

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes De MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Electronique

Spécialité : Instrumentation

Présenté par

Prénom Abdous Youcef

Prénom Ameer Idir

Thème

Parking Intelligent avec système de paiement en ligne avec plaque d'immatriculation

Mémoire soutenu publiquement le 25/06/ 2024 devant le jury composé de :

M Fethi Ouallouche

Enseignant au département d'électronique ummto Président

M Djamal Alouache

Enseignant au département d'électronique ummto, Promoteur

M Slimane Hamag

Enseignant au département d'électronique ummto, Examineur

Mme Oudjemia

Enseignant au département d'électronique ummto, Examinatrice

Résumé

Les paiements en ligne est une technologie qui a vu le jour aux alentours des années 90, c'est une méthode électronique de transfert de fonds d'une partie à une autre à travers une plateforme numérique d'une manière rapide et sécurisé grâce à des cartes bancaires, des cartes rechargeables ou de portemonnaies virtuels.

Les systèmes de Reconnaissance Automatique de Plaque d'Immatriculations (RAPI) sont des technologies utilisées pour lire et identifier les plaques d'immatriculations des véhicules à partir d'image ou de vidéos.

Depuis les années 80, les systèmes de Reconnaissance Automatique de Plaque d'Immatriculations (RAPI) ont connu diverses améliorations et évoluent de manière exponentielle et deviennent indispensables dans nombreux domaine tel que la surveillance de la sécurité, la gestion du trafic, et le contrôle des accès.

Ces systèmes sont divisés en deux parties qui sont : La détection de la plaque d'immatriculations qui consiste à identifier et reconnaître les plaques, et La reconnaissance de caractères alphanumériques qui consiste à l'indentification des caractères contenu dans les plaques et leurs extractions. Et pour cela, de nombreuses techniques sont utilisées et chacune avec ses avantages et ses défauts.

Dans ce projet de fin d'étude, nous avons pour objectif de concevoir un Système de Reconnaissance Automatique de Plaque d'Immatriculations Algériennes pour l'accès aux parkings en y implémentant un système de paiement automatique par internet. Ce système est conçu sur un Raspberry Pi 3 programmé en langage Python. Ce projet est divisé en trois grandes fonctions. La première, c'est la détection des plaques d'immatriculations et la reconnaissance des caractères, la deuxième c'est le contrôle des capteurs et des actionneurs pour assurer l'entrée et la sortie des véhicules, et la troisième c'est le processus de paiement par internet.

Abstract

Online payments are a technology that emerged around the 90s, it is an electronic method of transferring funds from one party to another through a digital platform in a fast and secure way thanks to bank cards, rechargeable cards or virtual wallets.

Automatic License Plate Recognition (ALPR) systems are technologies used to read and identify vehicle license plates from images or videos.

Since the 1980s, Automatic Registration Plate Recognition (ALPR) systems have seen various improvements, and are evolving exponentially, becoming indispensable in many fields such as security monitoring, traffic management, and access control.

These systems are divided into two parts, which are: The detection of the license plate, which consists in identifying and recognizing the plates, and the recognition of alphanumeric characters, which consists in the identification of the characters contained in the plates and their extractions. For this, many techniques are used and each one with its advantages and flaws.

In this final project, we aim to design an Algerian Automatic License Plate Recognition System for access to car parks by implementing an automatic payment system by internet. This system is designed on a Raspberry Pi 3 and programmed in Python language. This project is divided into three main functions. The first is the detection of license plates and character recognition, the second is the control of sensors and actuators to ensure the entry and exit of vehicles, and the third is the payment process via the internet.

ملخص

المدفوعات عبر الإنترنت هي تقنية ظهرت في التسعينيات تقريباً، وهي طريقة إلكترونية لتحويل الأموال من طرف إلى آخر من خلال منصة رقمية بطريقة سريعة وآمنة بفضل البطاقات المصرفية أو البطاقات القابلة لإعادة الشحن أو المحافظ الافتراضية.

هي تقنيات تستخدم لقراءة وتحديد لوحات ترخيص السيارة من (ALPR) أنظمة التعرف التلقائي على لوحة الترخيص الصور أو مقاطع الفيديو.

لتحسينات مختلفة وهي تتطور بشكل كبير (ALPR) منذ الثمانينيات، خضعت أنظمة التعرف التلقائي على لوحة التسجيل وأصبحت لا غنى عنها في العديد من المجالات مثل المراقبة الأمنية وإدارة حركة المرور ومراقبة الدخول.

تنقسم هذه الأنظمة إلى جزأين هما: الكشف عن لوحة الترخيص التي تتكون من تحديد اللوحات والتعرف عليها، والتعرف على الأحرف الأبجدية التي تتكون من تحديد الأحرف الموجودة في اللوحات واستخراجها. ولهذا، يتم استخدام العديد من التقنيات ولكل منها مزاياها وعيوبها.

في هذا المشروع النهائي، نهدف إلى تصميم نظام التعرف التلقائي على لوحة الترخيص الجزائري للوصول إلى مواقف المبرمج بلغة Raspberry Pi 3 السيارات من خلال تطبيق نظام الدفع التلقائي عبر الإنترنت. تم تصميم هذا النظام على وينقسم هذا المشروع إلى ثلاث وظائف رئيسية. الأول هو اكتشاف لوحات الترخيص والتعرف على الأحرف، Python. والثاني هو التحكم في أجهزة الاستشعار والمشغلات لضمان دخول المركبات وخروجها، والثالث هو عملية الدفع عبر الإنترنت

Table des matières

Introduction.....	14
Chapitre 1 : Généralité sur l'image et le traitement d'image	
Préambule.....	17
Table des matières.....	4
1 Image :.....	17
1.1 Image analogique :	17
1.2 Image numérique :	17
2 Représentation d'image :.....	18
2.1 Image matricielle :	18
2.2 Image Vectorielle :	19
19	
3 Echantillonnage et quantification :.....	19
4 Image à niveaux de gris :	20
5 Image en couleur :	20
6 Caractéristiques d'une image numérique :.....	21
6.1 Dimension :	21
6.2 Résolution :	21
6.3 Bruit :	21
6.4 Histogramme :	22
6.5 Luminance :	23
6.6 Contraste :	24
6.7 Seuillage :	25
7 Colorimétriques de représentation des images :	25
8 Mode couleur RVB :	25
8.1 Mode couleur CMJN :	26
8.2 HSV (Teinte, Saturation, Valeur) :	26
8.3 LAB (Luminosité, a, b) :	27
9 Traitement d'image :	27
10 Acquisition d'une image :	28
11 Prétraitements d'images :.....	28
11.1 Amélioration d'image :	28
11.1.1 La binarisation :	29

11.1.2	Lissage :.....	30
11.1.3	La normalisation :.....	30
11.1.4	Détection de contours :.....	31
11.1.5	Segmentation :.....	34
11.2	Analyse d'image :.....	35
Discussion.....		35

Chapitre 2 : Etat de l'art des systèmes RAPI

Préambule.....		37
1	Définition de la plaque d'immatriculation :	37
2	Caractéristiques des plaques algériennes :	38
3	Etat de l'art des méthodes RAPI :	38
3.1	Méthodes basées sur le traitement d'images :.....	38
3.2	Méthodes basées sur l'apprentissage profond :.....	39
3.3	Méthodes hybrides :.....	39
3.4	Utilisation de bases de données et de techniques de recherche :.....	39
3.5	Optimisation et traitement en temps réel :.....	39
4	Difficulté des systèmes de reconnaissance des plaques d'immatriculation :	39
4.1	Variabilité des plaques :.....	39
4.2	Conditions environnementales :.....	40
5	Les techniques de Reconnaissance des plaques d'immatriculation :.....	40
5.1	Technique basée sur la couleur de la plaque :.....	40
5.2	Technique basée sur la taille des caractères et leur distance les uns des autres :.....	41
5.3	Technique basée sur les contours de la plaque :.....	41
5.4	Technique basée sur la texture de la plaque :.....	42
5.5	Technique basée sur la heatmap (Carte de chaleur) :.....	43
5.6	Technique basée sur l'infrarouge :.....	44
6	Techniques de reconnaissances de caractères et de chiffres :	44
6.1	Technique de reconnaissance par détection de contours :.....	45
6.2	Techniques de reconnaissance par intelligence artificiel :.....	45
6.2.1	Reconnaissance optique des caractères OCR :.....	45
6.2.2	Etape de l'OCR :.....	45
7	Résumé des Techniques utilisées :	46
7.1	Techniques de détection :.....	46
7.2	Techniques de reconnaissance :.....	48
8	Domaines d'utilisation des systèmes RAPI :	48

8.1	Gestion du trafic et péage automatique :	48
8.2	Gestion des Parkings :	49
8.3	Identification des véhicules :	49
Discussion.....		50

Chapitre 3 : Les reseaux de neurones dans les systèmes RAPI

Préambule.....		52
1	L'intelligence artificielle :	52
1.1	Histoire de l'intelligence artificielle :	52
1.2	Importance de l'intelligence artificielle dans notre ère :	53
2	Les sous-ensembles de l'intelligence artificielle :	54
2.1	Machine Learning :	54
2.1.1	Fonctionnement du Machine Learning :	55
2.1.2	Les sous-domaines du Machine Learning :	56
2.2	Réseau neuronal :	58
2.2.1	Neurone Artificiel :	58
2.2.2	Couche de neurones :	59
2.2.3	Fonctionnement :	60
2.2.4	Les différentes architectures du réseau neuronal :	60
3	Les Réseaux neuronaux convolutifs :	61
3.1	Composants et opérations des CNN :	61
3.1.1	Couche de convolution :	61
3.1.2	Couche de pooling :	62
3.1.3	Couche d'activation ReLU :	63
3.1.4	Couches Sigmoides :	64
3.1.5	Couches Fully-Connected :	65
3.2	Fonctionnement des couches des CNN :	66
3.3	Entraînement d'un modèle CNN :	66
4	Deep Learning :	68
4.1	Fonctionnement du Deep Learning :	68
4.1.1	Comment le deep-learning utilise les réseaux CNN :	68
5	Reconnaissance de caractères avec Deep Learning :	69
5.1	EasyOCR :	69
5.1.1	Détection des zones dans les images contenant du texte :	70
5.1.2	Architecture du modèle Deep Learning utilisée dans EasyOCR (CRNN) :	70
5.1.3	Décodage des sorties :	71

5.2	Post-traitement des caractères reconnus :	71
6	Application de réseaux neuronaux dans les systèmes RAPI :	71
6.1	Deep Learning pour la détection de plaque d'immatriculation :	71
6.2	Deep Learning pour la reconnaissance de chiffres de la plaque d'immatriculation :	72
	Discussion.....	73
Chapitre 4 : Conception et réalisation		
	Préambule.....	75
1	Conception de Notre système:	75
1.1	Schéma global du fonctionnement de notre Système :	75
1.2	Matériels utilisés :	77
1.2.1	Carte Raspberry pi 3 modèle B :	77
1.2.2	Caméra Raspberry Pi :	78
1.2.3	Carte Arduino UNO :	79
1.2.4	Servo Moteur :	80
1.2.5	Capteur Infrarouge :	81
1.3	Logiciels et Programmes utilisés :	82
1.3.1	Langage de programmation Python :	82
1.3.2	Tensorflow :	83
1.3.3	TensorFlow Lite :	84
1.3.4	EasyOcr :	85
1.3.5	Tkinter :	86
1.3.6	OpenCV :	86
1.4	Organigramme de la détection :	87
1.4.1	Importation des bibliothèques :	88
1.4.2	Chemins des fichiers et initialisation :	88
1.4.3	Configuration du modèle TensorFlow Lite :	89
1.4.4	Lecture des labels :	89
1.4.5	Initialisation d'EasyOCR :	89
1.4.6	Fonction de détection et d'affichage :	90
1.4.7	Configuration de l'interface graphique Tkinter :	91
1.5	Jupyter Notebook :	92
2	Architecture du modèle CNN :	93
2.1	Extraction de chiffres avec Easyocr :	94
2.1.1	Importer la bibliothèque Easyocr :	94
2.1.2	Initialisation de Easyocr :	94

2.1.3	Fonction de Détection et Reconnaissance de Texte :	94
3	Utilisation de la carte Raspberry Pi :	94
3.1	Le système Raspbian :	94
3.2	Installer Raspbian et connecter la carte à un PC via un câble RJ45 :	94
3.3	Affichage de l'interface Raspbian sur PC via RJ45 :	99
3.4	Connexion de la carte Arduino à la carte Raspberry pi :	101
3.4.1	Configuration de la carte Arduino :	102
4	Les systèmes de paiements :	102
5	Plateforme de paiement élaborée :	103
5.1	Architecture du site web :	103
5.1.1	Page d'accueil :	103
5.1.2	Page d'inscription et de connexion :	103
5.1.3	Page de gestion de factures :	105
5.2	Base de données :	105
5.2.1	Création de Base de Donnée :	105
5.2.2	Schémas de Données	106
5.2.3	Sécurité :	107
5.2.4	Diagramme de la base de données :	107
5.3	Schéma fonctionnel global :	108
6	Schéma Final :	109
7	Branchement de la carte Arduino :	110
8	Connexion de la carte Arduino :	111
9	Tests et résultats :	112
9.1	Résultats de détection et de reconnaissance de la plaque :	112
9.1.1	Reconnaissance et détection :	113
9.2	Erreur de reconnaissance des chiffres de la plaque :	115
10	Comparaison entre EasyOCR et Tesseract :	118
11	Résultats du système de paiement :	119
	Discussion.....	124
	Conclusion.....	126

Liste des figures

Figure 1 Représentation image numérique.....	18
Figure 2 Image matricielle	18
Figure 3 Comparaison entre image matriciel et image vectorielle.....	19
Figure 4 Image a niveaux de gris	20
Figure 5 Bruit d'image.....	22
Figure 6 Histogramme de l'image	23
Figure 7 Luminance	24
Figure 8 Contraste	24
Figure 9 Seuillage.....	25
Figure 10 Mode de couleur CMJN et RVB.....	26
Figure 11 HSV	27
Figure 12 Schéma Synoptique du traitement d'image.....	28
Figure 13 Binarisation d'image	29
Figure 14 Lissage d'image.....	30
Figure 15 La normalisation d'image.....	31
Figure 16 Contours d'image avec le filtre Sobel	32
Figure 17 Contours avec le filtre de Prewitt	32
Figure 18 Contours avec le filtre de Roberts.....	33
Figure 19 Contours avec le filtre Laplace	33
Figure 20 Contours avec le filtre Canny	34
Figure 21 Segmentation par détection de contours	35
Figure 22 Caractéristiques des plaques algériennes	38
Figure 23 Technique RAPI basée sur la couleur de la plaque.....	41
Figure 24 Technique RAPI basée sur les contours	42
Figure 25 Technique RAPI basée sur la texture de la plaque	43
Figure 26 Technique basée sur le Heatmap.....	43
Figure 27 RAPI dans la gestion de trafic et de péage	48
Figure 28 Système RAPI pour la gestion de parking.....	49
Figure 29 Système RAPI pour l'identification de véhicules	49
Figure 30 Schéma synoptique du fonctionnement du Machine Learning.....	54
Figure 31 Schéma synoptique du processus de fonctionnement du Machine Learning	55
Figure 32 Apprentissage supervisé	56
Figure 33 Apprentissage non supervisé	57
Figure 34 Equation de base de la fonction d'activation de neurone artificiel	58
Figure 35 Couches de neurone artificiel.....	59
Figure 36 Réseaux de neurones convolutifs.....	61
Figure 37 Couche de convolution	62
Figure 38 Couche de Pooling	63
Figure 39 Fonction d'activation ReLU	64
Figure 40 Fonction de Sigmoidé	65
Figure 41 Couche Fully Connected.....	66
Figure 42 Fonctionnement des couches de CNN	66
Figure 43 Schéma synoptique d'entrainement d'un modèle CNN.....	67

Figure 44 Utilisation des CNN par le deep Learning.....	69
Figure 45 Détection de zones	70
Figure 46 Détection de plaque d'immatriculation avec le Deep Learning	72
Figure 47 Schéma global du fonctionnement de notre Système	75
Figure 48 Organigramme du fonctionnement de notre système	76
Figure 49 Carte Raspberry Pi	77
Figure 50 Caméra Raspberry pi	79
Figure 51 Carte Arduino Uno.....	80
Figure 52 Servo Moteur	81
Figure 53 Capteur infrarouge	81
Figure 54 Python	82
Figure 55 Python et Machine Learning	83
Figure 56 TebsorFlow	83
Figure 57 TensorFlow Lite.....	84
Figure 58 EasyOCR	85
Figure 59 OpenCV	86
Figure 60 Organigramme de la détection	87
Figure 61 Importation des bibliothèques.....	88
Figure 62 Chemins des fichiers et initialisation	88
Figure 63 Initialisation de la caméra	88
Figure 64 Configuration du modèle TensorFlow Lite	89
Figure 65 Lecture des labels.....	89
Figure 66 Initialisation d'EasyOCR.....	89
Figure 67 Fonction de détection et d'affichage	90
Figure 68 Configuration de l'interface graphique Tkinter.....	91
Figure 69 Libération des ressources	91
Figure 70 Jupyter Notebook.....	92
Figure 71 Architecture du modèle CNN	93
Figure 72 Importer la bibliothèque Easyocr.....	94
Figure 73 Initialisation de Easyocr.....	94
Figure 74 Le système Raspbian	94
Figure 75 Téléchargement de l'image de Raspberry Pi OS.....	95
Figure 76 Installation de Raspberry Pi Imager.....	96
Figure 77 Raspberry pi Imager.....	96
Figure 78 Choix du système d'exploitation dans Raspberry pi Imager.....	97
Figure 79 Configuration de l'interface réseau	98
Figure 80 Commande de l'adresse IP de la Raspberry Pi.....	98
Figure 81 Connexion à la Raspberry Pi via SSH	98
Figure 82 Mise à jour du système	99
Figure 83 Commande d'affiche de l'interface Raspbian.....	99
Figure 84 Commande de configuration de VNC server.....	99
Figure 85 Interface de VNC	100
Figure 86 Interface Graphique de Raspbian.....	100
Figure 87 Code d'initialisation de la communication entre Arduino et Raspberry pi	101
Figure 88 Code de configuration de la carte Arduino	102
Figure 89 Page d'accueil du site web	103
Figure 90 Page d'inscription.....	104

Figure 91 Page de connexion	104
Figure 92 Page de gestion de factures	105
Figure 93 Base de données	105
Figure 94 Schéma synoptique de la base de données.....	106
Figure 95 Schéma de données	106
Figure 96 Diagramme fonctionnel de la base de données.....	107
Figure 97 Schéma Global	108
Figure 98 Schéma final	109
Figure 99 Branchement des composants avec l'Arduino	110
Figure 100 Diagramme fonctionnel du programme Arduino.....	111
Figure 101 importation de la bibliothèque Serial	111
Figure 102 Initialisation de la connexion Série.....	112
Figure 103 Temps d'attente	112
Figure 104 Détection et reconnaissance de la plaque.....	113
Figure 105 Détection et lecture de la plaque	113
Figure 106 Résultats de la reconnaissance et de la détection de la plaque	114
Figure 107 Graphe de la détection de la plaque	115
Figure 108 Erreurs de reconnaissance de plaque	116
Figure 109 Graphe de confiance et d'erreur	117
Figure 110 Graphe d'erreur de lecture de la plaque	117
Figure 111 Inscription du client	120
Figure 112 Profile du client.....	120
Figure 113 Ajout de solde	121
Figure 114 Ajout de véhicule	121
Figure 115 Enregistrement d'entrée.....	122
Figure 116 Enregistrement de la sortie	122
Figure 117 Paiement de la facture	123
Figure 118 Rubrique admin.....	123

Liste des Tableaux

Tableau I Comparaison des techniques de détection de la plaque d'immatriculation	47
Tableau II Comparaison des techniques de reconnaissance de la plaque d'immatriculation ...	48
Tableau III Comparaison entre EasyOCR et Tesseract.....	119

Introduction

Introduction

Selon les statistiques, le nombre de véhicules automobiles est constitué d'environ 1.8 milliards de véhicules en circulation. Et selon les estimations, ce chiffre est constamment en croissance et pourrait doubler d'ici 2040. En Algérie, le parc automobile compte plus de 7 millions de véhicules en circulation en 2019 [21].

Et vu le grand flux de voitures en circulation, les Systèmes de Reconnaissances de Plaque d'Immatriculations (RAPI) ont vu le jour afin de faciliter le processus de reconnaissance et d'identification des voitures.

Les systèmes RAPI sont généralement dotés d'une caméra reliée à un PC ou un Serveur ; le PC traite et extrait les données de la plaque d'immatriculation envoyé par la caméra ce qui permet d'identifier et d'avoir des informations sur le véhicule en question. Le processus de reconnaissance automatique des plaques d'immatriculation (RAPI) peut avoir de nombreuses applications. Il peut être utilisé dans les parcs de stationnement, les postes de frontières et de péage, la surveillance des autoroutes ...

L'objectif principal de ce travail est la conception et réalisation d'un système électronique pour la gestion des parkings, en utilisant la reconnaissance automatique des plaques d'immatriculation (RAPI). En effet, pour mettre en profit le système de reconnaissance automatique des plaques d'immatriculation dans notre application, nous avons implémenté un système de paiement en ligne relié à une base de données où les plaques d'immatriculation feront office d'une carte bancaire.

De plus, dans notre système, le paiement est effectué rapidement et facilement via un smartphone c'est-à-dire, un paiement sans contact et sans tickets ainsi les clients n'auront plus besoin de faire la queue dans les bornes de paiements ce qui permet aussi une fluidité du trafic et un meilleur confort pour les clients.

En outre, notre système non seulement facilite la tâche aux clients, mais aussi permet une bonne gestion et organisation du parking pour les propriétaires, notamment en diminuant les coûts opérationnels et une collecte de données qui permettra d'évaluer et de connaître leurs clientèles.

Notre système est implémenté sur une carte Raspberry Pi 3 programmée sous le langage Python, connectée à une base de données qui gère le paiement en ligne, la carte Raspberry gère aussi la partie de détection et de reconnaissance de la plaque d'immatriculations fondée sur l'utilisation d'un réseau de neurones.

Introduction

Pour structurer notre travail, nous l'avons devisé en quatre chapitres, comme suit :

- Dans le premier chapitre, nous allons voir les généralités sur la vision artificielle et les notions de base du traitement d'image pour pouvoir les utiliser à notre profit.
- Le deuxième chapitre contient des informations sur les méthodes utilisées dans la détection de plaque d'immatriculations et la reconnaissance de caractères en citant les avantages et les défauts de chaque méthode.
- Le troisième chapitre va se concentrer sur l'utilisation et fonctionnement du Deep Learning dans la vision artificielle.
- Dans le quatrième chapitre, nous allons présenter la mise en œuvre de notre système RAPI côté matériels et logiciels en y implémentant le système de paiement et enfin l'analyse des résultats obtenus.

Chapitre 1 :

Généralités sur la vision artificielle

Préambule :

La vision artificielle est un domaine vaste et indispensable dans notre ère, elle est retrouvée dans de nombreux champs d'utilisation tel que l'imagerie médicale, la reconnaissance faciale, la sécurité routière et bien d'autres. Dans ce chapitre nous allons tout d'abord comprendre les caractéristiques fondamentales d'une image numérique avant de pouvoir s'intéresser aux fondements du traitement et d'analyse d'image et comprendre le fonctionnement de ce processus et examiner les différentes étapes qui composent un système de vision artificielle, depuis l'acquisition de l'image jusqu'à son interprétation finale.

1 Image :

Une image est une représentation visuelle ou graphique d'un objet, d'une scène, d'une personne ou de tout autre élément. Elle est utilisée pour communiquer des idées, des informations ou des émotions d'une manière visuelle. Les images peuvent être capturées par des dispositifs tels que des appareils photo, des caméras vidéo, des scanners ou créées à l'aide de logiciels de conception graphique.

1.1 Image analogique :

Une image analogique est une représentation visuelle d'un objet ou d'une scène qui est enregistrée sur un support physique, tels que du film photographique ou une plaque photographique. Contrairement aux images numériques, les images analogiques ne sont pas représentées par des données binaires mais plutôt par des valeurs continues. Elles sont souvent produites par des processus chimiques ou optiques et peuvent être imprimées sur du papier photographique ou projetées sur des surfaces physiques. Les images analogiques présentent une gamme continue de valeurs de couleur et de luminosité, ce qui les distingue des images numériques qui sont discrètes.

1.2 Image numérique :

Une image numérique est une représentation d'une image sous forme de données binaires, composée de pixels (points d'image) organisés en une grille. Chaque pixel contient des informations sur la couleur et la luminosité de l'image. Les images numériques peuvent être stockées, éditées, partagées et affichées sur des dispositifs électroniques tels que des ordinateurs, des smartphones et des tablettes. Elles sont généralement créées à l'aide de capteurs électroniques comme les capteurs CCD ou CMOS dans les appareils photo numériques.

Une image est un ensemble de pixels pour un ordinateur, chaque pixel est un élément individuel de l'image caractérisé par sa position (x, y) et sa valeur numérique, et la combinaison de tous les pixels crée l'image numérique.

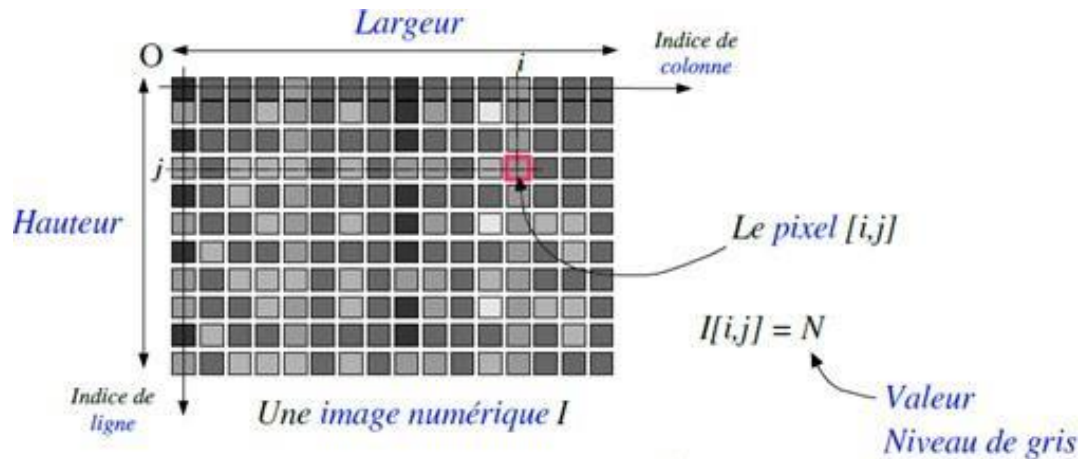


Figure 1 Représentation image numérique

2 Représentation d'image :

Lorsque nous regardons une image sur notre écran d'ordinateur, que ce soit une photo, un dessin ou une icône, il est important de comprendre qu'elle appartient à l'une des deux grandes catégories d'images numériques : les images matricielles et les images vectorielles.

2.1 Image matricielle :

Un ensemble de points ou de pixels forment une image matricielle (images bitmap), on parle de pixels pour les images stockées sous forme « binaire », et de points lorsque ces images sont imprimées ou destinées à l'impression.

L'image matricielle est composée d'un pavage carré à plusieurs dimensions spatiale, temporelle ou autre.



Figure 2 Image matricielle

2.2 Image Vectorielle :

L'image vectorielle est définie par des formes géométriques individuelles, telles que des cercles, Des segments de droite, des arcs et bien d'autres formes, chaque élément étant défini par des attributs précis de forme de couleur et bien plus encore. Imaginez un cercle : sa représentation vectorielle se résume à des informations mathématiques telles que la position du centre et le rayon, lui permettant d'être reconstruit à la perfection à chaque affichage.

Les images vectorielles peuvent être redimensionnées à l'infini sans aucune perte de qualité c'est l'avantage unique qu'offre ce mode de construction. Les images vectorielles complexes peuvent prendre plus de temps à s'afficher par rapport aux images matricielles, puisqu'elles nécessitent une puissance de calcul et une mémoire importante.

En contrepartie les images vectorielles offrent un atout majeur, la précision mathématique de leur construction permet de stocker l'image avec un minimum d'informations donc des fichiers plus légers que les images matricielles équivalentes.

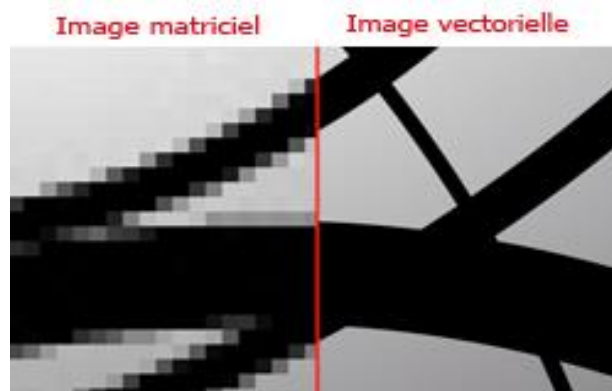


Figure 3 Comparaison entre image matriciel et image vectorielle

3 Échantillonnage et quantification :

Lorsqu'on numérise une image, on la convertit en un format compréhensible par les ordinateurs, ce qui implique les deux étapes cruciales l'échantillonnage et la quantification.

L'échantillonnage consiste à diviser l'image en une matrice de pixels, plus le nombre de pixels est élevé plus la résolution de l'image sera fine et détaillée. La quantification quant à elle, est un procédé qui consiste à limiter le nombre de valeurs différentes que peut prendre chaque pixel d'une image numérique, donc à attribuer une valeur numérique à chaque pixel, cette valeur représente la couleur et la luminosité du pixel.

4 Image à niveaux de gris :

Chaque pixel de l'image est représenté par une seule valeur numérique correspondant à sa luminosité. L'image ne comporte que des nuances de gris, du noir au blanc. Imaginons une échelle allant du noir profond (valeur 0) au blanc éclatant (valeur 255), où chaque nuance de gris trouve sa place. C'est ainsi que les images numériques à niveaux de gris prennent vie, encodant la lumière et l'ombre dans un langage binaire, plus le nombre de bits utilisés pour coder chaque pixel est élevé, plus la palette de gris est riche et subtile.



Figure 4 Image à niveaux de gris

5 Image en couleur :

Les applications multimédias ont souvent recours à des images en couleur pour offrir une expérience visuelle plus riche. Toutefois, il est parfois nécessaire d'utiliser des images en noir et blanc, notamment pour des besoins spécifiques ou pour des raisons esthétiques. En général, le traitement des images en couleur repose sur des techniques similaires à celles utilisées pour les images monochromes. Lorsqu'il s'agit de représenter des images en couleur, le choix du modèle de représentation est crucial. Les composantes primaires telles que le rouge, le vert et le bleu (RVB) sont couramment utilisées pour décrire la couleur des pixels. Ce modèle explique comment les couleurs sont formées en combinant différentes intensités de rouge, de vert et de bleu.

En pratique, les écrans d'ordinateur et d'autres dispositifs utilisent ces informations RVB pour afficher une gamme variée de couleurs. Par exemple, la figure suivante illustre une

image en couleur où chaque pixel est représenté par une combinaison spécifique de valeurs de rouge, de vert et de bleu.

En résumé, le choix du modèle de représentation des couleurs est fondamental pour manipuler et afficher efficacement des images en couleur, tout en permettant une reproduction fidèle des nuances et des teintes désirées.

6 Caractéristiques d'une image numérique :

6.1 Dimension :

La dimension d'une image est sa taille, exprimée en nombre de pixels, un pixel représente une petite zone de couleur et de luminosité qui est un élément individuel de l'image. L'image est stockée sous forme de matrice, un tableau à deux dimensions. Chaque ligne et chaque colonne de la matrice correspond à un pixel, ce qui fait que la résolution de l'image est déterminée par le nombre de lignes et de colonnes.

6.2 Résolution :

La résolution d'une image est sa capacité à afficher des détails fins et précis, elle est généralement exprimée en pixels par unité de longueur, comme les pixels par pouce (ppi) ou les pixels par centimètre (ppcm).

Plus la résolution est élevée plus l'image est nette et détaillée, par exemple, une image de 300 ppi aura une meilleure résolution qu'une image de 72 pp. La résolution est définie par le nombre de pixels horizontaux et verticaux sur les écrans d'ordinateur, par exemple, une résolution de 1920 x 1080 signifie que l'écran affiche 1920 pixels horizontalement et 1080 pixels verticalement.

Une résolution plus élevée est généralement meilleure, elle est un facteur important à prendre en considération lors du choix d'un écran ou d'une imprimante.

