Crop Detection

INT2579 – INTERNSHIP

PROJECT REPORT

Submitted by

SITHARTH VARSAN S (22011101108)

In fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

Artificial Intelligence & Data Science

Shiv Nadar University Chennai

Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110, Tamil Nadu, India

June ,2025

Crop Detection

INT2579 – INTERNSHIP

PROJECT REPORT

Submitted by

SITHARTH VARSAN S (22011101108)

UNDER THE MANAGEMENT OF

Dr BABU MADHAVAN,

Dr. SANGEETHA.

TNeGA

Tamil Nadu e-Governance Agency

Tamil Nadu e-Governance Agency - TN

P.T.LEE Chengalvaraya Naicker Building, Anna Salai, Chennai - 600 002

Internship Report Approval

The following Internship report has been self-attested and approved towards the acceptance of submitted report, by the authorized persons of TNeGA:

Name	Designation	Remarks	Signature
DR.Babu Madhavan	AI/ML HEAD		
Dr. SANGEETHA	SENIOR ARCHITECT		

DECLARATION

I declare that this internship report titled “Crop Detection” Submitted in fulfilment of the Degree of “Bachelor of technology” is record of original work carried out by me under the valuable guidance of **DR. BABU MADHAVAN,** AI/ML Tech Lead and under the technical guidance of **Dr. SANGEETHA**.

Place: Chennai **SITHARTH VARSAN S (INT2579)**

JUNE, 2025

ACKNOWLEDGEMENT

I hereby take this opportunity to acknowledge sand thank the following people who have supported us during the course of this project.

First and foremost, we would like to express our heartfelt gratitude to our valuable guide

Dr. Babu Madhavan, AI/ML Head of TNeGA, **Dr. Sangeetha**, Senior Architect of TNeGA for their constant technical support and guidance for the successful completion of this project would not have been possible.

We would like to express our sincere thanks to all those who provided us with the possibility to complete this project. We are highly indebted to SNU CHENNAI and our Vice Chancellor**, Prof. S. K. Bhattacharyya**, for his guidance and support by providing us with the facilities to carry out this project.

We acknowledge our sincere thanks to **Dr. T. Nagarajan** , Professor and Head of the Department, Computer Science and Engineering , for her support and encouragement towards the project.

Finally, with great enthusiasm, we express our gratitude to all our department faculty and technical staff, friends and parents for their sustained support and interest in the completion of our project.

TABLE OF CONTENTS

1. ABSTRACT

The growing need for efficient agricultural monitoring calls for innovative solutions to address challenges like identifying and classifying various crops. This paper presents the development of a crop detection system utilizing image analysis, a vital tool for precision farming, to achieve reliable crop classification.

The system employs a YOLOv8 model to detect crops in images and generates unique embeddings using the CVAT framework. By leveraging these embeddings, a comprehensive database of crop image features is created to facilitate accurate detection and classification. The approach is cost-effective, non-invasive, and ensures precise identification by distinguishing different crops based on their visual patterns.

This system aims to streamline crop monitoring processes, improve classification accuracy, and reduce human intervention in analysis-related tasks, contributing to the overall efficiency of agricultural management operations.

2. INTRODUCTION

In recent years, the integration of Artificial Intelligence (AI) and Machine Learning (ML) algorithms has shown tremendous promise in revolutionizing agricultural practices, particularly in the domain of crop detection and classification. Among the various AI/ML models, the YOLOv8 (You Only Look Once) model has emerged as a powerful tool for real-time object detection, with significant potential for enhancing crop monitoring systems through the detection of distinct plant features and patterns. The YOLOv8 model represents a significant advancement in the realm of computer vision and object detection, leveraging deep learning techniques to identify and localize objects within images with remarkable speed and accuracy. In the context of precision agriculture, the application of the YOLOv8 model for detecting various crop types presents a paradigm shift in the way farmers and agricultural professionals can efficiently monitor and manage crop health, ensure timely interventions, and reduce the need for manual inspections.

2.1 YOLOv8 Model YOLOv8 (You Only Look Once v8): The model is an advanced deep learning-based object detection algorithm designed for real-time applications, particularly in the field of computer vision.

