UNIVERSITÉ DE SOUSSE
ÉCOLE NATIONALE D'INGÉNIEURS DE SOUSSE

Département informatique industriel
Rapport de projet Semestriel (S4)

Estimation of olive fruit production of a single olive tree Using object recognition

Réalisé par :

Ouni Baha eddine

Werhani
Mohamed Aziz

Classe :

IA 2.2

Encadrée par :

M. Imed Bennour

Année universitaire : 2023/2024

Sommaire

Introduction Générale

La reconnaissance d'objets est une composante essentielle de l'intelligence artificielle, permettant aux machines de détecter, d'identifier et de comprendre les objets présents dans des données visuelles telles que des images ou des vidéos. Dans le cadre de ce projet, nous nous concentrons spécifiquement sur l'estimation de la production de fruits d'un olivier à l'aide de la reconnaissance d'objets. Cette tâche présente des défis uniques, notamment en raison de la nature variée des olives et des conditions environnementales changeantes. Pour relever ce défi, nous proposons de développer une application Python qui utilise des techniques de reconnaissance d'objets pour estimer le nombre d'olives dans un olivier.

Ce projet vise à explorer les principes fondamentaux de la reconnaissance d'objets, en mettant en évidence son importance croissante dans divers domaines, de l'agriculture à la surveillance industrielle. Nous aborderons également les différentes méthodes et technologies disponibles pour la détection d'objets, en mettant en évidence les avantages et les limitations de chacune. Ensuite, nous nous concentrerons sur l'architecture YOLOv8 (You Only Look Once version 8), en expliquant son fonctionnement et en soulignant pourquoi nous avons choisi de travailler avec ce modèle spécifique.

Nous décrirons également les étapes de gestion des données nécessaires à la création et à l'entraînement du modèle de reconnaissance d'objets, y compris la collecte des données, l'annotation et le prétraitement. Enfin, nous présenterons nos résultats et conclusions, en mettant en évidence les défis rencontrés, les solutions proposées et les perspectives d'avenir pour ce projet.

Chapitre 1

Principes Fondamentaux de Reconnaissance d'Objects

1.1 Introduction

La reconnaissance d'objets est une discipline clé de l'intelligence artificielle qui vise à permettre aux machines de détecter, d'identifier et de comprendre les objets présents dans des données visuelles telles que des images ou des vidéos. Ce chapitre explore les principes fondamentaux de la reconnaissance d'objets, en mettant en lumière son importance croissante dans un large éventail d'applications, de l'automatisation industrielle à la vision par ordinateur en passant par la conduite autonome.

1.2 Fondements théoriques

1.2.1 Origines de la Reconnaissance d'Objects

La reconnaissance d'objets trouve ses racines dans les premières recherches en vision par ordinateur et en traitement d'images. Les premiers travaux remontent aux années 1960, lorsque des chercheurs ont commencé à explorer des méthodes pour extraire des informations significatives à partir d'images, notamment la détection et la classification d'objets.

1.2.2 Principes Fondamentaux

La détection, la classification et la segmentation d'objets sont des concepts fondamentaux de la reconnaissance d'objets. Chacun de ces domaines vise à comprendre et à extraire des informations précieuses à partir d'images, mais avec des objectifs et des approches différents.

Détection d'Objets

La détection d'objets consiste à localiser et à identifier la présence d'objets spécifiques dans une image. Contrairement à la classification qui attribue une étiquette à une image entière, la détection d'objets identifie les emplacements précis des objets dans une image et les entoure généralement d'une boîte englobante. Cette approche est essentielle dans de nombreuses applications telles que la surveillance vidéo, la conduite autonome et la recherche d'objets dans des images médicales.

Classification d'Objets

La classification d'objets implique l'attribution d'une étiquette ou d'une classe à une image ou à une région de l'image. L'objectif est de déterminer la catégorie à laquelle appartient l'objet représenté dans l'image. Par exemple, dans le contexte de la reconnaissance d'objets, une image contenant un chat pourrait être classée comme appartenant à la classe "chat". Cette tâche est fondamentale pour de nombreuses applications, y compris la recherche d'images, la détection de spam et la vision par ordinateur en général.

