

**DEVELOPING AN AUTOMATED FALL ARMYWORM
IDENTIFICATION, EARLY WARNING AND MONITORING
SYSTEM USING A CONVOLUTION NEURAL NETWORK**

**BY
FRANCIS CHULU**

**A dissertation Submitted to the University of Zambia in partial fulfillment of the
requirement for the award of the degree in Master of Science in Computer Science**

**THE UNIVERSITY OF ZAMBIA
LUSAKA**

2020

COPYRIGHT

All rights reserved. No part of this dissertation may be reproduced or stored in any retrieval system or transmitted in any form or by any means, electronic, mechanical photocopying, recording, scanning or otherwise without the prior written permission of the author or the University of Zambia.

DECLARATION

I, Francis Chulu, do hereby declare that this work has not previously been submitted in candidature for any degree. The dissertation is the result of my own work and investigations except where otherwise stated. Other sources have been acknowledged by giving explicit references. A complete list of references is appended.

Signature

Date.....

Supervisor

Signature

Date

CERTIFICATE OF APPROVAL

This document by Francis Chulu is approved as fulfilling the requirements for the award of the degree of Master of Science in Computer Science of the University of Zambia.

.....

Examiner 1	Signature	Date
-------------------	------------------	-------------

.....

Examiner 2	Signature	Date
-------------------	------------------	-------------

.....

Examiner 3	Signature	Date
-------------------	------------------	-------------

.....

Chairperson, Board of Examiners	Signature	Date
--	------------------	-------------

.....

Supervisor	Signature	Date
-------------------	------------------	-------------

ABSTRACT

Since its reported presence in 2016, the Fall Armyworm (FAW) has caused major damage to a number of plant species including maize which is a staple food for most African countries. Their presence in Africa poses a challenge to the food security in many countries contributing to the already existing food problem that the continent has been facing. The outbreak of FAW if left uncontrolled is likely to have a devastating impact on the food supply. Zambia has not been spared by this scourge and currently both commercial and subsistence farmers in Zambia use a manual process to monitor crops for the presence of the fall army worm. Manual monitoring and observation is time consuming, labour intensive and also results in the delay of applying appropriate pest control measures by farmers. There is need to improve on the methods used in the monitoring of the pest if there is to be a proactive and reactive response by stakeholders to FAW.

This study was therefore aimed at improving on the FAW pest monitoring and early warning based on cloud technology using a Convolutional Neural Network model. The objectives were to (1) develop a model using Convolutional Neural Networks for FAW identification, analysis and classification, (2) develop web and mobile applications integrated with GIS technology based on the developed model in the first objective for FAW identification, analysis, processing and visualization.

A Convolutional Neural Network (CNN) was successfully trained to meet objective number (1). Due to lack of enough training dataset and good computing power, a CNN technique called transfer learning was used, by using Googles pre-trained InceptionV3 model as the underlying model. Objective number (2) was addressed by using the developed model in (1) and build a web and a mobile application for data processing, analysis and visualization.

Initial results showed that it is possible to build a Convolutional Neural Network model for automatic identification and classification of the FAW pest based on images acquired from field traps and use the developed model to build web and mobile application for data analysis and real time early warning of the pest occurrence. The developed system provides a real time early warning of the pest occurrence for farmers and stakeholders that never existed before in the country

Key words: Artificial Intelligence, Convolutional Neural Network, FAW, Mobile application, Web, Early warning

ACKNOWLEDGEMENT

My sincere gratitude goes to the almighty God for the opportunity, strength and courage he gave me to undertake this research. I would also like to acknowledge my supervisor Dr. Jackson Phiri for his continued support, patience, dedication and guidance he gave me in the course of my study. His guidance, commitment and motivation helped me a lot during the course of the research. My acknowledgement also goes to my second supervisor Prof. Phillip .O. Y. Nkunya for his guidance during the research period. Gratitude also goes to the Head of Department in the Department of Computer Sciences, Dr. Mayumbo Nyirenda and the entire Department for the support rendered during the period of doing the research. I also acknowledge my colleague Mr. Simon Chiwamba for working closely with me for the entire period of the research. I would also like to thank the Food Agriculture Organization of the United Nations (FAO) for the necessary monetary resources that made this research possible. Lastly but not the least, I would like to thank the University of Zambia and the Zambia Agriculture Research Institute (ZARI) for the support received during the course of this research.

DEDICATION

I dedicate this dissertation to my wife and my family for the support and encouragement I received from them during the period of study.

Table of Contents

COPYRIGHT	
DECLARATION.....	i
CERTIFICATE OF APPROVAL	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
DEDICATION.....	v
LIST OF FIGURES	x
LIST OF TABLES	xii
ACRONYMS AND ABBREVIATIONS	xiii
CHAPTER ONE	1
INTRODUCTION.....	1
1.1. Background.....	1
1.2. Statement of the Problem.....	2
1.3. Significance of the Study	2
1.4. Aim	3
1.5. Specific objectives	3
1.6. Research Questions	3
1.7. Research Contributions	3
1.8. Organization of the Dissertation	4
1.9. Summary	4
CHAPTER TWO	5
LITERATURE REVIEW	5
2.1. Background work.....	5
2.1.1. Agriculture in Zambia.....	5
2.1.2. Distribution and Spread of the FAW in Africa.....	6
2.1.3. The fall armyworm Life Cycle.....	7
2.1.4. Existing Crop Pest Monitoring Techniques	9
2.1.4.1. Trap Based Pest Monitoring.....	9
2.1.4.2. Image based Pest monitoring	11
2.1.4.3. Manual Pest Monitoring.....	12
2.1.4.4. Mobile Phone Pest Monitoring	13
2.1.5. Machine learning in Agriculture.....	14

2.1.6.	Machine Learning Algorithms	15
2.1.7.	System Development	20
2.1.7.1.	Software development methods	20
2.2.1.	Software Development Tools.....	24
2.3.	Related Works.....	24
2.3.1.	Summary of Gaps in the Literature.....	29
CHAPTER THREE		31
METHODOLOGY		31
3.1.	Introduction.....	31
3.2.	Study setup.....	31
3.3.	Current manual Process	31
3.4.	Conceptual Framework	32
3.5.	Ethical Considerations	33
3.6.	Convolutional Neural Network model Methodology	33
3.6.1.	Data Collection.....	33
3.6.2.	Data Selection and Distribution	35
3.6.3.	Model Training and Testing	36
3.6.4.	Model Deployment	37
3.7.	System Development methodology	39
3.7.1.	Requirements Analysis.....	39
3.7.1.1.	Functional Requirements	39
3.7.1.1.1.	Web Interface.....	39
3.7.1.1.2.	Mobile app	40
3.7.1.2.	Non-Functional Requirements	41
3.7.2.	System Architecture.....	41
3.7.3.	System Behavioral Modelling.....	43
3.7.3.1.	Use Case Diagrams	43
3.7.3.1.1.	Web Interface.....	43
3.7.3.1.2.	Mobile app	45
3.7.3.2.	Use Case Narrations.....	45
3.7.3.2.1.	Web Interface.....	45
3.7.3.2.2.	Mobile app	51
3.7.3.3.	Sequence Diagrams.....	51

3.7.3.3.1. Web Interface.....	52
3.7.3.3.2. Mobile app	60
3.7.3.4. Class Diagram.....	61
3.7.4. Data Models.....	62
3.7.4.1. E-R Diagram.....	62
3.7.5. System Implementation.....	63
3.7.5.1. Web application	63
3.7.5.2. Mobile application	63
3.8. Summary.....	63
CHAPTER FOUR.....	64
RESULTS	64
4.1. Introduction.....	64
4.2. Convolutional Neural Network training	64
4.3. System Prototype development.....	70
2.1.8. 4.3.1. Web Interface	70
4.3.1.1. Sample Dashboard Screen shoots	71
4.3.1.1.1. FAW Expert.....	71
4.3.1.1.2. Farmer.....	78
2.1.9. 4.3.2. Mobile app	80
4.3.2.1. Sample mobile app screen shorts	81
4.3.2.1.1. Login Page	81
4.3.2.1.2. Main Menu.....	82
4.3.2.1.3. Mobile app Dashboard.....	83
4.3.2.1.4. Mobile app spatial data	87
4.4. Sample Alert messages	87
2.1.10. 4.4.1. Email.....	87
2.1.11. 4.4.2. SMS Messages.....	88
4.5. Summary	88
CHAPTER FIVE	89
DISCUSSION AND CONCLUSION	89
5.1. Introduction.....	89
5.2. Discussion.....	89
2.1.12. 5.2. 1. Convolutional Neural Network	89

2.1.13. 5.2.2. Web and mobile application development	90
5.3. Conclusion	91
5.4. Recommendations	92
REFERENCES	93
LIST OF PUBLICATIONS	101
APPENDICES	102
Appendix 1: Flask Server code	102

LIST OF FIGURES

Figure 1: Current FAW distribution in Africa	7
Figure 2: Fall armyworm Lifecycle	8
Figure 3: Sticky Surface trap	10
Figure 4: Bucket Pheromone trap	11
Figure 5: A common ML approach	15
Figure 6: Typical Artificial Neural Network	16
Figure 8: Typical CNN Architecture	17
Figure 9: Illustration of CNN Transfer learning	19
Figure 10: InceptionV3 Architecture	19
Figure 11: Software development life cycle.	21
Figure 12: Waterfall model illustration.....	22
Figure 13: Agile Software development model.....	23
Figure 14: Iterative and increment development model	23
Figure 18: Conceptual Framework	32
Figure 19: Images collected from the (a) field, (b) lab setup and (c) internet respectively [101]	34
Figure 20: Sample of image Augmentation with (i) being the original image	34
Figure 21: FAW Moth Unique Features	35
Figure 22: Illustration of Tensorflow serving using flask	38
Figure 23: Tensorflow serving flask JSON response	39
Figure 24: System Architecture	42
Figure 25: Web app use case diagram	44
Figure 26: Mobile app use case diagram	45
Figure 27: Web login Sequence diagram.....	52
Figure 28: Manage Farmer/Farm/Trap sequence diagram.....	53
Figure 29: Crop/Lure Management sequence diagram.....	54
Figure 30: Manage Province/District/Ward Sequence Diagram	55
Figure 31: Set ward threshold sequence diagram	56
Figure 32: User management Sequence diagram.....	57
Figure 33: Predict and classify image sequence diagram	58
Figure 34: send notifications sequence diagram.....	59
Figure 35: Update farmer details sequence diagram.....	60
Figure 36: Web interface class diagram.....	61
Figure 37: Entity Relationship Diagram	62
Figure 38: CNN model Round 1 training accuracy	65
Figure 39: CNN model round 1 cross entropy.....	66
Figure 40: CNN model Prediction accuracy graph.....	66
Figure 41: CNN model round 2 training accuracy.....	67
Figure 42: CNN model round 2 cross entropy.....	68
Figure 43: CNN Model round 2 prediction accuracy graph	68
Figure 44: CNN model round 3 prediction accuracy graph.....	69
Figure 45: Sample ipms API trap request	71

Figure 46: Sample ipms API trap response.....	71
Figure 47: Sample landing page for both FAW expert and farmers.....	72
Figure 48: Sample FAW expert login page	72
Figure 49: Sample FAW expert dashboard.....	73
Figure 50: Spatial data Google maps	74
Figure 51: Spatial data Provincial distribution	74
Figure 52: User management landing page	75
Figure 53: User registration form	75
Figure 54: Farmer management landing page	76
Figure 55: Farmer registration form	76
Figure 56: View farmer page	77
Figure 57: Trap surveys	77
Figure 58: View survey.....	78
Figure 59: Farmer Dashboard page.....	79
Figure 60: Farmer spatial data page.....	79
Figure 61: Farmer Surveys page	80
Figure 62: App login page	81
Figure 63: Mobile app Main menu	82
Figure 64: Mobile app dashboard monthly stats.....	83
Figure 65: Mobile app dashboard provincial stats.....	84
Figure 66: Mobile app dashboard daily stats	85
Figure 67: Mobile app dashboard top affected farms	86
Figure 68: Mobile app spatial data.....	87
Figure 69: Sample email warning alert message	87
Figure 70: Sample SMS warning alert message	88

LIST OF TABLES

Table 1. Related works.....	29
Table 2. Data distribution	36
Table 3: Login Use case narration.....	46
Table 4: View spatial data use case narration	47
Table 5: Manage Crops/Lures use case narration	47
Table 6: Manage Locations use case narration.....	48
Table 7: Manage ward threshold use case narration	48
Table 8: Manage users use case narration	49
Table 9: Send Notifications use case narration.....	49
Table 10: Predict and classify image use case narration	50
Table 11: Farmer Update personal details use case narration	51
Table 12: CNN model round 1 prediction accuracies	65
Table 13: CNN model round 2 prediction accuracies	67
Table 14: CNN model round 3 prediction accuracies	69

ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
CSS	Cascading Style Sheet
CRUD	Create, Read, Update, Delete
ER	Entity Relation
FAO	Food Agriculture Organization
FAW	fall armyworm
GPU	Graphics Processing Unit
HTML	Hypertext Markup language
IDE	Integrated Development Environment
IPM	Integrated Pest Management
IPMS	Integrated Pest Management System
LAMP	Linux, Apache, MySQL, PHP Stack
NRC	National Registration Card
SMS	Short Message Service
SVM	Support Vector Machine
USSD	Unstructured Supplementary Service Data
ZARI	Zambia Agriculture Research Institute

CHAPTER ONE

INTRODUCTION

1.1. Background

There are a number of factors that can be classified as threats to the African and global food security. These may include but not limited to climate change, droughts, emerging diseases, salty soils, fertilizer dependence and insect pests [1]. Insect threats has been one of the yearly occurrences in many African countries with new pests being introduced on the continent from other regions of the world as stowaways on commercial aircrafts, brought in cargo containers or airplane holds before they are widely dispersed by wind throughout the continent [2]. One of the pests that have affected the African continent from other regions of the world is the fall armyworm (*Spodoptera frugiperda*- J. E. Smith) which was first reported as present on the continent in January 2016. According to [3], subsequent investigations have showed that the pest has been identified in over 40 Sub-Saharan African countries where it has caused extensive damage to crops especially maize. Although other new pests are introduced into the African agricultural environment and pose some degree of risk, FAW possess characteristics that make it more devastating than other insect pests [3]. Some of the characteristics reported in [3] include: (i) It is capable of feeding on over 80 different crop species making it a very destructive crop pest; (ii) It can spread quickly across large geographic areas covering more than 1100km/h; (iii) It can persist throughout the year; (iv) They have no diapause and can breed throughout the year; (v) Female moths can lay between 1500-2000 eggs; (vi) The pest has developed resistance to a number of synthetic pesticides. FAW occurrence has negatively impacted maize yield and the economies in most maize producing countries such as Ghana, Zambia and other countries. Its impact in Africa has been felt at national, continent and household level [4]. According to [4], the potential impact of FAW on the continents maize yield lies between 8.3 and 20.6 million tons per annum of total expected production of 39metric tones per annum and with losses lying between \$2,481million and \$6,187million per annum of total expected value of \$11,590.5million per annum. [4] Further states that FAW directly affects capital costs, through increased labour and increasing cost of production due to costs of control.

Owing to the characteristics of FAW, it presents a big challenge of controlling this insect pest. To apply pesticides, the presence of the pest in an area has to be reported. However the challenge is coming up with a method that will act as a near real-time early FAW warning system. Currently one of the methods being used in Africa is the use of pheromone traps to lure the male FAW moth. The field traps are monitored after a number of days and the trapped moths are counted and recorded on the data collection sheets. The pheromone moth trap counting is tedious, labour intensive, time consuming and expensive owing to the field visits and manual counting and recording of the moth. There is an urgent need to improve on the methods used in the monitoring of the pest if there is to be a proactive and reactive response to FAW attack. One of the methods that can be used is the automation of the insect identification process targeting mainly the FAW Moth. An introduction of fast systems integrated with modern image processing and analysis algorithms to accelerate the data collection process is needed [5].

1.2. Statement of the Problem

Early FAW pest occurrence warning and monitoring is usually hindered by the collection of data from field traps and processing that data. Currently the monitoring process being applied in Zambia and other African countries is the traditional pest monitoring process of manual observations and data collection and the use of Pheromone traps. The pheromone traps are deployed in the field containing a female hormone which attracts the male moth. Once at the trap, the moth is trapped and killed. Entomologists conduct field visits once in a week or so to identify, classify and count the trapped FAW insects and record the data onto data collection sheets. After the identifying and counting, the traps are emptied and the collected data is used to generate reports.

The pheromone trap method of FAW monitoring presents some challenges such as the field visits that are labour intensive, time consuming and expensive as it requires the engagement of field experts such as entomologists. In addition the longer the monitoring of the pest prevalence takes the longer the reaction takes to control the pest.

1.3. Significance of the Study

The damaging effects of the FAW has negatively impacted the country's food security significantly. The current method being used is the use of pheromone traps to lure the male FAW moth. The field traps are monitored after a number of days and the trapped moths are counted and recorded on the data collection sheets. The pheromone moth trap counting is tedious, labour

intensive, time consuming and expensive owing to the field visits and manual counting and recording of the moth. Developing an automated early warning system removes the error prone, manual counting of FAW from traps hence providing a near real time pest occurrence. Furthermore it will provide a knowledge platform that would help farmers and other stakeholders in making viable and informed decisions.

1.4. Aim

The overall aim of the study is to address current challenges that entomologists are facing when using the pheromone traps as a way of monitoring the occurrence of FAW pests in Zambia by developing an Artificial Intelligence model for automatic FAW identification and classification for the early warning and monitoring system.

1.5. Specific objectives

1. Develop a Convolutional Neural Network model for FAW identification, analysis and classification.
2. Develop web and mobile applications integrated with GIS technology for FAW identification, analysis, processing and visualization.

1.6. Research Questions

1. Can we develop a Convolutional Neural Network model for FAW identification, analysis and classification?
2. Can we use the developed CNN model in (1) to develop a web and mobile application integrated with GIS technology for FAW identification, analysis, processing and visualization?

1.7. Research Contributions

The study is beneficial to all stakeholders affected by the fall Armyworm which may include farmers, the Government, Research institutions and the community at large. The benefits of the study includes but not limited to the provision of an early warning system using readily available tools that will reduce on labour intensive, time consuming, expensive manual counting and monitoring of the FAW pest in the field. The proposed system if deployed, will provide effortless monitoring of farm areas using fewer resources. The system will provide an early warning system in Zambia that will enable all stakeholders involved to take appropriate counter measures. The

system will also provide a platform where would be users can generate statistics on the fall armyworm pest that would enable them to make viable and informed decisions. The system will provide critical data to researchers needed to carry out research on the fall armyworm pest. The study will also contribute significantly to the academic body of knowledge through the published research papers. The published papers are as listed under the List of Publications section of the dissertation.

1.8. Organization of the Dissertation

This dissertation includes five chapters with each chapter addressing a specific area of concern. Chapter 1 gives the introduction of the research and outlines the problem to be addressed, aim of the study and research questions.

Chapter 2 gives relevant literature reviewed to supplement the research problem. It gives the life cycle of the FAW, gives an overview of several machine learning techniques that can be applied to the research and gives an overview of software development methods. It closes with the related works relevant to this research.

Chapter 3 covers the research methodology used which are basically methods used to train the CNN model and implement the web and mobile application.

Chapter 4 gives the results of training the CNN model and the implementation of the web and mobile app. Chapter 5 discusses the findings. Chapter 6 draws conclusions and provides recommendations of the study.

1.9. Summary

The chapter introduces the research by providing the background, stating the problem, providing the aim of the study, providing the objectives of the study, providing the research questions and the significance of the study. The following chapter will focus on evaluating the significance of the study in relation to literature reviewed and related work that was done responding to similar challenges presented by the FAW and other pests.

CHAPTER TWO

LITERATURE REVIEW

2.1. Background work

2.1.1. Agriculture in Zambia

The Agriculture sector is one of the most important economic sectors in Zambia and plays a big role in the economic growth of the country. The principal cash crop is maize which is also the main staple crop. According to [7], per capita consumption of maize in Zambia is estimated at 105 kilograms annually, most of which is ground into meal/flour and consumed as stiff porridge, or fermented for beer, with by-products used as livestock feed. Other important crops include soybean, cotton, sugarcane, sunflower, wheat, sorghum, pearl millet, cassava, tobacco and various vegetables and fruit crops [7]. The sector is positioned as one of the driving engines of poverty reduction as it has the potential to increase food security and increase income generation. According to [8], the sector supports the livelihood of approximately 70% of the Zambian population. It contributes about 19% of the Gross Domestic Product (GDP) and contributes about three quarters to the labour force most of which is from rural areas [9]. Shows that the Agriculture sector has the potential to drive the country's economy forward but the sector faces challenges that needs to be addressed [8][9]. Of the yearly occurring factors that affect the sector are crop pests with the newest being the fall armyworm (FAW) which was just recently introduced and reported in Africa in 2016 [3].

Insect pests affect the world agriculture sector every year destroying crops at a large scale thereby reducing yields. Among insect pests that affects the Zambian agriculture sector includes but not limited to larger grain borer, stem borer, locusts and fall armyworm. Larger grain borers mostly affect stored maize, dried cassava and yam by eating its way into the grain leaving an empty shell [10]. The stem borers are damaging pests of maize and sorghum plants. They are most abundant in the drier and hotter seasons of the year [11]. Locusts are one of the pests that affects the Zambian farming sector with the most recent reports being the red locusts and the African migratory Locusts. The most notorious of all the insects is the fall armyworm as it can persist throughout the year [12].

2.1.2. Distribution and Spread of the FAW in Africa

One of the annual nuisances in most African countries are insect pests with new pests being introduced on the continent from other regions of the world as stowaways on commercial aircrafts, brought in cargo containers or airplane holds before they are widely dispersed by wind throughout the continent [2]. The most recent pest which was first reported in 2016 in West and Central Africa is the fall armyworm- FAW-*Spodoptera frugiperda*. Before its introduction in Africa, the pest has been a native of the tropical and subtropical regions of the Americas only but since its introduction in Africa, subsequent investigations have showed that the pest has been identified in over 30 African countries where it has caused extensive damage to crops especially maize fields [2] [3] [4] as shown by Figure 1. Its occurrence has negatively impacted maize yield and the economies in most maize producing countries such as Ghana, Zambia and other African countries. Its impact in Africa has been felt at national, continent and household level [2] [4].

The potential impact of FAW on the continents maize yield lies between 8.3 and 20.6 million tons per annum of total expected production of 39metric tones per annum and with losses lying between \$2,481million and \$6,187million per annum of total expected value of \$11,590.5million per annum [2][4]. Further FAW directly affects capital costs, through increased labour and increasing cost of production due to costs of control [4].

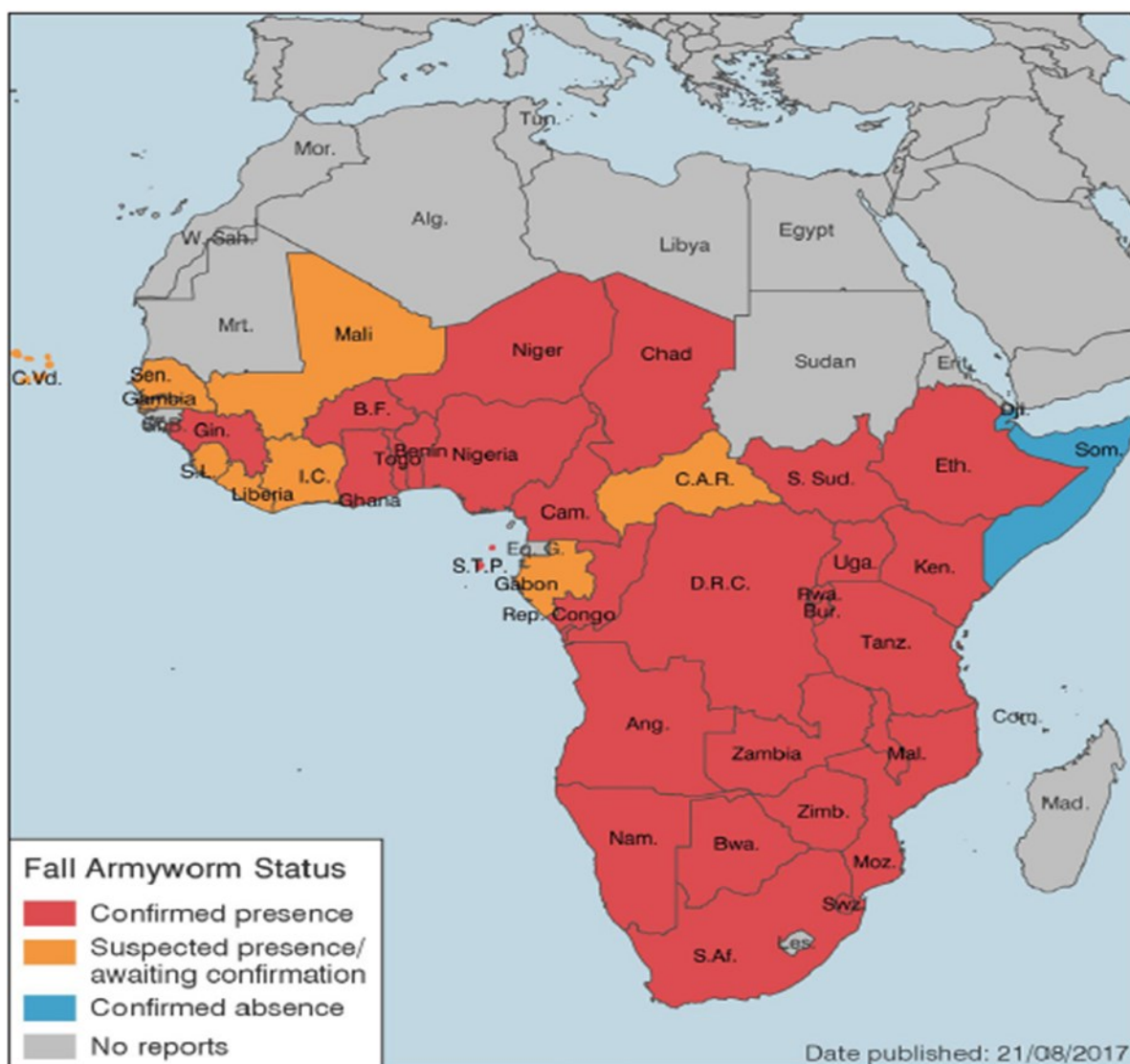


Figure 1: Current FAW distribution in Africa

Source: <https://www.cabi.org/Uploads/isc/Dfid%20Faw%20Inception%20Report28apr2017final.pdf>

2.1.3. The fall armyworm Life Cycle

Even though new pests are introduced every year on the African continent and pose some degree of risk, the fall armyworm possesses characteristics that makes it more devastating than other insects. It is capable of feeding on over 80 different crop species making it the most damaging crop pest, it can spread quickly across large geographic areas just like other moths in the genus *Spodoptera* and It can persist throughout the year [3]. Another of the devastating characteristics is the period it takes to reproduce which takes roughly 30 days beginning with egg laying.

- a) Egg stage: Approximately 1500 to 2000 eggs are deposited in layers with most eggs being spread over a single layer attached to foliage. After 3-5 days, the eggs hatch into larvae.
- b) Larval Stage: The larvae will feed for about 14 days in warm summers and for about 30 days in winter temperatures. The larval stage has six instars with the larvae having different characteristics at each instar. At this stage, the caterpillars superficially feed on the underside of leaves creating semitransparent patches on the leaves called windows. After maturing, the caterpillars drop to the ground where they feed on leaf margins of the crop.
- c) Pupa Stage: The Pupa stage takes place in the soil for about 8 to 9 days in the summer and sometimes 20 to 30 days in winter.
- d) Adult Stage: After Pupa development which is 8 to 9 days, the adult moth emerges from the soil or cocoon to start a new life cycle. The duration of the adult moth is on average 10 days with the male and female moth having different distinguishing characteristics [3][4][13][14].

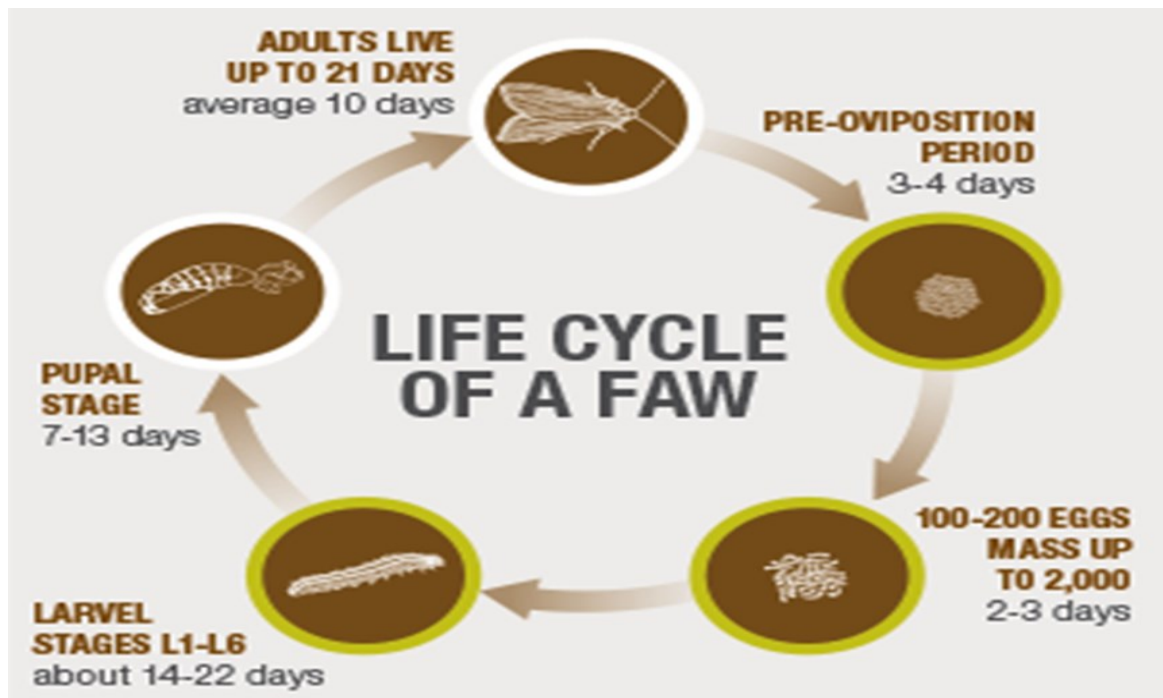


Figure 2: Fall armyworm Lifecycle

Source: <https://www.syngentaseedcare.com/news/pests/march-fall-armyworm>

Figure 2 summarizes the fall armyworm Lifecycle showing different characteristics at each stage of development. Monitoring of the fall armyworm pest can be done at any stage of its lifecycle depending on the study objectives but this study used the adult stage as a key parameter to capture

the images of the moth. The adult stage was the main focus of this study as we were riding on the fact that we need to capture the image of the male moth by luring it with a female hormone lure. Hence the moth need to fly into the trap for us to get its image for identification and classification and at the same time the moth is trapped in the trap where it eventually dies. Monitoring of the FAW presents the chicken egg problem as monitoring at larvae stage leaves the adult moth to breed and monitoring at adult stage allows the larvae to destroy the crops.

2.1.4. Existing Crop Pest Monitoring Techniques

To combat the devastation that insect pests cause, governments and farmers around the world in areas affected by the fall armyworm and other insect pests have used different methods of controlling and monitoring the pests. Some of the methods that have been used include but not limited to trapping, image based monitoring, manual monitoring, mobile phone monitoring, and use of the traditional pesticides and so on.

2.1.4.1. Trap Based Pest Monitoring

Insects can have both positive and negative impact on the livelihood of human beings hence the reason why they receive a lot of scientific attention. For instance bees play an important role in the pollination of plants hence one of the reasons why we have food from plants. Some insects are a nuisance to humans like the FAW which has had a negative impact on the supply of food. The negative impact of insects has led to scientists developing methods that can be used to monitor their presence. One of the methods that has been used for several years is the traditional method of traps [15]. Traps that are developed for capturing insects are varied according to the purpose for trapping, the targeted insect and the habitat in which they are used [16]. Traps are used for the general survey of insect diversity or for the detection of new invasions of insects in an area and may be used as direct control measures e.g. mass trapping, perimeter trapping or can be used for suppressing population buildup of insects [16] [17]. There are many trap types which include interception traps commonly used for faunal surveys in ecological studies which are suspended nets with an invagination along the top leading to a collecting funnel, sticky traps commonly used for faunal surveys in agricultural studies which are either panels, cylinders or spheres covered with sticky materials that retain insects that fly onto the panel [16]. One of the commonly used traps in the monitoring and controlling of the FAW insect are pheromone traps. Pheromone traps use pheromones to attract male insects [18]. A pheromone is a chemical secreted by (usually) a female

insect to attract males for mating which can travel by air very long distances and hence are very useful for monitoring insect presence [3]. The pheromone traps are useful for detection of early pest infestations, definition of areas of pest infestations, tracking the buildup of pest population and assisting in the decision making process for pest management [3][4].

Pheromone traps are of different designs manufactured by different manufacturers. Figure. 3 uses a sticky surface coated with a special non-drying glue which retains the insect once it flies onto it. The sticky trap has been used in Malaysia and found to be more effective in capturing small moths [19].



Figure 3: Sticky Surface trap

Source: <http://web.entomology.cornell.edu/shelton/swede-midge/images/DETECTION/trap2large.jpg>

The Funnel pheromone trap shown in Figure. 4 has a funnel section and a bucket. It has a pheromone dispenser holder which is located on top of the funnel under an umbrella type cover. Insects attracted to the trap fly around the pheromone dispenser until exhausted and then fall into the trap as shown. Insects find it impossible to fly out of the bucket and are eventually killed by the pesticide inside the bucket. The effectiveness of the bucket trap was tested in the capturing of male moths in corn fields [20].



Figure 4: Bucket Pheromone trap

Source: <https://4.imimg.com/data4/BY/BP/MY-5420398/pheromone-trap-250x250.jpg>

The trap method of insect identification and classification is the traditional method and has been used from time immemorial and has been found to be effective in the monitoring, mass trapping and control of insects. However good this method is, it has disadvantages such as it requires regular monitoring of and counting of insects. The traps if full needs to be emptied which requires going to the fields. The method if used as a monitoring and early warning system takes time for the warning to be passed to the relevant stakeholders hence delaying the decision making process. An automation of the counting process by the use of the latest technology we can go a long way in speeding up the early warning process at the same time speeding up the decision making process.