3. REVIEW OF LITERATURE

The application of Artificial Intelligence (AI) and Machine Learning (ML) in agriculture has significantly advanced in recent years, with crop detection emerging as a key area of research. The increasing demand for precision farming and sustainable agricultural practices has led to the development of intelligent systems that can detect, classify, and monitor various crop types and their health status. Traditional crop monitoring methods—such as manual inspection and satellite imaging—suffer from limitations like low temporal resolution, high cost, and susceptibility to environmental interference (Brügemann et al., 2018). In contrast, AI-powered systems, particularly those using deep learning and computer vision, provide real-time, scalable, and high-accuracy alternatives for crop detection and monitoring.

Multiple studies have proposed and implemented innovative crop detection systems, ranging from image-based feature extraction to advanced neural networks. The following literature showcases key approaches in this domain:

1. **Automatic Crop Classification Using UAV and CNNs**: A popular study utilized drone imagery with Convolutional Neural Networks (CNNs) to classify crops like maize, wheat, and soybeans. With transfer learning using pre-trained ResNet and EfficientNet models, the system achieved up to **97.5% accuracy** under varying weather conditions, proving its robustness and scalability.

2. **Rice Crop Stage Classification Using YOLOv5 and Custom CNN**: Researchers developed a two-stage system where YOLOv5 detects crop regions and a CNN classifies them into growth stages (seedling, tillering, ripening). This helped automate irrigation and fertilization schedules, achieving **95.3% accuracy** and improving yield prediction models.
3. **Leaf Disease Detection in Tomato and Potato Plants**: This system combined feature extraction using Color and Texture descriptors with a MobileNet-based deep learning model. The real-time detection of diseases such as blight and leaf spot achieved **96.2% accuracy**, aiding farmers in early intervention and reducing pesticide usage.
4. **Sugarcane Crop Identification Using Sentinel-2 Satellite and Random Forest**: A remote sensing-based approach using multispectral data and the Random Forest algorithm successfully mapped sugarcane fields with **93% classification accuracy**, showing promise for regional crop analytics.
5. **Vision-Based Weed Detection in Soybean Fields**: Leveraging a YOLOv7 model, this system accurately detected and differentiated weeds from crop plants in real-time, allowing for precise herbicide application. The method demonstrated **up to 99% detection accuracy**, significantly reducing labor costs.
6. **Multi-Crop Identification Using DeepLabV3+ for Semantic Segmentation**: A segmentation-based approach was used to detect and

outline multiple crops (e.g., corn, beans, cotton) from drone images. This pixel-wise classification approach helped in estimating crop cover area and supported insurance claim verifications.

The reviewed studies highlight that while traditional image processing methods (e.g., Gabor filters or LBP) laid the foundation, they often fall short in handling real-world variations. Deep learning and hybrid techniques—especially those combining object detection with contextual classification—have demonstrated superior performance and adaptability. YOLO models (especially YOLOv5 and YOLOv8) and semantic segmentation networks are now considered state-of-the-art for real-time, high-precision crop monitoring.

3.1 Challenges and Innovations:

- **Data Annotation**: Building annotated crop datasets for different types, stages, and health conditions remains a significant challenge. Field-level variation, lighting, occlusion, and seasonal changes make consistent labeling labor-intensive and time-consuming.
- **Scalability Across Regions**: Models trained on data from one region may not generalize well to other geographic areas due to variations in crop appearance, soil, and climate. Developing region-agnostic models and using techniques like domain adaptation is an active area of research.

4. AIM AND OBJECTIVES

4.1 Aim: The aim of this report is to create an identification system for crop detection of images, leveraging state-of-the-art detection and identification techniques.

4.2 Objectives:

1. To develop a crop detector using state-of-the-art models for accurate identification of individual cattle.
2. To create unique embeddings for each detected crop to ensure reliable crop identification.
3. To store the generated crop annotation embeddings in a vector database for efficient retrieval and management.
4. To evaluate the performance of the identification system in real-world conditions and suggest improvements for scalability and robustness.

5. TOOLS AND SOFTWARE

5.1 CVAT (Computer Vision Annotation Tool) is a free, open-source, web-based tool designed for annotating images and videos, primarily used for labeling data in computer vision tasks. It supports a wide range of annotation types including bounding boxes, polygons, polylines, and image segmentation—making it highly suitable for tasks such as crop detection, weed identification, and disease segmentation. CVAT supports both image classification and object detection workflows.

