International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 12 Issue 5 ǁ May 2024 ǁ PP. 369-375

Optimised Latency on Integrated Model for Detection and Prevention of Diseases in Sweet Potatoes

Laetecia.N Onyejegbu, Temitope Victor-Ime, Chidiebere Ugwu
laeticia.onyejegbu@uniport.edu.ng, xterakojede@gmail.com, chidiebere.ugwu@uniport.edu.ng

Abstract- *Network latency in the IOT system caused significant delays because of the 4 layers of IOT in existing research models in transferring data in record time for real monitoring. The problem is the need for an efficient and automated system that can accurately and timely detect and manage sweet potato pests and diseases. This work established a hybridized architecture combining Deep Learning and the Internet of Things (IoT) for managing illnesses and pests of sweet potatoes. The Object-Oriented Analysis and Design Methodology (OOADM) was employed in the system's design. The hybridized architecture comprised of the Ardunio ESP32 Processor and a raspberry Pi 4 processor, powered by solar. The existing system had a computational cost of training for 9hours, 30minutes, while the proposed system training lasted for 5 hours, 30minutes, with 4 frames per second of camera capture. This implied that the proposed system saved 4 hours out of the Training and increased the latency rate of the model for remote monitoring.*

Index Terms—*Artificial Intelligence, Internet ofThings IoT, Deep Learning, Latency*

Date of Submission: 16-05-2024 Date of acceptance: 31-05-2024

I. Introduction

Even while cloud computing powers many today's always-connected electronic gadgets, Internet of Things (IoT) manufacturers and application developers are starting to realize the benefits of doing additional computation and analytics on the devices directly. This on-device technique lessens reliance on the cloud, lowers latency for critical applications, and enhances data management for the massive volume of data generated by IoT [2]. The advancement of machine learning for Internet of Things applications is driving increased edge compute capabilities. Since many IoT devices are battery-powered, devices need to be able to run complex deep learning networks quickly while using very little power. This is what motivates the usage of diverse engines, such CPUs and GPUs, in heterogeneous computing architectures found in Internet of Things devices. in order to improve performance and power economy by allocating various workloads to the most efficient computing engine. Edge devices, situated at the edge of the Internet, can have their idle processing power and unused storage space utilized by edge-based deployments. Subsequently, devices nearer to the sites of data production and consumption will handle some of the data processing and storage, as a more environmentally friendly solution (Angel, et al., 2022).

Growing sweet potatoes is a major source of income and sustenance for many communities across the world. However, a number of pests and diseases can seriously impair the output and quality of sweet potato crops. Large-scale farming operations cannot afford to use traditional methods of pest and disease identification and management because they are frequently labor-intensive, time-consuming, and dependent on specialized knowledge. to handle the challenge of controlling diseases and pests in agricultural and urban settings. A recognized paradigm for delivering high bandwidth or low latency for mobile and Internet of Things applications is edge computing. The allure of edge computing has grown even more with the current availability of special-purpose technologies to accelerate specialized compute tasks, like deep learning inference, on edge nodes. In this study, we examine the benefits and drawbacks of adopting customized edge systems constructed using edge accelerators in comparison to more conventional edge and cloud computing models [3].

II. Review of Related Works

These days, businesses are fully integrating the internet of things into their current IT systems. But preparing to switch to IoT is one thing; putting it into practice successfully is quite another. When creating an IoT application, there are a number of things to take into account. Over the years, organizations have employed a range of architectures, but the four layer and seven layer structures are particularly notable (Sarveshbhatt, 2019).

