Multi-view Vehicle Detection based on Part Model with Active Learning

Mukesh Prasad[1], Chih-Ling Liu[2], Dong-Lin Li[2], Chandan Jha[1], Chin-Teng Lin[1], *IEEE, Fellow*

[1]*Centre of Artificial Intelligence, School of Software, FEIT, University of Technology Sydney, Australia*
[2]*Department of Electrical Engineering, National Chaio Tung University, Hsinchu, Taiwan*

***Abstract*—Nowadays, most of the vehicle detection methods aim to detect only single-view vehicles, and the performance is easily affected by partial occlusion. Therefore, a novel multi-view vehicle detection system is proposed to solve the problem of partial occlusion. The proposed system is divided into two steps: background filtering and part model. Background filtering step is used to filter out trees, sky and other road background objects. In the part model step, each of the part models is trained by samples collected by using the proposed active learning algorithm. This paper validates the performance of the background filtering method and the part model algorithm in multi-view car detection. The performance of the proposed method outperforms previously proposed methods.**

***Index Terms*—Vehicle Detection, Active Learning, Multi-view, Part Modeling, Background Filtering**

I. INTRODUCTION

With the ever-increasing number of vehicles in use, road transportation systems have to deal with problems such as long traffic jams, car accidents or other traffic rules violations. Road transportation systems have become significantly smart with recent advancements in image processing, computer vision, and machine learning techniques. Vehicle detection system is one such application that has helped to solve the majority of the problems being faced by the road transportation systems.

Background subtraction is an extensively researched method to detect moving objects in a vehicle detection system using static cameras. In the past, reference [1] used pixel grey scale difference between the consecutive image frames to detect moving objects. The method is susceptible to changes in light and shadow. Kalman filter based background filtering methods [2], [3] have also been widely used. It is a recursive background subtraction method. These methods cannot be applied to still images or still vehicles. They are applicable only to dynamic images or the background must be established in advance.

Wavelet transform based method [4] in combination with PCA classifier was proposed to obtain vehicle texture features and further separate the vehicle from the road area. Reference [5] used a generic Gabor filter to capture the edge characteristics in different proportions of size and rotation angle, and perform classification with Support Vector Machines (SVM). However, finding an optimum set of Gabor filters put a constraint on the detection speed.

In order to improve the detection speed, [6] used color transform to filter out background and preserve the area that might be vehicles. Reference [7] used the Histogram of Oriented Gradients (HOG) to record the gradient distribution of the vehicle's edges, and train the classifiers using SVM with a variety of kernel functions.

Most methods used the entire image to capture features and therefore, cannot effectively combat the problem of partial occlusion. For these reasons, methods [8-11] proposed part model methods to perform object detection. Reference [8] came up with the deformable part model (DPM), which splits the object into several parts, and uses HOG feature combined with SVM to train the classifier for each part. Reference [9] proposed 3-layer part model respectively root, first layer parts, and second layer parts. These methods detect objects by splitting parts and use the relative object position information to improve object detection accuracy. Part model methods have been proposed to be used in vehicle detection system [12], [13] to solve the problem of partial occlusion.

In this paper, a two-stage vehicle detection method is used. In the first stage which is a preprocessing stage, background subtraction is done by using color transform method in order to speed up the process of detection and improve the accuracy. In the second stage, the problem caused by the diversity of vehicles and partial occlusion is solved using the concept of the deformable part model [8]. An active learning algorithm is proposed to collect the part samples to ensure that part models are robust enough. This method can make the system more suitable to the dataset, and train a more robust model for vehicles. The details of each stage are explained in later sections.

II. PROPOSED SYSTEM

This section gives a detailed description of the proposed vehicle detection system. The system is divided into three parts, Background Filtering, building of the Part Model, and detection as shown in Fig. 1.

A. Background Filtering Method

By analyzing most vehicle scene, the background objects are usually trees, sky, and roads. Herein, the background objects are filtered out using color transform and smooth area detection to increase the detection accuracy and speed up the system.

