sensors

Article

Barrier Access Control Using Sensors Platform and Vehicle License Plate Characters Recognition

Farman Ullah [1], Hafeez Anwar [1], Iram Shahzadi [1], Ata Ur Rehman [1], Shizra Mehmood [1], Sania Niaz [1], Khalid Mahmood Awan [2], Ajmal Khan [1] and Daehan Kwak [3,*]

[1] Department of Electrical & Computer Engineering, COMSATS University Islamabad-Attock Campus, Attock 43600, Pakistan
[2] Department of Computer Sciences, COMSATS University Islamabad-Attock Campus, Attock 43600, Pakistan
[3] Department of Computer Science, Kean University, Union, NJ 07083, USA
* Correspondence: dkwak@kean.edu

Received: 16 April 2019; Accepted: 18 June 2019; Published: 9 July 2019

Abstract: The paper proposes a sensors platform to control a barrier that is installed for vehicles entrance. This platform is automatized by image-based license plate recognition of the vehicle. However, in situations where standardized license plates are not used, such image-based recognition becomes non-trivial and challenging due to the variations in license plate background, fonts and deformations. The proposed method first detects the approaching vehicle via ultrasonic sensors and, at the same time, captures its image via a camera installed along with the barrier. From this image, the license plate is automatically extracted and further processed to segment the license plate characters. Finally, these characters are recognized with the help of a standard optical character recognition (OCR) pipeline. The evaluation of the proposed system shows an accuracy of 98% for license plates extraction, 96% for character segmentation and 93% for character recognition.

Keywords: barrier control; sensors platform; vehicle detection; license plate recognition; raspberry-pi; features extraction; machine learning algorithms

1. Introduction

The security sensitive areas of a country, such as classified defense areas, government buildings and military installations, are under constant surveillance to avoid potential threats. Such surveillance also extends to the vehicles that constantly access these areas. A vast majority of the currently installed systems use barrier gates that are either manually operated [1] or use vehicle identification based on radio frequency identification (RFID) technology [2]. In RFID-based systems, every vehicle has an RFID tag and RFID reader installed at a gate to identify authorized vehicles. Such systems automatize the access control process; however, the installation of RFID tag in each vehicle makes such systems costly. Alternatively, we propose using a combination of a sensors platform and camera system for automatic barrier access control. The approaching vehicle is automatically detected via the ultrasonic sensors while a camera captures the image of the front side of the vehicle. This image is then further processed to extract and recognize the license plate (LP) of vehicle for authorization. Consequently, the barrier is opened only for authorized vehicles.

An automatic license plate recognition (ALPR) system is instrumental in identifying a vehicle from the image of its LP. As a common rule in various parts of the world, the government issues LPs with fixed aspect ratios, fonts and backgrounds. However, arbitrarily designed LPs is an ever growing problem in countries like Pakistan, where the ALPR becomes a challenging task for a few reasons. First, the position of an LP is not fixed on the front side of the vehicle. Second, there exists a huge variation in the aspect ratios of the LPs. Third, the backgrounds of the LPs vary from on to the other. Finally, variations are also found due to non-uniform font styles and font sizes. Some of these variations are depicted in Figure 1.

A typical ALPR system mainly consists of the following steps [3,4]:

1. Image capturing device acquires an image or extract an image from a video
2. Localization and extraction of license plate in the acquired image
3. Character segmentation and recognition OCR in the extracted LP

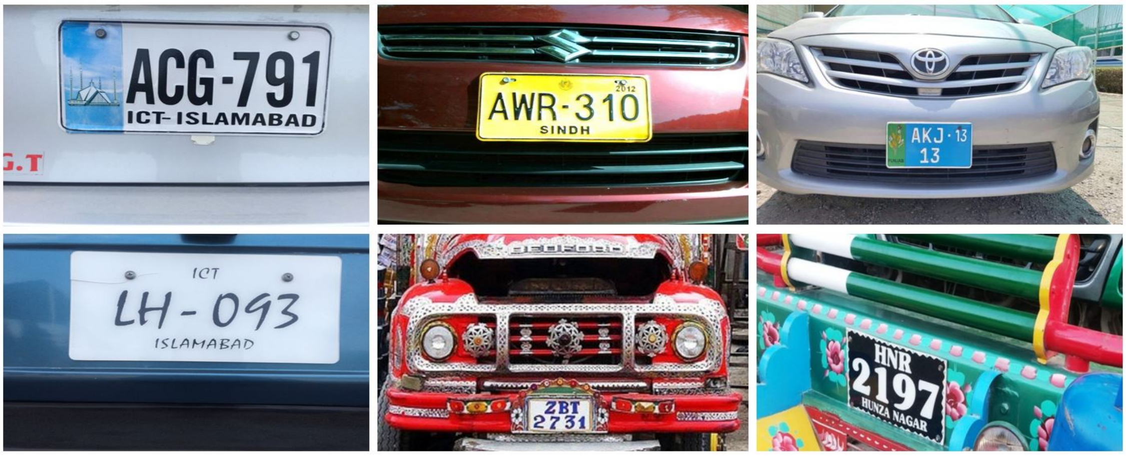


Figure 1. Overview of Pakistani license plates (LPs) with various background, foreground, characters fonts and font-sizes.

The ALPR process begins with LP localization and extraction from the vehicle image. LP localization techniques extract the rectangular bounding box or the text regions directly from the image [5]. Without any prior knowledge about the LP size and its location on vehicle, the entire image must be examined to extract the required LP region. We use a Canny edge detector-based method [6] followed by morphological operations and connected components detection to find the rectangular bounding box around the LP in vehicle image. The next step is the segmentation of the desired LP to extract individual characters for recognition. We propose a segmentation approach for characters that have variations in font size, style, and color. Finally, optical character recognition OCR is used to recognize the letters and digits of the extracted LP. To this end, we adapt the feature-based approach that extracts the features of each individual character. These feature include character contour, zoning of solid binary image character, and a skeleton of thin characters [7,8]. These features are used to train the model of a machine learning algorithm or classifier for character recognition. We evaluate a number of classifiers such as the support vector machine (SVM) [9,10], K-nearest neighbors (KNN) [11,12], artificial neural network (ANN) [13] and Decision Trees [14]. The main contributions of this paper are as follows:

1. Development of a prototype for barrier access control and then deploying it in a real world scenario.
2. An image dataset of challenging number plates commonly used in Pakistan with variations in background, position on vehicle, fonts and font styles.
3. Development of an algorithm for character extraction and segmentation of LPs having different background, position on vehicle, fonts and font styles.
4. An extensive performance evaluation of classifiers for optical character recognition.
5. A performance evaluation of the proposed system on two different hardware environments to select the one which is favorable for real-time application.

The rest of this paper is structured as follows: Section 2 outlines related work; Section 3 explains the proposed methodology; the dataset description, results and performance evaluation are discussed in Section 4; finally, Section 5 concludes the paper and outlines the future directions of the current research.

2. Related Work

In this section, we briefly introduce the related work about LP Localization in vehicle images, characters segmentation and characters recognition.

2.1. LP Detection and Localization

In an ALPR system, the starting step is LP detection and extraction. If the LP is not properly extracted, then the LP segmentation will be severely affected [15]. As a common practice, an LP has a rectangular shape. However, in the captured vehicle image there may be other rectangular objects such as the headlights. Therefore, for an effective segmentation, the properties and features of an LP such as its area and aspect-ratio, should be known beforehand. Tarabek et al. [16] proposed a connectivity based rectangular bounding-box extraction with fixed properties. The combination of edge detection and morphological operations is used for LP detection and localization [17–20]. Wang et al. [19] converted the RGB image to HSV color space and proposed a two-stage process for LP localization using color and edge information. Dun et al. [21] proposed an ALPR system for specifically yellow and blue Chinese LPs. A special threshold function was proposed to convert the RGB image to gray to highlight the yellow and blue colors. The transition between the LP background and characters are then used to remove the fake plates and reserve the real plate. In the final step, the accurate location is determined using character size and stroke width. Safaei et al. [22] proposed LP localization based on hierarchical saliency. The proposed algorithm has two steps: in the first step, the algorithm finds the saliency map and then using the connected component analysis detects the LP region. After finding connected components, a Sobel filter and a closing morphological operation is applied. It eliminates many non-number plate regions and then finds the most populated region using L_1-norm. Its result is then binarized using Otsu's method. The largest connected component covering the plate number is then cropped from the vehicle image.

