DEVELOPMENT OF MODEL FOR EARLY IDENTIFICATION OF TOMATO PLANT DAMAGES CAUSED BY *TUTA ABSOLUTA*

Lilian Emmanuel Mkonyi

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master's in Information and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

October, 2020

ABSTRACT

Plant pests and diseases challenge the agricultural sector. A high-yielding crop, such as tomato which has the potential to increase income of smallholder farmers, its production is threatened by invasive pest called *Tuta absoluta*. Despite many efforts made by farmers in its management, has continued to be a great constraint, hence calling for scholars to devise approaches of identifying and combating it before causing great losses to farmers. This study introduces, a deep learning based approach for the identification of the pest at early stages of tomato growth through classification of tomato leaf images. In this study, the Convolutional Neural Network architectures (VGG16, VGG19 and ResNet50) were trained on tomato images dataset captured from the field containing healthy and infested tomato leaves. Evaluation of performance for each classifier was done by considering accuracy of classifying the tomato leaf into correct category. Experimental results showed that VGG16 attained the highest accuracy of 91.9% in classifying tomato plant leaves into correct categories. This model can be deployed and used to establish tool for early detection of *Tuta absoluta* pest invasion at early tomato growth stages that will be used by farmers and extension officer.

DECLARATION

I, Lilian Emmanuel Mkonyi, do hereby declare to the Senate of The Nelson Mandela African Institution of Science and Technology that this dissertation is my own original work and that it has neither been submitted nor concurrently submitted for a degree or similar award in any other institution.

27/11/2020

Lilian Emmanuel Mkonyi Date

This declaration is confirmed by:

27/11/2020

Dr. Dina Machuve Date

27/11/2020

Dr. Baraka Maiseli Date

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by The Nelson Mandela African Institution of Science and Technology a dissertation titled: Development of a Model for Early Identification of Tomato Plant Damages Caused by *Tuta absoluta*, submitted in partial fulfillment of the requirements for the degree of Master's in Information and Communication Science and Engineering of The Nelson Mandela African Institution of Science and Technology.

Approved by:

[signature]	27/11/2020
Dr. Dina Machuve	Date
[signature]	27/11/2020
Dr. Baraka Maiseli	Date

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Almighty God, for His enduring mercy to me until completion of my Masters studies at The Nelson Mandela African Institution of Science and Technology. The journey wasn't easy but by His grace I have been able to reach the end. It is a great pleasure to thank many people that supported me during my studies.

I would like to express my deepest gratitude to my supervisors Dr. Dina Machuve and Dr. Baraka Maiseli (University of Dar es Salaam) for their tireless guidance towards completion of this research. I appreciate the support and valuable inputs from my classmates (ICSE cohort 7) while undertaking this work. I would like to thank the African Development Bank (AfDB) for providing me the funds that supported this work financially.

My heartfelt appreciation to my sponsor African Development Bank, through project No. P-Z1-IA0-016, and grant No. 2100155032816 for granting me full scholarship as well as supporting all activities related to financial matters in this research. This work would not be successful without their support. My sincere appreciation to my Employer University of Dar es Salaam, thank you for granting me study leave that enabled me to concentrate on my studies.

Lastly, I would like to thank my family members; my husband David, my children Ian and Ivy, my mother Faresiana, my mother-in-law Cecilia, my siblings (Emmaluice, Dorcas, Tumaini and Geofrey) and all other members who could not be mentioned here for their prayers, encouragement and patience during my study period.

May God bless you all.

DEDICATION

To my lovely husband David Materu and our lovely children Ian and Ivy; your love, patience, encouragement and acceptance have motivated me to accomplish this work.

To my lovely mama Faresiana Macha, thank you for your moral and material support at all times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

CNN	Convolutional Neural Network
COCSE	Communication and Computational Science and Engineering
DL	Deep Learning
FN	False Negative
FP	False Positive
NM-AIST	Nelson Mandela Institution of Science and Technology
ResNet	Residual Network
SGD	Stochastic Gradient Descent
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group
GDP	Growth Domestic Product
NBS	National Bureau of Statistics
NLS	Northern Leaf Bright

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

Tanzania as any other country depends on various sectors for its economic growth. Agriculture is one of primary sectors contributing to the growth of National Gross Domestic Product (GDP), food security as well as poverty reduction (Lyatuu, Nie & Fang, 2015). It is considered as the national's backbone as it employs up to about 65% of Tanzanian (Paul & Lema, 2018). Farmers cultivate different kinds of crops, fruits and vegetables that contribute to nutritional improvement and commercial purposes. For many years tomatoes have been grown in different parts of the country at an estimated total area of 26612 hectares per annum (Mutayoba & Ngaruko, 2018).

Tomato (*Lycopersicon esculentum*) is a nutrition-rich and an edible plant that is widely grown throughout the world (Schreinemachers, Simmons & Wopereis, 2018). In Tanzania, about 247135 tons of tomatoes were harvested in the total area of 54520 ha in 2016; this is equivalent to 64% of all fruits and vegetables (NBS, 2017). Tomato enables small-scale farmers to earn an income and therefore, the plant contributes largely to poverty reduction.

However, farmers face various challenges such as climate change, lack of tools, market fluctuation as well as pests and diseases that attack crops and lead to decrease in production (Arah, 2015). These factors hinder the prosperity of agricultural sector that could be of benefit to farmers as well as national economy at large. It is important to consider these challenges and find appropriate solutions to overcome them in order to improve productivity.

The production of tomato is threatened by an invasive pest called tomato leafminer (*Tuta absoluta*), which tends to attack the plant and weaken its growth and yield capacity. The pest has continued to be a major drawback to tomato production in the world. Tomato leafminer was originated from South America and then it spread to other parts of the world. In Tanzania the first occurrence was reported in Ngarenanyuki, Arumeru District (Arusha) in 2014 (Chidege *et al.*, 2016) and currently it has spread into around 14 regions. The pest has a high reproductive rate of around 12 generations per year. The mature female can lay between 250-300 eggs at once, and has four development stages in its life cycle: Egg, larva, pupa, and adult. The second stage (larva) is the most dangerous one because the pest at this stage can mine, develop and feed on leaves and fruits of tomato plant (Guimapi *et al.*, 2016). The larva has a lifespan of 10–15 days after which it turns to pupa then adult. Their hiding habit makes it difficult to control them due to inability of pesticides to reach them. Therefore, if the larva is left uncontrolled at the early stages of its growth, it may consume all plants in the farm (Zekeya & Chacha *et al.*, 2017).

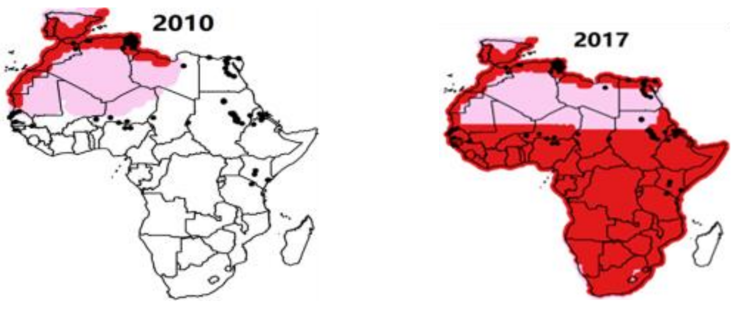
Management of tomato leafminer has continued to be a great constraint in the industry of tomato production, hence calling for scholars to devise approaches of identifying and combating it before causing great losses to farmers. Recent statistics show that tomato farmers, the main host for the pest, have continued to incur heavy yield losses, ranging from 80%-100%, due to invasion of the pest. Some of them have decided to switch to other low income generating crops in order to secure their economic livelihood (Chidege *et al.*, 2016; Zekeya, Ndakidemi, Chacha & Mbega, 2017).

In South America where the invasion started, farmers have continued to deal with the pest by biological control and other non-chemical pest management techniques (Gervassio, Aquino, Vallina, Biondi & Luna, 2019; Schäfer & Herz, 2020). These techniques are considered to be potential control tools as they have helped them to overcome the invasion. However, the identification of the pest is still done manually through physical diagnosis of the plants to see whether they are infected by *Tuta absoluta* or not. The major difference between South America and Sub-Sahara African countries is their improved extension services that helps them to perform faster diagnosis thereby reducing the disease burden.

In order to manage the spread and impact of *Tuta absoluta*, early identification of invasion is very important (Tonnang, Mohamed, Khamis & Ekesi, 2015) as it will help to control the population dynamics of the pest. Among the available approaches to address the issue, deep learning has demonstrated successful results (Lecun, Bengio & Hinton, 2015). Various deep learning techniques have been applied to identify, classify, and quantify diseases, pests and stress on different crops. The Convolutional Neural Network (CNN) (Yann & Yoshua, 1995) provides sophisticated ways of image analysis, and thus facilitates diagnosis of plant diseases. There is a need to apply these advanced techniques to develop more effective approaches for identifying early invasion of *Tuta absoluta* in tomato, so as to reduce the yield loss incurred by tomato growers. Therefore, this study aims at using the CNN to develop a model for identifying the invasion of *Tuta absoluta* in tomato at early stages. The approach reinforces classification of leaf images collected from a field setup in a controlled environment. The controlled environment refers to preventing the spread of *Tuta absoluta* to other neighboring tomato fields using net house.

1.2 Statement of the Problem

The invasion of *Tuta absoluta* has been causing great economic loss to tomato growers in the world. The pest invaded Africa in 2008 and by 2017; *Tuta absoluta* had spread to about 41 among 54 African countries as indicated in Fig.1. Farmers have been incurring higher costs in finding suitable solutions for controlling the pest. Despite existence of various ways of controlling crop pests, such as use of chemical pesticides, pheromone traps for population monitoring and cultivation of resistant tomato varieties; early identification of the pest remains an open-ended research question. In Tanzania, for instance, the agricultural system depends on extension officers as key facilitators in providing farmers with appropriate knowledge on pest and disease management. However, the extension service system is currently conducted locally by limited extension officers' visits to provide training and workshop to meet demands of all farmers in the given area (Maginga, Nordey & Ally, 2018). This challenge calls for a need to integrate sophisticated technologies, including those based on deep learning, into agriculture to identify pest and to maximize productivity (Raj, 2013). Therefore, this study focuses at developing a model for identifying invasion of *Tuta absoluta* at early stages to enable application of appropriate pesticides to control them and rescue the loss incurred by farmers.



(a) *Tuta absoluta* spread in Africa, 2010 (b) *Tuta absoluta* spread in Africa by 2017

Figure 1: The spread of *Tuta absoluta* in Africa within few years (Guimapi *et al.*, 2016)

Figure 2 shows the general impact of *Tuta absoluta* in tomato plants. (a) is the lifecycle of the pest with four development stages in which larvae mine and develop in tomato leaves, (b) is the impact of the pest on tomato leaves and fruits and (c) shows the impact of the pest invasion on uncontrolled farm at late stages. From Fig. 2 it can be seen that the impact of *Tuta absoluta* invasion is a serious threat to tomato productivity that needs an early intervention.

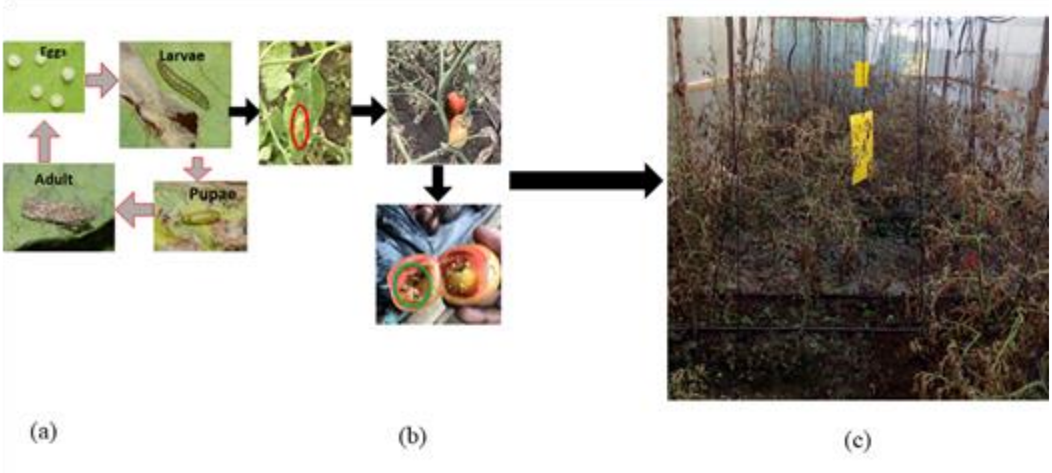


Figure 2: Impact of *Tuta absoluta* on tomatoes

1.3 Rationale of the Study

Deep learning approaches for image recognition have gained experience in recent years, various experts have been using them in identifying and classifying various patterns contained in image data. The CNN has thus become important technique of diagnosing various plant stresses. Extension officers and farmers have been using visual inspection as the method for diagnosing crops. However, they lack accuracy as sometime there is similarity between features associated with the infections. As a non-destructive method CNN provides sophisticated ways of image analysis and thus, facilitates diagnosis of plant diseases. Through developed model farmers will be able to accurately diagnose tomato crop and identify presence of *Tuta absoluta* pest and take appropriate measures.

1.4 Research Objectives

1.4.1 General Objective

To develop a deep learning model for early identification of tomato plant damages caused by *Tuta absoluta.*

1.4.2 Specific Objectives

(i) To identify the requirements for developing a model for identifying damages of tomato plant caused by *Tuta absoluta.*

(ii) To develop a deep convolutional neural network model for early identification of tomato plant damages caused by *Tuta absoluta.*

(iii) To evaluate the developed model.

1.5 Research Questions

The research answers the following questions:

(i) What are the requirements for developing a deep learning model for identifying damages of tomato plant caused by *Tuta absoluta*?

(ii) How can the model be developed?

(iii) What is the performance of the developed model?

1.6 Significance of the Study

Beneficiaries of this study are various tomato growers from different parties of Tanzania. The increase in product loss due to invasion of *Tuta absoluta* have necessitated the need of automating identification of the pest at early stages of tomato growth. Developing the model for early identification of the tomato pest will be of great achievement as farmers will be able to use it to identify presence of *Tuta absoluta.* This will enable them to act appropriately to overcome the invasion thereby increasing the productivity and avoid any great destruction that might be caused by their spread at later stages. Furthermore, the study have significant contribution to the body of knowledge through provision of a dataset containing images of tomato leaves infected by *Tuta absoluta* that can be used by other researchers in their studies.

1.7 Delineation of the Study

The image dataset used in this study was collected from the farm setup in a controlled environment inside a net house. In this study the behavior of the pest has been assumed the same regardless of agronomic factors. A layout of the plots in the nethouse considered the following agronomic factors: Type of irrigation used such as drip irrigation, furrow irrigation and basin irrigation, and the place where the farm is situated. The resulting model can be used with a dataset collected from different field and yet produce good results.

CHAPTER TWO

LITERATURE REVIEW

2.1 Tomato Infections

Tomato (*Solanum lycopersicum*) is an edible fruit from the nightshade family (Solanaceae) native to South America. Tomato is a very important vegetable due to its nutritional value as it a great source of vitamin C, potassium, folate and vitamin K (Frusciante *et al.,* 2007). It is been used in different ways, either raw or cooked, in many dishes, as salads and in drinks. It is also used for commercial purposes and processed into different kinds of products thereby contribute to the income of its growers.

Tomato cultivation is important as it benefits the farmers; however, it is vulnerable to various kinds of infection including pests and diseases. These pests and diseases are spread through the soil, infected water, wind, infected farming tools, animals and gardeners. Some common diseases affect tomato production include, septoria leaf spot, early blight, fusarium wilt, late blight, southern bacterial wilt and verticillium wilt. Also common pests that affects tomatoes include leaf miners and whiteflies (Park, 2017). These infections affects tomato plants in different growth stages and hence needs various forms of intervention including mechanisms for detecting them at early stages.

2.2 The Tomato Leaf miner (*Tuta absoluta*)

The tomato leaf miner is a devastating pest of tomato that originated from South America and, spread to: Europe, Africa, Middle East and Asia and has continued to be a major drawback to tomato industry in the world. The pest can spread quickly as its dispersal is mainly by; wind, travelling a number of kilometers by flying and can also be carried by human being through trading of infected tomatoes or seeds unknowingly (Zekeya & Chacha *et al.*, 2017). In Tanzania the first occurrence was reported in Ngarenanyuki, Arumeru District in Arusha in 2014 (Chidege *et al.*, 2016) and currently it has spread into most regions. The pest has high reproductive rate of about 12 generations per year. Mature female can lay 250 to 300 eggs at once and has four development stages in its life cycle (egg, larva, pupa and adult) (Guimapi *et al.,* 2016). The larva is the most dangerous stage as they feed and develop in leaf and mines in the inside of the leaf between the upper and lower epidermis but, may also be found inside fruits and stems. It has a lifespan of up to 10-15 days after which it turns to pupa then adult

(Cuthbertson *et al.*, 2013). Their hiding habit makes it difficult to control them due to inability of pesticides to reach them (Zekeya & Ndakidemi *et al.*, 2017).

2.3 Deep Learning

Deep learning is a class of techniques in machine learning technology, consisting of multiple processing layers that allow representation learning of multiple level data abstraction. The strength of deep learning is due to its capacity to create and extrapolate new features from raw representations of input data without having to be told explicitly which features to use and how to extract them (Lecun *et al.*, 2015; Lee, Chan, Mayo, & Remagnino, 2017).

2.3.1 Image Classification

Deep learning have been applied in various fields including computer vision, natural language processing, speech recognition, bioinformatics just few to mention. Specifically in computer vision, deep learning have demonstrated great achievement in image classification, object detection with very high accuracy. This is by using CNN which takes in an input image, process it and classify it under certain categories. The CNN model in training and testing will pass each image in series of convolutional layers with filters (kernels), pooling layers, fully connected layers (FC) and apply an activation function to classify an object with probability values between 0 and 1.

Figure 3 below is an overview of a CNN architecture showing how an image of a real world object is passed through a series of layers for classification. In here it is briefly explained what is happening in each layer.

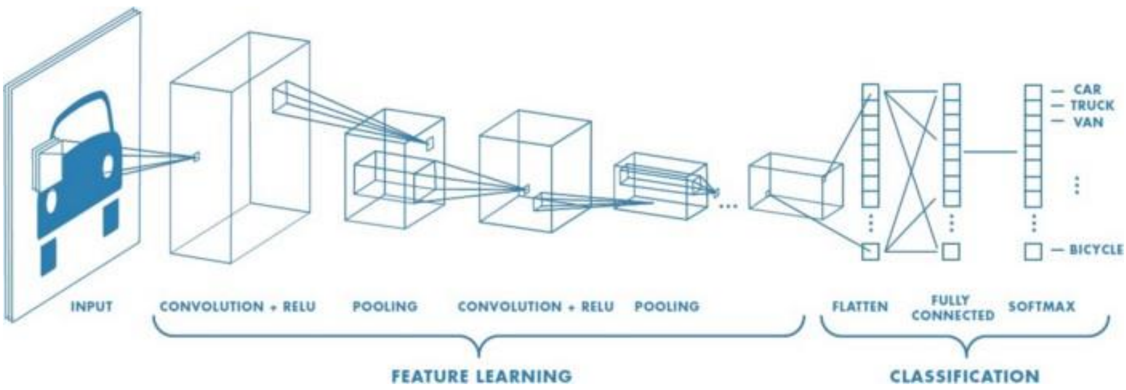


Figure 3: Convolutional Neural Network Architecture (Accessed from www.medium.com)

(i) Convolutional layer

Is a layer where convolution operation is performed. Convolution is a mathematical operation, which is used to find the pattern in inputs or filter out the features. Figure 4 below shows an image of 5 X 5 convoluted with a filter of size 3 X 3. The dimensions of an image will be reduced after applying convolution operation on it. In CNN Convolutional layer is the first layer that is used to perform feature extraction from an input image. It consists of input feature maps, feature extractor and output feature map. At each convolutional layer there was filters of 3 X 3 taken to slice an image and map them for it to learn different portions of an input image and a stride that is the number of pixel shift over input matrix used to move the filter. As convolution operation reduces dimension of an image, to maintain its original size zeros were added into the image boundary, a process known as padding.

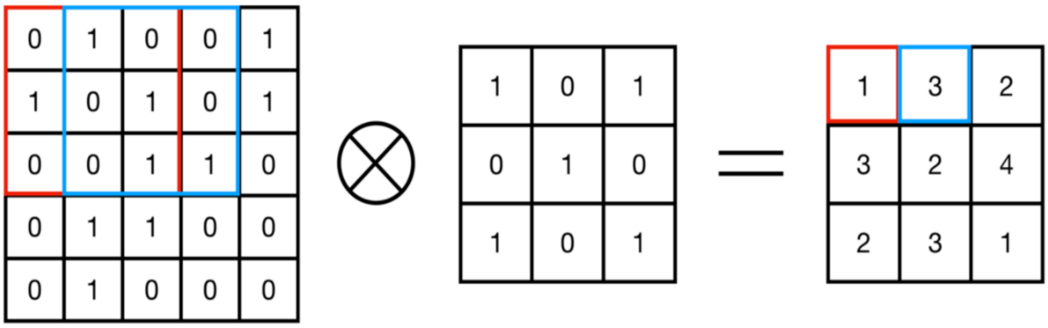


Figure 4: Convolution

(ii) ReLU

Is a short form of Rectified Linear Unit for a non-linear operation. It is the commonly used activation function in the neural network. The ReLU function has an output that is either zero or a positive number, this implies that an activation function will return zero if an input is negative and a number if an input is positive. The main purpose of using it in in Convolutional neural network is to introduce non-linearity and avoid negative values.

(iii) Pooling layer

Is the next layer after convolutional layer, it is an approach for decreasing the image dimensions but preserve important information. The feature maps, output of the convolutional layer are the inputs to this layer. There were MaxPooling that pools maximum value of pixel in feature map, and AveragePooling that computes average value of each region in a feature map. MaxPooling is the most popular form of pooling. For example in Fig. 5 a filter of 2 X 2 size and a stride of 2 were used to perform pooling operation on an input image.

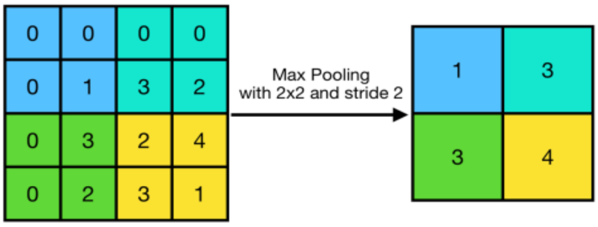


Figure 5: Pooling

(iv) Fully Connected Layer

Receives input from previous layers and performs classification to output the value of certain predicted class. The value of output would do the matrix multiplication with weights and bias, then followed by matrix flattening into vector and fed into a fully connected layer as in Fig. 6. Classification is performed inhere by using an activation function (softmax taken as example) to classify an image into its category.

