IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 9, 2020, accepted July 20, 2020, date of publication July 27, 2020, date of current version August 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012039

Image Segmentation Based on Weakly Supervised MKL on Mixed Visual Features

HAIFENG SIMA, JUNDING SUN, MINMIN DU, JING WANG, AND CHAOSHENG TANG
Department of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China

Corresponding authors: Junding Sun (sunjd@hpu.edu.cn) and Jing Wang (wjasmine@hpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602157, in part by the Henan Science and Technology Planning Program under Grant 202102210167, in part by the Henan Science and Technology Innovation Outstanding Youth Program under Grant 184100510009, in part by the Henan University Scientific and Technological Innovation Team Support Program under Grant 19IRTSTHN012, in part by the Young Scholar sponsored and Doctoral Foundation of Henan Polytechnic University under Grant B2016-37, and in part by the Henan Postdoctoral Foundation.

ABSTRACT Weakly supervised learning has outstanding ability to solve classification tasks, and multiformity middle-level visual features provide more abundant discriminant information for meaningful regions. In this paper, we study the integration of the middle-level visual features including homogeneity of superpixels, region objectness and texture map for segmentation. Then, three kernels are exploited to map visual features to high-dimensional space. A few labeled pixels are chosen for training support vector machines(SVMs) in a single image with hybrid kernels. On this basis, the remaining pixels are labeled with classified results of SVMs and refined the segmentation results by merging pre-segments of mean-shift. We perform sufficient experiments on Berkeley datasets and compared them with several excellent segmentation algorithms. Extensive experimental results of the proposed method show superior segmentation performance and expanded tests on PASCAL VOC datasets further validate the effectiveness of the algorithm.

INDEX TERMS Image segmentation, MKL, mixed visual features.

I. INTRODUCTION

The main difficulty of image segmentation technology is still that the semantic gap has not been solved. The bottleneck lies in the lack of learning and application of prior knowledge in the segmentation task. To solve this problem, many studies are explored on knowledge representation or supervised learning to guide the development of image segmentation technology in order to get more accurate segmentation results [1]–[6]. In the field of semantics analysis, segmentation means the annotation for pixel-level. Due to the high cost of annotation for pixel-level segmentation labels, the existing data sets are usually constrained by the lack of annotation examples and class diversity, which limits the segmentation to a small number of pre-defined object categories. How to use the weakly supervised tagged information has become a major challenge to semantic segmentation. Under the influence of supervised learning, many image segmentation methods employ Deep Neural Networks(DNN) to train the segmentation model [7]. The DNN models need a large number of training samples and achieve great success by building a network to learn the underlying features [8], [9]. However, one of the main obstacles to achieve semantic segmentation is the lack of complete and generalized data samples in free and realistic environment. The image contents are complex, diverse and uncertain, thus the patterns and semantics contained in it are difficult to predict [10], [11].

In computer vision, most images show multiple semantic regions, and these regions have different shapes and scales. So, it is necessary to design reasonable, reliable and recognizable feature descriptors for efficient representation of image content. As an important method of acquiring middle-level visual information, superpixels have attracted widespread attention and consequently a series of segmentation strategies and evaluation criteria are produced [12]. The superpixels are the homogeneous regions aggregating the group features of

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Asikuzzaman.

the pixels, which provide more information and knowledge for further image analysis and understanding [13]. In addition, segmentation tasks are gradually developing towards multi-level feature fusion that can improve the performance of segmentation. On the other hand, in addition to the local homogeneity brought by superpixels, another visual feature named objectness contains higher semantics information. It is the prediction information of the object regions estimated by multi-feature cues that provide important guidance for segmentation and classification. Finally, the texture is one of the indispensable feature for segmentation, and it is more robust in dealing with complex regions.

As a shallow network model, SVMs [17] is a convex optimization method to ensure the optimal solution with fewer parameters learning, and it is still an important shallow tool for deep learning. In addition, SVMs can avoid local extremum and gradient dispersion problems. To enhance the ability of similarity measurements, the samples can possess better separation after mapped to high-dimensional feature space through a kernel function, and integrates multiple features well with multiple kernel [18]. The multiple kernel method can map different feature components of heterogeneous data through appropriate single kernel and the data can be expressed in the new combination space more accurately and reasonably. Multiple kernel learning(MKL) can be regarded as an effective information fusion method between feature level and decision level fusion. MKL can naturally fuse these information and each kernel corresponds to different information sources, such as color, texture and edges.

On the basis of the above analysis, we propose an optimization merging framework based on weakly supervised learning with three middle-level visual features. The first kind of visual feature employs three different strategies to make multi-scale superpixels, and employs the internal mean features to get the homogeneous feature in the color space. The second kind of feature unites multi-scale J-image as the texture feature. The third feature map is the regional objectness which evaluating the internal object score of two types of candidate windows that cover objects. Thus the hybrid visual features mapped to MKL and imbedded into SVM classifiers to obtained pre-labels for image pixels, which are used to further refining for optimal segmentation results.

The contributions of this paper are as follows:

1. A framework of multi-kernel learning classification for region refining segmentation based on weakly label information of a single image is proposed. The trained MKL SVMs is helpful for accurate classification of pixels and the results are used for region refining in this model. Weights of different feature kernels is tested sufficiently to obtain the optimal combination of kernel coefficients on BSD500 and the significance of different visual features for classification accuracy is also discussed.

2. A complementary learning space combined three middle-level visual cues from different feature spaces are designed, including superpixels homogeneity, multi-scale texture and objectness. The SVMs that learns from a small number of samples can accurately classify the pixels of single image in our method. Our method provides a novel solution for the further research of unsupervised segmentation.

3. A more proper regional based objectness estimation strategy is proposed. Two types of candidate windows covering objects are defined to aid the computation of the object-proposal score. One type is based on random selection and the other is based on different distribution of superpixels. Then the corresponding objectness feature is employed to improve the accuracy of pixel classification.

The rest of this paper is organized as following. Section 2 introduces the related work, while in Section 3, we represent the proposed merging segmentation method by weakly supervised of MKL on mixed visual features. Experimental results and detailed analysis with comparisons are demonstrated in Section 4. Finally, Section 5 makes a conclusion of the paper. The architecture of the proposed framework is shown in Fig.1.

II. RELATED WORK

A. WEAKLY SUPERVISED SEGMENTATION

Weakly supervised learning has achieved notable success in machine learning task. Its main principle is to learn and build a prediction model through weakly supervision information. Weakly supervised segmentation has become a hot topic for the manual annotation process is completely avoided. It relies on lightweight annotation data such as image category labels. Papandreou *et al.* [14] proposed a model to estimate semantic regions by utilizing annotated bounding boxes or image-level labels. Huang *et al.* [15] proposed a training model starting from the discriminative regions and progressively increase the pixel-level supervision using seeded region growing. Wang *et al.* [16] proposed an iterative bottom-up and top-down framework which tolerates inaccurate initial localization by iteratively mining common object features from object seeds. It bridges the gap between high-level semantic and low-level appearance in weakly supervised semantic segmentation. Most of these models are based on deep network, and weak supervised learning for single image is still rare so far.

B. TEXTURE COMPUTING

JSEG image segmentation provides a method based on color texture [26]. Its segmentation results are more accurate and robust on texture regions. This method is more efficient and feasible than estimating the parameters of the texture model to recognize this homogeneity. The algorithm makes several assumptions about the image. Firstly, each image contains some similar color texture regions. Secondly, in an image, the color value of each region can be substituted by a quantized color. Finally, in the image, the color of the neighborhood is uninterrupted and distinguishable. JSEG algorithm consisted of two main steps: color quantization and spatial segmentation. The purpose of color quantization is to reduce the number of colors in the original color image and

