BSc (Hons) in Information Technology Specializing Data Science

Research Project - IT4010 **MLOPs Report**

Group ID: RP-24-25J-307

Project Title: Dynamic Crop Modeling for Salad Cucumbers Using Biomass and Irrigation Weights and Visuals

1. IT21267222: Real-Time Yield Prediction using Sensor Data
2. IT20418274: On demand irrigation system based on BIO-MASS weight and Wilted Leafe detection using Image Processing
3. IT21327094: Cucumber Crop Growth Monitoring: Integrating Image Processing and Machine Learning.
4. IT21225956: Cucumber Fruit Analysis: Integrating Image Processing Machine Leaning Techniques

		IT21267222	IT20418274	IT21327094	IT21225956
Data Pipeline	**Data Sources:**	Real-time sensor data from LOADCELL weight sensors, temperature, humidity, sunlight, and soil moisture sensors. Data collected from cucumber plants in controlled environments at regular intervals. All sensors interfaced with ESP32 modules and logged to Oracle	LOADCELL weight sensors were used to monitor real-time biomass weight fluctuations of cucumber plants. A top sensor was integrated to measure plant weight changes accurately. Environmental sensors recorded temperature, humidity, and light intensity to assess	Deployed five ESP32-CAM modules in front of five cucumber plants, capturing images every 30 minutes from 8 AM to 4 AM, starting from the flowering stage. Additional images of leaves were captured for disease detection and nitrogen deficiency analysis.	Captured using an ESP32-CAM module in a controlled environment with timestamp annotation tracking growth from bud to mature stages. Images stored in /home/opc/uploads on an oracle VM.

		VM and InfluxDB buckets.	external influences on plant hydration. Real-time data collection enabled precise irrigation decisions, reducing reliance on manual observations.	Data was stored in Oracle VM for model training.	
			Images were captured using ESP32-CAM modules deployed in front of cucumber plants, capturing images every 30 minutes from 8 AM to 4 AM.		
	Data Preprocessing:	Noise reduction: Moving average smoothing applied to sensor streams. Outlier removal: Statistical detection of anomalous spikes in	Noise reduction techniques, such as moving average smoothing, were applied to stabilize sensor readings affected by environmental factors.	Image augmentation to enhance the dataset. instance segmentation, pixel value normalization, data scaling, and leaf area segmentation.	Image augmentation (4x) to enhance dataset diversity. Instance segmentation for cucumber detection. Image resizing and normalization.

	sensor data (e.g., Z-score). Feature engineering: Derived features like weight change rate, evapotranspiration estimate, light-weight index. Data normalization: Min-Max Scaling applied across sensor modalities. Time series alignment: Ensured synchronized timestamps for fusion of weight, environmental, and yield labels.	Outlier detection was performed to remove extreme values and prevent errors from sensor misreadings. Feature engineering was conducted, including the calculation of weight change rate as a key parameter for detecting dehydration. Environmental data was normalized to maintain consistency across different sensor types. Data synchronization ensured uniform timestamps across all sensor sources for accurate analysis.	Integrated image data with tabular lux sensor data for nitrogen deficiency prediction.	Labeling for growth stages, quality assessment, and defect detection.
		Image augmentation to enhance dataset diversity, instance segmentation, pixel value normalization,		

		data scaling, and leaf area segmentation for better model accuracy.		
Data storage and versioning	MongoDB and InfluxDB for real-time sensor time-series data. Model inputs and predictions stored in versioned InfluxDB buckets. Scripts stored on Oracle VM. Model versioning handled with naming conventions and hash tagging for reproducibility.	InfluxDB was utilized for real-time data ingestion and storage, optimized for handling time-series sensor data. Irrigation-related data was stored within dedicated InfluxDB buckets for organized access and management. Captured and processed images were stored in an Oracle Virtual Machine (VM) for model training. Metadata and model predictions were stored in InfluxDB, ensuring version	Captured and processed images stored in Oracle VM. Metadata of new input data stored in a bucket in InfluxDB. Model predictions are stored in InfluxDB. Model versions tracked for reproducibility.	Processed images and metadata stored on a VM. Predictions stored in InfluxDB (CAM_module_prediction bucket). Model versions tracked for reproducibility.

