**A**

**Project Phase II Report on**

**Computer Vision Based Weed Killing Rover.**

### SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE



### IN THE PARTIAL FULFILMENT OF PROJECT PART-II OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE

### OF

**BACHELOR OF ELECTRONICS & TELECOMMUNICATION ENGINEERING**

### BY

Atharva Junonikar BE ENTC 07 Ayushi Singh BE ENTC 08

Prajit Nair BE ENTC 38

Under the guidance of

**Prof. Dr. Pallavi Adke**



### Department of Electronics and Telecommunication Engineering

Pimpri Chinchwad College of Engineering and Research,Ravet, Pune-412101

ACADEMIC YEAR:2023-24

#### PIMPRI CHINCHWAD EDUCATION TRUST’S

#### Pimpri Chinchwad College of Engineering and Research, Pune-412101



**CERTIFICATE**

This is certify that the Project Phase-II report entitled

#### “COMPUTER VISION BASED WEED KILLING ROVER”,

Submitted by

#### Atharva Junonikar B191223007 Ayushi Singh B191223008 Prajit Nair B191223038

is a bonafide work carried out by them under the supervision of **Prof. Dr. Pallavi Adke** and it is approved for the partial fulfillment of the Project Phase II of the requirement of Savitribai Phule Pune University,Pune for the award of the degree of Bachelor of Engineering (Electronics &Telecommunication).The Dissertation work has not been earlier submitted to any other institute or university for the award of degree.

Prof. Dr. Pallavi Adke Prof.Kishor Bhangale

#### Project Guide Project Co-ordinator

Prof. Dr.Rahul G.Mapari Prof.Dr.H.U.Tiwari

#### Head of Department Principal

**Acknowledgement**

We are greatly indebted to Project Guide **Dr. Prof. Dr. Pallavi Adke** for her able guidance throughout the course of this work.It has been anal together different experience to work with her and we would like to thank her for her help, suggestions and numerous discussions.

We are heartily thankful to **Prof. Dr.HarishU.Tiwari** (Principal, Pimpri Chinchwad College of Engineering & Research,Ravet)for providing research en vironment;also for his kind inspiration.

We are gladly taking this opportunity to thank Head of Electronics ’I&’ Telecommunication Engineering **Prof. Dr. Rahul Mapari** and project Coordinator **Mr. Kishor Bhangale** for their valuable guidance and providing facility during progress of seminar.

Last but not least we are also thankful to all those who help directly or indirectly to develop this Project work and complete it successfully. With Deep Reverence,

Atharva Junonikar Ayushi Singh Prajit Nair

Date:

Place:

**Contents**

1. [Introduction](#_bookmark0) 1
   1. [Introduction](#_bookmark1) 1
   2. [Research background](#_bookmark2) 1
   3. [Problem Statement](#_bookmark3) 2
   4. [Objectives of the Study](#_bookmark4) 2
   5. [Scope of the Study](#_bookmark5) 2
   6. [Overview of Proposed System](#_bookmark6) 2
   7. [Organization of the Report](#_bookmark8) 3
2. [Literature Review](#_bookmark9) 4
   1. [Literature Survey](#_bookmark10) 4
   2. [Finding from Literature Survey](#_bookmark11) 5
   3. [Gap Identification](#_bookmark12) 5
3. [Proposed Methodology](#_bookmark13) 6
   1. [System Implementation](#_bookmark14) 6
   2. [Procedure for training the model](#_bookmark16) 7
   3. [Yolov5 Architecture](#_bookmark18) 10
   4. [Design and Formulation](#_bookmark20) 11
   5. [Objectives](#_bookmark22) 11
   6. [Working Flow](#_bookmark23) 12
   7. [Hardware Setup](#_bookmark25) 13
      1. [Hardware Components and Specifications](#_bookmark26) 13
   8. [Software setup](#_bookmark34) 17
4. [Results and Discussions](#_bookmark41) 22
   1. [Software Results](#_bookmark42) 22
   2. [Hardware Results](#_bookmark47) 26
   3. [Summary](#_bookmark50) 27
5. [Advantage and Applications](#_bookmark51) 28
   1. [Advantages](#_bookmark52) 28
   2. [Applications](#_bookmark53) 28
6. [Conclusion and Future Scope](#_bookmark54) 29
   1. [Conclusion](#_bookmark55) 29
   2. [Future Scope](#_bookmark56) 29
7. [References](#_bookmark57) 30
8. [Project Outcomes](#_bookmark58) 32
   1. [Plagriasim Report](#_bookmark59) 32
   2. [Copyright Certificate and Work Uploaded](#_bookmark60) 33
      1. [Copyright Certificate](#_bookmark61) 33
      2. [Synopsis](#_bookmark62) 35
      3. [Form 14](#_bookmark63) 39
   3. [Project Competition Participation Certificates](#_bookmark64) 44
   4. [Publication Document](#_bookmark65) 47
      1. [Publication Work](#_bookmark66) 47
      2. [Publication Copyright Mail](#_bookmark67) 54

**List of Figures**

[1.1 Block Diagram of Proposed System](#_bookmark7) 3

* 1. [Block Diagram of proposed system](#_bookmark15) 6
  2. [Flowchart of YoloV5](#_bookmark17) 7
  3. [Network Architecture of YOLO V5](#_bookmark19) 10
  4. [Schematic Diagram of Rover](#_bookmark21) 11
  5. [Flowchart](#_bookmark24) 12
  6. [Arduino Mega](#_bookmark27) 13
  7. [16-channel 12-bit PWM servo driver](#_bookmark28) 13

[3.8 Servo 995](#_bookmark29) 14

* 1. [Servo MG90s](#_bookmark30) 15
  2. [Servo SG90](#_bookmark31) 15

[3.11 SMPS 12v, 5A](#_bookmark32) 16

* 1. [Jumper cable](#_bookmark33) 16
  2. [Anaconda Dashboard](#_bookmark35) 17
  3. [Jupyter Dasboard](#_bookmark36) 17
  4. [Google Colab](#_bookmark37) 18
  5. [Arduino IDE](#_bookmark38) 19
  6. [Sample Images](#_bookmark39) 20
  7. [Sample Images](#_bookmark40) 20
  8. [Confusion Matrix Graph](#_bookmark43) 22
  9. [F1-Confidence Curve](#_bookmark44) 23
  10. [Precision-Confidence Curve](#_bookmark45) 24
  11. [Recall-Confidence Curve](#_bookmark46) 24
  12. [Simulation results for the position of the robotic arm](#_bookmark48) 26
  13. [Results of working of the robotic arm](#_bookmark49) 26

#### Abstract

India’s primary industry, agriculture, accounts for the majority of the country’s GDP. Weed growth is one of the primary factors that lowers agricultural productivity. In the field, the combination of machine vision and image processing proved to be a reliable way for point-to-point real-time detection of weed and for other crop detection, providing accurate data to manage the weed. tailored to individual sites. The developments in information and electronic technologies are to blame for this. Manual herbicide spraying is one of the most common ways to keep weeds under control. However, this strategy has a lot of negative effects on both agricultural and consumer health. This report summarises the advances in weed detection using open-CV and image detection methods. A detailed explanation was given of the five primary weed identification techniques: firstly, pre-processing; secondly, feature extraction; third, segmentation; fourth, extraction; and lastly, classification. The primary difficulty with this model is distinguishing crops from weeds, which can occasionally have identical features and appearances. The model was trained with a sizable dataset in order to address this problem. As a result, this technology will now detect the weed and spray the herbicide in an adequate amount in a targeted manner. This research paper will provide a comprehensive study of various deep learning methods that can be useful for automatic detection and various image processing methodologies. It also provides performance metrics, such as accuracy and precision, for the various techniques employed.

**Chapter 1**

**Introduction**

# Introduction

Weeds were previously eliminated by handpicking and spraying chemicals. Robot technology has come up with another idea: a robot for efficient eradication of weeds using state-of-the-art technologies. Because it uses com- puter vision, this robot is able to comprehend its surroundings and make judgements based on its observations. It’s similar to educating a robot to perceive and respond. Among other things, computer vision helps in facial recognition and object tracking . When the robot detects weeds, it sprays them using a specialised arm which dispenses herbicide rapidly so that it easily finds weed using YOLOv5 DEEP LEARNING MODEL.

We just have to gather information, teach the robot using images of weeds, and then test the robot in order for it to function. Numerous methods of weed identification have already been studied by several scientists.

One way to distinguish weeds from other plants is to use their traits. Using the SMACH library to control the robot’s movements is an additional concept. To identify weeds, there are additional methods, such as Arti- ficial Neural Networks (ANN), that simulate the functioning of human brains. Although some robots have been designed to trim weeds, occasionally they can harm crops. Some concentrate on identifying weed stems in fields. Researchers are working to make these robots more capable of identifying a wider variety of weeds and operating in various environments. Additionally, there are robots that provide the ideal environment and nutrients to plants to enhance their growth. However, these are not weed-killing robots. All in all, researchers are exploring a variety of approaches to better control weeds with robots.

# Research background

Prior to starting a project that entails using robotics and computer vision to build a weed-killing rover, extensive research covering several crucial topics was done:

* + 1. **Weed Plant Biology :** [11]Acquired knowledge of the characteristics, stages, and growth cycles of weeds. Research was done to understand its visual characteristics at various growth stages.
    2. **Optimal Termination Techniques:** Explored the most efficient methods for weed killing that won’t damage the crops or lower output.

#### Automation and Robotics:

1. **Robotic Arm Capabilities:** [13] Analysed the accuracy of current robotic arm designs and their applica- bility to precise tasks.
2. **Mobile Robotics:** Investigated rover designs, propulsion systems, and techniques for traversing diverse terrains.

#### Computer vision and image processing:

1. **Algorithms:** [9][10]We used OpenCV to research various methods of object recognition and detection, including CNNs, Haar cascades, and contour detection for plant recognition.
2. **Image processing techniques:** [3]We inquired about colour thresholding, feature extraction, and other image processing techniques to distinguish weeds from the background.

# Problem Statement

To develop a weed-killing rover using computer vision which identifies and selectively sprays herbicides on weeds, reducing labour dependencies and decreasing damages on crops.

# Objectives of the Study

The objectives of the proposed works are mentioned as follows.

#### To develop an Image processing model which can identify weed perfectly.

#### To develop a spraying mechanism that can pinpoint the weed.

#### To construct a rover body that can withstand all terrain and will not destroy the crop.

#### To develop a product that is cost efficient and has low maintenance cost.

#### To construct a product that is robust, easy to assemble and maintain.

#### To develop a product that can be used for all types of crops.

# Scope of the Study

A project that builds a weed killing rover that works on computer vision model and that has a following advantages:

Technical features : creating an rover’s arm, cameras, sensors, and motors while ensuring proper hardware integration is known as hardware integration.

Software part: writing an algorithm for object detection , image processing and identifying it while using computer vision.

Creating an algorithm to control the robotics arm and identifying weeds from crops and spraying on it. Developing the navigation and control systems that will enable the rover to go through fields , avoid obstruc-

tions and result the designated crops field.

# Overview of Proposed System

The figure 1.1 displays block diagram of the proposed system. The suggested system records weeds’ color, shape, and size by using cameras placed in fields. CNN also known as Convolutional Neural Network is used by computer vision to analyze these images and discriminate between various weeds. A robotic arm is programmed to spray pesticides on weeds upon identification. With the use of labeled datasets, CNN is trained. The primary objective of the rover is to efficiently navigate the field.

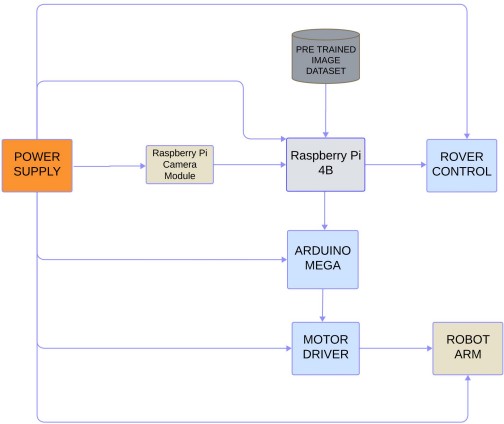


Figure 1.1: Block Diagram of Proposed System

# Organization of the Report

The project report is oraganized as follow:

Chapter 1 provides the information regarding introduction of area, need, relevance, problem statement, objectives and scope of the study. It gives an idea about the research background and the previous technologies used in the project. This chapter gives a brief idea about the scope of study as well. The problem statement and objectives are included in the chapter.

