Website: www.aceiot.ur.ac.rw
Mail: aceiot@ur.ac.rw

COLLEGE OF SCIENCE AND TECHNOLOGY

AFRICAN CENTER OF EXCELLENCE IN INTERNET OF THINGS

IoT and ML-Based Precision Agriculture: A Case of Rwanda Coffee

A dissertation submitted in partial fulfilment of the requirements for the award of masters of science degree in internet of things: Embedded computing system

Submitted By:

Meron Alemnew Kifle (Reg. No: 219013928)

November, 2022

UNIVERSITY of RWANDA

COLLEGE OF SCIENCE AND TECHNOLOGY

ACEIoT

AFRICAN CENTER OF EXCELLENCE IN INTERNET OF THINGS

IoT and ML-Based Precision Agriculture: A case of Rwanda Coffee

A dissertation submitted in partial fulfilment of the requirements for the award of masters of science degree in internet of things: Embedded computing system

Submitted By

Meron Alemnew Kifle (Reg. No: 219013928)

Supervised by:

Dr. Alexander Ngenzi

Dr. Frederic Nzanywayingoma

November, 2022

Student Declaration

I declare that this Dissertation contains my own work except where specifically acknowledged.

Meron Alemnew
219013928

Signature: ....

9th Nov 2022
Date: ……………………….........................

Bonafede Certificate

This is to certify that the project entitled “IoT and ML Based Precision Agriculture System A Case of Rwanda” is a record of original work done by Meron Alemnew Kifle with registration number 219013928 in partial fulfilment of the requirement for the award of masters of sciences in Internet of Things in College of Science and Technology, University of Rwanda, Academic year2018/2019.

This work has been submitted under the guidance of Dr. Alexander Ngenzi and Dr. Frederic Nzanywayingoma.

Dr. Alexander Ngenzi

Signature: ....

09/11/2022

Date: ....

Dr. Frederic Nzanywayingoma.

Signature: ……………………

09/11/2022

Date: …………………………

The Head of Masters and Trainings

Dr. James Rwigema

Signature: ……………………

ACKNOWLEDGEMENTS

My deepest gratitude goes to the Almighty God for making me strong, and keep me in good health through this research work in such difficult situation of my life. I would like to thank my supervisors Dr. Alexander Ngenzi and Dr. Frederic Nzanywayingoma, for their endless support, encouragement and for being so patient with me through all of it. I would like to thank ACE IoT center management team for their timely support and my fellow master's students who were there for me when I needed clarifications on my project. I would like to thank my parents and siblings for their endless support during my study.

Finally, I would like to thank my roommates Kareema and Denyse, who supported me during the difficult situation of my life (the pandemic and the war in Tigray).

.

Abstract

Coffee is grown in more than 50 countries in the world. And it is consumed worldwide in day-to-day life. In Rwanda it holds a unique position in the economy by making approximately 27 percent of total export revenue. Thousands relay on it as a livelihood. But its market share has not yet been fully developed because of several factors such as diseases, pests, insects, and limited use of advanced technologies. Coffee leaf rust (CLR) a coffee disease caused by a fungus called Hemileia Vastatrix is the most devastating one in Rwanda. It causes up to 50% leaf loss and up to 70% berry loss. This research is intended to develop an IoT and Machine learning based disease detection mechanism to monitor coffee leaf rust at early stage. A Pi camera sensor i are deployed to collect real-time data, a raspberry pi is configured to send data to google cloud platform and firebase for real-time data storage, analysis, visualization, and a web-based application is developed using FastAPI for user to access. A deep learning model is trained using ResNext algorithm, which performs 91% accuracy and it's deployed in google cloud platform. The methodology used in this study is excremental where prototype is developed to collect data and secondary data from existing studies. The expected results from the study are a customized dashboard showing real time values of variables collected using prototype developed in a graph and in application.

Keywords: IoT, Machine learning, Raspberry pi, Pi camera, ResNext, Google Cloud Platform, Firebase, FastAPI, Disease Detection

Abbreviations

IoT: Internet of Things

WHO: World Health Organization

ML: Machine Learning

CSI: Camera Serial Interface

API: Application Programming Interface

GB: Gege Bite

OS: Operating System

VNC: Virtual Network Computing

GDP: Gross Domestic Product

RAB: Rwanda Agriculture Board

AI: Artificial Intelligence

SVM: Support Vector Machine

ELM: Extreme Learning Machine

KNN: K-nearest neighbor

YOLO: you only look once

CNN: Convolutional Neural Network

SSD: Single Shot Multibox Detector

RFCN: Region-based Fully Convolutional Network

CAE: Convolutional Autoencoder

UAV: Unmanned Aerial Vehicle

LAN: Local Area Network

CPU: Central Processing Unit

BCM: Broadcom

SDHC: Secure Digital High Capacity

SDXC: Secure Digital EXtended Capacity

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Background

Coffee is the most important and valuable agricultural commodity and beverage cultivated and marketed throughout the world and worth up to US $ 14 billion annually for producing country [1]. More than 50 developing countries are earning 25 % of their foreign exchange from coffee [2]. Coffee was introduced into Rwanda by German missionaries in 1904 and was cultivated mainly for the colonial administration [3]. Production gradually increased over time. Coffee is now one of the most popular cash crops grown in Rwanda.

Owing to Rwandan soil characteristics, which are suitable for coffee production, coffee plantations are found in all four provinces of the country [3]. Arabica and Robusta are the major coffee species cultivated in Rwanda. Arabica coffee occupies approximately 95% of the country's total coffee plantations and is mostly planted at higher altitudes in the Southern, Northern, and Western Provinces of Rwanda (Figure 1.1). Robusta coffee comprises the remaining 5% and is planted at lower altitudes below 1400 m in the Eastern Province [3].

Coffee production is a critical source of income for hundreds of thousands of Rwandan families. Coffee is grown by some 400,000 small-scale farmers and their families, most of whom own less than quarter a hectare of land each (total area under production hovers at 42,000 hectares). It holds a unique position in its economy by making approximately 27 percent of total export revenue [4]. It is the second most important agricultural export commodity after tea earning in the country having over 15% of the foreign currency annually and it contributes 6.1% to national GDP [5]. Despite the importance of the crop to the economy, its productivity is still low. Fig. 1.3 shows the current coffee production figures in Rwanda.

Even though it has a significant impact in the Rwandan economy, the coffee production sector is facing several challenges, mainly low productivity. Coffee suffers yield losses from several factors including insect pests and diseases. Among those, coffee leaf rust (CLR) caused by a fungus called Hemileia Vastatrix is the most devastating one. Such low yields coupled with highly regulated prices have left farmers running into losses in coffee business. Low coffee yields and poor return on crop investment could be attributed to climatic, edaphic, biotic and socio-economic related constrains [6].

According to Rwanda agriculture board (RAB), the coffee varieties that are widely grown in Rwanda, BM 139, and Jackson, are known for their excellent cup quality attributes and both are susceptible to coffee leaf rust (CLR) of Hemileia Vastatrix, the most serious of devastating coffee disease in Rwanda. Coffee leaf rust reduces the value of coffee, market price available to farmers, and volume purchased by international buyers. CLR is a major disease of coffee plants caused by the fungus Hemileia Vastatrix .

It is characterized by Yellow-orange powdery spots on underside of leaves corresponding yellow-white patches on upper surface of leaf. It causes a major adverse economic effect and has been reported in over fifty countries. Coffee leaf rust infestation on a farm causes up to 50% leaf loss and up to 70% berry loss [7].

A survey was conducted in Rwanda to evaluate incidence and severity of CLR and other coffee pests and diseases. Results showed that all provinces were affected by CLR with the highest severity in Eastern province where the incidence was up to 100%.

However, Accurate, and timely detection can help farmers in applying timely treatment on the plants and thereby can reduce the economic losses substantially. The most practiced control methods are manual. Around 98% of coffee farmers uses pesticides and fungicide to control coffee pests, and diseases. Currently, the use of the pesticides in Rwandan agriculture is very limited and primarily applied only to some cash crops, in particular coffee, with some other economically profitable crops like potato and tomato [8].

In preventing and exterminating coffee leaf rust, development of different chemicals containing local-grown raw materials, such as pyrethrum, has recently been tested. Also, campaigns for farmers for prevention of pests and diseases are held in May and October each year.

Applying those chemicals are not affordable by all farmers it is also time consuming. Apart from that, those chemicals have a potential impact on environmental pollution. According to WHO Pesticides and fungicides are among the leading causes of death by self-poisoning, in low- and middle-income countries.

The reason why applying those chemicals cause those potential impacts are it is because the information about those species is not provided in a timely and accurate way to the farmers. Besides, if the information is provided accurately and in a timely manner and if farmers can access the information.

It will be easy to take possible measures at early stage. Traditionally, the information about pest species and diseases are acquired mainly through the visual judgment of humans[9]. Using this method, workers compare the pest's shape, color, texture, and other characteristics. Which is time consuming, labor intensive and error-prone.

Therefore, it's clear that we must emerge new technologies to mitigate the challenges in agricultural productivity. Precision agriculture is an approach that employs information technology to use data from a variety of data sources to integrate crop production management decisions[10].

IoT and machine learning both feed into each other and create an ecosystem of automation, IoT device collect data on millions of criteria, which is then collected in the cloud, used to train and improve AI algorithms[11]. ML algorithms in conjunction with sensors can lead to accurate detection of deceases with low cost and with no environmental issues and side effects[12]. Recently different methodologies have been used for pest, weed and daises identification and monitoring, which employ image processing and complex algorithms for detection and classification. The existing approaches have covered different areas in improving crop production but they are limited in emerging both IoT and deep learning technics specially in coffee production.

This study proposes an IoT and deep learning-based approach to detect CLR in coffee plants at early stage by collecting real time data using a deployed pi camera sensor and transmitting the data to the cloud for analysis and quick response. A web-based application is also integrated. The work will give a greater understanding on how Deep learning and IoT can emerge and give such accurate results. This research will give the background information on coffee production in Rwanda with statistics, brief description of related work, methodology used, system analysis and design, data collection and analysis, model training, validation, testing, and deployment, conclusion, and future recommendations.

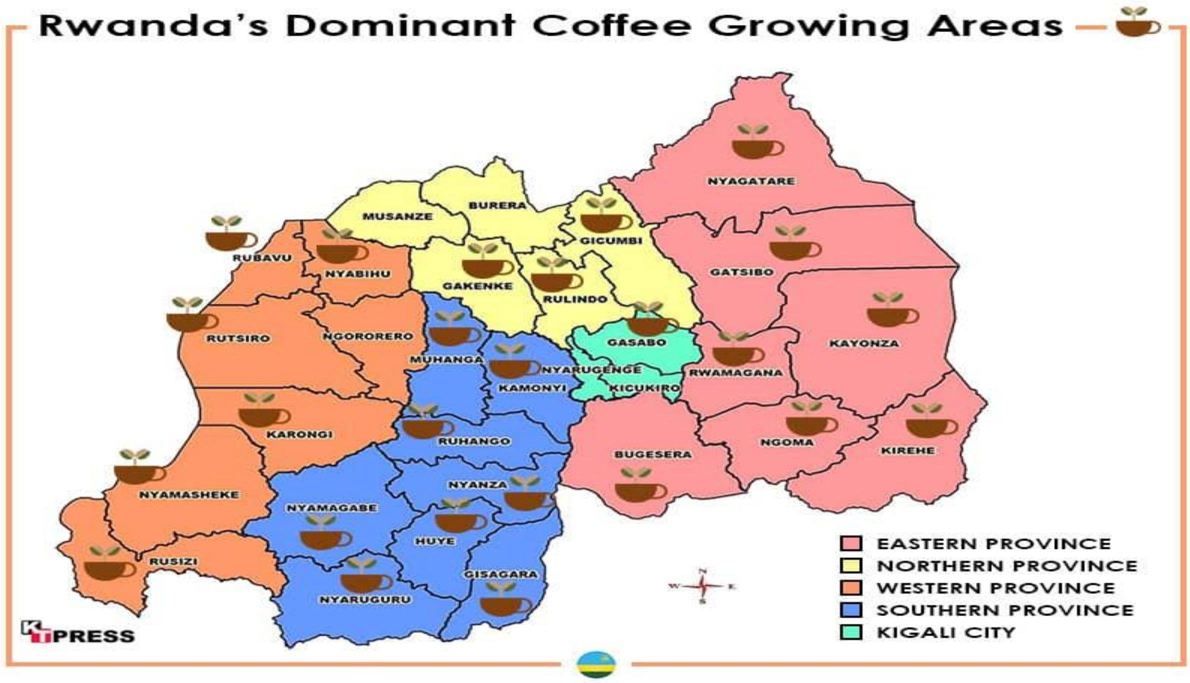

Figure 1.1: Coffee growing areas in Rwanda (Ktpress)