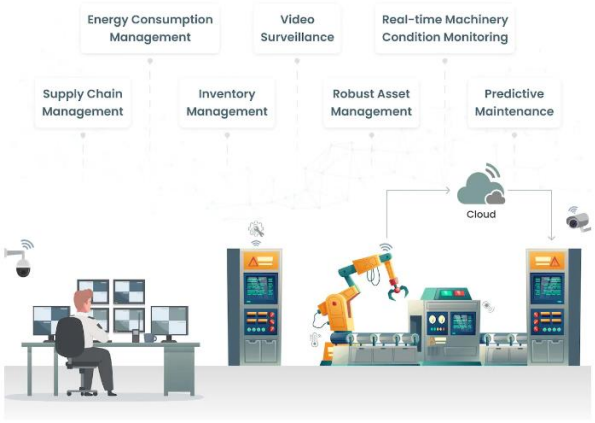

Fig. 1. Several Remote Monitoring Use Case Models (Kamal, 2023)

Jonathan et al. (2023) presented a field-based advising system that provides real-time feedback on crop disease detection. Their recommender system was built on question-answer pairings and leverages machine learning and natural language processing techniques. Among the algorithms they evaluated and tested are some of the most advanced in the field. The best working model was RetBERT, a phrase BERT model with a BLEU score of 50.8%, which we believe wass restricted by the limited amount of accessible data. The application solution combines online and offline services because the majority of farmers come from remote areas with inadequate internet connectivity. In the event that this investigation is successful, a large-scale trial will be carried out to verify its appropriateness for application in addressing the problem of sub-Saharan food security.

Cedric et al. (2023) proposed an Edge-AI device that includes the necessary hardware and software components for automatically identifying plant illnesses from a set of plant leaf pictures. The creation of an autonomous device that facilitated the identification of potential plant diseases was the main goal of this technique. This was accomplished by taking many pictures of the leaves and strengthening the categorization process with data fusion techniques. Numerous investigations have shown that using this device significantly increases the robustness of the classification reactions to possible plant illnesses. The gap that was left is the suggested system's inability to detect changes in the severity of plant diseases.

A dense leaf recognition system provided by Wang et al. (2022) and tested in a sweet potato farm. A redesigned Faster R-CNN framework and a visual recognition mechanism are the basis of a sweet potato leaf identification system that aims to improve the detection efficiency of plant leaves in natural settings with high obstruction overlapping and or form change. The proposed method performed well in identifying dense or hidden leaves and could provide valuable approaches for applications in smart agriculture and environmental inspection, like plant identification and growth monitoring.

III. Methodology

Object-Oriented Design Methodology (OODM) was employed in this proposed system. One of the elements of OODM, designed for this solution is the class diagram. This diagram depicts the overview of the classes within the model and provides details on the distinctive attributes and methods. In Fig.2. The class diagrams comprise of 5 classes namely, User, MonitorFarm, FarmParameters, AnomalyDetected, NoArnomaly. Starting from 'User', this class is designed to start with the name attribute and get updates to send to the 'MonitorFarm' class. The 'MonitorFarm class" is a class designed with the plot number, date and time attributes to continue the process and check for sweet potato pest and disease. If found "Yes", the process continues to read the " Farm Parameters" class with the nutrients, disease, pest and nutrients attributes, checking for nitrogen, potassium and phosphorous readings or values, if Yes, it returns the "AnomalyDetected" class designed with the date and time attributes, checking for the farm affected. If No, it returns with the "NoAnomality Detected class".

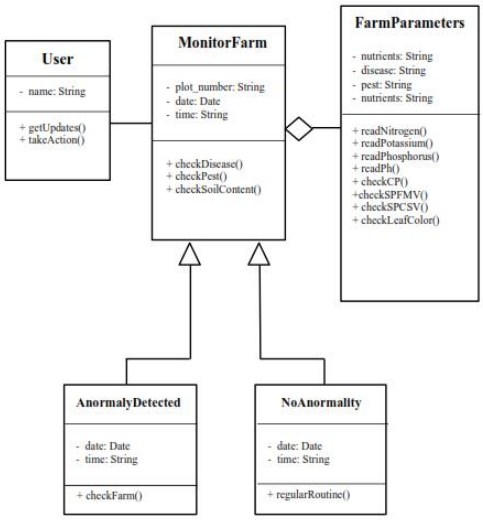

Fig. 2. Class Diargam of the Hybridized Architecture

A. Preview of Proposed System

The architectural design of the proposed system depicts the modules of the system and how they are integrated in Fig. 3.

