Industrial Internship Report on

”Crop and Weed Detector”

Prepared by

Tanishq Deepak Kokane

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. My project was (To identify and distinguish either there is crop or weed so that spraying system can avoid spraying on weed) This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

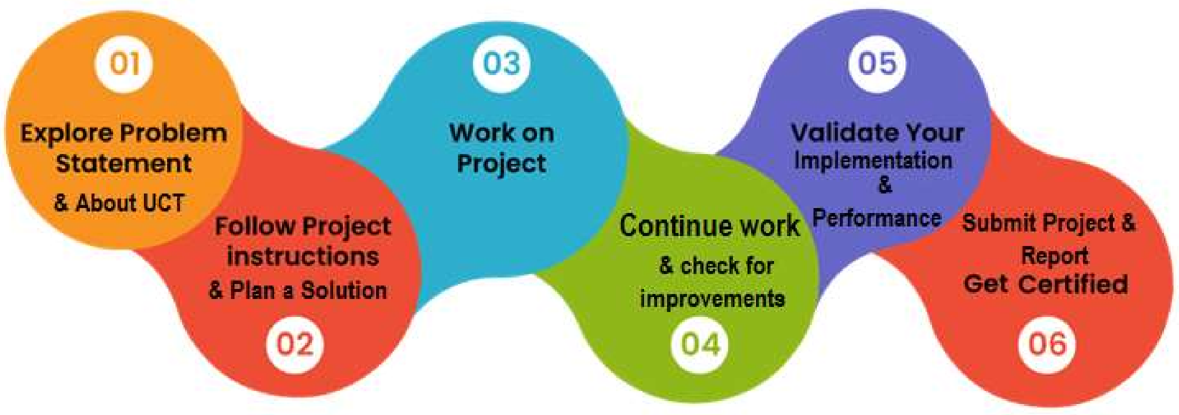
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



01
Explore Problem Statement & About UCT
02
Follow Project instructions & Plan a Solution
03
Work on Project
04
Continue work & check for improvements
05
Validate Your Implementation & Performance
06
Submit Project & Report Get Certified

Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

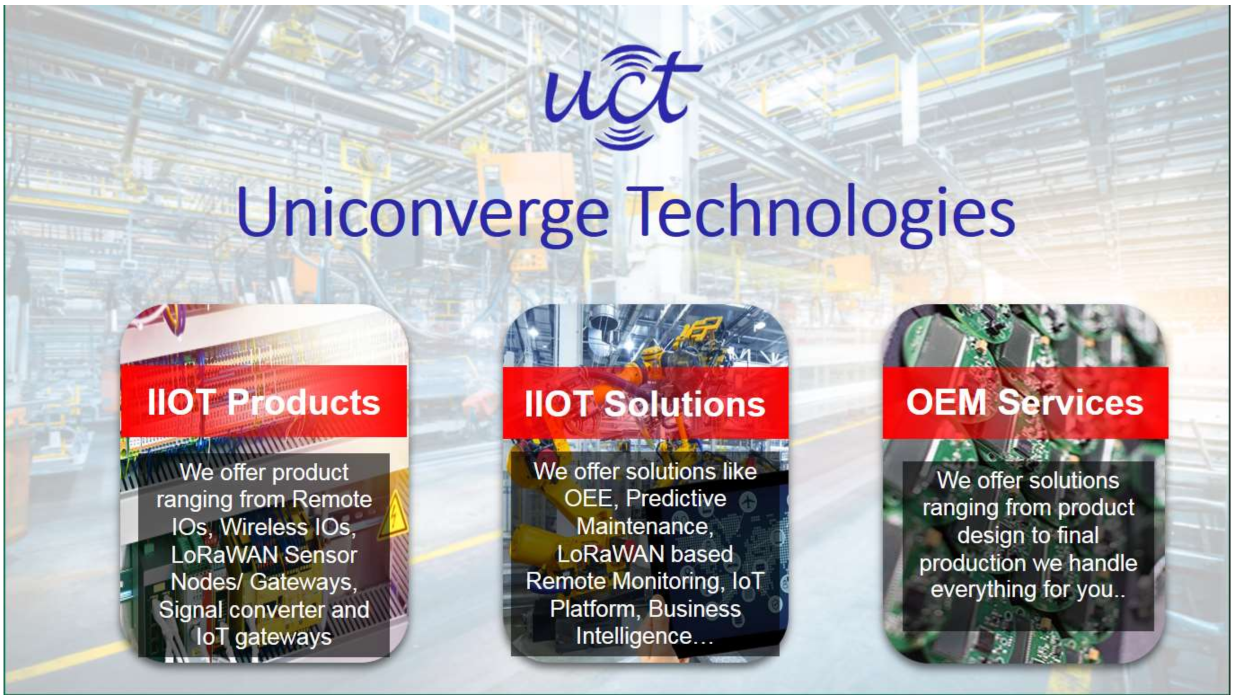
Your message to your juniors and peers.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.



uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

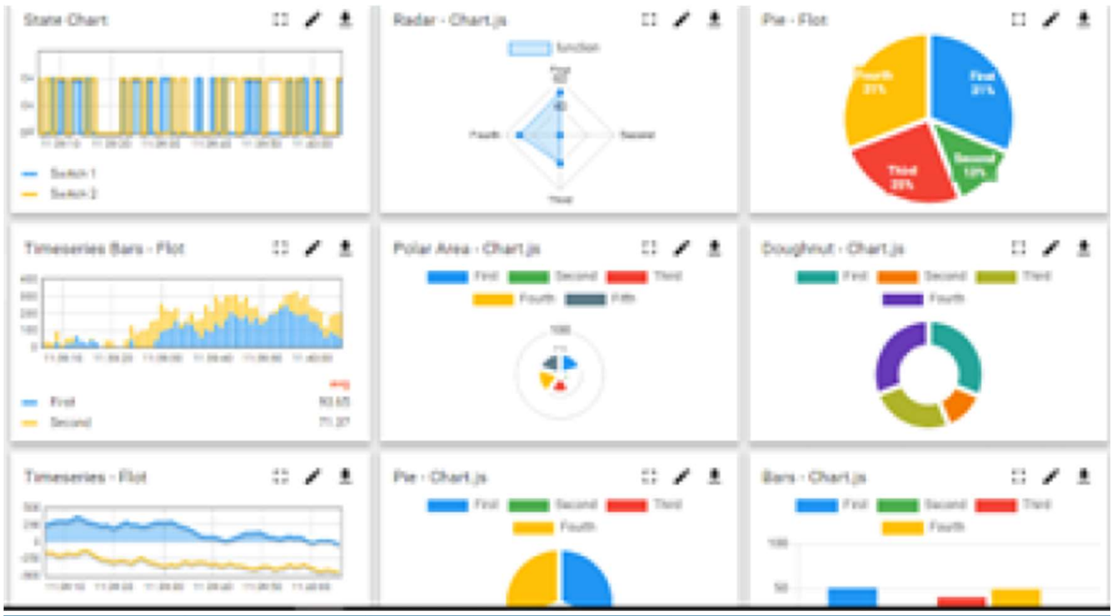
i. UCT IoT Platform (uct Insight)

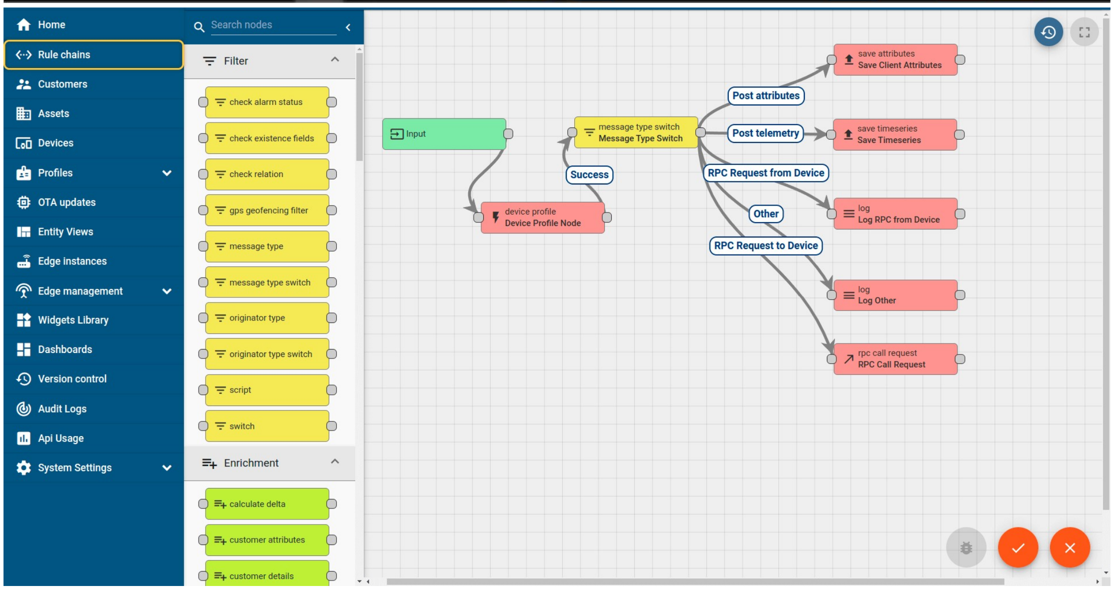
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





