

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière : Télécommunications

Spécialité : Systèmes des télécommunications

Présenté par

**ANABI Nadjet**

**&**

**DJELLOUT Manel**

# Reconnaissance de billets d'argent algérien par l'apprentissage profond

Proposé par : Pr. NAMANE ABDERRAHMANE

Année Universitaire 2022-2023

## Remerciements

---

*On remercie en premier lieu notre Allah qui nous a donné la volonté et le courage pour terminer ce modeste travail.*

*Nos remerciements les plus cordiaux sont adressés particulièrement à notre encadreur Mr **NAMANE** Abderrahmane, de nos avoirs soutenus et fait confiance durant notre projet avec une grande patience. Avec son expérience dans la recherche et l'enseignement, avec ses conseils, ainsi que sa disponibilité, nous avons pu découvrir le monde de la recherche scientifique dans le domaine d'intelligence artificielle et de traitement d'image.*

*Nous tenons également à remercier les membres du jury chacun par son propre nom pour l'honneur et l'intérêt qu'ils nous ont accordé en acceptant d'examiner et d'évaluer notre mémoire.*

*Nous souhaitons à présent adresser notre sincère remerciement à tous les enseignants de département d'électronique et particulièrement les professeurs de l'option « Télécommunication ».*

*Nous remercions nos parents pour leurs soutiens durant le cursus universitaire et toutes les personnes qui nous ont soutenus de près ou loin dans la réalisation de ce travail.*

## ملخص:

يتضمن هذا المشروع تطبيق الذكاء الاصطناعي للكشف عن الأوراق النقدية الجزائرية والتعرف عليها. للقيام بذلك، يتم استخدام التعلم العميق القائم على YOLOV5. تم إنشاء قاعدة بيانات للعملات الجزائرية لتعلم أو تدريب نموذج YOLOV5. وبعد مرحلة التدريب، أجريت مرحلة اختبار، وكانت النتائج التي تم الحصول عليها مرضية ووصل النموذج إلى درجة الدقة؛ 88.19% و 92.86% و 96.58% و 96.55% لـ 1000 (الوجه 1) و 1000 (الوجه 2) و 2000 (الوجه 1) و 2000 (الوجه 2) على التوالي. يتم تقديم خطوة إضافية تعتمد على معالجة الصور لتحديد أبعاد الورقة النقدية، خاصة العرض والطول وبالتالي النسبة (العرض/الطول). تستخدم هذه النسبة لقبول أو رفض التذكرة.

## الكلمات المفتاحية:

الذكاء الاصطناعي، التعلم العميق، YOLOV5، العملات الورقية، معالجة الصور

---

## Résumé :

Ce projet consiste à l'application de l'intelligence artificielle pour la détection et reconnaissance des billets de monnaies algériens. Pour ce faire, un apprentissage profond basé sur YOLOV5 est utilisé. Une base de données de monnaie algérienne a été créée pour apprendre ou entraîner le modèle YOLOV5. Après l'étape d'entraînement, une phase de test a été réalisée, les résultats obtenus sont satisfaisants et le modèle a atteint les précisions de ; 88,19%, 92,86%, 96,58% et 96,55% pour les billets de 1000 (face 1), 1000 (face 2), 2000 (face 1) et 2000 (face 2) respectivement. Une étape supplémentaire basée sur le traitement d'image est introduite afin de déterminer les dimensions du billet de monnaies, notamment, la largeur et la longueur et par conséquent, le rapport (largeur/longueur). Ce rapport est utilisé pour accepter ou rejeter le billet.

## Mots clé :

Intelligence artificiel, apprentissage profond, YOLOV5, billets de monnaies, traitement d'images

---

---

**Abstract :**

This project consists of the application of artificial intelligence for the detection and recognition of Algerian currency notes. To do this, deep learning based on YOLOV5 is used. A database of Algerian currencies has been created to learn or train the YOLOV5 model. After the training stage, a test phase was carried out, the results obtained are satisfactory and the model has reached the precision of; 88.19%, 92.86%, 96.58% and 96.55% for the notes of 1000 (side 1), 1000 (side 2), 2000 (side 1) and 2000 (side 2) respectively. An additional step based on image processing is introduced to determine the dimensions of the banknote, in particular, the width and the length and therefore the ratio (width/length). This ratio is used to accept or reject the banknote.

**Keyword**

Artificial intelligence, Deep Learning, YOLOV5, banknotes, image processing

---

## Listes des acronymes et abréviations

ANN : Artificiel Neural Network

CNN : Convolutional Neuronal Network

CONV : Convolution

DL : Deep Learning

FC : Fully Connected Layers

FPS : Frame Per Second

GPU : Graphics Processing Unit

IA : Intelligence Artificiel

IoU : Intersection Over Union

LeNet : LeCon Network

mAp : Mean Average Precision

ML : Machine Learning

NMS : Non Max Suppression

OpenCv : Open Source Computer Vision Library

PMC : Perceptron Multicouche

R-CNN : Region-based Convolutional Neural Network

RNA : Réseaux Neurone Artificiel

RNN : Recurrent Neural Network

ROI : Region Of Interest

SPPNet : Spatial Pyramid Pooling network

SVM : Support Vector Machine

YOLO : You Only Look Once

# Table des matières

INTRODUCTION GENERALE.....	1
1. Généralité sur les billets de monnaies.....	3
1.1 Introduction .....	3
1.2 Billets de monnaies .....	4
1.3 Historique des billets de monnaie .....	4
1.4 Billets algériens utilisés aujourd’hui .....	10
1.5 Travaux réalisés sur la reconnaissance des billets .....	13
1.5.1 Système de reconnaissance des billets en euros et mexicains basé sur l’apprentissage profond avec images de scènes réelles .....	13
1.5.2 Système de détection et de reconnaissance de devises basé sur YOLO-v3 pour les personnes malvoyantes.....	13
1.5.3 Détection et reconnaissance de devises basées sur l’apprentissage profond ...	14
1.5.4 Coin net : application indépendante de la plate-forme pour reconnaître les billets de banque indiens à l'aide de techniques d'apprentissage profond .....	14
1.5.5 Reconnaissance des billets de banque indiens à l'aide d'un réseau neuronal convolutif .....	14
1.6 Conclusion.....	15
2. Traitement d’image et Intelligence Artificielle .....	16
2.1 Introduction .....	16
2.2 Techniques du traitement d’images .....	17
2.2.1 Profil d’images ou projection :.....	17
2.2.2 Conversion d’image couleur en niveau de gris .....	18
2.2.3 Seuillage ou binarisation d’une image .....	18
2.2.4 Filtrage Gaussien .....	20
2.2.5 Morphologie mathématique.....	21
2.2.6 Détection de contour .....	22
2.3 Réseaux de neurones .....	23
2.3.1 Neurone biologique .....	23
2.3.2 Neurone artificiel .....	24
2.3.3 Structure des réseaux de neurones .....	26
2.3.4 Apprentissage.....	27
2.4 Intelligence artificielle (IA) .....	29

2.4.1	Définition.....	29
2.5	Apprentissage profond.....	30
2.6	Architectures de l'apprentissage profond .....	30
2.6.1	Réseaux neuronaux convolutifs (CNN).....	30
2.6.2	Réseaux neuronaux récurrents (RNN) .....	31
2.7	Détection et reconnaissance d'objets à l'aide d'apprentissage profond .....	32
2.7.1	Définition.....	32
2.7.2	Modèles de détection par CNN.....	32
2.8	Détection d'objets avec YOLO V5 : .....	36
2.8.1	Fonctionnements du YOLO : .....	37
2.8.2	Avantages et inconvénients de YOIO V5.....	40
2.9	Conclusion.....	40
3.	Implémentation et résultats .....	41
3.1	Introduction .....	41
3.2	Environnement de travail.....	41
3.2.1	Spécifications de l'ordinateur portable :.....	41
3.2.2	Logiciels utilisés :.....	41
3.2.3	Langage de programmation utilisé (Python) :.....	42
3.2.4	Bibliothèques utilisées : .....	43
3.3	Entraînement à l'aide du YOLOv5 : .....	43
3.3.1	Création de base de données :.....	43
3.3.2	Entraînement et résultats : .....	46
3.3.3	Test du modèle YOLOv5 : .....	49
3.4	Détermination du rapport (Longueur/Largeur) .....	51
3.5	Conclusion.....	60
	CONCLUSION GENERALE.....	61
	BIBLIOGRAPHIE .....	63

## Liste des figures

<b>Figure1. 1</b> : Billets d'argent mondiales .....	4
<b>Figure1. 2</b> : Jiaozi 1er papier de monnaies Song dynasty (960–1279) .....	5
<b>Figure1. 3</b> : Recto verso d'un billet suédois en 1666 .....	6
<b>Figure1. 4</b> : Billet ancien de 10 francs 1871 .....	7
<b>Figure1. 5</b> : Billet algérien de 5 dinars 1964 .....	7
<b>Figure1. 6</b> : Billet algérien de 50 dinars 1977 .....	8
<b>Figure1. 7</b> : Billet algérien de 20 dinars 1983 .....	9
<b>Figure1. 8</b> : Billet algérien de 1000 dinars 1998 .....	9
<b>Figure1. 9</b> : Billet algérien de 2000 dinars 2011 .....	10
<b>Figure 2. 1</b> : Extraction des caractères par projection horizontale et verticale .....	17
<b>Figure 2. 2</b> : Conversion l'image en couleurs vers une image à niveaux de gris .....	18
<b>Figure 2. 3</b> : Seuillage d'une image à niveau de gris .....	19
<b>Figure 2. 4</b> : Noyau de taille 3x3 du filtre gaussien .....	20
<b>Figure 2. 5</b> : Image formée d'objets .....	21
<b>Figure 2. 6</b> : La détection de contour d'une image d'étoile de mer .....	23
<b>Figure 2. 7</b> : Un neurone avec son arborisation dendritique .....	24
<b>Figure 2. 8</b> : Neurone formel .....	24
<b>Figure 2. 9</b> : Passage du neurone biologique au neurone artificiel .....	25
<b>Figure 2. 10</b> : Les fonctions d'activation usuelles .....	26
<b>Figure 2. 11</b> : Perceptron simple .....	26
<b>Figure 2. 12</b> : Perceptron multicouche .....	27
<b>Figure 2. 13</b> : Exemple d'apprentissage supervisé .....	28
<b>Figure 2. 14</b> : Exemple d'apprentissage non supervisé .....	28
<b>Figure 2. 15</b> : Exemple d'apprentissage .....	29
<b>Figure 2. 16</b> : Domaine de l'intelligence artificiel .....	30
<b>Figure 2. 17</b> : Réseaux neuronaux convolutifs .....	31
<b>Figure 2. 18</b> : Réseaux neuronaux récurrents .....	32
<b>Figure 2. 19</b> : Architecture de model R-CNN .....	33
<b>Figure 2. 20</b> : Architecture de model Fast R-CNN .....	34
<b>Figure 2. 21</b> : Chronologie des versions de YOLO (You Only Look Once) .....	35
<b>Figure 2. 22</b> : Architecture YOLO .....	35
<b>Figure 2. 23</b> : Architecture du YOLO v5 .....	36
<b>Figure 2. 24</b> : Division de l'image en NxN grille .....	37
<b>Figure 2. 25</b> : Détermination de boîtes de délimitation .....	38
<b>Figure 2. 26</b> : Boite de délimitation de chaque classe .....	39
<b>Figure 2. 27</b> : Sélection de grille 2 .....	39
<b>Figure 3. 1</b> : Echantillons d'images de la base de données .....	44
<b>Figure 3. 2</b> : Annotation d'images sur Roboflow .....	45
<b>Figure 3. 3</b> : Division de la base de données.....	45

<b>Figure 3. 4 :</b> Types des augmentations de données sur Roboflow .....	46
<b>Figure 3. 5:</b> Exportation de base de données .....	46
<b>Figure 3. 6 :</b> Digramme des étapes d'entraînement et évaluation.....	47
<b>Figure 3. 7:</b> Code de téléchargement de base de données .....	47
<b>Figure 3. 8 :</b> Téléchargement de base de données sur Colab .....	48
<b>Figure 3. 9 :</b> Début d'entraînement .....	48
<b>Figure 3. 10 :</b> Précision obtenue pour chaque classe .....	48
<b>Figure 3. 11 :</b> Graphes de la performance et la perte.....	49
<b>Figure 3. 12 :</b> Résultats de détection pour les quatre classes.....	49
<b>Figure 3. 13:</b> Résultat de test 1000F .....	50
<b>Figure 3. 14:</b> Résultat de test 1000P .....	51
<b>Figure 3. 15 :</b> Résultat de test 2000F .....	51
<b>Figure 3. 16 :</b> Résultat de test 2000P .....	51
<b>Figure 3. 17 :</b> Différentes couleurs pour chaque classe .....	53
<b>Figure 3. 18 :</b> Seuillage des images de la figure 3.17 en utilisant l'équation.....	53
<b>Figure 3. 19 :</b> Résultats de la projection horizontale et verticale des images de la figure 3.18 .	54
<b>Figure 3. 20 :</b> Résultats de de découpage d'une image de la figure 3.18 .....	55
<b>Figure 3. 21 :</b> Extraction de l'image billet .....	55
<b>Figure 3. 22 :</b> Conversion de l'image billet couleur en niveau de gris .....	56
<b>Figure 3. 23 :</b> Application du filtre gaussien de taille 3x3 .....	57
<b>Figure 3. 24 :</b> Binarisation de l'image filtrée .....	57
<b>Figure 3. 25 :</b> Application de la fermeture morphologique avec un élément structurant de taille 11x11.....	58
<b>Figure 3. 26 :</b> Application de l'ouverture morphologique avec un élément structurant de taille 11x11.....	58
<b>Figure 3. 27 :</b> Résultats des rapports obtenus de l'ensemble de test .....	59
<b>Figure 3. 28 :</b> Moyennes des rapports obtenues pour chaque classe .....	59

## Liste des tableaux

<b>Tableau 1. 1</b> : Tableau regroupant les billets algériens utilisés aujourd’hui .....	13
<b>Tableau 3. 1</b> : Nombre d’images créé pour chaque billet .....	44
<b>Tableau 3. 2</b> : Résultats de détection pour les quatre classes.....	50
<b>Tableau 3. 3</b> : Erreur entre le rapport réel et mesuré .....	60



# Introduction générale

---

L'intelligence artificielle est un vaste domaine de recherche qui a connu une révolution rapide ces dernières années. Parmi les domaines de recherche les plus connus en IA, on trouve le "Deep Learning" ou l'apprentissage profond, en particulier dans le domaine du traitement d'images. Grâce à divers modèles de détections, il est maintenant possible de détecter et reconnaître des objets tels que les billets de monnaie.

La reconnaissance des billets de monnaie est un domaine crucial dans de nombreux secteurs tels que les systèmes de paiement automatisés, les automates bancaires et la détection de contrefaçons. Dans le cadre de ce mémoire de fin d'études, nous concentrons spécifiquement sur la reconnaissance des billets de monnaie algériens en utilisant des techniques d'apprentissage profond.

Le modèle utilisé dans notre étude le YOLOv5 est un modèle d'apprentissage profond basé sur les réseaux neuronaux convolutifs (CNN), qui est largement reconnu pour sa précision et sa rapidité dans la détection d'objets. Nous l'avons adapté spécifiquement à la reconnaissance des billets de monnaie algériens (catégorie de 1000 DA et 2000 DA), en l'entraînant sur un ensemble de données étiquetées.

Cependant, la reconnaissance des billets ne se limite pas seulement à leur détection. Nous avons également utilisé des techniques de traitement d'image pour extraire des caractéristiques spécifiques des billets, notamment leurs dimensions (largeur et hauteur). Ces informations complémentaires permettent d'enrichir les données d'entrée du modèle et d'améliorer la précision de la reconnaissance des billets.

L'objectif principal de notre étude est d'entraîner le modèle YOLOv5 à base d'apprentissage profond pour la capacité de reconnaître et de classifier les billets de monnaie algériens, avec précision et rapidité plus élevée.

Le mémoire est organisé en trois chapitres :

Dans le premier chapitre, nous présentons de façon générale les billets de monnaies dans le monde entiers et de façon spécifique sur les billets algériens, et citer quelques travaux réalisés sur ce sujet.

Dans le second chapitre, nous présentons aussi les différentes techniques du traitement d'image utilisées, ensuite les réseaux de neurones artificiels et définir l'apprentissage profond, puis présenter en détail le modèle de détection utilisé dans notre projet.

Dans le dernier chapitre, l'implémentation des différentes étapes de notre projet, la base des données ainsi créée et les résultats obtenues seront présentés et discutés.

# Chapitre 1 Généralité sur les billets de monnaies

---

## 1.1 Introduction

L'argent est l'un des outils principaux utilisé pour l'achat et la vente à travers les époques, malgré les nouvelles méthodes et techniques développées dans le domaine des transactions financières. Il reste un moyen essentiel pour satisfaire une variété de besoins spécifiques dans la vie quotidienne, que ce soit pour l'acquisition ou le commerce de biens.

Les billets de monnaie ont joué un rôle essentiel dans les systèmes financiers du monde entier depuis des siècles. Cependant, avec l'évolution technologique, il est important de développer des systèmes et des programmes de reconnaissance des billets pour faciliter leur traitement et leur sécurité. Ces systèmes ont été conçus pour permettre une identification précise et rapide des billets, garantissant ainsi leur authenticité et leur valeur, pour éviter diverses opérations de fraude.

Ces programmes et systèmes sont développés pour aider les institutions financières à reconnaître les billets de la monnaie, pour les guichets automatiques et dans l'identification des billets neufs et anciens ou endommagés, aussi pour faciliter l'identification des billets pour les personnes malvoyantes, ainsi qu'aider à identifier les faux billets dans les entreprises et dans les transactions sociales.

Dans ce chapitre, nous allons définir les billets de monnaie, parler de l'histoire des billets dans le monde entier et national et les différents types des billets algériens utilisés aujourd'hui et enfin nous discuterons sur les différents travaux et systèmes développés sur la reconnaissance et l'identification des billets.

## 1.2 Billets de monnaies

Un billet d'argent, également appelé billet de monnaie, est une forme de monnaie papier émise par le gouvernement d'un État. Le papier-monnaie est souvent déterminé par les forces de l'offre et de la demande, la stabilité du gouvernement et le système politique du pays qui l'exporte, pas par la valeur de la marchandise qui le soutient, utilisée principalement comme moyen d'échange et de paiement dans une économie ou dans une région (Voir figure 1.1) [1].

