

Received October 4, 2020, accepted October 22, 2020, date of publication November 4, 2020, date of current version November 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035992

A Robust Deep Learning Approach for Automatic Iranian Vehicle License Plate Detection and Recognition for Surveillance Systems

ALI TOURANI¹, ASADOLLAH SHAHBAHRAMI¹, SAJJAD SOROORI¹, SAEED KHAZAEI²,
AND CHING YEE SUEN², (Life Fellow, IEEE)

¹Department of Computer Engineering, University of Guilan, Rasht 4199613776, Iran

²Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montréal, QC H3G 1M8, Canada

Corresponding author: Asadollah Shahbahrani (shahbahrani@guilan.ac.ir)

This work was supported by the Guilan Science and Technology Park.

ABSTRACT The process of detecting vehicles' license plates, along with recognizing the characters inside them, has always been a challenging issue due to various conditions. These conditions include different weather and illumination, inevitable data acquisition noises, and some other challenging scenarios like the demand for real-time performance in state-of-the-art Intelligent Transportation Systems (ITS) applications. This paper proposes a method for vehicle License Plates Detection (LPD) and Character Recognition (CR) as a unified application that presents significant accuracy and real-time performance. The mentioned system is designed for Iranian vehicle license plates, which have the characteristics of different resolution and layouts, scarce digits/characters, various background colors, and different font sizes. In this regard, the system uses a separate fine-tuned You Only Look Once (YOLO) version 3 platform for each of the mentioned phases and extracts Persian characters from input images in two automatic steps. For training and testing stages, a wide range of vehicle images in different challenging and straightforward conditions have been collected from practical systems installed as surveillance applications. Experimental results show an end-to-end accuracy of 95.05% on 5719 images. The test data included both color and grayscale images containing the vehicles with different distances and shooting angles with various brightness and resolution. Additionally, the system can perform the LPD and CR tasks in an average of 119.73 milliseconds for real life data, which illustrates a real-time performance for the system and usable applicability. The system is fully automated, and no pre-processing, calibration or configuration procedures are needed.

INDEX TERMS Deep learning, automatic license plate recognition, license plate detection, intelligent transportation systems.

I. INTRODUCTION

Nowadays, Intelligent Transportation Systems (ITS) are known as the vital means of municipalities and governments for applying modern traffic planning policies. ITS are mostly integrated with video cameras installed in different locations of the roadways and urban areas for surveillance and enforcement purposes [1]. Since the outputs of these cameras are video frames or a set of images, fetching valuable results from them using machine vision and Digital Image Processing (DIP) techniques, lead to the development

of numerous applications for ITS. Among most of these applications, Automatic License Plate Recognition (ALPR) is considered a major step [2]–[6]. ALPR refers to the process of recognizing the characters existing inside the vehicles' license plates through machine vision and artificial intelligence methods. Some of the most common applications in which the license plate recognition process is essential are traffic monitoring systems [7]–[9], navigation and vehicle tracking [10], [11], toll payments and control systems in public/private parking areas [12], [13], and the identification of stolen vehicles [14], [15]. In this regard, the correct extraction of characters from license plates is significantly essential to take further actions like enforcement and prosecution.

The associate editor coordinating the review of this manuscript and approving it for publication was Xian Sun.

A typical ALPR system consists of three main modules to provide reasonable outcomes. These three steps are known as License Plate Detection (LPD), Character Segmentation (CS), and Optical Character Recognition (OCR) which can be found in most of the approaches [2], [3], [5], [7], [14], [15]. In the first step of the pipeline, i.e., LPD, the candidate regions containing vehicles' license plates are detected and cropped for further processing. This phase plays a crucial role in the final performance of the system because the misclassification of non-license plate objects results in errors for later stages. Thus, the ideal output of this step is a set of correctly detected and cropped license plate areas with no misclassified objects or loss of valuable data, e.g., characters. In the character segmentation stage, the system receives the extracted license plates from the LPD stage and produces a set of cropped areas containing the characters. Consequently, the main goal of CS is to detect characters inside the license plate area and segment them from other portions of the license plate, such as a logo or blank fields. Finally, the third phase, i.e., OCR, is responsible for classifying and recognizing each of the segmented characters from the previous stage. The more the system can recognize the characters, the better the total accuracy of the ALPR system will be.

There are numerous methods for each of the mentioned stages in an ALPR system. Some of the most used techniques for LPD are edge and contour detection [17]–[20], texture analysis [21], and color and geometric feature extractions [7], [22]. It has to be mentioned that the ALPR systems face performance deterioration in complex circumstances. Some of the main challenges include occlusion, various license plate layouts and languages, different sizes and width-to-height ratios of the license plates, noisy or dirty input images, and a variety of illumination and weather conditions [4]. Employing an appropriate method or algorithm for each of the mentioned steps to handle these challenges with the lowest possible side effects is a formidable task.

The layout of the typical vehicle license plates in Iran comprises of two sections. The leftmost section provides a unique string of digits and characters for the vehicle, which includes two digits in the left, an alphabetic character in the middle, and three digits in the right. The range of the digits in this section is 0 to 9, and there are 26 classes of characters for the middle part, where eight of these characters are incredibly scarce (such as diplomatic, protocol, and service vehicles). In the rightmost section, there is a two digit code with decreased font size referring to the city where the vehicle owner lives. Here, the digit “0” can only appear in the right digit of the license plates' rightmost section. Furthermore, considering the type and usage of the vehicle, the background color of the license plates may vary. Fig. 1 shows some of the typical Iranian vehicle license plates. As can be seen, the license plates of the free trade zone vehicles have a different width-to-height ratio and layout, containing only numbers along with their transliterated English values.



FIGURE 1. Different types of Iranian vehicle license plates with their corresponding labels.

It should be noted the layout designed for motorcycles is completely different, and recognizing them is not the purpose of this manuscript. These variations make the LPD and CR processes in Iranian vehicle license plates a formidable challenge.

The main goal of this paper is to introduce a robust method for ALPR, which presents very high accuracy and real-time performance in realistic data. The proposed method utilizes Deep Neural Networks (DNNs) and considers both the LPD and CR phases as classification practices. In this regard, in the LPD stage, if one or multiple objects containing the visual features of Iranian vehicle license plates in various distances are found in a given image, they will be cropped and separated from the input data. Similarly, the recognition of characters and digits inside cropped license plate areas can be accomplished through a classification procedure. The proposed method uses You Only Look Once (YOLO) [23] approach for each of the mentioned stages. The main contributions of the proposed method are:

- Implementing a real-time ALPR system based on deep learning with excellent results and high accuracy in real life circumstances
- Training and fine-tuning two separate YOLO v.3 network architectures for LPD and CR purposes
- Solving the typical challenges of OCR by considering the recognition of characters and digits as a classification issue
- Collecting a vast amount of realistic condition data in both challenging and straightforward scenarios captured in various illumination, weather, and noise conditions
- Employing different augmentation techniques on the scarce samples to broaden the dataset

The rest of the paper is organized as follows: Section II reviews some background information and introduces several related approaches in summary. In Section III, the main architecture of the proposed method is presented. Section IV provides the experimental results and discusses the major challenges. Finally, the paper is concluded in Section V.

