2022

A Snake classification model for snakebite envenoming management.

Mabinda, Mariam
School of Computing and Engineering Sciences
Strathmore University

A Snake Classification Model for Snakebite Envenoming Management

Mariam Mabinda

136553

Masters of Science in Information Technology

2022

A Snake Classification Model for Snakebite Envenoming Management

Mariam Mabinda

136553

Submitted in partial fulfillment of the requirements for the Degree of Masters of Science in Information Technology at Strathmore University.

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

October, 2022

Declaration and Approval

Declaration

I, Mariam Mabinda declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Mariam Mabinda

Sign: ______________________

Date: 13/7/2022

Approval

The thesis of Mariamu Mabinda was reviewed and approved by the following:

Dr. Dickson Owuor,
Senior Lecturer, School of Computing & Engineering Sciences,
Strathmore University.

Dr. Julius Butime,
Dean, School of Computing & Engineering Sciences,
Strathmore University.

Dr. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University.

Abstract

Snakebite envenoming is a potentially life-threatening disease caused by the injection of toxins through a bite or venom sprayed into the victim's eyes by certain venomous snake species. WHO program dubbed Neglected Tropical Disease Program (NTD) of 2019 indicated that about 5.4 million snake bites occur each year, resulting in 1.8 to 2.7 million cases of envenoming. Of this about 81,000–138,000 deaths occur and approximately 400,000 people are permanently disabled annually. Kenya is approximated to have more than 15,000 bites annually. Correct identification of the snake species in question plays a critical role in the proper administration of the right first aid and suitable prescription of the anti-venom for the patient. Currently, there is no automated method of identifying snake species using images in Kenya. The usual practice is to, kill the snake and carry it along with the patient to the hospital or to give a visual feature description of the biting snake. Also, a blood test can be done to look for the presence of toxins associated with the described snake species. The challenge however is, that the time required for test results to be out can jeopardize patients' survival depending on the type of venom injected. Furthermore, the cost associated with the test is also punitive. The ability to correctly classify a snake species is a challenging task for both humans and machines mainly because of subtle differences between different snake species and strong variety within the same species. Existing studies used a combination of feature extraction methods and deep neural networks and yielded an accuracy of 90%. These models applied Principal component analysis (PCA) and Linear discriminant analysis (LDA) as feature extractors. However, the use of the Singular Value Decomposition (SVD) algorithm was not explored despite its apparent advantage. This research study solved the classification challenge by creating a Kenyan snake species dataset and developing a snake species classification model that predicts a snake species based on the image and classifies it according to its venom toxicity. The study carried out feature reduction of the images using the SVD algorithm and passed these features as input to a deep learning model using transfer learning. The model was trained using 4521 labelled snake images via supervised transfer learning using MobileNetV2. The model was trained, validated, tested, and achieved an outstanding classification accuracy of 96 %. The model surpassed the accuracy of the existing model.

Keywords: Singular Value Decomposition, snake classification, feature reduction, transfer learning with convolutional neural networks.

Definition of Terms

Classification Model	A term that describes drawing a conclusion from an input value given for training and can predict the category for new data (Sarker et al., 2019)
Neural Networks	This consists of thousands or even millions of simple processing nodes that are densely interconnected. They are a means of machine learning in which a computer learns to perform a task by analyzing specific examples (Ian et al., 2016)
Snakebite Envenomation	Injection of snake venom by venomous snake species to a human through a bite or spraying the venom into the eyes mostly for defense purposes (WHO, 2019).
Transfer Learning	Transfer learning is the use of stored knowledge gained from a pre-trained neural network model, to train a different but related neural network model with a smaller dataset (Perera & Patel, 2019).
Data Augmentation	The process of artificially inflating datasets using a series of transformations such as rotation, cropping, flipping (Taylor and Nitschke, 2017).
Deep Learning	This is a subfield of machine learning concerned with the development of neural networks that are deeper or consist of several convolutions. The deeper learning enables them to learn and model complex scenarios (Nielsen, 2015).

Table of Contents

List of Abbreviations

CEDD	Color and Edge Directivity Descriptor
CNN	Convolution Neural Networks
CPU	Central Processing Unit
ERD	Entity Relationship Diagram
GPU	Graphics Processing Unit
ICT	Information Communication Technology.
ID3	Iterative Dichotomiser 3
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
MOH	Ministry of Health
NLP	Natural Language Processing
NTD	Neglected Tropical Disease
OOAD	Object-Oriented Analysis and Design
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TFLite	Tensor Flow Lite
UML	Unified Modeling Language
VGG16	Visual Geometry Group 16
WHO	World Health Organisation

List of Tables

List of Figures

Acknowledgments

First and foremost, I thank the Lord God Almighty for giving me the strength, time, resources, and good health as I pursued this course. It is by His grace and mercies that I have completed this research project. I thank my immediate supervisor, Dr. Dickson Ouwor for his relentless efforts, encouragement, and support to see that this research is completed successfully. Sincere gratitude to Dr. Vincent Omwenga for his guidance, encouragement, input, and support during the class seminars and throughout the research process. I also want to thank Dr. Bernard Shibwabo who guided me in the early stages of my research work. I would like to acknowledge and thank my classmates, MSIS class of 2022, and MSIT class of 2022 for the encouragement, knowledge sharing, and all the light moments we shared. The journey was worthwhile and the load was made lighter with your support. I acknowledge and appreciate you all.

Dedication

It is with genuine gratitude and warm regard that i dedicate this thesis to my family. A special dedication to my husband Eng. Justus Aufridus Otwani, who has been a constant source of support and encouragement always. I am blessed and thankful to have you as my friend and partner. To my sons Schneider Ikol and Schourn Ejakait, thank you for the support, encouragement, perseverance, and understanding as I was undertaking this research. You are my greatest cheerleaders and may the good Lord bless you all. A special dedication to my mum Franciscah Mwikali for being the lady I always want to emulate and for your prayers. In loving memory of my father, Albert Mabinda, who always pushed me to be the best version of me, as you would have said, "We did it!" On to the next one. To my sisters and brothers thanks for cheering me on and being there always when I needed you.

Chapter 1: Introduction

1.1 Background

Snakes are limbless carnivorous reptiles of the suborder Serpents (MOH-Kenya, 2019). They are cold-blooded, amniotic reptiles, surrounded in sepals, just like other scaled reptiles (Progga et al., 2021). They balance the ecosystems since they act as natural pest controllers. They are generally categorized as non-venomous and venomous. Non-venomous have no venom or their venom has no adverse effects on humans. Venomous ones produce venom that they use mainly to subdue or kill prey rather than for self-defense (Abdurrazaq et al., 2019).

Rajabizadeh and Rezghi, (2021) indicated that, out of 3848 known species of snakes, around 800 species are venomous, among which only about 50 species are deadly or fatal to humans. In Kenya, out of the 140 known snake species, 29 are considered venomous. Of the 29 venomous species, only 13 are of medical importance as their venoms cause injury or death in extreme cases, and only 9 of these account for bites that require medical attention (MOH-Kenya, 2019).

Snakebite envenoming is a medical emergency that results from contact with snake venoms, which are injected when snake bites or when venom is sprayed into the victim's eyes (WHO, 2019). Krishnan, (2020) mentioned that snakebite envenoming is the second most deadly neglected tropical disease. In 2017, World Health Organisation (WHO) recognized snakebite envenoming as a Neglected Tropical Disease (NTD) that kills about 1,410 to 137,880 people annually (Abdurrazaq et al., 2019). Furthermore, research indicates that about 5.4 million people suffer from snake bites annually, causing 1.8 to 2.7 million cases of envenoming (Vasmatkar et al., 2020).

In Kenya alone, 15,000 bites are approximated annually (MOH-Kenya, 2019). Progga et al., (2021) indicated that the deaths are mainly attributed to a lack of expert knowledge by the victims, on which type of snake one has encountered. Consequently, wrong identification of snake species can lead to wrong antidote administration which can in turn cause an adverse medical reaction. Therefore, easy and correct identification of the biting snake species is a crucial part of the snakebite envenoming treatment process. When the snake in question is correctly classified based on the type of snake or the venom it contains, medical practitioners have an easy time administering the suitable anti-venom and carrying out correct first aid (Abdurrazaq et al., 2019).

However, Rajabizadeh and Rezghi, (2021) noted that it is difficult to classify snake species unless one is guided by expert knowledge. Therefore, there is a need to apply machine learning techniques to make use of this expert knowledge to help the public without access to this knowledge classify snakes according to their species and based on the venom type that they contain. With such a model, snake species classification is automated, knowledge of the venom category present in a snake species is made instant, the treatment process becomes efficient, and ultimately more lives are saved.

In the recent past, various models have been built to aid in snake species classification. These models made use of image processing techniques to come up with a classification. The images are captured from a camera device and run through algorithms to classify the snake (Joseph, 2021). The use of raw data as direct input to a classifier yielded low accuracy because of the varied features present on the snakes even of the same species (Rajabizadeh & Rezghi, 2021). Progga et al., (2021) proposed the use of feature extraction algorithms together with deep neural networks like CNN as a solution to this.

Nath et al., (2014) mentioned that feature extraction is an important step in image classification. It is the identification, extraction, and analysis of the most relevant image features of an image and assigning it to a label (Ahmed Medjahed, 2015). Generally, snakes have varied dorsal color patterns, elongated and flexible bodies, and usually pose in varied ways causing deformation of the body making acquiring features quite challenging. To overcome the challenge of wide variations in the body pose of snakes in the images, a feature extraction method was used to extract features from images for indexing and retrieval which was later used for classification (Rajabizadeh & Rezghi, 2021).

Dimensionality reduction approaches such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA)] were employed by Rajabizadeh and Rezghi, (2021) as the feature reduction technique in conjunction with transfer learning such as MobileNetV2 which yielded an accuracy of 93.16%. On the other hand, Mateen et al., (2019) used Singular Value Decomposition (SVD) in image classification as a feature extractor to decrease dimensionality based on reliability. The features that were not reliable for the classification were dropped and consequently the dimension of the image was reduced making classification easy and faster. Despite this apparent advantage present on SVD, it has not yet been exploited in studies classifying snakes based on their images.

It is on the same premise that this research is built on. The research used SVD to reduce the feature of the snake on an image. The acquired vectors were fed as an input to train our deep neural network using MobileNetV2 transfer learning. The main advantage of SVD is, the ability to rapidly reduced the number of parameters as well as reduced the number of computations in complex data like that of snake image (Mateen et al., 2019). Moreover, since the snake image dataset was relatively small, MobileNetV2 transfer learning was considered because of its ability to train a deep neural network with little data by applying weights of another model that achieved better results. Snake features are not easy to distinguish and describe hence difficulties in classifying them accurately and consequently administration of treatment (Abdurrazaq et al., 2019) (Abayaratne et al., 2019).

1.2 Problem Statement

Snake species are best classified based on their venom toxicity profile. There are five broad venoms toxicity profile categories. These include neurotoxic, hemotoxic, cytotoxic, cardio-toxic, and non-toxic. Neurotoxic destroys the nerves, hemotoxic causes abnormal blood changes, cytotoxic affects the muscle tissue, cardio-toxic inhibits the normal functioning of the heart, and lastly nontoxic are the ones that don't have toxins that affect human beings (MOH-Kenya, 2019). Administration of the antidote and first aid measures are different for each envenomation (Abdurrazaq et al., 2019). Correct identification of the snake species is a crucial step for correct and proper first aid and treatment of snake bites (Abayaratne et al., 2019).

Accurate classification of snake species is a challenging task for both humans and machines mainly because of subtle differences between different snake species and strong variety within the same species. Some distinguishing features are so subtle to distinguish in a quick flash or poor lighting and may get deformed or simply have wear and tear. This makes it difficult to make correct identification in the field. As such, only a detailed examination of the captured specimen image can assist in making a positive identification (Rajabizadeh & Rezghi, 2021).

Image classification, powered by advances in deep neural networks, is now performing many vision-related tasks faster and better than humans mainly due to convolutional neural networks (Devikar, 2016). Currently, dimensionality reduction methods are also used to perform feature reduction on images with wide varied features. PCA and LDA coupled with transfer learning of MobileNetV2 yielded an accuracy of 93.3%. SVD is however not exploited despite its apparent advantage of reducing the number of parameters besides reducing the number of computations in the complex image data while maintaining the reliability of the compressed image (Mateen et al., 2019)**.** SVD is based on low-rank matrix completion using singular value truncation and thresholding. Since the image is decomposed using singular value decomposition (SVD) to obtain a low rank of image data, which is approximated in compressed form which is later on exploited to retrieve visual quality of the compressed image using singular value thresholding algorithm the reliability of the compressed image is not compromised. In machine learning, having abundant and correct data is more important than having correct algorithms. However, in instances where the dataset is relatively small, then transfer learning is applied. As such, this research focused on solving the classification of snake species by first creating an image dataset of Kenyan snake species. The features of the image were then reduced using Singular Value Decomposition (SVD) and saved to a matrix. The matrix with the reduced features of the images was then used to train a convolutional neural network via supervised transfer learning using MobileNetV2.

1.3 Objectives

1.3.1 General Objective

This research aims to build a snake species classification model for snakebite envenoming management.

1.3.2 Specific Objectives

(i) To investigate challenges associated with managing snakebite envenomation.

(ii) To analyze machine learning techniques that support image classification.

(iii) To develop an image classification model that can classify snake species.

(iv) To validate the image classification model.

(v) To develop a mobile application to show case the working of the model.

1.4 Research Questions

(i) What are the challenges with snakebite envenomation management?

(ii) What are the machine learning techniques that support image classification?

(iii) How can an image classification model that identifies snake species be developed?

(iv) How can an image classification model be validated?

(v) How can the mobile application be developed?

1.5 Justification

Firstly, the model from the research can assist the victims of snakebite envenoming to correctly identify and classify the snake encountered and seek appropriate first aid.

Secondly, the public can benefit from this research by being able to correctly classify the snake species using the mobile application and administer the correct first aid to the victims.

Thirdly, health workers can benefit from the research by being able to know the snake species the patient has encountered and the type of venom it contains and therefore administer the correct antidote based on the image collected by the patient. This can enhance service delivery in hospitals and consequently reduce deaths and disability caused by delayed envenoming management.

Lastly, the study has contributed to a Kenyan dataset that can be used by academicians in the future for improving the model and carrying out research using the dataset. The finding of this study can be of importance to academicians and future researchers by contributing to the local and international body of literature. It can add value to the existing literature by providing new techniques based on applying ICTs in snakebite envenomation management.

1.6 Scope and Limitation

The research is limited to classifying snake species using captured image and provides the common name of the snake in English and their scientific name and indicating their venom toxicity category. The research will further use a subset of five venom toxicity categories of snake species, which includes the eighteen endemic snake species in Kenya, for the proof of concept. For one to use the system, one must have a devices that has camera capability or a smart phone with clear camera. Also, there is need for initial download and installation of the application that uses the model.

Chapter 2: Literature Review

2.1 Introduction

This chapter briefly introduced snakebite envenomation management, the challenges associated with it, and how technology has and can play a role in snakebite envenomation management. A review of the image classification and common methods currently in use was done. A comparison of different machine learning techniques that support image classification was conducted. This helped the researcher identify the suitable image classification model for snake classification. Finally, the researcher presented a conceptual framework.

2.2 Snakebite Envenomation Overview

Krishnan, (2020) mentioned that snakebite envenoming is the second most deadly neglected tropical disease. It is considered a medical emergency that results from contact with the venom of a venomous snake being injected either through a snake bite or by venom being sprayed into the victim's eyes (WHO, 2019). Research indicates that about 5.4 million people endure snakebites annually, causing 1.8 to 2.7 million cases of envenoming (Vasmatkar et al., 2020). In Kenya alone, 15,000 bites are approximated annually, and more go unreported (MOH-Kenya, 2019).

Community awareness campaigns focusing on the prevention of snakebite such as cutting grass, and decluttering thickets have largely been done by most concerned organizations but still, the envenomation rate is on the rise, especially in low-income populations (Pucca et al., 2020). Snakebite envenoming is largely reported by farmers, pastoralists, and fishermen among others. This is largely attributed to the fact that their normal duties revolve around the same habitat as the snake species. As a result, snakebite envenoming was considered an occupational disease (Bhargava et al., 2020).

Pucca et al., (2020) indicated that since most of the snakebite envenoming victims come from poor backgrounds, they normally shy away from seeking medical attention in health facilities near them. This accounts for the huge unrecorded number of snakebite envenoming. Instead, the victims seek attention from traditional healers whom they believe to be affordable and knowledgeable in matters of snakebite envenoming management (Bhargava et al., 2020).

Russell et al., (2021) indicated the importance of first aid management of the snakebite envenoming that occurs in the fields. He argued that correct and appropriate first aid done in the field contributes largely to the survival of the victim, especially when dealing with a fast-acting venom. First aid is usually done solely to keep the victim alive until they get medical attention. After correct administration of the first aid, the victim should be taken to a health facility for treatment or treated traditionally by herbalists or family members. The management of the snakebite envenoming is normally guided by correct identification and classification of the biting snake.

2.2.1 Traditional methods of Identifying snake

Bolon et al., (2020) mentioned that the most important step in snakebite envenoming management, whether in a health facility or by a traditional herbalist, is the correct identification of the snake and classifying it based on the right venom toxicity category. However, this is normally a challenging task for most victims because of a lack of expert knowledge (Ahmed Medjahed, 2015). Different methods have been used by the victims to identify the biting snake. One of the common methods used in the identification of the snake species is killing and capturing the biting snake and taking it to a health facility alongside the victim. This is normally done by the victims to assist health workers to identify the correct anti-venom in the shortest time possible. However, WHO, (2019) snakebite management guidelines stipulate that it is not appropriate for anyone to handle snakes whether dead or alive not unless you are an expert in that area. This is because even a dead snake can cause envenomation. Aside from that if the snake is not completely dead, it can strike and cause more harm to the victim or people nearby.

Another commonly used method in snake identification is a verbal description of the biting snake by either the victim or bystanders (Bolon et al., 2020). This involves describing the visual features of the biting snake e.g., color, size, pace, head shape, etc. However, victims are usually in a state of panic and confusion and might not describe the snake encountered well and exhaustively (Young, 2019). Aside from that, the description might be varied from one person to the other as much as they were all present at sight. Also, snakes from the same species might have varied visual features and of varied lengths depending on gender and age (Vasmatkar et al., 2020).

Biting snake identification can also be achieved by observing signs and symptoms the patient portrays and associating them with a particular type of species and the toxicity of the venom they have (Russell et al., 2021). Different venom toxicity has different signs and symptoms (Muguti, 2003). Cardiotoxic inhibits the heart's normal functioning, neurotoxic venom destroys the nerves, hemotoxic causes abnormal blood changes, cytotoxic affects the muscle tissue and nontoxic have little to no effect on human beings (Nag & Karforma, 2016). The drawback of this method is, that victims might exhibit signs e.g., shortness of breath due to panic and fear and not necessarily because of the venom injected. This will in turn give a wrong identification of snakes using symptoms exhibited by the victim.

Nag and Karforma, (2016) stated that testing the blood samples of the victim for the presence of venom antigens and associating it with the snakes that have certain venom toxicity found is another way of identifying and classifying the biting snake species.

Thereafter administration of the anti-venom is done based on the venom antigen found (Bolon et al., 2020). This is an accurate and more precise method. The only challenge with this method is that it can only be done in a health facility and requires laboratory equipment and expert knowledge. Aside from that, the cost associated with it can be punitive considering the persons affected by snakebite are generally of poor social-economic status. Furthermore, the time it takes for laboratory results to be out might not be favourable for a fast-acting venom case hence minimizing the survival chance of the victim.

Rajabizadeh and Rezghi, (2021) suggested that the use of pictures of the biting snake taken by the victim or bystanders can be used to classify the snake. The picture can then be given to the person with expert knowledge for correct analysis and classification. Once the herpetologist, correctly identifies and classifies the snake, a suggestion of the antidote to be administered is given. In the recent past, machine learning techniques have been used to classify snake species based on their image. The major challenge however is that the accuracy of the built models has been low due to the fact that snakes have varied features that are present even in the same species.

2.2.2 Challenges of Snakebite Envenomation Management

Snakebite envenomation management is the process of managing the venom injected into a victim by the snake (MOH-Kenya, 2019). This is normally done by the health officer, a traditional herbalist, family members, or any person present at the site. Bhargava et al., (2020) reinstate the importance of the first aid done at the site to increase the victim's survival chance and minimize the severity of the disability caused by envenomation. The type of snake identified, and the venom toxicity it contains, dictates the type of first aid to be carried (Bolon et al., 2020). The majority of the public and the victims lack the expert knowledge to correctly classify the snake species and predict its venom toxicity (Rajabizadeh & Rezghi, 2021). As such, victims end up receiving the wrong first aid which affects the treatment process (Bhargava et al., 2020). Dissemination of expert knowledge easily and conveniently is needed to increase the survival chances of the victims.

Further, Rusli et al., (2019) mentioned that when handling snakebite envenoming, time becomes of the essence when dealing with snake species that have fast-acting venom (Muguti, 2003). First and correct treatment becomes critical to victims' survival. Also, considering most snakebite envenoming victims come from poor socioeconomic status, they normally shy away from getting treatment at the health facility and instead opt to try traditional ways of treatment (Bolon et al., 2020). Some of these methods end up making the wound area infected and can lead to permanent disability in the long run.

Bhargava et al., (2020) and Ameade et al., (2021) indicated that most health workers have little knowledge about snakebite envenoming and the appropriate antidote to administer to patients in case of snake envenoming emergency. Bolon et al., (2020) indicated that wrong antidote administration can cause serve allergic reactions. As such, having correct knowledge about the snake biting species and its venom toxicity content becomes important to the treatment of the victim. Considering the challenges mentioned above, the use of technology in snakebite envenomation management has been researched exhaustively and models and methods have been proposed.