2.2. Characters Segmentation and Extraction form LP

Character segmentation divides the LP into individual characters and digits. Character segmentation becomes challenging due to multi-color background and foreground of an LP. Tabrizi et al. [23] proposed LP segmentation using morphological operations such as dilation, hole filling, erosion, and characters width and height. Gazcón et al. [24] proposed a bounding box technique and its properties to extract characters from the cropped LP. A Convolutional Neural Network (CNN) based two-stage process is proposed [25] to segment and recognize characters (0–9, A–Z). Tarigan et al. [26] proposed an LP segmentation technique consisting of horizontal character segmentation, connected component labeling, verification and scaling. Horizontal and vertical projections of characters are used to segment the cropped LP [27]. Zheng et al. [28] proposed an improved blob detection algorithm to segment LP characters. The segmentation process consists of three steps: first, character height is estimated using the lower and upper boundaries; character width is estimated; and finally, the character is labeled using the block extraction algorithm.

2.3. LP Extracted Characters Recognition

One of the main components of ALPR is the automatic recognition of characters. Chen et al. [29] proposed SIFT based features extraction and matching these features in order to recognize the Chinese characters. A template matching based LP characters' recognition [30] has been proposed for Arabic characters, to recognize 27 alphanumeric characters (17 alphabets and 10 numeric) of fixed size 50 × 25. A tesseract OCR engine [31] with modification is used in Reference [28] for LP characters' recognition. Tabrizi et al. [23] proposed a hybrid approach of k-nearest neighbor (KNN) and multi-class support vector machine (SVM) for Iranian LP recognition. First, the KNN classifies the characters using the structural, horizontal and vertical features. Then the SVM classifier is applied to the zoning features. Gazcón et al. [24] compared the proposed intelligent template matching (ITM) with the artificial neural network (ANN). Compared to the traditional template matching technique the ITM constructs trees of the character's skeleton. These trees are used to compare with the tree obtained from the testing character skeleton. ITM showed higher accuracy and also minimized the recognition time. Wang et al. [32] proposed LP detection and recognition simultaneously in a single forward pass by using a deep neural network algorithm. In the first step of this algorithm, a number of convolutional layers are used to extract and discriminate the features of LP. After this, the proposed network detects the objects on a LP. This technique takes the low level convolutional features and generates a set of bounding boxes. In the last step, a bidirectional recurrent neural network (BRNN) with Connectionist Temporal Classification recognizes the LP characters. Björklund [33] proposed an ALPR system trained on synthetic data that has varying pose conditions and illumination levels and showed precision and recall of 93%. Table 1 presents the overall literature review of LP detection, LP region of interest extraction, characters' segmentation, and character recognition.

Table 1. Literature summary of LP detection, extraction, characters segmentation, and characters recognition.

Ref.	Methodology	Character Language (0–9, A–Z)	Dataset Size	Platform	Plate Localization	Character Segmentation	Character Recognition	Processing Time
[28]	Plate Detection:Viola and jones algorithm, Character Segmentation: Blob detection, Characters Recognition: *OCR*	(0–9) (A–Z)	Testing dataset (160) Training dataset (300)	PC with Pentium 2.8 GHz CPU	96.4%	98.2%	99%	0.1 s
[26]	Recognition: Genetic algorithms	(0–9) (A–Z)	220 Images	PC	N/A	N/A	85.97%	0.2 s
[23]	Characters Extraction & Segmentation: Morphological operations, Recognition: KNN and SVM	Persian characters	257 Images	PC	96.01%	95.24%	97.03%	N/A
[34]	Segmentation: Otsu's method, Character Extraction:Vertical projection, Recognition: Backward propagation Neural Network (BP-ANN)	(0–9) (A–Z)	Training: 2700 & Testing: 354 characters	CPU i3 2.9G and 2G memory	N/A	N/A	93.5%	N/A
[35]	Detection, Extraction, and Recognition: Convolutional Neural Network (CNN)	(0–9) (A–Z) Excluding 'O'	Testing USA (328) Europe (550)	PC	99.0% USA 93.64% Europe	N/A	93.445% USA 94.54% Europe	N/A
[36]	Plate Detection:Modified visual attention model, Segmentation: Vertical and horizontal projection, Recognition:CNN and SVM	Chinese (0–9) (A–Z)	Testing: Chinese (620), 0–9 and A–Z (680): Training Chinese (930), 0-9 and A-Z (1020)	PC	98.9%	N/A	98.3% Chinese characters 99.1% (numerals and alphabets)	0.14 s
[24]	Through searching rectangles (plate extraction) Bounding Box technique (character extraction) ANN and ITM (character recognition)	(0–9) (A–Z)	Testing data (73)	Pentium Dual core, 1.73 GHz, 2 GB RAM	N/A	N/A	99.09% (ANN) 91.1% (ITM)	0.5 ms (ANN) 0.75 ms (ITM)
[27]	YOLO detector (vehicle and LP detection) CNN (character segmentation and recognition)	(0–9) (A–Z)	Data set (4500)	NVIDIA Titan XP GPU	100%	99.75%	97.83%	28.3 ms

3. Proposed System

This section explains the proposed architecture including the main functions from vehicle detection to the barrier control mechanism. Figure 2 illustrates the block diagram of the proposed system while Figure 3 depicts the algorithm flowchart of the proposed system. Following are the main steps.

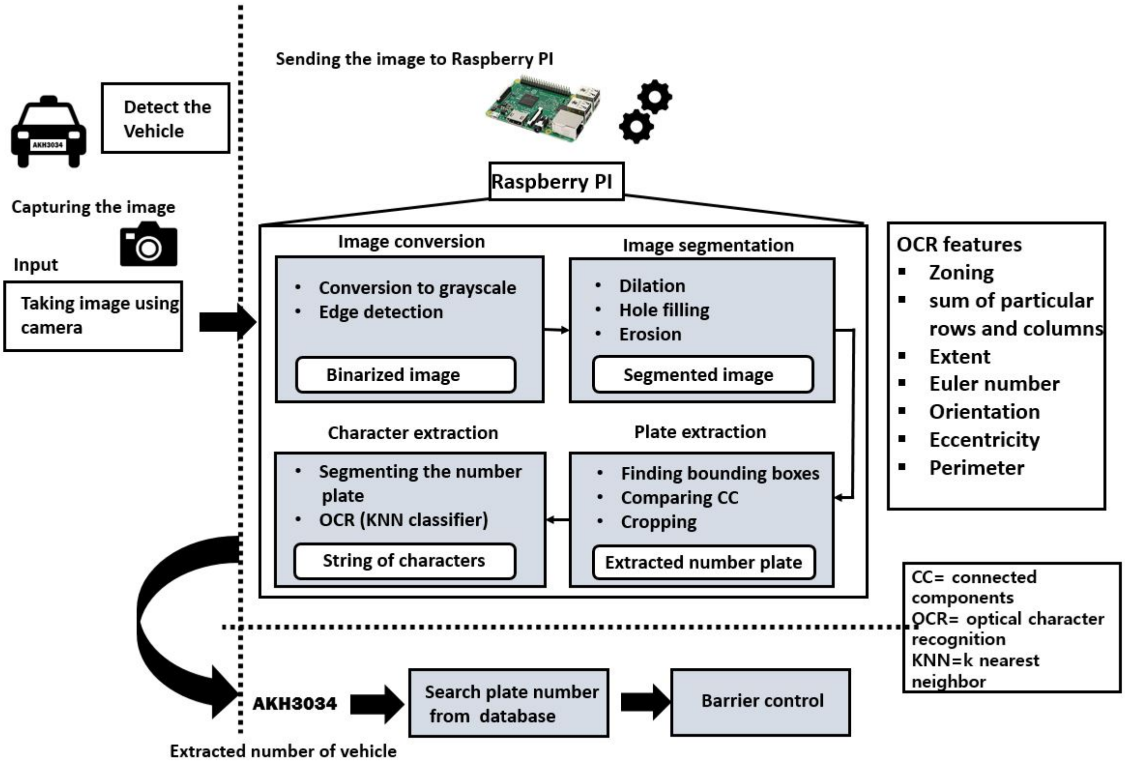


Figure 2. The proposed architecture of barrier access control for vehicle entrance using sensors platform and an image-based LP recognition.

