high-level feature analysis have been proposed. Pinheiro et al. [10] presented a method that generates segmentation object proposals directly from image pixels. Aksoy et al. [11] proposed a segmentation method that automatically generates soft segments for the input images. Wang et al. [12] used a color block weighting method to calculate the weight of edges in dual graphs. Each node represents a patch rather than a pixel, and each patch has local and global weights. Yu et al. [13] proposed an unsupervised image segmentation method based on the hierarchical clustering algorithm. Cheng et al. [14] proposed a real-time segmentation system called Hierarchical Feature Selection (HFS) that first uses over-segmentation to acquire seed regions. Zhou and Wei [15] proposed a deep image clustering (DIC) model, which consists of a feature transformation subnetwork (FTS) and a differentiable deep clustering subnetwork (DCS) for dividing the image space into different clusters. Ilyas et al. [16] proposed a CNN-based architecture for unsupervised segmentation that combined CNN architecture with a graph-based method to generate the target segments.

B. GCNs IN SEGMENTATION
Graph Convolutional Networks (GCNs) was initially introduced by Kipf in [17]. As a scalable approach based on an efficient variant of convolutional neural networks, GCNs can operate directly on graphs. GCNs achieve impressive performance in a wide variety of fields, such as scene understanding, molecule prediction, and visual recognition. Some works [18], [19] analyzed and explained spectral theory and GCN method in detail. Li et al. [20] conducted an extensive study on the number of layers and performance of GCN and pointed out that stacking more layers into a GCN leads to the common vanishing gradient problem.

In semantic segmentation, Zhang and Li [21] designed a dual graph convolutional network (DGCNet) to model the global context of the input feature. Their architecture integrates two orthogonal graphs. One is the coordinate space GCN that models the spatial relationships between the pixels in the image; the other is the feature space GCN that models interdependencies along the channel dimensions of the network feature map. Lin et al. [22] presented a graph-guided architecture search (GAS) pipeline to automatically search real-time semantic segmentation networks. The GCN is applied to model the relationship of adjacent cells in network architecture search. Li et al. [23] proposed a spatial pyramid based graph reasoning (SpyGR) layer method that uses an improved Laplacian formulation for graph reasoning in the original CNN feature space organized as a spatial pyramid. While the above-mentioned segmentation methods with GCN have achieved good results, they are mainly based on supervised and semisupervised segmentation methods. The application of GCN in unsupervised segmentation is still a practical problem worthy of further research.

III. PROPOSED METHOD
In this section, we describe the framework of the proposed unsupervised segmentation method in detail. Section 3.1 first describes the rationale and method of using oversegmentation to generate sub-regions. Section 3.2 describes the module for extracting representative feature maps. Section 3.3 presents sub-region based graph construction and uses GCN to generate deep semantic feature extraction by GCN. Finally, we describe the clustering procedure of the sub-regions in Section 3.4.
We take the first five convolution layers of the ResNet-101 network as the initial feature map extractor. The first convolution layer is a $7 \times 7$ convolution followed by $3 \times 3$ max pooling. Each of the last four convolution layers comprises several basic blocks, including $1 \times 1$, $3 \times 3$, and $1 \times 1$ convolution layers. The initial channels in each layer are 64, 128, 256, and 512. Then, three dilation convolutions are performed with dilation rates set to 6, 12, and 18. Finally, a $1 \times 1$ convolution with batch normalization is performed to obtain the feature maps that encode multi-receptive-field contextual information. In addition, bilinear upsampling is necessary to obtain the same resolution as the original image. We denote the initial feature maps collection as $F_{\text{map}}$, and use $m$ to represent the number of elements of $F_{\text{map}}$.

For a pair of feature maps, we use mutual information (MI) [26] to measure their correlation and mutual inclusion. For two feature maps, a greater MI value indicates a higher correlation. We obtain representative maps by filtering the collection of initial feature maps iteratively. The filtering process relies on intrinsic information of the feature maps instead of assigning the threshold parameters in advance.