2.3 Application of Technology in Snakebite Envenomation Management

Technology has been applied in various ways to solve different challenges in different sectors. In snakebite envenomation, technology has been used to address the two most important steps to envenoming management i.e., classification of the biting snake species and the dissemination of the right information to aid in treatment and first aid. Some of the methods include using machine learning in natural language processing to classify snakes based on victims' descriptions and use of image classifiers to classify snake species based on their images.

2.3.1Use Natural Language Processing (NLP)

Rusli et al., (2019) developed an intelligent system that predicted the type of the snake by taking into account the description of the biting snake in the unstructured text, using natural language processing. Common perceptions and words used by several diverse groups of people to describe many different snake species were analyzed. The text was then pre-processed, relevant keywords extracted based on their weight in the context and later used as features during classification by machine learning to learn and predict the species of snakes being described. The major drawback of this method is that it's dependent on visual descriptions of the snake and some snake species have varied different features even within the same family based on their age, location, and sex.

Niteesh. et al., (2021) on the other hand proposed a machine learning model that used the ID3 algorithm and the snakes' dataset which was embedded into the model, and the model was trained with the snakes' dataset. An input query to the trained model needed to include a description of the snake. The victim needed to mention the head shape, size of the snake, movement, pattern color on the skin, and color of the snake, then the model predicts the snake type so that the victim can take proper first aid treatment. A major drawback of this model is reliant on the ability of the victim to see and describe the feature of the biting snake correctly. This is usually a challenge since most victims are usually in a panic state after the snake bite and might be keen to correctly identify the biting snake or have a vivid memory of what they saw.

2.3.2 Use of Image Classifier

Image classification is a sub-group of computer vision that categorizes and labels groups of pixels or vectors within an image using a collection of predefined tags or categories and the visual features present. Image classification has been applied in several fields including plant disease detection, tumor identification, etc (Nyaga, 2019) (Nyaga, 2019). The image classification techniques are divided mainly into three categories. Supervised, unsupervised, and semi-supervised. Supervised classification is where model training includes showing the model the images along with their corresponding label. Unsupervised classification is where there is no labeling of the data points and the algorithm uses raw data for training. Semi-supervised classification considers the advantages of both supervised and unsupervised (Sanghvi et al., 2021).

As manual extraction of features describing the appearance of a snake is tedious, recent articles used automated feature extraction. Bloch et al., (2020) classified 22 snake species that were manually collected from Perlis Snake Park in Malaysia. Features of these collected images were extracted using Color and Edge Directivity Descriptor (CEDD). A multiple supervised machine learning algorithm, such as Naive Bayes, k-Nearest Neighbors, etc. was then used as a classifier. However, in the traditional machine learning technique, parameters in the feature extraction method are not trainable and need to be manually tuned.

Abdurrazaq et al., (2019) on the other hand classified snake images using a convolutional neural network (CNN). Three CNN architectures were evaluated using a dataset of 415 snake images from five common hazardous venomous snake species in Indonesia. Five-fold cross-validating shows that CNN is capable of classifying the snake images with high accuracy of 82%. The only drawback this method has is that CNN is a deep neural network model and a huge number of image datasets is needed to train the model. Aside from that, it is expensive both in the computational time it takes to train and test the model.

Based on the above literature, accurate identification of the features present in an image is the major objective of image classification (Nath et al., 2014). The main principle of an image classifier is to recognize the feature occurring in an image with the help of different mathematical techniques and use them to classify the image accurately. The steps for implementing the image classification process are shown in Figure 2.1 below.

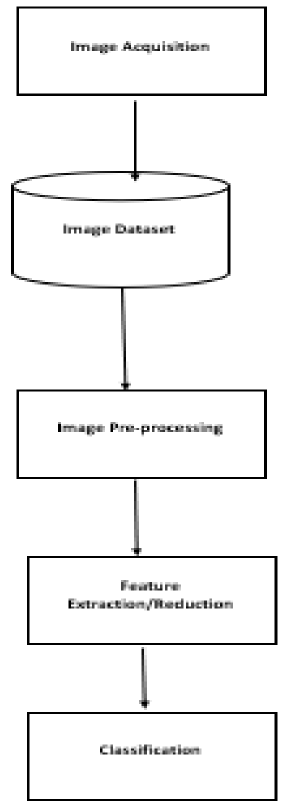

Figure 2. 1: Image Classification Process (Nath et al., 2014)

2.4 Steps of Classifying Image

2.4.1 Image Pre-Processing

Nath et al., (2014) indicated that pre-processing of image data is an essential part of the image classification process. It is normally done to reduce complexity and increase the accuracy of the applied algorithm. Aside from this, it also allows better feature extraction during classification (Vasmatkar et al., 2020). Raw data is normally characterized by unwanted distortion and noise. It is pre-processed by reshaping, resizing, and augmenting the image. Image Augmentation is the technique of creating new images from already existing images for training a deep learning model (Progga et al., 2021). This is normally done to avoid overfitting the model and to increase the size of the dataset (Bhargava et al., 2020). Due to the randomized transformations, the same images cannot be obtained each time. Good model accuracy is dependent on the pre-processing of the raw images that are fed as input into the classifier (Progga et al., 2021).

2.4.2 Feature Extraction.

Ahmed Medjahed., (2015) defined feature extraction as the process used to reduce the amount of data needed to describe a large set of data correctly with an aim of facilitating decision-making, e.g., pattern classification. The most relevant feature of an image is usually extracted using mathematical functions and assigned to a label for the classification model. Bloch et al., (2020) did feature extraction of the snake images using Color and Edge Directivity Descriptor. Rajabizadeh and Rezghi, (2021) explored the use of Principal Component Analysis and Linear Discriminant Analysis, to extract features of an image that were later fed as vectors to image classifiers. LDA performed well compared to PCA. Also, Singular Value Decomposition has been used to reduce the data redundancy inherent in the dataset. A good feature extraction technique will produce good relevant results (Farajzadeh et al., 2017).

2.4.3 Image Classification

The vectors acquired from the feature extraction process are usually fed into a classifier of choice and labels assigned to them. The images are then categorized into predefined classes using the selected classifier that compares the image patterns with the learned labelled patterns.

2.5 Feature Extraction Techniques

Different techniques can be applied for image feature extraction. The dimensionality reduction processes are important in reducing computational time and storage. They can also be used in the feature extraction of an image. Feature extraction is the act of mapping an image from the image space to the feature space. We review the three commonly used dimensionality reduction techniques used in image feature reduction.

2.5.1 Principal Component Analysis (PCA)

PCA is an unsupervised linear technique that uses an orthogonal transformation to project a set of variables into a lower dimension with maximum variance. In PCA a set of variables that are possibly correlated, convert into a set of values that are not correlated variables, called principal components.

To reduce each data, the PCA tries to find an orthogonal matrix so that the reduced data have the maximum variance (Rajabizadeh & Rezghi, 2021)

$$C = \frac{1}{N-1} \sum (X_i - \bar{x})(X_i - \bar{x})^{T} \tag{2.1}$$

Where

$\bar{x}$ is the mean.

N is the number of data.

2.5.2 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a statistical analysis technique that can be applied to reduce the dimensionality of hyperspectral data cubes and extract features for classification. LDA is a supervised dimensionality reduction technique that makes use of labels alongside the features in the dataset. LDA also aims to reduce high dimensionality by extracting low dimensional features which have the most sensitive discriminant ability from high dimensional feature space. For each data *x*, it tries to find an orthogonal projector *U* by minimizing the within-class distance and maximizing the between-class distance of the projected data. In this way, LDA can maximize the separability of species in the snake image datasets. Features that explain the highest variance among species are selected by LDA. LDA ranks features based on how much variation among the species they account for.

2.5.3 Singular Value Decomposition (SVD)

SVD is a reliable and robust orthogonal matrix decomposition technique. SVD is widely used for image analysis to solve the problems related to least squares, pseudo inverse computation of the matrix, and also in multivariate experiments. In the area of machine learning, SVD-based methods are used to decrease dimensionality, metric learning, manifold learning, and collaborative filtering (Farajzadeh et al., 2017). Singular Value Decomposition (SVD) is one of the matrix factorization methods that allows reducing the size while keeping the characteristics of a matrix. It is important to reduce the size of the feature vector to reduce the computational load for training the classifier. Given an m × m matrix A, the expression of its SVD is as shown in equation 2.2 below.

$$\boldsymbol{A} = \boldsymbol{U} \sum \boldsymbol{V}^{\mathbf{T}} \tag{2.2}$$

where U is an $m \times m$ matrix,

V is an $n \times n$ matrix a

Σ is the singular value of matrix A which is an $m \times n$ non-negative real diagonal matrix.

We use SVD on the feature matrix, to make a shorter feature vector that still holds the features' characteristics. The reason for using SVD to make feature vectors is that it is relatively straightforward to apply and gives the singular values, which contain information about the features generated in descending order. Ordering the elements of the feature vector in descending order gives an advantage when using a training data set for a neural network. SVD is a better option to decrease the dimensionality of huge amounts of data by speeding up the computational process. Particularly, in image classification, SVD also performs better in the form of classification accuracy with optimal computational time (Seok et al., 2012).

2.6 Machine Learning Techniques used in Image Classification

2.6.1 Support Vector Machine

SVM is a supervised learning binary classifier in which, known labels help indicate whether the SVM is classifying correctly or not. It can be used as a prediction tool to predict future data. SVM predicts based on machine learning theory that increases the accuracy of a classification and overfits to data automatically. SVM also performs mapping low dimensional space into high dimensional by using non-linear basis functions. SVM uses hypothesis space of linear functions on high dimension feature space that is trained with learning algorithm from optimization theory. SVM uses linear classifiers (hyperplanes) to separate the data (Mahajan & Chaudhary, 2019)

2.6.2 K-Nearest Neighbor (KNN)

KNN is one of the most popular non-parametric, supervised machine learning algorithms in the field of pattern recognition, used for classification and regression. The classification rules in this algorithm are produced by only the training dataset. KNN predicts the test class according to the K training samples which are the closest neighbors to the test sample. The algorithm simply depends on the distance between feature vectors and classifies unknown data points by finding the most common class among the k-closest examples. The prediction is based on the largest class probability. The probability is given by the equation 2.3 below.

$$\boldsymbol{P(X, C_j) = \sum_d SIM(X, d_i) . y(d_i , C_j)} \tag{2.3}$$

Where $\boldsymbol{y(d_i, C_j)}$ is a category attribute function

The similarities are calculated between all training samples and X.*i*th sample $\boldsymbol{d_i(d_i 1, d_i 2, \ldots \ldots d_i n)}$

$$\boldsymbol{And\ SIM\ (X, d_j) = \frac{\sum_{j=1}^{m} X_j \ d_{ij}}{(\sqrt{(\sum_{j=1}^{m} X_j)^2} \sqrt{(\sum_{j=1}^{m} X d_{ji})^2})}} \tag{2.4}$$

Training categories $\mathbf{C_1, C_2, C_3, \ldots, C_j}$ and sum of training samples is N

Sample X is the same feature vector of the form $\mathbf{(X_1, X_2, X_3 \ldots, X_m)}$

2.6.3 Convolution Neural Network (CNN)

The Convolution Neural Network (CovNet) is a widely used multi-layer neural network, designed to recognize visual patterns directly from pixel images with minimal pre-processing for implementing the deep learning technique. It comprises layers for feature extraction and classification. The advantage of CNN is its ability to identify the image features without supervision from the human which is an advantage to snake image classification for envenomation management (Manoj et al., 2018). Figure 2.2 below shows the architecture of a CNN image classification model.

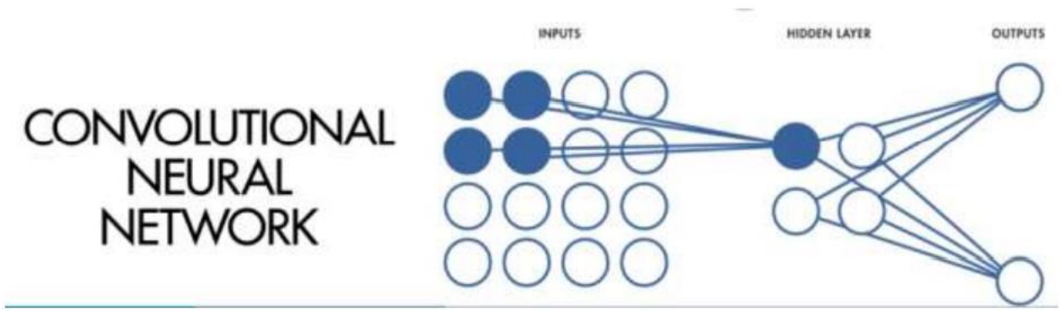

Figure 2. 2: CNN model Architecture (Progga et al., 2021)

A CNN consists of different layers, which include convolutional, an activation function usually rectified linear unit (ReLU), pooling, and the fully connected layers.

Convolution layer – This layer extracts features from the input image. Filters are used to extract these features. For example, assume a 32 x 32 image. The computer sees a 32 x 32 x 3 array of numbers, where the 3 represent the red, green and blue values for each picture element (pixel). Assume a 5 x 5 x 3 filter. The filter takes the first position at the top left corner of the image. It then slides, or convolves, across the input image multiplying the values in the filter with the original pixel values of the image. These multiplications are summed up to create one figure that represents where the filter is. The filter moves one unit or stride to the left and the process repeats. After sliding over the whole image, a 28 x 28 x 1 array of numbers called the feature map is created. If 6 5 x 5 filters are used then a 28 x 28 x 6 array will be created (Nielsen, 2015).

After each convolution, the output reduces in size and becomes the input of the next layer.

Activation function – Rectified Linear Unit (ReLU) – ReLU is an operation that replaces all negative pixel values in the feature map by zero. This is to introduce non-linearity since most of the real-world data is non-linear (Devikar, 2016). Pooling Layer – Pooling layers usually come after the convolution layer and its role is to simplify the information in the output from the convolutional layer. It reduces the dimensionality of each feature map but retains the most important information. A common pooling procedure is max pooling, where a pooling unit outputs the maximum value (Nielsen, 2015).

Pooling Layer-This layer comes after every convolutional layer. The role of the pooling layers is to simplify information in the output from the convolutional layer. The simplification process is achieved by the pooling layer taking each feature map output from the convolutional layer and preparing a condensed feature map. For instance, more precisely, each unit in the pooling layer may summarize a region of neurons in the previous layer. Max-pooling is one of common procedures used in the pooling layer. Max-pooling involves a pooling unit simply outputting the maximum activation in the 2×2 input region (Nielsen, 2015). For instance, considering 24×24 neurons output from the convolutional layer, after pooling, this will result in 12×12 neurons. This process reduces the dimensionality of each feature map while at the same time retaining important information.

Fully connected Layer – After several convolutional and pooling layers a fully connected layer takes all the neurons in the previous layer and connects it to every neuron in the layer to produce fully connected output. This is the layer that does the actual classification work.

Kamalraj, (2020) classified snake species using snake bite marks through image processing and Convolution Neural network. A picture of snake bite marks was uploaded to the system and later underwent image pre-processing. A convolution neural network was then used to classify the snakes as either venomous or non-venomous. However, getting a clear image of a snake bite is difficult because most patients tamper with the bite site while doing first aid. Furthermore, rarely will patients or doctors take images of the bitten site. Therefore, its biggest drawback becomes the difficulty in the collection of the images of all snake bite marks on individuals making the dataset too small to warrant model training. Also, the collection process has to be accurate and requires high-quality images for the process (Niteesh. et al., 2021).

Vasmatkar et al., (2020) proposed a model that involved techniques based on Image Processing, Convolution Neural Networks, and Deep Learning to classify snakes. CNN was used to extract features and utilize them for classification. However, deep learning methods like CNN require one to have a large dataset for better accuracy. This is a challenging task considering the risk and difficulty of collecting snake image data.

2.6.4 Transfer Learning

The drawback of implementing CNN on a small dataset is that it results in overfitting (Bhargava et al., 2020). A possible way of overcoming this challenge is to increase the size of the training data. However, this is a difficult and expensive task (Kaur & Gandhi, 2020). As such, Transfer learning becomes the appropriate solution. Zhuang et al., (2021) defined transfer learning as the use of stored knowledge gained from a pre-trained neural network model, to train a different but related neural network model with a smaller dataset. A major advantage of using Transfer learning is that it reduces the time a neural network takes to train a model and also results in a lower generalization error (Krishna & Kalluri, 2019). Sharing of pre-trained models is normally achieved with the help of millions of parameters or weights (Kaur & Gandhi, 2020). Figure 2.3 below shows a representation of Transfer learning.

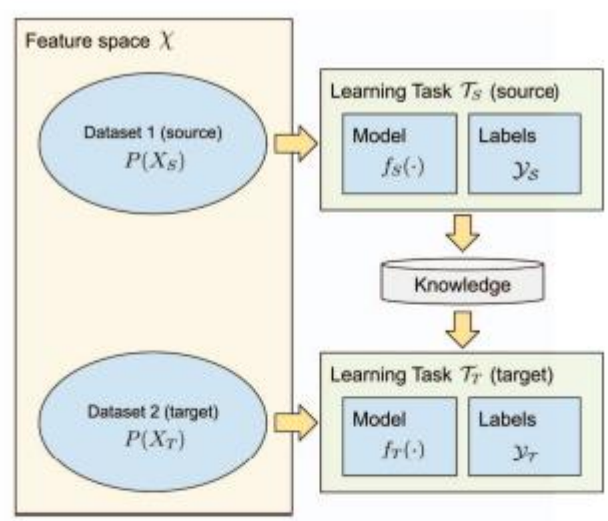

Figure 2. 3 : Transfer Learning Architecture (Ribani & Marengoni, 2019)

There are different models for transfer learning. Among them, DenseNet, MobileNetV2, VGG19 etc. DenseNet is a convolutional neural network model made up of dense blocks with a densely connected layer that uses a smaller number of parameters. Every dense block has a batch normalization layer, rectified linear unit (ReLu) activation, and 3x3 convolution forming a composite function. In DenseNet, all the feature maps are concatenated, but the feature maps of each layer have the same size. Transition layers between two blocks contain batch normalization, 1x1 convolution followed by average pooling ensures downsampling. Bottleneck layers have been introduced before each 3x3 convolution to reduce the number of input feature maps and thus, improve computational efficiency. Vasmatkar et al., (2020) classified snakes images using CNN transfer learning DenseNet161 but applied the GrabCut algorithm to the images as a feature extractor and performed data augmentation.

MobileNetV2 is a light weighted model mainly used for mobile and embedded vision applications. The model is built on depthwise separable convolution. It introduces two global hyperparameters, which are width multiplier and resolution multiplier which helps to trade-off between latency and accuracy. Rajabizadeh and Rezghi, (2021) explored the combination of convolutional neural networks that was achieved through transfer learning using MobileNetV2 and dimensionality reduction approaches for feature extraction of the images. The model achieved an accuracy of 93.16%.

2.6.5 Tensor Flow Framework

TensorFlow is a high-performance numerical computation framework or software library originally developed by researchers and engineers at Google. It is an end-to-end platform that is open source and has strong support for the deployment of machine learning and deep learning models. It offers a free GPU which makes it faster and easier to train a machine learning model compared to a CPU (TensorFlow, 2018). At Google, for example, the framework powers many of their products like Gmail and Google translate. This study used the convolutional neural network libraries and tensorboard, the visualization toolkit, of TensorFlow.

2.6.6 Tensor Flow Lite.

Tensor Flow lite is a set of tools that facilitates on-device learning, by enabling models to run on the mobile, embedded or edge devices. They are generally fast because the speed and storage are optimized. It provides the basic features needed for making inferences at the edge like, low latency, lightweight, optimal power consumption, secure among others. After the model is trained, it is converted to a tensor Tensor Flow Lite format using a converter. During the conversion process from Tensor Flow model to Tensor Flow Lite model, the side of the file of the model is reduced. There is normally a choice of the trade-off between reduced file size and execution speed. Tensor Flow Lite model is efficient in accuracy and is also a lightweight version that occupies less space hence its popularity in use for mobile devices.(Choe et al., 2022; David et al., 2021)

Table 2. 1 Summary of Literature Reviewed

Author	**Technique Used**	**Weakness**
Rusli et al., (2019	NLP	Visual descriptions of snake vary depending on environment and age, sex etc.
Bloch et al., (2020)	CEED &Naïve Bayes	Traditional machine learning parameters in feature reduction method are not trainable
Abdurrazaq et al., (2019)	CNN	Needs a huge dataset
Rajabizadeh and Rezghi, (2021)	PCA&MobileNetv2	Low accuracy
Niteesh. et al., (2021)	1D3	Victims give incorrect descriptions when in panic mode

2.7 Research Gap

Classification of snakes is still a major challenge without expert knowledge and relying on the human eye and descriptions has proven to be a daunting and difficult task. One would require expert knowledge to achieve this. The deep learning algorithms discussed in the preceding sections describe recent research on the classification of snake species through the use of images. However, the availability of snake specialized datasets is a challenge more so regarding the rare snakes. Moreover, extracting the distinct feature of snake species is very important to come up with better inspection methods. Previous research has explored the use of PCA and LDA but didn't explore the possibility of using

SVD despite its apparent advantages. Considering the small dataset at hand, this research proposed to explore feature extraction using SVD and later feed the vectors as input to a deep neural network using transfer learning to classify snakes according to their name and venom toxicity category, based on their images. This is achieved by harnessing the power of transfer machine learning.

