Serdica Journal of Computing 16(1), 2022, pp. 1-23, 10.55630/sjc.2022.16.1-23
Published by Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

Image Processing in Automatic License Plate Recognition Using Combined Methods

Nabila Hamdoun*, Driss Mentagui

Department of Mathematics, Faculty of Sciences,
Ibn Tofaïl University, Kénitra, Morocco
nabila.hamdoun@gmail.com
dri_mentagui@yahoo.fr

Abstract

There are many existing studies released in the field of Computer Vision, especially the field of Automatic License Plate Recognition. However, most of them are focused on using one method at the time, such as Thresholding algorithms, Edge Detections or Morphological transformations. This research paper proposes to automate the License plate recognition process, by combining four algorithms from the three methods mentioned above: Adaptive Thresholding, Otsu's Thresholding, Canny Edge Detection and Morphological Gradient applied to Edge Detection. The Goal achieved is to obtain the best binary image from those methods, and the statistical technique used in, is the median of pixel's intensity of all output images obtained by the four methods. Additionally, this research offers a comparative study on thresholding techniques to choose the best method for binarizing an image, which is the first and crucial step of Automatic License Plate Recognition Process.

Keywords: Image Pre-Processing, Deep Learning, Automatic License Plate Recognition, Thresholding Image, Canny Edge Detection, Morphological Transformations, Otsu Algorithm, Adaptive Thresholding, Global Thresholding

ACM 2012 CCS Concepts: Software and its engineering → Software notations and tools → Development frameworks and environments → Application specific development environments

Mathematics Subject Classification 2020: 94A08, 94A12, 94A29, 97R50, 97R60

Received: March 11, 2022, *Accepted:* March 28, 2022, *Published:* July 4, 2022

* *Corresponding author*

Thanks: This work was done while the authors were working on Deep Learning Research in Ibn Tofaïl University, Kénitra, Morocco.

Citation: Nabila Hamdoun, Driss Mentagui, Image Processing in Automatic License Plate Recognition Using Combined Methods, Serdica Journal of Computing 16(1), 2022, pp. 1-23, https://doi.org/10.55630/sjc.2022.16.1-23

1 Introduction

License Plate Recognition system (License Plate Recognition, LPR) occupies a very important place in the intelligent transportation system. At the same time, many applications in the field of Intelligent Transportation are based on Image Processing, Pattern recognition and Machine vision research. One of the key points is that Automatic license plate recognition (ALPR) mainly includes license plate positioning, Skew correction, Character segmentation and character recognition, etc. At present, there are many license plate location algorithms, and most of them are based on a certain theoretical knowledge system, Image color information, Texture analysis, Edge detection, Mathematical morphology, etc.

Automatic License Plate Recognition (ALPR) systems capture the vehicle's license plate and recognize the license number and other required information from the captured image, like Prefecture or Province and License Series.

ALPR become recently requested by many significant applications like Toll Gate systems, Law enforcement, National Gendarmerie, etc. The objective of this system is to recognize the License Plate with high accuracy.

In this study, we focus on three important phases: The first one explains the Pre-processing phase which enable the segmentation of the plate. The second one refers to Feature extraction, and last one contains extraction Learning. Then, to obtain a good extraction number, the setup of Pre-processing and binarization Image must be meticulous. In general, Methods that deal with image binarization are categorized in two main classes: Global approach and Local one.

In a Global approach, threshold is in a single value predetermined for the entire image, and there is a good separation between the foreground and the background. However, if the image suffers from shadows and non-uniform illumination, this approach doesn't give a satisfying result. Otherwise, in a Local approach, the threshold value is computed using a local information or neighborhood of each pixel.

Since no standard technique exists that performs in all degradation cases for all image type, there is many proposed studies [1] related to the combination of a set of binarization techniques. However, most of other studies are focused on using one method at the time. Indeed, there are recent publications on one of the parts covered in our article. For example, Al-Shemarry and Li's study (2020) [2], aims to develop an advanced detection using a robust preprocessing enhancement method for accurately detecting the license plates, which includes the combination of a Gaussian filter, an enhancement cumulative histogram equalization method, and a contrast-limited adaptive histogram equalization technique.

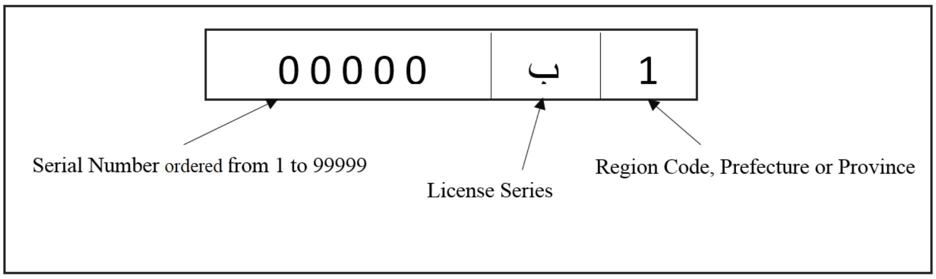

Figure 1: Moroccan License Plate Shape.

Chowdhury et al. (2018) [3], proposes an image segmentation technique to segment out the Region of Interest (ROI) from an image, using an improvised Sliding Concentric Window (SCW) algorithm. Chen et al. (2013) [4], propose in their study a combination of theoretical research and practice to license plate recognition using a visual result. Lin et al. (2022) [5] use an Edge-AI-Based Real-Time Automated License Plate Recognition System, tested in selected real-life test environments, which achieves a high recognition rate at a low computational cost.

