Computers, Materials & Continua
DOI:10.32604/cmc.2022.020110
Article

Tech Science Press

Deep Learning Based License Plate Number Recognition for Smart Cities

T. Vetriselvi[1], E. Laxmi Lydia[2], Sachi Nandan Mohanty[3,4], Eatedal Alabdulkreem[5], Shaha Al-Otaibi[6], Amal Al-Rasheed[6] and Romany F. Mansour[7,*]

[1]Department of Computer Science and Engineering, K. Ramakrishnan College of Technology, Trichy, 620002, India
[2]Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, 530049, India
[3]Department of Computer Science & Engineering, Vardhaman College of Engineering (Autonomous), Hyderabad, 501218, India
[4]Department of Informatics and Computing, Singidunum University, Serbia
[5]Computer Science Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, 11564, Saudi Arabia
[6]Information Systems Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, 11564, Saudi Arabia
[7]Department of Mathematics, Faculty of Science, New Valley University, El-Kharga, 72511, Egypt
[*]Corresponding Author: Romany F. Mansour. Email: romanyf@scinv.au.edu.eg

Received: 09 May 2021; Accepted: 14 June 2021

Abstract: Smart city-aspiring urban areas should have a number of necessary elements in place to achieve the intended objective. Precise controlling and management of traffic conditions, increased safety and surveillance, and enhanced incident avoidance and management should be top priorities in smart city management. At the same time, Vehicle License Plate Number Recognition (VLPNR) has become a hot research topic, owing to several real-time applications like automated toll fee processing, traffic law enforcement, private space access control, and road traffic surveillance. Automated VLPNR is a computer vision-based technique which is employed in the recognition of automobiles based on vehicle number plates. The current research paper presents an effective Deep Learning (DL)-based VLPNR called DL-VLPNR model to identify and recognize the alphanumeric characters present in license plate. The proposed model involves two main stages namely, license plate detection and Tesseract-based character recognition. The detection of alphanumeric characters present in license plate takes place with the help of fast RCNN with Inception V2 model. Then, the characters in the detected number plate are extracted using Tesseract Optical Character Recognition (OCR) model. The performance of DL-VLPNR model was tested in this paper using two benchmark databases, and the experimental outcome established the superior performance of the model compared to other methods.

Keywords: Deep learning; smart city; tesseract; computer vision; vehicle license plate recognition

1 Introduction

There is a tremendous increase in the usage of vehicles in recent years, thanks to rapid economic growth of the country. In smart cities, road safety can be achieved for people through automated VLPNR process. VLNPR makes a significant gain in real-time under several aspects. It is useful in several applications like automated toll fee collection systems [1], car parking access controls [2] and road traffic control [3]. VLPNR is an active research domain that received more attention in the recent years. Various applications have been developed recently deploying intelligent transportation and surveillance systems along with the enhancement of digital camera and increased computation complexity. These systems are intended to recognize vehicles using their number plates. Such systems offer automated identification and recognition of vehicle license plates from real-time images. After a vehicle's front view is captured using a camera, the captured image is fed into computer vision-based algorithms as input to examine, identify, and filter the plate areas from backdrop. The identification process performs the character segmentation in detected area followed by its recognition. Identification and recognition of number plates are two different tasks. Various models exist for a particular kind of number plate style (font size, text, font type, backdrop color, and shape) or for particular conditions like motion of camera, angle, lighting, occlusion, and so on.

Classical VLPNR models utilize Machine Learning (ML) models especially its hand-crafted features to represent the essential features that exist in vehicle license plate image. These models gather some morphological variables and are susceptible to the presence of noise in image and complicated backdrop. DL models offer an option i.e., automated feature selection from images with the help of learned representations of underlying data using altered filters. Convolutional Neural Networks (CNNs) is one of the advanced and effective DL models which gained significant attention in the recent years in various fields of computer vision like handwriting recognition [4], text recognition [5], visual object recognition, etc. Since identifying the location of a vehicle license plate is treated as a detection problem, diverse region-based CNN models can be employed to detect the objects in a rapid and precise manner. For VLPNR, the available DL-based models are segregated into two types namely, segmentation-based and segmentation-free models. The former model carries out the segmentation task for character separation and recognition of individual characters. The latter model, on the other hand, recognizes the characters without separation using particular architectural models like Recurrent Neural N(RNN).

In detection process, one of the processes is the localization of bounding boxes of vehicle license plate from the complete actual input image. The outcome of the process affects the accuracy of the detection process significantly. Several VLPNR models have been proposed and implemented in the literature. The classical ML models, with handcrafted features, depend on a particular set of descriptors like edges, color, and texture descriptors [6]. Vehicle license plate recognition is a process of detecting a homogeneous text region through the detection of characters straightaway from the image [7].

Though it is simple and rapid, the existing models produced minimum detection rate, since the features learned from the characters are not sufficient to identify every character present in the image. Besides, the other characters present in the image create confusion in detecting the vehicle license plate. The existing models assume the license plate as an area with high contrast and edge density or otherwise as a portion that is comprised of high intensity key points identified with Scale-Invariant Feature Transform (SIFT) descriptor [8].

Currently, DL-based models are in use to localize the vehicle license plates. Especially, A4-layer CNN-based models are in use to detect the text regions present in input image. Afterward, a second 4-layer plate/non-plate CNN classification model is applied to differentiate the vehicle license plates from typical textual characters. The authors, in the literature [9], utilized a classification model using FAST-YOLO network in the detection of front view of cars from the applied image. From this, the vehicle license plate details are extracted from the identified front view image. The literature [10] employed a pipeline architecture using a series of deep CNNs to detect vehicle license plate under diverse scenarios. The architecture operates around a set of different number plate designs. However, it is designed mainly for Arabian text due to which it cannot be applied for other languages.

The current research paper presents an effective DL-based VLPNR method to identify and recognize the alphanumeric characters, present in the license plate. The proposed model involves two main stages namely, license plate detection and Tesseract-based character recognition. License plate detection occurs with the help of Faster RCNN and Inception V2 model. Then, the characters in the detected number plate are extracted by Tesseract OCR model. The study validated the performance of DL-VLPNR model utilizing a set of two benchmark databases. The experimental outcomes established the optimal performance of the proposed method over compared techniques.