DATE	VALUE	CHANGE, %
2019	29,366	-24.01 %
2018	38,643	21.83 %
2017	31,718	31.50 %
2016	24,120	10.60 %
2015	21,808	33.15 %
2014	16,379	-10.72 %
2013	18,346	-8.25 %
2012	19,995	-8.36 %
2011	21,820	12.95 %
2010	19,319	-0.27 %
2009	19,372	-6.52 %

Figure 1.2: ***Rwanda - Green coffee production value change (United States Department of Agriculture)***

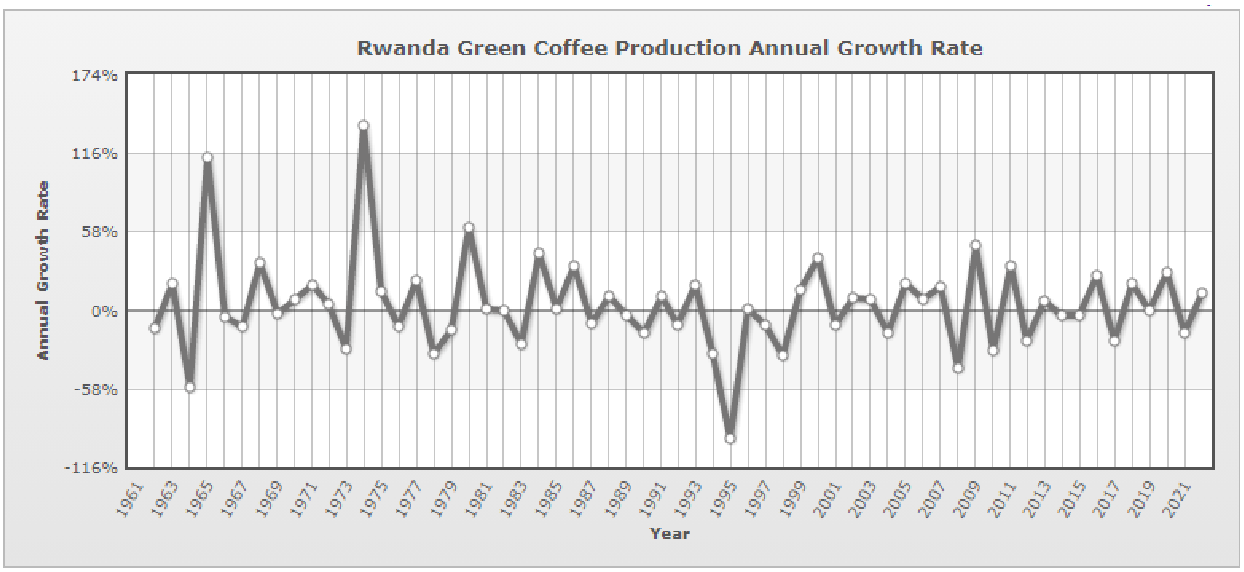

Figure 1.2: Rwanda Green Coffee Production by Year (United States Department of Agriculture)

1.1 Problem Statement

Coffee production is the main economic activity for smallholder farmers in Rwanda and it is the second agricultural commodity export after tea having over 15% of the foreign currency annually [13].

However, Coffee suffers yield losses from several factors, mostly coffee leaf Rust. Coffee leaf rust (CLR) caused by a fungus called Hemileia vastatrix is the most devastating factor for coffee yield production which affects the leaf. Coffee leaf rust is characterized by Yellow-orange powdery spots on underside of leaves corresponding yellow-white patches on upper surface of leaf. It causes a major adverse economic effect and has been reported in over fifty countries. Coffee leaf rust infestation on a farm causes up to 50% leaf loss and up to 70% berry loss [7].

The specialty coffee industry realizes now that Rwandan coffees will continue to be infected with this defect unless remedial actions are taken [14]. Different efforts have been undertaken by the government of Rwanda to mitigate this problem. As confirmed by farmers, the delay in fungicides, coupled with insufficient equipment for spraying, causes the fugus to migrate from one farm to the neighboring one until the spray tour is finish [5].

I believe that if the information is provided accurately and in a timely manner. It would be easy to take possible measures to mitigate the loose. Recently different methodologies have been used for pest, weed and daises identification and monitoring, which employ image processing and complex algorithms for detection and classification.

Few works [15] [16] [17] attempting to address using image processing technology, and IoT but we need to emerge different technologies for accurate, effective efficient and real time results which is basically an internet of things state of art. This research designed and implemented a

Precision Agriculture approach that employs information technology to use data from a variety of data sources to integrate crop production management decisions [10].

The system can identify and examine the results automatically and show the status in real-time. The proposed solution will help farmers, authorities in agriculture, and the economy of the country in producing and exporting high quality production coffee.

1.2 Objectives

The general Objective of this research is to develop an IOT and Deep learning based real-time system, which monitors coffee crop production.

The specific Objectives are:

1. To design, assemble, and configure IoT hardware devices
2. To develop a deep learning model
3. Evaluate the performance of the model
4. Deploy the model in cloud and create a cloud-based data base
5. To design and implement a web-based application for farmers to access

1.3 Hypotheses

1. How harmful are coffee diseases and what are the benefits of using the internet of things and deep learning approach to mitigate this and increase productivity?

2. How can current technology be used to determine states of coffee crop production to increase the productivity?
3. How to collect real time data?
   3.1 How the collected coffee leaf data will be analyzed and visualized?
   3.2 Which deep learning algorithm best fits analyze the collected data?

4 How to deploy the trained model?

4.1 How can user be able to access the result or the predicted output?

1.4 Study Scope

This research focuses on emerging IoT and ML techniques to monitor coffee yield production. The study consists using sensors integrated with other hardware and application platforms to monitor coffee leaves and detect coffee diseases a case of Rwanda.

1.5 Significance of the Study

1. This research will contribute to the existing pool of knowledge, many research has been done but there are limitations on emerging both IoT and deep learning technologies for better accuracy and efficiency. This study will be focusing on training a model of coffee leaves, to determine the states and give real-time results. This will help farmers and give researchers in academia sectors access to the model and to the state of the art. It would also benefit people in the agricultural industry most importantly Agri-tech.
2. The developed system will enable farmers to take possible actions on time based on the results shown.
3. It would also help in productivity and exporting quality coffee, which results in increasing the foreign exchange of the country.
4. The system will contribute in decreasing environmental pollution by reducing the use of chemicals.

1.6 Motivation of the Study

As coffee is the top export commodity and more than 400000 people depend on it for their survival. While facing growing challenges. Mostly due to diseases. And it affects the individuals and the country in general. Therefore, it is important to implement an integrated system to enhance the productivity.

1.7 Organization of the Study

This document is organized as follows: Chapter one introduction, comprises of background, problem statement, objectives, research questions, significance and scope of the study; Chapter 2 gives the breakdown of related work; Chapter 3 Methodology, presents the general introduction, descriptions of components and platform used; Chapter4 Implementation, comprises experiment setup, circuit diagram of prototype, configuration and deployment of the prototype; Chapter 5 Results and discussion, presents the analysis of collected data, visualization and interpretation of results; and Chapter 6 concludes the document by presenting the recommendations drawn from this study.

Chapter 2: Literature Review

A greater research and development capacity is needed to fuel the necessary advances in science and technology to overcome the challenges in agriculture to sustain production. Agricultural research is one of the main factors contributing to shifts in agricultural production systems and changes in the rural world. It is helping in improve productivity by integrating technology and agricultural practices. Various impact assessments have shown that it is one of the most effective investments when it comes to increasing agricultural production.

Many researchers developed algorithms for Segmentation, feature extraction, representation, and classification to detect crop diseases. Below are a handful of works related to crop disease detection and precision agriculture systems.

This section reports on the ones more related to my research. Those works can be grouped into three categories: Machine learning based (Section 2.1), IoT based crop disease detection and monitoring systems (Section 2.2) and Coffee Production statics (Section 2.3).

2.1 Machine Learning based Crop Disease Detection Systems

In [18] Plant disease and pest detection using deep learning-based features are developed. In this study, the performance of nine powerful (AlexNet, VGG16, VGG19, SqueezeNet, GoogleNet, Inceptionv3, InceptionResNetV2, ResNet50, ResNet101) architectures of deep neural networks were evaluated for plant disease detection. A dataset consisting of 1965 real plant disease and pest images within eight clusters were used.

The accuracy, sensitivity, specificity, and F1-score are all calculated for performance evaluation. The evaluation results show that deep feature extraction and SVM/ELM classification produced better results than transfer learning.

in [19] a Convolutional Neural Network (CNN) with the transfer learning approach is proposed for large volume of data and higher accuracy. Data processing techniques such as resize, crop and zero centralizing are used to improve data learning efficiency. The CNN Structure proposed in this paper has size image as input data and 4 outputs according to the periodontal state. It also uses momentum optimization technique for neural network optimization. It processes the images of all the infected parts of the croplike upper and lower side of the leaf, stem, root, and fruit images.

In [20] Support Vector Machine (SVM) is used for classification.tImages of leaves from the crop fields were taken and then various processing techniques were applied. First, the image is converted to a gray scale image. Then the pixels subtracted and calculated from the original image after calculating the background of the image. It is adjusted to improve the image quality. Binary image is used to detect the white fly from the image

In [21] a crop disease detection mechanism using a real-time object detection algorithm called YOLO (you only look once) was proposed. The model was trained under A public dataset containing 54,306 images of healthy and diseased leaves. A deep convolutional neural network is trained to detect 26 diseases of 14 several crops. To check the feasibility of this method, a trained model is tested on the test set. The trained model achieved an accuracy of 99.35%. On the other hand, when the model was tested on images from the internet, the model still achieved an accuracy of 31.4%. This accuracy is much better than on a random image set (2.6%).

In [22] a prediction is done using Crop Disease Dataset which is available publicly on Kaggle. Image preprocessing was done for noise reduction and edge sharpening. This makes the manual process of disease detection automatic or semi-automatic. CNN is applied for extracting features from images. A decision tree is used to reflect decisions and decision making in a visual and explicit way. Then the features extracted from the image are compared with the resultant database to detect whether the disease is present or not.

[23] in this paper a combination of convolutional neural networks and autoencoder is designed for a hybrid approach of detection of crop leaf diseases. The hybridized deep learning neural network is named as convolutional encoder network. 900-image dataset is used both for training and testing data set. In the encoding part, we take an input image and generate a high-dimensional feature vector; then, the features are aggregated at multiple levels. The proposed architecture of

convolutional encoder network consists of convolution encoder layers, max-pooling layers, and fully connected layers.

The activation function is applied internally in convolution layer. And Adam optimizer has been used to increase the accuracy and reduce the loss while training.

In [24] a predictive model using CNN for classification and prediction of disease in paddy crop were proposed. The model processes the image followed by image augmentation and then trains on it. Then with the help of the trained images, it predicts what kind of disease the given plant leaf might be suffering from. Python programming language and Keras library tool were used to create the model. The model gets an image as its input in the form of matrices. In a CNN model, matrices play a key role. In the convolution step, a filter or kernel matrix is taken and convolution is performed with the input image matrix by sliding the filter over the input image.

2.2 IOT based Crop Disease Detection Mechanism

In [25] a total of 1426 images of rice diseases and pests from paddy fields were collected in real-life scenarios and classified into more than eight classes. Three different training methods have been implemented on two state-of-the-art large CNN architectures and three state-of-the-art small CNN architectures (targeted towards mobile applications) on the rice dataset A new concept of two-stage training derived from the concept of fine-tuning has been introduced which enables the proposed Simple CNN architecture of this work to perform well in real life scenario.

In [26] a robust deep-learning-based detector for real-time tomato diseases and pests recognition was proposed. Using images captured by different cameras in place by camera devices with various resolutions. The dataset was trained in three main families of detectors: Faster Region-based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN), and Single Shot Multibox Detector (SSD), which for the purpose of this work are called "deep learning meta-architectures". Each of these combined meta-architectures with "deep feature extractors" such as VGG net and Residual Network (ResNet) for local and global class annotation and data augmentation to increase the accuracy and reduce the number of false positives during training.

