



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Logiciel

Par :

BENREZGANE MOHAMED BOUHARKET

BIAZ ABDENACER

Sur le thème

Systeme d'Identification de Personnes via la Plaque d'Immatriculation de leurs Véhicules

Soutenu publiquement le 18/09/2022 à Tiaret devant le jury composé de :

| | | | |
|-------------------|-----|--------------------------------|-----------|
| Mr. BAKAR Khaled | MAA | Université Ibn Khaldoun Tiaret | Président |
| Mr. GAFOUR Yacine | MCB | Université Ibn Khaldoun Tiaret | Encadreur |
| Mr DJAFRI Laouni | MCA | Université Ibn Khaldoun Tiaret | Examineur |

2021 -2022

Remerciements

*Nous tenons tout d'abord à remercier DIEU le tout puissant et
Le miséricordieux, qui nous a donné la force
Et la patience d'accomplir ce modeste travail*

Un très grand merci à :

Nos parents qui nous ont suivis pendant nos études.

*En second lieu nous tenons à remercier notre encadreur
Mr. **GAFOUR Yacine** pour son aide, pour son encouragement,
Et pour ses précieux conseils durant la réalisation de ce travail.*

Nos vifs remerciements vont également aux membres du jury

*Mr. **BAKAR Khaled** Mr. **DJAFRI Laouni***

*Qui ont pris de leur temps pour juger ce modeste travail,
Qu'ils trouvent ici l'expression de notre gratitude et tout notre respect.*

*Nous adressons aussi nos remerciements à tous les professeurs qui
Nous avons enseigné durant ce cursus universitaire.*

*En fin, nous remercions Nos collègues de la promotion **2021-2022.***

Tout en leurs souhaitons un avenir plein de réussite.



Dédicaces

Je dédie ce modeste travail à :

- ❖ *À mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leurs prières tout au long de mes études*
- ❖ *A mes sœurs je les souhaite beaucoup de joie et de bonheur dans la vie.*
- ❖ *A tout ma famille.*
- ❖ *A tous mes amis et mes camarades.*
- ❖ *A tous ceux qui me sont cher .*

انتااا

BENREZGANE



Dédicaces

❖ إلى من أفصلها على نفسي، ولم لا؛ فلقد ضحّت من أجلي ولم تدّخر جهدًا في سبيل إسعادي

على الدوام (أمي الحبيبة).

❖ نسير في دروب الحياة، ويبقى من يُسيطر على أذهاننا في كل مسلك نسلكه صاحب الوجه

الطيب، والأفعال الحسنة. فلم يبخل عليّ طيلة حياته (والدي العزيز).

❖ إلى أصدقائي من بينهم (لؤلؤة)، وجميع من وقفوا بجواري وساعدوني بكل ما يملكون، وفي

أصعدة كثيرة أقدم لكم هذا البحث، وأتمنى أن يحوز على رضاكم....

❖ *Qui est-ce que je préfère à moi-même et pourquoi pas ? Tu t'es sacrifié pour moi Tu n'épargnes toujours aucun effort pour me rendre heureux (Chère mère).*

❖ *Nous marchons sur les chemins de la vie, et celui qui contrôle nos esprits reste sur chaque chemin que nous empruntons. Le propriétaire d'un bon visage et de bonnes actions. Il ne m'a pas gardé toute sa vie (Mon cher père).*

❖ *À mes amis parmi eux (Perle) et à tous ceux qui m'ont soutenu et aidé dans tout ce qu'ils ont et de tant de façons Je vous présente cette recherche, et j'espère qu'elle vous satisfera....*

BIAZ NACER

Table de matières

| | |
|---|----|
| Liste des tableaux | IV |
| Liste des figures | IV |
| Résumé | 1 |
| Mots-clés | 2 |
| Introduction générale..... | 4 |
| Chapitre1 : Généralités sur les images | 5 |
| I. Introduction..... | 6 |
| II. Définition d'une image | 6 |
| III. Image numérique (numérisée)..... | 6 |
| IV. Caractéristiques d'une image numérique | 7 |
| 1. Pixel | 7 |
| 2. Résolution..... | 7 |
| 3. Dimension..... | 7 |
| 4. Bruit | 8 |
| 5. Occupation mémoire d'un pixel | 8 |
| 6. Luminance | 8 |
| 7. Contours et textures..... | 8 |
| 8. Contraste | 9 |
| 9. Histogramme | 9 |
| V. Types d'images | 10 |
| 1. Images binaires (en noir et blanc) | 10 |
| 2. Image en niveaux de gris | 10 |
| 3. Images en couleurs (Polychromes) | 11 |
| VI. Système de traitement d'image | 11 |
| 1. Acquisition et numérisation | 12 |
| 2. Visualisation | 12 |
| VII. Traitement numérique des images | 12 |
| 1. Filtrage Numérique..... | 12 |
| 1.1 Filtres linéaires | 12 |
| 1.1.1 Filtre passe-bas (lissage)..... | 13 |
| 1.1.2 Filtre Passe-haut (Accentuation) | 13 |
| 1.2 Filtres non linéaires | 13 |
| 1.2.1 Filtre médian..... | 13 |

| | | |
|---|---|-----------|
| 1.2.2 | Filtre maximum | 14 |
| 1.2.3 | Filtre minimum | 14 |
| VIII. | Conclusion | 14 |
| Chapitre 2 : Apprentissage profond | | 15 |
| I. | Introduction..... | 16 |
| II. | Machine Learning..... | 16 |
| III. | Fonctionnement d'apprentissage profond | 16 |
| IV. | Analyse d'image..... | 16 |
| V. | Réseaux de neurones convolutifs..... | 17 |
| 1. | Réseau de neurones multi couches..... | 18 |
| 1-1 | Couches convolutives | 18 |
| 1-2 | Couches de non-linéarité | 18 |
| 1-3 | Couches de pooling | 19 |
| 1-4 | Couches entièrement connectées..... | 19 |
| VI. | Algorithme de détection d'objet | 19 |
| 1. | YOLO | 20 |
| 2. | Fonctionnement de YOLO | 20 |
| VII. | Reconnaissance optique de caractères (OCR)..... | 21 |
| 1. | Reconnaissance en ligne..... | 21 |
| 2. | Reconnaissance Hors ligne | 21 |
| 3. | Problèmes liés à l'OCR..... | 22 |
| VII. | Conclusion | 23 |
| Chapitre 3 : Système de reconnaissance des plaques d'immatriculation | | 24 |
| I. | Introduction..... | 25 |
| II. | Etat de l'art..... | 25 |
| 1. | Algorithmes du système LAPI | 25 |
| 2. | Difficultés des systèmes LAPI | 25 |
| 3. | Applications des systèmes LAPI..... | 25 |
| 4. | Différents systèmes existants..... | 25 |
| III. | Approche utilisée..... | 26 |
| IV. | Etiquetage des données | 27 |
| V. | Localisation de plaque de matricule..... | 28 |
| 1. | Cloner le référentiel darknet..... | 28 |
| 2. | Créer les fichiers de configuration | 29 |

| | | |
|-------|--|----|
| VI. | Segmentation des caractères | 31 |
| 1. | Isolement de la plaque | 31 |
| 2. | Conversion de l'image en niveau de gris..... | 31 |
| 3. | Conversion de l'image en binaire | 31 |
| 4. | Binaire inverse | 32 |
| 5. | Détection de contour | 32 |
| VII. | Identifier le propriétaire du véhicule | 32 |
| VIII. | Expérimentation et résultats..... | 33 |
| 1. | Environnement de programmation..... | 33 |
| 2. | Ressources matérielles utilisées..... | 33 |
| 3. | Discussion des résultats | 33 |
| IX. | Conclusion | 36 |
| | Conclusion générale | 37 |

Liste des tableaux

| | |
|--|----|
| Tableau 1 : Masque du filtre passe-bas..... | 13 |
| Tableau 2 : Masque du filtre passe-haut..... | 13 |
| Tableau 3 : Filtre median..... | 14 |
| Tableau 4 : Filtre maximum..... | 14 |
| Tableau 5 : Filter minimum. | 14 |
| Tableau 6 : les innovations et les faiblesses de YOLO..... | 21 |

Liste des figures

| | |
|---|----|
| Figure 1 : Image numérique. | 6 |
| Figure 2 : Groupe de pixel | 7 |
| Figure 3 : Résolution d'une image | 7 |
| Figure 4 : Image avec et sans bruit..... | 8 |
| Figure 5 : Contours d'une image | 9 |
| Figure 6 : Histogramme d'une image. | 10 |
| Figure 7: Image binaires | 10 |
| Figure 8 : Image en gris. | 11 |
| Figure 9 : Image en couleurs. | 11 |
| Figure 10 : Composition d'un système de traitement numérique..... | 12 |
| Figure 11 : Terminologies <i>hierarchies</i> de l'intelligence artificielle..... | 16 |
| Figure 12 : Architecture Alexnet..... | 17 |
| Figure 13 : Différents systèmes, représentations et approches de reconnaissance. | 22 |
| Figure 14 : Un neurone artificiel (ou formel). | 17 |
| Figure 15 : Architecture du réseau LeNet. | 18 |
| Figure 16 : Principe de la convolution. | 18 |
| Figure 17 : Fonctions d'activation ReLu et sigmoid..... | 19 |
| Figure 18 : Principe du max pooling. | 19 |
| Figure 19 : Architecture globale de modèle YOLO..... | 20 |
| Figure 20 : Dimension des outputs du modèle YOLO..... | 21 |
| Figure 21 : Présentation du système..... | 27 |
| Figure 22 : Étiquetage d'une image..... | 28 |
| Figure 23 : Fichier '.txt'. | 28 |
| Figure 24 : Darknet sur drive. | 29 |
| Figure 25 : Yolo4 sur drive. | 30 |
| Figure 26 : Localisation de plaque de matricule..... | 31 |
| Figure 27 : L'isolement de la plaque..... | 31 |
| Figure 28 : La plaque en niveau gris. | 31 |
| Figure 29 : La plaque en binaire. | 32 |
| Figure 30 : La plaque en binaire inverse..... | 32 |
| Figure 31 : Détection de contour. | 32 |
| Figure 32 : Contours de forme rectangulaire..... | 32 |

| | |
|--|----|
| Figure 33 : IV - Identifier le propriétaire du véhicule..... | 33 |
| Figure 34 : Precision, Recall et F1 score | 35 |
| Figure 35 : Précision-Recall graphe | 35 |

Résumé

Avec l'augmentation importante du nombre des véhicules qui circulent quotidiennement, le contrôle du trafic routier et l'identification des propriétaires de véhicules sont devenus un problème majeur dans le monde entier. Toutefois, il est devenu difficile de contrôler et vérifier ces véhicules dans les situations suivantes :

- Conducteur d'un véhicule qui excède la limite de vitesse autorisée ou viole le code de la circulation.
- Conducteur d'un véhicule qui fonce sur un barrage.
- Véhicules qui circulent avec de fausses plaques d'immatriculation.
- Véhicules volés toujours en situation de circulation.

