© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LICENSE PLATE DETECTION AND REFINEMENT BASED ON DEEP CONVOLUTIONAL NEURAL NETWORK

HOANH NGUYEN

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

E-mail: nguyenhoanh@iuh.edu.vn

ABSTRACT

In this paper, a new method for detecting license plate in natural scene images based on deep convolutional neural networks. Faster Region-based convolutional neural network framework (R-CNN) and MobileNets architecture are used to design our framework for license plate detection. Edges information of license plates is then used to refine the detected bounding boxes and further improve the performance on small license plates, which are not easy to regress precisely. We train and evaluate our framework on public datasets. The results show that our method can apply to different types of license plates with better performance than current state-of-the-art methods in terms of detection accuracy and run-time efficiency.

Keywords: *License Plate Detection, Convolutional Neural Network, Intelligent Transportation Systems, Object Detection, Deep Learning*

1. INTRODUCTION

Vehicle license plate recognition plays an important role in intelligent transport systems, traffic management, and so on. Most of the existing methods only work well under limited conditions such as simple background and fixed illumination. In complicated conditions such as complicated backgrounds, illumination, distortion, occlusion, blurring and so on, it is still a challenging task to read license plates accurately. The difficulty lies in the diversity of character patterns, such as different sizes/fonts/colors across countries, character distortion and low-quality images caused by capturing condition. The highly complicated background makes the problem even intricate, especially the general text in shop boards, windows or bricks, which often lead to false alarms in detection. An automatic license plate recognition system includes license plate detection and license plate recognition. License plate detection takes images captured from camera to extract exactly license plate region, while license plate recognition segments and recognizes each character on the detected license plate. License plate detection has dramatic effect on the accuracy of the whole system. Thus, many approaches have been proposed to detect license plate. Traditional methods [5, 6, 8, 9, 10, 11] usually utilize morphological features of license plate such as color, dimension, character and so on. These methods perform well under certain limited conditions such as the same type of license plate, fixed illumination, fixed viewpoint and simple background. When these conditions change, these methods show poor performance. Recently, with fast development of deep learning, a certain number of methods for license plate detection based on deep learning [1, 2, 3, 4, 7] have been proposed. These methods show better performance than traditional methods. Deep CNN-based methods firstly create license plate candidates. Then, a deep CNN-based plate/non-plate classifier is used to reject non-plate candidates. Although these methods perform well in complex conditions, they cannot run in real-time. In this paper, we propose a new method for detecting license plate in outdoor scene images. We adopt Faster R-CNN framework along with MobileNets architecture to detect license plate in image. Recent state-of-the-art deep CNN-based object detector frameworks such as Faster, SSD [16] and Region-based Fully Convolutional Networks (R-FCN) [18] achieve significantly improved performance compared to traditional methods, so they can handle license detection very well. Compared to other frameworks, Faster R-CNN show the best accuracy with small objects [14]. Thus, we use this framework in this study based on the fact that license plates are usually small in outdoor scene images. Furthermore, small license plates cannot be localized precisely enough, making the subsequent

Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

recognition task more difficult. Thus, we perform a refinement step to refine the detected license plates and further improve the performance on small license plates. The main contributions of this paper are summarized as follows:

- We propose a new approach to detect different types of vehicle license plates in diverse outdoor scenes. Moreover, the results show that our approach achieves better accuracy than other methods, especially with small license plates in complex background.
- We combine and modify Faster R-CNN framework and MobileNets architecture to make the detection process more efficient in terms of detection accuracy and run-time efficiency.
- We propose a refinement stage for improving the detection quality of the detected bounding boxes and further improving the performance on small license plates.

This paper is organized as follows: an overview of previous methods is presented in Section 2. Section 3 describes the detail of our proposed method. Section 4 demonstrates experimental results. Finally, the conclusion is made in Section 5.

2. RELATED WORK

In this section, we brief introduce previous work on license plate detection, including traditional methods and recently proposed methods based on deep CNN.