1. Vehicle arrival detection and image acquisition
2. Image pre-processing and edge image generation
3. Image segmentation based on detected edges
4. LP extraction via the count of connected components
5. Character segmentation and features calculation
6. Optical character recognition
7. Vehicle authorization and barrier control system

These steps are further explained in the following subsections.

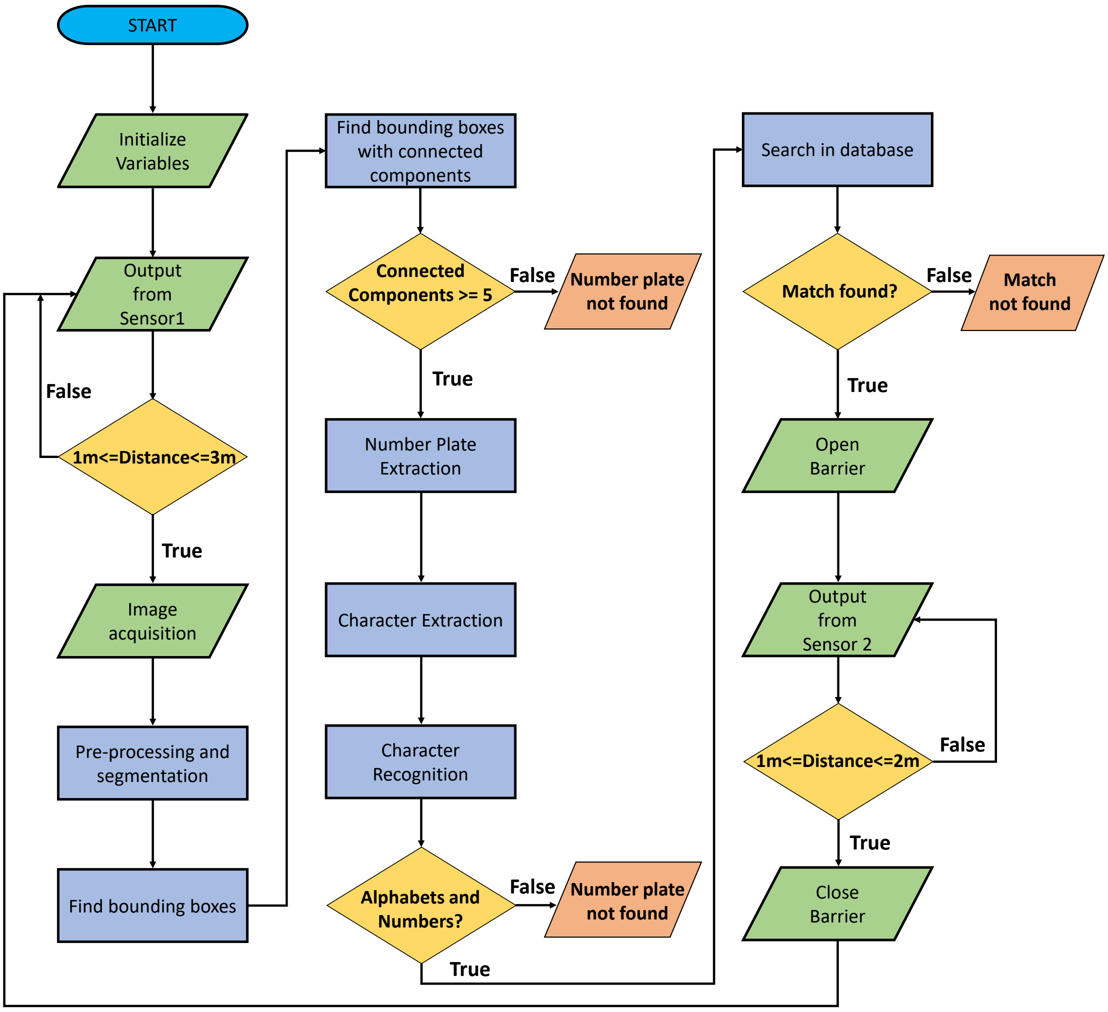


Figure 3. Algorithm flow chart of the proposed smart access control for vehicle entrance using sensors platform and an image-based LP recognition.

3.1. Vehicle Arrival Detection and Image Acquisition

Figure 4 shows the proposed hardware architecture for barrier access entrance control. Ultrasonic sensors installed at the barrier detect the approaching vehicle. The sensor emits 8-pulses of 40 KHz for 10 µs and listens to the echo signal for 100 µs to 36 ms. Using $S = \frac{Vt}{2}$, we find the distance between barrier and the vehicle where S is the distance, V is the speed of sound: .034 m/µs and t is the time in µs for transmission and its echo signal. The camera is only activated for image acquisition when the ultrasonic sensors detect the vehicle in a specific range of distance which is set from 1 to 3 m. As a common practice on gate entrances, a lane is built for the entering vehicle so that they are almost straight when the image is taken by the camera. Due to this reason, the image of the entering vehicle is taken with negligible rotations.

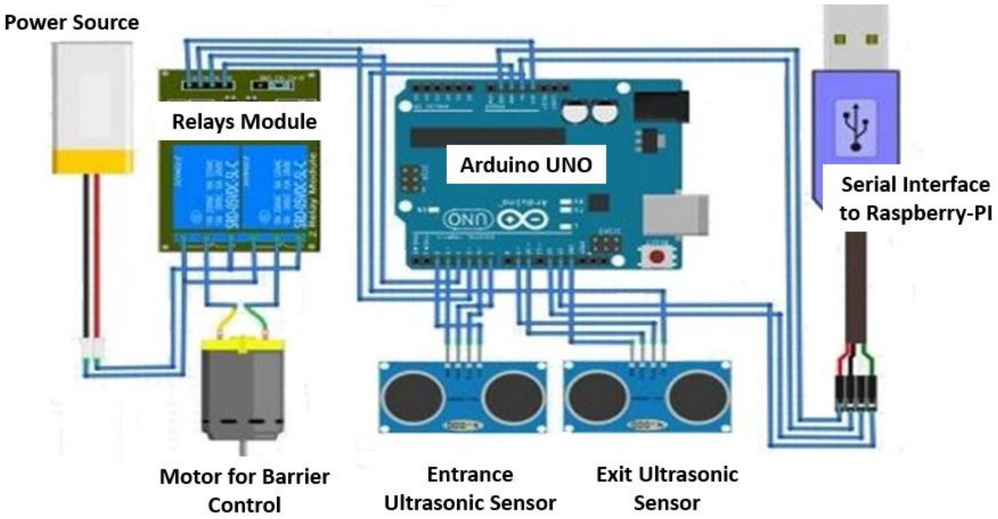

Power Source
Relays Module
Arduino UNO
Serial Interface to Raspberry-PI
Motor for Barrier Control
Entrance Ultrasonic Sensor
Exit Ultrasonic Sensor

Figure 4. Hardware architecture of an ultrasonic sensors-based vehicle entrance and exit detection.

3.2. Image Pre-Processing and Edge Image Generation

In the proposed ALPR method, we convert the captured image into grayscale. It reduces the processing complexity and processing time and is robust to color changes due to different lighting conditions. A canny edge detector is applied to this image to detect all the edges. The Canny edge detector is a combination of a Gaussian filter for smoothing and a Sobel filter for edge detection. Equation (1) shows the Gaussian filter that suppresses the noise in an image with σ as the standard deviation of the Gaussian filter.

$$\mathbf{G}(x,y) = \frac{1}{2\pi\sigma^2} e^{(}-\frac{x^2+y^2}{2\sigma^2}) \tag{1}$$

$$\mathbf{S_x} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad \mathbf{S_y} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2}$$

$$|S| = \sqrt{\mathbf{S_x}^2 + \mathbf{S_y}^2} \tag{3}$$

$$\angle S = \tan^{-1}(\frac{|\mathbf{S_y}|}{|\mathbf{S_x}|}) \tag{4}$$

After the Gaussian filter, we apply the Sobel masks [37] to detect the horizontal and vertical edges as shown by Equation (2). Equations (3) and (4) show the magnitude and direction of the Sobel gradient respectively. Considering the pixel magnitude, direction, non-maximum suppression and thresholds, the pixel is marked as an edge if its magnitude is greater than the threshold in the gradient direction. Finally, at this stage, we get an edge segmented image.