2.1.4.2. Image based Pest monitoring

To maintain the diversity of species within the ecosystem, various scientists in the world including computer scientists have gained an interest in the field of biodiversity and out of 1.3 million known species on earth, insects make up more than two thirds of these known species. For many years now, there has been different kinds of interactions between humans and insects and these interactions requires that insects are observed in close contact either by taking them to the lab or capturing their images. Several attempts have been made to create a method to perform insect identification accurately mainly because accurate insect identification requires great knowledge and experience on entomology [21]. A shortage of entomologists and the labour intensive process of collecting insect samples has led scientists trying to come up with methods that will do automatic insect identification and classifications such as those methods that use images of insects

captured in the field or lab. Several computer based techniques have been used to come up with classification and identification methods for different insects.

To accelerate the processing of image identification system, there is a clear need to introduce fast systems integrated with modern image processing and analysis algorithms. Computer scientists have tried to speed up the identification process by applying computer vision and pattern recognition techniques on models that infer semantically related visual (SRV) attributes from low level visual features of insects [22]. Other scientists have tried to investigate the possibility of using Artificial Neural Networks (ANN) as a tool to classify apple orchard pests using digital image analysis [23]. They have also tried to use ANN to classify and identify malaria causing anopheles mosquitos using the mosquito's ribosomal DNA string [24]. Others have conducted experiments and determined that there are a number of factors that affect the classification and identification process such as insect colour, shape features and texture features [25]. Others have tried to bridge the gap that exists between taxonomists and non-specialists by creating an automated pattern recognition system using digital images of female spiders [26].

Other scientists have presented computer-vision based systems that tried to classify insects as harmful or non-harmful to crops and at the same time prove the effectiveness of using computer based classification methods and algorithms in the monitoring of insects in the field and they have speculated that in the future field deployable insect monitoring systems would allow farmers to identify insects that inhabit a specific ecosystem and enable them to choose which biological or pesticide to use as a control measure [27]. Some scientists have tried to find the most effective computer algorithm for classification and identification of insects [28]. A number of factors affect the performance of image based identification systems some of which may include environmental factors i.e. temperature, humidity, image quality and resolution, the diversity of species, features of species such as size, system placement areas and so on. Therefore to come up with a new insect identification and classification system, most of the factors have to be taken into consideration.

2.1.4.3. Manual Pest Monitoring

Manual monitoring entails moving around the farm area inspecting the fields for any signs of damage or presence of pests. This method has been used from time in memorial by mainly small scale farmers around the world. A farmer will go around the field doing observations for crop diseases or pests and if any pest or disease that is able to cause damage to the crops is detected,

the farmer then would take the necessary action of prevention which is usually the use of pesticides. FAW usually destroys crops at an early stage of the crops lifecycle hence manual monitoring has a big impact on the yield as its usually late by the time the farmer gets to notice the damage the disease has done or presence of the pest. Not only is manual monitoring ineffective but it is also time consuming as it requires the farmer to move around the field where sometimes the fields are very large[2][3][4].

2.1.4.4. Mobile Phone Pest Monitoring

This involves the use of mobile phone applications in the monitoring and controlling of pests and diseases. Most of the applications are readily and freely available with sufficient information to provide proper pest disease diagnosis or used as crop data capturing tools. The apps have been used by farmers around the world and aid in the fight against pests and diseases [29]. FAMEWS a field scouting android application developed by the Food Agriculture Organization of the United Nations (FAO) is one such app [29]. The app is developed to aid farmers using pheromone traps to monitor FAW. The app is used every time the fields are scouted and used to collect farm and crop data, scouting data and trap data. The app provides real-time overviews with maps and analytics of FAW occurrence at global, country and sub-country levels [29]. In India a company called Progressive Environmental & Agricultural Technologies (PEAT) has rolled out a mobile app called Plantix which is used by farmers to identify pests and diseases and provides solutions for each particular diagnosis. The app provides a diagnosis based on the image uploaded by the farmer and steps a farmer needs to take to counter that disease or pest. In addition, Plantix provides disease preventive information for the next cropping season and also the biological treatments that the farmers need for pest and disease control [30].

In Zambia, the United Nations World Food Programme in 2016 piloted a mobile app called “Maano Virtual Farmers Market” which enables farmers to connect with buyers of the farm products. The app offers a platform where farmers are able to offer their produce and buyers can bid for the most competitive prices [31]. “Lima Links” is another app developed by Zambians entrepreneurs that links farmers to farming input suppliers and buyers. The app also provides information on how inputs can be used by a farmer on their farming areas. They provide the services to farmers on android devices and on Unstructured Supplementary Service Data (USSD) for farmers without smartphones [32]. With advancements in technology around the world,

Zambian farmers are also being equipped to move with technological advancements to solve farming challenges they face.

2.1.5. Machine learning in Agriculture

Machine learning is being applied in agriculture in what is termed as Smart Farming to tackle various challenges such as productivity, environmental impact, food security and sustainability [33]. Smart Farming enables the protection of natural ecosystems by means of using sustainable farming procedures [34]. Smart farming also provides data that enables a better understanding of the farming environment through interaction with dynamic crops, soil and weather conditions which in turn leads to more accurate and faster decision making by the relevant stakeholders [33][34]. Application of machine learning in agriculture include but not limited to plant disease detection, farming area classifications, plant recognitions, fruit counting, soil and vegetation mapping, identification of seeds and reorganization of species, irrigation, pest detection and management and weed detection, Greenhouse monitoring and so on [34][35]. The techniques of machine learning are being developed and applied in agriculture with the hope that it will tackle many agriculture challenges while reducing on the cost of farming.

Machine learning has emerged among other novel technologies such as big data and high performance computing as a technology that creates new opportunities to unravel, quantify and understand the data intensive operations in many areas [33]. Machine learning among other definitions is the field of science that gives computers the ability to learn without being strictly programmed. It is a field of computer science that evolved from the study of pattern recognition and computation learning theory in AI. It is the learning and building of algorithms that can learn from and make predictions on datasets. Machine learning involves a learning process whose objective is to learn from experiences to perform a certain task. Models are generated by the extraction of features or attributes from a dataset of examples (training data). The performance of the generated model in a specific task is measured by a performance metric that is improved with experience over time. The generated model can be used to perform classification, predications or clustering from new dataset (testing dataset) they have not encountered before [33][34][35][36]. Figure 5 shows a common Machine learning approach.

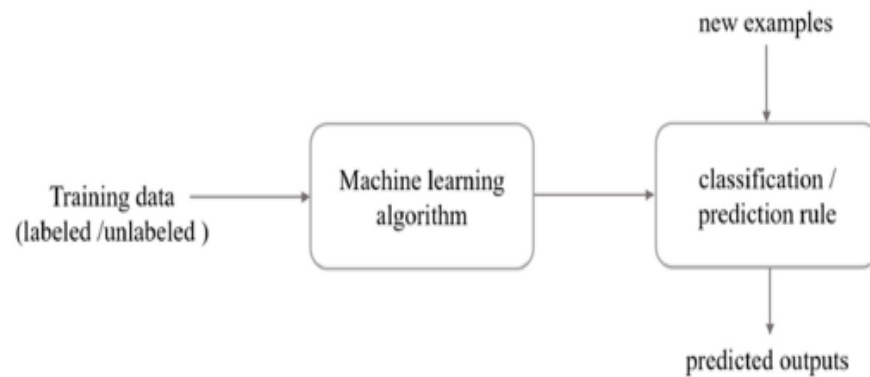


Figure 5: A common ML approach

Source: [37]

Depending on the learning type or models, machine learning tasks are divided into categories which include supervised, unsupervised and reinforced learning. In supervised learning, the model learns by being supervised with categorized input data. It is fed with input of examples and told what the output is likely to be. In unsupervised learning, the model learns by discovering patterns in the supplied input data and creating clusters. Unlike in supervised learning, the model in unsupervised learning is given a dataset that is not grouped and the model is able to give a response when supplied with new input data and the model generates a new class if it cannot find a class for the new input. In reinforced learning the model tries to learn using less input data. The model is not told what the output is likely to be but gets feedback from the environment or critics on whether it has achieved its goal or not. Reinforcement learning sits between supervised and unsupervised learning [37][38][39][40]. The choice of task and algorithm to use depends on the type and complexity of the problem that needs to be solved [41][42][43]. In this research supervised learning has been used to train a model to identify fall armyworm moths. The continuing trend of using machine learning in the Zambian agriculture sector and the world at large to solve challenges that the field faces gives hope that more and more challenges are likely to be solved hence reducing on the cost of farming and increasing food security.

2.1.6. Machine Learning Algorithms

There are many machine learning algorithms which can be applied to almost any challenge that the agricultural field may face or any data problem that one can face. Among the most common includes but not limited to Artificial Neural Networks, Convolutional Neural Networks.

Work on Artificial neural networks from its inception has been motivated by the Human brain. The brain is a highly complex, nonlinear, and parallel computer. It has the capability to organize its structural constituents, known as neurons, so as to perform certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer in existence today [41] [42] [43]. Artificial neural networks have the ability to adapt to new inputs and any changes that may occur in the surrounding environment [44]. Artificial neural networks are made up of processing units called neurons which try to imitate the structure and behavior of a natural neuron [45] [46]. Artificial neurons are made up of inputs, outputs and activation functions which determines the activation of the neuron. Artificial neuron inputs are multiplied by weights which are then computed by the activation function to determine the neurons activation function and the output function computes its output signal. The output of the neuron is different depending on the weights. Given certain inputs, it is possible to get the desired output by the adjustment of weights. However, when the network is made up of many neurons, it becomes complex to adjust the weights and obtain the desired output hence algorithms exists that can adjust weights. The process of adjusting an artificial neural networks weights to get the desired output is what is called ANN training or learning. Figure 6 shows a typical ANN structure with three inputs X_1, X_2, X_3 and two outputs Y_1 and Y_2 with the interconnections showing signals from one neuron to another [45][46][47][48].

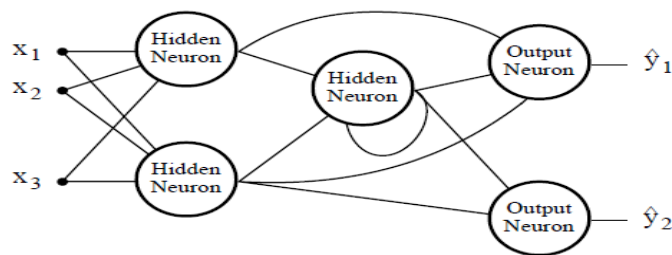


Figure 6: Typical Artificial Neural Network

Source: [45]

Artificial neural networks have been applied in solving challenges that the agricultural field faces. [49] Used artificial neural networks to predict crop yield using soil parameters such as type of soil, PH levels and atmospheric parameters such as rainfall, temperature and humidity. A feed forward ANN network was used and feed with the necessary crop parameter inputs which predicted the yield for the crop. While [50] carried a review that presented the use of ANN in predictions of crop

yields using various factors. Others have tried to use artificial neural networks to model Agricultural production using rainfall. Previous year's rainfall data was used to predict the current year's rainfall and in turn predict how much agricultural yield the country will get [51]. While ANNs have been applied to prediction of agriculture trends, it has also been applied in the classification domain and has proved to perform very well [52][53][54][55]. In as much as ANNs have proved to work well in other domains, they prove a bit of a challenge when it comes to image analysis and prediction as they require manual feature extraction.

Convolutional Neural Networks is the novel and most popular machine learning technology that has successfully been applied in computer vision and image classification. They are made up of a class of feedforward Artificial Neural Networks sharing weights [56][57][58][59][60]. The basic idea behind CNN is the local understanding of an image. They combine three architectural ideas: local understanding of the image, shared weights and subsampling to ensure some degree of shift and distortion invariance [61]. The main architectural benefits of CNN is few parameters which in turn improves the learning time and a reduction in the amount of training data however enough training data increases the prediction accuracy. They are made up of Convolutional layers which act as feature extractors from input images, pooling layers which perform sampling operations on the dimensions of the input image thereby reducing the dimensions and fully connected layers which acts as classifiers by using the high level features learned [60][61][62][63]. Figure 8 shows a typical Convolutional Neural Network Architecture where several convolutions are applied creating different representations of training datasets.

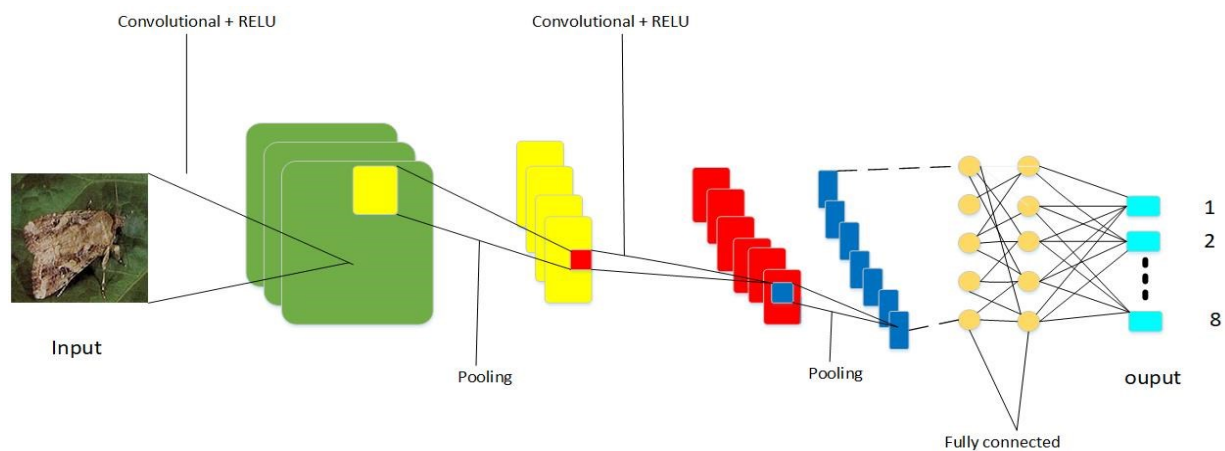


Figure 7: Typical CNN Architecture

Source: [70]

Traditionally classification problems were solved by first manually extracting image features and then serving the features as input to a trainable classifier which meant that the accuracy of the task was dependent on the extracted features. This limited the use of the manual approach [60] [63]. Deep learning algorithms particularly Convolutional Neural Networks have proved to overcome the challenges of manual feature extraction but the increase in the complexity and size of the networks meant that a huge amount of training dataset is required which brings in the need for good computing power [64]. Training a good CNN from scratch requires a huge dataset and good computational resources. Fortunately one can use a technique called transfer learning to solve a classification problem. Transfer learning is the process of using a CNN that has been trained on a generic image classification task to solve a more specialized classification task. It is the use of a pre-trained model on another task as the starting point to another model of a different task. It falls into two categories: (i) Feature extraction which is the extraction of meaningful features from new samples by using representations learned by the pre-trained model in the previous work. This involves the training of a Top layer (classification layer) which is usually a Softmax function which is the output function of the last layer in neural networks whose purpose is to turn the score produced by the network into values that can be interpreted by humans. The Softmax function is defined as follows [64]. The function is given by: $\sigma: R^n \rightarrow \text{int}(\Delta^{n-1})$

$$\sigma(z) = \frac{\exp(\lambda z_i)}{\sum_{j=1}^n \exp(\lambda z_j)}, \quad 1 \leq i \leq n \quad \text{Equation (I)}$$

When $\lambda = 1$, (1) is called the standard Softmax function.

(ii) Fine tuning which is trying to unfreeze a few layers from the pre-trained model and training them together with the new classification layer [65][66][67]. Figure. 9 shows an example of transfer learning architecture. There exists a number of pre-trained models which have successfully been used in recent years in transfer learning to solve classification problems such as ResNet50, inceptionV3, MobileNetV3 and VGG16. The models have been trained on huge datasets giving them the ability to generalize to new datasets when used in transfer learning [64]. Some of the models such as the InceptionV3 (Figure. 10) have been used in many image classification problems and have shown to have attained accuracies greater than 78.1% [68] [69]. The inceptionV3 was developed and trained using a Deep Learning framework called Tensorflow which provides several libraries for numerical computations [66] [69] [70] [71]. In addition, Tensorflow provides

a web based dashboard for visualizing the ongoing training called Tensorboard and makes it easy to deploy the trained model using what is called Tensorflow serving. In this research we are using the Feature extraction part of transfer learning since the inceptionV3 model has been trained on a large dataset enough for it to generalize to our datasets. Of course Fine tuning will be the best option to get better and higher accuracies.

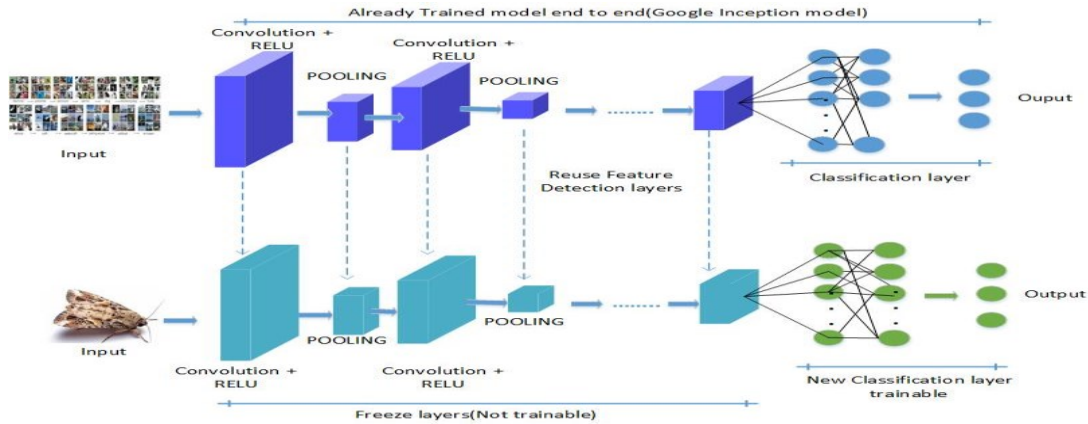


Figure 8: Illustration of CNN Transfer learning

Source: [70]

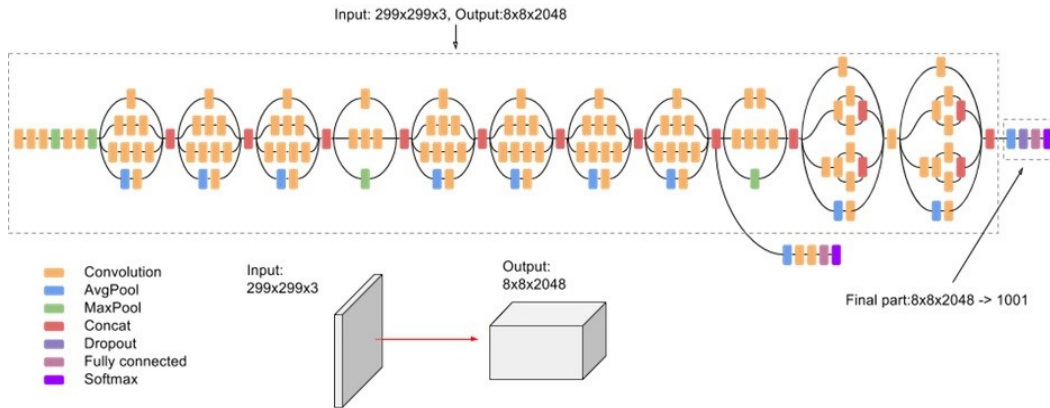


Figure 9: InceptionV3 Architecture

Source: [70]

The success that Convolutional Neural Networks have enjoyed in the classification problem domain can be attributed to a number of factors which may include but not limited to automatic feature detection, higher accuracies obtained, availability of good computing resources such as GPUs and availability of large datasets. Convolutional Neural Networks have successfully been used in many areas of agriculture in what is termed as Smart Farming. Some of the areas where

CNNs have been used in agriculture include crop type classification, plant disease detection and recognition and so on [60]. In [67], the authors developed a model for predicting skin cancer cells by image classification using CNN and obtained more than 85% prediction accuracy and [72] tried to automate the classification of ovarian cancer using CNN to aid pathologists in cancer diagnosis. They used CNN based on AlexNet to predict ovarian cancer from cytological images. They obtained an accuracy of 78.20% by using augmented datasets. [73] Presented a self-operating stretcher developed using CNN transfer learning that solves the problem of transporting patients from long distances or short distances. The stretcher is trained to detect and identify the objects in its path. [66] [72] [73] showed that CNN are also applicable in medicine and are helping to solve many challenges that the field faces. [74] Applied CNN to the automatic classification of mosquitoes in the field. They used 4,056 mosquito images as training dataset and obtained classification accuracy of 82%. [75] Presented a method and the performance of the model for detection and classification of insects on sticky traps using CNN and obtained an average accuracy of 87.4%. [76] Applied CNNs in the identification of bacteria using 3D microscopy datasets. They tried to use CNNs to distinguish bacteria from non-bacterial objects from images obtained from zebrafish intestines using a microscopy. They deduced that CNNs are as accurate as human experts. The application of CNNs is broader and wide and results have shown that they are as accurate as they can be provided the correct training is done.

2.1.7. System Development

A trained Convolutional Neural Network model can be used in many ways in production and one of the ways it can be used is by integrating it into developed software products. The development of the software product goes through many stages and makes use of so many software development tools to successfully develop a good product and within the required time frame.

2.1.7.1. Software development methods

These are basically various systematic steps that are followed in the life of a software product from development stage to maintenance stage ensuring that all user requirements are met and the product is of good quality and is produced within the budget. In its life cycle, a software product goes through a number of stages before it's a fully functional system. These stages include but not limited to [77] [78]:

- i. **Requirements Gathering:** which is a stage where an understanding of the required software product is undertaken by meeting with the software owners and getting all the required system requirements. At this stage developers have to understand the scope of the work involved to make sure that all system requirements are met [77] [78] [79] [80].
- ii. **System design:** This is the next phase after most of the requirements are understood. During this stage, system designers make high level designs of the system while discussing the technical details of the system with stakeholders and decides on the technologies that will be used to deliver the product to the client [77] [78] [79] [80].
- iii. **System Development:** At this stage, the implementation of the gathered requirements is done.
- iv. **System testing:** This is the stage where the software is checked for any possible errors and making sure that the implemented product meets the requirements gathered during the requirements gathering phase [77] [78] [79] [80].
- v. **Deployment and Maintenance:** During this stage the system is delivered to the client and setup at the client's setup machines. After deployment, the software product is supported until at the point where the client is comfortable with the product delivered by the developer [77] [78] [79] [80].

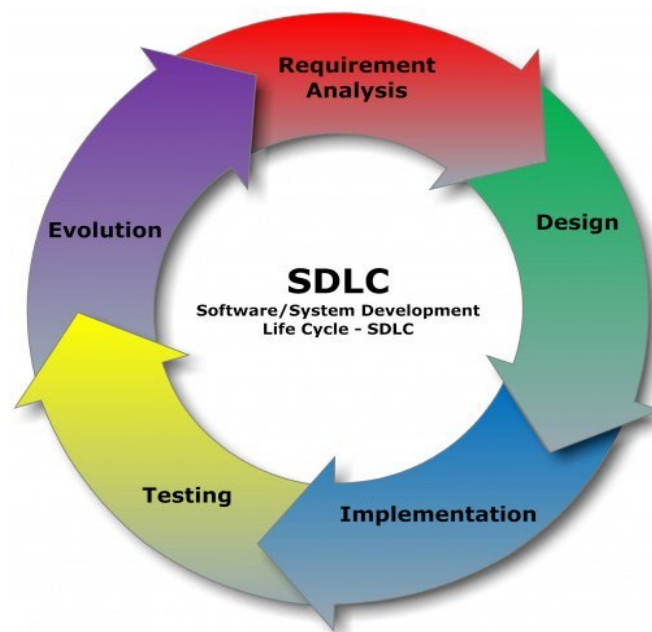


Figure 10: Software development life cycle.

Source: [83]

To meet the requirements of the proposed software product and make sure that the product is delivered on time, there is need to choose a proper software development model that meets the requirements. A software development life cycle model can be defined as a tool that is used to better develop the software product [77] [81] [82] [83]. Therefore an understanding of the available models is imperative as it gives an understanding of the advantages and disadvantages of each model in order to properly choose the suitable model for the project. Some of the models that can be used include but not limited to:

- a. Waterfall model which is a linear sequential process where every stage starts only after the previous has finished. The model does not give an opportunity to go back to previous stages in case of any changes in the requirements hence this means that the product owners feedback can only be included after the project has completed. This model works well for small scale projects where requirements are clear and detailed planning can be drafted for the project [77] [81] [82] [83].

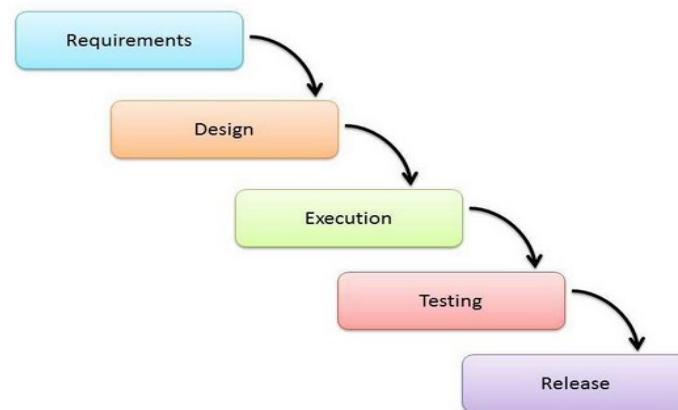


Figure 11: Waterfall model illustration

Source: [83]

- b. Prototyping Model which is the process of creating prototypes (incomplete versions) of the software product being developed which includes the critical components of the system. This model helps in refining the specifications since it acts as a baseline for communication between the development team and the project owner. The model takes many types such as: Throwaway prototyping where the prototype are used only as demo systems and do not become part of the final system, Evolutionary prototyping where prototypes evolve into the final system through incorporation of the new requirements, Incremental prototyping where the software product is built from separate prototypes and Extreme prototyping

where the prototypes are presented in phases until the final system is built [77] [81] [82] [83].

- c. Agile model which is an iterative and incremental model, where the software product evolve through collaboration between cross-functional teams [77] [81] [82] [83].

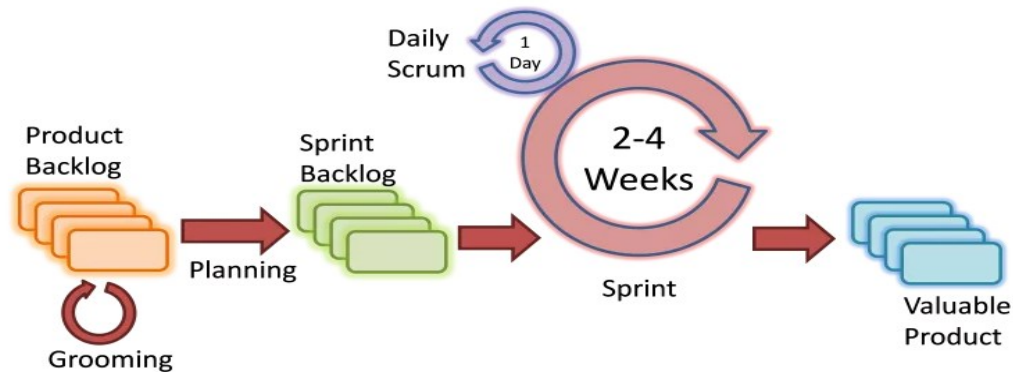


Figure 12: Agile Software development model

Source: [83]

- d. Iterative and Incremental development model enables the building of a software product following well defined iterative steps. An initial model is built based on initial specifications and then the multiple usable versions of the product called vertical increments are built from the initial version and each increment builds upon the previous version by adding user-visible functionality by analyzing any errors or omissions in the requirements to be revealed long before the system is handed over. Any new requirement that comes is reviewed at the end of each iteration [77] [81] [82] [83].

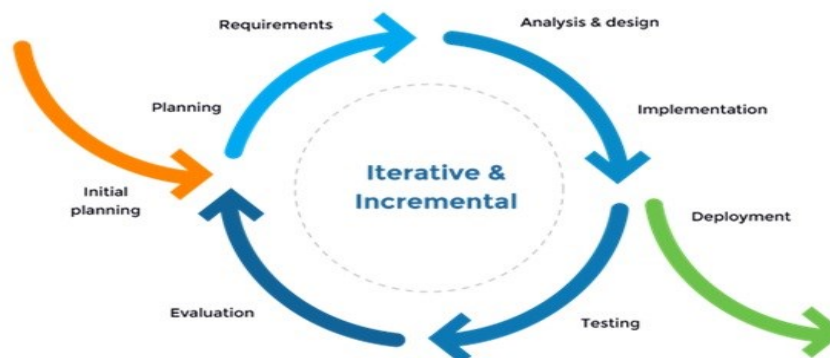


Figure 13: Iterative and increment development model

Source: [83]

After a comprehensive review of the software development phases and models to best chose the best model, this study used a combination of prototyping and Iterative and incremental software development which means new requirements where fused in as they come.

2.2.1. Software Development Tools

Diversity in the software being created and the reason for creating the software can lead to misunderstanding and disappointment if the software developed is not properly created [84] [85] hence the need to use proper software development tools. Tools that are used in the development and support of a software product is what is referred to as software development tools. They are computer programs that are used to create, debug, maintain or support software programs and applications [86]. They range from programming languages, software development IDEs, software versioning tools, and database management tools, design software tools, modeling software, encoding software, management software, and reverse engineering tools and so on[86] [87]. A review of the tools used in this study is imperative to give an understanding of how the products in the study were developed. The main programming language is PHP-Hypertext Processor. It is a server side scripting language which is often used to develop static websites or dynamic websites or web application. Data from the database can be retrieved using PHP and rendered using HTML in a web browser [88]. For rapid system development and prototyping, the study made use of a PHP framework called Yii which is an open source, object-oriented, component based MVC web application framework [89]. The framework is developed in PHP with HTML and CSS as defining elements. HTML-Hypertext Markup language is used for the purpose of defining elements on a web page while CSS- Cascading style sheets are used to define how the elements in HTML will be rendered. To provide some interactive effects, JavaScript is used in the framework. JavaScript is an object oriented programming language that is used for data manipulation [88] [89] [90]. The backend database is MySQL open source relational database management system which is used mainly to provide an interface where data can be stored, deleted and manipulated.

2.3. Related Works

Several attempts have been made to solve the challenge of various crop pests in the world and each research has adopted methods and technologies that best suits the pest under investigation. Owing to the similar nature of the problem of crop pests, most of the research studies adopt almost similar

technologies which means that subsequent research studies can build on any previous work that has been done by filling in the gaps that the previous research has done or start afresh using a new technology or same technology but different requirements. Table 1. Give the reviewed related research studies on fall armyworm monitoring and pest monitoring as a whole. The table gives a brief of the research in question, possible weaknesses, possible gaps that this research can build on. Existing work has been done proposing solutions that focuses on the fall armyworm and other pests.

A researcher in Kenya looked at how challenges that farmers and other stakeholders in Kenya are facing when dealing with the fall armyworm can be solved by proposing an Internet of Things (IoT) and Machine learning solution. The author looked at the manual observation of maize crops for detection of the Fall Armyworm which is usually done weeks after the pest has fully matured and damage to the crop has already started leading to late pest control measures application. An automatic FAW prediction model was developed which exploited the pupa stage of the pest and using soil temperature and humidity collected using sensors placed in the soil as parameters.

A three node forward propagation Artificial Neural Network (BP ANN) algorithm which uses soil temperature and humidity and FAW expert feedback as inputs was developed. AN early warning system that sends SMS messages to farmers and FAW experts to take appropriate measures was developed using the developed algorithm. The system architecture consisted of an IoT system that monitors and retrieves soil parameters using soil sensors specifically DHT11 sensors and an ESP32 development board. The collected soil data is pushed to a locally hosted server which does the prediction and classification of the FAW based on the data from the sensors and pushes SMSs using prediction results [91]. The author used the pupa stage of the life cycle of FAW hence the need to use soil parameters i.e. temperature and humidity to monitor the presence of the pest as pupa development takes place in the soil. The solution being proposed in this research is using the adult stage and utilizing the traps being used by entomologists hence the need to capture FAW images.

The problem of fall armyworms has led researchers to extend existing technologies meant to solve one problem to try and solve the FAW problem. [92] Explored a technology that focuses on the climate-adapted push-pull technology developed for control of cereal stem borers in drier agro-ecologies as a tool for the management of the fall armyworm pest. The tool employs the

intercropping of maize with drought resistance plants such as *Desmodium*. The maize crop was planted in between drought tolerant crops which provided protection to the maize crop by emitting semiochemicals that repels (push) stemborer moths while those released by border crops attract (pull) the stemborer. Data from 250 small scale farmers that had adopted the technology in Kenya, Uganda and Tanzania was randomly collected. Data on the number of fall armyworm larvae on maize, percentage of maize plants damaged by larvae and maize grain yields was collected. In addition, the perception of farmers on the technology was assessed using questionnaires. The authors reported an 82.7% reduction in the number of larvae per plant and an 86.7% reduction in plant damage per observed plot. A significant rise in the maize grain yields was found and the technology was rated significantly superior in the fall armyworm management by farmers. The results demonstrated that existing technology can be applied to the fight against the fall armyworm. The solution being proposed in this research will use the existing technology of pheromone traps in combination with machine learning for the development of an early warning system which is lacking in the authors research.

Other researchers have proposed new solutions to try and solve the problem of insect pests. [93] Have applied Deep learning in the detection and diagnosis of oil seed rape pests. The study proposed a deep learning approach to the detection of oilseed rape pests. The authors used Transfer learning by using a pretrained model trained on the COCO dataset to train several models in order to find the optimal model in terms of memory usage and loss, running speed and so on. They combined meta-architectures such as Faster R-CNN, R-FCN and SSD with feature extractors such as inception model, MobileNet and ResNet. The authors adopted the best model for use in a mobile platform to enable farmers use the program for real time pest management. The mobile app has diagnostic capabilities and is able to provide suggestions on how to control the pest [93]. The developed android application is helpful in integrated pest management. The authors applied machine learning to provide a mobile based pest management. The solution does not solve the FAW problem but provides a similar problem that this research is trying to solve. Being a mobile based solution means that a farmer/user needs to feed the system the information it needs thereby bringing in the manual aspect. The solution being proposed in this research will provide an automated pest monitoring and management which is an improvement from the authors proposed solution.

[94] Applied wireless sensors with special reference to acoustic device wireless sensors to create a pest monitoring and control system for sugarcane. The authors focused on monitoring the main pests that affect sugarcane production such as stalk borer, rood borer, top shoot borer and sugarcane wooly aphid. They adopted acoustic device wireless sensors placed over the field of the farm to monitor the noise level of the pests and if the noise level crosses a set noise threshold, an alarm is triggered notifying the farmer of possible pest infestation in the field. The sensors send noise level data from the field to a control room computer via a ZigBee 802.15.4 digital communication device. The received noise data from the field is displayed at the control room computer in the form of Fast Fourier Transformation FFT [94]. The automation of the pest monitoring system in the authors proposed solution means that the farmer does not need to go to every part of the field to conduct periodical checks of possible pest infestation. The solution that the authors proposed is similar to the solution being proposed in this research the difference is in the technology being used and the pest being monitored.

