License Plate Detection and Recognition using CRAFT and LSTM

Conference Paper · September 2022

DOI: 10.1007/978-3-031-29104-3_31

3 authors, including:

Le Duy Tan
Ho Chi Minh City International University
27 PUBLICATIONS **86** CITATIONS

SEE PROFILE

License Plate Detection and Recognition using CRAFT and LSTM

Anh Kiet Huynh, Tan Duy Le*, and Kha-Tu Huynh*

School of Computer Science and Engineering, International University, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
kiethuynhnbk2000@gmail.com, {ldtan, hktu}@hcmiu.edu.vn

Abstract. This work proposes a solution for developing a license plate detection and recognition system (LPDR). In the first stage, the poly region of the license plate's word line(s) can be detected by Character-Region Awareness For Text Detection (CRAFT). Specifically, the text line of the one-line license plate and two lines of the multi-line license plate can be detected effectively. Secondly, each region proposed as a plate number region by CRAFT will be passed to Mobilenet architecture to extract features. Finally, these features will be fed to Bi long short-term memory (Bi-LSTM) architecture with Connectionist Temporal Classification to predict output text in each input region. By applying this solution, the problem of multi-line license plates can be appropriately handled.

Keywords: CRAFT · License Plate Detection · License Plate Recognition · deep neural network

1 Introduction

Along with the development of modern cities, the License Plate Detection and Recognition system (LPDR) has many practical applications such as parking lots, toll collection or traffic enforcement, etc. Most of the current LDPR systems concentrate on one-line license plate problems. However, in many countries, for instance, Vietnam and Brazil, multi-line license plate accounts for a significant part.

To deal with the problem of license plate recognition, the well-known solutions are applying a segmentation algorithm to locate each character in the plate and classify each of them to obtain sequence output [1]. Unfortunately, the recognition result is critically dependent on segmentation output with classification characters result, which can be easily affected by skew, light, contrast condition of environment or binary process. Non-segmentation approach is another common approach for license plate recognition [2]. However, it works well on the one-plate license plate and has high latency for the license plate detection phase.

*Corresponding author.

For the multi-line license plate, reorganization feature maps extracted from Convolutional Neural Network (CNN) can be applied to recognize two-plate license plates [3]. Unfortunately, it requires a license plate should be an ideal image.

In our proposed LPDR system, Character-Region Awareness For Text Detection (CRAFT) [4] was applied to deal with the issue of the multi-line license plate. In detail, this method can detect the two-line of a license plate separately. Then, each line will be fed to the recognition phase. These lines can be treated as a sequence of characters and predicted by Connectionist Temporal Classification. Therefore, the output does not rely on the segmentation process and does not require an ideal license plate.

The main contributions of our work are:

- We propose a solution that uses CRAFT as an approach for text(s) detection in the license plate. License plate text region can be extracted by a combination of text score and affinity score generated by the CRAFT model. Each detected text line of the license plate will be treated as a sequence-like feature map. By applying a text recognition framework, the number plate from each detected number plate line can be extracted precisely without the segmentation process. By dividing the text lines of two-line license plates, it decreases workload compared with the recognition of two-line license plates directly.
- Build a lightweight model with a pretraining manner for the recognition phase, which has a trade-off between accuracy and latency at an acceptable level and is possible to apply to a practical system.

2 Background and Related Research

Every country sets different regulations for license plates. Therefore, depending on the information displayed in the license plates, there are different identification methods. Recently, many research related to the license plates of the countries have been published, such as Spanish and Indian [5], Bangladesh [6], and China [7], etc. The general scheme of license plates recognition is built by 02 steps of detection and recognition. The development of CNN has supported researchers to build effective image detection and recognition algorithms, especially vehicle license plates. YOLO4, with its ability to detect objects, has been used by many authors in the detection of letters and numbers [8], in the license plates. In addition, some methods using Fractal Series Expansion, DELP-DAR, ResNet [9] also give reliable results.

