License Plate Recognition using Convolutional Neural Networks (CNN): A Review

Ahmed Afify, Amine Boussada, and Mahmoud Nasr

Abstract— **The demand of smart transportation systems has been drastically increasing over the last few years due to the ever-rising numbers of vehicles. License Plate Recognition (LPR) has been one of the most frequently used systems for traffic control and law enforcement. The system first captures a vehicles image then applies some image-processing techniques to enhance the content of the input image. After that, the license plate is precisely located before any character segmentation is performed. Convolutional Neural Network (CNN) then extracts several features, which are utilized for character detection. In this review, we examine different LPR systems based on CNN. We also study the major problems of recent LPR techniques and the proposed solutions to overcome them.**

I. INTRODUCTION

License Plate Recognition (LPR) has become one of the most important aspects of today's Intelligent Transportation Systems [1]. Recently, LPR has been massively used in various traffic and security systems such as traffic toll, custom security, and traffic rule violation. Although LPR has effectively contributed in the development of surveillance and traffic systems, it is still facing major issues which limit its performance. Controlling several vehicles in a traffic jam and tilting plates are some of the challenges of the LPR. Moreover, environmental factors such as low lighting conditions, bad weather, and night vision are other challenges in LPR [2]. Several methods have been developed for license plate recognition. However, in this review, we will solely discuss the systems that make use of Convolutional Neural Networks (CNN). In the last few years, CNN has been widely utilized in multiple computer vision applications such as image classification and objects tracking. LPR systems are composed of three main stages. The first two stages are to locate the license plate and segment its content. The final stage is the process of character recognition. Usually, some image preprocessing techniques such as morphological dilation [3], gray-scale conversion [4], binarization [5] and noise filters [4] are required before doing further analysis. For character segmentation, horizontal projection [2], [5], [6] and vertical projection [7], [8], [5], [6] are usually used. Almost all the methods perform the CNN only for characters classification [2], [5], [6]. However, other researchers used CNN instead for the whole recognition process [9], [10]. The CNN first extracts various features from the input image through multiple deep neural network layers. Then, passes them to following layers which classify the characters according to the extracted features. The proposed algorithms yield a high detection accuracy under normal surrounding conditions. Our review paper first introduces pre-processing image techniques then briefly explains the common methods utilized for plate detection and character segmentation. After that, detailed description of CNN architectures for characters recognition is presented. The review then concludes with a discussion of the advantages and disadvantages of the algorithms followed with proposed future research directions.

II. LICENSE PLACE EXTRACTION

A. *Using Neural Networks*

In recent years, researchers have become more inclined to depend completely on neural networks in the entire process of license plate detection including localizing the license plate in the image. [7] used a Single Shot MultiBox Detector (SSD) consisting of a single layer to develop a binary classification problem to find the plate. [2] went a little further by first detecting the car in the scene using YOLOv2 CNN. This was followed by Support Vector Machine (SVM) algorithm to find the license plate using one-versus-rest technique combined with the Histogram of Oriented Gradients (HOG), which is a feature descriptor used in image processing. This method was found to be both faster and more accurate than using solely SVM for the plate detection. In [10], however, the license plate detection was done to both decide whether it was found and where exactly it was. It utilized a method similar to sliding window to scan the entire image and a binary classifier (Sigmoid function) that would judge the presence of the license plate. An image pyramid was created by generating images of different sizes. The image pyramid is searched, and the LP is extracted if its confidence exceeds 95%. The method proposed in [10] utilized the same layers to recognize characters and digits. This is explained in more details in later sections.

[1] devoted the entire paper to localize the license plate accurately using a Convolutional Neural Network. The method was divided into three modules. The first module was used to extract the convolutional feature maps. Then, the second module made use of the sliding window technique on the feature maps in an attempt to find a subwindow containing the entire license plate. After the subwindow was found, a final module used a regression network to find the exact location of the license plate.

Similarly, [9] chose to divide his technique into two stages; the first stage checks the presence of the license plate while the second stage localizes the plate more accurately. The classification stage consists of 10 3x3 convolutional layers with 4 max pooling layers, followed by 3 fully-connected layers and a binary output judging the existence of the plate. The second stage localizes the four corners of the plate using

3 fully-connected layers which are connected to regression units.

B. Using Image Processing Techniques

1) Binary Threshold: When images are fed to the LPR system as grayscale, they have to be turned into black and white (binary) before the actual processing for plate number detection. This process makes it easier to find the license plate, segment the image into several images with characters, and later on recognize the characters. Although most of previous research prefer to do this step after finding the plate, other researchers still used it in the pre-processing stage [4], [5], [6], [8]. In [4], thresholding technique was used, which sets an intensity limit below which any pixel is set to white while all those above the limit are set to black. [5] and [6] had a better idea were they opted for adaptive thresholding. It allows setting a different threshold to every part of the image to accommodate different lighting intensities across the image and therefore result in better grayscale-to-binary transformation. In [8], Otsu method was suggested as an alternative that would search for the optimum threshold to binarize the image.