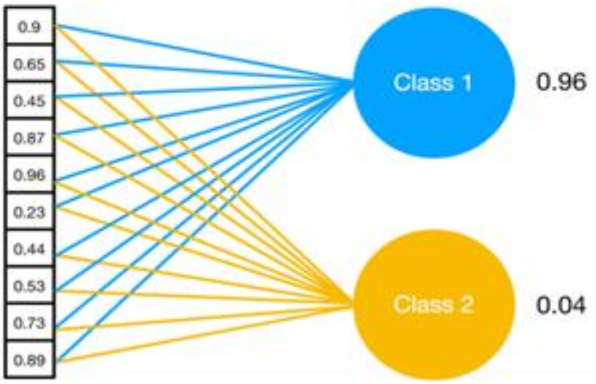


Figure 6: Fully connected layer

2.3.2 Convolutional Neural Network Architecture (CNN)

The CNN architecture is formed by a stacking different layer that transforms the input object into an output. In this work, a reviewed number of CNN architectures have been made that will be used to train the classifiers.

(i) VGGNet

The VGGNet is a CNN model proposed by (Simonyan & Zisserman, 2014) from the University of Oxford. It is trained on images from the ImageNet database. It receives an input image of 224 X 224px size and return a vector of size 1000 with probability of belonging in each class. It has two varieties VGG16 that has 13 convolution layers, 3 fully connected layers and 5 pooling layers and VGG19 with 16 convolution layers, 3 fully connected layers and 5 pooling layers. The VGGNet can classify images into 1000 object categories, such as keyboard, mouse, pencil and many animals.

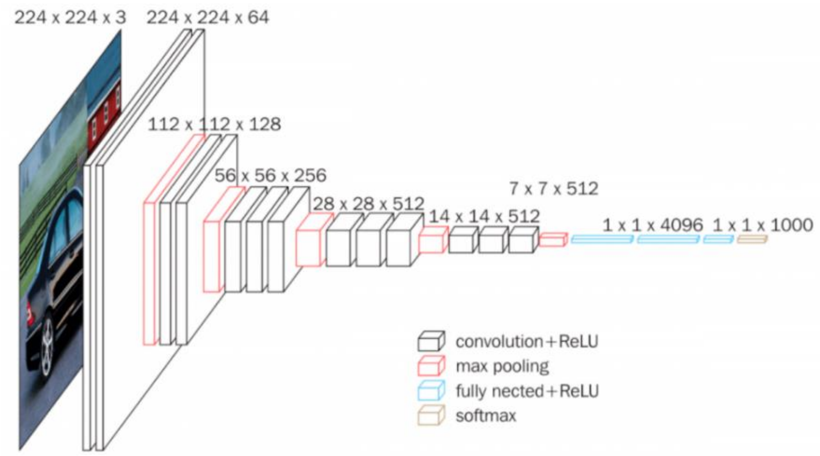


Figure 7: The VGGNet architecture (Accessed from www.researchgate.net)

(ii) ReseNet50

The ResNet50 is a 50 layers residual network. It is a CNN that is trained on more than a million images from the ImageNet database (He, Zhang, Ren & Sun, 2016). It is 50 layers deep and can classify images into 1000 object categories, such as keyboard, mouse, pencil and many animals. It accepts an input image of 224 X 224px. It uses skip connection to add output from previous layer to the next layer. The ResNet was a winner of ImageNet challenge 2015.

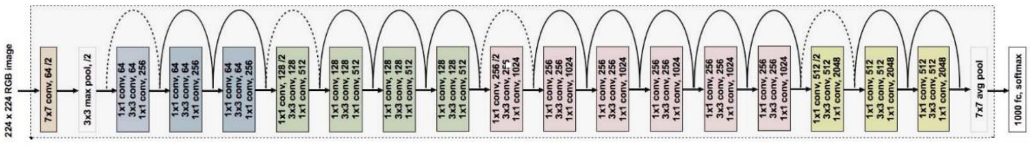


Figure 8: The ResNet architecture (Rezende, Ruppert, Carvalho, Ramos & De Geus, 2017)

2.4 Related Works

2.4.3 Deep Learning in Computer Vision

Recently, application of deep learning has been witnessed in various computer vision tasks such as image classification and object detection. Lecun *et al.* (2015) predicted the future of deep learning to be most important as it will bring about transformation in artificial intelligence and outperform traditional machine learning approaches which are limited in their capability to process raw data. Computer vision is one among these transformations as it allows machine to be fed with data particularly images captured and automatically extract features required for classification or detection. Deep learning methods have multiple representation levels to allow the slight transformation of each representation from low to higher level so that complex functions can be learned.

2.4.4 Computer Vision in Agriculture Crops

Advancements in technology has made great revolution in agriculture. This is attributed by the integration of various devices and mechanisms in automating agricultural activities, which have affected the sector at large. Recently, it has been witnessed an increase in yield and easy disease diagnosis brought by this advancement. Deep learning is one of the advancement that have revolutionized many sectors including agriculture.

Due to the fact that farmers need to produce health crops, scientists have turned their focus towards plant stress identification. Deep learning through computer vision have been used to identify plant stresses and quantifying their level of impacts (Singh, Ganapathysubramanian, Singh & Sarkar, 2016). The method have been successful as it uses non-destructive sampling by just taking images of the plant rather that detaching some parts of the plant. Various literatures have indicated the usefulness of this techniques and few of them have been included in this study.

Ramcharan *et al.* (2017) used a pretrained inception V3 to detect incidence of three cassava diseases and two pests on the image dataset with 11670 images collected from a field in Tanzania. The model could correctly identify the diseases and pest damages with various accuracies: 98%, brown leaf spot; 96%, red mite damage; 95%, green mite damage; 98%, cassava brown streak disease; and, 96%, cassava mosaic disease. The study recommended transfer learning as a powerful deep learning technique for developing highly performing classifiers.

Maize, the source of starch crop grown worldwide, is also affected by diseases and pests that have devastating effects on its productivity, a consequence that threatens food security. Dechant and his colleagues (2017) used CNN to detect a disease called Northern leaf blight (NLB) in maize. The study involved inoculation of maize leaves with fungal, a causal agent of NLB, for acquiring dataset from the infected plant. The analysis was carried out on 1796 images composed of health and infested images, and the authors' model yielded an accuracy of 96.7% on the dataset.

Ouppaphan (2017) used a pretrained deep learning model to identify three corn leaf diseases. They used PlantVillage dataset containing 8506 healthy and unhealthy corn leaf images; the unhealthy ones had the following diseases: Common rust, Northern blight, and Gray spot. The results obtained after training the model were 98.95%, 98.25%, and 98.79% for the ResNet50, Inception V3 and MobileNet respectively. The study revealed that pretrained deep learning models perform well and can be widely adopted in other agricultural crops.

Another work by Liu *et al.* (2017) proposed a deep learning model to classify four diseases from apple leaves dataset containing 1053 leaf images that include unhealthy and healthy. The author used AlexNet architecture to classify apple Mosaic, Rust, Brown spot and Alternaria leaf spot. The approach attained an accuracy of 97.62%.

Furthermore, Lu *et al.* (2017) successfully identified 10 rice diseases from 500 images containing health, and infected leaves of rice and stems. The study used CNN in classifying the images into their respective categories. The authors concluded that CNN yields better results compared with the traditional machine learning techniques of identifying diseases on rice.

2.4.5 Deep Learning Models

Several studies have proposed deep learning as an effective approach of diagnosing various tomato stress. Consequently, witnessed great revolution in agriculture, including substantial increase in crop production. The work done by Zhang *et al.* (2018) for instance, applied CNN architectures, pretrained on 5550 images (from an open access repository), to identify eight tomato diseases: Early blight, yellow leaf curl, corynespora leaf spot, leaf mold, virus, late blight, septoria leaf spot and two-spotted spider mite. All the authors' models could clearly and correctly classify the diseases at the following performances: 95.83%, AlexNet; 95.66%, GoogleNet; and, 96.51%, ResNet50.

Brahimi *et al.* (2017) compared the performances of shallow models (Simple Vector Machine and Random Forest) against pretrained models (AlexNet and GoogleNet) to identify nine tomato diseases. The pretrained deep models outperformed the shallow models by identifying the diseases with high accuracies of 98.66% and 98.53% for AlexNet and GoogleNet respectively; Simple Vector Machine and Random Forest generated accuracies of 94.53% and 95.46% respectively, much lower than those depicted deep models.

Rangarajan *et al.* (2018) applied two pretrained deep learning models, VGG16 and AlexNet, in classifying six tomato diseases. They used images from PlantVillage dataset containing healthy leaves and unhealthy ones with six tomato diseases: Two-spotted spider might, target spot, mosaic virus, late blight, leaf mold and yellow leaf curl virus. The models attained classification accuracies of 99.24% and 96.51% for VGG16 and AlexNet respectively.

Ferentinos used deep learning, specifically the VGG model, to recognize eight tomato plant diseases and two tomato pests from a dataset of 87848 tomato leaves images (Ferentinos, 2018). The model exhibited a great performance of 99.53% in plant disease detection. This high-level performance suggests that CNN are suitable for the diagnosis of plant pest and diseases through the analysis of leaf images.

2.5 Research Gap

Generally, various techniques have been proposed for plant disease detection. These techniques have exhibited good performance, however, there are no studies that have focus in detection of tomato leaf miner (*Tuta absoluta*), also there is no publicly available dataset with images of tomato leaves infected by *Tuta absoluta*. Therefore, this study proposes a deep learning model for *Tuta absoluta* identification by using the images collected directly from the field to enable training the model with real data. The study will contribute the dataset as open source to the research community for further studies in tomato infested with *Tuta absoluta*.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Study Area

This study was carried out in an in-house experiment located in USA-river ward, Arumeru district in Arusha region situated at Northern Tanzania as indicated in the map contained in Fig. 9. This study targets various tomato farmers around the country. As it is carried out in controlled manner, the resulting model can be easily adopted to any farm data and yet produce better results.

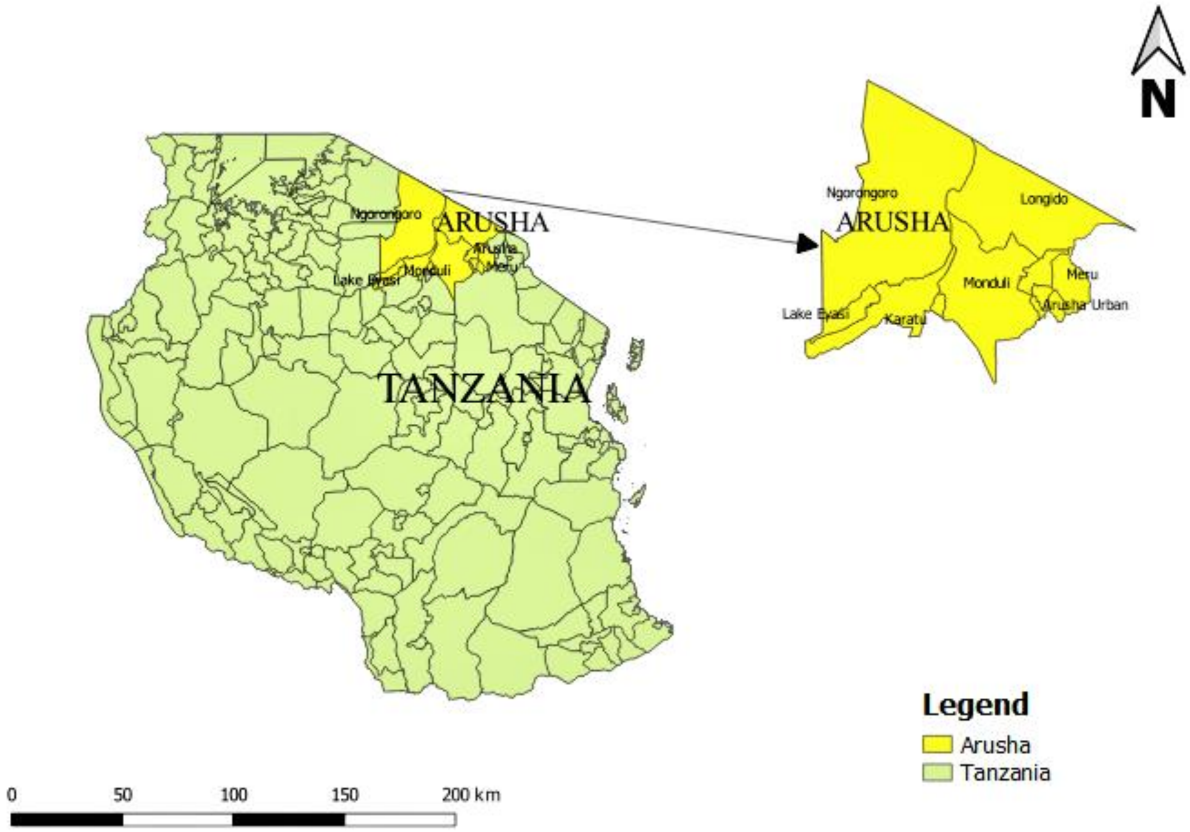


Figure 9: A map showing the study area

3.1.1 Field Trial Setup

Two in-house experiments were conducted between two tomato growth seasons (August-November 2018 and January - May 2019) in the nethouse where we planted tomatoes as shown in Fig. 10. The setup enabled to overcome any incoming pests into our farm and prevent *Tuta absoluta* from getting out from the experiment so as to maintain controlled environment for the study. Images collected from this farm were used to develop and test the model.

Figure 10: A nethouse for field trial on *Tuta absoluta*

3.1.2 Inoculation Process

Two days after planting the seedlings, some of tomatoes were infested (inoculated) with *Tuta absoluta* (controlled from other pests) under commonly practiced agronomic practice at the early growth stage. It involved placing *Tuta absoluta* larvae on top of leaves of tomato plants selected at random so that they can mine inside the leaves and start eating to create the feature that need to carry out the classification task. The process was carried out by an agricultural expert in Fig. 11.

3.2 Data Collection

Images of tomato plants were captured from the nethouse by using Canon EOS Kiss X7 camera with resolution of 5184 X 3456 pixels to makeup the dataset of this study. The dataset contained colored images of healthy and unhealthy tomato plants (where the unhealthy ones were inoculated with *Tuta absoluta* larvae), making 2145 images (330 infested with *Tuta absoluta*) as examined and labelled by an agricultural expert. The images were collected within 14 days from the day of inoculation in each experiment. The focus was on capturing the upper part of the plant at nadir, approximately 40 cm away from the plant, specifically the plant crown because this part is always affected at early growth stages of the plant. Figure 12 shows development of mines associated with *Tuta absoluta* (a) health leaf before inoculation, (b), (c), (d) and (e) are the infected leaf on the 2nd, 5th and 8th days respectively after inoculation with the *Tuta absoluta*. The red bounding box shows the mines.

Figure 11: An agricultural expert performing inoculation in the field

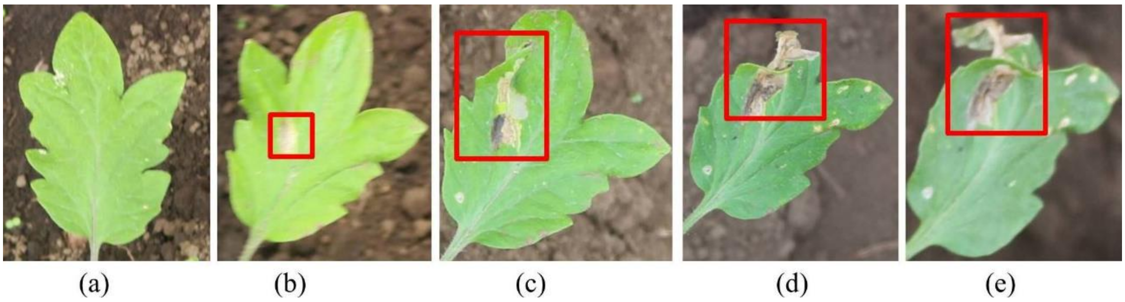


Figure 12: The images showing development the mines associated with *Tuta absoluta* infection

Research framework gives an illustration of the flow of our research. It show how the research will be undertaken from data collection to model development and validation until we deliver an optimized model. Figure 13 show the research framework that depict logical flow of this work work. It starts with tomato leaves image dataset as input, then the images are preprocessed to enhance quality. Upon completion of preprocessing the dataset is subdivided into training, validation and test sets. The training set applied in training classifier, the DeepNet model with the hyper–parameters, the validation set is there for evaluating training process at the end of each epoch. Performance evaluation results from validation will allow doing appropriate hyper-parameter optimization to search for a set of hyper-parameters that give optimal results after which the obtained training model will be tested by using a test set to give out prediction of images contained in the test set. The prediction result will lead us to an optimized classifier.

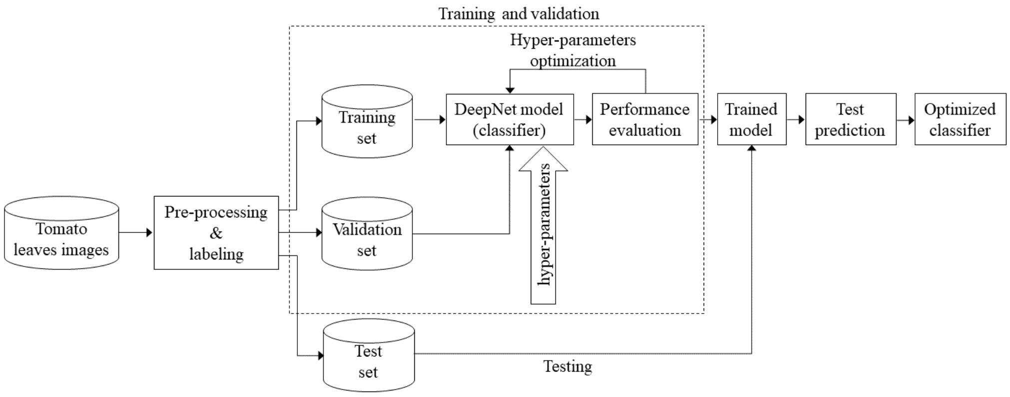


Figure 13: Research Framework

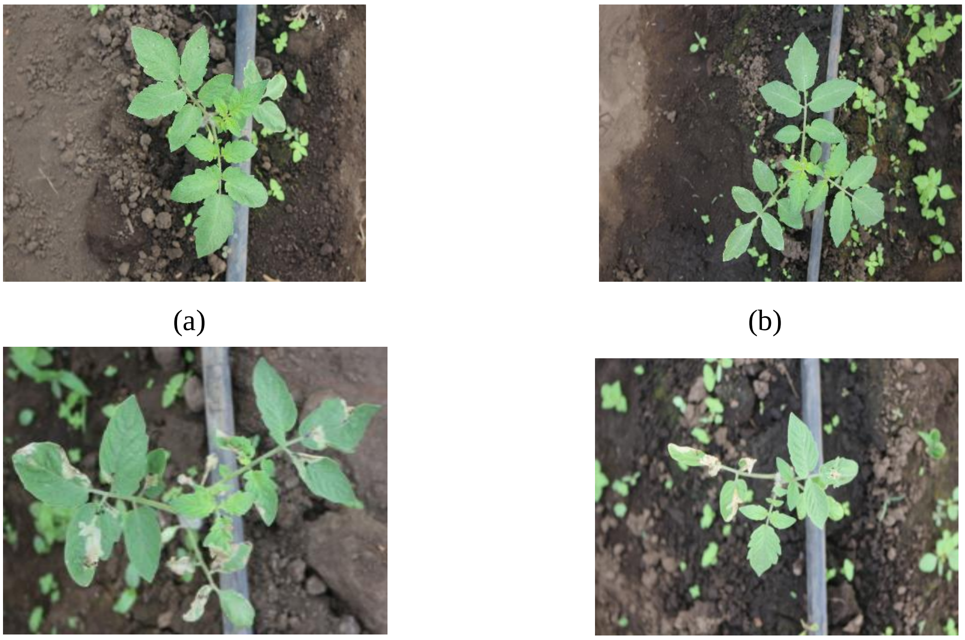
3.3 Image Preprocessing

Image preprocessing refers to the manipulation of raw image data before feeding it to a deep learning model for the purpose of enhancing its quality. Building a well performing model needs careful consideration of the network architecture and quality of the input data. Therefore, the dataset was pre–processed to allow the proposed model undertake intelligent diagnosis of extracting appropriate features from the images (Jeong, Park & Ha, 2018). The pre-processing involved three stages: image labeling, resizing, and augmentation.

3.3.1 Labelling

The classification task falls under supervised learning that requires input values with their corresponding output. In order to enable the model to carryout correct image classification and to predict the class of unseen data, it is necessary to provide it with labelled input. The images collected were manually labelled into their respective class, being health with images that are not infected with *Tuta absoluta* and non-health class with the images that are infected with *Tuta absoluta*.

The labelling process was done with the help of an agricultural expert. Figure 14 show sample images as labelled by an expert and used during training process. (a) and (b) are the samples labelled as being health (H), (c) and (d) are the samples labelled as being non-health (NH).



(a) (b) (c) (d)

Figure 14: Sample images used during training

3.3.2 Resizing the Images

Images needs to be prepared before being passed into a neural network as CNN architectures have been made to perform optimally on the images of a specific input size being 224 X 224px for VGG16, VGG19 and ResNet50. Keras has a standard resize function which resamples the images to target size by default. For VGGnet and ResNet50 models all the images were resized to 256 X 256 px size for them to be have uniform size to enable resize function to resample them as per architecture requirement.

3.3.3 Augmentation

Deep convolutional neural networks dependes much on the availability of large amount of training data in order to clearly learn features contained in them and attain high classification accuracy on unseen data (Perez & Wang, 2017). However, given the limited time and fund for conducting this study, it was impossible to collect a large quantity of data that will be sufficient to train the model. Training a model with insufficient amount of data leads to overfitting that makes it fit well to training data and unable to generalize testing data.

A helpful solution to this problem is data augmentation, a technique that involves generation of additional data samples through transformations performed on original data. The transformation applied to original data do not change their meaning but produces synthetic data on the fly that is added on the dataset.

In this study ImageDataGenerator class have been used to perform image augmentation in training datagenerator as indicated herein:

(i) **Rescaling:** Was done by multiplying an image with a certain factor provided. For the case of this study, rescaling of images was done by 1./255 factor.

