Automatic Weeding Arm

Spring 2020 ECE 445 Senior Design

Final Report

Team 9

Sowji Akshintala

Sophie Liu

Shuyue (Lucia) Zhang

TA: Johan Mufuta

May 8, 2020

Abstract

Our project aims to design an automatic weeding arm which can be implemented on a robotic base to identify and cut weeds within 3-20cm tall. The homing, searching, classification and cutting mechanisms of the robotic arm are achieved by the coordination of the following four subunits: control unit, recognition unit, motor unit and power unit. Through the serial communication between the ATMega328P microcontroller and Raspberry Pi 4 board, the arm is able to control the motion of the motors to achieve a cutting and homing mechanism, locate plant seedlings within 2cm errors, and identify the type of the plant with an accuracy of 95.5% through a trained convolutional neural network model. Further details and results of the project are documented in this report.

Contents

1 Introduction

1.1 Background

For generations, humans have used manual labor to curb the growth of weeds which leech nutrients and resources from staple crops. As agricultural demands and farm sizes grew, the industry started to invest in chemical herbicides to ensure maximum yields. Herbicide use, however, has become controversial for its carcinogenic potential and environmental concerns [1,2]. Runoff from the herbicide sprays threatens the natural ecosystem through the groundwater and soil [3]. Long-term exposure to herbicides has also been linked to kidney, liver and spleen complications, as well as developmental problems in pregnancies, and hormone disruption and miscarriages [4,5]. In terms of economics, chemical crop control has been slowly bleeding farmers dry. Agrochemical companies have been selling genetically modified (GM) crops, which only boosts their sales over time as weeds have evolved into "superweeds" and require stronger doses to kill [6]. This ballooning effect can be clearly noted in the soy industry, where, as of 2008, 92% of soy plants had become glyphosate-resistant, requiring GM crops to be used with liquid herbicides in tandem [7]. Meanwhile, agrochemical companies have quietly quintupled their prices for both GM seeds and chemical herbicides within the last two decades [8]. Ethically, herbicide use must be phased out, but regressing to human labor is not a realistic solution. Naturally, robotics is an ethical, cost-and-labor-effective and chemical-free alternative to solve the current herbicide problem.

1.2 Objectives

We propose a solution of an automatic robotic weeding arm that can identify and cut post-emergent weeds with an attached blunted sheer. Although automatic weeding robots have been commercialized, they still rely on chemicals, which perpetuates the risks of repeated use of herbicides in the long run [8]. Since there are existing plant monitoring robots that can navigate the difficult terrain of crop fields, such as the TerraSentia, we are not focusing on the robotic base [9]. Rather, we consider the arm as a potential extension, allowing us to target the specific problem of chemical-free weed removal.

Our arm focuses on the identification of various seedling species and the automatization of the weeding process. The arm is fitted with a camera that can differentiate different plant seedlings through a convolutional neural network training and can enable real-time video monitoring from a connected computer screen. Once the arm detects the unwanted plant, it can maneuver and cut the weed with its motorized sheer. We decided to cut instead of pulling the weeds because cutting requires less force and is more efficient with tall plants. To

accomplish this function, the arm will ideally have 3 degrees of freedom, providing enough flexibility to trim the weeds. With the arm's trainability, it can also be easily repurposed to perform many different agricultural functions. For example, once the arm can learn from various plant databases, it could easily be used to pick fruit or trim foliage just by switching out the shear-hand attachment for other applicable tools.

Figure 1 shows the concept design of the automatic weeding arm. It has three gear motors and two servo motors. Two of the gear motors control the up-and-down motion in the x-y plane. The base gear motor controls a 360-degree rotation on the y-z plane. A servo motor controls the up-and-down motion of the sheer while another controls the open-and-close motion.

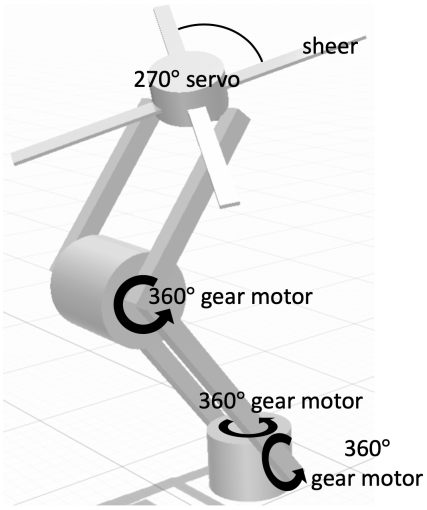


Figure 1: Concept Design

In terms of electrical design, Figure 2 presents the block diagram of the system. Our project is composed of four different units: Motor Unit, Control Unit, Recognition Unit and Power Unit. The motor unit consists of servo and gear motors. The control unit consists of an LED to indicate the status of the input image captured by camera (weed/crop), a microcontroller and an ultrasound module. The recognition unit consists of the camera module and the Raspberry Pi which processes the neural network model to make a prediction. The power unit consists of a 12V battery, a voltage divider and a switch.

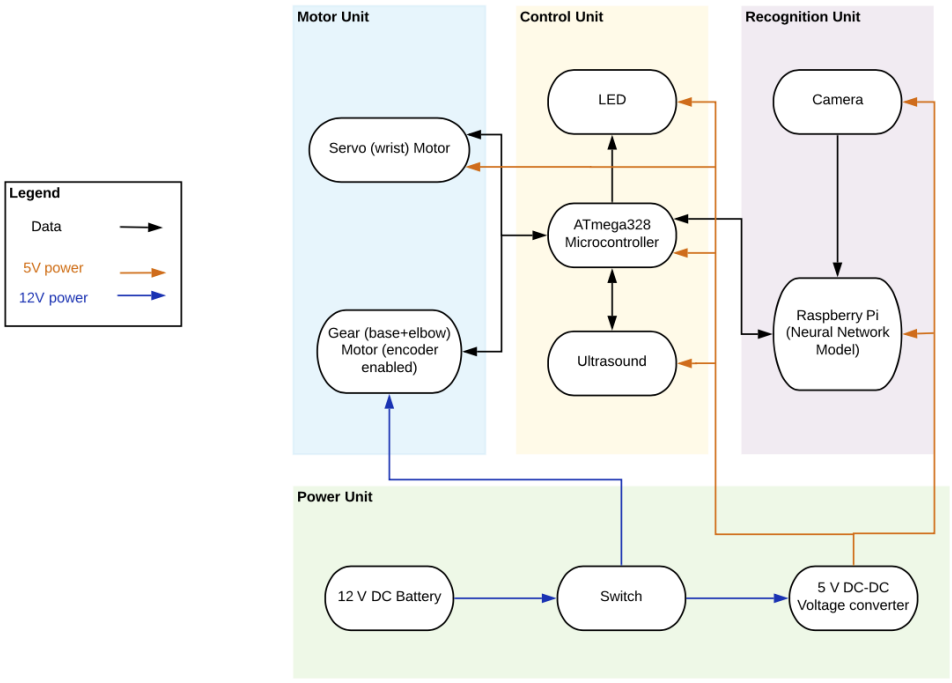


Figure 2: High-level Block Diagram

Figure 3 presents a proposed cutting mechanism. Since the seedling database mainly consists of the top view of the plants, we implemented the camera to attach to the bottom of the sheer. There are 2 ultrasound modules and 2 cameras monitoring the front and the bottom-to-sheer information.