6.3 Bruit :

Le bruit dans une image (parasite), est une variation aléatoire de la luminosité ou de la couleur des pixels, il provient de l'éclairage des dispositifs optiques et électroniques du capteur, d'une sensibilité ISO élevée ou d'un problème technique.[3]

On peut distinguer deux types de bruits d'image qui s'accumulent :

- Le bruit de chrominance, qui est une variation aléatoire de la couleur des pixels d'une image, il est visible sous forme de taches de couleurs aléatoires généralement rouge, verte ou bleu, qui dégradent la qualité de l'image. Le bruit de chrominance est

généralement causé par un faible niveau d'éclairage ou une sensibilité ISO élevée, comme il peut également être causé par un problème technique avec l'appareil photo ou le traitement de l'image.

- Le bruit de luminance, qui est la composante lumineuse des pixels bruités, Il se manifeste par des taches grises ou des grains qui dégradent la qualité de l'image, lui donnant un aspect granuleux, il est souvent plus visible dans les zones sombres d'une image.



a) Image originale



b) Bruit Blanc gaussien



c) Bruit poivre de sel



d) Bruit multiplicatif

Figure 5 Bruit d'image

6.4 Histogramme :

L'histogramme d'une image est un graphique qui révèle la répartition des niveaux de gris ou des couleurs dans l'image, il montre combien de pixels de l'image ont chaque niveau de gris ou de couleur, et il permet de donner des informations sur la distribution des niveaux de gris, c'est la fonction discrète "h" telle que " $h(i) = n$ ", où n est le nombre de pixels de l'image ayant l'intensité i, que l'on appelle « effectif ».[1]

L'histogramme n'est pas qu'un simple outil d'analyse d'image, il est aussi utilisé pour améliorer la qualité de l'image (Rehaussement d'image) en introduisant quelques modifications pour pouvoir extraire les informations utiles de celle-ci. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

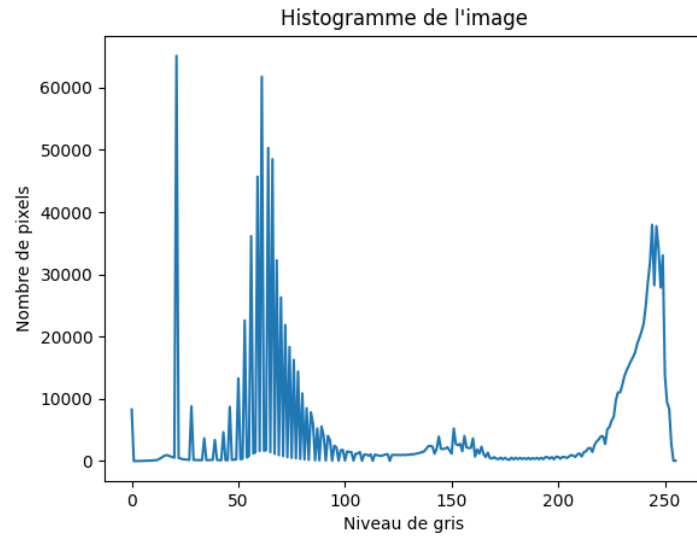


Figure 6 Histogramme de l'image

6.5 Luminance :

La luminance se définit comme le degré de luminosité de chaque point, en d'autres termes elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface, pour une image optimale la luminance doit répondre à trois exigences fondamentales :

- Un contraste équilibré : le contraste est crucial pour une image dynamique et captivante, de trouver un juste équilibre entre les zones claires et les zones sombres, sans tomber dans les extrêmes du blanc ou du noir qui peuvent estomper les détails.
- La luminosité : la lumière donne vie à l'image et permet de distinguer les détails et les textures.
- Absence de parasites : le bruit visuel, tel que le bruit de luminance ou de chrominance

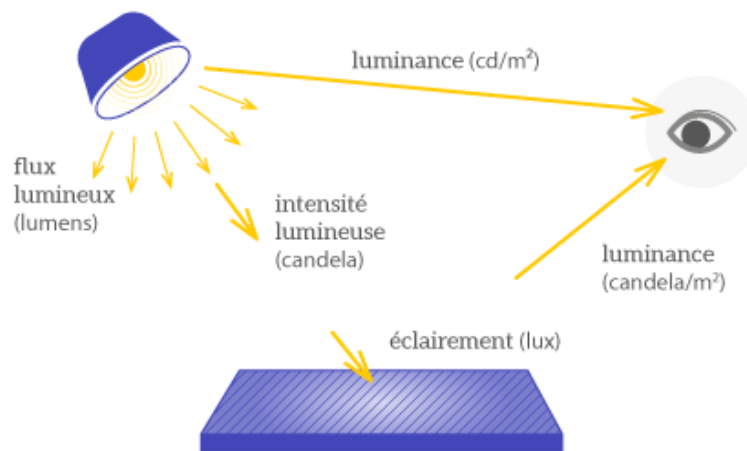


Figure 7 Luminance

6.6 Contraste :

Le contraste est la différence de luminosité entre les zones sombres et claires de l'image il caractérise la répartition lumineuse d'une image, il lui donne une profondeur et un dynamisme.

- Un contraste élevé se caractérise par des blancs éclatants et des noirs profonds, ce qui permet de créer une image saisissante et expressive.
- Un contraste faible, en revanche, se traduit par une image plus douce et moins définie, avec des tons plus proches les uns des autres.

Le contraste est un élément essentiel de la composition d'une image, il permet de mettre en valeur certains éléments, de créer une atmosphère particulière et de guider le regard du spectateur vers des points importants de l'image.



Image avec contraste

image originale

Figure 8 Contraste

6.7 Seuillage :

Le seuillage d'image est une technique fondamentale de segmentation d'image. En partant d'une image en niveaux de gris, cette méthode permet de créer une image où chaque pixel est représenté par une valeur de luminosité soit noir soit blanc. Cela offre une segmentation binaire de l'image, où les pixels sont catégorisés en fonction de leur intensité lumineuse, facilitant ainsi l'identification de régions d'intérêt.



Figure 9 Seuillage

7 Colorimétries de représentation des images :

La représentation colorimétrique des images fait référence à la manière dont les couleurs sont représentées et interprétées dans un système donné. Il existe plusieurs modèles colorimétriques utilisés dans le domaine de l'imagerie pour décrire et manipuler les couleurs. Voici quelques-uns des modèles colorimétriques les plus couramment utilisés :

8 Mode couleur RVB :

Les images numériques sont généralement élaborées dans l'espace colorimétrique RVB (Rouge, Vert et Bleu). Pour les visualiser sur différents types d'écrans, le mode de couleur RVB est privilégié. En combinant le rouge, le vert et le bleu, et en ajustant leurs intensités respectives, une source lumineuse à l'intérieur d'un appareil peut générer une gamme variée de teintes.

Le blanc pur est obtenu lorsque les lumières rouges, vertes et bleues sont mélangées en quantité égale. La saturation, la vivacité et les nuances peuvent être ajustées en modifiant simplement l'intensité de l'une des trois couleurs sources. Étant donné que ces ajustements sont effectués numériquement, les concepteurs ont la possibilité de manipuler la façon dont la lumière est affichée sur l'écran afin d'obtenir la teinte désirée.

8.1 Mode couleur CMJN :

L'espace colorimétrique CMJN (Cyan, Magenta, Jaune, Noir) est utilisé dans l'impression de documents. Contrairement au RVB qui est utilisé pour les écrans, les imprimantes combinent différentes quantités d'encre physique de couleur cyan, magenta, jaune et noir pour créer des images. Ce processus est appelé mélange soustractif, car chaque couche d'encre réduit la luminosité initiale pour former la couleur finale.

Lors de l'impression, toutes les couleurs commencent par un blanc de base, puis chaque couche d'encre ajoute de la couleur en soustrayant la luminosité. Lorsque toutes les couleurs sont superposées, elles créent un noir pur.

Il s'agit d'un processus différent de celui des écrans, où la lumière est émise pour créer des couleurs. Dans le cas de l'impression, la couleur est créée en absorbant la lumière incidente.

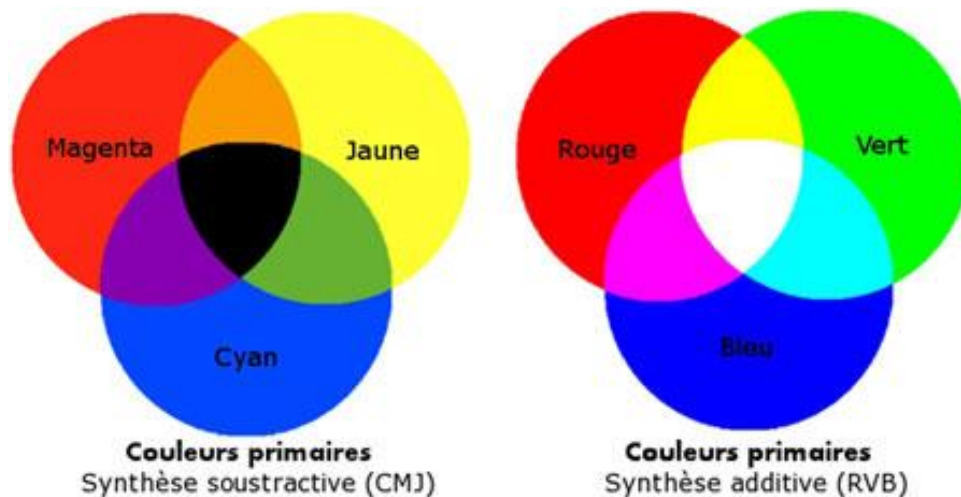


Figure 10 Mode de couleur CMJN et RVB

8.2 HSV (Teinte, Saturation, Valeur) :

Contrairement aux modèles RVB et CMJN, HSV est basé sur la perception humaine de la couleur. La teinte représente la couleur elle-même, la saturation représente l'intensité de la couleur et la valeur représente la luminosité de la couleur. Ce modèle est souvent utilisé dans les logiciels de conception graphique pour manipuler les couleurs de manière intuitive.

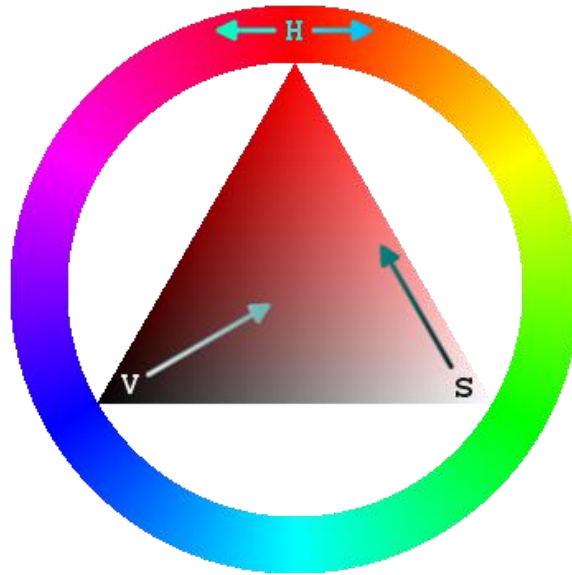


Figure 11 HSV

8.3 LAB (Luminosité, a, b) :

Le modèle LAB est basé sur la perception de la couleur par l'œil humain. Il divise la couleur en trois composantes : luminosité (L), la composante a qui représente le rouge-vert, et la composante b qui représente le bleu-jaune. LAB est souvent utilisé dans la gestion des couleurs et les applications où une précision colorimétrique est nécessaire.

9 Traitement d'image :

Le traitement d'images représente un domaine crucial de l'informatique et des mathématiques appliquées, axé sur l'analyse et la transformation d'images numériques afin d'améliorer leur qualité ou d'en extraire des informations pertinentes.

Il constitue une sous-discipline du traitement du signal, se concentrant spécifiquement sur les images et les données associées, telles que la vidéo, par opposition à d'autres types de signaux, comme le son. Contrairement aux méthodes analogiques telles que la photographie traditionnelle, le traitement d'images se déroule entièrement dans le domaine numérique.

Dans le contexte de la vision artificielle, le traitement d'images intervient après les phases d'acquisition et de conversion en données numériques. Il englobe diverses manipulations et analyses appliquées aux images en vue de les comprendre et de les interpréter. Cette phase d'interprétation est de plus en plus enrichie par l'utilisation de l'intelligence artificielle, exploitant ainsi les connaissances sur le contenu des images traitées.

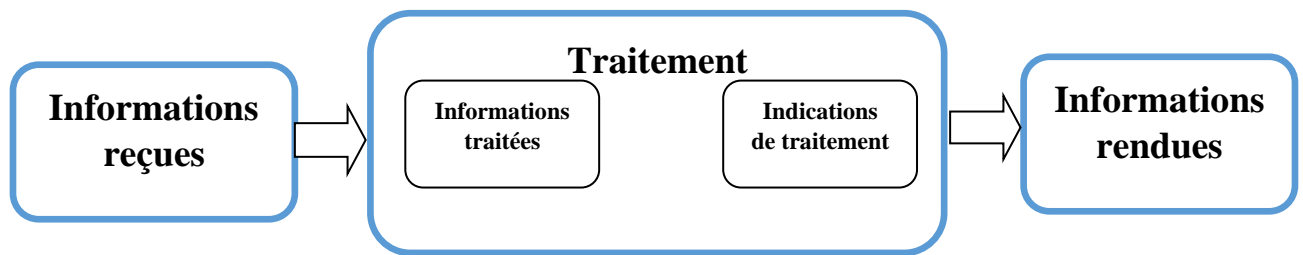


Figure 12 Schéma Synoptique du traitement d'image

10 Acquisition d'une image :

L'acquisition joue un rôle fondamental dans tout le processus de création et de production d'image, avant toute manipulation d'image sur un système informatique, il est impératif de la transformer pour la rendre lisible. Cette transformation de l'image externe originale vers sa représentation interne, adaptée au traitement informatique, est réalisée à travers une procédure de numérisation. Ces dispositifs de capture, connus sous le nom d'appareils optiques, peuvent être regroupés en deux catégories principales : les caméras numériques et les scanners.

11 Prétraitements d'images :

Cette phase a lieu juste après l'acquisition des images, les prétraitements prépare l'image où elle est nettoyée et optimisée pour son analyse ultérieure, on peut corriger des imperfections, améliorer la luminosité et le contraste, ou encore supprimer du bruit indésirable, c'est donc d'obtenir une image théorique en absence de toute dégradation, ainsi ils peuvent par exemple corriger :

- Les défauts radiométriques du capteur
- Les défauts géométriques de l'image dus au mode d'échantillonnage spatial, à l'obliquité de la direction de visée, au déplacement de la cible, etc.
- Le filtrage ou réduction de fréquences parasites
- Les dégradations de l'image dues à la présence de matière entre le capteur et le milieu observé.

11.1 Amélioration d'image :

L'objectif de l'amélioration d'image est de rendre les images plus visibles en éliminant ou réduisant le bruit et en mettant en évidence certains éléments tels que les contours.[2] Cette procédure est souvent effectuée sans pré-connaissance des éléments spécifiques de l'image. Les principales méthodes utilisées incluent :

- L'amélioration de contraste
- Le filtrage linéaire, qui lisse ou met en évidence les contours en utilisant des opérateurs tels que "Image - Image lissée", ainsi que l'utilisation de la transformée de Fourier pour manipuler les fréquences de l'image, les rendant visibles ou les atténuant selon les besoins.
- Filtrage non linéaire (filtres médians, ... etc.)

11.1.1 La binarisation :

Dans de nombreux algorithmes de traitement d'images, en particulier ceux utilisés pour des tâches telles que la reconnaissance d'objets, la segmentation d'images, ou la détection de contours, il est souvent nécessaire de convertir des images en niveaux de gris en images binaires. Cette conversion, appelée binarisation ou seuillage, est une étape cruciale.

Le processus de binarisation commence généralement par le calcul d'un seuil. Ensuite, chaque pixel de l'image est comparé à ce seuil : s'il a une intensité supérieure au seuil, il est étiqueté comme faisant partie du premier plan (c'est-à-dire un objet d'intérêt), sinon il est étiqueté comme faisant partie de l'arrière-plan.

La binarisation est souvent l'une des premières étapes dans les algorithmes de traitement d'images car elle détermine la manière dont les objets d'intérêt seront détectés et analysés par la suite. Par conséquent, la qualité du résultat de la binarisation influence grandement la performance de l'ensemble du système de traitement d'images. Une binarisation précise et robuste est essentielle pour obtenir des résultats précis dans les tâches de traitement d'images.

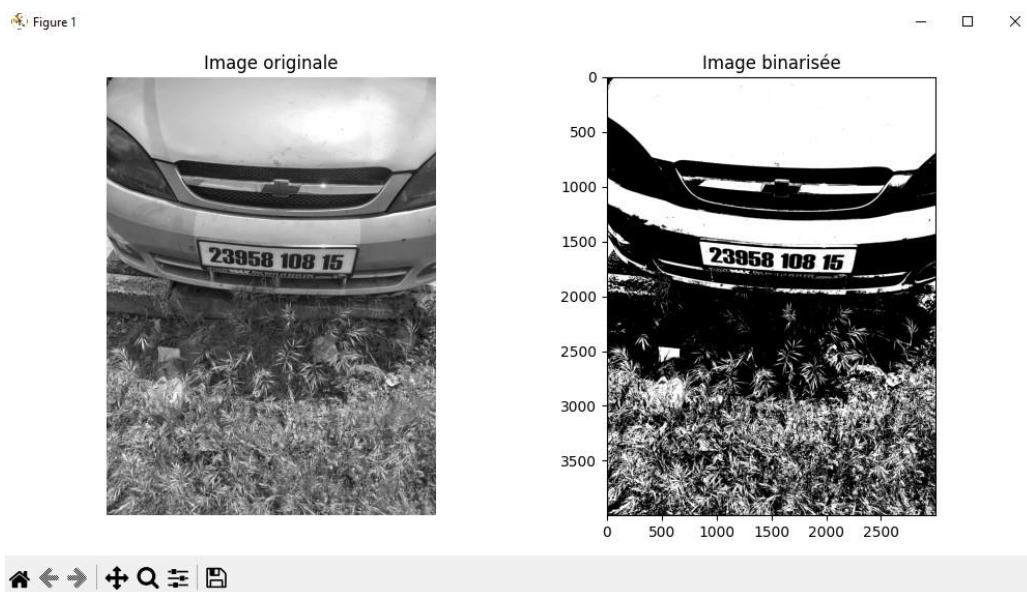


Figure 13 Binarisation d'image

11.1.2 Lissage :

Ces filtres sont des filtres passe-bas qui agissent en atténuant les hautes fréquences du signal d'image. Leur utilisation principale réside dans la réduction du bruit qui peut provenir de diverses sources et qui peut altérer les informations essentielles de l'image, notamment dans le contexte de la détection de contours.

D'un point de vue technique, ces filtres sont des approximations discrètes de filtres continus. Comme leurs homologues continus, ils ne modifient pas le niveau global du signal. En d'autres termes, la somme des coefficients de la matrice de convolution est généralement constante et égale à un, car ces coefficients sont des entiers à diviser par leur somme.



Figure 14 Lissage d'image

11.1.3 La normalisation :

Dans le processus de reconnaissance des plaques d'immatriculation, l'une des difficultés majeures réside dans la variabilité de la taille des images, ce qui peut entraîner des problèmes lors de la segmentation. Cette segmentation est essentielle pour extraire les caractères de la plaque. Ainsi, une étape de normalisation est souvent nécessaire pour standardiser la taille des images et garantir une segmentation précise et fiable.[4]

Dans un système de reconnaissance de chiffres, cette variabilité de taille peut être encore plus prononcée en raison de la diversité des styles de caractères. Par conséquent, la

normalisation devient cruciale pour garantir que les caractères segmentés ont des tailles cohérentes, ce qui facilite leur traitement et leur reconnaissance ultérieure.



Figure 15 La normalisation d'image

11.1.4 Détection de contours :

La détection de contours représente l'une des techniques essentielles dans le traitement d'images numériques. Son objectif est de repérer les endroits sur l'image où l'intensité lumineuse change brutalement, indiquant des éléments importants de structure dans l'objet représenté. Les contours sont souvent utilisés pour faciliter la reconnaissance de formes et l'identification d'objets dans une image.

Dans le contexte du traitement d'images, la détection de contours peut être réalisée à l'aide de divers filtres et algorithmes, chacun ayant ses propres avantages et inconvénients :

- a) Filtre de Sobel : Ce filtre utilise des opérateurs de gradient pour détecter les contours dans une image. Il calcule les gradients horizontaux et verticaux de l'image et combine ces informations pour identifier les changements d'intensité lumineuse.



Figure 16 Contours d'image avec le filtre Sobel

- b) Filtre de Prewitt : Similaire au filtre de Sobel, le filtre de Prewitt est utilisé pour détecter les contours en calculant les gradients de l'image le long des axes horizontal et vertical.



Figure 17 Contours avec le filtre de Prewitt

- c) Filtre de Roberts : Ce filtre utilise des masques de convolution simples pour détecter les contours en mettant en évidence les variations d'intensité entre les pixels adjacents.

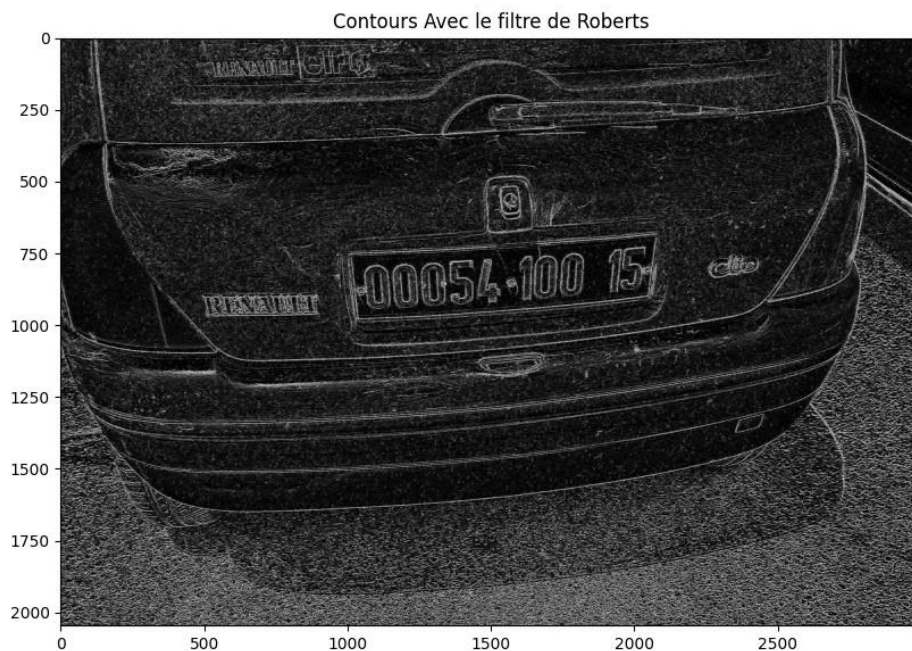


Figure 18 Contours avec le filtre de Roberts

- d) Filtre de Laplace : Aussi connu sous le nom de filtre de détection de bord de Laplace, ce filtre utilise la deuxième dérivée de l'image pour mettre en évidence les changements rapides d'intensité, indiquant la présence de contours



Figure 19 Contours avec le filtre Laplace

- e) Filtre de Canny : Ce filtre combine plusieurs étapes, y compris la suppression du bruit, la recherche des gradients d'intensité, la suppression des pixels non-maxima et la détection des contours par seuillage pour produire des contours nets et précis.



Figure 20 Contours avec le filtre Canny

11.1.5 Segmentation :

Son objectif est de diviser une image en différentes régions présentant des caractéristiques internes similaires, que ce soit en termes statistiques, géométriques ou de forme. Cette division permet ensuite de traiter et d'analyser chaque région de manière indépendante, en fonction des politiques ou des objectifs de traitement définis. Ces régions sont généralement plus petites que l'image dans son ensemble, mais plus grandes que les entités ponctuelles locales, ce qui permet à une application d'utiliser des mesures régionales et locales sur des points d'intérêt comme signature d'objet.

Pour réaliser cette segmentation, plusieurs méthodes sont utilisées, parmi lesquelles on retrouve :

- La segmentation morphologique.
- La segmentation basée sur la couleur.
- La segmentation par seuillage.
- La segmentation par détection de contours.

La détection de contour facilite la reconnaissance de formes et l'identification d'objets



Figure 21 Segmentation par détection de contours

11.2 Analyse d'image :

L'objectif de l'analyse d'image est d'évaluer et d'extraire diverse caractéristique de l'image à traiter pour faciliter son interprétation ultérieure, Ces caractéristiques peuvent être des données statistiques sur des comptes numériques (moyenne, histogramme, etc....), ainsi que des mesures dérivées telles que les dimensions ou les orientations des objets présents dans l'image

Le choix des informations à rechercher dépend généralement du niveau de compréhension nécessaire pour interpréter l'image, ce qui peut varier selon le domaine d'application, dans certains contextes, comme la détection de défauts dans des matériaux, la décision peut être binaire (défectueux ou non défectueux) et peut reposer sur des critères spécifiques tels que la détection de caractéristiques d'absorption particulières. En revanche, pour des applications telles que la navigation robotique, le processus de décision doit simuler le raisonnement humain en mouvement, ce qui nécessite la construction en temps réel d'une carte de l'environnement avec les obstacles à éviter.

Discussion :

Dans ce chapitre nous avons pu avoir une vue générale sur l'image numérique ainsi que les bases du traitement et analyse d'image. On a abordé les caractéristiques principales des images numériques et quelques techniques utilisées dans la vision artificielle.

Dans le chapitre suivant, nous allons voir certaines techniques utilisées pour la détection d'objet et reconnaissance de caractères.

Chapitre 2 :

Etat de l'art sur les systèmes RAPI

Préambule :

Le domaine de la Reconnaissance Automatique de Plaque d'Immatriculation (RAPI), est en expansion et de nouvelles techniques sont utilisées, pour comprendre ces différentes techniques nous devons d'abord voir de quoi se compose une plaque d'immatriculation, dans notre cas nous allons étudier la plaque d'immatriculation Algérienne. L'utilisation des systèmes RAPI sont divers, on peut aller d'une simple utilisation pour usage personnel à une utilisation dans différents domaine et secteur d'activités (les parkings, les péages...), nous allons dans ce chapitre citer quelques domaines d'application des RAPI, ensuite nous allons non seulement voir les différentes techniques de détection mais aussi la reconnaissance des chiffres et lettres à partir de la plaque.

1 Définition de la plaque d'immatriculation :

La plaque d'immatriculation, un dispositif de signalisation routière fixé à un véhicule, joue un rôle crucial en arborant un numéro d'identification unique attribué par l'autorité compétente de chaque pays. Typiquement composée d'une plaque en métal ou en plastique, elle porte des caractères alphanumériques permettant l'identification du véhicule.

Ses fonctions principales sont multiples :

- 1- Identification du véhicule : Grâce à son numéro unique, la plaque d'immatriculation permet une identification précise parmi tous les véhicules en circulation, facilitant le suivi en cas d'infractions, d'accidents ou de transactions commerciales.
- 2- Enregistrement et traçabilité : Les détails associés à la plaque, comme le modèle du véhicule et les données du propriétaire, sont consignés dans une base centrale, offrant ainsi un historique complet et garantissant sa légalité.
- 3- Contrôle et sécurité : Sa présence autorise les autorités à vérifier la conformité des véhicules vis-à-vis de la législation routière, tout en simplifiant l'identification des véhicules impliqués dans des incidents routiers, renforçant ainsi la sécurité.
- 4- Prévention des vols : La plaque constitue un rempart contre le vol de véhicules. En cas de vol, les autorités peuvent utiliser son numéro pour retrouver et identifier le véhicule.

Les spécificités des plaques varient selon les pays, incluant le format, la couleur, la taille et le style des caractères. Régies par la législation routière, elles doivent respecter les normes établies par les autorités compétentes.

2 Caractéristiques des plaques algériennes :

Les plaques d'immatriculation algériennes sont généralement blanches à l'avant et jaunes à l'arrière avec des caractères noirs, suivant un format rectangulaire standardisé. Elles comportent des caractères alphanumériques disposés en groupes selon une disposition réglementée, facilitant ainsi leur reconnaissance automatique. Certaines plaques peuvent présenter des motifs spécifiques, tels que le logo national, et peuvent contenir des indications régionales

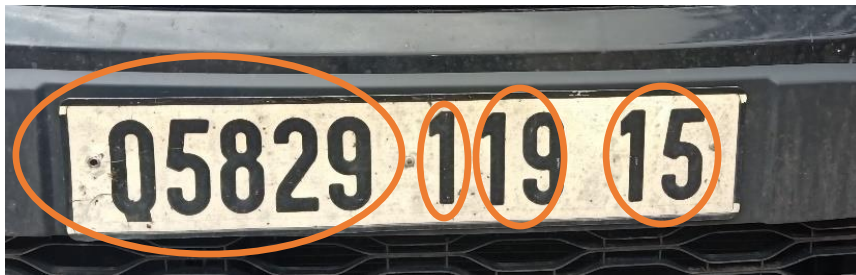


Figure 22 Caractéristiques des plaques algériennes

Nous commençons par la droite :

15 : correspond au code de la wilaya.

19 : correspond à l'année de fabrication du véhicule.

1 : correspond au type de l'automobile. Ici c'est 1 car c'est une voiture touristique, 2 pour les camions, etc.

5829 : signifie que ce véhicule est le 5829^{ème} qui est enregistré dans la wilaya 15

3 Etat de l'art des méthodes RAPI :

La reconnaissance automatique des plaques d'immatriculations (RAPI), est en constante évolution, avec les progrès rapides dans le domaine de la vision par ordinateur et de l'apprentissage automatique.

3.1 Méthodes basées sur le traitement d'images :

Une image est capturée par une caméra, puis on extrait les caractéristiques visuelles de la plaque d'immatriculation en utilisant des techniques de traitement d'images pour segmenter, identifier et reconnaître la plaque, cela peut inclure des algorithmes de seuillage, de détection de contours, de reconnaissance de formes et de classification de caractères.

3.2 Méthodes basées sur l'apprentissage profond :

Les réseaux de neurones profonds ont gagné en popularité pour la reconnaissance des plaques d'immatriculation. Les réseaux de neurones convolutifs (CNN) sont souvent utilisés pour extraire automatiquement des caractéristiques pertinentes à partir des images et effectuer la reconnaissance de caractères. Ces méthodes peuvent être entraînées sur de grandes quantités de données pour améliorer leur précision

3.3 Méthodes hybrides :

Certaines approches combinent des techniques de traitement d'images traditionnelles avec des méthodes d'apprentissage profond pour obtenir de meilleures performances de reconnaissance. Par exemple, un réseau de neurones profond peut être utilisé pour détecter les régions d'intérêt dans une image, puis des techniques de traitement d'images plus classiques peuvent être appliquées pour extraire les caractéristiques de ces régions et reconnaître les caractères.

3.4 Utilisation de bases de données et de techniques de recherche :

Les systèmes RAPI utilisent souvent des bases de données de plaques d'immatriculation pour comparer les plaques détectées avec les enregistrements existants. Des techniques de recherche efficaces telles que la recherche par indexation et la recherche approximative peuvent être utilisées pour accélérer le processus de correspondance.

3.5 Optimisation et traitement en temps réel :

Pour être utilisables dans des applications en temps réel, les méthodes RAPI doivent être optimisées pour la vitesse de traitement et la consommation de ressources. Cela peut impliquer des techniques d'optimisation spécifiques aux plateformes matérielles, telles que l'utilisation de processeurs graphiques (GPU) ou de processeurs spécialisés dans le traitement d'images.

4 Difficulté des systèmes de reconnaissance des plaques d'immatriculation :

4.1 Variabilité des plaques :

Les plaques d'immatriculation peuvent varier en taille, forme, couleur, police de caractères, disposition des caractères, etc. Cette variabilité peut rendre la détection et la reconnaissance des plaques plus complexes.

4.2 Conditions environnementales :

- Des images floues : Elles sont souvent causées par le mouvement qui est très fréquent dans des installations mobiles.
- Un mauvais éclairage et un faible contraste à cause d'une surexposition, d'un reflet, ou d'ombres.
- Angle de vue et distance : Les plaques d'immatriculation peuvent être capturées sous différents angles de vue et à différentes distances par les caméras, ce qui peut rendre la détection et la reconnaissance des plaques plus complexes.
- Un objet obscurcissant une partie de la plaque, souvent une barre de remorquage, ou de la poussière sur la plaque.

