Implementation of k-Nearest Neighbor for PC-Based Character Recognition of Philippine Vehicle Standard License Plate

Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña and Cristina P. Dadula

***Abstract*—This study focused on the development of a PC- based licensed plate recognition system using Visual Basic programming language. A system that is able to recognize Philippines' currently used standard vehicle plate numbers using EmguCV image processing and K-Nearest Neighbor machine learning algorithm. The system accepts image as an input or a snapshot of the image from video of a moving vehicle. There were twenty-two (22) unique images of a vehicle in acquired in 3 different positions: upright position (UP), skewed to right position (SR), and skewed to left position (SL). Image processing techniques were applied to the images such as grayscale conversion, Gaussian blurring, and thresholding. Another processing is the detection of plate number area. Optical character recognition is applied to this area where the characters in the image were segmented and individually recognized. The output equivalent characters and the cropped region of the plate number area are displayed on the user interface. The results showed that in: UP, the system recognition accuracy is 83.12%; in SR, it is 39.97%, and 46.21% in SL. The best system accuracy rate was obtained when the captured image of the vehicle is in the upright position which is 83.12%. For better performance, future works may consider the use of exact font style, different angle and position of the license plates, and different lighting conditions of sample license plates for training**

Keywords: EmguCV, PC-Based, KNN Machine Learning Algorithm, Blob Analysis

I. Introduction

Originally, in modern technology, licensed plate recognition (LPR) systems are developed for security purposes and automation. LPRs are integrated into intelligent transportation systems and become possible to automatically monitor motorway collection, analyze traffic and intersection roads, improve law enforcements, and many more [1]. It is believed that currently, more than half a billion cars have their vehicle identification number (VIN) also known as license plate number as their primary identifier. The vehicle's identification number is the identifying code for a specific automobile. VIN serves as the car's primary "fingerprint" as no two or more vehicle has the same identification number. A VIN can be used to track recalls, registrations, warranty claims, thefts and insurance coverage [1].

Our country is confronted with problems regarding traffic violations in road intersection. Common road accidents in everywhere are attributed to the collision of vehicles, pedestrian, or a collision with an object that would result to death, disability and damage to property. These road accidents were caused by driver's errors (26%), over speeding (18%), mechanical defect (12%), drinking spree before driving (1%), and damaged roads (5%) [2]. The common causes of crashes in the intersections of streets according to studies are drivers' negligence and recklessness where drivers are running through red lights, ignoring the yield and stop signs [3]. Beating red light (traffic signal light which means to STOP) is one of the minor problems we are facing today and most of the time, we tend to ignore it, not knowing that it might be a cause of an enormous accident that can probably result to a sudden death of victims. According to Department of Transportation and Communications – Land Transportations Office (DOTC-LTO) of the Republic of the Philippines, it is a crime against the law to disregard traffic signs, and violators shall pay a fine of P1, 000.00 [4]. With a numerous amount of accidents and traffic rule violations, identification of vehicles has become a task of prime importance. With various optical character recognition (OCR) techniques, any valuable information such as vehicle license plate number is obtainable from captured images [5].

Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña and Cristina P. Dadula, Mindanao State University, General Santos City, Philippines (e-mail: cris_dadula@yahoo.com)

A means of technology such as, CCTV camera, road intersections incidents can now be monitored with the help of a video analyst. They can interpret and review CCTV footages [6]. They can help track those who beat traffic lights that might cause serious accidents and report it immediately to authority in order for them to stop tolerating these behaviors and receive proper sanction. Thus, developing a PC-Based automatic license plate recognition is the goal of this study which attempts to automate the identification of vehicles. The system could probably help lessen the hours spent for investigations as well as strengthen the enforcement of law on traffic lights to avoid road accidents. Also, this study makes use of the *EmguCV* Library references for Microsoft Visual Studio Community 2017 - Windows Forms Application (.NET Framework), Visual Basic as its language, and K-Nearest Neighbor as character recognition training machine.

II. Plate Number Detection

Vehicle detection is necessary to perform license plate recognition systems from a video input. An easy way to do vehicle detection is by using a background subtraction algorithm and the tracking of the vehicle can be achieved by using blob tracker algorithm (cvBlob or OpenCVBlobsLib) or blob detection [7]. The cvBlob library provides some methods to get the centroid, the track and the ID of the moving objects. A bounding box can also be set to draw on the frame surface, and also the centroid and angle of the tracked object. [8]. After vehicle tracking, the centroid of the moving object will be checked if it has crossed the region of interest or virtual line in the video. So when a vehicle passes through the virtual line, a frame capture will be executed in the system.