2.8 Conceptual Framework

The following conceptual framework links the reviewed literature with the research problem and objectives. Based on the literature review, the framework shown in figure 2.4 combines techniques cited to come up with a model that can achieve the objectives of classifying snakes according to their species name based on the image. This framework was established to provide a solution to the problem in the study of classifying the snake species and venom toxicity based on their images.

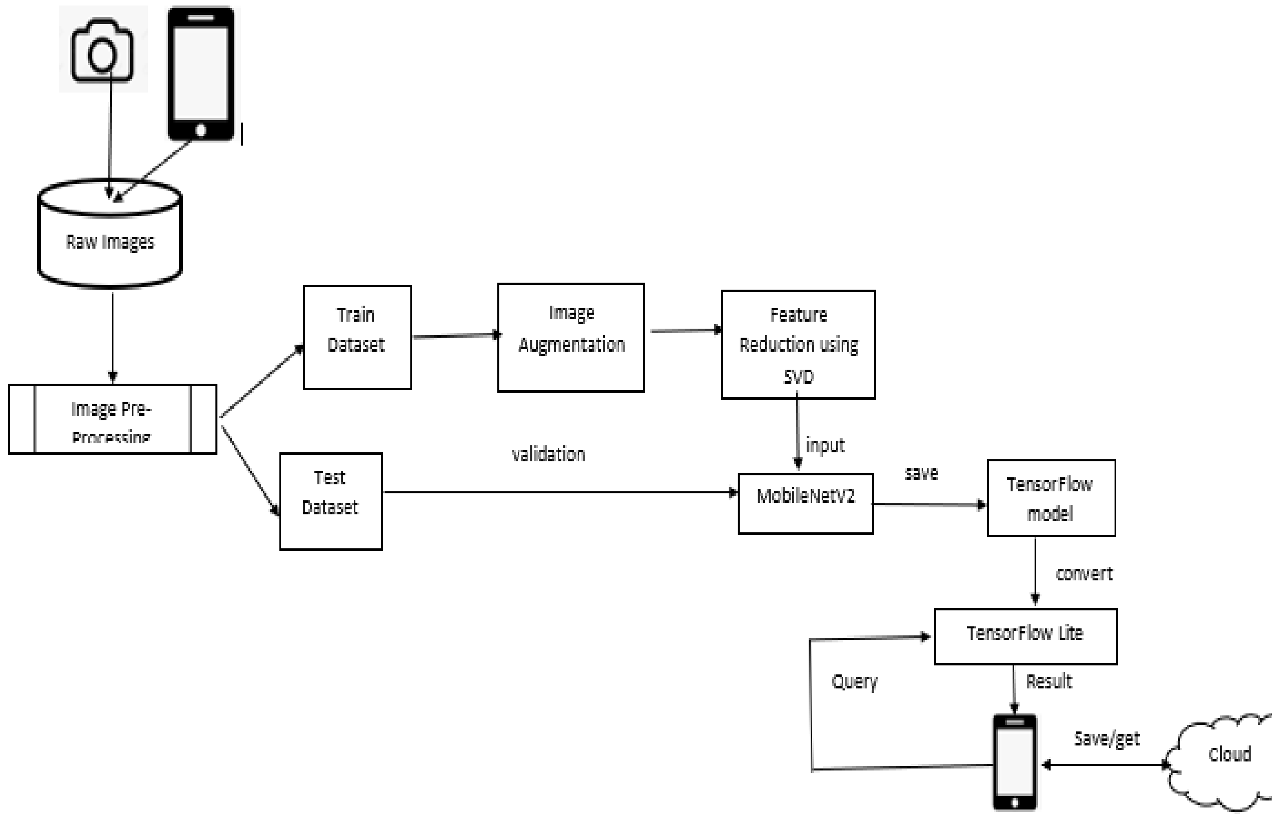

Figure 2. 4 Conceptual Framework of the Snake Image classification Model

Chapter 3: Research Methodology

3.1 Introduction

This chapter explored the research design, data collection, and the development of the image classification model. Additionally, the chapter covered the methodology that would be used in system development. Finally, the chapter made a statement on the research quality and the ethical considerations for the research.

3.2 Research Design

A research design is a blueprint describing how a research study is to be completed: operationalizing variables so they can be measured, selecting a sample of interest to study, collecting data to be used, and analyzing the results (Thyer, 1993). The objective of this research is to design an image classification model that can classify snake species for snakebite envenomation management. This is applied research because it solves the real-life problem of managing snakebite envenoming. The research paradigm is quantitative in nature and the data collected from the images taken can be quantified in numbers to represent the output. The method is correlational in nature because features from the snake images can be used to classify the outcome of an image.

3.3 Population and Sampling

3.3.1. Population

Several but quite limited studies have been done on snake species worldwide and none so far to the best of the researcher's knowledge had been conducted in Kenya. Considering the difficulties and risks experienced in capturing the image of these stealth and dangerous species, the dataset is normally small and not varied. Furthermore, there is currently no secondary dataset that exists of images of Kenyan snake species to the best of researcher's knowledge. Therefore, primary data was used for the study. The snake species images will be the target population mainly from the Kenya National Museum of Kenya and the Stedmark snake park Garden in Karen. The population consists of a total of 4521 images from eighteen different species that covered the five categories of snake venom toxicity. African Rock Python had 145 images, Black Mamba had 377 images, Black-necked spitting cobra had 128 images, Boomslang had 152 images, Brown House snake had 211 images, East African garter snake had 127 images, Egyptian Cobra had 232 images, Gabon viper had 253 images, Green Mamba had 428 images, Jackson Tree snake had 57 images, Jameson Mamba had 381 images, Large Brown spitting cobra had 433 images, Mt. Kenya bush viper had 203 images, Puff Adder had 187 images, Red spitting cobra had 331 images, Rufous beaked snake had 396 images, Sand Boa had 160 images, and finally western cobra had 320 images.

3.3.2 Sampling

The sampling technique used was a combination of clustered and convenient sampling of the data. This is because non-probability sampling supports its use where the researcher has limited control of the representation of the selection. Snakes generally hide and reside in forests and thickets hence the difficulty in getting the images. Furthermore, they are dangerous and wild and can attack when they feel threatened. Therefore, the snake species available in the National Museum and Stedmark Garden snake park in Karen became a representation of the snake species found in Kenya. The species available in these two snake parks have been picked from different parts of Kenya including but not limited to, Meru, Nairobi, Kibwezi, Mombasa, Kakamega, etc. The primary images were collected under the guidance of a herpetologist and with the help of a snake handler for proof of concept.

3.4 Data Collection

Primary data consisting of the snake images, venom toxicity category, common name, type of general antidote administered, first aid guidelines, and scientific names were collected from Kenya National Museum, Nairobi, and Stedmark Gardens snake Park, Nairobi. The primary devices for data collection from the sources were digital cameras and smartphone devices which is the target device for the application. The image files collected were in .jpg and .arw format since the quality of the images was of high importance to building a good model. A sample of the primary data collected is attached in appendix G. The image of the snake species type was saved in a folder using its common name. An excel file was created and it contained the snake species' common name, scientific name, and the venom toxicity category of the species. Since the users are not experts, the model predicts the snake species on the image and classifies it based on the venom toxicity category on their behalf.

3.5 Image Pre-Processing and Analysis

3.5.1 Image Augmentation

Image augmentation artificially expands an image dataset. Image augmentation operations include flipping, rotation, shear, zoom, cropping, deforming, adjusting the hue, adjusting saturation, or adjusting brightness and contrast (Custom Image Augmentation: Towards Data Science, 2017). Considering the small dataset available for the research, the images were augmented to expand the image dataset by three or four times and helped the model cope with all distortions that occur in real-life images.

3.5.2 Standardization

The images were cropped to remove excess background and ensured at least 60% of the image coverage size was composed of the snake. The raw images were of different formats and therefore were converted to a standard format .jpeg also which is the format of our target device. The images were pre-processed as well to increase the brightness and contrast and resized to 224*224, which is the image size input requirement for MobileNetV2. Feature reduction was implemented using SVD, to pick the most relevant feature for the classification. These images served as an input to the model for classification.

3.5.3 Folder Structure

Approximately 70% of the images formed the training dataset and the remaining 30% formed the validation dataset. The images collected were put into the respective dataset folder and subfolders corresponding to the species name.

3.6 System Development Methodology

3.6.1 Agile System Development

An agile development methodology was used in this research because it is incremental and iterative. Most importantly it has a theoretical background that is consistent with the problem-solving approaches tackled in this research.

3.6.2 System Analysis

The system detected the features on the snake images and classified the snakes based on their name and venom toxicity. The camera was the primary device used to capture the images needed for analysis. Therefore, the application classified a particular snake based on its species name and venom toxicity.

3.6.3 System Design

The system architecture is comprised of both hardware components and software components. The hardware components are a camera for capturing images and a processor whereas the software components are the snake species classification model, the software tools used such as Google Colab, programming language, and its corresponding libraries.

The snakebite envenomation management system which will be a mobile application, software modeling diagrams will be used to show the structure and logical interactions of the various components. System design modeling will be done using the following UML modeling diagrams.

i. Use Case Diagrams: To provide a simplified graphical representation of what the system should do in a use case, which is a description of the tasks that a user can perform on the platform and how
ii. Sequence Diagram: To show the interaction logic between the objects in the platform in the time order that the interactions take place
iii. Entity-relationship Diagrams: To describe how various entities in the platform relate to each other
iv. Class Diagrams: To model the objects that make up the system, display the relationships between the objects, and describe what those objects do and the services that they provide.

3.6.4 System Implementation

3.6.4.1 Model

The development of the model was carried out on the Google Colab platform. Google Colab is a free cloud service that offers free GPU (Graphics Processing Unit) that were essential for building this model. The main input for the mobile application is the user-uploaded image, which is fed into the MobileNetV2 transfer model for classification. The result obtained is shown to the user. The application of deep learning required very expensive hardware and powerful libraries such as Keras, OpenCV, sklearn, and TensorFlow. This was created using the Google Colab and was the testbed environment. All training and validations were done in the same environment.

3.6.4.2 Test Bed

The testbed was required to test the model. This can include hardware and software. The hardware included a mobile phone with a camera. Software components included image classification algorithms were used to predict and classify different snake species and their toxicity category. The testbed was Google Colab where all test images and respective labels were inputted into the model to predict their classes. The testing was done using a test dataset that was set aside earlier during the research. Accuracy, precision, and recall were the metrics used to evaluate the performance of the model.

3.6.4.3 Programming Language

The programming language used in this research was python because of its wide array of open libraries that were easily accessed. This language easily works through the Keras and TensorFlow libraries. The algorithm can run on the Google Colab which offered graphics processing units for running the decay algorithm.

3.7 Research Quality

This was a very important aspect of maintaining accuracy in this research, therefore, validity, objectivity, and reliability of the research were important. This research achieved this by limiting bias in the data collected. The research minimized the effects of extraneous independent variables that were undesirable. This was achieved by using deep learning techniques that focussed on every aspect of the features of the images and the training can minimize their effects. Accuracy, precision, and loss function will be used to measure the validity of the model. A confusion matrix was used to generate a classification report in this research.

3.7.1 Research Validity

Validity refers to the extent to which the data accurately measures what it was intended to. Automated model testing and a confusion matrix was used to ensure the validity of the model. Additionally, the MobileNetV2 network model's loss or learning rate was visually tracked using Tensorboard; this made it easier to debug and optimize the model.

3.7.2 Reliability

Reliability focuses on the extent to which the data collection method will yield the same findings if replicated by others. Proper documentation of system requirements was done to ensure the reliability of the study.

3.8 Ethical Considerations

Ethics are standards or codes of conduct that help distinguish between what is acceptable and what is not (Ethical Considerations: CIRT, 2018). Images used in this study were obtained from the National Museum of Kenya with permission to use for educational research. All previous works were cited appropriately and due acknowledgment was given to the respective authors. To ensure adherence to ethical codes of conduct, the collection process was in line with the permission for educational research. An approval from the Strathmore University Ethics Review Board attached in appendix E was obtained before commencing the study. To ensure the originality of the study, a similarity check was done and the report is as shown in appendix F.

Chapter 4: System Analysis, Design, and Architecture

4.1 Introduction

The main purpose of this study was to design a model that will classify Kenyan snake species using their image, based on their venom toxicity for the purposes of managing snakebite envenoming. Object oriented analysis and design (OOAD) research design was employed in this research. The chapter outlined the design architecture and explained the components of the developed system, the interactions between the different components of the developed system, and the interactions between the users and the developed system. Unified modeling language (UML) was employed to model and illustrate the interactions of the system using the use case diagram, system sequence diagram, entity-relationship diagram, and class diagram. Wireframes were also drawn to display the functional elements of the mobile application.

4.2 Requirements Gathering and Analysis

The primary raw images data were collected using a camera and a smartphone from the National Museum of Kenya and Stedmark Garden Snake Park, Kenya. The images had to capture the head and part of the body of the snake. Due to subtle differences that exist in snakes of the same species and similarities of different species, one way of distinguish between the snakes is to have a view of the body and the head together in image capture for better identification. They were in .jpeg and .arw format. They were all converted to standard format .jpg format using Adobe lightroom. Blurred, small, and unusable images were removed from the dataset. Images were manually cropped to get the best view with less background noise. They were categorized in folders and saved according to the snake species' local name. An excel file was created to document the snake's common name, scientific name, and the venom present in that particular species. The images collected were of different shapes and therefore they were all reshaped to size (224x224x3) to fit the model input requirements. SVD was used to do feature reduction.

4.2.1 Functional Requirements

i. The system should allow first-time users to register.
ii. The system should allow the user to upload a snake image in .jpeg or.jpg format as the input to the classification model.

iii. The system would predict the snake species in the input image.

iv. The system should classify snake species according to their venom toxicity category.

4.2.2Non-Functional Requirements

i. Usability - The proposed users of the application are the general public and medical practitioners who would like to use it for snakebite envenoming management application. Therefore, the system should be easy to use.

ii. Supportability Requirements - The model should support images from a wide range of input cameras with different formats. This will promote the usability of the model in a real-world environment.

iii. Reliability Requirements - The model should be able to provide consistent results when provided with the same input data.

iv. Maintainability The model should be easy to maintain and support.

v. Scalability - The system should gradually improve on the accuracy levels. It should also adapt to new datasets from different environmental conditions. This can ensure a good application response to new data.

4.3 System Design

The MITRE Corporation, (2014) defined system design as visualizing the internal system or creating a blueprint of the system while deriving a solution that can be traced back to the user requirements. This research adopted agile development methodology which is incremental and iterative and therefore object-oriented analysis and design were used.

4.3.1 System Architecture

The system architecture for the snake image classification model is shown below in Figure 4.1. Raw images from the camera or mobile phones are pre-processed and later split to create training and testing datasets. Image augmentation was done to increase the size of the dataset. Feature extraction was done using SVD and vectors fed to the model. Subsequently, the model was trained and validated using the transfer learning model MobileNetV2. The main input for the model is the user-uploaded images, which were fed into the model for classification. The result obtained was shown to the user.

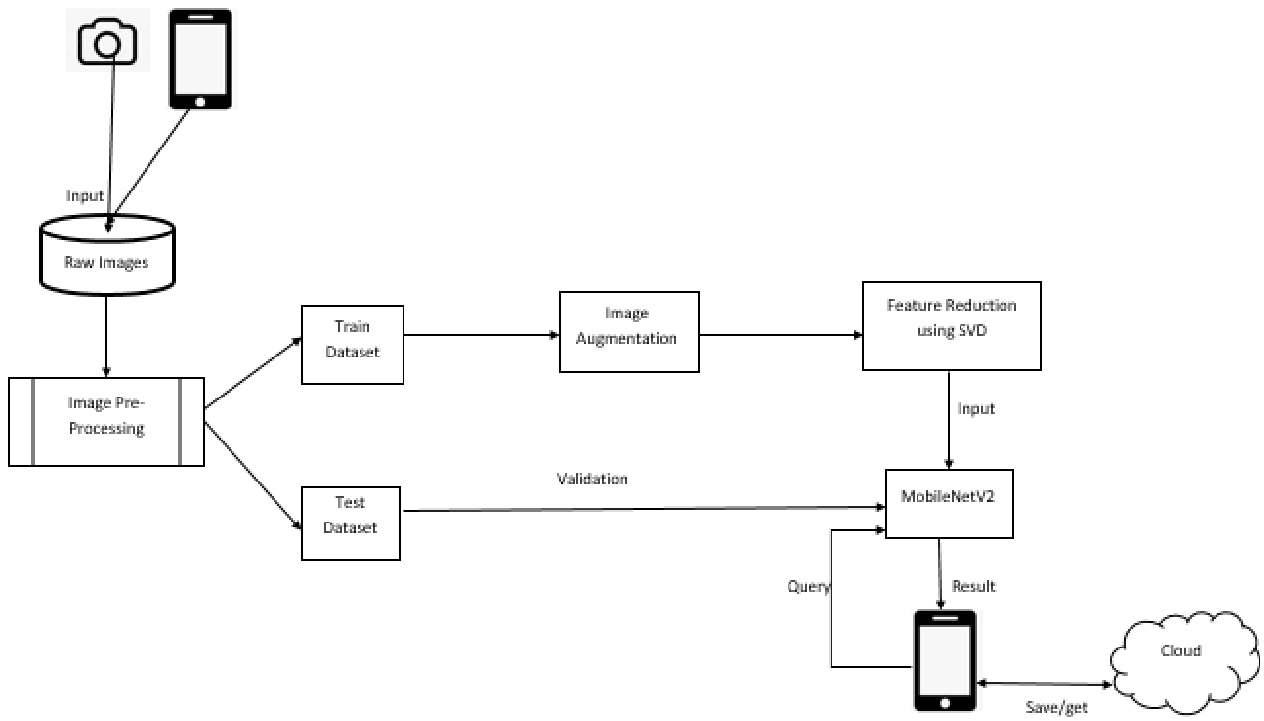

Figure 4. 1System Architecture

4.3.2 Use-Case Diagram and Descriptions

A use case is a list of actions and event steps that shows and defines the interactions between an actor and a system to achieve a goal. The actors in this system are the general users and the developer. The developer collates and pre-processes the image dataset and trains the transferred deep neural network as explained in chapter three. The developer then creates a model that is used to infer the images uploaded by the user. This model makes the snake image classification. The victim, who is a sub-entity of the user, provides the system with an image to classify. Figure 4.2 below shows the system use case diagram.

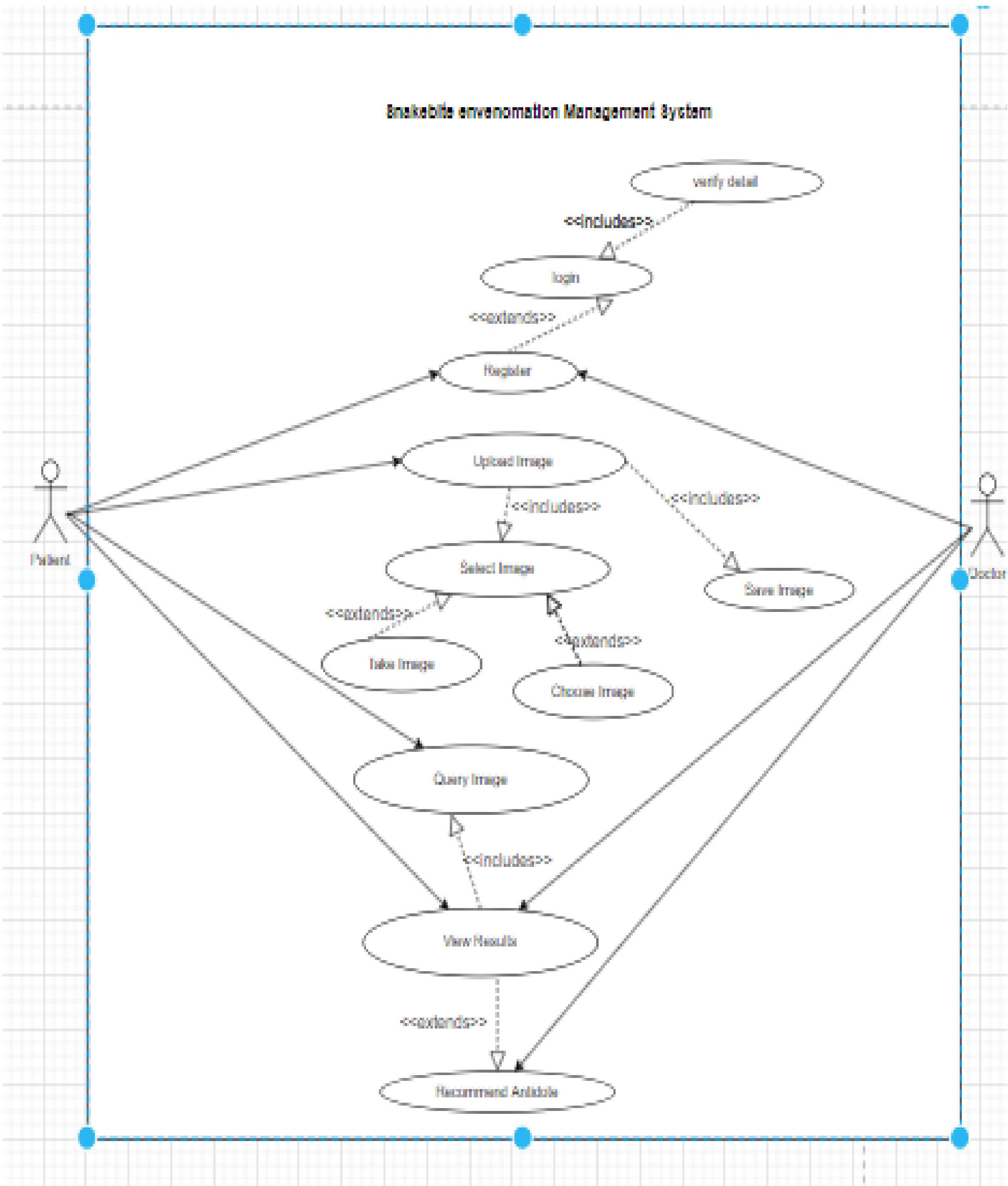

Figure 4. 2Use case Diagram

Table 4. 1 Register and Login Description

Use Case	Register and Login
Primary	The User (Patient and Doctor)
Brief Description	This use case describes how the users will register and login.
Pre-Condition	The user wants to classify a snake image and know its details and antidote, Registration Details
Post-Condition	Registered user, Authenticated User
Major Steps Performed	
Actor	**System**
The user enters Registration Details	
	System Verifies and saves registration details
The user enters login details	
	The system verifies and authenticates the user

Table 4. 2Load an Image and Classify Description

Use Case	Load an Image and Classify
Primary	The User
Brief Description	This use case describes how the users load an image into the system for classification.
Pre-Condition	Available Image, Trained model
Post-Condition	Classified Image with its information
	Major Steps Performed
Actor	**System**
The user selects an image from the photo file or takes an image using the phone's camera.	
	The system loads the image as input to the MobileNetv2 trained model
User uploads a .jpg image for classification	The system performs classification
	The system displays the results
The user views the result	

Table 4. 3 Query and View Results Description

Use Case	Query and View Results
Primary	The User (Patient and Doctor)
Brief Description	This use case describes how the users query a snake image for classification and view the results
Pre-Condition	Logged in. Uploaded Image
Post-Condition	Image classified and results displayed
	Major Steps Performed
Actor	**System**
The user enters login details	
	The system verifies and authenticates the user
User uploads a .jpg image for classification	
	The system queries the model and classifies the image
	The system displays the results of the uploaded image and its information.

