



A two-stage deep neural network for multi-norm license plate detection and recognition

Yousri Kessentini^{a,b,c,*}, Mohamed Dhia Besbes^a, Sourour Ammar^{a,b}, Achraf Chabbouh^{a,d}

^a Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, 3021 Sfax, Tunisia

^b MIRACL Laboratory, University of Sfax, Sfax, Tunisia

^c LITIS EA 4108, University of Rouen Saint-Etienne du Rouvray, France

^d Institut Supérieur des Etudes Technologiques de Sidi Bouzid, Tunisia



ARTICLE INFO

Article history:

Received 2 March 2019

Revised 26 May 2019

Accepted 16 June 2019

Available online 18 June 2019

Keywords:

License plate detection and recognition

Semi-automatic annotation

Convolutional neural networks

Recurrent neural networks

YOLO

Deep learning

ABSTRACT

In this work, we tackle the problem of multi-norm and multilingual license plate (LP) detection and recognition in natural scene images. The system architecture use a pipeline with two deep learning stages. The first network was trained to detect license plates on the full raw image by using the latest state-of-the-art deep learning based detector namely YOLOv2. The second stage is then applied on the cropped image to recognize captured license plate photographs. Two recognition engines are compared in this work: a segmentation-free approach based on a convolutional recurrent neural network where the recognition is carried out over the entire LP image without any prior segmentation and a joint detection/recognition approach that performs the recognition on the plate component level. We also introduced a new large-scale dataset for automatic LP recognition that includes 9.175 fully annotated images. In order to reduce the time and cost of annotation processing, we propose a new semi-automatic annotation procedure of LP images with labeled components bounding box. The proposed system is evaluated using two datasets collected from real road surveillance and parking access control environments. We show that the system using two YOLO stages performs better in the context of multi-norm and multilingual license plate. Additional experiments are conducted on the public AOLP dataset and show that the proposed approach outperforms over other existing state-of-the-art methods.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The use of vehicles in our current live is increasing exponentially due to the rapid economic development. Automatically identifying vehicles through their license plates proves to be a solution of a significant role in this active world. Many related applications are proposed in this context including automatic toll collection system (Suryatali & Dharmadhikari, 2015), car parking access control (Sen et al., 2014) and real road traffic surveillance (Naimi, Kessentini, & Hammami, 2016). The Automatic License Plate Recognition (ALPR) is a field of research that gained a lot of interest during the last decade with many applications in Intelligent Transportation and Surveillance systems along with the improvement of digital camera and the increase in computational capacity. These sys-

tems aim to identify vehicles through their license plates. They provide automatic detection and recognition of vehicles license plates within real view camera scene. Once the camera takes an image of the front of the vehicle, this image is given as an input to processing algorithms to analyze it, locate and extract the plate regions from the background. The recognition phase consists on the segmentation of the characters within the detected region and their identification. The detection (Al-Shemarry, Li, & Abdulla, 2018; Arth, Limberger, & Bischof, 2007; Hsieh, Yu, & Chen, 2002; Yuan et al., 2017) and recognition (Cheang, Chong, & Tay, 2017; Du, Ibrahim, Shehata, & Badawy, 2013; Gou, Wang, Yao, & Li, 2016; Li, Wang, You, & Shen, 2018; Sedighi & Vafadust, 2011) of license plates are two separate processes, so the research on these two processes has always been performed separately. Most of the existing solutions work for a specific type of license plate template (size, text and background color, layout, etc.) or in specific conditions (camera movement, camera angle, lighting, occlusion, etc.).

Traditional ALPR systems using machine learning techniques employ hand-crafted features to represent the underlying features of the image license plate. Such techniques capture certain

* Corresponding author at: Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, 3021 Sfax, Tunisia.

E-mail addresses: yousri.kessentini@litislab.eu (Y. Kessentini), meddhiabesbes@gmail.com (M.D. Besbes), sourour.ammar@crns.rnrt.tn (S. Ammar), chabbouhachraf.ac@gmail.com (A. Chabbouh).

morphological attributes (color, text, etc.) and they are sensitive to image noise and complex background. Deep learning techniques provide an alternative to automatically select the features from the image by learning representations of the underlying data using modified filters. The strongest deep learning methods involve Convolutional Neural Networks (CNNs). These techniques have demonstrated impressive performance and success on many fields of research within computer vision, such as handwriting recognition (LeCun et al., 1989), text recognition (Wang, Wu, Coates, & Ng, 2012), visual object detection and recognition (Ciresan, Meier, Masci, Gambardella, & Schmidhuber, 2011), etc.

Since localization of license plates can be considered as a detection problem, different regional CNN methods that were recently developed for fast and precise object detection can be applied such as Faster-RCNN (Ren, He, Girshick, & Sun, 2017), YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016), YOLOv2 (Redmon & Farhadi, 2016) and SSD (Liu et al., 2015). In the context of license plate recognition, existing deep learning based works can be divided into two categories : segmentation based methods that perform first a segmentation step to separate characters and then recognize each of them separately, and segmentation-free methods that perform recognition of all text without separating the characters by using specific architectures such as recurrent neural network (RNN) like Bidirectional Long Short-Term Memory networks (BLSTM).

In this work, we propose an end-to-end deep learning License Plate recognition system for Tunisian multi-norm license plates based on state-of-the-art CNN architectures. The proposed method solves the ALPR task using a two-stage training deep neural networks (DNN). In the first stage, the state-of-the-art YOLO object detection CNN is used to detect the license plate in the image. Then we crop the image around the license plate and send it to the second stage. In the second stage, two recognition engines are compared. The first one is based on a segmentation-free recognition approach based on convolutional recurrent neural network (CRNN) where the recognition is carried out on the whole plate image. The second strategy is based on a joint detection/recognition approach where the recognition of the plate components is performed as part of the detection. The prediction of the full license plate number is based on the detected LP components order. We evaluated the proposed system on two datasets collected from real road surveillance and parking access control environments.

The main contributions of this paper can be summarized as follows:

- A real-time end-to-end ALPR system is proposed using a two-stage approach for multi-norm LP detection and recognition. The detection system is based on the state-of-the-art YOLOv2 object detection CNN. For the recognition stage, two LP recognition engines are compared based on a segmentation-free recognition approach using CRNN and a segmentation based detection/recognition approach using YOLOv2 object detection CNNs. We demonstrate that the second recognition approach performs better in the case of multi-norm Tunisian LP¹.
- In order to reduce the time and cost of annotation processing, we propose a semi-automatic annotation process of LP images to complete partial annotations with location boxes. To the best of our knowledge, the proposed semi-automatic process is completely original.
- A new public multi-norm LP images dataset named GAP-LP² with 9175 fully annotated images (over than 80,000 LP components)

is provided freely for research purpose. To the best of our knowledge, GAP-LP is the largest-ever LP dataset available to the academic research community, making it suitable for trying and evaluating deep learning techniques.

- We experimentally evaluate the benefits of data augmentation by generating synthetic LP examples in order to augment the training dataset and to reduce the variance of a classifier, thereby reducing error.

