University of Guelph

Major Project

Course No: ENGG*6100

Course Title: Machine Vision

Project Title: DeepGreen: Weed Detection and Localization via Deep Learning in Agri-Imagery

Date of Handout: 12.03.2025

Submission Deadline: 28.04.2025

Submitted by	Submitted to
Pramit Dutta *ID: 1319376* *MASC (Engg + AI)* *University of Guelph*	**Medhat Moussa** *Professor,* *School of Engineering,* *University of Guelph.*

Table of Contents

List of Figures

List of Figures

Project Title: DeepGreen: Weed Detection and Localization via Deep Learning in Agri-Imagery

1. Introduction

Weed and Crop detection in agricultural fields is a critical task for ensuring crop health, maximizing yield, and reducing herbicide use. The widespread use of herbicides raises significant health and environmental concerns. Herbicide exposure has been linked to both acute and chronic toxicities in humans, including respiratory issues and other long-term health risks [1]. Moreover, environmental contamination of soil and water sources from herbicides can adversely affect non-target organisms, contributing to biodiversity loss [2].

Traditional manual approaches are labor-intensive and time-consuming, often leading to inconsistent results. With the rise of computer vision and deep learning, automated weed detection systems have shown promising results in identifying and classifying plant species in real-time. In Canada, the adoption of precision agriculture technologies has shown measurable benefits. For instance, farmers in Guelph, Ontario, reported up to a 30% increase in crop yields and cost savings of approximately CAD 25–45 per acre using GPS-guided machinery, drone imagery, and crop monitoring systems [3]. Additionally, studies show that weeds can reduce crop yields in USA and Canada by as much as 44% if not effectively managed [4].

Farmers can improve productivity while reducing operational costs by automating weed detection, Computer vision systems can precisely identify weedy areas, enabling targeted herbicide application and minimizing chemical usage [5]. This project develops a Machine Vision Pipeline that includes image augmentation, enhancement

techniques, YOLO-based object detection, and model evaluation. This integrated approach aims to support sustainable, cost-effective, and scalable solutions for real-time weed detection in agricultural fields.

1.1 Problem Description

This project addresses the need for accurate, efficient, and scalable detection of weeds and crops in modern agricultural systems. The task involves not only developing a reliable detection model but also overcoming practical challenges related to variable field conditions and dataset limitations. A robust solution must balance detection accuracy, processing speed, and adaptability for deployment in real-world farm environments. The following subsection outlines the broader context of the problem, highlighting both the agricultural significance and the technical barriers encountered when designing such systems.

1.1.1 Overview of the Problem

Weeds are one of the main causes of crop yield loss, as they compete with crops for vital resources such as nutrients, water, and sunlight. Conventional methods for weed identification and removal, including manual scouting and uniform herbicide spraying, are often inefficient, time-consuming, and environmentally harmful. These approaches also lead to unnecessary chemical usage, increasing costs and posing long-term ecological risks.

Modern agriculture demands automated systems that can distinguish between weeds and crops and enable precise interventions. Object detection methods based on machine vision and deep learning provide promising capabilities for identifying plant species in real time using image data.

Several factors complicate the development of such systems. Real-world conditions often result in images with poor lighting, motion blur, occlusion, and inconsistent backgrounds, all of which degrade detection performance. Agricultural environments also vary significantly in appearance across seasons and locations, adding to the complexity. On top of that, many deployment scenarios involve limited computing resources, which places constraints on model size and processing speed. A dependable solution must therefore perform reliably under diverse and uncontrolled conditions, while remaining efficient enough for practical use in the field.

1.1.2 Dataset Description

This project utilizes the Weed Detection dataset available on Kaggle [6], which comprises a total of 1,176 annotated RGB images with bounding boxes across 8 weed types and 6 crop class representing two distinct classes: weed and crop. The dataset is designed to support object detection and classification tasks in agricultural environments. Each image captures real-world field conditions, featuring varying backgrounds, lighting conditions, and levels of occlusion.

1.1.2.1 Dataset Overview: An overview of image-level class distribution helps clarify how different samples are represented in the dataset. In this project, a dataset is used which includes images where*:*

- Samples containing only crops
- Samples containing only weeds
- Samples with both weeds and crops

The samples demonstrate the diversity in how weed and crop instances appear across different field scenarios. The following image contains three distinct examples of crop, weed and both classes present in a single image:

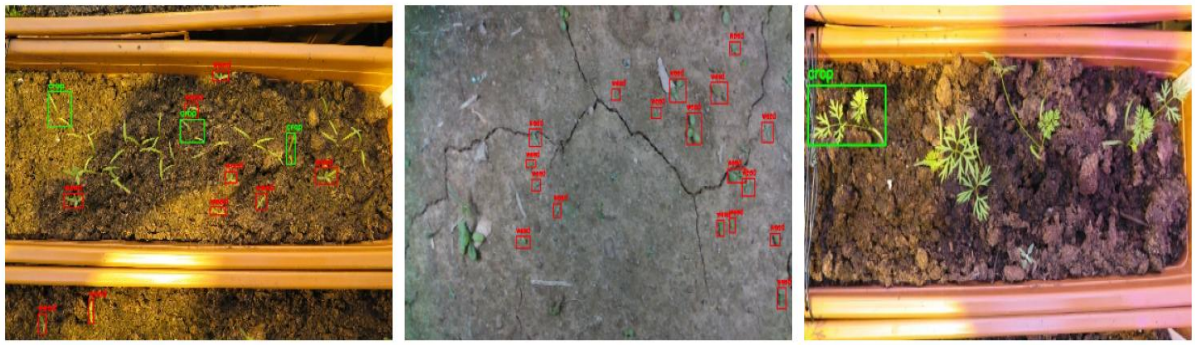


Figure 1: Labeled image samples. (a) Both weed and crop, (b) Only weed, and (c) Only crop.

The images represent annotated examples from the dataset where the image highlights the presence of weed and crop classes using bounding boxes for model training and evaluation. These images may contain either multiple or single instances of the respective classes.

1.1.2.2 Dataset Distribution: The dataset used in this study comprises annotated agricultural images with bounding boxes assigned to two object classes: weed and crop. The distribution of images across the three key categories, images containing only weed, only crop, and both weed and crop is shown in Figure 2.

A total of 1,176 images are analyzed across the combined training, validation, and test sets. Among these, 1,073 images (91.24%) contain only weed instances, 59 images (5.01%) contain both weed and crop, and 44 images (3.74%) contain only crop. This image-level distribution reveals a significant class imbalance, with a heavy bias toward weed-only images.

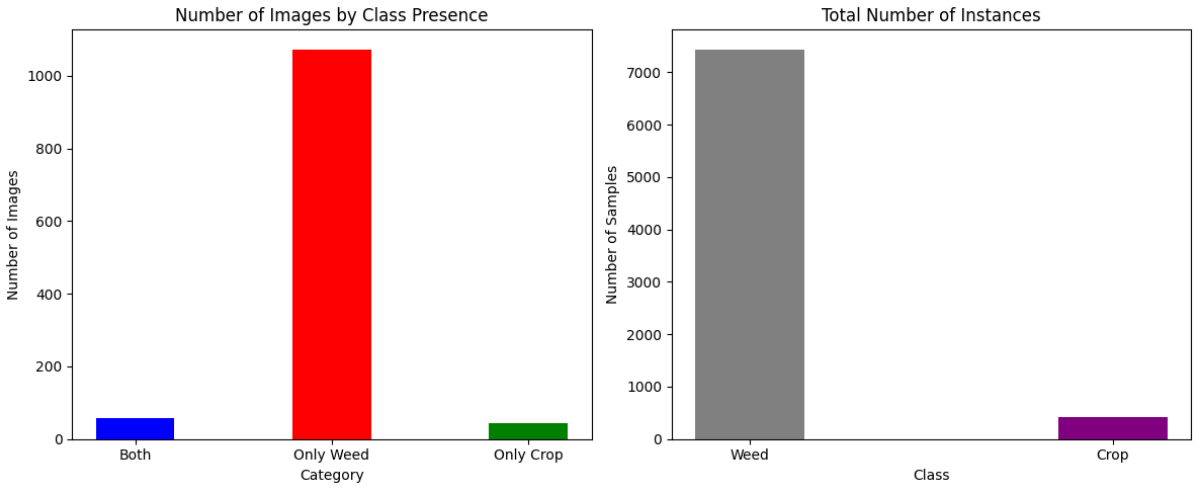


Figure 2: Distribution of class and object instances across the dataset. Number of images containing both weed and crop, only weed, and only crop (left) and total number of labeled object instances for weed and crop classes (right).

The bar plot on the right side of Figure 2 illustrates the total number of object instances (bounding boxes) for each class. A total of 7,442 weed instances are present, compared to only 411 crop instances, reflecting an even more imbalance at the object level. The distribution shows, the dataset has more than 18 times weed objects than crop objects.

1.1.3 Challenges and Considerations

A reliable weed–crop detection model must overcome several critical challenges. This section highlights the main issues affecting model performance, robustness, and deployment in real-world conditions.

1.1.3.1 Class Imbalance: While the previous section highlighted the numerical disparity between weed and crop instances, the implications of this imbalance are critical for model performance and reliability. With over 91% of the images containing only weed and more than 18 times as many weed instances as crop instances, the dataset presents a strong bias toward the majority class. This imbalance

can cause deep learning models to develop skewed representations, leading them to favor detecting weeds while underperforming on crop identification.

Such bias is especially problematic in real-world deployment, where accurate detection of both classes is essential for decision-making and field management. A model trained on this dataset without addressing the imbalance may fail to recognize crop regions or misclassify them as weeds, resulting in inaccurate predictions and potential agricultural loss.

Moreover, the imbalance is present at both the image level and object instance level, further compounding the difficulty. Standard evaluation metrics may mask poor performance on the minority class if not disaggregated, creating a false sense of model accuracy. Addressing this challenge requires the use of strategies such as targeted data augmentation and class-aware evaluation metrics to ensure the model learns balanced representations and performs robustly across both classes.

1.1.3.2 Dataset Diversity and Real-World Complexity: Training a robust weed–crop detection model goes far beyond recognizing plant shapes in images. It requires the ability to interpret visual information across inconsistent, messy, and often unpredictable environments. The dataset used in this study reflects exactly that wide range of real-world field conditions, collected under challenging and varied outdoor settings. These variations are not just superficial differences in appearance; they represent entirely different visual contexts that require different types of understanding from the model.

As illustrated in figure 3, some images are captured in planter boxes where crops appear in organized rows, while others depict irregular open-field conditions with

cracked soil, dense weed clusters, or unexpected elements like synthetic ground covers. In addition to environmental variation, some images suffer from low visual quality due to motion blur, occlusion by other plants or debris, and distortion from focus, further complicating accurate object detection. Environmental factors such as lighting and shadow add another layer of complexity, as they can drastically shift the appearance of the same crop across time and space. As shown in figure 1, even images taken in similar settings display noticeable differences in lighting, which can vary with time of day, weather, or camera exposure. These seemingly minor shifts can significantly impact model performance during inference.

Figure 3: Examples of visual diversity and environmental variability in the dataset. Organized planting tray with uniform spacing (left), naturally cracked open field with scattered weeds (center), presence of synthetic material introducing domain-specific noise (right).

As a result, a model that performs well at one moment may fail in the same field just hours later, even without any flaw in its internal logic. Visual cues can change noticeably due to shifts in lighting, shadows, or other natural factors, making consistent detection much more difficult. This variability presents a serious challenge for real-world deployment, where models must deliver accurate results across a wide range of environmental conditions.

1.1.3.3 Model Efficiency and Feasibility: In addition to accuracy and robustness, the practical deployment of weed and crop detection models depend heavily on their computational feasibility. In real-world agricultural settings, especially in resource-constrained environments, models must be capable of running efficiently on low-power or edge devices without compromising detection quality. High-complexity models may achieve strong performance in controlled experiments but fail to meet runtime or memory constraints during field deployment.

A feasible model must strike a balance between detection accuracy and processing speed, ensuring that it can analyze images quickly and with minimal hardware requirements. This is particularly important when models are integrated into autonomous systems, such ground robots or automated field equipment where real-time decision-making is essential. Therefore, model selection and design must consider not only learning capacity but also inference time, memory footprint, and ease of deployment in diverse agricultural scenarios.

