



Liping Zhu*, Shang Wang, Chengyang Li and Zhongguo Yang

License Plate Recognition in Urban Road Based on Vehicle Tracking and Result Integration

<https://doi.org/10.1515/jisys-2018-0446>

Received November 12, 2018; previously published online September 7, 2019.

Abstract: Multiple surveillance cameras provide huge video resources that need further mining to collect traffic stream data such as license plate recognition (LPR). However, these surveillance cameras have limited spatial resolution, which may not always suffice to precisely recognize license plates by existing LPR systems. This work is focused on the LPR method in low-quality images from surveillance video screenshots on urban road. The methodology we proposed is based on vehicle tracking and result integration, and we recognize the plate with an end-to-end method without character segmentation. First, we track each vehicle to get vehicle tracking sequence. Moreover, a plate detector based on an object detection framework is trained to detect license plates of each vehicle from the sequence and a license plate sequence is formed. In addition, an end-to-end convolutional neural network architecture is applied to recognize license plates from the sequence. Finally, we integrate the recognition result of continuous frames to get the final result. Evaluation results on multiple datasets show that our method significantly outperforms others without segmentation or integration in real traffic scene.

Keywords: Convolutional neural network (CNN), license plate recognition (LPR), low quality, result integration, vehicle tracking.

1 Introduction

License plate recognition (LPR) is a common and key function of intelligent traffic system. Based on the LPR system, there are some common application instances such as traffic flow analysis, parking violation enforcement, and automatic vehicle speed measurement.

However, the real traffic environment in urban road is complicated. Usually, license plate images are too vague to be recognized, which means that the pixels are lower than 32×100 and the rotation angle is beyond -10° to 10° . Other problems such as partly blocking and uneven illumination also affect the recognition results. Under the above conditions, extracting features from images only may not perform well. To alleviate this situation, some papers came up with new methods that connect license plate images from video stream.

In Figure 1, we propose an LPR framework based on vehicle tracking and result integration. At first, we track one vehicle from video stream and get its trajectory sequence. Then, license plate images and recognition results are collected through the convolutional neural network (CNN) model. At last, the results are integrated based on some features, such as the frequency of each character shown or the total confidence of each label. Our framework can complement advantages of each recognition result and

*Corresponding author: Liping Zhu, China University of Petroleum Beijing, Key Lab of Petroleum Data Mining, 18 Fuxue Road, Changping, Beijing 102249, China, e-mail: zhuliping@cup.edu.cn. <https://orcid.org/0000-0002-7841-5578>

Shang Wang and Chengyang Li: China University of Petroleum Beijing, Key Lab of Petroleum Data Mining, 18 Fuxue Road, Changping, Beijing, China

Zhongguo Yang: North China University of Technology, Information Science and Technology, No. 5 Jinyuanzhuang Road, Shijingshan District, Beijing, China



Figure 1: Framework of the LPR System Based on Vehicle Tracking and Result Integration.

eliminate the bad cases. The experiment results show that the LPR framework is robust to most kinds of situations.

2 Related Work

The LPR system mainly consists of two function parts: license plate detection and character recognition. Several approaches of license plate detection were mentioned in Refs. [4, 11, 12, 20, 22, 27], but we do this with object detection algorithms [6, 7, 14, 16–19] based on deep learning. Single-shot multibox detector (SSD) [14] is selected as the method to locate license plates in the manuscript.

2.1 Segmentation-Based LPR

Traditional methods of LPR are typically based on optical character recognition (OCR) techniques and they take segmentation as a previous step of recognition. Qiao et al. [15] used histograms of pixels projected horizontally and vertically to locate positions of characters and separated them along the projected axis. In contrast, Anagnostopoulos et al. [1], Giannoukos et al. [5], and Chang et al. [2] used connected component techniques, which separated the characters in binarized license plate images depending on four or eight neighborhood connectivity. As the development of machine learning, Wen et al. [26] and Thome et al. [24] proposed that OCR can be replaced with classifiers based on machine learning, which improved recognition rate obviously. The disadvantage of segmentation-based LPR methods was their sensitivity to rotation and

distortion because of the significant influence of segmentation. A robust way of recognition is imperative to seek.

2.2 Deep Learning-Based LPR

With the development of deep learning and its successful application, some novel methods have been adopted in the LPR system. Cheang et al. [3], Jain et al. [9], and Špaňhel et al. [23] applied deep neural network to LPR without character segmentation. The features of license plate images were extracted directly by CNN and used to train an LPR model. Compared to the segmentation-based method, this method was more robust. However, it usually did not have good performance because license plate images are not clear enough in real traffic scenes.

