

Research Article

An End-to-End Deep Learning Approach for Plate Recognition in Intelligent Transportation Systems

Jamshid Pirgazi , **Mohammad Mehdi Pourhashem Kallehbasti** ,
and **Ali Ghanbari Sorkhi** 

Department of Computer Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran

Correspondence should be addressed to Jamshid Pirgazi; j.pirgazi@mazust.ac.ir

Received 24 November 2021; Revised 22 December 2021; Accepted 28 December 2021; Published 24 January 2022

Academic Editor: Mohammad R Khosravi

Copyright © 2022 Jamshid Pirgazi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate and fast recognition of license plates is one of the most important challenges in the field of license plate recognition systems. Due to the high frame rate of surveillance cameras, old license plate recognition systems cannot be used in real-time applications. On the other hand, the presence of natural and artificial noise and different light and weather conditions make the detection and recognition process of these systems challenging. In this paper, an end-to-end method for efficiently detecting and recognizing plates is presented. In the proposed method, vehicles are first detected using a single-shot detector (SSD-) based deep learning model in the video frames and the input images. This will increase the speed and accuracy in identifying the location of the plate in the given images. Then, the location of the plate is identified using the proposed architecture based on convolutional networks. Finally, using a deep convolutional network and long short-term memory (LSTM), the characters related to the plate are recognized. An advantage of our method is that the proposed deep network is trained using different images with different qualities that leads to high performance in detecting and recognizing plates. Also, considering that in the proposed method the vehicles are first detected and then the plate is detected in the vehicle image, there is no limit in the number of identified plates. Moreover, plate detection in the vehicle rectangle, instead of the whole frame, speeds up our method. The proposed method is evaluated using several databases. The first part of the evaluation focuses on robustness and recognition speed. The proposed method has the accuracy of 100% for vehicle detection, 100% for plate detection, and 99.37% for character recognition. In the second part of evaluation, the proposed method is evaluated in terms of overall speed. The experimental results witness that the proposed method is capable of processing 30 frames per second without losing any data and also outperforms several methods proposed in recent years, in terms of time and accuracy.

1. Introduction

Due to the increasing number of vehicles, manually controlling and monitoring traffic is time-consuming, costly, inaccurate, and sometimes impossible. This makes automatic vehicle plate recognition a recurrent research topic. Since the plate is the unique ID of vehicles, several prominent applications are found for automatic plate recognition, including traffic control, driving offence detection, vehicle speed estimation, self-driving vehicles, and surveillance [1]. To this end, many cameras are installed in cities, roads,

highways, borders, parking lots, and protected areas for better and more accurate control of vehicles. These cameras are constantly monitoring the images of passing vehicles. Vehicles and their plates cannot be detected and recognized without processing and analyzing these images.

There is a need for a system based on image processing and machine learning to detect vehicles and extract other information such as plate number [2]. In vehicle plate detection and recognition systems, the quality of the input images has a direct impact on the result accuracy and processing speed. Many parameters are influential in the quality of the

captured images including environmental and weather conditions such as light projection angle, light intensity, rainfall, fog, dust, humidity, dark, glare, occluded, rainy, snowy, tilted, or blurred scenarios.

There are other concerns in automatic plate recognition systems, including different plate alignments on vehicles, size, plate aspect ratio, various shapes of plates, various angles of camera placement, too low or too high lighting, presence of several plates in the image, various plate background colors, excessive plate dirtiness, and various arrangements of letters and numbers in plates [3, 4]. As shown in Figure 1, a vehicle plate detection and recognition system consists of the three main steps: (i) plate detection, (ii) segmentation of characters in the plate, and (iii) character recognition [5].

The most important and challenging step is plate detection. If plates are not detected properly, the subsequent steps will not work as expected and the final result will be entirely wrong.

There have been many attempts to efficiently detect plates. In order to make this step more efficient, a phase of preprocessing is performed to denoise and improve image quality. The location of the plates is then identified in the input image. According to the previous studies on plate detection, existing approaches fall in five categories: edge-based methods, color-based methods, texture analysis methods, methods based on image global features, and hybrid methods. Different methods of image processing and machine vision have been used in each of these categories [6–11]. In the next step, plate segmentation is performed. In this step, the detected plate image is first converted to a binary image. Then, based on methods such as morphology [12], connected component analysis algorithm [13], and histogram-based methods, the parts related to characters are separated. The problem with this step is that most binarization methods have acceptable results only for clean plates. Using these methods on dirty plates leads to huge data loss. More precisely, some character parts of the plate may not be recognized as characters or some noncharacter parts may be recognized as characters. As can be seen in Figure 2, some noncharacter parts are considered as characters.

The next step in plate recognition systems is character recognition. This step is divided into two phases. In the first phase, different features are extracted from the segmented parts of the previous step. Then, the training of machine learning models is done based on the extracted features. In the feature extraction phase, various methods are proposed including active regions [14], HOG [15], horizontal and vertical mapping [16], and multiclass AdaBoost methods [17]. Some algorithms use key points locating methods such as SIFT [18] and SURF [19]. After creating the feature vector, classification is done for each section. In various works, artificial neural networks, support vector machine, Bayesian classifier, K nearest neighbor, etc. have been used for character recognition [20, 21]. Extracted features can be numerous and not all of them are useful for classification purposes. In other words, some features are redundant and irrelevant. The existence of large number of features makes the

machine learning models prone to overfitting. To address this problem, feature selection and dimension reduction are performed in some works.

Nowadays, due to the massive production of labeled data in different phases and increasing computing power, the use of deep learning methods has received considerable attention. Some works propose end-to-end methods for plate detection and recognition. In these works, only one or two of these steps are performed using deep learning methods due to lack of required data to train the deep learning model.

Gou et al. [22] proposed a method based on character-specific extremal regions and hybrid discriminative restricted Boltzmann machines (HDRBMs). Vertical edge detection, morphological operators, and different evaluations have been used for top-hat transfer plate detection. Specific character regions are identified as candidate regions for plate characters. Then, a trained model of HDRBM is used for character recognition. Their proposed method is resistant to changes in light intensity and weather conditions.

Wang et al. [23] proposed a multifunctional convolutional neural network (CNN) to detect and recognize Chinese vehicle plates with better accuracy and lower computational cost. In their proposed method, the detection network consists of three layers (P-Net, R-Net, and O-Net) and also a fully connected layer. The output of this part is the input of the detection network. In this part, the features are normalized and then the plate characters are recognized using a CNN. Wen-bin et al. [24] used an improved convolutional recurrent neural network to recognize Chinese plates. In this method, deep CNNs, recurrent neural networks (RNNs), spatial transformer networks, and connectionist temporal classification model are combined. In this method, there is no need for plate segmentation that causes erroneous plate detection.

