University of SAAD DAHLAB -BLIDA -1

Faculty of Sciences

Computer Science Department

MASTER'S THESIS

In computer science

Specialty: Natural language processing

THEME :

Design and Implementation of an Intelligent & Real-Time Vehicle Access Control System

Authors:

MEKID Hamza & NACHEF Abdelkrim

Jury members

HIRECHE Célia, Professor assistant class B, Blida 1	President
MIDOUN Khadidja, Professor assistant class B,, Blida 1	Examiner
CHERIGUENE Soraya, Professor assistant class B,, Blida 1	Promoter
LALAOUI Omar, Application systems department head, Icosnet	Supervisor

2021/2022

Acknowledgments

We thank ALLAH for giving us the health and courage to be able to complete this project. This work is the culmination of a long journey during which we have benefited from the guidance, encouragement, and support of several people, to whom we would like to say a deep and sincere thank you.

First of all, we express our great gratitude to our supervisors Mm. cherigeune Soraya, Lalaoui Omar, and Brahimi Aimen for agreeing to supervise us and proposing this subject, for having been present with her advice, her help throughout this journey, and for the patience with which they have demonstrated. Their sense of responsibility and their professional rigor showed us the paths to follow to move forward.

Through these few words, we extend our warmest thanks to our dear parents and families for the support and encouragement they have always given us, also Nadjib Zennir, Korchi Chaima, and Oulfa Ezzeroug Ezzraimi for the help, as well as to all our friends who have contributed directly or indirectly to our success.

We want to give a special thanks to our professor Melyara Mezzi for the big help she gave us in all our master career and for her help in this thesis.

We would also like to thank the members of the jury for having done us the honor of judging and evaluating our work.

Dedications

I want to dedicate this Work to:

My parents for their love, and their continued support, present to them our most sincere feelings and our gratitude.

My dear Brothers Youcef, Imane, kholoud, and wassim for their constant encouragement and moral support.

All my friends and colleagues as well as any other person, Specially Rayen Beldjehem for her help throughout my master career.

I also want to remember Mrs. Mezzi Melayara, Ms. Bacha sihem, and Mrs. Cherigeune Soraya for their valuable information during my master degree.

For all; please accept My deep gratitude.

Mekid Hamza.

Dedications

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, Mahmoud and Djenet Razika whose words of encouragement and push for tenacity ring in my ears. and to my dear sister Yousra who has been always by my side.

I also dedicate this dissertation to my many friends and Mosque family who have supported me throughout the process. I will always appreciate all they have done, especially Cheikh kada abdelkrim, Bouaibibsa Hocine, and Dr. Aouali bilal for helping me develop myself and my soft skills.

I also want to remember Mrs. Mezzi Melayara, Ms. Bacha sihem, Mr. Mahdoum Ali, and Mrs. Cherigeune Soraya for their valuable information during my master degree, also Mr. Taha Zerrouki and Mrs. Ouahrani Leila, Mr. Karim Hemina for their help in my bachelor graduation.

I dedicate this work and give special thanks to Mr. Smail Mokhtar, and also Aissiou, Arkam, Mechehed, and Ben ali khouja families. for being there for me throughout my entire career.

Next, my dedications are to the ITCommunity members for their help and motivation in my last four years.

In the month of independence, Let me finish this dedication with my Algeria despite all the difficulties and problems, It was and it still my beloved country.

Nachef Abdelkrim.

Abstract

This thesis explains how to combine learning software and image processing to construct a system that can extract a license plate number from a video of a car taken with a camera and recognize the driver's face. This project was offered by Icosnet which wants to improve the efficiency of its staff's workflow.

We trained a convolutional neural network using different ways to detect objects, and we used the dataset we acquired to train it.

After that, we develop a software solution that allows us to control automobile access by utilizing our database of personnel with access to the workplace parking lot. we make that by recognizing the license plate of their cars or by recognizing their faces. This solution gives us a high accuracy in a short execution time.

Keywords: Image processing, License plate recognition, Access control, Face recognition, Convolutional neural network

Résumé

Cette thèse explique comment combiner un logiciel d'apprentissage et un traitement d'images pour construire un système capable d'extraire un numéro de plaque d'immatriculation d'une vidéo d'une voiture prise avec une caméra et reconnaître le visage du chauffeur. Ce projet a été proposé par icosnet qui souhaite améliorer l'efficacité du flux de travail de son personnel.

Nous avons formé un réseau neuronal convolutif en utilisant différentes façons de détecter des objets, et nous avons utilisé l'ensemble de données que nous avons acquis pour le former.

Par la suite, nous développons une solution logicielle qui nous permet de contrôler l'accès automobile en utilisant notre base de données du personnel ayant accès au stationnement du lieu de travail. Nous le faisons en reconnaissant la plaque d'immatriculation de leurs voitures ou en reconnaissant leurs visages. Cette solution nous donne une grande précision dans un temps d'exécution court.

Mots clé : Traitement d'images, Reconnaissance des plaques d'immatriculation, Reconnaissance de visage, Réseau neuronal convolutif

ملخص

تشرح هذه الاطروحة كيفية الجمع بين برامج التعلم ومعالجة الصور لبناء نظام يمكنه استخراج رقم لوحة ترخيص من فيديو لسيارة تم التقاطها بكاميرا والتعرف على وجه السائق. تم تقديم هذا المشروع من قبل الشركة ايكوسنات التي ترغب بحسين كفاءة سير عمل موظفيها.

قمنا بتدريب شبكة عصبية التفافية باستخدام طرق مختلفة لاكتشاف الأشياء، واستخدمنا مجموعة البيانات التي حصلنا عليها لتدريبها

بعد ذلك، قمنا بتطوير حل برمجي يسمح لنا بالتحكم في الوصول إلى السيارات من خلال استخدام قاعدة بيانات الموظفين لدينا مع إمكانية الوصول إلى ساحة انتظار السيارات في مكان العمل. نحقق ذلك من خلال التعرف على لوحة ترخيص سياراتهم أو من خلال التعرف على وجوههم. يمنحنا هذا الحل دقة عالية في وقت تنفيذ قصير.

الكلمات المفتاحية: معالجة الصور، لوحة الترخيص، التعرف على الوجوه، الشبكة العصبية التفافية

Table Of Contents

Table Of Figures

List Of Tables

List of Abbreviations

AI - Artificial Intelligence

RFID - Radio Frequency Identification

FPS – Frame per Second

ANN - Artificial Neural Network

OCR - Optical Character Recognition

CNN - Convolutional Neural Network

OpenCV - Open-Source Computer Vision

SSD - Single Shot Multi-Box Detector

HSV - Hue, Saturation, Value

SVM - SUPPORT VECTOR MACHINE

RGB - Red, Green, Blue

MLP - Multilayer Perceptron

VGG - Visual Geometry Group

mAP - Mean Average Precision

ReLU - Rectifier Linear Unit

ALPR - Automatic License Plate Recognition

SVM - Support Vector Machine

PCA - Principal component analysis

LDA - Linear Discriminant Analysis

GMM - Gaussian Mixtu

General introduction

General context

Protecting personal property is one of the most common problems that a person can face since ancient times. And every day trying to find new ways to enhance the protection of these properties, humans have developed some old methods such as doors and locks to prevent strangers from accessing. Recently this problem has become a great challenge for companies as the old methods were useless for the large number of people authorized to enter, so it was necessary to develop these methods so that we can control this huge number of people.

After the development of artificial intelligence in various fields, it was necessary to adapt these technologies to find solutions in this field.

One of the most common problems that companies have faced is controlling access to their car parks, as it is a special place to park their employees' cars and protect them from damage, allowing workers to focus on their work and not worry about their cars or belongings left in the car park, one of those companies is ICOSNET which is a communication solutions provider.

ICOSNET is an Algerian operator of broadband internet access, convergent telecommunications solutions, and Cloud solutions.

Created in 1999, and equipped with a multidisciplinary team, ICOSNET has been able to capitalize on significant experience and forge considerable relationships with the various players in the ICT sector on a national and international scale.

Problematic

Icosnet, and due to the large number of employees cars and their continuous increase, it has become impossible to control them, where cars can be damaged if unauthorized persons are allowed to enter. and its difficulties for human verification and tracking of all the cars entering, and it is expensive to hire security guards 24/24h. So they defined the problem of the management of their cars.

Objectives

Our work traits the access control security problem, so we divided it into two problems, the first problem is the recognition of an Algerian license plate numbers on real-time video frames, and the second problem is the driver face recognition. The first step in this part is to extract the face and check if it is a real face or a spoof to ensure a high-security level. Finally, we recognize if the person is authorized or not.

The biggest challenge is to realize those two problems with high accuracy and efficiently in a short execution time.

Thesis plan

In order to reach our objectives, we have organized our thesis into State of art and contribution, the first one contains 3 chapters:

Chapter 1 Automatic Access Control Systems: In this chapter, we will talk about access control systems with some comparisons between them, and some notions about image processing and deep learning.

Chapter 2 Automatic license plate recognition related works: this section is dedicated to explaining how the ALPR systems works and some used technics in similar projects.

Chapter 3 Facial recognition related works: this chapter explains how the Facial recognition systems works and some used technics in similar projects.

The contribution part is also divided into 3 chapters:

Chapter 4 Solution Modeling: In this chapter, we will go over the many design strategies that have been used to fulfill the goal that has already been stated.

Chapter 5 Automatic license plate recognition Implementation And Evaluation: We present the implementation of the different tasks of our system as well as the results of the models formed for the Algerian license plate recognition.

Chapter 6 Face Recognition implementation & evaluation And Interface presentation: We present the implementation of the different tasks of the facial recognition system as well as the results of the models. and at the final, we take a look at the user interface.

Stat of the Art

Chapter 01
Automatic access control systems

1.1 Introduction

This chapter contains access control systems and their importance to protect our private areas and our property, the process of restricting or controlling access to a resource is known as access control, and present some deep learning technics, especially technics used in image processing.

1.2 Access control

Authentification can be accomplished in a variety of ways; most of us will be familiar with using a username and password pair to sign in and authentificate ourselves to a computer [1] or other locations.

In the past few days hospitals, for example, businesses considered bringing people back to work, and regulating hospital entry ports was a top priority. Now that we understand the health implications of a worldwide pandemic and the role of busy workplaces as possible breeding grounds for sickness, healthcare facilities are debating how they may improve access management [2].

Access control systems have been around for a decade or more. But technology and innovation have gifted us with new tools for rapidly improving our safety and security response. Now, Artificial Intelligence (AI) works in the technology powering entryways with touch-free, highly secure solutions capable of learning and evolving to improve security continually. AI systems elevate efforts to protect people and assets in hospitals and other facilities [2].

1.3 Access control system

Access control is a security measure that restricts who can view, utilize, or enter a restricted environment. Access control systems can be found in the doors, key locks, fences, biometric systems, motion detectors, badge systems, and other security systems [4].

In Access Control Policies to comply with the principle of least privilege, an administrator must allow users based on a policy, often known as an access control policy or model, or a security policy. Different access control policies do not have to be mutually incompatible; they can be coupled to simulate more complicated requirements [1].

Furthermore, creating or planning effective security measures using these access control systems starts with understanding the principles involved. We need to break down access control security into its essential aspects to properly comprehend it, which include [4]:

- Identification: The primary aim of an access control system is to protect our building or a restricted area from unauthorized access. The access control system also regulates how people move throughout the building.

  Hence, we can use the access control system to determine the identification of everyone who enters the building or enters any of the secure or restricted areas.

- Authentication: Once an individual has been identified by the access control system, their identification must be verified. This will ensure that only the appropriate people have access to the building or location.

- Authorization: Building access control systems will be able to provide the individual entrance to the building or other secure or restricted locations if the system successfully authenticates the individual's identity.

1.3.1 Physical access control security

Physical access control can take many shapes, but the basic idea is to put up barriers to keep unauthorized people out of a physical place. To put it another way, physical access control ensures that only those who are permitted to enter a space can do so.

Physical barriers to physical access include closed doors, turnstiles, and fences, as well as authority barriers. A person or a sign outlining any restrictions is an authority barrier. Consider a restaurant with a sign on the door that reads, "Only employees beyond this point," a hospital receptionist admitting visitors during or outside visiting hours, or an electronic door in a corporation where only special workers have access to open it, as in (figure 1).

When we talk about physical access control, we're referring to a system of physical barriers that function in concert with authority barriers and authorization plans to let the proper people in while keeping the others out.

Physical access control is right up there with digital security in terms of importance in any solid security plan. While digital security safeguards data, which can be exploited to harm a company's or individual's reputation, finances, or performance, physical security safeguards people and equipment in a more tangible sense. The evening news is full of stories of the unintended consequences of a violent individual carrying a weapon into a workplace. Physical barriers can prevent ignorant people from walking or entering server rooms or other locations with hazardous products and equipment in a less dramatic manner [4].

Figure 1 Access control using an electronic door [4].

1.3.2 Technical access control security

User access defines data sensitivity levels and assigns user clearance levels accordingly. Use the space in our access control plan to define rules for setting passwords and specify technical aspects of wiring, routers, permissions, and user access control [4].

Network encryptions and IT security protocols. Despite its user-friendliness, our plan needs to include methods for Certain access control plan examples to tackle this in a comprehensive document; others restructure the information in appendices for trained IT staff [5].

1.3.3 Administrative access control security

Administrative security is Important to secure the property within the company and keep the administrative resources safe.

It is mandatory to make a plan for Access Control that works. The strategy is an essential component of any company's security. Make sure it contains rules for employees at various risk levels, incident response and recovery processes, and procedures for access control system maintenance, upgrades, and audits.

This aspect of role-based security needs to reflect the distribution of authority within the company, state roles, and link associated duties and responsibilities. Identify staff responsible for overseeing the plan, supervising the set controls, and enabling training for staff and testing of the equipment [4].

1.4 Background and motivation

The access control system is very important in our days and in this section, we will explain how can an access control system assist us, Automated access control, for example, can safeguard employees while also letting administrators know who enters the building. The following are the most important advantages of access control systems [6]:

- simplify the access for employees
- keep a record of every person access.
- money and energy savings
- protect from unwanted visitors
- prevent information breaches
- create a safe work environment
- reduce theft and accidents

1.5 Vehicle access control system

Vehicle Access Control System can control the entry of a vehicle in private places like parking and garages and so many other places. A vast majority of the currently installed systems use barrier gates that are manually operated but in our days the Vehicle access control predates Radio Frequency Identification (RFID) access control in general [7]. In the early 1980s, magnetic stripe swipe cards and later contactless RFID badges were introduced, kicking off the present access control sector. Manufacturers, integrators, and clients have debated the optimum way to regulate vehicle access to estates, sites, and parking lots since the early days. Automatic License Plate Recognition (ALPR) [8] is one of the most popular techniques of vehicle access control these days. This technology allows for reading distances of over 10 meters, making it a convenient and reliable method of automobile access control.

However, after a decade of stagnation, it appears that change is on the way. New technologies is knocking on our door, forcing us to reconsider our approaches to vehicle identification that are both convenient and secure [9].

1.5.1 Vehicle access control systems comparison

Hereafter, we summarize in Table 1 the principal advantages and disadvantages of each technique.

Table 1 Vehicle Access Control Systems comparison.

Systems	Advantages	Inconvenient
Security Guard	1. Heightened Security and less possibility of mistakes 2. It provides a Quick Response Time	1. More than one guard is needed to keep or place secure all the day 24h 2. A lot of money is needed to hire more than 2 guards 3. a guard is a human who can be tired or sleep during the night shift and this decreases the security
RFID [7]	1. Rapid charging/discharging 2. Simplified patron self-charging/discharging 3. High reliability 4. High-speed inventorying 5. Long tag life 6. Fast Track Circulation Operation	1. High cost 2. Vulnerability to compromise 3. Removal of exposed tags 4. Evaluating RFID from different vendors 5. Security feature 6. Tag memory capacity 7. Tag functionality 8. Cannot work if there is a power outage

ALPR	1. Money and energy savings 2. The Provide a Quick Response Time 3. easy of manipulation of the system 4. possibilities of adding or deleting any license plate easily 5. high distance detection and recognition process	1. possibility of making mistakes 2. possibilities of algorithm problems 3. Cannot work if there is a power outage

1.5.2 IA and ALPR

Nowadays, we have witnessed a big revolution of Artificial Intelligence in many aspects of everyday life, and especially in the computer vision domain, we can see that IA gives a good result in applications systems nowadays, this is due to the development of deep learning algorithms, it prompted many people to consider how mankind may benefit from it. So, peoples start building different AI systems to facilitate daily work and, in this thesis, we will explain how we build AI algorithms to do Pattern recognition to know license plate cars numbers to manipulate the access control of vehicles in some private places like companies' parking or home garages, to increase the security of those places and keep people who we don't need to be there away, all of that without spending a lot of many.

1.6 Image processing

The domain of image processing corresponds to the digital processing of an RGB image (multy colors) like patterns recognition and features extraction by an algorithm, is discussed in this section [10].

1.6.1 RGB

RGB stands for Red, Green, and Blue, and is a color space named after the three main colors. This model can be shown as a three-dimensional cube, with each point's coordinates corresponding to the red, green, and blue component values. Furthermore, while this model is suitable for displaying color, it is insufficient for color analysis due to the significant correlation between the components, i.e., the three components (red, green, and blue) vary in response to changes in intensity [11].

Nrgb is a normalized variant of RGB that is less sensitive to changes in illumination.

$$r = \frac{R}{R+G+B}, \quad g= \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \text{ [10]}$$

1.6.2 Features extraction

Hough Transform and Template Matching are two approaches for extracting features from images covered in this section.

The Hough Transform is a feature extraction method for finding shapes in images. This is a patented technique for detecting circumferences, ellipses, and lines in images [12]. To detect the presence of a given pattern within an image, this approach maps the points of the image into an accumulator space.

Template matching is the process of calculating the degree of similarity between each template and the target image. The classification of the target image with the class of the template with the highest similarity, or the location of the latter in the image, is the result of this approach [13].

1.6.3 Image segmentation

In image processing, image segmentation is a critical step. The segmentation methods are based on two properties: discontinuity and resemblance [14], and they are used to divide an image into separate regions of interest. (Figure 2) depicts some of the image segmentation approaches that will be covered in this section.

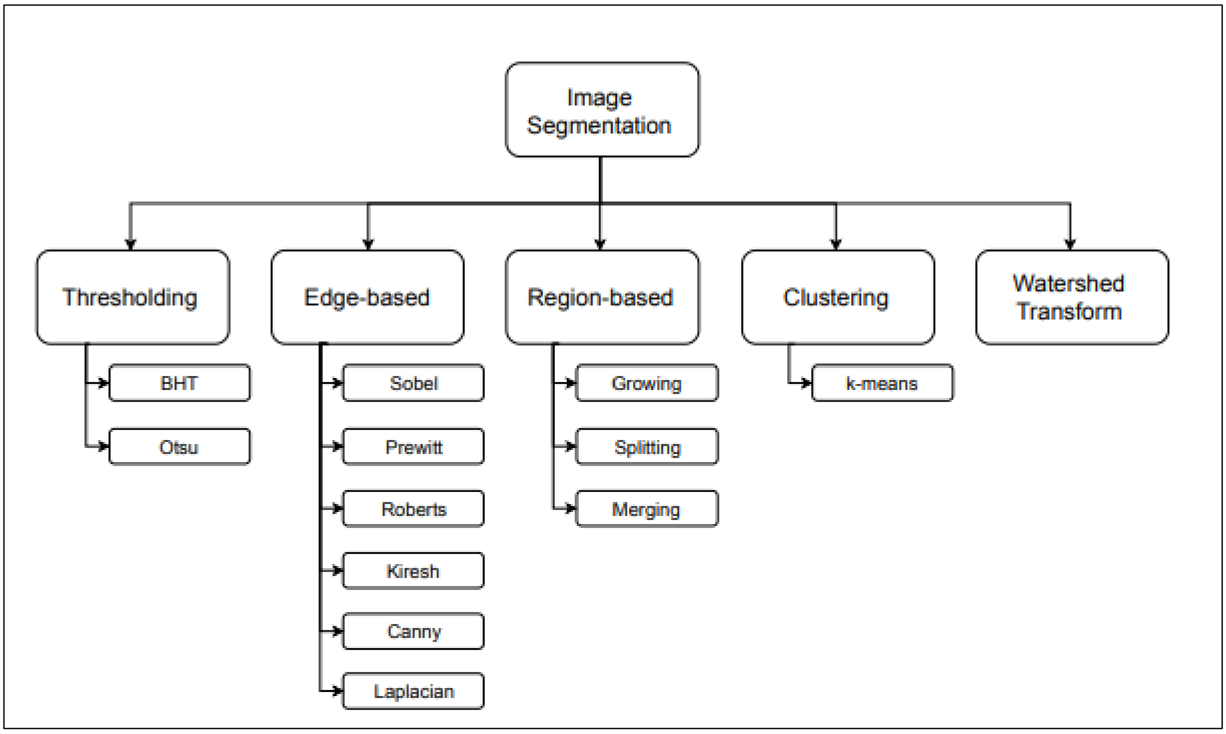


Figure 2 Image segmentation techniques [10].

a) Thresholding histogram

Thresholding is an image segmentation technique that divides an image's histogram into peaks corresponding to distinct regions and calculates a threshold value that corresponds to the valley between two different peaks [10].

Several approaches for thresholding segmentation based on histograms are utilized, including:

- The Balanced Histogram Thresholding (BHT) approach determines the best threshold level for dividing the picture histogram into two classes: foreground and background. To balance the histogram, BHT determines the weight of the histogram, confirms which class is heavier, and removes weight from the latter until it becomes lighter [15].

- The Otsu technique divides pixels into two classes: background and foreground. This threshold is determined by maximizing the variance between classes [16].

b) Edge-based

Edge detection-based segmentation is an image segmentation method that detects discontinuities in natural image features such as texture, color, and gray levels to locate boundaries [14]. Sequential and parallel edge detection techniques are the two types of edge detection techniques. A dependent relationship between the outcome at one point and the results at previous places is indicated by sequential detection. Parallel detection, on the other hand, shows an independent relationship between the decision of a set of points being or not being on an edge and the decision of other sets of points lying or not lying on an edge [11]. Sobel, Prewitt, Roberts, Kiresh, Canny, and Laplacian are some of the currently known edge detectors.

c) Region-based

Image segmentation methods based on regions, such as region expanding, region splitting, region merging, and their combinations, aim to group pixels into areas [11]. The region growth approach begins with the selection of a reference region, which is then enlarged to include its neighbors who meet a similarity criterion, resulting in a cluster. This technique is repeated until the entire image has been covered in pixels. A method in which the first seed region corresponds to the entire image is known as region splitting. If the seed region is not homogenous, it is then divided into four sub-regions, each of which becomes a new seed region [11]. This method is repeated until homogeneity is achieved in all subregions.

d) Clustering

Clustering is the division of picture content into groups or classes in image processing.

K-means is the most often used clustering algorithm, which separates a picture into k clusters. Some clusters' centers are placed in specific positions, each pixel is assigned to the cluster with the closest center, and the center of each cluster is recalculated using the average of the pixels within the cluster. The two preceding stages are performed until no pixels are swapped between clusters [17].

e) Watershed transform

The watershed transform is a segmentation technique that treats the image as a topographic map, with each pixel's intensity representing its height [18]. The goal of this strategy is to discover places that are so close together that their edges touch. Assuming two catchment basins are being filled with water, when they are both full, the locations where floodwater from distinct basins meet are recognized, and pixel barriers are built in these regions, partitioning the image in a similar way to how this method works.

1.6.4 OpenCV contours

Contours are curves that connect points with similar attributes like color and intensity to create an outline that surrounds a specific item. This approach is quite effective for object detection and recognition, and it performs better on binary images (black and white images) [10]. The contours obtained in an image are shown in (Figure 3).

Contours contain a variety of helpful techniques and properties in OpenCV, such as getting a contour area, perimeter, moments - spatial, central, and central normalized - and the respective bounding box that may be used to extract a Region Of Interest (ROI).

The parent-child connection, often known as a hierarchy, is one distinctive element of curves. When searching for contours in a picture, some of the contours found may be nested, with the outside contours being the "father" and the inside ones being the "children." The hierarchy of contours is defined by their relationship [10].

Figure 3 OpenCV contours on a image [10].

Figure 4 shows numbered forms ranging from zero to four, with 2 and 2a indicating the corresponding rectangle's external and inside outlines. Contours 0, 1, and 2 are in hierarchy level 0 according to the image, whereas contour 2a is at the next hierarchy level (level 1) and is regarded as a child of contour 2. (or contour 2 is its parent). Finally, contours 3 and 4 are both children of contour 2a and belong to the last hierarchy level (level 2).

The contours hierarchy in OpenCV is represented as a four-valued array: [Next, Previous, First Child, Parent]. Next denotes the next hierarchical level contour, Previous denotes the previous hierarchical level contour, First Child denotes the first child contour, and Parent denotes the parent contour [10]. The indexes of the contours are represented by the values in the array.

The Contour Retrieval Mode controls how the hierarchy is calculated across the contours (CRM). OpenCV offers four types of CRM [10]:

1. LIST: Obtains all contours without regard for their relationships with one another.

With this mode, the concept of parent and kid does not exist.

2. EXTERNAL: Only the outer contours are retrieved, ignoring the child's contours.

3. CCOMP: All contours are obtained and sorted into a two-tier hierarchy. Level 1 contours are used on the outside, whereas level 2 contours are used on the inside. If any inner contour (level 2) has another within it, the first is placed in level 1, the second in level 2, and so on.

4. TREE: All contours are extracted, and a complete family hierarchy list is constructed.

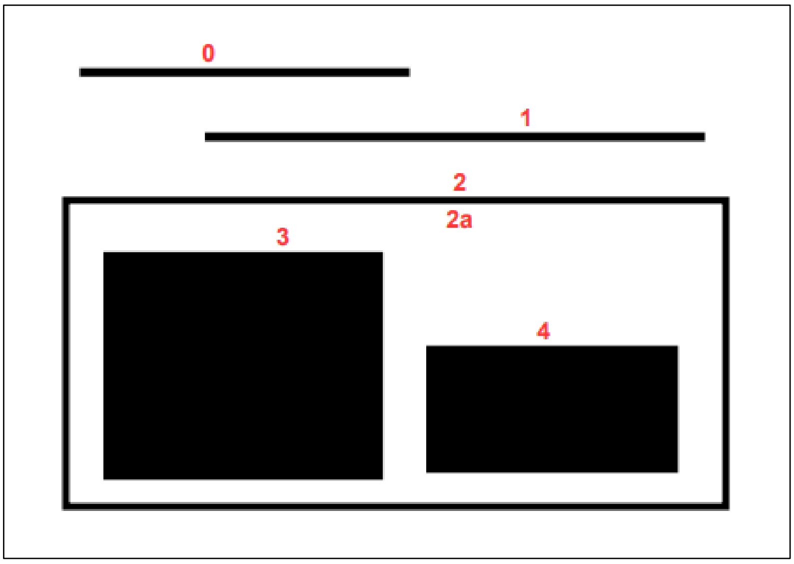


Figure 4 Illustrative example of contours hierarchy [10].

The last mode (TREE) is particularly intriguing because it allows for filtering operations to discover relevant contours using the whole hierarchy produced. In reality, this technique is utilized.

Table 2 Examples of contours hierarchy arrays [10].

Contour	Index	Hierarchy array
0	0	[1, -1, -1, -1]
1	1	[2, 0, -1, -1]
2	2	[-1, 1, 3, -1]
2a	3	[-1, -1, 4, 2]
3	4	[5, -1, -1, 3]

4	5	[-1, 4, -1, 3]

A short examination of the above table reveals that if only the child contours of the contour 2a are desired, all that is required is to check the equality of the final element of the hierarchy arrays of all other contours with the index of the contour 2a.

1.7 Artificial neural network

Artificial neural networks (ANN) are processing systems that are heavily influenced by biological nervous systems. They are non-linear classification and regression models that are quite powerful. ANNs are generally made up of a vast number of interconnected computational nodes known as neurons, whose work is spread to learn from the input and maximize the final output [19].

Figure 5 illustrates the basic construction of a neural network.

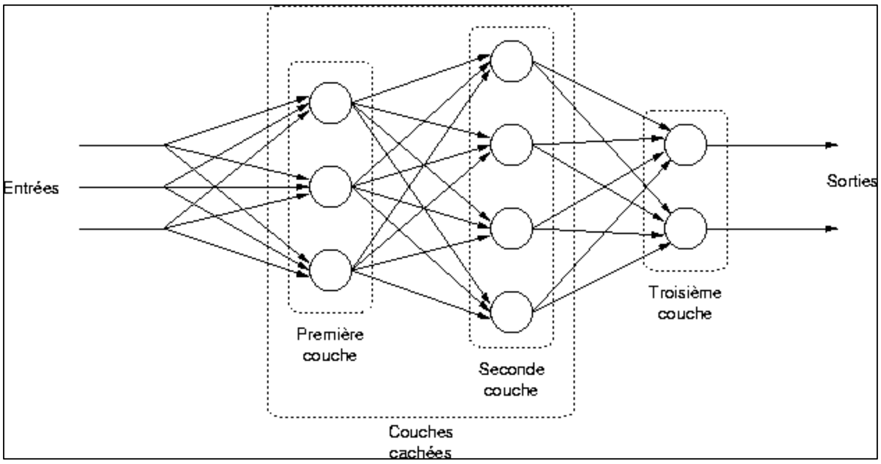


Figure 5 Artificiel Neural Network [19].

The input is typically a multidimensional vector of the input layer, which serves as the distribution for the hidden layers. The learning process involves the hidden layers making judgments based on the previous layers decisions and evaluating how a stochastic modification enhances the final output on its own. Deep learning is defined as having numerous hidden layers piled on top of each other. Forward propagation and reverse propagation are two crucial phases in artificial neural networks.

1.7.1 Forward propagation

This phase, as the name implies, is directed forward in the depth of a network, or from left to right, with no way to return to prior layers. The purpose is to get an output S that represents the effort done [20].

Forward propagation goes through a few processes to make this phase operate. These procedures, such as pre-activation and activation, will be carried out at the level of the hidden layers and the output layer. These two steps are carried out on each neuron that is placed on the hidden layers and the final layer. These specified steps will be discussed in the following sections [20].

1.7.2 Pre-activation

At the neuronal level, it constitutes the first stage of forward propagation. It entails ensuring that the values of each input connected by a hidden unit X are multiplied by the weight between the two (the input and the hidden unit). After multiplying all of the entries by their weights, the sum of the acquired results is determined by adding a static value known as bias is the mathematical expression for the pre-activation function [21].

$$V = (Wi\ Xi) + \text{bias}$$ [21]

1.7.3 Backpropagation

Neural networks learn by backpropagation (Figure 6). It is critical to comprehend not only the theory but also the mathematics underpinning backpropagation. This is one of our material's few mathematically rigorous portions.

Backpropagation is the step of learning a model, and it operates in general as follows. When forward propagation is complete, an error rate must be calculated, as shown in the next diagram. This error rate is determined by the model's final outcome and the expected result. After then, the model is run in the other direction (from the exit to the entry) [19].

The goal of this path is to lower the calculated error rate by adjusting the model's parameters. Optimization methods based on derivatives will be utilized to alter the parameters.

The error functions, optimization strategies, and parameter adjustments are all covered in the next section [19].

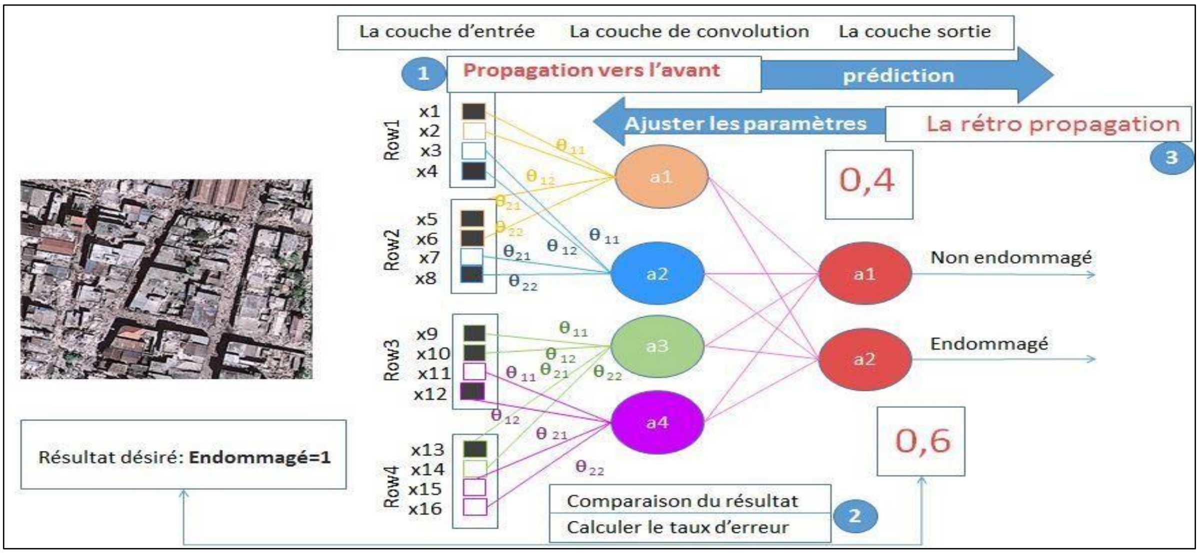


Figure 6 Back propagation [19].

1.7.4 Activation function

Activation functions are mathematical equations that provide meaning to a model's classification. These functions can be engaged or deactivated based on the relevance of the neuron, which is determined by the pre-activation value; in either scenario (function activated or deactivated), the activation function outputs the neuron's final outcome. The outcome varies based on the function and its state. In all of these phases, the activation functions are critical to the neuron's functioning; selecting a suitable activation function provides a good prediction result, rapid model parameter modification, and faster training [19].

1.7.5 Type of activation functions

Artificial neural networks use a variety of activation functions (Figure 7), the most common of which are [19].

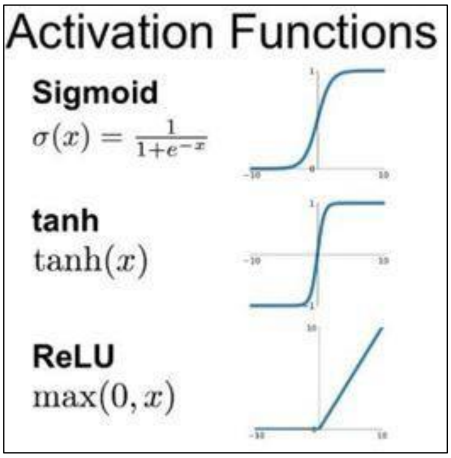


Figure 7 Some type of activation functions [19].

a) The sigmoid function

It is the oldest non-linear activation function, and its mathematical representation is [22]:

z = the outcome of the pre-activation

This function is graphically represented by

The derivative of the sigmoid function will be employed in back-propagation. This derivative is defined mathematically and graphically as:

$$F'(z) = F(z)\ (1\ F(z))\ [22]$$

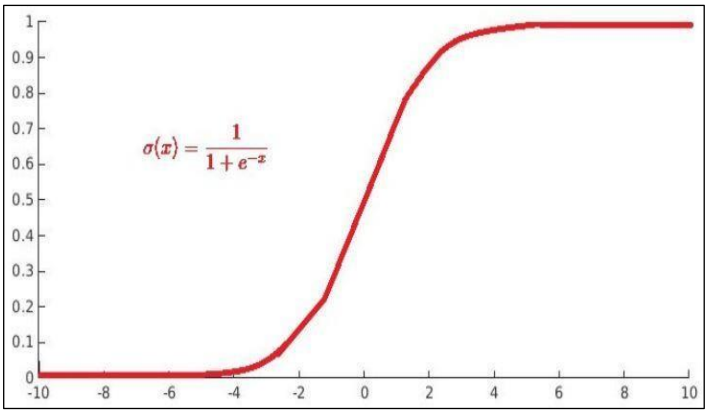


Figure 8 Sigmoid function [19].

b) The ReLu function

The most commonly used feature nowadays. According to [23], the rectified linear unit's activation function is to provide an output of value x to inputs of positive values x, or output of value 0 to inputs of negative values [23].