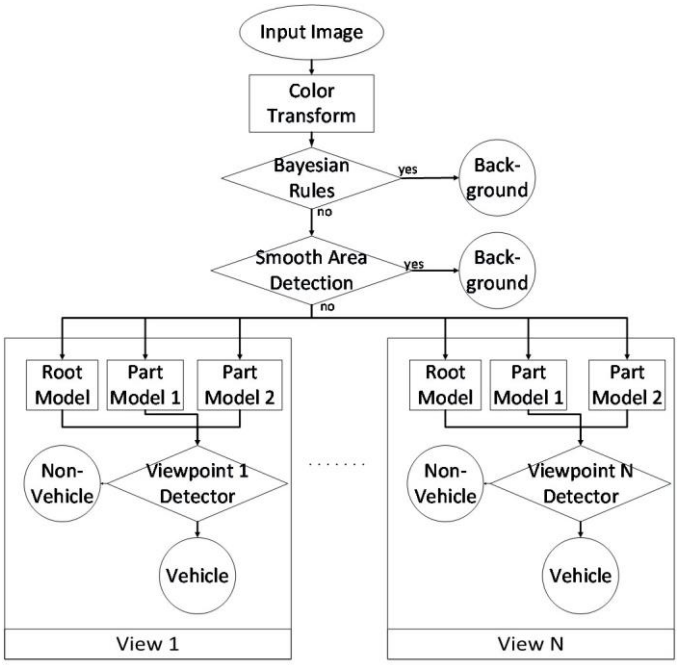


Fig. 1. System Flowchart

1. Color Transform

Color transform method used is same as in [14]. Training sample images have been used from CBCL Street Scenes dataset [15], as shown in Fig. 2.

To reduce the amount of data and noise, the size of all images is set as 35 x 35 and their statistics of their RGB value is calculated. The covariance matrix of the images are calculated as

$$\Sigma = \begin{matrix} E[(R-\mu_R)(R-\mu_R)] & E[(R-\mu_R)(G-\mu_G)] & E[(R-\mu_R)(B-\mu_B)] \\ E[(G-\mu_G)(R-\mu_R)] & E[(G-\mu_G)(G-\mu_G)] & E[(G-\mu_G)(B-\mu_B)] \\ E[(B-\mu_B)(R-\mu_R)] & E[(B-\mu_B)(G-\mu_G)] & E[(B-\mu_B)(B-\mu_B)] \end{matrix}$$

where μ is the average value of this color, and E(.) is the calculation of expected value. The eigenvector with respect to the largest eigenvalue is (0.33, 0.33, 0.33). The other two eigenvectors are (0.77, -0.6, -0.17) and (0.25, 0.55, -0.8). Therefore, the transform equation is derived as

$$u_p = \frac{0.77R_p - 0.6G_p - 0.17B_p}{z_p} \quad \text{and} \quad v_p = \frac{0.8B_p - 0.25R_p - 0.55G_p}{z_p} \quad (1)$$

(a) (b)

Fig. 2. Training sample of color transform (a) Tree images (b) Non-tree images

This transform equation can be used to transform the RGB value of each pixel p into (*u*,*v*) domain, where

$Z_p = (R_p + G_p + B_p)/3$ is used for normalization. However, the noise would increase if RGB color components are used separately rather than together. Therefore, the first component is replaced by grey level Z_p and the new transform equation is derived as

$$u_p = \frac{0.77z_p - 0.6G_p - 0.17B_p}{z_p} \quad \text{and} \quad v_p = \frac{0.8z_p - 0.25R_p - 0.55G_p}{z_p}. \quad (2)$$

As in Fig. 3, the color of tree pixels would gather in a smaller area than the color of non-tree pixels. Bayesian rules [1] are used to determine whether each pixel is tree or not.

