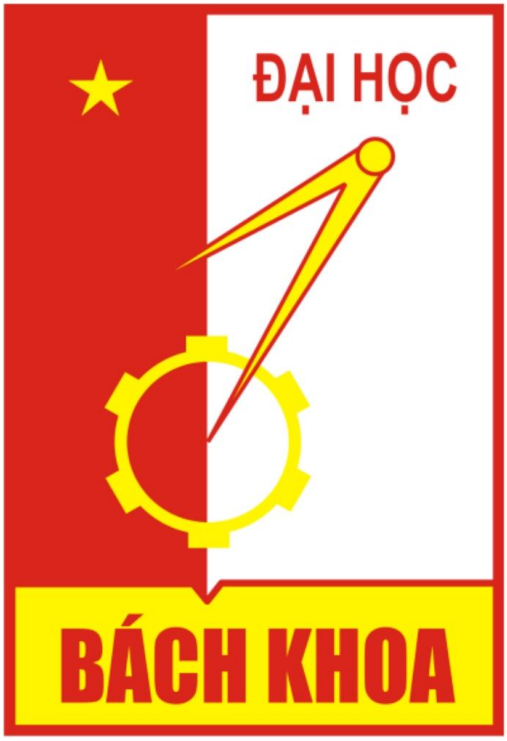
Hanoi University of Science and Technology



Project III

Final report

A System License Plate Recognition Based on the YOLO Detector

Students:
Trinh Quyet Tien – 20211262M

Table of Contents

Abstract
Automatic License Plate Recognition (ALPR) has been a frequent topic of research due to many practical applications. However, many of the current solutions are still not robust in real-world situations, commonly depending on many constraints. This paper presents a robust and efficient ALPR system based on the state-of-the-art YOLO object detector. The Convolutional Neural Networks (CNNs) are trained and finetuned for each ALPR stage so that they are robust under different conditions (e.g., variations in camera, lighting, and background). Specially for character segmentation and recognition, we design a two-stage approach employing simple data augmentation tricks such as inverted License Plates (LPs) and flipped characters. In our proposed dataset, the trial versions of commercial systems achieved recognition rates below 70%.

I. INTRODUCTION

Automatic License Plate Recognition (ALPR) has been a frequent topic of research due to many practical applications, such as automatic toll collection, traffic law enforcement, private spaces access control and road traffic monitoring. ALPR systems typically have three stages: License Plate (LP) detection, character segmentation and character recognition. The earlier stages require higher accuracy or almost perfection, since failing to detect the LP would probably lead to a failure in the next stages either. Although ALPR has been frequently addressed in the literature, many studies and solutions are still not robust enough on real-world scenarios. Many computer vision tasks have recently achieved a great increase in performance mainly due to the availability of large-scale annotated datasets (i.e., ImageNet) and the hardware (GPUs) capable of handling a large amount of data. In this scenario, Deep Learning (DL) techniques arise. However, despite the remarkable progress of DL approaches in ALPR, there is still a great demand for ALPR datasets with vehicles and LPs annotations. The amount of training data is determinant for the performance of DL techniques. The amount of training data is determinant for the performance of DL techniques. Higher amounts of data allow the use of more robust network architectures with more parameters and layers. Hence, we propose a larger benchmark dataset focused on different real-world scenarios.

To the best of our knowledge, the SSIG SegPlate Database (SSIG) is the largest public dataset of Vietnam LPs. This dataset contains less than 800 training examples and has several constraints such as: it uses a static camera mounted always in the same position, all images have very similar and relatively simple backgrounds, there are no motorcycles and only a few cases where the LPs are not well aligned.

When recording the UFPR-ALPR dataset, we sought to eliminate many of the constraints found in ALPR applications by using three different non-static cameras to capture 4,500 images from different types of vehicles (cars, motorcycles, buses, trucks, among others) with complex backgrounds and under different lighting conditions. The vehicles are in different positions and distances to the camera. Furthermore, in some cases, the vehicle is not fully visible on the image. To the best of our knowledge, there are no public datasets for ALPR with annotations of cars, motorcycles, LPs and

characters. Therefore, we can point out two main challenges in our dataset. First, usually, car and motorcycle LPs have different aspect ratios, not allowing ALPR approaches to use this constraint to filter false positives. Also car and motorcycle LPs have different layouts and positions.