The upcoming sections of the paper are as follows. Section 2 briefs the works related to VLPNR model. Section 3 discusses the presented DL-VLPNR model. The validation of the proposed DL-VLPNR model is presented in Section 4, and the paper is concluded in Section 5.

2 Literature Survey

Segmentation-dependent models extract every individual character from vehicle license plate in its earlier stage. Afterwards, the OCR algorithm recognizes every character from the extracted image. The existing models on vehicle license plate image segmentation are of two different types such as projection-based and connected component-based. Between these methods, the former one makes use of the characters and backdrops that are different in color in number plate and the method offers contrary values in binary image. The histograms of vertical and horizontal pixel projections could be utilized in the segmentation of characters [11]. These models can be easily influenced by rotating the vehicle license plate. The connected component-based model segments the characters by labeling every linked pixel in binary image to components. Though it is robust to rotate, it fails in proper segmentation of characters, once they are combined or divided. After the segmentation of characters, recognition process occurs as classification, with an individual class for every alphanumeric character.

The existing techniques perform partition in two ways namely, template matching and learning-based techniques. The former one comprises of similarity comparison of a provided character against the template. In this method, the high resemblance character is chosen. Various similarity metrics are presented for instance, Mahalanobis and Hamming distances [12]. These are employed in binary images and are restricted since it operate only for original character size and font. It does not support rotating or broken letters. The latter model is highly robust and operates with characters of different sizes, fonts, and rotation. It makes use of ML models in differentiating the characters with the help of one or many features like edges, gradient, and SIFT. In the study conducted earlier [13], a 5-layer CNN was used to recognize the Malaysian vehicle license plate where every character undergoes manual extraction and segmentation. VLPNR process is treated as a classification process, including 33 classes. It has the capability to achieve a maximum accuracy of 98.79% on a limited sample count. A CNN-based model was introduced in the study conducted earlier [14] for VLPNR which used several preprocesses like filtering, thresholding, and segmentation.

In segmentation-free models, VLPNR is carried out on global vehicle license plate images with no character segmentation. Generally, a sliding window is utilized over an input image for the generation of several tentative characters in few steps. Then, every tentative character is, utilized by a recognition model. Once the input image is completely swiped by the sliding window, the predicted output is investigated, and the end sequence is decided. The successive identical characters are treated as a single character whereas the character space is applied for separation. Under VLPNR, some of the models use segmentation-free approaches via DL approaches. A CNN model was proposed in the literature [15] to obtain the features on vehicle license plate and RNN so as to find the series of characters. In the study conducted earlier [16], a VLPNR was proposed in the recognition of license characters as sequence labeling problem by RNN with Long Short-Term Memory (LSTM). A deep (16-layers) CNN was used in the study [17] depending on Spatial Transformer Networks to carry out a lesser sensitive character identification in spatial conversions on entire license plate image. This model is also used to avoid the crucial process of segmenting the image into characters. A YOLO-based network was proposed for VLPNR using an integrated classification-detection model. In the literature [18], a CNN model was utilized to identify the characters in vehicle license plate and localize the character bounding box corners. It dealt with classification process of a set of 33 classes for Italian VLPNR. Generally, it is observed that the DL models for VLPNR are still under progress and are limited to particular scenarios. Few of the models discussed above have performed VLPNR in a dedicated way while many models were based on hand-crafted features.

3 The Proposed DL-VLPNR Model

The working process involved in the proposed DL-VLPNR method is depicted in Fig. 1. The proposed DL-VLPNR model has two main stages. Number plate detection occurs with the help of Faster RCNN and Inception V2 model. Then, the characters in the detected number plate are extracted by Tesseract OCR model. At this point, the Tesseract OCR engine is used to realize the alphanumeric features present in the detected plate. The applied Tesseract engine is trained to improve the accuracy of analysis. Training process involves the development of characters in the images which need to be predicted using the desired fonts. Further, a dictionary of viable characters should be identified in number plate which has other information such as regional codes, suffixes as well as registration numbers. The result of this phase arrive at the text representation of vehicle number.

3.1 Faster R-CNN with Inception V2 Model for Number Plate Detection

Faster R-CNN method has two major stages such as Region-Based Proposals (RPN) and Fast R-CNN technique. When RPN is constrained with reliable feature rules, then the Fast R-CNN model explores the objects. The identification outcome is provided to RPN to generate the region proposals. Faster R-CNN approach obtains the whole image and the value of object proposals in the form of input in order to forecast the abnormalities that exist in the input image as shown in Fig. 2 [19].