1.1.3.1 Class Definition and Labeling Challenge: The dataset used in this study contains images of 8 weed species and 6 crop types, each with distinct visual characteristics. However, due to the labeling format and the need for simplified model output, the dataset is divided into only two classes: weed and crop. As a result, we were bound to group a wide variety of plant species where each differing in shape, texture, and appearance—into just two categories.

This presents a considerable challenge for the detection algorithm. It must learn to generalize across a diverse range of intra-class variations while still maintaining a clear boundary between the two classes. For instance, some weeds may appear similar

to certain crops at specific growth stages or under varying lighting conditions, making the task of consistent classification more difficult. Therefore, this binary class setup not only simplifies the output space but also tests the model's ability to abstract high-level features from visually complex and overlapping categories.

1.2 Objectives

The goal of this project is to develop a robust machine vision pipeline for weed and crop detection to support precision agriculture. The objectives are:

1. To implement and evaluate an efficient YOLO-based object detection algorithm that accurately classifies and localizes weeds and crops in field images using a dataset formatted in the YOLO annotation style.

2. To enhance the model's performance through class-specific data augmentation and image preprocessing techniques, ensuring improved detection accuracy for both minority and majority classes under variable field conditions.

1.3 Proposed Machine Vision Pipeline

The proposed machine vision pipeline consists of three key stages designed to improve object detection performance on the weed and crop dataset. It begins with class-aware augmentation to address the class imbalance by increasing the representation of underrepresented classes. This step is particularly challenging, as many images contain both weed and crop, making selective augmentation difficult. Additionally, corresponding label files must be adjusted to ensure bounding boxes remain accurate after transformation. An image enhancement stage follows, aimed at improving visual clarity and contrast under varying field conditions. The final and

central component of the pipeline is the YOLOv10 (You Only Look Once) model, which performs object classification and localization. YOLOv10 is chosen for its modern architecture and strong performance in single-stage detection tasks. All preprocessing steps in the pipeline are designed to maximize the effectiveness of YOLOv10 by delivering balanced, and optimized inputs.

1.4 Rationale for Chosen Methods

The proposed machine vision pipeline consists of three stages: class-aware augmentation, image enhancement and performing object detection and classification via YOLOv10.

1.4.1 Rationale for Class-Aware Image Augmentation

Addressing the severe class imbalance in the dataset where weed instances outnumber crop instances by more than 18 to 1, a class-aware augmentation strategy was implemented. This targeted approach increased the representation of underrepresented classes by selectively duplicating and transforming images that contained crops while preserving label's integrity by carefully adjusting the bounding boxes according to the augmentation technique. This method helped the model develop more balanced feature representations and reduced the tendency to overfit to the dominant weed class, resulting in improved and fairer detection performance.

1.4.2 Rationale for Image Enhancement Techniques

In this project, the variability of real-world field conditions ranging from inconsistent lighting to blurred or low-contrast images posed a significant challenge for object detection. Since the model's ability to learn and generalize is only as good as the input data quality, this work followed the core machine learning principle of "garbage in,

garbage out.” A poor-quality image directly hampers feature extraction, especially in complex agricultural scenes where weeds often blend with background textures. The enhancement step follows image augmentation because of that it can also correct visual inconsistencies introduced during augmentation, improving the reliability of feature extraction and overall model performance.

This challenge was addressed using a custom image enhancement pipeline involving CLAHE (Contrast Limited Adaptive Histogram Equalization) on the Value channel in HSV space, followed by a sharpening filter. CLAHE increases local contrast and improves brightness dynamics in shadowed or overexposed regions, making subtle details in foliage more distinguishable. This is especially helpful in field images with poor lighting or heavy background clutter. The sharpening filter enhances edge details around plant structures, improving the visibility of leaf margins, stems, and fine contours, which are crucial for accurate bounding box placement and class differentiation.

Such preprocessing not only improves human interpretability but also significantly enhances the discriminative power of the YOLOv10 detector, which relies on high-frequency spatial features to make confident predictions. Clearer, higher-contrast images across varying field conditions lead to more robust and generalizable model performance.

1.4.3 Rationale for YOLOv10

Given the constraints of real-time field deployment and the need for high detection precision, YOLOv10 was selected as the object detection backbone for its ability to combine accuracy with computational efficiency. Unlike two-stage detectors such as

Faster R-CNN, which separate region proposal and classification steps and often suffer from latency overhead, YOLOv10 performs single-stage detection, enabling simultaneous prediction of bounding boxes and class labels with minimal delay.

YOLOv10 is particularly suitable for weed and crop detection as in this application small, overlapping, and visually similar plant structures must be differentiated under challenging conditions. The architecture's ability to retain spatial resolution through advanced feature aggregation such as its use of Efficient Layer Aggregation Networks (ELAN) enhances its sensitivity to fine-grained plant features like leaf edges and stem contours. This granularity is critical for distinguishing between weed and crop instances that often share similar shapes or textures.

Additionally, YOLOv10 incorporates dynamic label assignment strategies that improve detection in scenes with ambiguous object boundaries which is common in mixed-field scenarios. Its strength in capturing complex spatial relationships helps reduce misclassification between closely spaced weed and crop regions, ultimately leading to more robust and accurate detection performance.

These methods collectively create a reliable system for detecting weeds and crops under real-world agricultural conditions. This approach supports accurate model testing and lays the groundwork for real-time use in precision farming.

2. Literature Review

Accurate weed detection is important in modern farming, leading to the need for smart and reliable methods that work well in real field conditions. Many studies have

focused on this topic, using different approaches to improve detection in changing environments.

R. Goyal et al. [4] explore the use of deep learning for weed detection in complex and heavily occluded potato fields, addressing a practical challenge that has seen limited attention in earlier studies. They introduce a new image dataset collected from Indian farms and apply both Mask RCNN and YOLOv8 models to distinguish between crops and weeds. While YOLOv8 shows slightly better overall detection performance, Mask RCNN proves more effective at accurately identifying weeds, making it especially useful for targeted weed management. This study contributes significantly to post-emergence weed detection by demonstrating how advanced computer vision techniques can operate reliably in real-world agricultural conditions.

N. Islam et al. [7] investigate the potential of traditional machine learning algorithms for early weed detection in chili farms using UAV-acquired RGB images. Their study applies image processing techniques to extract features like normalized color bands and vegetation indices, followed by classification using random forest (RF), support vector machine (SVM), and k-nearest neighbors (KNN). The authors demonstrate that RF and SVM significantly outperform KNN, particularly in terms of precision and reliability, suggesting their practical suitability for real-world agricultural applications. The research focused on early-stage detection when crops are most vulnerable to weed competition. This work contributes to cost-effective and timely weed management strategies in precision farming.

S. K. Valicharla et al. [8] applied Mask R-CNN for weed recognition in agriculture, focusing on both ground-level and aerial imagery. Addressing dataset limitations, they

developed a synthetic dataset comprising 80 weed classes and employed Detectron2 for model training and evaluation. Their experiments on the adjusted synthetic dataset (2-class setup) achieved an average precision (AP) of 50.03% and AP@50 of 75.95%, corresponding to an estimated F1 score of approximately 0.72. The study further explored localized neural style transfer and geometric transformations to augment UAV imagery, underscoring the practical challenges of real-world weed identification and highlighting the potential of deep learning in advancing precision agriculture.

A. N. V. Sivakumar et al. [9] conducted a comparative study on deep learning models for detecting mid- to late-season weeds in soybean fields using UAV imagery. The research evaluated the performance of object detection models Faster RCNN and Single Shot Detector (SSD) and a patch-based CNN approach. The models were assessed based on detection metrics such as precision, recall, F1 score, and mean Intersection over Union (IoU). Among the models, Faster RCNN achieved an F1 score of 0.66, demonstrating better generalization and confidence in detections compared to SSD, which showed an F1 score of 0.67 but required a lower confidence threshold (0.1). Additionally, Faster RCNN outperformed patch based CNNs in both accuracy and inference time. This study underscores the suitability of object detection frameworks like Faster RCNN for near real-time, on-farm weed detection applications using UAVs.

J. Lekha et al. [10] developed an enhanced weed detection system combining YOLOv7 with Internet of Things (IoT) sensors to improve precision agriculture outcomes. Acknowledging the limitations of prior YOLO models, their study demonstrated YOLOv7's superior detection accuracy and real-time performance for

identifying weeds and crops in both image and video datasets. The authors collected synchronized sensor data (e.g., soil moisture, temperature, humidity, pH, and light intensity) alongside annotated visual data to inform weed growth patterns. The model was trained on a labeled dataset of sesame crops and evaluated using both real-world video footage and benchmark image datasets. Through this multimodal approach, the system achieved a macro F1 score of 0.78, indicating strong performance in detecting and classifying weeds under dynamic field conditions. This study highlights the potential of integrating sensor analytics and deep learning for sustainable, real-time weed management.

I. Matvienko et al. [11] introduced a novel Bayesian aggregation approach to improve crop classification from single satellite images. Recognizing the limitations of multi-temporal data due to cloud cover and seasonal constraints, they focused on single-image classification using Sentinel-2 data. Their study compared classical machine learning methods (Random Forest, Gradient Boosting, KNN) and a U-Net model with SE-blocks for pixel-wise crop classification, and tackled class imbalance using resampling and weighted loss techniques. They proposed Bayesian aggregation for converting pixel-level predictions into field-level classifications. This method outperformed traditional majority and averaging strategies, achieving the best field-wise accuracy of 77.4% and a macro F1-score of 0.66, thus demonstrating its potential in precision agriculture applications with limited data availability.

P. De Marinis et al. [12] proposed RoWeeder, an unsupervised weed mapping framework that integrates crop-row detection with a noise-resilient deep learning model. Acknowledging the challenges of extensive manual labeling in agricultural

datasets, they introduced a method that generates pseudo-ground truth by identifying linear crop-row patterns using the Hough Transform. This information is used to train a lightweight SegFormer-based segmentation model for distinguishing crops from weeds in UAV-captured multispectral images. Their study compared different decoder architectures (pyramid vs flat) and fusion methods to optimize segmentation performance. Despite training on noisy labels, the model achieved a F1-score of 75.3%, outperforming several baselines. The results emphasize RoWeeder's potential for real-time, annotation-free weed detection in precision agriculture.

X. Jin et al. [13] introduce a novel weed identification approach tailored for vegetable plantations where traditional crop-focused detection methods often fail due to random plant spacing and diverse weed species. Instead of directly detecting weeds, their method first uses a CenterNet based deep learning model to identify and segment vegetables. Any green regions outside the detected vegetable bounding boxes are then classified as weeds using a custom designed color index optimized via genetic algorithms. This indirect strategy significantly reduces training complexity and enhances adaptability, particularly in unstructured field environments. The results demonstrate high accuracy and robustness under varied lighting, backgrounds, and growth stages, offering a promising solution for intelligent robotic weeding in sustainable agriculture.

N. Razfar et al. [14] present a lightweight, vision-based deep learning approach for weed detection in soybean crops, emphasizing practical deployment in precision farming. Their work evaluates standard models like MobileNetV2 and ResNet50 alongside three custom CNN architectures, all trained and tested on a dataset of over

15,000 image segments. Notably, the custom 5-layer CNN outperformed other models in accuracy, latency, and memory efficiency, making it well-suited for edge deployment on devices like Raspberry Pi. This research prioritizes low computational overhead and high inference speed, this study addresses real-world constraints in agricultural automation and demonstrates how tailored CNN architectures can optimize weed detection in resource-constrained environments.

M. A. Haq et al. [15] introduce a CNN-LVQ-based automated weed detection framework using UAV imagery for soybean fields. Their method combines the feature extraction power of deep convolutional networks with the adaptive learning capability of Learning Vector Quantization to classify weeds into categories such as broadleaf and grass while also distinguishing them from crops and soil. The system is trained on a diverse and segmented UAV image dataset and is rigorously optimized through hyperparameter tuning. The results demonstrate strong classification performance across all categories especially in challenging cases where weeds and crops have similar appearances. This model stands out for its robustness and adaptability which offers a scalable solution for intelligent and UAV-enabled precision agriculture. It also highlights the potential of combining classic vector quantization techniques with modern deep learning to enhance real-world decision-making in field conditions.

3. Methodology

This section outlines the complete methodology used to develop and train the weed and crop detection model based on YOLOv10. The process begins with dataset preparation and concludes with evaluation on unseen data.

