Vehicle Detection using Simplified Fast R-CNN

Shih-Chung Hsu
*Dept. of Electrical Eng.,
National Tsing-Hua University,
Hsin-Chu, Taiwan
E-mail: chvjohnff@gmail.com

Chung-Lin Huang†
Dept. of M-Commerce and Multimedia App.
Asia University,
Tai-Chung, Taiwan
E-mail: clhuang@asia.edu.tw

Cheng-Hung Chuang
Dept. of Computer Sci. and Inf. Eng.
Asia University,
Tai-Chung, Taiwan.
chchuang@asia.edu.tw

***Abstract*—This paper proposes a simplified fast region-based convolutional neural network (R-CNN) for vehicle detection. Fast R-CNN is a well-known method for object recognition using deep convolution networks. The original fast R-CNN consists of two separated parts: regional proposal and object recognition. The object recognition part in Fast R-CNN is redundant for our system which can be removed to speed up the training process. In the experiments, we test our method by using SHRP 2 NDS database [10] offered by Virginia Tech Transportation Institute (VTTI) to show the detection accuracy.**

***Keywords*—Vehicle detection; Region-based convolutional neural network (R-CNN); Object recognition.**

I. INTRODUCTION

Vehicle detection methods [1, 2] have been proposed to identify the object as a vehicle. It can be used for multi-view vehicle tracking and verification [3]. Recently, convolution neural networks (CNN) has been proposed that shows considerably high image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). CNN can also be applied for object detection, such as R-CNN [4], which shows a great improvement on object detection accuracy compared with the conventional feature-based detectors.

In [4], they combines the region proposals with CNNs, which is called R-CNN that regions with CNN features. In R-CNN, the possible objects are extracted by selective search, which proposes 2000 object regions. Then, the extracted image content is aligned to the same size (227x227). Finally, the CNN with SVM classifiers assigns what type of objects the image content of region belongs to. However, the construction of R-CNN is slow because it performs a ConvNet forward pass for each object proposal, without sharing computation. Spatial pyramid pooling networks (SPPnets) are proposed to speed up R-CNN. However, it has some drawbacks; (1) training is a multistage pipeline that requires disk storage which is very time consuming, (2) fixed convolution layers limits the accuracy of very deep networks.

Fast R-CNN [5] processes the whole image with several convolutional and max pooling layers to produce feature map. For each object proposal, a region of interest (ROI) pooling layer extract a fixed length feature vector from the feature map. Fast R-CNN appends a ROI max pooling layer and two full connection layers with after CNN layers. One of the full connection layers is designed for object category recognition and the other can fine-tune ROI position as a regression method.

Furthermore, faster R-CNN [6] improves the system by proposing Regional Proposal Networks (RPNs) that share convolutional layers with object detection networks. It merges a ROI proposal layer which gave k-possible region proposals and decides whether each region proposal contains an object.

Recent evidence [7] reveals that the detection network depth is of crucial importance, and the leading results on the challenging ImageNet dataset all exploit "very deep" network models, with a depth of sixteen to thirty. However, when the network depth increases, the accuracy may not improve furthermore. An obstacle to accuracy enhance is the notorious problem of vanishing/exploding gradients [8], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [8] and intermediate normalization layers [9], which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation.

However, the degradation problem occurs for deeper networks. Once the network depth increases, the accuracy gets saturated and then degrades rapidly. Such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error. To reduce the gradient degrading, we reduce the complexity of the deep learning network for identify the specific vehicle objects in different views.

Our detection system consists of several convolution and max pooling layers, followed by region-of-interest (ROI) pooling layers, and finally connected to a sequence of fully-connected (*fc*) layers which branch into two *fc* output layers: softmax and bounding-box regressor. The former produces the probability of the existence of a vehicle, whereas, the latter output four real values numbers for the bounding box enclosing the vehicle.

II. VEHICLE DETECTION

In a video surveillance system, cameras are fixed at the specific positions to monitor the target objects. We propose a method to detect the vehicles in the video. Here, we simplify the complex deep learning construction of fast R-CNN with fast learning process. Different from image classification, vehicle detection requires to localize the

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

vehicle in an image. We treat the localization as a regression problem which can be solved by a slide-window detector. We propose the sliding-window approach by using CNN with five convolution layers and strides. CNN can extract the fixed-length feature vectors which are then classified into vehicle or non-vehicle.

The other challenge faced is that the amount of currently available labeled data is insufficient for training a large CNN. We adopt the conventional method by using un-supervised pre-training followed by supervised fine-tuning. We do the supervised pre-training on a large auxiliary dataset (ILSVRC) followed by the domain-specific fine-tuning on a small dataset (SHRP 2 NDS database [10]), is an effective approach for CNN learning when data is not sufficient.

