A new car logo extraction method based on license plate detection

Hanwen Ren\textsuperscript{1*}

\textsuperscript{1}Brown University, Providence, RI, USA
\textsuperscript{*}renhanwen2015@163.com

Abstract. Nowadays, intelligent transportation system is becoming the new trend because of the development of modern traffic, which serves to help gather traffic data, control traffic and improve road safety. One big challenge within intelligent transportation system is car model detection, which provides important information for traffic monitoring and planning. Currently, most car model extraction focus more on one-pass solution, which directly finds out the car logo region. In this paper, a new hybrid car logo extraction method based on license plate detection is proposed. First, a text detection is applied on car images to locate the license plate. Then, within a region of interest above license plate, a sub region is extracted to locate the car logo based on edge information. The fundamental concept is that using the relative location information between the license plate and car logo can take the most advantage of existing state-of-the-art license plates detection methods which is practically accurate, robust and efficient. Experimental results on OpenALPR dataset demonstrate its good performance in extracting car logos.

1. Introduction

Traditional traffic system is current facing more and more problems with the increasing number of vehicles on the road. Thus, intelligent transportation system is in great need to address this situation. To achieve traffic monitoring and planning, intelligent transportation system\cite{1} should be able to recognize car features from surveillance camera images or videos. Currently, intelligent transportation system relies heavily on using car license plate information, and, thus, many works have been done on the car license plate detection\cite{2}. Even though they have shown robust, efficient, and accurate performance in most of practical applications, car plate detection can also fail in some special cases like fake plates and noisy image resulted from poor illuminance\cite{3,4}. When car plate detection fails, drivers may receive improper fines or even more serious consequences. Thus, a more accurate way to locate a car is to use license plates as well as car model.

Currently, there are already some work that is done in Vehicle Make and Model Recognition (VMMR)\cite{5} field. Gao Y et al.\cite{6} proposed a model detection algorithm using deep learning. In their research, they first crop the front part images vehicles and feed them into a three-layered Restricted Boltzmann machine. Their system is claimed to have a better result compared to other methods like local binary pattern (LBP)\cite{7} and scale invariant feature transform (SIFT)\cite{8}. Pearce G et al.\cite{9} introduces a way that utilize frontal images of cars to do classification. In their experiment, they feature extractor are Improved Square Mapped Gradients (ISMG) and canny edge detector. Extracted features are fed into K nearest neighbour and Naive Bayes to do classification. They claim to achieve the best accuracy of around 90%. Sarfraz MS et al.\cite{10} uses a probabilistic method in model detection. A feature descriptor called Local Energy based Shape Histogram (LESH) are used to extract features from image.
LESH results are further used to calculate the maximum posterior from database and finally do the vehicle type recognition. Their accuracy is claimed to be 94%. Psyllos AP et al.[11] purposed an enhanced SIFT based feature matching scheme for VMMR. Their research uses phase congruency which is a segmentation method to find car logos and SIFT based features combined with a multiple match method called merged feature matching. Yadav AA et al.[12] uses Histogram of Oriented Gradients to build feature vectors for logo detection and achieves precise and accurate result. In addition, there are also researches that focus on improving the performance of existing method instead of coming up with new methods. Keser RK et.al.[13] uses a variation of SIFT with reduced dimensions to save half of the time with less than 20% drop in accuracy.

Even though many car model detections have been developed, they still suffer from limited accuracy in practical applications. To address this, this paper proposes a new hybrid car logo extraction method to make the most of existing state-of-the-art car license plate detection which has been demonstrated to achieve high accuracy. This algorithm is proposed based on the observation that the relative location between license plate and car logo is fixed in most cases. Car logo always locates above license plate. Considering the relationship between car license plate and car logo are fixed as well as car license plate detection is more accurate, it is best to develop a sequential framework to achieve car plate license detection and car model detection. First, we will use existing text detection algorithm to find out the region of car license plate. The second step is to extract a region that contain car logo information based on the size of the car license plates. As car logo contains more edges compared to the background, we do edge detection in the extracted area and use a fixed size small sliding window to find out the region what contain the most edges to be the final region that contains car logo. The unique feature of this hybrid method is that it can find out car license plate and car logo consequently instead of finding them parallelly, which saves computation power. Another advantage is that since the final logo is extracted based on license plate location, it is more trustworthy compared to extract logo directly. The reason of this is that if filters out some fake car logo that is not from an actual car on the road. These fake car logos may come from billboards on roads as well as other structure that can be detected as car logo.