2.1 Traditional License Plate Detection Method

Traditional methods usually utilize morphological information such as color, edge, texture, character and so on. to detect license plate [12], [20]. Recently, some methods combining these morphological features for detecting license plate have been proposed. In [5], a new Riesz fractional model was first used to improve low quality license plate images affected by multiple factors. Since the proposed model involves differential operation by performing convolution operation over each input image with the Riesz fractional derivative window, it enhances edges irrespective of distortions created by multiple factors. After improving the quality of the input license plate images, a modified MSER algorithm was used for character candidate detection by adding stroke width information. Finally, a classifier was adopted to eliminate non-character. In [6], Yuan *et al.* proposed a novel image downscaling method for license plate detection which can substantially reduce the size of the image without sacrificing detection performance. For creating license plate candidates, the paper proposed a novel line density filter for extracting license plate candidates. After that, these candidates will be refined by connected-component labelling algorithm. Finally, the real license plate is obtained from among the detected candidate regions by cascaded license plate classifier, which is based on linear SVMs and color saliency features. Gou *et al.* in [8] used morphological operations, various filters, different contours and validations for detecting coarse license plates. Then, character-specific ERs were selected as character regions through a Real AdaBoost classifier with decision trees. Accurate character segmentation and license plate location were achieved based on the geometrical attributes of characters in standard license plates. Finally, characters were extracted and recognized by using an off-line trained classifier based on HDRBM. In [9], Li *et al.* extracted MSER as character candidates, the paper then introduced a CRF model to describe the contextual relationship among the candidates. Finally, license plates were located through CRF inference. In [10], Ashtari *et al.* proposed an Iranian vehicle license plate recognition system based on a modified template-matching technique by the analysis of target color pixels to detect the location of a license plate, along with a hybrid classifier that recognizes license plate characters. Zhou *et al.* in [13] proposed a scheme to automatically locate license plates with principal visual word, discovery and local feature matching algorithm. The paper brings in the idea of using the bag-of-words model popularly applied in partial-duplicate image search. Traditional methods can be processed in real-time with many systems, and these methods achieve good performance in limited conditions such as the same license plate type, simple background and environment and so on. However, the performance of these methods will significantly reduce in complex conditions.

2.2 Deep CNN-based License Plate Detection Method

Recently, a certain number of methods for license plate detection based on deep CNN have been proposed. In [1], Li *et al.* used a 4-layer with 37-class CNN classifier in a sliding-window fashion across the entire image at first stage to detect the presence of text and then create a text

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

saliency map. At second stage, a 4-layer plate/non-plate classifier was adopted to reject the false negative. In [2], Kim *et al.* proposed a method for vehicle detection instead of detecting license plate at first. Faster R-CNN algorithm was adopted for detecting vehicle regions. The hierarchical sampling method was then used for generating license plate candidates. Finally, non-plate candidates were filtered out by a plate/non-plate classifier. Rafique *et al.* in [3] used Faster R-CNN framework with VGG16/ZF architecture to directly detect license plates. There are no standard license plate datasets available for training Faster R-CNN, so they extended Pascal VOC2007 data by combining a publicly available license plate dataset. Bulan *et al.* in [4] proposed a two-stage approach, where a set of candidate regions are firstly extracted by a weak sparse network of winnows classifier trained with successive mean quantization transform features and then scrutinized by a strong readable/unreadable CNN classifier in the second stage. Images which fail a primary confidence test for plate localization were further classified to identify the reasons for failure such as license plate not present, license plate too bright, license plate too dark and no vehicle found. Xie *et al.* in [7] introduced a new MD-YOLO model for multi-directional car license plate detection. The proposed model could elegantly solve the problem of multi-directional car license plate detection and could also be deployed easily in real-time circumstances because of its reduced computational complexity compared with previous CNN based methods. Because the image area proportion of the car license plate is usually very small, and small image portions such as the car license plate may deem to introduce some redundant information, the paper used a prepositive CNN which plays an important role in removing redundant information.

3. APPROACH

In this section, we present the details of our proposed method. As illustrated in Figure 1, we combine and modify Faster R-CNN framework with MobileNets architecture for detecting license plate in image. MobileNets architecture is used as base network and the classifier in Faster R-CNN framework. A refinement stage is designed and performed to further improve the precision of localization and the accuracy of our method.

3.1 License Plate Detection

License plate detection involves not only recognizing license plate in an image but localizing the position of each license plate by drawing the appropriate bounding box around it. Because license plates are normally in a rectangular/square shape with a specific aspect ratio and colour, traditional methods for license plate detection usually utilize edges/colour/texture information for extracting license plates in an image. Traditional methods are usually fast in computational, but they tend to fail in many difficult environments.

In this paper, Faster R-CNN [15] framework with MobileNets [17] architecture as a base network is adopted to detect license plate. Faster R-CNN introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. RPN creates a set of anchor boxes from the convolution features created by the base network. For each anchor box, RPN outputs two predictions including objectness score and bounding box regression. Faster R-CNN is chosen for detecting license plate in this study because this framework shown the best performance for detecting small objects compared to other state-of-the-art deep CNN-based object detection frameworks such as R-FCN and SSD [16], while license plates usually have small size in an image. There are some deep CNN architectures that showed state-of-the-art performance on many competitions such as VGG-16, Resnet-101, Inception v2 and so on. Google recently released an efficient model called MobileNets for mobile and embedded vision applications. MobileNets splits the convolution into a 3x3 depthwise convolution and a 1x1 pointwise convolution, effectively reducing both computational cost and number of parameters. It introduces two parameters that we can tune to fit the resource/accuracy trade-off, including width multiplier and resolution multiplier. The width multiplier allows us to thin the network, while the resolution multiplier changes the input dimensions of the image, thus reducing the internal representation at every layer. In this study, MobileNets is adopted to build the base convolutional layers in Faster R-CNN. The overall framework of our method for detecting license plates is illustrated in Figure 2. The base network takes input image to generate convolutional feature maps. The RPN takes all the reference boxes and outputs a set of good proposals for license plate. Region of interest pooling layer then extracts fixed-sized feature maps for each proposal. The MobileNets-based classifier finally classifies proposals into license plate and background class and adjusts the bounding box for each of detected license plate. In the following subsections, we give

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a detailed description about each component in our framework.