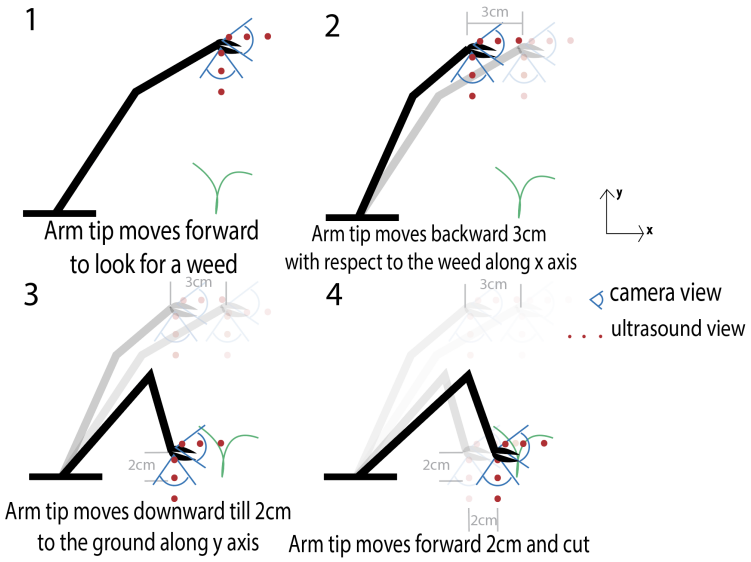


Figure 3: Primary Cutting Mechanism

1.3 General Design and Modification

Figure 4 presents the assembly of the arm. Due to the COVID-19 lock-down, the machine shop was only able to build part of the arm as shown in Figure 4. We added a base (Figure 5), servo motors, sensor units and the sheer (Figure 6).

There are several physical modifications we had to make due to this situation:

- We planned to have 3 degrees of flexibility along the x, y and z axis. However, as the shaft extender on the bottom-base motor was not assembled, it disconnects the base from the motor. Since we did not have proper tools and the extender to fix the issue, we decided to give up one degree of freedom along the y-z plane. The elbow motor and base-top motors were functional to be driven along the x-y plane.
- We planned to attach two ultrasounds to the arm tip. However, due to the two-piece, shaft-extender-to-arm, the joint was not strong enough to hold the extra sensor. To avoid the arm from falling whenever the joint move past a threshold, we added glue to the joint, reduced to one ultrasound unit and replaced one of the servo motors to a lighter substitute.
- Similar as the ultrasound module, the arm was not able to host the second camera. We eventually used only 1 camera.

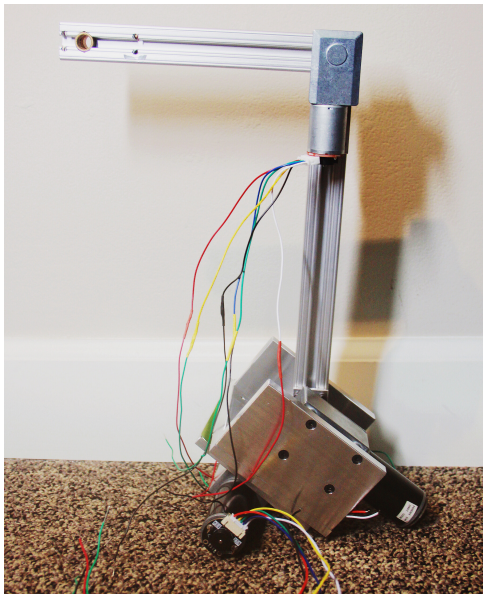


Figure 4: Arm frame

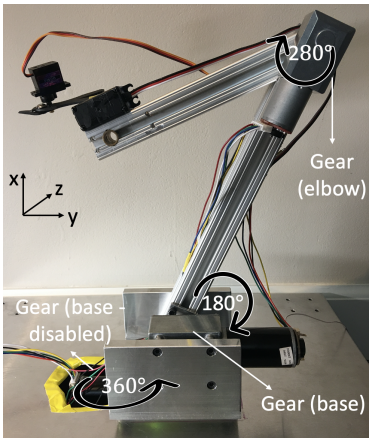


Figure 5: Motor presentation

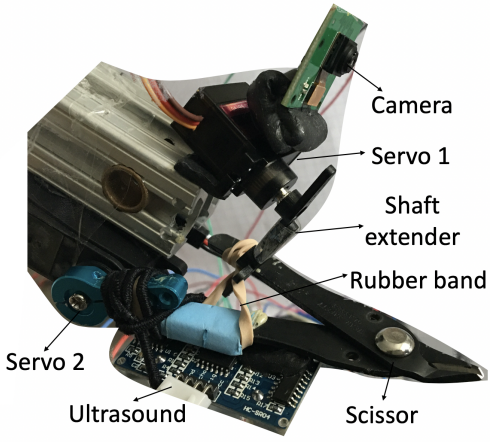


Figure 6: Sensor presentation

The cutting mechanism we adjusted to coordinate with the current physical design is explained by Figure 7.

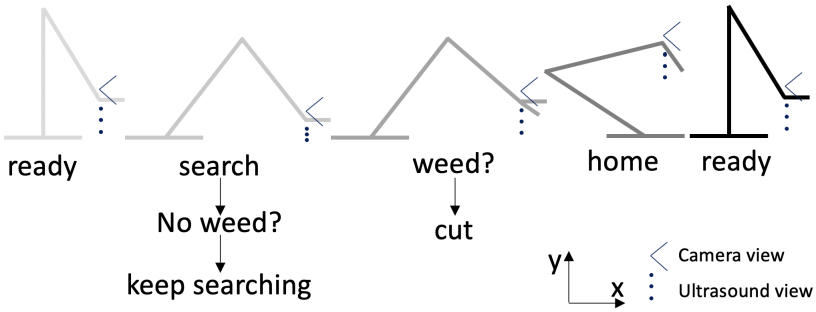


Figure 7: Adjusted Cutting Mechanism

2 Design

2.1 Control Unit

The control unit consists of a microcontroller, a LED, and ultrasound modules to update distance information between the arm tip to the plants or the ground. However, due to the mechanical restraint in weight and space, we were only able to install one ultrasound module at the bottom of the arm tip. Therefore, the main functionality of the ultrasound became detecting distance to the ground in order to avoid crushing.

2.1.1 Microcontroller

The main functionality of the microcontroller is to control different components such as the ultrasound module and motors, and to communicate with the Raspberry Pi. The microcontroller we used was ATmega328-PU because it provides enough I/O, has enough memory to host the code and is easy to acquire. Figure 14 in Appendix B presents the schematics of the PCB with the microcontroller connected to different components. Figure 15 a and b in Appendix B show the PCB footprints and the fabricated board.

2.1.2 Ultrasound

Since only one ultrasound module was able to be installed eventually, its main functionality was to detect distance and ground. We chose a HC-SR04 ultrasound module because it can update distance fast (trigger = 10us), have a high precision limit (accurate to 1cm) and is easy to acquire.

2.2 Recognition Unit

The recognition unit consists of three components: a Raspberry Pi 4 Board, an Arducam 5MP OV5647 Raspberry Pi camera module, and a trained convolutional neural network (CNN) model. The main goal of the recognition unit is to capture the image, achieve real-time monitoring, and classify crop or weed type of a plant seedling.

2.2.1 Raspberry Pi Board

The entire recognition is built upon the 4th-generation Raspberry Pi, powered by a 1.5GHz 64-bit quad-core ARMv8 CPU with 2GB LPDDR4 SDRAM. Compared to other alternative models like NVIDIA Nano, the Raspberry Pi does have limited computation power. However, it is a more economical choice and the features mentioned above are able to ensure a sufficiently high performance in serial communication with the microcontroller as well as basic image processing and classification for a prototype. In addition, a 64GB SanDisk mi-

croSD Card was also used for the Raspberry Pi Board to expand its usable memory storage. Since processing a machine learning model is computationally heavy, this allows a faster processing speed and enables further improvement of the CNN model.

2.2.2 Camera

The Arducam 5MP OV5647 Raspberry Pi camera module is selected as the device for image capturing and real-time monitoring. This camera was chosen due to its easy connection with Raspberry Pi board, motorized focus lens and sufficient resolution (still picture resolution = 2592×1944 and max video resolution = 1080p).

