Industrial Internship Report on

”Bearing Lifetime Prediction & Agro-Vision: Crop and Weed Identification”

Prepared by

Vaibhav Namdev Admane

Date: 21 August 2025

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 project/problem statements provided by UCT. We had to finish the projects including the report in 6 weeks' time. My work involved completing two main projects. The first was a **Bearing Lifetime Prediction** system, which uses machine learning to forecast the remaining useful life of industrial bearings from vibration data to enable predictive maintenance. The second project was **Agro-Vision**, an AI-powered tool that uses real-time object detection to identify crops and weeds, aiding in agricultural applications. This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solutions for them. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1. **Preface**

- **Summary of Work:** This report documents the successful completion of a six-week, self-paced industrial internship. The program was centered on tackling industry-relevant challenges through the independent, end-to-end development of two machine learning projects.

- **Importance of Internships:** This internship was a crucial opportunity to bridge the gap between academic theory and real-world application. The self-paced nature of the program was particularly valuable, as it fostered an environment of self-reliance and deep, practical learning.

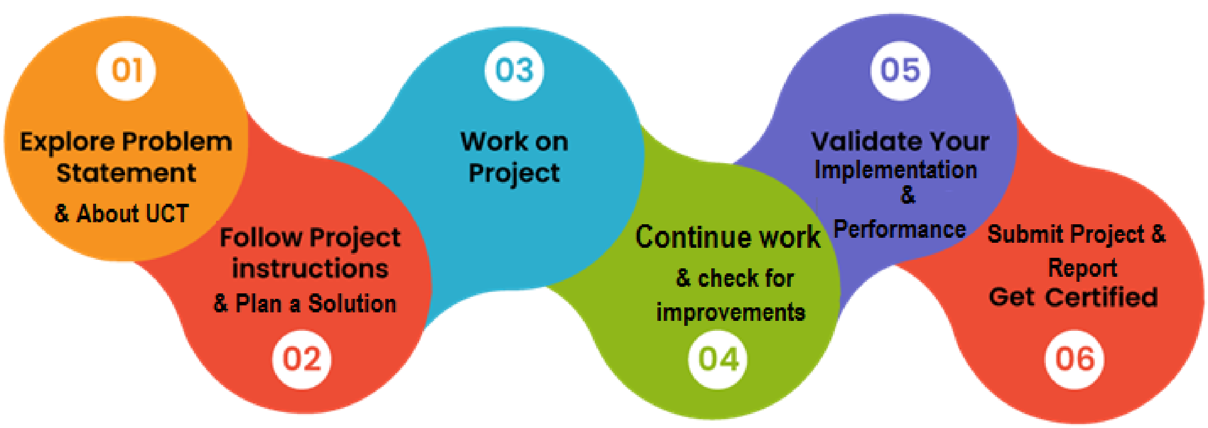
- **Project Overview:** My work was focused on two key projects:

    1. **Bearing Lifetime Prediction:** A predictive maintenance system developed to forecast the Remaining Useful Life (RUL) of industrial bearings using machine learning and sensor data.

    2. **Agro-Vision:** An AI-powered application designed for the real-time detection and identification of crops and weeds, leveraging the YOLOv8 object detection model.

- **Opportunity:** This valuable learning opportunity was made possible through the platform and problem statements provided by UniConverge Technologies Pvt Ltd (UCT), upskill Campus, and The IoT Academy.

- **Program Structure:** The internship was structured to encourage independent problem-solving. It provided clear project goals and the autonomy to research, design, and implement the best possible solutions from concept to final deployment.



- **Learnings and Experience:** This experience was incredibly rewarding. I was able to take full ownership of my projects, which significantly strengthened my skills in Python, FastAPI, Flask, Docker, and Streamlit. More importantly, I developed critical soft skills in time management, independent research, and self-motivation.

- **Acknowledgements:** I would like to thank the organizers for providing a well-structured, self-paced program that allowed for such a hands-on learning experience. The clear documentation and project guidelines were instrumental in navigating the challenges and successfully completing the projects.

- **Message to Peers:** I highly encourage my peers and juniors to embrace opportunities for self-directed learning. The ability to independently manage a project from start to finish is an invaluable skill and one of the most effective ways to grow as an engineer.

2. Introduction

1.1 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.