1) The Base Network: MobileNets architecture is used as a base network in this study. Since we use only the convolution layers in MobileNets architecture, the size of the input image does not have to be fixed. Supposing the size of the input image is 224 x 224 x 3, the architecture of the base network is defined as Table 1.

where:
'Conv' represents as a standard convolution
'Conv dw' represents as a depthwise separable convolution
's1' represents that the convolution stride is 1 x 1
's2' represents that the convolution stride is 2 x 2

Depthwise separable convolution are made up of two layers: depthwise convolutions and pointwise convolutions. Depthwise convolutions is used to apply a single filter per each input channel, while pointwise convolution, a simple 1x1 convolution, is used to create a linear combination of the output of the depthwise layer. MobileNets use both batchnorm and ReLU nonlinearities for both layers. The reduction of computational cost is in proportion to the number of output feature map channel and the square of kernel size. More details about MobileNets architecture can be found in [17].

2) The Region Proposal Network (RPN): First, the RPN takes the convolution feature map and generates a set of anchor boxes. An anchor is centered at the sliding window and is associated with a scale and aspect ratio. Because license plates are usually in rectangular or square shape, we use three scales and three aspect ratios for each anchor, yielding 9 anchors at each sliding position. For a convolutional feature map of a size 14 x 14, there are 1764 anchors in total as shown in Figure 3.

Next, the RPN takes all the anchor boxes and outputs two different outputs for each of the anchors. The first one is objectness score, which means the probability that an anchor is an object. The second output is the bounding box regression for adjusting the anchors to better fit the object. Using the final proposal coordinates and their objectness score, we then have a good set of proposals for objects.

The RPN is implemented efficiently in a fully convolutional way, using the convolutional feature map returned by the base network as an input. First, we use a convolutional layer with 512 channels and 3 x 3 kernel size and then we have two parallel convolutional layers using a 1 x 1 kernel size, whose number of channels depends on the number of anchors per point.

Since anchors usually overlap, proposals end up also overlapping over the same object. Non-Maximum Suppression (NMS) is used to solve the issue of duplicate proposals. The proposal whose region overlaps a ground truth region more than 70% is regarded as a positive proposal. Otherwise, it is regarded as a negative proposal. After applying NMS, we keep the top 256 proposals sorted by score.

3) Region of Interest (RoI) Pooling: The RoI pooling takes a section of the input feature map that corresponds to positive proposals proposed by the RPN and scales it to 14 × 14 by bilinear interpolation as shown in Figure 4. The result is that from a list of regions with different sizes, we can quickly get a list of corresponding feature maps with a fixed size. Fixed size feature maps are needed for the classifier in order to classify them into a fixed number of classes.

4) The Deep CNN-based Classifier: The deep CNN-based classifier is the final stage in our license plate detection framework. After extracting features for each of proposals via RoI Pooling, we finally need to use these features for classification. The classifier has two different goals: Classify proposals into license plate and background class and adjust the bounding box for each of detected license plate. Our classifier for this stage is defined in Table 2. We use the structure of depthwise separable convolution in MobileNet architecture to build the classifier. The classifier has two fully connected (FC) layers, a box classification layer and a box regression layer. The first FC layer has two outputs, which are fed into the softmax layer to compute the confidence probabilities of being license plate and background. The second FC layer with linear activation functions regresses the bounding boxes of license plate. All convolutional layers are followed by a batch normalization layer and a ReLU layer.

3.2 License Plate Refinement

We found that in some cases, especially with small license plates, license plate cannot be localized precisely enough, making the subsequent recognition task more difficult. Therefore, we perform a refinement step to improve the result of detecting license plate. License plates usually have rectangular or square shape, and characters on license plate are printed on vertical direction. Thus, we use edges information to refine the detected license plates. As illustrated in Figure 5, we first extend boundaries of the detected license plate in both horizontal and vertical direction by 25% to ensure that the extended bounding box contains the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

whole license plate. Next, Sobel operator is used to detect vertical edges and horizontal edges. Considering the strong edges in horizontal direction, we use a 3×1 convolution mask to estimate the gradient in horizontal directions, and a 3×3 convolution mask to estimate the gradient in vertical directions. After applying Sobel operator, we get two grayscale edge maps for each extended bounding box. To further enhance the estimated edges and generate binary edge images, we perform adaptive thresholding [21] on the previously generated grayscale edge maps. Adaptive thresholding first defines a local h×h window surrounding every pixel in grayscale edge images. Then, a threshold value $T(x, y)$ is calculated as a mean of all the pixel location that lie in the h×h box minus C as follows:

$$T(x,y) = \frac{1}{hxh} \sum_{\substack{-h/2< i\, h/2 \\ -h/2<j<h/2}} I(x+i, y+j) - C \qquad (1)$$

where C is the coefficient used to control the threshold.