The CNN processes the entire image with several convolutional and max pooling to produce a conv feature map. For each object proposal a region of interest (RoI) pooling layer extract a fixed-length feature vector from the feature map. Each feature vector is fed into a sequence of fully connected layer that finally branch into two sibling output layers, one is the softmax layer that differentiates the vehicle object from the background, the other layer outputs four real value numbers for the vehicle object which encodes the refined bounding-box positions of the object.

The RoI max pooling layer converts the feature inside a RoI into a small feature map with a fixed spatial extent of H×W(or 7×7) . Each RoI is defined by four tuple (*r, c, h, w*) of which (*r, c*) represents its top-left corner and (*h, w*) denotes it height and width. The RoI max pooling divides the *h*×*w* window into *H*×*W* grid of sub-windows of size (*h*/*H*)×(*w*/*W*). Then the max-pooling values from the sub-windows are put into the corresponding output grid cell.

To develop our system, we split fast R-CNN into two separate parts: region proposal and object recognition. The region proposal will determine whether the objects appear in an image. If we specify the region proposal as vehicle proposal, we can remove the object recognition. The architecture construction of the fast R-CNN for our vehicle detector is shown as bellow.

Figure 1. Deep learning vehicle detector.

Vehicle proposal is constructed by position regression and object position classification. Each offset corner coordinate, *x* or *y* of the left top or right bottom corner, is regressed from a convolution layer. Each convolution layer contains 15x15 possible object targets. The position regression proposes 9 layers of 15×15 targets. There are 36 layers for position regression. In position classification, each possible object target is assigned to the possibility of vehicle object. Finally, vehicle proposal applies non maximum suppression on removing the redundant region proposals and output the final result.

We modify the fast *R-CNN* network which has two output layers. The 1st output layer generates a discrete probability p_v (per RoI), p_v indicating the likelihood of the existence of a vehicle. The p_v is computed by a softmax of a fully connected layer. The second output layer generate the bounding-box regression offsets, $\boldsymbol{t}_v = (x_r, y_t, x_l, y_b)$ for the vehicle. $\boldsymbol{t}_v$ specifies a scale-invariant translation and height/width shift relative to the object proposal. Each training RoI is labeled with vehicle or non-vehicle with a ground-truth bounding box regression offset $\boldsymbol{t}_v$. The non-maximum regression algorithm is shown in the following.

Non-maximum suppression for Region proposals:

Input: Corner offsets (x_r, y_t, x_l, y_b) from 4 sets of 15x15 convolution layers, the possibility scores **s** from a 15x15 convolution layer, region overlapped threshold **t**.

```
Begin
    Sort s with the information of box(·) (x_r, y_t, x_l, y_b) by
    descending order.
    For i = start index of s
      For j = start index+1 of s
        If overlapped ratio of box(i) or box(j)>t
          Erase s(j) and box(j)
        End if
        j++
      Next
    Next
End
```

Output: The region information of **box(.)**

Figure 2. Algorithm of non-maximum suppression

III. THE EXPERIMENT

We modify the open source codes of CaffeNet from R-CNN[4] for our experiment. In the experiments, we use the pre-trained ImageNet model that is available on-line. Our vehicle detector is developed by training a binary image region classifier. An image region tightly enclosing a vehicle is a positive example, whereas the image region of non-vehicle is a negative example which may be another object or background.

We test our detector by using the videos captured by the cameras installed on two different crossroad in Hsin-Chu City, Taiwan. Our method can detect and localize the vehicles in the video as shown in Figure 3. The two crossroad are separated by 1.5-km. The vehicles can be extracted from different illuminances, slight occlusion.

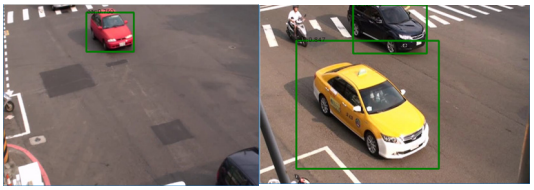
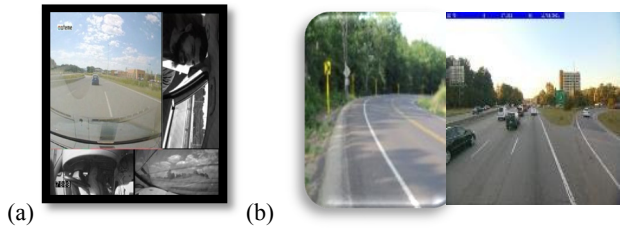


Figure 3. Vehicle detected by our deep learning detector.