2.3 Time Sequence-Based LPR

The methods we mentioned above could not recognize license plates accurately in real traffic scenes; thus, some new research intended to study the recognition based on time sequence. Li and Shen [13] combined recurrent neural network and long short-term memory to train a model of feature sequence of license plate images leading to a better recognition performance of CNN. According to vehicle trajectory, Seibel et al. [21] got several license plate images at different moments. These images with poor resolution were integrated to a single super-resolution image. Then, they used normal LPR algorithm to recognize the license plate. Wang et al. [25] tried to generate synthetic training data with generative adversarial networks to improve the recognition rate based on a moving camera. Although the time-sequence information was mentioned in these papers, it was just used to train models or fix up the images.

Time-sequence information is used to integrate all recognition results, and we can get an affirmatory result. Excluding the bad cases, the final result is much closer to the true label. Our method outperforms surveillance videos in real traffic scenes.

3 Methodology

3.1 Process

The process of the proposed method consists of three steps: vehicle detection and tracking, plate detection based on trajectory, and plate recognition. The first step is to detect and track vehicles based on SSD algorithm. In addition, we apply a trained plate detector model also based on SSD to locate the license plates. Then, we recognize the plates using a well-trained CNN model. In the last step, several recognition plate results are integrated to one final plate result referring to the frequency or confidence of each character.

3.2 Vehicle Detection and Tracking

We locate the vehicle in each frame of the surveillance video. Tracking process is to line up the vehicle images to establish the trajectory of each vehicle. We use a similarity model to judge if the vehicles of two neighboring frames are matched. The similarity of two objects is estimated with the help of color histogram, Jaccard overlap, and cell matching similarity between two object patches.

In the vehicle similarity computing process, the widely used features are color histogram [10] and cell matching similarity. In addition, we added the Jaccard similarity to the final similarity result, which ruminates the little change in the spatial space for a vehicle in the adjacent frames. Figure 2 gives the procedure of vehicle similarity computing.

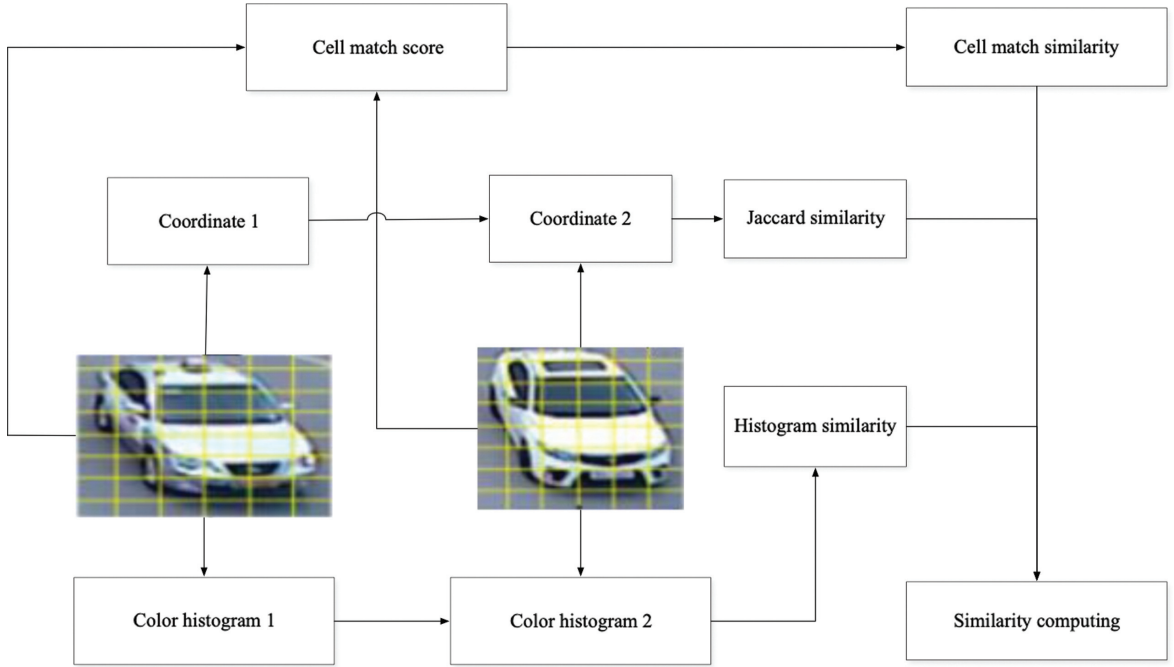


Figure 2: Procedure of Computing the Similarity between Two Objects.

