جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL
كلية العلوم
FACULTÉ DES SCIENCES

Département d'Informatique

Licence fondamentale :

Sciences Mathématiques et Informatique (SMI)

Projet de Fin d'Etudes

Intitulé :

Date Fruit Detection using Deep Learning Algorithms

Réalisé par :

- Mohamed KABOURI
- Bouchra MILOUDY

Encadré par :

Pr. Brahim AKSASSE

Soutenu le 15 juillet devant le jury :

- **Pr. Mohamed EL ANSARI, Faculté des Sciences- Meknès (Président)**
- **Pr. Badraddine AGHOUTANE, Faculté des Sciences- Meknès (Membre)**
- **Pr. Brahim AKSASSE, Faculté des Sciences- Meknès (Encadrant)**

Année Universitaire : 2023-2024

Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce rapport de fin d'étude intitulé "Date Fruit Detection using Deep Learning Algorithms".

À nos parents

Nous exprimons notre gratitude à nos familles pour leur amour et leur soutien constants. Leur confiance en NOUS a été une source inestimable de motivation.

À notre encadrant

Nous adressons nos sincères remerciements à notre encadrant, Brahim AKSSASE, pour son encadrement exceptionnel et ses conseils avisés tout au long de ce projet. Ses vastes connaissances et sa disponibilité ont été essentielles à notre progression. Nous avons particulièrement apprécié ses suggestions lors de nos réunions, qui ont toujours été très constructives.

À nos professeurs

Nous tenons également à remercier le personnel enseignant de Faculté de sciences 'Moulay Ismail' pour son soutien et ses précieux enseignements. Chaque cours a enrichi notre compréhension et notre compétence au cours de nos études universitaires.

Enfin, nous remercions toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail, y compris celles qui ont participé aux tests et à la validation des algorithmes.

Merci à tous.

Résumé

Dans ce travail, nous proposons un cadre de vision industrielle efficace pour les robots de récolte des dattes. Le cadre se compose de trois modèles de classification utilisés pour classer les images de fruits datés en temps réel en fonction de leur type, de leur maturité et de leur décision de récolte. Dans les modèles de classification, les réseaux neuronaux convolutifs profonds sont utilisés avec l'apprentissage par transfert et le réglage fin sur des modèles pré-entraînés. Pour construire un système de vision robuste, nous créons un riche ensemble de données d'images de grappes de dattes dans un verger qui se compose de plus de 9092 images de cinq types de dattes à différents stades de prématurité et de maturité. L'ensemble de données présente un grand degré de variations qui reflète les défis de l'environnement des vergers de dattiers, notamment les variations d'angles, d'écailles, de conditions d'éclairage et les régimes de dattes recouverts de sacs. Les modèles de classification des dattiers proposés atteignent des précisions de 90,01 %, 90,25 % et 90,59 % avec des temps de classification de 20,6, 20,7 et 35,9 ms pour les tâches de classification du type, de la maturité et de la décision de récolte, respectivement.

Liste des Tableaux :

Liste des figures

Table des matières

Introduction générale

La région de Drâa-Tafilalet est l'une des principales zones de production de dattes au Maroc, couvrant plus de 48 453 hectares, y compris les nouvelles zones d'extension sur l'axe Meski-Boudnib. Cette région produit environ 94 000 tonnes de dattes par an, soit 85% de la production nationale pour la période 2010-2018, et emploie plus de 100 000 producteurs regroupés en 21 groupements d'intérêt économique.

Face aux défis posés par la récolte manuelle, ce projet vise à améliorer les processus de récolte grâce à la détection et à la classification des images. En tant qu'informaticiens, nous avons la capacité de contribuer de manière significative en développant des outils avancés pour le traitement et la manipulation des images. La détection précise et la classification fiable des dattes optimisent la récolte en automatisant l'identification de la maturité et de la variété des fruits, ce qui réduit le temps et les efforts nécessaires pour les trier manuellement.

L'objectif principal de ce projet est de développer une interface graphique intégrant un outil de détection et de classification des dattes en utilisant YOLOv8 (You Only Look Once version 8). YOLOv8 est un modèle de détection d'objets de pointe qui permet d'entraîner des modèles robustes capables de détecter et de classer les fruits de dattes selon leur maturité, en diverses conditions d'éclairage et d'angle.

Ce travail se décompose en plusieurs étapes clés :

Étude Théorique : Compréhension des bases de la détection et la classification d'images et des techniques de vision par ordinateur.

Implémentation avec YOLOv8 : Utilisation de YOLOv8 pour détecter et classifier les dattes. YOLOv8 permet de traiter efficacement les images pour reconnaître les fruits de dattes et déterminer leur stade de maturité.

Analyse et Modélisation : Développement de l'application incluant une interface utilisateur pour le traitement des images et la visualisation des résultats.

Grâce à ce projet, nous espérons automatiser le processus de récolte des dattes, augmentant ainsi la productivité et la précision, tout en réduisant les coûts de main-d'œuvre dans la région de Drâa-Tafilalet.

Figure 1 : Une image des palmiers**.**

Chapitre 1 : Les notions de traitement d'image

I. Introduction

Les humains communiquent par divers moyens, dont la parole et l'image sont d'une importance cruciale. L'image en particulier est un puissant moyen de communication, qui permet à chacun de déchiffrer et d'extraire des informations selon sa propre perspective.

Ce chapitre explore les concepts de base liés au traitement d'image, ainsi que les principales caractéristiques nécessaires à sa détection et à sa classification. L'objectif est de rendre ces processus plus faciles, plus efficaces et plus réalisables, tout en permettant une analyse précise et pratique des données visuelles.

II. Notion d'image

Une image est une représentation visuelle de quelque chose, que ce soit une personne, un objet ou une scène. Cette représentation peut être réalisée par divers moyens tels que la peinture, la sculpture, le dessin, la photographie ou le film.

II.1 Image numérique

Une image numérique est une représentation d'une scène visuelle ou d'un objet stocké sous la forme d'une collection de données binaires. Il est composé de pixels, qui sont les plus petites unités d'une image, chaque pixel ayant une valeur de couleur spécifique. Les images numériques peuvent être créées par des appareils tels que des appareils photo, des scanners ou générées par infographie. Ils sont couramment utilisés dans diverses applications telles que la photographie, l'imagerie médicale et la vision par ordinateur. La qualité et la résolution d'une image numérique dépendent du nombre de pixels et de la profondeur de représentation des couleurs.

II.1.1 Type des images numérique

Les images numériques peuvent être classées en deux classes : les images matricielles et les images vectorielles.

II.1.1.1 Image matricielle

C'est une image sous forme d'une matrice (représentée par une matrice) composées d'une grille de pixels individuels, où chaque pixel a une valeur de couleur spécifique. La qualité et les détails d'une image raster sont déterminés par sa résolution, mesurée en pixels par pouce (PPI) ou en points par pouce (DPI).

II.1.1.1.1 Pixel

Le mot pixel vient de l'expression « picture element ». C'est l'unité indivisible permettant de coder l'information relative à la luminosité en certaine position dans l'image.

Un pixel est décrit par ses coordonnées dans l'image (i,j), et sa valeur I(i,j), représentant son niveau de gris ou sa couleur.

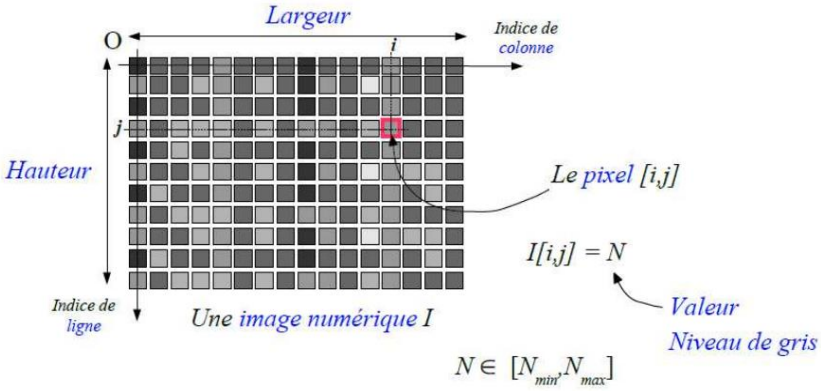

Figure 2 : Image a plusieurs pixels.

II.1.1.1.2 Image a niveau de gris

Une image à niveau de gris est une image matricielle où chaque pixel peut prendre une valeur entière de 0 à N, où N est la valeur maximale de niveau de gris.

Figure 3 : Image a niveau de gris.

II.1.1.1.3 Image couleur

Une image couleur est généralement représenter par trois matrices, chacune correspondant à un canal de couleur : rouge (Red), vert (Green) et bleu (Blue). Ces canaux, lorsqu'ils sont combinés, forment l'image complète.

Figure 4 : Image couleur.

II.1.1.1.4 Image binaire

Une image binaire est une image matricielle (M x N) qui a uniquement deux valeurs possibles pour chaque pixel. Ces valeurs sont 0 et 1 ; 0 représentant le noir et 1 représentant le blanc.

Figure 5 : Image binaire.

II.1.1.2 Image vectorielle

Contrairement aux images matricielles formées de pixels individuels, les images vectorielles se basent sur des équations mathématiques et un réseau de points, de lignes, et de courbes pour représenter une image. Grâce à cette technique, les images vectorielles peuvent être agrandies ou réduites indéfiniment

sans perdre en qualité, car les calculs mathématiques permettent d'ajuster l'image à la taille ou à la résolution désirée.