3.1 Overview of the Implementation

Figure 4 presents a high-level schematic of the workflow. It captures the flow from raw dataset processing through augmentation, training, and evaluation.

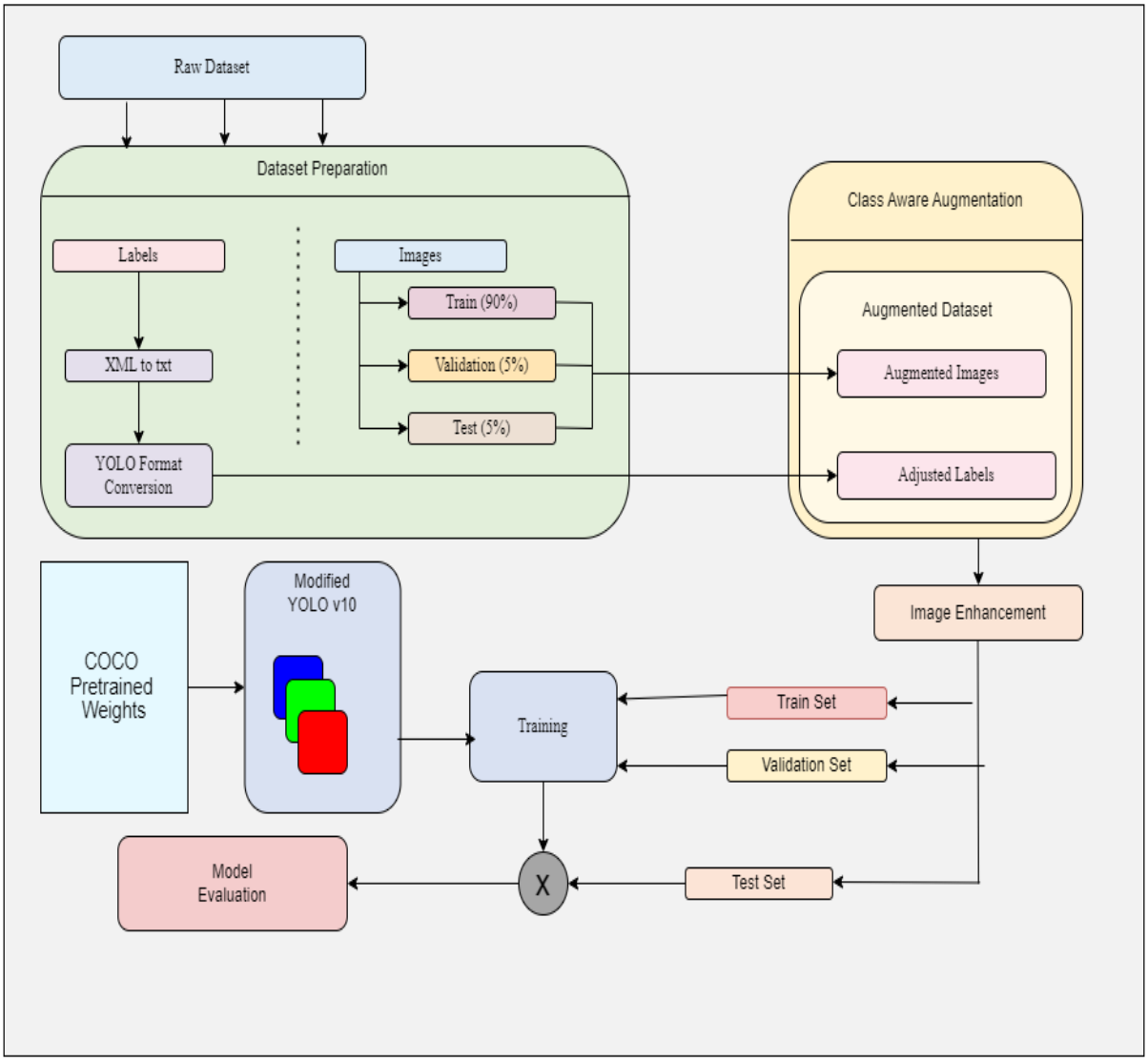


Figure 4: Overview of the YOLOv10 training and evaluation pipeline with class-aware augmentation and preprocessing.

Each step is modular and designed to be adaptable to various dataset conditions. The pipeline includes dataset preparation, class-aware data augmentation, image enhancement, model training, and evaluation. The workflow has been designed to address class imbalance and improve detection robustness through targeted preprocessing and augmentation strategies.

3.2 Proposed Algorithm

The workflow illustrated above is implemented through a structured algorithm that guides the process from dataset ingestion to model deployment. Each stage in the pipeline is designed to address class imbalance, improve data quality, and maximize detection performance through carefully sequenced operations.

Step 1: Dataset Initialization: Load raw dataset containing images and annotations. Apply label conversion to unify the format.

Step 2: Dataset Splitting: Split the dataset into training, validation, and test sets based on predefined ratios.

Step 3: Class-Aware Augmentation: Apply different augmentation intensities per category to balance class representation. Update corresponding labels.

Step 4: Image Preprocessing: Apply enhancement techniques to improve visual quality.

Step 5: Model Configuration: Modified YOLOv10 model and for weed–crop detection.

Step 6: Model Train and Evaluation: Initialized with COCO-pretrained weights, then evaluated on test set after training.

3.3 Dataset Initialization

In the first step of the pipeline, the dataset annotated in XML format was converted into the format required by the YOLO object detection framework. YOLO expects each object to be labeled with a class ID followed by four normalized values: the x- and y-coordinates of the bounding box center, and the width and height of the box where all expressed as fractions of the image dimensions. Each XML annotation was

parsed to extract object labels and bounding box coordinates. The bounding box was originally defined by absolute pixel values x_{min} and x_{max} (the horizontal positions of the top-left and bottom-right corners) and y_{min} and y_{max} (the vertical positions). These values were converted into a normalized format using the following equations, where image width and height refer to the total dimensions of the image:

$$x_{center} = \frac{x_{min} + x_{max}}{2 \times image\ width} \quad (3.1)$$

$$y_{center} = \frac{y_{min} + y_{max}}{2 \times image\ height} \quad (3.2)$$

$$width = \frac{x_{max} - x_{min}}{2 \times image\ width} \quad (3.3)$$

$$height = \frac{y_{max} - x_{min}}{2 \times image\ height} \quad (3.4)$$

This normalization ensures that the bounding boxes are consistent across different image resolutions and prepares the data for effective training in YOLO. For this project, object classes were mapped as follows: weed was assigned class ID 0, and crop was assigned class ID 1. Any objects not defined in this mapping were excluded to maintain dataset integrity and consistency.

3.4 Dataset Splitting

The dataset was divided into three subsets: training, validation, and test. This split ensures proper model generalization and unbiased performance evaluation. A balanced strategy was used to maintain equal class distribution across all subsets, addressing the uneven presence of crop and weed classes. Figure 5 presents a detailed

visual summary of the dataset composition across the training, validation, and test splits.

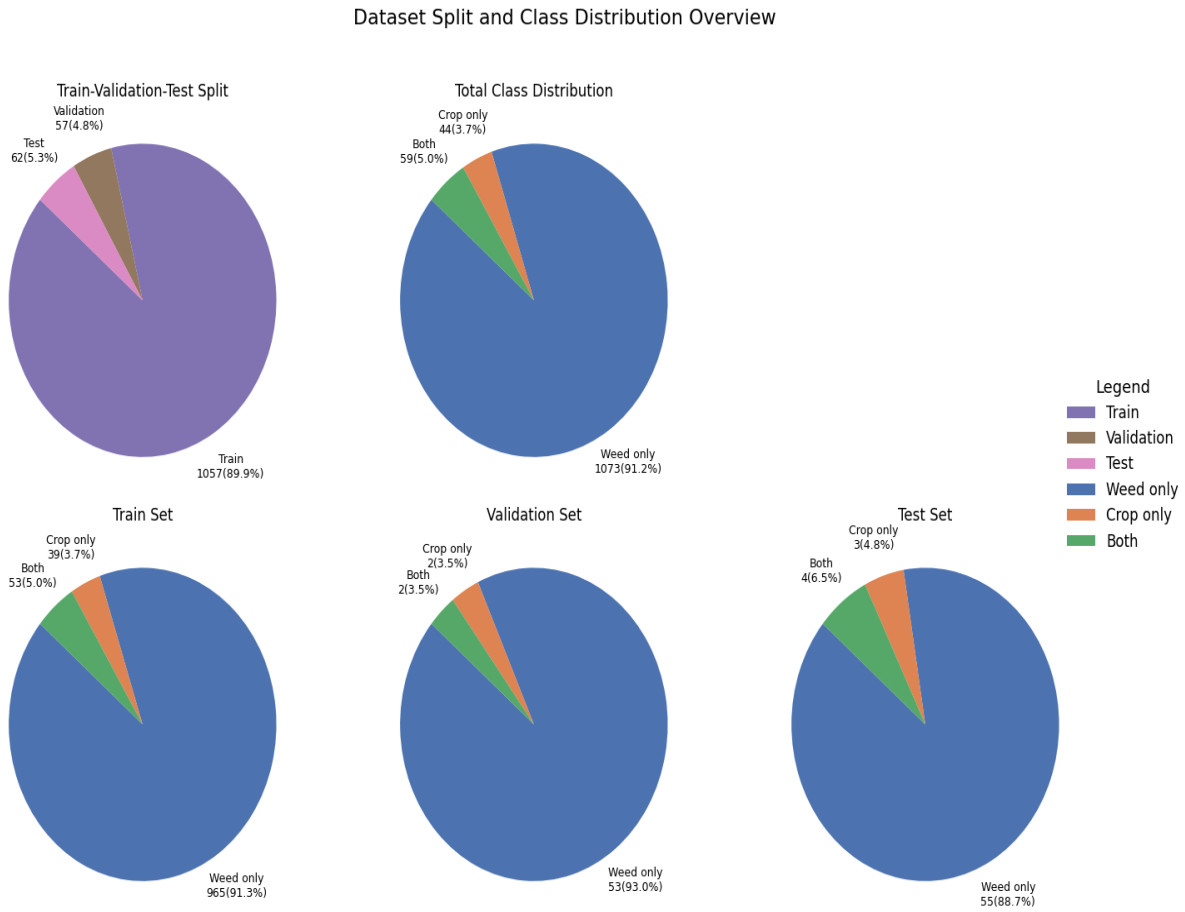


Figure 5: Class and split-wise distribution of the dataset. Top: dataset split and overall class distribution. Bottom: label breakdown in train, validation, and test sets with counts and percentages.

The top-left pie chart shows the number of images allocated to each subset: 1,057 images in the training set, 57 in the validation set, and 62 in the test set. The top-right chart displays the overall label distribution, categorized into three classes: weed only (images containing only weed samples), crop only (images containing only crop samples), and both (images that contain both weed and crop samples). The bottom row includes individual pie charts for each dataset subset. These charts illustrate the internal class composition of the training, validation, and test sets. Specifically, the training set includes 965 weed-only images, 39 crop-only images, and 53 images

containing both classes. The validation set includes 53 weed-only, 2 crop-only, and 2 both-class images, while the test set comprises 55 weed-only, 3 crop-only, and 4 both-class images. Each pie chart is annotated with the number of images and their corresponding percentages that represent the relative proportions of each class within the respective subset. Finally, a yaml configuration file was created to ensure reproducibility and standardized training, defining dataset paths and class names according to YOLOv10 formatting conventions.

3.5 Class-Aware Augmentation

A class-aware augmentation strategy was designed to improve generalization and address class imbalance in the dataset. Augmentations were not applied uniformly across the dataset; instead, transformations were selectively used based on the class composition of each image—weed only, crop only, or both. This targeted approach ensures that underrepresented combinations are synthetically enriched during training, leading to more balanced learning across categories.

In addition to balancing the dataset, the augmentation strategy also captures realistic variability that can occur in field conditions—such as shifts in lighting, orientation, or partial plant occlusion that helps the model become more robust to practical deployment scenarios. Augmentation levels were carefully adjusted to match each image type, and label annotations were scaled accordingly to preserve spatial accuracy.

3.5.1 Augmentation Techniques

The figure-6 presents a visual overview of eight augmentation techniques applied individually to a representative image. This demonstration allows for a direct

comparison of how each transformation affects the spatial and color characteristics of the image.

