

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Djilali Bounaama Khemis Miliana



Faculté des Sciences et de la Technologie
Département de Technologie

Mémoire du Projet de Fin d'Etudes
Pour l'obtention du diplôme de

Master

En

« Télécommunications »

Option :

« Systèmes des Télécommunications »

Titre :

**Détection et gestion des automobiles au sein d'un parking par la
reconnaissance des matricules avec des techniques avancées**

Réalisé par :

- TABI Mohammed Riad
- SEGHIRIA Youcef Salim

Encadré par :

Mme BOUNIF Aouda

Année Universitaire : 2019/2020

Remerciements

*Nous remercions Dieu « ALLAH » tout puissant qui nous a
Donné le courage pour réaliser le défi de ce projet, et la force de finaliser ce travail.*

*Nous voudrions dans un premier temps remercier monsieur le président ainsi les
membres du jury de nous avoir fait l'honneur d'accepter d'évaluer notre travail de
mémoire. Nous avons également été honorés de votre participation à notre jury de
soutenance malgré la période critique que nous vivons.*

*Nous remercions notre directrice de mémoire madame BOUNIF Aouda pour sa
disponibilité et surtout ses judicieux conseils. Nous la remercions de nous avoir
encadré, orienté, conseillé et surtout de nous avoir fait confiance.*

*Nous souhaitons adresser nos remerciements les plus sincères
aux personnes qui nous ont apporté leur aide et qui ont contribué à
l'élaboration de ce mémoire particulièrement Mr. SELMI Anwar.*

*Enfin nos profonds et sincères remerciements vont vers nos parents et familles
respectives.*

Dédicaces

*Aucune dédicace ne saurait être assez éloquente pour exprimer ce que nos parents méritent pour tous les sacrifices qu'ils n'ont cessé de nous donner.
Pour cela, nous dédions ce modeste travail à nos parents respectifs.*

*Nous tenons à rappeler qu'aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de nous combler, que Dieu leur procure bonne santé et longue vie
Ce travail est le fruit de leurs sacrifices.*

*Moi, Mohamed Riad, je dédie aussi ce modeste travail à ma grande famille et en particulier mon frère Mohamed Rafik, ma sœur Ryma et mon beau-frère Hichem pour le soutien moral qu'ils m'ont tous donné.
Je le dédie aussi à mes petits neveux Abd'El Ali et Norhane.*

Mohammed Riad

Dédicaces

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que nos parents méritent pour tous les sacrifices qu'ils n'ont cessé de nous donner.

*Pour cela, nous dédions ce modeste travail à nos parents respectifs .
Nous tenons à rappeler qu'aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de nous combler , que Dieu leur procure bonne santé et longue vie
Ce travail est le fruit de leurs sacrifices.*

Moi, Youcef Salim, je dédie aussi ce modeste travail à ma grande famille et en particulier mes soeurs Racha et Amira, mes oncles ,mes cousins et tous mais amis pour le soutien moral qu'ils m'ont tous donné.

Youcef Salim

Résumé

La dernière décennie fut marquée une augmentation significative du nombre de véhicules et avec le développement de leur conception et l'augmentation correspondante du nombre de vols ciblés en particulier pour les composants automobiles coûteux, il est devenu nécessaire de profiter de l'énorme développement dans le domaine des télécommunications, réseaux, et l'intelligence artificielle afin de créer un parking adapté programmé pour faire fonctionner et gérer cette énorme quantité de véhicules et ainsi réduire le phénomène de stationnement chaotique et éviter l'exposition au vol. Les entreprises publiques ou privées dont le contenu sensible est exposé à l'espionnage ou au vol, tentent toujours de sécuriser et bien gérer leur parking. Donc répondant aux besoins ce mémoire focalise sur le développement d'une application bureautique qui repose sur la création d'un système de stationnement en reconnaissant la plaque d'immatriculation comme une source d'identité de la personne, que nous extrayons d'une base de données conçue liée à une interface graphique montre les résultats en fonction de la nature de l'interaction du système et selon que nous voulons .

Mots clés : Parking , matricule , ROC(reconnaissance optique des caractères) , vision par ordinateur , Python ,Open CV, apprentissage approfondie , base de données, GUI.

Abstract

The last decade has been marked by a significant increase in the number of vehicles and with the development of their design and the corresponding increase in the number of targeted thefts especially for expensive automotive components, it has become necessary to take advantage of the huge development in the field of telecommunications, networks, and artificial intelligence in order to create a suitable parking program programmed to operate and manage this enormous amount of vehicles and in consequence reduce the phenomenon of disorder parking and avoid exposure to theft. Public or private companies, whose sensitive content is exposed to spying or theft, always try to secure and manage their parking properly. So responding to these needs, this dissertation focuses on the development of an software based on the creation of a parking system by recognizing the license plate as a source of identity for the person, which we extract from a database designed and linked to a graphical interface let us show the results according to the nature of the interaction of the system and according to what we want.

Keywords : Parking lot ,license plate , OCR, computer vision ,Python, Open CV , deep learning ,database,GUI.

Liste des abréviations

ANPR : automatique number plate recognition
LPR : Licence plate recognition
LAPI : Lecture Automatique de Plaques d'Immatriculation)
ROC : Reconnaissance Optique de Caractères (ROC)
RAPI : Remote Application Program Interface
JPEG: Joint photographic experts group
IP: internet protocol
OPENCV : Open Source Computer Vision
VMS : Video Management Software (various companies)
CSV : Comma Separated Variable
NET : network
RVB : Rouge Vert Bleu
CCD : charge-coupled device
IDE : Integrated development environment
CNN : Convolutional neural network
RNN : Recurrent neural network
R-CNN : Region-convolutional neural network
SGBD : Système de Gestion de Base de Données
SSD : Single shot detector
IOT : Intersection over union ratio
VGG : Visual Geometry Group
YOLO : You only look one
UML : Unified Modeling Language
RFID : Radio frequency –identification
SQL : Structured Query Language
GUI : Graphical User Interface

Liste des figures

Chapitre 1

Fig.1.1 : Une voiture avec immatriculation algérienne	4
Fig.1.2 : Exemple d'architecture matérielle de système LAPI	6
Fig.1.3 : Exemple d'architecture matérielle de système LAPI-NETWORK.....	7
Fig.1.4 : Utilisation de système VIRONIC pour la surveillance routière.....	8
Fig.1.5 : Image prise d'une photographie amazonienne.....	11
Fig.1.6 : Représentation matricielle d'une image.....	14
Fig.1.7 : Procédure de traitement d'image.....	17
Fig.1.8 : Application de filtre moyenneur sur une image.....	18
Fig.1.9 : Application de filtre smooth sur une image.....	19
Fig.1.10 : Application de filtre gradient sur une image	19
Fig.1.11 : Application de filtre Laplacien sur une image	20
Fig.1.12 : Représentation des opérations compression et décompression.....	22
Fig.1.13 : Schéma d'un réseau d'apprentissage approfondi	26
Fig.1.14 : Schéma d'un réseau de neurones d'humain et de machine	27
Fig.1.15 : Schéma d'un réseau de neurones avec ses couches.....	28
Fig.1.16 : schéma d'un réseau de neurones avec ses couches.....	29

Chapitre 2

Fig. 2.1 : Architecture de SSD.....	34
Fig. 2.2 : Fenêtres de QT designer.....	36
Fig .2.3 : fenêtre d'invitation des commandes.....	37

Fig. 2.4 : Représentation du réseau local (sans fil).....	39
Fig. 2.5 : Matrice d'une région obtenue d'une image réelle.....	40
Fig. 2.6 : Matrice d'une région d'une image en niveau de gris.....	40
Fig. 2.7 : Matrice d'une région d'une image en noir et blanc.....	41
Fig. 2.8 : Liste de configuration des donnés.....	43
Fig. 2.9 : Image originale en couleurs.....	44
Fig. 2.10 : Image en niveau de gris.....	45
Fig. 2.11 : Image après application des filtres bilatéral et canny	45
Fig. 2.12 : Image avec les contours détectés.....	46
Fig. 2.13 : Image avec les contours proches de matricule	47
Fig. 2.14 : Image avec matricule la localisation de matricule.....	47
Fig. 2.15 : Extraction de numéro d'immatriculation.....	48
Fig. 2.16 : Ensemble des véhicules avec matricules détectés.....	48

Chapitre 3

Fig. 3.1 : Représentation de fonctionnement de système de gestion de parking.....	51
Fig. 3.2 : Diagramme des cas d'utilisations.....	52
Fig. 3.3 : Diagramme d'activité.....	53
Fig. 3.4 : Représentation les éléments de base de données.....	53
Fig. 3.5 : Structure de tableau 'emplacement'.....	54
Fig. 3.6 : Structure de tableau 'historique'.....	54
Fig. 3.7 : Structure de tableau 'propriétaire'.....	54
Fig. 3.8 : Structure de tableau 'voiture'.....	55
Fig. 3.9 : Structure générale de diagramme des bases de données.....	55
Fig. 3.10 : Interface principale.....	56
Fig. 3.11 : Interface de gestion.....	57
Fig. 3.12 : Page d'accueil d'application web phpMyadmin	58
Fig. 3.13 : Insertion du nom de base données.....	58
Fig. 3.14 : Nom et prénoms d'un propriétaire	58
Fig. 3.15 : Véhicule enregistrée.....	59

Fig .3.16: Les emplacements disponibles dans le parking.....	59
Fig. 3.17 : Voiture entrante.....	59
Fig .3.18 : Visualisation de la voiture en temps réel.....	60
Fig .3.19 : Affichage de la photo capturé avec le matricule détecté.....	60
Fig. 3.20 : Affichage des informations du propriétaire enregistré.....	61
Fig.3.21 : Interface d'emplacement pour propriétaire entrant.....	62
Fig.3.22 : Réservation de l'emplacement A8	62
Fig.3.23 : Interface de sortie.....	63
Fig.3.24 : Résultat d'identification pour le cas d'un intrus.....	63
Fig.3.25 : Fenêtre pour l'ajout des visiteurs	64

Table de Matière

Remerciement

Résumé

Introduction générale.....

Chapitre 1 : Étude des techniques de détection de matricule et gestion des données dans un parking

1.1. Introduction.....	3
1.2. État de l'art.....	3
1.2.1. Un aperçu sur les parkings	3
1.2.1.1. Définition du parking	3
1.2.2. Plaque d'immatriculation.....	3
1.2.3. Exploitation des plaques d'immatriculations pour organiser les parkings.....	4
1.2.4. Les dernières technologies utilisées pour détecter les matricules.....	5
1.2.4.1. LAPI (Lecture Automatique de Plaques d'Immatriculation).....	5
1.2.4.2. Système VIRONIC.....	7
1.2.4.3. Système Auto Vu.....	8
1.2.4.4. OpenALPR.....	9
1.2.4.5. Autres applications en utilisant le numéro d'immatriculation.....	10
1.3. La vision par ordinateur.....	11
1.3.1. L'image.....	11
1.3.1.1. Le concept de l'image.....	11
1.3.1.2. L'image analogique et numérique.....	11
1.3.1.3. De L'analogique vers le numérique.....	12
1.3.1.4. Représentation d'image numérique.....	13
1.3.1.5. Caractéristiques d'une image numérique.....	14
1.3.2. Traitement d'image.....	15
1.3.2.1. La discipline	15
1.3.2.2. Les procédures utilisées dans le système de traitement.....	16
1.3.2.3. Le filtrage.....	17

1.3.2.4. Compression et décompression d'image	21
1.3.2.5. Les applications possibles avec le traitement d'image	22
1.3.3. Présentation du langage Python	23
1.3.3.1. Caractéristiques du langage	23
1.3.3.2. Les environnements de développement.....	24
1.3.4. L'apprentissage profond (Deep learning).....	25
1.3.4.1. Les réseaux de neurones.....	26
1.4. Base de données MySQL.....	29
1.4.1. Base de données.....	29
1.4.2. Langage MySQL.....	30
1.4.3. SGBD	30
1.5. Interface graphique	30
1.5.1. Application desktop	30
1.5.2. Réalisation d'une application bureautique.....	30
1.5.3. Application de base de données.....	30
1.6. Conclusion.....	31

Chapitre 2 : L'analyse et l'implémentation de système de détection

2.1. Introduction.....	32
2.2. Outils de développement.....	32
2.2.1. Bibliothèque Open CV.....	32
2.2.2. Bibliothèque Numpy.....	33
2.2.3. Bibliothèque Imutils.....	33
2.2.4. Bibliothèque pytesseract.....	33
2.2.5. Bibliothèque matplotlib.....	33

2.2.6. Bibliothèque urllib.....	33
2.2.7. Bibliothèque tensorflow.....	33
2.2.8. La technique de détection d'objets SSD (framework SSD).....	34
2.2.9. Bibliothèque MySQL.....	35
2.2.10. PHP MyAdmin.....	35
2.2.11. Bibliothèque PyQt5.....	35
2.2.12. Qt designer.....	35
2.3. La procédures du système de détection de matricule.....	37
2.3.1. Traitements réalisés par l'algorithme	38
2.3.1.1. Algorithme de prototype (1 ^{er} modèle).....	
2.3.1.2. Algorithme entraînée par le deep learning (2ème modèle).....	42
2.3.2. Exécution et discussion des résultats de chaque partie du programme....	44
2.3.2.1. Implémentation de 1er modèle	44
2.3.2.2. Implémentation de 2ème modèle (deep learning).....	48
2.4. Conclusion.....	49

Chapitre 3 : Étude conceptuelle et réalisation de l'application

3.1. Introduction.....	50
3.2. Cahier de charge.....	50
3.3. Diagrammes UML.....	52
3.3.1. Diagramme de cas d'utilisation.....	52
3.3.2. Diagramme d'activité.....	52
3.3.3. Diagramme de classes.....	53
3.4. Aperçu sur l'interface de l'application.....	56
3.4.1. Interface de temps réel.....	56
3.4.2. Interface de traitement avec la base de données.....	57
3.5. Description de fonctionnement de logiciel.....	57
3.5.1. Dans le cas d'appartenance au parking.....	58
3.5.2. Résultat d'identification pour le cas d'intrus.....	63
3.6. Conclusion.....	64
Conclusion générale.....	65
Bibliographie	

Introduction générale

INTRODUCTION GÉNÉRALE

Contexte

Au début du XX ème siècle, l'automobile était considérée comme un « jouet couteux» que seuls les riches expérimentés ou non pouvaient s'en acquérir. Elles étaient alors garées à la place des « voitures à cheval »

Avec l'immense développement que le monde de production de voitures a connu, l'automobile est devenue un outil accessible et indispensable : elle a désormais changé de statut. Pour faire face au nombre excessif de véhicules, des parkings sont nés et majoritairement sous les édifices dans un souci d'économie d'espace.

les pouvoirs publics ont mis en place le stationnement réglementé et le stationnement payant. Sa gestion est un moyen qui permet de réguler le déséquilibre entre l'offre et la demande de stationnement sur la voie publique. Quant 'à L'espace de stationnement réservé à une entreprise ou une administration, il répond à un autre type de gestion : seuls les employés y ont accès.

Problématique et objectif

Dans notre projet, nous allons créer un programme qui va permettre au système de s'assurer de la légitimité de la demande d'accès faite par une entité (être humain ou autre système), afin d'autoriser cette entité à des ressources du système conformément au contrôle d'accès .cet authentification se fera grâce à la capture d'images des plaques. Le défi le plus important réside dans le fait de pouvoir localiser le matricule et reconnaître son numéro, souvent dans toutes les conditions.

Ce que nous visons à travers notre travail est de réaliser la tâche d'identification des caractères du matricule et l'intégrer au logiciel qui contient ce projet.

Plan de mémoire

Pour cela, nous avons décomposé notre mémoire comme suit :

Dans le premier chapitre, nous commencerons par donner un aperçu sur les parkings, ainsi que les dernières technologies concernant la détection des plaques d'immatriculation. Ensuite, nous allons présenter les notions de base nécessaires à la compréhension de l'image et le

Introduction générale

traitement d'images, ainsi que le langage de python et les notions des tableaux MySQL de l'interface graphique.

Dans le deuxième chapitre, nous présenterons les algorithmes qui doivent être réalisés pour que notre programme puisse identifier une plaque d'immatriculation puis extraire son numéro grâce à la technique d'intelligence artificielle représentée en apprentissage approfondi.

Pour Le troisième et dernier chapitre, nous allons enfin mettre en pratique notre projet par la présentation des résultats expérimentaux afin de valider ou non son fonctionnement.

A la fin, Nous terminerons ce mémoire par une conclusion générale et d'éventuelles perspectives.

Chapitre 1

Étude des techniques
de détection de
matricule et gestion
des données dans un
parking

1.1 .Introduction

L'un des problèmes auxquels nous sommes confrontés dans les parkings est celui de la réglementation et de la sécurité. Pour éliminer ces obstacles, nous sommes amenés à identifier chaque voiture par une particularité spécifique telle que: la plaque de numérotation ou d'immatriculation.

Dans ce chapitre, nous passerons en revue les technologies utilisées par les systèmes et programmes qui fonctionnent grâce au tableau de numérotation dans la conduite des voitures et résoudre ainsi divers problèmes de circulation.

Nous montrerons également les différents types d'images et les mécanismes de leur traitement afin de détecter les plaques de numérotation. Enfin, nous adopterons le langage de programmation Python comme langage approprié pour ce système en plus des bases de données et de l'interface graphique que nous appliquerons pour développer un programme spécial pour ce système.

1.2.État de l'art

1.2.1 .Un aperçu sur les parkings

1.2.1.1. Définition du parking

Le parking d'automobiles est un terrain (une surface clôturée) de stationnement d'un ensemble des voitures immatriculées, pendant un temps déterminé. Il peut se trouver :

- dans un immeuble bâti en superstructure (partie en élévation à l'air libre) ou en infrastructure (partie enterrée ou en dessous du sol artificiel, dalle par exemple).
- sur une aire aménagée ou non pour le stationnement, sur une terrasse d'un immeuble, sous un immeuble bâti (sur pilotis ou en encorbellement).

1.2.2. Plaque d'immatriculation

Une plaque de matricule est une plaque portant une combinaison unique de chiffres ou de lettres (pour une zone géographique donnée), destinée à identifier facilement un véhicule terrestre (automobile, moto, véhicule agricole, etc.).

Pour l'Algérie Le numéro d'immatriculation est composé de 3 groupes de chiffres séparés par un espace :

- 1^{er} groupe : 5 chiffres, dont le 0 initial, qui correspond au numéro de dossier du véhicule ;
- 2^e groupe, composé de 3 chiffres : le premier permet d'identifier le type de véhicule (ex : 1 un véhicule familial, 2 pour un poids lourd, 3 voiture commerciale, 4 bus , 5 tracteur d semi remorque , 6 tracteur , 7 engins, 8 remorque, 9 moto .) ; les deux suivants renvoient à l'année de mise en circulation du véhicule ;
- 3^e groupe : 2 chiffres qui identifient la wilaya d'immatriculation.



Fig.1.1 : une voiture avec immatriculation algérienne.

1.2.3. Exploitation des plaques d'immatriculations pour organiser les parkings

Chaque voiture a son propre numéro d'immatriculation, c'est pourquoi nous pouvons l'utiliser Pour assurer une gestion de stationnement plus facile afin des buts d'organisation, de sécurité et aussi le service de paiement.

Avec le développement de la technologie, plusieurs idées ont émergé pour capturer les plaques d'immatriculations, en développant des algorithmes qui fonctionnent avec différents types de numérotation dans de nombreux pays

Ces techniques ont été utilisées par certaines institutions internationales telles que la police pour régler les routes ainsi que par des entreprises qui contiennent des parkings géants

Ce sujet a été mentionné dans de nombreux articles de recherche scientifique et il est ainsi devenu un champ ouvert pour le développement et la créativité.

Cette technique est normalisée sous le nom de LPR ou ANPR , Le LPR est une technologie de traitement d'image utilisée pour identifier les véhicules par leurs plaques d'immatriculation.

Le système utilise un éclairage et une caméra infrarouge pour prendre l'image de l'avant ou de l'arrière du véhicule, puis un logiciel de traitement d'image en fait l'analyse et extrait les informations de la plaque d'immatriculation [1].

1.2.4. Les dernières technologies utilisées pour détecter les matricules

La recherche scientifique et le besoin urgent dans ce domaine ont contribué au développement du système LPR en aboutissant à des solutions exigeant des plateformes et des logiciels tels que :

1.2.4.1. LAPI

La reconnaissance de plaques minéralogiques ou LAPI repose sur le procédé informatique de Reconnaissance Optique de Caractères (ROC). Ce procédé permet d'isoler et de capturer l'image d'une plaque minéralogique, d'en extraire les caractères alphanumériques et de les convertir en données numériques afin de les analyser et de les comparer avec une base de données.

Les algorithmes des logiciels LAPI sont en mesure de gérer la position du véhicule (orientation, distance, vitesse...), les problèmes d'éclairage et de contraste ainsi que les éventuelles barrières physiques (immatriculation couverte de poussière, police d'écriture fantaisie...). Leur vitesse d'exécution permet d'analyser un nombre élevé d'images prises en temps réel sur des axes de circulation à grande vitesse [2].

Caractéristiques

Le système LAPI est en mesure de posséder les caractéristiques suivantes [3] :

- Lecture et reconnaissance de plaques d'immatriculation des pays de l'Union Européenne et d'Afrique.
- Système automatique 24h/24 h et indépendant du matériel .La solution fonctionne avec tout type de caméras IP répondant à nos préconisations techniques.
- Interface simple à configurer disponible sur PC Windows et PC linux en 2 langues(Français/Anglais)
- Gestion des autorisations d'entrées et sorties
- Possibilité de chiffrer les données (numéros de plaques et photos)

Les produits qui appliquent système LAPI :

a. LAPI-ACCES

LAPI ACCES est une solution logicielle de contrôle d'accès par lecture automatique de plaques d'immatriculation s'adressant au marché de la sécurité. Le logiciel intègre de

nombreuses innovations algorithmiques permettant la reconnaissance en temps réel, de jour comme de nuit, des plaques d'immatriculation de tous types de véhicules (autos, motos, poids lourds) passant dans les champs de vision des caméras. LAPI ACCES est un produit autonome et indépendant du matériel disposant d'un taux de reconnaissance optimal.