II. BACKGROUND AND RELATED WORK

A. DEEP LEARNING APPROACHES IN MACHINE VISION

Deep architectures refer to the structure of DNNs, which contain hundreds and even thousands of hidden layers between the input and output layers. The hidden layers progressively extract valuable information from their previous layers by applying high-level filters [24]. In the fields of machine vision and object detection, Convolutional Neural Networks (CNNs) are the most popular branch of DNNs that attracted huge interest among researchers in recent years [25]. These networks have proven to provide superb accuracy comparing to other supervised learning models like Support Vector Machines (SVMs) in solving classic machine vision issues. By feeding the CNNs, the existing relations between the discriminative features are fetched, and the weights of the neurons are updated via the backpropagation algorithm [25]. The various types of layers in CNNs include convolution layers for high-level feature extraction (i.e., kernels), fully-connected layers for non-linear feature learning, and pooling layers for lessening of the input tensor's parameters.

Since the number of candidate regions as the output of CNNs is large [25], other extensions of these networks have been introduced. Region-based Convolutional Neural Networks (R-CNNs) [26] is a state-of-the-art approach that reduces the number of candidates through merging neighboring regions employing Selective Search greedy algorithm. Similarly, the R-CNN approach has other optimized extensions, including Fast R-CNNs [27] and Faster R-CNNs [28]. In Fast R-CNN, the training process has been improved by indirect feeding of data along with employing a SoftMax probability layer and a Region of Interest (RoI) structure in the pooling layers. The process has even been improved in Faster R-CNN through utilizing Region Proposal Network (RPN) instead of the Selective Search algorithm in R-CNN. Moreover, other deep learning-based approaches present different resolutions for classification and feature extraction goals. YOLO [23] is the most well-known approach which provides substantial performance and accuracy by equally segmenting the input image into equal-sized grids, each with a probability measure of containing a trained object. By looking at the whole image, this approach can bring about almost real-time performance, which is barely reachable through R-CNN approaches. Later versions including YOLO v.2 [29] and YOLO v.3 [30] provide even better results in terms of both accuracy and performance. The second version can classify more than 9000 objects due to an 11-layered architecture trained on ImageNet and a 19-layered network for classification through Common Objects in Context (COCO) dataset [31]. However, the newest version of YOLO, known as YOLO v.3, provides a trade-off between performance and accuracy. YOLO v.3 uses Darknet framework and is pre-trained by COCO dataset. Comparing to the previous version of YOLO, the third version consists of 53 layers as the trained network and another 53 layers for object detection. According to [30],

the main contribution for YOLO v.3 was to increase the accuracy rather than performance by redesigning the underlying Darknet. YOLO v.3 utilizes 53 Darknet layers trained on ImageNet along with another 53 object detection layers, which makes this 106-layered network stronger than previous versions. In addition, since YOLO v.3 applies down-sampling of the dimensions of the image by three times (i.e., 32, 16 and 8, respectively), the detection process takes place on three different sizes. Accordingly, and based on our analysis, YOLO v.3 can be a great choice for ALPR systems, especially for the CR phase, where the objects inside the license plates are rather small.

B. RELATED WORKS

Before the popularization of utilizing DNNs in object detection, an enormous body of research was devoted to the traditional DIP methods. In this regard, since the deep learning approaches provide excellent accuracies compared to the traditional image processing methods, we have only focused on the first and third categories of ALPR systems. In [32], a new two-stage workflow is introduced, which eliminates the need for annotation and segmentation processes. This approach utilizes a hierarchy structure, where in the first step, the candidates of license plates are detected using a weak sparse network, and the validation takes place using a strong CNN. This model has a strong statistical foundation and can handle realistic data challenges; however, the implementation cost is increased due to the structure of the merged stages. Authors in [33] proposed a Multi-Task Convolutional Neural Network (MTCNN) with the advantages of high accuracy and low computational cost. Their network consists of three layers known as P-Net for obtaining the license plate candidates, R-Net for training and fine-tuning of the networks, and O-Net for localization of the license plate corners. Finally, the candidate(s) with the highest score of being a license plate will be extracted using the Non-Maximum Suppression (NMS) algorithm. Zhang *et al.* presented a deep LPD method consisting of both CNNs and Recurrent Neural Networks (RNNs) [34]. Their method is also able to recognize the characteristics of the characters and predict their locations using a vertical anchor mechanism. Through this structure, they have obtained superb results in challenging scenarios like illumination variation and blurry negative effects. Several recent studies have utilized different versions of YOLO for LPD and CR stages, as presented in [44] and [45]. A study in [35] presented a novel system for multinational license plates recognition with a three-layered architecture, including LPD, Unified Character Recognition (UCR), and layout detection. In this approach, the authors have utilized a simplified YOLO v.3 for LPD and another YOLO v.3 with Spatial Pyramid Pooling (SPP) block for CR stages. Another YOLO object detector for ALPR is presented in [36], which works as an online framework to process video frames for detection and tracking targets. Authors in [37] proposed a method for Chinese vehicle license plate recognition, which employs YOLO v.2 detector for LPD and Convolutional Recurrent

Neural Network (CRNN) for CR stages. Their recognition architecture consists of a CNN for context feature extraction, and a two-layered Gated Recurrent Unit (GRU) for feature sequences decode. Hendry *et al.* [38] proposed a combination of YOLO and sliding-window processes. In their method, each character of Taiwan's vehicle license plates is detected by the sliding-window, the YOLO framework then identifies the window.

It has to be mentioned that although some of the presented ALPR approaches may have achieved great performances, they have not experimented under realistic circumstances, where the complexity is inevitable. Most of the employed datasets for training and testing phases include thousands of vehicle images captured in standard and controlled conditions with low acquisition noises and appropriate brightness and contrast. It should be emphasized that, to the best of our analysis, no one has presented a deep learning approach for Iranian LPR, which provides both superb accuracy and performance. The main reason behind this claim is due to the variation of Iranian vehicle license plates and the lack of realistic training data with annotations and determined classes. The mentioned reasons motivated us to propose a novel approach for Iranian vehicle LPR with substantial accuracy and real-time performance, along with the ability to become a commercial product for real condition surveillance. In contrast with most of the cases where there are some inter-dependencies among each stage of the ALPR process, the proposed approach provides an end-to-end recognition.

The authors of this paper have presented several published works in the field of deep learning-based ITS applications. In [16], we have proposed a license plate localization method based on YOLO v.3, which is similar to the approach presented in this paper. Experimental results figured the robustness of our approach with precision and recall values of 0.979 and 0.972. As another approach, we proposed a deep model for vehicle detection based on Faster R-CNN, which can detect the vehicles by their visual features [39]. For this research, we employed the Cars Dataset provided by Stanford University for training and testing purposes. Additionally, in [40], we presented a review of different approaches for ALPR using deep neural networks. In the mentioned paper, we discussed the variety of methods and techniques with their corresponding pros and cons in providing a well-designed ALPR system with high accuracy and acceptable performance.

III. PROPOSED METHOD

In this section, the flowchart to extract the character sequence existing in vehicles' license plates is introduced. The proposed system is aimed to be used as an automatic commercial software for driveway security gates; thus, we have focused on training the system with such circumstances. The system consists of two deep object detection layers, both implemented by YOLO architectures. Since we need to provide a real-time platform with high accuracy in LPD and CR, the third version of YOLO has been selected for both stages.