Images taken enter image processing through a series of algorithms which provide an alpha numeric conversion of the characters on the input image into text form [9]. The system localizes the plate number region and looks for the characteristics that would indicate that the chosen object is a license plate. Hence, the searching would continue to process until such time that a series of characters would be recognized by the system by using OCR [10], [11].

III. Methodology

The system developed accepts both image and video as input and gives an output of the vehicle's licensed plate number in both image and text format.

A. Equipment Used

This study makes the use of personal computer for software application development, steel tape for measuring the distance between a capturing device and a vehicle plate number, and a digital camera as a capturing device.

Preparation of PC Software Application

Free edition of Microsoft Visual Studio 2017 community from https://www.visualstudio.com/vs/ was downloaded and installed with default options of the IDE. Also downloaded and installed for free were the latest version of *EmguCV* executable installer without *cuda* support, and libemgucv-windows-universal-3.0.0.2157.exe. Visual Basic is the programming language and *EmguCV* is a suitable library that wraps *OpenCV* in .NET form [12] [13].

GUIs Form Layout

Visual Basic – Windows Forms Application was selected for starting up a *New Project*. A total of four (4) forms were created in Visual Basic.

- License Plate Recognition – Character Training (KNN Algorithm)
- License Plate Recognition – Testing the Trained System
- License Plate Recognition – Image of Vehicle with License Plate (Input)
- License Plate Recognition – Video Clip of Moving Vehicle (Input)

B. Image and Video Acquisition

Images of vehicle with standard license plate were captured using digital camera from selected parking lots around General Santos City. A total of twenty-two (22) unique vehicles were captured in three (3) different positions as shown in Figure 1: Group A, Group B, and Group C. These were captured in upright position, skewed to right position, and skewed to left position, respectively. The images were properly cropped and scaled.

(a) (b) (c)

Fig. 1. Different Capturing Positions, a) Group A; b) Group B; c) Group C

The video inputs used in this study were recorded from two (2) selected places in General Santos City, one (1) video per location was considered or testing. Only one vehicle per video was captured and tested for the system. A snapshot from each video is shown in Figure 2, wherein *Input 1* was obtained from a university main entrance, and *Input 2* was from a mall parking lot entrance.

(a)

(b)

Fig. 2. Snapshot from each Video, a) Input 1 and b) Input 2

The recognition of characters of the vehicle's license plates were separated for the image input and the video input. The outputs were properly tabulated for better viewing of results and comparisons.

C. *Preparation of Machine Learning Method*

KNN machine learning algorithm was settled on k = 1 or finding the single nearest neighbor producing optimal result because of the relatively small size of training data set. KNN algorithm is a simple algorithm which stores all available cases and classifies new cases [14].

The training was done by encoding a set of English alphabet and numbers with font similar to the font used in license plate, and saving this file in image or jpg format, a sample is shown in Figure 3. The characters on the number plate must have uniform fonts so that the OCR for number plate recognition would be less complex as compared to other methods [15]. OCR processes recognizes both handwritten and encoded/printed characters depending upon the quality of the subject (characters: letters and numbers) to be recognized [16]. The training set were encoded in six (6) batches including: four batches of *License Plate Font* from http://www.fontspace.com/dave-hansen/license-plate; one batch for *ARCADE R Font* from https://fonts2u.com/arcade-r.font and; one batch also for *GL NummernsChild Eng Font* from http://www.fontspace.com/gutenberg-labo/gl-nummernschild. The training of character recognition was done and saved as a first Visual Basic – Windows Forms Application GUI project for Visual Studio entitled: *License Plate Recognition – Character Training (KNN Algorithm*, as stated in section B, GUIs Form Layout.

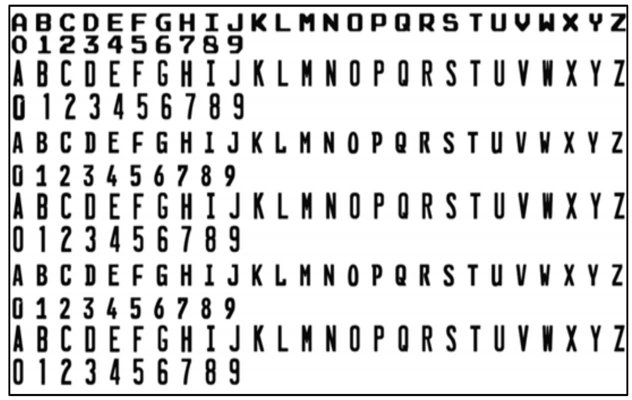


Fig. 3. Characters in Image Format

For testing and verification if the trained system can characterize letters and number, a set of sample text images shown Figure 4 was used as input to the second *Windows Forms Application project, License Plate Recognition – Testing the Trained System.* Since the characters were encoded, OCR was easily performed

