Automated License Plate Recognition of Philippine License Plates

Myra Josephine S. Villanueva*
Joel P. Ilao**
Computer Technology Department
College of Computer Studies
De La Salle University – Manila
myra_villanueva_3@yahoo.com*
ilaoj@dlsu.edu.ph**

Chloe Michelle K. Cervania
Vincent Spencer Y. Ku
Desmond Carvey T. Ragos Ty
Computer Technology Department
College of Computer Studies
De La Salle University - Manila

Abstract

Automated License Plate Identifier (ALDEN) is a License Plate Recognition (LPR) System developed to recognize images of Philippine license plates. ALDEN captures images of an approaching vehicle when a sufficient distance from the camera is obtained, localizes the license plate from the scene and reads its content. This system is designed to be robust against changes in illuminations, and to a limited degree, correct perspective distortions in the acquired images, while still maintaining real-time performances.

This paper describes the design and implementation of the ALDEN. Pre-processing techniques are first applied to the acquired raw images to correct uneven illumination, and perspective distortion. The license plate is then extracted from the visual scene, binarized, and segmented into characters using knowledge of the size and locations of characters in a license plate area using the Philippine License Plate format. Character Identification is then performed by segmenting the characters into sixteen regions, and comparing the regions with character templates using correlation as a similarity measure. Performance tests using this technique tested on 300 character images yield an average recognition rate of 93.75 percent.

1 Introduction

License Plate Recognition (LPR) systems are used for automated vehicle management and identification, such as entrance admission in highways and establishments, security, parking control, and road traffic control. A number of commercial softwares have already been developed for this purpose[4][5]. However, these softwares are developed for a particular license plate format, and therefore, incapable of accomodating different styles and formats such as license plates used in the Philippines [3].

In Philippine commercial parking lots, for example, vehicle registration is still done manually: the parking attendant encodes the license plates of an incoming vehicle and gives the parking ticket to the driver. This method, however, is prone to error caused by human fatigue or carelessness. Automatic automobile management through LPR can give a more consistent performance, and savings through reduction of labor costs to the company.

There is one LPR system that has already been developed for Philippine license plate recognition, in the College of Computer Studies, De La Salle University – Manila. Cue, et. al [1] used fuzzy techniques in thresholding, and statistical methods for the character recognition. The average accuracy rate of character recognition for the their study is 86.67%. Inaccuracies can be attributed to the system's inability to recognize skewed images, images taken outside the recommended proximity range, and license plates not having the recommended Philippine license format. Additionally, these systems lack pre-processing techniques that would make the recognition invariant to changes in the environment, such as, illumination caused by weather changes. Thus the recognition rate may be further improved by improving the image quality such that images become more compatible to the recognition technique. A whole system that will handle the acquisition of the image of a vehicle with the desired license plate, addressing the aforementioned issues and limitations, was developed.

In other countries, such as in Saudi Arabia and Korea, several studies and approaches are being done to develop an LPR that are used in several applications such as those mentioned above. These applications are suited to their own country's format. There are also a number of commercial LPR systems available, such as SeeCar, Perceptrics, and Pearpoint.[4][5]