In order to obtain focused images, we calculated the optimal distance between camera module and the plant based on the equations of angle of view (AoV) and field of view (FoV) listed in equation 1 and 2:

$$AoV = 2 \times \arctan(\frac{\text{sensor width}}{2 \times \text{focal length}}) \tag{1}$$

$$FoV = 2 \times \tan(\frac{\text{AoV}}{2}) \times \text{distance to object} \tag{2}$$

According to the datasheet of Arducam 5MP OV5647 Raspberry Pi camera [11], the sensor size of the camera is 3.672.74 mm (1/4" format). The camera has an angle of view (AoV) of 54×41 degrees, and a field of view(FoV) of $2.0 \times 1.33m$ at 2m. Therefore, the focal length of the camera can be calculated as

$$\frac{0.003672}{2 \times \tan(54°/2)} \approx \frac{0.274}{2 \times \tan(41°/2)} \approx 0.0036\text{m} \tag{3}$$

As the Raspberry Pi camera lens falls into the standard lens range, it ensures images to not have any kind of distortion. In order to capture weed and crop images similar to the images from CNN model training dataset, the ideal images captured by the camera should have objects take up >80% of the frame. As all types of weeds are relatively small objects, we approximate a field of view to be about $0.5m \times 0.35m$. Then the distance of the lens to the weed needs to be

$$\frac{FoV}{2 \times \tan(\text{AoV}/2)} = \frac{0.5}{2 \times \tan(54°/2)} \approx \frac{0.35}{2 \times \tan(41°/2)} \approx 0.5\text{m} \tag{4}$$

Since we were not able to obtain real crop plants, we used images on screen to test the CNN model performance on classifying crop types. The processed images shown on the screen are smaller, with a rough size of $0.015m \times 0.011m$. Therefore, the current optimal testing

distance between the lens and object is less than 0.5m, which is:

$$\frac{0.015}{2 \times \tan(54°/2)} \approx \frac{0.011}{2 \times \tan(41°/2)} \approx 0.015\text{m} = 15\text{cm} \tag{5}$$

The camera module was also used to take images of real weed plants to expand V2 Plant Seedling Dataset[12]. Sample images and The code for controlling the camera module to take photos and monitor in real-time can be found in Section Appendix D.

2.2.3 Convolution Neural Network (CNN) Model

A convolution neural network (CNN) model is chosen for the task of image classification. The model is trained through Google Colaboratory to utilize Tensorflow with GPU. CNN is widely used for various computer vision tasks nowadays. Two main advantages of CNN led us to finalize it as the final machine learning model. One main advantage of CNN is that it automatically detects the important features without any human supervision. Furthermore, it is also computationally efficient as it uses special convolution and pooling operations and performs parameter sharing. This ensures the trained CNN model will be able to fit and run on the Raspberry Pi Board.

The V2 Plant Seedling Dataset that can be obtained from Kaggle is the main source of our training dataset[12]. It contains images of 3 kinds of crops (i.e. wheat, maize, sugar beet) and 9 kinds of weeds (i.e. black-grass, loose silky bent, etc.) seedlings. Along with the images obtained from the camera module, we separate all the images into two different classes: crop and weed. The total number of images in V2 Plant Seedling Dataset is 5,539, with a crop vs. weed image ratio of about 1:5. As the image ratio is very unbalanced, the earlier trained models have a very low classification accuracy on detecting crop type. Therefore, we balanced the crop vs. weed image ratio to roughly 1:3 by expanding the dataset through basic image processing techniques. The image processing functions include rotation, flipping, and contract/brightness adjustment, and they are built upon the opencv library.

As the images obtained from V2 Plant Seedling Dataset and camera module have different backgrounds, the function mask_img is chosen to process the images both before training and predicting. This is inspired by the project done by Noura Hussein[13] along with relative tutorial materials found on StackOverflow[14] and openCV-Python-Tutorials[15] By extracting the green color from the image, the mask_img function is able to segment the plant and mask the image background. Focusing on purely plant's shape greatly improves the model overall performance.