[27] proposes a novel hybrid model based on Convolutional Autoencoder (CAE) network and Convolutional Neural Network (CNN) for automatic plant disease detection. In this work, the proposed hybrid model is applied to detect Bacterial Spot disease present in peach plants using their leaf images. The experiments performed in this paper use a publicly available dataset named Plant Village to get the leaf images of peach plants. The proposed system used 9,914 training parameters. This, in turn, significantly decreases the time required to train the model for automatic plant disease detection and the time required to identify the disease in plants using the trained model.

The dataset contains 4457 leaf images of peach plants, which are evenly distributed in two classes: healthy and diseased (Bacterial Spot). The healthy class contains 2160 peach leaf images, and the diseased (Bacterial Spot) class comprise of 2297 leaf images of the peach plant.

[28], in this paper Quadcopter(drone) is used to fly above the field for identifying the pests and their density, the quadcopter is with Raspberry pi along with a camera attached at the bottom, that microprocessor will be used for the identifying the pest and storing all information on the cloud. Using the camera big pest like mouse, snake, mongoose, spider, was successfully identified. In identifying small pests like Stem-borer, green leafhopper, Hispa, Mealy Bug.
All the living being have some amount of heat in their body which can be identified using a thermal camera and attaching a camera on the Quadcopter resulted in identifying all the available pest in all over the field by just flying the Quadcopter over the field. And support victor machine was used in clustering.

In [17], an Internet of Things (IoT) assisted Unmanned Aerial Vehicle (UAV) based rice pest detection model using Imagga cloud is proposed to identify the pests in the rice during its production in the field. The IoT assisted UAV focuses on artificial intelligence (AI) mechanism and Python programming paradigm for sending the rice pest images to the Imagga cloud and providing the pest information. The Imagga cloud detects the pest by finding the confidence values with the tags. The tag represents the object in that image. The tag with maximum confidence value and beyond threshold is selected as the target tag to identify the pest. If pest is detected then the information is sent to the owner for further actions.

[16] deployed different sensors in the farm, like soil moisture sensor, Temperature Humidity sensor and camera for detecting diseases on a leaf. Data collected from sensors and send it to Raspberry PI through wired or wireless devices. In server-side data is verified and matched with ideal values of data like temperature value, humidity value, and soil moisture value.

If difference occurred with respect to predefined threshold value, then notification send to the farmer on his mobile or website.

In [15], raspberry PI was used to detect and prevent plant disease from spreading. The k means clustering algorithm was used for image analysis. It has numerous focal points for use in vast harvest ranches and in this way distinguishes indications of sickness naturally at whatever point they show up on plant leaves. In pharmaceutical research, the recognition of leaf ailment is essential and a critical theme for research, because it has the advantages of monitoring crops in the field in the form and thus automatically detects symptoms of disease by image processing using an algorithm clustering k - means.

[29] proposes K-means clustering to identify the affected crops and provide remedial measures to the agricultural industry. Using k-mean clustering algorithm, the infected region of the leaf is segmented and analyzed. The images are fed to an application for the identification of diseases.

TABLE 2.1 SUMMARY OF THE LITERATURE REVIEWED AND THEIR GAPS

S/N	AUTHORS	Description and Technique applied	Accomplishment	Gaps and Remarks
1	Muammer et al. [18]	The performance of nine (AlexNet, VGG16, VGG19, SqueezeNet, Google Net, Inceptionv3, InceptionResNetV2, ResNet50, ResNet101) architectures of deep neural networks were evaluated for plant disease detection.	Calculated performance evaluation.	Evaluation was in a limited amount and not specific.
2	Dineshkumar B et al. [19]	A Convolutional Neural Network (CNN) with the transfer learning approach was applied for large volume of data to detect disease and pest.	Field conditioned images for both disease and pest detection has been combined	Accuracy is not specified and discussed
3	Danish Gondal et al. [20]	Image features are calculated for the acquired images. Gray level co-occurrence matrix (GLCM) is created for images and then features from that GLCM matrix are calculated.	An automatic approach for early pest detection	Weak classification approach

4	IJSREM [21]	YOLO algorithm is applied for fast crop disease detection of real-time images.	A single network is used for prediction.	Not an effective algorithm for disease detection
5	Nikita Kasar et al. [22]	Decision Tree and Random Forest ML algorithms are applied for disease detection	image segmentation is done on image in which visual image is divided into segments to simplify image analysis	Weak detection and training approach
6	Khamparia, Aditya Saini et al. [23]	Convolutional encoder net- works has applied on leaf images to detect crop disease.	designed a hybrid approach a combination of convolutional neural networks and autoencoders.	Complex and many architectural layers.
7	Sharma Ritesh et al. [24]	predictive model using CNN for classification and prediction of disease in paddy crop	Classification.	Not accurate and not suitable to apply it in other crops
8	Rahman, Chowdhury R et al. [25]	Keras framework, MobileNet, NasNet Mobile and SqueezeNet architectures were applied to detect crop pest and disease.	Small scale architecture	Not effective to detect both pest and disease.
9	Fuentes, Alvaro et al. [26]	Faster Region-based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN), and Single Shot Multibox Detector (SSD), were applied to detect pest and diseases to images in different camera resolutions	deep learning meta-architectures	Weak image pre processing because of the different camera resolutions.

10	Bedi, Punam et al. [27]	hybrid model based on Convolutional Autoencoder (CAE) network and Convolutional Neural Network (CNN) for automatic plant disease detection.	Aromatic classification.	Complex Architecture.
11	[28]	Quadcopter (drone) is used for image acquisition and support vector machine is applied for detecting pests in paddy plant	classifying the pest available in a field by using a microprocessor along with infrared camera and normal camera.	Not Suable in different locations. Expensive.
12	Bhoi, Sourav Kumar et al. [17]	UAV, 4G communication, Python programing is used to detect pests in rice field	Automation	Poor mechanism of communication to cloud, No Wi-Fi or LORA.
13	[16]	Various Sensors are deployed in farms using Raspberry pi to detect diseases.	Automation	No ML algorithm applied for Accuracy
14	Gandu, Kavya et al. [15]	Raspberry PI is used as a processer to detect plant disease using k means clustering.	Automation	Poor accuracy
15	Divya, Gadde et al. [29]	Raspberry pi is used in paddy plants to detect disease by applying Neural Network, Homogeneity, Standard Deviation, in MATLAB.	Automation	Poor ML training.

The summary of related work in Table 2.1 shows the existing work reviewed and the gaps for each study. Unlike the existing studies which focuses on IoT based, Machine learning based disease detection mechanisms. My work IoT and ML based precision agriculture focuses on emerging both IoT and Machine learning technics for accurate, efficient, and cost-effective mechanism to detect coffee disease. They are not that much done to increase coffee productivity in Rwanda apart from very few works which focuses on none technological aspect of it. Therefore, the authorities in charge of Agriculture and technology needs to upgrade the current system to increase coffee productivity.

2.3 Coffee Production statics in Rwanda

In [14], a survey was conducted in Rwanda to evaluate incidence and severity of CLR and other coffee pests and diseases and to determine how the crop management contributes to CLR severity.

In the survey random sample of 307 coffee farms was visited and the prevalence, incidence and severity were recorded. Results showed that all provinces were affected by CLR with the highest severity in Eastern province where the incidence was up to 100%. Results shown that all commercial cultivars were susceptible to CLR and most management practices such as mulching, pruning and fertilizer application were associated with lower levels of CLR severity except intercropping which resulted in higher disease intensity

In [30], Discuss the impact of temperature and rainfall variability and its effect on coffee diseased such as coffee leaf rust and other. [4] discusses to address this challenge of low coffee productivity in Rwanda, government programs and development partners emphasized dissemination and adoption of agricultural technologies are needed.

Chapter 3: Research Methodology

This chapter presents the high-level architecture of the proposed IoT and ML based precision Agriculture system. The system is subdivided into three phases (see Figure 3.1): data acquisition phase, computing phase, and data visualization (field owner communication phase). In this system a Pi camera module sensor is used to collect several images for data acquisition. The Pi camera module is connected to the CSI (Camera Serial Interface) port of the raspberry pi through a ribbon cable. The Raspberry Pi is connected to a google cloud API and Firebase API through Wi-Fi. And it sends the images that are taken at a regular interval to the APIs to store, preprocess, analyze, and visualize the processed data. A model is trained using a deep learning algorithm called ResNext with an online dataset called RoCoLe, which contains 1560 coffee leaf images. The trained model is deployed in Google cloud as well as in a local serve for a farmer to check the status of the leaves both in real-time and off time.

Figure 3.1: High level Architecture of IoT and ML based Precision Agriculture

3.1 Data Acquisition Layer

3.1.1 Pi Camera Module v1.3

Pi camera isa high-Definition camera module compatible with all Raspberry Pi models. Provides high sensitivity, low crosstalk, and low noise image capture in an ultra-small and lightweight design. It is applicable in different IoT, ML, projects by integrating with Raspberry Pi. Raspberry Pi Board has CSI (Camera Serial Interface) interface to which we can attach Pi camera module directly. This Pi Camera module comes with fixed focus lens, 2592@-1944 image size pixels, a, omnivision 5647 fixed-focus sensors, 2.9 aperture, 72.4A FOV, and camera serial interface. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data [31]

FIGURE 3. 2: PI CAM V1.3

3.2 Processing Layer

3.2.1 Raspberry Pi 3 Model B

The Raspberry Pi 3 Model B, a credit card sized board computer, comes with Quad Core 1.2GHz Broadcom BCM2837 64bit CPU; 1GB RAM; BCM43438 wireless LAN and Bluetooth Low Energy on board; 100 Base Ethernet; 40-pin extended GPIO; 4USB 2 ports; 4 Pole stereo output and composite video port; full size HDMI; DSI display port for connecting a Raspberry pi touchscreen display; Micro SD port for loading operating system and storing data; Micro USB power source up to 2.5A, and CSI camera port for connecting a Raspberry Pi camera.

Like any other computer, were Operating system acts as backbone for operation. Raspberry Pi, facilitates opensource operating systems based on Linux. The various accessories like Camera, Gert board and Compute Model Kit and it is capability of supporting various Programming languages like Python, the main programming language which is used by Pi makes it ideal for IoT and machine learning based projects [32].

Figure 3.3: Raspberry pi 3 Module B [32]

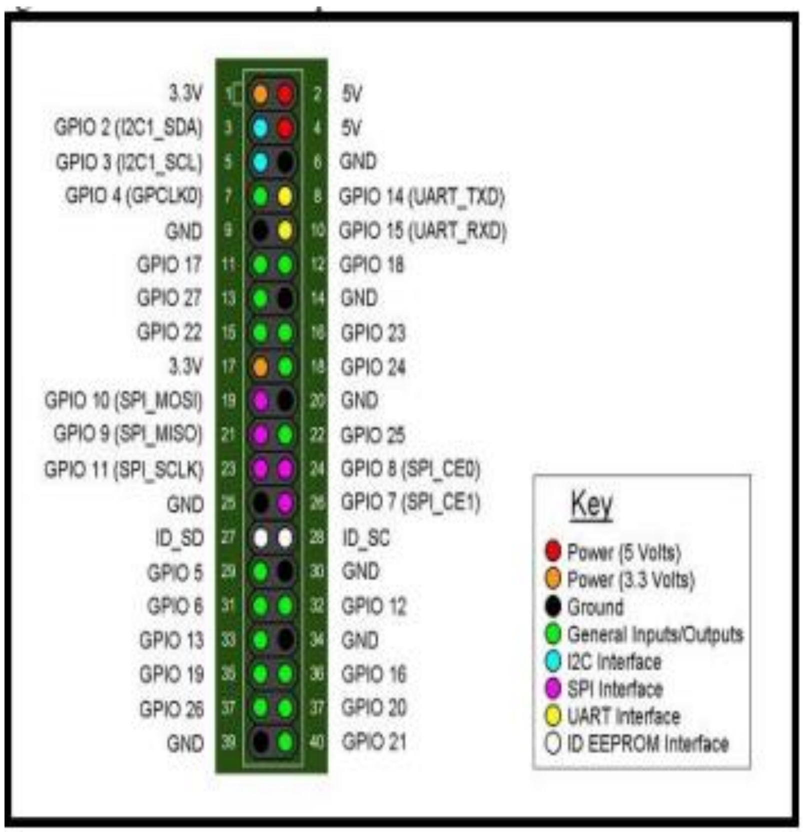

FIGURE 3.4: GPIO PINS OF RASPBERRY PI [33]

GPIO facilitates connecting all sorts of peripheral devices to Raspberry Pi. Raspberry Pi has onboard GPIO with 40 pins, 26 of which are used as digital inputs or outputs. More importantly, 9 of the 14 new GPIO pins are dedicated inputs/outputs, it also facilitates the onboard UART, I2C, SPI Bus and still large amount of free GPIO pins are there for add-on attachments.
One of the drawbacks of Raspberry pi is It cannot act as full-fledged computer because the Ethernet Port and Processing CPU is not so fast to process multitasking computing cycles. Secon it is not compatible fully functional Windows Operating System Doesn’t have built-in ADC Convertor.

