

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université ABDERAHMANE MIRA – BEJAIA



Faculté des Sciences Exactes

Département Informatique

Mémoire de Master en Informatique

Option : Systèmes d'Information Avancés

Thème

Système Intelligent de Reconnaissance des
Plaques d'Immatriculation Algériennes par
Apprentissage Profond

Présenté par :

OUALI Anes ALICHE Yazid

Soutenu devant le jury composé de :

<i>Président</i>	Mr K. AKILAL	MCB	U. A/Mira Béjaïa
<i>Examineur</i>	Mr R. OUZEGGANE	MAA	U. A/Mira Béjaïa
<i>Promotrice</i>	Mme Z. TAHAKOURT	MCB	U. A/Mira Béjaïa

Promotion : 2023-2024

Remerciements

D'abord, nous exprimons notre gratitude envers Dieu, le Tout-Puissant, pour nous avoir accordé la force et le courage nécessaires pour achever ce modeste travail.

Ensuite, nous exprimons notre gratitude envers notre promotrice, Madame TAHAKOURT ZINEB, qui est enseignante à notre université ABDERRAHMANE MIRA de BEJAIA. Nous lui sommes reconnaissants pour sa patience, son soutien, sa disponibilité, ainsi que pour ses idées et engagements judicieux qui ont joué un rôle incontestable dans la réalisation de ce mémoire.

Nous exprimons également notre gratitude envers le président et les membres du jury qui ont accepté d'évaluer ce projet. Qu'ils trouvent ici la preuve de notre profonde reconnaissance.

Je tiens à exprimer ma profonde gratitude envers tous les membres du personnel administratif de l'université de Bejaia, qu'ils soient enseignants ou responsables de spécialité, pour leur soutien tout au long de ces cinq années.

Nous exprimons enfin notre gratitude envers nos parents et nos proches pour leur soutien et leurs précieux encouragements.

Dédicace

Je dédie ce rapport à mes chers parents, qui ont été mes piliers tout au long de ce parcours.

À ma mère, dont la force et le dévouement ont été une source constante d'inspiration, et à mon père, dont le soutien inconditionnel m'a donné la confiance nécessaire pour avancer.

À mes frères et sœurs, complices de mes joies et compagnons de mes peines, je vous adresse toute ma gratitude.

Enfin, à mes amis, qui ont su être présents à chaque étape, Je leur adresse toute ma gratitude. Ce travail est le fruit de leur collaboration et de leur amitié.

Table des matières

Table des matières	iv
Table des figures	v
Liste des tableaux	vii
Liste des acronymes	viii
Introduction générale	1
I Généralités sur le traitement d'images	3
1 Introduction	3
2 Notions sur le traitement d'image	4
2.1 Image numérique	4
2.2 Types d'images	4
2.3 Caractéristiques d'une image	5
2.3.1 Définition et résolution	5
2.3.2 Profondeur de couleur	6
2.4 Formats d'images	6
2.5 Codage des couleurs	6
2.5.1 Image noir et blanc	7
2.5.2 Images à niveaux de gris	7
2.5.3 Images en couleur	8
2.6 Définition du traitement d'images	9
2.7 Les techniques de traitement d'image	10
2.7.1 Échelle de gris (gray scaling)	10
2.7.2 Filtrage d'image	10
2.7.3 Seuillage (Thresholding)	11
2.7.4 Égalisation d'histogramme	12
2.7.5 Détection de contours	13
2.7.6 Transformation de perspective	13
2.7.7 Transformation morphologique	14

3	Généralités sur l'apprentissage Automatique	15
3.1	Fonctionnement de l'apprentissage automatique	16
3.2	Algorithmes de classification	16
3.2.1	Modèles de classification	16
4	Apprentissage profond	18
4.1	Fonctionnement du deep Learning	18
4.2	Les différents algorithmes de Deep Learning	19
4.2.1	Réseaux neuronaux convolutifs (CNN)	19
4.2.2	Réseaux neuronaux récurrents (RNN)	19
4.2.3	Réseaux de mémoire à long et court terme (LSTM)	19
4.2.4	Réseaux adversariaux génératifs (GAN)	19
5	OCR	20
6	Conclusion	20
II	État de l'art sur les systèmes de reconnaissance des plaques d'immatriculation (ANPR)	21
1	Introduction	21
2	Définition d'un Automatic Number Plate Recognition (ANPR)	22
3	Les composants d'un ANPR	22
3.1	Composants matériels	23
3.2	Composants logiciels	23
3.3	Composants additionnels	23
4	Fonctionnement des ANPRs	24
5	Application et cas d'utilisation	24
6	Avantages d'utilisation des ANPRs	26
7	Facteurs impactant la robustesse des systèmes ANPR	26
8	Techniques et méthodes utilisés pour les systèmes ANPR	28
8.1	Détection de la plaque d'immatriculation	29
8.1.1	Méthode basée sur You Only Look Once (YOLO)	29
8.1.2	Méthode basée sur l'architecture SegNet	31
8.1.3	Méthode basée sur Convolutional Neural Networks (CNN)	31
8.1.4	Méthode basée sur un modèle d'attention visuelle	33
8.1.5	Méthode basée sur l'analyse des composants connectés	33
8.2	Segmentation des caractères	33
8.2.1	Méthode de détection des contours	33
8.2.2	Méthode de segmentation des caractères basée sur le seuillage et les opérations morphologiques	35

8.2.3	Méthode de segmentation basée sur la projection	35
8.2.4	Segmentation utilisant le clustering k-means optimisé. .	36
8.2.5	Méthode basée sur YOLO	36
8.3	Reconnaissance des caractères	37
8.3.1	Méthode basée sur Optical Character Recognition (OCR)	37
8.3.2	Reconnaissance des caractères avec CNN	37
8.3.3	Méthode basée sur les Support Vector Machines (SVM)s et les CNNs	38
9	Récapitulation des travaux antérieure	39
10	Conclusion	39

III DZANPR : Système de reconnaissance des plaques d'immatriculation algériennes **41**

1	Introduction	41
2	Objectifs du projet	42
3	Caractéristiques des plaques d'immatriculation algériennes	42
3.1	Structure et format des plaques	43
3.2	Numérotation des plaques	43
3.3	Classes des véhicules	43
3.4	Identifiant des wilayas	44
3.5	Spécifications des plaques	44
3.6	Plaques spéciales	44
4	Vue générale sur le système Algerian Number Plate Recognition System (DZANPR)	45
5	Architecture proposée	45
6	Phase de détection	46
7	Phase de segmentation	46
8	Phase de reconnaissance	47
9	Pourquoi avons-nous choisi YOLO pour les tâches de détection et segmentation ?	48
9.1	Avantages de YOLOv8	49
10	Conclusion	49

IV Expérimentations **50**

1	Introduction	50
2	Environnement de développement	51
2.1	Environnement matériel	51
2.2	Environnements logiciel	51

2.2.1	Langages de développement	51
2.2.2	Bibliothèques pour le traitement d'images et la visualisation	51
2.2.3	Frameworks et API d'Apprentissage Profond	52
2.2.4	Outils et plates-formes pour les données et l'entraînement	52
2.2.5	Outils pour l'interopérabilité et l'optimisation des modèles	53
2.2.6	Déploiement et interface de test	53
3	Ensembles de données (DataSets)	53
3.1	DataSet pour la détection des plaques	53
3.2	DataSet pour la détection des quatres points	54
3.3	DataSet pour segmentation des caractères	54
3.4	DataSet pour reconnaissance des chiffres	55
4	Entraînement des modèles	57
4.1	Entraînement de modèle de détection des plaques	57
4.2	Entraînement de modèle de segmentation des caractères	57
4.3	Entraînement de CNN pour la classification des chiffres	58
4.4	Tableau récapitulatif des modèles entraînés	58
4.5	Exportation des modèles	59
5	Évaluation des modèles	59
5.1	Evaluation de modèle de détection des plaques	59
5.2	Évaluation de modèle de segmentation des caractères	60
5.3	Évaluation de modèle de reconnaissance des chiffres	60
6	Évaluation du système complet	61
7	Interface de test	63
8	Conclusion	63
	Conclusion générale	64
	Bibliographie	66

Table des figures

I-1	Exemple d'une image pixelisée	5
I-2	Une image noir et blanc	7
I-3	Les niveau de gris	8
I-4	Représentation cubique de l'espace RVB	8
I-5	Représentation du modèle TSL (1)	9
I-6	Image en graysale	10
I-7	Exemple seuillage d'une image	11
I-8	Exemple seuillage d'une image en couleur	12
I-9	Exemple égalisation d'histogramme	12
I-10	Exemple de détection des contours	13
I-11	Exemple de la transformation de perspective	14
I-12	Etapas de la transformation morphologique	15
I-13	Exemple de fonctionnement d'un réseau de neurones.	18
II-1	Illustration d'un ANPR	22
II-2	Les composants principales d'un ANPR	23
II-3	Les facteurs Impactant la Robustesse des ANPR	28
III-1	Exemple des plaques d'immmatriculation algériennes	43
III-2	Schéma général du système	45
III-3	Architecture de DZANPR	46
III-4	Processus de segementation des des caractères	47
III-5	Architecture CNN	48
IV-1	Exemple de détection en temps réel avec YOLOv8	52
IV-2	Exemples d'images de dataSet pour la détection de la matricule	54
IV-3	Exemples d'images de dataSet pour la détection des 4 points de la plaque	54
IV-4	Exemples d'images de dataSet pour la segmentation des caractères	55
IV-5	Exemples d'images de dataSet pour la reconnaissance des chiffres	56
IV-6	Matrice de confusion pour la reconnaissance des chiffres	61
IV-7	Exemples d'images mal reconnues par le système	62

IV-8	Exemples d'images bien reconnues par le système	62
IV-9	Interface de test du système de reconnaissance des plaques d'immatriculation	63

Liste des tableaux

II.1 Comparaison des approches de détection, segmentation et reconnaissance des plaques d'immatriculation dans divers travaux de recherche	39
III.1 Classes des véhicules (2)	44
III.2 Spécifications des plaques (2)	44
IV.1 Tableau récapitulatif des différents datasets utilisés dans l'entraînement .	56
IV.2 Tableau des modèles entraînés	58
IV.3 Les délais d'exécution	59
IV.4 Matrice de confusion pour la détection des plaques d'immatriculation . .	60
IV.5 Matrice de confusion pour la segmentation des plaques d'immatriculation	60

Liste des acronymes

ANPR Automatic Number Plate Recognition
CCA Analysis of Connected Components
CNN Convolutional Neural Networks
DZANPR Algerian Number Plate Recognition System
EPS Encapsulated PostScript
GAN Generative Adversarial Networks
GIF Graphics Interchange Format
GPS Global Positioning System
HSV Hue Saturation Value (color model)
JPEG Joint Photographic Experts Group
K-means K-means Clustering
KNN K-Nearest Neighbors
LSTM Long Short Term Memory Networks
OCR Optical Character Recognition
PAO Publication assistée par ordinateur
PDF Portable Document Format
PNG Portable Network Graphics
RNN Recurrent Neural Networks
RGB Red Green Blue (color model)
SVG Scalable Vector Graphics
SVM Support Vector Machines
TSL Tint Saturation Lightness
YOLO You Only Look Once

Introduction générale

La reconnaissance automatique des plaques d'immatriculation (ANPR) est devenue une technologie omniprésente dans l'application de la loi, la gestion du trafic et la sécurité. Les approches ANPR existantes vont des méthodes traditionnelles aux techniques avancées d'apprentissage profond, permettant une grande variété d'applications réelles.

Cependant, obtenir des performances robustes reste un défi en raison de divers facteurs. Les conditions environnementales comme une mauvaise luminosité, les ombres et les intempéries peuvent avoir un impact significatif sur la qualité des images. De plus, les plaques d'immatriculation elles-mêmes peuvent être obscurcies par la saleté, les dommages ou des variations de géométrie, de langue, de polices et d'autres éléments visuels. Les facteurs d'image tels que la faible résolution, la distorsion, l'inclinaison et les mauvais réglages de la caméra compliquent encore davantage la tâche. Ces facteurs peuvent agir de manière indépendante ou se combiner pour créer des scénarios encore plus difficiles pour les systèmes ANPR.

En Algérie, l'absence d'un système ANPR robuste adapté aux défis spécifiques de la région représente un obstacle significatif. Nous proposons dans ce projet une nouvelle approche pour l'ANPR algérien qui exploite la puissance des techniques d'apprentissage profond, en particulier You Only Look Once (YOLO) et les réseaux de neurones convolutifs (CNN). En relevant les défis uniques rencontrés en Algérie, cette approche vise à obtenir une précision supérieure de détection et de reconnaissance des plaques d'immatriculation par rapport aux solutions génériques existantes

Ce mémoire est composé de quatre chapitres :

Dans le premier chapitre « **Généralités sur le traitement d'images** » nous explorons les bases fondamentales de la représentation et du traitement des images, ainsi que les principales techniques d'apprentissage automatique et de deep learning utilisées dans ce domaine.

Dans le deuxième chapitre « **Travaux connexes** » , nous présentons une revue des systèmes ANPR, en abordant leurs composants, leur fonctionnement, leurs applications, ainsi que les défis auxquels ils font face et les techniques couramment employées à chaque étape du processus.

Dans le troisième chapitre « **DZANPR : Système de reconnaissance des plaques d'immatriculation algériennes** » nous présentons notre approche spécifique pour la reconnaissance des plaques d'immatriculation algériennes. Nous décrivons les besoins fonctionnels et les caractéristiques spécifiques des plaques d'immatriculation en Algérie, et détaillerons l'architecture de notre système ANPR.

Dans le quatrième chapitre « **Expérimentations** » est consacré aux tests et résultats de notre système. Nous y décrivons l'environnement de développement, les ensembles de données utilisés, le processus d'entraînement des modèles, et les performances obtenues.

Nous terminons par une conclusion générale sur le résumé de travail. Nous identifions également plusieurs perspectives pour des recherches futures et des applications potentielles.

Chapitre I

Généralités sur le traitement d'images

1 Introduction

Le traitement d'image est un domaine de l'informatique qui se concentre sur l'analyse, la manipulation et l'amélioration des images numériques. Utilisé dans une variété d'applications allant de la médecine à la reconnaissance faciale, le traitement d'image permet de transformer des données visuelles brutes en informations utiles et interprétables. Dans ce chapitre nous explorons des concepts fondamentaux et des techniques avancées dans le domaine du traitement d'image et de l'apprentissage automatique, avec un focus particulier sur l'apprentissage profond.

Ce chapitre est structuré en trois parties principales, chacune visant à fournir une compréhension approfondie des éléments essentiels qui sous-tendent notre travail.

La première partie se concentre sur les notions de base du traitement d'image. Nous abordons la définition de l'image, les différents types et formats d'images, ainsi que leurs caractéristiques importantes. Cette section inclut également le codage des couleurs et une définition du traitement d'image, suivie des principales étapes et techniques utilisées dans ce domaine.

La deuxième partie porte sur les généralités de l'apprentissage automatique. Nous y définissons l'apprentissage automatique, expliquons son fonctionnement, et discutons des principaux algorithmes de classification, ainsi que des modèles de classification couramment utilisés.

Enfin, la troisième partie est dédiée à l'apprentissage profond. Nous y définissons le deep learning, expliquons son fonctionnement, et explorons brièvement les principaux algo-

rithmes tels que les réseaux neuronaux convolutifs (CNN), les réseaux neuronaux récurrents (RNN), et les réseaux adversariaux génératifs (GAN).

À travers ces trois sections, ce chapitre vise à fournir un cadre théorique et pratique solide pour comprendre comment les techniques modernes de traitement d'image et d'apprentissage automatique peuvent être appliquées pour résoudre des problèmes complexes et innovants.

2 Notions sur le traitement d'image

Une **image** est une représentation visuelle d'un objet, d'une scène, d'une personne ou d'une abstraction, produite sur une surface par le dessin, la peinture, la photographie, ou d'autres moyens artistiques. Elle peut également être une reproduction inversée d'un objet renvoyée par une surface réfléchissante, comme un miroir, ou une représentation mentale retrouvée par le souvenir des objets perçus. En résumé, une image est une reproduction visuelle d'un objet réel ou une représentation qui évoque une réalité, pouvant être produite de diverses manières artistiques ou optiques (3; 4).

2.1 Image numérique

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques) (5).

La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts). C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur (5)

2.2 Types d'images

Il existe deux types d'images numériques : les images matricielles et les images vectorielles (6).

Une **image matricielle** est composée d'une grille de points, appelés pixels. Plus la densité de ces points est élevée, plus la quantité d'informations est grande, ce qui augmente la résolution de l'image. En conséquence, la taille du fichier et le temps de traitement seront

également plus importants. Les images affichées sur un écran de télévision ou une photographie sont des exemples d'images matricielles. On obtient aussi des images matricielles avec un appareil photo numérique, une caméra vidéo numérique ou un scanner.

En revanche, une **image vectorielle** est constituée de formes géométriques simples décrites mathématiquement. Ces images sont principalement utilisées pour créer des schémas ou des plans. Les logiciels de dessin industriel fonctionnent selon ce principe, et les principaux logiciels de traitement de texte ou de Publication assistée par ordinateur (PAO) proposent également des outils de ce type. Les images vectorielles présentent deux avantages : elles occupent peu d'espace en mémoire et peuvent être redimensionnées sans perte de qualité.

2.3 Caractéristiques d'une image

L'image est un ensemble structure d'informations caractérisée par les paramètres suivants :

2.3.1 Définition et résolution

- a) **Définition** : Elle représente le nombre total de pixels composant l'image. Plus la définition est élevée, plus l'image est détaillée et riche en informations. Elle s'exprime généralement en nombre de mégapixels (MP), par exemple 12MP pour une image de 12 millions de pixels.
- b) **Résolution** : Elle exprime la densité de pixels par unité de surface (pixels par pouce, PPP ou DPI en anglais). Plus la résolution est élevée, plus les pixels sont fins et l'image nette. Elle est importante pour l'impression, car elle détermine la qualité du rendu sur papier.



FIGURE I-1 – Exemple d'une image pixelisée

(6)

2.3.2 Profondeur de couleur

Elle définit le nombre de couleurs pouvant être représentées dans l'image. Elle s'exprime en bits par pixel (bpp). Plus la profondeur de couleur est élevée, plus la palette de couleurs est large et les nuances subtiles. Les images les plus courantes utilisent 24bpp (couleurs 24 bits), permettant d'afficher plus de 16 millions de couleurs.

2.4 Formats d'images

Il existe plusieurs formats d'images (7).

- a) **JPEG ou JPG** c'est un des formats les plus utilisés en ligne pour les photos et les images. Grâce à sa capacité de compression, il est très pratique pour la publicité numérique. On peut l'éditer facilement dans plusieurs logiciels.
- b) **GIF** Ce type de fichier offre des capacités très intéressantes : il permet de gérer les images animées et la transparence (très pratique!), en plus d'offrir une grande capacité de compression. Ce n'est cependant pas un bon format pour l'impression, car la palette de couleurs est limitée.
- c) **PNG** on pourrait dire que ce format représente l'évolution du GIF. Il permet la transparence, tout en offrant beaucoup plus de couleurs (des millions!). Parfait pour le web et les illustrations, il n'est toutefois pas recommandé pour les photos et l'imprimé.
- d) **PDF** C'est un format bien connu qui possède l'énorme avantage d'être très portable : il fonctionne sur de multiples plateformes et dans de nombreux environnements différents. Pour le web ou l'imprimé, c'est un standard quasi universel.
- e) **EPS** Typiquement utilisé pour les logos, ce type de fichier vectoriel est parfait pour garantir la meilleure résolution, peu importe la dimension de l'image finale.
- f) **SVG** Adapté pour les écrans de type Retina, ce format offre de la haute résolution dans de très petits fichiers. Son contenu étant visible à partir des navigateurs web, c'est un format très utile pour les supports numériques.

