Scientific Research Publishing

Int. J. Communications, Network and System Sciences, 2020, 13, 139-159
https://www.scirp.org/journal/ijcns
ISSN Online: 1913-3723
ISSN Print: 1913-3715

An Image Segmentation Algorithm Based on a Local Region Conditional Random Field Model

Xiao Jiang[1], Haibin Yu[2], Shuaishuai Lv[2]

[1]School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou, China
[2]School of Electronics and Information, Hangzhou Dianzi University, Hangzhou, China
Email: jx@hdu.edu.cn, shoreyhb@hdu.edu.cn, lvshuai@hdu.edu.cn

How to cite this paper: Jiang, X., Yu, H.B. and Lv, S.S. (2020) An Image Segmentation Algorithm Based on a Local Region Conditional Random Field Model. *Int. J. Communications, Network and System Sciences*, **13**, 139-159.
https://doi.org/10.4236/ijcns.2020.139009

Received: September 9, 2020
Accepted: September 27, 2020
Published: September 30, 2020

Abstract

To reduce the computation cost of a combined probabilistic graphical model and a deep neural network in semantic segmentation, the local region condition random field (LRCRF) model is investigated which selectively applies the condition random field (CRF) to the most active region in the image. The full convolutional network structure is optimized with the ResNet-18 structure and dilated convolution to expand the receptive field. The tracking networks are also improved based on SiameseFC by considering the frame relations in consecutive-frame traffic scene maps. Moreover, the segmentation results of the greyscale input data sets are more stable and effective than using the RGB images for deep neural network feature extraction. The experimental results show that the proposed method takes advantage of the image features directly and achieves good real-time performance and high segmentation accuracy.

Keywords

Image Segmentation, Local Region Condition Random Field Model, Deep Neural Network, Consecutive Shooting Traffic Scene

1. Introduction

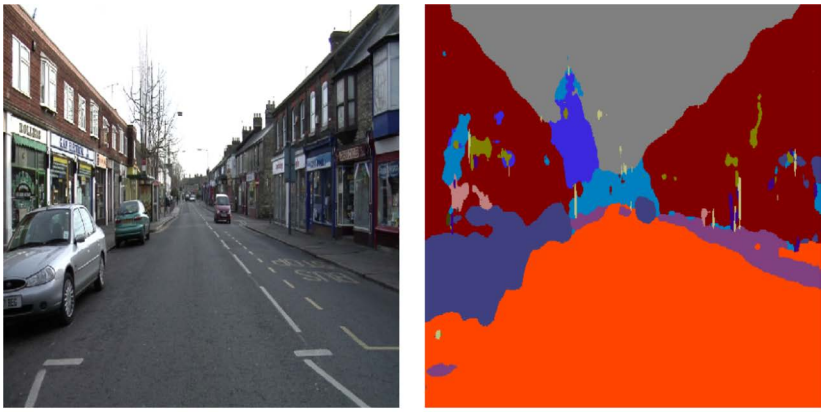
In the past twenty years, deep convolutional neural networks have gradually become a powerful tool for analysing images in various computer vision fields [1] [2] [3] [4]. Recently, convolutional neural networks have achieved good results on image semantic segmentation tasks [2] [3]. Semantic segmentation of images involves machine automatic recognition and segmentation and forms the foundation for understanding the image content [3]. In essence, each pixel is classified in the image. For example, in **Figure 1**, given the streetscape picture on the left as input, the machine will output a graph similar to the image on the right,

which shows the objects distinguished by different colours. Semantic image segmentation plays significant roles in the operation of autonomous vehicles and UAVs as well as wearable devices. An autonomous driving application is shown in Figure 1, where a computer is used to identify objects and assess the road conditions around the car. However, semantic segmentation for autonomous vehicle applications is different from general semantic segmentation because it must process video signals, which are acquired episodically by the car's cameras. Thus, semantic segmentation of images is concerned not only with the accuracy of the segmentation system but must also achieve real-time performance in autonomous vehicle situations.

At present, semantic image segmentation applications are divided into two main directions [1]. One is concerned with the speed of segmentation and usually uses a fully convolutional neural network whose most important feature is that it replaces the fully connected layers in the primitive neural network with convolutional layers to segment the image. The convolutional layers preserve the positional information of the image that the more primitive fully connected structures destroyed. Subsequently, by up-sampling, the output of a fully connected network is returned to its original size [5] [6] [7] [8]. Then, it categorizes each pixel in the image by features to achieve a pixel-level result [7]. This method focuses on the speed of segmentation; it involves only a general convolutional structure and does not need to establish the system mathematical model. Usually, however, the segmentation results produced by a fully convolutional network have poor effects in target boundary areas [9] [10] [11]. This problem occurs because the convolutional process does not alter the space, it simply obtains the relations between image areas, and it is difficult to obtain the dependency relationships at the image pixel level. However, the segmentation process needs to obtain more dependency relations at target boundary locations; otherwise, the inaccurate segmentation can cause the computer to fail to accurately judge the environment around the car (e.g., a fully convolutional neural network is generally poor in situations where pedestrians or other vehicles are close to the car. As shown in Figure 2, the adjacent area effects usually result in a relatively

(a) (b)

Figure 1. Automatic machine image segmentation results (a) The original image; (b) A segmented graph image showing the different objects in the graph using different colours.



(a) (b)

Figure 2. The segmentation result with a fully convolutional neural network. (a) The original image; (b) The segmentation result.

ineffective region-dependent effect for the two cars to the left of the image. This type of inaccurate judgement of critical objects on the road that cause a fully convolution network is difficult to put into use in automatic drive. The other model uses a fully convoluted neural network and a conditional random field model, which pays more attention to the segmentation effect.

The algorithm uses a conditional random field model to optimize the segmentation result of a fully convoluted neural network [12]. First, each pixel of the original image for the node is used to establish a conditional random field. Second, the output of a fully convoluted neural network is used as the value for a unary potential function. Then, the binary potential function's expression is established by establishing Gaussian mixture models. The final segmentation result is obtained by the field inference process.

Conditional random fields are more likely to group pixels with similar locations and colours into the same category. The dependency relationships at the pixel level can be captured by this method, and objects' boundaries can be seen clearly. However, the mean-field inference process in the conditional random field algorithm is similar to the iteration that occurs in a bilateral filter. The resulting high computational complexity makes real-time operation difficult to achieve.

