People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University May 8, 1945 guelma

Faculty of Mathematics, Computer Science and Material Sciences

computer science department

Bachelor's degree thesis

Branch : Computer Science

Option : Informatic Systems

design and realization of an Algerian license plate recognition system

Supervisor :

Hallaci Samir

Author:

Brahmia Abdelbacet

Nouar Ikram

2019 - 2020

Acknowledgements

We thank God for everything he offered us We would first like to thank our supervisor Mr. HALLACI SAMIR, for having supervised and directed this work with great scientific rigor, for its availability, guidance and help, we express to him our most respectful feelings and our deepest gratitude.

We would like to end by thanking our families, for the support and encouragement they have given us ...

in October 1, 2022.

Contents

List of Figures

List of Tables

Abstract

Automated license plate recognition is a technology for identifying vehicles through the use of optical character recognition (OCR) techniques.

The system runs on standard PC hardware and can be linked to other applications or databases. It begins by using a series of image manipulation techniques to detect, normalize and magnify the image of the license plate, and ends with optical character recognition (OCR), which is used to extract the alphanumeric characters from the plate.

This system can be used to manage the flow of cars in a parking, or to detect traffic violations automatically, or to find a car charged in a case of fraud, or a city toll project.

So to realize our system, the algorithm has to be broken down into five steps in order to overcome all the difficulties:

- Detecting the presence of movement of a car and segmenting it from its environment
- detect, transform and crop plates on an input image sequence. This ensures that the numbers are aligned before inserting them into OCR. This pre-processing step is very important.
- segment each character.
- recognize the plate.
- manage the access of the cars through their identities (license plate number).

La reconnaissance automatisée des plaques d'immatriculation est une technologie permettant d'identifier les véhicules grâce à l'utilisation de techniques de reconnaissance optique de caractères (OCR).

Le système fonctionne sur du matériel PC standard et peut être lié à d'autres applications ou bases de données. Il commence par utiliser une série de techniques de manipulation d'image pour détecter, normaliser et agrandir l'image de la plaque d'immatriculation, et se termine par la reconnaissance optique de caractères (OCR), qui est utilisée pour extraire les caractères alphanumériques de la plaque.

Ce système peut être utilisé pour gérer le flux de voitures dans un parking, ou pour détecter automatiquement des infractions au code de la route, ou pour trouver une voiture chargée en cas de fraude, ou un projet de péage urbain.

Alors pour réaliser notre système, l'algorithme doit être décomposé en cinq étapes afin de surmonter toutes les difficultés :

- Détecter la présence de mouvement d'une voiture et la segmenter de son environnement
- détecter, transformer et recadrer les plaques sur une séquence d'images d'entrée. Cela garantit que les nombres sont alignés avant de les insérer dans l'OCR. Cette étape de prétraitement est très importante.
- segmenter chaque caractère.
- reconnaître la plaque.
- gérer l'accès des voitures grâce à leurs identités (numéro de plaque d'immatriculation).

General introduction... 1

Due to the evolution and progress of the cars industry over the world, and the population has continued to grow, resulting a steady increase in the number of car's usage in many sides of daily life. Today, the car has become an integral part of people's lives at the rate of a car in every house in some countries.

As we use faces to identify individuals, the license plates are used for vehicle identification in all nations. They uniquely identify a car, and its image is important information to identify its owner. Indeed, vehicles are identified manually or automatically.

Automatic reading of vehicle registration plates is considered a mass surveillance approach using detection/localization as well as recognition, to identify a vehicle in still images or image sequences. Many applications such as traffic monitoring, the detection of stolen vehicles, electronic toll collection or the management of parking entrances/exits use this process.

However, the optimal use of the techniques of detection and recognition of license plate in operational scenarios requires controlled lighting conditions as well as a limitation in the installation, speed or quite simply type of plate. The automatic recognition of license plates then remains an open research problem.

The major contributions of this thesis are: detection of any motion, detect, transform and crop plates on an input image sequence. This ensures that the numbers are aligned before inserting them into OCR. This preprocessing step is very important, segmenting each character, recognize the plate, and finally manage the access of the cars through their identities (license plate number).

The plan of our work is as follows:

- The first chapter: in the first chapter we will present objectives of our project and the difficulties that we may face, fields of application, and licence plate forms, we are going to present the Algerian licence plates and their characteristics
- Second chapter: we present in this chapter our conception and step by step all the process and architecture followed in our project.
- Third chapter: the last chapter is dedicated to the application part by presenting the programming language, tools and package used to develop our system, and different tests and discussions.

we end this thesis by a general conclusion and perspectives.

Introduction 2

In this chapter we are going to present our objective for the realization of this system as well as the set of difficulties that may hinder our work.
we will also explain the essential entity which is the license plate and its different shapes around the world and focus on the Algerian license plate and its characteristics.

2.1 Problematic and objective:

The big problem is that our plate is in a very complicated environment (on a dumper of a car with various colours also on a very difficult nature) what makes the detection very difficult.
So our LPR system is the software part based on a set of methods starting from the acquisition using a camera with a specific characteristics, motion detection using some sensitivities, car detection and extraction, segmentation of numbers and finally recognition, until we reach our goal which is the detected and recognized license plate.

Figure 2.1 – *LPR process*

2.2 Difficulties:

Our project may face some problems and obstacles that can negatively affect the results to be achieved, among these obstacles:

- The bad resolution, which can caused by the low quality of the camera in use, or the distance between the camera and the car.

Figure 2.2 – *example of bad resolution*

- Blurred images, due to the bad installation of the hardware (cam movement).

Figure 2.3 – *example of blurred image*

- Poor lighting and low contrast, due to the overexposure, glare or shadows.

Figure 2.4 – *example of good and poor lighting*

- An obscuring object, it can be an object blocking the detection of the plate (example: another car) or something in the plate (example: dust . . .).

Figure 2.5 – *example of obscuring*

- An original font, we can find plates that don't respect the rules about the font of the Algerian plates.

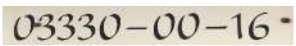


Figure 2.6 – *example of a different font*

- Moving object with high speed, when the car is moving at a high speed, which makes the detection of the plate very difficult if not impossible (the scene will be blurred).

Figure 2.7 – *example of object moves with high speed*

2.3 Fields of application:

Vehicle license plate recognition is a form of automatic vehicle identification system. LPR systems are in a considerable interest because of their potential applications in some areas such as electronic toll collection on motorways, automatic parking garages, petrol stations, zone surveillance, application of speed limits, security, customer identification allowing personalized services, ... etc. Bellow we describe some of those applications.

- Automated parkings, the LPR system is used to control automatically the subscribed members access IN and OUT of the parking, and calculate the parking fees for non-members (comparing exit and entry times).

Figure 2.8 – *parking access*

- Toll, vehicles are no longer required to stop to be classified, on the contrary of the manual toll gates which required the vehicles to be stopped.

Figure 2.9 – *cars waiting to pay for highway*

- Road safety, the registration number is used to find vehicles traveling at high speed, the illegal use of bus lanes and the detection of stolen or wanted vehicles.

Figure 2.10 – *camera to control cars speed.*

- Employee access, the registration number is used to define the authorization for certain vehicles to pass on roads or to enter it in certain spaces (limited length . . .) by the detection and the recognition of the plate which allow us to define the type of the vehicle and to know certain of her characteristic(weight, high,. . .).

Figure 2.11 – *access to special spaces*

- Border crossing, this application helps monitoring the borders by checking the vehicles in and out of the country, to avoid illegal immigration, international fraud, and even the smuggling operations.

Figure 2.12 – *car in crossing borders*

- Smart cities, future cities that will be able to react and communicate with citizens and even objects to facilitates and automates different tasks.

Figure 2.13 – *design of a smart city.*

2.4 LICENCE PLATE FORMS :

depending on the system, the plate may contain numbers, letters, letters and numbers in a specific order or in any order ,here is some model:

- in the European union: France, Germany, Sweden, Hungary, Italy, Span . . .

Figure 2.14 – *some European union plates*

- In the Arabic world:

  Tunisia, Jordan, Saudi, Egypt, Morocco, UEA . . .

Figure 2.15 – *some Arabic plates*

- Other parts in the world: Japan, USA, Russia . . .

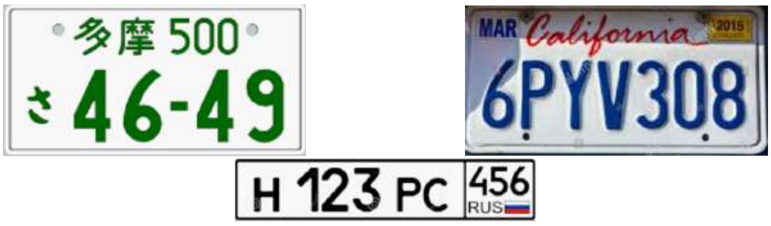


Figure 2.16 – *plates from around the world*

- Some special plate:
  diplomatic, priority cars ...