Le billet de banque est généralement fabriqué en papier spécialisé avec des caractéristiques de sécurité, telles que des filigranes, des micro-impressions, pour garantir son authenticité.



Figure1. 1 : Billets d'argent mondiales [2]

## 1.3 Historique des billets de monnaie

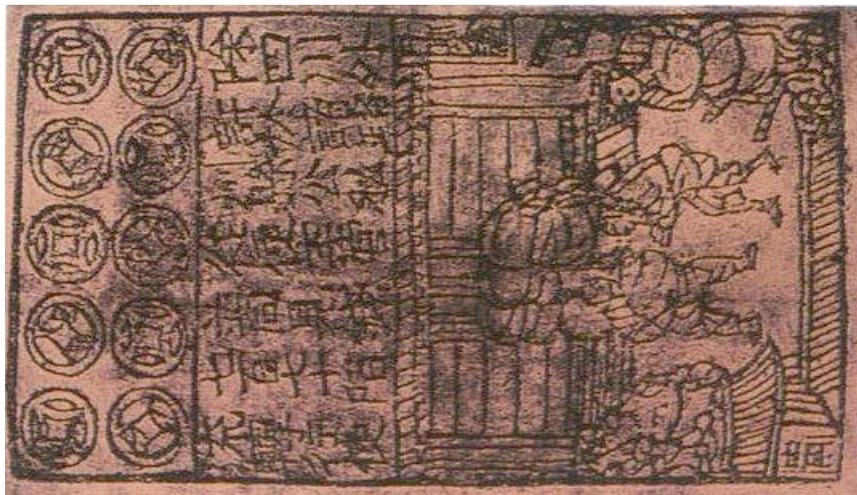
### A- Billets de monnaies dans le monde entier

#### 1. Premiers billets chinois

Le papier de monnaie est apparu pour la première fois en Chine au 7ème siècle après JC pendant la dynastie Tang (618-907). À cette époque, la monnaie officielle se composait principalement de pièces de fer, qui étaient lourdes et de faible valeur. Plutôt que de transporter ces pièces encombrantes avec eux, les gens les déposaient chez des commerçants en échange d'une lettre de change. Cette lettre de change a rapidement remplacé les pièces de monnaie en tant qu'instrument de paiement. Les Chinois ont

surnommé cette nouvelle monnaie "la monnaie volante" en raison de sa légèreté et de la facilité avec laquelle elle pouvait être utilisée dans toutes les circonstances.

Au début du XIe siècle, des commerçants chinois se sont regroupés pour émettre des billets appelés Jiao Zi, ayant une valeur fixe. Ces billets étaient imprimés sur des planches en bois à l'aide d'encre vermillon et noire. Chaque billet était unique et comportait un numéro distinct afin de prévenir la contrefaçon. Cependant, l'introduction de ces premiers billets a posé plusieurs problèmes, car certains commerçants en ont parfois fait imprimer plus qu'ils n'avaient de pièces de monnaie en réserve. Par conséquent, la dynastie Song a été contrainte de mettre en place un contrôle sur l'émission des billets Jiao Zi (voir figure 1.2) [3].



**Figure1. 2** : Jiaozi 1er papier de monnaies Song dynasty (960–1279) [5]

En 1168, une usine à papier appartenant au gouvernement a commencé à produire du papier à partir de l'écorce grise du mûrier, qui servait ensuite à imprimer les billets de banque. Au XIIIe siècle, la soie a été utilisée comme substitut à l'écorce de mûrier.

La monnaie de papier est restée un moyen de paiement très apprécié en Chine. Les souverains de la dynastie Ming (1368-1644) ont même interdit complètement l'utilisation de la monnaie métallique pendant une période d'environ cent ans [3].

## **2. Premiers billets européens**

En 1661, la Suède fait face à une pénurie d'argent métallique, ce qui entraîne une dépréciation de la monnaie en cuivre. Cela a causé une inflation catastrophique et a eu

de graves répercussions sur l'économie du pays. Johann Palmstruck, un commerçant hollandais, est intervenu pour aider en proposant une solution. Il a suggéré l'utilisation de lettres de change et a fondé la première banque suédoise, la Banque de Stockholm. Le 16 juillet 1661, cette banque a émis les premiers "certificats" ou bien papier de monnaies.

Les billets étaient numérotés à la main et portaient la signature des employés de la banque. Cependant, cela n'a pas empêché leur fréquente contrefaçon, ce qui a poussé la banque à introduire trois nouvelles mesures de sécurité : l'utilisation d'un papier à filigrane et des ornements complexes sur les bords des billets (voir figure 1.3) [3].



**Figure1. 3:** Recto verso d'un billet suédois en 1666 [4]

## B- Billets algériens

Pendant la période de colonisation prolongée et violente, la France étend son pouvoir sur une partie du Sahara en 1902, complétant ainsi la division de l'Algérie en provinces françaises.

Durant l'occupation française de l'Algérie, les billets de banque n'ont été introduits qu'une seule fois, à partir de 1851. Entre 1862 et 1875, la Banque d'Algérie a émis une série de billets bleus, techniquement identiques à ceux en circulation en France à la même époque.

Jusqu'en 1912, les premiers billets étaient généralement imprimés en deux ou trois couleurs. À partir de 1920, tous les billets ont été imprimés en plusieurs couleurs dès leur création (voir figure 1.4) [6].



**Figure1. 4** : Billet ancien de 10 francs 1871 [7]

Le dinar algérien est l'unité de base de la monnaie algérienne et symbole de la République algérienne, est divisé en 100 centimètres, en 1964 après deux ans de l'indépendance, par décret présidentiel, le dinar algérien a été commercialisé et la banque centrale algérienne introduit la première série de billets de banque qui se divise en 4 catégories. Cette série met en valeur les paysages des régions du nord et du sud du pays (voir figure 1.5) [8].



**Figure1. 5** : Billet algérien de 5 dinars 1964 [8]

La face avant du billet porte un dessin d'un paysage de montagne, composé d'un faucon (à gauche) et d'un aigle royal (au centre) sur deux pics rocheux. Au verso du billet, une scène pastorale composée de moutons broutant dans un paysage bordé d'aloë Vera, à l'avant, un panier et un petit mouton, et en arrière-plan, des moutons apparaissent.

Entre 1970 et 1978, pendant la période de la révolution industrielle et agricole du pays, une nouvelle série des billets en cinq catégories a été produite. Cette série présente la touche artistique de l'un des fondateurs de la peinture moderne, l'artiste Mohammed Isyakhim (voir la figure 1.7) [8].



**Figure1. 6** : Billet algérien de 50 dinars 1977 [8]

Le billet de 50 dinars algériens présente à l'avant une scène pastorale qui symbolise la troisième étape de la révolution agricole, avec un village socialiste et des motifs décoratifs inspirés de l'art traditionnel algérien, tels que des veines et des tresses. Le verso du billet comporte également une scène représentant la moisson du blé fauché, avec en arrière-plan un village socialiste, symbolisant ainsi la révolution agricole.

La durée du 1981 au 1984, l'Algérie introduit une autre série des billets de quatre catégories : 100 dinars, 20 dinars, 200 dinars et 10 dinars (voir figure 1.8) [8].



**Figure1. 7** : Billet algérien de 20 dinars 1983 [8]

La face du billet montre des éléments de la culture artisanale et architecturale de l'Algérie, représentés par des bijoux traditionnels berbères et de la poterie artisanale, ainsi qu'une nouvelle porte inspirée du style hispano-maghrébin. Le verso du billet de banque présente des sites historiques ainsi que des représentations d'inscriptions rupestres.

La décennie des années 90 a été marquée par l'émission des nouveaux billets, six catégories : 1000 dinars (1995), 100 dinars, 200 dinars, 500 dinars (1996), 500 dinars et 1000 dinars (1998) (voir figure 1.9) [8].



**Figure1. 8** : Billet algérien de 1000 dinars 1998 [8]

Le billet de 2000 DA, émis en 2011, le billet le plus sûr, porte l'image d'un amphithéâtre, d'un groupe de chercheurs, un palmier, un olivier, une résidence et un plan d'eau en dédicace à l'Algérie orientée vers le développement économique après un demi-siècle de construction (voir figure 1.10) [9].



Figure1. 9: Billet algérien de 2000 dinars 2011 [10]

#### 1.4 Billets algériens utilisés aujourd’hui

Aujourd’hui, il existe 8 catégories de billets utilisées en Algérie 200, 500, 1000, 2000 dinars, nous allons présenter dans le tableau ci-dessus les caractéristiques de chaque billet telles que leur année de réalisation et leur taille ainsi que la forme (voir tableau 1.1).

	<p><b>Type 1 :</b>            Année : <b>1983</b>            Valeur : <b>200 Dinars</b>            Composition : <b>Papier</b>            Taille : <b>165 × 79mm</b>            Forme : <b>Rectangulaire</b></p>
	<p><b>Type 1 :</b>            Année : <b>1992</b>            Valeur : <b>200 Dinars</b>            Composition : <b>Papier</b>            Taille : <b>140 × 70mm</b>            Forme : <b>Rectangulaire</b></p>

	<p><b>Type 2 :</b>  Année : <b>1992</b>  Valeur : <b>500 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>150 × 71mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 2 :</b>  Année : <b>1998</b>  Valeur : <b>500 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>150 × 71mm</b>  Forme : <b>Rectangulaire</b></p>

	<p><b>Type 2 :</b>  Année : <b>2018</b>  Valeur : <b>500 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>152 × 72mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 3 :</b>  Année : <b>1992</b>  Valeur : <b>1000 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>160 × 72mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 3 :</b>  Année : <b>1998</b>  Valeur : <b>1000 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>161 × 71mm</b></p>

	<p>Forme : <b>Rectangulaire</b></p>
	<p><b>Type 3 :</b>  Année : <b>2005</b>  Valeur : <b>1000 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>160 × 72mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 3 :</b>  Année : <b>2018</b>  Valeur : <b>1000 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>160 × 72mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 4 :</b>  Année : <b>2011</b>  Valeur : <b>2000 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>161× 71mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 4 :</b>  Année : <b>2020</b>  Valeur : <b>2000 Dinars</b>  Composition : <b>Papier</b>  Taille : <b>158 × 71,8mm</b>  Forme : <b>Rectangulaire</b></p>
	<p><b>Type 4 :</b>  Année : <b>2022</b>  Valeur : <b>2000 Dinars</b>  Composition : <b>Substrat hybride</b></p>

	<p>Taille : <b>160 × 72mm</b></p> <p>Forme : <b>Rectangulaire</b></p>
---	---

**Tableau1. 1** : Tableau regroupant les billets algériens utilisés aujourd’hui [10]

### **1.5 Travaux réalisés sur la reconnaissance des billets**

Plusieurs études ont été effectuées pour l’identification des billets de monnaie. Parmi ces travaux, nous pouvons citer cinq travaux sur la détection et la reconnaissance, identification des billets de monnaies.

#### **1.5.1 Système de reconnaissance des billets en euros et mexicains basé sur l’apprentissage profond avec images de scènes réelles**

Deysy Galeana-Pérez et Eduardo Bayro-Corrochano [11] ont proposé un système de reconnaissance des billets. Un taux élevé de reconnaissance et de classification des billets mexicains et en euros grâce aux réseaux neuronaux et à l’apprentissage profond a été atteint. Les résultats expérimentaux ont montré que cette approche est prometteuse, avec des performances remarquables. Elle permet une classification efficace des billets, utilisant des images de scènes réelles prises en plein soleil ou en lumière artificielle, tout en étant insensible à la rotation et à la translation des billets. Au total, 1 500 images ont été utilisées pour ce système avec 300 par classe. Ce système a une efficacité globale de 98,4% pour les billets mexicains et 99.93% pour les billets en euros.

#### **1.5.2 Système de détection et de reconnaissance de devises basé sur YOLO-v3 pour les personnes malvoyantes**

Rakesh Chandra Joshi et al.[12] ont proposé un système de détection et de reconnaissance de billets, basé sur le modèle YOLO-v3 CNN. Ce système a été développé pour offrir la rapidité et la précision. Des images représentant différentes dénominations de billets et capturées dans diverses conditions ont été collectées. Ces images ont ensuite été augmentées en appliquant diverses transformations géométriques et modifications d'image, afin de rendre le système plus robuste. Les images augmentées ont été annotées manuellement, permettant ainsi la création de jeux de données d'apprentissage et de validation. Par la suite, les performances du modèle entraîné ont été évaluées en utilisant des scènes en temps réel ainsi qu'un

ensemble de données de test. Les résultats des tests ont montré que la méthode proposée, basée sur le modèle YOLO-v3, atteint une précision de détection de 95,71 % et une précision de reconnaissance de 100 %. Le système dans son ensemble est autonome et fonctionne en temps réel [12].

### **1.5.3 Détection et reconnaissance de devises basées sur l'apprentissage profond**

Qian Zhang et W. Yan[13] ont utilisé principalement le modèle Single Shot MultiBox Detector (SSD) basé sur l'apprentissage profond comme cadre, et ils ont employé le modèle de réseau neurones convolutifs (CNN) pour extraire les caractéristiques de la monnaie papier, dans le but de reconnaître plus précisément la dénomination de la monnaies papier, à la fois recto et verso. Leur principale contribution réside dans l'utilisation de CNN et de SSD, la précision moyenne de la reconnaissance des devises pouvant atteindre 96,6%.

### **1.5.4 Coin net : application indépendante de la plate-forme pour reconnaître les billets de banque indiens à l'aide de techniques d'apprentissage profond**

Veeramsetty Venkataramana et al [14] ont développé une application à l'aide d'apprentissage profond pour les malvoyants qui éprouvent des difficultés à reconnaître les billets de monnaies.

Ils ont présenté une nouvelle approche, qui consiste à développer un modèle léger de réseau neuronal convolutif (CNN) pour permettre des applications Web et mobiles efficaces dans la reconnaissance des billets de monnaies indiens. Un nouvel ensemble de données spécifique aux billets de banque indiens a été créé pour entraîner, valider et tester ce modèle CNN. Les applications Web et mobiles basées sur le CNN fourniront une sortie texte et audio correspondant au billet de banque reconnu. Le modèle proposé a été développé en utilisant TensorFlow et a été amélioré par la sélection optimale des hyperparamètres, tout en étant comparé aux architectures CNN bien connues existantes qui utilisent l'apprentissage par transfert.

### **1.5.5 Reconnaissance des billets de banque indiens à l'aide d'un réseau neuronal convolutif**

Shubham Mittal et Shiva Mittal [15] ont présenté une méthode basée sur l'apprentissage profond pour identifier les dénominations des billets en roupie indienne à partir d'images en couleur. Ils ont utilisé un cadre de classification basé sur

l'apprentissage par transfert, où un grand réseau neuronal convolutif pré-entraîné sur des millions d'images naturelles est utilisé pour classer les images de nouvelles classes. Pour cela, ils ont préparé un ensemble de données d'images de billets de banque en prétraitant et augmentant des images réelles acquises à partir de différents angles et conditions d'éclairage à l'aide d'un smartphone. En ajoutant une nouvelle couche supérieure à un modèle pré-entraîné MobileNet, ils ont entraîné le modèle pendant quelques époques (epoch) sur une partie de l'ensemble de données afin d'obtenir une précision satisfaisante lors de la validation. Le modèle léger ainsi obtenu a atteint une précision de 96,6% sur un sous-ensemble de test, sans nécessiter d'ingénierie de fonctionnalités ou de prétraitement complexe. Les résultats expérimentaux prouvent qu'il est utilisable pour le développement de systèmes portables dédiés à l'identification des billets.

## **1.6 Conclusion**

Les billets de monnaie sont indispensables dans notre vie quotidienne et jouent un rôle important dans l'économie, que ce soit au niveau national ou mondial. Même avec les avancées des méthodes de paiement et les systèmes technologiques modernes, les billets de monnaie restent un élément essentiel de notre système économique. Dans ce chapitre, nous avons donné une définition des billets de monnaie et présenté leur histoire à l'échelle mondiale ainsi que spécifiquement en Algérie. Nous avons présenté un tableau récapitulatif des billets de monnaie algériens, comprenant leurs informations, l'année de leur création, leur valeur et leurs dimensions ainsi que leur forme et composition. De plus, nous avons examiné les travaux existants sur la reconnaissance et l'identification des billets. Ces travaux utilisent l'apprentissage profond comme moyen de classification et reconnaissance des billets.

# Chapitre 2 Traitement d'image et Intelligence

## Artificielle

---

### 2.1 Introduction

Le traitement d'images fonctionne parfaitement avec l'intelligence artificielle, et qui se concentre sur l'analyse, la manipulation et l'interprétation des images numériques dans le but d'améliorer leur qualité et extraire des informations utiles. Grâce aux avancées technologiques et aux algorithmes sophistiqués, le traitement d'images a ouvert la voie à de nombreuses applications, en particulier dans le domaine de la détection d'objets.

L'intelligence artificielle, et plus précisément le domaine du Deep Learning, joue un rôle important dans la détection d'objets à partir d'images. Les réseaux de neurones artificiels (RNA) sont une classe de modèles de Deep Learning très utilisés dans la détection d'objets telle que les billets de monnaies.

La détection d'objets avec les réseaux de neurones consiste à localiser et classifier les objets présents dans une image. Différentes architectures de réseaux de neurones ont été développées, parmi lesquelles on trouve R-CNN, Fast R-CNN et YOLO. Chaque modèle possède ses propres caractéristiques et performances.

Dans ce chapitre nous allons présenter les réseaux de neurones artificiels (RNA) et l'apprentissage profond et ses structures, ainsi nous allons nous concentrer sur les différents modèles de détections spécifiquement le modèle YOLO V5. D'autre part, nous abordons les techniques du traitement d'images utilisé pour calculer les dimensions des billets de monnaies et par conséquent le rapport largeur/hauteur.

## 2.2 Techniques du traitement d'images

Dans cette section, nous allons aborder les différentes techniques du traitement d'images utilisées dans ce travail. Nous examinerons le profil (projection horizontale et verticale) d'images, la conversion d'une image couleur en niveaux de gris, le filtrage, la binarisation, la morphologie mathématique ainsi que la détection de contours.

### 2.2.1 Profil d'images ou projection :

La méthode du profil d'images ou projection horizontale/verticale est principalement utilisée pour la segmentation ou l'extraction des objets dans une image.