After getting threshold values, binary edge images are generated by thresholding each pixel p(x, y) in the grayscale edge images using a threshold that is adaptively computed from a local window as follows:

$$I_b(x,y) = \begin{cases} 255\ if\ I_g(x,y) \geq T(x,y) \\ 0\ if I_g(x,y) < T(x,y) \end{cases} \qquad (2)$$

We set h and C to 15 and 18 in this study.

Finally, with the binary edge images generated via AT, we use horizontal projection to get the top and bottom boundaries of the license plate, and vertical projection to get the left and right boundaries of the license plate. This process is illustrated in Figure 5.

4. EXPERIMENTS

Our proposed method is implemented in Python on a PC with 2.8 GHz dual-core CPU, 8 Gb of RAM and GeForce GTX 1050Ti GPU. We use TensorFlow platform to build and perform deep CNN framework, and OPENCV library for real-time processing. In this section, we first introduce the datasets used for training and evaluating our proposed method. Then, evaluation metrics for evaluating our method are described. Afterward, we show the details of training process. Finally, we compare our approach with other state-of-the-art methods in terms of both detection accuracy and run-time efficiency.

4.1 Dataset

Since there are no standard datasets for evaluating license plate detection methods, we use some publicly available datasets, which are often used in recent studies such as Caltech Cars (Real) 1999 dataset [19] and Application Oriented License Plate (AOLP) dataset [11]. Caltech Cars dataset includes 126 images of cars from the rear. These images were taken in the Caltech parking lots with different conditions, backgrounds, distances, and colour. All images have resolution at 896 x 592. There is one USA license plate in each image. For training, we we collect extra 1200 images with US license plates from [22], [23]. Sine there is not much data for training deep CNN framework, we use rotation and affine transformation for data augmentation. AOLP dataset includes 2049 images of Taiwan car license plate and categorized into three subsets: access control (AC) with 681 images, traffic law enforcement (LE) with 757 images and road patrol (RP) with 611 images. AC refers to the cases that a vehicle is at low speed or full stop. The images in this subset are captured under different illuminations and weather conditions. LE refers to the cases that a vehicle is at high speed. The images in this subset are captured by roadside camera, so the background is really cluttered with road sign and multiple plates in one image. RP refers to the cases that a camera is held on a patrolling vehicle, and the images in this subset are taken with arbitrary viewpoints and distances. More details about AOLP dataset can be found in [11]. Because there are no other public images with Taiwan license plates, we use images from different subsets for training the framework and then evaluate on current subset. Again, rotation and affine transformation are used for data augmentation. Figure 6 shows some examples in these datasets.

4.2 Evaluation Metrics

Since there is no uniform criterion for evaluating the performance of different license plate detection methods, most studies follow the criteria used in text detection, i.e. precision, recall and detection ratio. With the purpose of comparing our results with other state-of-the-art results, we also use these criteria for evaluating our method. Detection ratio [8] is defined as ratio between number of correctly detected license plates and number of all ground truth license plates. A license plate is considered to be correctly detected only if the ratio of the area of overlap and the area of union between the detected bounding box and ground

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

truth bounding box is above a threshold. This ratio is defined as IoU as follows:

$$IoU = \frac{area\ (B\_B_{gt} \cap B\_B_{dt})}{area(B\_B_{gt} \cup B\_B_{dt})} \qquad (3)$$

where B_B_{gt} and B_B_{dt} represent the area of ground truth bounding box and detected bounding box respectively. Higher value of IoU mean better quality of the detection.

Precision is defined as ratio between number of correctly detected license plates and detected bounding boxes, while recall is defined as ratio between number of correctly detected license plates and number of ground truths. F-measure, which synthesize both precision and recall, is defined as the following equation:

$$F - measure = \frac{2 \times (precision \times recall)}{precision + recall} \qquad (4)$$

4.3 Performance results

1) Training: For each training image, 256 anchors mapped from the ground truth bounding boxes are randomly sampled. We set the ratio of negative anchors and positive anchors is to 1:1. As in [15], we reject anchor boxes which cross the boundaries of image. Three anchor box scales of 24^2, 34^2 and 54^2 and three anchor box ratios of 1:1, 1:2 and 2:1 are used for the trade-off between recall and processing speed. For the base network, we use MobileNet pre-trained model on ImageNet dataset [24] and further fine-tuned on our training data. The RPN and the MobileNet-based classifier are trained by turns. First, the RPN is trained on a mini-batch and the parameters of the RPN and the base network are updated once. Then, 256 proposals are selected via non-maximum suppression among many proposals generated by the RPN. Next, we use these proposals to train the classifier of the detector. The parameters of the classifier are updated once, and the parameters of the base network are updated once again. This process is iterated, and the entire detection network is optimized. Thus, the RPN and the MobileNet-based classifier share the base convolutional layers. The loss function and the parameterization of coordinates for bounding box regression in this study are the same as those in Faster R-CNN [15]. The Adam algorithm [25] is adopted for optimizing of the loss functions. The initial learning rates of the RPN and the classifier are set to 1e-4 with the learning rate decay of 4e-5 per mini-batch. We trained the network for 300 epochs.