The perception layer: Contains the primary dataset source from the remote camera feed, and the secondary dataset source from the ImageNet. The primary dataset goes into the Edge Layer via the Transport Layer for Remote Monitoring. While the secondary dataset is fed into the Preprocessing Module for Data Augmentation.

Preprocessing Module : The size of the dataset for this system was augmented to 224 X 224 pixels using the Pillow Library in Python. Afterwards, anchor boxes are used to label the data for validation/splitting. Following the preprocessing module, the dataset was validated/splitted using leave-one-out cross validation,

Data Splitting: A cross validation technique that minuses the errors associated with k-fold. The Dataset was splitted into 50% Training, 25% validation and 25% for testing. Following the Dataset Splitting is the MobileNet V2 Training Module.

MobileNet V2 Training Module : Training takes place in this module with the Teachable Machines tool. This tool leverages the Mobilenet V2 algorithm. MobileNet V2 algorithm is a pre-training deep learning that splits convolution into depthwise convolution and pointwise convolution. The 224 X 224-pixel standard image is fed into the depthswise convolution block. This process applies single convolutional filter for each input channel. Following this block, is the process of Rectified Linear Unit (ReLu) Function that output the input directly, if its positive, otherwise zero. The process continues with the SoftMax Activation Function for classification task.

The Edge Layer : After training is completed, the trained model is exported into the edge layer for integration with the IoT components. Following the edge computing, the model syncronises with the application layer.

The Application Layer : The Blynk IoT app resides in this layer and is configured with farm parameters for the farmer to remotely monitor the farm.

Energy Harvesting Module: Furthermore, the edge layer is powered by an energy harvesting tool to ensure connectivity.

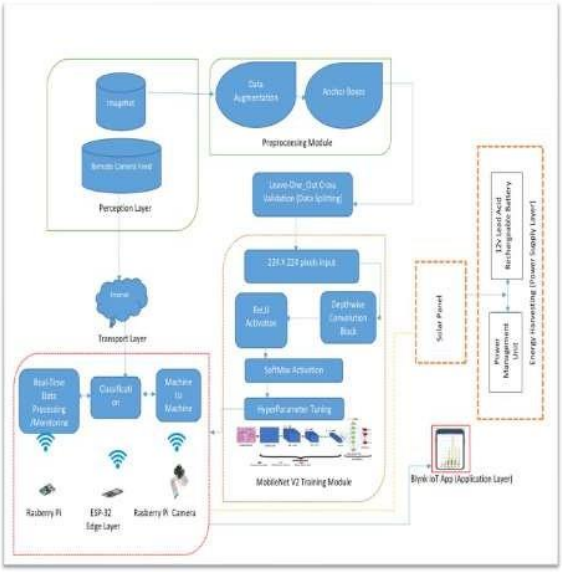

Fig. 3 Architecture of the Proposed System

B. Edge Device Hardware Main Processor (Raspberry Pi 4)

This is the 2nd device where the DL framework was deployed. It is main sub controlling unit of the entire system as most of the intensive processing or computing activity of the system is done here. The choice of Raspberry Pi 4 as the main processor is because its powerful computing capability as most available microcontroller models might not be able to handle the huge processing power required. this implies that data processing and image identification algorithms will be carried out on the hardware before results of processing are uploaded online for the end users view through a mobile phone app. This will improve the latency time of response of the system incase there are network issues.

Raspberry Pi Camera

This camera unit is needed for capturing of clear images required for training the deep learning model. In this project, the Raspberry Pi camera v2 will be used. It uses an 8-megapixel camera using Sony IMX219image sensor and can produce 1080p, 720 and 640x480 video. The camera board is shipped with a flexible flat cable that plugs into the CSI connector onthe Raspberry Pi.

