International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

WEED DETECTION AND EXTRACTION ROBOT

MR. JASON ANTONIO DE SOUZA
ELECTRONICS AND COMMUNICATION ENGINEERING
AGNEL INSTITUTE OF TECHNOLOGY AND DESIGN
ASSAGAO, INDIA
20lec03@aitdgoa.edu.in

MRS. PRAMILA VOLVOIKAR
ELECTRONICS AND COMMUNICATION ENGINEERING
AGNEL INSTITUTE OF TECHNOLOGY AND DESIGN
ASSAGAO, INDIA
pvo@aitdgoa.edu.in

MS. NEEDHI GAONKAR
ELECTRONICS AND COMMUNICATION ENGINEERING
AGNEL INSTITUTE OF TECHNOLOGY AND DESIGN
ASSAGAO, INDIA
ng@aitdgoa.edu.in

***Abstract*—Agriculture has been a major part of our lifestyle since decades. Life cannot exist without it. It is the main source of livelihood for many people worldwide. It is the back bone of the economic system and plays a critical role in the entire life of a given economy. Weeds are the most important biotic constraints to agriculture production. In general, weeds present the highest potential yield loss to crops. Weed control costs a fortune for farmers and use of herbicides for the same leads to high detriment of soil fertility. There is an acute urgency to find an alternate method to locate and deal with these weeds without using any weed killers. In this paper we are showcasing the implementation of a more efficient way to detect weeds in farms and fields.**

***Index Terms*—component, formatting, style, styling, insert**

I. INTRODUCTION

Detecting weeds is an extremely tedious, labour-intensive and pricey job. It has been one of the major reasons for loss of crop production. The objective of this project is to obtain a formula for building up a weed detection system by employing binary classifica tion and integrating it with a motor driven vehicle for efficient real time weed detection innocuously. Our main goal is to provide an eco-friendly, cheap and precise device solution which will increase the efficiency of the weed locating process and reduce man-power.

II. PROTOTYPE DESCRIPTION

The weed detection model is written in Python programming language, developed using TensorFlow on Jupyter Notebook platform and trained with a sufficient amount of data. The model is then integrated with a camera to perform real time object detection, which will provide live visuals of the field as an input to the model. The model will then use this input to detect weeds. Performing real time image processing, using an image recognition algorithm, the system will classify the plant either as weeds or crops through binary classification. To do so, the system will display boxes around the plant on the screen. A green box will indicate crops whereas a yellow box will indicate weeds. It will also display the accuracy of detection on top of these boxes in percentage. A motor driven vehicle will enter the field and move around the field, directed by an electronic speed controller. The camera will be attached to this vehicle. It will run on a rechargeable battery. This will make detection of weed easier

A. Direct Current Motor

A DCmotor is a rotary electrical motor that converts direct current (DC), i.e. electrical energy into mechanical energy by creating a magnetic field which is powered by a direct current. The capacity of DC motors to precisely control their speed, which is essential for industrial gear, is one of the reasons they are favoured over other types of motors. The ability of DC motors to instantly start, stop, and reverse is crucial for managing the functioning of production machinery.

The Johnson geared motor is renowned for its small size and enormous torque. x3 the torque of a centre shaft or side shaft geared motor. The motor has an off-centred shaft, a metal bushing for wear resistance, and a metal gearbox

Fig. 1. DC MOTOR

B. Electronic Speed Controller

An electrical circuit known as an electronic speed controller (ESC) controls and regulates an electric motor's speed. It might also offer dynamic braking and motor reversing. In radio-controlled models that are powered by electricity, tiny electronic speed controllers are employed. Systems are also present in full-size electric vehicles to regulate the speed of the drive motors.

Fig. 2. ELECTRONIC SPEED CONTROLLER

C. BATTERY

A lithium polymer battery is a rechargeable lithium-ion battery that uses a polymer electrolyte rather than a liquid electrolyte as the electrolyte. This electrolyte is made up of high conductivity semisolid (gel) polymers. These batteries are employed in applications where weight is important, such as mobile devices, radio-controlled aircraft, and some electric vehicles. They offer better specific energy than other lithium battery types.

Fig. 3. LIPO BATTERY

D. RADIO CONTROL

Radio control (RC) is the use of control signals transmitted by radio to remotely control any device. Flysky transmitter and receiver (fsi6). It is a two way RX providing telemetry from the sensors on the FC and to the display on the TX. The FS-i6 is a digital proportional radio control system that runs on the 2.4GHz global ISM band, making it easy for use everywhere in the world. Furthermore, it comprises of AFHDS 2A (Automatic Frequency Hopping Digital System Second Generation) technology that is used in the transmitter, which not only makes communication between user and vehicle secure but also lowers the transmitter's power consumption.

