Saudi License Plate Recognition Algorithm Based on Support Vector Machine

Khaled Suwais, Rana Al-Otaibi, and Ali Alshahrani

Abstract—License plate recognition (LPR) is an image processing technology that is used to identify vehicles by their license plates. This paper presents a license plate recognition algorithm for Saudi car plates based on the support vector machine (SVM) algorithm. The new algorithm is efficient in recognizing the vehicles from the Arabic part of the plate. The performance of the system has been investigated and analyzed. The recognition accuracy of the algorithm is about 93.3%.

Index Terms—Image processing, license plates recognition systems, support vector machine.

1. Introduction

The principles of image processing are originally derived from different disciplines, sciences, and fields including mathematics, computer science, optics, visual psychophysics, and surface physics. Processing of digital images can be made by various methods, however, there are three main operations that are commonly used to completely comprehend the meaning of an image: filtration or restoration, enhancement, and understanding. In filtration/restoration and enhancement operations, the aim is to clarify an image’s meaning through working on a corrupt image and trying to identically recover the original image that could be easily recognized by humans. The license plate recognition (LPR) uses the combination of methods to catch the license plate number of any moving vehicle.

This system is composed of a camera to take an image, illumination for clarification, and a suite of algorithms. Algorithms and illuminations have the main function of giving a meaning for the image that can be identified and recognized by human visual mechanisms. This system is very helpful and is regarded as one of the most intelligent systems for transportation since it can be applied in many areas that require monitoring, such as toll collection and traffic. The importance and usefulness of LPR have increased due to the rapid growth of vehicles in every city around the world. For a big city like Riyadh, the implementation of LPR has become very significant in order to monitor the movement of vehicles through identifying their license plate numbers. This technology of image processing has four major steps. The first is acquiring an image of the license plate; the second is to extract the license plate; the third step is making segmentation of the plate; and the fourth is the reorganization of the license plate. Technologies of LPR are helpful applications in traffic systems which include traffic movements, parking lots, and access control.

Arab countries use license plates that hold the name of the country or an abbreviation indicating the country’s name in Arabic or English letters. In addition, the plate also contains both Arabic and English letters and numerals to indicate the vehicle’s license number. Here, our focus is given to the recognition of the Arabic letters and numerals contained in the new license plate used in Saudi Arabia (see Fig. 1). A new algorithm will be implemented so that captured images can be processed immediately.

2. Literature Review

Invention of automatic recognition of number plates took place in 1976 and it was used for the first time as a working application in the UK in 1969. This new technology attracted many researchers to investigate and develop their techniques and mechanisms for recognizing vehicle’s license plates through image processing. The four major steps that compose the LPR system are investigated and discussed through different methodologies that yielded various important conclusions.

Fig. 1. New Saudi license plate.
In the study conducted by El-Adawi\cite{1}, license plate (LP) extraction was carried out by converting the RGB (red, green, blue) image to a grayscale one. The data amount in the image was decreased and the unwanted data was avoided by using the corresponding algorithm. The area that shared the similar properties with the license plate was extracted. Spatial filtering was employed to make the image’s edges detectable: horizontal and vertical directions were detected by using the convolution kernel. The following stage was to conduct the dilation process through the ripples that were found on a line and would make the line seem as broken. Lastly, the small lines that did not appear as part of the plate’s boarder in vertical and horizontal directions were rejected. The license plate was then extracted by using watershed transformation. For the digits in each row, the horizontal summation of black pixels was calculated in the segmentation stage. In addition, black pixels found in every column were vertically counted. In the final stage, which was concerned with recognition, a neural network composed of three layers was used. The final results showed that the ratio of recognition reached 80%.

In another study on Iranian plates conducted by A. Broumandnia and M. Fathy\cite{2}, a neural network based black-propagation algorithm was employed in the extraction stage to locate the license plate. Consequently, a noise-removal process using a non-linear and adaptive filter was applied. There were two major processes that constitute the extraction of characters found on the license plate. The first was the automatic local threshold selection and the second was the rectangular separation of characters. The recognition stage was claimed to be done with high performance if the two steps of city names and discrete letters recognition were followed. The use of these two steps was evaluated when applied on Iranian license plates which were classified to five types. The rates of recognition for the first type showed 98%, the second type 93%, the third 96%, the fourth 94%, and the fifth 93%.

In the segmentation stage, Wisamet et al.\cite{3} used vertical scanning to identify the first and last column of every alphabetical and numeric characters in the license plate. For the recognition stage, the researchers proposed two methodologies for the standard and non-standard plates. A fuzzy system was used for the standard plates and the template matching theorem was used for the non-standard one. The accuracy ratio of recognition was 95.5%.

In a research conducted by Alginahi\cite{4}, digital cameras were used to capture license plate images in the outdoor environment. Then, a grey-level image was created by averaging the full intensity of the three main colors. Afterwards, the technique of eight connected components was employed in the character segmentation stage so that all numerals and characters found on the image could be detected. Zoning and the profiles of horizontal projection were used as features for each character to be extracted. Lastly, a label was assigned to each character in the recognition step. And the identity to each label assigned is given according to the information obtained from the character’s extracted features.