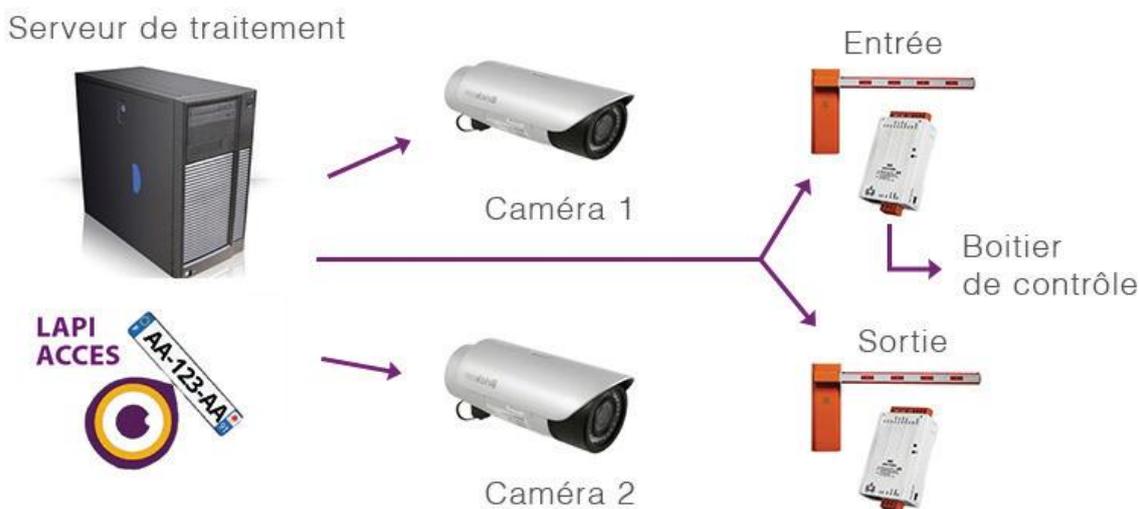


Fig.1.2 : Exemple d'architecture matérielle de système LAPI [3].

b.LAPI-ENGINE

Le produit LAPI ENGINE représente le cœur technologique permettant la Lecture Automatique de Plaques d'Immatriculation (LAPI - ANPR) qui a été intégré dans l'ensemble de solutions. Principalement dédiée à l'identification et la traçabilité de véhicules, LAPI ENGINE est un produit autonome pouvant s'adapter à un large éventail d'applications. Il s'intègre facilement dans des solutions "métiers" comme des logiciels de contrôle d'accès, de pesée industrielle ou de vidéosurveillance [3].

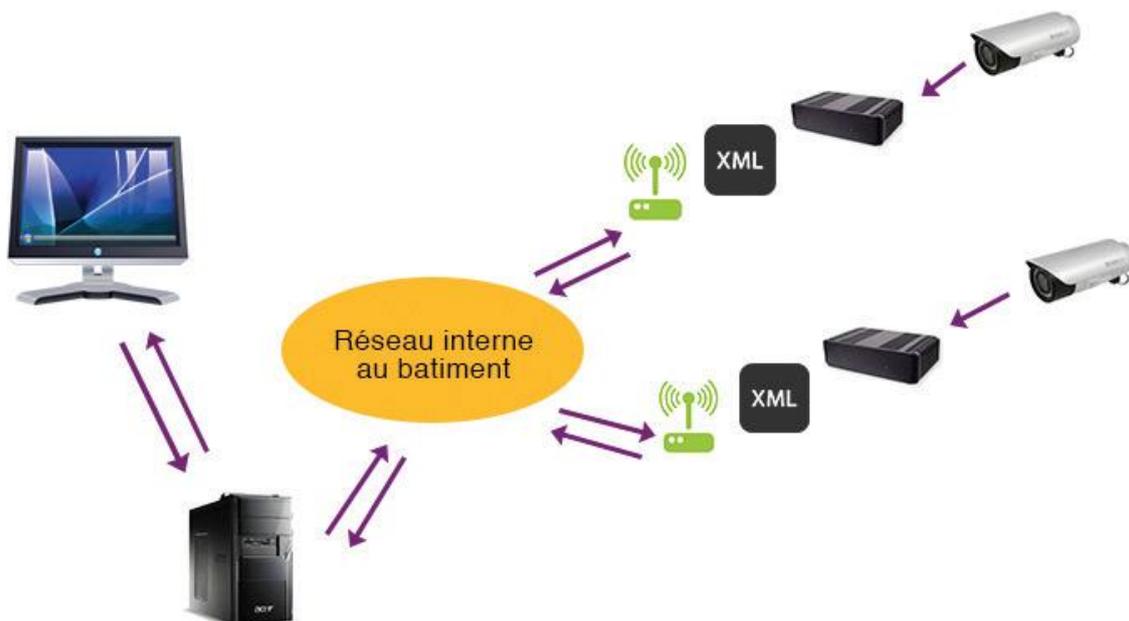


Fig.1.3 : Exemple d'architecture matérielle du système LAPI-NETWORK [3].

c.LAPI-SMARTPHONE

La solution LAPI SMARTPHONE permet la détection et la reconnaissance automatique de plaques d'immatriculation à partir de l'analyse d'un flux vidéo ou d'une photo, en temps réel. Dédiée principalement à la gestion et au contrôle de stationnement, cette application permet le contrôle d'une plaque minéralogique par rapport à une base de données locale ou distante et s'intègre facilement à d'autres systèmes de gestion. [3]

1.2.4.2. Système VIRONIC [4]

Les systèmes de reconnaissance des plaques d'immatriculation POLISCAN SURVEILLANCE de VITRONIC trouvent leur place partout où l'accès est réservé à un type ou à une catégorie de véhicules.

a. Fonctionnement

Le système POLISCAN SURVEILLANCE combine la technologie de reconnaissance performante . VITRONIC et un logiciel back office qui effectue des comparaisons entre les plaques d'immatriculation relevées et celles d'une base de données. Les plaques d'immatriculation qui ne correspondent pas à la base de données sont immédiatement supprimées. Les applications de type liste noire du système POLISCAN SURVEILLANCE incluent l'aide à la recherche, les contrôles aux frontières et la localisation de véhicules volés ou avec défaut d'assurance.

Dans le cas d'une procédure dite « de liste blanche », la base de données contient des immatriculations autorisées. Avec le contrôle d'accès par liste blanche, le système signale tout véhicule qui n'est pas enregistré. Le système POLISCAN SURVEILLANCE ne se limite pas à la reconnaissance des plaques d'immatriculation. Il peut aussi identifier d'autres caractéristiques graphiques, telles que des vignettes ou des macarons, et surveiller la circulation en continu. Il permet donc d'identifier, par exemple, tout véhicule sans pastille verte dans la file de circulation. Il peut aussi surveiller les véhicules qui empruntent les couloirs réservés aux bus.

b. Avantages

- Requête automatique dans des bases de données d'immatriculation
- Prise en charge des procédures de type liste noire/liste blanche
- Taux de lecture extrêmes, y compris à vitesse élevée
- Cryptage et transfert sélectif pour la protection et la sécurité des données



Fig.1.4 : Utilisation du système VIRONIC pour la surveillance routière [4].

1.2.4.3. Système Auto Vu [5]

Auto Vu™ est le système de reconnaissance de plaques d'immatriculation (RAPI) sur IP de Security Center, la plate-forme de sécurité unifiée de Genetec. Que ce soit dans votre véhicule ou au bureau, AutoVu vous aide à automatiser l'identification des plaques d'immatriculation des véhicules. AutoVu est une solution éprouvée qui permet aux organisations d'améliorer diverses applications, telles que la surveillance urbaine, la sécurité, le contrôle d'accès des véhicules, le maintien de l'ordre et le stationnement.

a. Fonctionnalités :

Système intégré de vidéosurveillance : Fusionnez le système de reconnaissance automatique de plaques d'immatriculation avec le contrôle d'accès et la vidéosurveillance dans Security Center. Ainsi effectuer une surveillance vidéo en direct à l'aide des lectures de plaques d'immatriculation et recevoir des alertes lorsque le système AutoVu repère un véhicule recherché.

Suivi du comportement et des caractéristiques des véhicules : AutoVu analyse les caractéristiques et les comportements des véhicules scannés, vous permettant de suivre les tendances et d'identifier les véhicules suspects. Outre la capture précise des numéros des plaques d'immatriculation, AutoVu peut identifier la marque et le sens de déplacement des véhicules, estimer leur vitesse et détecter l'État ou le pays d'origine de leur plaque.

Event-to-Action : Utilisez les correspondances d'événements et lectures RAPI pour déclencher des actions avancées, telles que l'enregistrement vidéo ou l'activation de barrières via le puissant module Event-to-Action de Security Center.

b. Avantages :

- Obtenez des lectures de plaques d'immatriculation précises
- Assurez des performances ininterrompues
- Fournisseur de solutions de RAPI complètes
- Unification avec la vidéosurveillance
- Combinez la RAPI fixe et mobile

1.2.4.4. OpenALPR [6]

OpenALPR est une bibliothèque open source de reconnaissance automatique de plaques d'immatriculation écrite en C ++ avec des liaisons en C #, Java et Python. Ce logiciel est distribué à la fois dans une version commerciale basée sur le cloud et dans une version open source . OpenALPR utilise les bibliothèques OCR OpenCV et Tesseract . Il peut être exécuté comme un utilitaire de ligne de commande, une bibliothèque autonome ou un processus d'arrière-plan. Le logiciel s'intègre également aux systèmes de gestion vidéo (VMS) tels que Milestone XProtect.. La bibliothèque analyse les images et les flux vidéo pour identifier les plaques d'immatriculation. La sortie est la représentation textuelle de tout caractère de plaque d'immatriculation.

OpenALPR comprend un utilitaire de ligne de commande. Il suffit de taper "alpr [chemin du fichier image]" pour commencer à reconnaître les images de la plaque d'immatriculation.

a. Avantages

Le logiciel peut être utilisé de différentes manières. Par exemple, avec OpenALPR, vous pouvez:

- Reconnaître les plaques d'immatriculation des flux de caméras. Les résultats sont consultables et peuvent déclencher des alertes . Le référentiel de données peut être dans le cloud ou être entièrement stocké dans votre réseau sur site.
- Reconnaissez les plaques d'immatriculation des flux de caméras et envoyez les résultats à votre propre application .
- Traitez un fichier vidéo et stockez les résultats de la plaque d'immatriculation dans une base de données CSV et SQLite.
- Analyser des images fixes à partir de la ligne de commande
- Intégrez la reconnaissance de plaque d'immatriculation dans votre application directement dans le code (C / C ++, C #, VB.NET, Java, Python, Node.js)

1.2.4.5. Autres applications en utilisant le numéro d'immatriculation

Il existe beaucoup d'applications qui bénéficient de la reconnaissance des plaques d'immatriculation, nous citerons quelques-unes d'entre elles :

- **Contrôles de paiement** : pour calculer les frais de voyage on utilise le matricule et c'est le cas des passages de péage autoroutiers
- **Contrôle des frontières** : le numéro d'immatriculation est enregistré dans une base de données avec autres informations comme le passeport pour contrôler tous les passagers frontaliers et surveiller son déplacement.
- **Protection de voitures** : le vol est l'une des choses les plus importantes que les propriétaires craignent, donc la reconnaissance d'immatriculation peut être utilisée pour l'empêcher , ce système réalise une comparaison entre les voitures qui passent et la liste des voitures volées puis une alerte est déclenchée pour informer l'agent de police de la voiture détectée.
- **Les systèmes de gestion de la circulation** : qui calculent la vitesse de circulation en mesurant le temps entre les passages devant deux points de lecture.
- **Système d'information Schengen** : détection de matricule pour le stocke dans une base de données des VISA à des raisons de sécurité.

1.3. La vision par ordinateur

C'est un domaine de l'informatique et du traitement d'image, il donne la possibilité aux systèmes informatiques de voir comme les humains.

1.3.1. L'image

1.3.1.1. Le concept de l'image [7]

Une image est une représentation visuelle de quelque chose ou de quelqu'un. L'image se définirait comme une reproduction de quelque chose. Elle vient du latin « *imago* » qui désignait une sorte de masque moulé, à partir de cire d'abeille, sur le visage d'une personne morte afin d'en conserver les traits, comme d'un portrait, et d'en produire éventuellement un moulage.

On peut dire que le mot image est une modification linguistique et on peut également distinguer deux types d'images :

- **L'image mentale** : comme image mentale, naturelle, et historique.
- **L'image artificielle** : enregistrement des photos, radiographie.



Fig.1.5 : Image prise d'une photographie amazonienne. [22]

1.3.1.2. L'image analogique et numérique

La grande différence entre images numériques et analogiques est que les images numériques peuvent se décrire par un ensemble fini de valeurs entières. Si on connaît cette suite de valeurs, on peut recréer une copie exacte de l'image d'origine. On peut assimiler cette suite de valeurs entières à un "code génétique" de l'image. Au contraire, une image analogique est liée à un support matériel: plaque photo, pigments de peinture et toile, par exemple. Il n'est pas possible de reproduire l'image originale à l'identique. Les copies sont nécessairement

dégradées par rapport à l'original. Ceci est un point fort très important des images numériques: on peut créer de nombreuses répliques de l'image sans dégradation et on peut facilement transmettre, stocker, dupliquer ce fameux "code génétique" d'une image numérique comme on transmettrait un texte. Par exemple il est absolument impossible de faire passer une diapositive par le réseau téléphonique, par contre transmettre une suite de valeurs entières c'est très facile! De plus le code de l'image numérique est indépendant du support : le support vieillit, le code, lui ne vieillit pas. Bien sûr, il y a des contreparties : · le nombre de valeurs permettant de décrire une image numérique est très grand (on atteint rapidement plusieurs millions) et d'autant plus grand que l'on souhaite une bonne qualité d'image. De plus, le code de l'image est lui-même stocké sur des supports matériels (disquettes informatiques, CD-ROM) qui vieillissent et peuvent même devenir obsolètes si les machines qui permettent de les lire n'existent plus! · Enfin, l'apparence finale d'une image numérique dépend beaucoup des logiciels et matériels (écrans, imprimante,...) qui vont interpréter le code de l'image. Même si le code de l'image est rigoureusement identique, deux écrans d'ordinateurs différents ne l'afficheront pas nécessairement avec les mêmes couleurs. [7]

1.3.1.3. De l'analogique vers le numérique [7]

En mathématiques, une image est une fonction. Cette fonction quantifie l'intensité lumineuse de n'importe quel point dans l'image. Comme une image est une surface délimitée du plan, on peut se munir d'un repère cartésien et repérer la position de ses points par les coordonnées (x,y) , avec x son abscisse et y son ordonnée. Plus formellement, l'image comme fonction mathématique s'écrit comme ceci :

$$I : \mathbb{R}^2(x,y) \rightarrow \mathbb{R} \quad I(x,y) \quad (\text{Eq 1.1})$$

Une image numérique est une discrétisation de notre modèle continu, Cette discrétisation se fait à la fois sur l'ensemble de définition de la fonction image (échantillonnage, et sur son ensemble d'arrivée (quantification).et enfin l'étape de codage :

L'échantillonnage : consiste à découper notre image continue suivant une grille régulière et à associer une intensité à chacune de ses cellules, appelées pixels .En d'autres termes, on procède à une discrétisation spatiale : seul un nombre fini de points de l'image continue est conservé. Cette étape est conditionnée par le théorème de Shanon .

La Quantification : faire correspondre une valeur discrète à la valeur mesurée de l'échantillon par rapport aux niveaux de quantification.

Le codage : pour coder et stocker dans la mémoire de l'ordinateur la valeur et les coordonnées de chaque point.

1.3.1.4. Représentation d'image numérique [8]

Les images numériques, destinées à être visualisées sur les écrans d'ordinateur, se divisent en 2 grandes classes :

- Les images matricielles
- Les images vectorielles

a. image vectorielle

Une image vectorielle en informatique, est une image numérique composée d'objets géométriques individuels (segments de droite, polygones, arcs de cercle, etc.) définis chacun par divers attributs de forme, de position, de couleur, etc. (définis de manière mathématique).

b. Image matricielle

Encore appelée image bitmap, Une image matricielle est formée d'un assemblage de points ou de pixels.

Nous parlons sur de points lorsque ces images sont imprimées ou destinées à

L'impression (photographies, publicités, cartes etc.) et nous parlons sur de pixels pour les images stockées sous forme « binaire » ou numérique. L'image sous forme d'une matrice existe en trois formes :

- **Image noir et blanc :** chaque pixel prend soit blanc soit noir donc chaque pixel coder en 1 bit. En affectant le 0 pour noir et 1 pour blanc.
- **Images à niveaux de gris :** Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, nous pouvons attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255 donc est codée sur un octet.
- **Image en couleurs :** à représentation usuelle des couleurs y est directement liée au système visuel humain. On mémorise en effet la quantité de rouge, de vert et de bleu (comme le percevaient les trois différents types de cônes de la rétine de l'œil humain). Usuellement, comme pour les niveaux de gris, ces quantités respectives sont mémorisées par 3 octets, représentant chacun respectivement le niveau de rouge, le

niveau de vert et le niveau de bleu d'un pixel. Dans ce cas, l'image est une matrice de triplets RVB.

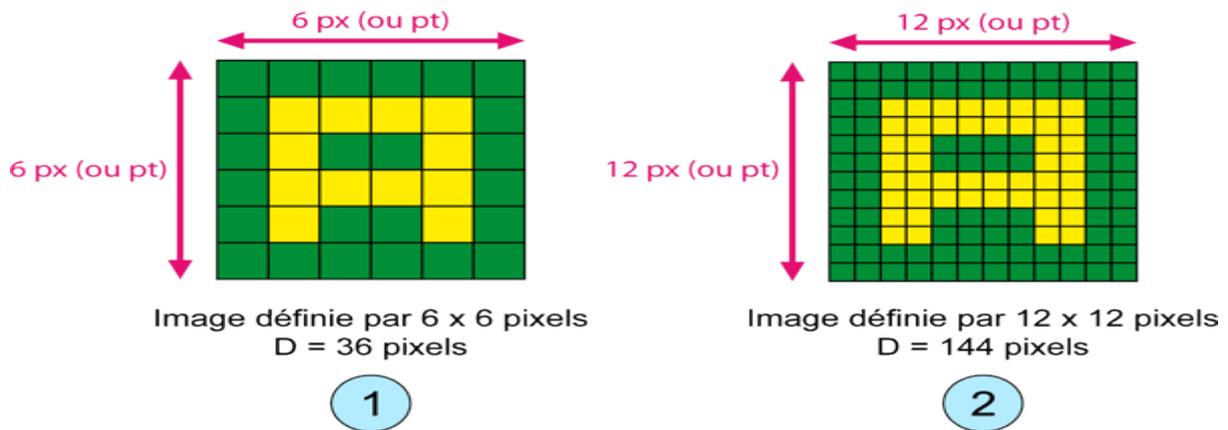


Fig.1.6 : Représentation matricielle d'une image [8].

1.3.1.5. Caractéristiques d'une image numérique [9]

a. Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

b. Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). Nous utilisons aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution.

c. Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

d. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

e. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

1.3.2. Traitement d'image**1.3.2.1. La discipline**

C'est une branche des médias automatisés et des mathématiques appliquées utilisée pour extraire des informations sur l'image et également améliorer sa qualité. Il se compose d'un ensemble de processeurs de signaux dédiés aux images et aux données dérivées telles que la vidéo. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur des données visuelles dans le but de la rendre plus simple, plus efficace et plus agréable, et améliorer l'aspect visuel de l'image d'en extraire des informations jugées pertinentes.

Pour obtenir le meilleur traitement d'image, il faut comprendre la composition de l'image. Les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information, il existe beaucoup de facteurs, les plus importants sont :

- La résolution d'acquisition et le type de codage utilisé lors de la numérisation, à partir lequel nous pouvons savoir la précision des dimensions.
- Les réglages optiques utilisés, pour la netteté de l'image.
- Les conditions d'éclairage.
- Le bruit de la chaîne de transmission d'image.

Parmi les informations les plus importantes que nous pouvons extraire en traitant l'image :

- La luminance
- Le contraste

- Détections des objets

1.3.2.2. Les procédures utilisées dans le système de traitement d'image [10]

a. Les aspects des traitements d'images

Il est commode de subdiviser les différents algorithmes de traitements d'images en des sous classes. il existe différents algorithmes pour différentes tâches et problèmes et souvent nous tenons à distinguer la nature de la tâche à accomplir :

- Amélioration d'images
- Restauration d'images
- Segmentation d'images

Amélioration d'images : Il s'agit du traitement d'une image de sorte que le résultat est plus approprié pour une application particulière exemple : Sharpening, mettre en évidence les contours, amélioration du contraste , ou de l'éclaircissement d'une image et suppression de bruit

Restauration d'images : Cela peut être considéré comme inverser les dommages causés à une image par une cause connue. Exemple : Suppression de flou causé par le mouvement linéaire, Suppression des distorsions optiques et la Suppression des interférences périodique.

Segmentation d'images : Il s'agit de diviser une image en des éléments distinctifs, ou d'isoler certains aspects d'une image. Exemple : trouver des lignes, des cercles ou des formes particulières dans une image. Sur une photographie aérienne, identifier des voitures, arbres, bâtiments, ou des routes.

Ces classes ne sont pas disjointes, un algorithme donné peut être utilisé à la fois pour l'amélioration d'images ou pour la restauration d'images.

Toutefois essayons de faire avec notre image simplement lui donner une meilleure apparence (amélioration), ou la suppression des dommages (restauration).

b. Les étapes du traitement d'images

La figure 1.7 illustre les étapes nécessaires pour accomplir une tâche d'extraction d'une zone choisie dans une image par un processus automatique d'image :

- **L'acquisition de l'image:** Caméra CCD ou scanner.
 - **Prétraitement:** Améliorer le contraste, la suppression du bruit et identifier les régions susceptibles que nous voulons prélever
 - **Segmentation:** Extraction de la section déterminée.
-

- **Représentation et description:** Chercher des courbes, les trous et les coins qui nous permettrons de distinguer les différentes formes de cette région extraite.
- **Reconnaissance et interprétation:** Attribuer des étiquettes à des objets en fonction de leurs descripteurs (de l'étape précédente), et assigner des significations à ces labels.

