Multiple Class Segmentation Using a Scene-based Framework

Shilin Wu*†‡§, Feng Zhu*‡§ and Xikui Miao*†‡§
*Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, PR China
Email:wushilin, fzhu, miaoxikui@sia.cn
†Graduate School of Chinese Academy of Sciences, Beijing 100049, PR China
‡Key Laboratory of Optical-Electronics Information Processing, Chinese Academy of Sciences, Shenyang 110016, PR China
§Key Laboratory of Image Understanding and Computer Vision, Liaoning Province 110016, PR China

***Abstract*—In this paper, we propose a new scene-based conditional model and investigate its performance on multiple class segmentation of images. By including the scene of an image, we also propose a new texture-environment potential to represent texture environment of a pixel. We compare our results to related work on the Olive & Torralba database and show that our model obtains greatly improved accuracy of the whole database. More significantly, a large perceptual improvement is gained, i.e. details of different objects are correctly labeled.**

I. Introduction

Different from object recognition methods, which aim to recognize a single object [1], [2], [3], [4], [5], [6], [7], multiple class segmentation methods [8], [9], [10], [11] attempt to perform concurrent multi-class object recognition and to classify all pixels in an image. Joint detection and segmentation of a single object class has been achieved by several authors [6], [7], whose works however cannot cope with arbitrary viewpoints or severe occlusion. In our work, by introducing scene of images into a conditional model (CM), we present a scene-based conditional model and show its effectiveness on multi-class images with arbitrary viewpoints and occlusion. The scene of an image describes the possible co-occurrence relationship of different objects.

We investigate our model on the Olive & Torralba database, which consists of 2688 images of 8 scenes: coast, forest, highway, inside city, mountain, open country, street and tall building. Images in the database are labeled with 15 classes: water, sky, mountain, tree, grass, road, sidewalk, building, rock, snow, sand, plant, car, sign and person. We split the Olive & Torralba database randomly into 15% training, 15% validation and 70% test sets. During training, validation and test on this database, accuracy values are all computed as the percentage of pixels assigned to correct class label, ignoring pixels labeled as void (black) in the groundtruth. The label 'void' is used to cancel pixels that do not belong to any of the 15 classes and to help quick hand-segmentation along the object boundaries. Some example images of several scenes and their corresponding groundtruth annotations from the Olive & Torralba database are shown in Figure 1. Colors show categories in the groundtruth of the images.

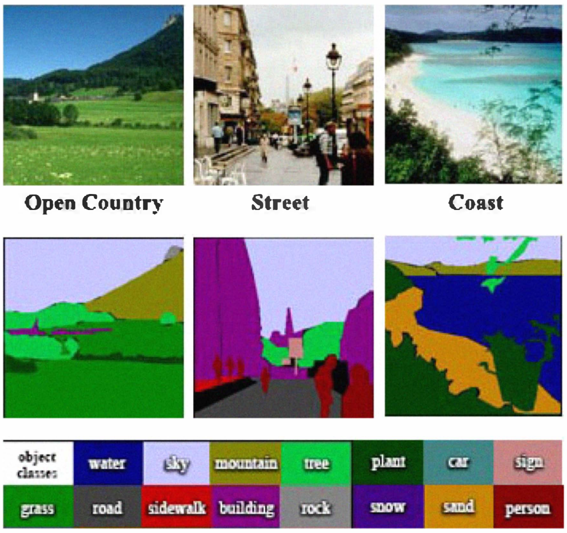

Fig. 1. Example images of several scenes with corresponding groundtruth from the Olive & Torralba database.

II. CM Model

We construct a conditional model (CM) to learn the conditional distribution over the class labeling. In order to model the visual variability of objects with different viewpoint and occlusion, we exploit not only the local features such as appearance and the location of objects but also longer range information as contextual information. The conditional probability of the class label c_i of pixel i in a given image $\mathbf{x}$ is defined as

$$\begin{aligned} P(c_i|\mathbf{x};\boldsymbol{\theta}) \propto \exp\{ & f^{texture}(c_i,\mathbf{x};\boldsymbol{\theta}_0) \\ & + f^{texture-env}(c_i,\mathbf{x};\boldsymbol{\theta}_1) \\ & + f^{location}(c_i,i;\boldsymbol{\theta}_2)\} \end{aligned} \tag{1}$$

$\boldsymbol{\theta} = \{\boldsymbol{\theta}_0,\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\}$ are the model parameters that we estimate from training data. In the following discussion, parameters in every potential f is omitted for clarity and will be introduced separately hereinafter.

