

Received January 30, 2020, accepted February 13, 2020, date of publication February 17, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974496

DIC: Deep Image Clustering for Unsupervised Image Segmentation

LEI ZHOU^{1,2} AND YUFENG WEI³

¹School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

²Shanghai Engineering Research Center of Assistive Devices, University of Shanghai for Science and Technology, Shanghai 200093, China

³School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China

Corresponding author: Lei Zhou (zmbhou@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61906121.

ABSTRACT Unsupervised segmentation is an essential pre-processing technique in many computer vision tasks. However, current unsupervised segmentation techniques are sensitive to the parameters such as the segmentation numbers or of high training and inference complexity. Encouraged by neural networks' flexibility and their ability for modelling intricate patterns, an unsupervised segmentation framework based on a novel deep image clustering (DIC) model is proposed. The DIC consists of a feature transformation subnetwork (FTS) and a trainable deep clustering subnetwork (DCS) for unsupervised image clustering. FTS is built on a simple and capable network architecture. DCS can assign pixels with different cluster numbers by updating cluster associations and cluster centers iteratively. Moreover, a superpixel guided iterative refinement loss is designed to optimize the DIC parameters in an overfitting manner. Extensive experiments have been conducted on the Berkley Segmentation Database. The experimental results show that DCS is more effective in aggregating features during the clustering procedure. DIC has also proven to be less sensitive to varying segmentation parameters and of lower computation costs, and DIC can achieve significantly better segmentation performance compared to the state-of-the-art techniques. The source code is available on <https://github.com/zmbhou/DIC>.

INDEX TERMS Unsupervised segmentation, deep image clustering, deep clustering subnetwork, iterative refinement loss, overfitting training.

I. INTRODUCTION

Object segmentation is a challenging problem in the field of computer vision and it has been widely applied in areas such as object recognition and image classification. Generally speaking, object segmentation methods can be divided into three categories, unsupervised, semi-supervised and fully supervised. The unsupervised methods, such as K-means [1], EM [2], FH segmentation [3], Active contour [4], normalized cut [5], meanshift clustering [6], MLSS [7] or SAS [8], are implemented without prior knowledge about the images. In the semi-supervised methods [9]–[12], users can label pixels as foreground or background with interactive segmentation approaches. User input can locate where the object is (location information), and colour and texture information contained in the scribbles provide prior knowledge about what the object is. Fully supervised segmentation methods

are mostly designed for detecting and extracting specific semantic object categories in images and accurately labelled training dataset is required. Currently, deep learning-based semantic segmentation solutions have also achieved significant improvement and attracted a lot of attention [13]–[17].

In this paper, we focus on studying the research topic related to unsupervised image segmentation. A typical unsupervised segmentation algorithm always contains two parts: feature extraction from images' pixels and dividing an image into nonoverlapping regions by pixels clustering, such as the details described in methods such as normalized cut [5], MLSS [7] and SAS [8]. For example, a segmentation framework based on bipartite graph partitioning is designed to aggregate multi-layer superpixels in SAS [8]. In MLSS [7], a semi-supervised learning strategy is applied to generate pairwise affinities based on the sparse graph constructed on pixels and over-segmented regions. Then the pairwise affinities are applied to the spectral segmentation algorithms. However, the performance of those methods may suffer from

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin¹.

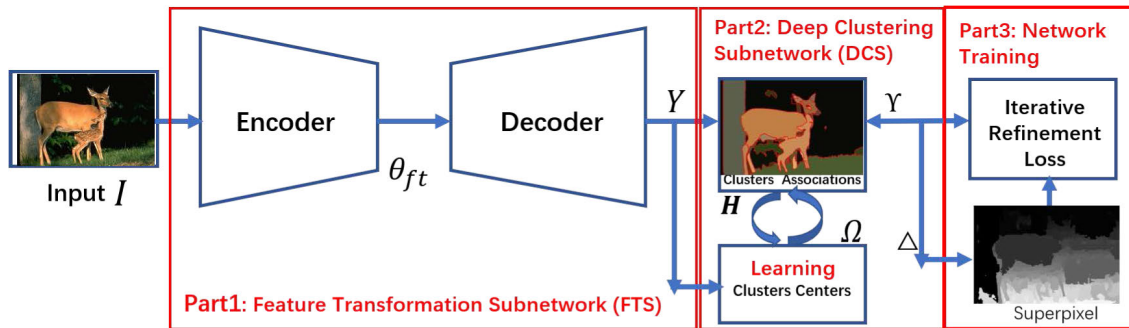


FIGURE 1. The illustration of the proposed DIC framework for unsupervised image segmentation. DIC consists of a FTS and a DCS and DIC is trained by an iterative refinement loss.

two principal drawbacks: being sensitive to the segmentation parameters such as cluster numbers and the whole flowchart is complex, which can not be optimized jointly. Neural networks have also been applied to solve the unsupervised segmentation problem. However, the existing framework such as W-Net [18] involves techniques such as designing complex loss functions. Hence, the training procedure is sophisticated.

We argue that a practical unsupervised segmentation method should exhibit characteristics such as adaptivity for incorporating additional cues and easiness for joint algorithm optimization. Encouraged by neural networks' flexibility and their ability for modelling complex patterns, a novel unsupervised segmentation framework based on neural network is proposed. Firstly, a deep image clustering (DIC) network is designed. DIC contains a feature transformation subnetwork and a trainable deep clustering subnetwork. An autoencoder based network architecture is applied to transform the pixel information of images to high-dimensional features. Then a deep clustering subnetwork implemented by iteratively calculating the cluster centers and cluster associations iteratively is proposed. Secondly, superpixels are taken as the grouping cues and a superpixel guided iterative refinement loss is designed to train DIC effectively. Finally, DIC is optimized via backpropagation in an overfitting manner on a single image's basis. The main contributions of the paper are summarized as follows and the flow chart of our approach is shown in Figure. 1.

- We propose 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.
- We propose a simple and effective superpixel guided iterative refinement loss for optimizing the DIC parameters in an end-to-end way. DIC can be optimized on a single image's basis in an overfitting manner.
- We achieve highly competitive results on Berkley Segmentation Databases with performance gains from 0.8319 to 0.8407 in PRI, and from 0.1779 to 0.1390 in GCE, and from 11.29 to 10.18 in BDE.

The remainder of the paper is organized as follows. The related work is introduced in section II, the proposed framework is introduced in section III and the experimental results are presented in section IV.

