FROM VIDEO TO TEXT: SEMANTIC DRIVING SCENE UNDERSTANDING USING A COARSE-TO-FINE METHOD

Huiyuan Fu, Huadong Ma

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing University of Posts and Telecommunications, Beijing 100876, China
fu_hui_yuan@yahoo.com.cn, mhd@bupt.edu.cn

ABSTRACT

Semantic understanding from video is one of the most challenging tasks in video analysis. However, it has not been taken enough attention. In this paper, we focus on understanding the semantics of video in the driving scene. We present a coarse-to-fine method to parse the driving scene, and obtain the high-level semantic information of the scene. In the coarse phase, we divide the captured frame into four separate parts based on edge density entropy and scene context. In the fine phase, we join multi-class object segmentation and detection algorithms together in a unified Conditional Random Filed (CRF) model for each part understanding. Moreover, the object probabilistic location prior knowledge based on training and previous edge density entropy result is also integrated into our approach for better object localization. Experimental results show that our proposed method is effective comparing to current state-of-the-art approaches.

Index Terms— Semantic understanding; Conditional Random Filed; multi-class segmentation; detection

1. INTRODUCTION

Semantic Scene understanding [1, 2, 3, 4] from video is one of the central goals in video analysis. However, because it is so tightly based on accurate object detection and segmentation that it has not been substantially developed for a long time. Recently, in light of the successes of detection [5] and segmentation [6] technologies, it has received increasing attentions. Brostow [1] et al. proposed a method for understanding the scene according to motion features after segmentation and detection of objects, and their experiment results show it is effective. Li [2] et al. tried to find an united model for total scene understanding by combining classification, annotation and segmentation together. To synthetically use many kinds of information of the scene, Hoiem [3] et al. presented an approach based on scene context. However, the output semantic information of these methods is imperfect because they merely focus on *what* and *where*. Ladicky [4] et al. presented an approach by combining the object detectors and CRFs, their work is impressive. However, their work focused more on the effect of segmentation results by fusing the detectors directly.

In this paper, we concentrate on understanding the driving scenes because of its broad applications and its typical representative means for other scenes. Our objective is to obtain the semantic information of the driving scene, such as *what*, *where*, *which* and *how many*: we distinguish each detected object, get their specific positions, know current particular road scene, and also compute out the amount of each object. In our proposed coarse-to-fine method, we first divide the captured image into four separate parts based on edge density entropy and scene context in the coarse phase. Then, we integrate the state-of-the-art multi-class segmentation [6] and object detection [5] algorithms into an unified CRF [7] model for each part understanding in the fine phase. Moreover, we also incorporate the object probabilistic location prior knowledge based on previous edge density entropy result and both the off-line and on-line training into our approach for better object localization. These object location based priors can correct the false object detection. Our experiments show that it is very helpful for obtaining the *how many* information. In our work, we classify the objects into two categories: things and stuff according to the definition of [8]. We detect [5] and localize the things by bounding boxes, such as cars, pedestrians and bicycles. Meanwhile, we segment both things and stuff using the multi-class segmentation approach [6]. We show that our method is better than two state-of-the-art approaches in the experimental section.

The rest of this paper is organized as follows: Section 2 describes our method of splitting the scene in the coarse phase, and Section 3 presents our method of understanding the scene in the fine phase, respectively. Experimental results and analysis are described in Section 4. Finally, Section 5 concludes the work.

This work is supported by the National Science Fund for Distinguished Young Scholars under Grant No.60925010, the National Natural Science Foundation of China under Grant No.60833009, No.60903072, and No.61003280, the Funds for Creative Research Groups of China under Grant No.61121001 and the Program for Changjiang Scholars and Innovative Research Team in University under Grant No.IRT1049.

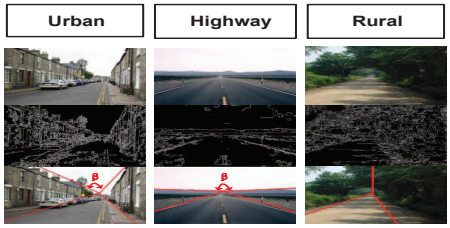

Fig. 1. Coarse split of driving scene based on edge density entropy (binary image) and context information.