Figure 8 shows the final architecture of the CNN model. We finalized the design after referencing two similar image classification projects done by Noura Hussein[13] and Simran Bansari [16]. The final CNN model utilized several layers such as convolutional, max pooling, dropout and fully-connected layers. A CNN model can be thought of as a combination of two components: the feature extraction part and the classification part [17]. The convolution and pooling layers are able to extract features from images. To ensure complex features can be learned, the convolutional layers are also built on top of each other. The fully connected layers act as a classifier on top of the features, and then assign probabilities corresponding to either crop or weed types for the input images.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (320, 320, 32)            896
_________________________________________________________________
conv2d_1 (Conv2D)            (320, 320, 32)            9248
_________________________________________________________________
conv2d_2 (Conv2D)            (320, 320, 32)            9248
_________________________________________________________________
max_pooling2d (MaxPooling2D) (160, 160, 32)            0
_________________________________________________________________
dropout (Dropout)            (160, 160, 32)            0
_________________________________________________________________
conv2d_3 (Conv2D)            (160, 160, 64)            18496
_________________________________________________________________
conv2d_4 (Conv2D)            (160, 160, 64)            36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (80, 80, 64)              0
_________________________________________________________________
dropout_1 (Dropout)          (80, 80, 64)              0
_________________________________________________________________
conv2d_5 (Conv2D)            (80, 80, 128)             73856
_________________________________________________________________
conv2d_6 (Conv2D)            (80, 80, 128)             147584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (40, 40, 128)             0
_________________________________________________________________
dropout_2 (Dropout)          (40, 40, 128)             0
_________________________________________________________________
flatten (Flatten)            (204800)                  0
_________________________________________________________________
dense (Dense)                (320)                     65536320
_________________________________________________________________
dropout_3 (Dropout)          (320)                     0
_________________________________________________________________
dense_1 (Dense)              (2)                       642
=================================================================
Total params: 65,833,218
Trainable params: 65,833,218
Non-trainable params: 0
_________________________________________________________________
```

Figure 8: CNN Model Architecture

To further improve the model and avoid overfitting, we set the learning rate to be 1e-5 and the optimizer to be RMSprop, with a learning rate of 0.001, a discounting factor of 0.9, a decay of 0.0 and an epsilon of 1e-08. RMSprop is similar to the gradient descent algorithm with momentum, the difference is how the gradient is calculated as illustrated in the figure below:

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw$$

$$v_{db} = \beta \cdot v_{dw} + (1 - \beta) \cdot db$$

$$W = W - \alpha \cdot v_{dw}$$

$$b = b - \alpha \cdot v_{db}$$

Gradient descent with momenttum

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2$$

$$v_{db} = \beta \cdot v_{dw} + (1 - \beta) \cdot db^2$$

$$W = W - \alpha \cdot \frac{dw}{\sqrt{v_{dw}} + \epsilon}$$

$$b = b - \alpha \cdot \frac{db}{\sqrt{v_{db}} + \epsilon}$$

RMSprop optimizer

Figure 9: Optimizer Comparison [17]

One advantage of RMSprop optimizer is that it can speed up learning by restricting the oscillations in the vertical direction and taking larger steps in the horizontal direction to update weights and biases.

2.3 Motor Unit

The functionality of the motor unit is to achieve "homing", "searching" and "cutting". Restricted by the physical design, the mechanism is updated in Figure 7, which enables two degrees of freedom for the joint movements. The system starts at a ready position. After CNN model is loaded, the motors execute the "searching" protocol. This protocol allows the arm tip to be lowered to a position that is close to the ground. Then, the camera is initiated to take a picture and pass to CNN model, which outputs a prediction. If the weed type is classified, the motors execute the "cutting" protocol followed by "homing" to adjust back to the ready position. If it is not a weed, the motors execute another "searching" protocol, which allows the arm to dynamically adjust tip-to-ground distance while extending forward. After every adjustment, the camera takes a picture and predicts weeds or non-weeds. The microcontroller controls motors to go into "cutting" or "searching" depends on the neural

network output. "Homing" and "searching" protocols are given in Figures 22 and 23 in Appendix E as examples.

2.3.1 Gear Motors

We chose a DC 12V 7rpm (torque = 30kgcm) gear motor for the base, which is sufficient to drive the arm based on the suggestion from technicians in the machine shop. The elbow motor is a DC 12V 30rpm (max torque = 25kgcm, rated torque = 7.4kgcm) gear motor. The two motors were selected from 20+ options due to their functionality, prices and delivery time span.

2.3.2 Servo Motors

We chose a servo motor with 20kgcm torque to drive the wrist up-and-down. This is because servo motors are usually cheaper, lighter and can provide enough torque to achieve the desired functionality. We planned initially to use the same servo motor to control the motion of the scissor in order to open and close. However, the extra weight of the servo made the arm hard to support. Eventually, we switched to a lighter servo. Even though it provides less torque (1kgcm), it can balance the weight.

2.3.3 H-bridge Motor Driver

The initial gear motors were not able to turn clockwise or counter-clockwise as expected. Therefore, we added a H-bridge motor driver in order to drive the motors in both directions.

2.4 Power Unit

The power unit consists of a 12V Lithium-ion battery, a 12V-5V voltage divider and a switch. The 12V power drives gear motors, and the 5V power drives the rest of the components such as servo motors and ultrasound.

2.4.1 12 Volts Rechargeable Lithium Iron Phosphate Battery

A 12 V rechargeable Lithium Iron Phosphate Battery was chosen because it is the most ideal battery to simulate the real-world application of having a free-roaming robot.

2.4.2 DC-DC Buck Step Down Voltage Converter

Since most of the electrical components run on 5 V, we incorporated a DC-DC Buck Step Down 12V to 5V Voltage Converter to drive the rest of the circuit. A circuit of the voltage divider is given in Figure 15 in Appendix B.

3 Verification

3.1 Control Unit

The control unit is critical to communicate and coordinate with each of the components. In order to test the communication of microcontroller with individual components, we connected motors and ultrasound to the microcontroller and implemented unit tests on each component individually. After measuring the movement angle of motors, it is verified that the microcontroller can deliver commands to the motors accurately control its motion. To ensure fast and accurate the communication between the microcontroller and the ultrasound, we constantly output distance values received by the microcontroller and display them on the terminal to monitor the change.

To check the validity of the ultrasound's performance, we checked to see the module was able to update the distance at least once per second. We also ensured that it had a distance accuracy outside of the 1 cm limit. The general performance of the ultrasound module was verified by checking that it was able to precisely update the distance on-call to avoid crashing and to make adjustments on an unstable, moving robotic arm.

3.2 Recognition Unit

The default resolution of the camera module is 1920x1080, which can be easily verified from the images taken by the camera module. Instead of setting the resolution to be 625×625 as intended, we decided to set the camera resolution to be 320×320 to capture images to ensure they have the same size as images from the training dataset. As Raspberry Pi cameras need at least 2 seconds before capturing an image to sense the light levels [18], we set its sleep time to be 5 seconds to ensure camera can fully wait till arm stop moving and take focused still images.

The size of the final CNN model is 502MB, which ensures it can be easily stored on the Raspberry Pi board. By outputting time required for loading and predicting operations on the terminal while running the arm, the model has an average loading time of 7 seconds, and a predicting time of 2 seconds.