Specifically, for a feature map $f_i \in F_{\text{map}}$, we calculate the MI value between it and other feature maps, and denote the feature maps with maximum and minimum MI values by $f_{\text{max}}$ and $f_{\text{min}}$, respectively. We remove $f_{\text{max}}$ from $F_{\text{map}}$ based on the consideration that $f_{\text{max}}$ is the feature map with the closest feature information to $f_i$. Then, we start the next remove operation from $f_{\text{min}}$ that has a significant discrepancy with $f_i$.

After iteratively removing operations on $F_{\text{map}}$, the remaining feature maps are representative maps. We denote the collection of representative maps as $R_{\text{map}}$. The number of iterations depends on the number of representative maps. The iterative removal operation stops when the number of representative maps reaches the expected value. We determined
the number of representative maps by conducting evaluation experiments on the BSDS500 and PASCAL VOC2012 datasets. The experimental results of the number of representative maps are shown in Table 1, and an example of representative maps is illustrated in Figure 2.

C. GRAPH CONSTRUCTION AND CONVOLUTION

We obtain nonoverlapping sub-regions after the image is oversegmented. Nevertheless, it is necessary to address the problem of the lack of well-defined prior knowledge to guide the correct connection of these sub-regions. Therefore, it is necessary to explore the deeper semantic content of each region. To solve this problem, we construct an undirected graph to encode contextual information and use a graph convolution network (GCN) to generate the global semantic features by transferring the long-range contextual information of sub-regions.

1) GRAPH CONSTRUCTION

We transform each sub-region into a corresponding graph node. The annotation of the graph node \( v_i \) is modeled by the concatenation of the feature vectors in the representative maps.

Through this initialization method, each node is equipped with high-level semantic information and local spatial information. Then, we apply the \( k \)-NN method to initialize node’s neighbors and use cosine similarity to estimate the correlation distance between graph nodes. We use cosine distance instead of Euclidean distance to calculate the feature similarity of two graph nodes. Compared with the Euclidean distance metric, cosine distance measures the difference in the direction of two vectors rather than length difference. For two feature vectors with similar information but significantly different lengths, the cosine distance can better measure their similarity than Euclidean distance. Considering that the range of cosine value is \([-1, 1]\), we need to further normalize it to \([0, 1]\). The correlation distance between two nodes is defined as follows:

\[
d(v_i, v_j) = \frac{1}{2} + \frac{1}{2} \cos(v_i, v_j),
\]

The greater the value of correlation distance, the more similar the two graph nodes are.

The first \( k \) nodes with the smallest distance are selected as the neighboring nodes of point \( v_i \), and a graph edge is added between two neighboring nodes. If the value of \( k \) is too small, the entire segmentation model becomes complicated, and the computational efficiency decreases. In addition, the experimental results are prone to over-segmentation because of the disability associated with distant sub-regions. On the other hand, if the value of \( k \) is too large, the sub-regions far away can be related to each other, and the segmentation model has strong robustness. However, its disadvantage is that the model is too simple to analyze the regional background information well. Considering the trade-off between efficiency and performance, in the experiment, we fixed the value of \( k \) to 4.

2) GRAPH CONVOLUTION

By converting the source image into a corresponding graph \( G \), we transform the image segmentation problem into the clustering task of graph nodes. Compared with the nodes that are closer in embedding space, we also need to establish a dependency relationship between nodes that belong to the same object but are far away from each other. In our method, we use GCN to transfer the feature information of the graph nodes and generate global semantic features based on long-range context.

In GCN, each node \( v_i \in \mathcal{V} \) is represented by a feature vector \( h_i \in \mathbb{R}^{d_f} \), where \( d_f \) is the feature dimension of node \( v_i \) after the \( l \)-th convolution layer. The message of the nodes can be passed by the edges in the graph neural network. The convolution operation \( \mathcal{F} \) in a multilayer GCN is given by Kipf et al. [17]:

\[
H^{(l+1)} = \mathcal{F}(H^{(l)}, A) = \sigma(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}W^{(l)}),
\]