Le profil d'images est calculé séparément pour différents axes. Le profil d'images le long de l'axe vertical est appelé profil de projection verticale. Le profil de projection verticale est calculé pour chaque colonne comme la somme de toutes les valeurs de pixels de ligne à l'intérieur de la colonne. Profil de projection horizontale est le profil de projection d'une image le long de l'axe horizontal. Le profil de projection horizontale est calculé pour chaque ligne comme la somme de toutes les valeurs de pixel de colonne à l'intérieur de la ligne.

Le profil de projection est calculé pour une image à seuil ou une image binaire où une image à niveaux de gris avec des valeurs de pixel de 0 ou 255. Les pixels de l'image sont remplacés par 1 et 0 pour les valeurs de pixel 0 et 255 respectivement [16].

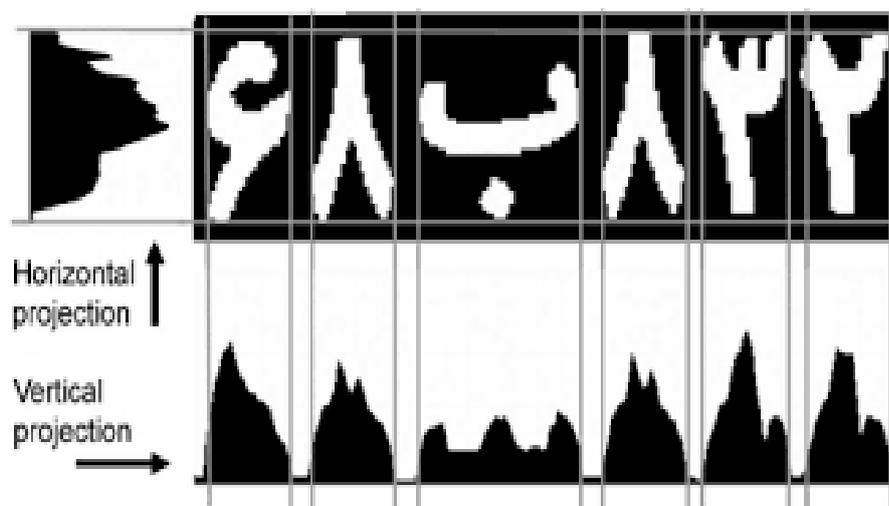


Figure 2. 1: Extraction des caractères par projection horizontale et verticale [17]

## 2.2.2 Conversion d'image couleur en niveau de gris

Pour passer d'une image couleur (RVB) vers une image à niveaux de gris ou noir et blanc, on doit exploiter une des deux équations du codage YUV ou YIQ qui sont identiques et qui est :

$$G = 0,299*R + 0,587*V + 0,114*B \quad (2.1)$$

Avec G est le niveau de gris du pixel, et R, V et B sont les composantes rouge, vert et bleue du pixel. [18]

Pour afficher une image à niveaux de gris sur un écran couleur, les trois composantes RVB doivent être identiques comme suit :

$$R = V = B = G$$

L'application de cette conversion est illustrée par les images suivantes (voir figure 2.2) :

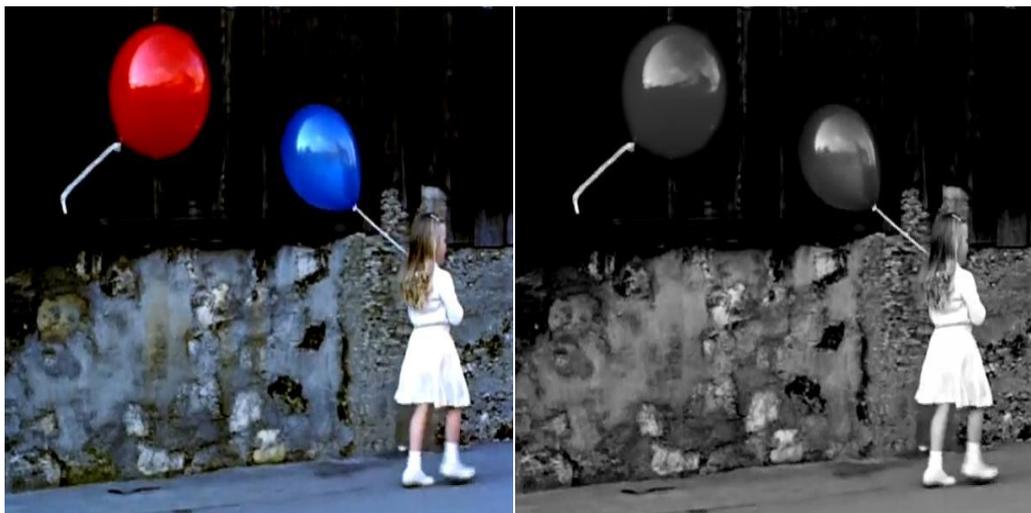


Figure 2. 2: Conversion l'image en couleurs vers une image à niveaux de gris [18]

## 2.2.3 Seuillage ou binarisation d'une image

Le seuillage ou binarisation d'image est une méthode de segmentation d'image considérée comme la plus simple. Elle consiste à prendre une image en niveaux de gris et à créer une version monochrome de celle-ci, ne contenant que deux valeurs : noir ou blanc. Lors du seuillage d'image, chaque pixel de l'image est remplacé individuellement en fonction d'une valeur seuil prédéterminée, telle que 123. Si la valeur d'un pixel est

supérieure au seuil (par exemple 150), il sera assigné à la valeur 255 (blanc), tandis que s'il est inférieur au seuil (par exemple 100), il sera assigné à la valeur 0 (noir). Cette méthode est également appelée binarisation car elle consiste à convertir l'image en une forme binaire voir (figure 2.3) [19].



**Figure 2. 3:** Seuillage d'une image à niveau de gris [20]

#### **A- Méthode d'Otsu**

En vision par ordinateur et traitement d'image, la méthode d'Otsu est une technique utilisée pour réaliser une binarisation automatique à partir de la forme de l'histogramme de l'image. Elle permet de convertir une image en niveaux de gris en une image binaire. L'algorithme d'Otsu repose sur l'hypothèse que l'image à binariser se compose uniquement de deux classes de pixels, à savoir le premier plan et l'arrière-plan. Il calcule ensuite le seuil optimal qui sépare ces deux classes de manière à minimiser la variance intra-classe [21].

La méthode d'Otsu consiste à rechercher le seuil qui minimise la variance intra-classe parmi tous les seuillages possibles :

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad (2.2)$$

Les poids  $\omega_i$  représentent la probabilité d'être dans la  $i$ ème classe, chacune étant séparée par un seuil  $t$

Finalement, les  $\sigma_i^2$  sont les variances de ces classes.

Otsu montre que minimiser la variance intra-classe revient à maximiser la variance inter-classe :

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_1(t)\omega_2(t)[u_1(t) - u_2(t)]^2 \quad (2.3)$$

Qui est exprimée en termes des probabilités de classe  $\omega_i$  et des moyennes de classes  $u_i$  qui à leur tour peuvent être mises à jour itérativement.

### 2.2.4 Filtrage Gaussien

Un filtre gaussien est un filtre passe-bas utilisé pour réduire le bruit (composants haute fréquence) et les zones de flou d'une image. Le filtre est implémenté comme un noyau symétrique de taille impaire qui est passé par chaque pixel de la région d'intérêt pour obtenir l'effet désiré [5]. Un filtre gaussien pourrait être considéré comme une approximation de la fonction gaussienne (mathématiques), la fonction gaussienne est :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.4)$$

Où  $\sigma$  est l'écart-type de la distribution, la distribution est supposée avoir une moyenne de 0.

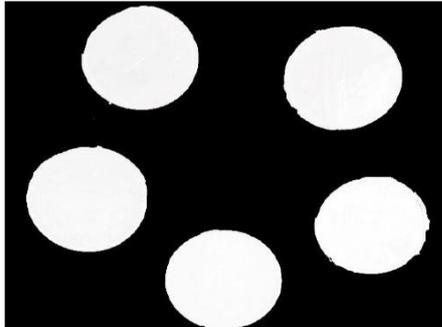
En utilisant la fonction ci-dessus un noyau gaussien de n'importe quelle taille peut être calculé, en lui fournissant des valeurs appropriées. Une approximation du noyau gaussien (bidimensionnelle) avec écart type = 1 apparaît comme suit (voir figure 2.4) :

$\frac{1}{16}$	1	2	1
	2	4	2
	1	2	1

**Figure 2. 4:** Noyau de taille 3x3 du filtre gaussien [22]

## 2.2.5 Morphologie mathématique

La morphologie mathématique est dédiée à l'analyse des structures spatiales. Elle est dite morphologique parce qu'elle vise à analyser la forme (morphologie) des objets. Une image est vue comme un ensemble de formes (voir figure 2.5).



**Figure 2. 5:** Image formée d'objets [22]

Les filtres morphologiques sont souvent utilisés pour éliminer des pixels isolés considérés comme un bruit dans une image. Pour analyser les images, ces méthodes utilisent des formes connues appelées éléments structurants (en anglais : structuring elements) [22].

L'analyse d'image par la morphologie mathématique nécessite l'utilisation d'opérations morphologiques, parmi ces opérations il y a la dilatation, l'érosion, l'ouverture et la fermeture mathématique.

### a) Dilatation :

La dilatation a pour objectif de dilater les objets ou formes dans une image. Pour une image binaire  $X$  et un élément structurant  $B$ , nous avons :

$$D^B(x) = X \oplus B = \{x | B_x \cap X \neq \Phi\} \quad (2.5)$$

Où

$$B_x = \{b + x | b \in B\} \quad (2.6)$$

$B_x$  : l'ensemble de  $B$  translaté de  $x$ .

Dans une image binaire cette opération permet d'éliminer les pixels noirs isolés mais ajoute des pixels blancs au contour des objets présents dans l'image. Le résultat de cette opération est l'augmentation de la taille de ces objets.

**b) Erosion :**

L'érosion a pour objectif de d'éroder les objets ou formes dans une image. Pour une image binaire X et un élément structurant S, nous avons :

$$E^B(x) = X \ominus B = \{x | B_x \subseteq X\} \quad (2.7)$$

Elle permet d'éliminer les pixels blancs isolés au milieu des parties noires de l'image. Le résultat de cette opération est la diminution de la taille des objets présents dans l'image.

**c) Ouverture :**

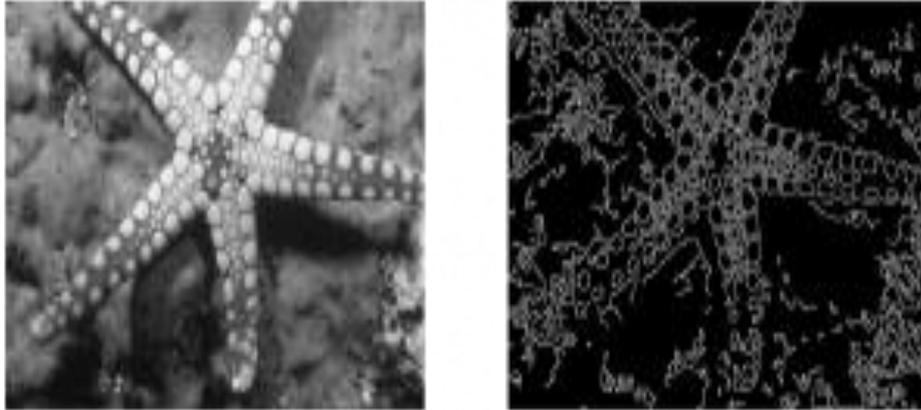
L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet d'éliminer les taches blanches dans le fond de l'image.

**d) Fermeture :**

La fermeture est l'opération inverse de l'ouverture, qui consiste à faire subir à l'image une dilatation suivie d'une érosion. Elle permet d'éliminer les noirs qui se trouvent dans l'objet.

### **2.2.6 Détection de contour**

La détection de contour est une méthode de segmentation d'une image en régions de discontinuité. C'est une technique largement utilisée dans le traitement d'image numérique comme : reconnaissance de modèle, morphologie de l'image et extraction de fonctionnalités. La détection des contours permet aux utilisateurs d'observer les caractéristiques d'une image pour une modification significative du niveau de gris. Cette texture indique la fin d'une région dans l'image et le début d'une autre. Il réduit la quantité de données dans une image et préserve les propriétés structurelles d'une image (voir figure 2.6) [23]



**Figure 2. 6:** La détection de contour d'une image d'étoile de mer [24]

## **2.3 Réseaux de neurones**

Les réseaux de neurones artificiels (RNA) tentent de reproduire les mêmes comportements que leurs modèles biologiques. La complexité d'un cerveau humain ne peut pas être reproduite, mais des approximations permettent d'obtenir des résultats utiles pour un grand nombre d'applications comme la reconnaissance de formes ou de données [25].

### **2.3.1 Neurone biologique**

Le cerveau humain, le système de traitement de l'information le plus puissant, est constitué d'environ 50-100 milliards d'unités appelées neurones [26], le neurone est l'une des cellules majeures du système nerveux. Ces cellules nerveuses sont responsables de la réception et de la transmission des impulsions nerveuses. Un neurone est constitué d'une cellule ou d'un corps de cellule somatique ou somatique, le centre de contrôle de cette dernière, qui additionne l'information qui l'atteint. Il traite ensuite l'information et renvoie le résultat sous forme de signaux électriques du corps cellulaire à l'entrée des autres neurones. C'est une cellule composée principalement de trois parties, qui jouent un rôle bien défini. Ce sont les dendrites, les axones, et le noyau. Les réseaux de neurones biologiques exécutent facilement certaines applications, comme la reconnaissance, la mémorisation ou la compréhension d'un langage humain tel qu'il est parlé, mais cette simulation reste loin du cerveau humain (voir figure 2.7).

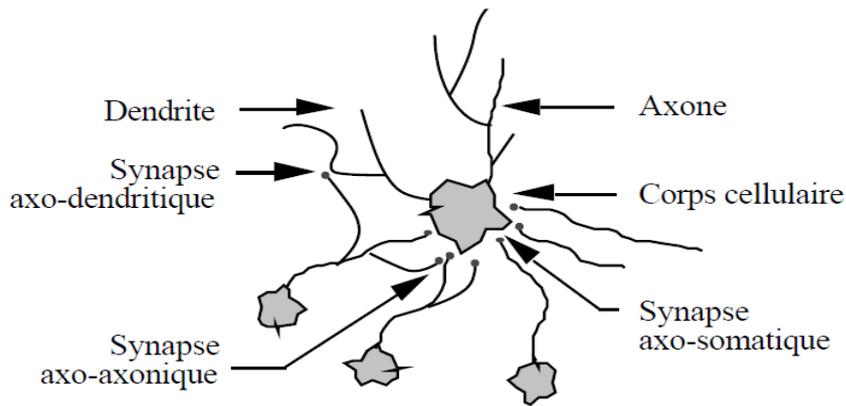


Figure 2. 7: Un neurone avec son arborisation dendritique [27]

### 2.3.2 Neurone artificiel

Un neurone artificiel ou neurone formel est la modélisation mathématique du neurone biologique basée sur la structure des neurones à l'intérieur d'un cerveau humain pour modéliser et résoudre des problèmes complexes (voir figure 2.8). Les réseaux neuronaux artificiels sont des réseaux fortement connectés d'unités de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique basée sur l'information qu'il reçoit [27], tel que :

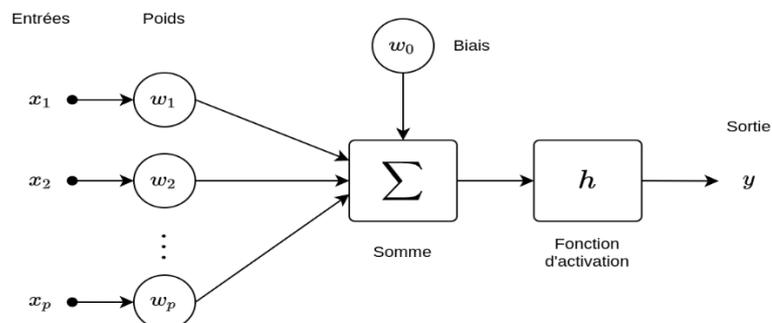


Figure 2. 8: Neurone formel [28]

- Les entrées ( $x_1, x_2, \dots, x_p$ ) apportent un signal (une valeur continue).
- Chaque entrée est multipliée par un poids (weight).
- La somme de ces entrées, multipliées par leurs poids respectifs, dont le résultat est calculé à l'aide de la fonction d'activation  $h$ .

La sortie  $h$ , se calcule comme une combinaison linéaire des poids et des entrées (plus un biais  $w_0$ ).

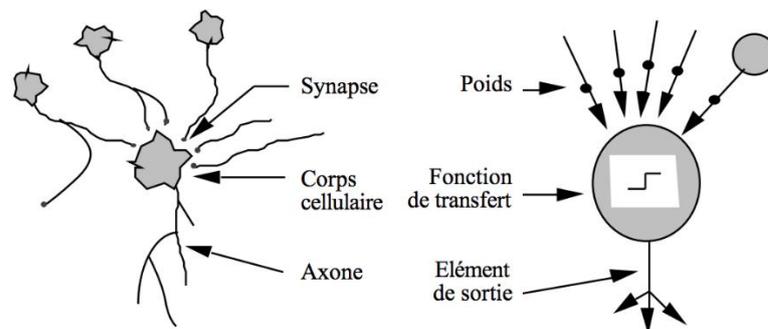
$$y = h\left(\omega_0 + \sum_{i=1}^p \omega_i x_i\right) \quad (2.8)$$

Cette fonction d'activation  $h$  permet de prendre une décision.

- Si la combinaison linéaire est supérieure ou égale à 0, alors la sortie  $y$  valait 1.
- Dans le cas contraire, la sortie  $y$  valait 0.