5 Les techniques de Reconnaissance des plaques d'immatriculation :

5.1 Technique basée sur la couleur de la plaque :

La reconnaissance de plaques d'immatriculation basée sur la couleur de la plaque repose sur l'exploitation des caractéristiques chromatiques de la plaque pour la détecter et la segmenter dans une image. Voici les étapes générales d'un tel processus :

- Prétraitement de l'image : L'image capturée est prétraitée pour améliorer la qualité et réduire le bruit. Cela peut inclure des opérations telles que la normalisation de l'exposition, l'égalisation de l'histogramme, et la réduction du bruit.
- Détection des régions d'intérêt : Les régions de l'image susceptibles de contenir une plaque d'immatriculation sont détectées. Dans le cas de la reconnaissance basée sur la couleur, cela peut impliquer la segmentation des parties de l'image qui correspondent à la couleur typique des plaques d'immatriculation.
- Filtrage par couleur : Les pixels de l'image sont filtrés en fonction de leur couleur pour isoler les régions qui correspondent à la couleur attendue d'une plaque d'immatriculation. Cela peut être réalisé en utilisant des techniques de seuillage ou des modèles de couleur prédéfinis.
- Analyse des régions candidates : Les régions identifiées comme potentielles plaques d'immatriculation sont analysées pour vérifier leur forme, leur taille et leur disposition. Des critères tels que la forme rectangulaire et le rapport largeur/hauteur peuvent être utilisés pour filtrer les faux positifs.

La reconnaissance de plaques d'immatriculation basée sur la couleur peut être efficace dans des environnements où les plaques d'immatriculation présentent des caractéristiques chromatiques distinctes et où le contraste entre la plaque et l'arrière-plan est élevé. Cependant, elle peut être sensible aux variations d'éclairage et aux conditions environnementales changeantes. Des techniques avancées de traitement d'images et d'apprentissage automatique peuvent être utilisées pour améliorer la robustesse et la précision du système de détection.



Figure 23 Technique RAPI basée sur la couleur de la plaque

5.2 Technique basée sur la taille des caractères et leur distance les uns des autres :

La technique de détection de plaque d'immatriculation basée sur la taille des caractères et leur distance les uns des autres utilise les caractéristiques géométriques des caractères de la plaque pour détecter la plaque d'immatriculation dans une image.

Cette approche est efficace dans les cas où les caractères de la plaque sont relativement bien définis et où la taille et la disposition des caractères présentent une certaine régularité. Elle peut être robuste aux variations d'éclairage et aux conditions environnementales changeantes, mais peut nécessiter une calibration minutieuse et des techniques avancées de traitement d'images pour obtenir des résultats précis.

5.3 Technique basée sur les contours de la plaque :

Cette méthode s'appuie sur les contours des caractères et des plaques d'immatriculation comme repères pour l'extraction. Elle exploite le contraste important entre l'intensité des pixels composant ces contours et celle des pixels environnants.

Cependant, cette technique montre ses limites lorsque les bords de la plaque présentent une faible variation d'intensité par rapport au reste de l'image.



Figure 24 Technique RAPI basée sur les contours

5.4 Technique basée sur la texture de la plaque :

La méthode basée sur la texture de la plaque d'immatriculation est une approche sophistiquée qui exploite les caractéristiques texturales uniques présentes sur les plaques pour leur identification. Voici en détail comment cette méthode fonctionne :

- **Prétraitement de l'image :** L'image acquise sera soumise à ce processus afin d'améliorer sa qualité et de réduire le bruit en utilisant différentes opérations telles que la normalisation de l'exposition, la réduction de bruit.
- **Extraction de descripteurs de texture :** Ces descripteurs sont des caractéristiques spécifiques qui capturent les informations sur la structure et la distribution des motifs texturaux présent dans l'image. Ils peuvent inclure des mesures de la rugosité.
- **Entraînement d'un modèle de classification :** Les descripteurs extraits sont utilisés afin d'entraîner le modèle ce qui permet au système d'apprendre à reconnaître les caractéristiques distinctives des plaques d'immatriculation parmi les autres objets.
- **Détection et reconnaissance de plaques :** A la fin de l'entraînement du modèle, ce dernier est utilisé pour détecter et reconnaître les plaques dans d'autres images et cela se fait en recherchant des régions dans la nouvelle image qui présentent les mêmes caractéristiques à celles apprises.
- **Validation et post-traitement :** Une fois la reconnaissance de la plaque effectuée, celle-ci est soumise à une vérification de la cohérence des caractéristiques extraites, la correction des erreurs et la fusion des régions adjacentes.



Figure 25 Technique RAPI basée sur la texture de la plaque

5.5 Technique basée sur la heatmap (Carte de chaleur) :

Dans le cadre de la détection de plaques d'immatriculation, une heatmap est souvent utilisée pour visualiser les parties de l'image qui sont les plus probables de contenir une plaque d'immatriculation. Ces heatmaps sont générées à partir des prédictions d'un modèle de détection, généralement un réseau de neurones convolutionnel (CNN). Le modèle attribue des scores à chaque région de l'image en fonction de sa probabilité de contenir une plaque d'immatriculation.

L'interprétation d'une heatmap de détection de plaques d'immatriculation est la suivante :



Figure 26 Technique basée sur le Heatmap

Les régions de l'image avec des valeurs élevées sur la heatmap sont celles où le modèle estime qu'une plaque d'immatriculation est plus susceptible d'être présente.

Les valeurs basses ou nulles sur la heatmap représentent les régions où le modèle estime qu'il est moins probable de trouver une plaque d'immatriculation.

En analysant visuellement la heatmap, on peut repérer les zones les plus "chaudes" de l'image, c'est-à-dire les régions où le modèle estime que les plaques d'immatriculation sont les plus susceptibles d'être situées. Cela peut être utile pour orienter le processus de localisation et de détection des plaques d'immatriculation, en concentrant l'attention sur les régions les plus prometteuses pour une inspection plus approfondie.

5.6 Technique basée sur l'infrarouge :

La détection de plaques d'immatriculation par infrarouge repose sur l'utilisation de capteurs infrarouges pour repérer les changements de température émis par les véhicules, incluant leurs plaques d'immatriculation. Voici son fonctionnement général :

- Capteurs infrarouges : Placés stratégiquement, ces capteurs détectent les variations de chaleur provenant des objets dans leur champ de vision, comme le long des routes ou aux intersections.
- Détection des véhicules : Lorsqu'un véhicule passe près des capteurs, ces derniers perçoivent les fluctuations de température associées à sa présence, étant donné que les moteurs et composants mécaniques produisent de la chaleur en fonctionnant.
- Localisation des plaques d'immatriculation : Les capteurs peuvent être positionnés pour détecter spécifiquement la chaleur émise par les plaques d'immatriculation. En fonction de leur emplacement et de leur configuration, il est possible d'identifier approximativement la position de la plaque sur le véhicule.
- Traitement des données : Les données infrarouges collectées sont analysées à l'aide d'algorithmes de traitement du signal pour extraire les informations clés, telles que la localisation et la taille de la plaque détectée.

6 Techniques de reconnaissances de caractères et de chiffres :

De nombreuses techniques sont utilisés pour la reconnaissance de caractères et de chiffres, on peut distinguer deux techniques principales :

6.1 Technique de reconnaissance par détection de contours :

C'est une méthode utilisée pour identifier et interpréter des caractères alphanumériques dans une image, voici les étapes essentielles de cette méthode :

- Prétraitement d'image : pour améliorer la qualité de l'image avant d'extraction de contours, des filtres doivent être appliqués tels que la conversion en niveaux de gris, réduction de bruit...
- Détection de contours : une fois que les filtres sont appliqués, on applique un détecteur de contours de Canny ou Sobel afin d'identifier les contours des caractères dans l'image.
- Segmentation : une fois les contours détectés, on applique une segmentation afin de déviser individuellement chaque caractère.
- Extraction

6.2 Techniques de reconnaissance par intelligence artificiel :

6.2.1 Reconnaissance optique des caractères OCR :

La reconnaissance optique de caractères, transforme divers types de documents textuels, comme des fichiers PDF, des images [7]. C'est la même chose que fait l'humain en lisant le texte sur un écran ou sur du papier avec l'OCR les ordinateurs peuvent comprendre et interprètent du contenu textuel. Grâce à des techniques de traitement d'image avancées, l'OCR analyse les formes et les structures des caractères pour ensuite les convertir en texte électronique, cette technique est utilisée dans divers domaines tels que la numérisation ou encore la reconnaissance de plaque d'immatriculations et bien d'autres domaines.

6.2.2 Etape de l'OCR :

- Prétraitement de l'image :
 - Conversion en niveaux de gris.
 - Binarisation.
 - Filtrage de bruit.
- Détection et segmentation des caractères :
 - Détection de contours.
 - Segmentation de caractères.
- Reconnaissance des caractères :
 - Utilisation de modèles de reconnaissance de caractères : Des modèles déjà entraînés ou des architectures de réseaux neuronaux convolutionnel (CNN) spécialement

développés pour la reconnaissance des chiffres et des caractères peuvent être exploités afin de détecter et d'identifier chaque caractère présent sur la plaque d'immatriculation.[9]

- Post-traitement : Les résultats obtenus grâce à l'OCR peuvent être soumis à un processus de post-traitement visant à rectifier les erreurs de reconnaissance et à accroître la précision globale. Ce post-traitement peut impliquer la vérification de la conformité aux normes de format des plaques d'immatriculation, la correction orthographique, ainsi que la fusion des caractères adjacents pour former des groupes de caractères cohérents.

- Extraction des propriétés de l'image :

Fondamentalement, il existe deux approches principales pour extraire les caractéristiques statistiques d'une image en OCR :

- L'algorithme de détection analyse les lignes et les traits pour définir un caractère, évaluant ainsi ses composantes individuelles.
- La reconnaissance des motifs fonctionne en identifiant l'intégralité du caractère, en analysant sa forme globale.[8]

Pour reconnaître une ligne de texte, nous recherchons des rangées de pixels blancs séparées par des pixels noirs. De manière similaire, pour déterminer les limites d'un caractère, nous identifions où il commence et où il se termine en analysant les variations de couleur et de luminosité dans l'image.

7 Résumé des Techniques utilisées :

7.1 Techniques de détection :

Technique	Avantages	Inconvénients
Basée sur la couleur de la plaque	Efficace lorsque les plaques ont des caractéristiques chromatiques distinctes.	Sensible aux variations d'éclairage et aux conditions environnementales changeantes.

Basée sur la taille des caractères et leur distance	Robuste aux variations d'éclairage et aux conditions environnementales changeantes.	Nécessite une calibration minutieuse et des techniques avancées de traitement d'images pour obtenir des résultats précis.
Basée sur les contours de la plaque	Utilise les contours des caractères et des plaques d'immatriculation comme repères pour l'extraction.	Limitée lorsque les bords de la plaque présentent une faible variation d'intensité par rapport au reste de l'image.
Basée sur la texture de la plaque	Identifie les caractéristiques texturales uniques des plaques pour une identification fiable et précise.	Nécessite des données d'entraînement importantes et des techniques avancées d'apprentissage automatique.
Basée sur la heatmap (Carte de chaleur)	Visualise les parties de l'image les plus susceptibles de contenir une plaque d'immatriculation.	Dépendante de la qualité des prédictions du modèle de détection.
Basée sur l'infrarouge	Fonctionne efficacement en conditions de faible luminosité ou de visibilité réduite.	Dépendante de la calibration et de la configuration des capteurs infrarouges
Basée sur l'intelligence artificielle	Utilise des réseaux neuronaux pour détecter et identifier les plaques d'immatriculation.	Efficacité tributaire des conditions d'éclairage et de la qualité des prédictions du modèle neuronal.

Tableau I Comparaison des techniques de détection de la plaque d'immatriculation

7.2 Techniques de reconnaissance :

Technique	Avantages	Inconvénients
Technique de reconnaissance par détection de contours.	Simplicité de mise en œuvre. Bonne performance sur les polices standards. Application en temps réel	Sensibilité au bruit. Limitation avec les polices stylisées. Manque de robustesse aux transformations.
Reconnaissance optique des caractères (OCR)	Automatise la conversion des documents en texte éditable.	Peut commettre des erreurs, en particulier avec des scans de mauvaise qualité ou une écriture manuscrite.

Tableau II Comparaison des techniques de reconnaissance de la plaque d'immatriculation

8 Domaines d'utilisation des systèmes RAPI :

8.1 Gestion du trafic et péage automatique :

Le système de péage urbain repose sur l'analyse des plaques d'immatriculation capturées par un réseau de caméras de surveillance positionnées aux entrées et sorties des zones de péage urbain, telles que les villes et les autoroutes... etc., la reconnaissance automatique de plaque d'immatriculation vise à identifier la catégorie du véhicule et à lire le numéro d'identification de sa plaque d'immatriculation, il confronte ensuite les informations recueillies avec une base de données centralisée pour vérifier la conformité des paiements dans les délais impartis. Ce système contribue à fluidifier et à simplifier la circulation.



Figure 27 RAPI dans la gestion de trafic et de péage

8.2 Gestion des Parkings :

Les caméras équipées de systèmes RAPI sont installées aux entrées et sorties des parkings pour enregistrer automatiquement les plaques d'immatriculation des véhicules entrant et sortant, puis avec un système de gestion il va déduire le temps de stationnement et ensuite déduire le montant à payé.

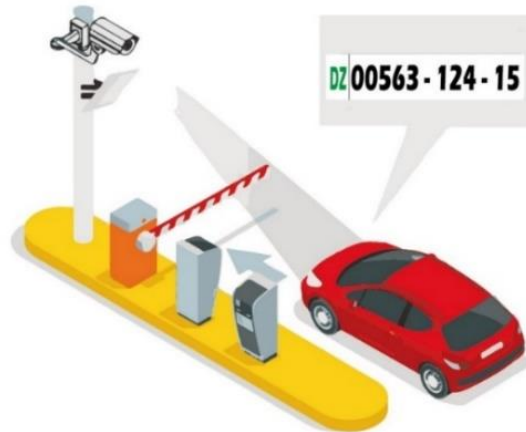


Figure 28 Système RAPI pour la gestion de parking

8.3 Identification des véhicules :

Les systèmes RAPI permettent d'identifier rapidement les véhicules autorisés à entrer dans une résidence privée par exemple, ou dans un lieu où y'a une autorisation d'entrée, en fonction de leur plaque d'immatriculation



Figure 29 Système RAPI pour l'identification de véhicules

Discussion :

La plaque d'immatriculation est une identification unique d'un véhicule, leurs utilisations facilitent de nombreuses applications tels que la sécurité routière, la gestion des parkings et d'autres.

Dans ce chapitre nous avons vu que les RAPI incluent principalement le traitement d'images classique, l'apprentissage profond avec les réseaux de neurones convolutifs, et des approches hybrides combinant ces techniques.

Chacune des techniques fait face à de nombreux défis, comme la variabilité de la taille, la couleur, la forme et la façon dont les caractères sont disposés sur la plaque.

Après avoir étudié chacune des méthodes, nous avons opté dans notre cas pour l'utilisation de l'intelligence artificielle pour la détection et la reconnaissance des plaques, c'est ce que nous allons voir dans le prochain chapitre.

Chapitre 3 :

**Les réseaux de neurones dans les
systèmes RAPI**

Préambule :

L'intelligence artificielle est au cœur de la révolution technologique actuelle, constamment utilisée dans de nombreux domaines qu'elle en devient indispensable. Ce chapitre a pour but de retracer l'histoire de l'intelligence artificielle et ses fondements en mettant en lumière ses applications pratiques dans les différents secteurs. Nous allons nous concentrer plus précisément sur le Deep Learning qui est un sous domaine de l'IA en détaillant son fonctionnement et son architecture interne où nous nous sommes concentrés sur les réseaux neuronaux pour pouvoir les utiliser dans la vision par ordinateur.

1 L'intelligence artificielle :

L'intelligence artificielle est un domaine de l'informatique qui se concentre sur le développement des systèmes et d'algorithmes capables de simuler les processus cognitifs humains et de réaliser des tâches qui nécessitent généralement l'intelligence humaine.

Elle est désignée comme la capacité qu'une machine puisse imiter le comportement lié aux humains, elle vise à reproduire certains aspects de l'intelligence humaine tels que le raisonnement, la planification et la créativité.[5]

L'intelligence artificielle utilise différentes techniques afin de se perfectionner tels que l'apprentissage automatique (machine Learning), l'apprentissage profond (deep Learning).

1.1 Histoire de l'intelligence artificielle :

L'intelligence artificielle (IA) trouve ses racines dans les récits anciens décrivant des machines intelligentes, remontant à des millénaires. Cependant, son véritable essor s'est produit au 20^e siècle. En 1956, lors d'une conférence, le terme "intelligence artificielle" a été formellement introduit par John McCarthy, marquant le début de décennies de recherche et de développement.

Les années 1950 et 1960 ont vu l'émergence des premiers programmes d'IA, tels que le "Logic Theorist" de Newell et Simon, capables de résoudre des problèmes complexes. Cependant, malgré ces avancées, les années 1970 ont été marquées par un scepticisme croissant envers l'IA, entraînant une diminution des financements et un "hiver de l'IA".[11]

Dans les années 1980, un nouvel intérêt pour l'IA a surgi, stimulé par des avancées technologiques et des applications pratiques émergentes comme les systèmes experts et les algorithmes d'apprentissage automatique. Les années 1990 ont été marquées par l'avènement

d'Internet et une augmentation de la puissance de calcul, ouvrant la voie à de nouvelles percées en matière d'IA.

Depuis les années 2000 jusqu'à aujourd'hui, l'IA a connu une croissance exponentielle, alimentée par le Big Data, le cloud computing et l'apprentissage automatique. Des applications pratiques de l'IA se sont généralisées dans divers domaines, transformant la santé, les transports, la finance, et bien d'autres encore. La recherche en IA progresse rapidement, avec des avancées notables dans des domaines tels que l'apprentissage profond, le traitement du langage naturel et la vision par ordinateur. L'histoire de l'IA est complexe mais continue de façonner profondément et significativement le monde moderne.

1.2 Importance de l'intelligence artificielle dans notre ère :

L'intelligence artificielle (IA) est essentielle dans la reconnaissance et le traitement d'images ainsi que dans d'autres secteurs liés à la vision par ordinateur et au traitement du signal. Par exemple, dans la sécurité, des systèmes de surveillance dotés de caméras et d'algorithmes d'IA détectent automatiquement les comportements suspects ou les objets indésirables, renforçant la sécurité dans les espaces publics, les aéroports et autres zones sensibles.

Dans l'industrie automobile, les systèmes d'assistance à la conduite et les véhicules autonomes utilisent des capteurs et des algorithmes d'IA pour repérer les obstacles, interpréter les panneaux de signalisation et prendre des décisions en temps réel pour garantir la sécurité des passagers et des autres usagers de la route.

En outre, l'IA est largement employée dans les applications de réalité augmentée et virtuelle pour créer des expériences immersives et interactives, que ce soit pour le divertissement, la formation professionnelle ou la simulation d'environnements complexes.

Dans le domaine médical, l'IA révolutionne la radiologie en permettant une détection plus précise des anomalies dans les images médicales comme les radiographies et les IRM, ce qui aide les médecins à poser des diagnostics plus précis et à proposer des traitements plus efficaces.

Enfin, dans l'industrie manufacturière, l'IA est utilisée pour optimiser les processus de contrôle qualité en identifiant automatiquement les défauts sur les produits, réduisant ainsi les coûts de production et améliorant la satisfaction client.

Dans tous ces domaines, l'IA offre des capacités d'analyse et de prise de décision comparables voire supérieures à celles des êtres humains, ce qui ouvre de nouvelles

perspectives pour améliorer l'efficacité, la sécurité et la qualité des services et des produits proposés.

2 Les sous-ensembles de l'intelligence artificielle :

2.1 Machine Learning :

Le Machine Learning, une composante de l'intelligence artificielle (IA), vise à enseigner aux machines à assimiler des informations à partir de données et à se perfectionner au fil du temps, plutôt que d'être explicitement programmées pour des tâches spécifiques. Dans ce domaine, les algorithmes sont formés pour détecter des schémas et des relations dans de vastes ensembles de données, leur permettant de prendre des décisions éclairées et de faire des prédictions basées sur leur analyse. Avec la pratique, les capacités des applications de Machine Learning s'améliorent, et leur précision augmente proportionnellement à la quantité de données à leur disposition.

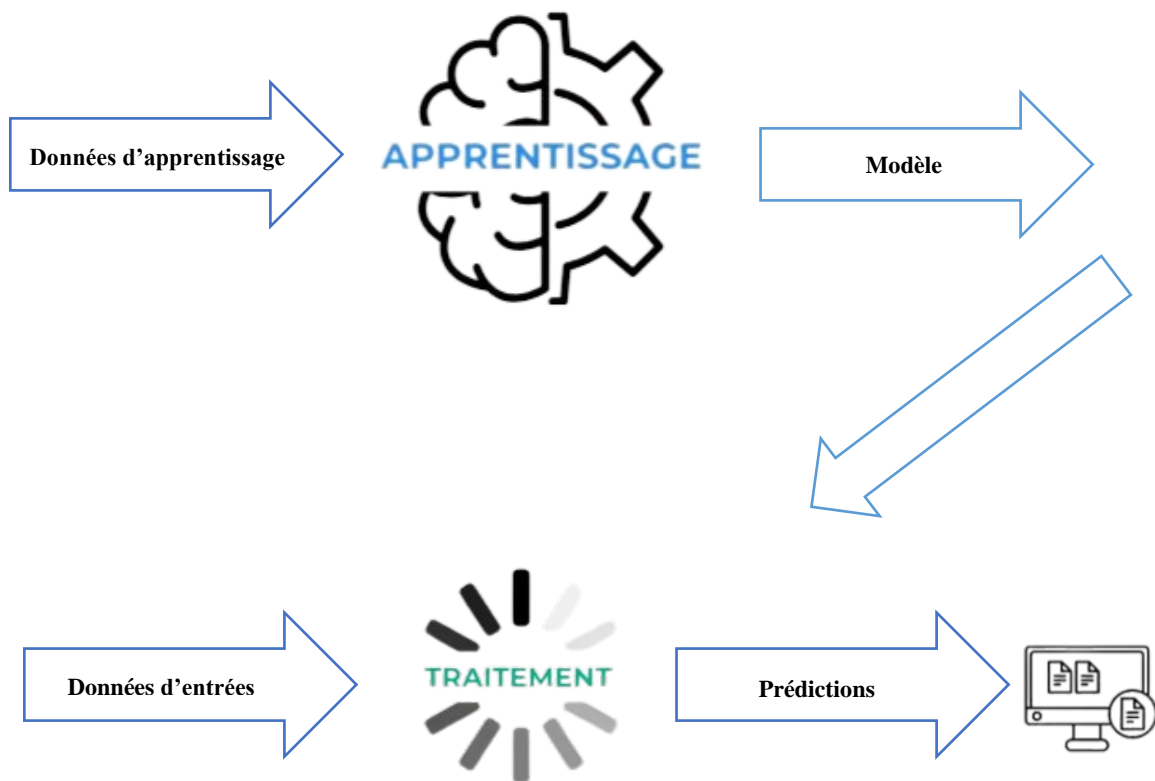


Figure 30 Schéma synoptique du fonctionnement du Machine Learning

2.1.1 Fonctionnement du Machine Learning :

Pour développer un modèle de Machine Learning on doit passer par quatre étapes principales.

1. La première étape consiste à collecter des données pertinentes pour le problème à résoudre, c'est-à-dire de sélectionner et de préparer un ensemble de données d'entraînement. Les données peuvent être catégorisées comme étiquetées, où les caractéristiques à identifier sont spécifiées pour le modèle, ou non étiquetées, où le modèle doit lui-même repérer et extraire les caractéristiques récurrentes. Dans les deux cas, il est crucial de préparer, organiser et nettoyer méticuleusement les données. Sans cela, l'entraînement du modèle de Machine Learning risque d'être altéré, ce qui impactera directement la précision de ses prédictions à venir.[6]
2. La deuxième étape consiste à choisir un modèle d'apprentissage en fonction des données disponibles, il existe différents types de modèles de Machine Learning, tels que les réseaux de neurones, les arbres de décision, les machines à vecteurs de support, etc. Chaque modèle a ses propres forces, faiblesses et applications spécifiques.
3. La troisième étape est d'entraîner le modèle, il s'agit d'un processus itératif où les variables sont exécutées à travers le modèle, et les résultats sont comparés à ceux qu'il aurait dû produire. Ensuite, les "poids" et le biais peuvent être ajustés pour améliorer la précision du résultat.
4. La quatrième et dernière phase implique l'utilisation et l'amélioration du modèle. Ce dernier est appliqué à de nouvelles données, spécifiques au problème à résoudre. Par exemple, un modèle de Machine Learning développé pour la détection de chiffres à partir d'une plaque d'immatriculation sera utilisé sur des plaques d'immatriculations.[10]

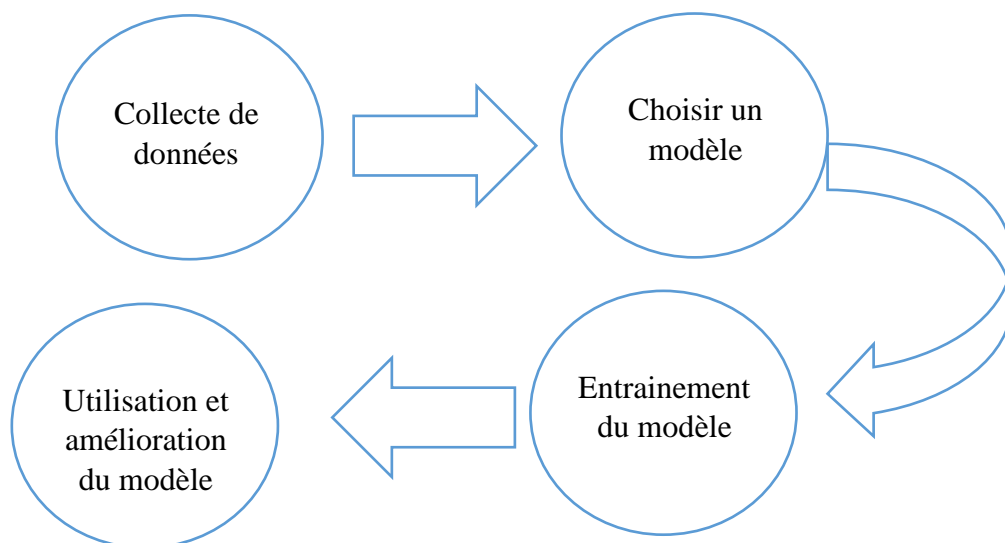


Figure 31 Schéma synoptique du processus de fonctionnement du Machine Learning

2.1.2 Les sous-domaines du Machine Learning :

2.1.2.1 Apprentissage supervisé :

L'apprentissage supervisé constitue une approche fondamentale du machine Learning, impliquant l'entraînement d'un modèle sur un jeu de données annotées [12]. Ces annotations offrent des indications sur les résultats attendus pour chaque instance de données, permettant ainsi au modèle de développer la capacité de faire des prédictions ou de prendre des décisions en se basant sur ces informations.

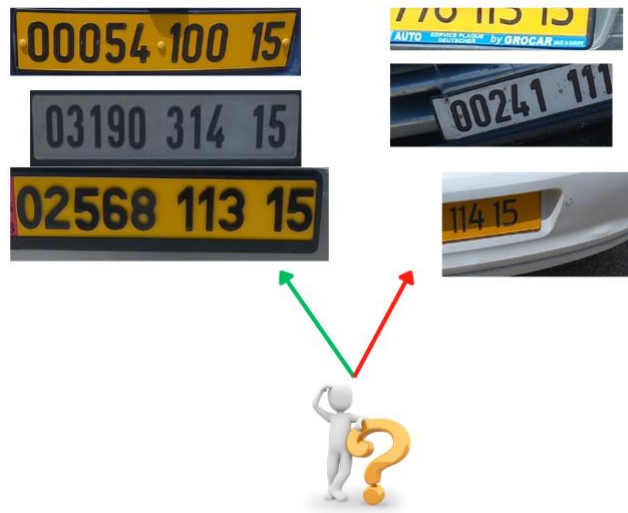


Figure 32 Apprentissage supervisé

Dans l'apprentissage supervisé, le processus d'entraînement se déroule généralement en plusieurs étapes :

- Préparation des données : Les données d'entraînement sont préparées en séparant les caractéristiques (variables explicatives) des étiquettes (résultats attendus). Il est crucial de nettoyer et de prétraiter les données pour garantir leur qualité et leur cohérence.
- Choix du modèle : En fonction de la tâche à réaliser (classification, régression, etc.) et des caractéristiques des données, un modèle adapté est sélectionné. Parmi les options possibles figurent les arbres de décision, les réseaux neuronaux, les machines à vecteurs de support, etc.
- Entraînement du modèle : Le modèle est entraîné sur les données d'entraînement en ajustant ses paramètres pour minimiser une fonction de perte, qui évalue la divergence

entre les prédictions du modèle et les étiquettes réelles. Cette phase utilise généralement des algorithmes d'optimisation comme la descente de gradient.

- **Évaluation du modèle** : Une fois que le modèle est entraîné, il est évalué sur un ensemble de données de test distinct pour estimer sa performance. Cette étape permet de vérifier si le modèle généralise correctement sur de nouvelles données et s'il est capable de fournir des prédictions précises.
- **Optimisation et ajustement des hyper paramètres** : Si nécessaire, les hyper paramètres du modèle (par exemple, le taux d'apprentissage, la taille du réseau, etc.) sont ajustés pour améliorer ses performances. Cela peut impliquer l'utilisation de techniques telles que la recherche d'hyper paramètres ou la validation croisée.
- **Utilisation du modèle** : Une fois que le modèle a été formé et évalué avec succès, il peut être utilisé pour faire des prédictions sur de nouvelles données. Ces prédictions peuvent être appliquées dans divers domaines tels que la classification d'images, la prédiction de prix, la recommandation de produits, etc.

2.1.2.2 Apprentissage non supervisé :

L'apprentissage non supervisé dans le domaine du machine Learning implique l'utilisation d'algorithmes pour traiter des données non étiquetées, sans intervention humaine directe.[15] Contrairement à l'apprentissage supervisé, il n'y a pas de variable cible à prédire. L'objectif principal est plutôt de détecter des structures, des clusters ou des modèles sous-jacents dans les données.



Figure 33 Apprentissage non supervisé

L'apprentissage non supervisé se révèle particulièrement précieux dans les cas où les données non étiquetées abondent, comme dans le domaine médical où l'étiquetage manuel s'avère coûteux. Il facilite des tâches telles que la segmentation, la détection de motifs et la réduction de dimensionnalité.

Néanmoins, les résultats obtenus par l'apprentissage non supervisé peuvent être plus variables que ceux de l'apprentissage supervisé, étant donné que la machine crée elle-même les groupes et les catégories sans l'intervention d'une supervision humaine. Un expert doit donc souvent intervenir pour valider les résultats.

2.2 Réseau neuronal :

Les réseaux de neurones sont des modèles de calcul inspirés par le fonctionnement du cerveau humain, largement utilisés dans l'intelligence artificielle et l'apprentissage automatique. Ils se composent de plusieurs couches de nœuds interconnectés, appelés neurones. Voici une explication des concepts fondamentaux des réseaux de neurones :

2.2.1 Neurone Artificiel :

Un neurone artificiel est une unité de base dans un réseau de neurones. Il reçoit une ou plusieurs entrées, effectue une opération de pondération sur ces entrées, puis applique une fonction d'activation pour produire une sortie. L'équation de base est :

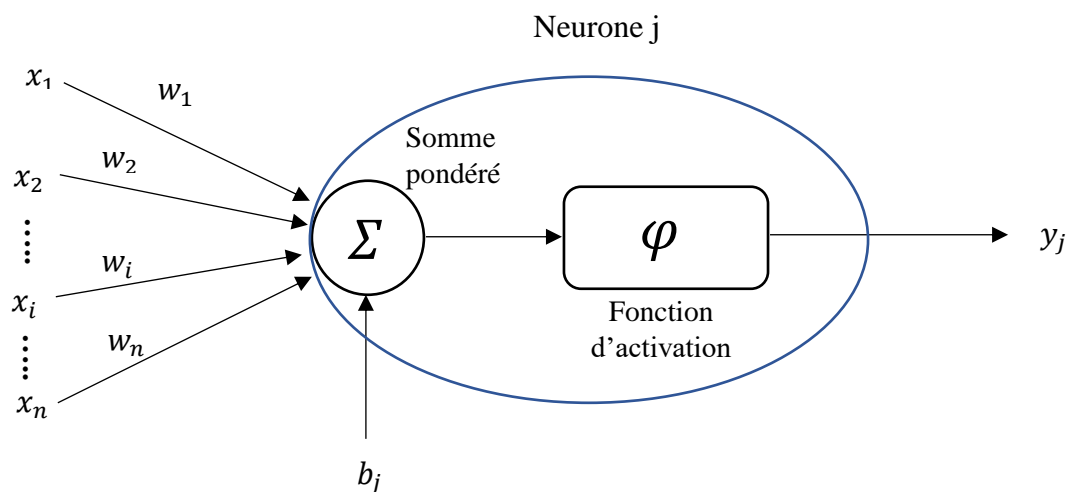


Figure 34 Equation de base de la fonction d'activation de neurone artificiel

$$y = f(\sum_{i=1}^n w_i x_i + b)$$

Où :

- x_i sont les entrées,
- w_i sont les poids associés aux entrées,
- b est le biais,
- f est la fonction d'activation,
- y est la sortie.