2. DRIVING SCENE SPLITTING IN COARSE

To obtain *which* information, we split the captured scene into four parts: left, right, top and bottom first, and we call it the coarse phase. Inspired by [9], we split all of driving scenes into three categories: urban scene, high way scene and rural scene(see Fig. 2), but our method is based on the edge density entropy and context of each scene.

First, we compute the edge density entropy based on segmentation algorithm [10] (see Fig. 1) by labeling the binary maps of the segmentation results. When we get the binary maps, we can straightly know that the sky and load parts have a lower edge density entropy. So we can easily obtain the center region that combining with sky and load. Once the center region is found, we can use it to split the scene in coarse. As seen in Fig. 1, three representative scenes are divided into four parts from the point which is the center of above center region. The four intersection angles, especially the top intersection angle β are the useful features for classifying different scenes (see Fig. 1). However, we may fail to distinguish the different scenes which have the similar angle β when using segmentation in some instances. At this time, we will incorporate context information into above approach for better scene classification. For example, we may use the area ratio of sky to the whole image, the area ratio of building to the whole image, etc. In this way, we can obtain the information of *which* scene.

In fact, above splitting of scene strategy in coarse is very useful for our proposed probabilistic location priors-based correction in the work.

3. DRIVING SCENE UNDERSTANDING IN FINE

3.1. Combine multi-class segmentation and detection

CRF [7] framework has become increasingly popular for modeling object segmentation problems because of providing a principled way to integrate things and stuff.

Multi-class object segmentation. TextonBoost proposed by Shotton et al. [6] is one of the state-of-the-art multi-class segmentation algorithms which combines recognition and image segmentation together. They use a boosted combination of texton features to encode the shape, texture and appearance of the object classes. A CRF was then used to combine the result of texton with colour and location based likelihood terms. The conditional probability of the class labels b and the given image Y can be defined as follows:

$$\log P(b|Y,\overline{w}) = \sum_i \gamma_i(b_i, Y; \overline{w}_\gamma) + \pi(b_i, Y_i; \overline{w}_\pi) + \lambda(b_i, i; \overline{w}_\lambda) + \sum_{(i,j)} \varphi(b_i, b_j, \mu_{i,j}(Y); \overline{w}_\varphi) - \log \chi(\overline{w}, Y) \quad (1)$$

where $\sum$ is the set of edges in the 4-connected grid, $\chi(\varpi, Y)$ is the partition function. γ_i, π, λ and φ means shape, color, location and edge, respectively. $\varpi = \varpi_\gamma + \varpi_\pi + \varpi_\lambda + \varpi_\varphi$ are the model parameters, and i and j index nodes in the grid (corresponding to positions in the image) [6]. Based on the likelihood terms of CRF, it is useful for labelling problem in multi-class segmentation. The important problem is to find the min Energy functions for segmentations. Energy functions for object segmentation can be defined as follows:

$$E(X) = \sum_{i \in \kappa} \varphi_i(X_i) + \sum_{(i,j) \in \delta} \varphi_{ij}(X_i, X_j) \quad (2)$$

where φ_i means the unary relation for pixel X_i, φ_{ij} means the pairwise relation for pixel X_i and pixel X_j. κ and δ are the pixel sets. The labels are used for indicating the multi-class segmentation results. Details can be seen in [6].

Object detection combined in CRF framework. In order to get the sematic information of *What* and *Where*, we also need to use the state-of-the-art object detection approach [5] for localization. Sometimes, the segmentation algorithm and detection algorithm both do well. So we can use this information to obtain the sematic information of *What* and *Where* very conveniently. But this is only the well instance. In fact, the results are not so good for most of time. As seen in Fig. 2, the segmentation algorithm and detection algorithm not both do well meantime. When the segmentation algorithm does well, the detection algorithm is wrong(see the right of Fig. 2). Meanwhile, When the detection algorithm does well, the segmentation algorithm is wrong(see the middle of Fig. 1). What can we do to deal with this issue? We propose a probabilistic object location priors based method for correcting the wrong instances above.

3.2. Our probabilistic location priors for correction

As seen in Fig. 2, current state-of-the-art detection [5] and segmentation [6] algorithms have the problem of running successfully together all-time. Thus, we are not able to achieve satisfied *What* and *Where* information. Meanwhile, this will also result in wrong statical analysis on *How many* information. To solve this problem, we bring a probabilistic object-wise based localization priors into our approach. In our mechanism, we consider the location priors based on prior coarse split results (see Fig. 3).