Thus, in this paper, we propose a novel methodology for an efficient pre-processing and binarization image. The idea behind, is to use the complementarity in success of each technique by combining them. The well-known Algorithms chosen for the proposed method was applied one by one, however only the ones with better results are selected, and some of them improved by Morphology transformations, to be part of the combined median algorithm.

2 Statement of the problem

2.1 License Plate Segmentation

The Recognition system extract license number portion of the plate which is transformed into alphanumeric characters for recognition. Then, these alphanumeric characters are compared to many databases to identify the license number of vehicle of interest. Moroccan License Plate is a way to identify vehicles registered in Morocco. This Registration system date to 2000 which includes a maximum of five-digits followed by two vertical lines. Between the vertical lines, there is an Arabic letter referring to the series number. Finally there are one or two digits region code (Prefecture or Province) for the origin of the vehicle. The License plate is usually show black text on a white background, see Fig. 1.

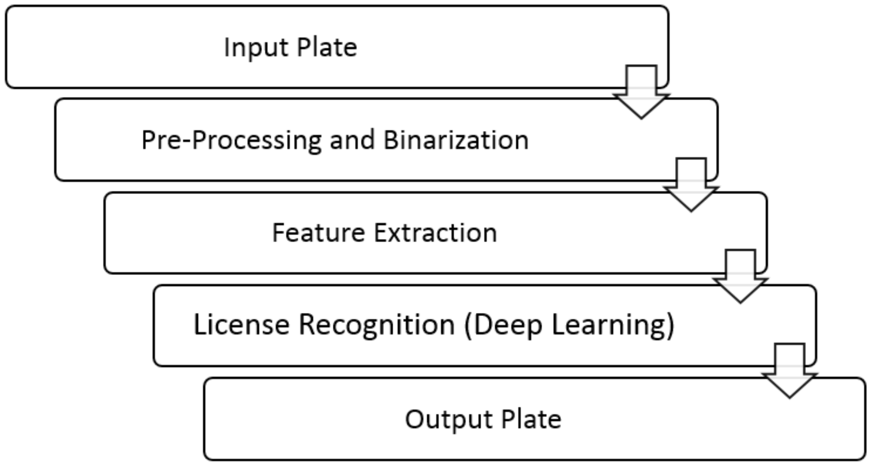

Figure 2: Process of Automatic License Plate Recognition System.

2.2 Problem related to Recognition System

There are many problems related to ALPR systems, especially when system try to identify the License plate in the image. Some of them include:

i. *Illumination:* Images Illumination can varies according to outdoor conditions such as rain, headlights lighting, fog etc.

ii. *Background:* License plate backgrounds may contain other letters related to state organizations and military corps vehicle. Those License plates are all characterized by a black background and white digit characters, and red letter which can be difficult to separate from the foreground of the image

The recognition system used in this study is presented as the following Fig. 2.

Multiple binarization techniques help to obtain satisfying results in the extraction of Region and license number in the license plate. The extraction of license plate information includes four stages:

1. Pre-processing the license plate
2. Binarization
3. Separation of Region and license number
4. Extraction of license plate

Preprocessing is an important step in image processing systems which helps to improve the visual appearance of the license plate and removes noise and distortion. The most important method used to achieve a higher quality segmentation is Histogram equalization, and we used it before some algorithms in binarization step. Indeed, most of studies focus on only one algorithm in binarization step, thus, in this study, we propose a combined binarization algorithm which shows a better result.

3 Experiments and main results

3.1 Preprocessing of License Plate

3.1.1 Histogram equalization

Histogram equalization is a method used in image processing of contrast adjustment using the image's histogram. It improve the contrast on the image in order to allow areas of lower local contrast to gain a higher contrast [6]. Histogram equalization do this by effectively spreading out the most frequent intensity values, but Basic histogram equalization uses global contrast which is not a good method for all the images. The advantage of this method is that it is a straightforward technique and easy to implement. A disadvantage of it is that it is indiscriminate: It may increase the intensity or contrast of background noise, while decreasing the usable pattern, signal or edge, see Figs. 3, 4, 5.

In a nutshell, we observe that, despite of the low quality of the image, Histogram Equalization provide a clear image by adjusting its contrast, which will help a lot in the next step: Binarization.

3.2 Binarization of the License Plate Image

Binarization is among the efficient methods to convert the gray scale in the image to a binary image. It consists on specifying two values of pixels: 1 for White and 0 for Black. Binarization is an important step in improving the quality of the process or algorithm used to extract the license number of the plate, because we need to separate the foreground and background pixels in the image in order to detect or recognize numbers and letters in the image. To accomplish this step, we need to determine a threshold value: each pixel with value greater than threshold is classified as foreground pixel and with value less than as background pixel.

For an image intensity value $I(x, y)$ where (x, y) is location of pixel, let t be

Figure 3: Original Image on left before applying Histogram Equalization, Preprocessed Image by HE on right.