II.1.2 Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par plusieurs paramètres, on cite quelques-uns :

II.1.2.1 Dimension

Les dimensions d'une image numérique sont déterminées par sa résolution, c'est-à-dire le nombre de pixels en largeur et en hauteur. Par exemple, une photo a une résolution de 1920 * 1080 pixels par 1920 pixels et une largeur de 1080 pixels. La résolution fait référence au niveau de détail qu'une image peut afficher : soit une haute résolution ce qui signifie plus de pixels, et donc plus de détails. Ou basse résolution : signifie une image avec moins de pixels, ce qui signifie moins de détails.

II.1.2.2 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

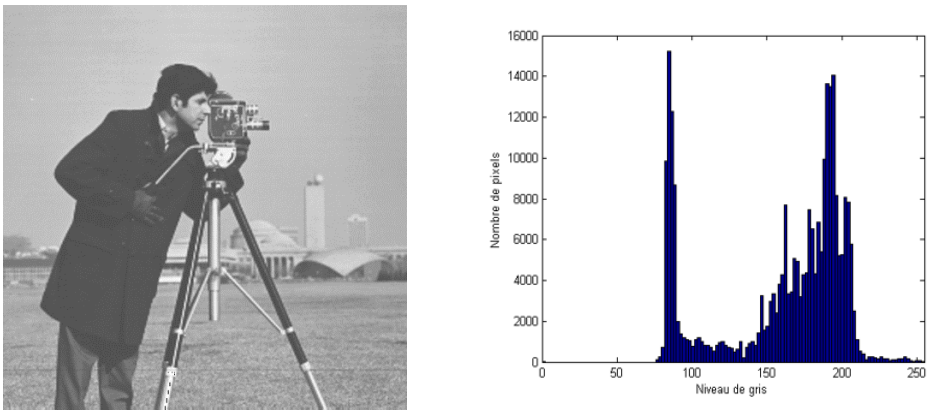

Figure 6 : Image avec son histogramme.

III. La détection et la classification des objets

III.1 La détection des objets

III.1.1 Définition

La détection d'objets est une technologie de pointe en apprentissage automatique et en apprentissage profond qui permet aux ordinateurs d'identifier et de localiser avec précision les objets au sein des

images ou des vidéos. Elle appartient à une branche de l'intelligence artificielle appelée "Computer Vision".

La détection d'objets est une tâche de vision par ordinateur dont l'objectif est de détecter et localiser des objets d'intérêt dans une image ou d'un flux vidéo. La tâche implique d'identifier la position et les limites des objets dans une image, ainsi que de classer les objets dans différentes catégories.

III.1.2 Les grandes étapes de la détection des objets

En effet, le principe de la détection d'objets est le suivant : pour une image donnée, on recherche les régions de celle-ci qui pourraient contenir un objet puis pour chacune de ces régions découvertes, on l'extrait et on la classe à l'aide d'un modèle de classification d'image – par exemple –. Les régions de l'image d'origine ayant de bons résultats de détection sont conservés et les autres jetés. Ainsi, pour avoir une bonne méthode de détection d'objets, il est nécessaire d'avoir un algo solide de détection de régions et un bon algo de détection et classification d'images.

Alors, en se basant sur plusieurs données, soit une variété d'images et de données d'entraînement, l'algorithme IA détectera progressivement les motifs, les textures et les formes communes aux images de test utilisées pour chaque catégorie et apprendra à les reconnaître. Il pourra alors les identifier automatiquement dans n'importe quelle nouvelle image.

Figure 7 : Une image et la détection des dattes.

III.2 La classification des images

III.2.1 Définition

La classification des images est la plus simple des trois tâches et consiste à classer une image entière dans l'une des classes prédéfinies.

Le résultat d'un classificateur d'images est une étiquette de classe unique et un score de confiance. La classification d'images est utile lorsque tu as besoin de savoir uniquement à quelle classe appartient une image et que tu n'as pas besoin de savoir où se trouvent les objets de cette classe ou quelle est leur forme exacte (la localisation des objets est une tâche de la détection des objets).

III.2.2 Les grandes étapes de la classification des images

La classification d'image implique de faire passer une image entière à travers un classificateur, généralement un réseau neuronal profond, pour obtenir une étiquette ou un tag correspondant. Les classificateurs analysent l'image complète mais ne fournissent pas d'informations sur l'emplacement spécifique de l'objet étiqueté à l'intérieur de l'image.

IV. Conclusion

Nous avons introduit les notions de base qui servent de fondement à la compréhension de différents types d'images numériques ainsi que leurs caractéristiques, et aussi une définition générale de la détection et de la classification.

Chapitre 2 : La solution proposée – YOLOv8

I. Introduction

Dans ce chapitre, nous allons explorer les concepts fondamentaux du deep learning, en nous concentrant particulièrement sur les réseaux de neurones convolutifs. Nous aborderons également les bases de la vision par ordinateur et la méthodologie YOLO, avec un accent spécifique sur YOLOv8, utilisée pour la détection et la classification des objets. Enfin, nous examinerons comment organiser les datasets, ce qui est crucial pour la précision et l'efficacité des modèles de détection et de classification.

II. Deep Learning (DL)

II.1 Définition

Le Deep Learning est une sous-branche de l'apprentissage automatique qui utilise des réseaux neuronaux artificiels pour modéliser des données complexes. Inspiré par le fonctionnement du cerveau humain, le Deep Learning utilise des couches de neurones artificiels pour extraire des caractéristiques et des modèles à partir de données brutes.

II.2 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont directement inspirés du cortex visuel des vertébrés. Un réseau de neurones à convolution, appelé aussi convnet, ou encore CNN se représente en général sous la forme suivante :

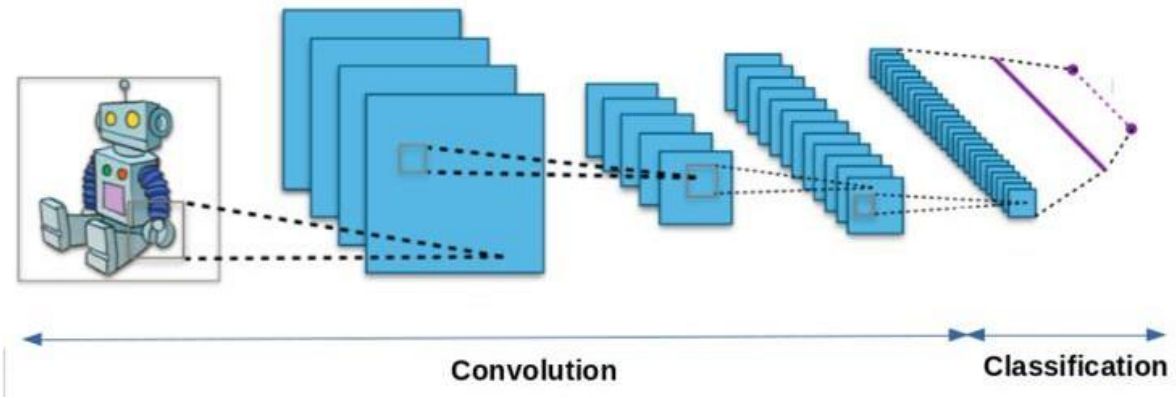

Figure 8 : Réseau de neurones convolutifs.

II.2.1 Architecture d'un réseau de neurones convolutifs

On distingue deux parties, une première partie que l'on appelle la partie convolutée du modèle et la seconde partie, que l'on va appeler la partie classification du modèle qui correspond à un modèle MLP.

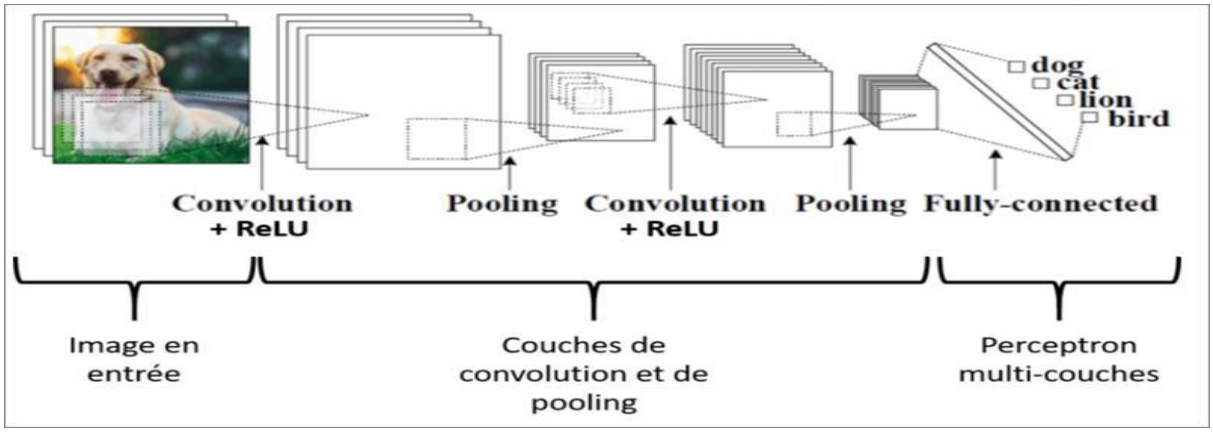

Figure 9 : Architecture d'un CNN.

