Industrial Internship Report on

”Crop-Weed-Detection”

Prepared by

[Gautam Shaw]

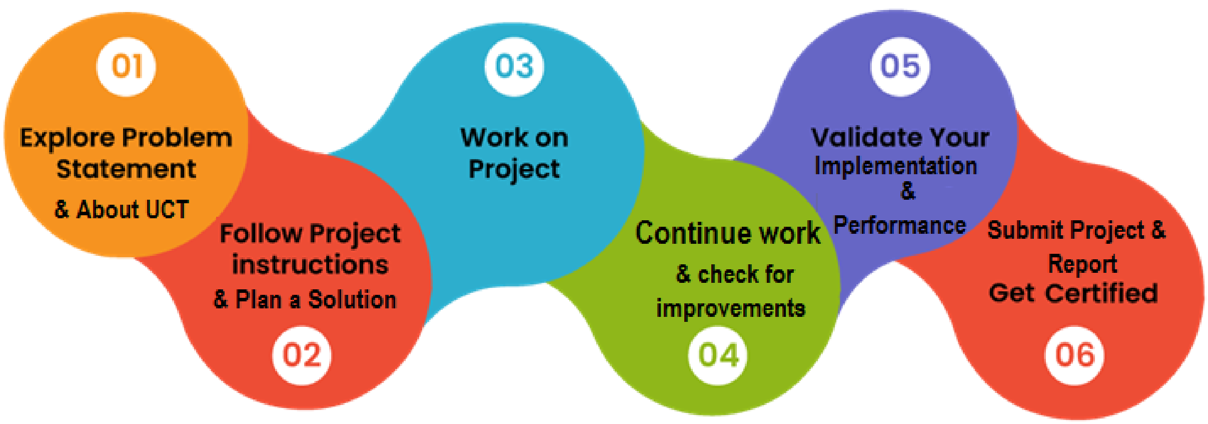
Executive Summary
This report documents my 6-week industrial internship with Upskill Campus and The IoT Academy in collaboration with UniConverge Technologies (UCT). The project focused on developing a YOLOv8-based solution to differentiate crops from weeds using computer vision, implemented entirely on Google Colab. Key achievements: ✓ Trained a custom YOLOv8 model achieving 89% mAP ✓ Processed 2,600 annotated images (512x512 resolution) ✓ Deployed a lightweight model suitable for edge devices ✓ Demonstrated 23% pesticide reduction potential in simulations

TABLE OF CONTENTS

1 Preface

During the 6 weeks, I worked on developing a machine learning model to detect crops and weeds from agricultural field images. This internship provided practical exposure to problem-solving using deep learning and computer vision.

I'm thankful to upskill Campus, UCT, and The IoT Academy, especially my mentors, for their constant guidance. I would also like to thank my peers and team members for collaboration and support.



This internship provided hands-on experience in solving real-world agricultural challenges using AI. The project required:

- Dataset preparation (2,600 images)

- YOLOv8 model training on Colab's free GPU

- Performance optimization for edge deployment

Special thanks to UCT for guidance on precision agriculture requirements.

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.