II. RELATED WORK

A. UNSUPERVISED SEGMENTATION

Many unsupervised segmentation methods have been proposed recently, such as mean-shift (MS), k-means [6], normalized cuts (NCuts) [5], Felzenszwalb and Huttenlocher's graph-based (FH) [3], SDTV [19], KM [20], UCM [21], CCP [22], MLSS [7] and SAS [8]. Mean-shift (MS) [6] builds a non-parametric probability distribution in a feature space and applies mean shift filtering in this domain to yield a convergence point for each pixel. Normalized cuts (NCuts) [5] focuses on minimizing the similarity between groups while maximizing the associations within groups. Other methods can be divided into two categories: region-based and contour-based methods. Region-based unsupervised segmentation methods focus on finding the similarity among neighbouring pixels and merge them using features including color, texture, contour or luminance. Superpixels are always taken as important cues for aiding segmentation and one of the typical works is MLSS [7], in which a multi-layer semi-supervised learning scheme is proposed to construct a dense affinity matrix over pixels and superpixels for spectral clustering. Another highlighted work is SAS [8], a novel segmentation framework based on bipartite graph partitioning to is designed to aggregate multi-layer superpixels. Contour-based methods focus on generating segmentation masks via contour cues. In [21], the image segmentation problem is constructed as a contour detection problem. A contour detection using multiscale local brightness, color, and texture is proposed firstly. Then an Ultrametric Contour Map (UCM) is constructed by generating a hierarchical region tree from contours. In CCP [22], a contour-guided color palette (CCP) is designed firstly. Then, it is further fine-tuned by post-processing techniques such as leakage avoidance, fake boundary removal, and small region mergence to generate robust segmentation masks.

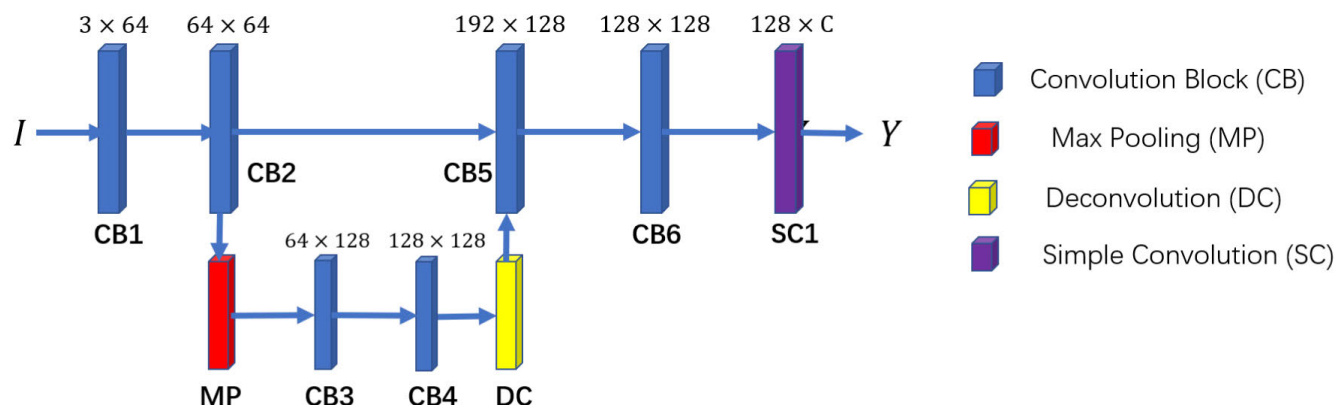


FIGURE 2. The flowchart of the feature transformation subnetwork. The convolution block (CB) consists of a 3×3 convolution, a batch-normalization and a relu. The main task of the max-pooling (MP) with the factor 2 is to downsample the feature by a factor of 2 and deconvolution (DC) is utilized for upsampling features by 2 times. The simple convolution (SC) with kernel size 1×1 is the last layer which generates the output feature. The input channel \times output channel of each component is displayed as well.

B. DEEP LEARNING AND IMAGE SEGMENTATION

In recent years, we have witnessed great progress for deep convolutional neural networks [23], [24] based image segmentation and various methods have been proposed. Deep learning has been applied to solve the problems related to fully supervised semantic segmentation [13], [14], [16], [17], [25]–[27], interactive segmentation [28]–[30] and unsupervised segmentation [18], [31], [32]. Besides the fully supervised semantic segmentation, deep learning has proven to be effective in solving the interactive segmentation problem as well. For example, in [30], the user annotations are converted into interaction maps by measuring distances of each pixel to the annotated locations firstly. Then, the forward pass is performed in a convolutional neural network to generate an initial segmentation map. Moreover, a backpropagating refinement scheme (BRS) is designed to correct the mislabeled pixels. Neural network has been applied to solve the unsupervised segmentation problem as well. In W-Net [18], a segmentation framework based on autoencoder is proposed and a k-way pixel-wise prediction is generated by the autoencoder, then the whole network is optimized by a reconstruction loss and a normalized cut loss jointly. Furthermore, techniques such as condition random field smoothing and hierarchical segmentation are used to refine the segmentation results. However, these methods may suffer from several drawbacks in applications: (1) The segmentation framework is of high computation cost; (2) The training procedure for optimizing the network parameters is sophisticated; (3) Additional cues such as superpixels cannot be incorporated into the framework adaptively.

In order to design a more effective and practical neural network based unsupervised segmentation framework, a deep image clustering module is proposed in this paper. DIC utilizes a light-weight network architecture to reduce the computation cost. Furthermore, a superpixel guided iterative refinement loss is designed to optimize the DIC parameters in a simpler way. Hence, the unsupervised segmentation network can be optimized easily in an overfitting manner.

III. THE PROPOSED METHOD

In this section, the framework for unsupervised image segmentation is proposed. Firstly, we introduce the deep image clustering model in subsection III-A, which consists of two modules: a subnetwork for feature extraction and a deep clustering subnetwork. Then, we present the superpixel guided iterative refinement loss in subsection III-B. Finally, the overfitting training protocol is described in subsection III-C to optimize the network parameters in an end-to-end way.