In contrast to the two previous functions, ReLu allows for faster learning because, as demonstrated in the mathematical representation, the derivative of positive values equals 1, which is beneficial to learning. The derivative of negative values, on the other hand, produces a value of 0. (Like the first two functions) [19]. And represented graphically by:

$$\mathbf{F}(\boldsymbol{z}) = \mathbf{max}(\mathbf{0}, \boldsymbol{x})$$

$$(\boldsymbol{z}) = \begin{cases} 0\ Si\ x\ <\ 0 \\ 1\ Si\ x\ \geq\ 0 \end{cases}$$

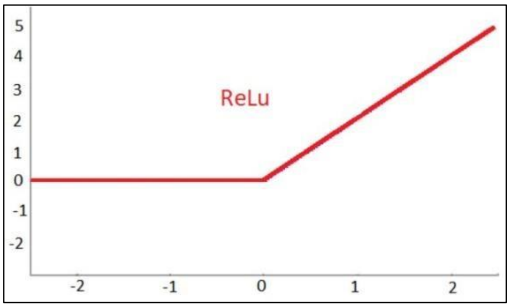


Figure 9 ReLu function [19].

c) Softmax function

To make a classification, the softmax function is applied to discrete or continuous data. It assigns a probability of prediction between 0 and 1 to each of the output's existing classes, with the sum of the probabilities equaling 1, and the chosen class being the one with the highest percentage of likelihood [24].

Unlike the sigmoid function (Figure 10), the Softmax function can be applied to a variety of classes (the latter is used in binary classification).

The Softmax function is mathematically represented as follows:

$$\sigma(\vec{z})i = \frac{e^{z_i}}{\sum_{i=1}^{K} e^{z_j}} \text{ [25]}$$

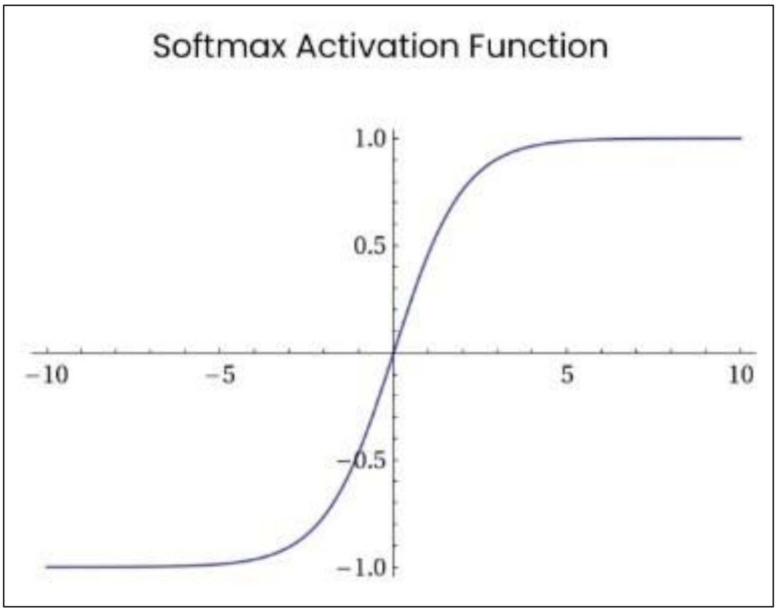


Figure 10 Softmax graph [25].

The error rate is calculated using the value of the desired result and the result obtained, and the loss function (Figure 11) communicates the model's performance. The loss function returns a very large value if the forecasts stray too much from the actual findings [26].

On the other hand, the closer the loss value is to zero, the lower the error rate, and the model's output will be more efficient and realistic.

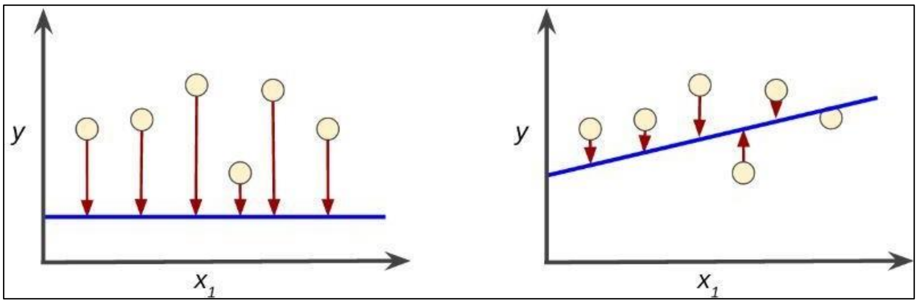


Figure 11 Representation of loss function [19].

Several loss functions can be divided into two groups (regression loss functions, and classification loss functions).

- Cross-entropy loss

The performance of a classification model whose output is a probability between 0 and 1 is measured by cross-entropy loss, also known as logarithmic loss. As the sample's expected probability differs from the actual value, cross-entropy rises. [26]

The following equation describes this function:

$$\boldsymbol{CE} = \sum_{j} \boldsymbol{y}(\boldsymbol{j}) \log\hat{}(yj) \ [19]$$

1.8 Convolutional neural networks

The Convolutional Neural Network (CNN) is a deep learning architecture that was inspired by live beings' innate visual perception mechanisms [19,27].

A neural network with a fully linked convolution block and layers is referred to as a CNN. Convolutional layers, activation functions, and down sampling layers are all found in the convolution block (pooling layer). Pattern recognition, object detection, image classification, semantic segmentation, and other tasks are performed using convolutional neural networks.

The greatest advantage of CNNs is that they reduce the number of parameters in artificial neural networks (ANN). This achievement inspired researchers to consider larger models to address challenging problems that could not be solved with traditional ANNs. The most fundamental assumption concerning CNN-solved problems is that they should not contain spatially dependent features [28].

1.8.1 Filters

Filter, often known as kernel or mask, is defined. As a two-dimensional matrix that enables the suppression of high frequencies in an image, i.e. picture smoothing, or the suppression of low frequencies, i.e. edge detection. Filtering involves processing the original image and creating new images based on those values as each pixel is treated. The new image is determined by the filter applied as well as the original image [29].

1.8.2 Stride

Stride is a component of convolutional neural networks and tailored neural networks for image and video compression (Figure 12). It is a neural network filter parameter that alters the amount of motion in an image or video [30].

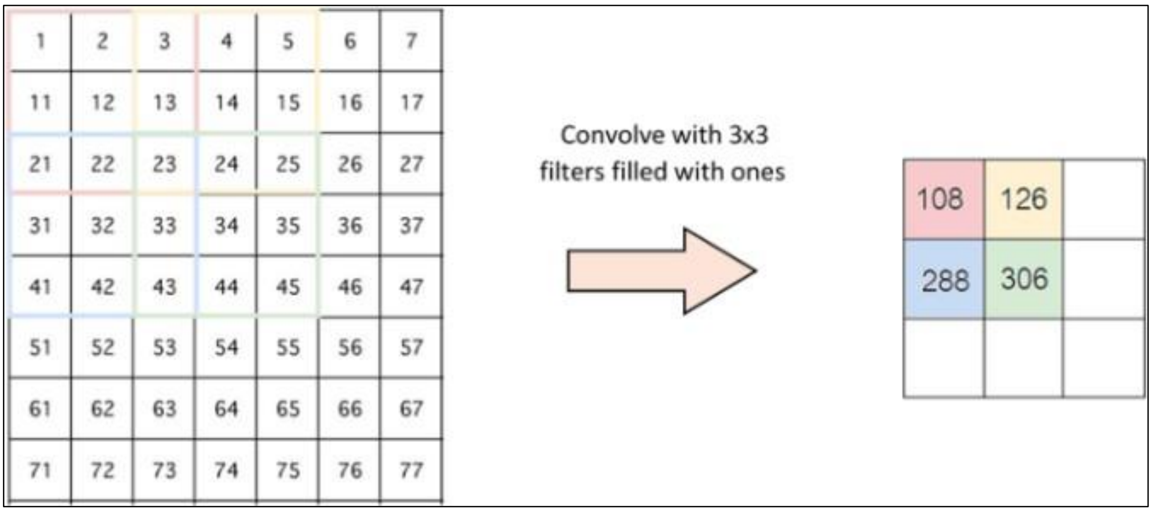


Figure 12 Stride of 2 Pixels [31].

1.8.3 Convolutional neural networks layers

The layers that make up and characterize the convolutional neural network will be presented in this section.

a) The convolution layer

The convolution layer is critical to the operation of CNNs; it ensures that features are extracted automatically by processing the input images.

This is accomplished by convolution filtering, which works by dragging the filter matrix over the image matrix and calculating the convolution product between the two. This operation yields a collection of output characteristics as a consequence (Figure 13).

The convolution layer takes numerous input images and convexes them with each filter. The filters represent the traits we're looking for in the input images. On each input, multiple convolutions can be run, each employing a different filter. The final output of the convolution layer is a grouping of feature sets produced by this layer [31].

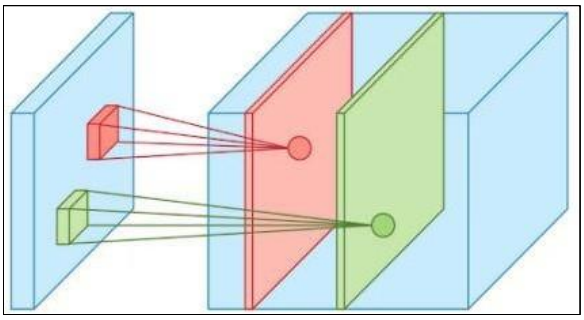


Figure 13 Applying multiple filters to an input image [32].

b) Zero padding layer

The size of the feature matrix is lowered after conducting a convolution. This size will continue to shrink as the number of convolution layers increases, resulting in:

- Loss of boundary information after each convolution layer.
- The decline in educational quality.
- After several convolutions, reducing the size of the feature map to 1x1.

To address these issues, the Zero Padding technique recommends surrounding the picture matrix with a layer of pixels with a zero value (Figure 14)

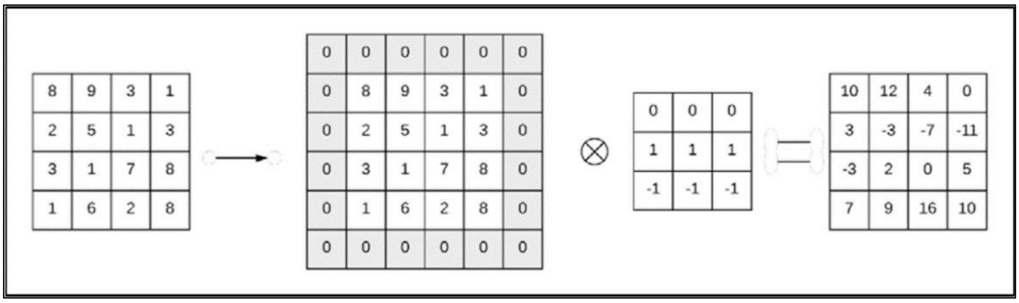


Figure 14 Using zero padding before a convolution [32].

This method preserves as much information as possible about the original input volume to retrieve its properties.

c) The ReLu layer

The non-linear function ReLu will be applied to all the values of the feature matrix shortly after it is created. This function ensures that training is completed quickly and precisely. The ReLu function equalizes negative values in the matrix [33].

d) Pooling layer

This layer is frequently sandwiched between two convolution layers. By applying the union operation to each input vector, it tries to gradually lower the dimensionality of the representation, reducing the number of parameters and computing complexity of the model while keeping their significant qualities [33].

e) Flattening layer

The flattening layers job is to convert feature matrices into vectors that may be passed on to the fully linked layer [33].

f) Fully connected layer

Fully connected layers will be added after the preceding levels to encompass the CNN architecture (Figure 15) [34]. Artificial neural networks have the same architecture.

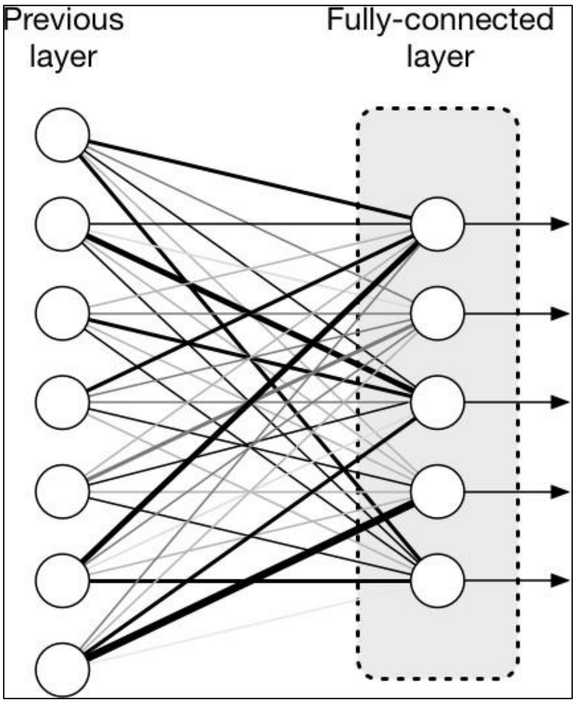


Figure 15 Fully connected layer [34].

1.9 Conclusion

In this chapter, we talk about access control systems and how it is important for making private places safe. we explained what is the benefits of having an automatic access control system nowadays. finally, we discussed how Artificial Intelligence is evolved in the twenty-first century and how much attention it is generating in the realm of security systems. After that, we looked at the various neural network as well as the key principles behind their architectures. Following that, we demonstrated the common topologies of a Convolutional Neural Network by presenting each layer that makes up the network and thoroughly describing each layer's utility. We discovered that the automation of the feature detection phase is what makes Convolutional Neural Networks so popular. Whereas traditional image processing approaches require professionals to extract them, CNN designs do it automatically.

The next chapter will be dedicated to the APLR related works with some comparisons between their solutions.

Chapter 02
Automatic license plate recognition related works

2.1 Introduction

In this chapter, we will start with LP history and use cases and some used techniques in LP recognition after that we will talk about some related works like deep learning models and the databases used and conclude by comparing the results given for all those works.

Vehicle license plates were first invented and used for carriages, but not for motors. The license plate was initially presented in 1884 in Victoria, Canada, for a horse-drawn hackney carriage. On April 14, 1899, the police in Germany gave Mr.Chubais Barthes the license plate, which is the world's first best-preserved license plate. This is a rectangular license plate with simply 1 Figure16 inscribed on it. On August 14, 1983, France implemented the world’s first license plate rule. All license plates must be registered with their name, address, and registration number under this rule [35, 8].

Figure 16 The first saved license plate [8].

2.2 License plate use cases

Vehicles are now widely used in every area of production and in people's lives as a primary mode of transportation in modern societies. With the growing number of vehicles on the road, traffic infractions such as running a red light or exceeding the posted speed limit will become more common. If only numbered traffic cops are used to prevent a large number of traffic offenses, public transportation will become paralyzed. Every car's license plate has unique information. And its image is a valuable resource for contacting and sharing information with its owner. As a result, images can be utilized as a primary medium for identifying people and vehicles. And image analysis technology was already used in a variety of areas of human life. As a result, using a digital picture to automatically gather and handle license plate information has proven to be an effective technique to monitor public transportation [8].

Because the license plate number is a simple and effective technique for identifying a vehicle, vehicle license plate recognition is an additional important aspect of modern intelligent transportation systems. Without the process of car license plate recognition, there would be no vehicle parking access control or auto payment check on the highway. The license plate recognition technology can be used for a variety of purposes [36]: 1) Automated highway tolling and monitoring management, 2) community automated parking management, 3) urban road monitoring and illegal-events management, 4) vehicle inspection, 5) traffic statistics, and safety management.

2.3 Characteristics of Algerian license plates

The registration plate standards are set by the decree of June 15, 1993, amending the decree of May 5, 1988, laying down the administrative rules relating to the registration number of motor vehicles.

The figures as well as the operating plates must meet the characteristics defined for the motor vehicle by the decree of 23 June 1975 relating to the registration and re-registration of motor vehicles.

The faceplate shall contain Arabic numerals on a gray-white reflective background, and the backplate shall contain black Arabic numerals on a yellow reflective background [37].

Also, the registration plate must be made of aluminum and have the following dimensions:

- 1 millimeter thick.
- 52 centimeters in length.
- 11 centimeters in width.
- The figures must be 7.5 centimeters high and must be distinct from the rest of the plate by 0.5 millimeters [37].

2.4 License plate recognition system organization.

There are two types of ALPR systems now available: multistage and single-stage approaches.

2.4.1 Single- stage license plate recognition systems

While the majority of prior work on license plate identification has concentrated on multi-stage methods, there have lately been numerous successful single-stage attempts. To the best of our knowledge, all of these approaches use a single deep neural network that is trained for end-to-end detection, localization, and recognition of the license plate in a single forward pass. License plate identification is an example of object detection in action. These models, like single-stage object detectors, can make use of the fact that license plate detection and recognition are highly correlated [38].

This enables models to share parameters and require fewer parameters than a traditional two-stage model. As a result, they may be more efficient and faster than two-stage methods [38], [39]. Li et al. [38] made the first attempt we are aware of. As a feature extractor, they employed VGG16, which is a convolutional neural network model [40]. Because the license plate covers a lesser area in a typical images, they modified VGG16 to employ only two pooling layers instead of five. The feature extractors' output is then input into a Region Proposal Network (RPN) [41]. They made changes such as using two rectangular convolution filters instead of the traditional three. This is to make use of the fact that license plates have a larger aspect ratio than square filters, therefore rectangle filters perform better [42][43].

2.4.2 Multi-stage license plate recognition systems

The multi-stage strategy, which comprises three primary steps, has been considered in the majority of existing ALPR solutions. The detection or extraction of license plates is the first stage. To locate the license plate in a picture, existing algorithms employ classic computer vision techniques and deep learning methods with object detection [43].

Traditional computer vision algorithms rely heavily on license plate characteristics such as shape symmetry texture and so on. The license plate is segmented in the second stage, and the characters are recovered using methods including mathematical morphology [44], linked components, relaxation labeling, and vertical and horizontal projection [45].

The character segmentation stage, on the other hand, is not required in every multi-stage ALPR system because certain segmentation-free algorithms skip this step [43].

The final stage is character identification using pattern matching techniques or classifiers such as neural nets and fuzzy classifiers. However, separating detection and recognition has a negative impact on the whole recognition process' accuracy and efficiency [43].

This occurs primarily due to defects in the detection process, such as bounding box prediction problems. For example, if the detection method misses a portion of a license plate, the overall accuracy of the recognition process would suffer. As a result, with a multi-stage method, achieving satisfactory results at each level is critical. The basic processing phases of a multi-stage plate recognition system are shown in Figure 17 [43].

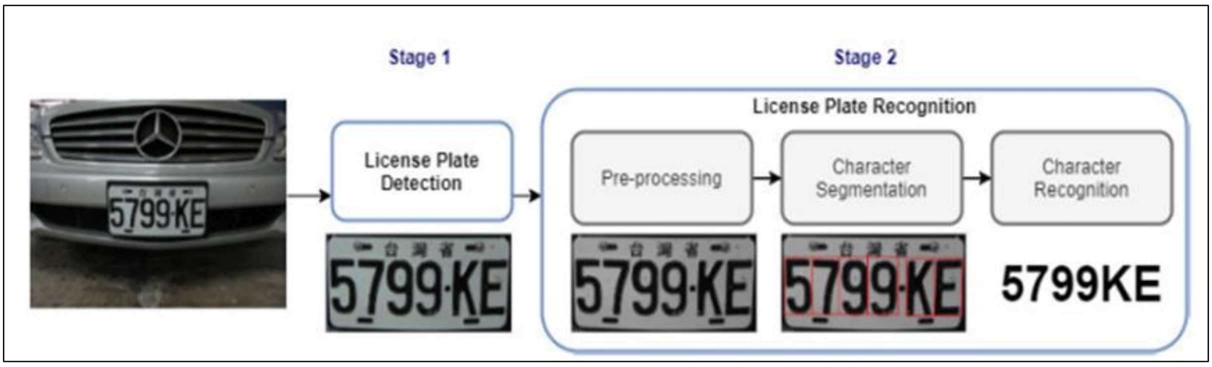


Figure 17 Main stages in a multi-stage license plate recognition system [43].

a) Image acquisition

Parking management and speed limit enforcement systems typically employ vehicle license plate recognition. This device helps to prevent traffic accidents and crimes.

Closed-circuit television (CCTV) cameras are used to record and store images on highways and buildings.

Video and image transmission The camera's settings, such as type, resolution, shutter speed, light, and direction must be considered.

According to the national statistical office of South Korea, around 1.2 million CCTV cameras have been installed in the country as of 2019. However, the information gathered by this system isn't being completely exploited. To fully utilize and analyze the images acquired by CCTV, it takes a lot of time and human resources, which becomes a limiting factor in responding quickly to an accident or crime. This issue can be efficiently solved by implementing an automated technique that can swiftly detect information such as the vehicle's kind, color, and license plate.

The image obtained through CCTV, which is normally captured from a great distance and with a broad angle of view, is used in the technology for recognizing a vehicle license plate. Due to varied environmental changes and camera installation locations, this presents issues such as resolution limitations, motion blur, and perspective distortion. Image processing technology that can reliably discern characters lower than the image size is sought. Studies in areas such as object detection, tracking, and camera acquired image recognition technologies have been done as image processing technology has progressed [46]. Vehicle license plate recognition research is also being actively done to efficiently manage vehicle information [47][48].

Figure 18 Pictures of real-life applications of ALPR system: ALPR system in traffic [49].

b) Automatic license plate detection

License plate extraction is a critical phase in an ALPR system that ensures the system's correctness. Given an input image, the purpose of this phase is to create a zone that contains a real license plate [50].

b.1) Image capturing and noise removing

A high-resolution digital camera is employed to capture a picture in this system. Images are taken using a variety of backgrounds, lighting situations, and distances between the camera and the car. The images are resized to fit the page (1024 X 768). On a grayscale image, all of the processing stages are carried out. Preprocessing is mostly used to increase picture processing speed, improve image contrast, and reduce image noise. As a result, we must filter this noise before processing the image. Low-pass filters are commonly employed to alleviate the problem of poor image quality and contrast of vehicle images. Original image acquired with a digital camera, as well as a grayscale image [50].

b.2) Vertical edge detection

Concerning the background, the license plate region has a lot of edges. To locate locations with a high pixel variance value, Sobel edge detection is performed. A threshold is used to pick rows with a specific white pixel density out over the candidate license plate area from the overall image. The effect of utilizing Sobel edge detection and threshold [50] is shown in Figure 19.

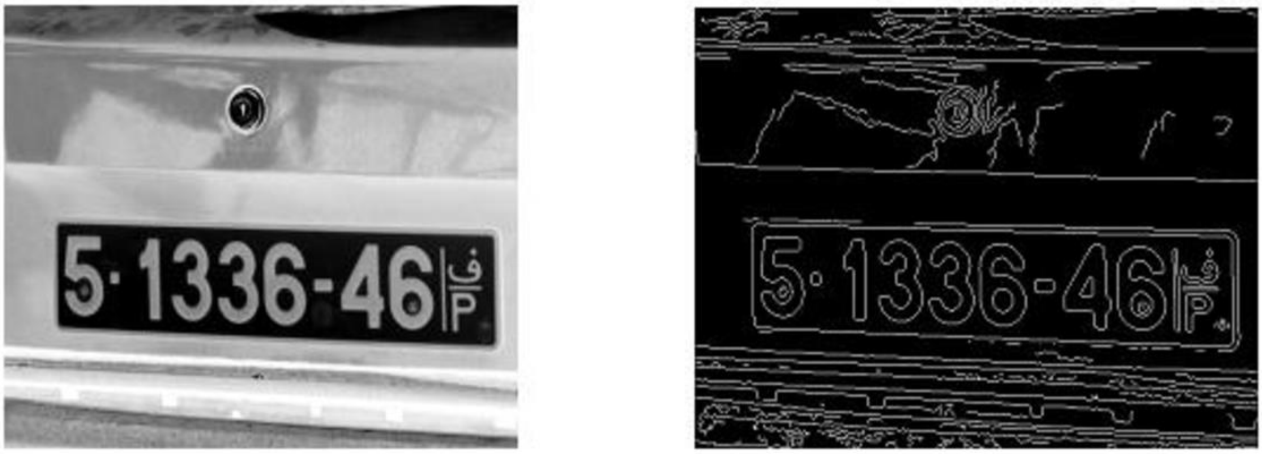


Figure 19 Vertical edge detection [51].

b.3) Candidate plate area detection

The goal of morphological operations is to eliminate unrelated objects from an image. To extract candidate plate portions from the overall image (Figure 20), dilation and erosion are applied. It is not uncommon for background areas to be designated as candidate plates. As a result, plate validation is done utilizing the aspect ratio of the plate and horizontal cuts in the license plate to eliminate bogus candidates. For real license plate extraction, an invert and threshold procedure is used [50].

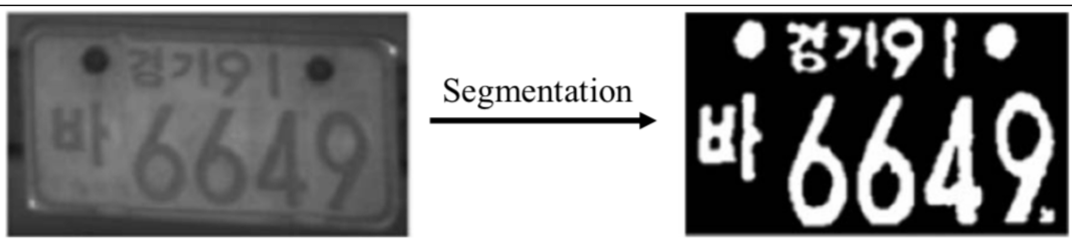


Figure 20 Detection and segmentation [52].

b.4) Skew detection and correction

Skew detection and correction are required to make text lines horizontal if the acquired image is skewed. The horizontal direction of the text lines of the document image is created by the skew angle. Correction of skew can be done in two steps. First, we will calculate the skew angle, and then we'll rotate the image using the skew angle [50].

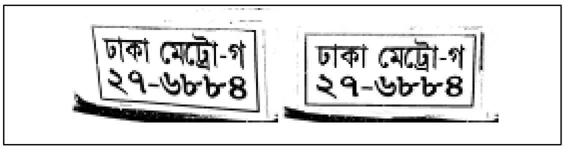


Figure 21 Skew detection and correction [50].

c) Segment the characters out from the license plate.

Character segmentation is a significant part of this system. this is one of the techniques to segment the character (Figures 22 and 23).

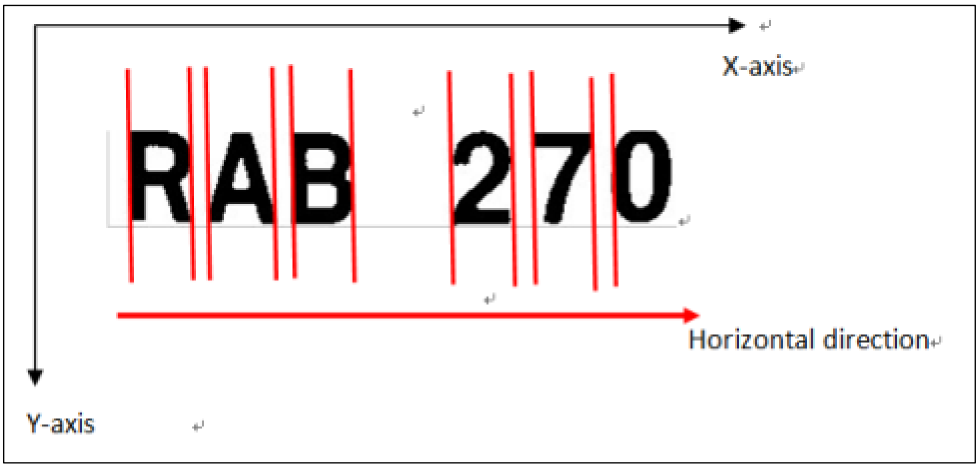


Figure 22 Character segmentation (European license plate) [8].

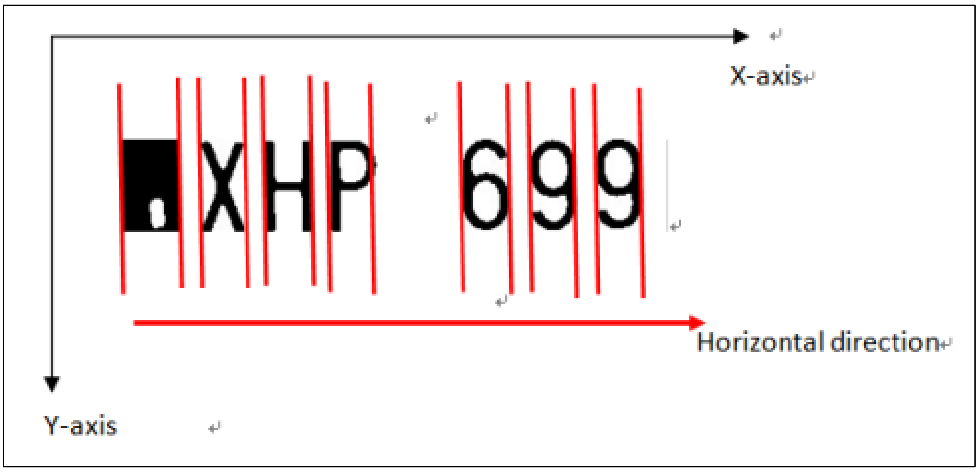


Figure 23 Character segmentation (Swedish license plate) [8].

We represent this binary image as a matrix. We begin by creating this matrix in the opposite direction. The sum function is then used to return the row vector containing the sums of each column. Then, using a loop, we should investigate the matrix in the horizontal direction. We may build a restricted condition based on the sum to determine if the loop should continue or end. When the sum of one column is less than one and the sum of the following column is more than one, the data is segmented from this column to the previous one [8].

It would undoubtedly continue to search if the criterion was not met. If the vehicle is part of the European Union (Figutr 24), the characters added to one logo are seven blocks. As a result, the segment must be repeated 14 times in total. Otherwise, only 12 times segmentation is required. We set a counter in the segmentation processing to calculate how much segmentation the programming requires. As a result, we can determine the country of the license plate which would greatly assist in the next phase [8].

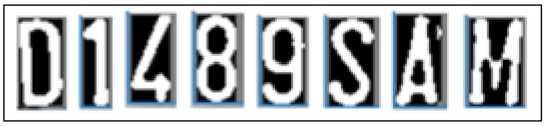


Figure 24 Results of segmenting characters [53].

d) Character recognition [54]

Character recognition has also been a hot topic in academia. Character Recognition can be accomplished using a variety of strategies, each of which is superior to the others in particular situations. One of the most well-known is the Support Vector Machine (SVM), which is best for tiny data sets and was created by segmenting real samples.

After segmentation, the license plate is thresholded so that binary data can be used for recognition and training. The average grayscale value is used as the threshold in adaptive thresholding. Thresholding is not required for character recognition; however, it is suggested for SVM because it decreases data set noise.

Each character is scaled down to 6x15 pixels for training. It is subsequently turned into a 90-dimensional one-dimensional vector. For each character, 20-30 such samples are collected and vectorized to train the SVM.

In OpenCV, the multiclass SVM utility is utilized, which handles all characters from 0 to 9. Because each data member is given a tag that indicates its class, it is supervised learning.

Because of the tiny number of training samples, a linear kernel produces satisfactory results. To train the classifier, auto grid search is utilized to discover the best gamma and C values.

d.1) Character recognition tool Tesseract-OCR

Tesseract is an open-source text recognition for converting printable text into editable text. [55] The Tesseract-OCR is an OCR engine where the user can input flags, values, and configurations with pre-trained models for different languages, to make the OCR software read text in different ways. There are many ways in which an OCR program can analyze and scan an image with text on it. While one configuration can be advantageous on an image with text forming words and sentences, another could be in favor of reading large characters randomly. This configurable OCR software is what differentiates the Tesseract-OCR from the ordinary OCR [56].

d.2) Character recognition tool Pytesseract

Pytesseract is a Python wrapper for Tesseract-OCR. This wrapper allows Python to call the Tesseract-OCR, which, because Python can read a wider range of image formats, allows the Tesseract-OCR to read the same formats. Because the OCR engine is called from Python, it isn't confined to writing its guesses to an output file; instead, it can write the text to Python and use it in a script [56].

2.5 Automatic license plate recognition related works

Automatic license plate recognition is used to identify vehicles based on their license plates and is crucial in several major transportation applications [57][58]. Furthermore, in the realm of image processing, it is a very popular and active study topic [58]. As a result, and as computers get more sophisticated, we are seeing new ways based on deep learning, which is presently the most common approach in the field of image processing.

In this chapter, we choose a set of models for automatic license plate identification and recognition using a deep learning technique and discuss how they are set up. These models include:

- Automatic detection models.

    1. SSD.
    2. Faster R-CNN.

- Automatic recognition models.

    1. Selmi et Al model.
    2. Model of Li and Shen (9-layers).
    3. Wu et Al model.
    4. Model of Spanhel et Al.

Finally, we compared the automatic detection models in terms of frame rate (FPS) and average precision (mAP) acquired by the dataset. In terms of model and dataset, we compared automated recognition models. The validation task yielded accuracy and error.

2.5.1 Automatic license plate recognition models

The detection of license plates is critical in many significant transportation applications [57]. Despite its relevance, it still has several issues, including difficulties in identification, which is dependent on device speed, and lengthy execution. Finally, no model is completely effective [59].

a) MultiBox single shot detector (SSD)

SSD [60] is a deep neural network-based approach for recognizing objects in pictures (Figure 25). Lui et al. suggested it in 2018 [60]. This approach discretizes the bounding box output space into a default set of boxes per position on the map, with varied aspect ratios and sizes. The network generates scores for the existence of each item category in each default box at prediction time, as well as area changes to better match the shape of A, I' objects. In addition, the network uses predictions from numerous feature maps with varied resolutions to control objects of various sizes intuitively [59].

SSD is more straightforward than approaches that rely on object proposals since it eliminates the proposal generation and pixel or feature resampling stages, encapsulating all calculations in a single network. This makes SSDs simple to manufacture and integrate into systems that require sensing. On the VOC2007 test at 59 FPS on an Nvidia Titan X, the SSD achieves 74.3 percent mAP,

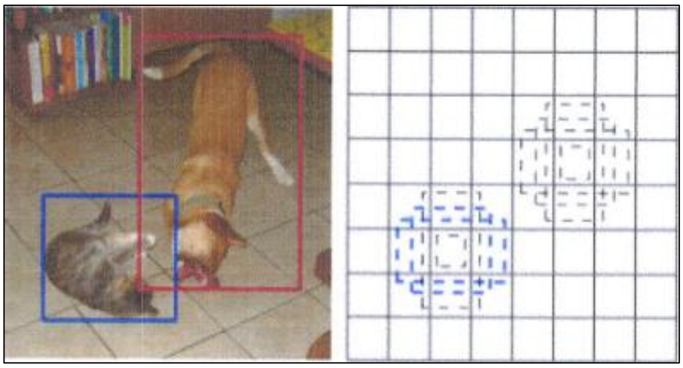


Figure 25 SSD [60].

and for a 512 x 512 input, the SSD achieves 76.9% mAP, outperforming my Faster R model -CNN. SSD provides substantially better accuracy than other one-step approaches, especially with smaller input image sizes [59].

b) Faster R-NN

Girshick presented the Fast R-CNN architecture (figure 26) [62] in 2015. Is a network that takes a full image and a collection of object propositions as input. To create a conv feature map, the network first analyses the entire image with several convolutional layers (CONV) and Max-pooling. A region of interest (RoI) clustering layer then extracts a fixed-length feature vector d from the feature map for each feature proposition. Each feature vector is input into a series of fully connected (FC) layers, which branch out into two sibling output layers: one that generates SoftMax probability estimates on K object classes plus a "background" class, and another that generates four actual values for each of the K object classes.

