



MAKERERE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCES.

DISEASE DETECTION IN POULTRY USING MACHINE LEARNING.

By Group CS22-5

**DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY.**

A Project Report submitted to the School of Computing and Informatics Technology
In Partial Fulfillment of the
Requirements for the Award of the Degree of Bachelor of Science in
Computer Science
Of Makerere University

Supervisor: Dr John Ngubiri

School of Computing and Informatics Technology, Makerere
University
ngubiri@cit.ac.ug, +256 776 543 466.

November 10, 2022

Declaration

We Group CS22-5 do hereby declare that this Project Proposal Report is original and has not been published or submitted for any other degree award to any other University before.

#	NAME	REGISTRATION NUMBER	Signature
1	NALWANGA PATRICIA	19/U/17304/EVE	
2	NAMUGANGA EMMILLY IMMACULATE	19/U/0126	
3	AWATH JAVAR ABDAT	19/U/0582	
4	KIRABIRA MWESIGWA MARVIN	19/U/16662/EVE	

Date:

Approval

This Project Report has been submitted for examination with the approval of the following supervisor.

Signed:

Date:

Dr. Ngubiri John
Department of Computer Science
School of Computing & IT
College of Computing & IT
Makerere University
ngubiri@cit.ac.ug

Contents

Declaration	iii
Approval	1
INTRODUCTION	3
Background to the Problem	4
Problem Statement	6
Objectives	6
Scope	6
Significance	6
LITERATURE REVIEW	7
Introduction	8
Brief overview of existing methods	8
Limitations	9
Why Machine Learning in Disease diagnosis	9
Merits of Machine Learning	9
Demerits of Machine Learning	10
Types of ML	10
Image classification techniques	10
Overview Of Existing Machine Learning Methods	11
Relationship between Coccidiosis and Droppings	14
Our System	15
RESEARCH METHODOLOGY	18
Introduction	18
Systems Analysis And Design	19
Requirements Collection And Analysis	19
Analysis Methods	19
Interviews	20
Document Review	20
Observation	20
Data collection instruments and tools.	20

System Implementation, Testing And Validation	21
Data collection, model building and Training.	22
Structure of Dataset.	22
Data Preprocessing	22
Splitting The Dataset	22
Augmentation	23
Creating a model	24
Transfer learning/fine-tuning	24
Efficient Net B6 Architecture	24
Batch Normalisation	25
Dense layers	25
Dropout layers	26
Final layer	26
Predictions	29
Evaluation Measures	29
Validation and accuracy of training	29
Confusion Matrix	30
Mobile Application Design.	35
System Design.	35
Use Case diagram.	35
Architecture design.	36
Data flow diagram.	37
Flow chart.	39
Technology decisions.	39
Flutter.	40
Docker.	40
4Sync.	40
Microsoft Azure.	40
Tensorflow.	41
Matplotlib and seaborn	41
Numpy and pandas	41
System implementation.	41
Android App features.	41
Android App interfaces.	42
Conclusion and Future work.	42
Conclusion.	49
Limitations of our study.	49
Future work.	50
Appendix and References.	50
Budget	51
Time Frame	52

INTRODUCTION

Poultry farmers maintaining a healthy flock highly contributes to the productivity of the farm. Therefore, in an emphasis to recognize some of the signs and symptoms some diseases for early diagnosis and prevention is key.

Background to the Problem

According to Shinyekwa et al. (2016), around 70% of the population is employed in the agricultural sector, contributing half of Uganda's export earnings and a quarter of the country's gross domestic product (GDP) thus posing as one of the most important sectors in Uganda's economy.

According to the Uganda National Household Survey (UNHS) 2016/17, agriculture was reported to be one of the main economic activities, which employed the highest percentage of the working population (64.3%) and accounted for the largest share of employment (36%). The sector is currently the third most important after manufacturing contributing about 21.9 percent to Gross Domestic Product (GDP) (UBOS, 2019). The Uganda National Population and Housing Census (NPHC 2014) estimated about 80 percent of households in the country involved in agriculture and of these about 90 percent were in the rural.

The sector also plays a predominant role of providing food to the ever-growing population both in urban and rural areas. In addition, the sector forms a basis for industrialization in the country by providing raw materials.

Notwithstanding its contribution to the economy, agriculture at the various stages of the value chain is susceptible to a number of constraints (Chatterjee and Oza, 2017); One of the most significant being disease and inadequate capital. Livestock farmers in particular have insufficient funds to sustain their farms in terms of feed, infrastructure and health.

In Uganda, poultry, the largest group of livestock species in the world, is a major component of rural livelihoods and is also reared in some urban areas today with chicken dominating the poultry industry and is mainly carried out for the production of eggs, meat, and put up for sale. The high number of people practicing poultry farming is attributable to its immense benefits: High profit rate and minimum requirements required to start, management being not all that easy nevertheless considering the inevitability of some poultry diseases which are caused by some factors like hygiene and climate change.([Gale et al., 2009](#))

Despite its popularity and dominance in livestock animals reared, it is risky and has faced several challenges like lack of information in poultry systems, financial scurvy consequently leading to feeding problems in terms of water and poultry feed, poor management, and lastly diseases ([Alemneh and Getabalew, 2019](#)). Diseases, like Salmonellosis, Coccidiosis and avian influenza, being king among the fatal frailties can be a devastating enigma that can render farms empty.

Coccidiosis is one of the world's most economically damaging poultry diseases and is caused by a protozoan called Eimeria which invade the cells of the poultry intestine. In the International Journal of Veterinary Science and Medicine ([Byaruhanga et al., 2017](#)), a retrospective study on cattle and poultry diseases in Uganda carried out between April 2012 and March 2014 was conducted using records for cattle and poultry diseases diagnosed at the Central Diagnostic Laboratory (CDL) to determine prevalent diseases in Uganda. The laboratory received 836 samples from poultry (36.3%) and cattle (63.7%). Of the 836 samples, 47.5% had a definitive diagnosis of disease causation. Most of the cattle and poultry diseases diagnosed were protozoan diseases (39.3%) followed by bacterial (21.4%), viral (17.1%), helminthiasis (11.1%), nutritional diseases (4%) and others (7.1%). For poultry, viral diseases (29.5%) and protozoan diseases (27.1%) especially Newcastle disease (44.3%) and coccidiosis (100%) respectively, were the most diagnosed.

Against this background, this research examines the need and relevance for poultry disease detection systems in regards to increasing production in farming and aims at employing an autoregressive model in order to extract a relationship between poultry droppings and coccidiosis as a precaution to disease outbreak in poultry. Fortunately, coccidiosis is treatable if detected early enough. This will allow for interactions to be taken sooner, for example by quarantining poultry houses or farms, thereby decreasing the chance that infection can spread to other groups of poultry. The ability to contain infectious diseases on poultry farms could benefit immensely from systems that first can rapidly detect unhealthy or sick birds, and secondly devices that can accurately and rapidly determine the causative agent that led to disease. ([Astill et al., 2018](#))

Problem Statement

Diseases in poultry farming have severe and detrimental health effects to poultry hence leading to death of the birds and loses for the bird owners if no remedies are provided. The process of diagnosis which involves the use of a specialist can be quite costly and may be dawdling since appointments have to be made with veterinary doctors to get treatment for ones poultry.

Poultry farmers are unable to keep track of the health status of their flock in order to prevent diseases through early diagnosis for diseases such as coccidiosis that can be detected and treated.

Objectives

Aside from developing a cost effective generic system that shall easily diagnose some diseases in poultry that can be detected in their stool, the research seeks to accomplish the following objectives:

- (a) *To scrutinize the existing methods of disease diagnosis in poultry.*
- (b) *To design a poultry disease diagnostics tool for poultry farmers.*
- (c) *To review literature related to the study problem so as to identify requirements.*
- (d) *To implement an automated mobile-based disease diagnostic system for poultry farmers.*
- (e) *To test and validate the system.*

Scope

This project attempts to detect diseases in poultry like Coccidiosis and also determine healthy birds. Coccidiosis is viral disease that is very fatal in chicken.

Secondary sources of data like Zenode dataset which contains thousands of faeces samples for both healthy and diseased poultry were used for designing the machine learning model on known data; we also collected sample images from some poultry farms.

Significance

The research attempts to

- (a) Reduce on the time and cost spent in disease diagnosis in poultry.
- (b) Provide an easy to use application that can be used as a data collection tool for other purposes of research.
- (c) Increase on productivity through early detection and quick diagnosis of poultry diseases.

LITERATURE REVIEW

Introduction

Diseases are a major limiting factor in large scale livestock production; they have disastrous effects on production economy, food safety and bio-security.

Brief overview of existing methods

During diagnosis of poultry diseases various methods can be used such as:

- i) Keeping records of daily mortalities, feed, water and egg production can help poultry farmers to detect an illness early enough to help start treatment early hence saving more birds. This approach to diagnosis is quite hectic as it requires the farmer to keep track of the quantities of uptake of poultry and their egg production.
- ii) External examination through observation of birds to look for signs such as severe bloody diarrhea, decreased growth rate, reduced food and water consumption, high mortality, development of culls, decreased egg production.
This method necessitates the farmer to have knowledge of disease symptoms which can be acquired through experience.
The demerit with this method is that the actual disease cannot be identified because of similarities of signs and symptoms between the various diseases. by the time these signs are seen it may also be too late to save the flock.
- iii) Internal examination such as necropsy where the bird is cut open and intestines examined for example for coccidiosis a disease of the intestinal tract will cause sores in the gut, thickened intestines and light spots on the surface of the gut. This method is usually limited to professionals such as veterinary doctors and usually done in labs after the death of the birds. It is quite complex for an average human to carry out as they usually bury the birds once they die.
This method can also only be carried out with 6 hours of the bird dying.

- iv) Lab tests for microorganisms like protozoa by using samples such as of faeces,cloacal swabs.These tests are quite expensive as they require laboratory services by professionals and are paid for.

Limitations

These methods are the traditional methods currently available to diagnose poultry diseases.

- (a) They are quite hectic to carryout and maintain.
- (b) Require some experience. However as not everyone starts with this kind of experience,it can be quite difficult for a normal person to figure out what is wrong with their flock without professional help.
- (c) They can take a long period of time to be diagnose such as signs which can disguise.

Why Machine Learning in Disease diagnosis.

Machine learning and natural language processing are forms of artificial intelligence that enable robust interrogation of multiple datasets to identify previously undiscovered patterns and relationships in the data.(Myszczynska et al., 2020)

Machine learning is an application of Artificial Intelligence that uses algorithms and statistics to find patterns in large amounts of data.(Nozari et al., 2021)

The Machine Learning algorithm looks for a set of rules that allow it to deduce the general characteristics of elements within a group with the objective of applying the learning to similar elements. Machine learning approaches have been applied in study of diseases and shows a lot of promise in early disease diagnosis and predictions.

Merits of Machine Learning

- i) Machine language saves time, effort and money by doing time consuming tasks in the least time possible using algorithms.

- ii) Machine learning can help to forecast or make predictions depending on the available data for instance (Rodriguez et al., 2021)
- iii) Machine learning easily identifies trends and patterns in large volumes of data that would be apparent to humans.
- iv) Usually does not involve human intervention as they are able to learn ,make predictions and improve on their own.

There are various machine learning algorithms available and choosing the right algorithm is substantial to attaining reliable results.