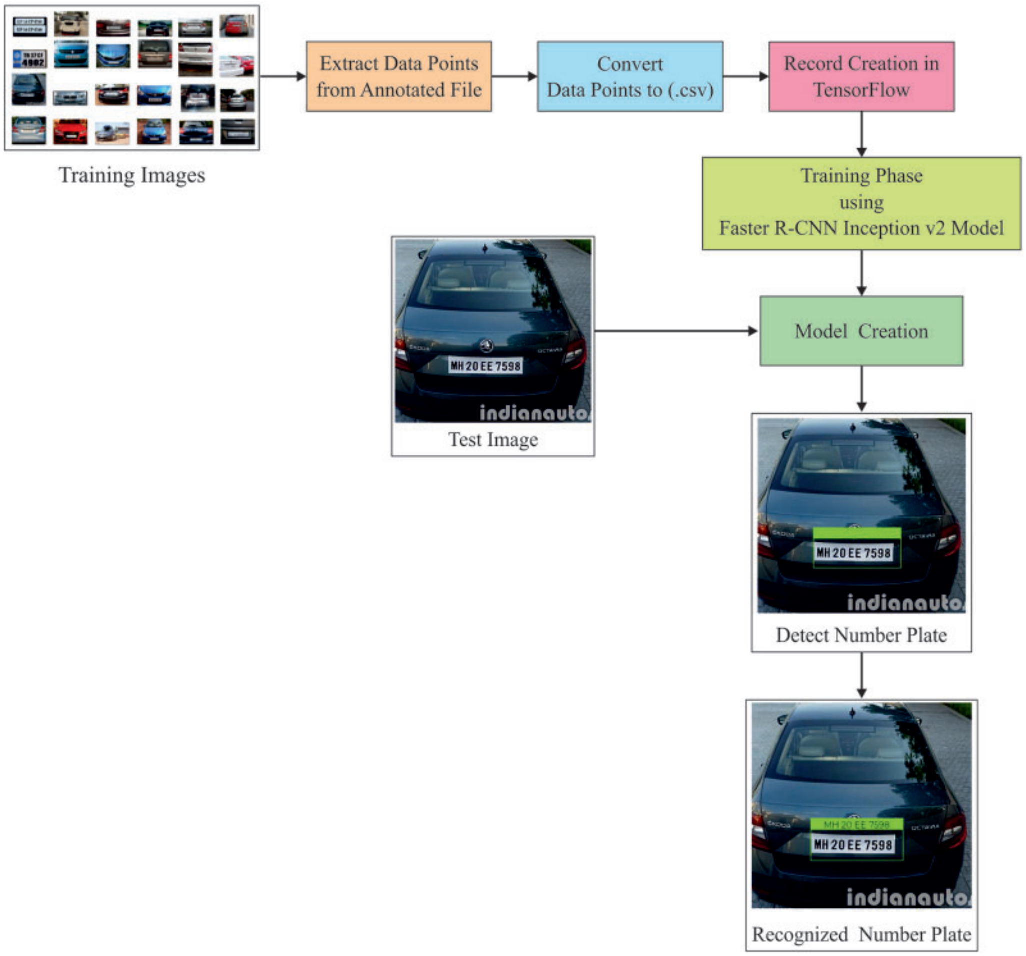
Training Images
Extract Data Points from Annotated File
Convert Data Points to (.csv)
Record Creation in TensorFlow
Training Phase using Faster R-CNN Inception v2 Model
Test Image
Model Creation
Detect Number Plate
Recognized Number Plate
MH 20 EE 7598
indianauto.

Figure 1: The workflow of DL-VLPNR model

Faster R-CNN characteristics are determined to filter the association, whereas, in second phase, the class labeling function is carried out. As a result, the class labels are declared for all the observed regions in a video frame. Later, the anomaly is predicted. A group of frames obtained from a video sequence of the tracked objects acts as the input for anomaly detection. A characteristic of Faster R-CNN performs this operation and maps the observed regions. When identifying the observed regions of a frame, the corresponding labels are assigned with respective prediction values.

RPN receives images of diverse sizes and offers the results as a group of rectangular object proposals along with specific objectless value. It is named after CNN and the main theme of this model is to allocate the processing with Fast R-CNN. To produce the region proposals, a tiny network is slid across a convolutional (Conv) feature map. It consumes the input as $n \times n$ spatial window. Each sliding window undergoes mapping with minimum dimensional feature and provides two fully connected layers.

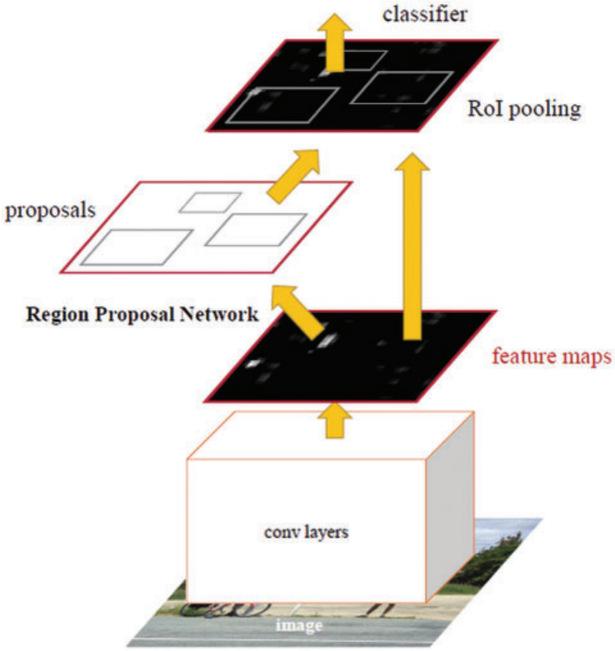

Figure 2: The faster RCNN model

RPN involves in a ranking process that ranks the anchors. The anchors are highly significant in the implementation of Faster R-CNN. An anchor is defined as a box whereas the anchors are provided at the image position.

Here, the RPN is trained by assigning a binary class for all the anchors. A positive label is allocated for two anchors such as anchors with high Intersection-over-Union (IoU) that overlaps a ground-truth box and anchor with 0.7 IoU. A ground-truth box is shared for positive labels. In most of the cases, an alternate procedure is sufficient to find the positive samples, and primary criteria are applicable in rare scenarios.

Faster R-CNN focuses on reducing the objective function by applying multi-task loss in Fast R-CNN. Hence, it can be formulated as in Eq. (1):

$$L(\{p_u\}, \{z_u\}) = \frac{1}{N_{cls}} \sum_u L_{cls}(p_u, p_u^*) + \lambda \frac{1}{N_{reg}} \sum_u p_u^* L_{reg}(z_u, z_u^*) \tag{1}$$

where u denotes the index and p_u implies a predictive probability of anchor u, being an object. The ground-truth label p_u^* is one if the anchor is positive and 0 for negative anchor. z_u represents a vector of four parameterized coordinate points and z_u^* signifies a ground-truth box. The classification loss L_{cls} is log over two class labels. In case of regression loss, $L_{reg}(z_u, z_u^*) = R(z_u - z_u^*)$, where R refers to robust loss function (smooth L1). In term $p_u^* L_{reg}$ is a regression loss that is inactive state for $p_u^* = 1$ and in an inactive state for $p_u^* = 0$. The outcomes of *cls* and *reg* layers contain $\{p_u\}$ and $\{z_u\}$. The bounding box regression parameters are shown in Eq. (2):

$$\begin{aligned} &z_a = (a - b_x)/w_x, \quad z_b = (b - b_x)/h_x \\ &z_w = log(w)/w_x, \quad z_h = log(h)/h_x \end{aligned} \tag{2}$$

$$z_a^* = (a^* - a_x)/w_x, \quad z_a^* = (b - b_x)/h_x$$

$$z_w^* = log(w^*)/w_x, \quad z_h^* = log(h)/h_x$$

where a, b, w, and h are center coordinates of the box and its corresponding width and height values. The variables a, a_x, and a^* imply the detected box, anchor box, and ground truth box.