3.3. Image Segmentation Based on Detected Edges via Morphological Operations

On the generated edge image, we perform various morphological operations such as dilation, horizontal erosion, vertical erosion and hole filling. Dilation adds the pixels to the boundary of edges to complete the boundary and increases the efficiency of LP extraction. Mathematically, Equation (5) shows the dilation.

$$\mathbf{I_{dilated}} = \mathbf{I} \oplus \mathbf{B} = \{z|(\hat{\mathbf{B}})_{\mathbf{z}} \cap \mathbf{I} \neq \emptyset\} \tag{5}$$

where **I** is edge segmented image and **B** is structure element. After dilation, we filled the closed boundaries and remove unnecessary parts of the image without affecting the LP area. A hole filling technique is used for this purpose and its mathematical expression is given by Equation (6).

$$\mathbf{I_{holefilled}} = \mathbf{X_k} = (\mathbf{X_{(k-1)}} \oplus \mathbf{B}) \cap \mathbf{I^c_{dilated}} \tag{6}$$

We use vertical and horizontal erosion to remove those pixels, which makes it difficult to extract the LP. All the unnecessary lines and parts connected to the LP area create problems for the LP extraction. Equation (7) shows the mathematical expression used for erosion.

$$\mathbf{I_{eroded}} = \mathbf{I_{holefilled}} \ominus \mathbf{B} = \{z | (\mathbf{B})_\mathbf{z} \in \mathbf{I_{eroded}}\} \tag{7}$$

3.4. LP Extraction via the Count of Connected Components

We find the 8-connectivity components based rectangular bounding box objects in the eroded image. In addition to the LP, there are other rectangular objects such as headlights, radiators, grille and bumper. Therefore, it is likely that these objects are also segmented along with the LP. Due to this reason, we use the count of connected components in each segment as a clue to differentiate between the LP and other rectangular objects. To this end, for a segment to be considered an LP, the number of objects inside that segment should be more than five. This is due to the fact that the Pakistani LP consists of at least five characters as shown in Figure 3. Once the mask of the LP is generated in this way, it is used to extract the LP from the RGB image.

3.5. Characters Segmentation from LP Segment and Features Calculation

Once the LP region is extracted, character segmentation is employed to extract the LP characters. For this purpose, as a first step, the LP region is binarized using the algorithm shown in Figure 5. First, we calculate the intensity histogram of the LP region image and then find the two highest peaks in this histogram. We considered the two highest peak because the LP mostly consists of two colors, that is, the LP background color and the characters' color. The threshold is the average value of these two peaks.An LP grayscale image is then binarized using this threshold. We extract characters from the binary image using 8-connectivity, considering a character height of 30 to 90, a width of 10 to 40 and an area of 700 to 800 pixels.

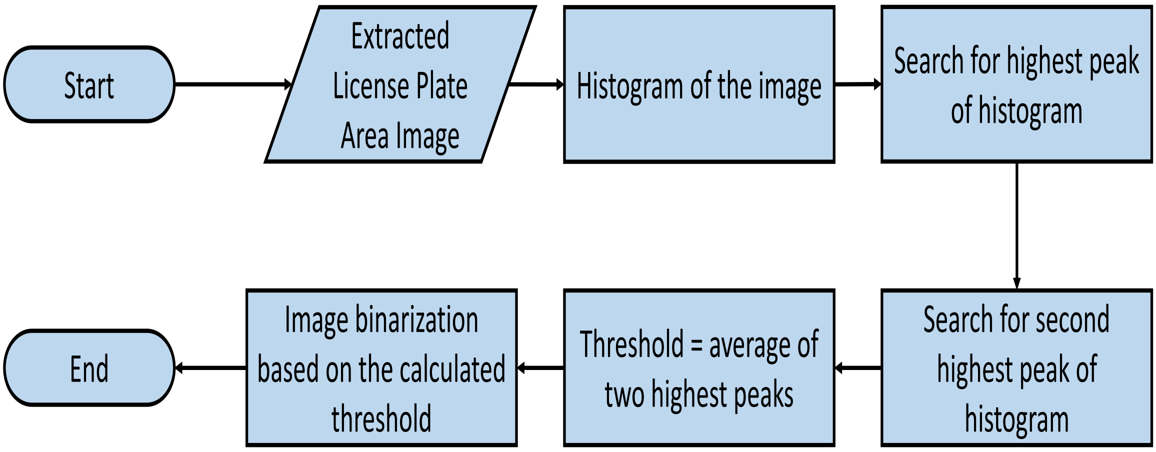
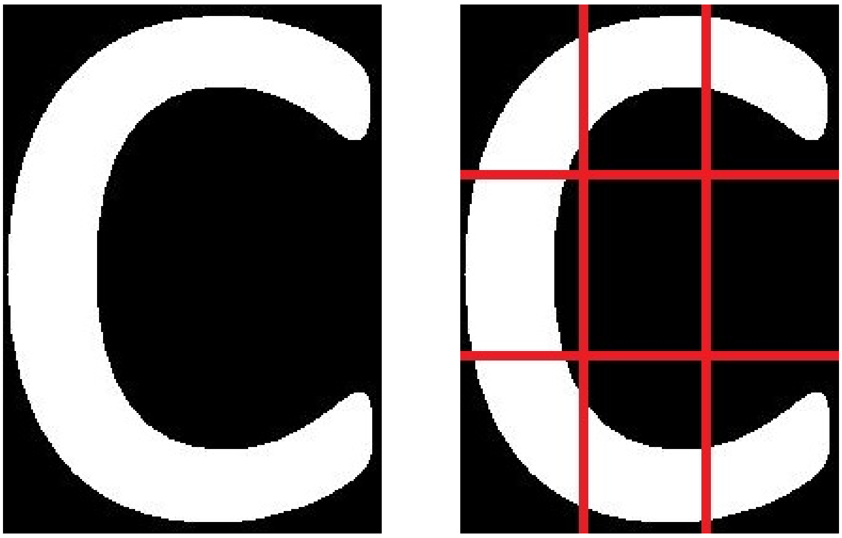

Start
Extracted License Plate Area Image
Histogram of the image
Search for highest peak of histogram
Search for second highest peak of histogram
Threshold = average of two highest peaks
Image binarization based on the calculated threshold
End

Figure 5. Thresholding algorithm to convert the LP grayscale image to binary.

In this paper, we focus on the features-based approach for character recognition. We extract the following features of a character.

- **Zoning**: It divides the character image into various sub-images. Figure 6 shows the overview of zoning a character image into 3 × 3 sub-images. The white pixels are summed in each sub-image and become a feature.



(**a**) Binarized image of a character (**b**) 3 × 3 zoning

Figure 6. Overview of zoning of an image into sub-images.

Mathematical it can be calculated by Equation (8).

$$zoning_{feature} = \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbf{subimage}(i,j) \tag{8}$$

where $M \times N$ is the size of sub-image. We considered a character 42 × 24 size of image and then divided it into nine sub-images of 14 × 8 each.

- **Perimeter:** The set of interior boundary pixels of a connected component (character image (**C**)) [33]. We considered 8-connectivity to find the perimeter. Equation (9) finds the perimeter of a character.

$$Permiter_8 = \{(a,b) \in \mathbf{C} | N_4(a,b) - \mathbf{C} \neq \emptyset\} \tag{9}$$

where (a,b) is the pixel location.

- **Extent:** It is the ratio of white pixels in an image to the total number of pixel in the binary image. Equation (10) finds the Extent value of a character image.

$$Extent = \frac{Number\_of\_White pixels}{Total\_Pixels\_Image} \tag{10}$$

- **Euler Number:** Euler number is the topology measure of an image. It is the number of objects in an image minus the number of holes in the image. Equation (11) finds the Euler number:

$$Euler\_number = 1 - number\_of\_holes \tag{11}$$

- **Particular Rows and Columns Pixels Summation:** In the paper, we consider some particular rows and columns to add their pixels. That particular row or column pixels summation is considered as a feature. Equation (12) finds the sum of a particular row i.

$$sum\_row\_i = \sum_{col=1}^{TotalCol} \mathbf{C}(i,col) \tag{12}$$

where $TotalCol$ shows the number of columns in the character image **C**. We considered the summation of rows third, fifteen, twenty-seven and thirty-seven as features. The summation of the column is given by Equation (13).

$$sum\_column\_j = \sum_{r=1}^{R} \mathbf{C}(r, j) \tag{13}$$

where R is rows in character image **C** and we find the summation of second, twelve and seventeen columns.