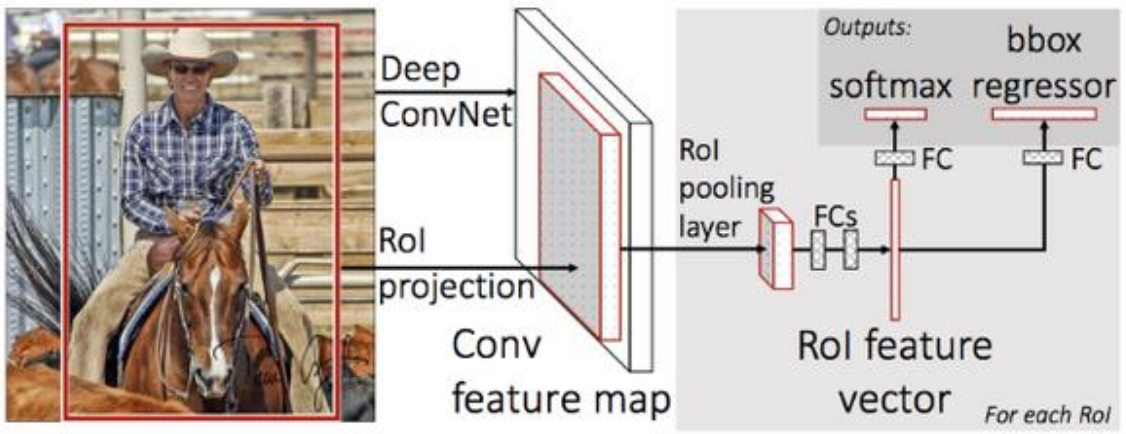


Figure 26 Fast R-CNN [62].

c) Comparison between license plate detection models

We chose a set of models from various methodologies used for automatic license plate detection in the deep learning approach for the comparison (Table 3). We chose SSD513 [60][63], DSSD513 [63] for the SSD technique, and Faster R-CNN w TDM [64] and Faster R-CNN w FPN [65] for the Fast R-CNN approach.

Table 3 Comparison between detection models with a Titan X GPU and COCO dataset [59].

	Approach	AP	AP(50)	AP(75)	AP(M)	AP(L)	Time AP(50) ms
Faster R-CNN w FPN [65]	Faster R-CNN	36.2	59.1	39.0	39.0	48.2	172
Faster R-CNN w TDM [65]	Faster R-CNN	36.8	57.7	39.2	39.8	52.1	
SSD513[60][63]	SSD	31.2	50.4	33.3	34.5	49.8	125
DSSD5[63]	SSD	33.3	.53.3	35.2	35.4	51.1	156

2.5.2 Automatic number plate recognition models

In many Transportation applications, the license plate numbers recognition stage is critical [57]. Despite its significance, it still has significant flaws, and no model is completely effective.

a) Selmi and Al model

This model [58] consists of a deep convolutional network (CNN) with 37 classes (10 digits) and a multilayer perceptron (MLP). For the output layer, there are 26 capital letters and a non-character negative category. Selmi et al suggested it in 2017 [58]. There are four convolutional layers in total. Two completely connected layers, three pooling layers, and one dropout layer The supplied image is scaled to a grayscale resolution of 32 by 32 pixels. Finally, to estimate the class membership probabilities for each input example, the network includes ReLu activation layers and a Softmax activation layer.

a.1) Dataset

On two datasets, Selmi et al investigated the effectiveness of the suggested license plate recognition method (datasets). The Caltech2 Cars (actual) dataset is the first, while the AOLP reference database is the second. This database is split into three sections: access control (AC) with 681 samples, road code application (LE) with 757 examples, and road patrol (RP) with 611 samples.

a.2) Model setup

The configuration of the model is shown in Table 4

Table 4 Character recognition setup of the Selmi et al model [58].

Layer type	parameters
Input	Gray image 32x32
Convolution ReLu	Depth 32, window 5x5 steps 1 p0
Max pooling	Window 2x2, steps 2
Convolution ReLu	Depth 64, window 3x3 steps 1
Max pooling	Window 2x2, steps 2
Convolution ReLu	Depth 128, window 3x3 steps 1
Convolution ReLu	Depth 256, window 3x3 steps 1
Max pooling	Window 2x2, steps 2
fully connected ReLu Dropout	Size :1024 neurons 0.5
fully connected	Size: 37 neurons
Softmax	Size: 37 classes

b) Li and Shen model (9-layers)

The output layer of this model [66] is made up of a Concurrent Deep Network (CNN) and a Bidirectional Recurrent Network (BRNN) with 36 classes (10 digits, 26 capital letters). In 2016, Li and Shen proposed it [66]. Six convolutional layers, four pooling layers, two dropout layers, and three fully connected layers make up this image. The supplied image is scaled to a grayscale resolution of 24 by 24 pixels. Finally, the network includes ReLu activation layers and a softmax activation layer to predict the probability of class membership for each input example to the output of a two-way recurrent network of a unit (BLSTM) [59].

b.1) Dataset

Li and Shen used the same dataset as Selmi to examine the efficacy of license plate recognition.

b.2) Model setup

The configuration of the model is shown in Table 5.

Table 5 The configuration of Li and Shen (9-couch) model for character recognition [66].

Layer type	parameters
input	Gray image 24x24
Convolution ReLu	Depth 64, window 3x3 steps 1 p1
Max pooling	Window 3x3, steps 1
Convolution ReLu	Depth 64, window 3x3 steps 1 p1
Max pooling	Window 3x3, steps 2
Convolution ReLu	Depth 64, window 3x3 steps 1 p1
Convolution ReLu	Depth 64, window 3x3 steps 1 p1
Max pooling	Window 3x3, steps 2
Convolution ReLu	Depth 64, window 3x3 steps 1 p1
Convolution ReLu	Depth 64, window 3x3 steps 1 p1
Max pooling	Window 3x3, steps 1
fully connected ReLu Dropout	Size :1000 neurons 0.5
fully connected ReLu Dropout	Size :1000 neurons 0.5
BLSTM	Output 1000 neurons
fully connected	Size :36 neurons
Softmax	Size : 36 classes

c) Wu and Al model

For the output layer, this model [67] is made up of a DenseNet [68] with 68 classes (31 Chinese characters, 10 numbers, 26 capital letters, and the non-character negative category). DenseNet is a deep convolutional network (CNN) with dense blocks. A convolutional block, three dense blocks, two transition blocks, and a fully linked layer were proposed by De Wu er al in 2018 [68]. Resize the input image to 136 × 136 grayscale pixels. Finally, the network includes ReLu activation layers and a softmax activation layer to estimate the probabilities of class membership for each input example.

c.1) Dataset

Wu et al. They use two datasets to investigate the effectiveness of license plate recognition. Dataset-1 has a training set of 203774 plates and a trial set of 9986 plates, whereas Dataset-2 is the one utilized by Selmi et al (AOLP) [67].

c.2) Model setup

The model's setup is shown in table 7.

Table 6 Wu et al model configuration for character recognition [67].

Bloc type	Layer type	Number of repeats	parameters
Input	input	X1	Gray image 5x 5 steps 2
BLoc Convolutive	Convolution Batch normalisation (BN) ReLu Max pooling	X1	Depth 64, window 5x5 steps 2 p1 Epsilon = 1.1e-5 Window 3x3, steps 2
BLoc Dense	Batch normalisation (BN) ReLu Convolution	X8	Epsilon = 1.1e-5 Depth 128, window 3x3
Bloc transition	Batch normalisation (BN) ReLu Convolution Average Pooling	X1	Epsilon = 1.1e-5 Depth 128, window 1x1 window 2x2, steps 2

BLoc Dense	Batch normalisation (BN) ReLu Convolution	X8	Epsilon = 1.1e-5 Size : 192, window 3x3
Bloc transition	Batch normalisation (BN) ReLu Convolution Average Pooling	X1	Epsilon = 1.1e-5 Depth 128, window 1x1 window 2x2, steps 2
BLoc Dense	Batch normalisation (BN) ReLu Convolution	X8	Epsilon = 1.1e-5 Depth 192, window 3x3
Output	fully connected softmax	X1	Size 68 neurons Size 68 Classes

d) Snahel et Al model

This model [69] consists of a deep convolutional network (CNN) connected to eight branches of multilayer perceptron (MLPs) that recognize characters in 36 classes. Spanhel et al. proposed it in 2017[69]. It has three sequences of three layers of convolutions with ReLu and Batch Normalization, three layers of pooling, eight branches, and two fully linked outputs on each branch. The input image is scaled to a color image of 200x40 pixels. Finally, a softmax activation layer was utilized to forecast the class probabilities for each branch [69].

d.1) Dataset

Spanhel and colleagues Three datasets were used to test the effectiveness of the proposed license plate recognition system. The first is the ReId Dataset, the second is HDR, and the third is the Svoboda et al Dataset, [70]. (bleached + debleached) [69].

d.2) Model setup

Table 8 shows the model configuration.

Table 7 The configuration of Spanhel et al character recognition model [69].

Bloc type	Layer type	Number of repeats	Number of branch	parameters
input	input	X1	X1	Color image 200 x 40
Sequential conv1	Convolution ReLu Batch normalisation (BN)	X3	X1	Depth 32, window 3x3
Poll1	Max pooling	X1	X1	Window 2x2 steps 2
Sequential conv2	Convolution ReLu Batch normalisation (BN)	X3	X1	Depth 64, window 3x3
Poll1	Max pooling	X1	X1	Window 2x2 steps 2
Sequential conv3	Convolution ReLu Batch normalisation (BN)	X3	X1	Depth 128, window 3x3
Poll1	Max pooling	X1	X1	Window 2x2 steps 2
output	fully connected fully connected softmax	X1	X8	Size 128 neurones Size 36 neurones Size 36 classes

e) Comparison between license plate recognition models

For this comparison, we employed a variety of alternative models for automatic license plate recognition in the Deep learning technique. The models will be compared in Table 8 based on:

• the networks that make up the model.

• the utilization of various layers and blocks.

• If not by characters, the correctness of each model by license plate.

Table 8 Comparison between (Selmi et Al) and (Li et Shen) models.

Type	Selmi & al Model [58]	Li & Shen Model (9-Layers)[66]
Model	CNN+MLP	BNN+BRNN
Input	Grey 32x32	Grey 24x24
Convolutional layer	X4	X6
Max pooling Layer	X3	X4
ReLu	X5	X8
Fully Connected Layer	X2	X3
Dropout Layer	X1	X2
Softmax	X1	X1
BLSTM (BRNN)	—	X1
Evaluation With Precision %		
AOLP	96.20%	94.85%
AOLP	95.40%	94.19%
AOLP	95.10%	88.38%
Caltech	94.80%	—

Table 9 Comparison between (Wu et Al) and (Spanhel et Al) models.

Type	Wu & Al. Model [67]	Spanhel & Al. Model [69]
Model	DenseNet (CNN)	CNN+MLPs (MLP x8)
Input	Grey 136x136	Color 200x40

Convolutional layer	X27	X9
Max pooling Layer	X1	X3
Moyenne Pooling Layer	X27	__
ReLu	X27	X9
Fully Connected Layer	X1	X16
Normalisation par lot Layer	X27	X9
Softmax	X1	X8
Convolutive Bloc	X1	__
Dense Bloc	X3	__
Transition Bloc	X2	__
Sequential Convolutional bloc	__	X3
Output	__	X8
Evaluation With Precision %		
AOLP (AC)	96.61%	__
ALP (LE)	97.80%	__
AOLP (RP)	97.00%	__
Dataset_1	97.90%	__
ReId	__	98.00%
HDR	__	90.30%
Dataset Svoboda & al. [70]	__	55.40%
Dataset Svoboda & al. [70] deblur	__	91%

All those models are based on CNN Architecture except the Li & Shen Model based in BRNN with different input shapes and different layers.

Selmi & al and Li & Shen models were evaluated on the same datasets (AOLP), the best result of the first one was given in AOLP dataset with 96.20% precision and the second one gives on the same dataset with 94.85% precision.

Wu & Al model evaluated on multiple datasets and its best result are on dataset_1 which is 97.90% precision and for the Spanhel & Al model, it gives a result of 98.00% on reld dataset.

2.6 Conclusion

Automatic license plate detection and recognition can be developed into two separate tactics as we saw in this chapter, single-stage and multistage.

We look at how the multistage technique works and what each phrase means. We also look at six technical literature papers, two in the part of the detection and four in the field of recognition. We describe the overall architecture of the suggested model, the dataset used, and the precision gained in each paper. We compared these diverse models in their respective fields at the end of this chapter.

In the next chapter, we will talk about the second part of our system named Facial Recognition related works.

Chapter 3 Facial recognition related works

3.1 Introduction

In this chapter, we start with introducing the face recognition system, after that, we will explain the steps in facial recognition, then we will talk about some related works like famous deep learning models and the databases used.

Finally, we will conclude by comparing the results given for all those works.

3.2 Face recognition system organization

The Face recognition system pass by multi-steps to recognize the face id which them:

3.2.1 Face detection

Face detection is an important step in facial recognition systems. It begins with the capture of a scene that contains a face and then extracts the face from the image using one of the detection methods to keep a region Region Of Interest (ROI) that contains the components of the face, which will then be refined using a preprocessing procedure. It necessitates not only face localization but also face tracking across a series of photos [74].

Because any identification system relies on detection to produce effective results, it is critical to adopt algorithms that are both robust and efficient. Many methods were developed for this goal.

Figure 27 Face detection [75].

3.2.2 Face recognition

Face recognition research has progressed significantly in recent years [77, 78,79], owing to the availability of large amounts of data and GPUs for training Deep Learning models. Face recognition with expression, position, aging, disguise, or illumination variations has been the focus of certain existing techniques [80,81].

a) Traditional face recognition methods

There are several classic face recognition algorithms in the literature, and Ojala et al. [82] introduced one of the earliest approaches by applying Extended LBP (ELBP) to improve the discriminative capabilities of faces. The ELBP performs a binary comparison between the core pixel and its neighbors, as well as encoding the exact grey-value changes between them using additional binary units.

To extract features, Wang et al. used stationary wavelet entropy and a single hidden layer feedforward neural network as a face classifier [83]. They also created the Jaya algorithm to avoid the classifier's training falling into local optimal points.

Other solutions for recognizing a face in unconstrained circumstances include model and geometry-based algorithms. Yin et al. presented the Associate Predict (AP) Model [84] to deal with the resemblance of human faces in face recognition under drastically variable stance, illumination, and expression settings. Yang et al. suggested a discriminative Multi-Dimensional Scaling (MDS)

Method for learning a mapping matrix that maps high-resolution and low-resolution facial photos to the same subspace [85]. To ensure discriminability, they introduce an inter-class constraint that enlarges the distances between distinct subjects in the subspace.

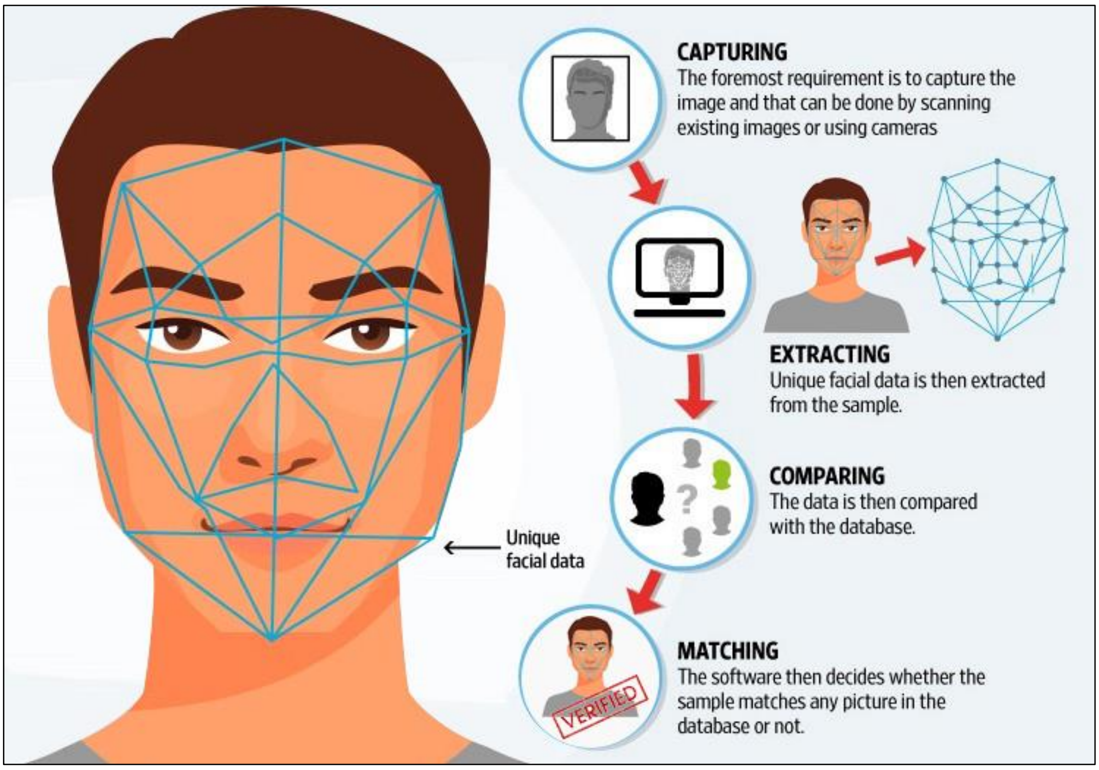


Figure 28 Face recognition steps [86].

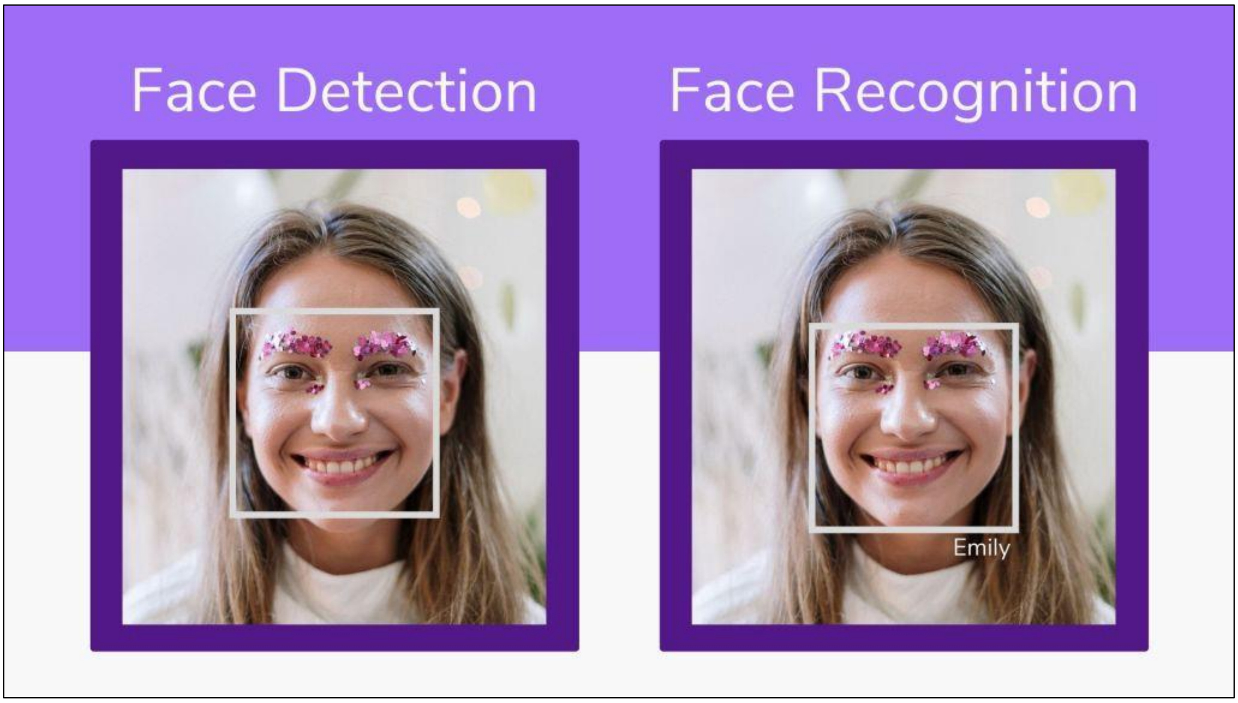


Figure 29 Different between face detection and face recognition [87].

3.4 Face recognition related works

The face recognition system contains multiple sup systems. The necessary subsystems are:

3.4.1 Face detection systems

Human face identification is a difficult task. Methods for detecting faces have been classed by:

Table 10 introduce [88] for Yang [89] four approaches:

- Knowledge-based approach.
- Approach based on invariant characteristics.
- Approach based on template matching.
- Approach based on appearance.

Table 10 Classification of face detection methods [88].

Approach	Representative Works
Knowledge-based	Multiresolution rule-based method
Facial feature	Grouping of edges
Texture	Space Gray-Level Dependence matrix (SGLD) of face pattern

Skin Color	Mixture of Gaussian
Multiple Feature	Integration of skin color, size, and shape
Pride ned face templates	Shape template
Shape template	Actives Shape Model (ASM)
Eigenface	Eigenvector decomposition clustering
Distribution-based	Gaussian distribution and multilayer perceptron
Neural Network	Ensemble of neural network and arbitration schemes
Support Vector Machine (SVM)	SVM with polynomial kernel
Naive Bayes Classi er	Joint statistics of local appearance and position
Hidden Markov Model (HMM)	Higher-order statistics with HMM
Information-Theoretical	Approach Kullback relative information

a) Knowledge-based approach: It is a rule-based system for representing the major and representative features of human faces (Figure 30). The link between facial features is frequently used to establish rules. Faces in images, for example, frequently have two symmetrical eyes, a nose, and a mouth. The distance between these individuals and their position might be used to show their relationship.

The difficulty of transferring knowledge about human faces into well-defined rules is a flaw in this method, which might lead to detection mistakes and render the system untrustworthy. [90]:

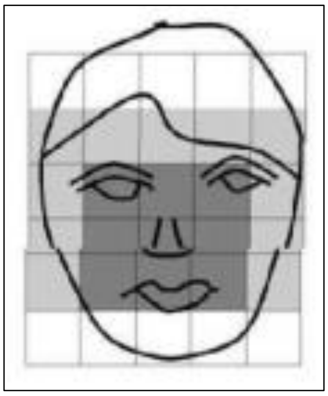


Figure 30 Typical face of the knowledge-based method [91].

The detection algorithm is as follows [91]:

- If the initial report is near to the first report of a normal face, we can compare the next report otherwise, the lineage being studied is not a face image.
- If all ratios are equal in the end. One image that I investigated is a image of a face. The issue with this strategy is obtaining specified good rules.
- If the guidelines are detailed, some faces that do not follow all of the requirements may be undetected.
- We can detect images that are not images of faces if the restrictions are broad enough. Because you can choose which features to impose, you can use a variety of ways in this approach. Yang et Huang is still the most successful among them.

b) Approach based on invariant characteristics: Even whether the face is in varied locations, light circumstances, or viewing angles, this family of algorithms seeks to uncover structural traits. The issue with this method is that the image quality might be substantially harmed as a result of illumination, noise, or occlusion. The face has various invariable traits or characteristics, the most important of which are listed below [88]:

b.1) Skin color: Human skin color has been utilized to recognize faces and is useful as a face-specific characteristic (Figure 31). The premise behind this technology is to use color information to distinguish between the skin and non-skin pixels. A color image's pixel is coded in a color space (for example RGB or YCrCb, ..).

This procedure can be broken down into three steps:

- Image preprocessing.
- Choosing a color space.
- Skin color thresholding and segmentation.

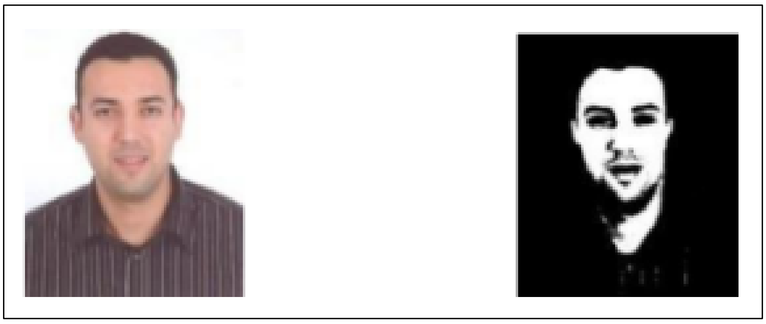


Figure 31 Image detection by skin color [91].

This system is distinguished by the quickness with which it treats patients and the ease with which it makes decisions. The approach is straightforward and limited to skin color, with no regard for scale or position effects. This dread, on the other hand, can lead to false positives and generate conflicts with the background [88].

b.2) Face features: Edge plants and heuristics are used in this technique to remove all groupings of edges except those that depict the contours of the face. A border between the background and the face is derived as an ellipse. This is characterized as being produced from the image's luminance (intensity) function's points of discontinuity. The main idea is to use previously established contour models to distinguish items in a picture. Two approaches will be provided to complete this task: the Hough transform (which allows you to extract and locate groups of points that share certain properties, using an equation of a well-defined form) and the Hausdorff distance (which aims to measure the distance between two separate sets of points) [92].

b.3) Multi-features: Face detection can be accomplished using a variety of techniques that incorporate several facial traits. The bulk of them use broad criteria such as skin color, shape, and face size to discover candidates, then scrutinize them locally based on details like eyes, brows, nose, and hair. Numerous works employ this strategy in the context of emotion recognition. Let us take Singh et al [93] as an example.

c) Approach fased on template matching: Learning standard examples of faces allows for the detection of full faces or parts of faces. The judgment is based on the correlation between the input photos and the recorded examples (templates) [88].

c.1) Faces predefined on faces: This technique is used to identify objects, and it's particularly interesting for face detection because of how simple it is to utilize. This method's idea is based on a comparison of any image with a specified model, to calculate the correlation and arrive at a yes/no result. Pixel-by-pixel matching is used.

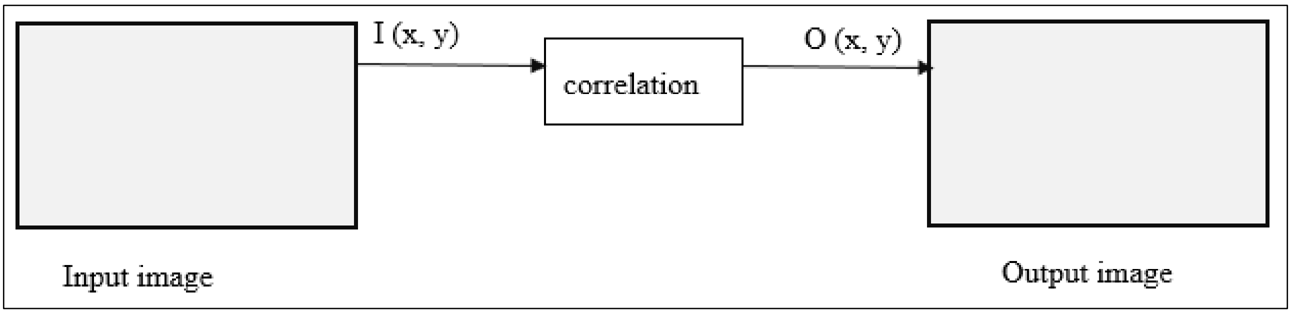


Figure 32 Matching approach [91].

This approach has the benefit of being easy, but it is heavily influenced by scale, position, and shape variations [88].

c.2) Deformable templates: This method is used to create facial features that are elastically adaptable to the current face model [93].

Face features are described using parameterized templates in this manner. To match the parameters on the templates, a function is constructed to connect the contours, vertices, and angles in the input image. Finding the energy function by minimizing the parameters is the best adaptation of the elastic model [93].

d) Approach based on appearance: Using statistical analysis and learning approaches organized through distribution models or a discriminant function, this methodology seeks to determine the significant properties of faces and non-faces. A random variable x (derived from an image or a characteristics vector) represents the face or non-face categorization. Among the approaches employed in this area, we cite: Viola-Jones, neural networks, support vector machines, models concealed from Markov HMM... [88].

d.1) Eigenfaces: In 1991, Turk and Pentland were the first to design Eigenface, which went on to become one of the most well-known face identification algorithms. The principle of this method is to project an image in space, then calculate the Euclidean distance between the original image and its projection, then use image coding in space to degrade the information contained in the image, then compare the distance to a threshold set a priori, if the loss of information is greater, the image is not well represented in space and does not contain a zone of a face or a non-face class. The benefit of this method is that it produces highly positive findings, but it takes a long time to calculate [92].

d.2) Support vector machine (SVM): The SVM method is one of V. Vapnik's statistical learning methods, which he proposed in 1995. The detecting technique is built around training photos, thus no prior knowledge of the image is required.

The basic idea is to establish a face's characteristic vector as a set of brightness values picked over a predetermined window, then learn the face and non-face classes. A hyper-surface that correctly classifies the data and is as far away from all the examples as possible characterizes the separation decision. Osuna et al. used the support vector machine as one of the earliest algorithms for face detection. SVM is a novel type of learning classifier that uses a polynomial function, a neural network, or a radial basis function (RBF). The majority of learning classifiers (Bayesian, neural network, RBF) are based on minimizing the learning error, or empirical error; however, SVM uses a different principle called structural risk minimization, which seeks to reduce greater leaps on likely generalized errors [90].

d.3) Neural networks: To categorize image pixels as face or non-facial, a neural network is used. A network topology must be designed for any use of neural networks. The latter is determined through a series of tests, and there is no set process for determining the best.

The eyes, nose, and mouth are the most distinguishing features of a face. As a result, the basic topology will be a final unit that provides a binary or probabilistic response. We'll hide the network's hidden layers behind this unit, which is referred to as a fundamental topology because the number of units, their size, and their position are all non-empirical and cannot be precisely fixed [91].

Multilayer perceptrons (PMC) are the most common and basic neural networks today, consisting of a series of 9 layers that are completely or partially interconnected.

The complete learning algorithm will entail sending the expected result to all PMC, each of which will process one unit. If the PMCs are to learn a face as an example, the initial phase is to teach the eyes, mouth, and nose. Following that, each PMC runs its own learning algorithm. The problem with this strategy is that it takes a long time to compute, which prevents real-time processing [94].

Rowley et al's neural network-based face detection algorithm (Figure 33) is broken into two steps[92]:

- Face localization.
- Verification of the results.

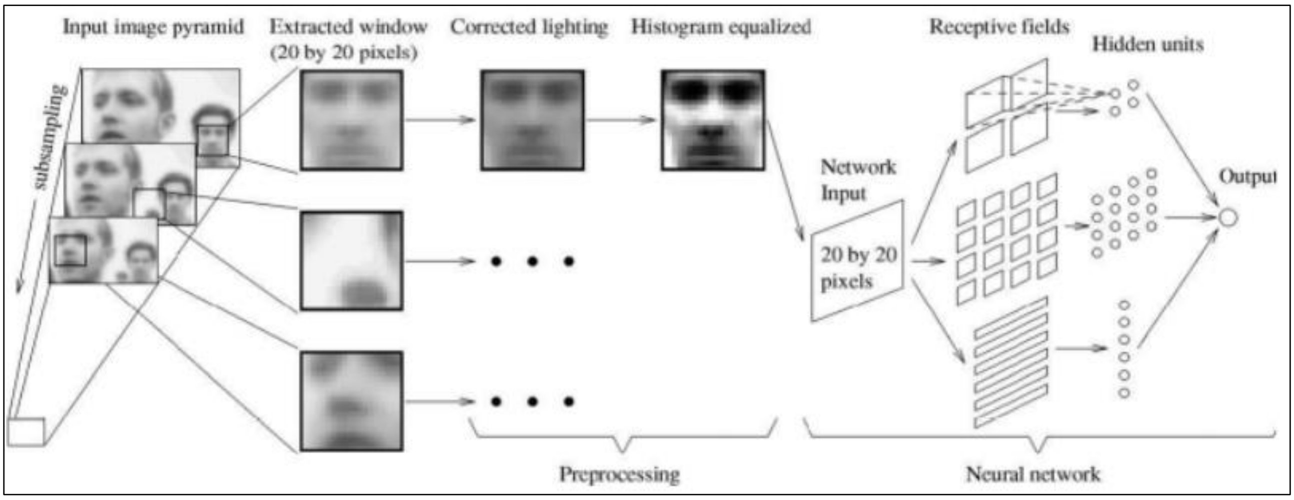


Figure 33 Rowley and al model [91].

d.4) Distribution-based methods: The method based on the distribution for face detection was created by Sung and Poggio, and it shows how the distribution of images belonging to a single class of item can be classified as an example of a positive or negative class. This system is made up of two parts: the first is a face and non-face distribution example model, and the second is a multilayer perceptron classifier. Each example face and non-face is first normalized, then scaled the picture (19 x 19 pixels) to consider it as 361-dimensional vectors. Using the modified K-means method, these structures are grouped into six bouquets of faces and non-faces [91].

The first distance is the Mahalanobis normalization between the test models and the clusters, and the second is the Euclidean distance between the test models and their projections in a 75-dimensional subspace. The final stage is to identify a face and non-face windows with a multi-layer perceptron network (MLP) [90].

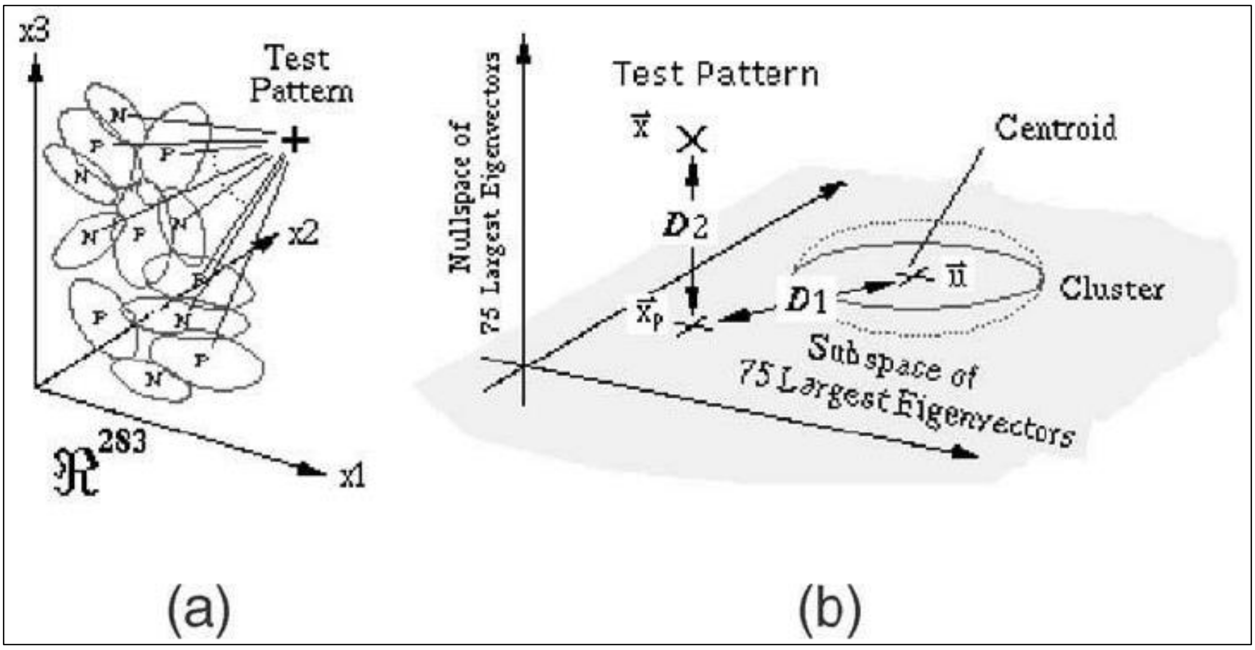


Figure 34 The distance measures used by sung poggio [91].

e) Comparison of methods

To get a general notion of their performance, here is a summary chart of the benefits and drawbacks of various methods (Table 11) [95].

