FAST VEHICLE DETECTION WITH LATERAL CONVOLUTIONAL NEURAL NETWORK

Chen-Hang HE, Kin-Man LAM

Centre for Signal Processing, Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

ABSTRACT

In this paper, we propose a fast vehicle detector for traffic surveillance. We first explore using different feature layers from a deep residual network to perform vehicle detection. Experiment results show that the high-resolution features from earlier feature layers contain more structural information, which is good to achieve fine-grained localization but yields low recall rates. The low-resolution features in the deep layers contain semantically strong information, which is good to represent the objectness but too coarse to achieve accurate localization. Therefore, we decouple the localization and objectness prediction from a single layer. Instead, we employ a lateral network that takes the features from earlier layers as input and outputs the localization residual. Our proposed detector can achieve fast detection at a rate of 28 frames/s, and a mean average precision (mAP) of 67.25% in the DETRAC vehicle detection benchmark.

***Index Terms*—** Vehicle detection, Convolutional neural network

1. INTRODUCTION

Vehicle detection is one of the hot computer-vision tasks and has been widely used in many applications, such as Driver Assistance Systems (DAS) and traffic surveillance systems. An intelligent vehicle detection system can assist automatic driving, collect traffic statistics, and perform traffic scene analysis. However, vehicle detection still remains a challenging task, because vehicles often appear with severe occlusion and are varied in type, size and view-point. In addition, they are often perceived under bad scene conditions, such as rain, haze, snow, etc., and low-light conditions.

Early detectors based on template matching using low-level features, like [1], have achieved notable results. Considering objects that can be posed with non-rigid transformation, Felzenszwalb et al. [2] proposed a discriminatively trained part-based model (DPM), where a HOG detector was applied as a root filter to capture the coarse information from the main body of an object, and to capture the finer details of other deformable parts with a quadratic model describing the deformation cost. Recently, detectors based on convolutional neural network (CNN) have been largely explored. The data-oriented features learnt by CNN are more robust and efficient to handle different variations. Among the CNN-based approaches, there are two main streams of detection paradigm. In the region-based detection paradigm, typical works, including RCNN [3] and Fast-RCNN [4], perform the detection by firstly generating a group of region proposals based on image cues, then classifying these proposals using a CNN. Further extension has been made in Faster-RCNN [5], where a region proposal network (RPN) was proposed to generate cost-free region candidates by sharing the convolutional features in Fast-RCNN [4]. Another detection paradigm, including YOLO [6] and SSD [7], regards the detection as a regression task, in which CNN is applied to directly regress the location and objectness of the object in a one-stage manner. In YOLO, the image space is sliced into a grid and the prediction is performed with respect to each grid cell. In SSD, a set of default bounding boxes with different scales and aspect ratios are manually picked as the spatial prior to enhance the localization. The regression-based detectors are relatively fast and efficient. However, compared to the region-based detectors, they are lagging behind, in terms of detection accuracy. This is because 1) the regression-based detectors only sample the possible object locations from the image space, which introduce class imbalance between the object and the background samples, 2) using a single layer to predict both the localization and objectness results in the detector is more vulnerable to scale variations, and 3) the object is predicted by the grid cell, in which the center of the object lies, which makes it hard to detect objects with severe occlusion. Therefore, we propose a novel regression-based detector, namely Lateral-CNN, with the following improvements. First, we propose a multi-cell prediction scheme, in which the grid cell that sees a part of an object can be used to predict the object. Second, we utilize the early layers in the network to predict the localization residual with a lateral network. Third, we apply the imbalance-free loss function proposed in [8] to train the network. Our detector can achieve comparable performance to the state-of-art region-based detector, and at the same time, reduce the detection time. The next section will describe and illustrate our proposed network, in more detail.

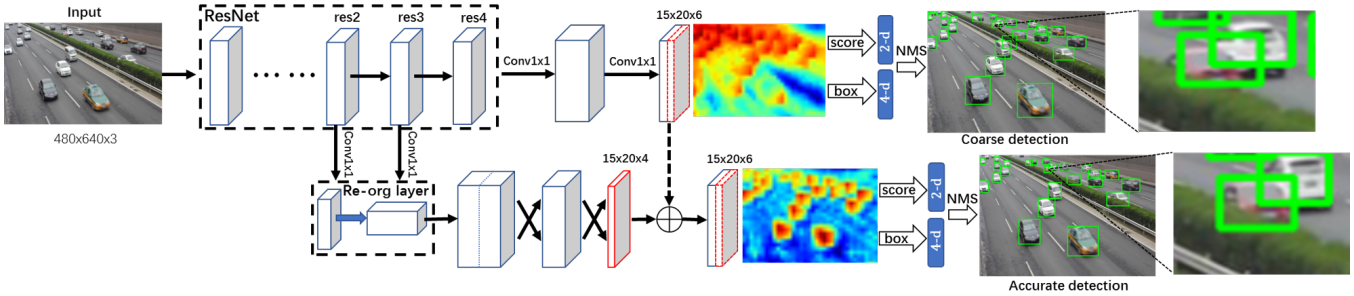

Fig. 1. The upper network is the coarse detector formed by a fully convolutional network, while the lower is the lateral network, which generates the fine-grained localization residuals. The average localization maps for each network are shown.