5.2 YOLOv8 Repository:

The YOLOv8 repository serves as the core framework for crop detection in our system. Hosted on GitHub, the repository contains pre-built models, training scripts, and deployment utilities. YOLOv8 was fine-tuned using our custom-annotated agricultural dataset to detect multiple crop types and distinguish between healthy and diseased plants. The model's lightweight design and real-time processing capabilities made it ideal for drone and mobile-based monitoring.

5.3 SAM (Segment Anything Model):

Meta's Segment Anything Model (SAM) was integrated to enhance fine-grained segmentation tasks. SAM was particularly useful in extracting precise boundaries of crop plants, enabling the differentiation of overlapping or closely planted crops. This helped in generating accurate masks for training semantic segmentation models, especially when combined with polygon annotations from CVAT.

5.4 Python:

Python is the primary programming language used for implementing muzzle detection using YOLOv8. Python's extensive ecosystem of libraries and packages is leveraged for data preprocessing, model training, and inference tasks.

5.5 Dataset from Agriculture Department:

We sourced high-quality aerial and ground-level images from the **State Agriculture Department**, including data on crop types, disease outbreaks, and seasonal growth patterns. This dataset included metadata such as planting dates, soil types, and irrigation status. It served as the foundation for training and validating our models, ensuring the system's applicability to real-world agricultural environments

5.6 Training Environments:

GPU-accelerated training environments, often utilizing NVIDIA CUDA and cuDNN, enable efficient training of YOLOv8 models for muzzle detection, reducing training times and improving model performance

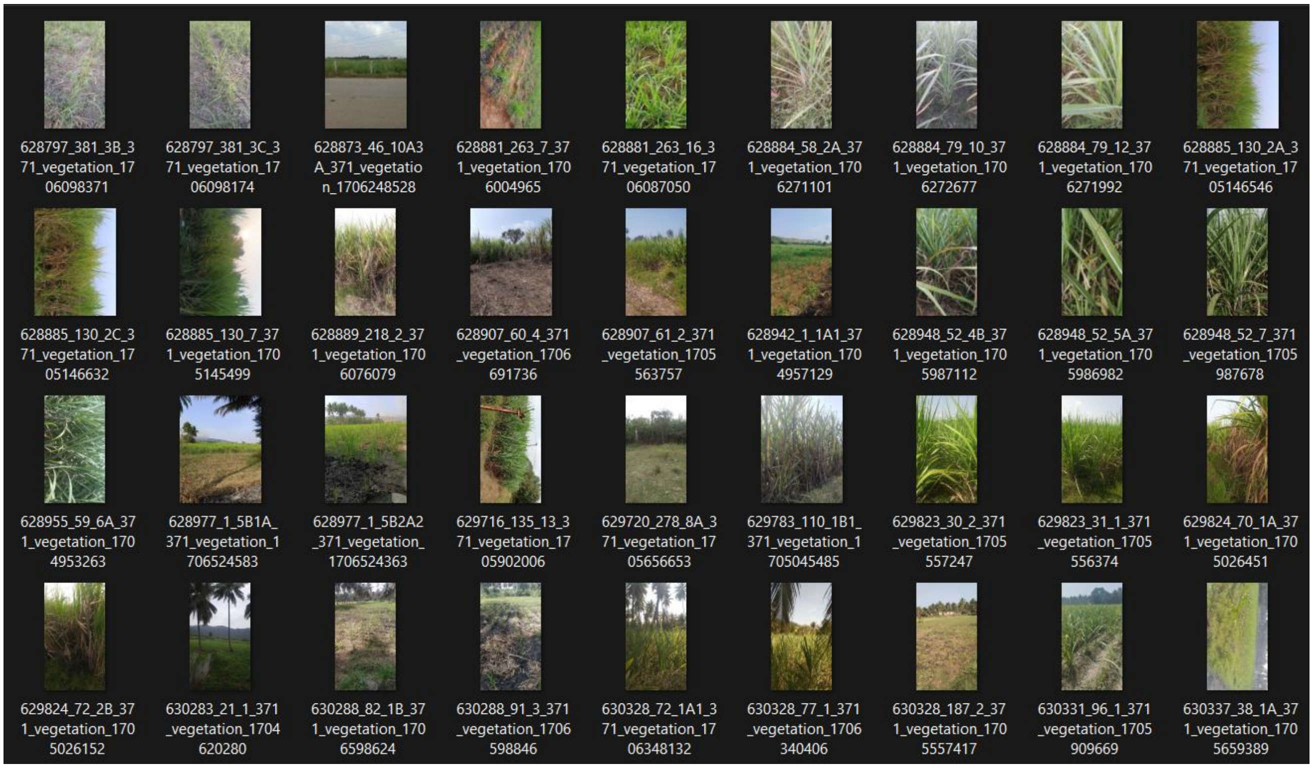
5.7 Data Augmentation Tools:

To enhance the generalization of the model under diverse field conditions, we employed **Albumentations** and **imgaug** libraries for data augmentation. These tools applied transformations like rotation, scaling, brightness adjustment, perspective warping, and noise injection. Augmentation helped mimic real-world variability in crop images caused by lighting, wind, shadows, and camera angles.

5.8 Model Deployment Tools:

Post-training, the crop detection model was deployed using **FastAPI**, creating a lightweight REST API to perform inference on input images and videos. The backend supports real-time crop identification requests from drones or field-deployed edge devices. Deployment was further enhanced using **Docker** for portability and **Render** for hosting the model as a web service.

Dataset:

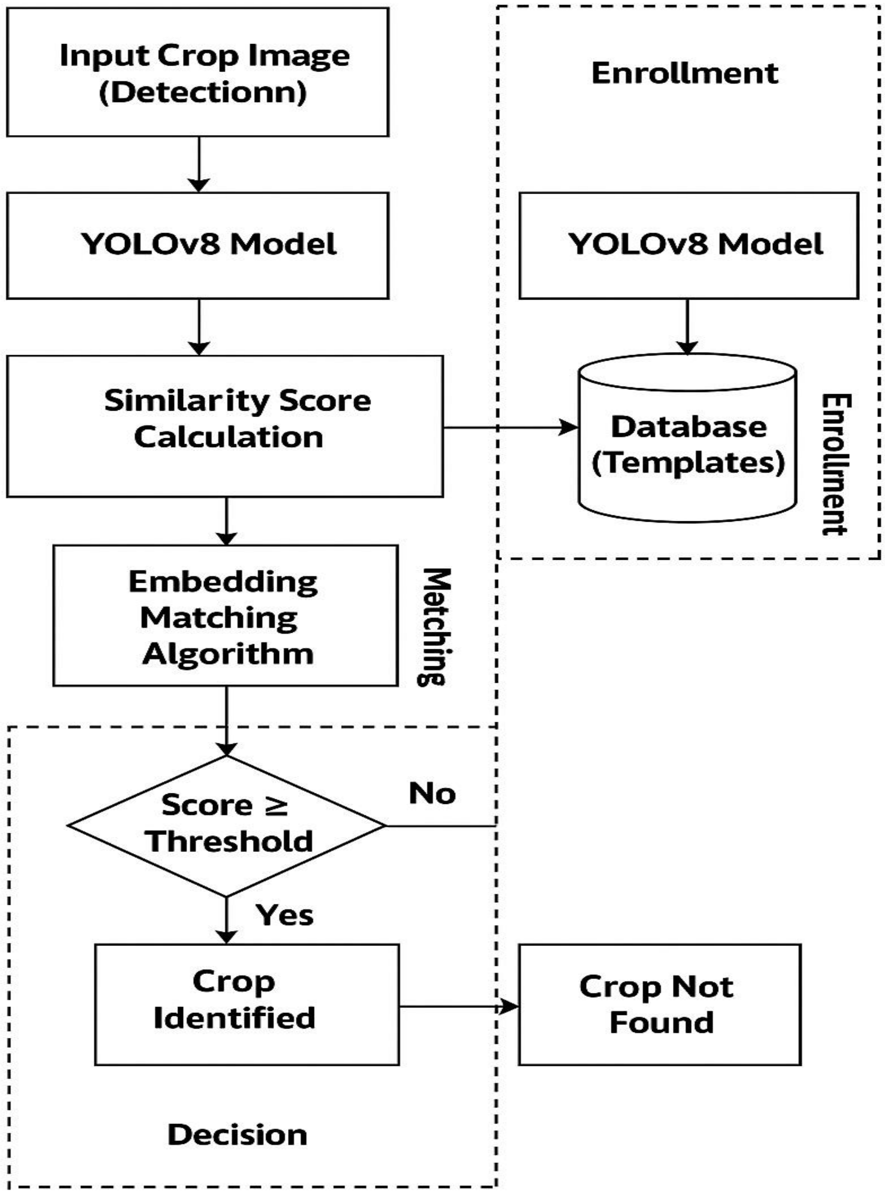


The dataset comprises a certain number of annotated images. The images are of varying resolutions and aspect ratios, covering a wide range of real-world scenarios related to the target application. The annotations include bounding boxes, segmentation masks, and class labels, providing comprehensive information for object localization and recognition tasks.

REAL-WORLD DATASET:

Datasets collected from real-time images provide a valuable resource for training machine learning models to operate effectively in dynamic and real-world environments.

6. OVERVIEW OF METHODOLOGY



Overview Methodologyт Crop Detection

7.Prerequistives for training the model.

7.1 Data preprocessing

The collected data was pre-processed using python function like shutil, hashlib and tqdm image manipulation capabilities, including resizing, cropping, and format conversion, to ensure uniformity and compatibility with machine learning algorithms.

7.2 Data augmentation

To enhance the diversity and robustness of the dataset, data augmentation techniques such as rotation, flipping, and brightness adjustment were applied using image annotation features.

7.3 Data splitting

The dataset was split into labels, validation, and data subsets to enable the evaluation and fine-tuning of machine learning models.

7.4 Training images

These are the images used to train the machine learning model. The model learns from the patterns and features present in these images to make predictions.

7.5 Testing images

The testing set is used to assess the performance of the trained model. It contains images that the model has not seen during training, allowing for an objective evaluation of its predictive accuracy.

