Real-time Automatic License Plate Recognition System using YOLOv4

Ju-Yeong Sung, Saet-Byeol Yu, Se-ho Park
Korea Electronics Technology Institute, Republic of Korea
wudeagles@gmail.com, sbyu@keti.re.kr, sehopark@keti.re.kr

Abstract

We introduce a real-time Automatic License Plate Recognition system that is computationally lighter by eliminating the ROI setting step, without deteriorating recognition performance. Conventional license plate recognition systems exhibit two main problems. First, clear license plate visibility is required. Second, processing actual field data is computationally intensive and the ROI needs to be set. To overcome these problems, we performed plate localization directly on the entire image, and conducted research taking low quality license plate detection into account.

We aim to recognize the license plates of cars moving at high speeds on the road as well as stationary cars using the NVIDIA Jetson TX2 module, which is an embedded computing device.

Keywords: Automatic License Plate Recognition, Plate localization, Character recognition, YOLOv4

1. Introduction

There are many license plate recognition systems these days [1-3]. Most of these systems target large or clear license plates. However, license plate data collected from the actual field are not always appropriate for processing. License plate images comprising field data are often too small or unclear. In fact, field data is often blurred because vehicles are moving at high speeds or there is a lot of noise. Therefore, most of the previous systems proposed have difficulty in processing field data. Recently, studies overcoming this problem have been published, but due to the large amount of computation, they cannot be used in real-time on light, embedded devices.

Our purpose is to create a system that allows real-time license plate recognition on a relatively light device with less computation. To do this, we need to develop a way to recognize relatively small license plates from high-resolution images, such as 3K or 4K. This allows you to skip the step of finding the car area, reducing the amount of computation and speeding up the system.

Real-time object recognition methods known as 1-stage detectors, such as YOLO or SSD, were used in previous works for license plate detection and character recognition. Initially, we trained the model using YOLOv3 and SSD, but these were not efficient because the license plate size was too small as the input image size.

However, YOLOv4 solved this problem. YOLOv4 was able to detect small license plates that were often undetectable by YOLOv3 or SSD.

2. Related work

In this paper, real-time 1-stage detectors SSD and YOLO were used for license plate detection and character recognition. The characteristics of each detector are as follows.

2.1. Single Shot MultiBox Detector

Single Shot MultiBox Detector (SSD), was published by W. Liu *et al.* in 2016 [4]. SSD uses VGG16 as the base network due to its high-quality image classification, and then convolutional feature layers that progressively reduce in size are added to its end so that it is able to predict object at different scales by the aspect ratio. The core of SSD is predicting category scores and box offsets for a fixed set of default bounding boxes using small convolutional filters applied to feature maps. This model has been improved and implemented by many researchers by using ResNet or MobileNet instead of VGG16.

2.2. You Only Look Once

The first version of the YOLO algorithm to detect objects was published in 2016, named YOLOv1 [5]. Prior work on object detection repurposes classifiers to perform detection. Instead, J. Redmon *et al.* frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. In 2017, the year after the initial YOLO release, YOLO9000 (YOLOv2) was published with various improvements, titled “YOLO9000: Better, Fast, Stronger” [6]. In 2018, YOLOv3 [7] was released with very slight changes. The latest version called YOLOv4 [8] was published in 2020. YOLOv4, like YOLOv3, is a combination of up-to-date techniques to improve its performance.

3. KETI-ALPR dataset

KETI-ALPR dataset contains information on license plates collected in South Korea. It consists of two main things. First, the position information of the license plate relative to the overall image is included. Second, information about the characters forming the license plate code is included. The license plate (LP) dataset consists of over 2000 images taken during the day and evening. Each image contains 1 to 4 license plates. Character dataset consists of over 3000 license plate images taken during the day and evening. Each image contains 7 to 9 characters.