External charger is used for ADC purpose. For such reasons in this system, we used an ethernet cable to connect it with a computer through a static IP address.

Figure 3.5: CSI Camera Connector

Raspberry Pi 3 model B has a 15-pin MIPI camera interface (CSI) Connector Mobile Industry Processor Interface (MIPI) Camera Serial Interface. CSI-2 facilitates connection of small camera to Broadcom BCM 2835 processor. The function of this interface is to standardize the attachment of camera modules to the processors for the mobile phone industry [34].

3.2.2 32GB Micro SD card

32GB micro-SD card is used in the system to store the OS of Raspberry Pi, program files of Raspberry Pi board, and applications like python IDE, visual studio. Digital Card slot (SD Card) slot is a solid-state removable storage device which is required to run operating systems on Raspberry Pi as Raspberry Pi doesn't have any onboard memory and data storage functionality. Raspberry Pi supports both SDHC (Secure Digital High Capacity) and SDXC (Secure Digital eXtended Capacity). The best suited card for proper running of all sorts of operating systems without any hiccup is Class 10 with speed @ 10MB/sec [32].

Figure 3. 6: SD-Card with 32 GB Capacity

3.2.3 Micro USB Power cable with output 2.5A

Micro-USB Power Supply: Raspberry Pi requires 5 volts (V) +/- 5% per USB 2.0 standard. Talking of various models of Raspberry Pi: Model B: 700mA at 3.5 watts (W); Model A: 500mA at 2.5 watts. The power port on PCB of Raspberry Pi is Micro-USB type B interface, so a Pi compatible

power supply uses standard USB A connector on one side and MicroUSB B connector on other side [35].
To power up the Raspberry Pi 3 board it requires a micro-USB power cable. In this system we used a USB power supply cable with input 0.5A and output 2.5A.

3.2.4 RJ45 Ethernet Cable

Raspberry Pi Model B comprise Ethernet Port: In order to enable Internet connection online and to update the software's or to install latest packages from online repositories, Raspberry Pi supports Ethernet Connection. Raspberry Pi (Every Model) comprises of RJ45 Ethernet Jack which supports CAT5/6 cables. It enables Raspberry Pi to be connected to Wireless Router, ADSL Model or any other Internet connectivity sharing device.
RJ45 is the standardized networking interface primarily used for Ethernet cables. In this system the RJ45 cable is used to connect the Raspberry pi and the laptop through VNC viewer.

3.3 Data Analysis and Visualization Layer

3.3.1 MODEL IMPLEMENTATION

A dataset is the first step in developing a machine learning or deep learning models. The performance of any machine learning or Deep Learning model depends on the quantity, quality, and relevancy of the dataset. In this system a set of images of healthy and unhealthy coffee leaves were needed to train a model which can perform a better accuracy. The quality of the dataset and its labelling has a great impact on the accuracy of the developed models. In this system several pictures of coffee leaves were taken using the raspberry pi camera at the same time we also took some pictures manually using a phone camera in a coffee field in Rwanda but those images were not quit enough to train our model at the same time those data were collected in one day. For such reasons we used an online coffee dataset called RoCoLe which consists 1560 images. The dataset consists 791 healthy images and 769 unhealthy images, in different condition. We chose this data set because of its size and variability. Below (**Figure 3.9**) is a sample of healthy and unhealthy (affected by CLR) images taken from a coffee field in Rwanda.

(a) (b)

FIGURE 3. 7: COFFEE LEAF: UNHEALTHY IMAGE (A), HEALTHY IMAGE (B)

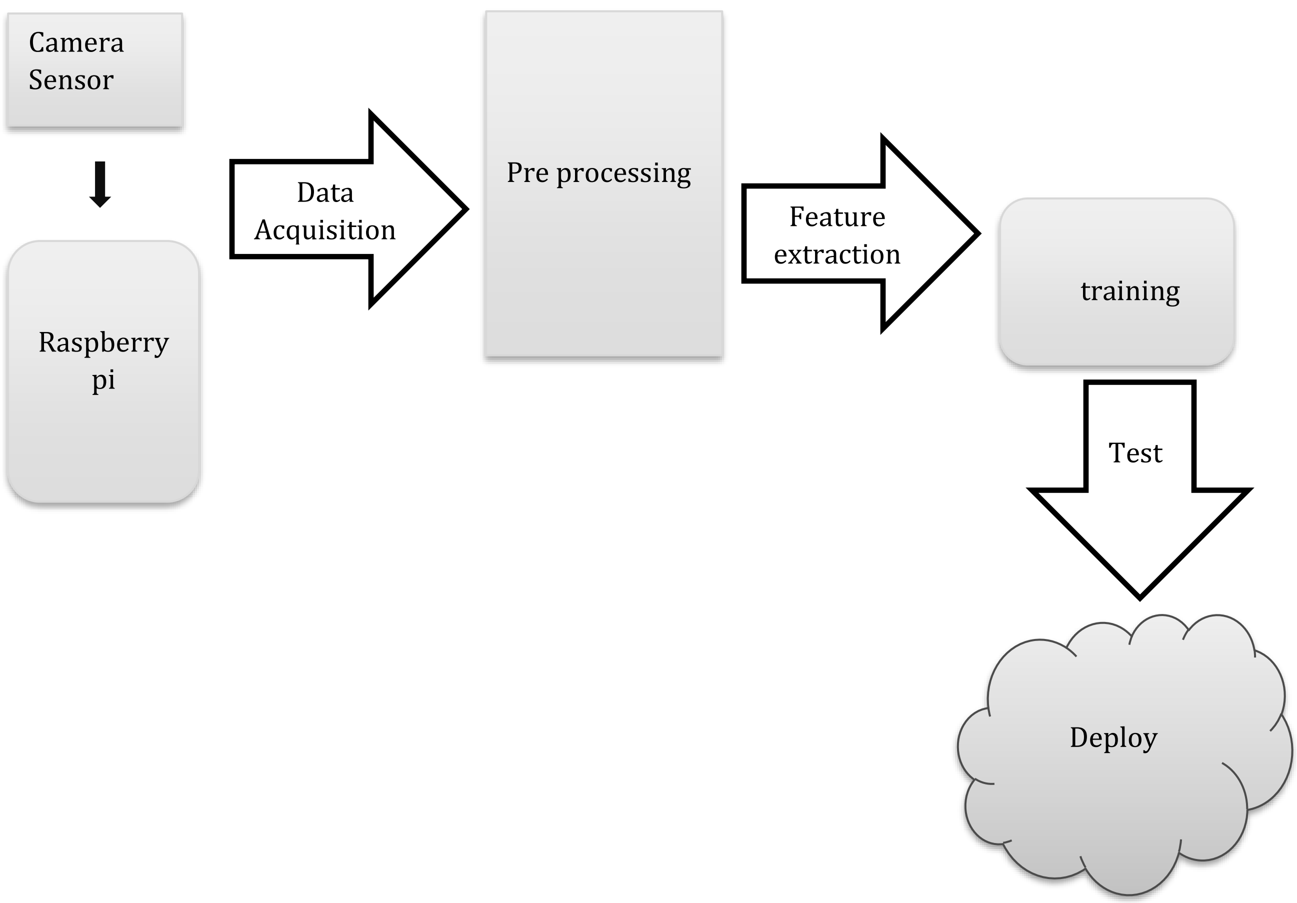

Figure 3.8: Block Diagram

3.3.2 Image preprocessing

There are many ways to prepare a dataset to use it for model training. Simple and easy way is creating different folders for each class(label) and putting each image in the corresponding folder. This makes it very easy to load the dataset and train it. Our dataset consists of 1560 images. We have used 80% of the dataset to train the model and 20% to test. Each is leveled as healthy and unhealthy (red spider mite presence, rust_level_1, rust_level_2, rust_level_3, rust_level_4)

TABLE 2: ILLUSTRATION OF THE FOLDER STRUCTURE

Category	**No of images**
Healthy	791
Red spider mite presence	167
Rust_level_1	344
Rust_level_2	166
Rust_level_3	62
Rust_level_4	30

	File	Binary.Label	Multiclass.Label
0	C1P1H1.jpg	healthy	healthy
1	C1P2E2.jpg	unhealthy	rust_level_2
2	C1P2H1.jpg	healthy	healthy
3	C1P3E1.jpg	healthy	healthy
4	C1P3E2.jpg	unhealthy	rust_level_2
5	C1P3H1.jpg	healthy	healthy
6	C1P3H2.jpg	unhealthy	rust_level_3

FIGURE 3.9: DATA ANNOTATIONS

(a) Original image **(b) Preprocessed image**

Size: 2322*4128 **Size: 200*200**

FIGURE 3. 10: (A) ORIGINAL IMAGE, (B) PREPROCESSED IMAGE

Image preprocessing such as normalization, resizing, random flip, random rotation, scale, and deform image wase applied. Before the model reads the image, it is set to resize to a given width and height (200 * 200) square area which is the actual training image size as it is showed in the Figure 3.12. After preprocessing data transform is applied to convert RGB image to pytourch tensor, normalizing, data augmentation for training set.

3.3.3 Training

After the dataset is preprocessed, the next step is training. Training is the key step in machine learning that results in a model ready to be validated, tested, and deployed. The algorithm used for this project is ResNext. It is a deep learning algorithm designed for image classification.

We choose this algorithm because of its better results on ImageNet comparing to ResNet, inception, VGGnet, and others. Also, it is relatively smaller than other big models. Inherited from ResNet, VGG, and Inception, ResNeXt includes shortcuts from the previous block to next block, stacking layers and adapting split-transform-merge strategy.

ResNext is a simple network for image classification, it uses the same block of layers multiple times that contains a set of transformation functions that helps in classifying the image it is also made up of same repeated blocks, the architecture (figure 3.13) is homogenous and the hyperparameters are reduced [35].

According to the authors ResNet, having shortcuts but ResNeXt has a much parallel stacking layer rather than sequential layers. Similar to Inception module, ResNeXt follows a split-transform-merge strategy but ResNeXt shares hyper-parameters while Inception has a different filter and size for each individual block.
Cardinality is introduced which means the size of the set of transformations. Increasing cardinality led to better result while keeping architecture complexity [36] .

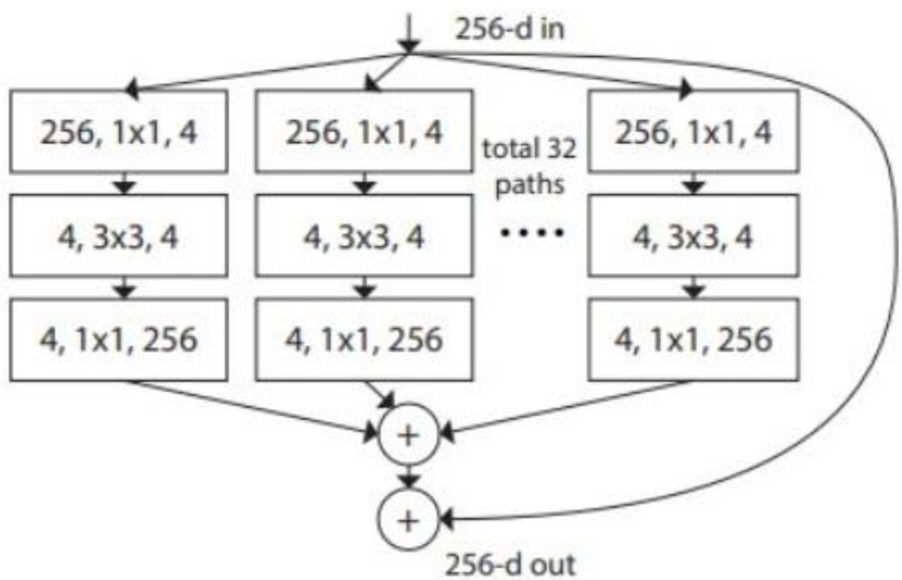

FIGURE 3.11: ***(A) ORIGINAL IMAGE, (B) PREPROCESSED IMAGE*** **[37]**

In our case we applied the serial neural networks as shown in the function below to find the local minimum of Serial Neural networks and Residual Neural networks

Serial Neural networks:

$$y=mx+b$$

$$y1=\sigma(W1x+b1)$$

$$y2=\sigma(W2y1+b2)$$

Residual Neural networks:

$$y=mx+b$$

$$y1=\sigma(W1x+b1)$$

$$y2=\sigma(W2y1+b2+x)$$

In the equations we have a loss which is a function of w and b. This function starts at some random point x, find the derivative of that function with respect to x evaluated at that point and subtract the derivative value from the x. This will move to the direction of local minimum for the loss.