Un neurone formel est une représentation artificielle et schématique d'un neurone biologique [29] (voir figure 2.9) :

- Les synapses sont modélisées par des poids,
- Le soma ou corps cellulaire est modélisé par la fonction de transfert, appelé aussi fonction d'activation
- L'axone par l'élément de sortie



**Figure 2. 9:** Passage du neurone biologique au neurone artificiel [29]

### A- Fonction d'activation

Il existe plusieurs types de fonctions d'activations possibles. La figure 2.10 présente les fonctions donnant généralement les meilleurs résultats possibles. Il est toutefois possible, dans certains cas particuliers, d'utiliser des fonctions plus spécifiques, liées à leur contexte d'utilisation afin d'obtenir de meilleures performances. Le choix de celle-ci se fait en fonction de la position du neurone dans le réseau et aussi en fonction du domaine d'application du réseau de neurones [25]

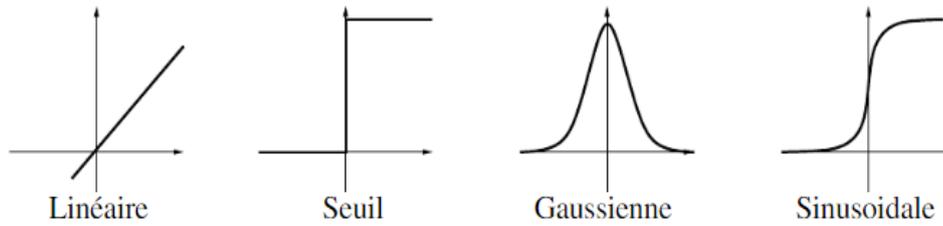


Figure 2. 10 : Les fonctions d'activation usuelles [25]

### 2.3.3 Structure des réseaux de neurones

#### A- Perceptron simple

Le perceptron simple est un réseau monocouche, acyclique (il ne comporte pas de boucle) et dont la dynamique (l'activité) est déclenchée par la réception en entrée d'information. Ce réseau est dit simple car il ne se compose que de deux couches : une couche d'entrée et une couche de sortie ce qui implique une seule matrice de poids. L'ensemble des unités de la couche d'entrée sont connectés à celles de la couche de sortie. Ces structures lui permettent d'être considéré comme un classifieur linéaire, il peut classer/séparer les données selon deux caractéristiques (Voir figure 2.11) [30].

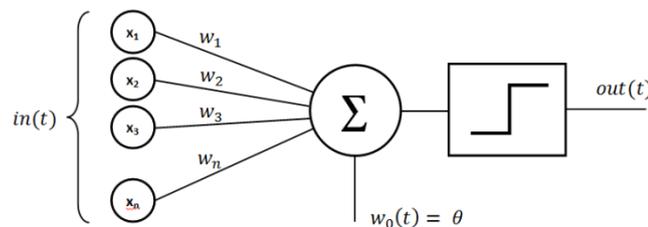
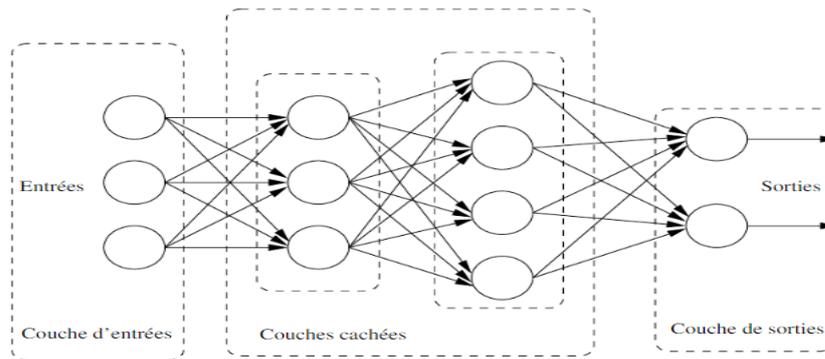


Figure 2. 11: Perceptron simple [31]

#### B- Perceptron multicouche

Le perceptron multicouche (PMC) en Anglais (MLP ; Multi-layer Perceptron) se compose d'une couche d'entrée, d'une couche de sortie et d'une ou plusieurs couches cachées. Si le réseau possède  $n$  couches, alors il possède  $n-1$  matrice de poids (une entre chaque suite de couches). Le MLP doté d'une couche cachée est théoriquement un approximateur universel de fonctions (Voir figure 2.12) [30].



**Figure 2. 12:** Perceptron multicouche [32]

### 2.3.4 Apprentissage

Le concept d'apprentissage englobe généralement deux étapes successives :

- La mémorisation, qui consiste à assimiler une quantité d'exemples, pouvant être nombreux, sous une forme compacte.
- La généralisation, qui permet, grâce aux exemples mémorisés, de traiter des exemples différents mais similaires qui n'ont pas encore été rencontrés [33].

Il existe deux types d'apprentissage base : l'apprentissage supervisé et l'apprentissage non supervisé.

#### **A- Apprentissage supervisé :**

Dans l'apprentissage supervisé l'être humain aide l'algorithme pour apprendre, un data scientist sert de guide et il apprend l'algorithme les résultats qu'il doit trouver. Le même cas lorsqu'on apprend un enfant pour identifier les fruits, en les mémorisant dans sa mémoire. Dans l'apprentissage supervisé, l'algorithme apprend grâce à un jeu de données déjà étiqueté et dont le résultat est prédéfini (voir figure 2.13) [34].

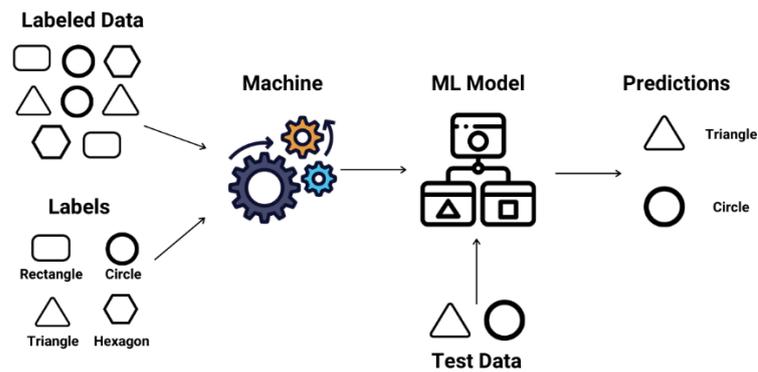


Figure 2. 13: Exemple d'apprentissage supervisé [35]

### B- Apprentissage non supervisé :

Avec l'apprentissage non supervisé la machine n'a pas besoin de l'aide pour apprendre. L'apprentissage non supervisé est une approche plus indépendante, dans laquelle un ordinateur apprend à identifier des processus et des schémas complexes sans aucun guide, Il implique une formation basée sur des données sans étiquette, qui ne contiennent aucun résultat spécifique (voir figure 2.14) [34].

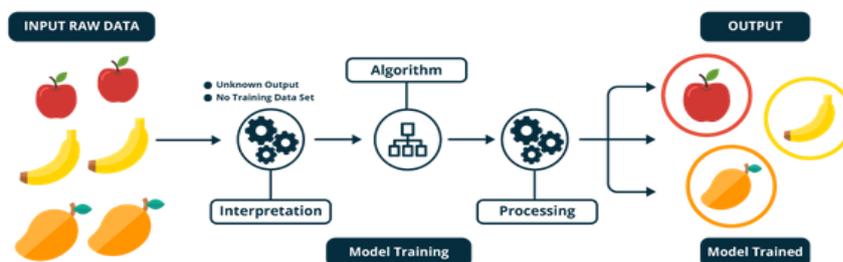
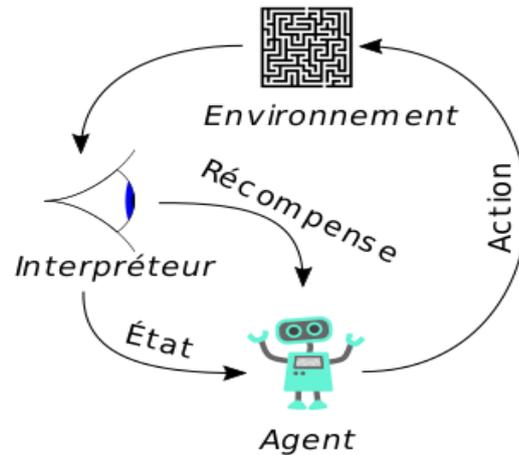


Figure 2. 14 : Exemple d'apprentissage non supervisé [35]

### C- L'apprentissage par renforcement :

L'apprentissage par renforcement est une méthode qui consiste à optimiser de manière itérative un algorithme uniquement à partir des actions qu'il entreprend et de la réponse associée de l'environnement dans lequel il évolue (voir figure 2.15) [35].



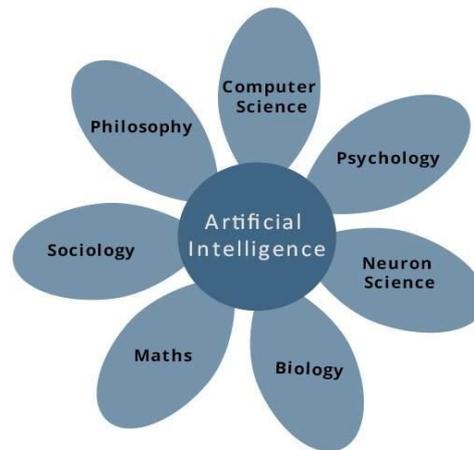
**Figure 2. 15:** Exemple d'apprentissage [35]

Cette méthode permet aux machines et aux agents de déterminer automatiquement le comportement idéal dans un contexte spécifique pour maximiser ses performances.

## 2.4 Intelligence artificielle (IA)

### 2.4.1 Définition

L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son objectif principal est de permettre aux machines de raisonner et de se comporter de manière similaire à celle des êtres humains. Le terme « intelligence artificielle » a été créé plus récemment, en 1955 par John McCarthy. En 1956, John McCarthy et ses collaborateurs ont organisé une conférence intitulée « Dartmouth Summer Research Project on Artificial Intelligence » qui a donné naissance au machine learning. L'IA est présente dans notre quotidien, elle est utilisée dans divers domaines (voir figure 2.16) [36].



**Figure 2. 16:** Domaine de l'intelligence artificiel [37]

## **2.5 Apprentissage profond**

L'apprentissage profond ou Deep Learning est une branche du Machine Learning utilisée pour former des systèmes informatiques appelés réseaux de neurones artificiels (ANN). Ces techniques reposent sur des algorithmes capables d'imiter les comportements du cerveau humain. Les algorithmes d'apprentissage profond mettent en œuvre plusieurs couches de neurones artificiels ou de nœuds avec diverses connexions entre eux. Les couches de nœuds sont reliées par plusieurs types de connexions. Ces connexions sont entraînées pour reconnaître et comprendre les caractéristiques d'un ensemble de données [38].

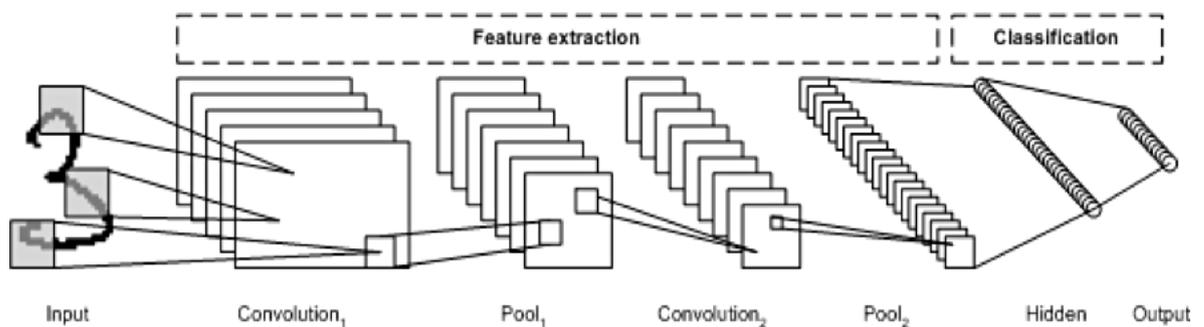
## **2.6 Architectures de l'apprentissage profond**

### **2.6.1 Réseaux neuronaux convolutifs (CNN)**

Un CNN est un réseau neuronal multicouche qui a été biologiquement inspiré par le cortex visuel animal. L'architecture est particulièrement utile dans les applications de traitement d'images. Le premier CNN a été créé par Yann LeCun en 1989, à l'époque l'architecture était axée sur la reconnaissance manuscrite des caractères, comme l'interprétation des codes postaux. En tant que réseau profond, les premières couches reconnaissent les caractéristiques (comme les bords), et les couches ultérieures recombinent ces caractéristiques en attributs de niveau supérieur de l'entrée.

L'architecture LeNet CNN est composée de plusieurs couches qui implémentent l'extraction des fonctionnalités puis la classification (voir la figure 2.17). L'image est

divisée en champs réceptifs qui alimentent une couche convolutive, qui extrait ensuite les caractéristiques de l'image d'entrée. La prochaine étape est la mise en commun, qui réduit la dimensionnalité des caractéristiques extraites (par le biais d'un échantillonnage descendant) tout en conservant les informations les plus importantes (généralement, par la mise en commun maximale). Une autre étape de convolution et de mise en commun est ensuite effectuée et alimente un perceptron multicouche entièrement connecté. La couche de sortie finale de ce réseau est un ensemble de nœuds qui identifient les caractéristiques de l'image (dans ce cas, un nœud par numéro identifié). Vous entraînez le réseau en utilisant la rétropropagation [39].



**Figure 2. 17:** Réseaux neuronaux convolutifs [39]

L'utilisation de couches de traitement profond, de convolutions, de pooling, et d'une couche de classification entièrement connectée a ouvert de nouvelles perspectives pour l'utilisation de réseaux neuronaux profonds dans diverses applications. En plus du traitement d'images, le CNN a été couronné de succès dans la reconnaissance vidéo ainsi que dans différentes tâches liées au traitement du langage naturel [39].

## 2.6.2 Réseaux neuronaux récurrents (RNN)

Le Réseau Neuronal Récurrent (RNN) est un type de réseau neuronal où la sortie de l'étape précédente est utilisée comme entrée pour l'étape actuelle. Dans les réseaux neuronaux traditionnels, toutes les entrées et sorties sont indépendantes les unes des autres, mais dans les cas où il est nécessaire de prédire le mot suivant d'une phrase, les mots précédents sont nécessaires, et donc il est nécessaire de se souvenir des mots précédents. C'est ainsi que le RNN est apparu, résolvant ce problème grâce à une couche cachée. La caractéristique principale et la plus importante du RNN est son état caché, qui mémorise certaines informations sur une séquence. Cet état est également appelé

état de mémoire car il se souvient de l'entrée précédente du réseau. Il utilise les mêmes paramètres pour chaque entrée, car il effectue la même tâche sur toutes les entrées ou couches cachées pour produire la sortie. Cela réduit la complexité des paramètres, contrairement à d'autres réseaux neuronaux [40].

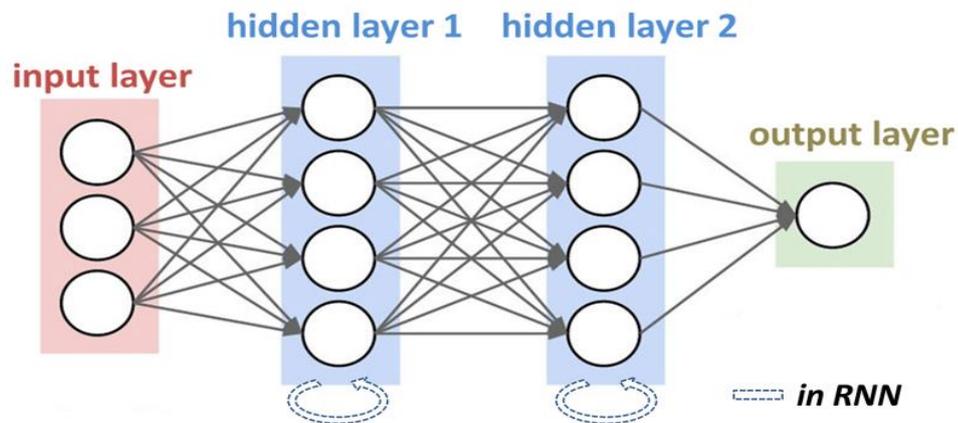


Figure 2. 18: Réseaux neuronaux récurrents [41]

## 2.7 Détection et reconnaissance d'objets à l'aide d'apprentissage profond

### 2.7.1 Définition

La détection d'objets est un domaine de vision par ordinateur qui a connu une évolution rapide. Cette technique combine la classification et la localisation d'objets. Un système de détection d'objet peut détecter, localiser et tracer l'objet (déterminer où se trouvent les objets dans une image donnée) et identifie la catégorie de cette dernière (personne, table, chaise, etc.). Il existe différents types d'algorithmes de détection d'objets, certains sont des techniques traditionnelles et d'autres des techniques modernes développées récemment. [42]

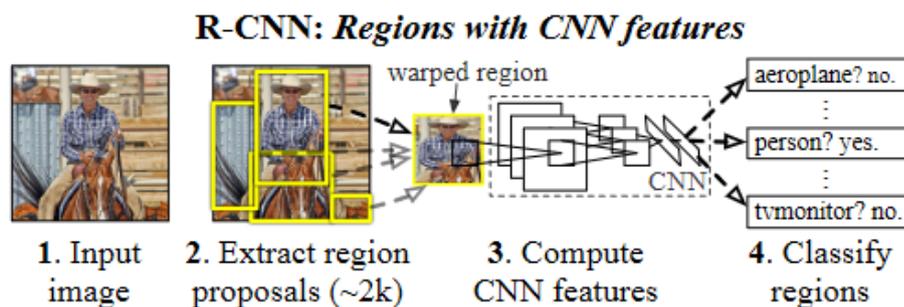
### 2.7.2 Modèles de détection par CNN

#### A- R-CNN

Ross Girshick [42] a proposé le modèle R-CNN de détection d'objets en 2014, qui est composé de trois modules :

- **Génération de propositions régionales** : indépendantes de la catégorie, qui définissent l'ensemble des détections candidates disponibles pour notre détecteur.
- **Extraction de caractéristiques** : le deuxième module est un grand réseau neuronal convolutif qui extrait un vecteur caractéristique de longueur fixe de chaque région.
- **Classification et localisation** : Le troisième module est un ensemble de SVM linéaires spécifiques à chaque classe.

Le modèle R-CNN a obtenu une précision moyenne (mAP) de 53,7% sur la base de données PASCAL VOC 2010. En comparaison, une approche utilisant une pyramide spatiale et une approche par sac de mots visuels a atteint une précision moyenne de 35,1% en utilisant les mêmes propositions de régions. Sur la base de données ILSVRC 2013 composé de 200 classes, la mAP de R-CNN s'élève à 31,4%, ce qui représente une nette amélioration par rapport à OverFeat, qui détenait précédemment le meilleur résultat avec 24,3% [42].