Figure 1: Examples from KETI-ALPR dataset

4. Proposed ALPR approach

4.1. LP Detection

In this research, it is important to quickly and accurately detect relatively small license plates in the image without setting an ROI (Region of Interest) on the input. Setting the ROI makes it easier to recognize license plates. However, if you set an arbitrary ROI, the system becomes limited, and detecting the vehicle area and setting it as ROI slows down the system.

We solved this by using YOLOv4. It has improved detection of small objects compared to previous YOLO versions or SSD. Performance and speed can be tuned by adjusting the input network resolution. In our case, the size of 256x256 was selected because it was based on the performance of the NVIDIA Jetson TX2 board. If you are not using a resource constrained platform, it is better to choose 512x512 or 608x608.

We were unable to conduct speed improvement tests because of camera device driver compatibility issues with TensorRT 4. However, it has been studied that using TensorRT 5 and YOLOv4 can provide speed improvements [9].

4.2. Character Recognition

At first glance, in a high-resolution image like 4K, the license plate looks clear. However, when cars are moving at high speeds, it is often difficult to recognize the license plates in its original size, as shown in Figure 2. When using an SSD or previous version of YOLO, accuracy is poor because the features to be calculated are not clear. This can also be solved with YOLOv4. Similar to the LP detection setup, network resolution is set to 256x256 in order to speed up the processing time of the real-time system.

5. Experimental results

We built the dynamic library by modifying only the necessary code of darknet and developed the system.

The license plate at the top of the image is small and the search rate tends to be low. Even if detected, the character part is crushed, making it difficult to recognize. The best recognition is when there is a license plate in the center or at the bottom as shown in Figure 2.

Figure 2: Results of KETI-ALPR system

The experimental results for accuracy and speed can be found in Tables 1 and 2. In general, the network resolution of 608x608 is more accurate than the case of 256x256, but in our experiment, 256x256 was better. This is probably because the input license plate images have widths ranging from 100 to 400 and heights ranging from 40 to 150, which is far smaller than the size of 608, so there are fewer features to calculate.

Table 1: The accuracy of each character detector

Detector	F1-score	IoU	mAP@0.5
YOLOv3 (416)	0.93	84.91	71.53
YOLOv4 (256)	1	94.25	98.36
YOLOv4 (608)	1	89.55	98.12

Table 2: The computational time for each method

	GeForce RTX 2080 Ti		NVIDIA Jetson TX2	
Detector	Time (ms)	FPS	Time (ms)	FPS
SSD	51	19.6	74	13.5
SSD (TensorRT)	4.38	228.3	32.5	30.77
YOLOv4 (256)	9	111	105	9.5
YOLOv4 (608)	25	40	470	2.1

6. Conclusions

We propose a fast and accurate automatic license plate recognition system that is suitable for processing field data. By enabling small detection in the image, the burden of setting the ROI area was relieved, reducing the amount of computation and increasing the speed. In addition, the performance is improved by using YOLOv4 detector that can recognize some contorted characters.

As future work, we propose to increase the character recognition dataset. Since most of the datasets were recently photographed in Seoul, the number of old license plates, regional license plates, and unusual character datasets is less than that of general numeric datasets. Addition of rare license plates to the dataset can improve the performance of the training model.

7. Acknowledgements

This work was supported by the IT R&D program of MOTIE/KEIT. [10077018, Development of integrated embedded platform and commercial prototype equipped with high-speed LPR function]

References

[1] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), 2018, pp. 1–10.

[2] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 9, pp. 2351–2363, Sep. 2017.

[3] R.-C. Chen, et al., Automatic license plate recognition via sliding-window darknet-YOLO deep learning, Image Vis. Comput. 87 (2019) 47–56.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 779–788.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 6517–6525

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, arXiv:1804.02767. [Online]. Available: https://arxiv.org/ abs/1804.02767

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[9] M. Verucchi, L. Bartoli, F. Bagni, F. Gatti, P. Burgio and M. Bertogna, "Real-Time clustering and LiDAR-camera fusion on embedded platforms for self-driving cars", in proceedings in IEEE Robotic Computing (2020)