Li et al. [25] proposed a plate recognition method that is a combination based on deep neural networks. In this method, recurrent neural networks with LSTM were used for plate feature extraction. The extracted sequence features in this part are given to a 37-class CNN for character detection. The advantage of using this method is that there is no need for plate segmentation.

The differences between the character features in terms of width, height, distance, and presence of noise such as heavy shadows, uneven light, different optical geometries, and poor contrast are challenging in plate recognition systems. Bulan et al. [26] used a two-stage classification method plate detection. For this purpose, the plate candidate points are first classified using a weak classifier, and then, the plate main points are detected using a strong classifier based on deep convolutional networks. After that, using the ALexnet architecture, features are extracted from the regions extracted from the previous step, and using support vector machine classification, the character detection is done based on the features obtained from the previous step.

In [27], an end-to-end method is proposed to detect and recognize vehicle plates. In this method, features are first extracted from the input image using CNNs. Plate regions are then identified based on the deep recurrent network.

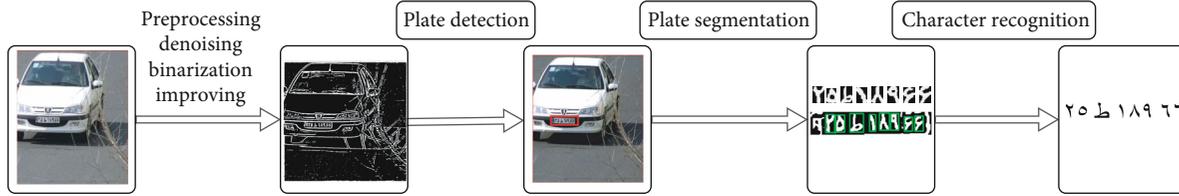


FIGURE 1: General steps of the vehicle plate recognition and detection system.

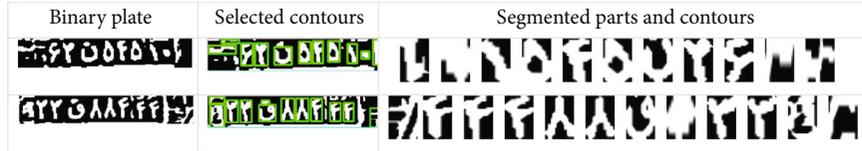


FIGURE 2: General steps of the vehicle plate recognition and detection system.

After that, the plate characters are recognized using convolutional and recurrent networks. Performing these steps makes the model to perform with acceptable speed.

In [28], fully convolutional networks are combined with a broad learning system. A fully convolutional network, which has been designed as a two-stage classification method at the pixel level, is used to detect objects by combining multiscale and hierarchical features. The AdaBoost cascade classifier was used to classify characters, and an extensive learning system was used to recognize characters.

Chen et al. [29] proposed a method based on CNNs for plate detection. In this method, an end-to-end network is proposed that simultaneously detects vehicles and plates. Different convolutional layers have been used in this method. Gao et al. [30] proposed an end-to-end network for plate detection and recognition based on encoder and decoder. The efficiency of traditional plate detection methods is affected by several factors including light intensity, shadow, and complex background. Using deep learning methods, plate recognition algorithms can extract deep features and improve the plate detection and recognition rate [31].

An efficient shared adversarial training network has been proposed for plate detection in [32]. This model can learn environment independent semantic features without perspective from real plates using prior knowledge of standard plates.

Authors in [33–35] considered using YOLO3 architecture for plate detection and recognition. In [33], YOLO3 is used for Brazilian plate detection and a three-layer convolutional network is used for character recognition. Authors in [34] used a two-stage convolutional layer in YOLO3 architecture in order to extract temporal features from plates and to reduce false positive detection rate. Authors in [35] used seven convolutional layers for plate detection and character recognition.

In the current paper, due to the massive production of labeled data in different parts, an end-to-end method for plate detection and recognition is proposed. Vehicles are first detected using the SSD-based deep learning model in the video frames and the input images. This increases the

speed and accuracy in identifying the location of the plate in the given image. Vehicle detection in the proposed method makes it suitable for smart transportation systems. In this mode, traffic volume measurement can be done based on vehicle detection. Moreover, vehicle movement direction and speed and also stopped vehicles in highways can be detected. In the next step, the location of the plate is identified using the proposed architecture based on convolutional networks. Finally, the characters related to the plate are detected and recognized using the deep convolutional network and LSTM.

An advantage of our approach is that the proposed deep network is trained with different images with different qualities that leads to high performance in detecting and recognizing plates. Apart from that, the vehicles are first detected and then the plates are detected in the vehicle images, instead of the whole image that speeds up our method. Moreover, there is no limit in the number of identified plates.

In real-time systems, analysis is performed on video frames captured by monitoring cameras. These cameras capture between 30 and 90 frames per second. Conventional systems can process 1 to 2 frames per second, while the proposed method is capable of processing 30 frames per second without losing any data. The rest of this paper is organized as follows. Section 2 presents the proposed method and describes its architecture in detail. Section 3 describes the experimental evaluation and compares the proposed method to other methods. Finally, Section 4 concludes the paper.

2. Proposed Method

Most of plate recognition methods fail to detect all the plates when there are many of them in the frame being processed. This makes these methods inapplicable for real-time situations when there can be many plates in every frame and the response time is limited. To address this issue, an SSD architecture is used in the proposed approach to make the processing speed fast enough for real-time plate recognition. In the proposed method, first vehicles are detected. Then, based on the deep learning architecture, in two phases, plates

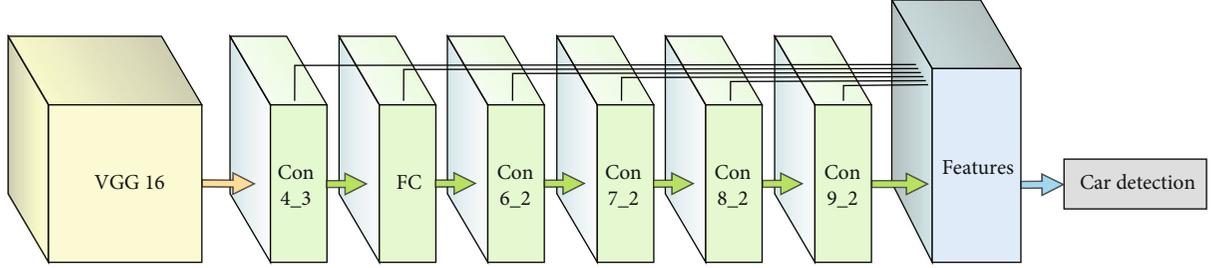


FIGURE 3: The deep architecture used in the proposed method for vehicle detection.

are first detected and then the text of the plates are extracted using a deep network with convolutional layers and recurrent layers. The proposed method is an end-to-end method that detects vehicles and plates simultaneously and recognizes the plates. The following sections explain different steps of this method.