The outcomes attained from the simulated RPN gives the presented regions of diverse sizes. A different-sized region represents a different-sized CNN feature map. It is highly complex to develop an efficient method that can perform the features of different sizes. The Region of Interest (ROI) pooling undergoes simplification by reducing the feature maps to a similar size. Unlike Max-Pooling, the ROI pooling divides the input feature map to a predefined number into identical regions, and Max-Pooling is employed for all the regions. Thus, the result attained from ROI Pooling is assigned as k.

In order to detect the abnormalities in pedestrian walkways, Fast R-CNN method is applied. The strategy of the working process learns the conv layers that have been distributed from RPN and Fast R-CNN. RPN as well as Fast R-CNN are trained autonomously. Hence, Conv. layers can be changed in diverse modules. So, there is a requirement for development which allows the Conv. layers to be shared between two networks. This is executed by replacing the learning model that has been carried out in two distinct networks. It is very complex to describe the individual network of RPN and Fast R-CNN that undergoes optimization with the help of Back Propagation (BP) technique. The training process for Fast R-CNN depends on the predetermined object proposals.

A 4-step training model is applied in training the distributed features by other optimization models. Then, a predictive network undergoes training using the Fast R-CNN model, a derivative of RPN. Here, RPN and Fast R-CNN models are not capable of sharing the Conv. layers. The detection of networks are utilized in the third phase to initiate RPN training. Therefore, the distributed Conv. layers are permanent, and exclusive layers of RPN are fine-tuned. Consequently, the shared Conv. layers are provisioned from a constant full Conv. layer of Fast R-CNN that underwent fine-tuning. Therefore, RPN and Fast R-CNN share a similar Conv. layer and creates a unique system.

3.2 Inception V2 Model

Many developers from Google established an Inception network developed for ImageNet competition to classify and predict the challenges. The method consists of a fundamental component called 'Inception cell' to process a sequence of Conv. layers at diverse scales and consecutively assemble the simulation outcome. To save the process, 1 × 1 Conv. has been applied to decrease the input channel depth. For every cell, a collection of 1 × 1, 3 × 3, and 5 × 5 filters are applicable to extract the features from input at various scales. Also, Max pooling is employed, albeit with 'same' padding to save the dimensions. So, both of these could be combined appropriately. Inception network plays a significant role in the development of CNN classification models. Before inception network, well-known CNNs are stacked with Conv. layers at a high depth to attain a better function.

3.3 OCR Engine: Tesseract

The pipeline of the Tesseract OCR engine is shown in Fig. 3. Initially, Adaptive Thresholding is applied to change the image into binary version using Otsu's method. Page layout analysis is the next step and is applied in extracting the text blocks within the region. Then, the baselines of every line are detected and the texts are divided into words with the application of finite spaces as well as fuzzy spaces.

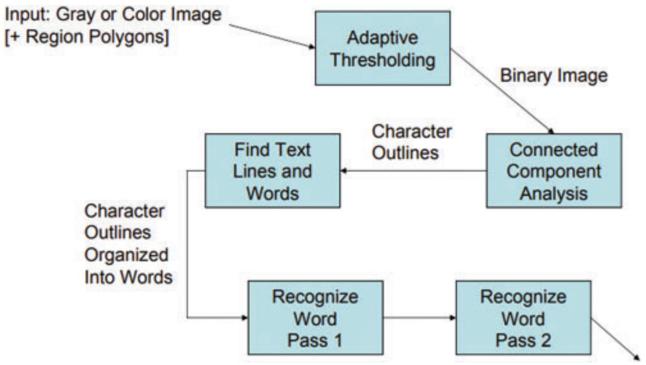
Input: Gray or Color Image
[+ Region Polygons]
Adaptive Thresholding
Binary Image
Connected Component Analysis
Character Outlines
Find Text Lines and Words
Character Outlines Organized Into Words
Recognize Word Pass 1
Recognize Word Pass 2

Figure 3: Processes involved in tesseract-based character recognition

In the next step, the character outlines are extracted from the words. Text recognition is initiated as a 2-pass method. In the first pass, word recognition is carried out with the application of static classification. Every word is passed satisfactorily to adaptive classifier in the form of training data. A second pass is run over the page by employing a novel adaptive classification model, in which the words are not examined thoroughly to re-examine the module.

A series of processes involved in the implementation of the presented method is summarized herewith.

- Initially, a collection of training images is provided.
- In the next stage, a set of data points is obtained from the available annotated image. Next, the conversion of data points to .csv file takes place.
- Then, the records are generated in TensorFlow. In the next stage, a training model is created using Faster R-CNN with Inception V2 method.
- Upon the completion of training model, new input images are provided to the system as shown in the figure.
- When new input images are provided, the Faster RCNN model detects the number plate at first instance correctly.
- Then, the text is recognized with the help of PyTesseract.
- Finally, the text in the vehicle number plate is identified correctly.

4 Performance Validation

4.1 Implementation Details

The proposed DL-VLPNR model was simulated using a PC i5, 8th generation, 16 GB RAM. The DL-VLPNR model was programmed using Python language with TensorFlow, Pillow, OpenCV, and Py Tesseract.

4.2 Results Analysis

Fig. 4 shows the visualization outcomes of the proposed DL-VLPNR model. It is inferred that the DL-VLPNR model can clearly recognize the license plate number on all the images. Fig. 4 shows that the presented model accurately recognized the number plate, even it is inappropriately captured. This illustrates the reliable function of the developed system under different circumstances.