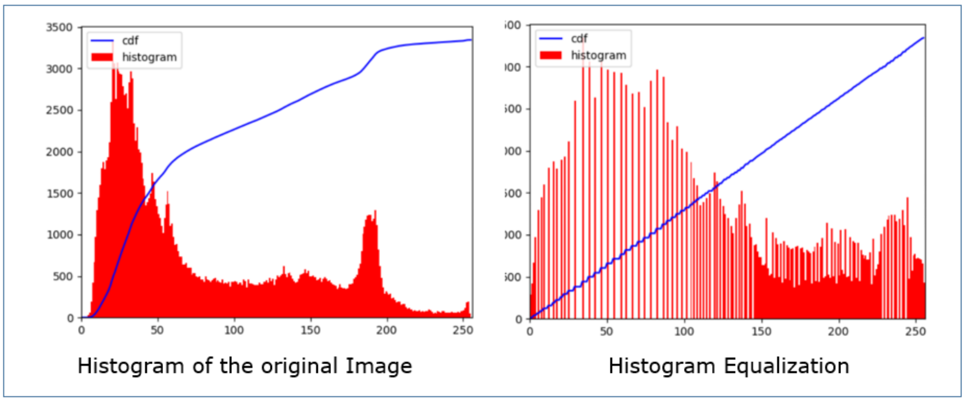

Figure 4: Histogram of the input image before and after Histogram Equalization Method.

Figure 5: Histogram Equalization used in an input image with low contrast on left.

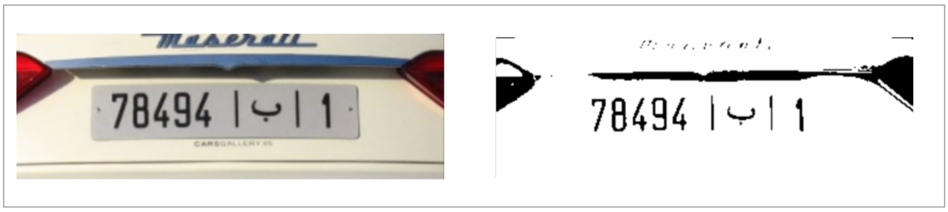

Figure 6: Binarization using Global Thresholding value =70, of a Moroccan License Plate.

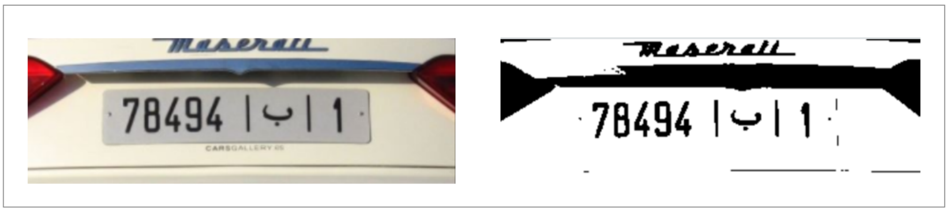

Figure 7: Binarization using Global Thresholding value =127, of a Moroccan License Plate.

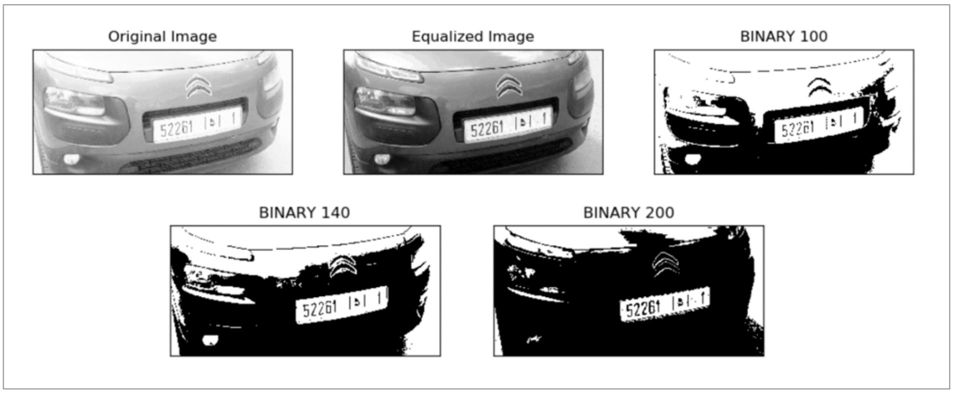

Figure 8: Global Thresholding applied on an equalized License plate image with many Threshold values.

the threshold fixed, then the binary image is given as the function $O()$:

$$\begin{aligned} O(x,y) &= 0 \quad \text{if } I(x,y) < t, \\ O(x,y) &= 1 \quad \text{if } I(x,y) > t. \end{aligned}$$

The two values 0 and 1 determine Black and White pixel. The value of the Threshold is quite difficult to determine, especially if there is gray level variance between background and foreground contrast, then in case the wrong threshold is selected, the results of classifying background and foreground won't be satisfying. The quality and the accuracy of License plate Extraction Process depends automatically on the quality of binarized image, then considering a single binarization method is not a good technique due to different environments where images are usually captured. This is why we propose to combine some of methods explained below, in order to obtain an hybrid Binarization Method.

Before constructing the hybrid Binarization technique, we apply some binarization methods to the input license plate and then select the best one depending on the quality of results obtained using the selection threshold or criteria.

3.2.1 Global thresholding

The Global Thresholding Method or the Fixed Thresholding Method, is one of the easiest binarization methods to apply. It selects a threshold value in order to classify a pixel of an image as foreground or background [7, p. 84]. As shown in Figs. 6 and 7, outputs can vary, depending on an input threshold value.

From the Figs. 6 and 7, it can be noticed that image output at threshold value T=127 is better for this license plate image.

We have implement the bellow algorithms on 300 images, captured by a mobile phone (8-megapixel 1/3.2-inch sensor and an f/2.5 lens.), using OpenCV library on Python in a machine (Core i7, 2,4 GHz; 8G RAM, 64bits).

The Fig. 8 shows several threshold values applied for the same image of license plate obtained after Histogram Equalization.