1.2.3 Les outils de reconnaissance d'objects

Les outils de reconnaissance d'objets présentés offrent une gamme variée de fonctionnalités et de performances pour répondre aux besoins spécifiques des applications de vision par ordinateur. Le SSD (Single Shot MultiBox Detector) et le Faster R-CNN (Region-based Convolutional Neural Network) sont deux approches largement utilisées pour la détection d'objets. Le SSD se distingue par sa rapidité et sa capacité à détecter plusieurs objets dans une seule passe, tandis que le Faster R-CNN est reconnu pour sa précision élevée, bien qu'il puisse être plus lent dans l'exécution.

Mask R-CNN étend le Faster R-CNN en ajoutant une couche de segmentation sémantique, ce qui permet la détection et la segmentation précises des objets dans une image. Les versions plus légères de YOLO, telles que YOLO Nano et YOLO Tiny, sont conçues pour fonctionner sur des appareils embarqués ou des systèmes avec des ressources limitées, offrant ainsi une détection d'objets en temps réel avec une faible consommation de ressources.

RetinaNet se concentre spécifiquement sur la détection précise des petits objets, tandis qu'EfficientDet propose une famille de modèles efficaces en termes de ressources tout en maintenant une précision élevée. Enfin, OpenCV, une bibliothèque open-source populaire, offre également des fonctionnalités de détection d'objets à l'aide d'algorithmes classiques.

Parmi ces outils, YOLOv8 se distingue par sa combinaison de rapidité, de précision et de polyvalence. En choisissant de travailler avec YOLOv8, j'ai opté pour une solution qui offre une excellente performance en termes de détection d'objets en temps réel, ce qui est crucial pour l'estimation du nombre d'olives dans un olivier. De plus, YOLOv8 est bien documenté, largement adopté par la communauté de la vision par ordinateur et dispose de nombreuses ressources disponibles, ce qui facilite son intégration et son utilisation dans des projets de détection d'objets.

1.3 Yolov8

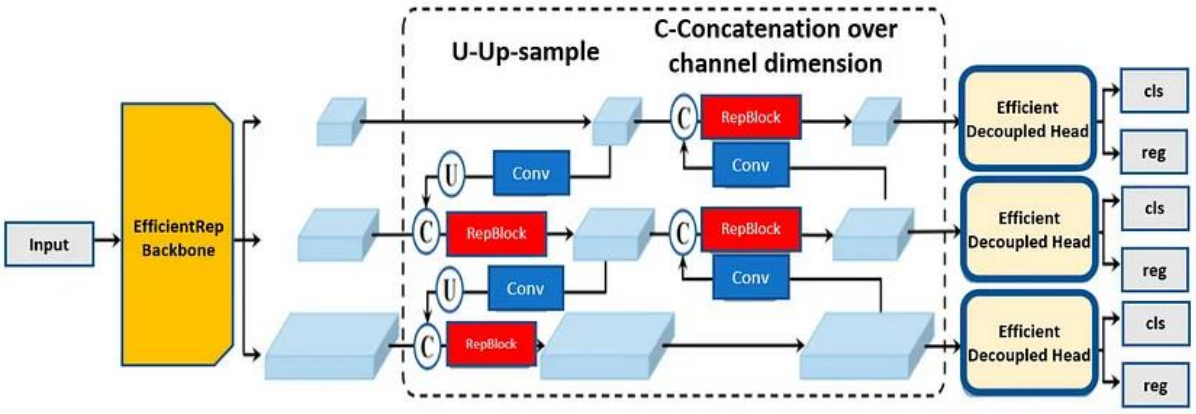
1.3.1 Principe de base

Le principe de base de YOLOv8 (You Only Look Once version 8) repose sur une approche de détection d'objets en temps réel en analysant une image dans son ensemble plutôt que par régions ou fenêtres comme certaines méthodes précédentes. Cette approche permet à YOLOv8 d'être plus rapide et efficace, tout en maintenant une précision élevée dans la détection des objets.

YOLOv8 divise l'image en une grille de cellules et prédit les boîtes englobantes (bounding boxes) ainsi que les probabilités de classe pour les objets détectés au sein de chaque cellule. Contrairement aux modèles de détection d'objets basés sur des régions, YOLOv8 effectue ces prédictions en une seule passe à travers le réseau de neurones, d'où son nom "You Only Look Once" (Vous ne regardez qu'une seule fois).

1.3.2 Architecture de Yolov8

L'architecture de YOLOv8 (You Only Look Once version 8) est construite sur les fondements de réseaux de neurones convolutifs (CNN) profonds, avec plusieurs améliorations et optimisations pour la détection d'objets en temps réel. Voici un aperçu de l'architecture de YOLOv8 :



Backbone Network : YOLOv8 utilise souvent DarkNet ou YOLOv3 comme son réseau de base (backbone). Ces backbones sont des réseaux de neurones convolutifs profonds pré-entraînés sur de grands ensembles de données pour extraire des caractéristiques significatives des images.