Demerits of Machine Learning

- i) As much as machine learning can help in diagnosis and predictions,it cannot replace a specialist who still has to administer the treatment.
- ii) The accuracy of these machine learning models largely depends on the quality of data they are fed.The images used have to be clear such that patterns can be identified.
- iii) In case of delay of updating a model,continuing to use a trained model that makes unreliable predictions can be dangerous. Kononenko (2001)

Types of ML

The types can be classified into four:

- i) Supervised: All data is labeled and algorithm learns to predict the output from input data
- ii) Unsupervised: All data is unlabeled and algorithm learns the inherent structure from the input data.
- iii) Semi-labeled: some data is labeled but most of it isn't and a mixture of supervised and unsupervised techniques are used.
- iv) Reinforcement: Deals with how software agents ought to take action in an environment to maximize the notion of reward.

Image classification techniques.

After training the model with images of droppings for the birds, the decision of choosing an appropriate and most accurate classification algorithm or technique was the next problem to consider. There are several classification techniques for image classification including but not limited to the following :-

- i) Support Vector Machine (SVM) : It builds a hyper-plane or a set of hyper-planes in a high dimensional space and good separation between the two classes is achieved by the hyperplane that has the largest distance to the nearest training data point of any class.
- ii) Decision Trees: Decision trees are based on a hierarchical rule-based method and permits the acceptance and rejection of class labels at each intermediary stage/level.
- iii) K-NN: This algorithm simply relies on the distance between feature vectors and classifies unknown data points by finding the most common class among the k-closest ones.
- iv) CNN: Convolutional neural networks (CNN) is a special architecture of artificial neural networks. CNNs uses some of its features of visual cortex and have therefore achieved state of the art results in computer vision tasks.

We shall therefore use CNN because of its high precision and recall rates. Their ability to develop an internal representation of a two-dimensional. This allows the model to learn position and scale in variant structures in the data which is important when working with images.

Overview Of Existing Machine Learning Methods

Images are artefacts that depict visual perception. They have been used for diagnosis and detection in various fields including medical, agricultural and other fields. This project focuses on using images of chicken droppings to detect diseases in chicken.

(Sadeghi et al., 2015) used Support Vector Machine (SVM) and Decision Tree to detect sick chickens infected with clostridium perfringens using the sound they make. In their study, vocals from both health and unhealthy chickens were taped. Facets were extracted and used to train the classifiers, whereby the accuracy of the neural network increased gradually up to 100%.

(Hepworth et al., 2012) predicted the regularity occurrence of hock burn, swelling skin around the hock in broiler chicken. Data from farms were collected in a period of over 36 months and, learned variables of dependency were extracted, and a classifier was trained to attain an accuracy of 78%.

In work by Hemalatha and Maheswaran (2014), SVM was used to diagnose avian pox in chicken, images of chicken from the farm were collected then split into training and test sets. The classifier was trained on the data and obtained an accuracy of 92.7%. Despite good prediction performance of the classical machine learning approaches, (Ferentinos, 2018) presented that traditional machine learning techniques are constrained in images and features processing.

The deep learning techniques have gained more attention in computer vision and image classification, particularly in enhancing the performance of image classification and retrieval as opposed to traditional machine learning approaches. Therefore, in this research a Convolution Neural Network (CNN) is used due to the following reasons:

1. It involves multi-layer processing.
2. It allows optimization of the extracted features.
3. It is fast and requires less computational power.

Deep Convolution Neural Networks (DCNN) enable the computer to interpret captured data objects (feature extraction and representation) for classification, localization and recognition to be automatically learned. It has been used for early disease detection in both plants (crops) and animals.

In poultry, digestive diseases such as coccidiosis are among the most common diseases of birds. Droppings and observation of the chicken's behaviour are some of the most precise techniques to detect the occurrence these digestive disease infections among the birds. This project focuses on using droppings as the method for disease detection. Other studies for example Zhuang et al. (2018) focused on the skeletal analysis of the chickens for early detection of sick broilers by image processing.

Checking the shape, colour and water content of the chickens' droppings helps detect these digestive diseases in the chicken. Droppings vary in shape, colour and texture depending on diet, time of the year, chicken species and health and since chickens can normally defecate up to 12 times a day; many samples of the dropping are readily available for detection.

According to Damerow (2016), chickens can have different colours and shapes of droppings depending on health or their diet. Hence for effective detection, only birds fed on a standard diet can variation in chicken droppings be useful for disease detection.

Wang et al. (2019) used two deep convolutional neural network algorithms and compared the results; the YOLO-V3 and Faster R-CNN. YOLO-V3 was also improved by optimizing the anchor box. They were non-intrusively detecting broiler chickens, their study focused on broilers because broilers are main contributors for white meat.

Faster R-CNN achieved 99.1% recall and 93.3% mean average precision while YOLO-V3 achieved 88.7% recall and 84.3% mean average precision.

Faster R-CNN performs 4-step target detection; it does the region proposal then feature extraction to classification and regression. It uses Region Proposal Network which improves its accuracy and speed.

While YOLO-V3 merges region proposal and classification into one step, task recognition, with the whole image as region of interest. It also merges feature interaction layer of 3 scale production and uses FPN algorithm to merge the features of different scales to enhance the detection effect of small targets.

All neural networks were trained by asynchronous stochastic gradient descent and also used region proposal network and ResNet models. They used Tensorflow and darknet frameworks too.

Recall (R) was calculated as:

$$R = \frac{TPos}{Pos} \quad (1)$$

TPos – Result of correct classification

Pos – Positive Sample

Average Precision (AP) calculated as

$$\int_0^1 P(R)dR \quad (2)$$

Mean Average Precision (mAP) as

$$\frac{\sum_{i=1}^C AP}{C} \quad (3)$$

C – Number of sample classification

They used 50000 iterations of training and only classified birds as either healthy or unhealthy.

Another group [Mbelwa et al. \(2021\)](#) performed a similar study on Tanzanian birds while using four classes for classification i.e. Healthy, Coccidiosis, Salmonella and New Castle Disease each corresponding to the health or disease the bird has.

These also used Transfer learning for image classifications using models like VGG, Resnet and XceptionNet. They collected images from different farms using mobile phones and never did any preprocessing on the images save for resizing them compared to the former study where they did several preprocessing on the images to augment their image dataset.

Relationship between Coccidiosis and Droppings

Coccidiosis is one of the most fatal diseases in poultry worldwide. It is caused by a protozoan called Eimeria which invades the cells of the poultry intestine. This disease is host specific meaning that other species of birds cannot transmit their coccidiosis to the chicken although the disease can spread like a wildfire from chicken to chicken with an incubation period of 4 to 8 days which means that infected poultry can be treated once the disease is spotted early. Coccidiosis usually affects birds 3-6 weeks before they have acquired immunity and is particularly difficult to combat because several different species of Eimeria exist in the field.

Poultry may become infected with different species because the immunity that develops after infection is specific only to one species. Eimeria has a very complex life cycle that involves many developmental stages within the host cells. Each Eimeria type is able to infect only one host species and each attacks a different segment of the intestine in their host.

In chickens there are nine described Eimeria species, and in turkeys there are seven species; however, not all are pathogenic. In addition to being species specific, most coccidia infect specific areas of the intestinal tract resulting in enteritis and typhlitis. An exception is renal coccidiosis reported in geese and ducks.

Coccidiosis is transmitted via the **droppings** from infected birds: When the organisms are spread by the faecal-oral route when birds ingest sporulated oocysts

from infected feed, water, litter or soil. Therefore, anywhere there's a microscopic trace of bird droppings such as in a waterer, a feeder, or in bedding, there's almost certainly coccidia present. Veterinarians and animal care staff can also spread disease because coccidia are readily transported on boots, shoes, clothing, crates, vehicle wheels and other animals and insects.

Droppings of chickens look and appear different depending on the status or condition of the chicken; hence therefore chicken droppings are a good source of data (visual data) to make inferences about the health of the chicken.

Normal droppings can vary depending on diet, time of the year and the health status. On farms where there is a standard diet, the change in droppings could be easily affiliated with the health status of the flock, and hence chicken droppings (faeces) can be one of the first signs of disease and illness.

It is important for a farmer to recognize abnormal droppings and what they signify then also know what to do about it. <https://the-chicken-chick.com/whats-scoop-on-chicken-poop-digestive/>

There are three common features to look out for when observing chicken droppings

- i) Color of the droppings
- ii) Texture of the droppings (watery or firm)
- iii) Blood stain in the droppings

For coccidiosis infected chicken, the birds have diarrhoea and dehydration, with high morbidity and mortality. Diarrhoea may be watery, mucoid and/or haemorrhagic. Infection progresses rapidly, and by the time the flock is examined the birds may have ruffled feathers, anaemia, weakness, listlessness and/or somnolence.

These features are usually observed by sight and therefore our interest to employ computer vision techniques to identify the disease they could be having.

The argument being that the image of normal droppings will defer from those with abnormal droppings hence classification.

Our System

1. Fast diagnosing process for poultry diseases such as coccidiosis through using machine learning models to predict the disease. High level approach to enable a normal farmer to use the technology through embedding it into a mobile application they can use to scan their flock droppings whenever they wish to keep track of their health status.

2. Early diagnosis as the farmer is able to use the technology at any time hence easily identify diseases early enough.
3. The farmer will be given advice on what to do once a disease has been detected and provide educative content about disease diagnosis.

Our study proposed a different model made after finetuning the EfficientNetB6 model that belongs to the family of efficientNet. These models are light weight and suggest a new way of scaling a model through uniformly scaling the dimensions of depth, width and resolution using a simple yet highly effective compound coefficient.

Resolution scaling deals with low and high resolution images, and if we trained it on the low resolution images, the model gets less features and if it learns on less information then it will be less accurate. So because Efficient net tries to increase the size of the input and increase resolution hence more accuracy is achieved.

Depth scaling (more number of layers) is necessary because of the high resolution of images that are used.

Width scaling is about increasing the number of channels /feature maps and this is bound to capture more features which will increase the models accuracy.

The compound scaling method is justified by the intuition that if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image.

EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters.

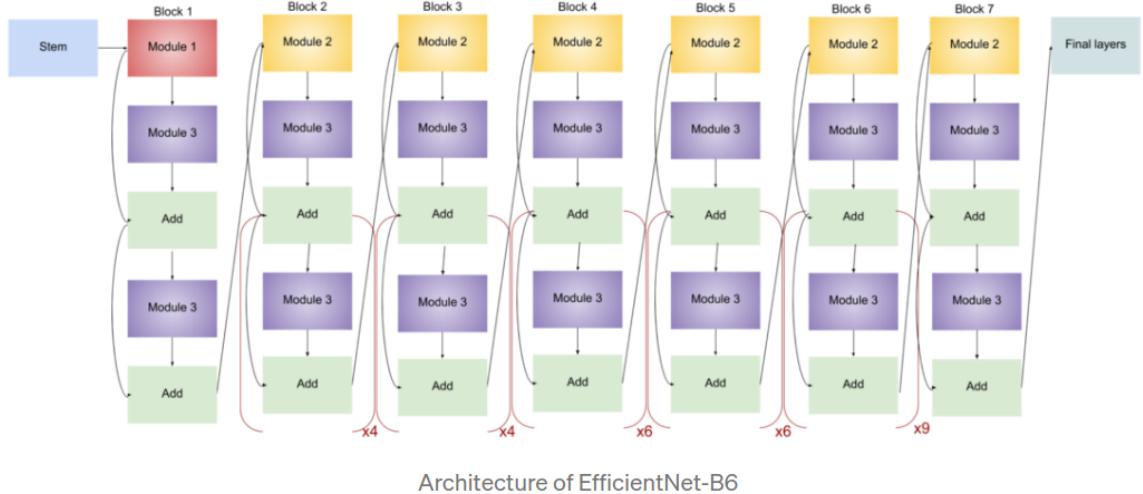
The EfficientNet-B7 achieves new state-of-the-art 84.4% top-1 / 97.1% top-5 accuracy, while being 8.4x smaller than the best existing CNN followed by the EfficientNetB6.

