

IRAQI LICENSE PLATE RECOGNITION BASED ON MACHINE LEARNING

Dunya A. Abd Alhamza ¹, Ammar D. Alaythawy ²

^{1,2} College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

dunyaali1994@gmail.com ¹, ammar@coie-nahrain.edu.iq ²

Received:7/03/2020, Accepted:26/07/2020

Abstract- The Iraqi License Plate Recognition (ILPR) is useful in many fields, such as private or public exits, parking lots, traffic control, and theft detection. This paper focusing on detecting the plate car and identifying the Arabic number on the plate. The main paper goal was to identify and understand plate cars with numbers / characters in the plate where it was difficult to detect the same object with different environments. It also, offers LPR which consists of three main stages; pre-processing with detect license plates, segmentation and recognition of characters. the first stage takes a camera picture then pre- processing it. License plate detection check for matching license plate to crop the appropriate plate in the picture. Segmentation done separately by splitting the numbers. The final stage is number recognition by using K- Nearest Neighbours (KNN) which is one of the basic machine learning algorithms used to match numbers with training data to provide a prediction that is accurate. The system was implemented using python3.5, open- cv library and, using 50 images, the accuracy performance results equal to 90 percent.

keywords: Iraqi license plate, ILPR, K- Nearest Neighbours (KNN) , Python.

I. INTRODUCTION

In the last years, license plate recognition considers a core technology for security and traffic application that use in traffic control, parking lot and theft surveillance. All cars have a unique identification number called a license plate (LP) The LP shape is rectangular and contains a collection of characters/ numbers. Each country has different types and variables on the license plate. In Iraq three types with different sizes, colors, and fonts. The old type before (2003) in Fig. 1, the second type for north Iraq in Fig. 2, the last type the new Iraqi license plate in Fig. 3 has the character/ number in English and Arabic with different color such as the white color for private car, red for taxi, blue for government etc. Some work used only detect plate [1] , another used segmentation and recognition [2] , and others used the three operations [3]. LPR is an image processing that used to identify cars by their LP. The LPR can be divided into three steps, the first step preprocessing image, segmentation characters/ numbers found in plate image and the last step recognize this segmentation to characters/ numbers, shown in Fig. 4 . Machine learning (ML) is a subset of Artificial Intelligence (AI) that allows software applications to be more careful in predicting outcomes without being candidly programmed. The basic introduction of machine learning is to build algorithms that can draw input data and use statistical analysis to predict a job while updating outputs as new data becomes available [4]. The caused behind this work is to detect and recognize the correct result of the car's plate number and used LPR in many applications which made it very important in real life [1]. The algorithm used in this paper is K- Nearest Neighbours (KNN) which is one of the machine learning algorithms. It is supervised learning, a nonparametric technique and used for classifier and regression in several applications such as image processing, data mining, image recognition, and other application. This algorithm is extremely easy and the output result of KNN depended on K number of training data closest to the required character / number and consider votes among K objects. To implement any application for AI and machine learning, you must use a programming language that is stable,

flexible and has tools available. Python shows all of this, which is why there are a lots of Python AI and machine learning projects today it's used [5].



Figure 1: The old type

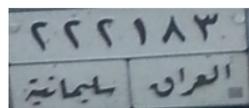


Figure 2: The north type



Figure 3: The new type

II. THE RELATED WORK

As license plate detection and recognition are usually dealt with separately, a brief reference to previous research can be given:

Ng Simin and Florence Choong Chiao Mei [6] proposed the Automatic car- plate detection and recognition system in Malaysian car license plates, the algorithm built is to FPGA (target field- programmable gate array as it's end job and use edge detection and ocr recognition. Rachana Chahar and et al [7] proposed an Automatic license plate recognition system based on image processing using Lab View in India. Eyad I. Abbas and Thaaer A. Hashim [8] proposed an Iraqi car license plate detection and recognition using edge detection and templates correlation for new Iraqi license plate and use in software matlab and hardware with sensor and camera. Yasser M. , Alginahi [9] proposed Automatic Arabic license plate recognition, use NN (neural network) for recognition characters / numbers. Furat Nidhal Tawfeeq and Yasmine Mazin Tabra [10] proposed a Gate control system for the new Iraqi license plate, that uses edge detection and template matching in software then controls the opening of the door gate based on LPR in Iraq. Sweta Kumari and et al [11] proposed Automatic License Plate Recognition Using Open-CV and Neural Network, the algorithm that used NN (neural network) and open- CV to implement ALPR. Table I summarized the comparison between the proposed system and another system. Which is used SVM, KNN and image processing, it first used Gaussian blur filter to reduce noise in images, a horizontal and vertical edge because of a rectangle shape, the morphology operation used for connects objects that fill up small holes