A. DEEP IMAGE CLUSTERING MODEL

1) NETWORK ARCHITECTURE FOR FEATURE TRANSFORMATION SUBNETWORK (FTS)

We use an autoencoder architecture and the skip connection for constructing the feature transformation subnetwork (FTS). The CNN for feature extraction is composed of a series of convolution layers interleaved with batch normalization (BN) and ReLU activations. The architecture of the feature transformation subnetwork is as shown in Figure. 2, FTS consists of six convolution blocks, one max-pooling operation, one deconvolution operation and a simple convolution operation. We use max-pooling, which downsamples the input by a factor of 2, after the 2nd convolution block to increase the receptive field. Then the 4th convolution block outputs are upsampled by deconvolution and concatenated with the 2nd convolution block outputs to pass onto the 5th convolution block. After the 6th convolution block and the simple convolution block, feature Y with dimension C is generated. We use 3×3 convolution filters with the number of output channels set to 64, 128 or 192 in each block, except the last CNN layer which outputs C channels. This CNN architecture is chosen for its simplicity and efficiency. Other network architectures are also conceivable for such algorithm. The resulting C dimensional features Y can be taken as coarse cluster associations. In order to aggregate the features more effectively, Y will be passed onto the following deep clustering module that iteratively updates the pixel-clusters associations and cluster centers for Γ iterations.

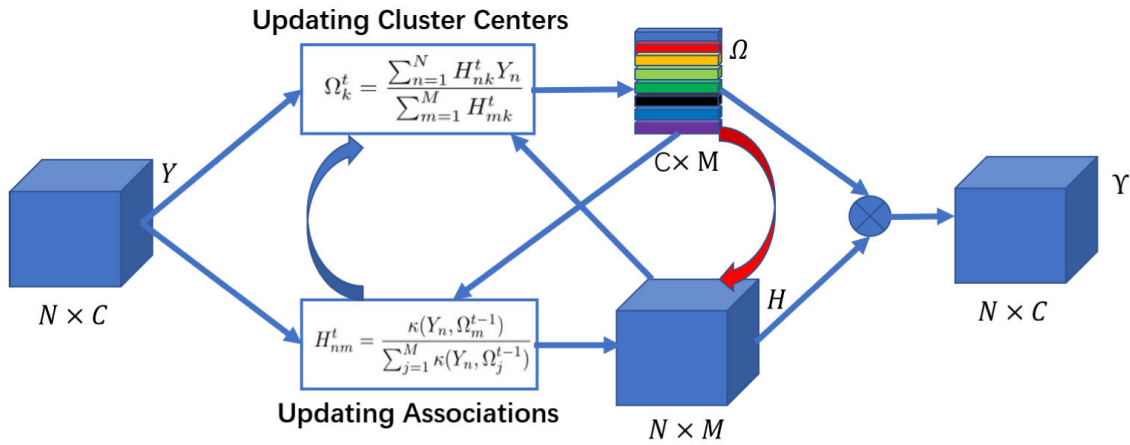


FIGURE 3. The flowchart of the deep clustering subnetwork. DCS contains two iterative steps: calculating cluster associations H and updating cluster centers Ω .

2) TRAINABLE DEEP CLUSTERING SUBNETWORK (DCS)

Firstly the extracted feature Y is flattened to the dimension $N \times C$, where $N = H \times W$, H is the height of image, W is the width of image and C is the channel number or superpixel number (SPN). Then a neural network based clustering procedure is designed. The cluster centers Ω are defined as the initializations for feature clustering. Assuming the cluster centers are defined as $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_M\}$, M is the number of default clusters and Ω_i is with dimension $C \times 1$. Then an iterative procedure is applied for adaptive feature clustering. In the t -th iteration of DCS, the associate rate H^t with dimension $N \times M$ is defined to calculate the similarity between the spatial features and respective centers Ω^{t-1} :

$$H_{nm}^t = \frac{\kappa(Y_n, \Omega_m^{t-1})}{\sum_{j=1}^M \kappa(Y_n, \Omega_j^{t-1})}, \quad n \subseteq \{1, \dots, N\}, \quad m \subseteq \{1, \dots, M\}. \tag{1}$$

where κ represents the general kernel function. We simply take the the exponential inner dot $exp(a^T b)$ as the kernel function. For implementing the neural network, the operation for updating associations in the t -th iteration is formulated as:

$$H^t = softmax(Y(\Omega^{t-1})^T) \tag{2}$$

With the updated association H^t , then the cluster centers Ω^t can be estimated using the weighted summation of Y . Hence Ω_k^t is defined as:

$$\Omega_k^t = \frac{\sum_{n=1}^N H_{nk}^t Y_n}{\sum_{m=1}^M H_{mk}^t}. \tag{3}$$

Then Ω^t can be generated respectively. Based on the above formulations, the associations H^t and cluster centers Ω^t are updated within Γ iterations. Once the iteration procedure is implemented, the final associations H^Γ and cluster centers Ω^Γ are used to construct the aggregated features Υ . The aggregated features are formulated as:

$$\Upsilon = H^\Gamma (\Omega^\Gamma)^T \tag{4}$$

As Υ is constructed from a compact basis set, it has the low-rank property compared with the input Y . It is obviously that the iterative updating procedure is free of parameters. The whole pileline of DCS is as illustrated in Figure. 3.

Once the aggregated feature Υ is obtained, the final cluster assignment Δ_n for each pixel is set by selecting the dimension that has the maximum value of Υ_n using the following rule:

$$\Delta_n = arg \max_j (\Upsilon_{n,j}), \quad j \subseteq \{1, \dots, C\}. \tag{5}$$

Then C can be interpreted as the cluster number in the following subsections.

B. SUPERPIXEL GUIDED ITERATIVE REFINEMENT LOSS

In the setting of unsupervised image segmentation, there does not exist reliable supervision information. The training protocol of DIC is formulated as a classification task in a self-training manner, in which the superpixel guided grouping cues are taken as the principal supervision information. The intuition behind is that pixels in the same superpixels tend to be assigned with the same cluster number and the superpixels can also serve as the cues for refining the object boundaries. Hence a superpixel guided iterative refinement loss is defined for network training. Firstly, the constraints which favor the rule that cluster labels should be the same for those of neighboring pixels are enforced by superpixels. K superpixels $\{S_k\}_{k=1}^K$ are extracted from the input image I , where $\{S_k\}$ denotes a set of indices of pixels that belong to k -th superpixels using the technique such as MCG [21], superpixels with higher quality can generate more reliable supervision. Hence all of the pixels in each superpixel are guided to have the same cluster label in the training procedure. In iteration $t - 1$, Δ^{t-1} is refined by the superpixels so as to enforce the label consistency constraint. Assuming S_p is the superpixel that contains pixel n , then a histogram is constructed by counting the superpixel assignment

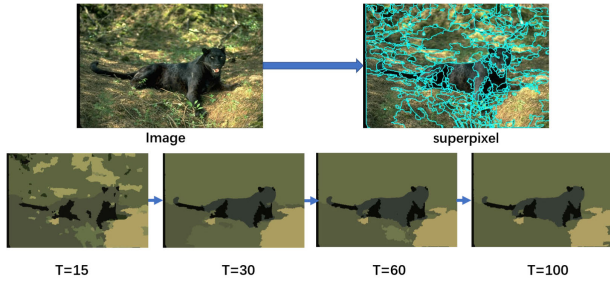


FIGURE 4. Illustration of the iteration clustering process.