Figure 2.17 – *diplomatic and gendarme plates*

2.5 Algerian license plate:

Using the Algerian regulations published in the official journal,[8][10] we managed to extract the following properties of Algerian license plates:

2.5.1 Colour

Algerian LPs have a reflective white-grey background for front-plates, and yellow for rear-plates, with black Arabic digits.

2.5.2 Composition

The Algerian license plate registration number is composed (from right to left) of:

- First set of the two first numbers representing the Wilaya of registration (from 01 to 48).
- A group of three Arabic digits, separated from previous by an apparent dash. The first two digits represent the release year, and the last one represents the vehicle's category (see table 1.1)

Vehicle category	Number
Tourism vehicle	1
Truck	2
Van	3
Auto-cars and Auto-bus	4
Road tractor	5
Other tractors	6
Special vehicles	7
Trailer and semi-trailer	8
Motorcycles	9

Table 2.1 – *Algerian vehicle category numbers*

- A group of five Arabic digits separated from previous by an apparent dash, representing the chronological order number of registration in its category.

2.5.3 Size

Algerian vehicle license plates have a rectangular shape placed on a dumper in front of the car, and the back side it placed on rear bumper or trunk, where the longer side is horizontal. The dimensions of normal plates with one row are shown in table 1.2, based on Algerian low, and plates with two rows are shown in table 1.3

Width	455 to 590 mm
Height	100 to 110 mm
Digits height	75 mm
Digits width except "1"	135 mm
Width of the digit "1"	120 mm
Gap between digits	10 mm
Gap between digits and edge	10 mm at minimum

Table 2.2 – *Dimensions of Algerian one row license plates*

Width	275 mm
Height	200 mm
The other dimensions are similar to 1-row plates	

Table 2.3 – *Dimensions of Algerian two row license plates*

We present an example of Algerian license plate in the following figure:

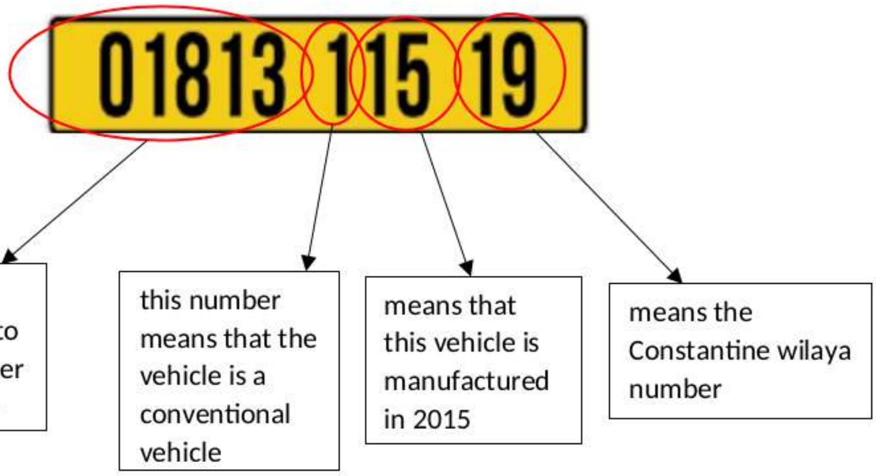


Figure 2.18 – *Algerian license plate*

2.6 CONCLUSION:

In this chapter, we have presented the architecture of any Licence Plate Recognition, we were able to know the field of application of our system, the difficulties and challenges that can affect any LPR system, we exposed the different shapes of the license plates and especially the characteristics of the Algerian license plates.

Conception of our system 3

3.1 Introduction

In any project and in order to have a good design and implementation that suits the defined objectives, it is necessary to write a specification. This is why we decided to develop a specification for the project realization.

3.2 Detailed process of our system

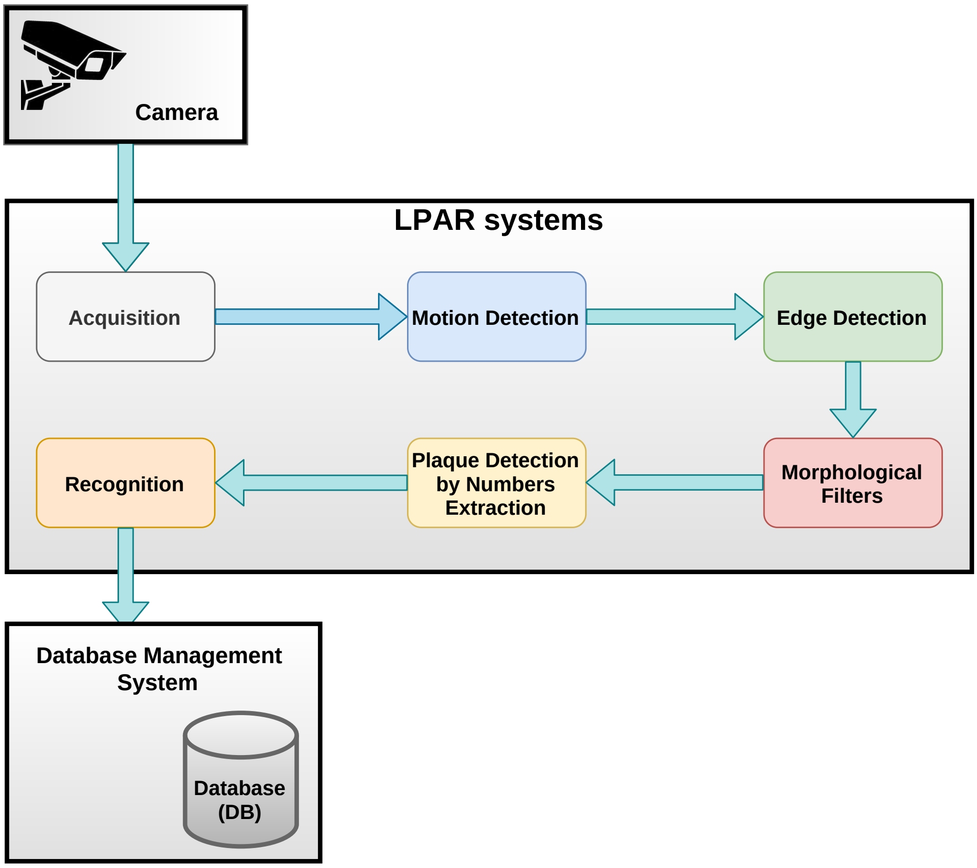


Figure 3.1 – *Conceptual diagram of the proposed system.*

The operating scenario of our system is very simple, as shown in Figure 3.1. At the beginning, the system receive data (images / videos) acquired by the camera. Each input image goes through a set of processing steps starting with motion detection which works on detecting motion in the camera's field of view (any object) using the background subtraction technique. In the next step, the results of the previous step are improved by morphological filters then, the edge detection technique is applied to find the license plate area in the frame for the OCR process. We describe these modules in detail in the forthcoming subsections.

3.2.1 Module Acquisition

The acquisition is the first step in the pattern recognition process. It allows us to transform the object to be recognized from a real or so-called physical shape into a digital shape understood by computers. A camera placed in a place where the object can be detected can do this. The parameters that have to be taken into consideration are:

3.2.1.1 Camera location:

The automatic recognition of license plates consists of a combination of electronic and computerized means using computer vision techniques. Among these means, the camera that must be placed correctly to ensure the best vision.
The placement of the camera can be as shown in the Figure 3.2.

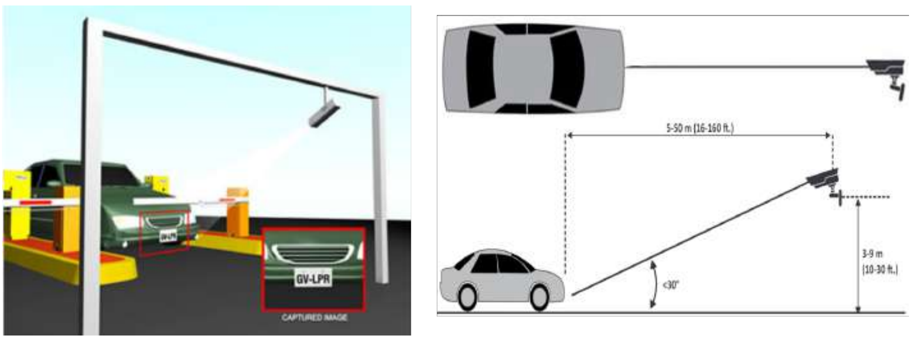


Figure 3.2 – *Camera location*

3.2.1.2 Image acquisition

This is a very important step to have the image through a camera with certain characteristics:

- **FPS**: The number of frames per second or images per second is a unit of measurement corresponding to the number of images displayed in one second by a camera. The measurement of the display frequency can also be expressed in Hertz. The higher the number of frames, the smoother the video.
  If the FPS at the time of projection is higher than at the time of shooting, an acceleration is obtained. In this case, the degree of complexity of processing, transmission, storage, etc. is increased. Conversely, if the number of images per second at the time of projection is less than the number of images per second at the time of filming, you get a slow motion, which negatively influences the real-time system processing.

