Image Semantic Segmentation Based on FCN-CRF Model

Hao Zhou*, Jun Zhang, Jun Lei, Shuohao Li, Dan Tu
Collage of Information System and Management
National University of Defense Technology
Changsha, China
e-mail: zhouhao10@nudt.edu.cn

***Abstract*—Image segmentation is a key point for analyzing and understanding image, which occupies an important position in image processing. Recent studies have attempted to tackle pixel level labeling tasks using deep learning. In our paper, we propose an approach of combining fully convolutional network and conditional random field for image semantic segmentation. We utilize FCN model to automatically learn features directly from original image data, and create local predictions and global structure consistency by combining fine layers and coarse layers. CRF is a probabilistic graph and used to fully exploit the context information. Our model train the whole deep network end-to-end with the back-propagation algorithm and maximum likelihood estimation. The key of jointing FCN and CRF is sensitivity of neurons, calculating the sensitivity by CRF and then transferring it to FCN. Experiments show our method achieves improved accuracy compared with several other methods on Pascal VOC 2012 dataset.**

Keywords-image semantic segmentation; FCN; CRF; end-to-end

I. INTRODUCTION

The research contents of computer vision mainly include five aspects: image input device, low level vision, middle level vision, high level vision, and system architecture. And low level image processing has been the bottleneck of the development of computer vision. Image semantic segmentation [1]-[5] is a typical low level computer vision problem, which involves assigning a label to each pixel in an image. It also plays an important role in the field of image retrieval [6], object recognition [7], [8] and face recognition [9]. The purpose of segmentation is to simplify or change the representation of the image, which makes the image more meaningful and easier to analyze. There are two challenges in image segmentation: uncertainty and fuzziness, and so it is important to get the better feature representation for accurately labeling each pixel.

In order to reduce the effects of these uncertainties and fuzziness on image labeling, one of the main methods is to make use of the information contained in the image as much as possible. Generally speaking, there are two important forms of information in the image: (1) Pixel value for each pixel, from which we can extract kinds of image features, such as the basic color features (RGB). (2) The correlation between image elements (such as pixels, plots, and objects), which is the so-called context information.

The feature extraction method in previous studies usually relies on hand-crafted features, such as HOG [10] and SIFT [11]. In recent years, with the emergence of deep learning, to a certain extent, the problem has been effectively solved. At present, Convolutional Neural Network (CNN) is the most effective method for feature extraction. In high level computer vision, CNNs have been successful in object detection, image to text and so on. But there are some shortcomings for CNNs in lower level computer vision. General CNNs cannot be able to achieve high accuracy in image segmentation for they filter out the low level information. And the results of image segmentation usually are non-sharp boundaries and blob-like shapes for convolutional filters with large receptive fields and max-pooling layers. So this promotes the development of CNNs to the low level computer vision. And how to combine the features of the upper and lower layers is the key to solve the problem.

Another problem is to combine context information. Significant progresses have been made on this problem in recent years, but the main features of these methods are mostly designed for specific tasks and not unified theoretical framework. The probabilistic graph model capable of fusing context information can solve these problems well. Conditional Random Fields (CRF) [12] is one of discriminative probability graph model, which combines with the characteristics of the maximum entropy model and hidden Markov model. For some common classifiers, they rarely use adjacent samples while CRFs can be tagged and predicted with context relations. In computer vision, CRFs are often used for object recognition and image segmentation.

In this paper, we mainly research the image semantic segmentation, using the FCN and CRF to make neural network end-to-end. Through FCN, we can extract the high level features and the low level features of the image. And then according to these features, CRF will classify each pixel. The end-to-end train makes the training process do not require manual interventions. In the next section, we review related works on FCN and some approaches for image semantic segmentation using CNN. The following sections explain the formulation of our model, including the way of CRF modeling context information and FCN-CRF training end-to-end. Finally, we demonstrate state-of-the-art results on PASCAL VOC 2012.