2.1. Vehicle Detection. In order to achieve a better detection rate in real-time applications, it is better to start the process with vehicle detection. Conventional systems based on deep learning look for plate in the whole frame. Since plates compose a very small part of a frame, this process makes the existing approach slow. In the proposed method, vehicles are first detected in order to avoid processing of the parts of the frame that are unrelated to vehicles. This step eliminates a great deal of unnecessary processing effort and therefore speeds up the overall process. Vehicle detection is done using a deep network composed of several convolutional layers.

Vehicle detection is done based on the high-level features that are extracted by the designed convolutional network. In fact, the proposed method combines low-level features in primary convolutional layer and high-level features extracted by convolutional and recurrent network to detect vehicles. Moreover, several layers are used for various feature extractions from input images. Since the kernels in various layers vary in size, features are extracted with different details. This helps to have images with different details and to extract relevant and discriminant features. Figure 3 depicts the proposed architecture for vehicle detection.

In [36], we proposed a practical method that produces data for various phases of Iranian plate recognition system. Using deep learning methods requires sufficient data for training models. Due to the lack of data for Iranian plate recognition, the current paper uses the proposed approach in [36].

In the proposed architecture, a model based on VGG16 is used for vehicle detection, that is followed by convolutional layers. These layers decrease in size as we move forward such that we are able to detect vehicles with different sizes. Each layer extracts a different high-level feature from vehicle for vehicle detection.

In the proposed model, training objective is derived from the multibox objective [37, 38] but is extended to handle multiple object categories. Let $x_{i,j}^p = \{0, 1\}$ be an indicator for matching the i -th default box to the j -th ground truth box of category p . In the matching strategy above, we can

have $\sum_i \mathbb{1} x_{i,j}^p > 1$. Equation (1) presents the overall objective loss function is a weighted sum of the localization loss (loc) and the confidence loss, where N is the number of matched default boxes.

$$L(x, c, l, g) = \frac{1}{N} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)). \quad (1)$$

If $N = 0$, the loss is set to 0. The localization loss is a smooth L_1 loss [33] between the predicted box (l) and the ground truth box (g) parameters. The weight term α is set to 1 by cross validation.

2.2. Plate Detection. Most of plate images are noisy due to environmental conditions. Using Gaussian filter is a common way to address this issue. Gaussian filter is used as a preprocessing step in image processing and is a linear filter that smooths and removes noise. The first activity in this step is to improve quality of the vehicle image using Gaussian filter. Then, a deep convolutional network-based method is proposed in order to detect the plate location. Figure 4 shows the proposed architecture for plate detection. The input resolutions in deep network are 320×320 , 416×416 , and other multiples of 32. Therefore, the input vehicle image is resized to a 1:1 square, in the proposed method. Deep network preserves the aspect ratio of the images, and if the input image is not in a square shape, it automatically adds black bars to conform to a square shape.

In the proposed method, 416×416 resolution is used for plate detection. Plate detection works the best using this value, which is obtained from several experiments on the size of Iranian vehicles. Deep architecture of proposed method performs segmentation of size $M \times M$, where M is the size of window that can have different values. Two parameters should be given in prior for plate detection process: N that is the number of bounding boxes and S that is the confidence score for each bounding box and determines whether there is an object inside the box or the probability of its existence. We should calculate evaluation metric such as the Intersection over Union (IoU) to compare predicted bounding boxes and the dimensions and location of ground truth, to measure the S parameter. The IoU is computed as shown in Equation (2), where X and Y are the two bounding boxes. Higher values for IoU indicate that the two bounding boxes have

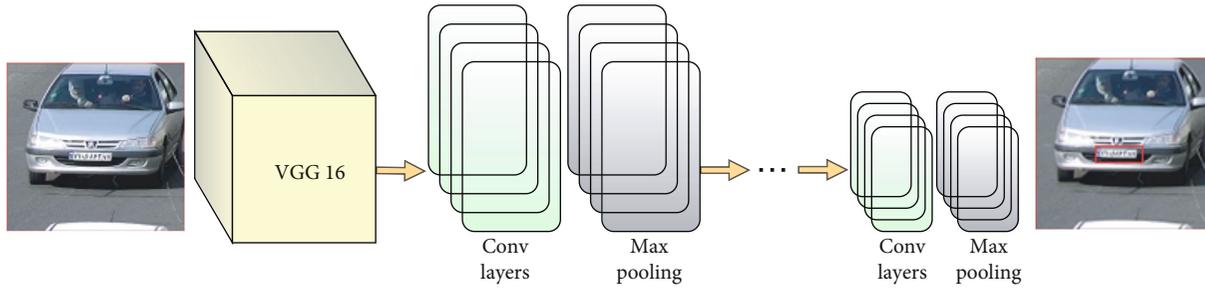


FIGURE 4: The deep architecture used in the proposed method for plate detection.