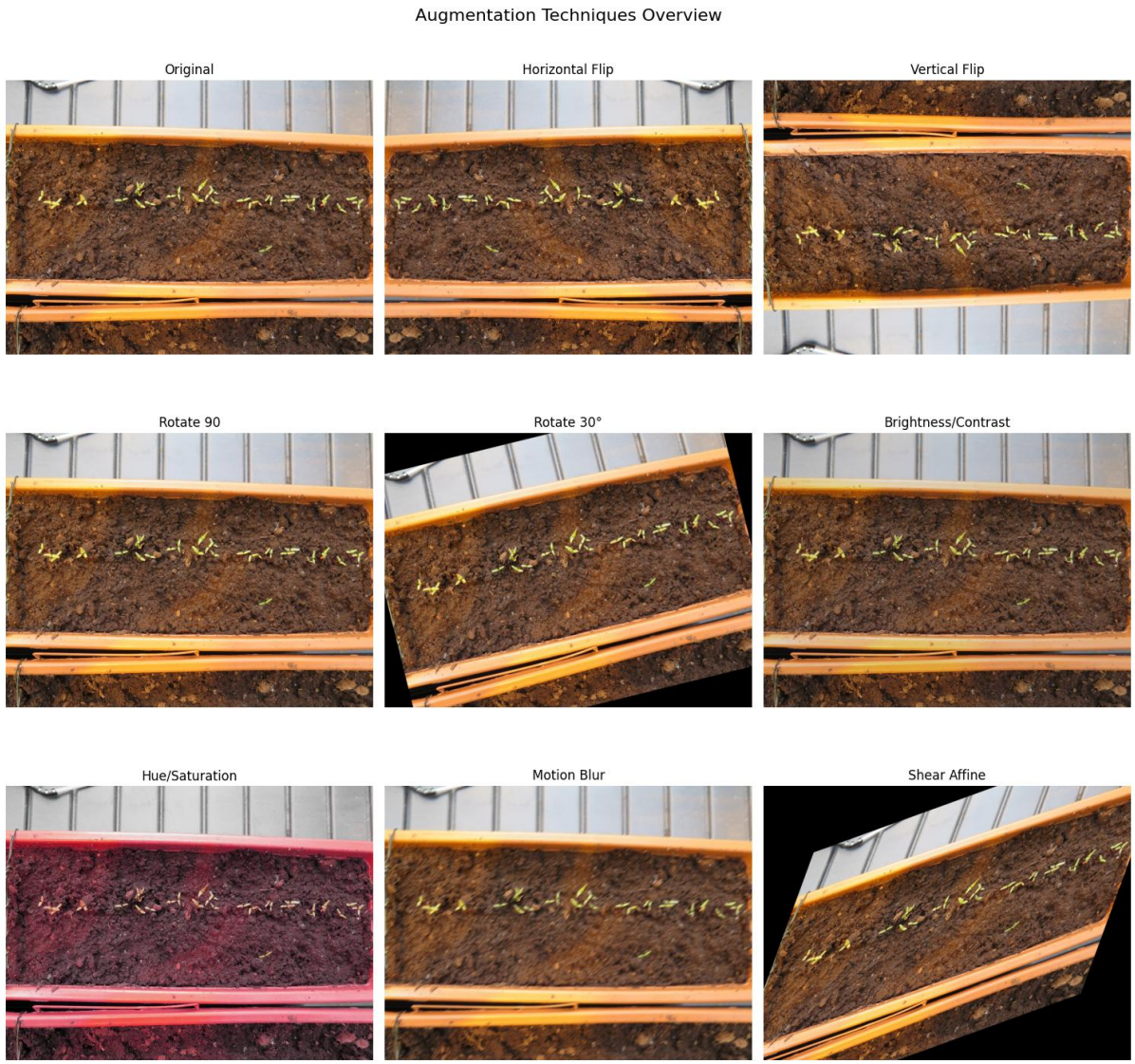


Figure 6: Image Augmentation Techniques. Transformations include flips, rotations, brightness/contrast, hue shifts, blur, and shear to enhance training diversity.

Techniques shown include flipping, rotation, brightness and contrast adjustments, hue and saturation shifts, motion blur, and affine shear. These transformations reflect a variety of real-world conditions that might be encountered in field environments, such as orientation changes, lighting variability, and camera-induced motion effects. Although each transformation here is applied in isolation for illustrative purposes, the

actual implementation within the augmentation pipeline employs probabilistic application. Most transformations, including flips, rotations, and color adjustments, are applied with a probability of 0.5 (1 in 2). Motion blur and affine shear, which can more aggressively distort image structure, are applied with slightly lower probabilities of 0.2 (1 in 5) and 0.3 (3 in 10) respectively. This configuration balances diversity in augmented samples while maintaining the semantic integrity of the original image content during training.

3.5.2 Label Adjustment

Following data augmentation, it is critical to ensure that object annotations remain spatially consistent with the transformed image content. This step, referred to as label adjustment, involves the recalibration of bounding box coordinates to align accurately with the augmented positions of the annotated objects. Figure 7 illustrates this alignment where the transformed annotations clearly align with the new object positions, demonstrating the robustness of the label adjustment process.

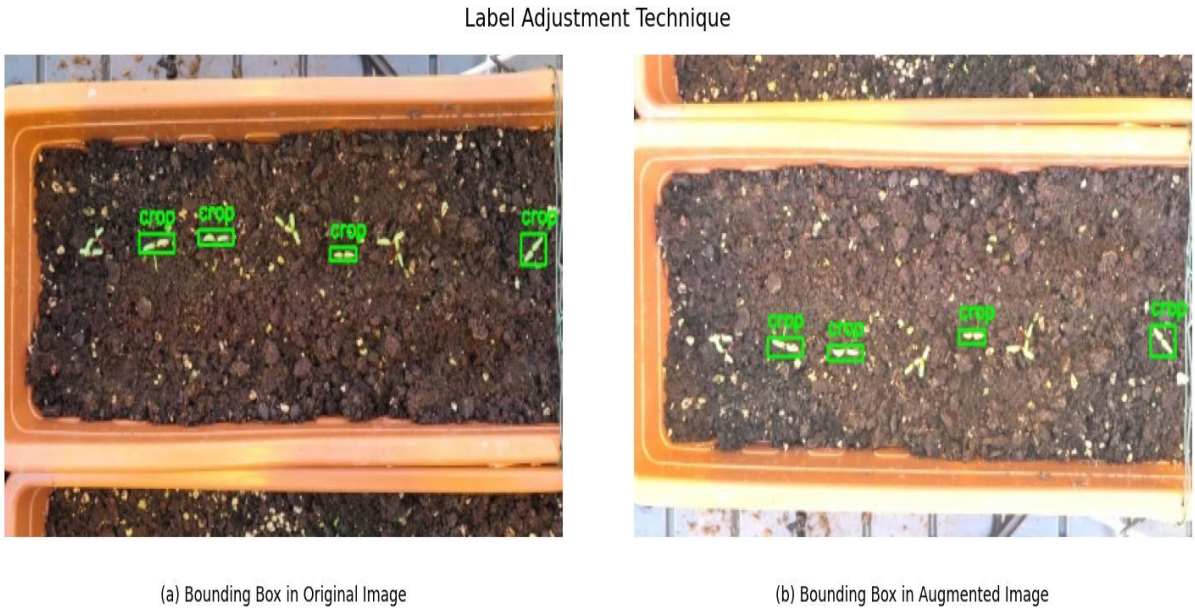


Figure 7: Label Adjustment after Augmentation. The bounding boxes in the original image (a) and the corresponding bounding boxes after augmentation (b).

This step is essential for maintaining annotation accuracy and ensuring reliable model training, particularly in tasks involving small or closely spaced objects such as early-stage crop detection. In this study, bounding boxes were defined in the YOLO format, and transformations were applied using a pipeline that preserved the integrity of object labels during geometric and photometric augmentations. The adjustment process ensured that the relative positions, sizes, and associations of all labeled objects remained consistent with the original semantic meaning post-augmentation.

3.5.3 Augmentation Strategy for Class Distribution Equalization

The original dataset exhibited a significant class imbalance, with the crop class being heavily underrepresented compared to weed-class instances. This imbalance was most pronounced in the training split, which contained only 39 crop-only images out of a total of 1,057. A class-specific augmentation strategy was implemented to address this limitation and prevent biased learning. Figure 8 illustrates the resulting image distribution before and after augmentation.

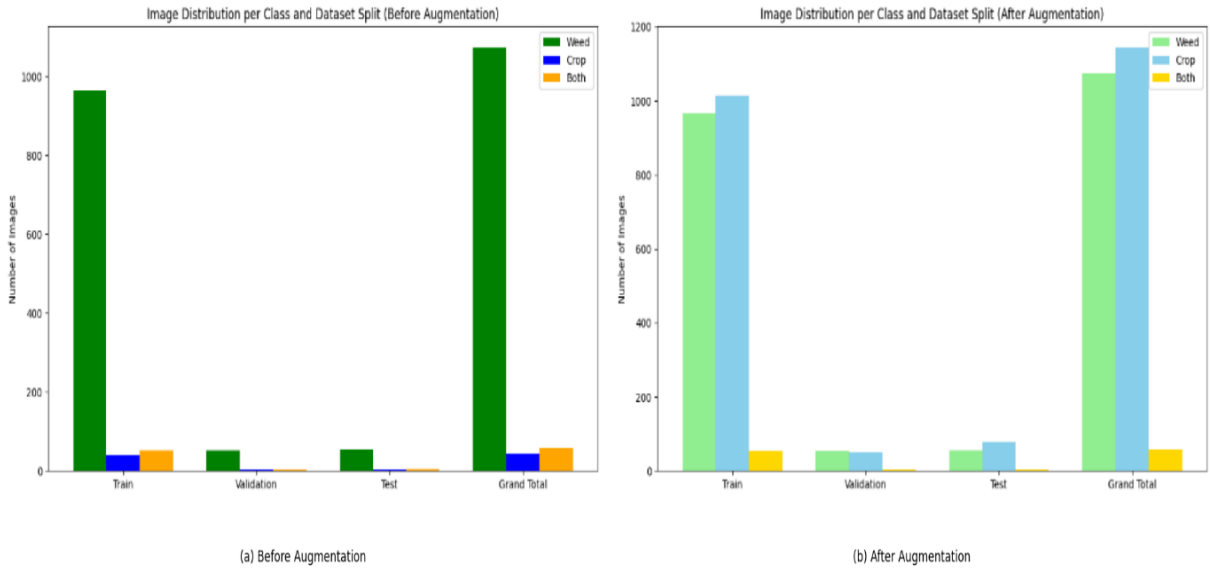


Figure 8: Comparison of image distribution across dataset splits before augmentation (a) and after augmentation (b).

A 26-fold augmentation was applied exclusively to crop-only images, increasing their count from 44 to 1,144 in the full dataset. Notably, the number of weed and both-class images remained unchanged to preserve the original data distribution for those classes. This targeted approach led to a more balanced representation across all dataset splits, particularly in the training set where crop images increased from 39 to 1,014. This augmentation strategy played a critical role in improving the dataset's diversity and balancing the class representation. The model was exposed to a broader variety of crop appearances during training by amplifying the presence of the crop class without altering the relative proportion of weed and both-class instances. This adjustment is expected to enhance generalization and reduce bias toward the majority class.

3.6 Image Preprocessing

This uniform enhancement process involved CLAHE (Contrast Limited Adaptive Histogram Equalization) on the Value (V) channel of the HSV color space, followed by a sharpening filter. These steps were applied after data augmentation, which often introduces visual inconsistencies such as uneven lighting or reduced sharpness especially in synthetic transformations like flipping, rotation, or exposure adjustment. Applying preprocessing after augmentation helps reduce visual distortions introduced during augmentation, resulting in clearer and more feature-rich inputs for model training. This step aligns with the core machine learning principle of "garbage in, garbage out" which means poor-quality input images can significantly impair the model's ability to extract meaningful features and make accurate predictions.

Figure 9 illustrates the effect of the image processing pipeline applied consistently to all images in this study after augmentation. The top row presents the images as they

appear before preprocessing, while the bottom row shows the same images following the enhancement step. This visualization highlights how the applied processing techniques refine image quality by improving contrast, sharpness, and overall feature clarity—critical factors for effective object detection in complex agricultural scenes.