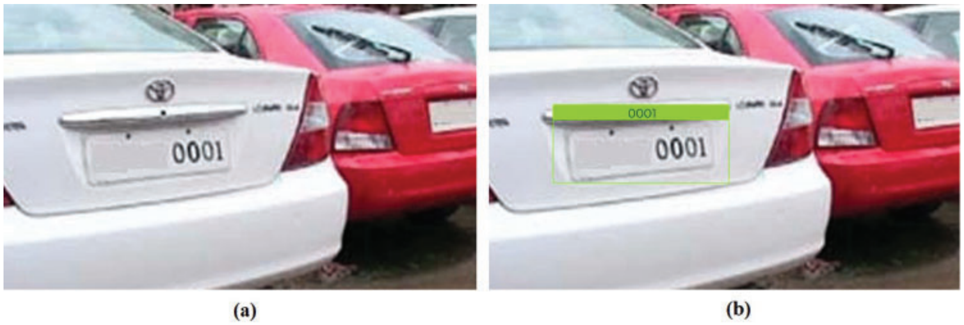

Figure 4: Sample recognition output

Table 1: Performance analysis of proposed model on FZU cars dataset

No. of Runs	Precision	Recall	F1-score	mAP
Run-1	0.9756	0.985	0.9754	0.9849
Run-2	0.9754	0.9742	0.9775	0.9655
Run-3	0.9673	0.9868	0.9554	0.9622
Run-4	0.9787	0.9751	0.9708	0.9749
Run-5	0.9741	0.9754	0.9839	0.9673
Run-6	0.9658	0.9776	0.9736	0.9636
Run-7	0.9876	0.9849	0.9838	0.9735
Run-8	0.9903	0.9771	0.9766	0.967
Run-9	0.9875	0.9965	0.9771	0.9641
Run-10	0.9783	0.9875	0.9859	0.9672
Average	0.9780	0.9820	0.9760	0.9690

Tab. 1 and Fig. 5 shows the results of the analysis of the proposed model on FZU Cars dataset under different runs [20,21]. From the obtained results, it is apparent that the proposed model can demonstrate the maximum recognition in terms of precision, recall, F1-score, and mAP. The proposed model attained a high average precision of 0.9780, recall of 0.9820, F1-score of 0.9760, and mAP of 0.9690.

Tab. 2 and Fig. 6 show the results offered by different VLPNR models on the applied FZU Cars dataset. The table values indicate that the ZF model produced ineffective detection outcomes with minimum precision, recall, F1-score, and mAP values i.e., 0.916, 0.948, 0.932, and 0.908 respectively. At the same time, it is pointed that the VGG16 model performed well than the previous model and attained slightly higher precision, recall, F1-score, and mAP values such as 0.925, 0.955, 0.940, and 0.912 respectively.

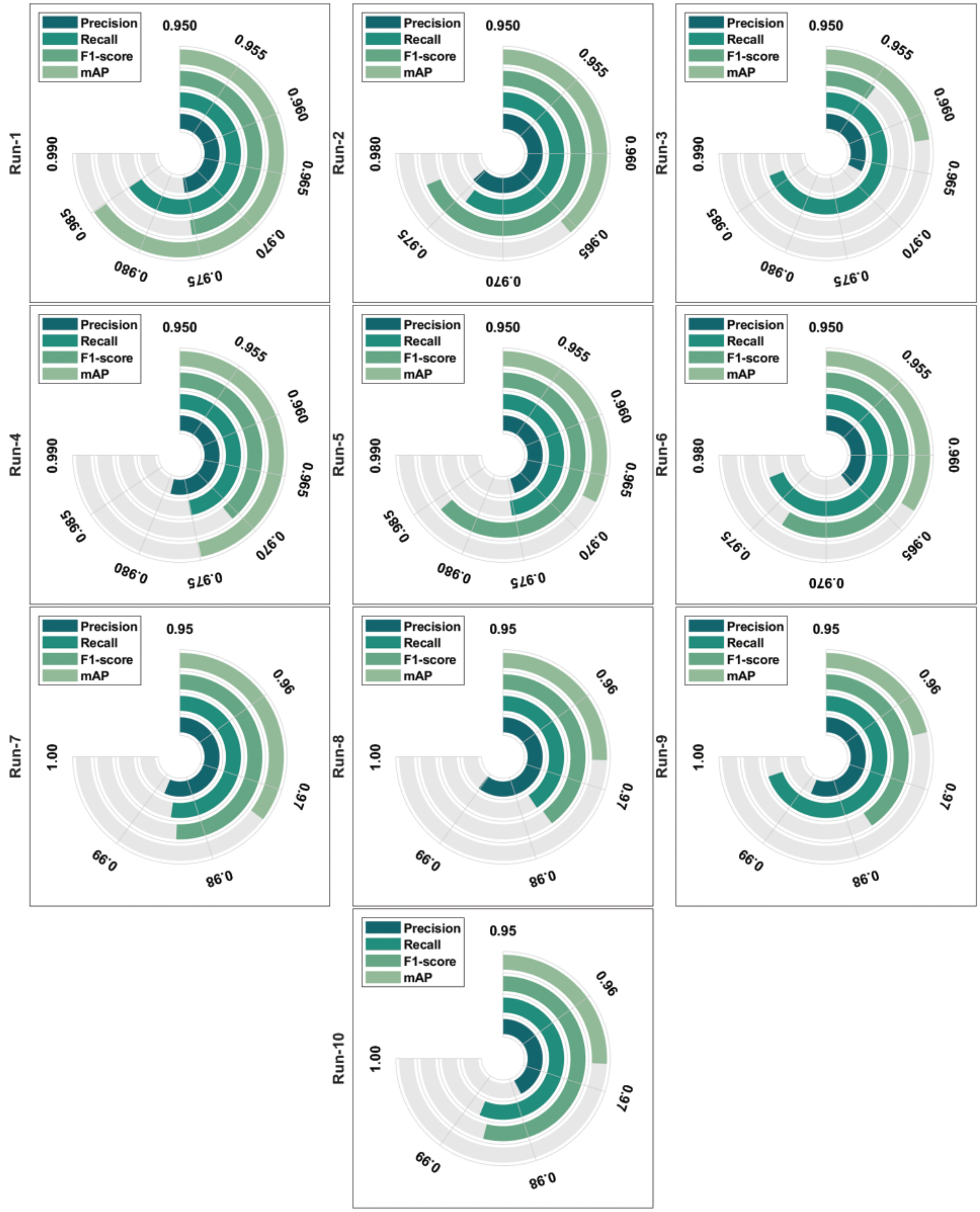

Figure 5: Performance analysis of DL-VLPNR model on FZU cars dataset

Table 2: Comparative study of existing with presented DL-VLPNR for FZU cars dataset

