Nutrient Deficiency Detection and Management for Improved Crop Quality and Quantity

Anupam Gupta
PhD, Dept. of Computer Science Engineering
IIT Kharagpur
Kharagpur, India
anupamguptacal@gmail.com

Raj Shekhar Vaghela
M.Tech, Dept. of Computer Science Engineering
IIT Kharagpur
Kharagpur, India
vaghelaraj9420@gmail.com

***Abstract*—This project introduces a novel approach to addressing nutrient deficiency in banana crops using deep learning and IoT technologies. By implementing the YOLOv5 object detection model in an Android application, crop health can be analyzed in real-time through smartphone cameras. The optimized YOLOv5 model enables accurate detection of leaf deficiencies, even in mobile settings, benefiting traditional Indian farming practices. Additionally, the model has been optimized for IoT devices, such as drones, through quantization techniques, ensuring efficient inference despite hardware constraints. The report details the design, implementation, and evaluation of the solution, highlighting methodologies for model optimization and quantization. Empirical testing confirms the system's performance, underscoring its potential to revolutionize crop management and contribute to sustainable farming practices.**

I. Introduction

Modern agriculture faces the challenge of optimizing crop yields while minimizing resource usage. Precision agriculture, empowered by technological advancements, offers a solution by providing farmers with real-time data for informed decision-making. This project focuses on leveraging deep learning and IoT technologies to address a crucial aspect of precision agriculture: nutrient deficiency detection and remediation in banana crops. By integrating object detection, IoT communication, and automated nutrient delivery systems, our aim is to revolutionize crop management practices, enhance yields, and promote sustainable agriculture.

A. Motivation:

The motivation behind this project arises from the imperative to enhance agricultural practices to meet the needs of a growing global population while ensuring environmental sustainability. Nutrient deficiencies jeopardize crop health and productivity, leading to decreased yields and economic losses. Traditional nutrient assessment methods, relying on manual observation or labor-intensive soil testing, are impractical for large-scale operations. Moreover, delayed intervention exacerbates the problem, resulting in substantial economic losses. Therefore, leveraging deep learning and IoT, we aim to develop a system that can accurately detect nutrient deficiencies in banana crops in real-time and autonomously deliver required nutrients, optimizing resource utilization and promoting sustainable farming.

B. Problem Statement:

This project addresses the inefficient detection and remediation of nutrient deficiencies in banana crops, causing suboptimal yields and resource wastage. Existing methods, relying on manual observation or labor-intensive soil testing, are impractical for large-scale operations, leading to delayed intervention and economic losses. The challenge is to develop a system that accurately identifies nutrient-deficient areas in crops, determines lacking nutrients, and autonomously delivers required nutrients promptly and precisely.

1) Research Objectives:: The primary objective is to design and implement a comprehensive solution for nutrient deficiency detection and remediation in banana crops using deep learning and IoT technologies. Specifically, the research aims to:

1) Develop an Android application integrated with the YOLOv5 object detection model for real-time crop health analysis via smartphone cameras.
2) Design and deploy IoT devices, particularly drones, for aerial surveillance of banana crops to identify nutrient-deficient areas.
3) Implement a communication protocol between drones and IoT Seeder devices controlled by a Raspberry Pi controller for automated nutrient delivery.
4) Optimize the YOLOv5 model for resource-constrained IoT devices through post-training quantization and other techniques, ensuring efficient inference without accuracy loss.
5) Evaluate the proposed solution's performance through empirical testing in real-world agricultural settings, assessing its efficacy in detecting nutrient deficiencies and improving crop yields.

II. Literature Review

This literature survey provides an in-depth exploration of the landscape of IoT in agriculture, focusing on existing research and emerging trends. It delves into diverse applications of IoT technologies in farming, ranging from precision agriculture to livestock management and environmental monitoring. The survey scrutinizes the utilization of sensors, data analytics, and connectivity solutions, offering a comprehensive overview of the methods employed to collect and process agricultural data. Furthermore, it investigates the challenges and opportunities associated with the integration of IoT into the agricultural sector, with a keen eye on sustainability and environmental concerns. This literature survey will serve as the cornerstone for our forthcoming methodology, which seeks to build upon the foundations laid by previous research while addressing their limitations. Our approach aims to create a more efficient and environmentally conscious IoT system for agriculture, fostering responsible and sustainable practices that are essential for the future of farming.

A. *Edge Intelligence: Architectures, Challenges, and Applications*

The paper titled "Edge Intelligence: Architectures, Challenges, and Applications" [1] is a comprehensive review paper that presents an overview of edge intelligence, which refers to the integration of artificial intelligence (AI) and edge computing technologies to enable intelligent processing of data at the edge of the network. The paper begins by discussing the motivation for edge intelligence, which includes the need to reduce latency, improve network bandwidth efficiency, and increase security and privacy.

The paper reviews the main components of edge intelligence architectures, including edge devices, edge gateways, and edge servers, and highlights the challenges facing the design and implementation of these architectures. The authors discuss the key features of edge devices, such as their low power consumption and mobility, and the challenges associated with designing intelligent algorithms for these devices. The authors also review the different types of edge gateways, including fog computing, cloudlets, and mobile edge computing, and the advantages and disadvantages of each type.

As shown in fig. 1, In edge intelligence, intelligent application tasks are carried out at the edge using locally-generated data in a distributed fashion, as contrast to traditional intelligence where all data must be uploaded to a central cloud server.