7.6 Validation images

The validation set is used to fine-tune the model's hyperparameters and assess its performance during training.

8. RESULTS

The annotation process for the sugarcane classification task was successfully completed. The dataset was organized, annotated, and formatted as per the requirements of deep learning model training. The annotations included class labeling, YAML configuration, and validation file creation. The results of this annotation pipeline are described below:

8.1. Class Label Annotations:

The bounding box coordinates for each image were successfully annotated and stored in the label format. The structure follows the YOLO format:

```
4 0.328000 0.202703 0.322000 0.203829 0.322000 0.206081 0.326000 0.208333 0.330000 0.208333
0.328000 0.202703
4 0.392000 0.191441 0.392000 0.193694 0.402000 0.195946 0.408000 0.194820 0.408000 0.192568
0.402000 0.191441 0.400000 0.188063 0.392000 0.191441
4 0.312000 0.188063 0.312000 0.190315 0.318000 0.189189 0.312000 0.188063
4 0.288000 0.179054 0.286000 0.180180 0.278000 0.180180 0.280000 0.182432 0.288000 0.184685
0.286000 0.186937 0.280000 0.189189 0.290000 0.189189 0.296000 0.190315 0.298000 0.192568
0.304000 0.192568 0.304000 0.181306 0.302000 0.179054 0.288000 0.179054
4 0.332000 0.173423 0.340000 0.176802 0.346000 0.176802 0.344000 0.174550 0.332000 0.173423
4 0.294000 0.157658 0.304000 0.157658 0.300000 0.156532 0.294000 0.157658
4 0.266000 0.138514 0.260000 0.141892 0.266000 0.146396 0.278000 0.147523 0.282000 0.147523
0.282000 0.145270 0.278000 0.143018 0.278000 0.140766 0.274000 0.138514 0.266000 0.138514
4 0.532000 0.034910 0.536000 0.037162 0.536000 0.039414 0.536000 0.046171 0.538000 0.047297
0.536000 0.052928 0.536000 0.050676 0.534000 0.047297 0.516000 0.046171 0.512000 0.047297
0.502000 0.051802 0.498000 0.052928 0.482000 0.052928 0.476000 0.054054 0.472000 0.056306
0.470000 0.059685 0.472000 0.061937 0.474000 0.063063 0.478000 0.064189 0.482000 0.064189
0.488000 0.069820 0.498000 0.072072 0.514000 0.073198 0.538000 0.073198 0.544000 0.075450
0.552000 0.075450 0.572000 0.075450 0.600000 0.074324 0.608000 0.073198 0.622000 0.069820
0.634000 0.069820 0.642000 0.067568 0.650000 0.067568 0.656000 0.066441 0.662000 0.065315
```

Fig 8.1

This format ensures normalized values between 0 and 1 for compatibility with the YOLOv5/YOLOv8 training pipelines.