where \( H^{(l)} \) is the node features of the \( l \)-th layer. \( A \) is the adjacent matrix, and the term in \( A \) is defined as \( A_{ij} = 1 \) if \( v_i \) and \( v_j \) are connected with an edge; otherwise, \( A_{ij} = 0 \). \( D \) is the degree matrix with \( D_{jj} = \sum_j A_{ij} \), \( W^{(l)} \) is the learnable weight matrix in layer \( l \), and \( \sigma \) is a nonlinear activation function. (A more detailed derivation and analysis can be found in [19], [27].) In our method, the GCN takes the representative feature matrix of graph nodes together with the adjacency matrix as input. The output of GCN is a transformed node feature matrix. The GCN in our method has two hidden layers, and the feature dimension of the graph node in each layer are 200 and 180, respectively. The feature dimension of the graph node in the output layer is 120. We train the GCN using the Adam optimizer [28] on a cross-entropy loss function.

D. CLUSTERING

We generate deeper global semantic features by converting the image into a graph structure and propagating each layer’s node features with GCN. Then, we cluster similar sub-regions by taking the argmax function on the prediction of graph nodes’ labels. Algorithm 1 gives the pseud-code of the clustering process. We use a 4-layer CNN model to create the initial graph node labels. Each layer consists of a convolution layer (kernel size is 3, 3, 3, 1, respectively), batch normalization, and a ReLU function. We use this model to process the source image once and count pixels’ prediction categories belonging to each sub-region. We take the group with the most frequent occurrence as the initial label of the graph node. Then, training the GCN model updates the embedded features of the graph nodes. The cluster categories of the graph nodes are assigned via argmax classification.
Algorithm 1 The Clustering Procedure

Input: input RGB image $I = \{x_i\}_{i=1}^{m \times n}$
Output: labels of graph nodes $L = \{l_i\}_{i=1}^{N}$

1: Generate initial sub-regions $S = \{S_1, \ldots, S_N\}$
2: Calculate representative maps $R_{map}$
3: Calculate nodes’ features $N_f$
4: Construction Graph $G$ based on $k$-NN Method
5: Calculate Initial Labels:
6: for $t = 1$ to $T$ do
7: $y_i = \{\text{Conv}(x_i)\}_t$
8: $\tilde{y}_i = \{\text{BatchNorm}(y_i)\}_t$
9: $\hat{y}_i = \{\text{ReLU}(\tilde{y}_i)\}_t$
10: end for
11: $\tilde{l}_j = \text{arg max}_i \hat{y}_i \in S_j$
12: Training Graph Covolutional Network:
13: for $k = 1$ to $K$ do
14: $G_{net} \leftarrow \text{GCN}(G, N_f)$
15: $N_f \leftarrow \text{CrossEntropy}(G_{net}, \{\tilde{l}_j\}_{j=1}^{N})$
16: end for
17: $L \leftarrow \text{arg max} N_f$
18: return $L$

The segmentation label of the image pixel depends on the final category of the graph node to which it belongs.

IV. EXPERIMENTS

A. DATASETS

We perform evaluation experiments on three datasets, including the BSDS500 [2], Pascal VOC2012 [3], and MS-COCO [4] datasets. The BSDS500 dataset consists of 500 images and human-annotated ground truths, and it is a commonly used benchmark for image segmentation and boundary detection methods. The Pascal VOC2012 dataset refers to the visual object classes challenge, it containing 11,530 images and 6,929 segmentations. The MS-COCO is an enormous dataset designed for the detection and segmentation of objects occurring in their natural context. This dataset consists of 328K real-world photos collected from Flickr. For each image, there is pixel-level segmentation of 80 object categories. We evaluate the performance of the key components in this method through ablation experiments. Besides, we make statistics on the execution time by employing different computer configurations.

B. IMPLEMENTATION DETAILS

Our method only needs to input a single color image. In the experimental process, we did not carry out the enhancement and clipping operations on the input images. Since our approach belongs to the category of unsupervised segmentation, we do not use any ground truth labels in the training process, and the ground truths are only employed to evaluate the quality of the segmentation results.