- **Eccentricity:** Finds how close an object is to being a circle. It is ratio of the linear eccentricity to the semi-major axis.
- **Orientation:** The major axis of an ellipse around the object and then finding the angle which the major axis made with the x-axis.

3.6. Optical Character Recognition of LP Characters

In this paper, we evaluated various supervised learning algorithms (classifiers) to recognize characters on the LP. Figure 7 shows the process of LP character recognition. As a first step, these characters are manually extracted from the images. The aforementioned features of each extracted character are calculated in order to represent each of them in a single feature vector of length 21. A given classifier is then trained on these features. For testing, the proposed extraction algorithm first extracts the LP characters automatically while the trained classifier recognizes the characters by predicting their labels. We used KNN, Decision Trees, Random Forest, SVM, and ANN for LP character recognition.

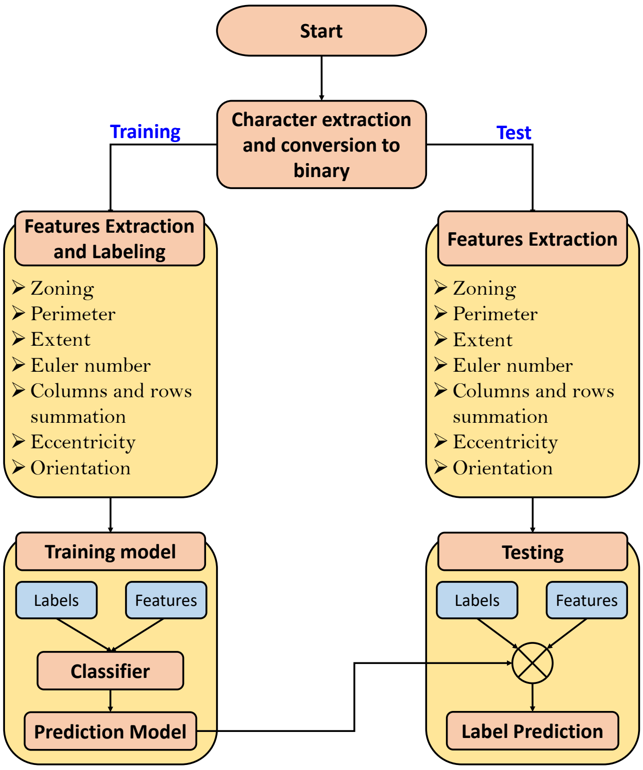
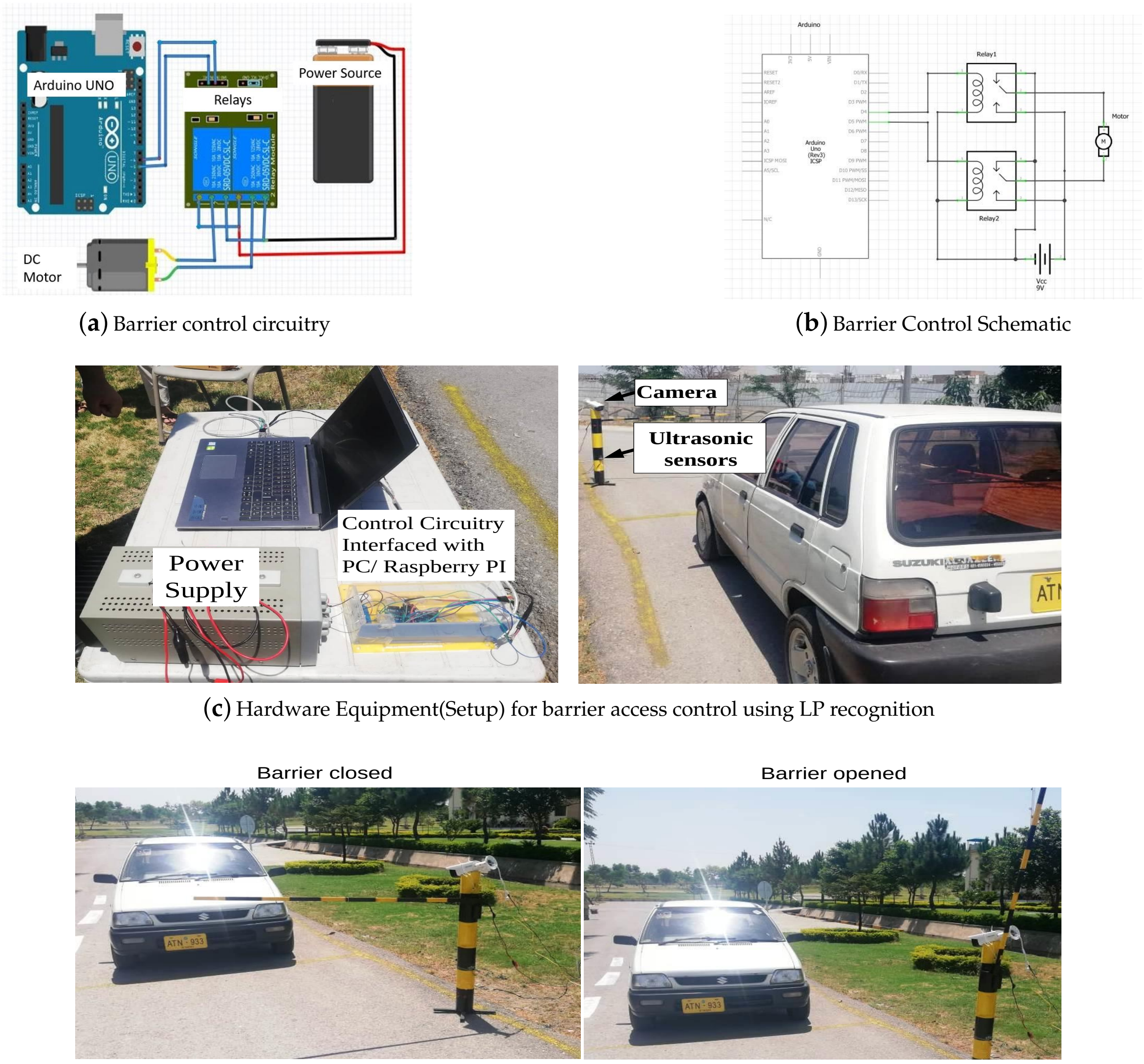


Figure 7. An overview of LP characters recognition process.

3.7. Vehicle Authorization and Barrier Control System

The real-time system for vehicle detection and authorization is implemented on a Raspberry Pi. The ultrasonic sensors interfaced to the Raspberry Pi detect the approaching vehicle on entrance. The

LP of this vehicle is then verified using its image. If the vehicle is permitted then the Raspberry Pi sends a command to open the barrier. Figure 8 shows the circuit, schematic, hardware setup and access mechanism of the barrier control system. Figure 8c shows the real-time hardware setup used to detect the vehicle, recognize the LP and control the barrier position. A camera and two ultrasonic sensors installed on the barrier are also shown. The front ultrasonic sensor detects the vehicle at the entrance and the rear ultrasonic sensor detects the exited vehicle. The barrier control circuitry is interfaced using the RS-232 serial port to the LP processing system. We used a DC motor [38] to control the barrier access that rotates between 0° and 360°. We used 90° and +180° for closed and open barrier systems respectively, as shown in Figure 8d. The motor rotates the barrier bar from open to closing when the first relay is active and second is de-active and vice-versa. In real implementation, we used the DC motor rated 24 V of high torque which can easily move a barrier bar that weighs upto 8 kg. However, in a PC based simulation, we used 9 V to simulate the controlling of the motor.



(**a**) Barrier control circuitry

(**b**) Barrier Control Schematic

(**c**) Hardware Equipment(Setup) for barrier access control using LP recognition

(**d**) Barrier positions

Figure 8. Hardware Setup and mechanism of barrier system control using DC motor.

4. Results & Discussion

The proposed system is implemented on the following frameworks.