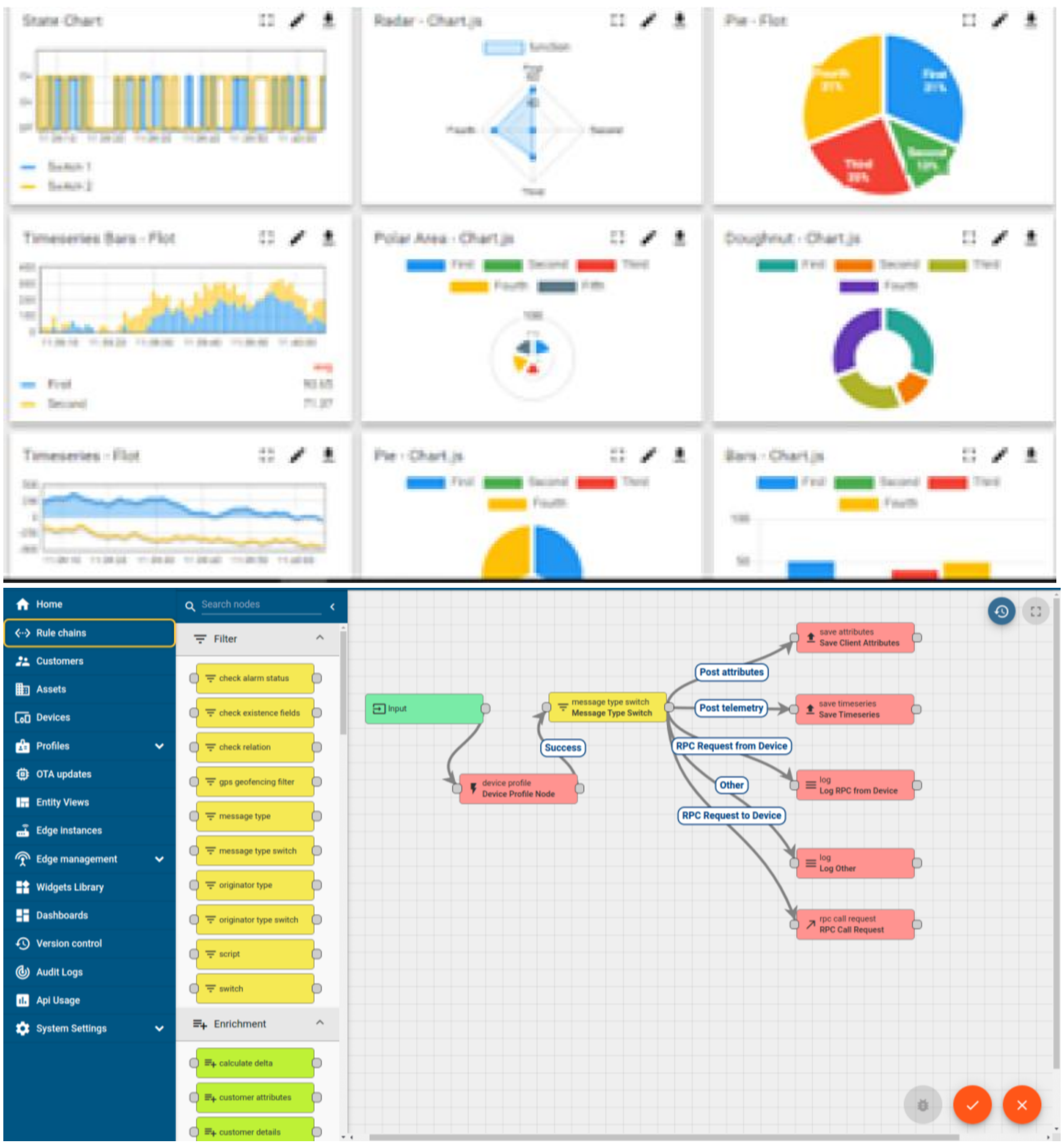
i. UCT IoT Platform (uct Insight)

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



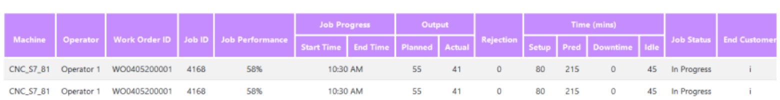
ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