The rest of the paper is organized as follows. Section 2 reviews some related works on License Plate Detection and Recognition. Section 3 describes the specificity of Tunisian License Plate. In Section 4, we present the proposed ALPR system including the detection and the recognition stages. We report and discuss the results of our experiments in Section 5. Conclusions and future work are given in Section 6.

2. Related work

In this section, we present an introduction about previous works on license plate detection and recognition. We briefly review several recent works based on deep learning in the context of ALPR.

2.1. License plate detection

The detection step consists in localizing the bounding box of the license plate from the full raw input image. The output of this step greatly influences the accuracy of the recognition step. Plenty of license plate detection algorithms have been proposed. Most of them used traditional machine learning techniques with hand-crafted features based on specific descriptors such as edge, color and texture descriptor (Anagnostopoulos, Anagnostopoulos, Loumos, & Kayafas, 2006; Du et al., 2013; Zhou, Li, Lu, & Tian, 2012). The license plate detection can be considered as a detection of a homogeneous text zone by detecting the characters directly from the image (Nguyen, Ardabilian, & Chen, 2009). Although simple and fast, this kind of approaches leads to a low detection accuracy because the features learned from the characters may not be enough to confidently find all plate characters in the image. In addition, other characters may exist in the image which can be confusing for the license plate detection. Other approaches consider the license plate as a region with rich contrast and high edge density (Wu, Chen, Wu, & Shen, 2006), or as a region containing high density of key points detected with scale invariant feature transform (SIFT) descriptor (Ng, Tay, Liang, Mokayed, & Hon, 2015). Recently, deep learning based approaches have been proposed for localizing license plates. Li et al. (2018) use a 4-layer CNN to detect the presence of text zones in the input image. Then a second 4-layer plate/non-plate CNN classifier is used to distinguish license plates from general text. Silva and Jung (2017) uses a classifier based on the FAST-YOLO network architecture to detect the frontal view of the car from the input image, and then extracts the LP from the detected frontal view image. Naimi et al. (2016) propose a LP detection method based on Faster-RCNN network which assure both high accuracy and low time cost. Masood, Shu, Dehghan, and Ortiz (2017) introduces a pipeline architecture based on a sequence of deep CNNs for LP detection under different conditions (variations in pose, lighting, occlusion, etc.) and working across a variety of license plate templates (sizes, backgrounds, fonts, etc.). Nevertheless, their architecture is not adapted to multilingual LP and especially LPs containing Arabic text.

2.2. License plate recognition

In the literature, we distinguish two kinds of approaches for LP recognition: the segmentation-based approaches and the

¹ See Section 3 for the specification of Tunisian LP.

² GAP-LP is freely available to research community at <https://sites.google.com/site/matdbparking>.

segmentation-free approaches. Du et al. (2013) presents a complete review of traditional approaches for license plate recognition. We focus more in this part on works using deep learning techniques.

2.2.1. Segmentation-based approaches

Segmentation based approaches extract each character in the license plate firstly. Then an optical character recognition (OCR) algorithm is performed to recognize each of them. Existing works on license plate segmentation can be divided into two main categories: projection-based and connected component-based. Projection-based approaches exploit the fact that characters and background have obviously different colors in a license plate giving opposite values in the binary image. Histograms of vertical and horizontal pixel projections can then be exploited for character segmentation (Guo & Liu, 2008; Qiao, Zhu, Li, Liu, & Zhang, 2010). Such approaches can be easily affected by the rotation of license plate. Connected Component-based approaches (Chang, Chen, Chung, & Chen, 2004; G. Casey & Lecolinet, 1996; Giannoukos, Anagnostopoulos, Loumos, & Kayafas, 2010; Jiao, Ye, & Huang, 2009) perform segmentation by labeling all connected pixels in the binary image into components. This type of method is robust to rotation but it fails to segment correctly characters when they are joined together or broken. Since the characters are segmented, a recognition step can be performed as a kind of classification task with one class per alphanumeric character. Existing algorithms can be divided into two categories : template matching based and learning based methods.

Template matching based methods (Jalil, Basari, Salam, Ibrahim, & Norasikin, 2015; Khalil, 2010) consist on comparing the similarity of a given character and templates. The most similar template is then chosen. Several similarity measures have been proposed including Mahalanobis distance and Hamming distance (Du et al., 2013). These methods have usually been applied to binary images and are limited since they work for single character size and font and they do not support rotation or broken characters.

Learning based methods are more robust and can deal with characters of different font, illumination, or rotation. They use machine learning techniques to discriminate characters using one or multiple features such as edge density, gradient, scale invariant transform (SIFT) etc. Radzi and Khalil-Hani (2011) uses a 5-layer CNN to recognize Malaysian license plate where individual characters are manually extracted and segmented. The recognition is then performed as a classification task with 33 classes and achieve 98.79% accuracy on a reduced number of samples.

Pham, Dinh, and Nguyen (2018) proposes an approach using a CNN classifier for the recognition of license plate characters and uses firstly some pre-processing techniques on input images, such as filtering, thresholding, and then segmentation.

2.2.2. Segmentation-free approaches

For segmentation-free methods, the recognition is performed on the global license plate image without character segmentation. Usually, a sliding window over the input image is used to generate many tentative characters in small steps. Each tentative character is then used by a recognizer. When the input image is totally swiped by the sliding window, the predicted outputs are analyzed and the final sequence is decided. Consecutive same characters are considered a single character, and character space is used to separate others. In the context of License Plate Recognition, few works have proposed segmentation-free approaches based on DL techniques. Cheang et al. (2017) propose a CNN to perform feature extraction on the license plate and a recurrent neural network to learn the sequential order of character features. Li et al. (2018) proposes to recognize the license characters as a sequence labeling problem using a recurrent neural network with long short-term memory (LSTM) which is trained to recognize the sequential features extracted

from the whole license plate via CNNs. Jain et al. (2016) uses a deep (16-layers) CNN based on Spatial Transformer Networks (Jaderberg, Simonyan, Zisserman, & Kavukcuoglu, 2015), to perform a less sensitive character recognition to spatial transformations on whole LP image and avoiding challenging task of image segmentation into characters. Silva and Jung (2017) use YOLO-based network to detect and recognize license plate characters using a joint classification-detection engine. Bjrkklund, Fiandrotti, Annarumma, Francini, and Magli (2019) use a CNN architecture for identifying characters within an image of license plate and localizing the character bounding box corners. This step deal with a 33 classes classification task for Italian LPs.

In general, we note that the use of deep learning techniques to license plate detection and recognition is still limited and restrictive to specific conditions (specific nations, uni-lingual plates, special LP formats...). Only few works perform ALPR in an end-to-end fashion and most of the proposed methods rely on hand-crafted features. To the best of our knowledge, our approach is the first to apply a state-of-art deep learning based detector for multilingual and multi-norm LP detection and recognition.