Chapter 2 gives survey of various articles on the proposed technology and a comparison between the previous and latest technology along with that provides the major gaps that paves the way for the development of proposed system. It shows what many researchers have concluded after studying about the topic.

Chapter 3 provides the methodology used in implementing the technology and studying it’s applications. This includes the system implementation, how AR works and the procedure for home automation.

Chapter 4 consist of the results, actual requirements and implementation of the system along with the software and hardware setups.

Chapter 5 carries the information about the outcomes followed by the Advantages and applications of the presented technology.

Chapter 6 Concludes the purpose of the study and gives the futuristic capabilities of the system.

Chapter 7 comprises of all the references utilised during the study of the same. It gives the references to the research work which helped studying the technology better.

**Chapter 2**

**Literature Review**

# Literature Survey

During the earlier research years, a number of surveys were carried out to determine the shortcomings of the current weed management methods. A technique that uses the characteristics and image formation of plants to identify weed species was proposed by MP Arakeri et al. [4].

Cluster four is home to dicotyledon weeds, while clusters zero through three are home to monocotyledon weeds. The training process can be streamlined by classifying the clusters, and prototypes can be identified as belonging to that class. Weed patterns were categorised using this information, although the process required manual control. [14] Kadeghe Fue suggested that the system make use of SMACH, a ROS-independent library for finite state machines. with the use of a 2 dimensional manipulator which moves linearly in both vertical and horizontal directions. The location is determined using the stereo camera settings, and the manipulator and rover are guided to their destination via the finite state machine. PID control is utilised to regulate the rover’s path to the intended spot. [19]

The approach that Riya et al. suggested focuses on the accuracy of weed identification and includes a variety of picture preprocessing techniques. This research also does not focus on the removal methods; instead, ANN, which are used to estimate or approximate functions that can depend on a large number of inputs and are typically unknown. They are inspired by the neuronal structure of the human brain.[20]

A robotic system for classifying weeds and crops based on visual texture was proposed by Pusphavalli et al. However, the issue with that was that using blades to cut and remove the undesirable plants increased the chance of damaging the plants. This uses a lot of electricity as well. [21] Sebastian et al.’s primary focus is on identifying plant stems and estimating their positions in natural environments. According to him, it’s imperative to expand the number of weed species that are automatically recognised and the conditions in which weeds can be identified.[22] A robotic system that grows plants according to human requirements, such as soil nutrients, climate, and seed topology, was demonstrated by Aravinth et al. These methods are effective in addressing the green revolution rather than weed control. [23] A weed detection technique developed by Hossein et al. can identify leaf forms

with 92[24]

Astranand and Baerveldt construct a robotic sugar beet weed management device. It consists of a selective rotary hoe for weeds when crop emergence rates are high and weed densities are low, along with machine vision guidance. In order to get trustworthy results, it also emphasizes the identification of unique circumstances. It clarifies situations like row ends, when plants may be missing in large numbers and rows are not in a straight line. When such occurs, the system ought to sound an alert so that it may take the appropriate action and complete the tasks. Herbicide protection can be achieved by creating various systems for site-specific herbicide spraying to infected areas; however, these systems are unable to distinguish between weeds and crops. [25]

A machine vision system for precision sprayers was proposed by Jeremie et al. It uses a blob detecting feature and a gabor filter to distinguish between plants and weeds. The gabor filter aids in identifying the crop rows in the provided image and uses the Fourier transform to identify the parameters of the filter. It uses color coding to distinguish between crop and weed; the soil is grey, the crop portions are white, and the weeds are black. However, none of their algorithms could identify weed if it was inside a crop row; instead, they were designed to identify only interrow WIR (weed infected rates).

# Finding from Literature Survey

Following a thorough examination of numerous deep learning methods for weed killing, we identified the follow- ing gaps that required attention.

* + 1. Focus on the detection of the weeds was more than precisely killing them.
    2. Accuracy in differentiation of weed from the crops.
    3. Implementation cost due to hardware, software, and maintenance.
    4. Autonomous systems must have less maintenance time.
    5. Obstacles on way that made it difficult for the rover to move and reach the destination.
    6. Could not identify weeds which were hidden behind some branches.

# Gap Identification

Based on extant literature and the current understanding of object detection, several limits may emerge in the approach, observed results, and unresolved questions. All of the gaps and limitations found throughout the papers’ analysis are listed in the table below. This essay explains the current approach, as well as the formulation, design, and data collection techniques, along with some of their shortcomings. It is possible to address problems that were encountered in earlier research publications in light of this approach. After about 25 papers were examined, the gaps listed below were found:

In this project the gaps identified were that in actual fields, the electronic misters employed in the spraying system can be less resistant to wind power[1]. The robot made had a gap that even in perfect settings, it is difficult to categorize plants as either crops or weeds with a system accuracy of just about 70%[2]. The pipeline might not be able to accurately detect cotton blooms due to the diminished ground resolution at higher flying altitudes[3]. In this project the accuracy is affected by lighting conditions. Battery usage need to be optimized to increase battery life[4]. Less effective when different species of weeds are present. If terrain is bad it affects the performance of the robot[5]. Not entirely effective for real-world field situations because of feature-based PBVS’s erratic visual servoing commands, which are brought on by hazy pictures and insufficient light[6]. In early growth stages, the fusion-based algorithm performed less well in identifying agricultural plants[7].

One drawback of the text describing the current 3D reconstruction techniques is that the approaches are less dependable for bigger leaf sizes, such as those found in maize plants[8]. Identification of certain weed species is required for focused management measures like dosing and spraying herbicides[9]. In this project various plant growth phases and the occlusion or overlap of weeds and crops’ leaves[10]. With a system accuracy of only roughly 70%, it is challenging to classify plants as either crops or weeds even in ideal conditions[11]. The spraying system’s electronic misters may be less wind-power resistant in real fields[12]. The algorithm produced a lot of erroneous negative predictions because it struggled to identify the rows with shorter or taller plants[14]. For use on farms, a system needs to be robust and dependable. Robots with small-diameter wheels may encounter difficulties when navigating obstacles in the field[15].

Among the restrictions on using ML and DL algorithms in agriculture are issues with data collection, abnor-

malities in the landscape, restricted control, and computational expense[17]. Lighting conditions have an impact on accuracy. Optimizing battery consumption is necessary to prolong battery life[18]. Training costs a lot. It takes a long time to process for big neural networks[19]. Because it depends on non-maxima suppression, which may reject some stems, it might not detect all stems. In addition, non-stem regions like pinnate leaves or overlapped plants occasionally mimic stem regions[21]. The energy needed to plough a vast area because it takes a lot of energy to cut and turn the dense soil[22]. The kind of weeds and crops growing in the field, the sunlight, the camera’s quality, and other variables could all affect how well this strategy works[23]. The new approach, which is based on the Hough transform, can only recognize plant rows that are either slightly bent or straight. For track- ing non-straight rows, the system must be locally level across the row arrangement.[24]. The method becomes ineffective at high Weed Infestation Rates (WIR) of up to 40%, highlighting the program’s limitations in terms of efficiency[25].

**Chapter 3**

**Proposed Methodology**

Implementing a trained model called YOLO v5 for weed detection is shown in the above block diagram. It is capable of recognizing weed plants from their surroundings.

Four thousand distinct annotated images using Roboflow were used as a database for training this model for improved accuracy.The trained model is then to be implemented by converting the .pt file into .tf file and then creating an environment on Rasberry-Pi.

The Raspberry-Pi initiates the model’s operation by receiving a video feed from the raspberry cam, we use openCV to access camera feeds and use them as inputs. The trained model then processes the video frame by frame and analyses each frame for possible detection, the detected weeds are enclosed in a frame whose coordi- nates are fetched and fed back into a code for the robotic arm which aligns itself with the help of several servos to spray herbicides.

# System Implementation

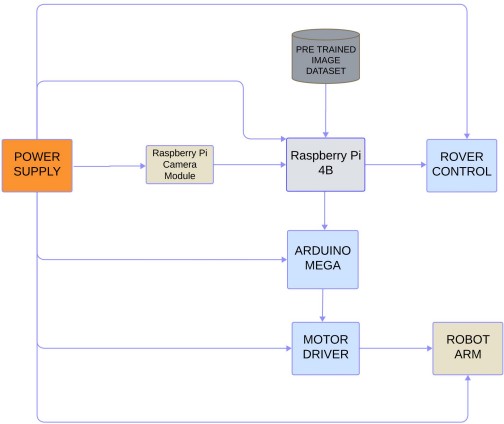


Figure 3.1: Block Diagram of proposed system

The figure 3.1 shows block diagram with a trained weed identification model, known as YOLO v5. It can distinguish cotton bolls from their environment. Using Roboflow, 1421 unique annotated photos were utilized to train the model on the Google Collab Plus platform. The robotic arm might be used to detect weed by deploying the trained model on a Raspberry Pi. In the future, this model will be combined with the robotic arm. The Raspberry Pi receives a video feed from the Raspberry Cam to start the model’s operation. After watching the video, the trained model delivers the user the marijuana that it has identified. The robot arm with six degrees of freedom then sprays the pesticide on the weed.

Why YOLOv5 -

* + 1. It is approximately 88% smaller (27 MB vs. 244 MB) than YOLOv4.
    2. It is about 180% faster (140FPS vs. 50FPS) than YOLOv4.
    3. It performs roughly the same as YOLOv4 on the same workload (0.895mAP versus 0.892mAP). Setting up YOLOv5 -

1. Step 1: Installation of image processing libraries (Open CV and Pillow), image classification libraries (Ten- sorflow and PyTorch).
2. Step 2: Download labelled dataset. Extract the zip file and move it to YoloV5 directory.
3. Step 3: Training of the model using YoloV5. YoloV5 generates a text file containing summary of accuracy and losses achieved at each epoch.
4. Step 4: Testing of the model using YoloV5.

The initial stage of training a model is data collection. This information was compiled by us from other sources. The internet provides the training dataset. Algorithms from artificial intelligence and machine learning are used to annotate images. Often, image annotation is carried out by human annotators who use an image annotation tool to label photos or tag pertinent information, such as assigning the proper classifications to different objects in an image. We created three classes for our model, as mentioned in the section on picture datasets. For simple tasks like classification and segmentation, pre-trained models are often accessible, and these models can be tailored to specific use cases with the help of Transfer Learning and little data. The data preprocessing section provides specifics regarding the Roboflow platform, which was used to finish the annotation. The data is sent to the Colab for model training after the platform’s annotation process; the data is split into three sections: test, validation, and train. Before exporting the dataset to colab, it is first expanded using the augmentation procedure—a detailed explanation of which may be found in the data pretreatment section. We use the Python train command with the required arguments to run the script and obtain the data, configuration files, and hyperparameters. We use the trained model to perform object detection in test set photos when testing on unseen data is finished. After object detection, the model’s accuracy and efficacy in object detection are evaluated, and metrics like mAP and precision are measured and examined. When a weed is detected, a box enclosing it appears, which appears to represent the degree of confidence. This level only reflects the degree of confidence that the object being recognised is or is not weed (for instance, a level of 0.7 indicates an accuracy rate of 70% in weed detection).

# Procedure for training the model

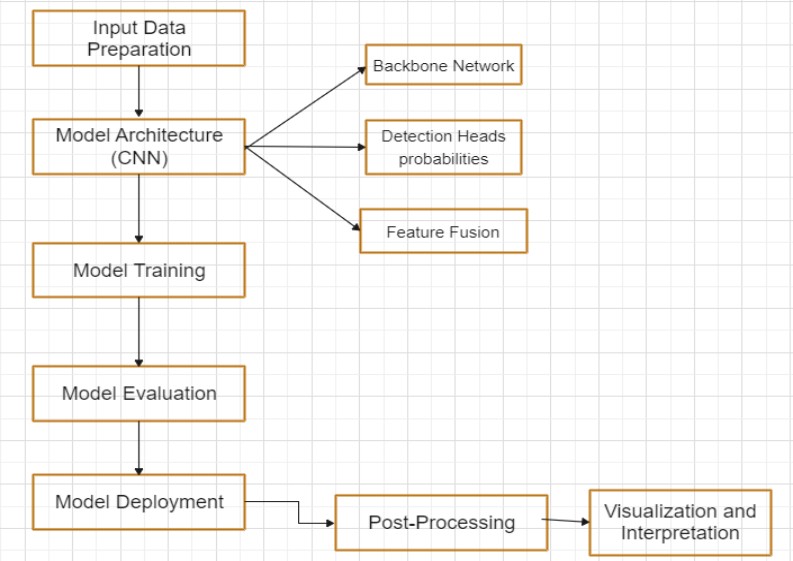


Figure 3.2: Flowchart of YoloV5

The figure 3.2 displays the flowchart of the YOLOv5 model.