Fig. 4. FLY RADIO CONTROL

III. SOFTWARE REQUIREMENTS

A. Project Jupyter

For interactive computing in several programming languages, Project Jupyter offers open-source software, open standards, and services. The interactive computing tools JupyterHub, JupyterLab, and JupyterNotebook have all been created and supported by Project Jupyter.The term "Jupyter Notebook" can be used to refer to either the user-facing programme used to edit text and code or the underlying file format that can be used with a variety of implementations. A web-based interactive computational environment for authoring notebook documents is Jupyter Notebook (formerly IPython Notebook). Several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax, are used in the construction of Jupyter Notebook. A Jupyter Notebook application is a browser-based REPL that has an ordered set of input/output cells with code, text (written in Github Flavoured Markdown), math, graphs, and rich media as possible contents. A Jupyter Notebook document is a JSON file with the ".ipynb" extension that follows a versioned structure. The metadata, notebook format, and cell list are the three essential components of Jupyter notebooks. To set up and show the notebook, metadata acts as a data dictionary of definitions. The software's version number is Notebook Format. There are various cell types in a list, including cells for Markdown (to display), Code (to execute), and cells for code output. While ".ipynb" and JSON are the most common and default format it is possible to forgo some features (like storing images and metadata), and save notebook as markdown documents using extensions like JupyText. Jupytext is often in conjunction with version control to make diffing and merging of notebook simpler. While ".ipynb" and JSON are the most popular and default file types, it is possible to :forego some functionality and save your notebook as markdown documents by using extensions like JupyText. Version control is frequently used in conjunction with Jupytext to streamline notebook merging and diffing.

B. Tensorflow

TensorFlow is a free and open-source software library for artificial intelligence and machine learning. Although it can be applied to many different tasks, deep neural network training and inference are given special attention. The Google Brain team created TensorFlow for use in internal Google research and production. 2015 saw the maiden release under the Apache Licence 2.0. TensorFlow 2.0, the upgraded version of TensorFlow from Google, was launched in September 2019. Python, JavaScript, C++, and Java are just a few of the programming languages that support TensorFlow.This adaptability allows for a wide range of applications across numerous industries. Some APIs used with tensorflow are stated below.

C. Numpy

TensorFlow enables integration and compatibility with NUMPY, one of the most widely used Python data libraries, and its data structures. The native data type of the library, Numpy NDarrays, is automatically transformed to TensorFlow Tensors in TF operations; the opposite is likewise true. As a result, the user is spared from having to create explicit data conversions in order for the two libraries to operate together. Additionally, by having TF Tensors share the underlying memory representations of Numpy NDarrays whenever possible, the integration extends to memory optimisation.

D. Keras

Keras is a high-level, concise, and simple to use Python library that runs on the TensorFlow framework. The creation of layers for neural networks that keep the ideas of forms and mathematical details is one of the Deep Learning techniques that are the subject of this work.

E. DroidCam

DroidCam app is a free software which turns a mobile phone camera into a webcam. It is a cross-platform program which is compatible with various operating systems. In order to use this free software, users must have the DroidCam Wireless Webcam application installed on their mobile device and the PC Client component running on their computer. This is necessary because the PC Client must install the webcam drivers in order to connect the computer to the portable device

IV. IMPLEMENTATION DETAILS

A. Implementation Summary

In this project we are choosing Radish fields. We are going to feed a dataset of Radish leaves and weed images to train our model to distinguish between weeds and Radish crops. To identify and differentiate between Crop leaves and Weed we need to build a machine learning model. Our output can be predicted given the input . The available data is used to learn the relationship between the output and input. Fig(6.1) represents the block diagram of project Implementation in brief. A file that has been trained to recognise particular patterns is known as a machine learning model. A model is trained using a set of data and an algorithm that allows it to analyse and learn from the data. Using Python programming language to develop this model, since it is easy to code with.

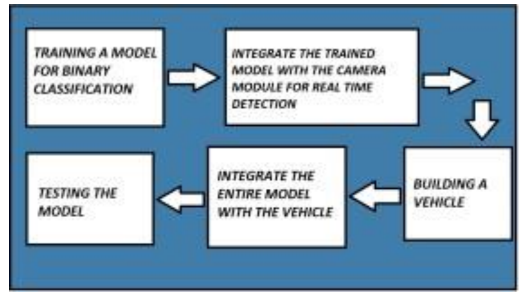


Fig. 5. IMPLEMENTATION BLOCK DIAGRAM

We need a dataset for the machine learning model before we train it. We have a dataset with hundreds of pictures of weeds and radish leaves. Our machine learning model is trained using a test dataset and training dataset. The form, size, and pattern of each leaf are labelled for each class after the dataset has been collected, as illustrated in the figWe need a dataset for the machine learning model before we train it. We have a dataset with hundreds of pictures of weeds and radish leaves. Our machine learning model is trained using a test dataset and training dataset. The form, size, and pattern of each leaf are labelled for each class after the dataset has been collected, as illustrated in the figWe need a dataset for the machine learning model before we train it. We have a dataset with hundreds of pictures of weeds and radish leaves. Our machine learning model is trained using a test dataset and training dataset. The form, size, and pattern of each leaf are labelled for each class after the dataset has been collected, as illustrated in the fig below.