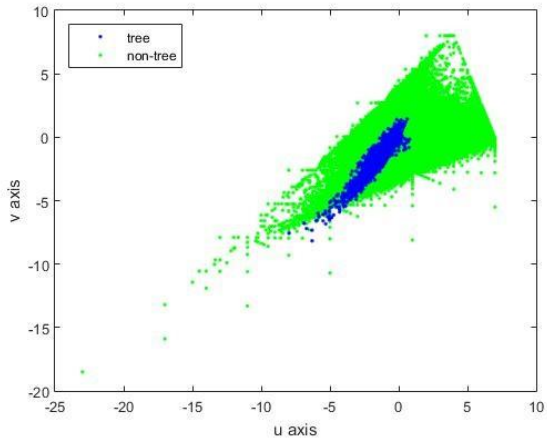


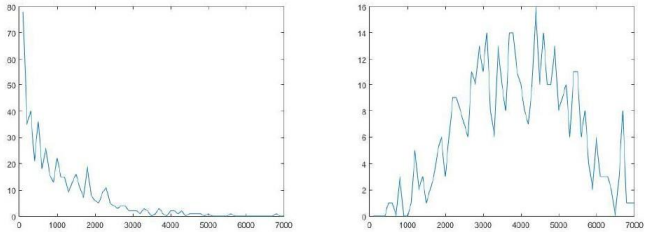
Fig. 3. Color distribution of tree and non-tree pixels in (*u*,*v*) domain

2. Smooth Area Detection

To filter out road and sky these background, road and sky images from CBCL Street Scenes Challenge Framework dataset [15] are used variance distribution in each image is calculated, as in (3)

$$Variance = \frac{\sum(X\text{-}mean)^2}{area} \tag{3}$$

According to the statistic, it is found that the variance of these images mainly concentrate close to zero, as in Fig. 4. The vehicle variance distribution in this dataset [15] is calculated, and the result shows that the variance is very widely distributed, but the minimum value is around 500. Therefore, the minimum is set as the threshold, and sliding window is used to scan through the whole image. If the variance is smaller than the threshold, it is smooth area, and is filtered out.



(a) (b)

Fig. 4. (a) Road and sky variance distribution (b) Vehicle variance distribution

B. Model Construction

At first, the vehicle samples are separated into 8 viewpoints to build models, as in Fig. 5. The vehicle in each viewpoint is composed of a root model and several part models, built by SVM [16].

1. Active Learning for Part Sample Labeling

In order to increase the number of samples and enhance the training effect, the number of viewpoint models is reduced from eight to five viewpoint models, by horizontally flipping over the symmetric vehicles, respectively front, front side, side, back side, and back viewpoints. These viewpoint groups are used to perform the following part sample labeling. It is found that these viewpoints are unique for each vehicle type and models. Further, each viewpoint is identified by cropping them by the parts they contain as shown in Fig. 6. For example, the front viewpoint has front windshield, left mirror (including the left mirror with part of windshield and hood), the right mirror (including the right back mirror with part of windshield and hood), and radiator; likewise for the other viewpoints.

For the cropping method of all part samples, an active learning for part samples labeling algorithm is proposed. Algorithm 1 is the proposed active learning algorithm. In order to reduce human intervention and also the part samples are not specific enough, only 20 vehicle images are provided for part sample labeling during the initial learning of the model. While learning, the number of sample for human to label each time is 5 positive samples and the negative samples are generated by these positive samples. The learning model adopts SVM [16] together with HOG feature [17]. These 5 positive samples are labeled from 4 vehicle images including some unsure samples detected by the learned model, which means there are some samples in this image that their responses are between (-0.1, 0.1), and one randomly chosen unlabeled vehicle image so as to avoid the situation of neglecting some samples, whose appearances are more special and have never been learned. These vehicle images are used to label their positive part samples, and the negative samples would be generated automatically by calculating the overlapping area with the positive sample. There are three termination conditions in this algorithm: the iteration times reaches 5, all vehicle images have been labeled, or the response value of the samples detected by the model are all located outside of (-0.1,0.1).

Once the positive and negative samples from the above algorithms are obtained, the learned part model is used to perform detection on the labeled vehicle images.