* + 1. Preprocessing:

Data preprocessing means turning raw data into a format that computers and machine learning systems can understand and study. Preprocessing aims to enhance the data’s clarity, consistency and relevance.

In order to prevent irrelevant data from impeding the model’s effectiveness, data prep- aration is crucial for improving overall quality. Preprocessing improves the perfor- mance of both DL and ML models by remov- ing unnecessary input and only keeping relevant information, which results in more precise predictions and insights.

* + 1. Model Architecture (CNN):
       1. Backbone Network: YOLOv5 utilizes a CNN-based backbone network, typically based on the CSPDark- net architecture, to serve as the foundation for feature extraction. The backbone network consists of mul- tiple convolutional layers organized in a hier- archical manner. These layers are responsible for processing the input image and ex- tracting features that are relevant for object detection. By including CNNs in the backbone network, YOLOv5 can effectively capture hier- archical features at different levels of abstraction. This feature representation is needed for detecting objects of various sizes, shapes, and complexities.
       2. Feature Extraction: Within the backbone network, the input image undergoes processing through multi- ple convolutional layers. These convolutional layers apply filters to the input image, ex- tracting low-level features such as edges, textures, and colors. While the image progresses through the network, the spatial dimensions are downscaled through operations such as pooling or convolution with a stride greater than
          1. This down sampling process helps in capturing features at multiple scales and resolutions. The feature extraction process transforms the raw input image into a high-dimen- sional.
       3. Feature Fusion: Feature fusion enables the model to combine information from different levels of ab- straction, allowing for better localization and classification of objects across various spatial scales within the image. By fusing features from multiple scales, YOLOv5 can effectively capture contextual information and spatial relationships, leading to more accurate and robust object de- tection performance. Feature fusion plays a crucial role in enabling YOLOv5 to han- dle objects of different sizes and appearances within complex scenes effectively.
    2. Model Training:

When it’s time to teach the YOLOv5 model to recognize objects, we first get it ready by setting up its starting point. We can either start from scratch, or use some pre-trained model. Then, we show the model lots of images of the desired weed and tell it where these objects are by drawing boxes around them. This helps the model under- stand what objects look like and where they’re located in different pictures. As the model learns from these examples, it adjusts its predictions to get better at recognizing objects and figuring out what they are. During this training process, the model keeps trying to make its predictions closer to the real objects in the pictures. YOLOv5 also uses a tool called PyTorch, which is like a helper, to make this training process effi- cient and effective. With PyTorch, the model can learn quickly and accurately, be- coming better at object detection with each training session.

* + 1. Model evaluation:

Once we’ve finished teaching the YOLOv5 model to spot objects, it’s important to check how well it’s learned. We do this by testing it on a different set of images that it hasn’t seen before. This helps us see if the model can correctly identify objects in new situations. We use various tests to measure its performance, like the mean Average Precision (mAP), the precision, the recall, and the F1-score.

1. Mean Average Precision (mAP): It is a metric commonly used in object detection tasks to evaluate the accuracy of a model across multiple classes. It’s calculated by averaging Average Precision (AP) scores for each class.
2. Precision: Precision gauges how well the model predicts favorable outcomes. It is the proportion of accurate positive predictions to all of the model’s positive predictions.
3. Recall: Recall quantifies the model’s accuracy in identifying each pertinent instance of a class. It is the proportion of accurate positive predictions to all of the dataset’s real positive occurrences.
4. F1-score: The F1-score offers a fair assessment of a model’s performance since it is the harmonic mean of precision and recall.
5. Model Deployment:

Once the model’s performance meets the desired standards, it is ready for deployment to analyze new, unseen data. YOLOv5 facilitates deployment across diverse plat- forms, encompassing CPUs, GPUs, and edge devices. Subsequently, the trained model is applied to input images or videos to detect objects in real-time or through batch processing scenarios.

1. Post – processing:

Following the initial detection stage, post-processing methods are implemented to enhance the accuracy of identified bounding boxes and eliminate false positives. Key post-processing techniques include non- maximum suppression (NMS), which serves to remove redundant overlapping bounding boxes and preserve only the most reliable detections with high confidence levels.

NMS works:

Initial Detection: After running an object detection model on an input image, it pro- duces multiple bound- ing boxes around objects it believes to have detected.

Overlap Calculation: NMS calculates the overlap (or intersection over union, IoU) between pairs of bound- ing boxes. IoU measures how much two bounding boxes overlap with each other.

Suppression: NMS ranks the bounding boxes according to the confidence scores (likelihood of containing an object) for each class of objects. Then, starting with the bounding box that has the highest confidence score:

* It keeps this bounding box as a detection.
* It suppresses (removes) any other bounding boxes that have a high overlap (IoU) with this selected bound- ing box.

Iteration: NMS repeats this process until all bounding boxes have been considered.

1. Visualization and Interpretation: Visualizing the detection results involves displaying the identified objects by placing bounding boxes around them on input images or generating annotated videos. Interpreting the model’s predictions entails analysing how it performs, identifying its effectiveness, and recognizing its limi- tations. Through this analysis, we gain insights into the model’s strengths and weaknesses. If necessary, we can enhance the model by making iterative improvements based on the insights acquired from interpretation.

# Yolov5 Architecture

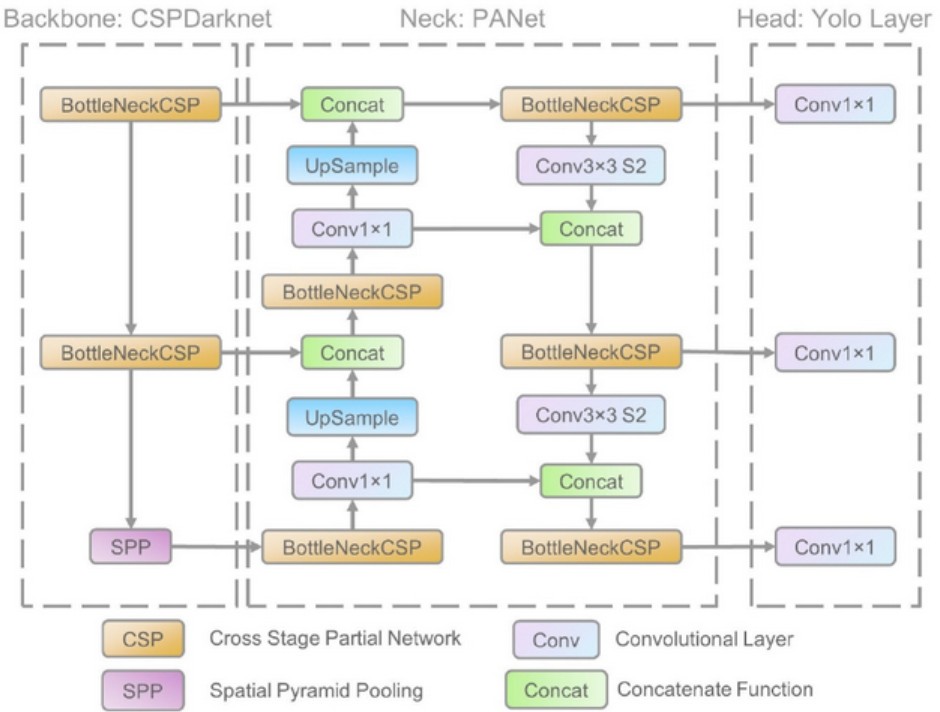


Figure 3.3: Network Architecture of YOLO V5

Figure 3.3 displays the Network architecture of the YOLOv5 Model

#### Backbone:

CSPDarknet: YOLOv5’s backbone is CSP-Darknet53. CSP-Darknet53 is essentially the Cross Stage Partial (CSP) network approach applied to the convolutional network Darknet53, which served as the foundation for YOLOv3. It is a deep network that uses dense and residual blocks to get over the vanishing gradient issue and allow informa- tion to flow to the deepest layers. Nevertheless, one advantage of having thick and residual blocks is the problem of recurring gradients. CSPNet helps to solve this problem by truncating the gradient flow.

#### Neck Of YOLOv5:

YOLOv5 significantly altered the model neck in two ways. First, a variant of Spatial Pyramid Pooling (SPP) was used, and the Path Aggregation Network (PANet) design featured the BottleNeckCSP.

#### Path Aggregation Network (PANet):

The feature pyramid network PANet was utilised in the earlier iteration of YOLO (YOLOv4) to enhance informa- tion flow and aid in the accurate localization of pixels during the mask prediction task. As seen in the network architecture diagram, the CSPNet technique has been used to this network in YOLOv5.

#### Spatial Pyramid Pooling (SPP):

The SPP block creates a fixed-length result by combining the data it receives from the inputs. Thus, it provides the advantage of significantly increasing the receptive area and dividing the most important context information without degrading network performance. This block was previously used in YOLO versions 3 and 4 to separate the most important features from the main body. But in YOLOv5 (6.0/6.1), network performance was increased by using SPPF, which is essentially an extra SPP block.

#### Head Of the Network:

The heads of YOLOv5, YOLOv3, and YOLOv4 are identical. It consists of three convolution layers that forecast scores, item classifications, and bounding box locations (x, y, height, and width). The following figure shows how, from previous iterations, the equation used to determine the target coordinates for the bounding boxes has changed.

*bx* = σ(*tx*) + *cx* (3.1)

*by* = σ(*ty*) + *cy* (3.2)

*bw* = *pw* ∗ *et*

*x*

*bh* = *ph* ∗ *et*

*h*

(3.3)

(3.4)

*bx* = (2 ∗ σ(*tx*) − 0.5) + *cx* (3.5)

*by* = (2 ∗ σ(*ty*) − 0.5) + *cy* (3.6)

*bw* = *pw* ∗ (2 ∗ σ(*tw*))2 (3.7)

*bh* = *ph* ∗ (2 ∗ σ(*th*))2 (3.8)

Equations 3.1,3.2,3.3 and 3.4 refers to equations used in YOLOv2 and YOLOv3. Equations 3.5,3.6,3.7 and 3.8 refers to equations used in YOLOv5.

# Design and Formulation

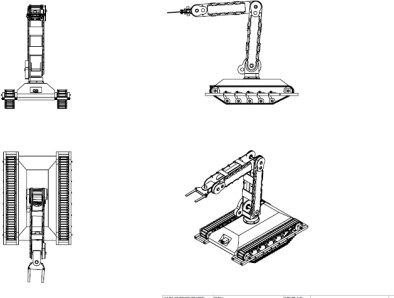


Figure 3.4: Schematic Diagram of Rover

The design schematic of the robot arm and rover designs are depicted in the figure 3.4 . The autonomous weed-killing rover consists of a wheeled platform that allows it to traverse fields. A hand extension on the rover allows it to reach weeds and spray pesticides on them. The rover can take detailed pictures of the weeds thanks to its sensors and cameras. Computer vision algorithms are used to process these images in real time, enabling the separation of weeds from other plant parts like leaves and stems. To precisely detect weeds, the computer vision system makes use of a range of visual cues, such as size, texture, and color. The rover’s control system determines the best path and location for the hand attachment to reach and spray weeds once they are found. The precise and cautious removal is made possible by the actuators in the hand extension. It is configured with clear paths and algorithms to remove obstacles so that it may cover the field effectively without running into obstacles.

# Objectives

* + 1. To choose an image processing model that is simple to use on any processing equipment with less specifi- cations for weed detection in real-world conditions.
    2. To train the YOLO v5 model to identify and detect weed from the plant when provided with a real-life video feed through a camera module.
    3. To design a robot arm to spray herbicides on an identified weed.
    4. To create a robotic arm that, when manually operated, can accomplish particular, coordinated tasks in three dimensions.

# Working Flow

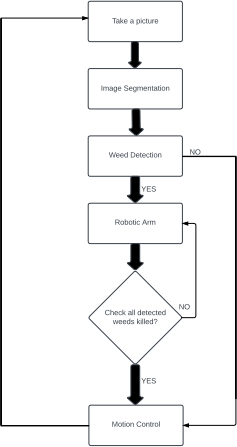


Figure 3.5: Flowchart

The figure 3.5 provides an explanation of the YOLO weed detection model.