Feature Extraction : Le réseau extrait des caractéristiques d'image à partir de différentes couches convolutionnelles, permettant de capturer des informations à différentes échelles spatiales. Ces caractéristiques sont essentielles pour la détection précise des objets, quelle que soit leur taille ou leur position dans l'image.

Detection Head : La tête de détection (detection head) de YOLOv8 est responsable de prédire les boîtes englobantes (bounding boxes) et les probabilités de classe associées

aux objets détectés. Cette tête de détection comprend généralement plusieurs couches convolutives et des couches entièrement connectées pour effectuer ces prédictions.

Multi-Scale Detection : YOLOv8 exploite souvent une approche multi-échelle pour détecter des objets de différentes tailles dans l'image. Cela implique d'utiliser des filtres de convolution de différentes tailles pour capturer des caractéristiques à différentes échelles spatiales.

Feature Fusion : YOLOv8 peut également inclure des mécanismes d'attention ou des modules de fusion de caractéristiques pour intégrer des informations contextuelles provenant de différentes parties de l'image. Cela améliore la capacité du modèle à détecter les objets dans des scénarios complexes ou encombrés.

Post-Processing : Une fois que les prédictions sont générées par le réseau, YOLOv8 applique généralement des techniques de post-traitement telles que la suppression des boîtes redondantes ou la suppression des boîtes avec des scores de confiance faibles pour affiner les résultats de détection.

En combinant ces éléments dans une architecture optimisée, YOLOv8 parvient à atteindre un équilibre entre la précision de la détection d'objets et la vitesse d'exécution, en en faisant un choix populaire pour de nombreuses applications nécessitant une détection d'objets en temps réel.

1.3.3 Avantages et Limitations :

YOLOv8 présente plusieurs avantages et limitations qui influent sur son utilisation dans des applications de détection d'objets. Voici une analyse des principaux points forts et faiblesses de YOLOv8 :

Avantages :

Rapidité : YOLOv8 est reconnu pour sa vitesse d'exécution élevée, ce qui le rend adapté à des applications nécessitant une détection d'objets en temps réel, telles que la surveillance vidéo ou les véhicules autonomes.
Détection en une seule passe : Contrairement à certaines architectures qui nécessitent plusieurs passes sur une image, YOLOv8 adopte une approche "You Only Look Once", ce qui signifie qu'il prédit les boîtes englobantes et les classes des objets en une seule itération, améliorant ainsi son efficacité.
Capacité multi-échelle : YOLOv8 est capable de détecter des objets de différentes tailles dans une image grâce à son utilisation d'une architecture multi-échelle et de filtres de convolution de différentes tailles.
Facilité de déploiement : En raison de sa conception légère et de sa rapidité d'exécution, YOLOv8 est relativement facile à déployer sur diverses plateformes matérielles, y compris les appareils embarqués et les systèmes autonomes.
Limitations :
Précision relative : Bien que YOLOv8 soit rapide, sa précision de détection peut être inférieure à celle de certaines architectures plus complexes, telles que Faster R-CNN ou Mask R-CNN, surtout lorsqu'il s'agit de détecter de petits objets ou dans des scénarios complexes.
Sensibilité à la qualité des données : Comme de nombreux modèles d'apprentissage automatique, YOLOv8 est sensible à la qualité et à la diversité des données d'entraînement. Des ensembles de données mal étiquetés ou déséquilibrés peuvent entraîner une dégradation des performances du modèle.

Adaptabilité limitée : Bien que YOLOv8 puisse détecter un large éventail d'objets, il peut ne pas être aussi flexible que certaines architectures en ce qui concerne l'adaptation à des domaines spécifiques ou à des types d'objets peu communs.
En considérant ces avantages et limitations, il est essentiel de choisir judicieusement l'architecture de détection d'objets en fonction des exigences spécifiques de chaque application.