2.5 Codage des couleurs

Nous avons vu une image apparait comme une matrice ou chaque case contient des nombres associés à une couleur. Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- Le noir et blanc.
- Les niveaux de gris.

— En couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique (6).

2.5.1 Image noir et blanc

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux (6).



FIGURE I-2 – Une image noir et blanc

(8)

2.5.2 Images à niveaux de gris

Le codage dit en niveaux de gris permet d'obtenir plus de nuances que le simple noir et blanc. Il offre des possibilités supplémentaires pour coder le niveau de l'intensité lumineuse. La couleur est codée souvent sur un octet soit 8 bits ce qui offre la possibilité d'obtenir 256 niveaux de gris (0 pour le noir et 255 pour le blanc). On peut aussi le faire avec 16 niveaux de gris (6).

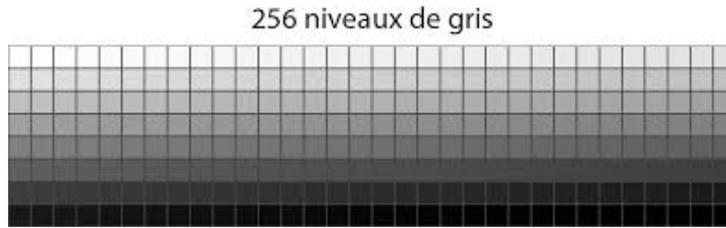


FIGURE I-3 – Les niveau de gris
(6)

2.5.3 Images en couleur

Les espaces colorimétriques d'une image sont des modèles mathématiques qui représentent les couleurs visibles par un être humain. Voici quelques explications sur les principaux espaces colorimétriques :

- a) **Espace RVB** : Le modèle de couleur RVB (rouge, vert, bleu) est un modèle de couleur additif utilisé pour créer une variété de couleurs en combinant différentes quantités de rouge, vert et bleu. Lorsque ces trois couleurs sont mélangées, elles produisent une large gamme de couleurs visibles à l'œil humain. Le RVB est largement utilisé dans les écrans électroniques tels que les télévisions, les moniteurs d'ordinateur, les caméras numériques, et même dans l'éclairage. Chaque composante RVB peut varier de 0 (noir) à 1 (blanc), avec des valeurs allant de 0 à 255 pour chaque couleur. En combinant ces valeurs, on peut obtenir jusqu'à 16,777,216 couleurs différentes dans la palette RVB (9).

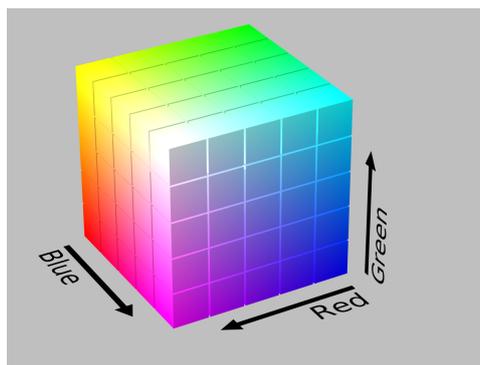


FIGURE I-4 – Représentation cubique de l'espace RVB
(10)

- b) **Modèle TSL** : est un modèle colorimétrique perceptuel car il se rapproche fortement de la perception physiologique de la couleur par l'œil humain. Dans ce système,

les couleurs sont toujours caractérisées par trois dimensions mais qui ont une signification tout autre que dans le modèle RVB, puisqu'elles représentent ici la teinte, la saturation et la luminosité. On représente généralement le modèle TSL à l'aide de deux cônes inversés placés l'un au-dessus de l'autre (1).

- a) **La teinte** qui correspond à la perception de la couleur est mesurée sur une échelle circulaire (cercle de chromaticité de Newton) par un angle de 0° à 360° .
- b) **La saturation** mesure le degré de pureté d'une couleur, c'est-à-dire la quantité de gris ajoutée à la couleur. Elle est représentée par le rayon d'une section circulaire du cône et varie de 1 (couleur pure ou saturée) à 0 (niveau de gris correspondant).
- c) **La luminosité** représente le degré d'éclaircissement ou d'assombrissement d'une couleur. Elle est définie selon une échelle linéaire allant de 0 (noir) à 1 (blanc) en passant par tous les niveaux de gris. Sur la figure ci-dessous, la luminosité est représentée par la droite joignant les deux sommets des cônes et passant par le centre du cercle chromatique. Les teintes sur le cercle chromatique sont donc toutes au même niveau de luminosité de 50%.

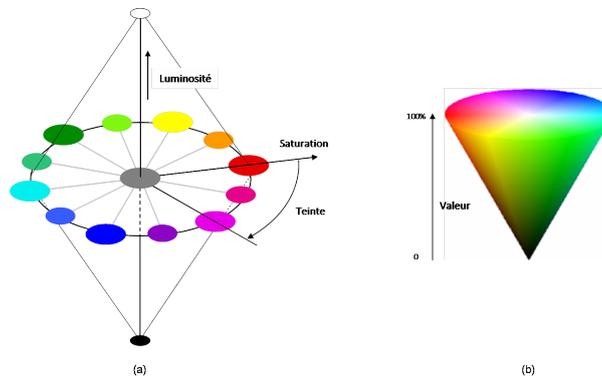


FIGURE I-5 – Représentation du modèle TSL (1)

2.6 Définition du traitement d'images

Le **traitement d'images** est une discipline de l'informatique et des mathématiques appliquées qui se concentre sur l'étude des images numériques et leurs transformations, avec pour objectif d'améliorer leur qualité ou d'en extraire des informations pertinentes. Il s'agit d'un sous-ensemble du traitement du signal, spécifiquement dédié aux images et aux données dérivées telles que la vidéo, contrairement à celui consacré aux sons et autres types de signaux. (11)

2.7 Les techniques de traitement d'image

Le traitement d'image englobe diverses techniques permettant d'analyser et de manipuler les images numériques pour en extraire des informations utiles

2.7.1 Échelle de gris (gray scaling)

En photographie numérique, imagerie générée par ordinateur et colorimétrie, une image en niveaux de gris est composée uniquement de nuances de gris, représentant des intensités de lumière sans informations de couleur. Ces images diffèrent des images binaires en noir et blanc, qui n'ont que deux couleurs : noir et blanc.

Les images en niveaux de gris peuvent être créées en mesurant l'intensité de la lumière à chaque pixel selon une combinaison spécifique de fréquences. Elles sont réellement monochromatiques lorsqu'une seule fréquence est capturée. Ces fréquences peuvent appartenir à n'importe quelle partie du spectre électromagnétique, comme l'infrarouge ou la lumière visible.

Une image en niveaux de gris colorimétrique a un espace colorimétrique défini qui mappe les valeurs de gris aux propriétés de la vision humaine. Sans un espace colorimétrique défini pour l'image couleur d'origine, il n'existe pas de méthode unique pour convertir une image couleur en une image en niveaux de gris tout en conservant la même intensité perçue (12).



FIGURE I-6 – Image en graysale
(13)

2.7.2 Filtrage d'image

Le filtrage d'image est une opération courante en traitement d'images qui permet de modifier l'apparence d'une image en appliquant différents filtres mathématiques. Voici quelques explications sur les principaux types de filtrage :

Le filtrage par moyenne : remplace chaque pixel par la moyenne des pixels voisins, ce qui a pour effet d'adoucir les détails et d'atténuer le bruit dans l'image. C'est un filtre simple mais efficace pour lisser les images. (14)

Le flou gaussien : est similaire au filtrage par moyenne mais utilise une fonction gaussienne pour pondérer l'influence des pixels voisins. Cela permet d'obtenir un lissage plus doux et naturel des contours. Ce filtre est souvent utilisé pour réduire le bruit tout en préservant les structures importantes. (14).

Le filtrage médian : remplace chaque pixel par la médiane des valeurs des pixels voisins. Ce filtre est particulièrement efficace pour réduire le bruit impulsionnel (pixels isolés très clairs ou très sombres) tout en préservant les contours. Il est très utilisé en pré-traitement pour nettoyer les images. (14).

Le filtrage bilatéral : combine un filtrage par moyenne pondéré par la distance spatiale et un filtrage par la différence d'intensité entre pixels voisins. Cela permet de lisser les zones uniformes tout en préservant les contours nets. C'est un bon compromis entre lissage et préservation des détails (14).

2.7.3 Seuillage (Thresholding)

Le seuillage d'image est la méthode la plus simple de segmentation d'image. À partir d'une image à niveau de gris, le seuillage d'images peut être utilisé pour créer une image comportant uniquement deux valeurs, noir ou blanc (monochrome). On remplace un à un les pixels d'une image par rapport à une valeur seuil fixée (par exemple 123). Ainsi, si un pixel a une valeur supérieure au seuil (par exemple 150), il prendra la valeur 255 (blanc), et si sa valeur est inférieure (par exemple 100), il prendra la valeur 0 (noir) (6).

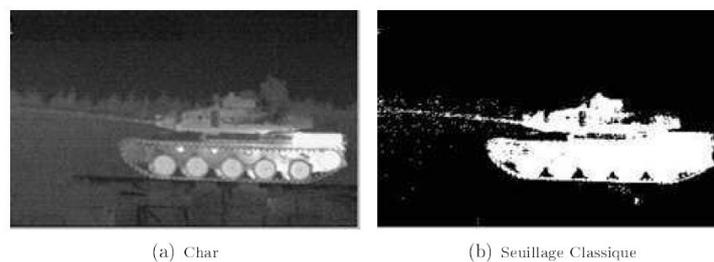


FIGURE I-7 – Exemple seuillage d'une image

(15)

Avec une image en couleur, on fera de même avec les trois composantes rouge, vert et bleu. Il y aura ainsi huit couleurs possibles pour chaque pixel : blanc, noir, rouge, vert, bleu, magenta, jaune et cyan (6).



FIGURE I-8 – Exemple seuillage d'une image en couleur
(6)

2.7.4 Égalisation d'histogramme

Égalisation d'histogramme est une méthode d'ajustement du contraste d'une image numérique qui utilise l'histogramme. Elle consiste à appliquer une transformation sur chaque pixel de l'image, et donc d'obtenir une nouvelle image à partir d'une opération indépendante sur chacun des pixels. Cette transformation est construite à partir de l'histogramme cumulé de l'image de départ.

L'égalisation d'histogramme permet de mieux répartir les intensités sur l'ensemble de la plage de valeurs possibles, en étalant l'histogramme. L'égalisation est intéressante pour les images dont la totalité, ou seulement une partie, est de faible contraste (l'ensemble des pixels sont d'intensité proches). La méthode est rapide, facile d'implémentation, et complètement automatique (11).

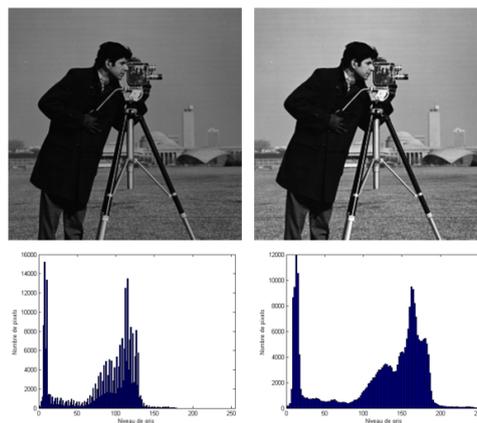


FIGURE I-9 – Exemple égalisation d'histogramme
(6)

2.7.5 Détection de contours

La détection de contours est le procédé permettant de repérer les points d'une image matricielle qui correspondent à un changement brutal de l'intensité lumineuse. Ces changements de propriétés de l'image numérique indiquent en général des éléments importants de structure dans l'objet représenté. Ces éléments incluent des discontinuités dans la profondeur, dans l'orientation d'une surface, dans les propriétés d'un matériau et dans l'éclairage d'une scène (16).

La détection des contours dans une image réduit de manière significative la quantité de données en conservant des informations qu'on peut juger plus pertinentes. Il existe un grand nombre de méthodes de détection des contours de l'image mais la plupart d'entre elles peuvent être regroupées en deux catégories. La première recherche les extremums de la dérivée première, en général les maximums locaux de l'intensité du gradient. La seconde recherche les annulations de la dérivée seconde, en général les annulations du laplacien ou d'une expression différentielle non linéaire (16).

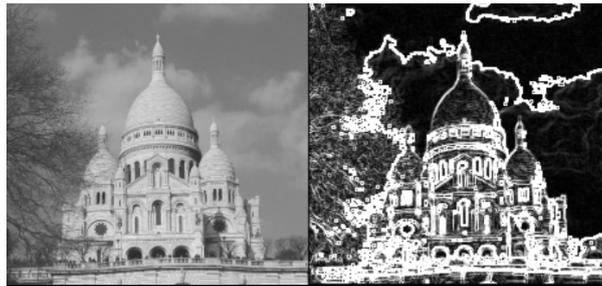


FIGURE I-10 – Exemple de détection des contours
(6)

2.7.6 Transformation de perspective

Est une fonctionnalité très utile si on souhaite aligner correctement l'image. Il transforme l' image de manière directe après que la transformation de perspective lui soit appliquée (17).

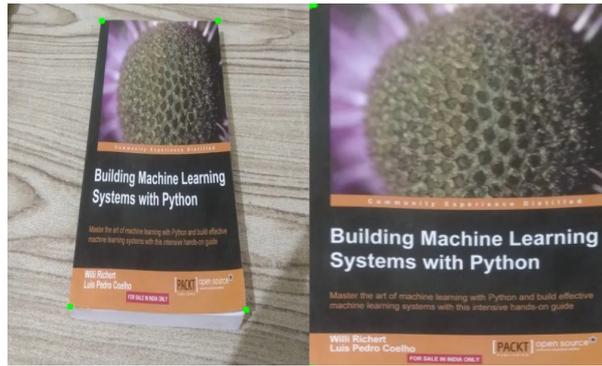


FIGURE I-11 – Exemple de la transformation de perspective
(18)

2.7.7 Transformation morphologique

Est un ensemble d'opérations simples basées sur la forme de l'image. Il est normalement réalisé sur des images binaires. Il a besoin de deux entrées, l'une est notre image originale, la seconde est appelée élément structurant ou noyau qui décide de la nature du fonctionnement. Deux opérateurs morphologiques de base sont l'érosion et la dilatation. Puis ses variantes se forment comme l'ouverture, la fermeture, le gradient, etc., entre également en jeu. (19).

- a) **Erosion** : L'érosion en traitement d'image réduit la taille des objets de premier plan en utilisant un noyau glissant. Un pixel reste 1 seulement si tous les pixels sous le noyau sont 1 ; sinon, il devient 0. Cela diminue l'épaisseur des objets de premier plan, élimine les petits bruits blancs et sépare les objets connectés.
- b) **Dilatation** : La dilatation est l'opposée de l'érosion en traitement d'image. Un pixel devient 1 si au moins un pixel sous le noyau est 1, augmentant ainsi la taille des objets de premier plan. Après l'érosion pour enlever les bruits blancs, on utilise souvent la dilatation pour restaurer la taille des objets sans réintroduire le bruit. Cela permet également de rassembler les parties cassées d'un objet.
- c) **Ouverture** : L'ouverture n'est qu'un autre nom d'érosion suivie de la dilatation. Il est utile pour éliminer le bruit, comme nous l'avons expliqué ci-dessus.
- d) **Fermeture** : Est inverse de l'ouverture, la dilatation suivie de l'érosion. Il est utile pour fermer les petits trous à l'intérieur des objets de premier plan, ou de petits points noirs sur l'objet.
- e) **Gradient morphologique** : C'est la différence entre la dilatation et l'érosion d'une image. Le résultat ressemblera au contour de l'objet.
- f) **Chapeau supérieur** : C'est la différence entre l'image d'entrée et l'ouverture de

l'image.

- g) **Chapeau noir** : C'est la différence entre la *fermeture* de l'image d'entrée et l'image elle même.



FIGURE I-12 – Etapes de la transformation morphologique

3 Généralités sur l'apprentissage Automatique

Après avoir exploré les différentes méthodes de traitement d'image, il est crucial de comprendre comment ces techniques se combinent avec des approches plus avancées pour améliorer la précision et l'efficacité des algorithmes. C'est ici qu'intervient l'apprentissage automatique, une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre et de s'améliorer à partir de données. En exploitant des algorithmes sophistiqués, l'apprentissage automatique offre des outils puissants pour l'analyse et la reconnaissance des motifs dans les images.

L'apprentissage automatique, aussi appelé apprentissage machine ou encore auto-apprentissage, désigne une branche de l'intelligence artificielle (IA) qui utilise des algorithmes afin d'imiter la capacité de raisonnement de l'être humain. Ce champ d'études est aujourd'hui appliqué dans de nombreux domaines et permet d'assister les humains dans la prise de décision et la résolution de problèmes complexes (20).

L'apprentissage automatique fonctionne à l'aide d'algorithmes capables de traiter une quantité considérable de données, impossibles à analyser à l'échelle humaine. Avec le

temps, les capacités de stockage et de traitement des données ont connu une croissance exponentielle, offrant une puissance de calcul toujours plus importante et des prédictions toujours plus précises. Voyons ensemble sur quels principes repose son fonctionnement (20).

3.1 Fonctionnement de l'apprentissage automatique

Le machine learning se charge d'analyser des données afin d'identifier des modèles et de prendre des décisions, qui pourront ensuite être utilisées dans divers cas d'application. Les algorithmes possèdent en général une fonction de classification ou de prévision : par exemple, savoir distinguer un chat d'un chien, identifier le meilleur coup à jouer aux échecs ou encore estimer l'itinéraire le plus rapide pour une application GPS (20).

Comme son nom l'indique, l'apprentissage automatique implique également que la machine "apprend" d'elle-même, c'est-à-dire qu'elle détermine la décision à prendre sans intervention humaine et améliore sa précision de façon autonome et itérative (20).

3.2 Algorithmes de classification

Un algorithme est un ensemble d'opérations suivies dans un ordre précis afin de résoudre un problème ou d'apporter de nouvelles solutions. Comme l'apprentissage par un système d'intelligence artificielle (21).

Algorithme de classification est utilisé en machine learning. Ils permettent au logiciel d'apprendre en toute autonomie à partir de plusieurs ensembles de données. L'idée est alors de classer les différents éléments d'un jeu de données en plusieurs catégories. Ces dernières regroupent les datas en fonction de leur similarité. Comme les datas présentent des caractéristiques communes, il est plus facile de prédire leur comportement (21).

3.2.1 Modèles de classification

Il existe de nombreux algorithmes d'apprentissage supervisé basés sur la classification. Nous avons réuni les principaux dans cette liste non exhaustive.

La machine à vecteurs de support (SVM) cet algorithme est considéré comme un classificateur linéaire. Son rôle est de séparer les jeux de données à travers des lignes (appelés hyperplans). Pour cela, l'algorithme doit maximiser les distances entre la ligne séparatrice et les différents échantillons situés de part et d'autre. Les plus proches de la ligne sont appelés les vecteurs supports (21).

L'idée est donc de trouver l'hyperplan optimal ; celui qui distingue parfaitement les deux classes afin de minimiser les erreurs de classification. Cela permet de séparer clairement les données pour identifier facilement des classes simples. Par exemple, des grandes et des petites villes. Lorsqu'il s'agit de données plus complexes, comme le matériel génétique d'un individu, il n'est pas si simple d'identifier les différentes catégories. Cet algorithme ne sera donc pas le plus pertinent (21).

Les arbres de décision L'arbre de décision est un algorithme classant les différentes données sous forme de branche. Il part d'une racine où chaque data prend une certaine direction en fonction de son comportement. Ce qui permet ensuite de prédire les variables de réponse.