3 System Design

Our proposed LPDR system is divided into two phases, including detection and recognition phases, which are illustrated in Figure 1.

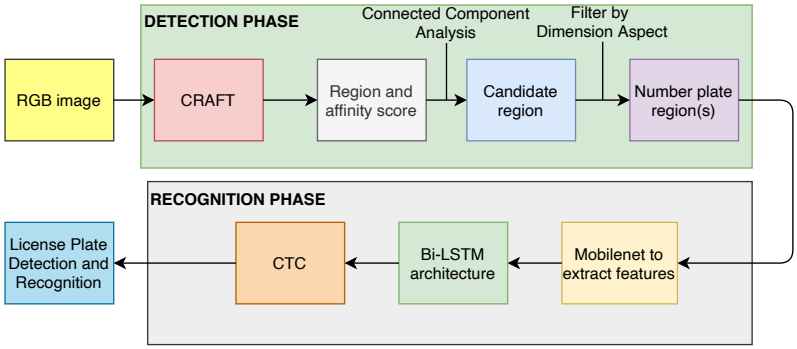


Fig. 1: Overview of the License Plate Detection and Recognition System (LPDR)

3.1 Number plate detection

The first phase of the LDPR system is license plate detection. In this phase, we use CRAFT approach to detect each text line of the license plate.

CRAFT is designed based on convolutional neural network architecture. Each input image will be resized to a fixed shape and fed to the model. This model generates a character region score and an affinity score. The region score is used to locate each character in the image, and the affinity score is used for grouping characters in the word group. With the image dataset does not have a character annotations level, we will apply a weakly-supervised learning framework for estimating character regions of each word box ground truth in the image.

CRAFT backbone: A combination of VGG16 [10] and Unet [11] is used as the backbone of CRAFT model. This backbone has skip connections in decoding to catch low-level features. The final output has 16 channels, each channel has width = (width of the input image) div 2 and height = (height of the input image) div 2. The predicted region score and affinity score are first and second channel respectively.

Ground Truth Label Generation: For each training image and its character annotation ground truth, we can generate a ground truth label for the region, and affinity score. Region score illustrates the probability that the pixel is center of character and affinity score represents the probability given pixel is center of the region between two adjacent characters.

We encode the probability of the character center with a Gaussian heatmap. This heatmap representation has been used in other applications, such as in object detection and tracking [12] due to its high flexibility when dealing with ground truth regions that are not rigidly bounded. We use the heatmap representation for learning both the region score and the affinity score.

The authors of CRAFT used the following steps to approximate and generate the ground truth for both the region score and the affinity score:

- prepare a 2-dimensional isotropic Gaussian map;
- compute perspective transform between the Gaussian map region and each character box;
- warp Gaussian map to the box area.

For the ground truths of the affinity score, the affinity boxes are defined using adjacent character boxes. By drawing diagonal lines to connect opposite corners of each character box, we can generate two triangles which we will refer to as the upper and lower character triangles. Then, for each adjacent character box pair, an affinity box is generated by setting the centers of the upper and lower triangles as corners of the box.

Weakly-Supervised Learning: In many license plate datasets, the number of datasets that have character-level annotations is not significant. With CRAFT, the ground truth region and affinity score should be generated from bounding boxes of characters in the input image. Authors of CRAFT propose a method for generating pseudo character bounding boxes ground truth for datasets that do not have character annotations.

In that, with the dataset, having world bounding box ground truth, the region of the word will be cropped from the original image. The cropped region will be passed to the interim model to predict region and affinity score. With output region and affinity score, the watershed algorithm is applied to split the region of characters in the word box. From that, we can obtain the character bounding boxes in each word box ground truth and convert their coordinates to original coordinates in the image.