Table 11 Comparison between the different methods [95].

Method	Advantages	Disadvantages
Colors (invariable characteristics)	1) Speed 2) Effective skin detection	1) Poor eye detection 2) Conflicts with the background
Template matching	1) Conceptually simple	1) Multi-scale research

	2) Similarity measurement	2) Filtering multiple detections 3) Low precision 4) Representative model
Eigenfaces	1) Template less crucial 2) Less sensitive to lighting- 3) Similarity measure	1) Same as template matching
SVM	1) Simple to implement	1) low performance 2) greedy in time and precision
neural networks	1) Machine learning 2) Ability to generalize	1) Low precision 2) Multi-scale research

3.4.2 Face recognition systems

Facial recognition algorithms are created from facial feature extraction algorithms, which are divided into three categories: local, global, and hybrid.

a) Global methods: The principle is based on well-established statistical analysis methods. Apart from normalizing the photos, it is not essential to detect specific characteristic points of the face (such as eye centers, mouth centers, and so on). The facial images (which can be considered as matrices of Pixel values) are processed globally and converted into vectors, which are easier to manipulate, in these methods.

Global methods have the advantage of being reasonably quick to implement, and these basic calculations are of medium complexity. They are, however, extremely sensitive to changes in lighting, position, and facial expression [96].

a.1) Principal component analysis (PCA): The ACP algorithm was created in 1991 as a result of the work of MA. Turk and AP. Pentland at the MIT Media Lab. It consists of expressing the "M" images of the starting learning base according to a base of particular orthogonal vectors - the eigenvectors - containing information independent of one vector to another. As a result, the new data is expressed in a style that is more suitable for face recognition [97].

We want to extract the attributes of the image of the face with this method, then encode it efficiently so that we can compare it to a database of images encoded with the same method.

This entails determining the eigenvectors of the covariance matrix created by our learning base's various images. As a result, the PCA requires no prior knowledge of the image and is more successful when used in conjunction with the MahCosine distance measurement, but its ease of use is offset by a high sensitivity to changes in illumination, position, and face expression [98].

a.2) Linear discriminant analysis (LDA): In 1997, this algorithm was created. It is known as Fisher-Faces at Yale University in the United States, and it uses a separation of classes. To use it, the learning base of the image must be organized into numerous classes, one class per individual, and several photos per class.

The LDA examines the data scatter matrix's eigenvectors to maximize variations between images of different persons (interclass) while limiting variations between images of the same person (intraclass). The Fisher-Face method entails locating an appropriate location on which the images of the learning base and the test base can be projected [99].

The identification is carried out by comparing the test image's projection with each of the learning base's image projections. G is the vector of dimension N that corresponds to an image I of the learning base, which is made up of M images, as before.

PCA can be used to tackle a variety of classification and dimension reduction problems, as well as discriminant analysis of classes not covered by PCA.

However, in the case of data with very large dimensions, this method cannot be applied immediately without first lowering the data's dimension. Instead of using the images' pixel values directly, an ACP is applied to the data first, and the data is then utilized to represent the images in the face space [99].

a.3) Neural network: Rowley introduced this technique, which is based on neural network classification to recognize faces and works in two stages:

- The first step is to apply the filters to the entire image to investigate each region in search of the face.

- Individual filter detections then eliminate overlapping detections.

The first component of this system is a filter that takes a 20x20 pixel portion of a picture as input and outputs a value between 1 and -1, indicating the presence or absence of a face. To detect faces larger than the window size, the filter is applied to each position in the image, the input image is continually scaled down (by downsampling) and the filter is applied to each size [97].

A position and an invariance scale are required for this filter. How many scales and positions it should be applied to is determined by the quantity of invariance. The resistance of neural networks to noise is one of their advantages. However, neural networks are sometimes difficult to construct since their structure (for example, the number of hidden layers in perceptrons) has a significant impact on the outcomes, and there is no way to automatically detect this structure.

The learning phase is tough to complete since the examples must be carefully selected (both in terms of number and configuration) [100].

a.4) SVM: Vapnik and coworkers proposed support vector machines (SVM) as a very efficient solution for general-purpose pattern detection in 1995. The SVM finds the hyperplane that separates the largest possible part of points of the same class and on the same side, increasing the distance of either class to the hyperplane as much as possible, minimizing the risk of classifying not only the training sets but also the unseen test examples in an intuitive way [97].

In the realm of face recognition, the SVM technique has demonstrated that trained discrimination functions can provide substantially higher identification accuracy than the widely used standard eigenface approach, in which eigenfaces are employed to represent images of the face. face.

The SVMs learn the discrimination functions between each pair after the features have been retrieved. The disjoint test is then used to define the recognition system [101].

a.5) Gaussian mixture (GMM): Conrad SANDERSON et al proposed the GMM technique, which mainly leverages local features (i.e. features that describe only part of the face).

This is in contrast to general features, which are used in the PCA-based technique and describe the complete face with a feature vector. By evaluating a face block by block, local features can be extracted [97].

Each block is frequently subjected to feature extraction using the Discrete Cosine Transform (DCT), which yields a set of feature vectors. Then, using a linear combination of many Gaussians to represent a human model, model their distribution.

This method is based on the HMM technique, in which the spatial relationship between the main features of the face (such as the eyes and the nose) is preserved; however, in the GMM approach, the spatial relationship is effectively lost (because each block is processed independently), providing good robustness to imperfectly located faces. It has proven to be surprisingly effective, especially in terms of accuracy and execution time [102].

a.6) The statistical approach and the probabilistic approach: To tackle classification and classification problems, the approach relied mostly on decision theory, with classification based on Bayes' theorem and widely employed. Yang and Ahuja suggested a method for recognizing human faces in color images. Where a multi-variant statistical analysis model of human skin color is created to capture color attributes and a probability method for recognizing human faces is described utilizing a mixture of factor analyzers, however, the disadvantage of this approach is the high processing cost [103].

b) Local methods: Face recognition in local approaches is based on local facial traits. Rather than a single high-dimensional vector, the face is represented by a set of low-dimensional feature vectors in these algorithms.

The methods are divided into two groups depending on their geometric and graphical approaches.

b.1) Basic points of interest: These algorithms, which are among the oldest in face recognition, detect points of entry and then cite the characteristics found on these sites of interest.

They are all focused on extracting specific geometric features like the width of the skull and the distances between the eyes. This information is then used by classifiers or individuals to recognize them [104].

b.2) Methods based on facial appearance: The face is partitioned into small parts with these methods, and the local properties are extracted directly. And we use it in a modular approach for different facial areas, then the global model is defined by combining the individual local models, and this division ensures that the different facial regions are independent in the case of occlusion by various sources of variability [97].

Wearing sunglasses, for example, has a significant impact on the eye area, but smiling has a greater impact on the mouth area. The form (rectangle, ellipse) and size of the local regions of the face are utilized to define them. An examination of gray level values, as well as techniques such as Gabor or Harr wavelets, and fractal analysis, are used to discover the features of the local regions.

Gray-value-based features, on the other hand, preserve texture information while Gabor features are more resistant to changes in light and geometric transformations [105].

c) Hybrid methods: It uses global and local aspects of faces to improve face recognition performance by integrating the benefits of both methods. Global and local features are two distinct qualities that aim to integrate the benefits of both methodologies in order to improve categorization.

The concept of integrating and utilizing one's strengths to compensate for the weaknesses of the other. For the time being, the effective coupling of local and global characteristics is an issue, and there is minimal work on its application to the problem of facial recognition [105].

Table 12 presents the difference between local and global characteristics.

Table 12 The difference between the two types of characteristics [97].

Variation factors	**Local characteristics**	**Global characteristics**
Illuminations	Very sensitive	Sensitive
Expressions	Not sensitive	Sensitive
Laid	Sensitive	Very sensitive
Noise	Very sensitive	Sensitive
Occlusion	Not sensitive	Very sensitive

3.5 Conclusion

Facial recognition can be developed into two separate models, face detection and face recognition.

First, we look at those models and we present the different steps of the facial recognition system and how it is work. Then, also look at some technical literature papers, on the parts of detection and recognition. And we compared these diverse approaches in their respective fields at the end of this chapter.

In the contribution part, we will represent the Solution Modeling and implementation of our system.

Contribution

Chapter 4 Solution Modeling

4.1 Introduction

The main objective of the current work is to exploit security cameras by applying deep learning techniques to decide to improve car access control security.

In this chapter, we will present the different design methods and steps undertaken to achieve the objective already mentioned in an explicit way in the General introduction.

4.2 Design process

We must base ourselves on a technique to be able to build an appropriate solution for the system once it has been delimited and the key principles established. The methodology adopted must respond to the nature of the problem being addressed and must make it easier to start designing the solution.

We chose the CRISP-DM approach for the system in this work, which will be briefly described in the following sections.

4.3 CRISP-DM

The IBM-designed CRISP-DM [1]approach (Cross Industry Standard Process for Data Mining), is widely utilized in data processing fields. It is described as an iterative process that allows for the visualization of the whole data life cycle.

The CRISP-DM method is broken down into six parts, as shown in the diagram below.

• Problem understanding: This phase entails obtaining the best possible understanding of the goals to be met.

• Data understanding: The CRISP-DM data understanding phase entails analyzing the available data to avoid data-related issues in future operations.

• Data preparation: This section entails acquiring, cleaning, processing, and labeling data sets, as well as augmentation and normalization.

• Modeling: This is where the technical choices and configurations of the suggested models come into play. This step is frequently repeated numerous times, with each iteration returning

[1] https://www.ibm.com/support/knowledgecenter/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html

to the data preparation phase to make the changes thought required to improve the models' performance.

• Evaluation: We quantify the performance of each model built using a new dataset on which it has not been trained during this step.

• Deployment: The deployment step of the CRISP-DM technique entails employing our system inside a well-defined framework of use to examine the results collected and debate possible perspectives and changes that may be made.

We were able to arrange our design using the CRISP-DM approach, which highlighted any potential interconnections between the various processes. Between modeling and data preparation, as well as the evaluation and deployment stages, which we will discuss in the future chapter.

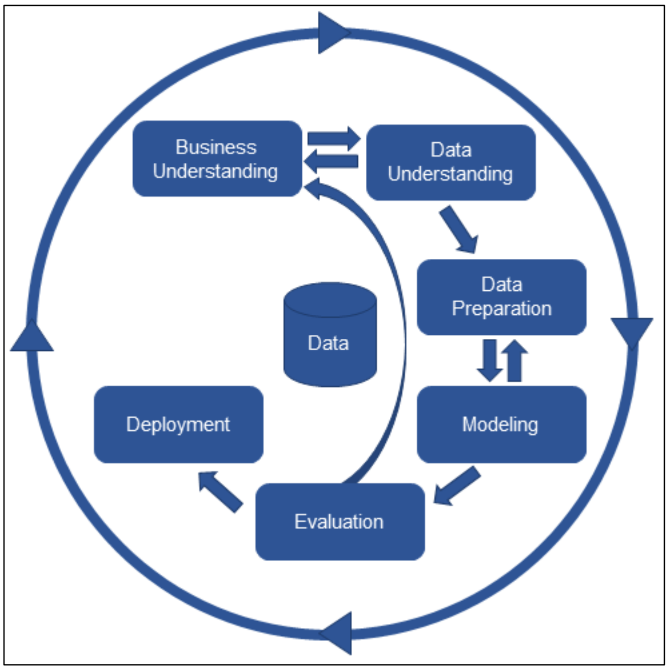


Figure 35 CRISP-DM approach.

4.3.1 Problem understanding

Nowadays, and due to the large number of cars and their continuous increase, it has become impossible to control them, especially in private places such as car parking, where cars can be damaged if unauthorized persons are allowed to enter. and its difficulties for human verification and tracking of all the cars entering, and it is expensive to hire security guards 24/24h.

4.3.2 Proposed solution

Following our investigation of the problem, we are now prepared to provide a solution that will aid in the security of the access control system.

We propose two systems: the first is an automatic license plate recognition system that consists of three subsystems that provide high security and accuracy in a short amount of time.

Face recognition is the second system, which is used to identify the driver if the first system fails or if one of the allowed persons obtains a new car and tries to get access to the private area.

4.3.3 Datasets search

To implement a system based on deep learning technologies, we require a large database to achieve good results and precise predictions, which will allow us to classify our images in the end.

4.4 Datasets collection

All the used datasets to train our models presented in this part.

4.4.1 License plate extraction dataset

As mentioned in the dataset understanding part we need to find datasets containing car images with license plate localization.

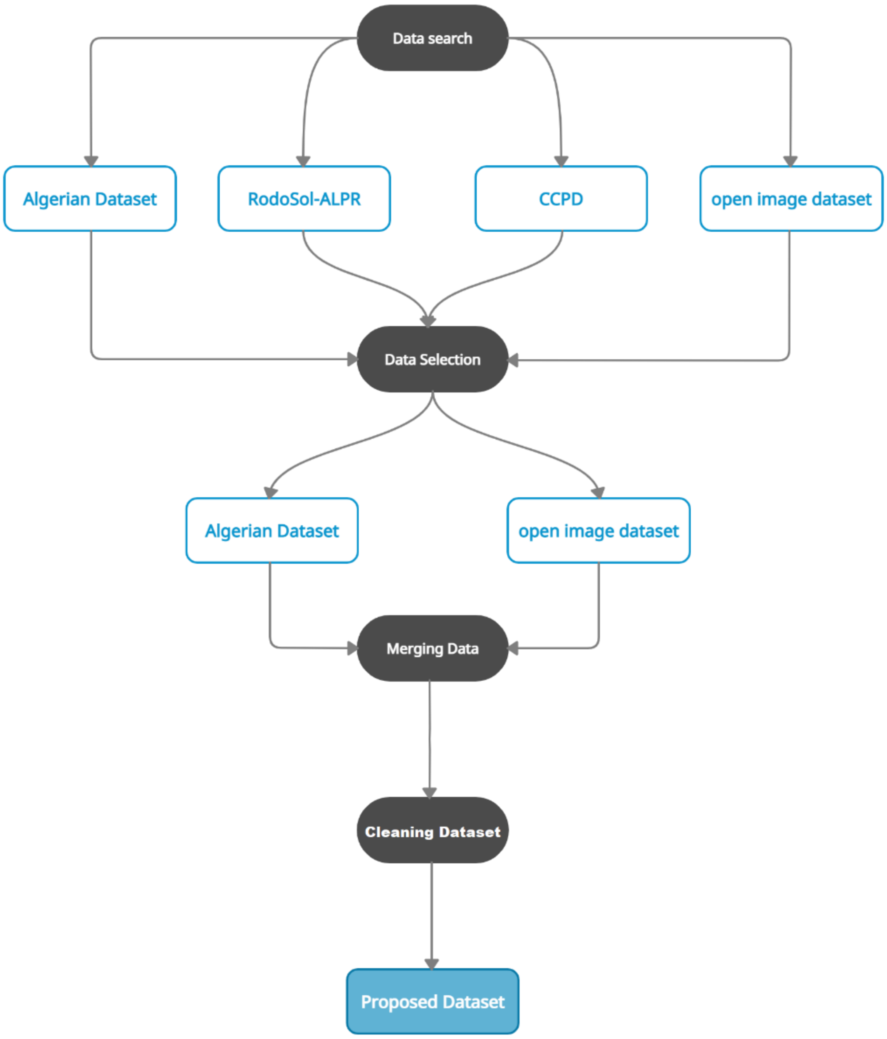


Figure 36 Our LP Data workflow.

a) Dataset collection

First, we conducted in-depth research on data domains related to our topic.

We found many datasets, and after a thorough examination, we came up with a selection. There are some of our founded datasets:

a.1) Chinese city parking dataset

The Chinese City Parking[2] dataset is used to detect and recognize license plates. It has over 250k distinct car photos with geographical annotations for license plates.

Figure 37 Chinese City Parking Dataset.

a.2) RodoSol-ALPR

The RodoSol-ALPR[3]collection contains 20,000 photos collected by pay toll cameras operated by the Rodovia do Sol (RodoSol) concessionaire, which operates 67.5 kilometers of highway (ES-060) in the Brazilian state of Espirito Santo.

Images of various sorts of vehicles (e.g., automobiles, motorbikes, buses, and trucks) were taken at different times of day and night, from different lanes, on clear and rainy days, and the distance between the vehicle and the camera varies slightly. All of the photos have a 1280x720 pixel resolution.

The proposed dataset includes photos of two different license plate (LP) layouts: Brazilian and Mercosur (we refer to "Brazilian" as the standard used in Brazil before the adoption of the Mercosur standard to ensure consistency with current works).

The vehicle type (car or motorcycle), the LP layout (Brazilian or Mercosul), the text (e.g., ABC-1234), and the position (x, y) of each of the image's four corners are all provided in a text file for each image. Instead of only labeling the LP bounding box, we labeled the corners to allow for the training of algorithms that investigate LP rectification as well as the deployment of a larger range of data augmentation approaches.

In response to privacy concerns about our dataset, we point out that in Brazil, the LPs are associated with the relevant automobiles, i.e., no public information about the vehicle drivers/owners

[2] https://github.com/detectRecog/CCPD
[3] https://paperswithcode.com/dataset/rodosol-alpr

is available. Furthermore, in each image, all human faces (e.g., drivers or RodoSol staff) were manually redacted (i.e., blurred).

Figure 38 RodoSol-ALPR.

a.3) Vehicle registration plate open image dataset

It is a subset of the open image V6 Dataset[4] from google and it contains 6867 cars images of different sizes and text files of license plate locations under the name of bounding boxes those bounding boxes are four points (2 start points (x, y) and height, width (h, w)).

Google Open Images is a 9 million-image dataset that includes image-level labels, object bounding boxes, object segmentation masks, visual relationships, and localized narratives.

Has 16 million bounding boxes for 600 item types on 1.9 million photos, making it the largest collection with object position annotations currently available. To ensure accuracy and consistency, the boxes were mostly drawn by hand by skilled annotators. The images are diverse, and many of them feature intricate scenarios with multiple items (8.3 per image on average).

[4] https://storage.googleapis.com/openimages/web/visualizer/index.html?set=train&type=segmentation&r=false&c=%2Fm%2F025fsf

Figure 39 Vehicle registration plate open image dataset.

a.4) Algerian license plate

This data set was collected by Mr. Bensouilah Mouâd and Dr. Zennir Mohamed Najib [59]. It contains Algerian license plates[5] of 2,289 train images, 436 for testing and 118 for validation, and from Facebook (meta) MarketPlace and Ouedkniss, they annotated those images using the "BBox-label-tool" provided by Qiu[6].

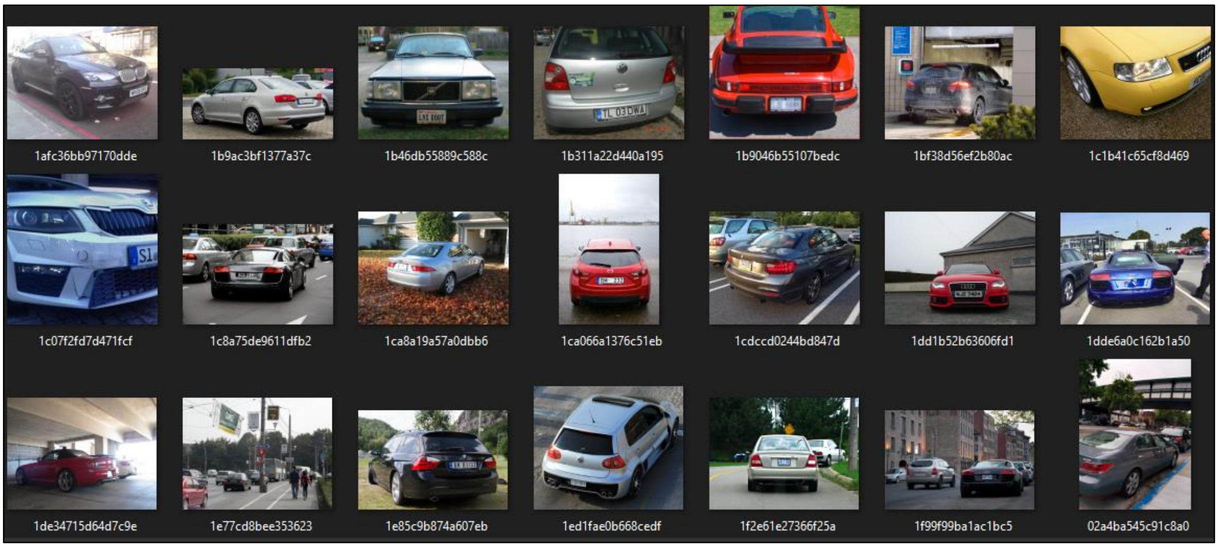


Figure 40 Algerian dataset.

[5] https://github.com/mouad12345/License_Plates_of_Algeria_Dataset/tree/master/Detector
[6] https://github.com/puzzledqs/BBox-Label-Tool

a.5) Proposed dataset

After taking a look at the dataset mentioned previously, we noticed that the Algerian license plate dataset shape is not satisfying to get high accuracy so we decided to merge it with another dataset.

After that, we can see that the Vehicle registration plates from the open image dataset features are so close to the Algerian ones in the size of the license plate.

So that was the reason behind merging the open image dataset with le Algerian to create our Dataset.

Both datasets as mentioned before contain labels with a bounding box in text files.

To finalize our dataset, we had to transform this type of label into another form (text to image).

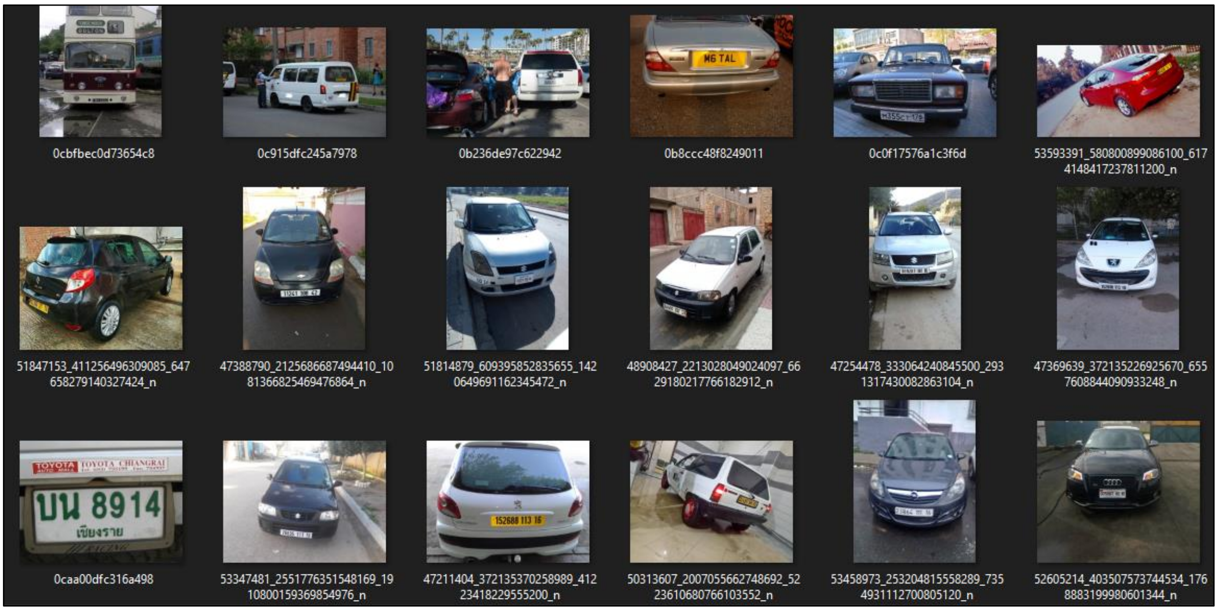


Figure 41 Our proposed Dataset.

a.5.1) Cleaning dataset

It is necessary to clean our databases utilized earlier during this stage. Choosing which data to maintain in our dataset and which to remove. To do so, we will develop the following selection criteria:

- As many images without a license plate as feasible should be avoided.
- The image should be crisp and clear.

• Remove any images with poor visibility due to interference (such as human interference).

Figure 42 Sample of removed images.

4.4.2 Optical character dataset

Figure 43 explaines the OCR dataset selection. The process explained in the next points.

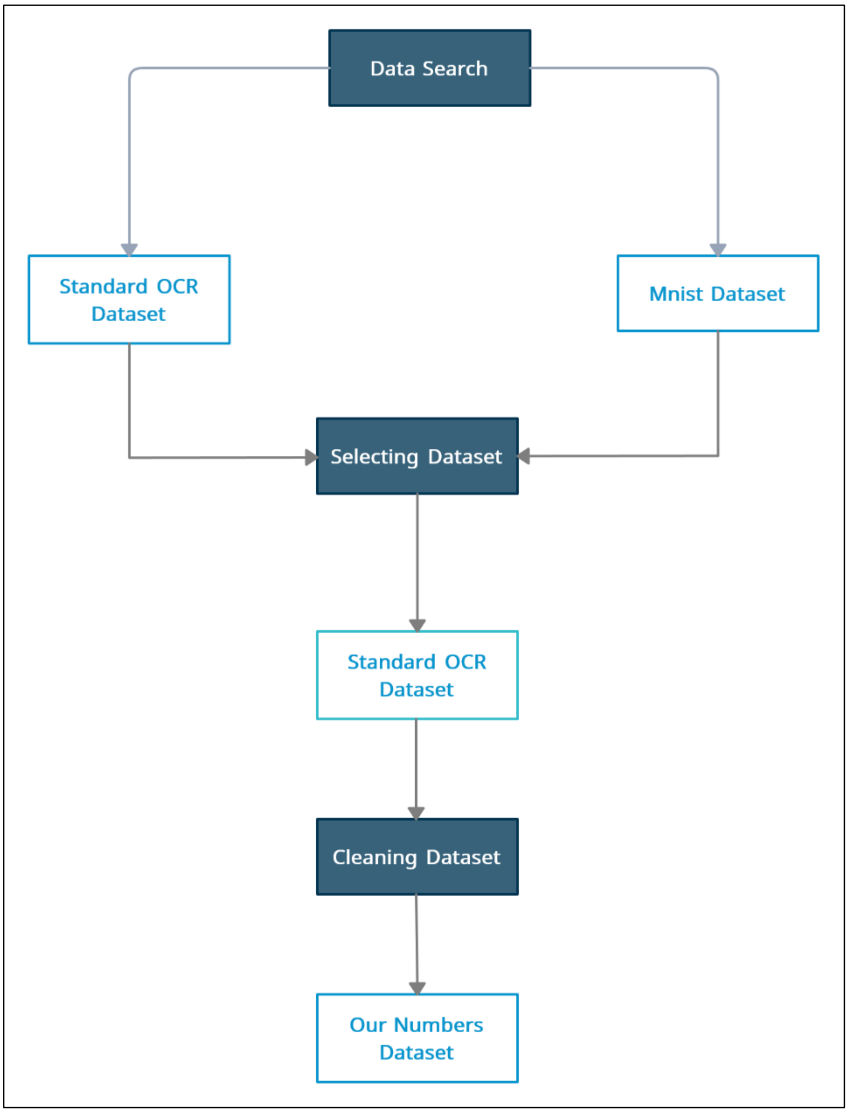


Figure 43 Our Numbers Dataset workflow.

a) Data collection

First, we conducted in-depth research on data domains related to the character recognition topic.

We found many datasets, with different characteristics, and after a thorough examination. We made our choice by taking the type of problem we want to solve into consideration.

There are some of our founded datasets:

a.1) MNIST dataset

The MNIST[7] (figure 44) collection of handwritten digits contains a training set of 60,000 samples and a test set of 10,000 examples. It is a subset of a bigger list provided by NIST. In a fixed-size image, the digits have been size-normalized and centered.

It is a fantastic database for folks who want to experiment with learning techniques and pattern recognition methods on real-world data with minimal preparation and formatting.

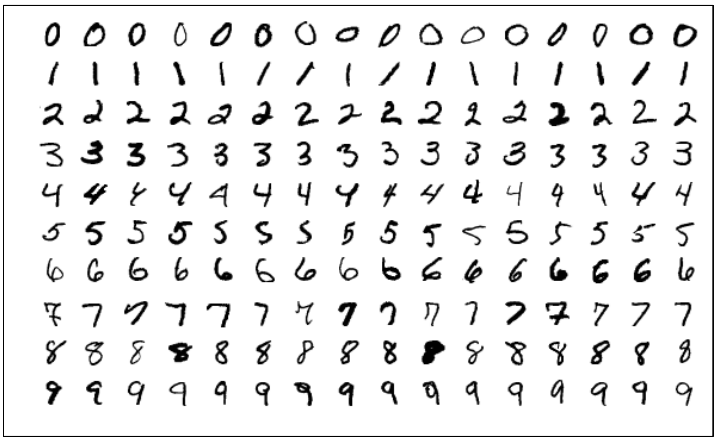


Figure 44 Mnist Dataset sample.

[7] http://yann.lecun.com/exdb/mnist/

a.2) Standard OCR dataset

Various Fonts and Styles Optical Character Recognition Dataset. This dataset includes 26 letters (A-Z) with a total shape of 15626 (601 pictures for each character) and ten numerals (0-9) with a total shape of 6010. (601 images for each character).

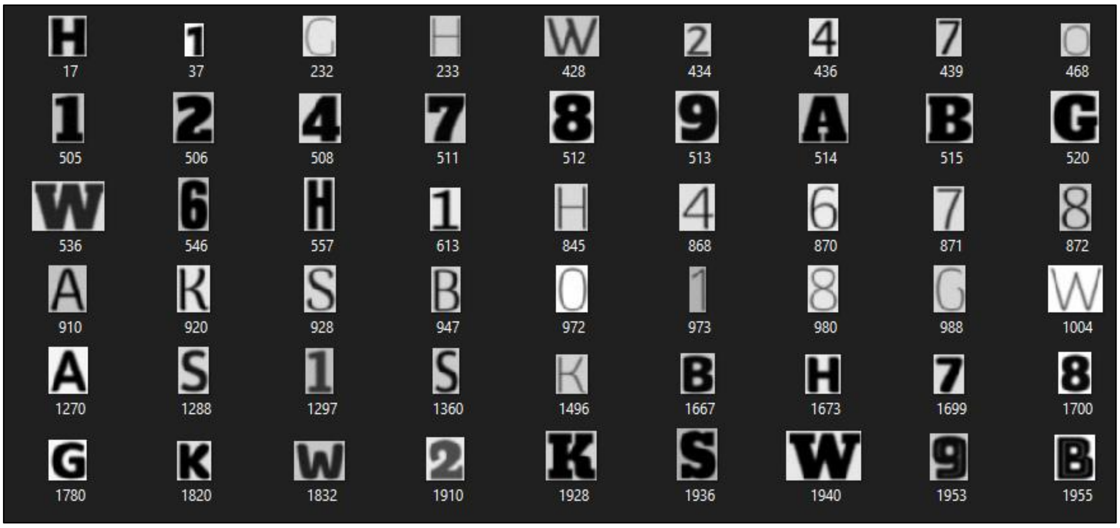


Figure 45 Standard OCR dataset.

a.3) Selected dataset

To select the dataset, we faced the choice between the mnist Dataset and Standard OCR Dataset and by studying both datasets we decided that the 2nd Dataset is the best for us to work on it because the mnist is a handwriting dataset. Even if the OCR dataset was the best one but we needed to adapt it to the Algerian license plate number characteristic mentioned in the 3rd chapter by removing all the letters.

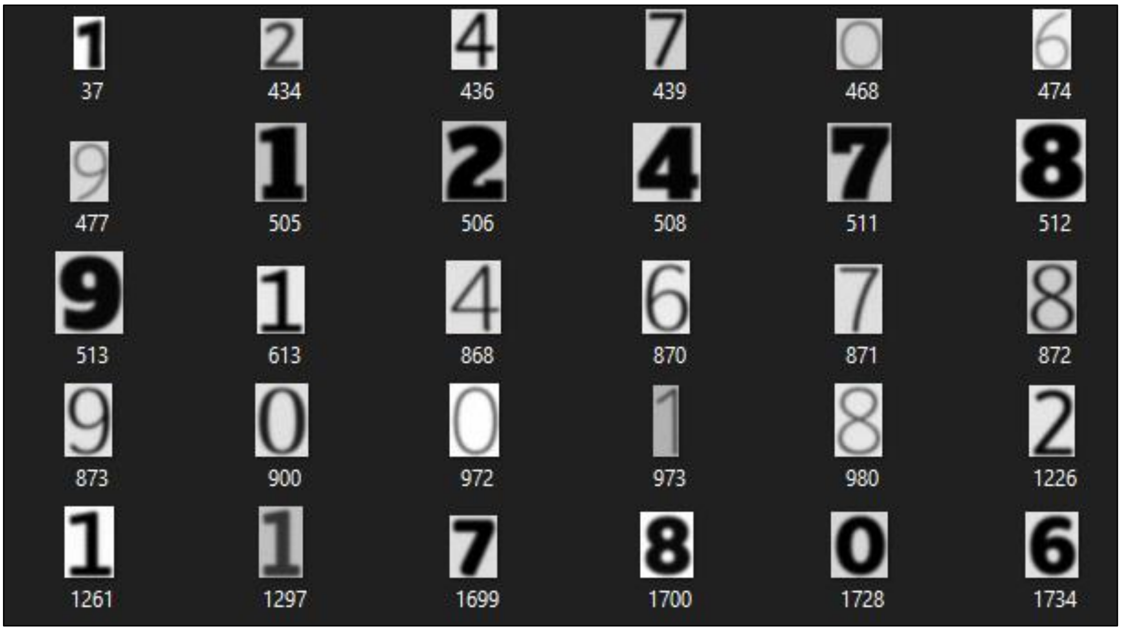


Figure 46 Final dataset.

4.4.3 Anti-spoofing dataset

a) Data dollection

After thinking about the spoofing problem. we did some research to solve it. we found that there is some project working on the same problem so we use one of those projects' datasets. and we augment it.

a.1) Used dataset

The anti-spoofing dataset[8] contains real and spoofs face images (two classes), with 5172 images. Figure 47 shows a sample of this dataset.

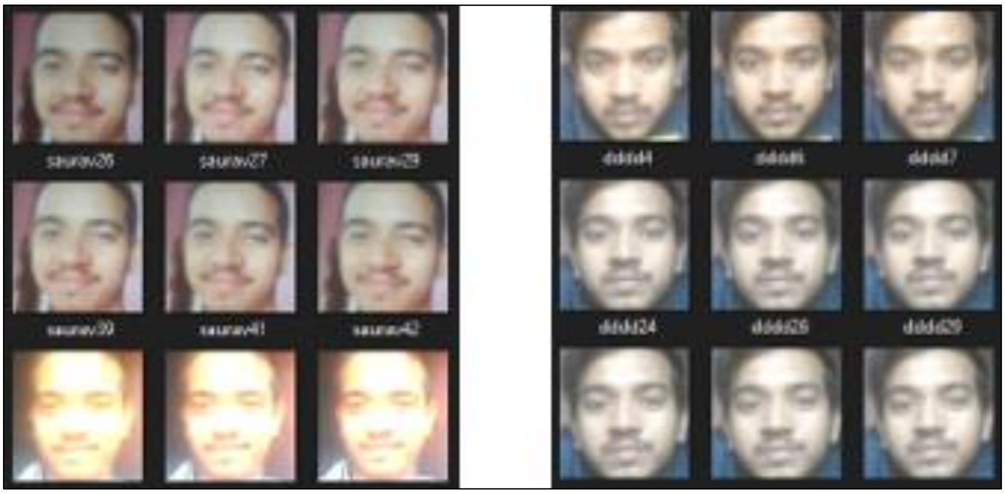


Figure 47 Sample of spoofing dataset.