Fig. 4. Set of Images of Characters

D. *Optical Character Recognition (OCR) and Image Processing*

The character recognition process is shown in Figure 5. Given the captured image of the vehicle, the system automatically searches for possible plate number region. This region undergoes the optical character recognition process shown in Figure 6. If the system recognizes characters, then system decides that the plate number area was found and it displays the recognized characters as the plate number of the vehicle in text format as an output. Otherwise, it keeps on searching for possible plate region until certain characters were recognized.

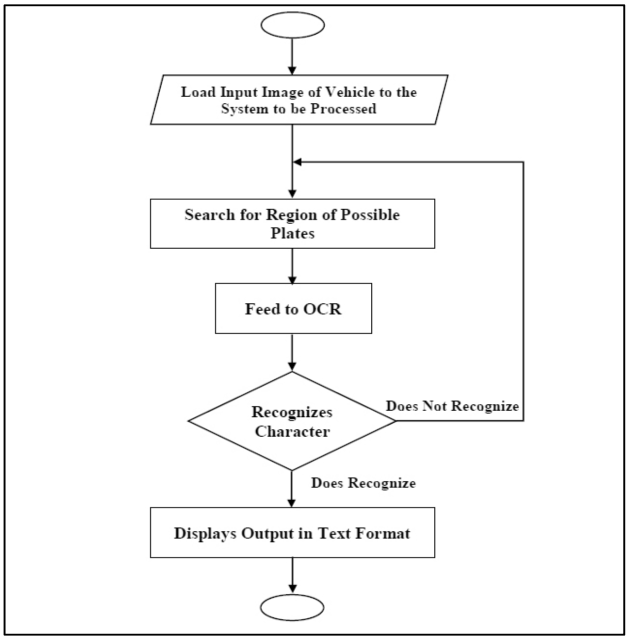


Fig. 5. Character Recognition System Flowchart (1)

The process of OCR involves the application of image processing techniques. Image processing is a method to perform some operations in an image, in order to get an enhanced image or extract some useful information from it [17]. In image processing, the images are converted as desired, for instance, to have a simpler processing of the image, an image is converted from red-green-blue (RGB) layers to gray scale layer [10]. In Figure 6, the image acquired need to undergo processing such as noise removal, image thresholding, edge detection segmentation, feature extraction and classification. The OCR cannot perform segmentation and classification without KNN training. An overview of KNN is shown in Figure 7. The output of the system are the characters composed of letters and numbers in text format.

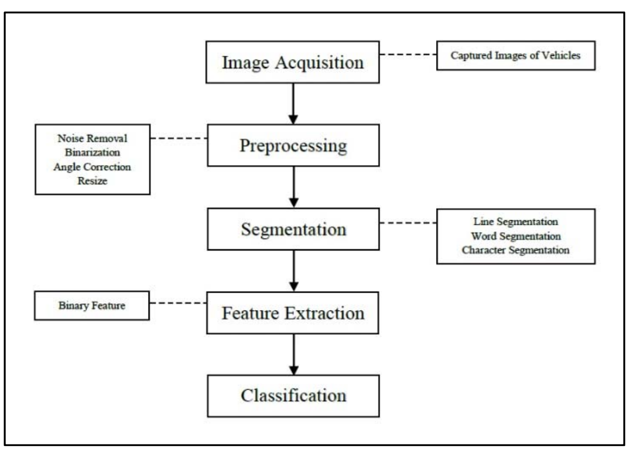


Fig. 6. OCR Processing

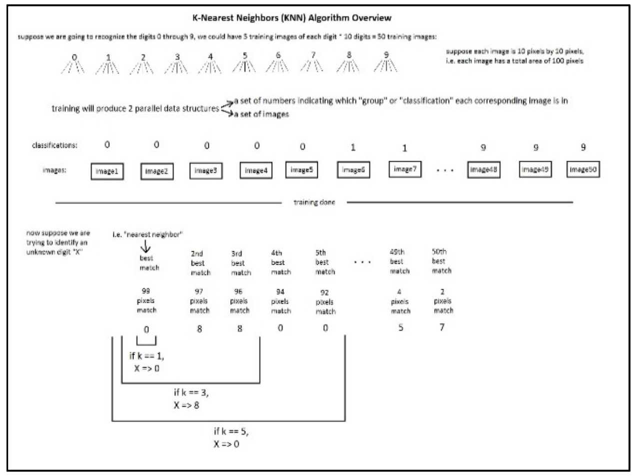


Fig. 7. KNN Overview [18] [19]