The CNN model has performed well in classifying the type of plant seedlings. As shown in the accuracy and loss graphs (Figure 10), after 150 training epochs, the model training and validation accuracies have reached above 95% (training accuracy = 0.9728, validation accuracy = 0.9551); the losses have fallen below 0.2 (training loss = 0.0786, validation loss = 0.1235).

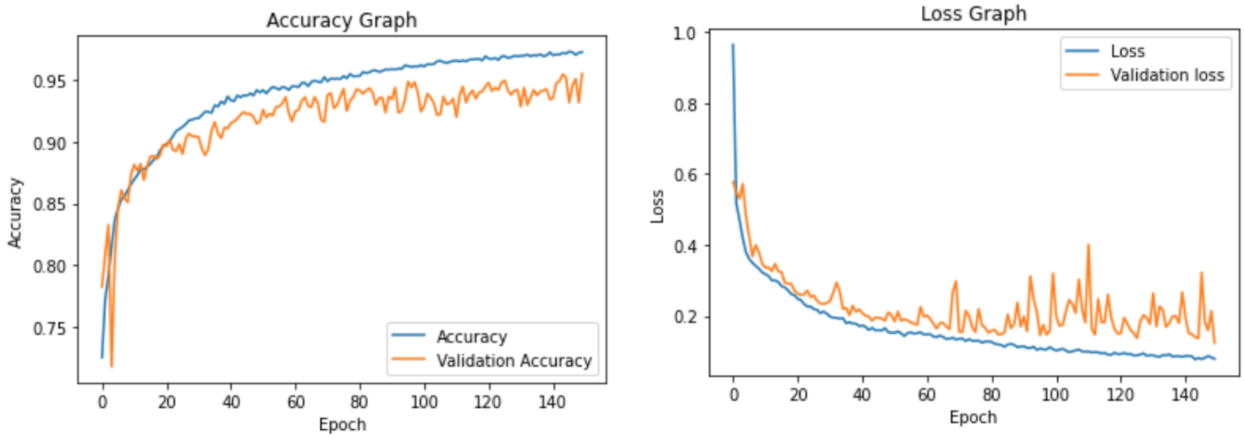


Figure 10: CNN Model Accuracy and Loss Graphs

The confusion matrix in Figure 11 shows the testing results of the model. Among 2226 testing images, only 69 of them are misclassified.

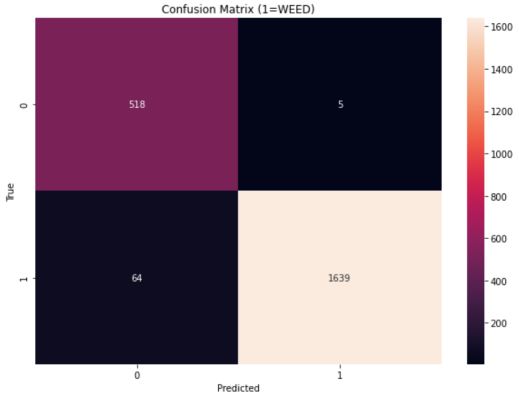


Figure 11: Confusion Matrix

3.3 Motor Unit

The motor unit is composed of two gear motors, two servo motors, and an H-bridge motor driver. The gear motors and motor driver are for the general arm movement of "homing" and "searching". The two servo motors are for "cutting".The integrated and unit testing results for each component yielded that the motors are able to move both clockwise and counter-clockwise under control. In addition, they were able to coordinate to move the arm tip to a certain position. To verify the general performance, we checked that the motors

have achieved the ability to move to a certain degree under control. The motors were also checked to ensure they could coordinate to execute the "cutting" motion.

However, because the arm was not built by professionals due to lock-down, we were not able to implement a more efficient cutting mechanism for the servo. The servo we switched to for lighter weight was not able to achieve the equivalent torque. As a consequence, the sheer can only cut the weed but was not able to it into 2 pieces. An explanation of this circumstance is shown in Figure 12.

The H-Bridge motor driver was not in the initial design document because it is a modified component we added later to enable backward motor motion. However, its performance is still critical to the control of motors. The H-bridge motor driver was first tested for its ability to drive the motors clock- and counterclock-wise. The H-Bridge was also controlled through the microcontroller code to ensure that it can execute the designated protocols. Finally, the H-Bridge was able to distribute signals with 5V and motors with 12 V. With these verifications, we concluded that the H-Bridges were functional for our purpose.

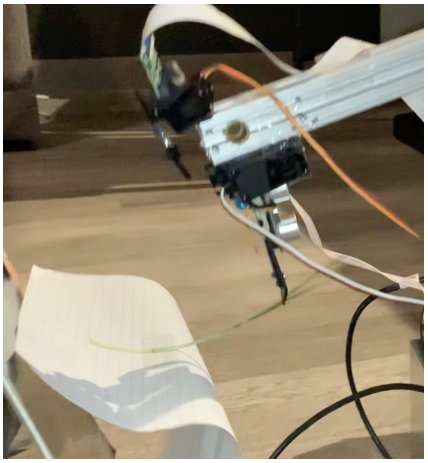


Figure 12: Testing Code of Camera Resolution/Speed

4 Cost

Physical Parts	Unit Cost	Quantity	Subtotal Cost
SunFounder 20KG Servo Motor Waterproof High Torque Servo, SF3218MG Metal Gear Digital Servo	$14.99	1	$14.99
WINGONEER 10Pcs MG90S Metal Gear RC Micro Servo for ZOHD Volantex Airplane RC Helicopter Car Boat Model	$1.99	1	$1.99
Yosoo High Torques Worm Geared Motor DC 12V Reduction Motor with Encoder Strong Self-locking	$29.17	1	$29.17
uxcell DC 12V 7RPM 30Kg.cm Self-Locking Worm Gear Motor with Encoder and Cable, High Torque Speed Reduction Motor	$34.99	1	$34.99
Arducam 5 Megapixels 1080p Sensor OV5647 Mini Camera Video Module for Raspberry Pi Model	$12.99	1	$12.99
Sheer from ECE Supply Shop	$10	1	$10
Raspberry Pi 4 2GB with CanaKit Power Supply	$49.04	1	$49.04
SanDisk 64GB Ultra MicroSDXC UHS-I Memory Card	$19.61	1	$19.61
HiLetgo 5pcs DC-DC Buck Step Down Voltage Module 6-24V 12V/24V to 5V 3A	$7.59	1	$7.59
SainSmart HC-SR04 Ranging Detector Mod Distance Sensor	$4.95	1	$4.95
Mighty Max Battery ML9-12 12V 9Ah Rechargeable SLA Battery	$21.99	1	$21.99
			Subtotal: $207.31

Table 1. Physical Parts Cost

Labor Type	**Number of Workers**	**Hourly Cost**	**Total Hours**	**Subtotal Cost**
ECE Machine Shop Machinist	1	$19 [19]	35 hrs	$665
Team 9 ECE Engineers	3	$50	8 hrs × 15 weeks	$18,000
				Subtotal: $18,665

Table 2. Labor Cost

Combining the physical parts cost and the labor cost, the grand cost for the entire project is **$18872.3**.

5 Conclusion

5.1 Accomplishments and Limitations

Ultimately, we were able to differentiate weed and crop seedlings through a CNN model trained on V2 Plant Seedling Dataset and images obtained from the Raspberry Pi Camera module. The above 95% accuracy rate of the CNN model ensured accurate weeds classification. Furthermore, two degrees of freedom was sufficient to execute motor functions. The "cutting" and "homing" mechanisms were successfully implemented through the coordination of motor, ultrasound and camera modules.

We were unable to fully cut through the weeds due to the limitation of servo motor which drove the cutting motion. We foresee solving this problem by having professionals build the mechanical parts.

5.2 Ethics and Safety

The IEEE Code of Ethics [20] states that it is the responsibility in our profession that we hold ourselves to high ethical standards. This includes respecting others' intellectual property, ensuring the safety of others, and strive to implement ethical designs.

1.To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;
Our project aims to achieve 100% mechanical weeding to replace chemical weeding, in order to promote sustainability. The sheer we attached to the arm was also under safety concern. With safety in mind, we trained the arm without the blade first and followed by adding the blade with the cutting mechanism locked. Finally, we added the cutting function to let it execute the full cutting motion.

2.To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
We do not have any conflicts of interest and pursued this project purely out of passion.

3.to be honest and realistic in stating claims or estimates based on available data;
All calculation and implementation are honestly presented based on our own results or acquired open-source dataset. If a certain plan did not work, we quickly adapted alternative methods to achieve the same functionalities.

4.To reject bribery in all its forms;

We did not participate in any bribery during the process of this project.

5.To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;
The project is inspired to progress the chemical-weeding industry into automation. We are also willing to provide any detailed description of the robotic design to all potential users for their safety and interests.

6.To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
As ECE students, we used our technical and general problem-solving skills to build this robot. As we have built the entire system from scratch (mechanics, hardware, software), we have learned multiple restraints and trade-offs from the project. We envision to improve these skills in future education and careers.

7.To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
We have sought and accepted criticism from TAs, peers and professors. We also appreciate the help we received from the ECE machine shop and our families.

8.To treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;
We have never discriminated people based on their races, religions, gender, disabilities, ages, national origin, etc.
9.To avoid injuring others, their property, reputation, or employment by false or malicious action;
We have taken precautions to avoid injury in our project. We have properly cited our sources, especially our neural network training dataset and code of reference.

10.To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.
We take our peers' professional development as seriously as we do ours and we treat them well based on these IEEE Code of Ethics.

5.3 Project Improvements

Although we were able to meet most of our design and implementation goals, there are a few improvements to make for future commercialization:

1.Raspberry Pi to NVIDIA Jetson Nano
An improvement would be to use the NVIDIA Jetson Nano instead of the Raspberry Pi to support the neural network. Upgrading our board will improve processing and computing power. More computing power provides the possibility to host multiple cameras in the future for real-time monitoring and more accurate plant locationing.

2.Expand training dataset
As we are currently heavily relying on V2 Plant Seedling Dataset[12], which only has three types of crop images. To increase the diversity of the plant types in the training dataset, we would also like to expand our training dataset to increase the amount and types of plants. Ideally, we can also obtain more images from the actual fields, and especially from the robot's view.

3.Integrate into existing product
Another goal is to integrate the arm into an existing robotic base. This will allow us to have a moving bot capable of autonomously finding and cutting weeds in a plot.

4.Upgraded mechanical design
We also want to improve the mechanical design to make the movements more accurate and seamless. The current issue we are facing is that the arm and motor shaft don't move accordingly due to the heavy weight at the arm tip. Although it is temporally fixed through Gorilla glue, a stronger joint design can definitely improve movement accuracy. The first mechanical improvement is to create a single piece locked T-shaft extender for the base motor [Figure 22] to avoid slipping of the arm from the motor shaft. The second improvement is to include potentiometers outside of the joints. This is because although shaft movement is tracked through the encoder, it does not accurately update location information of the joint. By incorporating potentiometers at the end-effecter of the motion, better homing and searching mechanism can be achieved.