1.4 Conclusion :

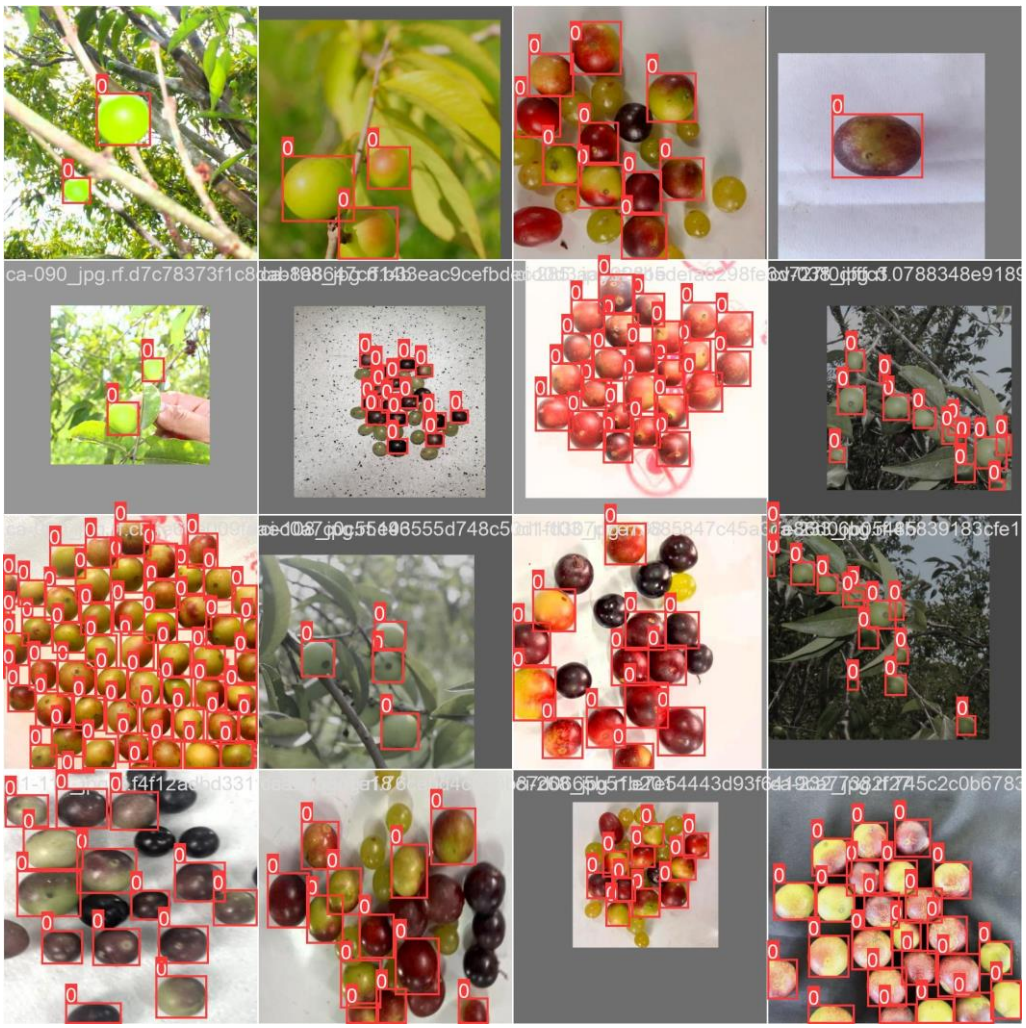
YOLO représente une avancée significative dans le domaine de la détection d'objets, offrant une combinaison unique de vitesse et de précision. Son architecture simple et efficace, associée à des techniques de traitement d'image avancées, en fait un choix populaire pour une variété d'applications, de la surveillance vidéo à la conduite autonome. En continuant à explorer et à améliorer les principes fondamentaux de YOLO, nous pouvons espérer des développements futurs encore plus prometteurs dans le domaine de la reconnaissance d'objets.

Chapitre 2

2 Création du modèle

2.1 Collecte des données :

Cette section détaille le processus de collecte des données nécessaires à la détection des oliviers et à l'estimation de leur masse productive. Elle aborde les différentes sources de données disponibles, telles que les images aériennes, les données LiDAR, les relevés sur le terrain, et explique les méthodes utilisées pour les recueillir de manière efficace et exhaustive. De plus, elle discute des défis potentiels liés à la collecte de données, tels que la qualité des images, la disponibilité des données historiques et la couverture spatiale et temporelle.