There were 5 class definition in Visual Basic for the image processing as described in Figure 8: *Preprocesssing*, *PossiblePlates*, *DetectPlates*, *PossibleCharacters*, and *DetectCharacters*. The concept of pre-processing method involves converting the original image to grayscale and finding its edges for easy detection of plate region. Finding possible license plate involves localizing possible plate numbers. Several image contouring was done until a possible plate number was located, this helps the system to easily detect and localize the plate number. In license plate detection, the license plate detected was extracted and cropped to be processed again for classifying or defining characters. The cropped region was processed again with techniques like gray scaling, thresholding, and several contouring until certain characters were detected. The segmented characters were localized and loaded to KNN trained system for recognition. The characters defined were converted to a text format and was displayed in the textbox of the form as license plate characters.

Preprocess	Possible Plates	Detect Plates	Possible Characters	Detect Characters
Original Image Grayscale Threshold	Find possible plate Contouring until detection	Detect/Localized plate region Extracted and cropped plate	Find possible characters Contouring until characters detected	Detect characters (Load KNN Training) Extract characters Text output

Fig. 8. Image Processing.

E. Video Input Processing

Video processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams [6]. The last *Visual Basic – Windows Forms Application* project made was the *License Plate Recognition – Video Clip of Moving Vehicle (Input)* which uses video as input. The video undergoes video processing such as *blob analysis*. The system creates a virtual line in the image frame that can be used to determine if a vehicle passes through. If the system detects the presence of a vehicle, it captures an image frame and subject that frame to image processing. The process is similar to the works of references [20], [8], and [21].

F. Evaluation of System's Performance

The performance of the system is evaluated using the following parameters: *plate number correctness* and *recognition accuracy.* The *plate number correctness* is expressed as the weighted score of the plate, $s_w(\mathbf{P}^R)$, defined as:

$$s_w(\mathbf{P}^R) = \frac{\left|\{P_i^R | P_i^R = P_i^C\}\right|}{\left|\{P_i^R\}\right|} \times 100\% = \frac{m}{n} \times 100\% \qquad (1)$$

Where $\mathbf{P}^R$ is plate recognized by a system, and $\mathbf{P}^C$ is the correct one, m = number of correctly recognized characters and n = total number of characters in a plate. For example, if the plate "ABC 1234" has been recognized as "ABC 1284", the weighted correctness score s_w is 6/7 or or 85.7%. The *system accuracy* A, was calculated from the equation

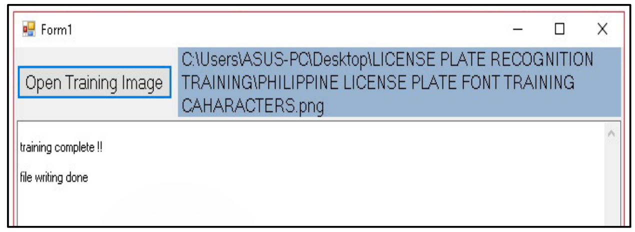
$$A = \frac{\sum weighted\ score\ of\ the\ license\ plate}{number\ of\ total\ plates} \qquad (2)$$

III. Result & Discussion

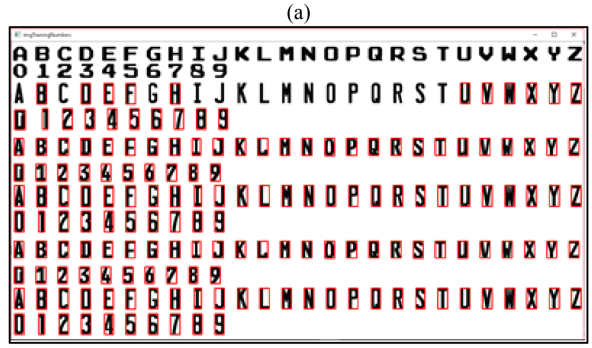
Results of the images that underwent image processing are shown in the following subsections.

A. Character Recognition Training and Preliminary Testing

OCR were performed on typed-text input in .png format as a valid input with the letters of the English alphabet (capital/uppercase letters only) and digits 0 – 9 as classes of characters [22]. Figure 9(a) shows the interface that opens the image file used for training. During training, the letters in the image were selected and was surrounded by a red rectangular shaped outline one by one as the user inputs the corresponding letter as shown in Figure 9(b). When the selection of all letters were finished, the system notifies that the training was completed as shown in Figure 10. A sample of the result of the tests after the training was completed is shown in Figure 11(a) and (b), the system recognized all the characters correctly.