close to each other and smooths the object outline by filling up narrow deviations, then used SVM as a classifier the license plate and lastly used KNN for recognition numbers/ characters. The dataset that used in this work was more than used in a proposed system in SVM is used 100 license plates and 112 non- license plates, and in KNN is used 400 numbers (10 classes, 0~ 9), 5120 letters (40 classes) but in the proposed system used one image with multiple numbers/characters with different sizes. The accuracy of the system was 90%.

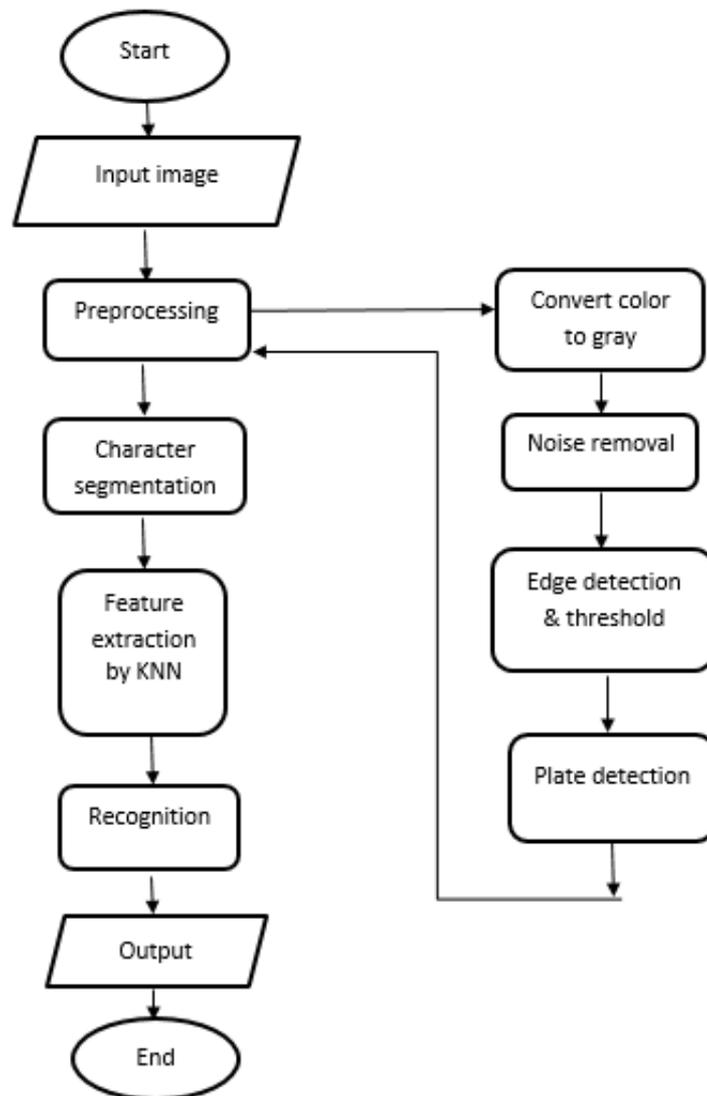


Figure 4: General flowchart of step LPR

TABLE I
 COMPARISON THE PROPOSED SYSTEM WITH OTHERS SYSTEM

Parameter	Proposed System	[12]
K	1	15
Accuracy	90%	90%

III. THE PROPOSED SYSTEM METHODOLOGY

Can explain the methodology of the proposed system at the following points of:

A. *K - nearest neighbor classification*

The classification method was a collection of test data to classes specified by machine learning [13]. Classification of KNN was determined by the amount of data from the distance test with the training data.

The main advantages of KNN for classification are:

- 1) Very simple implementation.
- 2) Robust with regard to the search space.
- 3) Few parameters to tune: distance metric and k.

The main disadvantages of the algorithm are:

- 1) Does not work well with large dataset.
- 2) Need feature scaling: We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.
- 3) Sensitive to noisy data, missing values and outliers [4].