According to the descriptions presented in Section II, YOLO v.3 is beneficial to the proposed method, where the license plate localization process should be done in real-time, and most of the characters inside the license plates are small in size. Fig. 2 illustrates an overview of the proposed method in brief. According to this figure, the input is an image, a series of images, or video frames, and the output is a sequence of characters referring to the context inside the vehicles' license plates. As a vehicle image is fed to the system, some image enhancement methods are utilized to improve the given image quality and produce a new image which is better in some sense compared to the original one. After that, the LPD stage triggers and seeks for the object(s) with the features of the vehicle license plates. In this phase, the system utilizes a pre-trained YOLO v.3 deep architecture to detect all existing license plates. For CR, a second YOLO v.3 deep network is triggered to recognize the characters and digits inside the cropped license plates. As a result of this, the character segmentation and OCR processes are combined to build a classification process for CR, which treats each character or digit as a unique object. If no license plates have been detected in the LPD stage, the next modules, including license plate cropping and character recognition are ignored, and the system waits for another input image.

Hereby and to explain more about the diagram, in the first step the input image is processed by means of some image enhancement techniques. In this stage, a Histogram Equalization (HE) method is adopted to standardize the intensities of the input image by spreading out the most recurrent ones, which results in an improved contrast. Fig. 3 demonstrates the impact of applying the histogram equalization technique on some of the challenging images. Accordingly, images with dark and indistinguishable license plates have been improved.

After improving the image quality, the second stage, i.e., deep learning-based license plate detection, will be triggered. Fig. 4 shows the structure of the mentioned YOLO v.3 networks of the proposed approach. As it can be seen, the input images of each deep network are resized into a 1:1 square image, because the input resolutions in YOLO v.3 can be 320×320 , 416×416 , and other multiples of 32. YOLO keeps the aspect ratio of the images. If the input image is not a square, it automatically uses black bars to fit an image with different aspect ratios to a square one. In the proposed method, we have used the resolution of 416×416 for LPD and 224×224 for CR networks. These values have been obtained empirically to acquire the best performance for the system. The deep architecture of YOLO performs a grid segmentation on the image, where the grid sizes are equally $M \times M$. The prediction process needs two parameters, including several bounding boxes N and a confidence score S for each bounding box, known as the "objectness" or how likely the box contains an object. To measure the S parameter, we should calculate the Intersection over Union (IoU) evaluation metric to compare the location and dimensions of ground truth and predicted bounding boxes. Equation (1) shows how IoU is

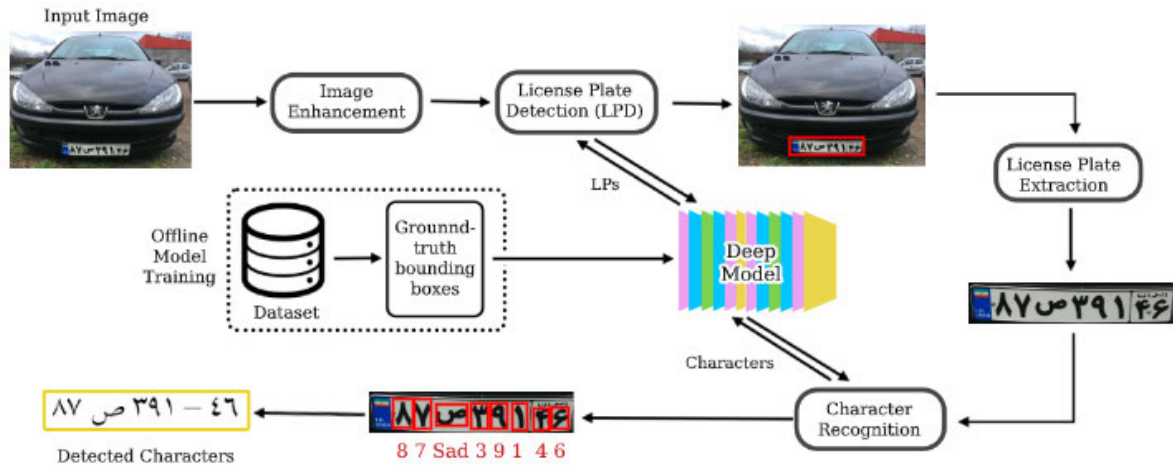


FIGURE 2. Block diagram of the proposed method.



FIGURE 3. The influence of applying histogram equalization on the input images.

calculated:

$$IoU(A, B) = \frac{Area of\ Overlap(A, B)}{Area\ of\ Union(A, B)} \quad (1)$$

where A and B are two bounding boxes. Higher values for IoU indicates that the two bounding boxes have a similar dimension and location, to a large extent.

A. LICENSE PLATE DETECTION NETWORK

In the first stage, i.e., the vehicle license plate detection, the application searches for the existing license plates with their specified visual features. In this regard, the input of the LPD network is a set of color or grayscale images containing vehicle images. For better detection of license plates and provision of a better IoU, we changed the last convolution layer of YOLO v.3 based on the one possible class of “Iranian Vehicle License Plate”. Other parameters, including the

confidence threshold and anchors, remained unchanged based on the default configurations to access a beneficial performance. As Fig. 1 illustrates, there are some specific features inside the Iranian license plates which are extracted by the deep architecture. These features include the height-to-width ratio (4.77:1 for typical license plates and 2:1 for free trade zone vehicles), the logo section with a blue background, the edges of the characters inside the license plate area, and other extractable features by high-pass filters. The system automatically ignores the license plates with small and unusual resolutions by a pre-defined threshold. In this stage, there is only one class, i.e., Iranian vehicle license plate, and all the existing objects with the features that represent this class will be detected through a classification solution. After detecting the license plate candidates, they are cropped and stored as separate objects for further processing. Now, by configuring and fine-tuning of the LPD deep module according to the presented description, the region of interest for CR is limited to one or several objects. This flow of the architecture results in a huge reduction in the misclassification of characters in the next phase.

B. CHARACTER RECOGNITION NETWORK

The input of the second network is a set of Iranian vehicle license plate images resized to 224×224 for the recognition process. As for the LPD network, the system looks at the CR phase as a separate classification problem. Accordingly, the segmentation and character recognition stages are combined in this architecture. The second network utilizes YOLO v.3 as well due to the requirement for classification of small objects, i.e., digits and characters. In this stage, the network should classify 28 different classes, where ten of them are digits, and the remaining eighteen classes are characters. As it has been described in Section I, the font size for the two rightmost items is smaller than the other five digits in the leftmost