II.2.2 Caractéristiques et avantages des CNN

Un avantage majeur des CNN est l'utilisation d'un poids unique associé aux signaux entrants dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au perceptron multicouche, qui, lui, considère chaque neurone indépendant et affecte donc un poids différent à chaque signal entrant.

Lorsque le volume d'entrée varie dans le temps, il devient intéressant de rajouter un paramètre le long de l'échelle de temps dans le paramétrage des neurones. On parlera dans ce cas de réseau neuronal à retard temporel.

Comparés à d'autres algorithmes de classification d'images, les CNN utilisent relativement peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres, ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN.

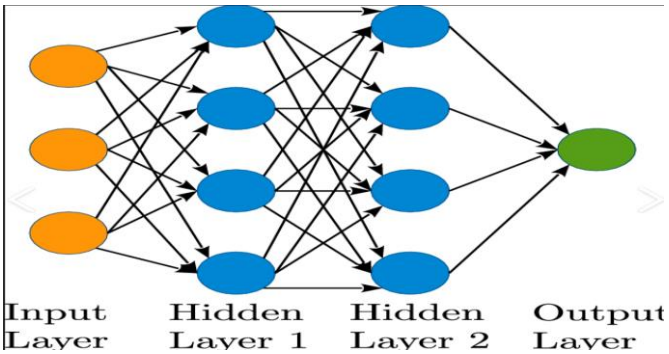

Figure 10 : Split.

II.3 Révolution d'apprentissage profond

II.3.1 L'évolution des CPU et des GPU

L'amélioration des processeurs, notamment les GPU, a été un catalyseur majeur dans l'avancée des réseaux de neurones en fournissant la puissance de calcul nécessaire pour entraîner des modèles profonds et complexes.

II.3.2 Une Activation du GPU

Nous devons activer l'accélération GPU dans notre Google Colab afin que le système YOLO.v3 puisse traiter les détections plus de 100 fois plus vite que le CPU.

- Le GPU utiliser par Cloud : TESLA P4 Pour cela il y a des étapes à faire :
- Cliquons sur Modifier (Edit) en haut à gauche de notre carnet (Notebook).
- Cliquons sur Paramètres (Notebook settings) dans la liste.
- Sélectionnons GPU et appuyez sur "Save".

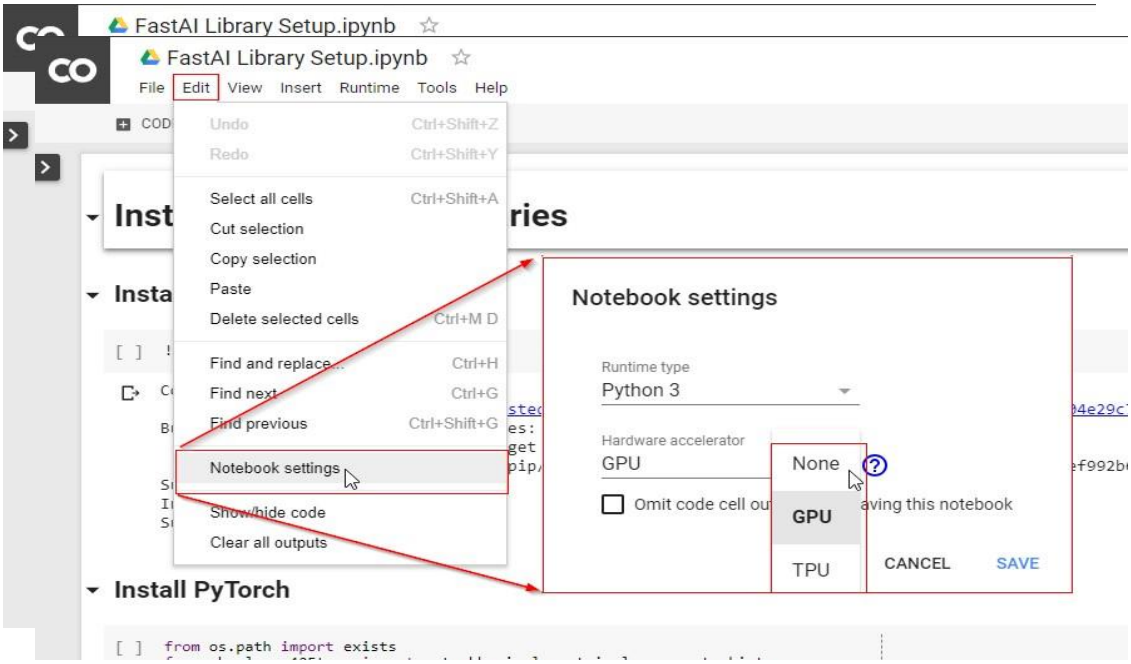

Figure 11 : L'activation du GPU.

III. Environnement de développement

III.1 Environnement matériel (hardware)

Un ordinateur avec les caractéristiques suivantes :

- **Processeur :** Intel i5 8550u Ram

- **RAM :** 8go
- **Disque Dur :** 1000 go HDD Webcam intégrée de marque

III.2 Environnement immatériel (software)

- **Système d'exploitation :** Windows 11 64bit
- **Logiciel :** python 3.7

IV. YOLO (You Only Look Once)

IV.1 Définition (Yolo, c'est quoi?)

YOLO, ou "You Only Look Once", est un outil spécial qui aide les ordinateurs à voir rapidement et avec précision les choses dans les images, les fichiers textes ou les vidéos.

Créé par les experts Joseph Redmon et Ali Farhadi en 2015, YOLO est plus rapide que les outils plus anciens car il analyse l'image entière en une seule fois. Cette vérification rapide permet à YOLO d'identifier rapidement s'il y a d'autres objets, comme des voitures, des arbres ou des animaux, et où ils se trouvent dans l'image.

IV.2 Les versions de YOLO

Les créateurs de YOLO continuent de le mettre à jour l'algorithme pour l'améliorer continuellement ; il existe de nombreuses versions, de YOLOv1 à YOLOv9 (le plus récent, sorti en Février 2024), chaque nouvelle version étant plus rapide et précise. YOLO est devenu très populaire car il donne aux machines des superpouvoirs pour voir et comprendre le monde rapidement et repérer des objets pour une multitude d'applications dans le monde réel.

IV.2.1 YOLO V1

La première version de YOLO a révolutionné la communauté des chercheurs en IA / Computer Vision avec ses capacités de détection d'objets en temps réel, offrant des vitesses d'inférence bien plus rapides que les méthodes existantes telles que R-CNN.

YOLO v1 divise l'image entrante en une grille et prédit plusieurs boîtes englobantes et probabilités de classe pour chaque cellule de la grille.

Cependant, avec cette première version, la précision était un compromis. YOLO avait alors du mal avec les petits objets et produisait de nombreuses erreurs de localisation d'objet.

IV.2.2 YOLO V2 et V3

Les versions suivantes, telles que YOLO v2 et v3, ont introduit des améliorations notables et de nouvelles fonctionnalités comme les boîtes d'ancrage, utilisant le clustering k-means pour prédire des coordonnées de boîtes englobantes plus précises.

Ces versions ont également profité de la normalisation par lots et la gestion d'images d'entrée de plus haute résolution, conduisant à des performances de détection nettement meilleures sur des benchmarks tels que les ensembles de données Pascal VOC et COCO.

IV.2.3 YOLO V4 et V5

Avec pour objectif d'atteindre à la fois une grande vitesse et une haute précision, YOLO v4 a introduit des fonctionnalités telles que le pooling pyramidal spatial et une architecture YOLO plus complexe basée sur des réseaux convolutionnels de pointe.

YOLO v5 s'est quant à lui concentré sur la simplification et l'optimisation, lui permettant de fonctionner extrêmement rapidement sur du matériel moins puissant tout en maintenant une haute précision.

IV.2.4 YOLO V6 à V8

Les versions les plus récentes de YOLO, à partir de la version 6, introduisent des améliorations continues axées sur les applications concrètes de YOLO, telles que les véhicules autonomes ou la surveillance vidéo. Plus on avance dans le temps, et plus YOLO s'éloigne de la communauté des chercheurs pour atteindre le grand public et les cas d'usage appliqués à la vie réelle.

Ces versions ont affiné l'utilisation des techniques de Deep Learning, y compris diverses formes d'augmentation des données et d'algorithmes d'optimisation qui ont aidé à améliorer la précision moyenne et la capacité à détecter une gamme diversifiée de classes d'objets.

IV.2.5 YOLO V9

Le 21 février 2024, Chien-Yao Wang, I-Hau Yeh et Hong-Yuan Mark Liao ont publié l'article "YOLOv9 : Apprendre ce que vous voulez apprendre en utilisant l'Information de Gradient Programmable", qui introduit une nouvelle architecture de modèle de vision par ordinateur YOLOv9.

YOLOv9 représente une avancée majeure dans la série des modèles YOLO, offrant des améliorations significatives en termes de précision et d'efficacité pour la détection d'objets en temps réel. Il se distingue de ses prédécesseurs, notamment YOLOv8, par une réduction de 49% du nombre de

paramètres et de 43% de la complexité computationnelle, tout en augmentant la précision moyenne sur le jeu de données MS COCO de 0,6%.