- ❏ **Resolution**: The resolution is actually the pixel density of the image and not the number of pixels (which is screen definition). "Display resolution" is often misused to refer to the size of images displayed on computer or television screens (resolution), when the first is expressed in pixels and the second in pixels per inch.

- ❏ **Spatial resolution**: The Spatial resolution is a measure of the fineness of detail in an image for a given dimension. A Bitmap image is composed of pixels; a displayed image is defined by a size (in centimeters or inches); the conjunction of these two data is expressed as the number of pixels per unit length. This spatial resolution, which indicates "pixel density", is commonly referred to simply as "resolution".

- ❏ **Colour space**: A colour space is a three-dimensional mathematical model representing the whole of the colours perceptible, usable or reproducible by a human being or a device. Each colour that it contains is thus associated with coordinates determining a precise point and corresponding, for example, to values such as luminance, saturation and hue.

  - ❖ *RGB space*: The RGB colour model is an additive colour model in which red, green, and blue light are added together in various ways to reproduce a broad array of colours.
    The main purpose of the RGB colour model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB colour model already had a solid theory behind it, based in human perception of colours.
    RGB is a device-dependent colour model: different devices detect or reproduce a given RGB value differently, since the colour elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same colour across devices without some kind of colour management. The representation of the colours in this space gives a cube called cube of Maxwel (see Figure 3.3).

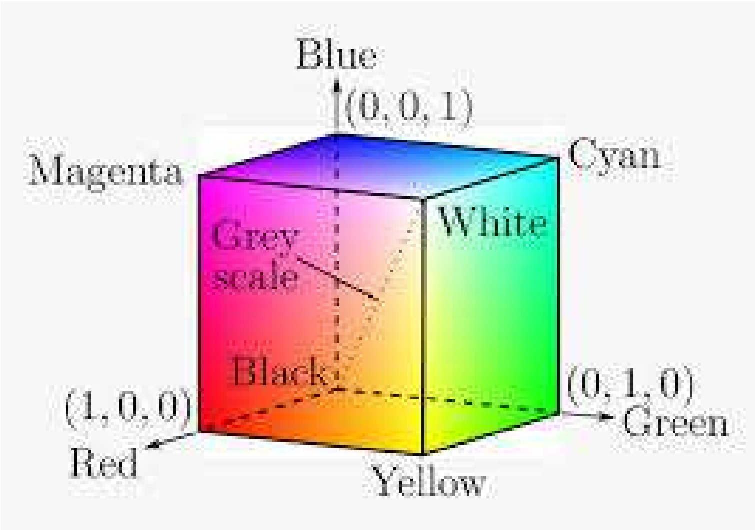


Figure 3.3 – *Presentation of RGB Model*

❖ *HSV space*: The HSV or Hue Saturation Value representation models the way paints of different colours mix together, with the saturation dimension resembling various tints of brightly coloured paint, and the value dimension resembling the mixture of those paints with varying amounts of black or white paint.
HSV is cylindrical geometry (see Figure 3.4), with hue, his angular dimension, starting at the red primary at 0°, passing through the green primary at 120°and the blue primary at 240°, and then wrapping back to red at 360°. the central vertical axis comprises the neutral, achromatic, or grey colours ranging, from top to bottom, white at lightness 1 (value 1) to black at lightness 0 (value 0).

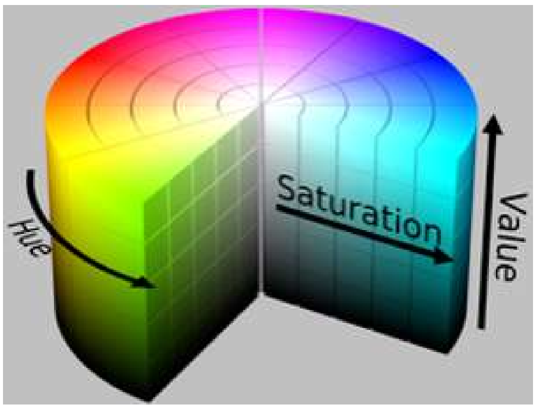


Figure 3.4 – *HSV Model*

❖ *Ycbcr*: The YCbCr model, or more precisely Y'CbCr, is a way to represent the colourimetric space in video. An image captured by any camera is the sum of the colors that compose it, whether the result is in color or black and white. Therefore, even in a black and white image, the signal Y' which represents the luma (not to be confused with the relative luminance noted Y, the

prime symbol of Y' indicating a gamma correction), was created by the sum of red, blue and green.
Send Y', the luminance signal (black and white), and two chrominance information, Cb ($Y' - blue$) and Cr ($Y' - red$). The receiver can recreate green and reproduce a color image. Indeed, if we have Y' ($red + green + blue$) and Cb ($Y' - blue$) and Cr ($Y' - red$), the green can be recreated mathematically using equation (3.1)[5].

$$Y' = 0.3 \cdot R' + 0.6 \cdot V' + 0.1 \cdot B' \tag{3.1}$$

Those three (*FPS*, *Resolution*, *Colour space*) characteristics can influence the result and response time, i.e. having better results in minimum time.

3.2.1.3 Preparation

This step allows us to prepare our frames (acquired images) for the following modules in order to facilitate handling and improve the quality of the results. Among the preparation methods, we used the following techniques:

- *Cropping and resize image*: After the acquisition, we have to resize the image i.e. change the width and height of the image. The objective of this method is to reduce the amount of information to reduce the time of their processing. In our system, we will choose the size (320x240) to keep the quality of the information in the picture, process it in the shortest time, and give us good results.

- *Y channel or grey scale*: *Y* channel or *grey* scale: there is not really a big difference between these two spaces it is almost the same thing, we can consider the *Y* space as a *grey* level but it is more efficient in case we are looking for white objects (it takes into account the light).

  The following steps are followed to calculate the value of the *grey* scale:

  - Convert the colour image from the RGB space to a grey scale space by averaging the three RGB components according to the equation 3.2.

    $$grey = \frac{red + green + blue}{3} \tag{3.2}$$

    The pixels of the resulting image contain a single value between 0 and 255. In our system, we use the space colour Ycbcr and we take only the Y because it is one layer image from 0/255.

- *Noise elimination*: In the literature, there are many noise elimination filters that can be used to reduce the effect of noise. Table 3.3 summarizes the most important of them.

In our proposed system, we have chosen one of the most popular filters which is the linear Gaussian filter. It's usually used to restore the image

Table 3.1 – *Noise elimination filters*

[4]

Filters	Definition	Advantages	Disadvantages
Mean Filter	The principle of this filter is to replace the values of the pixels by calculating the average of their neighborhood	- Very simple and fast. - Very accurate on homogeneous surfaces	- The problem of the division by 9 not very robust because all the pixels got the same wight detail attenuation. - Edge degradation and blurring effect. - Strong influence of isolated aberrant pixels.
Gaussian Filter	The main idea of the filter is to calculate the weighted average of the neighborhood values, while giving a higher weight to pixels closer to the central pixel than to those further away.	- Configurable filter for a better ponderation (sigma) - Adapt to the problem. - Limits the blur effect (better preserved contours)	i) Complexity (floating point and non-integer computation) - Low attenuation of detail, but less than the mean filter
Median Filter	This filter is the most widely used non-linear filter in the literature. The principle of the filter is to replace the value of a pixel by the median of the values of its neighbors	- Increased noise resistance. - Powerful on salt and pepper noise and isolated pixels (impulsionnel). - Better preserves the contours without altering the background.	- Very slow, Expensive in computing time (sorting). - Removes fine details that are not noise. - Destroys corners.

or to reduce noise. Our choice is based on the advantages of a Gaussian filter over other filters. The latter consists in the fact that it is faster because multiplying and adding is probably faster than sorting, it eliminates noise without making changes to the edges and gives better results.

3.2.2 Module Motion Detection

According to Figure 3.1, the module of motion detection have a flow of data (images) as input. In order to avoid waste time processing them we ignore the images that doesn't contain objects or small objects (negligible) and we focus only on the images where there is an object which can be a car. Therefore motion detection is the detection of any object that makes a movement in field of view of the camera, there are several methods to have motion detection. Table 3.3 summarizes the most important of them.