CRAFT's authors also propose a method to evaluate the output of the interim model. That is based on the number of characters in the word box ground truth and number of bounding boxes generated from the output region, and the affinity score generated from the model. Let R(w) and l(w) be the word bounding box region and length of the word sample respectively. l^c(w) is the number of characters bounding box obtained from the predicted region and affinity score of the word region. So, the confidence score can be calculated with the formula:

$$s_{conf}{}^{(w)} = \frac{l(w) - min(l(w), |l(w) - l^c(w)|)}{l(w)} \tag{1}$$

And the pixel-wise confidence map of image will be set as :

$$s_c{}^{(p)} = \begin{cases} s_{conf}{}^{(w)}, & \text{if } p \in R(w) \\ 1, & \text{otherwise} \end{cases}$$

Inference: With the output predicted region score and affinity score, the first map has a shape equal to the shape of the region score map and all pixels are set to 0. The pixel will be set to 1 if its region score > Ts or its affinity score > Ta, where Ts and Ta are the threshold value of the region score and affinity score, respectively. By applying the connected component analysis(CCA) algorithm to

the output binary map, we can obtain the predicted word box for the input image by minAreaRect function in OpenCV library.

Apply CRAFT for detecting plate numbers: Among public license plate datasets, the number of datasets that have character annotations is very limited. In the detection phase, we use UFPR dataset [13], which has character-level annotations. With the dataset not having character annotations, we apply the weakly-supervised learning mentioned above for generating region and affinity scores from an interim model. In the dataset that just have license plate coordinates, we consider license plate as word box region in the image. With a multi-line plate, we divide the height of the license plate region by 2 to obtain regions of two lines of text. We apply this method for generating two ground-truth maps for AOLP [14], RodoSol [15], and synthesis datasets.

3.2 Number plate recognition

In this phase, the plate number will be extracted from each candidate region passed from the stage of detection. We treat the recognition number plate as a sequencing labeling problem. By applying Bi-long short-term memory with CTC, we can extract the number plate from the candidate region properly. In this stage, we divide the work into three sub-tasks: feature extraction, sequence labeling, and sequence decoding.

Feature extraction stage: In this stage, a CNN model will feed input image(s). In this phase, we choose Mobilenet [16], which bases on a streamlined architecture that uses depth-wise separable convolutions to build lightweight deep neural networks with pretraining weight 'imagenet', as the backbone architecture of CNN.

Each candidate region will be reshaped to 224 x 224 x 3 (use padding zero for keeping the ratio of dimensions for an input image) and feed-forward to Mobilenet to extract input image features. The features of the input image will be extracted, converted to feature sequences, and passed to long short-term memory to learn these sequence features.

Sequence labeling: The outputs of the feature extraction step will be reshaped to the sequence of features. To overcome the gradient vanishing or exploding and lack of context information, Long short-term memory (LSTM) is applied to generate a better sequence of time frame features since it contains memory blocks that can store context features for a long time.

With the text recognition problem, the context information from two directions is better than the one from only one direction. Each Bi-LSTM has two hidden layers, the first one collects feature sequence in a forward way, while the second one handles feature sequence in a backward way. The outputs of Bi-LSTM are provided information in both directions from two hidden layers.

At each time step, $h^t(f)$ is defined as the output of the LSTM at time step t in a forward way, and $h^t(b)$ is defined as the output of LSTM at time step t in the backward direction, and x_t is feature at time step t. Soft-max activation function is applied for transforming LSTM states to the probability distribution of the number of available characters in the dataset plus space. The formula is represented below:

$$p_t(c = c_k|x_t) = Softmax(h^t(f), h^t(b)) \tag{2}$$

k in 2 = 1,2,...n,where n is the number of characters available in dataset + 1.