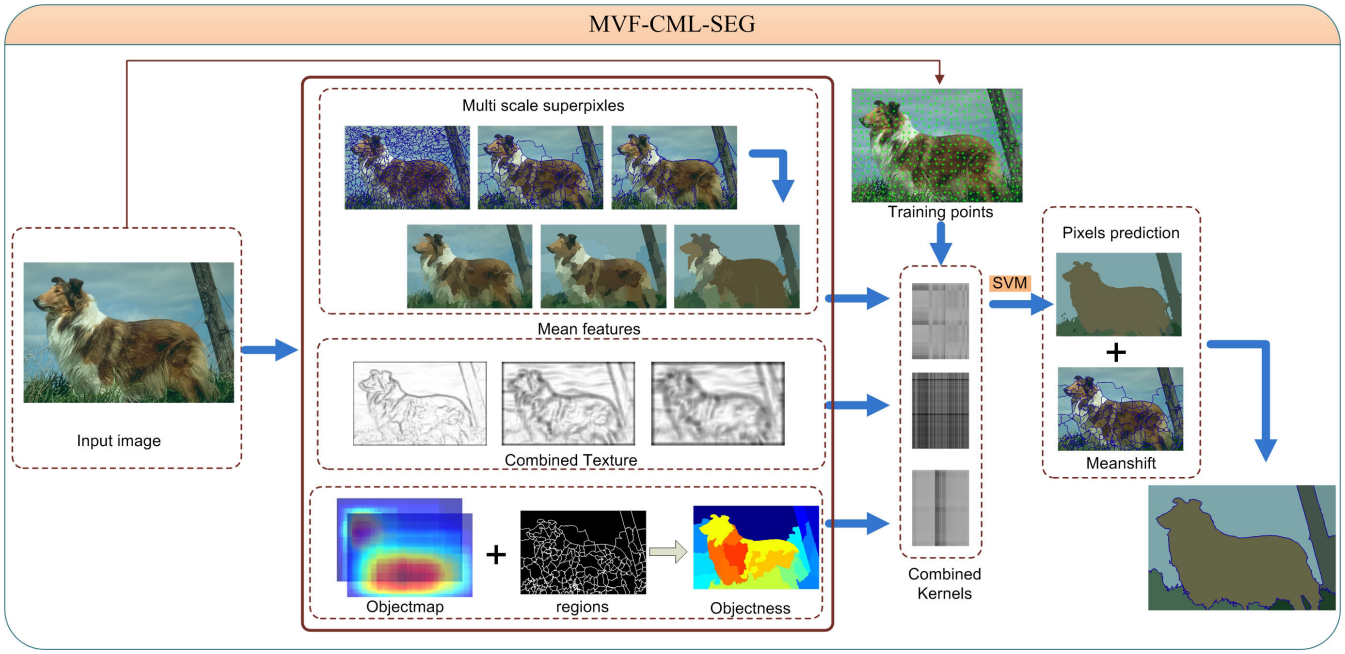


FIGURE 1. The architecture of the proposed method.

decrease the complexity of the algorithm. Generally, 10 to 20 representative colors are extracted in quantization, and only these colors are treated in the texture computing. The key step of quantization is to smooth denoising in LUV space using a non-linear Peer Group Filtering. Then, the general vector quantization algorithm is used to cluster the pixels and get the class-map. Thus the J-image is calculated with circular window templates based on the class-map. The texture features J-image can show the boundary and interior of the region.

C. SUPERPIXELS COMPUTING

Superpixels segmentation is to over-segment an image into a set of connected uniform regions. Different superpixels algorithms present diverse appearance and attributes. Shi and Malik proposed the basic idea of graph-based segmentation, named normalized cut [27]. The graph-cut segmentation [28] adopts minimum spanning tree to cluster pixels and the goal is to make the pixels in the same region as similar as possible and the pixels in different regions as dissimilar as possible. Mean-shift segmentation [29] seek the extreme modality in the joint feature space for clustering, and obtains the reasonable segmentation region by changing the bandwidths. Levinstein *et al.* [30] proposed a growing superpixels model based on geometric flows. SLIC was proposed by Achanta *et al.* [31], clustering super-pixels based on K-means iteration. Liu *et al.* [32] proposed a new objective function based on entropy rate and balance term for calculating superpixels in a graph-based framework. LSC(Linear Spectral Clustering) [33] designs a linear spectral clustering strategy, which maps the pixel values and coordinates explicitly into the high-dimensional feature space, approximates the similarity measure by using the kernel function, and optimizes the objective function of normalized cutting by iterating simple K-means clustering in the feature space.

So far, all these methods have achieved satisfactory results on their established indicators. Therefore, three strategies are employed to calculate different scales of superpixels, and their average performance are used for feature optimization.

D. OBJECTNESS

When the human eye observes an image, the brain vision system will quickly give a semantic information to each region in the image and locates the contour integrity of the target region, which is the ability acquired from long-term training. Alex gives the definition of objectness [34]: the objectness of an image is the possibility that a pixel or an area is contained by an object in the image. The semantic information can be determined according to the details of color, contour, region, texture and so on. Of course, it depends more accurately on the target itself. So if we know the objectness of an image, we can approximately know the semantic information for segmentation. Therefore, we try to introduce objectness compute method to assist segmentation. There are many applications about objectness, such as salient computation [35], segmentation [36] and target detection [37], [38].

Zhang and Zhou [35] redefined the calculation method of objectness, and designed a self-train structural ranker across a group of images to rank the proposals and obtain the proposal-level saliency map. Yao *et al.* [39] computed co-saliency by multi-view spectrum clustering square based on the co-occurrence relation between objectness and superpixels. Jia and Han [40] used weighted linear combination

of image objectness score as feature map to assist salient object detection and the feature map can roughly describe the position of objects in the image. Frintrop *et al.* [41] integrated the saliency system into an object proposal generation framework to obtain segment-based saliency maps and boost the results for salient object segmentation. Carreira and Sminchisescu [42] used bottom-up processes and mid-level cues to produce and rank the objects hypotheses in an image. They extracted object by solving a sequence of constrained parametric min-cut problems (CPMC) on a regular image grid without prior knowledge. Chang *et al.* [38] improved their estimates model of objectness by building new graphical models and energy functions through iterative optimization. Specifically, the energy function includes objectivity, saliency and interaction energy. Jiang *et al.* [43] overlapped all scored object windows as pixels objectness and averaged them in regions level by segmentation to compute regional objectness for salience detection.

E. MULTI-KERNEL LEARNING

Multi-kernel methods are widely used in many fields, such as face recognition [18], visual image classification [19], [20], automatic target recognition [21], multi-spectral and hyperspectral remote sensing data analysis [22], [23] etc. The main reason is that most of these problems need to be solved in pattern recognition are non-linear problems. The kernel method has strong ability to cope with non-linear problems and unique advantages in the case of solving high-dimensional data with small number samples [24].

The MKL methods can be divided into many kinds according to different combination modes, training strategies, multi-scale analyses, etc. The combined kernel model can simplify the process of parameter learning in this paper. The basic idea of combining kernel can be summarized as following:

Suppose that the samples have M kinds of feature representations, a training pixel denoted as $x^m(x^1, x^2 \dots x^M)$. The kernel method maps x^m to high-dimensional space using inner product function $K_m(x_i^m, x_j^m) = \langle \Phi^m(x_i), \Phi^m(x_j) \rangle$, $\Phi_{x_i}^m$ is an implicit mapping function and the most widely used kernel is the radial basis function (RBF) kernel. The multi-class partitioning problem of combined kernel SVM can be expressed as the following optimization problem:

$$\begin{aligned} &\min_{\mu} \min_{\omega,\xi} \{ \frac{1}{2} \sum_{m=1}^{M} \|\omega_m\|^2 + G \sum_i \xi_i \} \\ &s.t.\ y_i(\langle \omega_m, \Phi^m(x_i) \rangle + b) > 1 - \xi_i, \quad \xi_i \geq 0\ i \subset 1, 2 \dots N \end{aligned} \tag{1}$$

In the solution of SVMs, N pairs of examples $(y_i, x_i)_{i=1}^N$ are selected for training. y_i is the label of x_i. ω_m is the weight for component $\Phi^m(x_i)$, ξ_i is the vector of slack variables, G is pre-defined regularization parameter that trades off the margin with error. A composite kernel can be effectively computed by a weighted average of multiple kernels:

$$K_m = \sum_{m=1}^{M} \mu_m K_m(x_i^m, x_j^m) \tag{2}$$

As show in [25], the parameters α_i and the classification rules are estimated by solving the Lagrange equation of Eq.(1) and the resultant decision function is obtained as follows:

$$f(x) = \sum \mu_k \sum \alpha_i y_i K_m(x, x_i) + b \tag{3}$$

III. METHODS

In this section, We define three types of features that contain weak semantic information for training MKL-SVMs to classify image pixels. Then, we refine mean-shift segments with the pre-labeled information in order to produce a better segmentation result. The detailed implementation process is described as following.