978-1-5090-3755-1/16/$31.00 ©2016 IEEE

II. RELATED WORKS

In this section, we review some works relevant to ours. The main relevant work is on using deep learning and CNN for image segmentation. Recently CNNs have been a great success in high level computer vision, such as image classification [13], [14], object detection, then some researchers promote CNNs to lower level computer vision, like Image segmentation.

There are varieties of approaches that tackle the image segmentation using deep learning. Grangier D et al. [15] use supervised greedy learning training deep CNN, and take pixels as the input. And this is the preliminary way to utilize deep CNN, but they also get some improvements over CRF learning. Farabet C et al. [16] use a multi-scale CNN and extract dense feature vectors to encode regions of multiple sizes. The method produces a powerful representation for capturing texture, shape and contextual information. Pinheiro PHO et al. [17] propose a Recurrent Convolutional Neural Network (RCNN) which allow them to consider a large input context and limit the capacity of the model. Schulz H et al. [18] propose a convolutional network architecture with multiple output maps, suitable loss functions, and pairwise class location filters. Long J et al. [19] propose a Fully Convolutional Networks (FCN) which can take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning, which just replace the fully connection layers with the convolutional layers. And they combine a deep, coarse layer with a shallow, fine layer to produce accurate and detailed segmentations. In our model, we also use the FCN for feature extraction, and combine a high layer with a lower layer which can get more accurate information. The deep, coarse features can know what the object is and the shallow, fine features can know where the object is. And different from [19], our FCN is just used for features extraction which are the inputs of CRF. Then CRF rather than FCN can label each pixel.

Liu F et al. [20] propose to combine a pre-trained large CNN to generate deep features for CRF learning. The deep CNN is trained on the ImageNet dataset and the CRF is trained by SSVM. To exploit context information, they construct spatially related co-occurrence pairwise potentials and incorporate them into the energy function. In their model, CRF is post-processed which is different from our model. In our model, the FCN and CRF are end-to-end trainable which makes the errors transfer from top to bottom and makes the model data consistency. Shuai Zheng et al. [21] introduce a feed-forward architecture where map superpixels to rich feature representations, which exploit statistical structure in the image and no explicit structured prediction in the label space mechanisms to avoid complex and expensive inference. They propose a new CNN model that combines the strengths of CNN and CRF, and exploits mean-field approximate inference for CRF with Gaussian pairwise potentials as Recurrent Neural Network (RNN), and this method make it possible to train the whole deep network end-to-end. But we don't use the CRFasRNN for end-to-end training. In our model, FCN and CRF joint end-to-end training is based on maximum likelihood estimation method and gradient descent method so that can get rid of the limitations of using mean-field method to find the optimal solution.

III. METHOD

A. CRFs and Context Information Modeling

In this section, we briefly introduce the CRFs for image segmentation and context information modeling in our paper. CRF is a discriminative undirected probabilistic graph model, which, in essence, is a Markov random field with a given observations. When given the observation sequence, CRF has a unified exponential model for the joint probability of the whole sequence. CRF models have strong reasoning ability, and can be able to train and inference with complex, overlapping and non-independent features, which make full use of the context information in the process and arbitrarily add other external features for model to get abundant information.

Definition 2.1. Let G be a factor graph over X and Y. Then (X,Y) is a CRF if for any value x of X, the distribution $p(y \mid x)$ factorizes according to G.