In our system we've hybridized two of the most known methods: set background subtraction with temporary differentiation, and we tried to update the background every time there is no motion during 10 consecutive frames, by respecting the following diagram shown in Figure 3.5:

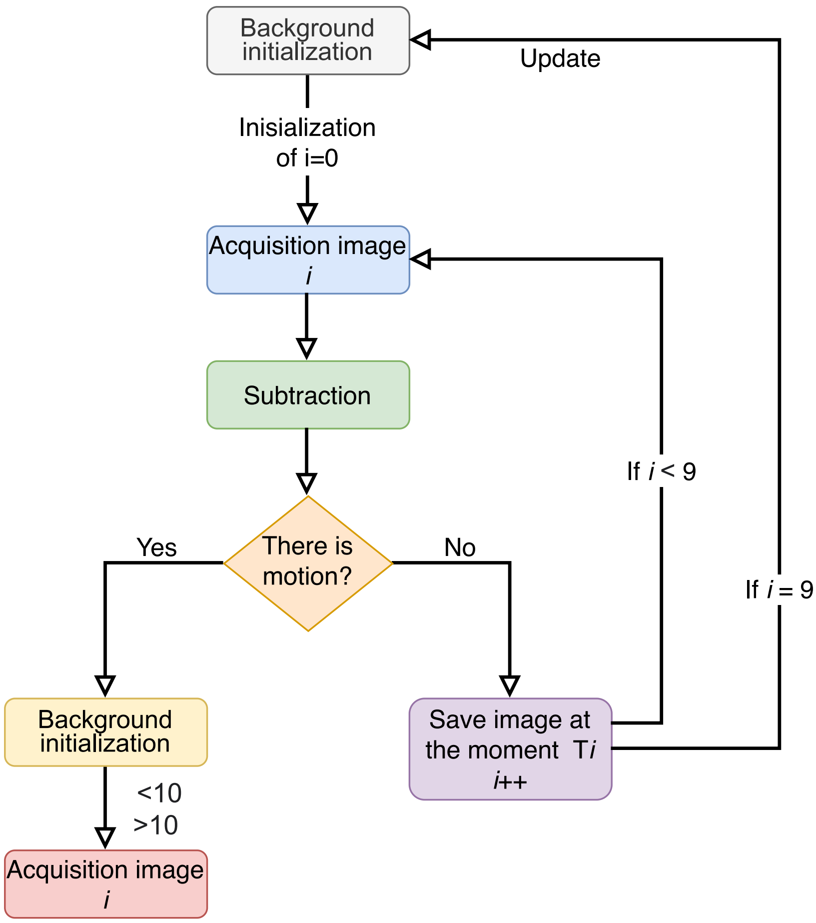


Figure 3.5 – *Conceptual diagram of the motion detection module*

- *Background initialization* : The background shot is initialized by taking the average of the different frames in different moments (close moments) to take into consideration the small changes that occur in

Table 3.2 – *Motion detection methods*

Object detection method		Basic principle	Precision / Calculation time	Comments	Image result
Temporal Differentiation		Pixel subtraction of the current image and the background image	High / Low	- Requires a background frame with fixed objects - Easy to implement - Sensitive to dynamic changes	image
Frame Differentiation	The current image is subtracted from the background image	Moderate to High / Low to Moderate	- Cannot be used for real-time applica-tions - Simple back-ground subtraction	image	
Optical Flow Uses		features of distribution of optical flow pixels of the object	High / Moderate to high	- Cannot be used for real-time applications - Simple background subtrac-tion	image

Table 3.3 – *Motion detection methods (continued)*

Subtraction background	Approximate Median	Simple subtraction between the middle frame and the test frame	Moderate / Low to moderate	- Requires a buffer with recent pixel values - No need for adequate background modeling	Image
	Gaussian mean running	Based on the Gaussian probability density function of pixels	Moderate / Moderate to high	- Statistical calculations take more time - Suitable for real-time applications	Image
	Gaussian mixture	Based on multi-modal distribution	Moderate to high / Moderate to high	- Cannot handle objects and noise - Low memory requirements	Image

the environment so that the subtraction result will be more accurate later.

- *Acquisition of image i* : before we start we initialize a counter i to 0, then we take different frames in a different moments i (in an average of 1 frame in each time the process ends)
- *Subtraction*: we apply the subtraction between the currant image in the moment T_i and the initialized one (initialized background) so we will get one of two result : there is a motion or not.
- *If there is no motion* : what does it mean that there is no change or negligible change that we do not take into consideration (there is not a detected object that appears in the field of view of our camera) so we save our image in the moment i , increment our counter. If $i < 9$, if k = 0 the system returns to the acquisition step. If $i = 9$ the system updates the background according to the formula (3.3) and initialize the counter to 0 again. Else we find any motion we confirm that it is a worthy or unworthy motion through sensitivity.

$$updatedbackground = \frac{((initialized\,frame \cdot 2) + \beta)}{3} \tag{3.3}$$

Where, β is the average of the 10 frames and is calculated by:

$$\beta = \frac{(framesT_0 + \cdots + framesT_9)}{10} \tag{3.4}$$

- *Sensitivity confirmation*: here we calculate the difference in the intensity of each pixel (between the background and the frame in the moment i when the change happened). If we found it more than 10 degrees we continue to sensitivity by zone, less than 10 the system returns to the acquisition step again.
- *Sensitivity by zone*: we calculate number of changed pixels. If is equal to a given value N we can say that our object is important (it may be a car) else back to acquisition step.

3.2.3 Morphological operations

Morphological transformations are simple operations based on the image shape. they are normally performed on binary images. they need two inputs, one is our original image, second one is called **structuring element** or **kernel** which decides the nature of operation. Two basic morphological operators are *Erosion* and *Dilation*. Then its variant form like Opening, Closing, Gradient, etc. We will see them one-by-one with help of following image:

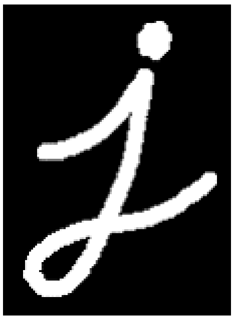


Figure 3.6 – *Image test*

3.2.3.1 Erosion

The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object (Always try to keep foreground in white). So what it does? The kernel slides through the image (as in 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero). So what happens is that, all the pixels near boundary will be discarded depending upon the size of kernel. So the thickness or size of the foreground object decreases or simply white region decreases in the image. It is useful for removing small white noises, detach two connected objects etc.[2]

➠ *Result*: Figure 3.7 shows the result of erosion when we use a 5x5 core with full of ones

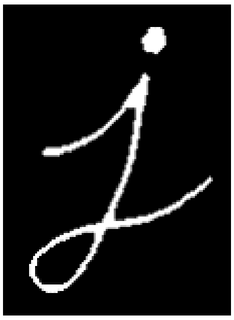


Figure 3.7 – *Erosion result*

3.2.3.2 Dilation

It is just opposite of erosion. Here, a pixel element is "1" if at least one pixel under the kernel is "1". So it increases the white region in the image or size of foreground object increases. Normally, in cases like noise removal, erosion is followed by dilation. Because, erosion removes white noises, but it also shrinks our object. So we dilate it. Since noise is gone, they won't come back, but our object area increases. It is also useful in joining broken parts of an object.[2]

➠ *Result*: See Figure 3.8.

Figure 3.8 – *Dilation result*

3.2.3.3 Opening

Opening is just another name of **erosion followed by dilation**. It is useful in removing noise, as we explained above.[2]

- *Result*: The application result on the image (Figure 3.6) is shown in the Figure 3.9.

Figure 3.9 – *Opening result*

3.2.3.4 Closing

Closing is reverse of Opening, Dilation followed by Erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object.[2]

- *Result*: The result shown in the Figure 3.10.

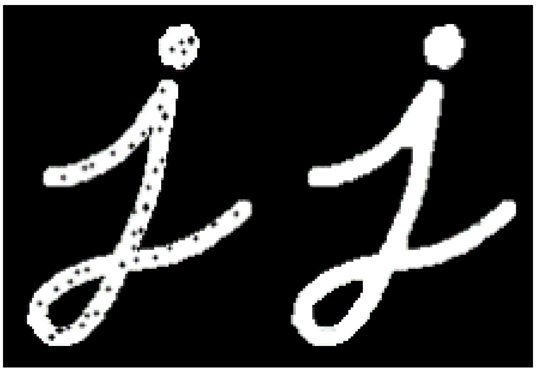


Figure 3.10 – *Closing result*

3.2.4 Edge Detection

3.2.4.1 Canny filter

Canny's method implements an estimation of the gradient of the image using the Sobel filter, followed by hysteresis thresholding of the gradient module. A high threshold and a low threshold are to be defined. All pixels where the gradient modulus is higher than the first threshold are classified as belonging to the contours of the image, contours of the image are thus formed. Pixels with a higher modulus at the low threshold and which are previously segmented are defined as contour points in the resulting binary image.[3]

Figure 3.11 – *Application of canny*

➠ *Canny Criteria*: the most important characteristics of the Canny criteria are the following:

- ✔ *Good detection*: all contours must be detected (without omitting some pixels on the contour to detect).
- ✔ *Good localization*: the detected contours must be in their ideal position.
- ✔ *Multiple Response Suppression*: a detector must not provide multiple responses, or false contours. For this, two thresholds are set, a high threshold (Sh) and a low threshold (Sb). On first select the points that exceed the high threshold and then apply the low threshold keeping only the related components that contain a dot over (Sh). In in other words from each point above (Sh) one "follows" a path made of points above (Sb), this path is the desired contour.