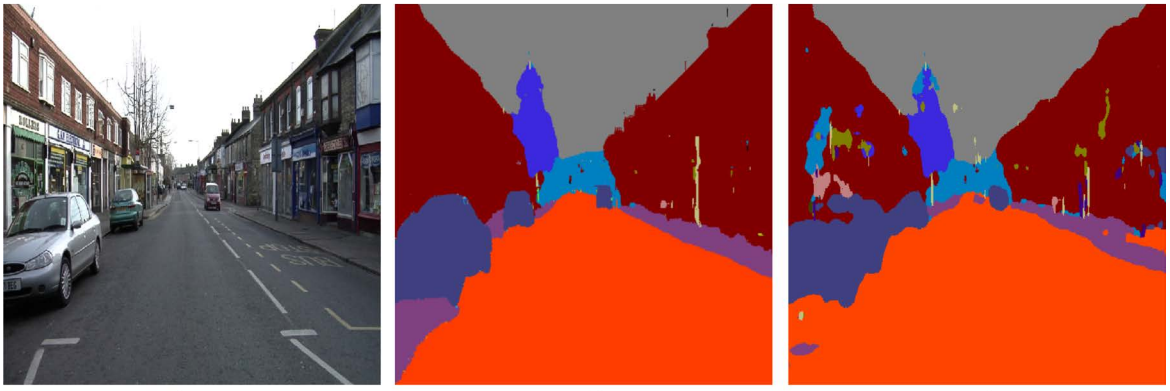
In the first of the above research directions, the representative networks include the fully convolutional network (FCN), SegNet and Unet, among others [5] [13] [14]. The FCN was the first network to introduce convolutional neural networks into the semantic segmentation field. The FCN was also the first to apply a deep neural network to image segmentation as opposed to traditional semantic segmentation [15] [16] [17] [18]. Based on the two advantages of FCNs for semantic segmentation, there are two advantages for semantic segmentation networks based on FCN. 1) FCNs achieve a qualitative improvement in accuracy compared with the traditional semantic segmentation method because the traditional semantic segmentation method uses manually constructed, difficult-to-design but well-rounded image features—a task that is much easier to perform

by automatically training deep neural networks. 2) Because FCNs discard fully connected layers in their structure, they can process images of any size, while general convolution neural networks cannot. There is no need to perform input image size transforms when using an FCN; thus, the they avoid the accompanying distortions of the input images. However, FCNs also have the following shortcomings. 1) Because the convolution process is feature-space invariant, only the relations between locations in the image can be obtained, and it is difficult to obtain pixel level dependency relationships. 2) The pooling process in a convolutional neural network causes a loss of considerable spatial position information. To some degree, an FCN offsets the lost information by using jump connection structures at different scales. Nevertheless, the issue remains. To address this problem, SegNet adopted different polling structures in its FCN. The maximum position before polling is recorded in the polling area. Then, it is put into the same location in the characteristic chart during up-sampling. In this way, SegNet improved the issue. Each coding structure corresponds to a decoding structure, adds image features directly before and after coding, and obtains the summation; in other words, SegNet can integrate multi-scale image features. Overall, the SegNet model captures more object spatial information and improves boundary areas to a certain degree compared with the FCN model. To further improve the segmentation results, Unet improves the use of the encoding and decoding structure in SegNet and also fuses characteristic pictures by direct stacking both before encoding and after decoding. This approach has the following two advantages. 1) The feature information works on segmentation results that can be guaranteed through direct stacking. 2) The quantity of characteristic pictures at different ratios can be selected based on requirements. Although Unet improved the problem of missing object spatial information during the polling process, the defects of the convolution structure's applicability to semantic segmentation have not been completely solved. Because the spatial invariance during the convolution process is stable, the pixel-level relations in images are difficult to obtain. Thus, the segmentation results are insensitive to object boundary information. Consequently, an inevitable problem exists in the application of a fully convolutional network that generally results in low segmentation accuracy when many target objects exist and are discretely distributed. To a great extent, it confines the application of a fully convolutional network in a road traffic scene.

The second research direction for semantic segmentation combines the conditional random field model into the traditional image segmentation algorithm. The conditional random field is used to obtain the dependency relationships between pixels. The representative network structures in this direction are DeepLab v2 and CRFasRNN [19] [20]. The fully convolutional network of DeepLab v2, which is based on the FCN structure, uses spatial pyramid pooling to extend the receptive field during the convolutional process [5], allowing more spatial information to be obtained. Most importantly, DeepLab v2 adopts a fully connected conditional random field model to smooth the segmentation results in a

fully convolutional network, making it highly accurate at obtaining object boundary information and resolving the problem of obtaining pixel-level dependency relationships in previous methods. However, the inferential process in a fully connected conditional random field model carries high computational costs; thus, the use of a probabilistic graph model will also carry high costs. Consequently, this approach is difficult to apply in real-time systems. In addition, the application processes in the fully convolutional and conditional random field models involve two independent steps, which is not in accordance with an end-to-end system structure currently advocated by the industry. This system cannot be fully automated; in other words, it adds the cost and uncertainty of manual intervention. Subsequently, CRFasRNN was proposed to combine the conditional random field model and fully convolutional network interact into a single end-to-end system. The predicted values of the segmentation results of the fully convolutional network are used as a unary potential function value by modifying the magnitude of loss between the minimum conditional random field model results and the input tags to optimize the network parameters. In other words, the deep convolution neural network and conditional random field can function as an end-to-end system. Compared to the DeepLab v2 model, the training process in a conditional random field model takes full advantage of the image features extracted from a neural network, which are not manually designed. Generally, neural networks can extract many features that cannot be manually designed. However, the CRFasRNN network does not optimize the method of accounting for mean-field inference in a conditional random field; complex calculations still occur in this process (it usually requires approximately 800 ms to obtain the segmentation results for one 300 × 300-pixel image frame using a commonly used GPU such as a GTX 1060). Therefore, it is difficult to apply CRFasRNN in systems where real-time performance is required.

In autonomous driving, although the representative networks of the first approach can be applied in a real-time sematic segmentation network, they cannot guarantee sufficient segmentation accuracy. The representative networks of the second approach can guarantee sufficient accuracy but cannot be applied in real time. Thus, this paper attempts to change the way the conditional random field is applied to make it suitable for real time systems. By observing the traffic-scene image, we found that paying too much attention to the accuracy of the segmentation results (e.g., DeepLab, CRFasRNN) is wasteful given the conditions required for real-time monitoring of traffic scenes. Additionally, in traffic scenes, the accuracy improvements after applying the conditional random field are not obvious. In **Figure 3**, the left image is the input picture, the middle image is the result of the fully connected conditional random field, and the right image is the result of the convolutional neural network. By analysing the segmentation results in the convolution neural network, we found that the segmentation results of the fully connected conditional random field approach are not much improved in areas such as the sky, the road and buildings compared to the results



(a) (b) (c)

Figure 3. Segmentation results of different methods (a) Original image; (b) Fully connected conditional random field; (c) SegNet.

obtained by only traditional convolutional neural networks. Those areas have a common feature—they are obviously highly distinguishable. For these high-continuity areas, a traditional convolutional neural network can obtain a good segmentation result. For the not-continuous areas, as shown in **Figure 3**, the three cars are not continuously distributed in the picture, and the car areas are not obviously distinguishable, but the segmentation results obtained from the conditional random field are considerably improved. Therefore, to make use of the conditional random field model effectively, this model applies it only to the areas where it is most beneficial. In this way, the computational complexity can be reduced, and the conditional random field model can be applied to real-time autonomous driving. In addition, simply putting the CRFasRNN into use in a traffic scene neglects the most essential part of the time dimension in the video. It turns a video that is a time-continuous segment into several independent pictures for processing. The connections between frames result in double counting and this approach lacks a connection with time. All these factors may lead to inaccurate segmentation results.