Below is a list of their performance.

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	76.3%	93.2%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	78.8%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	79.8%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.1%	95.5%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.6%	96.3%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.3%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.9%	43M	1x	19B	1x
EfficientNet-B7	84.4%	97.1%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

Figure 1: Performance list

EfficientNet-B6



Architecture of EfficientNet-B6

Figure 2: EfficientNet Architecture

RESEARCH METHODOLOGY

Introduction

For any project, data is paramount and the usefulness the information gathered is proportional to how well managed it is to produce desirable results lest it fails.

As system analysts, this data or information should be critically analyzed to ensure the smooth flow of the Systems Development Life Cycle(SDLC). It is upon this analysis that the Agile methodology was selected for our project. We shall therefore embark on this method as a model for the Disease Detection in Poultry whereby we shall work on various prototypes to purposefully engage with the intended end users through various interviews which shall be held during the requirements collection and analysis stage.

Basing on our analysis, this shall enable us set our priorities and redirect plans inexpensively with reduced disruption. The catch is to produce release cycles of small incremental changes from previous releases which is imperative for the Agile method of systems development([Petersen and Wohlin, 2010](#))

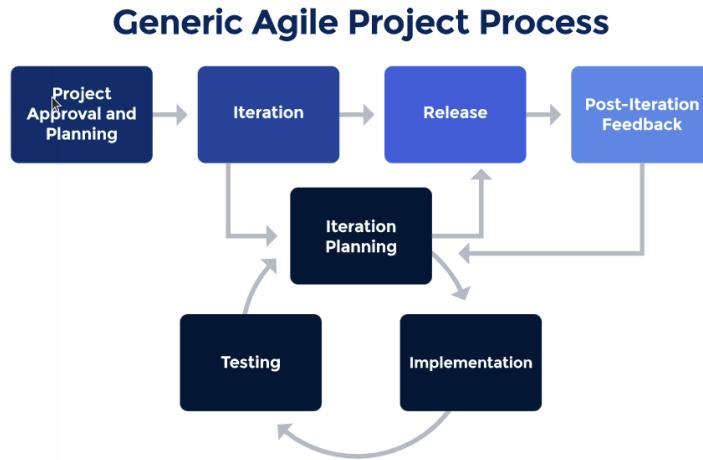


Figure 3: Agile Method

Systems Analysis And Design

Systems Analysis and Design (SAD) simply refers to the methodologies used for developing Information Systems which combine Information Technology, people and Data to support business requirements.(Dennis et al., 2015)

System analysis refers to the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components and is conducted for the purpose of studying a system or its parts in order to identify its objectives.(Boaduo, 2011)

Systems design refers to the process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements.

This systems analysis and design is carried out in phases which include requirements collection and analysis, systems design, Implementation, testing; deployment and maintenance.

Requirements Collection And Analysis

During this phase, all the relevant information was collected using both qualitative and quantitative methods to develop a product as per user expectation and ambiguities are to also be resolved in this phase and a deliverable called a Business Analysis Report is drafted as a summary of this stage.

Analysis Methods

Analysis is defined as examination of the data with the purpose of making meaning out of it. And it shall involve strategically gathering and analysing the data facts for the desired poultry disease detection system. It will essentially streamline the required information and various methods or techniques employed. The quantitative methods used shall help in evaluating the impact of the current existing methods of poultry disease detection and classification. Interviews ,personal observations of the other systems and document reviews shall used in order to identify the User Core Requirements for the desired system.

Interviews

An interview is an important qualitative research method in which researchers collect data directly from the intended system users. This technique is time saving, less costly and is flexible since its design depends on the situation at hand and in-depth analysis. Therefore, we shall collect first hand information that will enable proper analysis of the problem.

Document Review

Using this research analysis method, we shall collect information from already existing research from online sources as internet, journals, reports as well as articles to get a general background on the system requirement specifications. With the study, a gap between the literature can be identified thus providing a basis for the development of the developed system.

Observation

Depending on the poultry farm visits we intend to carry out, we shall carefully look out for any useful information that can aid the objectives of this project while observing the process through which the disease is diagnosed.

Data collection instruments and tools.

The dataset will be obtained from zenode and will be used to design, build and train the model. To test the model images from the local farms verified by the veterinary doctor will be used.

The tools that will be used to collect the data include.

1. Phones Cameras to take pictures of chicken droppings from the farms.

2. Audios and video data of the interviews and visits shall be stored and used to analyse the data.



Figure 4: Smaple Images

System Implementation, Testing And Validation

Data collection, model building and Training.

Structure of Dataset.

The dataset was obtained from <https://zenodo.org/record/5801834#.YzrgPtJBxki> in a similar study carried out in Arusha and Kilimanjaro regions in Tanzania between September 2020 and February 2021. They are annotated with classes "Coccidiosis", "Healthy", "New Castle Disease", and "Salmonella". We got a subset of the dataset for our study, Coccidiosis and Healthy, and hosted it on Kaggle at <https://www.kaggle.com/datasets/emmillyimmaculate/poultry-cocci>. The dataset consists of a total of 4880 images of chicken droppings. 2476 images of coccidiosis and 2404 images of healthy. All the images are within one folder which is the Train folder with a CSV file that lists each image with its label.

Data Preprocessing

Splitting The Dataset

The dataset has been divided into the train, validate and test datasets using the *train_test_split* function of the *sklearn.model_selection* package in python which is used to split arrays or matrices into random subsets for train and test data. Two splits were performed, the first being on the whole dataset where we used a 0.8 *train_size* meaning that 80% of images were used for training. This means that we had 1981 Coccidiosis and 1923 Healthy images for training, 165 coccidiosis and 161 Healthy images for validation, and 330 coccidiosis and 320 healthy images for testing. We particularly made the testing dataset bigger to get a better view of the accuracy of the model.

The second split was performed on the remaining 20% of images where $\frac{2}{3}$ of the dataset were left for testing and $\frac{1}{3}$ for validation.

We applied a function to balance the training dataset to reduce the effect of unequal number of images from either classes leaving us with 1900 images of each class for training.

Additionally, we collected around 100 images of healthy and coccidiosis that were added to the testing dataset which we used to check the accuracy of the model in relation to our home environment in Uganda. These images were collected with a smart phone. Three farms were visited where there were confirmed outbreaks of coccidiosis from the farms. We bought one of the infected birds. The bird was close to death and this could not allow us collect bigger datasets however many images were taken of the same droppings.

The dataset was annotated with images of cocci having cocci.number.jpg. The number being a variable.

The Healthy images were annotated healthy.number.jpg as well.

Augmentation

To counteract over-fitting and memory we performed data augmentation on our dataset using the keras ImageDataGenerator which augments images in real time as the model is still running. This means that for each epoch new transformed images will be used for training which will be better than us physically storing the transformed images.

The transformations applied include:

1. Zoom range of 0.2

This will be important for images were taken from a bit far such that we can get closer look at their contents through bringing them closer hence a zoom range less than 1 specified.

2. Horizontal Flip.

This means that images will be flipped horizontally.

3. Rotation range of 60.

This means that images were rotated at 60 degrees during each rotation transformation

4. Width shift range of 0.3
This will shift 30% of the total width of the image to the between a range of -30% of total image width to +30% of total width of the image.
5. Height shift range of 0.3
This will shift 30% of the total height of the image to the between a range of -30% of total image width to +30% of total height of the image. These transformations were applied to the training dataset. We did not perform any augmentation for the validation and test datasets however we used it to generate a Dataframe iterator of batches of images in with shape (batch-size, target-size, channels) with their corresponding labels. The training dataset was also shuffled during the process unlike the validation and test datasets because shuffling affects learning in the updates of parameters of the model hence train dataset is shuffled but since no learning is happening during the usage of the validate and test datasets so it would not make any difference. To train faster and for memory efficiency we used a batch size of 20 for the train and validation dataset while for the test dataset having around 650 images we used a batch-size of 50 to ensure we go through the test dataset exactly once.
6. Resizing of images
The images were also resized to 224 by 224 during the augmentation process.

Creating a model

Transfer learning/fine-tuning

Transfer learning consists of taking features learned on one problem, and leveraging them on a new, similar problem. For instance, features from a model that has learned to identify racoons may be useful to kick-start a model meant to identify tanukis. From https://www.tensorflow.org/guide/keras/transfer_learning

Efficient Net B6 Architecture

The Efficientnet model is based on the EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks paper. It suggests a new way of scaling a model through uniformly scaling the dimensions using a simple yet highly effective compound coefficient.

Using the knowledge of transfer learning we used EfficientNeTB6 pretrained model on imagenet dataset. Because the EfficientNet models achieve both higher accuracy

and better efficiency over existing CNNs, reducing parameter size and FLOPS by an order of magnitude.

We basically decided to use all the layers of the EfficientNetB6 model as feature extractor component of our new classifier. This was done by loading the model and simply adding new layers.

During loading the EfficientNetB6 model with its required modal weights, we set `\include_top` to false which excludes the fully-connected output layers of the model used to make predictions and allows us to add a new trainable output layer to be added and trained. Following that we also set the “input-shape” which specifies the fixed-size input of the model to be changed. Since a model without a top will output activations from the last convolutional or pooling layer directly, we set the pooling to max to summarize these activations for their use in the classifier. This results is a vector that will be used as a feature descriptor for our input. Since we wish to use the weights trained on imangenet we set the weights to imangenet.

We set the EfficientNetB6 model to trainable such that we could train all its layers on our dataset.

Batch Normalization

We also added a batch normalization to the outputs of the EfficientNetB6 by adding a batch Normalization Layer to improve the learning speed by maintaining the mean output close to 0 and the output standard deviation close to one. This ensures that the weights in the network don't become imbalanced with extremely high or low values since We set axis to 1 to refer to the features axis as the one to be normalized. Momentum to 0.99 for the moving average. Epsilon to 0.001 to add it to variance in order to avoid division by zero.

Dense layers

We added two hidden layer with 1024 and 128 nodes respectively both using relu as activation function.

Regularizers

To apply modifications to the learning algorithm in order to reduce generalization errors on the dataset. We applied them to put penalties on layer parameters or layer activity during optimization. So for each hidden layer we set 3 regularizers. We set the [L2 regularizer](#) for the *kernel_regularizer* which is the regularizer function applied to the kernel weights matrix and L1 regularizer for both *activity_regularizer* which is the regularizer function applied to the output of the layer and the *bias_regularizer* which is the regularizer function applied to the bias vector. All this is to help reduce over-fitting.

Dropout layers

Due to the fact that our neural network is quite large, there are expectations of very many weights and very many biased parameters which may lead to over-fitting. So we applied dropout layers after every dense layer because they are the ones with the greater number of parameters and thus they're likely to excessively co-adapting themselves causing over-fitting. These layers randomly select neurons that are ignored during training. This means that during the forward pass their input is not taken and during the backward pass their weights are not updated.