➠ *Canny Algorithm*: In the following points, we summarize the principle of the canny algorithm:

- ✔ Convert to gray scale
- ✔ Thresholding
- ✔ Gaussian filter
- ✔ The gradian (e.g. Sobel)
- ✔ Removal of non-maxima
- ✔ Thresholding then edge detection

3.2.4.2 Labeling

This operation is called analysis (or labeling) of connected components (connected-component analysis / labeling, or blob extraction). The principle of this technique is summarized in the following steps (see Figure 3.12):

- Starting from a binary image, find groups of connected pixels, called connected components or blobs.
- We obtain an image in which each "object" is identified.

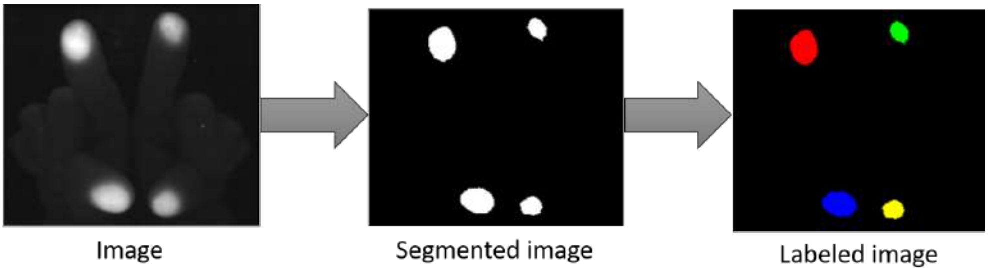


Figure 3.12 – *connected-component analysis / labeling module algorithm*

3.2.5 Plate Detection

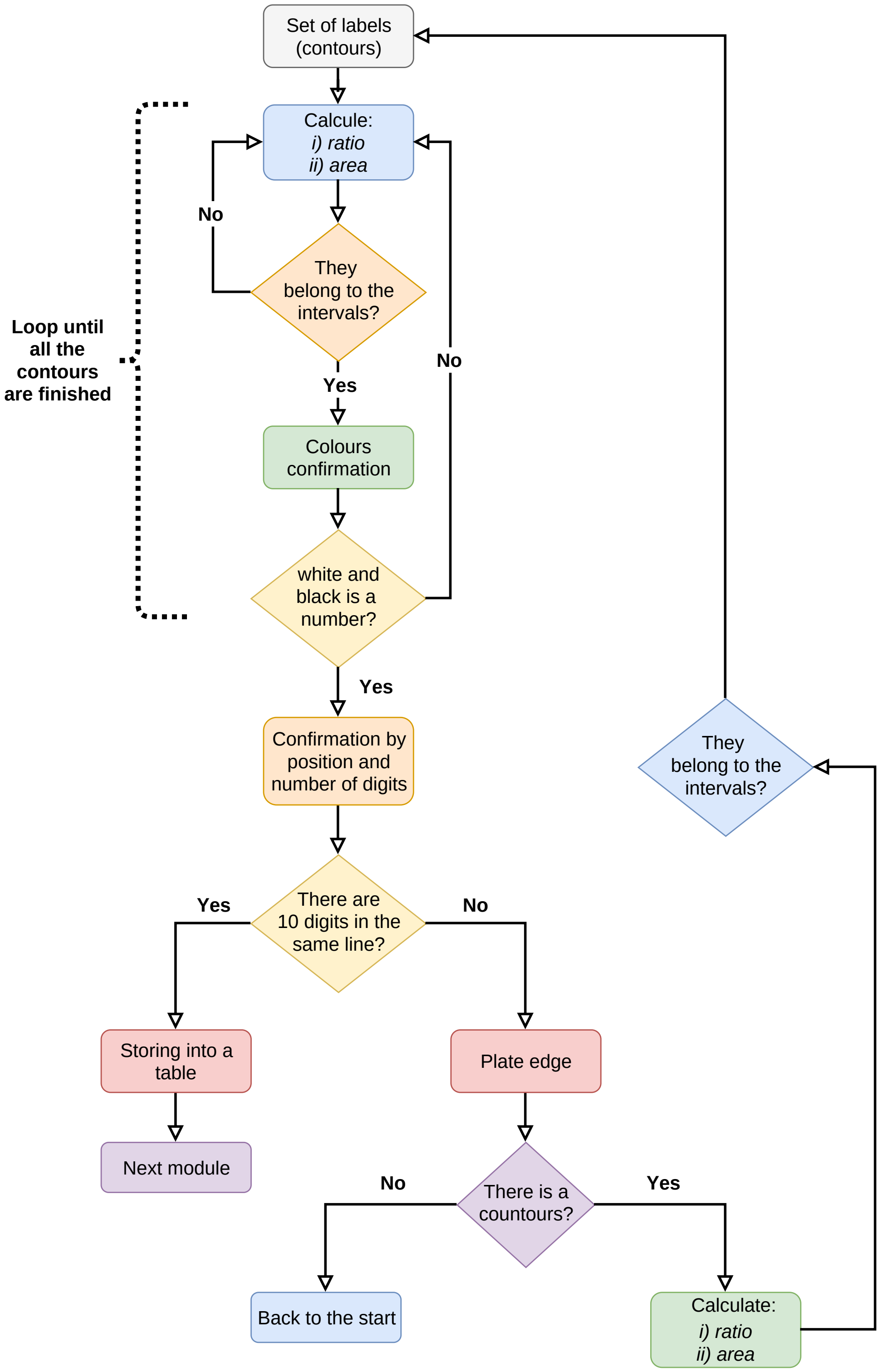


Figure 3.13 – *Conceptual diagram of the plate detection module*

In the rest of this section, we explain in detail the different steps of the plate detection module illustrated in Figure 3.13.

- for each contour we calculate :

- **ratio**: It is the ratio of width to height of bounding rectangle of the object.

$$AspectRatio = \frac{Width}{Height}[2] \tag{3.5}$$

- **area**: in our system we need to calculate area by definition that the area is the number of pixel in this object

- we do another filtering by the surface so that we eliminate the objects too big and too small (impossible that they are numbers)
- we initialize an interval from the minimum value to the maximum one of the two values (ratio, area)$[n \ldots N][n \ldots M]$, those intervals are of possible ratio and possible area for a different numbers form from 0 to 9.
- If the combination of the two results does not belong to those intervals we skip to next contour.
- Otherwise the combination belongs to the intervals, then the detected contours can be a number.
- To confirm if it is a number or not, we make a comparison with the stored copy of the initialized coloured background (original image).
- If it is any colour except black then it is impossible for it to be a number, we skip to next contour.
- Else if it is black so it is possible to be a number (there is a great chance) we do next confirmation with the placement of those numbers (they must be in the same line) we do it by comparing their coordinates if they have the same Y then they are in the same line we add them to table for recognition
- When all contours finished the result will be a table of contours which can be numbers.

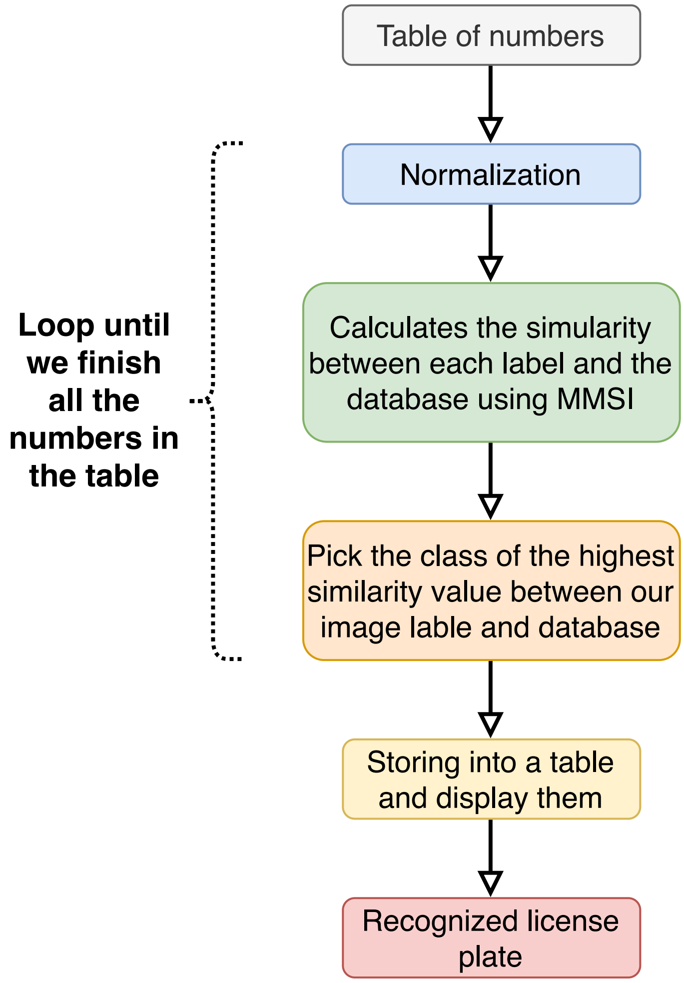


Figure 3.14 – *Conceptual diagram of the plate recognition module*

3.2.6 Recognition module

Figure 3.14 shows the different stages of the recognition module which are explained below:

- *Normalization*: it is the fact of cutting the images to a size compatible with that in the database to make it easier to compare.
  Calculate the similarity between each label and the data base using MMSI: we pic every label that we had from the plate then we do a comparison with all photos of each class

- **The structural similarity index measure**: The structural similarity index measure (SSIM) is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. The basic model was developed in the Laboratory for Image and Video Engineering (LIVE) at The University of Texas at Austin and further developed jointly with the Laboratory for Computational Vision (LCV) at New York University. Further variants of the model have been developed in the Image and Visual Computing Laboratory at University of Waterloo and have been commercially marketed.
  SSIM is used for measuring the similarity between two images. The

SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE).