(ii) **Shearing:** This function shears an image with a random shearing angle that is calculated from the given intensity. In this work a shear range of 0.1 was used.

(iii) **Zooming:** This function enlarges an image with a given factor. This study have used a zoom range of 0.2.

(iv) **Flipping:** The image is randomly flipped along the horizontally.

(v) **channel shifting:** Involves random shifting of values of the channels by values specified.

(vi) **Rotation:** Involves rotation of each image upto the angle specified.

3.4 The Deep Learning Approach

The CNN was used in developing the model for identification of the impact of *Tuta absoluta* in infested tomato plants. The choice of CNN is because it is fast and requires less computational power than other traditional machine learning techniques. Also CNN does not rely on handcrafted features rather it performs automatic feature extraction. The image data set will be subdivided into training set to enable the model to learn required patterns and test set that will be used to test the model so as to evaluate its performance.

When building a CNN model one may opt to build it from scratch or use transfer learning. Building a model from scratch involves random initialization of all the parameters in the neural network. It requires well labelled data and a lot of computational resources as compared to using pretrained models through transfer learning. Transfer learning involves reuse of the models that were previously trained on different data and for a similar or different task. In transfer learning, model training starts with some learned weights that come from a pre-trained model. In this work we propose transfer learning approach based on CNN models.

Transfer learning is the best approach in building very powerful classifiers by using insufficient data, by fine-tuning the hyper-parameters of network that was trained on a large dataset (Mehdipour Ghazi, Yanikoglu & Aptoula, 2017). In this study four CNN architectures VGGNet (VGG16 and VGG19) (Simonyan & Zisserman, 2014) and ResNet50 (He *et al.*,

2016) were explored and evaluated their performance with using the dataset on classifying images into correct category. It was expected that convolutional neural network can learn feature contained in the training data automatically and use them to classify unseen data. Hence, will no longer rely on experts in identifying the features associated with *Tuta absoluta* infection.

3.4.1 Experimental Setup

The experiment was carried out in a Desktop computer pre-installed with Ubuntu 18.04 equipped with one Intel Core i9-9900 3.6 GHz Processor (16 Gb RAM) with one GeForce RTX 2080Ti GPU (12 GB memory). Python was used as programming language throughout this work. Neural network implementation was carried by using Keras library as it is suitable for handling neural network models. TensorFlow was used in the backend which offers the high performance numerical computations.

The dataset collected from the field contains more images with healthy tomato leaves than those infected with *Tuta absoluta*. This situation introduces data imbalance, implying a huge difference between number of samples in a class. To reduce the bias that the neural network under this study might encounter towards health samples, number of samples per class should be balanced. Therefore, to address the data imbalance, we used the following approach: 10% of the infected images were held out as test set while the remaining 90% was subdivided into training and validation set in ratio of 75:25, 80:20 and 85:15. For health leaves, the images equivalent to 10% of the infected images were held out for testing and the remaining images were divided into 6 clusters each with 297 images. The images of each cluster were then subdivided into various datasets in the ratios of 75:25, 80:20 and 85:15 as elaborated in Table 1. The overall accuracy is computed by averaging the accuracies over the six runs on the clusters.

Table 1: Dataset division

Class	Dataset	Number of images for training	Number of images for validation	Number of images for testing
H	75:25	223 X 6*	74	33
	80:20	237 X 6*	60	33

	85:15	252 X 6*	45	33
	75:25	74	74	33
NH	80:20	60	60	33
	85:15	45	45	33

*Refers to six clusters, H-images not affected with *Tuta absoluta*, NH-images affected with *Tuta absoluta*.

3.4.2 Training the Classifier

Transfer learning was used to train the classifier under this study on the three architectures (VGG16, VGG19 and ResNet50). It involved using of a model that have been pretrained on large dataset such as ImageNet to train the dataset. Transfer learning was selected because of its ability of improving the performance of a neural network by speeding up the time taken to train models through reuse of models that were trained on similar tasks. Transfer learning allows the use of few data in training a neural network as compared to training from scratch that requires large amount of data.

The approach used in this study involved replacing the fully connected layer of the pretrained architecture, which corresponds to the ImageNet classification task (i.e. 1000 classes), by new fully connected layer to obtain probabilistic outputs for the two classes. Then, the convolutional blocks were fine-tuned for VGGnet and the top residual block for ResNet50. All layers except the new added fully connected layer were firstly freezed, for it to be trained on the output of the final convolutional layer and the weights learned here are the initial values of fine-tuning. Thereafter, the top convolutional layers for VGG16 and VGG19, the top residual block for ResNet50 are unfreezed and trained along with the new fully connected layer.

The hyper-parameters search was conducted when training the models of under this study, so as to get the ones with an optimal performance. They include:

(i) **Epochs:** Number of times which a dataset is passed forward and backward through a neural network. The dataset gets divided into batches for it to be passed through a neural network as it is impossible to pass the entire dataset onto it.

(ii) **Batch size:** Number of samples contained in single batch.

(iii) **Optimizer:** is the function used to update the weight parameters to minimize the loss function. This work used a Stochastic Gradient Descent (SGD) as optimizer.

(iv) **Learning rate:** is the value that used to control the amount in which a model can be changed in response to the estimated error each time the weights of the model are updated. A learning rate of 1e-5 was chosen for this study.

(v) **Dropout:** is a technique that involves ignoring randomly selected neurons during training. It is very important as helps in reducing overfitting. The value of dropout indicates the percentage of neurons whose output will be forced to zero in such a way that the model will learn nothing from them.

(vi) **Momentum**: is method which helps to bring about faster model converging by accelerating gradients vectors in the right directions.

(vii) **Early stopping:** is a technique for controlling overfitting in machine learning models, by stopping training process when performance on validation set start to degrade.

The hyper-parameters indicated in Table 2 achieved an optimal performance as compared to others.

Table 2: Model hyper-parameters

Parameter	**Value**
Epoch	1000
Batch size	8
Optimizer	SGD
Learning rate	1e-5
Dropout	0.5
Momentum	0.9
Early stopping	50 epochs

Each cluster indicated in Table 1 was trained individually; therefore, six clusters were to be trained in each dataset division. The overall performance metrics were obtained by averaging performance along all six runs on a specific dataset division (Wang, Sun & Wang, 2017).

An early stopping was employed to stop training process at the point where performance on a validation dataset starts to degrade. The training process is measured at the end of each epoch by the validation set and if the performance on the validation set decrease as compared to performance at prior epoch (if validation loss starts to increase) the training process is stopped. A stopping patience of 50 epochs was setup that will allow training to continue for additional

50 epochs after the point where performance started to degrade for it to find additional improvement. Thereafter, the learning rate is reduced by 0.2 if there is no improvement observed after the patience, and if the stopping patience is reached again with no improvement, the training is terminated.

After completion of the training process, a trained model was delivered ready for testing. The trained models in all the three architectures were tested by using the images contained in the test set. Evaluation techniques employed after testing are presented in section 3.6.

3.5 Evaluation of the Classifiers

Evaluation of the classifiers involved assessing performance of the model during training and testing processes. In this study, evaluation was performed by using two different ways; by observing the classification accuracy and loss during training process and by using the confusion matrix plotted after observing the prediction on test set.

3.5.1 Training Accuracy and Loss

The training set of dataset was applied to train the CNN models. The main aim of training was come up with a set of model hyper–parameters (weights and biases) that minimize the loss and maximize accuracy. During training process, parameters of the models were updated. Validation set was there to evaluate the model during training process while tuning hyper–parameters to allow selecting the best performing model. At the end of each training process a graph showing epochs versus training and validation accuracy as well as epochs versus training and validation loss were plotted. The graph gives the visualization of how the training process was going.

3.5.2 Confusion Matrix

A confusion matrix gives us a matrix as output and describes the complete performance of the model. It shows correct and incorrect classifications for each class. Confusion matrix plots are used to visualize performance of trained model in predicting the classes of data contained in the test set (unseen data). The test set assess the likely future performance of a model. Figure 16 shows the general layout of a confusion matrix used in models.

True values \ Predicted values	No Tuta	Tuta
No Tuta	TN	FP
Tuta	FN	TP

Figure 15: Confusion matrix

where TP = "True Positive", number of images with *Tuta absoluta* and classified as having *Tuta absoluta*; TN = "True negative", number of images with no *Tuta absoluta* and classified as not having *Tuta absoluta*; FP = "False Positive", number of images with no *Tuta absoluta* and classified as having *Tuta absoluta*; and, FN = "False Negative", number of images with *Tuta absoluta* and classified as not having *Tuta absoluta*.

Confusion matrix is very useful in calculating various evaluation metrics for evaluating models. In here the metrics have been explained in brief and show how they can be calculated from the confusion matrix. Later it was shown the confusion matrix of the prediction of each classifier on the testing set and discussed their performance and chose the best classifier for our problem.

(i) Accuracy

Accuracy in classification problems is the number of correct classification of unseen data done by the model over all classifications made on unseen data. The accuracy is calculated from the confusion matrix by using the formula indicated in Equation (1)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

(ii) Precision

Is the value calculated by dividing total number of correctly classified positive examples by the total number of predicted positive examples. High Precision value indicates an object classified as positive is really positive.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

(iii) Recall

Is the ratio of the total number of correctly classified positive to the total number of positive examples. High Recall value indicates the class is correctly recognized.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

(iv) F1-score

Is the value calculated by dividing total number of correctly classified positive examples by the total number of predicted positive examples. High Precision value indicates an object classified as positive is really positive.

$$F1 - score = 2 \text{ X } \frac{Precision \ X \ Recall}{Precision + Recall} \tag{4}$$

The held out test set containing 66 images was used to evaluate performance of the models in predicting unseen data, confusion matrix was plotted to enable visualizing the prediction results. The test contained 33 images of health leaves and 33 images of infected leaves. Classification results are obtained by using prediction results that is having probability of an image belonging to a particular class. The confusion matrices was plotted for each prediction in order to visualize model prediction performance. Results for the prediction will be presented in the result and discussion section of Chapter 4.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Data Collection Results

The data collection one on the field experiment came up with the dataset of 2145 colored images that were used in training and testing the classifiers.

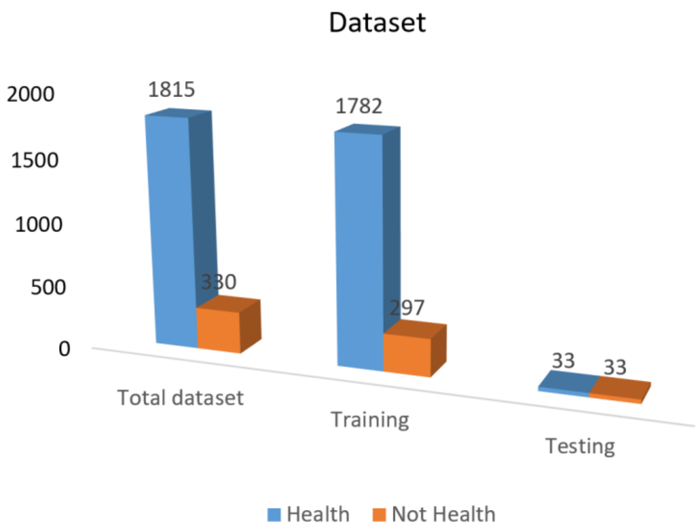


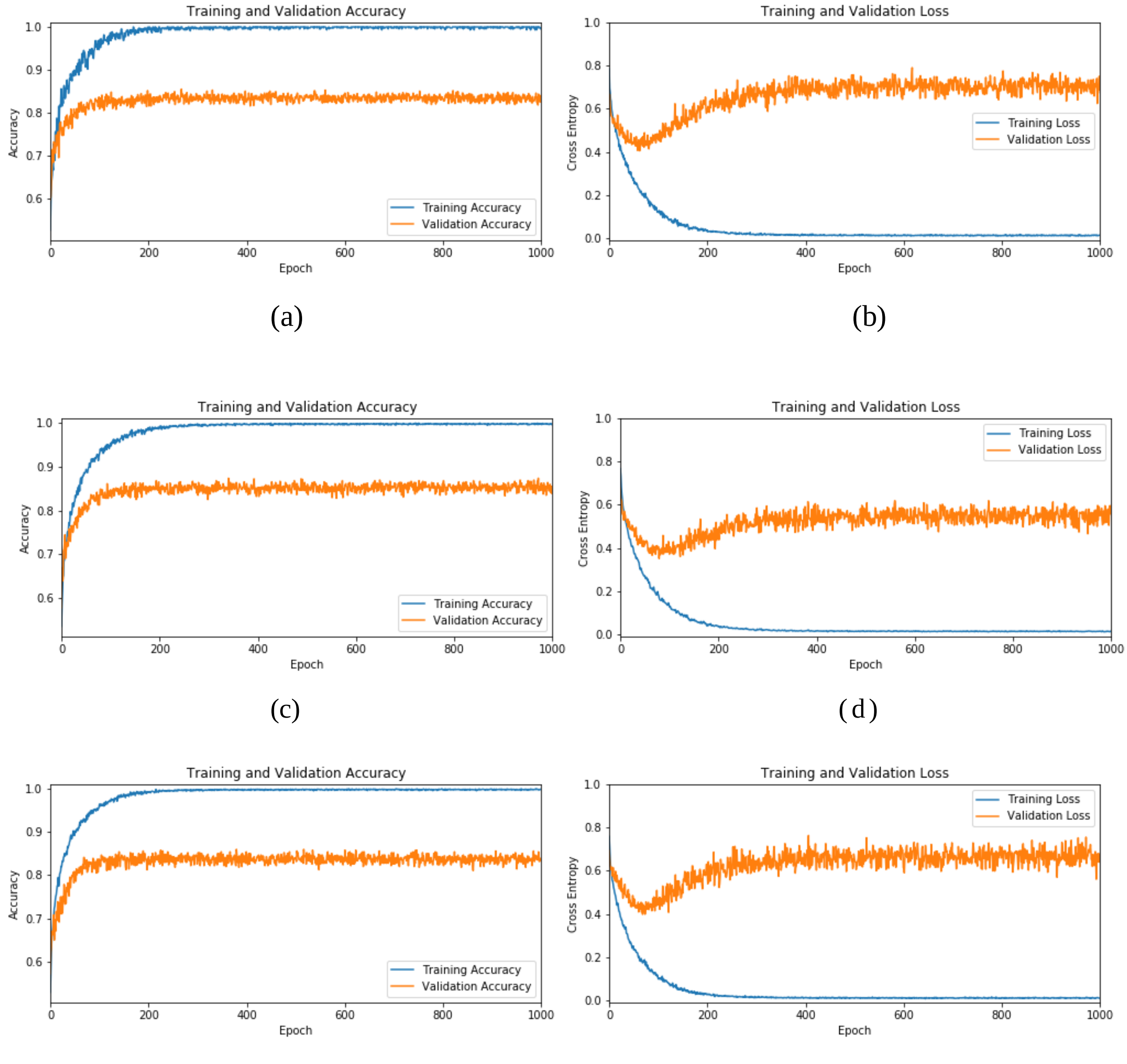
Figure 16: Dataset distribution

4.2 Training Accuracy and Loss Results

As indicated in section 3.5.1 during training process, the validation set is used for evaluation at the end of each epoch. In here included the graphs that depicts training and validation accuracies and losses that helped to visualize the training process. Later these results and perform model test are going to be discussed as well as reporting its results.

4.2.1 Training Accuracy and Loss Graphs for VGG16

The VGG16 architecture was trained on our dataset on both three dataset subdivisions (75:25, 80:20 and 85:15). Training results at each epoch can be visualized in the graphs indicated in Fig. 17. Graphs (a), (c) and (e) are the training and validation accuracies for training on 75:25, 80:20 and 85:15 dataset division respectively whereas (b), (d) and (f) are losses for training on the respective dataset.



(a) (b)

(c) (d)

(e) (f)

Figure 17: Graphs for training VGG16 on the dataset

As observed in Fig. 17 (a), (c) and (e), the training accuracy across each fold starts at the minimal point of around 70% and improve as the number of epochs increases; when approaching the 1000th epoch the accuracy is 100%. This indicates that the model is learning well the features contained in our training dataset. In the same way the validation accuracy starts at the minimal point and start to increase as the number of epoch increase and become steady at around 200th epoch. This also indicates that the model learns well the features contained in our training dataset.

On other hand Fig. 17 (b), (d), (f) the training loss starts at around 50% and keep on de- creasing as the number of epochs increases. Coming to validation loss at initial points the loss starts to decrease until the 200^{th} epoch where it starts to increase slightly and remain almost steady. This observation implies that the model learns well the features contained in dataset at initial and later stages.

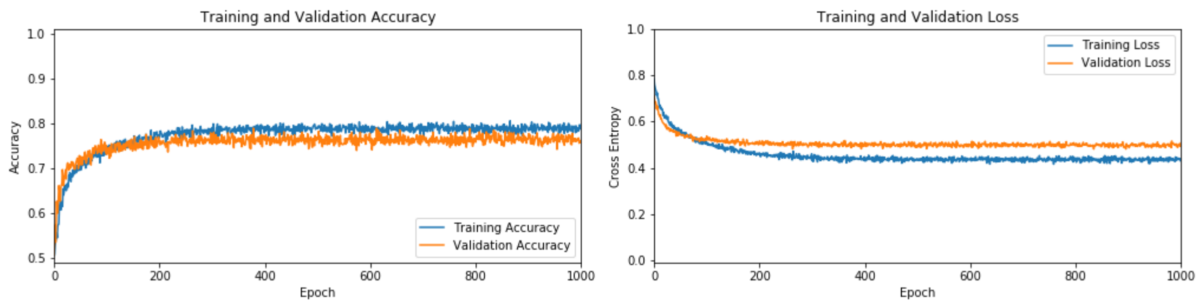
Table 3 shows the summaries of the maximum training and validation accuracies attained during training of VGG16 on the dataset divisions. On both the datasets training accuracy attained the maximum value of 100% and the validation accuracy was highest for 80:20 dataset division.

Table 3: Training performance for VGG16

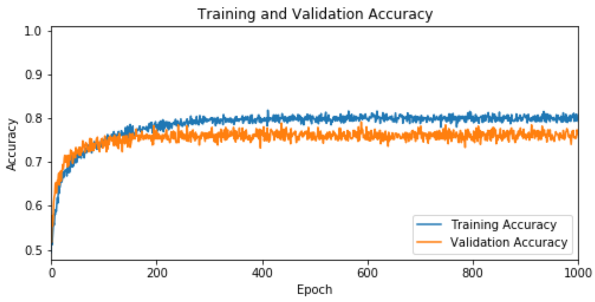
Dataset	Training accuracy	Validation accuracy
75:25	100%	85.53%
80:20	100%	87.30%
85:15	100%	85.90%

4.2.2 Training Accuracy and Loss Graphs for VGG19

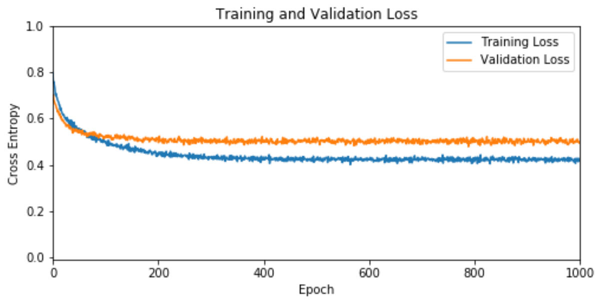
The VGG19 pretrained model was trained on the dataset of this study on both three dataset subdivisions (75:25, 80:20 and 85:15) and the graphs showing the training process have been indicated in Fig.18. Finally, the table showing summaries of performance of model on all architectures have been indicated.



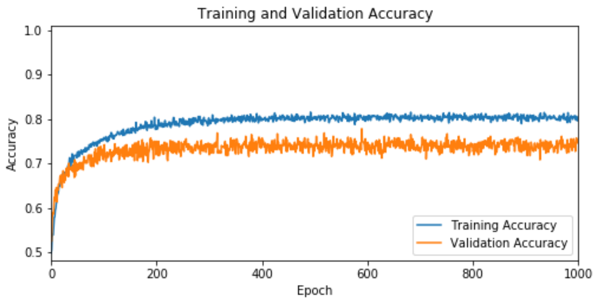
(a) (b)



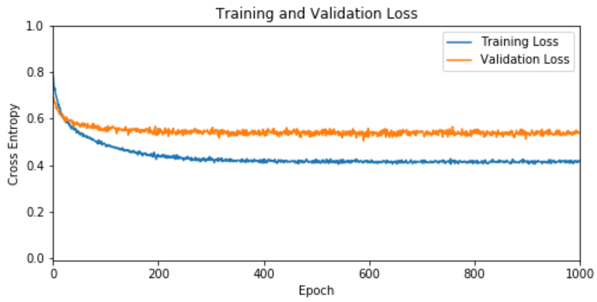
(c)



(d)



(e)



(f)

Figure 18: Graphs for training VGG19 on the dataset

In Fig. 18 (a), (c) and (e), the training accuracy across each cluster starts at the minimal point of around 50% and improve as the number of epochs increases; when approaching the 1000th epoch the accuracy is 80%. This indicates that our model is learning well the features contained in training dataset. In the same way the validation accuracy starts at the minimal point and start to increase as the number of epoch increase and become steady at around 300th epoch. This also indicates that the model learns well the features contained in training dataset.

On the other hand, Fig. 18 (b), (d), (f) the training loss starts at around 70% and keep on decreasing as the number of epochs increases. Coming to validation loss at initial points, the loss starts to decrease until the 200th epoch where it starts to become steady. This observation implies that the model under this study learns well the features contained in dataset at initial and later stages.

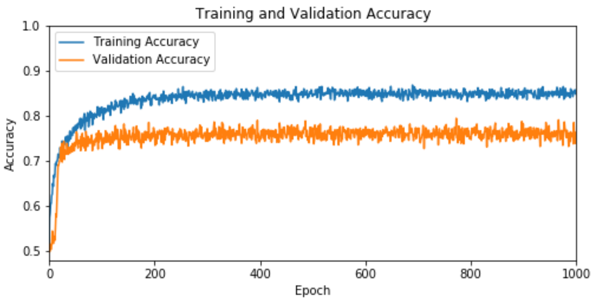
Also from this training it can be seen that the gap between training and validation graph have been decreased. This is a good indication that the models are not overfitting. Table 4 summarizes the accuracies of our models in each dataset. The accuracies are almost similar with minor differences. The model performed well on 80:20 dataset division than in other subdivisions.