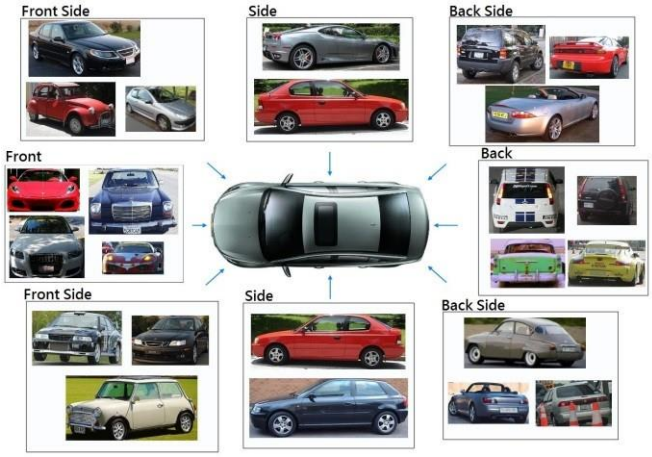


Fig. 5. Illustration of 8 vehicle viewpoints in Pascal VOC 2007

Fig. 6. Parts cropping of 5 viewpoint vehicles

2. Construction of Part Model

HOG feature is extracted from the collected training samples for each part model. SVM is then used to train each part model for detection. While the part model trains, there are too many negative samples in comparison to the positive samples resulting in serious data imbalance problem. The problem of data imbalance grows with increasing appearances of similar background. Bootstrapping method [18] is used to reduce the data imbalance problem. The initial filter is trained by using the positive samples and randomly chosen negative samples, and the initial filter detects the original image again. Then, the false positive samples are added to the original negative samples and the filter is trained with the positive samples again. These steps are repeated until the F-measure of the detection does not increase further. This method makes the filter more adaptive to the dataset and completes the final filter.

Active Learning Algorithm for Part Sample Labeling

Given:

B: number of examples to be selected
L: set of labeled samples
P: set of unlabeled samples
V: set of vehicle images with unlabeled samples

Algorithm:

loop until stopping criterion is met

1. Learn model M from L
2. Detect V with model M, and derive the response
3. Normalize positive r of each P to $(0,1)$, and
4. Select B samples from P, whose r is between $(-0.1,0.1)$
5. Query human annotator for labels of the vehicle images with these B samples
6. Label samples from P, whose r is between $(0.8,1)$, as positive sample, and generate the other negative samples according to this positive sample.
7. Move newly labeled samples from P to L, and remove the vehicle images with these labeled samples from V

return L

3. Fusion of Part Model

To determine whether a sliding window is a vehicle or not, both the appearance and position information of all the part models and root model needs to be considered. The statistics of the relative position of each part model with respect to the vehicle model of that viewpoint are calculated. Since the number of training samples is limited, Gaussian distribution is used, to sum up, the relative position and normalize the distribution results into 0 and 1. At last, the relative position distribution map of each part model is derived. For example, Fig. 7 is the relative position distribution map of each part models in the front viewpoint vehicle.

While the system performs detection, it would derive the relative position of the object, detected by each part models or root model, and use the relative position according to the distribution map of this part or root model to give weight to this object. The score of this object is calculated by the weight multiplying the response value. If there is more than one object detected by the same part model in a sliding window, the one with the highest score is selected. At last, all the score of each part and root in this sliding window is summed, and then the final response value of this sliding window is obtained. In this way, each part models and root model are combined.

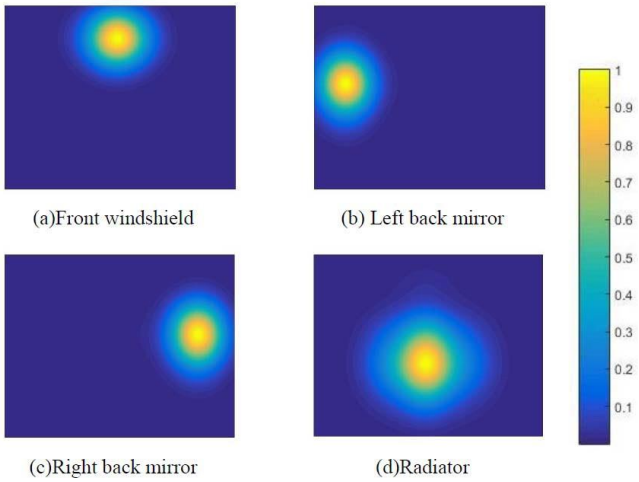


Fig. 7. Relative position distribution map of each part models from the front viewpoint vehicle

C. Detection System

Once all the models are constructed, the input images are inserted to perform detection. The detection flowchart is shown in Fig. 9. First, the input image would be processed by the color transform, and then Bayesian rule [19] is used to label each pixel as tree or non-tree. Then, use the sliding window to scan through the whole image. If the tree pixels are more than 50% in this sliding window, this area would be viewed as background and filtered out. Then the image comes to the detection of each vehicle models. The vehicle model of each viewpoint includes root model and several part models. These filters are used to detect and calculate the scores of each detected objects. Once the detection scores of all objects are obtained, the sliding window is used again to scan the image and calculate the final score of this sliding window. If the final score is higher than the threshold, this sliding window would be determined as a vehicle. The setting of the threshold can be derived using training data to perform detection. The threshold which makes the F-measure achieve the highest would be used. Finally, all the detected vehicles are merged and the final result is obtained.