(a)



(b)

Fig. 9. Character identification

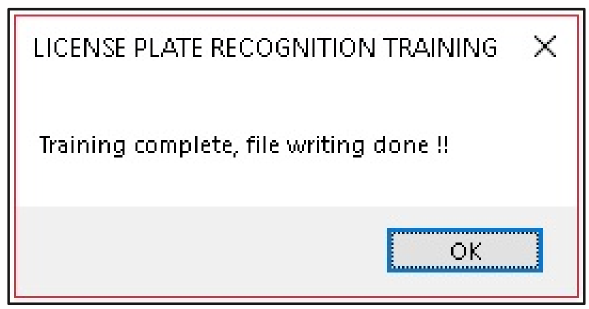
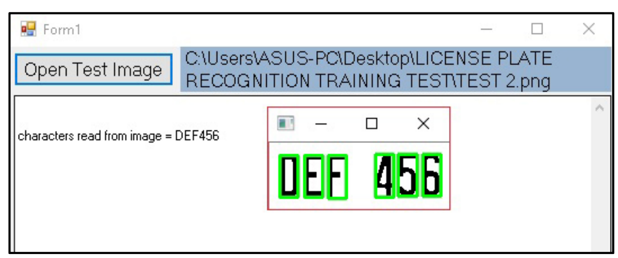
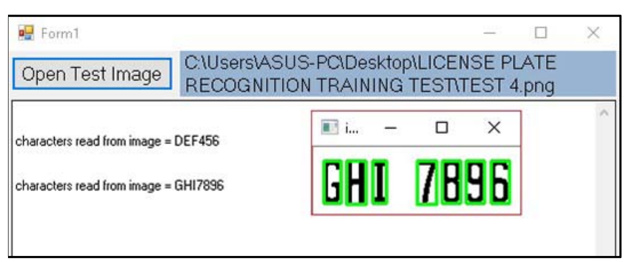


Fig. 10. Notification that the training was completed and that the file writing was done.



(a)



(b)

Fig. 11. Testing the System

B. *Image Processing of License Plate*

The figures shown in Figure 12 are sample of the system's main interface with an image input file was loaded. The system detects the plate region, the one enclosed by a red box, the segmented image in the lower right corner, and above it is the recognized characters of the license plate in text format. The system saved the results including the cropped licensed plate region and its gray scaled output. Table 1 shows sample of the saved results: original cropped license plate region, result of thresholding, result of segmentation, recognized characters in text format. The characters in red font color indicates an incorrect character recognition. Note that the table also shows the results for all three different positions.

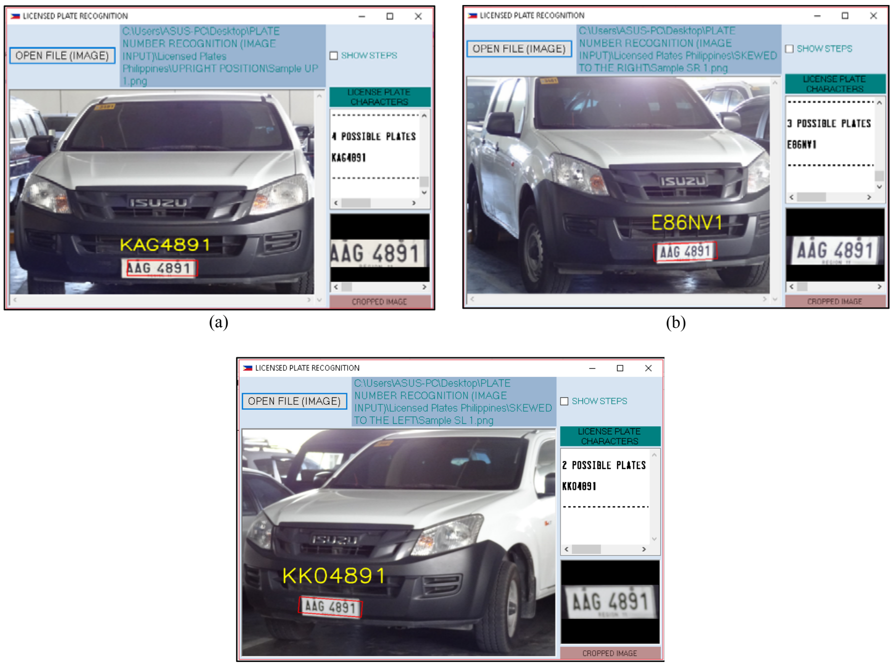
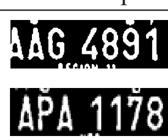
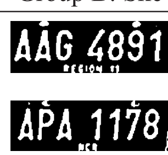
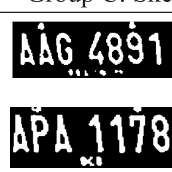
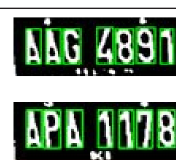