To apply a semantic segmentation network to real-time traffic-monitoring, this paper proposes a new semantic segmentation network that is combined with a probabilistic graph model. This approach not only improves the use of the traditional probabilistic graph model but also considers the frame-to-frame relationships. Compared with other semantic segmentation networks, our method has the following advantages [19] [20] [21].

1) Our method adopts a special way of applying the conditional random field model that first selects the areas that will profit most (the area with the most important object, such as an area containing a person, bicycle or motor vehicle) as the area to be optimized. Then, the conditional random field model is applied in the selected area. This approach can be referred to as a locally connected conditional random field model. This approach limits the calculational complexity by concentrating only on the most pertinent area, which substantially reduces the computational cost.

2) We modify the fully convolutional network structure by replacing the VGG-16 structure in the original DeepLab v2 model by the ResNet-18 structure. Cavitation convolution is used to expand the receptive field during the convolution process [21]. The experiments show that the ResNet structure is better at extracting picture features. Furthermore, cavitation convolution can expand the receptive field of characteristic pictures by preserving the spatial information of pictures [22].

3) We use the improved SiameseFC tracking network to capitalise on the strong correctional information between frames [23]. The segmentation results in the areas of interest (generally, in a conditional random field model, the regional variations of those areas exceed a certain threshold. These areas are considered misclassified areas. Through observation, continuous frame pictures usually generate the same misclassified areas), and the corrected results obtained from the conditional random field model are the defining principle of subsequent video frames. This information was used to modify the misclassified area in nearly the same position from a continuous video frame. This approach takes advantage of the essential feature of video—the continuity between frames. Compared with a pure fully connected convolutional neural network, our approach greatly improves the segmentation results.

4) The input images for SiameseFC are not normally RGB images, but the segmentation results are. The total number of categories ranges from 0 - 255, the grey level value of each pixel. This approach makes it possible to effectively track the areas where additional attention is needed. The segmentation is the result of the action of both the grey image fully connected convolutional neural network and the conditional random field model, which highlight both the spatial and boundary information of target objects. The advantages obtained by this feature result in a segmentation that greatly reduces the operational complexity of the feature extraction process. In addition, using grey images avoids the need for massive colour a priori information of the target object during the traditional network tracking process. The features extracted from the grey images through the deep neural network are more stable and more effective than those extracted from RGB images. Therefore, the samples required to train the network can be obtained easily by clipping a few pictures around the current frame target area.

2. The Proposed Model

2.1. Overview of the Model

The proposed model consists of two parts, each of which plays an independent role in the semantic segmentation network. The first part is LRCRF, which is described in this paper. During the training process, a DeepLab-Resnet 18 network modifies the systematic segmentation results through the produced LRCRF. In addition, loss minimization between the output results and the input labels of LRCRF is used to optimize the segmentation network parameters. A flowchart of LRCRF process is shown in **Figure 4**. The second part of the pro-

posed model performs regional tracking implemented by the improved SiameseFC. The data inputs to the tracking network are the segmentation results from the first part. A few grey image samples clipped from around the misclassified areas (discussed in the preceding section) in the segmentation results are used to train the tracking network. In addition, they are used to track and modify the same misclassifications in consecutive continuous-frame images. A flowchart of the method is shown in **Figure 5**.

2.2. The Local Conditional Random Field Model

The LRCRF model proposed in this paper consists of two parts, as described above. One is DeepLab-Resnet 18, and the other establishes a local conditional random field model.

A rough segmentation result is obtained by the DeepLab-Resnet 18 structure, which is derived from DeepLab v2, but the VGG-16 network in DeepLab v2 is replaced with Resnet-18 [24] [25] [26]. A structure diagram is shown in **Figure 6**.

For any input traffic scene image, DeepLab-Resnet 18 is used to obtain rough segmentation result. The segmentation result is the maximum enclosing rectangles of the pedestrian, bicycle and motor vehicle. These areas are then used as the input to the second structure. The area selection process is shown in **Figure 7**.

The second structure establishes a local area conditional random field model using the following steps. First, the areas recorded above in the original picture as input. For any input area, we treat every pixel as a node; then, we rearrange all the pixels in the regions into a vector. Thus, for any input area X, $X=(x_1,x_2,x_3,\cdots,x_N)$ (where x_i is the pixel value of the i-th point and N is the number of pixels in this area) corresponding to an output area Y, $Y=(y_1,y_2,y_3,\cdots,y_N)$ (where y_2 is the segmentation result of the i-th output area, the range value is L, $L=(l_1,l_2,l_3,\cdots,l_N)$

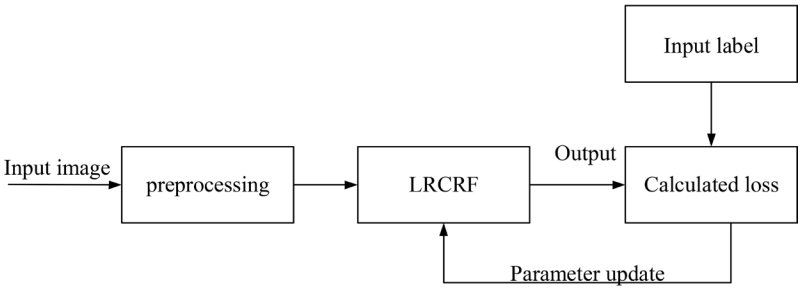


Figure 4. LRCRF training process.

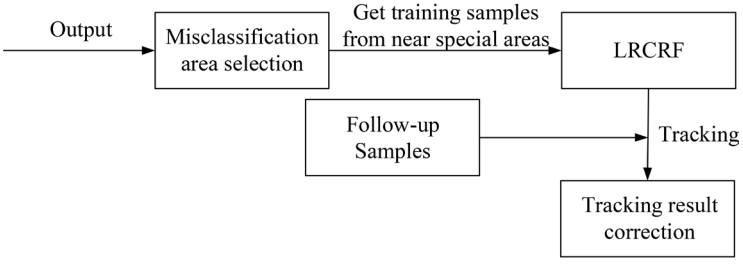


Figure 5. LRCRF regional tracking.

