

Contents

1	Introduction	2
2	État de l'art	2
2.1	Intelligence Artificielle (IA)	2
2.2	Détection d'objets	2
2.2.1	Revue de la littérature	2
2.2.2	Algorithmes de détection	2
2.2.3	Yolo	2
2.2.4	Algorithme R-CNN (Regions with Convolutional Neural Networks)	2
2.2.5	Algorithme Fast R-CNN (Regions with Convolutional Neural Networks Fast)	2
2.2.6	Algorithme Faster R-CNN	2
2.2.7	Comparaison des algorithmes	3
3	Implémentation	3
3.1	Environnement de développement	3
3.1.1	Outils et bibliothèques	3
3.2	Création du jeu de données	3
3.3	Collection d'image	3
3.4	Annotation	4
3.5	Modèle YOLOv5	4
3.5.1	Architecture	4
3.5.2	Installation de l'environnement YOLOv5	4
3.5.3	Entraînement	4
4	Résultats et Discussion	4
4.1	Métriques de performance	4
4.1.1	Précision	4
4.2	Application web	5
5	Perspectives	5
6	Conclusion	5

1 Introduction

La mangue joue un rôle crucial dans l'agriculture au Burkina Faso, représentant une part significative des exportations. Cependant, la détection manuelle de la maturité des mangues est sujette à des erreurs et peut entraîner des retards dans la distribution. Cette étude vise à développer un programme de détection automatisée de mangues mûres en utilisant l'algorithme YOLOv5, offrant ainsi une solution objective et efficace. L'objectif est d'améliorer les processus de tri, de récolte et de distribution, réduisant ainsi les pertes post-récolte et optimisant les opérations logistiques. L'étude aborde l'analyse, la conception et l'implémentation du programme, avec une évaluation des performances basée sur des métriques telles que la précision et le rappel. En contribuant à l'amélioration des techniques de détection, cette recherche promet d'optimiser la production et la récolte de mangues au Burkina Faso.

2 État de l'art

2.1 Intelligence Artificielle (IA)

L'intelligence artificielle (IA) est une branche de l'informatique qui se concentre sur l'automatisation du comportement intelligent. Son but est de créer des systèmes capables de fonctionner de manière intelligente et indépendante. Il est subdivisé en plusieurs domaines : le Machine Learning, le deep learning, etc.

2.2 Détection d'objets

2.2.1 Revue de la littérature

Plusieurs approches ont été utilisées pour la détection des mangues par le passé. On peut citer : Susovan Jana, Saikat Basak et Ranjan Parekh ont proposé une méthode d'apprentissage en profondeur utilisant un R-CNN plus rapide pour la classification d'un multi-fruits, Akshay Ramesh Amrutkar, Hemant Balu Jaisingpure, Pavan Ashokrao Bhujade pour identifier les différentes étapes de maturation des fruits climatériques comme la mangue en utilisant l'IDE Arduino qui a le rôle d'envoyer le résultat de la détection à distance via le module GSM, etc.

2.2.2 Algorithmes de détection

La détection d'objets est un phénomène de vision par ordinateur qui implique la détection de divers objets dans des images ou des vidéos numériques.

2.2.3 Yolo

L'algorithme YOLO (You Look Only One) a été proposé en 2015 [10]. Par la suite, d'autres versions de cet algorithme ont vu le jour de la version YOLOv2 à la version YOLOv8. Il sera donc utilisé pour la détection des mangues.

2.2.4 Algorithme R-CNN (Regions with Convolutional Neural Networks)

L'algorithme R-CNN (Regions with Convolutional Neural Networks) détecte des objets en identifiant d'abord des zones candidates, extrayant ensuite des caractéristiques via des réseaux neuronaux, et enfin classifiant et ajustant précisément les boîtes englobantes des objets détectés. Cette méthode a été une avancée significative dans la détection d'objets dans les images.

2.2.5 Algorithme Fast R-CNN (Regions with Convolutional Neural Networks Fast)

2.2.6 Algorithme Faster R-CNN

L'algorithme Fast R-CNN améliore la vitesse de détection d'objets par rapport à R-CNN en intégrant une seule passe de traitement pour l'extraction des caractéristiques et la classification. Il utilise une région de

proposition unique pour l'ensemble de l'image, optimisant ainsi l'efficacité tout en maintenant une précision élevée. Cette évolution a contribué à rendre la détection d'objets plus rapide et plus efficace.