Tableau 1 : Des différentes versions et évolutions de YOLO.

Version	Amélioration	Compromis Vitesse / Précision	Application
V1	Précision par cellules de grille, méthode à tir unique	Rapide mais moins précis	Détection en temps réel fondamentale (recherche)
V2 et V3	Boîtes d'ancrage, normalisation par lots	Plus rapide et plus précis	Diverses applications en temps réel
V4 et V5	Pooling pyramidal spatial, optimisations	Equilibre entre vitesse et précision	Environnements exigeants, comme le transport
V6 à V8	Optimisations ciblées, architectures améliorées	Très précis et en temps réel	Applications spécialisées, comme la surveillance
V9	Amélioration de la détection des petits objets, intégration avec d'autres modèles d'IA et IA explicable	Précision et vitesse accrues	Applications telles que l'imagerie médicale, la conduite autonome ou la détection de défauts industriels

IV.3 Fonctionnement de YOL

Voici comment l'algorithme de détection d'objets YOLO (You Only Look Once) fonctionne, expliqué en étapes simples :

IV.3.1 Prendre une photo

Tout d'abord, l'algorithme YOLO commence par une image, tout comme lorsque vous prenez une photo avec un appareil photo. C'est ce que nous appelons la détection d'objets basée sur la classification d'images !

IV.3.2 Diviser l'image

Ensuite, il divise l'image donnée en petits carrés, comme un damier. Chaque carré sera vérifié pour voir s'il contient un objet (un chat, un chien ou encore une boîte de conserve, par exemple).

IV.3.3 Chercher des indices

Pour chaque carré, YOLO recherche des indices ou des caractéristiques comme des bords, des formes ou des textures qui pourraient indiquer quel objet s'y trouve. Il les entoure avec des boîtes englobantes. Comme YOLO a besoin d'apprendre pour bien comprendre un nouveau jeu de données et interpréter, on lui a parfois donné un jeu de données de référence (ou "vérité terrain") dans lequel il peut puiser pour avoir des points de comparaison.

IV.3.4 Faire des prédictions

L'algorithme fait une supposition pour chaque carré d'une image : quel objet cela pourrait-il être et où exactement se trouve-t-il dans le carré ? Il attribue à chaque supposition un score pour montrer son niveau de certitude.

IV.3.5 Éliminer les excédents

Certains carrés ont des suppositions de différents objets qui se chevauchent, comme deux carrés devinant une partie de la même voiture. YOLO choisit la meilleure supposition pour chaque objet, en se débarrassant des suppositions superflues.

IV.3.6 Montrer ce qu'il a trouvé

À la fin, YOLO vous montre où il pense que chaque objet se trouve en dessinant des boîtes autour d'eux et en les étiquetant, comme "voiture" ou "arbre". Si vous lui donnez 1'000 images contenant des chiens et des chats, et lui dites d'identifier les chats, il vous montrera des images enrichies de métadonnées pointant vers les chats.

IV.4 Evolution de la détection d’objets : de YOLO 1 à YOLO 9

YOLO, acronyme de "You Only Look Once", est un algorithme de détection d'objets en temps réel qui a connu d'importantes améliorations depuis sa création. En tant que détecteur "one shot", il traite les images et identifie les objets en prédisant des boîtes englobantes et des probabilités de classe en une seule passe. Au fil du temps, YOLO est devenu de plus en plus résilient et performant, comme l'illustre très bien la dernière publication de ses auteurs :

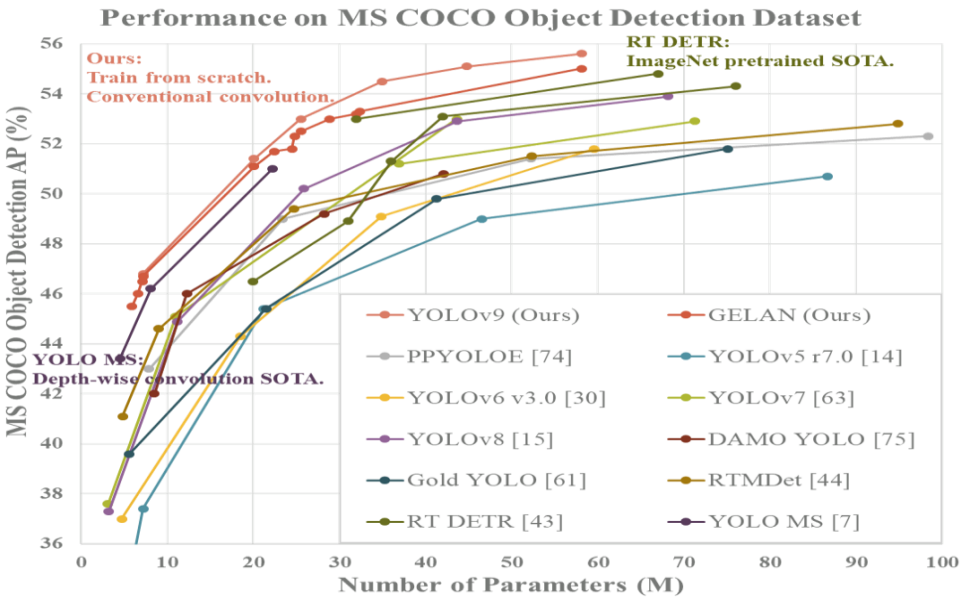

Figure 12 : Illustration de la performance de YOLO.

IV.5 La version utilisée – YOLOv8

Dans notre projet, nous avons utilisé YOLOv8. C'est la dernière version du célèbre modèle de détection d'objets et de classification d'images en temps réel. S'appuie sur des avancées de pointe en matière d'apprentissage profond et de vision par ordinateur, offrant des performances inégalées en termes de rapidité et de précision. YOLOv8 peut réaliser la détection et la localisation d'objets en une seule étape.

IV.5.1 YOLOv8 pour la Détection d'Objets

Identifier et localiser plusieurs objets dans une image. Pour chaque objet détecté, le modèle prédit une boîte englobante et une classe.

IV.5.1.1 L'organisation de dataset et le format YOLO

format Ultralytics YOLO est un format de configuration du jeu de données qui te permet de définir le répertoire racine du jeu de données, les chemins relatifs vers les répertoires d'images d'entraînement/de validation/de test ou les chemins relatifs vers les répertoires d'images d'entraînement/de validation/de test. *.txt des fichiers contenant les chemins d'accès aux images, et un dictionnaire de noms de classes. Voici un exemple :

```yaml
path: '/content/drive/MyDrive/PFE/Dataset_Bouisthami'
train: ../train/images
val: ../val/images
test: ../test/images

nc: 1
names: ['Date_fruit']
```

Figure 13 : L'organisation de dataset dans le fichier data.yaml.

Les étiquettes de ce format doivent être exportées au format YOLO avec une *.txt par image. S'il n'y a pas d'objets dans une image, aucune *.txt est nécessaire. Le *.txt Le fichier doit être formaté avec une ligne par objet en class x_center y_center width height format. Les coordonnées de la boîte doivent être en xywh normalisé (de 0 à 1). Si tes cases sont en pixels, tu dois diviser x_center et width par la largeur de l'image, et y_center et height par la hauteur de l'image. Les numéros de classe doivent être indexés à zéro (commencer par 0).

Figure 14 : Une image annoter avec Roboflow.

Le fichier d'étiquettes correspondant à l'image ci-dessus contient 2 personnes (classe 0) et une égalité (classe 27) :

zidane.txt

```
0  0.481719 0.634028 0.690625 0.713278
0  0.741094 0.524306 0.314750 0.933389
27 0.364844 0.795833 0.078125 0.400000
```

Figure 15 : Annotation d'image précédente.

Chaque image de votre ensemble de données doit avoir un fichier .txt correspondant avec tous les objets de l'image avec un [id_class x0 y0 x1 y1] normalisé.

IV.5.1.2 Comment entraîner YOLOv8 sur votre ensemble de données personnalisé

- **Importer et configurer YOLOv8**

La première étape consiste à importer la bibliothèque nécessaire et à charger un modèle YOLOv8 pré-entraîné. Nous pouvons choisir parmi plusieurs versions du modèle (par exemple, yolov8n.pt, yolov8s.pt, yolov8m.pt, etc.), où n, s, m, etc., désignent respectivement les versions nano, small, medium, etc.

Tableau 2 : Les versions du modèle YOLOv8 pour la détection.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Voici comment on peut importer le modele YOLOv8 :

```
from ultralytics import YOLO
model = YOLO("yolov8n.pt")
```

Figure 16 : Code d'importation du modèle yolov8.

- **Configurer l'entraînement**

Ensuite, il faut configurer les paramètres d'entraînement. Cela inclut des options telles que le chemin vers le fichier data.yaml, le nombre d'époques, la taille du batch, la taille des images, etc.

Voici un script Python détaillé pour configurer et démarrer l'entraînement :

```
result = model.train(data=os.path.join(ROOT_DIR, "data_2.yaml"), epochs=30)
```

Figure 17 : Code d'entrainement du modèle yolov8.