A. AVERAGE COLOR FEATURES OF THREE LAYER SUPERPIXELS

Superpixels segmentation can obtain homogeneous regions, and different computing strategies can obtain superpixels of various scales and shapes which imply the class attributes of the pixels. In literature [22], the average feature of superpixels has been used for classification, and it achieves better results. Entropy rate(ER) segmentation [32], Mean-shift [29] and NNG [28] are used to compute homogeneous regions by Entropy rate of random walk, feature space clustering and minimum spanning tree strategy respectively. Therefore, The average color feature of three kinds of superpixels segmentation is defined as one of the input of classification learning. For segmentation results I^{sup} contains M superpixels, and superpixel I^i consists of a set of pixels $x_j = 1, 2 \dots N$, the mean color value M_C^i of the internal pixel x_j is defined as

$$M_C^i = \frac{1}{N} \sum_{1}^{N} C_j \quad x_j \in I^i,\ i = 1, 2 \dots M \tag{4}$$

Each inner pixel is assigned M_C^i, and all assigned pixels constitute the mean features of images I_{mean}^{sup}. The mean color featuer of three-layer superpixels are calculated as

$$M_C^i = \boldsymbol{Mean}(M_C^{i,ER} + M_C^{i,MS} + M_C^{i,NNG}) \tag{5}$$

B. TEXTURE

The value of the pixel in the J-image is expressed by the local J-value, and the calculation of the local J-value of each pixel is described by a class map of the color uniformity of the circular template centered on the pixel x_c: Let the neighborhood of x_c consist of N pixels in the window W, so that $z \in Z$. Z be the set of all N pixels in the class-map, and it is assumed to be divided into C color classes, namely

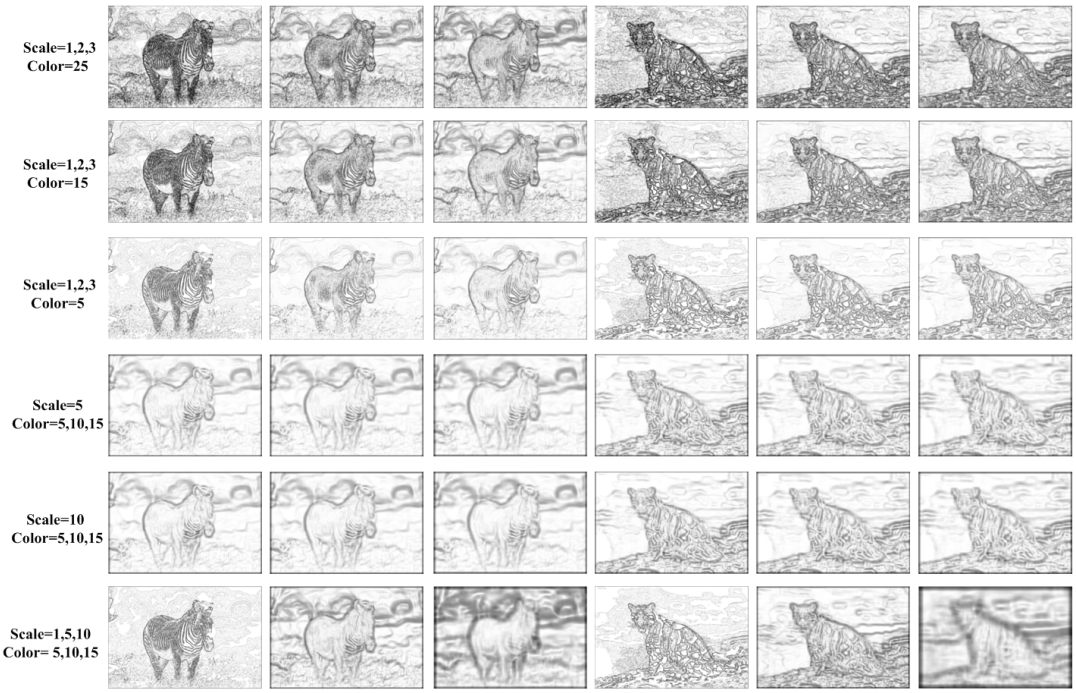


FIGURE 2. The illustration of the J-image of various color quantization and template scales.

$Z_i = 1, 2, .., i, ..C$. The mean value of Z is m and m_i be the mean of the N_i data points of class Z_i:

$$m_i = \frac{1}{N} \sum_{z \in Z_i} z \tag{6}$$

Define S_T as the total variance of the data set Z.

$$S_T = \sum_{z \in Z} \|z - m\|^2 \tag{7}$$

S_W represents the sum of variances for each class, namely intra-class differences.

$$S_W = \sum_{1}^{C} S_i = \sum_{1}^{C} \sum_{z \in Z_i} \|z - m_i\|^2 \tag{8}$$

Then the J-image is calculated by the relationship between S_T and S_W to describe the uniformity of image color:

$$J = \frac{S_T - S_W}{S_W} \tag{9}$$

When the color class of image distributes uniformly in the whole image, the J-value is smaller. If color distribution is uniform and the color class is kept separate from each other, the J-value will be larger. The number of color quantization and window scale is essential for texture calculation. Through observation of Fig.2, we find that larger color number or smaller texture template will bring a more detailed texture. On the contrary, smaller color number or larger texture template will smooth homogeneous texture and bring large-scale texture regions. In order to make texture features more adaptable, we combine color quantization (5,10,15) with window scale (1,5,10) respectively to obtain three scale texture energy maps as the texture components of MKL in this paper:

$$T = [(J_{(S=1,C=5)}, J_{(S=5,C=10)}, J_{(S=10,C=15)})] \tag{10}$$

C. OBJECTNESSS

In [34], objectness mainly estimated the scores of the alternative windows of four cues: multi-scale saliency, color contrast(CC), edge density(ED) and superspixel straddling(SS). The saliency detection only focuses on the extraction of interesting regions and emphasizes the difference of background and foreground, which will influence the segmentation performance. So we only use three other cues to calculate objectness. All detailed definition of *CC*, *ED* and *SS* can be found in [34].

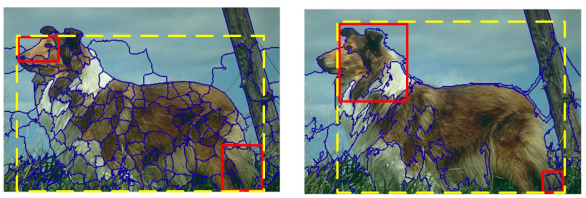


FIGURE 3. The illustration of outer rectangles of W_{seg} (a). MS (b). NNG.

The selection of object proposal windows consists of two parts. The first part is the random sampling windows $W_{default}$ obtained by learning from [43]. On the second part, we use the superpixels of MS and NNG to estimate the alternative windows. The outer rectangles of all superpixels compose a set *Rect*(such as red rectangles in Fig.3), and the outer

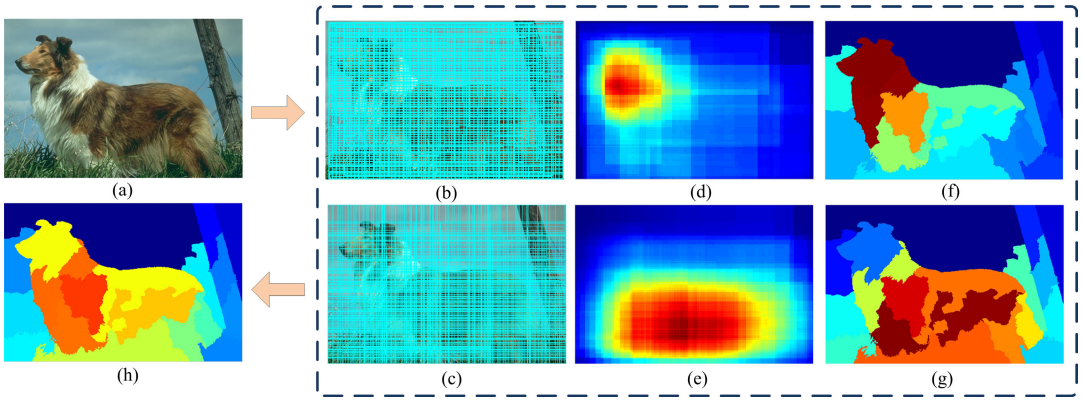


FIGURE 4. The computation process of the objectness.(a). Input image (b).Sampling windows $W_{default}$ (c).Sampling windows W_{seg} (d).Objectness of $W_{default}$ (e).Objectness of W_{seg} (f). Region objectness of $W_{default}$ (g).Region objectness of W_{seg} (h).Hybrid region objectness W_{seg} and $W_{default}$.

rectangles of any combination of two rectangles in *Rect* is more likely to contain a relatively complete object compared with random windows(see yellow rectangles in Fig.3). Therefore, we choose all the outer rectangles of any combination of two rectangles in *Rect* as the candidate windows W_{seg}. Next, we estimate a probability score (the probability of there being a complete object) for the aforementioned two parts of windows to compute pixel-level objectness.

The objectness score of a test window w is defined by the Naive Bayes posterior probability as:

$$P_{\mathrm{w}}(obj|Cue) = \frac{p(Cues|obj)p(obj)}{p(Cues)} \tag{11}$$

where *Cue* is the combined cues set of CC, ED and SS, $p(obj)$ is the priors estimation value of the objectness of training data. All the specific definition of $p(Cues|obj)$, $p(obj)$ and $p(Cues)$ are described in [34].