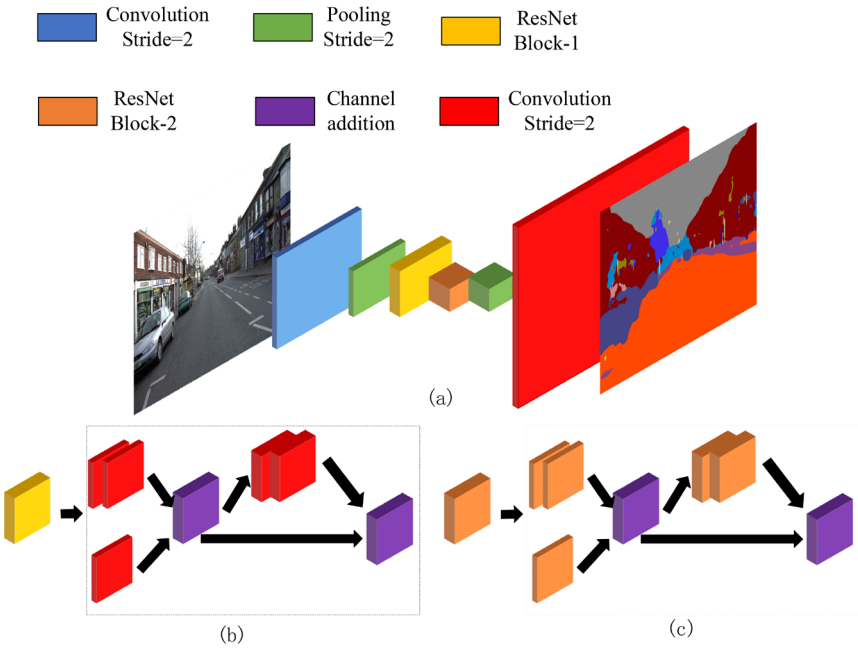


Figure 6. The DeepLab v2 structure with Resnet-18: (a) The overall network structure; (b) The structure of block 1 with Resnet; (c) The structure of block 2 with Resnet.

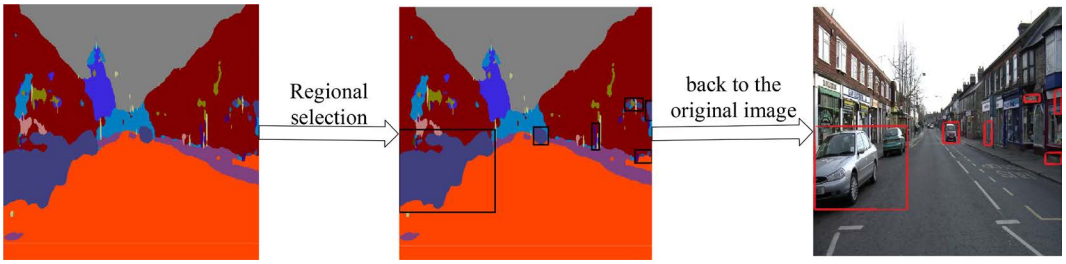


Figure 7. The area selection process.

where l_i is the i-th label category). These input and output areas appear in pairs and are called the Markov random field [19]. Thus, a conditional random field can be established for every input and output area. For the output Y, when the conditional probability $P(Y|X)$ takes the maximum value, the conditional random field can be described as follows:

$$P(Y|X)=\frac{1}{Z(X)}\exp\left(-E(Y|X)\right) \tag{1}$$

where $E(Y|X)$ is the variation trend of the random variable Y in the expression, and it is also called the energy function. Here, $Z(X)=\sum_{X,Y}\exp\left(-E(Y|X)\right)$ is a normalizing factor for the probability value of the potential function. From the above expression, it is obvious that our goal is to evaluate the Y output when the energy function $E(Y|X)$ is at the minimum. According to the definition of a conditional random field, the expression of the energy function can be described as follows:

$$E(Y|X)=\sum_i \varphi_u(y_i)+\sum_{i<j}\varphi_p(y_i,y_j) \tag{2}$$

where $\varphi_u(y_i)$ is a unary potential function describing the probability that the i-th pixel point is assigned to label y_i—that is to say, it describes the cost of assigning label y_i to the i-th pixel point, and $\varphi_p(y_i, y_j)$ is a binary potential function that describes the cost of assigning pixels i and j to the same label. In this model, the unary potential function is taken from a fully connected convolutional neural network; that is, the predictive value of every pixel label is obtained through a fully connected convolutional network. Because the unary potential function does not consider the smoothness property of the picture or the dependency relationship between pixels, the binary potential function is designed to compensate for this defect; it has a picture-smoothing process and encourages assigning positionally adjacent pixels with similar colours to the same label. According to [19] [20], the binary potential function is designed as a mixed Gaussian model as follows.

$$\varphi_p(x_i, x_j) = \mu(x_i, x_j)\sum_{m=1}^{M} w^{(m)} k^{(m)}(f_i, f_j) \tag{3}$$

where $w^{(m)}$ is the m-th Gaussian kernel weight value; $k^{(m)}$ is the number of Gaussian kernels ($m = 1, \cdots, M$), and the Gaussian kernel method for selecting M is the same as the binary potential function [19]. In this paper, the colour and spatial features of the picture are selected as the Gaussian kernel. The spatial feature Gaussian kernel is used to describe the relative positions of two pixel points in a conditional random field. The farther apart their relative positions are, the larger the value of the binary potential function is. However, the calculation method cannot be applied well in the model described in this paper. As shown in **Figure 7**, the conditional random field areas of the input image are different and independent. Some are small and contain few target objects. In those areas, the traditional spatial-feature Gaussian kernel may classify the pixels with relatively distant locations into different categories. However, we want most pixels to be segmented into one category in those small areas. To avoid this problem, we set a threshold area value to obtain the desired Gaussian kernel. Under the selected threshold value, in a small area (usually the threshold value is set to 1/25, 1/20, 1/15 of the original picture size), the Gaussian does not use an empty information feature; thus, the binary potential function can be expressed as follows:

$$\varphi_p(x_i, x_j) = \begin{cases} \mu(x_i, x_j) wk(f_i, f_j), S < \text{thre} \\ \mu(x_i, x_j)\sum_{m=1}^{M} w^{(m)} k^{(m)}(f_i, f_j), S \geq \text{thre} \end{cases} \tag{4}$$

where S is the size of the selected area, and the threshold value is pre-set. For specific spatial and colour features, the formula can be described as follows:

$$\varphi_p(x_i, x_j) = \begin{cases} \mu(x_i, x_j)\exp\left(-\dfrac{\|I_i - I_j\|}{2\sigma^2}\right), S < \text{thre} \\ \mu(x_i, x_j)\left(w^{(1)}\exp\left(-\dfrac{\|I_i - I_j\|}{2\sigma^2} - \dfrac{\|p_i - p_j\|}{2\theta^2}\right) + w^{(2)}\exp\left(-\dfrac{\|p_i - p_j\|}{2\theta^2}\right)\right), S \geq \text{thre} \end{cases} \tag{5}$$

where I_i, I_j represent the colour feature values of two pixels, p_i, p_j represent the spatial position feature values of the two pixels, and $w^{(2)}$, σ and θ represent the parameters of the mixed Gaussian kernel model obtained through learning.