$\Delta_i^{t-1}, i \subseteq \{S_p\}$ and a superpixel level histogram is built as:

$$Hist_p(\Delta_i^{t-1}) = Hist_p(\Delta_i^{t-2}) + 1, \quad i \subseteq S_p. \quad (6)$$

Finally, the refined superpixel is defined as the following rule:

$$\tilde{\Delta}_n^{t-1} = \arg \max_j Hist_p(j), \quad j \subseteq \{\Delta_i^{t-1} | i \in S_p\}, \quad n \in S_p. \quad (7)$$

where $\{\Delta_i^{t-1} | i \in S_p\}$ stands for the unique label set defined in $Hist_p$.

Then the refined superpixel $\tilde{\Delta}^{t-1}$ is taken as the supervision for training the DIC network in iteration $t - 1$. Similarly, Δ^t will be generated and it can be refined subsequently as the supervision for the t -th iteration. Given the assignment Δ^t and the refined superpixel $\tilde{\Delta}^t$, the iterative refinement loss is defined to optimize the DIC network using cross entropy:

$$L_{itef}^t = - \sum_{i=1}^N \Delta_i^t \log \tilde{\Delta}_i^t, \quad (8)$$

The iterative clustering procedure is as illustrated in Figure. 4, we can see that the cluster assignment is updated and refined iteratively.

C. OVERFITTING TRAINING FOR UNSUPERVISED IMAGE SEGMENTATION

Encouraged by the works such as deep image prior [33] or deep decoder [34], an overfitting manner is used for training the DIC network. Different from many current practices using large scale datasets, the overfitting training protocol is simpler and the network parameters are not required to be stored. Given a target image as the input, the training procedure will be described as below: (1): In the forward process of the t -th iteration, Y^t is generated by the feature transformation subnetwork with parameter Θ_{ft}^{t-1} . (2): The deep clustering subnetwork runs for Γ iterations to aggregate the features adaptively. Hence the feature Υ^t is obtained and the superpixel Δ^t is generated. (3): Then the refined superpixels $\tilde{\Delta}^t$ is generated. In the backward process, $\tilde{\Delta}^t$ is taken as the supervision with the iterative refinement loss. The parameter Θ_{ft}^t of the feature transformation subnetwork is updated according to the back propagation rule using the iterative refinement loss L_{itef}^t . The detailed training and inference procedures are introduced in Algorithm 1.

Algorithm 1 The Algorithm for Overfitting Training of DIC

- 1: **Input:** I , the RGB image, iteration numbers T and Γ .
- 2: **Output:** Δ , the cluster assignment.
- 3: Generate the superpixels $\{S_i\}_{i=1}^K$.
- 4: Initial the network parameter Θ_{ft}^0 .
- 5: **for** $t = 1$ to T **do**
- 6: Extract the feature Y^t using the feature transformation subnetwork with initialization Θ_{ft}^{t-1} .
- 7: Initialize the cluster center with Ω^0 .
- 8: **for** $q = 1$ to Γ **do**
- 9: Generate the associations H^q based on Y^t and Ω^{q-1} according to Eq. (2).
- 10: Update cluster center Ω^q according to Eq. (3).
- 11: **end for**
- 12: Generate Υ^t based on H^Γ and Ω^Γ according to Eq. (4).
- 13: Generate cluster Δ^t according to Eq. (5).
- 14: Generate the refined cluster $\tilde{\Delta}^t$ according to Eq. (7).
- 15: Take $\tilde{\Delta}^t$ as the supervision and the network parameters Θ_{ft}^t are optimized using the iterative refinement loss defined in Eq. (8).
- 16: **end for**
- 17: Δ^T is taken as the final cluster assignments for image I .

IV. EXPERIMENTAL RESULTS

In this section, we first introduce the experimental setup in subsection IV-A. Then, we compare the empirical results of our DIC with other state-of-the-arts in subsection IV-B. Moreover, analysis on the iteration number T and cluster numbers for overfitting training is presented in subsection IV-C and IV-D. The role of the deep clustering subnetwork is evaluated in subsection IV-E and the computational cost is reported in subsection IV-F. The analysis on the application for semantic segmentation is presented in subsection IV-G.

A. EXPERIMENTAL SETUP

The segmentation results on two Berkley Segmentation Databases (BSDS300 and BSDS500) [35] which consists of 300 and 500 natural images respectively, are reported. To quantitatively evaluate the segmentation results, five criteria are used: 1) Probabilistic Rand Index (PRI) [36]; 2) Variation of Information (VoI) [37]; 3) Global Consistency Error (GCE) [35]; 4) Boundary Displacement Error (BDE) [38]; and 5) Segmentation Covering (SC). The segmentation performance is better if PRI and SC is large and the other three are smaller compared to the ground truths [8]. In the implementation of the deep clustering module, Γ is set as 3 according to the cross-validation experiments. In the training procedure, The training epoch T is set as $T = 100$, the learning rate is set as 5×10^{-2} and the momentum is set as 0.9.

B. COMPARISON WITH STATE-OF-THE-ART METHODS

In order to evaluate the proposed method DIC comprehensively, we compare the average scores of the DIC's

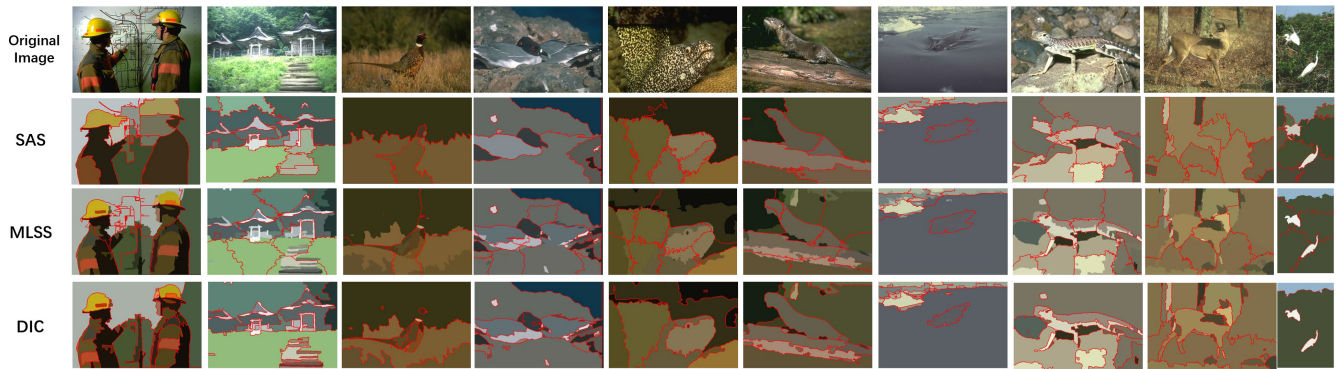


FIGURE 5. The visual comparison between DIC and other state-of-the-arts, such as MLSS, SAS.