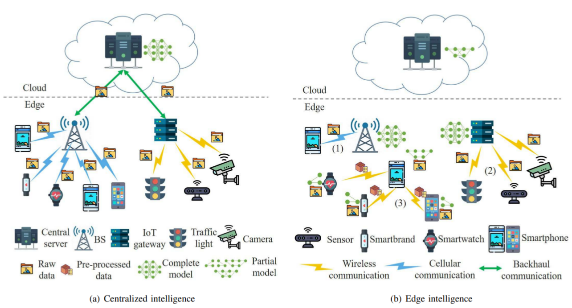

Fig. 1. An analysis of traditional intelligence versus edge intelligence from an implementation standpoint

The paper provides a comprehensive review of the current state-of-the-art in edge intelligence applications, including intelligent transportation systems, smart homes, and precision agriculture. The authors discuss the benefits of edge intelligence in these applications, such as improved safety and efficiency in transportation, enhanced energy efficiency and comfort in smart homes, and optimized crop management practices in precision agriculture.

The paper also highlights several challenges and future directions for edge intelligence research, including the need for more advanced machine learning algorithms for edge devices, the development of more efficient and secure communication protocols for edge networks, and the integration of edge intelligence technologies with other emerging technologies such as block-chain and 5G.

B. *Edge-Based Video Analytics: A Survey*

The authors outline and contrast a number of current edge-based video analytics system architectures in this paper [2]. By processing video data at the network edge, these architectures lower latency and enhance real-time processing. The following are the main architectures that were covered:

1) **Edge/Fog-based Architecture**
   - Real-time computing and low-latency processing are made possible by edge computing, also known as fog computing, which brings cloud computing to the edge of the network.
   - This architecture typically comprises of two tiers: end device and edge computing, allowing for real-time video analytics.
   - It can be categorized into dedicated edge-based architecture (with dedicated servers for each camera) and shared edge-based architecture (where cameras share resources on a single edge server).
   - Shared edge-based architecture is preferred in cases with a high number of cameras.
2) **P2P-based Architecture**
   - This architecture focuses on improving analytics performance by utilizing cross-camera correlations.
   - It enables any appropriate camera to assign an analytics job to process a video stream in real-time.
   - The inference results are shared between analytics pipelines, allowing for dynamic task assignment based on video streams.
3) **Hierarchical Architecture**
   - There are three layers in the hierarchical architecture: end device layer, edge layer, and cloud layer.
   - Each layer may have different hardware types, such as GPUs, FPGAs, and ASICs.
   - This architecture allows for the flexible allocation of computational resources across different layers and nodes, reducing latency and improving energy efficiency.
   - The hierarchical architecture has been found to be efficient and scalable for edge-based video analytics.

C. Literature Review on Unmanned Aerial Spray Systems in Agriculture

A thorough analysis of the state-of-the-art in unmanned aerial vehicle (UAV) use for agricultural spraying is given in the paper "Literature Review on Unmanned Aerial Spray Systems in Agriculture" [3].

The authors then provide an overview of the main components of UAV-based agricultural spraying systems, including the UAV platform, the spray system, and the navigation and control systems. The authors discuss the key features of these systems, such as the size and weight of the UAV, the capacity and efficiency of the spray system, and the accuracy and reliability of the navigation and control systems.

The paper then provides a comprehensive review of the current state-of-the-art in UAV-based agricultural spraying technology. The authors discuss the different types of UAV platforms used in agriculture, including fixed-wing, rotary-wing, and hybrid UAVs, and the advantages and disadvantages of each type. The authors also review the different types of spray systems used in agriculture, including boom sprayers, rotary atomizers, and electrostatic sprayers, and the factors that influence the efficiency and effectiveness of each system.

The authors then review several case studies of UAV-based agricultural spraying, which illustrate the potential benefits of this technology for improving efficiency, reducing environmental impact, and increasing profitability. The case studies include examples of precision spraying, variable rate spraying, and spot spraying.

Finally, the authors discuss several challenges and future directions for research in UAV-based agricultural spraying, including the need for more research on the environmental impact of UAV spraying, the development of more advanced navigation and control systems, and the integration of UAV spraying systems with other precision agriculture technologies such as GPS and GIS.

Limitations: While employing UAVs in agriculture appears to offer significant benefits, it also comes with certain limitations. These limitations are as follows:

- **Regulatory Issues:** UAV operations are subject to various regulations, which can vary by country and even region. These regulations may restrict the use of drones in agriculture, and compliance can be time-consuming and expensive.
- **Cost:** Acquiring, maintaining, and operating UAVs can be expensive. High-quality, specialized agricultural drones can be a significant upfront investment, and ongoing costs can include maintenance, repairs, and software updates.
- **Limited Payload Capacity:** Agricultural tasks often require the use of sensors, cameras, and other equipment. Many UAVs have limited payload capacities, which may restrict the types of sensors and cameras that can be used, and reduce the range and duration of their operations.
- **Weather Dependency:** Drones are weather-sensitive devices. Drone flying can be dangerous or impossible in the event of rain, strong winds, or other unfavorable weather. This limits their availability for certain critical tasks and during specific seasons.
- **Limited Flight Time:** Most agricultural drones have limited flight times, typically ranging from 30 minutes to a few hours. This may necessitate frequent recharging or battery replacement, which can be time-consuming and limit the area that can be covered in a single flight.
- **Skill and Training:** Drone operators need to be highly skilled and trained in both data analysis and flight operations. The training can be time-consuming, and not all farmers or agricultural workers may have access to this expertise.
- **Interference:** In areas with a high density of other wireless devices or in regions with limited connectivity, signal interference can disrupt drone operations and data transmission.
- **Limited Automation:** While there have been advancements in automation, many agricultural drone operations still require a significant level of manual intervention, including takeoff, landing, and data analysis.
- **Crop and Terrain Variability:** Drones may struggle to adapt to the variability of crops and terrain. Low-altitude flying drones may encounter obstacles, and the effectiveness of sensors can be limited by factors like crop height, density, and color.

