An Efficient License Plate Recognition System Using Convolution Neural Networks

Cheng-Hung Lin[1], Yong-Sin Lin[1], and Wei-Chen Liu[2]

Department of Electrical Engineering, National Taiwan Normal University, Taipei 106, Taiwan
Email: brucelin@ntnu.edu.tw; Tel: +886-2-7734-3432
Email: mike83311@gmail.com; Tel: +886-920-420881
Email: chien0928@gmail.com; Tel: +886-928-178846

Abstract

In recent years, license plate recognition system has become a crucial role in the development of smart cities for vehicle management, investigation of stolen vehicles, and traffic monitoring and control. License plate recognition system has three stages, including license plate localization, character segmentation, and character recognition. Although the license plate recognition system has been successfully applied to the environment-controlled smart parking system, it still faces many challenging in the surveillance system such as congested traffic with multiple plates, ambiguous signs and advertisements, tilting plates, as well as obscure images taken in bad weather and nighttime. In this paper, we propose an efficient license plate recognition system that first detects vehicles and then retrieves license plates from vehicles to reduce false positives on plate detection. Then, we apply convolution neural networks to improve the character recognition of blurred and obscure images. The experimental results show the superiority of the performance in both accuracy and performance in comparison with traditional license plate recognition systems.

Key words: license plate recognition system, convolution neural networks, smart city

I. Introduction

License Plate Recognition (LPR) has been widely used in many traffic applications, such as smart parking system, traffic toll system, and security system. In recent years, LPR has played a crucial role in the development of smart cities as a surveillance system for vehicle management, investigation of stolen vehicles and traffic monitoring.

Although LPR has been successfully applied to environment-controlled smart parking systems, it still faces many challenging in the surveillance system such as congested traffic with multiple plates, ambiguous signs and advertisements, tilting plates, as well as obscure images taken in bad weather and nighttime. These variations result in false positives on plate detection and poor LPR accuracy as shown in Fig. 1.

To solve these problems, we propose an efficient hierarchical methodology for license plate recognition system that first detects vehicles using deep learning techniques and then retrieves license plates from detected vehicles to reduce false positives on plate detection. Then in the final stage, we propose a LPR convolution neural networks (LPRCNN) to improve the character recognition of blurred and obscure images. Experimental results show that the methodology achieves 96.12% of vehicle detection rate and 94.23% of plate detection rate. Using LPRCNN, we achieve 99.2% of character recognition accuracy.

Fig. 1: Only using SVM leads to many false positives

This methodology shows the superiority in both accuracy and performance in comparison with traditional license plate recognition systems.

II. Related Works

Conventional license plate recognition system has three stages, including license plate localization, character segmentation, and character recognition. The first stage of license plate localization belongs to the object detection approach, including object localization, feature extraction, and image classification in three stages. Since images may have many different sizes of objects, conventional object detection approaches such as Deformable Parts Models (DPM) [1] use different sized sliding windows to scan the entire image to obtain candidate localizations. And then, object features are extracted from candidate localizations using scale-invariant feature transform (SIFT) [2]. Finally, the object features obtained by SIFT are submitted to the support vector machines (SVM) [3] for classification. The approach of sized sliding windows causes a lot of unnecessary computation. To reduce unnecessary computations, the regional CNN (RCNN) [4] first predicts about 2,000 to 3,000 regional proposals through selective search, then adopts CNN models to extract features from regional proposals and finally completes classification by SVM. After the classification is completed, RCNN optimizes the detection results through the bounding-box regression. The RCNN has two major drawbacks. The first is that the RCNN requires each region proposal to pass the CNN forward, resulting in a large amount of repetitive computations for each single image. The second disadvantage is that it has to train three different models separately. A CNN that generates image features, a classifier that predicts classes, and a regression