As great advances in object detection were achieved through YOLO-inspired models. we decided to fine-tune it for ALPR. YOLOv5 is a state-of-the-art real-time object detection that uses a model with 19 convolutional layers and 5 max-pooling layers.

In this work, we propose a new robust real-time ALPR system based on the YOLO object detection Convolutional Neural Networks (CNNs). Since we are processing video frames, we also employ temporal redundancy such that we process each frame independently and then combine the results to create a more robust prediction for each vehicle. The proposed system outperforms previous results and two commercial systems in the SSIG dataset and also in our proposed UFPR-ALPR. The main contributions of this paper can be summarized as follows:

- A new real-time end-to-end ALPR system using the state-of-the-art YOLO object detection CNNs.
- A robust two-stage approach for character segmentation and recognition mainly due to simple data augmentation tricks for training data such as inverted LPs and flipped characters.
- A public dataset for ALPR with 8500 fully annotated images (over 60,000 LP characters) focused on usual and different real-world scenarios, showing that our proposed ALPR system yields outstanding results in both scenarios.
- A comparative evaluation among the proposed approach, previous works in the literature and two commercial systems in the UFPR-ALPR dataset.

This paper is organized as follows. We briefly review related work in Section II. The UFPR-ALPR dataset is introduced in Section III. Section IV presents the proposed ALPR system using object detection CNNs. We report and discuss the results of our experiments in Section V. Conclusions and future work are given in Section VI.

II. RELATED WORK

In this section, we briefly review several recent works that use DL approaches in the context of ALPR. For relevant studies using conventional image processing techniques, please refer to [1], [2], [13]–[19]. More specifically, we discuss works related to each ALPR stage, and specially studies works that not fit into the other subsections.

LP Detection: Many authors have addressed the LP detection stage with object detection CNNs. Montazzolli and Jung [20] used a single CNN arranged in a cascaded manner to detect both car frontal-views and its LPs, achieving high recall and precision rates. Hsu et al. [21] customized CNNs exclusively for LP detection and demonstrated that the modified versions perform better. Rafique et al. [22] applied Support Vector Machines (SVM) and Region based CNN (RCNN) for LP detection, noting that RCNNs are best suited for real-time systems.

Li and Chen [5] trained a CNN based on characters cropped from general text to perform a character-based LP detection, achieving higher recall and precision rates than previous approaches. Bulan et al. [3] first extracts a set of candidate LP regions using a weak Sparse Network of Winnows (SNoW) classifier and then filters them using a strong CNN, significantly improving the baseline method.

Character Segmentation: ALPR systems based on DL techniques usually address the character segmentation and recognition together. Montazzolli and Jung [20] propose a CNN to segment and recognize the characters within a cropped LP. They have segmented more than 99% of the characters correctly, outperforming the baseline by a large margin. Bulan et al. [3] achieved very high accuracy in LP recognition jointly performing the character segmentation and recognition using Hidden Markov Models (HMMs) where the most likely LP was determined by applying the Viterbi algorithm.

Character Recognition: Menotti et al. [23] proposed the use of random CNNs to extract features for character recognition, achieving a significantly better performance than using image pixels or learning the filters weights with back propagation. Li and Chen [5] proposed to perform the character recognition as a sequence labeling problem. A Recurrent Neural Network (RNN) with Connectionist Temporal Classification (CTC) is employed to label the sequential data, recognizing the whole LP without the character-level segmentation. Although Svoboda et al. [24] have not performed the character recognition itself, they achieved high quality LP deblurring reconstructions using a text deblurring CNN, which can be very useful in character recognition.

Miscellaneous: Masood et al. [7] presented an end-to-end ALPR system using a sequence of deep CNNs. As this is a commercial system, little information is given about the used CNNs. Li et al. [6] propose a unified CNN that can locate LPs and recognize them simultaneously in a single forward pass. In addition, the model size is highly decreased by sharing many of its convolutional features.

Final Remarks: Many papers only address part of the ALPR pipeline (e.g., LP detection) or perform their experiments on datasets that do not represent real-world scenarios, making it difficult to accurately evaluate the presented methods. In addition, most of the approaches are not capable of recognizing LPs in real-time, making it impossible for them to be applied in some applications. In this sense, we employ the YOLO object detection CNNs in each stage to create a robust and efficient end-to-end ALPR system. In addition, we perform data augmentation for character recognition, since this stage is the bottleneck in some ALPR systems.