2 ALDEN

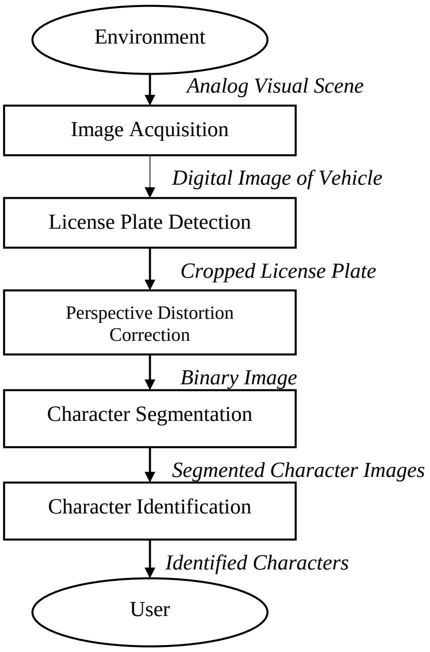


Figure 1. System Block Diagram

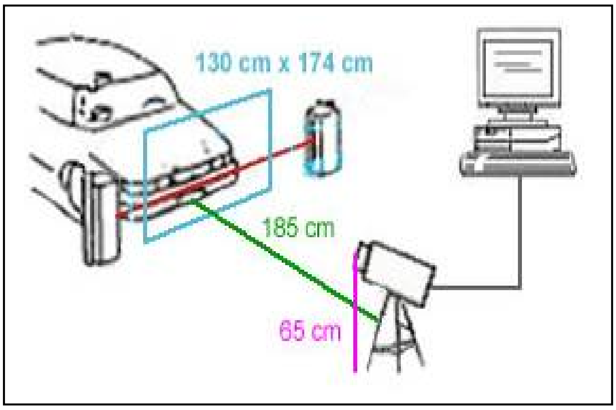


Figure 2. System Setup

Figure 1 shows the block diagram of the ALDEN system, while Figure 2 shows its system setup. When ALDEN detects the presence of a vehicle through a laser trip sensor, a web camera captures a digital image of the vehicle's front. Image enhancements are next applied to the captured image for more effective processing. After the license plate has been localized, perspective distortions in the license plate are then corrected. The license plate image is then converted to a binary image, before being segmented into individual characters. Finally, each segmented character is identified through correlation and template matching.

Figure 3 shows the currently used two formats for Philippine license plates with their corresponding measurements. Noticeable differences in plate background, foreground and background color, font style, font size and character spacing are evident between the two formats of license plates.

Figure 3. Philippine License Plate Formats

2 License Plate Detection Module

In preparation for license plate localization, the raw images undergo an image enhancement process, which aim to correct uneven illumination and enhance the image contrast. Figure 4 shows the flowchart for the image enhancement submodule, and actual visual results as indicated in Figure 5.

Adaptive histogram equalization is applied to the grayscale image of the captured image, seen as Figures 5a and 5b, respectively, and to the mean values of the red and green intensities of the captured image. Then, intersection of the binarized resulting images of the adaptive histogram equalization is done. This serves as the mask to select objects of interest, seen as Figure 5c.

Filtering was also done to correct uneven illumination. The result of top-hat filtering, Figure 5d, with a square shaped structuring element is then contrast adjusted by three different levels, namely 0.3, 0.5 and 0.7. The image is then binarized resulting to an image similar to Figure 5e. Objects of interest in the resulting binary image are slected using the earlier prepared mask. Pixels from this image must overlap the masking image. This process results to an image which looks like Figure 5f.