- **Structural similarity**: The difference with respect to other techniques mentioned previously such as MSE or PSNR is that these approaches estimate absolute errors; on the other hand, SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or "texture" in the image.

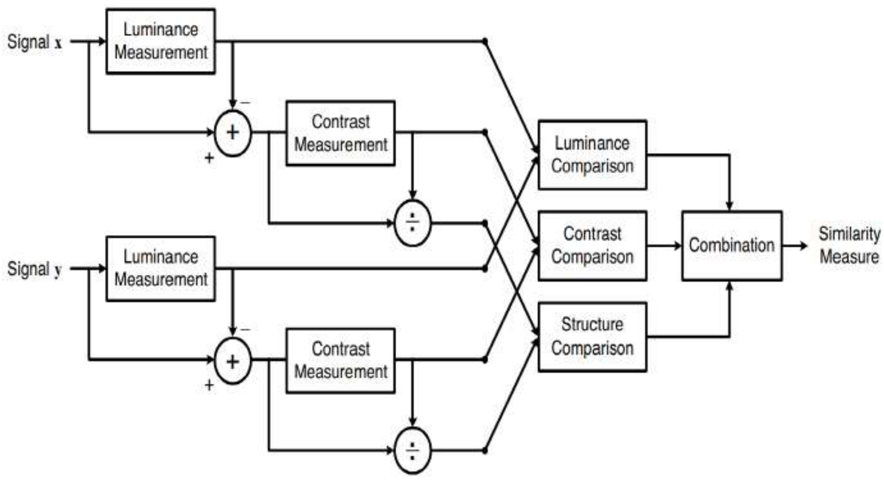


Figure 3.15 – *Diagram of the structural similarity (SSIM) measurement system*

Then we choose the class that have the highest value of similarity between our image label and the data base (class which the assemble photo belongs to)
We store the results into a table in order with their coordinates ($x_0 \dots x_{10}$)
Here we have reached our goal we have our recognized plate.

3.3 Conclusion

in this chapter we have represented the conception of our LPR system, either the different steps and methods that we followed starting from the acquisition to the recognition.

Implementation of our system

4

4.1 Introduction

In this chapter we are going to present and explain the practical part of our project which is a software based on the algorithms explained in the previous chapters. We are going to start by explaining the tools that we used (languages, packages, hardware, optimal configuration, camera, PC,... etc). Then, we will present our application's interface detailing its main functions. Afterwards, we will discuss the results of different day to day case scenarios that our application may encounter (bad lighting, car with non-standard colours, abnormal plate...). Finally, we will conclude the chapter with some conclusions.

4.2 Development tools

In order to implement and test our idea, we used several tool as mentioned below.

4.2.1 Hardware:

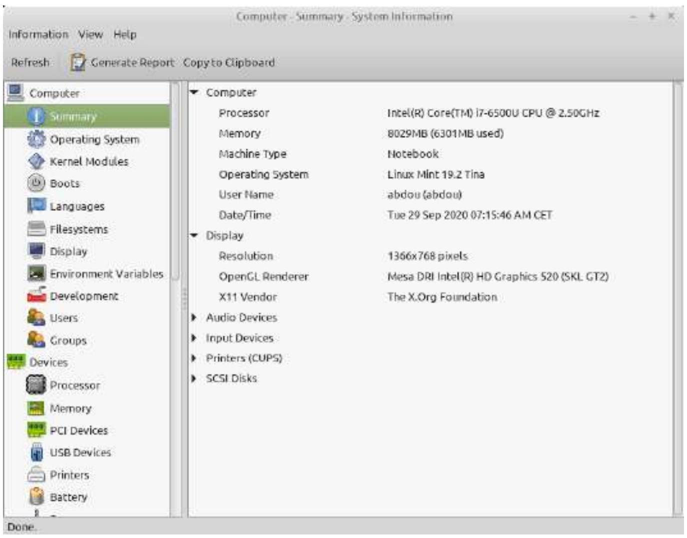


Figure 4.1 – *Characteristics of our computer used*

PC used

Camera F .3

Arduino Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

Servo motor :A servo motor is a rotary or linear actuator that provides high precision, high response control of angular or linear position, speed and acceleration. As a motor capable of accurately controlling the angle of rotation and speed, it can be used for a variety of equipment.

Ultra sound sensor An electronic component that allows us to detect if the car has passed the barrier or yet.

4.2.2 Software

Python3.7 Python is an object-oriented programming language with a multi-platform character. It is a powerful language that is both easy to learn and rich in possibilities. It is also very easy to extend existing features, so there are libraries that help the developer to work on particular projects. Several libraries can thus be installed.

Arduino C Arduino C is a programming language it is based on C and C++ . It used to programming Boards like (uno , nano , mega ...)

Visual studio code Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git

IDLE IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit.

Arduino The open-source Arduino Software make it easy to write code and upload it to the board. It runs on Windows, Mac OS X and Linux.

4.2.3 Libraries

OpenCv is an open source library for image processing and analysis and videos with interfaces for the main programming languages C,C++,JAVA,Python3...etc. It is especially optimized for applications in time real.
Among the functions it offers

- Image manipulation (loading, saving, copying ,converting) ;
- Manipulation and acquisition of videos ;
- Matrix manipulation and linear algebra;
- Image processing (filtering, edge detection...) ;

- Shape recognition (markove model, ACP);
- Graphical user interface (display of images, videos , event management);

MySQL The SQL language is a language universally recognized by MySql and other databases of data and allowing to query and modify the content of a database.

XML: (eXtended Markup Language) is a general text-oriented document format. It has established itself as an essential standard in information technology. It is used for document storage as well as for data transmission between applications. Its simplicity, flexibility and extensibility have made it possible to adapt it to a wide range of fields, from geographic data to vector graphics and trade. Numerous technologies have developed around XML, enriching its environment.

CSV The CSV module implements classes to read and write tabular data in CSV format. It allows programmers to say, "write this data in the format preferred by Excel," or "read data from this file which was generated by Excel," without knowing the precise details of the CSV format used by Excel.

Pyqt5 is a free module that allows to link the Python language with the Qt library distributed under two licenses: a commercial one and the GNU GPL. It allows to create graphical user interfaces in Python.

Skimage (formerly scikits.image) is an open-source image processing library for the Python programming language. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy

Datetime: module supplies classes for manipulating dates and times in both simple and complex ways

threading: This module constructs higher-level threading interfaces on top of the lower level thread module , Python threading allows you to have different parts of your program run concurrently

Numpy NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

OS This module provides a portable way to use operating system dependent features such as reading and writing inside a file

Serial Serial is a library for the Python programming language, which is an easy library that allows programmer to manipulate Arduino

4.2.4 Interface

4.3 Car detection

Our system's objective is to detect and recognize the license plate following a sequence of steps :

4.3.1 Background initialization

Our system will initialize the background from a set of frames like tho following one:

Figure 4.2 – *Initial background*

4.3.1.1 Update background

Due to the problems of light and the atmosphere changing, we decided to update the background because with the change of the weather, the colors change, thus reducing its effectiveness For this we have updated it according to the following equation

Normalization: In this step we are going to do a normalization in the colorimetric and scale. In the colorimetric level we will transform The image that is originally in color into a grayscale image with this expression :

cv.cvtColor()

One pixel is represented on 8 bits and the number of color components for each pixel will therefore be equal to 1.

In the scale level we will do a resize to make the picture in lower quality level to win time , because we do not need too much details in this step. these two transformations are done to get images in the same presentation (colour, size). The result will be a reduced image in size and color.

cv2.resize(image, size, interpolation = cv2.INTER_AREA)

Figure 4.3 – *Normalized Background*

4.3.2 Motion detection

(a) *motion detected(no object)*

(b) *motion detected(object)*

(c) *motion detected (car)*

Figure 4.4 – *Different frames*

The first thing to do is to apply the normalisation on this frames as we did with the background.