Table 4: Training performance for VGG19

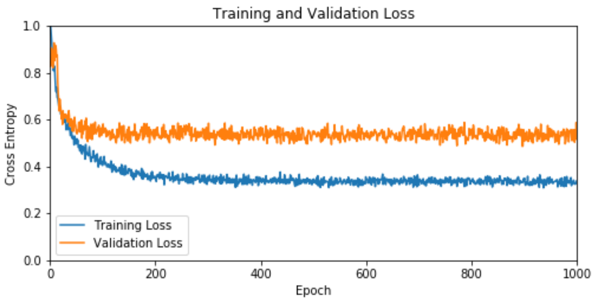
Dataset	Training accuracy	Validation accuracy
75:25	80.56%	78.93%
80:20	81.80%	87.30%
85:15	81.58%	77.84%

4.2.3 Training Accuracy and Loss Graphs for ResNet50

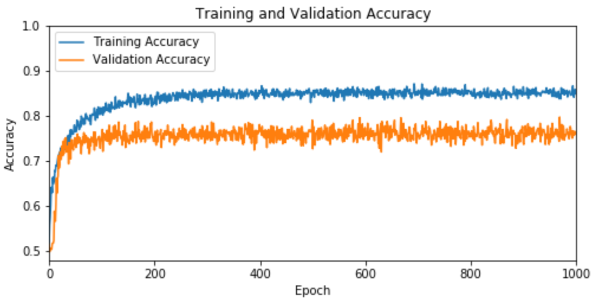
The ResNet50 pretrained model was trained on our dataset on both three dataset subdivisions (75:25, 80:20 and 85:15). Figure 19 shows the training thee learning curves during model training with this architecture. Thereafter, Table 5 summarizes the maximum accuracies attained on training and validation set. Training on 80:20 dataset attained the maximum accuracies as compared to all other dataset subdivision.



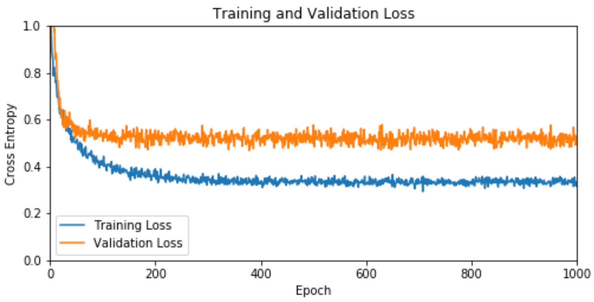
(a)



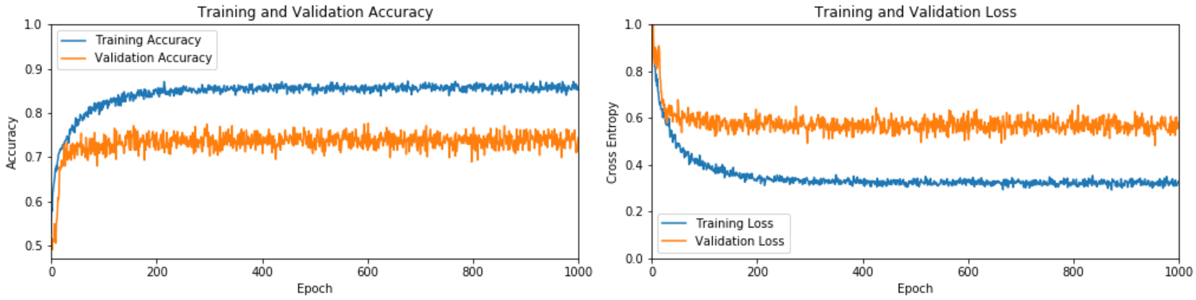
(b)



(c)



(d)



(e) (f)

Figure 19: Graphs for training VGG19 on the dataset

In Fig. 19 (a), (c) and (e), the training accuracy across each cluster starts at the minimal point of around 60% and improve as the number of epochs increases; when approaching the 1000th epoch the accuracy is around 86%. This indicates that the model is learning well the features contained in training dataset. In the same way the validation accuracy starts at the minimal point of about 50% and start to increase as the number of epoch increase and become almost steady at around 350th epoch. This also indicates that the model learns well the features contained in training dataset.

On other hand, Fig.19 (b), (d), (f) the training loss starts at around 70% and keep on decreasing as the number of epochs increases. Coming to validation loss at initial points the loss starts to decrease until around 350th epoch where it start to become steady. This observation implies that the model learns well the features contained in the dataset at initial and later stages.

The gap between training and validation curves is small, this is good sign that the model is able to extract the features contained in both training and validation sets without overfitting. It indicates that upon adding more data the performance can be improved to more satisfactory values.

Table 5: Training performance for ResNet50

Dataset	**Training accuracy**	**Validation accuracy**
75:25	86.87%	79.51%
80:20	87.13%	79.72%
85:15	87.20%	77.65%

4.2.4 Discussion on training results

As observed from sections 4.2.1, 4.2.2 and 4.2.3, all the three architectures VGG16, VGG19 and ResNet50 were trained in dataset and achieved better performance. Some slight variation that have been observed during training are attributed by insufficient number of samples in our dataset. In general, all the three architectures managed to perform at their best with regard to the number of samples we had.

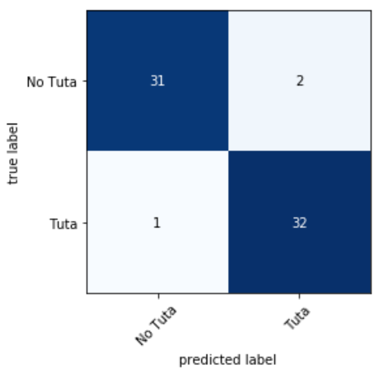
In order to improve training performance in future, there is a need to add more data in the dataset. As it is known that training a deep learning model requires a lot of data for it to be able to extract required features from the data.

4.3 Confusion Matrix Results

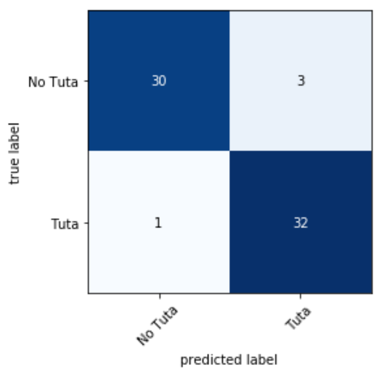
After training three architectures by using training and validation dataset and obtained a well trained models, it came to the point of testing the trained models on the new dataset. The test set used to evaluate the performance contained 66 images; 33 being health and 33 unhealthy. The images were passed to the trained models for it to predict their classes. Prediction results were visualized by using the confusion matrices, and, it is from the values of confusion matrix the evaluation metrics were calculated for evaluating the models' performance on new data.

4.3.1 The Confusion Matrices for VGG16

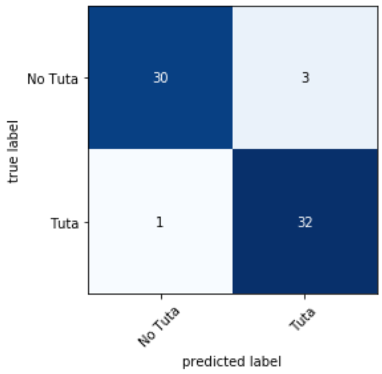
Trained VGG16 classifiers were used to predict the classes of images contained in the test set. Figure 20 (a), (b) and (c) are the confusion matrices for VGG16 model trained on 85:15, 80:20 and 75:25 dataset subdivisions respectively. From each group the best performing confusion matrix was picked.



(a)



(b)



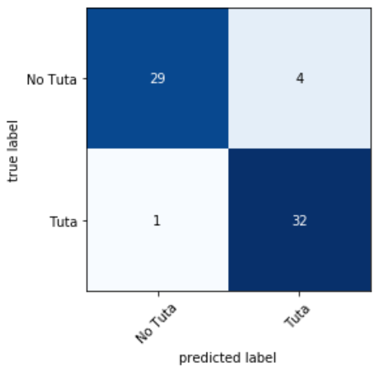
(c)

Figure 20: Prediction results for VGG16 model

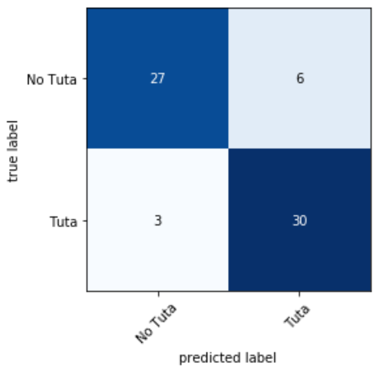
From these confusion matrix plots it can be seen that all the classifiers were able to predict the class of images with high accuracy, whereby the model trained on 75:25 performed the best by classifying almost all images into correct category.

4.3.2 The Confusion Matrices for VGG19

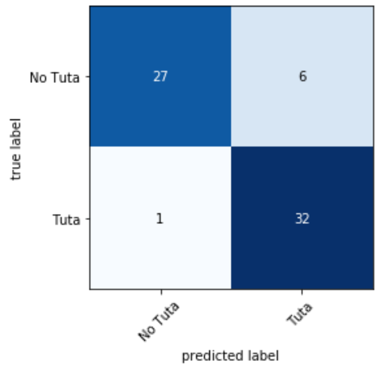
The predictions conducted by using VGG19 model on our test set images can be visualized in Fig. 21 (a), (b) and (c) being for 75:25, 80:20 and 85:15 dataset subdivisions respectively. It can be observed that the classifier trained on 75:25 dataset division performed the best by predicting images into correct categories.



(a)



(b)

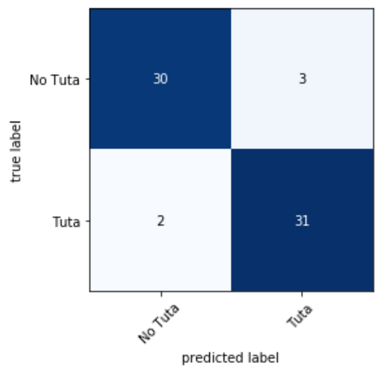


(c)

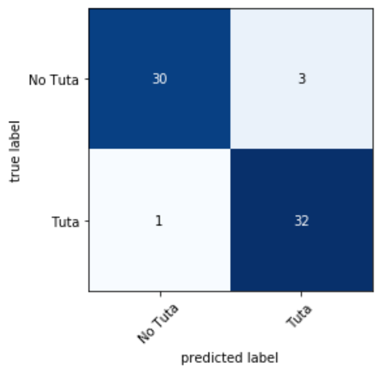
Figure 21: Prediction results for VGG19 model

4.3.3 The Confusion Matrices for ResNet50

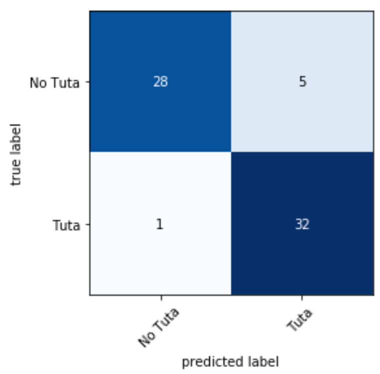
Figure 22 illustrates the prediction results for the ResNet50 model trained on the dataset subdivisions. It can be observed that the model trained on 80:20 dataset subdivision, (b), performed the best by being able to classify almost all images into their correct categories.



(a)



(b)



(c)

Figure 22: Prediction results for ResNet50 model

4.3.4 The evaluation metrics

From the confusion matrices that were plotted during models testing, different metrics were calculated that are used as an evaluation criteria for the classifiers. These metrics are now used to compare performance of the classifiers and come up with the best one. Table 6 reports evaluation metrics that were used to evaluate the classifier. The individual plots on the performance of every classifier is indicated in the subsequent sections.

Table 6: For every Dataset, F1–score{ mean precision, mean recall, overall accuracy}

Dataset	VGG16	ResNet50	VGG19
75:25	0.901{0.909, 0.901, 0.901}	0.852{0.856, 0.853, 0.853}	0.839{0.852, 0.841, 0.841}
80:20	0.906{0.915, 0.915, 0.905}	0.854{0.867, 0.856, 0.856}	0.831{0.853, 0.841, 0.836}
85:15	0.919{0.922, 0.919, 0.919}	0.868{0.871, 0.868, 0.868}	0.831{0.851, 0.833, 0.833}

Overall accuracy was considered as evaluation metric for the experiments. The best performance is attained by VGG16 on 85:15 dataset with an accuracy of 91.9%.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This study was carried out to develop a deep learning model for early identification of tomato plant damages caused by *Tuta absoluta*. Development of a sophisticated technological solution for early identification of tomato plant damages caused by *Tuta absoluta* is of high demand due to the need of rescuing the tomato productivity loss incurred by farmers. Extension officers have been using their efforts to ensure that farmers get appropriate knowledge on identification of crop pests and diseases. However, due to limited number of extension officers, sometimes farmers do not get consultation on time. To avoid this problem and reducing workload to extension officers, there is a need of integrating technological solution in pest and disease identification.

Therefore, this study focused on developing deep learning model for early identification of tomato plant damages caused by *Tuta absoluta* by using image data collected from experimental setup in field. It took advantage of recent advancement in technology particularly deep learning to develop classifiers for identifying the presence of *Tuta absoluta* in tomato leaves images.

The study findings shows that deep learning is the best approach for classifying leaf images and hence can be easily used to identify any unusual features thereby enabling to identify whether the leaves are infected or not. It performs well and enables quick and accurate way of diagnosing plant diseases as compared to the visual inspection that is done manually, needs a lot of resources to reach more farmers and is time consuming. Hence, through the use of deep learning, little human intervention will be required in diagnosing the plants as they will just need to capture images that will be imported into the model and be able to receive results of whether it is infected. This will enable the extension officers to advice farmers on appropriate measures to be taken to overcome the farm situation thereby save the economic loss that could be incurred.

The research contributes to the model to be used by farmers and extension officers in detecting invasion of *Tuta absoluta* at early stage of tomato plants growth. Farmers will be able to know the farm status with regard to invasion and take appropriate measures before they spread to large scale. It will also contribute an open source dataset to facilitate further research in *Tuta* *absoluta* identification from diseased tomato plants. Researchers can be able to utilize the dataset in studying various dynamics of *Tuta absoluta* thereby contribute to efforts for combating the invasion.

5.2 Recommendations

The study has revealed that, integration of technology in agriculture, particularly plant pests and diseases diagnostics, has potential to overcome the shortage of extension services. Farmers will be able to use these advanced technologies to monitor status of their farms and there by rescuing losses that might be attributed by lack of technology.

This study considered the use of image data collected from an in-house experiment and not the ones from outside fields, this might hinder applicability of the model in field environment where there might be more than one infection in the same plant. The amount of data collected was insufficient for training a deep learning model given limited time and financial constraint to conduct research. The model has not been deployed for testing with stakeholders in the tomato farms.

In future researchers may focus on increasing the size of our dataset and retrain the models so that they can perform well as deep learning requires a lot of data. After obtaining satisfactory performance the model can be deployed in mobile phone for use in the field. Also the researcher may need to test the model with field data (uncontrolled environment) where there may be co-infection so as to check whether it yields better results.

This research work can be expanded into a number of directions in the future. Foremost, this research recommends a further stabilization of the robustness of the proposed segmentation method, by expanding the diversity of the diabetic foot ulcer image database in terms of ulcer infection type, shape, the color composition for different skin types as well as surrounding tissues. Secondly, although the validation assessment results of the method show good agreement with clinicians' scores, more validation data is needed to further evaluate our method. Finally, the use of neural networks to extract more features and increase the accuracy of the segmentation method is also recommended.

REFERENCES

Arah, I. K. (2015). *An overview of post-harvest challenges facing tomato production in Africa Africa : Diversity and Development An overview of post-harvest challenges facing tomato production in Africa*. (November).

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligence*, *31*(4), 299–315. https://doi.org/10.1080/08839514.2017.1315516

Chidege, M., Al-zaidi, S., Hassan, N., Julie, A., Kaaya, E., & Mrogoro, S. (2016). First record of tomato leaf miner Tuta absoluta (Meyrick) (Lepidoptera: Gelechiidae) in Tanzania. *Agriculture & Food Security*, *5*(1), 1-17. https://doi.org/10.1186/s40066-016-0066-4

Cuthbertson, A. G. S., Mathers, J. J., Blackburn, L. F., Korycinska, A., Luo, W., Jacobson, R. J., & Northing, P. (2013). Population Development of Tuta absoluta (Meyrick) (Lepidoptera: Gelechiidae) under Simulated UK Glasshouse Conditions. *Insects*, *4*(2), 185–197. https://doi.org/10.3390/insects4020185

Dechant, C., Gore, M. A., Dechant, C., Wiesner-hanks, T., Chen, S., Stewart, E. L., … Nelson, R. J. (2017). *Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning*. (October). https://doi.org/10.1094/PHYTO-11-16-0417-R

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, *145*(September 2017), 311–318. https://doi.org/10.1016/j.compag.2018.01.009

Frusciante, L., Carli, P., Ercolano, M. R., Pernice, R., Di Matteo, A., Fogliano, V., & Pellegrini, N. (2007). Antioxidant nutritional quality of tomato. *Molecular Nutrition and Food Research*, *51*(5), 609–617. https://doi.org/10.1002/mnfr.200600158

Guimapi, R. Y. A., Mohamed, S. A., Okeyo, G. O., Ndjomatchoua, F. T., Ekesi, S., & Tonnang, H. E. Z. (2016). Modeling the risk of invasion and spread of Tuta absoluta in Africa. *Ecological Complexity*, *28*, 77–93. https://doi.org/10.1016/j.ecocom.2016.08.001

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–

778.

Jeong, H. J., Park, K. S., & Ha, Y. G. (2018). Image Preprocessing for Efficient Training of YOLO Deep Learning Networks. *Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing, BigComp 2018*, 635–637. https://doi.org/10.1109/BigComp.2018.00113

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444. https://doi.org/10.1038/nature14539

Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*, *71*, 1–13. https://doi.org/10.1016/j.patcog.2017.05.015

Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, *10*(1), 1-11.

Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks R. *Neurocomputing*, *267*, 378–384. https://doi.org/10.1016/j.neucom.2017.06.023

Lyatuu, E. T., Nie, F., & Fang, C. (2015). *The role of agriculture in the economic growth and poverty reduction in tanzania*.

Maginga, T. J., Nordey, T., & Ally, M. (2018). Extension System for Improving the Management of Vegetable Cropping Systems. *Journal of Information Systems Engineering & Management*, *3*(4), 1-10. https://doi.org/10.20897/jisem/3940

Mehdipour, G. M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing*, *235*(August 2016), 228–235. https://doi.org/10.1016/j.neucom.2017.01.018

Mutayoba, V., & Ngaruko, D. (2018). Assessing Tomato Farming and Marketing Among Smallholders in High Potential Agricultural Areas of Tanzania. *International Journal of Economics, Commerce and Management*, *VI*(8), 577–590.

NBS. (2017). Annual Agriculture Sample Survey Crop and Livestock Report. *The United Republic of Tanzania (URT),* 1–181. Retrieved from https://www.nbs.go.tz/nbs/takwimu/Agriculture/2016-17_AASS Report _Final.pdf

Ouppaphan, P. (2017). Corn Disease Identification from Leaf Images Using Convolutional Neural Networks. *2017 21st International Computer Science and Engineering Conference (ICSEC)*, 1–5. https://doi.org/10.1109/ICSEC.2017.8443919

Park, D. S. (2017). *Characteristics of Tomato Plant Diseases*. (October 2016).

Paul, F., & Lema, A. (2018). The Dynamic Synergies between Agricultural Financing and Economic Growth of Tanzania. *African Journal of Economic Review*, *6*(2), 46–60.

Perez, L., & Wang, J. (2017). *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. Retrieved from http://arxiv.org/abs/1712.04621

Raj, S. (2013). e-Agriculture prototype for knowledge facilitation among tribal farmers of North-East India: innovations, impact and lessons. *The Journal of Agricultural Education and Extension*, *19*(2), 113–131.