En conséquence, il n'est pas évident d'identifier, traquer et rechercher de tels véhicules puisque le personnel de la circulation pourrait ne pas être en mesure de récupérer tous les numéros de véhicules roulants.

De ce fait, il est nécessaire de développer un système de détection et de reconnaissance automatique des véhicules en identifiant leurs numéros de plaque d'immatriculation.

Abstract

With the significant increase in the number of vehicles circulating daily, road traffic control and identification of vehicle owners has become a major problem worldwide. However, it has become difficult to control and verify these vehicles in the following situations:

- Driver of a vehicle that exceeds the authorized speed limit or breaks the rules of the road.
- Driver of a vehicle entering a roadblock.
- Vehicles driving with fake license plates.
- Stolen vehicles still in use.

As a result, it is not straightforward to identify, track and search for such vehicles since traffic personnel may not be able to retrieve all moving vehicle numbers.

Therefore, it is necessary to develop a system for automatic detection and recognition of vehicles by identifying their license plate numbers.

ملخص

إن إرتفاع عدد المركبات بشكل يومي جعل التحكم في سيرها وتحديد مالكيها مشكلة رئيسية في جميع أنحاء العالم، فقد أصبح من الصعب التحقق منها في المواقع التالية:

- سائق مركبة تجاوز السرعة المسموح بها أو يخالف قانون المرور.
- اصطدام سائق المركبة بالحاجز.
- قيادة المركبات التي تحمل لوحات ترخيص مزورة.
- السيارات المسروقة ما زالت متداولة.

نتيجة لذلك، ليس من السهل تحديد هذه المركبات وتتبعها والبحث عنها لأن أفراد المرور قد لا يتمكنون من استرداد جميع أرقام المركبات لذلك، من الضروري تطوير نظام للكشف والتعرف التلقائي على المركبات من خلال تحديد أرقام لوحات الترخيص الخاصة بها

Mots-clés

Identification de personnes, plaque immatriculation, yolo, détection d'objet, apprentissage profond, reconnaissance optique caractère.

Keywords

People's identification, license plate, yolo, object detection, deep learning, optical character recognition

Introduction Générale

Introduction générale

Dans nos jours, le stockage de grands volumes de données (Big Data) est devenu possible et abordable. Parallèlement, la puissance de calcul des microprocesseurs a décuplé, et les appareils photos numériques sont devenus extrêmement performants pour un coût de plus en plus bas. Ainsi, depuis que le traitement des données vidéo en temps réel est devenu sérieusement envisageable, pour des problématiques aussi diverses que l'analyse statistique de la fréquentation d'un lieu, la sécurisation de l'accès à des bâtiments, ou encore la surveillance routière.

Le numéro d'immatriculation représente un moyen efficace pour identifier les véhicules et donc d'identifier leurs propriétaires. Il s'agit d'une information unique pour chaque véhicule. Généralement, il est nécessaire d'identifier les plaques d'immatriculation des véhicules pour la sécurité. Les informations extraites peuvent être utilisées pour plusieurs intérêts, comme le contrôle d'accès et de flux, la surveillance des passages aux frontières et aux péages, la recherche de véhicules suspects ou encore la lutte contre la criminalité. Cependant, trouver le numéro de plaques d'immatriculation dans des vidéos ou des images pose un réel défi pour les humains en raison de la quantité énorme d'informations qui nécessite des efforts et un grand nombre de personnes, qui se prépare à une intervention automatique pour résoudre ce problème.

La localisation et la lecture automatique de caractères sur une image enregistrée dans des mauvaises conditions (capture rapide et de faible luminosité) c'est le vrai problème pour identifier des plaques d'immatriculation. De plus, il y a très peu d'images haute résolution sur la plupart des vidéos car une existence de bruit lors de la capture des images. Pour cela, il faut procéder à l'avance un prétraitement de ces images pour permettre une détection et bonne reconnaissance des caractères.

Ce que nous visons à faire par notre travail est de faciliter la tâche de localisation du matricule en exploitant les bibliothèques que peut offrir le traitement d'image. Pour cela, nous avons développé une application qui consiste à l'identification de personnes via la plaque d'immatriculation de leurs véhicules.

Nous avons décomposé notre mémoire comme suit :

Le premier chapitre, nous commencerons par généralités sur les images, présentant les caractéristiques et les types d'une image. Nous finirons ce chapitre par quelques exemples de traitement des images.

Dans le deuxième chapitre, nous présentons une étude sur l'apprentissage profond et les notions de base nécessaires à la compréhension de ces techniques.

Le troisième et le dernier chapitre, nous présenterons l'implémentation détaillée de l'application en expliquant les étapes de réalisation de notre système.

Nous concluons ce mémoire par une conclusion générale et des perspectives.

Chapitre 1 : Généralités sur les images

I. Introduction

Afin d'extraire des informations les plus importantes contenues dans les images, nous appliquons un ensemble de techniques et de méthodes connues sous le nom de traitement d'images. Le traitement d'images permet d'analyser, visualiser, interpréter, stocker les images afin de les utiliser dans différents domaines scientifiques (physique, biologie, médecine, astronomie...).

II. Définition d'une image

L'image est une représentation d'un objet vivant ou concept, réels ou métaphore d'une manière naturelle (ombre, reflet) ou artificielle (sculpture, peinture, photographie) qui est inexploitable par la machine, ce qui nécessite de structurer sous la forme d'une fonction "I (x, y)" de brillance analogique continue dans un domaine borné. Les x et y sont des coordonnées d'un point de l'image et le "I" une fonction d'intensité lumineuse et de couleurs [1].

III. Image numérique (numérisée)

L'appellation d'image numérique désigne toute image qui peut être exploitée par des applications sur un ordinateur. Capturé par des appareils photos, les scanners ou les caméscopes numériques ou peint grâce à hardware comme un stylo électronique, souris etc. La numérisation d'une image est la conversion de celle-ci représentée par une matrice bidimensionnelle de série de bits, ayant chacun comme caractéristique de couleurs ou un niveau de gris sous forme de tableau I de n lignes et p colonnes, stockée sur un support physique (HDD, USB, CD, DVD etc...) [1].

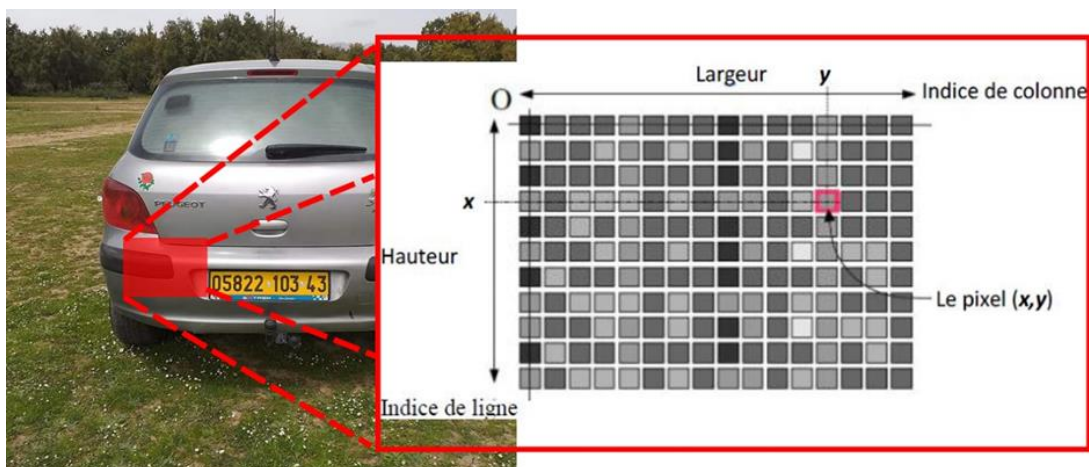


Figure 1: Image numérique.

IV. Caractéristiques d'une image numérique

1. Pixel

Le pixel abréviation du terme anglaise Picture Element, c'est la plus petite unité que les appareils peuvent manipuler pour traiter les images [2].

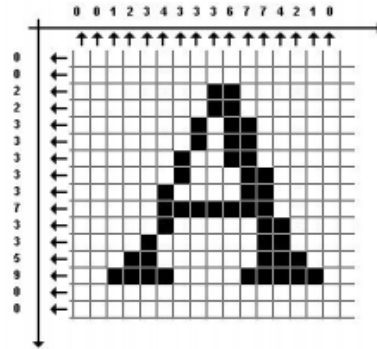


Figure 2 : Groupe de pixel [2].

2. Résolution

La résolution liée au nombre de pixels par unité en dpi (dots par *inch*) ou en ppp (points par pouce). Plus l'unité contient des pixels, plus l'image et la quantité d'informations qu'elle contient sont précises, plus la résolution est élevée, par exemple, 300 dpi signifiés que largeur et la hauteur d'image sont 300 pixels [3].

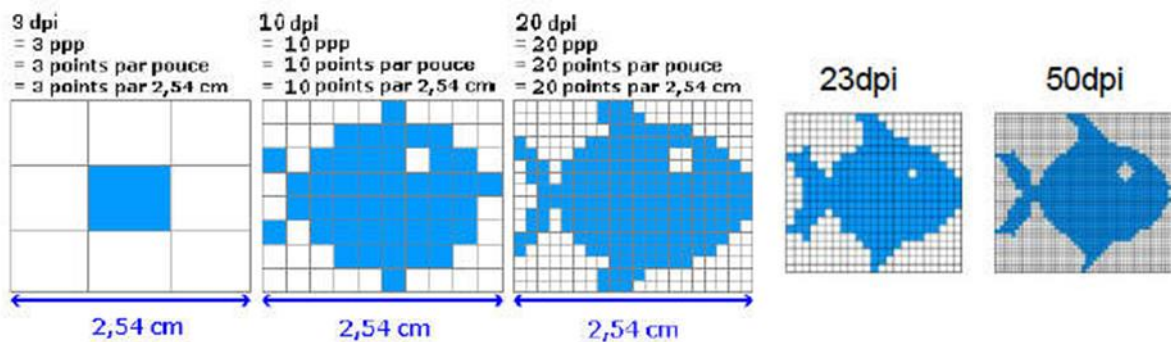


Figure 3: Résolution d'une image [33].