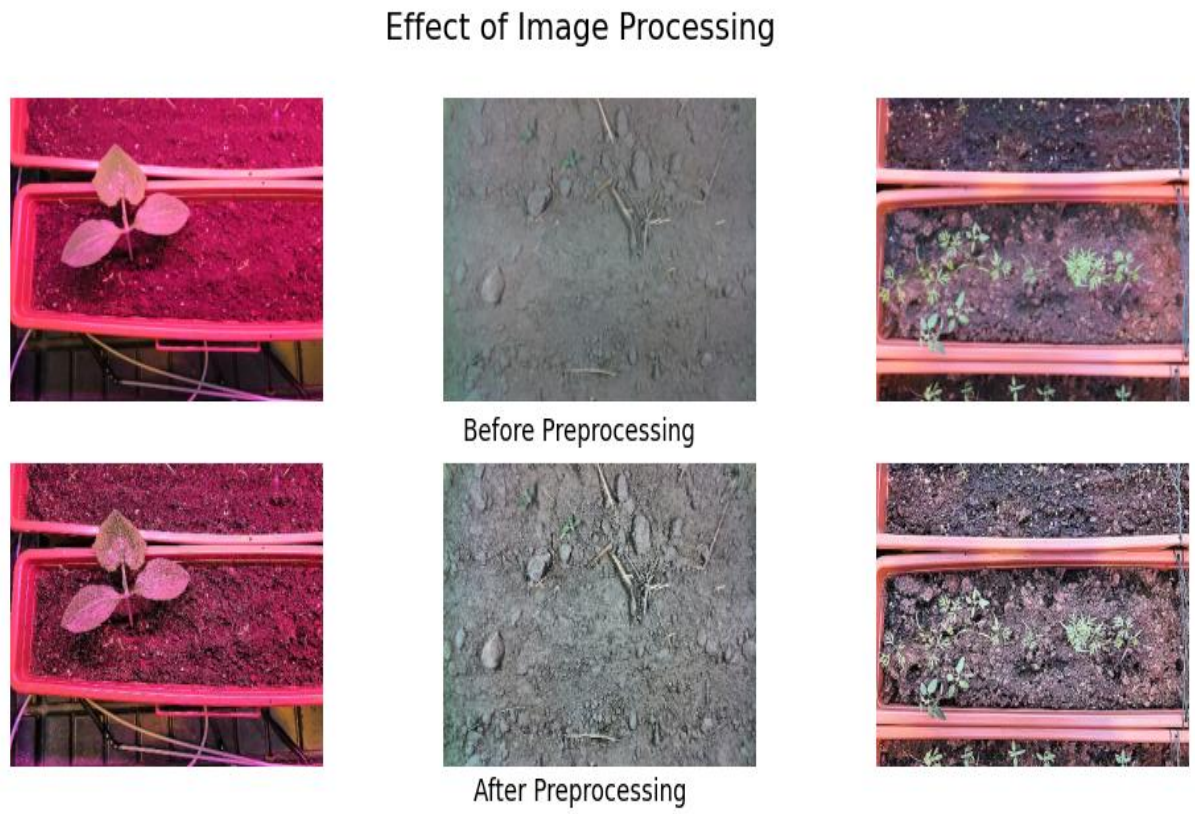


Figure 9: Effect of image preprocessing applied after augmentation. The top row shows images before preprocessing, while the bottom row displays the same images after enhancement using CLAHE and sharpening.

The three columns in Figure 9 depict varied agricultural scenes with distinct textures, lighting conditions, and crop arrangements. In the first column, the contours and vein structures of the plant leaves become more pronounced after preprocessing, which helps in highlighting fine features that are otherwise muted under the strong artificial lighting. The second column shows an open soil field where preprocessing enhances both the color uniformity and the texture of the soil, making the background more visually consistent and easier to distinguish from small weeds or plant roots. In the

third column, the original image suffers from uneven lighting and dull contrast, causing the soil texture and plant outlines to appear washed out. After preprocessing, these elements are significantly more defined, contributing to better visual clarity and more reliable feature extraction for downstream detection tasks. These enhancements are particularly valuable for the YOLOv10 model, which depends heavily on sharp spatial features to localize and classify objects accurately.

3.7 Model Configuration

This study employs a customized version of the YOLOv10m architecture [16], modified for enhanced efficiency and generalization. The architecture, referred to here as the modified YOLO model, follows the core principles of the "You Only Look Once" (YOLO) detection paradigm: fast, end-to-end object detection in a single network pass, without the need for region proposals or multi-stage pipelines. While the base model is built upon the YOLOv10m design, several targeted modifications especially the integration of Ghost Convolution layers have resulted in improved computational efficiency and stronger performance in challenging detection environments.

3.7.1 Model Overview

The architecture processes an input image through a unified feature extraction and prediction pipeline. It generates three primary output features P3, P4, and P5 corresponding to increasing receptive fields (i.e., for small, medium, and large objects). These outputs are then routed through a series of upsampling and downsampling blocks that align the spatial resolutions, enabling effective fusion of semantic and spatial features.

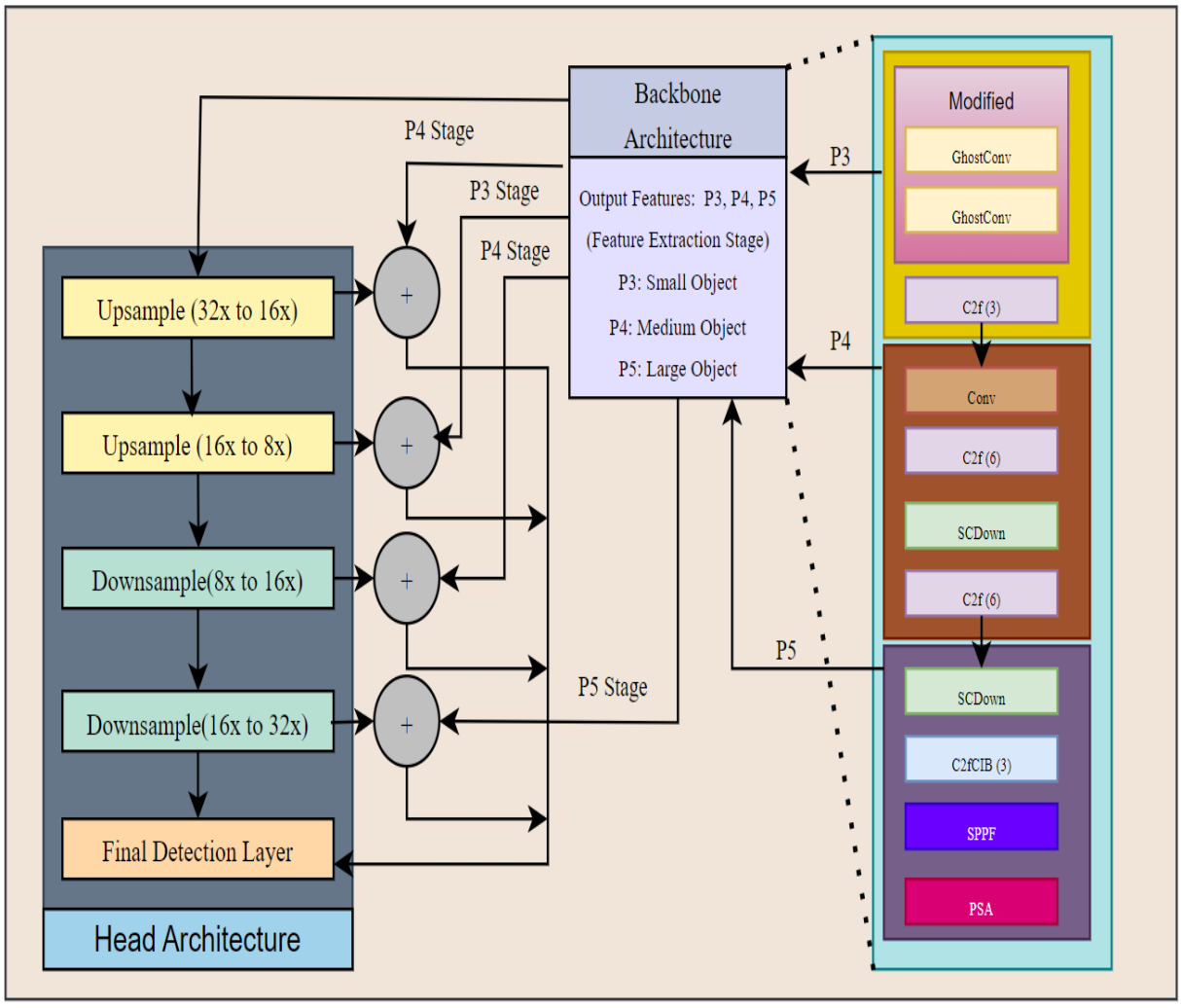


Figure 10: Diagram of the modified YOLO model showing backbone feature extraction, feature fusion pathways, and final detection output.

The design philosophy is centered around maximizing information reuse across scales while minimizing computational redundancy. Each component of the model plays a specific role in building a progressively richer representation of the input scene, ultimately resulting in a robust final detection layer that performs classification and localization in a single pass.

3.7.2 Backbone Architecture

The backbone architecture is responsible for extracting deep hierarchical features from the input image. It processes the image through successive stages, each composed of

carefully designed convolutional and attention modules generates multiscale feature maps (P3, P4, and P5). These stages represent different downsampling levels: 8× (P3), 16× (P4), and 32× (P5) reductions in spatial resolution from the input image. The key components include:

3.7.2.1 Ghost Convolution Layers: For our algorithm, the ghost convolution layer replaced traditional convolution layers at the early stages. GhostConv generates a smaller number of intrinsic feature maps through conventional convolutions, and then expands them into a richer set of features using inexpensive linear operations (e.g., depthwise convolutions). This drastically reduces computational overhead without compromising representational power.

3.7.2.2 C2f and C2fCIB Modules: These structures improve channel mixing and feature reusability. The C2f module acts like a residual block with multiple convolutional branches, enhancing feature diversity. The C2fCIB variant further integrates channel attention to selectively emphasize useful features.

3.7.2.3 SCDown Layers: SCDown learns to retain critical features during spatial compression, helping preserve information that would otherwise be lost.

3.7.2.4 SPPF (Spatial Pyramid Pooling - Fast): Aggregates spatial context at multiple scales efficiently, aiding in accurate localization, especially for small and irregularly shaped objects.

3.7.2.5 PSA (Polarized Self-Attention): Introduced at the deepest stage (P5), PSA enhances long-range feature dependency, allowing the model to detect occluded or ambiguous objects more effectively.

3.7.3 Head Architecture

The head architecture is responsible for refining and aligning features extracted by the backbone. It plays a crucial role in enabling scale-aware detection by ensuring that features from P3, P4, and P5 are made compatible in resolution and information content before final predictions. This architecture includes:

3.7.3.1 Upsampling Blocks: Used to increase the spatial resolution of deeper (lower-resolution) feature maps such as those from P4 and P5. Nearest-neighbor interpolation is employed to minimize computational complexity while preserving essential patterns.

3.7.3.2 Downsampling Blocks: Used to reduce the resolution of shallower feature maps (e.g., P3) when combining them with deeper ones. This ensures alignment across spatial dimensions and preserves semantic context.

3.7.3.3 Feature Addition: Features from different resolutions are merged using element-wise addition and concatenation. These additive connections help bridge the semantic gap between shallow and deep layers, improving gradient flow and robustness.

3.7.3.4 Final Prediction Layer: The final prediction layer, aligned with the YOLO framework, outputs bounding boxes and class probabilities directly from the fused multiscale features. This layer leverages the aligned P3, P4, and P5 representations to simultaneously detect wide range of crop and weed.

This end-to-end configuration in a single unified pass, combining classification and localization for all scales ensures:

- High-speed inference, suitable for real-time applications.
- Compact model size, enabling deployment on edge devices.
- High accuracy, especially in scenes with object scale variation.

3.8 Model Train and Evaluation

The modified YOLOv10m model was initialized using COCO-pretrained weights, which provide a robust foundation for transfer learning by leveraging knowledge from over 118,000 annotated images spanning 80 object categories. This initialization accelerates convergence and enhances generalization, especially beneficial for tasks with limited domain-specific data. The model was trained on the augmented weed–crop dataset using standard optimization settings and validated with a reserved validation set to monitor performance. Following training, the model was evaluated on an unseen test set to assess its real-world detection capability in diverse agricultural environments.