Off-line Training. The CamVid [11] database is used for training in our experiment. It consists of 101 960 $\times$ 720 pixel images in which each pixel was manually assigned to one

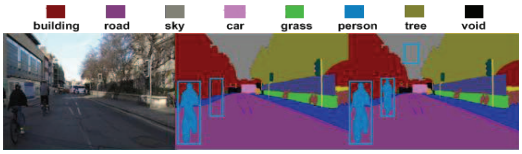

Fig. 2. Two cases of false object detection or segmentation. (left) input image from CamVid [11]; (middle) detection is right, but segmentation is wrong; (right) segmentation is right, but detection is wrong. (Best viewed in colour)

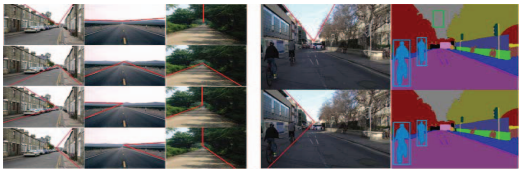

Fig. 3. Our probabilistic object location priors for correction. (left) each scene is divided into four parts as priors; (right) the correction results of our method. (Best viewed in colour)

of the 32 object classes(colour labeled in Fig. 2, here show 8 categories) that are relevant in a driving environment. Our training mechanism is composed of two parts: off-line training and on-line training. First, we choose 500 pictures from the CamVid database, and compute out the statical probability of the each object class (note that the objects are only things, such as pedestrians and cars). As seen in left of Fig. 3, each scene is divided into four parts for off-line training.

On-line Training. When we get the initial probability of each object, we use the probability to as a feedback for correct object localization. As seen in the top-right of Fig. 3, the false object detection labeled in green is not elected by our probabilistic location priors. As seen in the bottom-right of Fig. 3, the false segmentation person is been found again. Meanwhile, we will add each probability into initial prior probabilities when the detection and segmentation algorithms both localize it. We call it on-line training. In this way, we will improve the accuracy both for detection and segmentation results.

4. EXPERIMENTAL RESULTS

4.1. Datasets

Training Data. The CamVid [11] video database which contains typical driving scenes is used for training in our experiment. As seen in Fig. 8, we label each pixel in CamVid as one of 8 categories: building, road, sky, car, grass, person, tree and void. **Testing Data**. Part of our testing video data are captured from the driving car, and the other part of testing video data (or images) are captured from the internet.

4.2. Results analysis

The goal of the paper is to understand the sematic information: *Which*, *What*, *Where* and *How many* in the videos of the driven scene in text format. Based on our experiments, the final complete text format result containing these important semantic information will be generated finally.

Which. As seen in Fig. 1, we have shown how to obtain *Which* information. We divide the driving scene into three categories: urban scene, highway scene and rural scene. Now we estimate the performance of our method using a confusion matrix (see Fig. 4) in total collected 200 driving scene videos for test. The experimental result can be seen in Fig. 4. We can obtain a high overall accuracy about 91.4% comparing to the ground truth results. We find that the recognition of highway scene for *Which* information is more correct than the urban scene and rural scene.

What and Where. In our experiment, we use HOG- LBP [5] based detection method to detect the things which contain Pedestrians, Cars and Bicyclists. To get the information of *Where*, we take TextonBoost [6] based multi-class segmentation method to obtain locations of objects contain- ing of both things and stuff. We compare the detection results of cars and pedestrians in three methods: only HOG-LBP, HOG-LBP combine with TextonBoost and HOG-LBP both combine with TextonBoost and our proposed probabilistic location priors based correction, respectively. The result can be seen in Fig. 5. We use FPPI (false positives per image) versus miss rate to estimate the performance of these three algorithms. We find that our method can detect cars and pedestrians better than other two state-of-art approaches (for example, our method has a lower miss rate at the same FPPI), so it can obtain better semantic information *What* and *Where*.

How many. In order to estimate our method for obtaining *How many* information, we compare our method with HOG-LBP [5] method and the combined algorithm of HOG-LBP and TextonBoost [6] for counting the cars and pedestrians, respectively. We both testify them in sparse scenes (means the scene contain low flow of cars and pedestrians) and dense scenes. The experimental results can be seen in Fig. 6. They show that our proposed method is more robust than above two typical algorithms for getting a better How many information.