**Figure 2. 19** : Architecture de model R-CNN [42]

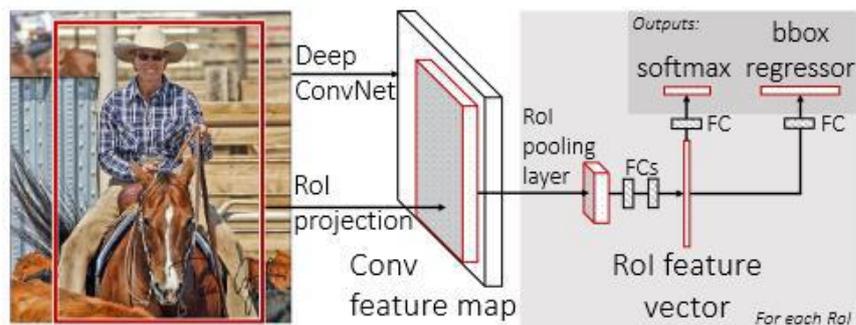
### **B- Fast R-CNN :**

Ross Girshick [42] a présenté un nouvel algorithme d'apprentissage qui remédie aux limitations du R-CNN et du SPPNet, tout en améliorant leur vitesse et leur précision. Cette méthode est connue sous le nom de R-CNN rapide (Fast R-CNN) car elle se distingue par sa rapidité d'entraînement et de test en comparaison. Ce réseau prend en entrée une image entière et un ensemble de propositions d'objets. Le réseau traite d'abord l'image entière avec plusieurs couches convolutives (conv) et de mise en commun maximale pour produire une carte de caractéristiques.

Ensuite, pour chaque proposition d'objet, une couche de mise en commun des régions d'intérêt (RoI) extrait un vecteur de caractéristiques de longueur fixe à partir de la carte

de caractéristiques. Chaque vecteur de caractéristiques est ensuite introduit dans une séquence de couches entièrement connectées (FC) qui se ramifient finalement en deux couches de sortie sœurs.

Le Fast R-CNN obtient un résultat de (66.1%) sur le VOC2010 et le meilleur résultat sur le VOC12 avec un a mAP de 65,7 % (et 68,4 % avec des données supplémentaires) [42].



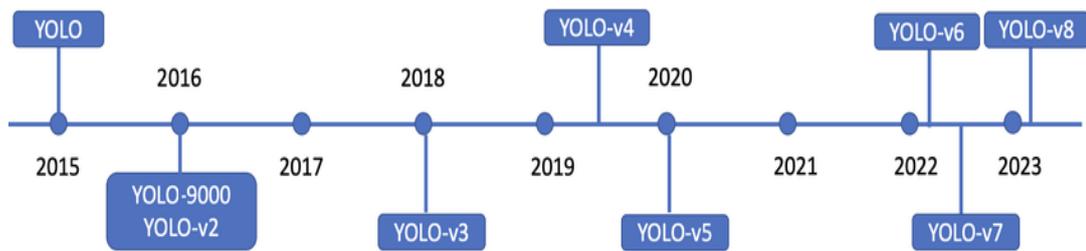
**Figure 2. 20:** Architecture de modèle Fast R-CNN [42]

#### **D- YOLO : You Only Look Once**

YOLO (You Only Look Once) est un modèle de détection d'objets populaire connu pour sa vitesse et sa précision. Il a été introduit par Joseph Redmon et al [43].

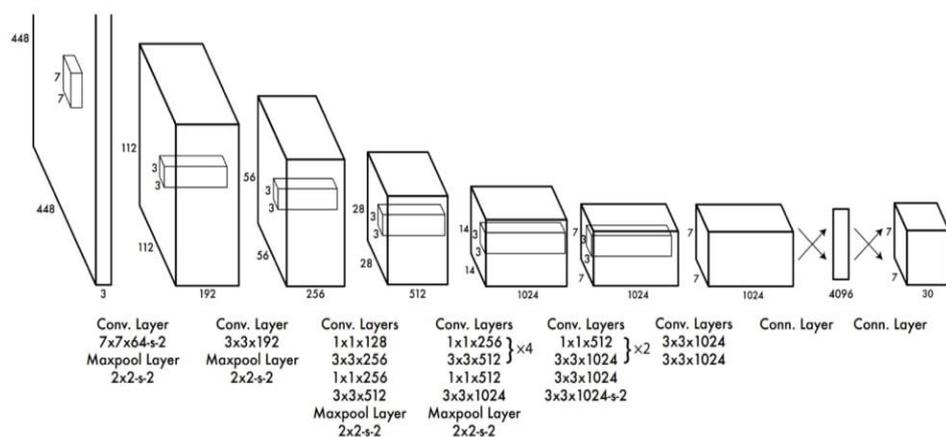
You Only Look Once (YOLO) propose l'utilisation d'un réseau neuronal de bout en bout qui effectue des prédictions à la fois pour les boîtes englobantes (bounding boxes) et les probabilités de classe. Cela diffère de l'approche adoptée par les algorithmes précédents de détection d'objets, qui réutilisaient des classificateurs pour effectuer la détection. Grâce à une approche radicalement différente de la détection d'objets, YOLO a réussi à atteindre des résultats de pointe, surpassant largement les autres algorithmes de détection d'objets en temps réel.

Plusieurs nouvelles versions du même modèle ont été proposées depuis la première version de YOLO en 2015, chacune s'appuyant sur son prédécesseur et l'améliorant. Voici une chronologie des développements de YOLO ces dernières années (voir la figure 2.21).



**Figure 2. 21 :** Chronologie des versions de YOLO (You Only Look Once) [44]

L’algorithme YOLO prend une image comme entrée, puis utilise un simple réseau de neurones à convolution profonde pour détecter les objets dans l’image. L’architecture du modèle CNN qui forme l’épine dorsale de YOLO est illustrée ci-dessous.



**Figure 2. 22 :** Architecture YOLO [43]

Les 20 premières couches de convolution du modèle sont pré-formées à l’aide d’ImageNet en branchant une couche temporaire de mise en commun moyenne et entièrement connectée. Ensuite, ce modèle pré-formé est converti pour effectuer la détection puisque des recherches précédentes ont montré que l’ajout de convolution et de couches connectées à un réseau pré-formé améliore les performances. La dernière couche entièrement connectée de YOLO prédit à la fois les probabilités de classe et les coordonnées des boîtes limitatives.

YOLO divise une image d’entrée en une grille de taille  $S \times S$ . Si le centre d’un objet tombe dans une cellule de la grille, cette cellule est responsable de la détection de cet objet. Chaque cellule de la grille prédit B boîtes englobantes et des scores de confiance pour ces boîtes. Ces scores de confiance reflètent à quel point le modèle est confiant que la boîte contient un objet et à quel point il est précis dans sa prédiction de la boîte.

YOLO prédit plusieurs boîtes de limitation par cellule de grille. Au moment de la formation, nous voulons qu'un seul prédicteur de boîte limite soit responsable de chaque objet. YOLO attribue un prédicteur pour être « responsable » de prédire un objet basé sur lequel la prédiction a le plus haut courant IoU (en anglais ; Intersection over Union) avec la vérité du terrain (en anglais ; ground truth). Cela conduit à une spécialisation entre les prédicteurs de boîte limite. Chaque prédicteur est mieux à prédire certaines tailles, rapports d'aspect, ou des classes d'objets, améliorant le score de rappel global [43].

## 2.8 Détection d'objets avec YOLO V5 :

Comme évoqué précédemment, YOLO est un algorithme de détection d'objets en temps réel, avec différentes versions disponibles. Pour notre étude, nous avons utilisé la version YOLOv5 en raison de sa vitesse et de sa précision. Dans cette partie nous allons discuter l'architecture du YOLOv5 et son fonctionnement.

Le réseau YOLO v5 dispose de trois parties importantes, backbone, neck et head (tête) (voir Figure 2.23).

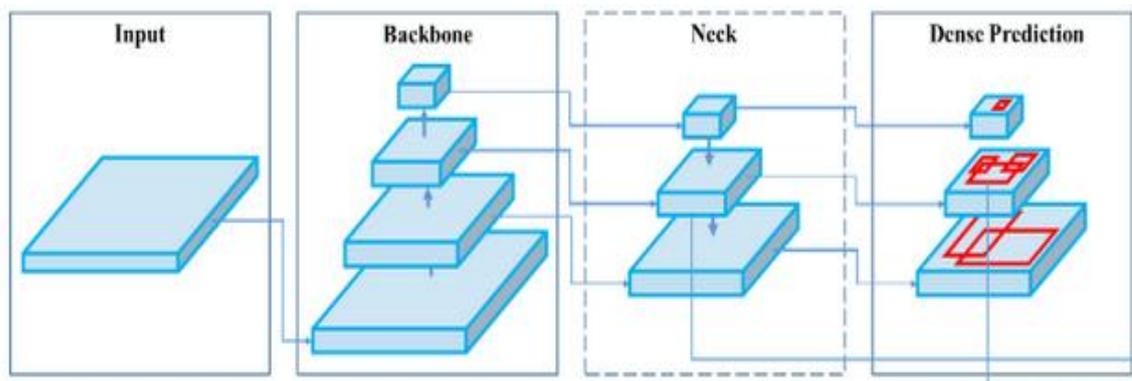


Figure 2. 23 : Architecture du YOLO v5 [45]

### A- Backbone :

Le "backbone" est un réseau pré-entraîné utilisé pour extraire une représentation riche en caractéristiques des images. Cela permet de réduire la résolution spatiale de l'image et d'augmenter sa résolution en termes de caractéristiques.

### B- Neck :

Le "neck" est utilisé pour extraire des pyramides de caractéristiques. Cela permet au modèle de bien généraliser aux objets de différentes tailles et échelles.

### C- Head :

La "head" est utilisée pour effectuer les opérations finales. Elle applique des boîtes d'ancrage (boxes) sur les cartes de caractéristiques et génère la sortie finale : classes, scores de présence d'objet et boîtes englobantes [45].

## 2.8.1 Fonctionnements du YOLO :

L'algorithme du YOLO fonctionne selon les quatre approches suivantes [46] :

- Blocs résiduels
- Régression des boîtes limitatives
- Intersection Over Unions (IOU)
- Suppression non maximale

### a) Blocs résiduels :

Cette première étape commence par diviser l'image originale (A) en cellules de grille  $N \times N$  de forme égale, où  $N$  dans notre cas est de valeur 4 comme le montre l'image de droite de la figure 2.23. Chaque cellule de la grille est responsable de la localisation et de la prévision de la classe de l'objet qu'elle couvre, ainsi que de la probabilité / valeur de confiance.

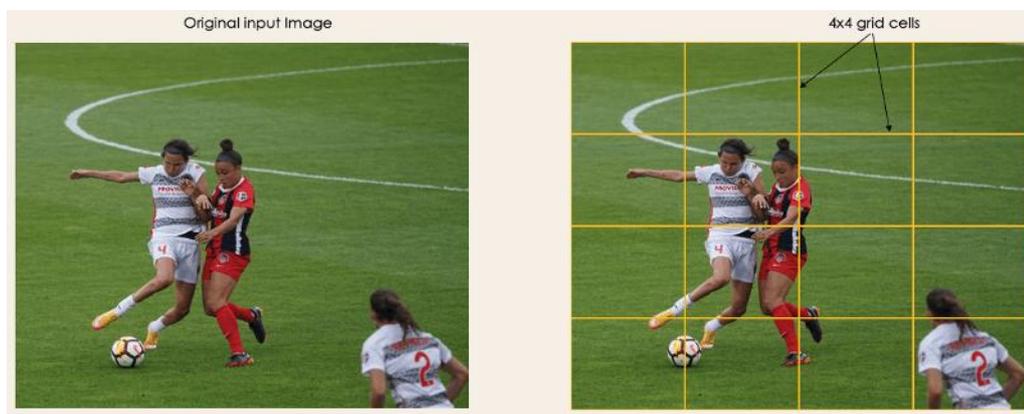


Figure 2. 24 : Division de l'image en  $N \times N$  grille [46]

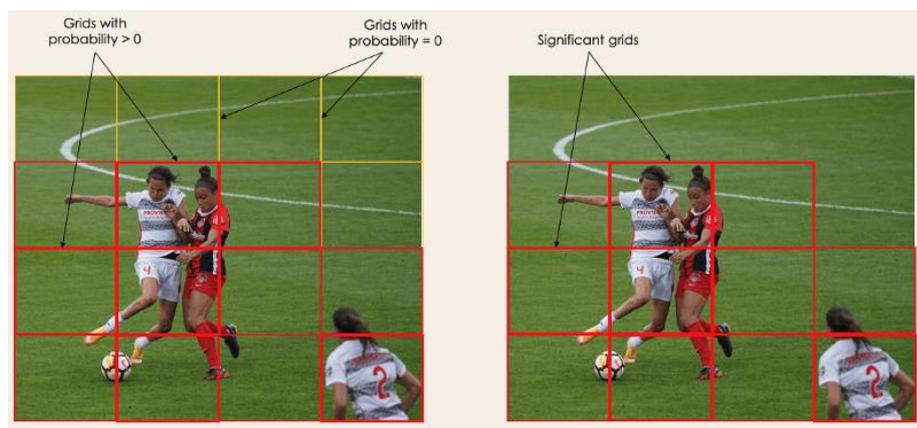
## b) Régression des boîtes limitatives :

L'étape suivante consiste à déterminer les boîtes de délimitation qui correspondent à des rectangles mettant en évidence tous les objets de l'image. Nous pouvons avoir autant de cases limitatives qu'il y a d'objets dans une image donnée.

YOLO détermine les attributs de ces boîtes de délimitation en utilisant un module de régression unique dans le format suivant, où Y est la représentation vectorielle finale pour chaque boîte de délimitation.

$$Y = [pc, bx, by, bh, bw, c1, c2]$$

**Pc** : correspond au score de probabilité de la grille contenant un objet. Par exemple, toutes les grilles en rouge auront un score de probabilité supérieur à zéro. L'image de droite est la version simplifiée puisque la probabilité de chaque cellule jaune est zéro (voir figure 2.24).



**Figure 2. 25** : Détermination de boîtes de délimitation [46]

**bx,by** : sont les coordonnées x et y du centre de la boîte de délimitation par rapport à la cellule de grille enveloppante.

**bh, bw** : correspondent à la hauteur et la largeur de la boîte de délimitation par rapport à la cellule de grille enveloppante.

**c1, c2** : correspondent aux deux classes Joueur et Ballon (voir figure 2.25).

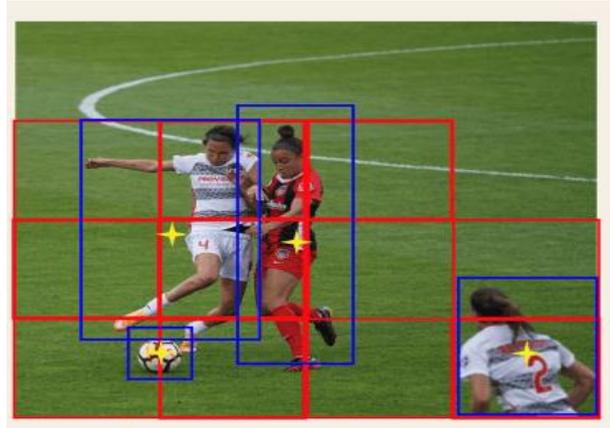


Figure 2. 26 : Boite de délimitation de chaque classe [46]

**c) Intersection over Unions (IoU) :**

La plupart du temps, un seul objet d'une image peut présenter plusieurs possibilités de prédiction, même si elles ne sont pas toutes pertinentes. Le but de l'IoU (une valeur comprise entre 0 et 1) est de jeter ces boîtes de grille pour ne garder que celles qui sont pertinentes. Voici la logique derrière cela :

L'utilisateur définit son seuil de sélection IoU, qui peut être, par exemple, 0,5.

Ensuite, YOLO calcule l'IoU de chaque cellule de grille qui est la zone d'intersection divisée par la zone de l'Union.

Enfin, il ignore les prédictions des cellules de grille ayant un IoU ≤ seuil et considère uniquement celles ayant un IoU > seuil.

Voici une illustration de l'application du processus de sélection de grille à l'objet en bas à gauche (voir figure 2.26). Nous pouvons observer que l'objet avait initialement deux candidats de grille, puis seulement "Grille 2" a été sélectionné à la fin.

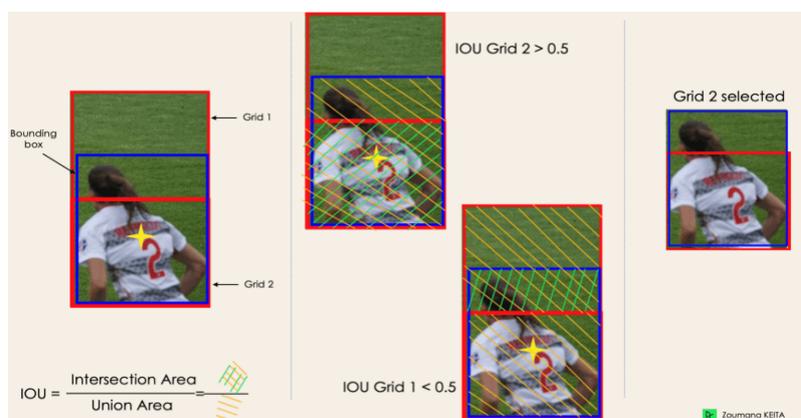


Figure 2. 27 : Sélection de grille 2 [46]

#### **d) Suppression non maximale :**

Fixer un seuil pour l'IOU n'est pas toujours suffisant, car un objet peut avoir plusieurs boîtes avec un IOU dépassant le seuil, et conserver toutes ces boîtes pourrait inclure du bruit. C'est ici que nous pouvons utiliser la suppression non maximale (NMS) pour ne garder que les boîtes ayant le score de probabilité de détection le plus élevé.

### **2.8.2 Avantages et inconvénients de YOLO V5**

- Il est environ 88% plus petit que YOLOv4 (27 Mo contre 244 Mo)
- Il est environ 180 % plus rapide que le YOLOv4 (140 FPS contre 50 FPS)
- Il est à peu près aussi précis que YOLOv4 sur la même tâche (0,895 mAP contre 0,892 mAP)
- Mais le principal problème est que pour YOLOv5 il n'y a pas de papier officiel a été publié comme les autres versions de YOLO. En outre, YOLO v5 est toujours en cours de développement et il y a des mises à jour fréquentes d'ultralytiques, les développeurs peuvent mettre à jour certains paramètres à l'avenir [47].