D. Edge-Based Live Video Analytics

In the paper titled "Edge-Based Live Video Analytics for Drones" by J. Wang et al. [4], published in IEEE Internet Computing, the authors provide a comprehensive exploration of the applications and advancements in live video analytics for drone technology. Drones have gained increasing popularity across various domains, such as surveillance, agriculture, construction, and environmental monitoring. Real-time video analytics plays a pivotal role in enhancing the capabilities and functionalities of drones in these applications.

The following algorithms and strategies are discussed in the paper to optimize live video analytics for drone technology:

- **Early Discard Strategy:** By using on-board processing to filter and send only pertinent data, the Early Discard technique efficiently uses bandwidth. It makes use of the Early Discard Algorithm for object recognition and image classification, which makes use of the MobileNet model. Transfer learning is used to refine the pretrained MobileNet model with a training set of sample photos taken from an aerial perspective in order to identify mission-specific targets. For improved analysis, high-resolution frames are also tiled.
- **Sampling + Early Discard:** The combination of sampling and Early Discard [Fig. 2] involves selecting a subset of frames from the video stream based on factors like scene dynamics and motion detection. This intelligent frame selection reduces the consumption of processing resources and bandwidth while maintaining adequate

recall rates, with the possibility of transmitting as few as one frame per second for selected tasks.

- **JITL Strategy (Just-in-Time Learning):** In order to maximize the benefits of early discard, the JITL technique [Fig. 2] attempts to modify the drone's pipeline to the demands of the present mission. To distinguish between true positives and false positives produced by the Early Discard filters, a reasonably priced classifier is used. JITL filtering is applied to frames that Early Discard detects as positive; if the filter returns a negative result, the frame is regarded as a false positive and is not communicated. For the JITL filter, the implementation uses a linear support vector machine (SVM), which has been proved through experiments to be able to filter out an extra 15% of frames after Early Discard without compromising recall rates for most jobs.

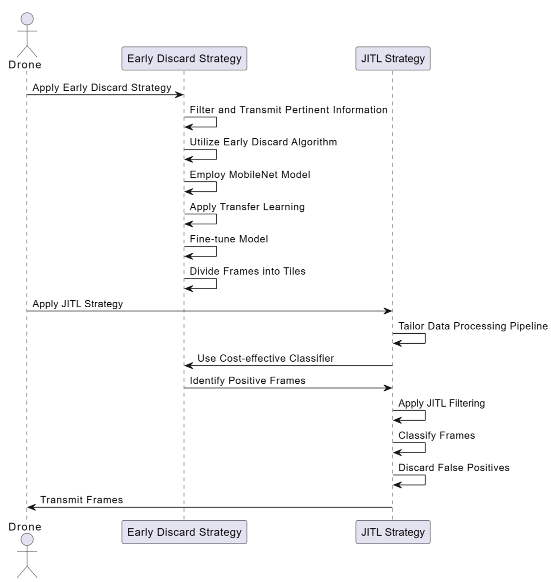

Fig. 2. Smart Image Filtering: A Sequence Diagram of Early Discard, Sampling, and JITL Techniques

- **Reach-Back Strategy(Specific to Activity Detection):** Complete videos are saved locally on the drone as authoritative sources in the live video analytics system, where only segments of the video feeds are transmitted back to cloudlets for analysis. In order to finish the study, the reachback approach makes it easier to get more frames from the drone's storage. In situations involving activity detection, where a series of frames is needed to precisely identify actions inside the scene, this approach shows to be extremely helpful.
- **Context-Aware Strategy:** The core of the context-aware approach is to leverage the special qualities of the current mission to improve the drone's video processing speed and accuracy. Additionally, it enables human operators to recognize distinguishing features that can be leveraged to develop more efficient context-aware filters, enhancing precision and lowering on-board computation.

The paper demonstrates that these strategies and techniques are essential for optimizing live video analytics in the context of drone technology, enabling efficient data processing and decision-making in real-time scenarios.

E. Related Works

In previous research, Divyanshu Tirkey created an Android app for the detection and classification of crop diseases [5], and their proposed solution achieved accuracy rates of 98.75%, 97%, and 97% using YoloV5, InceptionV3, and CNN models, respectively. Manoj Kumar R and Bala Murugan MS [6] introduced an AI-powered drone system for cashew farming. It focuses on early disease detection and precision pesticide management, demonstrating the integration of AI, IoT, and drone technology in agriculture. With the aid of unmanned aerial vehicles (UAVs), Tej Bahadur Shahi and colleagues conducted research on the early detection of crop diseases using machine learning and deep learning techniques [7]. They found that DL methods were more efficient than the ML methods as they were classfying crops with an accuracy of 93% to 97%.

Anu Jose and S. Nandagopalan identified various nutrient deficiencies [8], including nitrogen, phosphorous, potassium, magnesium, calcium, and sulfur in tomato crops. Their approach involved various image processing techniques, such as resizing, enhancement, noise reduction, image segmentation, feature extraction, and classification. The system achieved a range of accuracy rates, with magnesium deficiency detection reaching the highest accuracy at 93% and sulfur deficiency detection attaining the lowest at 68%.

III. Objective of Work

The aspect of Precision Agriculture will be carried out practically as follows:

- Scanning of the cropland in order to find areas having lower nutritional content (using UV spectroscopy or real-time analysis) and marking them on the given map.
- Communicating the details of areas found and the type of nutrition deficiency to the system or server.
- The system/server receives this information (given by the drone).
- The system will analyze the locations and will decide which seeder to send to which location and with which nutritional content.
- The system will then send signals to seeders which it selects by its algorithm.
- The seeder will collect the information sent by the system or server and starts its journey to reach that particular location.
- During the journey, the seeder will take pictures in front of it from time to time and make decisions in real time about which direction it should move to in order to reach that particular location.
- Seeder will need a bigger picture of the field too in order to reach the location with precision, which will be communicated via drone and server.
- Since the path from origin to destination is not obstacle free, the seeder is required to do some computations for path finding in real time.
- The seeder reaches its destination and precisely provides the required nutritions on the crop roots.