Final result. Based on prior result of *Which*, *What*, *Where* and *How many*, we can get a total understanding of a driving scene in the input video. We first obtain Which based on edge density entropy. Then, we get the What, Where and How many using the combination of HOG-LBP [5], TextonBoost [6] and probabilistic location priors. As seen in Fig. 7 and Fig. 8, we present a visual procedure of our proposed method for obtaining the above semantic information. By combing them together, we can export the whole of sematic information in text (see Fig. 8)

5. CONCLUSIONS

In this paper, we have proposed a new approach for understanding the video semantics in text format. Specifically, we focus on the driving scene video as a case, using our presented

Fig. 4. Comparison on *Which*. Our algorithm can reach an overall accuracy of 91.4% comparing to the ground truth results in the testing videos.

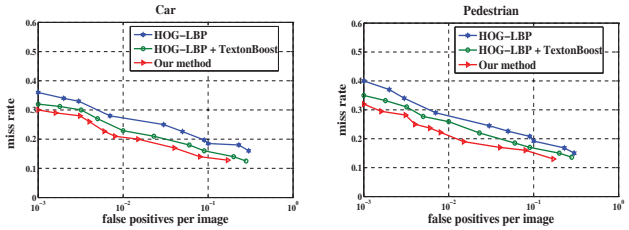

Fig. 5. Comparison on *What and Where*. Our algorithm compares with two state-of-art algorithms for car detection (left) and pedestrian detection (right) in the testing videos.

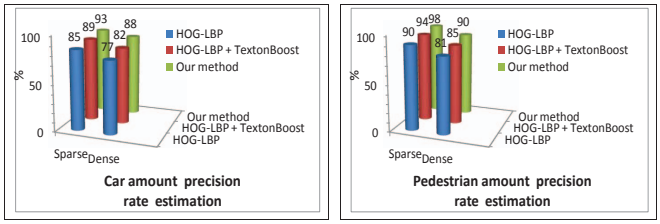

Fig. 6. Comparison on *How many*. Our algorithm compares with two state-of-art algorithms for car amount (left) and pedestrian amount (right) estimation in the testing videos.

coarse-to-fine method. Currently, we mainly work on images which constitute the video directly. In the future, we will try to obtain *When* information as an effective complement between the frames.

6. REFERENCES

[1] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," *IEEE European Conference on Computer Vision*, vol. 1, pp. 44–57, 2008.

[2] L. J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," *IEEE European Conference on Computer Vision*, 2009.

[3] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 7, 2006.

[4] L. Ladicky, C. Russell, P. Sturgess, K. Alahari, and P. H. S. Torr, "What, where and how many? combining object detectors and crfs," *IEEE European Conference on Computer Vision*, 2010.

[5] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," *IEEE International Conference on Computer Vision*, vol. 1, pp. 1–8, 2009.

[6] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," *IEEE European Conference on Computer Vision*, pp. 1–15, 2006.

[7] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," *IEEE International Conference on Machine Learning*, vol. 18, pp. 282–289, 2001.

[8] D. A. Forsyth, J. Malik, M. M. Fleck, T. Leung, C. Bregler, C. Carson, and H. Greenspan, "Finding pictures of objects in large collections of images," *IEEE European Conference on Computer Vision*, pp. 335–360, 1996.

[9] B. Yao, X. Yang, L. Lin, M. W. Lee, and S. C. Zhu, "I2t: image parsing to text description," *Proceedings of IEEE (Invited)*, vol. 98, pp. 1485–1508, 2010.

[10] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[11] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A highdefinition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

Fig. 7. (left) The procedure of Which; (right) The procedure of What and Where. (Best viewed in colour)

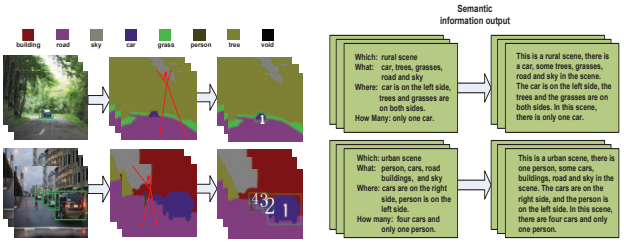

Fig. 8. (left) The procedure of How many; (right) Semantic text information is generated finally based on fusing obtained semantic information automatically. (Best viewed in colour)