Other authors have tried to make use of insect images in the monitoring of pests. [95] developed a real time insect monitoring system using field servers. The main insect that was being monitored was the Diamondback moth (*Plutella xylostella*). The insect was monitored using a counting system that employed a commercial home-use insect killer in which a bar coated with pheromones to attract the insect being monitored was installed. The field servers were equipped with low power consuming cameras which were used for the capturing of the image of the insect. Once setup in the field, the male insects were attracted to the system by the pheromones and were killed by a high voltage discharge which in turn triggers the emission of radio signals which is counted as an insect killed by the insect counting module using a CPU chip. The authors integrated the developed image monitoring system with the web service developed for image surveillance [95]. The author's solution makes use of insect images just for counting purposes only but the solution proposed in this research makes use of the FAW images for identification and classification of the FAW insects.

Other authors have also attempted to modify the pheromone trap to automate the monitoring of insects. Authors in [96] presented preliminary work on the development of electronic traps for automated monitoring of the population of insects. They presented the design and implementation of two prototypes, a light dependent resistor sensors prototype and an infrared sensors prototype

with both using a commercially available bucket trap as a base frame. The electronic traps were fitted with an on board microcontroller which enables the traps to be able to detect target insects entering the trap and record each event with a corresponding timestamp. The traps are deployed in the field with each one of them doing automatic counting of the target insect. The insects are attracted to the traps by a lure placed on top of the trap and upon reaching the trap, the insect is intoxicated and falls down the funnel. During its fall, the insect is detected using optical sensors at the bottom of the funnel which triggers the microcontroller to combine the signals from all sensors and applies an algorithm to count the falling insects. The insect data in the trap is transmitted to a decision support system using a network gateway where interested stakeholders have data pertaining to what is happening in the fields. The system provides real-time, high resolution insect data to stakeholders enabling them to make informed decisions when applying appropriate actions [96]. The proposed solution by authors is similar to the solution being proposed in this research only that the authors focused on the use a modified trap and sensors instead of a camera placed in the trap to capture the image to count the insects which is being proposed in this research.

Other authors have used other estimators as classifiers in image processing problems. [99] Attempted to use image processing and soft computing to detect leaf pests using Support vector Machine (SVM) as a classifier of the images over a wide population. 24-bits color resolution pest affected leaf images are obtained using a digital camera and then they are processed to get gray colored images using image segmentation, image classification techniques to detect pests on the leaves. The quality of the image is analyzed using an analysis algorithm in MATLAB software. The RGB image is then segmented using blob like algorithm for segmentation of pests on leaves. The segmented leave part is now analyzed for estimating pest density in the field. The main aim of the study was to detect the type of pests that is affecting the leaves and provide the necessary pesticide information on the detected pest. In addition the authors aimed at providing the right information about the pest to avoid using the wrong pesticide thereby reducing the cost of acquiring pesticides. The authors focus was on detecting which pest affects crops by using pest affected leaf images and recommending the right pesticide that can be applied. The work is similar to the work being presented in this research only that the research is focusing on detecting the actual pest that is affecting the crops which is the fall armyworm.

Different technologies have been proposed in the fight against pests. [100] discussed the role of Internet of Things (IoTs) in the fight against agricultural diseases and pest control. The authors also proposed a three level disease and insect pest control system using IoTs. The system provides a new way to access agricultural information for the farm using the developed monitoring platform in real time. The proposed system consists of sensor nodes for deployment in the target farmland for acquiring environmental conditions, communication system for transmission of data from sensors over the Internet, information service terminal and monitoring software and hardware system for data visualization and analysis. The monitoring platform is an integrated disease and insect pests system that enable would stakeholders such as administrators, experts to access real time and historical disease and Pest data. The authors proposed an IoT based insect monitoring solution which is similar to the solution proposed in this research. The main difference is that we are focusing on monitoring a known pest which is FAW. In addition this research is proposing an early warning system the pushes notifications to farmers and would be stakeholders of possible infestations.

2.3.1. Summary of Gaps in the Literature

Table 1. Related works

Related work	Year	Author	Findings	Gaps
Fall armyworm prediction model on the maize crop in Kenya: an internet of things based approach [91]	2018	S. A. Musungu	<ul style="list-style-type: none"> - Early FAW detection at Pupa stage - Use of obvious parameters temperature and humidity - Push of warnings to farmers and experts 	<ul style="list-style-type: none"> -Include wind speed and wind direction as parameters - Provide a dashboard for visualization and report generation
A climate-adapted push-pull system effectively controls fall armyworm, <i>Spodoptera frugiperda</i> (J E Smith), in maize in East Africa [92]	2018	C.A.O. Midegaa, J.O. Pittchara, J.A. Pickettb, G.W. Hailua, Z.R. Khana	<ul style="list-style-type: none"> - The use of simple and readily available technology developed for control of cereal stem borers in drier agro-ecologies - The tool employs the intercropping of maize with drought resistance plants such as desmodium. 	<ul style="list-style-type: none"> -Provide farmer warnings in case of possible infestation

Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests [93]	2019	Y. He, H. Zeng, Y. Fan, S. Ji, J. Wu	<ul style="list-style-type: none"> - Use of a deep learning approach to the detection of oilseed rape pests -Use of pretrained CNN models increase the prediction accuracy -Provision of a Public mobile app for farmers to use 	- Lack of a provision of notifications to farmers and stakeholders on possible field infestation.
Pest Monitor and Control System using Wireless Sensor Network (With Special Reference to Acoustic Device Wireless Sensor) [94]	2013	N. Srivastava, G. Chopra, P. Jain, B. Khatter	Use of wireless sensors with special reference to acoustic device wireless sensors to create a pest monitoring and control system for sugarcane	- System uses an alarm when there is a possible infestation which means the warning requires the farmer to be present for them to hear the alarm
Real-time Insect Monitoring System by Using Field Server [95]	2008	M. Hirafuji, H. Yoichi, T. Watanabe, M. Asai, H. Haoming, K. Tanaka, T. Fukatsu, T. Kiura, S. Ninomiya	<ul style="list-style-type: none"> -The system employed field servers and low power consuming cameras installed in pheromone traps -System takes immediate action of killing the insect using high voltage discharge. 	-Notifications to users
Electronic Traps for Automated Monitoring of Insect Populations [96]	2010	G.A. Holguin, B.L. Lehman, L.A. Hull, T. Clement, V.P. Jones and J. Park	- Multipurpose and multi-insect monitoring traps	-Provision of dashboard for data visualization
Early Detection of Pests on Leaves Using Support Vector Machine [99]	2014	M. Manoj, J. Rajalakshmi	-Provision of pesticide information reducing on costs associated with pesticide application	-Provision of a pest infestation warning system
Internet of Things Application to Monitoring Plant Disease and Insect Pests [100]	2015	Y. Shi, Z. Wang, X. Wang, S. Zhang	Provision of an easily accessible real time agricultural disease and pest information database	- Provision of a warning system

CHAPTER THREE

METHODOLOGY

3.1. Introduction

The chapter discusses the methods and procedures used in the research. The methods used to collect fall armyworm images used in the training of the convolutional neural network model and the methods used to train the model are presented here. In addition, the chapter presents the methods used in the design, development and implementation of the web and mobile applications.

3.2. Study setup

A Qualitative method was used in this research. To properly understand the problem at hand, the study was done in phases. The first phase was to understand the behavior of the moth which needed interaction with entomologists and biologists. This phase involved attending a one week expert workshop which was organized by the Food Agriculture Organization (FAO) in Lusaka. The workshop brought together entomologists and biologists from different African countries to provide and discuss requirements that would be used to develop a fall armyworm monitoring app. The workshop provided an opportunity for informal interviews to be conducted to better understand the manual process of fall armyworm monitoring.

The second phase was to understand how the manual process is actually done using the pheromone trap. The second phase involved an interaction with experts from the School of Agriculture Sciences at the University of Zambia. Interaction with experts provided an opportunity for informal interviews to be conducted on how best the pheromone trap can be modified for automated fall armyworm monitoring.

The last phase was data collection. This was the process of collecting field male moth images to be used in the training of a Convolutional neural network. The image collection was done from Zambia Agriculture Research Institute (ZARI) fields. The images were collected with the aid of entomologists who played the role of identifying the fall armyworm moth.

3.3. Current manual Process

Currently the fall armyworm is monitored using manual methods as observed during an interaction with FAO, ZARI and UNZA School of Agriculture Sciences experts. Farmer has to visit the field/farm to check for possible FAW occurrence by looking out for presence of larvae on the

plants or by checking the presence of FAW moths in the field. A presence of larvae or pupae or moths indicates the presence of the fall armyworm. The other approach is the use of pheromone traps placed in the field. Pheromones are traps that contain a lure which mimic natural FAW pheromones that attract male moths for mating. The traps are able to trap the moth once its inside by using some sticky substance placed at the bottom of the trap. The traps are usually monitored after a week and the caught moths are counted to indicate the presence of the fall armyworm.

The proposed solution will make use of the fact that the traps are able to attract the male moth by capturing the image of the moth as it enters the trap then the image is pushed to the AI model for identification which is counted if it is correctly identified hence the need to capture the images of the FAW moth.

3.4. Conceptual Framework

Figure .18 shows a model of the system being proposed in this study. The model will use fall armyworm images captured from the traps in the field. The model has a Convolutional Neural Network model that predicts and classifies the received images if it is FAW or not. If it is FAW, the received data is recorded in the database and the web based system makes the data available to farmers and other stakeholders for analysis. In addition, the system sends notifications (SMS/email) to farmers and other stakeholders as warnings of possible field infestation.

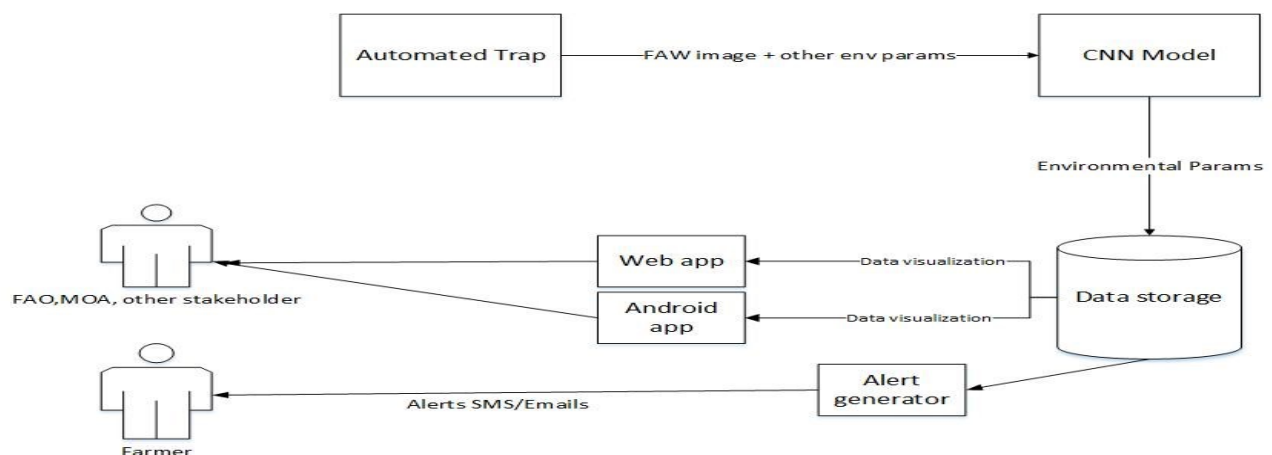


Figure 14: Conceptual Framework

3.5. Ethical Considerations

The research was given as a project from FAO and did not involve formal interviews with experts hence did not necessary need an ethical clearance but the research applied for ethical clearance from UNZA and was granted to conduct the study.

The field data collection did not involve any species that is considered endangered hence no permits were required for us to collect data from the field. In addition confidentiality was maintained throughout the study as no sensitive information belonging to FAW experts at ZARI was used in a wrong way or in a way that compromises their organization.

3.6. Convolutional Neural Network model Methodology

3.6.1. Data Collection

Data used for the training of the CNN model was collected from 4 sources namely the field, Lab setup, and Internet crawling and data augmentation. Field data was collected from the Zambia Agriculture Research Institute (ZARI) in Chilanga and taking photos of identified male fall armyworm moth using a Cannon PowerShot SX430 IS Camera. With the aid of entomologists who are experts in fall armyworm moth identification, data was obtained from pheromone traps and from maize plants as fall armyworm Favours maize fields and most of them were resting in the fields as they are active during the night due to their delicate nature when exposed to the sun. In addition to taking photos and to supplement the collected data, live fall armyworm moths were collected in beakers and set up in a lab kind of environment. Since the study was dealing with the identification of live moths, the research tried as much as possible to take photos of live moths strategically positioned in the pheromone trap or in the beaker though there were also some images of dead fall armyworm moths which made it easier to strategically position them. The collected data was not enough so it was mixed with data from the field and lab setup with images from the Internet of fall armyworm moths. Images were manually collected from the internet by searching using the browser to make sure that the fall armyworm moth images being collected resemble the field images in terms of features as shown by Figure. 21 and making sure that the collected image contains only the FAW moth as shown by Figure 19 (c).



Figure 15: Images collected from the (a) field, (b) lab setup and (c) internet respectively [101]

The data collected from the field, Lab setup and Internet was not enough to get a good confidence that a reasonable model will be trained. To get a dataset that was more reliable in terms of model training and testing, data augmentation was used, in fact 60% of the dataset came from augmentation. A python script was used to generate at least 8 67x65 sized images as shown by Figure. 20, from an image by applying random cropping, random flipping, and rotation using a rotation range of 40 degrees and random zoom in and Zoom out.

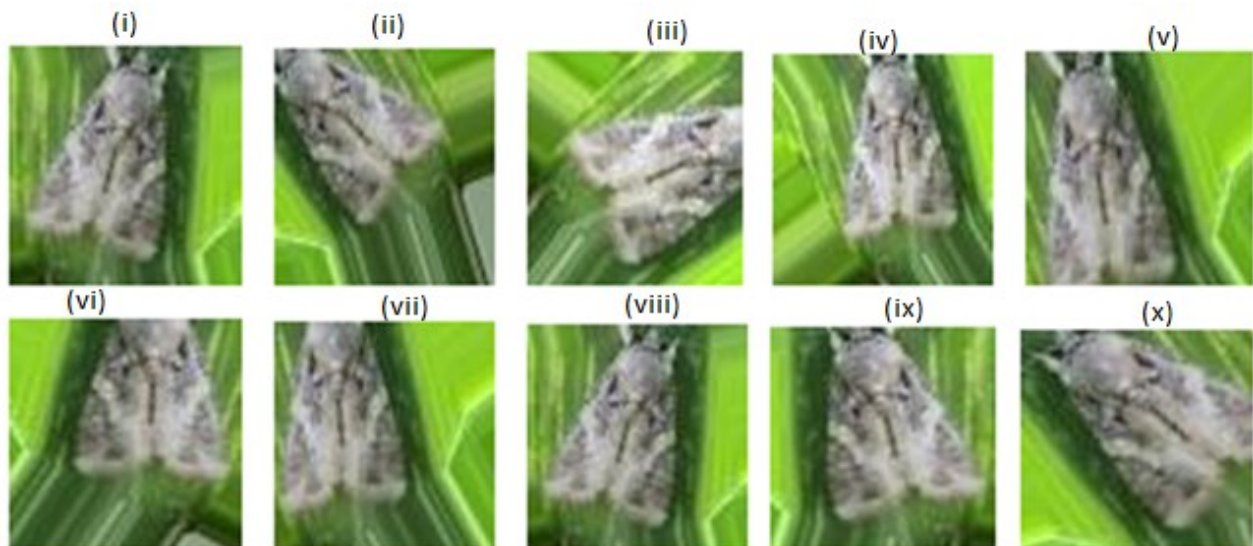


Figure 16: Sample of image Augmentation with (i) being the original image

The focus in this study was mainly on the automated identification of fall armyworm moth but Convolutional Neural Networks requires that there are at least two classes to perform a classification problem properly, therefore other insects and moths were used which were referred to as false positives. There was no big reason that compelled the study to use certain insects as a false positive class because it is hoped that the lure used in the traps which will act as the source of images will attract only fall armyworm moth or at least 90% of the attracted insects will be the

fall armyworm and the majority attracted will be the male moths. Therefore a variety of insects with majority of them being moths which usually or are likely to attack crops in Zambia, are likely to be attracted to the pheromone traps and have similar features with the fall armyworm which included but not limited to African bollworm moth, African fly, mosquitoes, corn borer moth, cutworm moths, stalkborer moths, stem borer moths were included. The images of other insects were all collected by crawling the internet choosing the best images in terms of resolution and applied data augmentation to increase the dataset. To see how the model will perform if it was trained on unclear images (pixilated at 10pixels) and tested on clear actual fall armyworm images, another dataset of blurred fall armyworm images from the dataset we made using a data augmentation python script was created.

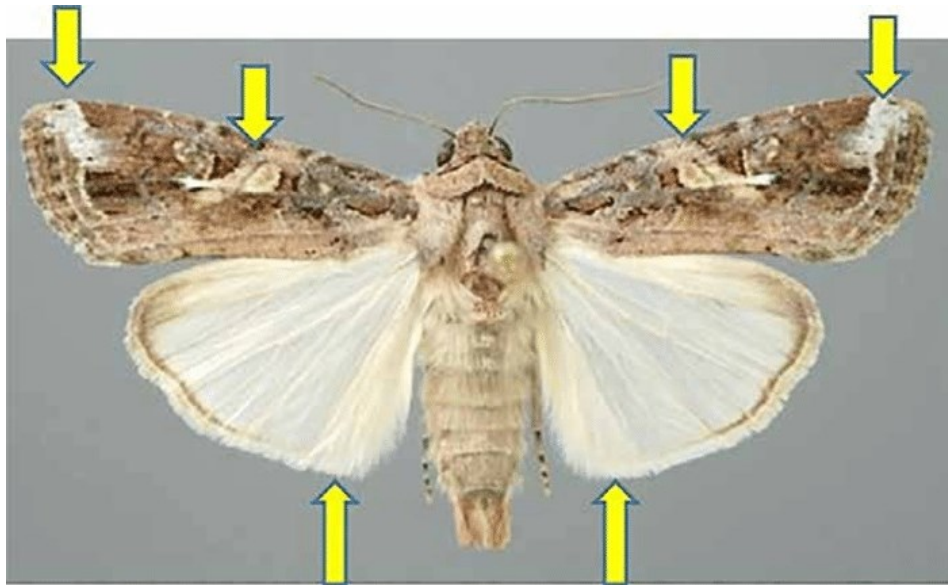


Figure 17: FAW Moth Unique Features

Source: https://www.researchgate.net/publication/323612839_Host_plant_resistance_to_fall_army_worm

3.6.2. Data Selection and Distribution

Once the dataset was structured, training, validation and testing datasets were determined. Before doing data augmentation, 30 FAW images and 15 false positive images were selected as test datasets. Mostly original clear unaugmented images as test images were selected. The selected test images were not used in the training to avoid overfitting and were used only after the model was trained. The test datasets was divided into three parts of 15 images each namely Augmented, Actual Clear FAW and False Positives. The test Augmented images were a mixture of pixilated

(at 5 pixels) images and randomly rotated, random vertically flipped and rescaled images. A total of 750 FAW images and 400 false positive images were generated using the process described in data collection section when data augmentation was done. Out of 780 FAW images, 10% were used as validation images since Tensorflow uses auto validation and the rest of the images were used as training dataset. From 400 false positives, 10% were used as validation dataset as well and the rest as training dataset. Table. 2. Shows the division of the dataset.

Table 2. Data distribution

Category	Total	Training	Validation	Test
FAW	780	645	75	30
False positives	400	360	40	15

3.6.3. Model Training and Testing

This entailed the use of test data to make sure that the model was trained properly by the analysis of the actual model output versus the expected output. Through validation, any errors identified in the performance evaluation was used to adjust the model and make the necessary corrections for a much accurate model. Transfer learning was used to train a Softmax layer on Top of the InceptionV3 model trained on the imageNet dataset which is a dataset of about 15 million labeled high resolution images with around 22, 000 categories/classes. The inception model gives high accuracy but is slower during training and looking at the dataset we had, it was the ideal choice. The training of the model was done on a Corei5 machine with 4GB RAM and 1TB hard disk. The operating system used was Ubuntu 18.04.

A python virtual environment were all the dependents were installed to avoid conflicting dependent versions was created. Before training, Convolutional Neural Network algorithms demands defining parameters before training as they are likely to influence the classification results therefore all the parameters such as dropout, batch size, image cropping and so on were left as they are defined in the pre-trained model and only altered the learning rate and training steps. A simple open source model training script provided by Google was used which was customized to fit the needs of this study. The customization done on the script among others included the provision of our dataset folder which consisted of two classes namely faw and other, adjustments

of the learning rate and training steps and renaming of the protobuf files to the names that are needed for this study. 8 trials were ran and in each trial changing the learning rate and the number of training steps (epochs) in order to get the best and tested the models. From the 8 trials the best two trials which gave the best test accuracy and error rate were selected. Out of the two selected trials, the parameters of the model that gave the best results after being tested on test data were picked and trained a pixilated FAW images model. The purpose of the pixilated trial was to test how the model will perform if trained on pixilated (blurred) images and tested on clear, augmented and false positive images. After training the Pixilated FAW images model, the test data was run using the model and recorded the results and at the end there was three trials namely i). Clear FAW images, 0.01 learning rate, and 1500 training steps ii). Clear FAW images, 0.001 learning rate and 5500 training steps iii). Pixilated (blurred) FAW images at 10 pixels, 0.01 learning rate and 1500 training steps and ranked them according to the best performing model in terms of identifying the actual and augmented FAW images and false positives.

3.6.4. Model Deployment

After training a model that has an accuracy that gives confidence in terms of FAW prediction accuracy, the model needed to be hosted as a service to be used in the automated prediction and classification system. This was archived using what is called Tensorflow serving which is a flexible, high performance way of serving machine learning models as a service in production. Tensorflow serving makes it easy to deploy machine learning algorithms and experiments while keeping the same architecture and APIs. It provides out-of-the-box integration with Tensorflow models and can easily be extended to serve other types of models and data. Using Tensorflow serving direct introduces some tight coupling between the serving server and the frontend system hence to eliminate the tight coupling, a flask server was used on top of Tensorflow serving. Using flask server on top of Tensorflow server makes sure that any preprocessing technicalities are abstracted from the frontend users, brings in the use of another language aside from python as a backend server language, gives capabilities of serving multiple models without creating many domain URLs as it allows addition of new routes/functions at the backend while maintaining the same URL and a flask server would enable better exception handling.

In a nutshell, Tensorflow serving using flask server works as follows. A client will make an http request to a defined URL on a flask server with the image as a parameter. Then upon receiving the

request, the flask server builds a request that Tensorflow understands and calls the latest version of the model. Upon receiving the request from the flask server, the Tensorflow server does its preprocessing and returns a prediction and classification JSON response to the flask server which responds to the calling client with a predication JSON response as well. The Tensorflow server-flask server architecture is shown by Figure. 22 below.

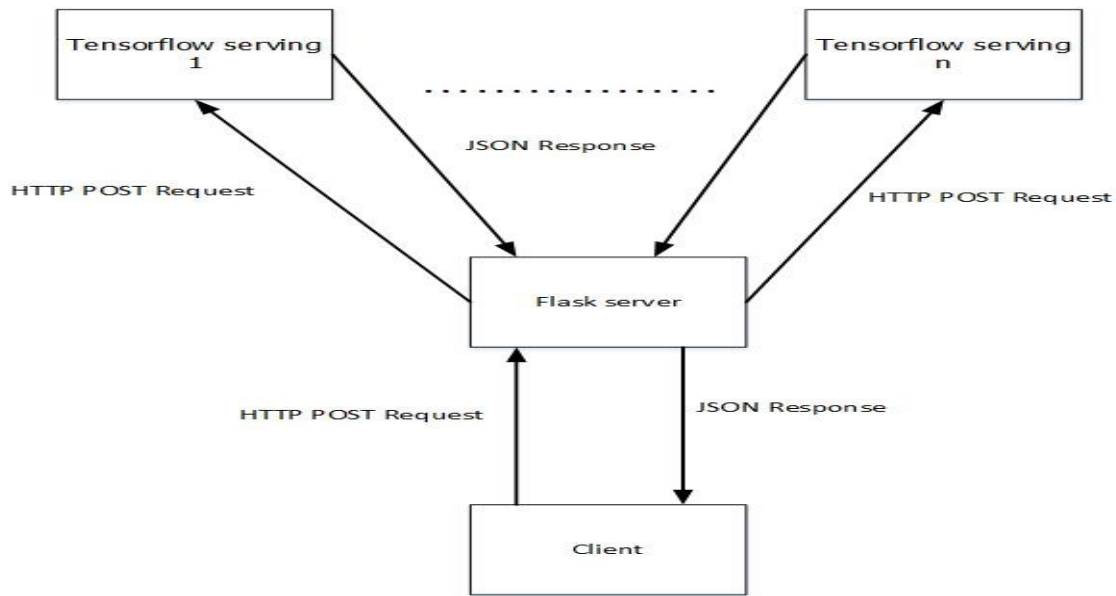


Figure 18: Illustration of Tensorflow serving using flask

Due to lack of resources to host the applications separately, the flask server is hosted on the same server as the other applications hence it was assigned to run on port 8000. The flask server runs in the same python virtual environment as the Tensorflow model. The flask server exposes a URL to be invoked by the client application which in this case is the IMPS API. The URL has the format as http://IP:PORT?file=path_to_image where IP is the internet protocol of the server where the flask server is running, PORT is the port number assigned to the flask server and file is the name of the image to be classified which is usually the path to the image. Once invoked with the image, the flask server will respond to the client with an array response as shown by Figure. 23 below.

```
{
  {
    'faw',
    'other'
  } {
    '0.892535',
    '0.107465'
  }
}
```

Figure 19: Tensorflow serving flask Array response

3.7. System Development methodology

This section describes the techniques used in the development of the web based and mobile applications. It also explores the design and architecture of the proposed prediction system which illustrates the interconnections between the system components and interactions. Detailed requirements and various conceptual models pertaining to the system are given which will be achieved through UML diagrams.

3.7.1. Requirements Analysis

This subsection describes some of the system requirements that were identified for the system. The section reviews user expectations for the FAW monitoring applications. From the objectives and user requirements, this part outlines most of the requirements that will be achieved in the research.

3.7.1.1. Functional Requirements

These are requirements that dictates what the system is supposed to do with regard to functions of a system or its components where a function is described as a specification of behavior between outputs and inputs. The following are the components that makeup the FAW monitoring system.

3.7.1.1.1. Web Interface

This system is composed of the dashboard, the API and SMS/Email alert module. Its functional requirements will include but not limited to:

- The system should be able to allow farmer registration and management which should include farm registration as well.

- The system should be able to allow trap management. This should include trap registration, lure being used at the trap, location of the trap and so on.
- The system should be able to allow crop management
- The system should be able to allow farmer and FAW expert user management
- System should allow management of locations i.e. provinces, districts, wards
- System should allow management of pest thresholds per ward
- System should allow view of statistical reports(dashboard or graphical or tabulated)
- System should allow the view of spatial data (Map of Zambia showing location of traps in each farm) per province.
- System should allow the download of reports.
- System should be able to classify the FAW images and log the data together with environmental parameters such as temperature, humidity, and wind speed and wind direction.
- System should be able to send alerts both sms and email to farmers and FAW experts if the pest threshold in ward is reached
- System should be able to allow farmer and FAW expert Login
- System API should allow farmer and FAW expert mobile app login authentication
- System API should provide functions to be invoked by the mobile app and other external system for data visualization

3.7.1.1.2. Mobile app

The mobile app will mainly be used for data visualization by both farmers and FAW experts. Its functional requirements will include but not limited to:

- The app should allow farmer and FAW expert login
- The app should allow visualization of data based on user filters
- App should allow farmers to view pest data based on the ward where their farm is located
- App should allow users to view data, at province, district or ward level
- App should allow farmer to update their personal details.
- App should allow users to view counts of total farmers, farms, traps and crops being monitored

3.7.1.2. Non-Functional Requirements

These are requirements that defines the properties that the system is expected to have that can be used to judge its operations instead of behavior. For the proposed FAW monitoring system, the following non-functional requirements are needed for all its components:

- System must be easy to use meaning that users must be able to learn and operate the expected functions with easy. System must be intuitive enough for farmers to learn its usage.
- System must be able to accommodate a large number of users and user groups
- System must be scalable meaning that the system must be able to accommodate new requirements incorporation without affecting the existing implemented requirements.
- System must be readily available meaning that users such as farmers and FAW experts must be able to depend on the system to be available at all times regardless of the factors that may affect it.
- System must be able to handle hardware related issues and continue providing the required services.
- System must meet the defined requirements
- System should provide role based access meaning that each user must belong to a certain role in the system with specific rights that the role can perform.
- System should be easy to maintain meaning that system errors and faults must be easy to fix.
- System should be able to respond quickly meaning that it should be able to respond within the acceptable time to the user.

3.7.2. System Architecture

The system architecture is meant to give an understanding of the interconnections and interaction of the system components. The architecture addresses how the system components will be able to support the achievement of the system functionality. Figure 24 shows the proposed system architecture for this study. It shows how the different sub-systems are related and interconnected and how the data will flow from the time it is collected from the trap to the time it is viewed by a user.

First the data will be collected at the trap site by the modified and automated pheromone traps. Depending on the cellular network connectivity at that particular time, the collected data will be sent to the cloud server for processing. On the cloud server, the data is received by a REST API which invokes the flask server for FAW identification and classification. When the flask response is received, the API, compares the prediction accuracy with that of the acceptable FAW accuracy of which if it is less than the acceptable value, the API discards the data and responds back with a success. If the prediction accuracy received is equal or greater than the acceptable FAW prediction accuracy, the API Loads the data in the database. Once the data is loaded onto the database, the relevant stakeholders can access it via the web interface or mobile app. The API also receives http requests from the android app for processing which include mobile app user login, mobile app data visualization requests and any other mobile application processing requests.

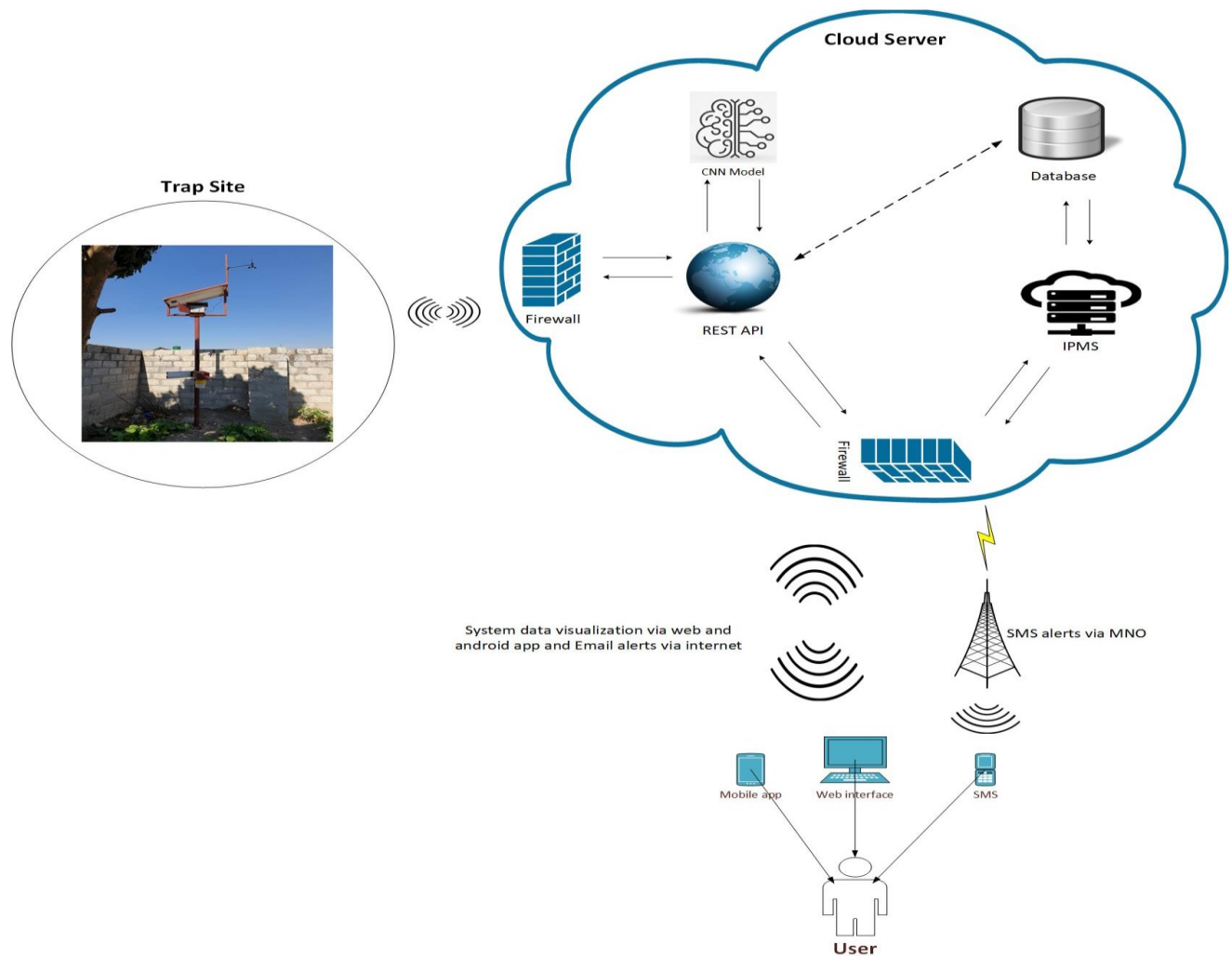


Figure 20: System Architecture

Based on the configured pest count threshold per ward, the alerts modules gets counts per ward and compares with the configured thresholds. If the threshold for that particular ward is equal or more than the set threshold, the alerts module push SMS and email alerts to farmers and FAW experts based on their preference whether they should receive SMS or email alerts only or both and if they can receive alerts.

3.7.3. System Behavioral Modelling

Functional System requirements define how the system will behave or functions that the system is supposed to perform. Use case diagrams were used to show the functional requirements of the system as they would be viewed by the user.