1. PC(Intel(R), Core(TM) i3-4010U CPU 1.70GHz,RAM: 4.00GB) running Matlab($R2013a$, 64-bits) and interfaced the Arduino using an RS232 serial port for the barrier control system. Matlab programming and Arduino C-based code are used to implement the system
2. Raspberry Pi- system on chip single board computer with 1.4 GHz 64-bits quad-core processor and interfaced the Arduino using RS232 serial port for the barrier control system. Python 3, OpenCV 3.4.0, and Arduino C-based code is used to implement the system.

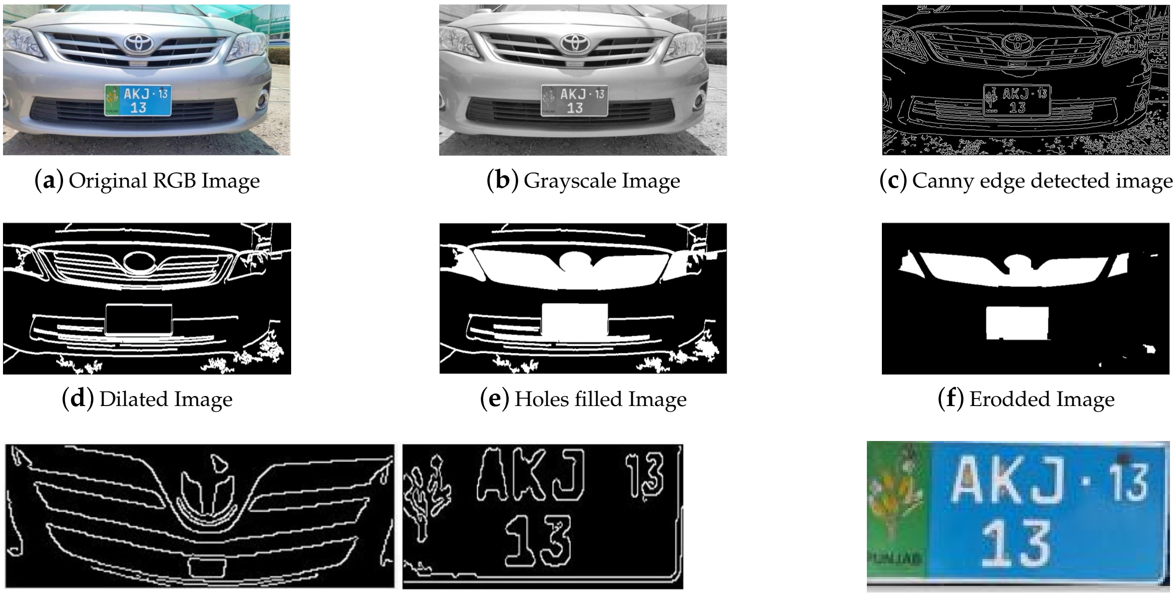
Table 2 shows the details of the acquired dataset used as training and test images. Images were taken with a camera in daylight conditions.

Table 2. Dataset description for LP extraction, characters segmentation and recognition.

Sequence No.	Description	No. of Images
1	Images for LP extraction	500
2	Images for characters segmentation	500
3	Characters (0–9) & (A–Z)	3643
4	Features vector dimension	21

4.1. Results of Pre-Processing, Edge Detection, and LP Area of Interest Extraction

Figure 9 shows the qualitative results of pre-processing before LP extraction. Figure 9a shows the original RGB captured image and the resized image when detected by the ultrasonic sensor in the specified range. Figure 9b shows the RGB image converted to grayscale and Figure 9c shows the detected edges in the image via Canny edge detector. Figure 9d–f shows various morphological operations applied to the edge image. A 5 × 1 structure element of Dilation enlarges the edges. Hole filling fills the connected objects and erosion removes the pixels on object boundaries and the single pixel objects (lines). Connected components based segments are extracted using a constraint of connected objects on a segment as shown in Figure 9g. Finally, Figure 9h shows the extracted LP area of interest.



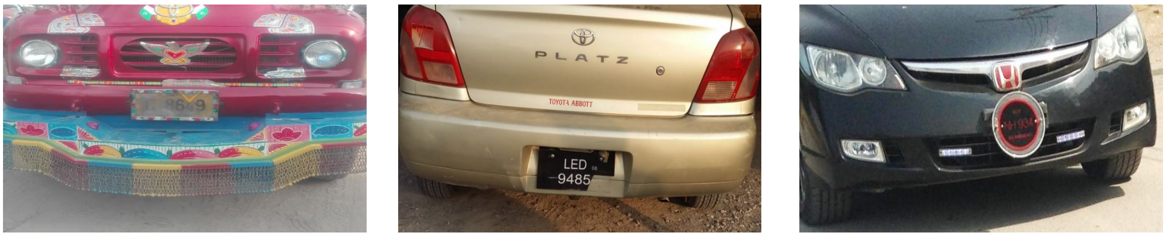
(**a**) Original RGB Image (**b**) Grayscale Image (**c**) Canny edge detected image

(**d**) Dilated Image (**e**) Holes filled Image (**f**) Erodded Image

(**g**) Connected Components (Bounding Box) Extracted Segments (**h**) Extracted LP area

Figure 9. Results of pre-processing, edge detection, morphological operations, connected components extraction and final LP area extraction.

For 500 images, the LP extraction accuracy of the proposed method is 98%. Figure 10 shows images where the LP is not correctly extracted due to various reasons such as character occlusion due to dirt, non-rectangular LP and broken LP.



(**a**) Severly occluded LP (**b**) Damaged LP (**c**) Non-rectangular LP

Figure 10. Vehicle images where LPs are not correctly segmented due to various reasons.

4.2. Results of LP Characters Segmentation

The step-by-step result of the LP segmentation and character recognition are visually shown in Figure 11. The variations in the LP background, font sizes and styles of the characters' positions can be observed in the different types of LPs. There are also additional numbers and characters in the LPs. The proposed method clearly shows its robustness to such challenges and extracts the bounding boxes that enclose only those characters that belong to the license plate's number. The LP character segmentation of the proposed method for an image dataset of 500 images is 96%.

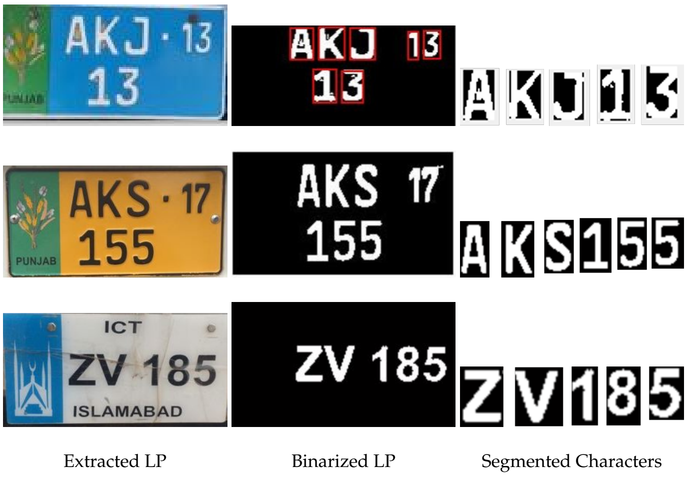


Figure 11. LP extraction, binarization and character segmentation.

4.3. Results of LP Characters Recognition

We assigned class labels 0–9 to the digits and 10–35 for alphabets A–Z to recognize the LP characters. A total of 3643 characters were extracted from images of computerized and handwritten LPs. The aforementioned features were calculated for each of the extracted characters. In the current setting, we evaluated various classifiers such as KNN, Naive Bayes, Bayes Network, SVM using linear kernel, MLP, Decision Tree and RF for LP character recognition. The classifiers training and testing was done using 10-fold cross validation where a given dataset is split into 90% training set and 10% test set. The performances of classifiers were compared with respect to their classification accuracy, true positive rates, false positive rates, precision, recall, F-measure and ROC area. Figure 12 shows the classification accuracies of all the classifiers on the given dataset. Table 3 shows the detailed comparison of classifiers with respect to other metrics.