Comme pour les arbres, les intersections sont appelées les nœuds, et leurs finalités, les feuilles. Les nœuds représentent alors les règles permettant de séparer les datas en différentes catégories, et les feuilles sont les informations elles-mêmes (21).

La répartition en K-moyennes (K-means) Cet algorithme de classification trie les données en différents groupes en fonction de leurs caractéristiques. Pour cela, il établit une moyenne de référence parmi un jeu de données, ce qui permet alors de définir un profil type.

L'avantage de l'algorithme K-means est sa précision. Et ce, même en traitant de grand volume de données rapidement. De par son efficacité, le K-means, rencontre une multitude d'applications : les moteurs de recherche pour proposer des résultats pertinents au regard des attentes des utilisateurs, les entreprises pour anticiper le comportement des prospects ou internautes, les responsables informatiques pour analyser la performance des systèmes et réseaux (21).

Le classificateur bayésien naïf Cet algorithme reprend le théorème de Bayes et les probabilités conditionnelles. Il repose sur les jeux de données étiquetés, et les associe à d'autres données non étiquetées pour les classer. Le classificateur bayésien naïf est principalement utilisé dans le traitement du langage naturel. Autrement dit, c'est ce qui permet aux machines de comprendre plus facilement le langage humain (21).

L'algorithme K-Nearest Neighbors (KNN) Le KNN peut être utilisé à la fois comme algorithme de régression ou de classification. Mais c'est souvent dans cette deuxième hypothèse qu'il est utilisé (21).

L'idée est alors de classer les variables d'un jeu de données en analysant les similitudes

entre elles. Pour cela, le KNN utilise un graphique et calcule la distance entre les différents points. Ceux qui sont les plus proches sont enregistrés dans la même catégorie (21).

4 Apprentissage profond

Le deep learning ou apprentissage profond est une branche de l'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 (22).

4.1 Fonctionnement du deep Learning

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 exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit (22).

À chaque étape, les mauvaises réponses sont éliminées et renvoyées vers les niveaux en amont comme illustre la figure I-13 pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ai jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant (22).

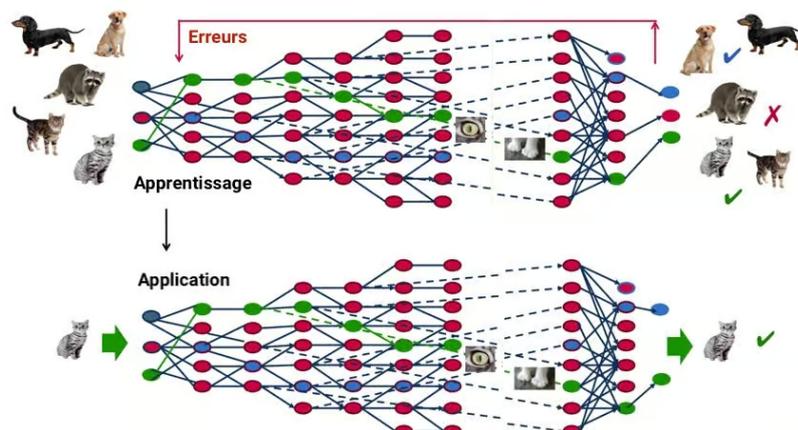


FIGURE I-13 – Exemple de fonctionnement d'un réseau de neurones.

(22)

4.2 Les différents algorithmes de Deep Learning

Plusieurs types d'algorithmes sont utilisés en apprentissage profond. Chaque algorithme possède ses propres spécificités et applications.

4.2.1 Réseaux neuronaux convolutifs (CNN)

Également appelés ConvNets, les CNN sont constitués d'une multitude de couches chargées de traiter et d'extraire les caractéristiques des données. De manière spécifique, les réseaux neuronaux convolutifs sont utilisés pour l'analyse et la détection d'objets. Ils peuvent donc servir par exemple à reconnaître des images satellites, traiter des images médicales, détecter des anomalies. (23).

4.2.2 Réseaux neuronaux récurrents (RNN)

Les réseaux neuronaux récurrents possèdent des connexions qui constituent des cycles dirigés. Cela permet aux sorties du Long Short Term Memory Networks (LSTM) d'être exploitées comme entrées au niveau de la phase actuelle. La sortie du LSTM se transforme en une entrée pour la phase actuelle. Elle peut donc mémoriser les entrées précédentes à l'aide de sa mémoire interne. Dans la pratique, les Recurrent Neural Networks (RNN) sont utilisés pour le sous-titrage d'images, le traitement du langage naturel et la traduction automatique (23).

4.2.3 Réseaux de mémoire à long et court terme (LSTM)

Les LSTM sont des dérivés de RNN. Ils peuvent apprendre et mémoriser des dépendances sur une longue durée. Les LSTM conservent ainsi les informations mémorisées sur le long terme. Ils sont particulièrement utiles pour prédire des séries chronologiques, car ils se rappellent des entrées précédentes. Outre ce cas d'utilisation, les LSTM sont également utilisés pour composer des notes de musique et reconnaître des voix (23).

4.2.4 Réseaux adversariaux génératifs (GAN)

Les GAN créent de nouvelles instances de données qui s'apparentent aux données d'apprentissage profond. Ils possèdent deux principaux composants : un générateur et un discriminateur. Si le générateur apprend à produire des informations erronées, le discriminateur, quant à lui, apprend à exploiter ces fausses informations. Les GAN sont généralement utilisés par les créateurs de jeux vidéo pour améliorer les textures 2D (23).

5 OCR

L'OCR, ou reconnaissance optique de caractères, est une technologie qui permet de convertir des images de texte imprimé ou manuscrit en données textuelles exploitables par un ordinateur. Elle permet de numériser et d'extraire le contenu textuel à partir de documents papier, de PDF, d'images, etc. pour le rendre modifiable et searchable (24).

L'OCR est largement utilisé dans de nombreux domaines tels que la bureautique, l'archivage, l'automatisation des processus, etc. Elle facilite grandement la gestion et l'exploitation des documents papier en les transformant en format numérique.

6 Conclusion

Dans ce chapitre, nous avons présenté un aperçu introductif des notions fondamentales dans le domaine du traitement d'image, de l'apprentissage automatique et de l'apprentissage profond. Nous avons exploré les définitions essentielles de ces disciplines et mis en lumière leur importance croissante dans divers domaines d'application, en particulier la reconnaissance automatique de modèles visuels complexes. Ces technologies constituent le socle de nombreuses innovations modernes, allant des véhicules autonomes aux systèmes de surveillance avancés.

Dans le chapitre suivant, nous présenterons un état de l'art détaillé de notre mémoire, en examinant les avancées récentes et les méthodes utilisées pour la reconnaissance automatique des plaques d'immatriculation. Nous analyserons les techniques actuelles, telles que les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN) et les architectures de type YOLO (You Only Look Once), en expliquant comment elles permettent d'atteindre des niveaux de précision et d'efficacité sans précédent.

Nous aborderons également les défis spécifiques de ce domaine, comme la gestion des variations d'éclairage, des angles de prise de vue et des occlusions, ainsi que les solutions proposées par la recherche récente pour surmonter ces obstacles.

Chapitre II

État de l'art sur les systèmes de reconnaissance des plaques d'immatriculation (ANPR)

1 Introduction

Les ANPRs sont devenus des outils indispensables dans divers domaines, allant de la gestion du trafic à la sécurité publique. Le présent chapitre a pour objectif de fournir une vue d'ensemble entier de ces systèmes, en abordant leurs composants, leur fonctionnement, leurs applications, leurs défis, ainsi que les techniques et algorithmes les plus couramment utilisés pour chaque phase du processus ANPR.

Nous débuterons par une définition précise de ce qu'est un système ANPR. Ensuite, nous détaillerons les composants principaux d'un système ANPR, y compris les caméras, les dispositifs de traitement d'images, et les logiciels spécialisés.

La section suivante portera sur le fonctionnement général d'un système ANPR. Nous décrirons le flux de travail typique, depuis la capture de l'image jusqu'à l'identification finale de la plaque d'immatriculation. Cette description inclura les différentes phases de traitement d'images et de reconnaissance de caractères.

Les applications et cas d'utilisation des systèmes ANPR seront également examinés en profondeur. Nous illustrerons comment ces systèmes sont utilisés dans des contextes variés, tels que le contrôle d'accès, la gestion du stationnement, l'application de la loi, et l'analyse du trafic.

Ensuite, nous aborderons les défis rencontrés par les systèmes ANPR. Nous discuterons des problèmes liés à la variation des conditions d'éclairage, des angles de vue, des vitesses des véhicules, ainsi que des différences de format et de style des plaques d'immatriculation.

Enfin, nous conclurons ce chapitre par une analyse des techniques et des algorithmes les plus utilisés pour chaque phase clé d'un système ANPR : la détection, la segmentation, et la reconnaissance. Cette section mettra en lumière les approches traditionnelles ainsi que les avancées récentes dans le domaine, en soulignant les avantages et les limites de chaque technique.

En fournissant une compréhension complète et structurée des systèmes ANPR, ce chapitre prépare le terrain pour les discussions plus techniques et spécifiques des chapitres suivants.

2 Définition d'un ANPR

L'ANPR est une technologie qui lit et interprète les plaques d'immatriculation des véhicules en utilisant la reconnaissance optique de caractères (OCR) et des techniques d'analyse d'images ou de vidéos pour lire et capturer les informations des plaques d'immatriculation des véhicules. Les systèmes ANPR sont couramment utilisés dans divers domaines tels que l'application de la loi, la collecte de péages, la gestion du stationnement et la surveillance du trafic (25; 26) .



FIGURE II-1 – Illustration d'un ANPR

(27)

3 Les composants d'un ANPR

L'ANPR combine plusieurs composants matériels et logiciels pour identifier et vérifier les plaques d'immatriculation des véhicules.



FIGURE II-2 – Les composants principales d'un ANPR
(28)

3.1 Composants matériels

1. **Caméras ANPR** : Ce sont des caméras haute résolution spécifiquement conçues pour capturer des images claires des plaques d'immatriculation, même dans des conditions de faible luminosité. Elles peuvent utiliser la technologie infrarouge pour voir la nuit.
2. **Unité de traitement** : Un ordinateur ou un autre dispositif utilisé pour traiter les images capturées par les caméras et exécuter le logiciel ANPR.

3.2 Composants logiciels

1. **Logiciel ANPR** : Ce logiciel extrait les informations des plaques d'immatriculation à partir des images capturées par les caméras. Il utilise généralement une combinaison de traitement d'images et de reconnaissance optique de caractères (OCR) pour identifier les caractères sur la plaque d'immatriculation.
2. **Base de Données** : Stocke les informations sur les plaques d'immatriculation, telles que le propriétaire du véhicule et les informations d'enregistrement. Cette base de données peut être utilisée pour vérifier si un véhicule est volé, recherché ou a des amendes de stationnement impayées.

3.3 Composants additionnels

- **Illuminateurs externes** : Des dispositifs fournissant un éclairage supplémentaire pour les caméras, ce qui peut être utile dans des conditions de faible luminosité.
- **Systèmes de déclenchement** : Des systèmes tels que les lasers ou les boucles inductives qui détectent l'approche d'un véhicule et déclenchent les caméras ANPR au moment optimal pour capturer l'image.

- **Interfaces de communication** : Permettent au système ANPR de communiquer avec d'autres systèmes, tels qu'une base de données de véhicules volés ou un système de gestion de parking.
- **Logiciels de révision** : Permettent aux opérateurs de vérifier et de valider les résultats de l'ANPR.

4 Fonctionnement des ANPRs

Les caméras ANPR surveillent constamment une zone, comme une route ou une entrée de véhicule. Les systèmes ANPR prennent des photos des véhicules, capturant non seulement les données des plaques d'immatriculation, mais aussi la date, l'heure et les coordonnées Global Positioning System (GPS) (26).

À l'intérieur de chaque caméra ANPR, il y a un petit ordinateur qui travaille en secret. Cet ordinateur traite le flux vidéo en direct de la caméra et l'envoie au logiciel ANPR. Ce logiciel, véritable cerveau de l'opération, accomplit un exploit remarquable : il convertit l'image de la plaque d'immatriculation en texte lisible, en utilisant la technologie (OCR) pour lire les plaques d'immatriculation dans presque toutes les conditions météorologiques, sans intervention humaine. La plupart des caméras ANPR collectent généralement plusieurs images (frames) et sélectionnent automatiquement les meilleures pour lire la plaque d'immatriculation.

Ces détections peuvent déclencher une alerte en temps réel pour améliorer la prise de conscience de la situation, être recherchées et analysées pour des informations d'enquête, et être utilisées pour automatiser des processus tels que l'application des lois sur la circulation ou le contrôle d'accès.

5 Application et cas d'utilisation

Les ANPR se sont des outils précieux pour améliorer l'efficacité et la sécurité dans divers domaines. Leur capacité à identifier rapidement les véhicules, à suivre les mouvements et à extraire des informations précieuses des plaques d'immatriculation en fait des outils polyvalents et puissants. Voici quelques-unes des applications et des cas d'utilisation les plus courants des systèmes ANPR (25).

Application de la loi

- Identification des véhicules volés ou recherchés : Les ANPR aident les autorités à

identifier les véhicules volés ou recherchés, suivre les suspects et détecter les véhicules impliqués dans des activités criminelles.

- Contrôle des véhicules non assurés et non immatriculés : Les systèmes ANPR détectent les véhicules sans assurance ou sans immatriculation valide.

Gestion de trafic

- Surveillance de trafic : Les ANPR assistent dans le suivi des vitesses moyennes, l'identification des infractions au code de la route et l'application des péages ou des charges de congestion.
- Régulation du flux de circulation : Ils permettent de surveiller et de réguler le flux de circulation sur les autoroutes et dans les zones urbaines très fréquentées.

Gestion du stationnement

- Systèmes de stationnement automatisés : Les ANPR permettent une entrée et une sortie sans ticket, gèrent l'occupation des parkings et détectent les infractions de stationnement.
- Facilité pour les utilisateurs : Ils offrent une expérience de stationnement plus fluide pour les employés et les clients, leur permettant de se garer rapidement et de commencer leur travail sans perte de temps.

Contrôle des frontières et sécurité

- Points de contrôle frontaliers et zones de haute sécurité : Les ANPR surveillent les mouvements de véhicules et identifient les véhicules suspects ou non autorisés.

Collecte de péages électroniques

- Systèmes de péage automatisés : Les ANPR détectent et facturent automatiquement les véhicules passant par les postes de péage sans nécessiter d'intervention manuelle.

Autres cas d'utilisation

- Contrôles de vitesse moyenne.
- Vérification des véhicules pour des contrôles techniques (MOT) valides.
- Localisation des véhicules volés.
- Enregistrement des infractions mineures au code de la route.
- Application des voies de bus.
- Gestion des parkings.
- Surveillance des stations-service.

6 Avantages d'utilisation des ANPRs

L'utilisation des ANPRs offre de nombreux avantages, contribuant à améliorer la sécurité, l'efficacité et la tranquillité publique. Voici quelques-uns des principaux avantages :

- **Réduction de la criminalité et du désordre** : Les ANPR aident à identifier rapidement les véhicules associés à des activités criminelles, ce qui permet une intervention plus rapide des forces de l'ordre et une réduction générale de la criminalité.
- **Augmentation des taux de détection et des résultats positifs en justice** : En fournissant des preuves précises et en temps réel, les ANPR augmentent les chances de résolution des affaires criminelles et d'aboutissement à des verdicts positifs dans le système judiciaire.
- **Dissuasion du terrorisme** : Les ANPR jouent un rôle crucial dans la détection et la surveillance des véhicules suspects, aidant ainsi à prévenir les actes terroristes et à renforcer la sécurité nationale.
- **Renforcement de la tranquillité publique** : La présence de systèmes ANPR contribue à rassurer le public en démontrant un engagement visible et efficace en matière de sécurité.
- **Amélioration du renseignement** : Les ANPR collectent et centralisent des données précieuses sur les mouvements de véhicules, permettant aux agences de sécurité de mieux analyser et comprendre les tendances criminelles et de mener des enquêtes plus approfondies.
- **Amélioration de la sécurité routière** : En surveillant et en détectant les infractions au code de la route, les ANPR contribuent à la réduction des accidents et à la promotion d'une conduite plus sûre.

7 Facteurs impactant la robustesse des systèmes ANPR

Les ANPRs sont soumis à divers facteurs qui peuvent influencer leur efficacité et leur fiabilité. Voici une analyse approfondie de ces facteurs(29; 30; 31; 32)

1. Conditions d'éclairage inadéquates et ombres prononcées :

Les ANPRs peuvent rencontrer des difficultés à lire les plaques d'immatriculation dans des conditions d'éclairage faible, ainsi que lorsqu'il y a de fortes ombres portées. Cela peut entraîner une baisse de la qualité des images capturées et donc une réduction de la précision de la reconnaissance.

2. Images de faible résolution et trames vidéo de basse qualité :

Les images de faible résolution et les trames vidéo de qualité médiocre peuvent rendre difficile la reconnaissance des caractères sur les plaques d'immatriculation, en particulier lorsqu'il s'agit de détails fins ou de textes flous. Cela peut entraîner des erreurs de lecture et une diminution de la fiabilité du système.

3. Distorsion et inclinaison des images :

Les images de plaques d'immatriculation peuvent être déformées ou inclinées en raison de divers facteurs, tels que l'angle de la caméra, la vitesse du véhicule ou les vibrations. Cette distorsion peut compliquer la tâche de reconnaissance et conduire à des résultats erronés.

4. Paramètres de la caméra, distance, météo et plaques Sales :

Les réglages incorrects de la caméra, la distance entre la caméra et le véhicule, les conditions météorologiques défavorables (pluie, neige, brouillard) et les plaques d'immatriculation sales ou endommagées peuvent tous affecter la qualité des images capturées et la performance de l'ANPR.

5. Contraste insuffisant, géométries et langues différentes, et facteurs combinés :

Un contraste insuffisant entre les caractères et l'arrière-plan de la plaque d'immatriculation, la présence de différentes géométries de plaques d'immatriculation et l'utilisation de langues différentes peuvent rendre la reconnaissance plus difficile. De plus, la combinaison de plusieurs de ces facteurs peut entraîner une baisse significative de la robustesse de l'ANPR.