III. THE ALPR DATASET

The dataset contains 8,000 images taken from parking lots over a variety of times and vehicles. Images obtained with different cameras do not necessarily have the same quality, although they have the same resolution and frame rate. This is due to different camera specifications, such as autofocus, bit rate, focal length and optical image stabilization. In Vietnam, the LPs have size and color variations depending on the type of the vehicle and its category. Cars' LPs have many sizes of 28cm × 20cm, 33cm ×

16.5cm, 47cm × 11cm, 520cm × 11cm while motorcycles LPs have 19cm × 14cm. Private vehicles have white LPs, while buses, taxis and other transportation vehicles have yellow, blue, red LPs.

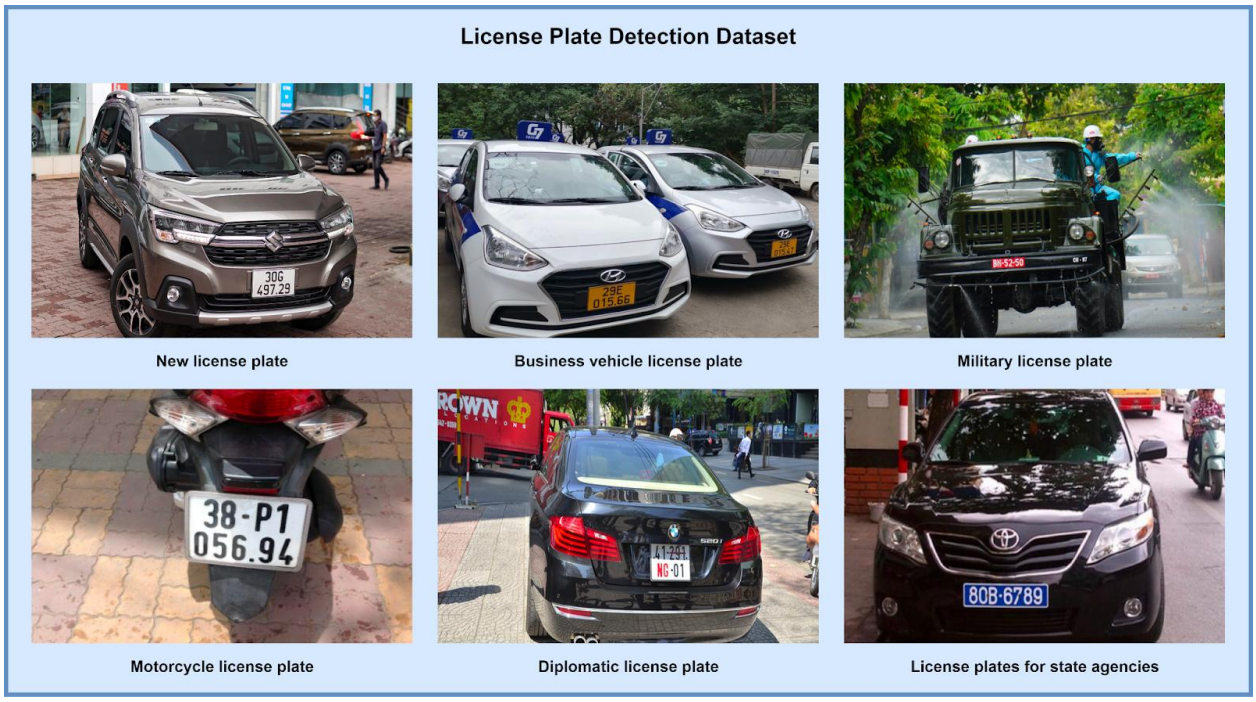


Fig. 1. License Plate Detection Dataset

The dataset is split as follows: 40% for training, 40% for testing and 20% for validation. The dataset distribution was made so that each split has the same number of images obtained with each camera, taking into account the type and position of the vehicle, the color and the characters of the vehicle's LP, the distance of the vehicle from the camera (based on the height of the LP in pixels) such that each split is as representative as possible. In Vietnam, each state uses particular starting letters for its LPs which results in a specific range.

