FDRE TECHNICAL AND VOCATIONAL TRAINING INSTITUTE

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL ELECTRONICS AND INFORMATION COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY (ICT)

CLASSIFICATION OF GOAT AND SHEEP SKIN DISEASES USING DEEP LEARNING APPROACHES

A Thesis Submitted to the Department of ICT in Partial Fulfillment of the Requirements for the Masters of Science in Information Communication Technology

Prepared By:

Hiluf Nuguse Gidey

Advisor:

Dr. Temtim Assefa. (PhD)

April 2026

Addis Ababa, Ethiopia

Certification

This is to certify that the thesis entitled "Classification of Goat and Sheep skin Diseases using Deep Learning Approaches", submitted by **Hiluf Nuguse Gidey** in partial fulfillment of the requirements for the degree of Master of Science in Information Communication Technology at FDRE Technical and Vocational Training Institute, is an original work and has been carried out under my supervision.

This thesis meets the required academic standards and the institute's regulations for originality and quality.

1. Name of Advisor: Dr. Temtim Assefa (PhD) Signature: _____________ Date: ______________
2. Name of Internal Examiner: Prof. Ravindira Babu (PhD) Signature: _______Date: ___________
3. Name of External Examiner: Dr. Haylay Beyene (PhD) Signature: ________Date: ___________

Declaration

I, Hiluf, the undersigned, declare that this thesis entitled "**classification of Goat and Sheep skin diseases using Deep learning approaches"** is my original work. I have undertaken the research work freely with the guidance and supervision of my research advisor.

Declared by: Hiluf Nuguse Signature: ________ Date 16 April 2026

This thesis has been submitted for examination with my approval as university thesis advisor.

Advisor:

Dr. Temtim Assefa (PhD) Signature_____________ Date_________

Acknowledgments

I would like to thank my advisor, Dr. Temtim Assefa (PhD) who assisted me. He has been an invaluable source of academic support throughout this research. I appreciate the guidance, encouragement, and feedback that he provided me with during my research time. I valued his thoughts, feedback and criticism. His feedback helped improve my work significantly. He consistently encouraged me to do my best that is why I was motivated. Thanks to Dr. Temtim Assefa.

I equally owe a debt of gratitude to Dr. Haftu Guben (Veterinary Doctor) and Hadis Kahsay (Coordinator of ESARDHO) at Endamoni wereda due to their abundant assistance and cooperation in the field of veterinary. He was very helpful in my fieldwork and clinic. Dr. Haftu Guben is very aware of skin diseases of goats and sheep. He took pleasure in telling about what he knew about working with animals. He also helped me to collect pictures of these diseases and label them appropriately. This was a help to my research. My sincere gratitude to farmers and animal health workers of the Endamoni wereda also would mentioned. They allowed us to capture their animal's image.

Abstract

Animal farm has played one of critical roles in the socio economic wellbeing of most developing countries, Ethiopia being one of them as sheep and goats play major role in providing food security, generate incomes and export income. Nevertheless, there are a number of skin diseases, which have a severe impact on the quality, and yields of animal products. In the rural set ups, they are not handled well because of the inaccessibility of veterinary clinics and shortage of expertise in the field. In this paper, an enhanced method of deep learning approach provided to identify and classify skin diseases in sheep and goats with deep learning. The system automation CNN architectures mean that features are automatically trained by skin images by which skin diseases can be classified and classified to either bacteria and virus, parasite or healthy. The evaluated and tested model underwent testing and evaluation on the standard measures, which include accuracy, precision, recall and F1-score. The data employed in this thesis is 2341 images of data. The training dataset classified as 80% (1873 images), 10% (234 images) and 10% (234 images) in training, validation and test performance respectively. The classification work will consist of four categories namely bacterial, viral, parasitic, and healthy skin conditions. This study compares the performance of VGG19, Custom CNN, DenseNet201, and InceptionV3 for image classification using accuracy, loss, and training time. Results show that DenseNet201 achieved the best overall performance with the highest test accuracy of 0.9744, the lowest test loss of 0.1291, and the fastest training time of 18.89 minutes. The Custom CNN also performed well with a test accuracy of 0.9487, while VGG19 required the longest training time (28.89 minutes) with slightly lower accuracy. InceptionV3 showed comparatively lower performance with a test accuracy of 0.9017. Overall, DenseNet201 proved to be the most effective model, offering the best balance between accuracy and efficiency for the given dataset.

Keywords: Deep Learning, Convolutional Neural Network, Image Classification, Goat and Sheep Skin Diseases, Computer Vision,

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Adam	Adaptive Moment
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	deep learning
ESARDHO	Economic Sector Animal Resource and Healthcare Office
FC	Fully connected
FP	False Positive
FN	False Negative
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbor
ML	Machine Learning
NBC	Naïve Bayes Classifiers
RGB	Red, Green, Blue
ReLU	Rectified Linear Unit
TN	True Negative
TP	True Positive
VGG-16	Network Visual Geometry Group-16
VGG-19	Network Visual Geometry Group-19

Chapter One

1.1 Introduction

Ethiopia has a large number of animals. It has the largest livestock population in Africa and the fifth largest in the world[1]. The country has around 61.5 million cattle. It also has 33 million sheep and 38.9 million goats. There are 1.76 million camels and over 59.4 million chickens in Ethiopia[1]. This huge animal endowment underscores the great potential of Ethiopia to export large amounts of live animals and animal products including skin. However, this potential has been curtailed by the widespread occurrence animal skin diseases in the country and which has limited the overall economic advantages that may be reaped from livestock production [1].

Livestock farming is very important in the Eastern African region, as it is the major animal resource in the continent, which accounts approximately 56% of the total livestock population[1]. Goats and sheep are significant to rural people, as they can withstand severe environmental factors more than cattle and are mostly reared by small-scale farmers throughout Ethiopia to earn their family income, milk, meat, and skin [2]. The skins of small ruminants, especially sheep and goat skin, are universally recognized for their high quality. In Ethiopia, skins represent the second-largest source of foreign currency earnings, with an annual production of approximately 16.2 million export-grade skins. Despite this potential, around 35% of sheep skin and 56% of goat skin downgraded or rejected by Ethiopian tanneries[3]. Parasitic skin diseases, including sheep and goat pox, mange mites, ticks, lice, and fleas, are major contributors to this deterioration [4]. These conditions greatly lower the quality of skin and provide a major threat to the livestock industry and the skin export industry. The sarcoptic, psoroptic, and demodectic mites that are prevalent in the tropical areas where conducive ecological factors and poor animal husbandry practices dominate require special attention due to their significant impact on the development of livestock[5].

Regardless of Ethiopia's large livestock population, the economic contribution has not reached its full potential because of the widespread existence of infectious diseases, particularly sheep/goat pox. These diseases are among the key restrictions preventing the productivity of the goat and sheep. Moreover, they represent significant economic threats worldwide, especially in developing countries [6].

In Ethiopia, animal farming is the second major source of foreign currency through export of live animals, meat, skins and hides. Skin from sheep and goats are important economic products contributing to the largest share and agricultural export supplies followed by live animals. However, this rank has declined to levels because of rejection and down grading on hides and skin defects due to infestation by external parasites. But also, due to pre-and post-slaughter skin management problems[5].