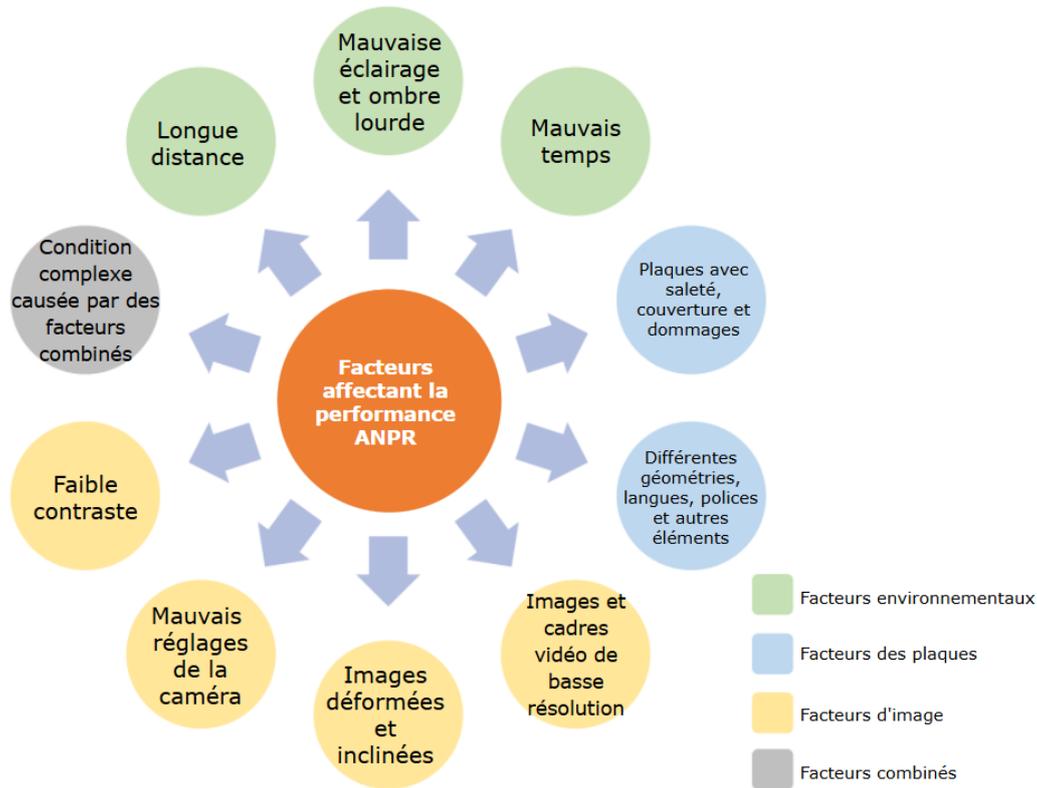


FIGURE II-3 – Les facteurs Impactant la Robustesse des ANPR

8 Techniques et méthodes utilisés pour les systèmes ANPR

Un système ANPR typique fonctionne en trois étapes cruciales : la détection, la segmentation et la reconnaissance. Chacune de ces phases joue un rôle fondamental dans le succès global du système.

La détection consiste à localiser la plaque d'immatriculation dans une image, ce qui nécessite des techniques robustes pour s'adapter à diverses conditions environnementales et angles de vue. Ensuite, la segmentation vise à isoler les caractères individuels de la plaque, un processus délicat qui doit gérer les variations de police, de taille et de couleur des caractères. Enfin, la reconnaissance des caractères implique l'interprétation des caractères segmentés pour en extraire l'information alphanumérique, nécessitant des algorithmes performants et précis.

Dans les sections qui suivent, nous allons examiner en détail les méthodes les plus utilisées pour chacune de ces phases, en offrant un aperçu des techniques actuelles et de leurs

applications dans le domaine de l'ANPR.

8.1 Détection de la plaque d'immatriculation

La détection des plaques d'immatriculation constitue la première étape fondamentale dans un système de reconnaissance automatique des plaques d'immatriculation (ANPR). Cette phase consiste à localiser avec précision les plaques d'immatriculation dans les images capturées, que ce soit à partir de vidéos de surveillance, de caméras embarquées ou de dispositifs fixes. Une détection efficace est cruciale, car elle conditionne la réussite des phases suivantes de segmentation et de reconnaissance. Les méthodes de détection doivent être robustes et capables de gérer diverses conditions environnementales, telles que les variations d'éclairage, les angles de vue multiples, et la présence d'objets obstructifs. Dans cette sous-section, nous allons explorer les techniques les plus couramment utilisées pour la détection des plaques d'immatriculation.

8.1.1 Méthode basée sur YOLO

V. Gnanaprakash (33) et al. ont proposé une méthode pour la reconnaissance des véhicules et des plaques d'immatriculation. Dans cette approche, les véhicules et leurs plaques d'immatriculation sont d'abord détectés et extraits de leurs images d'origine en utilisant YOLO.

Le jeu de données se compose de 1000 images de plaques d'immatriculation, divisé en deux parties : entraînement et test. Pour l'entraînement, 800 images sont prises et annotées.

Les résultats obtenus pour la détection des plaques d'immatriculation montrent que YOLO est très performant, atteignant une précision de 98,5%.

Diogo M. et F. Izidio (34) ont proposé une méthode de détection de plaques d'immatriculation basée sur Tiny YOLOv3, une version réduite de YOLOv3, adaptée aux applications mobiles.

Le processus de détection commence par le redimensionnement de l'image à une résolution de 416×416 pixels, puis par la division de l'image en deux grilles de sortie de 13×13 et 26×26 , chacune responsable de la détection d'objets à différentes échelles à l'aide de boîtes d'ancrage prédéfinies pour chaque élément de la grille. L'utilisation de deux grilles différentes permet au système de détection de reconnaître des plaques d'immatriculation à des échelles variées sans nécessiter de redimensionnements et de lissages répétés des images. Ils ont utilisé les configurations et valeurs par défaut des boîtes d'ancrage, où la grille de 13×13 utilise les trois plus grandes boîtes d'ancrage et la grille de 26×26

utilise les trois plus petites.

Après l'entraînement, 150 images supplémentaires ont été prises pour les tests, en suivant les mêmes caractéristiques que le jeu de données précédent, permettant ainsi d'évaluer le taux de détection de leur réseau entraîné.

Pour éviter les cas où la plaque d'immatriculation n'est pas entièrement à l'intérieur des limites de la boîte de délimitation, la largeur et la hauteur de la boîte prédite par sont augmentées de 10%.

Dans (35), **Aissa Amrouche** et al. ont employé YOLO-v3 pour détecter les plaques d'immatriculation des véhicules en temps réel. YOLO-v3 repose sur une variante de Darknet, initialement entraînée sur ImageNet avec un réseau de 53 couches. L'architecture de base de YOLO-v3 comprend un total de 106 couches, avec 53 couches supplémentaires dédiées à la détection. Grâce à l'utilisation de cartes de caractéristiques de trois tailles distinctes et à trois emplacements différents dans le réseau, YOLO-v3 effectue des détections en utilisant des noyaux de détection de 1×1 . Le réseau s'occupe de la détection des boîtes englobantes.

Dans (36), **Rezki Zibani** et al. ont utilisé YOLOV4 pour localiser les plaques d'immatriculation en vue de les extraire. Le processus de localisation se déroule comme suit :

a. Entraînement du modèle YOLOv4 : Pour cette tâche, il est nécessaire d'annoter les images et de générer des fichiers d'entraînement. Ces fichiers contiennent les coordonnées des boîtes englobantes et la classe correspondante (dans ce cas, une seule classe, celle des plaques d'immatriculation).

b. Détection des régions d'intérêt : Après l'entraînement du modèle, celui-ci est capable d'analyser de nouvelles images de véhicules pour identifier les régions où une plaque d'immatriculation est susceptible d'être présente.

c. Propagation des prédictions : Le modèle effectue des prédictions pour chaque région d'intérêt, en attribuant des probabilités à chaque classe d'objets possible, y compris la classe spécifique des plaques d'immatriculation.

d. Suppression des non-maxima : Pour éviter la redondance, YOLOv4 utilise une technique appelée suppression des non-maxima. Cette étape élimine les boîtes englobantes dupliquées ou fortement superposées, ne conservant que les prédictions les plus fiables pour les plaques d'immatriculation.

e. Localisation de la plaque d'immatriculation : Le modèle utilise la boîte englobante avec la probabilité la plus élevée pour localiser précisément la plaque d'immatriculation dans

l'image.

Dans leur travail, **Mouad Bensouilah et al.** (37) ont exploité le détecteur d'objets YOLO, une architecture CNN largement reconnue pour sa capacité à détecter des objets en temps réel tout en préservant un équilibre entre vitesse et précision. Ils ont choisi YOLOv3, une version améliorée de YOLO, qui permet de prédire les boîtes englobantes à trois échelles différentes en utilisant une approche similaire à celle des réseaux pyramidaux. En parallèle, ils ont comparé l'extracteur de caractéristiques Darknet-53, spécifique à YOLOv3, avec MobileNets, une alternative connue pour sa rapidité et ses performances.

8.1.2 Méthode basée sur l'architecture SegNet

Dans leur étude (38), **Naaman Omar** et al. ont utilisé l'architecture SegNet pour détecter les régions des plaques d'immatriculation. SegNet est une méthode de segmentation sémantique d'images qui prédit l'étiquette catégorielle de chaque pixel dans une image donnée. Cette méthode de détection atteint une précision de 94,43%, ce qui la place dans la catégorie des performances très bonnes, bien que ce ne soit pas la meilleure.

8.1.3 Méthode basée sur CNN

Dans leur étude (39), **Zied Selmi** et al. ont présenté une méthode de détection de plaque d'immatriculation qui repose sur un prétraitement et une classification CNN. Avant de procéder à la classification, plusieurs opérations sont effectuées sur l'image d'origine. Nous décrivons ci-dessous ces étapes :

a. Conversion de l'image RGB en HSV : Leur système commence par l'introduction d'une image au format RGB, qui est ensuite convertie en espace colorimétrique HSV. Dans cet espace, la teinte (H) représente le type de couleur et la saturation (S) indique l'intensité de la couleur, avec une plage de valeurs de 0 à 255. La valeur (V) correspond à la luminosité de la couleur, également avec une plage de valeurs de 0 à 255, où 0 représente une couleur sombre et 255 une couleur claire.

b. Filtrage morphologique pour maximiser le contraste : Pour maximiser le contraste de l'image et extraire les petits éléments et les détails, des opérations morphologiques telles que la transformation en chapeau haut sont utilisées. Il y a deux types de transformations : le chapeau haut blanc et le chapeau haut noir. Pour obtenir une image avec un contraste maximal, l'image en niveaux de gris est ajoutée au résultat de la transformation en chapeau haut blanc, puis soustraite de l'image de la transformation en chapeau haut noir.

c. Filtre de flou gaussien : Après avoir maximisé le contraste d'une image en niveaux de gris et amélioré les résultats du traitement ultérieur de l'image, ils ont appliqué un filtre de flou gaussien pour éliminer les détails et le bruit de cette image à l'aide d'un noyau 5×5 afin d'améliorer les résultats du traitement ultérieur de l'image.

d. Seuil adaptatif : Étant donné que les images sont prises dans différentes circonstances, conditions d'éclairage et zones, ils ont appliqué un seuillage adaptatif pour éliminer les régions non importantes dans l'image. leur algorithme calcule deux valeurs : la valeur de seuil de la zone de voisinage dont la taille est de 19 et la somme pondérée des valeurs de voisinage où les poids sont une fenêtre gaussienne de taille 9.

e. Recherche de tous les contours : Une fois les régions non pertinentes de l'image éliminées, l'approche adoptée consiste à identifier et à définir les contours qui délimitent la forme de l'objet d'intérêt. Étant donné que la plaque d'immatriculation est rectangulaire, l'attention se porte uniquement sur les 4 points nécessaires pour définir son contour. Cela est réalisé en utilisant une méthode d'approximation de contour simple afin de simplifier le processus de détection des contours.

f. Filtrage géométrique : Après avoir mis en œuvre les étapes précédentes, il est nécessaire que le système respecte un ensemble de contraintes concernant la taille et la forme de la boîte de délimitation pour extraire une Région d'intérêt dans une image RGB. Par conséquent, ils appliquent des règles de filtrage géométrique pour améliorer la précision de la détection de la plaque d'immatriculation en rejetant les objets très grands et très petits. Ainsi, les ratios d'aspect très grands et très petits sont rejetés. Les règles de filtre utilisées concernent la largeur, la hauteur et le ratio.

g. Détection de plaque d'immatriculation avec CNN : À cette étape, ils extraient les différentes boîtes de délimitation possibles qui peuvent être considérées comme des plaques d'immatriculation.

h. Boîte de délimitation de la plaque d'immatriculation : Le résultat de l'étape finale du processus de détection sera donné par leur classifieur si la prédiction est positive et supérieure à 0,7, qui est le seuil minimum du score obtenu par le classifieur. En fait, cette boîte de délimitation est considérée comme une plaque d'immatriculation positive, et par conséquent, une boîte de délimitation sera dessinée sur l'image qui montre la plaque d'immatriculation. Cependant, si elle est inférieure à 0,7, elle sera considérée comme une non-plaque d'immatriculation.

8.1.4 Méthode basée sur un modèle d'attention visuelle

Dans (40) **Di Zang** et **Zhenliang Chai** et all. ont proposé un modèle d'attention visuelle modifié, inspiré du modèle traditionnel. Ce modèle utilise des caractéristiques de couleur, d'intensité et d'orientation pour construire des pyramides gaussiennes, sur lesquelles des opérations de centre-périphérie sont appliquées pour générer des cartes de caractéristiques correspondantes. Ces cartes sont ensuite fusionnées pour former une carte de saillance, indiquant les régions d'intérêt potentielles pour la plaque d'immatriculation. Ils utilisent une méthode de seuillage automatique d'Ostu pour adapter les seuils de la carte de saillance. Enfin, en se basant sur cette carte de saillance ajustée, ils génèrent un masque binaire pour séparer la plaque d'immatriculation de l'image de la voiture initiale.

8.1.5 Méthode basée sur l'analyse des composants connectés

Dans son article (41), le professeur **Joy Iong Zong Chen** a développé une méthode basée sur l'analyse des composants connectés. Cette méthode prend en compte les règles spécifiques des plaques d'immatriculation ainsi que les effets d'éclairage sur celles-ci, ce qui peut entraîner des variations dans la localisation des composants dans l'image. Étant donné que l'utilisation d'un modèle binaire seul ne permet pas d'obtenir les résultats souhaités, des techniques binaires locales sont employées. Ces techniques sont appliquées à des blocs de dimensions $m \times n$ dans l'image, chaque bloc étant analysé séparément avec un modèle binaire. Dans cette approche, deux méthodologies binaires sont utilisées pour traiter les sous-blocs. La méthode d'Otsu est appliquée pour traiter les variations importantes dues à l'éclairage. Par ailleurs, pour compenser les effets d'un éclairage irrégulier, une méthode améliorée de suppression de l'ombre est mise en œuvre. Il est important de noter qu'un bloc typique est défini par un point (x, y) représentant une valeur de gris dans l'image.

8.2 Segmentation des caractères

8.2.1 Méthode de détection des contours

V. Gnanaprakash (33) et all. ont proposé une méthode de segmentation de la matricule basée sur la détection des contours de canny, d'abord les images des plaques d'immatriculation sont converties en niveaux de gris. Ensuite, un filtre bilatéral est appliqué sur l'image en niveaux de gris. Chaque pixel est remplacé par les valeurs d'intensité pondérées des pixels voisins. Plus précisément, il préserve les bords tout en éliminant les bruits dans l'image. Ensuite, la détection de contours de Canny est appliquée. Le processus de détection de contours comprend cinq étapes :

1. Un filtre gaussien est appliqué pour éliminer les bruits dans l'image.
2. Une suppression des non-maxima est appliquée pour éliminer les réponses parasites de la détection de bords.
3. Un double seuillage est appliqué pour déterminer les bords potentiels.
4. Supprimer les autres bords qui ne sont pas connectés aux bords forts par histérèse.

Après la détection des bords, les contours sont tracés. Les caractères segmentés extraits sont envoyés en entrée pour la reconnaissance des caractères.

Dans (39), **Zied Selmi** et all. ont segmenté les caractères en utilisant la détection des contours de Canny. Avant cela, la plaque d'immatriculation doit être extraite de l'image du véhicule. Voici les étapes de cette méthode :

1. Conversion en niveaux de gris :

Convertir l'image de la plaque d'immatriculation de l'espace colorimétrique RGB à une image en niveaux de gris.

2. Maximisation du contraste :

Améliorer et clarifier l'image en maximisant le contraste.

3. Détection des contours avec la technique de Canny :

Appliquer la technique de détection de contours de Canny de manière automatique :
Calculer les intensités moyennes des pixels dans l'image.

- Utiliser la valeur moyenne pour établir des seuils inférieurs et supérieurs.
- Contrôler ces seuils en fonction de l'argument sigma (0.20 dans ce cas).
- Appliquer le détecteur de contours de Canny avec ces seuils.

4. Extraction des contours :

Utiliser une technique d'extraction de contours de manière hiérarchique pour identifier les contours pertinents dans l'image.

5. Opérations géométriques :

Effectuer quelques opérations géométriques pour minimiser une boîte de délimitation afin d'améliorer la précision de la segmentation.

Dans leur article (36), **Rezki Zibani** et all. ont proposé une méthode de segmentation basée sur la détection des contours, comprenant deux étapes principales : le prétraitement de l'image et la séparation des caractères. Voici les étapes détaillées :

1. Prétraitement de l'image

- Mise à niveau des niveaux de gris
- Filtrage du bruit
- Binarisation de l'image à l'aide de la méthode d'Otsu
- Dilatation de l'image

2. Détection des contours

- Détection des contours : Utilisation d'un algorithme de détection des contours (ex : Canny)
- Élimination des contours indésirables : Suppression des faux contours et petites imperfections en utilisant le rapport hauteur/largeur des bords
- Extraction des caractères : Extraction des contours individuels pour chaque caractère en sous-images

8.2.2 Méthode de segmentation des caractères basée sur le seuillage et les opérations morphologiques

Dans leur article (38), **Naaman Omar** et al. ont développé une méthode de segmentation des caractères qui intègre des techniques de traitement d'image traditionnelles ainsi que des opérations morphologiques. Voici les principales étapes de cette méthode :

1. **Conversion en niveaux de gris** : La zone de la plaque est transformée en une image en niveaux de gris.
2. **Filtrage gaussien** : Un filtre gaussien est appliqué pour éliminer le bruit présent dans l'image.
3. **Soustraction d'image** : L'image filtrée est soustraite de l'image en niveaux de gris pour améliorer le contraste du premier plan.
4. **Seuillage** : Les caractères, constituant le premier plan, sont extraits en utilisant la méthode de seuillage d'Otsu.
5. **Opérations morphologiques** : Des opérations morphologiques telles que l'érosion et la dilatation sont réalisées pour éliminer les objets binaires indésirables.
6. **Détection des caractères** : Des boîtes englobantes sont utilisées pour détecter et localiser chaque numéro de la plaque dans l'image binaire résultante.

8.2.3 Méthode de segmentation basée sur la projection

Dans (40), l'auteur **Di Zang** et all. ont utilisé la projection pour segmenter les caractères en sept blocs (ce cas s'applique aux plaques d'immatriculation chinoises). Voici les étapes

de cette méthode :

1. **Transformation en image binaire :**

Transformer l'image de la plaque d'immatriculation en une image binaire où les caractères sont blancs et l'arrière-plan est noir.

2. **Projection des pixels blancs :**

Projeter les pixels blancs à la fois horizontalement et verticalement.

3. **Détermination des limites horizontales :**

Utiliser la projection horizontale pour trouver les limites supérieure et inférieure des caractères afin de calculer la hauteur approximative.

4. **Segmentation des régions verticales :**

Vérifier les points zéro de la courbe de projection verticale pour identifier les régions à segmenter.

5. **Calcul de la largeur moyenne des segments :**

- Selon la configuration de la caméra, sélectionner les régions dont la largeur de pixel est comprise entre 15 et 25.
- Calculer une largeur moyenne pour découper la plaque d'immatriculation en sept segments.

6. **Segmentation finale :**

- Découper la plaque d'immatriculation en sept segments basés sur la largeur moyenne calculée.
- Le segment le plus à gauche contient le caractère chinois.

8.2.4 Segmentation utilisant le clustering k-means optimisé.

Dans son article (41), le professeur **Joy Iong Zong Chen** a employé une méthode basée sur le clustering k-means optimisé.

L'analyse neutrosophique, utilisant les concepts de vérité, d'indétermination et de fausseté, permet de déterminer l'incertitude des données d'un pixel. Ensuite, la méthode de clustering K-means est appliquée pour regrouper les objets en clusters distincts, facilitant ainsi la segmentation et la reconnaissance des caractères.