3. Specification of Tunisian LP

Tunisian license plates have a specific format that is different from European and American plates. They are composed of multi-lingual characters (Arabic and Latin), Arabic words and digits. The Tunisian LP has 26 norms depending of their pattern and colors. Standard pattern for Tunisian license plates are "XXX تونس XXXX",

where X is a number between 0 and 9 and تونس means "Tunisia"

in Arabic script. However, license plates for some other special vehicles differ from this pattern. "XXXXX ن ت" designs vehicles with suspensive regime, "XX CD س XX" for diplomatic vehicles.

Some plates are also double lined. Others contain Tunisian flag. The most commonly plates are writing in white on a black background. However, special vehicles may have other background/font colors. Rental cars are writing in white on a blue background. Government departments and offices of state vehicles are writing in red on white background. Fig. 1 shows some examples of Tunisian license plates of different norms.

4. System architecture

We propose an end-to-end approach for both detecting and then recognizing the vehicle LP in a cascade fashion. The proposed system proceeds on two stages : plate detection stage and plate recognition stage (as shown in Fig. 2). The first module takes the full raw image as input, detects plates in the image using a YOLOv2 architecture and outputs the cropped plate images to the second module. The recognition step takes as input the cropped plate image. Two LP recognition engines are compared in this work. The segmentation-free approach integrates the advantages of both CNN for feature extraction and RNN for LP sequence modeling and transcription and do not need any prior segmentation in the training step. The second approach is based on a joint detection/recognition approach using YOLOv2 and requires sophisticated manual labeling of the LP components with bounding-boxes.

4.1. License plate detection

License plate detection is the first stage of the proposed pipeline. It has a great influence on the whole system performance. The detection system must be robust to environment variations and to Tunisian license plate specifications where the LP formats



Fig. 1. Examples of Tunisian license plates.

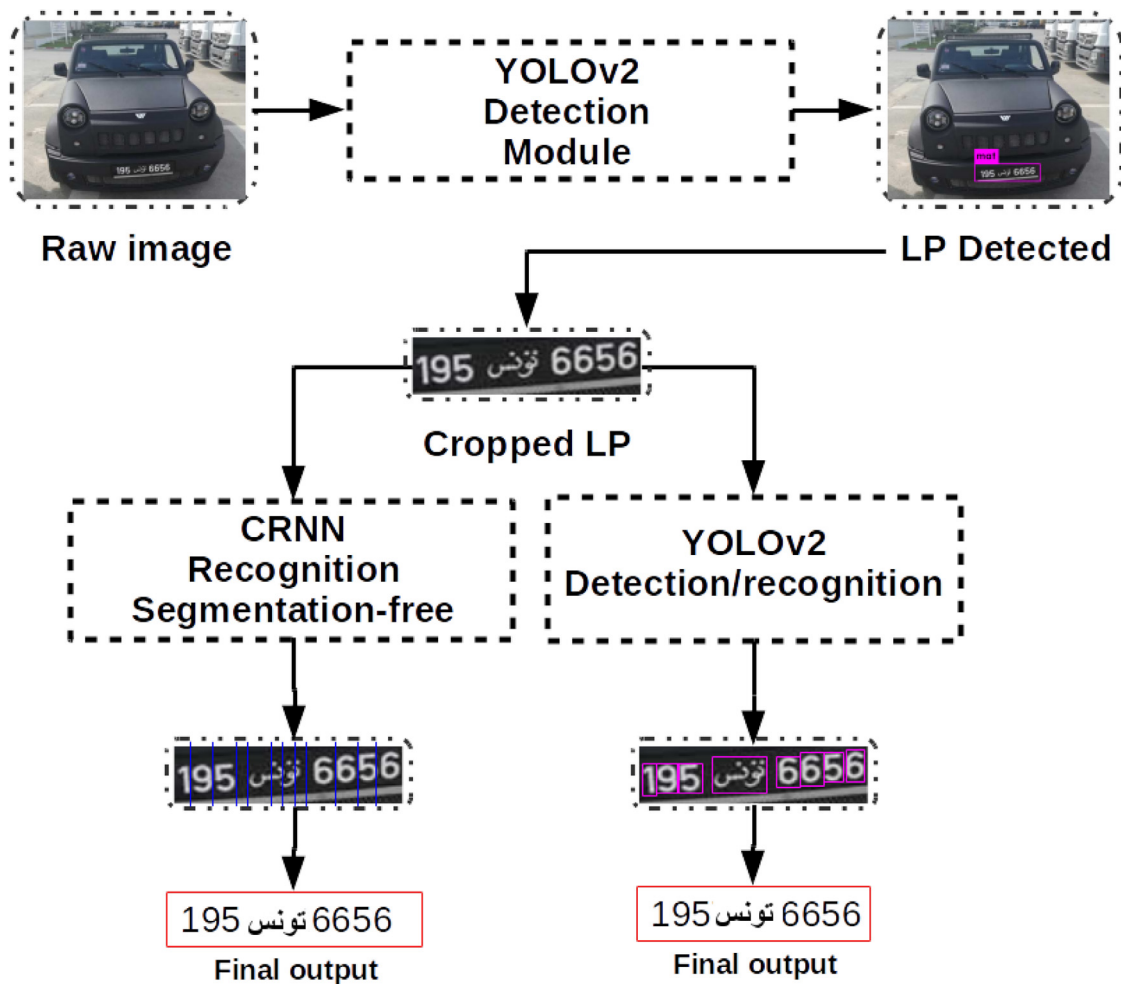


Fig. 2. Illustration of our license plate detection/recognition system based on YOLOv2 and CRNN. The output of the YOLOv2 detection module becomes the input of both CRNN recognition module and YOLOv2 detection/recognition module.

vary significantly as described in Section 3. This requires a combination of a strong visual feature extractor and a mechanism to efficiently find the LP boundaries. To perform LP detection and achieve a good compromise between accuracy rates and running times, we use a very popular state of the art real-time object detection system in its second version dubbed YOLOv2 (Redmon & Farhadi, 2016). YOLOv2 reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. It uses a multi-scale training method and offers a trade-off between speed and accuracy. To detect license plates, YOLO uniformly divides the image into cells. During training, the deep neural network takes images from the training dataset as input, forms a loss function that incorporates the cross-entropy loss, the L_2 regression loss and the randomness and back-propagates the gradients to update the parameters. Thus, during testing, the network gives as output the LP bounding box.

We adapt the original YOLO Architecture to perform LP detection. YOLO predicts B bounding boxes ($B = 5$) each with four coordinates and confidence, so the number of filters is given by: $N_{filters} = (N_{classes} + B) \times 5$. As we intend to detect only one class corresponding to the LP, so the number of filters is fixed to 30.

YOLO returns a confidence score for each LP bounding box it predicts. However, only if the confidence score is above a certain threshold, the bounding box will show up in the final prediction. By changing this threshold value, we can directly impact the detection precision. A higher threshold means that only highly confidence detections are returned, yielding fewer detections and therefore fewer false positives and more false negatives. A lower threshold yields more detections and therefore more false positives and fewer false negatives. In our case, the confidence value does not considerably affect the detection performance since each image in our datasets contains only one vehicle, so only one LP. So that, we keep only the detection with the largest confidence in cases where more than one LP is detected. In this work, the value of 0.25 is empirically chosen based on recall/precision on the validation set. It guarantees that YOLO does not predict a LP bounding box when the confidence is low.