2. PROPOSED VEHICLE DETECTOR

The proposed LateralCNN is shown in Figure 1, where a deep residual network [9] is used to extract rich feature representations from a given image. Two convolutional layers are employed to the last feature layer to generate 6 feature maps (4 localization maps and 2 objectness maps), which are used to coarsely regress the bounding-box coordinates and the corresponding objectness scores. The features from earlier layers are passed through a lateral network with 1×1 reduction. The lateral network can produce highly resolved feature residuals on the localization maps (*red*). The final output bounding box is determined by applying non-maximum suppression.

2.1. Objectness vs. localization

Features in different scales have different semantical meanings in object detection. To explore which feature layer is efficient for detecting vehicles at a particular size, we use the different residual blocks from a pre-trained 101-layer ResNet to perform detection. We refer to these blocks as res2, res3, and res4 features, and conduct exploratory experiments on the DETRAC [10] dataset. We evaluate the detection performance in terms of the average precision (AP) and the average recall (AR), with IoU threshold of 0.5. From Table 1, we observe that the AP increases from res4 to res3, but drops from res3 to res2, and res2 yields an extremely low recall rate. Based on this, we infer that the high-resolution features from early layers, which contain low-level structural information, is less sensitive to semantics. In contrast, the low-resolution deep layers provide more high level abstraction of the object, which is semantically strong to represent the objectness but too coarse to give accurate localization, especially for small objects. We argue that there should exist a trade-off between a better localization and a higher recall, so we decouple the objectness and localization prediction from a single layer. As a result, we select the semantically strong res4 feature and employed two convolutional layers to regress the object. This fully convolutional structure maintains relatively higher recall, which allows more room for our model to achieve better accuracy. In the later Section 2.3, we will show how to utilize the early layers to enhance the localization accuracy.

Layer	Resolution	AP	AR
res4	15x20	63.6	86.3
res3	30x40	67.2	85.1
res2	60x80	58.3	78.7

Table 1. Average precision (AP) and average recall (AR) are evaluated when using different feature layers to perform detection. The resolution of the input image is 480×640.

2.2. Multi-cell prediction

Similar to YOLO, we sample the image space into a grid, and for each grid cell, the height, width, centre offsets relative to the cell, and objectness scores of the bounding box are predicted. However, instead of focusing on only one grid cell where the center of the object lies, as it does in YOLO, we encode the object's bounding-box information into multiple grid cells that overlap with this object. If multiple objects are overlapped with one cell, the object with the highest overlapping area is predicted by that cell. Using multiple cells to perform prediction can largely increase the robustness to object occlusions. Figure 2 illustrates a case where the two objects are highly overlapped, so that their bounding-box centers lie in the same cell. In this case, using the single-cell prediction can detect one object only, while the multi-cell prediction can detect the occluded object by another cell where the object partially lies.

2.3. Lateral residual network

Semantically strong features from the deep layers can achieve relatively higher recall, but they are too coarse to detect an object at a small scale. Compared to the deep layers, the high-resolution features from earlier layers can provide much more detailed information about tiny objects and fine-grained representations. Thus, we incorporate these features into our model by passing it through a lateral network. In the

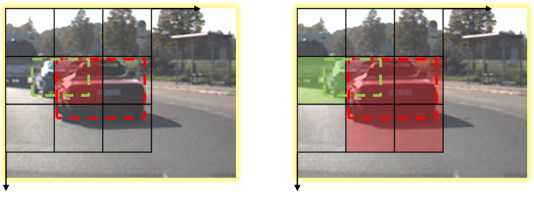

Fig. 2. Single-cell (*left*) and Multi-cell (*right*) predictions. The colored cells are responsible for predicting the same colored bounding-box during the inference.

lateral network, 1×1 reduction layer is first applied, then a re-organizing layer, which can transform the tensor with the shape of $(height, width, depth)$ into the tensor with the shape of $(height/scale, width/scale, depth \times scale \times scale)$ is employed. The transformed high-resolution features have the equal scale of the low-resolution features, Thus we can concatenate these features and apply two fully connected layers to predict the localization residual efficiently. With the help of the lateral network, the model can finely resolve the localization map efficiently.