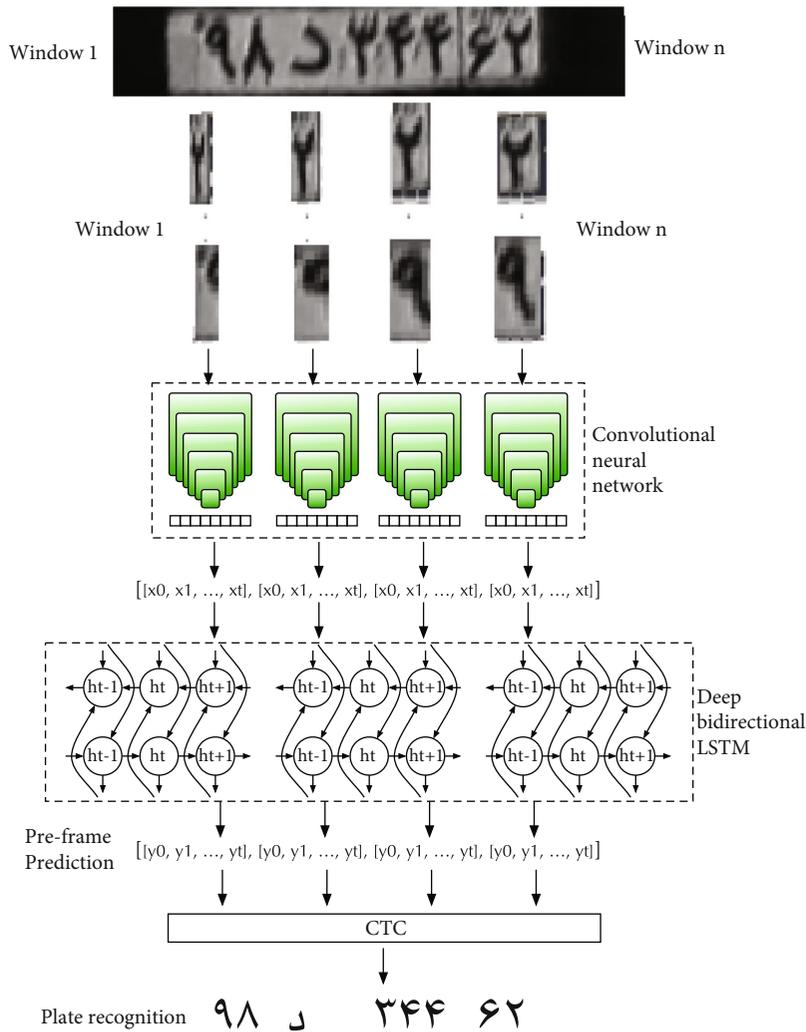


FIGURE 5: The deep architecture used in the proposed method for character recognition.

a similar location and dimension, to a large extent.

$$IoU(X, Y) = \frac{\text{Area of overlap } (X, Y)}{\text{Area of union } (X, Y)}. \quad (2)$$

For the sake of better performance, other parameters such as confidence threshold and anchors are constant

based on the default setting. There are special features that are extracted by the deep architecture of the proposed method. These features include maximum and minimum length and height, maximum and minimum number of pixels, and area. Spall plates with unusual resolution are ignored in the proposed architecture. Classification is done in the late stage of process. Those objects that have plate features are classified as plate and are ignored otherwise.

2.3. Character Recognition. Character detection and recognition is one of the most important steps in plate recognition. In this paper, a method based on convolutional and recurrent networks is used for plate detection. LSTM architecture is very efficient here due to the fact that characters in Iranian plates follow a specific pattern. This method is fast enough for real-time applications since plate images enter to the network, and in the end, characters are recognized.

Unlike classic methods, this method does not require separate stages of segmentation, feature extraction, and classification. Figure 5 depicts the proposed deep architecture for plate detection that contains convolutional and recurrent layers. The network configuration used in our experiments is summarized in Table 1. The main advantage of the proposed architecture is that it only requires plate images and their label sequences and there is no need for plate segmentation. Moreover, various features must be extracted from plate for classifying characters to 37 possible classes. Convolutional networks help to extract these discriminant features. Features extracted from plates using CNN enter LSTM networks. Using LSTM units in RNN is very useful since characters and digits have a special order in Iranian plates. In this method, features are extracted from plate image based on shape and sequence using convolutional layers. These features are used as the input of recurrent layers in the end part of network. These layers consider all the sequence record. LSTM units are used that contain memory cells and gates instead of RNN units, in order to avoid gradient vanishing. In this stage, characters are classified in 37 classes by the network. These classes include 10 digits (0 to 9) and 24 characters that are shown in Table 2. There is also a special character for disabled drivers. In this kind of plates, an image of a wheelchair replaces a Persian character. Moreover, there are two English characters, *S* and *D*, that are used for special purposes. It is worthy to mention that scarcity of plates with special characters prevents training networks with these characters. To address this issue, we used deep networks to produce plates with these special characters. Samples of these plates are shown in Figure 6. This paper also proposes a method based on generative adversarial networks (GAN) to produce plate images with different qualities. Existing data is used to train the network in the proposed method; then, this trained network is used to produce various plate images.

3. Experimental Results

Since the proposed method is based on deep learning, there is a need for a large volume of data that are labeled in various phases. To this end, data are gathered from many cameras in streets and highways during several days and nights under various illumination and weather conditions. There are over four million frames, and each of which contains several, one, or no vehicle.

Classic methods are used for labeling data in vehicle detection, plate detection, and plate recognition process. The proposed model is trained using these frames that are labeled for vehicle detection. Three million frames are used for plate detection and recognition. Images have various illu-

TABLE 1: Network configuration summary: “*k*”, “*s*”, and “*p*” stand for kernel size, stride, and padding size, respectively.

| Type | Configuration |
|---------------------|--|
| Input | $W \times 32$ |
| Conv | #kernels: 64, $k: 3 \times 3$, $s: 1$, $p: 1$ |
| Max pooling | Windows: 2×2 , $s: 2$ |
| Conv | #kernels: 128, $k: 2 \times 2$, $s: 1$, $p: 1$ |
| Max pooling | Windows: 2×2 , $s: 2$ |
| Conv | #kernels: 256, $k: 3 \times 3$, $s: 1$, $p: 1$ |
| Conv | #kernels: 256, $k: 3 \times 3$, $s: 1$, $p: 1$ |
| Max pooling | Windows: 1×2 , $s: 2$ |
| Conv | #kernels: 512, $k: 3 \times 3$, $s: 1$, $p: 1$ |
| Batch normalization | |
| Conv | #kernels: 512, $k: 3 \times 3$, $s: 1$, $p: 1$ |
| Batch normalization | |
| Max pooling | Windows: 1×2 , $s: 2$ |
| Conv | #kernels: 512, $k: 2 \times 2$, $s: 1$, $p: 0$ |
| Map to sequence | |
| Bidirectional LSTM | #hidden unit: 256 |
| Bidirectional LSTM | #hidden unit: 256 |
| Bidirectional LSTM | #hidden unit: 256 |
| Transcription | |

mination, shadow, reflection, weather conditions, and qualities in order to better train the model. It is worthy to mention that all hyperparameters are chosen based on numerous random search tests. First, a set of hyperparameters was chosen and the model was trained using the training data. The set of hyperparameter was then evaluated based on the testing data. This process was repeated, and the hyperparameters with highest accuracy were chosen. All hyperparameters keep their values in different datasets.

In this step, the neural network presented in Section 2.1 is used in order to address the problem of scarcity of plates with special character; some of which are shown in Figure 6. Plates containing these characters are rare and belong to few organizations.

Figure 7 shows several plate samples in our database. It should be noted that background color of Iranian plates can be white, red, blue, yellow, or green. The advantages of deep learning-based methods over other methods such as color-based methods are that they are not dependent on color, have lower fault rate, and can be used under different illumination situation such as high, low, and uneven illumination. In the proposed method, vehicles are first detected using deep networks to expedite the plate detection process.