Fig. 12. License Plate Recognition, a) Upright Position; b) Skewed to Right Position; c) Skewed to Left Position

TABLE 1
IMAGE SAMPLE OF CROPPED LICENCE PLATE REGION

Original image (input)	Gray scaling/Tresholding	Character Segmentation	Text output
Group A: Uright position			
			KAG4891 APAJJ78
Group B: Skewed to right position			
			E86NV1 APA117O
Group C: Skewed to left position			
			KK0489 APA1198

C. License Plate Recognition for Image Input

D. The results of the license plate recognition tests for the three groups of images are tabulated in Table II, II and IV. Each row shows the input, output, the number of characters correctly recognize, and the weighted score computed using Equation1. A character indicated in red color indicates an incorrect character recognition. The accuracy of the system for each group of images was calculated using the Equation 2 and they are summarized in Table V.

TABLE 2
LICENSE PLATE WEIGHTED SCORE FOR GROUP A: UPRIGHT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	KAG4891	6	85.71%
2	APA 1178	APAJJ78	5	71.43%
3	ABO 6206	AB062O6	5	71.43%
4	ABH 7344	ABH7344	7	100.00%
5	AOA 2299	A0A2299	6	85.71%
6	ABC 5018	ABC5O18	7	100.00%
7	ADO 9171	ADO9171	5	71.43%
8	AKA 5591	AKA5591	7	100.00%
9	AOA 1583	AOA1583	7	100.00%
10	AAS 4083	AAS4O83	7	100.00%
11	ACU 9594	ACU9594	7	100.00%
12	ABA 8101	ADA81OJ	4	57.14%
13	ADP 4510	AOP45O	5	71.43%
14	AOA 2665	A0A2665	6	85.71%
15	ADP 4557	AOP4667	4	57.14%
16	A0A 2665	A0A2665	6	85.71%
17	AHA 1624	AHA1624	7	100.00%
18	AOA 1875	A0A1B75	5	71.43%
19	AOA 2092	A0A2O92	5	71.43%
20	AAG 5979	AA05979	6	85.71%
21	ABT 3242	ADT3242	6	85.71%
22	A0A 1879	A0A1079	5	71.43%
			Average Accuracy	83.12%

Table 3
License Plate Weighted Score for Group B: Skewed to Right Position

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	E86NV1	1	14.29%
2	APA 1178	APA117O	6	85.71%
3	ABO 6206	AD06206	5	71.43%
4	ABH 7344	ASW93AA	1	14.29%
5	AOA 2299	A0324--	1	14.29%
6	ABC 5018	A-C5018	6	85.71%
7	ADO 9171	Y1SOS171	3	42.86%
8	AKA 5591	-JA5591	4	57.14%
9	AOA 1583	A0A1583	7	100.00%
10	AAS 4083	JJS4O83	4	57.14%
11	ACU 9594	ACU97--	4	57.14%
12	ABA 8101	ADA81O1	5	71.43%
13	ADP 4510	JOP45-O	5	71.43%
14	AOA 2665	K0A26-5	5	71.43%
15	ADP 4557	L116A--	0	00.00%
16	A0A 2665	74Z	0	00.00%
17	AHA 1624	--	0	00.00%
18	AOA 1875	1ZY4277	1	14.29%
19	AOA 2092	--	0	00.00%
20	AAG 5979	--	0	00.00%
21	ABT 3242	--	0	00.00%
22	A0A 1879	--9A97-	2	28.57%
			Average Accuracy	38.97%

Table 4
License Plate Weighted Score for Group C: Skewed to Left Position

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	KK04891	4	57.14%
2	APA 1178	APA1198	6	85.71%
3	ABO 6206	AD06206	5	71.43%
4	ABH 7344	ABH7344	7	100.00%
5	AOA 2299	A0A22S9	5	71.43%
6	ABC 5018	A8C50J8	5	71.43%
7	ADO 9171	ABD9171	7	100.00%
8	AKA 5591	AKA5591	7	100.00%
9	AOA 1583	A0A1583	6	85.71%
10	AAS 4083	LLS4083	5	71.43%
11	ACU 9594	ALIAGA2	1	14.29%
12	ABA 8101	A0A2454	6	85.71%
13	ADP 4510	--	0	00.00%
14	AOA 2665	--	0	00.00%
15	ADP 4557	--	0	00.00%
16	A0A 2665	--	0	00.00%
17	AHA 1624	--	0	00.00%
18	AOA 1875	--	0	00.00%
19	AOA 2092	--	0	00.00%
20	AAG 5979	A3D---3	3	42.86%
21	ABT 3242	KDT3242	5	71.43%
22	A0A 1879	--	0	00.00%
			Average Accuracy	46.21%