In the similarity computing process, the color histogram with 24 bins with each RGB channel is spat into 8 bins. The color histogram is normalized by

$$\text{hist}(i)' = \frac{\text{hist}(i)}{\sum (\text{hist}(i))^2} \quad (1)$$

The correlation of the two histograms is calculated by the Pearson product-moment correlation coefficient equation, which is

$$\gamma_{xy} = \frac{n \sum \text{hist}_x'(i) \text{hist}_y'(i) - \sum \text{hist}_x'(i) \sum \text{hist}_y'(i)}{\sqrt{n \sum \text{hist}_x'^2(i) - (\sum \text{hist}_x'(i))^2} \sqrt{n \sum \text{hist}_y'^2(i) - (\sum \text{hist}_y'(i))^2}} \quad (2)$$

The Jaccard coefficient measures the similarity between finite sample sets and is defined as “the size of the intersection divided by the size of the union of the sample sets”. The computing process is intuitively given by

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

To estimate the image matching score, we divide the image into 6×8 cells. For each cell, the mean and variance of pixels are computed. The distance of two cells is

$$\text{cell}_{xy}(i, j) = \begin{cases} 1, & \text{if } |\text{mean}_x(i, j) - \text{mean}_y(i, j)| < \theta_m \\ & \text{and } |\text{var}_x(i, j) - \text{var}_y(i, j)| < \theta_v \\ 0, & \text{x} = 0 \end{cases} \quad (4)$$

The final cell matching similarity of two images is

$$\text{cell_match_similarity}_{xy} = \frac{\sum_{i,j} \text{cell}_{xy}^2(i, j)}{\text{Total_number_of_cells}} \quad (5)$$

The similarity of the two detected objects consists of three parts: histogram similarity, cell matching similarity, and Jaccard similarity, which can be expressed as

$$\text{Similarity}_{xy} = (\gamma_{xy} + \text{Jaccard Similarity}_{xy} + \text{cell_match_similarity}_{xy})/3 \quad (6)$$

3.3 License Plate Detection

As the primary step of the LPR system, license plate detection has great influence on both recognition speed and rate. In our work, we use SSD to locate the license plates. The precision and recall of license plate detection are 82.5% and 89%, respectively, under the real environment of urban road.

3.4 End-to-End LPR System

DenseNet169 is chosen as the CNN architecture to extract features, and three fully connected (fc) layers are added behind that. The second and third fc layers are both sliced to seven layers. Each layer of fc8 outputs different numbers of feature maps based on each character's class. For example, fc8_1 outputs 31 feature maps because fc8_1 represents Chinese characters and it has 31 classes. Figure 3 shows the schematic diagram of our model.

Based on multilabel learning, we can build an end-to-end model of Chinese LPR. There are seven characters on a Chinese license plate.

Because of the poor recognition on Chinese characters, it is ignored in this paper. We mainly focus on the recognition of numbers and English letters.

3.5 Integration of LPR Results

However, license plate images are too blurred to be recognized just by the LPR system under the environment of real urban road. The integration of recognition results becomes a trend. Based on vehicle trajectory, we