ISBN 978-1-5386-4342-6

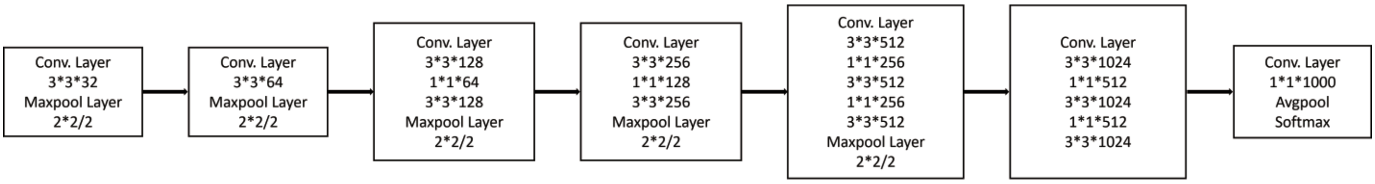

Fig. 2: YOLOv2 Darknet-19 architecture [14]

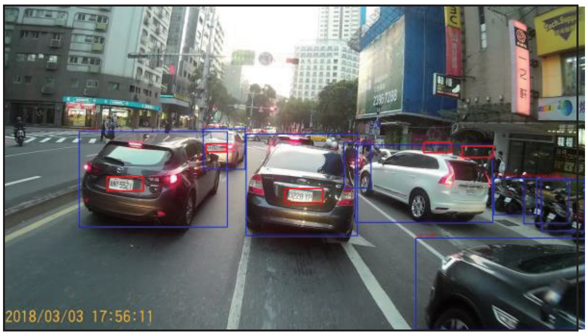

Fig. 3: Vehicle plate detection after vehicle detection by YOLO

model that refines the bounding boxes. This makes RCNN extremely difficult to train.

Subsequently, fast RCNN [5] and faster RCNN [6] are proposed to improve the performance of RCNN. The Fast RCNN proposes a technique called RoIPool (Region of Interest Pooling), which runs a CNN [7] only once for each image to share the calculations in multiple regional proposals. By replacing the SVM with a softmax layer and adding a linear regression layer parallel to the softmax layer to output bounding box coordinates, the Fast RCNN integrates the CNN, classifier, and bounding box regression in a single model. In addition, both RCNN and fast RCNN use selective search to generate a bunch of potential bounding boxes or regions of interest to test. This process is very slow and becomes the bottleneck of the whole process. The Faster CNN proposes to use a single CNN to implement region proposals and classification.

YOLO [8] treats object detection as a single regression problem. YOLO adopts a single CNN model and uses entire images in training and testing phases, so YOLO has better detection accuracy on the images with complex backgrounds.

A typical YOLO can have 45 FPS on a single NVIDIA Titan X GPU and a lightweight version of 150 FPS. Compared with the sliding window and region proposals approaches, YOLO has better accuracy and performance.

III. Hierarchical License Plate Recognition System

A. Vehicle Detection

To accommodate the complexity of images taken from cameras in crossroads, we propose to first detect vehicles and then detect plate on the vehicles. This method can avoid misidentifying traffic signs or advertisements as license plates. Traditional object detection methods such as RCNN and Fast-RCNN, are based on sliding windows or selective search to find possible targets, and then use CNN or other methods to determine whether the target is an object. Due to the size and complexity of photos taken on roads, the use of a sliding window can be very time-consuming and can easily catch the wrong information. In this paper, we adopt YOLOv2 [9] to detect vehicles. As shown in Fig. 2, the Darknet-19 [14] model adopted by YOLOv2 has 19 convolutional layers and 5 maxpooling layers. YOLOv2 first extracts features, reduces dimensions, and performs compression for an entire image using the 19 convolutional layers and 5 maxpooling layers. The original images are reduced to 7*7 or 13*13. Then, YOLOv2 directly performs object recognition and predicts object positions on the reduced images. Compared with the Faster RCNN and SSD (Single Shot MultiBox Detector) [10], YOLOv2 is the most efficient real time object detection approach, which has higher recognition rate and processing speed. In our experiments, YOLOv2 has an average of 96.12% detection rate for detecting vehicles. Fig. 3 shows that after using YOLOv2 to locate vehicle positions, many false positive vehicle license plates were eliminated.

B. License Plate Localization

After Capturing vehicles, we adopt the SVM to detect vehicle's license plates. SVM is a supervised learning method used for classification and regression analysis. We use the SVM OAR (one against rest) architecture with the HOG values of an image as features to train a classifier. In order to identify license plates correctly, we need to train a classifier that can classify license plates and non-license plates. We have to prepare positive and negative samples during the training phase. The positive samples are the license plates and the negative samples are the vehicle's regions without license plates.