The main constraints of goat and sheep production in Ethiopia are parasites, virus, bacteria, feed shortage, labor scarcity, predators, land shortage, lack of improved goat and sheep breeds, and poor market access to sell goats and sheep products. This indicates that disease is graded as the primary constraint, while feed scarcity and predators were ranked second and third in all three agro environments[7].

Goat and sheep farming is a livelihood option for small-scale farmers, particularly in rural and semi-rural areas, because it is simple for farmers to adopt and manage small number of goats or sheep alongside their core agricultural activity. With rising demand for sheep and goat meat and market prices, sheep and goat farming has become a feasible and profitable option for farmers with limited land and resource [8].

The number of goat and sheep farming undertakings that started each year is lower than the number that shut down. This happens because goat and sheep farming ventures have heavy losses. There are some reasons why many goat and sheep farming ventures fail. These reasons include the fact that the goats and sheep do not get the vaccinations they need. In addition, the weather can change quickly and this can hurt the goats and sheep. Another reason is that the goats and sheep do not get the care they need. The lack of security is also a problem, for goat and sheep farming ventures. Many goat and sheep farming ventures fail because of these reasons.[8].

Goat and sheep production is practiced across vast geographical locations, mainly in low and middle altitude areas, and supporting rural smallholder farmers’ livelihood in harsh environments, specifically in the study area. Despite its potential, it does not maximize the economic benefits of small farmers. Although there are rapidly growing small ruminant populations, especially goats, the sector is failing to specialize and intensify. The existing research emphasizes on goat and sheep breeding characteristics and management practices in different areas of Ethiopia. There is limited attention that has been given to socio-economic of goat and sheep production in general and there is no research specifically focused on the study area [7].

Therefore, this study is about goat and sheep production. It wants to fix the problems that other studies on goat and sheep production had. The study also wants to learn more about how people raise goats and sheep in areas of Ethiopia. The study focuses on identifying problem with the skin of goats and sheep when they get sick on their skin. This negatively affects how well the goats and sheep are doing and if they are healthy. The study aims to address diseases that affect the skin of goats and sheep. In this regard, the research aims at improving the early management of skin diseases among the animals through the application of deep learning technology, thus improving the health benefits and productivity of the animals.

1.2 Statement of the Problem

Ethiopia has the largest livestock population in Africa, with millions of sheep and goats that produce meat, milk, skins and serves as a source of income. The productivity and economic value of sheep and goats are still restricted by several factors, such as skin diseases, despite this enormous resource potential. According to studies, skin defects caused by diseases, especially parasitic infections like mange, viral infections like orf, and bacterial infections like foot rot are a major problem for the production of sheep and goats as well as the leather industry[9].

Despite the huge number of livestock in Ethiopia, there has been a decline in national and per capital production of livestock and livestock products, export earnings from livestock and per capital consumption of food from livestock origin in comparison to other African countries due to disease and other constraints[10].

Muhaba et al. **(2022)** made phone application that looks at pictures of skin and classify five common human skin diseases using mobileNetV2 architecture. It is good at figuring out what is wrong with the skin. The approach achieved 97.5% classification accuracy[11].

On the same note, **Hwang et al (2022)** used deep learning to classify three dog skin diseases on an image basis (RGB and multispectral). They compared various CNN architectures (ResNet, DenseNet and MobileNet) and observed that consensus models with multispectral inputs performed better than single-source inputs with the highest accuracy of 89%. Although they have advanced this area, their research is limited to the species of dogs and does not address disease classification in livestock animals such as goats, which have other environmental and health related problems[12].

Traditional analytic approaches for sheep and goat skin diseases in the case study presented by **Moriello and Cooley (2001)** involves histopathology analysis and lesion observation to identify conditions like contagious pustular dermatitis[4].

Niloy et al. (2024) found machine-learning predictors of livestock diseases with tabular data that ordered by symptoms. Their models worked with large datasets but did not include visual data like skin lesions or species-specific data that applied to sheep and goats. Thus, their results are not easily applicable in automated dermatological recognition farm tasks [13].
Difficulties of Manual Classification of Goat and sheep skin disease.

1. **Time-Consuming:** Every illness requires a thorough examination, and which is time-consuming, particularly when it comes to working with vast amounts.
2. **Labor-Intensive:** It requires significant human labor force that requires veterinary training and in farming locations and which is difficult to manage.
3. **Subjective and Prone to Errors:** When it comes to classifying a disease through visual inspection, the various specialists may categorize it differently because of the subjectivity of making a decision and thereby arriving at different conclusions.

Thus, it is fundamental to create robust and improved model that applies deep learning method in identifying and categorizing skin conditions in sheep and goats. Finally, the livestock resources would be more productive and of better quality, which would also contribute to the application of early diagnosis, assist veterinary decision-making, and minimized financial losses in Ethiopia.

1.3 Motivation of the Study

Farmers in Ethiopia and other developing countries rely on millions of smallholder farmers to make their living since it supplies them with raw materials such as skin and leather, food and income. Nonetheless, many skin diseases such as parasitic, bacterial, as well as viral affect the productivity and the market values of goats and sheep. Such skin diseases often misdiagnosed due to lack of proper veterinary infrastructure and use of manual, experience-based methods of examination. Traditional diagnosis methods such as skin scraping are tedious, invasive and cannot used on a large scale or in field conditions.

The advances in deep learning and image processing have revolutionized diagnostic procedures in human diseases and some animal species where the two have demonstrated immense accuracy in dermatological issue diagnosis based on images. Nevertheless, unlike with these technologies, which have emerged as more and more promising, there have been limited studies regarding the application of these technologies towards the diagnosis of skin diseases in goats and sheep. The existing models have not tailored to the specifics of the skin, hair thickness, and disease presentation patterns of goats as they are often developed using cattle or human data.

1.4 Research Questions

For guiding the research, the following research questions are proposed:

1. How can images of sheep and goat skin diseases be effectively labeled and preprocessed to ensure high-quality input for deep learning models?
2. Which deep learning (CNN) architectures and training configurations achieve the best performance for classifying sheep and goat skin diseases, and how do they compare in terms of accuracy and computational cost?
3. Which model achieves the highest classification performance on the dataset, and how does it compare to other models in terms of accuracy, precision, recall, F1-score, and confusion matrix results?
4. How can the best performing model be efficiently integrated into a web-based application to provide reliable and real-time skin disease classification for end users?

1.5 Objectives of the Study

1.5.1 General Objective

The general objective of this study is to develop and evaluate classification of goat and sheep skin diseases using Deep learning models from skin diseases image dataset.

1.5.2 Specific Objectives

The specific objectives of this study are:

1. To label and preprocess a dataset of sheep and goat skin disease images to ensure high-quality input for model training.
2. To design and develop deep learning (CNN) models for the classification of sheep and goat skin diseases.
3. To compare and evaluate the performance of different CNN architectures in terms of classification accuracy, precision, recall, F1-score, and computational cost, and to identify the best-performing model.
4. To integrate the best-performing model into a web-based application for efficient and real-time skin disease classification.

1.6 Scope of the Study

This research focuses on the classification of common visible goat and sheep skin diseases using image-based data and deep learning techniques. The study targets rural farming communities where access to veterinary services is limited. It confined to skin diseases that are visually classify through external symptoms and does not include internal or laboratory-diagnosed diseases. The scope of the study is limited to the classification of four categories**:** bacterial, viral, parasitic skin diseases, and healthy skin conditions.