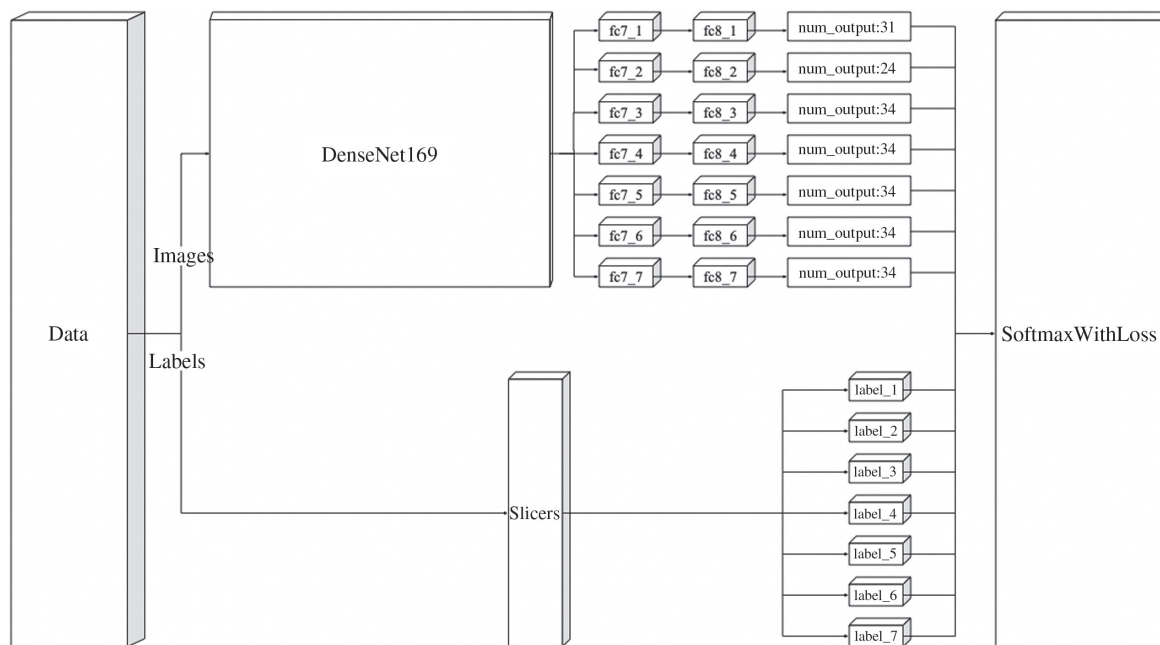


Figure 3: CNN Architecture.

get vehicle tracking sequence of one vehicle. After license plate detection and recognition, the integration of all the results will be done. We propose two methods for result integration and they are frequency and confidence based correspondingly.

3.5.1 Integration Based on Frequency

Counting results based on frequency of each character’s recognition result is the simplest method as far as we are concerned. According to the literal meaning, we count the number of each result’s occurrence of each character and figure out the most frequent one. Then, we line them up to get the final result.

Figure 4 shows us an example of a plate numbered “JAM301”. During vehicle tracking, we catch this plate five times. However, the LPR result of each frame is not correct. What we need to do is to integrate the recognition result on each character. The most frequent label of the first character’s recognition result is “J”, which appears five times. The most frequent result of the second character is “A”, whose frequency is 3. The rest concluded in this way are “M”, “3”, “0”, and “1” separately. Therefore, the integration result is “JAM301”, which is the correct number of this plate.

This method is better than LPR without result integration, but it is not convincing enough as shown by the experiments.

3.5.2 Integration Based on Confidence

Because the method of counting frequencies is not convincing, another way, which is counting confidences, is proposed. One character has only one recognition result, whereas, thanks to Softmax, each label associated is given a confidence to represent the probability to be the true value. What we should do is to count these confidences and find the label with the maximum confidence. This label is the result of the character that we are looking for.

Figure 5 demonstrates an example LPR of a plate numbered “JT0354”. For convenience, some of recognition results are omitted. At each recognition result, we give four label ordered by confidence at each character. The confidence of each label is shown behind the label in each table. Taking the first character, which is “J”, as an example, during the trajectory sequence, we compute the sum of each label’s confidence and select

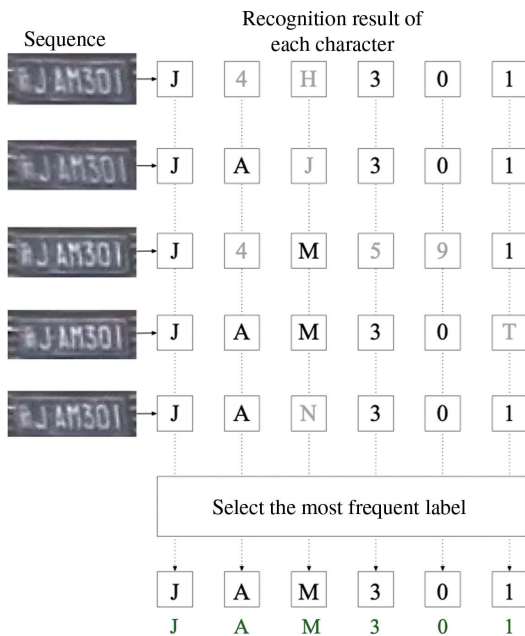


Figure 4: Integration Based on Frequency.