			tracking and reproducibility.		
Model Development	**Model Selection**	Hybrid Model with novel ensemble architecture: Random Forest Regressor for non-linear biomass-environment interactions. Gradient Boosting to refine weight-to-yield mapping. LSTM (Long Short-Term Memory) model for temporal sequence analysis of sensor trends. Stacking Ensemble Regressor used to combine above models using meta-learner (Linear Regression).	Long Short-Term Memory (LSTM) was selected due to its ability to process sequential sensor data and capture long-term trends in plant hydration. Temporal Convolutional Networks (TCN) and Deep Neural Networks (DNN) were also evaluated for their predictive capabilities. A comparative study of different models, including ResNet50, VGG16, and a custom CNN, was performed. The VGG16 was selected for its superior accuracy in	**Crop Growth Classification:** Custom CNN selected as the best model from ResNet50, VGG16, and Custom CNN. **Leaf Disease Detection and Identification:** Two CNN models (one for detecting healthy/unhealthy leaves, another for disease classification). **Nitrogen Deficiency Detection:** ResNet50 used for feature extraction, followed by a regression model to predict nitrogen deficiency percentage using NDVI values.	**Maturity Identification:** • Trained the custom CNN, VGG16, RestNet50 and select the best model as CNN model. **Defect Detection & Quality Assessment:** • Trained the custom CNN, VGG16, RestNet50 and select the best model as CNN model. O identify the quality of the fruit. If the quality is low identify the defect type of the fruit.

	Rationale: Leverages both static relationships and dynamic trends.	classifying wilted and non-wilted leaves. .		
Model Training	Dataset split: 80% training, 20% testing. Time windowed training data for LSTM: sliding window of 10 timestamps. Targets: harvest quantity (number of fruits) and average fruit size. Loss: MSE, Metrics: RMSE, R^2, and MAPE. Frameworks: Scikit-learn for RF/GB,	80% of the collected data was allocated for training, while 20% was used for validation. The Mean Squared Error (MSE) loss function was employed to minimize prediction errors. Evaluation metrics included RMSE for accuracy, Precision & Recall for dehydration classification, and MAPE for real-world performance assessment.	Trained on a labeled dataset: **Growth Stages:** Flowering, Fruiting Stage 1, Fruiting Stage 2. **Leaf Disease Classification:** Healthy vs. Unhealthy leaves. **Disease Identification:** Anthracnose, Bacterial Wilt, Downy Mildew, Gummy Stem Blight. **Nitrogen Deficiency Estimation:** NDVI-based analysis for nutrient assessment.	Trained on a labeled dataset: • Growth Stages: Bud, Developing, Mature. • Quality Assessment: High vs. Low quality. • Defects: Belly Rot, Discoloration, Pythium Fruit Rot. Performance evaluation using accuracy, precision-recall, and F1-score

		TensorFlow/Keras for LSTM. Training augmented using sensor-derived simulated data for generalization.	The training dataset included labeled images of wilted and non-wilted leaves. Transfer learning was applied using pre-trained models, and hyperparameter tuning was conducted to enhance performance. The final model was trained on an optimized dataset to improve classification accuracy.	Performance evaluation using accuracy, precision-recall, F1-score, and NDVI correlation for nitrogen deficiency.	
Model Integration	**Tools used**	Python, TensorFlow/Keras, Scikit-learn, Pandas, NumPy InfluxDB for time-series storage.	TensorFlow/Keras was implemented for LSTM model development and training. InfluxDB and Telegraf enabled real-time data	**Data Processing:** OpenCV, numpy TensorFlow/Keras, PyTorch.	**Data Processing:** OpenCV, TensorFlow/Keras, PyTorch. **Database:** InfluxDB (for storing predictions).

		Grafana for real-time visualization. FastAPI for serving predictions via REST endpoints.	streaming from IoT sensors. Grafana was used for visualizing real-time predictions and irrigation recommendations.	**Database:** InfluxDB (for storing metadata and predictions). **Visualization:** Grafana (for real-time dashboard). **Backend & APIs:** Flask/FastAPI for serving predictions.	**Visualization:** Grafana (for dashboard). **Backend & APIs:** Flask/FastAPI for serving predictions.
			Image processing and model training were conducted using OpenCV, NumPy, TensorFlow/Keras, and PyTorch. Database management was handled using InfluxDB, while Grafana was used for real-time visualization. FastAPI was utilized to serve model predictions.		