1.7 Limitations of the Study

Despite the proposed approach working well, the study has some limitations. The model only uses image data so it cannot classify diseases that are not visible or are inside the body. In addition, factors such as weather and disease progression has progressed not taken into account. The study only looked at images, which is a limitation. It did not consider other influencing factors.

1.8 Significance of the Study

This study has significant implications for the goat and sheep skin disease sector in Ethiopia

For Researchers: The research provides a valuable dataset of sheep and goat skin diseases that can be used in future research and training models. It provides access to the best preprocessing methods, activation functions and local feature descriptors used in the goat and sheep skin disease classification tasks. Overall, the study contributes to knowledge of the impacts of the modeling on the performance in the classification of goat and sheep skin diseases and, therefore, provides a platform upon which further research in the field conducted.

For Veterinarians: The proposed system helps identify diseases of goat and sheep skin, which have similar but not identical visual manifestations and thus the diagnostic accuracy and effectiveness. The system further supports veterinarians in classifying and treating diseased animals. This suggests that to improve animal health and welfare by enabling timely and accurate intervention. In addition, the system helps in minimizing the requirement of making repetitive trips to the veterinarian clinic since the system can help classify diseases and conduct initial diagnosis. This is possible because of the disease classification that is done automatically.

1.9 Organization of the Thesis

This study divided into five chapters: chapter one explains the background of this study, the problem, purpose, scope and limitation, significance of the study, and methodology. Chapter two describes concepts and methods relevant to the proposed approach. Chapter three describes the method of the study, proposed methods, and algorithms. The preprocessing, segmentation, feature extraction, and classification are described in detail. Chapter four explains the results of experiments, and chapter five presents conclusion and recommendations for researchers interested in this field.

Chapter Two

Literature Review and Related Work

2.1. Introduction

This chapter provides a review of past studies and the existing literature on sheep and goat skin disease classification using deep learning techniques. The purpose of this review is to provide the theoretical and empirical background to the current study by evaluating similar research works in the fields of animal diseases diagnosis, computer vision, and machine learning. Goats and sheep are an important part of the rural economy in developing countries such as Ethiopia as they produce large amounts of meat, milk, skin, and income. Skin infections such as mange and orf significantly affect productivity and the quality of Diseases such as foot rot and pox lead to economic losses and poor hide and skin quality. Traditional diagnosis primarily relies on visual examination by veterinarians and is generally slow, subjective in nature and limited in rural regions where veterinarians are scarce.

2.2. Theoretical Review of Animal Skin

The skin is the largest organ of the animal body. It covers the entire external body and serves as the first line of defense against pathogens. It provides a mechanical barrier against ultraviolet light, chemicals, and physical trauma. It also regulates body temperature and water loss [14].

2.2.1 Animal Skin Structure

Skin cells measure between 25 and 40 micrometers depending upon the position of the cells on the skin. Animal skin cells have an average size of 30 micrometers. The Epidermis is composed mostly keratinocytes. The basement membrane (also referred to as dermo-epidermal junction) is located below the epidermis. This layer attaches the epidermis to the dermis to the dermis. The hypodermis lies below the dermis, and is mainly composed of fat tissue [15].

Epidermis: The epidermis is the surface layer of the skin and the skin made up of the squamous type of keratinocytes in various levels of differentiation whose major cellular component, was the keratin protein synthesized. The epidermis stores the nutrients, waste using the basement membrane, and depends on the dermis to provide them with blood vessels. The main object of the skin known as epidermis; this formed to safeguard the skin against the irritation and entry of allergens. It is also a water retainer, and body temperature is maintained [16].

Dermis: It is an internal section of the skin and is much thicker than the epidermis (1-5mm). Dermis is the layer between the basement membrane and the subcutaneous tissue and the primary role of this layer is to support and sustain epidermis. The roles of the dermis include: that of protection, cushioning the underlying structures against mechanical trauma, preservation of the epidermis, and wound healing [17].

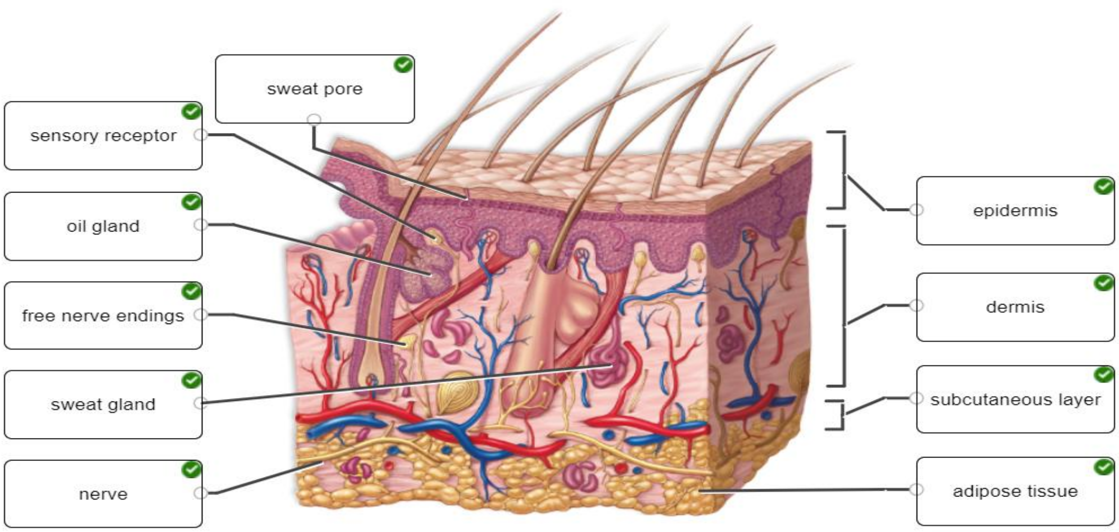


Figure 1 structure of animal skin

2.2.2 Goat and sheep skin characteristics

Sheep and goats skin is also dissimilar in terms of their uses in the leather industry in terms of their biological composition, fiber content and other physical properties. The goats skin is harder and more durable and has smaller grain whereas sheep skin is less tough and is more flexible and has more fat in it.

Breed or Type: The breed characteristics that are undesirable in sheep skins are higher than the goat skins. The skins of high quality are usually fat or obese, thick and smooth-grained in texture, and possess proportions between their weight and surface area that are relatively small. This means that these skins produce low grade leather and are usually utilized in the manufacture of the lowest grade of leather products [18].

Sex or Age: The skin of male goats and sheep is generally heavier and has coarser grain, but the skin of the female is more likely to have a better tensile strength. Young animals have skins that are generally fine, close and possess tight grain patterns. The older an animal is, the more the grain surface becomes tough and coarse hence leading to poorer skin quality [18].
Nutrition: Low nutrition causes the animals to be smaller, and the skin becomes thin with low quality material making the leather not to be elastic and the skin being dull or dead to touch. These skins also depict the less fat deposition in the corium layer and generally have a less coarse yet weaker grain structure.