2) Performance results: In this section, we show the detailed performance of our method and then compare the performance of our method with other state-of-the-art methods on Caltech Cars and AOLP dataset. The detection results are presented in Figure 7 (on Caltech Cars dataset) and Figure 8 (on AOLP dataset). As we can see, our proposed method can locate exactly position of license plate in almost complex outdoor scenes.

Comparison results with other methods in Table 3 (on Caltech Cars dataset) and Table 4 (on AOLP dataset) show that our approach performs better on both datasets with the same evaluation criterion. To be specific, with Caltech Cars dataset, our method achieves 98.8% F-measure, with 2.6% higher than method proposed by Li *et al.* [1], and 1.3% higher than the method proposed by Kim *et al.* [2]. With AOLP dataset, our method achieves the best detection performance compared to the previous methods.

Furthermore, as shown in Table 3, our method shows more effective performance when IoU threshold is increased. With IoU at 0.8, our method achieves 97.2% F-measure. Higher values of IoU mean better quality of system. As mentioned above, small license plates are not easy to regress precisely. The localization refinement algorithm designed previous section can be used to future improve the precision of coordinate regression.

The computational speed of our method on AOLP dataset is showed in Table 5. Our framework takes about 180*ms* to get a detection result on a low-end system, while Li et al.'s method [1] costs 1*s*−2*s*, and Hsu et al.'s [11] approach needs averagely 260*ms*. Thus, our method meets the requirement for real-time processing on low-end systems.

5. CONCLUSIONS

This paper presents an effective method for detecting license plate. Our method consists of two steps: deep CNN-based object detector for license plate detection and license plate refinement. First, Faster R-CNN framework and MobileNets architecture are used to detect license plate in images. Then, a refinement step is adopted to improve the result of detecting license plate. We test our method on widely used datasets including Caltech Cars and AOLP datasets. Experimental results show that our method achieves better results than other state-of-the-art methods in terms of detection accuracy and run-time efficiency. Our method can be implemented in real-time on low-end systems. The proposed framework has been proved to be able to detect license plate beyond the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

limitations of colors, shapes, illumination, rotations and distortions in the image. However, the recall of license plates requires further improvement in our future work by adding more annotated data in learning process.

REFERENCES:

[1] Li, H., Shen, C., "Reading car license plates using deep convolutional neural networks and LSTMs", arXiv preprint arXiv:1601.05610, 2016.

[2] Kim, S.G., Jeon, H.G., Koo, H.I., "Deep-learning-based license plate detection method using vehicle region extraction", Electronics Letters, 2017, pp. 1034-1036.

[3] Rafique, M.A., Pedrycz, W., Jeon, M., "Vehicle license plate detection using region based convolutional neural networks", Soft Computing, 2017.

[4] Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M., "Segmentation- and Annotation-Free License Plate Recognition with Deep Localization and Failure Identification", IEEE Transactions on Intelligent Transportation Systems, 2017, pp. 2351-2363.

[5] K. S. Raghunandan, K.S., Shivakumara, P., Jalab, H.A. *et al.*, "Riesz Fractional Based Model for Enhancing License Plate Detection and Recognition", IEEE Transactions on Circuits and Systems for Video Technology, 2017.

[6] Yuan, Y., Zou, W., Zhao, Y. *et al.*, "A Robust and Efficient Approach to License Plate Detection", IEEE Transactions on Image Processing, 2017, pp. 1102-1114.

[7] Xie, L., Ahmad, T., Jin, L. *et al.*, "A New CNN-Based Method for Multi-Directional Car License Plate Detection", IEEE Transactions on Intelligent Transportation Systems, 2018, pp. 507-517.

[8] Gou, C., Wang, K., Yao, Y. *et al.*, "Vehicle License Plate Recognition Based on Extremal Regions and Restricted Boltzmann Machines", IEEE Transactions on Intelligent Transportation Systems, 2016, pp. 1096-1107.

[9] Li, B., Tian, B., Li, Y. *et al.*, "Component-Based License Plate Detection Using Conditional Random Field Model", IEEE Transactions on Intelligent Transportation Systems, 2013, pp. 1690-1699.

[10] Ashtari, A.H., Nordin, M.J., Fathy, M., "An Iranian License Plate Recognition System Based on Color Features", IEEE Transactions on Intelligent Transportation Systems, 2014, pp. 1690-1705.

[11] Hsu, G.S., Chen, J.C., Chung, Y.Z., "Application-Oriented License Plate Recognition", IEEE Transactions on Vehicular Technology, 2013, pp. 552-561.

[12] Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Psoroulas, I.D. *et al.*, "License Plate Recognition from Still Images and Video Sequences: A Survey", IEEE Transactions on Intelligent Transportation Systems, 2008, pp. 377-391.

[13] Zhou, W., Li, H., Lu, Y. *et al.*, "Principal Visual Word Discovery for Automatic License Plate Detection", IEEE Transactions on Image Processing, 2012, pp. 4269-4279.