2.2.2 Couche de neurones :

Les réseaux de neurones sont généralement organisés en plusieurs couches :

- Couche d'entrée : La première couche qui reçoit les données brutes.
- Couches cachées : Une ou plusieurs couches situées entre l'entrée et la sortie. Ces couches effectuent des transformations sur les données reçues.
- Couche de sortie : La dernière couche qui produit le résultat final du réseau.

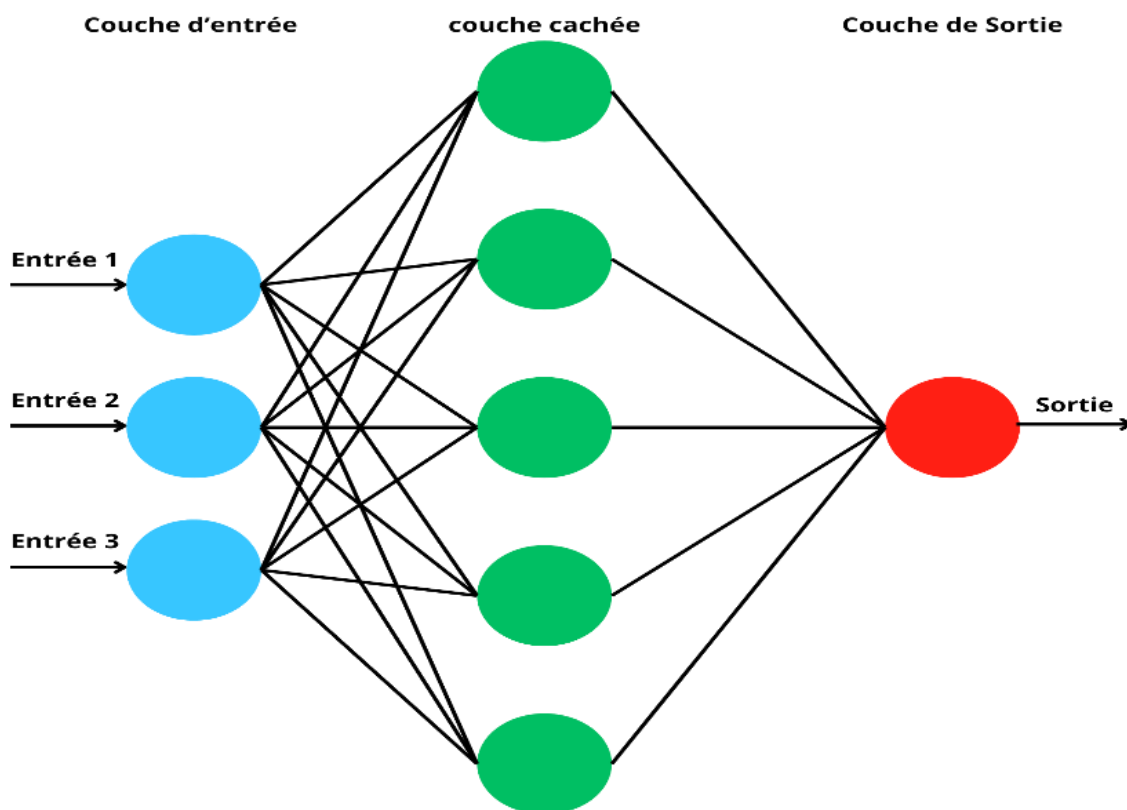


Figure 35 Couches de neurone artificiel

2.2.3 Fonctionnement :

Le fonctionnement d'un réseau neuronal repose sur des unités de traitement nommées neurones artificiels ou perceptrons, qui sont agencés en couches interconnectées. Chaque couche comporte plusieurs neurones qui traitent les signaux d'entrée en transmettant ces derniers à travers le réseau. Ces signaux subissent des transformations via des poids et des fonctions d'activation, avant de générer des résultats en sortie.

2.2.4 Les différentes architectures du réseau neuronal :

2.2.4.1 Réseaux de neurones multicouches (MLP) :

Ils se composent de multiples couches de neurones, incluant une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Les signaux se propagent de l'entrée vers la sortie sans boucles de rétroaction.

2.2.4.2 Réseaux neuronaux récurrents (RNN) :

Ils intègrent des mécanismes de rétroaction qui facilitent la circulation d'informations à travers le réseau à différents moments dans le temps. Les RNN sont particulièrement appropriés pour le traitement de données séquentielles, comme la reconnaissance vocale, la traduction automatique, et la prédiction de séries temporelles.

2.2.4.2.1 Réseaux neuronaux convolutifs (CNN) :

Les Convolutional Neural Networks (CNN), également connus sous le nom de réseaux neuronaux convolutifs, sont spécialement conçus pour le traitement de données structurées telles que les images. Ils sont largement utilisés dans le domaine de la vision par ordinateur pour des tâches telles que la classification, la détection d'objets, la segmentation sémantique, et bien d'autres.[14]

Leur efficacité découle de leur capacité à exploiter la structure spatiale des données d'entrée. Contrairement aux réseaux neuronaux traditionnels où chaque neurone est connecté à tous les neurones de la couche précédente.

Les CNN utilisent des opérations de convolution pour extraire des caractéristiques des données tout en préservant leur arrangement spatial.[20]

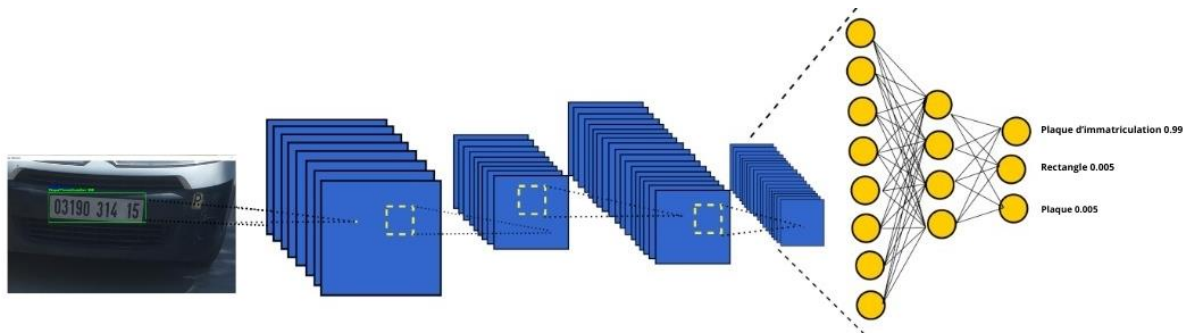


Figure 36 Réseaux de neurones convolutifs

3 Les Réseaux neuronaux convolutifs :

3.1 Composants et opérations des CNN :

3.1.1 Couche de convolution :

La couche de convolution est l'élément central d'un réseau de neurones convolutionnel (CNN). Elle utilise des filtres ou noyaux pour analyser les données d'entrée, généralement des images, et extraire automatiquement les caractéristiques pertinentes. Ces caractéristiques, ou "features", peuvent inclure des contours, des textures, et d'autres motifs visuels essentiels à la tâche d'apprentissage. Chaque filtre détecte un aspect spécifique de l'image, et la combinaison de plusieurs filtres permet au CNN de saisir des structures de plus en plus complexes au fur et à mesure que l'on progresse dans les couches du réseau.

Mathématiquement, la convolution d'une image $I(x, y)$ avec un filtre $W(i, j)$ s'écrit :

$$(I * W)(x, y) = \sum (\sum I(x - i, y - j) * W(i, j))$$

- $I(x, y)$ est la valeur du pixel à la position (x, y) dans l'image d'entrée
- $W(i, j)$ est la valeur du poids à la position (i, j) dans le filtre
- (x, y) parcourt tous les pixels de l'image
- (i, j) parcourt tous les éléments du filtre

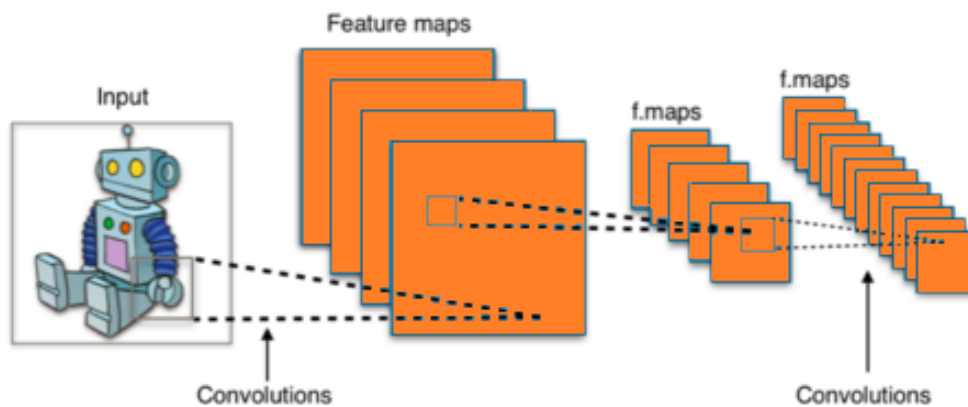


Figure 37 Couche de convolution

Cette opération est répétée pour chaque filtre, générant ainsi une carte d'activation pour chacun d'eux.

3.1.2 Couche de pooling :

Après chaque couche de convolution, il est habituel d'incorporer une couche de pooling pour diminuer la dimensionnalité des données et améliorer la résilience du modèle contre les translations et les déformations mineures des données. Les opérations de pooling, telles que le max pooling ou le average pooling, fusionnent les informations sur des zones locales de la carte des caractéristiques, réduisant ainsi la taille spatiale des données.

La couche de pooling dans les réseaux de neurones convolutifs (CNN) utilise généralement deux types de fonctions pour réduire la taille des feature maps tout en préservant les caractéristiques les plus importantes :

- Max pooling :

La fonction de max pooling sélectionne la valeur maximale au sein d'une fenêtre de pooling de taille $n \times n$. Mathématiquement, pour une fenêtre (x) , la fonction de max pooling est définie comme suit :

$$f(x) = \max(x)$$

Le max pooling conserve les valeurs les plus élevées, qui correspondent souvent aux caractéristiques les plus significatives détectées.

- Average pooling :

La fonction d'average pooling calcule la moyenne des valeurs au sein d'une fenêtre de pooling de taille $n \times n$. Elle est définie par :

$$f(x) = \text{mean}(x)$$

L'average pooling sous-échantillonne la feature map en conservant une représentation plus lisse des activations.

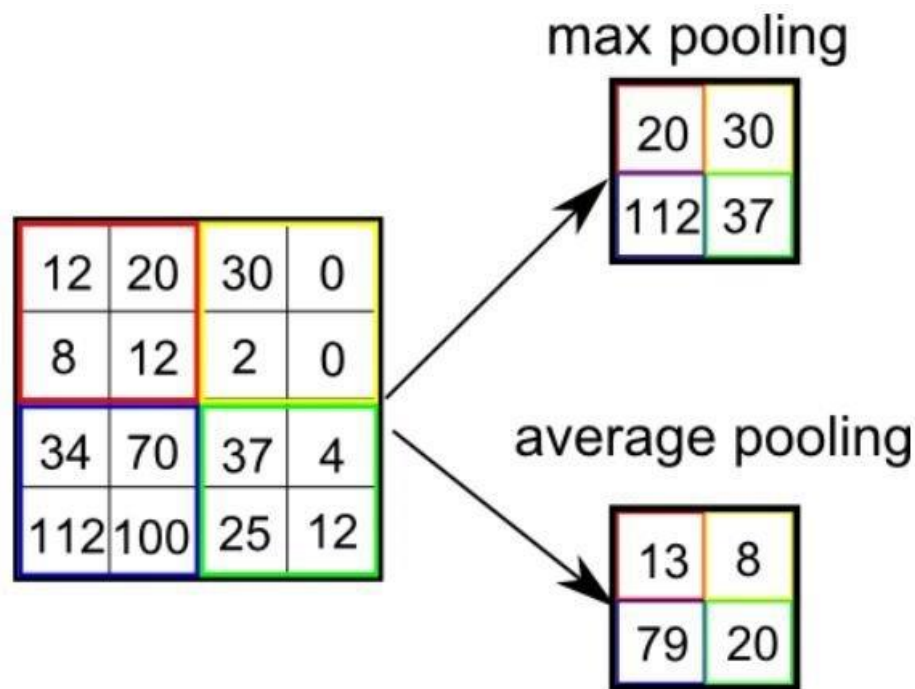


Figure 38 Couche de Pooling

3.1.3 Couche d'activation ReLU :

La fonction d'activation ReLU (Rectified Linear Unit) est l'une des plus populaires utilisées dans les réseaux de neurones artificiels. Elle accélère l'entraînement par rapport à d'autres fonctions d'activation en éliminant les valeurs négatives dans le réseau de neurones. ReLU est largement adoptée dans les réseaux de neurones profonds, car elle facilite une formation plus rapide et efficace des architectures neuronales profondes sur des ensembles de données volumineux et complexes.

La fonction d'activation ReLU (Rectified Linear Unit) fonctionne en éliminant les valeurs négatives dans un réseau de neurones. Si l'entrée est négative, la sortie sera 0, tandis que si l'entrée est positive, la sortie sera égale à l'entrée. Bien que la ReLU soit non différentiable à zéro, ce qui peut poser des problèmes avec des valeurs proches de zéro, elle reste largement

utilisée. Cependant, la ReLU n'est pas centrée, ce qui crée une asymétrie en ne traitant que les valeurs positives et peut entraîner une gestion déséquilibrée des données.

$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad f(x) = \max(0, x)$$

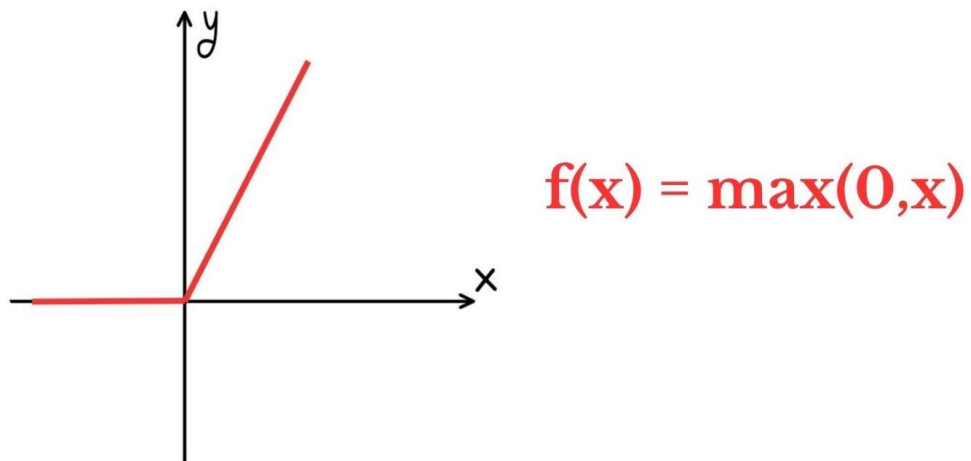


Figure 39 Fonction d'activation ReLU

3.1.4 Couches Sigmoides :

La fonction d'activation sigmoïde est une fonction mathématique non linéaire couramment utilisée dans les réseaux neuronaux. Elle joue un rôle essentiel dans le processus d'apprentissage pour les réseaux, leur permettant de modéliser des relations complexes entre les entrées et les sorties.

Elle prend des valeurs réelles en entrée pour renvoyer des valeurs de sorties comprises entre 0 et 1.

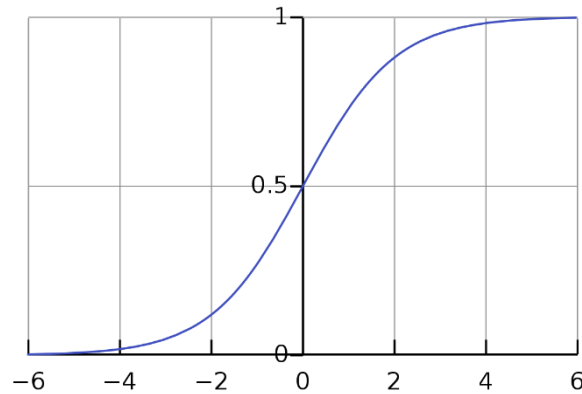


Figure 40 Fonction de Sigmoide

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

3.1.5 Couches Fully-Connected :

Une couche fully connected (FC) est un type de couche utilisée dans les réseaux de neurones artificiels, en particulier dans les réseaux de neurones convolutifs (CNN).

Dans une couche fully connected (FC), chaque neurone est relié à tous les neurones de la couche précédente, ce qui permet d'établir des relations complexes entre les caractéristiques extraites par les couches antérieures. Les couches FC sont généralement situées à la fin d'un CNN, après les couches de convolution et de pooling. Leur rôle est de combiner les caractéristiques détectées précédemment pour effectuer la classification finale de l'image.

Contrairement aux couches de convolution qui se concentrent sur des motifs locaux, les couches FC adoptent une approche globale en combinant toutes les caractéristiques. Chaque neurone d'une couche FC effectue une combinaison linéaire des sorties de la couche précédente, suivie d'une fonction d'activation, ce qui permet d'apprendre des fonctions complexes.

Le nombre de neurones dans la dernière couche FC correspond généralement au nombre de classes de classification, chaque neurone indiquant la probabilité d'appartenance à une classe spécifique.

En résumé, les couches FC permettent d'intégrer les caractéristiques locales extraites par les couches précédentes pour réaliser une classification globale de l'image dans un CNN.

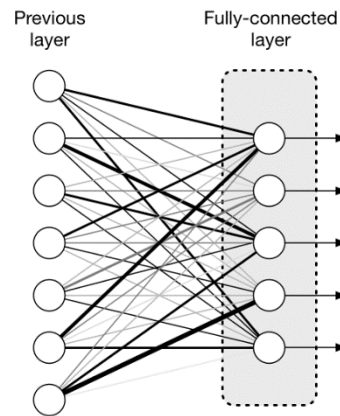


Figure 41 Couche Fully Connected

3.2 Fonctionnement des couches des CNN :

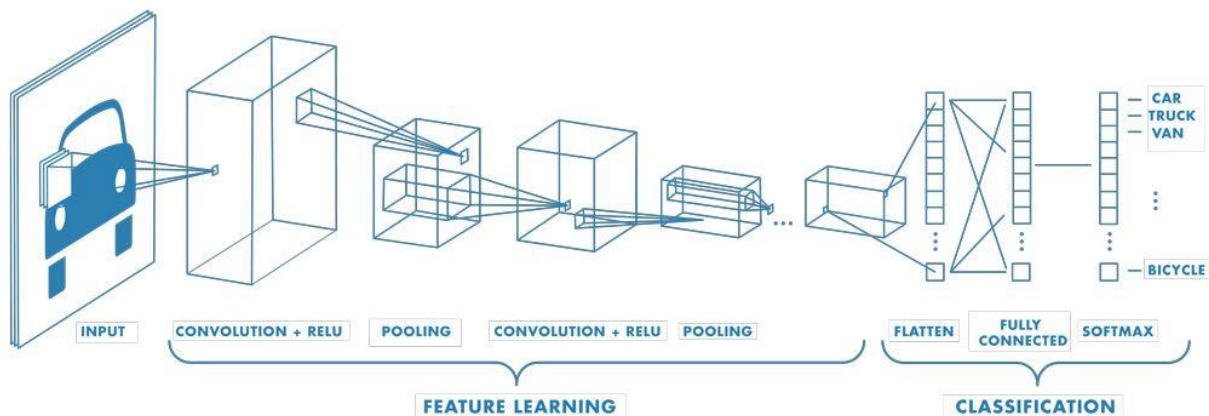


Figure 42 Fonctionnement des couches de CNN

3.3 Entraînement d'un modèle CNN :

- Préparation des données d'entraînement :
 - Établir une base de données d'images étiquetées qui représente le problème à résoudre, ces images doivent appartenir à des classes prédéfinies.
 - Ensuite il faut réaliser une base de données pour l'entraînement du modèle et une base de données pour la validation.
- Architecture du CNN :
 - Définir l'architecture du modèle CNN (les différentes couches).
- Entraînement du modèle :
 - Initialiser aléatoirement les poids du CNN

- Appliquer le CNN sur les images d'entraînement et calculer l'erreur entre les prédictions et les étiquettes réelles
- Répéter plusieurs fois le processus sur toutes les images d'entraînement.
- Évaluation et optimisation :
 - Suivre et évaluer les performances du modèle.
 - Surveiller le sur-apprentissage (modèle trop performant sur l'entraînement) et le sous-apprentissage (modèle peu performant)
 - Ajuster l'architecture, les hyper paramètres ou ajouter des données pour optimiser les performances.

Remarque :

Nous pouvons utiliser un modèle pré-entraîner ou lieu de tout reprendre de zéro, cela nous aidera à gagner du temps, et l'entraînement d'un modèle nécessite un matériel performant.

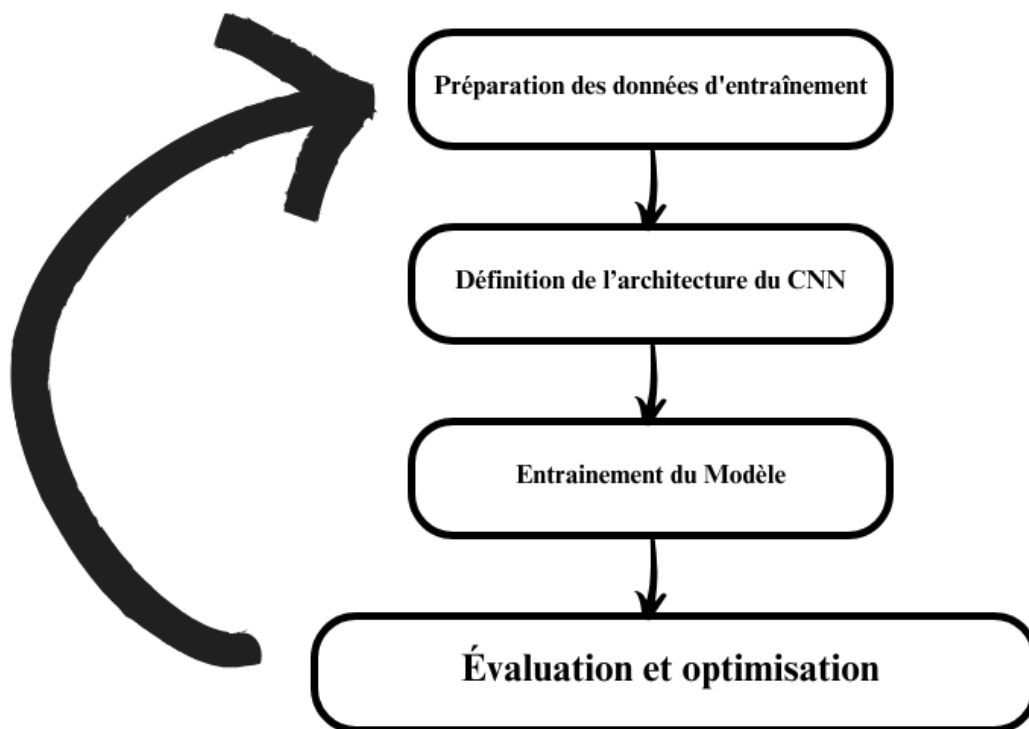


Figure 43 Schéma synoptique d'entraînement d'un modèle CNN

4 Deep Learning :

Le Deep Learning, également connu sous le nom d'apprentissage profond, représente une forme d'intelligence artificielle issue du machine Learning, où la machine acquiert la capacité d'apprendre de manière autonome [13]. Cette méthode diffère de la programmation traditionnelle, où la machine se contente d'exécuter des instructions définies à l'avance, sans capacité d'auto-apprentissage.

4.1 Fonctionnement du Deep Learning :

Le Deep Learning repose sur un réseau de neurones artificiels inspiré du fonctionnement du cerveau humain [16]. Ce réseau est constitué de plusieurs couches de neurones, allant parfois jusqu'à des dizaines voire des centaines de couches. Chaque couche reçoit et traite les informations provenant de la couche précédente. Par exemple, le système peut apprendre à reconnaître les lettres avant de passer à l'identification des mots dans un texte, ou à déterminer la présence d'un visage sur une photo avant d'identifier la personne représentée.

4.1.1 Comment le deep-learning utilise les réseaux CNN :

- Reconnaissance d'Images :
Les réseaux de neurones convolutifs (CNN) sont utilisés pour la classification d'images, où chaque image se voit attribuer une étiquette parmi un ensemble de classes prédéfinies. Voici le processus typique :
 - Prétraitement : Les images sont normalisées et redimensionnées.
 - Extraction de Caractéristiques : Les couches de convolution et de pooling extraient les caractéristiques des images.
 - Classification : Les couches entièrement connectées (FC) situées à la fin du réseau utilisent ces caractéristiques pour classer les images.
- Détection d'objet :
La détection d'objets implique la localisation et la classification des objets dans une image. Les réseaux de neurones convolutifs (CNN) sont utilisés dans plusieurs architectures pour cette tâche :
 - R-CNN (Regions with CNN features) : Génère des propositions de régions et utilise un CNN pour classifier chaque région.
 - YOLO (You Only Look Once) et SSD (Single Shot MultiBox Detector): Ces architectures divisent l'image en une grille et prédisent simultanément les cadres de délimitation et les classes d'objets pour chaque cellule de la grille.
 - Segmentation d'Images :

- La segmentation d'images consiste à attribuer une étiquette à chaque pixel, permettant ainsi de distinguer différentes régions ou objets dans une image. Les réseaux de neurones convolutifs (CNN) utilisés pour la segmentation incluent :
- FCN (Fully Convolutional Networks) : Remplace les couches entièrement connectées par des couches de convolution pour produire des cartes de segmentation.
- U-Net : Utilisé principalement dans les applications médicales, il possède une architecture symétrique qui permet des segmentations précises.

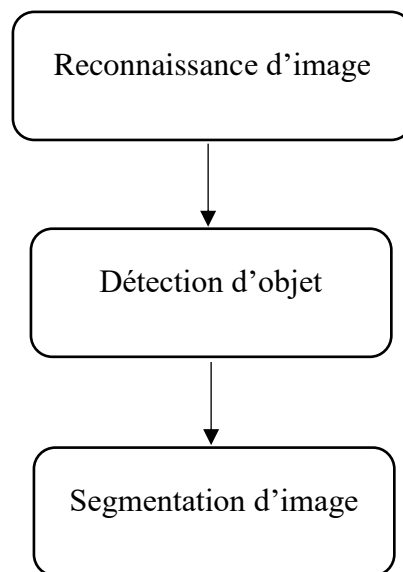


Figure 44 Utilisation des CNN par le deep Learning

5 Reconnaissance de caractères avec Deep Learning :

5.1 EasyOCR :

C'est un outil de reconnaissance optique de caractères open-source utilisé pour l'extraction des textes à partir des images. L'architecture de EasyOCR se base sur des modèles du Deep Learning et utilise différentes techniques pour traiter les images et en extraire le contenu voulu.

Ces techniques sont représentées par :

- La réduction du bruit : Elle supprime le bruit des images en utilisant différents filtres comme le filtre flou gaussien et le flou médian.
- La binarisation : Elle convertit les données en images binaires en utilisant différentes méthodes tel que le thresholding adaptative.

- La correction d'angle : EasyOCR détecte et corrige l'inclinaison des caractères dans les images pour permettre une lecture correcte des textes.

5.1.1 Détection des zones dans les images contenant du texte :

Autrement appelé CRAFT (Character-Region Awareness for Text detection), est un modèle de détection de texte précis et efficace qui peut détecter les textes sous n'importe quelle forme, région, orientation et police. Il utilise un réseau neuronal CNN pour produire deux cartes de sortie :

- La carte du score de région : C'est une carte binaire qui indique l'appartenance d'un pixel à une zone de caractères, elle est utilisée pour localiser les caractères individuels dans l'image
- La carte du score d'affinité : C'est une carte binaire qui indique la probabilité que deux caractères adjacents appartiennent à la même instance du texte, elle sert à localiser si les caractères appartiennent au même mot.

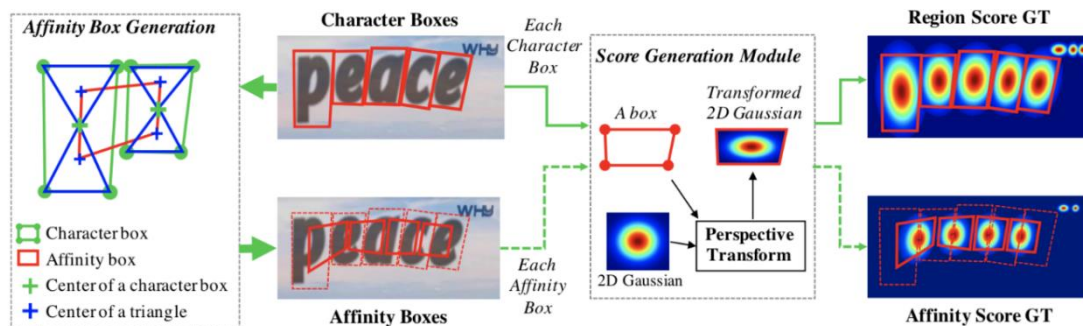


Figure 45 Détection de zones

5.1.2 Architecture du modèle Deep Learning utilisée dans EasyOCR (CRNN) :

L'architecture de ce modèle est composée de trois éléments et fonctionne de manière successive, ils sont représentés par :

- Le réseau de neurones convolutifs (CNN) : Cette partie est responsable du traitement de l'image d'une manière qui préserve les relations spatiales entre les pixels, tout en réduisant la dimensionnalité de l'image en utilisant un réseau composé de couches convolutives et couches de max-pooling.
- Le réseau de neurones récurrent (RNN) : Ce réseau utilise les caractéristiques extraites à partir de son prédécesseur pour le traitement, il se compose de plusieurs couches de cellules LSTM (Long Short-Term Memory) qui sont conçues pour capturer des

dépendants à long terme dans des données séquentielles en se souvenant sélectivement et en oubliant des informations.

- Classification temporelle connexionniste (CTC) : Cet élément permet de décoder les prédictions par image faite par le réseau RNN afin de les interpréter en une séquence d'étiquettes de texte écrit et à pouvoir prédire les sorties.

5.1.3 Décodage des sorties :

Après que le traitement soit fini, et que la distribution de probabilité générée sur les caractères de chaque image d'entrée, la stratégie de décodage est généralement utilisée dans le cadre du processus du décodage, en particulier au CTC afin de trouver la séquence de caractères la plus probable en tenant en compte des probabilités de sorties.

Exemples de décodeurs :

- Décodeur gourmand : Il fonctionne en sorte à sélectionner le caractère le plus probable à chaque étape de la séquence. Pour chaque séquence donnée le modèle de reconnaissance produit une distribution de probabilité sur tous les caractères possibles.
- Décodeur de recherche en faisceau : Ce modèle peut être utilisé afin d'améliorer la précision de la détection du texte, contrairement au précédent, ce décodeur fait en sorte qu'à chaque étape de la séquence, il considère de multiples possibilités pour chaque caractère.

5.2 Post-traitement des caractères reconnus :

Cette étape implique l'utilisation de nombreuses applications de techniques et d'algorithmes supplémentaires afin d'affiner les sorties. Son but principal est de corriger toute éventuelle erreur ou incohérence qui a pu être détecté dans le texte reconnu et d'améliorer la qualité de l'extraction.

6 Application de réseaux neuronaux dans les systèmes RAPI :

L'utilisation des réseaux de neurones pour la détection et la reconnaissance de plaque d'immatriculation est de même pour d'autres objets, la seule différence est les données à utiliser pour l'entraînement du modèle CNN.