Climate: Warmer climates are usually known to have shorter hair on animals and the skin because of which the leather produced is usually made of better substances with less thick and less coarse grain patterns. Animals kept in colder or higher climates grow longer hair or wool, and the leather, which is produced, is generally of a lower character, and their grain coarser. These climatic influences are more so in goat and sheep skin.[18].

2.2.3 Goat and sheep skin diseases

Goat and sheep yield-limiting causes such as diseases, poor breeding practices, and poor management practices are almost present in all countries. Skin disease is a malady of exterior of an animal. There are numerous skin diseases of sheep and goats. Doctors, field veterinarians, animal health workers, animal scientists and animal farmers who rear goats and sheep should be aware of these diseases. The bacterial, fungal, parasitic and viral infections are some of the common skin diseases that afflict sheep and goats. Health hazards can also be caused by such things as problems or being unable to get the right food [19]. Common skin diseases that affect sheep and goats grouped according to their causes are as follows.

1. **Virus**

Goat pox/sheep pox: It is a febrile and transmittable disease of goats and sheep with nodules or ulcerations. Sheep or goat pox may attack any age but the high risk is in lambs and kids. Sheep or goats pox is caused by the goat and sheep pox viruses of the genus Capripoxvirus and family Poxviridae. The two viruses are similar to each other and they differ in their antigenome [20].

A. Goat pox B. Sheep pox

Figure 2 Goat pox and sheep pox

Orf: Contagious ecthyma (contagious pustular dermatitis or orf) is a contagious disease of goats and sheep which presents as pustular and scabby lesions on the muzzle, commissures of the lip and nostrils. The disease is caused by a contagious ecthyma virus belonging to the genus Parapoxvirus and family Poxviridae and has an impact on the skin. Transmittable ecthyma that is marked initially by the emergence of erythema, which subsequently transform into papules and pustules. At the time of rupture of the pustules, the pus is dense in a crust of grey hue and this is subsequently followed by discrete and thick scabs that are crumbly and yet stuck to the underlying tissues. The lesions typically start at the commissures of the mouth and then extend to lips, muzzles, nostrils and ears. Lesions can also be present on the coronet, interdigital cleft, skin of the udder and teats, vulva, preputial orifice, perineal area, thighs and axillae. The scabs of coalesce adjoining form continuous plaques. [20].

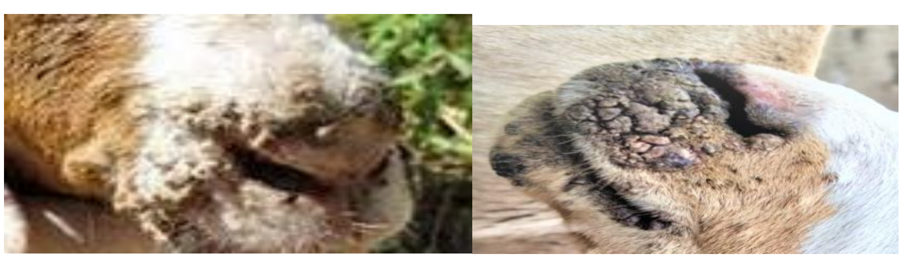


Figure 3 Orf viral

1. Bacterial

Foot rot: Foot rot is a contagious disease of the feet, which is established by irritation of the skin horn junction, under running of horn, ulceration and necrosis of sensitive laminae and severe lameness. Foot rot has no typical pathological appearances despite grossly having interdigital necrosis. It has a typical, black and foul smell, because of the bacterial decay of the horn. Infection spreading to the joints may lead to the pus accumulation in the joint cavity. Although the frequency of the occurrence of lameness with the resulting morbidity is high and with a great number of clinical cases of sheep in Ethiopia, the literature indicating the epidemiological significance of foot rot appears to be insufficient [20].

Figure 4 foot rot bacterial

2. Parasitic

Mange Mite: Mange Mite: These belong to the subclass Acari of the arachnid class but the majority of veterinarily significant ones are found in the order Trombidiformes or Sarcoptiformes. The Sarcoptiformes are oribatid mites that are intermediate hosts to various species of tapeworms such as Anoplocephala, Moniezia and Stilesia. Mite infections may cause acariasis and lead to severe dermatitis referred to as mange which can cause serious welfare issues and economic damages. The parasitic mites are tiny creatures, the majority of them being under 0.5 mm long, although some species that feed on blood can grow up to several millimeters in length when full [20].

Figure 5 Mange mite parasitic

2.3 Fundamentals of Digital Images

2.3.1 What is Image?

An image consists of two-dimensional array composed of numerous small square or rectangular units referred to as pixels. A voxel is a section of the brain tissue in three dimensions that is defined by every pixel. Each pixel is usually given the value of grayscale between black and white or in any particular color that needs numerical data [21].

2.3.2 Types of Digital Images

The main types of image include grayscale, binary, color and multispectral images.

1. **Color image:** Color images may be referred to as three and distinct bands of monochrome where each band corresponds to a single color channel. The grayscale intensity values of each of these bands are stored in the digital image. The color images are mostly in red, green and blue format. In the case where each channel represented an 8-bit image, the resulting image is a 24-bit per pixel image with 8 bits in the red, green and blue channels[22].

2. **Binary image:** Binary images are the simplest images that have two possible values typically black and white or 0 and 1. Since it takes only one bit to represent a pixel they are called one-bit image. These images were commonly employed in tasks where simple shape or structure should be used like optical character recognition. Binary images are binary images gained by a natural process on top of grayscale images via thresholding where every pixel whose value exceeds some threshold is considered black and the rest of the pixel is considered white[23].

Binary

3. **Gray-scale image:** Grayscale images or monochrome images are those images that have no color content and only contain gray. The number of bits that are allocated to each pixel can represent the amount of gray levels. Grayscale images have bits per pixel of the 8 bits approves the use of 256 gray levels. Although they are often used in areas with 12 or 16 bits per pixel like in clinical imaging[24].

Grayscale

4. **Multispectral images:** Multispectral images extract a portion of information at wavelengths that are invisible to our eyes. This consists of the infrared, ultra-violet and X-rays. Our human eyes fail to see these wavelengths. We are able to transform them into images. This is accomplished by giving red, green and blue colors on a screen wavelengths. This aids us to view things which cannot be perceived by the human eye. This information is used in multispectral images in wavelengths. They project them onto the visible colors[24].

Multispectral (simulated)

2.4 Vision System

Vision system is a system whereby visual information or images are captured, interpreted and then processed. A machine or software has the capability to view and make decisions based on what it views which resembles those of human vision work. The computer vision is capable of processing inputs fed to it once some preprocessing done to the image and derives useful information with the help of a feature-extracting algorithm. Thus, it is able to recognize the real world object it is capable of forming visual images and process the perceived information, known as a vision system. The vision system has numerous applications. Vision system divided into human and computer vision.

2.4.1 Human Vision

Human vision is the biological mechanism of human perception of the visual environment interpretation and perception of the information with the help of the eye and the brain. Human vision is the innate capacity to interpret, decompose and react to the objects in the real world. Taking into consideration the sense of organ, human beings are able to process objects with the help of their brain. Thus, humans have the sense of seeing the object with naked eye and interpreting the information of the object. This organ sense along with the brain form the human vision system. Humans are interpreting the facts with the light that comes to the eye [25].