C. Software Components

1. Operating System – A lightweight operating system suitable for edge devices, such as Raspbian for Raspberry Pi or a custom OS for specialized devices was installed.
2. Deep Learning Framework – TensorFlow 2.5.0, Python 3.7 are deployed on theRasberry Pi 4 device
3. Dataset – ImageNet and Live Data from the field
4. Image Processing –Mobilenet V2 pretrained algorithms are deployed to preprocess, label and classify data.
5. Training – Teachable Machines from Googleare deployed on the Rasberry Pi 4 machine totrain the data.
6. Classification – Mobilenet V2 was used for classification.
7. Database – My SQL Engine was integrated with the Teachable Machines.

D. Algorithm of the Proposed System

Step 1: Initialise Rasberry Pi 4 and Load Object Detection Model
Step 2: Optimise the hyperparameters of the SSD MobileNet model with Teachable Machines
Step 2.1: Name classes on Teachable Machines
Step 2.2: Upload Data on Teachables Machines
Step 2.3: Train model on Teachable machines
Step 3: Initialise Rasberry Pi 4 camera and set it to stream video frames
Step 4: Initialise CWT Sensor and set it to capture soil nutrients
Step 5: Start the Video stream from the Pi camera

Step 6: for each frame
Step 7: Pass the frame through the Object Detection Model
Step 8: Obtain the corresponding output from the video frame i.e class of the detected object(Cylas Puncticolis, Cylas Bruneus, SPFMV, SPCSV, White flies)
Step 9: Obtain the corresponding output from the sensor i.e the soil nutrients data (Good or bad)
Step 10: Generate a message from the detected class and soil nutrient
Step 11: Pass it to the Blynk App.

E. Data Gathering and Classification

The image of the insect was collected from the Pest lure / pheromone deployed. Other images of Pest and Diseases are collected from the Rasberry Pi 4 camera capture. Offline images are collected from ImageNet Dataset. Features of the Images from Live Capture and ImageNet Dataset include:

1. Name
2. Colour
3. Height
4. Weight

Soil parameters were collected from the Continuous Wavelength Transformation (CWT) Soil Sensors, A total 5 classes of images created for this model include:

1. Healthy_Potato Leaves
2. SPFMV_Disease
3. Cylas_Puncticollis
4. SPCSV_Disease
5. Yellow Potato Leaves

After training is done, the model was exported to the Rasberry Pi 4 device to identify and classify data based on the parameters that has been set.

F. Field Testing

The field testing was achieved in the field which consisted of three farm fields. The camera mounted on the hardware of the model was positioned to focus on the insect trap and potato leaves, the camera is mounted 90 degrees horizontally and connected to the internet via a 2.4 GHz home Wi-Fi network on the Rasberry Pi device embedded in the hardware. Furthermore, the trained model was set up on a Rasberry Pi (a credit-card-sized-microcomputer) which handles the processing layer duty, in order to complete the pest and disease identification task. The purpose of the experiment was to track the pests foundon the trap sheets for a continuous period of 3months

Fig. 4. Field Mapping of the Pheromones / StickyTrap

Fig. 5. Replicate of the Sweet Potato Field

Fig. 6. Power on the System (using the black switchon the hardware)

IV. Results and Discussion

In Table 1., The existing system had a computational cost of training for 9hours, 30minutes, while the proposed system training lasted for 5 hours, 30minutes, with 4 frames per second of camera capture. This implied that the proposed system saved 4 hours out of the Training and increased the latency rate of the model for remote monitoring.