#### Picture Division:

To distinguish the weeds from the surrounding landscape and other things, divide the photographs into parts. One popular method for teaching convolutional neural networks (CNNs) to recognize weed features is to train them using annotated datasets.

#### Recognition of weed:

Use a deep learning model for weed recognition. Teach the model to distinguish between different types of weeds. Features including color, texture, and shape can be used to identify and locate specific weeds within the segmented images.

#### Arm Control via Robots:

To translate the determined areas of marijuana into precise robotic arm movements, apply control algorithms. Consider feedback methods such as sensors on the robotic arm to account for changes in the locations of the plant and weed.

#### Weed Killing:

Provide a method for the end effector of the robotic arm, which is the part that interacts with the environment to use herbicides to kill the weed. Make use of a spray cannon that can be adjusted to suit different weed sizes and positions.

#### Iterative Method:

For each frame in the film or each plant in the field, repeat the entire process. To adapt to changing conditions, such as variations in plant growth and illumination, use real-time processing.

# Hardware Setup

The Weed Killing Robot has extensive hardware in the Robotic body where three different types of motors are used, these motors are controlled using a motor driver whose current rating is more than 5 Amperes.

## Hardware Components and Specifications

#### Arduino Mega

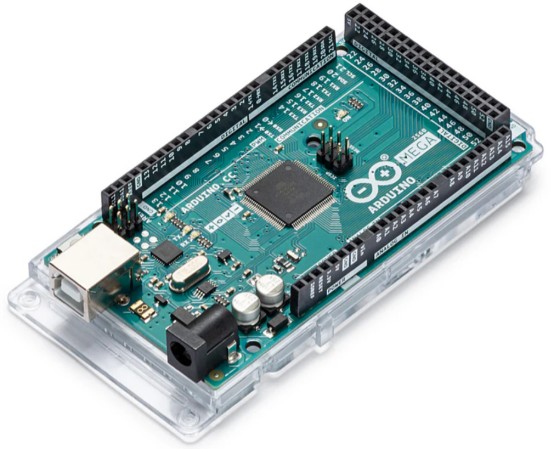


Figure 3.6: Arduino Mega

In Figure 3.6, an Arduino Mega is shown. A microcontroller board based on the ATmega2560 is called the Arduino Mega. With a lot more power than many other Arduino boards, this board is ideal for more complex projects. It can accommodate larger programs and more variables because to its enormous memory capacity. Potentiometer, temperature, and light. Among the many types of sensors that can be linked to the 16 analog inputs on the Arduino Mega are sensors. Furthermore, it contains 54 digital pins for input and output (I/O), of which 15 can be used as PWM (pulse width modulation) outputs. Its enormous number of I/O pins allows it to control multiple components at once, such as LEDs, relays, buttons, and screens. The clock speed is 16 MHz. It has a flash memory 256kb.

#### Servo Driver



Figure 3.7: 16-channel 12-bit PWM servo driver

A servo driver with 16 channels and 12 bits is shown in figure 3.7. The 16-channel 12-bit PWM servo driver is a small module that can precisely control multiple servo motors. It precisely positions servos using pulse- width modulation (PWM) signals, providing 12-bit resolution for more fluid motions. Accurate motion control is necessary for robotics and automation applications because it can handle is necessary—up to 16 servos independently.

It has a PCA9685 chip in it. There are 16 number of channels and the resolution is 12-bit. It has PWM frequency range of 40 Hz to 1000 Hz.

#### Servo Motor



Figure 3.8: Servo 995

Figure 3.8 shows a Servo 995. Standard-sized servo motor Servo 995 is renowned for its versatility and dependability. With a torque range appropriate for numerous robotic and hobbyist applications, it runs between 4.8 and 6 volts. Both model airplanes and animatronics use its widely compatible standard servo interface.

It has a rotating range of 180 degree. It is a DC motor which has 3 pins i.e. power,ground and signal.

#### Servo MG90s



Figure 3.9: Servo MG90s

Figure 3.9 shows a Servo MG90s. The compact design and low weight of MG90s and other small servo motors are well known. It is suitable for small-scale robotics and radio control projects since it can operate on 4.8 to 6 volts and has decent torque and speed performance. It has a dead bandwidth of 5 microsecond and a rotating range of 180 degree. It’s operating range is between -30 degree celsius and 60 degree celsius.

#### Servo SG90

The lightweight and small size of the SG90 micro servo makes it a cost-effective option. Its moderate torque and speed, along with its 4.8V to 6V operation, make it ideal for small mechanical operations, remote control vehicles, and lightweight robotics applications. Figure 3.10 displays a Servo SG90



Figure 3.10: Servo SG90

#### Power Supply



Figure 3.11: SMPS 12v, 5A

Figure 3.11 shows a SMPS of 12v and 5A. Switched mode power supply, or SMPS 12V, 5A, is a small, effective power source. With its steady 12 volt output and 5 amp maximum current capacity, it is appropri- ate for a variety of electronics and do-it-yourself applications. It’s switching strategy guarantees the least amount of heat generation and maximum energy efficiency.

#### Connecting Wires

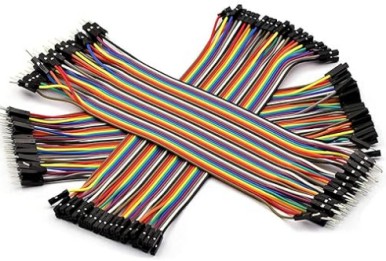


Figure 3.12: Jumper cable

Figure 3.12 displays Jumper cables which are used as connecting wires is also used to open or close circuit components. Two or more connecting points control its electrical circuit board. They are used to configure the motherboard and additional computer peripherals. Assume that your motherboard served as the intrusion detection system. A jumper can be used to switch it on or off. Jumper wires are electrical cables with connection points on both ends. In circuits, they are used to connect two points together without the use of solder. Jumper wires can be used to make changes to or diagnose problems with a circuit.

# Software setup

#### Initial Setup

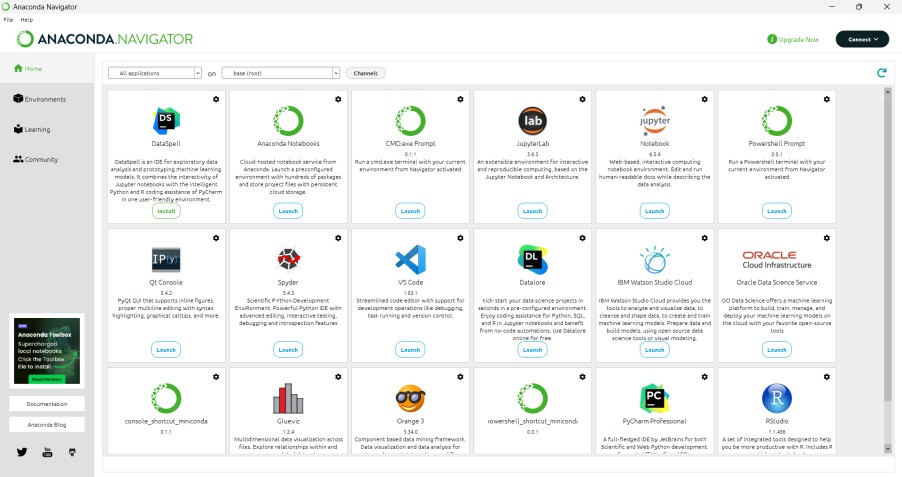


Figure 3.13: Anaconda Dashboard

Figure 3.13 shows an Anaconda Dashboard. First, install Anaconda on your laptop and create an environ- ment. It is an open-source package and environment management system that runs on Windows, Linux, and macOS. Conda swiftly installs, launches, and upgrades packages and dependencies. It also simplifies the process of creating, saving, loading, and switching between environments on your own computer. It is capable of packaging and delivering software for any language, although Python projects are its primary focus.

#### Jupyter Notebook

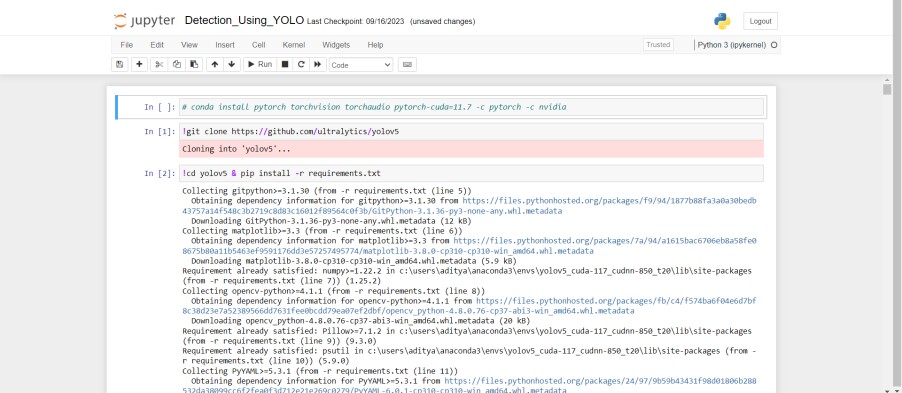


Figure 3.14: Jupyter Dasboard

The above figure 3.14 displays a Jupyter Dashboard from Jupyter Notebook. It is a popular open-source web application that allows you to create as well as share documents with live code, equations, pictures, and narrative text. Data science and ML make heavy use of it for interactive computing and prototyping.

Numerous packages and frameworks are available for Jupyter Notebook object identification, however one of the most widely used solutions for Python is TensorFlow or PyTorch. Add the file locations for Yolov5, Cuda, and Jupyter Notebook to the environment that has been constructed to guarantee that all processing is done on the GPU.

Cudnn. Using Jupyter Notebook which is an open-source web tool, data scientists may create and share documents with live code, equations, and other multimedia features.

#### PyTorch

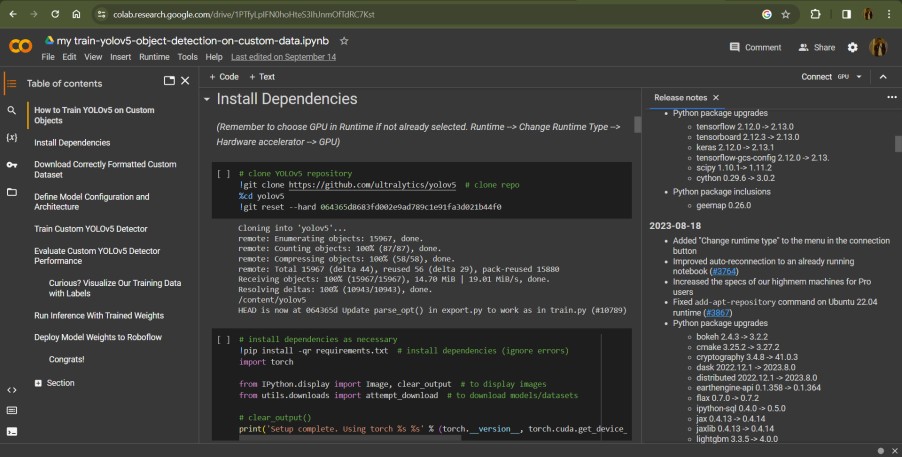


Figure 3.15: Google Colab

Figure 3.15 shows the dashboard of Google Colab. Install the appropriate version of Pytorch on the com- puter. Deep learning models, a type of machine learning commonly used for tasks like image identification and language processing, may be created with PyTorch, a feature-rich framework. Because it’s built in Python, most machine learning developers find it to be rather easy to comprehend and use.