In the case of residual networks in normal network the information will flow from one layer to the next layer. Let's say we have two-layer neural network and let's represent the first layer as a function f and the second layer as function g. If we have input x the output from the model(y) can be written as:

Y=g(f(x)) or with intermediate output represented as h (output from the first layer)

$$h=f(x)$$

$$y=g(h)$$

but in the case of residual neural networks the information might jump to the next layers. For example, with residual neural network the above two layer could be

$$h=f(x)$$

$$y=g(h+x)$$

as you can see here the x is input to both the first and the second layer. This way there is two way the information the input is reaching the second layer. First from f and second directly from x.

And loss is described in terms of w and b, which are weights and bias for a layer respectively.

$$y = mx + b$$

y_1 = (W_1x + b_1)

y_2 = (W_2y_1 + b_2)

Optimization:

Wnew=Wold−ΔwLossbnew=bold−ΔbLoss

3.3.4 Google Colab

We use google collab as a notebook to train our model. Because of its free resources specifically GPU.

Figure 3.12: Google Colab

Chapter 4: Configuration and Deployment

This chapter tells us how the hardware and software part of the system are configured and how it is deployed both in the cloud and in a local server. In this prototype (Figure 4.1 shows) a pi camera sensor connected to a raspberry pi for capturing an image. The raspberry pi pushes the data over wireless network to google cloud platform and firebase, to store analyze and visualize the results.

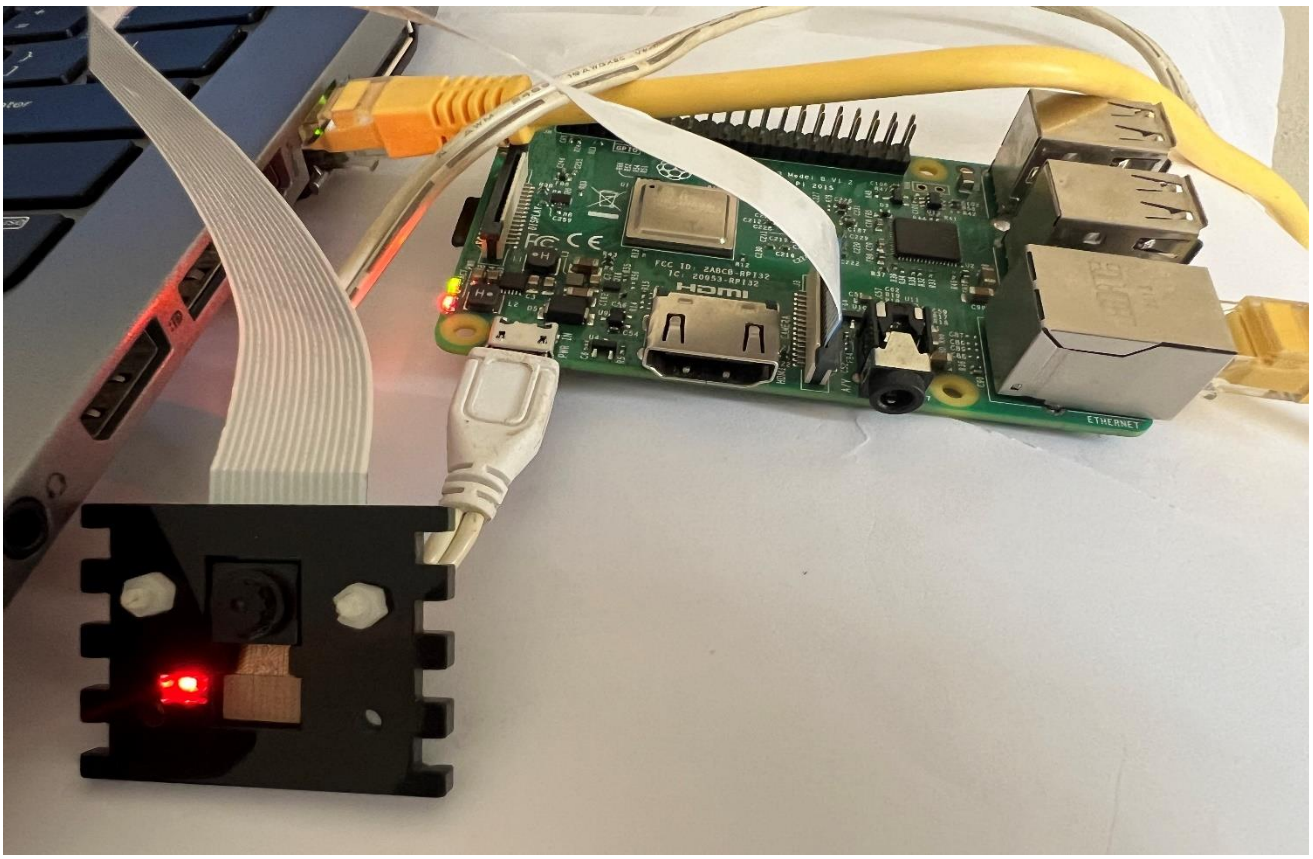

FIGURE 4.1: EXPERIMENTAL SETUP OF PROTOTYPE

4.1 Components Configuration

The circuit diagram bellow (figure 4.2) shows how the camera module is connected to the raspberry pi of CSI through ribbon cable. One of the draw backs of raspberry pi is It cannot act as full-fledged computer [34]. So, we connected it to a computer through the Ethernet Port. We installed VNC viewer in the computer and assigned an Ip address to connect it with raspberry pi.

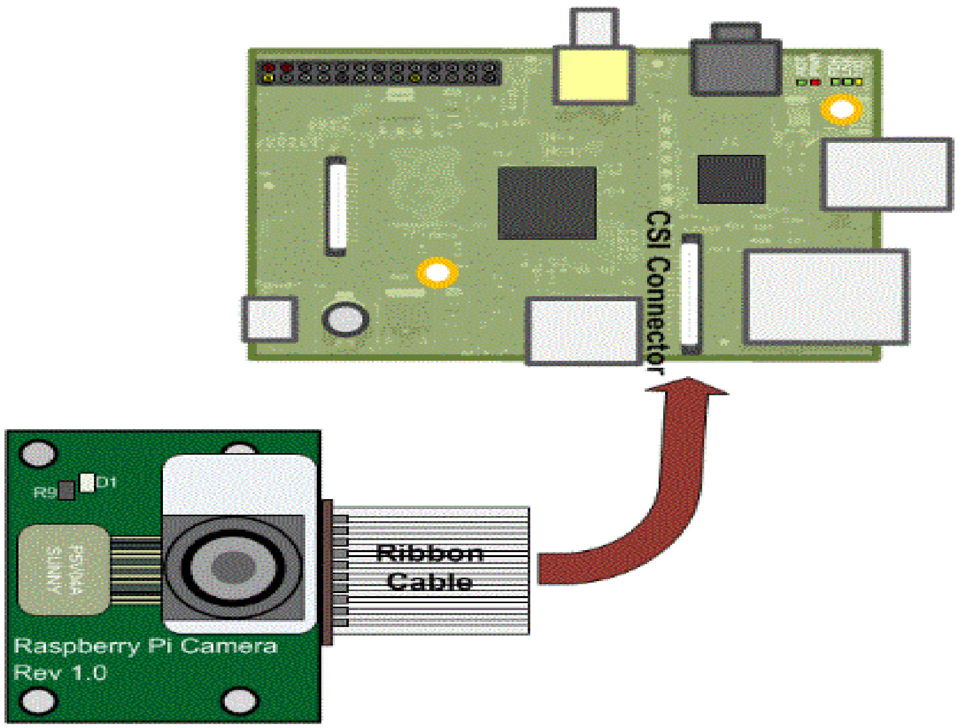

Figure 4.2: Circuit diagram

4.2 VNC Viewer

To easily access and control the raspberry pi we installed VNC viewer in the laptop and set up a static IP address. The raspberry pi is connected to the computer through RJ45 ethernet cable through the Ethernet ports. VNC offers a deceptively simple service - it allows you to view and control a remote system as though seated next to it, wherever you are. The compact VNC Server 4 application runs on the system to be controlled [38].

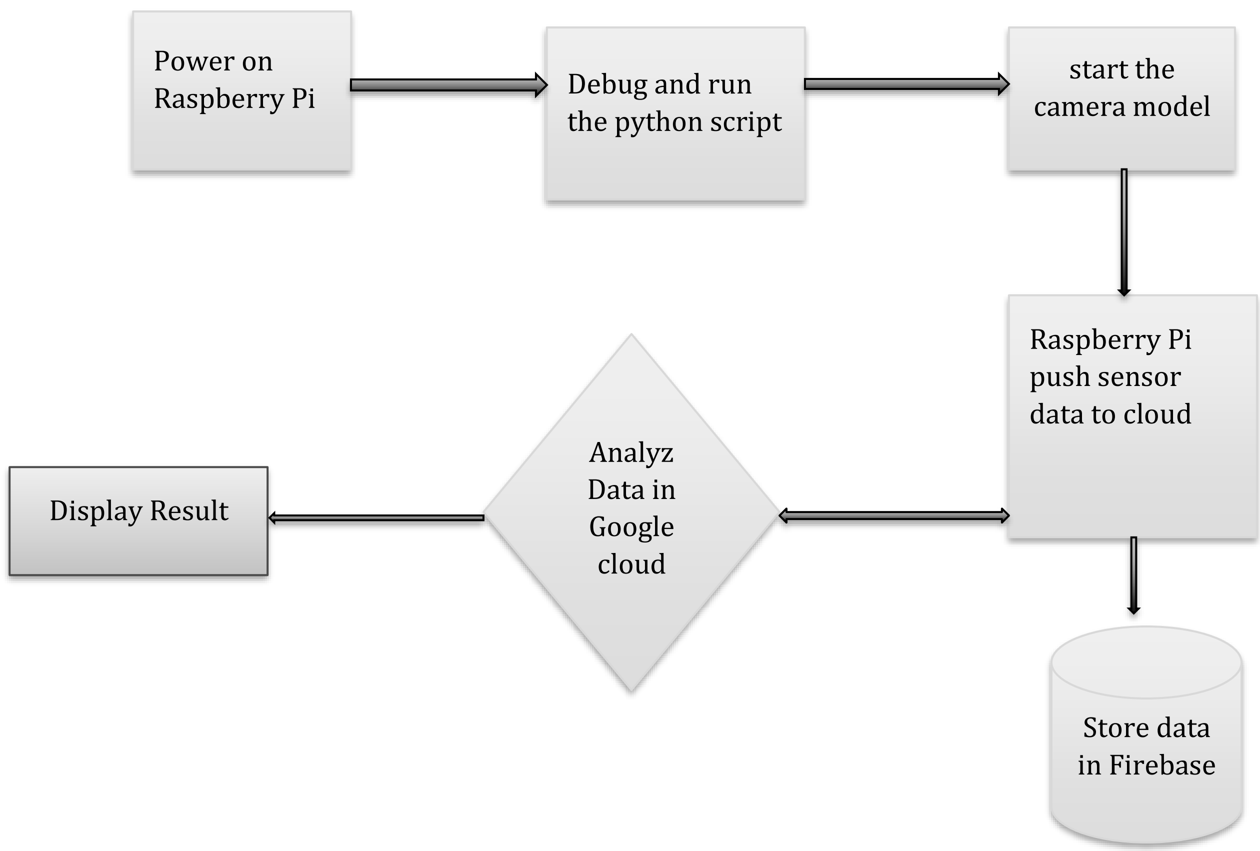

Figure 4.4: System and Software Design Architecture

Figure 4.4 shows the flow diagram of the system. At first the Raspberry pi will boot and after it boot the predefined python script will automatically run. Then the camera sensor captures images. When Raspberry Pi receives the data, it will push those data to the Firebase and google cloud for storage and analysis. User will use the application which is connected to google cloud and fire base to retrieve the data.