2.4.2 Computer Vision

In the human visual capacity, which was also referred to as computer vision as explained by the camera as the primary element of the computer vision system, it was mimicked electronically. Computer vision systems used in most fields including class defect classification, face recognition, on-road safety, medical diagnosis, etc. Computer vision starts with raw data collection followed by application of machine learning algorithms, deep learning and image processing methods to recognize patterns [25]. As they described, the main development of computer vision is mimicking the ability of human vision and achieving complex visual abilities. These are inspection, sorting, and gauging.

2.5 Digital Image Processing

In today's digital era, images are common in everyday life. An image is a visual symbol of an object, person or scene while a digital image is a discrete form of such representation that contains both layout and color information[26]. A digital image can labeled as a two-dimensional representing the projection of a three-dimensional scene onto a two-dimensional flat denote the spatial location of a pixel and the function value corresponds to its intensity. When the spatial coordinates and intensity values are discrete, the image classified as a digital image. Mathematically, a digital image is represented as a matrix composed of a finite number of elements, commonly referred to as pixels (picture elements or pels), which together form the two-dimensional image[26].

2.5.1 Image Acquisition

Image acquisition is the process of obtaining a digital representation of a scene, where the resulting elements are called pixels [27]. The primary objective of image acquisition is to convert an optical image serving real world data into a numerical array that can be processed and analyzed on a computer. Before any image, processing the image must first capture by a camera and converted into a convenient digital form.

2.5.2 Image Processing Operation

Image processing includes a variety of techniques and algorithms marked at enhancing, analyzing or transforming digital images. Key image processing operations include:

Binarization: This process converts a color or grayscale image into a binary image with only two intensity levels Binarization simplifies the image and speed up processing it useful for many image analysis tasks[28].

- **Smoothing:** A technique used to soften or blur the details in an image often applied to reduce noise or minor variations.
- **Sharping:** Image sharping techniques enhance edges and fine details in an image clarity and visibility for human view.
- **Noise Removal and De-blurring:** Prior to further processing, noise reduction filters applied to minimize unwanted variations in the image. De-blurring methods can employed depending on the type and extent of blur or noise present in the image.
- **Edge Extraction:** To find various objects before analyzing image content, edge extraction performed.
- **Segmentation**: The process of dividing an image into various parts called segmentation.

2.5.3 Image Enhancement

Image enhancement is a core discipline of image processing that seeks to add value of images towards a particular application [29]. It aimed at enhancing the perception and interpretability of information about images to human observers and giving appropriate inputs to further automated image processing operations. Image enhanced mainly aimed at modifying the image features in such a way that the image that emerges is more typically adapted to a specific task and viewer. It entails changing one or more image characteristics, whereby the choice of attributes and improvement policies is task specific [30].

2.5.4 Image Resizing

Image resizing is a process consisting of magnification and reduction and is a popular process used widely in image processing to alter image resolution. It was regularly used in areas like digital photography, multimedia applications and electronic publishing to scale the pixel dimensions to output device resolution needs in printers and display monitors. There is also the use of image resizing to create preview images as well as preparing digital images to be distributed online [31].

2.5.5 Image Segmentation

Image segmentation refers to the process of splitting an image into several r objects that are meaningful. It has a critical part in a broad scope of applications. Autonomous driving systems, video surveillance and augmented reality used in medical image analysis. The literature has suggested a vast number of image segmentation algorithms starting with the simple techniques like histogram-based methods, region growing, k-means clustering and watershed algorithms up to more sophisticated ones [28]. Deep learning models have produced a new set of image segmentation models with high performance improvements that usually have the highest levels of accuracy rates on popular benchmarks that has produced a paradigm shift in the area [32].

2.5.6 Feature Extraction

The feature extraction is one of the most important fields of artificial intelligence. It involves the categorization of the most relevant features of an image. Image classification can analyzed by determining the aspects of the image features and categorizing the numerical features. The process of extracting features is an important part of the image classification. It facilitates the best depiction of the content of an image. Another method of improving performance and efficiency of machine learning algorithms is feature extraction that simplifies data. Image feature extraction is a very important stage in computer vision and image processing. It entails the location and description of significant features of an image to be further analyzed like classification [33].

2.6 Machine Learning and Deep Learning

2.6.1 Machine Learning

Machine learning is also a branch of AI that allows systems to learn based on data and identify the patterns behind them, as well as make decisions with a minimum amount of human intervention. One of the most important attributes of machine learning models is that they are able to adjust and become better at their tasks depending on the previous experience. These models are able to make sound predictions through generalization and learning new and unknown data by studying past records [34].

The present paper is devoted to the DL algorithms that are especially efficient in such tasks as image classification, natural language processing, and other complicated jobs since they can learn without the involvement of a human. Machine-learning algorithm provides us with the images and allows the system to create a model automatically, based on the data. This obviously provides a multitude of potential opportunities to useful applications, in which some problems with existing data cannot be formulated or programmed properly [35]. **According to Zhuhai** [36], outlines three significant methods applied in the field of Machine Learning, which depend on the machine learning problem and the input data. These are supervised learning, unsupervised learning and reinforcement learning.

Supervised Learning

Supervised learning is one of the most popular machine learning methods that is based on performance metrics and historical data sets to acquire experience. Supervised learning has its basis on the method of finding the most efficient way of mapping input data to desired output by learning past data sets [37]. Supervised Learning works with labelled information and also generate labelled information. The primary approaches of Supervised Learning are Classification, Regression, and the most popular algorithms of each Supervised Learning techniques Classification or Regression[38].

- **Unsupervised Learning:** The second technique of machine learning is unsupervised learning. When no prior data labels to train on were available, it used to make a prediction model.

Unsupervised learning is a method based on the identification of trends in the data when dealing with unlabeled data. Unsupervised learning algorithms divide new groups after analyzing the input, which is not labeled with relevant information. The algorithm has to find the hidden structure in the unlabeled data/input in unsupervised learning [39]. It does so by feeding the learning algorithms with bulk of numbers of input data, characteristics and features so that the algorithms would be able to make the necessary observation to determine the most favorable grouping or correlation between data. K-means are the most popular clustering algorithms and a similar clustering algorithm is Hierarchical clustering algorithms [40].

- **Reinforcement Learning:** Reinforcement learning involves the utilization of previous data with no prior prediction to enhance the manipulation of dynamic system in future[[41]. It is a learning method in which an agent acts on an environment and it is through this that the agent learns about errors or rewards. The agent introduced into an unfamiliar environment so that the agent could act and learn through signaled. These indicators may be rewards or punishment [42].

2.6.2 Deep Learning

Deep learning represents a concentrated field of machine learning that concentrates on algorithms that are motivated by the design and organization of the human brain also known as artificial neural networks. These were models that are meant to mimic some of the mental processes that take place in the brain in the manner they organize interdependent neurons that process and transmit information [43]. The main strength of the deep learning is its hierarchical structure that enables the model to acquire hierarchical examples of data. The deep learning methods confirm good performance when they are trained on large data and can be successful in their accuracy by performing the learning process repeatedly [44].