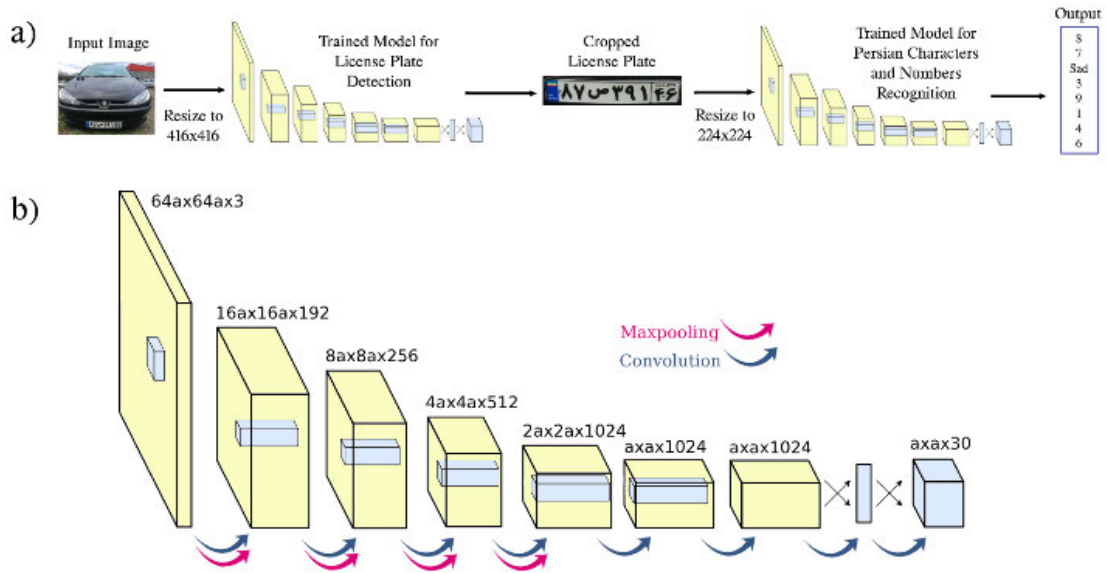


FIGURE 4. The deep architecture used in the proposed method: a) the sequential steps to extract the character sequence inside a given license plate, and b) the architecture of the employed YOLO v3 used in both LPD and CR steps with different configurations.

section of the Iranian vehicle license plates. Consequently, we have defined a threshold based on the two rightmost digits for the class resolutions.

After recognizing all the candidates of Persian characters/digits in a given license plate, they will be sorted in descending order based on their confidence scores. In this regard, candidates with higher confidence scores have more chances to be selected as the recognized class. There is also a threshold θ_{CS} that leads to ignorance of the candidates with confidence scores less than a pre-defined value. We have chosen 0.6 for θ_{CS} , which means that only the candidate classes with confidence scores more than 0.6 are acceptable, and the remaining classes should be ignored in an ordered queue, named as the rejection list. This process will result in multiple predicted classes with a high probability of correct classification. If the number of final candidates with acceptable confidence scores was more than eight, i.e., the number of characters in a typical Iranian car license plate, only the first eight are chosen and concatenate together to produce the final character sequence.

C. DATASETS

It has to be noted that the training data are color samples with various resolutions. However, we discuss the performance of the system for grayscale images in Section IV. The main reason for collecting color images is that to the best of our analysis, most of the commercial ALPR systems use color video cameras. For night-time license plate recognition, they need illumination sources to provide lighting for the surveillance zone. However, infrared cameras produce grayscale images that can provide a better representation of the photos in low illumination conditions, like night and late afternoons.

Additionally, the training data are collected from realistic camera outputs existing at tollbooths and organization entrance gates. These wide-angle security cameras are typically mounted 1.5 to 2.5 meters above the ground and cover a range of 3 to 10 meters. Comparing to the high-quality cameras installed in highways or city junctions where the coverage area can be even more than 15 meters, the cameras used in entrance gates aim to capture images of the vehicle with a quality suitable for character recognition. We divide this section into two different parts to introduce the datasets used for training and testing the system.

1) LICENSE PLATE DETECTION DATASETS

To train and evaluate the proposed method based on realistic Iranian license plate images, we have utilized three different datasets to cover various illumination, shadow, noise, and weather conditions. Table 1 illustrates the main characteristics of the mentioned datasets in detail. Accordingly, the first dataset is gathered from the Fashtam semi-industrial complex with the contribution of Guilan Science and Technology Park (GSTP). Most of the images of this dataset are captured in normal light circumstances with different weather conditions. The second dataset is gathered from Roudbar toll road with the supervision of the Iranian Traffic Police, which contains the images of the vehicles entering the toll collection zone. The image acquisition system in this toll road is a camera mounted on a two-meters pole to cover a ten-meters distance from the tollbooth. Since these two datasets include all the possible characteristics and challenges of the real condition vehicle surveillance systems, we have utilized them for both test and train stages. Finally, to evaluate the performance of the proposed method on a wide range

TABLE 1. Overall characteristics of the datasets (used for both training and testing the deep networks).

Dataset name	Number of data	Description
Fashtam semi-industrial complex driveway security camera outputs	5,219	<ul style="list-style-type: none"> Resolution: 1100*624 Color images Used for training and test Different weather and illumination conditions Various poses of the vehicles (rear/back) Multiple views of vehicles and license plates
Roudbar toll road camera outputs	6,371	<ul style="list-style-type: none"> Resolution: 710*379 Color images Used for training and test Different illumination conditions Contains challenging scenarios like occlusion and noise
Collected real-condition data	3,982	<ul style="list-style-type: none"> Resolution: various Used for test only Contains both challenging and straightforward scenarios Color and grayscale images (2733 color and 1249 grayscale)
Total	15,572	

of real condition data, we have collected several images of the vehicles with Iranian vehicle license plates. Some of these images are captured using mobile phone cameras, while others have been gathered from realistic surveillance systems. In contrast to the two previous datasets which are utilized for both training and testing purposes, the images of the third dataset are not used for training the model. It should be noted that the third dataset includes 1249 grayscale images to experiment with the performance of the system in the output of the infrared cameras.

Accordingly, these instances are collected explicitly for training and evaluating of Iranian ALPR systems. The dataset consists of more than 15000 vehicle images, captured in realistic environmental conditions. In contrast with some of the large-scale Iranian vehicle license plate datasets [41]–[43], our datasets cover all types of Iranian vehicle license plates, even free trade zone license plate samples. On the other hand, these images are the output of real functional systems, which makes them more challenging compared to the standard conditions. Fig. 5 shows some instances of the mentioned datasets as well. As can be seen, different shooting angles and vehicle orientations, weather and illumination variations, harsh light and shadow conditions, various types of license plates, and both color and grayscale samples are included.

Since there were highly limited instances for some of the particular Iranian vehicle license plates, we have ignored them for training the character recognition stage. Hereby, eight out of 26 license plate variations belonging to armed forces (three variants), political (two variants), protocol, agricultural, and historic vehicles were not taken into account for the CR training purpose. The remaining instances, including

**FIGURE 5.** Sample instances of the utilized datasets: a) Fashtam complex driveway security camera outputs, b) Roudbar toll road camera outputs, and c) collected data for the experiment.

private cars (13 different characters), taxis, public vehicles, governmental, police, and private cars of people with disabilities exist in these datasets.

2) CHARACTER RECOGNITION DATASET

For the character recognition stage, we have utilized a dataset containing all the digits and typical characters existing on Iranian vehicle license plates. As a result, we have used all the cropped license plates produced by the LPD deep network. The characteristics of the CR dataset are presented in Table 2. As can be seen, the dataset contains ten digits, 17 characters, and one symbol for private cars of the disabled people. We have considered an equivalent label for each of the mentioned classes.

According to Table 2, there were only a limited number of samples among some of the classified characters. These classes include “0” (only possible in the rightmost character of the license plate), “Alef” (governmental vehicles), “M”, “N”, “Ayn” (public vehicles), “Taxi”, and “Janbaz” (private cars of people with disabilities). To provide more samples of these classes, we used data augmentation based on reference [46] mentioned in Table 2. Accordingly, each of the scarce classes of characters was fed to the designed augmentation module to increase the size of the dataset artificially. In this section, we have applied Gaussian Blur, Contrast Normalization, Additive Gaussian Noise, Multiply, Rotation, Translation, Scaling, and Shear augmentation techniques to enlarge the scarce samples. Fig. 6 shows some of the augmented license plates. As can be seen, the output license plates cover a more extensive range of different poses and states of the mentioned character, which may occur in practice.