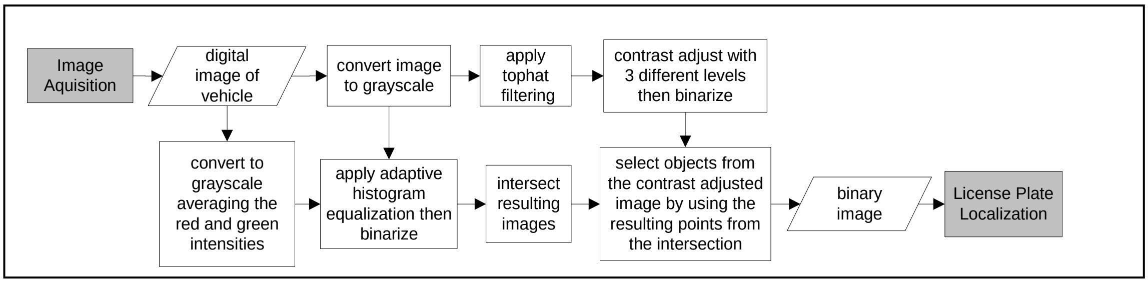


Figure 4. Image Enhancement Submodule Flowchart

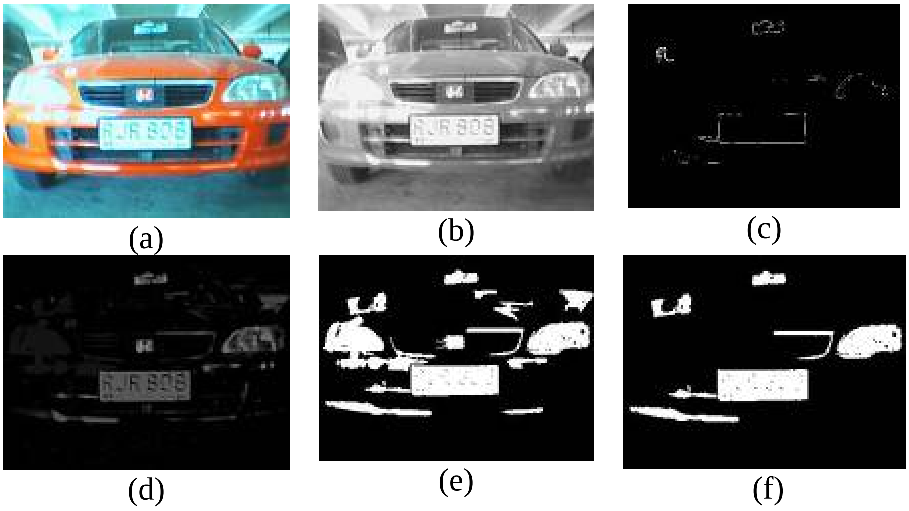


Figure 5. Image Enhancement Submodule Process Images

Adaptive histogram equalization is used as an alternative for histogram equalization for it gives better results. The adaptive form operates on small regions of an image, called tiles. Each tile's contrast is enhanced, so that the histogram of the output region approximately matches a flat histogram. Top hat filtering, on the other hand, is a nonlinear operator that finds the brightest or darkest pixels in two different size neighborhoods, and then keeps the extreme values. It locates features that match the size and shape of the neighborhoods, either as features of interest or as noise to be eliminated.

The image contrast is adjusted to improve the visibility of the results; this is done using three different levels. Using one level of contrast adjustment, significant edges on images are sometimes not detected due to varying levels of brightness.

Figure 6 and Figure 7 show the flowchart of the license plate localization submodule, and the corresponding visual results when applied to a specific image, respectively.

A binary image, as in Figure 7a, undergoes perimeter filtering. The perimeter of the objects seen as Figure 7b is obtained by setting the pixel to 0 if its 4-connected neighbors are all 1's. The vertical and horizontal lines of the objects are isolated in two different images. The vertical and horizontal edges are isolated by creating a vertical and horizontal line structuring elements. Lines that are neither vertical nor horizontal are discarded. Vertical and horizontal lines are defined to be at least two and three connected pixels, respectively. The resulting lines are elongated to compensate possible incomplete lengths due to uneven lighting conditions. The horizontal and vertical lines are seen in Figure 7c and Figure 7d, respectively.

Figure 7e shows the intersection of the resulting lines which produces several quadrangles. Quadrangles are detected by checking whether two horizontal and two vertical lines intersect in four points. The quadrangle is labeled as a candidate license plate if its proportions are close to the defined proportions of a license plate with respect to the image resolution.

If there are more than one candidate plates, the quadrangle with the closest ratio of proportions to the defined one is considered to be the plate. Figure 6f shows the region where the plate is located. The points