The output of the proposed method in the vehicle detection phase with unfixed shooting angles and different qualities in real scenes is shown in Figure 8. As it can be seen, the proposed method is capable of detecting vehicles under various illumination conditions and from front, back, and different angles. The proposed method is able to detect vehicles with high accuracy owing to extraction of high-level features for vehicle detection process. The advantages of this method is that there is no limitation in the number

TABLE 2: Different letters and digits in Iranian plates.

| Type | Characters |
|------|---------------------------------|
| 1 | 0 1 2 3 4 5 6 7 8 9 و ط ل م ژ ز |
| 2 | SD ی ه ن گ ک ق ع د ج |
| 3 | الف ف ص س ش ت ث پ ب |



FIGURE 6: Samples of plates with special characters.



FIGURE 7: Samples of plates with special characters.

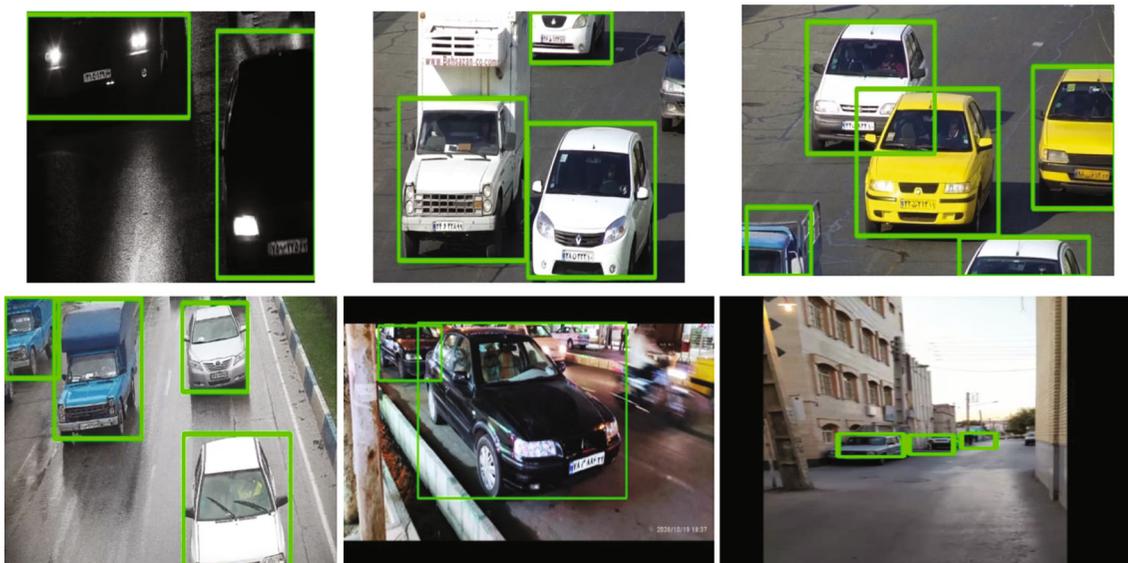


FIGURE 8: Output of the proposed method in the vehicle detection phase.

of vehicles and the vehicles can even occlude in the image. The proposed model is trained using images with different qualities, shooting angles, and illumination conditions in

order to improve detection rate. Training images also include images taken during day and night and different weather conditions. This phase of the proposed method is

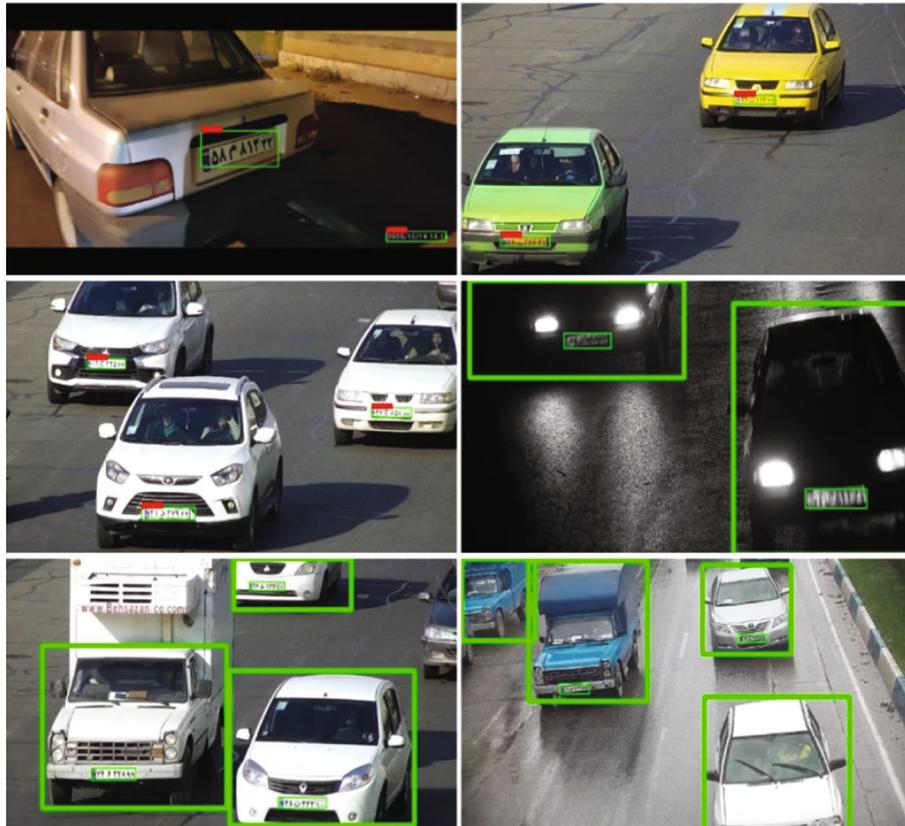


FIGURE 9: Output of the proposed method in the vehicle detection phase.

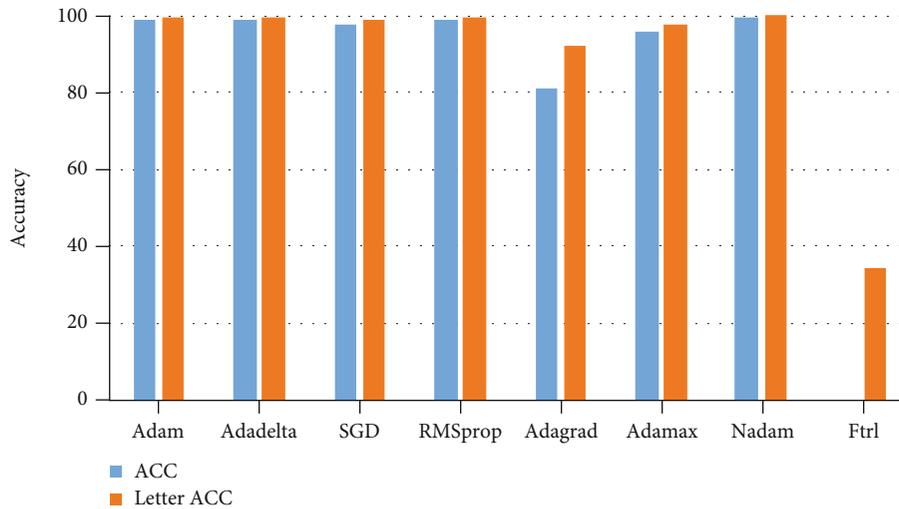


FIGURE 10: Detection rate of the proposed method with different optimizer functions.

based on labeled data, and the detection rate is 100%. This detection rate guarantees that no vehicle is missed by the proposed method.