2.2.7 Comparaison des algorithmes

Après une comparaison approfondie des différents algorithmes, nous avons décidé d'utiliser YOLOV5, car YOLOv5s est nettement meilleur que les autres versions en termes de performances et de vitesse.

3 Implémentation

3.1 Environnement de développement

- Le langage Python. le langage Python pour l'implémentation de notre algorithme.
- Google-Colab qui permet l'écriture et l'exécution du code Python dans un navigateur avec aucune configuration requise, accès sans frais au GPU.
- Roboflow est une plateforme de vision par ordinateur qui fournit des outils et des ressources pour faciliter le processus de développement d'applications basées sur la vision par ordinateur. La plateforme prend en charge plusieurs architectures de modèles populaires, comme YOLO, SSD, Faster RCNN, etc. Il offre des options d'exportation des modèles pour différents frameworks tels que TensorFlow, PyTorch, etc

Nous avons choisi d'utiliser Roboflow pour sa simplicité, la présence des datasets d'images déjà annotés. En plus cette plateforme intègre l'augmentation des données en utilisant la rotation, le flippe et autres. Aussi, après l'annotation, c'est possible de diviser les images en trois (3) parties : les images d'entraînements, de tests et de validations. Roboflow utilise plusieurs formats d'annotations tels que les formats YOLOv5 PyTorch qui est adapté dans notre étude et le format Pascal VOC. Labeling a été utilisé car il est en local et simple à utiliser.

3.1.1 Outils et librairies

- Pytorch. YOLOv5 est implémenté en utilisant le framework PyTorch[13]. Cela signifie que le code source de YOLOv5 est écrit en utilisant les fonctionnalités et les structures de PyTorch.
- OpenCV possède plusieurs fonctionnalités qui favorisent, le prétraitement des images, effectue des opérations de redimensionnement, de filtrage, de conversion de couleurs, et autres actions. Étant donné que YOLOv5 s'applique dans la détection d'objets, les résultats de détection avec YOLOv5 peuvent ensuite être utilisés par OpenCV pour effectuer d'autres actions, telles que le suivi d'objets, l'affichage des résultats, etc.
- Streamlit est un framework open-source pour la création rapide d'applications web en Python

3.2 Création du jeu de données

Dans cette recherche, un ensemble de données des images de mangues était nécessaire. Pour obtenir un modèle YOLOv5 robuste, il est recommandé de s'entraîner avec plus de 1 500 images par classe et plus de 10 000 instances par classe.

3.3 Collection d'image

1. Diversité. Il est toujours préférable pour le modèle de voir l'objet dans autant de contextes différents que possible, cela l'aide à apprendre de manière robuste

- 2. Résolution moyenne. Les images haute résolution prennent beaucoup plus de temps à traiter et peuvent capturer des détails inutiles, tandis qu'une basse résolution ne contiendra aucune fonctionnalité à apprendre pour le modèle. Il est préférable de redimensionner vos images afin que l'objet soit plus grand que 32 x 32 pixels
- 3. Vaste ensemble de données. Plus le nombre d'images avec les objets que le modèle voit, mieux il apprend. Ainsi, des images ont été collectées au Burkina Faso et en ligne sur RoboFlow

3.4 Annotation

Le critère d'annotation est la couleur vert, jaune et orange. labelImg et RoboFlow ont été utilisés pour cette annotation. Les images annotées sont ensuite enregistrées sous forme de fichiers XML au format PASCAL VOC ou au format YOLO.

3.5 Modèle YOLOv5

3.5.1 Architecture

L'architecture des détecteurs d'objets à un étage (comme YOLO) est composée de trois composants : Backbone, Neck et Head pour effectuer des prédictions denses.

3.5.2 Installation de l'environnement YOLOv5

Le code source de YOLOv5 est open source et est disponible sur Github [23]. Ce code source est composé de plusieurs fichiers, organisés de manière hiérarchique dans une structure de répertoires.