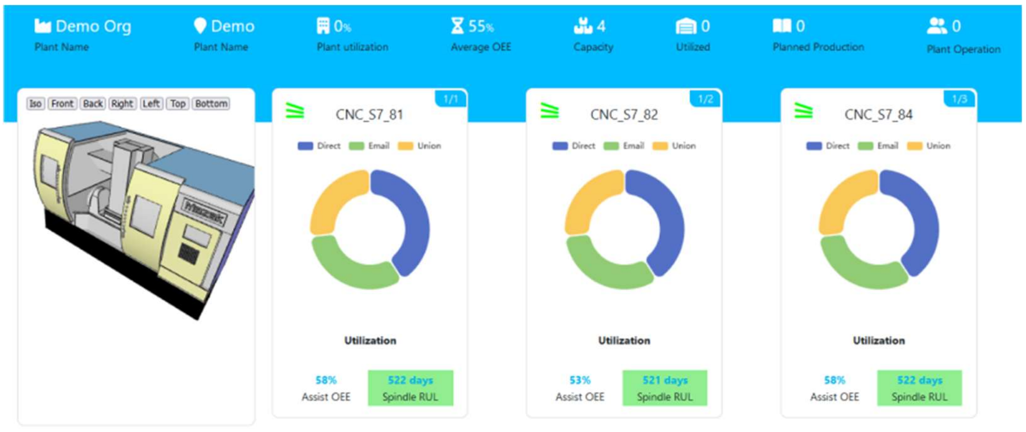
ii. Smart Factory Platform (FACTORY WATCH)

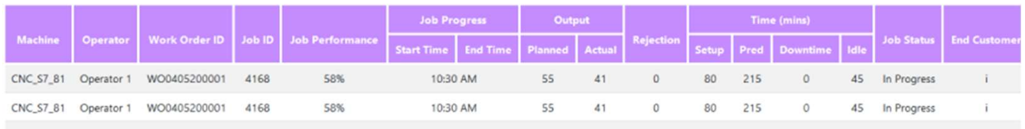
Factory watch is a platform for smart factory needs.

It provides Users/ Factory

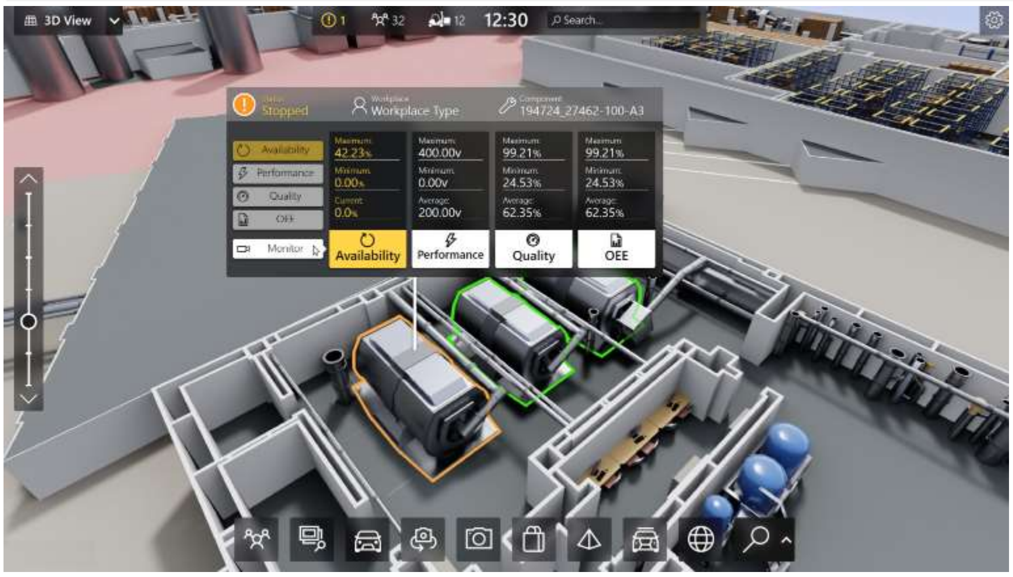
- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.





Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

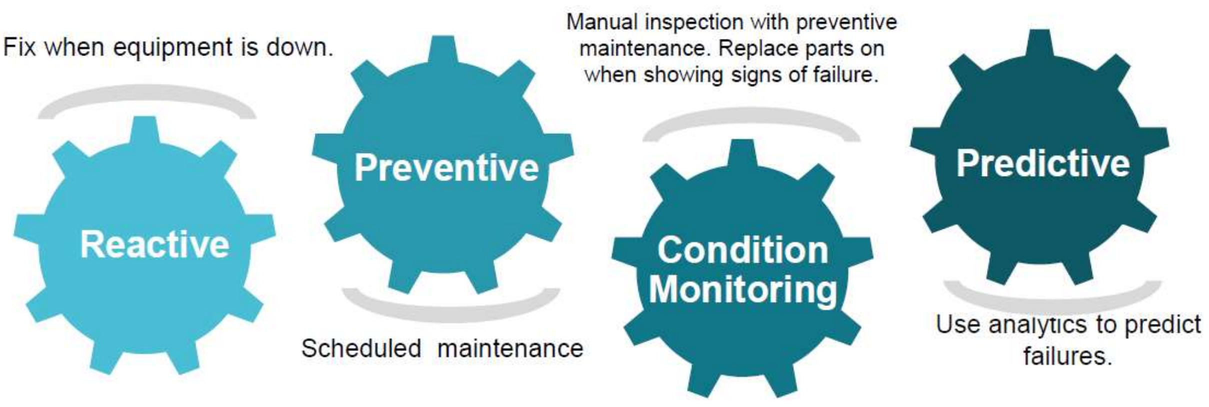


iii. LoRaWAN™ based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

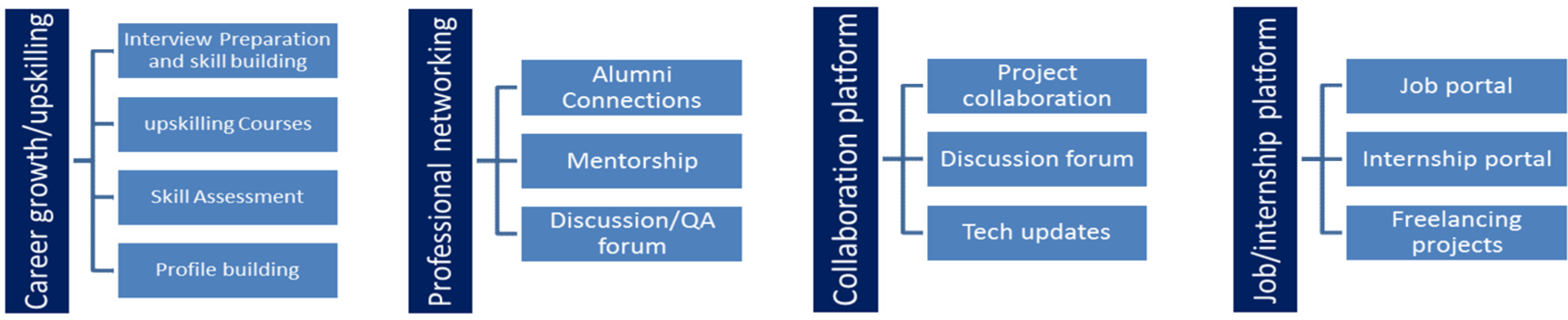
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] Github

[2] ChatGPT

[3] Youtube

3 Problem Statement

In the assigned problem statement

Weed is an unwanted thing in agriculture. Weed use the nutrients, water, land and many more things that might have gone to crops. Which results in less production of the required crop. The farmer often uses pesticides to remove weed which is also effective but some pesticides may stick with crop and may causes problems for humans.