The next step is plate detection where the detected vehicle image from the previous step is searched instead of the whole image. This considerably speeds up the plate detection process. Training the proposed deep network is done using three million images with labeled plates. Image processing methods are used

for labeling images and detecting plates. There are images with different qualities and shooting angles in the training set in order to improve the detection rate in the proposed method. Images with different illumination condition are also present in the training set. In the proposed method, different plate features including maximum and minimum length and width and plate area are extracted based on the training dataset. Figure 9 depicts output of the proposed method for six images.

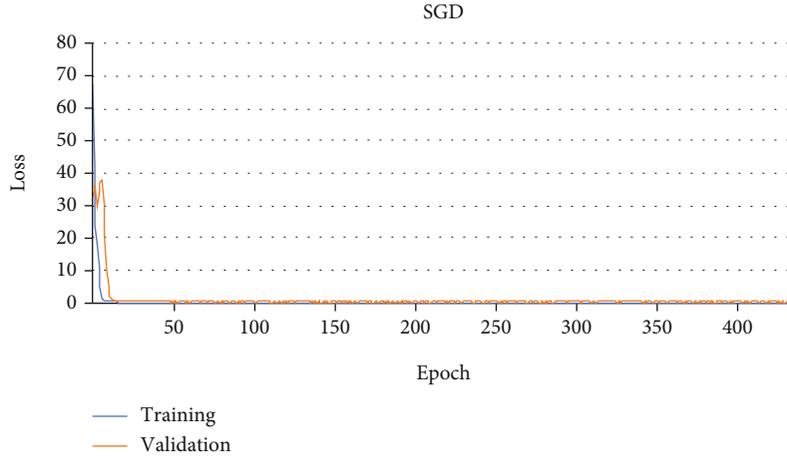


FIGURE 11: Loss diagram of the proposed method with SGD optimizer function.

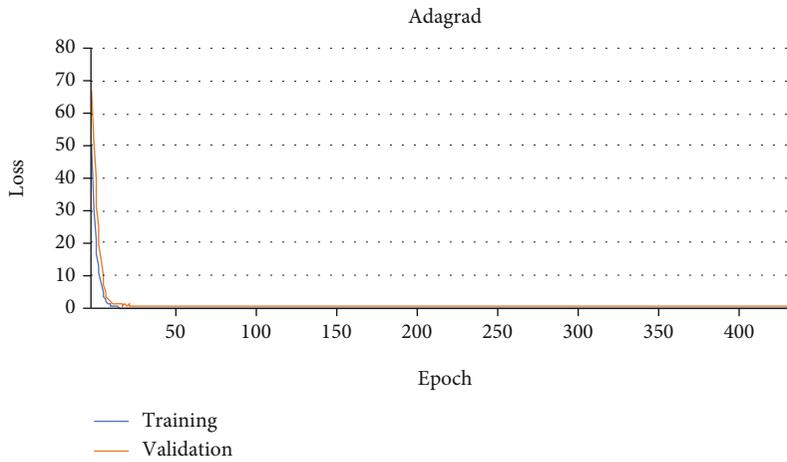


FIGURE 12: Loss diagram of the proposed method with Adagrad optimizer function.

TABLE 3: Comparison between the proposed method and other methods on different datasets.

| Dataset | Plate detection Acc | | Plate recognition Acc | | Overall Acc | |
|---------|---------------------|----------------|-----------------------|----------------|--------------|----------------|
| | Reported Acc | Our method Acc | Reported Acc | Our method Acc | Reported Acc | Our method Acc |
| [10] | 99% | 100% | 97% | 100% | 96% | 100% |
| [17] | 96% | 100% | 94% | 100% | 90% | 100% |
| [34] | 98% | 100% | 98% | 100% | 96% | 100% |

As it can be seen, the proposed method is able to detect plates in images with different shooting angles (including front and back), distances, and illumination conditions. Detection rate in this step is 100%. In the next step, the characters of the detected plates should be recognized. To this end, the convolutional and recurrent network mentioned in Section 2.3 needs to be trained using different plates. Then, the proposed model is evaluated based on the test data.

Since the proposed model is trained using different datasets, different features from plates are extracted in various layers. These features include edges, corners, and structures and are extracted based on CNN layers. Moreover, in LSTM

layers, sequence features are extracted. This architecture helps to successfully extract discriminant features for character recognition.

The experiments on the test data show that the detection rate of all characters in the proposed method is 99.37%. Moreover, character detection rate is 99.92% that is calculated based on each character and not the whole plate. Choosing a proper optimizer function is a determining factor. The proposed model is tested with several optimizers.

The validation results using different optimizers are shown in Figure 10. As it can be seen, Adma-based optimizer functions outperform other functions and have a recognition rate of more than 99%. Ftrl optimizer does not

TABLE 4: Comparison between the proposed methods and the other methods.