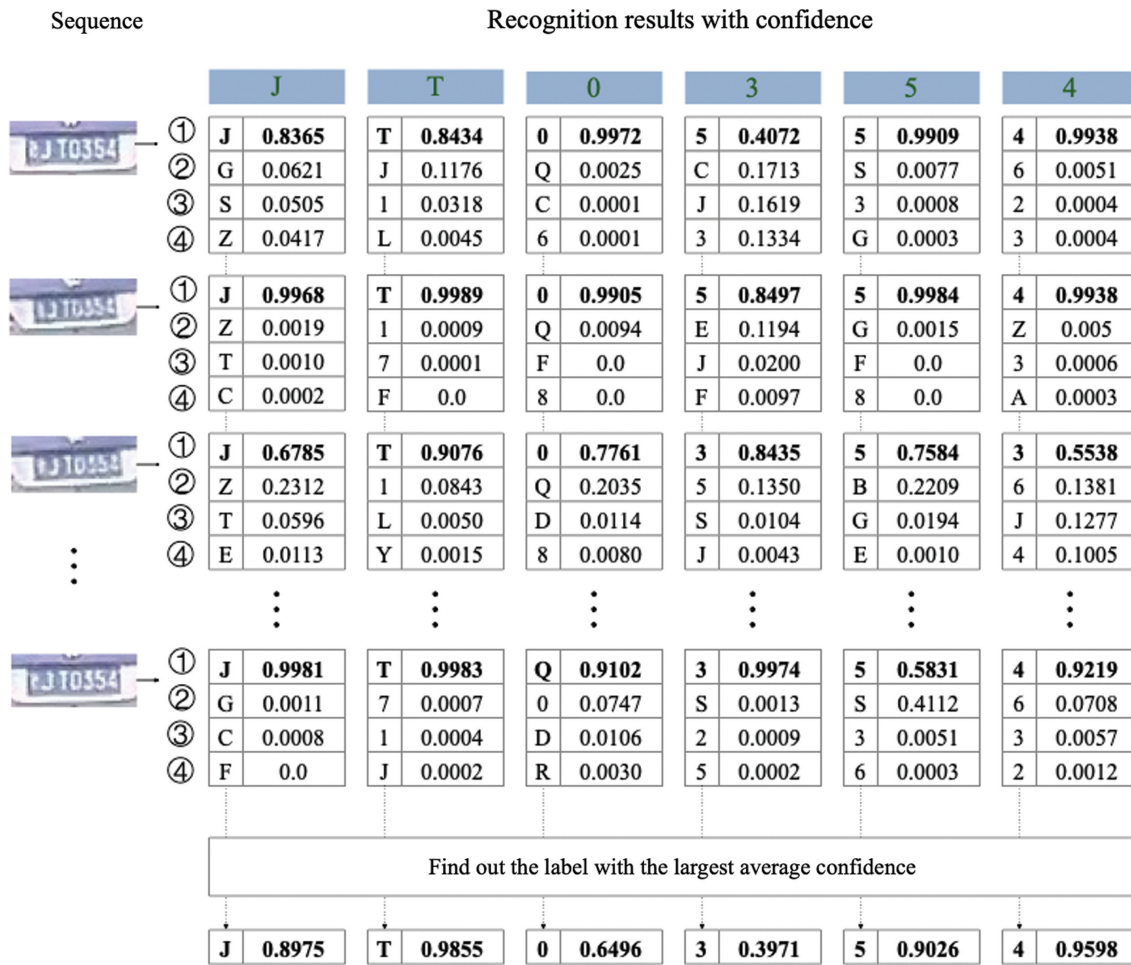


Figure 5: Integration Based on Confidence.

the maximum one as the eventual result and we can get the eventual recognition result of the first character, which is “J”. This method is used on the rest of the characters, and we can get “JT0354” in order, which is the final recognition result of this plate.

This method is efficient and it can be verified by the experiment.

4 Experiment

4.1 Datasets

XJ_ReId. In Figure 6A, this is our own dataset captured from surveillance video of real urban road in China. Based on vehicle trajectory tracking algorithm [28] and SSD, we collected 19,020 images from 1033 vehicles. For convenience, we only marked the last six characters, ignoring the Chinese character.



Figure 6: Example Images of Three Datasets: (A) XJ_ReId, (B) ReId, and (C) HDR.

Reid. Špaňhel et al. [23] collected 182,336 license plate images of different lengths, image blur, and slight occlusion on Europe real urban road. This dataset belongs to 8762 vehicles and they selected 105,924 images for training and 76,412 images for testing. Some instances are given in Figure 6B.

HDR. This dataset was also collected by Špaňhel et al. [23]. Pictures were captured by a digital single lens reflex camera with three different exposures. There are 652 images in it, which belong to 299 vehicles, and this dataset is just used for testing. These images contain rotated license plates that are of different types than the ones used for training. Figure 6C shows some samples of this dataset.

4.2 Comparative Experiment

Different LPR methods were taken as control group in the experiment.

EasyPR. EasyPR is an open source system for LPR written in C++ based on OpenCV computer vision library. It can detect and recognize Chinese license plate with high quality. In the experiment, detection phase was skipped and LPR was evaluated on cropped images.

OpenALPR. OpenALPR is an open source library for Automatic LPR written in C++. It is based on OpenCV computer vision library and Tesseract OCR and mainly used in EU license plates. Like EasyPR, the detection phase is skipped too.

SOTA. As a comparison, we reproduce a state-of-the-art paper [8]. The SOTA method consists of four modules: image acquisition, license plate extraction, segmentation and recognition. In the experiment, the segmentation and recognition part of this method is evaluated by all of the datasets above.