8.2. **YAML File Configuration:**

A YAML configuration file was generated to define the dataset structure for training and validation. It includes the paths and the class names, as illustrated in Figure 8.2:

```yaml
Validation: Validation.txt
names:
  0: Sugarcane_vegetation
  1: Sugarcane_fullgrowth
  2: Sugarcane_harvesting
  3: sugarcane_flowering
  4: Sugarcane_flowering_rect
path: .
```

Fig 8.2

This configuration file helps the training pipeline to map class IDs to their corresponding names.

8.3 **Validation Image Annotations:**

A Validation.txt file was created to list all the annotated validation image paths. This file enables the model to identify which images belong to the validation set during the training process. Sample entries from the file are shown in Figure 8.3:

```
data/images/Validation/628789_286_4_371_flowering_1706943489.jpg
data/images/Validation/628789_286_5_371_flowering_1706943308.jpg
data/images/Validation/628789_286_7_371_flowering_1706943634.jpg
data/images/Validation/628789_287_1_371_flowering_1706944250.jpg
data/images/Validation/628789_287_2_371_flowering_1706944402.jpg
data/images/Validation/628789_287_3_371_flowering_1706944541.jpg
data/images/Validation/628789_287_4_371_flowering_1706944672.jpg
data/images/Validation/628789_287_5_371_flowering_1706944047.jpg
data/images/Validation/628789_287_6_371_flowering_1706942558.jpg
data/images/Validation/628789_287_8_371_flowering_1706942945.jpg
data/images/Validation/628874_275_15A_371_flowering_1705551362.jpg
data/images/Validation/628876_237_8A_371_flowering_1705140579.jpg
data/images/Validation/628876_237_8B_371_flowering_1705140430.jpg
data/images/Validation/628876_238_6B_371_flowering_1705140122.jpg
data/images/Validation/628876_240_1B2_371_flowering_1705139636.jpg
data/images/Validation/628876_240_2A_371_flowering_1705139467.jpg
data/images/Validation/628876_240_2B_371_flowering_1705139346.jpg
data/images/Validation/628876_243_1_371_flowering_1705138360.jpg
data/images/Validation/628884_162_10C1A_371_flowering_1705661603.jpg
data/images/Validation/628884_162_13_371_flowering_1705661756.jpg
data/images/Validation/628884_76_10A_371_flowering_1706264078.jpg
data/images/Validation/628887_126_10_371_flowering_1705047917.jpg
data/images/Validation/628887_64_1_371_flowering_1706334267.jpg
data/images/Validation/628913_103_16_371_flowering_1706076860.jpg
data/images/Validation/628938_129_2_371_flowering_1706696562.jpg
data/images/Validation/628938_141_13B_371_flowering_1706168238.jpg
```

Fig 8.3

The image filenames carry embedded metadata related to crop stage, field ID, and timestamp, which is crucial for temporal tracking

9. Discussion

9.1 **Unique Solution**

- The proposed system automates **crop stage classification and field monitoring** using advanced **image processing** and **machine learning techniques**.
- It detects crop stages such as **vegetation, full growth, flowering, and harvesting** using **YOLOv8** for high-accuracy detection.It captures muzzle features automatically from a safe distance.
- Real-time detection enables continuous tracking of crop development stages for efficient agricultural planning.

9.2 **Advantages of the Model**

- **High Accuracy:** Accurately detects and classifies crop stages using deep learning models trained on annotated datasets.
- **Scalability**: Easily deployable on small farms to large agricultural fields; compatible with drone and satellite imagery.
- **Contactless Identification**: Minimizes the need for manual field visits, reducing labor costs and ensuring safety in hazardous conditions
- **Efficiency**: Accelerates crop monitoring and decision-making through automated, real-time stage classification.
- **Cost-Effective:** Reduces resource wastage by enabling precise intervention based on detected crop stage.