To be more specific, let Y be the label random variable, and can take any value from $\{y_1, y_2, ..., y_n\}$; Let X be the vector formed by the random variables $X_1, X_2, ..., X_N$, and the number of pixels in the image is N. Given a graph $G = (V, E)$, where V are the vertices of the graph and E are the edges of the graph, then CRF model can be expressed as:

$$P(Y \mid X;\Theta) = \frac{1}{Z(X)} \exp(-E(Y, X;\Theta)) \tag{1}$$

here $E(Y, X;\Theta)$ is called the energy function, and $Z(X)$ is the partition function where $Z(X) = \sum \exp(-E(Y, X;\Theta))$. In our method, to combine the context information, we should model the relationship between pixels, then the energy function $E(Y, X;\theta)$ is:

$$E(Y, X;\Theta) = \sum_{p \in N} \Phi^{(1)}(y^p, x;\theta) + \sum_{(p,q) \in S} \Phi^{(2)}(y^p, y^q, x;\theta) \tag{2}$$

where $\Phi^{(1)}$ is the unary energy component, and $\Phi^{(2)}$ is the pairwise energy component and the energy function is briefly written as $E(X)$. In our method, unary energy components are based on the features of a single pixel from FCN, which are one-to-one relationships between pixel and unary energy components, while the pairwise energy components are based on not only themself, but the adjacent pixels including eight pixels, four adjacent edges and four adjacent vertices. However, not all eight adjacent pixels will be used for modeling the pairwise energy components due to the complexity and computational feasibility. So in this paper, we make joint modeling of four adjacent edges pixels to reduce the computational complexity and as little as possible to reduce the effect. And at last, we should minimize the energy function $E(X)$ to yield the most probable label assignment X for the given image by Maximum Likelihood

Estimation (MLE). As the same time, the unary energy components are called state function for merely depending on the current location, and the pairwise energy components are called transition functions for depending on the current location and the adjacent location in graph.

In our paper, we use the simple state and transition function for CRF modelling. The unary energy function is defined as:

$$\Phi^{(1)}(y^p, x) = \lambda_m I(y, y^p) x(l) \tag{3}$$

where $I(,)$ is the indicator function which equals 0 or 1, $x(l)$ is the $l-$ th entry of feature vector x, λ_k is the parameter and state function $s_m = I(y, y^p)x(l)$. And the pairwise energy function is defines as:

$$\Phi^{(2)}(y^p, y^{\P}, x; W) = \mu_m I(y, y^p) I(y', y^{\P}) \tag{4}$$

where μ_k is the parameter and transition function $t_m = I(y, y^p)I(y', y^{\P})$. The pairwise energy function is just the relation between two pixels. In order to obtain more contextual information, we consider the superpixel pairs of four adjacent spatial relations: above, below, left and right. Then the feature mapping for the pairwise potential can be written as:

$$\sum_{(p,\P)\in S} \Phi^{(2)}(y^p, y^{\P}, x; \theta) = \sum_{(p,\P)\in S_1} \Phi_1^{(2)}(y^p, y^{\P}, x; \theta) + \sum_{(p,\P)\in S_2} \Phi_2^{(2)}(y^p, y^{\P}, x; \theta) + \sum_{(p,\P)\in S_3} \Phi_3^{(2)}(y^p, y^{\P}, x; \theta) + \sum_{(p,\P)\in S_4} \Phi_4^{(2)}(y^p, y^{\P}, x; \theta) \tag{5}$$

where S_1, S_2, S_3, S_4 are the adjacent spatial relations, and respectively represent 'above', 'below', 'left' and 'right'.

B. FCN with High and Lower Level Combination

In this section, we mainly introduce the FCN for pixelwise prediction and the way of high and low level combination. Matan et al. [22] first propose a Convnet with arbitrary-sized inputs to recognize strings of digits extending the classic LeNet [21]. In general, FCN is a net with only layers computing a nonlinear filter, and the main feature of FCN takes arbitrary size inputs and produces correspondingly-sized outputs, because there are no fully connection layers in FCN which can fix dimensions and throw away spatial coordinates. When transforming fully connected layers into convolution layers, a classification net enables to output a heatmap, and that is why FCN is more suitable for dense prediction compared to CNN with fully connected layers.

