Al-Mukhtar Journal of Sciences 37 (1): 01-12, 2022
ISSN: online 2617-2186 print 2617-2178
Journal Homepage https://omu.edu.ly/journals/index.php/mjsc/index

Libyan Vehicle License Plate Recognition with Support Vector Machine

Aeyman M. Hassan, Sami A. Ghoul, Aya A. Alkabir
School of Computer Engineering, University of Zawia, Zawia, Libya.

Received: 17 September 2021/ Accepted: 06 March 2022
Doi: https://doi.org/10.54172/mjsc.v37i1.525

Abstract: Computer vision has become widely used in many aspects of our daily lives. There are a great number of applications that consider computer vision as a core part of them, such as those associated with law enforcement. This paper presents the design and implementation of a model for a vehicle building entrance, using a license plate recognition algorithm for Libyan vehicles. In addition, a small dataset of Libyan vehicles was created for research purposes. As with most recognition systems, there are mainly three stages to be distinguished: plate detection using vertical and horizontal histograms, character segmentation is performed through a connected-component labeling algorithm, and finally, optical character recognition (OCR) by using support vector machines (SVMs). An Arduino board was used to control the gate opening and closing processes according to the authorized vehicles stored in the database. Ultrasonic sensors were used to detect a vehicle stop at the gate. The system was programmed with MATLAB executed on a 2.20GHz Core i7 CPU, 8 GB RAM, Windows 10. Despite the limited size of the vehicle images dataset, the experiments showed promising performance in terms of average accuracy estimated at 83.3%, and the computation time was 5 seconds.

Keywords: Support Vector Machine; Number Recognition; Arduino.

INTRODUCTION

Nowadays, biometric information has many forms such as signature, voice, iris scan, etc. This information is used throughout to identify people for various purposes. For instance, many people use this information to lock their mobile phones. However, on different occasions, there might be more interest in identifying a person via his vehicle. Therefore, it is crucial to design a system that can distinguish vehicle owners by recognising their license plate numbers. Such systems are capable of providing appropriate solutions in several aspects, for example, monitoring traffic within restricted or dangerous areas or even granting entrance permission to employees in companies, etc. An automatic vehicle license plate recognition system is an important topic due to its various uses, such as those in security applications: monitoring institution entrances or vehicles on the road, detecting stolen cars, etc. These systems are of great interest because of their benefits for everyday life. The system starts with the stage of image acquisition until identifying the contents of the license plate. It goes through image preprocessing, license plate localisation in the image, and then dividing the contents of the license plate (character segmentation). The last stage involves extracting basic symbols to identify them using OCR (Abdullah et al., 2009).

***Corresponding author:** Aeyman M Hassan: *a.hassan@zu.edu.ly*, School of Computer Engineering, University of Zawia, Zawia, Libya.

Many challenges affect the proper functioning of the system. Most difficulties come from the surrounding environment since these systems are normally designed to work in outdoor settings. The variations in lighting conditions, rain, and fog are among such difficulties. All these environmental conditions affect the quality of the captured image, which thus reduces the effectiveness and efficiency of the system (Selmi et al., 2020).

There is a significant amount of research looking into solving this problem. Korean license plate detection was done using a vertical edge. The character recognition was based on a template matching method (Yu & Kim, 2000). Shohreh Kasaei et al. used the same recognition method for Iranian car plates, but they used morphological parameters in the segmentation stage instead (Kasaei et al., 2010). Although template matching recognition might give a reasonable result, it has a large computation time because the comparison operation is performed pixel by pixel, in contrast with other machine learning approaches which deal with image features rather than pixels at the recognition stage, as like the one used in this work. In another study (Duan et al., 2005), Hough transform was utilised to detect vehicle plates and a hidden Markov model for the recognition stage. Arth et al. used the same method to localise a vehicle plate. SVM was used for number recognition instead (Arth et al., 2007). YOLO (You Only Look Once) deep learning was used to recognise Taiwan's car license plates (Chen, 2019). Different versions of YOLO were applied in each stage of the recognition algorithm (Henry et al., 2020). Laroca et al. used two convolution neural networks (CNNs) for vehicle and license plate detection. Two CNNs were also used for letters and digits recognition (Laroca et al., 2018). Although deep neural networks were successful in many recognition tasks, you have to provide a large amount of data during the training stage (Björklund et al., 2019; Li & Shen, 2016; Selmi et al., 2020). Due to the limitation of the dataset size used, a decision was made to use histogram projection and support vector machine for the detection and recognition stage, respectively, instead of deep learning approaches (Singh & Bhushan, 2019).