In our experiments, for the representative map set $R_{map} = \{f_i\}_{i=1}^{N}$, we take the z-score standardized method to normalize data to prevent the segmentation results from collapsing into one class. According to suggestions in [20] that GCNs do not run well in deep architectures because stacking in multiple layers of graph convolution leads to high complexity and vanishing gradients. Our method adopts a GCN with two layers in
all experiments, and weights are initialized randomly. We use ReLU as the activation function, and Adam as the optimizer. Batch normalization and activation functions are applied in the first layer. We trained the network for 200 epochs, and the initial learning rate is set to 0.0001. The image pixel’s final segmentation label is consistent with the clustering label of the sub-region (corresponding to a graph node) to which it belongs.

C. RESULTS ON BSDS500 AND PASCAL VOC2012
The BSDS500 and Pascal VOC2012 are the standard benchmark datasets for image segmentation and clustering tasks. We first perform ablation experiments on these two datasets to evaluate the performance of the key components of our method. We take the entire BSDS500 dataset and 500 images derived from Pascal VOC2012 as the experimental images and use the mean intersection of union (mIoU) as the quantitative evaluation metric.

1) EVALUATION OF FEATURE EXTRACTOR
To analyze the influence of feature extraction on segmentation accuracy, we employ ResNet-50 and ResNet-101 as the backbone to extract feature maps. In the experiment, we obtain 500 feature maps for each image in the experiments and alter the number of representative maps to 200, 300, and 400. We calculate the IoU value of each prediction segments and ground truth segments, then use the mean value of all of the segments to measure the performance. The higher value of mIoU is better. The evaluation experimental of the number of representative maps are shown in Table 1. The results show that the mIoU decreases with the increasing \( r_n \). We consider that the increase in feature maps will cause some slight interference with the segmentation results. In terms of performance, different backbones have no significant effect on the segmentation results. Our method uses ResNet-101 as the backbone to extract feature maps and sets the number of representative maps to 200.

![Table 1: The evaluation of the number of representative maps. We use two different backbones as the feature extractor. \( r_n \) denotes the number of representative maps.](image1)

| Backbone   | \( r_n \) | mIoU |
|------------|-----------|------|
| ResNet-50  | 400       | 0.676|
| ResNet-50  | 300       | 0.676|
| ResNet-50  | 200       | 0.678|
| ResNet-101 | 400       | 0.628|
| ResNet-101 | 300       | 0.628|
| ResNet-101 | 200       | 0.691|

2) GENERATION OF INITIAL LABELS
In our method, we use CNN to estimate the initial label of graph nodes. Other strategies can also be used for initialization. For example, on the premise of keeping efficiency, using K-means clustering [5], Felzenswalb [8], or SLIC methods [24] assign node labels directly. In the experiment, we set the max cluster number of SLIC and K-means to 100, and the value of \( \sigma \) in Felzenswalb is 0.5. The experimental results obtained by these methods are shown in Table 2. The mIoU value of the SLIC method is low, and the difference between the K-means method and CNN method is small. In contrast to these methods that need to specify the number of initial clusters or fixed parameters in advance, we adopt the CNN model to determine the initial labels automatically.

![Table 2: Test segmentation results with different label initialization strategy of graph nodes.](image2)

| Methods      | SLIC  | Felzenswalb | K-means | CNN |
|--------------|-------|-------------|---------|-----|
| mIoU         | 0.415 | 0.582       | 0.676   | 0.691|

3) EVALUATION OF GCN LAYERS
Table 3 shows the average performance of segmentation results obtained by setting different layer numbers of GCN. We set the number of GCN layers as 2, 3, 4, and apply the K-means method and our CNN model to initialize the labels for each case.