2. a new car logo detection based on license plate detection

Figure 1. Flowchart of the proposed car logo detection based on license plate detection

The overall structure of our hybrid method is shown in the block chart above. In this section, we will discuss each stage in detail.

2.1 Car license plate detection with EAST

We detect car license plate by a text detector called EAST (an Efficient and Accuracy Scene Text detection pipeline), which is proposed in Zhou et.al.’s[14] paper. This method is a powerful pipeline that can achieve fast and accurate text detection in natural scenes. It can directly predict word or text lines of arbitrary orientations and quadrilateral shapes with a signal neural network. EAST consists of two parts: a fully convolutional network (FCN) and a Non-Maximum Suppression (NMS) stage. The FCN outputs text regions and NMS give final results. This method is tested on several standard benchmarks like ICDAR 2015, MSRA-TD500 and COCO-Text. The results show that they can outperform most of the existing methods in both accuracy and efficiency. Since the content of car license plates are text, we regard EAST as a suitable method to be our license plate detector.

\[
Plate(x_s, x_e, y_s, y_e) = EAST(original\_img)
\]

where \(x_s, x_e, y_s, y_e\) means the starting and ending position for the detected license plate in x/y axis.
2.2. Car logo region of interest extraction

As most of the car logo is located above license plate, in this stage, we want to extract a region of interest based on the size of detected license plate. What we want for this region is that: 1. it is large enough to contain the car logo, 2. it is smaller enough to make later computation fast and avoid containing too much noise. In real image, the vehicle can appear close to the camera or far away from camera, which makes it impractical for us to use a fixed size extraction region. If the vehicle is too close, fixed size extraction region may not be able to contain the car logo we want. On the other hand, if the vehicle is too far away, fixed size extraction region may be too large and include other noise to make the detection fail. Thus, we use a dynamic method when choosing the size of region that contains car logo, which is based on the size of detected license plate. Let’s assume our license plate region has width D and height H, we choose the region with width 2*D and height 4*H. The new region is located above license plate region with their horizontal center aligned with each other.

\[
ROI(x_e) = Plate(x_e) + \left(Plate(x_e) - Plate(x_s)\right) \times 0.5
\]

\[
ROI(y_e) = Plate(y_e) + \left(Plate(y_e) - Plate(y_s)\right) \times 4
\]

\[
ROI(x_s) = Plate(x_s) - \left(Plate(x_e) - Plate(x_s)\right) \times 0.5
\]

\[
ROI(y_s) = Plate(y_s) - \left(Plate(y_e) - Plate(y_s)\right) \times 4
\]

2.3. Edge detector

We use the canny edge detector to get edges in our region of interest that contains logo information. The canny edge detector is an edge detection operator that contains multiple stages to detect edges in images. This method was development by Josh F. Canny in 1986[15]. Basically, this method can be divided by five stages. The first stage is gaussian filter that is used to smooth image and remove noise. The second stage is to calculate the gradient magnitude and direction for each pixel of the image. Then, Non-maximum stage[16] kicks in to get the local maximum of the edge perpendicular to the gradient direction and result in thinner edges. The fourth stage is to get rid of fake edges caused by noise or color variation.
with double threshold. Finally, edge tracking by hysteresis is used to further filter out noise edges by checking if all weak edges are connected to strong edges. Canny edge detection is widely used in computer vision field and proved to be reliable in edge detection, which is why we chose it as our edge detector.

\[
\text{Edges} = \text{Canny edge detector}(\text{ROI}(x_s, x_e, y_s, y_e))
\]  

(6)