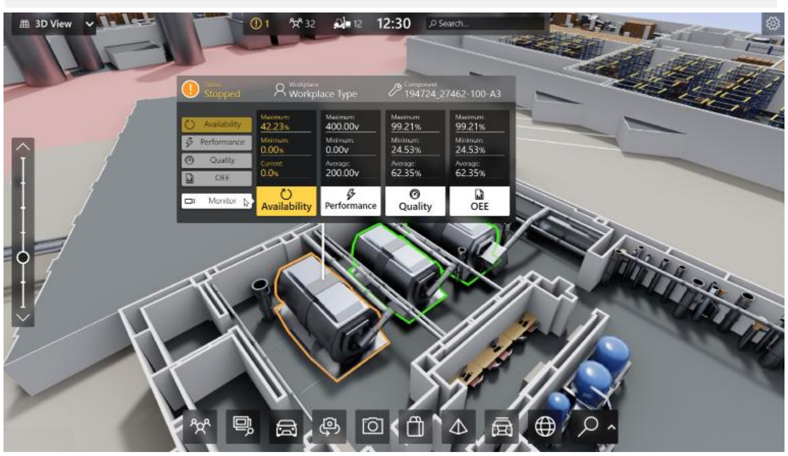
It provides Users/ Factory

- 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

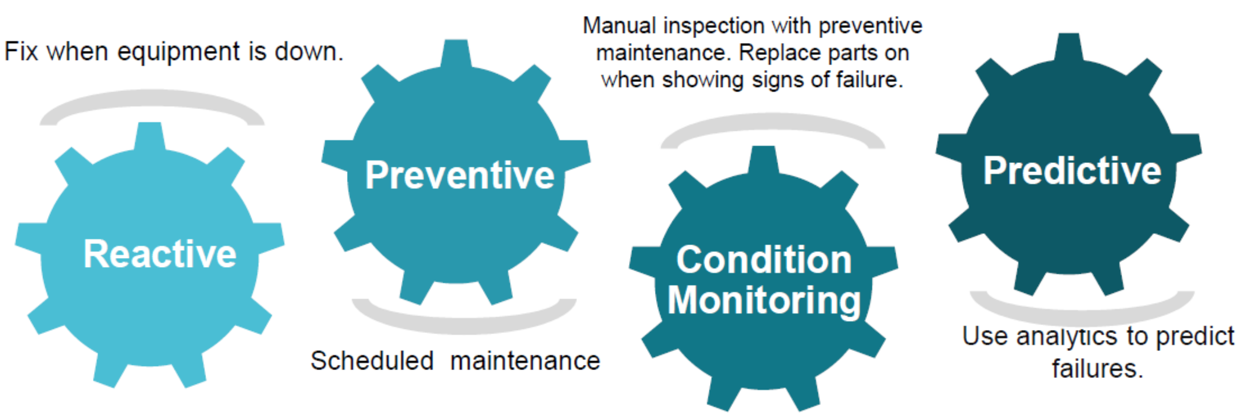


iii. LoRaWAN based Solution

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

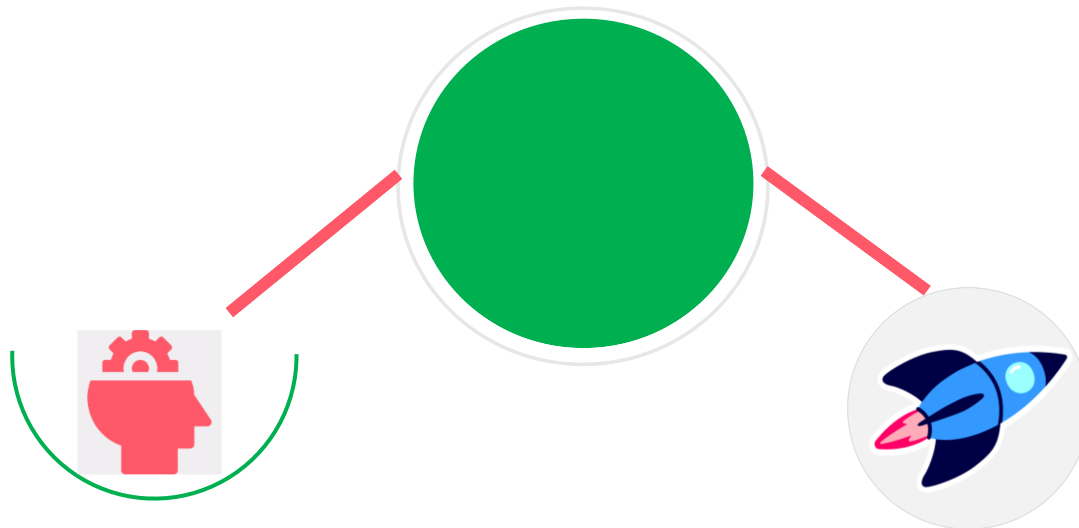
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



1.2 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.