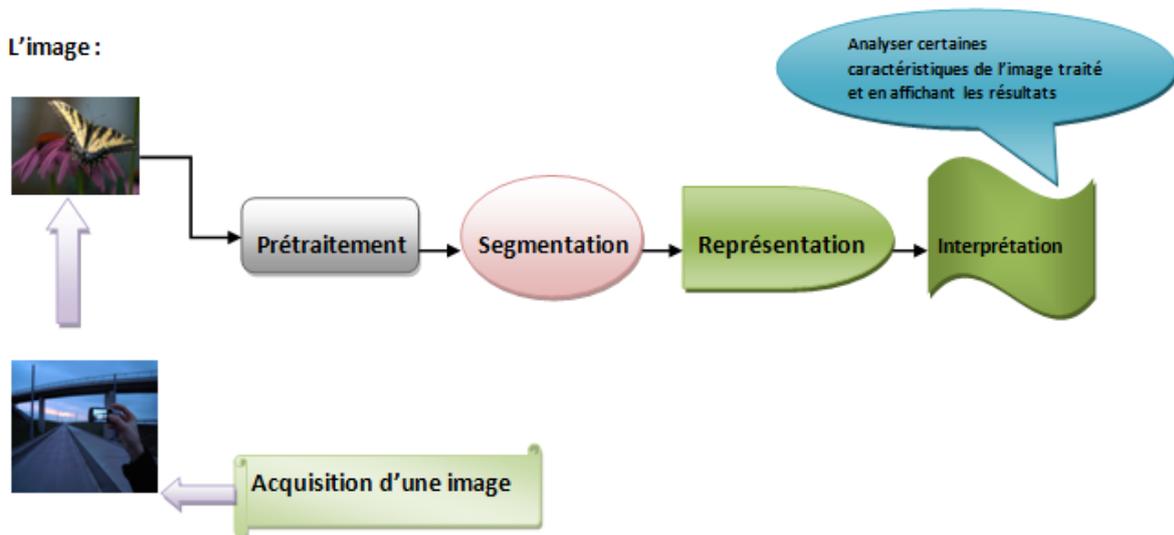


Fig.1.7: Procédure de traitement d'image.

1.3.2.3. Le filtrage

Un filtre est un système servant à séparer des éléments dans un flux. L'action du filtre consiste à retenir, supprimer, rediriger ou modifier les éléments indésirables du flux, et à en laisser passer librement les éléments utiles. Les filtres sont utilisés dans différents domaines.

L'objectif avoué du filtrage dans le traitement d'image est de réduire les variations d'intensité au sein de chaque région de l'image tout en respectant l'intégrité des scènes : les transitions entre régions homogènes, les éléments significatifs de l'image doivent être préservés au mieux. Différentes méthodes de filtrage ont été développées suivant le type et l'intensité du bruit, ou les applications auxquelles on destine l'image. Les premières et les plus simples de ces méthodes sont basées sur le filtrage linéaire stationnaire (invariant par translations), mais les limitations de ces techniques (en particulier leur mauvaise conservation des transitions) a conduit au développement des filtres "non-linéaire".

a. Les filtres linéaires [11]

Le filtre est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage. On distingue deux catégories les filtres lisseurs ou passe-bas

consistant à atténuer les composantes de l'image ayant une fréquence haute (filtre moyenneur, filtre gaussien..) et les filtres passe-haut appelés détecteurs de contours permettent d'accentuer les détails et le contraste

Filtre passe bas :

Filtre moyenneur : Le niveau de gris du pixel central est remplacé par la moyenne des niveaux de gris des pixels environnants. La taille du kernel dépend de l'intensité du bruit et de la taille des détails significatifs de l'image traitée.

Les effets du filtre moyenneur varient avec la taille du noyau : plus les dimensions du noyau seront importantes, plus le bruit sera éliminé ; mais en contrepartie, les détails fins seront eux-aussi effacés et les contours étalés.

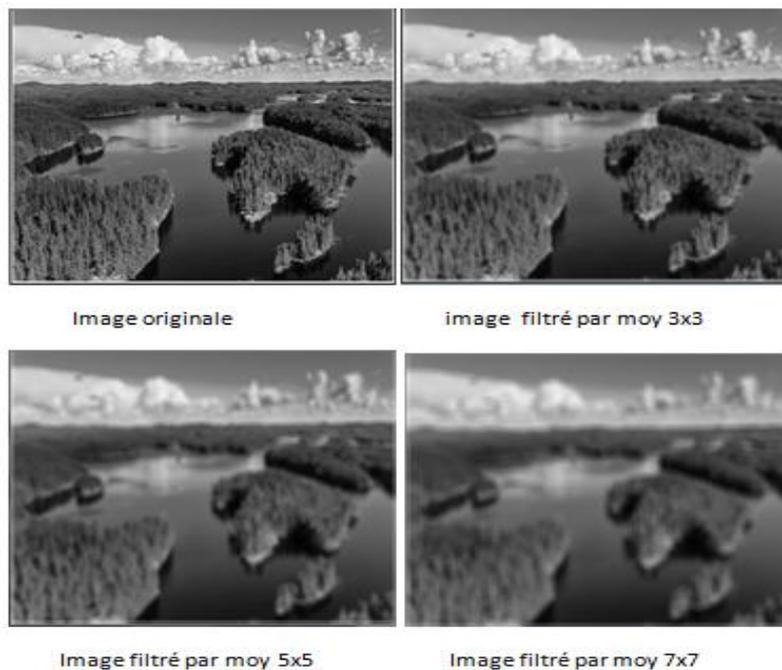


Fig.1.8 : Application de filtre moyenneur sur une image.

Filtre gaussien: un filtre dont la réponse impulsionnelle est une fonction gaussienne ce filtre très populaire utilise la loi de probabilité de Gauss (voir Loi normale multidimensionnelle).

Des approximations de plus en plus précises peuvent être obtenues, selon le théorème central limite par itération de l'un des filtres précédents.

Filtre smooth: Les coefficients du kernel pour un filtre 3*3 sont, les coefficients sont calculés en utilisant des pondérations gaussiennes. des itérations successives permettent d'obtenir le smooth 5*5 (2 itérations) et le smoot 7*7 (3 itérations).

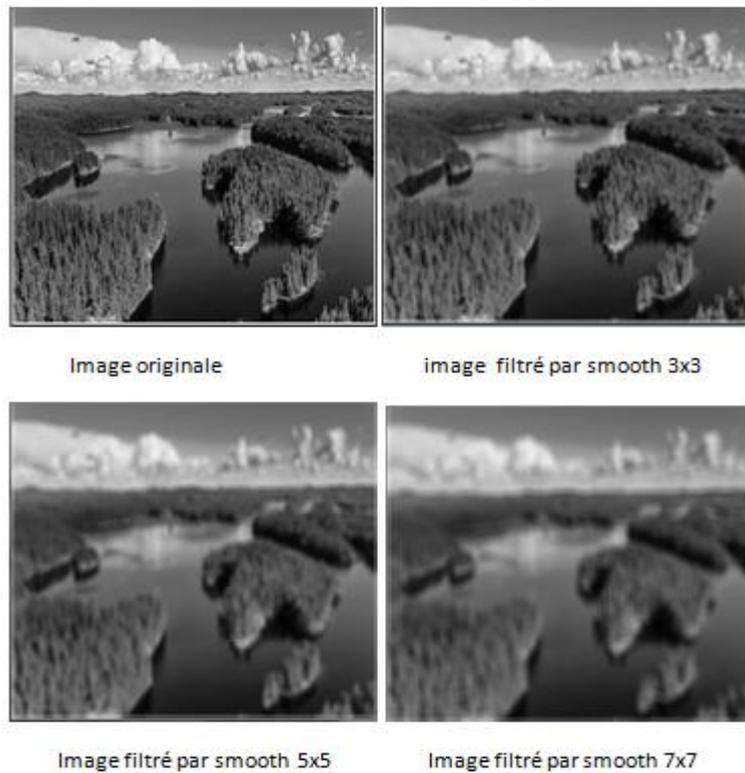


Fig.1.9 : Application de filtre smooth sur une image.

Filtre passe haut : L'image obtenue par un filtre passe-haut correspond en général à ce qui « reste » après un filtrage passe-bas. Ce filtre consiste à détecter les contours à partir d'une dérivation selon les deux coordonnées matricielles. Cette dérivation apparaît comme un filtre passe-haut qui introduit donc du bruit à l'origine de faux contours.

Filtre gradient: le filtre le plus simple consiste à calculer les différences entre pixels voisins sur les horizontales puis sur les verticales. Chaque extremum correspond à un point d'un contour.

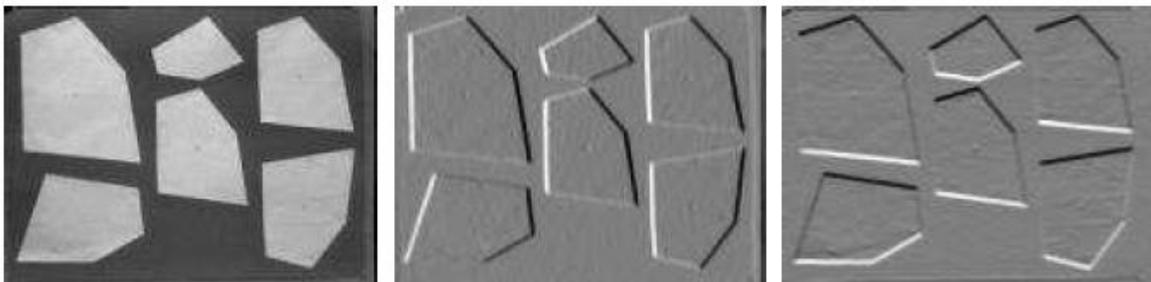


Fig.1.10 : Application de filtre gradient sur une image [12].

Filtre Laplacien : celui-ci se calcule simplement en différences finies et c'est maintenant un changement de signe qui correspond à un point d'un contour.

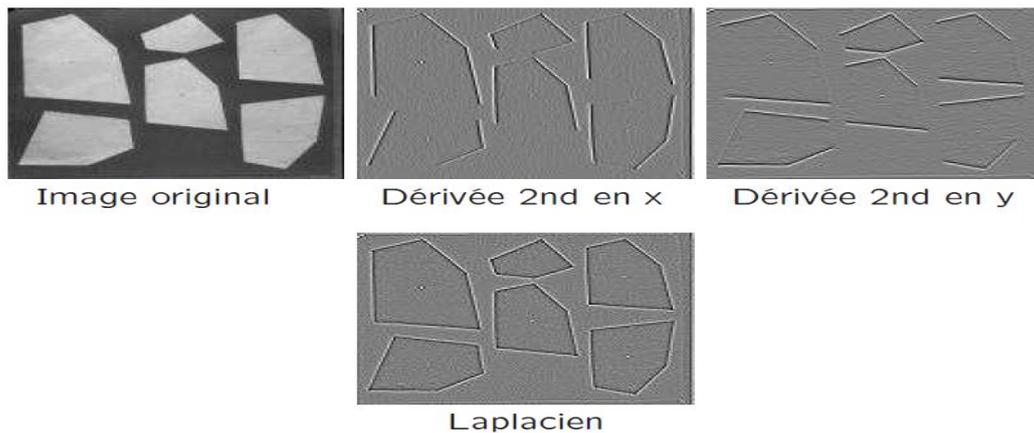


Fig.1.11 : Application de filtre Laplacien sur une image [12].

Filtre Canny: est un algorithme de détection de bord populaire. Il a été développé par John F. Canny, Il s'agit d'un algorithme à plusieurs étapes :

Réduction : Étant donné que la détection des contours est sensible au bruit dans l'image, la première étape consiste à supprimer le bruit dans l'image avec un filtre gaussien 5x5.

Calcul du gradient d'intensité de l'image par la relation::

$$\text{EDJE_GRADIENT}(G) = \sqrt{Gx^2 + Gy^2} \quad (\text{Eq.1.2})$$

Non-maximum Suppression : Après avoir obtenu l'amplitude et la direction du gradient, une analyse complète de l'image est effectuée pour supprimer tous les pixels indésirables qui pourraient ne pas constituer le bord

Seuil d'hystérésis : Cette étape décide quels sont tous les bords qui sont vraiment des bords et lesquels ne le sont pas. Pour cela, nous avons besoin de deux valeurs de seuil, minVal et maxVal. Tous les bords avec un gradient d'intensité supérieur à maxVal sont sûrs d'être des bords et ceux en dessous de minVal sont sûrs d'être non-bords, donc jetés.

b. Les filtres non-linéaires [13]

Les limitations des filtres linéaires ont donc conduit à la conception de filtres non-linéaires. Ces opérateurs ont été développés pour pallier aux insuffisances des filtres linéaires

Filtre médian: Le niveau de gris du pixel central est remplacé par la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur le pixel. La taille du noyau dépend de la fréquence du bruit et des la taille des détails significatifs de l'image traitée. Le médian garde la netteté de l'image pour les éléments de dimensions importantes (par rapport au noyau du filtre), mais élimine les détails fins de manière irrémédiable.

Filtre bilatéral : Le filtre bilatéral utilise un filtre gaussien dans le domaine spatial, mais il utilise également un autre composant de filtre gaussien (multiplicatif) qui est fonction des différences d'intensité des pixels.

1.3.2.4. Compression et décompression d'image [14]

La compression d'image est une application de la compression de données sur des images numériques. Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement.

On peut classer les méthodes de compression en deux types : avec perte et sans perte.

a. La compression sans perte

La compression sans perte signifie que lorsque des données sont compressées et ensuite décompressées, l'information originale contenue dans les données a été préservée. Aucune donnée n'a été perdue ou oubliée. Elles sont utiles lorsqu'on veut garder une grande précision, tel que pour des balayages médicaux. Parmi les méthodes de compression sans perte : RLE et Huffman sont appliquées lors de la compression JPEG.

- **RLE :** appelée codage par plages, est un algorithme de compression de données en informatique basé sur les répétitions. Ainsi, le RLE s'emploie à réduire la taille physique d'une répétition de chaîne de caractère.
- **Huffman :** cet algorithme est capable de faire une analyse statistique de données et ainsi associer aux séquences qui se répètent un code plus court qu'aux séquences plus rares. Elle utilise un dictionnaire de codes où sont stockées les données issues de l'analyse statistique et ce dictionnaire doit être conservé entre la compression et la décompression.

b. La compression avec perte

La compression avec perte autorise une dégradation de l'image pour diminuer l'empreinte mémoire. En effet, elle permet la suppression des données afin de diminuer l'espace de stockage nécessaire pour un fichier d'image. Après la décompression on va obtenir une image différente de ce qu'elle était avant la compression mais cette différence passe inaperçue. Les méthodes les plus importantes de compression avec perte sont :

- la réduction de l'espace des couleurs aux couleurs les plus fréquentes dans une image. Les couleurs choisies sont indiquées dans la palette de couleur dans l'en-tête de l'image compressée.
- le sous-échantillonnage de la chrominance. Cette méthode profite du fait que l'œil perçoit la luminosité plus vigoureusement que la couleur, en laissant tomber au moins la moitié des informations de chrominance dans l'image.
- le codage par transformation. C'est généralement la méthode la plus utilisée. La transformée en cosinus discrète et la transformation par ondelettes sont les transformations les plus populaires. Le codage par transformation comprend l'application de la transformation à l'image, suivie d'une quantification et d'un codage entropique.

c. Norme JPEG

JPEG est une méthode de compression non conservative de l'image, elle s'appuie sur une analyse de la perception de l'œil humain, ainsi que sur l'usage de codeurs classiques (RLE, et Huffman), ce format a l'avantage de fournir des images de bonne qualité et de petite taille, donc particulièrement utile sur Internet.

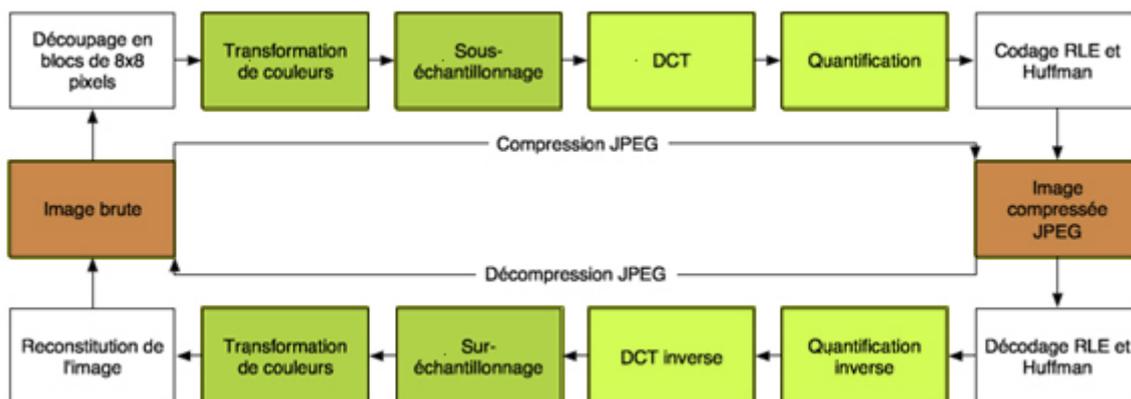


Fig.1.12 : Représentation des opérations compression et décompression [14].

1.3.2.5. Les applications possibles avec le traitement d'image

- **Reconnaissance de l'écriture** : La reconnaissance de l'écriture manuscrite progresse de jour en jour. Elle est suffisamment opérationnelle pour que la majorité des adresses, même manuscrites, soient reconnues automatiquement sur le courrier postal.
- **Analyse de la vidéo** : L'objectif de cette technique devenue une discipline depuis les années 2000 (lorsque la puissance des processeurs peu onéreux et en particulier des PC a permis des traitements puissants en temps réel) est d'interpréter les faits observés

à l'image afin de signaler ou d'enregistrer des faits marquants. Le plus souvent, la caméra est fixe et observe les mouvements d'une scène. Les applications sont nombreuses : protection des biens (détection d'intrusion, détection d'objet abandonné ou déposé...), identification (biométrie faciale), sécurité des personnes (détection de chutes de personnes, franchissement de rambardes...), animations (planchers animés selon les mouvements des danseurs en boîte de nuit), détection de feux (industriel, forêts, tunnels...), surveillance de tunnels (comptage, mesure de vitesse, détection de fuites/anomalies dans les plafonds), surveillance de tuyaux et autres process industriels...

- **Le système de reconnaissance faciale** : La reconnaissance de visage a de nombreuses applications en vidéosurveillance, biométrie, robotique, indexation d'images et de vidéos, recherche d'images par le contenu, etc. Ces systèmes sont généralement utilisés à des fins de sécurité pour déverrouiller
- **La reconnaissance d'objets** : est une branche de la vision artificielle et un des piliers de la vision industrielle. Elle consiste à identifier des formes pré-décrites dans une image numérique, et par extension dans un flux vidéo numérique
- **Modélisation d'objets ou d'environnements** : p. ex. dans l'analyse d'image médicale ou la topographie ;
- **Interaction en tant que dispositif d'entrée** : pour une interaction homme-machine

1.3.3. Présentation du langage Python

Python est un langage de programmation libre (open source) d'utilisation générale et multiplateforme. Il fonctionne sous un grand nombre de plateformes incluant Linux, Windows, Mac OS X, OS/2 ainsi que les nombreuses variantes dérivées d'Unix.

Il est parfois considéré comme un langage réservé au script, à tort, car il offre un grand confort dans le développement d'applications de divers types.

Python est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles.

1.3.3.1. Caractéristiques du langage Python [15]

Le langage python est caractérisé par

- Python est portable, non seulement sur les différentes variantes d'Unix, mais aussi sur les OS propriétaires : Mac OS, BeOS, NeXTStep, MS-DOS et les différentes variantes de Windows.
- Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux.
- La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires...), conduit à des programmes à la fois très compacts et très lisibles.
- Python est un langage qui continue à évoluer, soutenu par une communauté d'utilisateurs enthousiastes et responsables, dont la plupart sont des supporters du logiciel libre. Parallèlement à l'interpréteur principal, écrit en C et maintenu par le créateur du langage, un deuxième interpréteur, écrit en Java, est en cours de développement.
- Python contient une variété de bibliothèques et d'extensions dans tous les domaines.
- Python est dynamique (l'interpréteur peut évaluer des chaînes de caractères représentant des expressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflexif (il supporte la métaprogrammation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même).
- Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.

1.3.3.2. Les environnements de développements

Un IDE ou un Environnement de développement est un environnement de programmation complet qui se présente sous la forme d'une application. Il se compose généralement d'un éditeur de code, d'un compilateur, d'un débogueur et d'un générateur d'interface graphique. Les meilleures IDE pour la langage python sont :

Atom : Atom est un éditeur de code dont le développement a été lancé en février 2014 par GitHub et est sous License MIT. Il a été construit à partir du code de Chromium et de

node.js et est disponible sous Windows, Linux et Mac. Pour faire évoluer Atom en Atom-IDE, GitHub et Facebook ont alors décidé de lancer une collaboration pour faire passer Atom d'un simple éditeur de code à un environnement de développement intégré abouti. Atom-IDE peut déjà prendre en charge plusieurs langages tels que TypeScript, Flow, Javascript ou encore PHP et bénéficie d'un UI très ergonomique et particulièrement agréable.

Visual studio : Visual Studio est une suite de logiciels de développement pur Windows et Mac OS conçue par Microsoft. La dernière version s'appelle Visual Studio 2017. La première version a été publiée en 1997 et se nommait Visual Studio 97. Depuis plusieurs versions sont sorties dont la dernière, la version 2017 15.6.4 qui est sortie le 22 mars 2018. Visual Studio permet de générer des applications Web ASP.NET, des services web XML, des applications bureautiques ou encore des applications mobiles. Visual Studio dispose des fonctionnalités d'un IDE de base et depuis la version 2015 de nouvelles fonctionnalités ont été ajoutées, tels que la possibilité de développer sur plusieurs plateformes mobiles (IOS, Android, Windows Phone), ou encore l'ajout de la prise en charge du débogage des applications DirectX 12 ou l'ajout d'analyse dynamique et bien d'autres.