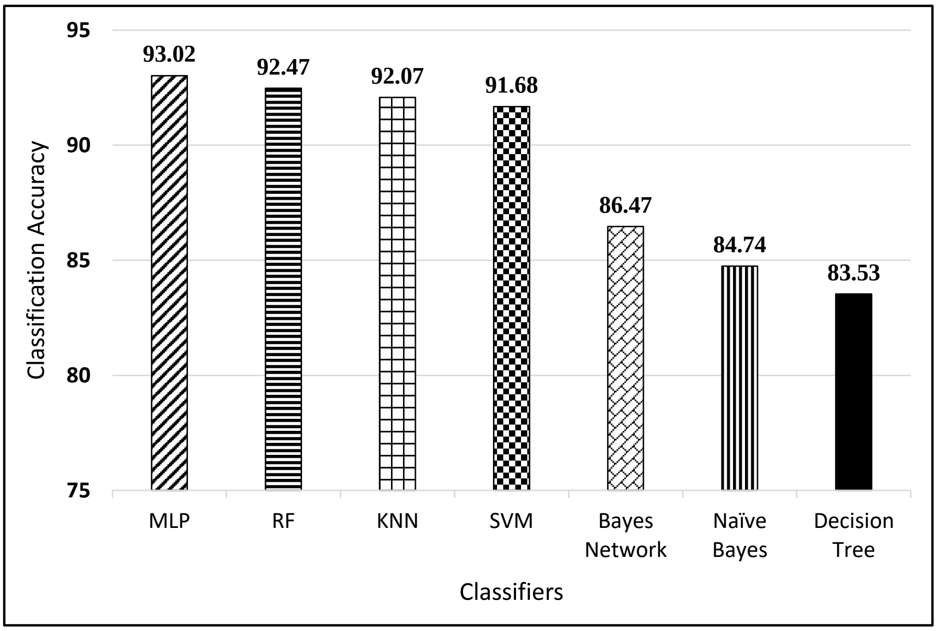


Figure 12. Classifiers accuracy performance comparison for OCR.

Table 3. Comparison of performance metrics (*TPR*, *FPR*, *Precision*, *Recall*, *F-measure*, and *ROC Area*) of classification algorithms for OCR.

$\frac{\mathit{Classifiers}}{\mathit{Parameters}}$	TPR	FPR	Precision	Recall	F-measure	ROC Area
KNN	0.921	0.003	0.921	0.921	0.92	0.959
Bayes Network	0.865	0.005	0.868	0.865	0.865	0.994
Naive Bayes	0.847	0.006	0.859	0.847	0.85	0.99
SVM	0.917	0.004	0.913	0.929	0.916	0.956
MLP	**0.93**	**0.002**	**0.931**	**0.93**	**0.93**	**0.995**
Decision Tree	0.835	0.007	0.832	0.835	0.833	0.93
RF	0.93	0.004	0.93	0.925	0.926	0.965

Tables 4 and 5 show the confusion matrix of OCR using the KNN and MLP algorithms, respectively. The characters on the LPs are handwritten and computerized. Mostly, the handwritten characters such as 0 have higher similarity with *O* and also *Q* with 0. 5 and *S*, and *M* and *N* have higher similarity. If they are computerized based *O*, *Q*w and 0 are used it will increase the recognition accuracy.

The accuracy of MLP, KNN, SVM and RF are close to each other. Therefore, we implemented the KNN algorithm for real-time ALPR both on Matlab and Raspberry Pi based proposed systems. Table 6 shows the time analysis of the proposed system, both implemented using a PC with Matlab and Raspberry Pi with Python and OpenCV library. It depicts the time from vehicle detection to LP number recognition and barrier access control opening. The time consuming part of the proposed system is the pre-processing-LP localization and recognition of LP characters. We compared the timing for 100-LPs that had 5-characters, 6-characters, and 7-characters, respectively. The Raspberry Pi based system had the lowest computation time, a small size and low power requirements. It can be easily installed in the constraint area for vehicle detection and control access to a restricted area.

Table 4. Confusion matrix of OCR using KNN classifier algorithm.

Actual Class *Predicted Class*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	148	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
1	0	156	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	1	223	3	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	3
3	0	1	3	195	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	212	0	1	0	0	1	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	2	1	0	198	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
6	0	0	0	0	1	0	213	0	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	213	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
8	0	0	0	0	0	1	1	0	213	1	1	13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	2	224	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	3	0	1	0	0	0	232	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	1	1	0	23	0	0	74	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	1	0	0	0	0	0	68	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
D	9	0	0	0	0	0	0	0	1	0	1	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	1	0	0	2	1	0	0	1	0	0	1	0	167	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
F	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	48	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
G	2	0	0	0	0	0	2	0	0	0	0	1	1	0	1	0	21	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	30	0	0	0	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0
I	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
J	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	2	0	0	84	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0
L	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	185	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	23	8	0	0	1	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0	0	1	0	1	55	0	0	0	0	0	0	0	0	7	0	0	0
O	5	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	7	0	1	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
Q	5	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	2	0	3	1	1	1	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0
S	0	0	3	1	0	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	24	0	0	1	0
W	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	1	0	0	0	0	23	0	0	0
X	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	15	0
Z	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	42

Table 5. Confusion matrix of OCR using MLP classifier algorithm.

Actual Class / *Predicted Class*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	145	0	0	0	0	0	1	0	1	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0
1	0	154	1	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	209	1	0	0	0	4	0	2	0	2	0	0	1	0	1	0	0	2	0	0	2	1	0	0	0	0	4	0	0	0	0	2	0	5
3	0	0	0	200	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
4	0	0	0	0	211	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0
5	0	1	0	1	0	199	0	0	1	0	0	2	0	1	0	0	0	0	2	0	0	0	0	2	0	0	0	0	3	0	0	0	0	0	0	0
6	0	0	0	0	1	0	210	0	1	0	0	1	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	2	1	0	0	0	0	0	0	0
7	0	2	0	0	0	0	0	213	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	2	0	0	218	1	0	7	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	2	0	0	0	0	0	0	0	0
9	0	0	0	1	0	0	0	0	1	221	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0
A	0	0	0	0	5	0	2	0	2	0	225	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	1	0	0	13	0	0	81	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	69	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	2	0	0	0	0	0	0	0	0	0	0	2	0	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	172	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	2	1	0	0	0	0	0	0	0	0	1	0	0	0
I	0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	2
J	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	84	0	0	0	0	0	0	4	0	0	0	0	0	1	0	1
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	188	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	5	0	0	0	0	0	0	0	0	0	1	0	0
N	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	60	0	0	0	0	0	0	1	0	2	0	0	0
O	8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0 1	9	0	1	0	0	0	0	0	0	0	0
Q	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	6	0	0	0	1	0	0	0	0	0
R	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	3	0	1	0	0	0	0 4	9	0	0	0	1	0	0	0	0
S	0	1	1	3	0	3	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0 3	7	0	0	0	0	1	0	0
T	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 3	1	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0 2	8	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0 2	3	0	0	2	0
W	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	1	0	0	0	0	0	1	0	0 3	1	0	0	0
X	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 4	1	0	1
Y	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 1	5	0
Z	0	1	3	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0 3	9

Table 6. Performance comparison of time (Seconds) taken by the proposed system implemented on a PC (running Matlab) and a Raspberry Pi (Python + OpenCV).

	PC[Matlab]			**Raspberry Pi (Python)**		
Characters on LP *Processing on LP*	**5 Characters**	**6 Characters**	**7 Characters**	**5 Characters**	**6 Characters**	**7 Characters**
Vehicle Detection and LP Extraction	3.2	3.19	3.21	0.13	0.11	0.14
LP Characters Segmentation	2.1	2.14	2.27	0.1	0.12	0.13
Characters Recognition and Barrier Control	3.22	3.26	3.37	0.16	0.17	0.21
Total Time Taken (Seconds)	8.52	8.59	8.85	0.39	0.4	0.48

5. Conclusions

We presented a robust, accurate, industrial barrier access control system using a sensor platform and vehicle license plate recognition. The proposed system automatically detects a vehicle at an entrance via ultrasonic sensors and then recognizes it by image-based recognition of its license plate, which can have various backgrounds, fonts and font styles. To this end, a performance evaluation of various classifiers was carried out to find out that which had the best recognition rate. Lastly, the proposed system was implemented both on a PC running Matlab and on a Raspberry Pi (system on chip) running Python with OpenCV. The Raspberry Pi-based system had low computational time, a smaller size, and low power consumption, due to which it was used in the real-time application. In future, we are working to increase the dataset of handwritten LP characters to improve accuracy and laser beam-based vehicle detection to increase the detection range.