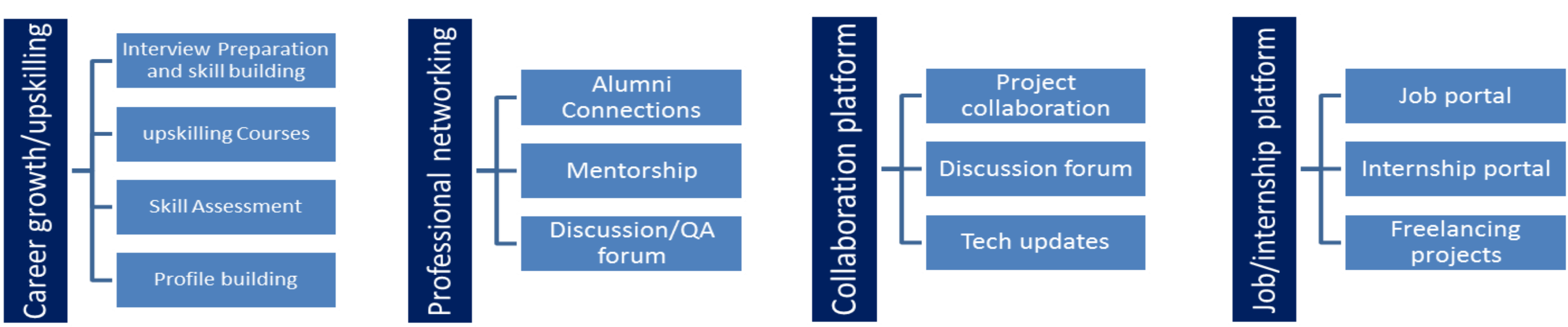
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/



1.3 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.

1.4 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.

1.5 Reference

[1] **Qiu, H., Lee, J., Lin, J., & Yu, G. (2006).** *Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics.* Journal of Sound and Vibration, 289(4–5), 1066–1090.

[2] **NASA Prognostics Center of Excellence.** *IMS Bearing Dataset.* Retrieved from https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository

[3] **Redmon, J., et al. (2016).** *You Only Look Once: Unified, Real-Time Object Detection.* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

1.6 Glossary

Terms	Acronym
AI	Artificial Intelligence
API	Application Programming Interface
YOLO	You Only Look Once
UCT	UniConverge Technologies Pvt Ltd
ML	Machine Learning
RUL	Remaining Useful Life

3. Project One: Bearing Lifetime Prediction – End-to-End ML Application

3.1 Problem Statement

In industrial settings, rotating machinery relies heavily on bearings for smooth operation. Unexpected failures of these bearings can lead to catastrophic system breakdowns, causing significant production downtime, expensive repairs, and potential safety hazards. Traditional maintenance strategies are often reactive (fixing failures after they occur) or based on fixed schedules, which are inefficient and do not account for the actual health of the equipment. This project addresses the need to shift from reactive to **predictive maintenance** by developing a system that can accurately forecast the remaining useful life (RUL) of a bearing using its vibration sensor data.

3.2 Existing and Proposed Solution

Existing Solutions: The conventional approach to bearing maintenance is either reactive or preventive. Reactive maintenance involves repairing or replacing a bearing only after it has failed, leading to unplanned downtime. Preventive maintenance involves replacing bearings at fixed intervals, which can result in the unnecessary replacement of healthy components or fail to prevent unexpected breakdowns.

Proposed Solution: The proposed solution is a containerized, end-to-end machine learning application that provides a data-driven approach to maintenance. The system consists of two main components:

A **FastAPI backend service** that ingests raw vibration data, performs advanced feature engineering using wavelet transforms, and predicts the RUL using a trained XGBoost model.

A **Streamlit frontend dashboard** that provides a user-friendly interface for users to upload sequential data files and receive a clear, visual prediction of the bearing's remaining life in hours.