The objectness value of all sampling windows is computed by Eq.(11), and summed up by pixels locations as pixel-level objectness with Eq.(12).

$$Obj(x) = \sum_{x\in \mathrm{w}}^{\mathrm{w}\in W} P_{\mathrm{w}}(x) \tag{12}$$

where W is the combined sample windows of W_{seg} and $W_{default}$, w is the single window in W and $P_{\mathrm{w}}(x)$ is the score of sample window computed by Eq.(11). The objectness of pixle-wise can not represent the similarity between pixels which shows disorder, so we calculate the average objectness value of regions as the region objectness of all pixels in the corresponding regions of MS segmentation I_{Seg}^{MS}.

$$O(x) = Mean_{x\in R_i} Obj_p(x) \quad R_i \subset I_{MS}^{Seg} \tag{13}$$

In Fig.4, we show the computation process of objectness.

D. PIXELS PREDICTION WITH MKL SVMs

In this paper, a small number of labeled pixels(The training pixels shown in Fig.1) $(x_1, x_2, \ldots x_N)$ with corresponding labels $(y_1, y_2, \ldots y_N)$ are selected uniformly on the image plane to train classification rules. In addition, RBF kernel [34] has been proved to be a good kernel function to improve the classification effect and is widely used. Therefore, SVM classifier combines RBF kernel function with weighted training of three kinds of features in this paper, and they are: mean feature of combined superpixels M_C, texture map T and region-based objectness O. All data are normalized by $x = x/\|x\| * 255$. The three kernels are computed by a weighted sum formula:

$$K_{CML}(y_i, y_j) = \mu_M k_{M_C}(x_i, x_j) + \mu_T k_{tex}(x_i, x_j) + \mu_O k_O(x_i, x_j) \tag{14}$$

where the kernel weights $\mu_M + \mu_T + \mu_O = 1$, and the optimal weights combination of three kernels is tested in 4.1. Thus with the learned α_i and b from Lagrange multipliers of Eq.2, the resultant decision function of the combined kernels can be obtained as:

$$f(x) = \sum \mu_k \sum_{i=1}^{N} \alpha_i y_i K_{CML}(x, x_i) + b \tag{15}$$

E. REGION REFINING

The pixels label predicted by SVM are discrete results, not ideal segmentation results. In this section, these labeled results are used to optimize the merging of regions. It is well known that when the appropriate parameters are selected, the MS segmentation I_{MS} has a good performance in the details and regional integrity. So we use MKL classification results(All pixels of image are classified into L-classes) to refine and merge the superpixels of I_{MS} as the final segmentation. For I_i in I_{MS} with L regions, count the pixels number(N_l) of different labels $l(l \in L)$ in I_i, and the label l_k of max N_l is used to redefine all pixel labels in I_i to get the final image segmentation results.

$$l_k = \arg\max_{l} Conut(x_l) \in I_i \qquad l \subset L \tag{16}$$

$$I_{SEG} = Merge \sum_{1}^{M} (Label(I_i) = l_k) \qquad I_i \subset I^{MS} \tag{17}$$

IV. EXPEREIMENTS

In this section, the experiment includes four sub test to evaluate the performance of the proposed method. The implementation is conducted with MATLAB on a standard computer(Intel i5 Core 2.3GHz CPU with 8G memory) and evaluated on Berkeley Segmentation Data Set (BSD500) [44]. BSD500 is widely used in image segmentation testing, which includes 200 training images, 200 test images and 100 validation images. We use a variety of measurements to evaluate and quantitatively the performance of the algorithm: the Probabilistic Rand Index (PRI) [45], the Variation of Information (VoI) [46], the Global Consistency Error (GCE) [47], and the Boundary Displacement Error (BDE) [48]. The PRI is a measure of likelihood of a pair of pixels being grouped consistently between two segmentations. The VOI is defined as the relative entropy between proposed segmentations and groundtruth segmentations. GCE computes the degree to which two segmentations are mutually consistent. The BDE evaluates the average displacement error of boundary pixels between two segmented images by computing the distance between the pixel and the closest pixel in the other segmentation. Higher PRI score indicates that the algorithm has good performance, and lower scores of VOI, GCE and BDE indicate better performance.

In the first experiment, we test and analyze the segmentation performance on all possible weight combinations of multi-feature kernels, and get the kernel combination of the optimal segmentation results. The second experiment investigates the influence of different number of training samples on the segmentation performance. In the third experiment, we evaluate the segmentation results and compare them with several state-of-the-art methods on the BSD300(a subset of BSD500). In the fourth experiment, the proposed method is performed on a large scale datesets PASCAL VOC 2012 segmentation benchmark, and the results are compared with a deep learning segmentation method.

A. SEGMENTATION TEST OF KERNEL WEIGHT COMBINATION

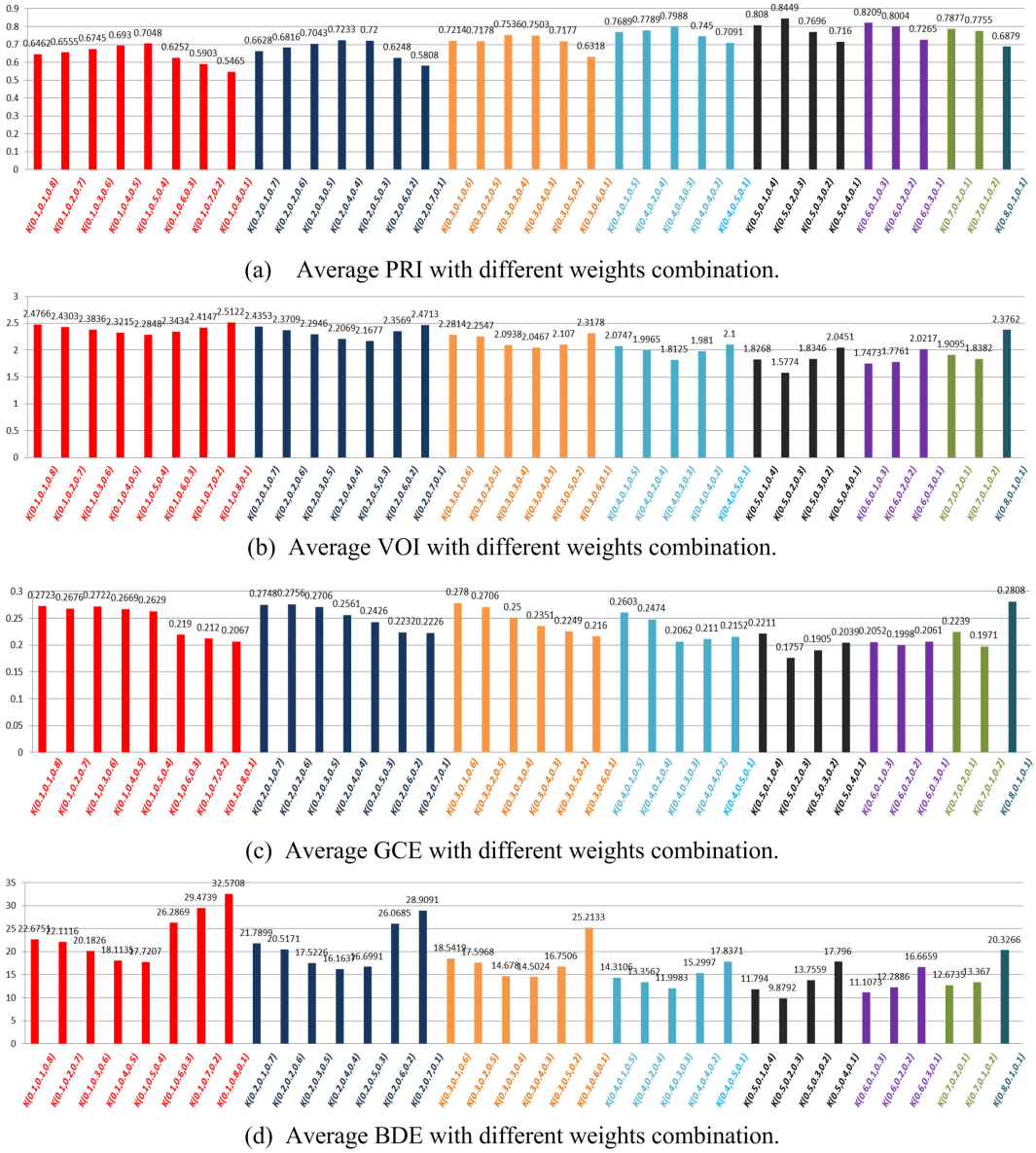
In this section, we investigate the kernel combinations sensitivity of the proposed method to find the optimal combinations based on the BSD500. We display the influence on segmentation performance with various superpixels number of ER method in Fig.7, and it can be seen that the the change of superpixes have little effect on segmentation performance. The superpixel numbers of *MS* and *FH* are uncontrollable, and region refining makes the presegments have little effect on the overall segmentation performance. Therefore, we set the segmentation parameter of *MS* and *FH* with higher boundary recall with fixed *Minarea*. The calculation parameters of three superpixels are set to $ER(K = 400)$, $MS(h_r = 10, h_s = 8, Minarea = 200)$, $NNG(\sigma = 0.6, k = 200, Minarea = 200)$. The goal of multi-scale superpixels is to make them contain more visual content.