Pycharm : Il est multi-plateforme et fonctionne sur Windows, Mac OS X et Linux. Il est disponible en trois versions, la version Community sous licence Apache, la version Educational (Edu) et la version Professional sous licence propriétaire. Les deux premières versions sont open source, donc gratuites alors que la version Professional est payante. La version Community, celle qui nous intéresse, possède différentes fonctionnalités tels que la coloration syntaxique, l'auto-complétion ou encore la vérification de code en direct. La version payante possède évidemment des fonctionnalités que la version gratuite ne possède pas tels que la gestion complète de bases de données ou encore une panoplie de frameworks plus importante que sur la version communautaire.

1.3.4. L'apprentissage profond (Deep learning)

Tous d'abord nous examinerons le concept d'intelligence artificielle car il s'agit d'un vaste domaine à travers lequel nous essayons d'imiter le comportement humain dans le but de mettre les connaissances dans la machine et accomplir de nombreux types de tâches telles que la résolution des problèmes

L'intelligence artificielle constitue d'un champ des apprentissages automatiques. Ainsi, avec cette vague de techniques avancées d'apprentissage machine qui donne à l'IA un pas en avant, où est la place de DL (deep learning). Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par : [16]

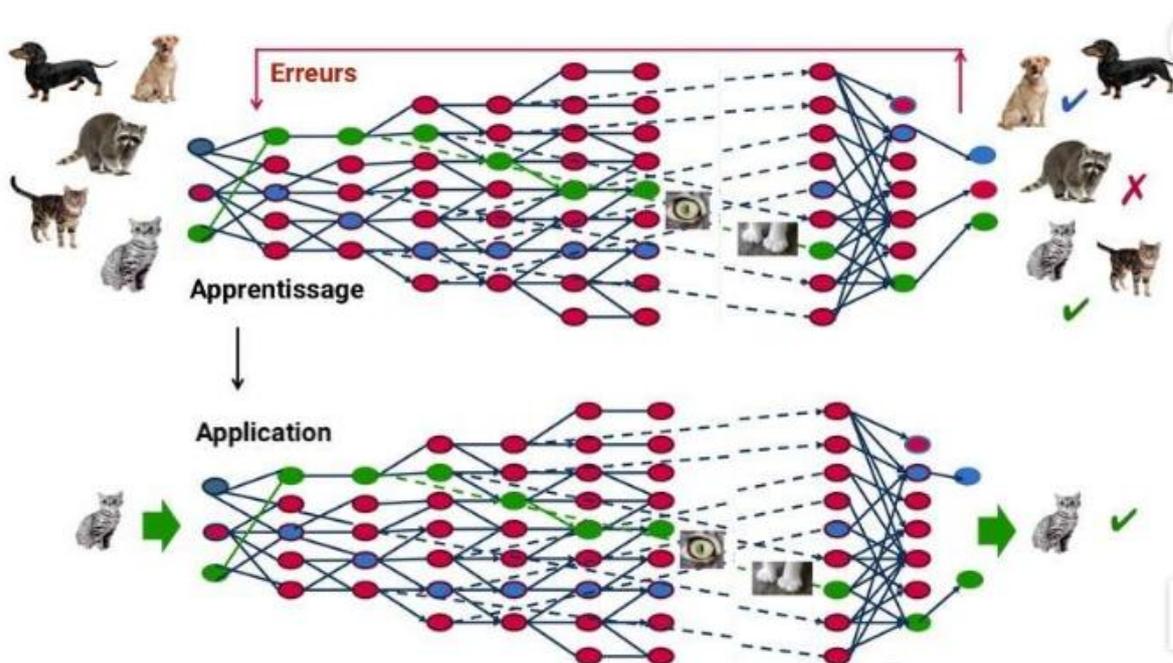


Fig.1.13 : Schéma d'un réseau d'apprentissage profond [17].

Il existe différents algorithmes de Deep Learning. Nous pouvons ainsi citer :

1.3.4.1. Les réseaux de neurones [17]

Les réseaux de neurones n'ont que très peu à voir avec le système neuronal et le cerveau. L'analogie entre un neurone et un réseau de neurones à une couche est essentiellement graphique, dans la mesure où il y a un flux d'information d'un bout à l'autre du réseau.

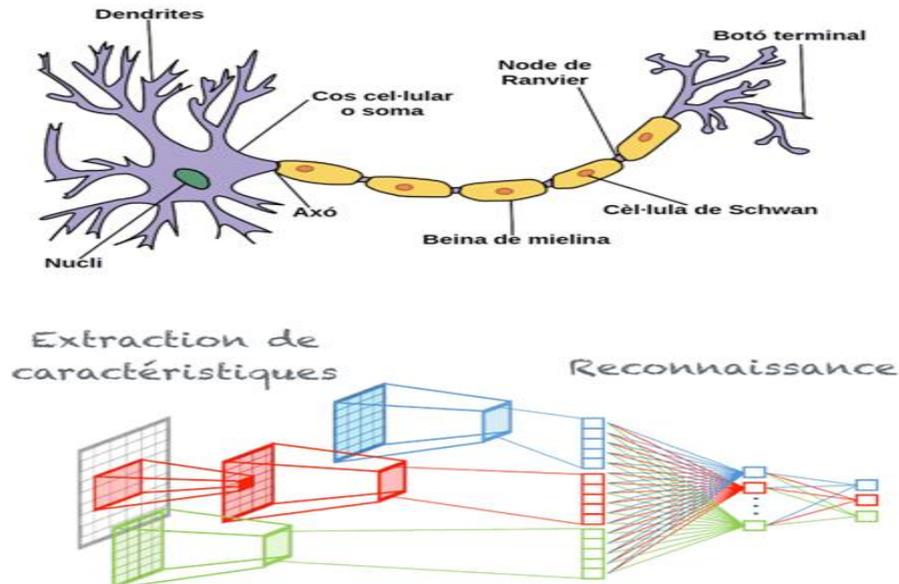


Fig.1.14 : Schéma d'un réseau de neurones d'humain et de machine [18].

La première couche d'un réseau de neurones est celle d'entrée (input). C'est par cette couche que vont rentrer les données dont nous disposons. Avant de pouvoir "nourrir" le réseau, il faudra préalablement transformer vos données en nombres si elles n'en sont pas déjà.

Et toutes les couches entre la couche d'entrée et de sortie, des couches dites "cachées", sont autant de représentations différentes des données..

a. Les reseaux de neurones dits convolutifs (CNN)

Ces réseaux sont utilisés pour tout usage autour de l'image ou de la vidéo dont fait partie la reconnaissance faciale ou encore la classification d'image. L'entreprise Bai Du (l'équivalent de Google en Chine) a par exemple mis en place des portiques actionnés par la reconnaissance visuelle qui laissent passer uniquement leurs employés. Snapchat et de nombreuses applications mobiles ont utilisé la percée de l'apprentissage et des CNN pour augmenter leurs fonctionnalités de "filtres".

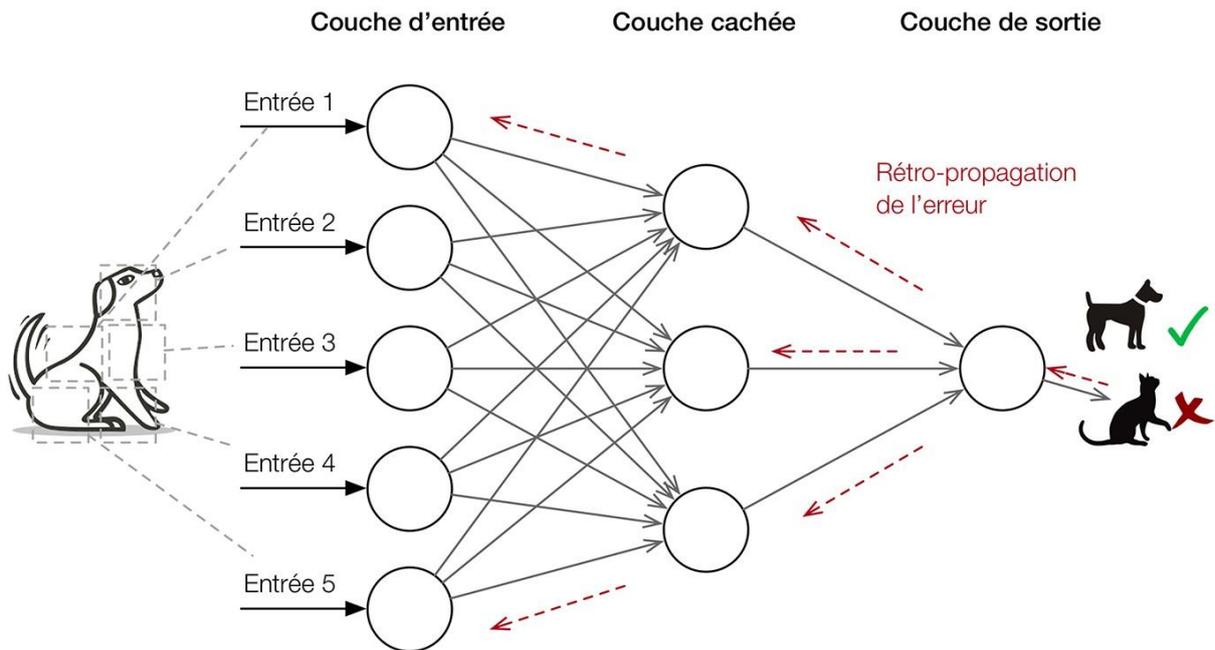


Fig.1.15: Schéma d'un réseau de neurones avec ses couches. [19]

Le nom réseau convolutif renvoie à un terme mathématique: le produit de convolution. En termes simples, l'idée est qu'on applique un filtre à l'image d'entrée, les paramètres du filtre seront appris au fur et à mesure de l'apprentissage. Un filtre appris permettra par exemple de détecter les angles dans une image si les angles servent à classifier au mieux l'image.

L'image est d'abord décomposé dans les 3 canaux (R,G,B) pixels par pixels, on obtient donc 3 matrices de taille $n \times n$ (où n est le nombre de pixels).

b. Les reseaux de neurones dits récurrents (RNN)

Les réseaux de neurones récurrents sont au cœur de bon nombre d'améliorations substantiels dans des domaines aussi divers que la reconnaissance vocale, la composition automatique de musique, l'analyse de sentiments, l'analyse de séquence ADN, la traduction automatique. La différence principale avec les autres réseaux de neurones vient du fait que ces derniers tiennent compte de l'enchaînement successif des données, bien souvent de leur enchaînement dans le temps. Par exemple dans le cas de l'analyse d'une série de mesures de capteurs (séries temporelles) le réseau aura encore en mémoire tout ou en partie des observations précédentes.

c. Les autos encodeurs

Les auto encodeurs sont appliqués principalement à la détection d'anomalie (par exemple pour détecter la fraude en banque ou bien pour trouver des anomalies dans une ligne de

production industrielle). Ils peuvent également servir à la réduction de dimension (proche dans l'idée d'une Analyse en Composante Principale). En effet le but des auto encodeurs est d'apprendre à la machine en quoi consiste des observations "normales". L'architecture de notre réseau est la suivante:

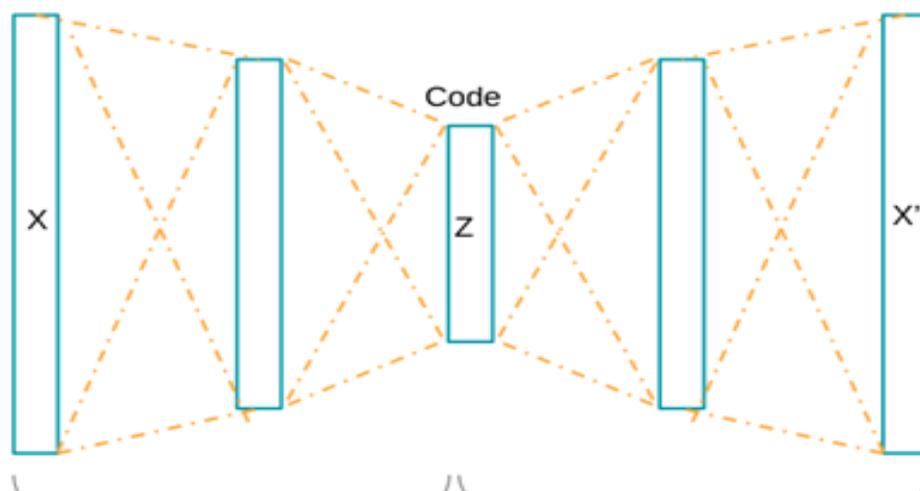


Fig.1.16: schéma d'un réseau de neurones encodeur avec ses couches [17].

Le deep Learning est utilisé dans de nombreux domaines :

- reconnaissance d'image
- traduction automatique
- voiture autonome
- diagnostic médical

1.4. Base de données MySQL [20]

1.4.1. Base de données

Une base de données informatique est un ensemble de données qui ont été stockées sur un support informatique, organisées et structurées de manière à pouvoir facilement consulter et modifier leur contenu.

Une base de données seule ne suffit donc pas, il est nécessaire d'avoir également :

- un système permettant de gérer cette base ;
- un langage pour transmettre des instructions à la base de données (par l'intermédiaire du système de gestion).

1.4.2. Langage MySQL

MySQL est donc un Système de Gestion de Bases de Données Relationnelles qui utilise le langage SQL. C'est un des SGBDR les plus utilisés. Sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel open source, ce qui signifie que son code source est librement disponible et que quiconque en ressent l'envie et/ou le besoin peut modifier MySQL pour l'améliorer ou l'adapter à ses besoins. Une version gratuite de MySQL est par conséquent disponible. À noter qu'une version commerciale payante existe également.

1.4.3. SGBD

Un Système de Gestion de Base de Données (SGBD) est un logiciel (ou un ensemble de logiciels) permettant de manipuler les données d'une base de données. Manipuler, c'est-à-dire sélectionner et afficher des informations tirées de cette base, modifier des données, en ajouter ou supprimer.

1.5. Interface graphique

1.5.1. Application desktop

Une application bureautique est une représentation graphique d'un programme ou d'un logiciel qui réalise différentes tâches dans un système exploitation. Il permet la création d'un environnement interactif entre l'utilisateur et le système.

1.5.2. Réalisation d'une application bureautique

La modélisation d'une application est basée sur le rôle de laquelle on veut réaliser .Il est soutenue par une interface graphique interactive à travers les étapes suivantes :

- Création des bases de données
- Conception d'interface graphique
- Connexion entre la base de données et l'interface graphique

1.5.3. Application de base de données

Les applications de base de données font appel à un logiciel moteur – le SGBD – pour définir, stocker, retrouver et manipuler le contenu de la base de données, ainsi que contrôler la redondance des données, imposer de la cohérence, et partager les informations entre plusieurs utilisateurs. Dans ce type d'application, les utilisateurs expérimentés peuvent manipuler directement le SGBD, pour répondre à des demandes complexes, tandis que les demandes

ordinaires sont effectuées en utilisant l'application et ne requièrent pas de connaissances techniques.

1.6. Conclusion

Enfin , nous avons pu voir les différents systèmes d'identification des plaques d'immatriculation et leur utilisation pour pouvoir organiser les parkings grâce au traitement d'image supporté par un langage de programmation de haut niveau.

Chapitre 2

L'analyse et
l'implémentation de
système de détection

2.1. Introduction

Ce chapitre présente diverses méthodes de développement du système de détection de matricule, où nous présenterons deux modèles de ce système, un modèle programmé en standard et un autre soutenu par une technologie d'apprentissage approfondie et en détaillant les étapes de chaque modèle.

2.2 .Outils de développement

Le système sur lequel nous comptons pour détecter les panneaux de numérotation et la régulation du stationnement est divisé en trois parties où chaque partie se distingue par ses propres mécanismes et ceci en fonction de la nature de la tâche effectuée. Les étapes de ce projet commencent par la capture de la plaque de numérotation, puis la détection de la plaque d'immatriculation à partir de l'image capturée et enfin l'extraction optique des caractères depuis la plaque de matricule détecté.

Chaque étape correspond aux commandes, instructions et outils présentés ci-dessous.

2.2.1. Bibliothèque Open CV

Open CV est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. Tous ses travaux et recherches sont dans le domaine open source. Elle met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes en partant des données brutes et propose la plupart des opérations classiques en traitement des images et des vidéos :

- lecture, écriture et affichage d'une image ;
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs: lissage, filtrage ;
- seuillage d'image (méthode d'Otsu, seuillage adaptatif)
- détection de droites, de segment et de cercles par Transformée de Hough
- cascade de classifieurs.
- détection de mouvement, historique du mouvement
- détection de points d'intérêts
- Toutes les opérations matricielles
- les applications d'apprentissage machine

2.2.2. Bibliothèque Numpy

Numpy est un package open source utilisé pour le calcul scientifique avec python. Cette bibliothèque s'applique aux images numériques et contient différentes fonctions permettant de créer ou sauvegarder un tableau et manipuler des vecteurs, matrices et polynômes.

2.2.3. Bibliothèque Imutils

C'est une série de fonctions de traitement d'image de base destinées surtout pour la rotation, le redimensionnement, la squelettisation et l'affichage des images.

2.2.4. Bibliothèque pytesseract

Python-tesseract est un outil de reconnaissance optique de caractères (OCR) pour python. Autrement dit, il reconnaîtra et «lira» le texte intégré aux images. Il est également utile en tant que script d'invocation autonome pour tesseract, car il peut lire tous les types d'images pris en charge par les bibliothèques d'imagerie Pillow et Leptonica, y compris jpeg, png, gif, bmp, tiff et autres. De plus, Python-tesseract imprimera le texte reconnu sous forme d'une chaîne de caractère.

2.2.5. Bibliothèque matplotlib

C'est un package open source qui sert à des visualisations statiques, animées et interactives en Python.

2.2.6. Bibliothèque urllib

C'est un paquet qui collecte plusieurs modules travaillant avec les URLs :

- urllib.request pour ouvrir et lire des URLs ;
- urllib.error contenant les exceptions levées par urllib.request ;
- urllib.parse pour analyser les URLs ;
- urllib.robotparser pour analyser les fichiers robots.txt.

2.2.7. Bibliothèque tensorflow

TensorFlow est une bibliothèque open source d'apprentissage automatique. Il s'agit d'une boîte à outils et un écosystème complet et flexible de ressources communautaires qui permet aux chercheurs de résoudre des problèmes mathématiques extrêmement complexes avec aisance et aux développeurs de créer et de déployer facilement des applications propulsées par ML

2.2.8. La technique de détection d'objets SSD (framework SSD) [21]

Le framework SSD est une méthode pour détecter des objets dans des images en utilisant un seul réseau de neurones profond. L'approche SSD discrétise l'espace de sortie des boîtes englobantes en un ensemble de boîtes par défaut sur différents rapports d'aspect et échelles par emplacement de carte d'entités. Au moment de la prédiction, le réseau génère des scores pour la présence de chaque catégorie d'objet dans chaque boîte par défaut et produit des ajustements à la boîte pour mieux correspondre à la forme de l'objet. Il élimine complètement la génération de propositions et l'étape de rééchantillonnage de pixels ou de fonctionnalités ultérieure et encapsule tous les calculs dans un seul réseau. De plus, le réseau combine les prévisions de plusieurs cartes d'entités avec différentes résolutions pour gérer naturellement des objets de différentes tailles. Cela rend SSD facile à se former et simple à s'intégrer dans les systèmes qui nécessitent un composant de détection.

Le nom SSD est l'abréviation de::

- **Single Shot:** cela signifie que les tâches de localisation et de classification des objets se font en une seule passe directe du réseau
- **MultiBox:** c'est le nom d'une technique de régression de boîte englobante développée par Szegedy. Dans MultiBox, les chercheurs ont créé les prieurs qui sont des boîtes englobantes de taille fixe précalculées qui correspondent étroitement à la distribution des boîtes de vérité au sol d'origine. Ils sont sélectionnés de telle manière que leur rapport Intersection ou Union (IOU) soit supérieur à 0,5. n'est pas toujours suffisant, mais il fournit cependant un bon point de départ pour l'algorithme de régression.
- **Détecteur:** le réseau est un détecteur d'objets qui classe également les objets détectés

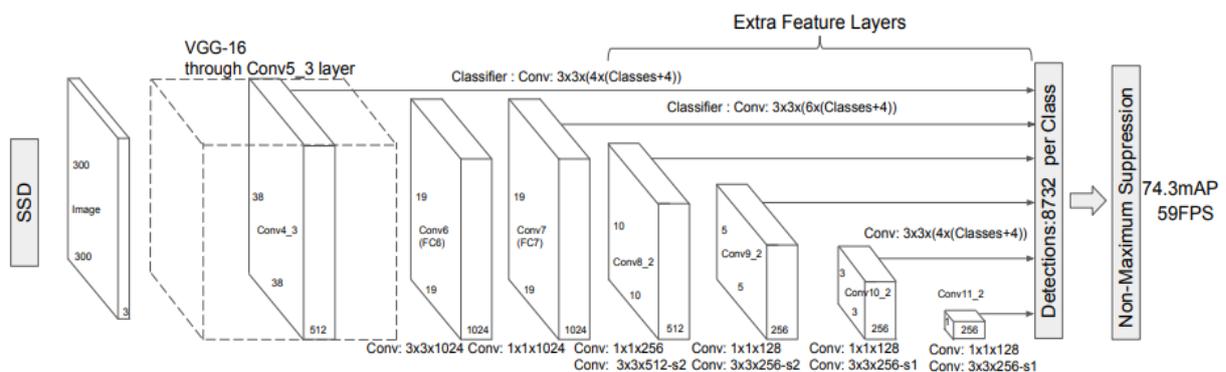


Fig.2.1 : Architecture de SSD [21].

2.2.9. Bibliothèque MySQL

C'est un package virtuel du langage sql .Il fonctionne pour relier la base de données à l'application bureautique, grâce à des commandes et des fonctions qui gèrent les données.