B. *KNN steps of work:*

Step 1: determining the k value distance amount of data, the value of k can be $k = 1, 2, \dots, n$, n can be any number but the best numbers are an odd number to not confuse the select. this paper use $k = 1$ because the output of a recognition plate is a single result.

Step 2: calculate the value of distance test data to each training data by Euclidean formula.

$$Di = \sqrt{\sum_{i=1}^p (x_2 - x_1)^2} \quad (1)$$

The result of this formula is the distance between the binary matrixes of test data with all training data matrix contained in the feature extraction (file) to be classifier [10].

Step 3: Classification the result of distance value.

Step 4: Vote the minimum value of distance.

Step 5: Check the minimum value of the class, then the test images are also included in the class.

Step 6: Recognize characters with reference to the type of class.

Algorithm (1): Finding number of car by Using KNN

Input: Capture Image.

Output: Numbers / Characters.

Begin.

Scenario 1: Convert RGB image to Grayscale.

Scenario 2: Transforms Grayscale image to binary image & Filter Morphological Transformation.

Scenario 3: Filter Gaussian for Blurs image.

Scenario 4: Finding all the contours in the image.

Scenario 5: Search & recognize all possible characters in an image.

Scenario 6: Crop part of the image with the highest candidate Plate.

Scenario 7: Crop the Plate from the original image.

Scenario 8: Split the numbers/ characters separately.

Scenario 9: Matching to recognize the numbers/ characters.

Scenario 10: Print the numbers/characters in the command window and in GUI.

End

C. Feature extraction

A good feature set contains discriminating information, which can distinguish one object from other objects. It must be as robust as possible in order to prevent generating different feature codes for the objects in the same class. The selected set of features should be a small set whose values efficiently discriminate among patterns of different classes, but are similar for patterns within the same class [3].

D. Training data

The system proposed was designed for the ILPR system. Starting the KNN method needs to train data using Arabic characters / numbers images with different font types and sizes. As shown in Fig. 5 [9] the training done by split each character in the image then extracted features from all image samples then saved in a file and will be classified the feature by KNN algorithm to numbers and characters in another file and finally converted these samples to English numbers / characters.

E. Extraction license plate of car

The operations are divided by parts and Fig. 4 it explains briefly:

1) Preprocessing

The camera takes the first step in pre- processing, and this image converts to grayscale image (RGB → grayscale) in Fig. 7, then applying a binary image (0- 1) and morphological transformation to the edges (to clarify the basic edges of an image) found in Fig. 8 .Use the Gaussian blur filter to reduce noise and make the image more smoothly in Fig. 9.

2) License plate detection

After image processing by morphological transformation with open- CV using contours tool to detect license plate by finding all contours image in Fig. 10 depended on matching features that saved with training and features possible in image and then extracted the correct plate in Fig. 11, then converted the plate to grayscale and binary in Fig. 12 (a & b).

3) Segmentation

Fig. 13 , each character/ number has been individually segmented for easy identification after extraction of the plate and converting to binary, by contour tool to draw small rectangular around each number.

4) Recognition license plate

The last step of LPR was to match these segmented characters / numbers according to features extracted given by KNN and the classification to recognize the numbers of the plate by OCR (optical character recognition) to show the characters. Each number/ character classify according to the classify that converted to English numbers /characters, which saved in a file. printing the last result of the license plate was on the image in English numbers/ characters, command window by using python GUI (graphical unit interface) in Fig. 14 (a & b & c) .



Figure 5: The data

IV. EXPERIMENTAL RESULTS

All tests were performed on a 4 GB RAM 2.53 GHz Intel Core i5 processor and implemented with the open- CV library using python3.5. In the experiment, 50 image samples were used in various sizes (in width and length, with high resolution, the maximum, minimum distance of the car was (1- 2 meters) , and the car was parked) and the car’s front and back images were taken by a mobile camera for the test. This work was used a simple method of machine learning (KNN) and with the minimum data training. The system’s accuracy for 50 images is 90 percent, as 45 images can be recognized and for some reason, 5 images cannot be recognized. This may be

- 1) dirty, ambiguous, or damage license plate ,
- 2) Famine light ,

3) shadow characters/ numbers in license plate due to light angle, which is why it is difficult to recognize characters/ numbers.