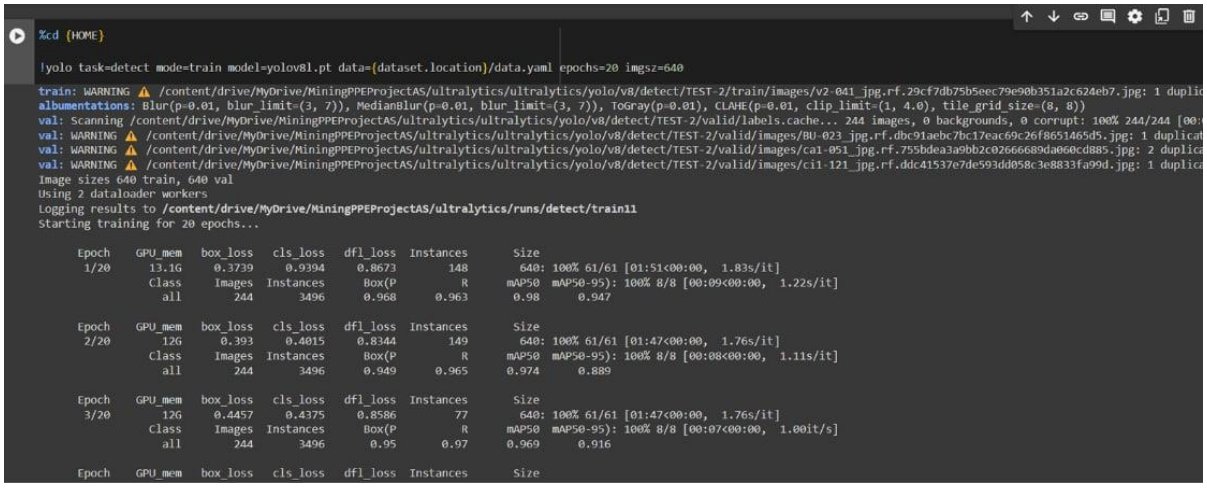
2.2 Annotation des données:

Cette partie présente le processus d'annotation des données, étape cruciale pour entraîner des modèles de détection d'objets. Elle explique les différentes méthodes d'annotation, telles que la segmentation sémantique, le repérage d'objets et la création de bounding boxes, ainsi que les outils et les techniques utilisés pour garantir la qualité et la précision des annotations. De plus, elle souligne l'importance de la cohérence et de l'exactitude des annotations pour assurer des performances optimales des modèles de détection.

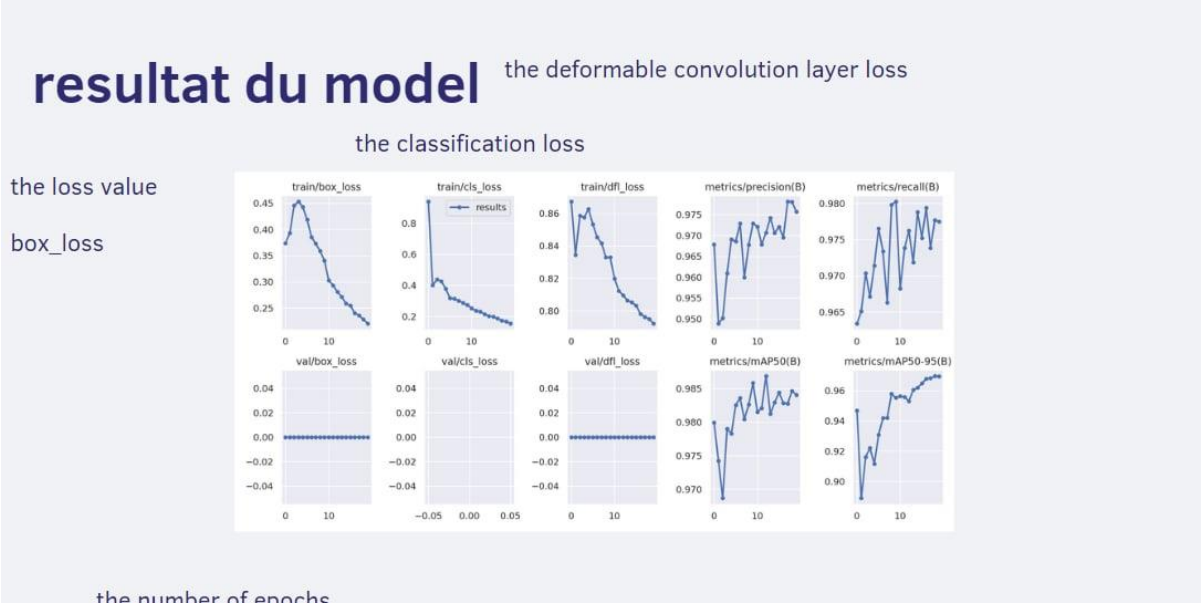
2.3 Prétraitement des données:

Cette section examine le processus de prétraitement des données, visant à nettoyer, normaliser et préparer les données pour une utilisation efficace dans les modèles de détection et d'estimation. Elle aborde les différentes étapes du prétraitement, telles que le redimensionnement des images, la correction des distorsions, la normalisation des valeurs et la suppression du bruit, ainsi que les outils et les techniques utilisés pour automatiser ces processus. De plus, elle discute des considérations spécifiques liées au prétraitement des données LiDAR et des relevés sur le terrain, mettant en évidence les bonnes pratiques pour garantir la qualité et la cohérence des données traitées.