Fig. 6. LABELLING OF WEED AND RADISH LEAVES

Following the collection and labelling of the data, we now train the data to produce a model that will predict the output depending on the input provided by the camera. We employ sequential classification models to categorize a series of photographs with the aid of the open-source programs Jupyter Notebook and TensorFlow. The sequential class can be used in creating a sequential model. Transfer learning using SSD MobileNet (Convolutional layers in Single Shot Detector are to check boxes of various aspect ratios at various locations with various scales). Once the model is trained we proceed to Real-Time Detection. This we achieve using Get the camera stream, then make frames to show the items that are being detected. The vehicle is made using a 300rmp Johnson motors connected to an ESC (mobitronix 60d or Cytron md10). It will

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

be controlled via flysky fsi6 transmitter and a fsi6 receiver. It will run over a 3s lipo battery. Once the entire weed detection system is attached to the vehicle, it is ready to test in the fields.

B. Implementation Procedure

1) Import Dependencies: Dependencies are all of the software parts that the model needs in order to function properly and prevent runtime issues. Python modules can access code from other modules by using the import command to include the file or function. The most frequent method of using the import mechanism is the import statement. Fig(6.1) represents the block diagram of project Implementation in brief. One of the major dependency is OpenCV (where CV stands for Computer Vision). The widely used open-source package OpenCV offers a wide range of computer vision and image processing capabilities. It provides a wide range of tools and methods that enable programmers to create innovative applications in a variety of industries, including robots, augmented reality, facial recognition, object detection, and many more. OpenCV has established itself as a top choice for developers all around the world thanks to its robust features and approachable design. Importing Computer Vision - Version 2 (cv2) for the model. Second most important dependancy is UUID (Universal Unique Identifier), a Python package that makes it possible to create random objects with 128-bit identifiers. It offers uniqueness since it produces IDs based on time and computer hardware (MAC, etc.). After all the dependencies are imported, the next step is to set up the environment. It refers to creating a workspace to make changes without altering any data in a live environment. An environment compatible with Python 3. Here we use Project Jupyter (Jupyter Notebook), which is an open source platform for coding, developing and training a machine learning model.

2) Defining and Capturing Images: The first step is to form categories (labels) for classifying the objects in a given image.. Here we have two labels, namely 'weed' and 'crop.

Fig. 7. DATA CONTENT

The performance and accuracy of a machine learning model is thought to be significantly influenced by the dataset size. Small datasets may result in over-fitting, while large datasets typically result in superior classification results. The database consists of a dataset with hundreds of real time images collected from various Radish fields. It includes images of crop at different stages of Radish crop's growth and images of different types of weeds that usually grow in a Radish feild.

3) Image Labeling: In computer vision, Image labeling is the technique of giving specific tags to the specific items or objects existing in an image so that the machine learning model can easily distinguish between them. This technique can be applied on raw data like images and videos. Image labelling is a manual process and requires quality time to complete. LabelImg, shown in fig(6.4), is an open-source graphical image annotation tool which is written in python and it uses Qt for its graphical interface, allows you to draw visual boxes around your objects in each image, serves this purpose. To install LabelImg for Python 3 we use pip command ,'pip3 install labelImg'.

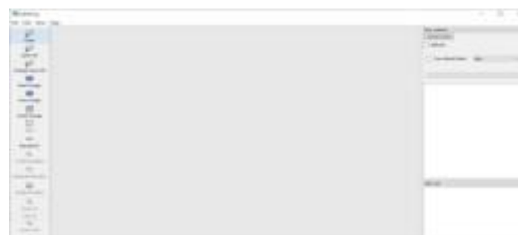


Fig. 8. LABELLING TOOL

Fig. 9. XML FILE INCLUDING LABELLED IMAGES

The act of annotating an image basically involves adding metadata to a dataset to specify an object. In simple words, image annotation is a type of data labeling also known as tagging or transcribing. The object label information and position information marked in each image gets saved in an xml file format for training. The software automatically generates this xml file with the same name as that of the image used for labelling. The fig(6.5) shows the content of an XML file. Some more benefits of using LabelImg are, light weight and easy utility, cross platform performance, fast annotation, supports multiple formats and supports multi-object annotations.

4) Move labelled images into a Training and Testing Partition: The splitting of the dataset in train and test helps to gauge how well machine learning algorithms work when employed with prediction-based methods.The standard ratio for partitioning the data from the dataset is 9:1, which means

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

90 percent of the photos are utilized for training and the remaining 10 percent are kept for testing, but you can choose whichever ratio best suits your model. In our model we choose the ratio to be 7:3, so the train set includes 70 percent of the entire dataset in which 35 percent are weed images and the remaining 35 percent are crop (Radish) images. Similarly the test set consists of 30 percent of the entire dataset in which 15 percent are weed images and 15 percent are crop images.