Table 4. 4 Recommend Antidote Description

Use Case	Recommend Antidote
Primary	The Doctor
Brief Description	This use case describes how the Doctor will recommend the antidote based on the results of the queried image.
Pre-Condition	Result of the query image
Post-Condition	Recommended Anti-dote
	Major Steps Performed
Actor	**System**
	The system displays the results of the classified image with its venom toxicity information
The Doctor views the result of the classified image to identify which toxins it contains.	
The doctor recommends the appropriate antidote and logs it into the system	
	The system displays the antidote recommended
User (patient) views the Recommended antidote	

4.3.3 System Sequence Diagram

The system sequence diagram shows the interactions between the main entities in the system. Figure 4.3 below shows the flow of activities in sequence on how the user gets registered and logs in to the system.

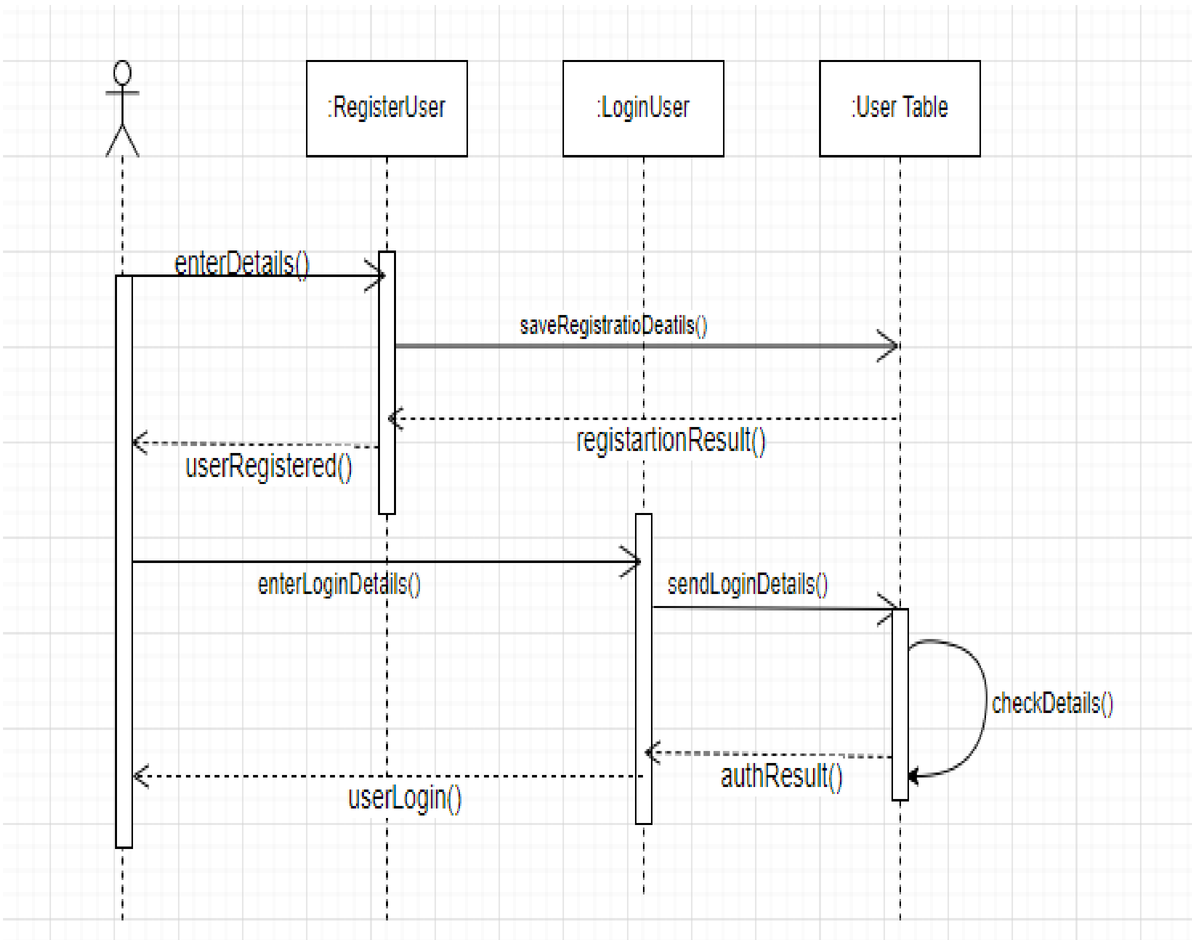

Figure 4. 3 Login System Sequence Diagram

The patient selects or takes a picture that they want to be classified. The patient then loads the image to the image classifier model to classify the uploaded image. The image classifier queries the trained transferred neural network to obtain a result. The result is displayed to the patient. The doctor requests to view the result of the classified image and recommends an antidote based on the venom content of the classified snake. The patient gets to view the recommended antidote. This is shown in figure 4.4 below

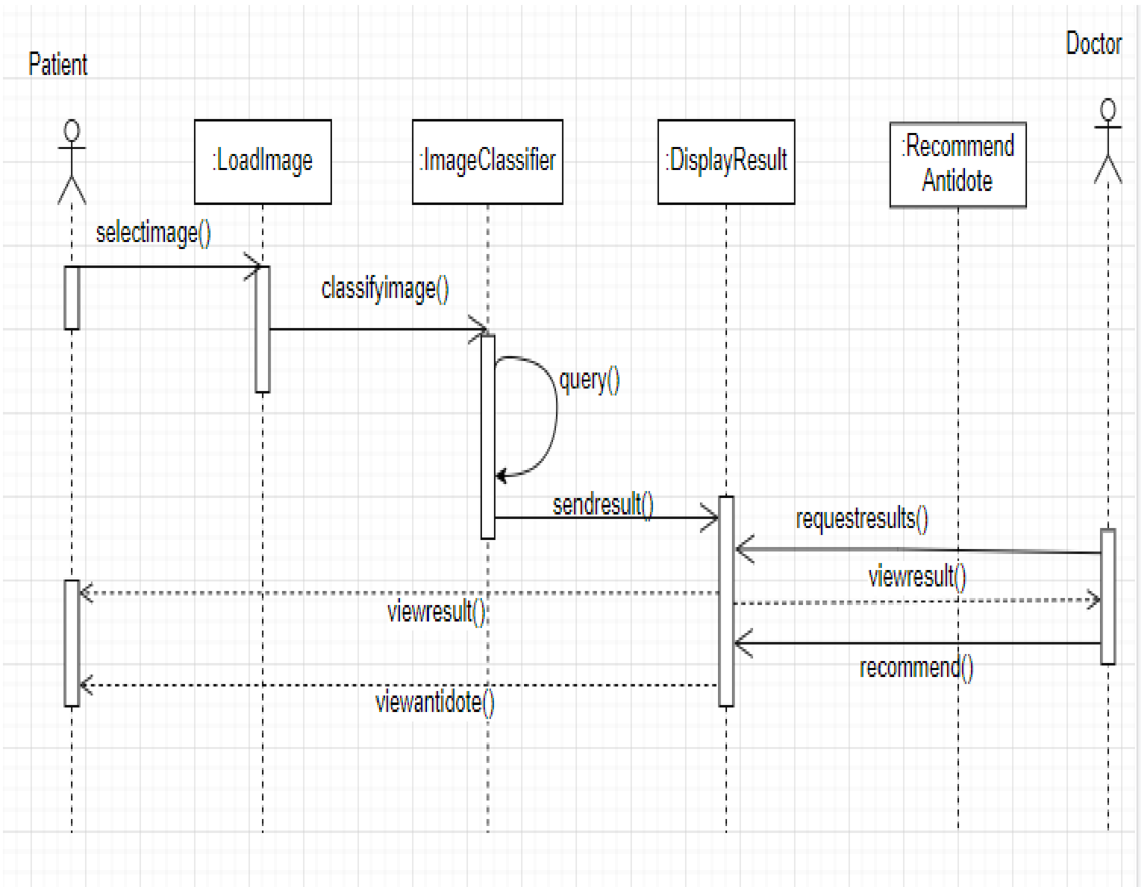

Figure 4. 4 Classifying System Sequence Diagram

4.3.4 Entity Relationship Diagram

Figure 4.5 describes information using a wide range of entities, attributes, and relationships. Entity Relationships (ER) are high-level conceptual models designed to facilitate the development of databases. The ERD diagram below, therefore, forms the blueprint of the database used in the detection model.

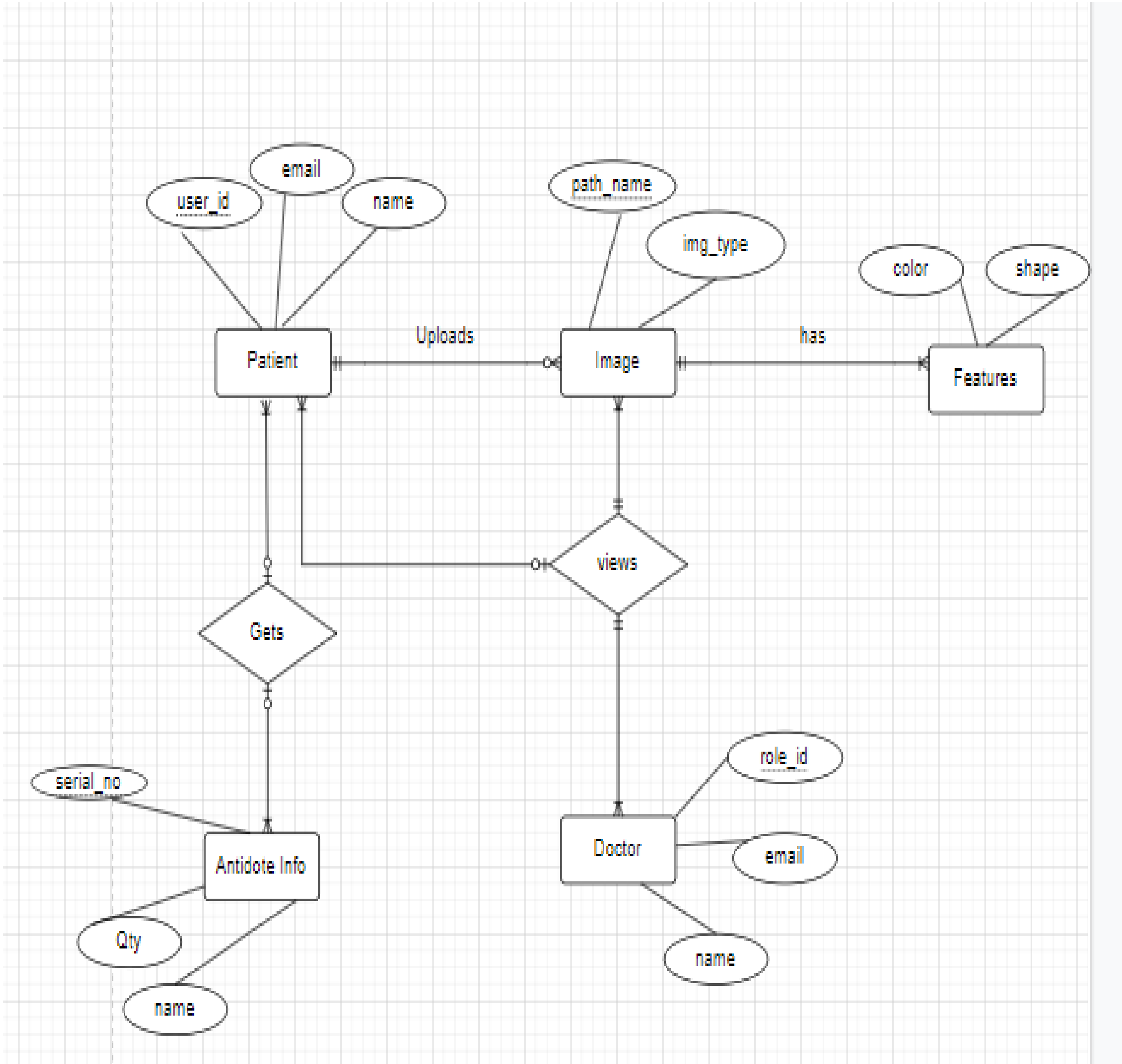

Figure 4. 5 Entity Relationship Diagram

4.3.5 Class Diagram

A class diagram defines a structure diagram that shows the classes, operations, attributes, and associations of the model. Class diagrams highlight artifacts (classes) that create objects based on common (shared) operations, relationships, and attributes (Alhumaidan, 2012). The same is outlined in figure 4.6 below.

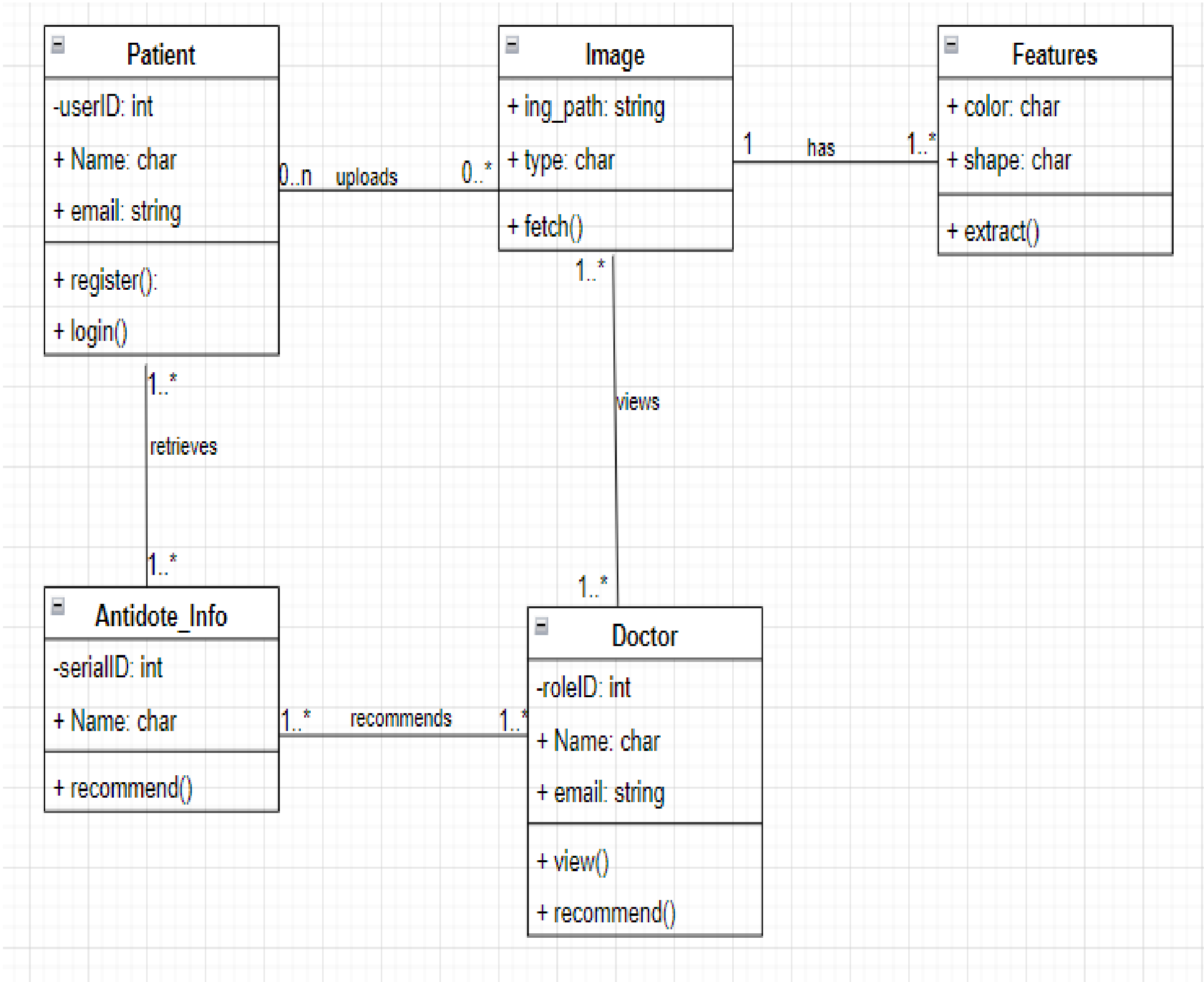

Figure 4. 6 Class Diagram

4.3.6 Wireframes of the Mobile Application

Figure 4.7 and 4.8 below shows the mobile application's wireframes. The user interface is designed to be simple and easy to use. The user has to register and log in to upload the image and retrieve the venom category of the snake and its recommended antidote.

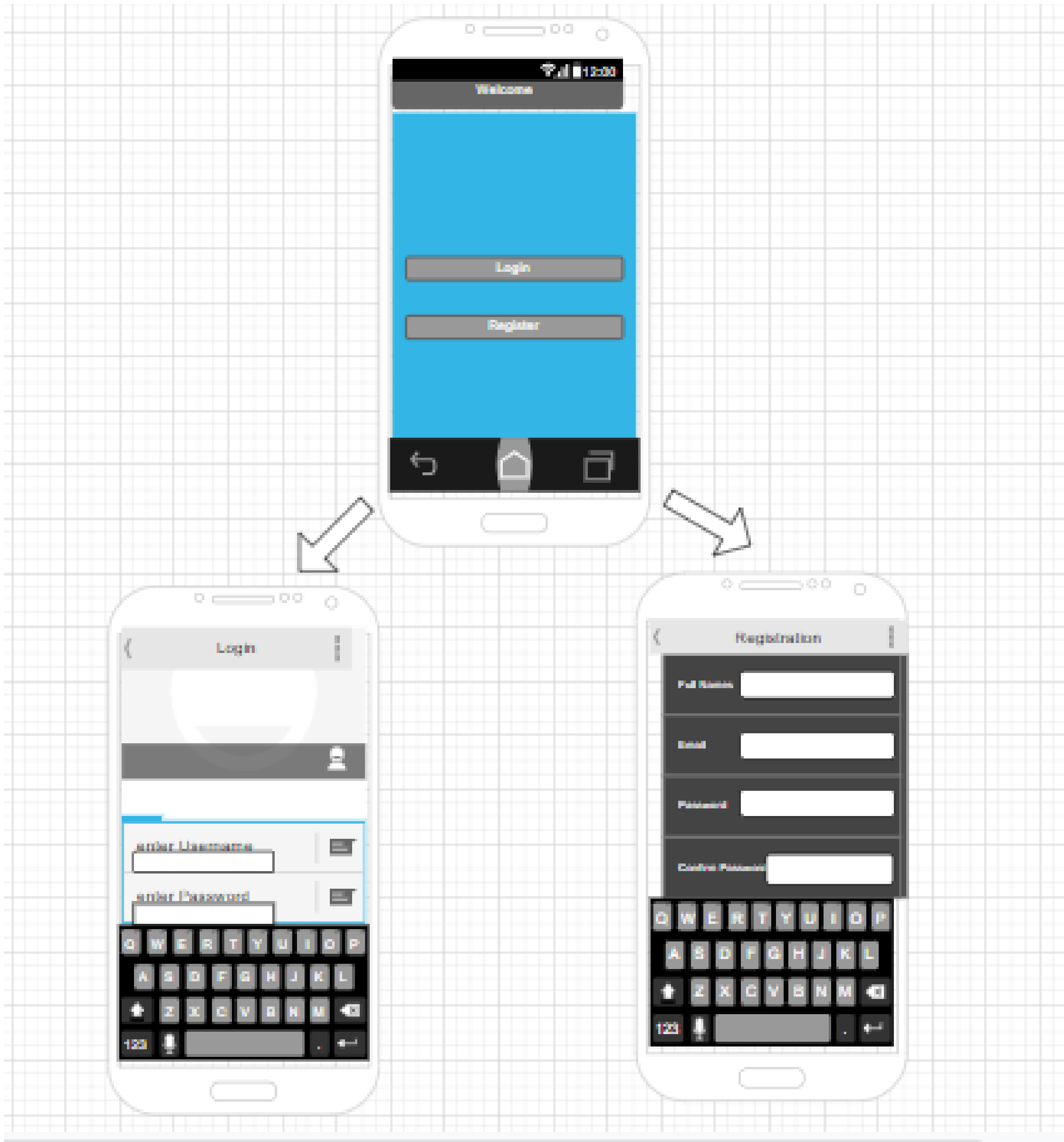

Figure 4. 7 Login Wireframes

Figure 4. 8 Classification Wireframes

Chapter 5: Model Development, Implementation, and Testing

5.1 Introduction

This chapter described the layout of the system. The model implementation stage described data preparation, the various steps that were taken to build the model, and its testing. In addition, the programming tools, and development environment are all discussed in this chapter. As described in the conceptual model in the literature reviewed in chapter two, the research entailed retraining, validating, and testing the MobileNetV2 model using Kenyan Snake Image datasets. The resulting model was embedded into the Android mobile application. Testing included functional and usability tests to check if the developed model accomplished the objectives set out at the beginning of the research project. This study aimed to build a snake species classification model for snakebite envenomation management.