Après L'entrainement, un dossier runs créé par défaut par les scripts d'entraînement YOLO pour stocker les résultats de l'entraînement, y compris les modèles sauvegardés, les logs, et les visualisations. Ce

dossier contient un autre dossier nommé « weights », Ce sous-dossier contient les poids du modèle sauvegardés au cours de l'entraînement. Typiquement, il contient deux fichiers principaux :

- **best.pt** : Les poids du modèle correspondant à la meilleure performance sur l'ensemble de validation.
- **last.pt** : Les poids du modèle à la fin de la dernière époque d'entraînement.

On peut utiliser ce fichier comme un modèle pour continue l'entrainement.

IV.5.2 YOLOv8 pour la Classification d'Images

Classification d'Images est Assigner une classe unique à une image entière.

IV.5.2.1 L'organisation de dataset

Pour Ultralytics YOLO pour les tâches de classification, l'ensemble de données doit être organisé dans une structure spécifique de split-direcotry sous la forme d'un tableau. Afin de faciliter la formation, les tests et les processus de validation facultatifs. Cette structure comprend des répertoires distincts pour la formation (train) et les tests (test), avec un répertoire optionnel pour la validation (val).

Chacun de ces répertoires doit contenir un sous-répertoire pour chaque classe de l'ensemble de données. Les sous-répertoires portent le nom de la classe correspondante et contiennent toutes les images de cette classe. Veille à ce que chaque fichier image porte un nom unique et soit stocké dans un format commun tel que JPEG ou PNG.

- **Répertoire d'Entraînement (train/)** : Ce dossier contient les images utilisées pour former le modèle.
- **Répertoire de Validation (val/)** : Ce dossier contient les images utilisées pour valider le modèle pendant la formation. La validation permet d'ajuster les hyperparamètres et d'évaluer la performance du modèle sur des données inédites, aidant ainsi à prévenir le surapprentissage.
- **Répertoire de Test (test/)** : Ce dossier contient les images utilisées pour évaluer la performance finale du modèle après la formation. Les données de test sont complètement séparées de celles d'entraînement et de validation pour fournir une évaluation objective de la performance du modèle.

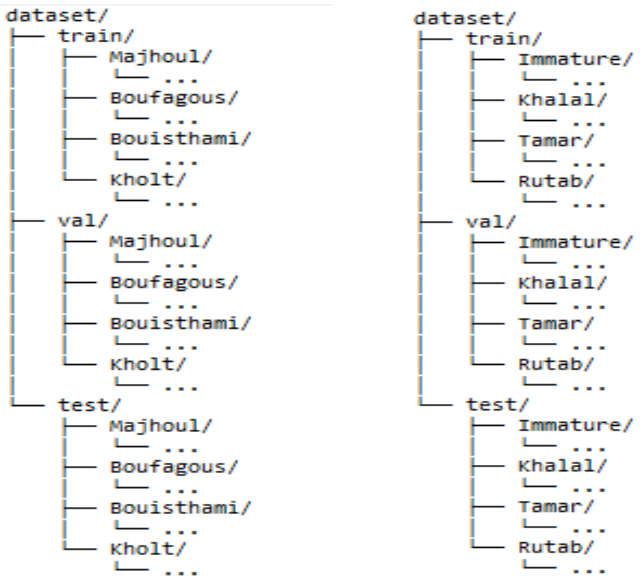

Figure 18 : L'organisation de dataset pour la classification selon le type et selon la maturité.

YOLOv8 offre une gamme de modèles pour la classification d'images, chacun conçu pour répondre à des besoins spécifiques en termes de ressources de calcul et de performance. Le tableau suivant présente quelque version :

Tableau 3 : Les versions du modèle YOLOv8 pour la classification.

Modèle	taille (pixels)	acc top1	acc top5	Vitesse CPU ONNX (ms)	Vitesse A100 TensorRT (ms)	params (M)	FLOPs (B) à 640
YOLOv8n-cls	224	69.0	88.3	12.9	0.31	2.7	4.3
YOLOv8s-cls	224	73.8	91.7	23.4	0.35	6.4	13.5
YOLOv8m-cls	224	76.8	93.5	85.4	0.62	17.0	42.7
YOLOv8l-cls	224	76.8	93.5	163.0	0.87	37.5	99.7
YOLOv8x-cls	224	79.0	94.6	232.0	1.01	57.4	154.8

V. Conclusion

Dans ce chapitre, nous avons défini les concepts clés du deep learning, en nous focalisant sur les réseaux de neurones convolutifs. Nous avons également exploré la méthode YOLO, en particulier YOLOv8, pour la détection et la classification des objets. De plus, nous avons discuté de l'importance de l'organisation des datasets pour ces tâches. Dans le prochain chapitre, nous analyserons les résultats obtenus grâce à l'application de ces techniques sur notre projet de détection et de classification.

Chapitre 3 : Description de dataset

I. Introduction

Dans ce chapitre, nous fournirons une description détaillée du dataset utilisé dans notre projet, y compris son origine, le nombre d'images par classe de classification, et les diverses catégories basées sur la variété et la maturité des dattes.

La détection automatique des fruits de dattes représente une avancée majeure dans l'optimisation des processus de récolte grâce à la vision par ordinateur et à l'apprentissage automatique. Le jeu de données intitulé "Date Fruit Detection Dataset for Automatic Harvesting" offre une base de données essentielle pour l'entraînement et l'évaluation des systèmes de détection des fruits de dattes, permettant d'automatiser la récolte, de réduire les coûts de main-d'œuvre et d'augmenter la productivité.

II. La description des données

Le jeu de données contient 9092 images de grappes de dattes capturées à diverses échelles et angles entre juin et septembre 2022. Ces images ont été prises dans deux vergers au Maroc avec un appareil photo Canon et un smartphone Samsung Ultra. Les images sont annotées pour inclure les étapes de maturité des dattes (immature, Khalal, Rutab, et Tamar) et varient selon les conditions d'éclairage, la couverture par des sacs, et l'angle de vue.

II.1 Conception Expérimentale et Méthodes

Les données ont été collectées dans deux vergers situés au Maroc, utilisant une caméra Canon et un smartphone Samsung Ultra. Les images sont annotées pour détecter et localiser les fruits de dattes en utilisant le logiciel Roboflow, avec des fichiers de sortie au format YOLO pour la détection d'objets.

II.1.1 Caractéristiques principales

- **Types de données** : Images (JPG), vidéos (MP4), fichiers texte (TXT).
- **Variétés de dattes** : Majhoul, Boufaguos, Kholt, et Bouisthami.
- **Étapes de maturité** : Immature, Khalal, Rutab, Tamar.
- **Annotations** : Détection des fruits, classification des variétés, et classification des étapes de maturité.

- Le jeu de données est publié gratuitement sur ZENODO et vise à améliorer la recherche en agriculture robotisée en fournissant des données variées et annotées nécessaires pour développer des systèmes de vision machine robustes pour la récolte automatique des dattes.

II.1.2 Valeur des Données

- **Modèles de détection d'objets :** Entraîner des modèles pour reconnaître et localiser les fruits de dattes dans un verger, permettant aux robots de naviguer et d'identifier les fruits prêts à être récoltés
- **Robotique en agriculture** : Aider à la classification des fruits de dattes à différents stades de maturité, pour des tâches telles que la détection, la segmentation, la récolte automatisée, l'analyse de maturité, le contrôle de qualité, et l'estimation du rendement.
- **Estimation du rendement** : Utiliser des algorithmes d'apprentissage automatique pour analyser le jeu de données et créer des modèles capables d'estimer avec précision le rendement de la récolte à venir, aidant ainsi les agriculteurs à prendre des décisions éclairées.

II.1.3 Description Détaillée des Données

Le jeu de données couvre toutes les étapes de la récolte, y compris la détection des dattes, la classification, l'analyse de maturité, et les décisions de récolte. Il comprend des images prises sous différents angles et conditions, telles que l'éclairage naturel, les dattes couvertes par des sacs, et les feuilles de palmier. Les stades de maturité des dattes sont classifiés en quatre catégories : Immature, Khalal, Rutab, et Tamar.

- **Étape Immature :** Connue sous le nom de Kimri, caractérisée par des dattes vertes, dures, et en croissance rapide.
- **Étape Khalal :** Les dattes changent de couleur, atteignent leur taille et poids maximums, prêtes pour une première récolte.
- **Étape Rutab :** Les dattes deviennent brunes ou noires, deviennent molles, idéales pour le stockage à froid.
- **Étape Tamar :** Dernière phase, avec des dattes brunes ou noires, texture douce et légèrement collante, conservation longue durée.

- **Statistiques Générales**

- **Total des images : 9092**
- **Total des grappes de dattes : 128**
- **Total des palmiers : 28**

Tableau 4 : Répartition par Stade de Maturité.

Maturité	Boufagous	Majhoul	Bouisthami	Kholt
Immature	913	304	95	140
Khalal	563	573	137	414
Rutab	1675	1345	46	649
Tamar	1174	423	82	559

Tableau 5 : Utilisation du Pourcentage de Maturité pour la Récolte.

Variété	Immature	Khalal	Rutab	Tamar
Boufagous	Non récoltée	Non récoltée	Récoltée (>30%)	Récoltée
Majhoul	Non récoltée	Non récoltée	Récoltée (>50%)	Récoltée
Bouisthami	Non récoltée	Non récoltée	Récoltée (>70%)	Récoltée
Kholt	Non récoltée	Non récoltée	Récoltée (>60%)	Récoltée

Tableau 6 : Nombre d'Images par Variété de Dattes.