Figure 4. Canny edge detection

2.4. Car logo Extraction with sliding window

After we edge detection in our car logo region, the final step is to extract a sub region that contains actual car logo from the previous detected region of interest. Car logo should contain more edges compared to other parts in this region. Thus, we choose the sub region that contains to most edges. A sliding window method[17] is used there. The size of the square sliding window W in equation (7) depends on the size of region of interest and it goes from left to right, top to bottom. Each time, the window shifts one eighth of its width. All windows store a value which means how many edges they contain, and the region contains the most edges is returned as our car logo extraction result. Note that sometimes we have more than one region of interest from stage two, in this case, we will loop through all of them and still return one region as final result.

\[
W = \min (\text{ROI}(x_e) - \text{ROI}(x_s), \text{ROI}(y_e) - \text{ROI}(y_s))
\]  

(7)

\[
\text{final region} = \arg\max(\sum \text{edges in window})
\]  

(8)

Figure 5. car logo extraction

2.5. Discussion

Overall, our method has the following advantages. Firstly, it is based on license plates detection which is more mature and developed compared with directly car logo detection/extraction methods. Secondly, our method requires less computational power. Thirdly, the logic is easy to follow which in result makes it easy to finetune and debug in the future compared to methods involving deep learning.
3. Experiment result

3.1. Image dataset
The dataset we used in this experiment is OpenALPR benchmarks. OpenALPR is a car license plate detection software. As the first stage is car license plates detection, we thought their dataset would be a good choice to test our algorithm. This dataset contains images taken from the front and back of cars in the US (United States), EU (European Union) and BR (Brazil). From their description, these samples are randomly selected from a set of annotated images. These images come from real work samples are not guaranteed to be ideal.

![Sample Images from OpenALPR Benchmark](image)

3.2. Experiment Results
As we want to check if the result images contain car logo or not, we first get rid of those car models without logo at front or back. In order to verify experiment results, we manually go over all extracted regions from our algorithm and verify if it contains the actual car logo or not. The following table shows the summary of results.

|       | US    | EU    | BR    | Total |
|-------|-------|-------|-------|-------|
| Success | 64    | 41    | 73    | 178   |
| Fail   | 20    | 19    | 22    | 239   |
| Success rate | 76.2% | 68.3% | 76.8% | 74.5% |

(a) (b)
3.3. Success cases analysis:
The images shown in Figure 6 are some examples that we successfully extract car logos from the image with our algorithm. All these examples contain a valid car logo around center region which confirms the feasibility of our method in real world dataset. In some of the cases, the car logo is not exactly at the region of the box, which is caused by the final region size choice. It seems like our final box choice large enough to cover the whole logo region. However, in some cases, the final box can be too big compared to the logo due to car model difference or license plates size difference. Overall, our method is practicable in most cases we do experiments on.

3.4. Failed Cases Analysis:
Since there are four stages in our method, we will analyse our failed cases by stages.

To begin with, EAST algorithm may not find the current area that car license plates are located. If the algorithm fails at this stage, all later stage will not make too much sense. There are cases when EAST algorithm finds background noise as car plate because they also contain text, such as billboards along the highway and the text painted on the car. When this situation happens, there is high chance that our region of interest does not contain any car logo and in result cause the algorithm to fail.

Secondly, due to some special cases, our region of interest logo may not actually contain the car logo. This happens when we encounter some special car models. Currently, the criteria when selecting our bounding box area is that we want it to be large enough to contain car logo. On the other hand, we also want it to be small in order to get rid of the noise. The reason why car logo can be found in our algorithm is that it contains more edges compared to the background which is part of the car front or trunk in most cases. When the region is too large, not only car front/trunk will be included but also background near the car, such as road sequence and another car. Thus, after experimenting with several settings, the ideal bounding box selection rule is to use twice the plate width and forth the plate height. However, this does not apply for all car models and all situations. Some car models have their logo outside our region of interest. In these cases, our algorithm will fail as the region of interest does not contain the car logo.

Thirdly, canny edge detector may not be able to find the edges of the car logo. This can happen when the car logo is too small and blurry. As the image dataset is not ideal, some vehicles can be far away from the camera. In these cases, edge detector may not treat them as valid edges as the signal is too weak.