5) *Setting up the Paths:* There exists numerous algorithms for image detection, namely, Region-based Convolutional Neural Networks (R-CNN), Spatial Pyramid Pooling (SPP-net), Histogram of Oriented Gradients (HOG),etc.. The algorithm used in this model is Single Shot Detector (SSD) MobileNet. It is an object detection model that computes the output bounding box and object class from the input image.

- SSD MobileNet Architecture

The SSD architecture is a single convolution network that learns to predict bounding box locations and classify these locations in one pass.

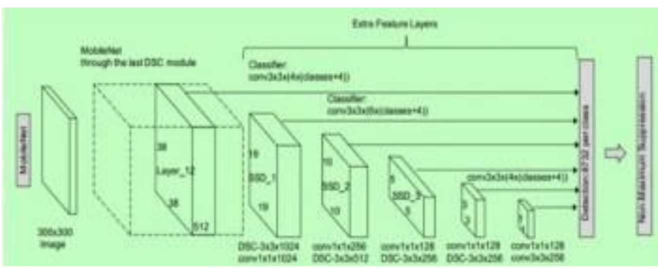


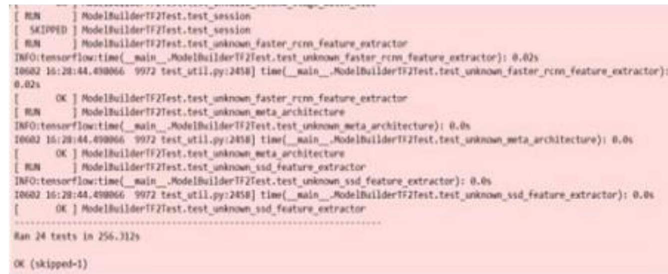
Fig. 10. SSD MOBILE-NET ARCHITECTURE

The SSD network consists of base architecture followed by several convolution layers, as shown in figure. By using SSD, we only need to take one single shot to detect multiple objects within the image.The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

6) *Download TF Pretrained Models from Tensorflow Model Zoo and Install TFOD:* A pre-trained model is a saved model that has previously undergone training over a large dataset, typically on a large-scale image-classification function. Without having to construct an entire model from scratch, you can easily add machine learning capability the edge device applications by using pre-trained TensorFlow Lite models. The TensorFlow model zoo consists of symbolic models which are useful for inference. Certain parameters for each model in this model zoo have already been trained for the particular datasets they use. Using TensorFlow model zoo makes compilation easier. Here we install TensorFlow Object Detection API.

- Tensorflow Object Detection API

TensorFlow object detection API is the framework for creating a deep learning network that solves object detection problems. There are already pretrained models in their framework which they refer to as Model Zoo. This includes a collection of pretrained models trained on the COCO dataset, and the Open Images Dataset.model. They are also useful for initialising the models when training on the novel dataset. We need to verify the installation using a verification script, producing the output shown in the fig.



```
[ RUN      ] ModelBuilderTF2Test.test_session
[  SKIPPED ] ModelBuilderTF2Test.test_session
[ RUN      ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.02s
I0602 16:28:44.498066  9972 test_util.py:2458] time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.02s
[       OK ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
[ RUN      ] ModelBuilderTF2Test.test_unknown_meta_architecture
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
I0602 16:28:44.498066  9972 test_util.py:2458] time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_meta_architecture
[ RUN      ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
I0602 16:28:44.498066  9972 test_util.py:2458] time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
----------------------------------------------------------------------
Ran 24 tests in 256.312s

OK (skipped=1)
```

Fig. 11. VERIFICATION SCRIPT OUTPUT

If any dependency is yet to be imported, this script will notify once its run giving an error.If the installation is completely successful, the code will return 1(OK). The TensorFlow Object Detection API uses Protobufs for configuring the model and training parameters, it is a type of data format used for serializing the structured data.

V. DETECTING FROM AN IMAGE

After the model is completely train, here we test the model. An image is given to the model from the test dataset where it detects crops by building a green box around it and a yellow box for weeds. The same is displayed in fig.

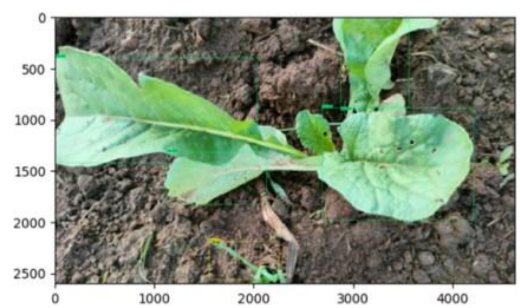


Fig. 12. OBJECT DETECTION FROM AN IMAGE

This is a crucial step as here we can determine if we are getting the desired output and accuracy or if the model still lacks proper training. If the model is unable to detect the objects correctly then we need to retrain the model over and over again, feeding more data until it detects the objects correctly in this step.

A. Real Time Detection from Camera

Real Time Object Detection is a cv (computer vision) task that comprises of identifying, detecting and locating objects of interest in any real-time video sequences with a fast conclusion while maintaining the base level of computational accuracy.