(a) *motion detected(no object),gray*

(b) *Motion detected(object),gray*

(c) *motion detected (car),gray*

Figure 4.5 – *Different frames normalised*

We do the subtraction with the background to detect if there is any movement (object) that interrupts our scene and we ignore the small changes (smaller than M)

((initialized background*2)+the new frame)/3

We do this update if there is no change in a number of consecutive images in order to keep the background always new

if (pixel[i,j]-background[i,j]> M) then result[i,j]=255

else result[i,j]=0

(a) *motion detected(no object)* (b) *motion detected(object)* (c) *motion detected (car)*

Figure 4.6 – *Different frames after Subtraction*

If there is an N pixel change in the picture (ex: 100 pixels changed from 500) ,in other words *a big change* , then we analyse the picture to figure out if it was a car or simply a man who passed by there.

4.3.3 Opening

Opening is just another name of erosion followed by dilation, we apply it to eliminate the useless noise (the holes) with the expression:

cv2.morphologyEx(image , cv2.MORPH_OPEN, kernel)

(a) *motion detected(no object)* (b) *motion detected(object)* (c) *motion detected (car)*

Figure 4.7 – *Different frames after Opening*

4.3.4 Closing

Closing is the reverse process of the opening, dilation followed by Erosion , we used the closing to connect separate objects like : many times we came across the problem where the bumper was separated from the car despite the fact that it is a single object ,we do it following the next expression :

cv2.morphologyEx(image , cv2.MORPH_CLOSE, kernel)

(a) *Motion detected(no object)* (b) *motion detected(object)* (c) *Motion detected (car)*

Figure 4.8 – *Different frames after Closing*

in both cases (closing and opening) we use a mask [3,3] with one iteration , We have chosen a small kernel because of the precedent preparation (quality minimization) so a big kernel can destroy details that we need or generate details we don't need

4.3.5 Existing motion

now we check if it's a car or not , to do that we calculate area for each labeled object and define the biggest one that have an area superior than a specific value N (must be a car)

Figure 4.9 – *Object detected as car*

The LPR system will crop the region where our object is in, so we don't spend a lot of time by processing all of the photo but just the pertinent information (processing information no need to)

Figure 4.10 – *object detected in real frame*

After we get our wanted result which is a cropped picture containing the object (must be a car) we pass to the plate extraction.

4.4 PLATE EXTRACTION

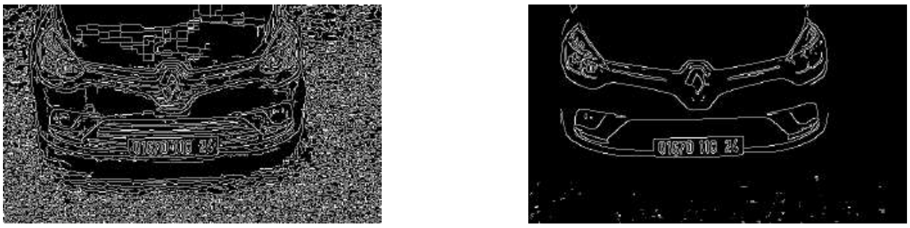
in this part we do Two ways to detect numbers plaque :

4.4.1 Extraction by detect numbers

4.4.1.1 Edges detection

We have chosen canny edge detection because it gives us better result : fine contours, not lined, separates objects correctly We have done two test with canny contours one based on black color with mask of (0,35), and the other one based on white (220,255) then we choose the second one (white base) because in the black base it gives us too much objects (so much unwanted edges,almost all objects in the nature have dark edges) so white base gives us only some edges content our wanted object

cv2.Canny(image,220,255)



(a) *[0 , 35]* (b) *[220 , 255]*
Figure 4.11 – *Canny Edge Detection*

4.4.1.2 elimination by area

Figure 4.12 – *Area elimination*

We applied an elimination for all objects having big area or too small, we select only objects with a specific area belongs to an interval (that can be a numbers)

4.4.1.3 contour extraction

After having detected the contours with canny , we perform the extraction of each one by the function :

findContours(image,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE) after that we extract the characteristics for each one by using the functions boundingRect ()

4.4.1.4 Elimination by the ratio

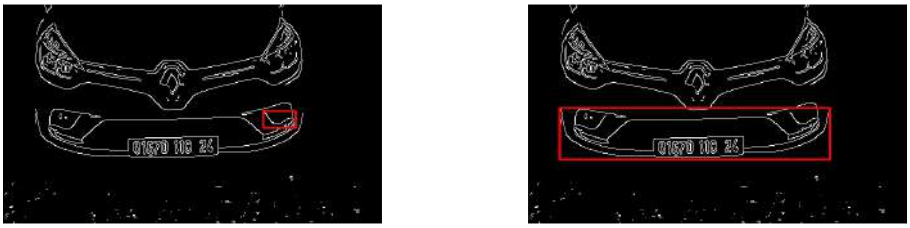


Figure 4.13 – *Ratio elimination*

We calculate the ratio for each edge after that we select only edges with a ratio that belongs to a defined interval (ratio interval of different number)

4.4.1.5 Elimination by colour

the input of this method will be the combination of the two past methods, then the detected contours can be a number. To confirm if it is a number or not, we make a elimination by colour by applying a comparison with a colored image (original image)

4.4.1.6 Position elimination and verification

Among the problems that we face is that the numbers in our lives are found anywhere, and some cars contain adhesive tape that contains the primary numbering plate as in the picture.

Figure 4.14 – *Position elimination*

And for this we resorted to excluding all numbers except for the number plate numbers (10 numbers at the same level of height)
We also make sure that the reading is correct and that we were able to identify all the plate numbers by checking the number of the plate numbers identified.

The result will be like the next images

Figure 4.15 – *Plate numbers extraction*

4.4.2 Extraction by detect numbers plaque

When we can't find the numbers, in the picture because in some cases they are internal edges not external, to solve this we apply canny filter for the same reasons that we said before, the only changes is that in the first step we look for the plate contour by looking for external edges.

Figure 4.16 – *Plate numbers extraction*

Once we found it we redo the precedent steps in the detection by numbers inside the plate, until we get our wanted result. But in this method we don't search for contour external cv2.RETR_EXTERNAL , but we search for all contours cv2.RETR_TREE
NB: We have tried search for the numbers in all counter always , but this way the time becomes very long and sometimes the process finds many things other than numbers and they suppose that it is number and this is the best solution to get it

4.5 Numbers recognition

In this part we do a comparison with data base to recognize the number in the image

4.5.1 Database creation

In order to identify the numbers in our LPR system, we need a database of Algerian plate samples, so we have made two database :

4.5.1.1 Database 1

We have scanned up the numbering plates of many Algerian cars in all possible forms (legal and illegal)

Figure 4.17 – *numbering plates of many Algerian cars*

We cropped manually every plate individually

Figure 4.18 – *Cropped plate*

Then by a program we converts the picture into grayscale, then We copy the picture and apply the canny filter on it to get the edges. We extract all contours by only focusing on external contours because we know that the number is not inside other object (plate). after that we apply an elimination the characteristic of the contour area. We are checking whether his area is limited to field (M, N) and whether he belongs to the field, we extract its characteristics through a boundingRect(), We cut out the location of this number in the gray image, and keep it with a serial number. Then we delete all wrong pictures if found, then separate them (each number in one file). Afterwards, we make a set of pictures of every number almost the same among the numbers, so that a number does not prevail over another. And we make all the pictures in the same resolution (60,40) to not lose time, and finally we applied a threshold with (170,255) to make it black and white only **This database contains 30 images for each number**

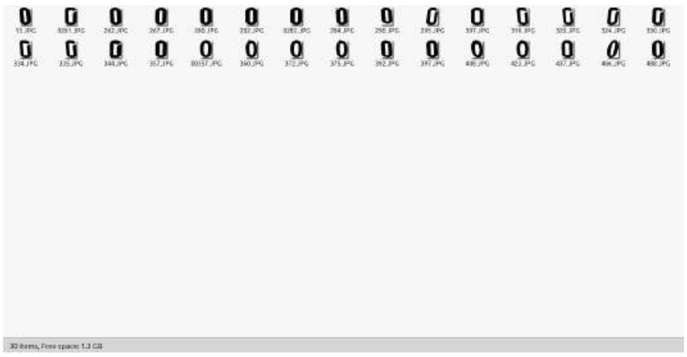


Figure 4.19 – *Database 1 size*

4.5.1.2 Database 2

We imported from the network a large database that contains pictures of the numbers, and we cleaned them according to the types of fonts used in Algeria, Then we crop and resize all images (make them in lower resolution)
This database contains 160 images for each number.