#### Arduino IDE

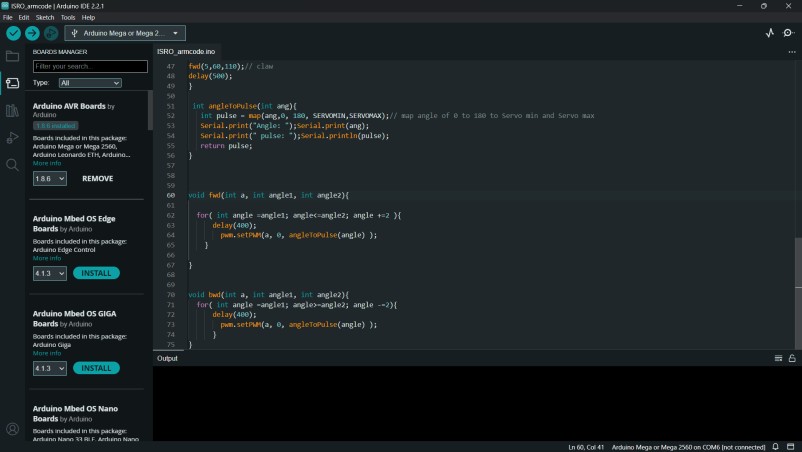


Figure 3.16: Arduino IDE

The Arduino Integrated Development Environment (IDE) is the software used for writing, modifying, and uploading code to Arduino boards. For developing sketches or programs using setup() and loop(), it features a code editor with syntax highlighting. The support for many built-in and external libraries provided by the IDE allows users to select the appropriate board and communication port for their applications. It’s crucial to verify, assemble, and upload code to the Arduino hardware. In order to aid in learning, it also provides a serial monitor for easier connection between the board and the PC in addition to a ton of tutorials and sample sketches. Its open-source design and compatibility for Windows, macOS, and Linux make it a well-liked and flexible choice.The above figure 3.16 dispalys the Arduino IDE used

#### Database

The internet was used to gather the image dataset, which was taken from several surrounding farms. The images were taken from various angles, positions, and backdrops to ensure that the model’s testing would be sufficient for a variety of settings. The gathered internet dataset was made available by Roboflow. The dataset contains about 12000 images. The preparation portion contains the 16 classes, or 16 groups, into which the Roboflow dataset is split. Prior to preprocessing, the dataset’s image resolution was 640x640 pixels.

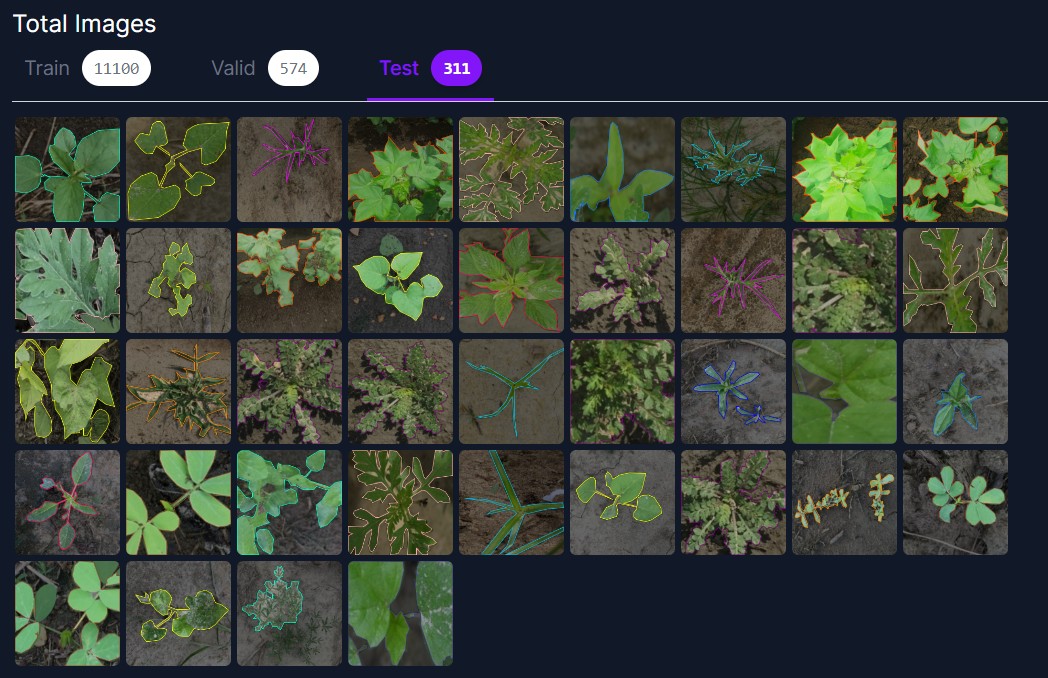


Figure 3.17: Sample Images

The above figure 3.17 shows our dataset. About 11000 images are in the training set, 574 in the validation set and 311 images for testing.

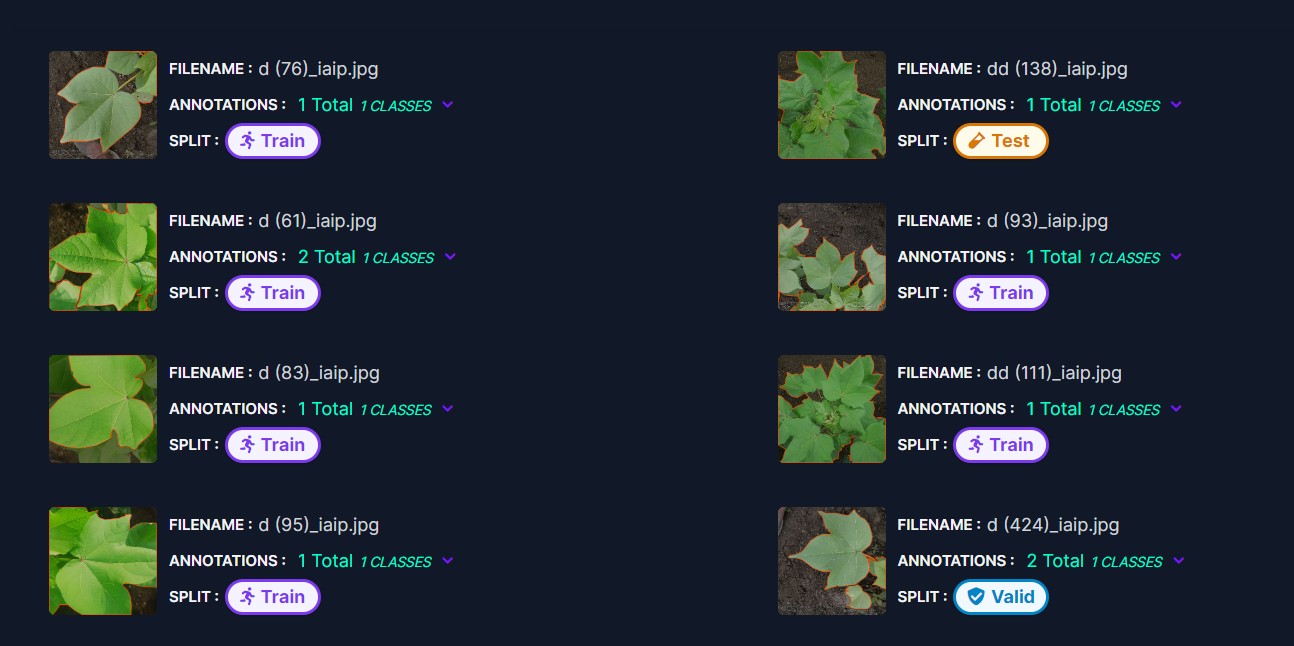


Figure 3.18: Sample Images

Figure 3.18 shows the different varieties of weeds we annotated. Data preprocessing transforms unpro- cessed data into a format that computers and machine learning can understand and analyse as part of the

data analysis process. We annotated the photographs by putting an anchor box around the tick that has to be detected in order to obtain raw data for our investigation. There are different varieties of weed here. There are sixteen categories for the weeds. The next stage is data augmentation after each image has been annotated. The photos are enhanced, and then they are turned 90 degrees either clockwise or anticlockwise. They can also be turned horizontally or vertically. To make the image 640x640 pixels, it is scaled. The augmented photographs have an auto contrast adjustment ranging from +12% to -12%. The image has been stripped of any filters and cropped with a minimum zoom of 0% and a maximum zoom of 25%. Also, there is up to 0.4% blurring in the images.

**Chapter 4**

**Results and Discussions**

# Software Results

In the field of machine learning, specifically with regard to statistical classification issues, a confusion matrix—also called an error matrix—is a specific table configuration that makes the performance of an algorithm easier to visualise. This approach is usually associated with supervised learning; in unsupervised learning, it is often called a matching matrix.

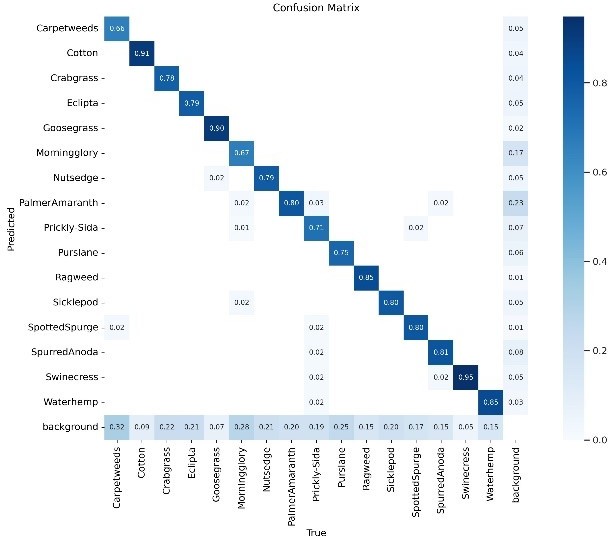


Figure 4.1: Confusion Matrix Graph

Figure 4.1 displays a confusion matrix that provides the true values for 16 different types of weeds along the X-axis. For carpetweed, the genuine figure is 66%, and for crabgrass, it is 78%. The percentages of Eclipta, Goosegrass, Morning Glory, Nutsedge, Palmer Amarnath, Prickly-Sida, Purslane, Ragweed, Sicklepod, Spotted Spurge, Spurred Anoda, Swinecress, and Waterhemp are 79%, 90%, 71%, and 75%, respectively.

The confusion matrix shows us that the genuine background value is only 3%, indicating that backdrop has no effect on weed detection.

Table 4.1: Confusion Matrix Table

|  |  |
| --- | --- |
| **Class** | **Correct Predictions** |
| Carpetweeeds | 0.66 |
| Cotton | 0.91 |
| Crabgrass | 0.78 |
| Eclipta | 0.79 |
| Goosegrass | 0.90 |
| Morningglory | 0.67 |
| Nutsedge | 0.79 |
| PalmerAmaranth | 0.80 |
| Picky-Sida | 0.71 |
| Pursiane | 0.75 |
| Ragweed | 0.85 |
| Sicklepod | 0.80 |
| SpottedSpurge | 0.80 |
| SpurredAnoda | 0.81 |
| Swinecress | 0.95 |
| Waterhemp | 0.85 |

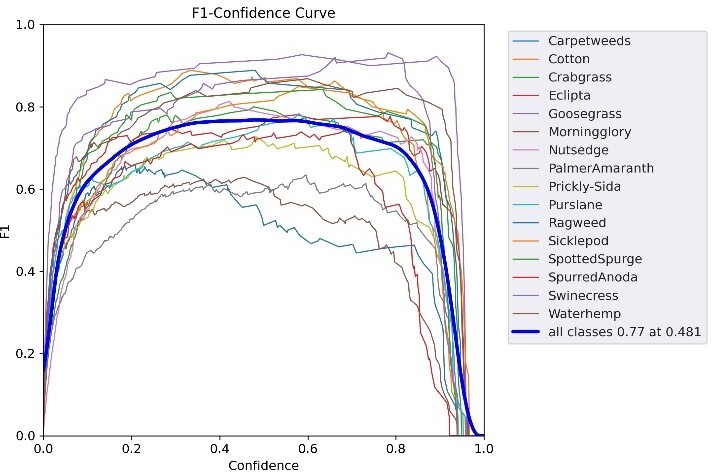


Figure 4.2: F1-Confidence Curve

The confidence value of 0.481 in Fig. 4.2, which uses the F1 curve as a reference for all classes, maximises both recall and precision. In many cases, a higher confidence value is better. The confidence value for this model needs to be 0.75, which is not too far from its maximum confidence of 0.77.

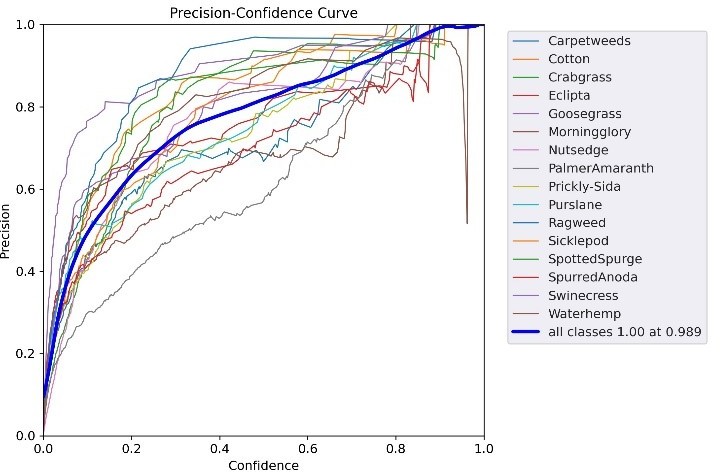


Figure 4.3: Precision-Confidence Curve

The Fig 4.3 represents the relationship between the model’s confidence and precision.