IV. PROPOSED ALPR APPROACH

This section describes the proposed approach and it is divided into four subsections, one for each of the ALPR stages (i.e., vehicle and LP detection, character segmentation and character recognition) and one for temporal redundancy. Fig. 2 illustrates the ALPR pipeline, explained throughout this section. We use specific CNNs for each ALPR stage. Thus, we can tune the parameters separately in order to improve the performance for each task. The models used YOLOv5.

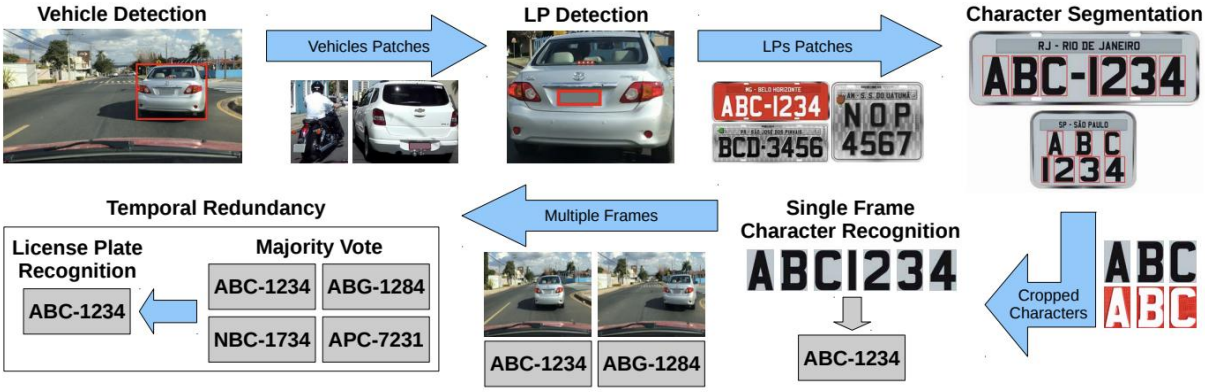


Fig. 2. An usual ALPR pipeline having temporal redundancy at the end.

A. Vehicle and LP Detection

We train two CNNs in this stage: one for vehicle detection in the input image and other for LP detection in the detected vehicle. Recent works [20], [25] also performed the vehicle detection first.

While the entire frame and the vehicle coordinates are used as inputs to train the vehicle detection CNN, the vehicle patch (with a margin) and the coordinates of its LP are used to learn the LP detection network. The size of the margin is defined as follows. We evaluated, in the validation set, the required margin so that all LPs would be completely within the bounding boxes of the vehicles found by the vehicle detection CNN. This is done to avoid losing LPs in cases where the vehicle is not very well detected/segmented.

By default, YOLO only returns objects detected with a confidence of 0.25 or higher. In the validation set, we evaluated the best threshold in order to detect all vehicles having the lowest false positive rate.

A negative recognition result is given in cases where no vehicle is found.

For LP detection we use threshold equal 0, as there might be cases where the LP is detected with very low confidence (e.g., 0.1). We keep only the detection with the largest confidence in cases where more than one LP is detected, since each vehicle has only one LP.

B. Character Segmentation

Once the LP has been detected, we employ the CNN proposed by Montazzolli and Jung [20] (CR-NET) for character segmentation and recognition. However, instead of performing both stages at the same time through an architecture with 35 classes (0-9, A-Z, where the letter O is detected jointly with the digit 0), we chose to first use a network to segment the characters and then another two to recognize them. Knowing that all Vietnam LPs have the same format: one letters and six or seven digits, we use 26 classes for letters and 10 classes for digits. As pointed out by Gonc¸alves et al. [25], this reduces the incorrect classification.

The character segmentation CNN (architecture described in Table II) is trained using the LP patch (with a margin) and the characters coordinates as inputs. As in the

previous stage, this margin is defined based on the validation set to ensure that all characters are completely within its predicted LP.

The CNN input size (416 × 416) was chosen based on the LP's ratio of Vietnam cars (1.4 × 1, 2 × 1, 4.27 × 1, 4.73 × 1), however the motorcycles LPs are nearly square (1.36 × 1). That way, we enlarged horizontally all detected LPs (to 2.75 × 1) before performing the character segmentation. We also create a negative image of each LP, thereby doubling the number of training samples. Since the color of the characters in the Vietnam LPs depends on the category of the vehicle (e.g., private or commercial), the negative images simulate characters from other categories.