![Table 3: Experimental results on different number of network layers.](image3)

| Methods                  | mIoU | mAcc |
|--------------------------|------|------|
| K-means+2-layers GCN     | 0.676| 0.762|
| K-means+3-layers GCN     | 0.673| 0.759|
| K-means+4-layers GCN     | 0.667| 0.747|
| CNN+2-layers GCN         | 0.691| 0.776|
| CNN+3-layers GCN         | 0.678| 0.772|
| CNN+4-layers GCN         | 0.661| 0.754|

For a fair comparison, we fix the dimension of graph nodes to 180 and 120 in each layer of the 2-layers GCN; the node dimensions of the 3-layers GCN are 180, 160, and 120; and the node dimensions of the 4-layers GCN are 180, 160, 120, and 100, respectively. In addition to mIoU, we also report the evaluation results of mean accuracy (mAcc) on the segmentation task. The segmentation quality will decrease with the increase in the number of GCN layers, mainly because more layers will generate more fragments. The experimental results show that using CNN for label initialization and two-layers GCN is better.

D. RESULTS ON MS-COCO
The MS-COCO dataset is a very large-scale image dataset that can be used for various visual tasks such as object detection, keypoint detection, stuff segmentation, panoptic segmentation, and image annotation. The whole dataset provides 82K images for training, and 40K images for verification, and the test set comprises more than 80K images. For the segmentation tasks, we can use the stuff segmentation part of the MS-COCO dataset for the evaluation of our method. This portion provides 118K labeled images for training and 5K images for testing. Compared to BSDS500 and Pascal VOC2012, this image dataset challenges semantic segmentation at the pixel-level because many images contain complex scenes or backgrounds.
We compare our method with different combination strategies by considering other node label initialization methods and clustering methods. In addition to the mIoU metric, we use two other criteria namely precision and variation of information (VI) for performance evaluation. Accuracy measures the proportion of the predicted category labels belonging to the objects of interest. VI measures the distance between the predicted results and the benchmark results based on the average conditional entropy. For VI metrics, a lower value indicates superior performance. For the other two indicators, a higher value indicates better performance.

Considering that the training time of our method cost on the entire MS-COCO training set is enormous, we divide the original training set into five portions. Then, we extract 2000 images from each portion, that is 10K images in total for training. We evaluate the metrics one by one and take the average values as the evaluation results. The quantitative evaluation results are summarized in Table 4. In the table, K-means+GC and SLIC+GC represent K-means and SLIC as label initialization methods and follow our graph clustering process. CNN+spectral means the use of CNN as label initialization and spectral clustering method for image segmentation. CNN+K-means represents the segmentation methods with CNN as the label initialization method and K-means as the clustering methods.

We also make statistics on the execution time by employing different computer configurations. We tested the performance of the main components on an Intel i7 with a GTX1060 GPU and an Intel i9 with an RTX2080Ti GPU. Table 5 gives the evaluation results of the running time of the main steps. In this table, we still use $r_n$ to represent the number of representation maps. The experimental results show that the most time-consuming procedure is graph construction, which takes about 7 seconds on Intel i7 with GTX1060, and about 5 seconds on Intel i9 with RTX2080Ti. The main reason is that most images in the MS-COCO dataset are large. The corresponding graphs have a high number of nodes and edges, leading to the construction of a similarity matrix that takes plenty of time. Approximately 5 seconds is required for Intel i7 with GTX1060 to extract the representative features and GCN clustering. Intel i9 with RTX2080Ti can significantly improve the efficiency of these two procedures, reducing the required time by almost half. In addition, there is no significant difference between the two devices for the nodes'
TABLE 5. The average time costs measured in seconds corresponding to different numbers of nodes.

(a) The experiments on intel Core i7 with GTX1060 GPU.

| r_n  | 200   | 300   | 400   |
|------|-------|-------|-------|
| FE   | 4.6   | 4.8   | 4.8   |
| GC   | 6.3   | 6.8   | 7.1   |
| IL   | 1.4   | 1.5   | 1.7   |
| GCN Training | 5.3 | 5.3 | 5.7 |

(b) The experiments on intel Core i9 with RTX2080Ti GPU.

| r_n  | 200   | 300   | 400   |
|------|-------|-------|-------|
| FE   | 1.8   | 1.9   | 2.2   |
| GC   | 4.8   | 5.2   | 5.3   |
| IL   | 1.3   | 1.4   | 1.6   |
| GCN Training | 3.4 | 3.5 | 3.7 |

FIGURE 4. Visual comparisons of our method with the other unsupervised segmentation algorithms. The first column displays source images from the BSDS500 dataset. The second to fourth columns show the results generated by HFS [14], DIC [15], and W-Net [32], respectively. The last column shows the results of our method.