Fig.5 displays the test results of all kernel combinations of three kinds of features on BSD500. We arrange all possible combinations in the main sequence of increasing μ_M $(0.1 - 0.8)$. As can be seen in Fig.5, different weights combinations present significant changes in the segmentation accuracy. Increasing the proportion of any single feature will lead to the decrease of the segmentation accuracy, and larger weight of component M_C performs better($\mu_M = 0.4, 0.5, 0.6, 0.7$) compared with other weights. According to our observation of Fig.5, when the weight combination is set to ($\mu_M = 0.5, \mu_T = 0.2, \mu_O = 0.3$), the segmentation performance is the best and the index values reach the optimal ($PRI = 0.8449, VOI = 1.5774, GCE = 0.1757, BDE = 9.8792$). The above results indicate that the performance of the proposed algorithm is sensitive to the change of the weight coefficients, and the mean feature of superpixels has the greatest affection on segmentation accuracy, followed by objectness and texture feature.

Next, we verify the qualitative performance of weight combinations on several specific images. We take the μ_T as a variable to observe the segmentation accuracy with change of weight combinations. Fig.6 shows several image segmentation results under the best weight combinations of three kernels when μ_T increases from 0.1 to 0.8(see in Fig.5). There are corresponding segmentation results and image evaluation indexes PRI and BDE. All μ_T are highlighted in red color in Fig.6. It can be seen that raising the proportion of any weight will not bring ideal visual segmentation effect and index performance. Segmentation results are relatively ideal only when each feature space plays a role. As can be seen from Fig.6, The segmentation results are visually better and more accurate in the case of ($\mu_{M=0.5}$, $\mu_{T=0.2}$, $\mu_{O=0.3}$), which is consistent with the performance of the entire dataset.

B. SEGMENTATION TEST ON VARIOUS TRAINING PIXELS

In this section, we investigate the effect of the number of training samples on the performance of the proposed algorithm. 50-1000 training samples are tested on BSD500 respectively. In Fig.7, we show the changes of four evaluation indexes(PRI,VOI,GCE and BDE) with the increase of the number of training samples. From the overall trend of four indexes, it can be seen that they are improved. A comparison of several test images with different training samples are presented in Fig.8. It can be observed that with the increase of the training samples, the segmentation accuracies of test images are also improved and visually better. On the other hand, with the increase of training samples, object areas to be segmented become more complete, such as the chimpanzees and people on images in the first and fourth rows. When there are enough training samples, more details will appear, such as the pattern of butterfly wings on images in the fifth row. In addition, the location of the training samples will also have an impact on the segmentation results. The location of the small area may be disqualified because there are no relevant samples, such as face and grasses from images in the second and third rows. When the training samples range from 400 to 1000, the change of the four evaluation indexes



(a) Average PRI with different weights combination.

(b) Average VOI with different weights combination.

(c) Average GCE with different weights combination.

(d) Average BDE with different weights combination.

FIGURE 5. **Average Performance with different weights combination $K(\mu_M, \mu_T, \mu_O)$ on four criteria.**

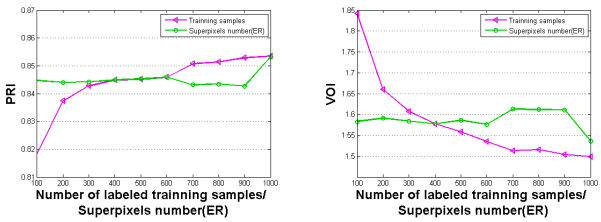
become flat. Therefore, the number of training samples in this paper is set to 400 to improve operating efficiency.

C. SEGMENTATION COMPARISON

In this section, we choose several state-of-the-art segmentation methods for comparison, including JSEG [26], CTM [47], SAS [49], G-graph [50] R-Graph [51], BUP [15] and AIS [16]. Parameter settings of several comparing methods are as follows: The bandwidth parameter of mean shift segmentation is $MS(h_r = 10, h_s = 8, Minarea = 500)$. The parameter $(\sigma = 1.2, K = 400, M = 500)$ of FH is designed to achieve a more complete semantic region of the test image. The JSEG algorithm requires three predefined parameters: 1). quantised-colors = 10, 2). scales = 5, 3), merging-parameter = 0.78. The key parameter of CTM is set to $(\lambda = 0.2)$. The parameters of SAS and GL-graph are provided by the authors in the opened source codes, in which the number of classes K needs to be predefined. R-graph proposed multi-class segmentation strategy by utilizing graph partitions based on eigenvector histogram. There are many kinds of toolboxs being used, including SE, supepixels, kernel density estimation, and we have tested using the default parameters values in the corresponding codes. BUP [15] and AIS [16] are two merging segmentation methods based on superpixels and experiments implemented with 900 superpixels for BUP and self-adaptation for AIS separately.

Firstly, we select seven images from BSD300 for comparison with seven other algorithms and display the visual

FIGURE 6. Visual segmentation examples of different weights combination.



(a) PRI VS Taining samples/ Superpixels number. (b) VOI VS Taining samples/ Superpixels number.

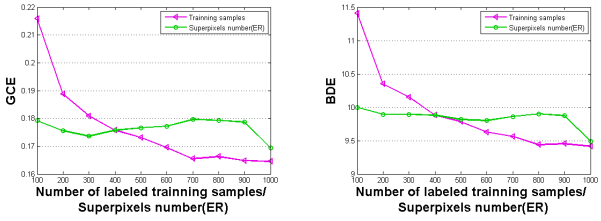


FIGURE 7. Segmentation evaluation curves with various training samples and superpixels(ER) on BSD500.

segmentation effects in Fig.9 and four indexes in Table.2. In Fig.9, the first column is the original image. From column 2 to column 8, the segmentation results of various algorithms are JSEG, CTM, SAS, GL-graph, R-graph, BUM and AIS, and the last column are results of the proposed algorithm. From the perspective of overall visual performance, our segmentation results are relatively complete, and have better consistency with human visual cognitive results in local regions. For example, the overall contours of the objects and background regions are relatively complete in images 48055, 113004, 118020. For complex images with multiple objects and cross-mixing regions, the local segmentation results are visually better than compared algorithms, such as images 65074 and 97033.

From Fig.9, we can observe that JSEG has good adaptability to extract texture regions, but perform badly in integrity. It is prone to produce over-segmentation or under-segmentation regions. Results of CTM are similar to JSEG, but this method takes considerable time to compute textures. SAS and GL-graph are bipartite-graph-based methods work on multi-scale superpixels that can conveniently regroup pixels according to different superpixel results. SAS and GL-graph work better than other methods in regional entirety, but they can not work well in complex regions like in the third row. R-graph depends heavily on the pre-defined graph partition parameters L, and we can see that the fixed number of classes(11) provide in [51]) is not suitable for all image segmentation. The merging process of BUM is started from smaller superpixels, and the edge of segmentation results will appear uneven. AIS uses four types of feature constraint merging process, which is obviously affected by feature weight, and also prone to incomplete merging.

Table 1 reports the comparison of the evaluation indexes of the test image in Fig.9. It is clear that, our method performs the best six times on PRI, five times on VOI, four times on GCE and BDE. To sum up, the segmentation results and visual consistency of our method are significantly better than other algorithms.