FCN for image segmentation cannot achieve the highest accuracy for the features of the FCN outputs are coarse features. After max-pooling layers, the main features can be retained while others discarded, so most of features cannot be used for image segmentation, then the edges of the object become blurred and some pixels are assimilated. Due to less max-pooling, the lower layers have finer features, then we should combine the high, coarse layers with law, fine layers to get the features of different levels and let the model make local predictions with respecting global structure that high, coarse features are mainly used for what is the object which law, fine features are used for where is the object.

To summarize, in our paper, we use the FCN-8s architecture of [19] for the extracting features of the network, which provides the input data for CRF, and Fig. 1 shows the process of our FCN-CRF model and main points. The way of FCN-8s combining high with lower level is that the pool4 prediction layers combining with the final layers FCN-16S, then additional predictions from pool3. FCN-8s is based on the VGG-16 network [20] which is used for image classification and restructured to perform pixel-wise prediction. FCN-8s fine-tune all layers by backpropagation through the whole net. It is unnecessary for image to class balancing while fully convolutional training can balance classes by weighting or sampling the loss, so image labels and background can be mildly unbalanced.

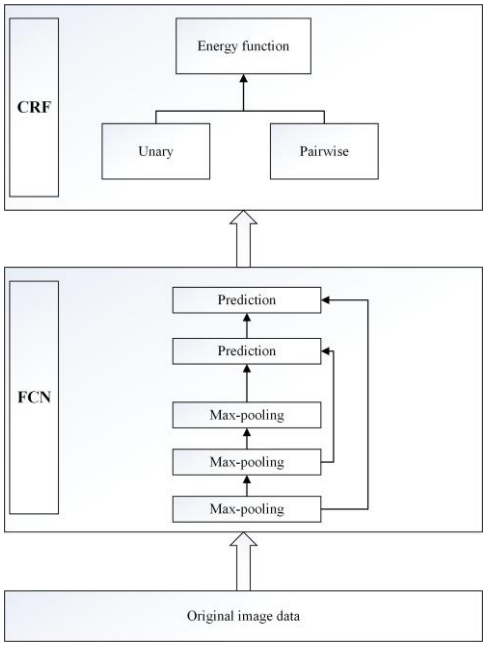

Figure 1. The process of FCN-CRF model and main points.

To unify the strengths of both FCN and CRFs, we train an end-to-end network using the back-propagation algorithm [23] and the Stochastic Gradient Descent (SGD) procedure. Different from [21], to combine the ability of deep learning and graphical modelling, they propose that a single iteration of the mean-field algorithm for CRF can be modelled as a stack of common CNN layers, then formulate dense CRF as an RNN for forming an end-to-end trainable system for image semantic segmentation, and FCN-8s only provides unary potentials to the CRF without pairwise potentials. Our model combine FCN with CRF for end-to-end trainable by transferring CRF sensitivity to FCN based on Stochastic Gradient Descent.

To sum up, the FCN-CRF model combines the FCN and the CRF, and CRF can be converted a shallow structure to a deep structure. The FCN-CRF model can directly use the image data, and automatically learn and extract features through multiple layers of abstractions. In our model, we

denote function of FCN as $f(X,\varpi)$, where ϖ are the parameters of FCN and X are the image pixels inputs. After passing through FCN, we obtain the image features denoted by $f(X,\varpi)=\{f(x_1,\varpi),f(x_2,\varpi)...f(x_N,\varpi)\}$. Then these features are inputted to CRF model. (1) is converted to:

$$P(Y\mid f(X,\varpi);\Theta)=\frac{1}{Z(f(X,\varpi))}\exp(\sum_{p\in N}\Phi^{(1)}(y^p,f(X,\varpi);\Theta)+\sum_{(p,q)\in S}\Phi^{(2)}(y^p,y^q,f(X,\varpi);\Theta)) \tag{6}$$

and we donate $\Omega=\{\varpi,\Theta\}$ are parameters of the model.