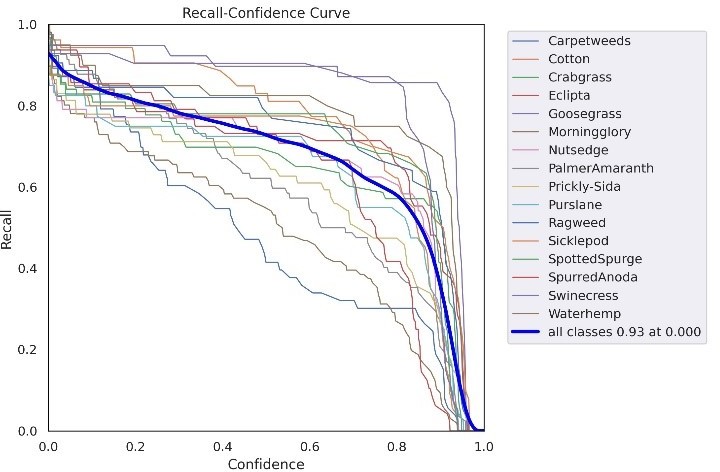


Figure 4.4: Recall-Confidence Curve

Fig 4.4 shows the relation between trained model’s recall and confidence.

* + 1. Accuracy measures overall correctness which is calculated by the formulae:

*Accuracy* = (*TP* + *TN*)/*Tn p* (4.1)

* + 1. Precision is specific to a category, given by:

*Precision* = *TP*/((*TP* + *FP*)) (4.2)

* + 1. Recall talks about the successful detection of a specific category, by using the formulae:

*Recall* = *TP*/((*TP* + *FN*)) (4.3)

* + 1. F1 score balances precision and recall:

*F*1*S core* = 2 *Precision* ∗ *Recall*

∗

*Precision* + *Recall*

And accordingly, the output is classified and represented in the result form.

(4.4)

# Hardware Results

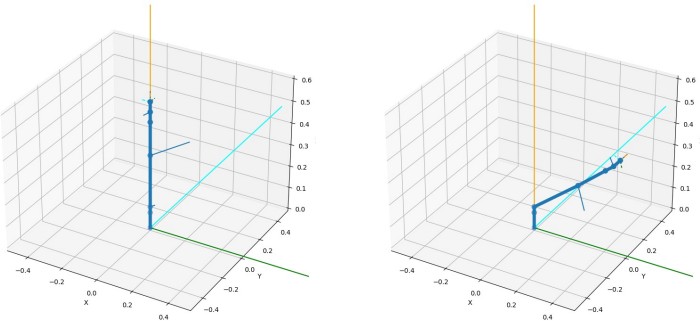


Figure 4.5: Simulation results for the position of the robotic arm

Figure 4.5 displays the simulation results for the position of the robotic arm. Regarding the aforementioned graphic, the location of the robotic arm was represented by a Python program that was ran on a Jupyter Notebook to produce the simulation results. The software was given the 3D coordinates in order to deter- mine the position and necessary angles for the arm’s movement. The program then computed the angles for each joint, which led to the end effector’s placement at the specified 3D coordinates.

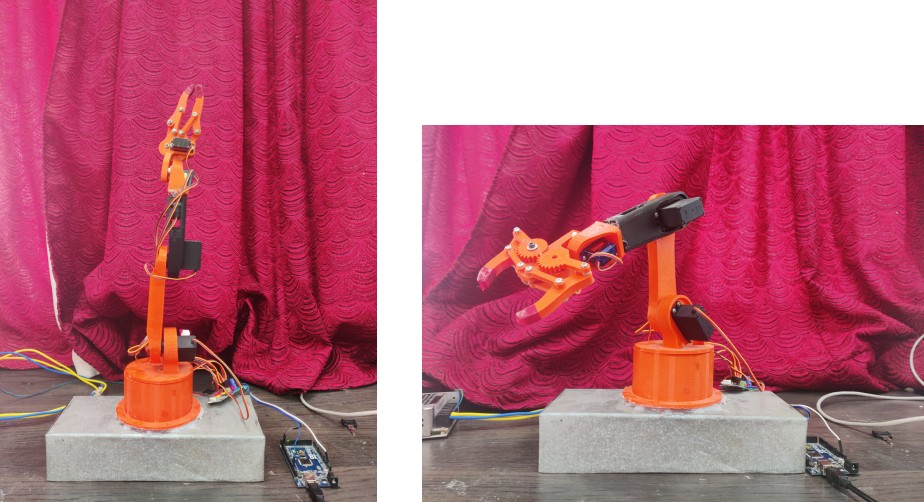


Figure 4.6: Results of working of the robotic arm

Figure 4.6 shows the results of the working robotic arm For real-time servo control, the Python code that produced the simulation results was converted to Embedded C and uploaded to an Arduino Mega. Based on the computation of the angles for each joint, the micro controller was able to operate the servos in accordance with the given coordinates. The outcomes of the simulation and the arm’s actual operation were comparable. The model is displayed in the figure above.

# Summary

With OpenCV at its core, the weed-killing rover offers a cutting-edge approach to precision farming. The rover uses computer vision to autonomously locate and target weeds in agricultural fields, thereby lowering the environmental impact and herbicide usage. Accurate weed detection is made easier by the integration of OpenCV, and the rover’s capabilities are further enhanced by continuous improvements in machine learn- ing. Operating independently, the rover navigates fields and applies herbicides as efficiently as possible. Full field analysis is provided by multi-sensor integration, which includes infrared and hyperspectral cameras. The project, which focuses on weed distribution and herbicide effectiveness, emphasizes data analytics for well-informed decision-making. Regulatory compliance, cost effectiveness, and environmental sustainabil- ity are important factors that make the weed-killing rover a promising tool for effective and long-term weed control in agriculture.

**Chapter 5**

**Advantage and Applications**

# Advantages

The advantages of the proposed system can be summarized as follow:

* + 1. Weed-killing robots can precisely target and spray herbicides only on weeds, reducing chemical usage and minimizing the environmental impact.
    2. By automating the process, these robots reduce human exposure to herbicides and improve safety for farm workers.
    3. Effective weed control helps crops thrive by reducing competition for resources like water, sunlight, and nutrients.
    4. OpenCV allows the robot to make real-time decisions based on the current field conditions, adapting to changes in weed distribution and density.
    5. By automating weed detection and herbicide application, the risk of human exposure to herbicides is significantly reduced, enhancing safety for farm workers.

# Applications

The proposed system can be employed at variety of applications such as :

* + 1. Weed Identification
    2. Agriculture
    3. Maintaining Sports Fields
    4. Gardening Purposes

**Chapter 6**

**Conclusion and Future Scope**

# Conclusion

The expected conclusion of the problem statement is to develop a robust system which can precisely detect and eliminate weeds using herbicides. The use of OpenCV in weed killing rover shows great potential for combining robotics and computer vision to effectively and precisely to kill weeds. The rover can analyze real-time visual data using OpenCV’s image processing and accurately differentiate between crops and weeds. The rover has the ability to independently navigate through the fields using image recognition algorithms to differentiate between required plants and invasive weeds. By using a targeted approach, we can minimize the use of broad-spectrum herbicides, which in-turn reduces impact of the harmful herbicides on the environment and encourages sustainable agricultural practices. By integrating advanced sensors and cameras with flexibility of OpenCV, rover will be able to travel through diverse terrains and weather conditions.

# Future Scope

* + 1. **Precision Agriculture**: With the integration of the weed-killing rover into precision agricultural sys- tems, weeds in crop fields may be identified and targeted using computer vision. Herbicide use is minimized and the environmental impact is decreased with this focused strategy.
    2. **Machine learning integration**: Using machine learning methods in conjunction with OpenCV can improve the rover’s capacity to identify various weed species and adjust to shifting environmental circumstances. This can maximize the administration of pesticides and increase the precision with which weeds are detected.
    3. **Autonomous Navigation**: In the future, the rover’s autonomy could be improved, enabling it to tra- verse fields, recognize weeds in real time, and decide where and how much herbicide to spray. As a result, less human intervention would be required.
    4. **Environmental Monitoring**: Sensors to keep an eye on weather patterns, soil health, and moisture levels might be installed on the rover. Sustainable agriculture can be supported and weed-killing tactics can be optimized with the help of this knowledge.
    5. **Cost Reduction and E**ffi**ciency**: Ongoing technological advancements may result in cheaper produc- tion and upkeep costs for these rovers. Farmers may save money overall if they can control weeds more effectively.

**Chapter 7**

**References**

1. X. Jiang et al., ”Graph Convolutional Network for Weed Species Classification,” IEEE Transactions on Agricultural Engineering, vol. 8, no. 1, pp. 45-52, 2020.
2. S. Chavan et al., ”AgroAVNET: A Hybrid CNN Model for Crop and Weed Classification,” IEEE Journal of Agricultural Robotics, vol. 5, no. 2, pp. 87-94, 2018.
3. A. Ramirez et al., ”Comparative Analysis of Deep Learning Architectures for Weed Detection,” IEEE Transactions on Sustainable Agriculture, vol. 11, no. 3, pp. 176-183, 2020.
4. Y. Hu et al., ”Graph Weeds Net: A Graph-Based Deep Learning Model for Weed Classification,” IEEE Journal of Precision Agriculture, vol. 6, no. 4, pp. 210-218, 2020.
5. M. Fawakherji et al., ”U-Net Architecture for Semantic Segmentation of Crop and Weed Plants,” IEEE Transactions on Agricultural Imaging, vol. 3, no. 1, pp. 33-39, 2019.
6. P. Osorio et al., ”Comparison of RPN Models for Weed Detection in Agricultural Fields,” IEEE Trans- actions on Sustainable Computing, vol. 5, no. 2, pp. 87-94, 2020.
7. J. Lottes et al., ”Fully Convolutional Networks for Pixel-Wise Semantic Segmentation of Crop and Weed Plants,” IEEE Journal of Precision Agriculture, vol. 7, no. 3, pp. 150-157, 2021.
8. H. Zhang et al., ”Weed Detection in Rice Fields Using Multispectral Imaging and Convolutional Neu- ral Networks,” IEEE Transactions on Sustainable Agriculture, vol. 9, no. 4, pp. 210-217, 2022.
9. K. Lee et al., ”Deep Learning-Based Weed Detection System for Precision Farming,” IEEE Journal of Agricultural Engineering, vol. 4, no. 3, pp. 120-127, 2019.
10. Y. Wang et al., ”Feature Fusion-Based Weed Detection Using Convolutional Neural Networks,” IEEE Transactions on Sustainable Computing, vol. 8, no. 2, pp. 80-88, 2021.
11. Badkhal, T., Sharma, R., Sonu, I., Patil, S., and Mapari, R. Automatic Weed Killing Robot for Agri- culture Purpose and Insect Killing.
12. Scholz, Christian, Maik Kohlbrecher, and Arno Ruckelshausen. ”Camera-based selective weed control application module (“Precision Spraying App”) for the autonomous field robot platform BoniRob.” International Conference of Agricultural Engineering, Zurich. 2014.
13. Wu, Xiaolong, et al. ”Robotic weed control using automated weed and crop classification.” Journal of Field Robotics 37.2 (2020): 322-340.
14. Michaels, Andreas, Sebastian Haug, and Amos Albert. ”Vision-based high-speed manipulation for robotic ultra-precise weed control.” 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015.
15. Gai, Jingyao, Lie Tang, and Brian L. Steward. ”Automated crop plant detection based on the fusion of color and depth images for robotic weed control.” Journal of Field Robotics 37.1 (2020): 35-52.
16. Badhan, Siddhesh, et al. ”Real-Time Weed Detection using Machine Learning and Stereo-Vision.” 2021 6th International Conference for Convergence in Technology (I2CT). IEEE, 2021.
17. Hasan, ASM Mahmudul, et al. ”A survey of deep learning techniques forweed detection from images.” Computers and Electronics in Agriculture 184(2021): 106067.
18. Wang, Aichen, Wen Zhang, and Xinhua Wei. ”A review on weed detectionusing ground-based ma- chine vision and image processing techniques.” Computers and electronics in agriculture 158 (2019): 226- 240.
19. T. Badkhal, R. Sharma, I. Sonu, S. Patil, and R. Mapari, “Automatic weedkilling robot for agriculture purpose and insect killing.”
20. Y. Du, B. Mallajosyula, D. Sun, J. Chen, Z. Zhao, M. Rahman, M. Quadir,and M. K. Jawed, “A low-cost robot with autonomous recharge and navigation for weed control in fields with narrow row spacing,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3263–3270, IEEE, 2021.
21. Fue, K. G., et al. ”Visual row detection using pixel-based algorithm and stereo camera for cotton- picking robot.” 2019 ASABE Annual InternationalMeeting, Boston, Massachusetts, 2019.
22. E. Barnes, G. Morgan, K. Hake, J. Devine, R. Kurtz, G. Ibendahl, A. Sharda, G. Rains, J. Snider, J. M. Maja, et al., “Opportunities for robotic systems and automation in cotton production,” AgriEngineer- ing, vol. 3, no. 2, pp. 339–362, 2021.
23. M. H. Saleem, J. Potgieter, and K. M. Arif, “Automation in agriculture bymachine and deep learning techniques: A review of recent developments,” Precision Agriculture, vol. 22, pp. 2053–2091, 2021.
24. G. A. Kumar, M. Ramya, et al., “Wedding of robots with agriculture,” in 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT’12), pp. 1–4, IEEE, 2012.
25. H. Nejati, Z. Azimifar, and M. Zamani, “Using fast fourier transform forweed detection in corn fields,” in 2008 IEEE International Conference onSystems, Man and Cybernetics, pp. 1215–1219, IEEE, 2008.
26. https://universe.roboflow.com/annotation-3jpwp/m.tech
27. https://iq.opengenus.org/yolov5/ M.TECH Object Detection Dataset (v15, 2024-02-22 3:41pm) byAN- NOTATION (roboflow.com)