4. Experimentation

This section outlines the experimental setup adopted for model development and evaluation. It details the training configuration, computing environment, and the evaluation metrics used to assess model performance.

4.1 Training Setup

The model training was conducted using the default loss functions provided by the YOLO framework, which include three key components: bounding box regression loss, classification loss and distribution focal loss. These losses were jointly optimized to ensure precise object localization, accurate class prediction, and improved bounding box quality. The combination of these loss functions is well-suited for object detection

tasks where both localization and classification accuracy are critical. A linear learning rate scheduler were incorporated as optimization techniques to enhance training efficiency and model convergence. A summary of the key hyperparameters used during training is provided in Table 1.

Table 1: Summary of Training Configuration

Hyperparameter	Value	Comment
Epochs	50	Number of passes
Image Size	640 × 640	Size of input images
Initial Learning Rate	0.001	Sets the initial learning rate
Final Learning Rate Factor	1%	Determines how much the learning rate is reduced by the end of training
Warmup Epoch	3	Smooths the start of training by gradually increasing the learning rate

4.2 Training Environment

The model training was performed using the PyTorch framework and Ultralytics YOLO implementation in Python. All experiments were conducted on Google Colab using freely available GPU resources (PGPUs), without any additional paid upgrades. OpenCV was employed for basic image preprocessing, while Albumentations was used as the primary augmentation framework to apply advanced data augmentation techniques. This setup ensured an efficient, flexible, and cost-effective environment for model development.

4.3 Evaluation Metrics

Multiple metrics were employed, assessing both localization accuracy and classification reliability to evaluate the object detection model's performance. The primary metrics include Precision, Recall, F1 Score, Intersection over Union (IoU),

Mean Average Precision at IoU threshold 0.5 (mAP@50), and Mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 (mAP@50-95). Their mathematical definitions are provided below:

Precision measures the proportion of correct positive predictions out of all positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.1}$$

Where TP = True Positives, FP = False Positives. A high precision indicates that the model predicts an object, it is usually correct.

Recall measures the proportion of actual positives correctly detected:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.2}$$

Where TP = True Positives, FN = False Negatives. A high recall indicates that most of the actual objects present are correctly detected by the model.

F1 Score is the harmonic mean of Precision and Recall which provides a single metric that accounts for both false positives and false negatives.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

IoU measures the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{4.4}$$

Where Area of Overlap is the intersection between the predicted and ground truth bounding boxes and Area of Union is the total combined area of both bounding boxes.

Mean Average Precision at IoU 0.5 (mAP@50) evaluates detection performance by considering a prediction correct if its IoU with the ground truth is greater than 0.5. The average precision (AP) is computed for each class, and mAP is the mean across all classes:

$$\text{mAP@50} = \frac{1}{N}\sum_{i=0}^{N} AP_i \quad \text{(at IoU threshold} \geq 0.5) \tag{4.4}$$

Where N = Number of classes, AP_i = Average Precision for class i

Mean Average Precision from IoU 0.5 to 0.95 (mAP@50-95) is a stricter metric where the AP is averaged over multiple IoU thresholds (from 0.5 to 0.95 in steps of 0.05):

$$\text{mAP@50} - 95 = \frac{1}{10}\sum_{iou=0.5}^{0.95} \text{mAP at each IoU threshold} \tag{4.5}$$

This metric gives a more comprehensive view of model robustness across different degrees of localization accuracy.

5. Result Analysis

This section presents a detailed analysis of the model's performance through quantitative evaluation, qualitative assessment, ablation studies, and comparative analysis. For all computations, a confidence threshold of 0.25 was applied, and the levels provided by the dataset were considered as the ground truth annotations.

5.1 Quantitative Analysis

Table 2 presents a comprehensive evaluation of the model's performance using key detection metrics. The results are reported for all classes combined, and separately for the weed and crop classes, allowing a detailed assessment of class-specific behavior.

Metrics considered include Precision, Recall, F1 Score, Intersection over Union (IoU), mean Average Precision at IoU threshold 0.5 (mAP@50), and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@50-95).

Table 2: Performance metrics of the model across all classes, weed class, and crop class, including Precision, Recall, F1 Score, IoU, mAP@50, and mAP@50-95.

Class / Metrices	All	Weed	Crop
Precision	0.773	0.645	0.902
Recall	0.794	0.808	0.780
F1 Score	0.783	0.717	0.837
IoU	0.715	0.794	0.689
mAP@50	0.804	0.739	0.869
mAP@50-95	0.517	0.467	0.567

As shown in Table 2, the model achieved a strong overall balance between precision (0.773) and recall (0.794). The crop class demonstrated higher precision (0.902) and F1 Score (0.837) compared to the weed class (precision 0.645 and F1 Score 0.717), indicating more confident and consistent detections for crops. Meanwhile, the weed class achieved slightly higher recall (0.808) than crops (0.780), suggesting that weeds were detected more frequently, albeit with lower precision. The IoU and mAP values further reinforce these trends, with an overall mAP@50 of 0.804 and mAP@50-95 of 0.517, confirming the model's strong but class-dependent detection capabilities.

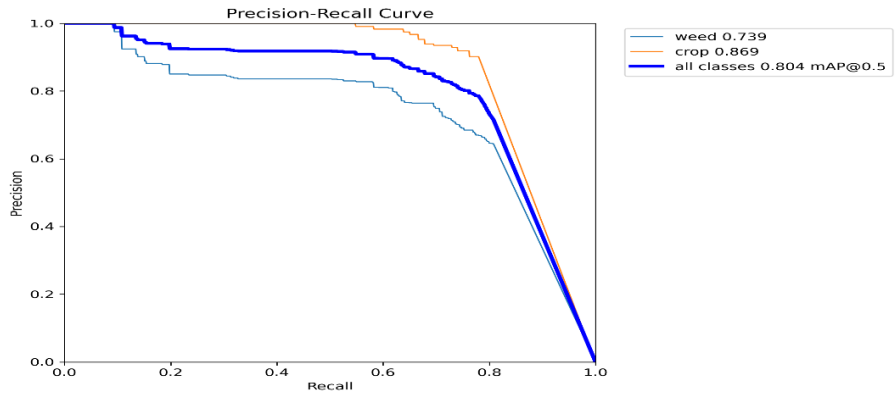


Figure 11: Precision–Recall (P–R) curves for weed, crop, and all classes, illustrating class-specific detection performance.

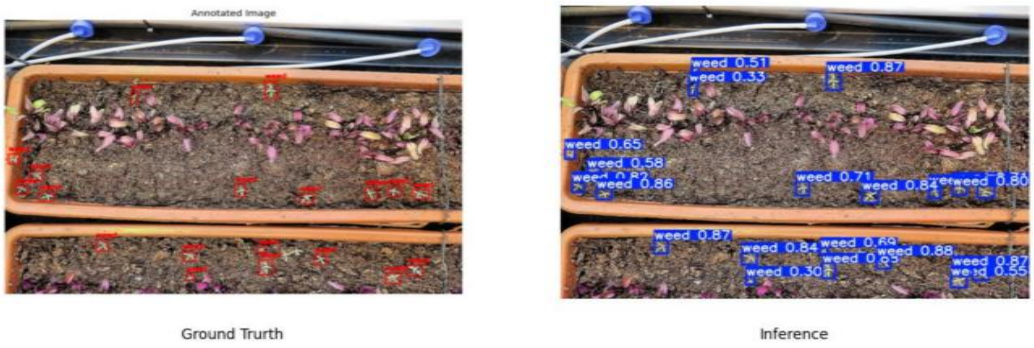
Figure 11 provides a visual representation of these trends through the precision–recall (P–R) curves for each class. The crop class exhibits a steep curve, maintaining high precision across varying recall thresholds, consistent with its strong precision and F1 Score. In contrast, the weed class shows a more gradual decline in precision as recall increases, reflecting its higher recall but lower precision. The overall curve for all classes demonstrates a balanced behavior, achieving a mAP@50 of 0.804. These P–R curves further validate the model's ability to perform reliably across classes, highlighting its robustness and effectiveness across diverse detection thresholds.

5.2 Qualitative Analysis

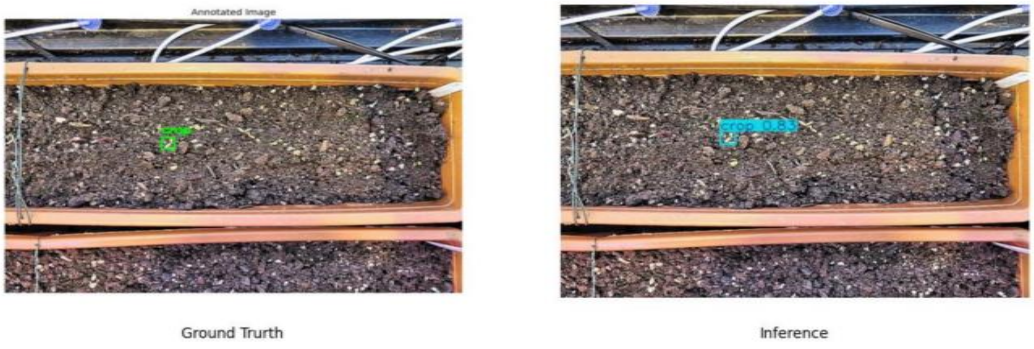
A qualitative analysis was conducted to visually assess the model's performance across different scenarios. Figure 5.2 presents side-by-side comparisons of ground truth annotations and model inferences for various sample types.

In the first set (Figure 11(a)), multiple weed instances are present. The model effectively localizes and classifies the weeds with corresponding confidence scores, demonstrating its ability to detect densely clustered objects despite background clutter. The second and third sets (Figure 11(b) and Figure 11(c)) focus on crop detection. In Figure 11(b), a small emerging crop is correctly identified with high

confidence, reflecting the model's sensitivity to early-stage crop features. Figure 11(c) presents an augmented version of the same sample, where the model continues to accurately detect the crop with stable confidence values, indicating strong robustness and generalization even under augmentation-induced variations.



(a) Weed Class Interference



(b) Crop Class Interference

(c) Interference on augmented Sample

Figure 12: Qualitative comparison between ground truth annotations and model inferences (a) Weed class detection on a sample with dense weed growth. (b) Crop class detection on an original sample. (c) Detection on an augmented version of the crop sample

Overall, the qualitative results visually confirm the model's ability to maintain high localization and classification accuracy across both original and augmented data samples. These findings highlight the model's strong capability to reliably recognize both weed and crop instances under varying conditions. Furthermore, the model effectively leverages the diversity of the training dataset, enabling it to generalize well to augmented scenarios that simulate open-environment variability and potential real-world changes.

5.3 Ablation Analysis

Table 4 presents the results of an ablation study conducted to investigate the individual and combined effects of data augmentation and image preprocessing on the model's performance. Three experimental setups were tested: (i) with both augmentation and preprocessing, (ii) with augmentation but without preprocessing, and (iii) without augmentation or preprocessing. Performance was evaluated using key detection metrics for all classes and separately for the weed and crop categories.

As presented in Table 4, the application of both augmentation and image preprocessing resulted in the highest model performance across all evaluation metrics, with a precision of 0.773, recall of 0.794, F1 Score of 0.783, mAP@50 of 0.804, and mAP@50-95 of 0.517. Notably, for the crop class, the model achieved a precision of 0.902, demonstrating strong classification confidence, while for the weed class, a recall of 0.808 indicated the model's ability to successfully detect most weed instances despite a relatively lower precision (0.645).

Table 3: Ablation study evaluating the effects of augmentation and image preprocessing on model performance, reported for all classes, weed, and crop, using metrics including Precision, Recall, F1 Score, mAP@50, and mAP@50-95.