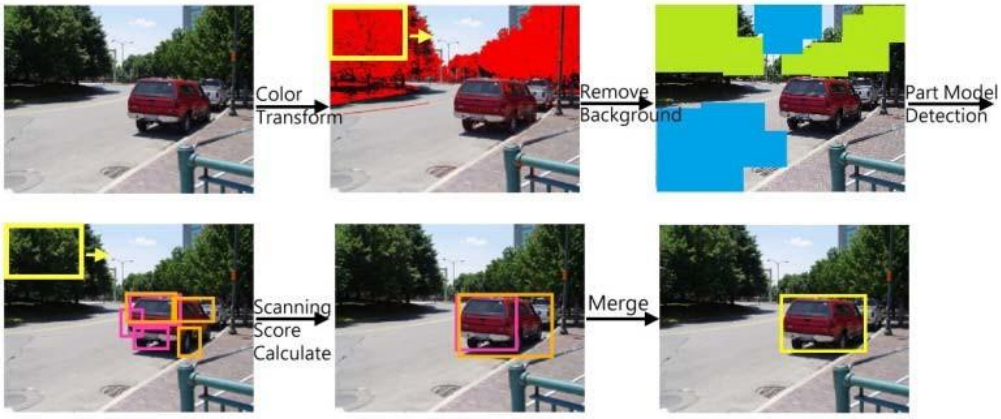


Fig. 9. Flowchart of the detection system

III. Experimental Results

Herein, CBCL street scenes challenge framework database [15] is used for training and testing our preprocessing methods, including color transform and smooth area detection. The labels of tree, sky, and road are used as training dataset. The training and testing part in the vehicle model use the vehicles' label in PASCAL VOC 2007 database [20]. In this database, it contains various backgrounds, different types, and sizes of vehicles. In the experiment, PASCAL VOC 2007 Challenge [20] is adopted as the evaluation standard.

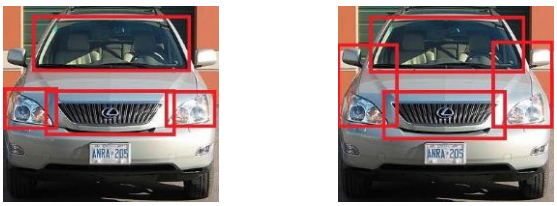
A. Part Model Changing

In this section, the front view of the vehicle is considered as an example. The original selection of parts is as in Fig. 10 (a). However, after detecting the labeled vehicle images, the F-measure of the headlight is very low. It can cause that the trained classifier not able to classify effectively. Therefore, the area is expanded as in Fig. 10(b). Then, these two cropping methods are tested using the collected front view images. From Table I, the result of changed parts raises by 11% in precision and 8% in recall rate. The F-measure increased by 9%.

B. Comparison of Previous Vehicle Detection System

In this section, the proposed system is compared with other multi-view non-deep-learning algorithms. The vehicle database in Pascal VOC 2007 [20] is used for the comparison purpose, and the evaluation standard adopts Average Precision.

In Table II, Oxford [21], MKL (Multiple Kernel Learning) [22], and Selective Search [23] all use the whole image to perform training and detection. Although these methods have improved the detection results, they are still unable to solve the problems of deformation and occlusion. Then, LHS [9], DPM [8], Fast Feature Pyramids [10], Contextualizing [11], and Probabilistic Inference [24] use part model to detect vehicles. It is observed that the proposed system achieved 58.4% in average precision.