2.4. Network training

The localization predictions in each cell contain the information of the bounding box coordinates, in terms of the center offset relative to the grid cell, its height and width. For simplicity, we use p_i^{loc} to denote the 4-d localization prediction in each grid cell, where $i = 1 \dots N$ is the index of the grid cell. The objectness prediction in each cell is represented by a 2-d foreground-background probability distribution, which is simply denoted by p_i^{obj}. Following the multi-cell encoding scheme, we can calculate the localization ground truth, which is denoted by t_i^{loc}. We use c_i to indicate the class encoded by each grid cell, which is set to 1 if the cell is assigned to detect the object and set to 0 if the background. For the lateral network, we denote its output by δp_i^{loc}, which represents the localization residual. Thus, the network can be trained in two stages. The first stage is to train the coarse detector, with the following loss function:

$$\mathcal{L}_1 = L_{loc}(p, t) + \alpha L_{obj}(p, c). \tag{1}$$

The second stage is to train the lateral network to predict the localization residual, with the following loss function:

$$\mathcal{L}_2 = L_{loc}(p + \delta p, t). \tag{2}$$

The network can be jointly trained with the total loss as follows

$$\mathcal{L} = \mathcal{L}_1 + \beta \mathcal{L}_2. \tag{3}$$

L_{loc} is an l_1 loss, which is defined as:

$$\mathcal{L}_{loc}(p, t) = \frac{1}{N} \sum_{i=1}^{N} c_i \left\| p_i^{loc} - t_i^{loc} \right\|. \tag{4}$$

To solve the foreground-background class-imbalance issue in our detector, a more dedicated focal loss [8] is used with Sigmoid normalization σ. Following [8], L_{obj} can be defined as:

$$\mathcal{L}_{obj}(p, c) = -\frac{1}{N} \sum_{i=1}^{N} \omega (1 - p_i^*)^\gamma \log(p_i^*), \tag{5}$$

where

$$p_i^* = \begin{cases} \sigma(p_i^{obj}) & \text{if } c_i = 1, \\ 1 - \sigma(p_i^{obj}) & \text{otherwise.} \end{cases} \tag{6}$$

α is set to 0.1, which is for balancing the loss contribution between the localization and objectness predictions. β is set to 0.05, which is determined by cross validation. ω and γ are the hyper-parameters in the focal loss equation. We empirically set them to 0.5 and 2, respectively, which are the optimized values mentioned in [8].

3. EXPERIMENTS

We evaluated the proposed LateralCNN on DETRAC [10], which is a traffic vehicle detection benchmark with 140K captured frames from traffic surveillance. The numbers of training and testing samples are 84K and 56K, respectively, and each sample has a resolution of 960×540. The test set is categorized into different challenge levels, which consider different object sizes and degrees of occlusion, and different scene conditions, such as sunny, cloudy, rainy and night.

To train our detector, we resized the training images into the resolution of 640×480, and added jitter to the images. We initialized the feature extractor with a pre-trained 101-layer residual network [9], and initialized the other parts of the network with a zero-mean uniform distribution. Batch normalization is applied to all the convolutional layers. The initial learning rate is 0.001 for the first 100K iterations, and then dropped by 50% after every 30K iterations. The network was totally trained for 200K iterations with a mini-batch size of 16, using stochastic gradient descent.

3.1. Ablation experiments

We split the samples into two subsets, 75% for training and 25% for validation. Then, we performed the ablation experiments to evaluate the effectiveness of different components. The first three entries in Table 3 show the results of the coarse detector, with and without batch normalization (BN), and with either cross-entropy loss (CE) or focal loss (FL). The last two entries show the results of using the lateral network to perform fine-grained detection with different early layers. We found that using batch normalization and focal loss can improve the detection accuracy by $1\% \sim 2\%$, and with the lateral network, the detector can achieve nearly 9% improvement. Although the lateral network introduces the additional runtime, the detection can still achieve 28 frames/s with a moderated GPU.

Method	Overall	Easy	Medium	Hard	Sunny	Cloudy	Rainy	Night	FPS	GPU
RCNN[3]	48.95	59.31	54.06	39.47	59.73	39.32	39.06	67.52	0.10	Tesla K40
Faster RCNN[5]	58.45	82.75	63.05	44.25	66.29	69.85	45.16	62.34	11.11	TitanX
CompACT[11]	53.23	64.84	58.70	43.16	63.23	46.37	44.21	71.16	0.22	Tesla K40
EB[12]	**67.96**	**89.65**	73.12	**53.64**	**72.42**	73.93	53.40	**83.73**	10.00	TitanX
LateralCNN	67.25	89.56	**73.59**	51.61	69.11	**74.36**	**55.77**	78.66	**28.46**	1080Ti

Table 2. The detection performances on DETRAC, under different challenges and weather conditions.