[14] Huang, J., Rathod, V., Sun, C. *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors", CVPR, 2017.

[15] Ren, S., He, K., Girshick, R. *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks", Advances in neural information processing systems, 2015, pp. 91–99.

[16] Liu, W., Anguelov, D., Erhan, D. *et al.*, "SSD: Single shot multibox detector", arXiv preprint arXiv:1512.02325, 2015.

[17] Howard, A.G., Zhu, M., Chen, B. *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", CoRR, 2017.

[18] Dai, J., Li, Y., He, K. *et al.*, "R-FCN: Object detection via region-based fully convolutional networks", arXiv preprint arXiv:1605.06409, 2016.

[19] "Caltech Cars 1999 (Rear) 2 dataset", http://www.vision.caltech.edu/Image_Datasets/cars_markus/cars_markus.tar, accessed 2003.

[20] Du, S., Ibrahim, M., Shehata, M. *et al.*, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review", IEEE Transactions on Circuits and Systems for Video Technology, 2013, pp. 311-325.

[21] D. Bradley, G. Roth, "Adaptive thresholding using the integral image", J. Graph. Tools, vol.12, no.2, pp.13-21, 2007.

Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22] Benchmarks. [Online]. Available: https://github.com/openalpr/benchmarks, 2013.

[23] L. Dlagnekov, S. Belongie, "UCSD/CALIT2 Car License Plate, Make and Model Database", [Online]. Available: http://http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html, 2005.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[25] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", [Online]. Available: https://arxiv.org/abs/1412.6980, 2014.

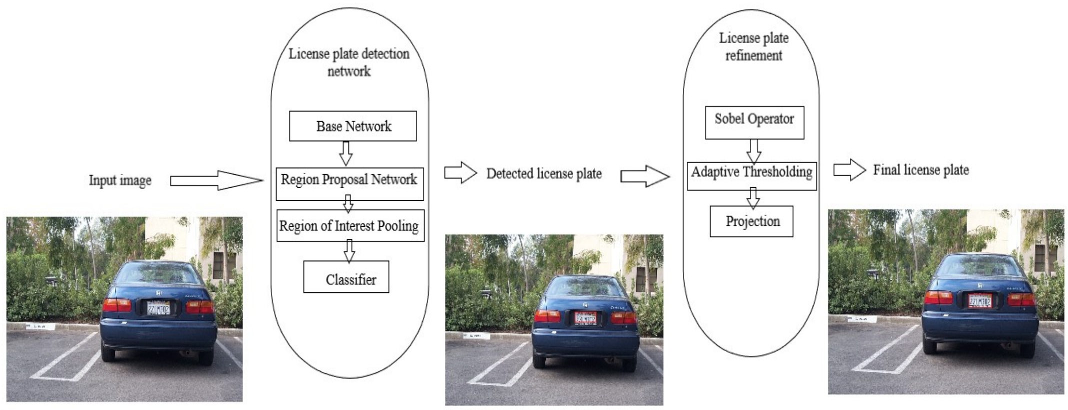

Figure 1: The Overall Framework of Our Proposed Method.

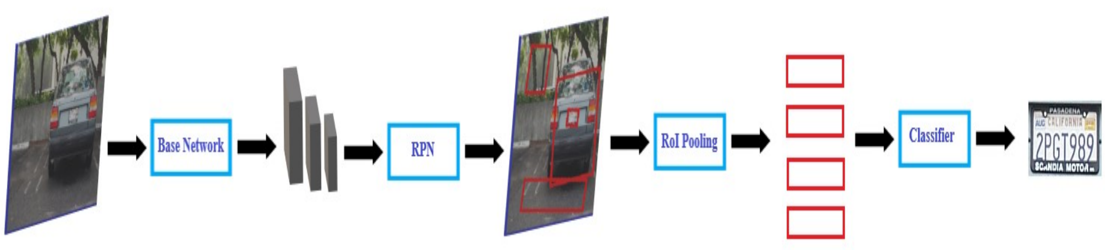

Figure 2: The Overall Framework for License Plate Detection.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