Several pieces of research have been conducted for Libyan vehicle license plate recognition. Two methods have been applied in the recognition stage, which are template matching and an approach based on endpoints (Algablawi et al., 2013). Abdella used vertical and horizontal projection to detect the license plate, while template matching was also used for number recognition (Abdella, 2016). A Combination between template matching and a probabilistic neural network was used to recognise digits in Libyan license plates (Jabar & Nasrudin, 2016). Although the current research might obtain a reasonable result in the recognition stage for Libyan vehicles using the template matching method, this method is not shift or scale-invariant. Furthermore, comparing images pixel by pixel causes an increase in the computation time. Vehicle license plates vary from country to country in colour and shape. They also have different contents, of which many are digits, while others are a mix between numbers, characters, and symbols. All these variations influence the choice of an appropriate method that can do well with a particular application. Therefore, a kind of balance between accuracy and computation time must be considered. This paper aims to design and implement a reliable recognition system, which is able to recognise Libyan vehicle license plates despite the small dataset size. Features extraction is performed in this work instead of methods that mainly work at the pixel level. The recognition stage was performed using one of the machine learning algorithms. Furthermore, building a small model to simulate a vehicle control access to institutional premises.

The paper is structured as follows. The second section demonstrates the methodology of this work in terms of hardware and software. The details about the data and experiments are discussed in the third section. It also presents a comparison with other approaches. The last section of this paper is about summarizing the research findings. In addition, it suggests potential future works that might assist in improving the current results.

MATERIALS AND METHODS

The system's model contains two vehicle gates. The first gate is specified for the entrance, while the second is for the exit, as shown in Figure 1. This is what might usually be found in a real company building. To clarify the working procedure, a vehicle stops at the entrance gate. At this moment, the driver still sees a red light on the gate.

Figure: (1). Model of the vehicle plate recognition system

An ultrasonic sensor detects the vehicle and sends a signal to a MATLAB algorithm. The algorithm starts working by sending a command to the camera mounted on the gate to capture the vehicle. After completing the operation of number recognition for the current license plate, there are two possibilities. The first possibility is to find the number within the database list for those who have permission to get inside. A green light will be on, and the gate will be opened automatically. In the second case, the vehicle is not allowed to enter. The gate will be kept closed and the person will be alerted with the red light. The working of the exit gate is rather simpler than that. There is no need to validate the license plate number. The ultrasonic sensor detects the vehicle here as well. Consequently, the red light will be turned off, and the green light will be on. Finally, the gate opens to allow the vehicle to exit. This gate is not connected to the recognition algorithm. The gate was built using Arduino and ultrasonic sensors. Figure 2 shows the connection between all system components.

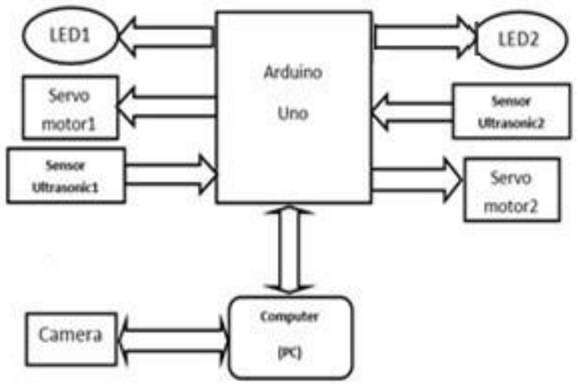

Figure: (2). A block diagram of our algorithm

HARDWARE

This section explains the specifications of each component that is used in this work and clarifies its role in achieving the final task.

Arduino: It is the main development board that is used to control vehicle gates in this simple model. It has a microcontroller board based on (ATmega328) with 14 I/O digital pins. Arduino Uno was chosen for the circuit for the sake of simplicity.

Ultrasonic sensor: The working principle of this sensor is based on sound waves. It consists of four pins (Vcc, Trigger, Echo, Ground). It works by sending a wave at the speed of sound that travels through the air. Once it hits an object in its way, the wave then returns and is collected by the sensor receiver. The following equation is used to calculate the distance.

$$D = \frac{t * 0.034}{2} \qquad (1)$$

Where D is the required distance, 0.034µs is the speed of sound, t is the time it takes for the sound waves to return to the ultrasonic sensor after striking an object. The HC-SR04 module is chosen for this model.

Servo motor: This kind of motor is a DC (Direct current) motor equipped with an electronic circuit to precisely control the position and direction of rotation of the motor shaft.

Camera: In this project, a USB-type camera (Microsoft Life Cam VX-800) is used in this model. The connection between all hardware components is shown in Figure 3.

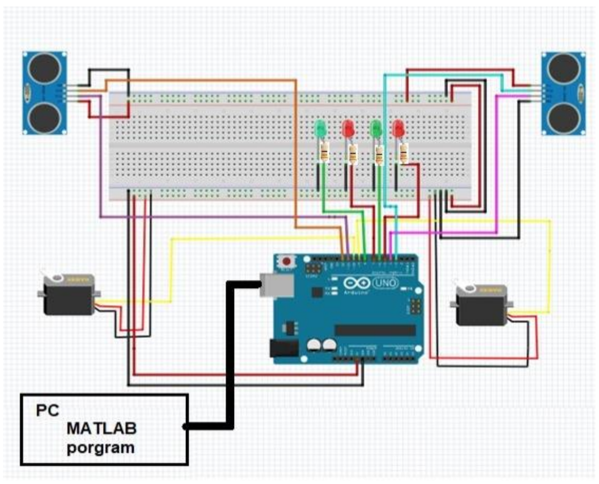

Figure: (3). Gate controlling circuit

SOFTWARE

The algorithm starts with image preprocessing operations to prepare an image for the three main stages of the recognition algorithm: car plate detection, number segmentation, and number recognition, respectively.