C. Model Learning and Inference with End-to-End

In this section, we introduce the way of CRF training and FCN-CRF end-to-end training and inference. Given the training image data $(x_{i,j},y_{i,j})$, $i,j=1,...n$ for graph. Parameter estimation is typically performed by penalized maximum likelihood, then the objection function can be denoted as:

$$l(\Omega)=\sum_{i=1}^{N}\log p(y^{(i)}\mid x^{(i)}) \tag{7}$$

Our model has a large number of parameters due to combining FCN and CRF, and to avoid measure over-fitting, we should use regularization. A common choice is based on the Euclidean norm of θ and a regularization parameter $1/2\sigma^2$. Then the objective function is:

$$l(\Omega)=\sum_{i=1}^{N}\log p(y^{(i)}\mid x^{(i)})-\frac{1}{2\sigma^2}\|\Omega\|_2 \tag{8}$$

After substituting in the CRF model (6) into the likelihood (8), we get the following expression:

$$l(\Omega)=\sum_{i=1}^{N}(\sum_{p\in N}\Phi^{(1)}(y^p,f(X,\varpi);\Theta)+\sum_{(p,q)\in S}\Phi^{(2)}(y^p,y^q,f(X,\varpi);\Theta))-\frac{1}{2\sigma^2}\|\Omega\|_2-\sum_{i=1}^{N}\log(Z(f(X,\varpi))) \tag{9}$$

To minimize the objective function, we use gradient descent method to optimal parameters. First, we introduce the way of CRF training, which is computing the gradient respect to parameter θ. Parameter λ_k are associated with unary energy function $\Phi^{(1)}$, and the gradient is:

$$\begin{aligned}\frac{\partial l(\Omega)}{\partial\lambda_k}&=-\sum_{i,j,k}P(x=a\mid Y_i,f(X,\varpi),\Theta)s_k\\&+\sum_{i,Y_i',j,k}P(x=a\mid Y_i,f(X,\varpi),\Theta)s_k+\frac{\lambda_k}{\sigma^2}\\&=-\sum_i\sum_j\sum_k s_k+\sum_i\sum_{Y_i'}\sum_j\sum_k s_k+\frac{\lambda_k}{\sigma^2}\end{aligned} \tag{10}$$

Parameter μ_k are associated with pairwise energy function $\Phi^{(2)}$, and the gradient is:

$$\begin{aligned}\frac{\partial l(\Omega)}{\partial\mu_k}&=-(\sum_{i,j,a_l,b}P(x_{m-1,n}=a_l,x_{m,n}=b\mid Y_i,f(X,\varpi),\Theta)t_k\\&+\sum_{i,j,a_r,b}P(x_{m+1,n}=a_r,x_{m,n}=b\mid Y_i,f(X,\varpi),\Theta)t_k\\&+\sum_{i,j,a_b,b}P(x_{m,n-1}=a_b,x_{m,n}=b\mid Y_i,f(X,\varpi),\Theta)t_k\\&+\sum_{i,j,a_a,b}P(x_{m,n+1}=a_a,x_{m,n}=b\mid Y_i,f(X,\varpi),\Theta)t_k)\\&+(\sum_{i,Y_i',j,a_l,b}P(x_{m-1,n}=a_l,x_{m,n}=b,Y'\mid Y_i,f(X,\varpi),\Theta)t_k\\&+\sum_{i,Y_i',j,a_r,b}P(x_{m+1,n}=a_r,x_{m,n}=b,Y'\mid Y_i,f(X,\varpi),\Theta)t_k\\&+\sum_{i,Y_i',j,a_b,b}P(x_{m,n-1}=a_b,x_{m,n}=b,Y'\mid Y_i,f(X,\varpi),\Theta)t_k\\&+\sum_{i,Y_i',j,a_a,b}(\quad,\quad x_{m,}\quad\mid Y_i,f(\quad),\Theta)t_k)+\frac{\mu_k}{\sigma^2}\\&=(\sum_{Y_i'}-1)\sum_i\sum_j(\sum_{k_l}t_k+\sum_{k_r}t_k+\sum_{k_b}t_k+\sum_{k_a}t_k)+\frac{\mu_k}{\sigma^2}\end{aligned} \tag{11}$$

We use the belief propagation algorithm [24] for computing the marginal probabilities and can train CRF model parameters well based on it. The FCN parameters can be trained by back propagation algorithm. The difficulty for combining FCN with CRF is how to transfer the error from CRF to FCN. In our model, the sensitivity of neurons are used to build the bridge between CRF and FCN for transferring the error.