4.3 Firebase

Firebase is a set of hosting services for any type of application. t provides many features like Authentication & Security, Realtime Database & File Storage. The Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time.

The data is synced across all clients in real time and is still available when an app goes offline [39]. In our system we used firebase as real-time database to store images for training and future use of mobile application development. We integrated firebase with raspberry pi through an API using a python code. Once the camera module starts taking images the raspberry pi pushes the data immediately to firebase through Wi-Fi.

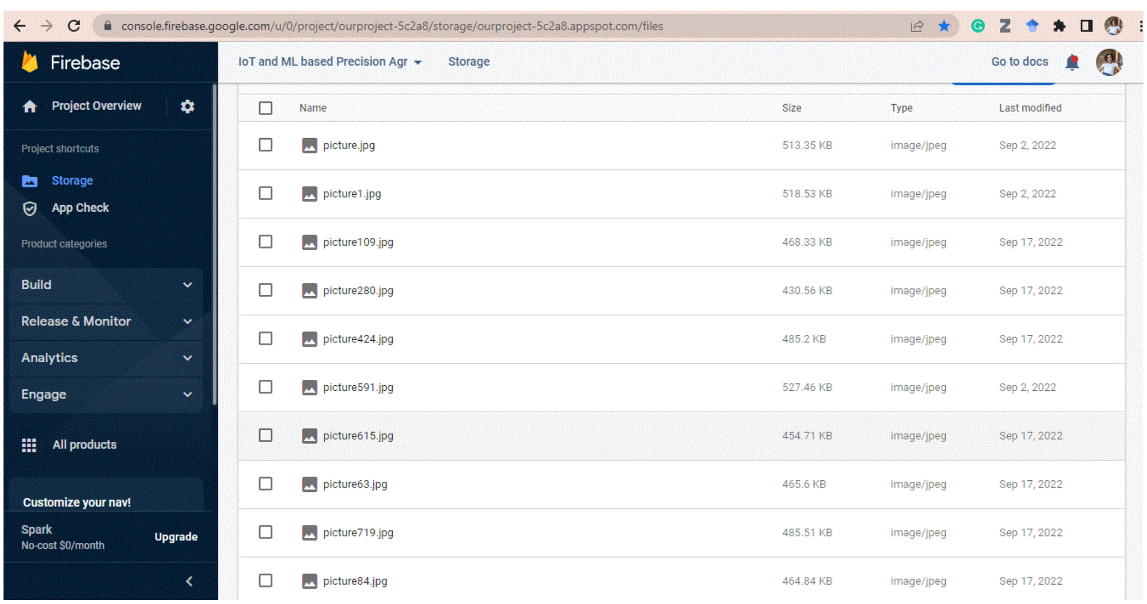

FIGURE 4.5: FIREBASE REAL-TIME DATABASE

4.4 Model Deployment

After the model is trained and tested. We deployed it both in google cloud platform and in a web-based application using FastAPI for user accessibility.

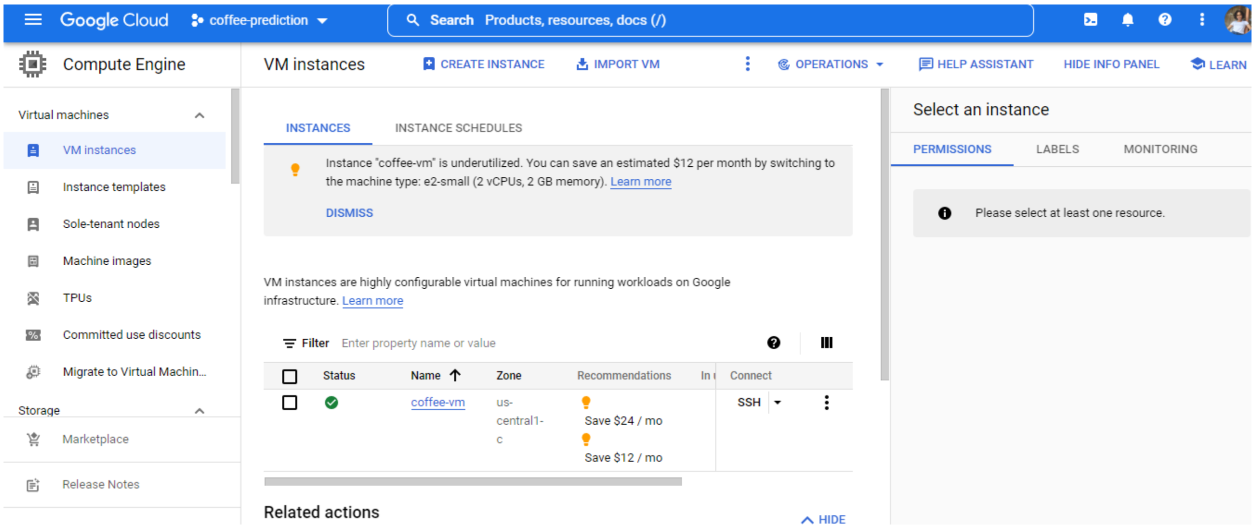

FIGURE 4.6: GOOGLE CLOUD PLATFORM MODEL DEPLOYMENT

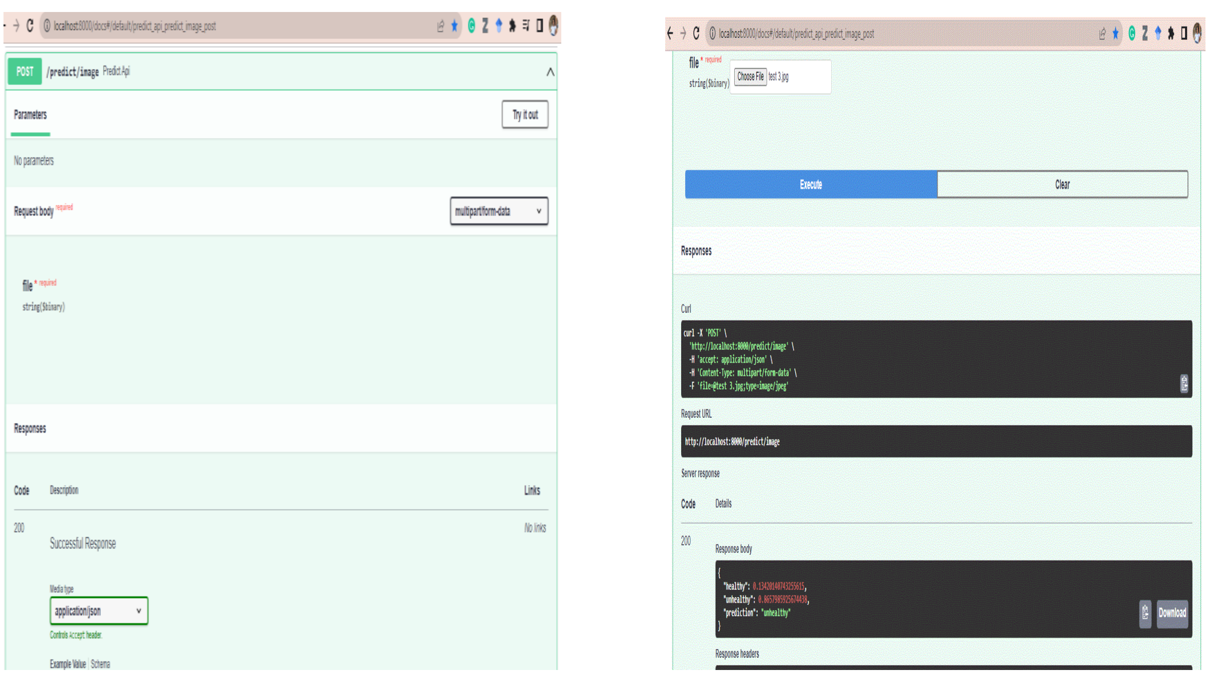

FIGURE 4.7: GUI OF THE WEB-BASED APPLICATION

Chapter 5: Results and Analysis

5.1 Introduction

In our prototype IoT and ML based precision agriculture, the camera sensor and raspberry pi are correctly integrated and configured to train the model. A machine with GPU were needed because of the speed and high computation on deep learning. It could have been trained on CPU but GPU speeds up the task and it trains faster. For such reasons we used Google Collab GPU, where we store the results in google drive. We imported all the required packages and the model is successfully trained in Google Collab with 12 GB RAM, and 2.3 GHx frequency.

The trained model is deployed on google cloud platform and an API is generated as computing as a service. The raspberry pi provides data to google cloud which is computing as a service task through the google cloud API to post image, store and get the analyzed result through a Wi-Fi. A web-based application is also developed for easy access to user.

Our model uses 80% of the dataset for training and 20% for testing. The trained model achieved 91% training accuracy and 87% test accuracy through different experiments. The purpose of the experiment was to increase the accuracy of the model by evaluating the performance by changing different hypermeters (learning rate, number of epochs).

5.2 Experiment one

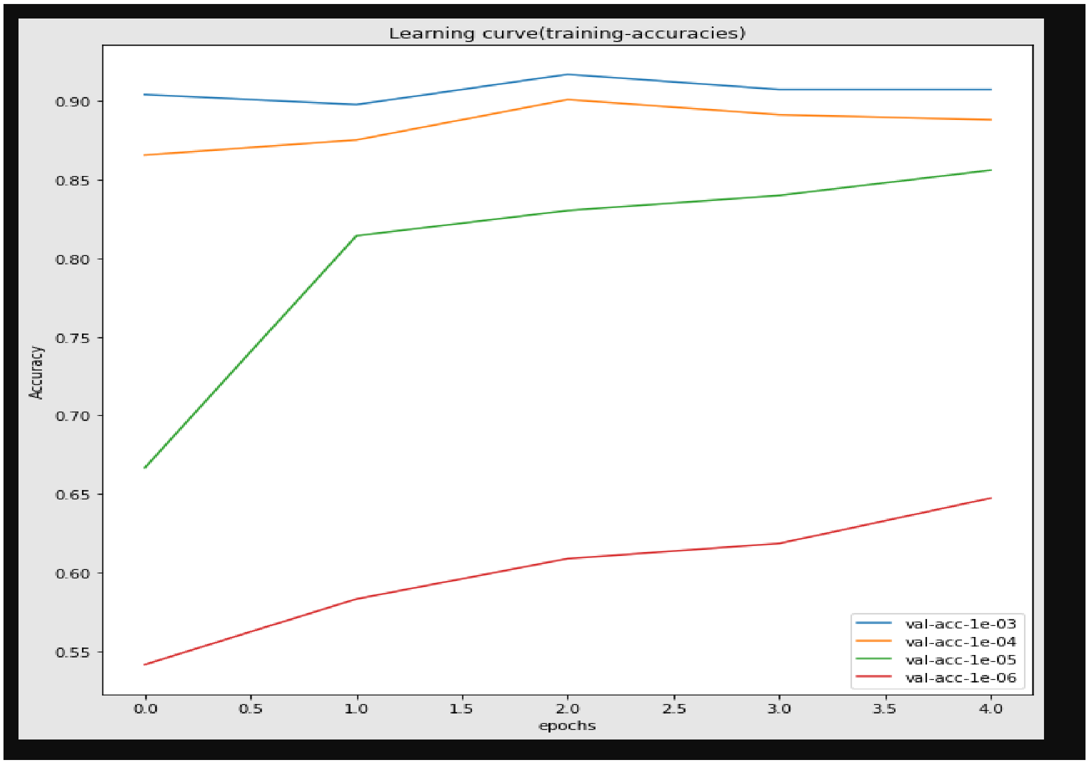

FIGURE 5.1: LEARNING CURVE 1

In this experiment 80% of the dataset were used to train the model and the remaining 20% were used to test the model. The batch size of the training was set to 30. The learning rate was set to Ten the power of minus three ($10^{-3)}$, Ten the power of minus four (10^{-4}), then the power of minus five (10^{-5}), and Ten the power of minus six (10^{-6}). Number of epochs was set to five. Bellow figure 5.2 shows validation loss and accuracy.

```
Epoch 1/5
----------
train Loss: 0.6955 Acc: 0.5056
valid Loss: 0.6951 Acc: 0.4904

Epoch 2/5
----------
train Loss: 0.6936 Acc: 0.5120
valid Loss: 0.6920 Acc: 0.4936

Epoch 3/5
----------
train Loss: 0.6919 Acc: 0.5040
valid Loss: 0.6901 Acc: 0.4936

Epoch 4/5
----------
train Loss: 0.6905 Acc: 0.5224
valid Loss: 0.6883 Acc: 0.5064

Epoch 5/5
----------
train Loss: 0.6876 Acc: 0.5377
valid Loss: 0.6861 Acc: 0.5256

Training complete in 6m 53s
Best val Acc: 0.525641
```