5.Additional sensors
We could also implement real-time positioning through extra sensors. This will allow for better supervision and monitoring of the robot's actions. For example, by adding more cameras, we can avoid "blind" searching, which is what currently the robot is doing. More ultrasound sensors can also make the robot move safer and smoother. However, implementing

additional sensors has a prerequisite that the arm needs to be strong enough to host them, which requires an upgraded mechanical design.

6.Refined cutting mechanism

By adding a ball-bearing system to open and close the sheers instead of a direct servo shaft, we will improve the accuracy and power as well as the safety of the mechanism. Due to the lack of necessary tools and mechanical experience, we were not able to do that. However, with the help of machine shop, we envision us to improve it to truly cut through the weeds.

References

[1] E. Dixon, "Common weed killer glyphosate increases cancer risk by 41%, study says," CNN, 15-Feb-2019. [Online]. Available:https://www.cnn.com/2019/02/14/health/us-glyphosate- cancer-study-scli-intl/index.html.

[2] A. H. C. Van Bruggen, M. M. He, K. Shin, V. Mai, K. C. Jeong, M. R. Finckh, and J. G. Morris, "Environmental and health effects of the herbicide glyphosate," Science of The Total Environment, vol. 616-617, pp. 255–268, Mar. 2018. Available: https://doi.org/10.1016/j.scitotenv.2017.10.309

[3] "How Herbicides Work". US Fish Wildlife Service [Online]. Available: https://www.fws.gov/invasives/stafftrainingmodule/methods/chemical/impacts.html

[4] "Hazards of Herbicides", SF Gate Business. [Online]. Available: https://homeguides.sfgate.com/hazards-herbicides-groundwater-78877.html

[5] "Hazards of the World's Most Common Herbicide", MotherEarthNews.[Online]. Available: https://www.motherearthnews.com/organic-gardening/hazards-of-worlds-most-common-herbicide-zmaz05onzsel

[6] "Formula Calibration Method," Pesticide Environmental Stewardship. [Online]. Available: https://pesticidestewardship.org/calibration/formula-calibration-method/.

[7] "Big Ag's Dirty Little Secret," Pesticide Action Network. [Online]. Available: https://www.panna.org/gmos-pesticides-profit/big-ags-dirty-little-secret.

[8] G. Schnitkey, "Historic Fertilizer, Seed, and Chemical Costs with 2019 Projections," farmdoc daily, 05-Jun-2018. [Online]. Available: https:// farmdocdaily.illinois.edu/2018/06/historic-fertilizer-seed-and-chemical-costs.html.

[9] "The autonomous robot weeder from Ecorobotix," Ecorobotix. [Online]. Available: https://www.ecorobotix.com/en/autonomous-robot-weeder/.

[10] "EarthSense, Inc.," EarthSense, Inc. [Online]. Available: https://www.earthsense.co/.

[11]Arducam 5MP Mini Camera OV5647 Sensor 1080p 720p video for Raspberry Pi 4/3B+/3 Camera official Board. [Online]. Available: https://www.arducam.com/product/arducam-ov5647-standard-raspberry-pi-camera-b0033/

[12] Marsh, "V2 Plant Seedlings Dataset," Kaggle, 13-Dec-2018. [Online]. Available:

https://www.kaggle.com/vbookshelf/v2-plant-seedlings-dataset105.png.

[13] N. Hussein, "Nourahussein/Plant-seedling-classification," GitHub, 10-Jun-2018. [Online]. Available: https://github.com/Nourahussein/Plant-seedling-classification.

[14] "OpenCV Android Green Color Detection". [Online]. Available: https://stackoverflow.com/questions/31590499/opencv-android-green-color-detection

[15] Morphological Transformations. [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_

[16] S. Bansari, "Plant Seedlings Classification using CNNs," Medium, 24-Apr-2019. [Online]. Available: https://becominghuman.ai/plant-seedlings-classification-using-cnns-ea7474416e65.

[17] R. Gandhi, "A Look at Gradient Descent and RMSprop Optimizers," Medium, 19-Jun-2018. [Online]. Available: https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b.

[18] "Getting started with the Camera Module," projects.raspberrypi.org. [Online]. Available: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/5.

[19] "Hourly Rate for Industry: Machine Shop," PayScale. [Online]. Available: https://www.payscale.com/research/US/Industry=Machine_Shop/Hourly_Rate

[20] "IEEE Code of Ethics," IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

Appendix A Requirement and Verification Table

The requirement and verification for each unit and component are shown below.

Table 3: System Requirements and Verifications

Requirement	Verification
Control Unit: Microcontroller 1. The microcontroller must be able to receive electrical signal 2. The signal transmission between microcontroller and therecognition unit can be indicated by the LED	1. To test the microcontroller can receive electrical signal: (a) Connect the microcontroller to a battery-voltage converter and verify with a multimeter to ensure current flows through 2. To test LED can indicate detecting status: (a) Check the LED remains on when running pictures from weed databases through neural network, and off when running pictures from non-weed databases through neural network (b) Mix weed and non-weed pictures. Run the pictures through neural network one-by-one (c) Record true positive, false positive, true negative and false negative rates
Control Unit: Ultrasound 1. The ultrasound must be able to detect the distance to an obstacle with an error margin within ±3mm	1. To test distance detection: (a) Microcontroller code for ultrasound module is free of bugs. (b) Experiment each ultrasound module by drawing a line on the table, which is 3cm away from a wall. Hand-hold each module (connected to Raspberry Pi and power), and move it from a distance larger than 3cm to the line (c) If Raspberry Pi successfully output 3cm±3mm at the line, the ultrasound module is working properly

Continued on next page

Table 3 – continued from previous page

Requirement	Verification
Recognition Unit: Camera 1. The camera must be able to capture clear images with a resolution greater than 625×625 to be qualified for the neural network dataset images. 2. The camera must be able to capture the image in 6 seconds.	1. To test the camera vision: (a) Connect the camera to Raspberry pi through ribbon cable (b) Take 50 pictures with the camera with default resolution (c) Verify that at least 95% of the pictures have resolution over 625×625 2. To test the speed of photo capturing: (a) Connect the camera to Raspberry pi through ribbon cable. (b) Calculate the time of taking a single image.
Recognition Unit: Raspberry Pi 1. Raspberry Pi can connect the camera module to the computer screen for testing and streaming within a delay $\leq$ 1s. 2. The ethernet communication speed is above 10MB/s	1. To test the signal delay: (a) After connecting the Raspberry Pi to computer through USB port, enable camera module through Python Record the monitor screen to measure the time needed for view shifting by adjusting camera orientation 2. To test ethernet speed: (a) Test the communication speed through terminal by transferring large files (1GB) through USB port
Recognition Unit: Neural Network 1. The model must reach a classification accuracy above 75% 2. The size of the neural net model is reasonable to be transferred to Raspberry Pi	1. To test classification accuracy: (a) Verify that prediction loss$<$ 0.8, classification accuracy$>$75% by the end of the training 2. To test neural network size: (a) The size of neural network model is less than size of the Raspberry Pi SDRAM storage (1GB)

Continued on next page

Table 3 – continued from previous page

Requirement	Verification
Power Unit: 12 Volts Rechargeable Lithium Iron Phosphate Battery 1. The battery must be able to distribute 12V of power to motors	1. To test the battery output: (a) Fully charge the battery (b) Disconnect from charger (c) Connect to a multimeter and measure voltage output (d) Connect the multimeter to LabView and monitor the voltage for 3hrs to verify a constant output of 12V with a margin of error of 5%
Motor Unit: Gear Motors & Servo Motors 1. The motors must be able to achieve "homing" 2. The motors must be controlled by microcontroller to achieve "cutting" mechanism	1. To test whether the motors can implement "homing" mechanism: (a) Program and compile code specifying a "homing" position (b) Load the code to microcontroller (c) Connect the microcontroller to each motor (d) Connect battery to motor and battery-voltage converter to microcontroller (e) Verify whether the motor-driven-arm moves to the "homing" position 2. To test "cutting" mechanism: (a) Program code to conduct "cutting" mechanism (b) Load the code to microcontroller (c) Connect the microcontroller to each motor (d) Run the program to cut the weed

Appendix B Hardware Design and Schematics

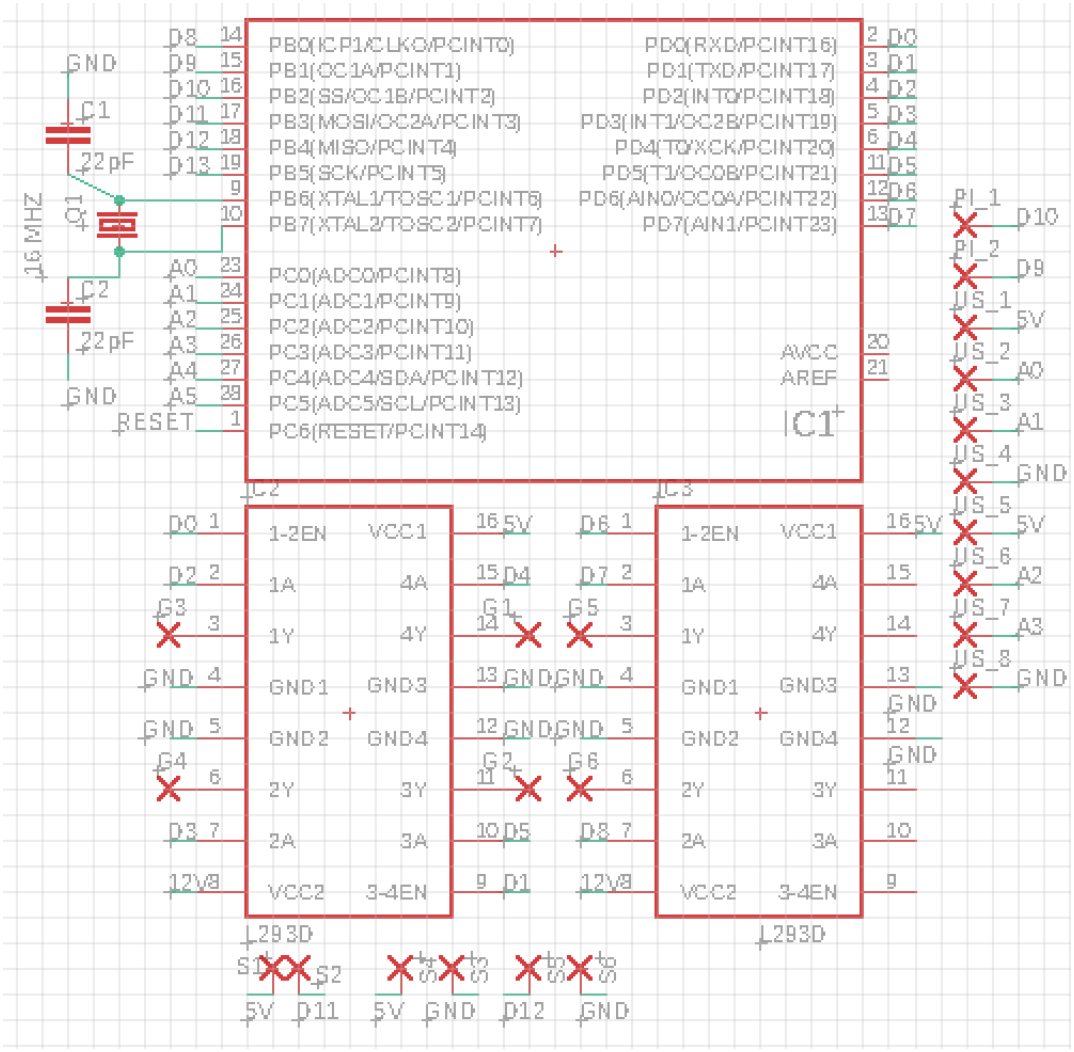
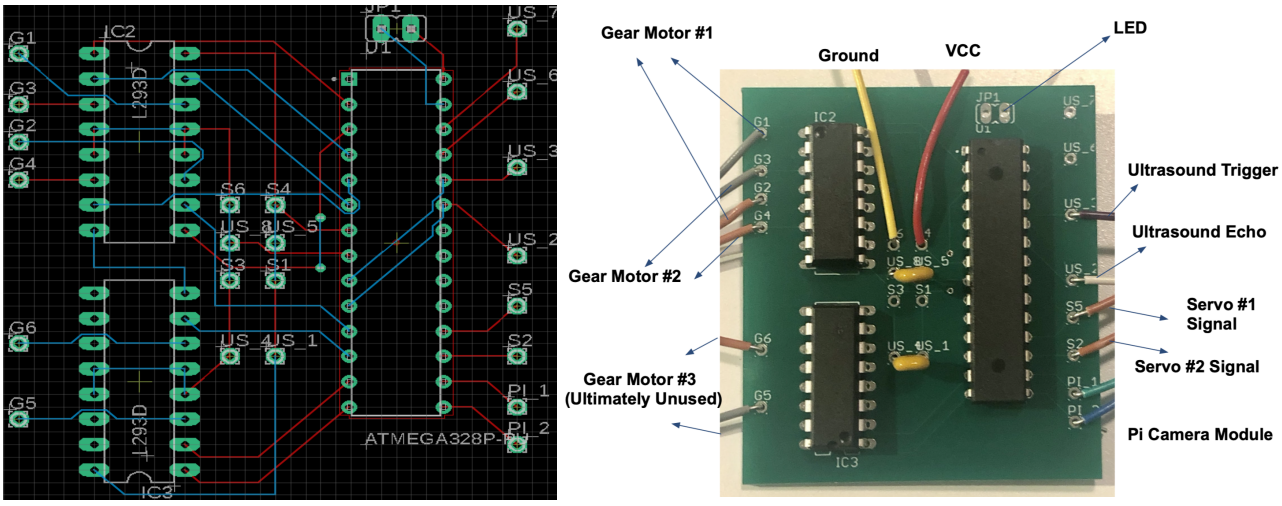


Figure 13: PCB Schematics



(a) PCB footprints

(b) Fabricated board

Figure 14: PCB

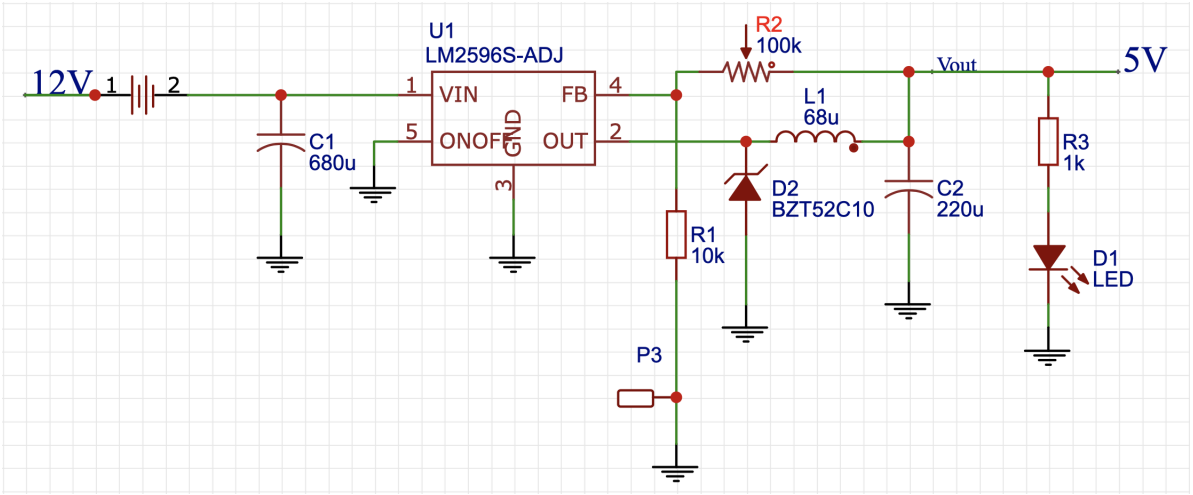


Figure 15: Voltage Converter Schematics

Appendix C Software Code for Recognition Unit