Given the node of FCN's output is O, $O=\{o_1,o_2,...,o_n\}$, n is the number of FCN's output, then $o_i=f(x_i,\varpi)$. The sensitivity of node can be written as:

$$\begin{aligned}\delta_i=\frac{\partial l(\Omega)}{\partial o_i}=\frac{\partial l(\Omega)}{\partial f(x_i,\varpi)}&=\sum_{i,j,a}P(x_j=a\mid Y_i,f(x_i,\varpi),\Theta)\lambda_{ai}\\&-\sum_{i,Y_i',j,a}P(x_j=a,Y'\mid f(x_i,\varpi),\Theta)\lambda_{ai}\end{aligned} \tag{12}$$

And then the FCN's parameters ϖ can be iterative calculation by:

$$\omega_{i,j}(s+1)=\omega_{i,j}(s)+\eta\delta_j x_i \tag{13}$$

In summary, FCN-CRF model can be trained using end-to-end in which the error can be transferred from the outputs to inputs. In this way, the training speed is accelerated and the accuracy is also improved because each input corresponds to once CRF training rather than a complete CRF training. And the process of training and inference are shown in Fig. 2.

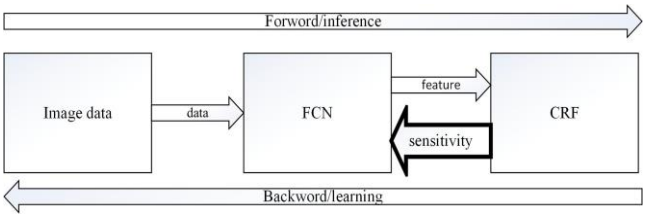

Figure 2. The process of training and inference.

After training, the parameters Ω^* are learnt, and can be used for inference. Given a new image graph X, the most probable label for pixel i is estimated by:

$$\hat{y}_i = \arg\max_{y_i} P(y_i \mid X, \Omega^*) \tag{14}$$

IV. EXPERIMENTS

In this section, we present the experimental results with our FCN-CRF model. We first introduce our experimental setup and then compare with state-of-the-art methods on Pascal VOC 2012 image segmentation dataset.

A. Experimental Setup

We test our model on Pascal VOC 2012 dataset for it is standard to evaluate image semantic segmentation approach and can compare with existing methods where have 2913 images containing objects from 21 categories. Then we choose 2000 images as our training and validation set and we divide the dataset into two parts: training set and test set that seventy-five percent of the images are used for training, and twenty-five percent are used for validation. We first train the model based on the training set and then validate our model based on the validation set, and last we evaluate the performance of our model with the standard VOC measure with test sets which consist of VOC 2012. At the preprocessing stage, we should resize the image into 500*500 for the size of the original image is not the same and the biggest width and height of images are 500 pixels.

We use the FCN-8s as the feature extraction, in which is composed of 13 convolutional layers and 5 max-pooling layers where pool1 and pool2 have two convolutional layers while pool3, pool4 and pool5 have three convolutional layers, then high layers combine with lower layers combines using 5 convolutional layers and some other layers such as drop layers and crop layers. Especially, in the training process, the last layer is euclidean loss layer rather than softmax loss layer for calculation error. The FCN-CRF model is training based on Caffe.