Khalili\cite{5} has developed a recognition algorithm for the Saudi license plates. The researcher depends on the technique of template matching to help recognize not only the Saudi license plates, but also the Egyptian ones. This recognition technique is used for the old format of Saudi and Egyptian license plates. The only difference between the Khalili’s recognition technique and other recognition systems is the use of a moving window template matching classifier.

In the next section we present our LPR algorithm to handle the Saudi license plates as a particular case study in this research.

3. LPR Algorithm Based on SVM

The proposed LPR algorithm is composed by several stages, including image acquisition, license plate extraction, horizontal and vertical projection, segmentation and recognition of characters and numbers as shown in Fig. 2.

3.1 Image Acquisition

In this system, a high quality camera is used to capture a picture of the front car plate then send the captured image to the input/output (I/O) device.

3.2 License Plate Extraction

The license plate extraction stage is one of the most important stages in the LPR system, on which the next stages largely depend. At this stage the captured RGB color image is converted into a grayscale image. Then, the vertical edge detection (Soble) is performed on the binary image. After completing these steps, the MATA LB function “strel” is used. SE=strel(‘rectangle’, MN) creates a rectangle shaped structuring element, where MN specifies the size. MN must be a two-element vector of non-negative integers. The first element of MN is the number of rows of the structuring element, and the second element is the number of columns.

![Plate recognition system components.](image)

Fig. 2. Plate recognition system components.
After that, we perform the binary erosion based on the structuring element object returned by the “strel” function. After that, we remove noise in the picture (e.g. dust or noise resulting from the transfer of the image), which has significant impacts on the upcoming phases, such as detection and extraction of the car license plate, as well as the character segmentation. The morphological process is then used to search the entire image for small connected elements and remove them. Next, the morphological dilation process (filling gaps and holes in image) is used to connect different parts of a license plate. Finally, the analysis of components procedure will be performed for each connected component to locate car license plate then extract it. Each connected object is given a ratio then the one with the highest ratio, will be extracted as the license plate as shown in Fig. 3. Fig. 4 presents the license plate extraction procedure interface. Fig. 5 illustrates the license plate extraction stage.

### 3.3 Horizontal and Vertical Projection

Horizontal and vertical projection is the step where the image is segmented into many parts. In this stage the extracted grayscale car plate image is segmented into four parts: Arabic letters, Arabic numbers, English letters, and English numbers. The proposed projection process is based on the size of the car plate. In the vertical projection, the number of columns in the image is counted and then the result is divided by two, resulting in two separate images. Consequently, the horizontal projection is performed on the two separate images. The idea of this projection is to count the number of columns/rows then divide the result by two. At the end of this stage we get four separate images. The projection stage is presented in Fig. 6.

### 3.4 Characters and Numbers Segmentation

In the segmentation step, the plate’s elements (characters and numbers) are extracted from the plate’s background. The next stage is to make further segmentation by segmenting each group of Arabic letters and numbers. In this stage, we found that it is proper to extract connected objects in the plate. First, we convert the grayscale (Arabic letters and Arabic numbers parts) image into a binary image by computing a threshold level by using the Otsu’s method. Then, the black connected objects (letters, numbers, dots, and lines) are labeled using the “bwlabel” function. This function returns a matrix L of the same size as the input binary image, containing the labels for the connected objects. Finally, by using the “find” function, the labeled object will be extracted. The function returns vectors of indices for the pixels that make up a specific object by giving the row number (“row” in Fig. 7) and the column number (“column” in Fig. 7). The segmentation phase is presented in Fig. 7.

---

1. `image_g=rgb_to_grayscale(image)`
2. `Perform vertical_edge_detection(image_g)`
3. Create a rectangle shaped structuring element `rect`
4. `image_binary=binary_erosion(image_g, rect)`
5. Remove the small connected elements in the image `image_binary`
6. `image_plate=do_dilation(image_binary)`

---

1. `im_size=get_size(image_plate)`
2. `im_rows=get_rows(image_plate)`
3. `im_cols=get_columns(image_plate)`
4. `im_rows_new=im_rows/2`
5. `im_cols_new=im_cols/2`

---

1. Consider `image_p` as a projected image.
2. `image_p=to_binary(image_p, threshold)`
3. Label black connected objects, where each object is denoted by `O`.
4. `NumO=get_total(O)`
5. For `i=0` to `NumO`
   - 5.1 `[row, column]=find_pixels_coordinates(O_i)`
   - 5.2 `Ormax=max_row(S), OriMin= min_row(S),
       Ocmmax=max_col(S), OcmMin= min_col(S)`
   - 5.3 For `row=Ormin` to `Ormax`
     - 5.3.1 For `column=OcmMin` to `OcmMax`
     - 5.3.1.1 Extract the current object.
6. End for loop.
3.5 Characters and Numbers Recognition

Recognition is the final step in the LPR system where the output from the segmentation stage is being understood by the system. In this stage we use the support vector machine (SVM) as one of the supervised machine learning algorithms [6]. Supervised learning is a machine learning technique for deducing a function from training data. In this type of learning there is a label associated with each example. It is supposed to be the answer to a question about the example. If the label is discrete, then the task is called a classification problem, otherwise a regression problem. In order to construct the proposed SVM-based model we should define the data training set to train the SVM algorithm and test the SVM model.