By evaluating the unary and binary potential function, $E(Y|X)$ is obtained; then, $P(Y|X)$ and the segmentation result Y are obtained by the conditional random field. Finally, the network parameters are trained by evaluating the losses on the final revised result diagram. The process is shown in Figure 8.

2.3. The Local Conditional Random Field Model

By applying the locally connected conditional random field, it is possible to obtain more accurate segmentation results in the selected areas shown above (the pedestrian, bicycle and motor vehicle areas). To make full use of the continuity between frames, we record the areas where the segmentation results variation exceed a certain threshold value after applying the conditional random field (this part is recorded as a misclassified area, and the threshold values are usually set to 0.5, 0.7, and 0.8). Then, these areas are used as baselines for correction. If these areas are followed by misclassifications in the subsequent frame, the uncorrected result will be replaced with the corrected reference result.

The proposed SiameseFC model is used for tracking. The training sample used as input to tracking model is obtained by cutting around the area of the above misclassification [23]. In this paper, the improvised SiameseFC consists of two parts. 1) The training image used to input SiameseFC is a greyscale map of the segmentation results based on a locally conditional random field. In this way, the network can make full use of the target boundary information extracted by the locally conditional random field. 2) The SiameseFC convolutional structure contains only two convolutional layers and pooling layers. This simple structure greatly reduces the computational time required by the original SiameseFC network. Moreover, the performance of this simplified convolutional structure performance is no worse than the performance of more complex convolutional structures regarding the greyscale-map segmentation results of images with little information.

For the cropped greyscale sample, feature extraction and tracking training are

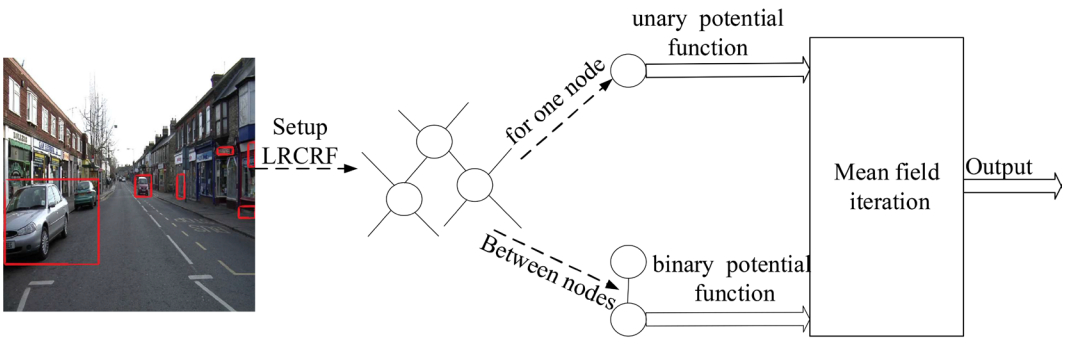


Figure 8. The process for training the network parameters of the proposed method.

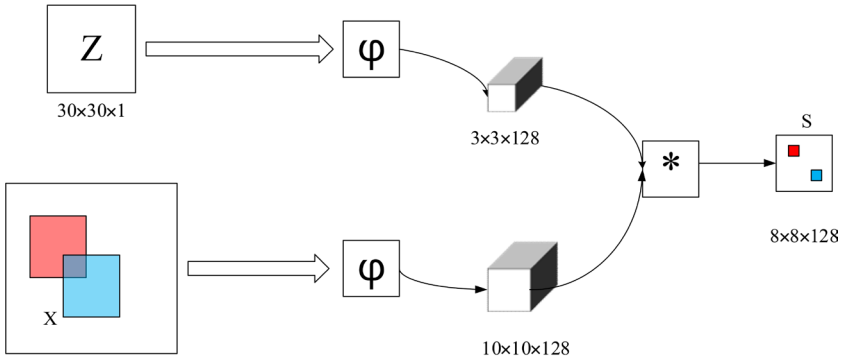


Figure 9. The structure for feature extraction and tracking training.

performed using the network structure shown in **Figure 9**.

In the tracking network, z is the training sample in the input tracking network. X denotes the pending search area, and the area determined by the above mentioned misclassification area is doubled in size in the segmentation result image, where φ represents two different convolutional structures each designed according to their own environment. These convolution structures are designed to extracting image features. The images tracked in this paper are greyscale images of the segmentation results, with well-described boundary information of the target objects through the conditional random field process. The architectures with two convolutional layers and pooling layers serve as the structure of φ. This approach substantially improves the tracking speed, and the simple convolution structure still performs well on the greyscale segmentation result map in an image where the information is obvious. When defining the loss of the tracking network, the point-by-point loss method in SiameseFC is used. That is, the loss is obtained for each pixel in the final 8 × 8 × 1 feature map. In the map, the misclassified areas correspond to the label 1, and the remaining areas are labelled as −1. For each point in this feature map, the loss is obtained as follows:

$$l(y,v) = \log\left(1+\exp(-yv)\right) \tag{6}$$

where $l(y,v)$ is the loss function, v is the output of each point on the 8 × 8 × 1 feature map, and y is the label value of the corresponding points. The final loss $L(y,v)$ is obtained by summing the loss from all points

$$L(y,v) = \frac{1}{|D|}\sum_{u\in D} l\left(y[u], v[u]\right) \tag{7}$$

where D is the set of all points in the 8 × 8 × 1 feature map. The tracking network is trained by stochastic gradient descent (SGD) [23]. This process can be described as finding the most suitable function f to train the network parameters θ by making any input z and x such that the output value $f(z,x;\theta)$ obtains the minimum loss value, which can be described as

$$\theta^* = \arg\min_\theta L\left(y, f\left(z,x;\theta\right)\right) \tag{8}$$

Finally, in consideration of the uncertain misclassifications and information singularity of the greyscale image of the segmentation result, the tracking net-

work is designed as a form of online learning that works only during the testing stage of the whole model, as shown in **Figure 5**. During the testing stage, a few grey-scale samples around the misclassification area are input to the tracking network for training. Usually, the tracking network converges in the training samples after iterations over 5 epochs.

3. Model Training and Testing

The model training process involves training only the LRCRF field model. The tracking network works only in the testing stage in an online fashion. This is because the prior information of misclassified areas required by the tracking process is missing, and there is not much similarity in far-apart images in different frames. Therefore, the relevant characteristics of the misclassified areas cannot be obtained via pre-training. For any input LRCRF training sample, the first step is to zoom to a fixed size and perform mean-value subtraction for the dataset image; then, the weight coefficient of the convolution kernel in the DeepLab-Resnet 18 network is initialized using the Xavier method [27]. The desired training parameters are initialized using constant values for CRF. Then, we use cross entropy as a model loss function and train the network using Adam until it converges. Finally, we save the training model parameters [28] [29].