(a) (b)

Fig. 10. (a) Original selection of parts of front view vehicle (b) Changed selection of parts of front view vehicle

TABLE I. Part changing results

Measurement Units	Original parts	Changed parts
Precision	0.83	0.94
Recall	0.67	0.75
F-measure	0.74	0.83

TABLE II. Comparison with other vehicle detection systems

Method	AP (%)
Oxford [21]	43.2
MKL[22]	50.6
LHS [9]	51.3
DPM [8]	51.6
Selective Search [23]	53.7
Fast Feature Pyramids [10]	50.1
Contextualizing [11]	56.0
The Proposed System	58.4

IV. Conclusion

A vehicle detection system which can detect multi-view vehicles has been proposed. In preprocessing, two methods are used to rapidly filter out the background objects. Furthermore, a novel active learning algorithm for part sample labeling is proposed to make part models robust. In the end, the location probability information is used to make the system more

adaptive to the database. At the same time, the detection accuracy is raised by changing the weaker parts and achieved comparable results.

REFERENCES

[1] A. Lipton, H. Fujiyoshi and R. Patil, “Moving Target classification and tracking from real-time video,” Proceedings Fourth IEEE Workshop on Applications of Computer Vision., New Jersey, USA, 1998

[2] M. Boninsegna and A. Bozzoli, "A tunable algorithm to update a reference image," Signal Processing: Image Communication, vol. 16, no. 4, pp. 353-365, 2000.

[3] D. Koller, J. Weber and J. Mallik, “Robust multiple car tracking with occlusion reasoning,” European Conference on Computer Vision, pp.189-196, Berlin, Heidelberg, 1994

[4] J. Wu, X. Zhang and J. Zhou, "Vehicle detection in static road images with PCA-and-Wavelet-Based classifier," IEEE Intelligent Transportation Systems. Proceedings, California, USA, 2001.

[5] Z. Sun, G. Bebis and R. Miller, “On-road vehicle detection using Gabor filters and support vector machines,” 14th International Conference on Digital Signal Processing Proceedings, Santorini, Greece, 2002

[6] L. Tsai, J. Hsieh and K. Fan, “Vehicle Detection Using Normalized Color and Edge Map,” IEEE Transactions on Image Processing, vol. 16, no. 3, pp.850-864, 2007

[7] Q. Yuan, A. Thangali, V. Ablavsky and S. Sclaroff, "Learning a Family of Detectors via Multiplicative Kernels," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 3, pp. 514-530, 2011.

[8] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, 2010.

[9] L. Zhu, Y. Chen, A. Yuille and W. Freeman, "Latent hierarchical structural learning for object detection," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA, 2010.

[10] P. Dollar, R. Appel, S. Belongie and P. Perona, "Fast Feature Pyramids for Object Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 8, pp. 1532-1545, 2014.

[11] Q. Chen, Z. Song, J. Dong, Z. Huang, Y. Hua and S. Yan, "Contextualizing Object Detection and Classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 1, pp. 13-27, 2015.

[12] H. Xu, Q. Huang and C. Kuo, “Car detection using deformable part models with composite features,” 2016 IEEE International Conference on Image Processing, Phoenix, Arizona, USA, 2016

[13] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng and F. Li, "Fine-Grained Car Detection for Visual Census Estimation," Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[14] L. Tsai, J. Hsieh and K. Fan, "Vehicle Detection Using Normalized Color and Edge Map," IEEE Transactions on Image Processing, vol. 16, no. 3, pp. 850-864, 2007.

[15] S. Bileschi, “CBCL streetscenes challenge framework," Cbcl.mit.edu, 2007. Available: http://cbcl.mit.edu/software-datasets/streetscenes/.

[16] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[17] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, California, USA, 2005.

[18] K. Sung and T. Poggio, “Example-based learning for view-based human face detection," IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no.1, pp. 39-51, 1998.

[19] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” MULTIMEDIA ’01 Proceedings of the ninth ACM international conference on Multimedia, pp. 107-118, Ottawa, Canada, 2001

[20] M. Everingham, L. V. Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 Result," 2007. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[21] O. Chum and A. Zisserman, "An Exemplar Model for Learning Object Classes," 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, Minnesota, USA, 2007.

[22] A. Vedaldi, V. Gulshan, M. Varma and A. Zisserman, "Multiple kernels for object detection," 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 2009.

[23] K. van de Sande, J. Uijlings, T. Gevers and A. Smeulders, "Segmentation as selective search for object recognition," 2011 International Conference on Computer Vision, Barcelona, Spain, 2011.

[24] C. Wang, Y. Fang and H. Zhao, “Probabilistic Inference for Occluded and Multiview On-road Vehicle Detection,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 1, pp. 215-229, 2016