with sixteen benchmark algorithms, such as Ncut [5], Mean-shift [6], FH [3], JSEG [39], Multi-scale Ncut (MNcut) [40], NTP [41], SDTV [19], KM [20], gPb-owt-ucm [21], MLSS [7], SAS [8], W-Net [18] and CCP [22] on BSD300 firstly. Similar to the strategy proposed in MLSS [7] or SAS [8], the optimal Image scale (OIS) is selected for segmenting images in the Berkley Segmentation Database. OIS means that the cluster number is selected optimally for each image. Seven settings as SPN=20,M=16; SPN=50,M=32;SPN=90,M=32;SPN=120, M=32; SPN=160, M=64;SPN=200, M=64;SPN=300, M=64 are used to generate segmentation results for each image and then the best segmentation masks are selected. The scores of different methods for comparison are collected from [8], [20], [22] and the segmentation results are reported in Table. 1, with the best results highlighted in bold for each criterion. We can see that DIC achieves the highest PRI 0.8407, the lowest BDE 10.18, the second-lowest VoI and the second-lowest GCE scores compared to other methods. In terms of VoI, gPb-owt-ucm outperforms DIC slightly. CCP achieves a GCE of 0.127 which is slightly better than 0.139 of DIC. Compared with neural network based W-Net [18], DIC achieves 0.03 point gain in terms of PRI even if the post-processing procedure is not applied. The performance improvement is primarily owed to the deep clustering module for feature aggregation and DIC’s ability to incorporate additional cues such as superpixels. Moreover the segmentation results on BSDS500 are reported as well. DIC can obtain the best SC as 0.66 and PRI as 0.864 compared with methods (reported in Table.2) such as Ncut [5], Mean-shift [6], MLSS [7], gPb-owt-ucm [21],SF [42] and W-Net [18]. When compared with gPb-owt-ucm, DIC achieves higher SC and PRI because it can aggregate similar superpixels more effectively to generate fewer superpixels in a self-guided manner. However the value of VoI will increase due to a smaller amount of superpixels of DIC.

The visual comparison is illustrated in Figure. 5 as well in which the segmentation maps generated by SAS, MLSS and DIC are displayed. We can find that

TABLE 1. Performance evaluation of the proposed method against other state-of-the-arts on the Berkley Segmentation Database 300 (BSDS300).

	PRI	VoI	GCE	BDE
Ncut	0.7242	2.9061	0.2232	17.15
Mean Shift	0.7958	1.9725	0.1888	14.41
FH	0.7139	3.3949	0.1746	16.67
JSEG	0.7756	1.8217	0.1989	14.40
MNcut	0.7559	2.4401	0.1925	15.10
NTP	0.7521	2.4954	0.2373	16.30
KM	0.76	2.4	X	X
SDTV	0.7758	1.8165	0.1768	16.24
ASC	0.7738	2.0292	0.2225	16.09
gPb-Hoiem	0.724	3.149	NA	14.795
TBES	0.80	1.76	NA	NA
W-Net	0.81	1.71	NA	NA
gPb-owt-ucm	0.81	1.68	NA	NA
CCP	0.8014	2.4723	0.1270	11.29
MLSS	0.8146	1.8545	0.1809	12.21
SAS	0.8319	1.6849	0.1779	11.29
DIC (proposed)	0.8407	1.7491	0.1390	10.18

TABLE 2. Performance evaluation of the proposed method against other state-of-the-arts on the Berkley Segmentation Database 500 (BSDS500).

	SC	PRI	VoI
Ncut	0.53	0.80	1.89
Mean Shift	0.58	0.81	1.64
MLSS	0.60	0.84	1.59
W-Net	0.62	0.84	1.60
SF	0.65	0.851	1.43
gPb-owt-ucm	0.65	0.862	1.41
DIC (proposed)	0.66	0.864	1.63

DIC works better in merging similar pixels and separating diverse regions by learning from local image patterns adaptively.



FIGURE 6. Five images are selected to illustrate the procedure of loss convergency. (a) are original images and (b) are the generated segmentation masks.

TABLE 3. Performance evaluation of iteration number T in the overfitting training procedure with $SPN=100$ and $M=32$ on $BSDS300$. TC stands for the time cost measured in seconds.

T	50	100	200	300
PRI	0.8038	0.8108	0.8094	0.8101
VOI	2.0667	1.9905	0.1963	1.8966
GCE	0.1569	0.1519	0.1535	0.15923
BDE	11.62	11.37	11.35	11.42
TC(s)	7.2	15.0	30.1	44.7

C. EVALUATION OF ITERATION NUMBER T FOR OVERFITTING TRAINING

DIC is optimized using an overfitting manner on a single image's basis. There is a trade-off between the iteration epoch and the segmentation performance. If more iterations are used, the computation costs will increase, otherwise the training procedure may not converge. An experiment is conducted to evaluate the effect of different iteration numbers T with $SPN=100$ and $M = 32$ on the segmentation performance, and the evaluation results are displayed in Table. 3. As shown in Table. 3, PRI of 0.8038 can be achieved when $T = 50$. If $T = 100$, PRI can be boosted to 0.8108 and the segmentation performance gains can be witnessed for the measures VoI, GCE and BDE as well. When T is set as a value larger than 100, only marginal performance gain can be witnessed, but the computation cost will increase greatly. When T is small, similar superpixels have not been merged successfully, probably because the algorithm will fail to converge with small T . It's observed that the overfitting training procedure will converge in around 100 iterations. Larger T may increase the computation cost dramatically, but the segmentation performance has been improved marginally. Hence $T = 100$ is set as the default setting in the following experiments.