4.5 Access control system

The deep learning models that we must develop are the foundation of our system. These models must be evolved using data until their findings are satisfactory for them to be ready to accomplish their roles in our system.

We may claim that the training phase has been successful when the model results are at a level that ensures the work.

In this section, we willl look at the models and steps that have been tailored to our two proposed systems. Figure 48 present our general access control system. The process in the next figure explained in the next steps.

[8] https://drive.google.com/drive/u/0/folders/1-SWLkGAi34e5ef3tZxOun2hlNk4T4Df3

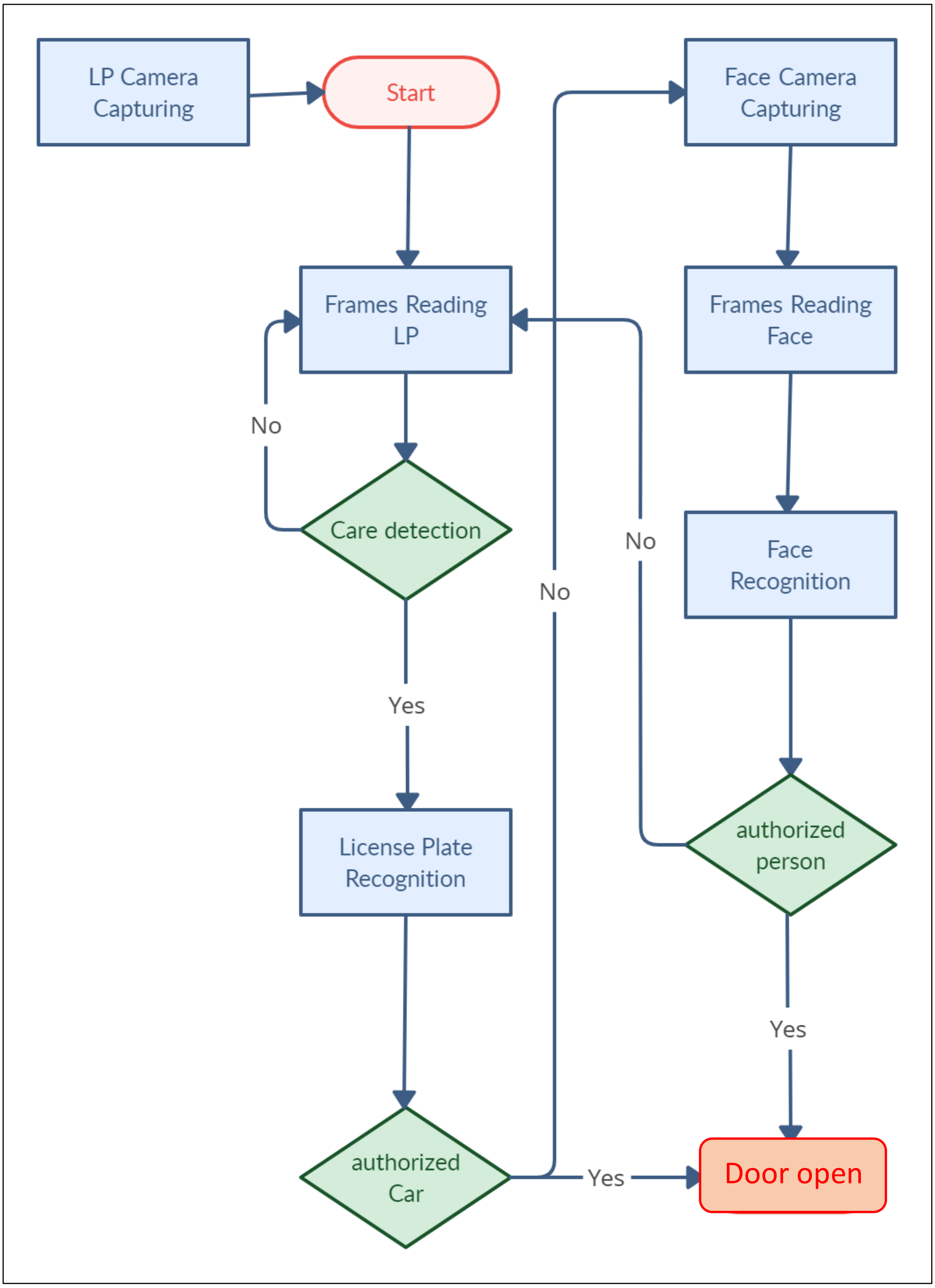


Figure 48 Architecture of our proposed system.

4.5.1 Car detection

As previously noted, the multi-stage method's solution begins with car detection, which is thoroughly discussed in chapter 5 page 90 with the intention of triggering the processing of license plate recognition to avoid the system activating for no reason as we see in figure 49, using a security camera video as input.

Figure 49 Car detection main reason.

4.5.2 License plate recognition system

a) License plate extraction

When the previous system detects a car license plate detection model launches automatically to be able to locate the license plate in the frame image, we will use an architecture (models) appropriate to our situation, those models which are parts of the architectures of convolutional neural networks, they are used to ensure semantic segmentation in our images.

Figure 50 Car image before LP extraction.

Figure 51 LP after extraction.

b) Characters extraction

After getting the License plate image from the extraction model, we segment it by characters (Figure 52) for the purpose of classifying registration numbers.

Figure 52 Segmented characters.

c) Optical characters recognition

As we already said above the automatic license plate recognition system ends with the optical character recognition model.

This model uses the numbers selected dataset we mentioned before in its chapter to train on the character recognition system using CNN architecture.

In this part, we use the segmented characters from the previous part as an input to this model to predict the final license plate number.

4.5.3 Face recognition system

This system was made to avoid security problems and ensure that authorized persons could access even if the license plate recognition model failed to recognize the car.

This system is a combination of three models:

a) Face detection

Face detection it is the first step to creating our recognition system. in this step we take an input frame from a video camera to get the face localization and extract it, for this task we will use the face cascade Classifier.

b) Anti-spoofing detector

One of the most popular challenges in face recognition systems is The ability to detect identity theft. So, in this subsystem, we try to solve it by creating a model that can detect this problem using mobilnetV2 architecture to classify the validity of input faces from the output image got using the face detector subsystem.

The 3 possible types of spoofing are shown in the next figure.

Figure 53 Example of spoofing types.

c) Face recognition

This is the last part of the face recognition system, it takes as an input a valid face (none spoof) to identify whether the person is authorized or not to access specific areas.

The model of face recognition is built by the architecture of facenet**.**

4.6 Model evaluation

It will be necessary to evaluate the model's performance on a fresh dataset known as the "Test set" once it has been trained. These data were not included in the training and will be used to determine if a model generalizes effectively or suffers from under-learning or over-learning.

We use in the model evaluation 5 metrices (precision, recall, F1score, Accuracy, and loss) to describe a classification models performance on a set of test data for which the real values are known.

In a classification test, we classify items into one of two categories: positive or negative, with correct or erroneous results. We differentiate four possible results combinations for this:

- True positives (TP): When a positive element is accurately categorized, the result is considered to be true positive.
- False positives (FP): When an element is labeled as positive when it is not, the result is said to be false positive.
- True negatives (TN): When a negative element is accurately categorized, the result is considered to be a true negative.
- False negative (FN): When an element is classed as negative even though it is positive, the result is considered to be false positive.

While we evaluate our system models, we used some metrices:

- **Precision**: It is an excellent means to assess when false positives are high based on how precise the model is out of the anticipated positives (how many are truly positive).

  If the precision is not high enough, the algorithm can be confused between classes.

$$Precision = \frac{\text{True Positive}}{\text{True Positive+True Negative}}$$

- **Recall**: This is a measure of a class's quality. We shall divide the number of well-classified elements in the class by the total number of individuals who truly belong to the class because there is only one.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+False Negative}}$$

- **Accuracy**: It is a metric for how frequently the algorithm successfully classifies a data item (how many true positives and true negatives were predicted in total).

$$\text{Accuracy} = \frac{\text{True Positive+True Negative}}{\text{Total Number of Predictions}}$$

- **F1 Score:** If we need to find a balance between Precision and Recall AND there is an unequal class distribution, F1 score could be a better metric to employ.

$$\text{F1 Score} = \frac{2\ x\ Precision\ x\ Recall}{\text{Precision+Recall}}$$

After this review, we may either deem a model to be efficient and export and archive it, or unacceptable and redo the training stage with a new network design if necessary.

4.7 Data management interface

To finalize our work and to facilitate our system management by updating authorized cars and persons we want to make a simple Graphical user interface. It will be used by the admin who will be able to manage the license plates and faces database.

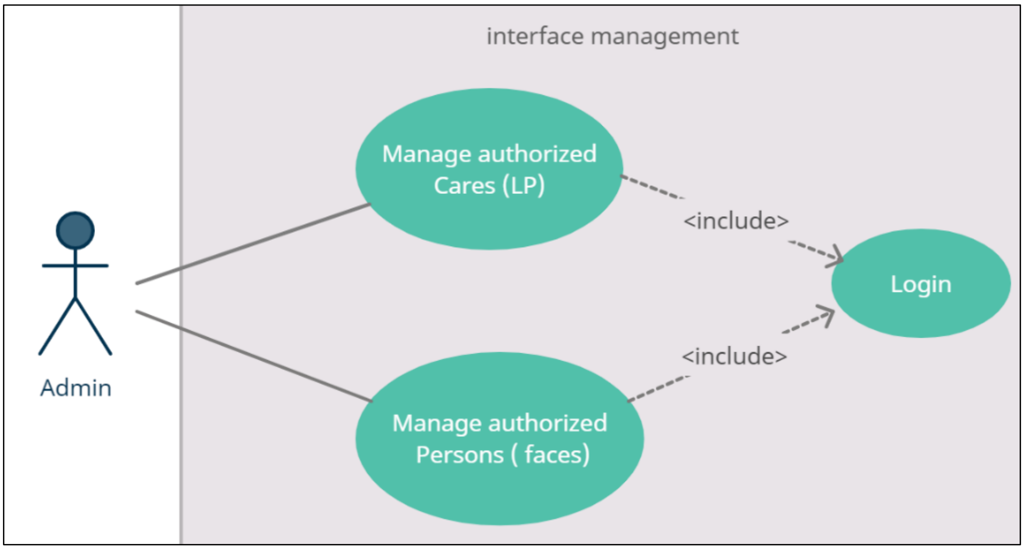


Figure 54 Use Case Diagram.

4.8 Conclusion

We have covered the entire methodologies we utilized and the expected outcomes, in this chapter.

We have outlined the design phases that will enable us to implement the proposed solutions to our problem. These ideas are realized in a variety of ways and provide distinct outcomes.

In the next chapter, we will look at our ALPR system's implementation stages and examine the outcomes of these solutions and the results.

Chapter 5 License plate Implementation & Evaluation

5.1 Introduction

In this section, we've looked at and tested a variety of license plate recognition systems and facial techniques to find the best solution to our situation.

It is a description of our project's conceptual study. We'll start by justifying our system selection, then show the overall architecture and the various phases involved in completing this project.

5.2 Development environment

We used this environment to implement our license plate recognition system to achieve the described goals.

5.2.1 Local ressources

Table 13 Local ressources.

Operation System	Processor	Ram	**GPU**
Windows 10	i7 3630MQ CPU 2.40 GHz	8GO	intel HD 4000 2GB

5.2.2 Software environment

a) Gradient Paperspace

Gradient[9] is a platform for building and scaling real-world machine learning applications**.**

Table 14 Used Resources.

Type	**Name**	**GPU**	**Memory**	**vCPUs**	**Price**
CPU	**C5**	**-**	**8GB**	**4 vCPUs**	**$0.08/hr**
GPU	**A4000**	**16GB**	**45GB**	**8 vCPUs**	**$0.76/hr**

9 https://gradient.run/

b) Kaggle

Kaggle[10] provides a configurable Jupyter Notebooks environment that requires no setup.

Table 15 Used Kaggle Ressorcess

Type	Name	GPU	Memory	Free
CPU	**Intel Xeon**	**-**	**16GB**	**N/A**
GPU	**Tesla P100**	**17GB**	**16GB**	**30h/week**

c) Google Colab

For short, Google Research's Collaboratory[11], or "Colab" is a product. Colab is a web-based Python editor that allows anyone to write and run arbitrary Python code.

It is notably useful for machine learning, data analysis, and education. Colab, in more technical terms, is a hosted Jupyter notebook service that requires no installation and provides free access to computer resources, including GPUs.

Table 16 Used Google Colab ressorcess.

Type	Name	GPU	Memory	Free
CPU	**Intel Xeon**	**-**	**13GB**	**N/A**
GPU	**Tesla K80**	**12GB**	**13GB**	**6h/24h**

[10] https://www.kaggle.com/
[11] https://research.google.com/colaboratory/faq.html

- **Jupyter Lab:** The most recent web-based interactive development environment for notebooks, code, and data is Jupyter-Lab[12]. Using its versatile interface, users can create and arrange workflows in data science, scientific computing, computational journalism, and machine learning. Extensions to enhance and enrich functionality are encouraged by a modular architecture.

- **Python**[13] is a cross-platform, interpreted programming language that supports multiple paradigms. It encourages imperative programming that is organized, functional, or object-oriented. It contains robust dynamic typing, garbage collection-assisted memory management, and an exception-handling system. Python is a language that may be used in a variety of situations and can adapt to any type of use thanks to its extensive library. We utilized Python 3.8 to create our system.

- **Visual Studio Code**[14] is a lightweight yet capable source code editor for Windows, macOS, and Linux that runs on your desktop. It contains built-in support for JavaScript, Typescript, and Node.js, as well as a large ecosystem of extensions for additional languages and runtimes (such as C++, C#, Java, Python, PHP, and Go) (such as .NET and Unity)

- **OpenCV**[15] (Open-Source Computer Vision Library) is a free software library for computer vision and machine learning. It was created to establish a standard infrastructure for computer vision applications and accelerate the adoption of machine perception in commercial products. More than 2500 optimized algorithms are included in the library, which includes a comprehensive set of computer vision and machine learning algorithms. OpenCV version 4.5.5 was used.

- **TensorFlow**[16] is an open-source software library for machine learning and deep neural network research that was created by the Google Brain Team. TensorFlow is a library that combines a variety of machine learning and deep learning models and methods. Python is used to provide a user-friendly front-end API for creating structured applications. It was utilized to create our neural networks. '2.9.1' is the version utilized.

[12] https://jupyter.org/
[13] https://www.python.org/
[14] https://code.visualstudio.com/
[15] https://opencv.org/
[16] https://www.tensorflow.org/

- **Keras**[17] is a Python-based high-level neural network API that runs on top of TensorFlow. It was created to enable rapid testing; the ability to conduct high-quality research is critical for getting from idea to outcome as rapidly as feasible. Keras has several advantages, including: - Easy and quick prototyping (user-friendliness, modularity, and extensibility). Convolutional and recurrent networks, as well as combinations of the two, are supported. Compatible with both CPU and GPU.

- **NumPy**[18] is an open-source project that uses Python to do numerical computations. It's a Python library that includes a multidimensional array object, as well as derived objects (such as hidden arrays and matrices) and several functions for rapid operations. '1.22.4' is the version used here.

- **Pandas**[19] is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.

- **Google Drive**[20] is a free cloud storage service that lets users save and access files they've previously posted online. The service synchronizes documents across all user devices, making them easier to access no matter what platform they're on. We used it to save our numerous data sets as well as our project's progress.

- **Tkinter**[21] is the de facto technique to develop graphical user interfaces (GUIs) in Python, and it comes with all standard Python distributions. It's the only framework that's included in the Python standard library.

5.3 Datasets preparation

5.3.1 License plate segmentation dataset

In order to prepare our dataset for the learning steps, we needed to reform the label types to match the needs of the field of the semantic segmentation that was used in our thesis.

We went through many processes.

[17] https://keras.io/
[18] https://numpy.org/
[19] https://pandas.pydata.org/
[20] https://www.google.dz/drive/about.html
[21] https://docs.python.org/3/library/tkinter.html

Figure 55 Car sample.

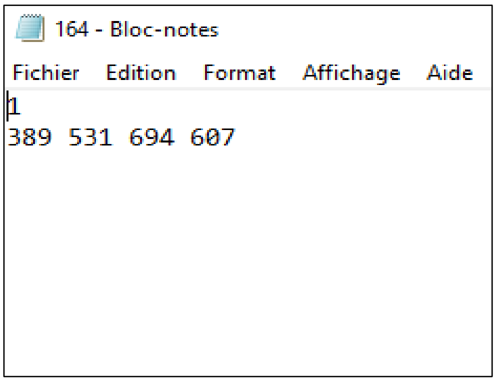


Figure 56 Previous car label.

Figure 57 Shows text files containing the 4 points presenting the location of the license plate bounding box (x, y, h, w). x and y for the license plate start point, the h for height, and the w for the width. the number at the top represents the object class.

Figure 57 Label presentation in a sample from dataset.

a) Data cleaning

The first step we took was deleting non-valuable images as we mentioned in the design chapter. As we can see the first number in (Figure 58) presents the class of the label, this type of label is prepared for the classification using the YOLO[22] model and as we know we have just one class which is the license plate in this step so we deleted each class number manually.

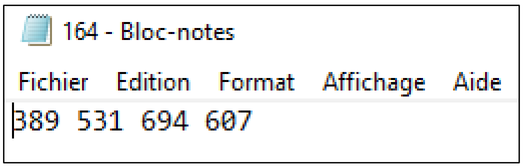


Figure 58 Label ofter manual deleting class name

[22] https://pjreddie.com/darknet/yolo/

b) Labels to masks

While using the semantic segmentation technic we need to transform our labels into image masks (binary images).

Figure 59 shows an example of a cat image with its mask.

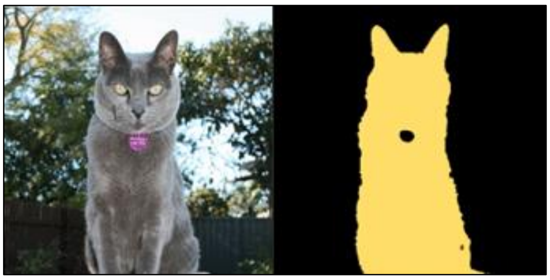


Figure 59 Cat image with the mask [106].

So, for the creation of our mask, we iterate every single image and create a black image with the same shape as the original image using the NumPy array of zeros (Figure 60).

Figure 60 Black image with the same size as the original one.

By using the bounding boxes in the text files (labels) we draw a rectangle in the license plate Perimeter colored white on the black image. (Figure 61) and store each mask with the same name as the original images in the masks folder.

Figure 61 Sample of the final dataset.

c) Data splitting

Next, we loaded the dataset and apply the resizing process for all images to 512x512 pixels. this step entails creating three different sets of training 60%, validation 20%, and test data 20%, which will be used in a variety of ways during the system's implementation.

The training and validation files, which contain different images of automobiles with masks, will be used to train our neural network model, while the test folder will be used to assess the results achieved during training.

5.3.2 OCR dataset

Like every dataset we have to split this data into 3 folders train 60%, validation 20%, and test 20%. and to prepare this dataset for the model training we needed to resize images to 64x64 pixels.

And to get a better result we did an augmentation for this data Figure 62, with a rotation of 5%, zoom of 10%, and brightness range [0.9, 1.1].

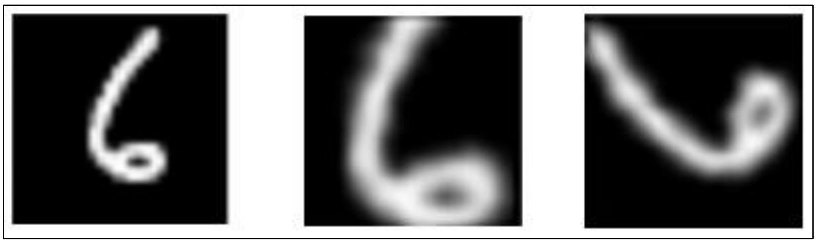


Figure 62 Data augmentation sample.

5.4 Vehicle access control system development

5.4.1 Car detection step

As an input to this model, we take video frames in real-time with shapes 1280x720 pixels or higher for the frame. This model aims to start the license plate detection system.

a) Convolutional neural network model

As explained in chapter 1 CNN model is an architecture for image classification. we decided to try it in our car detector.

So, the real-time video frame will be resized to be high-definition (1280x720) and fitted to the model that makes the classification for car existence (Figure 63).

Figure 63 Sample of CNN result.

b) Object detection algorithm

In this algorithm, many steps are followed to get the final result which is the existence of a car.

- **Movement detector**

It is a cv2 function named createBackgroundSubtractorMOG2 that takes a specific number (250) from a video frame to treat it as a background and colors its pixels with black as Figure 64.

Every change in the next frame pixels (car entry) will be considered as a new object (car detected).

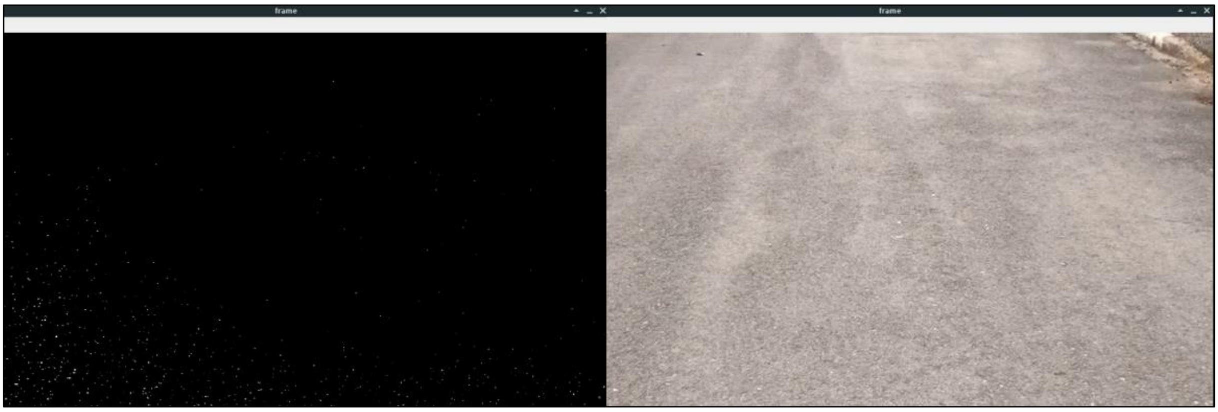


Figure 64 Result of no mouvement detection.

- **Thresholding**

In the previous image, some shadows could be considered noise that could affect the car detection, this shadow appears in the output of the movement detector as gray pixels (Figure 65). So, we had to make a threshold for binarizing our image (black and white), that's means every pixel is not white (255) and transformed to black (Figure 66), to make sure that the shadow vanished to not affect the real object (car).

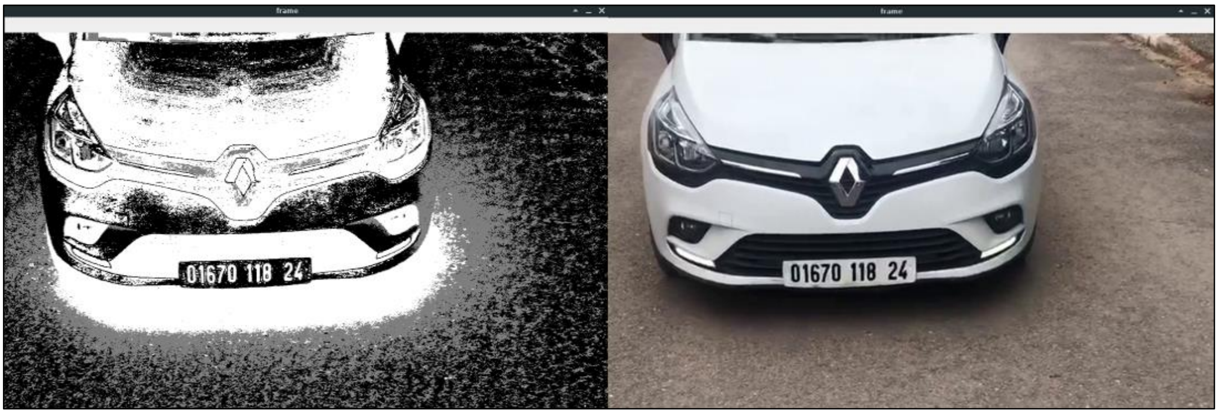


Figure 65 Mouvement detected.

Figure 66 Result of thresholding.

- **Get contours**

Using the thresholded image and to detect the car we tried to get all contours in the last image.

Those contours are defined by 4 points (x,y,h,w), using those 4 points every contour area was calculated the get the largest one to avoid noise area and useless white pixels that could affect our prediction result negatively.

After that, the largest contour area is taken passing it on a threshold that was 20000-pixel x pixel to avoid false objects from detection (cats - humans).

The next Figure shows a sample of contours detected, the largest contour is surrounded by a green rectangle, and the noise with red (Figure 67).

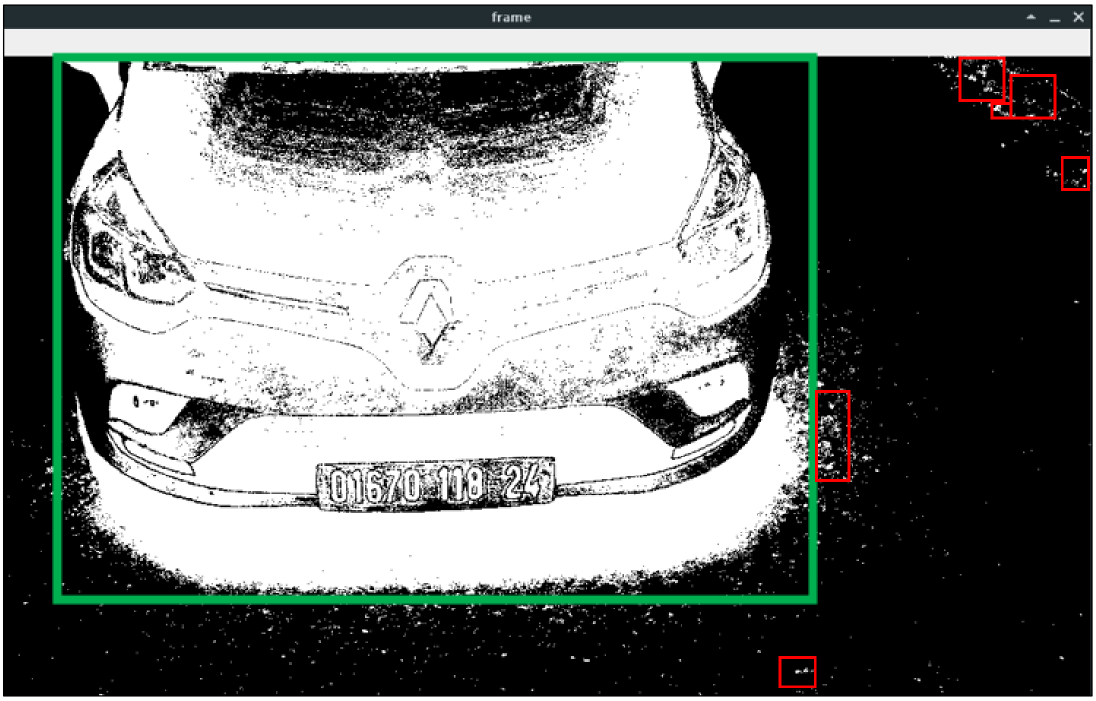


Figure 67 White area detection.

5.4.2 License plate detection models

As a result of the car detection model of car existence, this model will be activated and takes a region from the video frame 720 x 720 pixels. This region is made for 2 reasons:

- **Image corruption:** If we try to resize an image with a high resolution to 512x512 pixels, like Figure 68 the image will be compressed
- **Side objects:** Some cameras could detect objects beside our region of interest, but those objects are mostly unwanted.

Figure 68 Before splitting the region of interest.

Figure 69 Without splitting the region of interest.

a) MobilNetV2-Unet

After the bibliographic research, we did. We found the most appropriate solution for our problem is semantic segmentation technics (the technique of assigning a class name to all pixels in the frame), and as we said before we adapted our data for this type of solution.

One of the famous models for semantic segmentation that gives a good result is the U-Net [107] model.

And for this purpose, we try to adapt this model to get a better and more satisfying result for the LP detection problem.

- **U-Net Model:**

It is a deep learning model (Convolutional Networks) Made to segment images for Biomedical reasons (tumor detection).

The network architecture of U-Net is depicted in Figure 70. It is made up of a contraction path (on the left) and an expanding path (on the right) (right side). The contractual path follows a convolutional network's standard topology. It comprises two 3x3 convolutions (unpadded convolutions) that are applied repeatedly, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with step 2 for downsampling. We quadruple the number of feature channels with each downsampling step.

Each step of the expansive route consists of an oversampling of the feature map, a 2x2 convolution ("up convolution") that halves the number of feature channels, and concatenation with the path contraction's cropped feature map, and two 3x3 convolutions, each followed by a ReLU.

Due to the loss of border pixels in each convolution, cropping is required. A 1x1 convolution is employed at the final layer to transfer each 64-component feature vector to the desired number of classes. The network comprises a total of 23 convolutional layers. It is critical to set the input tile size such that all maximal 2x2 pooling operations are executed to a layer with size x and y pair to enable transparent tiling of the output segmentation map.

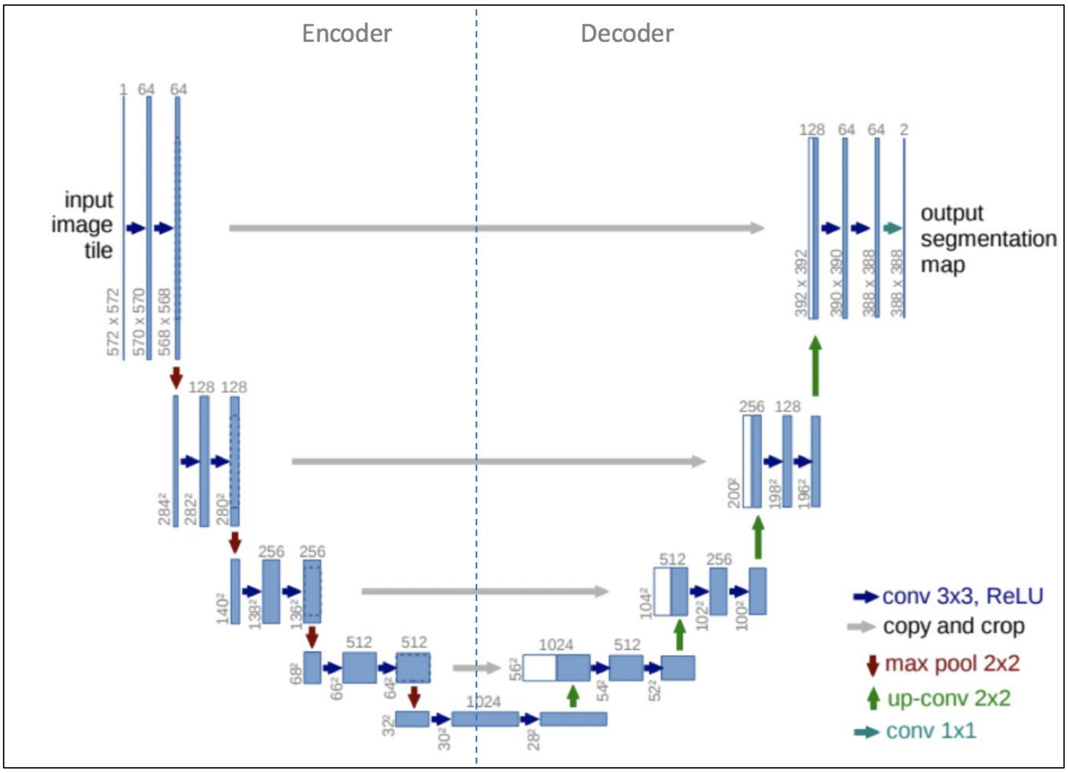


Figure 70 U-Net architecture [107].

In figure 70 we noticed that U-Net architecture is composed of an encoder and decoder. And to improve the Semantic segmentation result we decided to keep the decoder part that gives us output as an image (mask) and replace the encoder with another deep+ learning model for the feature extraction for that operation we tried another mode explained in the next point.

- **MobilnetV2 [108]:**

The encoder was replaced in the U-Net model for two reasons:

The main reason was to improve the feature extraction process in the image got from the image detector mentioned early.

The second reason was to decrease the number of parameters for good stability and better speed performance. After A lot of research, we find that MobilenetV2 is one of the most lightweights models for this type of task and has good results in feature extraction.

Now we will go over the specifics of MobilnetnetV2 architecture. The basic building piece is a bottleneck depth-separable convolution with residuals.

Table 17 Bottleneck residual block transforming from k to k 0 channels, with stride s [108].

Input	Operator	Output
H x W x K	1x1 conv2d, ReLU6	H x W x (TK)
H x W x TK	3x3 dwise s=s, ReLU6	H/s x W/s x (TK)
H/s x W/s x TK	linear 1x1 conv2d	H/s x W/s x K^1

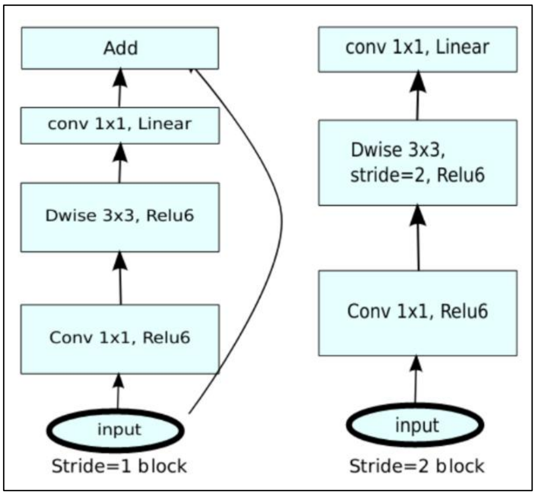


Figure 71 MobilnetV2 stride affect [108].

Table 18 shows the exact structure of this block. MobileNetV2 architecture includes a fully convolutional layer with 32 filters, followed by 19 residual bottleneck layers, as shown in Table 17. Because of its robustness when utilized with low-precision computation, we use ReLU6. During training, kernel size 3x3 is always used, which is common for current networks, and it contains dropout and batch normalization.

In the MobilnetV2 architecture, which is designed to be lightweight, trade-off hyper factors such as the input picture resolution and width multiplier are tunable hyper parameters that can be modified depending on desired accuracy/performance trade-offs. This core network (width multiplier 1, 224 224) uses 3.4M parameters and has a computational cost of 300 million multiply-adds. The cost of computing a network range from 7 multiply-adds to 585 million M-Adds, with model sizes ranging from 1.7 million to 6.9 million parameters [108].

Table 18 MobileNetV2 architecture [108].

Input	Operator	Expansion Factor	channels	iteration	stride

$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

- **MobilNetV2-Unet**

We mentioned before that we use a U-Net decoder and replaced the decoder with mobilnetV2. So, let us go deep-dive into our model architecture.

First, we modified the V2 model by deleting the responsible layer of the classification (output and softmax layers). and we kept the rest of the layers. In this encoder, the image is down-sampled from 512x512x3 to 32x32x192 eliminating the down-sample to 16x16, (we tried to work with 512x512 pixels to get the high-quality image and to guard most of the essential features).