5.2 Development environment and Language

The development of the model was carried out on the Google Colab platform. Google Colab is a free cloud service that offers free GPU (Graphics Processing Unit) that were essential for building this model. A GPU performs matrix operations like convolution much more quickly than a CPU (LeCun, Bengio, & Hinton, 2015). The application of deep learning required very expensive hardware and powerful libraries such as Keras, OpenCV, sklearn, and TensorFlow. The Google Colab platform was more preferred since it provides access to powerful libraries such as OpenCV, Keras, and TensorFlow which were essential in developing the model. It was also possible to automatically back up the notebooks containing the code and the image datasets on the google drive. All training and validations were used in the same environment.

5.2.1 Software Requirements

The software resources consisted of the cloud-based hardware components used on the Google Colab development environment. There was limited control over these resources as they were being accessed on a Google Colab which is a cloud-based platform. Python language was preferred because of its ease of use and compatibility with the Colab. The table below highlighted the software requirements of this research.

Table 5. 1 Software Requirements

Software	Details	
Google Cloud GPU	GPU	1xTesla K80, having 2496 CUDA cores, compute 3.7, 12GB (11.439GB Usable) GDDR5 VRAM
Python 3.10	**Libraries**	**Version**
	Tensorflow	2.8
	Tonsorboard	2.8
	Numpy	1.21.2
	Keras	1.16.2
	Augmentor	8.0.1
	PIL	8.0.1
	Scikit Learn	0.24
	Cv2	4.5.1
	Matplotlib	3.3.4
	Split-Folder	0.4.3
Kotlin/Java	XML & Groovy	
Adobe Lightroom		3.7.0
Android Studio		4.5
SQL lite		2.2.0

5.2.2. Hardware Requirements

The system was developed in the following hardware environment.

i. HP Pavilion 15Notebook PC, 64-BIT operating system, x64-based processor
ii. Intel(R) Core (TM) i7-5500U CPU @ 2.30GHz, 2.40 GHz
iii. 12.0 GB installed RAM
iv. 500GB Hard Disk
v. Microsoft Windows 10 pro

5.3 Model Components

The model was created using transfer learning. Features extracted from the activation of the MobileNetV2 deep neural network, which has been trained on a large, fixed set of image classification tasks, were re-purposed specifically for this project. As reviewed in Chapter two, MobileNetV2 was chosen instead of bigger networks like DensNet because the MobileNetV2 is specifically designed for on-device vision tasks like image classification. They create smaller, fast models yet do not lose much in terms of accuracy (Sandler & Howard, MobileNetV2: The Next Generation of On-Device Computer Vision Networks, 2018). The MobileNetV2 model created had several components as summarised in Figure 5.1 below. The graphical representation of the model was obtained by running the tensorboard visualization tool that is “tensorboard --logdir on the colab.

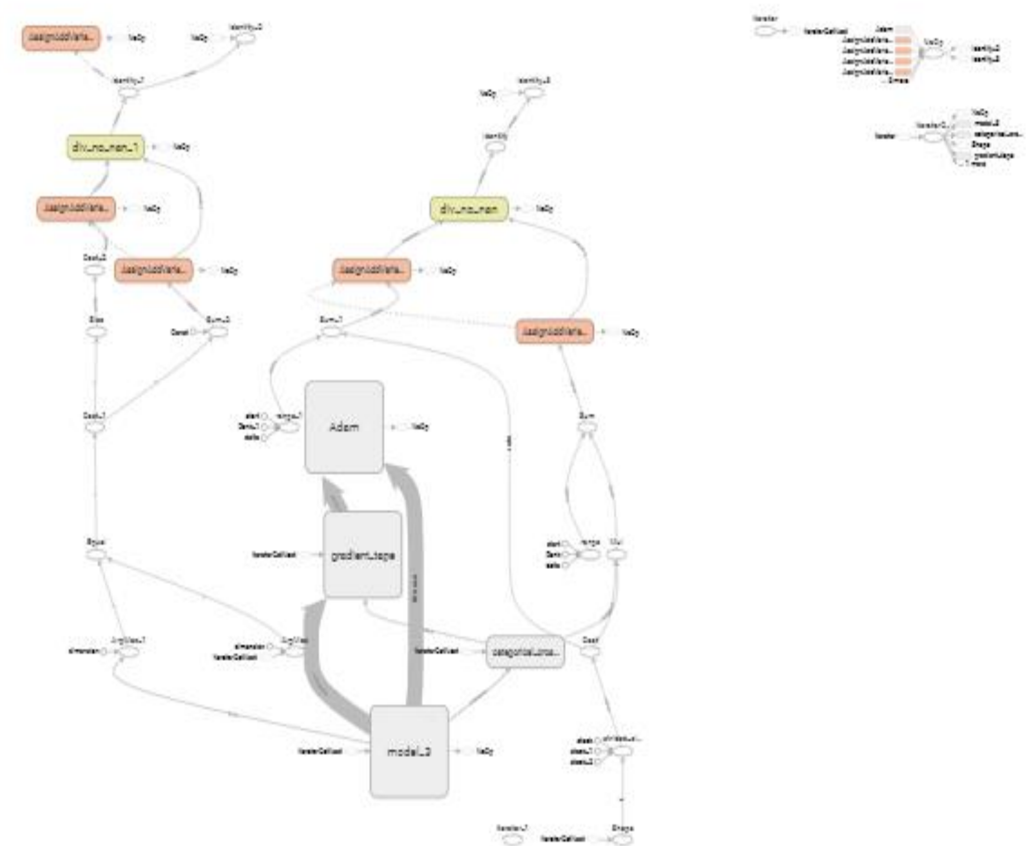

Figure 5. 1 Graphical Representation of the model

5.3.1 Storage

The collected images were stored on Google Drive, a cloud-based storage platform offered by Google. This platform enabled the larger image dataset to be loaded in a fast and efficient manner. The notebooks used for writing and editing code were also stored on the platform enabling efficient synchronization with the dataset. The choice of storage also offered a cheaper platform as compared to acquiring hardware devices for data storage.

5.3.2 Input Layer

This is the first layer of the model and it interacts with the external environment. This layer consists of the input nodes, place holder, and module which provided input images to the network. This was implemented to accept standardized input images (224X224) in the form of arrays and pass this information to the hidden layers. The graphical representation of the input layer is shown in figure 5.2 below.

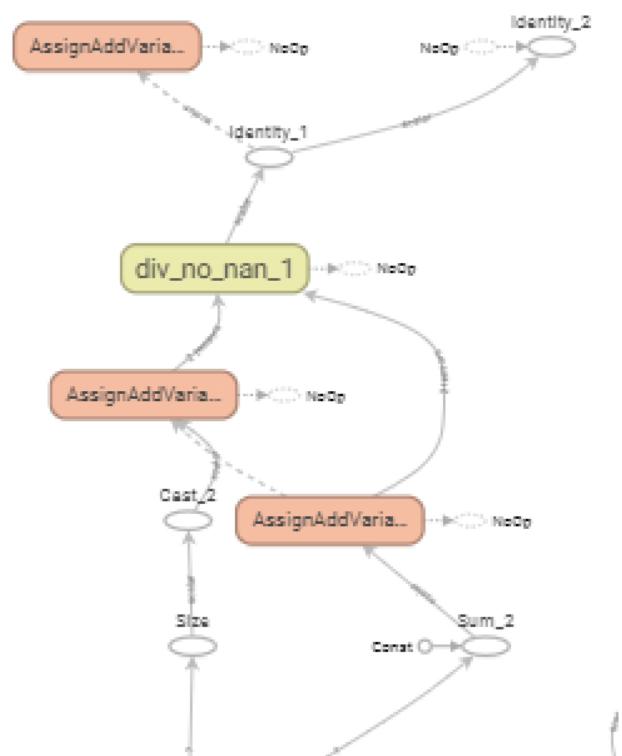

Figure 5. 2 Graphical Représentation of the input Layer

5.3.3 Hidden Layer

This refers to a series of network nodes that are not visible from the outside world. They are responsible for performing the necessary computations for feature extraction from the input images. This layer passes information from the input layer to the output layer having extracted the relevant features. The following processes were implemented within the hidden layer: Convolution, pooling, and activation. The model made use of the SoftMax activation. The graphical representation of this layer is shown in figure 5.3 below.

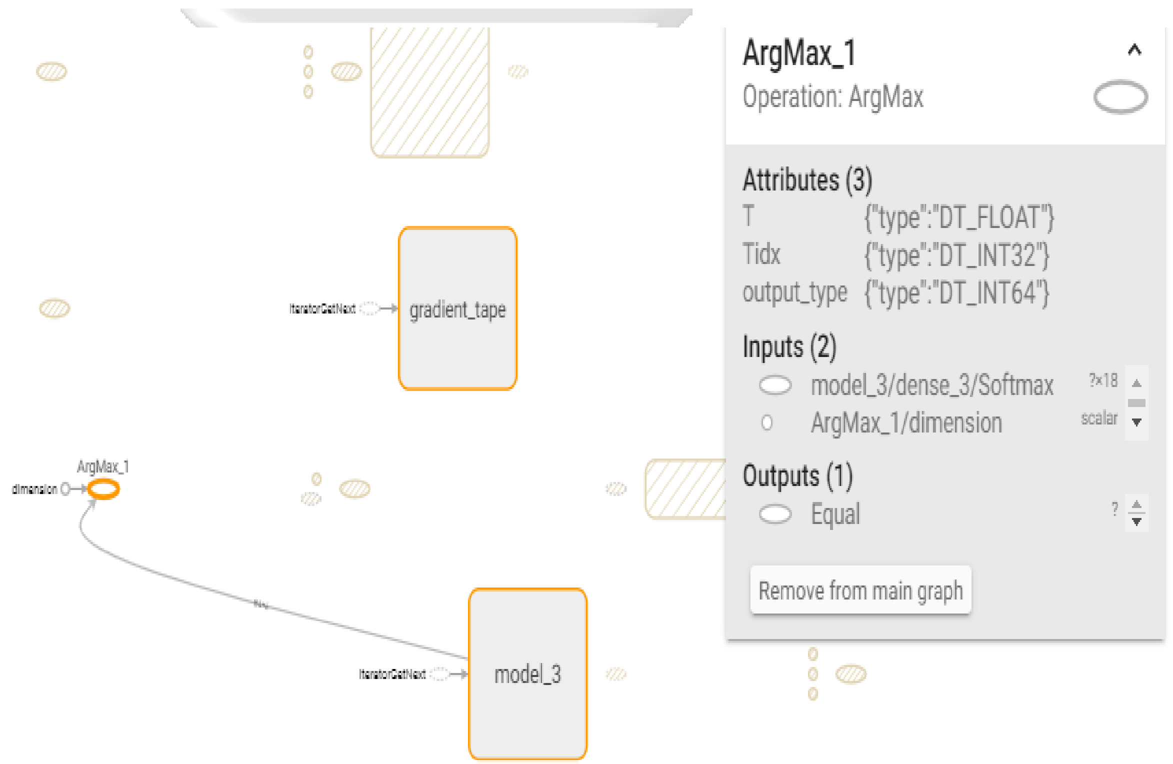

Figure 5. 3 Graphical Representation of the hidden layer

5.3.4 Output Layer

This is the last layer of the model and is responsible for performing the classification hence giving the final output of the model. SoftMax activation function was used for image classification. The categorical cross-entropy function was used to determine how close the predictions were to the actual labels. This was suitable as the model was a multiclass model. The graphical representation of the output/classification layer is shown in figure 5.4.

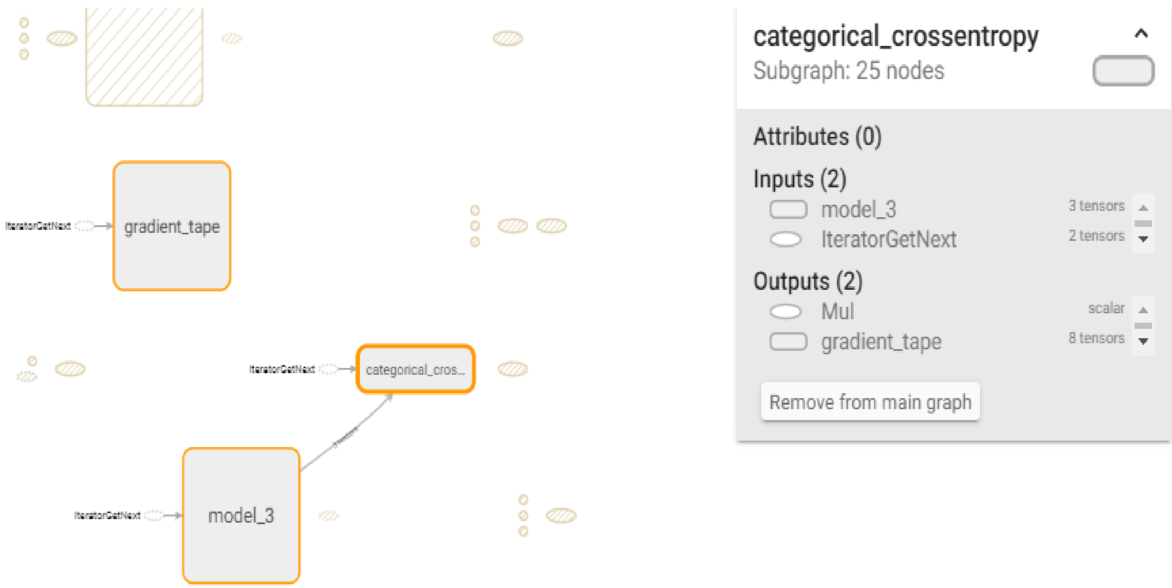

Figure 5. 4 Graphical representation of the output layer

5.4 Model Development

To develop the snake image classification model, the processes were mapped to the literature reviewed in chapter two. As a result, the following steps were taken.

5.4.1 Image Dataset Collection

Images were primarily collected from Kenya National Museum and Stedmark Garden snake park, Karen using a mobile phone and a digital camera. The images were then placed into corresponding folders whose folder name would be the image label. The folder name naming convention was “common name”. An excel file was created as shown in figure 5.5. It contained the local name of the snake species, the scientific name of the species, and the venom toxicity category that are present in that species. The rows of the excel file were mapped to the respective image folder using python code. This enabled the researcher to associate the common name, the scientific name, and the venom toxicity present in a species.

1	LOCAL NAME	SCIENTIFIC NAME	NON-VENOMOUS	HEMOTOXIC	CYTOTOXIC	NEUROTOXIC	CARDIOTOXIC
2	AFRICAN ROCK PYTHON	Python Sabae	1	0	0	0	0
3	BLACK MAMBA	Dendroaspis polylepis	0	0	0	1	1
4	BLACK NECKED SPITTING COBRA	Naja nigricollis	0	0	0	1	0
5	BROWN HOUSE SNAKE	Lamprophis fuliginosus	1	0	0	0	0
6	EAST AFRICAN GARTER SNAKE	Elapsoidea loveridgei	0	1	0	0	0
7	EGYPTIAN COBRA	Naja haje	0	0	0	1	0
8	GABOON VIPER	Bitis gabonica	0	0	1	0	0
9	GREEN MAMBA	Dendroaspis angusticeps	0	0	0	1	0
10	JACKSON'S TREE SNAKE	Thrasops jacksoni	1	0	0	0	0
11	JAMESON'S MAMBA	Dendroaspis jamesoni	0	0	0	1	0
12	LARGE BROWN SPITTING COBRA	Naja ashei	0	0	1	1	0
13	MT.KENYA BUSH VIPER	Atheris desaixi	0	0	1	0	0
14	PUFF ADDER	Bitis arietans	0	0	1	0	0
15	RED SPITTING COBRA	Naja pallida	0	0	1	1	0
16	RUFOUS BEAKED SNAKE	Rhamphisophis oxyrhynchus	0	1	0	0	0
17	SAND BOA	gongylophis colubrinus loveridgei	1	0	0	0	0
18	WESTERN KENYA FOREST COBRA	Naja melanoleuca	0	0	0	1	0
19	BOOM SLANG	Dispholidus typus	0	1	0	0	0

Figure 5. 5 Collected Data in Excel File

5.4.2 Import of necessary Library

The coding for the model was produced and implemented in Google Colab using the python programming language. Keras, Tensorflow, NumPy, and Matplotlib were the libraries utilized in the whole study. The backend of the framework was chosen as Tensorflow, and Keras has been utilized to offer additional built-in functionality such as activation functions, optimizers, layers, and so on. Keras API was used to enhance the dataset. NumPy is a Python library for mathematical evaluation. Confusion matrix, split train and test files, model checkpoint, callback mechanism, as well as other schematic representations like confusion matrix, the loss against epoch's graphs, accuracy against epochs curves, and many more, are all generated using Sklearn. The matplotlib library is also needed to create visual representations of the previously mentioned diagrams, such as the confusion matrix and classification report.

5.4.3 Fetch the image

The snake images were stored in a folder called processed_snake dataset on google drive. Snake species were categorized into 18 distinct classes based on their name. A total of 5 classes of venom toxicity categories were realized from a total image dataset of 4525. To fetch the images, the Colab was mounted on google drive as shown in the code snippet below.

```
# Mount Google Drive
from google.colab import drive
drive.mount ('/content/drive/')
```

Figure 5. 6 Python code for mounting the colab to the drive.

5.4.4 Data split

As indicated earlier in chapter three, the dataset was split into 70% training dataset and 30% for testing the model. To achieve this in every class, 30% of images were randomly selected to form the test data set and the remaining 70% formed the training dataset. A total of 3157 images become train data set and 1364 images became test data set

```
import splitfolders # Used to split dataset into training and validation
input_folder = data_dir
output = "/content/drive/MyDrive/processed_snake/final_data"
splitfolders.ratio(input_folder,output, seed=1337,ratio=(.7,.3)) # 70% training and 30% testing
```

5.4.5 Feature Reduction

Feature reduction was done on the training dataset using the singular value decomposition method. In SVD, color or luminance, texture, and edge information are sorted according to their significance. This information is integrated simultaneously and takes into account the relation between them. Hence, all the edge, color, and texture information are encoded into a single representation. In addition, there is no redundant information in SVD since left and right singular vectors are orthonormal. Furthermore, SVD takes into account human visual perception. In this algorithm, the features were derived by applying SVD directly to the raw images. Feature reduction is performed in a multi-granularity partitioning manner as shown in the equation 5.1 below.

$$\boldsymbol{A_i = \sum_{i-1}^{Z} \sigma_i \, U_i V_i^T} \tag{5.1}$$

Where z is the number of U_i (left singular vector) and V_i (right singular vector) pairs used. Each U_iV_i specifies a layer of the image geometry, whereas the singular value is the weight assigned to this layer and specifies the luminance of that image layer. The code snippet below was used to perform SVD on the training dataset.

```
for folder in folders:
    images = load_images_from_folder(folder)
    array_images = np.array(images)
    train_dat = np.reshape(array_images,(-1,1))


svd = TruncatedSVD(n_components=100)
train_data_svd = svd.fit_transform(train_dat) # Fitting svd on train data
```

Figure 5. 7 Python code for performing SVD on train dataset

5.4.6 Image Pre-processing and Augmentation

The images collected from the field were of varied sizes. The images were manually cropped to reduce the background noise. MobileNetV2 model image input requirement is normally 224x224 and therefore the images were resized to fit the model input requirements as shown in the code snippet below. The training dataset was augmented so that the model does not train on the same pictures more than once, to avoid overfitting. It was also used to increase the number of images in the training dataset. The model could not view the same image more than twice. The test dataset was not augmented because the images it contained were used for testing and validating the model. The augmentation operations were chosen experimentally to see the best fit. The final augmentation operations performed were, rotating the images by 40%, width shift range of 0.2, height shift range of 0.2, shear range of 0.2, zoom range of 0.2, horizontally flipping the images and Fill mode was set to the nearest. Augmentation enabled the researcher to create thousands of images from the original images while retaining label data. The code snippet below shows how data augmentation was achieved.

```python
# Data Augmentation

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   rotation_range=40,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True,
                                   fill_mode='nearest')
test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen .flow_from_directory(
                train_dir,
                target_size = IMAGE_SIZE,
                batch_size = BATCH_SIZE,
                class_mode = 'categorical',
                    )
testing_set = test_datagen.flow_from_directory(
                test_dir,
                target_size = IMAGE_SIZE,
                class_mode = 'categorical',
                batch_size = BATCH_SIZE,
                shuffle = False  )
```