2.2.10. PHP MyAdmin

C'est une application Web pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP. Elle permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données. Elle permet aussi l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder une base de données sous forme de fichier .SQL et d'y transférer ses données, même sans connaître SQL.

2.2.11. Bibliothèque PyQt5

PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt . Il permet ainsi de créer des interfaces graphiques en Python. Une extension de Qt Creator (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python d'interfaces graphiques.

2.2.12. Qt designer

Qt Designer est un programme qui permet de construire rapidement les fenêtres à la souris. Il nous évite l'écriture de nombreuses lignes de code. Ce n'est pas un programme magique qui réfléchit à notre place mais il nous permet simplement de gagner du temps et d'éviter les tâches répétitives d'écriture du code de génération de la fenêtre.

Qt Designer existe sous forme de programme indépendant mais il est aussi intégré au sein de Qt Creator,. En effet, Qt Designer est complètement intégré dans Qt Creator. Comme c'est le plus simple et que cette solution n'a que des avantages, nous allons donc travailler directement dans Qt Creator..

Lorsqu'on crée notre fenêtre, Qt Designer s'ouvre au sein de Qt Creator comme le montre la figure suivante:

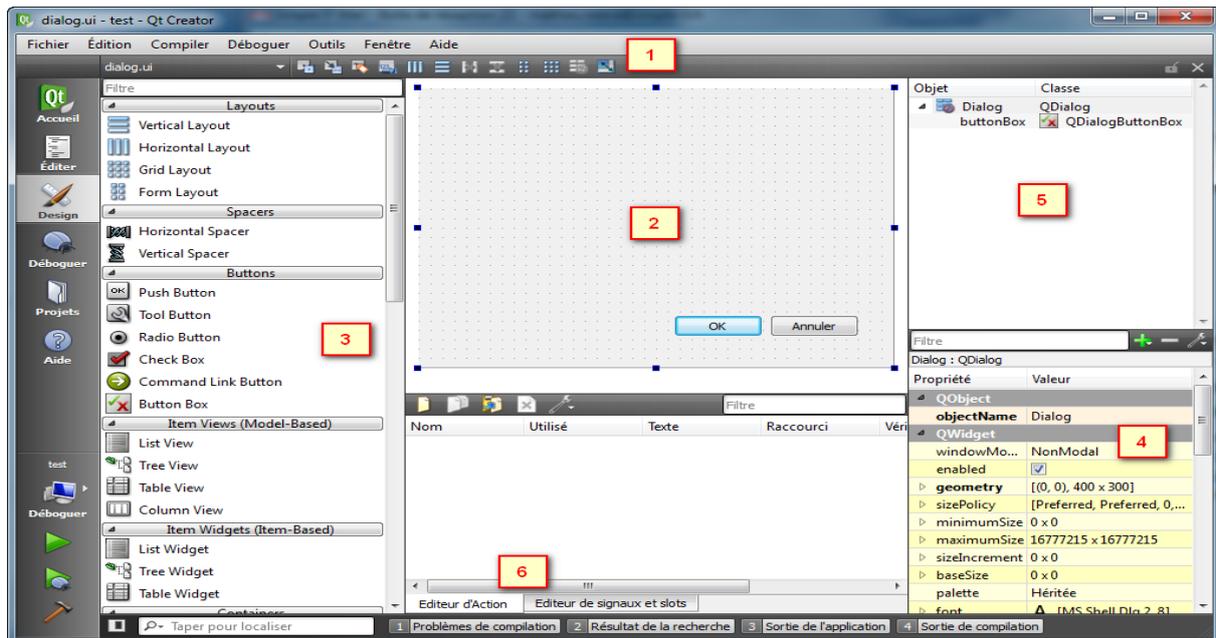


Fig.2.2 : Fenêtre de QT designer.

Détaillons chacune des zones importantes dans l'ordre :

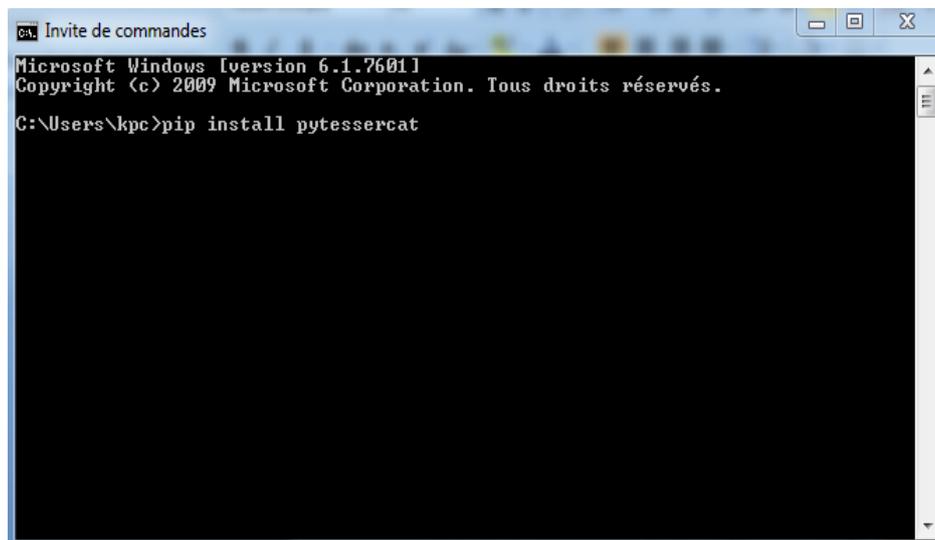
- La zone « (1) » présente la barre d'outils, où se trouve des boutons importants qui nous permet de passer d'un mode d'édition à un autre.
- La zone « (2) » est la fenêtre principale dans laquelle on crée et on dessine notre interface souhaitée..
- La zone « (3) » présente le widget box, ce dock nous donne la possibilité de sélectionner un widget à placer sur la fenêtre. Pour mettre un de ces widgets sur la fenêtre, il suffit de faire un glisser-déplacer. On a un large choix d'outils à utiliser. Heureusement, ceux-ci sont organisés par groupes pour les voir plus clairement.
- la zone « (4) » s'appelle Property Editor ,dans cette fenêtre on peut modifier les propriétés d'un widget qu'on a sélectionné sur notre fenêtre principale ,à noter que les widgets possèdent en général beaucoup de propriétés, et que celles-ci sont organisées en fonction de la classe dans laquelle elles ont été définies..
- la zone « (5) » s'appelle Object Inspector ,cet fenêtre affiche la liste des widgets placés sur la fenêtre, en fonction de leur relation de parenté, sous forme d'arbre.

- la zone « (6) » s'appelle Éditeur de signaux/slots et éditeur d'action : ils sont séparés par des onglets. L'éditeur de signaux/slots est utile si vous avez associé des signaux et des slots, les connexions du widget sélectionné apparaissant ici.

2.3. La procédure de système de détection de matricule

Grace aux moyens et outils mentionnés ci-dessus, nous avons pu développer deux programmes tels que le premier est préliminaire et le deuxième est entraîné par l'apprentissage au profond qui détecte la plaque d'immatriculation.

L'utilisation de ces moyens exige l'importation des bibliothèques open source dans le langage de programmation python, et cela en les téléchargeant depuis cmd.



```
cmd: Invite de commandes
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Users\kpc>pip install pyesseract
```

Fig.2.3 : fenêtre d'invitation des commandes.

Dans cette section, on verra comment incarner ces mécanismes pour mettre en œuvre les différentes théories du traitement d'image et vidéo pour l'extraction du matricule puis on a amélioré l'efficacité de notre processus et on va mentionner également le principe de fonctionnement de la base de données et l'interface graphique.

2.3.1. Traitements réalisés par l'algorithme

Afin d'atteindre le meilleur résultat pour la capture du matricule, notre travail a été divisé en deux phases dont la première consiste à développer un programme spécial permettant l'identification et la reconnaissance de la plaque d'immatriculation. Alors que, la deuxième étape représente un autre modèle entraîné par l'apprentissage approfondi. Ce dernier, est rapide et efficace, fonctionnant en temps réel pour découvrir la plaque de numérotation. Enfin, nous avons attaché chaque étape à une base de données et une interface graphique qui sont utilisés dans une application bureautique pour gérer le parking.

2.3.1.1 Algorithme de prototype (1^{er} modèle)

Dans ce premier modèle, qui n'est pas encore entraîné, nous passerons en revue les étapes principales du traitement d'image. Chaque étape étant attachée à un processus sur l'image matricielle pour illustrer l'effet des transformations et les filtres sur les pixels. Nous pouvons convertir diverses images en matrices 2D grâce à ce que la bibliothèque numpy fournit.

a.L'acquisition

L'acquisition et la capture des images s'effectue au minimum en deux étapes. La première étape représente la création d'un réseau local sans fil (WLAN) afin d'établir une connexion entre les caméras-ip sans fil et notre programme. Afin de réaliser cette tâche, on va nous allons utiliser un point d'accès et deux caméras-ip. Dans notre cas, on va utiliser deux Smartphones en tant que caméra-ip sans fil et, à l'aide d'une application Android qui se nomme « ip-webcam », cette dernière donne les fonctionnalités d'une caméra ip wifi aux Smartphones en affectant une adresse ip aux appareils et ensuite utiliser ces caméras pour faire une vision en temps réel du contenu visionné. une fois la tâche accomplie, l'utilisation de la bibliothèque (urllib) aura lieu, l'importation de cette bibliothèque va servir à importer l'adresse url de flux généré par la caméra du Smartphone pour visualiser le contenu en temps réel.

Une fois le véhicule positionné dans l'endroit convenable la plaque d'immatriculation est automatiquement détectée, une frame de la vidéo sera capturée dans un chemin bien défini.

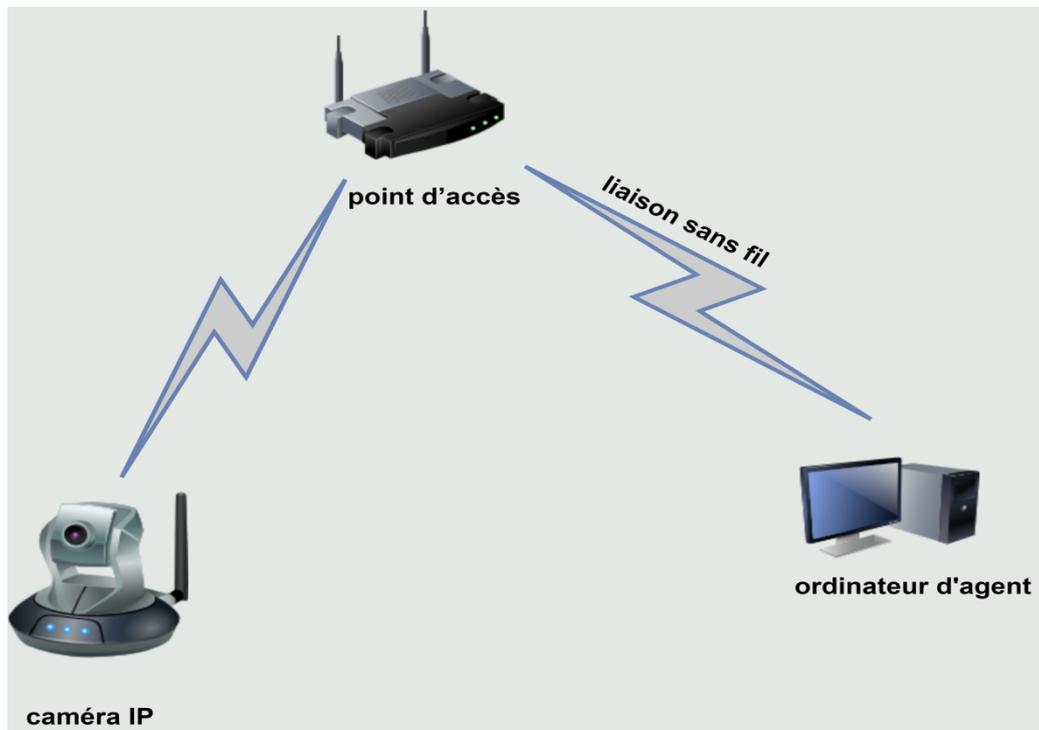


Fig .2.4 : Représentation du réseau local (sans fil).

b. Détection de matricule

La plaque d'immatriculation est l'une des choses que l'on trouve dans la voiture et par conséquent de nombreux facteurs physiques l'entourent qui changent constamment. Il existe de nombreuses interférences dans les images originales de la voiture, telles que la taille de l'image, l'éclairage et la qualité de l'image, qui influencent sérieusement les performances de reconnaissance. Les variations des types de plaques ou des environnements posent des problèmes de détection. Donc, l'objectif principal est d'expérimenter en profondeur et de trouver des solutions alternatives aux problèmes de segmentation d'image et de la reconnaissance de plaque d'immatriculation dans le cadre de reconnaissance de caractères contenu dans le matricule. En combinant de nombreuses techniques de détection d'objets et de traitement d'images, nous en tirons les trois étapes suivantes comme solution initiale:

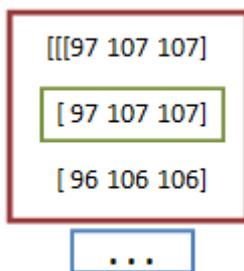
- **Prétraitement de l'image originale de la voiture** : le contenu de l'image du véhicule est lu et pris comme une photo compressée qui sera convertie en matrice:

```

[[ [ 97 107 107 ]
 [ 97 107 107 ]
 [ 96 106 106 ]
 ...
 [132 140 139 ]
 [132 140 139 ]
 [132 140 139 ] ]
[[ [ 97 107 107 ]
 [ 97 107 107 ]
 [ 96 106 106 ]
 ...
 [132 140 139 ]
 [132 140 139 ]
 [132 140 139 ] ]
[[ [ 97 107 107 ]
 [ 97 107 107 ]
 [ 96 106 106 ]
 ...
 [132 140 139 ]
 [132 140 139 ]
 [132 140 139 ] ]

```

Fig.2.5 : Matrice d'une région obtenue d'une image réelle.



Les dimensions de l'image sont très grandes donc dans cet exemple, nous pouvons manipuler un échantillon de la matrice représentée par une ligne et par cela, nous notons les changements dans chaque étape des transformateurs et des filtres.

La valeur du pixel est encadrée en vert et les trois valeurs 97, 107,107 sont les composantes R G B respectivement. L'encadrement en violet, représente la ligne de la matrice, correspondant aux 3 pixels et chaque pixel est représenté par 3 octets. Les trois points indiquent une prochaine ligne. Il existe aussi une opération de redimensionnement pour l'adaptation de l'image.

Ensuite, nous faisons la conversion RGB en niveau de gris afin de simplifier et minimiser les informations dans l'image. Chaque pixel, de la matrice obtenue, est représenté par un octet. C'est à dire on ne garde que l'information de luminance.

```

[[106 106 105 ... 139 139 139 ]
 [106 106 105 ... 139 139 139 ]
 [106 106 105 ... 139 139 139 ]
 ...
 [161 161 161 ... 124 124 124 ]
 [161 161 161 ... 124 124 124 ]
 [161 161 161 ... 124 124 124 ] ]

```

Fig.2.6 : Matrice d'une région d'une image en niveau de gris.

Après cela, le filtre bilatéral réduit le bruit dans l'image pour garder les frontières des objets dans l'image et une meilleure détection des contours.

Enfin, nous utilisons le filtre de canny pour spécifier bien tous les contours comme prétraitement pour l'étape d'identification de la plaque numérique dans la voiture. Lors de

l'application de ce filtre, nous obtenons une image dont la majorité des pixels sont noirs ce qui donne une matrice presque remplie de zéro comme le montre la figure suivante.



Fig .2.7 : Matrice d'une région d'une image en noir et blanc.

- **Recherche et localisation de la région de matricule :**

Ici, les contours sont des courbes qui nous aideront à identifier la plaque d'immatriculation de la voiture à partir de l'image.

Des opérations morphologiques d'érosion et de fermeture d'image sont effectuées pour exclure des petites parties qui n'appartiennent pas à la région de la plaque d'immatriculation. Ensuite, toutes les zones candidates restantes sont croisées avec une comparaison de la longueur et de la largeur de la plaque réelle, et donc la détermination de notre plaque tandis que les autres régions candidates sont toutes supprimées.

Après avoir découvert la zone de la plaque nous devons encadrer cette dernière à l'aide d'un positionnement précis basé sur la méthode de projection intégrale horizontale et verticale qui sont adoptées.

La projection intégrale horizontale peut être décrite par l'équation suivante

$$r(i, j) = |f(i, j) - f(i, j - 1)|, i = 1, 2, 3, \dots, m, j = 2, 3, 4, \dots, n \quad (\text{Eq.2.1})$$

où, la valeur en pixels de l'image, sont la hauteur et la largeur de l'image, respectivement.

Ensuite, la valeur de projection de la ligne nommée peut être obtenue en accumulant la valeur de pixel de l'image par ligne, qui peut être représentée comme suit :

$$T_1(i) = \sum_{j=2}^n r(i, j) \quad (\text{Eq.2.2})$$

La valeur moyenne est utilisée pour localiser efficacement la frontière. et peut être exprimée comme suit :

$$T_1(i) = \frac{(T_1(i-1) + T_1(i) + T_1(i+1))}{3} \quad (\text{Eq.2.3})$$

Après ce traitement, les statistiques de chaque ligne sont enregistrées dans un tableau, où est la valeur de la ligne correspondante. En analysant la valeur de projection horizontale, les régions de caractères correspondent généralement aux régions intermédiaires denses et hautes qui expliquent la position, et les bords supérieur et inférieur de la plaque d'immatriculation correspondent aux régions gauche et droite qui expliquent la position. De cette façon, la plaque d'immatriculation peut être positionnée avec précision dans le sens vertical.

- **Reconnaissance optique des caractères**

Nous avons maintenant identifié les bordures de la plaque, reste à l'extraction de son contenu et le convertir du format image au format texte d'une chaîne de caractères (lettre, chiffre ...) et cela se fait grâce à la technique tesseract. La zone de la plaque d'immatriculation contenant du texte peut désormais être décomposée en lignes, mots et caractères et le système tesseract est désormais capable de faire correspondre les caractères grâce à des algorithmes de comparaison et de détection basée sur un modèle d'apprentissage approfondi qui traite l'image ligne par ligne dans des boîtes. Ainsi, le texte contenu dans l'image est donné comme résultat final.

2.3.1.2. Algorithme entraîné par le deep learning (2ème modèle)

Bien que le premier modèle détecte une grande quantité de plaque d'immatriculation, cependant beaucoup de changements dans leur forme, qualité et les objets environnants entraînent de nombreux obstacles. Par conséquent, la solution optimale pour obtenir un système efficace avec une bonne précision est l'application d'un entraînement basé sur l'apprentissage profond à partir de la plate forme Tensorflow avec le Framework SSD.

a. Le réseau neuronal à convolution régionale (R-CNN)

Un R-CNN est un type spécial de CNN capable de localiser et de détecter des objets dans les images: la sortie est généralement un ensemble de boîtes englobantes qui correspondent étroitement à chacun des objets détectés.

b. Principe de détection pour le SSD

Le SSD est conçu pour la détection d'objets en temps réel qui sont des matricules dans notre système. Faster R-CNN utilise un réseau de propositions de région pour créer des boîtes de délimitation et utilise ces dernières pour classer les objets. Pour récupérer la baisse de précision, le SSD applique quelques améliorations, notamment des fonctionnalités multi-échelles et des boîtes par défaut. Ces améliorations permettent au SSD de correspondre à la

précision du R-CNN en utilisant des images de résolution inférieure, ce qui augmente encore plus la vitesse de traitement en temps réel et bat même la précision du Faster R-CNN.

c. Entraînement de notre modèle par SSD

Nous allons passer en revue toutes les étapes nécessaires pour créer notre détecteur d'objets, de la collecte des données jusqu'au test de détecteur :

- Rassemblement et étiquetage des données : Pour former un classificateur robuste, nous avons besoin de beaucoup d'images qui devraient différer beaucoup les unes des autres. Elles doivent donc avoir des arrière-plans différents, des objets aléatoires et des conditions d'éclairage variables, dans notre dataset il existe la diversité dans les matricules et les voitures avec plusieurs positions.

Afin d'étiqueter nos données, nous avons besoin d'une sorte de logiciel d'étiquetage d'images. On a utilisé LabelImg qu'est un excellent outil pour étiqueter des images. Le but de l'étiquetage est Pour créer le cadre de sélection et annoté l'image.

- Configuration de la formation : La dernière chose que nous devons faire avant la formation est de créer une carte d'étiquettes et un fichier de configuration de la formation. La carte d'étiquette mappe un identifiant à un nom. Nous le placerons dans un dossier appelé training, qui se trouve dans le répertoire matricule_detection. La carte vue ci-dessous

```
élément {
  id: 1
  nom: '036587 00 16'
}élément {
  id: 2
  nom: 'WW-049-RS'
}élément {
  id: 3
  nom: 'DM-730-LS'
}
```

Fig.2.8 : Liste de configuration des données.

On doit entraîner le modèle jusqu'à ce qu'il atteigne une perte satisfaisante.

1. Exportation du graphique d'inférence : Maintenant que nous avons un modèle entraîné, nous devons générer un graphique d'inférence, qui peut être utilisé pour exécuter le modèle. Pour ce faire, nous devons d'abord trouver le numéro d'étape enregistré le plus élevé.

2. Test du détecteur d'objets : la dernière étape est de tester notre modèle nouvellement créé. Maintenant, nous pouvons exécuter toutes les cellules et nous verrons une nouvelle fenêtre avec une ouverture de flux de caméra et l'apparence de détection des matricules en temps réel.
3. L'extraction du numéro d'immatriculation : après le test de notre modèle on a appliqué la reconnaissance optique des caractères pour imprimer le matricule en texte.

2.3.2 Exécution et discussion des résultats de chaque partie du programme

Dans cette section, on va discuter le résultat de l'implémentation des deux modèles. Après l'importation des bibliothèques requises on va prévisualiser les résultats.

2.3.2.1 Implémentation de 1ere modèle

L'image est lue en niveau RGB à partir des fonctions `cv2.imread` de la bibliothèque `opencv` et redimensionné avec `imutils`.