Machine learning and deep learning have a number of similarities because they both rely on the classification of data and the models being trained. Deep learning enables the machine to learn and extract patterns in examples. It is a subdivision of machine learning that is more sophisticated and thus demands a huge amount of data and a lot of computing power to effectively process information. Image processing, natural language processing and voice recognition are some of the fields that apply deep learning extensively [45]. The various methods involved in deep learning to train models based on the data. Convolutional neural network, Recurrent Neural Networks, Generative Adversarial Networks, Transformers, and Autoencoders are some of the popular techniques. RNNs are especially effective in dealing with sequential data, and Long Short-Term Memory networks improve RNNs by improving the ability to capture long-term dependencies with special gating structures[45].

Convolutional Neural Network: Convolutional Neural Network or ConvNet is popular with image recognition, object image classification. The CNNs are made up of a number of layers, which may include convolutional layers that enable the extraction of features in the input images, a pooling layer that thereby reduce the size of the information and finally, fully connected layers that complete the classification. The most famous CNN models are ResNet, AlexNet, VGG, and GoogLeNet (Inception). These models have shown the state of the art performance on benchmark datasets like ImageNet [46]**.**

Recurrent Neural Networks: Recurrent Neural Networks Recurrent Neural Networks are neural network-based models developed to handle sequential data by having an abstract state that retains information over previous time steps. Long Short-Term Memory is an improved version of RNN, which can capture long-term dependence and solve the vanishing gradient problem. RNNs and LSTMs are common in natural language processing operations including machine translation, sentiment analysis, and speech recognition among others [47].

Bidirectional LSTM: Bidirectional LSTM is an optimization of the LSTM network, which takes a sequence data in both forward and backward directions. Bi-LSTM has the ability to extract more contextual data connections in the data because it is able to analyze past and future time steps. This is better than what is possible with a basis-count TM, as it becomes more useful in language modeling or sentiment analysis, where it is always necessary to have access to the entire context of a sequence [48].

2.6.3 Convolutional Neural Network Architectures

Convolution is a mathematical process that is used to transform a function with the help of a given operator in order to create another representation that demonstrates the similarity between functions in a certain region. Convolution is a scanning method to classify certain patterns or features by moving an appropriate operator over large pixel blocks [49].

Convolutional neural networks are the deep learning algorithms that are widely applicable in identifying the skin-related diseases through the analysis of medical images. Convolutional neural network that is used to process image processing tasks, learn and know major features in images automatically. In skin disease recognition system, CNN models are trained on massive collections of skin disease images in such a way that they are able to learn the patterns and other attributes of different diseases [50]. These networks consist of several layers and include convolutional layers, pooling layers and fully connected layers, and are combined to deliver correct classification [51]. In the convolutional neural network model, features are obtained by using the learnable filters with convolutional layers on pixel values in the training process. Then, the pooling layers are used to down sample the resulting feature maps, thus, dimensionality is reduced and their computational efficiency is enhanced. Lastly, fully connected layers are used to perform the classification task by using the extracted features [51].

Convolutional neural networks or ConvNets are a type of deep learning architecture that is specifically created to process input data that has a natural spatial organization. They are seriously used in computer vision problems, which require image data since they can conveniently take advantage of the grid pattern of pixels. The reason behind CNNs is that they first learn the low-level features like edges and key points, which are gradually integrated into representations that are more complex. The convolutional term is named after the application of convolution operations that are linear mathematical operations utilized to derive meaningful features out of the input data [52].

The convolutional neural networks (CNNs) utilize convolution processes within at least one of the layers as opposed to general matrix multiplication as favored by the traditional feedforward deep neural networks [53]. Compared to other methods of classification, CNNs need much less input data preprocessing. Due to these strengths, CNNs have been massively used in activities like face recognition, person recognition, tumor classificaion, object recognition, traffic sign classification and diagnosis of skin diseases. Convolutional neural networks (also referred to as ConvNets) are deep learning models that are capable of processing input data in a spatial format. They take advantage of the grid-like pixel arrangement, which is most often utilized in computer vision tasks in computer vision [54].

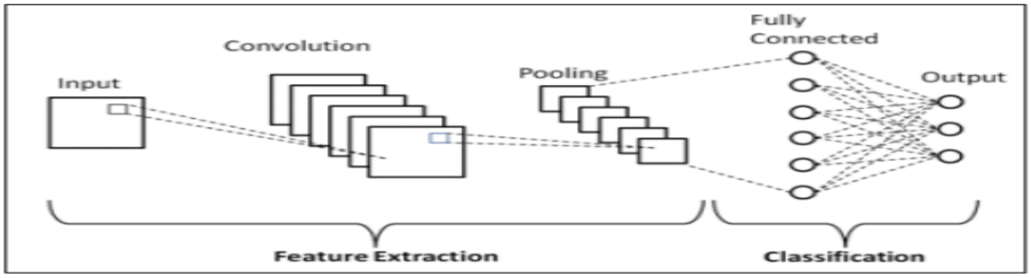


Figure 6 CNN architecture

Convolution Layer

A convolutional neural network with a specific depth can learn a lot of varied features on an input data through the several convolutional layers. The initial convolutional layers tend to learn the simpler features of the image like edges, lines, and textures, whereas the later layers acquire the more complex and abstract features of the input image [55].

The convolutional layer is a key component in a CNN. The convolutional layer can learn features from input data, or feature maps. CNNs use kernel filters to convolute the input that provide output via element wise multiplication and addition. These feature maps can sourced from next layers. Unlike other layers in a neural network, the convolutional layer utilizes learnable filters that transform the input image into a meaningful feature map. The convolutional filter is different from the feature maps. Convolutional filters are two-dimensional matrices, and feature maps are extracted based on the convolution filters. For example, here is a 4x4 pixel input image; below are the two convolution filters.

These results of the above 4x4 pixel input image in the two filters a and b are listed below c and d respectively. Both results are 3x3 pixels as shown below.

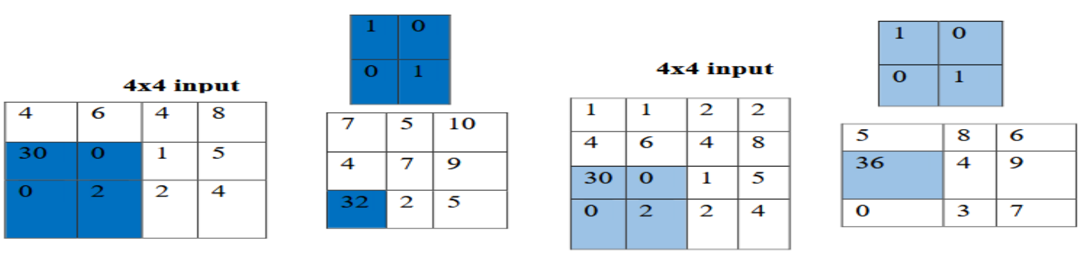


Figure 7 Result of CNN Convolution Operation on Two Different Filters

The largest value of result c is 32, which is also generated from the block like the convolutional filter in that both have value at the first and fourth positions and zero value at the second and third positions. When the input block matches the convolutional filter, the convolution operation has a large value. The largest value of result d is 36 with the same 4x4 input but second convolution filter b. It is also generated from the block like the convolutional filter b, which I used in my example above.

Pooling Layer

In CNNs, a convolutional layer is succeeded by a pooling layer to unravel the spatial dimension of the feature maps produced in the convolution by decreasing the inter layer connections but acting on each feature map individually. One can pool in various forms however, its major job is to summarize and retain the most pertinent details in the preceding convolutional layer. Calculation of poling is easy, and maximum pooling as well as average poling are the most frequent forms of polling [56].