The average quantitative evaluation of eight comparison algorithms on BSD300 is shown in Table 2. From the performance on the whole dataset, PRI and VOI indexes of the proposed method are superior to other compared algorithms except BUM and AIS for their superpixels number are optimized, GCE ranks second three and BDE ranks second in the compared methods. The resolution of the test images in BSD are 321 $\times$ 481. The average time consumption of our

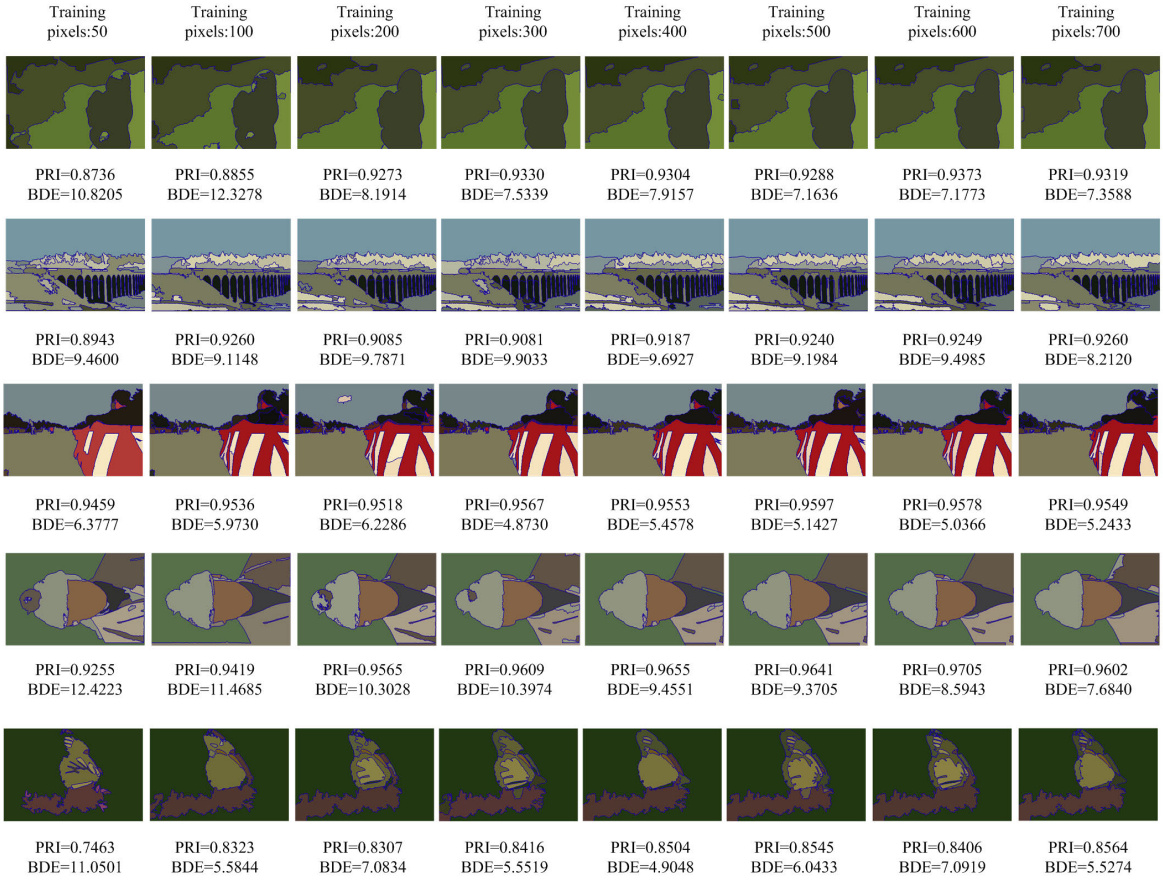


FIGURE 8. Visual segmentation examples with changing training samples.

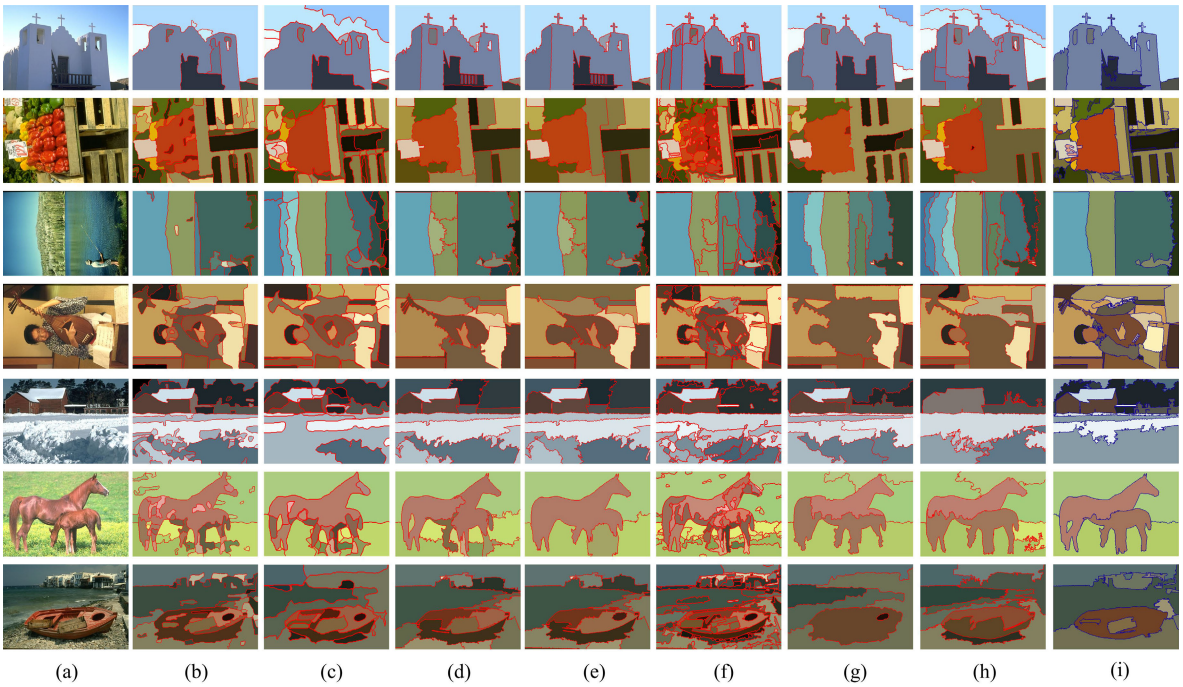


FIGURE 9. Visual comparison of segmentation results with eight algorithms. (a).Input images (b).JSEG (c).CTM (d).SAS (e).G-graph (f).R-graph (g).BUM (h).AIS (i).Ours.

method is 22.8s, including feature extraction, training and region merging. The production of multi-scale superpixels with ER, MS and FH takes 5.8 second, extracting all the visual features takes 7 seconds, MKL and region merging

TABLE 1. The segmentation results comparision of eight images.

No.	index	JSEG	CTM	SAS	G-graph	R-graph	BUM	AIS	Ours(Rank)
24036	PRI	0.8224	0.8646	0.8759	0.8760	0.8862	0.8133	0.7567	0.8906(1)
	VOI	1.5173	1.8437	1.2281	1.2281	1.4639	1.6657	2.2097	1.1092(1)
	GCE	0.1353	0.1755	0.0972	0.0973	0.0732	0.1711	0.1708	0.0724(1)
	BDE	11.160	11.634	9.2182	9.2275	8.1456	10.942	10.737	8.3313(2)
25098	PRI	0.8954	0.8809	0.9013	0.9014	0.8839	0.8559	0.8484	0.9062(1)
	VOI	2.4249	3.4136	1.8810	1.8812	3.1447	2.1745	2.0368	2.1384(4)
	GCE	0.2069	0.2175	0.2035	0.2035	0.1697	0.2101	0.1641	0.2455(6)
	BDE	11.998	12.387	11.215	11.208	10.836	12.087	13.897	9.6294(1)
48055	PRI	0.8952	0.7954	0.9053	0.9053	0.8586	0.8287	0.8234	0.9276(1)
	VOI	1.2090	3.1659	1.2848	1.2847	1.9775	1.7040	1.9662	0.8715(1)
	GCE	0.1067	0.1185	0.1139	0.1139	0.1075	0.1970	0.1691	0.0548(1)
	BDE	10.297	14.454	11.515	11.526	8.4490	19.300	16.036	9.3773(2)
65074	PRI	0.9427	0.9422	0.9420	0.9420	0.9529	0.8073	0.8416	0.9356(6)
	VOI	1.9696	2.6482	1.6061	1.6035	2.0376	2.2369	2.3785	2.1066(5)
	GCE	0.2560	0.2211	0.1821	0.1817	0.2024	0.1502	0.2016	0.2926(5)
	BDE	3.6686	4.5892	5.5878	5.5854	3.9733	7.5400	6.9726	4.2362(3)
97033	PRI	0.7753	0.7609	0.7706	0.7707	0.7426	0.7697	0.7551	0.8207(1)
	VOI	2.4656	3.0302	2.1951	2.1949	3.2403	2.2482	2.1694	1.7042(1)
	GCE	0.1599	0.1349	0.2306	0.2307	0.2269	0.2314	0.3084	0.2019(3)
	BDE	19.583	16.279	22.482	22.207	23.337	16.796	27.932	6.9220(1)
111344	PRI	0.7291	0.7223	0.8088	0.8018	0.7262	0.7440	0.7662	0.8977(1)
	VOI	2.9647	3.7268	1.4839	2.0605	3.3526	1.7983	1.7387	0.7535(1)
	GCE	0.1463	0.1067	0.2192	0.1767	0.0803	0.1975	0.1459	0.0597(1)
	BDE	8.0407	8.2169	5.9001	6.4110	8.5130	8.2605	12.381	2.2077(1)
118020	PRI	0.8776	0.8805	0.8909	0.8909	0.8903	0.8302	0.8895	0.9018(1)
	VOI	2.2463	2.9203	2.0509	2.0510	2.9193	2.0008	1.9742	1.5324(1)
	GCE	0.2843	0.1911	0.2919	0.2919	0.1454	0.2382	0.2671	0.1762(2)
	BDE	7.6198	7.2389	8.7596	8.7582	6.8327	10.648	7.9588	5.8014(1)

TABLE 2. The segmentation results comparison on BSD300.