We aim to develop a system that only sprays pesticides on weed and not on the crop Which will reduce the mixing problem with crops and also reduce the waste of pesticides.

3.1 GitHub Link

https://github.com/tanishqk26/Weed-Detector.git

4 Existing and Proposed solution

. System Overview:

- **Components:**

  - **Imaging System:** High-resolution cameras or drones equipped with multispectral or RGB cameras to capture images of the fields.

  - **Processing Unit:** A powerful computing unit (e.g., edge devices, GPUs) to process images in real-time.

  - **Spraying Mechanism:** Precision sprayers controlled by the processing unit to target specific areas.

** Machine Learning Model:**

- **Data Collection:**

  - Collect a diverse dataset of images from different crop fields, covering various growth stages, lighting conditions, and weed species.

- **Data Annotation:**

  - Manually label the images to distinguish between crops and weeds.

- **Model Training:**

  - Choose an appropriate model architecture (e.g., Convolutional Neural Networks, YOLO, Faster R-CNN) for object detection and classification.

  - Train the model using the annotated dataset, ensuring robust performance across different scenarios.

- **Model Evaluation:**

  - Validate the model on a separate dataset to evaluate accuracy, precision, recall, and F1 score.

  - Perform cross-validation to ensure the model's generalizability.

. Real-time Processing:

- **Image Acquisition:**

  - Capture real-time images using the imaging system.

- **Preprocessing:**

  - Apply preprocessing techniques like normalization, resizing, and data augmentation to improve model performance.

- **Inference:**

  - Run the trained model on the acquired images to identify and classify crops and weeds.

  - Use bounding boxes or masks to locate the weeds precisely.

. Precision Spraying:

- **Sprayer Control:**

  - Integrate the processing unit with the sprayer mechanism.

  - Use the coordinates of detected weeds to direct the sprayers accurately.

- **Spraying Optimization:**

  - Optimize the spraying mechanism to adjust the amount and direction of pesticide application, minimizing wastage.

4.1 Code submission (Github link)

4.2 Report submission (Github link) : first make placeholder, copy the link.

5 Proposed Design/ Model

. Data Collection and Preparation:

- Dataset Acquisition:Collect images of crops and weeds from various sources or capture them using cameras in different conditions and growth stages.

- **Data Annotation:** Label the collected images with the correct classifications (crop or weed).

- **Preprocessing:** Apply image augmentation techniques such as rotation, scaling, and flipping to increase the dataset size and variability. Normalize the images for consistent input to the model.

5.1

5.2 **2. Machine Learning Model:**

5.3 - **Model Selection:** Choose a suitable convolutional neural network (CNN) architecture such as ResNet, VGG, or MobileNet, which are effective for image classification tasks.

5.4 - **Training the Model:**

5.5 - **Splitting Data:** Divide the dataset into training, validation, and test sets.

5.6 - **Training Process:** Train the CNN model on the training set, using the validation set to tune hyperparameters and avoid overfitting.

5.7 - **Loss Function and Optimization:** Use a categorical cross-entropy loss function and an optimizer like Adam or SGD (Stochastic Gradient Descent).

5.8

5.9 **3. System Integration:**

- **Real-Time Image Capture:** Integrate cameras with the agricultural machinery to capture images of the field in real-time.

5.10 - **Image Processing Pipeline:**

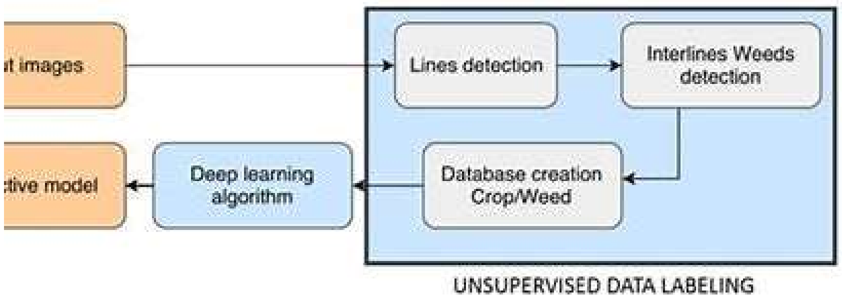
5.11 - **Image Acquisition:** Capture images at regular intervals or continuously.