```python
import time
from picamera import PiCamera
def take_photo(number, png_path, pixel_resolution=320):
    camera = PiCamera()
    camera.resolution = (pixel_resolution, pixel_resolution)
    camera.start_preview()
    for i in range(number):
        time.sleep(5) # >2 to capture an image
        png_name = png_path + "pi_image" + str(i) + ".png"
        camera.capture(png_name)
    camera.stop_preview()
    camera.close()
```

Figure 16: Camera Module Code for Taking Photos and Real-time Monitoring

```python
alpha_beta_set = [(1.5, 0), (0.8, 0), (1, 20), (1, -10), (1.5, 20), (0.8, -10)]
rotate_angle_set = [90, 180, 270]
flip_axis_set = [0, 1, -1]

# random set contrast/brightness, rotation angle and flip axis
for i in range(random_times):
    rand_idx1 = random.randint(0, len(alpha_beta_set)-1)
    rand_idx2 = random.randint(0, len(rotate_angle_set)-1)
    rand_idx3 = random.randint(0, len(flip_axis_set) -1)
    rand_a, rand_b = alpha_beta_set[rand_idx1]
    rand_angle = rotate_angle_set[rand_idx2]
    rand_flip = flip_axis_set[rand_idx3]

    processed_img = image_processing(img_arr, alpha=rand_a, beta=rand_b,
                                     rotate_angle=rand_angle, flip_axis=rand_flip)
```

Figure 17: Code of Random Data Augmentation on a Single Image