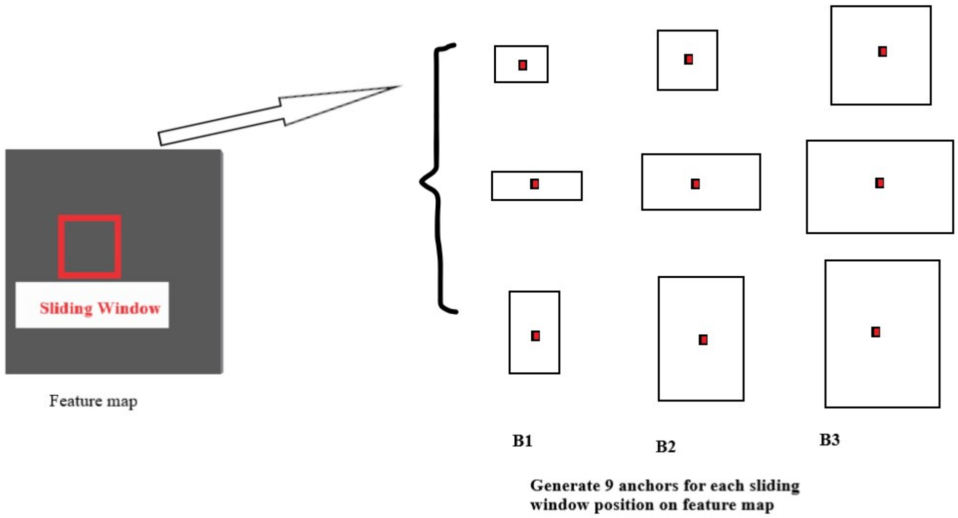

Figure 3: Anchor Boxes Generated By RPN. In This Paper, Each Position of The Sliding Window Generates 9 Anchor Boxes with Different Scales and Aspect Ratios.

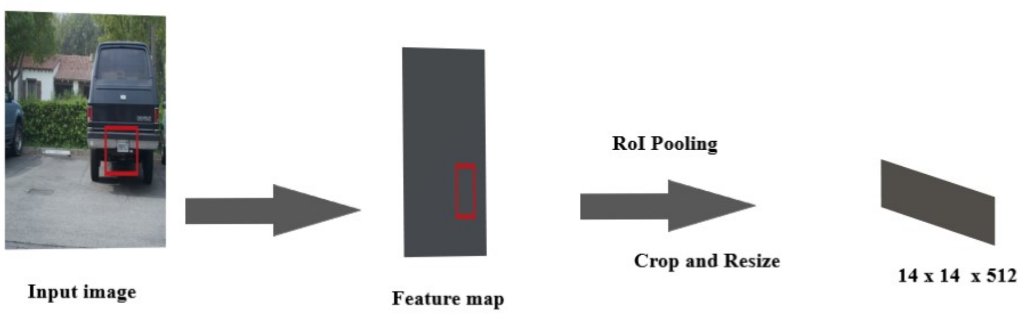

Figure 4: RoI Pooling Process. RoI Pooling Crop and Resize Proposals Generated By RPN To A Fixed Size Feature Map.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

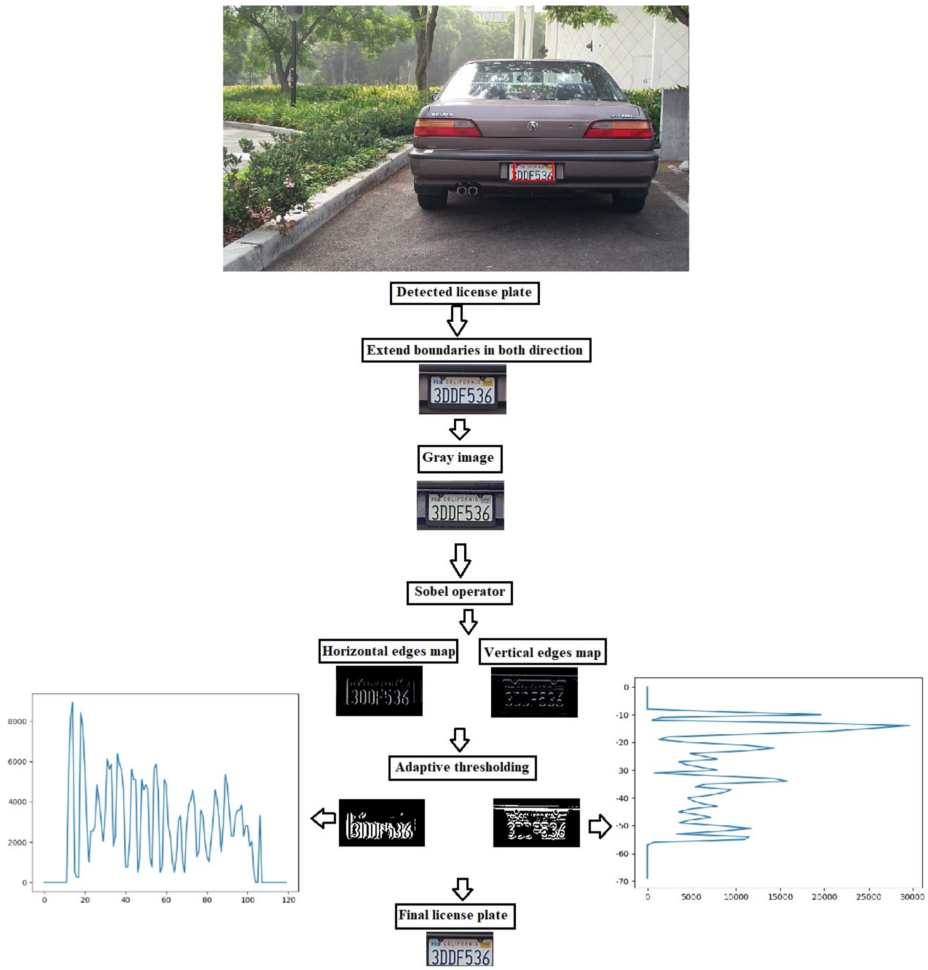

Figure 5: License Plate Refinement Process.

a)

b)

Figure 6: Example Images from Caltech Cars Dataset (a) And AOLP Dataset (b).