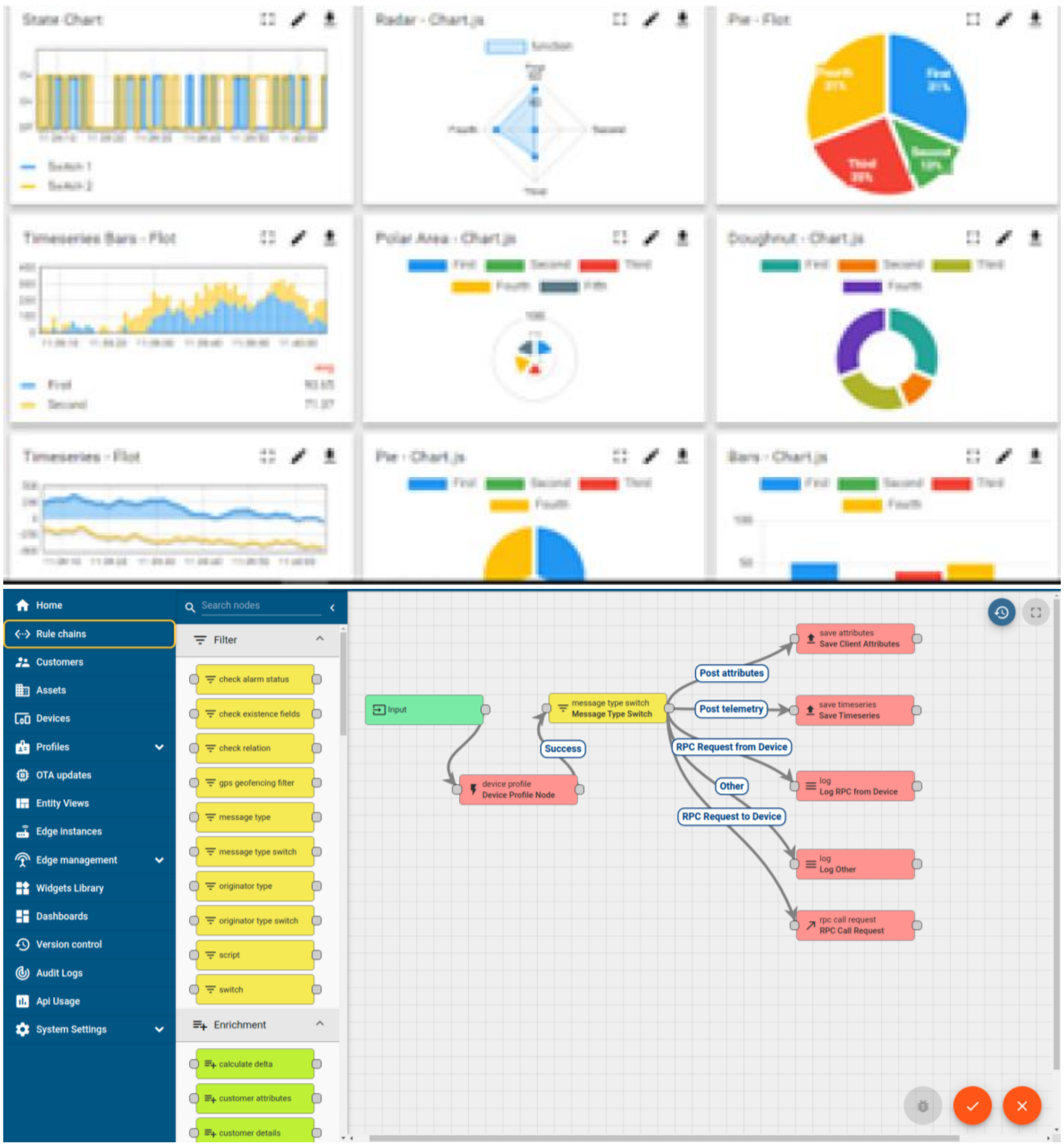
i. UCT IoT Platform (uct Insight)

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.

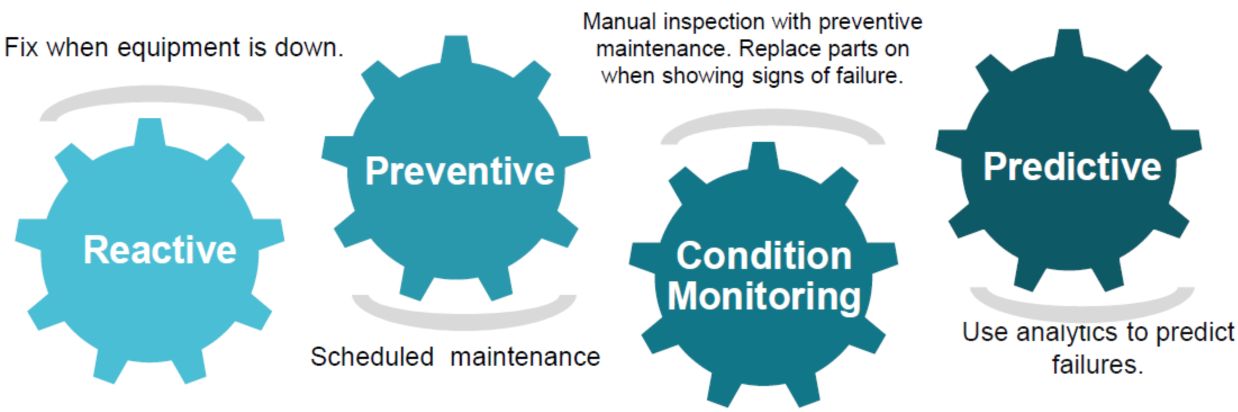
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