A. Texture potential

The texture potential $f^{texture}$ is used to represent appearance of objects. We first convolve the training images with a

978-1-61284-276-9/11/$26.00 ©2011 IEEE

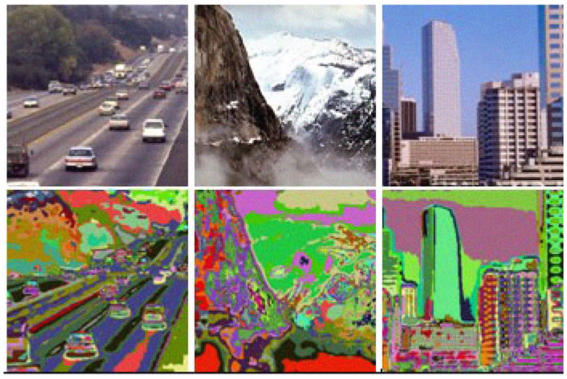

Fig. 2. Example texture maps with original images

17-dimensional filter-bank [12] including scaled Gaussians, x and y derivatives of Gaussians and Laplacians of Gaussians, and then employ the unsupervised $K-$means ($K = 400$) clustering on the filter responses. By assigning each pixel of the images to the nearest cluster center, we finally get the texture maps, several of which are shown in Figure 2 along with their original images. Colors show texture indices in texture maps. We believe texture contains sufficient information for pixel-wise segmentation, thus in this paper, we no longer bring in color or edge potential, which is frequently used in other models.

B. Texture-environment potential

The second potential $f^{texture-env}(c_i, \mathbf{x})$ is named texture-environment potential because it reflects the texture environment of pixel i (including contextual information of different textures of one object and those of other objects in an image), while ignoring the texture of pixel i itself. Aside from the scene index S_t, the texture-environment potential is topologically equivalent to shape-texture potential $\psi_i(c_i, \mathbf{x}; \theta_\psi)(= \log P_i(c_i|\mathbf{x}))$ in [8]. The classification confidence $P_i(c_i|\mathbf{x})$ in this paper is obtained by similar feature-selecting method described in [8], which will be introduced in section D.

C. Location potential

The dependence of the class label on the absolute location of the pixel in the image is also considered in this paper. For instance, sky is frequently observed in the upper part of an image, while grass is often in the lower part in natural images. We obtain this dependence through the location potential $f^{location}$, which we define as

$$f^{location}(c_i, i) = \omega_3 \log \frac{N_{c_i,i} + \alpha_3}{N_i + \alpha_3} \tag{2}$$

where $\{\omega_3, \alpha_3\}$ forms $\boldsymbol{\theta}_2$ and $N_{c_i,i}$ is the number of pixels of class c_i at normalized location i in all images in the training set, while N_i is the total number of pixels at location i in those images.

In this paper, during training and validation, real labels are given. Weights ω_n are all included in their corresponding functions f^n ($f^{texture-env}$, $f^{texture}$ and $f^{location}$) and are treated as part of parameters of the functions. While feature potential functions are separately estimated during training, the weights are obtained during validation.