D. TRAINING THE DEEP NETWORKS

To train the LPD deep network, we have used 85 percent of the images existing in Fashtam complex and Roudbar toll road (4437 and 5416 instances, respectively). The training samples were selected randomly from each dataset, and the remaining 15 percent did not feed to the network. When we gathered the data, a labeling process to determine the classes, i.e., Iranian vehicle license plates, was needed. We have employed YOLO BBox Annotation Tool [47] for the

TABLE 2. Existing Persian characters in Iranian vehicle license plates and their English equivalent and number of characters used in the training phase.

Persian Character/Symbol	Equivalent Label	Number of data
۰	0	2501 (augmented)
۱	1	3495
۲	2	3930
۳	3	2745
۴	4	5774
۵	5	3610
۶	6	5753
۷	7	3736
۸	8	3583
۹	9	3528
الف	Alef	2517 (augmented)
ب	B	2511
پ	P	2519
ت	T	2505
ث	He	2558
ج	D	2504
چ	Sin	2445
ح	Sad	2515
خ	Ta	2512
د	Gh	2482
ذ	L	2502
ر	M	2500 (augmented)
ز	N	2558 (augmented)
س	V	2509
ش	Ye	2491
ص	Ayn	2508 (augmented)
ط	Taxi	2551 (augmented)
ظ	Janbaz	2502 (augmented)
Sum		83,844

**FIGURE 6.** Various augmentation processes applied to scarce characters to enlarge the dataset.

annotation process. Firstly, the annotation/labeling stage was performed manually for a set of 1200 random instances of the training dataset. The LPD network was trained based on this small portion of data. After generating a primitive LPD model, the remaining data were fed to the network to produce

TABLE 3. Overall specifications of the employed hardware system.

Resource	Specifications
CPU	Intel Core i7 930 2.8 GHz
GPU	NVidia GeForce GTX 1080 – 8GB Memory
RAM	48 GB DDR3
cache	8 MB

annotations on the images. With this, the labeling process was performed automatically for the remaining images, and a validation check was applied to them. Finally, all of the training samples, including 9853 images, were fed to the LPD network to provide the final model. To train the network, we have employed a personal computer with specifications presented in Table 3. The whole system was implemented in Python 3.7, and we have utilized CUDA, TensorFlow-GPU, and CUDNN for implementations.

YOLO v.3 uses three YOLO layers, each with 255 outputs as default, makes it able to cover 80 classes of COCO dataset. The relation between the number of outputs and anchors of the deep network is shown in Equation (2):

$$\text{output} = (BC + OC + CC) * \text{anchors} \quad (2)$$

where BC refers to box coordinates, OC refers to object confidence, and CC shows class predictions. In this regard, the default configurations of YOLO v.3 are $BC = 4$, $OC = 1$, and $CC = 80$, whereby multiplying the sum of these values in the number of anchors, i.e., 3, the output = 255 is obtained. For tuning the deep networks, since the number of classes in the proposed method is different, we have configured these values based on our class count to filter the output results. For LPD, there is only one class, so that the output becomes $(4 + 1 + 1) * 3$ that is equal to 18. Similarly, in CR, there are 28 classes, which results in $(4 + 1 + 28) * 3$ or 99 outputs. The learning rate for the LPD phase was $1 \times e-4$, and the number of epochs for this task was 100. However, the training step stopped after 34 epochs, where there was no considerable improvement in loss function (loss did not improve from 1.65037).

The training process was similar to the LPD network training and was performed on the same hardware. The number of classes in this phase was 28. Thus, at first, we selected at least 100 samples of the license plates containing each of the classes for manual annotation. Similar to the previous stage, we recursively perform the automatic annotation phase along with manual validation. Finally, the dataset presented in Table 2 was produced and has been fed for training to the second deep network. In this stage, the training process for CR was done in 45 epochs, and loss did not improve from 2.46 after that. Fig. 7 shows some samples of manual/automatic labeling outputs.

IV. PERFORMANCE EVALUATION

To validate the effectiveness of the proposed method in both Iranian vehicle license plate detection and Persian character recognition, we have employed the datasets mentioned in



FIGURE 7. Some instances of the labels applied on cropped license plates by automatic labeling and manual validation.

TABLE 4. Accuracy in the detection of Iranian vehicle license plates.

Dataset	# of data	# of samples for testing	# of detected license plates	Accuracy
Fashtam semi-industrial complex driveway security camera outputs	5,219	782	771	0.986
Roudbar toll road camera outputs	6,371	955	946	0.991
Collected real-condition data	3,982	3,982	3,874	0.973
Total	15,572	5,719	5,591	0.978

the previous section. For LPD, as it has been noted before, we have used 85 percent of Fashtam complex and Roudbar toll road datasets samples for training, and the remaining 15 percent of the sample was utilized for evaluation. This portion includes 1737 images where 782 instances are from the Fashtam complex, and 955 of them belong to the Roudbar toll road dataset. The third dataset was gathered by the authors of this paper for evaluation purposes only and did not participate in the training phase. On the other hand, and for CR, the same segmentation of train and test data was considered, and 15 percent of the data introduced in Table 2 were used for evaluation.

In order to evaluate the performance of the system, we have used Precision and Recall measures. Precision can be calculated by dividing the number of correctly classified samples (license plates in LPD and characters/digits in UCR) on the total number of classified instances. Similarly, Recall can be calculated by dividing the number of correctly classified samples by the total number of ground truth data. Precision and recall are mathematically defined as follows:

$$Precision = TP / (TP + FP) \quad (3)$$

$$Recall = TP / (TP + FN) \quad (4)$$

where TP, FP, and FN represent true positive, false positive, and false negative, respectively.

A. LICENSE PLATE DETECTION

The performance of the LPD network was evaluated on the proposed datasets. In this experiment, we checked the



FIGURE 8. Sample outputs of the proposed LPD module on real condition data.

TABLE 5. Performance of Persian character recognition network.

Type of data	Total number of samples	# of samples for test	Precision	Recall
Digits (0-9)	37,189	5,578	0.979	0.994
Characters	46,655	6,998	0.979	0.989
Total	83,844	12,576	0.979	0.991

detected bounding boxes referring to the license plates against the coordinates available on the ground-truth labels. Table 4 shows the performance of the system for Iranian vehicle license plate detection on different datasets. As an evaluation criterion, the Intersection over Union introduced in Equation (1) is used to determine the area of overlap between the ground-truth labels and generated bounding boxes were calculated. According to Table 4, the precision value is much more accurate compared to the realistic LPD systems introduced in the previous section.

Additionally, Fig. 8 shows some sample outputs of the LPD stage. The second row of the figure indicates some gathered highway cameras grayscale images, in which the system was able to detect them as well, while there were no grayscale images in the training set. It shows that the proposed LPD module can also work on the outputs of infrared cameras.

B. PERSIAN CHARACTER RECOGNITION

Table 5 indicates the performance of the CR deep network in brief. The experiment was performed on 15 percent of the remaining data from the Persian characters dataset introduced in Table 2. According to this table, the precision value for both Persian digits and characters are nearly equal, while the recall factor for digits is better than for characters. The results indicate that the CR deep classifier can correctly recognize the characters in most of the cases.