Early layer(s)	Use BN?	Loss	mAP	FPS
-	no	CE	68.5	35
-	yes	CE	70.3	35
-	yes	FL	72.2	35
res3	yes	FL	80.5	29
res3+res2	yes	FL	81.6	28

Table 3. Ablation experiments using different configurations, with the mean average precisions (mAP) and frames-per-second (FPS) reported.

3.2. Comparison to other detectors

We compare LateralCNN with other state-of-the-art methods in terms of average precision, and evaluate the detection time, in terms of frames-per-second (FPS). As shown in Table 2, our detector thoroughly outperforms most of the state-of-the-art approaches, like RCNN [3], Faster RCNN [5], and CompACT [11], and achieves similar accuracy to EB[12], but with a lesser inference time. The reason behind this is that our detector does not rely on the proposal generation. We directly regress the object bounding box from the output. Figure 3 shows the precision-recall curves of our method and the other methods. Note that our detector can achieve better localization accuracy, but a lower recall rate compared to EB. This is because our detector has a regression-based architecture. Figure 4 shows the qualitative results of our method on the DETRAC test set. Most of the vehicles are at different sizes and view points, and under severe occlusion, but our method can detect them successfully.

4. CONCLUSION

In this paper, we present a fast vehicle detection approach, which detects objects from coarse to fine. Our coarse detector can achieve relatively high recall, by using semantically strong features. A lateral network is employed to fuse the high-resolution features from earlier layers to achieve fine-grained localization. In our experiments, we show that, with this lateral network, our detector can achieve better detection performance, with an average precision of 67.25% on the DETRAC benchmark. Furthermore, the detector is suitable for real-time applications, which achieves a detection rate of 28 frames/s.

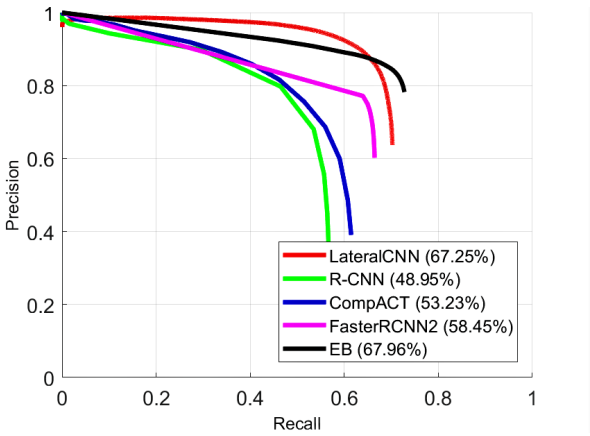

Fig. 3. The precision-recall curves of our proposed method and different state-of-the-art methods, tested on the DETRAC test set.

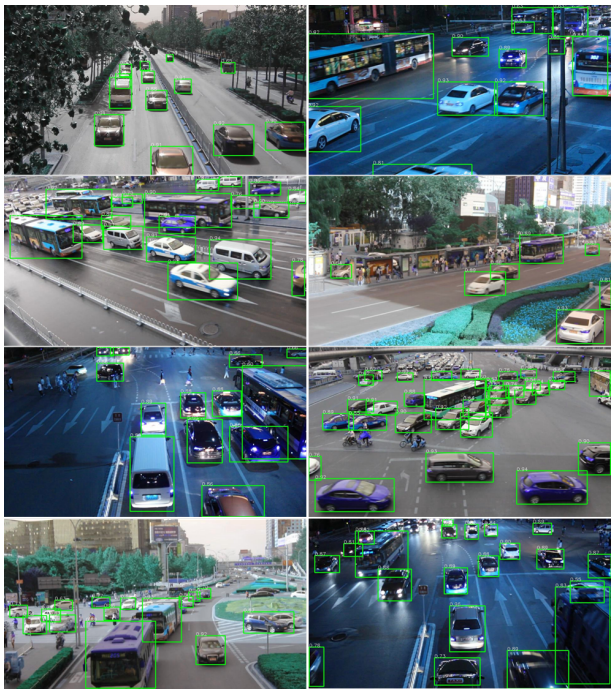

Fig. 4. Qualitative results of the proposed method on the DETRAC test set.

5. REFERENCES

[1] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 886–893.

[2] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[4] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[8] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," *arXiv*, vol. abs/1708.02002, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[10] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu, "DETRAC: A new benchmark and protocol for multi-object tracking," *arXiv*, vol. abs/1511.04136, 2015.

[11] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 236–243.

[12] Li Wang, Yao Lu, Hong Wang, Yingbin Zheng, Hao Ye, and Xiangyang Xue, "Evolving boxes for fast vehicle detection," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2017, pp. 1135–1140.