Sequence decoding: Sequence decoding is the last step of number plate recognition, the input of this step is the sequence of probability estimation from BiLSTM. From that output, we can extract the sequence of characters in the input image. Connectionist temporal classification (CTC) is used for sequence classification without character segmentation, it allows for predicting non-fix length sequences with fixed length input. The objective function of CTC is shown below and it also is the objective function of the recognition model.

$$O = -\sum_{(c,z)\in S} \ln P(z|c) \tag{3}$$

Where S is the training set in 3, it is comprised of pairs of input sequence labels (They are the outputs of Bi-Lstm) and target sequences (c and z). $P(z|c)$ is the conditional probability of output prediction equals to target sequence z when input is c. We need to find parameters to maximize the value of $P(z|c)$ in 3. In detail, the formula of $P(z|c)$ is calculated as:

$$P(z|c) = \sum_{\pi:B(\pi)=z} P(\pi|c) \tag{4}$$

B in 4 is the operation of elimination of the repeated label and space in the same group to obtain the final result. For example B(a-1aa-c) = B(acc-1bb-c) = a1c

4 Implementation and Results

4.1 Dataset

In this paper, we use:

- AOLP dataset, which contains 2049 Taiwan vehicle images and is divided into three subsets: access control(AC), road patrol(RP), and traffic law enforcement (LE).
- UFPR dataset, which contains 4500 images of Brazilian vehicles in real-world scenarios. Each image has character annotations for the number plate as well as the position of the vehicle and license plate in the image.

Table 1: **Result of CRAFT for detecting number plates in AOLP subsets**

Method	AC-precision	AC-recall	RP-precision	RP-recall	LE-precision	LE-recall
Hsu at el. [14]	91%	96%	91%	95%	91%	94%
Our LPDR system	95.17%	99.69%	97.69%	91.37%	88.43%	99.6%

Table 2: **Result of CRAFT for detecting number plates in RodoSol subsets**

RodoSol subset	Precision	Recall
cars-br	99.25%	99.65%
cars-me	94.63%	99.47%

- RodoSol-ALPR dataset, which contains 20000 images captured by static cameras located at pay tolls. There are images of different types of vehicles (e.g., cars, motorcycles, buses, and trucks), captured during the day and night, from distinct lanes, on clear and rainy days, and the distance from the vehicle to the camera varies slightly.
- Synthesis dataset: We collected images of vehicles and their license plates in many countries such as Vietnam, etc.

In the detection phase, we use images in AC, RP subsets of AOLP, UFPR-training, validation subset, RodoSol-ALPR training-validation, and synthesis images for training and validation for CRAFT. We use LE subset of AOLP dataset, RodoSol-ALPR testing section for testing CRAFT model.

In the recognition phase, we use candidate regions generated by CRAFT in the detection phase that have an IOU score between themselves and the ground-truth bounding box greater than a specified score to pass to the recognition phase.

4.2 Evaluation criterion

In the detection phase, we use precision and recall as standards for the evaluation detection model. A true positive is a candidate region with the IOU (intersection over union) between this region and the ground-truth bounding box should be greater than 0.5.

During the recognition phase, we evaluate the number of license plates that are correctly recognized. A correctly recognized sample is extracted from the original image and all characters of this license plate are recognized exactly. The training samples of the recognition phase are obtained from the detection result.

4.3 Result of the number plate detection phase

The detection result for the AOLP dataset are shown in Table 1, while Table 2 shows the detection result of car types in the RodoSol-ALPR dataset. With

Table 3: **Result of recognition for number plates in AOLP subsets**

Method	AC	RP	LE
Hsu at el. [14]	88.5%	85.7%	86.6%
Our LPDR system	97.8%	86.41%	89.9%

Table 4: **Result of recognition model in RodoSol-ALPR subsets**

RodoSol-ALPR subset	Accuracy
cars-br	95.95%
cars-me	97.8%
motors-br	91.95%
motors-me	93.65%

the evaluation mentioned above, we can evaluate CRAFT detection model by precision and recall. In AOLP dataset, we choose the LE subset for testing CRAFT since it has the most complex environment among the three subsets of AOLP. In RodoSol-ALPR dataset, we use the testing section for testing purposes.