3.7.3.1. Use Case Diagrams

Use case diagrams represents system functions or goals which responds to inputs and produce outputs that are useful to the user. They specify the expected behavior of the system and not how that behavior will happen. They show actors, their use cases and their associated relationship. They can be denoted by both textual and visual representation. Use case diagrams are an effective way of communicating system behavior in the user's terms by specifying all the external visible system behavior [97]. Three primary actors and roles were identified in the system namely farmers, FAW experts and system administrators but more actors and roles can be added as sub-set roles of what the admin can do.

3.7.3.1.1. Web Interface

Figure 25 illustrates the use case diagram of the proposed web app. The identified actors include but not limited to farmers, FAW experts, administrators and the system with their corresponding use cases. It gives some of the identified functions that the expected system users will perform. Some of the functions that system will perform include responding to requests, pushing notifications, processing prediction requests and receiving trap data. A farmer is expected to perform functions such as system login, viewing surveys, viewing the dashboard, viewing spatial data and so on. While the FAW expert is expected to perform functions like farmer, farm, trap and crop registration, view dashboard, manage notifications, login, and manage data collections methods and setup of traps among other functions. The system administrator is expected to perform all the functions that other system users can perform and other functions such as user and user role management.

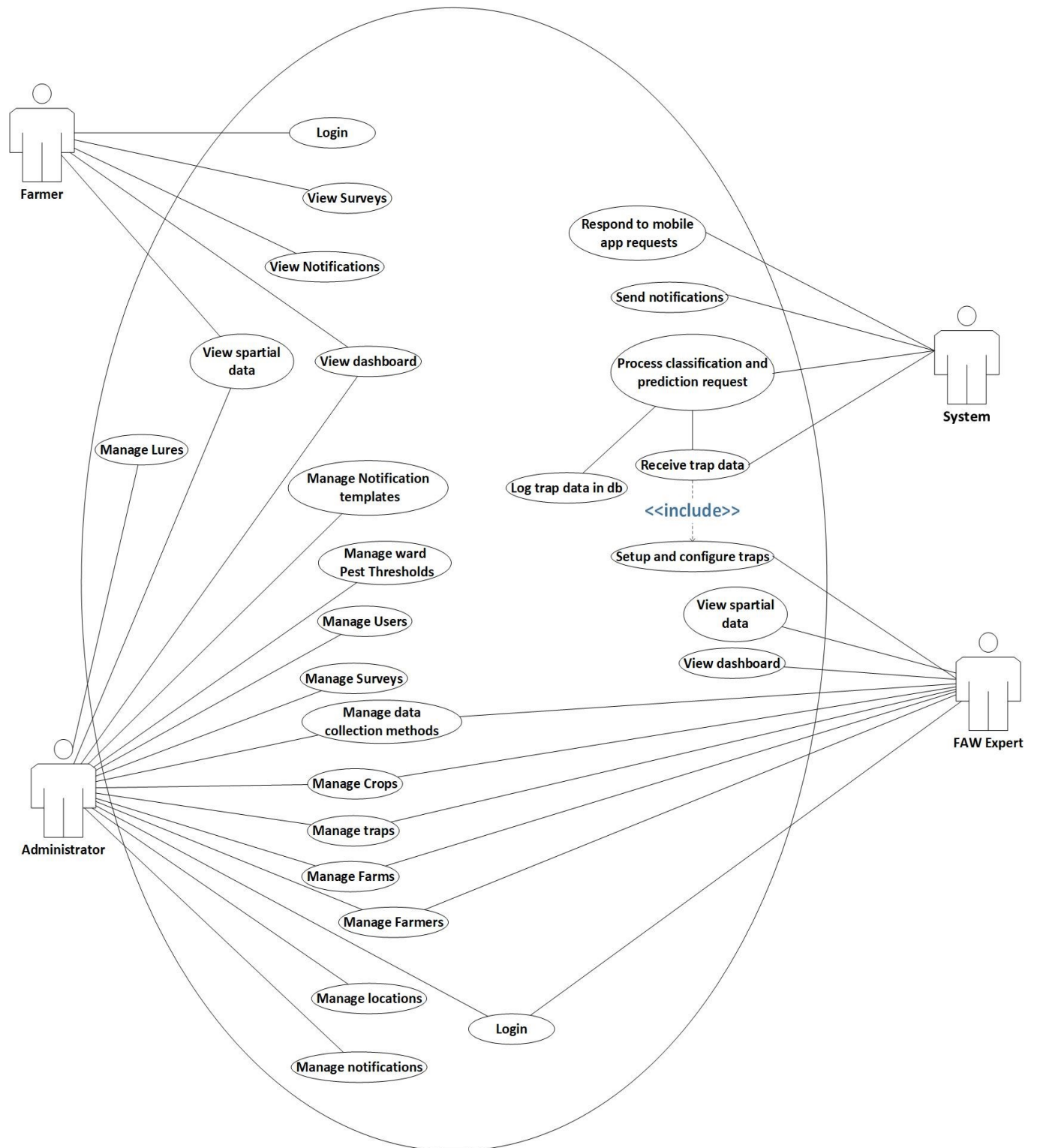


Figure 21: Web app use case diagram

3.7.3.1.2. Mobile app

Figure 26 illustrates the use case diagram of the proposed mobile app with some of the identified use cases. The identified actors are farmers and FAW experts.

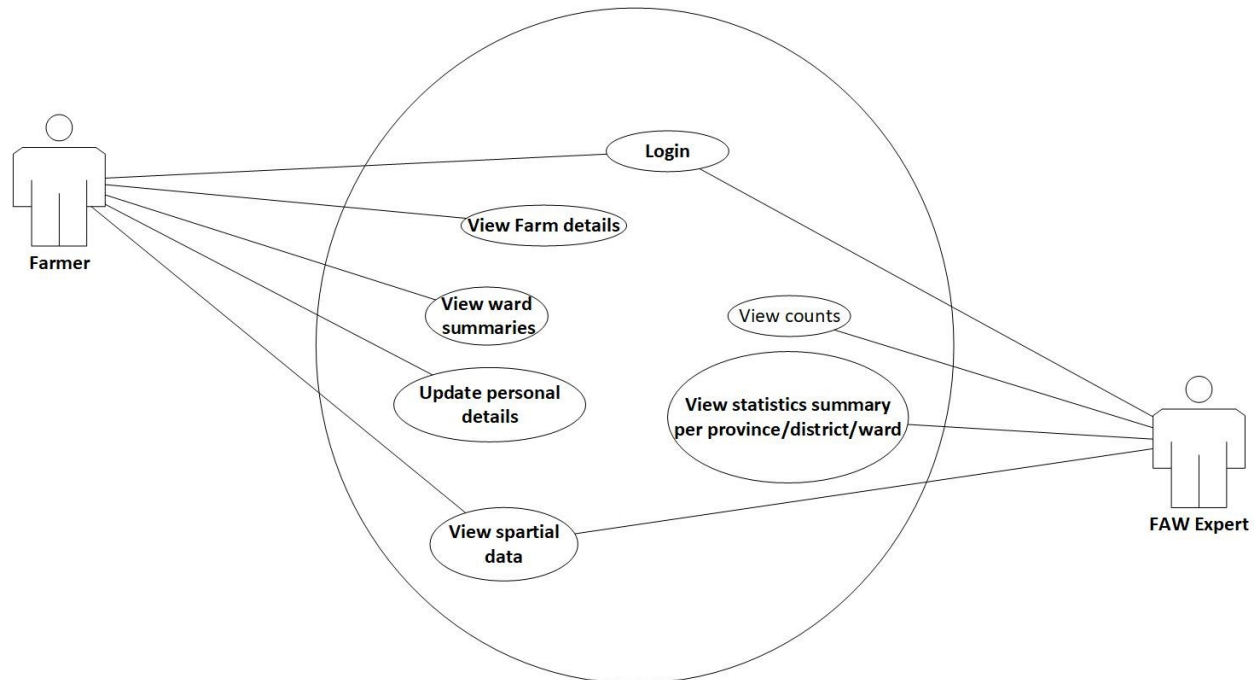


Figure 22: Mobile app use case diagram

3.7.3.2. Use Case Narrations

These are textual representation of events that are encountered when an actor is interacting with the system [97]. In this section, some of the important use case narrations for the system are presented.

3.7.3.2.1. Web Interface

Use case narrations for the web interface include but not limited to Login, view spatial data, Manage crops/Lures, predict and classify, Manage locations, Manage users.

Table 3: Login Use case narration

#	Use case	Primary Actor	Precondition	Post condition
1	Login	Farmer/FAW expert/Administrator	User must be registered in the system	User successfully login into the system
	Main Success Scenarios			
	Course of events		System action	
	i. User enters username and password ii. User clicks login button		<ul style="list-style-type: none">System authenticates userSystem Logs user into the system	
	Alternative course of events			
	User entered wrong credentials, system prompts user to enter again			
#	Use case	Primary Actor	Precondition	Post condition
2	View dashboard	Farmer/FAW expert/Administrator	<ul style="list-style-type: none">User must be registered in the systemUser must login into the system	User can view monitoring dashboard
	Main Success Scenarios			
	Course of events		System action	
	i. User navigates to dashboard page		<ul style="list-style-type: none">System takes user to dashboard page	
	Alternative course of events			

Table 4: View spatial data use case narration

#	Use case	Primary Actor	Precondition	Post condition
3	View spatial data	Farmer/FAW expert/Administrator	<ul style="list-style-type: none">User must be registered in the systemUser must login into the system	User can view spatial data
	Main Success Scenarios			
	Course of events		System action	
	i. User navigates to spatial data page		<ul style="list-style-type: none">System takes user to spatial data page	
	Alternative course of events			

Table 5: Manage Crops/Lures use case narration

#	Use case	Primary Actor	Precondition	Post condition
5	Manage Crops/Lures	FAW expert/ Administrator	<ul style="list-style-type: none">User must be logged in	<ul style="list-style-type: none">Crop/Lure successfully added/updated/deleted
	Main Success Scenarios			
	Course of events		System action	
	i. User navigates to Configs page ii. Crop/Lure addition/updating/deletion initiated		<ul style="list-style-type: none">System takes user to Configs pageSystem saves crop/lure details into the system	
Alternative course of events				
Crop/Lure management fails, user repeats the process				

Table 6: Manage Locations use case narration

#	Use case	Primary Actor	Precondition	Post condition
6	Manage Locations	Administrator	<ul style="list-style-type: none">User must be logged in	<ul style="list-style-type: none">Province/District/Ward successfully added/updated/deleted
	Main Success Scenarios			
	Course of events		System action	
	i. User navigates to locations page ii. Province/District/Ward addition/updating/deletion initiated		<ul style="list-style-type: none">System takes user to locations pageSystem saves Province/District/ward details into the system	
	Alternative course of events			
Province/District/Ward management fails, user repeats the process				

Table 7: Manage ward threshold use case narration

#	Use case	Primary Actor	Precondition	Post condition
7	Manage ward threshold	Administrator	<ul style="list-style-type: none">User must be logged in	<ul style="list-style-type: none">Ward threshold successfully added/updated/deleted
	Main Success Scenarios			
	Course of events		System action	
	i. User navigates to Configs page		<ul style="list-style-type: none">System takes user to Configs page	
	ii. Ward threshold addition/updating/deletion initiated		<ul style="list-style-type: none">System saves ward threshold details into the system	
	Alternative course of events			
Ward threshold management fails, user repeats the process				

Table 8: Manage users use case narration

#	Use case	Primary Actor	Precondition	Post condition
8	Manage users	Administrator	<ul style="list-style-type: none">Admin user must be logged in	<ul style="list-style-type: none">User successfully added/updated/deleted
	Main Success Scenarios			
	Course of events		System action	
	<ul style="list-style-type: none">i. Admin user navigates to user management pageii. User addition/updating/deletion initiated		<ul style="list-style-type: none">System takes user to user management pageSystem saves user details into the system	
Alternative course of events				
User management fails, user repeats the process				

Table 9: Send Notifications use case narration

#	Use case	Primary Actor	Precondition	Post condition
9	Send Notifications	System	<ul style="list-style-type: none">System should have ward statistics equal or above the set thresholdSystem should have farmers/Users that can receive alerts	<ul style="list-style-type: none">Alert successfully pushed
Main Success Scenarios				
Course of events			System action	
			<ul style="list-style-type: none">System fetches counts for all wards that have counts equal or greater than set thresholdSystem pushes the alert	
Alternative course of events				
Alert push fails, system repeats the process				

Table 10: Predict and classify image use case narration

#	Use case	Primary Actor	Precondition	Post condition
10	Predict and classify image	System	<ul style="list-style-type: none">Traps setup and configured	<ul style="list-style-type: none">Image successfully predicted and classifiedTrap data logged into the db
	Main Success Scenarios			
	Course of events		System action	
			<ul style="list-style-type: none">System receives trap data and calls model server with imageModel server predicts and classify image and sends back responseSystem logs trap data into the databaseSystem responds back to trap	
	Alternative course of events			
	<ul style="list-style-type: none">System does not receive all required parameters, system responds with missing params responseSystem receives other insect prediction and classification from the model, system discards trap data			

3.7.3.2.2. Mobile app

Use case narrations for the mobile app include but not limited to farmer details update.

Table 11: Farmer Update personal details use case narration

#	Use case	Primary Actor	Precondition	Post condition
8	Update personal details	Farmer	<ul style="list-style-type: none">Farmer must be registeredFarmer must have logged in into the app	<ul style="list-style-type: none">Farmer details successfully updated
	Main Success Scenarios			
	Course of events		System action	
	i. Farmer logs into the system ii. Farmer navigates to profile iii. Farmer updates personal details		<ul style="list-style-type: none">System logs farmer into the appSystem takes farmer to profile pageSystem sends the farmer details to the APIAPI saves farmer detailsSystem notifies farmer on detail update	
	Alternative course of events			
	Personal details update fails, farmer repeats the process			

3.7.3.3. Sequence Diagrams

Sequence diagrams are used to model the interactions between the actors and the objects in the system and also show interactions between objects themselves [98]. Sequence diagrams are meant to show the dynamic side of the system. The following are sequence diagrams for important functions in the system components.

3.7.3.3.1. Web Interface

Login

User tries to login into the web interface by supplying their username and password. The system authenticates the supplied credentials. If credentials are wrong, system prompts user to enter correct credentials. If credentials are correct, system logs user into the system. Users in this case refers to farmer, FAW expert or administrator.

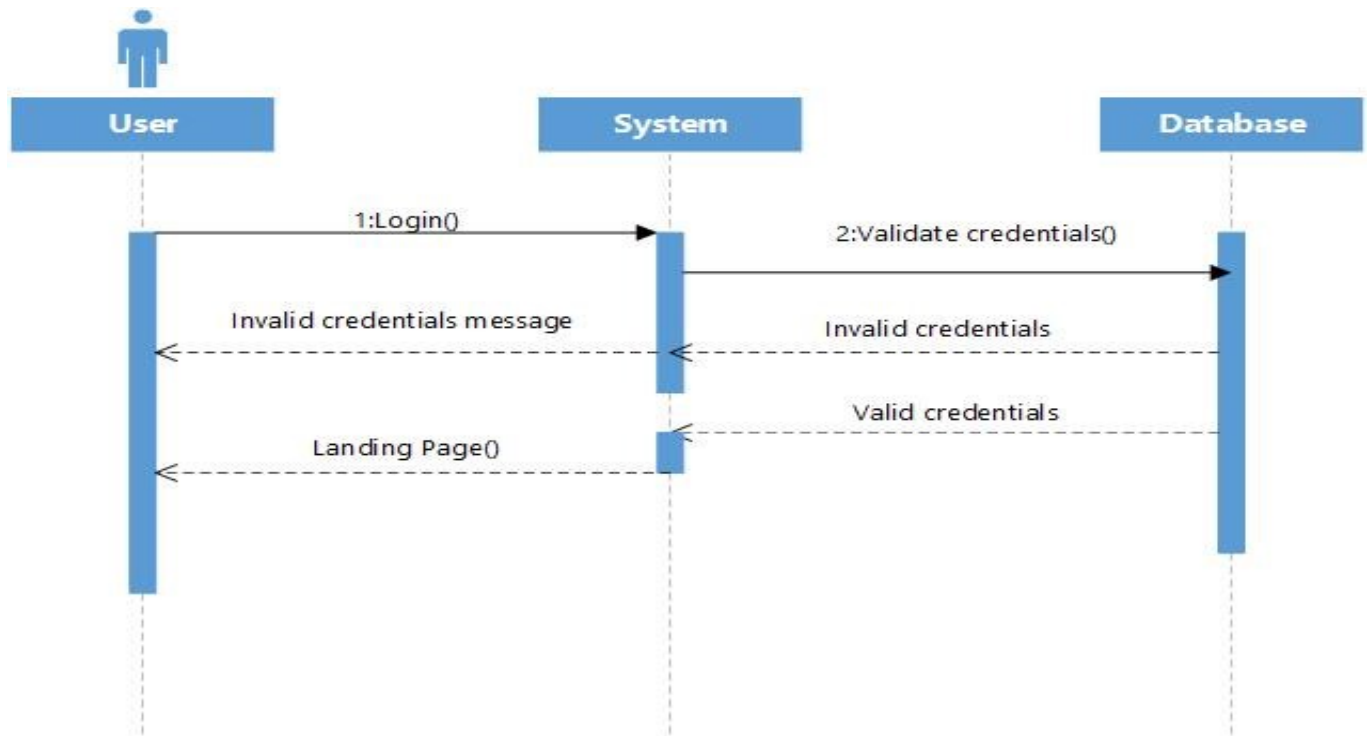


Figure 23: Web login Sequence diagram

Manage Farmer/Farm/Trap

User navigates to Farmer management page and performs the described actions by going through the relevant pages for each action. FAW experts/Admin can manage farmers, farms and Traps as shown by Figure 28.

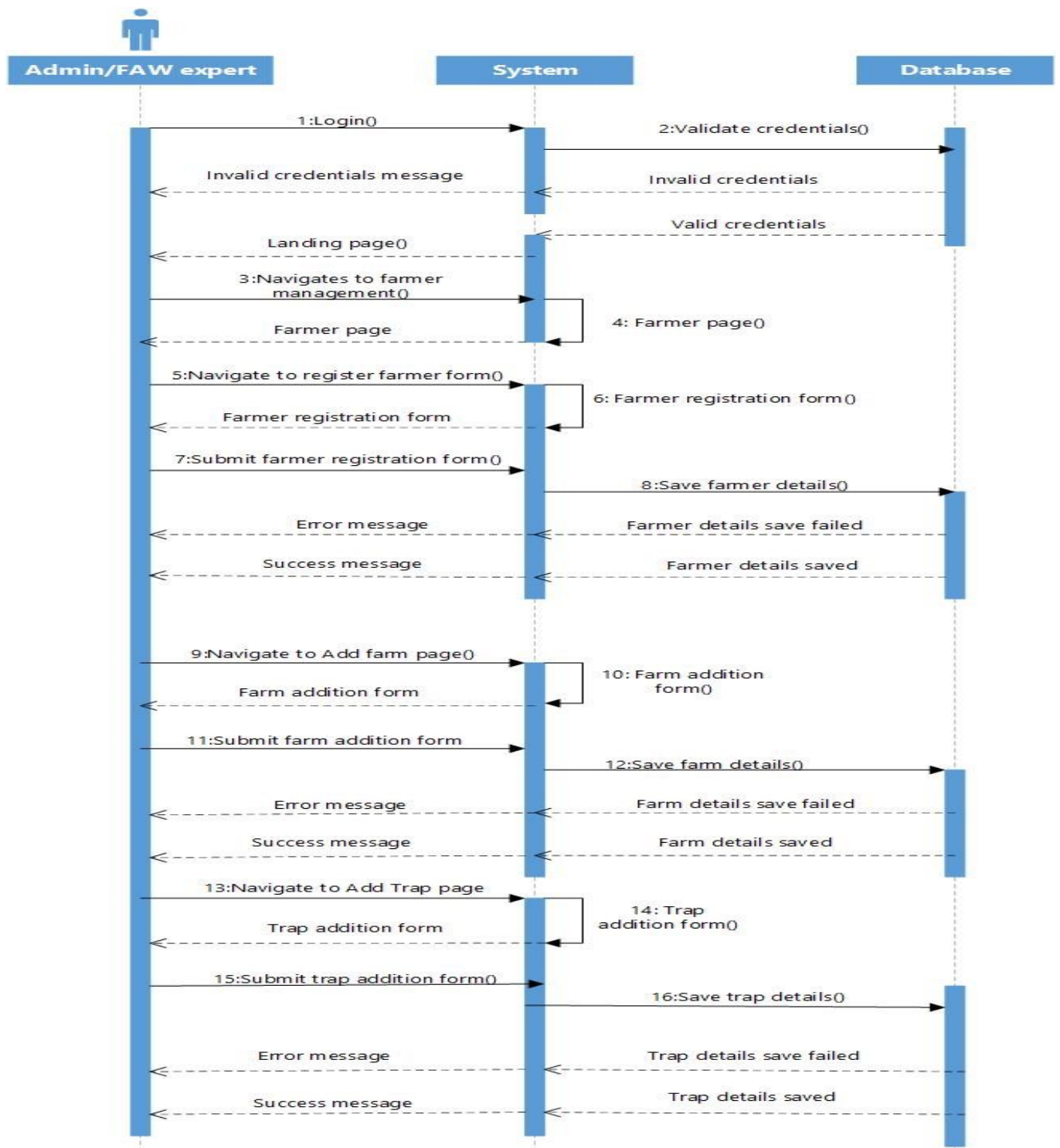


Figure 24: Manage Farmer/Farm/Trap sequence diagram

Manage Crops/Lures

One of the use cases for the FAW expert/Administrator is the management of crops being monitored and lures being used at the traps. Figure 29 shows the actions and events that the user will go through to manage a crop/lure.

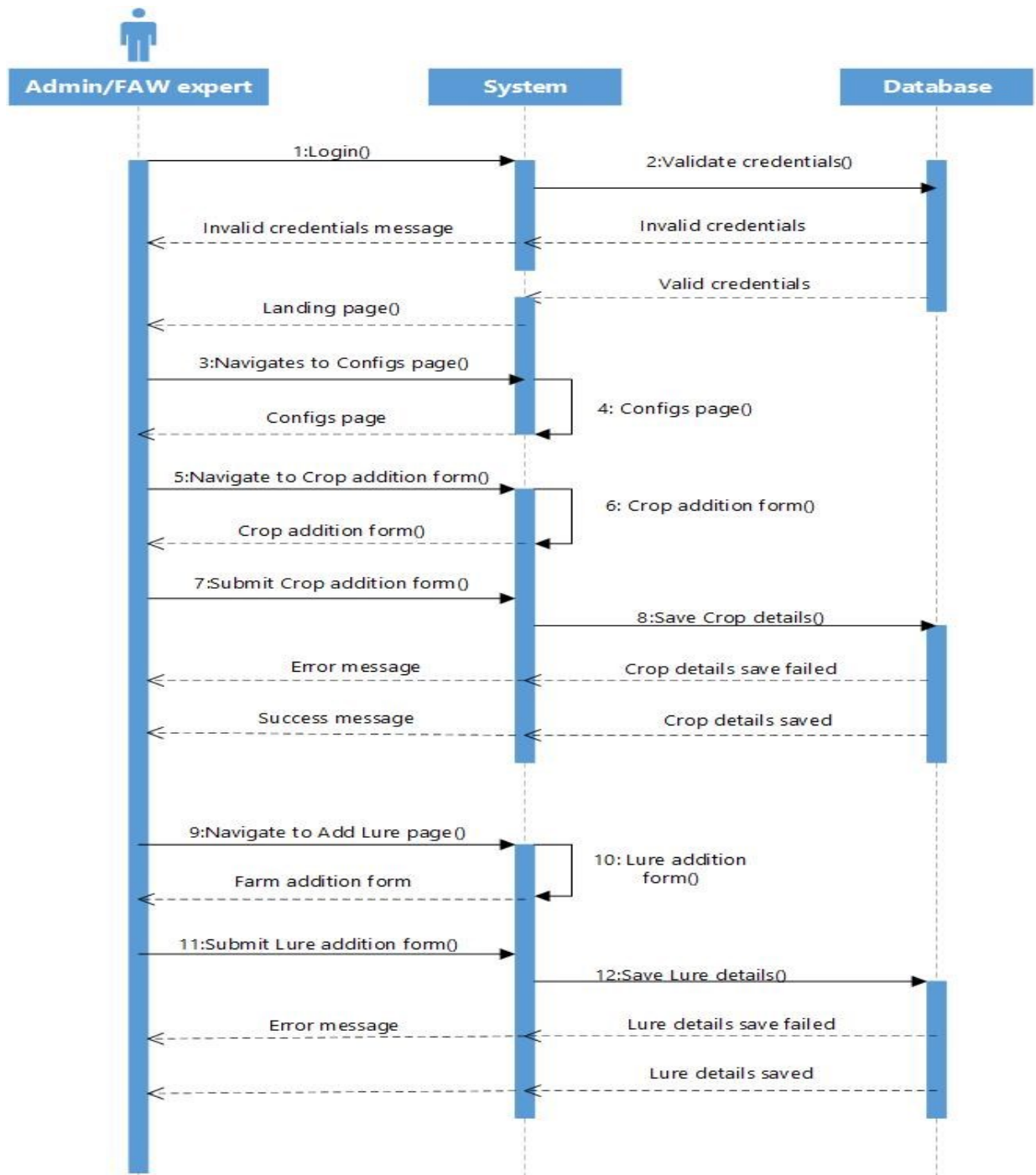


Figure 25: Crop/Lure Management sequence diagram

Manage Locations (Provinces/Districts/Wards)

Management of provinces, districts and wards is one of the functions that the administrator will be performing. Management of locations includes all the CRUD actions. Figure 30 shows the sequence diagram for management of locations. Locations are needed as a farmer and farms are located in provinces, districts and in wards.

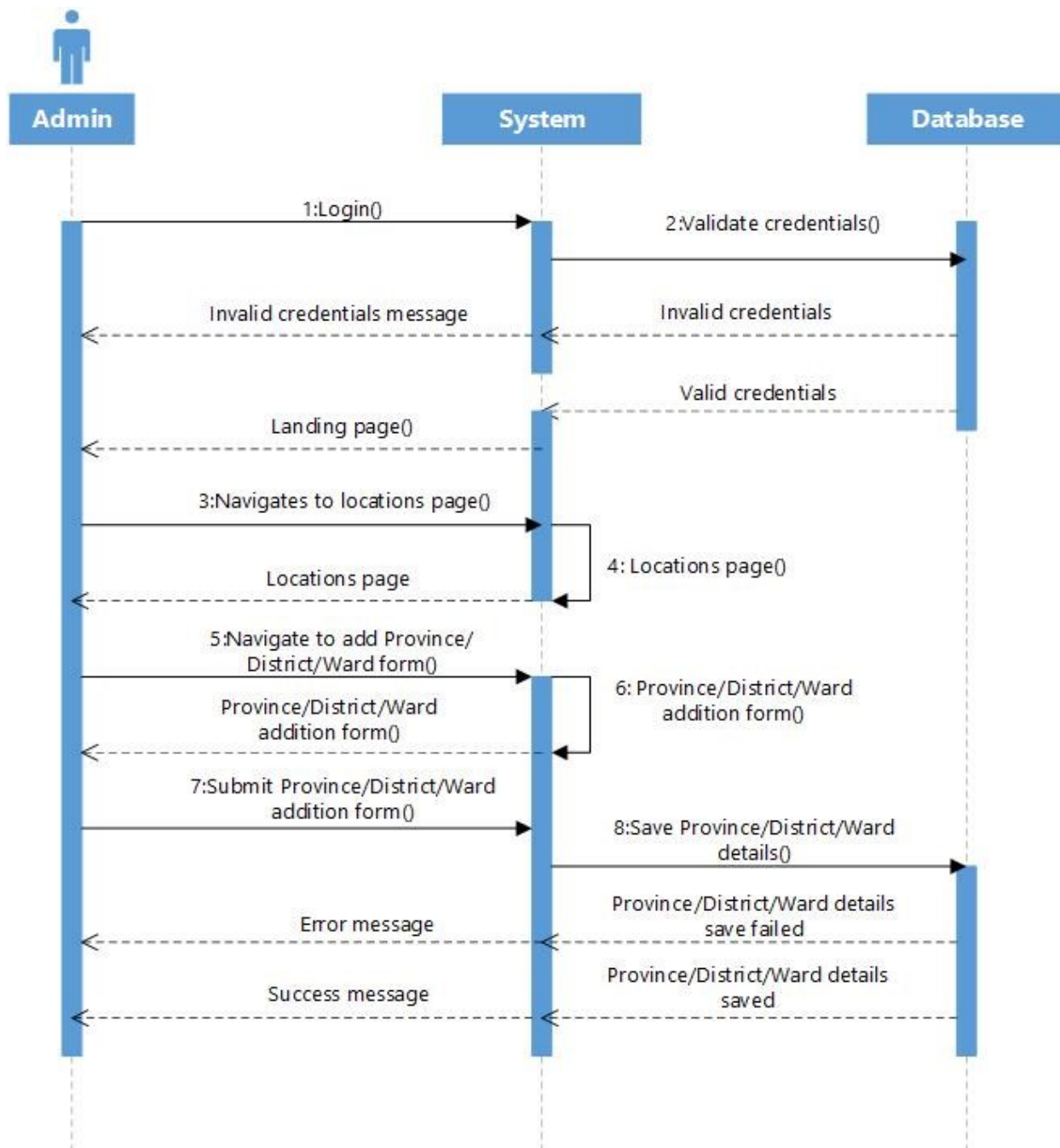


Figure 26: Manage Province/District/Ward Sequence Diagram

Manage Ward Thresholds

For warning alerts to be pushed by the system to the farmers, FAW experts and other stakeholders, there is need to set the threshold for each ward. This is the one of the functions that the administrator will be performing. Figure 31 show the actions and events that will be taken during the process of setting and adjusting of ward thresholds.

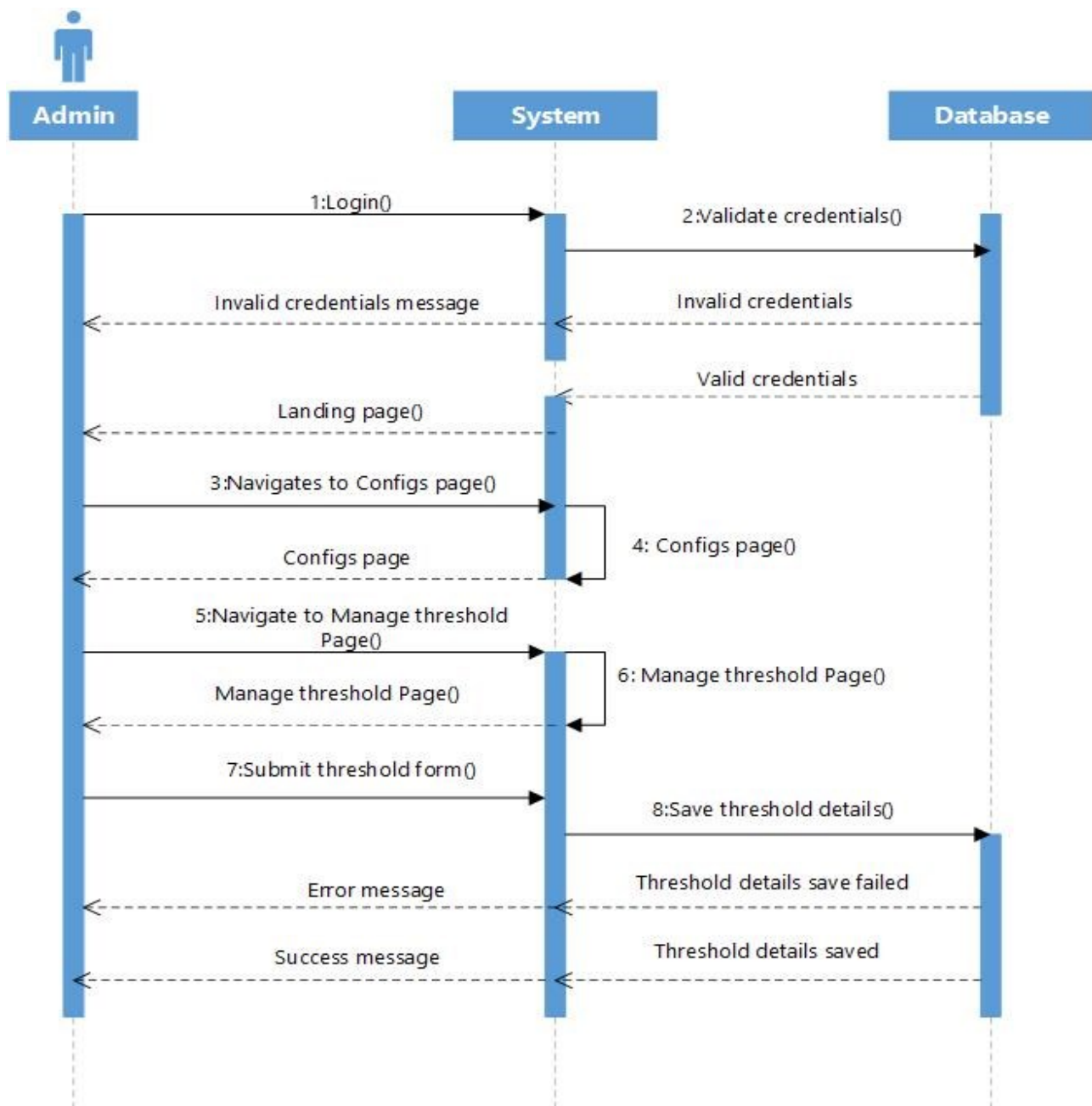


Figure 27: Set ward threshold sequence diagram

Manage Users

Management of system users is the function of the system administrator. Some of the users of the system will include but not limited to Farmers and FAW experts. The function of managing users will include registration of users, update of user details, assignment of roles to users, creation of user roles, and assignment of system permissions to user roles and so on. Figure 32 shows the sequence diagram that describes the actions and events that will occur during the process of user management.