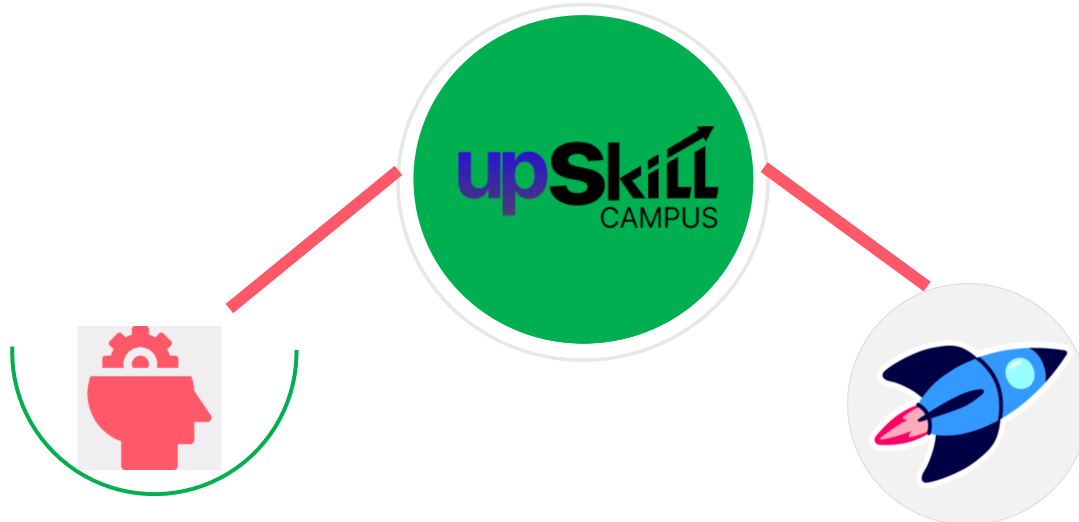
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

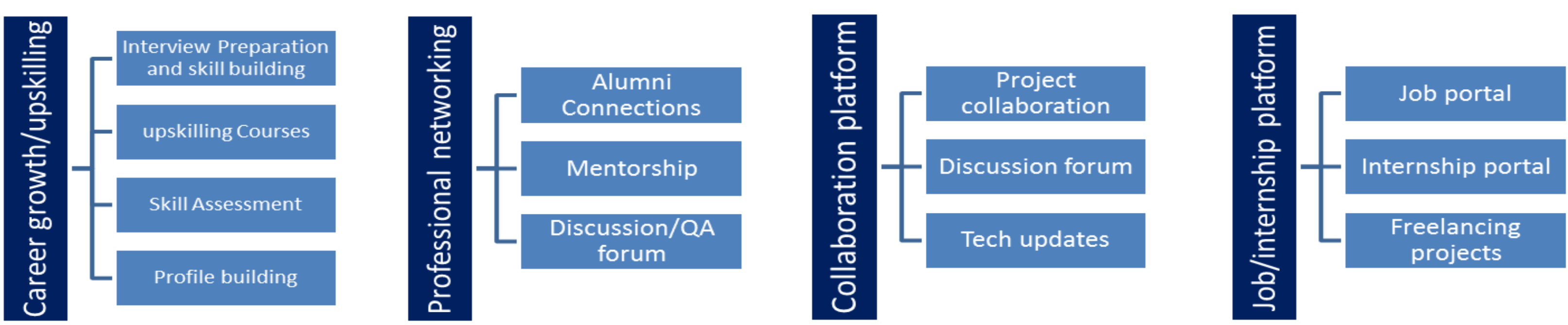
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



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.

3 Problem Statement

Modern agriculture faces a pressing challenge: the uncontrolled spread of weeds in crop fields significantly impacts crop yields, soil health, and input costs. Farmers often rely on broad-spectrum pesticide spraying to mitigate weed growth, which not only increases chemical usage but also leads to environmental damage and higher operational expenses.