It is clear that in this case the threshold value 127 is not the ideal one, because we obtained a better binarization with a threshold = 140 as shown in the figure 8, used in the proposed python application. Thus, this means that the threshold value T=127 is not optimal for all the images. So it is really difficult to decide which threshold value is optimal for a current input image. To overcome this issue we will discuss further more binarization methods in which optimal threshold value is computed depending on input image.

Algorithm 1: Basic Code for Global Thresholding

1. Initial estimate of the Threshold T.
2. Segmentation of two classes, foreground and background:
   - $G1$, pixels brighter than T. Greater than T. Set 1 for White,
   - $G2$, pixels darker than/or equal to T. Less than T. Set 0 for Black.
3. Show the new image with 0 and 1 only.

3.2.2 Adaptive Thresholding

Whereas the conventional Global thresholding applied previously, which uses a global threshold value for all pixels, Adaptive thresholding changes the threshold dynamically over the image depending on lighting conditions in the input image. Simple and fast functions most applied, include the mean of the local intensity distribution to find the optimal threshold value [7, pp. 91-96].

In order to take into consideration the variation in illumination throughout the image, the common solution is adaptive thresholding. The main difference in this method is that a different threshold value is computed for each pixel in the image by calculating the mean or the median of specified neighboring pixels. This technique provides more robustness for changes in illumination.

Specifically, while traversing the image, an approximate moving average of the last s pixels seen is calculated. If the value of the current intensity of pixel is lower than the average then it is set to black, otherwise it is set to white. This method works better because comparing a pixel to the average of nearby pixels will preserve hard contrast lines and ignore soft gradient changes. There is also another manner to compute the threshold value of the bloc, by using Gaussian Weighted Average [8, p. 173]. But after applied both methods, we obtained approximately the same result and decided to keep the first one.

Algorithm 2: Basic Code for The Adaptive Thresholding. Import an image with the dimension $n \times p$. Let $S \times S$ be the region around the pixel and C is the constant subtracted from the threshold value calculated for each pixel:

1. Select a value of S and C
2. For i in 1 to n

   2.1. For j in 1 to p

2.1.1. Compute the Threshold value for each pixel ij:
$T_{i,j} = mean(Intensities\ for\ the\ bloc\ S \times S\ ) - C$

2.1.2. Segmentation of two classes, foreground and background:

- If pixel is brighter than $T_{i,j}$ / Greater than $T_{i,j}$.
Set 1 for White.
- If pixel is darker than/or equal to $T_{i,j}$ / Less than $T_{i,j}$.
Set 0 for Black.

3. Show the new image with 0 and 1 only.

Comparison between Global Thresholding and Adaptive Thresholding. In order to show the difference between the two algorithms, we've applied both of them on many license plate image with varied illumination in many sides, see Fig. 9.

It is visually remarkable that the Global Thresholding method cannot reach a satisfying result with such image with varied illumination, where the Adaptive Thresholding Method brings a better result with the moving average method specified previously. We have applied the Adaptive Thresholding Method using OpenCV on Python, and tested many values of the constant C and decided to fixed $C = 2$, which show a clear output image and set S, the size of the neighborhood, to 11 for an image size of 614×295.

3.2.3 Otsu algorithm

Otsu's method, named after Nobuyuki Otsu [9] is another method used in converting a gray-level image to a binary image or to perform segmentation image thresholding. It is considered as a statistical decision theory problem which minimize the error incurred in assigning pixels to two classes: foreground pixels and background pixels.

The Algorithm maximizes the variance between the foreground and background pixels in the image, by calculating the optimum threshold separating the two classes.

Let t be the optimum threshold.

The first objective is to minimize the variance within classes $\sigma_w^2(t)$:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t).$$

Weights $q(t)$ are the probabilities of the two classes separated by a threshold t, and I is the maximum level of gray in the image:

$$q_1(t) = \sum_{i=1}^{t} P(i), \qquad q_2(t) = \sum_{i=t+1}^{I} P(i).$$

And the class means are given by:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)}, \qquad \mu_1(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}.$$

Finally, the individual class variances are:

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}, \qquad \sigma_2^2(t) = \sum_{i=1}^{t} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}.$$

All we need to do is just run through the full range of t values [1,256] and pick the value that minimizes the variance within classes.

Otsu Method shows that minimizing the intra-class variance is the same as maximizing inter-class variance. Indeed the basic idea is that the total variance does not depend on threshold because for any given threshold, the total variance σ_T^2 is the sum of both variances, between σ_B^2 and within-classes σ_w^2 [10]:

$$\sigma_T^2 = \sigma_w^2(t) + q_1(t)[1 - q_1(t)][\mu_1(t)\mu_2(t)]^2.$$

Algorithm 3: Basic Code of Otsu Thresholding Method.

1. Compute histogram and probabilities of each intensity level.
2. Set up initial $w_i(0)$ and $\mu_i(0)$.
3. Step trough all possible Threshold value t from 1 to Maximum Intensity.
    - 3.1. Update w_i and μ_i.
    - 3.2. Compute Between-classe Variance σ_B^2.
4. Desired threshold value correponds to the maximum of σ_{B^2}.

Finally Binary image is obtained by assigning each pixel value in the input image less than T (the optimum threshold) value as background pixel and pixel value greater than threshold value as foreground pixel in the image.