Besides, Fig. 9 shows some instance outputs of the character recognition stage in brief. As can be seen, the system



FIGURE 9. Sample outputs of the proposed Persian character recognition stage.

can find the characters and numbers in variations of Iranian vehicle license plates effectively. The angle shot and resolution of the license plate images did not affect the accuracy to a large extent. It is worth noting that for the license plates of the free trade zone vehicles, only the Persian digits have been classified, and their translated English sequence is ignored due to lack of data for training.

C. END-TO-END RECOGNITION ACCURACY

This experiment indicates the performance of the proposed system for commercial usage. The system should be able to correctly detect and crop the existing license plates and extract the correct text from there. In this regard, if the LPD stage fails to detect all the existing license plates or the CR deep network fails to correctly classify all the characters and digits compared to the ground-truth classes, the end-to-end accuracy degrades. For evaluation, the LPD stage test data presented in Table 4 was fed to the system, and the number of correctly extracted character sequences calculated. Table 6 illustrates the end-to-end experiment results in detail. The test data contain vehicle images captured in different illumination and brightness conditions, various resolutions, and different shooting angles. Since there are different sizes of the image in this dataset, we claim that our method is not dependent on image resolution and can cope with low quality small images to a large extent.

D. EXECUTION TIME

To prove the real-time performance of the proposed method, we have calculated the time distribution for a random sample of 8000 images from the introduced datasets. Fig. 10 shows the evaluation results in detail. Accordingly, the license plates of most vehicles have been detected in 50 milliseconds, while characters and digits are recognized in 80 milliseconds in

TABLE 6. The performance of the proposed approach in extracting the correct character sequence out of a given image.

Dataset	# of data	# of correctly detected license plates	# of correct LP text extraction	Accuracy
Fashtam semi-industrial complex driveway security camera outputs	782	771	744	95.14%
Roudbar toll road camera outputs	955	946	921	96.44%
Collected real-condition data	3,982	3,874	3,771	94.70%
Total	5,719	5,591	5,436	95.05%

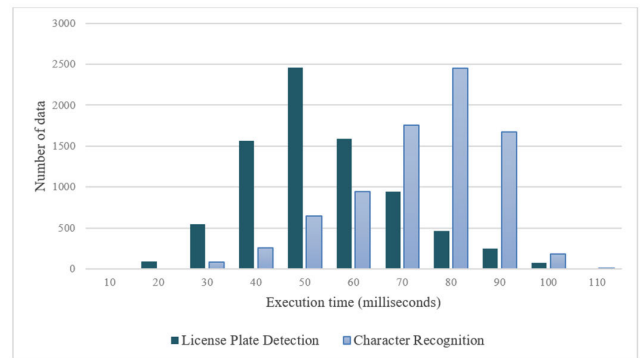


FIGURE 10. Elapsed time for license plate detection versus character recognition stages on the generated dataset.

most of the cases. We obtained these execution times automatically where the outputs of each deep networks were produced. In this regard, the LPD network requires an average of 54.18 milliseconds to detect license plates inside an image. The number of available license plates, the resolution of the input image, and the challenging factors like low illumination have significant impacts on this stage. Furthermore, an average time of 73.78 milliseconds is needed for the CR network to classify all the characters inside a given license plate image. The end-to-end average time to extract a character sequence out of a given vehicle image is 119.73 milliseconds, which proves the real-time performance of the proposed method.

E. COMPARISON OF OUR FINE-TUNED DNN WITH OTHER DNNs

In this experiment, we have analyzed the speed and accuracy of the proposed two-stage method by employing different deep learning-based object detectors. Hereby, we have utilized some well-known deep networks, including Fast R-CNN, Faster R-CNN, YOLO, and YOLO v.2 instead of each of the proposed fine-tuned YOLO v.3 architectures for both LPD and CR phases. The experiment was an end-to-end recognition analysis, and the test data was the same as what was introduced in Section IV part C. Table 7 indicates the accuracy and speed of the proposed method compared to other deep learning architectures. Accordingly,

TABLE 7. The comparison among different deep architectures employed for Iranian car license plate detection and Persian character recognition.

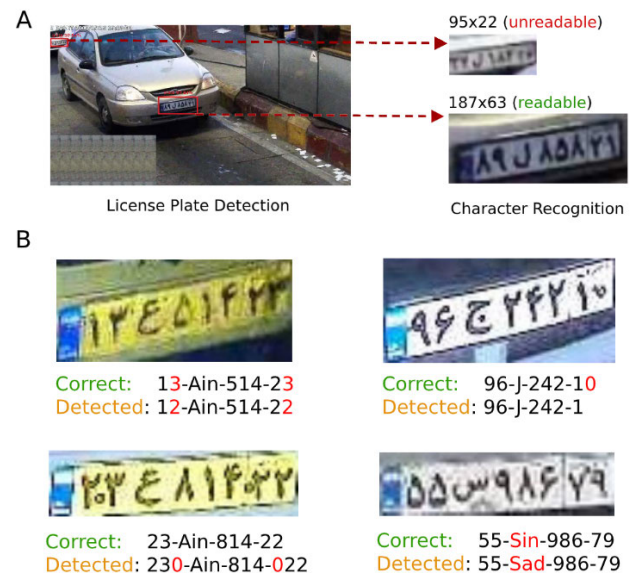
Method	# of data	# of correctly classified objects		End-to-end recognition accuracy	Average execution time	
		license plates	characters/digits		License plate detection	Character recognition
Fast R-CNN [27]	5,719	5,398	5,318	93.00%	139.774	159.006
Faster R-CNN [28]		5,452	5,367	93.84%	107.261	125.758
YOLO [23]		5,437	5,386	94.18%	74.729	90.239
YOLO v.2 [29]		5,402	5,335	93.29%	51.322	69.482
Proposed (YOLO v.3)		5,591	5,436	95.05%	54.183	73.778

**FIGURE 11.** Some of the given samples in which the system failed to localize the license plates.

the proposed method provides the best end-to-end recognition accuracy due to its 106-layered structure and processes the input image in real-time, with an average processing time of shorter than 120 milliseconds for end-to-end recognition. However, by employing YOLO v.2, the execution time dwindles to some extent, but the accuracy decreases as well. Due to the trade-off between real-time performance and excellent accuracy, choosing YOLO v.3 seems reasonable as it has been employed in the proposed method.

F. DISCUSSION

Although the proposed method provided significant accuracy and performance for the task of Iranian vehicle license plate detection, some challenging scenarios lead to a failure in localizing the license plates or misclassification of the characters. The system failures are due to the low variety of data for training in some cases, problems in tuning the

**FIGURE 12.** Some of the misclassified samples in the character recognition stage.

network, and the impacts of brightness and illumination. In this regard, Fig. 11 shows some of the images in which the LPD deep network failed to localize the license plates. As can be seen in sample A, if the image acquisition stage provides a partially visible license plate, the LPD network fails to detect it. Additionally, intense illumination along with the reflection of sunlight from the license plate negatively affects the performance in both color (samples D and F) and grayscale (sample B) images and results in failure or misclassification of the license plates.

In night-time situations, low illumination (sample E) may reduce the performance of this stage as well, and the images may even have some blurred noises (sample C) due to the vehicle's headlights beams. Furthermore, the proposed method failed to localize the license plate in sample G because the border of the license plate is occluded with snow. Finally, sample H shows a fake license plate that has dissimilar descriptions compared to a real one.