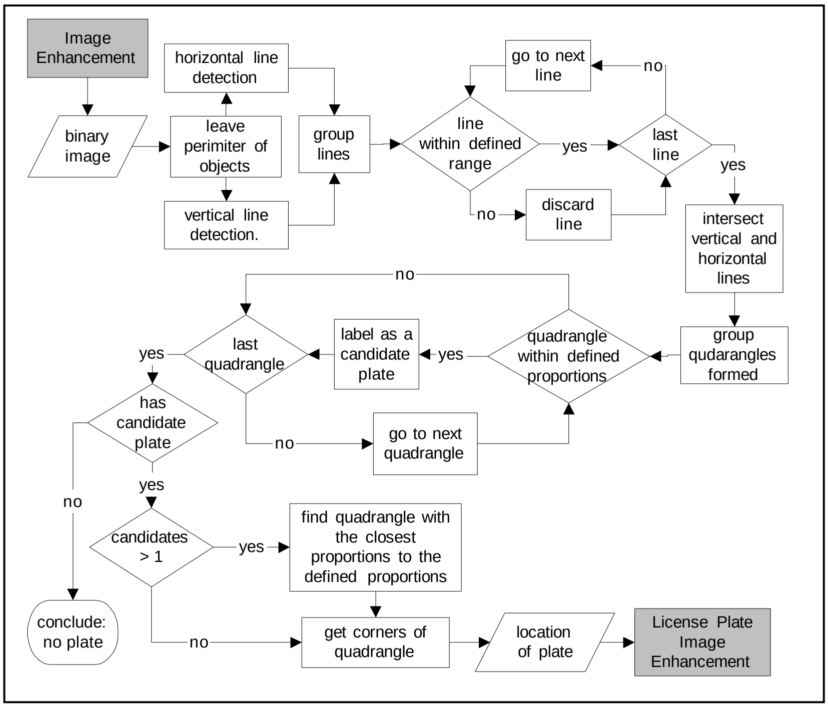


Figure 6. Flowchart of Implemented License Plate Localization Submodule

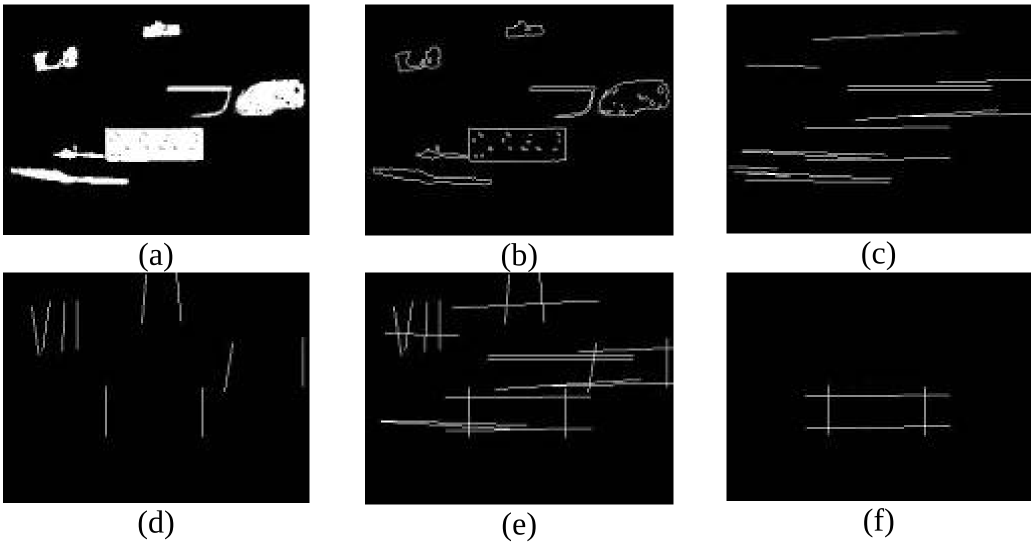


Figure 7. Process Images License Plate Localization Submodule

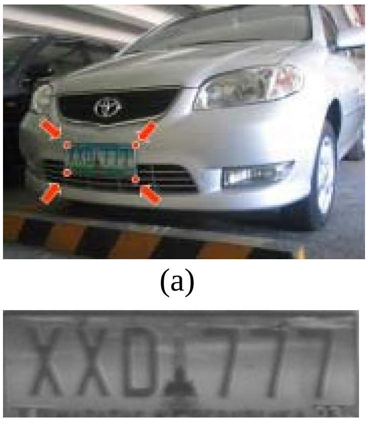
of the four corners of the detected plate are then passed for plate enhancement.

3 Perspective Distortion Correction Module

The input image is first converted to grayscale. With known corner points of the license plate from the previous module, the image of the license plate is now segmented. An application of a 2-D spatial transformation is done to the extract the plate region.

The destinations of each corner points $f(x,y)$ in the output image are specified. These are used as the control points for mapping each pixel of the input image to a new point $g(x',y')$ in the output image. The new pixel value of the output image is determined by using bi-cubic interpolation. This type of interpolation preserves the fine details in the input image. The value for an interpolated pixel is the weighted average value of its nearest *4x4* neighborhoods. The *4x4* neighborhoods can be acquired by extending one pixel to the left and above and two pixels to the right and below the central pixel. The weight of each neighborhood pixel is based on its distance to the interpolated pixel. Once the destinations of all the pixels are calculated, it is remapped to form the output image.

After the bi-cubic interpolation, the image of the license plate is aligned and proportioned. The given destinations of each corner points are accountable for the proportionality of the license plate. Figure 8 shows an example of an image that has undergone perspective distortion correction.