Techniques	**Class** / **Metrices**	**All**	**Weed**	**Crop**
With Augmentation and Image Preprocessing	Precision	0.773	0.645	0.902
	Recall	0.794	0.808	0.780
	F1 Score	0.783	0.717	0.837
	mAP@50	0.804	0.739	0.869
	mAP@50-95	0.517	0.467	0.567
With Augmentation and without Image Preprocessing	Precision	0.575	0.492	0.658
	Recall	0.588	0.719	0.457
	F1 Score	0.581	0.584	0.539
	mAP@50	0.570	0.563	0.577
	mAP@50-95	0.311	0.293	0.328
Without Augmentation and Image Preprocessing	Precision	0.427	0.855	0
	Recall	0.018	0.036	0
	F1 Score	0.035	0.069	0
	mAP@50	0.178	0.357	0
	mAP@50-95	0.063	0.125	0

Removing image preprocessing while retaining data augmentation led to a significant degradation in precision (dropping from 0.773 to 0.575) and F1 Score (dropping from 0.783 to 0.581), particularly for crops. This suggests that image preprocessing steps

including CLAHE and sharpening filter are crucial for maintaining the input consistency needed for high-precision detections. While recall values for weeds remained comparatively higher even without preprocessing (0.719), precision dropped, indicating an increase in false positives.

In contrast, the scenario without both augmentation and preprocessing resulted in serious performance downgrade, especially for the crop class, where both precision and recall dropped to zero. For weeds, the model achieved high nominal precision (0.855) but extremely poor recall (0.036), implying that while the few detections made were correct, the model failed to identify the majority of weed instances. The mAP@50 and mAP@50-95 values similarly reflect these trends, with overall mAP@50 falling from 0.804 (with augmentation and preprocessing) to just 0.178 (without either technique).

These results conclusively demonstrate that data augmentation and preprocessing are not optional refinements but essential components for achieving robust, generalized performance in object detection tasks. Augmentation primarily improves recall and generalization by diversifying the training data, while preprocessing stabilizes feature extraction by ensuring consistent input quality. The combined effect of both techniques is particularly critical in challenging scenarios involving class imbalance, where weaker classes like weed can otherwise suffer substantial detection losses.

5.4 Comparison

Table 5 provides a comparative analysis between the proposed machine vision pipeline and several state-of-the-art weed detection techniques reported in the

literature. For consistency and fairness, the F1 Score was selected as the reference metric for comparison, as it effectively balances both precision and recall, offering a comprehensive evaluation of detection performance. The table includes different models, datasets, and corresponding F1 Scores reported in prior works, alongside the performance achieved by the proposed method.

*Table 4: Comparison of the Proposed Machine Vision Pipeline with State-of-the-Art Weed Detection Techniques (**Bolded result** indicates the performance of the proposed method)*

Reference	Methods	Dataset	F1 Score
S. K. Valicharla et al. [8]	Mask R-CNN (Detectron2) with synthetic data and augmentation	Synthetic dataset (2-class setup)	0.720
A. N. V. Sivakumar et al. [9]	Faster R-CNN (200 proposals)	Soybean field image (mid- to late-season weeds)	0.660
J Lekha et al. [10]	YOLOv7 integrated with IoT sensor data	Labeled image and video datasets for sesame crops	0.780
I. Matvienko et al. [11]	U-Net with SE-blocks and Bayesian aggregation	Single-image Sentinel-2 satellite data	0.660
P. De Marinis et al. [12]	RoWeeder: Hough Transform + SegFormer	UAV multispectral images with pseudo-labels	0.753
Proposed Machine Vision Pipeline	YOLOv10 with Augmentation and Image Preprocessing	Crop and Weed Dataset [6]	**0.783**

As shown in Table 5, the proposed machine vision pipeline achieved the highest F1 Score of 0.783 on the Crop and Weed Dataset [6], outperforming all compared state-of-the-art methods. Compared to other techniques based on Mask R-CNN [8], Faster R-CNN [9], YOLOv7 with IoT integration [10], U-Net with Bayesian aggregation [11], and transformer-based segmentation models [12], the proposed method consistently delivered superior detection accuracy. The strong performance can be

attributed to class-aware augmentation, which addressed class imbalance, and robust preprocessing strategies that improved input quality. While other methods reported F1 Scores in the range of 0.660 to 0.780, the proposed pipeline achieved a better balance between precision and recall, demonstrating its effectiveness for practical weed detection applications.

6. Discussion and Analysis

This section discusses the model's performance across generalization, confidence threshold effects, reliability, use case alignment, deployment feasibility, and project modifications from initial project proposal. Each subsection provides a detailed analysis supporting the practical application of the model in real-world agricultural environments.

6.1 Cross-Dataset Generalization

The generalization ability of the model was evaluated by testing it on a different crop and weed detection dataset [22]. The table below summarizes the model's detection performance on the external dataset:

Table 5: Performance Metrics for Weed and Crop Classification

Class / Metrices	All	Weed	Crop
Precision	0.716	0.684	0.741
Recall	0.594	0.583	0.602
F1 Score	0.673	0.629	0.737
IoU	0.655	0.674	0.573

A total of 50 images were randomly selected from this dataset. Before evaluation, all images underwent the same preprocessing steps used during training. A fixed confidence threshold of 0.25 was applied during inference to maintain consistency with earlier evaluations. Compared to the original test set results, the model maintained reasonable precision and Intersection over Union (IoU) values, despite some expected drops in recall due to the domain shift. These results indicate that the model can generalize to new environments without significant degradation in performance, supporting its practical utility in diverse agricultural settings.

6.2 Effect of Confidence Threshold Analysis

The figure-13 below present the Precision-Confidence, Recall-Confidence, and F1-Confidence curves for the weed class, crop class, and all classes combined. These curves provide a visual summary of how precision, recall, and F1 score vary with different confidence thresholds during model inference. As the confidence threshold increases, precision improves for both crop and weed detection, while recall drops sharply, causing a steady decline in the F1 score.

A high threshold results in fewer false positives but also misses many true positives, which can be critical in agricultural applications where maximizing detection is important. On the other hand, a low threshold captures more true positives but also introduces many false positives, as environmental variations such as soil, shadows, or residue may be mistakenly classified as crops or weeds. Considering these trade-offs, a confidence threshold of 0.25 was selected for this project to maintain a balance between precision and recall, ensuring reliable detection while minimizing errors that could impact practical field decision-making.

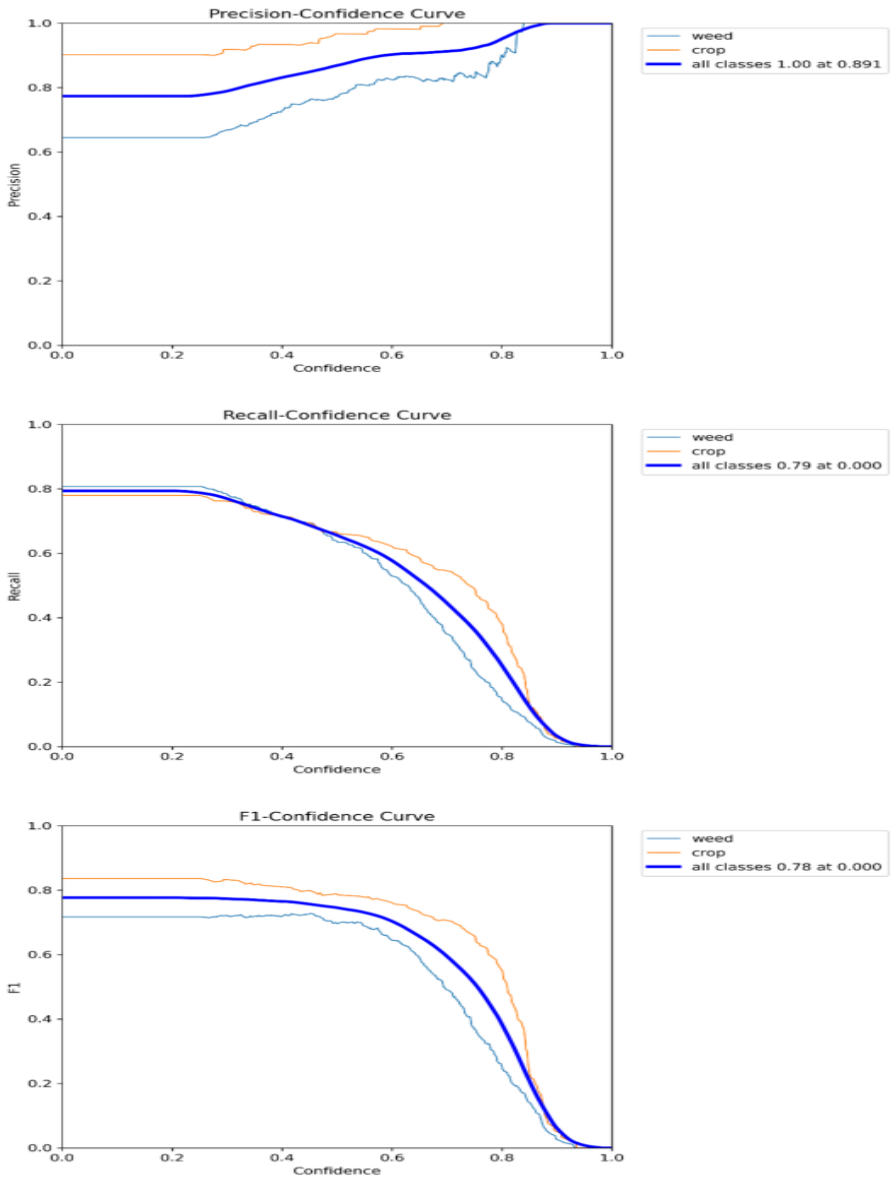


Figure 13: Precision, Recall, and F1 Score versus Confidence for Weed and Crop Detection

From the above consideration, a trade-off between precision and recall is necessary to optimize detection performance under practical field conditions. Considering these trade-offs, a confidence threshold of 0.25 was selected for this project to maintain a balance between precision and recall, ensuring reliable detection while minimizing errors that could impact practical field decision-making.

6.3 Evaluation Reliability

Figure 14 illustrate a normalized confusion matrix across three classes- background, weed, and crop. Using normalization helps interpret the results more clearly in the presence of class imbalance, as background regions dominate field images. The matrix reveals that while the model performs well in detecting crops and weeds, a considerable portion of background areas were incorrectly predicted as weeds. This suggests a mismatch between model predictions and ground truth annotations.

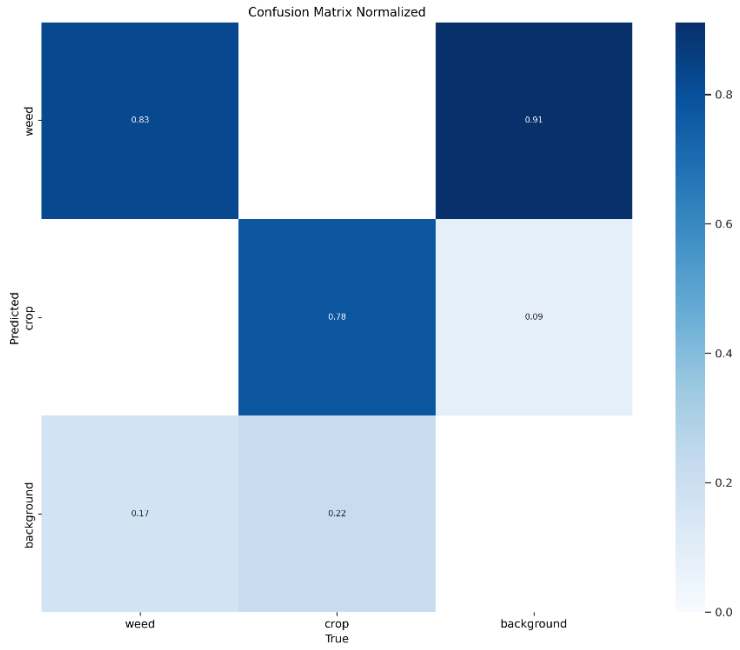


Figure 14: Normalized confusion matrix showing classification performance across three categories: weed, crop, and background.

This confusion likely arises because certain true crop and weed instances were not annotated in the ground truth, leading the model's correct detections to be treated as false positives. As a result, the evaluation metrics may underestimate the actual performance of the model. To further investigate this, a qualitative comparison between ground truth labels and model inference results is provided, showing examples where the model detected crops and weeds that were not labeled.