Final layer

We finally added an output layer with 2 categories and softmax activation function.

```

base_model.summary()
165240832/165234480 [=====] - 0s 0us/step
165249024/165234480 [=====] - 6s 0us/step
Model: "efficientnetb6"

Layer (type)          Output Shape         Param #     Connected to
=====
input_1 (InputLayer)   [(None, 224, 224, 3) 0      []
)
rescaling (Rescaling) (None, 224, 224, 3) 0      ['input_1[0][0]']
normalization (Normalization) (None, 224, 224, 3) 7      ['rescaling[0][0]']
stem_conv_pad (ZeroPadding2D) (None, 225, 225, 3) 0      ['normalization[0][0]']
stem_conv (Conv2D)      (None, 112, 112, 56 1512    ['stem_conv_pad[0][0]']
)
stem_bn (BatchNormalization) (None, 112, 112, 56 224    ['stem_conv[0][0]']
)
stem_activation (Activation) (None, 112, 112, 56 0      ['stem_bn[0][0]']
)
block1a_dwconv (DepthwiseConv2D) (None, 112, 112, 56 504    ['stem_activation[0][0]']
)
block1a_bn (BatchNormalization) (None, 112, 112, 56 224    ['block1a_dwconv[0][0]']
)
block1a_activation (Activation) (None, 112, 112, 56 0      ['block1a_bn[0][0]']
)

```

Figure 5: Top layers

Model Compilation was done with Adamax and learning rate of 0.01.
We finally trained the model by feeding it our *train_dataset* that was augmented as well as the validation dataset.

```

    - -
    □ block7c_se_reduce (Conv2D)      (None, 1, 1, 144)   497808   ['block7c_se_reshape[0][0]']
    block7c_se_expand (Conv2D)      (None, 1, 1, 3456)   501120   ['block7c_se_reduce[0][0]']
    block7c_se_excite (Multiply)   (None, 7, 7, 3456)   0       ['block7c_activation[0][0]', 'block7c_se_expand[0][0]']
    block7c_project_conv (Conv2D)   (None, 7, 7, 576)   1990656  ['block7c_se_excite[0][0]']
    block7c_project_bn (BatchNormaliza- (None, 7, 7, 576)   2304   ['block7c_project_conv[0][0]']
    lization)
    block7c_drop (Dropout)         (None, 7, 7, 576)   0       ['block7c_project_bn[0][0]']
    block7c_add (Add)              (None, 7, 7, 576)   0       ['block7c_drop[0][0]', 'block7b_add[0][0]']
    top_conv (Conv2D)              (None, 7, 7, 2304)  1327104  ['block7c_add[0][0]']
    top_bn (BatchNormalization)   (None, 7, 7, 2304)  9216   ['top_conv[0][0]']
    top_activation (Activation)   (None, 7, 7, 2304)  0       ['top_bn[0][0]']
    max_pool (GlobalMaxPooling2D) (None, 2304)        0       ['top_activation[0][0]']

=====
Total params: 40,960,143
Trainable params: 40,735,704
Non-trainable params: 224,439

```

Figure 6: Bottom layers

```

Epoch 1/14
163/163 [=====] - 189s 916ms/step - loss: 18.5581 - accuracy: 0.9445 - val_loss: 11.3000 - val_accuracy: 0.9286 - lr: 0.0010
Epoch 2/14
163/163 [=====] - 142s 871ms/step - loss: 7.6848 - accuracy: 0.9790 - val_loss: 5.3427 - val_accuracy: 0.9821 - lr: 0.0010
Epoch 3/14
163/163 [=====] - 142s 872ms/step - loss: 3.7846 - accuracy: 0.9821 - val_loss: 2.6460 - val_accuracy: 1.0000 - lr: 0.0010
Epoch 4/14
163/163 [=====] - 142s 872ms/step - loss: 2.0602 - accuracy: 0.9858 - val_loss: 1.5498 - val_accuracy: 0.9964 - lr: 0.0010
Epoch 5/14
163/163 [=====] - 143s 873ms/step - loss: 1.2482 - accuracy: 0.9904 - val_loss: 0.9642 - val_accuracy: 1.0000 - lr: 0.0010
Epoch 6/14
163/163 [=====] - 142s 870ms/step - loss: 0.8017 - accuracy: 0.9945 - val_loss: 0.6347 - val_accuracy: 1.0000 - lr: 0.0010
Epoch 7/14
163/163 [=====] - 142s 872ms/step - loss: 0.5539 - accuracy: 0.9975 - val_loss: 0.4595 - val_accuracy: 0.9964 - lr: 0.0010
Epoch 8/14
163/163 [=====] - 142s 872ms/step - loss: 0.4208 - accuracy: 0.9963 - val_loss: 0.3488 - val_accuracy: 1.0000 - lr: 0.0010
Epoch 9/14
163/163 [=====] - 142s 871ms/step - loss: 0.3272 - accuracy: 0.9975 - val_loss: 0.2882 - val_accuracy: 0.9964 - lr: 0.0010
Epoch 10/14
163/163 [=====] - 143s 876ms/step - loss: 0.2801 - accuracy: 0.9948 - val_loss: 0.2373 - val_accuracy: 0.9964 - lr: 0.0010
Epoch 11/14
163/163 [=====] - 142s 870ms/step - loss: 0.2404 - accuracy: 0.9966 - val_loss: 0.2049 - val_accuracy: 1.0000 - lr: 0.0010
Epoch 12/14
163/163 [=====] - 143s 877ms/step - loss: 0.2157 - accuracy: 0.9988 - val_loss: 0.1899 - val_accuracy: 1.0000 - lr: 0.0010
Epoch 13/14
163/163 [=====] - 144s 882ms/step - loss: 0.2025 - accuracy: 0.9988 - val_loss: 0.1816 - val_accuracy: 0.9964 - lr: 0.0010
Epoch 14/14
163/163 [=====] - 143s 879ms/step - loss: 0.1852 - accuracy: 0.9988 - val_loss: 0.1666 - val_accuracy: 1.0000 - lr: 0.0010
training elapsed time was 0.0 hours, 35.0 minutes, 2.83 seconds)

```

Figure 7

Results/Performance

Classification Accuracy

Classification accuracy is calculated by the number of correct predictions divided by the total number of accurate predictions.

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (4)$$

Recall

. Recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect Coccidiosis samples. The higher the recall, the more positive samples detected. The recall is calculated as:

$$Recall = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (5)$$

Precision

. Precision refers to the number of true positives divided by the total number of positive predictions. Precision is one of such measures for accuracy of a model , which is characterised as:

$$Recall = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (6)$$

F1 Score

. One well-known metric that combines precision and recall is called the F1-score, which is defined as:

$$F1Score = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

Validation and accuracy of training

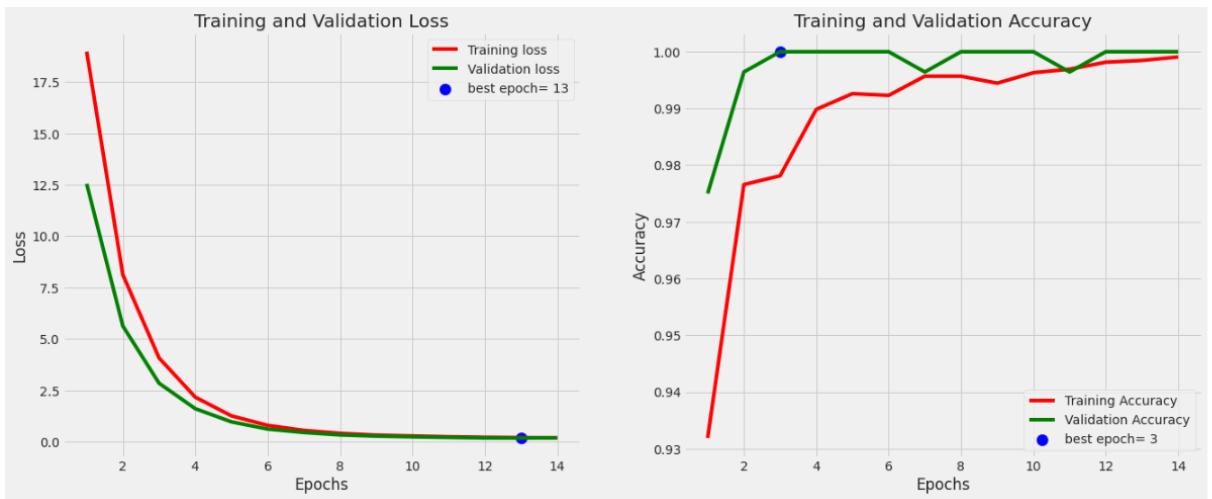


Figure 8: Validation and accuracy of training

Confusion Matrix

The matrix compares the actual target values with those predicted by the machine learning model. In this case a 2 by 2 matrix for the two classes considered. It can clearly be seen that 2 of the samples of coccidiosis were predicted as Healthy.

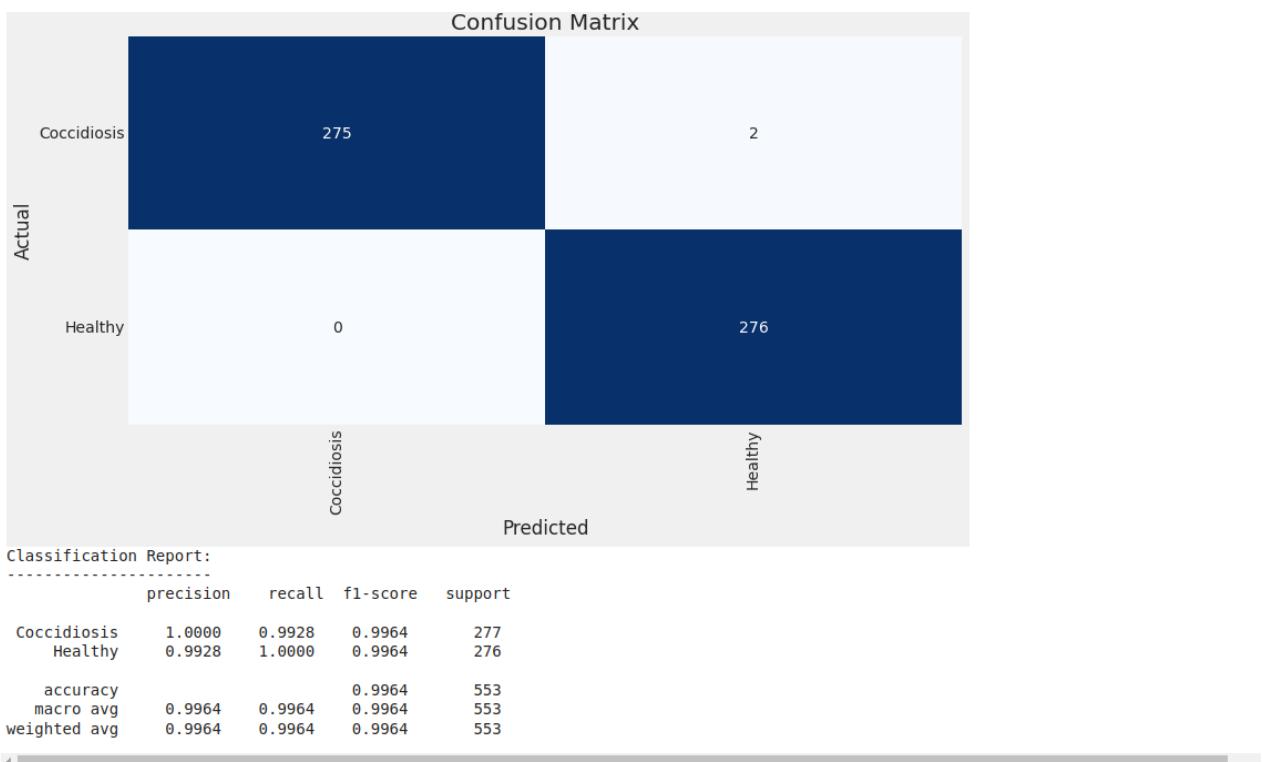


Figure 9: Confusion Matrix

Without Using EfficientNetB6 Model to initialise weights of our model

In comparison to removing the EfficientB6 model ,the performance of the model was very poor. There model was probably overfitting and required longer times to train.