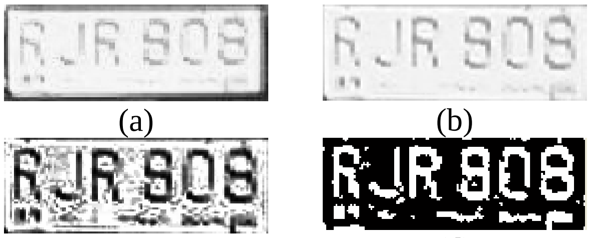
(a)

(b)

Figure 8. Perspective Distortion Correction

4 Character Segmentation Module

Image enhancements are again, applied to the resulting image, prior to binarization and segmentation of license plate images. A sharpening mask is then used to filter the input image, followed by a 3x3 median filter in order to minimize image noise. Improving the image’s contrast is better done with histogram equalization operating on small regions of the image. Rather than operating on the whole image, each 8x8 region of the image is enhanced to have a flat histogram. As a final step a simple thresholding process is applied to the image resulting to a binary image with character objects as logic 1 and the background as logic 0. Thresholding is done by locating regional maxima in the image. This final step was able to experimentally segment the characters of the different types of license plate images under varying illuminations. Figure 9 shows an input image (9a), sharpened image (9b), contrast-enhanced image (9c), and the final image (9d) respectively.



(a) (b)

(c) (d)

Figure 9. Input image (a) undergoing sharpening (b), contrast enhancement (c), and binarization (d)

The resulting binary images are then segmented into individual characters, using knowledge of the standard sizes of the characters relative to the area of the whole license plate, and their approximate positions within the license plate. Blobs not corresponding to license plate characters are removed through horizontal and vertical segmentation, respectively. Segmenting the printed characters on the plate can be implemented by statically removing a constant number of rows from the top and bottom parts of the image. This is computationally inexpensive. However, it may result to inaccurate and inefficient segmentation of desired regions of the plate.

Horizontal segmentation can be done by counting the number of white pixels of every row, given the characters are white in color in the given binary image. Rows with few or minimal pixel counts signifying a non-character row are neglected. This process leaves a binary image horizontally segmented from any unimportant vertical regions, like the margins, stickers, and other plate labels. Figure 10 shows a sample image that has undergone horizontal segmentation.

Figure 10. Horizontal Segmentation Input Image (left) and the horizontally segmented Image (right)

After the horizontal segmentation process, the characters from the resulting image are then extracted by statically segmenting the plate into six overlapping regions, where characters are assumed to be found. Labeling of connected components are then applied, and the six largest labelled regions are selected as the six characters of the plate. Figure 11 shows some results of the character segmentation process.

The resulting six objects however are likely to be imperfectly segmented seen Figure 11c. The length of each of these six bounding boxes is checked if it is within a correct range signifying the character's length. If it is too long or too short, that the located character object is attached to another object or the located character object is just a portion of it, respectively, bounding box's length is set to the value of the computed value acquired earlier. This procedure is also done for each width of the six bounding boxes. As an additional parameter, the distances between characters are also used as basis for the widths' correctness. This parameter is suitable for determining the correctness since the distances between characters are constant for all plates of the same type.

After acquiring the locations of the correct bounding boxes that exactly contain the character objects, the input binary image is conclusively segmented into six character images, as in Figure 11d. All pixels within a rectangle are grouped and the three largest groups are selected and are considered parts of a character.

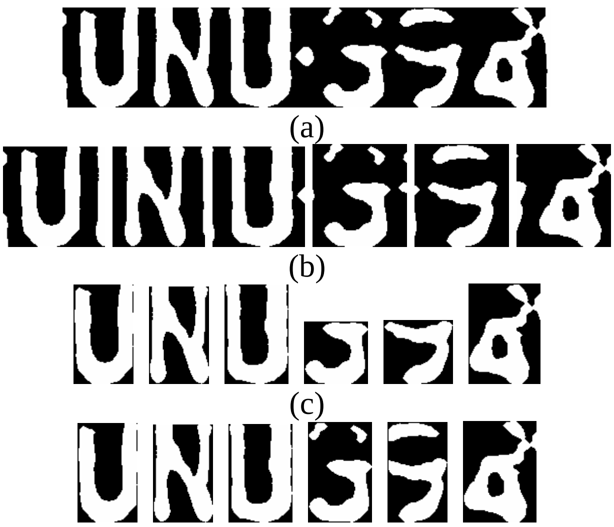


Figure 11. Character Segmentation Results. Horizontally-segmented input image (a), Statically segmented image using approximate locations of the six characters (b), Six largest labelled regions from the six character images (c), final segmented image (d)

5 Character Identification Module

The Character Identification module uses template matching to identify the characters of the license plate. Binary image templates of the characters 'A' to 'Z' and '0' to '9' are created for this purpose. A total number of forty-two character templates are used to account for perceptual differences in the old and new license plate formats for some characters. The character image of the template and the input image from the segmentation module are first cut into different regions. Correlation is then computed for all the regions of the input image and the character template. The sixteen regions are regions of the character where the differences in between characters are significantly found. These regions are shown in Figure 12.