8.2.5 Méthode basée sur YOLO

Dans leur article (36), **Rezki Zibani** et all. ont proposé une autre méthode de segmentation des caractères basée sur l'intelligence artificielle, en utilisant le détecteur d'objets

YOLOv4. Le prétraitement de l'image reste la première étape. Voici les étapes de cette méthode :

1. **Prétraitement de l'image**

2. **Détecteur d'objets YOLOv4**

- Annotation des images
- Entraînement de YOLOv4 sur 10 classes (caractères de 0 à 9)
- Segmentation : Utilisation de YOLOv4 pour segmenter les caractères avec des paramètres spécifiques (mini-lots de 64 images, taille d'image de $128 \times 64 \times 1$, taux d'apprentissage de 0,001)

8.3 Reconnaissance des caractères

8.3.1 Méthode basée sur OCR

Dans leur étude (33), l'auteur **V. Gnanaprakash** et all. ont utilisé la technologie OCR pour la reconnaissance des caractères. L'OCR est un processus de conversion d'images de texte manuscrit ou imprimé en texte pouvant être lu par une machine. Il existe plusieurs moteurs OCR disponibles, et ce système spécifique utilise Tesseract, l'un de ces moteurs OCR. Les caractères segmentés sont ensuite fournis à l'OCR en tant qu'entrée, permettant à l'OCR de reconnaître ces caractères. Les données extraites sont ensuite stockées dans un fichier de données.

8.3.2 Reconnaissance des caractères avec CNN

Dans leur étude sur la reconnaissance de caractères (38), **Naaman Omar** et all. ont introduit une méthode qui repose sur l'utilisation d'un réseau de neurones convolutif (CNN). Ce réseau est composé de plusieurs couches, chacune ayant un rôle spécifique dans le processus de reconnaissance. Voici des explications de chaque couche :

- **Couche de convolution** : Cette couche est responsable d'extraire les caractéristiques importantes des images en appliquant des filtres pour détecter des motifs spécifiques tels que des bords ou des textures.
- **Couche de regroupement (Pooling)** : La fonction de cette couche est de réduire la dimensionnalité des caractéristiques extraites par la couche de convolution en sélectionnant les valeurs les plus importantes dans des régions spécifiques de la carte de caractéristiques.

- **Couches entièrement connectées** : Ces couches prennent les caractéristiques extraites par les couches précédentes et les combinent pour effectuer une classification. Chaque neurone dans ces couches est connecté à chaque neurone de la couche précédente, permettant ainsi au réseau de prendre des décisions globales basées sur les caractéristiques locales.
- **Activation** : Les fonctions d'activation, telles que ReLU ou sigmoïde, sont appliquées pour introduire de la non-linéarité dans le réseau, aidant ainsi le modèle à apprendre des relations complexes entre les caractéristiques d'entrée et les étiquettes de sortie.

Dans l'article (34), **Diogo M** et all. ont proposé une méthode de reconnaissance de caractères utilisant un réseau de neurones convolutifs (CNN). Ce CNN de reconnaissance reçoit des images de plaques d'immatriculation découpées et redimensionnées, puis produit des prédictions de caractères.

Dans (39), l'auteur **Zied Selmi** et all. ont utilisé des (CNNs) pour la reconnaissance des caractères. La dernière étape de leur système consiste à identifier les caractères des plaques d'immatriculation en utilisant un modèle CNN basé sur Tensorflow.

8.3.3 Méthode basée sur les SVMs et les CNNs

Dans leur étude (40), **D. Zang et al.** ont utilisé une méthode qui combine les deux techniques :

1. Extraction de caractéristiques avec CNN :

- Architecture du CNN : Le modèle CNN comporte plusieurs couches convolutionnelles et de sous-échantillonnage, permettant l'extraction hiérarchique des caractéristiques à partir des images d'entrée. Les images sont redimensionnées et traitées à travers ces couches pour extraire des cartes de caractéristiques.
- Convolution et sous-échantillonnage : Le modèle utilise différentes tailles de noyaux de convolution (7x7, 5x5, 3x3) et applique le sous-échantillonnage pour réduire la sensibilité aux distorsions et aux transformations affines. Chaque couche convolutionnelle génère des cartes de caractéristiques qui sont ensuite réduites en taille par sous-échantillonnage.
- Partage de poids : Les poids des connexions sont partagés au sein des cartes de caractéristiques, ce qui permet au CNN d'être invariant aux changements de décalage et de rotation.

2. Classification avec SVM :

- SVM Multiclasses : Chaque CNN extrait des caractéristiques qui sont ensuite passées à des SVM pour la classification finale. Les SVMs utilisent les caractéristiques extraites pour trouver un hyperplan qui sépare les différentes classes dans l'espace des caractéristiques.
- Vote majoritaire : Pour la reconnaissance des caractères chinois, les probabilités de classification générées par trois SVM sont combinées par un processus de vote majoritaire pour obtenir le résultat final.

9 Récapitulation des travaux antérieure

Dans cette section, nous présentons un tableau récapitulatif des différentes méthodes utilise dans la réalisation des systèmes ANPR.

Source	Détection	Segmentation	Reconnaissance
(37)	YOLO	\	RNN + GRU units
(10)	YOLO	\	CR-NET
(34)	Tiny YOLOv3	\	CNN
(42)	Edge detection	Regionprops	Template Matching
(41)	K-Means clustering	K-Means clustering	CNN
(43)	Analysis of Connected Components (CCA)	optimal K-means clustering	CNN
(38)	Deep semantic segmentation network	Encoder-decoder network	CNN
(44)	Edge detection	CNN	Random Forest
(39)	CNN	Edge detection	CNN
(45)	Mask-RCNN	Mask-RCNN + geometric filtering	Mask-RCNN
(46)	Contours detection	Contours detection	CNN
(36)	YOLO	YOLO + Edge detection	Data fusion

TABLE II.1 – Comparaison des approches de détection, segmentation et reconnaissance des plaques d'immatriculation dans divers travaux de recherche

10 Conclusion

Dans ce chapitre, nous avons présenté une revue des systèmes ANPR, en abordant leurs composants, leur fonctionnement, leurs applications, ainsi que les défis auxquels ils font

face et les techniques couramment employées à chaque étape du processus. En ayant une vision claire de l'évolution et des capacités actuelles des systèmes ANPR, nous sommes maintenant prêts à aborder le cœur de notre travail : la méthodologie de développement de notre propre système ANPR.

Dans le prochain chapitre, nous détaillerons notre approche méthodologique pour concevoir et développer un système ANPR performant. Nous présenterons les étapes clés de notre démarche, les choix technologiques que nous avons faits, les algorithmes que nous avons implémentés, et les défis spécifiques que nous avons surmontés. Ce processus méthodologique sera illustré par des exemples concrets et des résultats expérimentaux, offrant ainsi une compréhension approfondie et pratique de la création d'un système ANPR.

Chapitre III

DZANPR : Système de reconnaissance des plaques d'immatriculation algériennes

1 Introduction

À la lumière des recherches et des analyses présentées dans le chapitre précédent, où nous avons examiné les différentes approches et technologies employées dans les ANPR, nous nous concentrerons désormais sur la conception et le développement de notre propre système baptisé DZANPR. Ce système sera spécifiquement adapté aux plaques d'immatriculation algériennes, répondant ainsi à un besoin précis et très demandé dans le contexte local.

La spécificité des plaques d'immatriculation algériennes, tant dans leur format que dans leurs caractéristiques distinctives, nécessite une approche méthodologique bien définie et ciblée. L'objectif est d'optimiser notre système pour atteindre une précision et une efficacité maximales, en tenant compte des particularités locales. Cela nous permettra également de répondre à des besoins concrets en matière de gestion et de contrôle des véhicules dans différents contextes, tels que la sécurité routière et la gestion du stationnement.

Ce chapitre est structuré de manière à présenter de manière détaillée notre méthode. Nous commencerons par décrire les besoins fonctionnels et les caractéristiques spécifiques des plaques d'immatriculation algériennes. Ensuite, nous offrirons une vue d'ensemble de notre système, en détaillant l'architecture proposée et en justifiant les choix techno-

logiques effectués, notamment l'utilisation de YOLO et des (CNNs). Par la suite, nous aborderons le développement du système en trois étapes principales : Détection et extraction des plaques, la segmentation des caractères et la reconnaissance des caractères.

2 Objectifs du projet

L'objectif principal de notre projet est de concevoir et développer un ANPR adapté aux plaques algériennes. Ce système doit être capable de détecter et de reconnaître les plaques avant et arrière de tous types de véhicules avec une précision élevée et un temps d'exécution minimal.

Voici une description des fonctionnalités clés attendues du système :

- **Reconnaissance des plaques d'immatriculation algériennes**
- **Traitement d'images de différentes tailles et formats** : Le système doit être capable de traiter des images quel que soit leur taille ou format.
- **Traitement des vidéos en temps réel** : Le système doit pouvoir analyser des vidéos en direct provenant des caméras de sécurité.
- **Portabilité et intégration** : Le système doit être portable et facile à intégrer avec d'autres systèmes (réservation, parking, péage, etc.).
- **fonctionnement dans des conditions difficiles** : Le système doit être résistant aux facteurs pouvant affecter la reconnaissance, tels que l'angle de la caméra, les plaques sales, et les conditions environnementales.
- **Flexibilité et adaptabilité** : Le système doit être conçu de manière modulaire pour faciliter les améliorations futures et l'ajout de nouvelles fonctionnalités.

3 Caractéristiques des plaques d'immatriculation algériennes

Les plaques d'immatriculation algériennes se distinguent par un format spécifique et des règles strictes. Elles sont conçues pour être facilement identifiables et uniformes, avec des distinctions claires entre les plaques avant et arrière des véhicules.

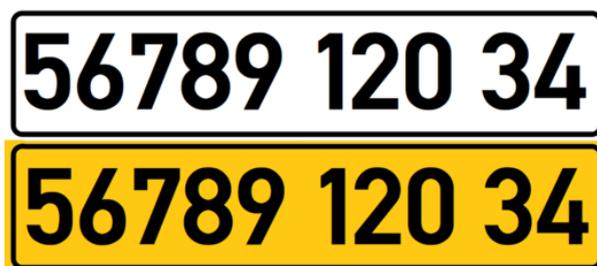


FIGURE III-1 – Exemple des plaques d'immatriculation algériennes

3.1 Structure et format des plaques

Les plaques standard algériennes sont composées de dix chiffres répartis en trois groupes, séparés par des espaces. Elles sont disponibles en deux couleurs :

- Blanc avec des chiffres noirs pour les plaques avant.
- Jaune avec des chiffres noirs pour les plaques arrière.

3.2 Numérotation des plaques

La plaque se compose de trois groupes de chiffres :

- Le premier groupe de cinq à six chiffres représente le numéro d'enregistrement du véhicule, y compris les zéros en tête.
- Le deuxième groupe de trois chiffres comprend :
 - Le premier chiffre indiquant la classe du véhicule illustre dans le Tableaux suivant.
 - Les deux chiffres suivants indiquent l'année de fabrication du véhicule.
- Le troisième groupe de deux chiffres identifie la wilaya (province) où le véhicule a été enregistré pour la première fois [01- 59].

3.3 Classes des véhicules

Chaque type de véhicule est identifié par le premier chiffre du deuxième groupe de trois chiffres, selon la classification présenté par le tableau suivant :

Classe	Signification
1	Véhicules privés
2	Camions
3	Vans (camionnettes)
4	Bus
5	Tracteurs routiers
6	Autres tracteurs
7	Véhicules spéciaux
8	Remorques
9	Motocycles

TABLE III.1 – Classes des véhicules (2)

3.4 Identifiant des wilayas

Les deux derniers chiffres de la plaque indiquent la wilaya d'enregistrement. Les numéros vont de 01 à 58, représentant les différentes wilayas d'Algérie. À partir de janvier 2022, les nouvelles wilayas créées en décembre 2019 ont leurs propres codes spécifiques.

3.5 Spécifications des plaques

Type de plaque	Longueur (mm)	Hauteur (mm)	Hauteur des chiffres (mm)	Largeur des chiffres (mm)	Espacement des chiffres (mm)
Plaques à une ligne	455 à 520	100 à 110	75	30 (20 pour le chiffre 1)	10
Plaques à deux lignes	275	200	75	30 (20 pour le chiffre 1)	10
Plaques pour motos	140	120	45	26 (15 pour le chiffre 1)	-

TABLE III.2 – Spécifications des plaques (2)

3.6 Plaques spéciales

Plaques diplomatiques Fond teal clair avec des chiffres noirs. Trois groupes de chiffres séparés par des tirets.

- Le premier groupe représente le numéro d'enregistrement du véhicule.
- Le deuxième groupe (deux chiffres) indique le statut diplomatique.
- Le troisième groupe (deux chiffres) identifie l'ambassade.

4 Vue générale sur le système DZANPR

Le système proposé constitue la partie logicielle d'un ANPR qui s'exécute sur une unité de traitement dédiée. Il reçoit en entrée une image, qui peut provenir directement d'une interface graphique, de trames vidéo, ou d'un flux de caméra de surveillance. Cette image peut contenir une ou plusieurs plaques d'immatriculation, voire aucune.

Le système commence par identifier les plaques d'immatriculation présentes dans l'image en utilisant des techniques de deep learning. Une fois les plaques détectées, elles sont extraites pour être reconnues. Enfin, le système retourne les plaques sous forme de texte, permettant ainsi leur utilisation dans divers contextes tels que la gestion du stationnement, le contrôle de sécurité ou la gestion du trafic.

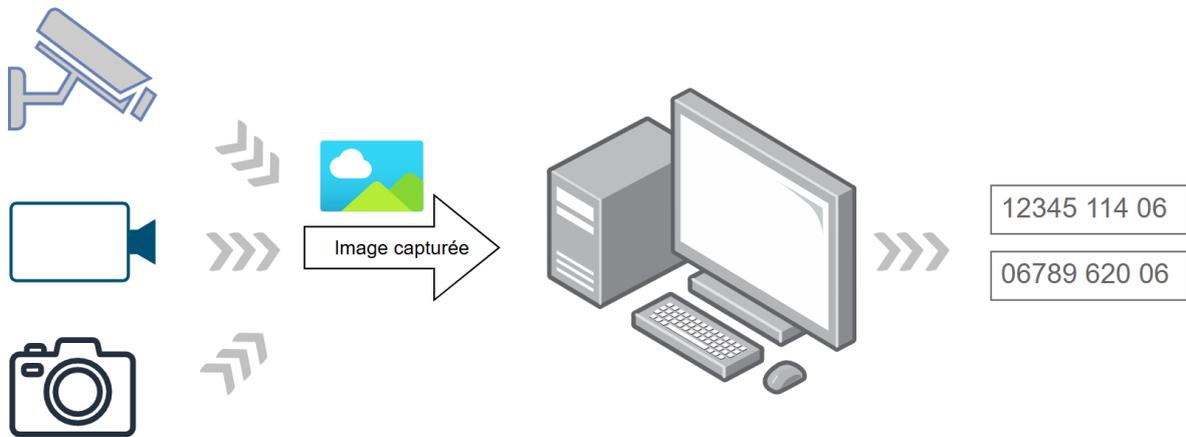


FIGURE III-2 – Schéma général du système

5 Architecture proposée

L'architecture de DZANPR est structurée en trois phases principales, chacune jouant un rôle crucial dans le traitement et l'analyse des données des plaques.

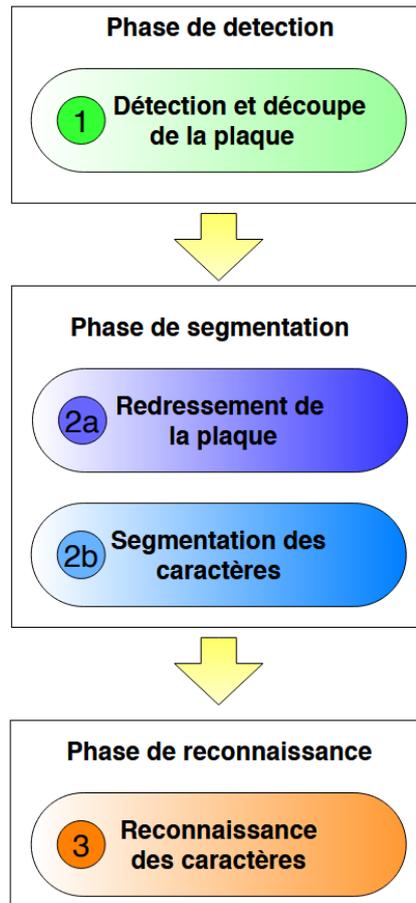


FIGURE III-3 – Architecture de DZANPR

6 Phase de détection

Notre système commence par détecter les plaques d'immatriculation dans les images d'entrée en utilisant YOLO , un réseau de neurones convolutifs avancé. YOLO divise l'image en régions et prédit les boîtes englobantes avec des probabilités associées, permettant une identification rapide et précise des plaques. Ce modèle, entraîné sur un ensemble de données algérien comprenant diverses conditions et types de véhicules.

7 Phase de segmentation

Pour la segmentation, nous avons expérimenté plusieurs méthodes traditionnelles telles que la détection des contours, l'analyse des objets connectés et les profils de projection. Cependant, aucune de ces méthodes n'a bien généralisé sur les différents cas de plaques détectées. Cela nous a conduit à envisager l'utilisation du deep learning pour remédier à ces problèmes.

Nous avons divisé cette tâche en deux étapes pour maximiser la performance du système. La première étape consiste à redresser la plaque détectée en identifiant les quatre coins et en appliquant une *transformation de perspective*. Cette étape ne corrige pas seulement l'orientation de la plaque, mais elle élimine également le fond inutile pour l'étape suivante.

La deuxième étape est la segmentation des caractères. Pour cela, nous détectons les boîtes englobantes des caractères à l'aide de YOLO, qui s'est avéré efficace pour la détection de multiples objets avec une grande précision et rapidité. En combinant ces deux approches, nous optimisons la performance et la précision de la segmentation des plaques d'immatriculation.

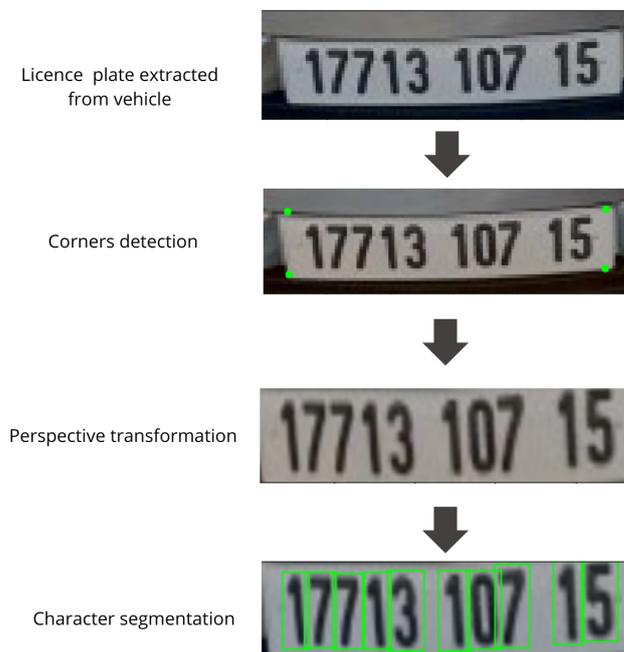


FIGURE III-4 – Processus de segmentation des des caractères

8 Phase de reconnaissance

Pour la reconnaissance des caractères, nous avons opté pour un CNN en raison de sa capacité à extraire des caractéristiques complexes à partir d'images. Les CNN sont particulièrement bien adaptés à la classification d'images, car ils peuvent apprendre des représentations hiérarchiques à partir des données d'entrée, ce qui les rend efficaces pour la reconnaissance de motifs visuels.