A general summarized workflow of the system is shown below:

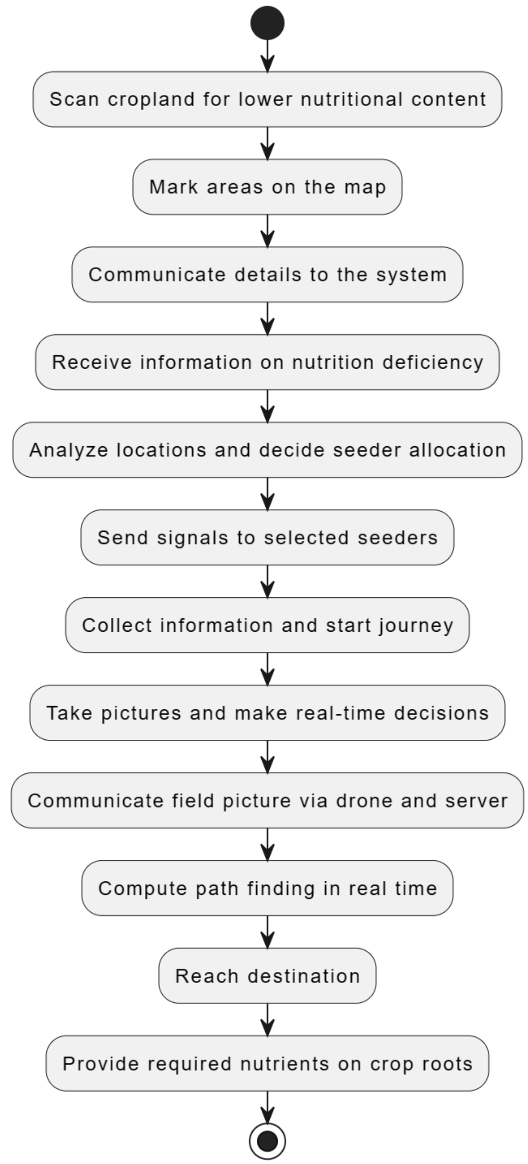

Fig. 3. Summarized workflow of the system

IV. Hardware and Software requirements

Types of Hardware Required:

- **UAV Drone** For scanning crop fields, marking the nutrition deficient areas on the map, communicating the data to the server
- **Remote Control Device** For controlling the drone flight
- **Server** This is a fog computing server device responsible for the communication between the Drone and the Seeders (Task executors). It also does the computations for selection of Task executor for doing a particular task.
- **Seeder** This is Task executor of the framework which does the actual tasks. It also does the real time computa-

tion of path finding using the current view image along with the big picture of the field captured by the drone.

V. Hardware Recommendations

A. Drone Device

Parrot Sequoia drone: The Parrot Sequoia is a drone that is specifically designed for agricultural applications. It comes with a built-in multi-spectral camera that captures images in four distinct bands of light, including red, green, red-edge, and near-infrared. These images can be analyzed using specialized software to identify areas of nutrient deficiency, pest infestation, and other crop health issues.

The Parrot Sequoia drone is equipped with GPS for accurate positioning and can be flown manually or programmed to fly specific routes using GPS way points. The drone also has an autopilot feature that allows it to fly pre-planned missions, which can be customized based on the specific needs of the crop.

B. Seeder Device

Custom IoT-Based Robotic Seeder Machine: Our project entails the creation of a cutting-edge robotic seeder machine with an intelligent sprayer system, enhancing precision in crop treatment. This device utilizes GPS coordinates for targeted application and deploys path-finding algorithms for autonomous navigation. Additionally, it boasts a front-facing camera for real-time crop row detection and path adjustment, making it an efficient and precise tool for modern agriculture.

C. Computing Server device

Raspberry Pi 4: The Raspberry Pi 4 is a small and affordable single-board computer that can serve as a central computing device for precision agriculture applications. It has built-in WiFi and Bluetooth capabilities for wireless communication, as well as GP-IO pins for connecting to other devices.

The Raspberry Pi 4 can run a variety of operating systems and programming languages, including Python, which is commonly used in agricultural applications. You can program the Raspberry Pi 4 to receive GPS locations from other devices and use those locations to make decisions about which devices to select for specific purposes. You can also use it to run a scheduling algorithm to optimize the use of resources and devices.

VI. Network Architectures used in our project

Within our project, we have implemented two distinct network architectures to facilitate seamless communication and coordination between critical components:

A. Server and Drone

Exclusive pair model: This model establishes a dedicated connection between the server and the drone, ensuring a full-duplex communication channel. The necessity for real-time data transmission is paramount in this context. It allows the drone to transmit a comprehensive view of the crop field to the server, empowering the task executors to make informed, real-time decisions for efficient path finding.

B. Server and Seeder

Request-Response Model: The communication model employed between the server and the seeder is based on the request-response paradigm. When a task executor becomes available, it sends a request to the server, indicating its readiness to undertake a job. In response, the server promptly provides the job location and pertinent details. Upon task completion, this process repeats, ensuring a continuous and efficient workflow.

VII. Methodology for Nutrient Deficiency Detection in Banana Crops

In this section, we detail the methodology utilized for detecting and managing nutrient deficiencies in banana crops, incorporating state-of-the-art deep learning techniques and mobile technology.