Since all parameters are fixed, we train YOLO network on the fully annotated LP regions. For the detection stage, we run the network over the full raw input images without any pre-processing. Since the LP is detected, it is cropped and sent to the second stage for recognition (see Fig. 2).

4.2. License plate recognition

The second stage of the proposed system is the LP recognition. In this work, two different recognition engines are compared as described in the following.

4.2.1. Segmentation-free recognition approach

The first approach consider the recognition of the license plate characters as a sequence-labeling problem. The network architecture consists of two components, including the convolutional layers and the recurrent layers. The deep recurrent neural network with a bidirectional long short-term memory is trained to recognize the sequential features extracted from the whole license plate image via CNNs. The main advantage of this approach is that it is segmentation free, we need only LP images and their corresponding label sequences, avoiding the labor of labeling positions of individual plate characters.

The CNN model is used to extract a sequential feature representation from the input LP image. The sequence of feature vectors are then used as the input of RNN which takes the whole sequence history into consideration. To avoid gradient vanishing, LSTM is employed, instead of vanilla RNN unit. LSTM is a special type of RNN unit, which consists of a memory cell and gates. To

Table 1

The network configuration of the convolutional recurrent neural network k : filter size, s : stride, p : padding.

Layer type	Configurations
Transcription	–
Bidirectional-LSTM	#neurons:256
Bidirectional-LSTM	#neurons:256
Map-to-Sequence	–
Convolution	#filters : 512, $k : 2 \times 2$, $s : 1$, $p : 0$
MaxPooling	kernel : 1×2 , $s : 2$
BatchNormalization	–
Convolution	#filters : 512, $k : 3 \times 3$, $s : 1$, $p : 1$
BatchNormalization	–
Convolution	#filters : 512, $k : 3 \times 3$, $s : 1$, $p : 1$
MaxPooling	kernel : 1×2 , $s : 2$
Convolution	#filters : 256, $k : 3 \times 3$, $s : 1$, $p : 1$
Convolution	#filters : 256, $k : 3 \times 3$, $s : 1$, $p : 1$
MaxPooling	kernel : 2×2 , $s : 2$
Convolution	#filters : 128, $k : 3 \times 3$, $s : 1$, $p : 1$
MaxPooling	kernel : 2×2 , $s : 2$
Convolution	#filters : 64, $k : 3 \times 3$, $s : 1$, $p : 1$
Input	174×32 gray-scale image

integrate both past and future context into the model, we combine two LSTMs, one forward and one backward, into a bidirectional LSTM which is generally beneficial for image text recognition. The network configuration used in our experiments is summarized in Table 1. It is based on the architecture proposed in Shi, Bai, and Yao (2015). The architecture of the convolutional layers is based on the VGG architecture. To make it suitable for recognizing LP, max-pooling layers are modified to 1×2 sized rectangular pooling windows instead of the conventional squared ones. The convolutional layers are succeeded by 2-layer bidirectional LSTM. The output of BLSTM is connected to Connectionist Temporal Classification (CTC) that extends the use of RNN to non-segmented data. In fact, by adding a new class (Blank) the CTC transforms the output of BLSTM into a conditional probability distribution over the classes (characters and blanks). Finally, the output from the CTC is decoded by removing the blanks into the most probable sequence of characters.

Before being fed into the network, all the images need to be scaled to the same height. The image width was chosen based on the Tunisian LPs ratio (4×1) to make it suitable for recognizing the Tunisian license plates. Experiments on the validation set have shown that setting up the input image resolution to 174×32 help to extract effective and useful features for recognition. To train the network, we only need images and their corresponding label sequences. We define 23 character models corresponding to digits, letters composing the Arabic word “تونس”, the four Arabic letters “ن، ت، س، د”، the Latin letters “C, D, R, S” and the space character. The full recognition steps are shown in Fig. 3.

4.2.2. Detection/recognition approach

The second LP recognition strategy that we propose in this work is based on YOLOv2 network where segmentation and recognition are performed as part of the detection step. Detecting LP components directly from the cropped LP image would perform better than the pipeline detecting LP components on the full raw image since it will only need to search in the regions where license plates are found.

This module detects and recognizes all possible components in Tunisian plate including digits, the Arabic word “تونس”, the Arabic letters “ن، ت، س، د” and the Latin letters “C, D, R, S”. This second detection/recognition approach presents many advantages

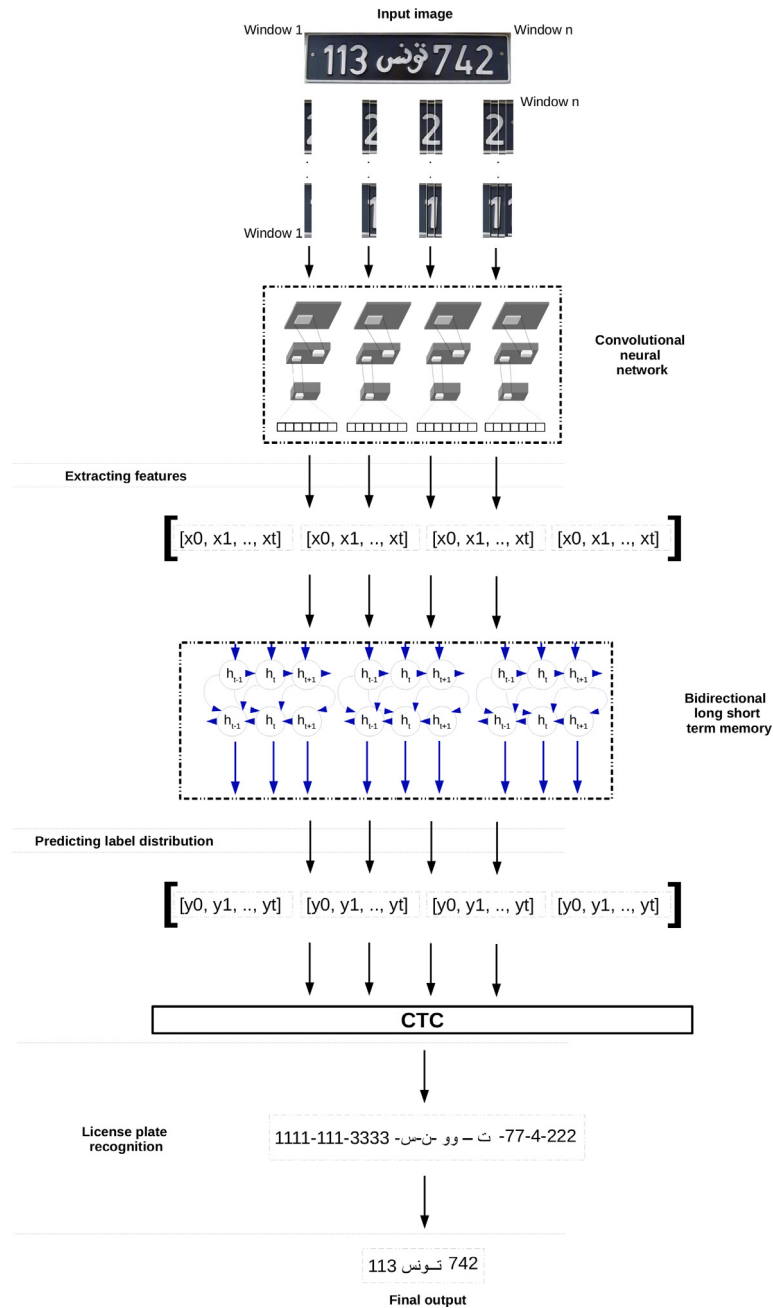


Fig. 3. Architecture of the segmentation-free LP recognition engine.