Model	Precision	Recall	F1-score	mAP
ZF	0.916	0.948	0.932	0.908
VGG16	0.925	0.955	0.940	0.912
ResNet50	0.938	0.951	0.944	0.916
ResNet101	0.945	0.958	0.951	0.922
DA-Net136	0.961	0.964	0.962	0.942
DA-Net160	0.965	0.966	0.965	0.952
DA-Net168	0.966	0.968	0.967	0.955
DA-Net200	0.969	0.971	0.970	0.958
DL-VLPNR	0.978	0.982	0.976	0.969

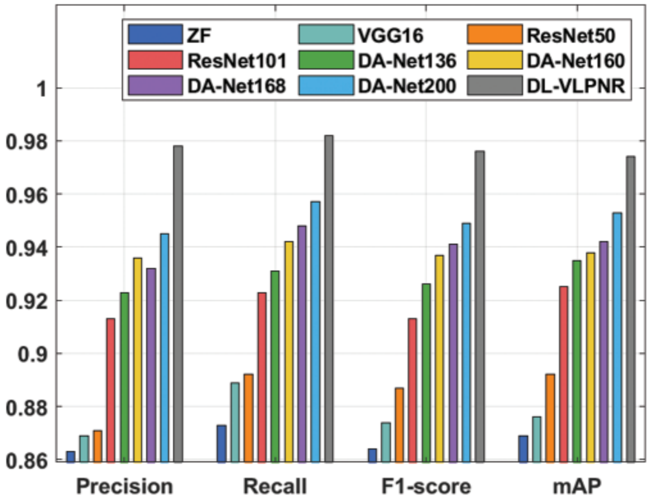

Figure 6: Comparative detection result analysis of DL-VLPNR model on FZU cars dataset

Along with that, the ResNet 50 model achieved even higher detection outcomes with precision, recall, F1-score, and mAP values such as 0.916, 0.948, 0.932, and 0.908 respectively. In line with this, it is observed that the ResNet 101 model produced an acceptable recognition with precision, recall, F1-score, and mAP values such as 0.945, 0.958, 0.951, and 0.922 respectively. Besides, the DA_Net136, DA_Net160, DA_Net168, and DA_Net200 approaches produced competitive and near similar recognition rates over the compared methods. The DA-Net136 model showed a slightly manageable outcome with precision, recall, F1-score, and mAP values such as 0.961, 0.964, 0.962, and 0.942 respectively. Next, the DA-Net160 offered slightly higher precision, recall, F1-score, and mAP values such as 0.965, 0.966, 0.965, and 0.952 respectively. Afterward, even higher performance was achieved by DA-Net168 with precision, recall, F1-score, and mAP values such as 0.966, 0.968, 0.967, and 0.955 respectively.

In line with this, the DA-Net200 model produced near-optimal results with precision, recall, F1-score, and mAP values such as 0.978, 0.982, 0.976, and 0.969 respectively. However, the proposed DL-VLPNR model accomplished the optimal performance with precision, recall, F1-score, and mAP values beings 0.978, 0.982, 0.976, and 0.969 respectively.

Tab. 3 and Fig. 7 show the results for detection analysis of the proposed model on HumAIn2019 dataset under different runs. From the resultant values, it is obvious that the proposed model exhibited an improved outcome in terms of precision, recall, F1-score, and mAP. The proposed model accomplished higher average precision of 0.9780, recall of 0.9820, F1-score of 0.9760, and mAP of 0.9740.

Table 3: Performance analysis of the proposed model on HumAIn2019 dataset

No. of Runs	Precision	Recall	F1-score	mAP
Run-1	0.9658	0.9851	0.9812	0.9666
Run-2	0.9885	0.9654	0.9688	0.9785
Run-3	0.9672	0.9886	0.9831	0.9732
Run-4	0.9647	0.9775	0.9751	0.9741
Run-5	0.9883	0.986	0.9635	0.9634
Run-6	0.9792	0.9896	0.9746	0.967
Run-7	0.9865	0.9753	0.9817	0.9854
Run-8	0.9763	0.988	0.978	0.9717
Run-9	0.9861	0.9763	0.9729	0.9858
Run-10	0.9774	0.9882	0.9811	0.9743
Average	0.9780	0.9820	0.9760	0.9740

Tab. 4 and Fig. 8 show the recognition performance achieved by diverse VLPNR models on the applied HumAIn2019 dataset. The table values denote that the ZF model produced the worst detection outcomes with minimum precision, recall, F1-score, and mAP values such as 0.863, 0.873, 0.864, and 0.869 respectively. At the same time, it is revealed that the VGG16 model is superior to ZF model as it achieved slightly higher precision, recall, F1-score, and mAP values such as 0.869, 0.889, 0.874, and 0.876 respectively. Concurrently, the ResNet 50 model produced higher detection performance with precision, recall, F1-score, and mAP values being 0.871, 0.892, 0.887, and 0.892 respectively. Simultaneously, it is noticed that the ResNet 101 model produced a slightly satisfactory recognition with its precision, recall, F1-score, and mAP values being 0.913, 0.923, 0.913, and 0.925 respectively. Besides, the DA_Net136, DA_Net160, DA_Net168, and DA_Net200 methodologies accomplished somewhat satisfactory results. The DA-Net136 model exhibited slightly convenient outcomes with the precision, recall, F1-score, and mAP values such as 0.923, 0.931, 0.926, and 0.935 respectively. After that, the DA-Net160 offered slightly higher precision, recall, F1-score, and mAP values such as 0.936, 0.942, 0.937, and 0.938 respectively. An even more high performance was produced by DA-Net168 with precision, recall, F1-score, and mAP values such as 0.932, 0.948, 0.941, and 0.942 respectively. Concurrently, the DA-Net200 model produced near-optimal results with precision, recall, F1-score, and mAP values being0.945, 0.957, 0.949, and 0.953 respectively. However, the proposed DL-VLPNR model accomplished superior results over the earlier models in terms of precision, recall, F1-score, and mAP values being 0.978, 0.982, 0.976, and 0.974 respectively.