In the test process, the input image is subtracted from the mean value and sent to the trained LRCRF model. Then, the misclassified area selected by the above-mentioned rules is obtained in the final result image. These areas are cropped from the segmentation result map to obtain a few grey-scale samples and sent to the improved SiameseFC [23]. As described in 2.3, the trained SiameseFC is used to track and correct the misclassified areas. Thus, the unnecessary cost introduced by applying a conditional random field to each image frame can be avoided for the similar areas in consecutive frames.

4. Experimental Results

4.1. Dataset and Evaluation Criteria

In this experiment, we used a computer equipped with an i5-7790k CPU and a GTX-1060 GPU. The fully connected conditional random field and the iteration of average field processes described in this paper are all iterated 10 times on the GPU. The dataset used in the experiment was acquired from three different source datasets. The first is a small sample dataset containing 106 paved-road scenes. The shooting location is urban roads, the weather conditions are good, and the road environment is relatively simple. This dataset contains 42 categories, including pedestrians, trucks, cars, buses, bicycles, roads, road signs, etc. The images are scaled to 720 × 1080 pixels. The second dataset includes all the images in the CAMVId dataset [29]. These images are taken from three different video sequences, including two acquired on sunny days and one acquired on a cloudy day. The cloudy dataset contains city scenes, and the sunny dataset contains suburban scenes. These three video sequences are separated by 30 frames,

and include a total of 853 pictures, including 11 label categories (pedestrian, motor vehicle, non-motorized vehicle, some simple road surfaces, etc.). All the images are scaled to 360 × 480 pixels. The last dataset was taken from the CVPR 2018 WAD Video Segmentation Challenge training set, which contains 2 video sequences. Most of the shooting locations are highway scenes with good weather conditions. The annotation interval is 10 frames. A total of 32,000 images were included, and there are 7 label categories (pedestrian, bicycle, motorcycle, automobile, truck, bus, tricycle, etc.). The images are scaled to 1600 × 1200 pixels. Due to their different annotation rules, the effects of the models proposed in this paper are verified using these differing data samples.

4.2. Evaluation Index

The segmentation performance of the proposed model is evaluated by commonly used industry standard metrics, including pixel accuracy, mean pixel accuracy and mean intersection over union [30]. Pixel accuracy is calculated as follows:

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=o}^{k}\sum_{j=0}^{k} p_{ij}} \tag{9}$$

where k is the total number of categories labelled in advance, p_{ij} is the total number of pixels in each category i, and the predicted category is j, which describes the percentage of correct classifications of all the pixels in the test data. Mean pixel accuracy is calculated by

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{10}$$

which describes the correct average value of the different classified pixels in the test data. Mean intersection over union is calculated as follows:

$$MIOU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k}\left(p_{ji} - p_{ij}\right)} \tag{11}$$

which describes the ratio of the true classifications of the different pixel categories in the test data to the misclassifications associated with that category.

We also adopt the region overlap index commonly used in the industry, which measures the intersection-over-union of the area to be tracked and the tracking network output areas and a pre-set threshold value. When the output of the tracking network is larger than the threshold value, the tracking is considered successful.

4.3. Comparison with a Method from the First Research Direction

Of the above datasets, the names dataset 1, dataset 2 and dataset 3 are used to denote the small dataset, the CAMVID dataset and the CVPR 2018 WAD Video

Segmentation Challenge dataset, respectively. To demonstrate the significant improvement in accuracy of model described in this paper, we compared it to the segmenting network SegNet, which is commonly used to segment road-traffic scenes in the industry. In this experiment, the basic SegNet network and the LRCRF model described in this paper are pre-trained on the ImageNet dataset. The experimental results are shown in Table 1. The time required to test a single frame image is shown in Table 2. The experimental results show that SegNet's scores on the three metrics on the different datasets are well below those of the model described in this paper (all the experimental results for the PA, MPA, and MIOU are shown as percentages). The single-frame image execution speed for the different datasets increases slightly in the LRCRF model, but for a general-purpose real-time system (where the frequency requirement is 10 - 30 fps) with a camera resolution of 720 p (which means 720 × 1080), the experimental results show that the model proposed in this paper is well-suited for real-time system applications under these conditions.

The model proposed in this paper is applied to the conditional random field for only some special areas. To compare the benefits of applying the conditional random field in those areas, this paper verified the segmentation indicators for certain special areas (e.g., pedestrian, bicycle and automobile) [31] [32]. As shown in Table 2, because the target marked in dataset 3 is the target selected in this paper, the experimental data used dataset 1 and dataset 2. The experimental results are shown in Table 3. In the table, F represents a special area, and N represents the remaining area. The experimental results show that the metrics for the model in this paper have been substantially optimized. This result occurs because the proposed model is able to obtain the pixel level dependency relationships in the images, which subtly help to partition the object boundaries.

Table 1. Comparisons of pixel accuracy, mean pixel accuracy and mean intersection-over-union between SegNet and LRCRF on dataset 1, dataset 2, and dataset 3.

Dataset	Dataset 1			Dataset 2			Dataset 3		
Evaluation index	PA	MPA	MIOU	PA	MPA	MIOU	PA	MPA	MIOU
SegNet	76.5	48.8	28.7	88.6	65.9	50.2	93.4	89.5	81.3
LRCRF	80.3	50.3	29.9	94.4	68.3	52.8	98.8	93.3	86.6

Table 2. Comparison of the time cost of SegNet and LRCRF on dataset 1, dataset 2, and dataset 3.

Dataset	Dataset 1		Dataset 1		Dataset 1	
Method	SegNet	LRCRF	SegNet	LRCRF	SegNet	LRCRF
Resolution	720 × 1080		360 × 480		1600 × 1200	
ms/frames	46.8	93.6	21.2	43.3	59.6	114.2

Table 3. Test set results on dataset 1 and dataset 2: our LRCRF compared with SegNet.

Dataset	Dataset 1						Dataset 2					
Evaluation index	PA		MPA		MIOU		PA		MPA		MIOU	
Areas	F	N	F	N	F	N	F	N	F	N	F	N
SegNet	66.2	77.8	60.2	47.4	44.8	28.0	89.2	88.1	68.4	65.1	60.3	44.4
LRCRF	84.5	79.7	67.7	48.6	51.2	27.6	97.2	91.8	75.3	64.3	66.4	45.0

4.4. Comparison with a Method from the Second Research Direction

In the second research direction, a fully connected conditional random field model is usually used to refine the segmentation results of the image. However, simply applying CRF to full scenes may cause inefficiency in the operation process in some areas. To verify that the model described in this paper is better than the non-special area, the effects of the fully connected conditional random field model and the proposed model are compared. The segmentation accuracy under different basic segmented network structures is also compared. We used the DeepLab-Resnet 18 structure for the basic segmented network of DeepLab-Resnet 18-CRF, and applied the fully connected conditional random field model. The experimental results are shown in Table 4. The experimental results show that the convolution structure of DeepLab v2 significantly improve the segmentation accuracy by replacing it with Resnet 18, and that the segmentation accuracy of the model described in this paper applied to the non-special regions does not lose much information. In addition, the segmentation accuracy does not lose much information in normal areas under our model.