3. Dimension

La dimension (ou la taille) d'une image est la multiplication de nombre de lignes par le nombre de colonnes de cette grille de pixels (matrice), qui nous donne le nombre total de pixels dans cette image [3].

4. Bruit

Le bruit est toute dégradation subit la qualité d'image et perd de détail ou fluctuation d'un pixel est visible sous forme couleur aléatoire ou plus claires par rapport à ses voisins. Les sources de parasite sont la qualité de l'éclairage, la température et la stabilité du capteur [4].



Figure 4:Image avec et sans bruit.

5. Occupation mémoire d'un pixel

- En noir et blanc : un pixel correspond à un bit.
- En 16 couleurs (standard VGA) : un pixel correspond à 4 bits.
- En 256 couleurs : un pixel correspond à 8 bits (soit 1 octet).
- En 65.536 couleurs : un pixel correspond à 16 bits.
- En mode 16 millions de couleurs : un pixel correspond à 24 bits [3].

6. Luminance

La luminosité est le degré de l'intensité lumineuse d'une surface par source de lumière. En photographie, elle est définie comme la puissance de blanc du point d'image. Plus un pixel contient du blanc, plus est lumineuse. L'absence de parasites est une conséquence d'une bonne luminance [5].

7. Contours et textures

Les contours représentent les points dans l'image qui séparent deux régions (textures) différentes, plutôt un ensemble des pixels ordonné sous forme la limitation dont les niveaux de gris présentent une information globale significative [6].

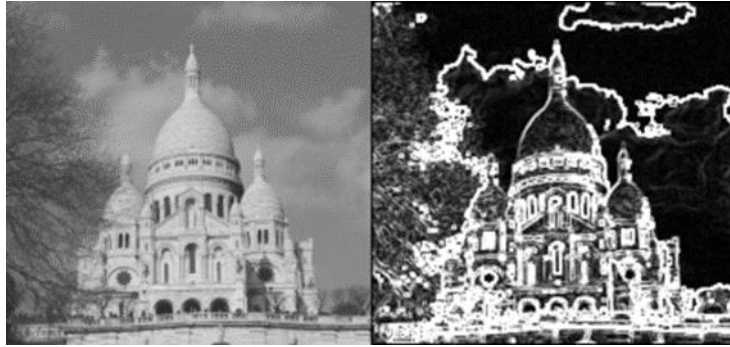


Figure 5: Contours d'une image [34].

8. Contraste

Le contraste représente la différence entre les régions les plus noires (sombres) et les plus blanches (claires), il indique l'opposition de densité lumineuse d'une deux zones approchantes. Un contraste nul pour une image grise, et un contraste élevé pour image en couleurs.

$$C = \frac{L1-L2}{L1+L2}$$

L1, L2 : degré de luminosité de deux zones voisines.

C : le contraste [4].

9. Histogramme

L'histogramme est une fonction permet de représenter la fréquence d'apparition des pixels pour chaque niveau de couleur ou niveau de gris. Il est utilisé pour donne la distribution de l'intensité lumineuse. [1].

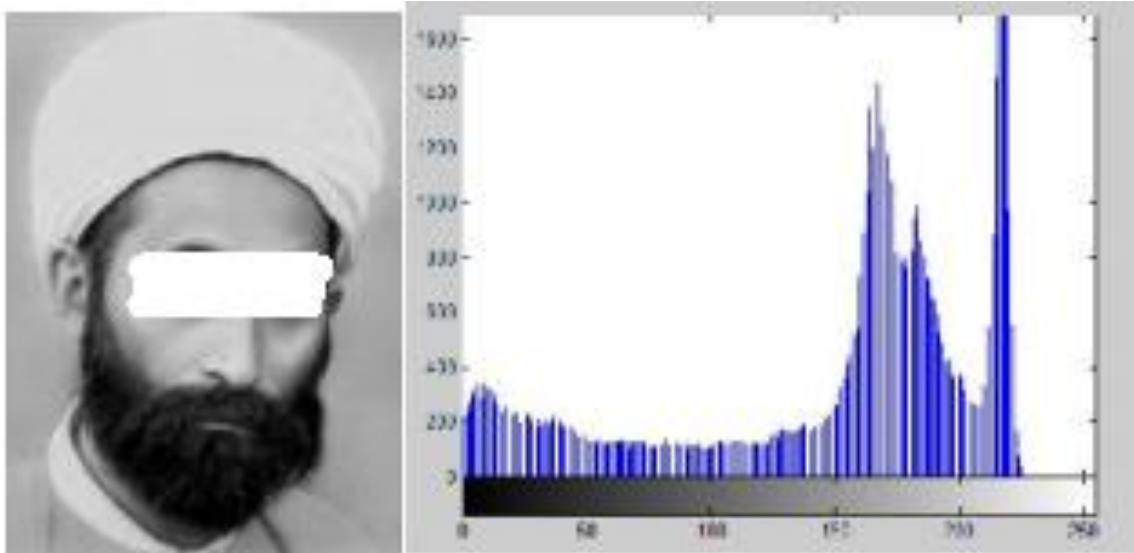


Figure 6 : Histogramme d'une image [1].

V. Types d'images

1. Images binaires (en noir et blanc)

L'image binaire sa forme matricielle stockée que les 0 et 1 (noire et blanche) chaque pixel correspond à un bit. Nous pouvons directement coder ou décoder vers la base 2 [1].

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 7: Image binaires [35].

2. Image en niveaux de gris

Dans une image simple, chaque pixel peut prendre une valeur de l'intensité lumineuse entre 0 (noir) et 255 (blanc), la valeur correspondant au niveau de gris [1].



Figure 8: Image en gris.

3. Images en couleurs (Polychromes)

Dans les images couleur le pixel est composé a 3 valeurs RVB, chaque couleur prend 256 valeurs (de 0 jusqu'à 255), $R = V = B = 0$ pour le noir et $R = V = B = 255$ pour le blanc [4].



Figure 9: Image en couleurs.

VI. Système de traitement d'image

Le traitement d'images est un terme informatique qui utilise opérations mathématiques ou des algorithmes pour reconnaissance, améliorer ou extraire de l'information [5].

Un système de traitement d'image compose des unités suivantes :

- Unités d'acquisition.
- Mémoire de masse.
- Unités visualisation.
- Unité centrale.

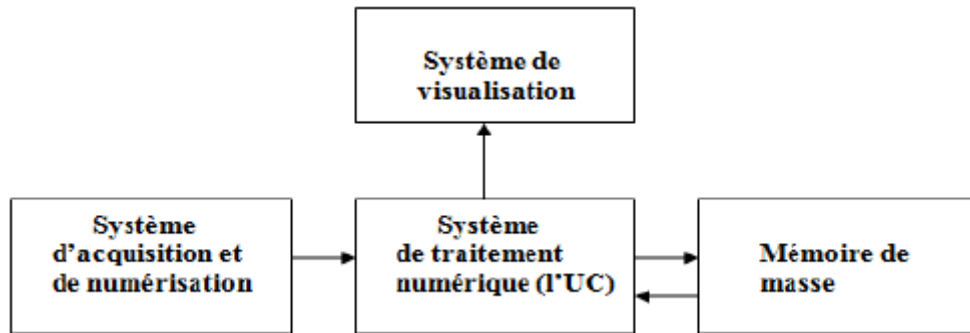


Figure 10 : Composition d'un système de traitement numérique [5].

1. Acquisition et numérisation

L'acquisition d'images est la première étape dans le traitement d'images qui consiste à récupérer l'image à partir d'une source matérielle. Pour traiter ces images nécessite de numériser c'est-à-dire convertir l'objet réel par représentation électronique (matrice) [7].

2. Visualisation

La visualisation est la transformation d'image numérique en un signal visuel analogique de l'œil de l'observateur et affiche sur les différents types de moniteur, programme de visualisation reproduire chaque pixel pour créer une image proche de l'image réelle [8].

VII. Traitement numérique des images

Le traitement d'images numériques utilise un ensemble des techniques qui permet d'extraire des données pour augmenter la qualité des images, extraire des caractéristiques, transformer des images [7].

1. Filtrage Numérique

Les images numériques contiennent du problème tel que les tâches, contours flous et points parasites donc sont très inutilisables. Pour cela il faut effectués des prétraitements (filtres) qui permet de réduire les bruits, éliminés les parasites et définis les contours [4]. On peut diviser les filtres en deux catégories :

1.1 Filtres linéaires

Les filtres linéaires multiple les données d'entrée par un facteur. La sortie est une combinaison linéaire de valeur voisinage. Cette opération appelée fonction de transfert [4]. Les filtres linéaires se divisent en deux.

1.1.1 Filtre passe-bas (lissage)

Le filtre passe-bas permet de retenir les informations à basse fréquence dans une image tout en réduisant les informations à haute fréquence. Une image est lissée en diminuant la disparité entre les valeurs de pixel en faisant la moyenne des pixels voisins. Un filtre passe-bas est la base de la plupart des méthodes de lissage [5].

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Tableau 1 : Masque du filtre passe-bas [5].

1.1.2 Filtre Passe-haut (Accentuation)

Le filtre passe-haut permet de laisser les hautes intensités de l'image et bloquer les basses fréquences, c'est l'inverse de passe-bas. L'utilisation de ce filtre permet de détecter des contours dans une image [5].

| | | |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 9 | -1 |
| -1 | -1 | -1 |

Tableau 2 Masque du filtre passe-haut [5].

1.2 Filtres non linéaires

Les filtres non linéaires remplacent les données d'entrée par des valeurs qui ne sont pas une combinaison linéaire. Ce type de filtre conçu pour couvrir les points faibles intensités. Les filtres non linéaires sont : [5].

1.2.1 Filtre médian

Le filtre médian calcule la valeur médiane des voisinages pour chaque pixel. Il est utilisé pour détecter le parasite et conservant les contours [5]. La procédure de filtre médian est le suivant :

- Ordonner les valeurs par ordre décroissant.
- Remplacer la valeur du pixel centrale par la valeur médiane de la triée.
- Répéter cette opération pour toute l'image.

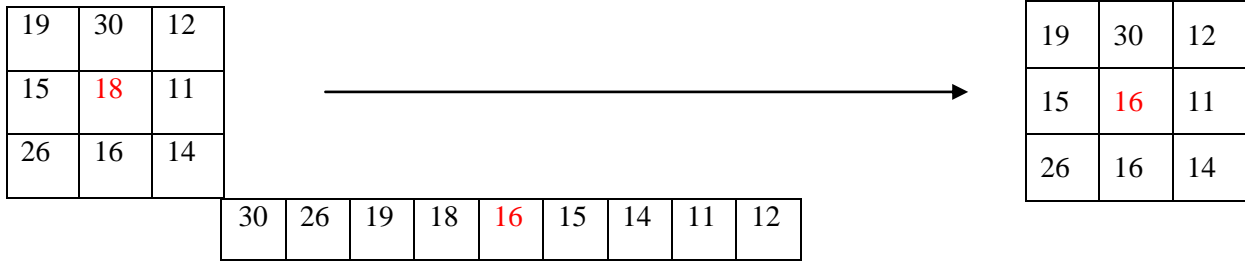


Tableau 3: Filter median [5].