In some cases, more than 7 characters might be detected. If there are no overlaps (Intersection over Union (IoU) ≥ 0.25), we discard the ones with the lowest confidence levels. Otherwise, we perform the union between the overlapping characters, turning them into a single character. As motorcycle LPs can be very tilted, we use a higher threshold (IoU ≥ 0.75) to consider the overlap between its characters.

C. Character Recognition

Since many characters might not be perfectly segmented, containing missing parts, and as each character is relatively small, even one pixel difference between the ground truth and the prediction might impair the character's recognition. Therefore, we evaluate different padding values (1-3 pixels) in the segmented characters to achieve higher recognition rates. As Fig. 3 illustrates, the more padding pixels the more noise information is added (e.g., portions of other characters or the LP frame).

Fig. 3. Comparison of different values of padding.

As previously mentioned, we use two networks for character recognition. For training these networks, the characters and their labels are passed as input. For digit recognition, we removed the first four layers of the character segmentation CNN, since in our tests the results were similar, but with a lower computational cost. However, for letter recognition (more classes and fewer examples) we still use the entire architecture of the character segmentation CNN. The use of two networks allows the tuning of network parameters (e.g., input/output size) for each task.

Fig. 4. License Plate Recognition Dataset.

Having knowledge of the specific LP country layout (e.g., the Vietnam layout), we know which characters are letters and which are digits by their position. We sort the segmented characters by their horizontal and vertical positions for cars and motorcycles, respectively. The first three or four characters correspond to the letters or digits and the last four or five to the digits, even in cases where the LP is considerably tilted. It is worth noting that a country (e.g., USA) might have different LP layouts, so this approach would not be suitable in such cases.

In addition to performing the training with the characters available in the training set, we also perform data augmentation in two ways. First, we create negative images to simulate characters from other vehicle categories (as in the character segmentation stage) and then, we also check which characters can be flipped both horizontally and vertically to create new instances. Table I shows which characters can be flipped in each direction.

Flip Direction	Characters
Vertical	0, 1, 3, 8, B, C, D, E, H, I, K, O, X
Horizontal	0, 1, 8, A, H, I, M, O, T, U, V, W, X, Y
Both	0, 1, 6(9), 8, 9(6), H, I, N, O, S, X, Z

TABLE I. The Characters that can be flipped in each direction to create new instances.

As in the LP detection step, we use confidence threshold = 0 and consider only the detection with the largest confidence. Hence, we ensure that a class is predicted for every segmented character.

D. Temporal Redundancy

After performing the LP recognition on single frames, we explore the temporal redundancy information through the union of all frames belonging to the same vehicle. Thus, the final recognition is composed of the most frequently predicted character at each LP position (majority vote). Temporal information has already been explored previously in ALPR [25], [26]. In both studies, the use of majority voting has greatly increased recognition rates.

V. CONCLUSIONS

In this paper, we have presented a robust real-time end-to-end ALPR system using the state-of-the-art YOLO object detection CNNs. We trained a network for each ALPR stage, except for the character recognition where letters and digits are recognized separately (with two distinct CNNs).

At present, the bottleneck of ALPR systems is the character segmentation and recognition stages. In this sense, we performed several approaches to increase recognition rates in both stages, such as data augmentation to simulate LPs from other vehicle's categories and to increase characters with few instances in the training set. Although simple, these strategies were essential to accomplish outstanding results.

As future work, we intend to explore new CNN architectures to further optimize (in terms of speed) vehicle and LP detection stages. We also intend to correct the alignment of inclined LPs and characters in order to improve the character segmentation and recognition. Additionally, we plan to explore the vehicle's manufacturer and model in the ALPR pipeline as our new dataset provides such information. Although our system was conceived and evaluated on two country-specific datasets from Vietnam, we believe that the proposed ALPR system is robust to locate vehicle, LPs and alphanumeric characters from any other country. In this direction, aiming a fully robust system we just need to design a character recognition module that is independent of the LP layout.

REFERENCES

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311– 325, Feb 2013.

[2] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1096–1107, April 2016.

[3] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentationand annotation-free license plate recognition with deep localization and failure identification," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 9, pp. 2351–2363, Sept 2017.

[4] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, June 2009, pp. 248–255.