FIGURE 5.2: VALIDATION LOSS VS ACCURACY

5.3 Experiment two

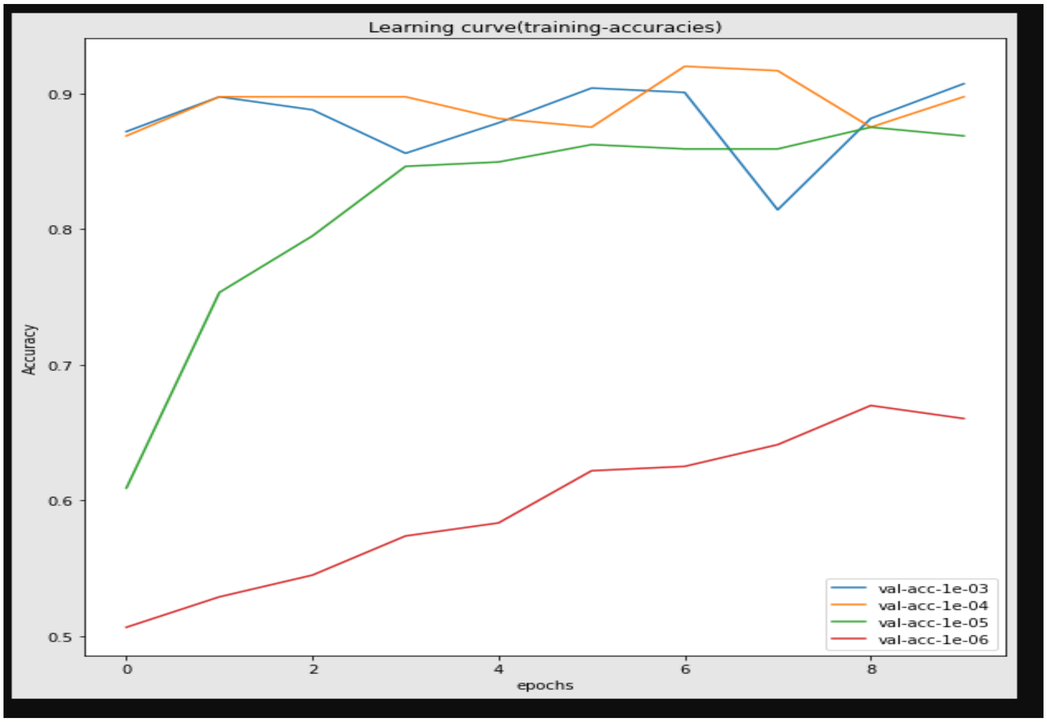

Figure 5.3: learning curve 2

In this experiment 80% of the dataset were used to train the model and the remaining 20% were used to test the model. The batch size of the training was set to 30. The initial learning rate was set to 10^{-3}. Learning rates were set to $10^{-3,}$ 10^{-4}, 10^{-5}, 10^{-6} and number of epochs was set to 10.

```
Epoch 6/10
----------
train Loss: 0.6790 Acc: 0.6370
valid Loss: 0.6800 Acc: 0.6218

Epoch 7/10
----------
train Loss: 0.6759 Acc: 0.6635
valid Loss: 0.6782 Acc: 0.6250

Epoch 8/10
----------
train Loss: 0.6760 Acc: 0.6450
valid Loss: 0.6751 Acc: 0.6410

Epoch 9/10
----------
train Loss: 0.6735 Acc: 0.6739
valid Loss: 0.6731 Acc: 0.6699

Epoch 10/10
----------
train Loss: 0.6714 Acc: 0.6939
valid Loss: 0.6715 Acc: 0.6603

Training complete in 13m 24s
Best val Acc: 0.669872
```

FIGURE 5. 4: ACCURACY VS VALIDATION LOSS

5.4 Experiment three

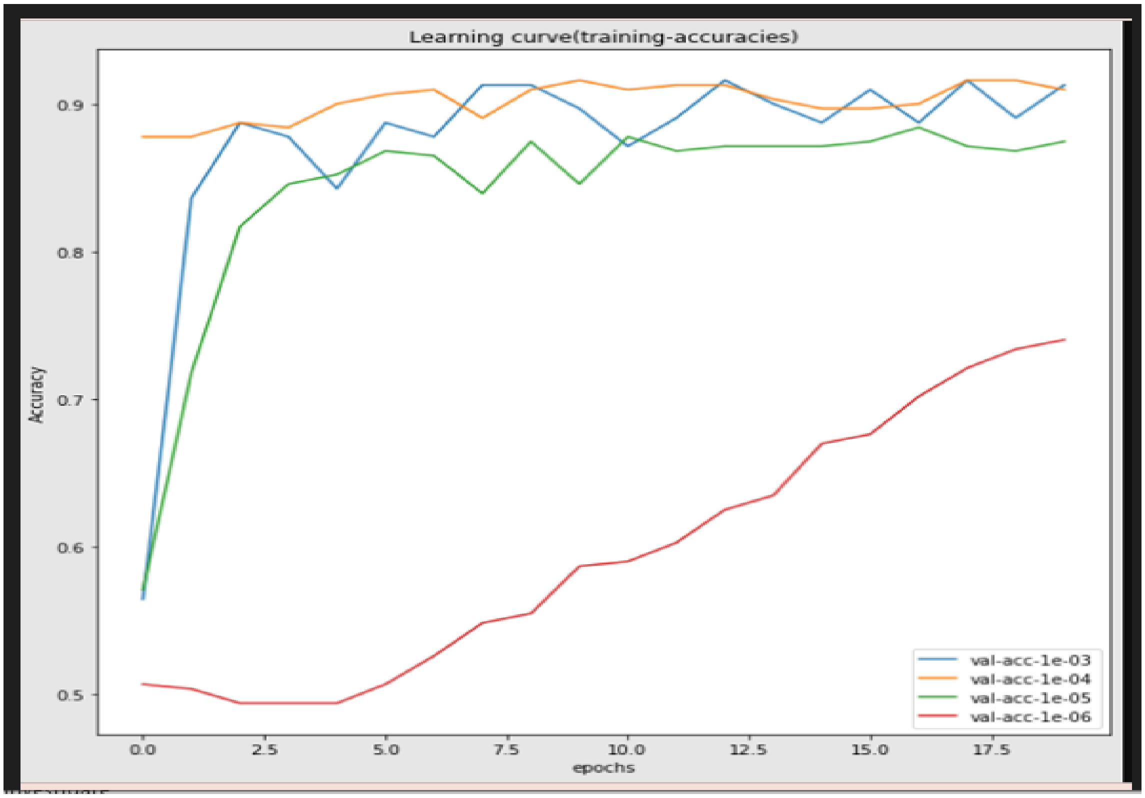

FIGURE 5.5: LEARNING CURVE 3

In this experiment 80% of the dataset were used to train the model and the remaining 20% were used to test the model. The batch size of the training was set to 30. The initial learning rate is set to 10^{-3}. Learning rates were set to $10^{-3,}$ 10^{-4}, 10^{-5}, 10^{-6} and number of epochs was set to 20.

```
Epoch 10/20
----------
train Loss: 0.6851 Acc: 0.5561
valid Loss: 0.6837 Acc: 0.5865

Epoch 11/20
----------
train Loss: 0.6830 Acc: 0.5745
valid Loss: 0.6811 Acc: 0.5897

Epoch 12/20
----------
train Loss: 0.6814 Acc: 0.5954
valid Loss: 0.6798 Acc: 0.6026

Epoch 13/20
----------
train Loss: 0.6786 Acc: 0.6170
valid Loss: 0.6767 Acc: 0.6250

Epoch 14/20
----------
train Loss: 0.6761 Acc: 0.6394
valid Loss: 0.6751 Acc: 0.6346

Epoch 15/20
----------
train Loss: 0.6752 Acc: 0.6442
valid Loss: 0.6732 Acc: 0.6699

Epoch 16/20
----------
train Loss: 0.6731 Acc: 0.6707
valid Loss: 0.6702 Acc: 0.6763

Epoch 17/20
----------
train Loss: 0.6718 Acc: 0.6811
valid Loss: 0.6679 Acc: 0.7019

Epoch 18/20
----------
train Loss: 0.6702 Acc: 0.6899
valid Loss: 0.6657 Acc: 0.7212

Epoch 19/20
----------
train Loss: 0.6674 Acc: 0.7075
valid Loss: 0.6641 Acc: 0.7340

Epoch 20/20
----------
train Loss: 0.6657 Acc: 0.7236
valid Loss: 0.6616 Acc: 0.7404

Training complete in 26m 51s
Best val Acc: 0.740385
```