Figure 5. 8 Python script for data augmentation

5.5 MobileNetV2 Retraining

In this study as earlier mentioned, MobileNetV2 pre-trained model was used for transfer learning and fine-tuning. Setting up all layers except the top and final layer of the model as trainable, the model was compiled using Adam optimizer with a learning rate of $1e^{-3}$ which is 0.001. Categorical cross-entropy was used as the loss function since there were more than two classes in this study. Accuracy was used as the metric to evaluate the model. All the output layer of the mobileNetV2 was flattened and passed to our final last dense layer. A final Dense layer with size 18 (number of snake species to be classified) and the activation function Softmax was added to the previously compiled pre-trained model for transfer learning. The model was trained using the model.fit_generator function which accepts the image generator defined earlier as an image input. The performance of the model during training is as indicated in figure 5. 9.

```
Epoch 1/20
99/99 [==============================] - 1801s 18s/step - loss: 3.0575 - accuracy: 0.7431 - val_loss: 12.6753 -
val_accuracy: 0.5345
Epoch 2/20
99/99 [==============================] - 1070s 11s/step - loss: 0.6307 - accuracy: 0.8907 - val_loss: 8.0910 -
val_accuracy: 0.6730
Epoch 3/20
99/99 [==============================] - 1092s 11s/step - loss: 0.4221 - accuracy: 0.9138 - val_loss: 3.8491 -
val_accuracy: 0.7801
Epoch 4/20
99/99 [==============================] - 1069s 11s/step - loss: 0.5733 - accuracy: 0.8993 - val_loss: 7.8324 -
val_accuracy: 0.5748
Epoch 5/20
99/99 [==============================] - 1076s 11s/step - loss: 0.4910 - accuracy: 0.9123 - val_loss: 4.9808 -
val_accuracy: 0.7141
Epoch 6/20
99/99 [==============================] - 1060s 11s/step - loss: 0.3240 - accuracy: 0.9366 - val_loss: 1.0138 -
val_accuracy: 0.8739
Epoch 7/20
99/99 [==============================] - 1095s 11s/step - loss: 0.2507 - accuracy: 0.9452 - val_loss: 0.2877 -
val_accuracy: 0.9633
Epoch 8/20
99/99 [==============================] - 1073s 11s/step - loss: 0.2361 - accuracy: 0.9525 - val_loss: 0.2458 -
val_accuracy: 0.9406
Epoch 9/20
99/99 [==============================] - 1096s 11s/step - loss: 0.2320 - accuracy: 0.9528 - val_loss: 2.4074 -
val_accuracy: 0.8270
Epoch 10/20
99/99 [==============================] - 1073s 11s/step - loss: 0.2587 - accuracy: 0.9424 - val_loss: 1.8552 -
val_accuracy: 0.7515
Epoch 11/20
99/99 [==============================] - 1092s 11s/step - loss: 0.1834 - accuracy: 0.9588 - val_loss: 0.1643 -
val_accuracy: 0.9663
Epoch 12/20
99/99 [==============================] - 1114s 11s/step - loss: 0.1325 - accuracy: 0.9671 - val_loss: 0.2101 -
val_accuracy: 0.9611
Epoch 13/20
99/99 [==============================] - 1079s 11s/step - loss: 0.1459 - accuracy: 0.9604 - val_loss: 0.0738 -
val_accuracy: 0.9773
Epoch 14/20
99/99 [==============================] - 1082s 11s/step - loss: 0.1978 - accuracy: 0.9569 - val_loss: 0.3118 -
val_accuracy: 0.9604
Epoch 15/20
99/99 [==============================] - 1079s 11s/step - loss: 0.3211 - accuracy: 0.9458 - val_loss: 0.2801 -
val_accuracy: 0.9348
Epoch 16/20
99/99 [==============================] - 1079s 11s/step - loss: 0.2551 - accuracy: 0.9544 - val_loss: 0.9635 -
val_accuracy: 0.9142
Epoch 17/20
99/99 [==============================] - 1097s 11s/step - loss: 0.5030 - accuracy: 0.9224 - val_loss: 10.3422 -
val_accuracy: 0.6224
Epoch 18/20
99/99 [==============================] - 1081s 11s/step - loss: 0.1941 - accuracy: 0.9569 - val_loss: 0.2201 -
val_accuracy: 0.9633
Epoch 19/20
99/99 [==============================] - 1088s 11s/step - loss: 0.0921 - accuracy: 0.9759 - val_loss: 0.1466 -
val_accuracy: 0.9839
Epoch 20/20
99/99 [==============================] - 1081s 11s/step - loss: 0.1383 - accuracy: 0.9652 - val_loss: 0.2605 -
val_accuracy: 0.9553
```

Figure 5. 9 Model Epochs

5.6 Hyperparameter tuning

This was done by adjusting mainly three variables: the optimization, the learning rate, and the number of epochs. The learning rate took the value of 0.001. The optimizer used was Adam. The epochs used in training consisted of 20 since this was going to train fast and our dataset was not enormous and hence will in turn take care of under fitting or overfitting of the model. This was possible because of the use of Colab which gave accessibility to its GPUs. Hyperparameters determined how a model is structured. They can be used for controlling system requirements such as the cost on memory which was an important aspect to a model. The hyperparameters adjusted in this model are as follows:

(i) Learning Rate: this is the most important of all hyperparameters. It assigns a quantity to the progress of learning. The learning rate controls the rate or speed at which the model learns. If the learning rate is of a low value, updates are few and optimization speed is reduced whereas a high value, the optimization diverges. The final learning rate used for the model was 0.001.

(ii) Epochs: this is a parameter that separates the training into different stages. The higher the number of epochs, the higher the probability of having an overfitted model. The epochs used for the model were 20.

(iii) Optimizer: adaptive moment estimation Adam is the most commonly used for deep learning models for image classification. It works in such a way that it calculates an exponential moving average of the gradient and the squared gradient.

5.7 Model Validation

The testing and validation of the model was done using a confusion matrix. This will indicate the accuracy and validation of the model. The graphs below show how the model was validated.

The progress of the model training is shown in Figure 5.10. The training accuracy was higher than the validation accuracy. The validation accuracy curve had several dips. This showed the model did not overfit the data. The performance was tracked over time. The x and y-axis represent epochs and accuracy values respectively.

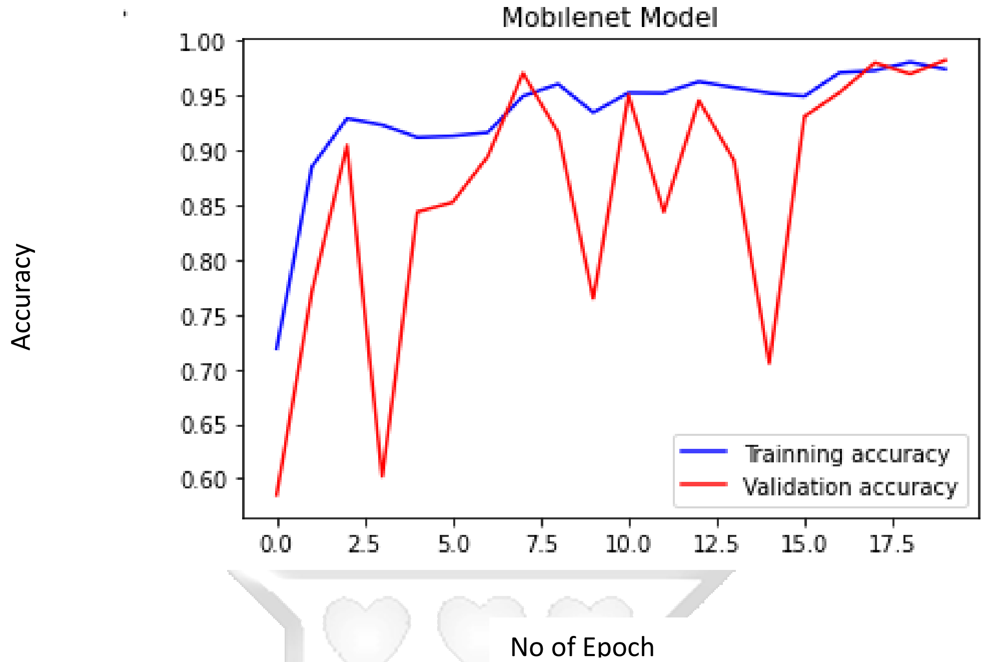

Figure 5. 10 Training accuracy vs validation accuracy.

The progress of the model as it trains as shown in Figure 5.11 indicate that, the training loss was lower than the validation loss. The validation loss curve has several dips. The performance was tracked over time. The x and y-axis represent epochs and loss values respectively.

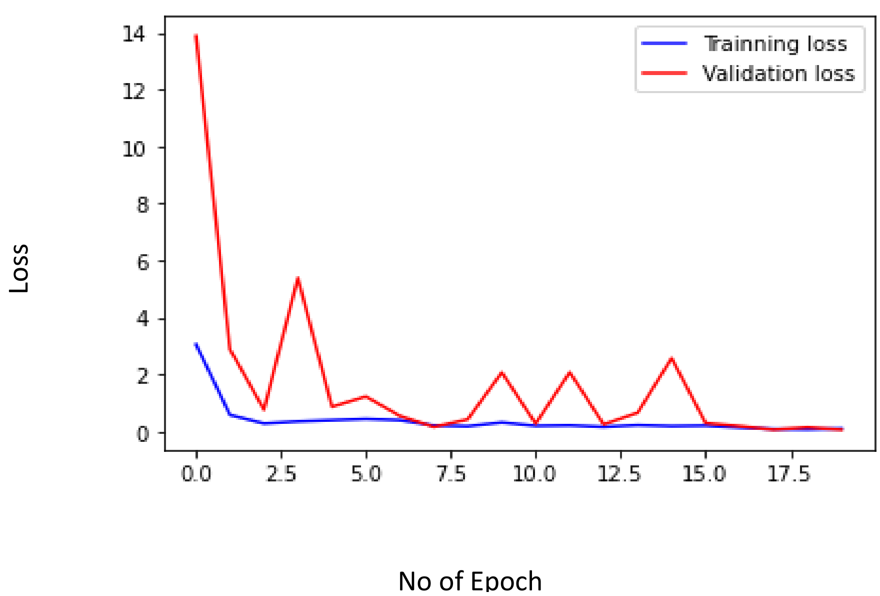

Figure 5. 11 Training Loss vs Validation Loss

5.8 Converting Tensor flow Model to Tensor flow Lite

The generated Tensorflow model was converted to a Tensorflow lite model using Tensorflow's Lite converter tool, tflite_convert. The code snippet below shows how the generated model was converted into a Tensorflow lite model.

```
# Convert the model #3
converter = tf.lite.TFLiteConverter.from_saved_model(export_dir)
converter.optimizations = [optimization]
tflite_model = converter.convert()
```

Figure 5. 12 Python script for converting the model to TFLite

The input graph is the Tensorflow model and the output file is the desired Tensorflow lite file. The input and output array are as per the graphical representation of the MobileNetv2 model architecture shown in Figure 5.1 above.

5.9 Android Mobile Application Development

5.9.1 Front End Development

Android studio was installed and initiated. The android studio dependencies packages were installed. To build for android, android studio IDE was started and Java code was written for the android application. Firstly, android design was done with XML, then Java code was written to connect to XML design. XML is a popular language used for creating user interfaces in Android applications. The resulting android APK file was then installed on a test mobile device. The mobile application was developed using Java 11, XML, and Groovy.

5.9.2 Login and Registration

The user must have registered for an account and logged into the system in order to classify a snake image and retrieve the venom type, first aid guideline and the recommended antidote. The user does not need to login to classify subsequent images. The user requires internet access to register or login. During registration, the user provides a valid email address and a password. The email address must be unique. The email and password are sent to the cloud database for verification. If the email is valid and unique and the password is more than six characters then the system stores the user's registration data. The system creates a unique user ID for each registered user. The password is hashed before being saved to increase system security. Figure 5.13 below shows screenshots of the registration and login screens

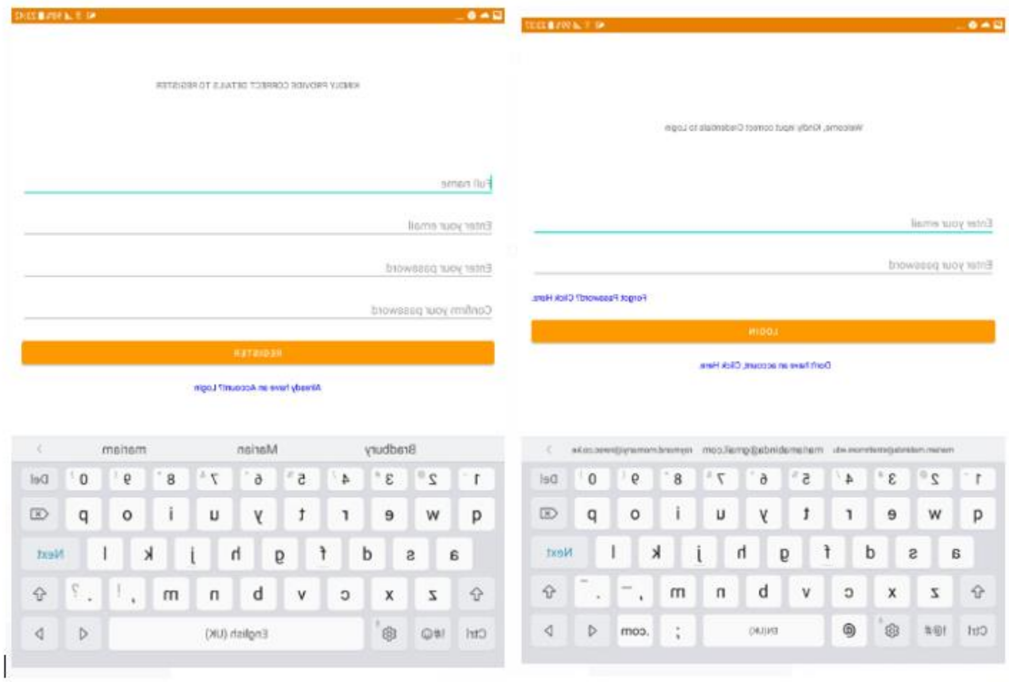

Figure 5. 13 Login and Registration Screenshot

5.9.3 Choosing an Image

Android image-picker, an android module that enables you to use a mobile interface to choose media from the mobile device library or the device's camera, was used to enable the user to choose an image. The module returns the chosen image path or image URL, which is passed to the image classifier for classification. Figure 5.14 below shows a screenshot of the image picker in action.

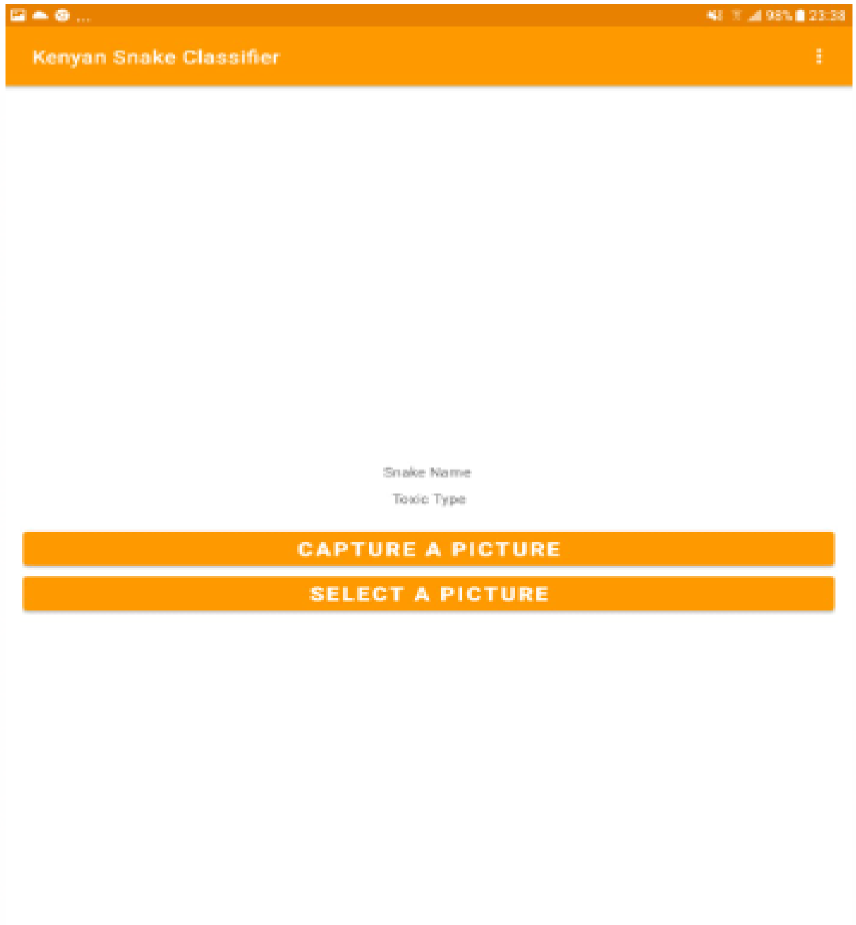

Figure 5. 14 Chose /select image Screenshot

5.9.4 Image Classification

The MobileNetV2 model developed was embedded into the mobile application using a TensorFlow-lite module for android application in android studio IDE. The module was initiated with the tflite model and its corresponding labels text file. The image path of the image chosen by the user was then used as an input to the module. The output obtained contains the snake species name and the venom type. The same screen has a view antidote button and First Aid guidelines button to make it convenient for the user to quickly access this important information. The capture and select image button are maintained on the same screen in case there is a need for recapture of the image.

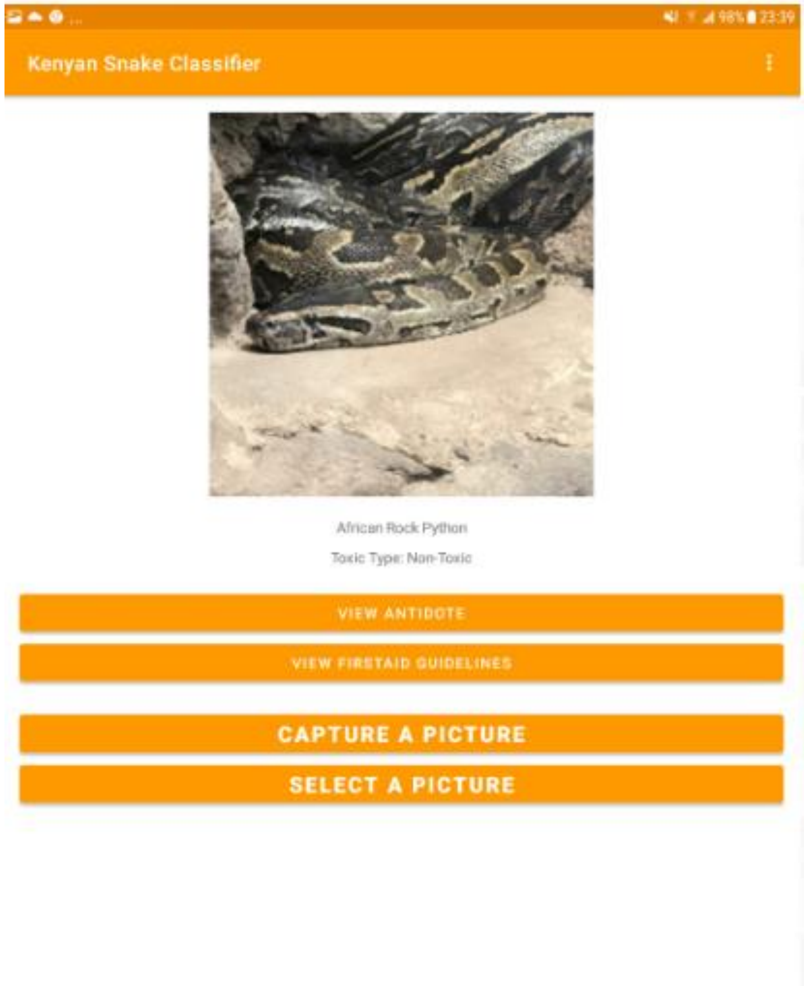

Figure 5. 15 Classification Screenshot.

5.9.5 Viewing Antidote and First Aid Guidelines

A registered user was able to view the first aid guidelines and the recommended antidote based on the snake type and the venom toxicity category. The user would tap on the “First Aid Guidelines” button on the screen shown in Figure 5.16 and a screen with outlined first aid guidelines will be displayed.

Figure 5. 16 First Aid Guidelines

Subsequently, the user would tap on the "View Antidote" button on the screen shown in Figure 5.17 and a screen with the recommended antidote will be displayed.

Figure 5. 17Antidote Information

5.10 Model Testing

The evaluate function is used to test the model accuracy across the test data that is stored in the test directory. The model achieved an accuracy of 96%. Functional testing was carried out on the model to certify the model's ability to meet the basic functional requirements. Non-functional testing test cases such as reliability and supportability testing were also carried out on the application prototype. Table 5.3 shows how the functional and non-functional requirements were achieved.

Table 5. 2 Functional and Non-functional Testing.

Test check	**Inspection Check**	**Priority**
Functional	Does the application allow a first-time user to register?	High
Functional	Does the application allow the user to upload an image by either capturing or from the library?	High
Functional	Does the application allow images in jpeg format?	High
Functional	Does the model predict the snake species in the input image?	High
Functional	Does the application classify snake species according to their venom toxicity category?	High
Supportability	Does the model support different image format	Moderate
Reliability	Does the model provide consistent results given the same input?	High
Reliability	Does the model provide warning messages to the user in case of wrong login data?	Moderate

Chapter 6: Discussions

6.1 Introduction

This chapter discusses the findings of the research, reviews the solution that was developed, and tries to find out whether the objectives of the research were achieved. This research study investigated the challenges associated with managing snakebite envenomation. Further, it analyzed machine learning techniques that support image classification. After reviewing these machine learning techniques, the researcher chose to focus on transfer learning of a deep convolutional neural network that is well suited for image processing tasks specifically MobileNetV2. An Android mobile application, with an embedded convolutional neural network that is suited for mobile devices that have constrained computing resources, was developed.