Image Preprocessing: The algorithm converts the captured image from an RGB image (Red, Green, Blue) to a greyscale image. The pixel intensity of the grey image is in the range (0-255). The value 0 represents the black colour while 255 is the white colour, and any value in between is a level in the grey colour. The algorithm takes the grey image as input and then converts it to a binary image, which is also called a logical image. The binary image has two possible intensity values, either 0 or 1 which represent the black or white colour respectively. This operation is performed by choosing a threshold value in which pixel values equal or above the threshold are set to 1 (white) while others are set to 0 (black). By converting the image from coloured to binary, unnecessary information is reduced, especially when image processing is machine-oriented. On the other hand, humans care considerably about image details and resolution. Figure 4 shows the steps of converting an image to a binary type.

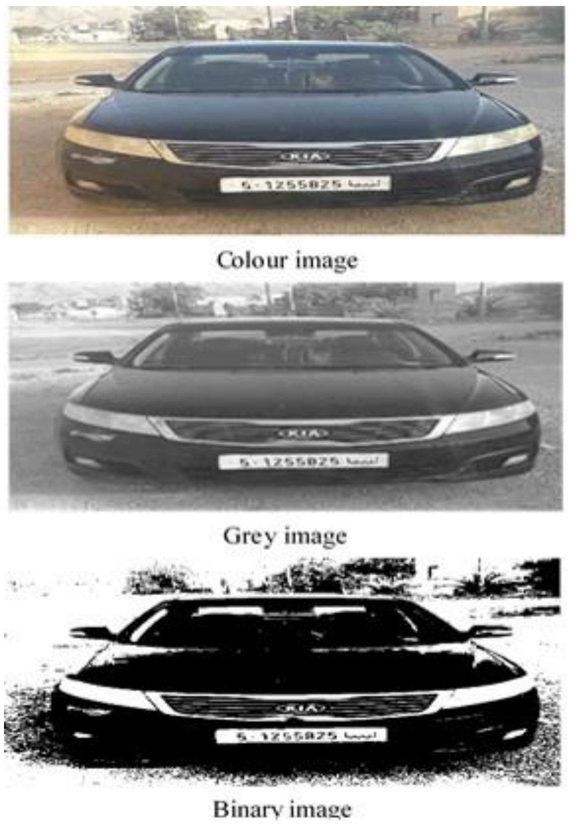

Figure: (4). Image preprocessing

Car Plate Detection: The input at this stage is a binary image. This image contains a vehicle and other objects in the scene. The output is an image of the extracted car license plate. This stage involves the following steps: edge detection, vertical and horizontal edge histogram projection, and finally, choosing the highest probability within image segments to be a license plate among all candidates.

Edge detection: This operation is performed using a Sobel filter. The filter calculates the image gradient from the intensity of its pixels. It finds the direction of the greatest gradient. The filter uses two matrices as shown in Equation 2. The first matrix finds intensity changes in the horizontal direction, while the second matrix is dedicated to the vertical direction (Kanopoulos et al., 1988).

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 1 & 0 & -1 \\ 1 & 2 & 1 \end{bmatrix} * A \qquad (2)$$

Where A represents an original image, G_x and G_y are the gradients in both the x and y-direction. To determine G_x and G_y, the filter kernel is moved over the input image to calculate the value of one pixel and then shift the kernel one pixel to the right. When reaching the end of the current row, the filter is placed at the beginning of the next row, as shown in Figure 5. The following equation illustrates an example of calculating one value of G_x.

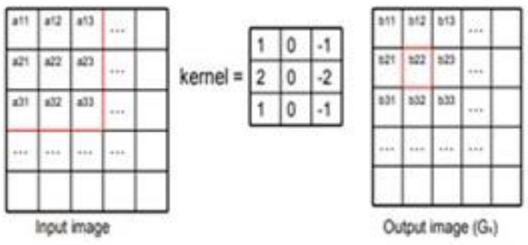

Figure: (5). Sobel filter implementation

$$b_{22} = a_{13} - a_{11} + 2 * a_{23} - 2 * a_{21} + a_{33} - a_{31} \qquad (3)$$

Equation 4 illustrates the calculation of the gradient magnitude, while Equation 5 calculates the gradient direction.

$$G = \sqrt{{G_x}^2 + {G_y}^2} \qquad (4)$$

$$\theta = \frac{G_y}{G_x} \qquad (5)$$

The obtained result after applying the filter on the image is shown in Figure 6.