An object detection system's task is to classify and localize (find) every object in a picture. In our system, the rectangular bounding box is listed as the output, and the fundamental input for the detector is an image that includes the item. In short, it means detecting objects from live camera and simultaneously displaying those detected objects on the respective screen. Here we keep capturing the camera stream and simultaneously keep creating frames for displaying all the detected objects. We use DroidCam software to connect the mobile

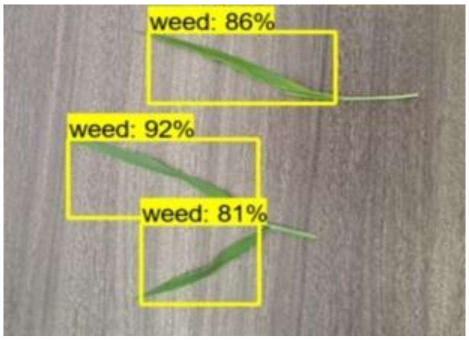


Fig. 13. REAL TIME DETECTION OF WEED

camera with our trained model. Fig(6.15) shows real time detection of weed leaves along with the computational accuracy in percentage.

Fig(6.16) displays real time detection of the crop (Radish leaves) and weeds along with the computational precision in percentage.

Fig. 14. REAL TIME DETECTION OF RADISH LEAVES AND WEEDS

VI. RESULTS

The machine learning model is compatible with real time weed detection, and the same is tested. Fig(7.1) displays the motor driven vehicle with a mobile phone camera capturing the input and sending it wirelessly to the model.

VII. FINAL RESULTS

The machine learning model is modified compatible with real time weed detection, and the same is tested.The bot is now able to detect and extract the same Fig(7.1) displays the motor driven vehicle with a pi camera capturing the input and sending it manually to the model which collects the frames and commands the arm to pick up the weed .

Fig. 15. THE VEHICLE

VIII. MECHANICAL-ARM

Fig. 16. MECHANICAL-ARM

Fig. 17. MECHANICAL-ARM

Fig. 18. MECHANICAL-ARM

A. DRIVEASY PICK AND PLACE ROBOTIC ARM

Fig. 19. Arm

- 2 x Driveasy Pick and Place Robot Arm Coupling
- 1 x Driveasy Pick and Place Robot Arm Gripper Motor Mounting Plate
- 1 x Driveasy Pick and Place Robot Arm 1st Axis
- 1 x Driveasy Pick and Place Robot Arm 2nd Axis
- 1 x Driveasy Pick and Place Robot Arm Gripper
- 1 x Amass XT30 Connectors Male and Female 1 Pair Nylon

B. BbOX TORQUE SERIES ECCENTRIC SHAFT 11 RPM D37R516E.

Fig. 20. Bbox Torque Series Eccentric Shaft 11 RPM D37R516E.

- Features:
- Low Speed: Operates at 11 RPM.
- Versatile Voltage Range: Compatible with 6V to 12V DC.
- High Reduction Ratio: Optimized for torque.
- Durable Construction: Metal gears for longevity.
- Reliable Torque: Provides 8kg.cm.
- Efficient Power: 1.1W rated power.
- The BBox Torque Series 11 RPM DC Motor, model no. D37R516E, epitomizes engineering excellence, meticulously crafted to deliver unparalleled reliability and precise performance across a diverse array of applications.
- Notably, boasting an operating voltage range spanning from 6V to 12V and a nominal voltage of 12V, this motor offers unmatched versatility and compatibility, seamlessly adapting to various power sources and configurations with ease and efficiency.
- Moreover, featuring a reduction ratio of 516, it guarantees meticulous speed control and efficient operation, ensuring optimal performance in tasks requiring intricate precision and control.
- Furthermore, with a rated base speed of 5500 RPM and a no-load speed of 11 RPM, this motor showcases its capability to deliver consistent and reliable performance, even in applications necessitating low-speed rotation.
- Constructed with robust metal gears, it not only ensures reliable operation but also enhances longevity, withstanding the rigors of continuous use with utmost resilience and durability.
- Additionally, boasting a rated torque of 10kg.cm and a rated power of 1W, this motor provides the perfect balance of strength and efficiency, empowering you to tackle your projects with confidence and precision.
- Elevate your creations and unlock new dimensions of precision and reliability with the unrivaled performance of the BBox Torque Series 11 RPM DC Motor, model no. D37R516E.
- Specs:
  1) Model No: D37R516E.
  2) Operating Voltage (VDC): 6v – 12v
  3) Nominal Voltage (V): 12 GB
  4) Reduction Ratio: 516
  5) Rated Base Motor RPM: 5500
  6) Motor No Load RPM: 11
  7) Gear Material: Metal
  8) Rated Torque (Kgf.cm): 10kg.cm
  9) Rated Power (W): 1W
  10) Shaft Diameter (mm): 6 mm
  11) Shaft Length (mm): 14 mm
  12) Gearbox Diameter (mm): 37 mm
  13) Motor Diameter(mm): 36 mm
  14) Total Length (mm): 83.8mm
  15) Weight (gm): 0.2kg
  16) Gear box Width (L): 31 mm

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

C.