compared to the free segmentation method presented previously. In fact it is more suitable to recognize double-lined plates thanks to its versatility and ability to learn general components features independently of their positions in the LP. Moreover, it is more adapted to recognize Arabic words in the LP, which are considered as a global text region, contrary to the CRNN which recognize a word by recognizing their character sequence. Finally it can be trained to detect flag images in the license plate which can perturb the recognition process of CRNN specially when it is located in the middle of the LP characters.

The full system architecture is shown in Fig. 4. To adapt the YOLOv2 architecture to detect the LP components, we have changed the number of filters in the last convolutional layer to 125 to match the number of classes (20 LP components).

The training data consists of labeled images with a rectangular bounding box around the license plate components. To gener-

ate annotated LP training data, we have proposed a semi-automatic algorithm as described in the following.

4.2.3. Semi-automatic annotation of LP components

License plate components detection/recognition usually requires large annotated training sets. The creation of such datasets requires expensive manual annotation. In order to reduce the time and cost of annotation processing while ensuring the accuracy, we propose a semi-automatic annotation procedure. The core idea is using an automatic incremental training to roughly annotate the LP image, just before the human review and revision.

To generate the automatic annotation, we start with a subset D of LP images. Each image in D is manually annotated at components level (bounding box + label), we denote by detailed annotation. The algorithm takes as input a set SNL of LP images and their corresponding label sequences (without detailed annotations)

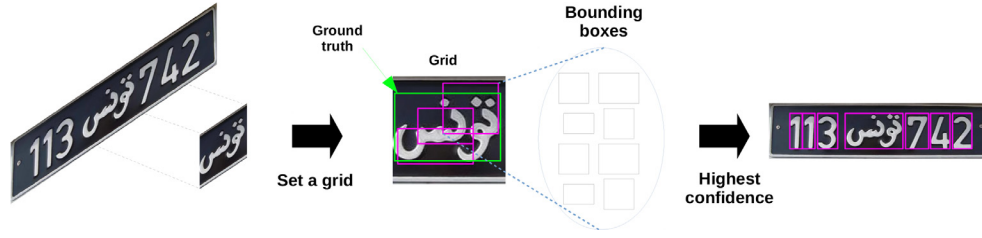


Fig. 4. LP recognition pipeline using YOLO network.

Table 2

Number per iteration and percentage (the total of all iterations) of annotated LP images on GAP-LP dataset.

start_size/ Iteration	1	2	3	4	5	6	7
500	6150 (86.41%)	328 (91.02%)	168 (93.38%)	68 (94.33%)	38 (94.87%)	17 (95.11%)	×
250	5130 (72.08%)	1247 (89.60%)	328 (94.21%)	57 (95.01%)	35 (95.50%)	33 (95.96%)	10 (96.10%)

Table 3

Number per iteration and percentage (the total of all iterations) of annotated LP images on Radar dataset.

start_size/ Iteration	1	2	3	4	5	6	7
500	1263 (63.18%)	124 (69.38%)	141 (76.43%)	55 (79.18%)	38 (81.09%)	23 (82.24%)	13 (82.89%)
250	380 (19%)	934 (65.73%)	197 (75.58%)	67 (78.93%)	24 (80.14%)	16 (80.94%)	×

and returns as output a set SL that contains all LP images that are automatically annotated at components level. The proposed semi-automatic annotation algorithm proceeds in several iterations. We start with training YOLO on D to obtain the network model M_0 . Then, for each iteration j , we construct a new subset S_j containing images of SNL that are correctly recognized by M_{j-1} and we fine-tune the YOLO network on S_j to obtain M_j . We reiterated this process until all the images of SNL are labeled at components level or no image of SNL can be correctly recognized. At each iteration, S_j is added to SL and each LP image that is correctly annotated is removed from SNL . The detailed algorithm steps are given in Algorithm 1.

We present in Tables 2 and 3 the evolution of the semi-automatic annotation on GAP-LP and Radar LP image datasets³ of varying levels of difficulty. For each iteration, we present the number of LP images which are correctly labeled at the component level. We also present the percentage (the total of all iterations) of the correctly labeled LP images. It can be shown that the proposed algorithm annotates 95.11% of the dataset in only 7 iterations when starting with 500 images with detailed annotation. We show that 250 detailed annotated images are enough to annotate 96.10% of the dataset as shown in the second line of Table 2. It is interesting to note that since the first two iterations more than 90% of the LP images are annotated. The proposed semi-automatic annotation algorithm is also evaluated on Radar dataset containing license plates images in various conditions such as low resolution, severe weather conditions and variable lighting conditions. As shown in Table 3, more than 80% of the images are automatically annotated starting with only 250 images with detailed annotation.

The annotation quality of the semi-automatic annotation process is evaluated using 250 LP images with hand annotated component locations. We obtained an average IoU of 88 for the GAP-LP dataset and of 85 for Radar dataset which confirms the robustness of the proposed algorithm.

Algorithm 1 Semi-automatic annotation of LP components.

Inputs :

$D = \{I_1, \dots, I_N\}$ a subset of images I_i with labeled LP components (detailed annotation)

$SNL = \{I_1, \dots, I_M\}$ a subset of LP images and their corresponding sequence labels $L = \{L_1, \dots, L_M\}$ (without detailed annotations)

Output :

$SL = \{I_0, \dots, I_k\}$: a set of LP images I_i and their corresponding components labels

- 1: Train YOLO on D to obtain a CNN model M_0
- 2: Initialize j to 1
- 3: $SL = \phi$
- 4: **repeat**
- 5: $S_j = \phi$
- 6: **for** each image I_i in SNL **do**
- 7: Test I_i with YOLO using M_{j-1} to detect its components labels
- 8: Concat LP components labels to obtain the sequence label $GT(I_i)$
- 9: **if** $(GT(I_i) == L_i)$ **then**
- 10: $S_j \leftarrow S_j \cup \{I_i\}$
- 11: $SNL \leftarrow SNL \setminus \{I_i\}$
- 12: **end if**
- 13: **end for**
- 14: $SL \leftarrow SL \cup S_j$
- 15: Fine-tuning of Yolo on S_j to obtain M_j
- 16: Increment j
- 17: **until** $SNL == \phi$ or $S_j == \phi$

5. Experimental results

In this section, we conduct experiments to verify the effectiveness of the proposed ALPR system. All the experiments were performed on a NVIDIA GTX 1080 Ti GPU (3584 CUDA cores and 11 GB of RAM). Experiments were conducted in two datasets collected

³ The description of the datasets is given in Section 5.