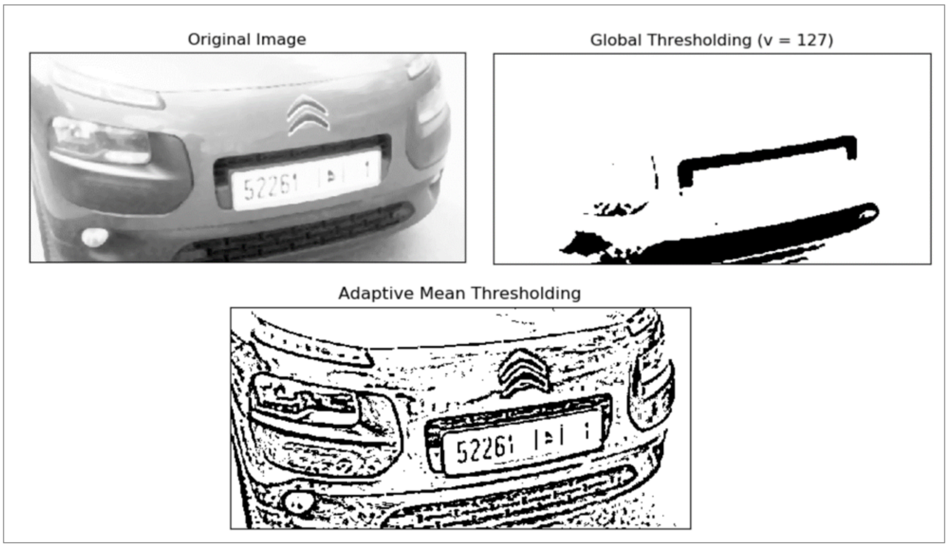

Figure 9: Global Thresholding and Adaptive Thresholding applied on the same input image.

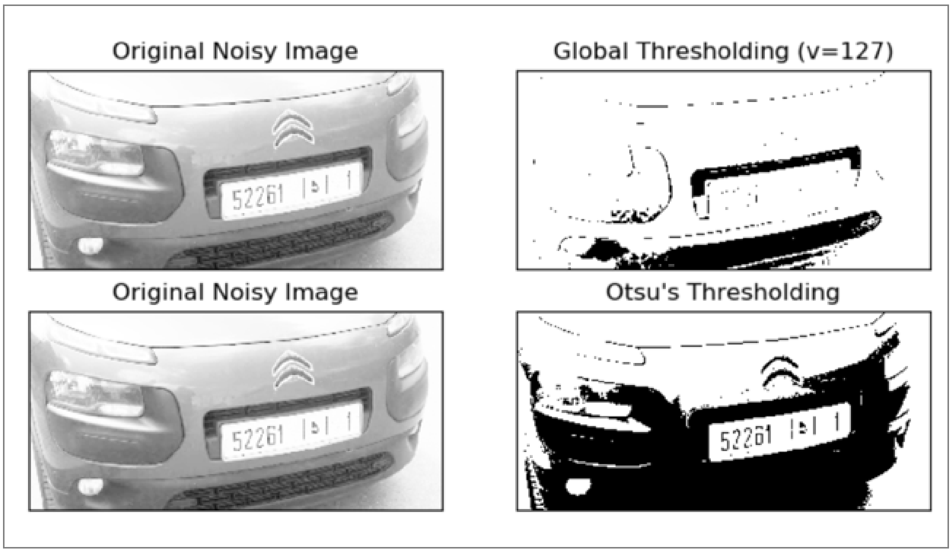

Figure 10: Global Thresholding and Otsu applied on the same input image.

Application of Global Thresholding and Otsu Thresholding on a license plate image. The Fig. 10 shows that the Otsu Thresholding Method provide a better result than The Fixed Thresholding Method. Thus, the Global Thresholding will not be implemented in the combination method which will contain satisfying algorithms in a later section.

3.2.4 Edge based binarization

Edge based binarization is a binarization method which detects the contour of the text in the license plate and utilizes a local thresholding method to decide the inner side of the pattern, and then fills up the contour to shape characters that are recognizable by Deep learning algorithm. This method is currently used for two typical situations where usually binarization methods failed; the first one is complex background and the second is low contrast. Using this algorithm, text or characters with complex or low contrast background can be well separated from each other, and hence can be correctly recognized by the software used.

Otherwise, when the characters in the image are too small, thresholding methods may work better, since the edge detector is not able to detect the character boundary accurately in this case. Specifically, if the height of the characters is greater than 30 pixels, our Edge binarization method will be applied, otherwise an Adaptive thresholding method works better as previously seen.

This method is used to find the edges for the input image by reducing the amount of data to be processed. It was developed by J.F. Canny in 1986 [11], and also called the Canny Edge Detector. There are three criteria for this method:

1. Low error rate by detecting a maximum of edge possible in the image.
2. Good localization: minimize the distance between edges detected and edges in the real image.
3. Minimal response: Detector gives only one response per edge.

Algorithm 4: Basic Code for Canny Edge Detector.

1. Import a grayscale image.

2. Filter out any noise by using a Gaussian Filter, choose for example a size s=5 :

$$K = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

3. Compute the intensity gradient of the image. For this, a procedure analogous to Sobel is applied :

   a. Apply a pair of convolution masks (in x and y directions):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \qquad G_Y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

   b. Calculate an approximation of the gradient at each point of the image by combining both results above: Compute the gradient strength and direction with

$$G = \sqrt{G_x^2 + G_y^2}, \qquad \theta = \arctan \frac{G_y}{G_x}$$

   The direction founded is rounded to one of four possible angles (namely 0, 45, 90 or 135)

4. Application of Non-maximum suppression to remove pixels that are not considered to be part of an edge. Hence, only thin lines (candidate edges) will be retained.