FIGURE 5.6: VALIDATION LOSS VS ACCURACY

5.5Experiment four

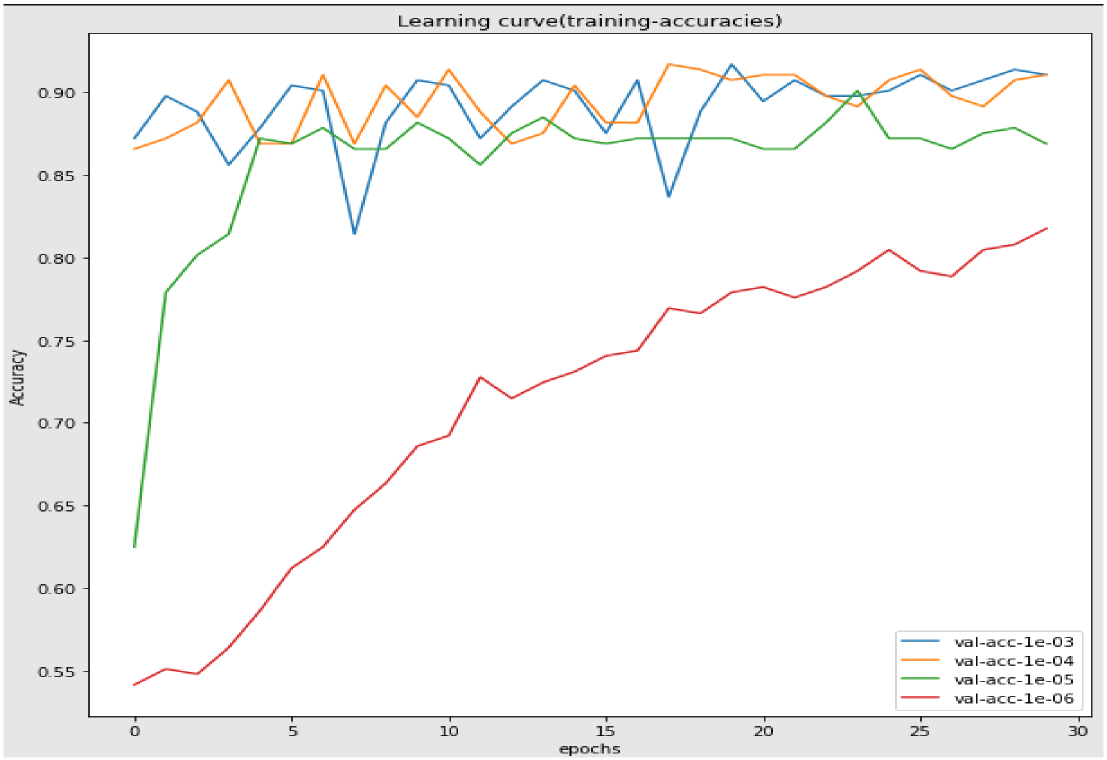

FIGURE 5.8: LEARNING CURVE 4

In this experiment 80% of the dataset were used to train the model and the remaining 20% were used to test the model. The batch size of the training was set to 30. The initial learning rate is set to 10^{-3}. Learning rates were set to $10^{-3,}$ 10^{-4}, 10^{-5}, 10^{-6} and number of epochs was set to 30.

```
Epoch 20/30
----------
train Loss: 0.3201 Acc: 0.8710
valid Loss: 0.2553 Acc: 0.9006

Epoch 21/30
----------
train Loss: 0.2934 Acc: 0.8798
valid Loss: 0.2480 Acc: 0.9103Epoch 22/30
----------
train Loss: 0.3165 Acc: 0.8646
valid Loss: 0.2487 Acc: 0.9167

Epoch 23/30
----------
train Loss: 0.3031 Acc: 0.8814
valid Loss: 0.2461 Acc: 0.9167

Epoch 24/30
----------
train Loss: 0.2784 Acc: 0.8878
valid Loss: 0.2482 Acc: 0.9167

Epoch 25/30
----------
train Loss: 0.3152 Acc: 0.8710
valid Loss: 0.2516 Acc: 0.9071

Epoch 26/30
----------
train Loss: 0.3212 Acc: 0.8654
valid Loss: 0.2564 Acc: 0.9071

Epoch 27/30
----------
train Loss: 0.3322 Acc: 0.8558
valid Loss: 0.2603 Acc: 0.8942

Epoch 28/30
----------
train Loss: 0.3048 Acc: 0.8742
valid Loss: 0.2714 Acc: 0.8878
```

```
Epoch 26/30
----------
train Loss: 0.3212 Acc: 0.8654
valid Loss: 0.2564 Acc: 0.9071

Epoch 27/30
----------
train Loss: 0.3322 Acc: 0.8558
valid Loss: 0.2603 Acc: 0.8942

Epoch 28/30
----------
train Loss: 0.3048 Acc: 0.8742
valid Loss: 0.2714 Acc: 0.8878

Epoch 29/30
----------
train Loss: 0.2999 Acc: 0.8806
valid Loss: 0.2657 Acc: 0.8974

Epoch 30/30
----------
train Loss: 0.2905 Acc: 0.8774
valid Loss: 0.2596 Acc: 0.9071

Training complete in 40m 30s
Best val Acc: 0.916667
```

FIGURE 5.7: ACCURACY VS VALIDATION LOSS

TABLE 3. 1: COMPARISON BETWEEN THE FOUR EXPERIMENT GRAPHS (NUMBER OF EPOCHS 5,10,20, AND 30)

Graph	Number of Epoch	Train Loss	Train Acc	Valid Loss	Valid Acc	Training complete(time)	Best Val Acc
1	5	0.6876	0.5377	0.6861	0.5256	6m 53s	0.525641
2	10	0.6714	0.6939	0.6715	0.6603	13m 24s	0.669872
3	20	0.6657	0.7236	0.6616	0.7404	26m 51s	0.740385
4	30	0.2878	0.8718	0.2416	0.9199	40m 43s	0.919872

Chapter 6: Conclusion, Recommendation and Future Directives

6.1 Conclusion

The effect of CLR in coffee plants is devastating and it can cause a huge economy loss which affects many lives. IoT and ML based precision agriculture is a simple and effective way to monitor coffee leave crops in a real-time, where sensed data is sent to the cloud via Wi-Fi for storage, analysis, and visualization on web-based application and in google cloud platform.

The system collects images using a pi camera sensor connected to a raspberry pi, and the raspberry pi sends the data to google cloud and firebase through Wi-Fi. A python script code is written to connect raspberry pi with google cloud and firebase through APi. A model is trained in deep learning algorithm, ResNext and deployed in google cloud platform. The trained model achieved 91% training accuracy and 87% test accuracy.

The main challenge was data collection as machine learning algorithms need a lot of quantity and quality data to achieve a better accuracy. Because of budget, time, and covid restrictions we did not get enough data to train our model. For such reasons we used an online dataset called ROCLe. And the model has trained in several days and achieved the expected result.

5.2 Recommendation

The system can be used in the different coffee growing areas in Rwanda and more sensors can be added in future if needed, to increase the accuracy and coverage area long range connectivity such as LoRaWAN and Sigfox can be used.

Pi Cam v1.3 camera sensor is not very accurate as it has less resolution. Other versions of camera sensors can be deployed to improve on accuracy and to obtain reliable results.

6.3 Future Directives

The experimented need to be repeated with more reliable and accurate sensors for a better training. We can connect those sensors to firebase to store real-time data. We can also deploy a mobile app for easy access to the user with firebase itself

References

[1] S. Krishnan and S. Krishnan, *Sustainable Coffee Production*, no. September. 2017. doi: 10.1093/acrefore/9780199389414.013.224.

[2] A. Abraham, "Impact of Antestia bug (Antestiopsis sp .) on Coffee (Coffea arabica L .) Production and Quality," *J. Biol. Agric. Healthc.*, vol. 6, no. 21, pp. 18–22, 2016.

[3] I. Nzeyimana, A. E. Hartemink, and J. de Graaff, "Coffee farming and soil management in Rwanda," *Outlook Agric.*, vol. 42, no. 1, pp. 47–52, 2013, doi: 10.5367/oa.2013.0118.

[4] J. Ngango and S. G. Kim, "Assessment of technical efficiency and its potential determinants among small-scale coffee farmers in rwanda," *Agric.*, vol. 9, no. 7, pp. 1–12, 2019, doi: 10.3390/agriculture9070161.

[5] A. Harelimana, G. Le Goff, D. T. N. Rukazambuga, and T. Hance, "Coffee Production Systems : Evaluation of Intercropping System in Coffee Plantations in Rwanda[1] A. Harelimana, G. Le Goff, D. T. N. Rukazambuga, and T. Hance, "Coffee Production Systems : Evaluation of Intercropping System in Coffee Plantations in Rwand," vol. 10, no. 9, pp. 17–28, 2018, doi: 10.5539/jas.v10n9p17.

[6] H. M. O. Otieno, "Impacts and Management of Termites (Isoptera: Termitidae) among Smallholder Farmers In East Africa," *J. Agric. Ecol. Res. Int.*, vol. 16, no. 1, pp. 1–12, 2018, doi: 10.9734/jaeri/2018/44842.

[7] P. Talhinhas *et al.*, "The coffee leaf rust pathogen Hemileia vastatrix: one and a half centuries around the tropics," *Mol. Plant Pathol.*, vol. 18, no. 8, pp. 1039–1051, 2017, doi: 10.1111/mpp.12512.

[8] R. Ntirushwa, "REPUBLIC OF RWANDA (RSSP II) PEST MANANAGEMENT PLAN (PMP) ARRANGEMENTS FOR RSSP II," no. Rssp Ii, 2007.

[9] J. Cho *et al.*, "Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis," *Iinternational J. Math. Comput. Simul.*, vol. 1, no. 1, pp. 46–53, 2007.

[10] Y. Mekonnen, S. Namuduri, L. Burton, A. Sarwat, and S. Bhansali, "Review—Machine Learning Techniques in Wireless Sensor Network Based Precision Agriculture," *J. Electrochem. Soc.*, vol. 167, no. 3, p. 037522, 2020, doi: 10.1149/2.0222003jes.

[11] D. Garg, S. Khan, and M. Alam, “Integrative Use of IoT and Deep Learning for Agricultural Applications Integrative Use of IoT and Deep Learning for Agricultural Applications,” no. November 2019, 2020, doi: 10.1007/978-3-030-30577-2.

[12] F. Balducci, D. Impedovo, D. Informatica, and A. Moro, “Machine Learning Applications on Agricultural Datasets for Smart Farm Enhancement,” 2018, doi: 10.3390/machines6030038.

[13] J. Ngango and S. G. Kim, “Assessment of technical efficiency and its potential determinants among small-scale coffee farmers in rwanda,” *Agric.*, vol. 9, no. 7, 2019, doi: 10.3390/agriculture9070161.

[14] J. Bigirimana, K. Njoroge, D. Gahakwa, and N. A. Phiri, “Incidence and severity of coffee leaf rust and other coffee pests and diseases in Rwanda,” vol. 7, no. 26, pp. 3847–3852, 2012.

[15] K. Gandu, V. Vyshnavi, and L. Guptha, “IOT based Plant Leaf Disease Detection System and Email Alert Using Raspberry PI,” vol. 8, no. 1, pp. 50–54, 2021.

[16] A. Thorat, S. Kumari, and N. D. Valakunde, “An IoT based smart solution for leaf disease detection,” *2017 Int. Conf. Big Data, IoT Data Sci. BID 2017*, vol. 2018-Janua, pp. 193–198, 2018, doi: 10.1109/BID.2017.8336597.

[17] S. K. Bhoi *et al.*, “An Internet of Things assisted Unmanned Aerial Vehicle based artificial intelligence model for rice pest detection,” *Microprocess. Microsyst.*, vol. 80, no. September 2020, p. 103607, 2021, doi: 10.1016/j.micpro.2020.103607.

[18] M. Türkoğlu and D. Hanbay, “Plant disease and pest detection using deep learning-based features,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 3, pp. 1636–1651, 2019, doi: 10.3906/elk-1809-181.

[19] P. P. Patel and D. B. Vaghela, “Crop Diseases and Pests Detection Using Convolutional Neural Network,” *Proc. 2019 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2019*, pp. 1–4, 2019, doi: 10.1109/ICECCT.2019.8869510.

[20] I. Journal, “Early Pest Detection From Crop Using Image Processing and Computational Intelligence,” *Interantional J. Sci. Res. Eng. Manag.*, vol. 06, no. 06, 2022, doi: 10.55041/ijsrem14606.

[21] A. Morbekar, A. Parihar, and R. Jadhav, “Crop disease detection using YOLO,” *2020 Int. Conf. Emerg. Technol. INCET 2020*, pp. 1–5, 2020, doi:

10.1109/INCET49848.2020.9153986.

[22] Nikita, S. Kasar, D. Abuj, and A. Vadvale, "CROP DISEASE PREDICTION AND SOLUTIONYadav," pp. 599–602, 2021.

[23] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, "Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network," *Circuits, Syst. Signal Process.*, vol. 39, no. 2, pp. 818–836, 2020, doi: 10.1007/s00034-019-01041-0.

[24] R. Sharma, S. Das, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, *A Model for Prediction of Paddy Crop Disease Using CNN*, vol. 1119. Springer Singapore, 2020. doi: 10.1007/978-981-15-2414-1_54.

[25] C. R. Rahman *et al.*, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosyst. Eng.*, vol. 194, pp. 112–120, 2020, doi: 10.1016/j.biosystemseng.2020.03.020.

[26] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors (Switzerland)*, vol. 17, no. 9, 2017, doi: 10.3390/s17092022.

[27] P. Bedi and P. Gole, "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network," *Artif. Intell. Agric.*, vol. 5, pp. 90–101, 2021, doi: 10.1016/j.aiia.2021.05.002.

[28] V. Agnihotri, "Machine Learning based Pest Identification in Paddy Plants," *2019 3rd Int. Conf. Electron. Commun. Aerosp. Technol.*, pp. 246–250, 2019.

[29] G. Divya, "RASPBERRY-PI USING NEURAL NETWORK," vol. 7, no. 4, pp. 1063–1069, 2020.

[30] S. Bugingo, "Impact of temperature and rainfall variability on coffee production in Maraba.," 2018.

[31] Rasperry Pi Foundation, "Raspberry Pi Camera Module," pp. 0–6, 2015, [Online]. Available: https://www.raspberrypi.org/documentation/raspbian/applications/camera.md

[32] G. Halfacree, *Raspberry Pi Beginners Guide*. 2020.

[33] H. Halvorsen, "GPIO with Python".

[34] A. Nayyar and V. Puri, "Raspberry Pi-A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no.

12, pp. 720–737, 2015.

[35] Raspberry Pi-c, “Raspberry Pi 12.5W Micro USB Power Supply,” no. October, 2021, [Online]. Available: https://www.raspberrypi.com/products/micro-usb-power-supply/

[36] S. Hitawala, “Evaluating ResNeXt Model Architecture for Image Classification,” 2018, [Online]. Available: http://arxiv.org/abs/1805.08700

[37] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5987–5995, 2017, doi: 10.1109/CVPR.2017.634.

[38] C. M. R. U. Guide, “User Guide User Guide,” *Computer (Long. Beach. Calif).*, no. September, pp. 169–232, 2020, [Online]. Available: www.impact-test.co.uk

[39] C. Khawas and P. Shah, “Application of Firebase in Android App Development-A Study,” *Int. J. Comput. Appl.*, vol. 179, no. 46, pp. 49–53, 2018, doi: 10.5120/ijca2018917200.