In the first step we define the data training set by dealing with Arabic characters and numbers that are used in the new Saudi license plate as shown in Fig. 8. And Fig. 9 illustrates the excluded characters.

The second step is to train the SVM algorithm since there are many types of classifiers can be used in the SVM algorithm especially for our situation that we are concerned in the optical character recognition (OCR) classification problem. We used a multi-class classifier recommended in the SVM user guide. The training algorithm for characters is presented in Fig. 10 and for numbers is presented in Fig. 11.

Testing the SVM model is the final step, which is obtained in the previous step. The overall algorithm for testing SVM is presented in Fig. 12.

Fig. 8. Used Arabic characters.

Fig. 9. Excluded Arabic characters.

Fig. 10. Training SVM algorithm for Arabic characters.

Fig. 11. Training SVM algorithm for Arabic numbers.

1. Load characters training set.
2. Set the SVM solver to MDM.
3. Set the SVM kernel to RBF.
4. Set kernel argument to 5.
5. Set regularization constant to 10.
6. Start training the SVM model.
7. Store the SVM model.

1. Load numbers training set.
2. Set the SVM solver to MDM.
3. Set the SVM kernel to RBF.
4. Set kernel argument to 5.
5. Set regularization constant to 10.
6. Start training the SVM model.
7. Store the SVM model.

1. Load SVM model.
2. Get the segmented character or number.
3. Start testing SVM.
4. If the character or number is recognized:
   1. Return its label, else
   4.2 Return a label of unrecognized character or number.
5. Match label with its character or number.

Fig. 12. Testing SVM model.

Fig. 13. Performance of the algorithm main stages.

4. Results and Analysis

The experiment was coded in Matlab 7.5 on a workstation with an Intel Core 2 Duo CPU 2.00 GHz processor and 3 GB RAM. The 40 images which were processed in this experiment were captured by a Cannon EOS 5D MarkII camera. This section presents the performance analysis on the proposed Saudi license plate recognition system.

The testing revealed that the proposed algorithm could process and recognize license plates of car images in approximately 8.5 seconds. As the algorithm is composed of multiple stages, we found that the character and number recognition stage is the time-consuming stage due to the intensive computation that carried out in that stage. Fig. 13 shows the approximate time distribution for processing a given image over the algorithm stages. However, from the accuracy perspective, we found that our algorithm has an accuracy ratio of 93.3%, which is higher than the accuracy ratio of the algorithm in [7], which is only 80%.

5. Conclusions

In this paper we present the LPR algorithm for the new Saudi license plates. The algorithm is composed of four main stages, where the image is firstly captured, then the license plate is detected, and the characters are segmented and recognized in the third and fourth stages, respectively. The core of the algorithm is based on the SVM. In addition, the algorithm is efficient in term of the processing time and accuracy ratio. Each captured image could be processed in approximately 8.5 seconds and with an accuracy ratio of 93.3%.
Acknowledgment

The authors would like to thank Arab Open University (Saudi Arabia branch) for supporting this study.

References


Khaled Suwais received the B.Sc. degree from Al Al-Bayt University (AABU), Al-mafraq in 2004 and the M.Sc. and Ph.D. degrees from Universiti Sains Malaysia (USM), Penang, Malaysia in 2005 and 2009, respectively. He is currently an assistant professor with the Faculty of Computer Studies, Arab Open University (AOU), Riyadh, Saudi Arabia. His research interests include information security, image processing, and parallel computing.

Rana Al-Otaibi received the B.Sc. degree from Al-Imam Mohammad bin Saudi Islamic University, Riyadh, Saudi Arabia in 2010. In Sep. 2010, she worked as a teaching assistant with the College of Computer Sciences and Information, Al-Imam Mohammed bin Saud Islamic University. In Feb. 2011, she joined King Abdulaziz City for Science and Technology (KACST), Riyadh, as an academic researcher. Her research interests include information security, computer networks, and mobile application development.

Ali Alshahrani received the B.Sc. degree from AOU, Riyadh, Saudi Arabia in 2008 and the M.Sc. degree from University of Essex, Colchester, the UK in 2011. He is currently an IT lecturer with ITC Department, AOU, Riyadh, Saudi Arabia. His research interests include network security, image processing, and e-learning.

(The authors’ photographs are not available at the time of publication.)