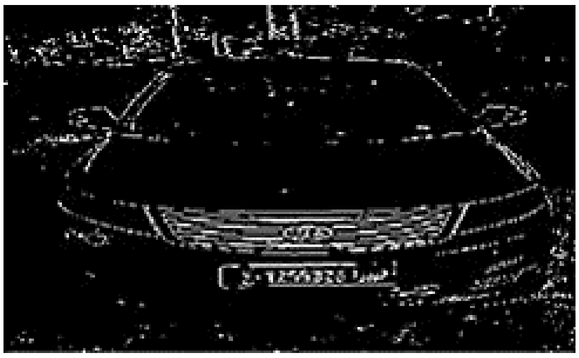

Figure: (6). Edge detection with Sobel filter

Histogram projection: In this step, the algorithm finds projections of the horizontal and vertical edge histograms for the obtained image from the last step. It processes column by column in the horizontal histogram. In each column, it starts by subtracting the second-pixel value from the first one. If the result exceeds a threshold value, it will be stored in a list, then moved down to the next pixels, and the same processes will be repeated until passing through the entire column. The values are added up to get the first value in the differences group. The algorithm moves to the second column and so on until reaching the last column in the image, as shown in Figure 7. The same procedure is applied to the vertical histogram, as shown in Figure 8, but the process is applied on rows instead (Jagannathan et al., 2013). As shown in Figure 7 and Figure 8, the noise noticeably appears at the edges of the horizontal and vertical histograms.

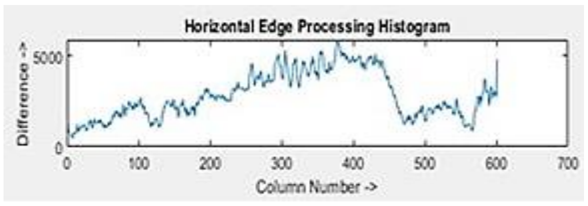

Figure: (7). Horizontal edge processing histogram

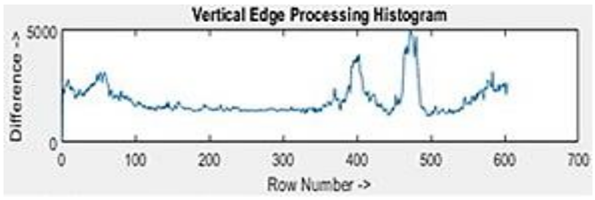

Figure: (8). Vertical edge processing histogram

A low-pass filter is used to smooth the line curves, as shown in Figure 9 for the horizontal edge and Figure 10 for the vertical edge. If the region has high horizontal and vertical histogram values, it is highly likely to be a vehicle license plate. Other regions are removed from an image by applying a dynamic threshold on both horizontal and vertical histograms. The dynamic threshold is equal to the average value of the histogram. The output of this operation is a graph showing the areas that have a high probability of being license plates, as shown in Figure 11 and Figure 12.

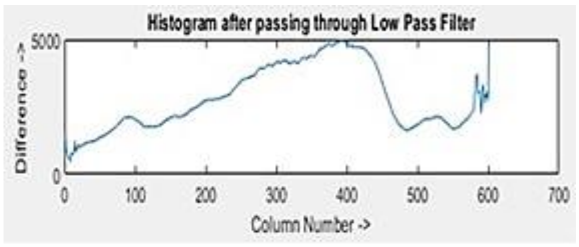

Figure: (9). Horizontal histogram after low pass filter

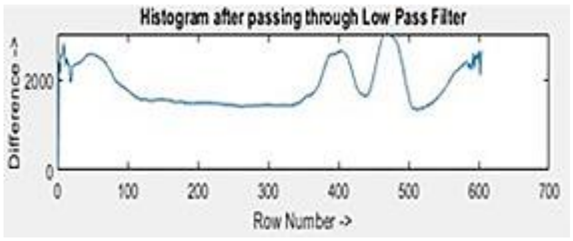

Figure: (10). Vertical histogram after low pass filter

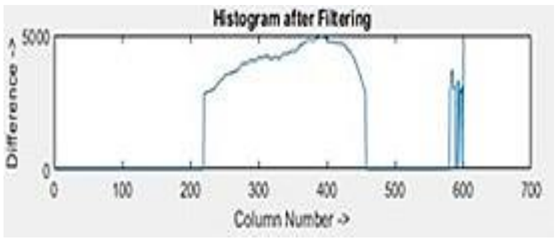

Figure: (11). Horizontal histogram after filtering

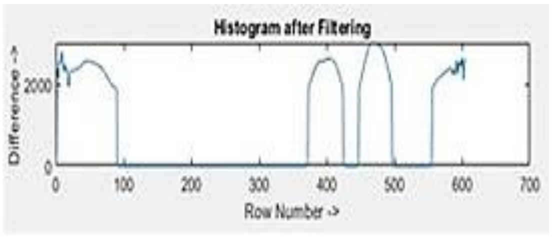

Figure: (12). Vertical histogram after filtering

The coordinates of all candidates are stored in an image array, as shown in Figure 13. The algorithm finds a common area with a maximum value in the horizontal and vertical graphs. This area is extracted as it represents the license plate. Then, the resulting image is converted to a binary image, while small objects are discarded, (which cannot be digits), as shown in Figure 14.