Nous avons utilisé le deep Learning pour la détection et pour la reconnaissance

6.1 Deep Learning pour la détection de plaque d'immatriculation :

Pour la détection de plaque d'immatriculation, nous avons remarqués que les réseaux neuronaux convolutifs CNN sont les plus adaptés et plus efficaces dans la détection, voici un

schéma synoptique qui explique brièvement le fonctionnement des CNN pour les systèmes RAPI :

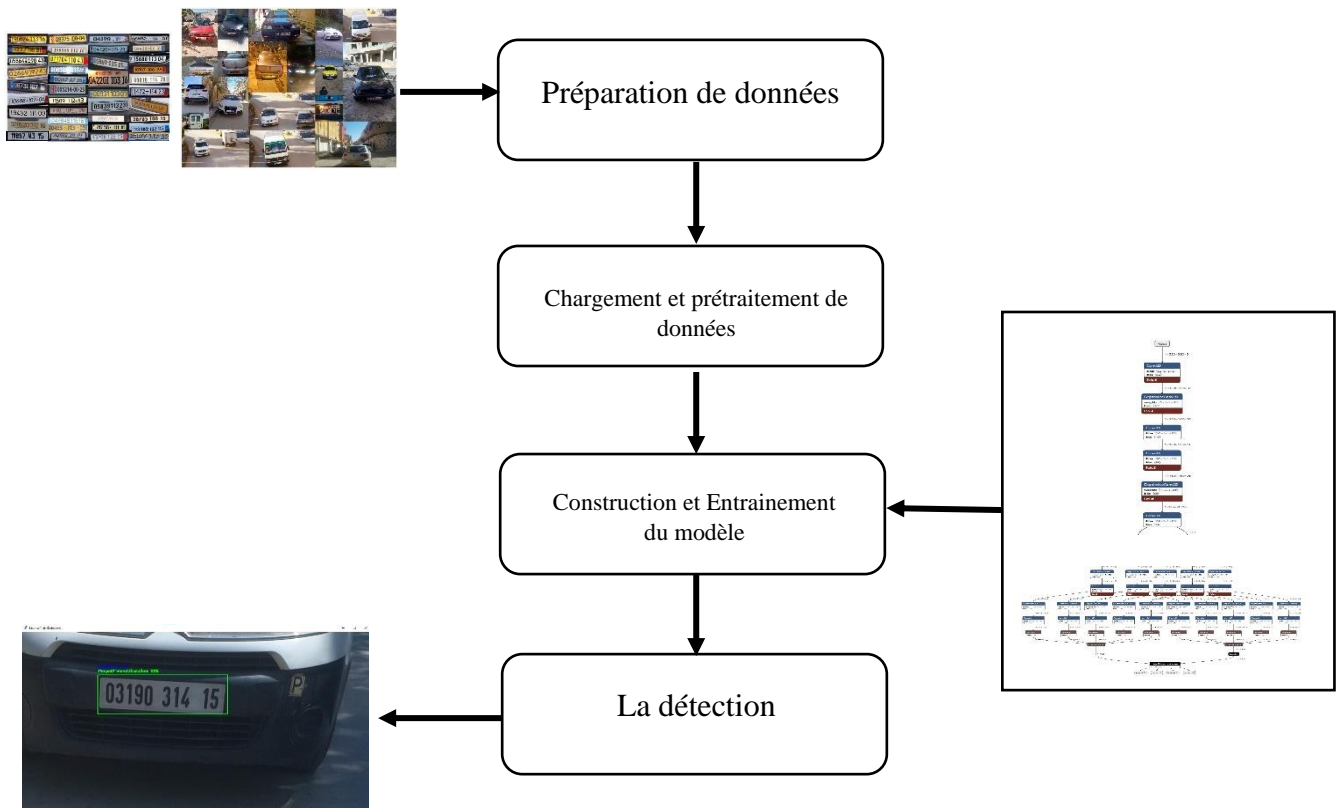


Figure 46 Détection de plaque d'immatriculation avec le Deep Learning

Nous pouvons traduire la détection de la plaque par la localisation de cette dernière dans une image, et ensuite l'extraire pour l'utiliser dans la prochaine étape qui consiste à reconnaître les chiffres de la plaque.

6.2 Deep Learning pour la reconnaissance de chiffres de la plaque d'immatriculation :

Comme nous l'avons précisé dans l'étape précédente, la reconnaissance des chiffres vient après la détection de la plaque, l'objectif de cette étape est d'identifier les chiffres présents sur la plaque.

Discussion :

Dans ce chapitre, nous avons présenté une vue générale sur l'IA et les notions importantes sur le Deep Learning où nous avons pu comprendre le fonctionnement des réseaux neuronaux.

Le chapitre suivant, va se concentrer sur les détails de la conception de notre système et les outils utilisés dedans en implémentant un modèle du Deep Learning pour la reconnaissance de plaque d'immatriculation.

Chapitre 4 :

Conception et Réalisation

Préambule :

Ce chapitre nous l'avons divisé en trois parties principales, à savoir une partie pour présenter la conception de notre système et les différents matériels utilisés, ensuite une partie où nous avons décrit les logiciels et la partie programmation, et une partie que nous avons dédié au système de paiement ainsi que la présentation de notre plateforme de paiement.

A la fin du chapitre nous allons voir les différents résultats, et un constat de l'utilisation du Deep-Learning pour la reconnaissance et la détection de plaque d'immatriculation. et une petite comparaison avec les méthodes existantes pour le paiement en ligne.

1 Conception de Notre système:

1.1 Schéma global du fonctionnement de notre Système :

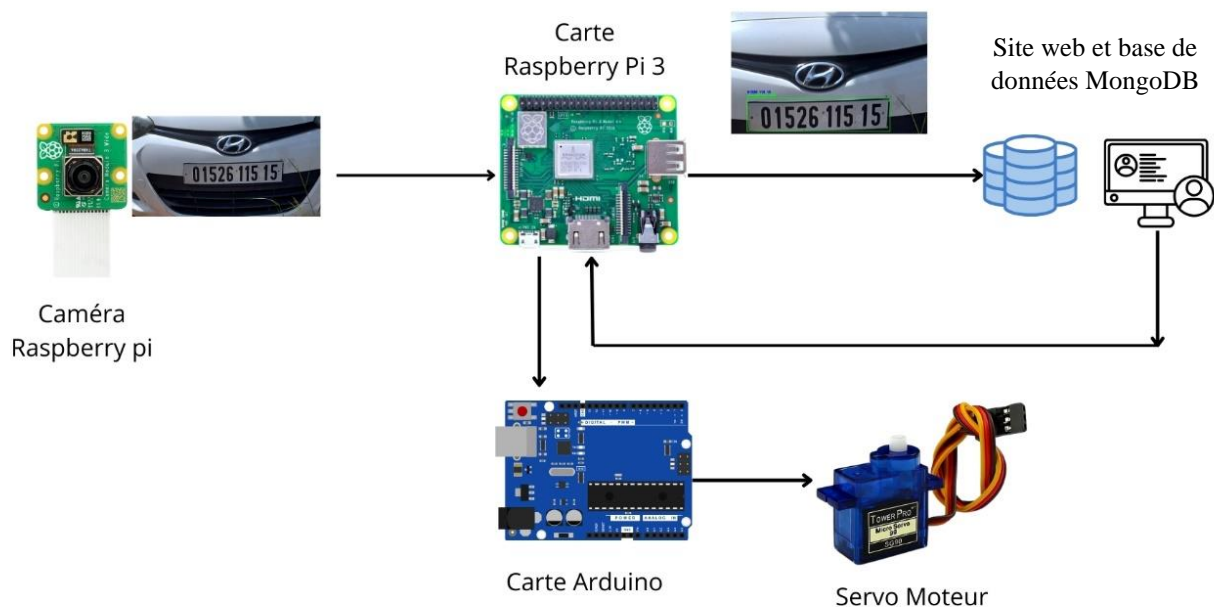


Figure 47 Schéma global du fonctionnement de notre Système

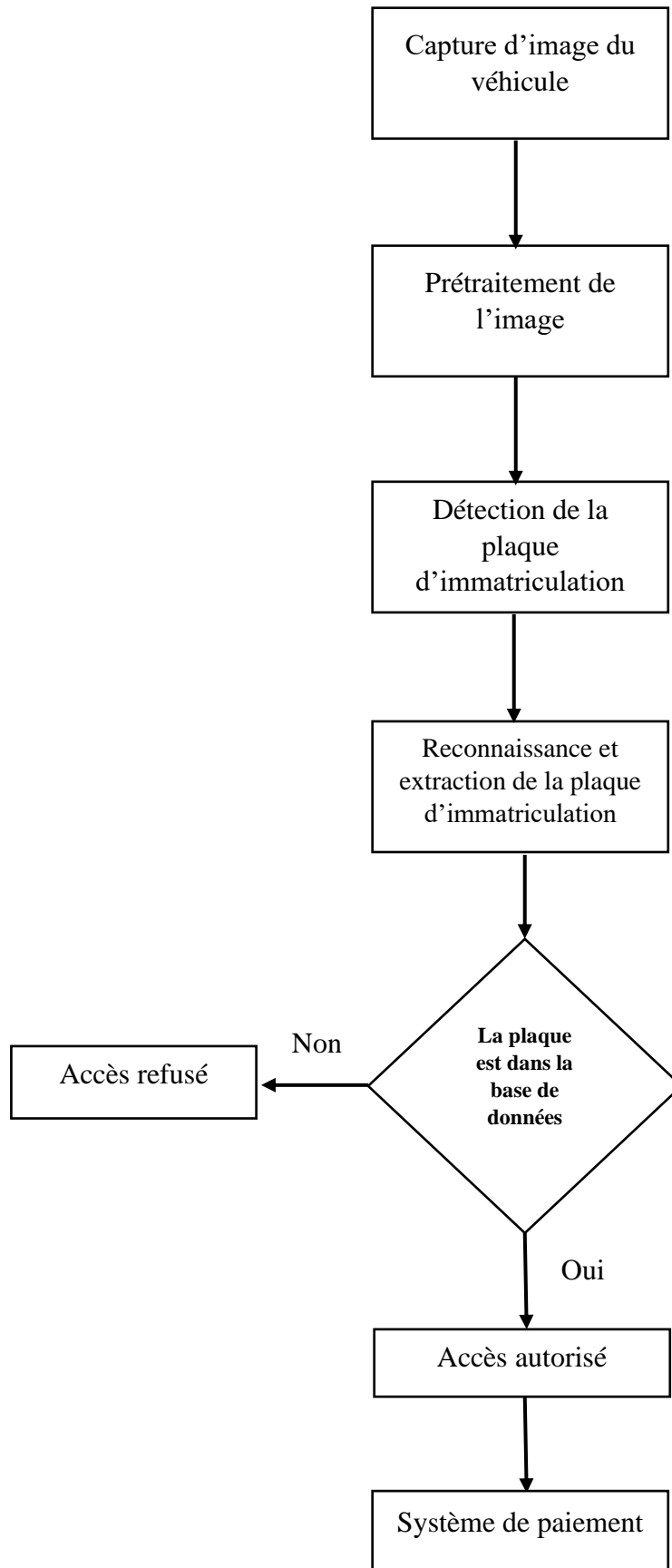


Figure 48 Organigramme du fonctionnement de notre système

1.2 Matériels utilisés :

1.2.1 Carte Raspberry pi 3 modèle B :

La Raspberry Pi 3 Modèle B est une carte mère à faible coût et de petite taille, conçue pour être polyvalente et accessible[17].

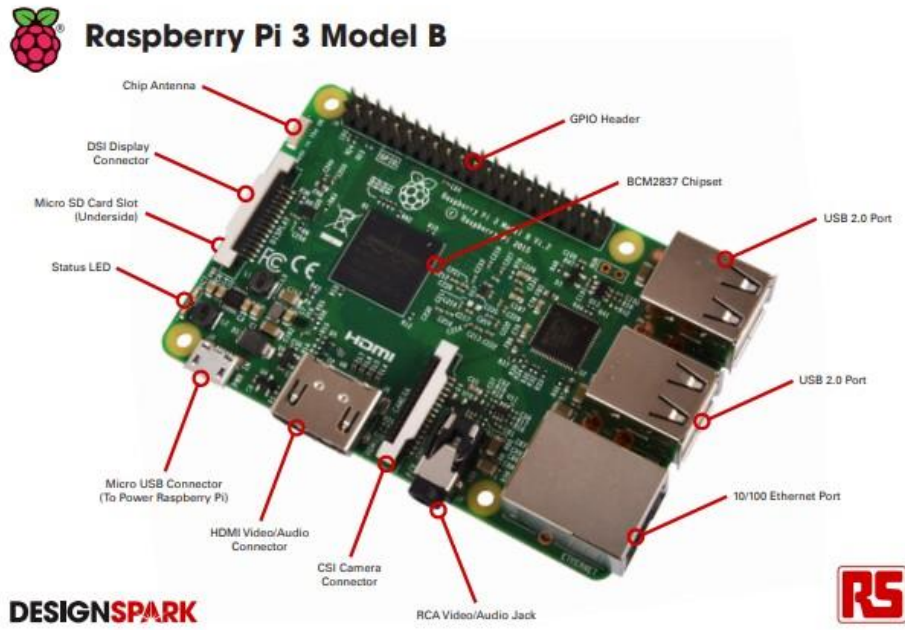


Figure 49 Carte Raspberry Pi

Processeur	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Mémoire	1GB LPDDR2
Système d'exploitation	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Allimentation	Micro USB socket 5V1, 2.5A
Wi-Fi	802.11n à 2,4 GHz

Bluetooth	Bluetooth 4.1, Bluetooth Low Energy (BLE)
Ethernet	10/100 Base-T Ethernet (Raspberry Pi) (The MagPi magazine)
USB	4 ports USB 2.0
Video	Sortie HDMI pleine taille, sortie vidéo composite via un connecteur 3,5 mm
Audio	Sortie stéréo 4 pôles
Stockage	Emplacement pour carte microSD pour le système d'exploitation et le stockage des données
Interfaces	Ports CSI pour caméra et DSI pour écran tactile (Raspberry Pi) (Element14 Community) (The MagPi magazine)
GPIO	En-tête GPIO à 40 broches pour la connexion de périphériques et de capteurs

1.2.2 Caméra Raspberry Pi :

La caméra Raspberry Pi 5MP V1.3 est un ancien modèle de caméra compatible avec les Raspberry Pi, sorti en 2013. Voici les principales caractéristiques de cette caméra :

1. Spécifications

- Capteur : OmniVision OV5647
- Résolution photo maximale : 5 MP (2592 x 1944 pixels)
- Résolution vidéo maximale : 1080p (1920 x 1080 pixels, 30 fps)
- Dimensions : 25 x 20 x 9 mm
- Poids : 3 g

2. Contenu

- Caméra Raspberry Pi 5MP V1.3
- Câble flex de 15 cm

3. Disponibilité

Cette caméra n'est plus produite par Raspberry Pi, mais elle est encore disponible chez certains revendeurs comme TinyTronics, Bits & Parts et Reichelt.

Il existe également une version avec éclairage infrarouge vendue par TinyTronics.



Figure 50 Caméra Raspberry pi

Depuis, Raspberry Pi a sorti de nouveaux modèles plus performants, comme la Caméra Module 2 (8 MP) en 2016 et la Caméra Module 3 (12 MP) en 2023, offrant des résolutions et des performances améliorées.

1.2.3 Carte Arduino UNO :

L'Arduino Uno est l'une des cartes les plus populaires et polyvalentes de la gamme Arduino. Voici quelques informations clés :

- Microcontrôleur : Basée sur l'ATmega328P
- Broches : 14 broches d'entrée/sortie numériques (dont 6 peuvent servir de sorties PWM), 6 entrées analogiques
- Horloge : Oscillateur à quartz de 16 MHz
- Connexions : Port USB, connecteur d'alimentation, connecteur ICSP, bouton de réinitialisation

1. Alimentation

- Option : Peut être alimentée via la connexion USB ou une source d'alimentation externe
- Tension recommandée : 7 à 12V

2. Dimensions

- Taille : 68,6 mm x 53,4 mm
- Usage : Taille standard permettant l'insertion dans un boîtier de protection

3. Programmation

- IDE Arduino : Environnement de développement intégré open source et multiplateforme
- Langage : Programmation en C/C++



Figure 51 Carte Arduino Uno

4. Applications

La carte Uno est idéale pour les débutants en Arduino et convient à une large gamme de projets, de l'électronique de base aux applications plus avancées, grâce à sa polyvalence et à la grande communauté de ses utilisateurs.

1.2.4 Servo Moteur :

Un servo-moteur est un dispositif électromécanique qui régule le déplacement d'un objet ou d'un mécanisme en réaction à un signal de commande externe. Il est largement utilisé dans une gamme variée d'applications telles que les robots, les drones, les systèmes de positionnement et les modèles télécommandés. Typiquement, un servo-moteur est composé d'un moteur électrique, d'un circuit de contrôle et d'un système de rétroaction, souvent un potentiomètre, qui fournit une information de retour sur la position. Ils sont particulièrement appréciés pour leur précision dans le positionnement et leur capacité à maintenir cette position même sous une charge variable.

- Fonctionnement :

Le servo-moteur opère en réceptionnant un signal de commande constitué de signaux électriques pulsés. Ce signal spécifie la position exacte vers laquelle le servo doit se déplacer. Ensuite, le servo compare cette commande à son propre capteur de position, généralement un potentiomètre, et ajuste son angle de rotation en conséquence afin d'atteindre la position désirée.



Figure 52 Servo Moteur

1.2.5 Capteur Infrarouge :



Figure 53 Capteur infrarouge

Le capteur Infrarouge est composé d'une diode infrarouge qui joue le rôle d'émetteur et d'une photodiode qui joue le rôle d'un récepteur. Suivant la distance à laquelle se trouve l'obstacle, le récepteur recevra plus ou moins de lumière infra-rouge réfléchi

Caractéristiques :

- Distance de détection : de 2 à 30 cm
- Dimensions : 3.1 cm * 1.5 cm
- Alimentation : 3.3 – 5 V.
- OUT : interface de sortie numérique de la carte (0 si détection, et 1 si aucune détection).

Une fois que le capteur est branché, une LED s'allume sur le capteur si un obstacle est détecté, la distance de détection peut être ajustée à l'aide d'un potentiomètre

1.3 Logiciels et Programmes utilisés :

1.3.1 Langage de programmation Python :

Python, un langage de programmation très prisé, a été conçu par Guido van Rossum et lancé en 1991. Il trouve son utilisation dans divers domaines, y compris le développement web, le développement logiciel, les mathématiques et le Scripting système. Ce langage est renommé pour sa syntaxe simple et proche de l'anglais, sa compatibilité avec plusieurs plateformes, et sa flexibilité qui facilite le développement rapide de logiciels. En outre, Python est largement employé dans des secteurs comme la science des données, l'intelligence artificielle, l'automatisation, les tests logiciels et le développement backend.

Il se distingue également par sa facilité d'apprentissage, sa communauté active, et son riche écosystème de modules et bibliothèques. Python est un langage de programmation puissant et adaptable, offrant de nombreuses opportunités aux développeurs.



Figure 54 Python

- Le python et le Deep-Learning :

Le Python est largement célébré dans le domaine du Deep Learning, en grande partie grâce à des Framework comme TensorFlow et EasyOCR. TensorFlow, développé en open-source par Google, est spécifiquement conçu pour Le Machine Learning et le Deep Learning, offrant des outils puissants pour la construction de réseaux neuronaux et de modèles d'apprentissage en profondeur. En parallèle, EasyOCR est une bibliothèque Python qui simplifie et optimise la reconnaissance optique de caractères.

Cette popularité du Python dans le domaine du Deep Learning s'explique par sa syntaxe limpide et sa grande adaptabilité, ce qui en fait un choix de prédilection pour les projets d'intelligence artificielle.

En tirant parti de bibliothèques telles que TensorFlow et EasyOCR, les développeurs peuvent mettre en œuvre des solutions avancées de Deep Learning pour diverses applications, allant de la reconnaissance d'objets à la détection de texte, et bien d'autres encore.



Figure 55 Python et Machine Learning

1.3.2 Tensorflow :

TensorFlow est une plateforme de machine Learning créée par Google et mise en ligne en 2015 sous licence Apache [19]. Elle est basée sur le système DistBelief de Google.

TensorFlow est une bibliothèque logicielle open source pour le calcul numérique, offrant une architecture flexible qui permet de déployer facilement des calculs sur différentes plateformes, comme les CPU, GPU, TPUs, les ordinateurs de bureau, les clusters de serveurs, et même les appareils mobiles. Elle est surtout utilisée pour Le Machine Learning et le Deep Learning, mais son noyau de calcul est aussi employé dans d'autres domaines scientifiques.

TensorFlow est conçue pour simplifier la création et l'entraînement de modèles d'apprentissage profond en permettant de construire des graphes de calcul.

C'est un outil puissant pour développer des modèles d'intelligence artificielle et il est largement utilisé dans la communauté du machine Learning.



Figure 56 TensorFlow

1.3.3 TensorFlow Lite :

TensorFlow Lite est une version allégée de TensorFlow conçue pour exécuter des modèles de machine Learning sur des appareils mobiles, intégrés et IoT [18]. Cette version de TensorFlow présente plusieurs avantages importants :

- Optimisation pour le Machine Learning sur appareil : TensorFlow Lite est conçu pour fonctionner directement sur l'appareil, en tenant compte de la latence, de la confidentialité, de la connectivité, de la taille du modèle et de la consommation d'énergie, ce qui permet de se passer d'une connexion Internet.
- Compatibilité multiplateforme : Il fonctionne sur diverses plateformes telles qu'Android, iOS, Linux intégré et les microcontrôleurs, offrant ainsi une grande flexibilité d'utilisation.
- Support de plusieurs langages : TensorFlow Lite est compatible avec plusieurs langages de programmation, notamment Java, Swift, Objective-C, C++ et Python, facilitant ainsi le travail des développeurs.
- Performances élevées : Grâce à l'accélération matérielle et à l'optimisation des modèles, TensorFlow Lite garantit une exécution rapide et efficace des modèles de machine Learning.
- Exemples pratiques : Il propose des exemples complets pour des tâches courantes de machine Learning, telles que la classification d'images, la détection d'objets, l'analyse des postures, la réponse à des questions et la classification de texte, sur plusieurs plateformes.

En résumé, TensorFlow Lite est une solution puissante et optimisée pour exécuter des modèles de machine Learning sur des appareils mobiles et intégrés, avec des fonctionnalités adaptées permettant de déployer l'apprentissage automatique directement sur l'appareil.



Figure 57 TensorFlow Lite

1.3.4 EasyOcr :

EasyOCR est une bibliothèque open source de reconnaissance optique de caractères (OCR) pour Python, idéale pour extraire du texte à partir d'images et particulièrement utile dans le deep Learning pour le traitement d'images contenant du texte.

Voici les points clés à connaître sur EasyOCR pour le traitement d'images en deep Learning :

- Facilité d'installation et d'utilisation : EasyOCR est simple à installer et utiliser, rendant l'OCR avec Python très accessible. Il suffit d'installer les paquets PyTorch (nécessaires sous Windows) et EasyOCR pour commencer à extraire du texte d'images.
- Support multilingue : EasyOCR prend en charge plus de 80 langues, offrant une grande flexibilité pour les applications multilingues.
- Extraction automatique d'informations clés : La bibliothèque peut extraire automatiquement des informations importantes à partir de documents, comme les données de cartes de visite ou le texte de documents imprimés, ce qui la rend puissante pour l'automatisation du traitement de documents.
- Compatibilité avec d'autres bibliothèques de deep learning : EasyOCR peut être utilisée avec des bibliothèques comme OpenCV pour des applications avancées de traitement d'images, permettant de profiter d'un large éventail d'algorithmes de vision par ordinateur.
- Connaissance d'autres techniques d'OCR : Bien qu'EasyOCR soit une solution complète, il est utile de connaître d'autres techniques d'OCR pour plus de flexibilité et pour choisir la méthode la plus adaptée à chaque projet.



Figure 58 EasyOCR

En résumé, EasyOCR est un outil puissant et facile à utiliser pour extraire du texte dans des applications de deep learning sur des images. Sa simplicité, son support multilingue et son

intégration avec d'autres bibliothèques en font un excellent choix pour de nombreux projets de traitement d'images.

1.3.5 Tkinter :

Tkinter est la bibliothèque graphique native pour Python, utilisée pour créer des interfaces graphiques. Elle est dérivée de la bibliothèque graphique Tk, initialement développée pour le langage Tcl. Tkinter est l'interface standard de Python pour la boîte à outils GUI Tcl/Tk.

1.3.6 OpenCV :

OpenCV, abréviation de Open Source Computer Vision Library, est une bibliothèque open source dédiée au traitement d'images en temps réel. Initialement développée par Intel, elle est désormais maintenue par une communauté de développeurs sous la fondation OpenCV. Jouant un rôle crucial dans la vision par ordinateur et l'apprentissage automatique, OpenCV permet de traiter des images et des vidéos pour identifier des objets, des visages, ou même des écritures humaines.

Écrite en C++, elle propose des interfaces en C, Python et Java, et est compatible avec les systèmes d'exploitation tels que Windows, Linux, MacOS, iOS et Android.

Conçue pour des applications en temps réel, OpenCV met l'accent sur l'efficacité computationnelle. Elle offre une large gamme de fonctionnalités pour le traitement d'images, incluant la détection d'objets, l'amélioration d'images, la reconnaissance faciale, la détection de mouvements, et la réalité augmentée. Utilisée dans divers domaines, OpenCV est essentielle pour la reconnaissance faciale, la détection d'objets, la vision par ordinateur, et l'apprentissage automatique.



Figure 59 OpenCV

1.4 Organigramme de la détection :

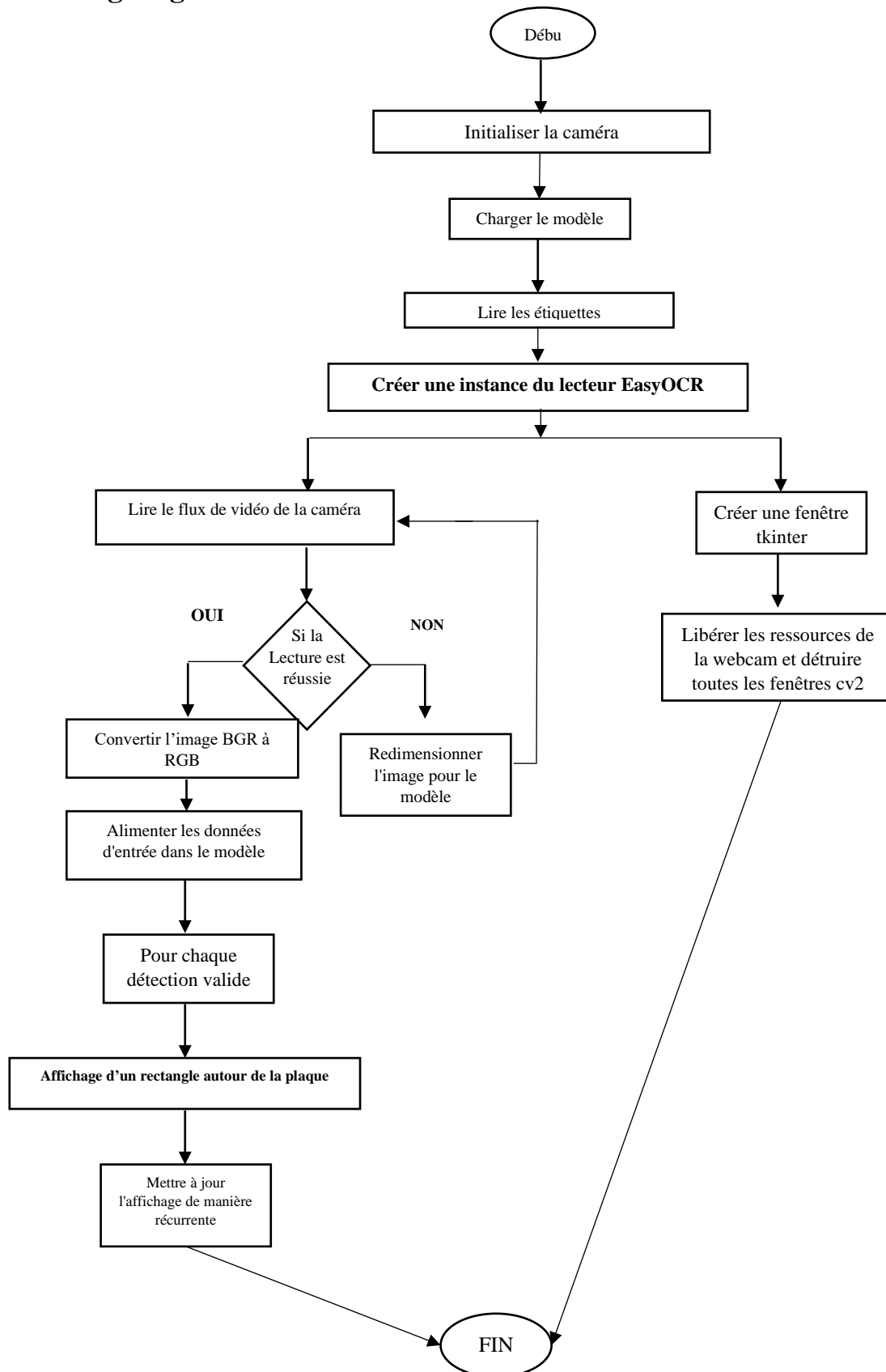


Figure 60 Organigramme de détection de la plaque

1.4.1 Importation des bibliothèques :

```
import numpy as np
from PIL import Image, ImageTk
import easyocr
import cv2
from tensorflow.lite.python.interpreter import Interpreter
import tkinter as tk
```

Figure 61 Importation des bibliothèques

- Numpy : bibliothèque pour les opérations numériques.
- PIL : bibliothèque pour le traitement d'images.
- Easyocr : bibliothèque pour la reconnaissance optique de caractères.
- Cv2 : OpenCV, bibliothèque pour le traitement d'images et l'utilisation de la webcam.
- Tensorflow.lite : pour utiliser TensorFlow Lite.
- Tkinter : bibliothèque pour créer des interfaces graphiques.

1.4.2 Chemins des fichiers et initialisation :

```
modelpath = 'C:\\Users\\PC\\Downloads\\Licence-Plate-Detection-using-TensorFlow-Lite-main\\Licence-Plate-Detect
lblpath = 'C:\\Users\\PC\\Downloads\\Licence-Plate-Detection-using-TensorFlow-Lite-main\\Licence-Plate-Detectio
min_conf = 0.5
```

Figure 62 Chemins des fichiers et initialisation

- Modelpath : chemin du modèle TensorFlow Lite pour la détection des plaques.
- Lblpath : chemin du fichier de labels pour les classes détectées.
- Min_conf : seuil de confiance minimale pour considérer une détection valide.

1. Initialisation de la caméra :

```
cap = cv2.VideoCapture(0)
```

Figure 63 Initialisation de la caméra

- cap : capture vidéo depuis la caméra (indice 0).

1.4.3 Configuration du modèle TensorFlow Lite :

```
interpreter = Interpreter(model_path=modelpath)
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
height = input_details[0]['shape'][1]
width = input_details[0]['shape'][2]

float_input = (input_details[0]['dtype'] == np.float32)

input_mean = 127.5
input_std = 127.5
```

Figure 64 Configuration du modèle TensorFlow Lite

- Interpreter : instance de l'interpréteur TensorFlow Lite.
- Allocate_tensors : allocation des tenseurs pour l'interpréteur.
- Input_details, output_details : détails sur les entrées et sorties du modèle.
- Height, width : dimensions d'entrée du modèle.
- Float_input : vérification si le modèle accepte les entrées de type float32.
- Input_mean, input_std : valeurs de normalisation pour les entrées du modèle.

1.4.4 Lecture des labels :

```
with open(lblpath, 'r') as f:
    labels = [line.strip() for line in f.readlines()]
```

Figure 65 Lecture des labels

- Labels : liste des labels des classes, lue à partir du fichier lblpath.

1.4.5 Initialisation d'EasyOCR :

```
reader = easyocr.Reader(['en'])
```

Figure 66 Initialisation d'EasyOCR

- Reader : instance d'EasyOCR configurée pour lire l'anglais.