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, *8*(October), 1–7. https://doi.org/10.3389/fpls.2017.01852

Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, *133*, 1040–1047. https://doi.org/10.1016/j.procs.2018.07.070

Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., & De Geus, P. (2017). Malicious software classification using transfer learning of ResNet-50 deep neural network. *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, *2017-Decem*, 1011–1014. https://doi.org/10.1109/ICMLA.2017.00-19

Salas Gervassio, N. G., Aquino, D., Vallina, C., Biondi, A., & Luna, M. G. (2019). A re-examination of Tuta absoluta parasitoids in South America for optimized biological control. *Journal of Pest Science*, *92*(4), 1343–1357. https://doi.org/10.1007/s10340-018-01078-1

Schäfer, L., & Herz, A. (2020). Suitability of european trichogramma species as biocontrol agents against the tomato leaf miner tuta absoluta. *Insects*, *11*(6), 1–18. https://doi.org/10.3390/insects11060357

Schreinemachers, P., Simmons, E. B., & Wopereis, M. C. S. (2018). Tapping the economic and nutritional power of vegetables. *Global Food Security*, *16*(September), 36–45. https://doi.org/10.1016/j.gfs.2017.09.005

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Singh, A., Ganapathysubramanian, B., Singh, A. K., & Sarkar, S. (2016). Machine Learning for High-Throughput Stress Phenotyping in Plants. *Trends in Plant Science*, *21*(2), 110–124. https://doi.org/10.1016/j.tplants.2015.10.015

Tonnang, H. E. Z., Mohamed, S. F., Khamis, F., & Ekesi, S. (2015). Identification and risk assessment for worldwide invasion and spread of tuta absoluta with a focus on Sub-Saharan Africa: Implications for phytosanitary measures and management. *PLoS ONE*, *10*(8), 1–19. https://doi.org/10.1371/journal.pone.0135283

Wang, G., Sun, Y., & Wang, J. (2017). Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. *Computational Intelligence and Neuroscience*, *2017*. https://doi.org/10.1155/2017/2917536

Yann, L., & Yoshua, B. (1995). *Convolutional Networks for Images, Speech, and Time-Series*. *4*(April 2016), 2571–2575. https://doi.org/10.1109/IJCNN.2004.1381049

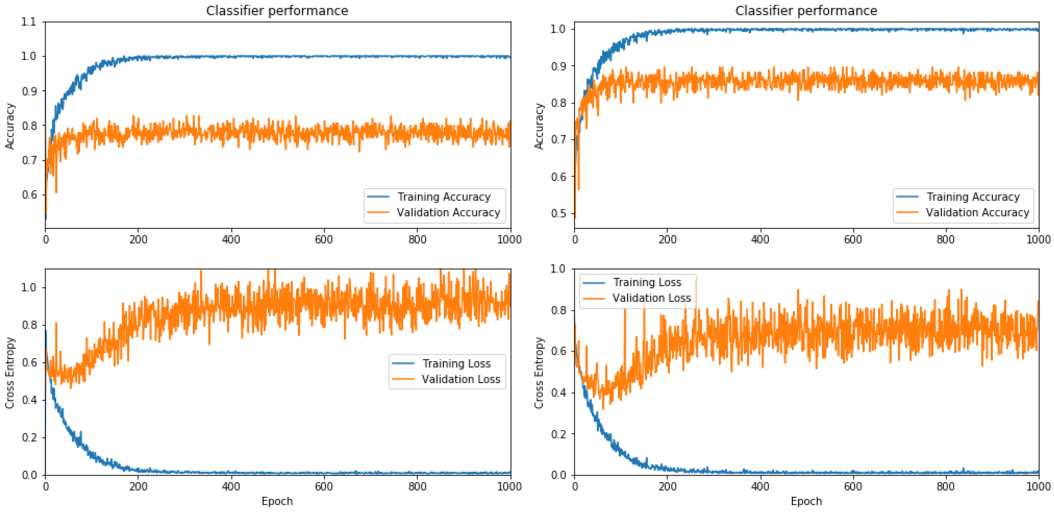
Zekeya, N., Chacha, M., Ndakidemi, P., Materu, C., Chidege, M., & Mbega, E. (2017). Tomato Leafminer (Tuta absoluta Meyrick 1917): A Threat to Tomato Production in Africa. *Journal of Agriculture and Ecology Research International*, *10*(1), 1–10. https://doi.org/10.9734/JAERI/2017/28886

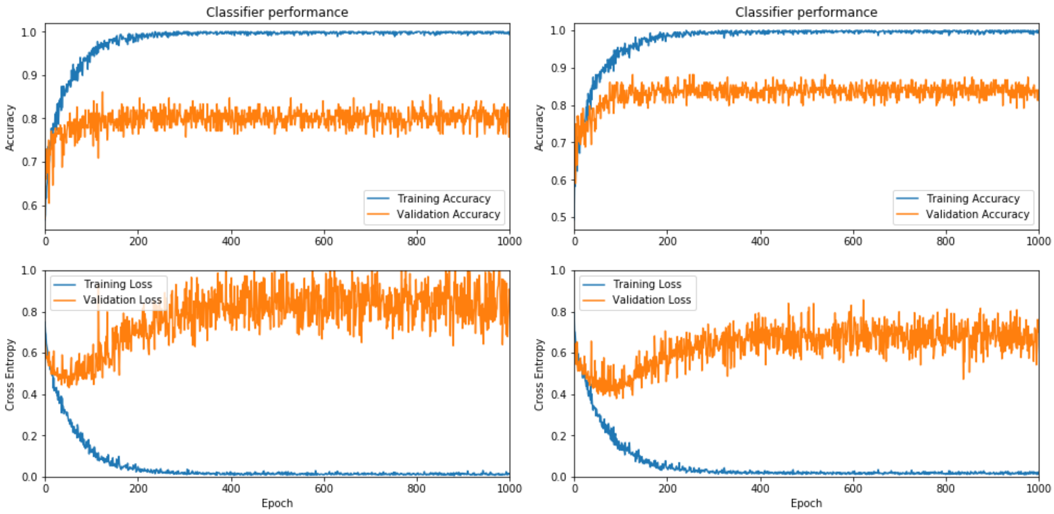
Zekeya, N., Ndakidemi, P. A., Chacha, M., & Mbega, E. (2017). *Tomato Leafminer , Tuta absoluta (Meyrick 1917), an emerging agricultural pest in Sub-Saharan Africa : Current and Prospective Management Strategies*. *12*(6), 389–396. https://doi.org/10.5897/AJAR2016.11515

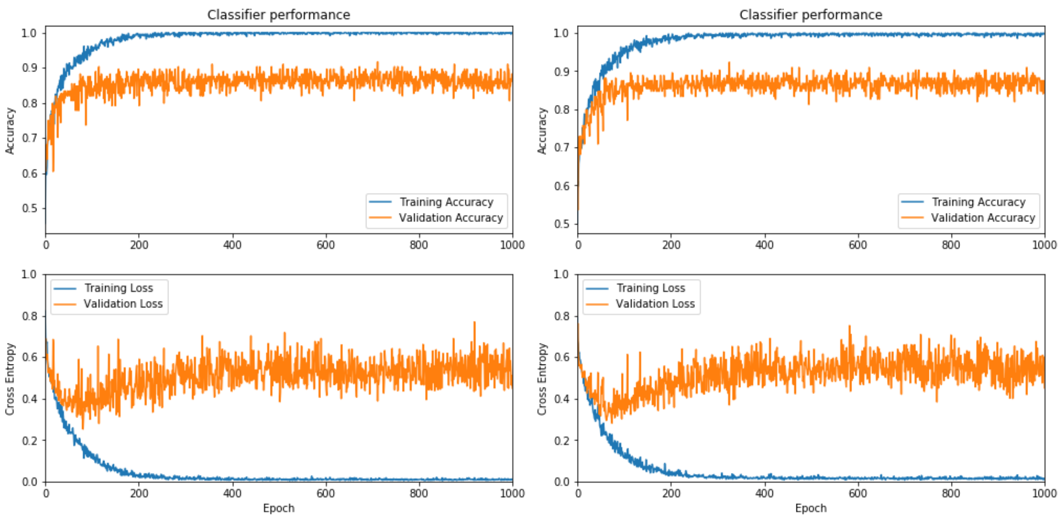
Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can Deep Learning Identify Tomato Leaf Disease? *Advances in Multimedia*, *2018*.

APPENDICES

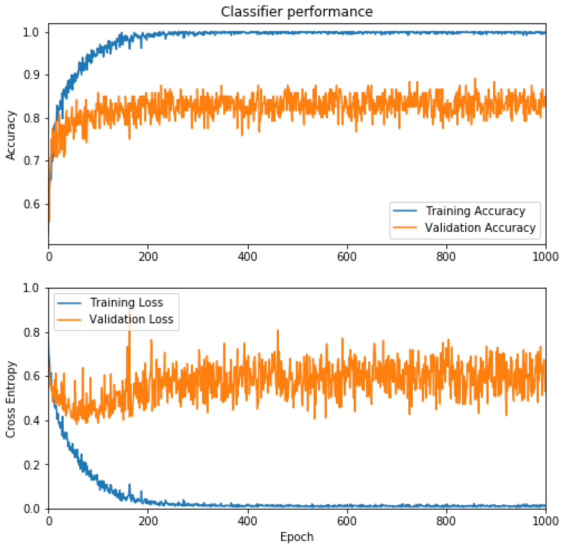
Appendix 1: Performance Graphs of VGG16 on 75:25

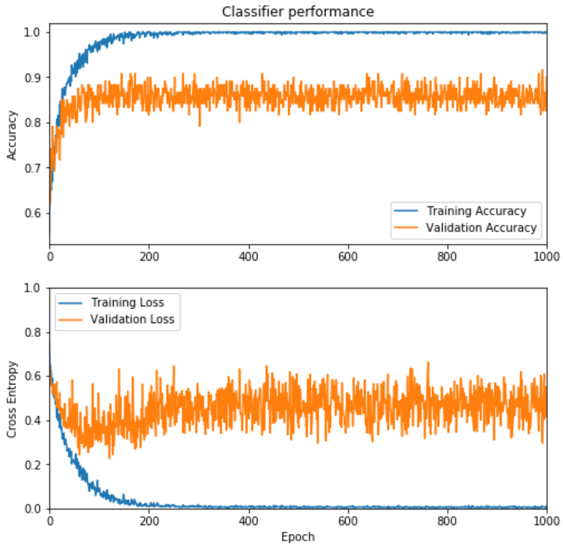


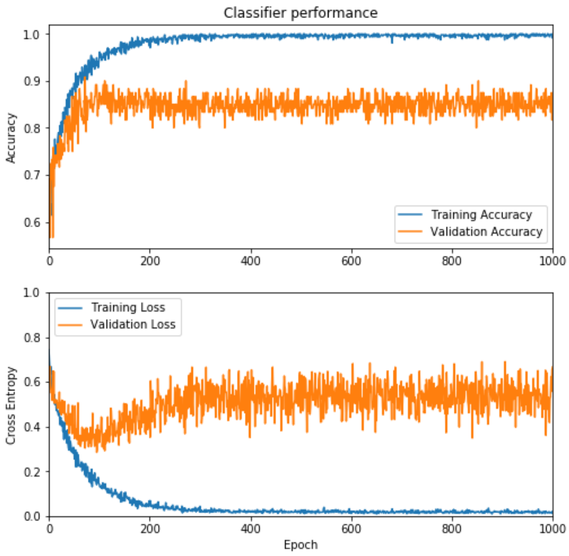


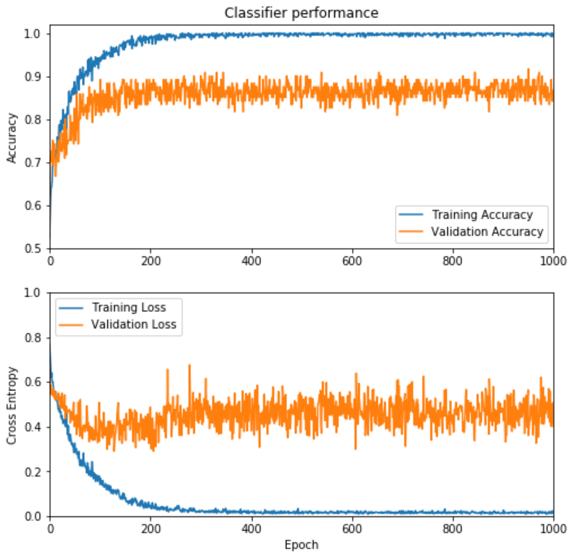


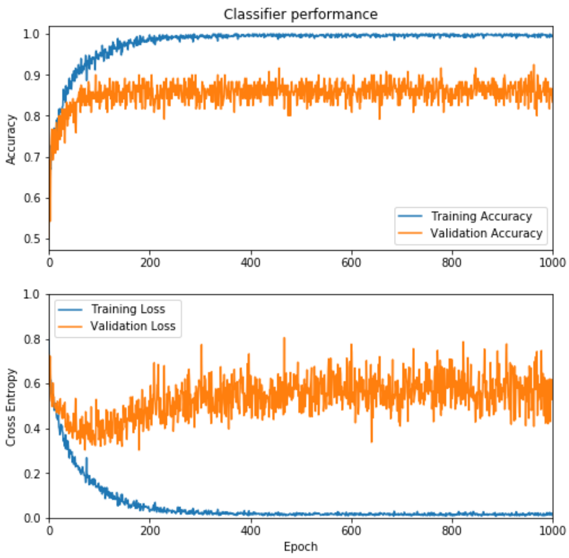
Appendix 2: Performance Graphs of VGG16 on 80:20

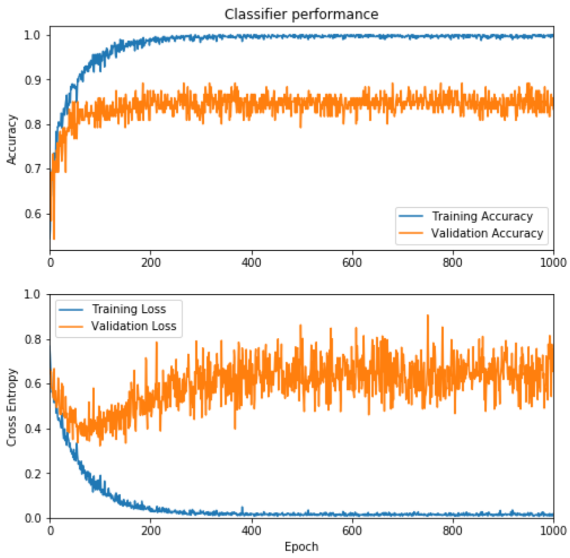




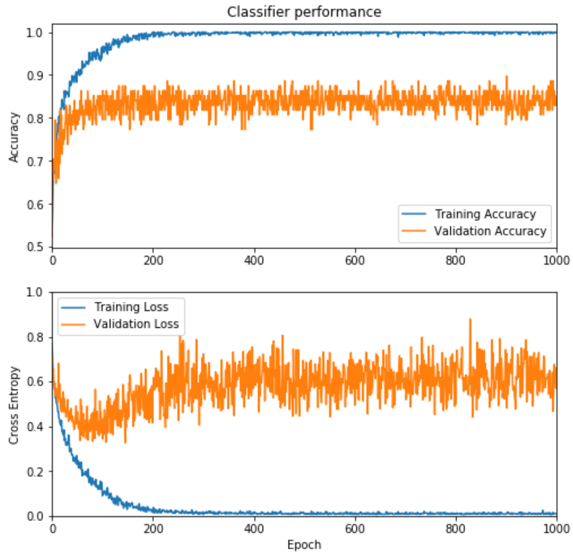


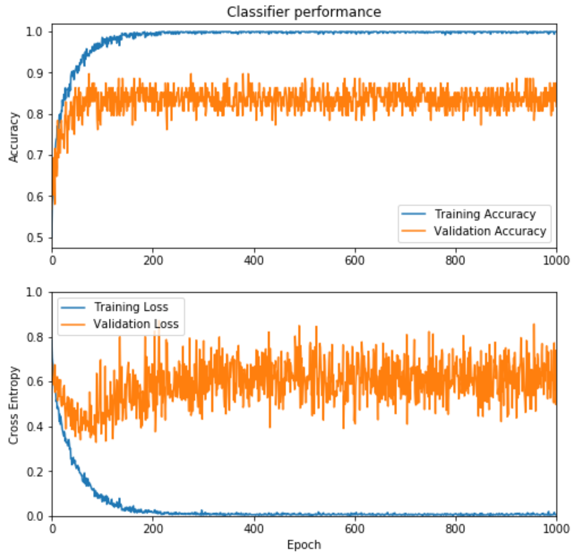


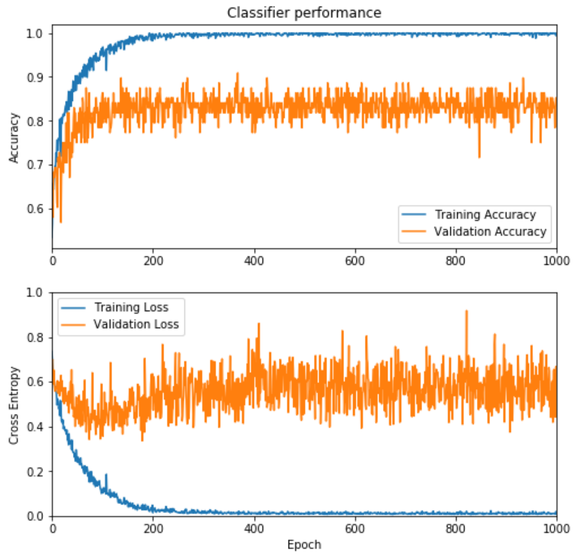


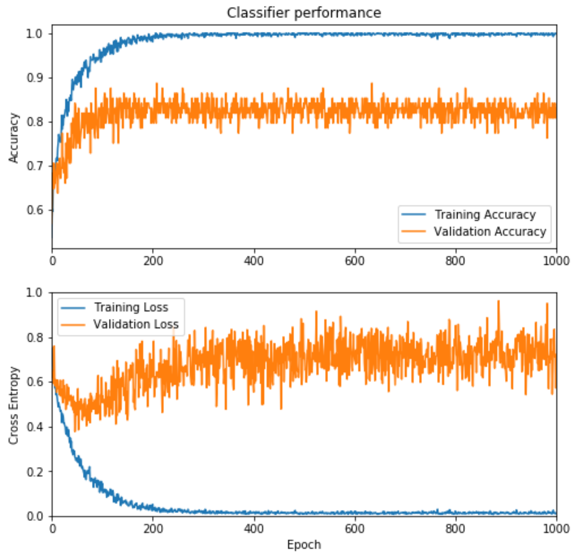


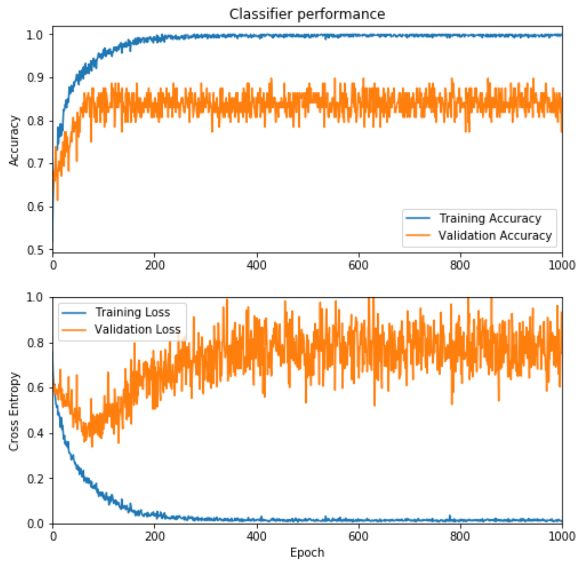
Appendix 3: Performance Graphs of VGG16 on 85:15

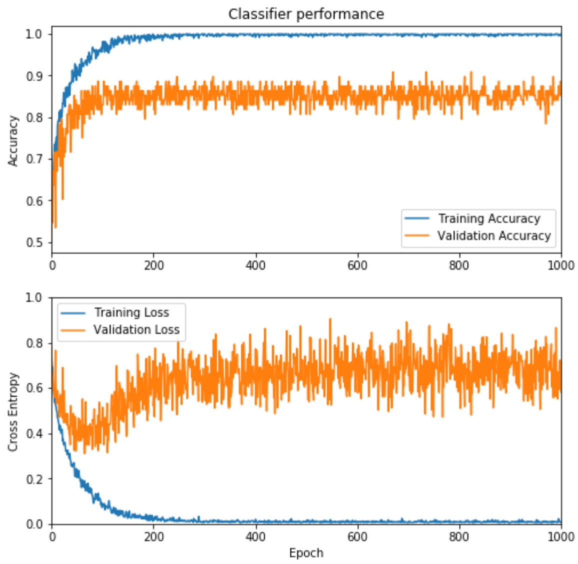




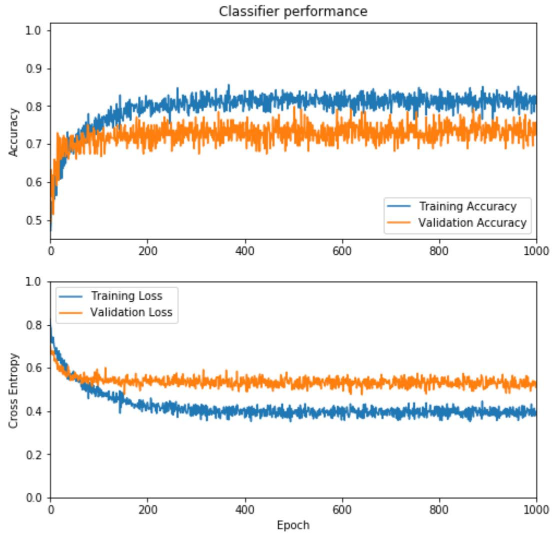


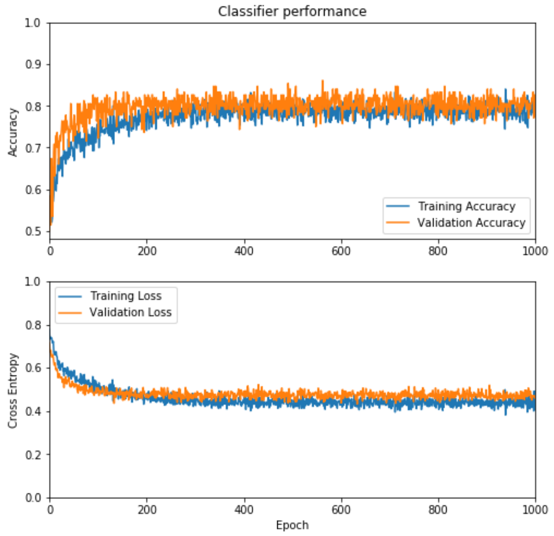


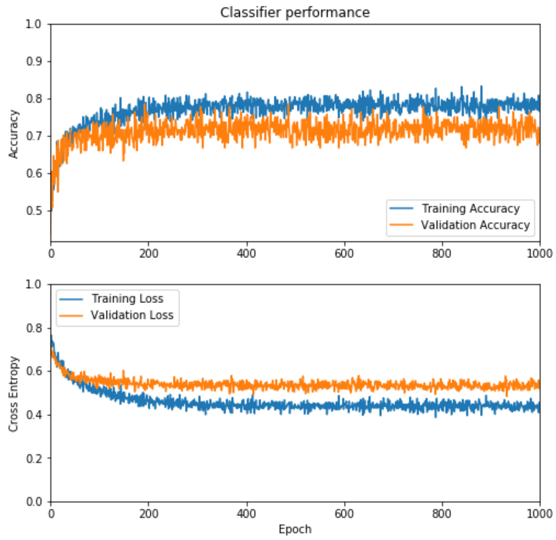


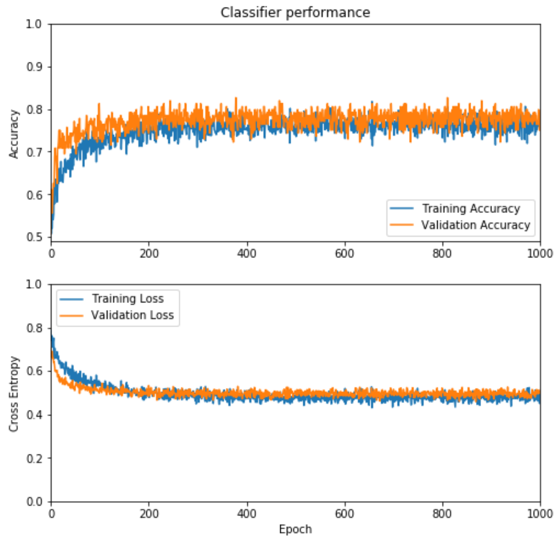


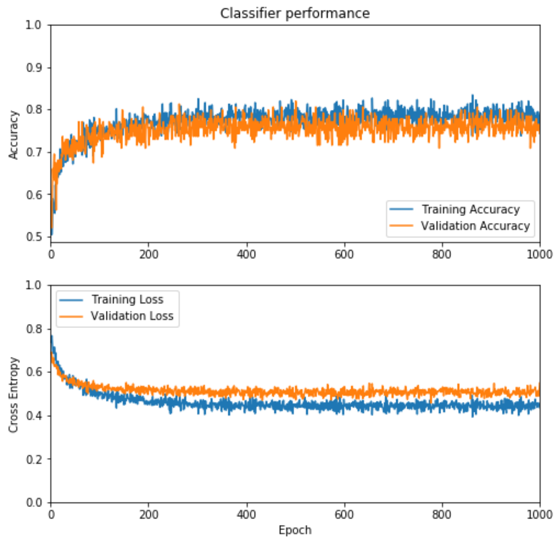
Appendix 4: Performance Graphs of VGG19 on 75:25