This signifies that the image's height and width have been reduced, while the channel size has been increased (from 3 to 192).

This output contains information about the image (image features). then the output of this model is concatenated with the 1st U-Net decoder layer. and then we did 3 skip connections (are deep neural network connections that transmit the output of some chosen layers from V2 with the opposite layers from the U-net Decoder (contained the same resolution). Table 20 explained the connected V2 blocks with the U-Net decoder.

We decided to use only 4 skip connection layers to reduce the parameter numbers from 4.5M to 416k and eliminate the down-sampling to 16x16 pixel space (last layer) in the encoder and its convolutional bloc from the decoder to improve the speed prediction time (Figure 72).

Table 19 Our MobilnetV2-Unet skip connections.

Layer (type)	Output Shape	Connected to
Input layer	512, 512, 3	up_sampling2d_3
block_1_expand_relu	256, 256, 48	up_sampling2d_2
block_3_expand_relu	128, 128, 48	up_sampling2d_1
block_6_expand_relu	64, 64, 96	up_sampling2d

```
==============================================================
Total params: 416,209
Trainable params: 409,025
Non-trainable params: 7,184
______________________________________________________________
```

Figure 72 Our Mobilnet-Unet parameter numbers.

The main purpose of the decoder is to get features from the encoder last layer (with a low resolution) and apply the upsampling process to get a higher resolution of pixel space. At the same time, we concatenated between skip connection layers explained in table 20. Figure 73 shown MobilnetV2-Unet architectur.

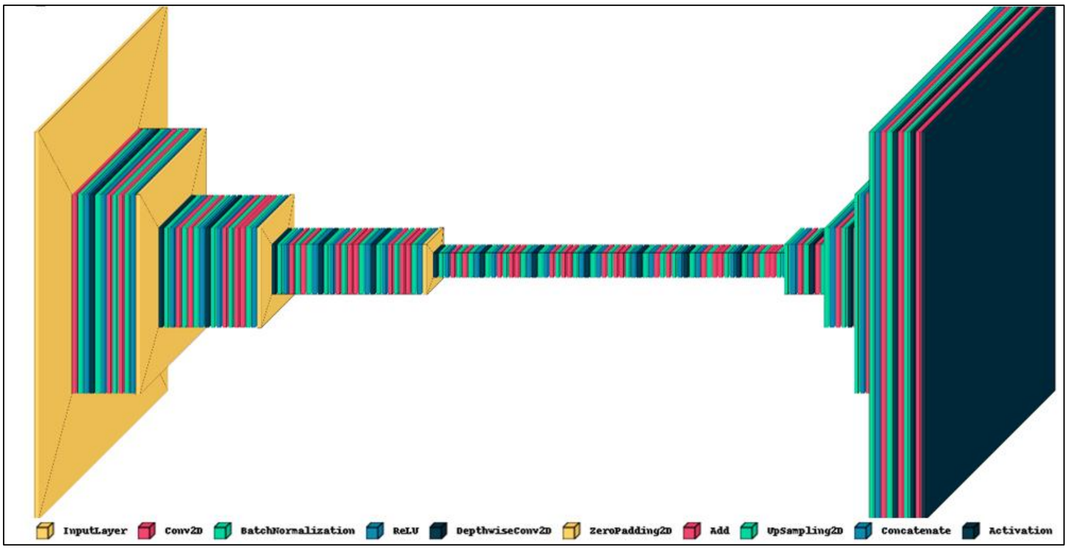


Figure 73 Our Mobilnet-Unet architecture.

b) MobilNetV3-Unet [109]:

For testing more encoders and getting the different results we tried another model which is mobilenetv3 large.

There are two MobileNetV3 models: large and small MobileNetV3. These models are made for low-end resource usage. all details about the V3 network are presented in Table 20 [109].

Table 20 MobilnetV3-large architecture [109].

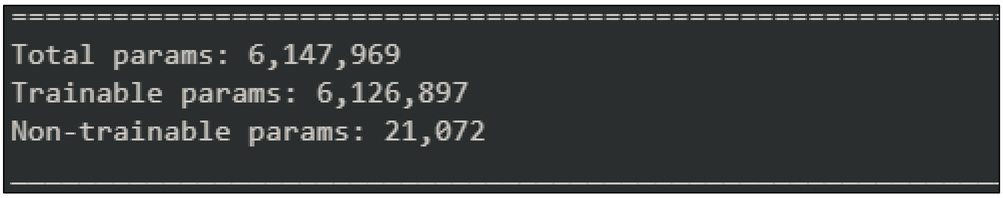
Input	Operator	exp size	#out	*s*
$224^2 \times 3$	conv2d	-	16	2
$112^2 \times 16$	bneck, 3x3	16	16	1
$112^2 \times 16$	bneck, 3x3	64	24	2
$56^2 \times 24$	bneck, 3x3	72	24	1
$56^2 \times 24$	bneck, 5x5	72	40	2
$28^2 \times 40$	bneck, 5x5	120	40	1
$28^2 \times 40$	bneck, 5x5	120	40	1
$28^2 \times 40$	bneck, 3x3	240	80	2
$14^2 \times 80$	bneck, 3x3	200	80	1
$14^2 \times 80$	bneck, 3x3	184	80	1
$14^2 \times 80$	bneck, 3x3	184	80	1
$14^2 \times 80$	bneck, 3x3	480	112	1
$14^2 \times 112$	bneck, 3x3	672	112	1
$14^2 \times 112$	bneck, 5x5	672	160	2
$7^2 \times 160$	bneck, 5x5	960	160	1
72×160	bneck, 5x5	960	160	1
$7^2 \times 160$	conv2d, 1x1	-	960	1
$7^2 \times 960$	pool, 7x7	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	1

Earlier, when we talked about data, we said that we reshaped the images to 512x512 pixels so we were obliged to modify the model input size to the data images shape (512x512) for every image in the encoding phase going from 512x512x3 to 16x16x720 pixel space, and it feeds to the U-Net decoder as we said before.

The number of skip connection layers between them is 5 connections presented in Table 21. This model number parameter was 6.14 million (figure 74).

Table 21 MobilnetV3-Large-U-net skip connections.

Layer (type)	Output Shape	Connected to
Input layer	512x512x3	up_sampling2d_4
re_lu_2	256x256x64	up_sampling2d_3
re_lu_6	128x128x72	up_sampling2d_2
re_lu_15	64x64x192	up_sampling2d_1
re_lu_29	32x32x528	up_sampling2d



```
Total params: 6,147,969
Trainable params: 6,126,897
Non-trainable params: 21,072
```

Figure 74 MobilnetV3-large-Unet parametres.

We also tried to decrease the number of skip connections to 4 reducing the number of parameters to 1.2 million, but while the training process we notice that the learning development was not good enough so we decided to ignore it.

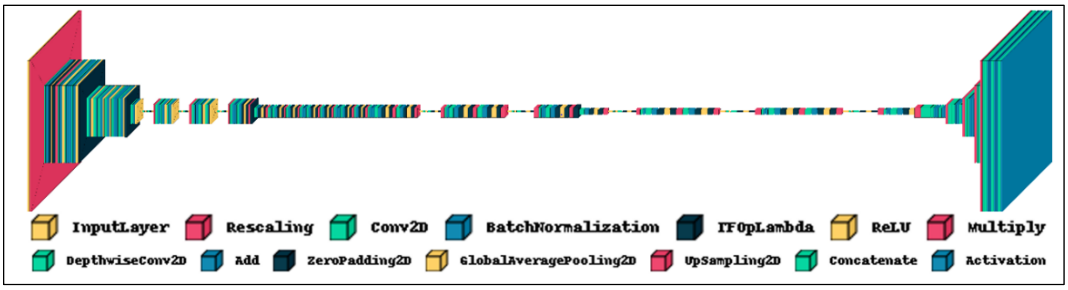


Figure 75 MobilnetV3-large-Unet architecture.

5.4.3 From mask to license plate

The segmentation model will give us a binary picture (black and white), this binary image contains white pixels in the license plate location (figure 76).

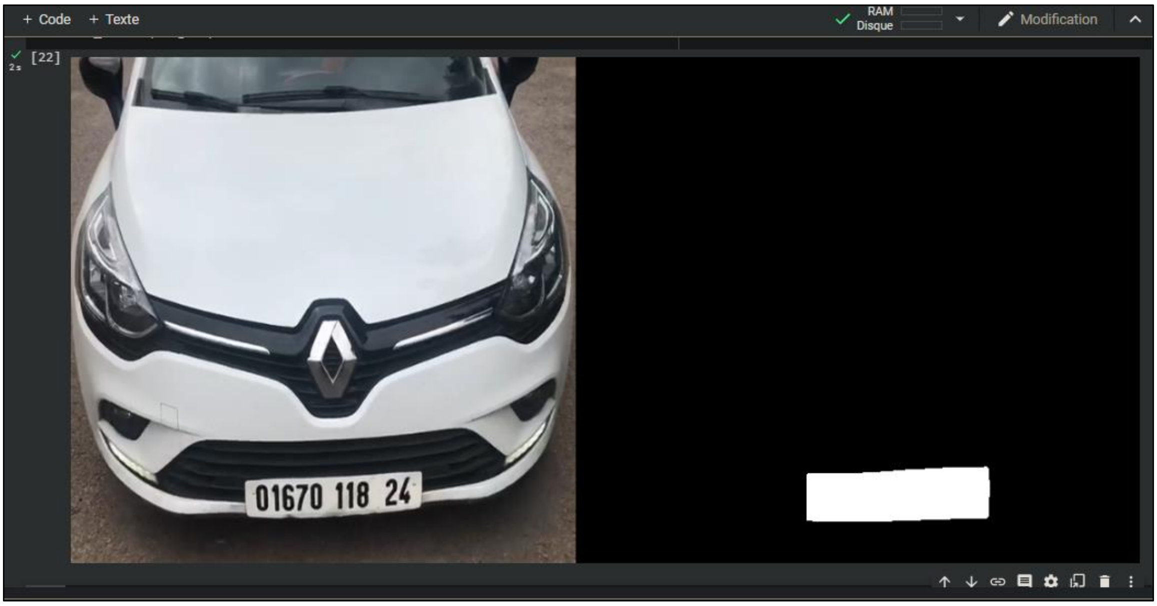


Figure 76 Car and its predicted license plate mask.

a) Predicted mask processing

However, in order to avoid noisy non-visible white pixels, we used two functions (Erosion[23] 3x3 and medianBlur 17x17).

- The first is the fundamental concept of erosion. If all of the pixels beneath the filter are 1, a pixel in the expected mask (either 1 or 0) will be considered 1, else it will be degraded (made to zero).

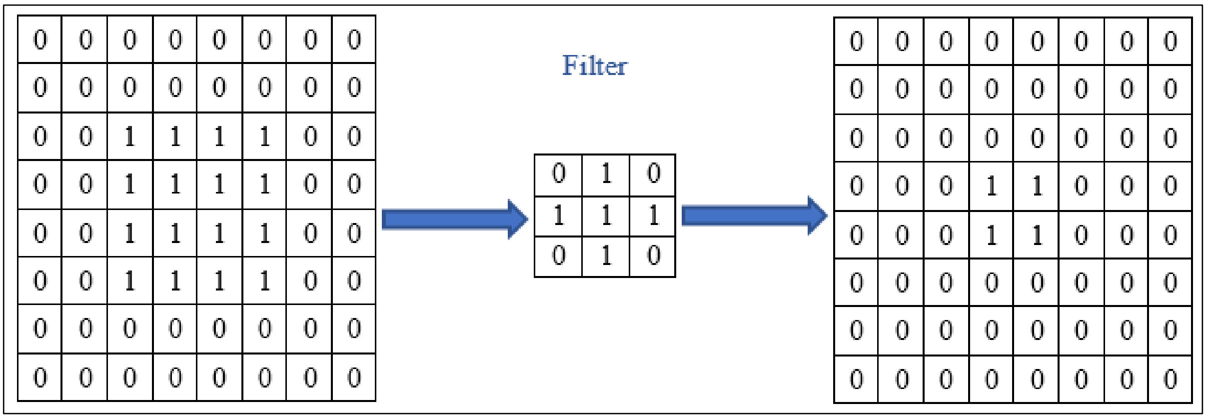


Figure 77 Erosion algorithm.

[23] https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

- Secondly, we used the medianblur to make the mask perfect by applying the filter 17x17 to remove the big noises (square of 17x17 pixels). It uses the values of the median pixels in the square to replace the center pixel value with black or white.

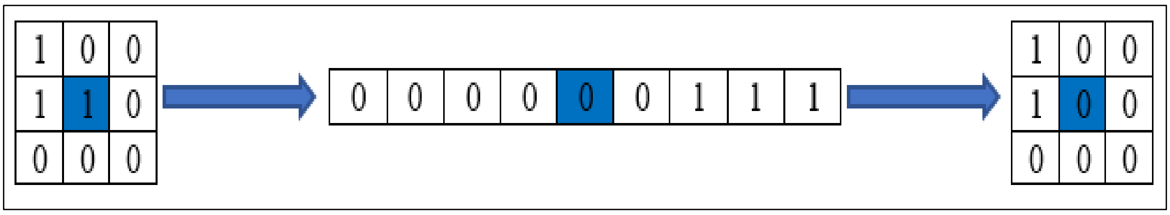


Figure 78 Sample of Median filter.

Figure 79 presents the detected mask after the Erosion and Medianblur.

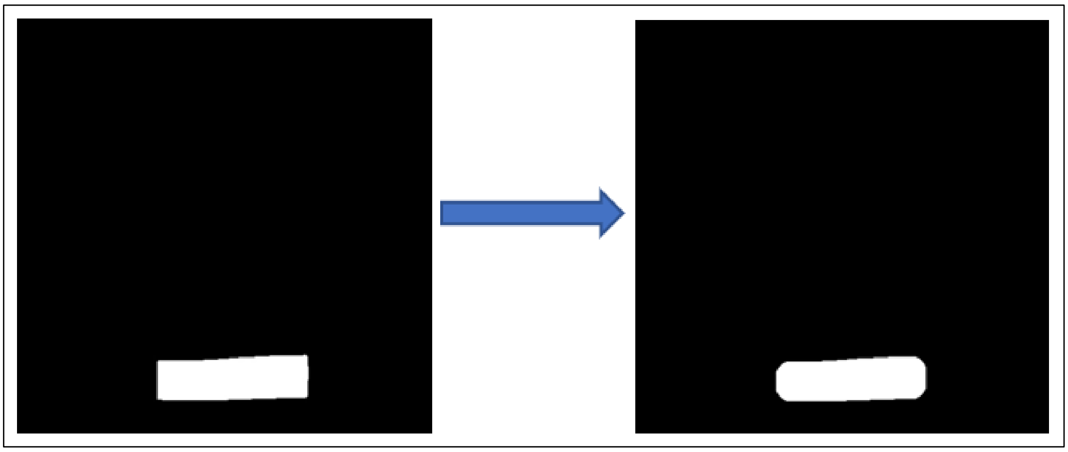


Figure 79 Predicted mask after erosion and the medianblur.

b) License plate extraction using predicted mask

The next step was to get the perfect mask and search for the largest contour area to obtain the license plate coordinate of the localization.

Using those coordinates we extracted the license plate from the resized frame (the same size as the predicted mask). and resize it to 1005x165 pixels.

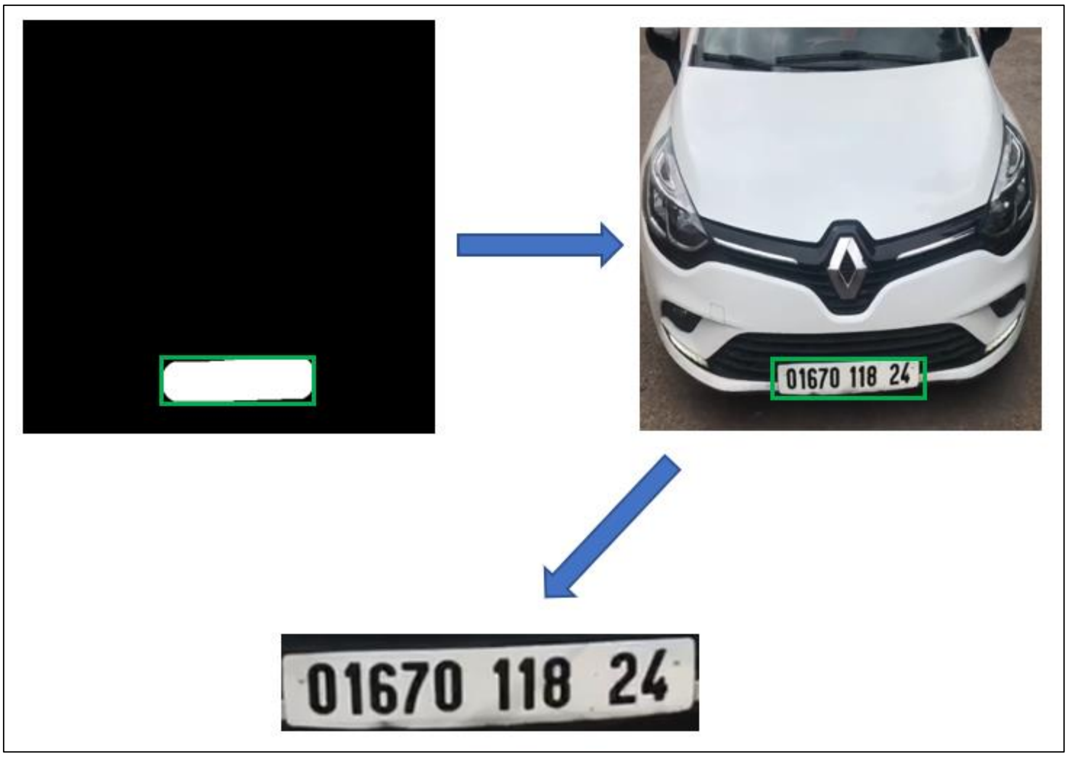


Figure 80 License plate extraction process.

5.4.4 From license plate to numbers

This step consists to use the extracted license plate to segment numbers using multiple steps. first, the colored license plate was converted to Gray Scale.

But the license plate was extracted from the initial image that is why we needed to smooth and remove noise from the license plate using the GaussianBlur algorithm. after that we thresholded the output from the last process (figure 81).

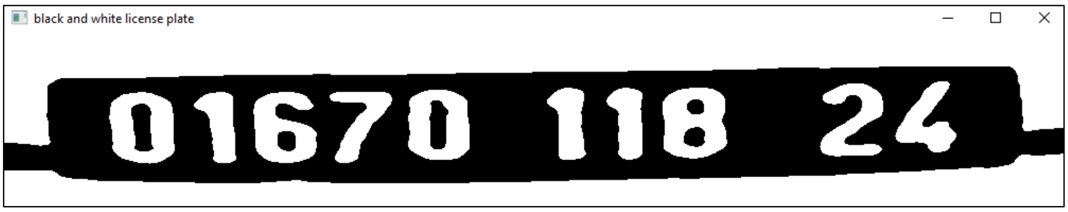


Figure 81 Smoothed and thresholded LP.

The second step is to get all contours using the same finding contours method and eliminate every contour that does not match with the Algerian license plate numbers form (mentioned in chapter 2). And extracted numbers from the black and white license plate.

Figure 82 License plate numbers.

5.4.5 Numbers recognition

This is the final step in the ALP Recognition. every single number got from the thresholded LP image resized to 64x64 pixels.

This is the preparation of the inputs for the number recognition model.

For this part, we trained the CNN model mentioned in chapter 1 with a 64x64 image shape.The Used CNN parameters shown in Figure 83.


```
=================================================================
Total params: 3,156,630
Trainable params: 3,156,630
Non-trainable params: 0
```

Figure 83 Our CNN parameters.

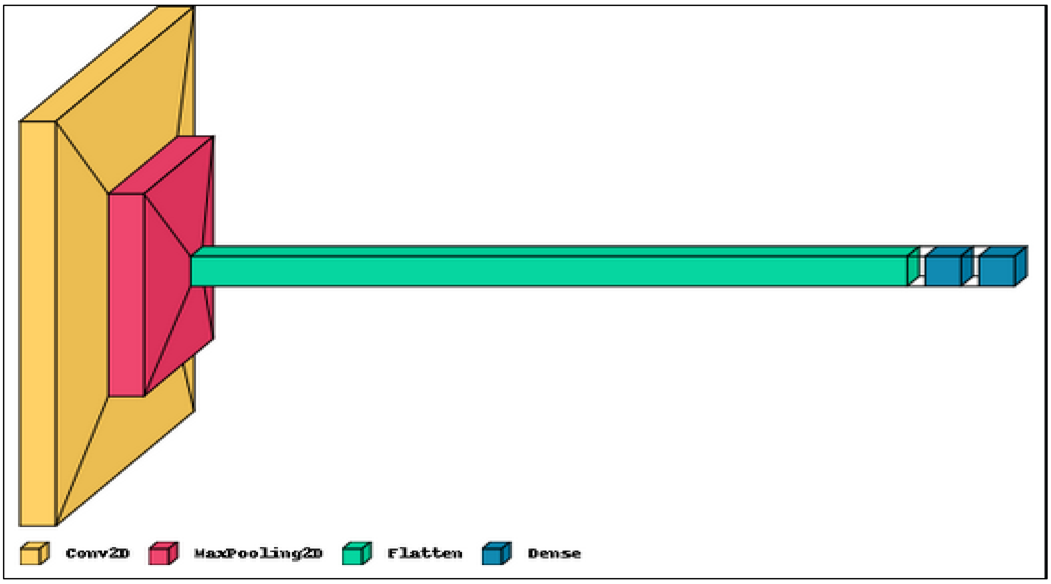


Figure 84 Our CNN architecture.

The output of the CNN model is a predicted number for the input image.

So, we just looped on number images to predict and concrete each number (Figure 85). this list of numbers will be compared with all authorized cars' license plates (employees' cars) for the garage opening.

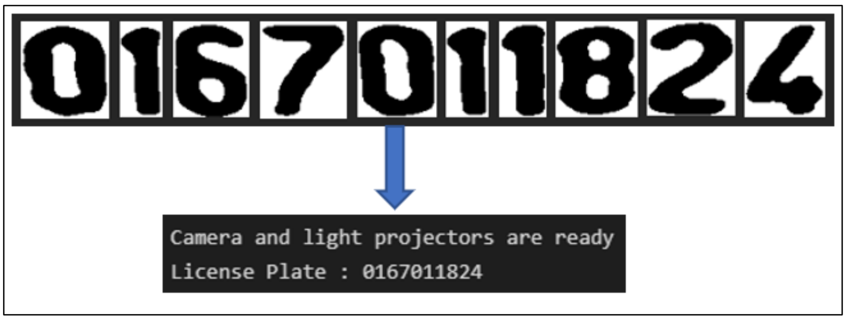


Figure 85 License plate numbers recognition.

5.5 Evaluation

This part concerned the evaluation of all models in the Automatic license plate recognition system.

5.5.1 Car detection

As mentioned in the implementation chapter we used two different methods for this task.

CNN model: was pre-trained on Google open image V6 car dataset with a number of epochs equal to 100, and using the early stopping function (10 iterations without development) to Avoid the overfitting. This last make it stop its training process on the 63 epochs, the results of this model are shown in Table 22.

Table 22 Car detection CNN test results.

	Precision	Recall	F1-score	Execution time
Train	97%	95%	95.9%	1 S/Frame
Test	94%	91%	92.5%	

Object detection algorithm: In this algorithm, we visually inspected the results because it is not a deep learning model. Table 23 shows the algorithm results.

Table 23 Car detection visually inspected.

The success ratio	Execution time
90%	0.03 S/Frame

- **Car detection model selection:** After observation of those two tables, we can see that the CNN model gives high-performance detection cars from the frame, unlike The Object detection algorithm that is more efficient based on the observation. But the main purpose-made us choose the best model is the real-time concept that needs a high-speed prediction what we can easily notice is that there is a huge difference between the two in favor of the object detection algorithm by 50 times.
  This big difference made us choose the last one instead of the CNN.

5.5.2 License plate extraction

We choose to adopt the semantic segmentation technique for the ALPR field for this assignment. for that, we trained two models, and this is their evaluation.

- **MobilnetV2-Unet:** We trained this model on the proposed dataset with a number of epochs equal to 80, and used the early stopping function (15 iterations without development) to avoid the overfitting, this last make it stop its training process on the 41. and we set the learning rate at 0.0001 and it will be reduced during the training by 10 times (0.0001 x 0.1) if there is no improvement for 4 epochs. Table 24 shows the model evaluation. The Figue 86 shows test result.

Table 24 V2-Unet evaluation

	Precision	Recall	F1 score	Loss	Execution time
Train	97%	94%	95.4%	4%	0.3S/ Fame
Validation	96%	92%	93.9%	6%	
Test	95%	92%	93.9%	6%	

```
loaded_model.evaluate(test_dataset, steps=test_steps)

43/43 [==============================] - 24s 540ms/step - loss: 0.0641 - f1score: 0.9394 - dice_coef: 0.9359 - recall: 0.9255 - precision: 0.9569
[0.06405746936798096,
 0.9394124150276184,
 0.9359426498413086,
 0.9254544377326965,
 0.9568992853164673]
```

Figure 86 V2-Unet model test evaluation.

- **MobilnetV3-Unet**: We also employed the early stopping function (15 iterations without development) to avoid overfitting, and we trained this model on the provided dataset with a number of epochs equal to 100, causing it to end training on the 41. And we set the learning rate at 0.00008, which will be reduced by ten times (0.00008 x 0.1) during the training if there is no progress after 4 epochs. The model evaluation is shown in Table 25 Figure 87 and 88 display the results of the train, validation, and test.

Table 25 V3-Unet evaluation.

	Precision	Recall	F1 score	Loss	Execution time
Train	99%	97%	98.3%	1%	0.6S/ Fame
Validation	88%	80%	83.9%	16%	
Test	95%	91%	92.9%	7%	

```
Epoch 41/100
360/360 [==============================] - 1692s 5s/step - loss: 0.0182 - f1score: 0.9831 - dice_coef: 0.9818 - recall: 0.9718 - precision: 0.9909

val_loss: 0.1611 - val_f1score: 0.8398 - val_dice_coef: 0.8389 - val_recall: 0.8078 - val_precision: 0.8856 - lr: 8.0000e-05
```

Figure 87 V3-Unet train and validation evaluation.

```
model.evaluate(test_dataset, steps=test_steps)

45/45 [==============================] - 19s 420ms/step - loss: 0.0735 - f1score: 0.9299 - dice_coef: 0.9284 - recall: 0.9167 - precision: 0.9562
[0.07346174120903015,
 0.9298785924911499,
 0.9284316301345825,
 0.9167206287384033,
 0.9561794996261597]
```

Figure 88 V3-Unet test evaluation

- **LP model selection:** Far from the fact that the project needs a high execution speed at this stage, high accuracy is necessary for the success of this process, which is the basis for the next stages, from these results, we can distinguish that those models are so close.

  Which made our choice so difficult but, we can observe that the execution time between both models is notable and this is due to the reduced number of parameters explained in the implementation chapter.

  So, for this aim, we choose the V2-Unet model that gives us the opportunity to run our system with higher accuracy and speed in real-time.

5.5.3 License plate numbers recognition

The CNN model was trained to solve the numbers recognition problem with 40 epochs, but it stopped on 25 epochs because of the early stopping we applied to avoid overfitting (7 iterations without development).

And with a 0.01 learning rate dividing it by 2 (0.01 x 0.5) during the training if there is no progress after 3 epochs and this is the final model in our ALPR system. and its evaluation is presented in Table 26.

Table 26 Number prediction CNN evaluation.

	Accuracy	Loss	Execution time
Train	99.9%	0.1%	0.02S/ Fame
Validation	99.8%	0.01%	
Test	99.9%	0.01%	

Figures 89 and 90 show the training and the validation accuracy and loss values involvement of the number recognition model during the training process in a graph

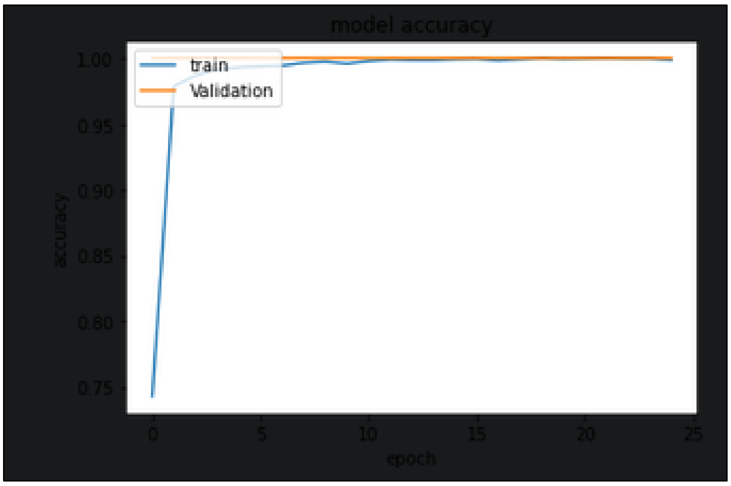


Figure 89 Number prediction CNN training and validation accuracy.

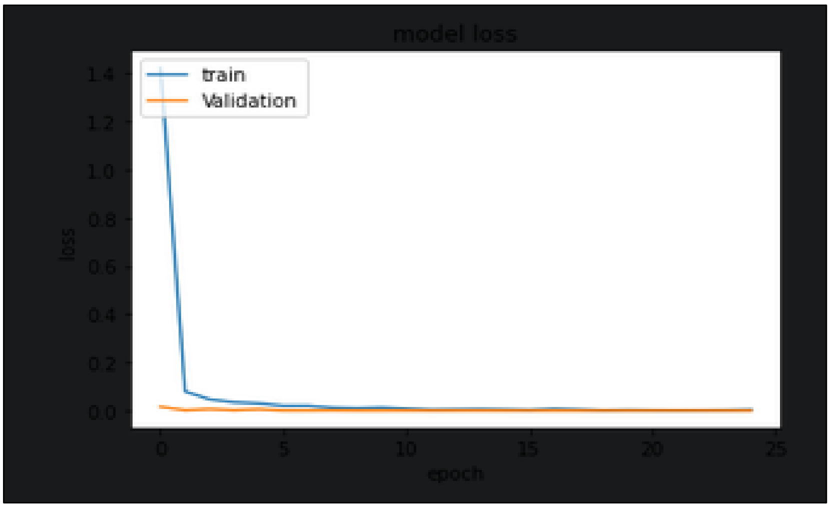


Figure 90 Number prediction CNN training and validation loss.

Figure 91 presents the number recognition model confusion matrix during the test evaluation

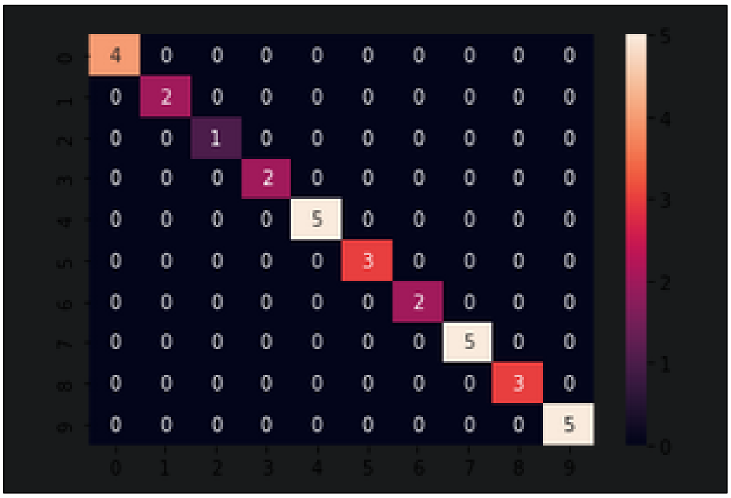


Figure 91 Confusion matrix of the number prediction model.

It is very clear in the confusion matrix that the Number prediction model in terms of prediction results or the execution time performs its intended purpose. the Algerian license plate contains between 10 and 11 numbers that made the model predict the LP numbers in 0.2s (0.02 x 10).

The whole Automatique license plate recognition system takes 0.63 seconds on a frame. by considering the numbers pre-processing for the Number Recognition model.

5.6 LP numbers recognition models comparison

This table compare between or LP numbers recognition model and the best one from the state of the art.

Table 27 LPR models comparison.

Models	Accuracy	Precision
Our model	99.9%	99.9%
Spanhel & Al Model	-	98%

Those two models evaluated on different dataset as shown in the prebious table our model gives a beter result on LP numbers recgnition compared to the state of the art models with result of 99.9% accurayc and precision.

5.7 Conclusion

We covered in this chapter the entire implementation process for the ALPR system, we started with data labels creation steps to get our final dataset.

Then, we discussed our model implementation parts with their architecture, training, and prediction method.

Finally, we finish by comparing them based on the mentioned criteria.

Chapter 6 Facial recognition Implementation & Evaluation and interface presentation

6.1 Introduction

This part describes the building phase of the facial recognition system. Made just in case of the LPR system failed to recognize the license plate or its output was not in our database (the car is not registered).

At this stage of work, we will explain face recognition step by step.

We used to implement this part of system the same implementation tools (hardware and software) mentioned in the previous chapter.

6.2 Dataset preparation

6.2.1 Anti-spoofing dataset

In order to prepare our anti-spoofing dataset, we split the dataset by train 70%, validation 10%, and test 20%.

After resizing it by 160x160x3 this Dataset was augmented using a brightness range between 0.8 and 1.2, rotation range of 30%, width shift range of 20%, height shift range of 20%, and a zoom range of 30%.

6.3 Face recognition system

This system will start following the license plate system if the aforementioned causes occur.

The face recognition system gets a frame from a second real-time video camera in front of the driver-side window.

6.3.1 Face detection

First, we start by getting that frame and converting it to grayscale. to obtain face localization details (start points, height, and width).

The algorithm examines a large number of smaller subregions and attempts to identify a face by looking for certain traits in each one. Because an image can have numerous faces of varied sizes, it was trained by checking many alternative placements and scales of faces. In this technique, Viola and Jones used Haar-like properties to detect faces. That is why We used the Viola-Jones algorithm to accomplish this task.

Haar features technique used in face detection and recognition. Some features of the human face are common among all mankind. We can distinguish that the eyes pixels of the human face are the darker and the nose pixels are usually the brighter [110].

So, we use the 3 haar types of extracting features (kernels). Edge features (Figure 94), Line features (Figure 93), and four-sided features (Figure 92).

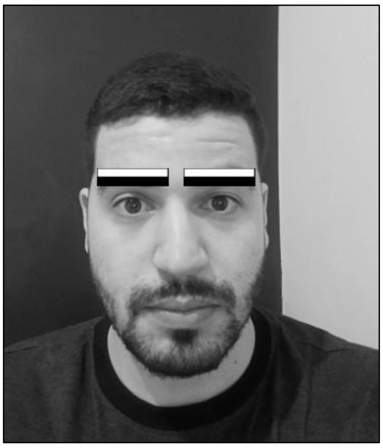


Figure 94 Haar Edge features.

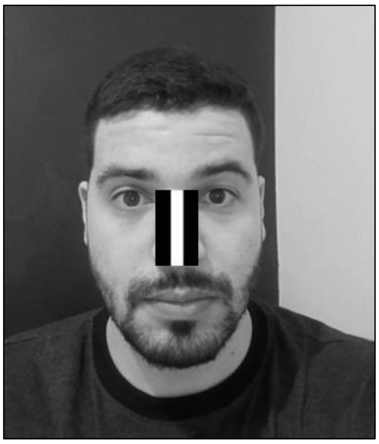


Figure 93 Haar Line features

Figure 92 Haar four-sided features

These features were treated with many processes using haar cascade frontal face and the output will be the face localization details, after that we extracted the face from the colored image (Figure 95).