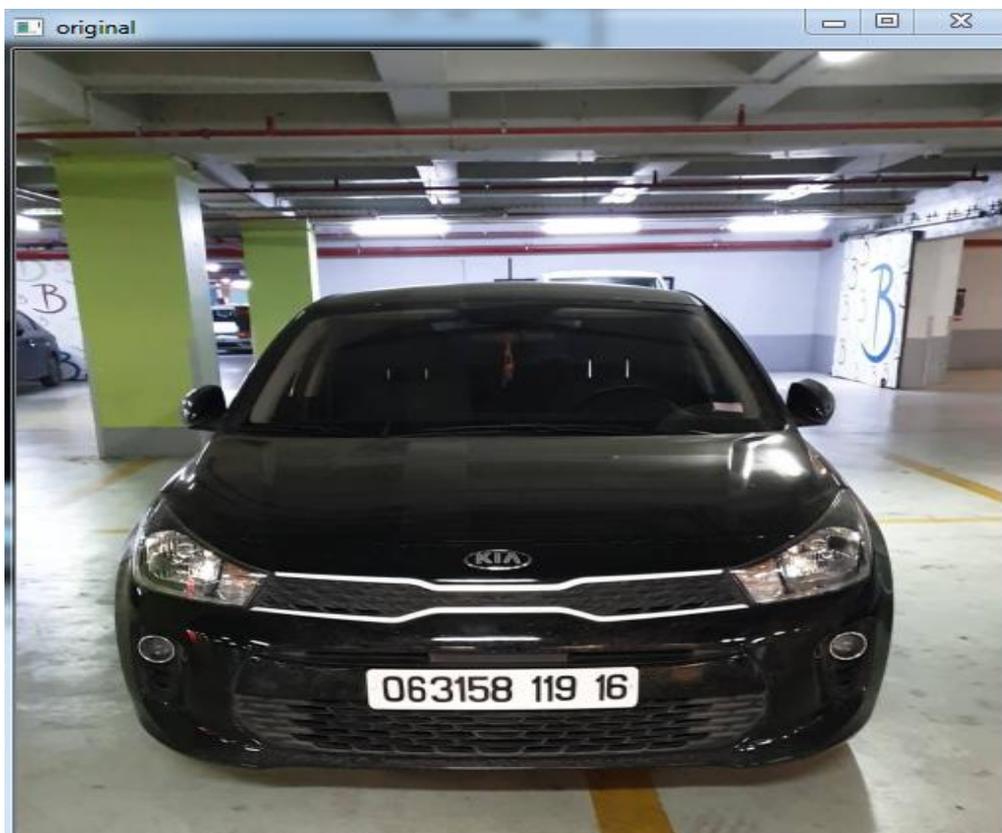


Fig.2.9 : Image originale en couleurs.

La conversion en niveau de donne l'image suivante :

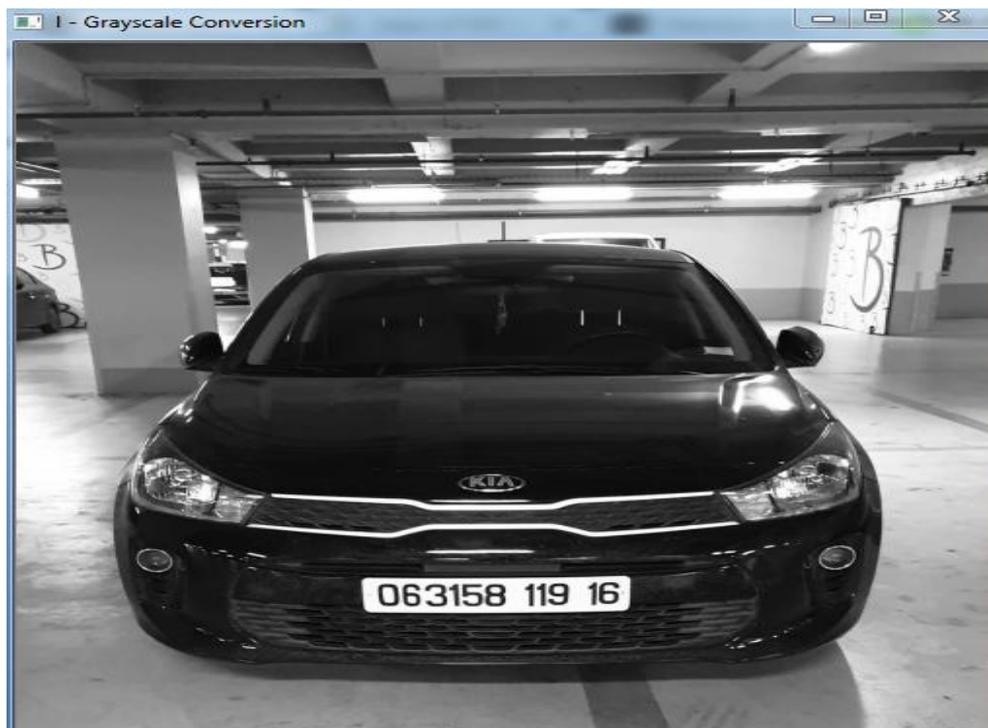


Fig.2.10 : Image en niveau de gris.

En utilisant les fonctions de cv2. Bilateral Filter et cv2.Canny qui prennent le premier argument comme image d'entrée, les deuxième et troisième arguments sont respectivement nos minVal et maxVal qui spécifient les valeurs de seuil, nous obtenons la sortie suivante :

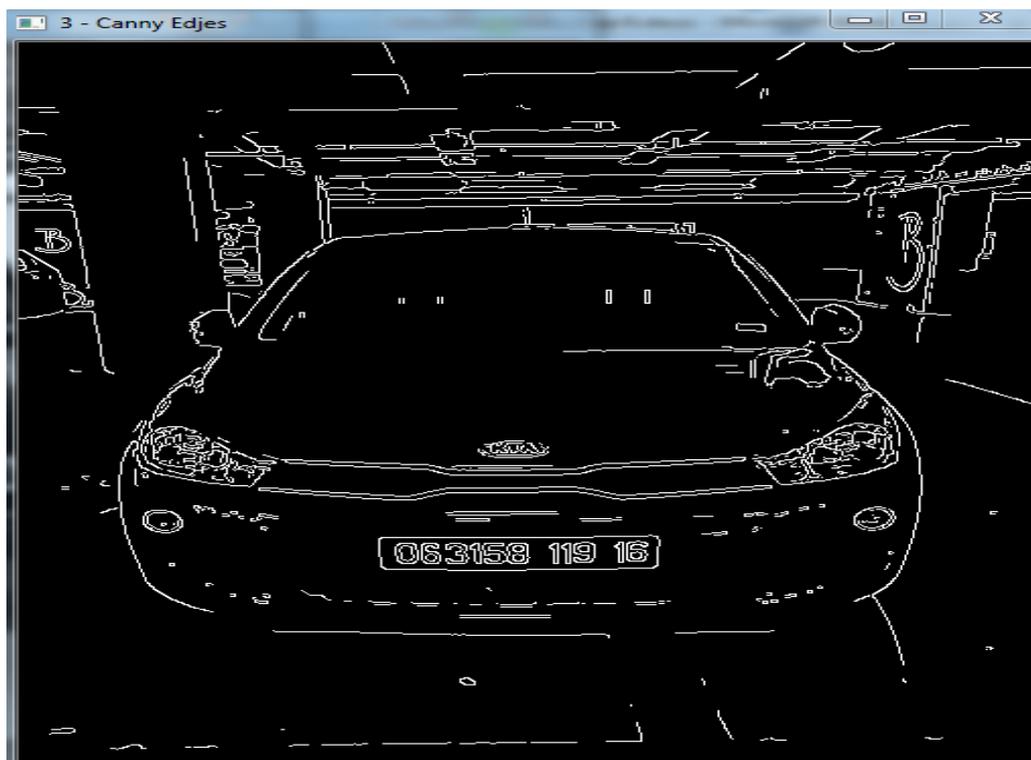


Fig.2.11 : Image après application des filtres bilatéral et canny.

Pour la localisation de matricule :

1. Le premier argument est l'image source.
2. Le deuxième argument est le mode de récupération de tous les contours détectés dans l'image.
3. Le troisième paramètre est la méthode d'approximation des contours. Stocké les extrémités des contours détectés.

La fonction `cv2.drawContours ()` prend cinq arguments-

1. Le premier argument est l'image dans laquelle les contours détectés seront dessinés.
2. Le deuxième argument est la variable qui stocke tous les contours détectés.
3. Ensuite, le troisième argument est les index de contour. Ici, nous utilisons la valeur -1 qui utilisera les index de tous les contours détectés dans notre image. Par conséquent, tous les contours seront dessinés sur l'image.
4. Le quatrième argument désigne la couleur dans laquelle les contours seront dessinés.
5. Le cinquième argument est l'épaisseur de la courbe de contour à tracer.

Cela donne la sortie suivante :

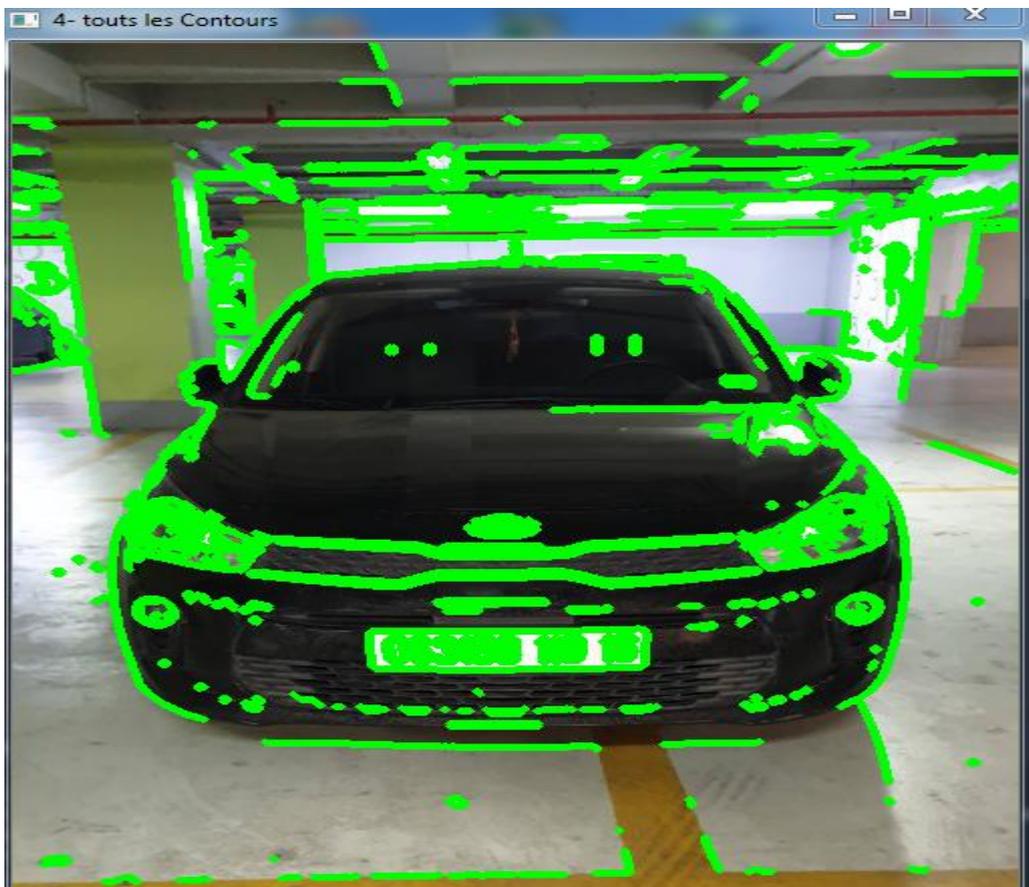


Fig.2.12 : Image avec les contours détectés.

Pour terminer cette étape, nous sélectionnons les 30 premiers contours en fonction de la zone par ordre décroissant car nous n'avons pas besoin des différents petits contours. Cela réduit les contours redondants et petits qui ne sont pas nécessaires

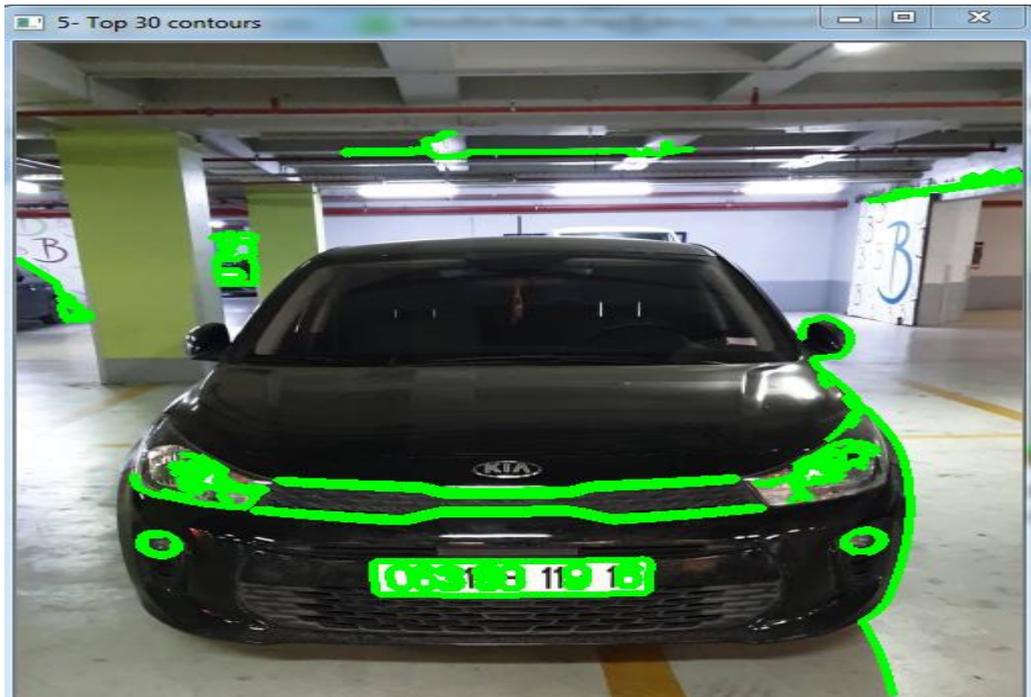


Fig.2.13 : Image avec les contours proches de matricule.

On utilise une boucle pour trouver le meilleur contour approximatif possible de la plaque d'immatriculation. Par la fonction `perim`, nous calculons le périmètre de chaque contour. Après avoir trouvé les coordonnées de la plaque à l'aide de la fonction `boundingRect` d'OpenCV, nous stockons l'image avec de nouvelles dimensions dans la variable `new_img` par la fonction `imwrite` d'OpenCV. Le contour sélectionné sur l'image originale qui sera sur la plaque d'immatriculation est dessiné comme suit :

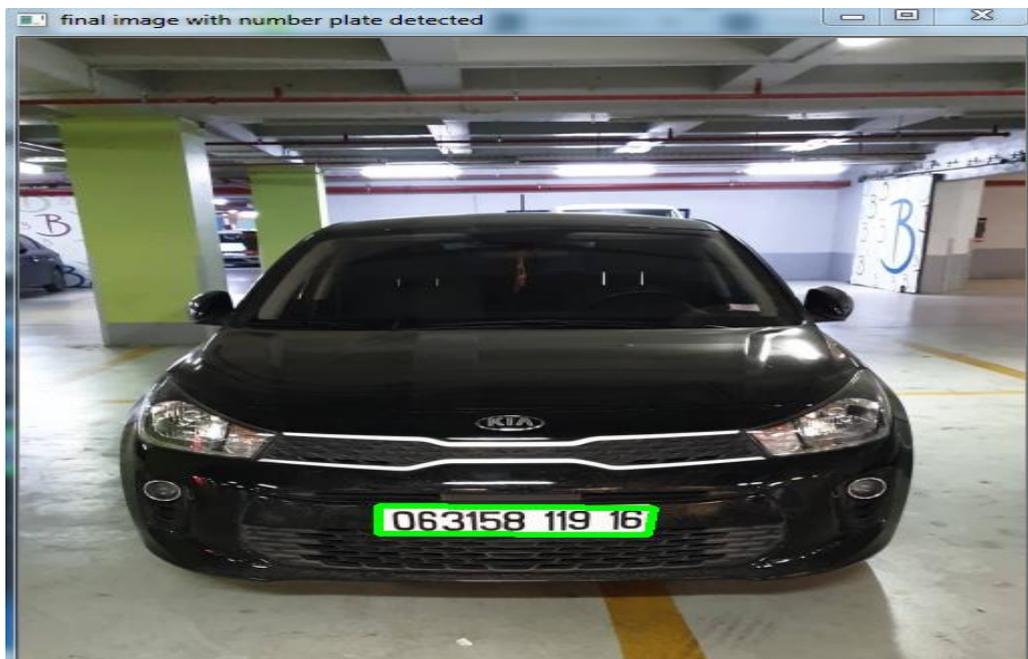


Fig. 2.14 : Image avec matricule la localisation de matricule.

En utilisant pytesseract, nous extrayons les caractères de l'image et imprimons le texte à l'écran comme vous pouvez le voir ci-dessous:

```
le numéro d'immatriculation est : 063158 119 16
```

Fig 2.15 : Extraction de numéro d'immatriculation.

2.3.2.2. Implémentation du deuxième modèle (deep learning)

L'une des choses les plus importantes dont nous bénéficions dans ce modèle est la possibilité de spécifier le matricule en temps réel, mais pour présenter ses avantages de manière idéale, il suffit de clarifier son exécution en considérant l'entrée de modèle comme une image prise par une caméra ip qui diffuse en direct.

Lors de l'exécution de ce modèle il est nécessaire d'apporter la bibliothèque de 'apprentissage au profond Tensorflow et aussi la bibliothèque time pour générer le fps, puis intégrer ou importer ce modèle dans notre code source général. Le résultat est la détection de la plaque d'immatriculation malgré le changement de voiture ,ainsi que la modification de la forme du contenu des plaques. L'exécution de ce modèle aux quatres véhicules nous a donné les résultats suivants :



Fig.2.16 : Ensemble des véhicules avec matricules détectés.

La puissance d'identification du panneau d'immatriculation change, et cela est dû à la précision du classificateur représenté par la boîte englobante.

Pour le camion Hyundai, nous notons qu'il existe deux types de classificateurs, le rectangle au dessus est la boîte englobante de la vérité et un autre rectangle au dessous la boîte englobante prévue. On constate qu'il n'y a pas d'identification entre eux, donc nous pouvons conclure le constant IOU qui est inclus au intervalle [0.5 1].

Quant aux trois autres voitures, on remarque que la planche est déterminée plus précisément, et chacun des deux packages est presque identiques. Donc le constant IOU=1.

2.4. Conclusion

Dans ce chapitre, nous avons discutés le système de détermination des plaques d'immatriculations. Tout d'abord on a présenté les différents outils pour développer la fonctionnalité de ce système. Ensuite, on a profité de ces méthodes pour produire deux modèles. Enfin, nous avons conclu l'importance de l'apprentissage en profondeur pour accroître l'efficacité et la force de l'identification de nombreux matricules.

Chapitre 3

Étude conceptuelle et
réalisation de
l'application

3.1. Introduction

Après avoir expliqué le fonctionnement des algorithmes de détection de la plaque d'immatriculation, nous illustrerons dans ce chapitre comment utiliser le numéro d'immatriculation pour faire fonctionner le parking. D'abord, nous exposerons le schéma de la base de données. Ensuite, nous allons connecter l'interface graphique avec notre programme .Enfin, nous présentons l'application bureautique comme produit final du système..

3.2. Cahier de charge

Pour les entreprises et sociétés qui contiennent un grand nombre de travailleurs, il est nécessaire de fournir un parking pour les véhicules. La diversité des postes et aussi l'augmentation du nombre d'employés exige de mettre en place un système intelligent de gestion et de maintien des opérations au niveau de parking. De cette façon, nous maintenons la sécurité interne et la confidentialité de l'entreprise.

Par l'étude de plusieurs échantillons de sociétés, nous avons déduit des multiples obstacles et problèmes, auxquels nous allons tenter d'apporter des solutions tels que :

1. Problème de la non-numérisation des données : les informations sur l'heure d'entrée et de sortie ne sont pas enregistrées dans une base de données, cela signifie qu'il y a un risque de perte de données.
2. Le flux de mouvement important de divers véhicules, y compris des voitures, des camions, des autobus à l'intérieur du parking: nécessité d'un suivi précis selon le rythme des changements de forme et de la nature de la météo sinon il entraîne des erreurs et des lacunes lors de la détection. Donc le système de reconnaissance du véhicule est limité et lent, non développé pour analyser et traiter l'image de la voiture.
3. Un grand gâchis dans l'organisation pour le stationnement des véhicules: ce qui provoque des encombrements beaucoup plus au niveau de la sortie.
4. Dans le cas d'existence de système de gestion de hangar, il n'y a pas d'interface graphique appropriée pour le logiciel afin de faciliter l'environnement de travail interactif.

5. Le système de vérification et de reconnaissance des coordonnées du véhicule (matricule, reconnaissance facial, carte grise) ne contient pas la puissance et l'efficacité.