BbOX TORQUE SERIES ECCENTRIC SHAFT 16 RPM D37R314E.

Fig. 21. gear 2

- Features:
- Low Speed: Operates at 16 RPM.
- Versatile Voltage Range: Compatible with 6V to 12V DC.
- High Reduction Ratio: Optimized for torque.
- Durable Construction: Metal gears for longevity.
- Reliable Torque: Provides 8kg.cm.
- Efficient Power: 1.1W rated power.
- The BBox Torque Series 16 RPM DC Motor, model no. D37R314E, stands out as a pinnacle of engineering excellence, meticulously crafted to deliver unparalleled reliability and precise performance across a diverse array of applications, making it an indispensable asset in your projects.
- Boasting an operating voltage range spanning from 6V to 12V and a nominal voltage of 12V, this motor offers unmatched versatility and compatibility, seamlessly adapting to various power sources and configurations with ease and efficiency.
- Notably, featuring a high reduction ratio of 314, it guarantees meticulous speed control and efficient operation, ensuring optimal performance in tasks requiring intricate precision and control.
- Furthermore, with a rated base speed of 5000 RPM and a no-load speed of 16 RPM, this motor showcases its capability to deliver consistent and reliable performance, even in applications necessitating low-speed rotation.
- Constructed with robust metal gears, it not only ensures reliable operation but also enhances longevity, withstanding the rigors of continuous use with utmost resilience and durability.
- Additionally, boasting a rated torque of 8kg.cm and a rated power of 1.1W, this motor provides the perfect balance of strength and efficiency, empowering you to tackle your projects with confidence and precision.
- Elevate your creations and unlock new dimensions of precision and reliability with the unrivaled performance of the BBox Torque Series Eccentric Shaft 12V 16 RPM DC Motor, model no. D37R314E.
- Specs:
  1) Model No: D37R314E.
  2) Operating Voltage (VDC): 6v – 12v
  3) Nominal Voltage (V): 12 GB
  4) Reduction Ratio: 314
  5) Rated Base Motor RPM: 5000
  6) Motor No Load RPM: 16
  7) Gear Material: Metal
  8) Rated Torque (Kgf.cm): 8kg.cm
  9) Rated Power (W): 1.1W
  10) Shaft Diameter (mm): 6 mm
  11) Shaft Length (mm): 14 mm
  12) Gearbox Diameter (mm): 37 mm
  13) Motor Diameter (mm): 36 mm
  14) Total Length (mm): 82.8mm
  15) Weight (gm): 0.2kg
  16) Gear box Width (L): 30 mm

D.

ELECTROCRAZE 20D-BD DUAL CHANNEL 20AMP BRUSHED MOTOR CONTROLLER

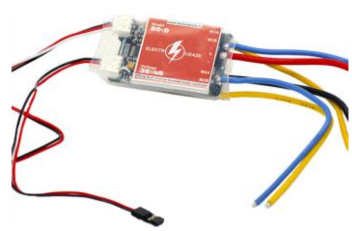


Fig. 22. ELECTROCRAZE 20D-BD DUAL

- Features:
- Reverse Option: Quick direction changes in tight spaces.
- Signal-Loss Failsafe: Activates failsafe mode for safety during signal loss.
- Isolated Heat Sink Plate: Efficiently dissipates heat for prolonged performance.
- Breaking Feature: Yes
- The Electrocraze 20D-BD, a cutting-edge electronic control module designed to electrify your projects with precision and power. Boasting a Built-in Battery Eliminator Circuit (BEC) capable of delivering a stable 5 volts at a maximum current of 100 milliamps, this compact dynamo ensures consistent performance without the need for external power sources.
- Crafted for versatility, the Electrocraze 20D-BD seamlessly integrates with R/C servo inputs, offering seamless compatibility across a range of applications. Weighing

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

in at just 50 grams (excluding wires), it adds minimal bulk while maximizing functionality, making it ideal for projects where weight is a concern.