1.4.6 Fonction de détection et d'affichage :

```

def detect_and_display():
    ret, frame = cap.read()
    if not ret:
        print("Erreur : la webcam n'a pas pu être lue.")
        root.after(10, detect_and_display)
        return

    image_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    imH, imW, _ = frame.shape
    image_resized = np.array(Image.fromarray(image_rgb).resize((width, height)))
    input_data = np.expand_dims(image_resized, axis=0)

    if float_input:
        input_data = (np.float32(input_data) - input_mean) / input_std

    interpreter.set_tensor(input_details[0]['index'], input_data)
    interpreter.invoke()

    boxes = interpreter.get_tensor(output_details[1]['index'])[0]
    classes = interpreter.get_tensor(output_details[3]['index'])[0]
    scores = interpreter.get_tensor(output_details[0]['index'])[0]

    for i in range(len(scores)):
        if (scores[i] > min_conf) and (scores[i] <= 1.0):
            ymin = int(max(1, (boxes[i][0] * imH)))
            xmin = int(max(1, (boxes[i][1] * imW)))
            ymax = int(min(imH, (boxes[i][2] * imH)))
            xmax = int(min(imW, (boxes[i][3] * imW)))

            roi = frame[ymin:ymax, xmin:xmax]
            result = reader.readtext(roi)
            plate_text = result[0][-2] if result else ''
            print(f"Detected license plate: {plate_text.strip()}")

            plate_numbers = ''.join(filter(str.isdigit, plate_text))
            print(f"Extracted numbers: {plate_numbers}")

            cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (10, 255, 0), 2)
            cv2.putText(frame, f'Labels[{int(classes[i])}]: {int(scores[i] * 100)}%', (xmin, ymin - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0))
            cv2.putText(frame, plate_text.strip(), (xmin, ymin - 25), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0))

    img = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
    imgtk = ImageTk.PhotoImage(image=img)
    lmain.imgtk = imgtk
    lmain.configure(image=imgtk)
    root.after(10, detect_and_display)

```

Figure 67 Fonction de détection et d'affichage

- ret, frame : lecture d'une image depuis la webcam.
- image_rgb : conversion de l'image en RGB.
- imH, imW, _ : dimensions de l'image.
- image_resized : redimensionnement de l'image aux dimensions d'entrée du modèle.
- input_data : préparation des données d'entrée pour le modèle.

- `interpreter.invoke()` : exécution du modèle.
- `boxes`, `classes`, `scores` : extraction des boîtes de délimitation, classes et scores de confiance.
- Boucle `for` : traitement de chaque détection valide (score supérieur à `min_conf`).
- `roi` : région d'intérêt de l'image pour la reconnaissance de texte.
- `result` : résultats de la reconnaissance de texte par EasyOCR.
- `plate_text` : texte de la plaque d'immatriculation détectée.
- `plate_numbers` : extraction des chiffres du texte détecté.
- `cv2.rectangle` : dessin de la boîte de délimitation autour de la plaque détectée.
- `cv2.putText` : ajout de texte sur l'image.
- `img` : conversion de l'image pour Tkinter.
- `imgtk` : création d'une image Tkinter.
- `lmain.imgtk` et `lmain.configure` : mise à jour de l'image affichée dans l'interface.

1.4.7 Configuration de l'interface graphique Tkinter :

```
root = tk.Tk()
lmain = tk.Label(root)
lmain.pack()
root.title("License Plate Detection")

root.after(10, detect_and_display)
root.mainloop()
```

Figure 68 Configuration de l'interface graphique Tkinter

- `root` : création de la fenêtre principale Tkinter.
- `lmain` : label Tkinter pour afficher l'image.
- `root.after` : appel périodique de la fonction `detect_and_display`.
- `root.mainloop` : boucle principale Tkinter.

2. Libération des ressources :

```
cap.release()
cv2.destroyAllWindows()
```

Figure 69 Libération des ressources

- `cap.release()` : libération de la webcam.
- `cv2.destroyAllWindows()` : fermeture des fenêtres OpenCV.

1.5 Jupyter Notebook :

Jupyter Notebook est une application web open source conçue pour simplifier la création et le partage de documents intégrant du code actif, des équations, des graphiques et des explications textuelles. Son utilisation est répandue dans divers domaines comme le traitement de données, la modélisation statistique, ou encore l'apprentissage automatique. Grâce à ses noyaux spécifiques, il est compatible avec plusieurs langages de programmation tels que Python, R, Julia et Scala. Son nom résulte de la combinaison des noms des trois langages initialement supportés : Julia, Python et R. Les Jupyter Notebooks adoptent une structure cellulaire, permettant d'inclure du code exécutable, du texte en format Markdown ou brut. Cette organisation cellulaire favorise l'interactivité et garantit la reproductibilité des recherches scientifiques.



Figure 70 Jupyter Notebook

2 Architecture du modèle CNN :

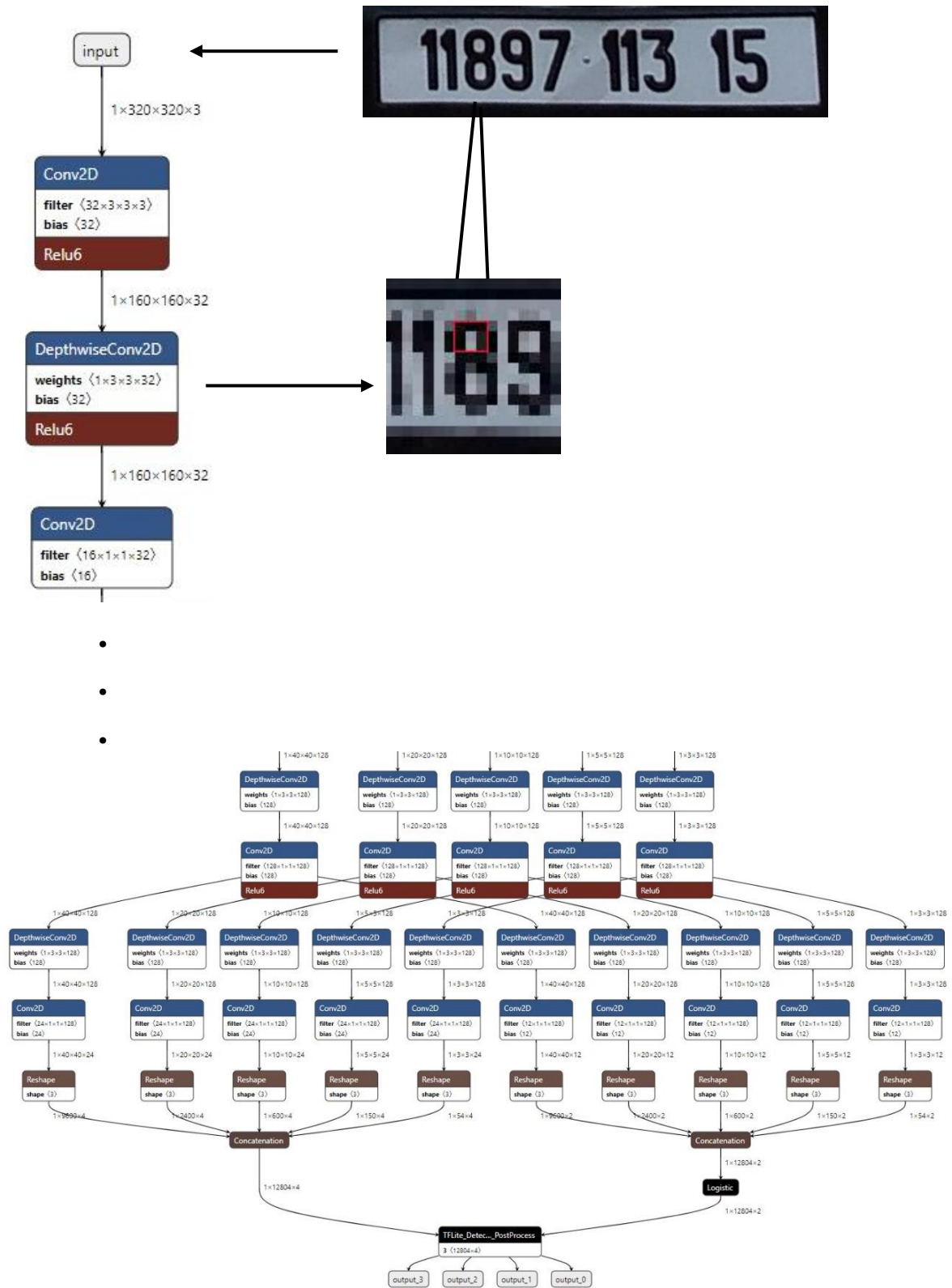


Figure 71 Architecture du modèle CNN

2.1 Extraction de chiffres avec Easyocr :

2.1.1 Importer la bibliothèque Easyocr :

```
import easyocr
import cv2
```

Figure 72 Importer la bibliothèque Easyocr

2.1.2 Initialisation de Easyocr :

```
reader = easyocr.Reader(['en'])
```

Figure 73 Initialisation de Easyocr

2.1.3 Fonction de

Détection et Reconnaissance de Texte :

Cette fonction utilise Easyocr pour lire le texte des régions d'intérêt extraites des images capturées par la caméra.

3 Utilisation de la carte Raspberry Pi :

3.1 Le système Raspbian :

Raspbian est un système d'exploitation libre et open source dérivée de Debian, spécifiquement créé pour les cartes Raspberry Pi. C'est la distribution Linux la plus prisée et conseillée pour les utilisateurs de Raspberry Pi. Raspbian propose un environnement de bureau convivial avec de nombreuses applications préinstallées et offre un support exceptionnel pour le développement logiciel. Il est optimisé pour exploiter pleinement les capacités matérielles du Raspberry Pi, y compris son processeur ARM et son GPU.

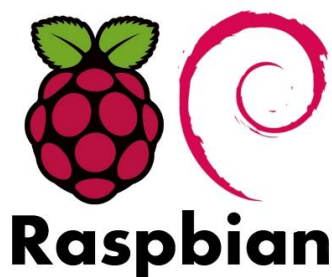


Figure 74 Le système Raspbian

3.2 Installer Raspbian et connecter la carte à un PC via un câble RJ45 :

Nous avons besoin du matériel suivant pour installer Raspbian sur une carte Raspberry Pi et la connecter à un PC via un câble Ethernet (RJ45) :

- Une carte microSD (minimum 8 Go recommandée).
- Un ordinateur avec un lecteur de carte microSD.
- Un Raspberry Pi.
- Un câble Ethernet (RJ45).
- Un adaptateur secteur pour votre Raspberry Pi.

Étapes d'installation :

1. Télécharger l'image de Raspbian :

- Aller sur le site officiel de Raspberry Pi à l'adresse suivante : raspberrypi.org/downloads.
- Télécharger l'image de Raspberry Pi OS.

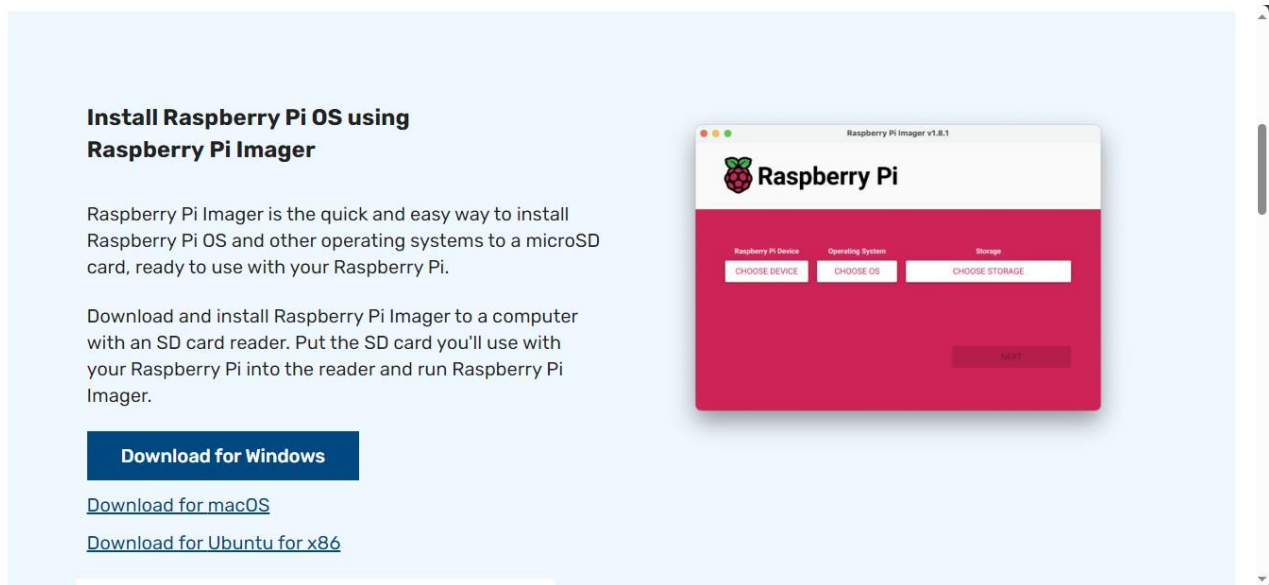


Figure 75 Téléchargement de l'image de Raspberry Pi OS

2. Télécharger et installer Raspberry Pi Imager :

- Télécharger Raspberry Pi Imager pour le système d'exploitation depuis : raspberrypi.org/software.

- Installer Raspberry Pi Imager sur le PC.



Figure 76 Installation de Raspberry Pi Imager

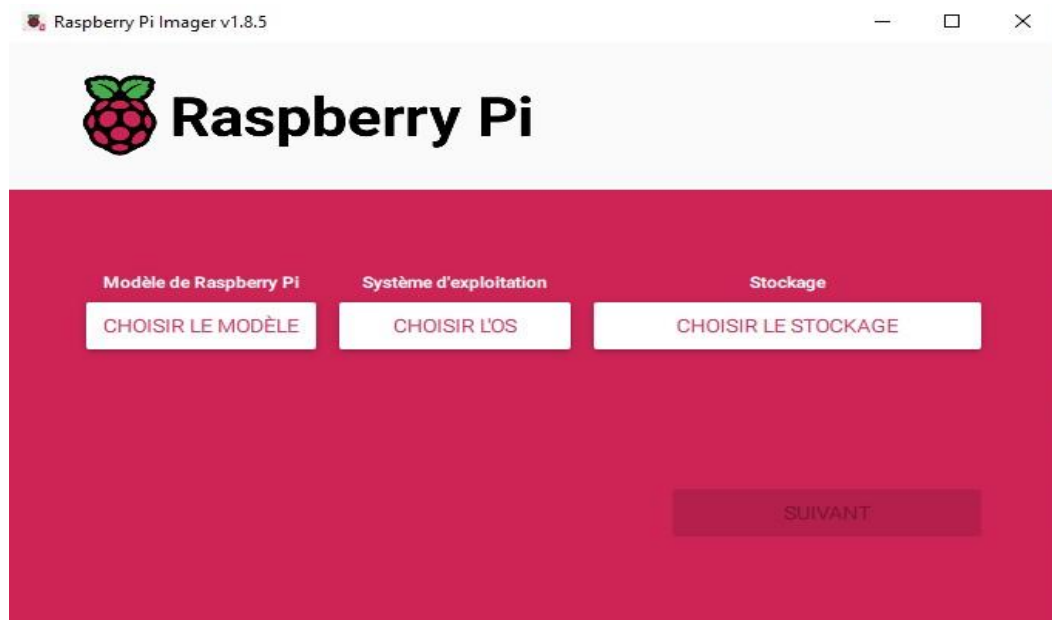


Figure 77 Raspberry pi Imager

3. Préparer la carte microSD :

- Insérer la carte microSD dans le lecteur de carte de l'ordinateur.
- Ouvrir Raspberry Pi Imager.
- Cliquer sur "Choose OS" et sélectionner "Raspberry Pi OS".
- Cliquer sur "Choose SD Card" et sélectionner votre carte microSD.
- Cliquer sur "Write" pour écrire l'image sur la carte microSD. Cela peut prendre quelques minutes.

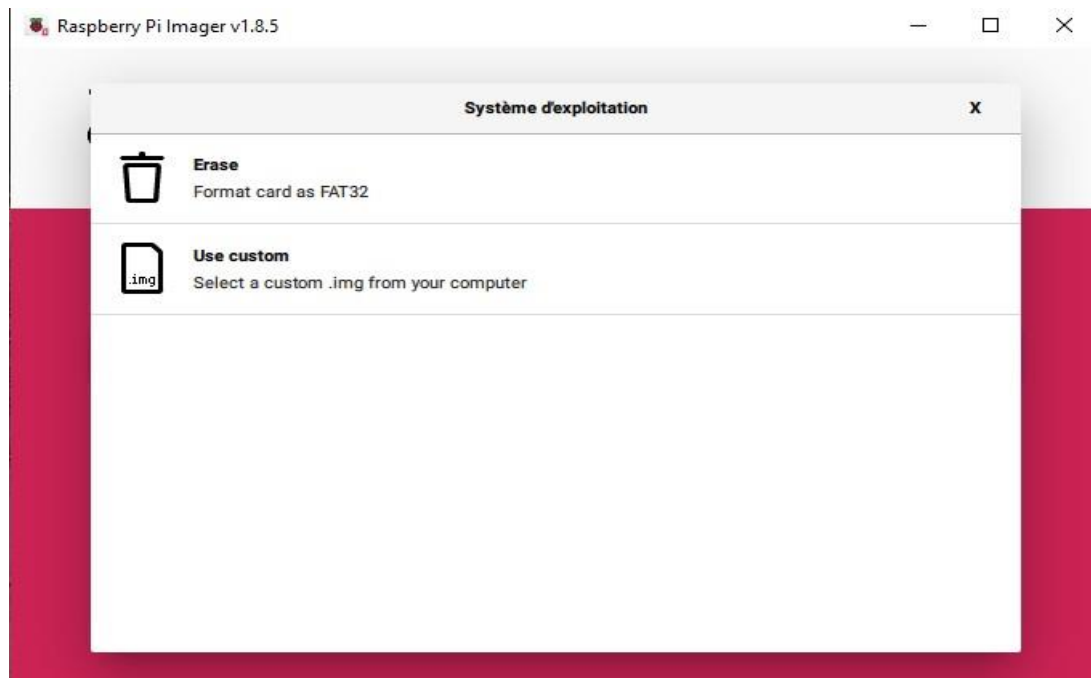


Figure 78 Choix du système d'exploitation dans Raspberry pi Imager

4. Configuration de l'accès SSH :

- Une fois l'image écrite, ouvrir la partition boot de la carte microSD sur le PC.
- Créer un fichier vide nommé "ssh" (sans extension) dans la partition boot pour activer SSH.

5. Configuration de l'interface réseau :

- Pour connecter directement le Raspberry Pi au PC via Ethernet, il faut configurer une adresse IP statique. Créer un fichier nommé "wpa_supplicant.conf" dans la partition boot et ajouter les informations suivantes :

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<Your_Country_Code>
network={
    ssid="<Your_SSID>"
    psk="<Your_Password>"
}
```

Figure 79 Configuration de l'interface réseau

Démarrer le Raspberry Pi :

- Insérer la carte microSD dans le Raspberry Pi.
- Connecter le Raspberry Pi au PC via le câble Ethernet.
- Brancher l'alimentation pour démarrer le Raspberry Pi.

Trouver l'adresse IP du Raspberry Pi :

- Sur le PC, ouvrir une fenêtre de terminal (ou l'invite de commande sur Windows).
- Utiliser la commande suivante pour trouver l'adresse IP de la Raspberry Pi :

```
arp -a
```

Figure 80 Commande de l'adresse IP de la Raspberry Pi

Se connecter au Raspberry Pi via SSH :

- Une fois l'adresse IP trouvée, nous allons utiliser SSH pour nous connecter :

```
ssh pi@<Adresse_IP_du_Raspberry_Pi>
```

Figure 81 Connexion à la Raspberry Pi via SSH

Configuration initiale et mises à jour :

- Lors de la première connexion, nous sommes invités à changer le mot de passe par défaut pour des raisons de sécurité.
- Mettre à jour le système en exécutant les commandes suivantes :

```
sudo apt update
sudo apt full-upgrade
```

Figure 82 Mise à jour du système

Nous avons maintenant installé Raspbian sur votre carte Raspberry Pi et configuré une connexion via un câble RJ45 à votre PC. Vous pouvez désormais utiliser SSH pour interagir avec votre Raspberry Pi directement depuis votre ordinateur.

3.3 Affichage de l'interface Raspbian sur PC via RJ45 :

1. Installation de VNC Server :

- Ouvrir un terminal sur la Raspberry Pi ou se connecter à distance via SSH.
- Exécuter la commande suivante pour installer VNC Server :

```
sudo apt update
sudo apt install realvnc-vnc-server
```

Figure 83 Commande d'affiche de l'interface Raspbian

2. Configuration de VNC Server :

- Une fois installé, nous exécutons la commande suivante pour démarrer le programme de configuration de VNC Server :

```
vncserver
```

Figure 84 Commande de configuration de VNC server

3. Configuration sur le PC :

Télécharger le client VNC :

- Télécharger et installer un client VNC sur le PC. RealVNC propose un client gratuit, mais il existe d'autres options telles que TightVNC ou TigerVNC.



Figure 85 Interface de VNC

Se connecter à la Raspberry Pi :

- Ouvrir le client VNC sur le PC.
- Entrer l'adresse IP de la Raspberry Pi suivie du numéro de port VNC par défaut, généralement :1 (par exemple, 192.168.1.100:1).
- Entrer le mot de passe défini lors de la configuration de VNC Server sur la Raspberry Pi.

Accès à l'interface graphique :

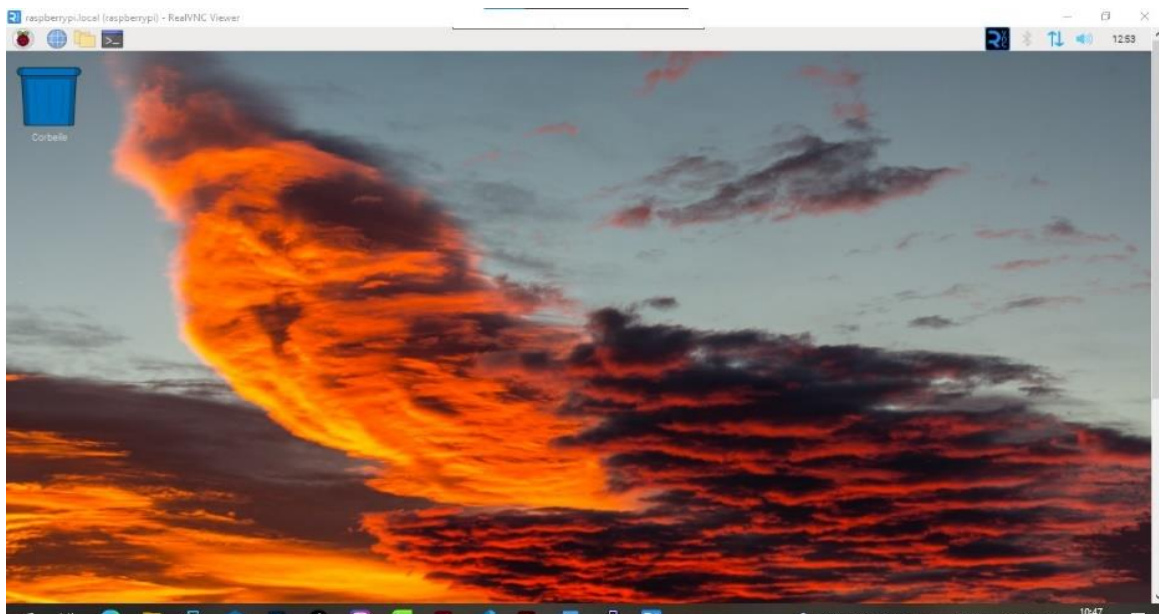


Figure 86 Interface Graphique de Raspbian

- Une fois connecté, nous avons le contrôle de l'interface graphique de la Raspberry Pi directement sur le PC.

3.4 Connexion de la carte Arduino à la carte Raspberry pi :

1. Première Etape :

Une fois la carte Raspberry Pi connectée, on peut connecter la carte Arduino avec un câble USB de type A à la carte Raspberry.

2. Deuxième Etape :

Installer la bibliothèque PySerial sur l'Arduino pour pouvoir communiquer entre les deux appareils en bidirectionnelle.

3. Troisième Etape :

Initialiser la communication entre la carte Arduino et la Raspberry pi 3.

```
import serial # Bibliothèque pour la communication série avec Arduino

# Initialisation de la connexion série avec l'Arduino
ser = serial.Serial('COM6', 9600) # Adapter le port série en fonction de votre configuration
```

Figure 87 Code d'initialisation de la communication entre Arduino et Raspberry pi

3.4.1 Configuration de la carte Arduino :

```

#include <Servo.h>

const int IR_PIN = A0; // Broche analogique où le capteur infrarouge est connecté
const int SERVO_PIN = 9; // Broche où le servomoteur est connecté
|
Servo servoMotor;

void setup() {
  Serial.begin(9600); // Initialiser la communication série
  servoMotor.attach(SERVO_PIN); // Attacher le servomoteur au pin
  servoMotor.write(0); // Mettre le servomoteur à 0 degrés au démarrage
}

void loop() {
  int irValue = analogRead(IR_PIN); // Lire la valeur du capteur infrarouge

  // Si le capteur détecte un objet à proximité, envoyer un signal
  if (irValue > 500) { // Ajustez cette valeur selon votre capteur et votre environnement
    Serial.println("Infrarouge déclenché");
    delay(1000); // Attendre un court instant pour éviter les rebonds du capteur
  }

  if (Serial.available() > 0) {
    char signal = Serial.read();
    if (signal == '1') {
      activateServo();
    }
  }
}