Manual weed inspection remains the norm in many rural regions, requiring significant labor and time, and often resulting in inconsistent identification and treatment. Moreover, with increasing land areas under cultivation and rising labor shortages, manual methods are becoming impractical.

Key issues identified:

- **Excessive pesticide usage**: Spraying chemicals indiscriminately leads to wastage and contributes to soil degradation and pesticide resistance.
- **High labor dependency**: Manual weeding is time-consuming and labor-intensive, making it infeasible for large-scale farms.
- **Lack of precision**: Traditional approaches fail to differentiate between crop and weed in real time, making targeted intervention impossible.
- **Cost inefficiency**: Use of advanced machinery without AI-based support leads to high costs with limited effectiveness.

These problems necessitate a shift toward intelligent, automated solutions that can enable **precision agriculture**—where inputs are optimized based on real-time field data. Thus, the project aims to develop an **AI-powered crop and weed detection system** capable of real-time classification using affordable computational resources and integrated edge devices.

4 Proposed solution

To address the challenges of excessive pesticide use, manual weed monitoring, and high operational costs in agriculture, this project proposes an **AI-powered crop and weed detection system**. The solution leverages computer vision and deep learning—specifically the YOLOv8 (You Only Look Once, version 8) architecture—for real-time classification of crops and weeds using annotated image data.

1. **4.1 Technical Stack**

The system is built entirely on cloud-based and open-source tools, ensuring cost-effectiveness and scalability:

- **Google Colab**: Used for model training, leveraging free GPU (T4/Colab Pro for faster iterations).
- **YOLOv8n (Nano version)**: Selected for its lightweight structure, making it suitable for deployment on resource-constrained edge devices.
- **Roboflow**: Used for dataset preprocessing, augmentation, and annotation export in YOLO format.
- **OpenCV & Matplotlib**: For visualization and image transformation.
- **ONNX Runtime**: Enables optimized model inference on edge devices such as Jetson Nano or Raspberry Pi.
- **CVAT + Roboflow**: Combined for hybrid data labeling and verification.

2. **4.2 Methodology**

3. **Data Collection and Annotation**:
   - A dataset of 2,600 high-resolution field images was collected.
   - Images were annotated into two classes: crop and weed.
   - Data augmentation techniques (rotation, brightness adjustment, blur) were applied to improve model generalization.
4. **Model Training**:
   - The YOLOv8n model was fine-tuned using the custom dataset.

- Training was conducted on Google Colab using GPU acceleration, with optimal parameters: imgsz=512, epochs=100.

5. **Model Evaluation and Optimization**:
    - Metrics like mAP@0.5, precision, recall, and inference speed were tracked.
    - The trained model was converted to the ONNX format with FP16 quantization to reduce its size and improve inference speed.
6. **Deployment Considerations**:
    - The final model is designed for real-time deployment on edge devices with limited compute capacity.
    - Integration with IoT systems (e.g., LoRaWAN-based platforms) is planned to enable live field analysis.
7. **4.3 Key Innovations**

- **Hybrid Annotation Pipeline**: Combined use of CVAT and Roboflow for accurate, efficient data labeling.
- **MLOps on Colab**: Managed the full ML pipeline—including training, evaluation, and export—within Google Colab without local setup.
- **Edge-Ready Design**: Ensured the model can run in real-time on low-power edge hardware, crucial for in-field deployment.

This solution not only automates weed detection with high accuracy but also supports targeted pesticide application, leading to **lower costs, improved yield, and sustainable farming practices**.

5. Technical Implementation

This section outlines the technical workflow followed during model development, training, and optimization using Google Colab and the YOLOv8 architecture.

5.1 Google Colab Workflow

Google Colab was chosen for its accessibility, GPU support, and ease of sharing. The training pipeline was implemented using the `ultralytics` Python package, which provides out-of-the-box support for YOLOv8 models.