Due to the reduced complexity of the negative samples, the recognition rate of license plates is increased.

C. Character Segmentation

After capturing license plates, we have to cut out the area outside the characters, filter out noise, and divided it into single characters for later identification. Fig. 4a shows an example of a detected license plate. The process of character segmentation consists of several steps. First, the captured image is converted to grayscale and then binarized to eliminate noise. Fig. 4b shows the noise-removed binarized plate. Then, we perform a horizontal projection of the license plate image to determine the position of the characters arranged on the license plate. As shown in Fig. 5, The upper and lower borders are removed by horizontal projection. Finally, we perform a vertical projection on the license plate image to determine the position of each character and then divide it into single characters, as shown in Fig. 6.

D. Character Recognition

In the final stage, we propose a LPRCNN model to identify

ISBN 978-1-5386-4342-6

(a) (b)

Fig. 4: (a) An example of a detected vehicle license plate, (b) a binarized plate.

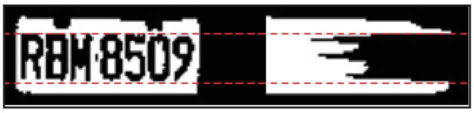

Fig. 5: Eliminating the upper and lower borders by horizontal projection.

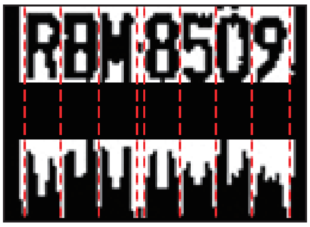

Fig. 6: Separating characters by vertical projection

blurred and skewed characters as shown in Fig. 7. The proposed LPRCNN model is composed of two convolutional layers, two maxpooling layers, two fully connected layers, and one output layer. The output layer contains 34 neurons to correspond to the 34 plate characters.

IV. Experimental Results

The experimental environment is equipped with a CPU host and a GPU device. The host uses the Intel® Core™ i7-4790 CPU, which includes 4 cores operating at 3.60 GHz and two hardware threads per core. The device uses NVIDIA® GeForce® GTX Titan X GPU, which includes 3,840 CUDA cores and 12GB DDR5X DRAM. The operating system is Ubuntu 16.04 LTS.

In this section, we describe our experiments in three parts. In the first part, we use coco 2017 dataset to train a YOLOv2 model for vehicle detection, including cars, trucks and motorbikes. The vehicle detection rate is 96.12%. After vehicles are detected, the detected regions of interest (ROI) of the vehicles will be transmitted to the next stage for license plate detection.

In the second part, we prepare 1,779 positive plate samples and 5,401 negative samples to train an SVM classifier for detecting license plates. All the samples in this experiment are Taiwan's standard license plates captured on the roadside in Taiwan. We use the HOG (Histogram of oriented gradient) as a feature to train the SVM classifier. The plate detection rate, also called recall rate, is 94.23%. After plates are detected, the ROI of the plates are delivered to the next stage for character segmentation and recognition.

In Taiwan, license plate characters consist of the alphabet A-Z, except for "I" and "O", and the number is 0-9, a total of 34 characters.

Fig. 7: blurred and skewed characters

In the third part, we prepare 14,627 labeled samples, of which 60% are training samples and 40% are testing samples, to train the LPRCNN model in Fig. 8. In our experiments, the LPRCNN model achieves 99.2% accuracy on blurred and skewed characters. However, it still face the challenges of recognizing similar characters such as "B" and "8", and "U" and "0".

Basically, precision rate and recall rate are often used to assess the effectiveness of a learning model. The precision rate is defined as (1) where TP and FP denotes true positives and false positives, respectively.

The recall rate is defined as (2) where TP and FN denotes true positives and false negatives, respectively.