E. COMPARISON EXPERIMENTS

We compare the performance and efficiency of the proposed method with several state-of-the-art unsupervised segmentation methods. Considering that our method is a clustering-based segmentation method, we compare our approach with three classical clustering methods, including the Mean Shift method [6], Ncuts method [9], and Felzenzwalb and Huttenlocher’s graph-based method (Felz-Hutt) [8]. Since our method uses sub-region clustering and feature aggregation strategies, we also compare the proposed method with the segmentation methods that use similar processing manners. These methods include PEF+K-means [29], MCG [30], gPb-owt-ucm [2], LGM [31], and HFS method [14]. In addition, we evaluate the proposed method with two unsupervised segmentation methods based on convolution networks: W-Net [32] and DIC [15].

Because the BSDS500 dataset contains multiresolution images, it is ideal for testing the clustering-based segmentation method. In addition, all the above-mentioned comparative segmentation methods use this image dataset as the benchmark dataset for experimental performance evaluation. Therefore, we conduct quantitative evaluations and comparative experiments on the BSDS500 dataset. We use three label initialization process, and both take approximately 1-2 seconds.

Figure 3 shows the average execution time of principal components and the total running time on different computer configurations. The test time is measured in seconds. In this figure, “FE” represents the feature maps extraction process, “GC” represents the graph construction, “LI” and “GT” denote label initialization and graph convolution training, respectively. “T_time” denotes the total running time of our method. Overall, the time consumption will increase with the number of representative feature extractions, and the more powerful GPU can significantly reduce the total running time in general.
TABLE 6. Performance evaluation of the proposed method on BSDS500 (The scores of different methods used for comparison are collected from the tables in [15], [29], [37]).

| Methods             | SC    | PRI   | VI    |
|---------------------|-------|-------|-------|
| NCuts[9]            | 0.53  | 0.80  | 1.89  |
| Mean Shift[6]       | 0.58  | 0.81  | 1.64  |
| PFE+K-means[29]     | 0.56  | 0.81  | 1.77  |
| Felz-Hutt[8]        | 0.57  | 0.82  | 1.87  |
| gPb-owt-ucm[29]     | 0.65  | 0.86  | 1.41  |
| LGM[31]             | 0.56  | 0.81  | 1.79  |
| IIFS[14]            | 0.61  | 0.84  | 1.68  |
| W-Net[32]           | 0.62  | 0.84  | 1.60  |
| DIC[15]             | 0.66  | 0.86  | 1.63  |
| Our Method          | 0.67  | 0.86  | 1.62  |

common metrics as quantitative indicators: segment cover (SC) [33], [34], probabilistic region index (PRI) [35], and variation of information (VI) [36]. The scores of different methods for comparison are collected from [15], [29], [37], and the quantitative performances are reported in Table 6. Larger values of PRI and SC correspond to smaller values of VI and better segmentation performance. The experimental results show that our method can obtain competitive performance at SC and PRI scores compared to the DIC, W-Net, and gPb-owt-ucm method. The VI value of our method is very close to the performance of the W-Net and DIC methods. Figure 4 illustrates the qualitative comparison examples of our method with other unsupervised segmentation methods. Different colors represent different segmentation regions. It can be seen from these figures that our proposed algorithm segments meaningful regions from the original input image without processing and reduce the appearance of small fragments.

The visual segmentation results on MS-COCO and Pascal VOC2012 datasets are shown in figure 5. The subjective visual results demonstrate that our proposed method can produce consistent and complete objects for images with simple scenarios and suppress the background regions well. For images with complex content, our method can segment the visually meaningful regions in intrinsic.