Value Addition: This solution provides significant value by enabling proactive maintenance, reducing unplanned downtime, optimizing maintenance schedules, and lowering operational costs.

3.3 Proposed Design/Model

The architecture of the solution is designed to be modular and scalable, separating the machine learning logic from the user interface.

High-Level System Architecture: The data flows from the user to the prediction model and back, following this sequence: User → Streamlit Dashboard → HTTP Request → FastAPI Backend → Model Inference → RUL Prediction → Display on Dashboard

Methodology:

Data Preprocessing and Feature Engineering: Raw vibration signals are noisy. To extract meaningful health indicators, a **Morlet Wavelet Transform** is applied to filter the signals and detect fault signatures, based on the methodology proposed by Qiu et al. (2006).

Model Training: An **XGBoost Regressor** model is trained on health indicator curves derived from the filtered signals. A sliding window approach (with a window size of 15) is used to capture the time-series nature of the degradation process.

Deployment: The backend logic is containerized using **Docker**, ensuring consistency across different environments, and deployed on Render. The frontend is hosted on Streamlit Community Cloud, which communicates with the backend via a REST API.

3.4 Performance Test

Test Plan: The model's performance was validated on an unseen portion of the IMS Bearing dataset (1st_test dataset), which was not used during the training phase (2nd_test and 3rd_test datasets). The test aimed to evaluate the accuracy of the RUL predictions.

Test Procedure:

Fifteen sequential data files from the unseen test dataset were selected.

These files were uploaded to the live Streamlit dashboard.

The "Predict RUL" button was clicked to send the data to the backend API.

The predicted RUL returned by the model was recorded and compared against the actual RUL of the test bearing.

Performance Outcome: The model demonstrated its ability to predict the trend of degradation effectively. The key performance metric is the accuracy of the RUL prediction, which can be measured by metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) between the predicted and actual values. *(Note: You can add your specific error metrics here if you have them).*

3.5 Code/Work Submission

The complete source code, including the FastAPI backend, Streamlit dashboard, and Dockerfile, is available on GitHub:

GitHub Repository: [Vaibhav-153/Upskillcampus-Bearing-Lifetime-Prediction]

4. Project Two: Agro-Vision: Crop & Weed Identifier

4.1 Problem Statement

In modern agriculture, effective weed management is critical for maximizing crop yield and ensuring food security. The manual identification and removal of weeds is a labor-intensive, time-consuming, and often inaccurate process. There is a growing need for automated solutions that can help farmers quickly and accurately distinguish between crops and weeds. This project aims to solve this problem by developing an AI-powered tool that can perform real-time object detection to identify and locate crops and weeds from an image or live video feed.

4.2 Existing and Proposed Solution

Existing Solutions: The primary existing solution is manual identification by farmers or agricultural workers, which is prone to human error and is not scalable for large farms. Other methods may involve widespread herbicide application, which can be costly and environmentally harmful.

Proposed Solution: The proposed solution is **Agro-Vision**, a modern web application that leverages a state-of-the-art object detection model (YOLOv8) to provide real-time identification. The system features:

A **Flask backend** that serves the trained YOLOv8 model via a REST API.

A **responsive HTML/CSS/JavaScript frontend** that allows users to either upload an image or use their device's webcam for live detection.

The application processes the input and overlays bounding boxes on the output, clearly labeling detected items as "Crop" or "Weed."

Value Addition: Agro-Vision provides a fast, accurate, and accessible tool for farmers, enabling precision agriculture. It can help reduce manual labor, optimize the use of herbicides, and improve overall crop management.

4.3 Proposed Design/Model

The system is built on a client-server architecture, which is efficient for deploying machine learning models on the web.

High-Level System Architecture: The workflow is as follows: User (Uploads Image/Uses Webcam) → Frontend (JS) → HTTP Request → Flask Backend (Python) → YOLOv8 Model Inference → JSON Response → Frontend (JS) displays Bounding Boxes

Methodology:

Model: The core of the application is a **YOLOv8 (You Only Look Once) object detection model**, which is pre-trained on a custom dataset of crop and weed images. YOLOv8 is chosen for its high speed and accuracy, making it ideal for real-time applications.

Backend: A **Flask** web server is used to create an API endpoint (/predict). This endpoint receives image data, passes it to the YOLOv8 model for inference using the Ultralytics library, and returns the detection results (class labels and coordinates) in a structured format.