2) Filtering: The first step in processing an image to detect the license plate is applying filters to remove noise and make its edges clearer. In [3], a guided filter was used to filter out noise and smooth the image-edge. After that, the edges were re-enhanced using morphological dilation process. On the other hand, iterative bilateral filter was used in [5], which helps reduce the noise in the image while maintaining the edges. [4] used Gaussian blurring to remove noise, which may have sacrificed the clarity of the edges in the process and thus had to enhance them as will be described next.

3) Edge Detection: After a clearer image is retrieved from the filtering process, the image goes through another operation meant to extract the edges. Several approaches for doing that are illustrated in literature, some of which were gathered by [6]. These include the counter algorithm, that is used to recognize closed boundaries. It is combined with Hough Transform, which is used to detect the parallel lines in these closed boundaries. In [8],Vertical Sobel Operator was used for vertical edge detection. [5] suggested utilizing a combination of both Sobel edge and Hough line detection techniques.The previous two papers followed the license plate extraction by connected component analysis, which is described later. [3] made use of the differences in intensities which, after the application of morphological dilation, led to discontinuities in intensity at some points. This allowed the use of Prewitt operators to detect the edges without using more computationally expensive operators such as Sobel.

4) Closing Method: After the edges and lines are detected in an image, all we have now is a set of lines; some are parallel and others intersect each other. A method is needed to find out which of these are actually the license plates. The logical sequence would be to find out the closed contours [6], [8] and the shapes that are in fact the same as that of a license plate; rectangular. A method called Closed Component Analysis (CCA) is usually used to do this job. As shown in [4], [6], [8], the method aims to find the closed contours and classify them into shapes so that we are able to extract the rectangles. After that, shape verification is performed using some parameters of the plates such as their relative height and width, or their aspect ratios. Only the rectangle that match the specified aspect ratio and size parameters are extracted as a license plate.

III. Character Segmentation

Character segmentation is an important stage before character recognition. It is used to split plate characters into individual characters. There are several algorithms for the character segmentation, some of which are based on image processing techniques, and others are based on using CNN.

A. Horizontal Projection

Horizontal projection is used to remove unwanted regions of a license plate. This method allows to extract the upper and lower boundaries of the characters in the plate [2], [5], [6]. It is obtained by summing up the white pixels horizontally along different lines. Fig.1 demonstrates the horizontal histogram of a license plate.

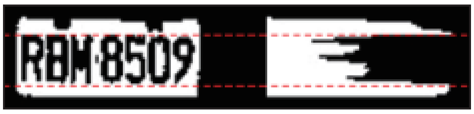

Fig.1. Horizontal projection of characters in a license plate [2]

B. Vertical Projection

After extracting the upper and lower borders of the characters in the plate, vertical projection is then used to find the discontinuities separating characters [7], [8], [5], [6]. The vertical histogram is calculated by summing up the white pixels vertically along a line. Fig.2 shows a vertical projection of characters in a license plate.

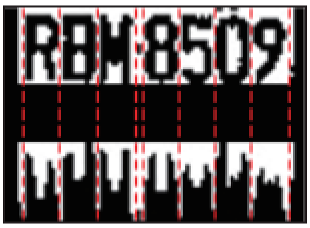

Fig.2. Vertical projection of characters in a license plate[2]

It is sometimes very hard to segment the characters properly due to some noise issues. For instance, the segmentation algorithm can detect screws as characters or mistake a character for two. To avoid such problems, the ratio constraint is added [5]. By setting a threshold for the ratio between the height and width, plate characters can be extracted precisely. This method is robust and can segment characters efficiently even for low resolution images.

C. CNN-based Segmentation

CNN-based segmentation is another method used for characters splitting. The network first extracts some features and then uses them to identify the corners surrounding each character. In [9], the CNN architecture used an input image of 24×40 pixels to detect EU plates. The feature extraction layer is composed of 7 3×3 convolutional layers, 3 Max Pooling and ReLUs. After the image enters the network, the features are extracted and then utilized to find the 4 boundaries of each character. Later, these characters are sent to the recognition stage.