A. Method Overview:

Our approach involves several interconnected steps, starting with the deployment of the YOLO object detection model. The methodology includes:

1) Nutrient Deficiency Detection: Real-time analysis using the YOLO model on video data to detect nutrient deficiencies.
2) Android Application Development: Creation of an app for easy user interaction and data processing.
3) Model Conversion: Transitioning the YOLO model from PyTorch to TensorFlow Lite (tflite) format to ensure compatibility with mobile platforms.
4) Integration and Optimization: Incorporating the converted tflite model into the Android application followed by applying post-training quantization and quantization-aware training to enhance model efficiency and performance.
5) Continuous Analysis: Generating and evaluating analytical results at each stage to ensure optimal functionality and accuracy.

B. Flow of Operations:

- Video Processing: Utilizing video feeds from drone-captured footage of banana crops, our system analyzes frames extracted via an Early Discard filter, which selects significant frames for further processing.
- Real-Time Object Detection: Frames are processed using the YOLOv5 model to identify and classify nutrient deficiencies in banana leaves. Detected deficiencies are labeled with specific nutrient information and tagged with GPS coordinates from the drone.
- Data Handling and Device Communication: Deficiency data along with GPS locations are sent to a server. This information orchestrates the deployment of seeder devices, equipped with real-time detection capabilities for precise nutrient dispensation.

1) Deployment and Server Interaction::

- Seeder Device Navigation: Using object detection and path-finding algorithms, seeder devices navigate the crop fields to address identified nutrient deficiencies. This involves obstacle detection and avoidance to ensure efficient and safe nutrient application.
- Server Coordination: A server manages the flow of information between the drone and seeder devices, using ParaDrop APIs for robust edge device communication.

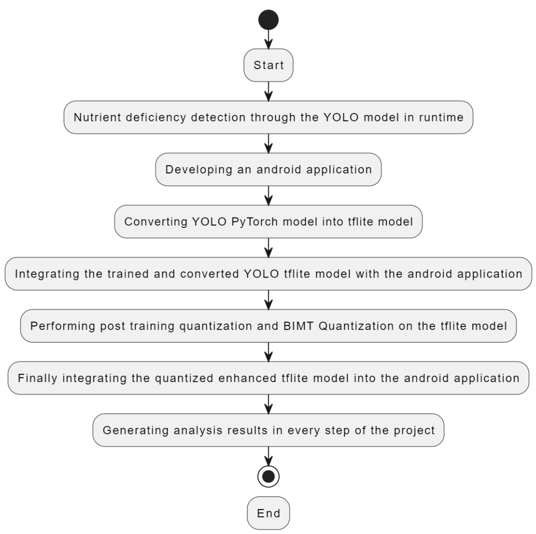

Fig. 4.

2) Empirical Evaluation::

- Dataset Utilization: A comprehensive dataset containing images of banana leaves with specific nutrient deficiencies is used to train the YOLOv5 classifier.
- Performance Testing: The system's efficacy is evaluated through empirical testing using test images to measure the classifier's accuracy in detecting and classifying nutrient deficiencies.

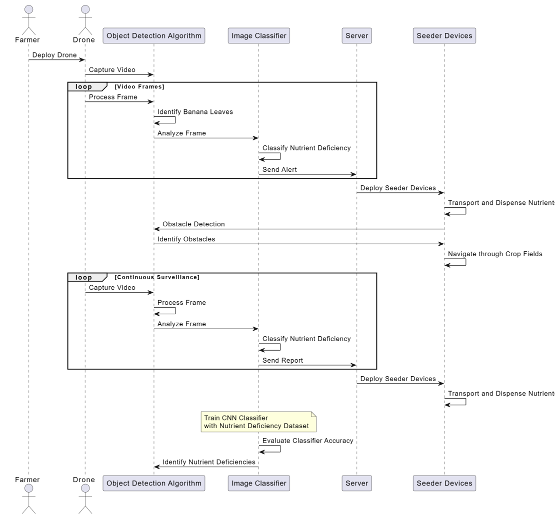

Fig. 5. Smart Agriculture: Enhancing Banana Crop Health with Drone-Based Nutrient Deficiency Detection

Figures integrated into this section provide a visual representation of the methodology and the sequence of actions, from detection through intervention, illustrating the full lifecycle of nutrient deficiency management in banana crops. These diagrams not only clarify the process but also highlight the innovative integration of advanced technologies in precision agriculture.

VIII. Technical Aspect of the Problem

The technical aspects pertaining to the research problem are as follows:

1) **Crop Selection:** The crop chosen for this research is the banana plant. This selection is based on the characteristic wide and large leaves of the banana plant, which make it an ideal candidate for the drone-based identification of nutrient deficiencies.
2) **Dataset Details:** The dataset utilized in this study was sourced from Mendely and comprises over 9000 images of banana leaves exhibiting deficiencies in nine distinct classes of nutrients. These nutrients include boron, calcium, iron, healthy, potassium, manganese, magnesium, sulphur, and zinc.
3) **Model Selection:** To perform classification tasks on the dataset, the YOLOv5 model was chosen as the deep learning architecture. This model is renowned for its efficiency and accuracy in video classification tasks.

IX. Why YOLO?

Utilizing the YOLOv5 model in the project for the detection and management of nutrient deficiencies in banana crops offers several advantages:

1) **Real-Time Detection:** YOLOv5 is renowned for its efficiency in real-time object detection. This capability allows the application to analyze crop health instantly, providing timely intervention for nutrient deficiencies.