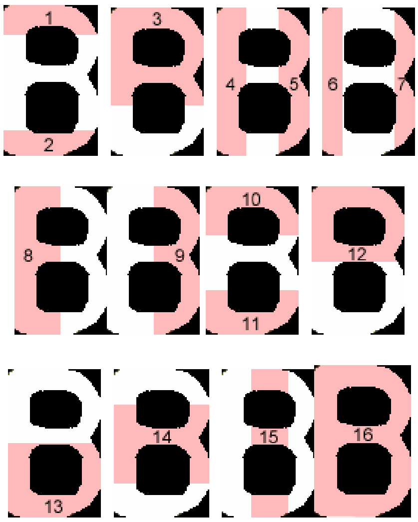


Figure 12. Regions used for Character Identification

With correlation, the emphasis is on the degree to which a linear model describes the relationship between two variables. Correlation computes the correlation coefficient using the Pearson's Correlation which is:

$$corr(i,j) = \frac{\sum\left(R_{u,i} - \overline{R_i}\right)\left(R_{u,j} - \overline{R_,}\right)}{\sqrt{\sum\left(R_{i,j} - \overline{R_,}\right)^2}\sqrt{\sum R_{u,j} - \overline{R_j}^2}}$$

where: $\overline{R_i}$ is the mean of the elements of $R_{u,i}$, and $\overline{R_j}$ is the mean of the elements of $R_{u,j}$. $R_{u,i}$ is an element of the matrix of the input image. $R_{u,j}$ is an element of the character template being compared to the input image.

Pearson’s Correlation reflects the degree of linear relationship between the input image given by the Character Segmentation Module and the character templates. Its result value is between -1 and +1. A value of +1 signifies that there is a perfect positive linear relationship between the input character and the character template. A value of -1 means that there is a perfect negative linear relationship between the two images being compared, and a value of 0 means there is no linear relationship at all.

Using a correlation-based matching technique, two images are considered not similar if one or more corresponding regions are found to be very dissimilar. Hence, in the Character Identification module, the degree is similarity between a character image and template is computed using the following formula:

$$S = \sum_{i=1}^{16} \left(C_{\max} - C_i \right) \times C_{\max}$$

where S is the measure of similarity, C_i is the correlation coefficient for the i^{th} region, and $C_{\max}$ is the maximum computed regional correlation coefficient for the given pair of images. The character template with the lowest sum is assumed to be the character identified. Character identification is performed six times for the six characters of the Philippine license plate.

Oftentimes, character images are not complete due to pre-processing and/or scratches and occlusions to the character prints. Even though there are problems that hinder the system to identify the characters correctly, the correctness of character identification process resulted to the average percentage of 93.75%.

6 Testing the Correctness of Character Identification

For implementation and testing, an A4 Tech View Cam Pro model number PK-635 USB web camera, operating on a 320x240 resolution, was used. The camera features a CMOS 350K pixels image sensor, a lens specification of 54º view angle, a capture resolution maximum of 640x480.

Correctness of the character identification module given a binary license plate image is tested by initially segmenting the image into six character sub-images manually. After the image segmentation, the character images are fed into the Character Identification module. The results are tallied to determine its performance. The test images of private vehicles’ fronts were taken from those parked in covered and uncovered university parking lots, under different times of the day and at different perspective angles. Fig 13 shows some of the test images taken.

Fig 13. Sample Images of Vehicles used for testing

7 Results and Analysis

The Character Identification Module was able to correctly identify 286 character images out of 300 from the older plate format (95.33%), and 164 character images out of 180 from the newer plate format (91.11%).

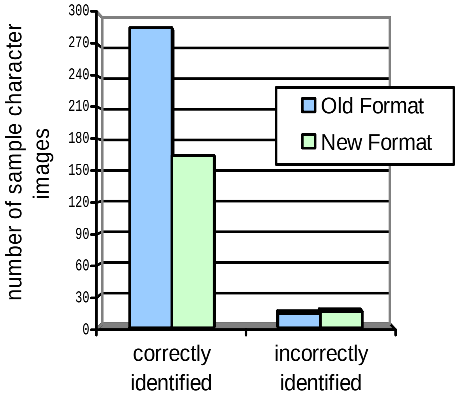


Fig 14. Results of Test for the Correctness of Character Identification

The figures shown below are some examples processed by the Character Identification Module.