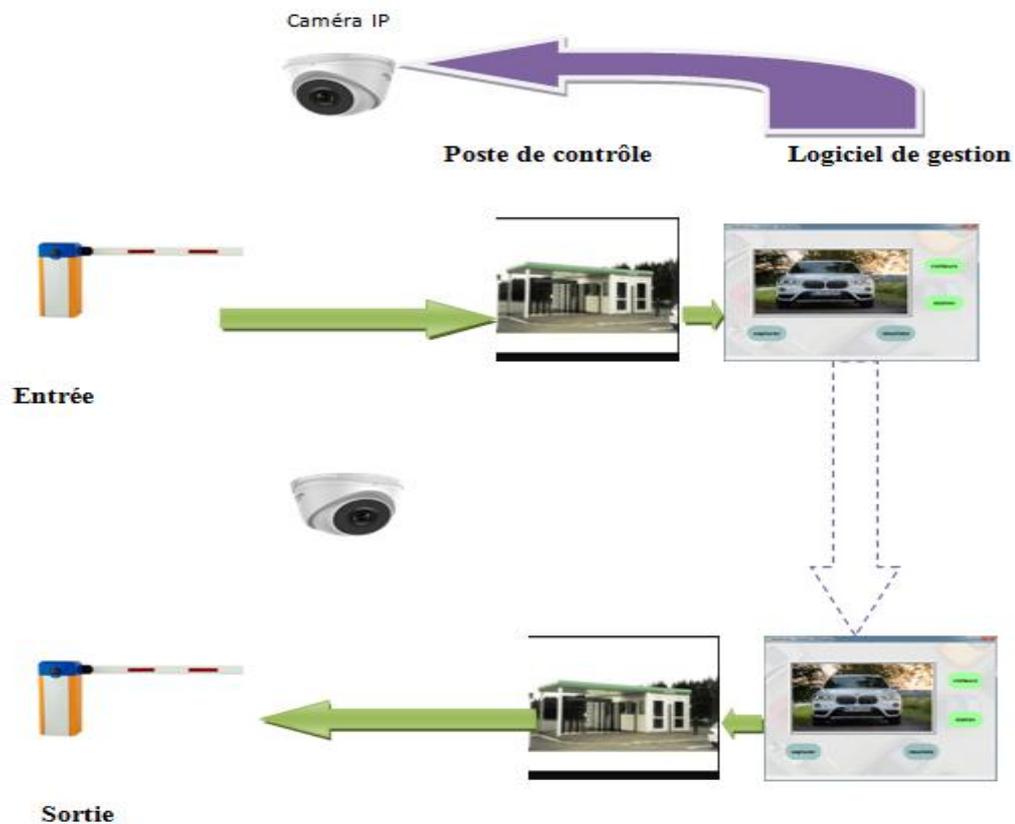


Fig. 3.1 : Représentation de fonctionnement de système de gestion de parking.

L'objectif de la création de ce système est de résoudre les problèmes dont souffrent de nombreux parkings. Les caractéristiques les plus importantes que doit posséder ce dernier sont les suivantes :

- Grande précision dans l'exploration de la plaque d'immatriculation et donc la rapidité pour l'identification du propriétaire de la voiture.
- La répartition régulière des places de stationnements et conservation automatique des heures d'entrée et de sortie.
- Une interface graphique interactive simple et facile à manipuler

3.3. Les diagrammes UML

Les utilisateurs (agents), ne sont pas nécessairement dans le domaine de télécommunication ou de l'informatique. Il leur faut donc un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation, diagramme d'activité et diagramme de base de donnée qui permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système.

3.3.1. Diagramme de cas d'utilisation

Ce diagramme montre les tâches accomplies par chaque élément utilisé pour cette application. Il spécifie les actions qui peuvent être prises pour chacun d'eux.

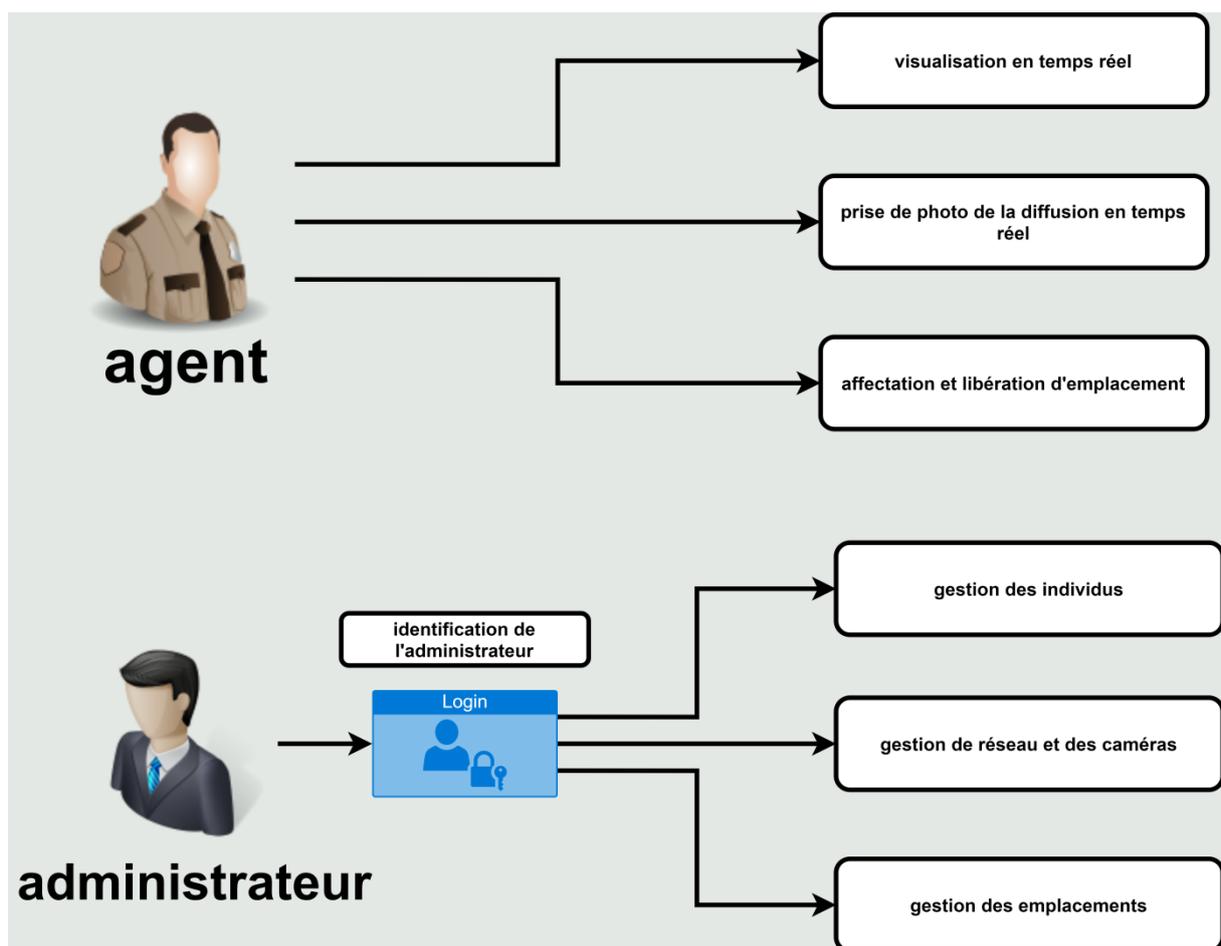


Fig.3.2 : Diagramme des cas d'utilisations.

3.3.2. Diagramme d'activité

Le diagramme d'activité est une présentation d'organigramme permet de modéliser le déclenchement d'événements, actions et décision en fonction des états du système.

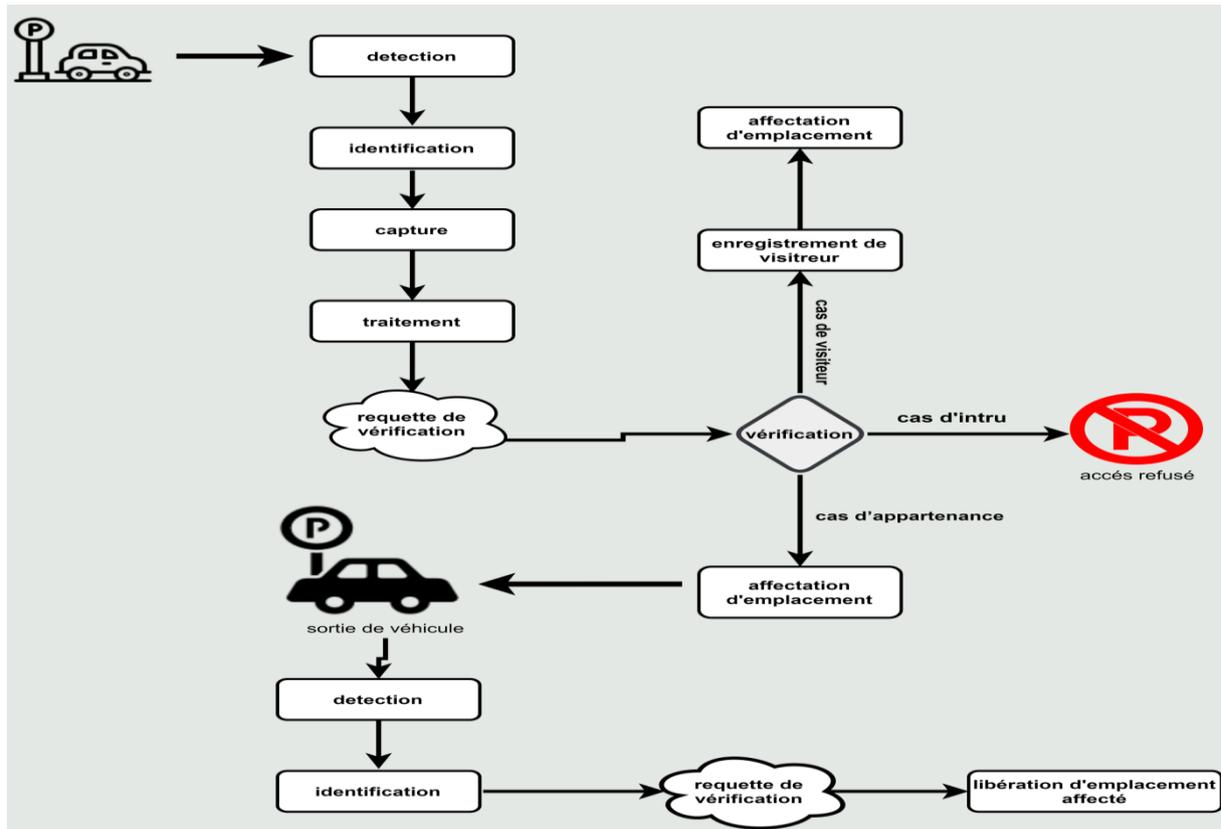


Fig.3.3 : Diagramme d'activités.

3.3.3. Diagramme de classes (base de données)

Il s'agit d'un schéma dans lequel nous pouvons en apprendre les différents composants de la base de données et les relations entre tous ses composants.

Pour comprendre le fonctionnement des tableaux relationnels, nous devons d'abord revoir les éléments de chaque tableau. Pour notre application quatre tableaux sont créés :

Table	Action
<input type="checkbox"/> emplacement	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> historique	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> propriétaire	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> voiture	★ Parcourir Structure Rechercher Insérer Vider Supprimer
4 tables	Somme

Fig.3.4 : Représentation les éléments de base de données.

- Emplacement : ce tableau sert à indiquer l'endroit où la voiture va stationner.
- Historique : ce tableau sert à sauvegarder le temps d'entrée et de sortie de chaque voiture.

- Propriétaire : ce tableau contient les informations des individus qui possèdent les voitures.
- Voiture : ce tableau contient les informations concernant les voitures inscrites dans la base de données.

Les éléments du tableau sont des constantes SQL ordinaires ou des expressions, on va montrer les éléments de chaque tableau et expliquer ceux qui ont plus d'importance comme suit

Emplacement : ce tableau contient trois éléments numérotés 1 à 3

	#	Nom	Type	Interclassement
<input type="checkbox"/>	1	id_garage 	int(11)	
<input type="checkbox"/>	2	nom	varchar(2)	latin1_swedish_ci
<input type="checkbox"/>	3	situation	tinyint(1)	

Fig.3.5 : Structure de tableau 'emplacement'.

- Nom : indique l'emplacement exact de véhicule.
- Situation : indique si la place est occupé dont (1=occupé, 0 = vide)

Historique : ce tableau contient cinq éléments :

	#	Nom	Type
<input type="checkbox"/>	1	id_sta 	int(11)
<input type="checkbox"/>	2	id_voiture 	int(11)
<input type="checkbox"/>	3	id_garage 	int(11)
<input type="checkbox"/>	4	dedans	timestamp
<input type="checkbox"/>	5	dehors	timestamp

Fig.3.6 : Structure de tableau 'historique'.

- Dedans : sauvegarde automatiquement le temps d'entrée au parking des utilisateurs
- Dehors : sauvegarde automatiquement le temps de la sortie du parking des utilisateurs

Propriétaire : ce tableau contient six éléments

	#	Nom	Type	Interclassement
<input type="checkbox"/>	1	id_prop 	int(11)	
<input type="checkbox"/>	2	nom	varchar(50)	latin1_swedish_ci
<input type="checkbox"/>	3	prenom	varchar(50)	latin1_swedish_ci
<input type="checkbox"/>	4	date naissance	date	
<input type="checkbox"/>	5	niveau	varchar(50)	latin1_swedish_ci
<input type="checkbox"/>	6	id_voiture1 	int(11)	

Fig.3.7: Structure de tableau 'propriétaire'.

- nom : contient les noms des propriétaires
- prénom : contient les prénoms des propriétaires
- date de naissance : contient les date de naissance des propriétaires
- niveau : indique le niveau administratif des utilisateurs

Voiture : ce tableau contient cinq éléments

#	Nom	Type	Interclassement
1	id_voiture	int(11)	
2	modele	varchar(50)	latin1_swedish_ci
3	marque	varchar(50)	latin1_swedish_ci
4	matricule	varchar(15)	latin1_swedish_ci
5	id_prop	int(11)	

Fig.3.8 : Structure de tableau 'voiture'.

Modèle : contient les noms des voitures reconnues dans le parking

- Marque : contient les marque des voitures des reconnues dans le parking
- Matricule : contient les numéros d'immatriculation des voitures reconnues

La figure suivante représente le diagramme de notre base de données :

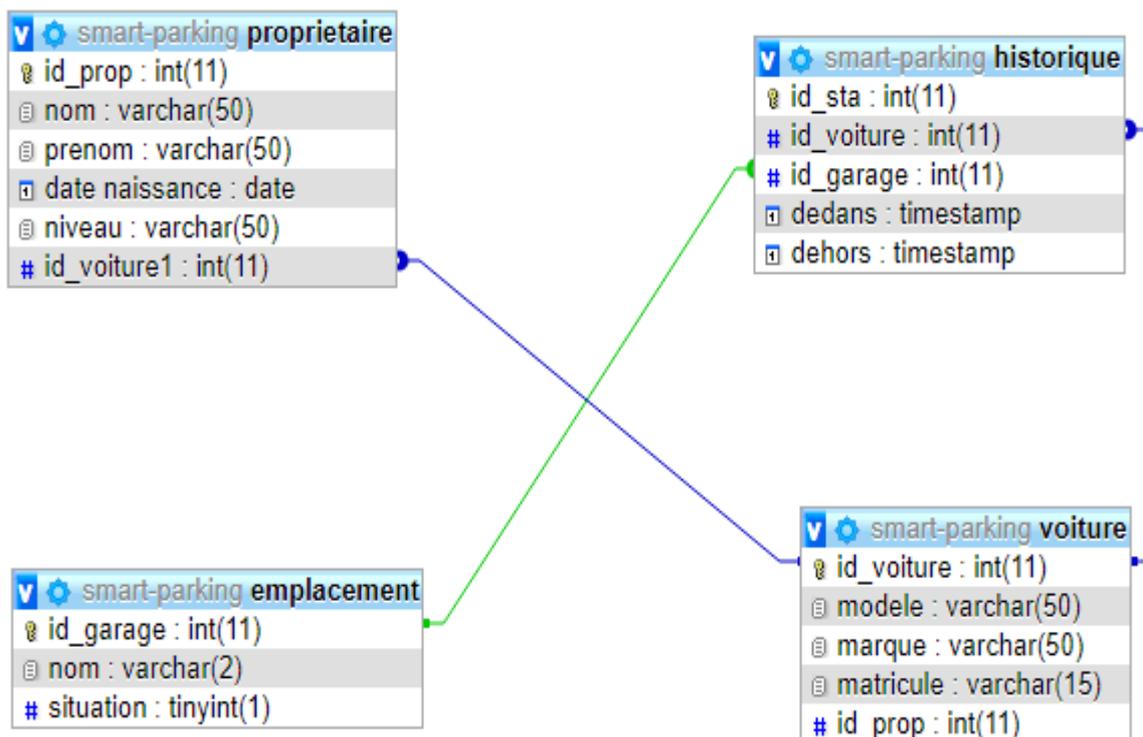


Fig.3.9 : Structure générale du diagramme des bases de données.

Le lien entre ces tableaux est l' (id) ; Le rôle de la relation entre le tableau de propriétaire et celui de voiture est lorsque le véhicule rentre dans le parking, et après avoir fait correspondre le numéro d'immatriculation, l'identité et les informations de la personne sont extraites.

Le rôle de la relation entre le tableau de voiture et d'historique est d'enregistrer les heures d'entrée et de sortie des propriétaires.

La dernière relation est entre les tableaux historiques et emplacement, pour réserver et enregistrer la place de stationnement au niveau du parking.

3.4. Aperçu sur l'interface de l'application

Afin de faciliter l'exécution de notre idée on a créé un programme exécutable qui facilite le travail à savoir commander et gérer ce programme et ce par n'importe quel individu même s'il possède une connaissance minimale de la manipulation d'ordinateur ; L'application bureautique de système de gestion pour les parkings est représentée par deux interfaces

3.4.1. Interface de temps réel

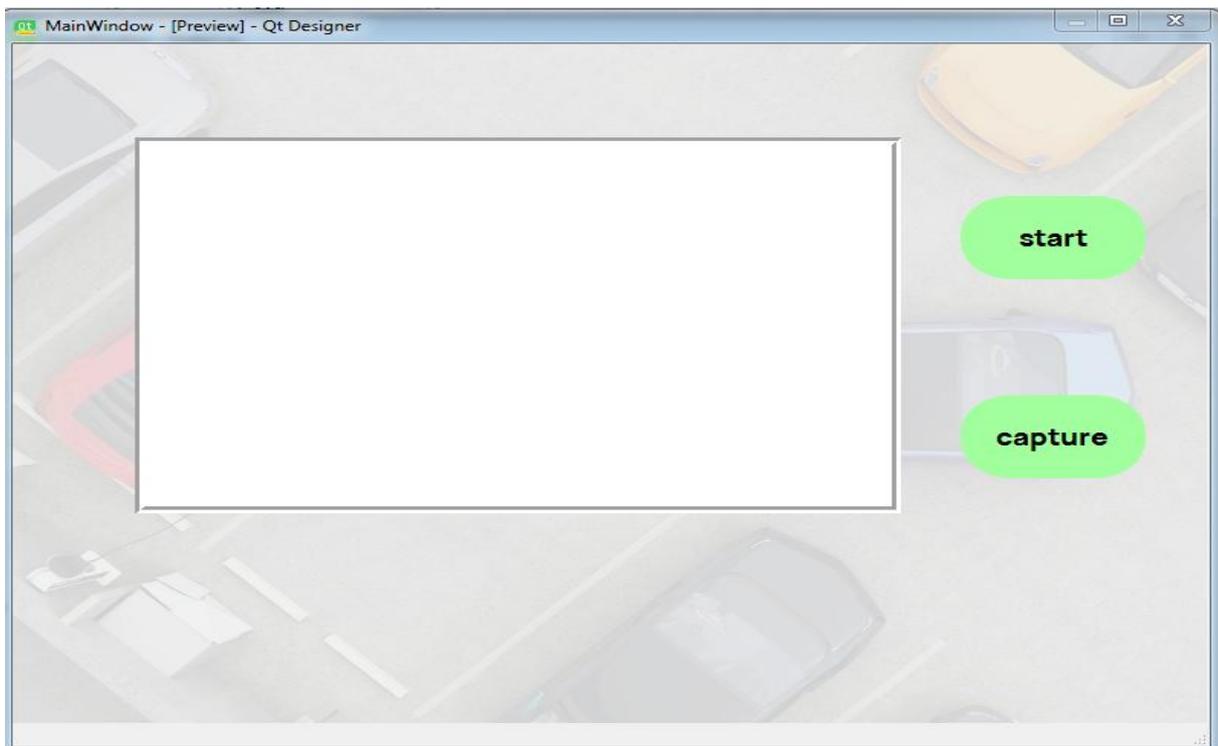


Fig.3.10 : Interface principale.

C'est l'interface qui est responsable pour le diffusion de caméra à partir le bouton Start et de prendre une photo avec le bouton capturer du véhicule entrante. Au même temps la procédure de localisation de matricule travaille.

3.4.2. Interface de traitement avec la base de données

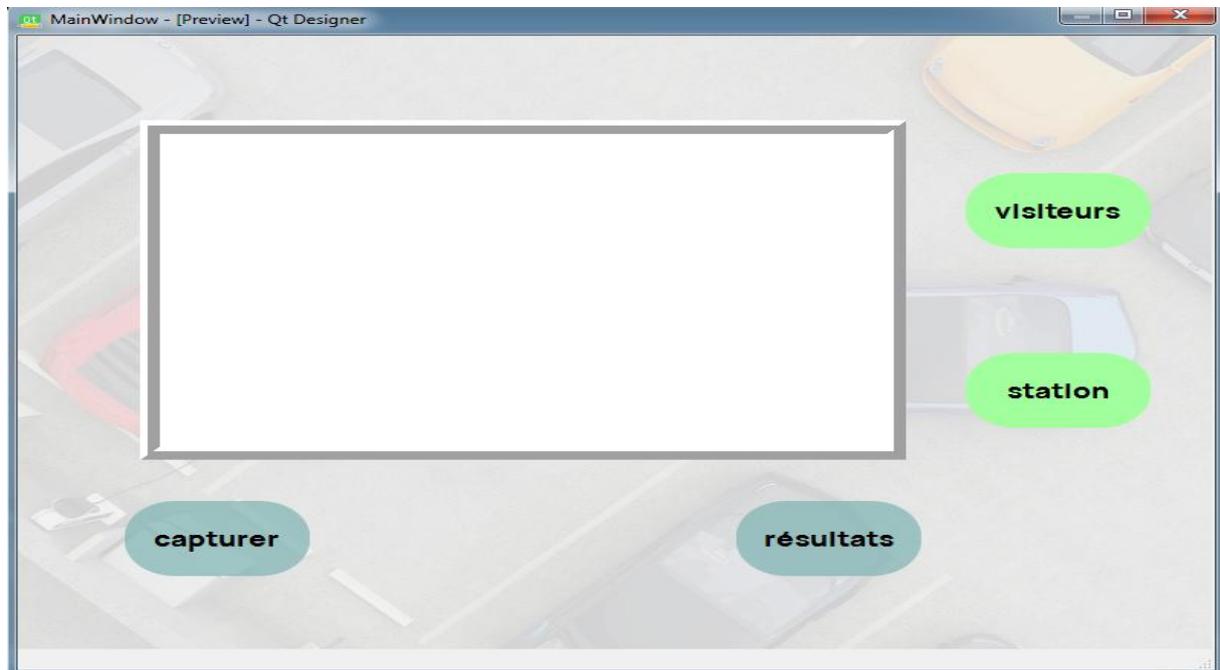


Fig .3.11 : Interface de gestion.