To demonstrate that the model in this paper has a lower time complexity than the traditional fully connected conditional random field model, we report the execution time of the fully connected conditional random field model and the proposed model on different test datasets. The experimental results in Table 5, show that the running time is drastically faster due to the elimination in unnecessary computing operations.

4.5. Verification of Tracking Refinement Effects

In this paper, the tracking effects of the tracking network are verified using three different datasets. For the test samples in the different datasets, the segmentation results of the prior frame of the current frame are obtained by LRCRF in each test sample in the testing dataset. Then, the test data are corrected in the tracking reference map. The frame interval between the tracking reference map and the test chart is gradually increased, and the refinement effect of the tracking reference map to the test data is validated at the different frame intervals. The accuracy values of the experimental results are shown in Table 6, and the execution time is shown in Table 7. The results show that the speed and accuracy of segmentation for the model proposed in this paper achieve optimal effects at small

Table 4. Comparison of pixel accuracy, mean pixel accuracy and mean intersection-over union-between DeepLab v2 and DeepLab-Resnet 18-CRF on dataset 1, dataset 2, and dataset 3.

Dataset	Dataset 1						Dataset 2					
Evaluation index	PA		MPA		MIOU		PA		MPA		MIOU	
Areas	F	N	F	N	F	N	F	N	F	N	F	N
DeepLab v2	81.6	78.2	64.7	47.3	49.4	27.0	95.8	87.0	73.4	63.8	65.2	44.1
DeepLab-Resnet 18-CRF	85.2	81.5	68.2	50.0	51.7	28.6	97.5	92.3	75.1	65.8	67.0	46.1
LRCRF	84.5	79.7	67.7	48.6	51.2	27.6	97.2	91.8	75.3	64.7	66.4	45.0

Table 5. The time cost of DeepLab-Resnet 18-CRF and LRCRF for images at different resolutions from dataset 1, dataset 2 and dataset 3.

dataset	dataset 1		dataset 2		dataset 3	
method	DeepLab-Resnet 18-CRF	LRCRF	DeepLab-Resnet 18-CRF	LRCRF	DeepLab-Resnet 18-CRF	LRCRF
resolution	720 × 1080		360 × 480		1600 × 1200	
ms/frames	896.4	93.6	475.2	43.3	1433.6	114.2

Table 6. The accuracy of LRCRF on dataset 1, dataset 2 and dataset 3.

evaluation index	frames	dataset 1			dataset 2			dataset 3		
		PA	MPA	MIOU	PA	MPA	MIOU	PA	MPA	MIOU
LRCRF	0	80.3	50.7	31.2	94.4	69.1	54.8	97.8	93.3	87.6
	3	79.6	50.3	30.8	93.8	68.7	54.5	97.2	92.8	86.8
	5	78.8	49.8	30.2	92.8	68.1	53.7	96.1	92.2	85.4
	7	77.3	49.1	29.1	91.2	66.8	52.2	94.4	91.4	83.8
	9	76.1	48.2	27.8	89.7	65.7	51.3	93.0	89.2	81.6

Table 7. The execution time (ms/frame) of LRCRF on dataset 1, dataset 2 and dataset 3.

	0	3	5	7	9
Dateset1	93.6	65.5	60.17	57.2	53.1
Dataset 2	43.3	30.0	26.8	24.3	23.1
Dataset 3	114.2	81.36	74.7	70.8	69.0

frame intervals (e.g. 3 frames) between the tracking reference map and test chart.

Finally, the tracking effects are compared when the RGB original image or the segmentation result greyscale image are used as the tracking reference map. The results are shown in Figure 10. When the RGB original images are used as tracking reference maps in simple convolution, it is difficult to fully extract the features of the object to be tracked. Moreover, using the RGB original graph as a tracking reference map does not fully capitalize on the image features extracted

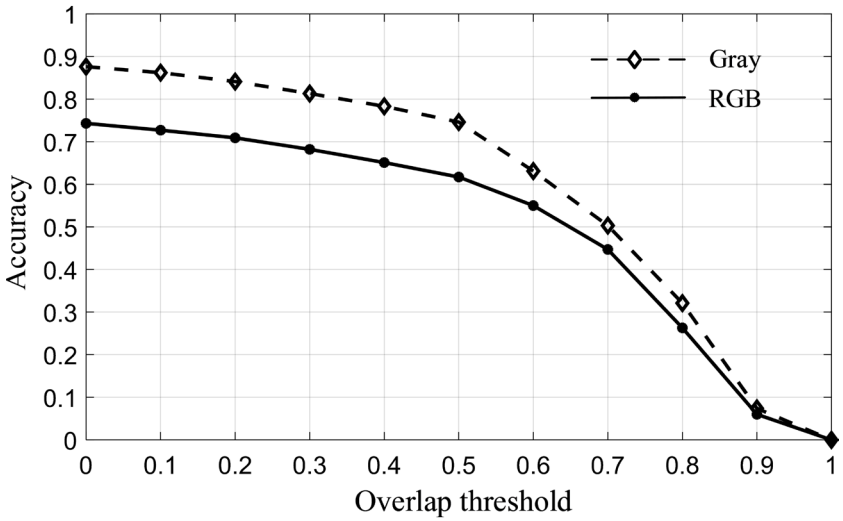


Figure 10. The accuracy as a function of the overlap threshold when employing RGB and greyscale images as the tracking reference maps.

from basic segmentation networks. Eventually, the tracking effect using the RGB original image becomes much lower than the tracking effect when using the greyscale map of the segmentation result.

5. Conclusion

An improved conditional random field model application and a new region tracking application are proposed in this paper. The proposed method applies the probabilistic graph model to systems with high real-time requirements. The unique features of traffic road scenes are analysed, the conditional random field model is selected to be applied to special areas, and different binary potential functions are selectively used. Using this approach, areas that have clear boundaries in traffic scenes are ignored, while the conditional random field model is applied to those areas with non-smooth or unclear boundary areas. This filtering process greatly reduces the time complexity of the system without losing too much precision, and further optimizes the operation time of the system. By considering the time interdependencies of sequential video images, the application tracking technology tracks and modifies the successive misclassified areas that appear in consecutive frames. The experimental results show that the proposed method achieves state-of-the-art performance.

Fund

This research was funded by the Key Research and Development Program of Zhejiang Province (Grants No. 2020C03098).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Gupta, S., Arbeláez, P., Girshick, R. and Malik, J. (2015) Indoor Scene Understanding with RGB-D Images: Bottom-Up Segmentation, Object Detection and Semantic Segmentation. International *Journal of Computer Vision*, **112**, 133-149. https://doi.org/10.1007/s11263-014-0777-6

[2] Wang, P., Shen, X.H., Lin, Z., Cohen, S. and Yuille, A. (2015) Towards Unified Depth and Semantic Prediction from a Single Image. 2015 *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 2800-2809. https://doi.org/10.1109/CVPR.2015.7298897

[3] Liu, Z., Li, X., Luo, P., Loy, C.C. and Tang, X. (2015) Semantic Image Segmentation via Deep Parsing Network. 2015 *IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1377-1385. https://doi.org/10.1109/ICCV.2015.162

[4] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. 2012 *International Conference on Neural Information Processing Systems*, Lake Tahoe, 3-6 December 2012, 1097-1105.