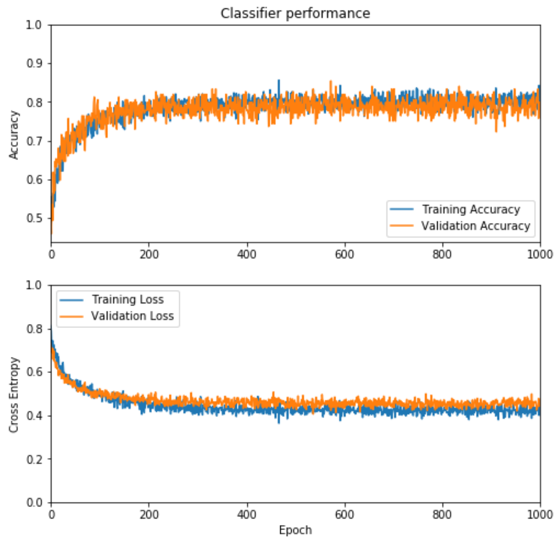




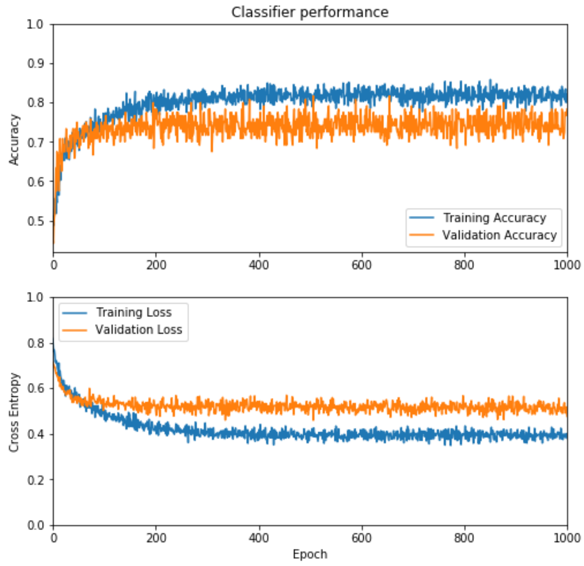


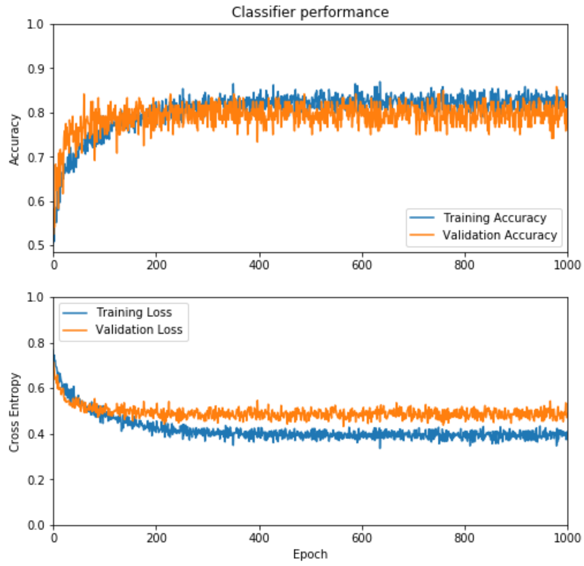


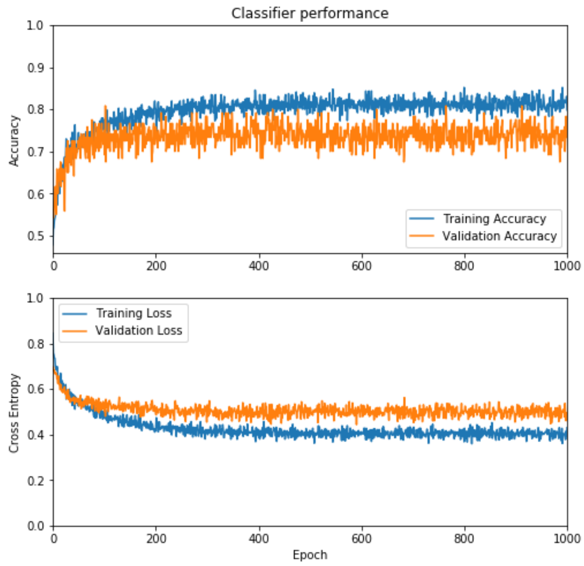


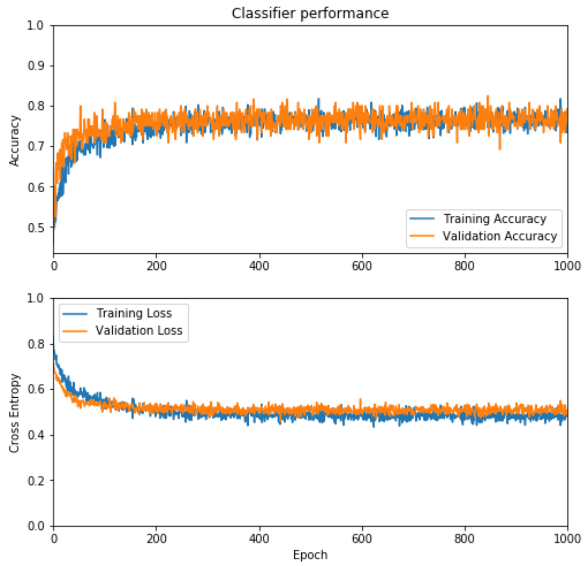


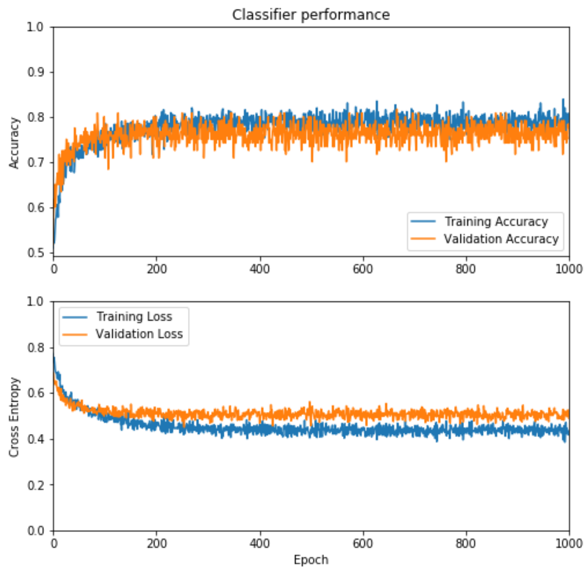
Appendix 5: Performance Graphs of VGG19 on 80:20

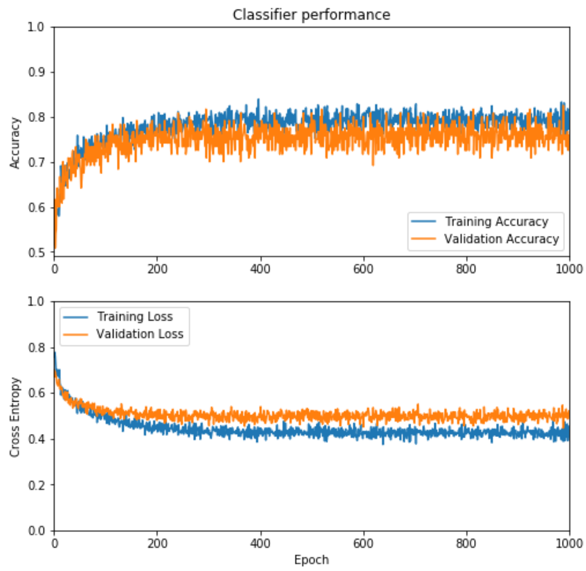




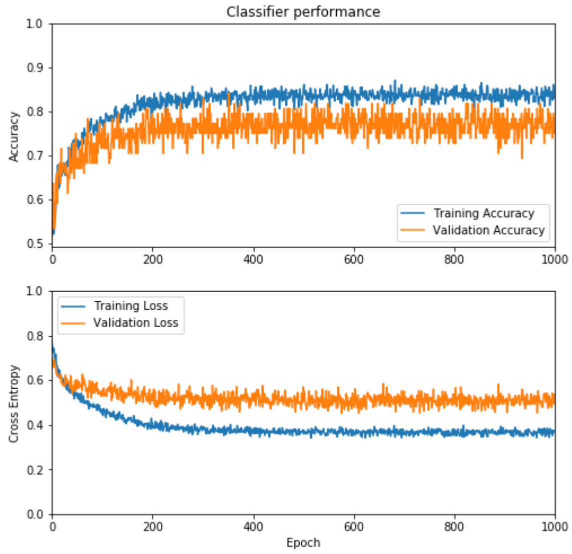


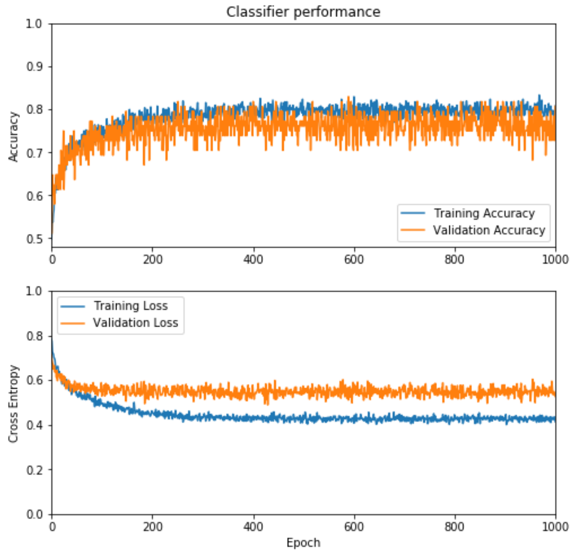


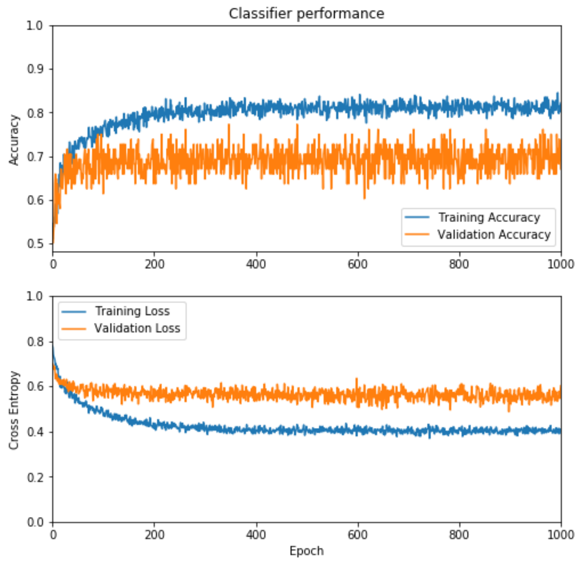


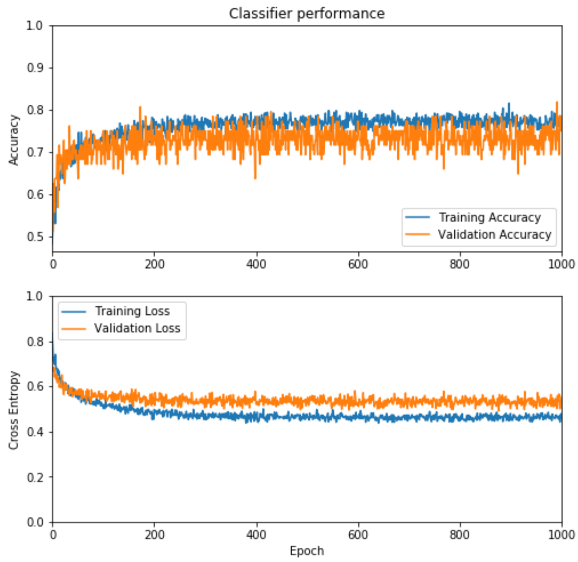


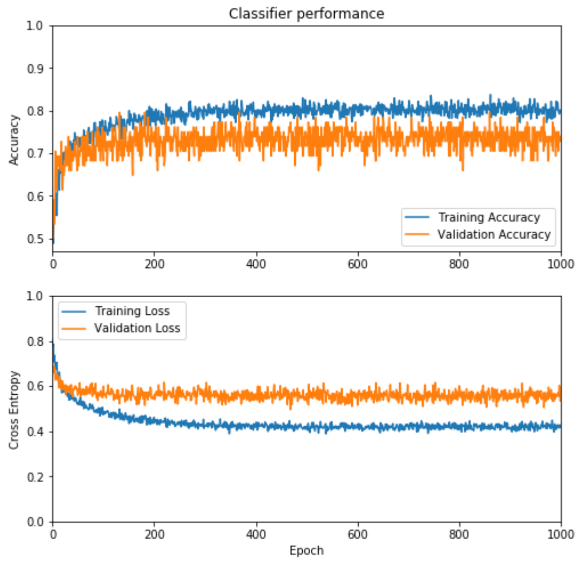
Appendix 6: Performance Graphs of VGG19 on 85:15

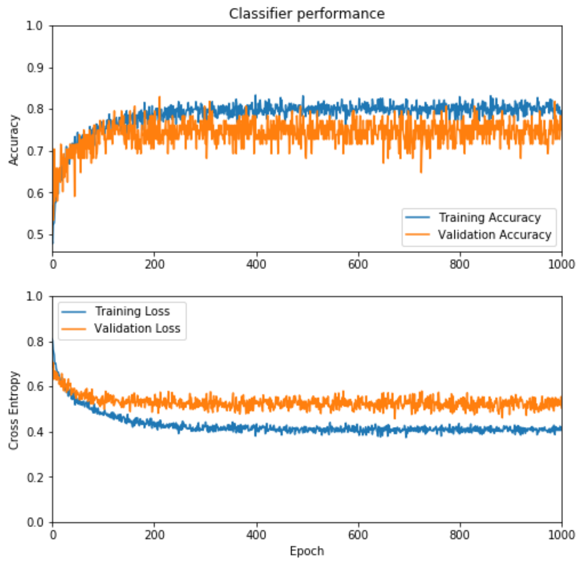




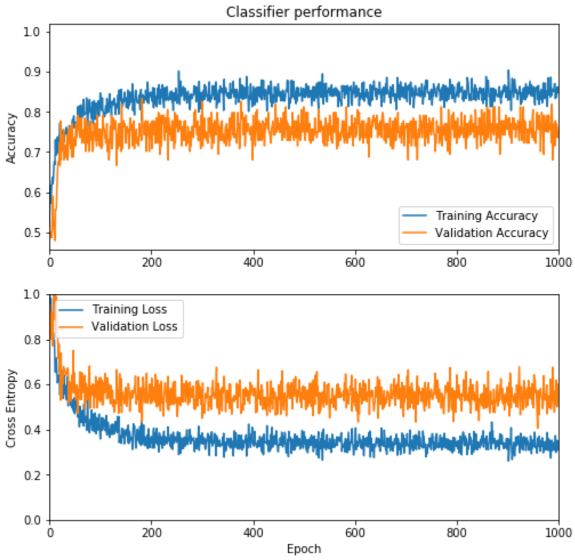


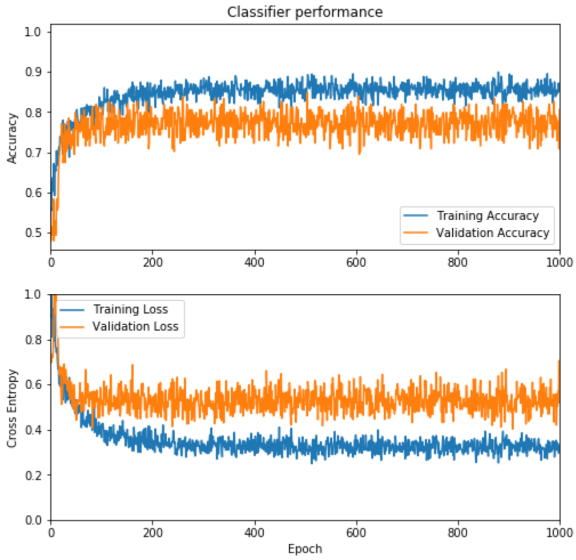


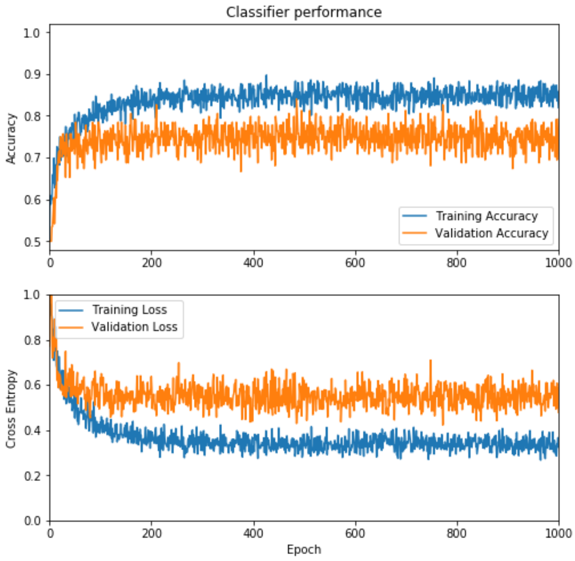


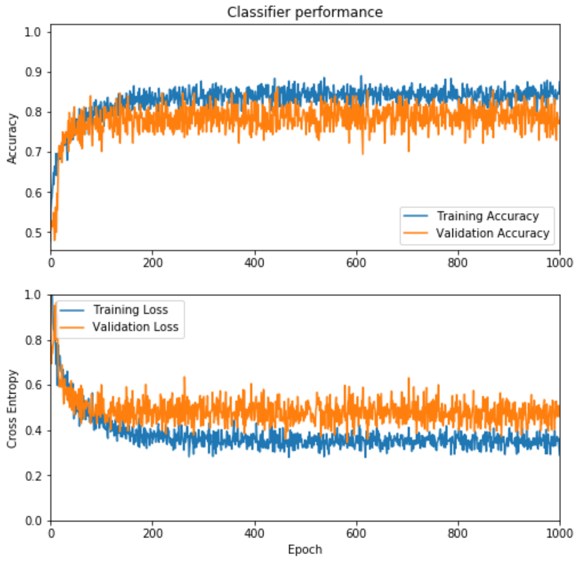


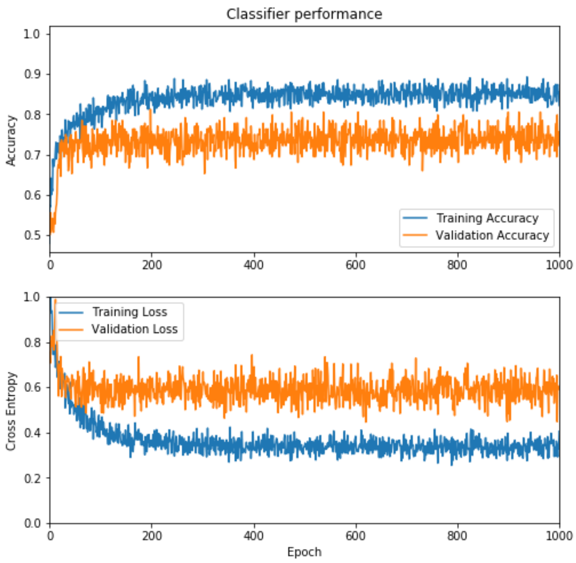
Appendix 7: Performance Graphs of ResNet50 on 75:25