V. CONCLUSION

In this paper, we have presented an unsupervised image segmentation method based on clustering and GCN. We use high-level semantically representative maps to describe the
source image. These maps keep the original attributes and resolution of the initial features. Our method uses a graph structure to establish the dependency relationship for sub-regions and applies the GCN to deliver contextual information between graph nodes. In the training process, we use the CNN model to automatically generate the initial labels, enabling the method to generate the segmentation region automatically without specifying the number of segments in advance. The proposed architecture does not need any training data. We conduct extensive experiments on the BSDS500, Pascal VOC2012, and MS-COCO datasets to evaluate our method. The experimental results demonstrate that our method can achieve better performance.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431–3440.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 5, pp. 898–916, May 2011.

[3] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes challenge: A retrospective,” Int. J. Comput. Vis., vol. 111, pp. 98–136, Jun. 2014.

[4] T. N. Kipf, “Deep learning with graph-structured representations,” Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 2017.

[5] Y. Yu, C. Fang, and Z. Liao, “Piecewise flat embedding for image segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1368–1376.

[6] T. Ilyas, A. Khan, M. Umraiz, and H. Kim, “SEEK: A framework of transfer learning for order statistic image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 2, pp. 335–347, Feb. 2019.

[7] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. ICLR, 2017, pp. 249–256.

[8] S. Liu, M. Yang, J. Yang, and Z. Tu, “GraphNets: A unifying deep learning framework for graphs,” in Proc. ICLR, 2019, pp. 1–9.

[9] R. M. H. Nguyen and M. S. Brown, “Fast and effective l0 gradient minimization by region fusion,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 208–216.

[10] X. Xia and B. Kulis, “W-Net: A deep model for fully unsupervised image segmentation,” 2017, arXiv:1711.08506.

[11] Y. Yu, C. Fang, and Z. Liao, “Piecewise flat embedding for image segmentation,” in Proc. ECCV, 2018, pp. 1869–1886.

[12] L. Zhou and W. Wei, “Unsupervised image segmentation using sparse graph cuts,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 3045–3053.

[13] J. Winn, and A. Zisserman, “The Pascal visual object classes challenge: A retrospective,” Int. J. Comput. Vis., vol. 111, pp. 98–136, Jun. 2014.

[14] T. N. Kipf, “Deep learning with graph-structured representations,” Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 2017.

[15] Y. Yu, C. Fang, and Z. Liao, “Piecewise flat embedding for image segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1368–1376.

[16] T. Ilyas, A. Khan, M. Umraiz, and H. Kim, “SEEK: A framework of transfer learning for order statistic image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 2, pp. 335–347, Feb. 2019.

[17] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. ICLR, 2017, pp. 249–256.

[18] S. Liu, M. Yang, J. Yang, and Z. Tu, “GraphNets: A unifying deep learning framework for graphs,” in Proc. ICLR, 2019, pp. 1–9.

[19] R. M. H. Nguyen and M. S. Brown, “Fast and effective l0 gradient minimization by region fusion,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 208–216.

[20] X. Xia and B. Kulis, “W-Net: A deep model for fully unsupervised image segmentation,” 2017, arXiv:1711.08506.

[21] L. Zhou and W. Wei, “Unsupervised image segmentation using sparse graph cuts,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 3045–3053.

[22] J. Winn, and A. Zisserman, “The Pascal visual object classes challenge: A retrospective,” Int. J. Comput. Vis., vol. 111, pp. 98–136, Jun. 2014.

[23] T. N. Kipf, “Deep learning with graph-structured representations,” Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 2017.

[24] Y. Yu, C. Fang, and Z. Liao, “Piecewise flat embedding for image segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1368–1376.

[25] T. Ilyas, A. Khan, M. Umraiz, and H. Kim, “SEEK: A framework of transfer learning for order statistic image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 2, pp. 335–347, Feb. 2019.

[26] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. ICLR, 2017, pp. 249–256.

[27] S. Liu, M. Yang, J. Yang, and Z. Tu, “GraphNets: A unifying deep learning framework for graphs,” in Proc. ICLR, 2019, pp. 1–9.

[28] R. M. H. Nguyen and M. S. Brown, “Fast and effective l0 gradient minimization by region fusion,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 208–216.