The accuracy equation is [10] :

$$\begin{aligned} \text{Accuracy} &= \frac{\text{correct data}}{\text{data total}} \times 100\% \\ &= \frac{45}{50} \times 100\% \\ &= 90\% \end{aligned} \quad (2)$$



Figure 6: Input image (RGB)



Figure 7: Grayscale image



Figure 8: Binary image & morphological transformation



Figure 9: Gaussian filter

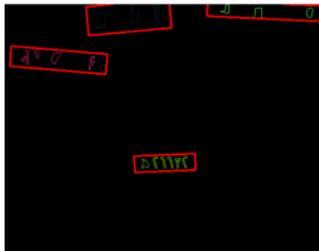


Figure 10: All contour



Figure 11: Correct plate



Figure 12: (a) Gray scale (b)Binary



Figure 13: Segmentation plate



Figure 14(a) Result in image

```
Python34Shell
File Edit Shell Debug Options Window Help
Python 3.5.4 (tags/v3.5.4:68389, Aug 8 2017, 02:17:10) [MSC v.1900 64 bit (AMD64)]
>>>
***** RESTART: C:\Users\PC\Downloads\opencl\Main.py *****
4 possible plates found
license plate read from image = H26672
>>>
```

(b) Result in command window



(c) Result in GUI

Figure 14: (a & b & c) Recognition of license plate

V. CONCLUSIONS

This paper developed an automatic electronic system for Iraqi License Plate Recognition based on the KNN method. The performance of the KNN classifier method under noise is investigated and compared with the vehicle license plate recognition. This work show that the overall accuracy performance of KNN classifier is 90% and it seems that this classifier has better performance than other methods.

REFERENCES

- [1] Al- Shemarry, Meeras Salman, Yan Li, and Shahab Abdulla, "An Efficient Texture Descriptor for the Detection of License Plates from Vehicle Images in Difficult Conditions" , IEEE Transactions on Intelligent Transportation Systems, 2019.
- [2] Khare, Vijeta, et al, "A Novel Character Segmentation- Reconstruction Approach for License Plate Recognition" , Expert Systems with Applications 131, 2019, pp. 219- 239.
- [3] Hidayah, Maulidia Rahmah, Isa Akhlis, and Endang Sugiharti, "Recognition Number of The Vehicle Plate Using Otsu Method and K- Nearest Neighbour Classification" , Scientific Journal of Informatics 4.1, 2017, pp. 66- 75.
- [4] Muller, Andreas C. and Sarah Guido, "Introduction to Machine Learning with Python: A Guide for Data Scientists" , Oreilly Media Inc. , 2016.
- [5] Raschka, Sebastian, "Python Machine Learning" , Packt Publishing Ltd, 2015.
- [6] N. Simin and F. Choong Chiao Mei, "Automatic Car- Plate Detection and Recognition System" , EURECA, pp. 113- 114, 2013.
- [7] J. Kumawat, H. G. Bhavsar and R. Chahar, "Automatic License Plate Recogization System Based on Based on Image Processing Using LabVIEW" , International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 4, pp. 999- 1002, April 2014.
- [8] E. I. Abbas and T. A. Hashim, "Iraqi Cars License Plate Detection and Recognition System using Edge Detection and Template Matching Correlation" , Eng. & Tech. Journal, vol. 34, pp. 257- 271, 2016.
- [9] Alginahi, Yasser M. , "Automatic Arabic License Plate Recognition" , International Journal of Computer and Electrical Engineering 3.3, 2011, pp. 454- 460.
- [10] Tawfeeq, Mr Furat Nidhal, and Mrs Yasmine Mazin Tabra, "Gate Control System for New Iraqi License Plate" , Iraqi Journal for Computers and Informatics ijci 41.1, 2014, pp. 1- 3.
- [11] Kumari, Sweta, Leeza Gupta, and Prerna Gupta, "Automatic License Plate Recognition Using OpenCV and Neural Network" , International Journal of Computer Science Trends and Technology (IJCTST) 5.3, 2017 , pp. 114- 118.
- [12] Kim, Jung- Hwan, et al, "License Plate Detection and Recognition Algorithm for Vehicle Black Box" , International Automatic Control Conference (CACCS) , IEEE, 2017.
- [13] Faaeq, Ainuddin, Huseyin Guruler, and Musa Peker, "Image Classification Using Manifold Learning Based Non- Linear Dimensionality Reduction" , 2018 ,26th Signal Processing and Communications Applications Conference (SIU), IEEE, 2018.