6.2 Model Validation

This study employed transfer learning in which a pre-existing convolutional network, MobileNetV2 in particular, was retrained and fine-tuned for snake species classification based on the venom toxicity profiles. The researcher used an image dataset of 4521 images. These images were primarily collected from Kenya National Museum and Stedmark Garden snake park, Karen as discussed in the research methodology section above. 70% of the images were used to create the training dataset. The remaining 30% of the images were used for validation and testing. The true performance of the model was measured by measuring its performance on the test dataset, which was not contained in the training data. The testing accuracy represents how well the trained model would classify completely new images.

The three metrics used to evaluate the performance of the model included accuracy, precision, and recall. The accuracy is simply defined as the fraction of correct predictions of the model to the total number of the predictions. Accuracy can also be calculated in terms of positives and negatives where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. The final test accuracy achieved was 96%. The test accuracy shows what percent of images used in the testing process were labeled with the correct label. From Figure 5.11 above it was shown that the loss function was dropping to almost zero. Thus, high test accuracy and a low loss function indicate that the model learned from the training data and can thus classify new images.

6.2.1Precision

Precision is the presentation of the accuracy and positive predictions of each class given by the equation 6.1 below.

$$\boldsymbol{PRECISION} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad \textbf{(6.1)}$$

where TP stands for True positive and FP stands for False positive. Figure 6.1 shows the precision metrics from the model.

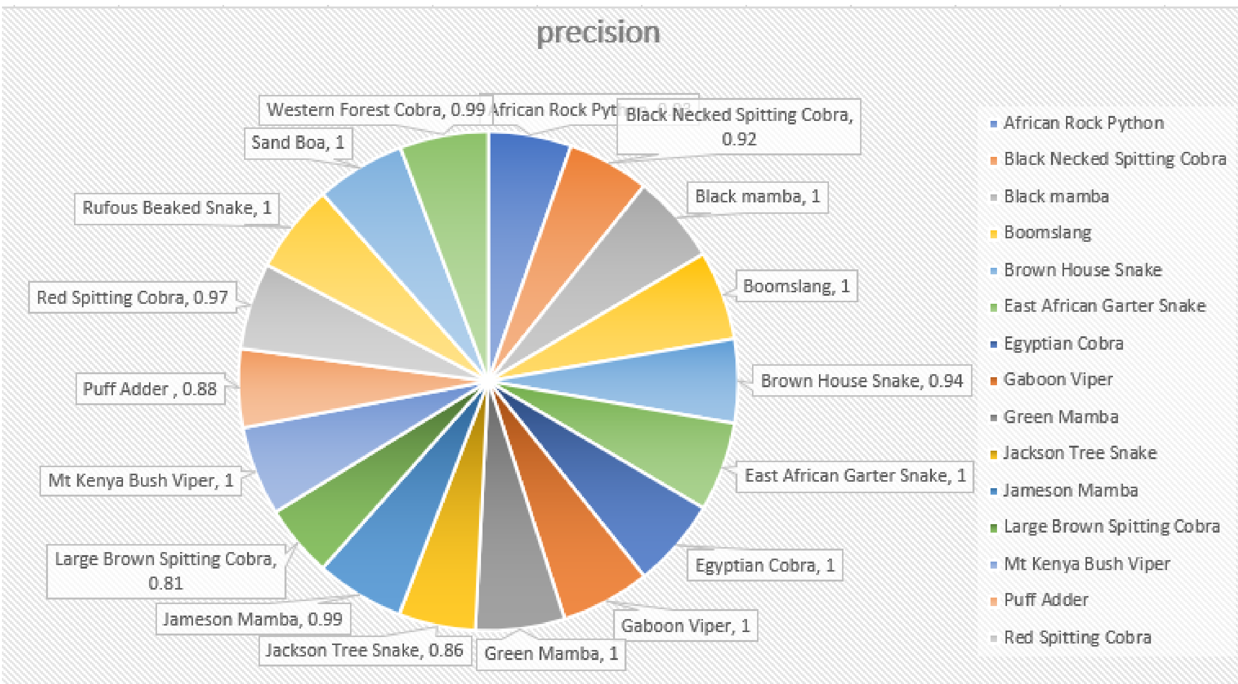

Figure 6. 1 Precision Metrics for the Model

Most of the snake species like Gaboon viper, Boomslang, Black mamba, and all the snake species that have a 1 presented the highest precision score because they were the most correctly predicted. Large Brown spitting cobra performed worst in terms of prediction and presented the worst precision score of 81%. Precision alone might not present the best conclusions about the model performance, therefore recall was introduced to back it up (Géron, 2019).

6.2.2 Recall

Recall also known as sensitivity is the fraction of test images from a class A that are correctly identified to be assigned to the class A. This gives the ratio of positive instances which are correctly identified by the model. It is given by the equation 6.2 below

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \qquad \textbf{(6.2)}$$

where FN stands for false negative. Figure 6.2 shows the recall metrics from the model

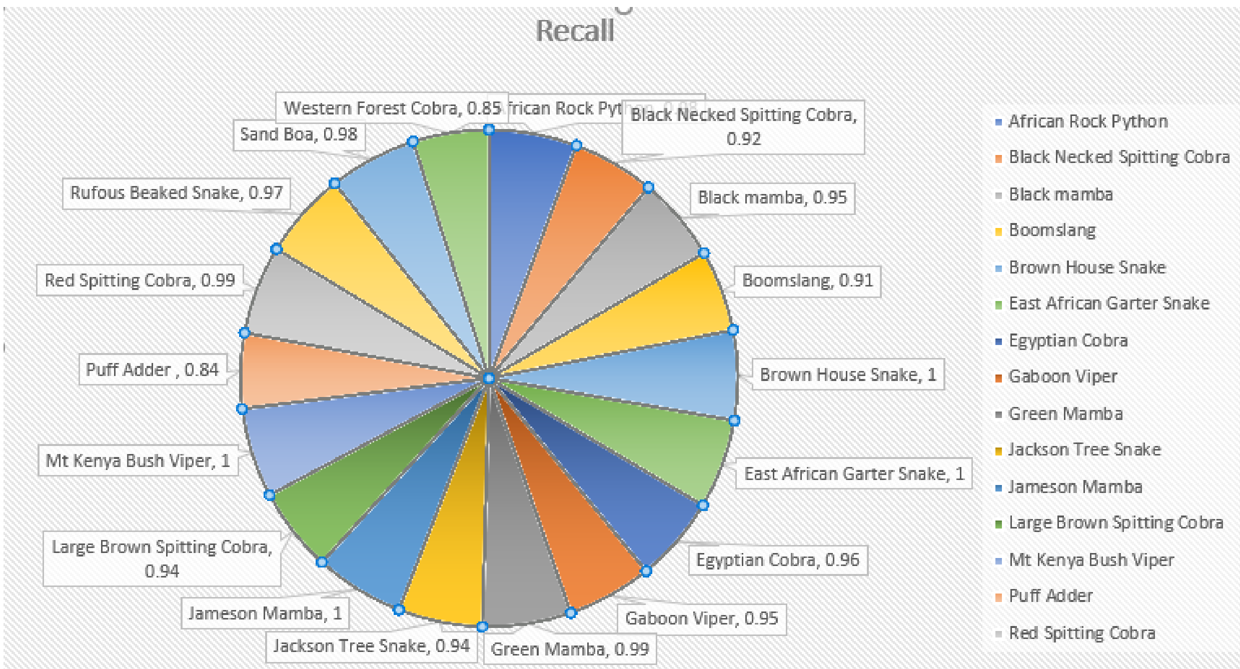

Figure 6. 2 Recall Metrics for the Model

Jameson Mamba, Mt. Kenyan Bush Viper, Brown House Snake, and East African Garter snake presented the highest recall score because they were the most correctly predicted. Puff Adder performed worst in terms of prediction and presented the worst recall score of 84%.

6.2.3 F1-Score

The average of the precision and recall could be interpreted as an F1 score, having its best value at 1 and worst value at 0. This gave the harmonic mean of the precision and recall values. Harmonic mean assigns more weight to low values. Whereas ordinary mean treats all values equally. Therefore, a target gets a high F1 score if both its precision and recall are high. The F1 score is given by equation 6.3 and shown in Figure 6.3 below.

$$\boldsymbol{F1} = \frac{2}{\frac{1}{precision}+\frac{1}{recall}} = 2x\frac{precision\ x\ recall}{precision+recall} = \frac{TP}{TP+\frac{FN+FP}{2}} \qquad \textbf{(6.3)}$$

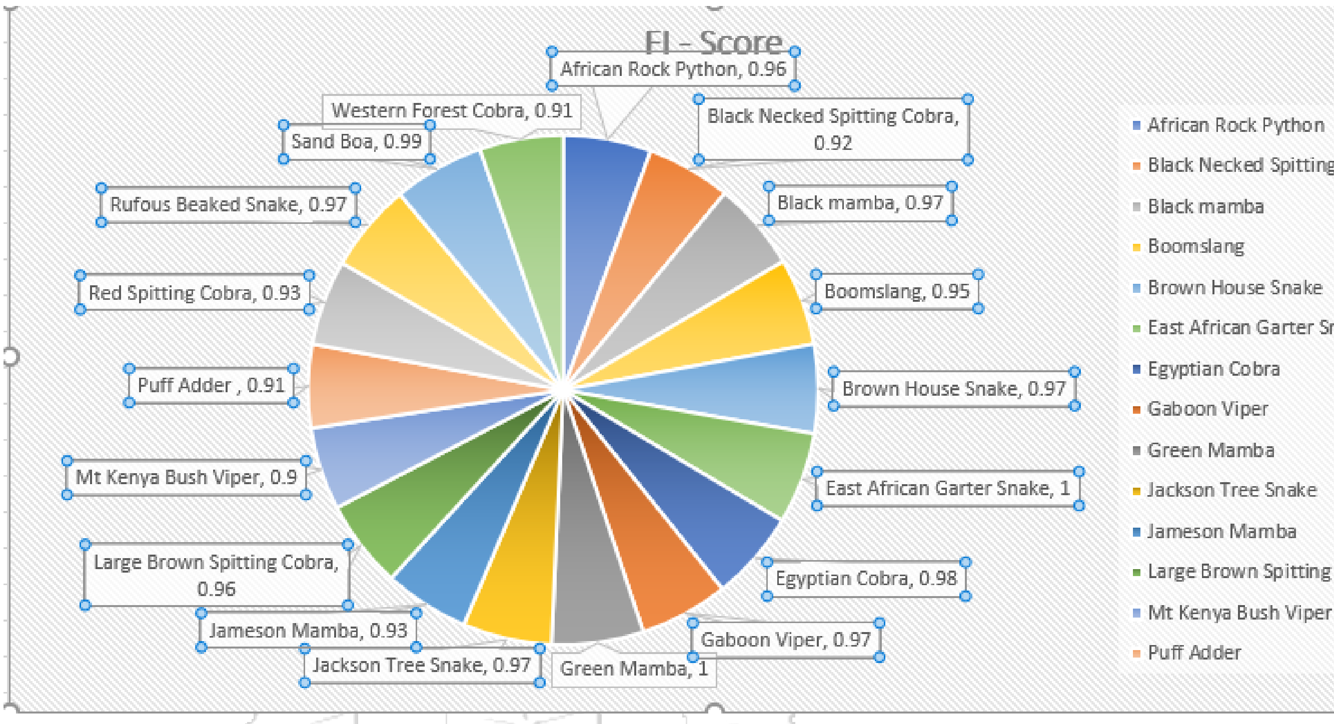

Figure 6. 3 F1 Score Metrics for the Model

The definition of true positive, true negative, false positive, and false negative in the context of the study is as follow :

i. True Positive (TP): Correctly identified prediction for each snake class.

ii. True Negative (TN): Correctly rejected prediction for certain snake classes.

iii. False Positive (TP): Incorrectly identified predictions for certain snake classes.

iv. False Negative (TP): Incorrectly rejected predictions for the snake classes.

6.3 Classification Report

A classification report was generated from the confusion matrix using the sklearn.metrics.classification_report function from the sklearn library. The number of images used per class should be almost the same to avoid data imbalance which affects precision and recall. False positive will be caused by species of less image. In this research, Precision and recall are almost similar indicating that there are fewer variations in the number of images in the different classes.

The classification report is shown in figure 6.4.

Classification Report

	precision	recall	f1-score	support
African Rock Python	0.93	0.98	0.96	44
Black Necked Spitting Cobra	0.92	0.92	0.92	39
Black mamba	1.00	0.95	0.97	111
Boomslang	1.00	0.91	0.95	46
Brown House Snake	0.94	1.00	0.97	64
East African Garter Snake	1.00	1.00	1.00	39
Egyptian Cobra	1.00	0.96	0.98	70
Gaboon Viper	1.00	0.95	0.97	76
Green Mamba	1.00	0.99	1.00	129
Jackson Tree Snake	1.00	0.94	0.97	18
Jameson Mamba	0.86	1.00	0.93	115
Large Brown Spitting Cobra	0.99	0.94	0.96	130
Mt Kenya Bush Viper	0.81	1.00	0.90	61
Puff Adder	1.00	0.84	0.91	57
Red Spitting Cobra	0.88	0.99	0.93	100
Rufous Beaked Snake	0.97	0.97	0.97	119
Sand Boa	1.00	0.98	0.99	48
Western Forest Cobra	0.99	0.85	0.91	98
accuracy			0.96	1364
macro avg	0.96	0.95	0.96	1364
weighted avg	0.96	0.96	0.96	1364

Figure 6. 4 Classification report of the model.

With reference to the classification report in figure 6.4, the model achieved an aggregated precision of 0.96, a recall of 0.95 and an f1-score of 0.9. Under the support column, we have the number of images per class used for validation. The figure 1364 denotes the total number of images used for validation. A precision of 0.96 implies that 96% of the prediction results returned by the model were relevant/correct. This indicates a good proportion of the relevant results by the model. The aggregate recall of 0.95 indicates good performance of the model in correctly identifying the relevant instances of the snake classes. An aggregate f1-score of 0.96 achieved by the model implies that the model achieved both higher recall and precision as f1-score is the harmonic mean of the two. This results in good accuracy in making the predictions.

With regard to these results, the model achieved a good classification accuracy of 96% and generalized well on the validation data. The ability of the model to generalize well on unseen data is indicated by the narrow gap between the training and the validation accuracy as indicated in figure 5.11. This in turn indicates a low overfitting effect by the model.

6.4 Research Contribution

The model provided a more efficient way of classifying snake species using their images. Classification of the snake species assists in fast and correct identification of the snake and in turn facilitate the doctors to manage snake envenomation efficiently. Further, it also gives the victim expert information and even provides guidelines on first aid. This eventually leads to a reduction in the number of deaths and disabilities caused by snake envenoming.

Automatic classification of the biting snake eliminated the use of expert knowledge which is important, rare and necessary for the diagnosis of the venom toxicity present in the snake and consequently in the administration of the antidote. This will also contribute to the reduction of the cost of treatment of snakebite envenoming e.g., the cost associated with the testing for venom presence in the hospitals. The primary data collected from the National Museum of Kenya and Stedmark snake park garden, Karen, also contributed to data availability for further research to academics interested in similar research.

6.5 First Aid Details

The researcher was able to add the first aid details button to the snake classification application. This enabled the user to get extra details i.e., a brief description guideline on the best first aid to give a victim, based on the toxin, without having to leave the system. This was done to enhance the user experience.

6.6 Accomplishment of the Objectives

In this section, a summary of how the objectives of this research were achieved. All the objectives were outlined in section 1.4 and their achievements are mentioned as follows:

The first objective was to investigate the challenges associated with managing snakebite envenomation. A review of the literature on problems associated with snakebite envenoming was done. Most of the problems highlighted lack of expert knowledge and use of alternative methods to manage envenoming due to the prohibitive cost associated with testing on health facilities. This problem narrows down to how best can the victim classify the biting snake and even know the venom the snake produces and the first aid measure as they find their way to the nearest health facility. If the venom toxicity category is well known, then the cost associated with the test is reduced. Also, by performing the right first aid, chances of adverse effects caused by neglect in the field are reduced. Further, the fact this expert knowledge is disseminated to the victim, it makes it easy for the victim to avoid confrontation with the biting snake.

The second objective was to analyze machine learning techniques that support image classification. To appraise the feature reduction techniques with an intention of increasing the accuracy of image classification models. This fed into the third objective which was to develop an in-image classification of snake species. Their review in chapter 2 helped the researcher to understand the existing solutions that are currently in use and how their application has helped envenomation management. The models in the review were noticed to have been applied to several other datasets and different types of research. This enhanced the scope of the research and common areas of interest to researchers. Their weaknesses and strengths were discussed and their shortfalls aided in the development of this model used for snake image classification. The main model was the MobileNetV2 network which was applied in this research combined with a feature reduction technique called SVD. It was also noted that the use of human vision is far much less efficient compared to computer vision due to human limitations and lack of prior expert knowledge of snake species and venom toxicity category.

The fourth objective was to develop a snake image classification model for classifying snake species for envenomation management. The classifier model was implemented in an automatic way which was the main objective of the study. The research methodology was clearly stated in chapter 3 on how the development was done. The system analysis then followed in chapter 4 on how the victims and the system interacted. The final implementation was clearly stated in chapter 5. This gave an account of how the model was built, trained, and tested.

The model output made it easier for victims without expert knowledge to be able to classify snake species based on their venom toxicity profile and further recommended first aid guidelines and anti-venom to be used to the user. There is a direct correlation between the snake class, venom type, and anti-venom. Incorrect classification of the snake species can cost someone their life or disability. The images that were fed into the model presented similar characteristics as those written in the literature review.

The model developed in this research made the classification of the snake images more accurate as opposed to the methods described in the literature review. The use of SVD in combination with MobileNetV2 gives a higher accuracy indicating the classification was accurate.

The researcher tested the validity of the developed model. The model was tested and achieved an accuracy of 96%. This accuracy level was good and acceptable compared to existing snake image classification models. The testing was done using the test dataset which was unseen data to the model. The validity was carried out using 3 metrics; the precision, recall, and F1 score. All these metrics' results were outputted in pie chart form for the researcher to analyze the model performance.

6.7 Advantages of the Developed System to Current Systems

Currently, snake classification systems that have been done focus on Asian or Indian snake species. The developed snake image classification system provided a solution to anyone wanting to determine the Kenyan snake species in an image and its venom toxicity for antidote facilitation. The user feeds the system an image and the system outputs a result showing the snake's common name and the venom toxicity of the snake classified. The system outputs the snake's common name and the toxin category of the displayed snake. There is a first aid button that gives the user guidelines on how to undertake the first aid safely. Furthermore, there is a view antidote button that will give the general recommended antidote depending on the venom toxicity category. This improves the user experience, especially for beginners. This research study has demonstrated that the system user does not need to have expert knowledge in identifying and classifying snake species but rather have an image of the species for easy identification and classification. Finally, the developed system included a first aid feature and view antidote feature, which none of the other systems have and would play a big role in the management of snakebite envenoming.

6.8 Shortfalls of the Research

The mobile-based image recognition system developed during this research study did not meet some of the expectations of the researcher.

i. The classifier would fail to correctly classify an image if two or more snake species were in the same photo.
ii. The image dataset did not have many different angles of the snake species as earlier anticipated.
iii. There will be a need for the doctor to be notified to recommend the antidote based on the classification result which has not been included in the scope of this study.
iv. There is no push notification to notify the doctor of a pending request for recommending an antidote.

6.9 Challenges Encountered

It was a challenge getting good quality snake images, especially the most venomous ones like the black mamba and the Gabon viper since they were constantly moving hence having most of the images blurred and unusable. Some snake species were not included in this prototype because the research could not get clear images because they hide under the sand or camouflage making it difficult to spot them. It was time-consuming sifting the collected images and remove unwanted or low-quality images and sorting the collected images into their respective folders.

Chapter 7: Conclusion and Recommendation

7.1 Conclusions

The objective of this research study was to develop a mobile application that can classify Kenya snake species from an image and identify the venom toxicity present in it. The problem of image visual classification is difficult for both humans and machines alike. Humans can easily identify a snake in an image but find it difficult to determine the snake species in the image and even more difficult to know the venom toxicity of a certain species. Thus, this study began by investigating the challenges associated with managing snakebite envenomation. Further, it analyzed machine learning techniques that support image classification. After reviewing these machine learning techniques, the researcher chose to focus on transfer learning of a deep convolutional neural network that is well suited for image processing tasks specifically MobileNetV2.This study showed that advances in computer vision have been able to create deep neural networks that can effectively and efficiently perform image classification. This research then presented a detailed approach on how to solve the problem of snake species classification using transfer learning and came up with a mobile application for classifying snake species using their images based on venom toxicity profiles. The model developed performed satisfactorily well and was able to predict the snake species in an image provided by the user and classify it based on the venom toxicity and further provide first-aid details to the user using the snake species classification result. The user could then save the classification results into a cloud database and later on retrieve the same in form of a report or information. Combining the classification of snakes based on the venom toxicity, giving first aid details and guidelines in a single application is a novel approach introduced in this research study

7.2 Recommendations

Based on the results of this research study, the researcher recommends the following

i. Include images of snakes that have camouflaged to their environment and classify them using these images.
ii. Show possible suggestions if the snake species cannot be correctly identified.

iii. Include images of snakes that have either hidden their head under their bodies and classify the image without having the head visible.