Finally, our sliding window method sometimes finds the wrong area and return it as the car logo area. This happens when the car images contain the following three cases: 1. the car is equipped with large and shining grill. 2. the car has a rear windshield wiper. 3. the lights of the car is on. 4. the car has some text painting. All these cases will result in strong edges in the background and the car logo cannot stand out. Since we want to choose the area that contain most edge points, it may also fail when the background has too many edges.
Figure 8. failed logo extraction examples, (a) grill (b) wiper (c) light (d) text painting

4. Conclusion
In this paper, we proposed a new car logo extraction method based on license plate detection. One advantage of this method is that it utilizes current matured method in car license plate detection and perform logo extraction based on that. Also, our algorithm does not require high computational power. We can get the car logo by just doing several simple steps after license detection. In this way, it views car logo extraction as an extension of car license detection instead of a whole different field. Our method consists of four stages: 1. car license plate detection with EAST. 2. car logo region of interest extraction. 3. edge detector. 4. actual logo extraction. We tested our result on OpenALPR benchmark and get an overall success rate of 74.5%.

Reference
[1] Smart Traffic Management System. International Journal of Recent Trends in Engineering and Research 4 (2018), 568–578.
[2] Han, J., Yao, J., Zhao, J., Tu, J. & Liu, Y. (2019) Multi-Oriented and Scale-Invariant License Plate Detection Based on Convolutional Neural Networks. Sensors 19.
[3] Karen Panetta, Long Bao, Sos Agaian, (2016) “Sequence-to-Sequence Similarity-Based Filter for Image Denoising,” IEEE Sensors journal, vol. 16, no. 11.
[4] Karen Panetta, Long Bao, Sos Agaian, (2018) “A New Unified Impulse Noise Removal Algorithm Using a new Referenced Sequence-to-Sequence Similarity Detector,” IEEE Access, vol. 6, pp. 37225-37236.
[5] Lee, H. J., Ullah, I., Wan, W., Gao, Y. & Fang, Z. (2019) Real-Time Vehicle Make and Model Recognition with the Residual SqueezeNet Architecture. Sensors 19.
[6] Gao, Y. & Lee, H. J. (2015) Moving Car Detection and Model Recognition based on Deep Learning.
[7] Yasmin, S. & Rana, M. M. (2016) Performance Study of Soft Local Binary Pattern over Local Binary Pattern under Noisy Images. International Journal of Electrical and Computer Engineering (IJECE) 6, 1161.
[8] Chris, L. A., Mulyawan, B. & Dharmawan, A. B. (2016) A Leukocyte Detection System Using Scale Invariant Feature Transform Method. International Journal of Computer Theory and Engineering 8, 69–73.
[9] Pearce, G. & Pears, N. (2011) Automatic make and model recognition from frontal images of cars.
2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).

[10] Sarfraz, M. S., Saquib Sarfraz, M., Saeed, A., Haris Khan, M. & Riaz, Z. (2009) Bayesian prior models for vehicle make and model recognition. Proceedings of the 6th International Conference on Frontiers of Information Technology - FIT '09.

[11] Psyllios, A. P., Anagnostopoulos, C.-N. E. & Kayafas, E. (2010) Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme. IEEE Transactions on Intelligent Transportation Systems 11, 322–328.

[12] Yadav, A. A. & Bhalchandra, A. S. (2018) Vehicle Logo Detection Using Histogram of Oriented Gradients. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS).

[13] Keser, R. K., Ergün, E. & Töreyin, B. U. (2018) Vehicle Logo Recognition with Reduced-Dimension SIFT Vectors Using Autoencoders. Proceedings 2, 92.

[14] Zhou, X. et al. (2017) EAST: An Efficient and Accurate Scene Text Detector. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Canny, J. (1987) A Computational Approach to Edge Detection. Readings in Computer Vision 184–203.

[16] Hosang, J., Benenson, R. & Schiele, B. (2017) Learning Non-maximum Suppression. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[17] Rao, N. M. & Naidu, M. M. (2017) Sliding Window Algorithm. Sliding Window Algorithm for Mobile Communication Networks 15–34.