9.3 **Impact on Livestock Management**

- Supports **early detection of crop stress, diseases, or abnormal growth patterns**, aiding proactive intervention.
- Enhances **smart farming practices, traceability, and transparency** contributing to sustainable agriculture

10.CONCLUSION

- The **crop detection system** developed using the **YOLOv8 model** marks a major advancement in smart agriculture. By leveraging YOLOv8's powerful object detection capabilities, the system accurately identifies various **crop stages** (e.g., seedling, vegetative, flowering, and harvesting) from field images, providing farmers with a **real-time, automated solution** for crop monitoring and decision-making.
- YOLOv8 is particularly well-suited for this task due to its **high detection accuracy**, **fast processing speed**, and **adaptability to diverse environmental conditions** such as lighting, soil type, and field clutter. The system eliminates the need for **manual crop inspection**, thereby reducing labor costs and improving efficiency.
- This solution enables **real-time detection of crop health, growth stage, or anomalies** (e.g., wilting, pest damage), allowing farmers to respond quickly and precisely. By providing accurate crop classification across large farmland using **drones or mobile devices**, it minimizes errors and enhances productivity.
- Moreover, the data collected through this system supports **precision agriculture** by enabling better planning of **irrigation, fertilization, pest control, and harvesting**. Farmers can use the insights to **optimize yield, reduce resource waste**, and improve overall crop quality.
- Incorporating YOLOv8 into crop detection systems is a vital step towards **automated, data-driven, and sustainable agriculture**.

11. REFERENCE:

- Liu, Y., Zeng, F., Diao, H., Zhu, J., Ji, D., Liao, X., & Zhao, Z. (2024). YOLOv8 Model for Weed Detection in Wheat Fields Based on a Visual Converter and Multi-Scale Feature Fusion. *Sensors, 24*(13), 4379.
  This paper presents a YOLOv8-based weed detection system in wheat fields, integrating multi-scale features for improved performance.
- Jianghong Zhao & Jifu Zhao. (2024, September). Crop disease detection based on enhanced YOLOv8. In *Fourth International Conference on Advanced Algorithms and Signal Image Processing (AASIP)*.
  Focuses on adapting YOLOv8 to detect crop diseases, achieving fast, accurate diagnostic capabilities.
- Ahmed, D., Sapkota, R., Churuvija, M., & Karkee, M. (2023). Machine Vision–Based Crop-Load Estimation Using YOLOv8. *arXiv*. arXiv:2304.13282.
  Demonstrates YOLOv8 application for crop-load estimation in apple orchards, combining segmentation and biomass estimation.
- Sonawane, S., & Patil, N. N. (2024). Crop-Weed Segmentation and Classification Using YOLOv8 Approach for Smart Farming. *Journal of Studies in Science and Engineering*.
  Explores YOLOv8 for precise segmentation of crops and weeds, yielding high precision, recall, and mAP.
- Dang Nguyen, K., Phung, T.-H., & Cao, H.-G. (2023). A SAM-based Solution for Hierarchical Panoptic Segmentation of Crops and Weeds. *arXiv*. arXiv:2309.13578.
  Combines SAM with YOLOv8 and DINO for nuanced crop–weed instance segmentation, achieving an impressive PQ+ score of 81.33.
- Gurav, R., Patel, H., Shang, Z., Eldawy, A., Chen, J., Scudiero, E., & Papalexakis, E. (2023). Can SAM recognize crops? Quantifying zero-shot performance of SAM for crop-type mapping using satellite imagery. *arXiv*. arXiv:2311.15138.
  Explores SAM's potential in zero-shot agricultural field segmentation, outlining strengths and limitations.
- Tripathy, P., Baylis, K., Wu, K., Watson, J., & Jiang, R. (2024, July). Investigating the Segment Anything Foundation Model for Mapping Smallholder Agriculture Field Boundaries Without Training Labels. *arXiv*. arXiv:2407.01846.
  Validates SAM for delineating field boundaries in satellite imagery—a key task in precision agriculture.