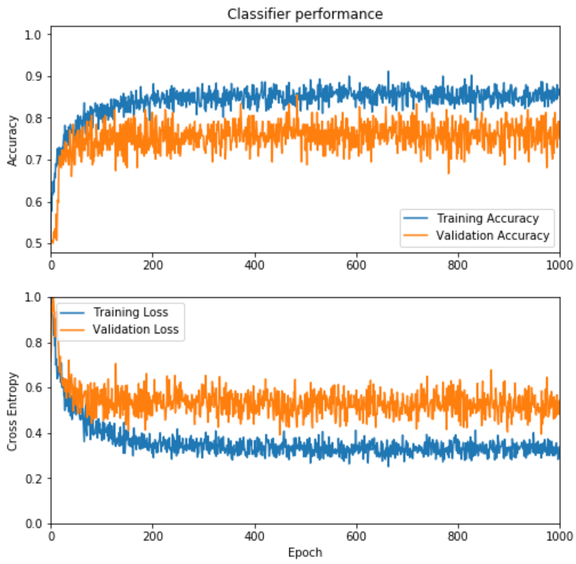




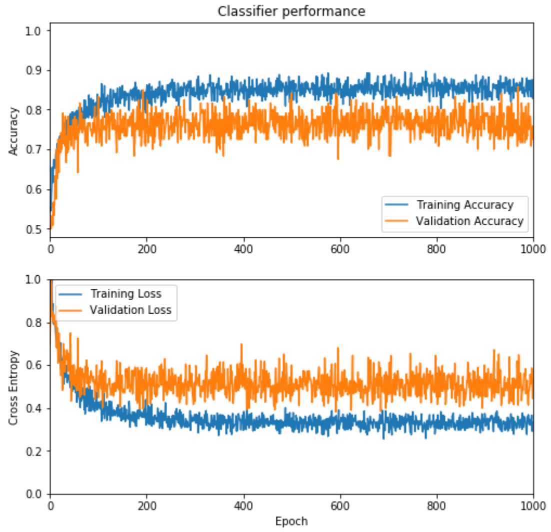


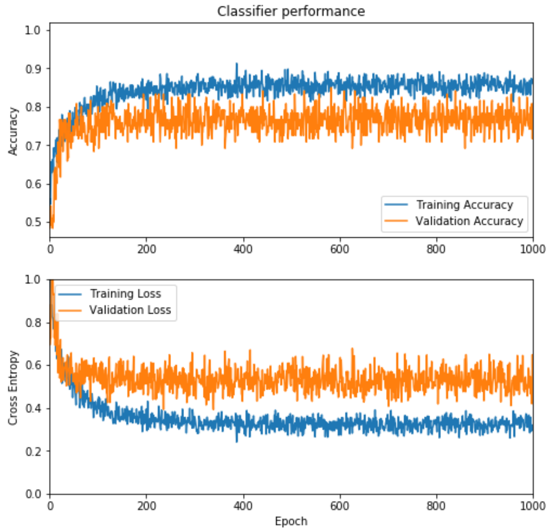


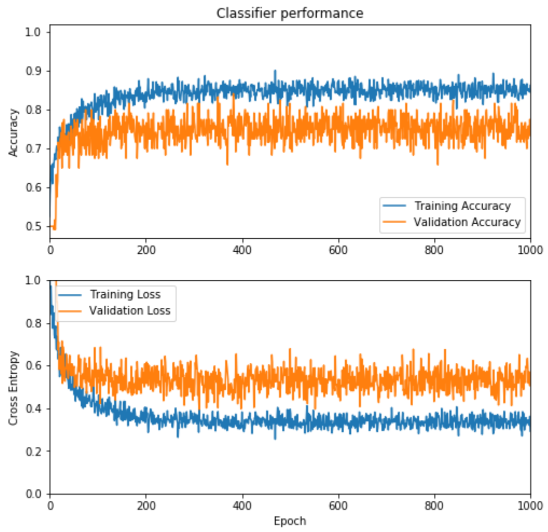


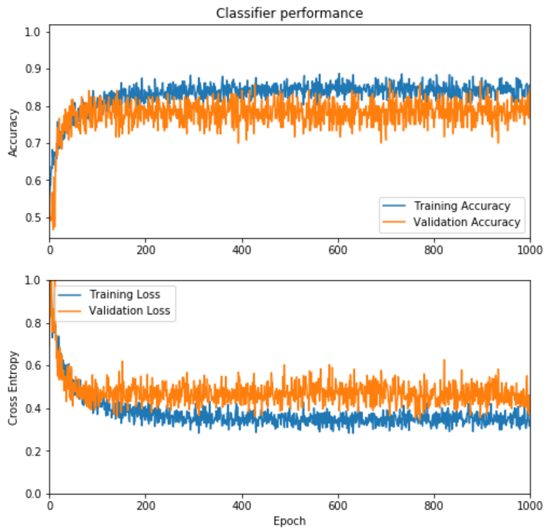


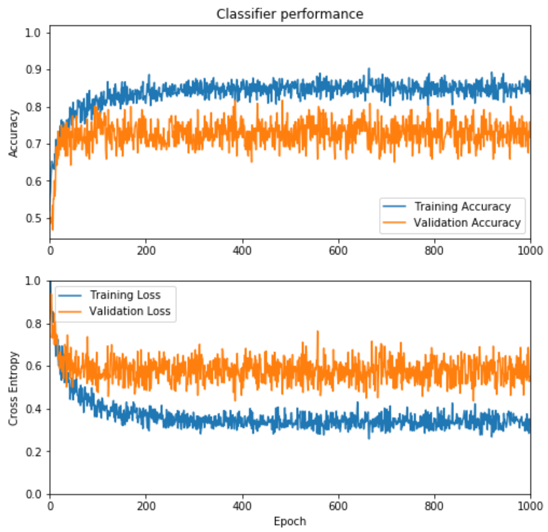
Appendix 8: Performance Graphs of ResNet50 on 80:20

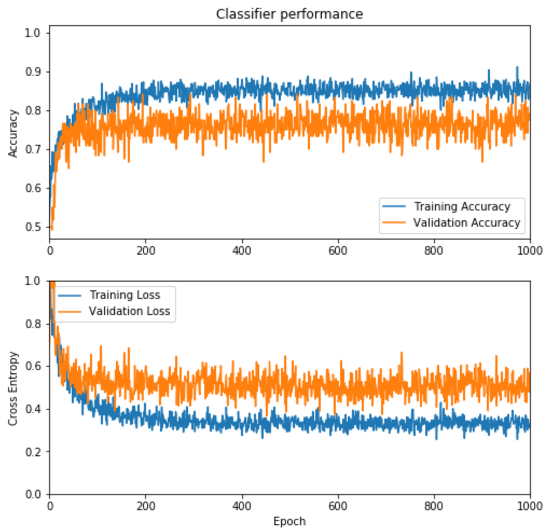




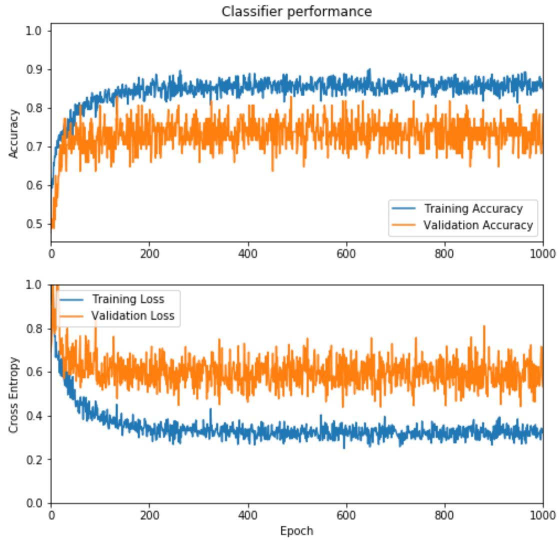


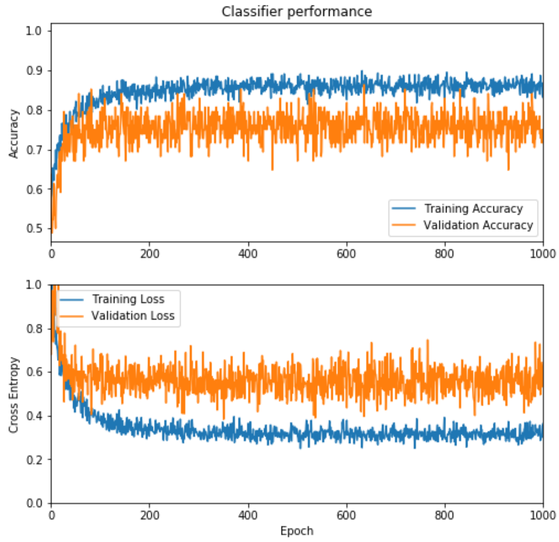


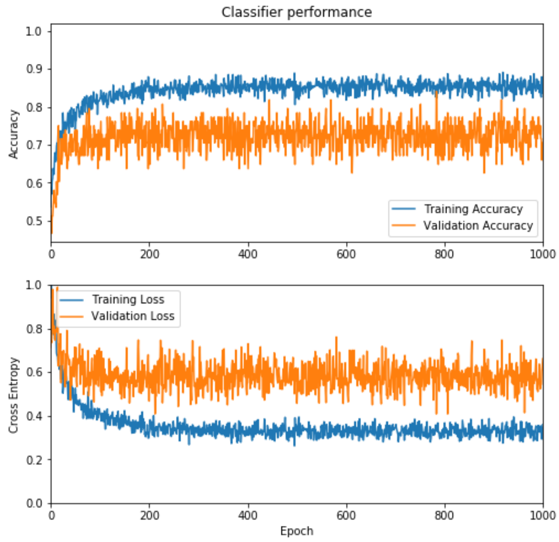


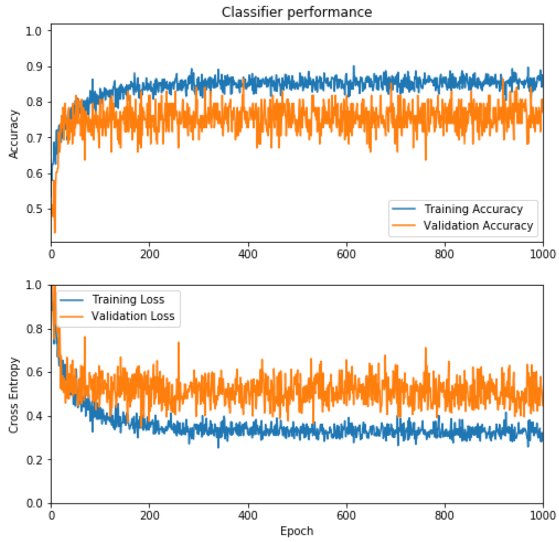


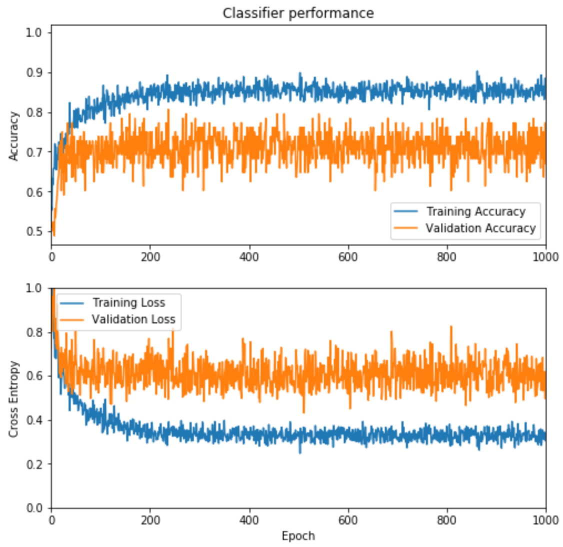
Appendix 9: Performance Graphs of ResNet50 on 85:15

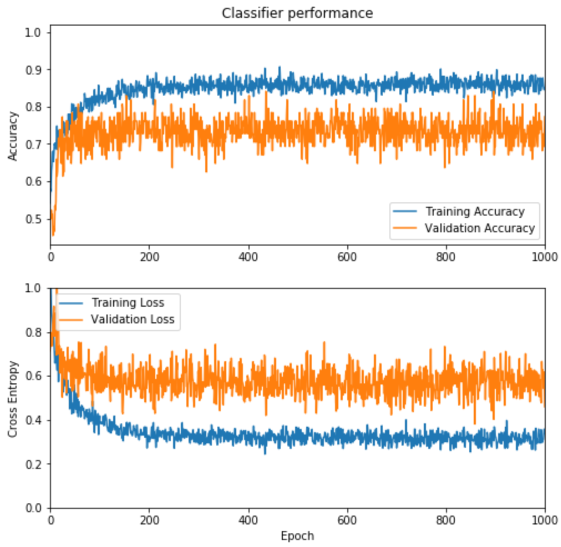




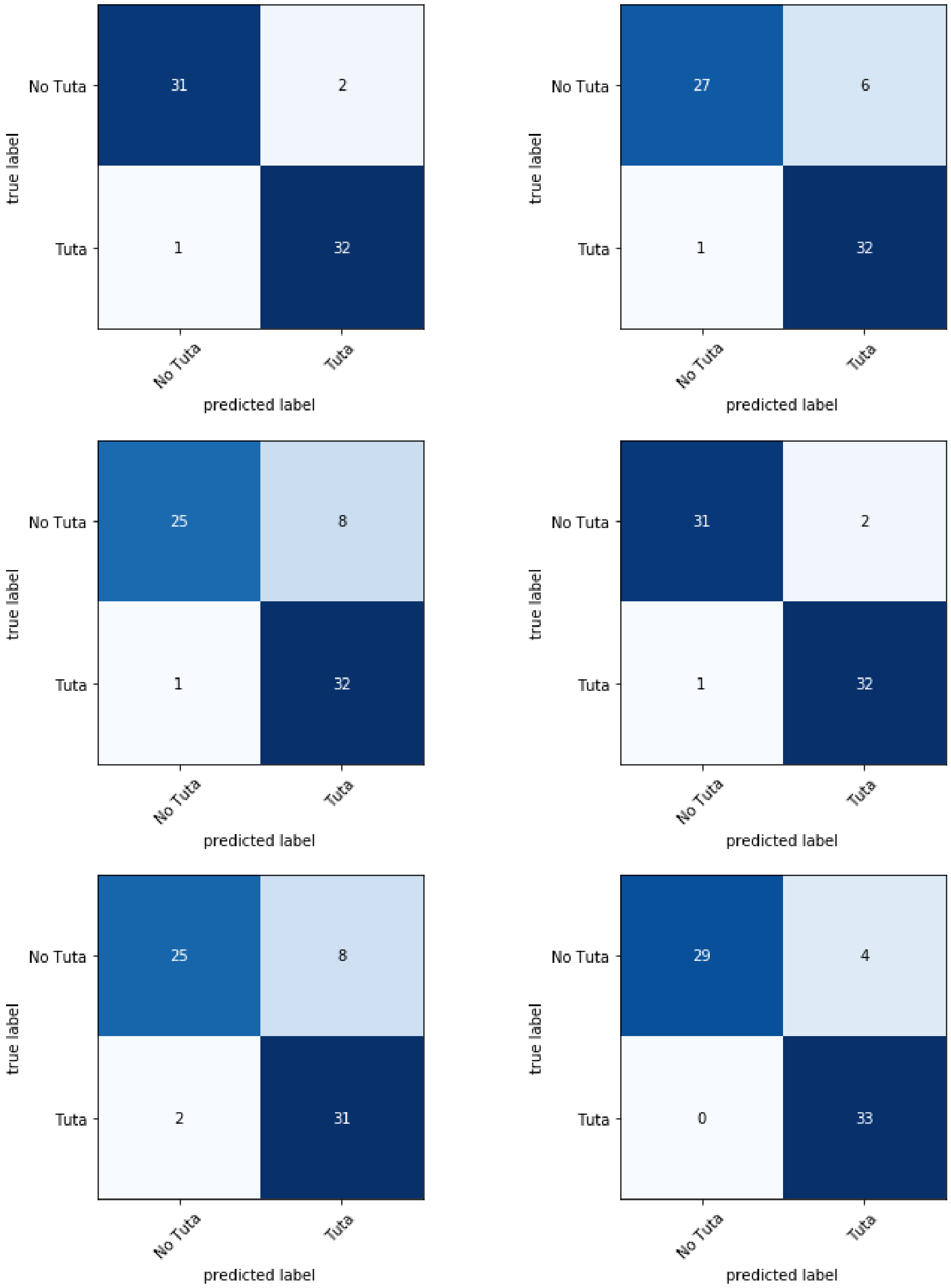




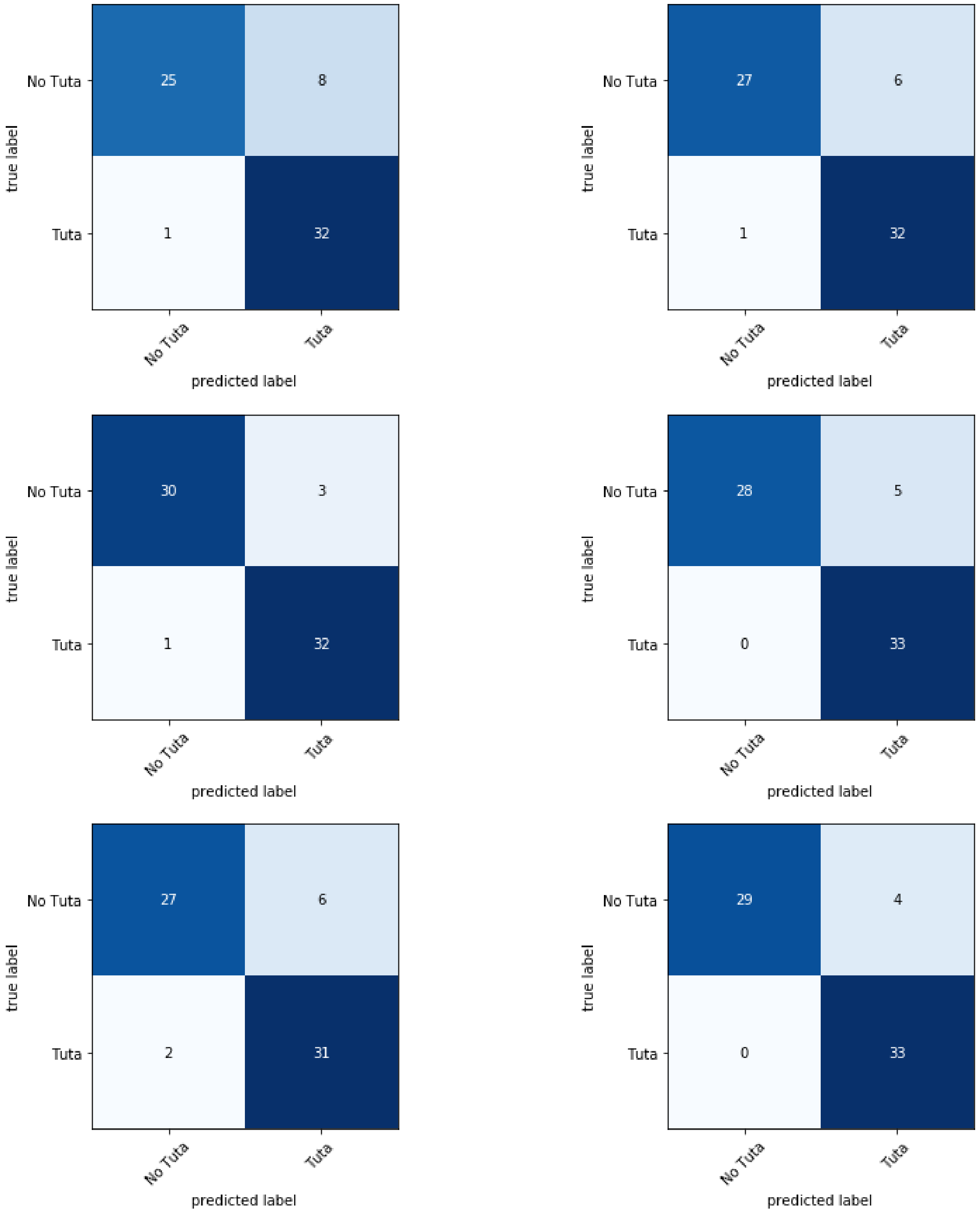




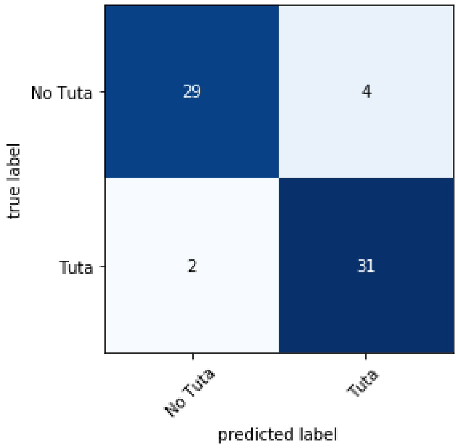
Appendix 10: Confusion matrices for VGG16 on 75:25 dataset

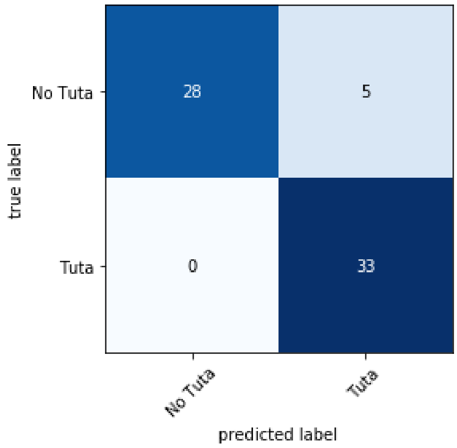


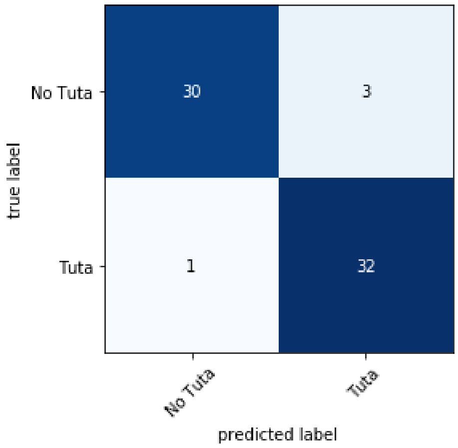
Appendix 11: Confusion matrices for VGG16 on 80:20 dataset

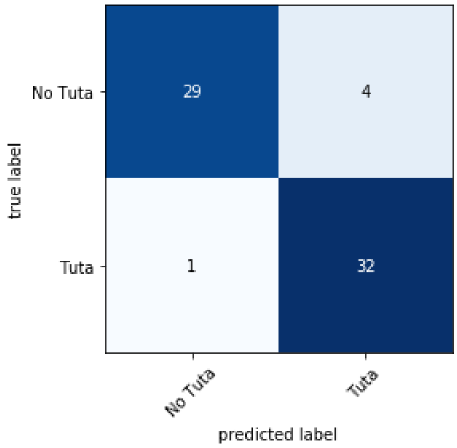


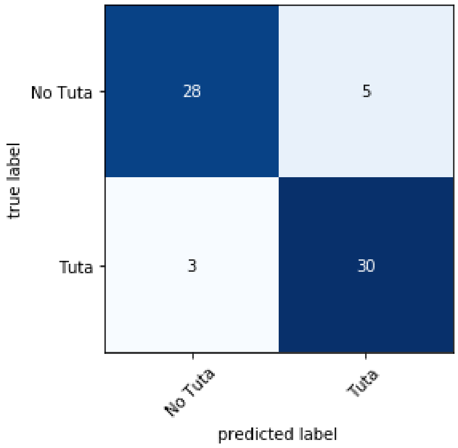
Appendix 12: Confusion matrices for VGG16 on 85:15 dataset