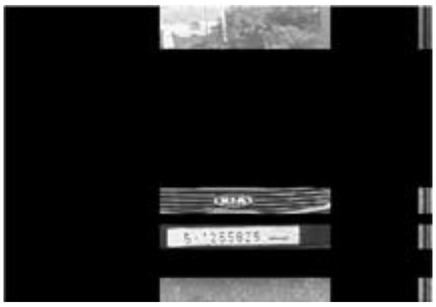

Figure: (13). Candidate areas

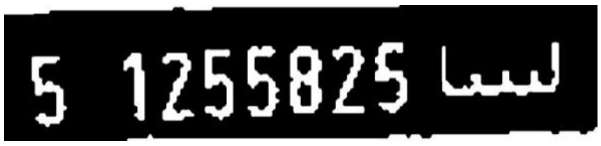

Figure: (14). The result of car plate detection

Number Segmentation: After acquiring the license plate, the algorithm separates each number in order to recognise the full series in the last stage. It is also discarding other objects which cannot be digits. This work is achieved using connected-component labelling (He et al., 2009).

The connected labelling components use a 4-connected neighbourhood or an 8-connected neighbourhood. For the 4-connected neighbourhood, they are the neighbours' pixels that share an edge with pixel p. While in the 8-connected neighbourhood, it is a set of pixels that share an edge or a corner with pixel p, as shown in Equation 6.

$$ccl_4 = \begin{matrix} & 1 & \\ 4 & p & 2 \\ & 3 & \end{matrix}, \; ccl_8 = \begin{matrix} 8 & 1 & 2 \\ 7 & p & 3 \\ 6 & 5 & 4 \end{matrix} \qquad (6)$$

$$ccl_4 = 1 \;\; 4\, p\, 2 \;\; 3 \;, \; ccl_8 = 8\; 1\; 2\; 7\; p\; 3\; 6\; 5\; 4$$

Each foreground pixel becomes a node in a tree graph. Two nodes are connected if the corresponding pixels are neighbours. When applying the connected-component labelling to the image in Figure 14, the result is a set of images in Figure 15. The ratio between the width and height of those images is used to exclude other objects that cannot be digits. For instance, the last word in Libyan vehicle plates (ليبيا) has a high-ratio value. Therefore, it will be removed. Furthermore, the connected-component labelling starts extracting objects in an image from top to down and left to right. Thus, it is suitable for Libyan vehicles license plates. These license plates have a rectangular shape, and all numbers appear in one line. Therefore, there is no need to do any kind of reordering after finishing the recognition stage.

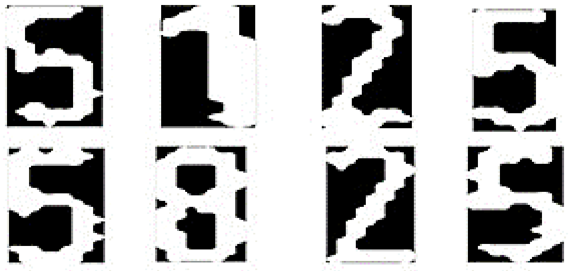

Figure: (15). Number segmentation

Number Recognition: At this point, the algorithm reaches its last stage. It classifies images into digits between 0 and 9. It also discards other objects with a low confidence rate. Therefore, data must initially be provided for the recognition algorithm to be used in the training step. The recognition stage consists of two main steps: features extraction using histogram of oriented gradient (HOG) and number classification using SVM.

Data collection: A group of one hundred images of Libyan vehicles has been captured from the front view. They were separated into two groups, a training group of seventy images to be used in classifier training and a test group of thirty images for algorithm evaluation.

Features extraction using HOG: one of the successful results of HOG was for pedestrians' detection (Dalal & Triggs, 2005). After that, HOG became widely used for detecting other objects such as animals, vehicles, motorcycles, etc. HOG starts by calculating the gradient using a one-dimensional (1D) discrete derivatives mask in both vertical and horizontal directions, as shown in Figure 16. The second step is building cell histograms. The histogram channel ranges from 0 to 180 or 0 to 360 degrees. An orientation-based histogram channel is determined from a weighted vote for each pixel within the cell, as shown in Figure 17. The weighted vote is based on the calculated gradient magnitude. The aim of the third step is to calculate descriptor blocks from the normalised cell histogram. Each block consists of four 8x8 pixel cells (16x16 pixels per block) with nine histogram channels. The last step in HOG is to normalise the descriptor blocks to mitigate the visual effect induced by light variations. The normalisation is performed using the L2 norm, which is a standard method to compute the length of a vector in Euclidean space. The images in Figure 15 were cropped to size 17 x 17 pixels before applying feature extraction in the training and test step of the SVM classifier.

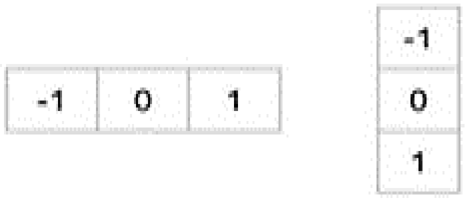

Figure: (16). Gradient kernels

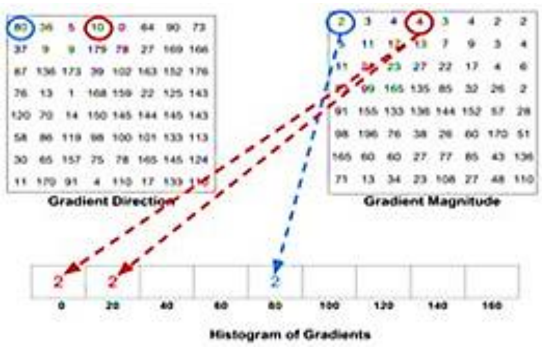

Figure: (17). Histogram of oriented gradient

Number classification using SVM: In machine learning, supervised learning is an algorithm that uses some labelled examples to train its classifier. As a result, the classifier will be able to predict other examples that have not been seen before. Due to its flexibility and simplicity, the support vector machine has been chosen for classification. SVM is a linear classifier that is used for data classification. It can learn through provided examples to predict the correct class among unseen examples. It is sometimes called a binary classifier because it separates two classes of data points via a hyperplane.