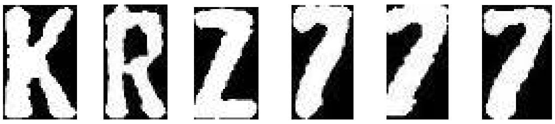


Fig 15. Character Identification Sample- High rate of Identification

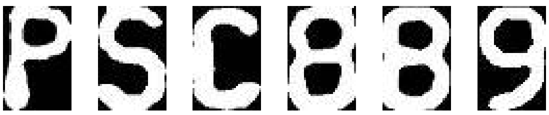


Fig 16. Character Identification Sample- High rate of Identification

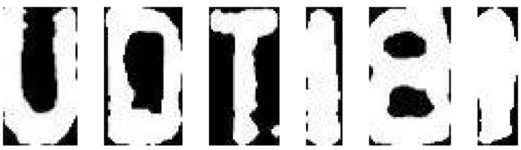


Fig 17. Character Identification Sample- Moderate rate of Identification

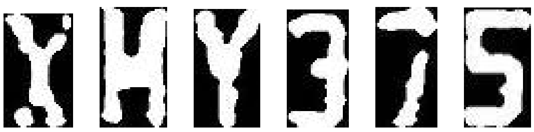


Fig 18. Character Identification Sample- Moderate rate of Identification

Table 1 Character Identification Sample Results

license plate	1	2	3	4	5	6	# of correctly identified	% correctness
KRZ777	K	R	Z	7	7	7	6	100%
PSC889	P	S	C	8	8	9	6	100%
UDT181	U	Q	T	1	8	1	5	83.33%
XMY375	X	H	Y	3	7	5	5	83.33%

Inaccuracies in the character recognition process can be attributed to the limitations of the pre-processing, license plate detection, and character segmentation blocks. Some conditions where LPR system fails to recognize are extreme darkness or brightness conditions, reflections on the license plate, image blur and partial occlusions. These conditions cause the less than ideal inputs to the Character Identification module, resulting to errors. For example, in Fig 17, the second letter has a curved feature in its bottom right corner and was mistaken for as letter ‘Q.’ The pre-processing technique has not been successful in aligning the license plate image. The second letter in Fig 18 was identified incorrectly because the middle part of letter M is missing and therefore looked like letter ‘H.’

8 Conclusions and Recommendations

In this paper, we have presented a method for performing character recognition of Philippine license plates, through ALDEN, a locally-developed LPR system. Acquired digital images undergo pre-processing and image enhancements, while the license plate is being extracted from the visual scene and segmented into different characters. The character identification module is implemented by using the correlation function. Cutting the image into several regions wherein distinct character features are evident was an important step for the identification process. Tests were also done to make sure that the selected regions would show the character’s uniqueness. Each of these selected regions was correlated to different templates. The one with the closest matches is revealed to be the character that resembles the image the most. An average correctness of 93.22% was computed for identifying the characters.

For better character identification results, the region of each template could be weighted. Weighted regions could better differentiate the uniqueness of each template’s region. Applying this gives the system a better probability of differentiating characters with similar features.

References

[1] Cue, A. Y., Haw, B. S., Lao, G. T., & Sy, T. J. L. (1998, August 3). *Plate Number Scanner*. Archives, College of Computer Studies, De La Salle University.

[2] Banares, C. J. O., Lee, B. T., & Ricerra, J. C. (2002, April 26). *Zip Code Identification, Verification and Association*. Archives, College of Computer Studies, De La Salle University.

[3] Koval, V., Turchenko, V., Kochan, V., Sachenko, A., & Markowsky, G. (2003, September 8-10). *Smart license plate recognition system based on image processing using neural network*. Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings of the Second IEEE International Workshop 123 - 127

[4] Sarfraz, M.; Ahmed, M.J. & Ghazi, S.A.. (2003, July 16-18) *Saudi Arabian license plate recognition system*. Geometric Modeling and Graphics 2003. 36 – 41

[5]Yu, M. & Kim, Y. D. (2000). *An approach to Korean license plate recognition based on vertical edge matching.*