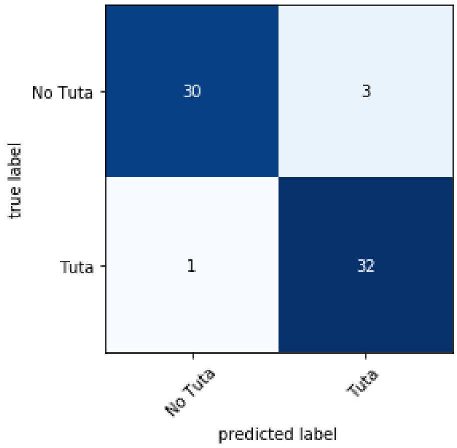




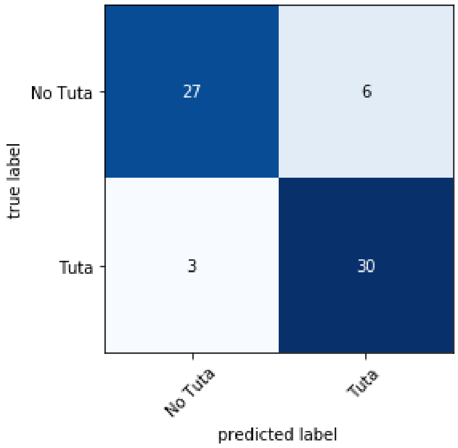


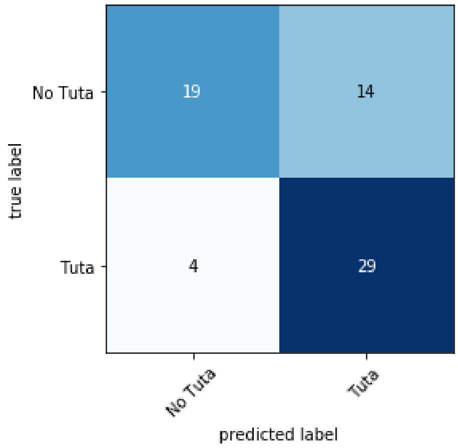


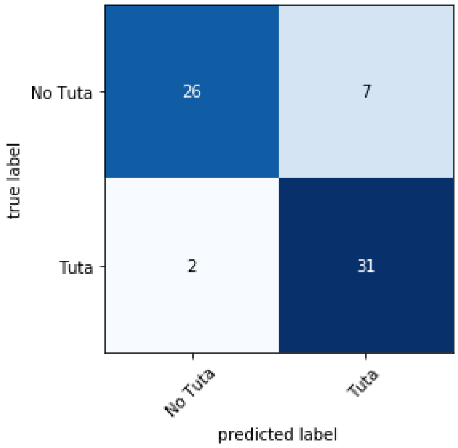


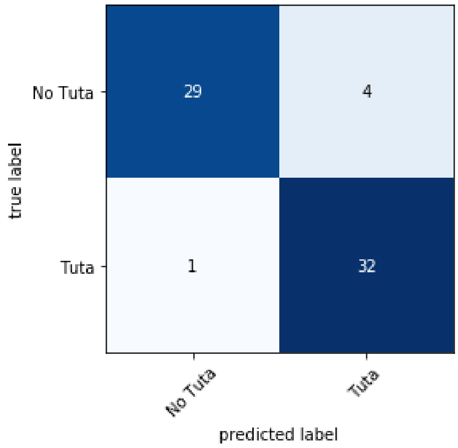


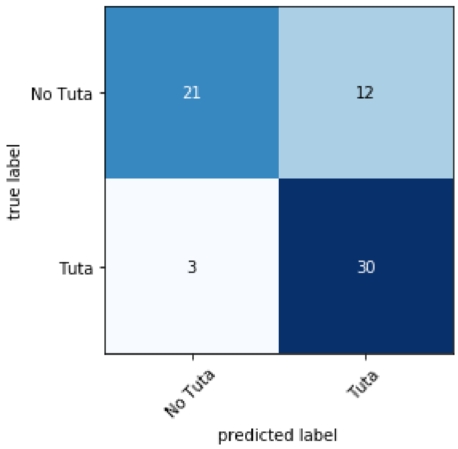
Appendix 13: Confusion matrices for VGG19 on 75:25 dataset

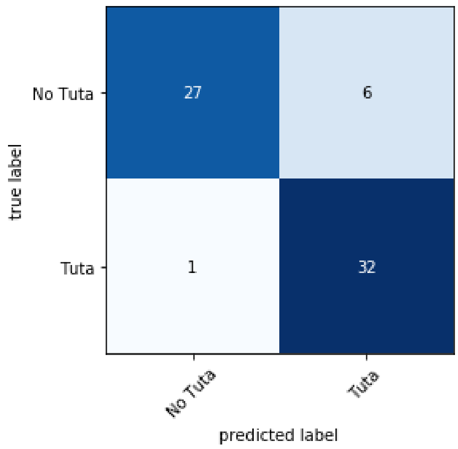




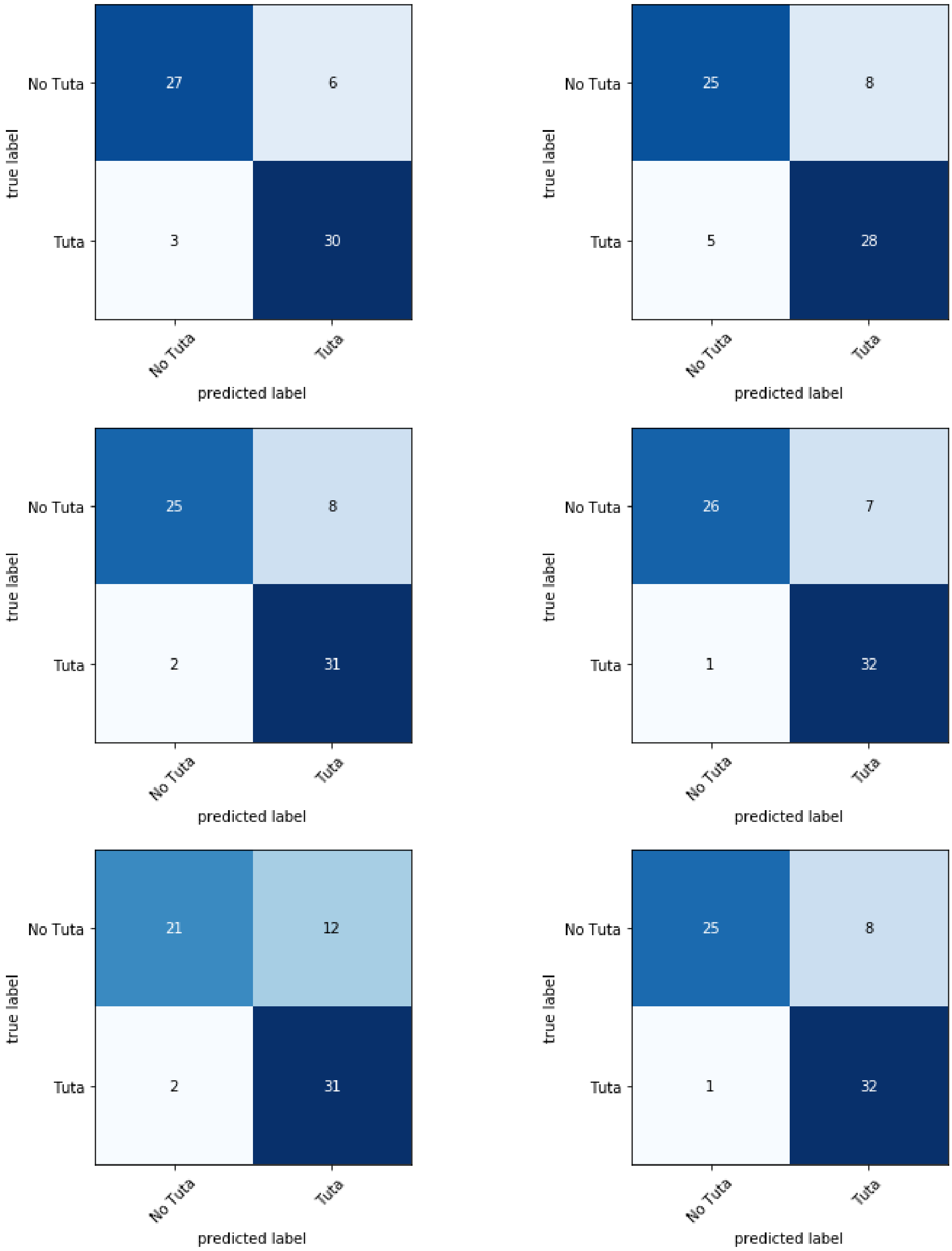




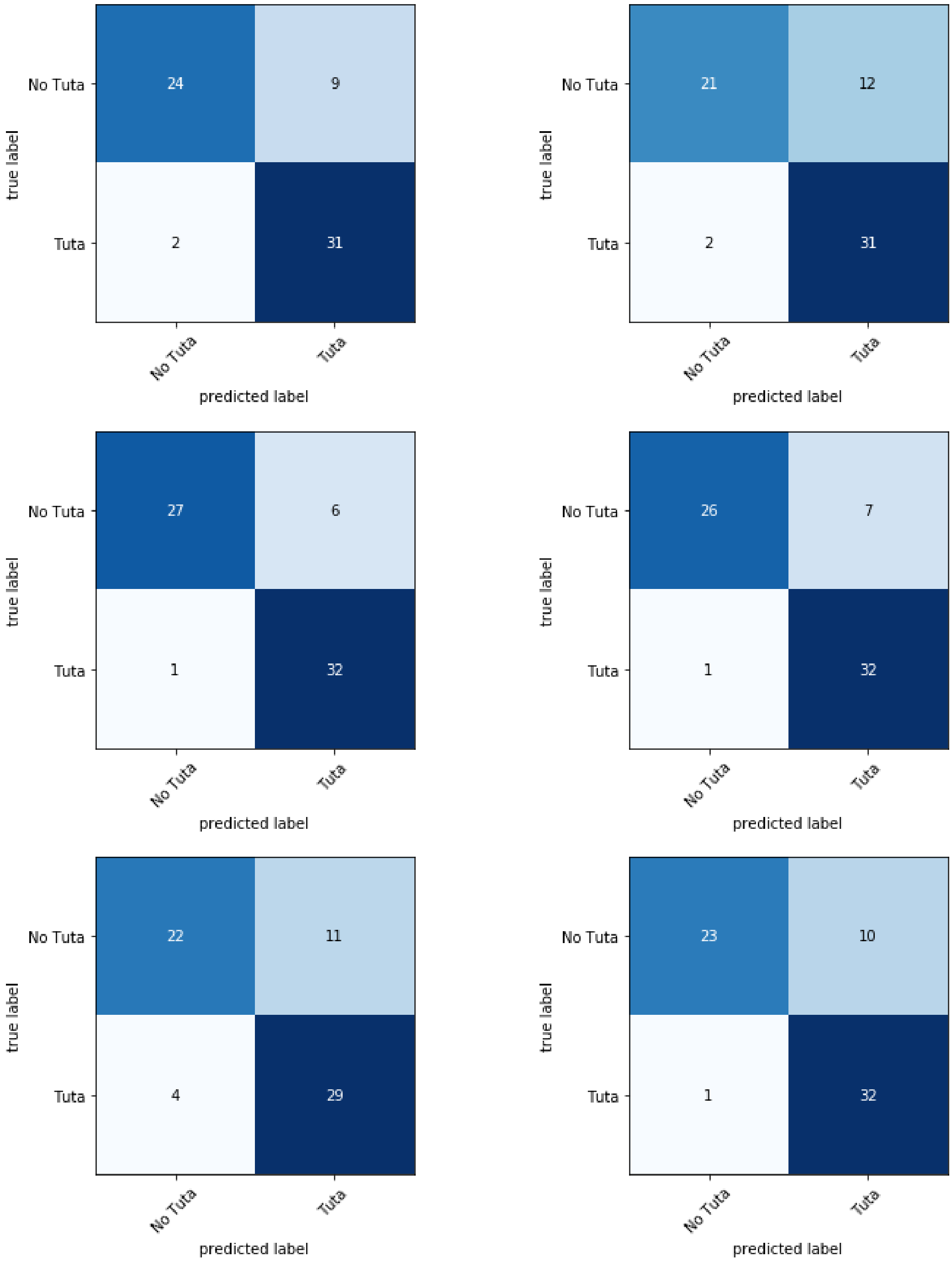




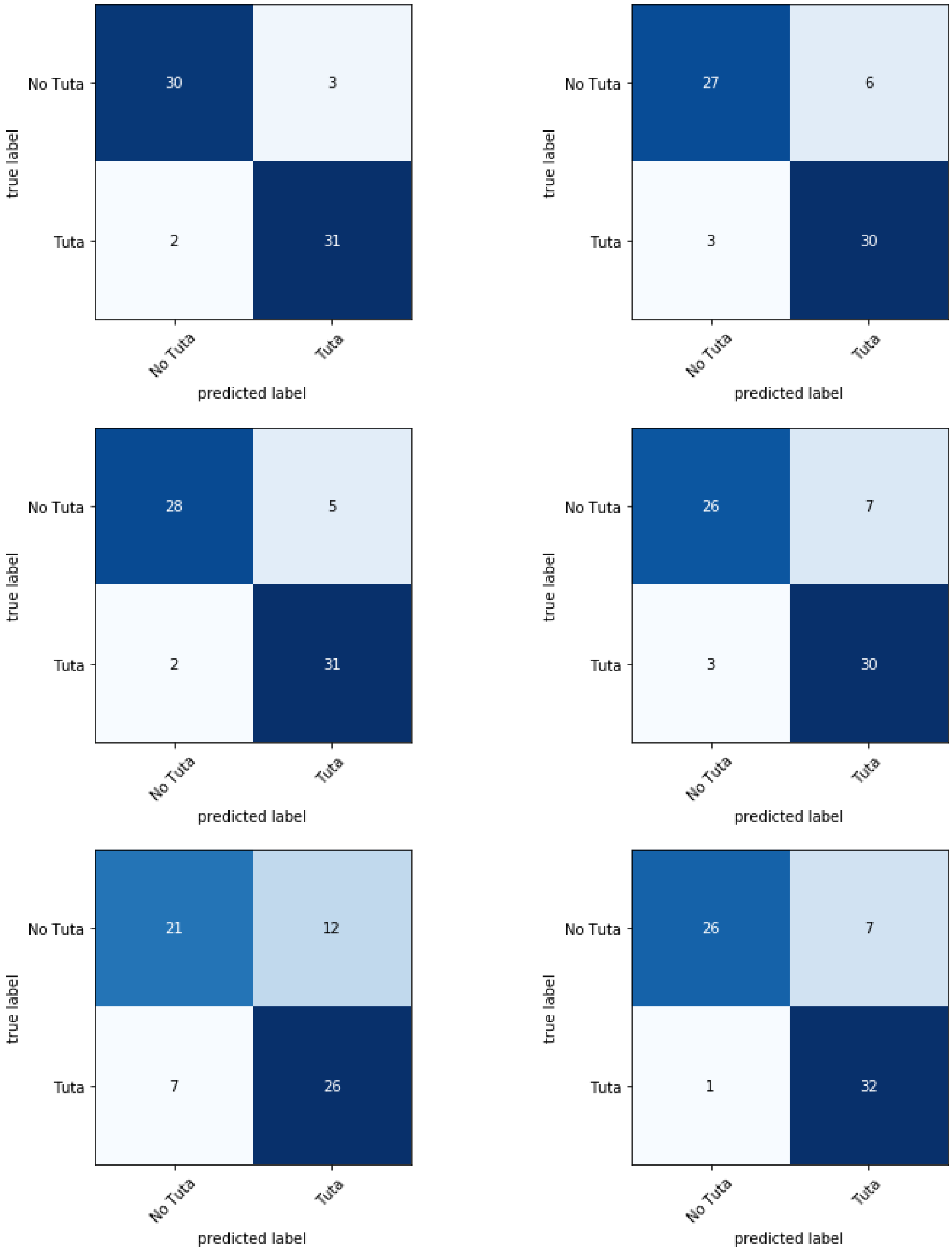
Appendix 14: Confusion matrices for VGG19 on 80:20 dataset



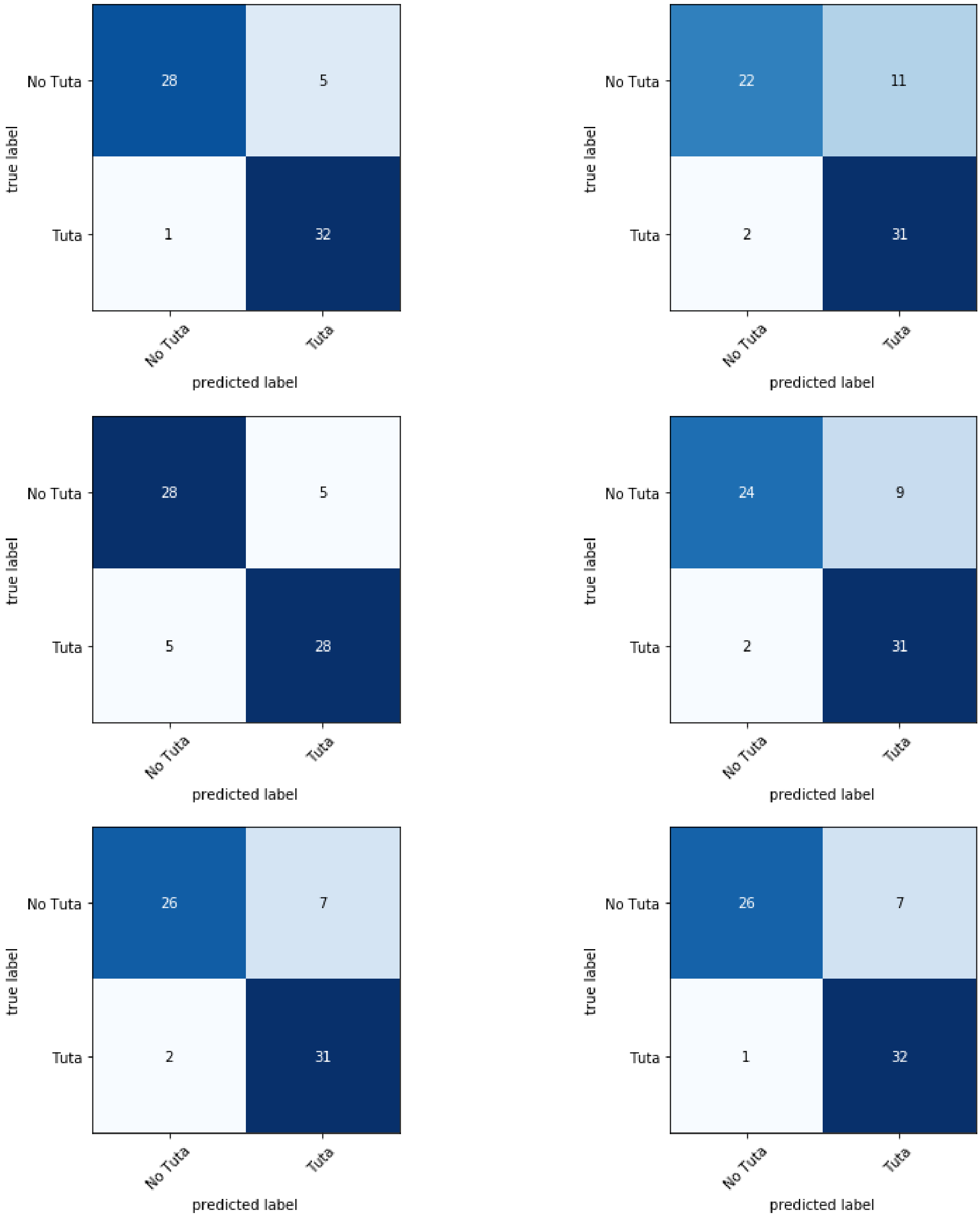
Appendix 15: Confusion matrices for VGG19 on 85:15 dataset



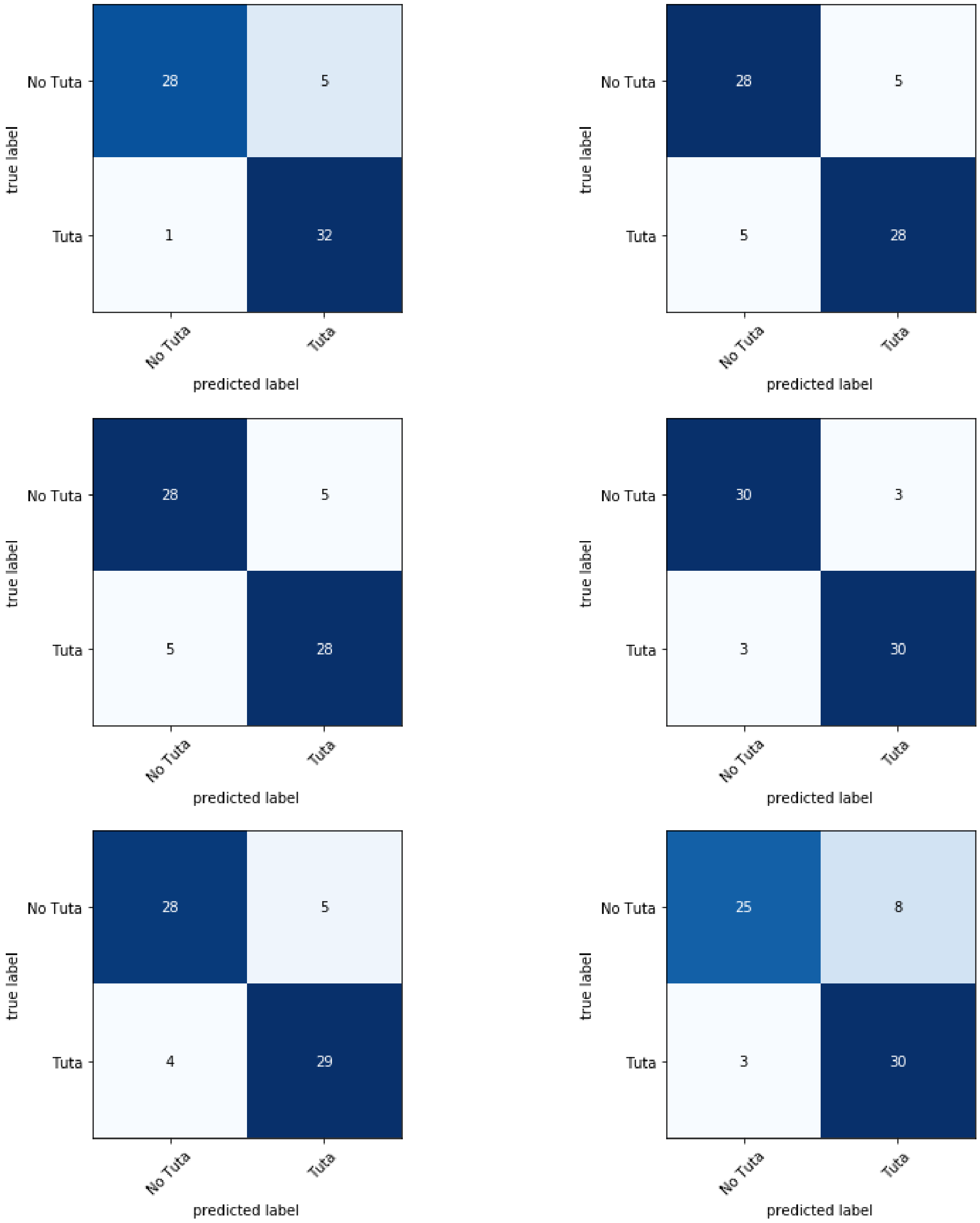
Appendix 16: Confusion matrices for ResNet50 on 75:25 dataset



Appendix 17: Confusion matrices for ResNet50 on 80:20 dataset



Appendix 18: Confusion matrices for ResNet50 on 85:15 dataset



RESEARCH OUTPUTS

1. **Rsearch Articles**
2. **Poster presentation**