void activateServo() {
  servoMotor.write(90); // Déplacer le servomoteur à 90 degrés (position médiane)
  delay(100000); // Attendre un court instant
  servoMotor.write(0); // Déplacer le servomoteur à 0 degré (position de départ)
  delay(1000); // Attendre un court instant pour assurer le retour à la position initiale
}

```

Figure 88 Code de configuration de la carte Arduino

4 Les systèmes de paiements :

Il existe différents moyens de paiement des Parkings, qui évoluent constamment avec les nouvelles technologies voici quelques moyens déjà existants :

- Moyens traditionnels :
 - Le paiement avec de la monnaie.
 - Le paiement avec des cartes de crédits.
- Bornes de paiement automatique.
- Des Applications mobiles.
- Paiement avec carte sans contact :
 - Carte RFID.
 - Paiement mobile.

- Système intelligents et connecté :
 - Parkings intelligent.
 - Intégration avec les villes intelligentes.
- Système de paiement avec code QR.
- Paiement avec reconnaissance faciale.
- Paiement avec plaques d'immatriculation.

5 Plateforme de paiement élaborée :

Puisque nous avons opté pour le paiement avec plaques d'immatriculations, nous avons créé un site web pour effectuer le paiement, une fois que le véhicule à quitter le parking le montant est déduit automatiquement.

5.1 Architecture du site web :

5.1.1 Page d'accueil :

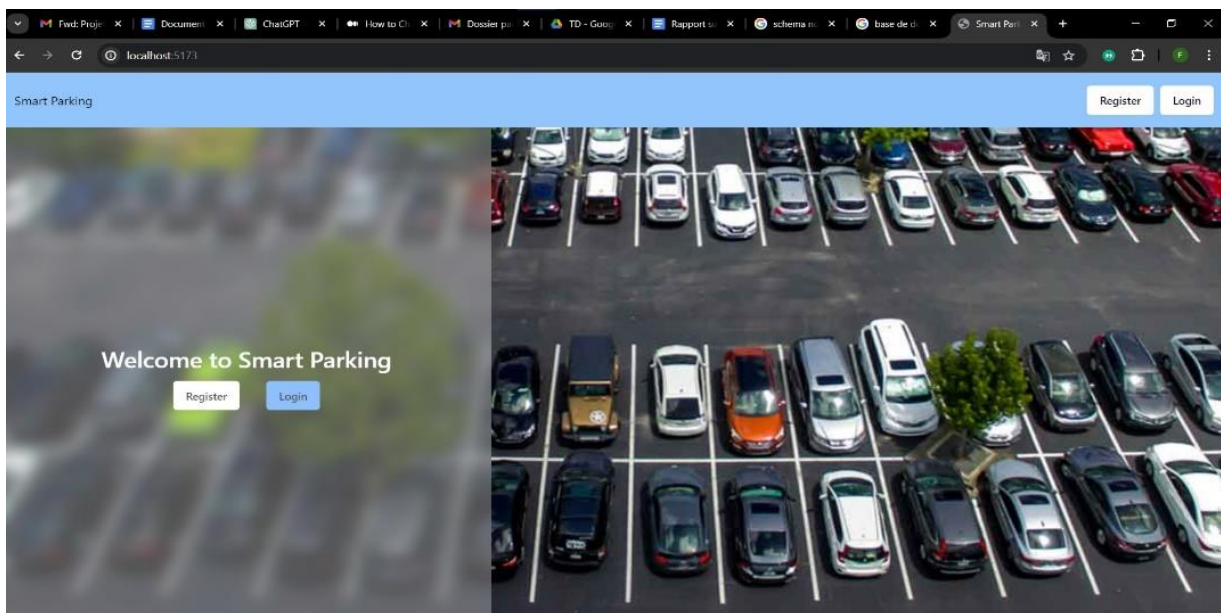


Figure 89 Page d'accueil du site web

5.1.2 Page d'inscription et de connexion :

5.1.2.1 Inscription :

- Accéder à la page principale
- Remplir le formulaire avec les informations demandées (Nom complet, email, numéro de téléphone, mot de passe)
- Valider le formulaire pour créer un compte

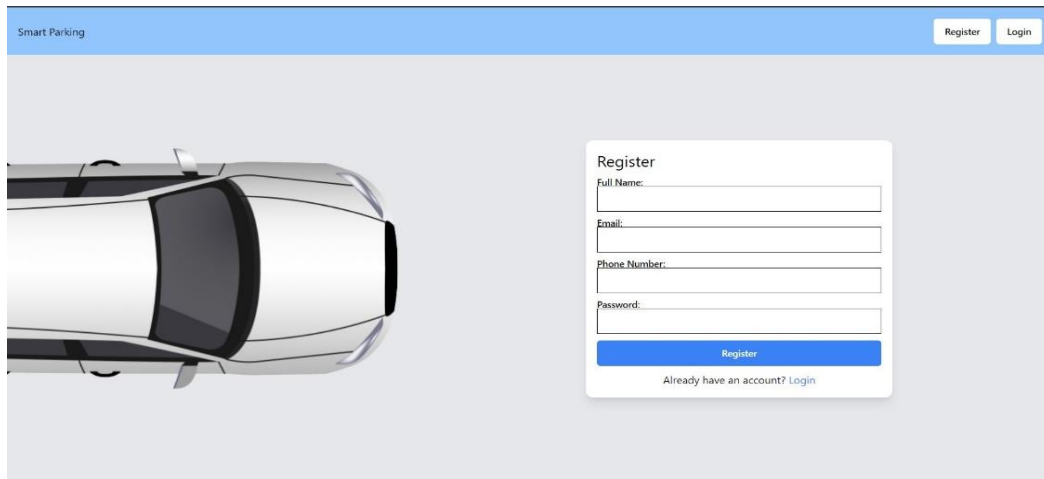


Figure 90 Page d'inscription

5.1.2.2 Connexion :

- Cliquer sur le bouton Login
- Entrer l'email et le mot de passe

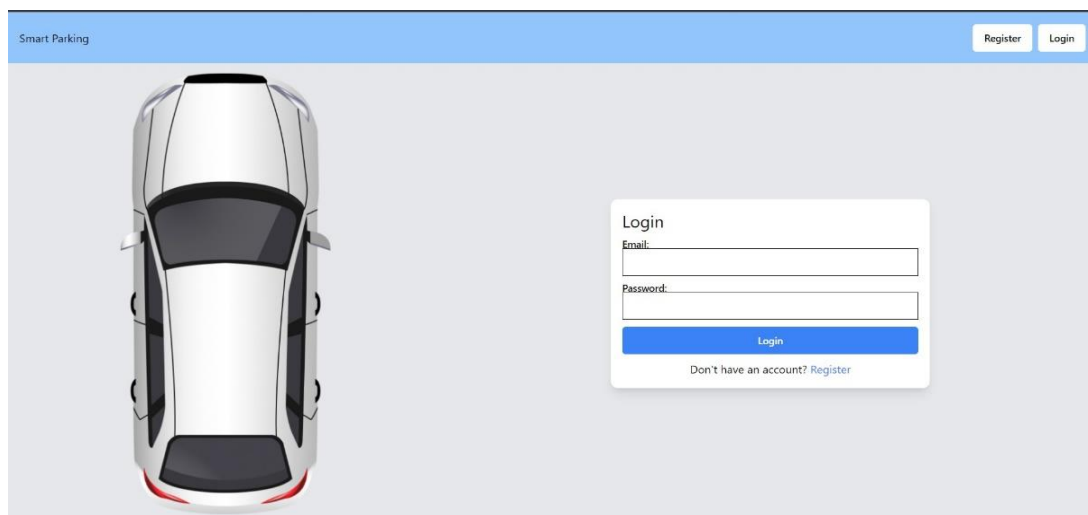


Figure 91 Page de connexion

Une fois toutes les informations sont introduites nous serons dirigés vers la page d'accueil.

5.1.3 Page de gestion de factures :

Pour consulter ses factures et régler les factures impayées

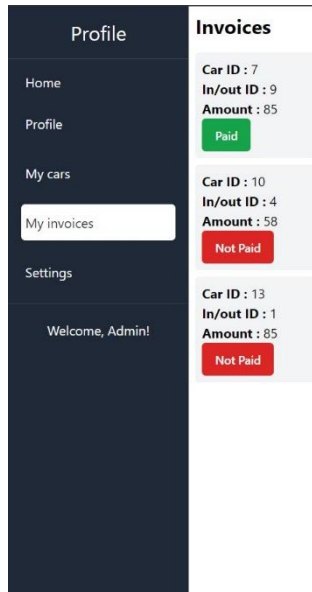


Figure 92 Page de gestion de factures

5.2 Base de données :

5.2.1 Création de Base de Données :

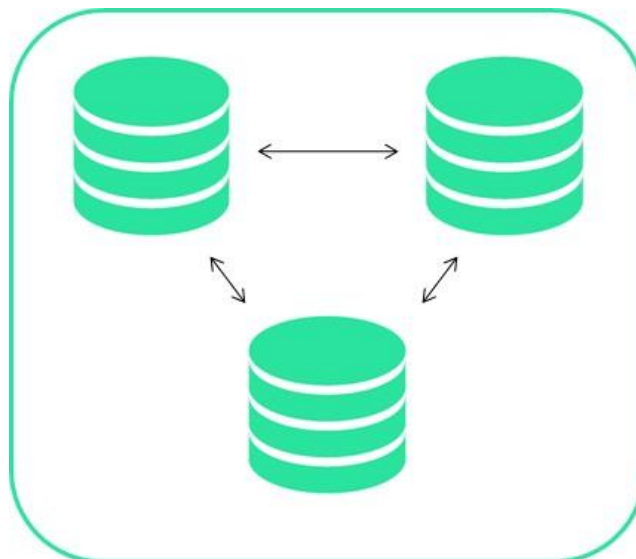


Figure 93 Base de données

L'objectif de ce rapport est de décrire l'architecture de la base de données distribuée mise en place pour notre application. Nous expliquerons les choix de conception, les schémas de données utilisés, ainsi que les stratégies de répartition des données.

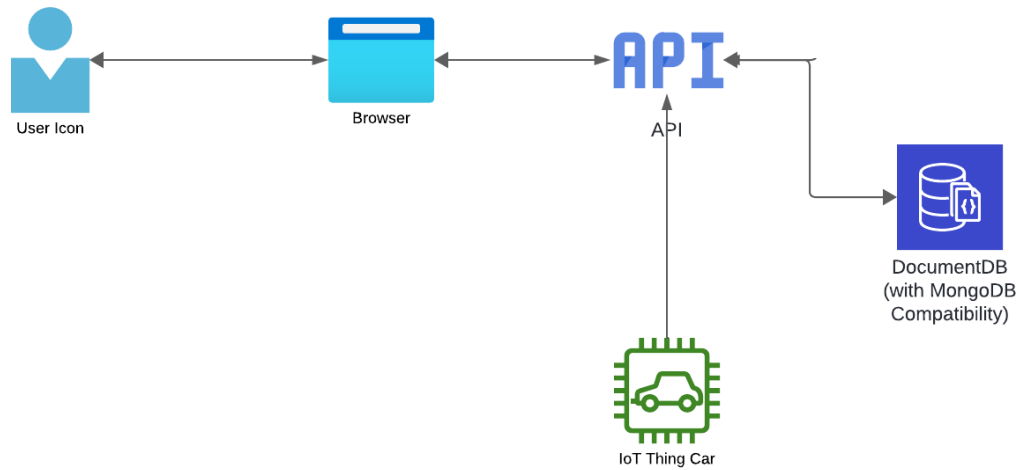


Figure 94 Schéma synoptique de la base de données

5.2.2 Schémas de Données

Les schémas de données comprennent des entités telles qu'Utilisateur, Facture, Entree_Sortie et Véhicule. Chaque entité est associée à ses attributs respectifs pour représenter les informations nécessaires à notre application.

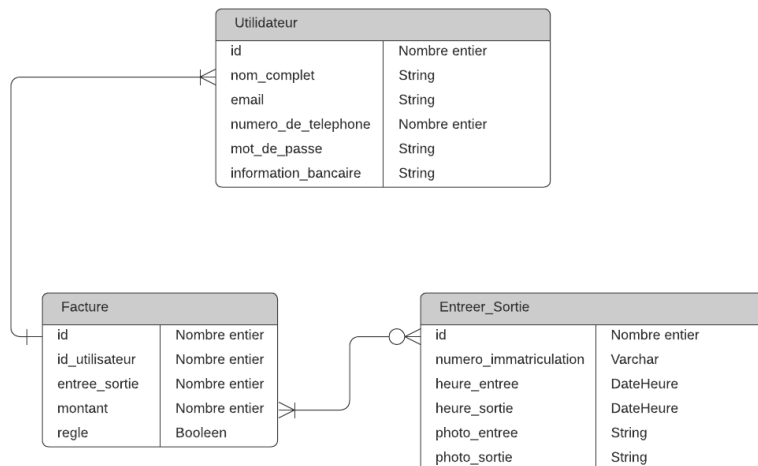


Figure 95 Schéma de données

5.2.3 Sécurité :

Des mesures de sécurité sont mises en place pour protéger les données sensibles et limiter l'accès non autorisé. Les mécanismes de sécurité intégrés à MongoDB sont utilisés pour renforcer la sécurité de la base de données distribuée.

- L'abstraction des coordonnées sur le fichier. Env inaccessible de l'extérieur.
- La gestion de l'authentification par le http only cookies assurant la non accessibilité à travers les scripts et les attaques XSS.
- Mise en place d'une politique de sécurité et de validation de login pour éviter l'accès non permis des sources externes.
- Définition des rôles (Admin)
- Hachage des données sensibles (SHA256 pour les mots de passe)

5.2.4 Diagramme de la base de données :

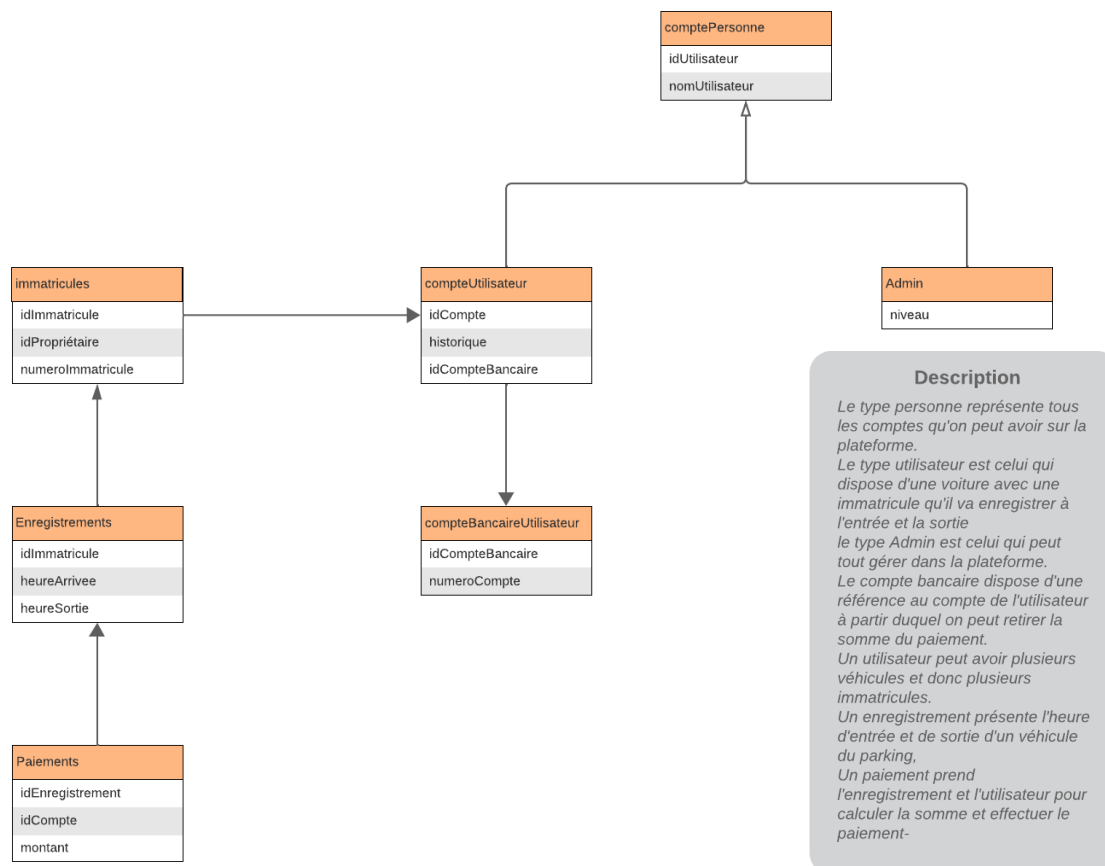


Figure 96 Diagramme fonctionnel de la base de données

5.3 Schéma fonctionnel global :

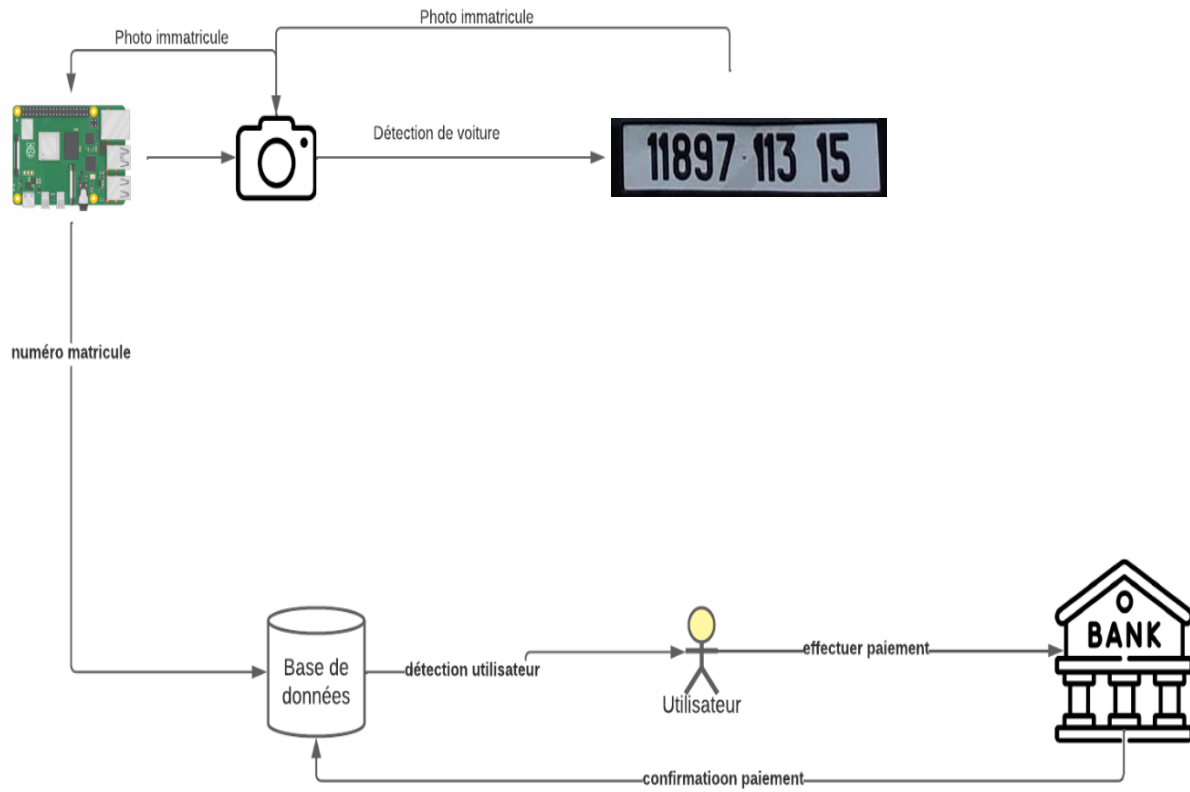


Figure 97 Schéma Global

6 Schéma Final :

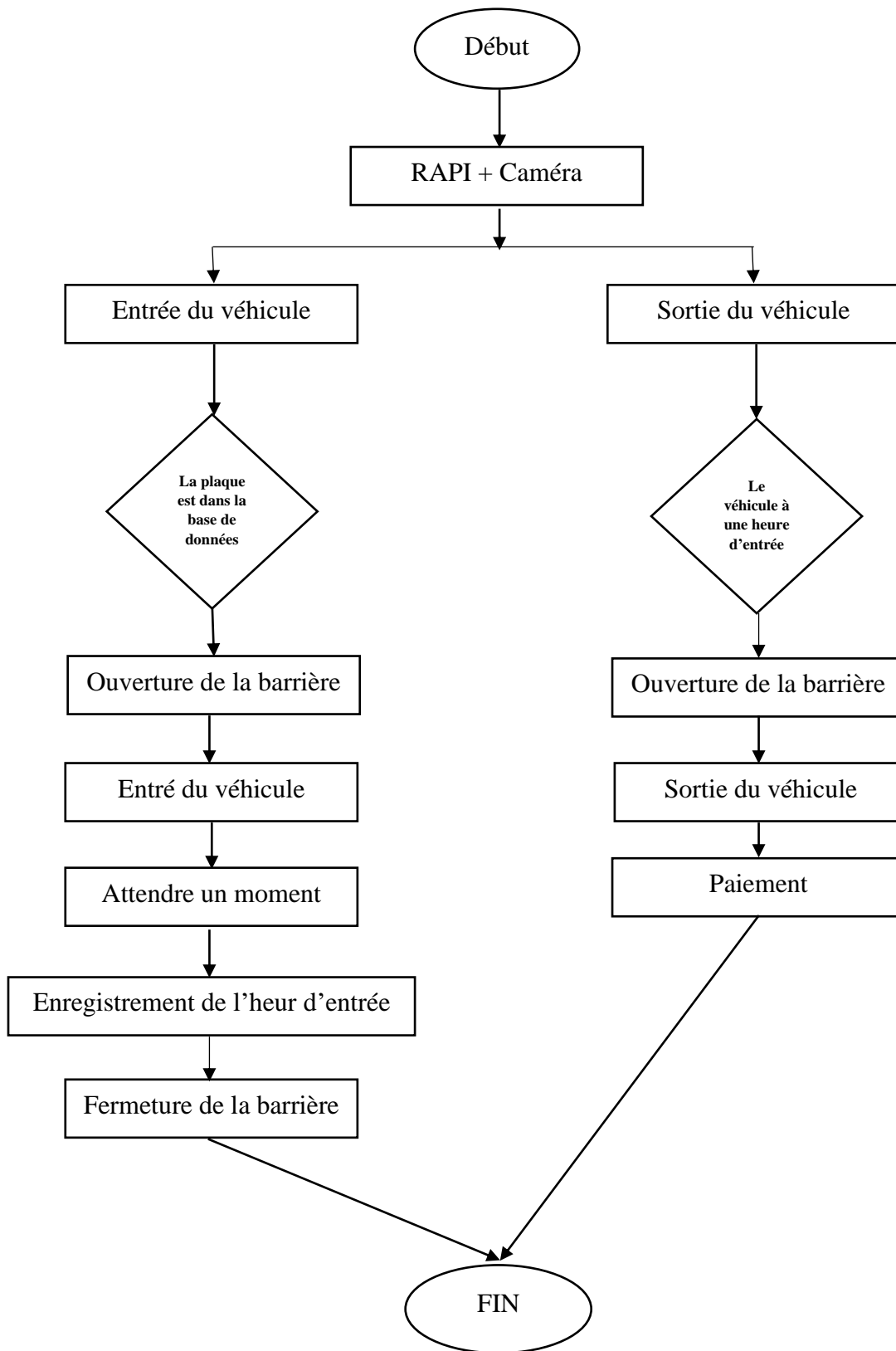


Figure 98 Organigramme fonctionnel du système

7 Branchement de la carte Arduino :

Pour le fonctionnement du système nous allons utiliser un servomoteur qui va jouer le rôle d'une barrière, et un capteur infrarouge pour détecter la présence d'un véhicule, du côté du branchement voici un diagramme fonctionnel qui explique le fonctionnement de ce système :

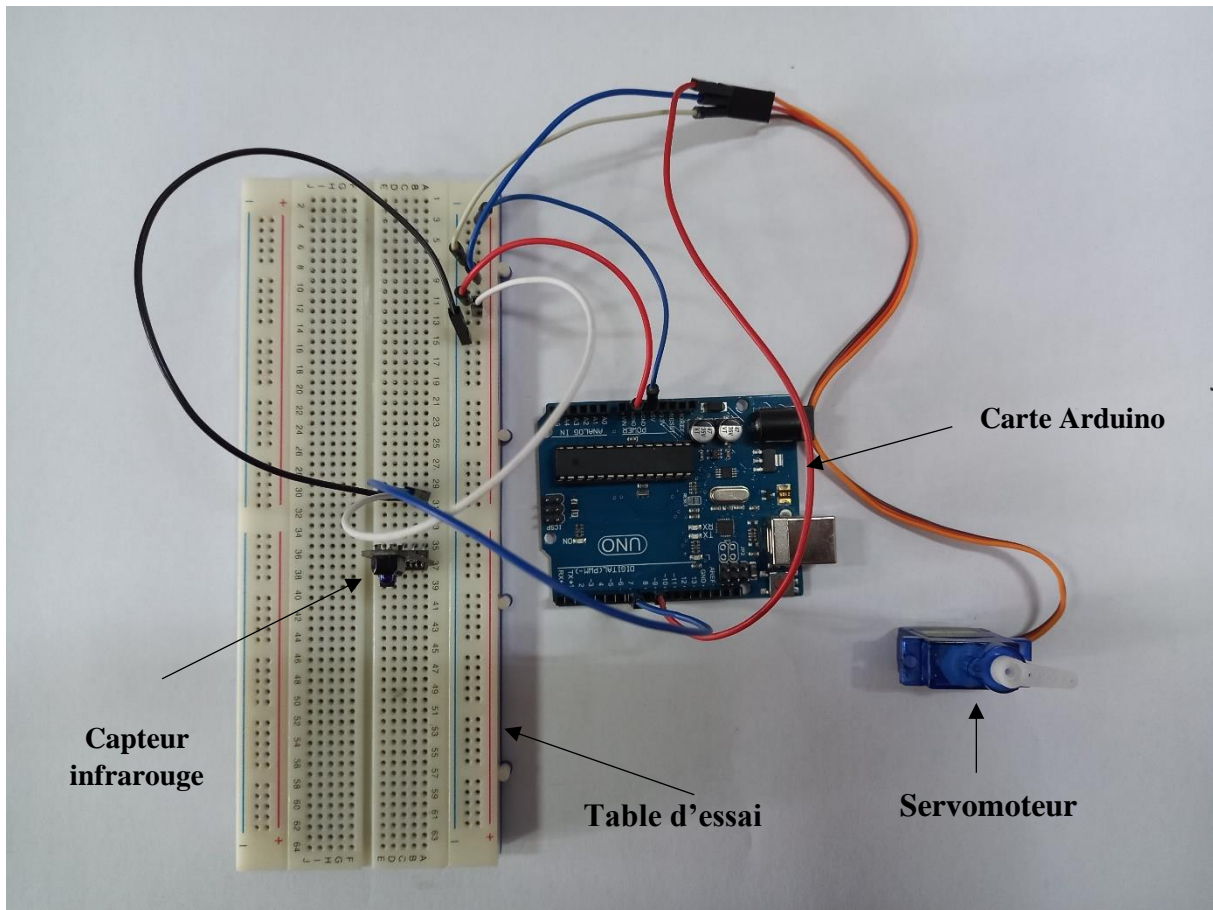


Figure 99 Branchement des composants avec l'Arduino

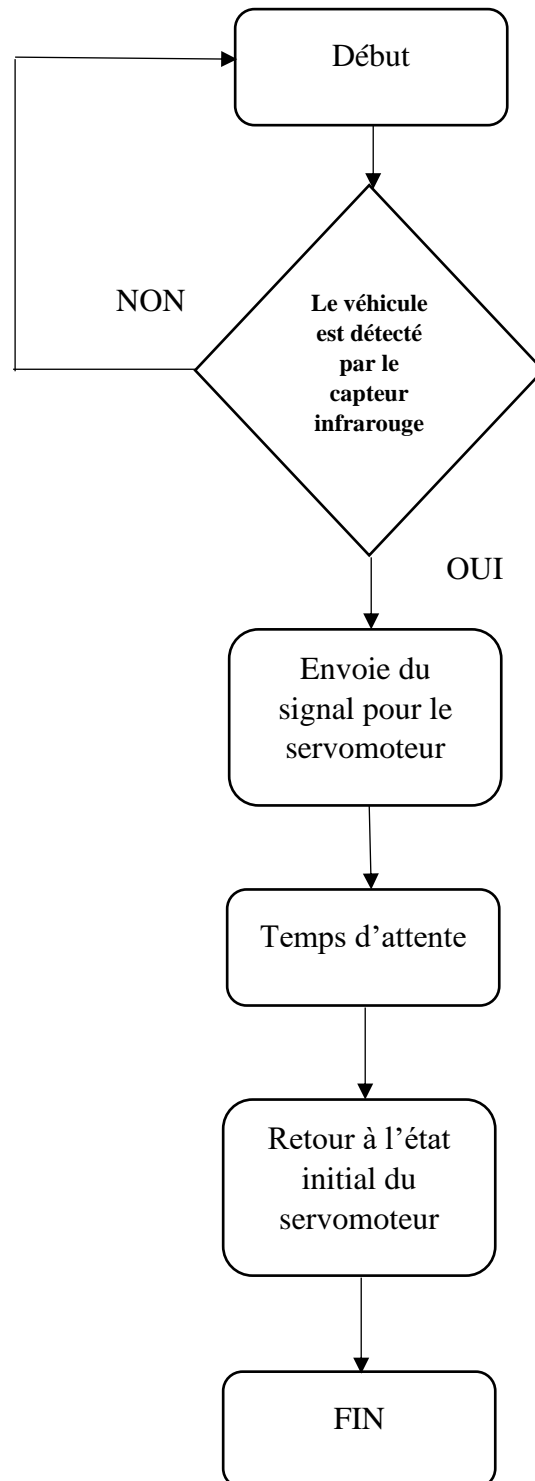


Figure 100 Organigramme du fonctionnement du programme Arduino

8 Connexion de la carte Arduino :

Pour connecter la carte Arduino, nous allons d'abord importer la bibliothèque Serial :

```
import serial
import time
```

Figure 101 importation de la bibliothèque Serial

Puis nous allons initialiser la connexion série avec l'Arduino :

```
ser = serial.Serial('COM6', 9600)
```

Figure 102 Initialisation de la connexion Série

Et pour finir nous allons attendre que la carte Arduino réponde :