## **2.9 Conclusion**

Dans ce chapitre nous avons abordé les notions importantes qui sont en relation avec les techniques de traitement d'images utilisées, aussi qu'une vision générale sur les réseaux de neurones artificiels et ses structures. De plus nous avons présenté l'intelligence artificielle et quelques modèles de détection par CNN, notamment les R-CNN et fast R-CNN, plus particulièrement le modèle de YOLO V5 utilisé dans notre projet.

# Chapitre 3 Implémentation et résultats

---

## 3.1 Introduction

Dans ce dernier chapitre, nous allons mettre en œuvre ce que nous avons discuté dans les chapitres ci-dessus sur l'environnement de travail aussi le processus de l'entraînement en utilisant le modèle YOLO V5, ainsi que la partie du traitement d'image utilisé pour calculer les dimensions des billets, puis à l'analyse des résultats obtenus.

## 3.2 Environnement de travail

### 3.2.1 Spécifications de l'ordinateur portable :

Nous avons travaillé dans notre projet avec un ordinateur portable ayant les caractéristiques suivantes :

- Dell Latitude 7490
- Processeur : Intel i5 8ème génération
- Mémoire vive : 8 Go
- Stockage : 256 Gb SSD
- GPU : Intel ® UHD Graphics 620

### 3.2.2 Logiciels utilisés :

#### A- Roboflow :

Roboflow est une plate-forme de vision par ordinateur qui permet aux utilisateurs de créer des modèles de vision par ordinateur plus rapidement et avec plus de précision grâce à la fourniture de meilleures techniques de collecte de données, de prétraitement et de formation de modèles. Roboflow permet aux utilisateurs de télécharger des ensembles de données personnalisés, de dessiner des annotations, de modifier les orientations des images, de redimensionner les images, de modifier le contraste des

images et d'effectuer une augmentation des données. Il peut également être utilisé pour former des modèles.

Roboflow dispose également d'un outil de conversion d'annotations universel qui permet aux utilisateurs de télécharger et de convertir des annotations d'un format à un autre sans avoir à écrire de scripts de conversion pour les ensembles de données de détection d'objets [48].

### **B- Google Collab :**

Colaboratory, ou "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation.

Pour utiliser Google Colab, il suffit d'aller sur Google Drive puis cliquez sur nouveau puis sur "plus" et choisissez "". Si vous n'avez pas l'option « Colaboratoire », cliquez sur « Associer plus d'applications », puis recherchez « Colaboratoire » et cliquez sur Connexion [49].

Nous pouvons effectuer les opérations suivantes à l'aide de Google Colab :

- Écrire et exécuter du code en Python
- Créer/Télécharger/Partager des blocs-notes
- Importer/enregistrer des blocs-notes depuis/vers Google Drive
- Importer/publier des blocs-notes depuis GitHub
- Importer des ensembles de données externes
- Intégrer PyTorch, TensorFlow, Keras, OpenCV
- Service Cloud gratuit avec GPU gratuit [50]

### **3.2.3 Langage de programmation utilisé (Python) :**

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils

font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [51].

### **3.2.4 Bibliothèques utilisées :**

- **PyTorch :**

PyTorch est une bibliothèque d'apprentissage automatique pour Python utilisée principalement pour le traitement du langage naturel. Le logiciel open-source a été développé par les équipes d'intelligence artificielle [52].

- **OPENCV :**

OpenCV (Open Computer Vision) est une bibliothèque graphique. Elle est spécialisée dans le traitement d'images, que ce soit pour de la photo ou de la vidéo. Sa première version est sortie en juin 2000. Elle est disponible sur la plupart des systèmes d'exploitation et existe pour les langages Python, C++ et Java [53].

### **3.3 Entraînement à l'aide du YOLOv5 :**

Au cours de cette partie, notre objectif sera de créer une base de données, d'entraîner notre modèle YOLOv5, puis de réaliser des tests afin de recueillir des statistiques sur le fonctionnement de notre système.

#### **3.3.1 Création de base de données :**

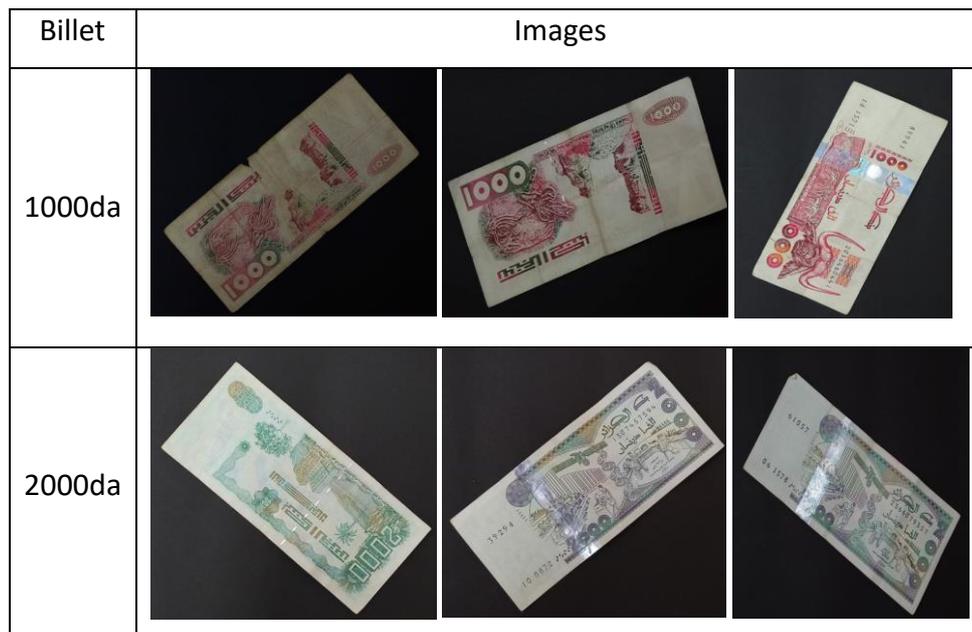
##### **A- Collecte d'images pour l'ensemble de données :**

La première étape de la création de base de données a été de collecter le maximum de billets de monnaies algériens, ensuite nous avons commencé à prendre des photos de ces billets placés sur du papier de couleur noir (fond uniforme) en utilisant des téléphones portables (voir figure 3.1).

Nous avons utilisé des billets algériens de 1000 et 2000 DA filmés selon différents angles. Nous avons créé 443 images, divisé comme l'indique le tableau 3.1 :

Billets	Photos de face	Photos de pile	Total de photos
1000 da	100	100	200
2000 da	100	113	213
Total	200	213	443

**Tableau 3. 1:** Nombre d'images créé pour chaque billet

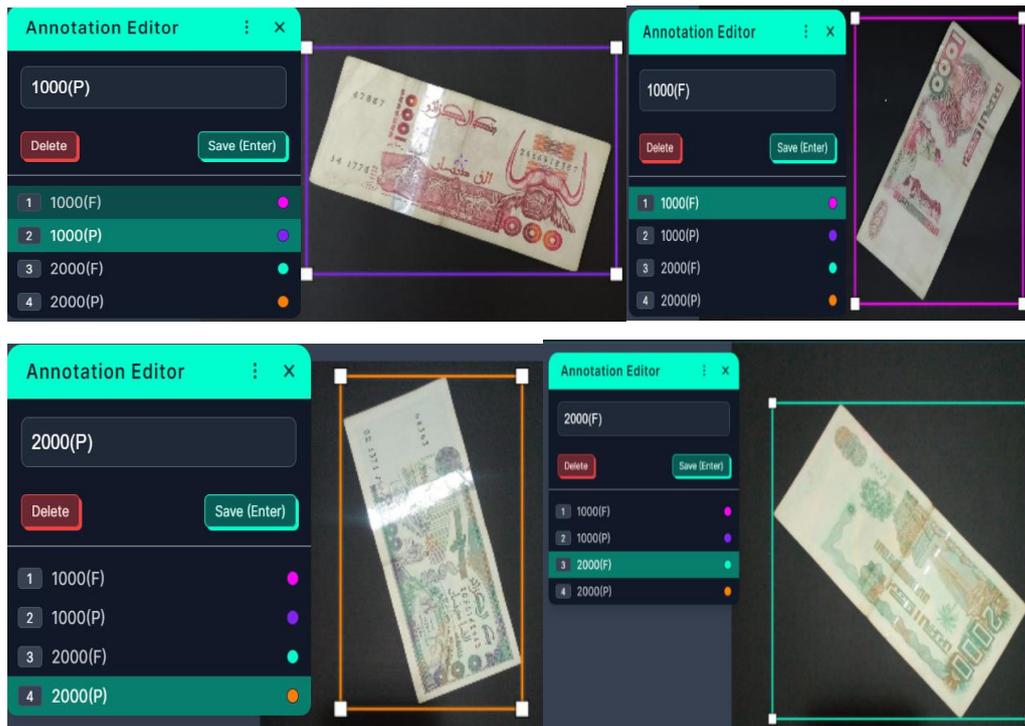


**Figure 3. 1:** Echantillons d'images de la base de données

### B- Annotation d'images :

Une fois les 443 images sont rassemblées, la prochaine étape consiste à annoter ces images afin d'extraire les informations nécessaires à chaque image. Notons que les classes utilisées sont au nombre de quatre classes ; 1000P, 1000F, 2000P et 2000F.

Nous avons utilisé le logiciel Roboflow, et voici quelques images après annotation comme l'indique la figure 3.2.



**Figure 3. 2 :** Annotation d'images sur Roboflow

### **C- Configuration de base de données**

Nous avons généré et divisé la base de données en 3 parties essentielles (voir figure 3.3).

- 79% (352 images) pour l'entraînement
- 19% (86 images) pour la validation
- 1% (5 images) pour le test



**Figure 3. 3 :** Division de la base de données

Nous avons aussi utilisé l'augmentation des données (data augmentation) par Roboflow (voir figure 3.4).

## Augmentation Options ✕

Augmentations create new training examples for your model to learn from.

### IMAGE LEVEL AUGMENTATIONS

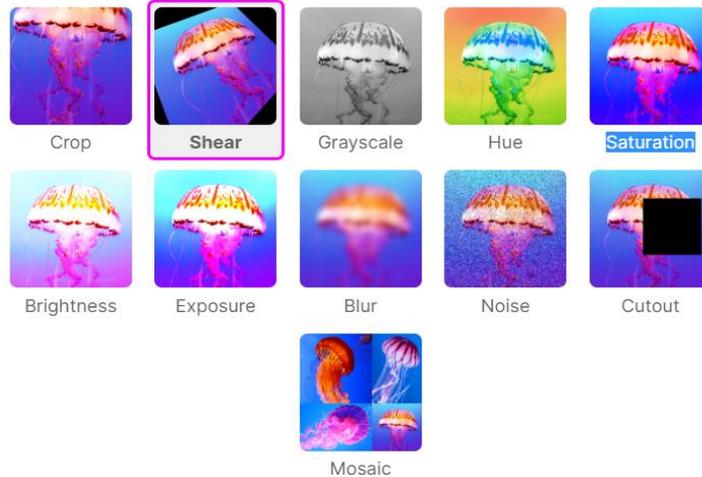


Figure 3. 4 : Types des augmentations de données sur Roboflow

#### Type des augmentations utilisé :

- Flip horizontal
- 30° and -30° rotation
- 90° rotation

Une fois l'augmentation est appliquée, nous avons réussi à augmenter le nombre d'images dans la base de données de 443 images à 1147 images.

Lorsque nous avons terminé l'augmentation de toutes les images, nous exportons notre base de données au format YOLOv5 Torch.



Figure 3. 5: Exportation de base de données

### 3.3.2 Entraînement et résultats :

Dans cette section, nous discuterons les étapes d'entraînement et enfin les résultats obtenus.



```

▶ from roboflow import Roboflow
rf = Roboflow(api_key="H1pLnYPXQm7JMNL3f74t")
project = rf.workspace("nadjjet-anabi-x82xq").project("detecion-4")
dataset = project.version(6).download("yolov5")

↳ loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in /content/datasets/detecion--4-6 to yolov5pytorch: 100%
Extracting Dataset Version Zip to /content/datasets/detecion--4-6 in yolov5pytorch:: 100%

```

**Figure 3. 8 :** Téléchargement de base de données sur Colab

La prochaine étape consiste à commencer l'entraînement (voir figure 3.9).

```

Model summary: 214 layers, 7030417 parameters, 7030417 gradients, 16.0 GFLOPs
Transferred 343/349 items from yolov5s.pt
AMP: checks passed ✓
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias
augmentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clahe_limit=(1, 16))
train: Scanning /content/datasets/detecion--4-9/train/labels... 1056 images, 3 backgrounds, 0 corrupt: 100% 1056/1056 [00:00<00:00, 210.28it/s]
train: New cache created: /content/datasets/detecion--4-9/train/labels.cache
train: Caching images (0.5GB ram): 100% 1056/1056 [00:05<00:00, 210.28it/s]
val: Scanning /content/datasets/detecion--4-9/valid/labels... 86 images, 0 backgrounds, 0 corrupt: 100% 86/86 [00:00<00:00, 87.00it/s]
val: New cache created: /content/datasets/detecion--4-9/valid/labels.cache
val: Caching images (0.0GB ram): 100% 86/86 [00:00<00:00, 87.00it/s]

AutoAnchor: 3.52 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✓
Plotting labels to runs/train/exp/labels.jpg...
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/exp
Starting training for 150 epochs...

```

**Figure 3. 9 :** Début d'entraînement

Nous avons attendu la fin de l'entraînement. Cet entraînement prend une durée de 24min24sec avec un GPU de 1.8GB. La précision que nous avons obtenu pour chaque classe et les graphes de la performance de notre entraînement sont représentés par figures 3.10 et 3.11 :

```

150 epochs completed in 0.404 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs

```

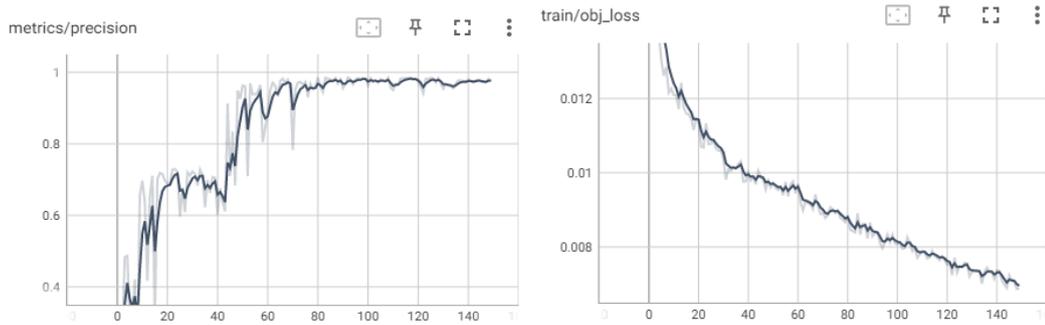
Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	86	86	0.982	0.988	0.983	0.981
1000-F-	86	21	0.989	0.952	0.98	0.98
1000-P-	86	23	0.998	1	0.995	0.992
2000-F-	86	20	0.985	1	0.995	0.99
2000-P-	86	22	0.955	1	0.964	0.964

```

Results saved to runs/train/exp

```

**Figure 3. 10 :** Précision obtenue pour chaque classe



**Figure 3. 11** : Graphes de la performance et la perte

D'après le graphe obtenu dans la figure 3.11 de précision, il est clair que notre niveau de précision maximal a été atteint après environ 75 epochs.

De plus, le graphe de la perte a diminué en fonction du nombre des epochs jusqu'à sa valeur minimale aux environ de 140 epochs. Ces résultats d'entraînement montrent bien que les performances obtenues sont satisfaisantes en utilisant le modèle YOLOV5.

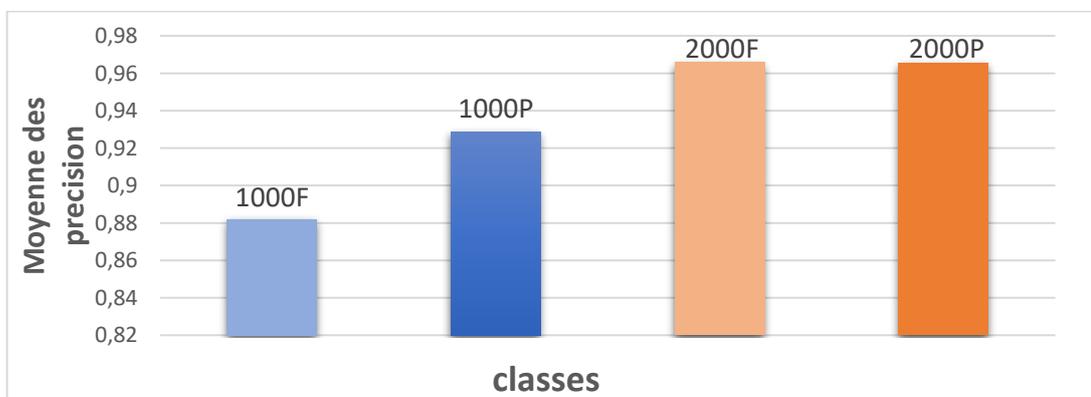
### 3.3.3 Test du modèle YOLOv5 :

Pour tester le modèle YOLOv5, une base de données test a été créée (en anglais ; test set). Cette base de 400 nouvelles images, ce qui représente 100 images/classe est testé les poids synaptiques (en anglais ; synaptic weights) déjà téléchargés.

Nous avons fait le test sur python avec une instruction de détection :

« python detect.py --weights best.pt --img 416 --conf 0.1 --source ../yolov5-master »

La figure 3.12 et le tableau 3.2 montrent les résultats de détection obtenus pour chaque classe des billets de la base de test.



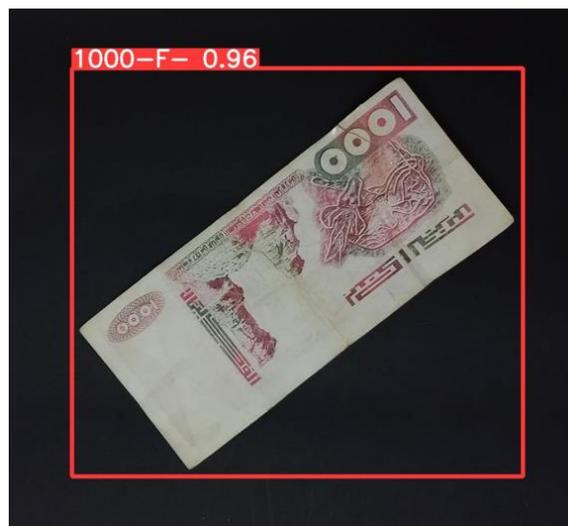
**Figure 3. 12** : Résultats de détection pour les quatre classes

Classe	1000F	1000P	2000F	2000P
Nombre d'image testé	100	100	100	100
Moyenne des Précisions	88,19%	92,86%	96,58%	96,55%

**Tableau 3. 2 :** Résultats de détection pour les quatre classes

Nous remarquons que la probabilité de détection de billets 2000 DA est plus élevée que les billets de 1000 DA, ceci est dû à plusieurs raisons telles que ; le vieillissement et la saleté de billets testés.