Maintenant on a la photo de véhicule avec le matricule détecté, chaque bouton de cette interface réalise une tâche pour visualiser les résultats

Le bouton capturer affiche la photo de voiture, résultats récupère et affiche les données personnels de propriétaire entrant, station montre où la personne est garée et bouton visiteurs pour ajouter des autres personnes intrus aux bases de données.

3.5. Description de fonctionnement de logiciel

Dans cette partie, nous examinerons la façon dont l'application de système d'identification des plaques d'immatriculations travail , où nous pouvons recevoir deux cas , le premier cas lié à l'entrée d'une personne appartenant à la liste d'inscrits au niveau du base de données et un autre personne qui n'appartient pas à notre base.

Pour connecter notre programme à la base de données, nous devons appliquer ce qui suit :

La première étape pour pouvoir accéder , créer et gérer la base des données on a besoin tout d'abord d'activer le server (wamp)

Une fois server le lancé, on accédé la page d'administration on introduisant le lien : (<http://localhost/phpmyadmin/>) sur le navigateur internet (on n'a pas besoin du service internet, un réseau (wlan) bien établie est suffisant)

On obtient la page :

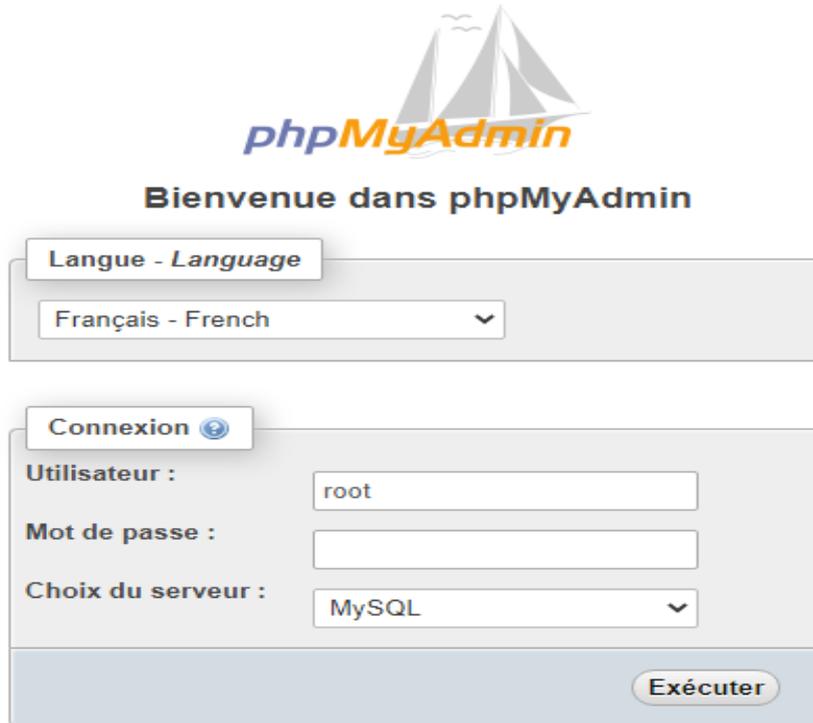


Fig.3.12 : Page d'accueil d'application web phpMyadmin.

Par défaut l'utilisateur est (root) et aucun mot de passe n'est configurée , on clique sur le Bouton (exécuter) .

Après cela on va créer la bas e de données suivants les étapes suivants :

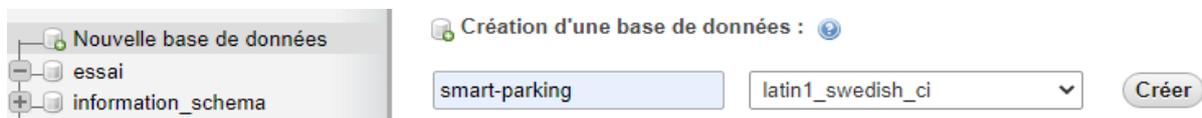


Fig.3.13 : Insertion du nom de base donnée.

On clique sur (nouvelle base de données) et on donne le nom.

3.5.1. Dans le cas d'appartenance au parking

Pour passer en revue les différentes étapes que l'interface graphique traverse dans ce cas, nous appliquons la personne suivante qui appartient à la base de données.

id_prop	nom_prop	prenom_prop
3	selmi	anwar

Fig.3.14 : Nom et prénoms d'un propriétaire.

Avec les données de voiture suivante :

2	berlingo	peugeot	26146 113 42	3
---	----------	---------	--------------	---

Fig.3.15 : Véhicule enregistrée.

Dans ce cas nous avons trois emplacements pour garer la voiture et le plus proche est A8

A6	1
A7	1
A8	0
A9	0

Fig.3.16: Les emplacements disponibles dans le parking.

Où il possède la voiture ci-dessous :



Fig.3.17 : Voiture entrante.

Dans la première phase L'image de la voiture est reçue via l'interface de temps réel :



Fig .3.18 : Visualisation de la voiture en temps réel.

Grâce à l'interface ci-dessous qui représente la deuxième phase, le numéro d'immatriculation extrait est comparé aux numéros de la base de données et cette interface mis à notre disposition plusieurs boutons pour accéder à plusieurs interfaces comme suit:

Bouton capturer : le rôle de ce bouton est de recevoir et d'afficher l'image de la voiture depuis l'interface en temps réel où le matricule y est déterminé. Le résultat est le suivant :



Fig .3.19 : Affichage de la voiture capturée avec le matricule détecté.

Bouton résultat : c'est le bouton le plus important où l'on peut voir les informations de la personne concernée pour l'entrée pour cela c'est une fenêtre d'application pour l'administrateur de ce système comme indiqué ci-dessous :

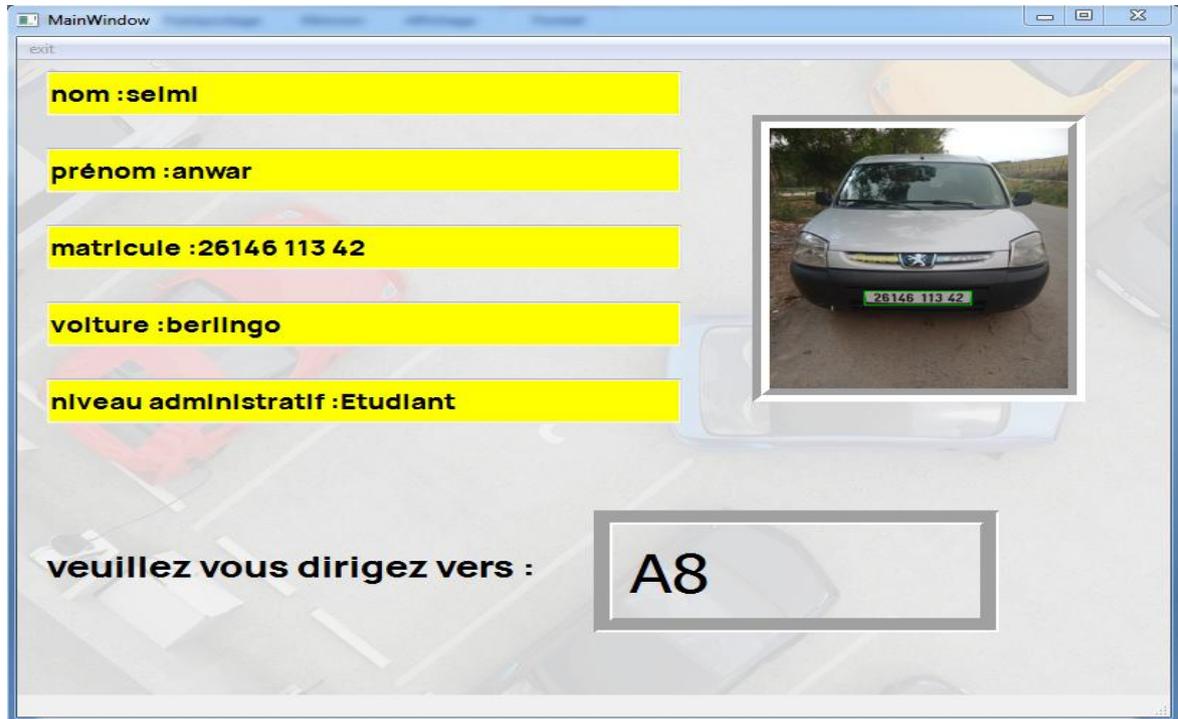


Fig.3.20 : Affichage des informations du propriétaire enregistré.

Bouton station : cliquer sur ce bouton nous donne cette fenêtre qui montre où la voiture est garée et donc elle est liée au conducteur qui veut entrer dans le parking.

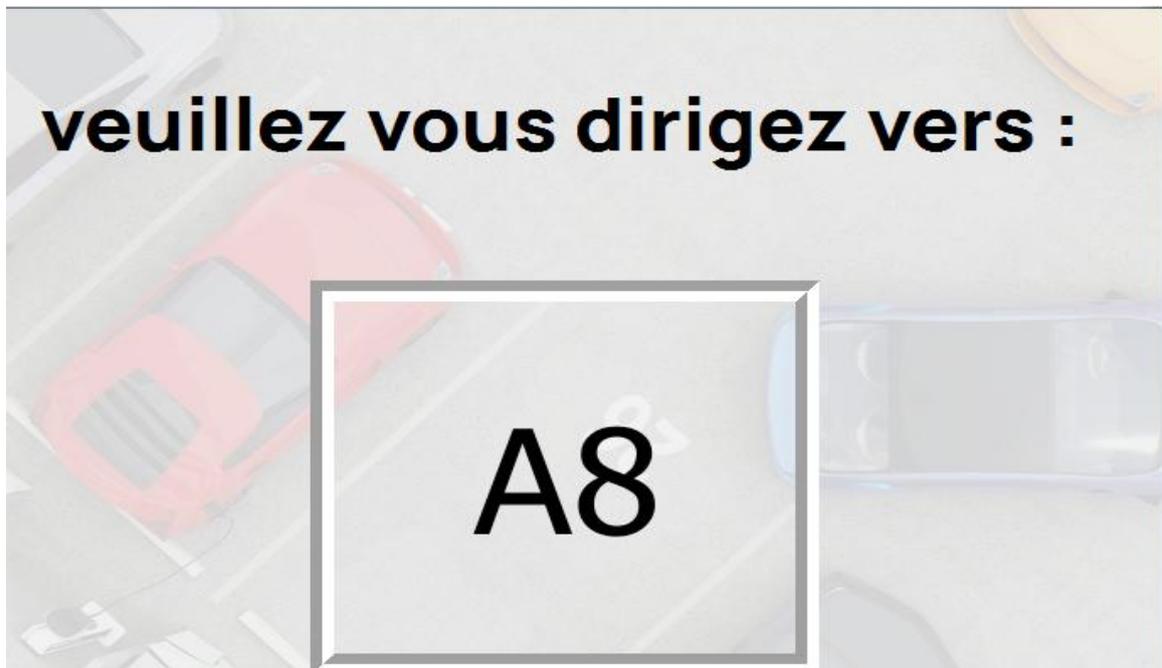


Fig.3.21 : Interface de l'emplacement pour propriétaire entrant.

Ceci indique que A8 est réservé et on peut l'observer aussi sur la base de données :

11	A6	1
12	A7	1
13	A8	1
14	A9	0

Fig.3.22 : Réservation de l'emplacement A8.

Dans la dernière phase nous avons une interface de sortie à travers laquelle le système d'identification des plaques d'immatriculations est réutilisé pour libérer la place de stationnement de la voiture qui a quitté. Parmi les informations qui apparaissent dans cette fenêtre est le temps d'entrée et de sortie de cette voiture comme indiqué ci-dessous :



Fig .3.23 : Interface de sortie.

3.5.2. Résultat d'identification pour le cas d'intrus

Si un véhicule essaye d'accéder au sein de notre établissement et que ce dernier n'est pas reconnu après la détection (premier accès du véhicule), si après traitement par le logiciel il n'y a pas d'identification entre le matricule et les numéros d'immatriculations dans la base de données, le résultat est donné sur la fenêtre suivante:

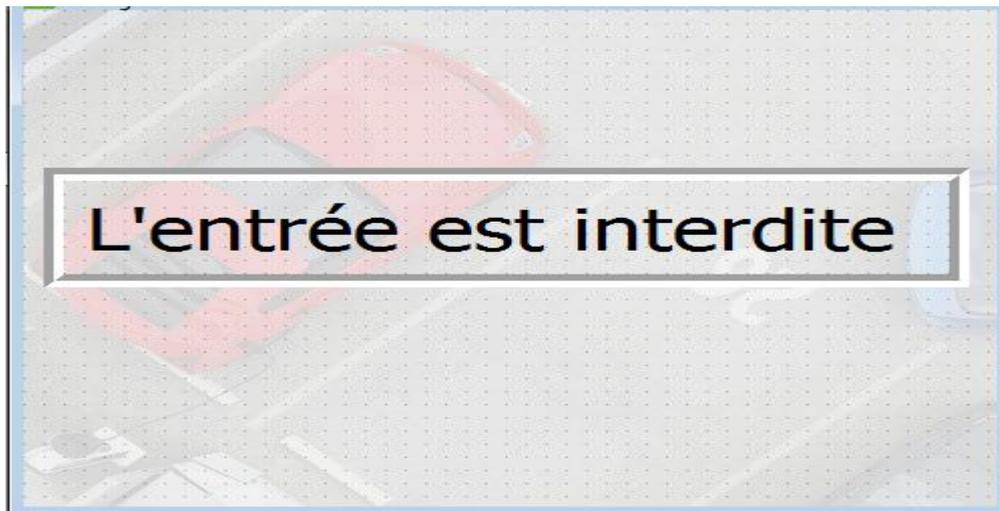


Fig.3.24 : Résultat d'identification pour le cas d'un intrus.

Pour les intrus qui sont des visiteurs qui ne sont pas enregistrés dans la base de données, ils peuvent être ajoutés en accédant à cette interface via le bouton visiteurs



The image shows a web form titled "enregistrement des étrangers". It contains five input fields, each with a label to its left: "nom :", "prénom :", "matricule de voiture :", "nom de voiture :", and "date et heure d'entrée :". Below these fields is a blue button labeled "charger". The form is set against a background of a car and some documents.

Fig.3.25 : Fenêtre pour l'ajout des visiteurs.

3.6. Conclusion

Dans ce chapitre, nous avons abordé la phase de conception et réalisation de notre logiciel en exposant les besoins dans le cahier de charge en premier lieu. Ensuite, nous avons entamé la conception par la représentation des différents diagrammes qui expliquent les méthodes pour que notre système déroule. Enfin pour montrer la simplicité de l'interface, nous avons mené une expérience d'utilisation de l'application.

Conclusion générale

CONCLUSION GÉNÉRALE

En raison de la croissance continue de l'utilisation des parkings, l'élaboration d'un système efficace permettant de gérer les parkings au niveau des entreprises est devenue un sujet intéressant.

Dans ce mémoire nous avons abordé la problématique de gestion des parkings. Ainsi, nous avons développé une application qui sera basée sur l'idée de gestion du parking à partir de la reconnaissance des matricules des voitures qui y ont accès.

Les principaux objectifs de notre projet étaient:

- La bonne exploitation et usage des lois de traitement d'image pour concevoir le meilleur modèle qui détecte la plaque d'immatriculation.
- L'obtention de la cohérence de stationnement tout en maintenant la plus grande sécurité dans les parkings sensibles.
- La création d'une interface graphique simple à utiliser qui représente ce système.

Au cours de ce travail, nous avons étudié, évalué et appliqué de nombreuses techniques avancées en traitement d'image tels que, les réseaux de télécommunications, la vision par ordinateur, la programmation en python, l'intelligence artificielle.

Dans le premier chapitre, et après avoir présenté un état de l'art sur les systèmes existants nous avons aussi cité les concepts de plusieurs technologies utilisées en expliquant le rôle de chacun d'elles ainsi que le bienfait qu'on peut en tirer. Après cela, dans le deuxième chapitre on a appliqué les technologies mentionnée dans le premier chapitre afin d'atteindre le meilleur algorithme pour le programme de localisation des plaques d'immatriculation et l'extraction de numéro. A la fin, nous avons mis en œuvre l'idée de la relation qui existe entre les plaques d'immatriculations des véhicules et l'identité de leurs propriétaires grâce à la base de données enregistrées et l'interface graphique et là nous avons pu afficher les informations dont nous avons besoin.

Grâce aux résultats que nous avons obtenus après avoir l'essai de ce système via le logiciel, il a montré son efficacité, précision et sa capacité à fonctionner en temps réel pour plusieurs opérations de haute qualité.

Ce projet nous a été très bénéfique car il nous a permis de maîtriser des outils très complexes comme c'est une bonne occasion pour réaliser un travail concret, avec des objectifs clairs et bien définis.

Conclusion générale

Des difficultés et obstacles ont été rencontrés durant les différentes étapes de notre projet, qui ont un petit peu freiné notre accès à une meilleure version et application du système, parmi ces problèmes citons ce qui suit :

- L'existence de plusieurs idées, leur émergence pendant des périodes intermittentes pour mettre en œuvre le système, ce qui a conduit à une mauvaise estimation pour attribuer certaines tâches.
- L'entraînement au modèle d'apprentissage approfondi nous a pris beaucoup de temps pour le mettre en œuvre et obtenir le résultat souhaité, et cela en raison de la quantité de données et de la faiblesse de données et de la faiblesse relative à notre ordinateur.
- En raison des conditions de pandémie de Covid 19, nous n'avons pas pu mener d'expérimentation pour ce projet au niveau d'université et nous n'avons donc pas eu l'occasion mieux évaluer le projet.

Ce projet acquiert de vastes perspectives d'avenir car son développement dépend du progrès de ses technologies, Nous les détaillons comme suit :

- L'Utilisation d'une méthode d'apprentissage approfondie plus efficace et arriver ainsi à un modèle plus précis pour la détection de matricule (YOLO Framework).
- L'ajout de système embarqué pour automatiser le contrôle de barrière.
- L'automatisation complète du système en éliminant le gestionnaire ou l'administrateur de l'application et l'adoption d'une carte (RFID) pour l'identification du propriétaire.
- Pour une meilleure sécurité en ajoutant la reconnaissance faciale en formant un modèle apprentissage automatique (OpenFACE framework).
- Développement de l'interface graphique vers une plateforme sur internet ou vers une application téléphonique.

Bibliographie



Bibliographie

- [1] Site web: <https://www.hubparking.com/fr/fonctionnalites/lecture-de-plaques-dimmatriculation-lpr> . Décembre 2019.
- [2] Site web: <https://www.vs-securite.fr/reconnaissance-de-plaque/> .Décembre 2019.
- [3] Site web : <http://www.alphanumeric-vision.com/fr/reconnaissance-plaques-immatriculation/> .Décembre 2019.
- [4] Site web: <https://www.vitronic.fr/ingenierie-du-traffic-routier/applications/peages-et-identification-des-vehicules/reconnaissance-de-plaques-dimmatriculation.html> .Décembre 2019.
- [5] Site web: https://www.imotionsecurite.com/pdf/genetec/FR-Genetec_AutoVu.pdf ,janvier 2020
- [6] Site web : <http://doc.openalpr.com/> .Décembre 2019.
- [7] Site web: <https://openclassrooms.com/fr/courses/>, Pascale Monasse, Kimia Nadjahi, «Classez et segmentez des données visuelles», .février 2020.
- [8] Site web: <https://www.imedias.pro/cours-en-ligne/graphisme-design/definition-resolution-taille-image/les-images-vectorielles-matricielles/> ,Communauté Imedias , .Mars 2019
- [9] Jean-Marie MUNGUAKONKWA BIRINGANINE ,«La liaison automatique des plusieurs images perçues sur un scanner » Mémoire master .Juin 2019.
- [10] Abdelouahed Abdou sabri , «Introduction au traitement d'images» , slideshare, 5 juin 2014.
- [11] Zine El Abadine , Lyazid Aoufi ,Moussa Semchedine , «Nouvelle Approche de filtrage Non-linéaire d'images numériques» ,Faculté des sciences Université Ferhat Abbas, Novembre 2011.
- [12] Max Mignotte , « Traitement d'image numérique » , Cours ,Université Montréal ,2012.
- [13] Site web : https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html .Mars 2019.
- [14] Ouchraa Selma, Badri hichem , Drissi el Maliani Zakaria , «Image numérique» , projet de fin d'études , Université Mohamed V Rabat ,2009-2010.
-

Bibliographie

- [15] Gérard Swinnen, « Apprendre à programmer avec Python 3 », Edition Eyrolles, 2000-2012.
- [16] Interview avec Jean-Claude Heudin , « Comment est née l'intelligence artificielle ? »,Groupe MadelnFutura , 2015.
- [17] GAEL , «3 Algorithmes de Deep Learning expliqués en Langage Humain », article DATAKEEN, 25 février 2018.
- [18] Site web: <https://www.senat.fr/>, «Pour une intelligence artificielle maîtrisée, utile et démystifiée-Rapport », Avril 2017.
- [19] Dominique Cardon ,Jean-Philippe Cointet ,Antoine Mazières , « Machine prédictives», Edition La Découverte , 2018.
- [20] Site web: <https://openclassrooms.com/> , Chantal Gribaumont, «Administrez vos base de données avec Mysql», février 2020 .
- [21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg , , « SSD: Single Shot MultiBox Detector »,article scientifique ,Université Cornell,2015.
- [22] Site web : <https://www.bigstockphoto.com/fr/> .février.2020.
-