Notre modèle CNN est conçu pour classifier les images de caractères individuels, en les associant à des étiquettes de classe correspondant aux chiffres de 0 à 9. Le modèle passe par plusieurs couches de convolution et de pooling pour extraire des caractéristiques

discriminantes des images, puis par des couches pleinement connectées pour effectuer la classification finale.

La figure suivante illustre l'architecture générale du modèle CNN pour la reconnaissance des caractères :

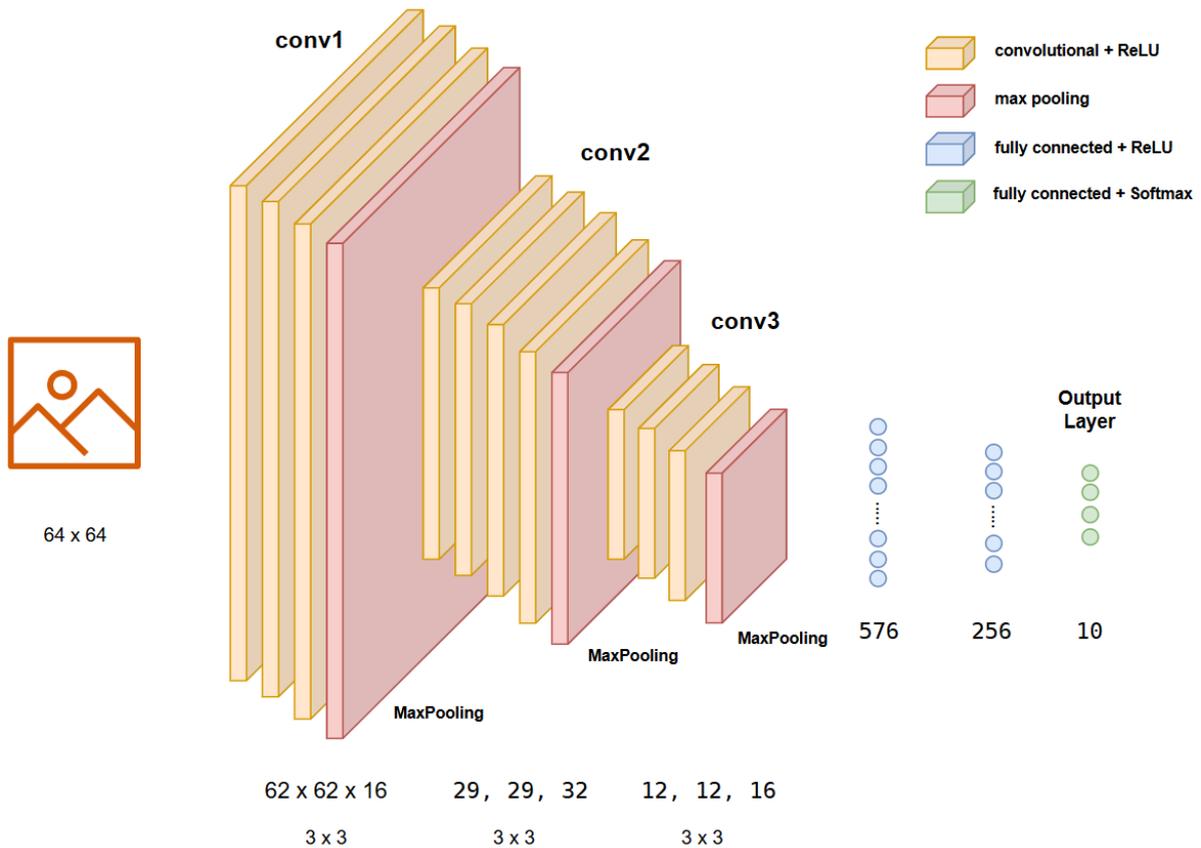


FIGURE III-5 – Architecture CNN

9 Pourquoi avons-nous choisi YOLO pour les tâches de détection et segmentation ?

YOLO est un modèle de pointe en matière de détecteurs d'objets, et il est devenu la technique la plus utilisée récemment (comme nous l'avons vu dans la section état de l'art du deuxième chapitre). Il offre plusieurs avantages par rapport à d'autres méthodes (47).

YOLO révolutionné le domaine de la vision par ordinateur. Il est rapide et efficace, ce qui en fait un excellent choix pour les tâches de détection d'objets en temps réel. Il a atteint des performances de pointe sur divers benchmarks et a été largement adopté dans diverses applications réelles.

L'un des principaux avantages de YOLO est sa rapidité d'inférence, ce qui lui permet de traiter les images en temps réel. Il est donc particulièrement bien adapté pour des applications telles que la vidéosurveillance, les voitures autonomes et la réalité augmentée. De plus, YOLO possède une architecture simple et nécessite peu de données d'entraînement, ce qui le rend facile à implémenter et à adapter à de nouvelles tâches.

9.1 Avantages de YOLOv8

YOLOv8 offre plusieurs avantages par rapport à ses prédécesseurs et à d'autres modèles de détection d'objets (48) :

- **Haute Précision** : YOLOv8 atteint une précision de pointe sur divers benchmarks de détection d'objets.
- **Vitesse en Temps Réel** : Le modèle se distingue par des vitesses d'inférence impressionnantes, le rendant adapté aux applications en temps réel telles que les véhicules autonomes et la robotique.
- **Efficacité** : YOLOv8 est léger et nécessite moins de ressources computationnelles par rapport à d'autres modèles, ce qui le rend idéal pour une utilisation sur des dispositifs en périphérie.
- **Open-source et Soutenu par la Communauté** : YOLOv8 est open-source et bénéficie du soutien d'une communauté dynamique, favorisant un développement et une amélioration continus.

10 Conclusion

Dans ce chapitre, nous avons présenté notre approche méthodologique pour concevoir et développer un système de reconnaissance automatique des plaques d'immatriculation (ANPR) adapté aux plaques algériennes. En nous appuyant sur les dernières avancées en matière de deep learning, nous avons proposé une architecture intégrée en trois phases pour détecter, segmenter et reconnaître les plaques d'immatriculation.

Dans le chapitre suivant, nous allons mettre en œuvre et tester le système conçu afin d'évaluer ses performances. Cette étape pratique nous permettra de valider notre approche théorique et d'identifier les améliorations potentielles pour rendre notre système encore plus efficace et fiable.

Chapitre IV

Expérimentations

1 Introduction

Dans le chapitre précédent, nous avons détaillé les étapes qui composent notre système DZANPR pour la reconnaissance des plaques d'immatriculation algériennes. Afin d'évaluer les performances de ce système, nous présentons dans ce chapitre les résultats obtenus lors de son expérimentation. Cette section est cruciale car elle valide l'efficacité et la performance de notre approche à travers une série d'expériences rigoureuses. Nous commencerons par décrire l'environnement de développement, en précisant le matériel et les logiciels utilisés. Ensuite, nous examinerons les ensembles de données employés pour l'entraînement et le test des modèles constituant notre système.

Nous expliquerons en détail le processus d'entraînement de chaque modèle, y compris les hyperparamètres utilisés et les configurations spécifiques pour la détection des plaques, la segmentation des caractères et la classification des chiffres. Un tableau récapitulatif des modèles entraînés sera fourni pour offrir une vue d'ensemble des performances des différents modèles.

Nous évaluerons également chaque modèle individuellement avant de procéder à une évaluation complète du système dans son ensemble. Cette évaluation comprendra des mesures de précision, de rappel et de score F1, ainsi qu'une analyse des erreurs. Enfin, nous présenterons le prototype que nous avons développé dans lequel nous avons intégré les fonctionnalités de DZANPR.

Ce chapitre se conclura par une discussion des résultats obtenus, mettant en lumière les points forts de notre système ainsi que les domaines nécessitant des améliorations futures.

2 Environnement de développement

2.1 Environnement matériel

Pour mener à bien notre projet, nous avons utilisé un ordinateur portable équipé des caractéristiques suivantes :

- Processeur : Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz
- Graphiques : Intel(R) UHD Graphics 620
- RAM : 16GB DDR4
- Disque Dur : 256GB SSD
- Système d'exploitation : Windows 11 Home 64bits

Pour un entraînement efficace des modèles, nous avons utilisé les ressources GPU fournies par Google Colab et Kaggle pour les phases d'entraînement et de test. Cette configuration nous a permis de tirer parti de la puissance de calcul nécessaire pour entraîner les modèles d'apprentissage profond de manière efficace.

2.2 Environnements logiciel

Pour le développement de notre système de reconnaissance des systèmes d'information, nous avons utilisé un ensemble varié de bibliothèques et d'outils logiciels, choisis pour leurs fonctionnalités spécifiques et leur compatibilité avec nos besoins. Voici une description des principaux outils et bibliothèques utilisés :

2.2.1 Langages de développement

Python 3.12 : Le langage principal utilisé pour le développement en raison de sa simplicité, de sa large adoption dans la communauté de l'apprentissage automatique, et de sa compatibilité avec de nombreuses bibliothèques et frameworks.

2.2.2 Bibliothèques pour le traitement d'images et la visualisation

- **openCV 4.9** : Utilisée pour le traitement des images, y compris les opérations de prétraitement telles que la conversion des images en niveaux de gris, le redimensionnement et la normalisation.
- **Matplotlib 3.8** : Utilisée pour la visualisation des données et des résultats, permettant de tracer des graphiques et des courbes illustrant les performances des modèles.

- **Numpy 1.26** : Utilisée pour les opérations mathématiques et la manipulation des tableaux de données, essentielle pour le traitement des images et la préparation des données d’entraînement.

2.2.3 Frameworks et API d’Apprentissage Profond

- **TensorFlow 2.16** : Utilisé pour le développement et l’entraînement des modèles de reconnaissance, offrant une flexibilité et une performance élevées pour les tâches d’apprentissage profond.
- **Keras 3.11** : API de haut niveau de TensorFlow, utilisée pour simplifier la construction, l’entraînement et l’évaluation des réseaux de neurones.
- **Ultralytics YOLOv8** : Utilisé pour entraîner les modèles de détection d’objets, la détection des coins (pose detection), et la segmentation des caractères. YOLOv8 offre des performances rapides et précises pour les tâches de vision par ordinateur.

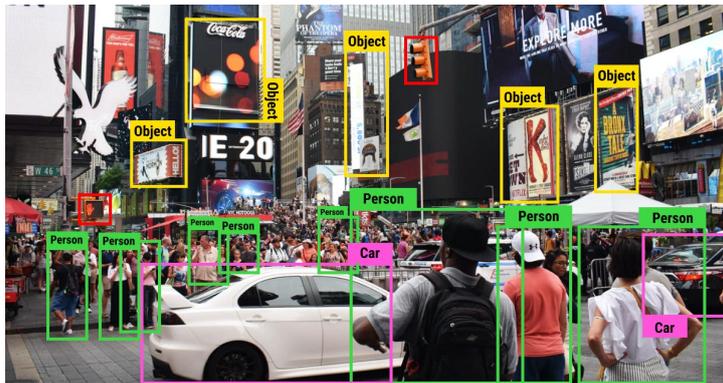


FIGURE IV-1 – Exemple de détection en temps réel avec YOLOv8
(49)

2.2.4 Outils et plates-formes pour les données et l’entraînement

- **Kaggle** : Plate-forme utilisée pour trouver des ensembles de données et pour entraîner les modèles grâce à l’environnement généreux offert, incluant deux GPU T2 avec une mémoire combinée de 30 GB.
- **Google Colab** : Environnement Jupyter dans le cloud offert par Google, utilisé pour la recherche scientifique et l’entraînement des modèles, facilitant le partage et la collaboration.
- **Roboflow** : Plate-forme complète pour les projets de vision par ordinateur, utilisée pour trouver des ensembles de données, tester les modèles entraînés, et gérer les flux de travail.

- **CVAT (Computer Vision Annotation Tool)** : Utilisé pour annoter les ensembles de données, facilitant la création d’annotations précises et de haute qualité pour l’entraînement des modèles.

2.2.5 Outils pour l’interopérabilité et l’optimisation des modèles

- **ONNX (Open Neural Network Exchange)** : Utilisé pour convertir les modèles entraînés entre différents formats, permettant une meilleure interopérabilité entre divers frameworks de deep learning.
- **OpenVINO** : Utilisé pour optimiser les modèles entraînés pour le matériel Intel, améliorant les performances d’inférence sur les dispositifs compatibles.

Ces outils et bibliothèques ont été soigneusement choisis pour leur capacité à traiter efficacement les données d’images, à entraîner des modèles complexes, et à optimiser les performances du système complet. Leur utilisation coordonnée a permis de développer un système de reconnaissance des systèmes d’information performant et fiable.

2.2.6 Déploiement et interface de test

Flask : Utilisé pour le déploiement du modèle et la création de l’interface de test. Flask a été choisi pour sa compatibilité avec le langage Python et son minimalisme, offrant le strict nécessaire pour déployer une application web légère et efficace.

3 Ensembles de données (DataSets)

Dans cette section, nous allons explorer en détail les différents ensembles de données utilisés tout au long du processus d’entraînement. Nous examinerons ensuite leurs sources et leur composition.

3.1 DataSet pour la détection des plaques

Pour faciliter la détection des plaques d’immatriculation, nous avons utilisé un dataset populaire et bien connu sur la plateforme Roboflow. Ce dataset comprend 395 images, réparties en trois parties : 277 images pour l’entraînement, 37 pour les tests, et 81 pour la validation. Chaque image est annotée, avec un fichier d’annotation portant le même nom que l’image correspondante. Ce fichier contient les informations sur les boîtes englobantes des plaques trouvées dans l’image correspondante.



FIGURE IV-2 – Exemples d’images de dataSet pour la détection de la matricule

3.2 DataSet pour la détection des quatres points

Pour détecter les quatre coins d’une plaque d’immatriculation, nous avons utilisé un jeu de données personnalisé spécialement conçu pour les plaques d’immatriculation (50). Ce jeu de données comprend des annotations pour chaque coin de la plaque d’immatriculation, méticuleusement marquées pour assurer la précision. Afin de simplifier le processus de détection, nous avons transformé ces annotations au format YOLO les optimisant pour une détection d’objets efficace.



FIGURE IV-3 – Exemples d’images de dataSet pour la détection des 4 points de la plaque

3.3 DataSet pour segmentation des caractères

Pour extraire chaque caractère des plaques d’immatriculation, nous avons utilisé un jeu de données créé par Francesco (51). Ce jeu de données comprend 167 images d’entraînement et 42 images de validation. Chaque image est annotée, et chaque étiquette contient tous les points délimitant chaque caractère.



FIGURE IV-4 – Exemples d’images de dataSet pour la segmentation des caractères

3.4 DataSet pour reconnaissance des chiffres

Dans notre recherche, nous avons exploité la puissance de trois ensembles de données distincts pour faire progresser l’état de l’art en matière de reconnaissance de caractères de plaques d’immatriculation. Premièrement, nous avons utilisé un ensemble de données comprenant des caractères de plaques d’immatriculation extraits d’un groupe de 209 plaques d’immatriculation, totalisant 2026 caractères. Ces caractères ont subi une augmentation, ce qui a permis d’étendre l’ensemble de données à plus de 10 000 caractères répartis sur dix classes (de 0 à 9), en se concentrant uniquement sur les caractères numériques.

De plus, nous avons utilisé un deuxième ensemble de données comprenant des chiffres imprimés avec diverses polices de caractères. De cet ensemble, nous avons conservé sélectivement les caractères qui ressemblaient étroitement aux polices couramment trouvées sur les plaques d’immatriculation, en éliminant ceux qui s’écartaient significativement de notre style ciblé.

En outre, nous avons développé un ensemble de données en capturant des plaques d’immatriculation et en découpant ensuite leurs caractères individuels. Ces caractères découpés ont ensuite été soumis au même processus d’augmentation que les caractères du premier ensemble de données. Cette addition a été motivée par le besoin de diversifier davantage les polices représentées dans notre ensemble de données et d’étendre la gamme des variations des chiffres. En capturant directement des caractères à partir de plaques d’immatriculation réelles, nous avons veillé à inclure des polices et des styles qui pourraient ne pas avoir été adéquatement représentés dans les ensembles de données précédents. Cette stratégie visait à enrichir l’ensemble de données avec un spectre plus large de polices et

de variations de chiffres rencontrés dans des scénarios réels, renforçant ainsi la robustesse et l'adaptabilité de notre système de reconnaissance.

En combinant ces trois ensembles de données, nous avons créé un ensemble de données complet contenant plus de 20 000 échantillons de chiffres. Cet ensemble de données amalgamé a non seulement enrichi nos données d'entraînement, mais a également assuré une représentation diversifiée des caractères de plaques d'immatriculation, améliorant ainsi la robustesse et les capacités de généralisation de notre système de reconnaissance.

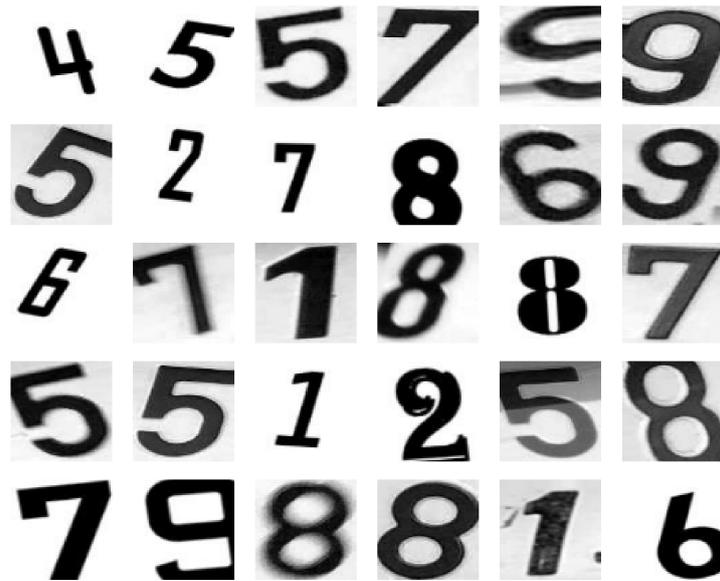


FIGURE IV-5 – Exemples d'images de dataSet pour la reconnaissance des chiffres

Dataset Name	Number of Images	Image Size (Width × Height)	Classes
Licence Plate Detection Dataset (52)	395	Variate	Licence Plate
License plate key point detection (50)	1254	128 × 64	Corners
License Plate Characters Detection OCR (51)	209	Variate	0..9, A..Z
English Font-Number Recognition (53)	62992	128 × 128	0..9, A..Z, a..z
License Plate Digits Classification Dataset (54)	36360	75 × 100	0..9, A..Z

TABLE IV.1 – Tableau récapitulatif des différents datasets utilisés dans l'entraînement

4 Entraînement des modèles

Dans cette section, nous allons analyser la performance des modèles développés en utilisant divers critères d'évaluation. Nous évaluerons chaque modèle individuellement, cette approche nous permettra de comprendre les performances spécifiques de chaque modèle avant de procéder à l'évaluation complète du système, qui fera l'objet d'une section dédiée.

4.1 Entraînement de modèle de détection des plaques

Le détecteur YOLO a été entraîné sur un ensemble de données mentionné précédemment, organisé selon une structure spécifique. Les images ont été placées dans un dossier "images" et les étiquettes correspondantes dans un dossier "labels", partageant le même nom de fichier. L'ensemble de données a ensuite été divisé en partitions d'entraînement (80%), de validation (10%), et de test (10%).

Le processus d'entraînement a tiré parti d'un modèle YOLOv8n pré-entraîné (poids yolov8n.pt) pour l'initialisation. Les hyperparamètres ont été définis comme suit : 100 époques, une taille d'image de 640x640 pixels, et une taille de lot de 16. Dans un environnement Google Colab avec un processeur Intel Xeon (2,00 GHz), 12 Go de RAM, 80 Go de disque, et un GPU T4 avec 15 Go de VRAM, le processus d'entraînement a duré environ 30 minutes.