| Methods | Plate detection | Reported Acc | | Overall Acc | Plate characters | Number of plates | Image size | Processing time | Method | |
|---------------------|-----------------|------------------------|-------------------|-------------|---------------------|------------------|-------------|-----------------|--|-----------------------|
| | | Character segmentation | Plate recognition | | | | | | D: detection method | R: recognition method |
| [5] | 98.7% | 100% | 97.6% | 96.33% | Persian | 10000 | Variable | 180 | D: CCA and RANSAC R: probabilistic SVM | |
| [10] | 99.33% | NR | 96.6% | 96% | Persian | 150 | 640 × 480 | NR | D: color features R: ANN | |
| [14] | 97.3% | NR | 94.5% | 91.94% | Persian | 320 | 640 × 480 | NR | D: edge features (Sobel) R: MLP | |
| [17] | 96.93% | 98.75% | 94.5% | 90.45% | Persian | 1185 | 1024 × 768 | NR | D: morphological operations and Adaboost R: SAMME | |
| [23] | 97.7% | — | 98.8% | NR | Chinese | 250 K | 96 × 32 | NR | D: MTCNN R: MTLPR | |
| [29] | 100% | — | 96.78% | NR | Chinese | — | 300 × 300 | — | D: CNN R: CNN | |
| [35] | NR | 99% | NR | NR | Korean | 120 | 640 × 480 | NR | D: sliding concentric windows R: ANN | |
| [30] | 100% | NR | 98.4% | NR | Chinese | 2507 | 1600 × 1200 | 42 | YOLOv3 (352 × 288) | |
| [30] | 98.27% | NR | 98.1% | NR | Chinese | 2507 | 1600 × 1200 | 161 | Faster_RCNN_ResNet101 (800 × 600) | |
| [27] | 98.04% | NR | 94.12% | NR | Chinese | 2049 | 48 × 640 | 400 | Unified deep neural network | |
| [39] | 97.16% | 98.34% | 97.88% | 93.54% | English Japanese | 9026 | NR | 288 | D: improved Bernsen algorithm R: SVM | |
| [39] | 96.5% | NR | 89.1% | 86% | English | 1334 | NR | 276 | D: sliding concentric windows and CCA R: probabilistic NN | |
| [40] | 95.9% | NR | 92.3% | 90% | Chinese English | 5026 | 720 × 576 | 125 | D: edge features (Sobel) R: feed forward NN | |
| [41] | 97.3% | NR | 95.7% | 93.1% | English | 1176 | 640 × 480 | 223 | D: salient features R: self-defined classifier | |
| [42] | 97.1% | NR | 96.4% | 93.6% | English | 332 | 867 × 623 | 594 | D: color features and Hough transform R: feed forward NN | |
| [43] | 94.43% | — | 99.37% | | Arabic | 600 | 380 × 540 | NR | D: DSSN R: CNN | |
| [44] | 97.76% | — | 95.05% | NR | Persian | 5719 | Variable | 54.18 | D: YOLO-v3 R: YOLO-v3 | |
| [45] | 99.37% | — | 99.53% | 98.9% | Brazilian | — | 1024 × 768 | — | D: YOLO-v3 R: CNN | |
| [46] | 99.72% | — | 87% | — | Jordanian | 187200 | 1920 × 1080 | — | D: YOLO-v3 R: CNN | |
| [47] | 98.22% | — | 87% | — | English | 2049 | Variable | — | D: YOLO-v3 R: YOLO-v3 | |
| The proposed method | 100% | — | 99.37% | 99.6% | Persian | 3 million | Variable | 46 | D: CNN R: CNN+LSTM | |

properly fit in the proposed method and shows plate recognition rate of 0% and character recognition rate of 37%. Therefore, Adadelta optimizer function that is based on Adam is used in the proposed method. In order to examine

the convergence of the proposed method, loss diagram of the proposed method is shown with two optimizer functions of SGD and Adagrad in Figures 11 and 12, respectively. As it can be seen, loss rates gradually converge to zero on the

TABLE 5: Average processing times (in millisecond) of the proposed method in different scenarios for different steps.

| Total time | Recognition plate | Plate detection | Vehicle detection | Number of vehicle |
|------------|-------------------|-----------------|-------------------|-------------------|
| 8 | 0 | 3 | 5 | 0 |
| 16 | 3 | 4 | 9 | 1 |
| 25 | 6 | 5 | 14 | 2 |
| 34 | 9 | 7 | 17 | 3 |
| 41 | 12 | 8 | 21 | 4 |
| 48 | 15 | 9 | 24 | 5 |
| 56 | 18 | 10 | 27 | 6 |
| 61 | 21 | 11 | 29 | 7 |

training and testing dataset. It shows that the model parameters are well learned based on the input data and lead to loss rate reduction during repetitive process of learning. In Figures 11 and 12, the horizontal axis presents epoch in the evaluation steps in the training and validation sets, and the vertical axis presents amount of loss in these sets.

3.1. Comparison with Other Methods. Several datasets are used in order to compare the proposed method with other methods in terms of efficiency. Table 3 shows this comparison using several datasets containing images of Iranian vehicles in terms of efficiency in different steps. According to the results, the proposed method performs well for dirty and low-quality images. In this part, used datasets contain images of already cropped vehicles in order to evaluate the proposed method in the plate detection step. Plate detection rate and character recognition rate are 100% in the proposed method. The images in the used dataset are captured by handheld cameras and have good quality.

Table 4 shows the result of experiments with four criteria to evaluate the proposed method and compare it with other methods. These criteria are plate characters, number of plates, image size, and processing time. Plate detection and recognition process are considered in this set of experiments. The advantage of the proposed method over other methods, that do not use deep learning, is that it is segmentation-free. This improves recognition rate and also speeds up the plate detection and recognition process.

As it can be seen in Table 4, the proposed method shows a great performance in vehicle and plate detection, and its accuracy is 100%. It should be noted that the overall accuracy (Overall Acc) is the amount of accuracy when plate detection and character recognition are done simultaneously.

3.2. Real-Time Evaluation. This set of experiments focuses on applicability of the proposed method in real-time situations. Here, the input is a video with 30 frames per second. Plate detection is a time-consuming step in plate recognition process and it gets more complicated as the number of plates increases in the input image. In the proposed method, vehicles are first detected to speed up the plate detection process. In this way, exploration is done only in the detected vehicle

image instead of the whole image. The character recognition process becomes slower as the number of vehicles in the image increases; therefore, inputs with different number of vehicles are used in order to demonstrate applicability of the proposed method. Table 5 shows average processing times for a 10-time execution of the proposed method in each step for different scenarios. The total processing time increases as the number of vehicles increases. As it can be seen, the proposed method is able to be used in real-time situation even with many vehicles in a single image.

4. Conclusion

Monitoring systems are necessary in highways and roads, especially due to the increasing number of vehicles and increasing traffic volume. Plates are used to recognize vehicles in videos and images captured by monitoring cameras, since plates are the unique identification of registered vehicles. In this paper, vehicle plate recognition contains three steps: vehicle detection, plate detection, and character recognition. The proposed method in this paper detects incoming vehicles in each frame that are inputs for the next step (plate detection and extraction). CNN is used for vehicles and plate detection, and deep model with CNN and RNN layer is used for character recognition.

The proposed method is evaluated using the frames of an input video. The experimental evaluations show that the proposed method is capable of performing all three steps of vehicle detection, plate detection, and character recognition efficiently with negligible error rates. Moreover, the proposed method is shown to be viable for real-time applications based on its response time. The proposed method in this paper can substantially improve plate recognition accuracy based on the obtained results.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] D. Shan, 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, 2012.
- [2] C.-C. Tsai, C.-K. Tseng, H.-C. Tang, and J.-I. Guo, "Vehicle detection and classification based on deep neural network for intelligent transportation applications," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1605–1608, Honolulu, HI, USA, 2018.