**Appendix A**

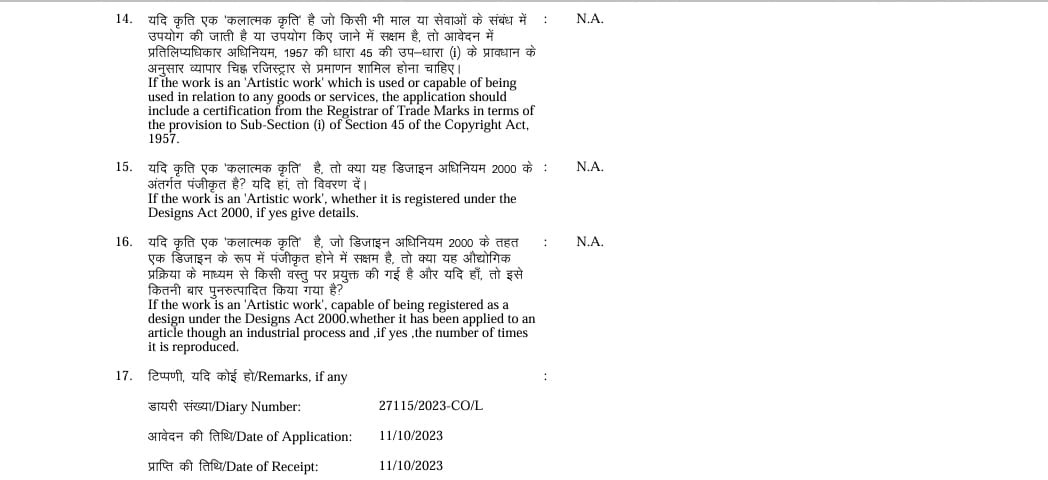
**Project Outcomes**

# Plagriasim Report

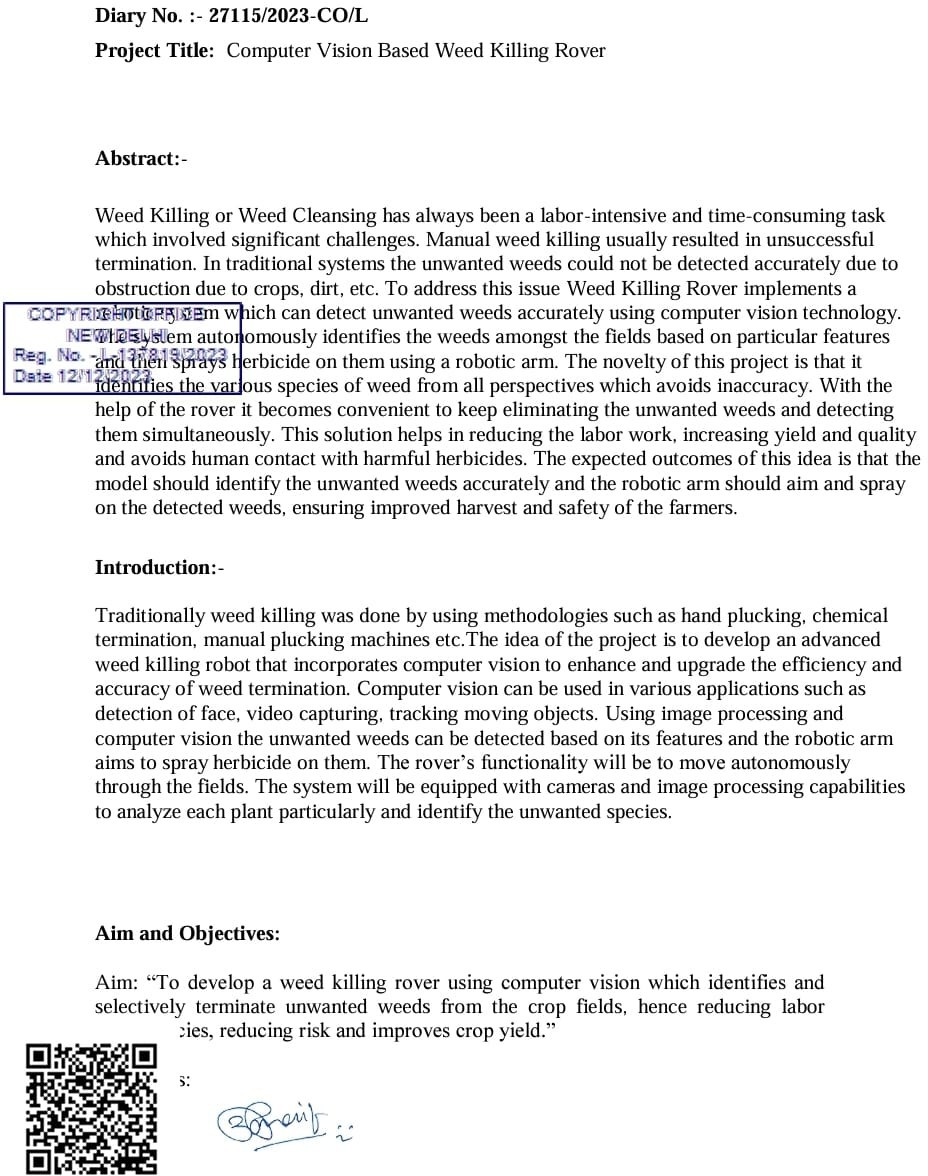
# Copyright Certificate and Work Uploaded