4.2 Entraînement de modèle de segmentation des caractères

Comme mentionné précédemment, le processus de segmentation passe par deux étapes : la correction de l'inclinaison et la détection des boîtes englobantes des caractères. Ainsi, nous avons entraîné deux modèles distincts.

Un détecteur de pose YOLO, spécifiquement l'architecture YOLOv8n-pose, a été choisi comme base pour notre modèle de correction de l'inclinaison. Pour entraîner ce détecteur, nous avons utilisé l'ensemble de données mentionné précédemment. Les hyperparamètres ont été soigneusement sélectionnés : 50 époques d'entraînement, une taille d'image de 128x64 pixels, et une taille de lot de 32. L'entraînement a été réalisé dans un environnement Google Colab équipé des spécifications mentionnées plus haut.

Un autre modèle a été entraîné en utilisant la même méthodologie pour la détection des boîtes englobantes des caractères, en utilisant un ensemble de données distinct. Les données ont été organisées et partitionnées selon les procédures décrites précédemment.

4.3 Entraînement de CNN pour la classification des chiffres

Le processus d'entraînement du réseau de neurones convolutionnel (CNN) pour la classification des images de chiffres implique plusieurs étapes clés. Ces expériences ont été réalisées sur la plateforme de Kaggle, en utilisant des ressources telles que deux GPU T4, 32 Go de RAM, et un processeur Intel Xeon à 2,00 GHz avec un espace disque de 73 Go.

Initialement, l'ensemble de données préparé contenant des images de chiffres, décrit précédemment, est partitionné en ensembles d'entraînement, de validation et de test, avec des proportions respectives de 70%, 20% et 10%. Une normalisation est appliquée aux images pour améliorer les performances et la généralisation du modèle.

L'architecture du CNN, détaillée précédemment, est ensuite compilée en utilisant la fonction de perte sparse categorical cross-entropy et l'optimiseur Adam. L'entraînement est effectué sur 30 époques, avec une taille de lot de 32 et un taux d'apprentissage de 0,001. Processus d'Entraînement.

Tout au long de l'entraînement, le CNN apprend progressivement à reconnaître des motifs et des caractéristiques dans les images de chiffres, en mettant à jour ses paramètres de manière itérative grâce aux passes avant et arrière. La validation est effectuée périodiquement pour surveiller le surapprentissage et ajuster les hyperparamètres si nécessaire.

4.4 Tableau récapitulatif des modèles entraînés

Modèles	Entraînement de Dataset	Temps d'entraînement	Environnement d'entraînement	Performance
Détecteur de plaque d'immatriculation	395 images	30 minutes	Google Colab	0.95 mAP
Détecteur des quatre coins des plaques	1254 images	45 minutes	Google Colab	0.95 mAP
Segmenteur de caractères	209 images	30 minutes	Google Colab	0.98 mAP
Reconnaisseur de caractères	32992 images	15 minutes	Kaggle	0.98 accuracy

TABLE IV.2 – Tableau des modèles entraînés

4.5 Exportation des modèles

Les modèles entraînés ont été exportés dans des fichiers de poids (weights) pour une utilisation ultérieure. Ces fichiers contiennent les paramètres appris par les modèles pendant l'entraînement, permettant de les charger et de les utiliser pour l'inférence sans avoir à ré-entraîner les modèles à chaque fois. Les fichiers de poids sont compatibles avec les frameworks de deep learning tels que TensorFlow et Keras, facilitant leur intégration dans des applications et des systèmes en production.

Afin d'optimiser la rapidité et l'efficacité de l'inférence, les modèles ont également été convertis en formats compatibles avec les bibliothèques d'inférence telles que OpenVINO. Ces formats optimisés permettent d'accélérer l'inférence sur des dispositifs matériels spécifiques, tels que les processeurs Intel, en exploitant les capacités de calcul spécifiques de ces dispositifs.

Modèle	Taille de l'image en (px)	Vitesse d'inférence sans optimisation en (ms)	Vitesse d'inférence avec OpenVINO en (ms)
Détecteur de plaques	720x1280	150 - 240	30 -50
Segmenteur de plaques (les 2 modèles combinés)	60x240	300 - 450	60-80
Reconnaissance des chiffres	64x64	15 - 40 par caractère	2 par caractère

TABLE IV.3 – Les délais d'exécution

5 Évaluation des modèles

5.1 Evaluation de modèle de détection des plaques

Pour évaluer les performances de notre modèle de détection des plaques d'immatriculation, nous avons utilisé le jeu de données LPA de (55), qui comprend des plaques d'immatriculation algériennes. Ce jeu de données contient 2 440 plaques annotées. Notre modèle a détecté 2 442 éléments, dont 52 fausses détections, ce qui donne une précision de 97%.

Voici la matrice de confusion pour l'évaluation du modèle de détection des plaques d'immatriculation :

Prédiction \ Vérité	Plaque d'immatriculation	Non-Plaque d'immatriculation
	Plaque d'immatriculation	2390
Non-Plaque d'immatriculation	20	0

TABLE IV.4 – Matrice de confusion pour la détection des plaques d'immatriculation

5.2 Évaluation de modèle de segmentation des caractères

La phase de segmentation a été évaluée en conjonction avec la phase de détection, traitant la sortie combinée comme un modèle unique. Nous avons annoté 50 images avec les valeurs réelles. Ces images contenaient 50 plaques d'immatriculation et 526 caractères au total. Notre modèle a correctement segmenté 525 caractères, ce qui représente un taux de réussite de 99%. Cependant, il a également ajouté incorrectement 4 caractères qui n'auraient pas dû être segmentés.

Prédiction \ Vérité	Caractère	Non-Caractère
	Caractère	525
Non-Caractère	1	0

TABLE IV.5 – Matrice de confusion pour la segmentation des plaques d'immatriculation

5.3 Évaluation de modèle de reconnaissance des chiffres

Pour évaluer les performances de notre modèle de reconnaissance des chiffres, nous avons utilisé un ensemble de données de test contenant 1 000 images de chiffres. Chaque image a été annotée avec le chiffre correspondant, permettant de comparer les prédictions du modèle aux valeurs réelles. Le modèle a correctement identifié 990 images, obtenant ainsi un taux de réussite de 99%. Les 10 erreurs restantes étaient principalement des confusions entre des chiffres similaires, tels que 6 et 0, ou 1 et 7.

Matrice de Confusion

1	95	0	0	0	0	0	0	0	2	0
2	0	100	0	0	0	0	0	0	0	0
3	0	0	100	0	0	0	0	0	0	0
4	0	0	0	99	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0
6	0	0	0	1	0	96	1	0	0	0
7	3	0	0	0	0	0	96	0	0	0
8	0	0	0	0	0	0	0	100	0	0
9	0	0	0	0	0	0	0	0	98	0
0	0	0	0	0	0	2	0	0	0	98
	1	2	3	4	5	6	7	8	9	0

Classe Réelle

Classe Prédite

FIGURE IV-6 – Matrice de confusion pour la reconnaissance des chiffres

6 Évaluation du système complet

Pour évaluer le système dans son intégralité, nous avons créé un ensemble de test contenant 50 images de véhicules simulant des conditions réelles avec différents angles de caméra. Chaque image a été annotée avec la plaque d'immatriculation correspondante (qui figure dans l'image). Le système reçoit une image en entrée et produit en sortie la plaque d'immatriculation sous forme de texte. En comparant cette prédiction à la valeur réelle, nous avons pu estimer les performances de notre système.

Après avoir traité toutes les images, nous avons inspecté les résultats. Le système a pu identifier parfaitement 47 images sur 50, obtenant ainsi un score de 94% de succès. En inspectant les trois images où le système s'est trompé, nous avons constaté qu'il s'agissait d'erreurs minimales (un chiffre non reconnu ou mal segmenté). La figure suivante montre les trois images que le système n'a pas bien reconnues.

Comparaison des valeurs réelles et prédites



FIGURE IV-7 – Exemples d’images mal reconnues par le système

Pour illustrer l’efficacité de notre système, nous présentons ci-dessous un échantillon d’images qui ont été correctement reconnues. Ces exemples montrent la capacité du système à identifier avec précision les plaques d’immatriculation même dans des conditions variées et parfois difficiles. La figure suivante montre quelques-unes des images bien reconnues par le système.

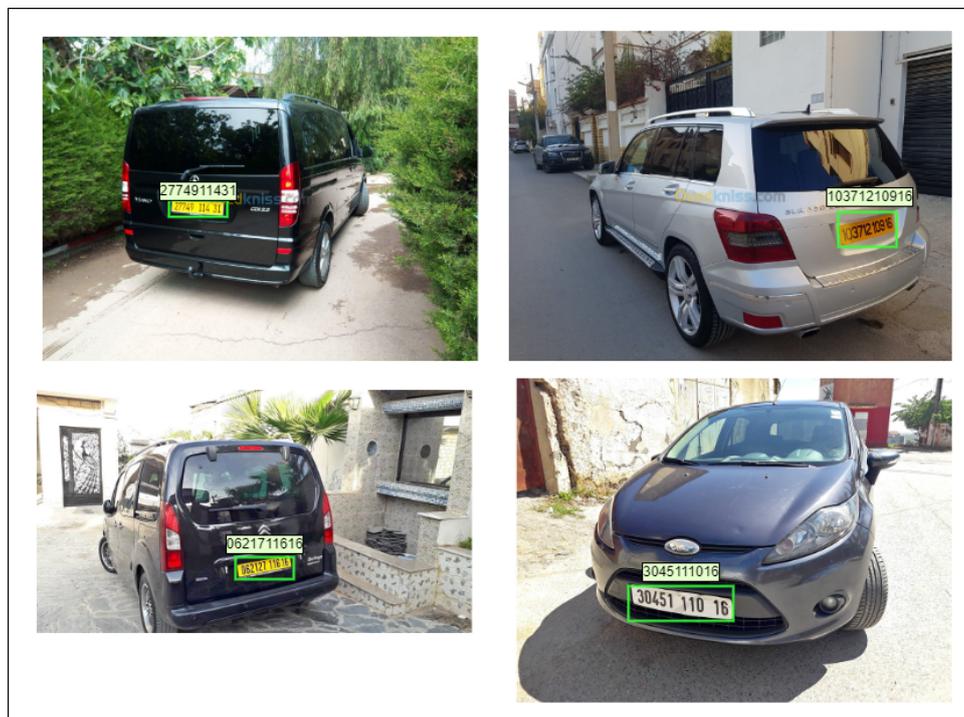


FIGURE IV-8 – Exemples d’images bien reconnues par le système

7 Interface de test

Pour faciliter l'évaluation du système et la démonstration de ses capacités, nous avons développé une interface de test conviviale. Cette interface permet à l'utilisateur de télécharger une image de véhicule, de soumettre l'image au système, et de visualiser le résultat de la reconnaissance de la plaque d'immatriculation.

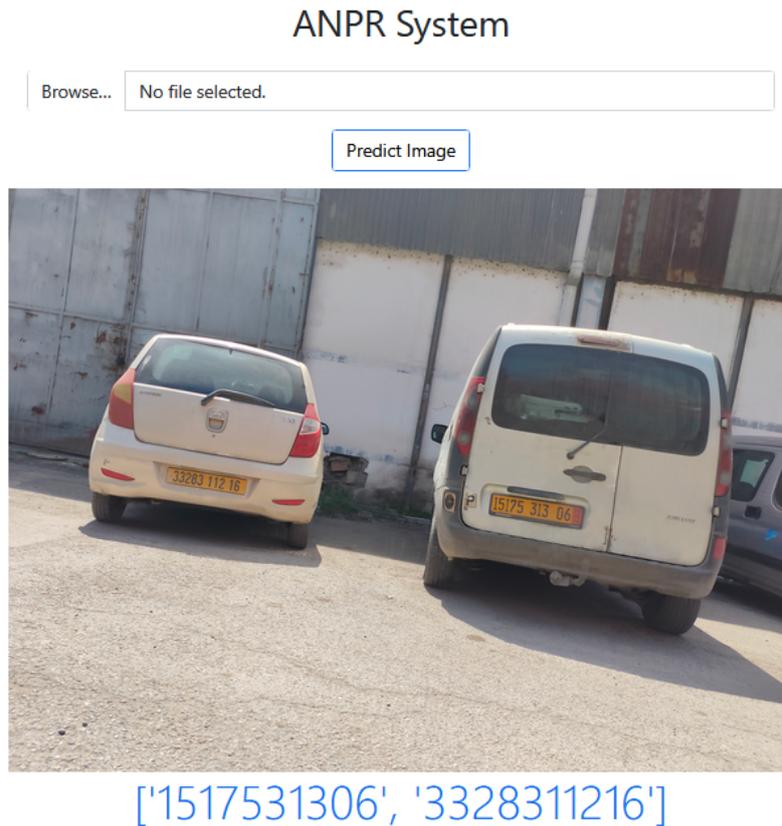


FIGURE IV-9 – Interface de test du système de reconnaissance des plaques d'immatriculation

8 Conclusion

Dans ce chapitre, nous avons présenté les tests et les résultats de notre système de reconnaissance des plaques d'immatriculation. Nous avons détaillé l'environnement de développement, les ensembles de données utilisés, le processus d'entraînement des modèles, et les performances des différents modèles. Nous avons évalué chaque modèle individuellement avant de procéder à une évaluation complète du système dans son ensemble. Enfin, nous avons décrit l'interface de test que nous avons développée pour faciliter l'évaluation du système par les utilisateurs finaux.

Conclusion Générale

L'objectif de notre projet est le développement d'un système de reconnaissance automatique des plaques d'immatriculation ANPR adapté aux plaques algériennes. Ce système est baptisé **DZANPR**. Le besoin d'un tel système se fait de plus en plus ressentir dans divers domaines tels que la gestion de la circulation, la sécurité routière, la gestion des parkings, et le contrôle d'accès. En Algérie, l'absence d'un système ANPR robuste représente un obstacle significatif en raison des défis spécifiques de la région, notamment les conditions environnementales, les caractéristiques des plaques d'immatriculation, et les contraintes techniques.

Notre recherche a suivi une méthodologie structurée en trois étapes : l'extraction des plaques, la segmentation des caractères, et la reconnaissance des caractères. Nous avons utilisé des techniques avancées d'apprentissage profond, en particulier le modèle de détection d'objets YOLO pour l'extraction des plaques et un CNN pour la reconnaissance des caractères. Ces choix technologiques se sont avérés judicieux pour relever les défis uniques posés par les plaques d'immatriculation algériennes.

Notre parcours de recherche a été riche et instructif. Nous avons commencé par explorer les bases fondamentales du traitement des images et les principales techniques d'apprentissage automatique et de deep learning. Ensuite, nous avons examiné les systèmes ANPR existants, en mettant en lumière les approches traditionnelles et les avancées récentes. Nous avons ensuite développé notre propre système, en adaptant les techniques à notre contexte spécifique. Enfin, nous avons procédé à des tests rigoureux pour évaluer les performances de notre système.

Les résultats obtenus sont prometteurs. Notre système a démontré une précision élevée dans la détection et la reconnaissance des plaques d'immatriculation algériennes, surpassant les solutions génériques existantes. Ces résultats sont le fruit de nombreuses itérations et optimisations, ainsi que de l'application de techniques avancées de prétraitement et de traitement d'image.

Pour l'avenir, plusieurs perspectives s'ouvrent à nous. Nous envisageons de développer un modèle *end-to-end* qui intégrera toutes les étapes de la reconnaissance des plaques d'immatriculation dans un seul pipeline. Un autre axe d'amélioration serait de concevoir un système intelligent capable d'appliquer des techniques d'amélioration de l'image uniquement lorsque cela est nécessaire, afin d'optimiser le temps de traitement. De plus,

nous prévoyons de continuer à affiner notre modèle pour améliorer encore la précision et la vitesse de détection.

Ce projet a été une expérience extrêmement enrichissante. Il nous a permis de mettre en pratique des concepts théoriques avancés et de surmonter des défis techniques complexes. Nous avons acquis une expertise précieuse en apprentissage profond et en traitement d'images, tout en contribuant à une solution concrète à un problème réel. Les compétences et les connaissances développées au cours de ce projet nous seront sans doute très utiles dans nos futures entreprises académiques et professionnelles.

En conclusion, le développement de notre système ANPR dédié aux plaques d'immatriculation algériennes représente une avancée significative dans ce domaine. Les résultats obtenus sont encourageants et ouvrent la voie à de nombreuses améliorations et applications futures. Nous espérons que notre travail pourra servir de base à d'autres recherches et développements dans ce domaine crucial pour la gestion de la circulation et la sécurité routière en Algérie et au-delà.

Bibliographie

- [1] “Le modèle TSL.” [Online]. Available : <https://e-cours.univ-paris1.fr/modules/ued/envcal/html/compositions-colorees/representations-couleur/modeles-perceptuels/tsl.html>
- [2] A. Ben Sada, “Machine Learning for Algerian Car Plate Number Recognition,” Master’s thesis, UNIVERSITY OF AMAR TELIDJI LAGHOUAT, 2016.
- [3] “image / imagé / Définition IMAGE / IMAGÉ.” [Online]. Available : <https://www.le-dictionnaire.com/definition/image>
- [4] “IMAGE : Définition de IMAGE.” [Online]. Available : <https://cmrtl.fr/definition/image>
- [5] “L’image numérique.” [Online]. Available : <http://www.map.toulouse.archi.fr/works/panoformation/imagenum/imagenum.htm>
- [6] Y. BENACHOUR and S. ANEDE, “Conception et Réalisation d’un Système de Reconnaissance de Plaque d’Immatriculation de Véhicules en Temps Réel,” Master’s thesis, Université SAAD DAHLAB de BLIDA, 2016.
- [7] “Types et formats de fichiers d’images.” [Online]. Available : <https://www.bang-marketing.com/publications/ebang/comprendre-enfin-les-differents-types-de-fichiers-graphiques/>
- [8] “What is ANPR? - ANPR International.” [Online]. Available : <https://www.anpr-international.com/what-is-anpr/>
- [9] “Le code RVB et les données colorimétriques.” [Online]. Available : <http://www.123couleurs.fr/explications/explications-lumiÃre/tl-codervb/>
- [10] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, “An efficient and layout-independent automatic license plate recognition system based on the YOLO detector,” *IET Intelligent Transport*

- Systems*, vol. 15, no. 4, pp. 483–503, Apr. 2021. [Online]. Available : <https://onlinelibrary.wiley.com/doi/10.1049/itr2.12030>
- [11] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 4th ed. New York, NY : Pearson, 2018.
- [12] “Grayscale,” Feb. 2024, page Version ID : 1202807394. [Online]. Available : <https://en.wikipedia.org/w/index.php?title=Grayscale&oldid=1202807394>
- [13] “Créer une image en niveaux de gris en ligne - Outil gratuit.” [Online]. Available : <https://grayscale.imageonline.co/fr/>
- [14] “Traitement d’images,” Oct. 2023, page Version ID : 208842044. [Online]. Available : https://fr.wikipedia.org/w/index.php?title=Traitement_d%27images&oldid=208842044
- [15] “Première Chaîne Complète de Segmentation.” [Online]. Available : <https://patrick-bonnin.developpez.com/cours/vision/apprendre-bases-traitement-image/partie-4-premiere-chaine-complete-segmentation/>
- [16] “Détection de contours,” Jun. 2023, page Version ID : 205329269. [Online]. Available : https://fr.wikipedia.org/w/index.php?title=D%C3%A9tection_de_contours&oldid=205329269
- [17] “OpenCv Perspective Transformation | by Raqueeb Shaikh | Analytics Vidhya | Medium.” [Online]. Available : <https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143>
- [18] “Amazon.com : Willi Richert : books, biography, latest update.” [Online]. Available : https://www.amazon.com/stores/Willi-Richert/author/B07G9SCCC?ref=ap_rdr&isDramIntegrated=true&shoppingPortalEnabled=true
- [19] “OpenCV : Transformations morphologiques.” [Online]. Available : https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html
- [20] “Qu’est-ce que l’apprentissage automatique ? | NordVPN,” Feb. 2023. [Online]. Available : <https://nordvpn.com/fr/blog/apprentissage-automatique/>
- [21] R. Kassel, “Algorithme de classification : Définition et principaux modèles,” Nov. 2022. [Online]. Available : <https://datascientest.com/algorithme-de-classification-definition-et-principaux-modeles>
- [22] C. Deluzarche, “Définition | Deep Learning - Apprentissage profond | Futura