Fig. 5. GAP-LP sample images and their respective plates.



Fig. 6. Radar image samples, license plates blurred for privacy. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

Table 4

Datasets statistics.

	Training set	Validation set	Test set
GAP-LP dataset	7117	456	1602
Radar dataset	3998	450	2000

from real road surveillance and parking access control environments. Datasets statistics are reported in Table 4.

GAP-LP dataset

The GAP-LP dataset images were acquired with different quality cameras under different resolutions, view angles and daylight lighting conditions to enable robust LP detection/recognition algorithms testing. GAP-LP dataset⁴ is freely available to the research community. It is composed of 9175 fully annotated images for both LP detection and recognition. The dataset is split as follows : 7117 images for training, 456 for validation and 1602 images for test. Fig. 5 shows some image samples from the GAP-LP dataset.

Radar dataset

The Radar dataset is composed of 6448 JPEG color images taken from radars on different Tunisian roads. The images were taken from real scenes with complex backgrounds and under various lighting conditions (day, night, sunshiny, raining, ...), from different angles and positions and for several license plate shapes (square and rectangle) and norms. The dataset is divided into three sets: 3998 images for training, 2000 for test and 450 for validation. Fig. 6 shows some image samples from the Radar dataset.

The experimental section is divided into three parts: the LP detection results, the LP recognition results and the comparison with state-of-the-arts ALPR systems. The following sections describe the evaluation details and the obtained results.

5.1. LP detection results

To evaluate the detection stage performance, we define the intersection over union (IoU) as:

$$IoU = \frac{area(Rd \cap Rg)}{area(Rd \cup Rg)} \quad (1)$$

⁴ <https://sites.google.com/site/matdbparking>.

Table 5

License plate detection: precision and recall rates for IoU threshold of 0.5.

Dataset	GAP-LP	Radar
Recall	100%	99.09%
Precision	100%	100%

Table 6

License plate detection: recall rates for IoUs ranging from 0.3 to 0.8.

Database/IoU	30	40	50	60	70	80
Radar (Recall %)	100	99.79	99.09	97.93	85.49	65.88
GAP-LP (Recall %)	100	100	100	99.06	93.70	67.83

Where Rd and Rg are LP regions of the detected bounding box and ground-truth respectively. A true positive, TP , is defined as a detection which has a minimum overlap (IoU) of 0.5 with the ground-truth box. Detections with lower overlap are false positives, FP . Missed ground-truth LP samples are denoted as false negatives, FN . The recall and precision are then computed by:

$$recall = \frac{TP}{(TP + FN)} \quad (2)$$

$$precision = \frac{TP}{(TP + FP)} \quad (3)$$

It can be observed from Table 5 that our method achieved impressive detection rates with a precision of 100% and a recall of 100% on the GAP-LP dataset and 99.09 on the Radar dataset. Here a detection is considered to be correct if the overlap between the detection and ground-truth bounding box is greater than 0.5 ($IoU > 0.5$).

In Table 6, we show the License plate detection recall values considering different IoU acceptance values (from 0.3 to 0.8). As can be observed, our method is not much sensitive for IoU variations, except for very high values.

Fig. 7 shows some qualitative LP detection examples from the Radar dataset in different illumination conditions. The results confirm the robustness of the proposed detection approach.

5.2. Recognition results

The LP recognition rate (LP-RR) is defined as the number of correctly recognized license plates divides by the total number of

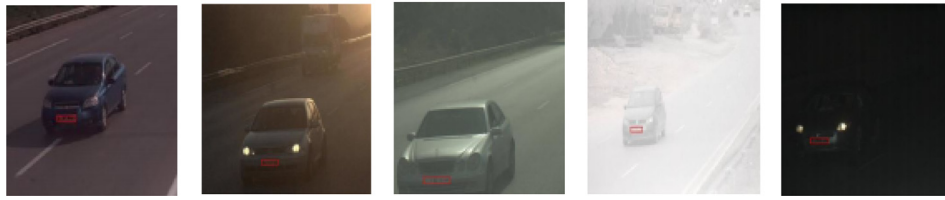


Fig. 7. Qualitative detection examples in different illumination conditions. The LPs were blurred due to privacy constraints.

Table 7

LP and character recognition rates obtained by YOLO on the GAP-LP dataset.

Training datasets	GAP-LP	Radar	GAP-LP + Radar
LP-RR	95.00%	90.35%	97.67%
CRR	97.74%	95.48%	98.95%

Table 8

LP and character recognition rates obtained by YOLO on the Radar dataset.

Training datasets	GAP-LP	Radar	GAP-LP + Radar
LP-RR	54.26%	91.46%	68.48%
CRR	66.28%	96.85%	80.41%

ground-truth samples. A correctly recognized license plate means all the characters of the LP are recognized correctly. We also report the character recognition rate (CRR) defined as the number of correctly recognized license plate characters divided by the total number of ground-truth character samples.

$$LP - RR = \left(\frac{\text{no. of correctly recognized LP}}{\text{Total no. of LP}} \right) \times 100 \quad (4)$$

$$CRR = \left(\frac{\text{no. of correctly recognized character}}{\text{Total no. of character}} \right) \times 100 \quad (5)$$

Tables 7 and 8 show the obtained results of the detection/recognition approach using GAP-LP and Radar test sets.

The proposed detection/recognition system (DR-LP) based on the YOLO network successfully recognized 97.67% of the license plates on the GAP-LP test set and 91.46% on the Radar test set. We notice that using different training datasets did not always improve the prediction accuracy. The recognition accuracy is decreased to 90.35% on the GAP-LP test set using the Radar training set and to 54.26% on the Radar test set when using the GAP-LP training set. We conclude that the decrease in similarity between the training and test sets seemed to have negative impact on the system performance. Using a training set composed of a mix of GAP-LP and Radar images, a relative improvement of 2.67% is obtained on GAP-LP test set. Contrarily, the performance on the Radar test set is largely decreased. We conclude that augmenting the training set by more complex images can help the network to recognize LP images with simple conditions, but the reverse is not true.

We present in Tables 9 and 10 the LP recognition accuracy of the segmentation free approach based on the convolutional recurrent neural network architecture. The obtained results show that the CRNN successfully recognizes 95.88% of the license plates on GAP-LP dataset. An improvement of 0.88% is achieved compared to YOLO when using the same training dataset (GAP-LP). Contrarily,

Table 9

LP and character recognition rates obtained by CRNN on GAP-LP dataset.

Training dataset	GAP-LP	Radar	GAP-LP + Radar
LP-RR	95.88%	92.88%	94.25%
CRR	98.83%	96.33%	98.11%

Table 10

LP and character recognition rates obtained by CRNN on Radar dataset.

Training dataset	GAP-LP	Radar	GAP-LP + Radar
LP-RR	73.03%	42.88%	78.13%
CRR	91.03%	80.33%	92.31%

Table 11

LP recognition rates obtained by CRNN using generated data.