Without using EfficientNetB6 we tested out 14 images for 20 epochs

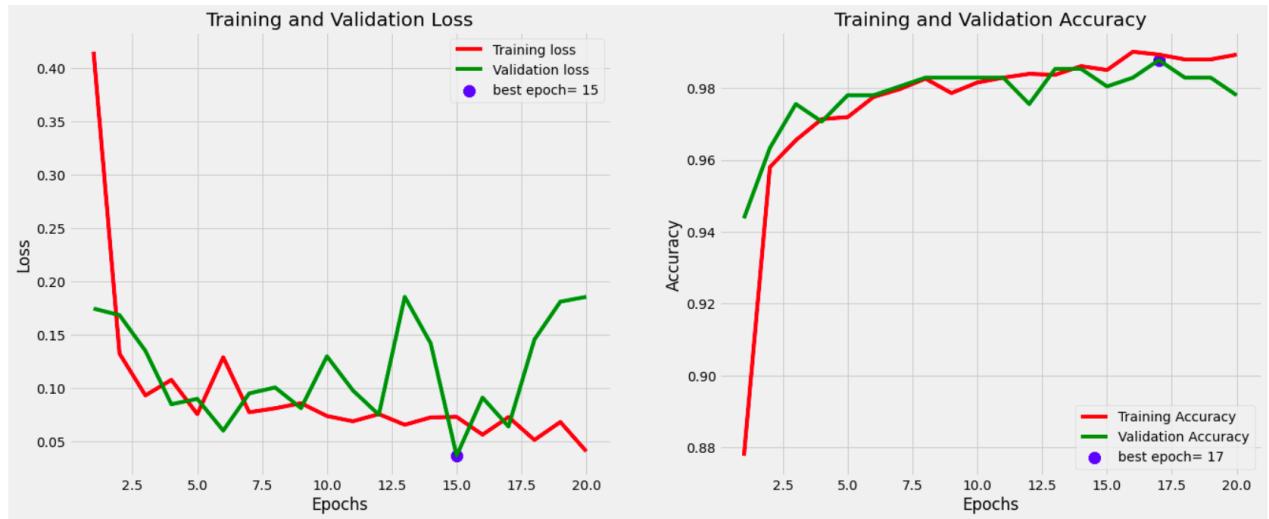


Figure 10: Without Initialising with EfficientNetB6 model

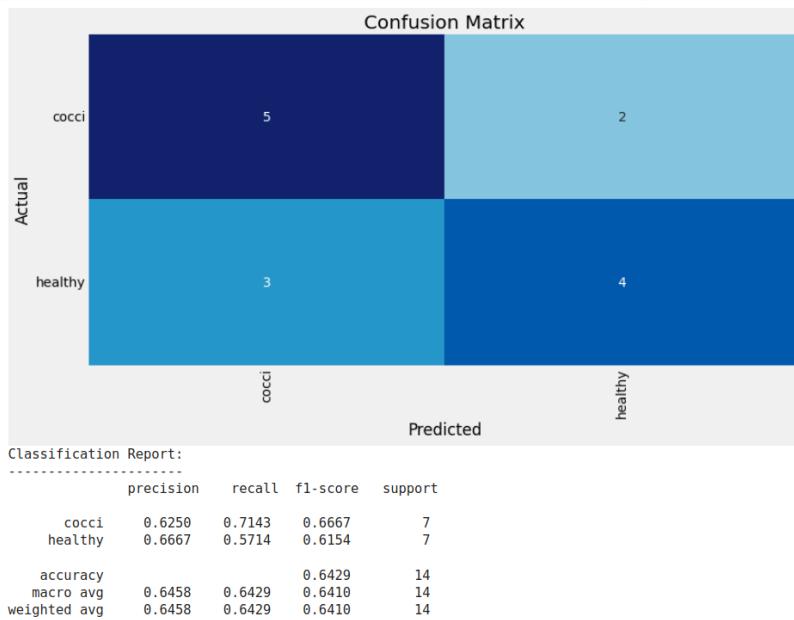


Figure 11: Confusion Matrix Without Initialising with EfficientNetB6 model

Technology Decisions

The physical realization of this research was implemented using carefully selected tools and technologies such as:

- (i) **Python:** Python is a high level interpreted language used for general purpose programming and is widely used for scientific computing and can be used for a variety of general tasks from data science to software development. Python shall be the main language used for this project.
- (ii) **NumPy:** NumPy is the fundamental package for scientific computing in Python (It stands for Numerical Python). It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. (Björk, 2021)
- (iii) **Pandas:** Pandas is defined as an open-source library that provides high-performance data manipulation in Python. is also used for data pre-processing and preparation. One of the main features about pandas is the DataFrame and Series data structure. These data structures are optimized and contain fancy indexing that allow a variety of features such as reshaping, slicing, merging, joining and etc to be available. Pandas and Numpy are extremely powerful when used together for manipulating data. (Nagpal and Gabrani, 2019)
- (iv) **Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is a cross-platform data visualization and graphical plotting library. Its numerical extension being NumPy and Matplotlib's combination with Jupyter Notebook is a popular way to visualize data using Python for all kinds of applications in science, technology, and education. (Der et al., 2017)
- (v) **OpenCV:** OpenCV (Open Source Computer Vision) is a well established computer vision library which is written in C/C++ and has been abstracted to interface with C++, Python and Java. This is a powerful tool when working with images and has a myriad of tools regarding image data manipulation, feature extraction Et Cetera (Gao et al., 2020)
- (vi) **Tensorflow:** TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources. Tensorflow at its core is about computations of multidimensional arrays called tensors but what makes Tensorflow great is its ability to be flexible to deploy computations on different devices such as CPU's and GPU's. (Hope et al., 2017)

- (vii) **Keras:** Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. Keras acts as an interface for Tensorflow library. It also contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.([Brownlee, 2016](#))
- (viii) **Google Colab:** Google Colab is a Deep Learning Platform in the Cloud. Google Colab allows developers to run heavy tasks on the cloud such as training the deep learning model or heavy preprocessing tasks. The CPU's and GPU's available on Google Colab are fully configured to work on different Frameworks such as Tensorflow. One of the main benefits to Google Colab is that it is quite powerful and easy to use with easy to follow documentation.([Arpteg et al., 2018](#))
- (ix) **Tensorflow Lite:** TensorFlow Lite is a mobile library that allows you to deploy models on mobile devices, microcontrollers, and other edge devices. This library makes running models on devices with low resources like mobile devices possible, enabling prediction or analysis of poultry droppings offline on Mobile devices possible.
- (x) **Github:** GitHub is a code hosting platform for version control and collaboration that helps developers store, manage, track and control changes to their code and also lets them work together on projects from anywhere. ([Fylaktopoulos et al., 2016](#))
- (xi) **Kaggle:** Kaggle is an online community of data scientists providing datasets on a wide range of topics. We used kaggle to extend the dataset we used for training our model.

Mobile Application Design.

System Design.

System design includes techniques such as Data Flow Diagrams (DFDs) used to show the flow of data in the system.

Use Case diagram.

A Use case diagram was used to develop a better understanding of the requirements; It gave us a view of the system components and entities of the system design. The use case identifies the type of interaction and the actor, in particular the farmer, server and superuser of the App, involved. The use case expounds on the system in detail.

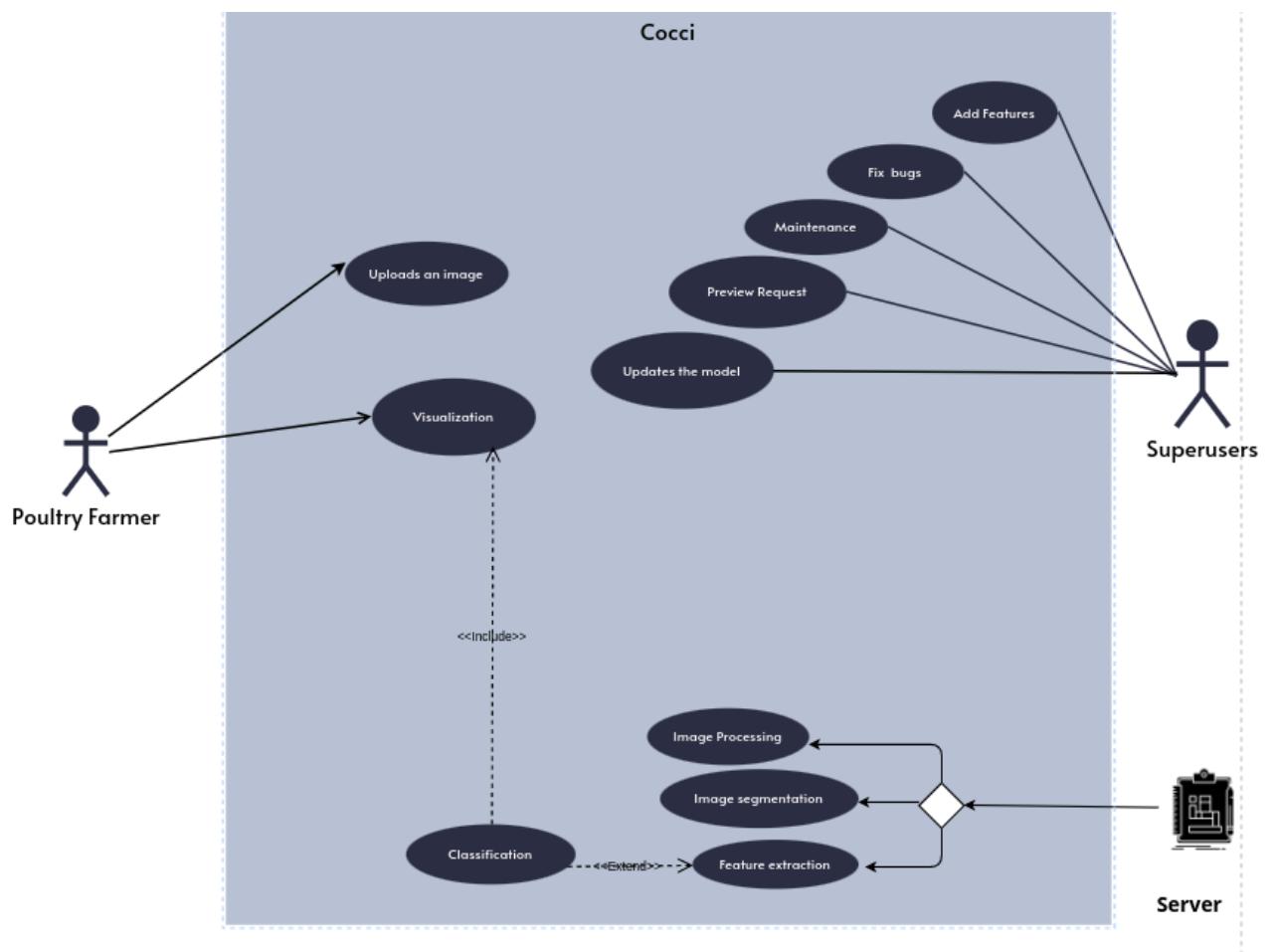


Figure 12: Usecase diagram for the Disease detection tool

Architecture design.