Model Deployment	**Testing Environments**	Real-time testing with live sensor feeds from ESP32 modules.	Local server testing was conducted using stored test datasets to validate initial model performance. Cloud deployment sandbox in Google Cloud was utilized to assess scalability and operational efficiency. Initial model testing was performed in Jupyter Notebook using real-time images captured through ESP32-CAM modules. Validation was carried out by comparing predictions with expert-labeled data. .	**Local testing** in Jupyter Notebook before deployment. **Validation with real-time cucumber images** captured through ESP32-CAM and sensor data	• Local machine testing before cloud deployment. • Validation with real-time cucumber images.

Deployment Platform	Oracle Virtual Machine (VM) for real-time data ingestion, model inference. Integration with Grafana and FastAPI for visualization and API-based communication.	Google Cloud Platform (GCP) was integrated with InfluxDB for real-time monitoring and prediction execution. Grafana Dashboard for real-time visualization and user insights. The model was deployed on Oracle Virtual Machine (VM), leveraging cloud resources for efficient processing and storage to handle image processing, predictions, and model inference.	Oracle Virtual Machine (VM) for executing models. Grafana Dashboard for real-time visualization and user insights.	Oracle Virtual Machine (VM) for model execution. Grafana Dashboard for real-time monitoring.
Deployment Method	Model served via FastAPI endpoint.	Model predictions were exposed as a REST API for		Model served using **Flask/FastAPI** on the VM.

		RESTful API receives live sensor data and returns predicted yield Predictions and inference logs pushed to InfluxDB for visualization	seamless communication with IoT devices. An automated pipeline was implemented to trigger irrigation actions based on real-time model predictions.	Models served using Flask/FastAPI on Oracle VM. Predictions and metadata pushed to InfluxDB for visualization. Alerts are triggered for detected diseases or abnormal nitrogen levels.	Predictions pushed to **InfluxDB** for real-time visualization. Alerts and notifications are triggered based on predictions.
			Models were served using Flask/FastAPI on Oracle VM, with predictions and metadata pushed to InfluxDB for visualization. To support timely intervention, alerts were triggered for detected wilted leaves.		
Future Enhancements	**Model Improvement:**	Meta-Reinforcement Learning: Use reinforcement reward system to adapt	**Adaptive Learning:** Implementing reinforcement learning for self-optimizing	**Enhanced Growth Stage Forecasting:** Improve CNN-based models by incorporating sequential	**Better Instance Segmentation:** Enhance object detection for precise growth tracking.

		prediction strategy over time based on harvest accuracy. AutoML & NAS: Integrate Neural Architecture Search for evolving LSTM + dense hybrid architecture automatically. Multi-variety Generalization: Extend to work across cucumber variants, integrating crop-specific parameters. Edge Optimization: Deploy quantized models on ESP32 + AI co-processors for ultra-low latency predictions without cloud dependency. Real-time Feedback Loop: Use discrepancies between predicted and	irrigation recommendations. **Multi-Crop Support:** Extending the model for diverse crops beyond cucumbers. **Integration with Weather Forecasting:** Enhancing predictions by incorporating rainfall probability data. Enhancements include the development of a smart leaf health index by integrating NDVI-based analysis with deep learning models. Adaptive image processing techniques will be implemented to handle varying lighting conditions.	learning techniques for more accurate growth stage projections. **Smart Leaf Health Index:** Develop an advanced leaf health scoring system by integrating NDVI with deep feature extraction from disease classification models. **Adaptive Image Processing:** Implement real-time adaptive filtering to handle varying lighting conditions in greenhouse environments, reducing image inconsistencies. **Real-Time Model Deployment on Edge Devices:** Optimize models for execution on low-power IoT devices (ESP32-based AI accelerators) for on-site processing without server dependency.	**Multi-Modal Data Fusion:** Integrate sensor data (humidity, temperature) for more accurate harvest predictions. **Automated Defect Detection:** Improve clustering and RGB masking for real-time defect alerts. **Edge Deployment:** Optimize models for IoT edge devices to enable on-site processing. **Expand Dataset:** Collect more diverse images under different environmental conditions. **Harvesting Robot:** First the robot identifies the mature fruit and picking the cucumber using the robot arm.

	actual harvest for active learning and retraining.	Future iterations will optimize models for execution on edge devices, reducing dependency on centralized servers. Additionally, expanding the dataset to cover multiple plant varieties and environmental conditions will improve model robustness and generalization.	**Expanded Multi-Condition Dataset:** Collect and incorporate additional datasets covering different cucumber varieties, environmental conditions, and disease severity levels to enhance model generalization.