Variété de dattes	Nombre d'images	Nombre de grappes	Nombre de palmiers
Boufagous	4325	50	10
Majhoul	2645	58	14
Bouisthami	360	14	2
Kholt	1762	6	2

II.2 Limitations

Le jeu de données inclut quatre variétés de dattes, mais ne couvre pas l'ensemble des variétés existantes, ce qui peut limiter l'adaptabilité du modèle à d'autres types de dattes. De plus, il ne prend pas en compte les variations saisonnières dans l'apparence des dattes.

II.2.1 Déclaration d'Éthique

La production des données n'a impliqué aucun sujet humain, expérimentation animale, ni données provenant de plateformes de médias sociaux. Les auteurs ont respecté les exigences éthiques pour la publication dans Data in Brief.

II.2.2 Disponibilité des Données

Le jeu de données est disponible sur ZENODO sous le lien suivant : ZENODO.

III. Conclusion

Dans ce chapitre, nous avons décrit en détail le dataset utilisé pour la détection et la classification des dattes, en précisant son origine, le nombre d'images par classe, et les critères de classification. Dans le chapitre suivant, nous analyserons les résultats obtenus.

Chapitre 4 : L'interface web

I. Introduction

Dans ce chapitre, nous présenterons l'interface web développée pour notre projet. Nous illustrerons également des exemples des résultats obtenus avec notre système de détection et de classification des dattes, en soulignant la précision et l'efficacité de l'application.

II. Les résultats obtenus avec YOLOv8

II.1 La courbe précision-rappel

- La courbe précision-rappel est un graphique utilisé pour évaluer la performance d'un modèle de classification. Elle trace la relation entre la précision (precision) et le rappel (recall) pour différents seuils de classification.

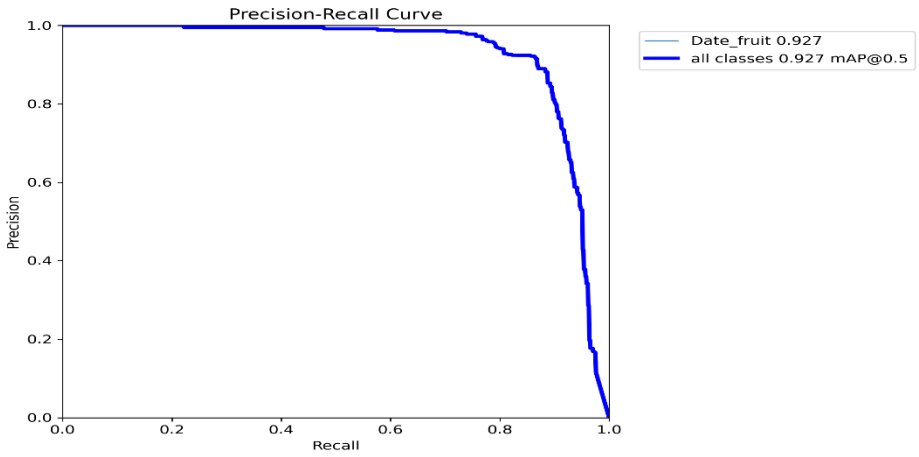

Figure 19 : La courbe précision-rappel.

- Axes du graphique :
  - Axe des abscisses (X) : Rappel (Recall)

Le rappel est la capacité du modèle à identifier toutes les instances pertinentes dans un jeu de données. Il est calculé comme le nombre de vrais positifs divisé par la somme des vrais positifs et des faux négatifs.

  - Axe des ordonnées (Y) : Précision (Precision)

La précision est la capacité du modèle à ne pas étiqueter comme positif un échantillon qui est réellement négatif. Elle est calculée comme le nombre de vrais positifs divisé par la somme des vrais positifs et des faux positifs.

II.2 La matrice de confusion normalisée

- La matrice de confusion normalisée évalue la performance d'un modèle de classification en montrant la répartition des instances des classes réelles dans les classes prédites.

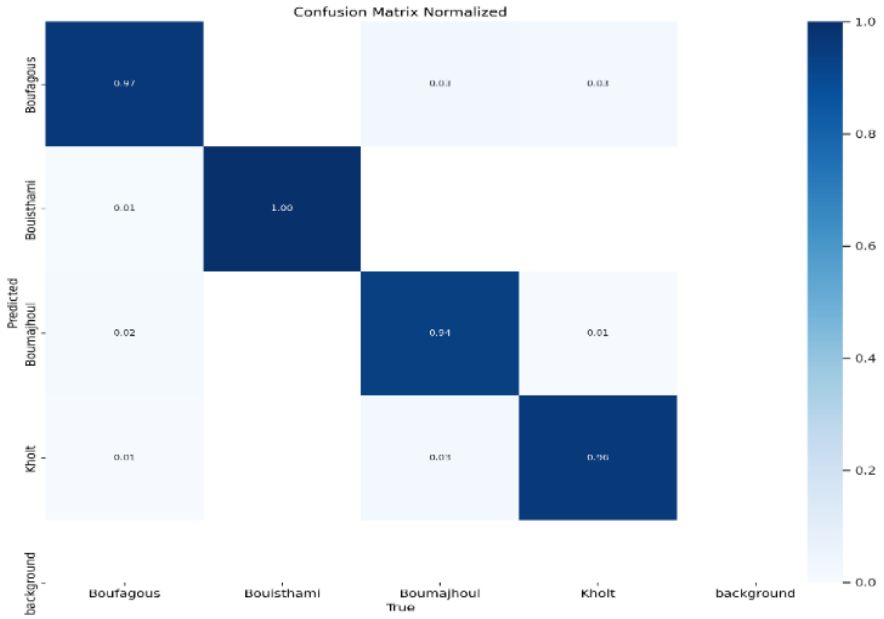

Figure 20 : La matrice de confusion normalisée.

- Axes :
  - Horizontal (True) : Classes réelles
  - Vertical (Predicted) : Classes prédites
- Interprétation des valeurs :

Les valeurs sont normalisées et vont de 0 à 1 (0 à 100%).

- Analyse des résultats :
  - Boufagous (Vraie classe) :
    - 97% correctement prédites comme "Boufagous".
    - 3% incorrectement prédites comme "Boumajhoul" et "background".
  - Bouishtami (Vraie classe) :
    - 100% correctement prédites comme "Bouishtami".
  - Boumajhoul (Vraie classe) :
    - 94% correctement prédites comme "Boumajhoul".
    - 4% incorrectement prédites comme "Boufagous".
    - 1% incorrectement prédites comme "Kholt".
  - Kholt (Vraie classe) :
    - 96% correctement prédites comme "Kholt".
    - 3% incorrectement prédites comme "Boumajhoul".

- 1% incorrectement prédites comme "background".

- Couleurs :
  - Bleu clair : Valeurs faibles
  - Bleu foncé : Valeurs élevées

II.3 Des graphiques de performance du modèle

- Ces graphiques présentent les performances du modèle de classification au cours de l'entraînement, en termes de perte (loss) et d'exactitude (accuracy) pour l'ensemble d'entraînement et de validation.

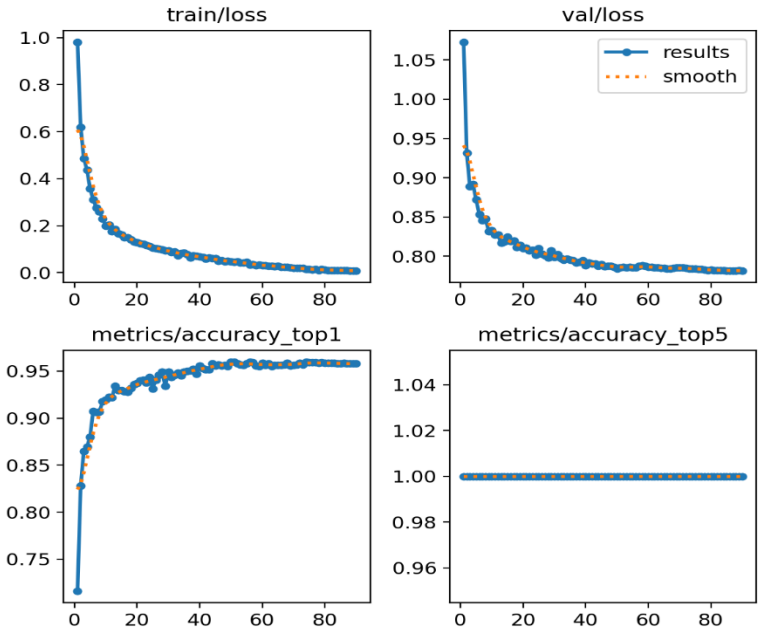

Figure 21 : Les performances du modèle de classification au cours de l'entraînement.