On the other hand, and in the Persian character recognition stage, most of the misclassifications were due to the low resolution of the input cropped license place, which leads to a

failure in recognizing the character. As it has been mentioned in Section I, the accuracy of the LPD stage directly impacts the CR model's precision. Thus, if the cropped license plates belong to the vehicles far from the camera, even if the LPD model detects the license plate, the inner characters may not be legible. Fig. 12-A shows an instance of a small license plate with unreadable inner text, which leads to CR output failure. The second important factor for correct classification in this stage is the similarity among some of the individual digits and characters of the Persian language. In our experiments, the digits "2" (labeled as "2") and "3" (labeled as "3") are incorrectly classified instead of each other in some cases, especially when there is a blurring effect in the license plate. Similarly, the characters "س" (labeled as "Sin") and "ص" (labeled as "Sad") face the same challenge. Another problematic case is "0" (labeled as "0") which is very tiny. In this regard, this digit could not be recognized in some cases, or it could be mistaken with the screws used to install the license plate on the bumper. These challenging scenarios have been illustrated in Fig. 12-B.

We believe most of the mentioned challenges can be handled by data augmentation and dataset clean-up. By widening the range of correctly labeled samples for training through reviewing the labels and providing even more variations of vehicle images and cropped license plates, an increase in accuracy for both LPD and CR models is expected. Additionally, by re-tuning the deep models through adjusting or resetting the existing hyperparameters and applying some other techniques like learning rate decay, we can prevent overfitting problems as well.

V. CONCLUSION

This paper proposed a unified License Plate Detection (LPD) and Character Recognition (CR) system for Iranian vehicle license plates that has the advantages of real-time performance and high accuracy. Our method utilizes two sequential YOLO v3 deep networks for this purpose. Hereby, we have trained the system with realistic condition data acquired in various weather, noise, and illumination conditions. These data were collected from functioning driveway security cameras installed for surveillance goals. Accordingly, a vast range of images captured in both challenging and straightforward realistic conditions has been covered in the training process. The proposed approach has been tested on separate realistic data, resulted in an accuracy of for the LPD stage, while the precision and recall measures for the Persian character recognition stage were 0.979 and 0.991, respectively. Furthermore, the system works in real-time with an average processing time of 119.73 milliseconds for each image or video frame to extract the license plate character sequence.

ACKNOWLEDGMENT

The authors would like to thank everyone associated with Guilan Science and Technology Park that helped in acquiring, labeling, and analyzing data. They would also appreciate the assistance of Dr. Alireza Akoushideh, Dr. Ali Basti, and

Dr. Hashem Mirbahari for their kindly feedbacks and provisions in this project. Correspondingly, the authors would also like to show gratitude to their colleagues in DadeKavan Khazar Pouya company for providing the hardware platform for training and testing the system. The presented approach is planning to be released as a commercial ALPR software for Iranian vehicle license plates.

REFERENCES

- [1] M. Bommers, A. Fazekas, T. Volkenhoff, and M. Oeser, "Video based intelligent transportation systems—State of the art and future development," *Transp. Res. Procedia*, vol. 14, pp. 4495–4504, Jan. 2016.
- [2] L. Jin, H. Xian, J. Bie, Y. Sun, H. Hou, and Q. Niu, "License plate recognition algorithm for passenger cars in Chinese residential areas," *Sensors*, vol. 12, no. 6, pp. 8355–8370, Jun. 2012.
- [3] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A robust and efficient approach to license plate detection," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1102–1114, Mar. 2017.
- [4] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A State-of-the-Art review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, Feb. 2013.
- [5] G. L. Corneto, F. A. da Silva, D. R. Pereira, L. L. de Almeida, A. O. Artero, J. P. Papa, V. H. C. de Albuquerque, and H. M. Sapia, "A new method for automatic vehicle license plate detection," *IEEE Latin Amer. Trans.*, vol. 15, no. 1, pp. 75–80, Jan. 2017.
- [6] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and networked vehicle surveillance in ITS: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 25–48, Jan. 2017.
- [7] M. A. Khan, M. Sharif, M. Y. Javed, T. Akram, M. Yasmin, and T. Saba, "License number plate recognition system using entropy-based features selection approach with SVM," *IET Image Process.*, vol. 12, no. 2, pp. 200–209, Feb. 2018.
- [8] X. Liu, Z.-R. Peng, and L.-Y. Zhang, "Real-time UAV rerouting for traffic monitoring with decomposition based multi-objective optimization," *J. Intell. Robot. Syst.*, vol. 94, no. 2, pp. 491–501, May 2019.
- [9] B. Nemade, "Automatic traffic surveillance using video tracking," *Procedia Comput. Sci.*, vol. 79, pp. 402–409, Jan. 2016.
- [10] J. Gao, L. Sun, and M. Cai, "Quantifying privacy vulnerability of individual mobility traces: A case study of license plate recognition data," *Transp. Res. C, Emerg. Technol.*, vol. 104, pp. 78–94, Jul. 2019.
- [11] Y. Zhang, B. Song, X. Du, and M. Guizani, "Vehicle tracking using surveillance with multimodal data fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2353–2361, Jul. 2018.
- [12] L. Janowski, P. Kozłowski, R. Baran, P. Romaniak, A. Glowacz, and T. Rusc, "Quality assessment for a visual and automatic license plate recognition," *Multimedia Tools Appl.*, vol. 68, no. 1, pp. 23–40, Jan. 2014.
- [13] S. Mahalakshmi and R. S. Kumar, "Smart toll collection using automatic license plate recognition techniques," in *Proc. Int. Conf. Comput., Anal. Netw.*, Singapore, 2017, pp. 34–41.
- [14] R. Sivakumar, G. Vignesh, V. Narayanan, S. Prakash, and V. Sivakumar, "Automated traffic light control system and stolen vehicle detection," in *Proc. IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2016, pp. 1594–1597.
- [15] P. Saini, K. Bidhan, and S. Malhotra, "A detection system for stolen vehicles using vehicle attributes with deep learning," in *Proc. 5th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Oct. 2019, pp. 251–254.
- [16] S. Khazaei, A. Tourani, S. Soroori, A. Shahbahrani, and C. Y. Suen, "A real-time license-plate detection method using a deep learning approach," in *Proc. 2nd Int. Conf. Pattern Recognit. Artif. Intell.*, 2020, pp. 425–438.
- [17] P. S. Ha and M. Shakeri, "License plate automatic recognition based on edge detection," in *Proc. Artif. Intell. Robot. (Iranopen)*, Qazvin, Iran, Apr. 2016, pp. 170–174.
- [18] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, "Vertical-edge-based car-license-plate detection method," *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 26–38, Jan. 2013.
- [19] T. K. Reddy, A. K. Reddy, P. D. Reddy, and R. Muthu, "Efficient license plate detection by unique edge detection algorithm and smarter interpretation through IoT," *Comput. Res. Repository*, vol. 1710.10418, 2017. [Online]. Available: <http://arxiv.org/abs/1710.10418>