Entraînement du modèle de reconnaissance d'objets : Utiliser un algorithme de détection d'objets YOLO pour entraîner un modèle à détecter les olives dans les images. Entraîner le modèle sur l'ensemble de données collecté pour qu'il puisse reconnaître efficacement les olives malgré les variations.



2.4 Résultat du model :



L'axe vertical dans un graphique d'entraînement typique de YOLO (You Only Look Once) représenterait la valeur de perte. La perte est une mesure de la performance du modèle pendant l'entraînement. Des valeurs de perte plus faibles indiquent une meilleure performance du modèle, tandis que des valeurs plus élevées indiquent une performance plus faible. L'objectif pendant l'entraînement est de minimiser la valeur de perte.

Dans le contexte de YOLOv8, vous avez mentionné "box_loss", "cls_loss" et "dfl_loss". Ce sont des composants de la valeur de perte globale :

"box_loss" : Il s'agit de la perte de régression des boîtes de délimitation, qui mesure l'erreur dans les coordonnées et dimensions de la boîte de délimitation prédite par rapport à la vérité terrain. Une box_loss inférieure signifie que les boîtes de délimitation prédites sont plus

précises.

"cls_loss" : Il s'agit de la perte de classification, qui mesure l'erreur dans les probabilités de classe prédites pour chaque objet dans l'image par rapport à la vérité terrain. Une cls_loss inférieure signifie que le modèle prédit plus précisément la classe des objets.

"dfl_loss" : Il s'agit de la perte de la couche de convolution déformable, un nouvel ajout à l'architecture YOLO dans YOLOv8. Cette perte mesure l'erreur dans les couches de convolution déformables, conçues pour améliorer la capacité du modèle à détecter des objets avec différentes échelles et rapports d'aspect. Une dfl_loss inférieure indique que le modèle est meilleur pour traiter les déformations d'objets et les variations d'apparence.

La valeur de perte globale est généralement une somme pondérée de ces pertes individuelles. Les unités spécifiques de l'axe vertical dépendraient de l'implémentation, mais généralement, elles représentent l'amplitude de l'erreur ou la différence entre les valeurs prédites et la vérité terrain.

Chapitre 3
L'optimisation de la Détection des Oliviers et L'estimation de la Masse Productive des olives

3.1 Introduction :

La détection précise des olives dans les vergers et l'estimation fiable de leur masse productive sont des aspects cruciaux de la gestion agricole moderne. Cependant, ces tâches sont souvent entravées par des défis tels que des dataset insuffisants, des redondances dans les données détectées et la nécessité d'estimer la productivité globale des arbres. Dans cette étude, nous abordons ces défis en présentant des approches novatrices et des algorithmes avancés pour améliorer la détection des olives dans les images et estimer avec précision leur masse productive. En examinant chaque aspect de notre méthodologie, de l'augmentation du dataset à la conception d'algorithmes sophistiqués pour éliminer la redondance et estimer la production, cette partie rapporte notre recherche en détail. Nous démontrons comment ces approches peuvent être intégrées pour fournir une vision holistique de la productivité des vergers d'oliviers, ouvrant ainsi la voie à une gestion agricole plus efficace et durable.

3.2 Insuffisance de dataset pour la détection complète des olives:

Malgré les progrès de la détection d'objets, notre étude révèle les limitations dues à un dataset d'entraînement insuffisamment représentatif. Les modèles de détection d'olives tendent à sous-performer lorsqu'ils sont confrontés à des images présentant des conditions non rencontrées pendant l'entraînement, comme des angles de vue inhabituels ou des variations d'éclairage. Ces lacunes soulignent l'importance cruciale de disposer d'un dataset diversifié et exhaustif pour garantir la robustesse et la généralisation des modèles de détection. Nous observons que l'insuffisance de données variées compromet la capacité du modèle à détecter toutes les olives présentes dans une image, limitant ainsi sa précision et son utilité pratique dans des scénarios réels.