[5] H. Li and C. Shen, “Reading car license plates using deep convolutional neural networks and LSTMs,” CoRR, vol. abs/1601.05610, 2016. [Online]. Available: http://arxiv.org/abs/1601.05610

[6] H. Li, P. Wang, and C. Shen, “Towards end-to-end car license plates detection and recognition with deep neural networks,” CoRR, vol. abs/1709.08828, 2017. [Online]. Available: http://arxiv.org/abs/1709. 08828

[7] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz, “License plate detection and recognition using deeply learned convolutional neural networks,” CoRR, vol. abs/1703.07330, 2017. [Online]. Available: http://arxiv.org/abs/1703.07330

[8] G. R. Gonc¸alves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, “Benchmark for license plate character segmentation,” Journal of Electronic Imaging, vol. 25, no. 5, pp. 053 034–053 034, 2016.

[9] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, “Spatially supervised recurrent convolutional neural networks for visual object tracking,” in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), May 2017, pp. 1–4.

[10] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, “SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), July 2017, pp. 446–454.

[11] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 6517–6525.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 779– 788.

[13] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, “License plate recognition from still images and video sequences: A survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 377–391, Sept 2008.

[14] G. S. Hsu, J. C. Chen, and Y. Z. Chung, “Application-oriented license plate recognition,” IEEE Transactions on Vehicular Technology, vol. 62, no. 2, pp. 552–561, Feb 2013.

[15] A. H. Ashtari, M. J. Nordin, and M. Fathy, “An iranian license plate recognition system based on color features,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 4, pp. 1690–1705, Aug 2014.

[16] M. S. Sarfraz, A. Shahzad, M. A. Elahi, M. Fraz, I. Zafar, and E. A. Edirisinghe, “Real-time automatic license plate recognition for CCTV forensic applications,” Journal of Real-Time Image Processing, vol. 8, no. 3, pp. 285–295, Sep 2013.

[17] R. Panahi and I. Gholampour, “Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 767–779, April 2017.

[18] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, “A robust and efficient approach to license plate detection,” IEEE Transactions on Image Processing, vol. 26, no. 3, pp. 1102–1114, March 2017.

[19] S. Azam and M. M. Islam, “Automatic license plate detection in hazardous condition,” Journal of Visual Communication and Image Representation, vol. 36, pp. 172 – 186, 2016.

[20] S. Montazzolli and C. R. Jung, “Real-time brazilian license plate detection and recognition using deep convolutional neural networks,” in 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images, Oct 2017, pp. 55–62.

[21] G. S. Hsu, A. Ambikapathi, S. L. Chung, and C. P. Su, “Robust license plate detection in the wild,” in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Aug 2017, pp. 1–6.

[22] M. A. Rafique, W. Pedrycz, and M. Jeon, “Vehicle license plate detection using region-based convolutional neural networks,” Soft Computing, Jun 2017.

[23] D. Menotti, G. Chiachia, A. X. Falcao, and V. J. O. Neto, “Vehicle ˜ license plate recognition with random convolutional networks,” in 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, Aug 2014, pp. 298–303.

[24] P. Svoboda, M. Hradis, L. Mar ˇ sˇ´ık, and P. Zemc´ık, “CNN for license plate motion deblurring,” in 2016 IEEE International Conference on Image Processing (ICIP), Sept 2016, pp. 3832–3836.

[25] G. R. Gonc¸alves, D. Menotti, and W. R. Schwartz, “License plate recognition based on temporal redundancy,” in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Nov 2016, pp. 2577–2582.

[26] M. Donoser, C. Arth, and H. Bischof, “Detecting, tracking and recognizing license plates,” in Computer Vision – ACCV 2007, Y. Yagi, S. B. Kang, I. S. Kweon, and H. Zha, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 447–456.

[27] J. Redmon, “Darknet: Open source neural networks in C,” http://pjreddie. com/darknet/, 2013–2016.

[28] OpenALPR Cloud API, http://www.openalpr.com/cloud-api.html.

[29] L. Yang, P. Luo, C. C. Loy, and X. Tang, “A large-scale car dataset for fine-grained categorization and verification,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 3973–3981.

[30] L. Dlagnekov and S. J. Belongie, Recognizing cars. Department of Computer Science and Engineering, University of California, San Diego, 2005.