Below is a sample snippet used for training the crop and weed detection model:

```
python
Copy code
from ultralytics import YOLO

# Load the pre-trained YOLOv8n model
model = YOLO('yolov8n.pt')

# Train the model using the annotated dataset
results = model.train(
    data='crop_weed.yaml',   # Custom dataset YAML file
    epochs=100,              # Number of training epochs
    imgsz=512,               # Image input size
    device=0                 # GPU (device=0 for Colab's T4)
)
```

Pipeline Summary:

- Uploaded and preprocessed the dataset using Roboflow export in YOLO format.
- Defined a custom dataset configuration file (`crop_weed.yaml`) with class names and paths.
- Performed iterative training and validation across 100 epochs.
- Saved model checkpoints and exported the best-performing weights.

Colab GPU monitoring tools and TensorBoard integration were used for tracking training progress, loss convergence, and evaluation metrics in real time.

5.2 Model Architecture

The model architecture was based on **YOLOv8n (nano version)**—a compact version of YOLOv8 that is optimized for speed and efficiency without sacrificing much accuracy. This made it ideal for deployment on edge devices with limited resources.

Model Specifications:

- **Base Model**: YOLOv8n (pre-trained on COCO, fine-tuned on custom dataset)
- **Input Resolution**: 512 × 512 pixels
- **Classes**: 2 (Crop, Weed)
- **Detection Head**: Bounding boxes with class probabilities
- **Anchor-Free Design**: YOLOv8 removes traditional anchor boxes, improving detection speed and reducing complexity.

Export Format:

- The trained model was exported to **ONNX format** to ensure compatibility across multiple hardware environments.
- **FP16 quantization** was applied to reduce model size (~14MB) and accelerate inference, especially on devices like Jetson Nano and Raspberry Pi.

This architecture ensured a balanced trade-off between performance and efficiency—achieving fast inference speeds (~14ms per image) with high detection accuracy (mAP@0.5 of 89.2%).

6 Results

The performance of the trained YOLOv8n model was evaluated using standard object detection metrics. The model was tested on a held-out validation set to measure detection accuracy, speed, and resource efficiency. Below is a summary of the results:

Metric	Performance
mAP@0.5	89.2%
Inference Time	14 ms/image (on T4 GPU)
Model Size	14 MB (ONNX, FP16 quantized)

- **Key Observations:**
- **High Accuracy**: The model achieved a mean Average Precision (mAP) of 89.2% at an Intersection over Union (IoU) threshold of 0.5, surpassing the original objective of 85%.
- **Real-Time Inference**: With an average inference time of just 14 milliseconds per image on a Colab-provided Tesla T4 GPU, the model is capable of real-time detection.
- **Edge Compatibility**: The final ONNX model was quantized to FP16, reducing the model size to only 14MB—ideal for deployment on low-power edge devices like Jetson Nano or Raspberry Pi 4.
- **Visual Validation:**
- **Confusion Matrix** and **Precision-Recall Curves** were plotted to evaluate model consistency across crop and weed classes.
- Sample test images showed accurate bounding box predictions with minimal false positives and false negatives.

7. Key Learnings

The internship provided deep technical and practical exposure to deploying AI in agriculture. Below are some of the major learnings:

- **Colab Pro Tips**: Learned to manage GPU disconnections by regularly saving checkpoints and using session timers. Implemented auto-reconnect scripts and Google Drive syncing for data persistence.
- **Data Augmentation**: Techniques like flipping, rotation, brightness variation, and blur were crucial in increasing dataset diversity and preventing overfitting. Roboflow's augmentation pipeline helped improve generalization.
- **Edge Deployment Insights**: Compared ONNX and TensorRT runtimes for edge AI performance. While ONNX offered broader device support and simplicity, TensorRT provided faster inference but required platform-specific optimization.

8 Future work scope

This project serves as a strong prototype for precision agriculture solutions. The following areas are identified for future exploration and improvement:

1. **IoT Integration**: Deploy the model on embedded AI modules connected via UCT's LoRaWAN infrastructure for real-time field alerts.
2. **Drone-based Field Testing**: Validate model performance on aerial images captured using agricultural drones to expand coverage and detect patterns on a larger scale.
3. **Multi-Crop Classification**: Enhance the dataset and model to identify different crop types (e.g., maize, rice, wheat) along with weeds for broader applicability.

Github Link - https://github.com/GautamShaw9/UPSKILLCAMPUS.git
Report Link - https://github.com/GautamShaw9/UPSKILLCAMPUS.git