1.2.2 Filtre maximum

Pour le filtre maximum, nous appliquons la même procédure de filtre médian mais en remplaçant la valeur médiane par le maximum [5].

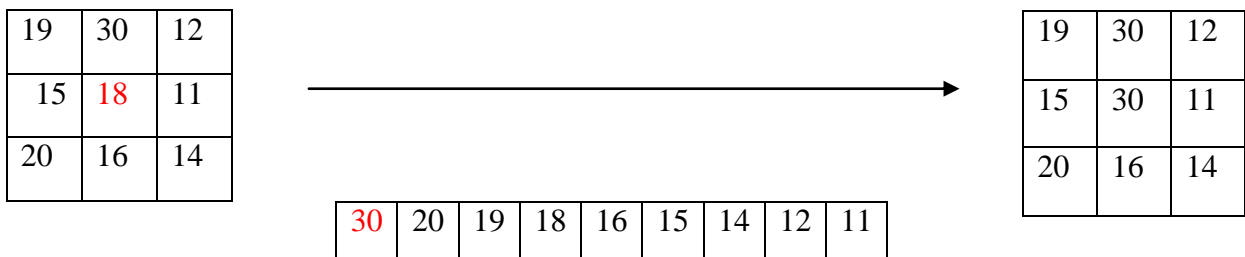


Tableau 4 : Filtre maximum [5].

1.2.3 Filtre minimum

Pour le filtre minimum, nous appliquons la même procédure de filtre médian mais en remplaçant la valeur médiane par le minimum [5].

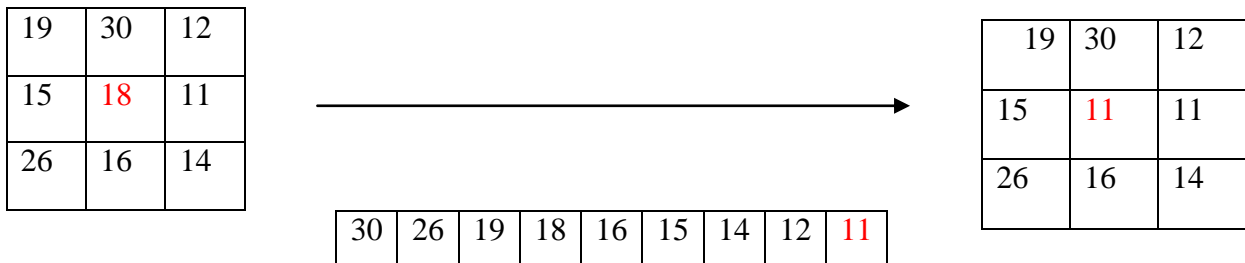


Tableau 5: Filter minimum [5].

VIII. Conclusion

Nous avons montré dans ce chapitre la notion de base qui représente l'image, ses types et les différentes techniques de traitement d'images. Les images sont un objet bidimensionnel représentant une espace tridimensionnelle qui le rend plus complexe puisque le volume des données à traiter est massif et leur structure est plus complexe. Ainsi que, nous avons présenté quelques caractéristiques des images et des méthodes de traitement afin d'améliorer la qualité des images. Par conséquent, les techniques de traitement d'image sont développées et améliorées pour détecter et reconnaître les formes dans une image.

Chapitre 2 : Apprentissage profond

I. Introduction

L'apprentissage profond (Deep Learning) est un type d'intelligence artificielle et un sous-domaine de l'apprentissage automatique. Son architecture s'appuie sur les réseaux de neurones à multiples couches capables d'apprendre par lui-même imitant les cerveaux humains. L'apprentissage profond donne un sens aux informations qui reçoivent et interprètent de couche courante a la couche précédente [9].

II. Machine Learning

Le Machine Learning est un ensemble des algorithmes avoir l'habilitier à détecter des patterns dans un flux de données et prendre décision ou prédicat en se basant sur des statistiques [10].

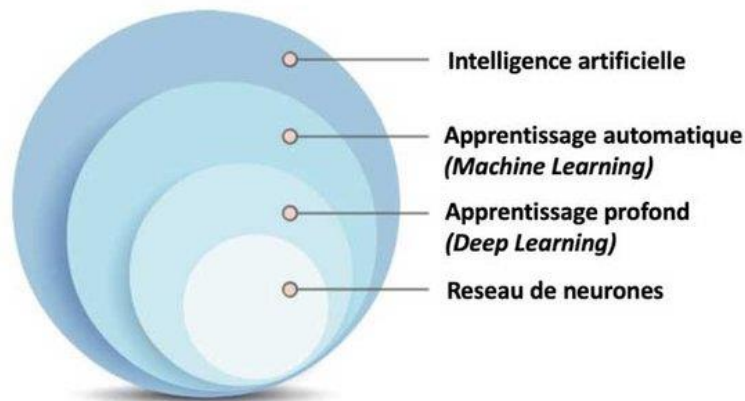


Figure 11 : Terminologies hiérarchies de l'intelligence artificielle.

III. Fonctionnement d'apprentissage profond

Deep learning entraîne les réseaux de neurones en introduisant des données complexes à la couche d'entrée puis les neurones intermédiaires traitent ces données. Les résultats obtenus sont transmis à la couche de sortie. Plus le réseau contient plusieurs de neurones (couche) plus il effectuer des tâches complexes [10].

IV. Analyse d'image

Un micro-ordinateur utilise des bits de bas niveau (0, 1) afin de représenter les informations. Deep learning utilise un outil mathématique appelé convolution pour traiter l'image selon la taille de filtre de convolution choisi. Le nombre de convolution est utilisé en faveur de détecter des caractéristiques importantes dans une image. L'augmentation du

nombre de couches de convolution permet de découvrir caractéristiques plus de plus complexe. Par exemple, Alexnet est réseau de neurones proposé en 2012, est composé de 8 couches, 5 convolutifs et 3 vecteurs classiques [11].

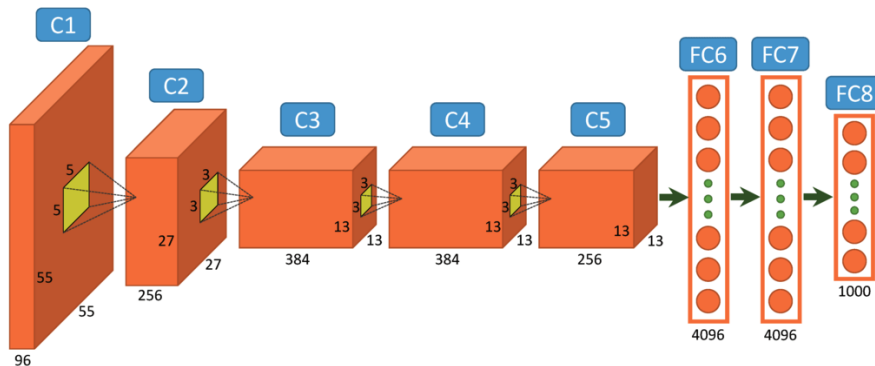


Figure 12: Architecture Alexnet [36].

V. Réseaux de neurones convolutifs

Un neurone artificiel est le cœur mathématique de l'apprentissage profond, il s'adapte parfaitement aux données avec un grand nombre de fonctionnalités de manière analytique. Les neurones associées l'entrée x avec sortie $h(x)$ par la multiplication des données d'entrées par le poids. Un neurone contient au moins deux couches, les autres couches appelées des couches cachées [12].

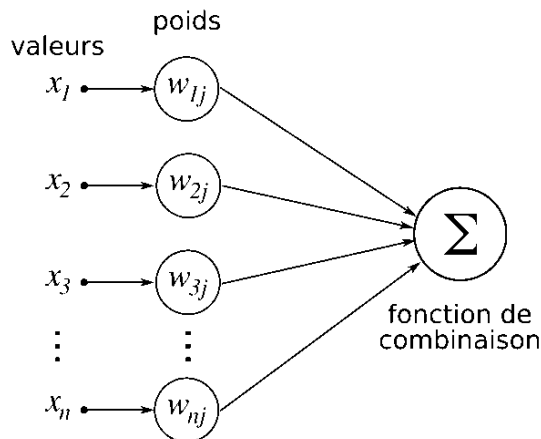


Figure 13 : Un neurone artificiel (ou formel).

Les réseaux de neurones convolutifs sont des nœuds adaptés au traitement d'image ou les données matricielles [9]. Plutôt que de la multiplication de chaque pixel par un chiffre, se balader toute la grille (image) à travers un filtre set idée obtenir à travers les travaux de Hubel et Wiesel [13]. Le Net est un exemple de réseaux de neurones convolutifs [14].

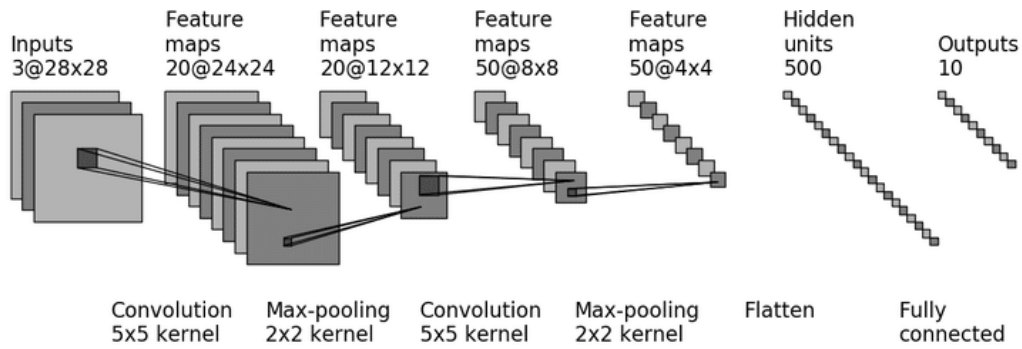


Figure 14 : Architecture du réseau LeNet.

1. Réseau de neurones multi couches

Les architectes de réseaux neurones permettent l'utilisation de convolution pour extraire des caractéristiques dans une forme simple ou complexes [15]. Les différentes couches de réseau de neurones convolutif sont :

1-1 Couches convolutives

Les couches convolutives utilisent une fenêtre placée tout en haut à gauche afin de parcourir la matrice jusqu'en bas à droite. Cette fenêtre se déplace selon la taille de filtre qui peut s'appliquer au n'importe quelle opération conforme avec le problème traité [15].