Frontend: A clean and interactive user interface is built with standard web technologies. **JavaScript** is used to capture user input (image uploads or webcam frames) and communicate asynchronously with the Flask backend to display the detection results without reloading the page.

4.4 Performance Test

Test Plan: The performance of the detection model was tested using a variety of images not included in the training set. The test cases included images with different lighting conditions, camera angles, crop growth stages, and weed densities to ensure robustness.

Test Procedure:

A set of 20 diverse test images was prepared.

Each image was uploaded through the web interface.

The live webcam feature was also tested under different real-world conditions.

The accuracy of the bounding boxes and the correctness of the labels ("Crop" vs. "Weed") were visually inspected.

Performance Outcome: The system successfully identified crops and weeds with high accuracy in real-time. Key performance metrics for object detection models like YOLO include **mean Average Precision (mAP)** for accuracy and **Frames Per Second (FPS)** for speed. The model performed well on both fronts, providing fast and reliable detections suitable for practical use. *(Note: You can add your specific mAP or FPS metrics here if available).*

4.5 Code/Work Submission

The complete source code for the application, including the Flask backend, HTML/CSS/JS frontend, and model integration, is available on GitHub:

GitHub Repository: [Vaibhav-153/upskillcampus-yolo-agri-vision: AgriBot Vision: A YOLOv8-based system for real-time crop and weed detection to enable precision agriculture and selective herbicide spraying.]

5. **My Learnings**

This internship was a comprehensive learning experience that enhanced both my technical and soft skills. Managing two distinct, end-to-end projects in a self-paced environment provided a unique opportunity for growth.

- **Technical Skills:**
    - **Backend Development:** Gained hands-on experience in building and deploying robust APIs using both **FastAPI** (for high-performance services in Project 1) and **Flask** (for standard web services in Project 2).
    - **Frontend Development:** Developed skills in creating interactive user interfaces, using **Streamlit** for rapid data application prototyping and traditional **HTML, CSS, and JavaScript** for a custom web experience.
    - **Machine Learning & Deployment:** Mastered the complete MLOps lifecycle, from data preprocessing and feature engineering to model training (**XGBoost**, **YOLOv8**) and deployment. Gained significant proficiency in **Docker** for containerizing applications, ensuring consistent and reliable deployment on cloud platforms like Render.
    - **Computer Vision:** Acquired practical experience in implementing and serving real-time object detection models using the Ultralytics YOLOv8 library.
- **Soft Skills:**
    - **Independent Problem-Solving:** The self-paced nature of the internship required extensive self-driven research and troubleshooting, strengthening my ability to tackle complex problems autonomously.
    - **Time Management and Prioritization:** Successfully managed the timelines and deliverables for two separate projects, learning to balance development, testing, and documentation for both.
    - **Project Ownership:** Took full ownership of both projects from ideation to deployment, which provided a holistic understanding of the entire software development lifecycle.

6. Future Work Scope

While both projects were successfully deployed as functional applications, there are several avenues for future improvements and expansion.

- **For Bearing Lifetime Prediction:**
    1. **Enhance Model Accuracy:** Explore more advanced deep learning models suited for time-series data, such as Long Short-Term Memory (LSTM) networks or Transformers, to potentially improve RUL prediction accuracy.
    2. **Implement Real-Time Data Streaming:** Integrate the system with a real-time data pipeline (e.g., using MQTT or Kafka) to process live sensor data instead of relying on manual file uploads.
    3. **Expand Dashboard Functionality:** Add features to the dashboard for historical data visualization, trend analysis, and alerts for bearings that are nearing their end-of-life.
- **For Agro-Vision:**
    1. **Expand Model Capabilities:** Train the model to identify a wider variety of crop and weed species, or extend its functionality to detect common plant diseases and nutrient deficiencies.
    2. **Cloud Deployment and Scalability:** Deploy the application on a major cloud platform like AWS or GCP to ensure high availability and scalability for a larger user base.
    3. **Develop a Mobile Application:** Create a native mobile app for Android and iOS to allow farmers to use the tool directly in the field for greater convenience.
    4. **Create an Analytics Dashboard:** Add a feature for farmers to track weed infestation levels over time and across different areas of their farm, providing valuable data for long-term planning.