2) **High Accuracy:** YOLOv5 has demonstrated impressive accuracy in object detection tasks, even when compared to earlier versions of YOLO. This ensures reliable detection of nutrient deficiencies in banana crops, reducing false positives and negatives.
3) **Lightweight Model:** YOLOv5 is designed to be more lightweight and efficient compared to its predecessors. This makes it suitable for deployment on resource-constrained devices such as drones and IoT devices, without compromising on performance.
4) **Flexibility:** YOLOv5 supports various backbone architectures (e.g., S, M, L, X) that allow users to choose the right balance between speed and accuracy based on project requirements. This flexibility enables customization to suit different deployment scenarios and hardware constraints.
5) **Ease of Deployment:** YOLOv5 can be easily integrated into Android applications and deployed on IoT devices after conversion to tensorflow format. Its compatibility with popular deep learning frameworks like PyTorch and TensorFlow facilitates seamless integration and deployment pipelines.
6) **Active Development and Support:** YOLOv5 is actively maintained and developed by a vibrant community, ensuring ongoing improvements, updates, and support. This ensures that projects benefit from the latest advancements and optimizations in object detection technology.
7) **State-of-the-Art Performance:** YOLOv5 represents the latest advancements in object detection models, incorporating state-of-the-art techniques and optimizations. By leveraging YOLOv5, projects can benefit from cutting-edge performance and capabilities in nutrient deficiency detection.

YOLOv5		Faster R-CNN [43]				
Y_m	Y_S	ResNet50 (FPN)	VGG16	MVGG16	Mobile-Net V2	Inception V3
86.96%	76.73%	91.9%	69.8%	81.4%	63.1%	72.3%
0.017	0.020	0.065	0.226	0.136	0.209	0.194
61.54%	58.9%	64.12%	35.3%	45.4%	30.5%	32.3%
0.012 s	0.009 s	0.098 s	0.114 s	0.047 s	0.036 s	0.052 s
0.013 s	0.009 s	0.065 s	0.119 s	0.052 s	0.032 s	0.056 s
16 s	12 s	124 s	173 s	105 s	80 s	95 s
19,200 s	14,400 s	12,400 s	17,300 s	10,500 s	8000 s	9500 s
43.3	14.8	165.7	175.5	134.5	329.8	417.2

Fig. 6. Comparison of YOLO with other CNN models [9]

X. Work Stages

The primary objectives of this research are as follows:

1) Collect a dataset of images illustrating common banana diseases resulting from macronutrient deficiencies.
2) Divide the dataset into appropriate training and testing sets.
3) Train the YOLOv5 classifier using the training image set.
4) Evaluate the model's accuracy with the test image set and monitor its performance.
5) Convert the YOLOv5 .pt model into a .tflite model in order to deploy it in an Android application (since the TensorFlow library is supported by Android Studio for deployment of models and not PyTorch).
6) Build an Android application using Java and integrate the tflite model, then test it on raw banana crop fields to evaluate its inference time and accuracy.
7) Perform post-training quantization of the model to reduce its precision from floating-point values to integer values.
8) Integrate the quantized model into the Android application again and compare its inference time with the previous model.
9) Now perform quantization-aware training and repeat the same steps as above.
10) Finally, compare the performance of all three models and declare the best model for object detection.

XI. Results

A. Yolov5 Integration

The YOLOv5 model underwent training on a dataset comprising approximately 9k images of banana leaves, showcasing 9 distinct nutrient deficiency classes: boron, calcium, healthy, iron, potassium, magnesium, manganese, sulfur, and zinc [**?**]. Training occurred over 200 epochs, employing a batch size of 16 and an image size of 255 x 255. The train-test split was 80:20. The Mean Average Precision (MAP) achieved was 78 percent, with a cls loss of 0.002843. The optimal weights model amounted to 13.6 MB in size. Given below, are the results after training the yolov5 model.