3.5.3 Entraînement

Notre algorithme applique la détection sur deux (2) types de mangues : "classe 1 : ripe" pour les mangues mûres et "classe 2 : unripe" pour les mangues non mûres.

Le framework YoloV5 a été généré en local pour travailler. Plusieurs fonctions sont utilisées pour effectuer l'entraînement du modèle. Organisation des data-set (Jeu de données) : (Test) dans lequel on a : (image, label), (Train) dans lequel on a : (images, labels, label.caches), (Valid) dans lequel on a : (images, labels, label.caches). Le cache est également utilisé pour accélérer le chargement des données. Le modèle YOLOv5s a été entraîné au cours de notre étude, avec un dataset d'images de mangues composé de plus de 500 images de mangues mûres et non mûres.

4 Résultats et Discussion

4.1 Métriques de performance

Une fois l'entraînement terminé, la performance du modèle peut être évaluée en utilisant l'ensemble de données de validation. Pour l'évaluation de la performance du modèle entraîné YOLOv5 dans la détection des mangues, nous avons adopté trois métriques : le rappel, la précision et la matrice de confusion et le F1-score. Les courbes de ces métriques sont construites en modifiant le seuil de détection. Par conséquent, l'efficacité du modèle peut être mesurée sur la base de ces mesures d'évaluation essentielles du modèle. D'autre part, une signification distincte et plus profonde de chacune de ces métriques est présentée dans ce qui suit.

4.1.1 Précision

Cette métrique mesure le pourcentage de résultats de détection pertinents [13]. Cela peut être déterminé à l'aide de l'équation suivante :

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

Dans cette équation, TP (True Positive) qui est le nombre d'objets correctement détectés et FP(False Positive) qui est le nombre d'objets non détectés. Ainsi, cette équation calcule le nombre de prédictions de classe positives qui appartiennent à la classe positive. Cela mesure à quel point les prédictions de la boîte englobante sont vraies et correctes. Une précision élevée signifie que le modèle fait moins d'erreurs de faux positifs.

4.2 Application web

Notre application web a été développée avec le framework Streamlit présenté ci-dessus, qui nous a permis de créer une interface conviviale et interactive pour la détection d'objets. Les utilisateurs peuvent ainsi télécharger leurs propres images ou vidéos, les soumettre à l'algorithme de détection de mangues et visualiser les résultats de manière claire et compréhensible. L'algorithme applique la détection sur images des mangues ou sur des vidéos, mais seulement la détection avec les images est prise dans notre application web

Problème : Le modèle a été entraîné avec beaucoup plus de mangue non mûres que des mangues mûres. Ainsi lors des chargement d'image, des mangues mure peuvent être vu comme des maques non mûres.

5 Perspectives

Étant donné que notre programme rencontre quelques difficultés de détection, il est nécessaire dans les futurs travaux de prendre en compte certains aspects.

- Amélioration de l'algorithme de détection. Il est nécessaire de disposer d'une base de données d'images diversifiée et représentatives des différentes variations de couleur, de texture et de forme des mangues. Cela permettra de mieux entraîner le modèle et de le rendre plus robuste face aux variations des caractéristiques des mangues.
- Validation expérimentale. Il serait également bénéfique de réaliser des expériences et des tests approfondis en optimisant les hyper-paramètres dans des conditions réelles pour évaluer les performances du programme de détection des mangues.
 - Évaluation la qualité des mangues pour identifier les mangues malades et non malades - Intégration du programme à un drone pour la cueillette automatique des mangues.

6 Conclusion

Dans cette étude, nous avons proposé un modèle personnalisé d'apprentissage en profondeur YOLOv5 utilisé pour la détection des mangues dans des images capturées à différentes distances. L'approche proposée a été évaluée à l'aide des images réelles prises sur le terrain et des images librement disponibles sur internet composé de différents types de mangues de différentes tailles, de formes et de couleurs. Le modèle a donné des résultats satisfaisants grâce aux réglages fins des paramètres du modèle pour qu'il s'adapte mieux à notre contexte. De plus, nous avons utilisé l'augmentation des données pour fournir à la phase de formation du modèle suffisamment de données pour qu'il soit correctement formé. Les résultats montrent que le réseau formé peut détecter des mangues de différents types avec une bonne précision.