5. The final step. Canny does use two thresholds (upper and lower Hysteresis):

   – If a pixel gradient value is higher than the upper threshold value, the pixel is accepted as an edge.
   – If a pixel gradient value is below the lower threshold value, then it is rejected.
   – If the pixel gradient is between the two thresholds, then it will be accepted only if it is connected to a pixel that is above the upper threshold value.

In this method, we have used the Sobel Filter [12] to compute the gradient intensity after applying Gaussian Filter. Indeed, among the edge detection methods developed so far, Canny Edge detection algorithm is one of the best methods strictly defined that provides reliable detection [11]. Thanks to its simplicity of process for implementation, it became one of the most popular algorithms for edge detection.

Figure 11: Canny Edge Detector applied on license plate image with two threshold values: 50, 100.

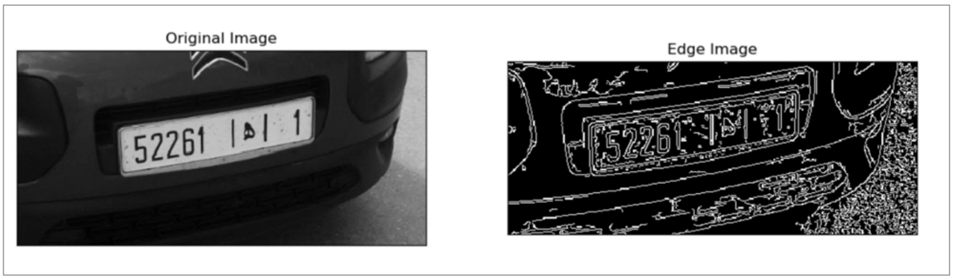

Figure 12: Canny Edge Detector applied on license plate image with two threshold values: 10, 50.

Figure 13: Canny Edge Detector applied on license plate image with two threshold values: 100, 200.

Figure 14: Morphological Gradient Transformation applied on the Canny Edge Detector Method.

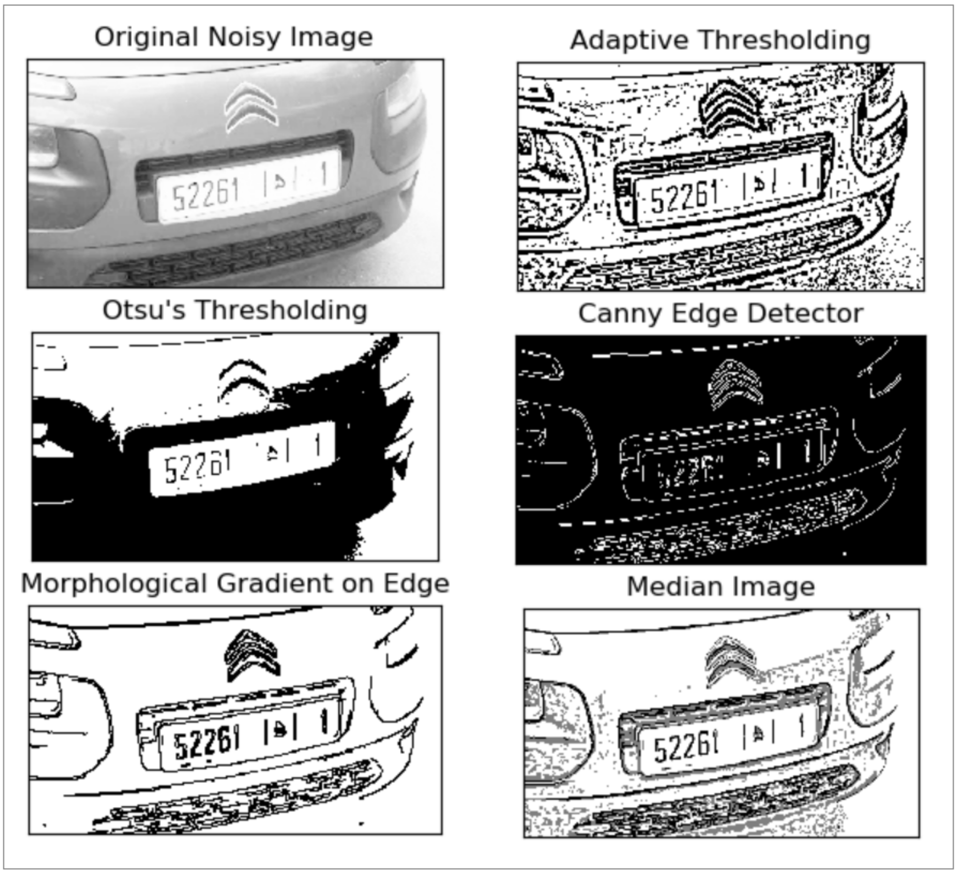

Figure 15: Experiments and Results of all algorithms mentioned and the proposed one.

Application or Canny Edge Detector on a license plate image. We have tested, as follow, many values of the two thresholds (lower and upper one) that must be specified in the Canny Edge Detector Method, see Figs. 11, 12, 13.

The results on several values let us to decide to fix the lower Threshold value to 100 and the upper Threshold Value to 200 in all experiments. The low threshold is typically set to 1/2 of the high threshold in this case.

Moreover, in the proposed algorithm relative to our work, we have needed to add a Morphological Transformations [13] on the Canny Edge Method, to obtain clear characters of the license plate. Thus, we've chosen Morphological Gradient, which is the difference between the dilation and the erosion of a given image. There are many other Morphological Transformations that we have tested, but the only one method which shows a better result is the Morphological Gradient, see Fig. 14.