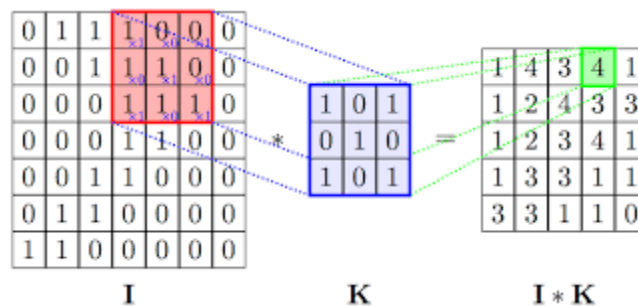


Figure 15 Principe de la convolution.

1-2 Couches de non-linéarité

Les couches non-linéaires dans une architecture CNN permettent de multiplier des données de sortie par une fonction d'activation généralement Sigmoid ou ReLu [15].

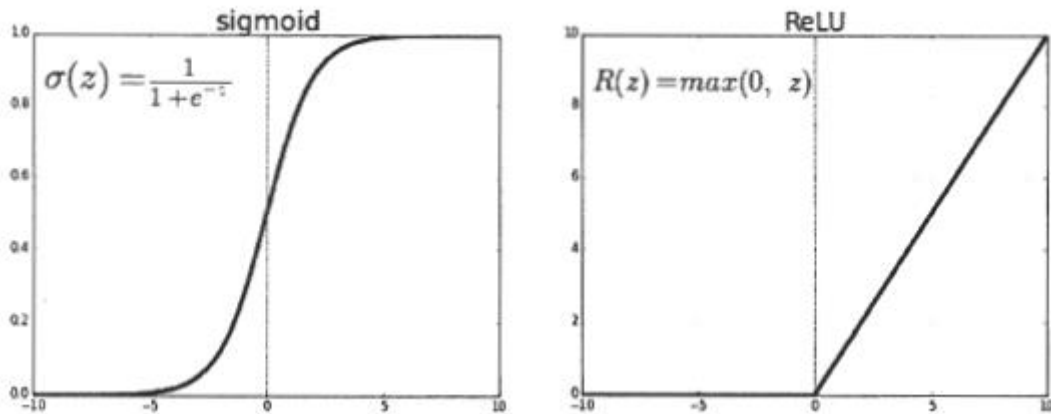


Figure 16 : Fonctions d'activation ReLu et sigmoid.

1-3 Couches de pooling

Pooling couche est une couche placée entre deux couches de convolutions avec un but de minimiser les données de sortie en maintenant les informations importantes, parmi les techniques de pooling prendre le maximum ou la moyenne [17].

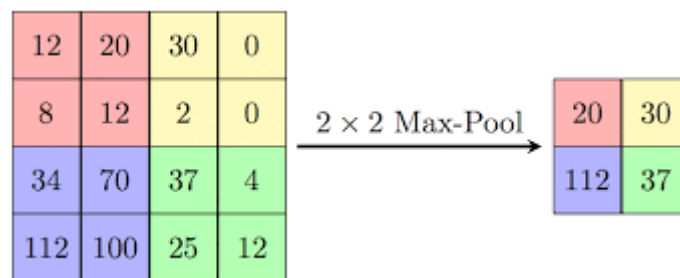


Figure 17 : Principe du max pooling.

1-4 Couches entièrement connectées

Les couches entièrement connectées dans un réseau de neurones sont les couches où toutes les entrées d'une couche sont connectées à chaque fonction d'activation de la couche suivante. Dans les modèles d'apprentissage automatique, les dernières couches sont des couches entièrement connectées qui utilisent les données extraites par les couches précédentes pour former la sortie finale [14].

VI. Algorithme de détection d'objet

La détection des objets dans une image est une approche difficile dans le domaine de la vision par ordinateur. En effet, les technologies font référence aux algorithmes qui extraient des caractéristiques afin de déterminer la localisation des objets dans une image [18].

1. YOLO

YOLO (You Only Look Once) est un modèle à une étape pour détection d'objet, il est inventé par Joseph Redmon, Ross, Girshick, Santosh Divvala et Ali Farhadi en 2015. Le principe d'algorithme est de parcourir l'image qu'une seule fois à travers un réseau de neurones qui soustrait directement les box rectangulaires et les labels de classe pour chaque photo [19]. La figure 19 montre l'architecture du modèle YOLO.

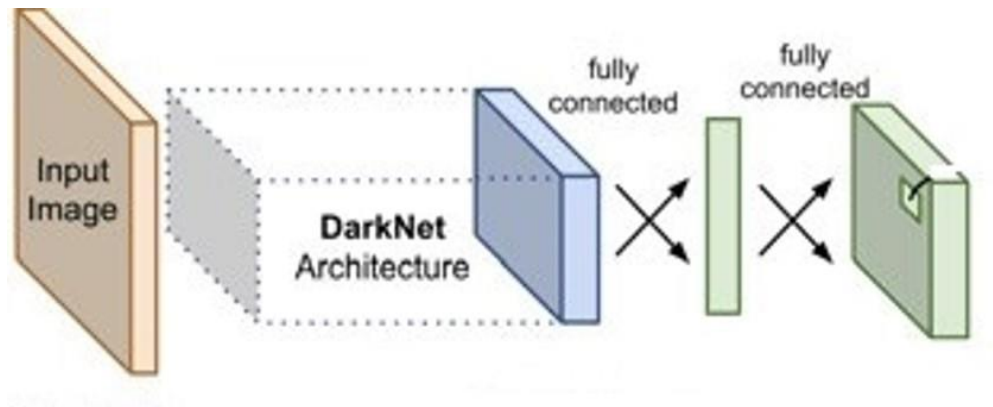


Figure 18 : Architecture globale de modèle YOLO

2. Fonctionnement de YOLO

Premièrement, le modèle minimise la taille d'image d'entrée par la division de sa dimension en une grille 3×3 par exemple. Les zones d'intéressantes sous forme un rectangle sont détectées en utilisant un nombre de convolutions et pooling sur chaque grille générée.

Deuxièmement, un vecteur Y ($pc, bx, by, bh, bw, c1, c2...$) est calculé pour chaque zone d'intéresse

- "pc" est la probabilité d'existence d'objet,
- "bx, by" les coordonnées de centre de box,
- "bh, bw" sont hauteur et largeur de box,
- "c" correspondant le nombre de classe,

Les zone d'intéresse qui ont une faible probabilité par rapport une valeur déjà défini sont supprimées [19].

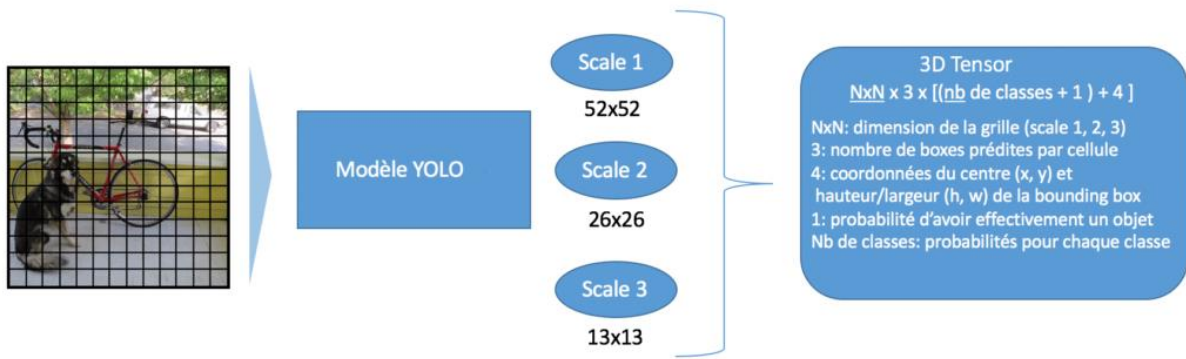


Figure 19 : Dimension des outputs du modèle YOLO.

| Algorithme | Innovations | Limitations |
|------------|---|--|
| YOLO | Utilise un nouveau réseau pour extraire les caractéristiques, le Darknet53. | La précision de détection s'améliore mais reste toujours inférieure par rapport aux algorithmes à deux étapes. |

Tableau 6 : les innovations et les faiblesses de YOLO.

VII. Reconnaissance optique de caractères (OCR)

Le système de Reconnaissance Optique de Caractères (OCR) permet de reconnaître les caractères des mots imprimés, les données textuelles complexes. Parmi les modes de classification peuvent citer :

1. Reconnaissance en ligne

La reconnaissance en ligne permet de récupérer et reconnaître les symboles en temps réel. Généralement, cette reconnaissance concerne à l'écriture manuscrite, les données bidimensionnelles (image) sont reconnues comme ensemble du point [20], [21]. Le typage corrigé et modifié de manière conversationnel car la réponse continue du système [22].

2. Reconnaissance Hors ligne

Ce mode fonctionne après l'acquisition du texte convient aux documents imprimés ou des images, la source de génération indépendante de l'interprétation de l'information [23]. Il existe plusieurs types de reconnaissance hors-ligne ainsi que la reconnaissance de l'imprimé, les manuscrits, le texte ou analyse de documents [24], [25].

3. Problèmes liés à l'OCR

Les problèmes qui compliquent la tâche de la reconnaissance ce sont les mauvais documents ou images sont qualité dégradée peut perd des parties du texte, pendant la capture d'images ou l'impression peut introduire des tâches masquer les caractères, les différents styles de font change le graphisme qui présente le caractère que complique la reconnaissance [21], [26].