5.12 - **Prediction:** Feed the captured images into the trained CNN model to classify each pixel or region as either crop or weed.

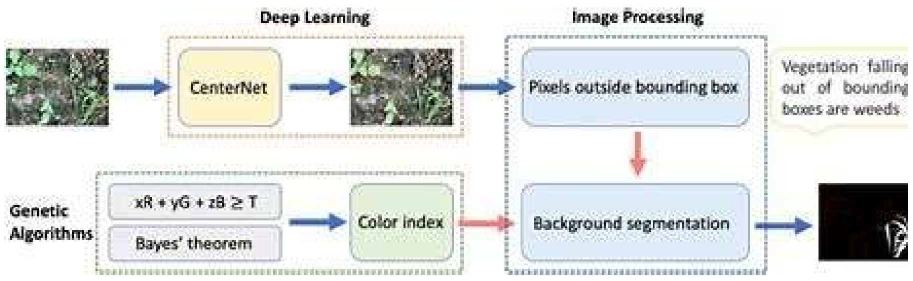
5.13 - **Localization:** Use techniques like bounding boxes or segmentation to precisely locate

5.14 - **Spraying MechanHigh Level Diagram (if applicable)

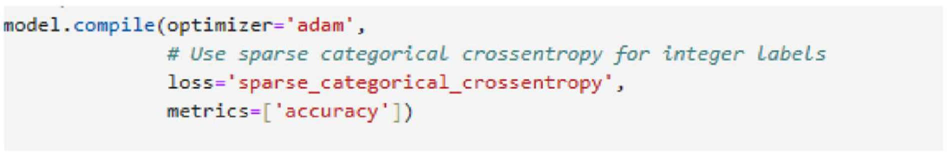
5.15 Interfaces (if applicable)



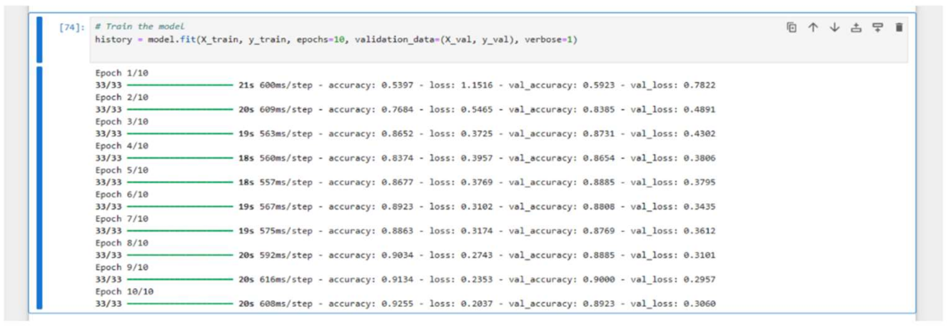
ıt images
Lines detection
Interlines Weeds detection
ctive model
Deep learning algorithm
Database creation Crop/Weed
UNSUPERVISED DATA LABELING



6 Performance Test



```
model.compile(optimizer='adam',
              # Use sparse categorical crossentropy for integer labels
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

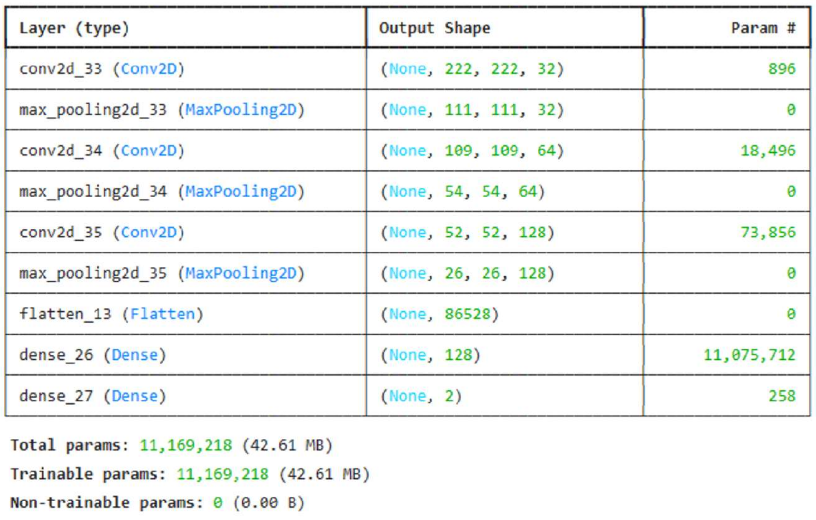