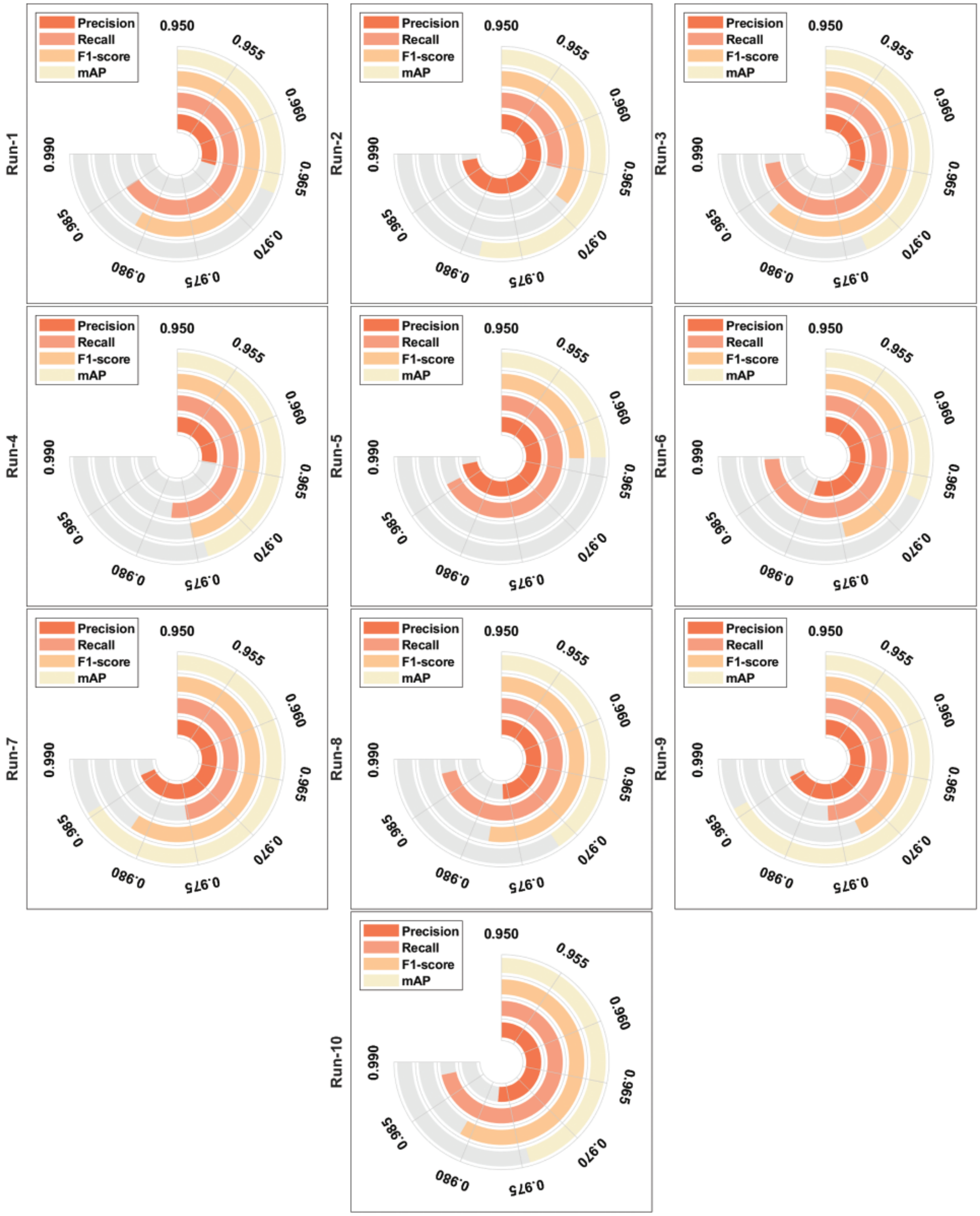

Figure 7: Performance analysis of DL-VLPNR model on HumAIn2019 dataset

Table 4: Comparative analysis of existing and the presented DL-VLPNR for HumAIn2019 dataset

Model	Precision	Recall	F1-score	mAP
ZF	0.863	0.873	0.864	0.869
VGG16	0.869	0.889	0.874	0.876
ResNet50	0.871	0.892	0.887	0.892
ResNet101	0.913	0.923	0.913	0.925
DA-Net136	0.923	0.931	0.926	0.935
DA-Net160	0.936	0.942	0.937	0.938
DA-Net168	0.932	0.948	0.941	0.942
DA-Net200	0.945	0.957	0.949	0.953
DL-VLPNR	0.978	0.982	0.976	0.974

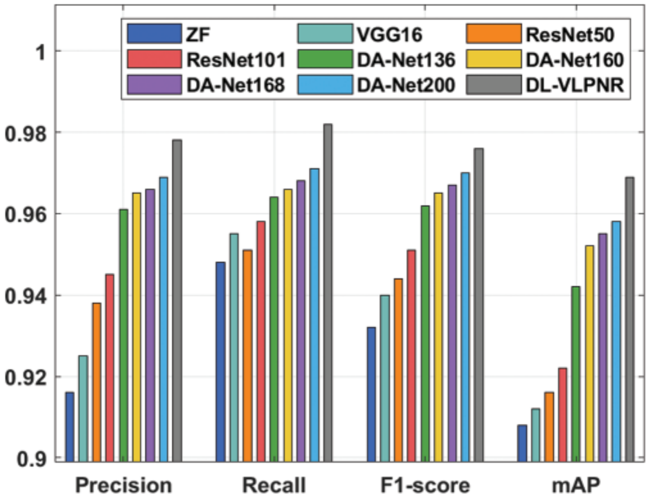

Figure 8: Comparative detection result analysis of DL-VLPNR model on HumAIn2019 dataset

Tab. 5 examines the overall accuracy analysis results offered by DL-VLPNR model with existing techniques on the applied dataset. It can be inferred that the presented DL-VLPNR technique leads in optimal recognition performance as it produced the highest accuracy of 0.986. At the same time, the VGG16 and ResNet 50 models produced near-identical and competitive outcomes with its accuracy values being 0.971 and 0.976 respectively. Along with that, the VGG_ CNN_M_1024 approach offered a somewhat low accuracy of 0.967, whereas minimum accuracy was achieved by ZF and ResNet 101 methods i.e., 0.942 and 0.943 respectively. Overall, the proposed DL-VLPNR methodology effectively recognized all the applied images compared to other methods.