Table 5
System Accuracy of Three Groups

Group	Total number of detected plates	Average Weighted score/ Group
A	22	83.12%
B	19	38.97%
C	19	46.21%

It can be observed in Table II that the system detected the plate numbers of all input images where vehicle are captured in the upright position. The system recognition accuracy is 83.12 %. In the case of the skewed vehicle position as shown in Table IV and V, there were cases that the system failed to detect the plate number area such as plate nos. 17, 19, 20, and 2 of Table IV; and plate nos. 13 to 19, and 22 in Table IV. The system recognition accuracy is 38.97 % for *skewed to right* position, and 46.21 % for *skewed to left* position. This is because the plate numbers of the vehicle in SL or SR position were not clear enough. Incorrect recognition also occurred when the plate number was not properly cropped such as the case in the first entry of Table I, portion of the first character was slightly cropped, the system recognized letter *K* instead of letter *A*. Another observation in Table II, the system failed to distinguished slightly similar characters such as: character *J* was misidentified as *I* in plate no. 2, character *O* was misidentified as *0* and vice versa in plate no.3, and character *B* was misidentified as *D* in plate no. 7. Same goes with plate nos. 12 to 22.

A number of considerable errors were also observed as shown in Table III and IV. In Table III, the characters *AAG 489* in plate no. 1 were incorrectly identified as *E86NV*, and the characters *ABH7344 in plate no. 4* were incorrectly identified as *ASW93AA*. In Table IV, the characters *ACU9594* were incorrectly identified as *ALIAGA2*. Due to these errors, the system recognition accuracy for SR and SL group of images is very low, 38.97% and 46.21%, respectively.

D. License Plate Recognition for Video Input

The two video clips fed to the system are shown is shown in Figure 13. The system captured frame and processed it repeatedly, and displayed the output of the system while playing the input video. It was observed many times that the output is incorrect. The procedure slowed down the whole process of license plate character recognition. However, after several attempts of locating plate number, the system finally locate the plate number area, segment, and recognize the characters correctly. In addition, the system successfully displayed the correct plate number in text format.

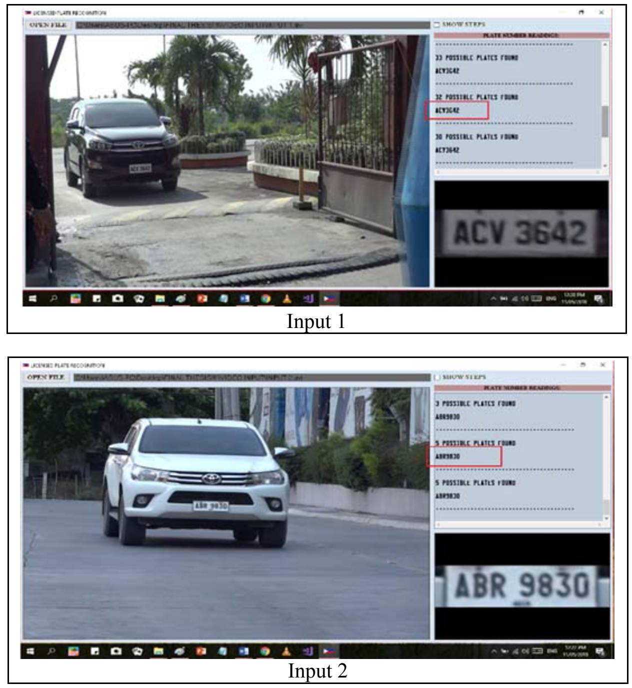


Fig. 13. Loading Video Input

V. Conclusion & Recommendations

The designed automatic license plate recognition system was successfully implemented in Visual Basic using EmguCV image processing library and KNN machine learning algorithm. It was able to recognized characters in the license plate using image or video input. The tests using two video inputs with one vehicle each was successful. The system was able to locate and recognize all the characters in the license plate accurately as displayed in the output. The system's performance using image inputs in three groups of images: UP, SR, and SL, is 83.12 %, 38. 97%, and 46.21%, respectively. The best performance is when the vehicle is in the upright position in the captured image that is 83.12 %. Incorrect recognition in the UP position is due the similarity of the characters such as O and 0, J and I, B and D. Considerable errors in SL and SR positions are due mainly to the poor quality of the captured image of the vehicle.