The SVM outputs a map of labelled data with a maximum margin. The margin represents a symmetrical distance on both sides of the hyperplane. The SVM tries to maximise this margin as much as possible. The data points near the hyperplane are called (Support-Vector). Figure 18 illustrates how the support-vector machine method distinctly separates data points (Noble, 2006).

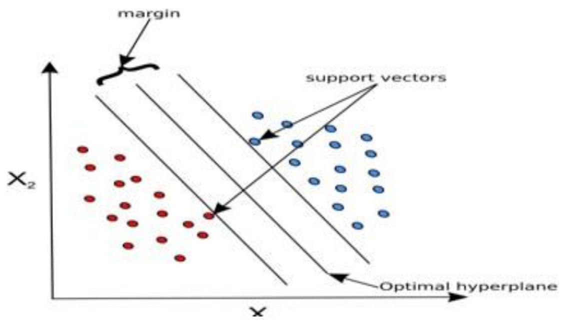

Figure: (18). Support Vector machine

The current research aims to recognise digits between 0 and 9. Although the support vector machine is a binary classifier, it can still be adapted for multiclass classification. This is performed using the One-vs-All approach. It finds many classifiers; each classifier can distinguish one digit from the rest. The SVM has a parameter called a kernel. The kernel is a mathematical function that specifies the shape of the SVM hyperplane such as nonlinear, linear, polynomial, radial basis function (RBF), Gaussian, and Sigmoid, as shown in Figure 19. The RBF kernel has been selected in the selected algorithm based on experiments and results.

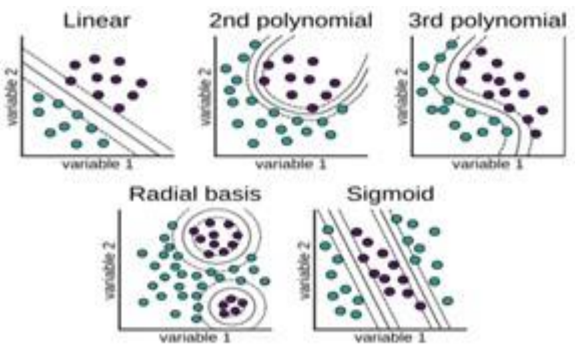

Figure: (19). SVM kernels

SVM is well known for its performance even with a small dataset, as in this case. In addition, it is being considered a brilliant classifier for binary classification.

RESULTS AND DISCUSSIONS

Many experiments have been conducted to evaluate the used algorithm's performance. In the training stage, different SVM kernels were tested using seventy images to decide which ones of them would be the best. As a result, the RBF was selected as a kernel. After being trained, the algorithm was tested on a group of test images. These images have not appeared along with the training processes. The test group represents 30% of the dataset. The algorithm recognises twenty-five images correctly out of thirty with satisfactory computation time (5 seconds). The accuracy of the used software is evaluated at 83.3%. According to the small size of the dataset, this is an encouraging result. Abdullah et al. obtained an accuracy of 80% for Malaysian vehicles (Abdullah et al., 2009). They also used SVM for the recognition process, but both results cannot be compared because of the difference in datasets in terms of their size and format. YOLO deep learning achieved 98.22% accuracy on license plate recognition. The computation time was less than one second (Chen, 2019). Yu et al. focus on plate localisation, which can lead to accuracy improvement for license plate recognition. Their method was based on a wavelet transform and empirical mode decomposition (EMD). The accuracy of their license plate detection was 97.91% (Yu et al., 2015). End-to-end deep learning architecture has achieved a recognition rate of more than 95% on four Chinese test sets. The computation time of this method was only 0.2 seconds (Wang et al., 2020). The result of the current study was also compared with results obtained from different methods dedicated for Libyan vehicles in terms of accuracy for each stage and overall accuracy as shown in Table 1.

Table: (1). Performance measurement

Method	Detection (%)	Segmentation (%)	Recognition (%)	Overall accuracy (%)	No. images
(Abdella, 2016)	67.5	-	81.4	25	200
(Algablawi et al., 2013)	96.15	85	92.19	75	104
The proposed method	97	92	96	83.3	100

Figure 20 shows the MATLAB graphical user interface (GUI) used. This GUI has an option to load a stream from a camera. The figure also displays three examples of successful results under different light conditions, where pictures were taken at different times during the daytime. The angles of the capturing were slightly different among images. However, all images contain the front part of vehicles because the system is designed for gate entrance. The algorithm was able to recognise the vehicle license plate number and present the result as a text at the bottom of the MATLAB GUI. As mentioned before, there are three stages for the recognition algorithm. The algorithm might fail at any stage of them. As a consequence, it affects the software performance. Figure 21 displays two instances of unsuccessful results. The source of error in the top image was due to the detection step. The software could not localise the correct position of the vehicle license plate. Therefore, it failed in the subsequent stages. In the second example of Figure 21, the algorithm succeeded in the first and second stages, but it failed in the last stage. The software was not able to identify the license plate number correctly because the digits were ambiguous. The ambiguity caused the numbers to be cut in the middle, which produced many objects during the segmenting stage.