D. Algorithm

In this paper, each potential is separately trained using piecewise training [13] with powers for efficiency. Parameters $\boldsymbol{\theta}_1$ of the normalized distribution $P_i(c_i|\mathbf{x}) = \frac{\exp H(c_i)}{\sum_{c'_i} \exp H(c'_i)}$ are learned as follows. Feature response at location i is the count of pixels with texture t within a rectangle randomly chosen from 100 rectangles whose four corners are chosen at random within a fixed 200 × 200 pixel wide bounding box centered in pixel i. Feature response at location i is defined as

$$v_{(l,w,x,y,t)}(i) = \frac{\sum_p [T(p) = t]}{area(v)} \tag{3}$$

where $\sum_p$ denotes to sum over every pixel p in the rectangle, $T(p)$ is the texture index of pixel p, and $area(v) = l \times w$ denotes the area of the rectangle. $[x]$ is a binary indicator function, if x is true, $[x] = 1$, otherwise $[x] = 0$. The parameters of rectangle $\theta_v = \{l, w, x, y, t\}$ are chosen by using Joint-Boosting algorithm of [14]. Then using Joint-Boosting algorithm of [14], a strong classifier $H(c)$ is built as a sum of M weak classifiers,

$$H(c) = \sum_{m=1}^{M} h_m(c, v) \tag{4}$$

each of which is a decision stump with parameters $\{\theta_v, C, a, b, \delta, \{k^c\}_{c \notin C}\}$ based on a threshold feature response and is shared among a set of classes C.

$$h_m(c, v) = \begin{cases} a[v_{\theta_v} > \delta] + b & c \in C \\ k^c & c \notin C \end{cases} \tag{5}$$

Here δ is a threshold and the constant k^c is used to counteract the influence of unequal numbers of training examples of each class. A new weak learner is chosen in round m by minimizing an error function incorporating the weights:

$$h_m = arg \min \sum_c \sum_i w^c_{i,m-1}(z^c_i - h(c, v))^2 \tag{6}$$

z^c_i denotes the target value of pixel i ($z^c_i = +1$ if i has groundtruth class c, -1 otherwise). The weights $w^c_{i,m}$ emphasizes poorly classified examples and ensures that the new classifier help the classification approach the target value.

$$\begin{aligned} w^c_{i,m} &= \exp\{-z^c_i H_m(c)\} \\ &= w^c_{i,m-1} \exp\{-z^c_i h_m(c, v)\} \end{aligned} \tag{7}$$

Up to now, the Joint-Boosting algorithm is briefly described, please see [8], [14] for detail. We finally obtain the parameters $\boldsymbol{\theta}_s = \{\{\theta_v, C, a, b, \delta, \{k^c\}_{c \notin C}\}_m\}$. Parameters $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}_0$, $\boldsymbol{\theta}_2$ are manually selected to minimize the error on the validation set. Given a set of parameters learned for the CM model, the result class label c^* of pixel i is the labeling that maximizes the conditional probability in Equation (1). The number of weak classifers, called BoostingNum (M) will be discussed in section IV.

III. sCM Model

By introducing the scene of an image into the conditional model(CM), we construct a new scene-based conditional model(sCM) to learn the conditional distribution over the class labeling. The conditional probability of the class label c_i of pixel i in a given image $\mathbf{x}$ is defined as

$$\begin{aligned}P(c_i|\mathbf{x}, S_t; \boldsymbol{\theta}) \propto \exp\{& f^{texture}(c_i, \mathbf{x}, S_t; \boldsymbol{\theta}_0) \\ &+ f^{texture-env}(c_i, \mathbf{x}, S_t; \boldsymbol{\theta}_1, \boldsymbol{\theta}_s) \\ &+ f^{location}(c_i, i, S_t; \boldsymbol{\theta}_2)\}\end{aligned} \tag{8}$$

Scene index $S_t \in \boldsymbol{S_t}$ represents the scene of image $\mathbf{x}$. $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_s, \boldsymbol{\theta}_0, \boldsymbol{\theta}_2\}$ are the model parameters that we estimate from training data.

A. Texture potential

By introducing the scene of an image texture potential $f^{texture}(c_i, \mathbf{x}, S_t)$ is redefined as

$$f^{texture}(c_i, \mathbf{x}, S_t) = \omega_2 \log \frac{N_{c_i,t,S_t} + \alpha_2}{N_{t,S_t} + \alpha_2} \tag{9}$$

where $\{\omega_2, \alpha_2\}$ forms $\boldsymbol{\theta}_0$ and N_{c_i,t,S_t} is the number of pixels of class c_i with texture t in all images with scene S_t in the training set, while N_{t,S_t} is the total number of pixels with texture t in those images.