Author Contributions: All authors contributed to the paper. F.U.: Conceptualization, Software, Hardware, Formal analysis, Writing; H.A.: Data curation, Software, Validation; I.S.: Hardware, Software, Writing; A.U.R.: Hardware, Writing; S.M.: Data curation, Software; S.N.: Data curation, Software; K.M.A.: Validation; A.K.: Data curation, Validation; D.K.: Conceptualization, Funding acquisition, Review and editing.

Funding: The research was funded by the Untenured Faculty Research Initiative (UFRI), Kean University and ICT Pakistan under the National Grassroot ICT Research Initiative (NGIRI).

Conflicts of Interest: The authors declare no conflict of interest.

Dataset Availabity: Dataset will be provided on request by the corresponding author.

References

1. Joshi, Y.; Gharate, P.; Ahire, C.; Alai, N.; Sonavane, S. Smart parking management system using RFID and OCR. In Proceedings of the 2015 International Conference on Energy Systems and Applications, Pune, India, 30 October–1 November 2015; pp. 729–734.
2. Zhou, H.; Li, Z. An intelligent parking management system based on RS485 and RFID. In Proceedings of the 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Chengdu, China, 13–15 October 2016; pp. 355–359.
3. Rashedi, E.; Nezamabadi-Pour, H. A hierarchical algorithm for vehicle license plate localization. *Multimed. Tools Appl.* **2018**, *77*, 2771–2790. [CrossRef]
4. Jin, L.; Xian, H.; Bie, J.; Sun, Y.; Hou, H.; Niu, Q. License plate recognition algorithm for passenger cars in Chinese residential areas. *Sensors* **2012**, *12*, 8355–8370. [CrossRef] [PubMed]
5. Gou, C.; Wang, K.; Yao, Y.; Li, Z. Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1096–1107. [CrossRef]
6. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698. [CrossRef]
7. Trier, Ø.D.; Jain, A.K.; Taxt, T. Feature extraction methods for character recognition-a survey. *Pattern Recognit.* **1996**, *29*, 641–662. [CrossRef]

8. Zhang, Z.; Shen, W.; Yao, C.; Bai, X. Symmetry-based text line detection in natural scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2558–2567.
9. Soora, N.R.; Deshpande, P.S. Review of Feature Extraction Techniques for Character Recognition. *IETE J. Res.* **2018**, *64*, 280–295. [CrossRef]
10. Liao, Y.; Zhang, J.; Wang, S.; Li, S.; Han, J. Study on Crash Injury Severity Prediction of Autonomous Vehicles for Different Emergency Decisions Based on Support Vector Machine Model. *Electronics* **2018**, *7*, 381. [CrossRef]
11. Liu, W.-C.; Lin, C. A hierarchical license plate recognition system using supervised K-means and Support Vector Machine. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; pp. 1622–1625.
12. Ullah, F.; Sarwar, G.; Lee, S. N-Screen Aware Multicriteria Hybrid Recommender System Using Weight Based Subspace Clustering. *Sci. World J.* **2014**, *2014*, 679849. [CrossRef]
13. Anagnostopoulos, C.N.E.; Anagnostopoulos, I.E.; Loumos, V.; Kayafas, E. A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 377–392. [CrossRef]
14. Pan, X.; Ye, X.; Zhang, S. A hybrid method for robust car plate character recognition. *Eng. Appl. Artif. Intell.* **2005**, *18*, 963–972. [CrossRef]
15. Liu, Y.; Huang, H.; Cao, J.; Huang, T. Convolutional neural networks-based intelligent recognition of Chinese license plates. *Soft Comput.* **2018**, *22*, 2403–2419. [CrossRef]
16. Tarabek, P. A real-time license plate localization method based on vertical edge analysis. In Proceedings of the 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, Poland, 9–12 September 2012; pp. 149–154.
17. Megalingam, R.K.; Krishna, P.; Pillai, V.A.; Hakkim, R.U. Extraction of license plate region in Automatic License Plate Recognition. In Proceedings of the 2010 2nd International Conference on Mechanical and Electrical Technology (ICMET), Singapore, 10–12 September 2010; pp. 496–501.
18. Bai, H.; Liu, C. A hybrid license plate extraction method based on edge statistics and morphology. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004; Volume 2, pp. 831–834.
19. Wang, J.; Bacic, B.; Yan, W.Q. An effective method for plate number recognition. *Multimed. Tools Appl.* **2018**, *77*, 1679–1692. [CrossRef]
20. Zheng, D.; Zhao, Y.; Wang, J. An efficient method of license plate location. *Pattern Recognit. Lett.* **2005**, *26*, 2431–2438. [CrossRef]
21. Dun, J.; Zhang, S.; Ye, X.; Zhang, Y. Chinese license plate localization in multi-lane with complex background based on concomitant colors. *IEEE Intell. Transp. Syst. Mag.* **2015**, *7*, 51–61. [CrossRef]
22. Safaei, A.; Tang, H.L.; Sanei, S. Robust search-free car number plate localization incorporating hierarchical saliency. *J. Comput. Sci. Syst. Biol.* **2016**, *9*, 93–103.
23. Tabrizi, S.S.; Cavus, N. A hybrid KNN-SVM model for Iranian license plate recognition. *Procedia Comput. Sci.* **2016**, *102*, 588–594. [CrossRef]
24. Gazcón, N.F.; Chesñevar, C.I.; Castro, S.M. Automatic vehicle identification for Argentinean license plates using intelligent template matching. *Pattern Recognit. Lett.* **2012**, *33*, 1066–1074. [CrossRef]
25. Zhang, Y.; Wang, Y.; Zhou, G.; Jin, J.; Wang, B.; Wang, X.; Cichocki, A. Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces. *Expert Syst. Appl.* **2018**, *96*, 302–310. [CrossRef]
26. Tarigan, J.; Diedan, R.; Suryana, Y. Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm. *Procedia Comput. Sci.* **2017**, *116*, 365–372. [CrossRef]
27. Laroca, R.; Severo, E.; Zanlorensi, L.A.; Oliveira, L.S.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. A Robust Real-Time Automatic License Plate Recognition based on the YOLO Detector. *arXiv* **2008**, arXiv:1802.09567.
28. Zheng, L.; He, X.; Samali, B.; Yang, L.T. An algorithm for accuracy enhancement of license plate recognition. *J. Comput. Syst. Sci.* **2013**, *79*, 245–255. [CrossRef]
29. Chen, H.; Hu, B.; Yang, X.; Yu, M.; Chen, J. Chinese character recognition for LPR application. *Opt. Int. J. Light Electron Opt.* **2014**, *125*, 5295–5302. [CrossRef]
30. Massoud, M.A.; Sabee, M.; Gergais, M.; Bakhit, R. Automated new license plate recognition in Egypt. *Alex. Eng. J.* **2013**, *52*, 319–326. [CrossRef]

31. Smith, R. An overview of the Tesseract OCR engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition, Parana, Brazil, 23–26 September 2007; Volume 2, pp. 629–633.
32. Li, H.; Wang, P.; Shen, C. Towards end-to-end car license plates detection and recognition with deep neural networks. *arXiv* **2017**, arXiv:1709.08828.
33. Björklund, T.; Fiandrotti, A.; Annarumma, M.; Francini, G.; Magli, E. Automatic license plate recognition with convolutional neural networks trained on synthetic data. In Proceedings of the 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, UK, 16–18 October 2017; pp. 1–6.
34. Zhu, Y.; Huang, H.; Xu, Z.; He, Y.; Liu, S. Chinese-style plate recognition based on artificial neural network and statistics. *Procedia Eng.* **2011**, *15*, 3556–3561. [CrossRef]
35. Masood, S.Z.; Shu, G.; Dehghan, A.; Ortiz, E.G. License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks. *arXiv* **2017**, arXiv:1703.07330.
36. Zang, D.; Chai, Z.; Zhang, J.; Zhang, D.; Cheng, J. Vehicle license plate recognition using visual attention model and deep learning. *J. Electron. Imaging* **2015**, *24*, 033001. [CrossRef]
37. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*; Publishing House of Electronics Industry: Beijing, China, 2002; Volume 141.
38. DC Motor Details. Available online: https://www.alibaba.com/product-detail/90-watt-12-volt-24-volt_60170246153.html?spm=a2700.7724857.normalList.46.170f239bl7CAAN (accessed on 15 May 2019).