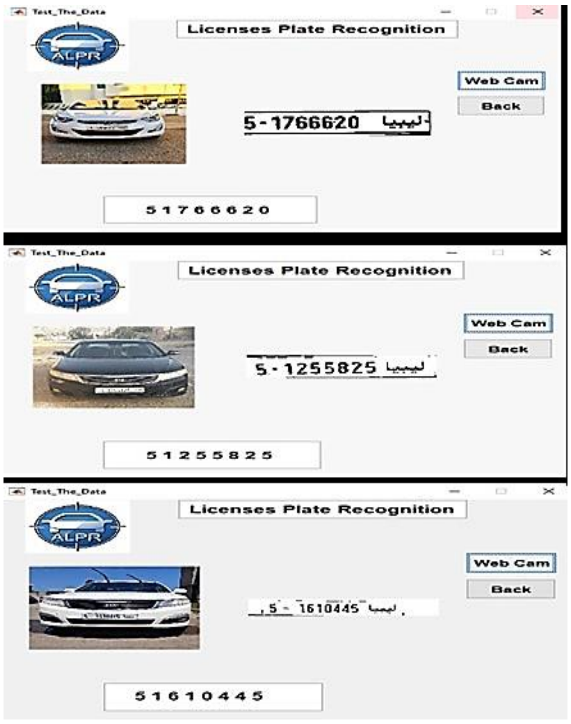

Figure: (20). Successful results

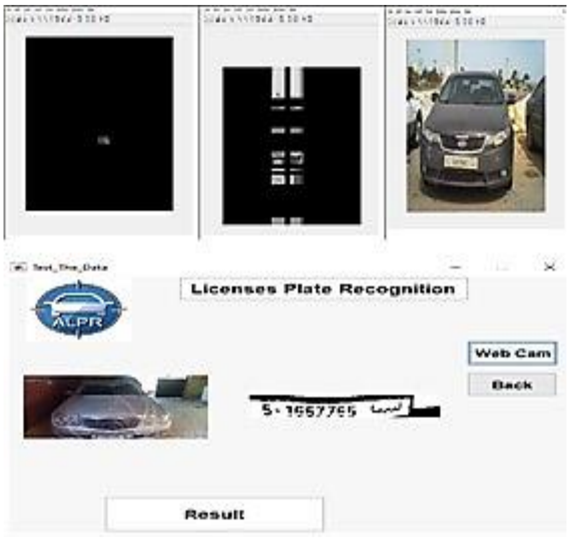

Figure: (21). Unsuccessful results

CONCLUSION

This paper presented a model for recognising Libyan car license plates. The procedure starts by detecting the car using an ultrasonic sensor. The Arduino sends a signal to the interface, which was designed by MATLAB to start capturing the vehicle. Then, the algorithm extracts the location of the vehicle plate in the image using a vertical and horizontal histogram. Next, the number segmentation is accomplished using connected-component labelling, which is followed by histograms of oriented gradients (HOG) for features extraction operation.

Finally, the support vector machine (SVM) is utilized to recognise digits. As a result, the gate is opening while the green led is being turned on for authorised vehicles, otherwise kept closed. As future work, collecting more vehicle images is a target to widen the proposed database in which deep learning can be used to improve the recognition rate. To reduce the cost of the current hardware, an embedded system such as Raspberry Pi could be used as the main board of the recognition software. In addition, a low-level programming language such as C++ could be used to lower the computation time cost.

REFERENCES

Abdella, A. A. E. S. (2016). Libyan licenses plate recognition using template matching method. *Journal of Computer and Communications, 4*(07), 62 .

Abdullah, S. N. H. S., Omar, K., Sahran, S., & Khalid, M. (2009). License plate recognition based on support vector machine. 2009 International Conference on Electrical Engineering and Informatics ,

Algablawi, W., BenAnaif, W., & Ganoun, A. (2013). Libyan Vehicle License Plate Recognition System. International Conference on Elecetrical and Computer Engineering ,

Arth, C., Limberger, F., & Bischof, H. (2007). Real-time license plate recognition on an embedded DSP-platform. 2007 IEEE Conference on Computer Vision and Pattern Recognition ,

Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., & Magli, E .(2019) .

Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recognition, 93*, 134-146 .

Chen, R.-C. (2019). Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. *Image and Vision Computing, 87*, 47-56 .

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05 ,(

Duan, T. D., Du, T. H., Phuoc, T. V., & Hoang, N. V. (200 .(5Building an automatic vehicle license plate recognition system. Proc. Int. Conf. Comput. Sci. RIVF ,

He, L., Chao, Y., Suzuki, K., & Wu, K. (2009). Fast connected-component labeling. *Pattern Recognition, 42*(9), 1977-1987 .