Moreover the training convergence analysis of the overfitting training procedure is presented as well. Firstly, five images are selected and the corresponding segmentation results are displayed in Figure. 6. Then the curves of losses defined in Eq.(8) for the overfitting training procedure within 150 iterations are displayed in Figure. 7. It can be witnessed in Figure. 7 that, the loss will decrease

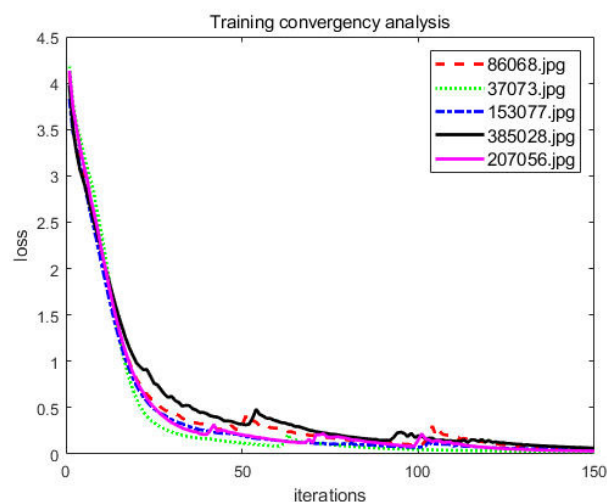


FIGURE 7. The training convergency analysis corresponding to images in Figure. 6 is displayed.

slowly when $T > 50$ and the overall losses will converge in around 100 iterations, which can proof the observation in Table. 3 that the segmentation performance will change slightly when $T \geq 100$.

D. ROBUSTNESS TO DIFFERENT CLUSTER NUMBERS

In this subsection, the segmentation performance conditioned on different cluster number C is evaluated. As shown in Table. 4, the performance of SAS will drop sharply when the cluster number increases, while the DIC is less sensitive to the number of superpixels. We find that DIC can generate relatively satisfactory results by learning from local image patterns adaptively even if the default cluster number is large. On the contrary, the performance of SAS will drop sharply as the cluster number C increases due to the phenomenon that consistent regions may be divided into small regions when the cluster number if large. It indicates that the adaptive clustering mechanism of DIC tends to converge to optimal solutions by merging similar pixels into one cluster adaptively. The visual comparison is as shown in Figure. 8, we can see that the performance of DIC is more reliable when the

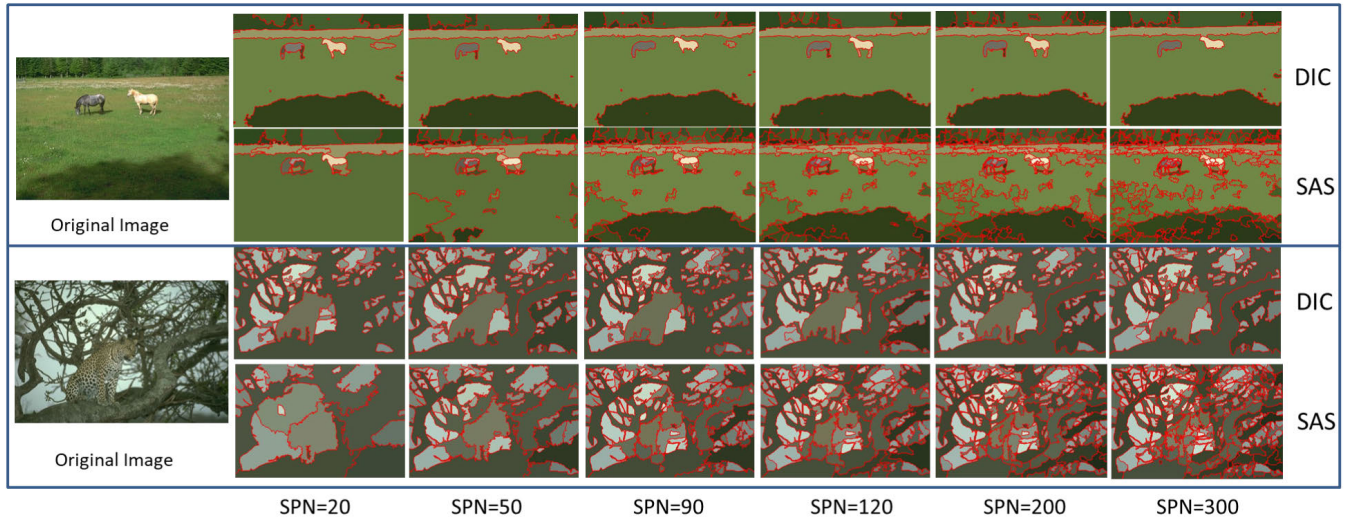


FIGURE 8. The visual comparison between the segmentation performance of SAS and DIC with various superpixel numbers.

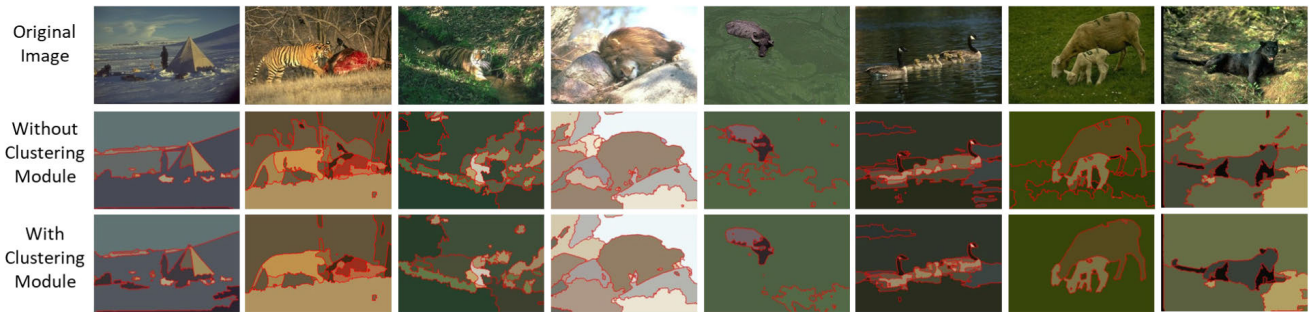


FIGURE 9. Illustration of the role of the deep clustering subnetwork. The first row demonstrates the original images, and the results without and with deep clustering module are listed in the second and third rows respectively.

TABLE 4. Performance evaluation of SAS and DIC with different superpixel numbers (SPN) on BSDS300. M is the default cluster number defined in Eq. (1).