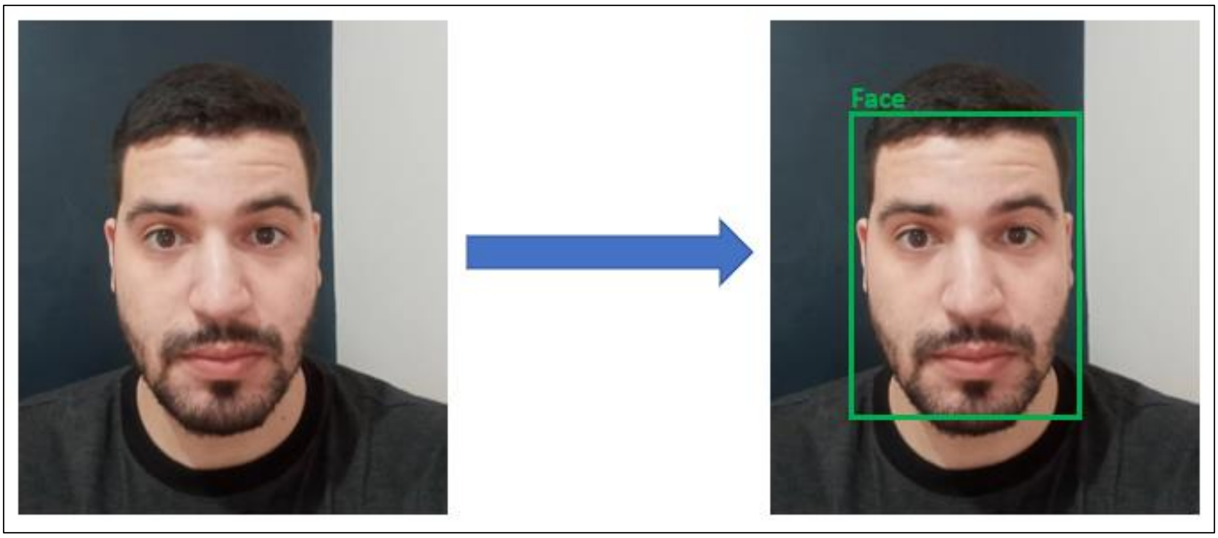


Figure 95 Single Face cascade classifier results.

This model works also with multi-face frames (more than one person in the image) see Figure 96, also, non-complete faces (covered) and indirect faces (faces in pictures) see Figure (97,98).

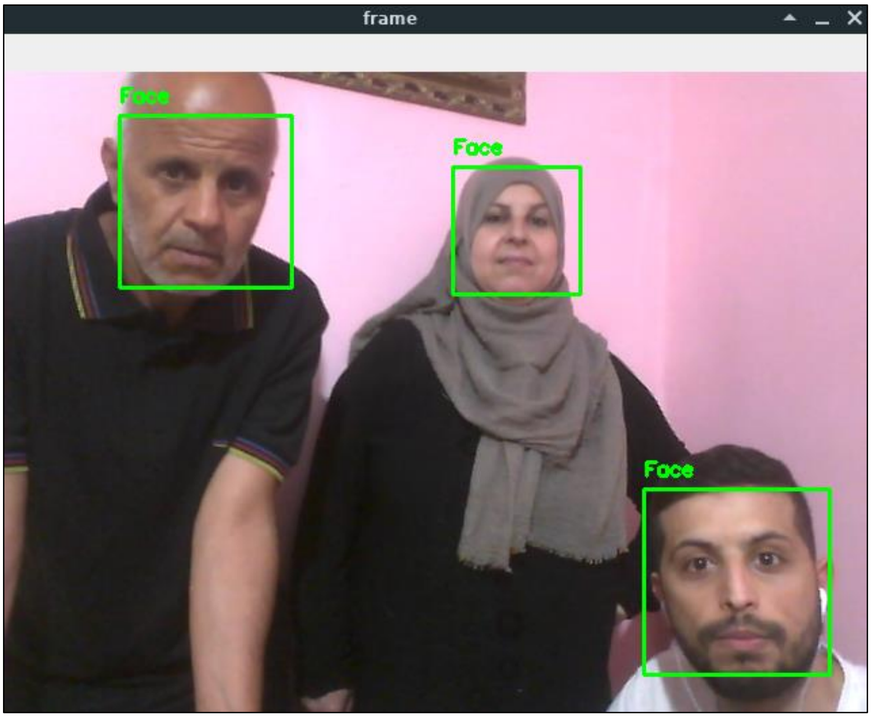


Figure 96 Multi Faces detection.

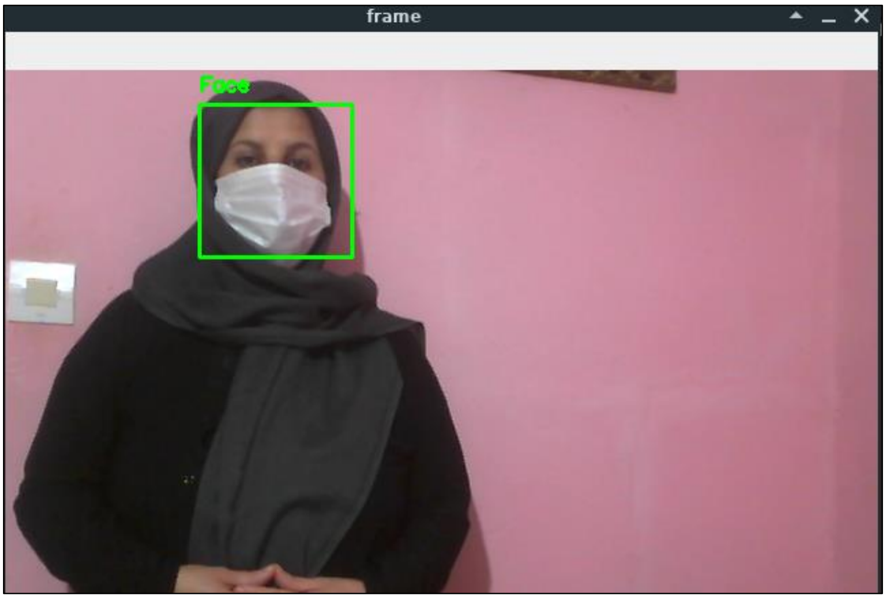


Figure 97 Covered Face detection.

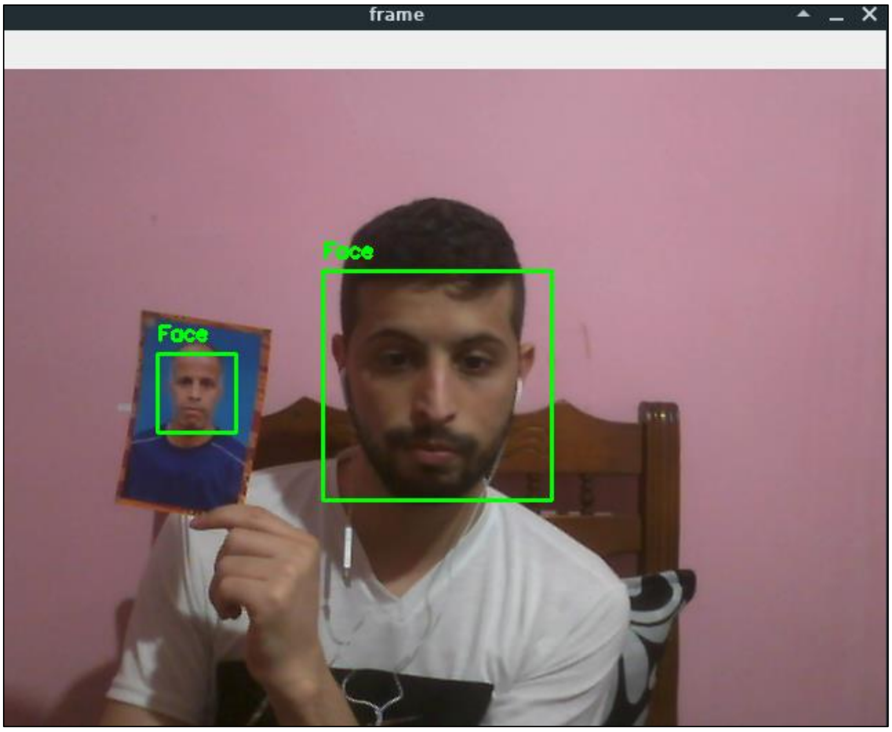


Figure 98 Indirect face detection.

6.3.2 Anti-spoofing model

Spoofing is the most common problem in face recognition systems. this system was made to solve this problem by taking the resized output of the face detection model (only face area) to 160x160x3.

For this step, we trained the MobilnetV2 model (explained earlier) with some freezing layers and the total number of parameters in Figure 99.

```
...
Total params: 2,514,001
Trainable params: 256,017
Non-trainable params: 2,257,984
```

Figure 99 MobilnetV2 Model Parametres.

This mobilnetV2 model architecture and ordered layers are presented in Figure 100.

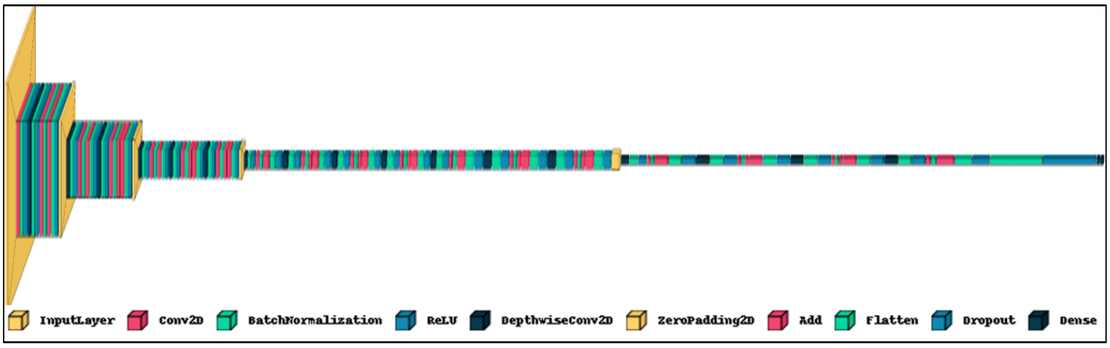


Figure 100 MobilnetV2 model parametre.

This model is used to extract the face image features to classify the input image if it is a real face or a spoof (Figure 101).

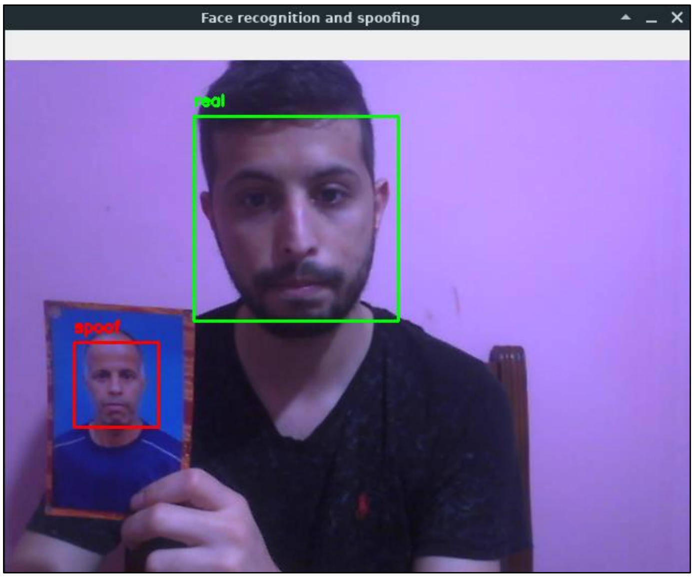


Figure 102 Spoof detection from a photo.

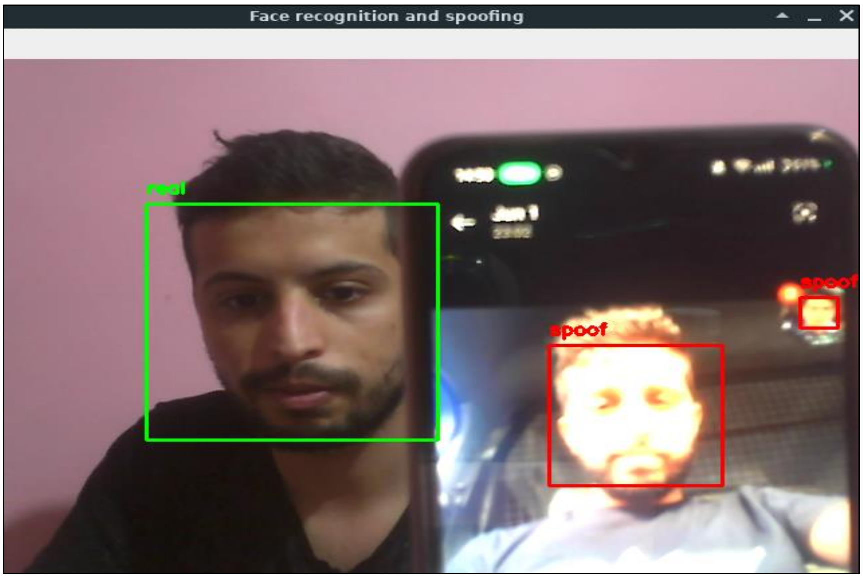


Figure 101 Spoof detection from a smartphone.

We decided not only to prevent spoofers from entering our private areas. So, we linked our application with an email service to automatically send an email to the security containing the frame of the camera in front of the face and also a frame of the camera in front of the LP (Figure 103) presenting the received email with these attachments.

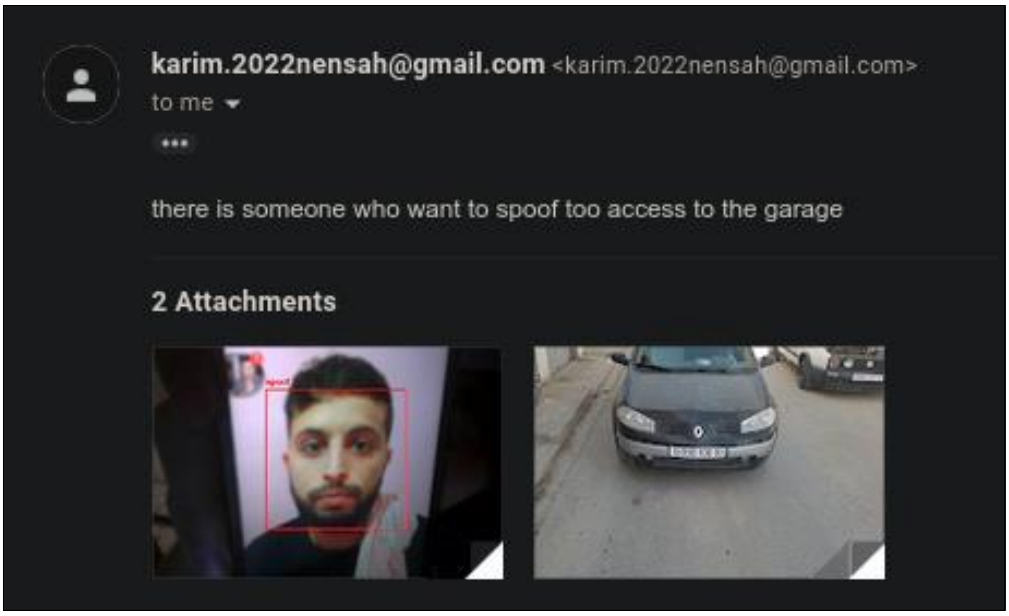


Figure 103 Email received about spoof traying using a phone.

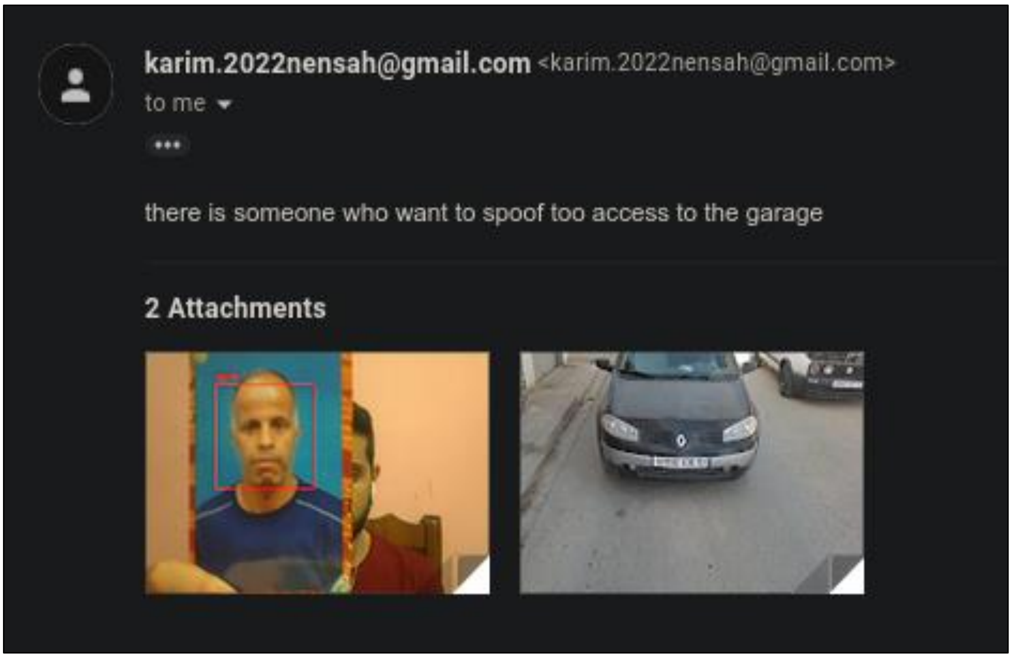


Figure 104 Sample of received email about spoof traying.

6.3.3 Face recognition model

Now we are sure that the face is detected and it is a real face we noticed that the detected area of the face covered also little parts that do not belong to the face, so we cropped that frame by minimizing 20 pixels from each side (Figure 105).

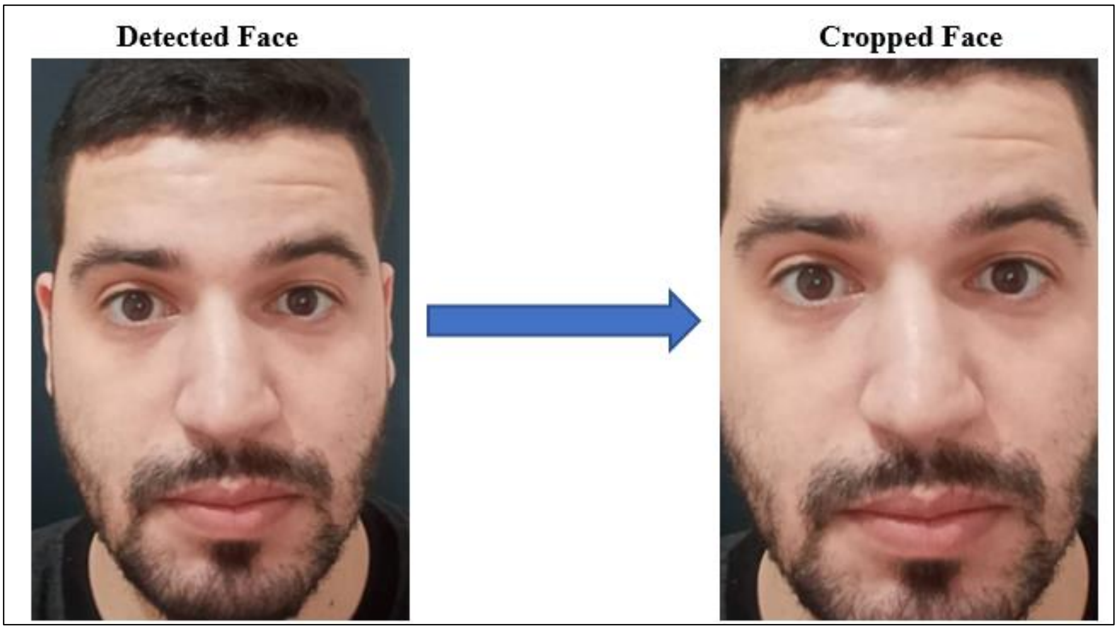


Figure 105 Cropped image.

Then we need to extract the face features, the best way to present those features as an array form of 512 lengths (Figure 106) from the resized image.

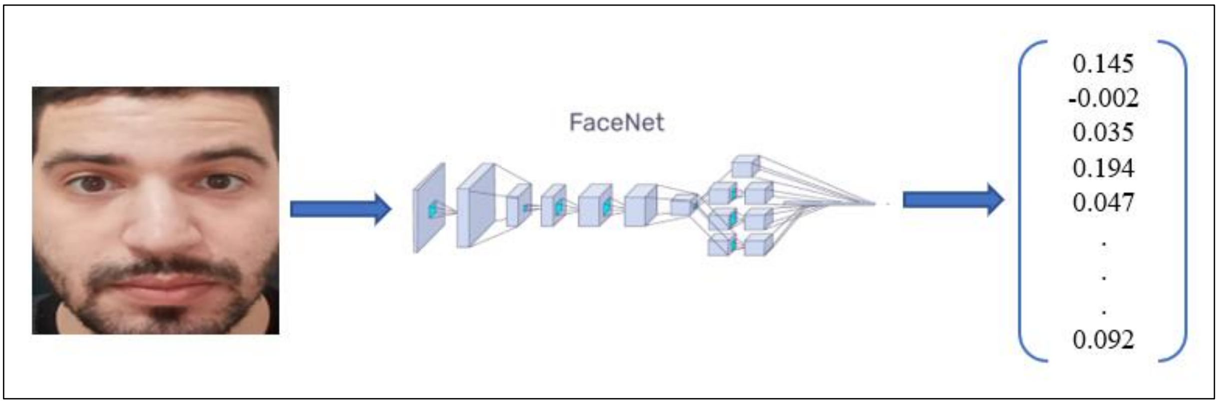


Figure 106 Embedding extraction from the face.

The model that makes this transformation is named Facetnet (parameters and layers shown in Table 28 and Figure 107) this model is a neural network architecture based on the triplet loss function principle to create the best embeddings.

Table 28 Facenet architecture [111].

layer	size-in	size-out	kernel	param
conv1	220×220×3	110×110×64	7×7×3, 2	9K
pool1	110×110×64	55×55×64	3×3×64, 2	0
rnorm1	55×55×64	55×55×64		0
conv2a	55×55×64	55×55×64	1×1×64, 1	4K
conv2	55×55×64	55×55×192	3×3×64, 1	111K
rnorm2	55×55×192	55×55×192		0
pool2	55×55×192	28×28×192	3×3×192, 2	0
conv3a	28×28×192	28×28×192	1×1×192, 1	37K
conv3	28×28×192	28×28×384	3×3×192, 1	664K
pool3	28×28×384	14×14×384	3×3×384, 2	0
conv4a	14×14×384	14×14×384	1×1×384, 1	148K
conv4	14×14×384	14×14×256	3×3×384, 1	885K
conv5a	14×14×256	14×14×256	1×1×256, 1	66K
conv5	14×14×256	14×14×256	3×3×256, 1	590K
conv6a	14×14×256	14×14×256	1×1×256, 1	66K
conv6	14×14×256	14×14×256	3×3×256, 1	590K
pool4	14×14×256	7×7×256	3×3×256, 2	0
concat	7×7×256	7×7×256		0
fc1	7×7×256	1×32×128	maxout p=2	103M
fc2	1×32×128	1×32×128	maxout p=2	34M
fc7128	1×32×128	1×1×128		524K
L2	1×1×128	1×1×128		0
Total				140M

```
==================================================================
Total params: 23,497,424
Trainable params: 23,467,824
Non-trainable params: 29,600
__________________________________________________________________
```

Figure 107 Our Facenet parameters.

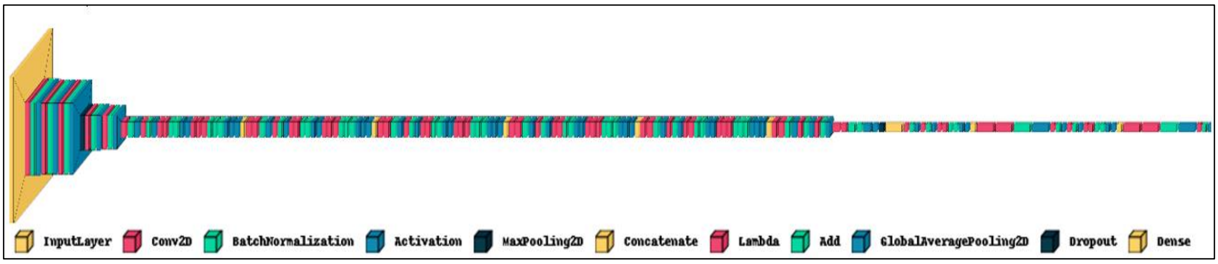


Figure 108 Our Facenet architecture.

- Triplet loss is a function that requires three pictures: an anchor, a positive image, and a negative image. The theory underlying the function is that the anchor should be closer to the positive images than the negatives. To obtain the result, we simply compute the distance between the anchor and positives embeddings, as well as the distance between the anchor and negatives embeddings, and then compare the two distances [112].

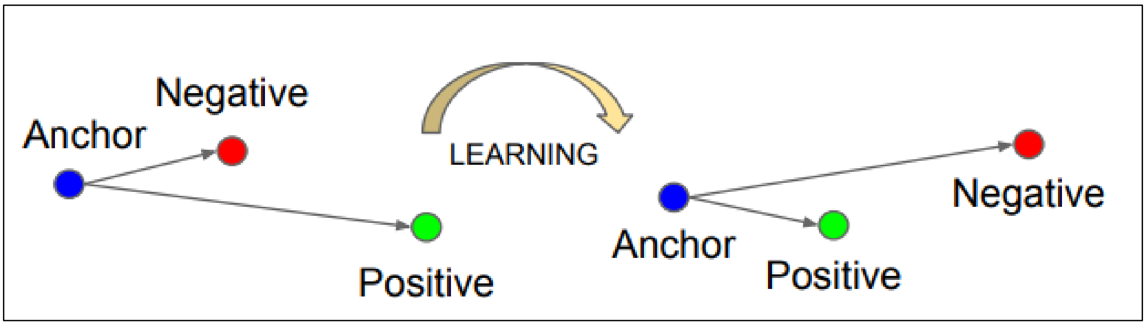


Figure 109 Triplet loss function [112].

After the embedding has been generated, similarity algorithms such as Cosine or Euclidean for recognition can be used to verify and recognize the face by calculating the distance between the output embedding with database embedding.

- The cosine similarity [113] function is a vectorial method that calculates the similarity of two arrays and the outputs of this function are between 0 and 1 by calculating the angle between those two arrays using this formula:

$$\text{Similarity (A, B)} = \frac{A.B}{||A|| \times ||B||} \text{ [112]}$$

- The euclidean distance is another vectorial method we used to compare face embeddings (the database and the predicted) using the formula:

$$\text{Distance} = \sqrt{\sum_{i=0}^{N}(x_i - y_i)^2} \text{ [112]}$$

After all this process, our system has the ability to recognize faces to detect authorized and unauthorized persons.

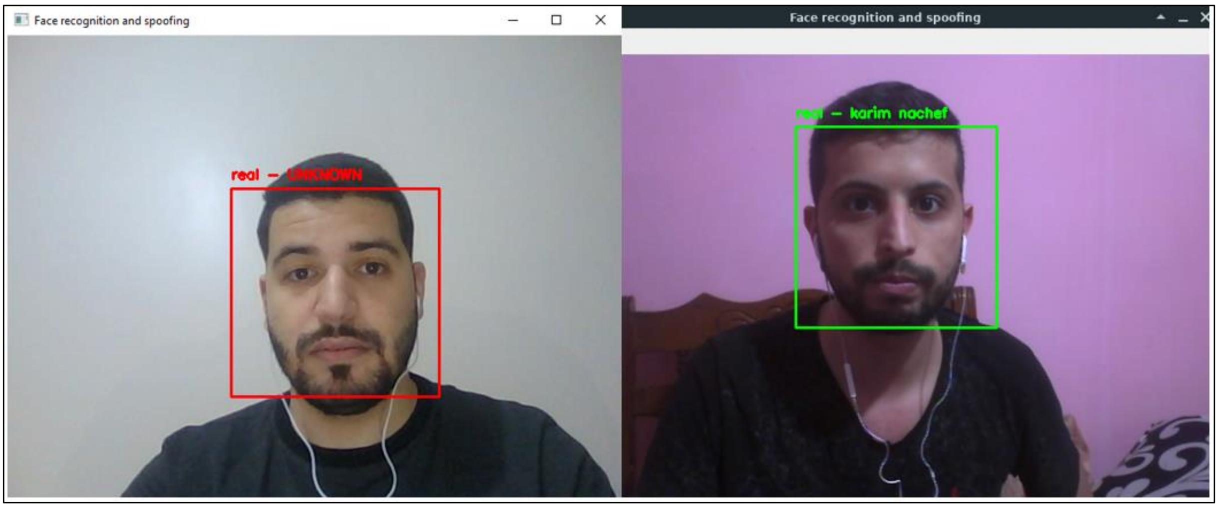


Figure 110 The final output of face recognition.

6.4 Face recognition evaluation

This part concerned the evaluation of all models in the Face recognition system.

6.4.1 Face detection model

The haar cascade is a pre-trained algorithm that uses edge detection features proposed by Viola and Jones in their research paper. Table 29 presents the results given by this algorithm using different test images (multi faces).

Table 29 Haar cascade results [3].

No.of faces in an image	Execution Time (sec)	No.of faces detected	Accuracy (%)
5	0.141	5	100
10	0.55	9	90
15	0.11	12	80
20	0.369	19	95

This model demonstrated its ability to detect the face with great accuracy, especially in cases where the image contains several faces, and this ability is accompanied by its high speed that complies with the requirements of our project (real-time).

6.4.2 Anti-spoofing model

We trained the anti-spoofing model on 100 epochs made it learn well and give a good result. we used a learning rate equal to 0.000001. Table 30 shows the model evaluation.

Table 30 Andi spoofing model evaluation.

	Loss	Accuracy	Execution time
Train	0.11%	95.8%	0.1S/Frame
Validation	0.06%	97.3%	
Test	-	98.7%	

Figures 111 and 112 show the training and the validation accuracy and loss values involvement of the anti-spoofing model during the training process in a graph.

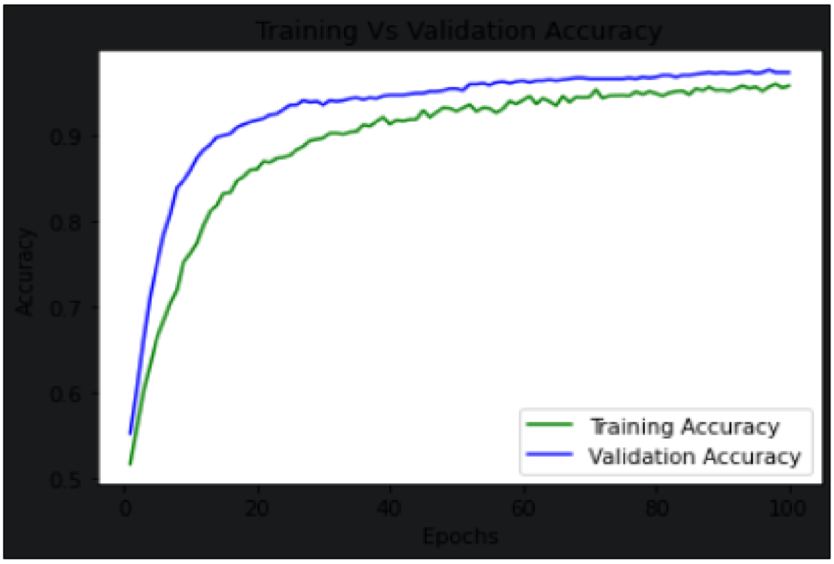


Figure 111 Anti-spoofing model training and validation accuracy.

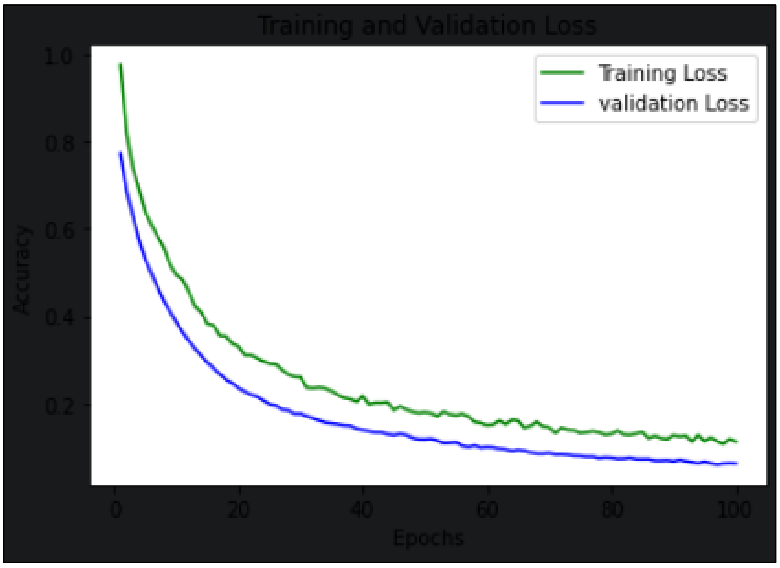


Figure 113 Anti-spoofing model training and validation loss.

Figure 113 presents the anti-spoofing model confusion matrix during the test evaluation.

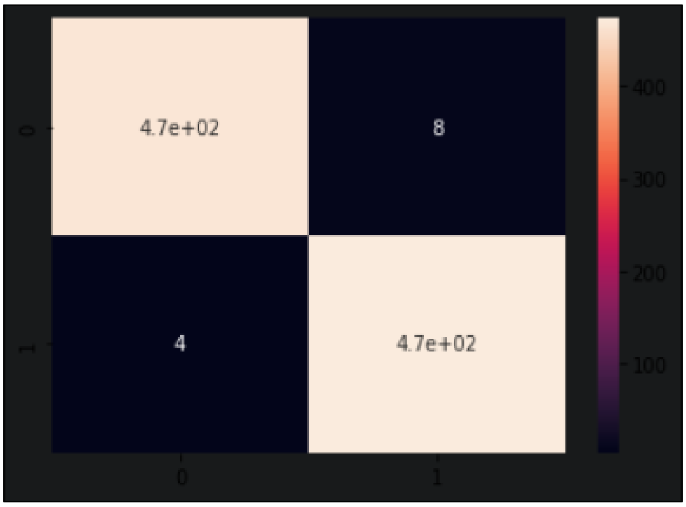


Figure 112 Confusion matrix during the test evaluation.

According to the results, we can be satisfied with this model, as we note that it has high accuracy, and this is necessary for the sensitivity of the task it performs, and all this is in a record time estimated at 0.1 seconds.

6.4.3 Face recognition model

Facenet is a pre-trained model developed by Florian Schroff, Dmitry Kalenichenko, and James Philbin in Google company. trained on the LFW dataset. its results are shown in the following table.

Those 2 are techniques for LFW pre-processing:

1. Fixed center crop of the image provided in LFW.

2. A face detector is used on LFW images if that fails then LFW face alignment is used.

Table 31 Facenet evaluation [76].

	Loss	Accuracy	Execution Time
1	0.15%	98.87%	0.1S/Frame
2	0.09%	99.63%	

This famous model in the face recognition field guarantees us get better results and minimize errors that's why we decided to use it to finalize our face recognition system.

6.5 User interface

In our system, we talked about some employees' data like license plate numbers and faces, but we did not explain how we control this data.