For the improvement of the system, it recommended to use exact font style of license plate in training the system, and capture samples images at different angle positions and lighting conditions. Obtain samples as many as possible. Also, the system can be improved by adding more features to the interface like time stamp or database and work for the methods to make the detection real-time.

References

[1] V. Gigante and S. B. Jolcanain, "Development of a Motorcycle License Plate Number Recognition System," pp. 1, 13, March 2014.

[2] A. Tamayo, "Occurrence of Traffic Accidents in the Philippines: An Application of Poisson Regression Analysis," *SSRN Electronic Journal,* p. 10.2139/ssrn.1438478, July 2009.

[3] J. Dr. Balcanao, "Would you say that accidents on Philippine roads are caused mostly by errant drivers?," Phil Star Global, Benguet, 2011.

[4] E. Tipan, "DOTC, LTO to impose stiffer fines for traffic violations," AUTO INDUSTRY NEWS, Quezon, 2014.

[5] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. Gan Lim, A. D. Fillone and E. P. Dadios, "Vehicle Detection and Tracking using Corner Feature Points and Artificial Neural Networks for a Vission Based Contactless Apprehension System," *Computing Conference 2017,* p. 688, 2017.

[6] "Video processing," 1 July 2017. [Online]. Available: https://en.wikipedia.org/wiki/Video_processing.

[7] R. K. Billones, A. Bandala, E. Sybingco, L. Gan Lim and E. P. Dadios, "Intelligent System Architecture for a Vision-Based Contactless Apprehension for Traffic Violations," p. 1, 2017.

[8] A. Sobral, "simple_vehicle_counting," 12 04 2017. [Online]. Available: https://github.com/andrewssobral/simple_vehicle_counting.

[9] License Plate Recognition, "How License Plates Recognition Works," 2010. [Online]. Available: www.licenseplatesrecognition.com/how-lpr-works. [Accessed May 2018].

[10] S. Bold and B. Sosorbaram, "Smart License Plate Recognition Using Optical Character Recognition Based on," *International Journal on Recent and Innovation Trends in Computing and Communication,* pp. 92-96, 2017.

[11] V. Ong and D. Suhartono, "USING K-NEAREST NEIGHBOR IN OPTICAL CHARACTER RECOGNITION," *ComTech Vol. 7 No. 1,* pp. 53-65, 2016.

[12] S. Shi, Emgu CV Essentials, Birmingham B3 2PB, UK: PACKT Publishing Ltd., 2013.

[13] S. Saxena and M. S. Tiwari, "Starting with Emgu CV," 19 September 2013. [Online]. Available: https://social.technet.microsoft.com/wiki/contents/articles/15385.starting-with-emgu-cv.aspx.

[14] S. Dr. Sayad, "K Nearest Neighbor," 2010. [Online]. Available: http://chem-eng.utoronto.ca/~datamining/.

[15] K. Kaur and V. Banga, "NUMBER PLATE RECOGNITION USING OCR TECHNIQUE," *IJRET: International Journal of Research in Engineering and Technology,* pp. 286-290, 2013.

[16] L. Eikvil, OCR Optical Character Recognition, P.B. 114 Blindern: Norsk Regnesentral, 1993.

[17] G. Anbarjafari, "Digital Image Processing," 2014. [Online]. Available: https://sisu.ut.ee/imageprocessing/book/1.

[18] C. Dahms, "OpenCV_3_KNN_Character_Recognition_Emgu_CV_3_Visual_Basic," 9 December 2015. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_KNN_Character_Recognition_Emgu_CV_3_Visual_Basic/blob/master/DocsAndPresentation/KNN_Overview.png.

[19] C. Dahms, "OpenCV_3_License_Plate_Recognition_Emgu_CV_3_Visual_Basic," 20 July 2017. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_License_Plate_Recognition_Emgu_CV_3_Visual_Basic.

[20] C. Dahms, "OpenCV_3_Car_Counting_Visual_Basic," 20 February 2016. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_Car_Counting_Visual_Basic.

[21] E. P. Dr. Dadios, A. Dr. Filllone, E. Engr. Sybingco, L. G. Dr. Lim and A. Dr. Bandala, "Contactless Apprehension of Traffic Violators on 24-hours Basis All-Vehicle Detection System (CATCH-ALL)," p. 18, 2015.

[22] J. Nguyen, "Introduction to Optical Character," p. 9, 2014.