Table 5: Accuracy analysis of diverse models

Methods	Overall accuracy
ZF	0.942
VGG16	0.971
VGG_CNN_M_1024	0.967
ResNet50	0.976
ResNet101	0.943
DL-VLPNR	0.986

5 Conclusion

The current research article proposed a productive DL-VLPNR model to identify and analyze the license plate characters of a vehicle. The proposed method utilizes Faster RCNN with Inception V2 model to detect the alphanumerical characters in license plate of a vehicle image. Afterward, the characters in the detected number plate are extracted by Tesseract OCR model. The performance of the DL-VLPNR model was validated using a set of two benchmark databases namely, FZU Cars and HumAIn2019 dataset. The results were analyzed in terms of different measures such as precision, recall, F1-measure, accuracy, and mAP. The experimental results evidently indicate that the DL-VLPNR model has the ability to achieve optimal detection and recognition performance as it attained the highest accuracy of 0.986. The proposed DL-VLPNR model can be employed as an appropriate tool for VLPNR. In future, the proposed model can be implemented in real-time traffic surveillance cameras in smart cities to identify the vehicles that cross the traffic signal at a faster rate.

Funding Statement: This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Suryatali and V. B. Dharmadhikari, "Computer vision based vehicle detection for toll collection system using embedded linux," in *2015 Int. Conf. on Circuits, Power and Computing Technologies*, Nagercoil, India, pp. 1–7, 2015.

[2] E. J. Sen, K. D. M. Dixon, A. Anto, M. V. Anumary, D. Mieheal, "Advanced license plate recognition system for car parking," in *2014 Int. Conf. on Embedded Systems*, Coimbatore, India, pp. 162–165, 2014.

[3] A. Naimi, Y. Kessentini and M. Hammami, "Multi-nation and multi-norm license plates detection in real traffic surveillance environment using deep learning," in *Int. Conf. on Neural Information Processing–ICONIP 2016: Neural Information Processing, Proc.: Lecture Notes in Computer Science LNCS*, New York City, NY, USA, vol. 9948, pp. 462–469, 2016.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[5] T. Wang, D. J. Wu, A. Coates and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. of the 21st Int. Conf. on Pattern Recognition*, Tsukuba, Japan, pp. 3304–3308, 2012.

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

[7] C. D. Nguyen, M. Ardabilian and L. Chen, “Robust Car license plate localization using a novel texture descriptor,” in *2009 Sixth IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Genova, Italy, pp. 523–528, 2009.

[8] H. S. Ng, Y. H. Tay, K. M. Liang, H. Mokayed and H. W. Hon, “Detection and recognition of Malaysian special license plate based on sift features,” *Computer Vision and Pattern Recognition*, pp. 1–7, 2015.

[9] S. Montazzolli and C. Jung, “Real-time Brazilian license plate detection and recognition using deep convolutional neural networks,” in *2017 30th SIBGRAPI Conf. on Graphics, Patterns and Images*, Niteroi, pp. 55–62, 2017.

[10] S. Z. Masood, G. Shu, A. Dehghan and E. G. Ortiz, “License plate detection and recognition using deeply learned convolutional neural networks,” *Computer Vision and Pattern Recognition*, pp. 1–10, 2017.

[11] J. M. Guo and Y. F. Liu, “License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1417–1424, 2008.

[12] C. Henry, S. Y. Ahn and S.-W. Lee, “Multinational license plate recognition using generalized character sequence detection,” *IEEE Access*, vol. 8, pp. 35185–35199, 2020.

[13] S. A. Radzi and M. K. Hani, “Character recognition of license plate number using convolutional neural network,” in *Visual Informatics: Sustaining Research and Innovations*, *Int. Visual Informatics Conf.*, *2011 Proc.: Lecture Notes in Computer Science LNCS*, New York City, NY, USA, vol. 7066, pp. 45–55, 2011.

[14] V. H. Pham, P. Q. Dinh and V. H. Nguyen, “CNN-based Character Recognition for License Plate Recognition System,” in *Asian Conf. on Intelligent Information and Database Systems ACIIDS 2018 Intelligent Information and Database Systems. Proc.: Lecture Notes in Computer Science LNCS*, New York City, NY, USA, vol. 10752, pp. 594–603, 2018.

[15] T. K. Cheang, Y. S. Chong and Yong Haur Tay, “Segmentation-free vehicle license plate recognition using convNet-rNN,” *Computer Vision and Pattern Recognition*, pp. 1–5, 2017.

[16] 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.

[17] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj *et al.*, “Deep automatic license plate recognition system,” in *Proc. of the Tenth Indian Conf. on Computer Vision, Graphics and Image Processing, in ICVGIP’16*, New York, NY, USA, pp. 6:1–6:8, 2016.

[18] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini and E. Magli, “Robust license plate recognition using neural networks trained on synthetic images,” *Pattern Recognition*, vol. 93, pp. 134–146, 2019.

[19] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-cNN: Towards real-time object detection with region proposal networks,” *Computer Vision and Pattern Recognition*, pp. 1–14, 2015.

[20] I. V. Pustokhina, D. A. Pustokhin, J. J. P. C. Rodrigues, D. Gupta, A. Khanna *et al.*, “Automatic vehicle license plate recognition using optimal k-means with convolutional neural network for intelligent transportation systems,” *IEEE Access*, vol. 8, pp. 92907–92917, 2020.

[21] T. Vaiyapuri, S. N. Mohanty, M. Sivaram, I. V. Pustokhina, D. A. Pustokhin *et al.*, “Automatic vehicle license plate recognition using optimal deep learning model,” *Computers*, *Materials & Continua*, vol. 67, no. 2, pp. 1881–1897, 2021.