B. Evaluation Metrics and Results

There are four metrics for evaluating the performance of common image semantic segmentation which are pixel accuracy, mean accuracy, mean IU and frequency weighted IU. Among of them, mean IU is the most commonly used as the evaluation standard and its calculation method is: let n_{ij} be the number of pixels of class i predicted to belong to class j, where there are n_{c1} different classes, and let $t_i = \sum_j n_{ij}$ be the total number of pixels of class i, then mean IU can be written as:

$$meanIU = \frac{1}{n_{c1}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{15}$$

And like [21], we compare with other model from two aspects to prove our model is effective. We first compare the results of our end-to-end model with other model no end-to-end on our validation sets. There are two methods for image segmentation without end-to-end which are FCN-8s without applying CRF, and FCN-8s with applying CRF for postprocessing but disconnecting from the training of FCN. The mean IU of three models are computed, just see Table I. It is obvious that our end-to-end model is better than two other models. The FCN-8s without applying CRF gets a low recognition accuracy of 62.7% in three models. FCN-8s with applying CRF for postprocessing method is 65.5%, high than that of FCN-8s without applying CRF method, demonstrating that CRF is better for extracting context information and get higher accuracy in image segmentation. Our FCN-CRF method achieves the highest recognition accuracy of 71.8%, and improves the performance of FCN-8s with applying CRF for postprocessing in some degree. This fully describes that the CNN component and the CRF component can be learned to co-operate with each other by end-to-end training.

TABLE I. MEAN IU ACCURACY OF OUR APPROACH EVALUATED ON VALIDATION SET

Method	***Mean UI***
FCN-8s[19]	62.7
FCN-8s with CRF for postprocessing	64.5
Our model	71.8

TABLE II. MEAN IU ACCURACY OF OUR APPROACH EVALUATED ON TEST SET

Method	***Mean UI***
SDS[25]	51.6
FCN-8s[19]	62.2
Zoomout[26]	64.4
Context-Deep-CNN-CRF[27]	70.7
CRFasRNN[21]	72.0
Our model	72.8

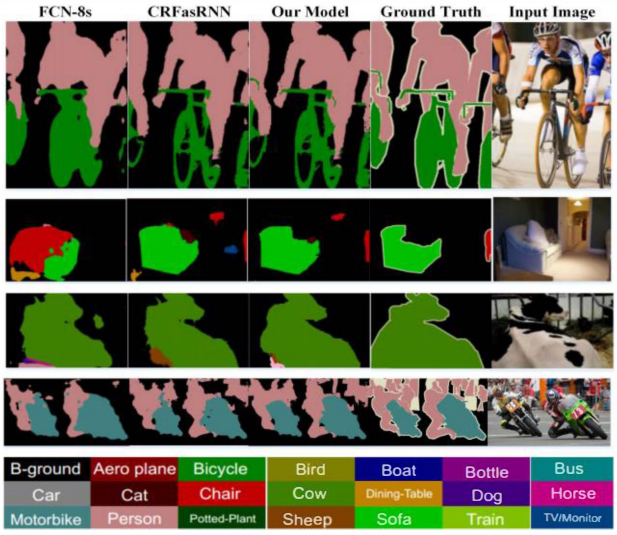

Figure 3. Our results on the validation set and compared with FCN-8s and CRFasRNN

Then we compare the results of our model with other state-of-the-art methods who use the standard Pascal VOC 2012 datasets on our test sets. The mean IU of these models are computed, just see Table II. And we can see that our approach results outperform other approaches in these datasets. In our test datasets, our results are slightly better than the state-of-the-art model. The FCN-8s model gets an accuracy of 62.2% and the FCN-8s with applying CRF for postprocessing method is 70.7%, the state-of-the-art CRFasRNN model get a accuracy of 72.0%, while our results are more accurate than their models, and the accuracy is 72.8%. Some examples of qualitative evaluation are showed in Fig. 3.