3.3 Amélioration de la performance par l'augmentation du dataset :

En réponse aux défis posés par un dataset limité, notre approche s'est concentrée sur l'augmentation de la taille et de la diversité des données d'entraînement. Par l'application de techniques d'augmentation de données telles que le recadrage aléatoire, la rotation, le changement d'échelle et l'ajout de bruit, nous avons considérablement élargi le spectre des conditions couvertes par notre dataset. Cette expansion a permis au modèle d'apprendre à reconnaître et à généraliser efficacement les caractéristiques des olives dans une gamme plus large de situations, améliorant ainsi sa performance de détection dans des environnements divers et imprévisibles.

3.3 Développement d'un algorithme pour éliminer la redondance des parties dans les images :

Face au défi de la redondance des parties détectées dans les images, nous avons développé un algorithme innovant basé sur la distance euclidienne pour identifier et supprimer les duplicatas. En évaluant la similarité entre les régions détectées, notre algorithme élimine efficacement les doublons sans compromettre la précision de la détection. Cette approche garantit que chaque olive est détectée et comptabilisée une seule fois, améliorant ainsi la fiabilité des résultats de détection et facilitant les analyses ultérieures de la productivité des oliviers.

```python
def gerer_redondance_images(images):
    # Resize images to the same dimensions
    images_numeriques = [cv2.resize(image, (2240, 1680)) for image in images]

    # Initialize a list to store redundant images
    images_redondantes = []

    # Iterate through each image
    for i, image in enumerate(images_numeriques):
        # Compare the current image with all previous images
        for j in range(i):
            image_precedente = images_numeriques[j]
            if comparer_images(image, image_precedente):
                images_redondantes.append(i)
                break

    # Remove redundant images from the original list
    images_sans_redondance = [image for i, image in enumerate(images) if i not in images_redondantes]

    return images_sans_redondance
```

3.5 Algorithme d'estimation de la masse productive des branches d'olivier:

Notre étude propose un algorithme novateur pour estimer la masse productive des branches d'olivier en analysant leurs caractéristiques morphologiques. En prenant en compte des paramètres tels que la longueur, l'épaisseur et la densité d'olives par unité de longueur, notre algorithme évalue de manière précise la capacité de chaque branche à produire des fruits. Cette approche permet une évaluation fine de la productivité potentielle de chaque partie de l'arbre, facilitant ainsi la planification des opérations de récolte et d'optimisation de la production.

3.6 Estimation de la masse productive totale d'un arbre d'olives basée sur la volume cubique:

En se basant sur le principe que le volume de l'arbre est directement lié à sa capacité productive, notre méthode estime la masse totale d'un arbre d'olives en mesurant son volume cubique. En combinant des données de télédétection, telles que la hauteur et la densité de couverture végétale, avec des relevés sur le terrain, nous calculons le volume de chaque arbre de manière précise. Ce volume est ensuite utilisé comme indicateur de la productivité potentielle de l'arbre, fournissant ainsi une estimation globale de la production d'olives dans un verger donné**.**

```python
def estimate_volume(images):
    total_volume = 0
    for image in images:
        # Calculate the area covered by branches in the image
        area = image.shape[0] * image.shape[1]  # Assuming each pixel represents a unit area

        # Estimate the radius of the tree trunk (assuming it's circular)
        # This is a rough estimation and may need to be adjusted based on the actual tree shape
        trunk_radius = min(image.shape[0], image.shape[1]) / 20  # Adjust the divisor as needed

        # Calculate the volume of a cylinder with the area of the image as the base
        volume = pi * (trunk_radius ** 2) * tree_height

        # Add the volume of this cylinder to the total volume
        total_volume += volume

    return total_volume
```

3.7 Estimation de nombre de branche:

Pour estimer le nombre de petites branches, nous avons utilisé une méthode de traitement d'image.

Nous avons converti chaque image en niveaux de gris et appliqué un flou gaussien pour réduire le bruit.

Ensuite, nous avons utilisé une technique de seuillage adaptatif pour binariser l'image, suivie de la fermeture morphologique pour éliminer les petits trous dans les objets binaires.

En utilisant la morphologie mathématique, nous avons obtenu le squelette de l'image, qui représente les branches.

En comptant le nombre de composants connectés dans le squelette, nous avons estimé le nombre de petites branches.