We also test our method using the videos obtained from SHRP 2 NDS database [10] offered by Virginia Tech Transportation Institute (VTTI). We used two video

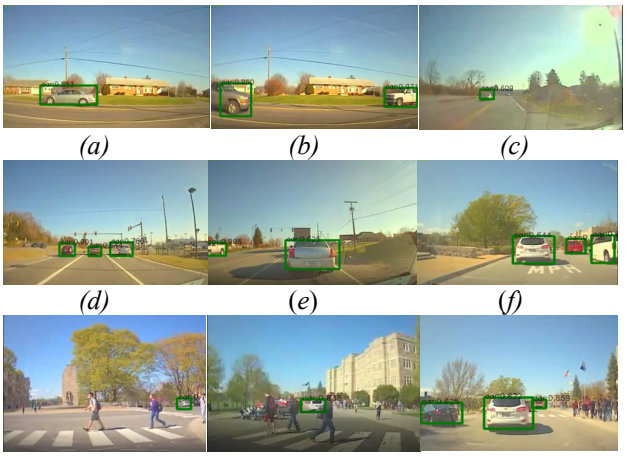
sequences captured from the front cameras installed in two different vehicles, one of them is shown in Figure 4.



(a) (b)

Figrue 4. (a) The camera installed inside the vehicle, (b) the captured videos.

Our method can detect the side-view of the vehicle as shown in Figure 5(*a*), the partial view of the vehicles as shown in Figure 5(*b*), the front view of the vehicle as shown in Figure 5(c), and the rear-view of the vehicles as shown in Figures 5(d) and Figure 5(*e*). It can identify multiple vehicles, one show-ups in its rear-view (or side-view) and the other partially appears as shown in Figure 5(b) and Figure 5(f). Our method can also localize the vehicles in the scene with the presence of other foreground objects, such as the pedestrians, as shown in Figure 5(*g*), Figure 5(*h*) and Figure 5(*i*).



(a) *(b)* *(c)*

(d) *(e)* *(f)*

(g) *(h)* *(i)*

Figure 5. Vehicle detected in the videos from SHRP 2 NDS database.

The accuracy of our detector applied on the two videos are listed in Tables 1 and 2.

Table 1. VTTI Vedio 4 Front View: 270 image containing 250 vehicles

Ground Truth	True positive	False Negative	False Positive
250	214	13	23

Table 1 shows the detection accuracy of our detector with 90.3% precision and 94.3% recall.

Table 2. VTTI Vedio 5 Front View: 250 image containing 235 vehicles

Ground Truth	True positive	False Negative	False Positive
235	207	3	25

Table 2 shows the detection accuracy of our detector with 89.2% precision and 98.5% recall. In our experiment, our detector can extract the vehicle properly and demonstrate satisfactory results in different datasets. The experimental results of using VTTI database show that our detector can detect and localize the vehicles in different front view, side view and rear view. The experiment also show that it can identify the vehicles with the presence of other foreground objects such as human objects.

IV. CONCLUSIONS

The paper presents a simple and fast method by modifying fast R-CNN for vehicle detection and localization. The CNN and max pooling provide sparse object representation of the ROI for the following fully-connect layer to classify the ROI. We apply the supervised pre-training and fine-tuning the network to solve the problem of highly non-sufficient labeled training data. In the experiments, we show that our method can detect and localized the vehicles in various views effectively.

REFERENCE

[1] J. Arróspide and L. Salgado, "Log-Gabor Filters for Image-Based Vehicle Verification," *IEEE Transactions on Image Processing,* pp. 2286-2295, 2013.

[2] J.-M. Guo , H. Prasetyo and K. Wong, "Vehicle Verification Using Gabor Filter Magnitude with Gamma Distribution Modeling," *IEEE Signal Processing Letters,* vol. 21, pp. 600-604, 2014.

[3] S. C. Hsu, I. C. Chang and C. L. Huang, "Object verification in two views using Sparse representation," *IEEE Int. Conf. on Pattern Recognition*, Cancun, 2016.

[4] J. Donahue, R. Girshick, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Columbus, 2014.

[5] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision*, Santiago, 2015.

[6] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Las Vegas, 2016.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Columbus, 2014.

[8] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Int. Conf. on Artificial Intelligence and Statistics*, Sardinia, 2010.

[9] S. Ioffe and C. Szegedy, "Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Int. Conf. on Machine Learning*, Lille, 2015.

[10] "InSight SHRP2 NDS," Virginia Tech Transportation Institute, 3 4 2013. [Online]. Available: https://insight.shrp2nds.us/home/index.