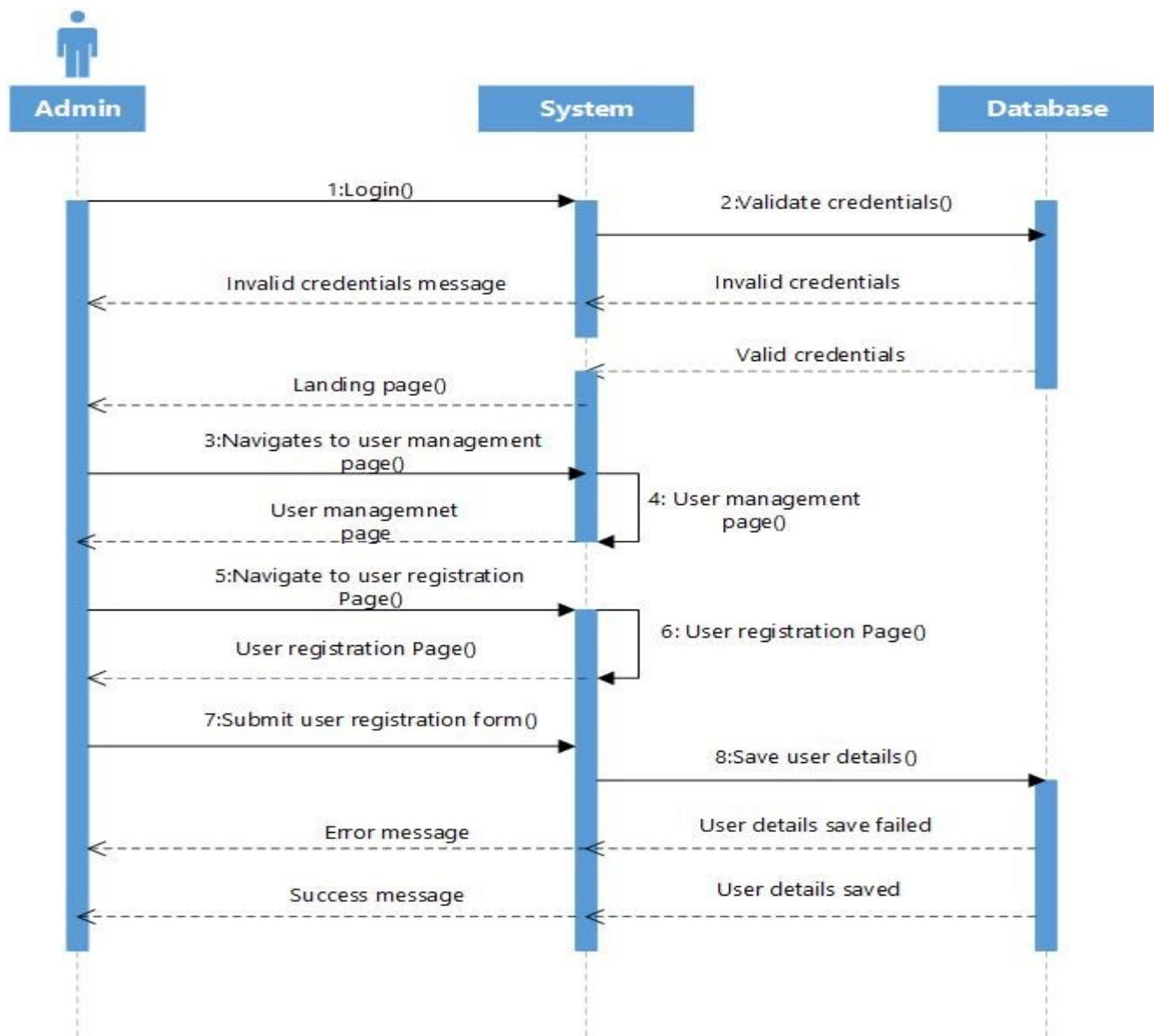


Figure 28: User management Sequence diagram

Predict and classify image

Once the request is received from the trap, the system validates the request making sure that all the required parameters are included in the request. Once the request is validated, the system calls the flask server that classifies and makes prediction. If the image is FAW, the request data is loaded in the database. Figure 33 shows the sequence diagram of actions and events that occurs in the process of predicting and classifying the image.

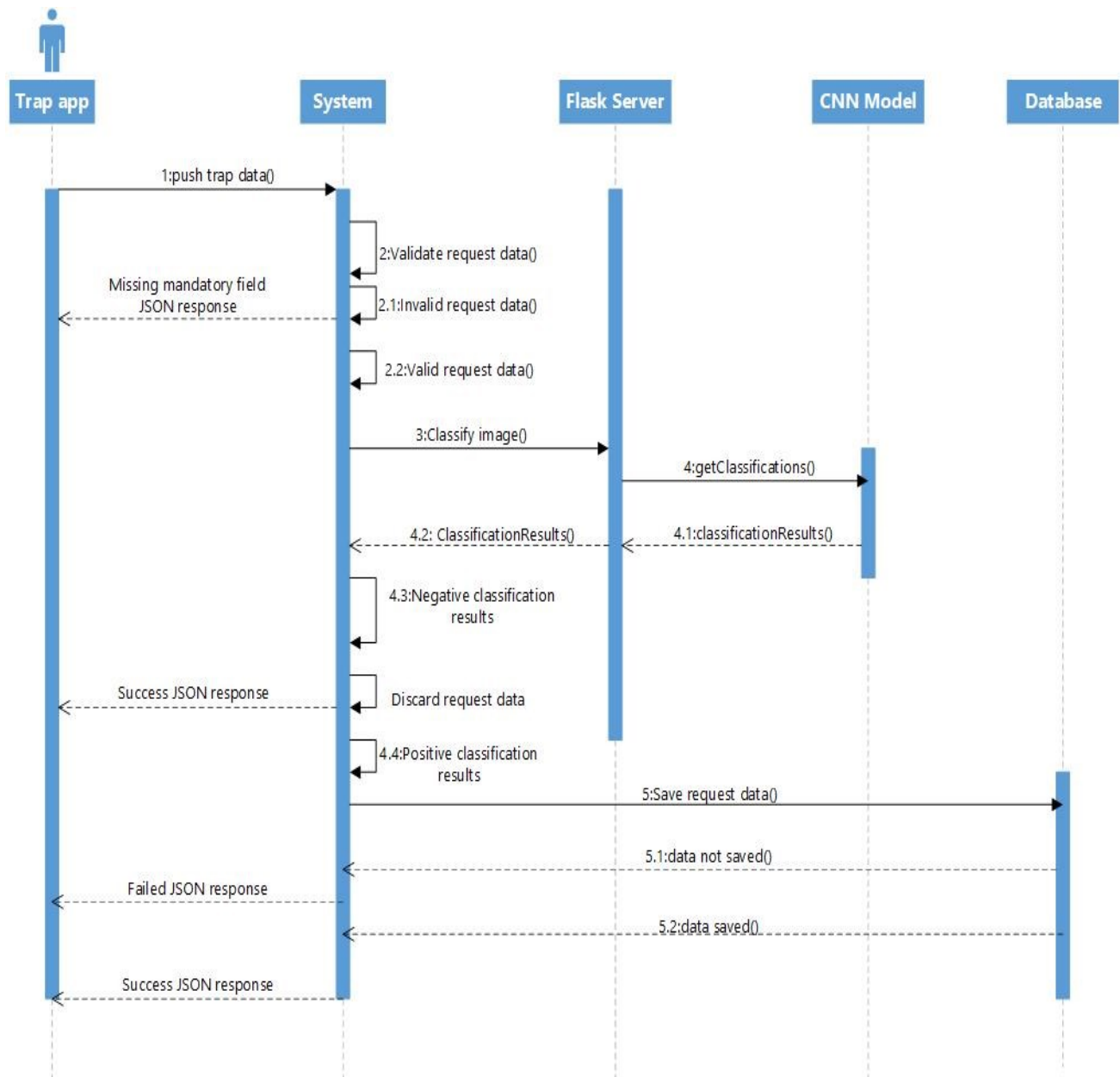


Figure 29: Predict and classify image sequence diagram

Send Notifications

The pushing of notifications will be done by Cron jobs that will be configured to run at specific times of the day. A Cron job is simply a service kind of application that runs in the background at specific times as configured. After waking up at the configured time, the Cron job will look into the database if there are any wards with counts greater than the set threshold. If the wards exists, the Cron job pushes the alerts to farmers and FAW experts per ward. Figure 34 below shows the actions and events that will occur during the process of sending the alerts.

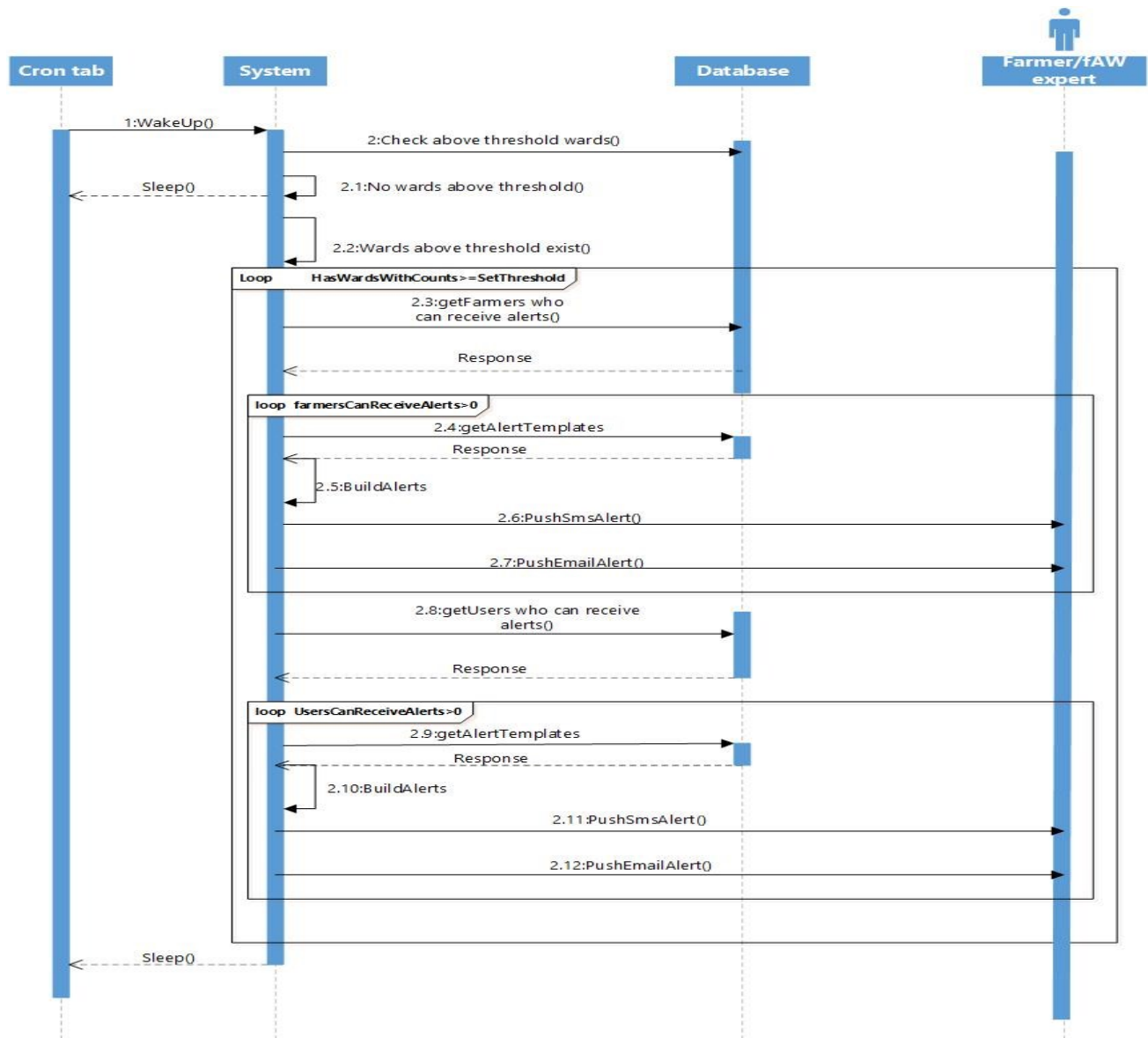


Figure 30: send notifications sequence diagram

3.7.3.3.2. Mobile app

Figure 35 shows the only sequence diagram for the mobile app with actions and events that can be documented. The sequence of events allow the farmer to use the mobile app to update their personal details. Farmer has to login first then navigate to the profiles page where they choose update profile. Farmer fills in details they think are wrong or missing and submits. Then the app invokes the ipms API with the farmer details request which is saved in the database and responds with a success JSON response.

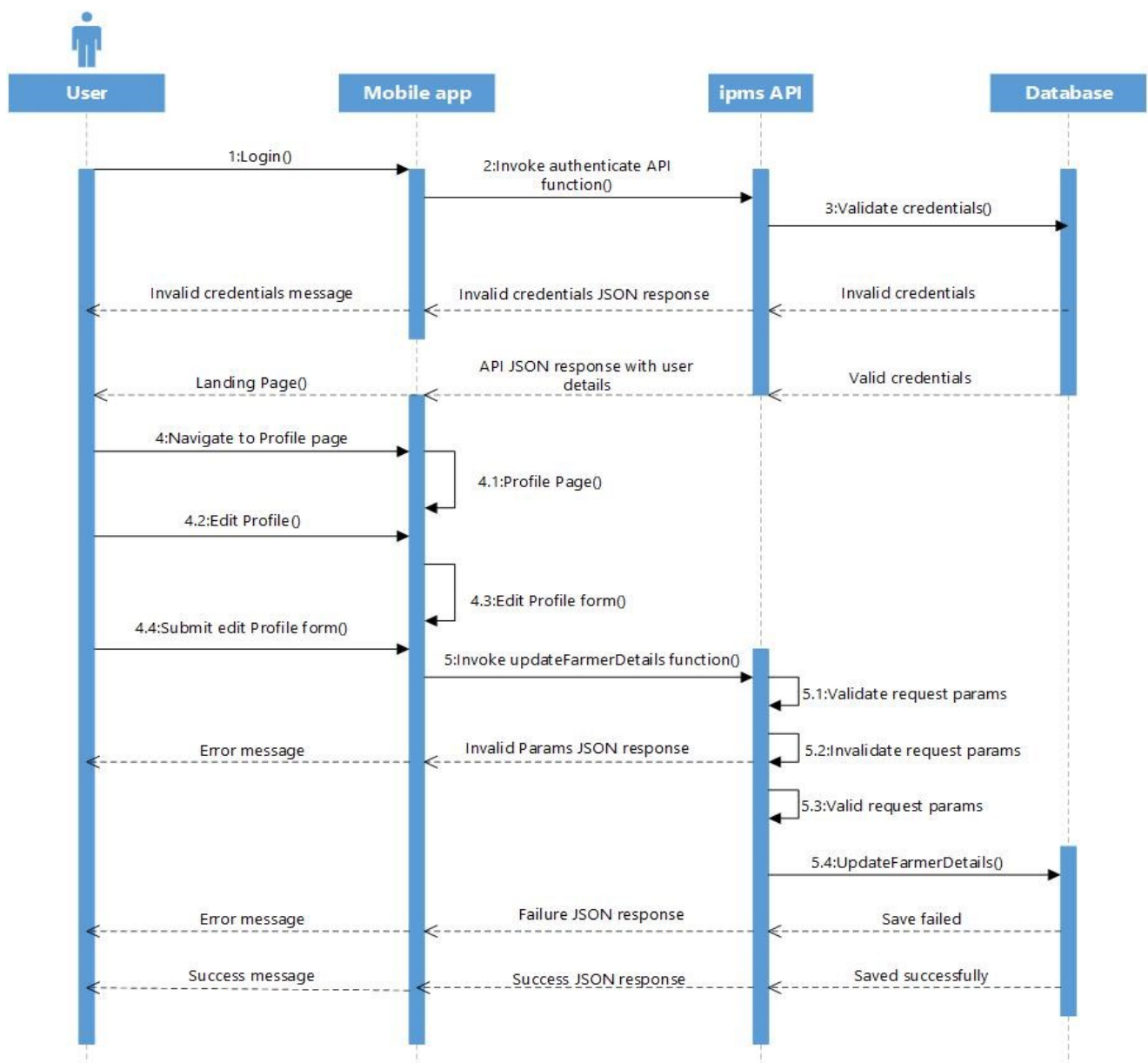


Figure 31: Update farmer details sequence diagram

3.7.3.4. Class Diagram

Class diagrams describe the objects and their static structures. They comprise of real-world objects or conceptual classes of the proposed system. They show the relationship between classes and the attributes that make up the classes. Figure 36 illustrates the class diagram for the web interface only.

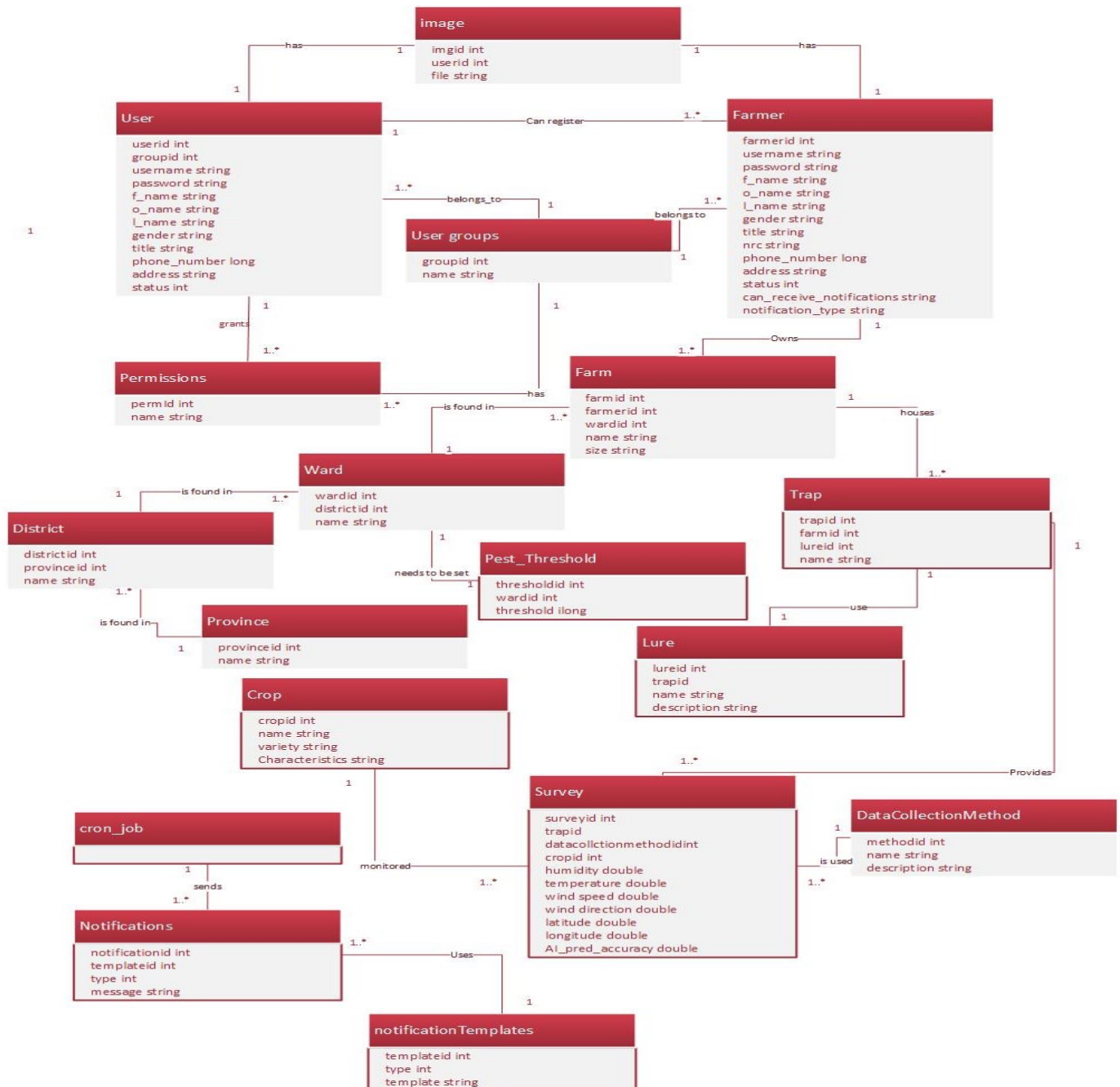


Figure 32: Web interface class diagram

3.7.4. Data Models

Data models comprises of the ER model and the data dictionary. The ER diagram shows how entity sets stored in the database are related. An entity is a basic object of data. An ER diagram gives an illustration of the logical structure of the database. Figure 37 gives the study database ER diagram and Appendix 2 gives the data dictionary of each table.

3.7.4.1. E-R Diagram

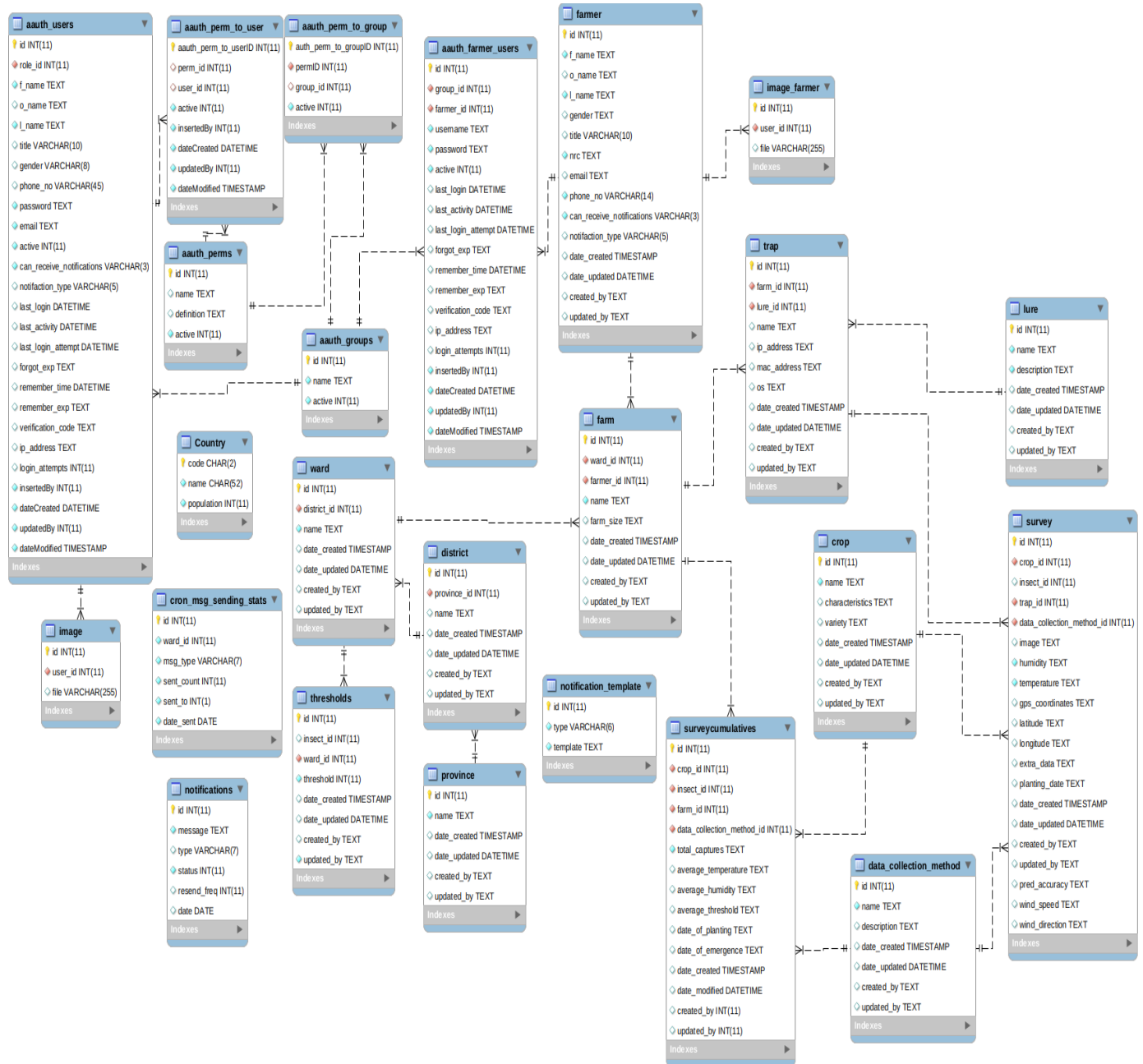


Figure 33: Entity Relationship Diagram

3.7.5. System Implementation

3.7.5.1. Web application

The web interface was developed using PHP using a Model View Controller (MVC) framework called Yii2. The development of the application was done on a machine running Ubuntu OS 18.11. PHP, MySQL, Apache server and other dependents were installed by installing a LAMP stack which is a suite of web service stacks made up of Linux operating system, Apache server, MySQL and PHP. For database design, a graphical user interface called MySQL workbench was used following the ER diagram and the data dictionaries given in this chapter. To develop the web interface, an Integrated Development Environment (IDE) called NetBeans 8.2 was used by following the system methodology and designs presented in this chapter. For rapid prototyping Yii framework was used which enables one to generate all the required forms within minutes and the generated forms amounts to a usable system. The study made use of GitHub as a deployment, collaboration and backup tool. The development machine was used as a test machine and therefore for production environment, the study used Google cloud server running the same OS as the development machine. The setup on the development machine was replicated on the cloud server and the web app was deployed on the cloud server by cloning the source files from GitHub.

3.7.5.2. Mobile application

The mobile app developed is an Android app with IOS being considered for future works. The app was developed using Android studio running on a windows 10 development machine.

3.8. Summary

The chapter highlighted the methods and materials that were used in the development of the web interface and mobile app and the methods and materials used in the training and implementation of the Convolutional Neural Network. The chapter presented the designs that were used in the development of the components needed in this study. The system architecture was also presented and discussed.

CHAPTER FOUR

RESULTS

4.1. Introduction

The chapter discusses the major findings of the research for the development of an automated fall armyworm monitoring and warning system using Artificial intelligence techniques. Main focus of the study was to develop an automated system that will assist farmers and other stakeholders in the monitoring of the fall armyworm. The chapter presents results for the training of the CNN model and the development of the web interface and the mobile app.

4.2. Convolutional Neural Network training

Using the data collection methods and training methods presented in chapter three, the model was trained and the results are presented and discussed here.

Three trials were conducted to get the best learning rate and training steps, out of which two gave the best learning rates, 0.01 using 1500 training steps and 0.001 using 5500 training steps. The learning rate 0.01 with 1500 steps was used to train another model using pixilated FAW moth images just to test the performance of the model when tested with clear images. The results of the three tests were grouped into groups according to the model that performed better on all categories of test data.

A. Clear faw images, 0.01 learning rate, and 1500 training steps

Classification accuracies of round one model on all the three test categories are given by Table 12. The model gave an average classification accuracy of 82% on actual clear FAW images, an average of 70.84% on augmented images and an average accuracy of 32.1% on false positives. The predication accuracy was low in augmented images due to the fact that the images were distorted with most of them being pixilated. The model performed well in false positives which were images of other insects not FAW and were not used in the training dataset.

Table 12: CNN model round 1 prediction accuracies

#	Augmented %	Actual %	False positives %
1	97.8	90.1	54.7
2	84.3	98.2	49.8
3	97.6	97.6	24.1
4	19	90.9	47.5
5	40	99.4	42.9
6	95.8	93.7	80
7	31.6	51.7	45
8	89.4	94.2	0.1
9	77.9	94.4	71.1
10	47	43.5	0.2
11	64.1	74.8	7.4
12	87.9	86.2	1.2
13	79.6	69.8	52.7
14	92.6	47.9	1.3
15	58	97.5	2.7

Figure 38 and 39 shows the Tensorboard training graphs for round 1 model training. Figure 38 gives the training accuracy graph which became constant around 0.99 accuracy meaning that the model could not learn any more new features.

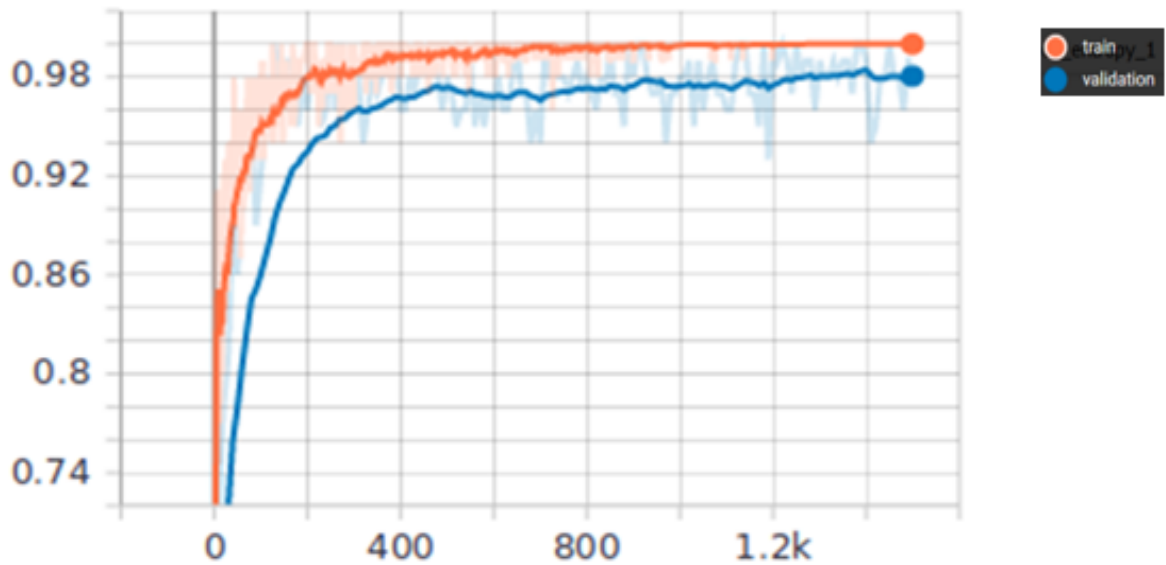


Figure 34: CNN model Round 1 training accuracy

Figure 39 is the round 1 cross entropy. From the graph, the entropy steadily reduced to zero as the epoch approaches 1500 indicating that the model training was progressing very well.

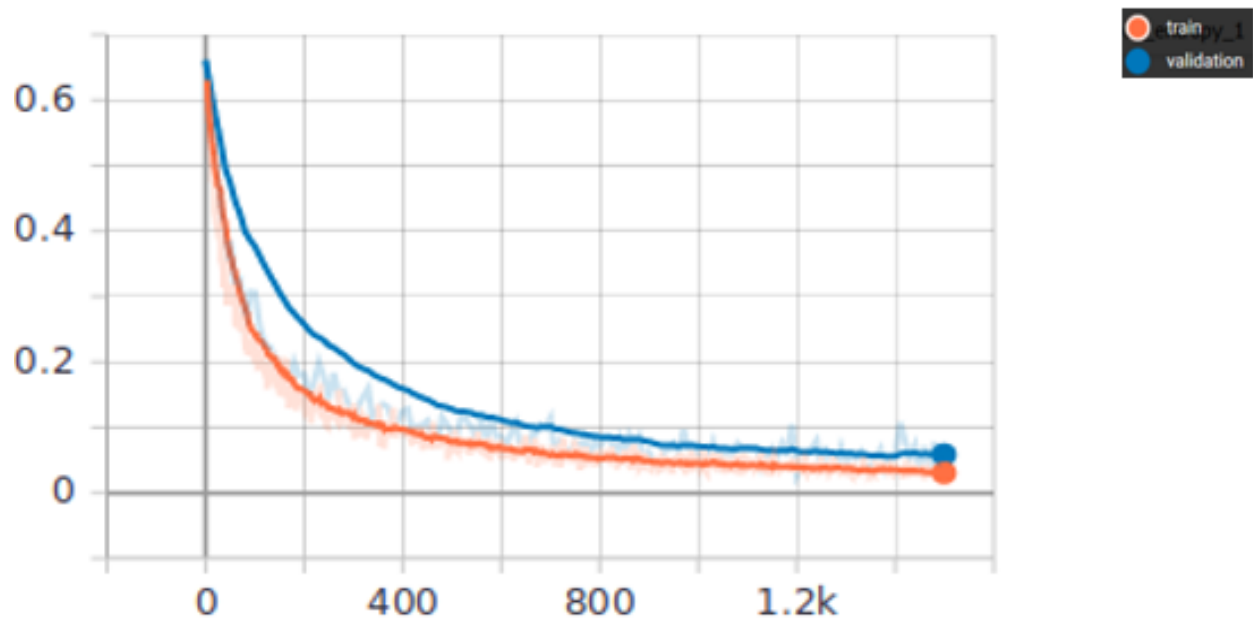


Figure 35: CNN model round 1 cross entropy

Figure 40 shows the prediction accuracies as a graph. The graph shows that the model performed well with actual images as compared to augmented and false positives.

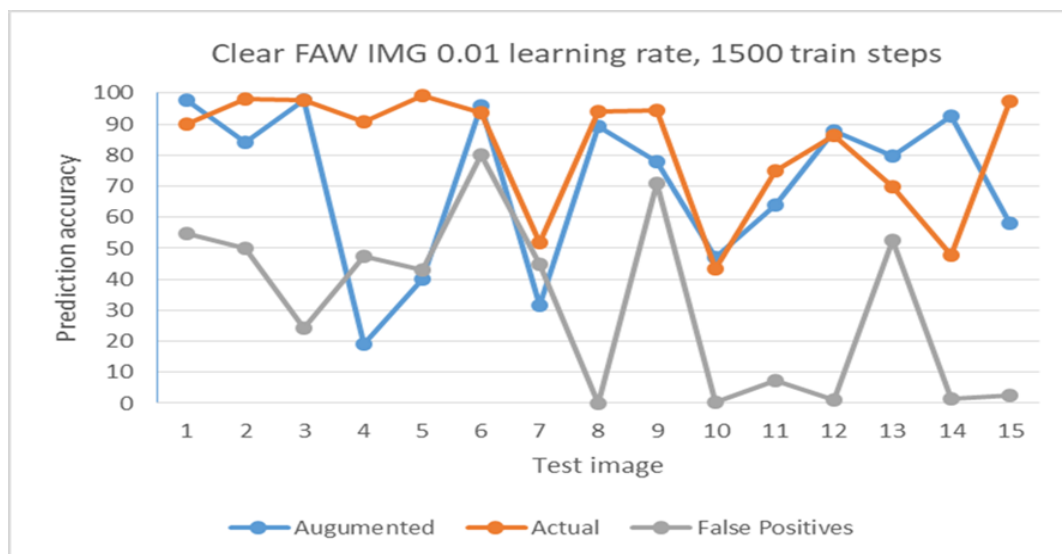


Figure 36: CNN model Prediction accuracy graph

B. Clear faw images, 0.001 learning rate, and 5500 training steps

Table 13 shows the prediction accuracies of the second round model and gave average prediction accuracies of 75.8% on actual clear FAW images, 76% on augmented images and 48.8% on false positives. The drop in the accuracies on actual images might be attributed to the rate at which the model was learning as we reduced the learning rate.