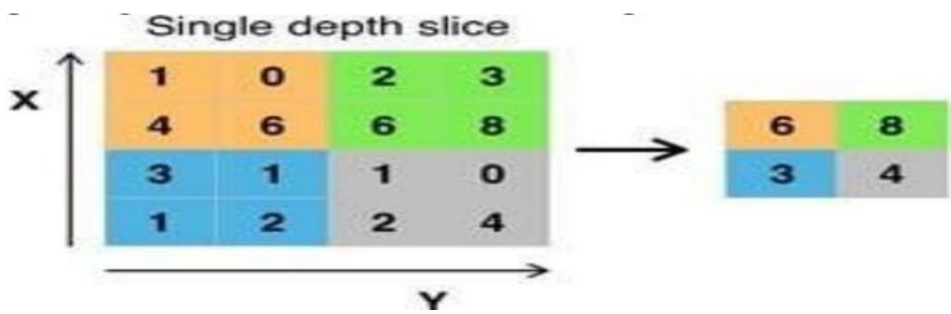


Figure 8 Max Pooling Operation on 4 x 4 Input.

Fully Connected layer

The original neural networks are intermediated by fully connected (FC) layers which are dense i.e. every neuron connects to all neurons of the inline layers. This plays a role in a high level of computational complexity. The most important aspect of FC layers is the full interconnection of sequential layers. This set of layers will compute the probability of each class and usually comes at the end of a Conventional neural network, preceding the classifier layer, which is normally implemented by a softmax function. Fully connected and dense layers are very significant to deep learning models since they perform the same task of classifying images. Dense layers are expected to make the models less complex in terms of the number of parameters presented in them and to keep the necessary information without overly extending it [57].

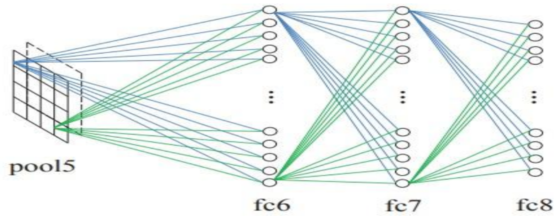


Figure 9 The operation of the fully connected layer

2.6.3.1 Custom Convolutional neural network

Convolutional neural network involves a series of perceptron units analyzing image inputs and they have learnable weights and biases to various parts of the image and have the ability to segment each other. It utilizes local spatial coherence of the images and weights are saved as some of the weights are shared. It is evident that CNN is memory and complexity efficient [58]. The initial layers learn the high-level features and the subsequent layers learn the low-level features. The figure 9 presents the conceptual model of CNN, and the types of layers are discussed below [59]. Custom CNNs give researchers the opportunity to enhance the network architecture to suit skin images to better classify disease in a computationally efficient manner. Besides, the research examines the impact of data augmentation methods on the model performance [60].

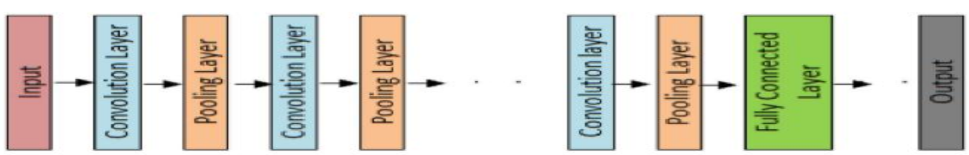


Figure 10 Architecture of custom Convolutional Neural Network

2.6.3.2 Inception-V3 Model

Google Net (Inception-v3) is a model optimized for Image Net. It was developed by Google to make it easier to scale deep networks by using computational resources more efficiently, and can handle images with a size of 299 by 299 pixels [61].

2.6.3.3 VGG19

VGG19 is a pre-trained conventional neural network with 19 layers as the name suggests. We finetune the trained VGG19 model on each of the layers and replace the top layers with four full connected layers. The classification completed with the help of a softmax activation function. In order to ensure the compatibility of the architecture with the original structure, we scale the images to fit the conventional size of VGG19. VGG19 original architecture was trained on ImageNet and had 1,000 categories of objects [62].

Figure 11 VGG19 Architecture

2.6.3.4 VGG16 Network

Image Net is an example of a dataset used with the Visual Geometry Group that was developed at the University of Oxford. The model characterized by the fact that it uses 16 layers and 3x3 filters in the convolutional layers. They are supposed to take up images of 224 by 224 size as input [59]. VGG-16 well reported as being simple but realistically structured, with the capacity to deliver robust performance in a gamut of computer vision challenges, which includes image categorization and object recognition. The architecture consists of sequential convolutional layers and then max-pooling layers with the depth of the network increasing in a regular manner. The hierarchical feature representations that the model learns with the help of this structural design achieves an accurate and robust prediction. Despite the numerous recently introduced architectures, VGG-16 is used in a substantial number of deep learning applications because it is a flexible yet reliable architecture [63].

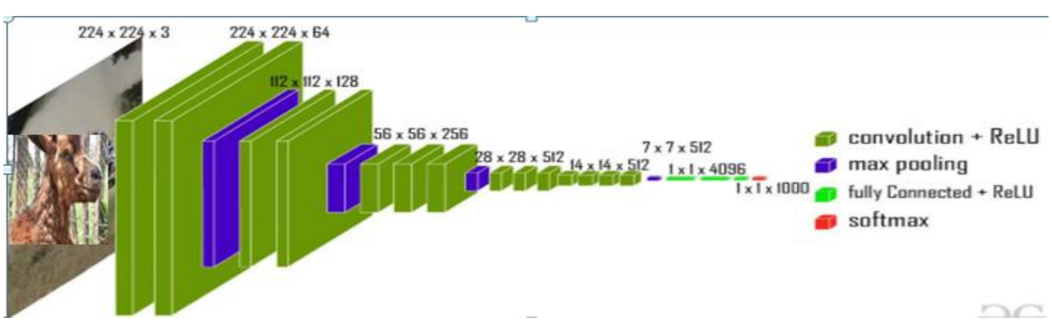


Figure 12 VGG-16 **Architecture**

2.6.3.5 ResNet50

ResNet50 Residual Network (ResNet) ResNet50 is a variation of Residual Network (ResNet), which is a convolutional neuron network model containing 48 convolutional layers, 1 max-pooling layer, and 1 average-pooling layer. Many of the applications of computer vision based on ResNet. The biggest addition to deep neural networks brought about by ResNet is the residual learning property of the network that enables the training of deep neural networks that contain over 150 layers. ResNet Sun was originally proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian in their 2015 article titled deep residual learning and explains the ResNet50 architecture [64].