Les figures 3.13, 3.14, 3.15 et 3.16 montrent la détection de billets et leurs scores par le modèle proposé.



**Figure 3. 13:** Résultat de test 1000F



**Figure 3. 14:** Résultat de test 1000P



**Figure 3. 15 :** Résultat de test 2000F



**Figure 3. 16 :** Résultat de test 2000P

### **3.4 Détermination du rapport (Longueur/Largeur)**

Dans le but d'extraire plus d'informations des billets détectés tels que leurs dimensions (longueur et largeur) et orientations, lesquelles le modèle d'apprentissage YOLOv5 ne peut pas d'extraire, nous avons introduit une partie relative à la détermination des dimensions à l'aide des techniques du traitement d'images afin de calculer le rapport  $R=(\text{longueur}/\text{largeur})$ .

✓ **Etape 1 :**

Extraction de la zone de billet à partir de la fenêtre englobante détecté par le YOLOV5 :

Pour ce faire, nous devons extraire l'intérieur de la boîte englobante de différentes couleurs, cette couleur dépend de la classe détectée :

-Pour 1000 DA face, cette couleur est de : (R=255, V=58, B=56)

-Pour 1000 DA pile, cette couleur est de : (R=255, V=158, B=151)

-Pour 2000 DA face, cette couleur est de : (R=255, V=114, B=27)

-Pour 2000 DA pile, cette couleur est de : (R=255, V=176, B=33)

Nous avons constaté que la couleur rouge R=200 est suffisante pour extraire la fenêtre englobante.

Cette couleur est utilisée comme seuillage pour extraire la fenêtre englobante et convertir la couleur correspondante en blanc, selon la relation suivante :

$$g = \begin{cases} 255 & \text{si } F(x, y, 2) \geq 255 \\ 0 & \text{sinon} \end{cases}$$

Avec F l'image avec boîte englobante (voir figure 3.17) et g l'image binaire (voir figure 3.18).



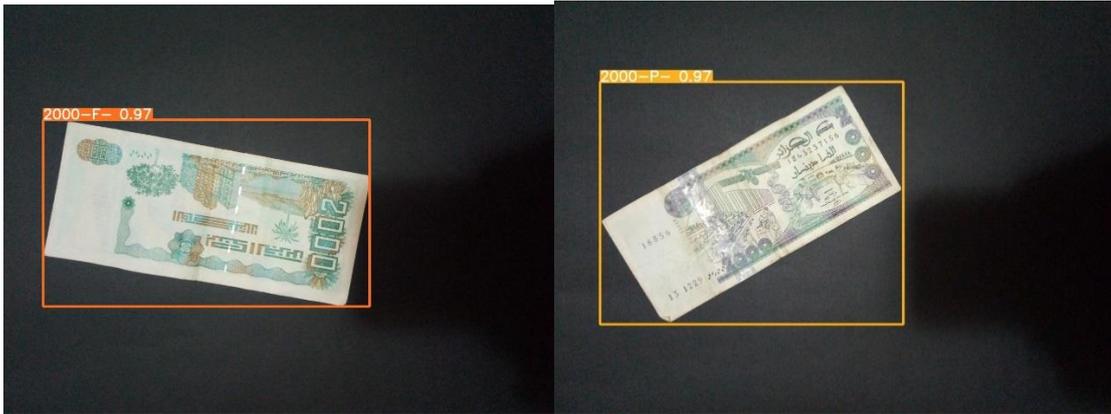


Figure 3. 17 : Différentes couleurs pour chaque classe.

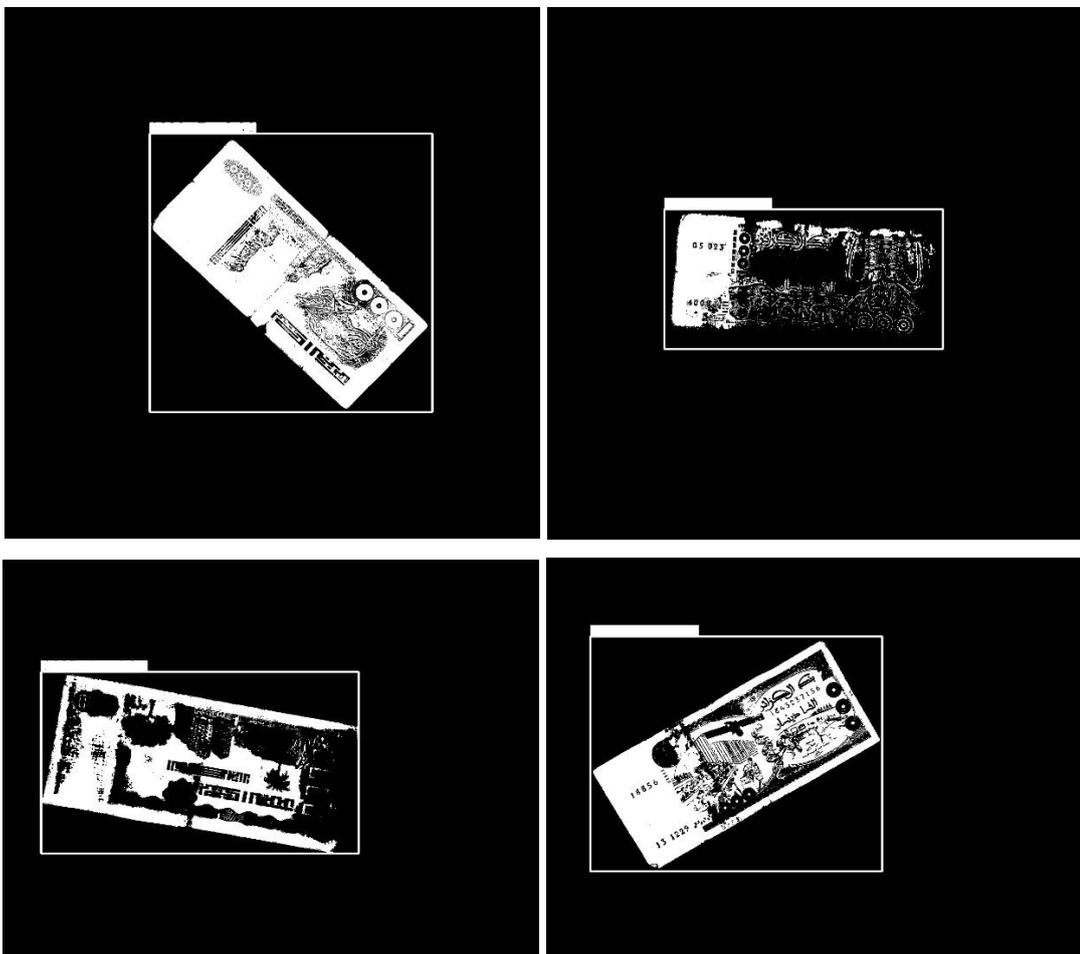


Figure 3. 18 : Seuillage des images de la figure 3.17 en utilisant l'équation.

Nous utilisons les images de la figure 3.18 pour extraire uniquement l'objet billet, nous avons introduit une procédure basée sur la projection horizontale et verticale de profile (voir figure 3.19).

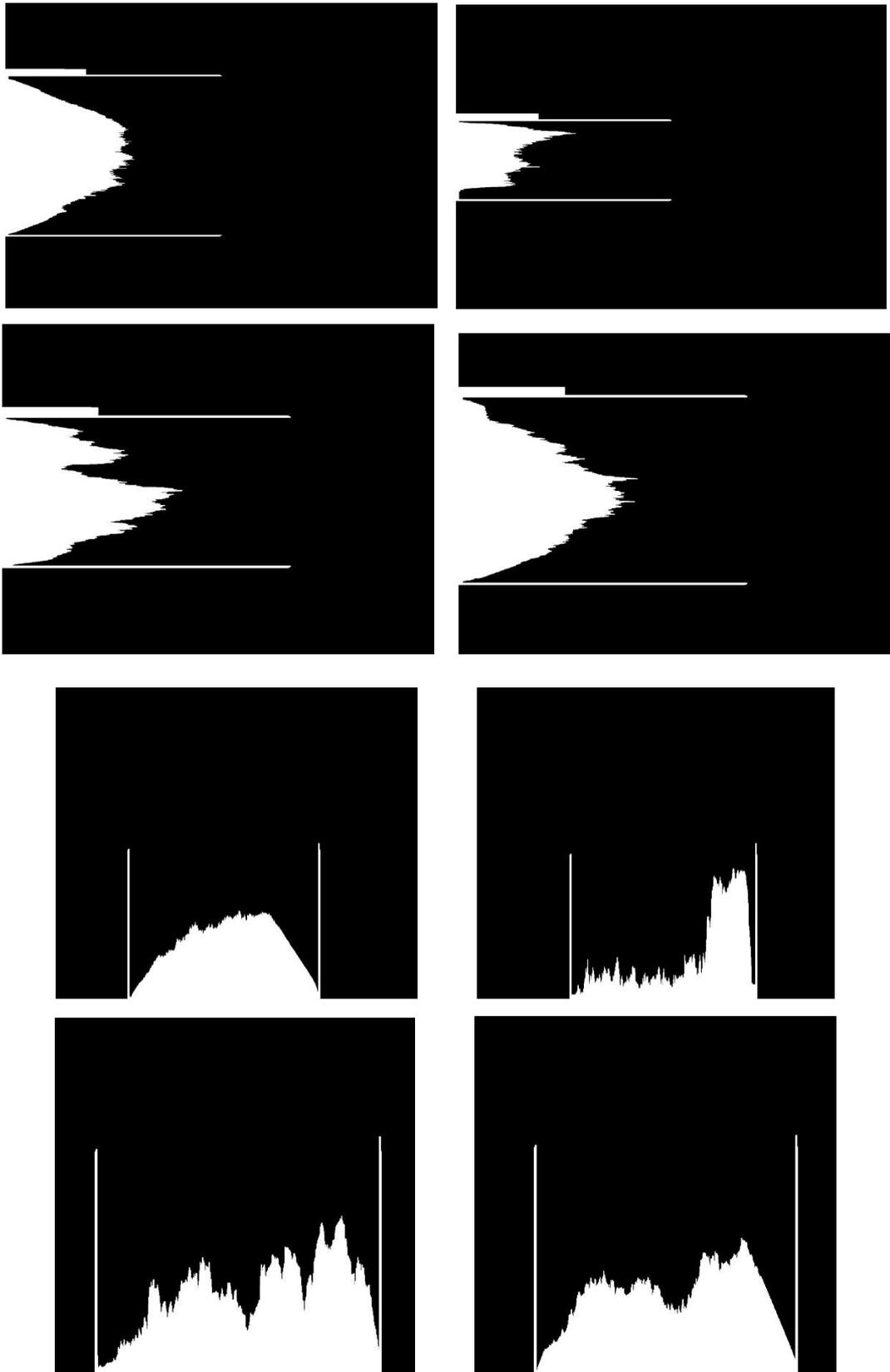


Figure 3. 19 : Résultats de la projection horizontale et verticale des images de la figure 3.18

Cette procédure nous a permis de déterminer le début et la fin dans le cas horizontale et verticale de la boîte englobante, ce qui nous a permis d'extraire l'objet billet de l'intérieur de la fenêtre englobante (voir figure 3.21)

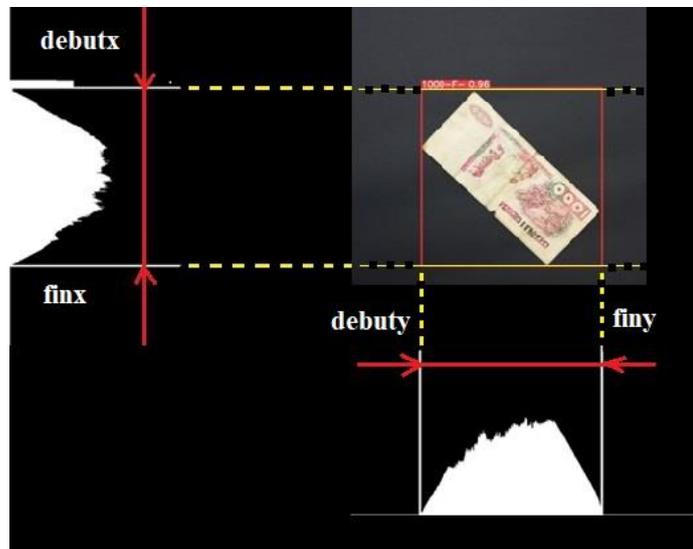


Figure 3. 20 : Résultats de de découpage d'une image de la figure 3.18



Figure 3. 21 : Extraction de l'image billet

## ✓ Etape 2 :

Une fois l'image de billet est extraite, nous appliquons des techniques du traitement d'image comme suit :

### A- Conversion d'image couleur en niveau de gris

Avec un code python à l'aide de la bibliothèque OPENCV et l'utilisation de cette instruction : `Cv2.cvtColor(image, COLOR_BGR2GRAY)` (voir figure 3.22).

Nous pouvons appliquer la conversion en niveau de gris.



Figure 3. 22 : Conversion de l'image billet couleur en niveau de gris

### B- Filtrage Gaussien

Notons que l'image à niveaux de gris de taille HxW a été ajustée en lui ajoutant 20 pixels à chaque côté, ce qui a donné une taille de  $(H+2 \times 20) \times (W+2 \times 20)$ . A l'image ajustée, nous avons utilisé l'instruction `Cv2.GaussianBlur` du filtre gaussien afin de réduire le bruit dans l'image (voir figure 3.23). Le filtre gaussien utilisé est de taille 3x3.

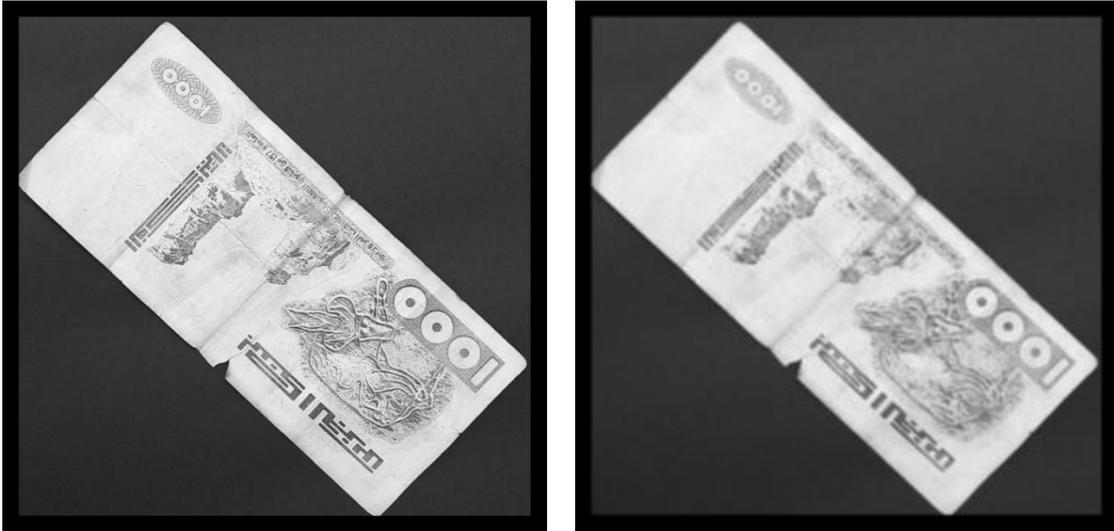


Figure 3. 23 : Application du filtre gaussien de taille 3x3

### C- Binarisation d'image

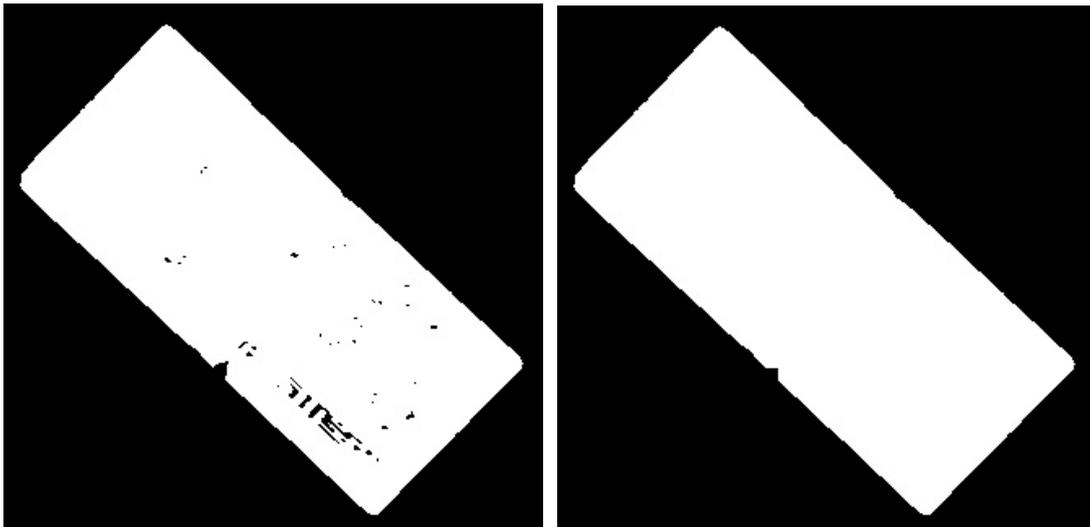
Dans cette étape une opération de binarisation globale basée sur la méthode d'Otsu est appliquée sur l'image filtrée de la figure 3.23. Nous avons utilisé l'instruction `cv2.threshold(img,0,255, cv2. THRESH_BINARY+ cv2. THRESH_OTSU)` de la bibliothèque **OpenCV** (voir figure 3.24).