- [3] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 580–596, Munich, Germany, 2018.
- [4] Y. Yuan, W. Zou, Y. Zhao, X. Wang, H. Xuefeng, and N. Komodakis, "A robust and efficient approach to license plate detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1102–1114, 2017.
- [5] R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2017.
- [6] M. Y. Arafat, A. S. M. Khairuddin, and R. Paramesran, "Connected component analysis integrated edge based technique for automatic vehicular license plate recognition framework," *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 712–723, 2020.
- [7] S. Yu, B. Li, Q. Zhang, C. Liu, and M. Q.-H. Meng, "A novel license plate location method based on wavelet transform and EMD analysis," *Pattern Recognition*, vol. 48, no. 1, pp. 114–125, 2015.
- [8] M. S. Al-Shemarry, Y. Li, and S. Abdulla, "Ensemble of Ada-boost cascades of 3L-LBPs classifiers for license plates detection with low quality images," *Expert Systems with Applications*, vol. 92, pp. 216–235, 2018.
- [9] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, "Vertical-edge-based car-license-plate detection method," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 26–38, 2013.
- [10] R. Azad, F. Davami, and B. Azad, "A novel and robust method for automatic license plate recognition system based on pattern recognition," *Advances in Computer Science: an International Journal*, vol. 2, no. 3, pp. 64–70, 2013.
- [11] A. Mousa, "Canny edge-detection based vehicle plate recognition," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, no. 3, pp. 1–8, 2012.
- [12] P. Dollár, T. Zhuowen, P. Perona, and S. Belongie, "Integral channel features," in *Proceedings of the British Machine Vision Conference*, BMVC Press, London, 2009.
- [13] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *IEEE International Joint Conference on Biometrics*, pp. 1–8, FL, USA, 2014.
- [14] S. Ghofrani and M. Rasooli, "Farsi license plate detection and recognition based on characters features," *Majlesi Journal of Electrical Engineering*, vol. 5, no. 17, pp. 44–51, 2011.
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.
- [16] M. M. Dehshibi and R. Allahverdi, "Persian vehicle license plate recognition using multiclass Adaboost," *International Journal of Computer and Electrical Engineering*, vol. 4, no. 3, pp. 355–358, 2012.
- [17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [18] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.
- [19] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Automatic license plate recognition with convolutional neural networks trained on synthetic data," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSp)*, pp. 1–6, Luton, UK, 2017.
- [20] 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*, pp. 594–603, Springer, 2018.
- [21] C. Henry, S. Y. Ahn, and S.-W. Lee, "Multinational license plate recognition using generalized character sequence detection," *Access*, vol. 8, pp. 35185–35199, 2020.
- [22] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, 2016.
- [23] W. Wang, J. Yang, M. Chen, and P. Wang, "A light CNN for end-to-end car license plates detection and recognition," *IEEE Access*, vol. 7, pp. 173875–173883, 2019.
- [24] G. O. N. G. Wen-bin, S. H. I. Zhang-song, and J. I. Qiang, "Non-segmented Chinese license plate recognition algorithm based on deep neural networks," in *2020 Chinese Control and Decision Conference (CCDC)*, pp. 66–71, Hefei, China, 2020.
- [25] 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.
- [26] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, 2017.
- [27] H. Li, P. Wang, and C. Shen, "Toward end-to-end car license plate detection and recognition with deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, 2019.
- [28] C. L. P. Chen and B. Wang, "Random-positioned license plate recognition using hybrid broad learning system and convolutional networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 444–456, 2020.
- [29] S.-L. Chen, C. Yang, J.-W. Ma, F. Chen, and X.-C. Yin, "Simultaneous end-to-end vehicle and license plate detection with multi-branch attention neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3686–3695, 2020.
- [30] F. Gao, Y. Cai, Y. Ge, and S. Lu, "EDF-LPR: a new encoder-decoder framework for license plate recognition," *IET Intelligent Transport Systems*, vol. 14, no. 8, pp. 959–969, 2020.
- [31] W. Weihong and T. Jiaoyang, "Research on license plate recognition algorithms based on deep learning in complex environment," *IEEE Access*, vol. 8, pp. 91661–91675, 2020.
- [32] S. Zhang, G. Tang, Y. Liu, and H. Mao, "Robust license plate recognition with shared adversarial training network," *IEEE Access*, vol. 8, pp. 697–705, 2020.
- [33] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, 2015.
- [34] M. Nejati, A. Majidi, and M. Jalalat, "License plate recognition based on edge histogram analysis and classifier ensemble," in *2015 Signal Processing and Intelligent Systems Conference (SPIS)*, pp. 48–52, Tehran, Iran, 2015.

- [35] K. Deb, V. V. Gubarev, and K.-H. Jo, "Vehicle license plate detection algorithm based on color space and geometrical properties," in *International Conference on Intelligent Computing*, pp. 555–564, Springer, 2009.
- [36] J. Pirgazi, A. G. Sorkhi, and M. M. P. Kallehbasti, "An efficient robust method for accurate and real-time vehicle plate recognition," *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1759–1772, 2021.
- [37] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2154, Columbus, OH, USA, 2014.
- [38] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," 2014, <http://arxiv.org/abs/1412.1441>.
- [39] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011.
- [40] J. Jiao, Q. Ye, and Q. Huang, "A configurable method for multi-style license plate recognition," *Pattern Recognition*, vol. 42, no. 3, pp. 358–369, 2009.
- [41] Zhen-Xue Chen, Cheng-Yun Liu, Fa-Liang Chang, and Guo-You Wang, "Automatic license-plate location and recognition based on feature salience," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3781–3785, 2009.
- [42] Jing-Ming Guo and Yun-Fu 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.
- [43] N. Omar, A. Sengur, and S. G. S. Al-Ali, "Cascaded deep learning-based efficient approach for license plate detection and recognition," *Expert Systems with Applications*, vol. 149, article 113280, 2020.
- [44] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaei, and C. Y. Suen, "A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems," *IEEE Access*, vol. 8, pp. 201317–201330, 2020.
- [45] D. M. Izidio, A. Ferreira, H. R. Medeiros, and E. N. Barros, "An embedded automatic license plate recognition system using deep learning," *Design Automation for Embedded Systems*, vol. 24, pp. 23–43, 2020.
- [46] S. Alghyaline, "Real-time Jordanian license plate recognition using deep learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, 2020.
- [47] Hendry and R.-C. Chen, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image and Vision Computing*, vol. 87, pp. 47–56, 2019.