- Tech.” [Online]. Available : <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>
- [23] “Quels sont les différents algorithmes de Deep Learning? | Jedha.” [Online]. Available : <https://www.jedha.co/formation-ia/algorithmes-deep-learning>
- [24] “What is OCR (Optical Character Recognition)? | IBM,” Apr. 2024. [Online]. Available : <https://www.ibm.com/topics/optical-character-recognition>
- [25] “What is ANPR (Automatic Number-Plate Recognition)? ANPR Meaning | Isarsoft.” [Online]. Available : <https://www.isarsoft.com/knowledge-hub/anpr>
- [26] K. T, “What Is ANPR? Automatic Number Plate Recognition Explained,” Oct. 2023. [Online]. Available : <https://platerecognizer.com/what-is-anpr/>
- [27] “Des données de formation de haute qualité alimentent des véhicules autonomes hautement performants | Shaip,” Apr. 2022, section : Artificial intelligence. [Online]. Available : <https://fr.shaip.com/blog/training-data-for-autonomous-vehicles/>
- [28] “Transport department approaches Central govt for ANPR cameras installation - Pioneer Edge | Uttarakhand News in English | Dehradun News Today| News Uttarakhand | Uttarakhand latest news,” Oct. 2021, section : State. [Online]. Available : <https://www.pioneeredge.in/transport-department-approaches-central-govt-for-anpr-cameras-installation/>
- [29] J. Tang, L. Wan, J. Schooling, P. Zhao, J. Chen, and S. Wei, “Automatic number plate recognition (ANPR) in smart cities : A systematic review on technological advancements and application cases,” *Cities*, vol. 129, p. 103833, Oct. 2022. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S0264275122002724>
- [30] C. Patel, D. Shah, and A. Patel, “Automatic Number Plate Recognition System (ANPR) : A Survey,” *International Journal of Computer Applications*, vol. 69, no. 9, pp. 21–33, May 2013. [Online]. Available : <http://research.ijcaonline.org/volume69/number9/pxc3887665.pdf>
- [31] Lubna, N. Mufti, and S. A. A. Shah, “Automatic Number Plate Recognition :A Detailed Survey of Relevant Algorithms,” *Sensors*, vol. 21, no. 9, p. 3028, Apr. 2021. [Online]. Available : <https://www.mdpi.com/1424-8220/21/9/3028>
- [32] W. Weihong and T. Jiaoyang, “Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment,” *IEEE Access*, vol. 8, pp. 91 661–91 675, 2020. [Online]. Available : <https://ieeexplore.ieee.org/document/9092977/>

- [33] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning," *IOP Conference Series : Materials Science and Engineering*, vol. 1084, no. 1, p. 012027, Mar. 2021. [Online]. Available : <https://iopscience.iop.org/article/10.1088/1757-899X/1084/1/012027>
- [34] D. M. F. Izidio, A. P. A. Ferreira, H. R. Medeiros, and E. N. d. Barros, "An embedded automatic license plate recognition system using deep learning," *Design Automation for Embedded Systems*, vol. 24, no. 1, pp. 23–43, Mar. 2020. [Online]. Available : <http://link.springer.com/10.1007/s10617-019-09230-5>
- [35] A. Amrouche, N. Hezil, Y. Bentrchia, and A. Abed, "Real-Time Detection of Vehicle License Plates Numbers," in *2022 2nd International Conference on New Technologies of Information and Communication (NTIC)*. Mila, Algeria : IEEE, Dec. 2022, pp. 1–5. [Online]. Available : <https://ieeexplore.ieee.org/document/10100479/>
- [36] R. Zibani, F. Sebbak, M. E. Y. Boudaren, M. Mataoui, R. Hadj Aissa, and Y. A. Benaissa, "Multi-attribute fusion-based approach for Algerian automatic license plate recognition," *Multimedia Tools and Applications*, vol. 83, no. 10, pp. 30 233–30 259, Sep. 2023. [Online]. Available : <https://link.springer.com/10.1007/s11042-023-16789-6>
- [37] M. Bensouilah, M. Zennir, and M. Taffar, "An ALPR System-based Deep Networks for the Detection and Recognition :," in *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods*. Online Streaming, — Select a Country — : SCITEPRESS - Science and Technology Publications, 2021, pp. 204–211. [Online]. Available : <https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010229202040211>
- [38] N. Omar, A. Sengur, and S. G. S. Al-Ali, "Cascaded deep learning-based efficient approach for license plate detection and recognition," *Expert Systems with Applications*, vol. 149, p. 113280, Jul. 2020. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S0957417420301056>
- [39] Z. Selmi, M. Ben Halima, and A. M. Alimi, "Deep Learning System for Automatic License Plate Detection and Recognition," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Kyoto : IEEE, Nov. 2017, pp. 1132–1138. [Online]. Available : <http://ieeexplore.ieee.org/document/8270118/>
- [40] D. Zang, Z. Chai, J. Zhang, D. Zhang, and J. Cheng, "Vehicle license plate recognition using visual attention model and deep learning," *Journal of Electronic*

- Imaging*, vol. 24, no. 3, p. 033001, May 2015. [Online]. Available : <http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.JEI.24.3.033001>
- [41] J. I. Zong Chen, “Automatic Vehicle License Plate Detection using K-Means Clustering Algorithm and CNN,” *Journal of Electrical Engineering and Automation*, vol. 3, no. 1, pp. 15–23, May 2021. [Online]. Available : <https://irojournals.com/iroeea/V3/I1/02.pdf>
- [42] A. Kashyap, B. Suresh, A. Patil, S. Sharma, and A. Jaiswal, “Automatic Number Plate Recognition,” in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. Greater Noida (UP), India : IEEE, Oct. 2018, pp. 838–843. [Online]. Available : <https://ieeexplore.ieee.org/document/8748287/>
- [43] I. V. Pustokhina, D. A. Pustokhin, J. J. P. C. Rodrigues, D. Gupta, A. Khanna, K. Shankar, C. Seo, and G. P. Joshi, “Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems,” *IEEE Access*, vol. 8, pp. 92 907–92 917, 2020. [Online]. Available : <https://ieeexplore.ieee.org/document/9088975/>
- [44] U. G. Naidu, R. Thiruvengatanadhan, S. Narayana, and P. Dhanalakshmi, “Character Level Segmentation and Recognition using CNN Followed Random Forest Classifier for NPR System,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, 2022. [Online]. Available : <http://thesai.org/Publications/ViewPaper?Volume=13&Issue=11&Code=IJACSA&SerialNo=2>
- [45] Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, “DELP-DAR system for license plate detection and recognition,” *Pattern Recognition Letters*, vol. 129, pp. 213–223, Jan. 2020. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S0167865519303216>
- [46] M. Z. Abedin, A. C. Nath, P. Dhar, K. Deb, and M. S. Hossain, “License plate recognition system based on contour properties and deep learning model,” in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. Dhaka, Bangladesh : IEEE, Dec. 2017, pp. 590–593. [Online]. Available : <https://ieeexplore.ieee.org/document/8289029/>
- [47] “YOLO Algorithm for Object Detection Explained [+Examples].” [Online]. Available : <https://www.v7labs.com/blog/yolo-object-detection>

-
- [48] “YOLOv8 Architecture : A Deep Dive into its Architecture - YOLOv8,” Jan. 2024, section : Blog. [Online]. Available : <https://yolov8.org/yolov8-architecture/>
- [49] D. I. Onat, “AI Testing Out-of-the-Box,” Oct. 2022. [Online]. Available : <https://medium.com/@denizidil.copur/ai-testing-out-of-the-box-609a87010a68>
- [50] M. Feng, “Changed-CNN/License-plate-key-point-detection,” Oct. 2019, original-date : 2019-08-26T03 :14 :35Z. [Online]. Available : <https://github.com/Changed-CNN/License-plate-key-point-detection>
- [51] P. Francesco, “License Plate Characters - Detection OCR,” 2022. [Online]. Available : <https://www.kaggle.com/datasets/francescopettini/license-plate-characters-detection-ocr/data>
- [52] Mochoye, “License Plate Detector Dataset,” Aug. 2023. [Online]. Available : <https://universe.roboflow.com/mochoye/license-plate-detector-ogxxg>
- [53] G. Yaswanth, “English Font-Number Recognition.” [Online]. Available : <https://www.kaggle.com/datasets/yaswanthgali/english-fontnumber-recognition>
- [54] Jelal, “License Plate Digits Classification Dataset.” [Online]. Available : <https://www.kaggle.com/datasets/aladdins/license-plate-digits-classification-dataset>
- [55] M. Bensouilah, M. N. Zennir, and M. Taffar, “License Plates of Algeria Dataset,” Apr. 2024, original-date : 2019-07-10T20 :22 :33Z. [Online]. Available : https://github.com/mouadb0101/License_Plates_of_Algeria_Dataset
- [56] W. Puarungroj and N. Boonsirisumpun, “Thai License Plate Recognition Based on Deep Learning,” *Procedia Computer Science*, vol. 135, pp. 214–221, 2018. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S1877050918314571>
- [57] C. Henry, S. Y. Ahn, and S.-W. Lee, “Multinational License Plate Recognition Using Generalized Character Sequence Detection,” *IEEE Access*, vol. 8, pp. 35 185–35 199, 2020. [Online]. Available : <https://ieeexplore.ieee.org/document/9003211/>
- [58] P. Francesco, “License Plate Characters - Detection OCR,” 2022. [Online]. Available : <https://www.kaggle.com/datasets/francescopettini/license-plate-characters-detection-ocr/data>
- [59] Jelal, “License Plate Digits Classification Dataset.”
- [60] M. Feng, “Changed-CNN/License-plate-key-point-detection,” Oct. 2019, original-

- date : 2019-08-26T03 :14 :35Z. [Online]. Available : <https://github.com/Changed-CNN/License-plate-key-point-detection>
- [61] G. Yaswanth, “English Font-Number Recognition.” [Online]. Available : <https://www.kaggle.com/datasets/yaswanthgali/english-fontnumber-recognition>
- [62] G. Yaswazfazefnth, “roro.” [Online]. Available : [number-recognition](#)
- [63] “History of ANPR - ANPR International.” [Online]. Available : <https://www.anpr-international.com/history-of-anpr/>
- [64] vp koen, “Automatic Number Plate Recognition ANPR in License Plate Technology,” Feb. 2024. [Online]. Available : <https://visionplatform.ai/anpr-automatic-number-plate-recognition/>
- [65] “Britannia Parking.” [Online]. Available : <https://www.britannia-carparkmanagement.co.uk/car-park-management-news-advice/how-anpr-systems-have-evolved-over-the-years>
- [66] I. Insider, “Your complete guide to understanding ANPR camera solutions | IFSEC Insider,” Jan. 2021. [Online]. Available : <https://www.ifsecglobal.com/video-surveillance/understanding-anpr-solutions/>
- [67] K. TOUAMRIA, “Reconnaissance automatique des plaques d’immatriculation,” Master’s thesis, Université 8Mai 1945 – Guelma, 2019.
- [68] S. GADOUI and F. KEBIR, “Lecture Automatique de Plaques d’Immatriculation par les Machines à Vecteurs de Support,” Master’s thesis, Université Ahmed Draia – Adrar, 2020.
- [69] A. BOUMIMEZ and M. DJIDJELI, “SYSTEME DE RECONNAISSANCE AUTOMATIQUE DES PLAQUES D’IMMATRICULATION EN UTILISANT L’APPRENTISSAGE ARTIFICIEL,” Master’s thesis, UNIVERSITE MOHAMED SEDDIK BENYAHIA -JIJEL-, 2021.
- [70] R. OUADI and H. ABDICHE, “Reconnaissance des plaques d’immatriculations par Raspberry Pi,” Master’s thesis, Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj, 2023.
- [71] M. BOUDISSA and M. KISSOUM, “Algeria license plate recognition system using Faster-RCNN and YOLO models,” Master’s thesis, University M’Hamed BOUGARA – Boumerdes, 2020.

- [72] O. e. k. AKACEM and N. RAHMANI, “Système de reconnaissance des plaques d’immatriculation Algérienne,” Master’s thesis, Université d’Adrar, 2015.
- [73] M. HAMMOUDI and R. A. KECHRA, “Systèmes d’Identification Automatique des Véhicules,” Master’s thesis, UNIVERSITE IBN KHALDOUN - TIARET, 2022.
- [74] A. K. Jain, *Fundamentals of digital image processing*, ser. Prentice Hall information and system sciences series. Englewood Cliffs, NJ : Prentice Hall, 1989.
- [75] R. C. Gonzalez and R. E. Woods, *Digital image processing*. New York, NY : Pearson, 2018.
- [76] “Détection de contours,” Jun. 2023, page Version ID : 205329269. [Online]. Available : https://fr.wikipedia.org/w/index.php?title=D%C3%A9tection_de_contours&oldid=205329269
- [77] R. Kassel, “OpenCV : tout savoir sur le principal outil de Computer Vision,” Sep. 2022. [Online]. Available : <https://datascientest.com/opencv>
- [78] “10 things you need to know about Ultralytics YOLOv8.” [Online]. Available : <https://www.augmentedstartups.com/blog/10-things-you-need-to-know-about-ultralytics-yolov8>
- [79] “TensorFlow : la plateforme de deep learning de Google,” Sep. 2022. [Online]. Available : <https://www.journaldunet.fr/intelligence-artificielle/guide-de-l-intelligence-artificielle/1501861-tensorflow-google/>
- [80] G. Xin and P. Fan, “A lossless compression method for multi-component medical images based on big data mining,” *Scientific Reports*, vol. 11, no. 1, p. 12372, Jun. 2021. [Online]. Available : <https://www.nature.com/articles/s41598-021-91920-x>
- [81] “Codage numérique des images matricielles – l’Informatique, c’est fantastique!” [Online]. Available : <https://info.blaise-pascal.fr/isn/isn-representation-de-linformation/isn-codage-numerique-des-images-matriciellles/>
- [82] d. r. Fabula, “Vivien Philizot, Images premières. Aux origines de la représentation visuelle,” Feb. 2023, publisher : Équipe de recherche Fabula, École Normale Supérieure, 45 rue d’Ulm, 75230 Paris Cedex 05. [Online]. Available : <https://www.fabula.org/actualites/112748/vivien-philizot-images-premieres-aux-origines-de-la-representation-visuelle.html>
- [83] R. Shaikh, “OpenCv Perspective Transformation,” Feb.

2024. [Online]. Available : <https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edfffb2143>
- [84] “Le tutoriel Python.” [Online]. Available : <https://docs.python.org/3/tutorial/index.html>
- [85] “Visual Studio Code : l’éditeur de code gratuit et complet de Microsoft.” [Online]. Available : <https://www.blogdumoderateur.com/tools/visual-studio-code/>
- [86] “NumPy : la bibliothèque Python la plus utilisée en Data Science.” [Online]. Available : <https://datascientest.com/numpy>
- [87] “Matplotlib — Visualization with Python.” [Online]. Available : <https://matplotlib.org/>
- [88] “Matplotlib — Visualization with Python.” [Online]. Available : <https://matplotlib.org/>
- [89] R. Kassel, “Keras : tout savoir sur l’API de Deep Learning,” Jun. 2021. [Online]. Available : <https://datascientest.com/keras>
- [90] “Computer Vision Annotation Formats.” [Online]. Available : <https://roboflow.com/formats>
- [91] J. Robert, “Kaggle : Tout ce qu’il faut savoir sur cette plateforme,” Sep. 2021. [Online]. Available : <https://datascientest.com/kaggle-tout-ce-quil-a-savoir-sur-cette-plateforme>
- [92] D. Eleyehou, “Google Colab : la force du cloud pour l’apprentissage automatique,” Feb. 2024. [Online]. Available : <https://datascientest.com/google-colab-tout-savoir>
- [93] R. Kassel, “Jupyter Notebook : Un outil indispensable en partage de code,” Mar. 2021. [Online]. Available : <https://datascientest.com/jupyter-notebook-tout-savoir>
- [94] —, “Flask – Un des frameworks les plus populaires de Python,” Nov. 2022. [Online]. Available : <https://datascientest.com/avantages-et-fonctionnement-de-flask>
- [95] “License Plate Detector Object Detection Dataset (v2, 2023-08-31 10 :01am) by Mochoye.” [Online]. Available : <https://universe.roboflow.com/mochoye/license-plate-detector-ogxxg/dataset/2>
- [96] “License Plate Detector Object Detection Dataset (v2, 2023-08-31 10 :01am) by Mochoye.” [Online]. Available : <https://universe.roboflow.com/mochoye/license-plate-detector-ogxxg/dataset/2>

- [97] SharkD, “RGB-Cube,” Mar. 2010. [Online]. Available : https://commons.wikimedia.org/wiki/File:RGB_Cube_Show_lowgamma_cutout_a.png?uselang=fr
- [98] —, “RGB-Cube,” Mar. 2010. [Online]. Available : https://commons.wikimedia.org/wiki/File:RGB_Cube_Show_lowgamma_cutout_a.png?uselang=fr
- [99] “Fichier :RGB Cube Show lowgamma cutout a.png — Wikilivres.” [Online]. Available : https://fr.m.wikibooks.org/wiki/Fichier:RGB_Cube_Show_lowgamma_cutout_a.png

Résumé

Le besoin d'un système de reconnaissance automatique des plaques d'immatriculation robuste et efficace se fait de plus en plus ressentir dans divers domaines tels que la gestion de la circulation et la sécurité routière. Ce mémoire présente un système dédié à la reconnaissance des plaques d'immatriculation algériennes.

Le système de reconnaissance des plaques d'immatriculation développé suit les trois étapes classiques du processus. La première étape consiste à extraire la plaque à partir de l'image d'origine du véhicule en utilisant le modèle de détection d'objets YOLO. Ensuite, la plaque extraite subit une amélioration par la détection des quatre coins suivie d'une transformation de perspective. Une fois cette étape complétée, la plaque est segmentée en caractères individuels.

Ces caractères sont ensuite analysés par un modèle de réseau de neurones convolutifs. Ce processus inclut un ensemble de techniques de traitement et de prétraitement d'image afin d'optimiser les résultats.

Mots-clés : ANPR, vision par ordinateur, CNN, YOLO, apprentissage profond, traitement d'image, OCR.

Abstract

The need for a robust and efficient automatic license plate recognition system is increasingly felt in various fields such as traffic management and road safety. This thesis presents a system dedicated to the recognition of Algerian license plates.

The developed License Plate Detection and Recognition system follows the three classic steps of the process. The first step involves extracting the license plate from the original vehicle image using the YOLO object detection model. Next, the extracted plate undergoes enhancement through the detection of its four corners followed by a perspective transformation. Once this step is completed, the plate is segmented into individual characters.

These characters are then analyzed by a Convolutional Neural Network. This process includes a set of image processing and preprocessing techniques to optimize the results.

Keywords : ANPR, computer vision, CNN, YOLO, deep learning, image processing, OCR.