B. Texture-environment potential

Introducing the index S_t, We define a new normalized distribution as

$$\begin{aligned}P_i(c_i|\mathbf{x}, S_t) &= \frac{P_i(c_i|\mathbf{x})P(c_i|S_t)}{\sum_{c_i' \in S_t} P_i(c_i'|\mathbf{x})P(c_i'|S_t)} \\ &\propto P_i(c_i|\mathbf{x})P(c_i|S_t)\end{aligned} \tag{10}$$

where $c_i \in S_t$ indicates that class c_i exists in some images with scene S_t. Then we obtain the definition of texture-environment potential

$$\begin{aligned}f^{texrure-env}(c_i, \mathbf{x}, S_t) &= \log P_i(c_i|\mathbf{x}, S_t) \\ &= \log P_i(c_i|\mathbf{x}) + \log P(c_i|S_t) \quad (c_i \in S_t)\end{aligned} \tag{11}$$

where $\log P(c_i|S_t) = \omega_1 \log \frac{N_{c_i,S_t}+\alpha_1}{N_{S_t}+\alpha_1}$, with parameters $\boldsymbol{\theta}_s = \{\boldsymbol{\omega}_1, \boldsymbol{\alpha}_1\}$. N_{c_i,S_t} is the number of pixels of class c_i in all images with scene S_t in the training set, while N_{St} is the total number of pixels in those images.

C. Location potential

Including the index S_t, we redefine the location potential as

$$f^{location}(c_i, i, S_t) = \omega_3 \log \frac{N_{c_i,i,S_t} + \alpha_3}{N_{i,S_t} + \alpha_3} \tag{12}$$

where $\{\omega_3, \alpha_3\}$ forms $\boldsymbol{\theta}_2$ and N_{c_i,i,S_t} is the number of pixels of class c_i at normalized location i in all images with scene S_t in the training set, while N_{i,S_t} is the total number of pixels at location i in those images.

Parameters are estimated by the same method described in the former section. At test time, the labels would be chosen by maximizing the conditional probability in Equation (8).

TABLE I
Comparison of the average pixel prediction accuracy with different boostingNum M in validation set .

	$M = 500$	$M = 5000$
CM	57.7%	59.3%
sCM	74.3%	73.5%

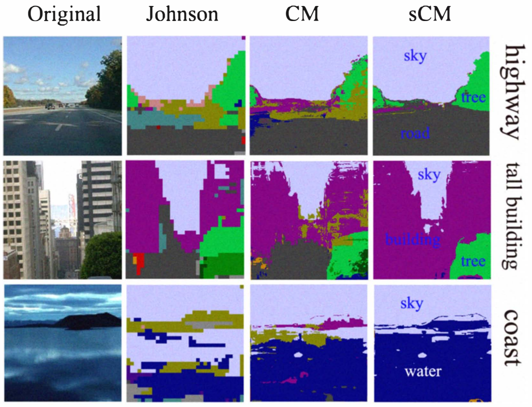

Fig. 3. Example results on the Olive & Torralba database. The first column shows original images, second column givesthe recognition and segmentation results of Johnson's model [15], third column shows results of our CM model, while fourth column shows results of our sCM model.

IV. Results & Discussion

In this paper, during training and validation, real labels are given and parameters are estimated by minimizing the error rate of inferred labels. We select $\boldsymbol{\theta}_s = \{1, 1\}$, $\boldsymbol{\theta}_0 = \{2, 1\}$, $\boldsymbol{\theta}_2 = \{2, 1\}$ in this paper. Weights ω_n are all included in their corresponding functions f_n and are treated as part of parameters of the functions. While feature potential functions are separately estimated during training, the weights are obtained during validation.

We also investigate how boostingNum M effects classification accuracy in the validation set of Olive & Torralba database. We gradually increase M and gain satisfactory classification result with $M = 500$, however, accuracy is barely improved with further increase of M. Table I shows that with M increased from 500 to 5000, the prediction accuracy of CM model increases only a little, while the accuracy of sCM model even decreases. Besides with $M = 5000$, the algorithm is quite time-consuming. Thus the results with respect to the test set described in the following, we always choose the boostingNum $M = 500$.