We observe that the Morphological Gradient applied on Canny Edge Detector shows a satisfying and clear result, so that the implementation of the output image in our proposed algorithm will perform better.

3.2.5 Niblack Algorithm

Niblack method, W. Niblack (1986) [14], is a well known local threshold binarization approach. The main purpose of this method is to compute the threshold value based on local standard deviation and local mean of the colors of the neighboring pixels in an area (window) of predefined size (w). The threshold for each pixel was determined by [14]:

$$T = mean + k * stdev$$

The optimal combination of k and w founded in experiments, which produce a good binary image is respectively (0.24 and 25).

3.2.6 Median of Images

The median is value separating the higher half of a data sample, from the lower half, or the center value of a sorted list of elements. If the list contain even number of elements, the median will be the average of the two middle values. This helps us avoid considering outliers or values that are outside the range. Our proposed algorithm in Image processing consist on computing the median of a list if pixels. Those pixels must be at the same location in a stack of output images from each method seen above: Adaptive Thresholding, Otsu Thresholding, Canny Edge Detector and Morphological Gradient applied to Edge Detector. Then, the resultant image pixel values is given by choosing the median value from the stacked images at each location. The implementation of the proposed Algorithm is provided in algorithm 5.

Algorithm 5: Basic Code for proposed algorithm, Median of Images.

1. Import an image.
2. Let size of the image be $m \times n$.
3. Let the output image be O of size $m \times n$.
4. Create 4 images output from the four algorithms : Adaptive Thresholding I_1, Otsu'sThresholding I_2, Edge Detector I_3, Morphological Gradient on Canny I_4.
5. For i in 1 to m do

    5.1. For j in 1 to n do

    $O[i,j] = median(I_1[i,j], I_2[i,j], I_3[i,j], I_4[i,j])$

    end for

    end for

Experiments and Results of the proposed algorithm are given in Fig. 15.

We observe that, when the quality of the image is not good enough, because of the high degree of illumination, each algorithm don't provide a satisfying result if it's applied alone. Otherwise, the median algorithm proposed shows that, even we have a noisy image, the result is clear enough to allow a good extraction of the characters. The duration of execution is 6,001 seconds.

In a nutshell, because one binarization technique does not provide efficient results, we consider multiple binarization techniques by using median. Indeed, considering multiple binarization techniques helps in segmenting the License plate regardless of the environment in which the image was captured.

3.2.7 Combination of binarization techniques

Let $I1(x,y), I2(x,y), ..., IN(x,y)$ represent the results images of N different binarization methods, which have been applied to input image $IH(x,y)$. N is selected as an odd number $(N = 2m + 1)$.

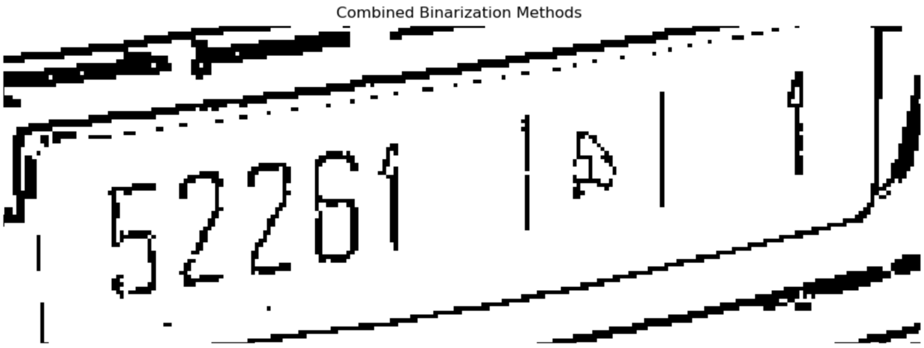

Figure 16: Results of Proposed Combined Method applied on a License plate input image.

Input Binary Images $I_i(x, y)$ are defined as follows:

$$I_i(x, y) = \begin{cases} 1 & \text{Background,} \\ 0 & \text{Foreground,} \end{cases}$$

where $1 \leq i \leq 2m + 1$.

Starting from applying selected binarization method presented previously, we calculate a binary image B_I which combines the N binary image results. The objective is to mark as foreground pixels only those pixels that the majority of the binarization methods selected classify effectively as foreground, B_I is calculated as follows:

$$B_I(x, y) = \begin{cases} 1 & \text{if } \sum_{i=1}^{2m+1} I_i(x, y) \geq m, \\ 0 & \text{otherwise.} \end{cases}$$

4 Experiments and Results

In this proposed Combined Method, we have chosen only Algorithm with the best binarization result: Otsu's method, Niblack's Method and Median of algorithms applied previously. Fig. 16 shows that the proposed Method helps to obtain a binary image without noise around the License Plate, in order to extract license number easier.

Rank	Binarization Method	Average Precision-Recall
1	Proposed Combined Method	98,4%
2	Otsu's Method	96,1%
3	Median of 4 Selected Algorithms	94,5%
4	Niblack Method	93,8%
5	Adaptive Thresholding	91,2%
6	Morph. Gradient on Canny Edge	90,1%
7	Sauvola Method	88,2%
8	Bernsen Method	78,9%

Table 1: Performance Measure of Binarization Methods.

We have implement the bellow algorithms on 100 images, captured by a mobile phone (8-megapixel 1/3.2-inch sensor and an f/2.5 lens.), using libraries on Python: OpenCV, Pillow, scikit-image, Mahotas and Matplotlib, in a machine (Core i7, 2,4 GHz; 8G RAM, 64bits). The obtained binary Image from the proposed Method is quit satisfying without adding a Skeleton Transformation. We compared the result with many other well-known algorithms: Bernsen Algorithm [15, 17], Sauvola adaptive method [16], as shown in Table 1.