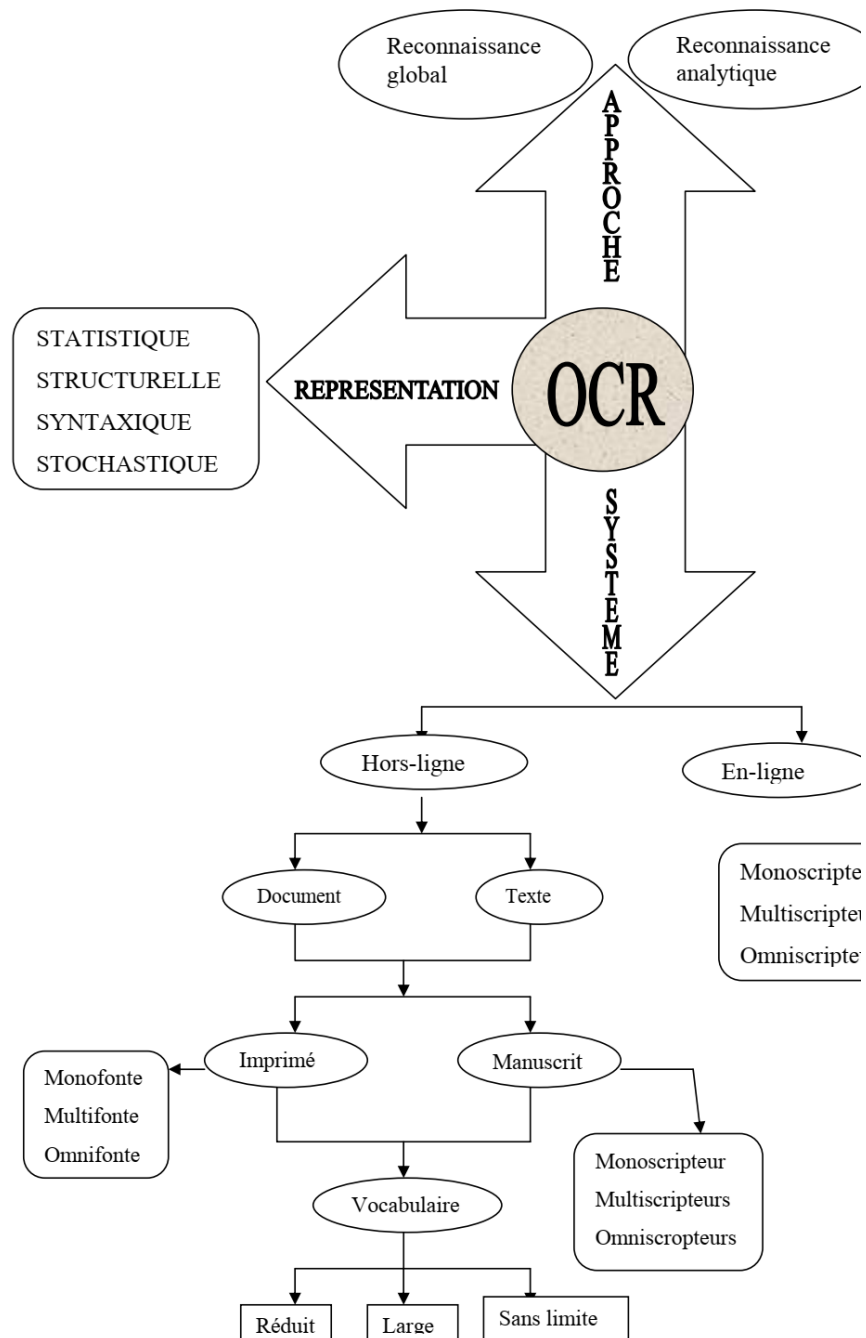


Figure 20 : Différents systèmes, représentations et approches de reconnaissance [16].

VII. Conclusion

Dans ce chapitre, nous avons présenté une brève introduction aux concepts liés à l'apprentissage profond. Ensuite nous avons défini comment analyser les images et les textes, nous présenter l'OCR et les problématiques liées à la reconnaissance de texte, enfin définir l'algorithme de détection YOLO et son fonctionnement. Bien que ce soit un détecteur d'objet à une étape qui offre le meilleur compromis entre la précision et la rapidité.

Chapitre 3 : Système de
reconnaissance des plaques
d'immatriculation

I. Introduction

Localisation automatique de plaque immatriculation (LAPI) est une technologie à la capacité de reconnaissance automatisée des plaques d'immatriculation basée sur l'exploitation des techniques de traitement d'image et texte (OCR). Un système LAPI devient une nécessité dans les différents aspects de vie dans le but de la protection, la sécurité routière et le suivi de criminels.

II. Etat de l'art

1. Algorithmes du système LAPI

La procédure de reconnaissance de plaque immatriculation presque c'est la même pour tous les applications LAPI avec une spécification selon le type de plaque (taille, couleur, design et numérotation...). En premier lieu, une caméra autonome capture le véhicule, puis logiciel détecte et localise la plaque sur l'image a travers un algorithme de détection d'objet, ensuite en applique des prétraitements a la plaque afin de normalisation, après en passe d'utilisation d'OCR pour que segmenter et reconnaître les caractères.

2. Difficultés des systèmes LAPI

Les difficultés auxquelles le système peut être confronté sont une mauvaise résolution, des images floues, l'existence de plusieurs fonts, invisible partait de la plaque...

3. Applications des systèmes LAPI

La lecture automatique de plaques immatriculations est utilisée pour l'identification de personnes, le contrôle de la circulation, gérer l'accès au parking, détecter de fausses plaques...

4. Différents systèmes existants

Les différents systèmes de la reconnaissance de plaques d'immatriculation existants sont :

1. AutoVu

Système de reconnaissance de plaques d'immatriculation et localiser les véhicules sur la plate-forme IP Security Center de Genetec [27].

2. LAPI ENGINE

Le produit LAPI ENGINE permet la lecture automatique de plaques d'immatriculation. Ce produit est dédié à le suivre de véhicules, il est adapté avec large application comme des applications de vidéosurveillance, de gestion péage [28].

3. SeeTec

Société fournie logicielle de sécurité et gestion en Europe [29].

4. VECON-VIS

Le produit d'Asia Vision, est un système de reconnaissance des plaques d'immatriculation polyvalent qui peut être appliqué dans divers domaines d'activité [30].

5. AGL (Application de Gestion LAPI)

Technologie de but d'automatiser les travaux de force de l'ordre utilisant des capteurs dédiés à la verbalisation électronique. [31].

6. Système LAPI-Pryncar

Systeme développée en France de but de contrôler d'accès ou d'une recherche de véhicule. [32].

III. Approche utilisée

Nous présenterons un système d'identification de personnes via la plaque d'immatriculation de leurs véhicules. Comme tout LAPI nous allons passer par des étapes. En premier lieu extraire à partir d'une image la plaque du véhicule en baser fondamentalement sur un algorithme de détection d'objet (YOLO). Le système passe par une collection de prétraitement comme les binarisation d'image, détection des contours afin d'arriver à un préférable résultat le plus possible, Cette sous-image passe par une segmentation en caractères afin d'être analysé par un système de reconnaissance optique de caractère (Tesseract), ensuite compare le string avec la BDD pour identifier le propriétaire du véhicule.

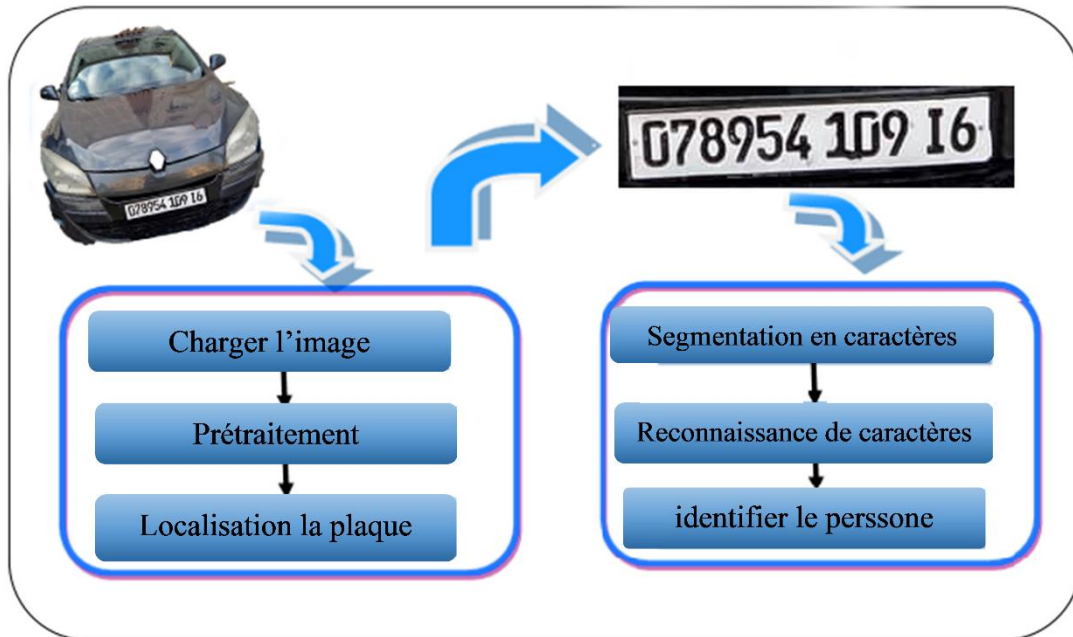


Figure 21 : Présentation du notre système de reconnaissance des plaques d'immatriculation.

IV. Etiquetage des données

- Télécharger label Img (<https://tzutalin.github.io/labelImg/>).
- Modifier fichier 'predefined_classes' (ajoutez le nom de votre classe "license_plate").
- Sélectionner une image de la BDD 'CreateRectBox'.
- Sélectionner des objets qui sont présentés dans l'image.
- Sélectionner la classe de l'objet.
- Télécharger (Upload) le dataset sur Google Drive.

Remarque : vérifier la génération des fichiers texte pour chaque image.

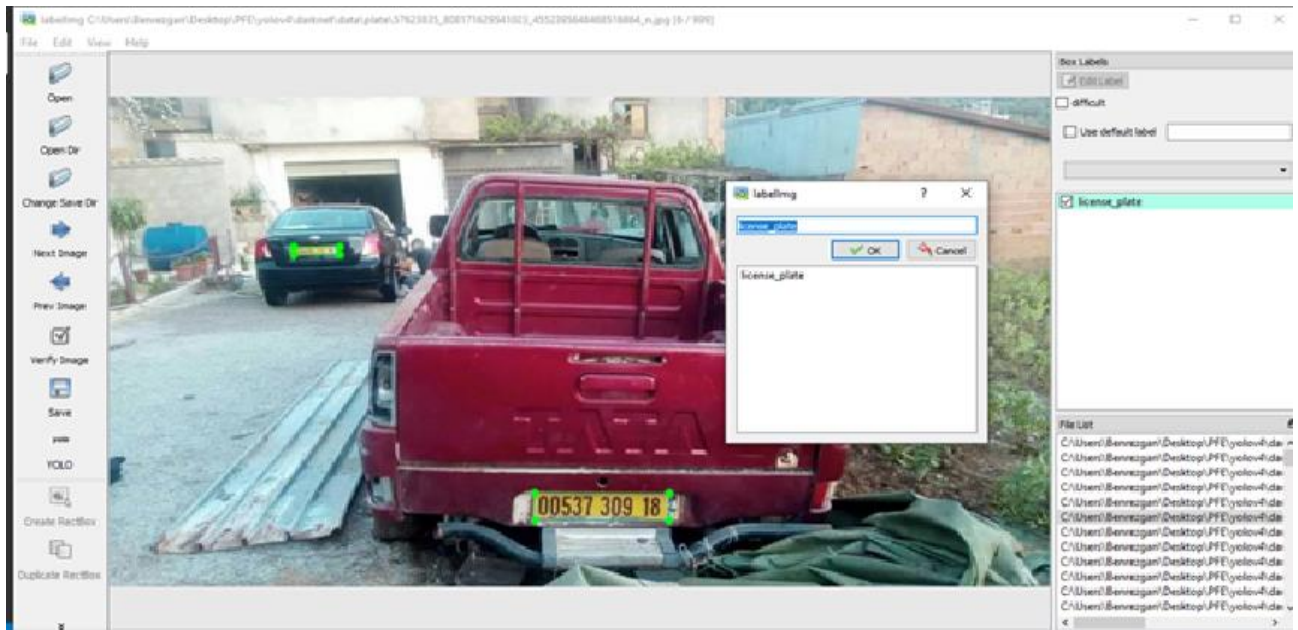


Figure 22: Étiquetage d'une image.