	PRI	VoI	GCE	BDE
SAS SPN=20	0.8015	2.0615	0.1843	12.48
SAS SPN=50	0.7841	2.6450	0.1406	12.85
SAS SPN=90	0.7683	3.1573	0.1131	13.26
SAS SPN=120	0.7590	3.4474	0.1020	13.44
SAS SPN=200	0.7476	3.9640	0.0837	13.64
SAS SPN=300	0.7392	4.3732	0.0727	13.80
DIC SPN=20, $M=16$	0.8039	1.8294	0.1837	11.55
DIC SPN=50, $M=32$	0.8140	1.9084	0.1638	11.37
DIC SPN=90, $M=32$	0.8093	1.9870	0.1537	11.17
DIC SPN=120, $M=32$	0.8097	2.0163	0.1526	11.34
DIC SPN=160, $M=48$	0.8088	2.0270	0.1511	11.40
DIC SPN=200, $M=64$	0.8090	2.0551	0.1485	11.40
DIC SPN=300, $M=64$	0.8081	2.0705	0.1416	11.34

setting of superpixel numbers changes. Even if SPN is set to a large value such as 300, DIC can also assign the same cluster number to the pixels with similar color or texture

cues by extracting more effective features and by adaptively features aggregating. By contrast, the segmentation errors may increase for SAS when the cluster number grows, which is fundamental because the representation ability of the features extracted by the traditional spectral analysis technique is limited and the k-means clustering is a lack of the mechanism for aggregating similar clusters accommodatively.

E. THE ROLE OF DEEP CLUSTERING SUBNETWORK

The deep clustering subnetwork is one of the major contributions in the proposed method and DCS works as a module for elaborately clustering. In this subsection, the role of DCS is evaluated. When DCS is not used, a softmax operation is applied for cluster number assignment as a replacement. According to the comparison and analysis in the above subsection, we set $SPN=50$ and $M = 32$ to evaluate the role of DCS. The quantitative evaluation is listed in Table. 5. When DCS is not used, the PRI will drop from 0.814 to 0.8064 and VoI will increase from 1.9084 to 1.9890, which indicates that DCS is more effective in aggregating pixels with similar features in the deep clustering process than the



FIGURE 10. The visual illustration of the unsupervised segmentation results of DIC and the semantic segmentation results of Deeplabv2-VGG-DIC-SCCN generated by images in PASCAL VOC 2012 dataset.

TABLE 5. Performance evaluation of method with different superpixel numbers (SPN) on BSDS300. wo stands for without.

	PRI	VoI	GCE	BDE
SPN=50 wo DCS	0.8064	1.9890	0.1635	11.41
SPN=50 with DCS	0.8140	1.9084	0.1638	11.37

TABLE 6. The Average Time Costs Measured in Seconds Corresponding to Different Superpixel Number for Processing Images in Berkley Segmentation Database. MLSS and SAS run on intel i7-6700k CPU with 32G RAM, and DIC runs on a GTX1080 GPU with 32G RAM.

SPN	20	50	90	120	200	300
MLSS(CPU)	21.1	31.5	53.9	67.8	128.3	275.5
SAS(CPU)	9.9	15.4	24.3	35.7	75.2	153.8
DIC(GPU)	13.3	14.9	15.5	15.9	16.6	21.5

simple softmax operation. The role DCS is also visually illustrated in Figure. 9. It's evident that the pixels with similar or diverse color or texture can be merged or separated by DCS with higher accuracy, while the simple softmax operation tends to generate over-merged segmentations (see the 1st, 3rd and 4th examples in Figure. 9) or always fails to merge pixels with similar features (see the 5th, 6th and 7th examples in Figure. 9).

F. ANALYSIS OF THE COMPUTATION COST

The computation cost of DIC is evaluated in this subsection. Two algorithms MLSS and SAS are selected for comparison and the results of time cost comparison are illustrated in Table. 6. For DIC, the iteration number Γ of DCI is set as 3, the overfitting training epoch T is set as 100 and M is

set according to the superpixel numbers described in Table. 4. MLSS and SAS run on intel i7-6700k CPU with 32G RAM using Matlab R2016a, and DIC runs on a GTX1080 GPU with 32G RAM using Tensorflow 1.5.0. It's obvious that DIC and SAS are faster than MLSS in general and we can find that DIC is comparable to or faster than SAS when the superpixel number is more significant than 20. When the SPN increases, the computation costs of MLSS and SAS will increase dramatically, while the computation cost of DIC changes slightly. For example, DIC takes around 21.5s for segmenting an image with size 321×481 on average when SPN=300, while SAS takes 153.8s and MLSS takes 275.5s. DIC is almost 13 times faster than MLSS. The results indicate that neither the segmentation performance nor the computation cost of DIC is sensitive to the parameters such as cluster number, while the costs of segmentation framework based on spectral analysis will increase greatly as the SPN raises. Moreover, the computation cost of DIC can be reduced easily by controlling the training epochs T for different applications according to the analysis in Table. 3. Even if DIC runs on a 1080 GPU while MLSS and SAS can only work on CPU, it is obvious that neural network based framework has higher potential for being optimized with the development of optimization algorithms and hardware, compared with the traditional spectral analysis based frameworks.

G. APPLICATION FOR SEMANTIC SEGMENTATION

In order to evaluate the quality of unsupervised clusters of DIC more comprehensively, the generated unsupervised superpixels are utilized for semantic segmentation. The joint training and inference framework proposed in [17] which uses the superpixels to improve the segmentation per-

TABLE 7. Overall accuracy on PASCAL VOC 2012 test dataset initialized with VGG. The IoU of twenty categories and the mean IoU are presented. Especially, the comparison with other semantic segmentation methods is presented.