Table 13: CNN model round 2 prediction accuracies

#	Augmented %	Actual %	False positives %
1	95.8	90.1	5.2
2	80.9	96.3	33.8
3	95.2	67.5	20
4	12.2	74.2	19.1
5	41.1	91	4
6	86.5	91.7	2.9
7	73.5	50.3	93.4
8	77.5	89	46.8
9	80.9	98.7	94.5
10	85	17.9	47.8
11	77.5	89.2	94
12	89	95.7	56.7
13	93.6	39.3	73.5
14	95.2	50	87.4
15	61	96.7	53.1

Figure 41 and 42 shows the Tensorboard training graphs. Figure 41 gives round 2 training accuracy. The graphs gives a visual representation of how the training is occurring and it shows that model could not learning any more as we approached accuracy 0.98.

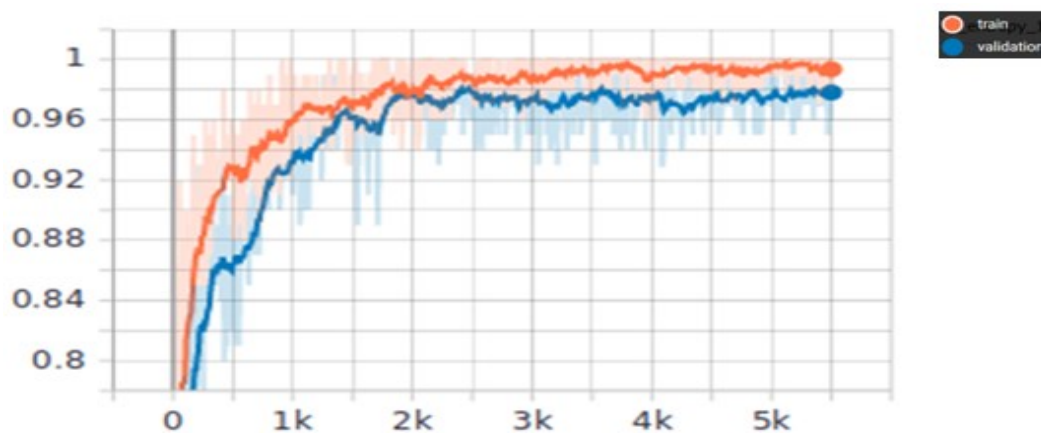


Figure 37: CNN model round 2 training accuracy

Figure 42 gives the round 2 cross entropy. From the graph, the entropy steadily reduced to zero as the epoch approaches 5500 showing that the training of the model was occurring well.

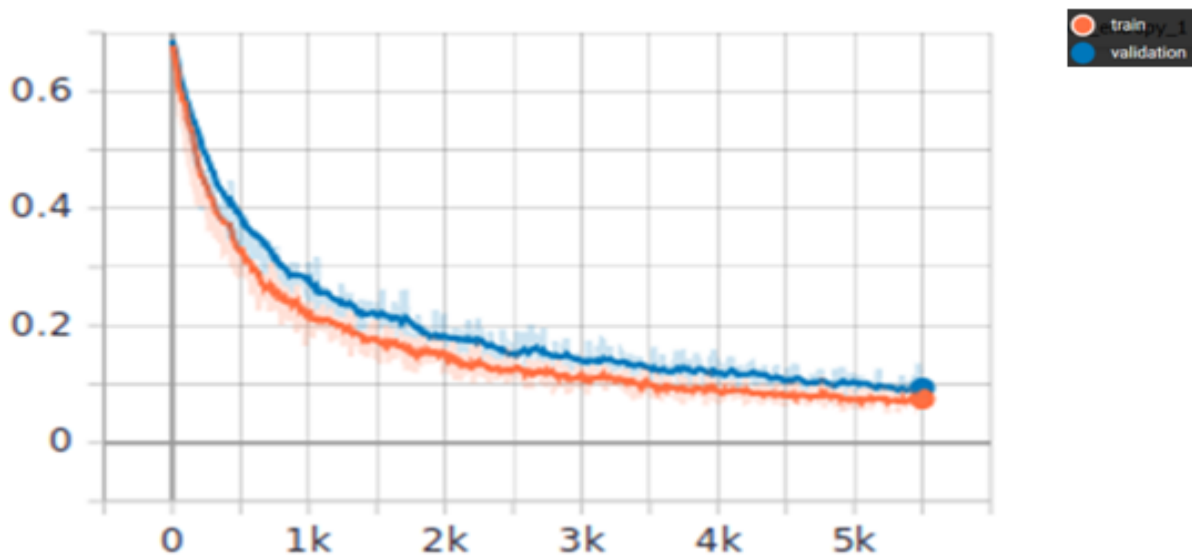


Figure 38: CNN model round 2 cross entropy

Figure 43 shows the round 2 prediction accuracies as a graph. The graph shows a reduction in the prediction accuracies which might be attributed to the reduced learning rate and reduced training steps.

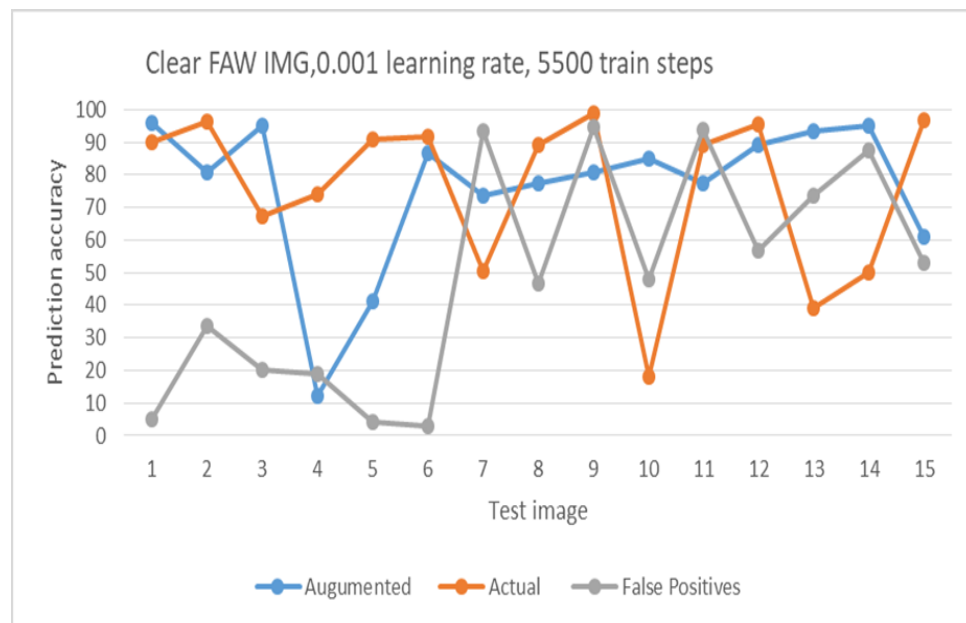


Figure 39: CNN Model round 2 prediction accuracy graph

C. Pixilated (blurred) faw images at 10 pixels, 0.01 learning rate and 1500 training steps

This model was trained in order to find out how the model would perform if trained on blurred images and tested using the test dataset using the learning rate and training steps of the model with a better average accuracy which was round 1. Table 14 shows the prediction accuracies of the third round model. The model performed badly in all categories when tested on all three test categories as shown by Figure 44.

Table 14: CNN model round 3 prediction accuracies

#	Augmented %	Actual %	False positives %
1	94.3	17	93.2
2	3.9	7.8	99.7
3	66	1.6	99.6
4	8.6	62.6	1.2
5	13.3	9.2	1.9
6	99.1	28.5	9.7
7	12.6	11.2	93
8	25	1.2	53
9	9.6	1.3	14.8
10	53.9	16	42
11	99.8	9.4	100
12	40.7	0.3	100
13	6.5	11.8	97.5
14	39.3	12.6	0.5
15	51	14.8	0.7

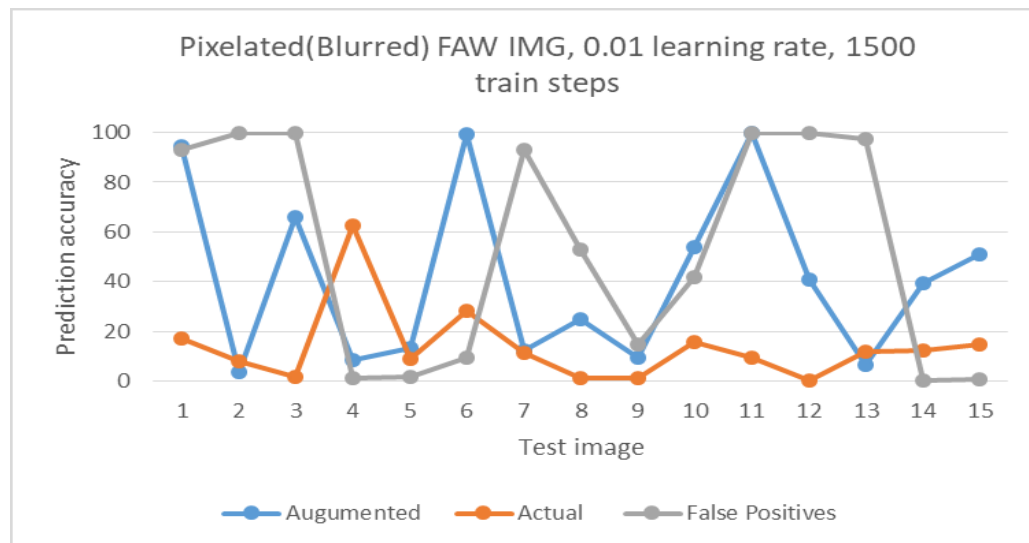


Figure 40: CNN model round 3 prediction accuracy graph

4.3. System Prototype development

2.1.8. 4.3.1. Web Interface

The web interface refers to the dashboard and Application Programming Interface (API). The application was developed using Yii2 framework and was deployed on a Google cloud server. The dashboard is meant to be used for registration of farmers, report generation, data visualizations among other things. The API is meant to receive requests from the trap and mobile app and for invoking the CNN model when a request from the trap contains an image. The dashboard allows FAW experts and farmers to login by providing separate URLs for each user. For FAW experts to access the system, they have to navigate to <http://IP:PORT/ipms/backend/web/> which will present them with a landing page. For farmers to access the system they have to navigate to <http://IP:PORT/ipms/frontend/web/> where they will be presented with a landing page.

Only FAW experts will have to know the required parameters to configure the trap in order to push trap data to the API. For pushing trap data, the API expects a request with a content type of *'application/x-www-form-urlencoded'* and responds with a JSON response. The use of a URL encoded request makes it easy for any application to push an image as a parameter. The expected request parameters are as shown in Figure 45 where username and password are for authentication and are provide to the FAW expert at the time of setting up the trap, crop_id, trap_id, data_collection_method_id should be the ones configured in the system, humidity, temperature, wind speed and wind direction should be read at the trap site and cannot be empty, longitude and latitude are GPS coordinates of the current location of the trap and cannot be empty and file is an image and cannot be empty. The API responds with a JSON response containing status, statusDescription which describes what the status and data which is an array of other details as shown by Figure 46.

```

- $data = [
    'username'=>'',
    'password'=>'',
    'crop_id' => '1',
    'trap_id' => '1',
    'data_collection_method_id' => '1' ,
    'humidity' => '30',
    'temperature' => '26',
    'latitude' => '14.24343',
    'longitude' => '-15.2656',
    'file' => $image,
    'wind_speed'=>'38',
    'wind_direction'=>'40'
];

```

Figure 41: Sample ipms API trap request

```

{
    'status':'100',
    'statusDescription':'Success',
    'data':[]
}

```

Figure 42: Sample ipms API trap response

For the mobile app, the API expects a JSON request and returns a JSON response similar to the one shown in Figure 46.

4.3.1.1. Sample Dashboard Screen shoots

4.3.1.1.1. FAW Expert

a) Landing Page

Figure 47 shows the envisioned site page for the system. The page is the first page that the user/farmer sees when they try to access the system and then choose login in order to login into the system. The image shown by Figure 47 is from the first version.

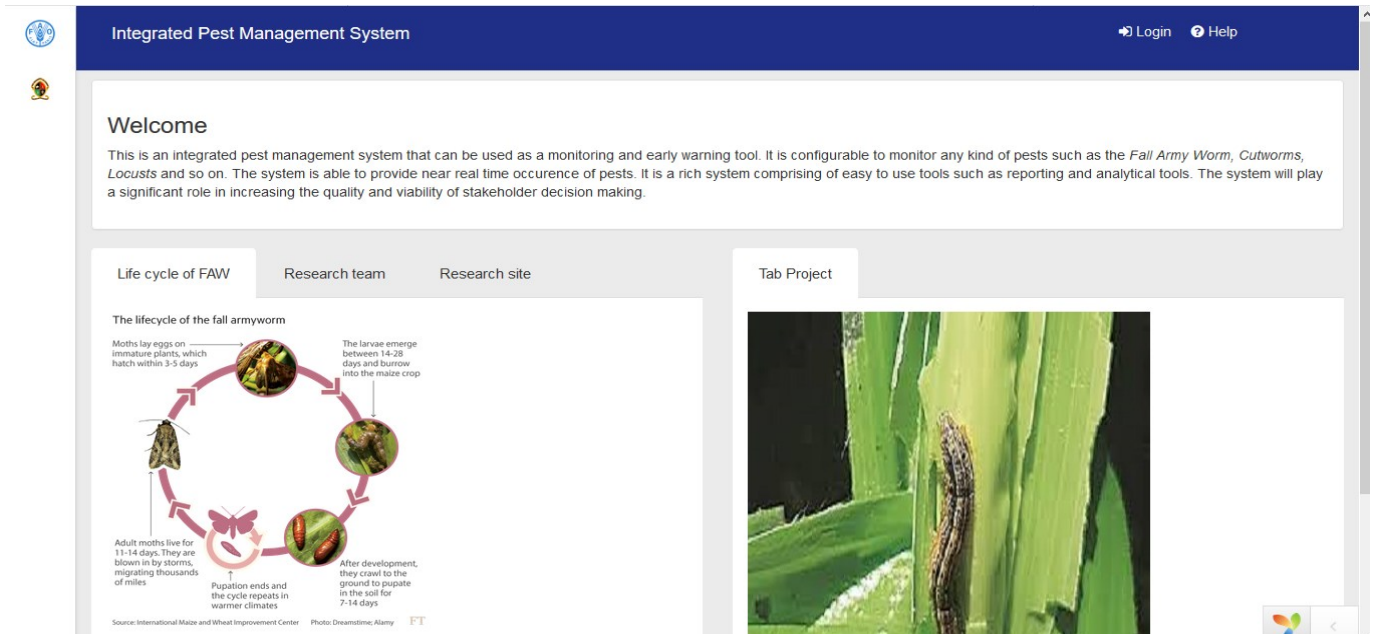


Figure 43: Sample landing page for both FAW expert and farmers

b) Login

The Login page gives the FAW expert access into the system by entering their password and username. In the case of the farmer and password and email in the case of the FAW expert. Figure 48 shows a sample login page for FAW expert.

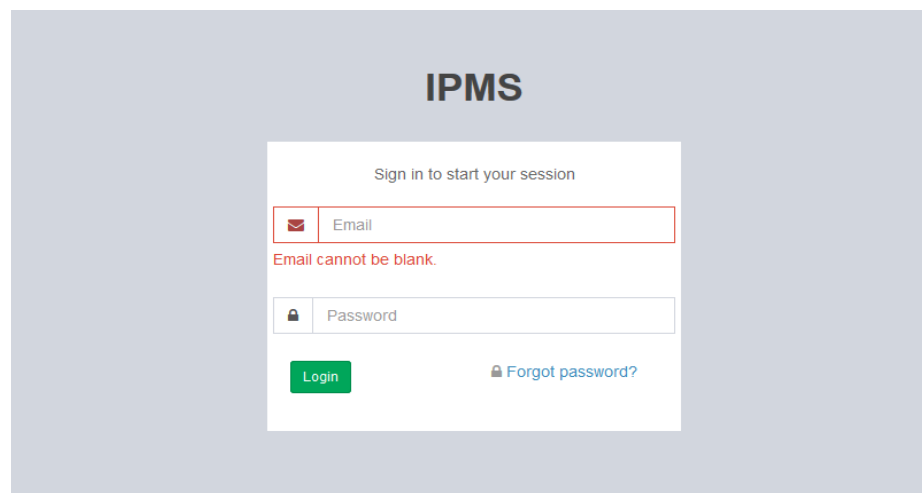


Figure 44: Sample FAW expert login page

c) Dashboard

Figure 49 shows a sample dashboard when the expert has logged in. The dashboard gives captures per month, percentage captures per province for the current month and the top 10 traps that have

high captures for the current day. It also shows the number of registered farmers, how many crops and traps are being monitored and how many farms have been registered.

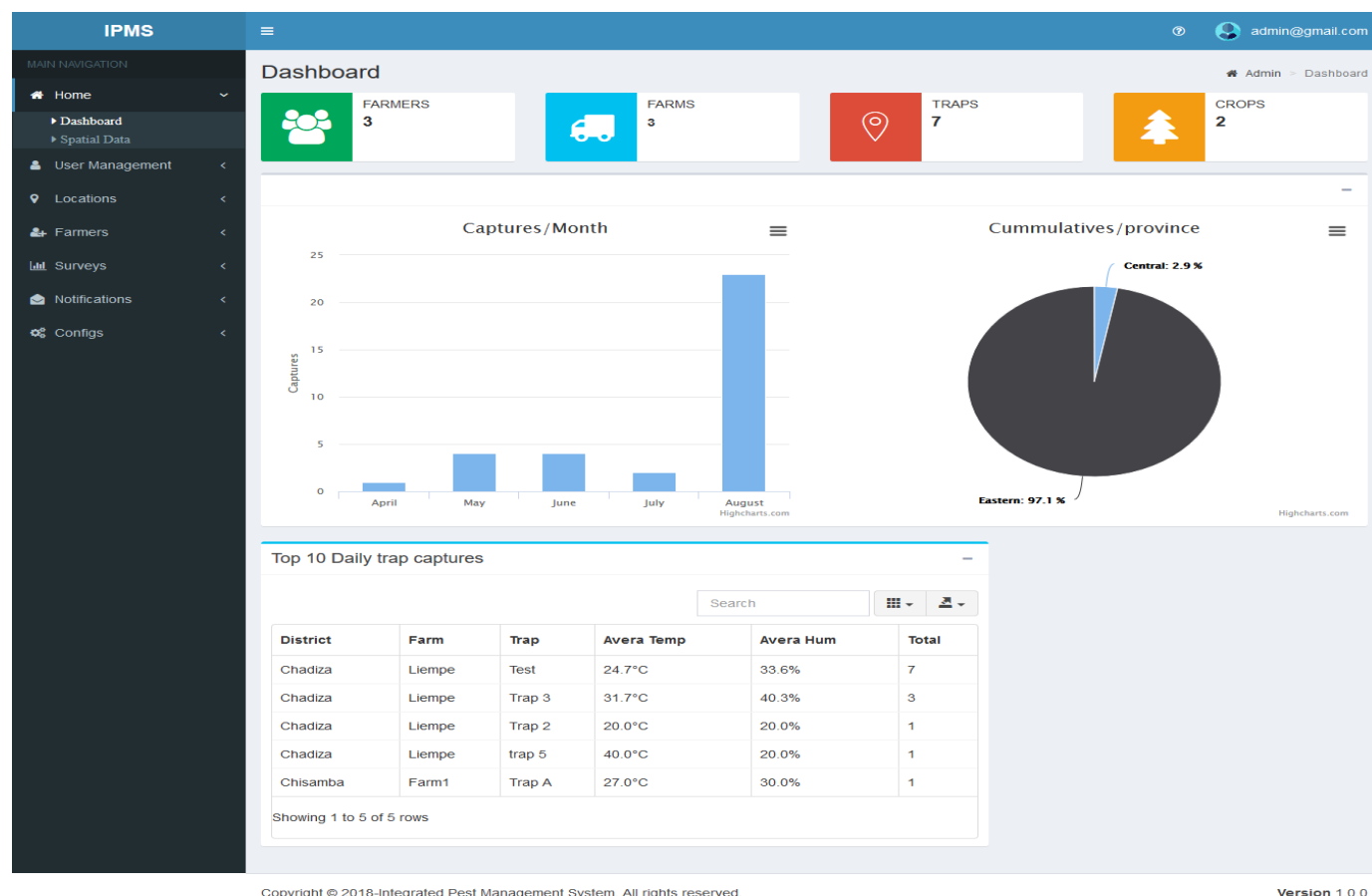


Figure 45: Sample FAW expert dashboard

d) Spatial data

Figure 50 and 51 shows the spatial data pages. Figure 51 is a map showing the location of the configured traps with the related details and their status. When a trap icon is bouncing means that the trap has captures above the set threshold for that ward. Figure 52 is the map of Zambia showing the distribution of the insect. Green shows that the traps in the shaded province have not yet captured any insects or has no trap setup yet. Orange gives a warning buildup of the insect and red gives an infestation in the province.

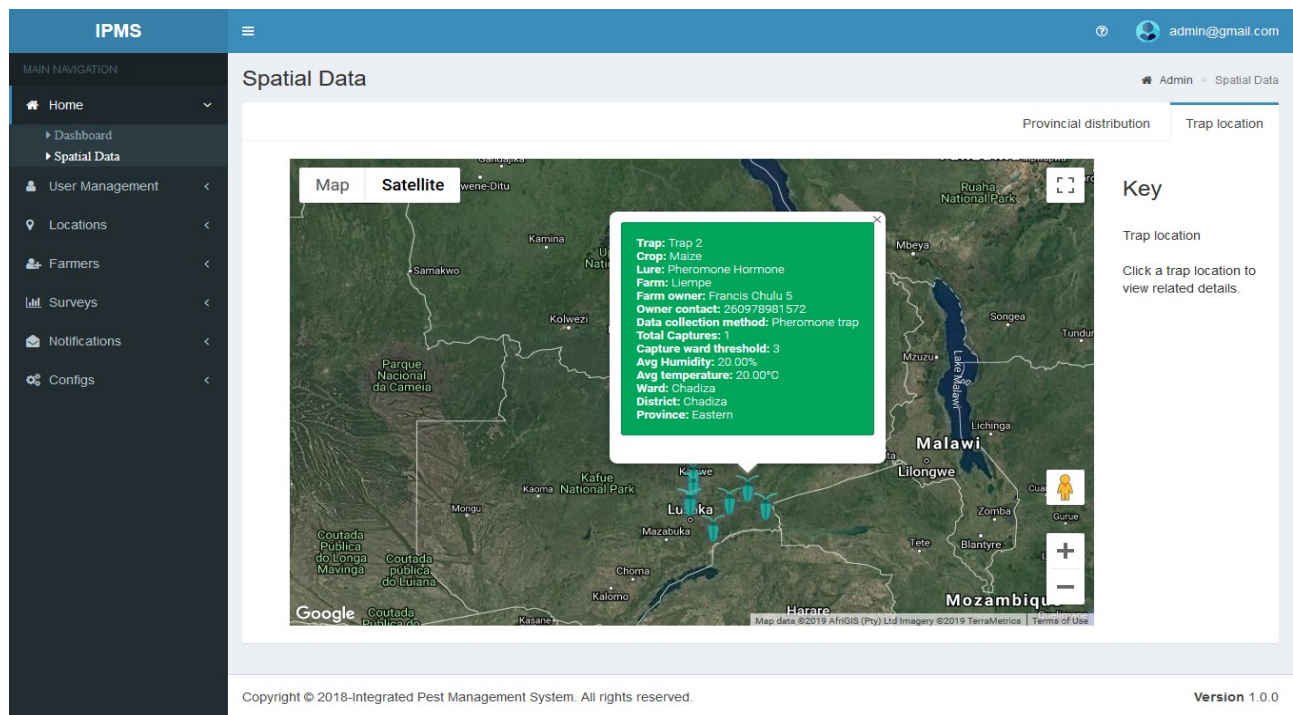


Figure 46: Spatial data Google maps

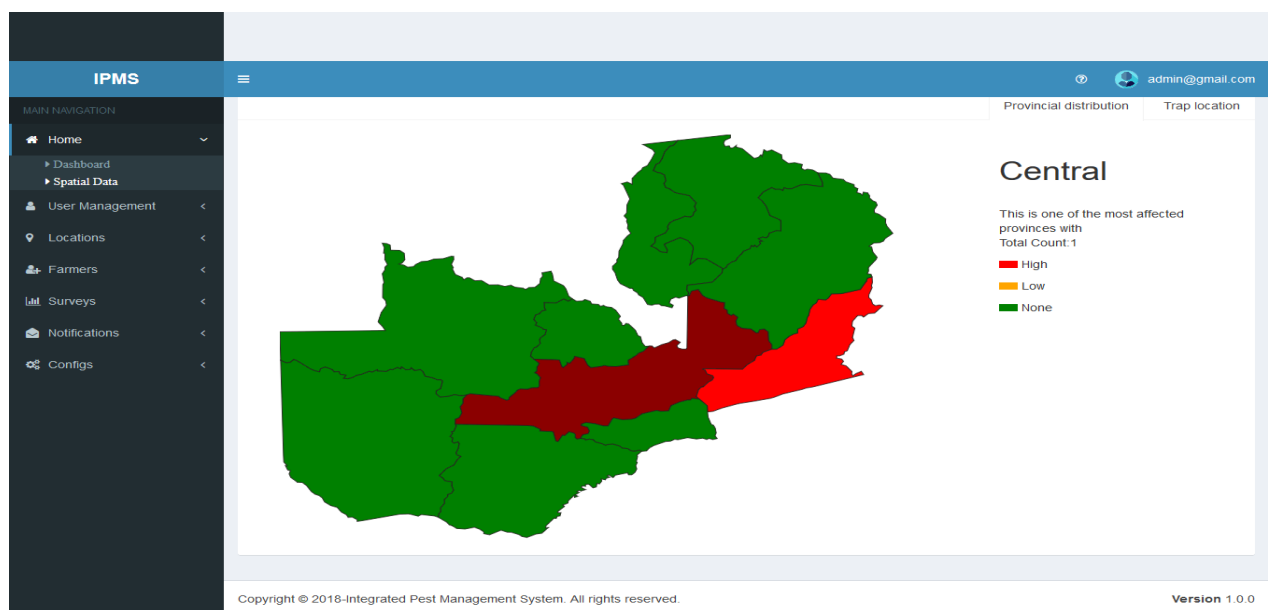


Figure 47: Spatial data Provincial distribution

e) User registration

Figure 52 shows the user management landing page listing all the users and the groups they belong to. Clicking add user takes the user to the user registration page as shown by Figure 53.

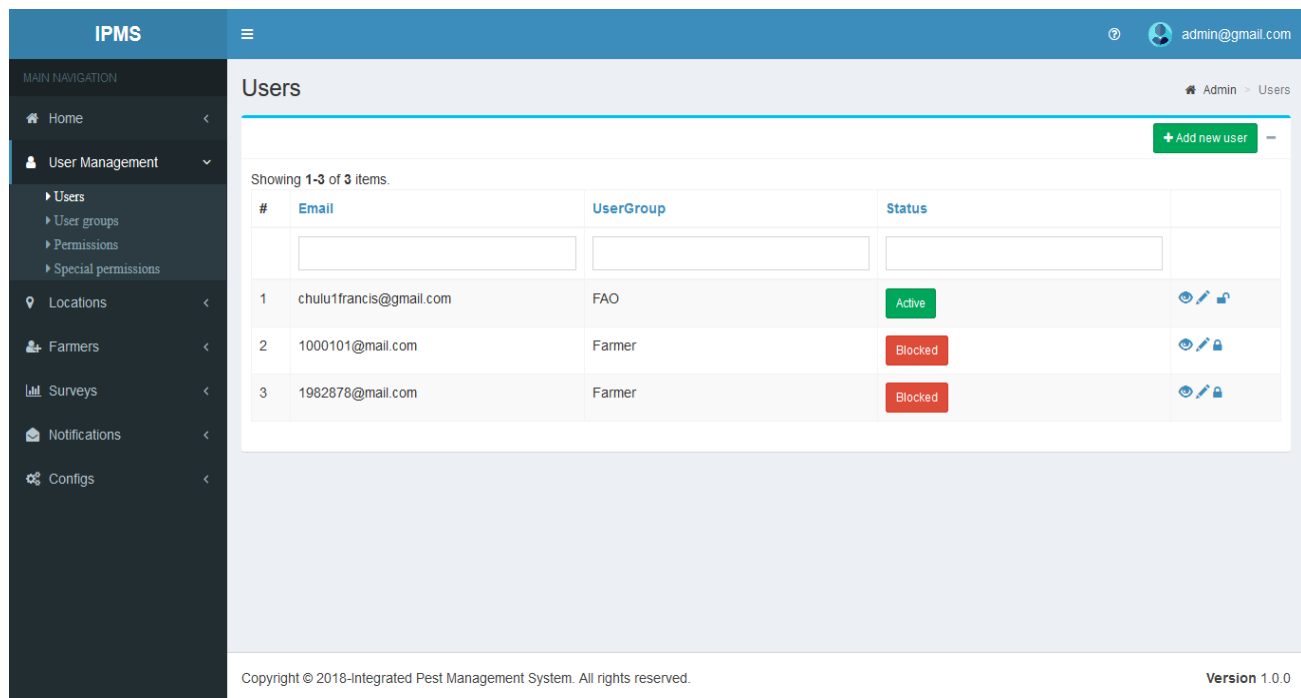


Figure 48: User management landing page

IPMS

admin@gmail.com

MAIN NAVIGATION

- Home
- User Management
 - Users
 - User groups
 - Permissions
 - Special permissions
- Locations
- Farmers
- Surveys
- Notifications
- Configs

Add User

Admin > Users > Add

First name

Other names

Surname

Gender

☐ Male ☐ Female

Title

Submit

Email

Phone No

UserGroup

Please select group

Can Receive Notifications

☐ Yes ☐ No

Notification Type

☐ Email ☐ SMS ☐ Both

Copyright © 2018-Integrated Pest Management System. All rights reserved. Version 1.0.0

Figure 49: User registration form

f) Farmer Registration

Figure 54 shows the landing page when the user navigates to farmer management listing all the farmers registered in the system. Clicking add farmer takes the user to the farmer registration page

as shown by Figure 55. After registering the farmer, the user can add farms belonging to the farmer. Figure 56 shows the details of the farmer after registration.

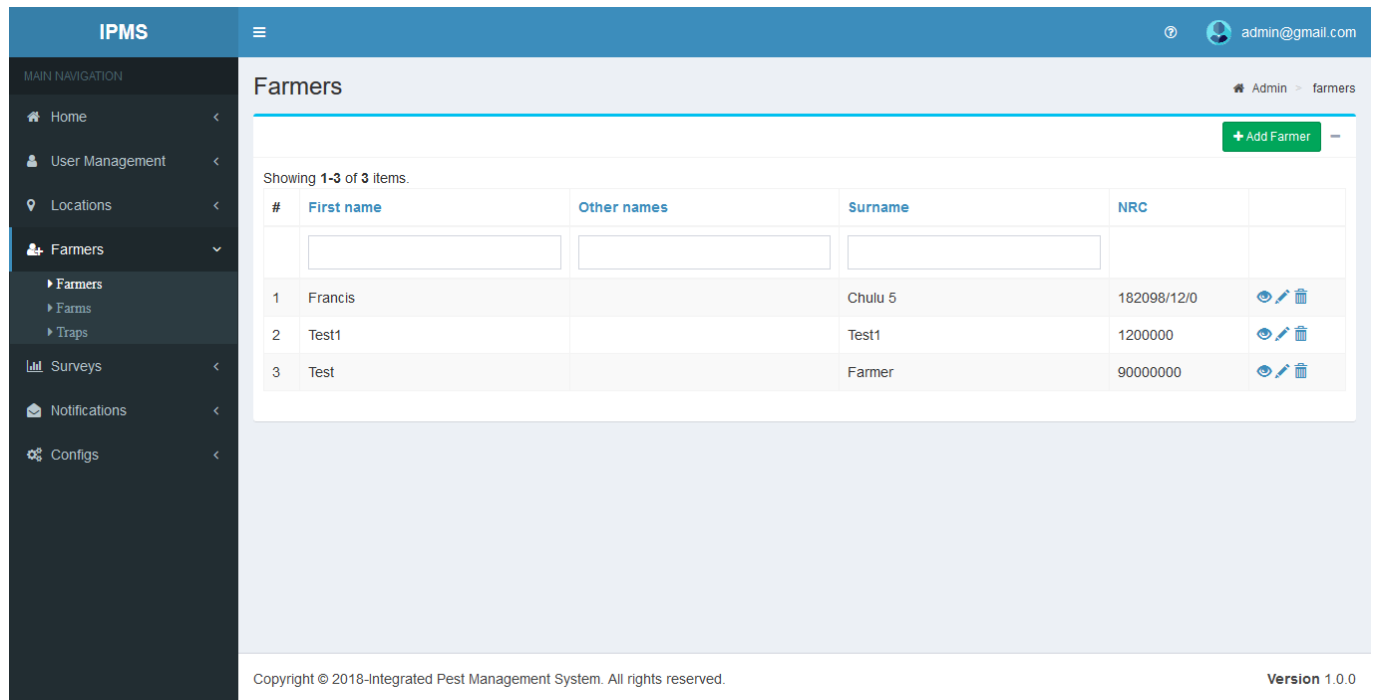


Figure 50: Farmer management landing page

The screenshot shows the 'add farmer' form. The fields are organized as follows:

- First name:** Text input field.
- Other names:** Text input field.
- Surname:** Text input field.
- NRC:** Text input field.
- Phone No:** Text input field with a dropdown arrow.
- Title:** Dropdown menu with 'Select title' as the placeholder.
- Gender:** Radio buttons for 'Male' and 'Female'.
- Can Receive Notifications:** Radio buttons for 'Yes' and 'No'.
- Notification Type:** Radio buttons for 'Email', 'SMS', and 'Both'.

A green 'Save' button is located at the bottom left of the form area.

Figure 51: Farmer registration form

IPMS

MAIN NAVIGATION

- Home
- User Management
- Locations
- Farmers
 - Farmers
 - Farms
 - Traps
- Surveys
- Notifications
- Configs

View Farmer

Admin > Farmers > Mr. Francis Chulu 5

Mr. Francis Chulu 5
Farmer user

Details

Personal

NRC: 182098/12/0
Gender: Male
Phone #: 260978981572

Login

Username: 90000000
Password: *****
Status: **Active**

Other

Total farms: 1
Total Traps: 5

Notify Farmer Yes

Notification Type email

Created by admin@gmail.com

Updated by admin@gmail.com

Date created 2019-01-03 18:51:13

Date updated 2019-07-20 20:30:26

Update **Deactivate login**

Copyright © 2018-Integrated Pest Management System. All rights reserved. Version 1.0.0

Figure 52: View farmer page

g) Surveys

Survey refers to the data received from the trap and the image has been classified as FAW. Figure 57 shows the list of surveys that the system has. Clicking the view icon gives the full details of the survey as shown by Figure 58.