Henry, C., Ahn, S. Y., & Lee, S.-W. (2020). Multinational license plate recognition using generalized character sequence detection. *IEEE Access, 8*, 35185-35199 .

Jabar, K. A., & Nasrudin, M. F. (2016). Libyan vehicle plate recognition using region-based features and probabilistic neural network. *Journal of Theoretical and Applied Information Technology, 94*(1), 104-114 .

Jagannathan, J., Sherajdheen, A., Deepak, R. M. V., & Krishnan, N. (2013). License plate character segmentation using horizontal and vertical projection with dynamic thresholding. 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN ,(

Kanopoulos, N., Vasanthavada, N., & Baker, R. L. (1988). Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits, 23*(2), 358-367 .

Kasaei, S. H., Kasaei, S. M., & Kasaei, S. A. (2010). New Morphology-Based Method for RobustIranian Car Plate Detection and Recognition. *International Journal of Computer Theory and Engineering, 2*(2 .264 ,(

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018). A robust real-time automatic license plate recognition based on the YOLO detector. 2018 international joint conference on neural networks (ijcnn ,(

Li, H., & Shen, C. (2016). Reading car license plates using deep convolutional neural networks and LSTMs. *arXiv preprint arXiv:1601.05610* .

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology, 24*(12), 1565-156 .7

Selmi, Z., Halima, M. B., Pal, U., & Alimi, M. A. (2020). DELP-DAR system for license plate detection and recognition. *Pattern Recognition Letters, 129*, 213-223 .

Singh, J., & Bhushan, B. (2019). Real Time Indian License Plate Detection using Deep Neural Networks and Optical Character Recognition using LSTM Tesseract. 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS ,(

Wang, D., Tian, Y., Geng, W., Zhao, L., & Gong, C. (2020). LPR-Net: Recognizing Chinese license plate in complex environments. *Pattern Recognition Letters*, *130*, 148-156 .

Yu, M., & Kim, Y. D. (2000). An approach to Korean license plate recognition based on vertical edge matching. Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0 ,

Yu, S., Li, B., Zhang, Q., Liu, C., & Meng, M. Q.-H. (2015). A novel license plate location method based on wavelet transform and EMD analysis. *Pattern Recognition*, *48*(1), 114-125 .

التعرف على رقم لوحة السيارات الليبية باستخدام آلة المتجهات الداعمة

أيمن مختار حسين*، سامي عبد المجيد الغول وآية على الكبير

قسم هندسة الحاسوب، جامعة الزاوية، الزاوية –ليبيا

تاريخ الاستلام: 17 سبتمبر 2021 / تاريخ القبول: 06 مارس 2022

https://doi.org/10.54172/mjsc.v37i1.525:Doi

المستخلص: أصبح مجال رؤية الكمبيوتر مستخدم على نطاق واسع في شتى جوانب حياتنا اليومية. أحد هذه التطبيقات المهمة استخدامه لفرض تطبيق القانون. يقدم هذا البحث نموذجا لعملية تصميم، وتنفيذ مدخل للسيارات الخاصة بالمباني باستخدام خوارزمية التعرف على رقم لوحة السيارات الليبية، بالإضافة إلى بناء قاعدة بيانات صغيرة خاصة بصور السيارة الليبية، والتي يمكن استخدامها في الأغراض البحثية، مثل معظم أنظمة التعرف، هناك ثلاث خطوات أساسية، بداية بكشف اللوحة باستخدام الرسم البياني الرأسي والأفقي (vertical and horizontal histogram)، بعد ذلك استخلاص الأحرف، والأرقام بواسطة خوارزمية (connected component labeling) ، الخطوة الأخيرة في نظام التعرف هي التعرف على الحروف optical character recognition (OCR) ، والتي يتم إجراؤها بواسطة إحدى طرق تعلم الآلة المعروفة باسم Support Vector Machine (SVM). تم استخدام لوحة الاردينو (Arduino) للتحكم في فتح وإغلاق البوابة، وفقًا لقائمة السيارات الموجودة في قاعدة البيانات، والمصرح لها بالدخول، تساعد المجسات فوق الصوتية في كشف توقف السيارة عند البوابة، عملية برمجة النظام تمت باستخدام برنامج الماتلاب (MATLAB)، ويتم تشغيلها على جهاز بمواصفات معالج من نوع Core i7 CPU بسرعة 2.20 جيجا هرتز، وذاكرة الوصول العشوائي هي 8 قيقا بايت، والنظام هو ويندوز10. ومع أن عدد صور السيارات (التي تم تجميعها خلال هذا العمل) في قاعدة البيانات المستخدمة محدود، فإن التجارب العملية تعطي نتيجة واعدة من حيث متوسط الدقة وهي: (83.3٪)، وزمن تنفيذ البرنامج هو: (5 ثوانٍ).

الكلمات المفتاحية: آلة المتجهات الداعمة، التعرف على الأرقام، لوحة الاردينو.

* أيمن مختار حسين: a.hassan@zu.edu.ly، قسم هندسة الحاسوب، جامعة الزاوية، الزاوية –ليبيا