III. La conception et l'interface graphique

III.1 La conception :

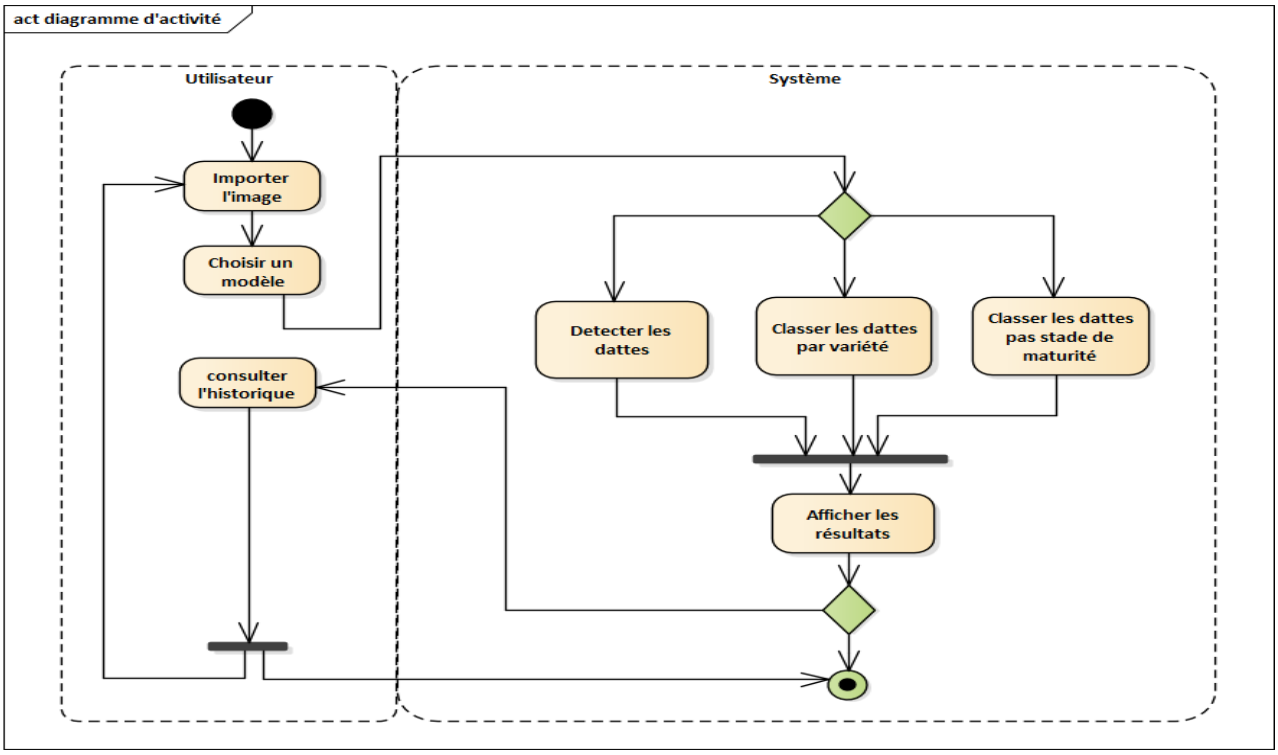

Figure 22 : Diagramme d'activités.

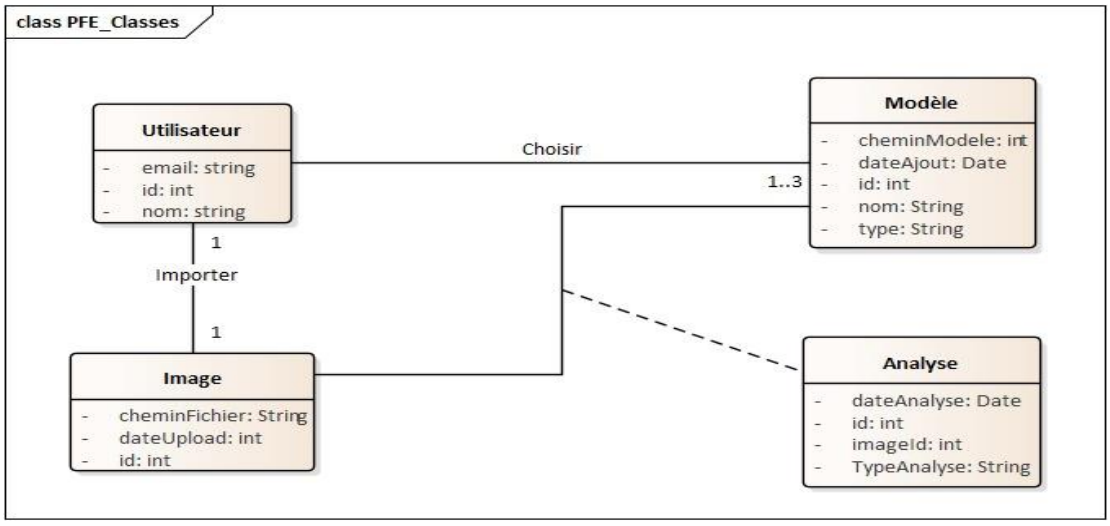

Figure 23 : Diagramme de classes de l'interface web.

III.2 L'interface graphique :

Figure 24 : Page d'accueil.

➢ **Options d'analyse :**

Trois boutons principaux sont disponibles pour l'utilisateur :

- **Détection** : Cette option permet d'identifier et de détecter les dattes présentes dans une image.
- **Classification par variété** : Cette option classe les dattes en fonction de leur variété, en utilisant des caractéristiques visuelles spécifiques.
- **Classification par maturité** : Cette option classe les dattes en fonction de leur stade de maturité, déterminé par leur apparence.

➢ **Utilisation des boutons :**

- Cliquer sur "Détection" : L'utilisateur peut télécharger ou fournir une image de dattes, et le système identifiera les dattes dans l'image.
- Cliquer sur "Classification par variété" : L'utilisateur peut télécharger ou fournir une image de dattes, et le système classera les dattes selon leur variété.
- Cliquer sur "Classification par maturité" : L'utilisateur peut télécharger ou fournir une image de dattes, et le système déterminera et classera les dattes selon leur maturité.

Un clic sur l'un des boutons affiche une deuxième fenêtre pour sélectionner une image.

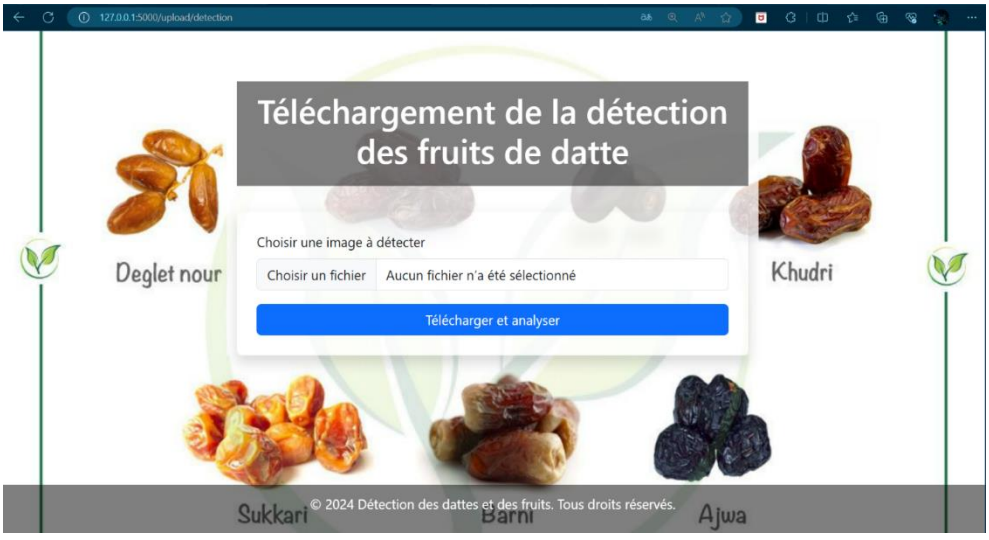

Figure 25 : L'importation des images.

Choisir une image à détecter :

- Cliquer sur le bouton "Choisir un fichier".
- Une fenêtre de sélection de fichier s'ouvre, permettant à l'utilisateur de sélectionner une image à partir de son ordinateur.

Télécharger et analyser :

- Une fois l'image sélectionnée, cliquer sur le bouton "Télécharger et analyser".
- Le système téléchargera l'image et procédera à l'analyse pour détecter les fruits de dattes présents dans l'image.

Un clic sur le bouton « Télécharger et analyser » affiche la fenêtre des résultats.

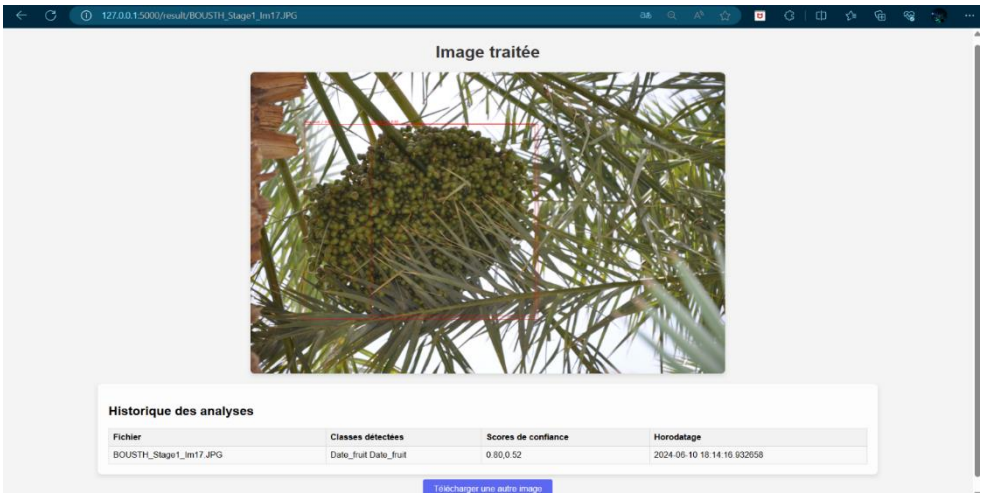

Figure 26 : Résultat de la détection.