Test Dataset	GAP-LP	Radar
CRNN	95.93%	80.86%

the performance on Radar test set is largely decreased to 42.88% when Radar training set is used. This can be explained by the reduced size of the Radar training set. Using a larger training set composed by a mix of Radar and GAP-LP datasets, an improvement of 30.15% is obtained. We conclude that CRNN need a larger training dataset than YOLO to correctly learn the parameters of CNN and LSTM. Hence, we have tried to expand the size of the training dataset with synthetic LP images.

5.2.1. Experimentation with generated data

As concluded above, achieving a higher performance requires a large amount of annotated license plate image data covering various situations. To tackle this problem, we develop a script to synthesize images of license plates with different fonts, colors and component composition rules. Fig. 8 shows some examples of the synthesized LP images. We have generated 40,000 LP training images.

Experiments show that the LP recognition system based on YOLO architecture trained with the additional generated data did not improve the prediction accuracy. Contrary, generated data slightly improves the performance of CRNN as shown in Table 11. An improvement of 2.73% is achieved on Radar test set and of 0.05% on GAP-LP test set.

Fig. 9 illustrates some of the recognition results obtained by the proposed system on the GAP-LP and Radar datasets. It is noteworthy that our system can generalize well and correctly recognizes LPs under different lighting conditions as presented in the



Fig. 8. Synthesized images of license plates. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)



Fig. 9. Qualitative results obtained by the proposed ALPR system in the GAP-LP and Radar datasets. Green text refers to correctly recognized LP, while miss-recognized LP are written in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 12

Comparison of plate detection rates on three subsets of the AOLP dataset.

Method	AC		LE		RP	
	Recall	Precision	Recall	Precision	Recall	Precision
Proposed	99.82	100	96.42	100	99.17	100
Hsu et al. (2013)	96	91	95	91	94	91
Selmi, Halima, and Alimi (2017)	96.8	92.6	93.3	93.5	96.2	92.9
Li et al. (2018)	98.38	98.53	97.62	97.75	95.58	95.28

first two rows of Fig. 9. The following rows show examples of incorrectly recognized LP by CRNN or by YOLOv2 or by both. It can be shown that CRNN can not recognize double-lined LP, contrarily to YOLO which is more suitable to recognize them thanks to its versatility and ability to learn general component features independently of their positions. We notice that some LPs written in red on a white background are not correctly recognized by neither YOLO nor CRNN. This is not surprising as the training datasets do not include enough LP images with this norm. In some other cases, the LP recognition fails due to the inclination of the LP or to the presence of noise.

5.3. Comparison with state-of-the-arts ALPR systems

To compare the proposed system with state-of-the art ALPR systems on GAP-LP and Radar datasets, two commercial systems Sighthound (Masood et al., 2017) and OpenALPR⁵ are evaluated. Unfortunately, the comparison was not possible because the two

commercial systems do not support Arabic script and fail to recognize Tunisian license plates. Alternatively, we have conducted additional experiments on the AOLP public dataset (Hsu, Chen, & Chung, 2013), which has 2049 images of Taiwan license plates. This database is categorized into three subsets with different level of difficulty: access control (AC), traffic law enforcement (LE), and road patrol (RP). The detection results of our LP detection system are compared to three state-of-the art LP detection systems. Note that all ALPR systems are evaluated in the same training and test sets. It can be shown from Table 12 that our LP detection approach produces a higher recall and precision for each dataset category. The detection results of experimentation for the AOLP dataset are provided in Table 12. It can be shown that our LP detection approach produces a higher recall and precision for each dataset category.

We present in Table 13 the character and the plate recognition rates of the proposed CRNN and YOLO recognition approaches and we compare their performance against five state-of-art results published on the AOLP dataset. The obtained results show that our LP recognition approaches provide the best performance in terms of character and LP accuracies.

⁵ OpenALPR Cloud API, <http://www.openalpr.com/cloud-api.html>.

Table 13

Comparison of plate Recognition rates on three subsets of the AOLP dataset.

Method	AC		LE		RP	
	CRR	LP-RR	CRR	LP-RR	CRR	LP-RR
Proposed (CRNN)	99.49	97.97	98.97	95.95	95.62	81.81
Proposed (YOLO)	95.31	93.75	97.62	98.91	93.43	90.42
Hsu et al. (2013)	95	88.5	93	86.6	94	85.7
Selmi et al. (2017)	96.2	–	95.4	–	95.1	–
Li et al. (2018)	–	94.85	–	94.19	–	88.38
Jiao et al. (2009)	90	–	86	–	90	–
Anagnostopoulos et al. (2006)	92	–	86	–	91	–

Table 14

The computational time required in each ALPR stage.

Stage	Time (s)
Detection	0.0367
Recognition (CRNN)	0.0076
Recognition (YOLO)	0.012

5.4. Runtime evaluation

Experiments have been done on a computer with 2.6GHz Xeon, 64GB RAM and NVIDIA GTX 1080 Ti GPU. We implemented our system on Linux Ubuntu operating system. For each processing stage, the average runtime has been computed from different runs made in the experiment. The time computation of the LP identification step and the LP recognition step are given in Table 14. It is worth noting that our system is able to process (detection + recognition) LP images in 0.0443 s with CRNN and in 0.0487 s with YOLOv2 which is sufficient for real-time usage.

6. Conclusion

In this paper, we have presented a robust real-time ALPR system using a pipeline with two deep learning stages. The LP detection stage is based on the state-of-the-art YOLOv2 object detection CNNs. For the second stage, we compare two recognition engines: a sequence labeling method which recognize the whole license plate without character-level segmentation and a joint detection/recognition approach that performs the recognition on the plate component level. The proposed system is robust to illumination and weather conditions and is capable to achieve a full LP recognition rate of 97.67% in the GAP-LP dataset, and of 91.46% in Radar dataset with a reasonable computational time. We also introduced a new public dataset for multi-norm and multilingual ALPR that includes 9175 fully annotated images. Compared to the existing datasets for this task, GAP-LP is the largest ALPR dataset making it suitable for trying and evaluating deep learning techniques. In order to reduce the time and cost of annotation processing, we have proposed a new semi-automatic annotation procedure of LP images with labeled component bounding boxes. Future work consists of integrating the recognition of vehicle make and model to improve the vehicle identity recognition process and help check correlation with data stored on police and homeland security databases.

Declaration of Competing Interest

None.

Credit authorship contribution statement

Yousri Kessentini: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project

administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Mohamed Dhia Besbes:** Software, Supervision, Validation, Visualization, Writing - review & editing. **Sourour Ammar:** Formal analysis, Methodology, Writing - original draft, Writing - review & editing. **Achraf Chabbouh:** Conceptualization, Data curation, Software, Writing - review & editing.

Acknowledgments

This research is supported by the VRR research fund from the **Tunisian Ministry of Higher Education and Scientific Research**. Authors would like also to express their deepest gratitude to our industrial partner KOTECH and to the Tunisian Ministry of the Interior for providing the dataset.