	JSEG	CTM	SAS	G-graph	R-graph	BUP	AIS	Ours
PRI	0.7756	0.7263	0.8319	0.8384	0.8370	0.8587	0.8359	0.8453
VOI	2.3217	2.1010	1.6849	1.8010	3.4467	1.5947	1.1795	1.5473
GCE	0.1989	0.2071	0.1779	0.1934	0.1342	0.1968	0.1613	0.1720
BDE	14.40	9.42	11.29	10.66	14.09	10.89	10.62	10.33

takes 10 seconds. The time consumption of JSEG(4.7s) and SAS(6.5s) are lower than our methods due to the development tool is C Language. The time performance of our method is close to R-graph(19s) and BUM(21.2s), and better than G-graph(26.8s), AIS(>30s),and CTM(>1 minutes).

D. SEGMENTATION ON PASCAL VOC 2012

Extensive experiments are conducted on PASCAL VOC 2012 datasets, and compared with two state-of-the-art deep learning segmentation methods Mask-RCNN [52] and COB [53]. COB established a single CNN forward pass for multi-scale contour detection and it combined multi-scale oriented contours for hierarchical segmentation. The Mask-RCNN is designed on the base of the region proposal network(RPN). It uses ROIAlign to refine region features of ROI and locates the corresponding binary mask as segmentations by RPN. Both of the two methods are well trained semantic segmentation networks and have excellent performance in target segmentation of various natural scenes. PASCAL VOC benchmark contains 11315 images, and 3812 images are used for test in this section. The most noteworthy feature of PASCAL dataset is that the image contains salient objects regions, and it was mainly used for testing of image and objects segmentation tasks. Fig.10 shows some qualitative results of the three compared algorithms and the overall performance test are shown in Table 3 with four segmentation indexes. From the comparison of segmentation results in Fig.10, it can be observed that our method can reduce false segmentation and produce clear segmentation results and contours. Moreover, the results of large-scale data sets are sufficient to demonstrate the generalization ability of our proposed method for diverse scenarios. From the quantitative results, we can see that the segmentation performance on four standard measurements of the proposed method algorithm is more accurate than that of Mask-RCNN and COB. Therefore, we argue that our strategy to learn the features of a single image is more pertinent than CNN architecture, and obtain accurate segmentations.

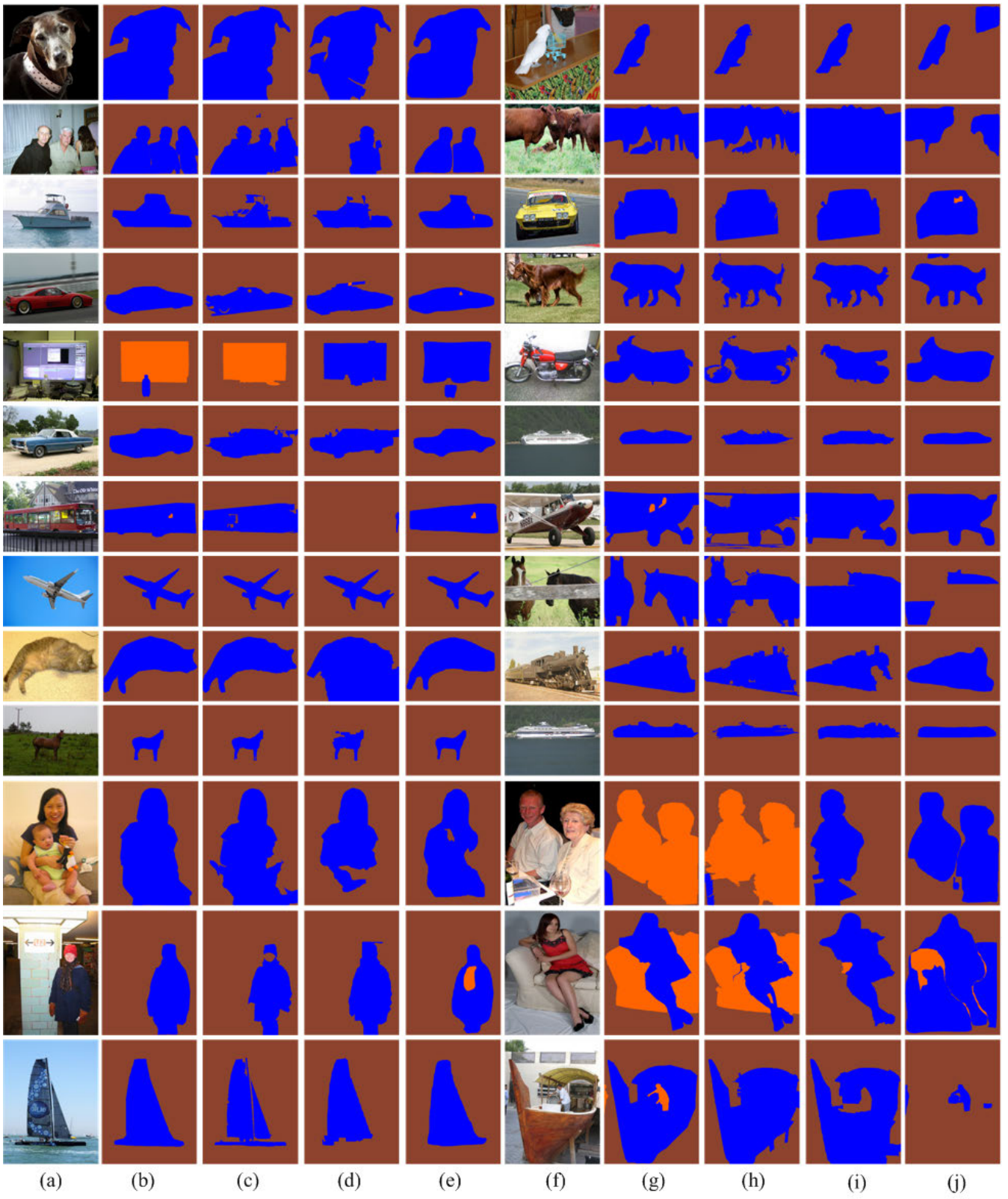


FIGURE 10. Visual segmentation examples on PASCAL VOC arranged by:(a,f).Input image (b,g). Groundtruth (c,h). Results of Ours (d,i). Results of COB (e,j). Results of Mask-RCNN.

TABLE 3. The segmentation results comparison on PASCAL VOC.

	COB	Mask_RCNN	Ours
PRI	0.7257	0.8113	0.8759
VOI	0.9031	0.7086	0.6463
GCE	0.1137	0.0897	0.1043
BDE	49.41	28.80	26.70

V. CONCLUSION

In this paper, we propose a segmentation strategy in using class labels of pixels from training SVM to merge regions on a single image. It is interesting to seek the best fusion scheme with MKL by combining color, texture and object clues to classify pixels. Thus, the class labels are used to refine the pre-segments results as segmentation. That will be a meaningful step towards weakly supervised image segmentation. Compared with previous state-of-the-art algorithms, we can see that our algorithm further improve the accuracy of segmentation by testing on open datasets. In the future, we are trying to find for more dimensional features and combinatorial optimization, learning guidance segmentation at the super-pixel level, and expect better performance.

REFERENCES

[1] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3992–4000, doi: 10.1109/CVPR.2015.7299025.

[2] F. Liu, G. Lin, and C. Shen, "CRF learning with CNN features for image segmentation," *Pattern Recognit.*, vol. 48, no. 10, pp. 2983–2992, Oct. 2015, doi: 10.1016/j.patcog.2015.04.019.

[3] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan, "STC: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2314–2320, Nov. 2017, doi: 10.1109/TPAMI.2016.2636150.

[4] A. Vezhnevets, V. Ferrari, and J. M. Buhmann, "Weakly supervised semantic segmentation with a multi-image model," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 643–650, doi: 10.1109/ICCV.2011.6126299.