7.3 Future Research Direction

More research needs to be done on improving the snake image classification capabilities of the developed model. Also, future research can explore the classification of multiple different types of snake species in an image. Future research can also look into using visual body scales and patterns of the snake to correctly classify a snake species without having the species head visible. Finally, a global positioning system feature can be added to the application that will show the victim the nearest health facility from the actual geographical location of the victim.

References

Abayaratne, S. B., Ilmini, W. M. K. S., & Fernando, T. G. I. (2019). *Identification of Snake Species in Sri Lanka Using Convolutional Neural Networks*. *December*, 101–110.

Abdurrazaq, I. S., Suyanto, S., & Utama, D. Q. (2019). Image-Based Classification of Snake Species Using Convolutional Neural Network. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 97–102. https://doi.org/10.1109/ISRITI48646.2019.9034633

Ahmed Medjahed, S. (2015). A Comparative Study of Feature Extraction Methods in Images Classification. *International Journal of Image, Graphics and Signal Processing*, *7*(3), 16–23. https://doi.org/10.5815/ijigsp.2015.03.03

Ameade, E. P. K., Bonney, I., & Boateng, E. T. (2021). Health professionals' overestimation of knowledge on snakebite management, a threat to the survival of snakebite victims—a cross-sectional study in Ghana. *PLoS Neglected Tropical Diseases*, *15*(1), 1–18. https://doi.org/10.1371/journal.pntd.0008756

Bhargava, S., Kumari, K., Sarin, R. K., & Singh, R. (2020). First-hand knowledge about snakes and snake-bite management: an urgent need. *Nagoya Journal of Medical Science*, *82*(4), 763–774. https://doi.org/10.18999/nagjms.82.4.763

Bloch, L., Boketta, A., Keibel, C., Mense, E., Michailutschenko, A., Pelka, O., Rückert, J., Willemeit, L., & Friedrich, C. M. (2020). Combination of image and location information for snake species identification using object detection and EfficientNets FHDO Biomedical Computer Science Group (BCSG). *CEUR Workshop Proceedings*, *2696*, 22–25.

Bolon, I., Durso, A. M., Mesa, S. B., Ray, N., Alcoba, G., Chappuis, F., & De Castañeda, R. R. (2020). Identifying the snake: First scoping review on practices of communities and healthcare providers confronted with snakebite across the world. *PLoS ONE*, *15*(3), 1–24. https://doi.org/10.1371/journal.pone.0229989

Choe, M., Yoo, J., Lee, G., Baek, W., Kang, U., & Shin, K. (2022). *MiDaS: Representative Sampling from Real-world Hypergraphs*. https://doi.org/10.1145/1122445.1122456

David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., & Warden, P. (2021). *TENSORFLOW LITE MICRO: EMBEDDED MACHINE LEARNING ON TINYML SYSTEMS*.

Farajzadeh, K., Zarezadeh, E., & Mansouri, J. (2017). Concept detection in images using SVD features and multi granularity partitioning and classification. *Journal of Information Systems and Telecommunication*, *5*(3), 172–182.

Joseph, R. S. (2021). *Plant Disease Detection A Review on Various Image Processing Technologies and ML*. *9*(13), 67–71.

Kamalraj, R. (2020). Deep learning model for identifying snakes by using snakes' bite marks. *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*, 2020–2023. https://doi.org/10.1109/ICCCI48352.2020.9104200

Kaur, T., & Gandhi, T. K. (2020). Deep convolutional neural networks with transfer learning for automated brain image classification. *Machine Vision and Applications*, *31*(3). https://doi.org/10.1007/s00138-020-01069-2

Krishnan, M. G. (2020). Impact of Pretrained Networks for Snake Species Classification. *CEUR Workshop Proceedings*, *2696*, 22–25.

Krishna, S. T., & Kalluri, H. K. (2019). Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering*, *7*(5), 427–432.

Mahajan, A., & Chaudhary, S. (2019). Categorical Image Classification Based on Representational Deep Network (RESNET). *Proceedings of the 3rd International Conference on Electronics and Communication and Aerospace Technology, ICECA 2019*, 327–330. https://doi.org/10.1109/ICECA.2019.8822133

Mateen, M., Wen, J., Nasrullah, Song, S., & Huang, Z. (2019). Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry*, *11*(1). https://doi.org/10.3390/sym11010001

MOH-Kenya. (2019). *Guidelines for Prevention Diagnosis and Management of Snakebite Envenoming in Kenya Government of Kenya*. 1–40.

Muguti, G. I. (2003). Management of snake bites. *The Central African Journal of Medicine*, *49*(11–12), 142–144. https://doi.org/10.1136/emj.14.5.349-b

Nag, A., & Karforma, S. (2016). *Classifying Snake Toxins Using Decision Trees*. *2277*, 5–6.

Nath, S. S., Mishra, G., Kar, J., Chakraborty, S., & Dey, N. (2014). A survey of image classification methods and techniques. *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, 554–557. https://doi.org/10.1109/ICCICCT.2014.6993023

Niteesh., I., Mohith, M. S., Vahed., S., Dattu.p, N., & Srilatha., M. (2021). Classification and Prediction of Snake Species based on Snakes' Visual Features using Machine Learning. *2021 2nd Global Conference for Advancement in Technology, GCAT 2021*, 2–6. https://doi.org/10.1109/GCAT52182.2021.9587711

Nyaga, G. M. (2019). *A Mobile-based Image Recognition System for Identifying Bird Species in Kenya*.

Paul, R. (n.d.). *A Computer vision-based model for tomato plant nutrient and disease classification*. http://hdl.handle.net/11071/12134Followthisandadditionalworksat:http://hdl.handle.net/11071/12134

Perera, P., & Patel, V. M. (2019). Deep transfer learning for multiple class novelty detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 11536–11544. https://doi.org/10.1109/CVPR.2019.01181

Progga, N. I., Rezoana, N., Hossain, M. S., Islam, R. U., & Andersson, K. (2021). A CNN Based Model for Venomous and Non-venomous Snake Classification. In *Communications in Computer and Information Science* (Vol. 1435). Springer International Publishing. https://doi.org/10.1007/978-3-030-82269-9_17

Pucca, M. B., Knudsen, C., Oliveira, I. S., Rimbault, C., Cerni, F. A., Wen, F. H., Sachett, J., Sartim, M. A., Laustsen, A. H., & Monteiro, W. M. (2020). *Current Knowledge on Snake Dry Bites*. 4–7.

Rajabizadeh, M., & Rezghi, M. (2021). A comparative study on image-based snake identification using machine learning. *Scientific Reports*, *11*(1), 1–16. https://doi.org/10.1038/s41598-021-96031-1

Ribani, R., & Marengoni, M. (2019). A Survey of Transfer Learning for Convolutional Neural Networks. *Proceedings - 32nd Conference on Graphics, Patterns and Images Tutorials, SIBGRAPI-T 2019*, 47–57. https://doi.org/10.1109/SIBGRAPI-T.2019.00010

Rusli, N. L. I., Amir, A., Zahri, N. A. H., & Ahmad, R. B. (2019). Snake species identification by using natural language processing. *Indonesian Journal of Electrical Engineering and Computer Science*, *13*(3), 999–1006. https://doi.org/10.11591/ijeecs.v13.i3.pp999-1006

Russell, J. J., Schoenbrunner, A., & Janis, J. E. (2021). Snake Bite Management: A Scoping Review of the Literature. *Plastic and Reconstructive Surgery - Global Open*, 1–12. https://doi.org/10.1097/GOX.0000000000003506

Sanghvi, K., Aralkar, A., Sanghvi, S., & Saha, I. (2021). A Survey on Image Classification Techniques. *SSRN Electronic Journal*, *January*. https://doi.org/10.2139/ssrn.3754116

Sarker, I. H., Kayes, A. S. M., & Watters, P. (2019). Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0219-y

Seok, J., Hyun, B., Kasa-Vubu, J., & Girard, A. (2012). Automated classification system for bone age X-ray images. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 208–213. https://doi.org/10.1109/ICSMC.2012.6377701

The MITRE Corporation. (2014). *Systems Engineering Guide*.

Vasmatkar, M., Zare, I., Kumbla, P., Pimpalkar, S., & Sharma, A. (2020). Snake species identification and recognition. *2020 IEEE Bombay Section Signature Conference, IBSSC 2020*, 1–5. https://doi.org/10.1109/IBSSC51096.2020.9332218

WHO. (2019). Snakebite envenoming: A strategy for prevention and control. *World Health Organization, Geneva*, *3*(1). https://doi.org/10.1038/nrdp.2017.64

Young, L. E. (2019). The Psychology of State Repression: Fear and Dissent Decisions in Zimbabwe. *American Political Science Review*, *113*(1), 140–155. https://doi.org/10.1017/S000305541800076X

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Appendices

Appendix A: Mobile Net Model

```
[ ] # Instatiating Mobilenet model
    mobile_model = MobileNet(input_shape= IMAGE_SIZE+[3],weights = 'imagenet',include_top= False)
```

```
[ ] # Do not train mobile net layers
    for layer in mobile_model.layers:
      layer.Trainable = False
```

```
[ ]
```

```
[ ] x = Flatten()(mobile_model.output) # Flatten the last layers
```

```
prediction = Dense(len(train_folder), activation="softmax")(x) # Last layer having softmax activation function
modelmobile = Model(inputs=mobile_model.input, outputs=prediction) #
```

```
[ ] # Compiling model
    from tensorflow.keras.optimizers import Adam
    optimizer = Adam(learning_rate=1e-3) # Initializing optimizer with a learning rate
    modelmobile.compile(loss="categorical_crossentropy", metrics=["accuracy"], optimizer=optimizer) # Compiling model
```

```
# Fitting  and training model
#callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3)

mobilenet_model = modelmobile.fit_generator(training_set,
                      validation_data=testing_set,
                      epochs=20,
                      steps_per_epoch=len(training_set),
                      validation_steps=len(testing_set))
                      #callbacks=[callback]
```

Appendix B: Model Summary

Model: "model_3"

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 224, 224, 3)]     0

 conv1 (Conv2D)              (None, 112, 112, 32)      864

 conv1_bn (BatchNormalizatio  (None, 112, 112, 32)     128
 n)

 conv1_relu (ReLU)           (None, 112, 112, 32)      0

 conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)     288

 conv_dw_1_bn (BatchNormaliz  (None, 112, 112, 32)     128
 ation)

 conv_dw_1_relu (ReLU)       (None, 112, 112, 32)      0

 conv_pw_1 (Conv2D)          (None, 112, 112, 64)      2048

 conv_pw_1_bn (BatchNormaliz  (None, 112, 112, 64)     256
 ation)

 conv_pw_1_relu (ReLU)       (None, 112, 112, 64)      0

 conv_pad_2 (ZeroPadding2D)  (None, 113, 113, 64)      0

 conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)       576

 conv_dw_2_bn (BatchNormaliz  (None, 56, 56, 64)       256
 ation)

 conv_dw_2_relu (ReLU)       (None, 56, 56, 64)        0
```

```
conv_pw_2 (Conv2D)          (None, 56, 56, 128)       8192

conv_pw_2_bn (BatchNormaliz  (None, 56, 56, 128)      512
ation)

conv_pw_2_relu (ReLU)       (None, 56, 56, 128)       0

conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)      1152

conv_dw_3_bn (BatchNormaliz  (None, 56, 56, 128)      512
ation)

conv_dw_3_relu (ReLU)       (None, 56, 56, 128)       0

conv_pw_3 (Conv2D)          (None, 56, 56, 128)       16384

conv_pw_3_bn (BatchNormaliz  (None, 56, 56, 128)      512
ation)

conv_pw_3_relu (ReLU)       (None, 56, 56, 128)       0

conv_pad_4 (ZeroPadding2D)  (None, 57, 57, 128)       0

conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)      1152

conv_dw_4_bn (BatchNormaliz  (None, 28, 28, 128)      512
ation)

conv_dw_4_relu (ReLU)       (None, 28, 28, 128)       0

conv_pw_4 (Conv2D)          (None, 28, 28, 256)       32768

conv_pw_4_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)
```

conv_pw_4_relu (ReLU)	(None, 28, 28, 256)	0
conv_dw_5 (DepthwiseConv2D)	(None, 28, 28, 256)	2304
conv_dw_5_bn (BatchNormaliz ation)	(None, 28, 28, 256)	1024
conv_dw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65536
conv_pw_5_bn (BatchNormaliz ation)	(None, 28, 28, 256)	1024
conv_pw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pad_6 (ZeroPadding2D)	(None, 29, 29, 256)	0
conv_dw_6 (DepthwiseConv2D)	(None, 14, 14, 256)	2304
conv_dw_6_bn (BatchNormaliz ation)	(None, 14, 14, 256)	1024
conv_dw_6_relu (ReLU)	(None, 14, 14, 256)	0
conv_pw_6 (Conv2D)	(None, 14, 14, 512)	131072
conv_pw_6_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_pw_6_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_7 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_7_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048

conv_dw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_7 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_7_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_pw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_8 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_8_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_dw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_8 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_8_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_pw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_9 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_9_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_dw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_9 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_9_bn (BatchNormalization)	(None, 14, 14, 512)	2048

conv_pw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_10 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_10_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_dw_10_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_10 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_10_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_pw_10_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_11 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_11_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_dw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_11 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_11_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_pw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pad_12 (ZeroPadding2D)	(None, 15, 15, 512)	0
conv_dw_12 (DepthwiseConv2D)	(None, 7, 7, 512)	4608

```
conv_dw_12_bn (BatchNormali  (None, 7, 7, 512)       2048
zation)

conv_dw_12_relu (ReLU)      (None, 7, 7, 512)        0

conv_pw_12 (Conv2D)         (None, 7, 7, 1024)       524288

conv_pw_12_bn (BatchNormali  (None, 7, 7, 1024)      4096
zation)

conv_pw_12_relu (ReLU)      (None, 7, 7, 1024)       0

conv_dw_13 (DepthwiseConv2D  (None, 7, 7, 1024)      9216
)

conv_dw_13_bn (BatchNormali  (None, 7, 7, 1024)      4096
zation)

conv_dw_13_relu (ReLU)      (None, 7, 7, 1024)       0

conv_pw_13 (Conv2D)         (None, 7, 7, 1024)       1048576

conv_pw_13_bn (BatchNormali  (None, 7, 7, 1024)      4096
zation)

conv_pw_13_relu (ReLU)      (None, 7, 7, 1024)       0

flatten_2 (Flatten)         (None, 50176)            0

dense_3 (Dense)             (None, 18)               903186

=================================================================
Total params: 4,132,050
Trainable params: 4,110,162
Non-trainable params: 21,888
```

Appendix C: Training process

Epoch 1/20

99/99 [==============================] - 1801s 18s/step - loss: 3.0575 - accuracy: 0.7431 - val_loss: 12.6753 - val_accuracy: 0.5345

Epoch 2/20

99/99 [==============================] - 1070s 11s/step - loss: 0.6307 - accuracy: 0.8907 - val_loss: 8.0910 - val_accuracy: 0.6730

Epoch 3/20

99/99 [==============================] - 1092s 11s/step - loss: 0.4221 - accuracy: 0.9138 - val_loss: 3.8491 - val_accuracy: 0.7801

Epoch 4/20

99/99 [==============================] - 1069s 11s/step - loss: 0.5733 - accuracy: 0.8993 - val_loss: 7.8324 - val_accuracy: 0.5748

Epoch 5/20

99/99 [==============================] - 1076s 11s/step - loss: 0.4910 - accuracy: 0.9123 - val_loss: 4.9808 - val_accuracy: 0.7141

Epoch 6/20

99/99 [==============================] - 1060s 11s/step - loss: 0.3240 - accuracy: 0.9366 - val_loss: 1.0138 - val_accuracy: 0.8739

Epoch 7/20

99/99 [==============================] - 1095s 11s/step - loss: 0.2507 - accuracy: 0.9452 - val_loss: 0.2877 - val_accuracy: 0.9633

Epoch 8/20

99/99 [==============================] - 1073s 11s/step - loss: 0.2361 - accuracy: 0.9525 - val_loss: 0.2458 - val_accuracy: 0.9406

Epoch 9/20

99/99 [==============================] - 1096s 11s/step - loss: 0.2320 - accuracy: 0.9528 - val_loss: 2.4074 - val_accuracy: 0.8270

Epoch 10/20

99/99 [==============================] - 1073s 11s/step - loss: 0.2587 - accuracy: 0.9424 - val_loss: 1.8552 - val_accuracy: 0.7515

Epoch 11/20

99/99 [==============================] - 1092s 11s/step - loss: 0.1834 - accuracy: 0.9588 - val_loss: 0.1643 - val_accuracy: 0.9663

Epoch 12/20

99/99 [==============================] - 1114s 11s/step - loss: 0.1325 - accuracy: 0.9671
- val_loss: 0.2101 - val_accuracy: 0.9611
Epoch 13/20
99/99 [==============================] - 1079s 11s/step - loss: 0.1459 - accuracy: 0.9604
- val_loss: 0.0738 - val_accuracy: 0.9773
Epoch 14/20
99/99 [==============================] - 1082s 11s/step - loss: 0.1978 - accuracy: 0.9569
- val_loss: 0.3118 - val_accuracy: 0.9604
Epoch 15/20
99/99 [==============================] - 1079s 11s/step - loss: 0.3211 - accuracy: 0.9458
- val_loss: 0.2801 - val_accuracy: 0.9348
Epoch 16/20
99/99 [==============================] - 1079s 11s/step - loss: 0.2551 - accuracy: 0.9544
- val_loss: 0.9635 - val_accuracy: 0.9142
Epoch 17/20
99/99 [==============================] - 1097s 11s/step - loss: 0.5030 - accuracy: 0.9224
- val_loss: 10.3422 - val_accuracy: 0.6224
Epoch 18/20
99/99 [==============================] - 1081s 11s/step - loss: 0.1941 - accuracy: 0.9569
- val_loss: 0.2201 - val_accuracy: 0.9633
Epoch 19/20
99/99 [==============================] - 1088s 11s/step - loss: 0.0921 - accuracy: 0.9759
- val_loss: 0.1466 - val_accuracy: 0.9839
Epoch 20/20
99/99 [==============================] - 1081s 11s/step - loss: 0.1383 - accuracy: 0.9652
- val_loss: 0.2605 - val_accuracy: 0.9553

Appendix D: Confusion matrix and Classification Report

```
Confusion Matrix
[[ 43   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0]
 [  0  36   0   0   0   0   0   0   0   0   2   0   0   0   1   0   0   0]
 [  0   0 105   0   0   0   0   0   0   0   5   0   0   0   0   1   0   0]
 [  0   0   0  42   0   0   0   0   0   0   1   1   1   0   0   1   0   0]
 [  0   0   0   0  64   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0  39   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0  67   0   0   0   2   0   1   0   0   0   0   0]
 [  2   0   0   0   0   0   0  72   0   0   0   0   1   0   0   1   0   0]
 [  0   0   0   0   0   0   0   0 128   0   1   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  17   0   0   0   0   0   0   0   1]
 [  0   0   0   0   0   0   0   0   0   0 115   0   0   0   0   0   0   0]
 [  0   3   0   0   2   0   0   0   0   0   3 122   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  61   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0  48   9   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   1   0   0   0  99   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   3   0   0   0   1 115   0   0]
 [  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  47   0]
 [  0   0   0   0   2   0   0   0   0   0   0   0  11   0   1   1   0  83]]
```

Classification Report

Classification Report

	precision	recall	f1-score	support
African Rock Python	0.93	0.98	0.96	44
Black Necked Spitting Cobra	0.92	0.92	0.92	39
Black mamba	1.00	0.95	0.97	111
Boomslang	1.00	0.91	0.95	46
Brown House Snake	0.94	1.00	0.97	64
East African Garter Snake	1.00	1.00	1.00	39
Egyptian Cobra	1.00	0.96	0.98	70
Gaboon Viper	1.00	0.95	0.97	76
Green Mamba	1.00	0.99	1.00	129
Jackson Tree Snake	1.00	0.94	0.97	18
Jameson Mamba	0.86	1.00	0.93	115
Large Brown Spitting Cobra	0.99	0.94	0.96	130
Mt Kenya Bush Viper	0.81	1.00	0.90	61
Puff Adder	1.00	0.84	0.91	57
Red Spitting Cobra	0.88	0.99	0.93	100
Rufous Beaked Snake	0.97	0.97	0.97	119
Sand Boa	1.00	0.98	0.99	48
Western Forest Cobra	0.99	0.85	0.91	98
accuracy			0.96	1364
macro avg	0.96	0.95	0.96	1364
weighted avg	0.96	0.96	0.96	1364

Appendix E: Ethical Approval

Strathmore
UNIVERSITY

7th April 2022

Ms Mabinda Mariam,
mariam.mabinda@strathmore.edu

Dear Ms Mabinda,

RE: A Snake Classification Model for Snakebite Envenoming Management

This is to inform you that SU-IERC has reviewed and **approved** your above **SU Masters'** research proposal. Your application reference number is **SU-IERC1311/22.** The approval period is **7th April 2022 to 6th April 2023.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Dr Ben Ngoye,**
Secretary; SU-IERC

Cc: Prof Fred Were,
Chairperson; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL
ETHICS REVIEW COMMITTEE
(SU-IERC)
07 Apr 2022
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu

Appendix F: Similarity Test Report

Curiginal

Document Information

Analyzed document	evenoming Mangaement Final.docx (D133600588)
Submitted	2022-04-14T05:05:00.0000000
Submitted by	
Submitter email	Mariam.Mabinda@strathmore.edu
Similarity	12%
Analysis address	library.strath@analysis.urkund.com

Appendix F: Link with demonstration of the App

Loon app was used to record the demonstration of how the app works. The link is attached below.

https://www.loom.com/share/c5ad913fa42f4c0c95d960b5be283fe3

Appendix G: Sample Snake Images Collected