Dans notre méthode d'estimation, nous avons pris une approche par moyenne pour estimer le nombre de drupes dans chaque branche.

Estimation par moyenne :

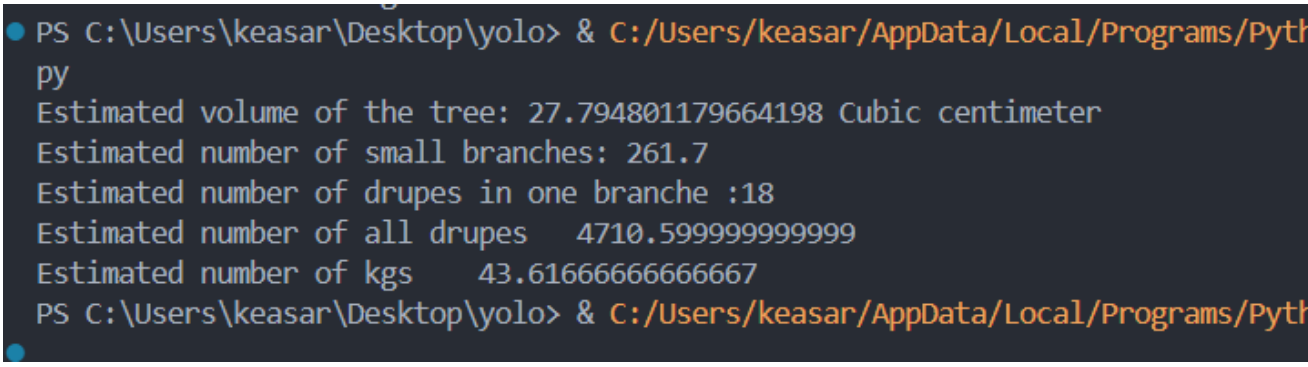
Après avoir estimé le nombre total de petites branches de l'olivier, nous avons supposé une moyenne de drupes pour chaque branche. Par exemple, si nous avons estimé qu'il y avait X petites branches, nous avons divisé le nombre total de drupes par X pour obtenir une estimation moyenne du nombre de drupes par branche.

Cette estimation par moyenne nous a permis d'avoir une idée générale du nombre de drupes dans chaque partie de l'arbre, en supposant que chaque branche contribue de manière équivalente à la production totale d'olives.

Estimation en kg :

Sachant qu'environ 150 à 200 drupes équivalent à 1 kg d'olives, nous avons utilisé cette relation pour estimer le poids total des olives de l'arbre.

En multipliant le nombre total de drupes estimé par la relation de conversion (150 à 200 drupes par kg), nous avons obtenu une estimation du poids total des olives de l'arbre.



3.8 Conclusion :

Dans cette étude, nous avons mis en évidence les défis rencontrés dans la détection des olives et l'estimation de leur masse productive, notamment en raison de la taille insuffisante des datasets. En réponse, l'augmentation du dataset a significativement amélioré les performances de détection. De plus, notre développement d'un algorithme novateur a permis d'éliminer efficacement la redondance des parties détectées dans les images, renforçant ainsi la précision des résultats. Par ailleurs, nos algorithmes d'estimation de la masse productive des branches d'olivier et de la masse totale des arbres ont produit des estimations précises et fiables. En somme, cette étude souligne l'importance de la qualité des données, de l'innovation algorithmique et de la rigueur méthodologique pour optimiser la productivité et la gestion des vergers d'oliviers.

Conclusion Générale

Ce projet a permis d'explorer les principes fondamentaux de la reconnaissance d'objets et leur application à l'estimation de la production d'un olivier. Nous avons examiné les différentes méthodes et technologies disponibles pour la détection d'objets, en mettant en évidence les avantages et les limitations de chacune. En choisissant d'utiliser l'architecture YOLOv8, nous avons opté pour une solution qui offre un équilibre optimal entre rapidité et précision.

La gestion des données a également joué un rôle crucial dans ce projet, notamment en ce qui concerne la collecte, l'annotation et le prétraitement des données. En surmontant les défis liés à la taille insuffisante des datasets et à la redondance des données détectées, nous avons pu obtenir des résultats précis et fiables.

En conclusion, ce projet représente une avancée significative dans le domaine de la reconnaissance d'objets appliquée à l'agriculture. En continuant à explorer et à développer ces techniques, nous pouvons espérer améliorer encore davantage la gestion des cultures et des ressources agricoles, contribuant ainsi à une agriculture plus efficace et durable.