```python
def mask_img(img, sensitivity=35, return_mask=False):

    image_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

    lower_hsv = np.array([60 - sensitivity, 100, 50])
    upper_hsv = np.array([60 + sensitivity, 255, 255])

    kernel = np.ones((8, 8),np.uint8)
    mask = cv2.inRange(image_hsv, lower_hsv, upper_hsv)
    mask = cv2.dilate(mask, kernel,iterations = 1)

    res = cv2.bitwise_and(img, img, mask=mask)

    if return_mask == True:
        return mask, res
    else:
        return res
```

Figure 18: mask_img function

Appendix D Sample Image Output for Recognition Unit

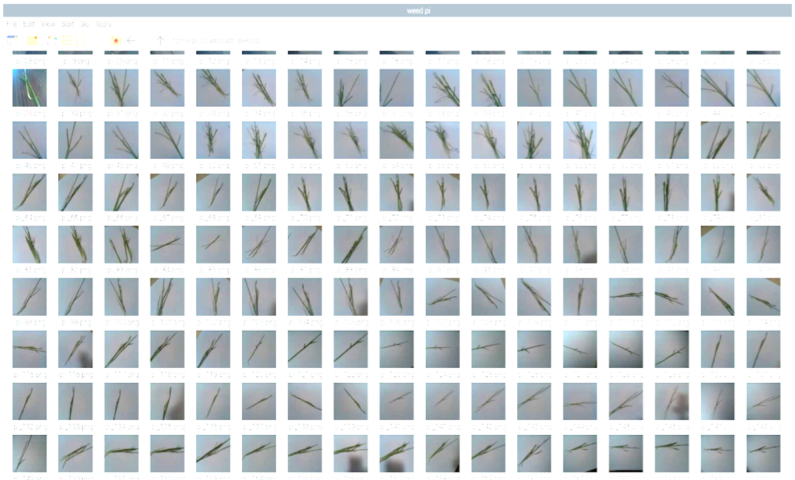


Figure 19: Sample Images Obtained From Raspberry Pi Camera Module

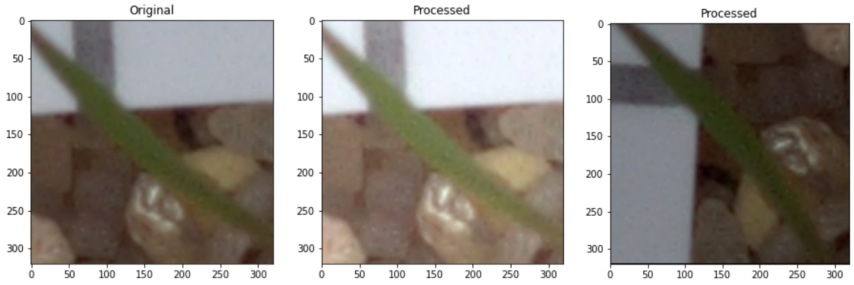


Figure 20: Example of Randomly Processed Images

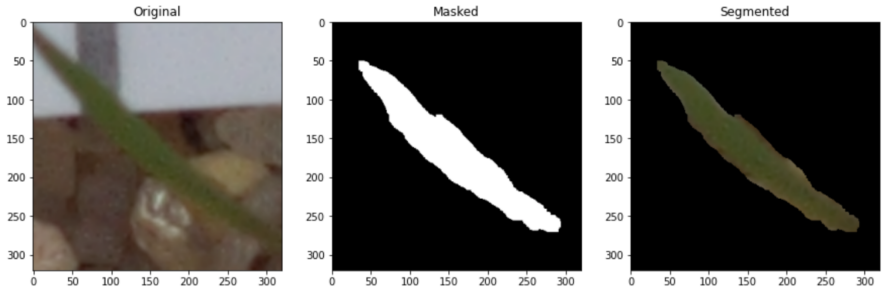


Figure 21: Sample Output of mask_img

Appendix E Code for Motor Unit Functions

```
void loop(){
  if (Serial.available() > 0) {
    int pi = Serial.read();
  }

  if (pi.equals(2)){ // start
    ultrasound();
    delay(2000);
    Serial.print("ultrasound distance1: ")
    Serial.println(distance1);
    if (distance1 >= 15){
      search()
    }
    if (distance1 > 10 && distance1 < 15){
      elbowf();
      delay(500);
      brake();
      delay(2000);
    }
    Serial.println("ready");
  }
  else if (pi.equals(0)){     //not weed or no plant detected
    ultrasound();
    delay(2000);
    Serial.print("ultrasound distance1: ");
    Serial.println(distance1);
    if (distance1 >= 15){
      search()
    }
    if (distance1 > 10 && distance1 < 15){
      elbowf();
      delay(500);
      brake();
      delay(2000);
    }
    brake();
    Serial.println("next");

  }
  else if (pi.equals(1)){     //weed
    ServoR.write(180);
    ServoR.write(140);
    home();
    Serial.println("home");
  }
  else{
    brake();
  }
}
```

Figure 22: Motor Unit Functions: Detect

```
void home(){
  //move to the back position
  ServoBangle = 0;
  ServoB.write(ServoBangle);
  baseb();
  delay(5000);
  elbowb();
  delay(2000);
  brake();
  delay(5000);

  //move to ready position
  ServoBangle = 100;
  ServoB.write(ServoBangle);
  basef();
  delay(3600);
  brake();
  delay(5000);
}
```

```
void search(){
  basef();
  delay(180);
  brake();
  delay(2000);
  elbowf();
  delay(180);
  brake();
  delay(2000);
  sec++;
  ultrasound();
}
```

```
void basef(){
  digitalWrite(pwme,LOW);
  digitalWrite(pwmb,HIGH);
  digitalWrite(g_b_pa,LOW);
  digitalWrite(g_b_pb,HIGH);
}

void baseb(){
  digitalWrite(pwme,LOW);
  digitalWrite(pwmb,HIGH);
  digitalWrite(g_b_pa,HIGH);
  digitalWrite(g_b_pb,LOW);
}

void elbowb(){
  digitalWrite(pwmb,LOW);
  digitalWrite(pwme,HIGH);
  digitalWrite(g_e_pa,HIGH);
  digitalWrite(g_e_pb,LOW);
}

void elbowf(){
  digitalWrite(pwmb,LOW);
  digitalWrite(pwme,HIGH);
  digitalWrite(g_e_pa,LOW);
  digitalWrite(g_e_pb,HIGH);
}
void brake(){
  digitalWrite(pwmb,HIGH);
  digitalWrite(pwme,HIGH);
  digitalWrite(g_b_pa,HIGH);
  digitalWrite(g_b_pb,HIGH);
  digitalWrite(g_e_pa,HIGH);
  digitalWrite(g_e_pb,HIGH);
}
```

Figure 23: Motor Unit Functions

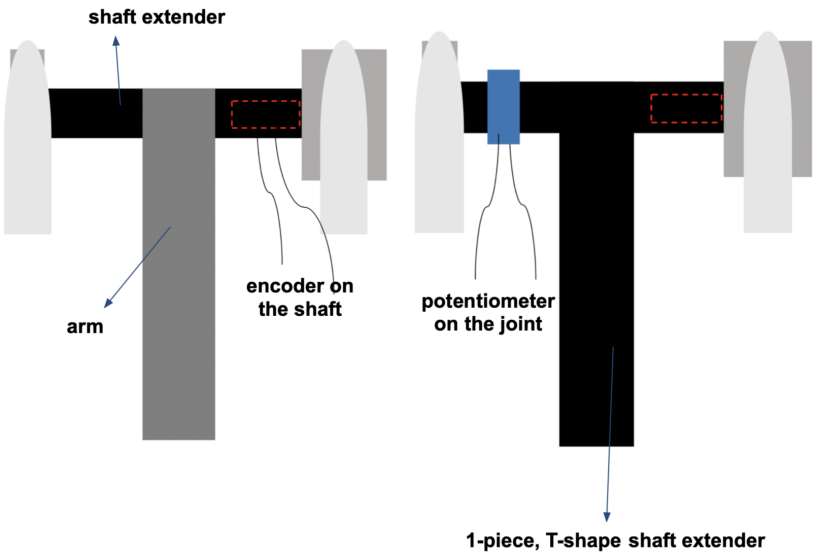


Figure 24: Motor improvement explanation