Table 1 compares the precision rate of SVM and YOLOv2+SVM. The experimental results show that the accuracy of using only the SVM is very low because many false positive plates are detected. On the other hand, using YOLO and SVM significantly improves the accuracy because many false positive plates are filtered. In general, increasing the sensitivity of license plate detection will increase the recall rate, but the precision rate will decrease as false positives increase. Our proposed architecture can reduce false positives and increase precision rate in high-sensitivity with no change in recall rate. And then, using the LPRCNN architecture to identify the license plate characters, the overall license plate recognition system has good accuracy.

V. Conclusions

In this paper, we have proposed an efficient hierarchical license plate recognition system. The integration of YOLOv2 model and SVM can capture license plates with high accuracy. The LPRCNN model also has high accuracy in character recognition. Experimental results show the superiority of the proposed license plate recognition system in both accuracy and performance.

ISBN 978-1-5386-4342-6

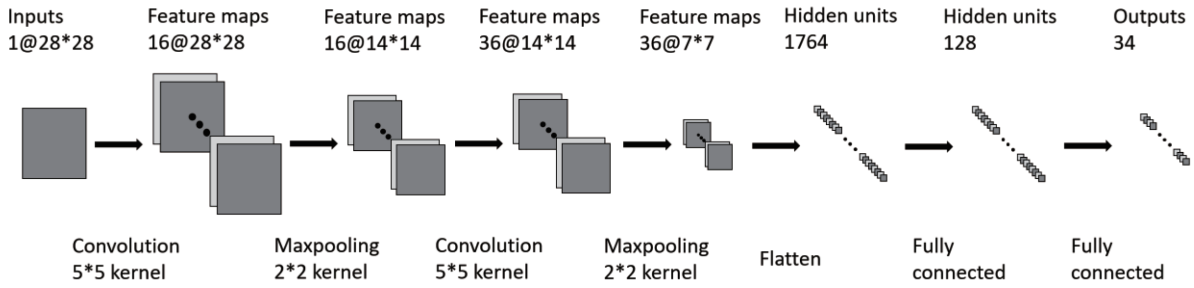

Fig. 8 The proposed LPRCNN for license plate character recognition

TABLE I
Comparison of the accuracy of SVM and YOLOv2 + SVM

Test Image	SVM			YOLOv2 + SVM			Improve
	TP	FP	Precision	TP	FP	Precision	
IMG_01	2	13	13.33%	2	0	100%	86.67%
IMG_02	3	27	10%	3	4	42.86%	32.86%
IMG_03	2	35	5.4%	2	2	50%	44.6%
IMG_04	2	23	8%	2	1	66.67%	48.67%
IMG_05	2	9	18.18%	2	1	66.67%	48.49%
IMG_06	4	42	8.7%	4	2	66.67%	57.97%
IMG_07	2	12	14.29%	2	5	28.57%	14.28%
IMG_08	3	5	37.5%	3	0	100%	62.5%

References

[1] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable Part Models are Convolutional Neural Networks. arXiv preprint arXiv:1409.5403, 2014. in CVPR, 2015.

[2] D. Lowe, Distinctive image features from scale-invariant keypoints. IJCV, 60 (2), pp. 91-110, 2004.

[3] N. Dalal B. Triggs "Histograms of Oriented Gradients for Human Detection" Proc. IEEE Conf. Computer Vision and Pattern

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14), pp.580-587, 2014

[5] R. Girshick, Fast R-CNN, in ICCV, 2015.

[6] K. Rose, "Title of paper with only first word capitalized," in press. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in NIPS, 2015.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015.

[9] J.Redmon and A.Farhadi. Yolo9000: Better,faster,stronger. arXiv preprint arXiv:1612.08242, 2016.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European Conference on Computer Vision, pages 21–37. Springer, 2016.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla ́r, and C. L. Zitnick. Microsoft COCO: Common objects in context. In ECCV. 2014.

[12] W.-C. Liu, C.-H. Lin, A Hierarchical License Plate Recognition System Using Supervised K-means and Support Vector Machine, in Proc. IEEE ICASI 2017, Sapporo, Japan, May 13-17, 2017.

[13] S. Du, M. Ibrahim, M. Shehata, and W. Badawy. Automatic license plate recognition (alpr): A state-of-the-art review. Circuits and Systems for Video Technology, IEEE Trans. on, 23(2):311-325, 2013.

[14] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013-2016.

ISBN 978-1-5386-4342-6