Figure 3. 24 : Binarisation de l'image filtrée

### D- Fermeture :

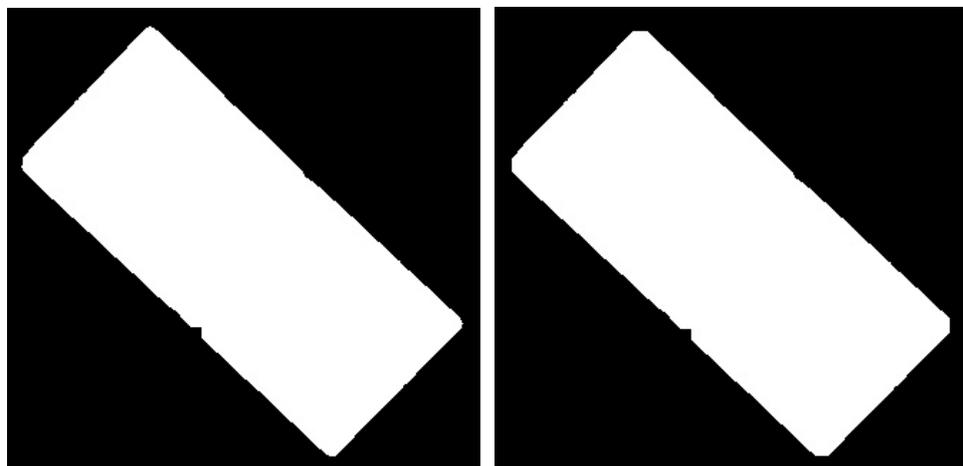
Afin d'éliminer les taches noires trouver sur la partie billet, nous avons utilisé l'instruction suivante : `Cv2.morphologyEx(img, cv2. MORPH_CLOSE)` (voir figure 3.25).



**Figure 3. 25 :** Application de la fermeture morphologique avec un élément structurant de taille 11x11

### E- Ouverture :

Afin d'éliminer les taches blanches sur l'arrière-plan de couleur noir, nous avons utilisé une instruction d'ouverture morphologique, **Cv2.morphologyEx(img, cv2.MORPH\_OPEN)**. (Voir figure 3.26)



**Figure 3. 26 :** Application de l'ouverture morphologique avec un élément structurant de taille 11x11

### F- Calcul les dimensions :

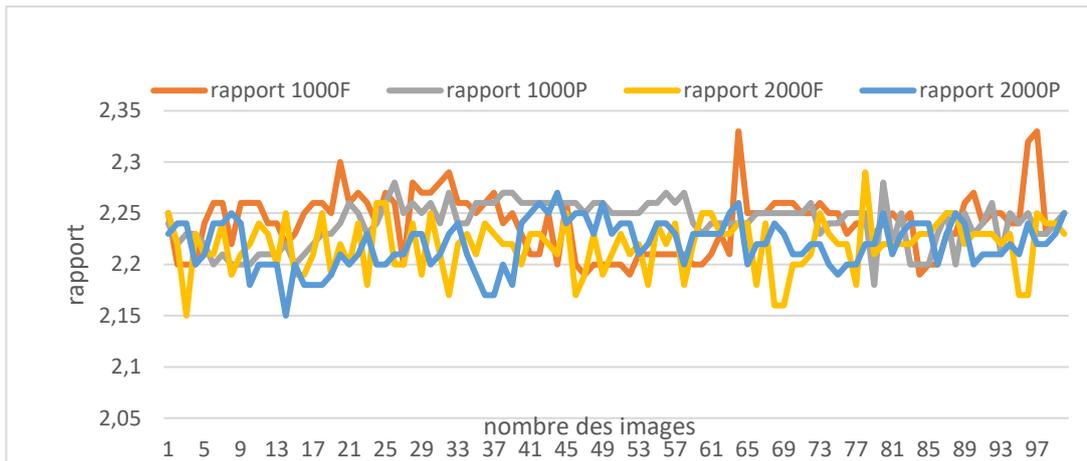
Pour déterminer les dimensions de billets, il est important de trouver les contours, et pour ce faire, nous avons utilisé cette instruction : **cv2.findcontours**

Une fois les contours sont détectés, nous avons utilisé d'autre instruction pour trouver les dimensions concernant la largeur et la hauteur ,

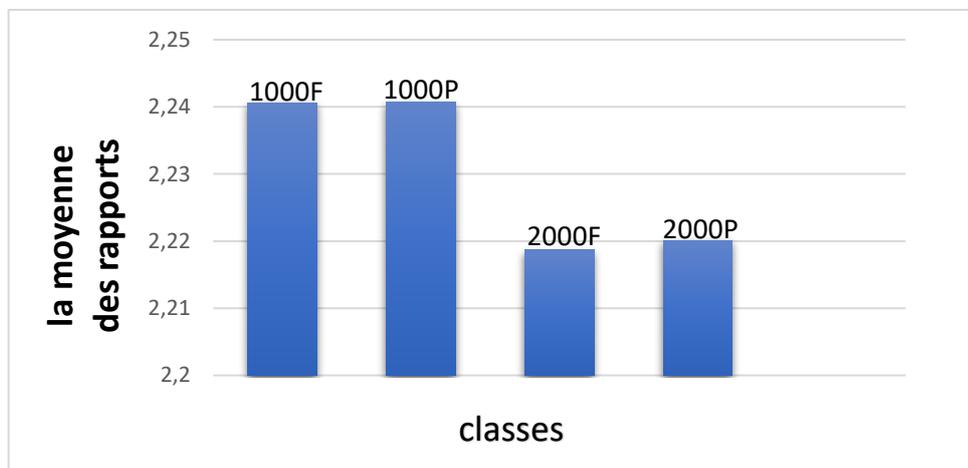
**cv2.minAreaRect(contours).** Après avoir trouvé les dimensions (largeur et hauteur), nous avons calculé le rapport comme suit :

$$\text{Rapport} = \text{largeur}/\text{hauteur}$$

Les figures 3.27 et 3.28 regroupent les résultats des rapports obtenus après l'application du programme sur billets de l'ensemble de test.



**Figure 3. 27 :** Résultats des rapports obtenus de l'ensemble de test



**Figure 3. 28 :** Moyennes des rapports obtenues pour chaque classe

Après les résultats de rapport obtenus, nous avons pris un échantillon de 10 nouveaux billets de 1000 DA et 2000 DA et nous avons mesuré leurs dimensions (par une règle) et calculé leurs rapports manuellement, afin de les comparer avec les résultats précédents.

Nous avons constaté que les dimensions des 10 billets sont fixes de valeur :

Largeur=16,1 , hauteur=7,1 , le rapport=2,26.

Puis, nous avons calculé l'erreur entre la moyenne des rapports mesurés par la méthode de traitement d'image et le rapport réel mesuré manuellement (voir table 3.3) :

Erreur=rappel réel-rappel mesuré

	Rappel réel	Moyenne de Rappel mesuré	L'erreur
1000F	2,26	2,24	0,02
1000P	2,26	2,24	0,02
2000F	2,26	2,21	0,05
2000P	2,26	2,22	0,04

**Tableau 3. 3:** Erreur entre le rapport réel et mesuré

Nous avons remarqué que l'erreur est entre 0.02 et 0.05, ce qui montre que notre méthode de traitement d'image est acceptable, car le rapport mesuré et le rapport réel sont proches et presque identiques.

### 3.5 Conclusion

Dans ce dernier chapitre, nous avons présenté les étapes d'entraînement de notre modèle YOLOV5 et tester ce dernier sur de nouvelles images de billets de monnaies algériens. Les résultats obtenus, montrent que les performances du modèle proposé sont satisfaisantes. La partie du traitement d'images utilisée pour l'extraction de l'objet billet, a aussi montré ses performances et nous avons obtenus des rapports proches du billet réel avec une erreur entre 2% et 5%.

## Conclusion générale

---

Dans ce mémoire, nous avons abordé l'étude sur la reconnaissance et la détection des billets de monnaie algériens en utilisant des techniques d'apprentissage profond et de traitement d'image. Nous avons étudié un algorithme d'apprentissage capable de reconnaître les différents billets de monnaie algériens.

Nous avons créé une base de données et l'avons annotée à l'aide du logiciel Roboflow en utilisant des boîtes englobantes. Ensuite, nous avons procédé à l'augmentation des images afin de disposer d'une variété d'images utilisables pour l'entraînement.

Comme modèle d'apprentissage, nous avons utilisé le modèle YOLOv5, pour détecter les billets dans des images réelles. Grâce à l'entraînement sur un ensemble de données étiquetées, le modèle développé a atteint une précision totale satisfaisante de 98,1%.

Pour tester la précision de notre modèle nous avons pris de nouvelles images, et les tester avec les poids synaptiques (synaptic weights) déjà obtenu, les résultats sont satisfaisants pour chaque classe des billets.

Ensuite, nous avons appliqué des techniques du traitement d'image pour extraire des caractéristiques spécifiques des billets, leurs dimensions (largeur et hauteur) lesquelles le modèle YOLOv5 ne peut pas extraire. Ces informations ont contribué à améliorer la précision de la reconnaissance des billets.

Il est important de noter que notre étude présente certaines limites. Par exemple, il peut rencontrer des difficultés lors de la détection de billets endommagés ou fortement froissés. L'exactitude de la reconnaissance peut être affectée par des variations de luminosité et d'angle de prise de vue.

Notre travail ouvre la voie à des applications pratiques dans divers domaines tels que les systèmes de paiement automatisés, les automates bancaires. En automatisant le

processus de reconnaissance des billets de monnaie algériens, notre étude permet de gagner du temps, de réduire les erreurs humaines et d'améliorer l'efficacité des opérations financières.

Pour des perspectives d'amélioration, il serait intéressant d'essayer d'autres modèles d'apprentissage profond pour améliorer plus de précision, et développer des applications qui fonctionnent sur un ordinateur portable ou un appareil Android, possibilité d'ajouter une sortie audio sur l'application développée.

## Bibliographie

---

- [1] العملة الورقية , <https://fbs.ae/glossary/flat-currency> , mai 2023.
- [2] Le monnaie dans le monde, <https://blog.remitly.com/fr/devises/monnaies-monde/> , mai 2023.
- [3] La première monnaie de papier. <https://monnaiesdumonde.net/content/15-la-premiere-monnaie-de-papier> , mai 2023
- [4] first notes european, <https://www.nbbmuseum.be/en/resources/first-notes-european-bank-swedish-product> , mai 2023
- [5] Redmond, P., & Redmond, P. (2022, 12 décembre). The Historie of Paper Money - Clear Junction. Clear Junction. <https://clearijunction.com/blog/the-history-of-paper-money/> , mai 2023
- [6] 2022. العملات الجزائرية القديمة. ضحى حماده , [www.almsal.com](http://www.almsal.com) , mai 2023.
- [7] algeria-banknotes, <https://www.numizon.com/> , mai 2023
- [8] notre histoire à travers le billet. 2022, <https://www.bank-of-algeria.dz/> , mai 2023
- [9] دينار جزائري , <https://www.marefa.org> , mai 2023
- [10] Numista, Algerian banknotes. 2007. <https://en.numista.com/catalogue/algerie-1.html> . mai 2022
- [11] Pérez, D. M. P., & Corrochano, E. H. (2018). Recognition System for Euro and Mexican Banknotes Based on Deep Learning with Real Scene Images. Computación Y Sistemas, 22(4). <https://doi.org/10.13053/cys-22-4-3079>

- [12] Joshi, R.C., Yadav, S., & Dutta, M.K. (2020). YOLO-v3 Based Currency Detection and Recognition System for Visually Impaired Persons. 2020 International Conference on Contemporary Computing and Applications (IC3A), 280-285.
- [13] Zhang, Q., & Yan, W. (2018). Currency Detection and Recognition Based on Deep Learning. <https://doi.org/10.1109/avss.2018.8639124> , mai 2023
- [14] Veeramsetty, V., Singal, G., & Badal, T. (2020). Coinnet : platform independent application to recognize Indian currency notes using deep learning techniques. Multimedia Tools and Applications, 79, 22569 - 22594.
- [15] Mittal, S., & Mittal, S. (2018, February). Indian banknote recognition using convolutional neural network. In 2018 3rd International Conference On Internet of Things : Smart Innovation and Usages (IoT-SIU) (pp. 1-6). IEEE.
- [16] GeeksforGeeks. Harshit saini, (2019). Projection Profile method. GeeksforGeeks.
- [17] Anagnostopoulos, C., et al. (2008). License Plate Recognition From Still Images and Video Sequences: A Survey. IEEE Transactions on Intelligent Transportation Systems, 9(3), 377–391. <https://doi.org/10.1109/tits.2008.922938> . Mai 2023
- [18] Mr. Namane Abderrahmane, Cour traitement d'image
- [19] Introduction au traitement des images et à la stéréo-vision - Seuillage d'une image. (n.d.). <https://www.traimaocv.fr/CoursStereoVision/co/Seuillage.html> , juin 2023
- [20] Cayla, B. Traitement d'images, Seuillage d'image.2022. <https://datacorner.fr/image-processing-3/> . Juin 2023
- [21] Wikiwand - Méthode d'Otsu. (n.d.). Wikiwand. [https://www.wikiwand.com/fr/M%C3%A9thode\\_d'Otsu](https://www.wikiwand.com/fr/M%C3%A9thode_d'Otsu) . Juin 2023
- [22] Boumehira Ossama, Ferhat Abdelfatah, système de reconnaissance des pièces de monnaies algériennes par l'apprentissage profond, mémoire de Master , système embarqué , Université Saad Dahleb, 2022

- [23] GeeksforGeeks. (2020). Image Edge Detection Operators in Digital Image Processing. GeeksforGeeks. <https://www.geeksforgeeks.org/image-edge-detection-operators-in-digital-image-processing/> . Juin 2023
- [24] Vikram M . (2022). Comprehensive Guide to Edge Detection Algorithms. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2022/08/comprehensive-guide-to-edge-detection-algorithms/> . Juin 2023
- [25] Sauget, M.2007. Parallélisation de problèmes d'apprentissage par des réseaux neuronaux artificiels. <https://www.researchgate.net/publication/30517821> . Juin 2023
- [26] C. S. Von Bartheld, et all. "The Search for True Numbers of Neurons and Glial Cells in the Human Brain: A Review of 150 Years of Cell Counting HHS Public Access," J Comp Neurol, vol. 524, no. 18, pp. 3865–3895, 2016, doi: 10.1002/cne.24040.  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5063692/> .Juin 2023
- [27] C. Touzet, les réseaux de neurones artificiels, introduction au connexionnisme. EC2, 1992.
- [28] Jumelle, M. (n.d.). Réseaux de neurones : tout comprendre. Formation Tech et Data en ligne | Blent.ai. <https://blent.ai/blog/a/reseaux-de-neurones-tout-comprendre> .Juin 2023
- [29] Kaadoud, I. C. (2018, October 8). Reprenons les bases : Neurone artificiel, Neurone biologique - Intelligence mécanique. Intelligence Mécanique.
- [30] Ikram Chraïbi Kaadoud & André Garenne.2018. Architecture des réseaux de neurones : Réseaux de neurones artificiels classiques (2/3),  
<https://scilogs.fr/intelligence-mecanique> .Juin 2023
- [31] Rédac, T.2022.Perceptron : qu'est-ce que c'est et à quoi ça sert ? Formation Data Science | DataScientest.com. <https://datascientest.com/perceptron> . Juin 2023
- [32] Sauget, M.2007. Parallélisation de problèmes d'apprentissage par des réseaux neuronaux artificiels. <https://www.researchgate.net/publication/30517821> . Juin 2023
- [33] Pr. Mustapha BOURAHLA,2021, Réseaux de neurones et apprentissage automatique Département d'Informatique, Université de M'Sila

- [34] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- [35] ] Belaidi, N.2022. L'apprentissage supervisé : définition et exemples. Consulté à l'adresse <https://blent.ai/blog/a/apprentissage-supervise-definition> , Juin 2023
- [36] NetApp.2019. Intelligence artificielle : définition et utilisations | NetApp. [www.netapp.com. https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/](https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/) , juin 2023
- [37] Artificial Intelligence Tutorial. 2023.[https://www.tutorialspoint.com/artificial\\_intelligence/](https://www.tutorialspoint.com/artificial_intelligence/), juin 2023
- [38] Cobbaï.. Deep learning | définition de l'apprentissage profond - Cobbaï.2022. <https://www.cobbai.com/blog/deep-learning-definition> ,juin 2023
- [39] Samaya Madhavan IBM Developer.2021. <https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/> ,juin 2023
- [40] GeeksforGeeks.2023. Introduction to Recurrent Neural Network. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/> , juin 2023
- [41] Shukai Ma et all.2019. Classification and prediction of wave chaotic systems with machine learning techniques.mai 2023
- [42] Mesbah Fethia , Détection d'objets par Deep Neural Network à l'aide du modèle YOLO en temps réel, mémoire de Master , Informatique , Université 8 Mai 1945, 2021
- [43] R. Kundu, "YOLO : Algorithm for Object Detection Explained [+Examples]," V7, Apr. 20, 2023. <https://www.v7labs.com/blog/yolo-object-detection#what-is-yolo> . Juin 2023
- [44] Zhang, Youshan. (2023). Stall Number Detection of Cow Teats Key Frames
- [45] Imane, C. (2022). YOLO v5 model architecture [Explained]. OpenGenus IQ : Computing Expertise & Legacy. <https://iq.opengenus.org/yolov5/> . Juin 2023
- [46] Keita, Z.2022, YOLO Object Detection Explained. <https://www.datacamp.com/blog/yolo-object-detection-explained#rdl> . Juin 2023

- [47] Garg, A.2023. How to Use Yolo v5 Object Detection Algorithm for Custom Object Detection. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/12/how-to-use-yolo-v5-object-detection-algorithm-for-custom-object-detection-an-example-use-case/> .Juin 2023
- [48] Samuel Théophile, Roboflow : conversion d'annotations pour la détection d'objets,3 sept. 202, <https://medium.com/analytics-vidhya> , juin 2023
- [49] Colaboratory, <https://research.google.com/colaboratory/faq.html> , juin 2023
- [50] What is Google Colab ? ,2021,https://blog.education-ecosystem.com/what-is-google-colab, juin 2023
- [51] Rédaction, L. (2020). Python. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/> .Juin 2023
- [52] DeepAI. (2020). PyTorch. DeepAI. <https://deepai.org/machine-learning-glossary-and-terms/pytorch> , juin 2023
- [53] Axopen. 2021.OpenCV, c'est quoi ? Axopen. <https://www.axopen.com/blog/2019/09/open-cv-cest-quoi/> . Juin 2023