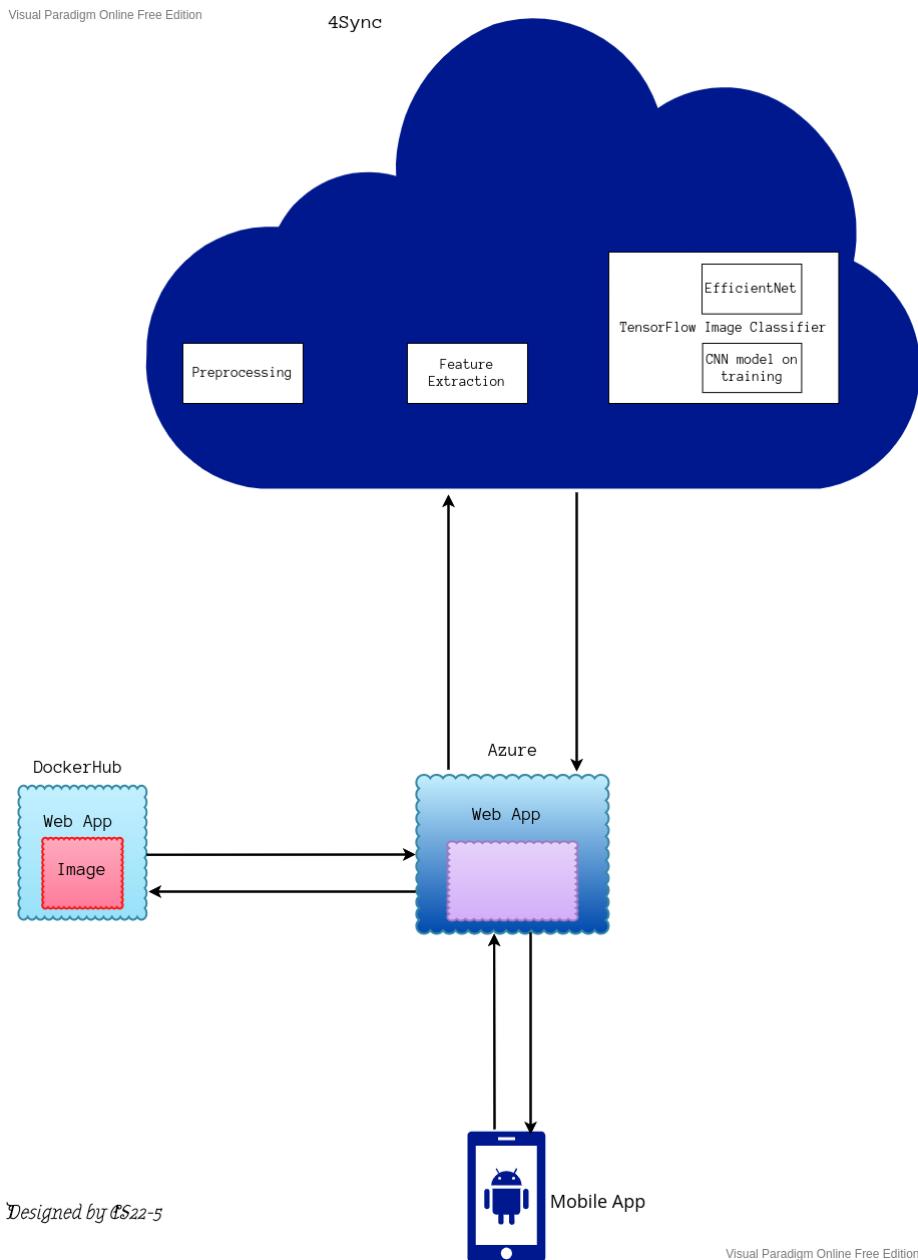


Figure 13: Architecture Design for the Disease detection system

Data flow diagram.

A data flow diagram was used to represent the system as a set of activities, each of which carries out some data transformation. It showed what kinds of data needed

to be input into and output from the system, where the data came from and go to, and where the data was stored. It also showed how the input to the process was transformed into an output.

The Stool Image is taken by a camera or uploaded from another repository; the analysis process is done by using some image processing techniques on that image thus outputting the result: Disease name and remedy.

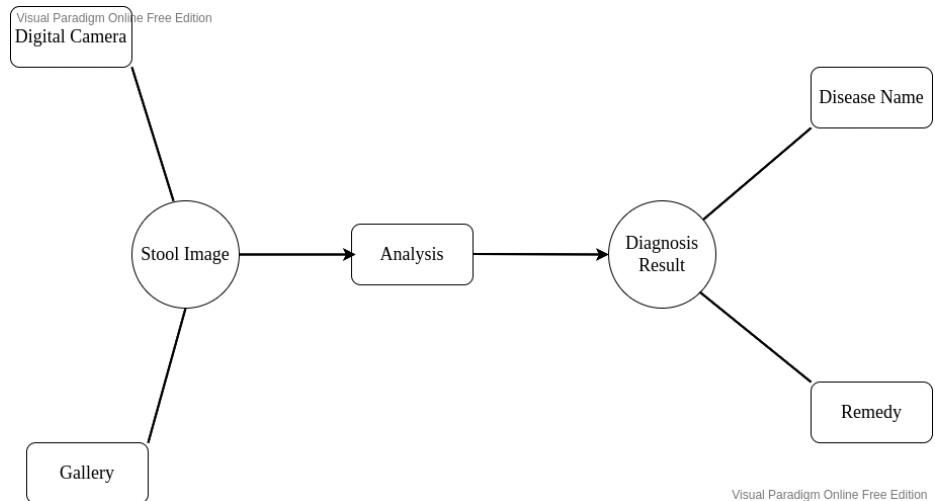


Figure 14: A Data Flow Diagram for the Disease detection system

Flow chart.

The flow chart represents the dynamic behavior of the objects and classes that have been identified as part of the system. The flow chart helped us describe the plan in order to perform the different tasks. It showed what was done when the decision was made and when to go to each process as a result. The flow chart helped us build a step-by-step picture of the processes of our system.

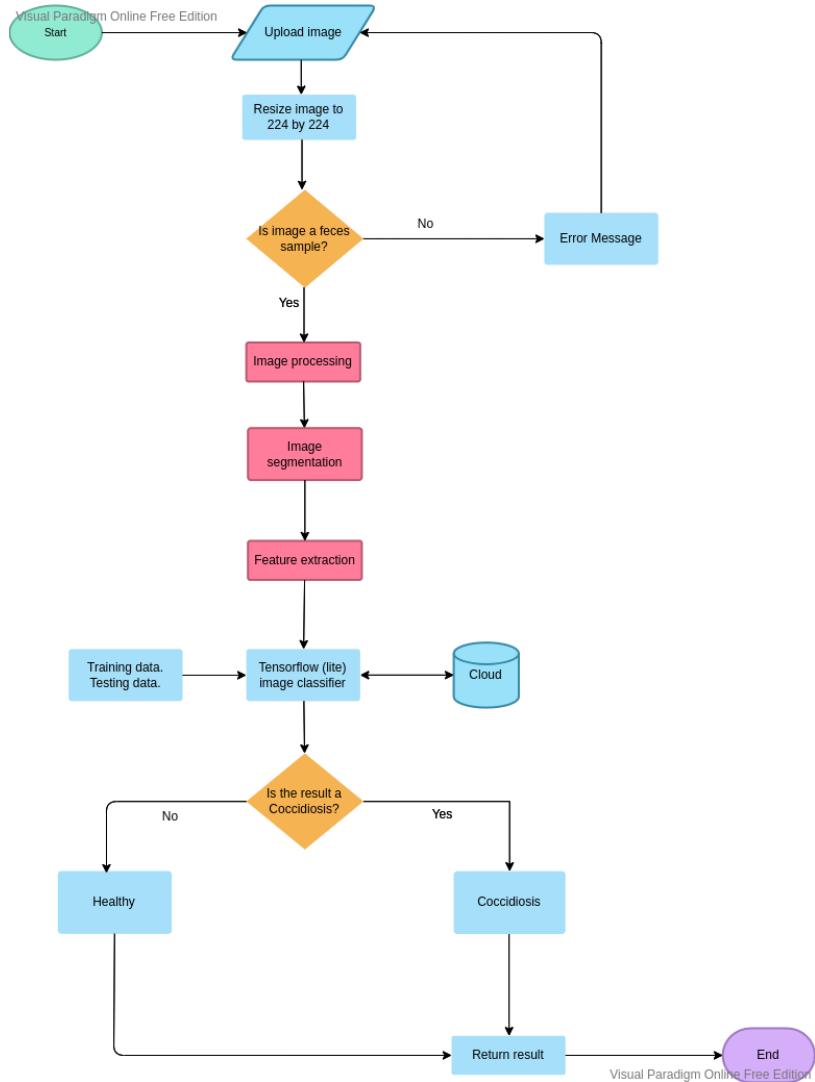


Figure 15: A flowchart for the Disease detection system

Technology decisions.

Delivering a proof of concept application to farmers who can take images of droppings and receive model predictions on the possibility of the birds having coccidiosis was the aim of this project. To achieve this, an Android application integrated with an API to a deep learning web service we created was designed. The android application was created with Flutter.

The web service which was hosted on Microsoft Azure was created using Python programming language with which the machine learning model was built with. The web service combined the Starlette.io Framework with Uvicorn ASGI Server. Starlette is a lightweight ASGI toolkit which is ideal for building asynchronous web services.

Flutter.

A UI software development kit called Flutter, developed by Google is used to create cross-platform applications from a single codebase for Android, iOS, Linux, Mac, Windows, Google Fuchsia, Web platform, and the web. We used this framework to create the android and iOS mobile applications. Flutter is easy to use because it is powerful yet simple enough as a design tool. It structures an application in terms of Stateless Widgets and Stateful Widgets which are used to make efficient code.

Docker.

Docker is a collection of platform-as-a-service products that use OS-level virtualization to deliver software in containers. Docker bundles up an application, for example in our case the ASGI Startlette web service we created using Python, into an image containing all the necessary resources it needs to run on a virtual OS. When these images are activated and run, they are termed as containers. Docker can then be used to push this image to a repository which can then be accessed and run by a cloud hosting platform like Microsoft Azure.

4Sync.

Because of the huge size of our deep learning model, we needed to upload it from Google Drive to a file sharing service that provided direct download links to the model which we can then use in our Python web service to download the model in the container for use in making predictions hence 4Sync which is a free file sharing service that provides direct links to uploaded files.

Microsoft Azure.

Azure is a cloud computing service operated by Microsoft for application management. We used this service to deploy our Python web service which made the predictions online and printed out the results both through an API with JSON response data and a web user interface. Azure was easy to start up with and provided more resources (which our web service needed) in their startup plans.

Tensorflow.

TensorFlow is a free and open-source machine learning and artificial intelligence software library. It can be used for a variety of tasks, but it is most commonly used for deep neural network training and inference. We used this for the training of the model.

Matplotlib and seaborn

Matplotlib and seaborn are plotting libraries for the Python programming language and we used them in the plotting of graphs, for example plotting the loss and accuracy graphs against the epochs for the testing and validation datasets we created which aided in the analysis and improvement of our model.

Numpy and pandas

Numpy is a Python library which offers an extensive number of functions for manipulating numbers which was very useful in the creation of our model. We also used Pandas for further data analysis to aid the creation of our model for example reading data from csv files.

System implementation.

Application features.

The following features were integrated in the Disease Diagnosis System application.