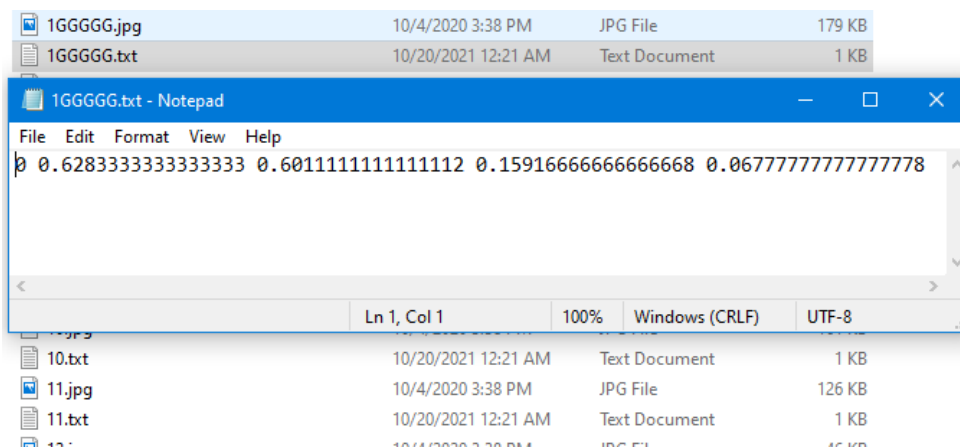


Figure 23: Fichier '.txt'.

V. Localisation de plaque de matricule

La localisation de la plaque est la phase la plus importante et la plus complexe. De ce fait, entraîner un modèle utilisons la base de données dans le Cloud en utilisant l'algorithme YOLO. Ce modèle permet de localiser la plaque de matricule dans une image.

1. Cloner le référentiel darknet

Cloner (copier)le référentiel darknet dans le dossier yolov4 sur notre drive <https://github.com/AlexeyAB/darknet> .

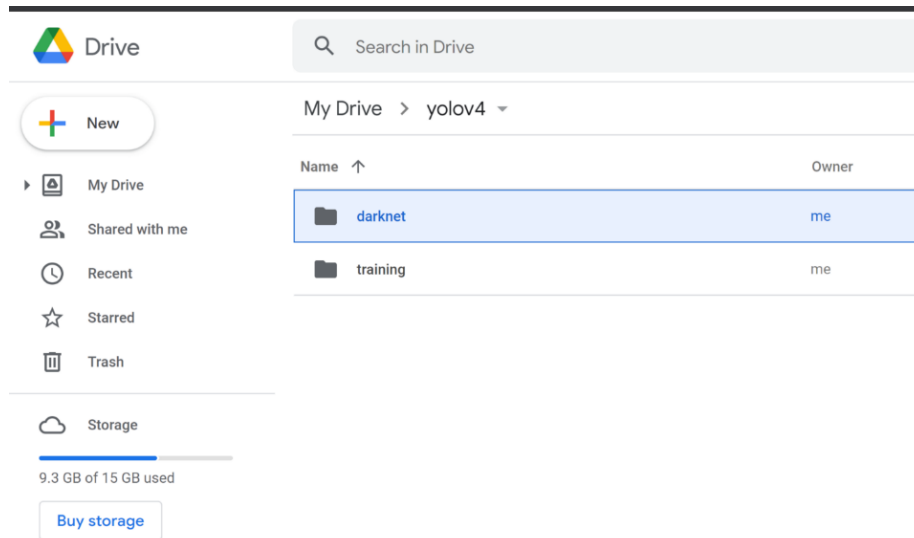


Figure 24 : Darknet sur drive.

2. Créer les fichiers de configuration

1. Fichier *.cfg personnalisé

- Télécharger le fichier yolov4-custom.cfg à partir du répertoire darknet/cfg,
- Changer la ligne batch en batch=64
- Changer les subdivisions de ligne en subdivisions=16
- Définir la taille du réseau width=416 height=416 ou toute valeur multiple de 32
- Changer la ligne max_batches2000
- Changer les étapes de ligne à 80 % et 90 % de max_batches, fe étapes = 4800,5400
- Changer [filters=255] par le filter=18

2. Fichier obj.data

- Le nombre de classes.
- Le chemin vers les fichiers train.txt et test.txt.
- Le chemin d'accès au fichier obj.names qui contient les noms des classes.
- Le chemin d'accès au dossier d'apprentissage dans lequel le modèle va enregistrer ses paramètres.

3. Fichier obj.names

Contient des noms d'objets chacun entre eux est placé dans une nouvelle ligne. Nous assurons que les classes sont dans le même ordre que dans le fichier class.txt utilisé lors de l'étiquetage des images.

4. Fichier de script <process.py>

Ce script '**process.py**' crée les fichiers **train.txt** et **test.txt** où le fichier **train.txt** a des liens de chemins vers 90 % des images et **test.txt** a des liens de chemins vers 10 % des images.

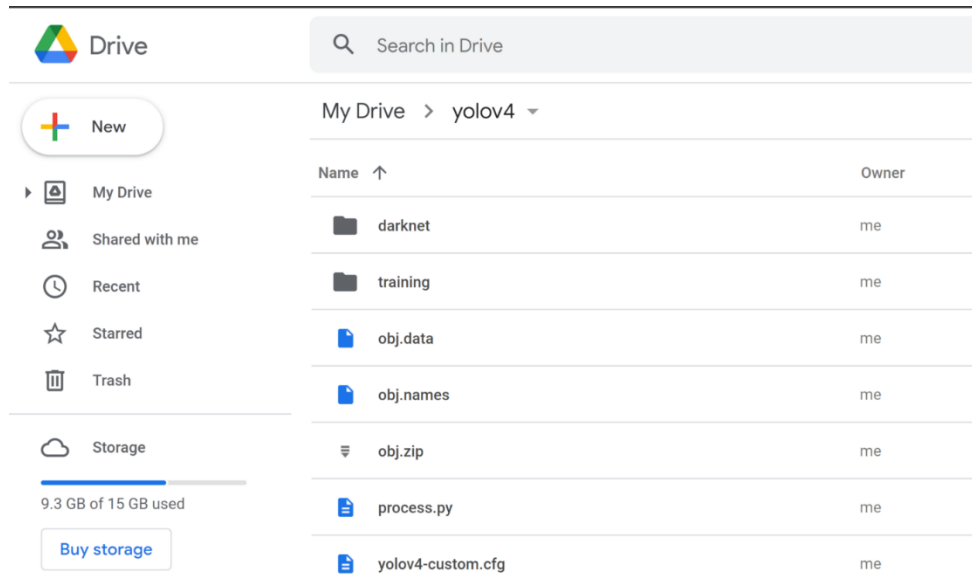


Figure 25 : Yolo4 sur drive.

Pour démarrer l'apprentissage nous exécutons la commande suivante :

```
./donnees de train du detecteur darknet/obj.datacfg/yolov4-custom.cfg yolov4.conv.137
-dont_show -map
```

Après le training nous obtenons un modèle qui permet de détecter les plaques immatriculation.



Figure 26 : Localisation de plaque de matricule.

VI. Segmentation des caractères

1. Isolement de la plaque

Nous prenons les coordonnées de bounding (Anchor) boxes créées par le modèle YOLOv4. C'est une sous-image qui représente la plaque d'immatriculation. Cette sous-image est très petite, nous utilisons la méthode (cv2.resize) pour agrandir sa taille d'origine.



Figure 27 : L'isolement de la plaque.

2. Conversion de l'image en niveau de gris

L'image originale en couleur sera transformée en une image en niveau de gris.



Figure 28 : La plaque en niveau gris.

3. Conversion de l'image en binaire

Dans cette partie, l'image sera transformée en noir et blanc.



Figure 29 : La plaque en binaire.

4. Binaire inverse

Les caractères blancs sur arrière-plan noir ont permis de faciliter de retrouver les contours des caractères dans l'image.



Figure 30 : La plaque en binaire inverse.

5. Détection de contour

La détection de contours permet une réduction importante de la quantité d'information relative à une image. Ensuite, nous utilisons Opencv pour trouver tous les contours de forme rectangulaire sur l'image et les trier de gauche à droite



Figure 31: Détection de contour.



Figure 32 : Contours de forme rectangulaire.

VII. Identifier le propriétaire du véhicule

Les caractères individuels du numéro de plaque d'immatriculation sont désormais les seules régions d'intérêt restantes. Nous segmentons chaque sous-image (régions d'intérêt). Ensuite, Ces caractères sont transmis à Tesseract (un logiciel de reconnaissance optique de caractères) pour obtenir les caractères de cette plaque.

En fin, nous obtenons les caractères formés à partir de la plaque d'immatriculation sous la forme d'une série de caractères, et les comparons avec celles enregistrées dans la base de données globale des plaques d'immatriculation pour identifier le propriétaire de la voiture.



Figure 33 :IV - Identifier le propriétaire du véhicule.

VIII. Expérimentation et résultats

1. Environnement de programmation

Le langage utilisé dans notre application est « Python 3.7 ». L'environnement de programmation est « Visual Studio », nous utilisons « label Image » pour étiquetage des données, l'entraînement de modèle fait dans « Google Colab ».

2. Ressources matérielles utilisées

Le travail d'implémentation a été réalisé sur un micro-ordinateur ayant les caractéristiques suivantes :

- Un microprocesseur Intel i3 6th avec 2.0 GHZ.
- Une mémoire de 8 GB de RAM.
- Un disque dur de 500 GB

3. Discussion des résultats

Nous allons calculer des métriques qui permettent d'analyser la performance de notre modèle.