4.4 Result of the number plate recognition phase

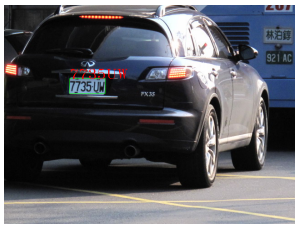
Table 3 shows the recognition results of three subsets in the AOLP dataset while the recognition results of four sections in the RodoSol-ALPR dataset are shown in Table 4. In the recognition phase, we use AOLP-AC, RP subset, full UFPR dataset and training, and validation section of RodoSol dataset for training and validation model. AOLP-LE and the testing section of the RodoSol-ALPR dataset are used for testing. With motorcycles number plates in the RodoSol-ALPR dataset, it seems to be a correct recognition if the number plates of two lines are properly recognized. For illustration purposes, we use the images below to represent the final result of our approach. Figure 2 shows the final result of three subsets of AOLP dataset, and Figure 3 presents the result of four sections of RodoSol-ALPR dataset.

5 Discussion and Conclusion

We present a solution for constructing a license plate detection and recognition system in this study (LPDRS). Firstly, CRAFT can detect the poly region of the license plate's word lines in the first stage. In detail, the text line of a one-line license plate and two lines of a multi-line license plate could be detected efficiently. Secondly, each region proposed by CRAFT as a plate number region will be subjected to the Mobilenet architecture to extract features. Finally, these characteristics will be fed into a Bi long short-term memory (Bi-LSTM) architecture with CTC, anticipating output text for each input region. The problem of a multi-line license plate can be solved with this technique.

(a) AOLP AC sample with an average inference time is 0.4s

(b) AOLP RP sample with an average inference time is 0.37s



(c) AOLP LE one plate sample with an average inference time is 0.36s

(d) AOLP LE two plates sample with an average inference time is 0.36s

Fig. 2: Recognition results of three subsets of AOLP dataset

(a) RodoSol cars-br subset with inference time=0.39s

(b) RodoSol cars-me subset with inference time=0.39s

(c) RodoSol motorcycles-be subset with inference time=0.38s

(d) RodoSol motorcycles-me subset with inference time=0.38s

Fig. 3: Recognition results of four RodoSol subsets

For future works, an end-to-end framework applying CRAFT as a detection core component for an LDPR system shall be developed. By integrating with the Internet of Things (IoT), our approach can be used in practice and meet real-time standards.

References

1. C. Patel, D. Shah, and A. Patel, "Automatic number plate recognition system (anpr): A survey," *International Journal of Computer Applications*, vol. 69, no. 9, 2013.
2. H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.
3. Y. Cao, H. Fu, and H. Ma, "An end-to-end neural network for multi-line license plate recognition," in *2018 24th international conference on pattern recognition (ICPR)*. IEEE, 2018, pp. 3698–3703.
4. Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9365–9374.
5. A. Menon and B. Omman, "Detection and recognition of multiple license plate from still images," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*. IEEE, 2018, pp. 1–5.
6. N. Saif, N. Ahmmed, S. Pasha, M. S. K. Shahrin, M. M. Hasan, S. Islam, and A. S. M. M. Jameel, "Automatic license plate recognition system for bangla license plates using convolutional neural network," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 925–930.
7. C. Xu, H. Zhang, W. Wang, and J. Qiu, "License plate recognition system based on deep learning," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. IEEE, 2020, pp. 1300–1303.
8. J.-Y. Sung and S.-B. Yu, "Real-time automatic license plate recognition system using yolov4," in *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2020, pp. 1–3.
9. K. D. Rusakov, "Automatic modular license plate recognition system using fast convolutional neural networks," in *2020 13th International Conference" Management of large-scale system development"(MLSD)*. IEEE, 2020, pp. 1–4.
10. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
11. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
12. F. Amherd and E. Rodriguez, "Heatmap-based object detection and tracking with a fully convolutional neural network," *arXiv preprint arXiv:2101.03541*, 2021.
13. 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 *International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–10.
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, 2012.
15. R. Laroca, E. V. Cardoso, D. R. Lucio, V. Estevam, and D. Menotti, "On the cross-dataset generalization in license plate recognition," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Feb 2022, pp. 166–178.
16. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.