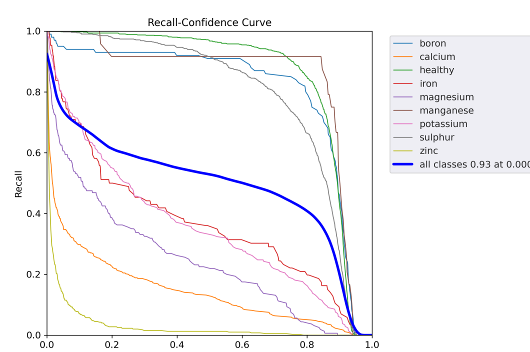

Fig. 7. R curve

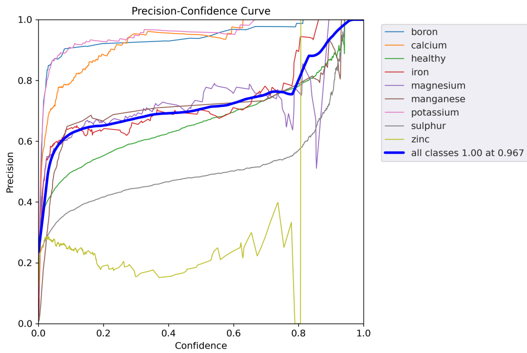

Fig. 8. P curve

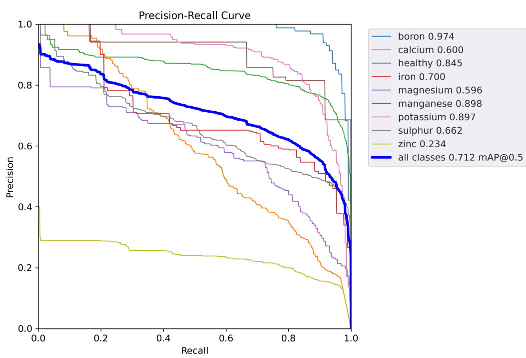

Fig. 9. PR curve

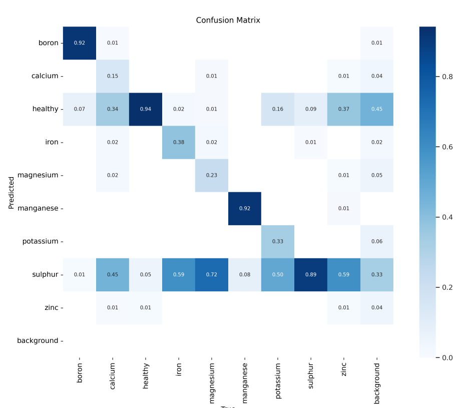

Fig. 10. Confusion matrix of YOLOv5

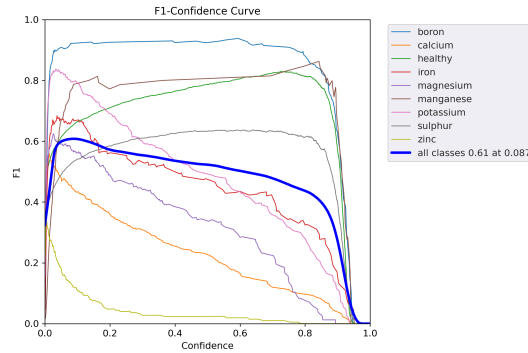

Fig. 11. F1 curve

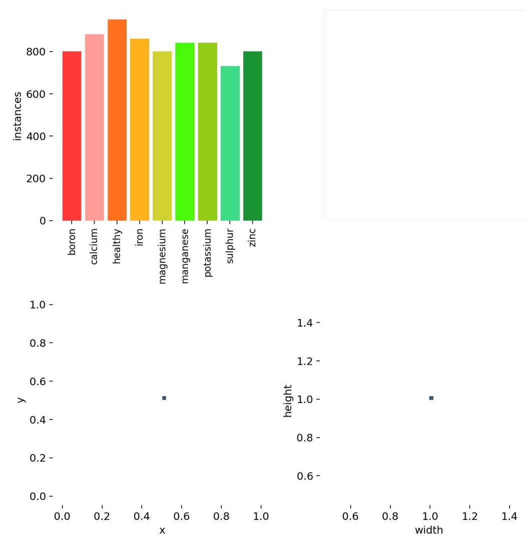

Fig. 12. Labels

Subsequently, the model underwent conversion from PyTorch (pt) format to TensorFlow Lite (tflite) format, given that Android Studio exclusively supports application development utilizing the TensorFlow library. Post-conversion, the new model weights were reduced to 13.5 MB.

Integration of the tflite model was undertaken to build the Android application, facilitating real-time classification of banana leaves across the 9 nutrient deficiency types. The application also provides insights into the model's inference time for banana leaf classification. Example images of the application are displayed below.

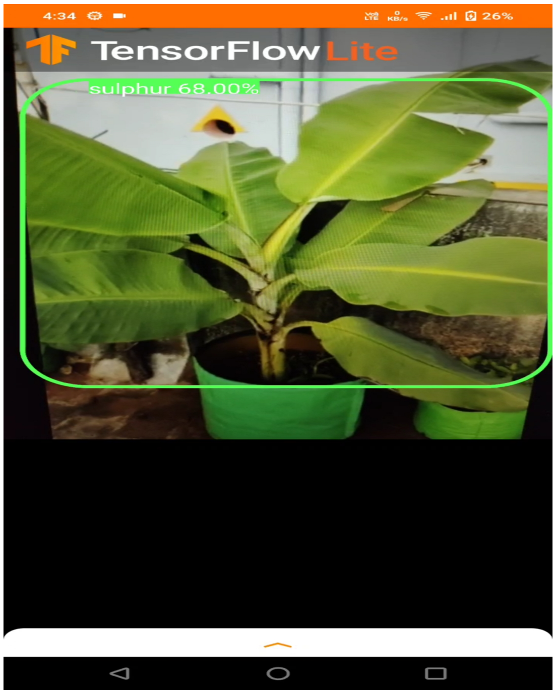

Fig. 13. Tflite Android App-1

Fig. 14. Tflite Android App-2

Fig. 15. Tflite Android App-3

Following Android app testing, further optimization of the model was pursued to ensure compatibility with IoT devices like drones and seeders, which possess limited resources such as memory, processing power, and battery capacity. To this end, Post-training Quantization and Quantization Aware Training (QAT) were implemented.

- **Post-training quantization** involves the reduction of the YOLOv5 model's size and computational demands by converting its weights and activations from floating-point precision to lower precision formats, such as 8-bit integers. This optimization is crucial for deployment on IoT devices like drones, which operate within resource constraints and energy efficiency requirements. TensorFlow Lite offers built-in support for post-training quantization, employing techniques such as dynamic range quantization and full integer quantization.
  - **Dynamic Range Quantization** entails the quantization of model weights to 8-bit integers while retaining activations in floating-point format, facilitating efficient computation on hardware with optimized instructions for mixed-precision operations.
  - **Full Integer Quantization** extends quantization to both the model's weights and activations, further diminishing the model's size and computational needs. However, additional calibration steps may be necessary to preserve accuracy.
- **Quantization Aware Training (QAT)** is a deep learning technique aimed at training models for efficient inference on hardware with limited computational resources. During QAT, the training process incorporates awareness of quantization effects, such as rounding and quantization

noise, applied to weights and activations. This enables the model to adapt to the reduced precision required for deployment, minimizing the accuracy loss typically observed during the transition from high-precision floating-point arithmetic to low-precision fixed-point arithmetic.

After performing these quantization techniques, we were able to reduce the size of our trained YOLOv5 model from 13.5 MB to 3.7 MB while also maintaining the previous accuracy (MAP). We then deployed the quantized TFLite models into our Android application to test them in the field. Additionally, this will help resource-constrained IoT devices to fit the model inside them and run appropriately.