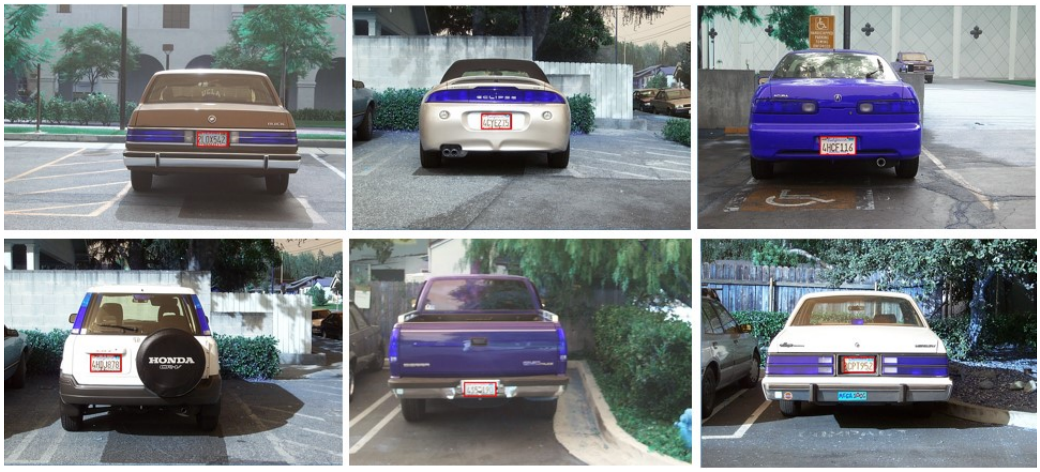

Figure 7: Detection Results on Caltech Cars Dataset.

Figure 8: Detection Results on AOLP Dataset.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: The Architecture of The Base Network.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3\times3\times3\times32$	$224\times224\times3$
Conv dw / s1	$3\times3\times32$ dw	$112\times112\times32$
Conv / s1	$1\times1\times32\times64$	$112\times112\times32$
Conv dw / s2	$3\times3\times64$ dw	$112\times112\times64$
Conv / s1	$1\times1\times64\times128$	$56\times56\times64$
Conv dw / s1	$3\times3\times128$ dw	$56\times56\times128$
Conv / s1	$1\times1\times128\times128$	$56\times56\times128$
Conv dw / s2	$3\times3\times128$ dw	$56\times56\times128$
Conv / s1	$1\times1\times128\times256$	$28\times28\times128$
Conv dw / s1	$3\times3\times256$ dw	$28\times28\times256$
Conv / s1	$1\times1\times256\times256$	$28\times28\times256$
Conv dw / s2	$3\times3\times256$ dw	$28\times28\times256$
Conv / s1	$1\times1\times256\times512$	$14\times14\times256$
$5\times$ Conv dw / s1 Conv / s1	$3\times3\times512$ dw $1\times1\times512\times512$	$14\times14\times512$ $14\times14\times512$

Table 2: The Architecture of The Classifier.

Type / Stride	Filter Shape	Input Size
Conv dw / s2	$3\times3\times512$ dw	$14\times14\times512$
Conv / s1	$1\times1\times512\times1024$	$7\times7\times512$
Conv dw / s2	$3\times3\times1024$ dw	$7\times7\times1024$
Conv / s1	$1\times1\times1024\times1024$	$7\times7\times1024$
Avg Pool / s1	Pool 7×7	$7\times7\times1024$
FC / s1	1024×2	$1\times1\times1024$
Softmax	Classifier	RoI $\times$ 2
FC / s1	1024×4	$1\times1\times1024$
Linear	Classifier	RoI $\times$ 4

Table 3: Performance Comparison on Caltech Cars Dataset (%)

Year	Method	Precision	Recall	F-measure
2012	Zhou et al. [13]	95.5	84.8	89.8
2016	Li et al. [1]	97.5	95.2	96.2
2017	Kim et al. [2]	98.3	96.8	97.5
-	Our method with IoU = 0.5	98.4	99.2	98.8
-	Our method with IoU = 0.8	96.8	97.6	97.2

Table 4: Performance Comparison on AOLP Dataset (%).

Year	Method	AC		LE		RP	
		Precision	Recall	Precision	Recall	Precision	Recall
2013	Hsu et al. [11]	91	96	91	95	91	94
2016	Li et al. [1]	98.5	98.3	97.7	97.6	95.2	95.5
-	Our method with IoU = 0.5	99.6	99.2	98.8	97.4	97.3	97.4

Table 5: Computational Speed Comparison (On AOLP Dataset).

Year	Method	Detection speed (ms)
2013	Hsu et al. [11]	260
2016	Li, et al. [1]	1000-2000
-	Our method	180