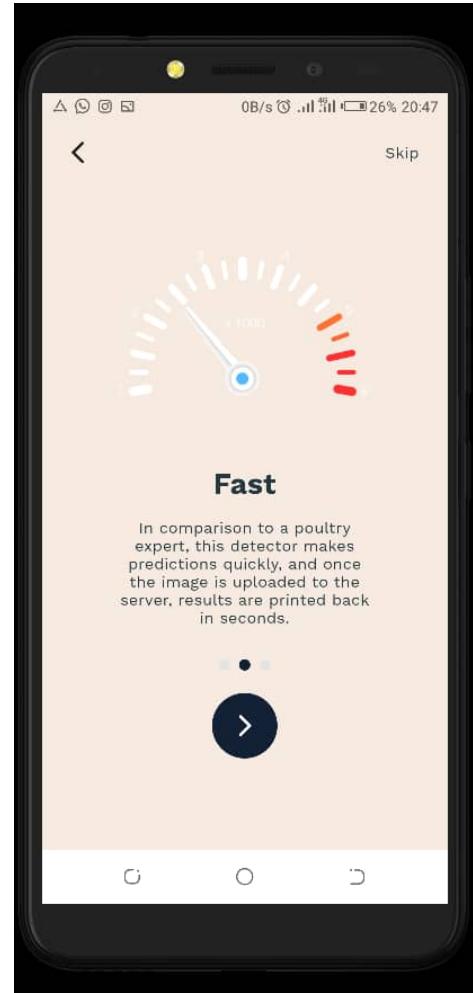
1. Usage of camera or device media to get an imageOur study proposes a different model made after finetuning the EfficientNetB6 model that belongs to the family of efficientNet.
2. Preview of the image, cropping, and sending it to online for disease detection.

3. Result page showing detected poultry disease and remedy.
4. User guide interface walk-through tutorial.

Application interfaces.



(a)



(b)



(c)

Figure 16: User guide interfaces

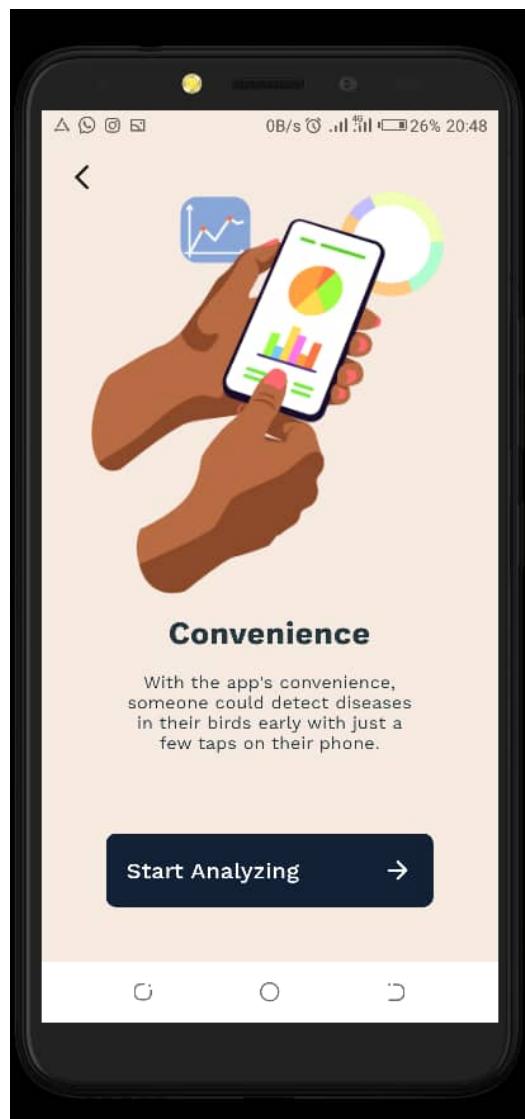
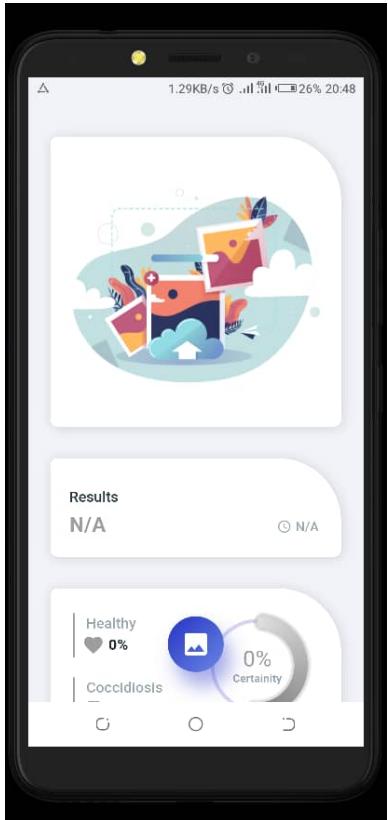
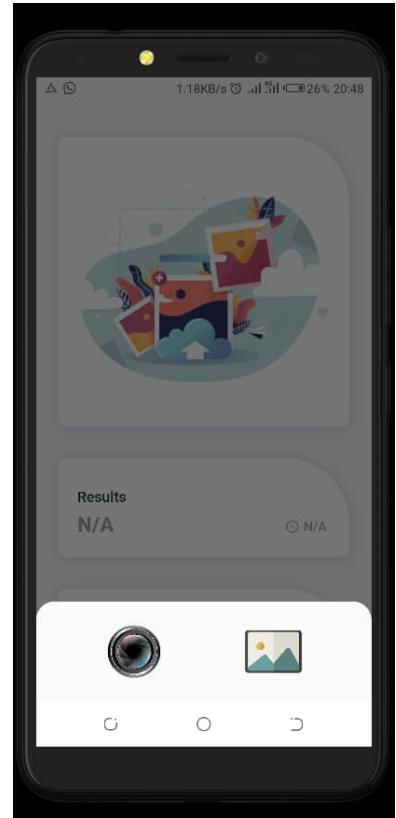


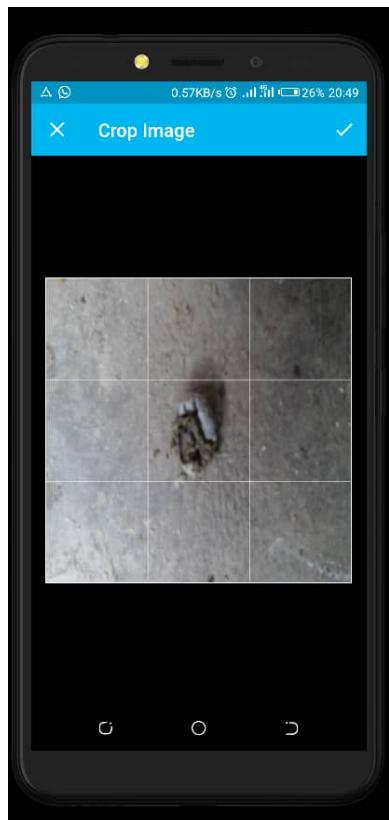
Figure 17: Snippet of Starting page



(a) Before uploading image



(b) Upload with Camera or Gallery



(c) Uploading Cropped image

Figure 18: Uploading Process

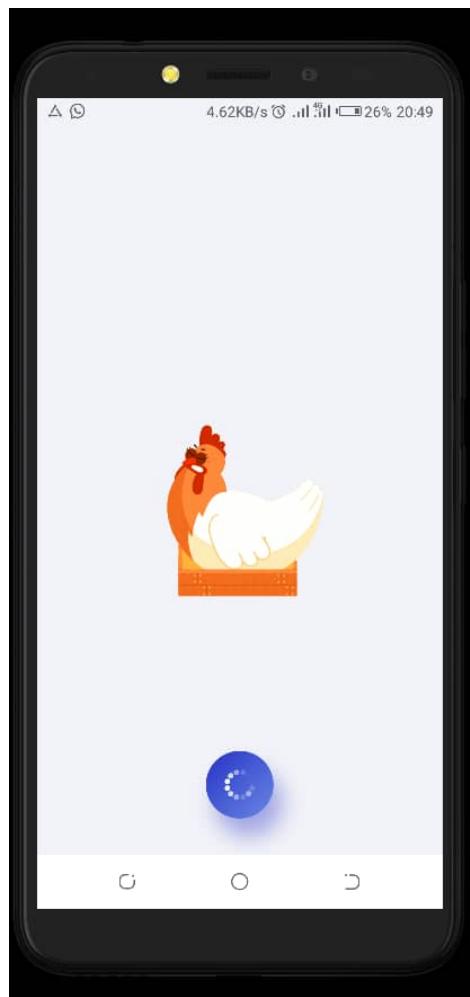
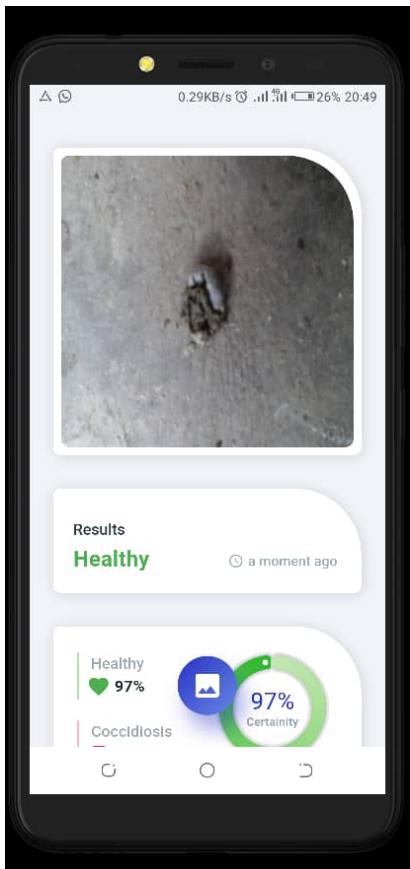
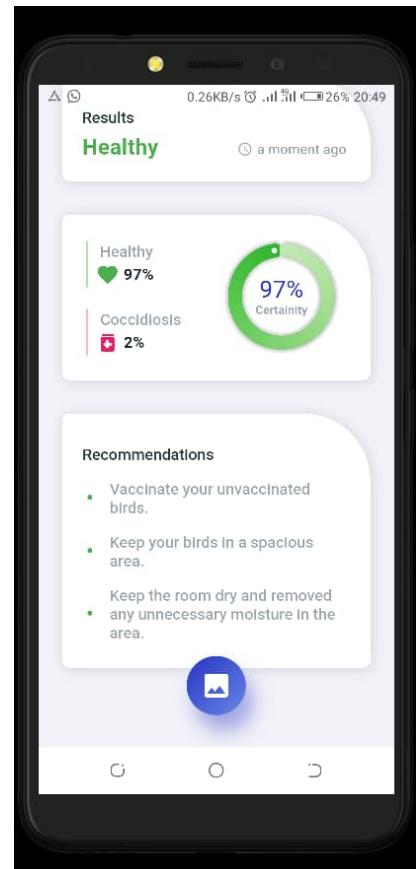


Figure 19: Snippet of Analysis process interface

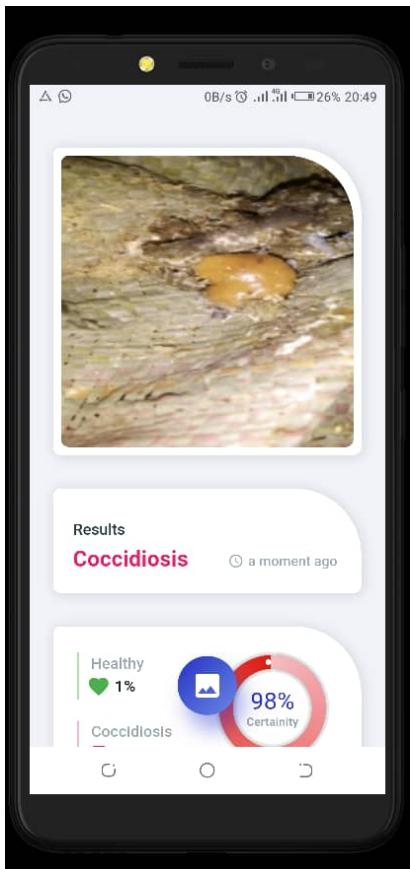


(a)

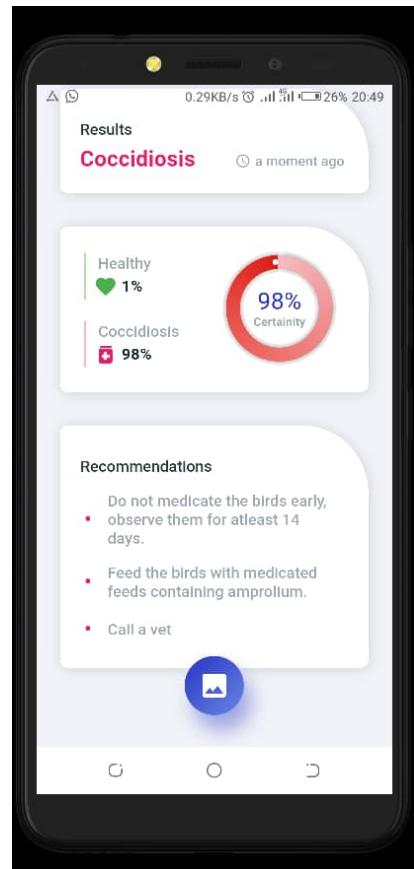


(b) Healthy result analysis

Figure 20: Analysis Results



(a) Coccidiosis result analysis



(b) Coccidiosis result analysis

Figure 21: Analysis Results 2

Conclusion and Future work.