Holistic. Holistic is a segmentation-free method and based on the end-to-end theory to recognize license plate images with low quality [23]. This method is robust and allows an angle between the camera and the license plate.

Because there are six characters in EU and Chinese license plate (ignoring Chinese character), we use Acc_6 , Acc_5 , and Acc_4 to evaluate the recognition rate. Acc_6 represents that the recognition result matches the label perfectly. Acc_5 indicates that not less than five characters are recognized correctly. In the same way, Acc_4 means at least four correctly recognized characters.

4.3 Experimental Results and Analysis

Holistic LPR methods with distinct kinds of CNN architectures have different recognition rates and several kinds of CNN models are applied for LPR. Table 1 shows the impacts of different CNN architecture on Holistic LPR methods. Processing speed was measured on a PC with Xeon@2.6 GHz and GTX 1080.

Considering the accuracy and speed, DenseNet169 is selected as the basic CNN architecture for the Holistic LPR method. Table 2 gives the evaluation of different LPR methods on each dataset.

The number of continuous frames may have an effect on recognition rate. We separate the license plate sequences according to the number of continuous frames and calculate the Acc_6 on each number. We found

Table 1: Evaluation of Holistic LPR Methods based on Different CNN Architecture.

System	Acc ₆ /Acc ₅ /Acc ₄ (%)			Speed (ms)
	XJ_Reid	Reid	HDR	GPU
ResNet50	41.53/57.98/69.38	87.91/95.98/97.53	59.53/82.61/91.97	0.48
ResNet101	42.76/ 59.78 /70.09	88.31/96.23/98.77	60.07/ 83.93 /92.30	0.84
DenseNet169	41.60/58.11/ 70.52	88.64/96.51/97.64	59.56/83.45/92.98	0.26
DenseNet201	41.61/58.54/70.02	89.19/96.98/97.93	60.09/83.81/93.01	0.38
Inception-v3	42.85 /59.25/70.30	88.71/96.20/ 98.80	60.08/83.77/ 93.84	0.46
CliqueNet-S3	39.09/55.93/66.93	83.76/92.83/94.87	60.01/81.93/92.19	0.27

Bold values represent the highest ACC values among these methods.

Table 2: Result of a Comparative Experiment.

System	Acc ₆ /Acc ₅ /Acc ₄ (%)		
	XJ_Reid	Reid	HDR
EasyPR	41.96/50.59/66.07	–	–
OpenALPR	–	34.00/50.59/66.07	40.00/50.59/66.07
SOTA	41.13/49.17/65.72	85.33/92.24/96.52	58.14/80.26/90.87
Holistic	41.60/58.11/70.52	88.64/96.51/97.64	59.56/83.45/92.98
Ours (frequency)	42.21/59.44/73.67	89.60/96.83/97.87	60.20/84.28/93.64
Ours (confidence)	47.44/63.03/76.94	89.80/96.91/97.95	62.21/84.95/93.65

Bold values represent the highest ACC values among these methods.