```
time.sleep(2)
```

Figure 103 Temps d'attente

9 Tests et résultats :

Pour les résultats nous avons exécuté le code principale et préparé le matériel nécessaire ainsi que le branchement des composants (Capteur infrarouge, servomoteur) avec la carte Arduino puis branché cette dernière à la carte Raspberry Pi3, nous avons ensuite effectué des tests avec la carte Raspberry Pi 3 mais nous avons eu quelques difficultés avec ses performances (Un faible taux d'image par secondes, manque de mémoire vive), donc nous avons eu recours à tester notre système en utilisant également un PC beaucoup plus performant disposant de composant plus robuste (Processeur Ryzen 5, 16 Go ram)

9.1 Résultats de détection et de reconnaissance de la plaque :

Pour nos tests, nous allons utiliser la caméra de notre Pc, et l'image est capturée à partir d'un téléphone.

A chaque itération de la boucle principale, le code va lire une séquence de la vidéo de la webcam, ensuite nous allons utiliser TensorFlow Lite pour le modèle de détection d'objet. Une fois la plaque est détectée nous allons faire appel à EasyOCR pour extraire les chiffres de la plaque.

Pour l'affichage des résultats nous avons affiché la vidéo de la webcam en direct dans une fenêtre Tkinter avec une superposition des rectangles de délimitation autour de la plaque d'immatriculation détectée ainsi que le texte de la plaque.

9.1.1 Reconnaissance et détection :



Figure 104 Détection et reconnaissance de la plaque

Une fois que le téléphone est rapproché à la Webcam nous pouvons voir que la plaque est détectée à 99%, et qu'EasyOCR a pu lire tous les chiffres présents sur la plaque.

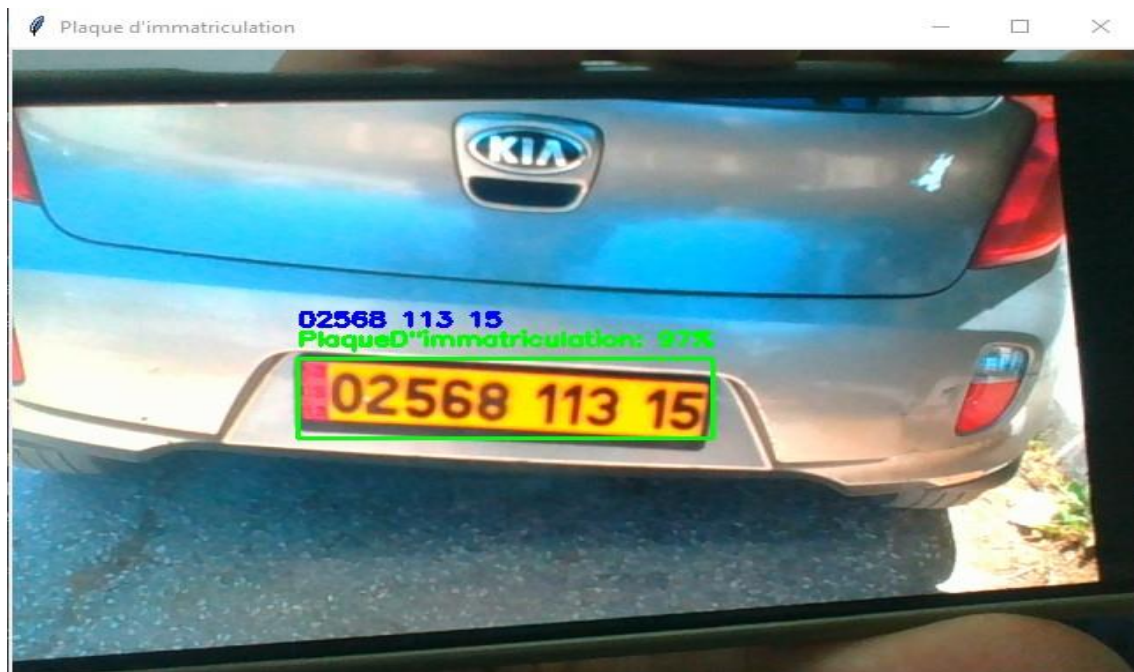


Figure 105 Détection et lecture de la plaque

Dans ce cas la plaque est légèrement floue, et l'image est volontairement inclinée pour tester l'efficacité de la détection mais aussi de la reconnaissance avec EasyOCR, nous avons pu constater que la lecture et la détection de la plaque est effectué avec succès, le pourcentage dans ce cas est de 97%.



Figure 106 Résultats de la reconnaissance et de la détection de la plaque

Ci-dessous un graphe de la détection de la plaque :



Figure 107 Graphe de la détection de la plaque

Dans ce graphe nous avons traduit l'erreur de lecture par un graphe, pour cela nous avons introduit une fonction qui calcule la marge d'erreur entre le texte de référence et le texte détecter.

Dans ce graphe l'axe des x représente le nombre d'instances de détection de la plaque, et l'axe des y représente le pourcentage de confiance de la détection, l'erreur est représentée en rouge et le pourcentage de confiance en bleu, plus la ligne bleue est élevée, plus le modèle est confiant dans sa détection de la plaque d'immatriculation à ce moment-là.

9.2 Erreur de reconnaissance des chiffres de la plaque :

Les erreurs lors de la lecture de la plaque sont dues à plusieurs facteurs, tels qu'une mauvaise visibilité, l'image est soit floue soit elle présente un manque d'éclairage, ou encore une difficulté à reconnaître les chiffres en raison d'une similarité visuelle des caractères, par exemple 2 avec z, 0 avec O, d'autres erreurs sont dues à une instabilité de la caméra.

Ci-dessous quelque testes des erreurs :

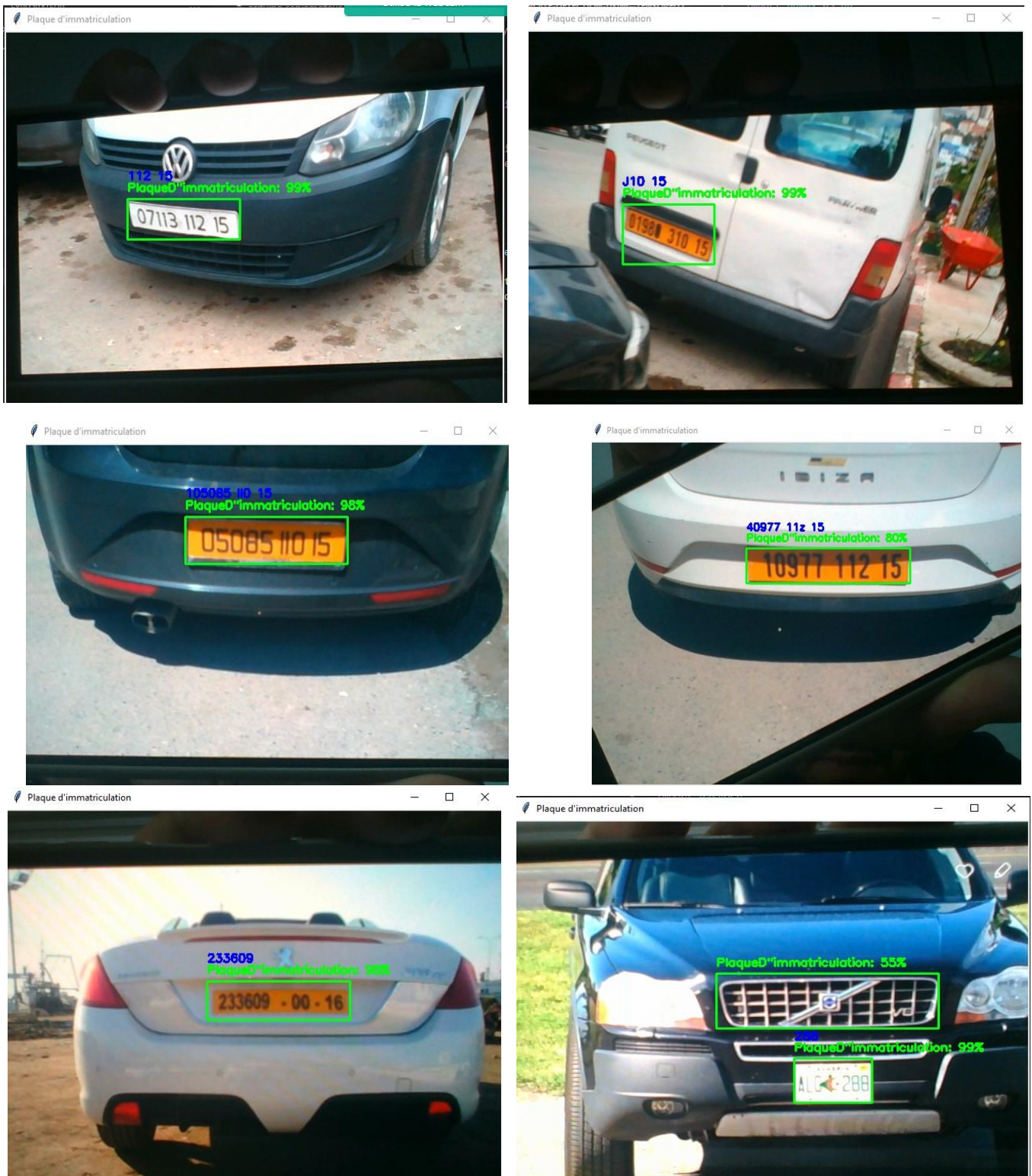


Figure 108 Erreurs de reconnaissance de plaque

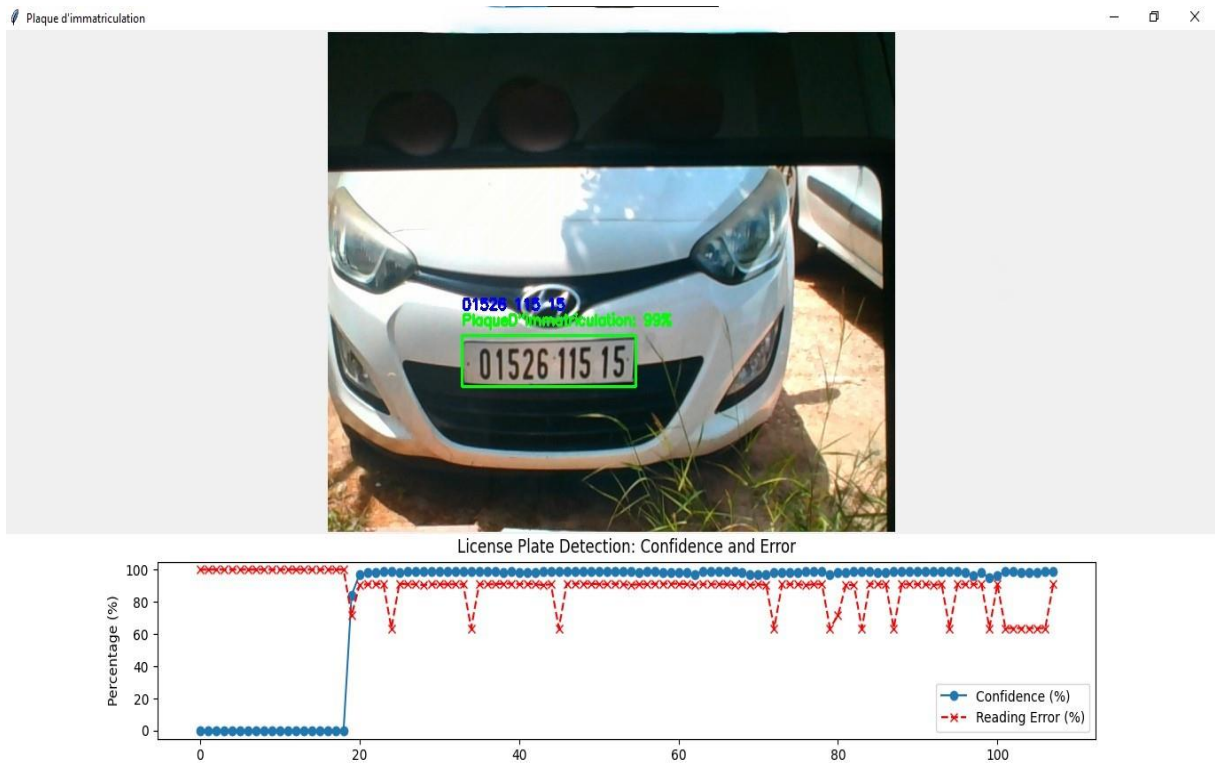


Figure 109 Graphe de confiance et d'erreur

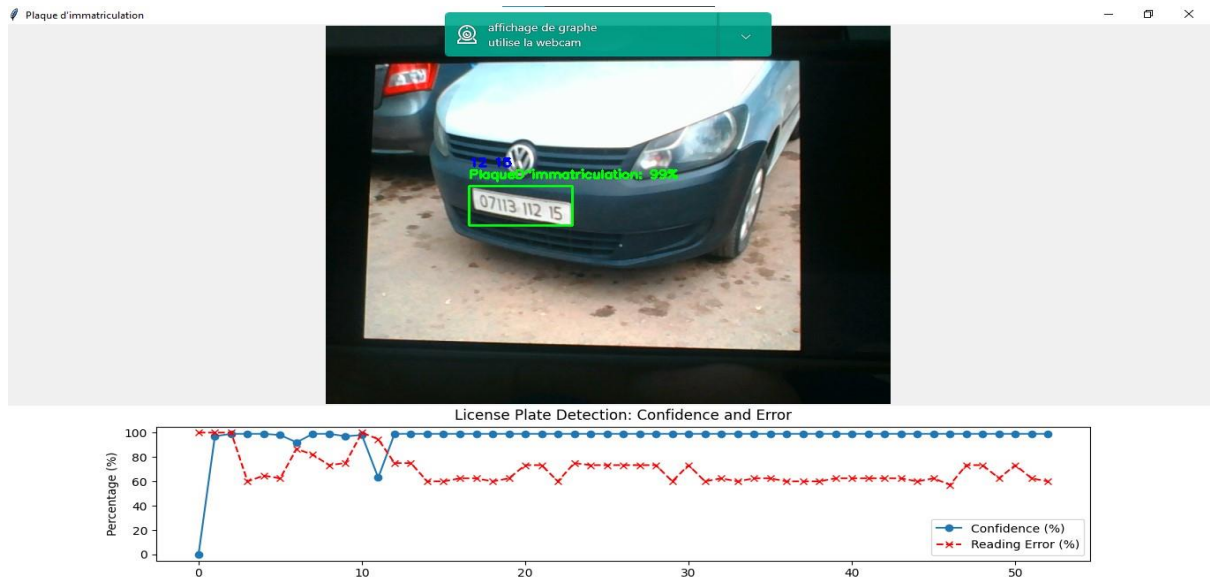


Figure 110 Graphe d'erreur de lecture de la plaque

10 Comparaison entre EasyOCR et Tesseract :

EasyOCR	Tesseract
	
	
	



Tableau III Comparaison entre EasyOCR et Tesseract

D’après ces résultats nous pouvons constater que la méthode avec EasyOCR est plus performante et elle nous offre une meilleure reconnaissance de la plaque par rapport à la méthode avec Tesseract, qui reste bien sur un bon outil pour la reconnaissance, mais ce que nous recherchons est une bonne lecture, puisqu’il s’agit d’un système de paiement donc nous devons minimiser les erreurs au maximum.

11 Résultats du système de paiement :

Comme nous l’avons mentionné dans la partie précédente, la reconnaissance des chiffres de la plaque est effectuée avec EasyOCR, une fois ces chiffres détectés nous allons ensuite les comparer aux chiffres déjà présents sur la base de données du site de paiement.

- Pour commencer nous avons ajouté une page pour que le client puisse s'inscrire sur la plateforme :

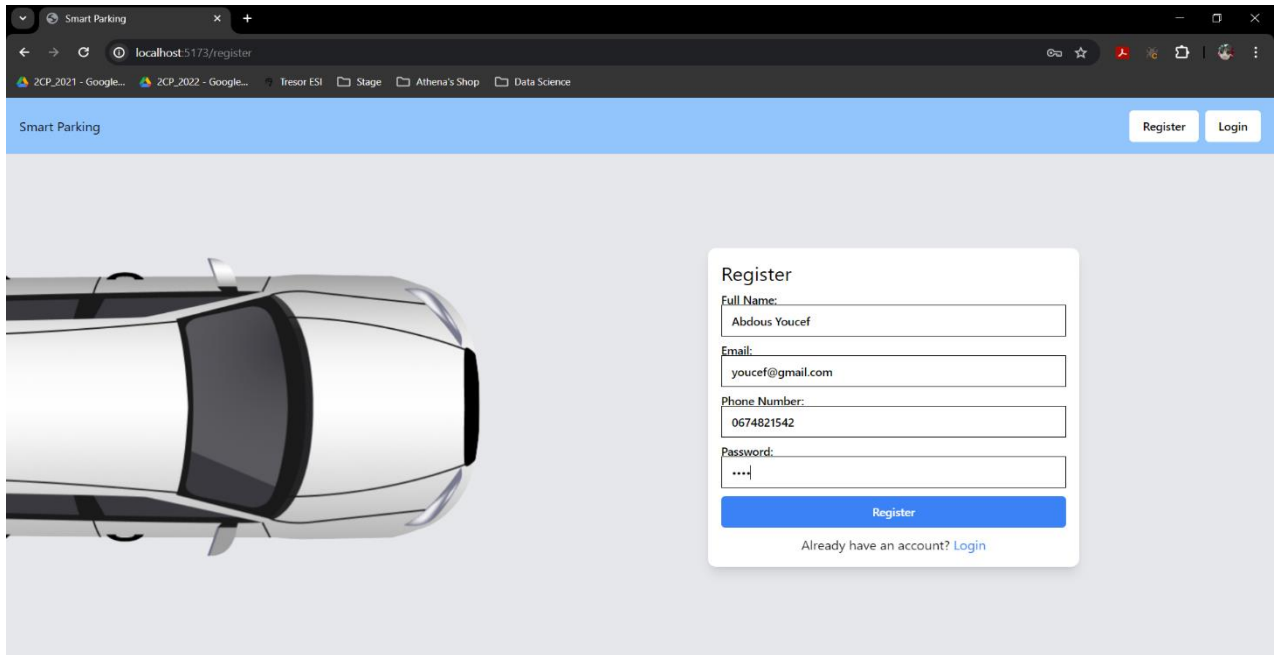


Figure 111 Inscription du client

- Nous avons également ajouté une page profile du client pour qu'il puisse consulter ses informations, une fois inscrit il sera dirigé vers la page de son profile où il pourra ajouter du solde à son compte :

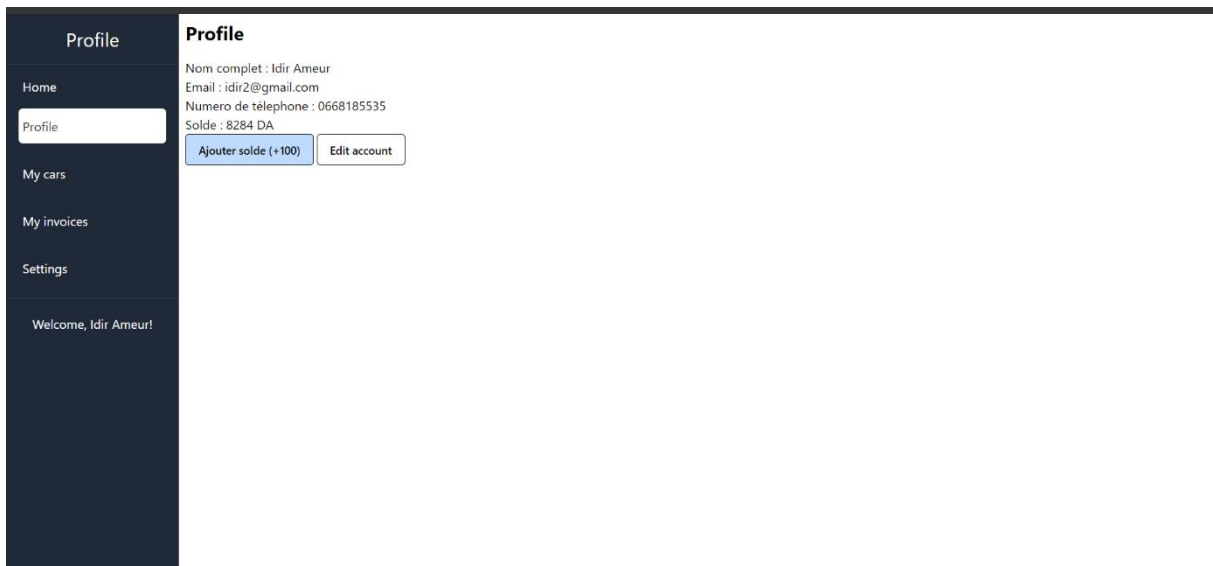


Figure 112 Profile du client

Profile

Nom complet : Idir Ameer

Email : idir2@gmail.com

Numero de téléphone : 0668185535

Solde : 8284 DA

Ajouter solde (+100)

Edit account



Figure 113 Ajout de solde

➤ Le client pourra aussi ajouter sa propre voiture et ajouter les informations relatives :

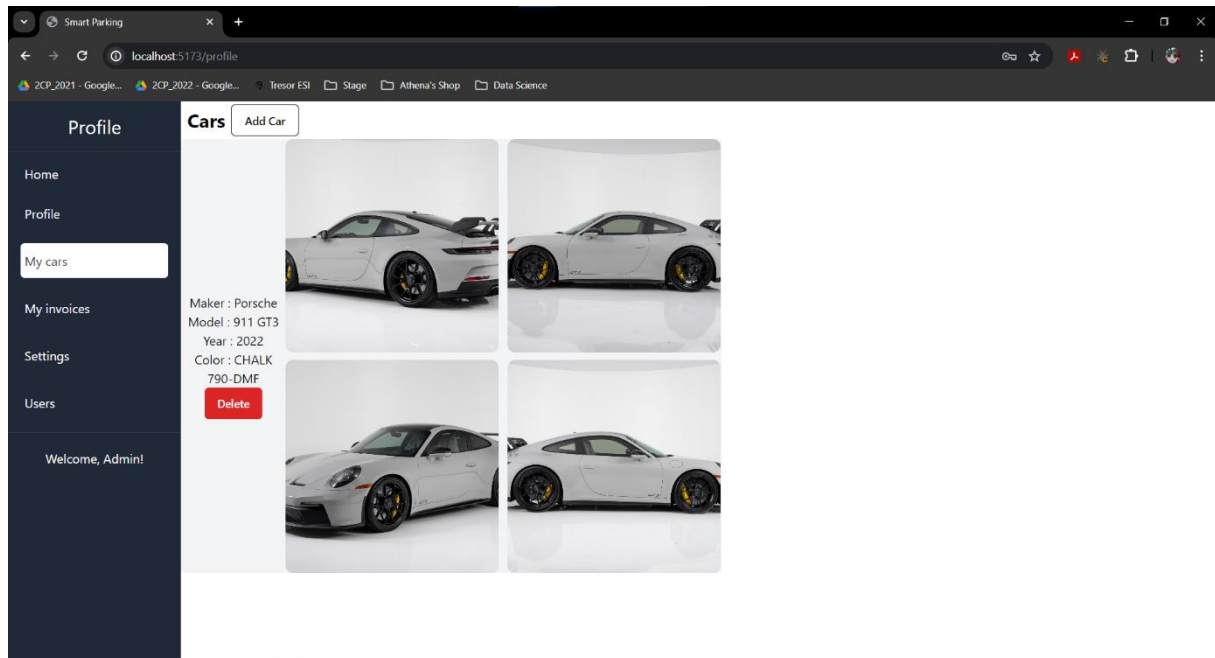


Figure 114 Ajout de véhicule

- Une fois la voiture enregistrée, le client pourra se présenter à l'entrée du parking où la détection et la reconnaissance de la plaque sera effectuée et sera lu dans la base de données et l'entrée sera enregistré :

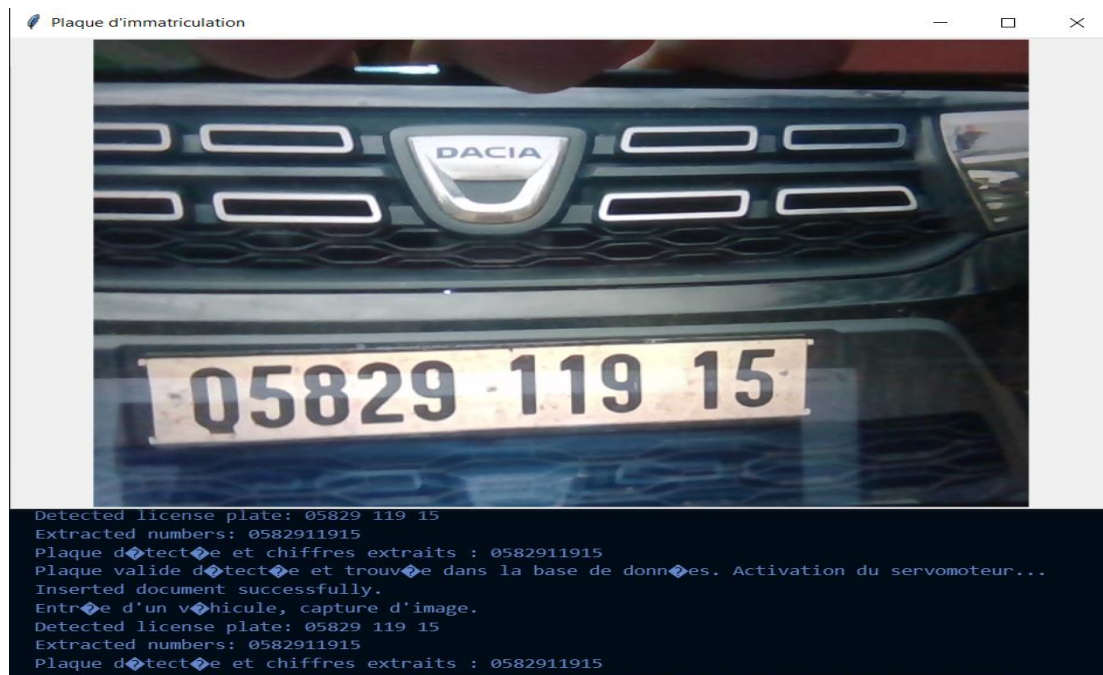


Figure 115 Enregistrement d'entrée

- A la sortie, le numéro d'immatriculation sera lu, s'il est identique à une immatricule enregistré en entrée, la barrière s'ouvrira et générera une facture sur l'application web en fonction du temps :

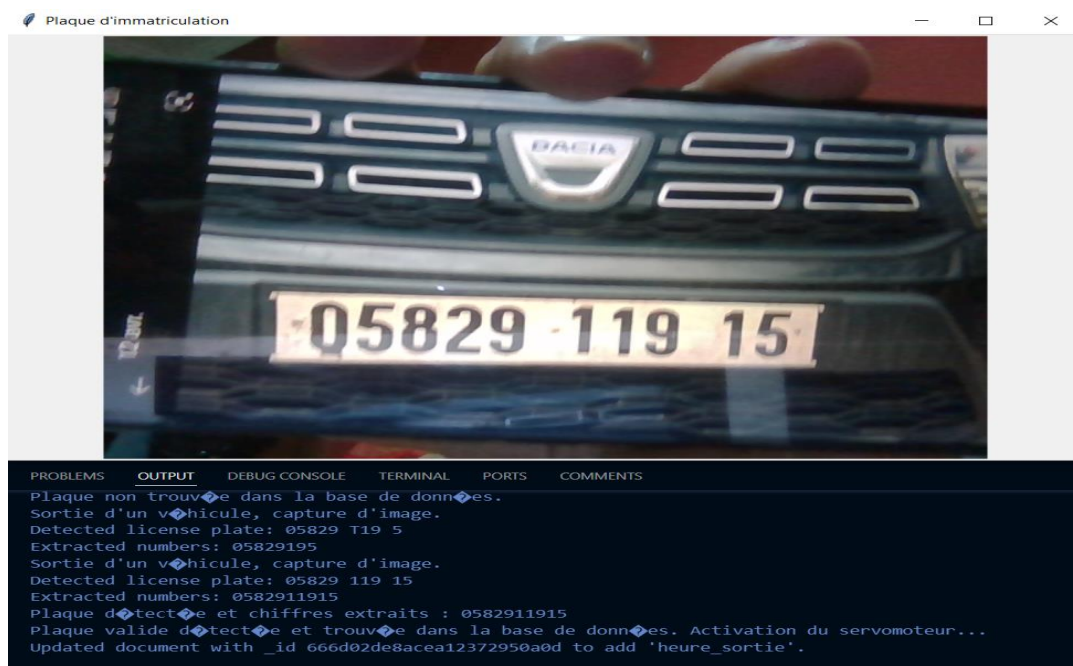


Figure 116 Enregistrement de la sortie

- La facture sera payée par le client à travers la rubrique ‘My invoices’ où le montant sera affiché ainsi que la date et l’heure d’entrée et de sortie :

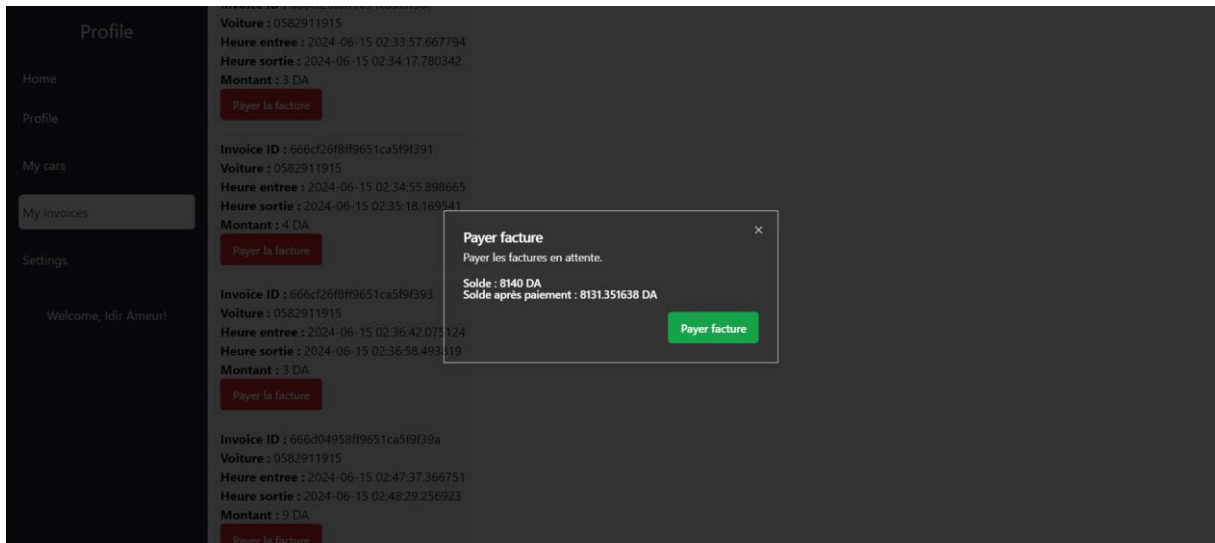


Figure 117 Paiement de la facture

- Pour le coté administrateur, il aura accès à une rubrique privée où il pourra consulter le solde de tous les clients ainsi que leurs informations respectives :

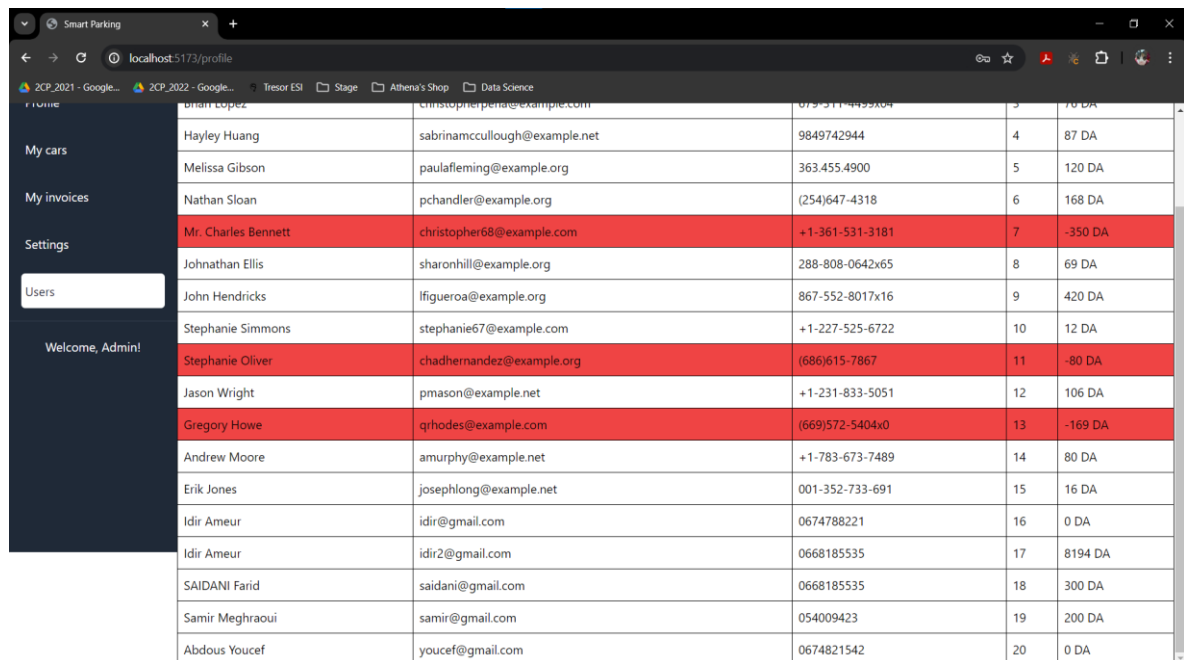


Figure 118 Rubrique admin

Discussion :

Dans ce chapitre nous avons vu comment fonctionne notre système, ensuite nous avons décrit tous les composants utilisés ainsi que les logiciels, puis nous avons aussi présenté les différents programmes python utilisés, à la fin nous avons présenté les résultats obtenus lors de la phase test, et nous avons effectué une comparaison entre les deux méthode connues pour la reconnaissance de la plaque d'immatriculation à savoir EasyOCR et Tesseract, c'est ainsi que nous avons constaté que l'utilisation de EasyOCR est plus efficace que Tesseract.

En dernier nous avons aussi décrit notre système de paiement et présenté la façon dont il fonctionne, à la fin nous avons donnés des résultats concernant ce dernier.

Conclusion

Conclusion

Notre projet consiste à la réalisation d'un système de paiement intelligent avec la plaque d'immatriculation, qui repose sur un système embarqué à base d'une carte Raspberry Pi 3, et une partie application web et ce dispositif utilise deux techniques majeures à savoir la Détection et la Reconnaissance de la plaque d'immatriculation en utilisant le traitement d'image et l'apprentissage automatique.

Dans un second temps, nous avons vu l'importance des systèmes de reconnaissance automatique des plaques d'immatriculation et les différents domaines d'application, nous avons étudié les différentes méthodes utilisées pour les RAPI tels que le traitement d'image classique, l'apprentissage profond.

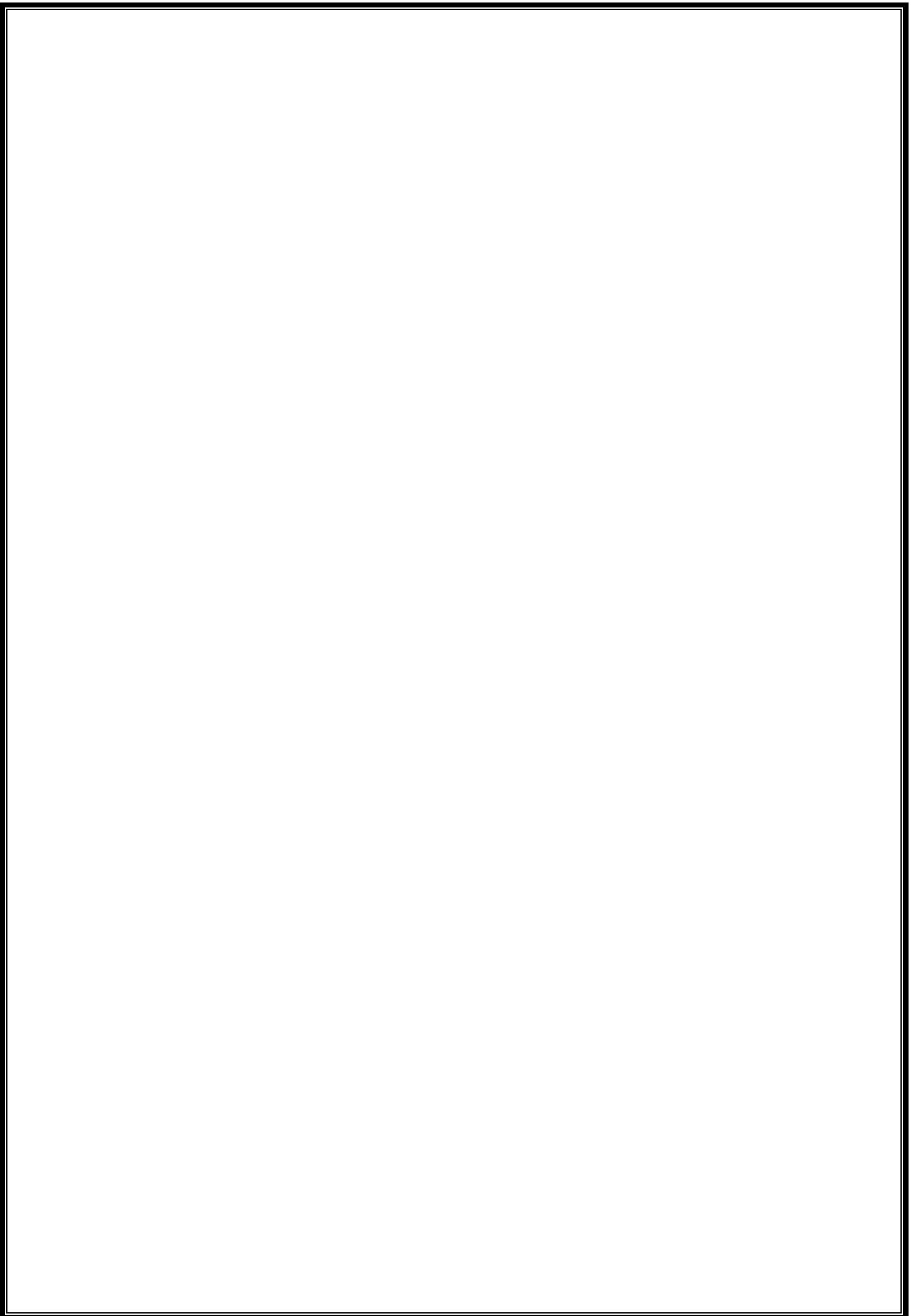
Dans ce système, nous avons utilisé les réseaux de neurones, que nous avons appliqués à la détection et à la reconnaissance de la plaque.

Il existe différentes méthodes de paiement des parkings, péages et autres qui offrent moins d'avantages, par rapport à notre système qui offre une meilleure approche pour les paiements, il permet non seulement à l'utilisateur de gagner du temps pour payer sa place dans un parking, mais aussi aux gérants d'avoir un contrôle en temps réel et une bonne gestion de son parking. L'interconnexion entre la carte Raspberry Pi 3 et l'application web permet à notre système d'être portatif ce qui le rend accessible de n'importe quel endroit disposant d'une connexion internet.

A travers ce travail, nous avons pu améliorer l'une des méthodes de paiement existantes, ce travail ouvre également des perspectives d'amélioration et d'extension. Il serait intéressant d'étendre les tests de notre système à d'autres types de plaques d'immatriculation provenant de différents pays, afin d'assurer sa robustesse et sa généralisation.

En conclusion, ce projet contribue à l'avancement des connaissances dans les domaines du traitement d'images, de l'intelligence artificielle et des systèmes de paiement.

Nous espérons que ce travail servira de référence pour d'autres travaux dans le domaine de la reconnaissance automatique des plaques d'immatriculation et des systèmes de paiement et encouragera davantage de recherches dans des domaines connexes.



Bibliographie

Bibliographie

[1] « (Qu'est-ce qu'un histogramme ?) »

<https://www.la-retouche-photo.com/comprendre-histogramme/>

[2] Edmond Boyer « (PDF)

<https://morpheo.inrialpes.fr/people/Boyer/Teaching/M2PGI/c2.pdf> »

[3] Université Biskra « (PDF) <http://thesis.univ-biskra.dz/2271/6/Chapitre%2003.pdf> »

[4] « <https://docplayer.fr/3172625-Avril-2008-universite-paul-sabatier-iut-departement-de-mesures-physiques-j-p-gastellu-etchegorry.html> »

[5] « <https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/> »

[6] « <https://www.sap.com/suisse/products/artificial-intelligence/what-is-machine-learning.html#:> »

[7] « <https://moov.ai/fr/blog/reconnaissance-optique-de-caracteres-ocr> »

[8] « <https://www.klippa.com/fr/blog/informations/ocr-guide-ultime/> »

[9] « <https://aws.amazon.com/fr/what-is/ocr/> »

[10] « <https://www.sap.com/suisse/products/artificial-intelligence/what-is-machine-learning.html> »

[11] « <https://experiences.microsoft.fr/articles/intelligence-artificielle/comprendre-utiliser-intelligence-artificielle/> »

[12] « <https://datascientest.com/machine-learning-tout-savoir> »

[13] « <https://www.jedha.co/formation-ia/reseau-neurones-deep-learning> »

[14] « <https://fr.blog.businessdecision.com/tutoriel-deep-learning-le-reseau-neuronal-convolutif-cnn/> »

[15] « <https://datascientest.com/apprentissage-non-supervise> »

[16] « <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/> »

[17] « <https://www.gotronic.fr/art-carte-raspberry-pi-3-b-24532.htm#:~:text=Le%20Raspberry%20Pi%203%20B,Linux%20ou%20Windows%2010%20IoT.> »

[18] « <https://www.tensorflow.org/lite/guide?hl=fr> »

[19] « <https://www.legratin.io/blog/3-frameworks-pour-votre-machine-learning> »

[20] « [https://fr.mathworks.com/discovery/convolutional-neural-network.html#:~:text=Un%20r%C3%A9seau%20neuronal%20convolutif%20\(CNN,des%20casses%20et%20des%20cat%C3%A9gories.](https://fr.mathworks.com/discovery/convolutional-neural-network.html#:~:text=Un%20r%C3%A9seau%20neuronal%20convolutif%20(CNN,des%20casses%20et%20des%20cat%C3%A9gories.) »

[21] « [ONS : Office National des Statistiques](#) »

Bibliographie