For that, we create a simple user interface to facilitate dealing with this data like inserting, removing, editing, or even collecting it for the future, in the next section we will present our interface.

Every app should be secure Especially in our project, so we decided to add a login interface (Figure 114) to avoid unwanted people to access data.

The login interface contains 2 text fields for the User name and the password, and we added the developers' emails in case of password forgetting.

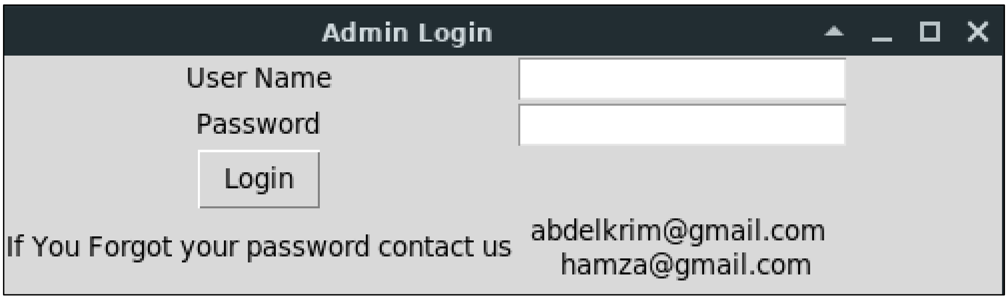


Figure 114 Login interface.

After we set the user name and the password correctly we faced the main user interface. It is divided into 2 big sections. One for displaying data information and the other for the management.

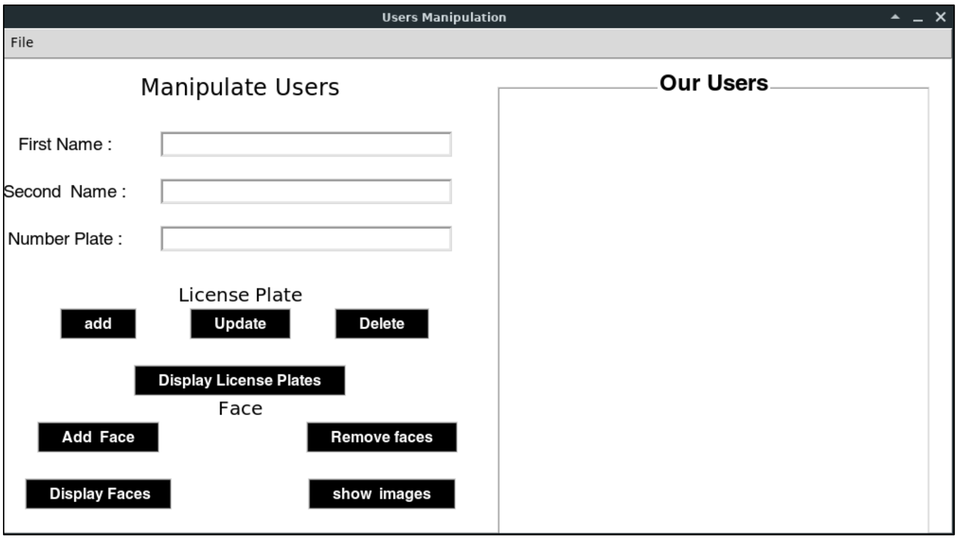


Figure 115 Main user interface.

At the top of the management section, we found some text fields that allow us to introduce our information to make one of those operations. Figure 116 numbering is explained the possible management in the license plate dataset.

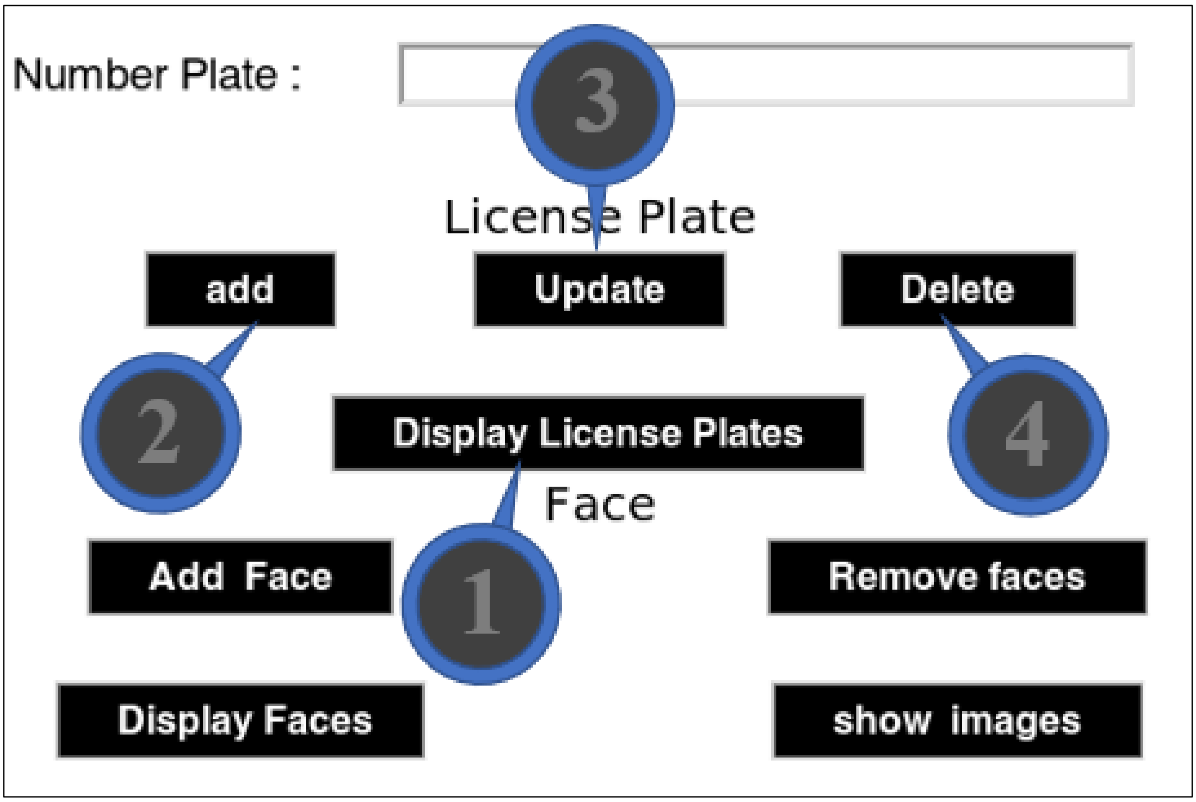


Figure 116 License plate management subsection.

1) The display license plate button is used for monitoring authorized cars in the database. and to check modifications.

2) Add license plate number button facilitate submitting. it is used to add new authorized cars to the database.

3) While the insertion processes the user could make some typing errors with this button, we can solve this problem.

4) The delete button gives us the ability to remove authorized cars from the dataset.

Figure 117 numbering is explained the possible management in the face Dataset.

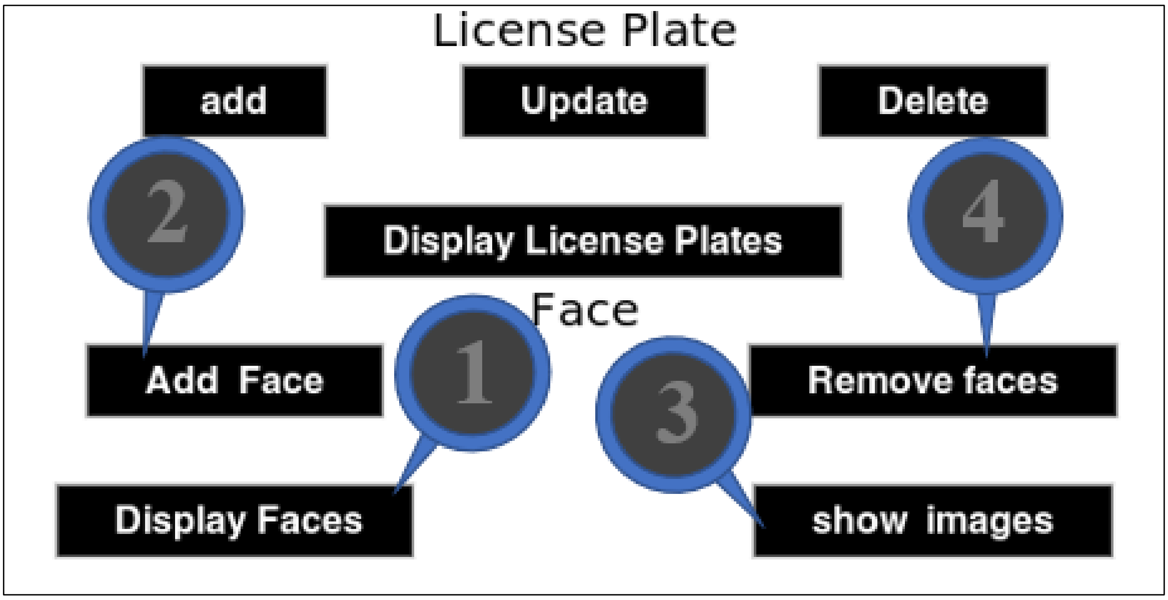


Figure 117 Face management subsection.

1) The display faces button is used for monitoring the names of authorized persons in the database. and to check modifications.

2) Add face button facilitate submitting. it is used to add new authorized persons to the database by saving the face embedding (face features) by taking 5 face images for each person to confirm a good facial recognition.

3) The show images button displays face images of authorized persons. and gives us the possibility to get the faces to improve our face recognition model by retraining it using those faces.

4) The remove faces button gives us the ability to remove an authorized person from the dataset.

Figure 118 is showing the authorized cars' license plates after adding a new car.

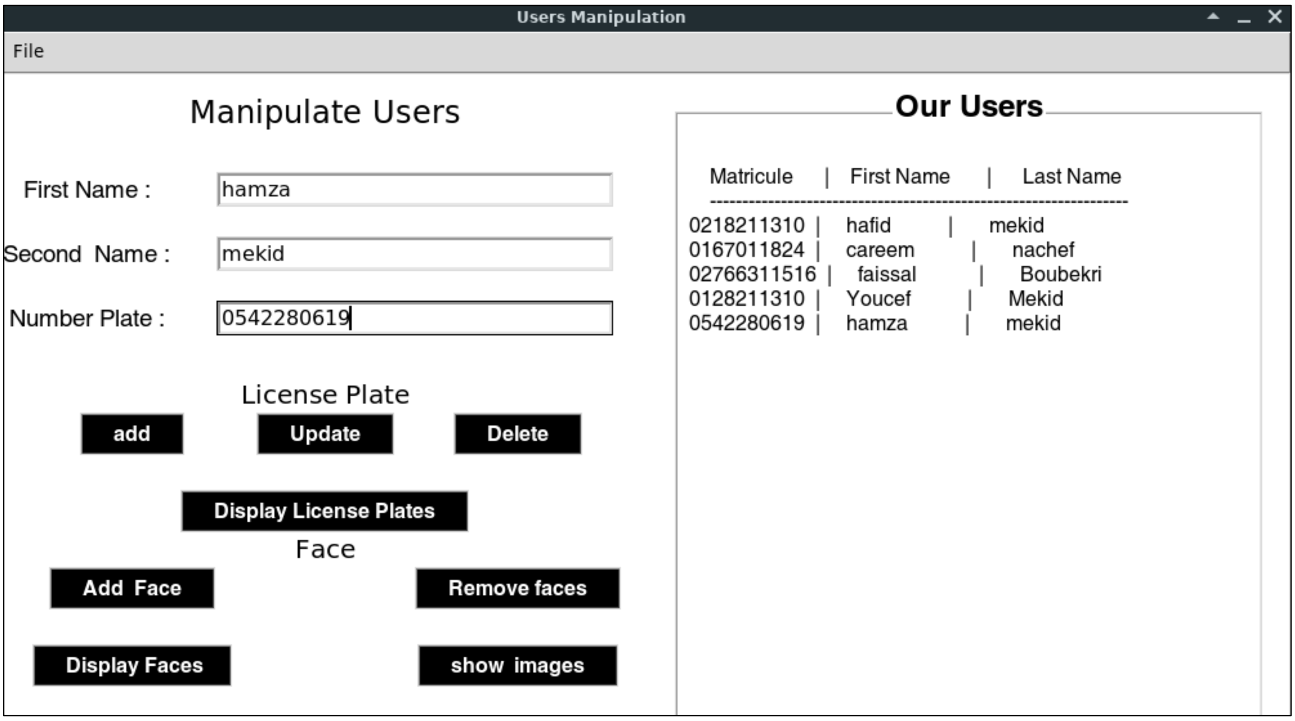


Figure 118 Display authorized cars.

Figure 119 explains how to add a new authorized face to the database.

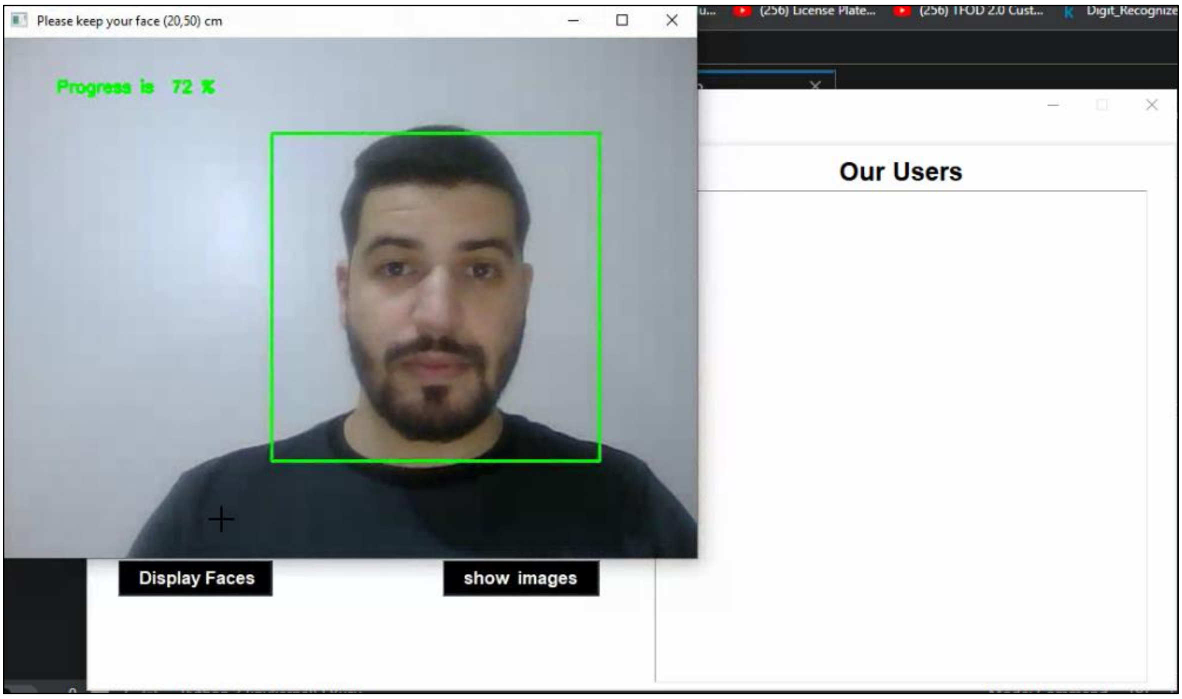


Figure 119 Adding face to database.

6.6 Conclusion

In this chapter, we presented the complete face recognition system that we built step by step with all details. starting with the face detection passing by the anti-spoofing system that improves the level of security in our system.

Then, we explained our face recognition model that gives the final decision about garage opening. After that, we made an evaluation of those models to confirm that they serve the purpose of our problematics.

Finally, we conclude this chapter by presenting our user interface and its functionalities.

General conclusion and perspectives

In order to achieve our desired goal, we divided our project into two main parts. In the first part, we dealt with the problem of recognizing license plate numbers. We used new specialized methods in the biomedical field and adapted them to our problem, which is usually solved by the methods we mentioned in ALPR related works. We used this method to extract the license plate, using our neural network architecture consisting of Unet and MobilnetV2, which gave us an excellent result estimated at 95 percent. After that, we extracted all the numbers from the license plate to be an input for the CNN model, which had previously trained to identify the numbers, where we obtained a wonderful result estimated at 99.9%. The whole license plate recognition part can predict 30 frames in 19 seconds.

The second part revolves around knowing the identity of the driver of the car using facial recognition technology, where we first extracted the face using a technique called haar cascade, and then we applied the anti-spoofing model, which allowed us to protect the companies parking from people trying to enter by spoofing an authorized person, this model gave us an excellent result estimated at 98.7 percent. If it is confirmed that there is an attempt to spoof the person, the program will send an automatic email containing the person's picture and the picture of the car he is riding to the person in charge of security in the company. After making sure that this is not a spoofing, we move on to the last stage, which is to sweat on the driver's face and determine his identity if he is allowed to enter, where we extract facial features and save them as embeddings using the Facenet model to calculate the similarity using cosine similarity function. The whole face recognition system can predict 30 frames in 9 seconds.

We can say at this stage that the objectives set at the beginning of this study have been achieved and we could have added some of our ideas like the anti-spoofing model, and that deep learning is an appropriate technique that allows the exploitation of real-time videos to control access to private parking, the deep learning allows us to achieve a high precision level with high-speed processing in all our access control system.

In the future, we can develop this system more to obtain more accurate results by applying self-learning technology using collected data from the predicted images.

Bibliography

[1] S.Sandhu & P.Samarati, Access control: principle and practice. In: IEEE communications magazine 32.9 (1994), pp. 40–48.

[2] C.Wu,S.G.Song,& S.Zhang, How many labelled license plates are needed? In Chinese Conference on Pattern Recognition and Computer Vision (PRCV) , pages 334-346.Springer ,2018.

[3] E.Winarno, W.Hadikurniawati, A.Nirwanto, D.Abdullah, Multi-View Faces Detection Using Viola-Jones Method, Conference Series, 2018.

[4] Greetly Digital Receptionist Jan 23, G. D. Receptionist, and TGDaily, Physical Security Access Control Systems: Greetly, Visitor Management System, Log Book, Badges & Support, 09-Jul-2019. [Online]. Available: https://www.greetly.com/blog/physical-security-access-control-systems. [Accessed: 01-Jun-2022].

[5] J. Martinez, 5 ways to Keep Your Network Safe, PCMAG, 14-Jul-2017. [Online]. Available: https://www.pcmag.com/news/5-ways-to-keep-your-network-safe. [Accessed: 3-Apr-2022].

[6] Benefits of implementing an access control system: Morefield Morefield Communications, 12-Feb-2020. [Online]. Available: https://www.morefield.com/blog/benefits-of-an-access-control-system/#:~:text=An%20access%20control%20system%20gives,the%20time%20of%20the%20incident. [Accessed: 19-May-2022].

[7] Madras Library Association - Kalpakkam Chapter and Indira Gandhi Centre for Atomic Research, Kalpakkam. Digital Libraries to Knowledge Systems. Conference Theme Digital Libraries to Knowledge Systems. 2005

[8] S.Li & Y.Chen. License Plate Recognition. University of Gävle , Sweden .2011.

[9] Trends and developments in vehicle access control, Nedap, 11-Sep-2018. [Online]. Available: https://www.nedapidentification.com/insights/trends-and-developments-in-vehicle-access-control/. [Accessed: 18-Apr-2022].

[10] D.da Silva. Computer Vision System for Tactode Programming, university of Porto. July 30, 2020

[11] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and

prospects. Pattern Recognition, 34(12):2259 – 2281, 2001. doi:https://doi.org/10.

1016/S0031-3203(00)00149-7

[12] P.Hough. A method and means for recognition complex patterns; US Patent: US3069654A, 1962. Available at https://patents.google.com/patent/ US3069654A/en. Last accessed 2020-06-10

[13] O.D. Trier, A.K. Jain, and T Taxt. Feature extraction methods for character recognition - A survey. Pattern Recognition, 29(4):641–662, 1996.

[14] S.Jain & V.Laxmi. Color image segmentation techniques: A survey. In Proceedings of the International Conference on Microelectronics, Computing & Communication Systems, pages 189–197, 2018.

[15] A.Anjos & H.Shahbazkia. Bi-level image thresholding - a fast method. BIOSIGNALS, 2:70–76, 01 2008.

[16] N. Otsu. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1):62–66, Jan 1979. doi:10.1109/TSMC.1979. 4310076

[17] A. Khan. Image segmentation methods : A comparative study. International Journal of Soft Computing and Engineering (IJSCE), 3:84–92, 2013

[18] S.Beucher & C.Lantuéjoul. Use of watersheds in contour detection. In International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, 132, 1979.

[19] B.Aimen ,N. Akram. Classification des images satellitaires pour l'aide à la gestion des catastrophes naturelles en utilisant l'apprentissage profond. Saad Dahlab University - Blida 1. 2020

[20] A. Moawad,Medium. 2022. *Neural networks and backpropagation explained in a simple way*. [online] Available at: <https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a- simple-way-f540a3611f5e> [Accessed 13 April 2022].

[21] F. Simon, Deep Learning, les fonctions d'activation Phd thesis,U C Etienne 2018.

[22] P. Jain, Complete Guide of Activation Functions edition MC.IA, 2019

[23] D. Liu, A Practical Guide to ReLU. web article, Medium 2019.

[24] S. Raschka, What is Softmax regression and how is it related to logistic regression?, Dr. Sebastian Raschka, 17-Jun-2022. [Online]. Available: https://sebastianraschka.com/faq/docs/softmax_regression.html. [Accessed: 22-May-2022].

[25] Softmax activation function, InsideAIML. [Online]. Available: https://insideaiml.com/blog/SoftMaxActivation-Function-1034. [Accessed: 05-Mar-2022].

[26] R. Gómez, Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names, neptune.ai, 2018

[27] A. Indolia, Conceptual understanding of convolutional neural network-A deep learning approach, edition Procedia Computer Science , 2018.

[28] Y. Abdel-hamid, Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition, p. 3366–3370, 08 2013.

[29] A. Mustafa, Image processing of an agricultural product: Determination of size and ripeness of a banana, International Symposium on Information Technology, 2008 .

[30] G. Zakaria, Classification des phases de la marche humaine par apprentissage profond - Application au domaine medical Thesis Master, Alger, 2019.

[31] S. Olivas, Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods and Techniques, IGI Global, 2009.

[32] N. Srivastara, Dropout :a simple way to prevent neural networks from overfitting

[33] H. Kaiming, X. Zhang, R. Shaoqing et S. Jian, Deep Residual Learning for Image Recognition, Microsoft Research,CPVR, Computer vision foundation, 2015

[34] A. Dertat, Applied Deep Learning - Part 4 : Convolutional Neural Networks, Towards data science, 2017

[35] C. Yu, M. Xi and J. Qi A Novel System Design of License Plate Recognition Computational Intelligence and Design, 2008. ISCID '08. International Symposium on, Page: 114, Issue Date: 17-18 Oct. 2008

[36] P. Wang Research of the license plate automatic recognition (in Chinese), Master thesis, Shanghai Southeast University: China. 2006

[37] Cockpit, Algerian license plate standards, COCKPIT, 23-Nov-2021. [Online]. Available: https://en.cockpitdz.com/post/algerian-license-plate-standards. [Accessed: 17-MAY-2022].

[38] H. Li, P. Wang, and C. Shen, Toward end-to-end car license plate detection and recognition with deep neural networks, IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 1126–1136, Mar. 2019.

[39] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, Towards end-to-end license plate detection and recognition: A large ,dataset and baseline, in Proc. Eur. Conf. Comput. Vis., in Lecture Notes in Computer Science, vol. 11217, 2018, pp. 261–277

[40] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[41] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[42] R. Girshick, Fast R-CNN, in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[43] J. Shashirangana, H. Padmasiri, D. Meedeniya and C. Perera, Automated License Plate Recognition: A Survey on Methods and Techniques, in IEEE Access, vol. 9, pp. 11203-11225, 2021, doi: 10.1109/ACCESS.2020.3047929.

[44] L. Zou, M. Zhao, Z. Gao, M. Cao, H. Jia, and M. Pei, License plate detection with shallow and deep CNNs in complex environments, Com-plexity, vol. 2018, pp. 1–6, Dec. 2018.

[45] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, Automatic license plate recognition (ALPR): A state-of-the-art review, IEEE Trans. CircuitsSyst. Video Technol., vol. 23, no. 2, pp. 311–325, Feb. 2013

[46] Pendleton S.D.Andersen H.Du X.X.Shen X.T.Meghjani M.Eng Y.H.Rus D.Ang M.H.Perception, Planning, Control, and Coordination for Autonomous Vehicles. 2017

[47] P.Nousi, D.Triantafyllidou, A.Tefas, Pitas, I. Re-identification framework for long term visual object tracking based on object

[48] D.Zheng, Y.Zhao, J. Wang , detection and classification. Signal Process. Image Commun. 2020, 88, 115969. [CrossRef] An efficient method of license plate location. Pattern Recognition. Lett. 2005, 26, 2431–2438. [CrossRef]

[49] Dr.Islam , Dr. Mohammad Mahfuzul Islam ,M.Sc. Engg. ThesisAutomatic License Plate Detection in Hazardous Condition.

[50] S. Sutton and G. Barto, Reinforcement Learning: An Introduction, The MIT Press Cambridge, Massachusetts London, England,2015.

[51] K. Oei, Locating number plates in cars‐opencv & python, Medium, 24-Mar-2019. [Online]. Available: https://medium.com/@keynekassapa13/locating-number-plates-in-cars-opencv-python-d6894deb5408. [Accessed: 10-May-2022].

[52] Peng, C.Cheng, C.J. Tsai, T.Y. Chang, J.Yeh, Hsun Dai, & M. Tsai. 2020. "A Fast and Noise Tolerable Binarization Method for Automatic License Plate Recognition in the Open Environment in Taiwan" Symmetry 12, no. 8: 1374. https://doi.org/10.3390/sym12081374.

[53] E.Syamsuddin. Designing Automatic Number Plate Recognition (ANPR) Systems Based on K-NN Machine Learning on the Raspberry Pi Embedded System.2018.

[54] J.Pandya. Real Time Badge Detection and Recognition for Bicycle Racing. THESIS. The Ohio State University. 2011

[55] R.Szeliski. Computer Vision - Algorithms and Applications. Texts in Computer Science. Springer, 2011

[56] S. Larsson ,F.Mellqvist. Automatic Number Plate Recognition for Android. Karlstad University. 2019

[57] F.Abtahi, Z.Zhu, and A.M Burry, A deep reinforcement learning approach to character segmentation of license plate images. in 2015 14th IAPR international Conference on Machine Vision Applications (MVA),pages 539-542. IEEE, 2015

[58] Z.Selmi, M.Ben Halima, and Adel M. Alimi. Deep learning system for automatic license plate detection and recognition. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) Volume 1, pages 1132-1138, IEEE. 2017.

[59] M. Zennir Mohales Nadjib. Bensouilah Mourad.Master Thesis Elaboration d'un système de détection et de reconnaissance de plaques minéralogiques algériennes 2019.

[60] W.Liu, D.Anguelov,, D.Erhan, C.Szegedy, S.Reed, C.Fu, & A.C.Berg, Ssd : Single shot multibox detector, In European conference on computer vision ,pages 21-37, Springer, 2016.

[61] K.Simonyan & A.Zisserman. Very deep convolutional networks for large -scale image recognition arXiv preprint arXiv :1409.1556,2014.

[62] R.Girshick Fast r-cnn In Proceedings of the IEEE international conference on computer vision ,pages 1440-1448, 2015

[63] C.Fn ,W.Liu, A.Ranga, A.Tyagi, & A.C.Berg Dssd: Deconvolutional single shot detector .arxiv preprint arXiv :1701.06659,2017

[64] A.Shrivastava. R.Sukthankar, J.Malik, & A.Gubapta . Beyond skip connections :Top-down modulation for object detection.arXiv preprint arxiv :1612.06851 ,2016

[65] T.Lin, P.Dollar ,Ross Girshick, K. He ,B. Hariharan , and S. belongie . Feature pyramid network for object detection.In Proceedings of the IEEE international conference on computer vision and Pattern Recognition ,pages 2117-2125, 2017.

[66] H.Li & C.Shen, Reading car license plates using deep convolutional neural networks and lstms .arXiv preprint arxiv :1601.05610, 2016.

[67] C.Wu, Shungong X.Guocong Song,& S.Zhang. How many labeled license plates are needed?, In Chinese Conference on Pattern Recognition and Computer Vision (PRCV) , pages 334-346.Springer ,2018

[68] G.Huang, Z.Liu , L.Van Der Maaten, & K.Q Weinberger. Densely connected convolutional networks . n Proceedings of the IEEE international conference on computer vision and Pattern Recognition ,pages 4700-4708, 2017.

[69] J.Spanhel, J.Sochor , R.Juranek, A.Heroutn, L.Marsik, & P.Zemcik.Holistic recognition of low quality license plates by cnn using track annotated data. In 2017 14 th IEEE n Proceedings of the IEEE international conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1-6.IEEE,2017.

[70] P.Svoboda, M.Haris, L.Marsik , & P.Zemcik.Cnn for license plate motion deblurring. In 2016 IEEE International Conference on Image Processing (ICIP),pages 3832-3836.IEEE,2016

[71] M. Andersen, A. Carlson, A. Carlson, D. Gronlund, Individual differences predict eyewitness identification performance, Pers. Individ. Differ. 60 (2014) 36–40, http://dx.doi.org/10.1016/j.paid.2013.12.011.

[72] W. Tanaka, D. Kaiser, S. Hagen, J. Pierce, Losing face: impaired discrimination of featural and configural information in the mouth region of an inverted face, Atten. Percept. Psychophys. 76 (4) (2014) 1000–1014, http://dx.doi.org/10.3758/s13414-014-0628-0.

[73] M .Elmahmudi & H.Ugail (2019) Deep face recognition using imperfect facial data. Future Generation Computer Systems. 99: 231-225.

[74] K. Tarek, C. Rasha. Facial recognition in real-time. Bachelor thesis. Université de Blida 1. 2020.

[75] Detect faces | cloud vision API | google cloud, Google. [Online]. Available: https://cloud.google.com/vision/docs/detecting-faces. [Accessed: 17-May-2022].

[76] F.Schroff, D.Kalenichenko, J.Philbin, A unified embedding for face recognition and clustering, Conference, 2015.

[77] D.Jiankang, G.Jia, X.Niannan, & Z.Stefanos. Arc Face: Additive angular margin loss for deep face recognition. In Computer Vision and Pattern Recognition(CVPR), pages 4690–4699, 2019.

[78] B.Amos, B.Ludwiczuk, & M.Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. CMU School of Computer Science, (2016), 2016.

[79] F.Schroff, D.Kalenichenko, & J.Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[80] S.Bharadwaj, S.Bhatt, M.Vatsa, & R.Singh. Domain-specific learning for newborn face recognition. IEEE Transactions on Information Forensics and Security, 11(7):1630–1641, 2016

[81] L.Zheng, K. Idrissi, C.Garcia, S.Duffner, & A.Baskurt. Triangular similarity metric learning for face verification. In 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), volume 1, pages 1–7. IEEE, 2015.

[82] T. Ojala, M. Pietikinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. Pattern Recognition, 29:51–59, 1996.

[83] H. Wang, P. Phillips, C. Dong, and D. Zhang. Intelligent facial emotion recognition based on stationary wavelet entropy and jaya algorithm. Neurocomputing, 272:668—-676, 2018.

[84] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 497—- 504, 2011.

[85] F. Yang, W. Yang, R. Gao, and Q. Liao. Discriminative multidimensional scaling for low-resolution face recognition. IEEE Signal Process Letter, 25(3):388—-392, 2018.

[86] Shadow system. [Online]. Available: https://shadowsystem.com/. [Accessed: 13-May-2022].

[87] A. Nadhim Razzaq, R. Ghazali, N. Khdhair El Abbadi, M. Dosh, A Comprehensive Survey on Face Detection Techniques, Webology, Volume 19, Number 1, January 2022

[88] A.Sakhri. Optimisation de la détection pour la reconnaissance de visage.

[89] M.ang, D.Kriegman, N.Ahuja, Detecting faces in images: A survey. IEEE Transactions on pattern analysis and machine intelligence, 24(1):34-58, 2002.

[90] A.Soufi, I.Addou, M.Kohili, et al. Réalisation d'un système de détection des visages en utilisant la matrice PSSM-positionspecific-scoring-matrix. PhD thesis, Ahmed Draia University -ADRAR, 2018.

[91] Z.Khawla, M.Khaoula, Détection de visage,Thesis,Larbi Ben M'hidi University of Oum el Bouaghi, 2021.

[92] D.Bouzit, Reconnaissance de visage basée sur une approche triangulaire, 2019.

[93] K.Belhouchette, H.Belhadef, Modélisation de l'état affectif dans les séquences vidéo, 2018.

[94] C.Berkane, A.Berkani, La détection des visages, 2014.

[95] Z.Yakouta, B.Safia, Détection de visages dans un environnement complexe, 2011.

[96] T.Imène, Le système biométrique : détection et reconnaissance de visage, Thesis, mohamed boudhief University Oran, 2019.

[97] B.Yassine, T.Seddik, Reconnaissance Du Visage Dans Des Conditions Incontrôlées,Thesis,8 Mai 1945 University Guelma,2020.

[98] P.Buyssens, Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e-transactions, Thesis, CAEN University, 2006.

[99] N. Morizet, Revue des algorithmes PCA, LDA et EBGM utilisés en Reconnaissance 2D du visage pour la biométrie, Thesis, Higher Institute of Electronics Paris (ISEP), 2006.

[100] S. Ababsa, Authentification d’individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D , Thesis, Evry University Valley of Essonne, 2008.

[101] Guo, Guodong, Z. Li, K.Chan. Face recognition by support vector machines, Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. no. PR00580). IEEE, 2000.

[102] F.Cardinaux, C.Sanderson, S.Bengio, User Authentication via Adapted Statistical Models of Face Images, In the IEEE Transaction on Signal Processing. Vol. 54, Issue 1, Jan 2006, Pages: 361-373.

[103] M.Benkiniouar, M.Benmohamed, Méthodes d'identification et de reconnaissance de visages en temps réel basées sur AdaBoost, P2-3, 2005.

[104] S.Mekkani, Reconnaissance de visage , Thesis, Larbi Ben M'hidi University Oum El Bouaghi, 2014.

[105] B.Chu, Neutralisation des expressions faciales pour améliorer la reconnaissance du visage , Thesis, Computer Science and Mathematics Doctoral School, 2015.

[106] S.Liu, M.Li ,M.Li ,Q.Xu, Research of animal image semantic segmentation based on deep learning, 2018.

[107] O.Ronneberger, P.Fischer,T.Brox Computer Science Department and BIOSS Center for Biological Signaling Studies, University of Freiburg, Germany,2015

[108] M.Sandler, A.Howard, M.Zhu, A.Zhmoginov, L.Chen, Google Inc.MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2019

[109] A.Howard, M.Sandler, G.Chu, L.Chen, B.Chen, M.Tan,W.Wang, Y.Zhu, R.Pang, V.Vasudevan, Q.Le, H.Adam, Google AI, Google Brain, Searching for MobileNetV3, 2019

[110] P.Viola, M.Jones, Rapid Object Detection using a Boosted Cascade of Simple Features,2001.

[111] F.Almatrooshi, I.Akour, S.Alhammadi, K.Shaalan, S.Salloum, A Recommendation System for Diabetes Detection and Treatment. 10.1109/CCCI49893.2020.9256676. 2020.

[112] F.Schroff, D.Kalenichenko, & Philbin, FaceNet: A unified embedding for face recognition and clustering. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815-823. 2015.

[113] P.Yoon Soo, L.Young-Sun, Diagnostic cluster analysis of mathematics skills. IERI Monograph Series: Issues and Methodologies in Large-Scale Assessments. 4. 75-108, 2011.