IV. Character Recognition using CNN

A. CNN Basic Architecture

A basic CNN consists of three main stages as illustrated in Fig. 3. A *convolution* stage is the main building block of CNNs as inferred from its name. This stage performs several convolutions (instead of general matrix multiplications) between input and predefined filter to produce a set of pre-activations, called *feature maps*. The next stage, the *detector*, is where each pre-activation is run through a nonlinear activation function such as rectified linear unit (RELU), Sigmoid or Softmax. The final stage serves to provide a summary statistic for all the input to make the data invariant to small translations and reduce the computational and statistical burden on the following layer. This is called the *Pooling* stage.

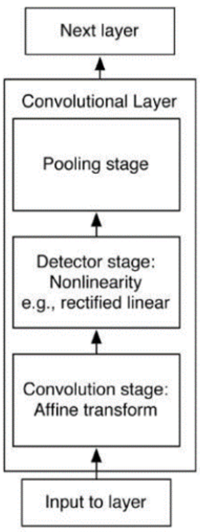

Fig.3. Basic Architecture of a Convolutional Neural Network [11]

B. CNN in Character Recognition

Previously, license plate character recognition was done using template matching. Recently, however, Convolutional Neural Networks become predominant in image recognition. In this section, we discuss the recent work done, and we group similar work together.

[2], [4], [5], [6], [7], [9], and [10] all worked on detecting license plates with 32-36 characters. To begin with, [5] used two colvolutional layers with 1 max pooling layer in between, followed by 2 fully connected layers. They attempted to work on English letters (A-Z) and numbers (0-9) and used 30,000 training images and 6,000 testing images. A downfall for this paper is that they suggested methods for license plate detection and the segmentation but only did the training and testing of the CNN network on separate characters and not on the characters obtained from the plates.

[2] and [7] both worked on detecting letters from A-Z, excluding *I* and *O* and the numbers from 0-9 utilizing very similar network structures. [2] used around 14,500 images with 60% set as training samples. With only 2 convolutional layers, 2 max pooling layers, and 2 fully-connected layers. The network managed to obtain an accuracy of 99.2%. [7] suggested a network structure with 2 convolutional layers and 3 fully connected layers. The paper utilized both high and low-quality images in training to create 2 separate networks and mixed them to obtain a third one, with 1,700 plate images and 50 images for each character of low resolution and about 3500 high resolution image in the training set. The system was tested to assess its accuracy on the testing sets high and low resolutions. While the high-resolution-trained network performed poorly in the low-resolution-testing, and vice-versa, the network trained on the mixed data performed very well on both testing sets with 99.23% accuracy.

A popular network in CNN is the Lenet-5 network made of 5 convolutional layers and 6 fully-connected layers. Lenet-5 is designed to extract the geometric features present locally and is guaranteed to have a high accuracy and can be trained to recognize different fonts and styles. This network, though extremely efficient and accurate, is very complicated, and therefore slow and not suitable for real-time applications. [4] decided to utilize Lenet-5 and can be used to show just how efficient this algorithm is. The paper gives a detailed implementation of the system including all the software used as well as a detailed structured explanation of CNNs. The results obtained show 100% accuracy even though the size of testing data was extremely small(only 200) and the computational time required was not mentioned. Nevertheless, this seems to agree with the well-known belief that the Lenet-5 is extremely efficient. [6] proposed a modification to Lenet-5 to improve its speed. In this network, the total number of layers was reduced to 6 layers. Moreover, the third convolutional layer (C3) was connected to 3x3 neighbours instead of 5x5 which made the system even faster. The idea behind this simplification came from the fact that the license plate is not expected to have more than 4 gray levels so the number of features can be reduced. The network was trained on 112,000 images and was tested on 36,500. These sets included images from the MNIST (Modied National Institute of Standards and Technology) database and other collected sets. The training set included good, normal, and hard subsets. Also, the network was tested on each separately. The accuracy was compared to that of the Lenet-5, and this gave very close accuracy with an average of 93.001% compared to 94.894% by the Lenet-5; therefore giving 20x higher speed and less than 2% lower accuracy.