Fig. 3 shows the comparison between our model to that of Johnson's [15] on Olive & Torralba database. Several images and their results are presented in Fig. 3, which show that most parts of the images are well recognized and segmented. Especially, in the image in the second row in Fig. 3, building and sky are both well segmented, though the boundary of building and sky is very blurry. Table II compares the performance of Johnson's model [15], our CM model and sCM model over the whole database. Our CM model improves the classification

TABLE II
COMPARISON OF THE ACCURACY OF OUR CM MODEL AND sCM MODEL WITH JOHNSON'S MODEL ON THE TEST SET OF OLIVE & TORRALBA DATABASE.

Algorithm	15-class accuracy
Johnson	48%
CM	57.3%
sCM	73.3%

accuracy from 48% to 57.3%, while the sCM model obtains much higher accuracy 73.3%, indicating that our CM and sCM model clearly outperforms Johnson's model.

V. CONCLUSION

In this paper, we first construct a CM model to incorporate different features of objects. Then based on the CM model, we present a new scene-based CM model called sCM model, along with a texture-environment potential. We investigate these two models on multiple class recognition and segmentation on the Olive & Torralba database. Comparison results of different methods indicate that our models outperforms Johnson's work. Especially, aside from improved accuray, our sCM model gains greatly improved perceptual result, with boundary of different objects correctly labeled.

REFERENCES

[1] Z.F. He, T.N.Tan, Z.N.Sun, Topology modeling for Adaboost-cascade based object detection. Pattern Recognition Lett. 31,2010, 912−919.

[2] N. Bourbakis, P. Yuan, S. Makrogiannis, Object recognition using wavelets, L-G graphs and synthesis of regions, Pattern Recognition 40(7) (2007) 2077−2096.

[3] S. Kim, K.J. Yoon, I.S. Kweon, Object recognition using a generalized robust invariant feature and Gestalt's law of proximity and similarity, Pattern Recognition 41(2) (2008) 726−741.

[4] J. Winn, J. Shotton, The layout consistent random field for recognizing and segmenting partially occluded objects, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 37−44.

[5] A. Opelt, A. Pinz, A. Zisserman, Incremental learning of object detectors using a visual shape alphabet, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 3−10.

[6] J. Winn, N. Jojic, LOCUS: Learning object classes with unsupervised segmentation, in: Proceedings of International Conference on Computer Vision. vol. 1, 2005, pp. 756−763.

[7] P. Kumar, P. Torr, A. Zisserman, Obj cut, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 18−25.

[8] J. Shotton, J. Winn, C. Rother, Criminisi A, TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation, in: Proceedings of European Conference on Computer Vision, vol. 3951, 2006, pp. 1−15.

[9] S. Gould, J. Rodgers, D. Cohen, G. Elidan, D. Koller, Multi-Class Segmentation with Relative Location Prior, International Journal Computer Vision 80(3) (2008) 300−316.

[10] L. Yang, P. Meer, D.J. Foran, Multiple class segmentation using a unified framework over mean-shift patches, in: Proceedings of IEEE conference on Computer Vision and Pattern Recognition, 2007. 1−8.

[11] F. Schroff, A. Criminisi, A. Zisserman, Object Class Segmentation using Random Forests, in: Proceedings of the 19th British Machine Vision Conference, 2008.

[12] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary. in: Proceedings of International Conference on Computer Vision. Vol. 2, 2005, pp. 1800−1807.

[13] C. Sutton, A. McCallum. Piecewise training of undirected models, in: 21st Conference on Uncertainty in Artifcial Intelligence, 2005, pp. 568−575.

[14] A. Torralba, K.P. Murphy, W.T. Freeman, Sharing visual features for multiclass and multiview object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5) 2007 854−869.

[15] M. Johnson. Semantic Segmentation and Image Search. Phd Thesis, University of Cambridge, 2008.