References

- Al-Shemarry, M. S., Li, Y., & Abdulla, S. (2018). Ensemble of adaboost cascades of 31-LBPS classifiers for license plates detection with low quality images. *Expert Systems with Applications*, 92, 216–235.
- Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Loumos, V., & Kayafas, E. (2006). A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Transactions on Intelligent Transportation Systems*, 7(3), 377–392.
- Arth, C., Limberger, F., & Bischof, H. (2007). Real-time license plate recognition on an embedded DSP-platform. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1–8).
- Bjrkund, T., Fiandrotti, A., Annarumma, M., Francini, G., & Magli, E. (2019). Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recognition*, 93, 134–146.
- Chang, S.-L., Chen, L.-S., Chung, Y.-C., & Chen, S.-W. (2004). Automatic license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*, 5(1), 42–53.
- Cheang, T. K., Chong, Y. S., & Tay, Y. H. (2017). Segmentation-free vehicle license plate recognition using convnet-RNN. CoRR. arXiv: 1701.06439.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). High-performance neural networks for visual object classification. CoRR. arXiv: 1102.0183.
- Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2013). Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2), 311–325.
- Casey, R. G., & Lecolinet, E. (1996). A survey of methods and strategies in character segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18, 690–706.
- Giannoukos, I., Anagnostopoulos, C.-N., Loumos, V., & Kayafas, E. (2010). Operator context scanning to support high segmentation rates for real time license plate recognition. *Pattern Recognition*, 43(11), 3866–3878.
- Gou, C., Wang, K., Yao, Y., & Li, Z. (2016). Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines. *IEEE Transactions on Intelligent Transportation Systems*, 17(4), 1096–1107.
- Guo, J. M., & Liu, Y. F. (2008). License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques. *IEEE Transactions on Vehicular Technology*, 57(3), 1417–1424.
- Hsieh, J.-W., Yu, S.-H., & Chen, Y.-S. (2002). Morphology-based license plate detection from complex scenes. In *Object recognition supported by user interaction for service robots: Vol. 3* (pp. 176–179 vol.3).
- Hsu, G. S., Chen, J. C., & Chung, Y. Z. (2013). Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*, 62(2), 552–561.
- Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). Spatial transformer networks. CoRR. arXiv: 1506.02025.
- Jain, V., Sasindran, Z., Rajagopal, A., Biswas, S., Bharadwaj, H. S., & Ramakrishnan, K. R. (2016). Deep automatic license plate recognition system. In *Proceedings of the tenth Indian conference on computer vision, graphics and image processing*. In ICVGIP '16 (pp. 6:1–6:8). New York, NY, USA: ACM.
- Jalil, N. A., Basari, A. S. H., Salam, S., Ibrahim, N. K., & Norasikin, M. A. (2015). The utilization of template matching method for license plate recognition: A case

- study in malaysia. In H. A. Sulaiman, M. A. Othman, M. F. I. Othman, Y. A. Rahim, & N. C. Pee (Eds.), *Advanced computer and communication engineering technology* (pp. 1081–1090). Cham: Springer International Publishing.
- Jiao, J., Ye, Q., & Huang, Q. (2009). A configurable method for multi-style license plate recognition. *Pattern Recognition*, 42(3), 358–369.
- Khalil, M. (2010). Car plate recognition using the template matching method. *International Journal of Computer Theory and Engineering*, 2(5), 683.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., & Hubbard, W. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541–551.
- Li, H., Wang, P., You, M., & Shen, C. (2018). Reading car license plates using deep neural networks. *Image and Vision Computing*, 72, 14–23.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., & Fu, C. (2015). SSD: Single shot multibox detector. *CoRR*. arXiv: 1512.02325.
- Masood, S. Z., Shu, G., Dehghan, A., & Ortiz, E. G. (2017). License plate detection and recognition using deeply learned convolutional neural networks. *CoRR*. arXiv: 1703.07330.
- Naimi, A., Kessentini, Y., & Hammami, M. (2016). Multi-nation and multi-norm license plates detection in real traffic surveillance environment using deep learning. In *Proceedings of the 23rd international conference on neural information processing - volume 9948* (pp. 462–469). New York, NY, USA: Springer-Verlag New York, Inc..
- Ng, H. S., Tay, Y. H., Liang, K. M., Mokayed, H., & Hon, H. W. (2015). Detection and recognition of malaysian special license plate based on SIFT features. *CoRR*. arXiv: 1504.06921.
- Nguyen, C. D., Ardabilian, M., & Chen, L. (2009). Robust car license plate localization using a novel texture descriptor. In *2009 Sixth IEEE international conference on advanced video and signal based surveillance* (pp. 523–528).
- Pham, V. H., Dinh, P. Q., & Nguyen, V. H. (2018). Cnn-based character recognition for license plate recognition system. In N. T. Nguyen, D. H. Hoang, T.-P. Hong, H. Pham, & B. Trawiński (Eds.), *Intelligent information and database systems* (pp. 594–603). Cham: Springer International Publishing.
- Qiao, S., Zhu, Y., Li, X., Liu, T., & Zhang, B. (2010). 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).
- Radzi, S. A., & Khalil-Hani, M. (2011). Character recognition of license plate number using convolutional neural network. In H. Badioze Zaman, P. Robinson, M. Petrou, P. Olivier, T. K. Shih, & S. Velastin (Eds.), *Visual informatics: Sustaining research and innovations* (pp. 45–55). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 779–788).
- Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, faster, stronger. *CoRR*. arXiv: 1612.08242.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.
- Sedighi, A., & Vafadust, M. (2011). A new and robust method for character segmentation and recognition in license plate images. *Expert Systems with Applications*, 38(11), 13497–13504.
- Selmi, Z., Halima, M. B., & Alimi, A. M. (2017). Deep learning system for automatic license plate detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR): Vol. 01* (pp. 1132–1138).
- Sen, E. J., Dixon, K. D. M., Anto, A., Anumary, M. V., Mieheal, D., & Jose, F. (2014). Advanced license plate recognition system for car parking. In *2014 International conference on embedded systems (ICES)* (pp. 162–165).
- Shi, B., Bai, X., & Yao, C. (2015). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *CoRR*. arXiv: 1507.05717.
- Silva, S. M., & Jung, C. R. (2017). Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 55–62).
- Suryatali, A., & Dharmadhikari, V. B. (2015). Computer vision based vehicle detection for toll collection system using embedded linux. In *2015 International conference on circuits, power and computing technologies [ICCPCT-2015]* (pp. 1–7).
- Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)* (pp. 3304–3308).
- Wu, H.-H. P., Chen, H.-H., Wu, R.-J., & Shen, D.-F. (2006). License plate extraction in low resolution video. In *18th International conference on pattern recognition (ICPR'06): Vol. 1* (pp. 824–827).
- Yuan, Y., Zou, W., Zhao, Y., Wang, X., Hu, X., & Komodakis, N. (2017). A robust and efficient approach to license plate detection. *IEEE Transactions on Image Processing*, 26(3), 1102–1114.
- Zhou, W., Li, H., Lu, Y., & Tian, Q. (2012). Principal visual word discovery for automatic license plate detection. *IEEE Transactions on Image Processing*, 21(9), 4269–4279.