Table 1. Computational Cost of Training on the Edge Device

Algorithm	Computational Cost for Training (Hours: Minutes)	Computational Cost for Real-Time Detection (Frames per second)
Faster RCNN ResNet	9:30	30
Proposed	5:30	4

Time	nitrogen	phosphorus	potassium	ph	moisture	leaf_color	pest	prediction	recommendation
7:35:49:am	26.53	25.74	27.39	6.46	24	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:50:am	26.29	25.5	27.16	6.46	23.75	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:50:am	26.43	25.65	27.3	6.46	23.9	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:51:am	26.24	25.45	27.12	6.45	23.71	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:51:am	25.96	25.16	26.84	6.45	23.41	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:52:am	25.91	25.11	26.79	6.45	23.36	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:52:am	26.53	25.74	27.39	6.46	24	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:53:am	26	25.21	26.89	6.45	23.46	Green	Whiteflies	SPCFV Disease	Apply pesticide
7:35:53:am	26.19	25.4	27.07	6.45	23.66	Green	Whiteflies	SPCFV Disease	Apply pesticide

Table 2. Real-Time Detection Results from

Table 2 gives a detailed report of the remote camera feed, sent to the model for classification, and sensor readings feed into the model. For each of the plots, every activity is recorded and stored per minute on the Rasberry Pi Device. Plot1, the first remote camera feed happened at 7:35:49s displaying data for Nitrogen as 26.53. This information confirms that Nitrogen nutrient in the soil is good. The Potassium, Phosphorus, Ph and moisture readings were 25.74, 27.39 respectively, 6.46, 24. This implied that the potassium and phosphorus nutrients in the soil is good, while the pH data implies acidity, moisture of the soil is also good. The leaves are green, but there is the presence of SPCFV disease, and the farmer is advised to apply pesticide as a pro-active measure. Furthermore, the next remote monitoring activity at 7:35:50s implies a low latency rate of 1s, indicating real-time classification. This instance of capture recorded captured data for Nitrogen as 26.29, Potassium as 25.5, Phosphorus as 27.16 ,Ph Acidity at 6.46 and Soil moisture gained more moisture at 23.75 due to little drops of rain, while the SPCFV disease was still present for the farmer to apply the recommendation of Applying Pesticide. This implied that for every second of remote monitoring, the soil nutrients either improves or declines. This indicated that the model has provided the real-time capacity for the farmer to be engaged.

V. Conclusion

The hybridized Internet of Things (IoT) and Deep Learning architecture for the management of sweet potato pests and diseases has yielded promising results, indicating its potential to revolutionize agricultural practices. The system demonstrated rapid response times and accurate decision-making logic, leading to timely and effective actuation in response to identified pest and disease issues. Contributing to increased sweet potato yield, cost savings, and a reduction in environmental impact through decreased pesticide usage, highlighting its economic and ecological benefits.

References

[1]. Angel, C., Mercedes, A., & Fuentes, L. (2022). Supporting IoT applications deployment on edge-based infrastructures using multi-layer feature models. *Journal of Systems and Software, 183*(111086), 1-14.

[2]. Cedric, M.-D., Carlos, C., Vicente, J., Jaime, R., Javier, P., & and Ramón, J. D. (2023). Robust Multi- Sensor Consensus Plant Disease Detection Using the Choquet Integral. *National Library of Medicine, 23*(5).

[3]. Jonathan, O., Estefania, T., Daniel, O., Dan, T., Emmanuel, O., & and Godliver, O. (2023). A field-based recommender system for crop disease detection using machine learning. *Frontiers in Artificial Intelligence, 6*, 1-8.

[4]. Raj, T. (2017). Fueling The IOT Revolution. NetworkWorld.

[5]. Qianlin, L., Prashant, S., & David, I. (2020). AI on the Edge: Characterizing AI-based IoT Applications Using Specialized Edge Architectures. *Distributed, Parallel and Cluster Computing*.

[6]. Sarveshbhatt. (2019). *ITTechBook*. Retrieved August 29th, 2023, from https://techtalkwithbhatt.com/2019/06/02/four-layer-and-seven-layer-architecture-in-the-internet-of-things/

[7]. Wang, M., Fu, B., Fan, J., Wang, Y., Liankuan, Z., & Xia, C. (2022). Sweet potato leaf detection in a natural scene based on faster R-CNN with a visual attention mechanism and DIoU-NMS . *Ecological Informatics, 73*, 1-12.