IPMS

MAIN NAVIGATION

- Home
- User Management
- Locations
- Farmers
- Surveys
 - Realtime Surveys
 - Cumulative Surveys
- Notifications
- Configs

Surveys

Admin > Realtime Surveys

+ Add Survey

Showing 1-10 of 34 items.

#	Crop	Trap	Humidity	Temperature	Actions
1	Maize	Test	30%	26°C	
2	Maize	Test	30%	26°C	
3	Maize	Test	30%	26°C	
4	Maize	Test	30%	26°C	
5	Maize	Test	30%	26°C	
6	Maize	Trap 2	20%	20°C	
7	Wheat	Trap 3	64%	27°C	
8	Maize	Trap 3	27%	28°C	
9	Maize	Test	25%	20°C	
10	Maize	Test	60%	23°C	

Copyright © 2018-Integrated Pest Management System. All rights reserved. Version 1.0.0

Figure 53: Trap surveys

IPMS

admin@gmail.com

MAIN NAVIGATION

- Home
- User Management
- Locations
- Farmers
- Surveys
 - Realtime Surveys
 - Cumulative Surveys
- Notifications
- Configs

View Survey

Admin > Surveys > View Survey

Update Delete

Crop	Maize
Trap location	Trap: Test Lure: Pheromone Hormone Farm Name: Liempe Farm Owner: Francis Chulu 5 Ward: Chadiza District: Chadiza Province: Eastern
Data Collection Method	Pheromone trap
Image	(not set)
Humidity	30%
Temperature	26°C
Wind Speed	38km/h
Wind Direction	40°
AI Accuracy	93.935263156891
Latitude	14.24343
Longitude	-15.2656
Extra Data	(not set)
Crop Planting Date	(not set)
Date Created	2019-08-18 20:13:03
Date Updated	2019-08-13 22:13:03
Created by	admin@gmail.com
Updated by	admin@gmail.com

Copyright © 2018-Integrated Pest Management System. All rights reserved. Version 1.0.0

Figure 54: View survey

4.3.1.1.2. Farmer

After the Farmer has logged in by navigating to the login page as shown by Figure 48, they are taken to the dashboard as shown by Figure 59. The dashboard shows only data that is related to the farmer. The dashboard shows the number of farms that the farmer has in the system, the number of traps and crops that are being monitored in the farmer's farms. The dashboard also gives the performance of each trap in the current month and also how the traps have been performing monthly. The dashboard gives the performance of traps on the current day and also gives the performance of the surrounding farms in the ward by giving the data of farms with high numbers.

The farmer can also view the location of other traps in the ward via Google maps as spatial data by navigating to the spatial data page as shown by Figure 60. The farmer can also view surveys coming from traps setup in the farmer's farm by navigating to Survey page as shown by Figure 61.

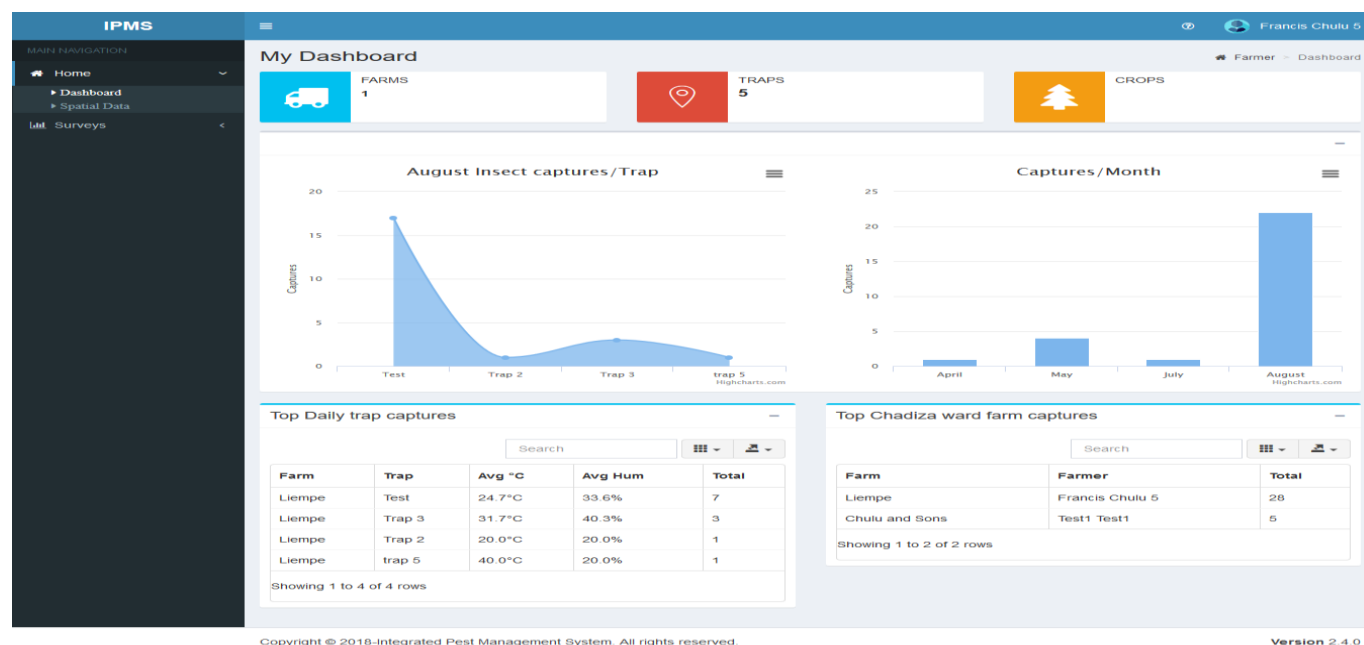


Figure 55: Farmer Dashboard page

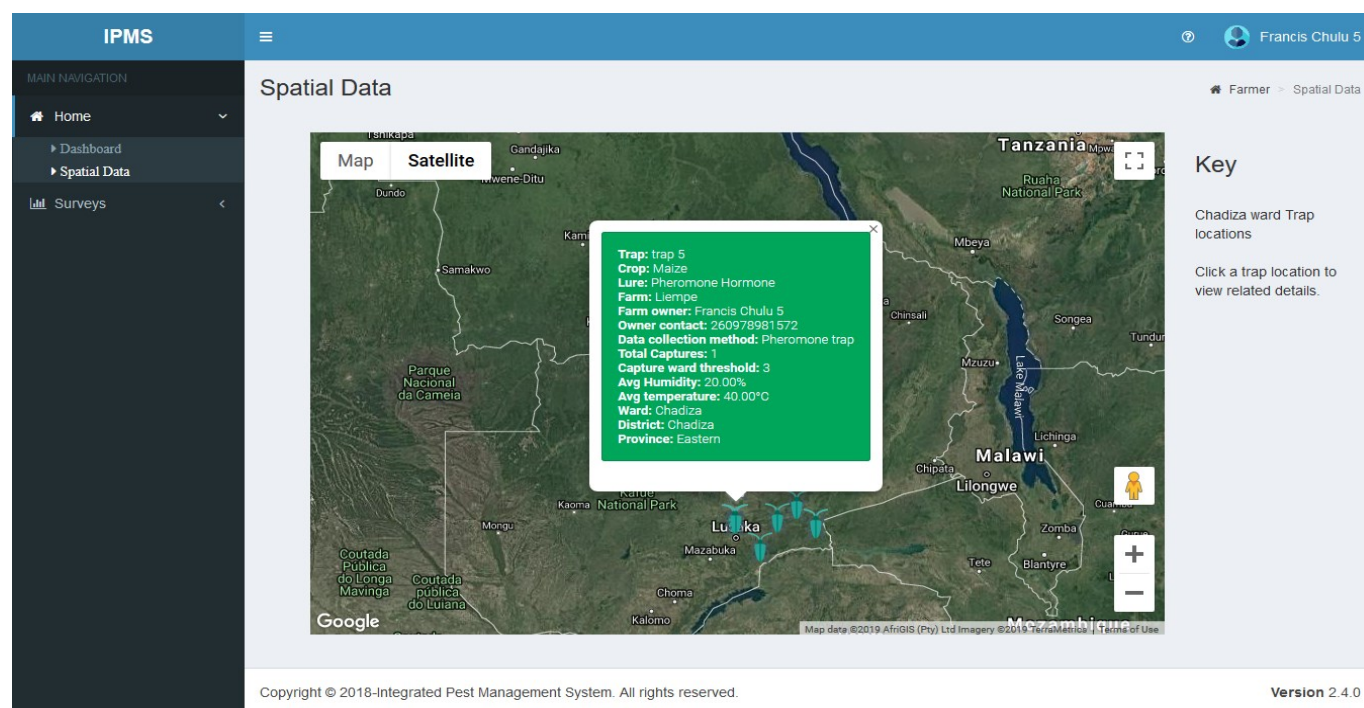


Figure 56: Farmer spatial data page

IPMS

MAIN NAVIGATION

Home

Surveys

Realtime Surveys

My Surveys

Farmer Realtime Surveys

Showing 1-10 of 28 items.

C

Filter

Sort

Edit

Share

#	Crop	Trap	Humidity	Temperaure	Actions	
1	Maize	Test	25%	20°C		☐
2	Maize	Test	60%	23°C		☐
3	Maize	Test	30%	27°C		☐
4	Maize	Test	30%	26°C		☐
5	Maize	Test	30%	26°C		☐
6	Maize	Test	30%	26°C		☐
7	Maize	Test	30%	26°C		☐
8	Maize	Test	30%	26°C		☐
9	Maize	Test	30%	26°C		☐
10	Maize	Test	30%	26°C		☐

«

1

2

3

»

Copyright © 2018-Integrated Pest Management System. All rights reserved.

Version 2.4.0

Figure 57: Farmer Surveys page

2.1.9. 4.3.2. Mobile app

The first version of the mobile application was created for Android devices only with the hope of accommodating IOS users in the future. The app was developed in Java as the preferred language using Android studio. The mobile app is mainly used for trap status monitoring and data visualization by stakeholders. The app provides statistics in form of graph and tables. It also provides the location and status of each and every trap that is being monitored. In addition the app provides real time surveys as they are recorded at the trap sites. The mobile app supports login for FAW experts and farmers. The statistics/data for the experts is for every area that is being monitored while for the farmer it's limited to the ward where their farm is located. The mobile app tries to mimic the web app in some functionalities.

4.3.2.1. Sample mobile app screen shorts

4.3.2.1.1. Login Page

This is the screen that the experts and the farmer uses to login by providing the user name and the password that they use to login into the web app. Figure 62 shows the login page for the mobile app. If a user provides a wrong username/password, the app notifies them with an error message. Upon successful login, the user is directed to the appropriate page depending on the level of access.

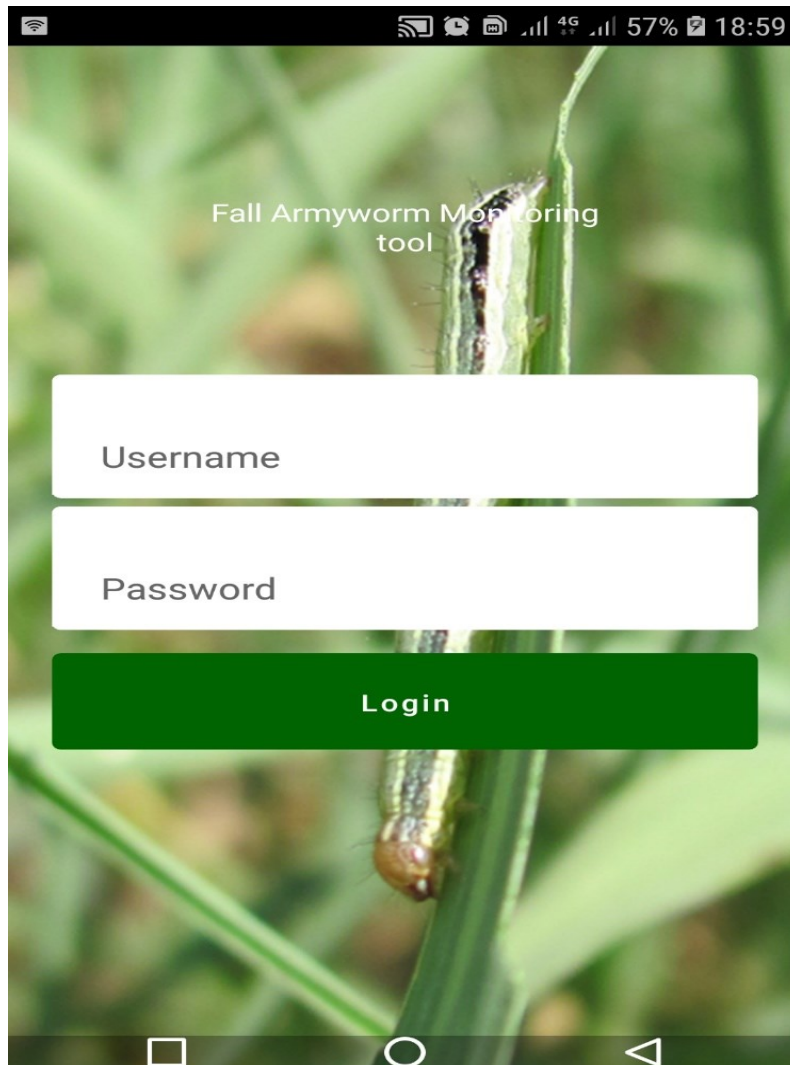


Figure 58: App login page

4.3.2.1.2. Main Menu

This is the main page that both the farmer sees upon a successful login. It gives the user the opportunity to choose which data to view. This version of the app provides a dashboard, spatial data and real time survey options only. Figure 63 shows the mobile app main menu.

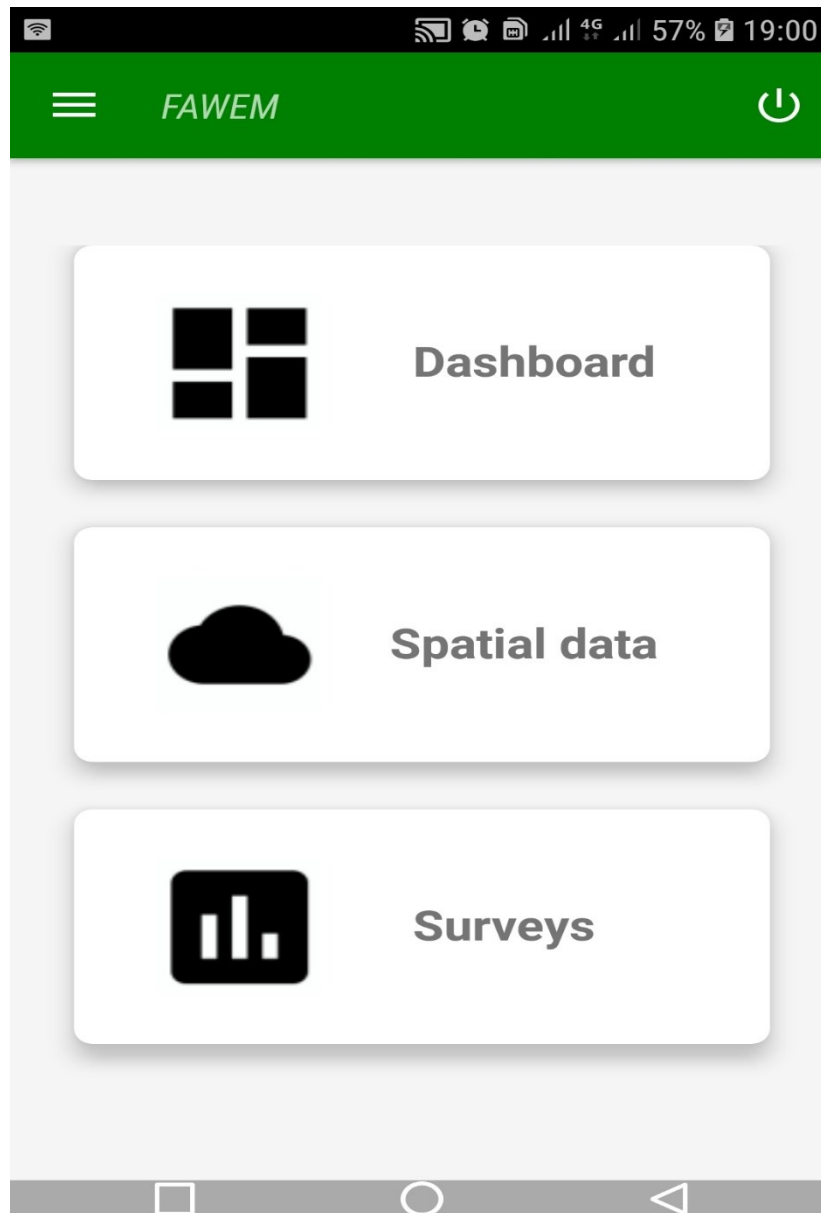


Figure 59: Mobile app Main menu

4.3.2.1.3. Mobile app Dashboard

Upon selecting Dashboard on the main menu, the user is taken to the dashboard. From the dashboard, the user is able to view monthly data statistics in the form of a bar chart as shown by Figure 64. Monthly data refers to how many insects were recorded by the configured traps. Farmers will be able to see only data pertaining to the ward where their farm belongs while FAW experts will be able to see the overall of all the traps.

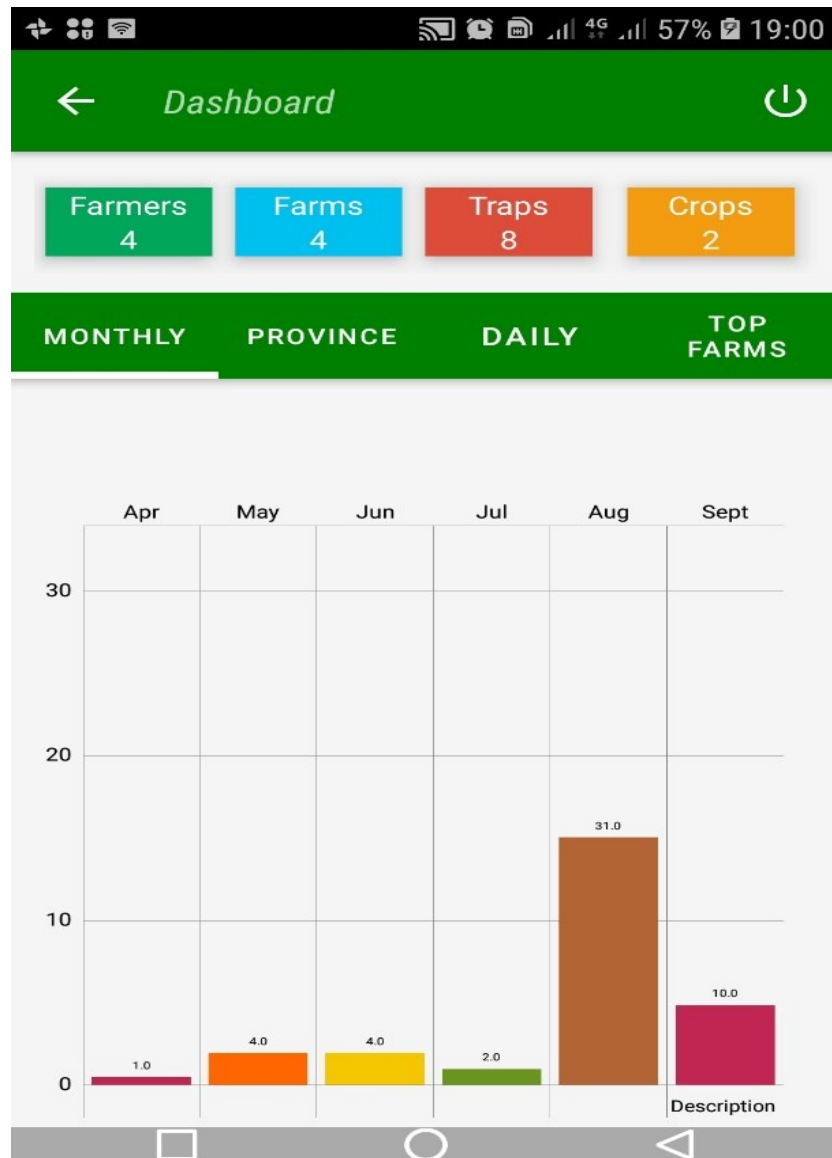


Figure 60: Mobile app dashboard monthly stats

The users will also be able to view current provincial distribution of the FAW in the form of a pie chart as shown by Figure 65. The pie chart gives percentage of how the provinces are currently affected. The pie chart is only available to FAW experts while the farmer will be able to see how the neighboring farms in their ward are affected.

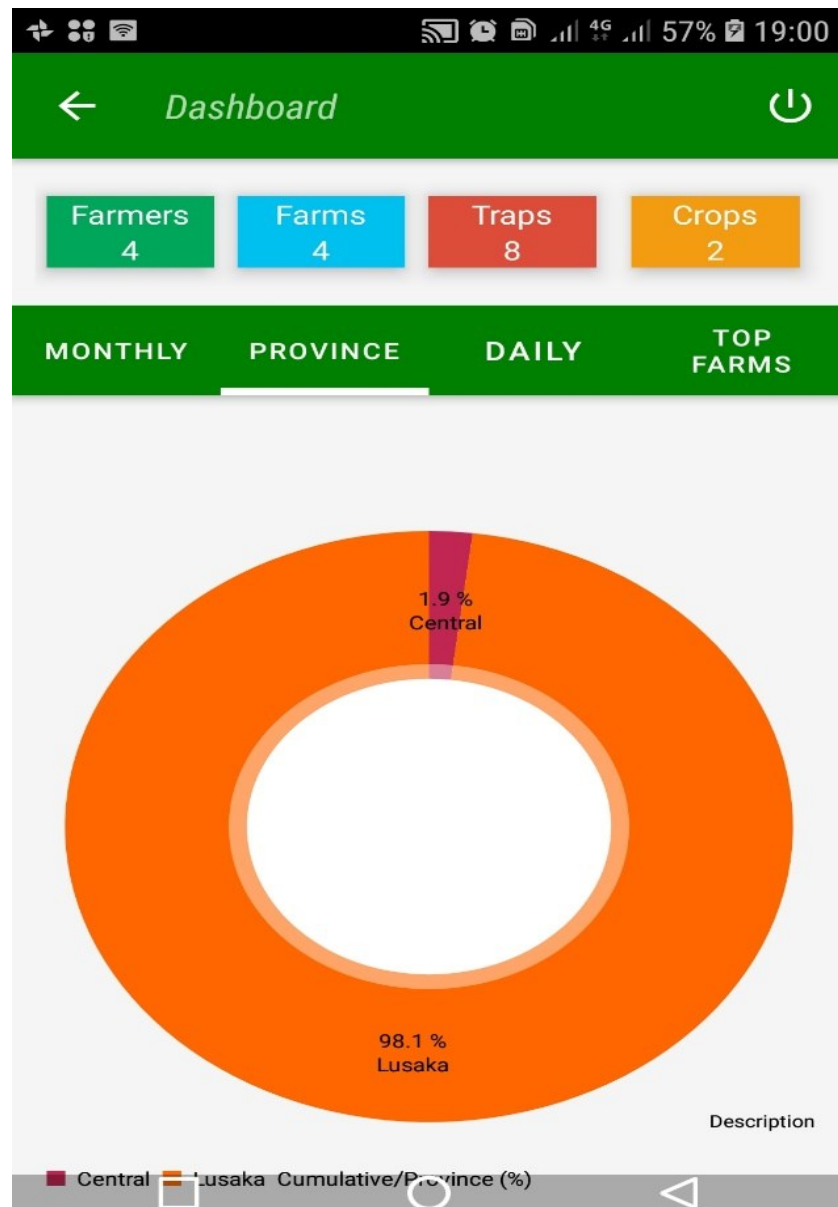


Figure 61: Mobile app dashboard provincial stats

The users will also be able to see the daily top ten trap performance by ranking the traps with the highest trap on top as shown by Figure 66. The page gives statistics from the most affected traps. Farmers will be able to see data from traps located within their ward while FAW experts will see the all the traps in the country.

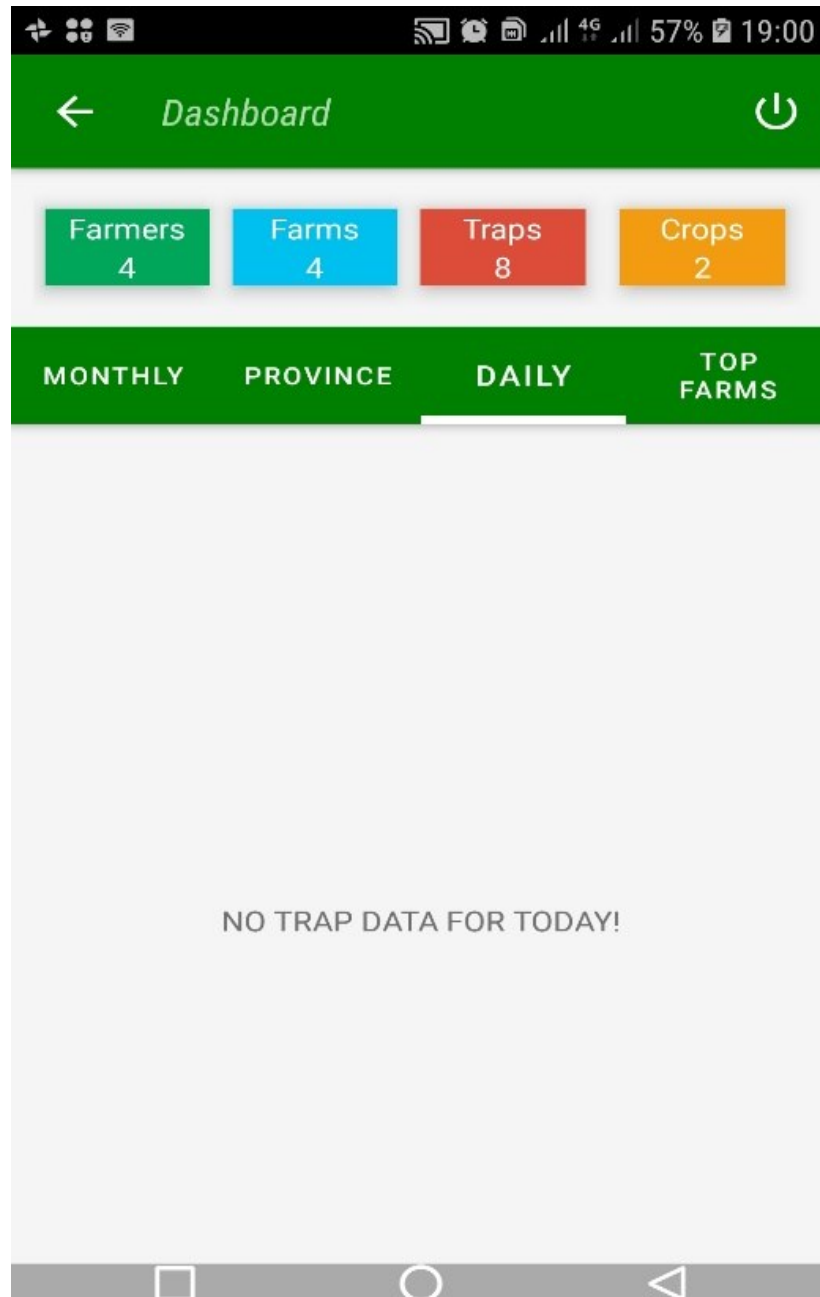


Figure 62: Mobile app dashboard daily stats

The users will also be able to see top twenty highest affected Farms as shown by Figure 67. The farmer will only be able to see farms in their ward while FAW experts will view all the farms registered in the system. In addition the dashboard should be able to show how many farmers and farms are registered in the system, how many traps and crops are being monitored.

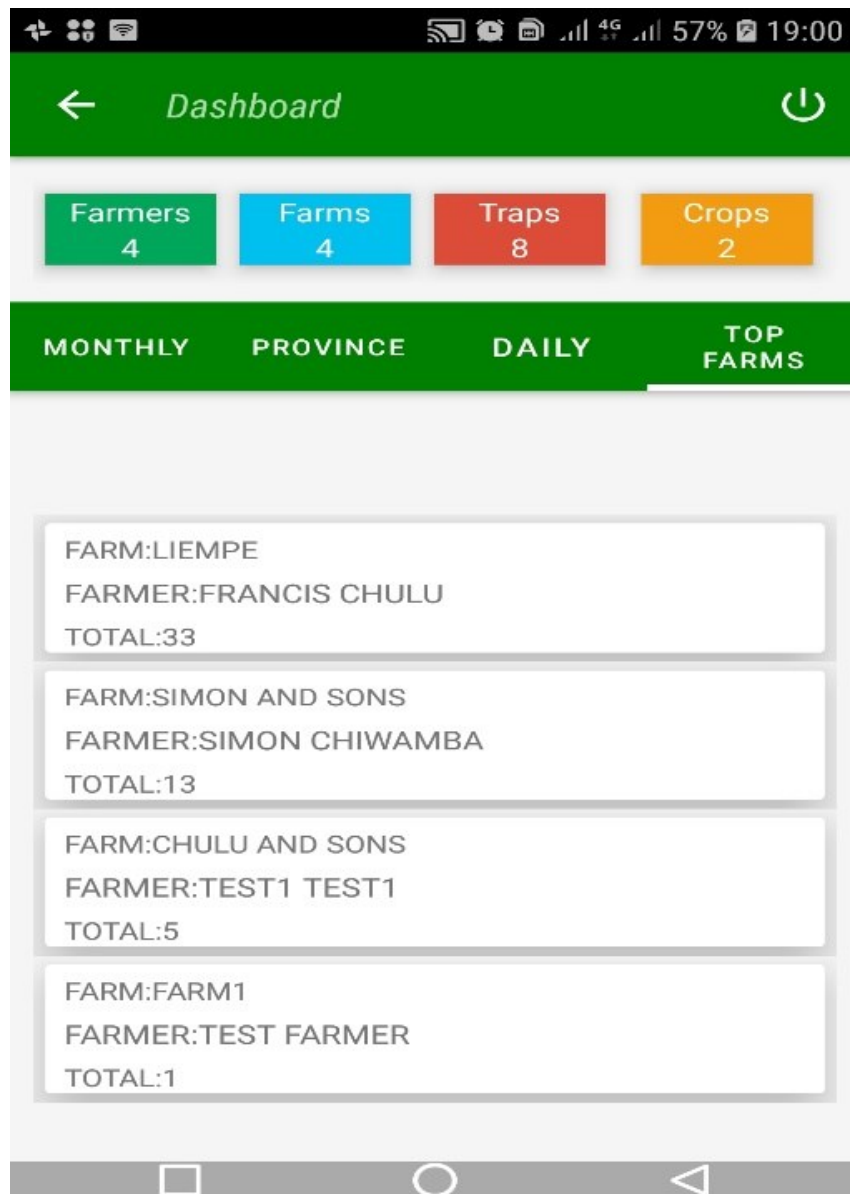


Figure 63: Mobile app dashboard top affected farms

4.3.2.1.4. Mobile app spatial data

Selecting spatial data on the mobile app main menu takes the user to the spatial data page.

Spatial data page gives the location details of the traps and their status. If a trap has captured data and the captures are above the set threshold of the ward where the trap is located, the Map icon will be bouncing giving an indication that the area is affected by fall armyworm as shown by Figure 68.

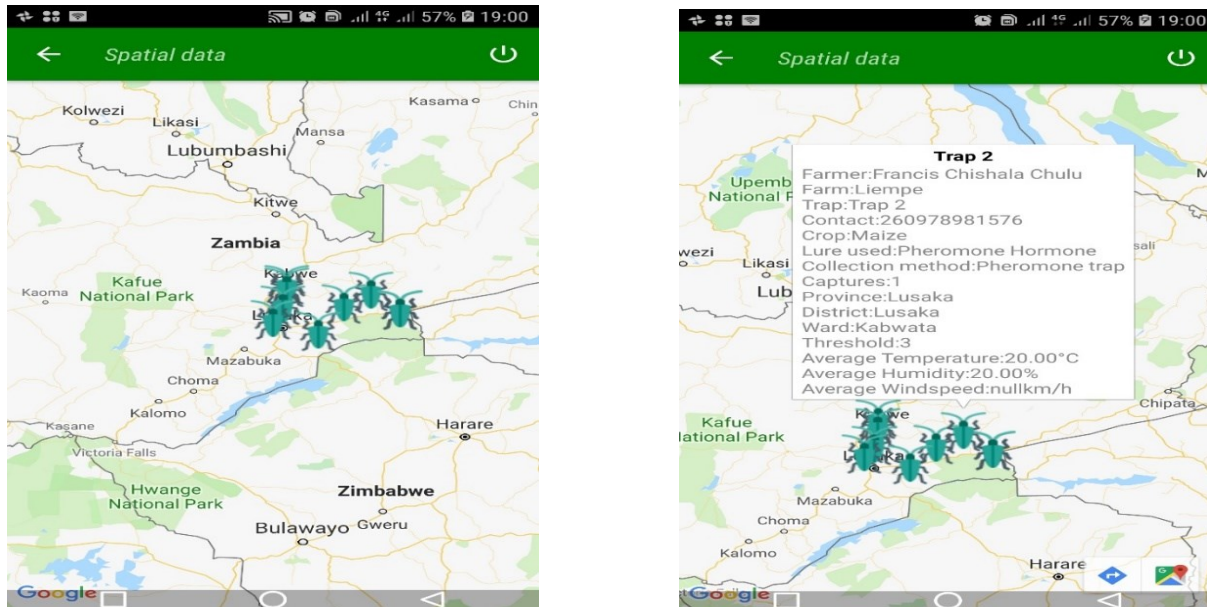


Figure 64: Mobile app spatial data

4.4. Sample Alert messages

2.1.10. 4.4.1. Email

Figure 69 shows a sample email warning message sent to both farmers and experts.