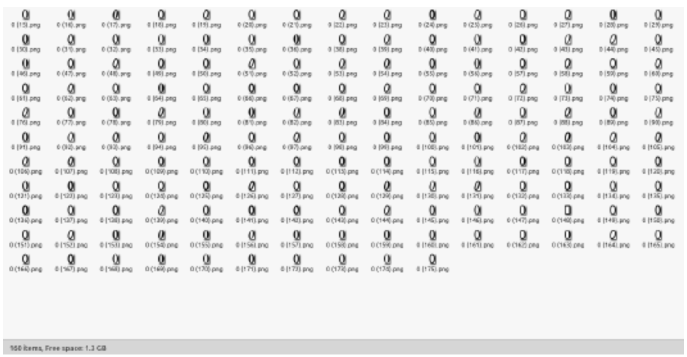


Figure 4.20 – *Database2 size*

NB : We can not leave a lot of pictures with high quality, to make the processing time short

4.5.1.3 CSV coding

Most of the world's databases are stored as CSV. It is also better than photo data base in order to reduce the number of access to the hard disk or encoding/decoding each picture, and that is to gain the time. So we made a program to turn them from photos to a single CSV file. We do it by : Browse all pictures and read them , convert every 2D image to 1D (linear image) Then we will add it in the CSV file with special information, which is the size of the image (resolution) and the number on the image, *meaning what information is in the image*, As in the picture below:

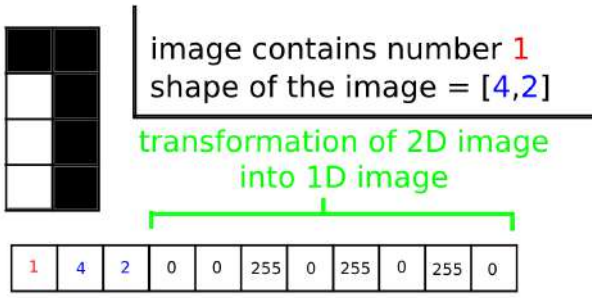


Figure 4.21 – *Example of converting an image for a CSV file*

In this way, we convert all the images and write them in one file.

4.5.2 Recognition

4.5.2.1 Preprocessing

After getting the coordinates of the numbering plate in the low-resolution picture. We crop them from the original image, which is in a good quality, because the recognition stage needs great accuracy. Then we turn them to gray-scale level , after that you have the option between using or without using thread *for gain time* .

4.5.2.2 thread

If you chose without thread we will identify the number one after another. But if you're choosing to use it, we'll identify all at the same time (in parallel).
The best one in time varies from processor to another

4.5.2.3 Identification of numbers

In order to identify the numbers, we change the size of the number image by the size of the picture in the database and we compare by SSIM algoritm.

The return is a number between 0 and 1 , The value closer to 1 is the most similar.
the return is a number confined between 0 and 1 and if the value was closer to 1 whenever the two photos are more similar.
This is how we search in the database about the picture that achieves the greatest analogy with the number picture from the numbering border.

4.6 Database management

In our system we make a communication with the system database, this database contains the customers information: plate number, first name, last name, phone number.
those customers can access to the space where we're using our system.

each time we have to detect the plate and recognize it, we check if he exists in our base or not *" in customer table "*. if it exists then he's able to access. otherwise the user is asked to enter it manually as a new client.

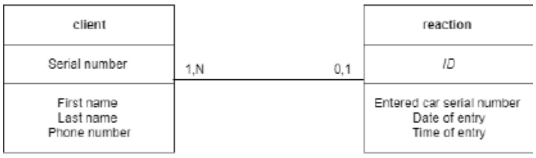


Figure 4.22 – *Database MCD*

4.7 our example project in reality

Our project can be used in a different field, any entry, so our process will start once a car is entered in the vision of the camera, step by step till we get our plate we confirm if he is already a client or not. if he is already client we allow the access directly, if he is not we send a request to add him manually. we detect the car is passed or not by a sensor, once the sensor detect that there is no car, that means that the car is passed we skip to the next car (close the access border).

Observations on the results obtained:

Figure 4.23 – *our project in reality*

4.8 RESULTS

4.8.1 among the successful experiment :

Figure 4.24 – *Correct tests*

4.8.2 Failed experiment :

it exist some cases where we weren't be able to to detect or recognize our plate like:

- some illegal plates contain some shapes on it

Figure 4.25 – *False test 1*

- Some of the paintings are illegal where the numbers are corrupt

Figure 4.26 – *False test 2*

4.9 System interface

Figure 4.27 – *interface 1*

Figure 4.28 – *interface 2*

(1) for starting the process

(2) stopping the process

(3) set of application parameters that allows us to manipulate the processing

(4) set of seuil to augment the results

(5) to display add client window

(6) to display manual insertion on clients that the process failed to detect

(7) to display delete clients window

(8) for the background updating

(9) add client window

(10) window of the clients manual insertion

(11) delete client window

4.10 Conclusion

In this chapter we have presented the implementation of our project with his different steps, also we presented tests that we have done and the results that we have reached.

GENERAL CONCLUSION

This dissertation, is concretized by the design and implementation of an Algerian Automatic License Plate Recognition system.

The realization of this project brought us two main things, First of all on the technical point, it allowed us to better understand and master the various technical tools such as the programming language python with the various packages necessary to generate a vision based system such as opencv, numpy, pyqt... and also the electronic part which is present in the opening mechanism of a door through a door controlled by arduino.

It also allowed us on the theoretical point to develop our team spirit, reinforced our know-how by forcing us to look for and find the necessary information especially for the notions of image processing, machine based vision and shape recognition that we never saw during our studies and that we have just discovered for the first time through this thesis.

So, the development of the proposed system was based at the beginning on the detection of movement of cars, followed by the extraction of the license plate using a method of contour and color detection. Then separated the characters of the license plate by segmentation and characteristics such as ratio, area, position, and orientation. Then the matching of the detected numbers of the plate with the models of the base is applied with the use of comparison by SSIM (structural similarity index measure). Finally, we end with a mechanical door opening reaction module with arduino and a sound sensor.

The results provided by the car license plate system have been tested in real time in different cases and with success, both in response time and accuracy.

As perspectives, we would like to

- Improve the results obtained by analyzing failure cases, and try to make the system effective in a complex environment.
- To apply our system on the University of Guelma's car park, to manage the access of people (teachers, employees, students and even visitors).
- improve the DBMS part of our system to better manage a real plate recognition scenario in a very specific problem such as the management of mulching for the access to the east-west highway, traffic management...etc.
- Introduce more electronic components to have a fully automatic system.
- Test machine based classification methods Learning to have a learning, and increase the accuracy rate for example by convolutional neural networks "CNN".

REFERENCES

1 :Régis CLOUARD, "TUTORIEL : DÉTECTER DES PLAQUES D'IMMATRICULATION", Équipe IMAGE - Projet Panthéon, 2007.

2 :https://docs.opencv.org/master/ consult the 03/09/2020 at 19:36

3 : Ameur CHHAYDER, Imene BELHADJ MOHAMED, « Système de Reconnaissance Automatique des Plaques Minéralogiques », 5th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, TUNISIA, March 22-26, 2009

4 :Lionel Lacassagne, Filtrage, Traitement d'Images, Institut d'Electronique Fondamentale

5 :https://www.hisour.com/fr/ycbcr-color-spaces-26075/consult the 10/09/2020 at 13:19

6 : Rodolpho Chabi A. BABATOUNDE, "Conception d'un système d'information de gestion du parc automobile béninois basé sur la lecture automatique des plaques minéralogiques", theme master, UNIVERSITÉ D'ABOMEY-CALAVI, 2018

7 : touaimia khaled, "reconnaissance automatique des plaques d'immatriculation", theme master, université de guelma, 2019.

8 : hachemi fouzia, "La détection et suivi des objets en mouvement dans une scène vidéo en utilisant la bibliothèque OpenCV", theme master, universite de telemcen, 2016.

9 : Pierre Chehwan, "Automating Background Color Removal with Python and OpenCV", Image processing basics, 2019.

10 : Yiğithan Dedeoğlu. Moving Object detection, Tracking and Classification for Smart Video Surveillance.

11 : Nils T Siebel. Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications.

12 : Qi Zang and Reinhard Klette. Object Classification and Tracking in Video Surveillance.

13 : Wilhelm Burger, Mark J. Burge: Principles of Digital Image Processing.

14 :Debbabi N, Seridi Z, Conception et réalisation d'un système de détection de mouvement : Application à la vidéosurveillance, PFE licence MTIC 2012.

15 : Heffaressas O, Belhouchi M, P Suivi d'objets dans une Séquence Vidéo en utilisant le Filtre de Kalman, PFE licence MTIC 2013.

16 : Ouertsi Kh, Khaldoum M, Réalisation d'un système de détection et de suivi de visage en mouvement pour l'application à la vidéo-surveillance, PFE licence MTIC 2014.

17 : Mesbah F, Amine, mise en place d'un système de vidéosurveillance et de contrôle d'accès par suivi de personnes : application au département d'informatique , PFE licence ISIL 2019.