To evaluate the performance of binarization methods applied, we calculate the Average Precision-Recall [18] which among famous metrics for evaluating the performance of a classifier or an algorithm.

Average Precision-Recall is a useful measure of success of a specific algorithm when the classes – intensities of pixels in this case – are very imbalanced. Moreover, precision is a measure of result relevancy, while recall indexes is a measure of how many truly relevant results, compared to input data, are returned.

By computing a precision and recall at every array of pixels in the obtained image, we can plot a precision recall curve, plotting precision $p(r)$ as a function of recall r. Average precision computes the average value of $p(r)$ over the interval from $r = 0$ to $r = 1$ [18]:

$$AverageP = \sum_{k=1}^{n} P(k)\Delta r(k).$$

Where k is the rank of pixel in the image, n is the number of total pixels, $P(k)$ is the precision at cut-off k in the list, and $\Delta r(k)$ is the change in recall from pixel $k - 1$ to k.

We observe, in Table 1, that the Proposed Method give an Average Precision-

Recall equal to 98,4%, which is the best performance indice comparing to other seven (7) applied methods.

4.1 Extraction of License Number

Most of the results published in ALPR concentrate only in license number, so identifying the region code for the vehicle is out of its scope. Also Algorithms involved in those results wasn't accurate enough and show an error superior to Deep Learning's Algorithms. Indeed, the task of recognition each of character in a plate is quite challenging due to outdoor illumination condition and the diversity of plate formats. This is why we must try such techniques mentioned previously, involved in pre-processing and binarization of the image plate efficiently.

5 Conclusion

In this paper, we described a combined method proposed in License plate recognition process. Indeed, lot of work has been done for evaluating some methods in Thresholding image, but our paper compared different types of threshold methods and showed the effect of using a combined method on standard image, rather than a single method to gain the complementarity of best selected binarization methods.

We proposed an methodology to aid text region extraction from the vehicle images, which starting with Pre-processing Image, and binarization using Median of intensities from many tested algorithms like Adaptive Thresholding, Otsu's Thresholding, Edge detection, Morphological Gradient applied to Canny edge Detector, Niblack Method, and then computing a new binary image combining the median and other methods.

Experimental results show that the proposed method performs better in terms of ability to extract clear characters with the least amount of noise compared to the other binarization schemes, and with the lowest rate of error. This method has been tested on a large number of vehicle images captured by a camera under varying illumination conditions.

Moreover, the major contribution of this article is to help people and researchers interested in the practical detection of vehicle registration numbers.

Future scope may include other segmentation or Morphological techniques using Real-Time Systems. Also using a larger dataset of images may lead to a better convergence in further studies.

References

[1] Lubna, N. Mufti, S.A.A. Shah, "Automatic Number Plate Recognition:A Detailed Survey of Relevant Algorithms", *Sensors*, 21(9), 2011, Article № 3028.

[2] M.S. Al-Shemarry, Y. Li, "Developing Learning-Based Preprocessing Methods for Detecting Complicated Vehicle Licence Plates", in *IEEE Access*, 8, 2020, pp. 170951-170966.

[3] W.S. Chowdhury, A.R. Khan, J. Uddin, "Vehicle License Plate Detection Using Image Segmentation and Morphological Image Processing", *Advances in Intelligent Systems and Computing*, 678, 2018, pp. 142-154, Springer, Cham.

[4] C.Y. Chen, B.Z. Cheng, X. Chen, F.C. Wang, C. Zhang, "Application of Image Processing to the Vehicle License Plate Recognition", *Advanced Materials Research*, 760, 2013, pp. 1638-1641, Trans Tech Publications.

[5] C.-J. Lin, C.-C. Chuang, H.-Y. Lin, "Edge-AI-Based Real-Time Automated License Plate Recognition System", *Applied Sciences*, 12(3), 2022, Article № 1445.

[6] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Third Edition, Pearson Education, 2011.

[7] E.R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, Third Edition, Elsevier, 2005.

[8] O. Lecarme, K. Delvare, *The Book of GIMP: A Complete Guide to Nearly Everything*, No Starch Press, 2013.

[9] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 1979, pp. 62-66.

[10] J.F. Canny, EECS Department, University of California, Berkeley, Tech. Rep., Retrieved 20 May 2009.

[11] J.F. Canny, "A Computational Approach to Edge Detection", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986, pp. 679-698.

[12] Bibliographic Citations for Irwin Sobel. `http://dblp.uni-trier.de/pers/hd/s/Sobel:Irwin.html`.

[13] E.R. Dougherty, J. Astola, *An Introduction to Nonlinear Image Processing*, SPIE Press, 1994.

[14] W. Niblack, *An Introduction to Digital Image Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1986.

[15] J. Bernsen, "Dynamic thresholding of gray-level images", Proceedings of the Eighth International Conference on Pattern Recognition, 1986, pp. 1251-1255.

[16] J. Sauvola, T. Seppanen, S. Haapakoski and M. Pietikainen, "Adaptive document binarization", Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1, 1997, pp. 147-152.

[17] Wan Azani Mustafa, Mohamed Mydin M. Abdul Kader, "Binarization of Document Images: A Comprehensive Review", *Journal of Physics: Conference Series*, 1019, 2018, Article № 012023.

[18] C.D. Manning, P. Raghavan, H. Schütze. *Introduction to Information Retrieval*, Cambridge University Press, 2008.