V. CONCLUSION

In this paper, we introduce an FCN-CRF model for image semantic segmentation to give each pixel a label. The FCN is adopted to automatically learn effective features directly from original image data, and combine a deep, coarse layer with shallow, fine layer to get more accurate and detailed information, thus avoid the need of constructing handcrafted feature. Then we combine FCN with CRF to incorporate image feature learning and dense predictions for per-pixel in a unified framework. As the same time, the training of FCN-CRF is in end-to-end fashion by computing transferring the sensitivity of neurons, which makes the errors transfer from top to bottom and makes the model data consistency. Last, experimental results demonstrate the accuracy of our method for image semantic segmentation.

ACKNOWLEDGMENT

This work was partially supported by the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing Grant 2015A04. We thank the Caffe team for their support.

REFERENCES

[1] Szummer M, Kohli P, Hoiem D. Learning CRFs using graph cuts. Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008: 582-595.

[2] Shotton J, Johnson M, Cipolla R. Semantic texton forests for image categorization and segmentation. Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008: 1-8.

[3] Lucchi A, Li Y, Smith K, et al. Structured image segmentation using kernelized features. Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012: 400-413.

[4] Ning F, Delhomme D, LeCun Y, et al. Toward automatic phenotyping of developing embryos from videos. Image Processing, IEEE Transactions on, 2005, 14(9): 1360-1371.

[5] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(8): 1915-1929.

[6] Paulin M, Mairal J, Douze M, et al. Convolutional Patch Representations for Image Retrieval: an Unsupervised Approach. arXiv preprint arXiv:1603.00438, 2016.

[7] Gidaris S, Komodakis N. Object Detection via a Multi-Region and Semantic Segmentation-Aware CNN Model. Proceedings of the IEEE International Conference on Computer Vision. 2015: 1134-1142.

[8] Ishii T, Nakamura R, Nakada H, et al. Surface object recognition with CNN and SVM in Landsat 8 images. Machine Vision Applications (MVA), 2015 14th IAPR International Conference on. IEEE, 2015: 341-344.

[9] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 815-823.

[10] Kadota R, Sugano H, Hiromoto M, et al. Hardware architecture for HOG feature extraction. Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on. IEEE, 2009: 1330-1333.

[11] Zhou H, Yuan Y, Shi C. Object tracking using SIFT features and mean shift. Computer vision and image understanding, 2009, 113(3): 345-352.

[12] Lafferty J, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[13] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012: 1097-1105.

[14] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[15] Grangier D, Bottou L, Collobert R. Deep convolutional networks for scene parsing. ICML 2009 Deep Learning Workshop. 2009, 3.

[16] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(8): 1915-1929.

[17] Pinheiro P H O, Collobert R. Recurrent convolutional neural networks for scene parsing. arXiv preprint arXiv:1306.2795, 2013.

[18] Schulz H, Behnke S. Learning Object-Class Segmentation with Convolutional Neural Networks. ESANN. 2012.

[19] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.

[20] Liu F, Lin G, Shen C. CRF learning with CNN features for image segmentation. Pattern Recognition, 2015, 48(10): 2983-2992.

[21] Zheng S, Jayasumana S, Romera-Paredes B, et al. Conditional random fields as recurrent neural networks. Proceedings of the IEEE International Conference on Computer Vision. 2015: 1529-1537.

[22] Matan O, Burges C J C, LeCun Y, et al. Multi-digit recognition using a space displacement neural network. NIPS. 1991: 488-495.

[23] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[24] Lafferty J, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.interface," ASME Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[25] Hariharan B, Arbeláez P, Girshick R, et al. Simultaneous detection and segmentation. Computer vision–ECCV 2014. Springer International Publishing, 2014: 297-312.

[26] Mostajabi M, Yadollahpour P, Shakhnarovich G. Feedforward semantic segmentation with zoom-out features. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3376-3385.

[27] Lin G, Shen C, Reid I. Efficient piecewise training of deep structured models for semantic segmentation. arXiv preprint arXiv:1504.01013, 2015.