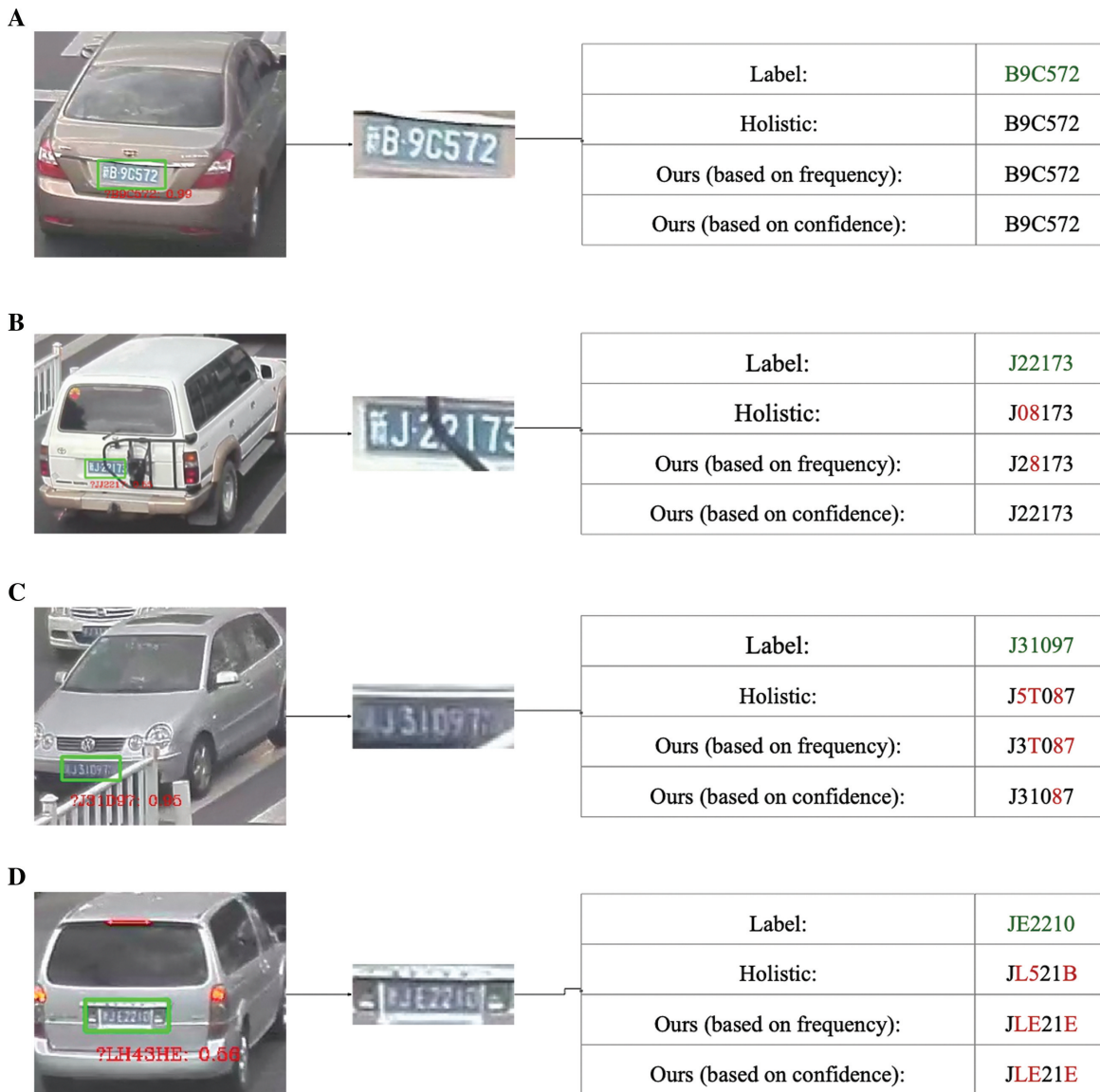


Figure 7: Comparison between Holistic and our methods on some possible situations in the real scene.

(A) The license plate is clear, (B and C) the license plate is slightly obscured or has low ambiguity, (D) the license plate has high ambiguity.

that there is little change of accuracy when enough information about the plate is given. Table 1 reveals that different kinds of CNN architecture have a negligible effect on LPR accuracy. Owing to the advantage on speed, DenseNet169 is selected as the basic CNN architecture on the Holistic method.

In Table 2, EasyPR, Holistic, and SOTA are about the same performance on XJ_ReId, which is our own dataset. Moreover, our method has a significant improvement on the same dataset. As for the two methods of result integration, the one based on confidence performs better than the one based on frequency; even the latter has a huge increase on the Holistic way. For further experiment, we take the public datasets, ReId and HDR, as comparison. The result shows that our method has better accuracy than other methods.

Through the evaluation, we can be sure that the number of continuous frames could affect the recognition rate. We acknowledge that the recognition rate will get better as the number of continuous frames increases, and the minimum number to collect the frames is 40, 30, and 30 for each dataset.

Figure 7 shows some possible situations in the real scene. All the LPR methods recognize the license plate accurately in Figure 7A. In Figure 7B and C, our method corrects the results of Holistic. Sometimes, the results of our method are not satisfactory such as the instance shown in Figure 7D. Fortunately, we do not need to worry about this situation because it appears rarely and does not make the result worse.

It can be seen from the experimental result that our method can filter out the bad case in the license plate sequence and integrate all the optimal recognition results. Under the environment of real urban road, we believe that our method could have a remarkable performance.

5 Conclusions

This paper presents an end-to-end LPR method based on result integration, which takes good use of timing-sequence information. Evaluation proves that the segmentation-free LPR method is robust to various illumination, rotation, occlusion, and image blur. Furthermore, the integration of the recognition results can exclude some wrong cases and form a more accurate result. It also shows that our method significantly outperforms existing open-source and state-of-the-art solutions on several datasets, whereas the processing speed is comparable when used on GPUs, and our solution is easy to use on GPUs.

In a word, our method is feasible and accurate on real traffic scenes, which will be well applied in surveillance videos of crossroads or urban road without electronic police.