Method	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
DeconvNet + FCN + CRF [25]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
GMF [43]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
Deeplabv2-VGG-DenseCRF [44]	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
Deeplabv2-VGG-GCRF [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	73.8
CRF-RNN [15]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
Deeplabv2-VGG-SCCN [17]	89.8	51.1	87.4	60.4	75.0	86.1	81.0	84.1	35.3	85.6	63.1	84.4	85.2	80.0	81.5	59.3	84.4	59.7	77.3	68.1	74.9
Deeplabv2-VGG-DIC-SCCN	90.0	52.2	87.9	60.8	75.3	86.3	81.6	84.5	34.3	85.8	65.2	84.2	85.6	80.6	82.0	61.0	84.4	58.9	77.4	68.8	75.3

formance is applied for semantic segmentation. Then the DIC superpixels are incorporated into the SCNN of [17]. The experiment is conducted on PASCAL VOC 2012 [46] which is a famous segmentation dataset containing a training set, a validation set and a test set, with 1464, 1449 and 1456 images each. Following common practice, we augment the dataset with the extra annotations provided by [47]. This gives us a total of 10,582 training images. The dataset provides annotations with 20 object categories and one background class. We do not provide evaluation scores such as PRI, GCE or SC on PASCAL VOC 2012 dataset because the ground truths of semantic boundaries are inaccurate due to the dilated boundary pixels. In the experiment, Deeplabv2-VGG [44] is connected with DIC superpixels, SCCN and the pairwise network to construct a model Deeplabv2-VGG-DIC-SCCN. As shown in Table. 7, Deeplabv2-VGG-DIC-SCCN achieves a mean IoU of 75.3¹ on the test set which is higher than Deeplabv2-VGG-SCCN (74.9) and other VGG+CRF based models such as CRF-RNN (74.7), GMF (73.2) and DeconvNet+FCN+CRF (72.5). Compared with the framework built on SLIC superpixels, the mean IoU of Deeplabv2-VGG-DIC-SCCN is 0.4 points higher. This is primarily because the quality of DIC superpixels generated by deep image clustering is better and they can provide more reliable supervision in the training procedure. The reported results also indicate the advantages of the proposed DIC superpixels in boosting the segmentation performance. The visual illustrations of DIC results, semantic segmentation results of images in PASCAL VOC 2012 dataset are displayed in Figure. 10.

V. CONCLUSION

We have presented a novel framework for unsupervised image segmentation based on neural network. One of our major contributions is that a neural network based deep image clustering model is designed for dividing the image space into different clusters by updating cluster centers and cluster associations iteratively. Then another contribution can be summarized as incorporating the low-level superpixels into DIC by

designing an iterative refinement loss to optimize the DIC parameters in an end-to-end way, in which the superpixels can serve as the grouping cues for encoding complex image patterns. Moreover, DIC is utilized to generate superpixels by learning from local image patterns in an overfitting manner. Compared with other unsupervised segmentation methods, DIC is less affected by the segmentation parameter, such as cluster numbers and of lower computation cost. Extensive experimental results on Berkley Segmentation Databases and PASCAL VOC 2012 database have also revealed the superior performance of DIC in both of the quantitative and perceptual evaluation.

REFERENCES

- [1] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k -means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [2] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1026–1038, Aug. 2002.
- [3] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.
- [5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
- [6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [7] T. Hoon Kim, K. Mu Lee, and S. Uk Lee, "Learning full pairwise affinities for spectral segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1690–1703, Jul. 2013.
- [8] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 789–796.
- [9] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2002, pp. 105–112.
- [10] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.
- [11] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 113–132, Apr. 2009.
- [12] L. Zhou, Y. Qiao, Y. Li, X. He, and J. Yang, "Interactive segmentation based on iterative learning for multiple-feature fusion," *Neurocomputing*, vol. 135, pp. 240–252, 2014.

¹<http://host.robots.ox.ac.uk:8080/anonymous/YUT4XH.html>

- [13] 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.
- [14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," 2015, *arXiv:1511.00561*. [Online]. Available: <http://arxiv.org/abs/1511.00561>
- [15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.
- [16] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille, "Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4545–4554.
- [17] L. Zhou, K. Fu, Z. Liu, F. Zhang, Z. Yin, and J. Zheng, "Superpixel based continuous conditional random field neural network for semantic segmentation," *Neurocomputing*, vol. 340, pp. 196–210, May 2019.
- [18] X. Xia and B. Kulis, "W-net: A deep model for fully unsupervised image segmentation," 2017, *arXiv:1711.08506*. [Online]. Available: <http://arxiv.org/abs/1711.08506>
- [19] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 817–824.
- [20] M. B. Salah, A. Mitiche, and I. B. Ayed, "Multiregion image segmentation by parametric kernel graph cuts," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 545–557, Feb. 2011.
- [21] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multi-scale combinatorial grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 328–335.
- [22] X. Fu, C.-Y. Wang, C. Chen, C. Wang, and C.-C.-J. Kuo, "Robust image segmentation using contour-guided color palettes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1618–1625.
- [23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [25] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
- [26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: <http://arxiv.org/abs/1412.7062>
- [27] W. Song, N. Zheng, X. Liu, L. Qiu, and R. Zheng, "An improved U-net convolutional networks for seabed mineral image segmentation," *IEEE Access*, vol. 7, pp. 82744–82752, 2019.
- [28] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 373–381.
- [29] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 616–625.
- [30] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via back-propagating refinement scheme," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5297–5306.
- [31] M. Chen, T. Artières, and L. Denoyer, "Unsupervised object segmentation by redrawing," in *Proc. Adv. Neural Inf. Process*, 2019, pp. 12705–12716.
- [32] W. Liu, L. Wei, J. Sharpnack, and J. D. Owens, "Unsupervised object segmentation with explicit localization module," 2019, *arXiv:1911.09228*. [Online]. Available: <http://arxiv.org/abs/1911.09228>
- [33] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454.
- [34] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," 2018, *arXiv:1810.03982*. [Online]. Available: <http://arxiv.org/abs/1810.03982>
- [35] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE 14th Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.
- [36] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.
- [37] M. Meila, "Comparing clusterings: An axiomatic view," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 577–584.
- [38] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2002, pp. 408–422.
- [39] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
- [40] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2005, pp. 1124–1131.
- [41] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan, "Normalized tree partitioning for image segmentation," in *Proc. CVPR*, Jun. 2008, pp. 1–8.
- [42] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1841–1848.
- [43] R. Vemulapalli, O. Tuzel, M.-Y. Liu, and R. Chellappa, "Gaussian conditional random field network for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3224–3233.
- [44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
- [45] S. Chandra and I. Kokkinos, "Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 402–418.
- [46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
- [47] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 991–998.



LEI ZHOU received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 2008, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2014, under the supervision of Prof. J. Yang. He is currently a Lecturer with the School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, Shanghai. His current research interests include semantic segmentation, image and video compression, and deep learning.

His research group has won six championships in CVPR 2018 and CVPR 2019 CLIC image compression challenges.



YUFENG WEI was born in 2000. He is currently pursuing the degree in automation with the Wuhan University of Science and Technology. In the Wuhan University of Science and Technology, his performance is at the forefront with the Innovation Laboratory for Intelligent Car and Electronic Design. His research interests are image processing, intelligent car, and deep learning.