Deep learning and Artificial Intelligence technologies performed significantly in disease detection to improve poultry health and consequentially will boost poultry farming by controlling the biotic and abiotic variables through early diagnosis.

In the study of coccidiosis, we compared the performance of the model developed using K-Nearest Neighbor vs the one with CNN with EfficientNetB6 as a pretrained model. The KNN model produced an accuracy of 86% and the CNN model produced an accuracy of close to 100%.

The CNN model took into consideration the colors of the image droppings and their spread which are crucial factors in coccidiosis identification while the KNN worked better at getting features such as looseness or spread of the droppings. The performance of the CNN was highly attributed to the EfficientNetB6 and was trained on our dataset. The techniques such as augmentation, regularizers, and dropout layers also played a crucial role in reducing over-fitting.

Conclusion.

Limitations of our study.

1. The model cannot differentiate between feaces samples and images of other objects.
2. The model was only trained on images for broilers and might not be efficient for other breeds of poultry.
3. The application was developed and tested on mainly android phones.
4. The model takes into consideration the colors of the image sample therefore accuracy could be affected by the surrounding of the image.

5. Clear pictures produce the best accuracy hence good cameras are needed.

Future work.

In future, this research would be extended to:

1. Add more classes for classification.
2. Add Veterinary doctors' contacts.
3. Improve the model with more data(sample images) for other poultry birds like turkeys and guinea fowls.

Appendix and References.

Budget

Table 1: Proposed Budget

Item	Amount (Shs)
Transport	80,000
Food	100,000
Chicken	20,000
Labor	50,000
Feeds for testing	20,000
Miscellaneous	50,000
Total	320,000

Disease detection in chicken
Read-only view, generated on 09 May 2022

 Instagantt

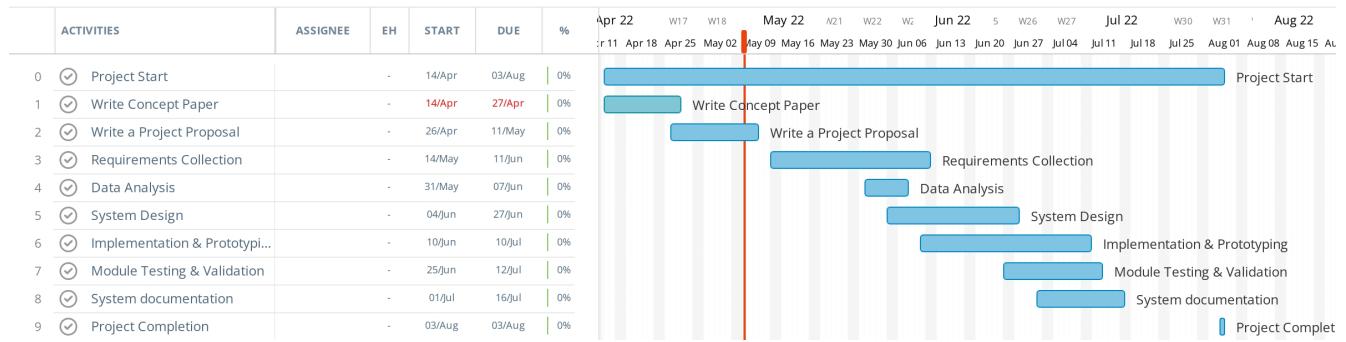


Figure 22: Gantt Chart

Time Frame

Table 2: Proposed time for project development

Task	Duration (Days)	Start Dates
Project start		14 th /April / 2022
Write a concept paper	12	14 th /April / 2022
Write a project proposal	15	26 th /April / 2022
Requirements collection	21	14 th /May / 2022
Data Analysis	7	31 th /May/ 2022
System design	23	4 th /June / 2022
Implementation & Prototyping	30	10 th /June / 2022
Module Testing & Validation	24	25 th /June / 2022
System documentation	15	1 th /July / 2022
Project completion	10	3 th /August / 2022

Github Repository

<https://github.com/Computer-Science-GroupWork/coccidiosis-detector>

Bibliography

- Alemneh, T. and Getabalew, M. (2019). Exotic chicken production performance, status and challenges in ethiopia. *International Journal of Veterinary Science and Research*, 5(2):039–045.
- Arpteg, A., Brinne, B., Crnkovic-Friis, L., and Bosch, J. (2018). Software engineering challenges of deep learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 50–59. IEEE.
- Astill, J., Dara, R. A., Fraser, E. D., and Sharif, S. (2018). Detecting and predicting emerging disease in poultry with the implementation of new technologies and big data: A focus on avian influenza virus. *Frontiers in veterinary science*, 5:263.
- Björk, T. (2021). Exploring change point detection in network equipment logs.
- Boaduo, N. A.-P. (2011). Systematic analysis and interpretation of collected data for a research study: A practical methodological framework for writing research report. *Educational Research and Reviews*, 6(2):140–146.
- Brownlee, J. (2016). *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery.
- Burnstein, I., Suwanassart, T., and Carlson, R. (1996). Developing a testing maturity model for software test process evaluation and improvement. In *Proceedings International Test Conference 1996. Test and Design Validity*, pages 581–589. IEEE.
- Byaruhanga, J., Tayebwa, D. S., Eneku, W., Afayo, M., Mutebi, F., Ndyanabo, S., Kakooza, S., Okwee-Acai, J., Tweyongyere, R., Wampande, E. M., et al. (2017). Retrospective study on cattle and poultry diseases in uganda. *International journal of veterinary science and medicine*, 5(2):168–174.
- Chatterjee, A. and Oza, A. (2017). Agriculture insurance.
- Damerow, G. (2016). *The chicken health handbook: a complete guide to maximizing flock health and dealing with disease*. Storey Publishing.

- Dennis, A., Wixom, B., and Tegarden, D. (2015). *Systems analysis and design: An object-oriented approach with UML*. John wiley & sons.
- Der, B. S., Glassey, E., Bartley, B. A., Enghuus, C., Goodman, D. B., Gordon, D. B., Voigt, C. A., and Gorochowski, T. E. (2017). Dnaphotlib: programmable visualization of genetic designs and associated data. *ACS synthetic biology*, 6(7):1115–1119.
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, 145:311–318.
- Fylaktopoulos, G., Goumas, G., Skolarikis, M., Sotiropoulos, A., and Maglogiannis, I. (2016). An overview of platforms for cloud based development. *SpringerPlus*, 5(1):1–13.
- Gale, P., Drew, T., Phipps, L., David, G., and Wooldridge, M. (2009). The effect of climate change on the occurrence and prevalence of livestock diseases in great britain: a review. *Journal of applied microbiology*, 106(5):1409–1423.
- Gao, K., Mei, G., Piccialli, F., Cuomo, S., Tu, J., and Huo, Z. (2020). Julia language in machine learning: Algorithms, applications, and open issues. *Computer Science Review*, 37:100254.
- Hemalatha, S. and Maheswaran, R. (2014). Recognition of poultry disease in real time using extreme learning machine. In *International Conference of Disciplinary Research in Engineering and Technology (CIDRET2014)*, pages 44–50.
- Hepworth, P. J., Nefedov, A. V., Muchnik, I. B., and Morgan, K. L. (2012). Broiler chickens can benefit from machine learning: support vector machine analysis of observational epidemiological data. *Journal of the Royal Society Interface*, 9(73):1934–1942.
- Hope, T., Resheff, Y. S., and Lieder, I. (2017). *Learning tensorflow: A guide to building deep learning systems.* O'Reilly Media, Inc.”.
- Karthikeyan, R., Satheesbabu, S., and Gokulakrishnan, P. (2021). Machine learning based student performance analysis system. *Information Technology in Industry*, 9(1):1174–1181.
- Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109.
- Mbelwa, H., Machuve, D., and Mbelwa, J. (2021). Deep convolutional neural network for chicken diseases detection.

- Myszczynska, M. A., Ojamies, P. N., Lacoste, A., Neil, D., Saffari, A., Mead, R., Hautbergue, G. M., Holbrook, J. D., and Ferraiuolo, L. (2020). Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. *Nature Reviews Neurology*, 16(8):440–456.
- Nagpal, A. and Gabrani, G. (2019). Python for data analytics, scientific and technical applications. In *2019 Amity international conference on artificial intelligence (AICAI)*, pages 140–145. IEEE.
- Nozari, H., Sadeghi, M. E., et al. (2021). Artificial intelligence and machine learning for real-world problems (a survey). *International Journal of Innovation in Engineering*, 1(3):38–47.
- Olsson, M. (2020). A comparison of performance and looks between flutter and native applications: When to prefer flutter over native in mobile application development.
- Petersen, K. and Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15(6):654–693.
- Ribeiro, A., Cardoso, A., Marques, J. A. S., and Simões, N. E. (2019). Web interface for river hydrodynamics simulation. In *2019 5th Experiment International Conference (exp. at'19)*, pages 278–279. IEEE.
- Rodriguez, A., Tabassum, A., Cui, J., Xie, J., Ho, J., Agarwal, P., Adhikari, B., and Prakash, B. A. (2021). Deepcovid: An operational deep learning-driven framework for explainable real-time covid-19 forecasting. *medRxiv*, pages 2020–09.
- Sadeghi, M., Banakar, A., Khazaee, M., and Soleimani, M. (2015). An intelligent procedure for the detection and classification of chickens infected by clostridium perfringens based on their vocalization. *Brazilian Journal of Poultry Science*, 17:537–544.
- Shinyekwa, I., Kiiza, J., Hisali, E., and Obwona, M. (2016). The evolution of industry in uganda. *Manufacturing Transformation*, page 191.
- Wang, J., Shen, M., Liu, L., Xu, Y., and Okinda, C. (2019). Recognition and classification of broiler droppings based on deep convolutional neural network. *Journal of Sensors*, 2019.
- Zhuang, X., Bi, M., Guo, J., Wu, S., and Zhang, T. (2018). Development of an early warning algorithm to detect sick broilers. *Computers and Electronics in Agriculture*, 144:102–113.