[5] Long, J., Shelhamer, E. and Darrell, T. (2014) Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **39**, 640-651. https://doi.org/10.1109/TPAMI.2016.2572683

[6] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, San Diego, 7-9 May 2015, 1-14.

[7] Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615

[8] Noh, H., Hong, S. and Han, B. (2015) Learning Deconvolution Network for Semantic Segmentation, 2015 *IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1520-1528. https://doi.org/10.1109/ICCV.2015.178

[9] Tu, Z.W. (2008) Auto-Context and Its Application to High-Level Visiontasks. 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 23-28 June 2008, 1-8. https://doi.org/10.1109/CVPR.2008.4587436

[10] Ladicky, L., Russell, C., Kohli, P. and Torr, P.H. (2009) Associative Hierarchical CRFs for Object Class Image Segmentation. 12th *IEEE International Conference on Computer Vision*, Kyoto, 29 September-2 October 2009, 739-746. https://doi.org/10.1109/ICCV.2009.5459248

[11] Mottaghi, R., Chen, X.J,, Liu, X.B., Cho, N.G. and Lee, S.W., *et al.* (2014) The Role of Context for Object Detection and Semantic Segmentation in the Wild. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 891-898. https://doi.org/10.1109/CVPR.2014.119

[12] Vemulapalli, R., Tuzel, O., Liu, M. and Chellappa, R. (2016) Gaussian Conditional Random Field Network for Semantic Segmentation. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 3224-3233. https://doi.org/10.1109/CVPR.2016.351

[13] Isobe, S. and Arai, S. (2017) Deep Convolutional Encoder-Decoder Network with Model Uncertainty for Semantic Segmentation. 2017 *IEEE International Conference on Innovations in Intelligent Systems and Applications*, Gdynia, 3-5 July 2017, 365-370. https://doi.org/10.1109/INISTA.2017.8001187

[14] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks

for Biomedical Image Segmentation. 2015 *Medical Image Computing and Computer-Assisted Intervention*, Munich, 5-9 October 2015, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[15] Zhu, S.P., Xia, X., Zhang, Q.R. and Belloulata, K. (2007) An Image Segmentation Algorithm in Image Processing Based on Threshold Segmentation. 3*rd International IEEE Conference on Signal-Image Technologies and Internet-Based System*, Shanghai, 16-18 December 2007, 673-678. https://doi.org/10.1109/SITIS.2007.116

[16] Brejl, M. and Sonka, M. (1998) Edge-Based Image Segmentation: Machine Learning from Examples. 1998 IEEE International Joint Conference on Neural Networks Proceedings. *IEEE World Congress on Computational Intelligence*, Anchorage, 4-9 May 1998, 814-819. https://doi.org/10.1109/IJCNN.1998.685872

[17] Karoui, I., Fablet, R., Boucher, J. M. and Augustin, J.M. (2006) Region-Based Image Segmentation Using Texture Statistics and Level-Set Methods. 2006 *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Toulouse, 14-19 May 2006, II. https://doi.org/10.1109/ICASSP.2006.1660437

[18] Zhang, Q. and Hu, Y.L. (2012) Image Segmentation Algorithm Based on Spectral Clustering Algorithm. *Journal of Shenyang Ligong University*, **33**, 6.

[19] Chen, L.C., Papandreou, G., Kokkinos, I., *et al.* (2018) DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **40**, 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[20] Zheng, S., Jayasumana, S., Romera-Paredes, B., *et al.* (2015) Conditional Random Fields as Recurrent Neural Networks. 2015 *IEEE International Conference on Computer Vision*, *Santiago*, 7-13 December 2015, 1529-1537. https://doi.org/10.1109/ICCV.2015.179

[21] He, K.M., Zhang, X.Y., Ren, S.Q. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[22] Yu, F. and Koltun, V. (2016) Multi-Scale Context Aggregation by Dilated Convolutions. *International Conference on Learning Representations*, Caribe Hilton, 2-4 May 2016, 1-4.

[23] Bertinetto, L., Valmadre, J., Henriques, J.F., *et al.* (2016) Fully-Convolutional Siamese Networks for Object Tracking. 2016 European Conference on Computer Vision, Amsterdam, 8-10, 15-16 October 2016, 850-865. https://doi.org/10.1007/978-3-319-48881-3_56

[24] Brostow, G.J., Julien, F. and Roberto, C. (2009) Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognition Letters*, **30**, 88-97. https://doi.org/10.1016/j.patrec.2008.04.005

[25] Manthira Moorthi, S., Gambhir, R.K., Misra, I. and Ramakrishnan, R. (2011) Adaptive Stochastic Gradient Descent Optimization in Multi Temporal Satellite Image Registration. 2011 *IEEE Recent Advances in Intelligent Computational Systems*, Trivandrum, 22-24 September 2011, 373-377. https://doi.org/10.1109/RAICS.2011.6069337

[26] Glorot, X. and Bengio, Y. (2010) Understanding the Difficulty of Training Deep Feedforward Neural Network. *Journal of Machine Learning Research*, **9**, 249-256.

[27] Pont-Tuset, J. and Marques, F. (2016) Supervised Evaluation of Image Segmentation and Object Proposal Techniques. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **38**, 1465-1478. https://doi.org/10.1109/TPAMI.2015.2481406

[28] Rubinstein, R.Y. and Kroese, D.P. (2004) The Cross-Entropy Method. Springer, New York, 92-92. https://doi.org/10.1007/978-1-4757-4321-0

[29] Liu, T.J., Liu, H.H., Pei, S.C., *et al.* (2018) A High-Definition Diversity-Scene Database for Image Quality Assessment. *IEEE Access*, **6**, 45427-45438. https://doi.org/10.1109/ACCESS.2018.2864514

[30] Pont-Tuset, J. and Marques, F. (2013) Measures and Meta-Measures for the Supervised Evaluation of Image Segmentation. 2013 *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, 23-28 June 2013, 2131-2138. https://doi.org/10.1109/CVPR.2013.277

[31] Wang, J.Y. and Yuille, A. (2015) Semantic Part Segmentation Using Compositional Model Combining Shape and Appearance. 2015 *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 1788-1797. https://doi.org/10.1109/CVPR.2015.7298788

[32] Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B. and Yuille, A. (2015) Joint Object and Part Segmentation Using Deep Learned Potentials. 2015 *IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1573-1581. https://doi.org/10.1109/ICCV.2015.184