- Equipped to handle a wide spectrum of voltages, the Electrocraze 20D-BD accepts inputs within the range of 11 to 17 volts, accommodating 3S to 4S batteries with ease. Its robust design ensures reliability under varying power conditions, giving you peace of mind during operation.
- When it comes to output, the Electrocraze 20D-BD truly shines. Delivering a continuous output current of 20 amps per channel, it empowers your projects with ample power for demanding tasks. Need an extra surge of power? No problem. With peak output currents of up to 45 amps per channel, the Electrocraze 20D-BD rises to the occasion, providing the boost you need to tackle even the most challenging projects with confidence.
- Whether you're building a remote-controlled vehicle, animatronic creation, or any other electrifying endeavor, the Electrocraze 20D-BD is your ultimate partner in power and precision. Harness its capabilities to bring your ideas to life and unleash the full potential of your projects like never before.
- Specs:
  1) Input Voltage Range: 11-17V (3S-4S)
  2) Output Current (Continuous): 20 amps/channel
  3) Output Current (Peak): 45 amps/channel
  4) Dimensions: 62 x 40 x 16 mm (excluding wires)
  5) Weight: 50g (excluding wires)
  6) Input: Standard Servo Input
  7) Signal Range: 1100 – 1900 us
  8) Throttle Centre: 1460-1540 us
  9) Dead Band (From Centre): 40 us
  10) BEC: Yes (5V 100mA max)
  11) Brake: Yes
  12) Signal Loss Protection: Yes
  13) Input Wire Thickness: 14 AWG Silicone Wire (Red and Black)
  14) Output Wire Thickness: 16 AWG Silicone Wire (Yellow and Blue)

E.

GNB GAONENG 3S 2200mah 11.1V LiPo battery

Fig. 23. battery for arm

- Features:
- Premium Cell Composition: Superior performance.
- Wide Compatibility: Fits various devices.
- Long-lasting Power: Extended use.
- Optimal Balance: Stable operation.
- Enhanced Drive Experience: Improved performance.
- Easy Integration: Hassle-free setup.
- The GNB GAONENG 3S 2200mah 11.1V LiPo battery is a high-performance power source designed for demanding RC applications such as drones, cars, boats, and planes.
- With a capacity of 3300mAh and a 3S1P configuration, it delivers a stable nominal voltage of 11.1V, ensuring consistent performance during operation.
- Its 90C continuous discharge rate and impressive burst rate make it capable of delivering high current outputs, providing the power required for fast acceleration and stable performance in high-drain systems.
- The battery is lightweight and compact, making it ideal for setups requiring a balance of power and portability. It features a JST-XH 4P charge connector and versatile discharge connector options, including XT60 and Deans, for seamless compatibility with various devices.
- Designed to operate efficiently in a wide temperature range of -20°C to 60°C, this battery also offers a robust life cycle of over 300 uses, making it durable and reliable for long-term use.
- Whether you're into racing or aerial photography, this battery is an excellent choice for maximizing performance and reliability in your projects
- Specs:
  1) Model No.: GNB22003S110A
  2) Capacity: 2200mAh
  3) Configuration: 3S1P
  4) Nominal Voltage: 11.1V
  5) Dimensions: 25 × 35 × 106 mm (Approx.)
  6) Weight: 185g (±5g)
  7) Max Continuous Discharge: 110C
  8) Max Burst (3 Sec): 220C
  9) Charge Connector: JST-XH 4P
  10) Discharge Connector: XT60, Deans, or other options
  11) Working Temperature (Charge): 0°C 45°C
  12) Working Temperature (Discharge): -20°C 60°C
  13) Working Humidity: 65
  14) Storage Temperature: -20°C 35°C
  15) Storage Humidity: 65
  16) Life Cycle: 300+ cycles

IX.

PLUCKING ROBOT PROGRAMMING WORKFLOW

- The core of this automated weed-plucking system is a Python-based control architecture running on a Raspberry Pi 3. It integrates a computer vision model from Roboflow, real-time camera input, motor control via

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

GPIO, and a graphical user interface (GUI) for live monitoring and system feedback. The program coordinates detection, decision-making, and mechanical action in a loop, allowing the robot to identify and remove weeds autonomously in a small agricultural setting.

X.

SYSTEM COMPONENTS AND CODE INTEGRATION

- Microcontroller: Raspberry Pi 3 Acts as the central processing unit, hosting the Python environment, GPIO interface, camera input, and the main control logic.
- Camera: Raspberry Pi Camera Module Rev 1.3 Configured using the Picamera2 library in preview mode to minimize frame capture latency.
- Machine Learning Integration: Roboflow Model The object detection model is hosted on Roboflow and accessed using their Python SDK. Weed presence is evaluated on every frame.
- Detection Logic and Trigger Condition A queue of the last 3 frame evaluations determines whether to proceed with the plucking sequence.
- Actuation: Motorized Arm and Gripper Controlled using GPIO PWM signals connected to a dual-channel ESC that drives two DC gear motors.
- Arm Motor: BBox Torque Series 11 RPM
- Gripper Motor: BBox Torque Series 16 RPM
- Motor Controller: ElectroCraze 20D-BD Dual Channel ESC
- GPIO Pins: Arm - GPIO 17, Gripper - GPIO 22
- PWM Frequency: 50 Hz (used with RPi.GPIO library)

A. Hardware Wiring and Power Setup

The hardware setup connects the Raspberry Pi GPIO pins to an ESC (Electronic Speed Controller), which in turn controls the DC motors. The ESC draws power from a dedicated 12V source, and each motor is wired to a separate channel on the ESC:

- **GPIO 17 (Pin 11)** controls ESC Channel A for the arm motor.
- **GPIO 22 (Pin 15)** controls ESC Channel B for the gripper motor.
- **GND connections** are shared between the Pi and ESC.
- The ESC power input is connected to a 12V supply capable of driving both motors.