[5] M. Xu, J. Zhu, P. Lv, B. Zhou, M. F. Tappen, and R. Ji, "Learning-based shadow recognition and removal from monochromatic natural images," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5811–5824, Dec. 2017, doi: 10.1109/TIP.2017.2737321.

[6] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to segment under various forms of weak supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, vol. 43, no. 1, pp. 3781–3790, doi: 10.1109/CVPR.2015.7299002.

[7] 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, doi: 10.1109/CVPR.2015.7298965.

[8] G. Bertasius, L. Torresani, S. X. Yu, and J. Shi, "Convolutional random walk networks for semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 858–866, doi: 10.1109/cvpr.2017.650.

[9] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2019, doi: 10.1109/TNNLS.2019.2958324.

[10] X. Mingliang, L. Pei, L. Mingyuan, F. Hao, Z. Hongling, Z. Bing, L. Yusong, and Z. Liwei, "Medical image denoising by parallel non-local means," *Neurocomputing*, vol. 195, pp. 117–122, Jun. 2016, doi: 10.1016/j.neucom.2015.08.117.

[11] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured AutoEncoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018, doi: 10.1109/TIP.2018.2848470.

[12] A. Schick, M. Fischer, and R. Stiefelhagen, "An evaluation of the compactness of superpixels," *Pattern Recognit. Lett.*, vol. 43, pp. 71–80, Jul. 2014, doi: 10.1016/j.patrec.2013.09.013.

[13] H. Sima, A. Mi, X. Han, S. Du, Z. Wang, and J. Wang, "Hyperspectral image classification via joint sparse representation of multi-layer superpixles," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 10, pp. 5015–5038, 2018, doi: 10.3837/tiis.2018.10.021.

[14] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1742–1750.

[15] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7014–7023.

[16] X. Wang, S. You, X. Li, and H. Ma, "Weakly-supervised semantic segmentation by iteratively mining common object features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1354–1362.

[17] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, no. 3, pp. 2491–2521, Nov. 2008, doi: 10.1007/s10846-008-9235-4.

[18] X. Wu, Q. Li, L. Xu, K. Chen, and L. Yao, "Multi-feature kernel discriminant dictionary learning for face recognition," *Pattern Recognit.*, vol. 66, pp. 404–411, Jun. 2017, doi: 10.1016/j.patcog.2016.12.001.

[19] D. Li, J. Wang, X. Zhao, Y. Liu, and D. Wang, "Multiple kernel-based multi-instance learning algorithm for image classification," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 1112–1117, Jul. 2014, doi: 10.1016/j.jvcir.2014.03.011.

[20] H. Wu and L. He, "Combining visual and textual features for medical image modality classification with ℓ_p norm multiple kernel learning," *Neurocomputing*, vol. 147, pp. 387–394, Jan. 2015, doi: 10.1016/j.neucom.2014.06.046.

[21] Y. Guo, H. Xiao, H. Fan, and Y. Zhu, "Multiclass multiple kernel learning for HRRP-based radar target recognition," *Proc. SPIE*, vol. 10443, Jun. 2017, Art. no. 1044306, doi: 10.1117/12.2280252.

[22] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "Classification of hyperspectral images by exploiting spectral–spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015, doi: 10.1109/TGRS.2015.2445767.

[23] L. Gan, J. Xia, P. Du, and J. Chanussot, "Multiple feature kernel sparse representation classifier for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5343–5356, Sep. 2018, doi: 10.1109/TGRS.2018.2814781.

[24] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013, doi: 10.1109/tgrs.2012.2230268.

[25] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, vol. 4, 2004, p. 6, doi: 10.1145/1015330.1015424.

[26] 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, doi: 10.1109/34.946985.

[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Aug. 2000, pp. 888–905, doi: 10.1109/cvpr.1997.609407.

[28] 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, doi: 10.1023/b:visi.0000022288.19776.77.

[29] 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, doi: 10.1109/34.1000236.

[30] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009, doi: 10.1109/tpami.2009.96.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012, doi: 10.1109/TPAMI.2012.120.

[32] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2097–2104, doi: 10.1109/CVPR.2011.5995323.

[33] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1356–1363, doi: 10.1109/CVPR.2015.7298741.

[34] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012, doi: 10.1109/tpami.2012.28.

[35] L. Zhang and Q. Zhou, "Salient object detection via proposal selection," *Neurocomputing*, vol. 295, pp. 59–71, Jun. 2018, doi: 10.1016/j.neucom.2018.01.050.

[36] L. Ye, Z. Liu, L. Li, L. Shen, C. Bai, and Y. Wang, "Salient object segmentation via effective integration of saliency and objectness," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1742–1756, Aug. 2017, doi: 10.1109/TMM.2017.2693022.

[37] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5244–5252, doi: 10.1109/CVPR.2017.557.

[38] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 914–921, doi: 10.1109/ICCV.2011.6126333.

[39] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3196–3209, Jul. 2017, doi: 10.1109/TIP.2017.2694222.

[40] Y. Jia and M. Han, "Category-independent object-level saliency detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1761–1768, doi: 10.1109/ICCV.2013.221.

[41] S. Frintrop, T. Werner, and G. M. Garcia, "Traditional saliency reloaded: A good old model in new shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 82–90, doi: 10.1109/cvpr.2015.7298603.

[42] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3241–3248, doi: 10.1109/cvpr.2010.5540063.

[43] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1976–1983, doi: 10.1109/ICCV.2013.248.

[44] P. Arbeláez, 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, doi: 10.1109/TPAMI.2010.161.

[45] 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, doi: 10.1109/TPAMI.2007.1046.

[46] M. Meilă, "Comparing clusterings: An axiomatic view," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 577–584, doi: 10.1145/1102351.1102424.

[47] 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. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 416–425, doi: 10.1109/ICCV.2001.937655.

[48] 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. (ECCV)*, in Lecture Notes in Computer Science, vol. 2352. Springer, 2002, pp. 408–422, doi: 10.1007/3-540-47977-5_27.

[49] 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, doi: 10.1109/CVPR.2012.6247750.

[50] X. Wang, Y. Tang, S. Masnou, and L. Chen, "A global/local affinity graph for image segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1399–1411, Apr. 2015, doi: 10.1109/tip.2015.2397313.

[51] Z. Zhang, F. Xing, H. Wang, Y. Yan, Y. Huang, X. Shi, and L. Yang, "Revisiting graph construction for fast image segmentation," *Pattern Recognit.*, vol. 78, pp. 344–357, Jun. 2018, doi: 10.1016/j.patcog.2018.01.037.

[52] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969, doi: 10.1109/iccv.2017.322.

[53] K.-K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. Van Gool, "Convolutional oriented boundaries: From image segmentation to high-level tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 819–833, Apr. 2018, doi: 10.1109/TPAMI.2017.2700300.

[54] H. Sima, P. Guo, Y. Zou, Z. Wang, and M. Xu, "Bottom-up merging segmentation for color images with complex areas," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 3, pp. 354–365, Mar. 2018, doi: 10.1109/TSMC.2016.2608831.

[55] X. Xie, G. Xie, X. Xu, L. Cui, and J. Ren, "Automatic image segmentation with superpixels and image-level labels," *IEEE Access*, vol. 7, pp. 10999–11009, 2019, doi: 10.1109/ACCESS.2019.2891941.

HAIFENG SIMA received the B.E. and M.E. degrees in computer science from Zhengzhou University, Zhengzhou, China, in 2004 and 2007, respectively, and the Ph.D. degree in software and theory from the Beijing Institute of Technology, Beijing, in 2015. Since 2007, he has been with the faculty of Henan Polytechnic University, Jiaozuo, China, where he is currently a Lecturer with the School of Computer Science and Technology. His current research interests include pattern recognition, image processing, image segmentation, and image classification.

JUNDING SUN received the B.S. degree in computer application and the M.S. degree in control theory and control engineering from Henan Polytechnic University, Jiaozuo, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer application from Xidian University, in 2005. His major interests are image processing, image retrieval, and pattern recognition.

MINMIN DU is currently pursuing the master's degree in computer technology with Henan Polytechnic University. Her major research interests are image processing and pattern recognition.

JING WANG received the B.S. degree from the Henan University of Science and Technology, China, in 2006, and the Ph.D. degree from the College of Computing and Communication Engineering, Graduate University of Chinese Academy of Science, Beijing, China, in 2012. She is currently an Associate Professor with the School of Computer Science and Techniques, Henan Polytechnic University, Jiaozuo, China. Her research interests include image processing, computer vision, and machine learning.

CHAOSHENG TANG received the Ph.D. degree from Yanshan University. He is currently a Lecturer with Henan Polytechnic University. His major research interests are deep learning, large scale data mining, and complex networks.