- [20] M. R. Asif, Q. Chun, S. Hussain, M. S. Fareed, and S. Khan, "Multinational vehicle license plate detection in complex backgrounds," *J. Vis. Commun. Image Represent.*, vol. 46, pp. 176–186, Jul. 2017.
- [21] M. S. Al-Shemarry, Y. Li, and S. Abdulla, "An efficient texture descriptor for the detection of license plates from vehicle images in difficult conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 553–564, Feb. 2020.
- [22] Q. Li, "A geometric framework for rectangular shape detection," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4139–4149, Sep. 2014.
- [23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [24] F. Q. Lauzon, "An introduction to deep learning," in *Proc. 11th Int. Conf. Inf. Sci., Signal Process. Their Appl.*, Montreal, QC, Canada, Jul. 2012, pp. 1438–1439.
- [25] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in *Proc. Int. Conf. Commun. Signal Process.*, Chennai, India, Apr. 2017, pp. 0588–0592.
- [26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [27] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [29] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
- [30] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [31] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [32] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2351–2363, Sep. 2017.
- [33] 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.
- [34] J. Zhang, Y. Li, T. Li, L. Xun, and C. Shan, "License plate localization in unconstrained scenes using a two-stage CNN-RNN," *IEEE Sensors J.*, vol. 19, no. 13, pp. 5256–5265, Jul. 2019.
- [35] 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.
- [36] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Goncalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–10.
- [37] H. Sun, M. Fu, A. Abdussalam, Z. Huang, S. Sun, and W. Wang, "License plate detection and recognition based on the YOLO detector and CRNN-12," in *Proc. Int. Conf. Signal Inf. Process., Netw. Comput.*, Singapore, 2018, pp. 66–74.
- [38] Hendry and R.-C. Chen, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image Vis. Comput.*, vol. 87, pp. 47–56, Jul. 2019.
- [39] A. Tourani, S. Soroori, A. Shahbahrami, S. Khazaei, and A. Akoushdeh, "A robust vehicle detection approach based on faster R-CNN algorithm," in *Proc. 4th Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Mar. 2019, pp. 119–123.
- [40] S. Soroori, A. Tourani, and A. Shahbahrami, "Employing deep learning approach for automatic license plate recognition: A review," in *Proc. 3rd Int. Conf. Soft Comput.*, 2019, pp. 1–6.
- [41] R. Serajeh, "Large Iranian car license plate database," in *Proc. 2nd Conf. Intell. Road Transp. Syst.*, Tehran, Iran, 2016, pp. 1287–1294.
- [42] M. M. Dehshibi and R. Allahverdi, "Persian vehicle license plate recognition using multiclass AdaBoost," *Int. J. Comput. Electr. Eng.*, vol. 4, no. 3, pp. 355–358, 2012.
- [43] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An Iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1690–1705, Aug. 2014.
- [44] 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.
- [45] H. Li, P. Wang, and C. Shen, "Toward end-to-end car license plate detection and recognition with deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1126–1136, Mar. 2019.
- [46] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, and A. Kraft, "Imgaug," GitHub Repository, Tech. Rep., 2020. [Online]. Available: <https://github.com/aleju/imgaug>
- [47] Ybat—YOLO BBox Annotation Tool. Accessed: Aug. 21, 2020. [Online]. Available: <https://github.com/drainsunsun/ybat>



ALI TOURANI received the B.Sc. and M.Sc. degrees in computer software engineering from the University of Guilan, in 2013 and 2019, respectively.

He is currently working as a Research Software Engineer with the Guilan Science and Technology Park Business Incubator and a Visiting Lecturer with the University of Guilan for several courses, including computational intelligence and software laboratory in Rasht, Iran. From 2018 to 2020, he was the Chief of the Student Branch of the "Iranian Society of Machine Vision and Image Processing" at the University of Guilan and held several workshops there. He also has the experience of being a teacher assistant for artificial intelligence, parallel processing, and multimedia systems courses. His research interests include machine learning, object detection/tracking, and image/video processing, besides employing these methods in video-based intelligent transportation systems.

Dr. Tourani's awards and honors include the Best Student Researcher with the Department of Computer Engineering, University of Guilan, ranked 1st among all of the Computer Software Engineering students for three sequential semesters, and top 15 business planners award from the 10th Sheikh Bahai National Technopreneurship Festival.



ASADOLLAH SHAHBAHRAMI received the B.Sc. and M.Sc. degrees in computer engineering (hardware and machine intelligence) from the Iran University of Science and Technology and Shiraz University, in 1993 and 1996, respectively, and the Ph.D. degree from the Delft University of Technology, in September 2008.

He was offered a faculty position with the Department of Electrical Engineering, University of Guilan. He has been working at the University of Guilan since August 1996. In January 2004, he joined the Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft, The Netherlands, as a full-time Ph.D. student, under the supervision of Prof. Stamatis Vassiliadis and Dr. Ben Juurlink. He has been an Assistant Professor position with the Department of Computer Engineering, University of Guilan. His research interests include advanced computer architecture, image and video processing, multimedia instructions set design, reconfigurable computing, parallel processing, and SIMD programming.



SAJJAD SOROORI received the B.S. degree in computer software engineering from the University of Guilan, Iran, in 2013, where he is currently pursuing the M.Sc. degree in software engineering.

His research interests include real-time object detection in videos and tracking them by deep learning approaches, development of NLP processes in the Persian language, and study of new deep learning based solutions in computer vision.

Dr. Soroori is a member of the Computer Vision Association of the University of Guilan since 2018, and has published two conference papers related to vehicle detection and automatic license plate recognition.



SAEED KHAZAEI received the Ph.D. degree from Concordia University, Montreal, Canada, in 2020. He has also been awarded several times related to his research such as the “Concordia Award of Excellence,” “Concordia Acceleration Award,” “Concordia University Conference and Exposition Award,” and “Concordia GSSP Award.” He has been in contact with several universities and companies to increase his research productivity. He was a full-time faculty member at Azad University and a Visiting Lecturer at the University of Guilan, Iran. He is currently with the Center for Pattern Recognition and Machine Intelligence, Concordia, as a Research Assistant. He does research in computer and society, data mining, image processing, and pattern recognition. He has published 19 journal or conference papers, and three book chapters in the field of data mining, pattern recognition, and image processing. He was also a reviewer for several international journals. He has also the experience of collaborating with Nanchang University as a Visiting Scholar for a full semester.



CHING YEE SUEN (Life Fellow, IEEE) received the Ph.D. degree from UBC (Vancouver) and the master’s degree from The University of Hong Kong.

He has served as the Chairman of the Department of Computer Science and as the Associate Dean (Research) of the Faculty of Engineering and Computer Science, Concordia University. He is the Director of CENPARMI and the Concordia Honorary Chair on AI and Pattern Recognition. He is a recipient of numerous awards, including the Gold Medal from the University of Bari, Italy (2012), the IAPR ICDAR Award (2005), the ITAC/NSERC National Award (1993), and the “Concordia Lifetime Research Achievement” and “Concordia Fellow” awards (2008 and 1998, respectively), and the “Teaching Excellence Award” given by the Concordia Council of Student Life in 1995. He is a Fellow of the IAPR (1994), and the Academy of Sciences of the Royal Society of Canada (1995). He is currently the Emeritus Editor-in-Chief of “*Journal of Pattern Recognition*,” an Adviser of “*Pattern Recognition Letters*,” and an Editor of five other journals, i.e., “*Expert system with Application*,” “*Signal, Image and Video Processing*,” and an Editor of a new book series on “*Language Processing and Pattern Recognition*.” He is not only the founder of three conferences: ICDAR, IWFHR/ICFHR, and VI, but has also organized numerous international conferences, including ICPR, ICDAR, ICFHR, ICCPOL, and as the Honorary Chair of numerous international conferences. In 1997, he created the IAPR ICDAR Awards, to honor both young and established outstanding researchers in the field of Document Analysis and Recognition.

• • •