Bibliography

- [1] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos and E. Kayafas, A license plate-recognition algorithm for intelligent transportation system applications, *IEEE Trans. Intell. Transport. Syst.* **7** (2006), 377–392.
- [2] S.-L. Chang, L.-S. Chen, Y.-C. Chung and S.-W. Chen, Automatic license plate recognition, *IEEE Trans. Intell. Transport. Syst.* **5** (2004), 42–53.
- [3] T. K. Cheang, Y. S. Chong and Y. H. Tay, Segmentation-free vehicle license plate recognition using ConvNet-RNN, arXiv preprint arXiv:1701.06439 (2017).
- [4] N. F. Gazcón, C. I. Chesñevar and S. M. Castro, Automatic vehicle identification for Argentinean license plates using intelligent template matching, *Pattern Recogn. Lett.* **33** (2012), 1066–1074.
- [5] I. Giannoukos, C.-N. Anagnostopoulos, V. Loumos and E. Kayafas, Operator context scanning to support high segmentation rates for real time license plate recognition, *Pattern Recogn.* **43** (2010), 3866–3878.
- [6] R. Girshick, Fast r-cnn, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, 2015.
- [7] R. Girshick, J. Donahue, T. Darrell and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Columbus, Ohio, 2014.
- [8] K. Indira, K. V. Mohan and T. Nikhilashwary, Automatic license plate recognition, in: *Recent Trends in Signal and Image Processing*, pp. 67–77, Springer, Singapore, 2019.
- [9] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj and K. R. Ramakrishnan, Deep automatic license plate recognition system, in: *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, p. 6, ACM, Guwahati, Assam, India, 2016.

- [10] K.-J. Kim, S.-M. Park and Y.-J. Choi, Deciding the number of color histogram bins for vehicle color recognition, in: *2008 IEEE Asia-Pacific Services Computing Conference*, pp. 134–138, IEEE, Yilan, Taiwan, 2008.
- [11] K. I. Kim, K. Jung and H. J. Kim, Fast color texture-based object detection in images: Application to license plate localization, in: *Support Vector Machines: Theory and Applications*, pp. 297–320, Springer, Berlin, Heidelberg, 2005.
- [12] H. E. Kocer and K. K. Cevik, Artificial neural networks based vehicle license plate recognition, *Procedia Comput. Sci.* **3** (2011), 1033–1037.
- [13] H. Li and C. Shen, Reading car license plates using deep convolutional neural networks and LSTMs, arXiv preprint arXiv:1601.05610 (2016).
- [14] 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*, pp. 21–37, Springer, Cham, 2016.
- [15] S. Qiao, Y. Zhu, X. Li, T. Liu and B. Zhang, Research of improving the accuracy of license plate character segmentation, in: *2010 Fifth International Conference on Frontier of Computer Science and Technology*, pp. 489–493, IEEE, Changchun, Jilin Province, China, 2010.
- [16] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You only look once: Unified, real-time object detection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, 2016.
- [17] J. Redmon and A. Farhadi, YOLO9000: Better, faster, stronger, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, Hawaii, 2017.
- [18] J. Redmon and A. Farhadi, Yolov3: an incremental improvement, arXiv preprint arXiv:1804.02767 (2018).
- [19] S. Ren, K. He, R. Girshick and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, Montreal, Canada, 2015.
- [20] G. A. Samra and F. Khalefah, Localization of license plate number using dynamic image processing techniques and genetic algorithms, *IEEE Trans. Evol. Comput.* **18** (2014), 244–257.
- [21] H. Seibel, S. Goldenstein and A. Rocha, Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos, *IEEE Access* **5** (2017), 20020–20035.
- [22] B. Shan, Vehicle license plate recognition based on text-line construction and multilevel RBF neural network, *JCP* **6** (2011), 246–253.
- [23] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík and P. Zemčík, Holistic recognition of low quality license plates by CNN using track annotated data, in: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, Lecce, Italy, 2017.
- [24] N. Thome, A. Vacavant, L. Robinault and S. Miguët, A cognitive and video-based approach for multinational license plate recognition, *Mach. Vis. Appl.* **22** (2011), 389–407.
- [25] X. Wang, Z. Man, M. You and C. Shen, Adversarial generation of training examples: Applications to moving vehicle license plate recognition, arXiv preprint arXiv:1707.03124 (2017).
- [26] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen and P. Shi, An algorithm for license plate recognition applied to intelligent transportation system, *IEEE Trans. Intell. Transport. Syst.* **12** (2011), 830–845.
- [27] M. Zahedi and S. M. Salehi, License plate recognition system based on SIFT features, *Procedia Comput. Sci.* **3** (2011), 998–1002.
- [28] Y. Zhang, J. Wang and X. Yang, Real-time vehicle detection and tracking in video based on faster R-CNN, *J. Phys. Conf. Ser.* **887** (2017), 012068.