1. Recall

Recall décrit la fraction des vrais positifs et de tous les positifs (c'est-à-dire, y compris les faux négatifs) :

$$Recall = \frac{VraiPositif}{VraiPositif + FauxNegatif}$$

2. Précision

Précision concerne la fraction des vrais positifs et de tous les positifs. Il indique à quel point les résultats sont valides. En d'autres termes, il indique combien d'éléments récupérés sont pertinents.

$$Precision = \frac{VraiPositif}{VraiPositif + FauxPositif}$$

3. F1 Score

Nous avons établi que la précision et le rappel fournissent tous deux des informations utiles, mais il serait pratique d'exprimer l'équilibre entre eux en un seul chiffre. C'est ce que fait le score F. Plus précisément, c'est un moyen harmonique de précision et de rappel. Le score F le plus simple (et le plus courant) est le score F équilibré, également appelé F₁. Le 1 indique qu'il pèse la précision et le rappel de manière égale.

$$F1\ score = 2 * \frac{recall * precision}{recall + precision}$$

Precision, Recall, & F1 Score - function

```

▶ def precision_recall_f1_score(true_labels, pred_labels):

    precision_value = precision_score(true_labels, pred_labels)
    recall_value = recall_score(true_labels, pred_labels)
    f1_score_value = f1_score(true_labels, pred_labels)

    print('Precision =', precision_value)
    print('Recall =', recall_value)
    print('F1 Score =', f1_score_value)

[ ] # classification metrics for training data
precision_recall_f1_score(Y_train, X_train_prediction)

```

```

Precision = 0.8299319727891157
Recall = 0.9242424242424242
F1 Score = 0.8745519713261649

```

Figure 34 : Precision, Recall et F1 score

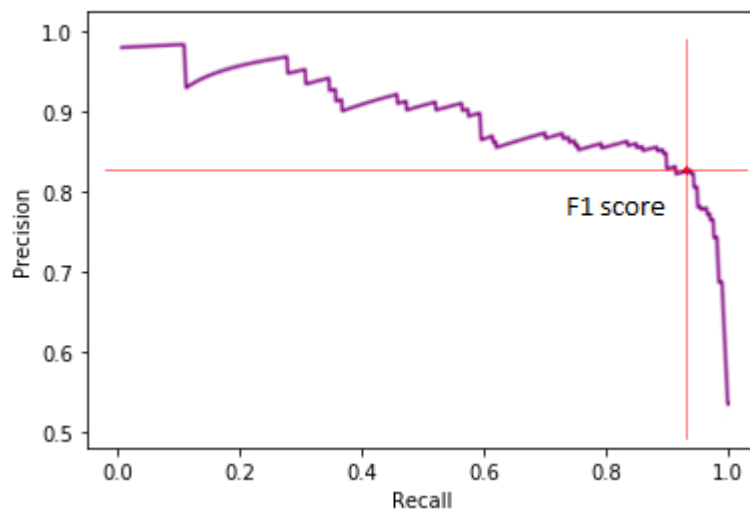


Figure 35 : Précision-Recall graphe

Les résultats obtenus sont très encourageants et prometteurs. Ces résultats montrent que l'algorithme YOLO donne des résultats précis. L'OCR reste sensible à certains caractères donc nous proposons d'augmenter la taille de dataset dans l'entraînement pour améliorer la prédiction.

IX. Conclusion

Nous avons illustré dans ce chapitre un algorithme d'identification des plaques d'immatriculation de véhicules. Cet algorithme permet de bien localiser le matricule dans une image. Cette localisation peut atteindre jusqu'à 96%. Par ailleurs la précision de l'OCR reste encore sensible à certains types d'images de mauvaise qualité ou bien à certaines positions de capture de matricule.

Conclusion générale

Conclusion générale

Lors de la préparation ce projet de fin d'études, nous avons essayé d'appliquer les connaissances acquises au cours de nos études universitaires. Ceci dans le but de réaliser un système d'identification de personnes via la plaque d'immatriculation de leurs véhicules. Cette application a la capacité d'extraire les caractères d'enregistrement avec une grande précision à partir la capture d'image sous la forme d'une série de caractères, et les comparons avec celles enregistrées dans la base de données globale des plaques d'immatriculation pour identifier le propriétaire de la voiture.

Ce projet nous a été très utile, il nous a permis de maîtriser de nombreuses techniques et de travailler avec des outils très complexes. C'était une bonne occasion de faire un travail modeste et concret afin de viser des objectifs clairs et bien définis. Cette opportunité nous permet d'en apprendre davantage sur l'environnement de développement professionnel.

En perspective, nous espérons dans le futur pouvoir intégrer dans un groupe de recherche pour élargir nos compétences et faire avancer nos connaissances dans le domaine de la vision par ordinateur. Cette intégration permet d'utiliser des technologies existantes comme le Cloud, des services web pour faciliter le stockage, le traitement et l'exécution des applications complexes comme le Big Data.

Bibliographies

- [1]. M Sandeli, traitement d'images par des approches bio-inspirées application à la segmentation d'images, université constantine 2. 2014.
- [2]. A.Medjaoui , f.fares , segmentation des images par contours actifs : application sur les Images Satellitaires à Haute Résolutions, Université Abou Bakr Belkaid–tlemcen.2012.
- [3]. T.Mohamed ,D.Djafar, la découverte des concepts sémantiques cachés avec plusieurs niveaux d'abstraction pour la recherche d'images,université kasdi merbah ouargla.2013.
- [4]. C.Houassine, segmentation d'images par une approche biomimétique hybride.université université m'hamed bougara- boumerdes. 2012.
- [5]. M .T.Chikh , amélioration des images par un modèle de réseau de neurones(comparaison avec les filtres de base), université abou-bakr belkaid tlemcen.2011
- [6]. N MERABET,M MAHLIA, recherche d'images par le contenu, université abou bakrbelkaid– tlemcen.2011
- [7]. Isabelle Block, TERI : traitement et reconnaissance d'image, cours master2IAD, ENST.
- [8]. K. Aounallah, les approches de segmentation d'image par coopération régions contours, université Kasdi Merbah ouargla 2010..
- [9]. Goodfellow, I, Bengio, Y. et Courville, A. (2016). Deep Learning. MIT Press.
- [10]. Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. Dans I. Guyon, G. Dror, V. Lemaire, G. Taylor, et D. Silver(dir.). Proceedings of ICML Workshop on Unsupervised and Transfer Learning, volume 27 de Proceedings of Machine Learning Research, 17-36, Bellevue, Washington, USA. PMLR.
- [11]. Mark Gales, 4F10: Deep learning, Cambridge Université.
- [12]. McCulloch, W. S. et Pitts, W. (1943). A logical calculus of the ideasimmanent in nervous activity. The bulletin of mathematical biophysics, 5(4),115- 133.
- [13]. Wiesel, T. N. et Hubel,·D. H. (1963). Single-cell responses in striate cortex of kittens deprived of vision in one eye. Journal of Neurophysiology, 26(6), 1003-1017. PMID : 14084161,.
- [14]. LeCun, Y, Baser, B, Denker, J. S, Henderson, D, Howard, R. E, Hubbard,W. et Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 541- 551.

- [15]. Frizzi, S, Kaabi, R, Bouchouicha, M, Ginoux, J.-M, Fnaiech, F. et Moreau,E. (2017). Détection de la fumée et du feu par réseau de neurones convolutifs.
- [16]. B. Al-Badr , R.M. Haralick : « Symbol recognition without prior segmentation ». Conference SPIE-EI 1994.
- [17]. Karpathy, A. (Sep. 2013)). Convolutional neural networks for visual.
- [18]. Krizhevsky, A, Sutskever, I. et Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, et K. Q. Weinberger (dir.), Advances in Neural Information Processing Systems 25 1097-1105. Curran Associates, Inc.
- [19]. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
- [20]. E. Lecolinet, O. Barrett : « Cursive word recognition : Methods and strategies ». In NATO/ASI , Fundamentals in handwriting recognition, Bonas, France June 21-july 3, 1993.
- [21]. B. Al-Badr , S.A. Mahmoud : « Survey and bibliography of Arabic optical text recognition ». Signal processing , vol. 41, pp. 49-77, 1995.
- [22]. P.M. Lalican, C. Viarp-Gaudin, S. Knerr : « From off-line to on-line handwriting recognition ». Proc. 7th workshop on frontiers in handwriting recognition, pp. 303-12, Amsterdam 2000.
- [23]. I.R. Tsang : « Pattern recognition and complex systems ». Thèse de doctorat, université d'Antwerpen, 2000.
- [24]. M.M.M. Fahmy, S. Al Ali : « Automatic recognition of handwritten Arabic characters using their geometrical features ». Studies in informatics and control journal (SIC journal), vol. 10, No 2, 2001.
- [25]. J. Anigbogu : « Reconnaissance de textes imprimés multiformes à l'aide de modèles stochastiques et métriques ». thèse de doctorat, Université de Nancy I, 1992.
- [26]. N. Benamara : « Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée ». Thèse de doctorat, spécialité Génie Electrique, Université des sciences, des Techniques et de médecine de Tunis II, 1999.
- [27]. A. NOBEL, "Reconnaissance automatique de plaques d'immatriculation (AutoVu) ", Genetec, 2013-Canada.

Web graphies

- [28]. "LAPIEngine", <http://www.alphanumeric-vision.com/fr/logiciel-lapi/>, Dernière visite : 31/05/17.
- [29]. "Reconnaissance de plaques d'immatriculation avec LAPI de SeeTec", <https://www.seetecvideo.com/fr/produits/extensions-seetec-cayuga/reconnaissance-des-plaquesdimmatriculation>
- [30]. "Asia Vision Technology Limited (AVT) ", <https://asiavisionweb.com>
- [31]. "Nos logiciels - AFS2R", <http://afs2r.fr/nos-produits/>, Dernière visite : 31/05/17.
- [32]. "TEB - Système LAPI : Lecture Automatique de Plaque d'Immatriculation", <http://www.tebonline.com/fr/analyse-images/lapi-lecture-automatique-plaque-immatriculation.php>
- [33]. "PROPRIÉTÉS D'UNE IMAGE NUMÉRIQUE", <https://www.lossendiere.com/2019/05/23/proprietes-images-numeriques/>.
- [34]. "Détection de contours", http://www.traitement-signal.com/detection_de_contours.php.
- [35]. "Codage en noir et blanc sur le format BMP", https://www.loutrel.fr/wikisn/doku.php?id=les_exposes:image_format_d_une_image_bmp.
- [36]. "Qu'est-ce que la détection d'objet ?", <https://www.saagie.com/fr/blog/quest-ce-que-la-detection-dobjet/>.