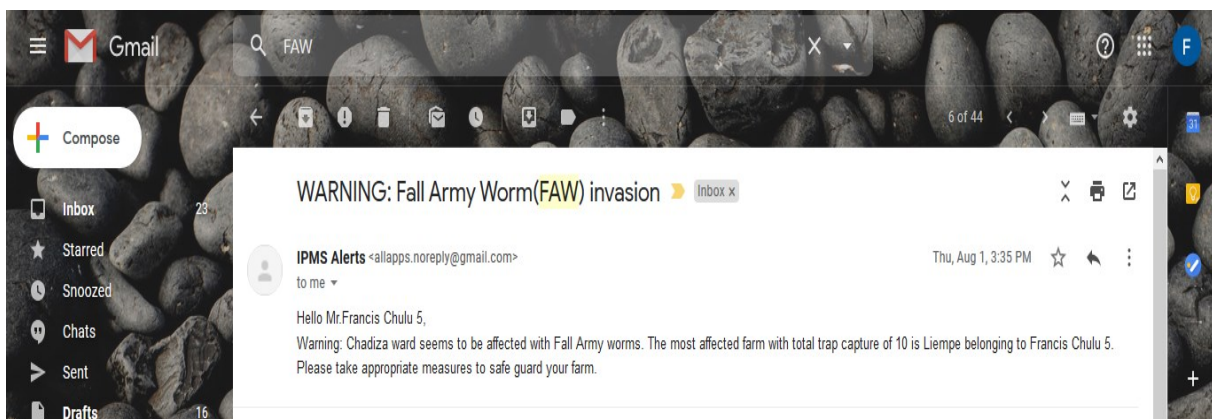


Figure 65: Sample email warning alert message

2.1.11. 4.4.2. SMS Messages

Figure 70 shows the sample sms warning message sent to both farmers and experts.

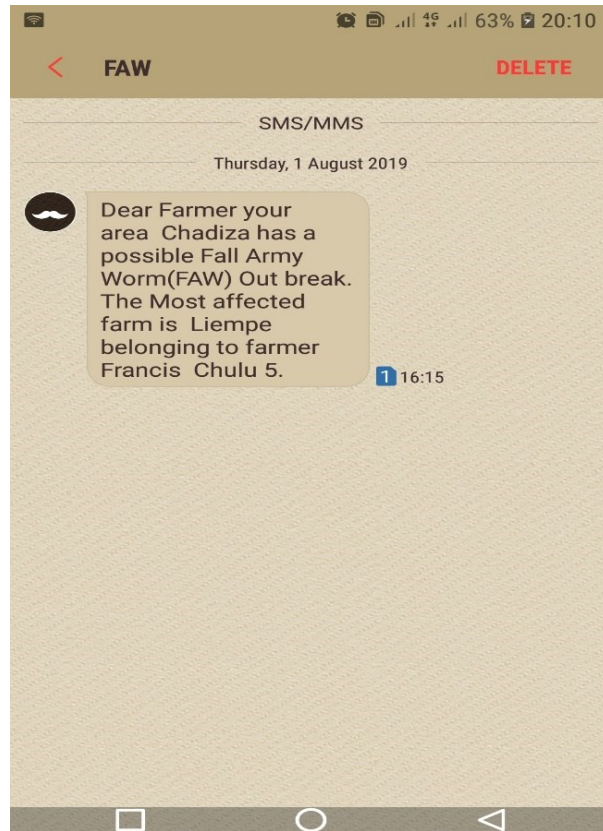


Figure 66: Sample SMS warning alert message

4.5. Summary

This chapter has presented the results of the research study. The results of training the Convolutional neural network (CNN) were presented and the implementation of the generated CNN model was presented. The chapter also covered the implementation of the web application which consisted of a web app and the API. The results of implementing a mobile application were also presented.

CHAPTER FIVE

DISCUSSION AND CONCLUSION

5.1. Introduction

This chapter discusses the findings and provides conclusions based on the study objectives and study questions presented in the first chapter. The main aim of the study was to develop a fall armyworm early warning and monitoring system using a Convolution Neural Network. The study utilized Convolutional Neural networks (CNN) to develop a FAW image prediction model using transfer learning. It also discusses CNN model and the development and implementation of the web and mobile applications.

5.2. Discussion

2.1.12. 5.2. 1. Convolutional Neural Network

A model for the automatic identification and classification of fall armyworm using convolutional neural networks was developed and tested. The major challenges faced in the study included but not limited to; the size of test data which was fairly small and the study had to rely on data augmentation which most likely affected results since to train a good CNN a good amount of data is needed. The lab setup was a bit challenging when dealing with live moth as it was near impossible to place them strategically hence the study relied on the dead ones most of the time. The other challenge was the lack of previous work on fall armyworm using CNN which could have provided the study with an opportunity to build on what others have done. Lack of good computing power meant that we had to use a pretrained model instead of building our own from scratch which would have given us an opportunity to fine tune the network in a way that suits our needs and hence get better results.

As CNNs require a fairly large amounts of data to train a good recognition model and we had little data, we applied transfer learning using the inceptionV3 pre-trained model. The prediction accuracies were largely affected by little training data. After several trials the results of round one gave a better prediction accuracy (82%) which could greatly be attributed to the higher learning rate. Many improvements can be done to try and improve the prediction accuracies such as using fine tuning instead of feature extraction transfer learning and unfreeze a few feature extraction layers of the inception model and train them together with our new

classification layer. We could also try and increase the training data set by collecting more images from the field and the internet and applying augmentation even though augmentation does not guarantee that the model will learn any new features. The other improvements we could use is the removal of noise from the current dataset such as removing the entire background from images. Another factor to consider would be the introduction of other insect species because in the presented results our focus was on FAW only hence we just introduced a false positive class and added insects which are likely to be attracted to the FAW pheromone hormone.

The model that gave a better prediction accuracy which was that of round one (average 82%) was used to implement the FAW prediction app using Tensorflow serving as a flask server. The model was deployed on a cloud test environment.

2.1.13. 5.2.2. Web and mobile application development

The initial prototypes of the web application and mobile application were developed and deployed on a cloud server as a development environment. The web application is made of a web interface and an API. The web interface has functionalities such as farmer registration, farm registration, trap registration, crop and data collection method entering, user registration based on access levels. The web interface also provides a dashboard for stakeholders to visualize monitored data using various filters such as monthly statistics, provincial statistics and so on. The web app also offers users to monitor the status of each and every trap that is registered and is being monitored in the system. It also gives various filters for environment variables temperature, humidity and wind speed. The user can filter and check the average environmental conditions for a province district, ward or trap for the last one hour, last 24 hours, last seven days, last 31 days or last one year in the form of graphs and/or table.

The web app allows both faw experts and farmers to login by providing them with different login interfaces. The farmer logs in using their NRC as username and NRC as password if they do not change it. The experts use email as username and the set password set when they received the password set on the email. The development of web app was done using a rapid prototyping method that enables incoming system changes to be included immediately. The web app also had functions that will run as Cron jobs to push warning alerts to farmers and FAW experts in the form of SMS and email. The development tools used includes Yii2

Framework which is a Framework done in PHP, HTML, JavaScript and CSS which enables rapid prototyping to be possible. The database engine MySQL was used as the backend. The web app was deployed and test on a cloud server.

The API was also developed using the web development tools used to develop the web app. The API has methods and functions that enables the mobile app to interact with the database. The API is the main application that interacts with the CNN model and inserts the prediction accuracy and the received data in the database upon successful predictions.

The mobile app was developed using android studio and Java as the main development language. The initial version of the mobile application is mainly used for data visualization. It is able to show how the monitored areas are affected by the FAW. It is also able to show the location and details of the traps being monitored and other functionalities.

5.3. Conclusion

This study aimed at developing an early warning and monitoring tool for the fall armyworm pest. From the reviewed literature, it can be concluded that there is little to no research that has been done in the monitoring of the fall armyworm using Artificial intelligence. The study also shows that there is almost no proper FAW monitoring and early warning process in Zambia apart from the manual field monitoring. The main objectives of the study were to train a Convolutional Neural Network model and use the model to develop a web and mobile application to use as monitoring tools. The study investigated how a CNN model can be used to identify and classify fall armyworm moth.

The results of training the model using transfer learning showed that a model can be trained that can be used to predict the fall armyworm moth. The trained model (with average prediction accuracy of 82%) was implemented using Tensorflow serving. Using the implemented model, a prototype web app was developed to act as a client to the model by pushing received faw images to the model and give back prediction response. The developed web app is also used for database access and as a data visualization tool. A prototype mobile application has also been developed to be used as a data visualization tool for all interested stakeholders. The training of the CNN model and its successful implementation and the development of the web and mobile applications has enabled this study to meet the set objectives.

It is hoped that the developed tools once tried and tested in a real environment will contribute significantly to the monitoring of the fall armyworm in Zambia and in turn contribute to improving food security.

5.4. Recommendations

In view of the monitoring tools developed, the study provides the following recommendations.

1. The developed model/system requires rigorous testing under field conditions.
2. Upon a successful testing of the system under 1 above, the farmer needs to be properly trained and sensitized on how they can properly utilize the developed tools in the fight against the fall pest.
3. The developed model and tools can be extended to the monitoring of other pests that affect crops in Zambia if it is rigorously tested.
4. The model's prediction accuracy can be improved by increasing the training dataset through collecting more FAW images and/or through data segmentation.
5. With enough server hard disk space, the images predicted with an accepted prediction accuracy can be kept to build a fall armyworm dataset that other interested researchers can utilize to expand the research on the fall armyworm.
6. The Use of Unstructured Supplementary Service Data (USSD) to accommodate farmers that have no access to an android device and a computer.
7. Increasing and cleaning the CNN training dataset by continuing to collect more from the field and the internet.
8. Trying the feature extraction of transfer learning and use the extracted features in a support vector machine and see how the results compare with the results of this study, apart from the InceptionV3, trying out other pre-trained models such as mobileNet and compare the results with that of InceptionV3 results. These steps may most likely improve the prediction accuracy of the CNN model.
9. Replicating the Android mobile into IOS to accommodate iPhone users and include the farmer, farm, and trap registration functionality on the mobile app

REFERENCES

1. “Four threats to global food security and what we can do about them,” *The conversation*, Available online [Accessed: 06-August-2019]
2. R. Day, Roger, P. Abrahams, M. Bateman, T. Beale, Tim, V. Clottey, M. Cock, M.J.W. Y. Colmenarez, N. Corniani, R. Early, J. Godwin, J. Gomez, P.G. Moreno, S.T. Murphy, B.O. Mensah, N. Phiri, C. Pratt, S. Silvestri, and A. Witt, “Fall Armyworm: Impacts and Implications for Africa,” *Outlooks on Pest Management*, Vol. 28, pp. 196-201, October 2017.
3. J.E. Huesing, B.M. Prasanna, and D. McGrath, P. Chinwada, P. Jepson, J.L. Capinera, “Fall Armyworm in Africa: A Guide for Integrated Pest Management,” 1st ed. 2018.
4. P. Abrahams; M. Bateman; T. Beale; V. Clottey; M. Cock; Y. Colmenarez; N. Corniani; R. Day, R. Early, J. Godwin, J. Gomez, P. G. Moreno; S.T. Murphy; B.O. Mensah, N. Phiri, C. Pratt; G. Richards, S. Silvestri, and A. Witt, “Fall Armyworm: Impacts and Implications for Africa, Evidence Note (2),” September 2018.
5. L. Feng, B. Bhanu, and J. Heraty, “A software system for automated identification and retrieval of moth,” *Pattern Recognition*, Vol. 51, pp. 225-241, September 2016.
6. L. Granado, O. Lopez, M. M. Rach, H. Migallón, P. Manuel, B. Pina, A. Bonastre, and S. Martín, and J.J. Serrano, “Monitoring Pest Insect Traps by Means of Low-Power Image Sensor Technologies”. *Sensors* (Basel, Switzerland), December 2012.
7. R. Chikowo,” Description of cropping systems, climate, and soils in Zambia”, Global Yield Gap Atlas. Available online: <http://www.yieldgap.org/zambia>. [Accessed: 06-August-2019]
8. M. Bwalya, and B. Deka, “Analysis of the Second National Agriculture policy 2016-2020”, Policy Monitoring and Research center. National-Agricultural-Policy-2016-2020.pdf. [Accessed: 06-August-2019]
9. “The National Agriculture Policy 2012-2030”, Ministry of Agriculture and Cooperatives, August 2011.
10. E. Chernoh, “Larger grain borer”, Africa Health Soil Consortium, November 2014. Available online: <http://africasoilhealth.cabi.org/wpcms/wp-content/uploads/2015/02/46-cereals-larger-grain-borer.pdf> [Accessed: 06-August-2019]
11. P.H. Sohat, E.M. Musonda, and M. Mukanga, “Distribution of cereal Stem borers in Zambia and release of *Cotesia Flavipes* Cameron, an exotic natural enemy of *Chilo*

- partellus* (SWINHOE)", Insect Science Application. Vol. 21, No. 4, pp. 311 – 316, January 2002.
12. "Locusts, Army Worms Threaten Food Security in East and Southern Africa", the *Zambian Analyst*. Available online: <http://paulshalala.blogspot.com/2017/02/locusts-army-worms-threaten-food.html> [Accessed: 06-August-2019]
 13. "Integrated Management of the fall armyworm on Maize, A guide for farmer field schools in Africa", Food and Agriculture organization of the United Nations, Rome 2018.
 14. J.L. Capinera, "Fall Armyworm, *Spodoptera frugiperda* (J.E. Smith) (Insecta: Lepidoptera: Noctuidae)", UF-IFAS extension, University of Florida.
 15. E.R. Mitchell, "Monitoring Adult Populations of the fall armyworm", *The Florida Entomologist*, Vol. 62, No. 2, pp. 91-98, June 1979.
 16. D. N Epsky, W.L. Morrill, and R.W. Mankin, "Traps for Capturing Insects" in *Encyclopedia of Entomology*, 2008, Ch. 10, pp. 3887-3901. Available online: https://www.researchgate.net/publication/303661972_Traps_for_Capturing_Insects
 17. C.T. Backer, "Use of Pheromone in IPM," in *Integrated Pest Management: Concepts, Tactics, Strategies and Case Studies*, Cambridge: Cambridge University Press, 2018, Ch. 21, pp. 273-285.
 18. A.M. Shelton, and F.R. B. Perez, "Concepts and applications of trap cropping in pest management", *Annual Review of Entomology*, Vol. 51, pp. 285–308. February 2006.
 19. S.N. Ahmad, and H.K. Norman, "Pheromone Trapping in Controlling Key Insect Pests: Progress and Prospects," *Oil Palm Bulletin*, 2011.
 20. S. Guerrero, J. Brambila, and R.L. Meagher, "Efficacies of Four Pheromone-Baited Traps in Capturing Male *Helicoverpa* (Lepidoptera: Noctuidae) Moths in Northern Florida," *Florida Entomologist*, Vol. 97(4), pp. 1671-1678, December 2014.
 21. S. N. A. Hassan, N. N. S. A. Rahman, Z. Z. Htike, and S. L. Win, "Vision Based Entomology: A Survey," *International Journal of Computer Science & Engineering Survey*, Vol. 5, pp. 19-32, January 2014.
 22. L. Feng, B. Bhanu, and J. Heraty, "A software system for automated identification and retrieval of moth," *Pattern Recognition*, Vol. 51, pp. 225-241, September 2016.
 23. P. Boniecki, K. Koszela, H. P. Boniecka, J. Weres, M. Zaborowicz, S. Kujawa, A. Majewski, and B. Raba, "Neural identification of selected apple pests," *Computers and Electronics in Agriculture Systems*, Vol. 110, pp. 9-16, September 2015.

24. A. K. Banerjee, K. Kiran, U. Murty, and C. Venkateswarlu, "Classification and Identification of mosquito species using artificial neural networks," *Computational Biology and Chemistry*, Vol. 32, pp. 442–447, June 2008.
25. J. Wang, L. Ji, A. Liang, and D. Yuan, "The identification of butterfly families using content-based image retrieval," *Biosystems Engineering*, Vol. 111, pp. 24-32, 2012.
26. M. Do, J. Harp, and K. Norris, "A test of a pattern recognition system for identification of spiders," *Bulletin of Entomological Research*, Vol. 89, pp. 217-224, 1999.
27. H. Gassoumi, N.R. Prasad, and J.J. Ellington, "Neural Network-Based Approach for Insect Classification in Cotton Ecosystems, January 2000.
28. J. Wang, C. Lin, L. Ji, and A. Liang, "A new automatic identification system of insect images at the order level," *Knowledge-Based Systems*, Vol. 33, pp. 102-110, March 2012.
29. "FAMEWS Mobile app". [Online]. Available: <http://www.fao.org/fall-armyworm/monitoring-tools/famews-mobile-app/en/> [Accessed: August. 06, 2019].
30. "Mobile app for pest and disease management of crops", ICRISAT Happenings Newsletter. [Online]. Available: <https://www.icrisat.org/mobile-app-for-pest-and-disease-management-of-crops/> [Accessed: May.21, 2019].
31. "Maano-Virtual Farmers Market", World Food Programme-Innovation Accelerator. [Online]. Available: <https://innovation.wfp.org/project/virtual-farmers-market> [Accessed: August.06, 2019].
32. "Lima Links" [Online]. Available: <http://www.limalinkszambia.com/> [Accessed: August.06, 2019].
33. K.G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine Learning in Agriculture: A Review", *Sensors*, vol. 18(8), pp. 2674, August 2018.
34. A. Kamilaris, and F.X. Prenafeta-Boldu, "Deep Learning in Agriculture: A Survey", *Computers and Electronics in Agriculture*. Vol. 147, pp. 70-90, 2018.
35. A. Kamilaris, and F.X. Prenafeta-Boldu, "A review of the use of convolutional neural networks in agriculture," *The Journal of Agricultural Science*, Vol. 156, Issue 3, pp. 312-322 , April 2018.
36. A. Simon, M.S Deo, S. Venkatesan, and D.R. Ramesh Babu, "An Overview of Machine Learning and its Applications", *International Journal of Electrical Sciences & Engineering (IJESE)*. Vol. 1, No. 1, pp. 22– 24, 2015.
37. Pooja D. Pancholil, and Sonal J. Patel, "A brief Introduction of Machine Learning with different Tasks and Applications", *International Journal of Research and Analytical Reviews (IJSAR)* .Vol. 6, No. 1, 20 Feb 2019.

38. Simeone .O, "A very brief introduction to machine learning with applications to communication systems". IEEE Transactions on Cognitive Communications and Networking, vol. 4, No. 4, 648-664, 2018.
39. D. Sharma, and N. Kumar, "A Review on Machine Learning Algorithms, Tasks and Applications". International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 6(10), 2017.
40. S. S. Shwartz, and S. B. David, "Understanding machine learning: From theory to algorithms". Cambridge university press, 2014.
41. M. Negnevistky, "Artificial Neural Networks" in Artificial Intelligence a guide to intelligent systems, 2nd ed. Harlow, England: Pearson Education limited, 2005, Ch. 6, pp. 165–216.
42. S.N. Sivanandam, and S.N.Deepa, "Artificial Neural Networks: An introduction" in Principles of soft computing, 1st Ed. New Delhi, India: Wiley, 2007 Ch. 2, pp. 11–47.
43. B. Yegnanarayana, "Artificial Neural Networks," in Artificial Neural Networks, 1st Ed. New Delhi, India: Prentice-Hall of India private limited, 2005, Ch. 2, pp. 15–39.
44. H.Kukreja1, N.Bharath, C.S. Siddesh, and S. Kuldeep, "An Introduction to Artificial Neural Network", IJARIE. Vol. 1, No.5, 2016.
45. J. Larsen, "Definition of Neural Networks" in Introduction to Artificial Neural Networks, 1st ed. Technical University of Denmark: Department of Mathematical Modelling, November 1999, pp. 2-5
46. D. Kriesel, "Components of artificial neural networks" in A Brief introduction to Neural Networks, 1st ed. 2005, Ch. 3, pp. 33-35
47. Kevin Gurney, "Neural Networks- An Overview" in An introduction to neural networks, 2nd ed. London, England: UCL Press, 2004, Ch. 1, pp. 11-19
48. Zhang Zhongheng, "A gentle introduction to artificial neural networks. Annals of translational medicine. vol. 4, No. 19, pp. 370, 2016.
49. S. Dahikar, and S.V. Rode, "Agricultural Crop Yield Prediction Using Artificial Neural Network Approach", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering. Vol. 2, No. 1, January 2014
50. S.K Bejo, S Mustaffha, and W.I.W. Ismail, "Application of Artificial Neural Network in Predicting Crop Yield: A Review", Journal of Food Science and Engineering. Vol. 4, pp. 1-9, 2014
51. C. Jeeva, and S.A Shoba, "An Efficient Modelling Agricultural Production Using Artificial Neural Network (ANN)", International Research Journal of Engineering and Technology (IRJET). Vol. 03, No. 05, May 2016.

52. R. K. Samanta, and I. Ghosh, "Tea Insect Pests Classification Based on Artificial Neural Networks", *International Journal of Computer Engineering Science (IJCES)*. Vol. 2, No. 6, June 2012.
53. L. Shi, Q. Duan, X. Ma, and M. Weng, "The Research of Support Vector Machine in Agricultural Data Classification", *Computer and Computing Technologies in Agriculture (CCTA)*. Vol. 5, pp.265-269, Oct 2011.
54. S. Akbarzadeh, A. Paap, S. Ahderom, B. Apopei, and K. Alameh, "Plant discrimination by Support Vector Machine classifier based on spectral reflectance", *Computers and Electronics in Agriculture*. vol. 148, pp. 250-258, 2018.
55. S.Y. xue, X. Huan, and Y.L. Jiao, "Support vector machine-based open crop model (SBOCM) - Case of rice production in China," *Saudi Journal of Biological Sciences*. Vol. 24, pp. 537–547, 2017.
56. M.A. Ebrahimi, M.H. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-based pest detection based on SVM classification method," *Computers and Electronics in Agriculture*. Vol. 137, pp. 52-58, 2017.
57. M. Manoja, and J. Rajalakshmi, "Early Detection of Pests on Leaves Using Support Vector Machine," *International Journal of Electrical and Electronics Research*. Vol. 2, No. 4, pp. 187-194, December 2014.
58. P.K. Sethy, C. Bhoi, N.K. Barpanda, S. Panda, S.K. Behera, and A.K. Rath, "Pest Detection and Recognition in Rice Crop Using SVM in Approach of Bag-Of-Words," *International Conference on Communication and Signal Processing*. Vol., pp., April 2017.
59. M.S. Colgan, C.A. Baldeck, J.B. Féret, and G.P. Asner, "Mapping Savanna Tree Species at Ecosystem Scales Using Support Vector Machine Classification and BRDF Correction on Airborne Hyperspectral and LiDAR Data," *Remote Sensing*. Vol. 4, pp. 3462-3480, 2012.
60. A. Kamilaris, and F.X. Prenafeta-Boldu, "Deep Learning in Agriculture: A Survey", *Computers and Electronics in Agriculture*. Vol. 147, pp. 70-90, 2018.
61. A. Kamilaris, and F.X. Prenafeta-Boldu, "A review of the use of convolutional neural networks in agriculture," *The Journal of Agricultural Science*, Vol. 156(3), pp. 312-322, April 2018.
62. Y. LeCun, and Y. Bengio, "Convolutional networks for images, speech, and time series", In Arbib MA (ed.), *The Handbook of Brain Theory and Neural Networks*, Cambridge, MA, USA, MIT Press, pp. 255–258, 1995.
63. Liu, T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of Training Convolutional Neural Networks", *Computer Vision and Pattern Recognition*, Vol. 2, June 2015.

64. W. Rawat, and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", *Neural Computation, the MIT Press Journal*, Vol. 29(9), pp. 2352-2449, September 2017.
65. M. Valan, K. Makonyi, A. Maki, D. Vondráček, F. Ronquist, "Automated Taxonomic Identification of Insects with Expert-Level Accuracy Using Effective Feature Transfer from Convolutional Networks", *Systematic Biology*, March 2019.
66. B. Gao, and L. Pavel, "On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning", 2017.
67. V.B. Sai, G. Rao, R.G. Narasimha, M. Ramya, S.Y. Sujana, and T. Anuradha, "Classification of skin cancer images using Tensorflow and inception v3", *International Journal of Engineering & Technology*, Vol. 7, pp. 717-721, March 2018.
68. S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.
69. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Computer Vision and Pattern Recognition*, Vol. 3, 2016.
70. Google Cloud Platform, "Advanced Guide to Inception v3 on Cloud TPU", Available online: <https://cloud.google.com/tpu/docs/inception-v3-advanced>. Accessed June 20, 2019.
71. P. Goldsborough, "A Tour of Tensorflow", *CoRR-Machine Learning*, Vol. abs/1610.01178, October 2016.
72. M. Wul , C. Yan, H. Liu, and Q. Liu, "Automatic classification of ovarian cancer types from cytological images using deep convolutional neural networks", *Biosci. Rep.* 38, BSR20180289doi:10.1042/BSR20180289, 2018.
73. S. Talpur, and N. Khoso, "Advanced Ambulatory Operating Stretcher Learned by Means of Convolutional Neural Network (CNN)", *Journal of Biomedical Engineering and Medical imaging*, Vol. 5(3), 2018.
74. D. Motta, A.B. Santos, I. Winkler, B. Machado, and D. Pereira, "Application of convolutional neural networks for classification of adult mosquitoes in the field", *PLOS ONE*, Vol.14 (1), 2019.
75. A. Nieuwenhuizen, J. Hemming, and H. Suh, "Detection and classification of insects on stick traps in a tomato crop using Faster RCNN", <http://edepot.wur.nl/463457>, Researchgate, Jan. 2018.
76. E.A. Hay, and R. Parthasarathy, "Performance of convolutional neural networks for identification of bacteria in 3D microscopy datasets", *PLOS Computational Biology*, Vol. 14(12), December 2018.

77. I. Sommerville, "Software engineering", 9th ed. Boston, USA: Pearson, 2011.
78. S. Barjtya1, A. Sharma, and U. Rani, "A detailed study of Software Development Life Cycle (SDLC) Models," *International Journal of Engineering and Computer Science*. Vol. 6, No. 7, pp.22097-22100, July 2017.
79. M. Helingo, B. Purwandari, R. Satria, and I. Solichah, "The Use of Analytic Hierarchy Process for Software Development Method Selection: A Perspective of e-Government in Indonesia," in *Procedia Computer Science* Proceedings of the 4th Information Systems International Conference. Vol. 124, pp. 405–414, 2017.
80. T. J. Lehman, and A. Sharma, "Software Development as a Service: Agile Experiences," 2011 Annual SRII Global Conference, San Jose, CA, 2011, pp. 749-758
81. P. Clarke, and R. V. O'Connor, "The situational factors that affect the software development process: towards a comprehensive reference framework," *Journal of Information and Software Technology*, Vol. 54, Issue 5, Elsevier, pp. 433-447, 2012.
82. "Iterative and Incremental Development", Wikipedia. [online] Available: http://en.wikipedia.org/wiki/Iterative_and_incremental_development. [Accessed: 8-August 2019]
83. Mohamed Sami, "Software Development Life Cycle Models and Methodologies", *Software Engineering & Architecture Practices*. Available online: <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/>. [Accessed: 8-August 2019].
84. M.O. Gewaltig, and R. Cannon, "Current Practice in Software Development for Computational Neuroscience and How to Improve It," *PLOS Computational Biology*. Vol. 10, No. 1, pp. 1-9, January 2014
85. D. Kuhn, "Selecting and effectively using a computer aided software engineering tool," in *Annual Westinghouse computer symposium*, Pittsburgh, USA, 1989.
86. B. Kernighan, and P. Plauger, "Software tools in Pascal", 1st ed. Boston, USA: Addison-Wesley, 1981.
87. R. Morales, V. Yarel, and A. Hern, "An analysis of tools for automatic software development and automatic code generation," *Revista Facultad de Ingeniería, Universidad de Antioquia*. No. 77, pp. 75-87, 2015.
88. K. Tatroe, R. Lerdorf, and P. MacIntyre, "Programming PHP". Sebastopol, CA: O'Reilly Media, 3rd ed., 2013.
89. B. Keck, "Yii 2 for Beginners". British Columbia: Leanpub, 2014.
90. J. Duckett, "HTML & CSS: design and build websites". Indianapolis, Indiana: John Wiley & Sons Inc., 2014.

91. S. A. Musungu, "fall armyworm prediction model on the maize crop in Kenya: an internet of things based approach", Strathmore University, 2018. Available online: <https://su-plus.strathmore.edu/handle/11071/5995>
92. C.A.O. Midegaa, J.O. Pittchara, J.A. Pickettb, G.W. Hailua, and Z.R. Khana, "A climate-adapted push-pull system effectively controls fall armyworm, *Spodoptera frugiperda* (J E Smith), in maize in East Africa," Crop Protection, Elsevier. Vol. 105, pp. 10-15, 2018.
93. Y. He, H. Zeng, Y. Fan, S. Ji, and J. Wu1,"Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests," Mobile Information Systems, Hindawi. Vol. 2019, Article ID 4570808, 14 pages, 2019.
94. N. Srivastava, G. Chopra, P. Jain, and B. Khatter, "Pest monitor and control system using wireless sensor Network (with special reference to acoustic device wireless sensor)" in Proceedings of International Conference on Electrical and Electronics Engineering, Goa, India. 27th Jan 2013.
95. M. Hirafuji, H. Yoichi1, T. Watanabe1, M. Asai1, H. Haoming1, K. Tanaka1, T. Fukatsu1, T. Kiura1, and S. Ninomiya, "Real-time Insect Monitoring System by Using Field Server," in Proceedings of World conference on agricultural information and IT, IAALD AFITA WCCA 2008, Tokyo University of Agriculture, Tokyo, Japan. pp 277–282, 24–27 August 2008
96. G.A. Holguin, B. L. Lehman, L.A. Hull, T. Clement, V.P. Jones, and J. Park, "Electronic Traps for Automated Monitoring of Insect Populations," IFAC Proceedings Volumes. Vol. 43, No. 26, pp. 49-54, 2010.
97. R. France, A. Evans, K. Lano, and B. Rumpe, "UML as a formal modelling notation," Computer Science, Software engineering. Vol. 19, no.7, pp. 325-334, Nov 1998
98. D. Harel, and S. Maoz, "Asset and negate revisited: Modal semantics for UML sequence diagrams," Software and System modeling manuscript. Vol. 7, no. 2, pp. 237-252, May 2008.
99. M. Manoj, and J. Rajalakshmi, "Early identification and extraction of pests on leaves using support vector machine," International Journal of Electrical and Electronics Research. Vol. 2, No. 4, pp. 187-194, December 2014.
100. Y. Shi, Z. Wang, X. Wang, and S. Zhang, "Internet of Things Application to Monitoring Plant Disease and Insect Pests," International Conference on Applied Science and Engineering Innovation. Vol. 1, No. 4, pp. 31-34, 2015.

LIST OF PUBLICATIONS

- [1].F. Chulu, J. Phiri, P.O.Y. Nkunika, M. Nyirenda, M. M.Kabemba, P. H. Sohati, “*A Convolutional Neural Network for Automatic Identification and Classification of Fall Armyworm Moth,*” International Journal of Advanced Computer Science and Applications (IJACSA). Vol.10, No.7, 2019. (SCOPUS)
- [2]. F. Chulu, J. Phiri, M. Nyirenda, M. M.Kabemba, P.O.Y. Nkunika, S. Chiwamba, “*Developing an automatic identification and early warning and monitoring web based system of fall armyworm based on machine learning in developing countries,*” Zambia ICT Journal. Vol. 3, No. 1, pp. 13-20, March 2019. Available at: <http://ictjournal.icict.org.zm/index.php/zictjournal/article/view/71>
- [3].F. Chulu, J. Phiri, P.O.Y. Nkunika, M. Nyirenda, M. M. Kabemba, M. Moonga, “*Developing an automated fall armyworm (FAW) identification and early warning and monitoring system based on ANN techniques,*” in Proceedings of the ICTSZ International Conference In ICTs (ICICT2018) - Lusaka, Zambia (12TH - 13TH December 2018).

APPENDICES

Appendix 1: Flask Server code

```
#!/usr/bin/env python3
# app.py
#
# A simple example of hosting a TensorFlow model as a Flask service
#
# Copyright 2017 ActiveState Software Inc.
# Copyright 2017 The TensorFlow Authors. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
=====
=

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import logging
import random
import time
import json
import os.path

from flask import Flask, jsonify, request

import numpy as np
import tensorflow as tf

app = Flask(__name__)

def load_graph(model_file):
    graph = tf.Graph()
    graph_def = tf.GraphDef()

    with tf.gfile.GFile(model_file, "rb") as f:
        graph_def.ParseFromString(f.read())
    with graph.as_default():
        tf.import_graph_def(graph_def)

    return graph

def read_tensor_from_image_file(file_name, input_height=299, input_width=299,
```

```

        input_mean=0, input_std=255):
    input_name = "file_reader"
    output_name = "normalized"
    file_reader = tf.read_file(file_name, input_name)
    image_reader = tf.image.decode_image(file_reader, channels = 3,
        name='image_reader')
    float_caster = tf.cast(image_reader, tf.float32)
    dims_expander = tf.expand_dims(float_caster, 0)
    resized = tf.image.resize_bilinear(dims_expander, [input_height,
input_width])
    normalized = tf.divide(tf.subtract(resized, [input_mean]), [input_std])
    sess = tf.Session()
    result = sess.run(normalized)

    return result

def load_labels(label_file):
    label = []
    proto_as_ascii_lines = tf.gfile.GFile(label_file).readlines()
    for l in proto_as_ascii_lines:
        label.append(l.rstrip())
    return label

@app.route('/')
def classify():
    file_name = request.args['file']

    t = read_tensor_from_image_file(file_name,
                                    input_height=input_height,
                                    input_width=input_width,
                                    input_mean=input_mean,
                                    input_std=input_std)

    with tf.Session(graph=graph) as sess:
        start = time.time()
        results = sess.run(output_operation.outputs[0],
                           {input_operation.outputs[0]: t})
        end=time.time()
        results = np.squeeze(results)

        top_k = results.argsort()[-5:][::-1]
        labels = load_labels(label_file)

    print('\nEvaluation time (1-image): {:.3f}s\n'.format(end-start))

    for i in top_k:
        print(labels[i], results[i])

    output_file = '{0}-output.txt'.format(os.path.splitext(file_name)[0])
    print('Output results file: {0}'.format(output_file))
    with open(output_file, 'w') as f:
        json.dump([labels,results.tolist()], f, indent=1)
    return jsonify(labels,results.tolist())

if __name__ == '__main__':
    # TensorFlow configuration/initialization
    model_file = "output_graph.pb"

```

```

label_file = "output_labels.txt"
input_height = 299
input_width = 299
input_mean = 128
input_std = 128
input_layer = "Mul"
output_layer = "final_result"

# Load TensorFlow Graph from disk
graph = load_graph(model_file)

# Grab the Input/Output operations
input_name = "import/" + input_layer
output_name = "import/" + output_layer
input_operation = graph.get_operation_by_name(input_name)
output_operation = graph.get_operation_by_name(output_name)

# Initialize the Flask Service
# Obviously, disable Debug in actual Production
app.run(debug=True, port=8000)

```