[9] and [10] followed a more complicated approach by attempting to take on the full problem from plate extraction to character recognition using CNN. [9] worked on Italian license plates, which contain 32 characters. The network had one output node for each character as well as 2 output nodes to give the confidence of plate's existence. The network uses 7 3x3 convolutional layers with 3 max pooling layers for the feature extraction. The characters have to be segmented before classification, and this was done by 3 fully connected layers to find the bounding box around the plate. The classification phase uses the same extracted features and assigns class confidence (each character representing a class) and outputs the confidences. The highest value is then taken as the correct character. An extra output node is also present for the class of no characters being found. [10] used the same approach where they used 9 convolutional layers for feature extraction. These features were then used to detect the car plate and recognize 33 characters and 10 digits. Therefore, it had 3 output branches, one for the plate presence, and 1 for each of the character and digit outputs. The concept of voting was also used to further increase the accuracy of the character recognition. In this approach, the areas where a candidate license plate is found are all detected by using different sizes for the image, or what is referred to as an *image pyramid*. In each of these plates, the characters are detected and only those detected with over a certain threshold value (90% in this paper) are considered to have a vote. The process is repeated for all the candidate plates and the characters with highest votes are output. The authors, however, took a step further into the training data where they developed a way to generate their own training and testing data. The paper proposes a method to generate license plate images with different backgrounds, lighting, noise, and tilting, they even generate incomplete plates. Thus, the network was trained on all kinds of data, where 200,000 positive (with LP) and 200,000 negative images were used, and has shown an accuracy of 98% which rose to 99% with the voting concept. A very important point of critic is that the system was tested on only 302 images, which is a small sample.

V. Discussion

The above proposed methods showed a high recognition accuracy for a license plate under a normal detection angle and good lighting conditions. However, most of them may fail if one of these conditions is violated. For character segmentation, vertical and horizontal projection are commonly used in the papers. These projection methods achieved a high accuracy. Yet, they both fail if the viewing angle is wide or the image is noisy. For instance, if the LP image is tilted, then the angle will be large, and this can lead to segmenting a single character as two or more characters. To overcome this problem, aspect ratio constraint is used [5]. By setting a threshold for the ratio between the height and width, plate characters can be extracted accurately. Fast real-time processing is one of the most crucial aspects of LPR systems. Nonetheless, only a few papers dealt with it. [1] and [6] designed a LPR system that recognizes a plate character in 30ms and 0.38ms respectively. Whereas [7] detected all license plate characters in mere 25ms. This is considered to be convenient for real-time applications. Designing a CNN that performs well under different illumination conditions is an attracting feature of LPR systems. [3], [6], [7] and [10] were the only ones that trained their networks on low and high resolution images. This made their network robust against various lighting conditions. Training the network on a small or an imbalanced data set is an another issue that some papers did not consider. It is recommended that the network is trained on an equal number of sample images for each character. Otherwise, the network could be biased and inaccurate. Training the CNN on a large number of sample images is generally required, but this can increase the computational cost and introduce overfitting. Therefore, the number and type of training images should be carefully chosen.

VI. Conclusion

In this paper, a careful study on different LPR systems based on convolutional neural networks has been demonstrated. The examined CNN systems show a good performance under normal environmental conditions. However, they are still very sensitive to a viewing angle and low illumination. Future investigation therefore should be directed in this regard.

References

[1] H. Li, R. Yang, and X. Chen, "License plate detection using convolutional neural network," in *Computer and Communications (ICCC), 2017 3rd IEEE International Conference on*. IEEE, 2017, pp. 1736–1740.

[2] C.-H. Lin, Y.-S. Lin, and W.-C. Liu, "An efficient license plate recognition system using convolution neural networks," in *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE, 2018, pp. 224–227.

[3] P. Dhar, S. Guha, T. Biswas, and M. Z. Abedin, "A system design for license plate recognition by using edge detection and convolution neural network," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*. IEEE, 2018, pp. 1–4.

[4] P. Rajendra, K. S. Kumar, and R. Boadh, "Design of a recognition system automatic vehicle license plate through a convolution neural network," *International Journal of Computer Applications*, 2017.

[5] S. Saunshi, V. Sahani, J. Patil, A. Yadav, and S. Rathi, "License plate recognition using convolutional neural network," *IOSR Journal of Computer Engineering (IOSR-JCE)*, 2017.

[6] V. H. Pham, P. Q. Dinh, and V. H. Nguyen, "Cnn-based character recognition for license plate recognition system," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2018, pp. 594–603.

[7] Q. Wang, "License plate recognition via convolutional neural networks," in *Software Engineering and Service Science (ICSESS), 2017 8th IEEE International Conference on*. IEEE, 2017, pp. 926–929.

[8] G. Kosala, A. Harjoko, and S. Hartati, "License plate detection based on convolutional neural network: Support vector machine (cnn-svm)," in *Proceedings of the International Conference on Video and Image Processing*. ACM, 2017, pp. 1–5.

[9] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Automatic license plate recognition with convolutional neural networks trained on synthetic data," in *Multimedia Signal Processing (MMSP), 2017 IEEE 19th International Workshop on*. IEEE, 2017, pp. 1–6.

[10] H.-H. Kim, J.-K. Park, J.-H. Oh, and D.-J. Kang, "Multi-task convolutional neural network system for license plate recognition," *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2942–2949, 2017.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.