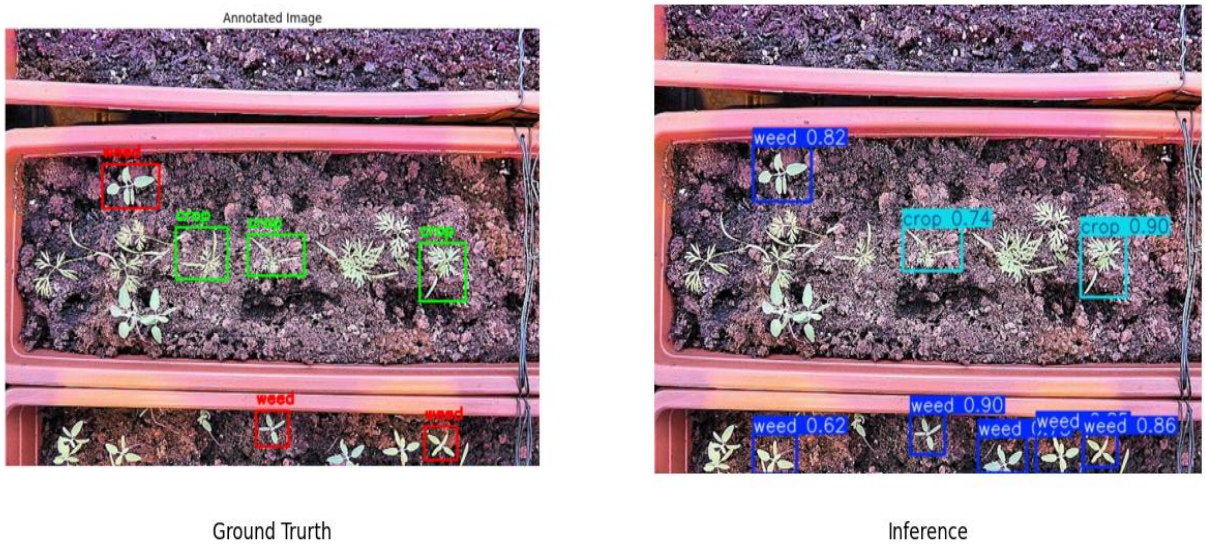


Figure 15: Evaluation of ground truth annotation reliability and Model Inference

The qualitative comparison confirms that the model can reliably detect additional plants that were overlooked in the ground truth annotations. Although these correct detections are penalized as false positives during standard evaluation, they demonstrate the model's strong practical performance in identifying real field targets. This misalignment also contributes to the lower precision observed in the weed class, as many correctly detected weeds are counted as errors due to missing annotations. Moreover, the observation suggest that the model is overperforming relative to the estimated evaluation metrics. Overall, this reinforces that the apparent background misclassifications partly reflect annotation gaps, rather than true model errors.

6.4 Hardware Requirements and Deployment Feasibility

A lightweight YOLO model containing approximately 15.3 million parameters was developed for weed and crop detection applications. The model requires about 60 MB of storage space and approximately 500 MB to 1 GB of active memory during inference, depending on the level of optimization applied. Additional memory is required for image preprocessing operations suggesting a practical minimum of 2 GB

RAM for smooth deployment. These requirements are easily met by most modern edge devices designed for AI inference. The table-6 below summarizes a selection of edge devices suitable for offline deployment of the model.

Table 6: Hardware specifications and suitability of selected edge devices for weed and crop detection model deployment.

Device	RAM	Power Req.	Price (CAD)	Max Parameters	Comments
Jetson Nano (GPU acceleration)	4 GB	5 – 10 W	$981.99 [17]	~30 – 40 M	TensorRT optimization recommended
Raspberry Pi 5	8 GB	~5 W	$179.99 [18]	~15 – 20 M	Best paired with Coral USB for faster inference
VIM3 Pro	4 GB	~5 W	$264.66 [19]	~40 – 50 M	Powerful NPU acceleration
OAK D-Lite	512 MB	2–3 W	$206.91 [20]	~20 – 30 M	Integrated computer system; ideal for embedded CV setups

All listed devices meet the memory and compute requirements necessary for offline inference, offering a balance of performance, power efficiency, and affordability. As the primary focus of this project was on developing a reliable detection algorithm, the hardware table demonstrates that practical deployment can be achieved in a cost-effective manner. The Jetson Nano and VIM3 Pro provide dedicated acceleration through GPU and NPU respectively, while the OAK-D-Lite offers an integrated solution combining computation and vision capabilities. The Raspberry Pi 5 supports the model with CPU-based inference and can be paired with the Coral USB Accelerator [21] to further improve inference speed for TensorFlow Lite deployments.

6.5 Changes from Original Proposal

The original proposal outlines a plan to use a U-Net architecture with a modified attention module. However, the project shifted to the YOLO model, as our dataset

provides bounding box annotations rather than pixel-wise segmentation masks. U-Net is typically used for pixel level segmentation, while object localization with bounding boxes is better addressed by YOLO models which also integrates attention mechanisms internally.

The evaluation methodology was similarly adjusted. Instead of K-Fold cross-validation, cross-dataset evaluation was used to assess generalization. Qualitative analysis also shifted from GradCAM based attention map visualization to inference-based analysis, aligning with the object detection framework. These adjustments kept the project aligned with its original objectives while better matching the dataset characteristics and application requirements.

7. Limitations

While the proposed model shows strong performance, several limitations should be noted. Some correct detections are penalized due to missing ground truth annotations, slightly underestimating the true model capability. The dataset's class imbalance, particularly the low number of crop instances, may affect generalization under different field conditions. Although the model is lightweight and suitable for edge deployment, extreme variations in lighting, occlusion, and soil appearance could still impact detection reliability. Broader validation across diverse datasets and seasons is recommended for future work.

8. Future Directions

In the short term, the model's strong weed detection precision makes it valuable for early field assessment before cultivation, when only weeds or background are present.

Integrating the model with GPS mapping could help estimate the spatial distribution of weeds across large areas, allowing farmers to identify high-risk zones and plan targeted weed control strategies before planting crops. Over the longer term, collaboration with agricultural experts could expand the system's capabilities to include broader biodiversity monitoring and field ecosystem analysis, extending its usefulness beyond simple weed detection. While integrating vision-based detection with large language models to create agricultural VQA (Visual Question Answering) is an exciting possibility, it would represent a significant step beyond the current project scope and would require substantial further research and development.

9. Conclusion

This project, titled Deepgreen, successfully developed and evaluated a machine vision pipeline for weed and crop detection under realistic agricultural conditions. The proposed system addressed key challenges such as class imbalance, environmental variability, and hardware deployment constraints. Comparative analysis showed that the proposed approach outperformed several state-of-the-art methods in detection accuracy. Although some limitations remain due to incomplete ground truth annotations, the model demonstrated reliable generalization to external datasets. The system offers significant potential to support early field assessment, automate weed monitoring, and reduce herbicide usage through more targeted and efficient agricultural interventions.

10. Tools and Resources

- Weed and Crop Dataset [6], [22]

- Google Colab
- Python, Ultralytics (Including YOLO with COCO weights)
- OpenCV, NumPy
- Large Language Model (LLM) for writing refinement

References

[1] J. M. Mostafalou and M. Abdollahi, “Pesticides and human chronic diseases: Evidences, mechanisms, and perspectives,” Toxicology and Applied Pharmacology, vol. 268, no. 2, pp. 157–177, 2013.

[2] M. El Jaouhari, G. Damour, P. Tixier, and M. Coulis, “Glyphosate reduces the biodiversity of soil macrofauna and benefits exotic over native species in a tropical agroecosystem,” Basic and Applied Ecology, vol. 73, pp. 18–26, 2023. doi: 10.1016/j.baae.2023.10.001

[3] Farmonaut, “Revolutionizing Canadian agriculture: How precision farming technology is boosting crop yields in Guelph,” Farmonaut, 2023. [Online]. Available: https://farmonaut.com/canada/revolutionizing-canadian-agriculture-how-precision-farming-technology-is-boosting-crop-yields-in-guelph/. [Accessed: Apr. 19, 2025].

[4] A. Shrestha, R. S. DeFelice, C. L. Sprague, and W. J. Everman, “Impact of weed interference on potato yield and economic return in Michigan,” Crop Protection, vol. 187, p. 106401, 2024. doi: 10.1016/j.cropro.2024.106401

[5] Health Canada, “Protecting your health and the environment,” Government of Canada, 2023. [Online]. Available: https://www.canada.ca/en/health-canada/services/consumer-product-safety/pesticides-pest-management/public/protecting-your-health-environment.html. [Accessed: Apr. 19, 2025].

[6] V. G. Gupta, "Weed Detection," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/vvatsalggupta/weed-detection. [Accessed: Mar. 10, 2025].

[7] N. Islam, M. N. S. Huda, K. R. Islam, M. U. Ahmed, and M. S. Moniruzzaman, "Early Weed Detection Using Image Processing and Machine Learning Techniques in an Australian Chilli Farm," Agriculture, vol. 11, no. 5, p. 387, Apr. 2021, doi: 10.3390/agriculture11050387.

[8] S. K. Valicharla, Weed Recognition in Agriculture: A Mask R-CNN Approach, M.S. thesis, Lane Dept. of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA, 2021. [Online]. Available: https://researchrepository.wvu.edu/etd/8102

[9] A. N. V. Sivakumar, J. Li, S. Scott, E. Psota, A. J. Jhala, J. D. Luck, and Y. Shi, "Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Mid- to Late-Season Weed Detection in UAV Imagery," Remote Sens., vol. 12, no. 13, p. 2136, Jul. 2020, doi: 10.3390/rs12132136.

[10] J. Lekha and S. Vijayalakshmi, "Enhanced Weed Detection in Sustainable Agriculture: A You Only Look Once v7 and Internet of Things Sensor Approach for Maximizing Crop Quality," Eng. Proc., vol. 82, p. 100, Nov. 2024, presented at the 11th Int. Electron. Conf. on Sensors and Applications (ECSA-11). doi: 10.3390/ecsa-11-20380

[11] I. Matvienko, M. Gasanov, A. Petrovskaia, M. Kuznetsov, R. Jana, M. Pukalchik, and I. Oseledets, "Bayesian Aggregation Improves Traditional Single-Image Crop Classification Approaches," Sensors, vol. 22, no. 22, p. 8600, Nov. 2022, doi: 10.3390/s22228600.

[12] P. De Marinis, G. Vessio, and G. Castellano, "RoWeeder: Unsupervised Weed Mapping through Crop-Row Detection," in Proc. CVPPA Workshop at the European Conf. on Computer Vision (ECCV), Oct. 2024, arXiv:2410.04983. [Online]. Available: https://arxiv.org/abs/2410.04983

[13] X. Jin, J. Che, and Y. Chen, "Weed Identification Using Deep Learning and Image

Processing in Vegetable Plantation," IEEE Access, vol. 9, pp. 10940–10950, Jan. 2021, doi: 10.1109/ACCESS.2021.3050296.

[14] N. Razfar, J. True, R. Bassiouny, V. Venkatesh, and R. Kashef, "Weed Detection in Soybean Crops Using Custom Lightweight Deep Learning Models," J. Agric. Food Res., vol. 8, p. 100308, Apr. 2022, doi: 10.1016/j.jafr.2022.100308.

[15] M. A. Haq, "CNN Based Automated Weed Detection System Using UAV Imagery," Comput. Syst. Sci. Eng., vol. 42, no. 2, pp. 837–849, Jan. 2022, doi: 10.32604/csse.2022.023016.

[16] Ultralytics, "Ultralytics YOLO GitHub Repository," GitHub, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: 25-Apr-2025]

[17] NVIDIA Jetson Nano Developer Kit, Amazon. [Online]. Available: Amazon Link. [Accessed: 26-Apr-2025]

[18] Raspberry Pi 5 Single Board Computer with Active Cooler, Amazon. [Online]. Available: Amazon Link. [Accessed: 26-Apr-2025]

[19] Khadas VIM3 Pro Single Board Computer, Amazon. [Online]. Available: Amazon Link. [Accessed: 26-Apr-2025]

[20] Luxonis OAK-D-Lite Auto-Focus Robotics Camera, Amazon. [Online]. Available: Amazon Link. [Accessed: 26-Apr-2025]

[21] Google Coral USB Accelerator, Amazon. [Online]. Available: Amazon Link. [Accessed: 26-Apr-2025]

[22] R. Dabhi and D. Makwana, "Crop and Weed Detection Data," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/ravirajsinh45/crop-and-weed-detection-data-with-bounding-boxes. [Accessed: Mar. 10, 2025].