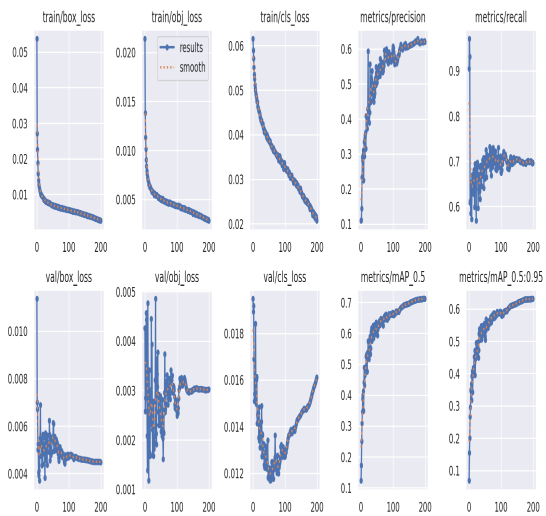

Fig. 16. Results

- **Original YOLOv5 Model**
  Upon training the YOLOv5 model on a dataset consisting of approximately 8000 images of banana leaves containing 9 classes of nutrient deficiency, the best-performing model, denoted as `best.pt`, was obtained. This model exhibited the following characteristics during testing on a sample set of images:
  - **Model Size:** The size of the `best.pt` model was measured at approximately 13.5 MB.
  - **Inference Speed:**
    * Pre-process: 0.7 milliseconds per image.
    * Inference: 20.7 milliseconds per image.
    * Non-Maximum Suppression (NMS): 111.2 milliseconds per image.
  - **Image Shape:** The model processed images at a shape of $(1, 3, 256, 256)$.
- **Quantized YOLOv5 Model (TFLite)**
  Post-training quantization techniques were applied to the original YOLOv5 model to obtain a quantized model in TensorFlow Lite (TFLite) format, denoted as `quantized_model.tflite`. This model aimed to reduce the model size while maintaining acceptable performance. The resulting characteristics during testing on the same set of images were as follows:
  - **Model Size:** The size of the quantized `quantized_model.tflite` was reduced to approximately 3.7 MB, representing a significant reduction from the original model size.
  - **Inference Speed:**
    * Pre-process: 0.6 milliseconds per image.
    * Inference: 140.4 milliseconds per image.
    * Non-Maximum Suppression (NMS): 1.6 milliseconds per image.
  - **Image Shape:** The model processed images at a shape of $(1, 3, 256, 256)$.

Analysis

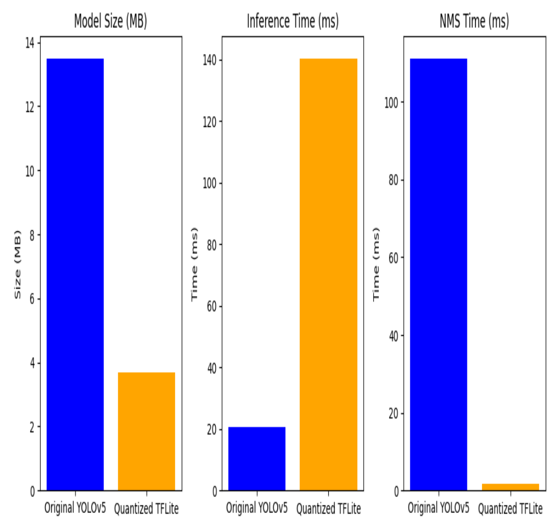

Fig. 17. Comparison of original fully trained YOLOv5 model with the Quantized model

Comparing the results between the original YOLOv5 model and the quantized TFLite model, several observations can be made: - The quantized model achieved a substantial reduction in size, indicating its potential for deployment on resource-constrained devices or environments where memory is limited. - However, there was a noticeable increase in inference time, particularly in the NMS stage, with the quantized model. This suggests a trade-off between model size reduction and inference speed, which should be considered based on the specific requirements of the deployment scenario. - Further optimization techniques or fine-tuning may be explored to mitigate the increase in inference time while preserving the benefits of model quantization.

References

[1] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Architectures, challenges, and applications," *Journal Name*, vol. Volume Number, p. Page Range, 2018.

[2] M. Hu, Z. Luo, A. Pasdar, Y. C. Lee, Y. Zhou, and D. Wu, "Edge-based video analytics: A survey," 2023.

[3] H. Chen, Y. Lan, B. K. Fritz, W. C. Hoffmann, and S. Liu, "Review of agricultural spraying technologies for plant protection using unmanned aerial vehicle (uav)," *OECD*, vol. NA, p. NA, 2020.

[4] J. Wang, Z. Feng, Z. Chen, S. A. George, M. Bala, P. Pillai, S.-W. Yang, and M. Satyanarayanan, "Edge-based live video analytics for drones," *IEEE Internet Computing*, vol. 23, no. 4, pp. 27–34, 2019.

[5] D. Tirkey, K. K. Singh, and S. Tripathi, "Performance analysis of ai-based solutions for crop disease identification, detection, and classification," *Smart Agricultural Technology*, vol. 5, p. 100238, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375523000680

[6] M. Kumar and B. M. Murugan, "Artificial intelligence based drone for early disease detection and precision pesticide management in cashew farming," *Journal Name or Conference Name*, vol. NA, p. NA, 2023.

[7] T. B. Shahi, C.-Y. Xu, A. Neupane, and W. Guo, "Recent advances in crop disease detection using uav and deep learning techniques," *Remote Sensing*, vol. 15, no. 9, p. 2450, May 2023. [Online]. Available: http://dx.doi.org/10.3390/rs15092450

[8] V. U. D. V. Anu Jose, S Nandagopalan, "Detection and classification of nutrient deficiencies in plants using machine learning," *J. Phys.: Conf. Ser.*, vol. 1850, p. 012050, 2021.

[9] "Smart pothole detection using deep learning based on dilated convolution," Scientific Figure on ResearchGate, accessed on 16 Apr, 2024. [Online]. Available: https://www.researchgate.net/figure/Comparison-of-YOLOv5-and-Faster-R-CNN-performance$_tbl1_3$57093620