✓ **Affichage de l'image traitée**

La partie supérieure de la page montre l'image de dattes traitée. Dans cette image :

- Les dattes détectées sont encadrées : Des boîtes de délimitation (bounding boxes) entourent les dattes détectées dans l'image.
- Les classes détectées sont étiquetées : Les étiquettes indiquent les catégories des dattes détectées, par exemple "Date_fruit".

✓ **Historique des analyses**

Sous l'image traitée, un tableau présent l'historique des analyses effectuées. Ce tableau contient les colonnes suivantes :

- Fichier : Le nom du fichier image téléchargé.
- Classes détectées : Les catégories de dattes détectées dans l'image.

- Scores de confiance : Les scores de confiance associés à chaque détection, indiquant la fiabilité de la détection.
- Horodatage : La date et l'heure à laquelle l'analyse a été effectuée.

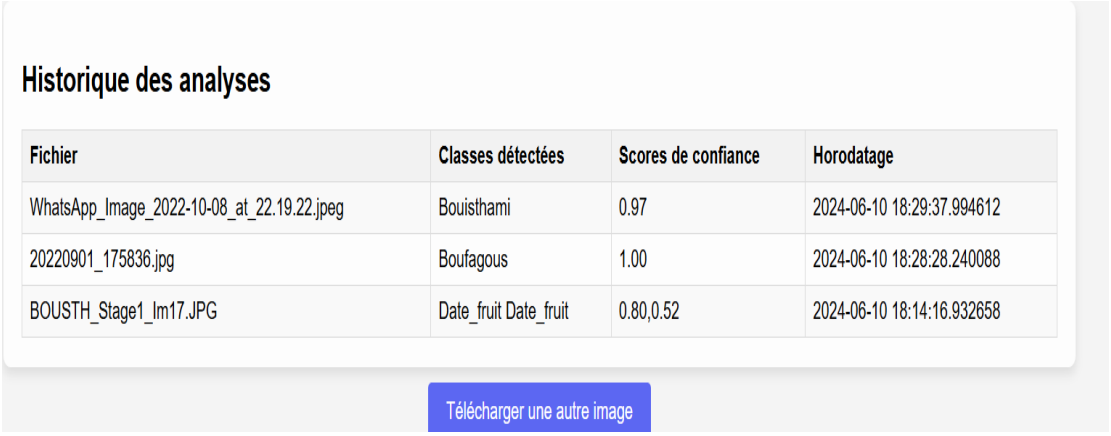

Figure 27: Page d'historique.

✓ **Utilisation de la page**

- Visualiser les résultats : L'utilisateur peut examiner les résultats de l'analyse en regardant l'image traitée et les boîtes de délimitation autour des dattes détectées.
- Consulter l'historique : L'utilisateur peut vérifier les détails des analyses précédentes dans le tableau d'historique.
- Télécharger une nouvelle image : Si l'utilisateur souhaite analyser une autre image, il peut cliquer sur le bouton "Télécharger une autre image" pour retourner à l'écran de téléchargement.

Figure 28 : Résultats de classification par variété.

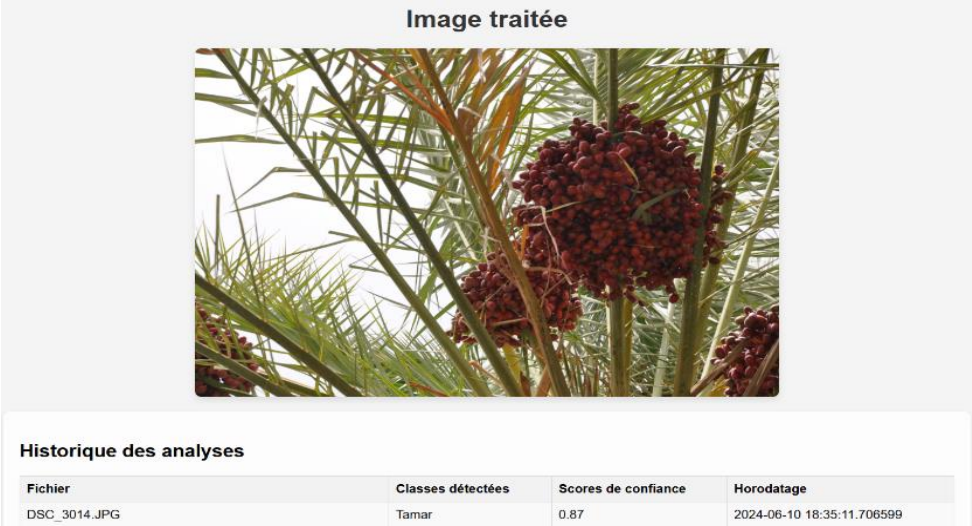

Figure 29 : Résultat de la classification par maturité.

IV. Conclusion

Dans ce chapitre, nous avons examiné l'interface web conçue pour notre projet et montré des exemples concrets des résultats obtenus. Nous avons démontré l'efficacité pratique de notre système de détection et de classification des dattes.

Conclusion générale

Au terme de notre cursus en licence, nous avons été chargés de réaliser un projet de fin d'études. Ce projet était axé sur la détection et la classification des objets, appliqué aux palmiers dattiers, ce qui nous a permis de découvrir une nouvelle discipline (le traitement d'image) et d'enrichir notre savoir ainsi que notre expérience.

Ce projet s'inscrit dans le cadre d'une licence en informatique et mathématiques à l'université MOULAY ISMAIL, faculté des sciences de Meknès.

Dans ce manuscrit, nous avons abordé la problématique de la détection d'objets et de la classification des dattes selon leur maturité et leur variété. Nous avons exploré diverses techniques de traitement d'image numérique, vectorielle et matricielle, afin de comprendre les défis associés à la gestion du bruit et à l'utilisation des histogrammes pour améliorer les images.

Nous avons étudié l'apprentissage profond (deep learning), en particulier dans le contexte d'architecture de réseau neuronal convolutif (CNN). Le modèle YOLO (You Only Look Once), notamment la version Yolov8, a été implémentée pour la détection et la classification des objets.

YOLOv8 s'est révélé être un outil puissant pour la détection rapide et précise des dattes, grâce à son architecture de réseau neuronal convolutif (CNN) optimisée. Cette approche a permis de surmonter les défis liés à la variation de l'éclairage et aux différents angles de prise de vue des images.

Notre interface web développée permet de charger des images de dattes, de les traiter en temps réel et de fournir des résultats sur la maturité et la variété des dattes détectées. Cette interface se veut intuitive et accessible, facilitant ainsi son utilisation par des non-spécialistes.

Comme nous avons pu le constater à travers cette étude, l'application des différentes méthodes de détection et de classification a abouti à des résultats significatifs. Cependant, ces techniques nécessitent souvent des ajustements de paramètres spécifiques et une certaine connaissance préalable des données pour des résultats optimaux.

Pour conclure, notre travail peut être sujet à des extensions et des modifications. En effet, nous envisageons d'ajouter davantage de fonctionnalités, comme une base de données pour enregistrer les résultats et les utiliser dans des processus de reconnaissance plus avancés, ainsi que l'intégration d'autres algorithmes de détection pour enrichir notre application et démontrer encore plus l'utilité de ce domaine fascinant.

Bibliographie

[1] **Nanobaly Camille Jo, Introduction à la détection d'objets en Computer Vision [2024]**: www.innovatiana.com

[2] Hamdi Altaheri, Mansour Alsulaiman, Ghulam Muhammad, Date Fruit Classification for Robotic Harvesting in a Natural Environment Using Deep Learning: https://ieeexplore.ieee.org/document/8807111

[3] Titrite HMOURI, Abdelmajid EL BAHRI. Licence Sciences et Techniques en Informatique : La segmentation d'Image : Application au Palmier Dattier 2019.

[4] Tochukwu Nwoke, Comment-entrainer-yolov8-dataset-personnalise#4-Le-format-YOLO : www.picsellia.fr

[5] Glenn Jocher, fatih c. akyon, Laughing, Burhan : Ultralytics : docs.ultralytics.com

[6] Wikipédia, www.wikipédia.org

[7] M. ISMAILI ALAOUI El Mehdi, Cours de Master IAAD, Université MOULAY ISMAIL, Apprentissage profond – Réseaux de neurones convolutifs (CNN)

[8] Raphael Kassel, You Only Look Once (YOLO), Qu'est-ce que c'est ? : www.datascientest.com

[9] H. D. Cheng, X.H. Jiang, Y. Sun & J. Wang, Color image segmentation : advances and prospects, Pattern Recognition, vol. 34, no. 9, 2001, pp. 22592281. http://www.mathworks.com/products/fuzzylogic/demos.html?file=/products/demos/shipping/fuzzy/fcmdemo_codepad.html

[10] - Jean-Marie Beaulieu, Reconnaissance des formes : Classification et regroupement, département d'informatique, Université Laval, 2002. http://www.ift.ulaval.ca/Info_Courante/Vitrine/Syllabus_IFT64321.pdf