```
[74]: # Train the model
      history = model.fit(X_train, y_train, epochs=10, validation_data=(X_val, y_val), verbose=1)

      Epoch 1/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 21s 600ms/step - accuracy: 0.5397 - loss: 1.1516 - val_accuracy: 0.5923 - val_loss: 0.7822
      Epoch 2/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 20s 609ms/step - accuracy: 0.7684 - loss: 0.5465 - val_accuracy: 0.8385 - val_loss: 0.4891
      Epoch 3/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 19s 563ms/step - accuracy: 0.8652 - loss: 0.3725 - val_accuracy: 0.8731 - val_loss: 0.4302
      Epoch 4/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 18s 560ms/step - accuracy: 0.8374 - loss: 0.3957 - val_accuracy: 0.8654 - val_loss: 0.3806
      Epoch 5/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 18s 557ms/step - accuracy: 0.8677 - loss: 0.3769 - val_accuracy: 0.8885 - val_loss: 0.3795
      Epoch 6/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 19s 567ms/step - accuracy: 0.8923 - loss: 0.3102 - val_accuracy: 0.8808 - val_loss: 0.3435
      Epoch 7/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 19s 575ms/step - accuracy: 0.8863 - loss: 0.3174 - val_accuracy: 0.8769 - val_loss: 0.3612
      Epoch 8/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 20s 592ms/step - accuracy: 0.9034 - loss: 0.2743 - val_accuracy: 0.8885 - val_loss: 0.3101
      Epoch 9/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 20s 616ms/step - accuracy: 0.9134 - loss: 0.2353 - val_accuracy: 0.9000 - val_loss: 0.2957
      Epoch 10/10
      33/33 ━━━━━━━━━━━━━━━━━━━━ 20s 608ms/step - accuracy: 0.9255 - loss: 0.2037 - val_accuracy: 0.8923 - val_loss: 0.3060
```

The model's performance was monitored using the validation set to prevent overfitting, with early stopping based on validation loss.

6.1 Test Plan/ Test Cases



Layer (type)	Output Shape	Param #
conv2d_33 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_33 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_34 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_34 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_35 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_35 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_13 (Flatten)	(None, 86528)	0
dense_26 (Dense)	(None, 128)	11,075,712
dense_27 (Dense)	(None, 2)	258

```
Total params: 11,169,218 (42.61 MB)
Trainable params: 11,169,218 (42.61 MB)
Non-trainable params: 0 (0.00 B)
```

4- Model Training

The CNN model was trained on the training set using the following configurations:

- **Optimizer:** We used a method called Adam to make the model learn efficiently. It helped the model find the best settings for its 'brain' by adjusting how much it learns from each example. we used it as for default and kept most of the training parameters as it as they were in other models.
- **Learning Rate:** We set how much the model can learn from each example to a value of 0.001. This was like finding a balance between learning quickly and not making mistakes.
- **Loss Function:** We used a method called Sparse Categorical Cross-Entropy to see how much the model's guesses differed from the actual answers. This helped the model get better at picking the right answers.
- **Epochs:** The model looked passed the dataset 10 times to learn from it.

7 My learnings

Learnings from the Project

1. Machine Learning and Computer Vision:

   - I gained a solid understanding of machine learning, particularly in the field of computer vision.

   - I learned how to select and implement convolutional neural network (CNN) architectures for image classification tasks.

2. Data Collection and Annotation:

   - I acquired skills in gathering and annotating large datasets, essential for training machine learning models.

   - I understood the importance of data preprocessing and augmentation to improve model performance and generalizability.

3. Model Training and Evaluation:

   - I learned to split datasets into training, validation, and test sets, and the significance of each in model development.

   - I became proficient in training models and tuning hyperparameters to achieve optimal performance.

   - I learned to evaluate model performance using metrics such as accuracy, precision, recall, and F1 score, and how to interpret confusion matrices.