B. Return to Origin Logic

- Ensures system alignment after each weed removal cycle.

C. Graphical User Interface (GUI)

- Provides real-time camera feedback and status updates to the user using Tkinter.

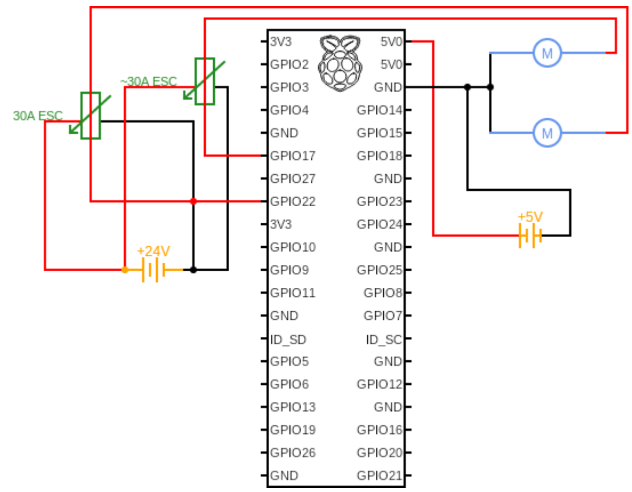


Fig. 24. hardware diagram

XI. ML MODEL TRAINING AND EVAULATION

A. Overview

This section provides an in-depth explanation of the programming approach used to train and implement a keras model for weed detection. The implementation includes data preparation, model creation, training, and evaluation along with merging and integrating software with the mechanical hardware

B. Dataset

The dataset used for this project is a large-scale collection designed to train and evaluate a machine learning model for weed and plant detection. It is organized into two main directories: ”train” for the training data and ”val” for the validation data, each contain ing three subfolders: ”weed”, ”no weed”, and ”background”. Each subfolder holds images that correspond to the respective class. For the training set, the dataset includes approxi mately 1800 images per class, providing a robust foundation for the model to learn to distinguish between the three categories—weed, no weed, and background. These images cover a wide range of real-world scenarios to ensure diverse training conditions, enhancing the model’s ability to generalize to various environments. Similarly, the validation set con sists of 900 images per class, which allows for an unbiased evaluation of the model’s performance during training and fine-tuning. The dataset also includes corresponding annotations in the form of bounding boxes for each image, allowing for supervised learning of object detection tasks. With this large and well-structured dataset, the model can be trained to accurately classify and detect weed and background in real-time applications, such as autonomous weed detection and removal systems.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04586
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

XII. MODEL EVALUATION

Class	Precision	Recall	F1 Score	Support (Count)
Osot	0.0000	0.0	0.0000	1
Shavel	0.0000	0.0	0.0000	4
crabgrass	0.0000	0.0	0.0000	15
weed	0.0000	0.0	0.0000	539
weeds	0.0000	0.0	0.0000	0
Bodyak	0.0000	0.0	0.0000	8
carpet weeds	0.0000	0.0	0.0000	8
none	0.0017	1.0	0.0035	1

TABLE I
CLASSIFICATION REPORT

Classification Report Analysis

1) The classification report summarizes the performance of the model on each class in the test dataset. It includes four key metrics:
2) Precision: The proportion of correct positive predictions for each class out of all positive predictions made for that class.
3) Recall: The proportion of actual positives for each class that were correctly identified by the model.
4) F1 Score: The harmonic mean of precision and recall, providing a balanced measure of model accuracy.
5) Support: The number of true instances for each class in the test dataset.

XIII. CONCLUSION

The evaluation clearly demonstrates that the model is highly biased toward the 'weed' class due to significant class imbalance. To improve performance:

- Collect more samples for underrepresented weed types.
- Use data augmentation to artificially balance classes.
- Consider techniques like class weighting or oversampling.

XIV. REFERENCES

1) Advances in ground robotic technologies for site-specific weed management in precision agriculture: A review Author links open overlay panel Arjun Upadhyay a, Yu Zhang a, Cengiz Koparan a d, Nitin Rai a, Kirk Howatt b, Sreekala Bajwa c, Xin Sun https://doi.org/10.1016/j.compag.2024.109363
2) Robotic weed control using automated weed and crop classification Wiley Journal of Field Robotics February 202037(2):322-340 DOI:10.1002/rob.21938
3) Recent Advances in Agricultural Robots for Automated Weeding by Chris Lytridis *ORCID andTheodore PachidisORCID HUMAIN-Lab, Democritus University of Thrace (DUTH), 65404 Kavala, Greece Author to whom correspondence should be addressed. AgriEngineering 2024, 6(3), 3279-3296; https://doi.org/10.3390/agriengineering6030187 Submission received: 27 August 2024 / Revised: 9 September 2024 / Accepted: 10 September 2024 / Published: 11 September 2024