## Copyright Certificate

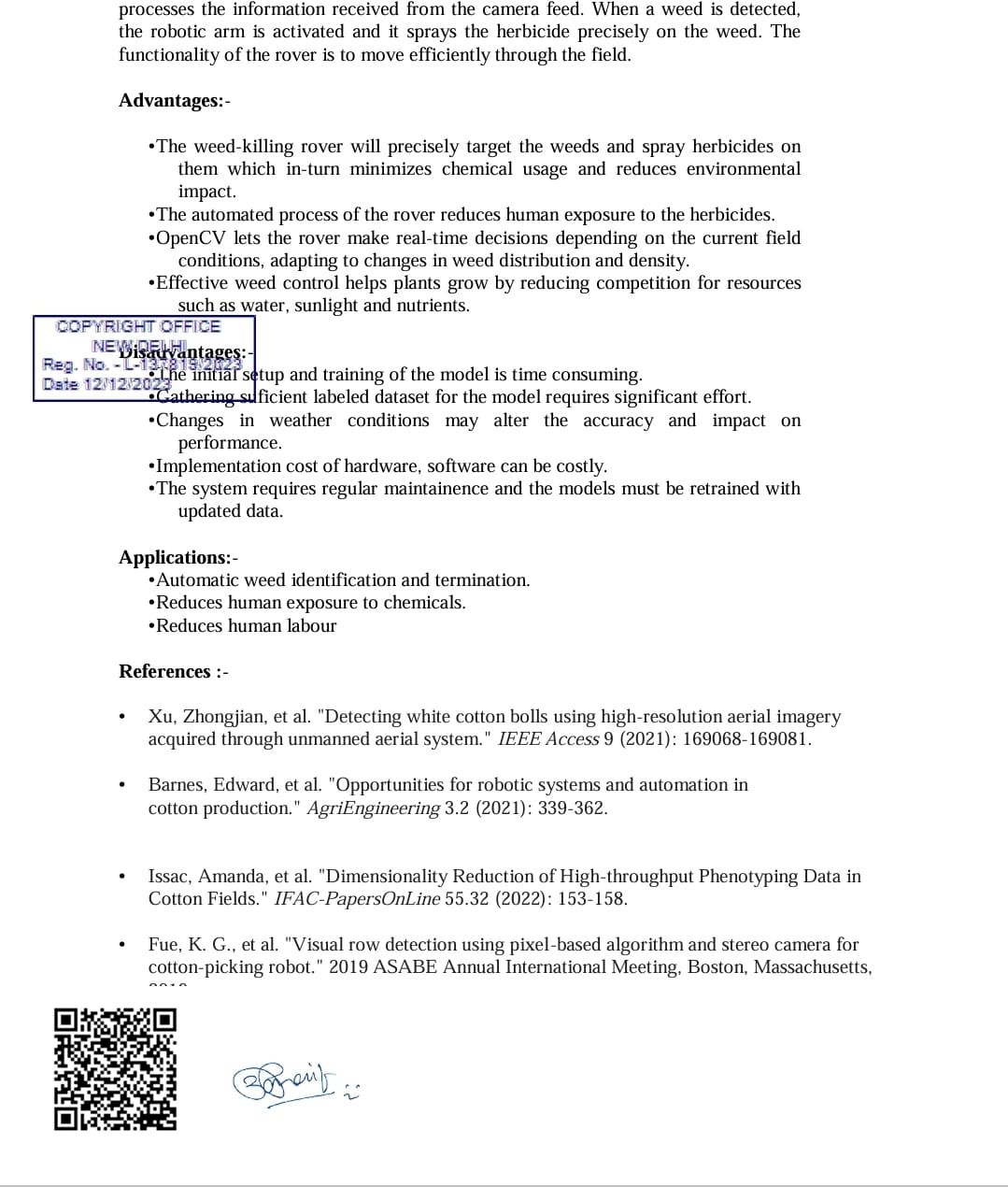


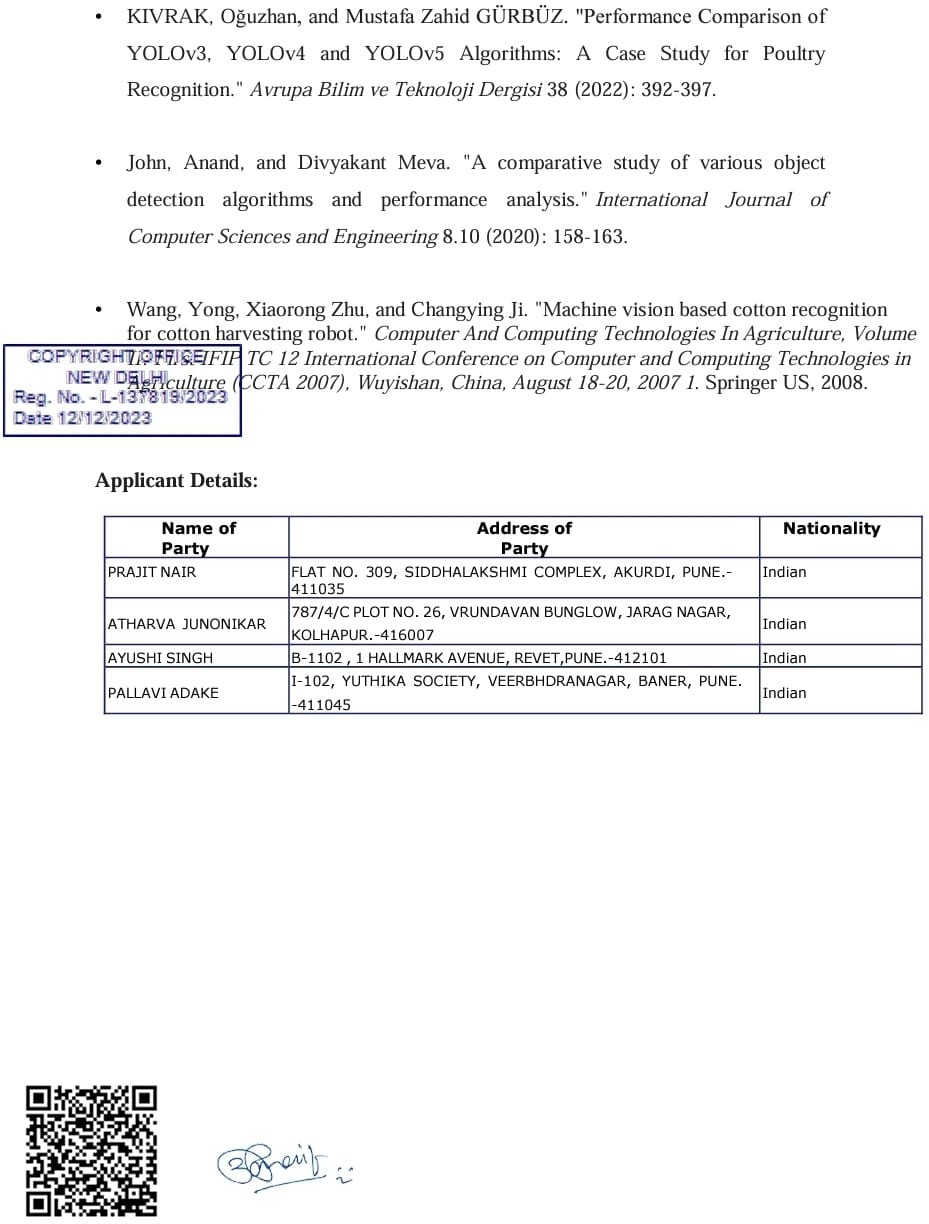


## Synopsis

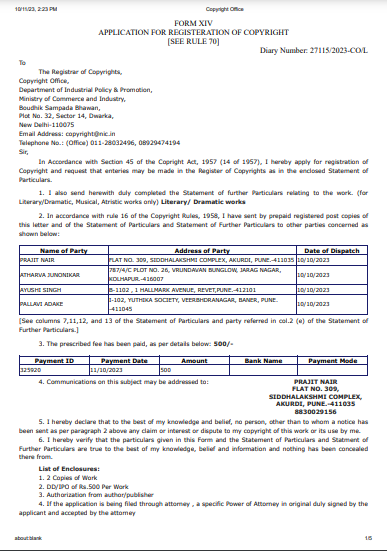


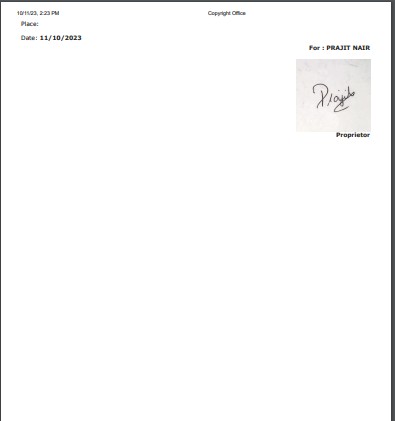


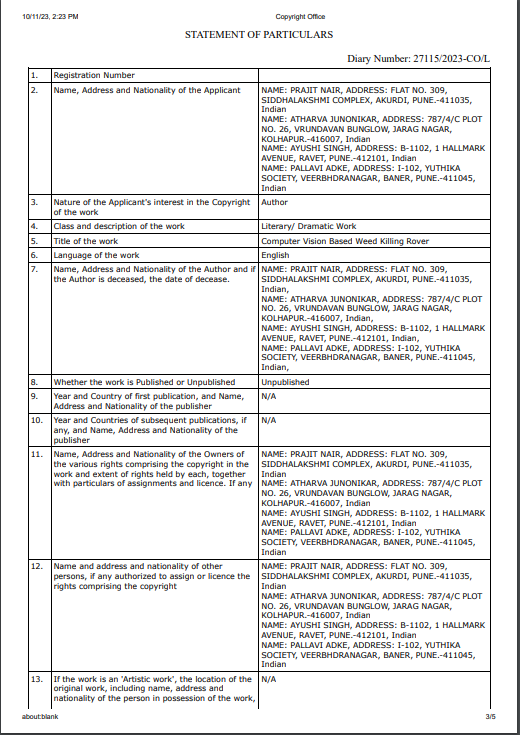




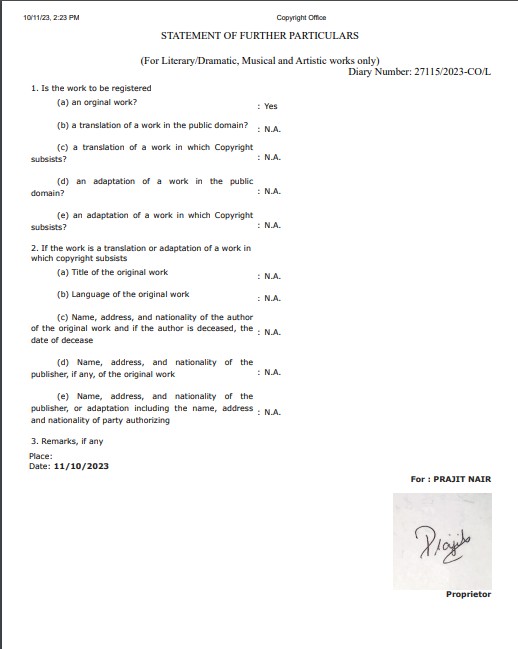
## Form 14











* 1. **Project Competition Participation Certificates**





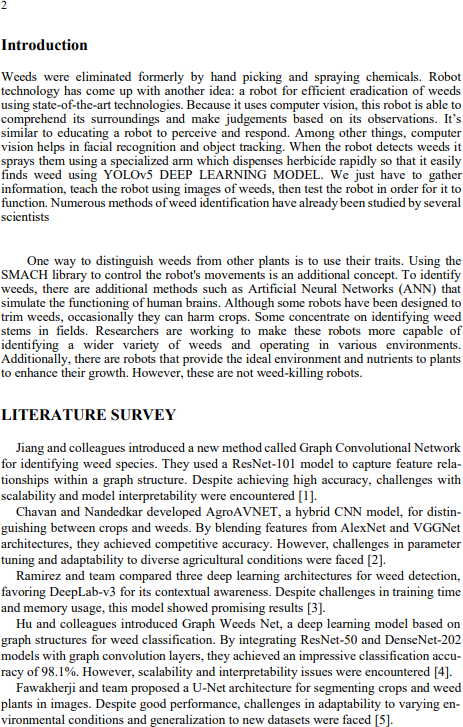


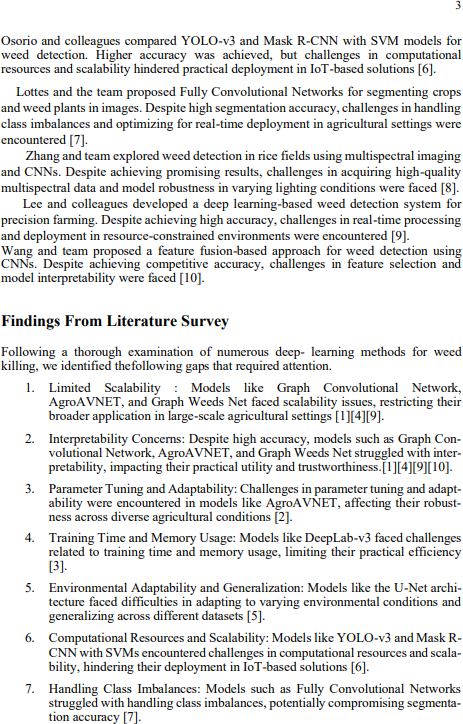


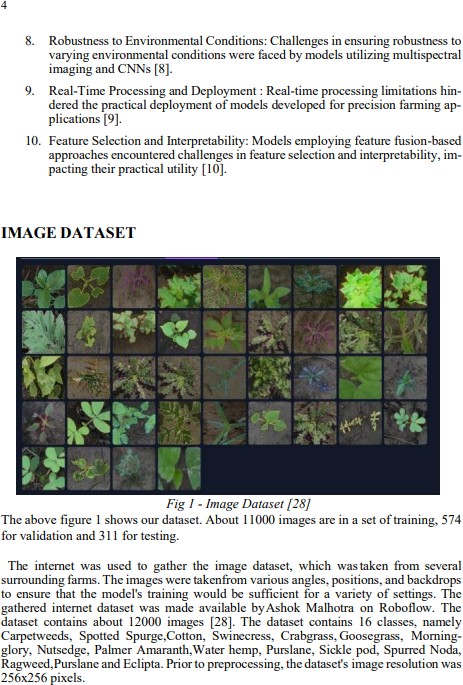


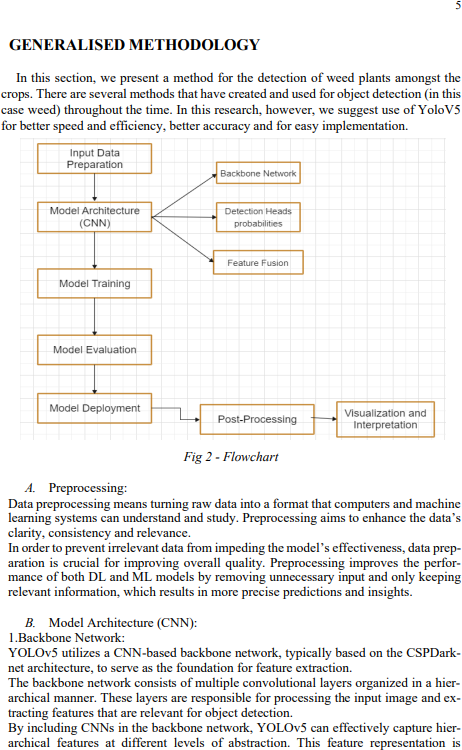
* 1. **Publication Document**
     1. **Publication Work**

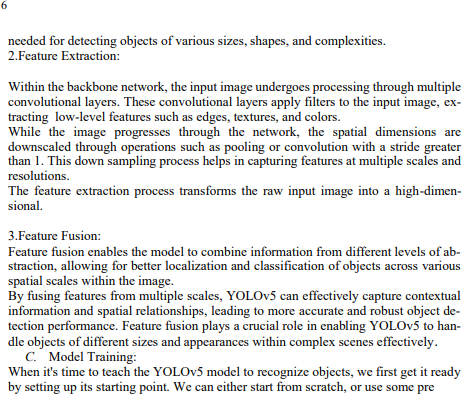


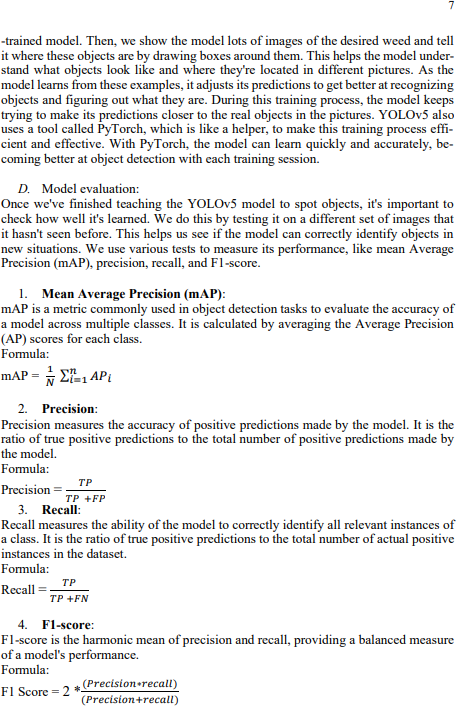












* + 1. **Publication Copyright Mail**