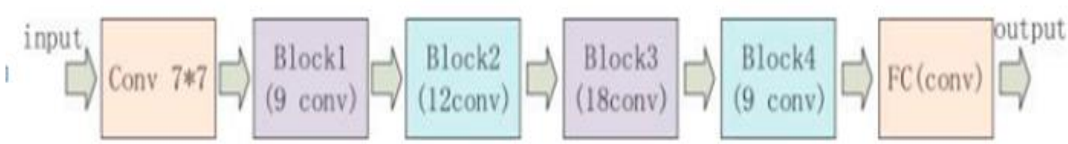


Figure 13 ResNet architecture

2.6.3.6 DenseNet201

DenseNet was developed by Gao Huang and his team in 2017 [65]. During four extremely simple object recognition tasks, the DenseNet outperformed the previous state-of-the-art architectures [65]. DenseNet is a type of deep learning models that are the expansion of ResNet that tries to enhance gradient flow within the network. DenseNet has the capacity of enhancing the flow of information and gradient flow of the network using fewer parameters as compared to the traditional convolution networks. All layers of the DenseNet structure are directly connected to the original signal and loss function, as well as, have dense connections, which affect the way data are arranged. DenseNet201 is a narrowed network that offers a strongly parameter effective model that is simple to train due to the re-use of features by layers that maximize variation in layer inputs and enhance performance [66].

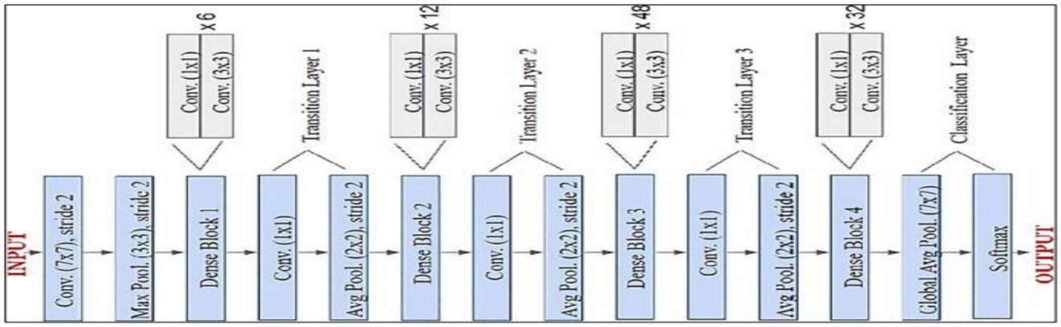


Figure 14 DenseNet201 Layered Architecture[67]

2.6.3.7 MobileNet

MobileNet is an extremely efficient CNN model that developed to work with mobile embedded vision. Google MobileNet V2 Designed by researchers that is an improvement of its predecessor, MobileNet V1, with better accuracy and lower complexity of computation [55].

2.7 Empirical Literature Review and Related work

There have been a number of studies conducted to find skin disease in animals using machine learning and image processing techniques, but the majority of them investigated either cattle or human skin diseases, while relatively less effort has been made to investigate sheep and goat skin diseases. We summarize these studies in the table below that these studies show that deep learning models can used for skin disease classification when trained from labeled and preprocessed image datasets. There are a large number of studies but only a large number of studies only focused on goat and sheep skin diseases.

Deep learning-based diagnostics in dermatology continue to be highly successful. **Shajahan and Joseph (2023)** demonstrated a CNN-based multiclass skin disease classification system training on transfer learning of diverse dermoscopic images. They demonstrated that CNNs were capable of extracting complex features of lesion and outperformed other machine learning methods with accurate results even in limited resources. However, the study focuses only on human skin disease and does not deal with the related issues in veterinary dermatology [51].

Another study comparing CNN models in dermatology demonstrated the utility of deep learning. **Rangaswamy et al. (2024)** evaluated InceptionV3 and VGG16 in classifying 13 human skin diseases using a large augmented dataset. InceptionV3 exhibited the highest accuracy, 80.88%, indicating that it has a higher capacity for feature extraction than VGG16. Although this study indicates that CNN architectures can be utilized in clinical screening, the study only explored human skin and did not address the classification needs of animals [68].

Rao et al. propose an algorithm using CNN to classify seven dermatological conditions from more than 10,000 dermatoscopic images in the HAM10000 dataset with about 93% accuracy. All the steps of data cleaning, enhancement, CNN modeling, and evaluation are accompanied by visual measures such as confusion matrices and loss/gain metrics [69].

Nigat et al. classify four fungal skin diseases using 407 dermatologist-labeled images, expanded to 1,069 by augmenting, and achieved 93.3% accuracy in their best-performing CNN using 224 x 224 images size. Their study contributes to the less well-studied field of tinea-related fungal infections and compares the proposed model with MobileNetV2 and ResNet50 where it performed better. While their internal results are excellent, the authors suggest that future multi-site data collections and further evaluation should be done before applying this method [70].

Samuel Gedefa develops deep learning models to classify cattle skin diseases, Lumpy skin disease, circle worm, wart, and normal from 444 images collected from Ethiopian field sites and online sources, and compares a custom CNN with InceptionV3 and EfficientNetB0, the latter having 99.1% accuracy. The thesis highlights the benefit of transfer learning for small data sets and details the preprocessing, training, and assessment of the models. But, the generalizability of our models is limited due to the dataset size, regional focus, and weak annotation quality, suggesting that larger data should be collected, expert-labeled, and field-ready methods, such as mobile applications, are required [71].

2.8 Research Gap

Table 1 summarizes related studies and highlights the limitations that motivate this research. Unlike existing works, this study focuses specifically on goat and sheep skin disease classification using deep learning with collected datasets.

Table 1 Gap analysis

Study	Focus Area	Model Used	Key Finding	Limitation
Muhaba et al. (2022)[11]	Human skin disease identification	MobileNet-V2 with clinical images + patient info	Accomplished 97.5% used smartphone images; real-time capable	Focused on human dermatology; no goats and sheep specific models.
Hwang et al. (2022)[12]	Dog skin disease classification	CNNs (ResNet, DenseNet, MobileNet)	Consensus models improved accuracy to 89%	Lacks disease severity classification; dataset not for goats
Fesseha et al. (2021)[72]	Mange mite prevalence in goats (South Omo)	Cross-sectional study, skin scraping, microscopy	Sarcoptes is the most prevalent mite; mange affects skin quality and production.	No use of imaging or AI; diagnosis is invasive, manual, and time-consuming.
Elias Girma (2021)[73]	Detection of Lumpy Skin Disease using deep learning	CNN for feature extraction, SVM for classification	Achieved 95.7% accuracy; image preprocessing improved performance.	Focused on cattle, not goat and sheep;
Workee G. M. (2021)[74]	Cattle skin disease identification	CNN, HOG, SVM; hybrid feature extraction	98.75% accuracy-using hybrid CNN + HOG + SVM; median filter effective in noise reduction.	Goat skin characteristics differ; dataset and results specific to cattle.

Summary

In spite of growing efforts to struggle with sheep and goat skin diseases, present studies disclose significant gaps in automatic classification approaches. Most current research, such as that by Fesseha et al., depends on labor-intensive clinical techniques like skin scraping and microscopy, which are aggressive, slow, and require skilled handling. While deep learning models like conventional neural networks effectively apply to cattle and human skin diseases, these models are not tailored to the unique characteristics of sheep and goats skin. Moreover, there is a lack of marked, goat, and sheep specific image datasets and no real-world deployment of deep learning models in field settings for real-time diagnosis.

To address these gaps, the present study focuses on developing a deep learning-based classification system specifically for goat and sheep skin diseases**,** using real-world image data and evaluating multiple CNN architectures. Furthermore, the study emphasizes practical deployment through a web-based system**,** enabling real-time and accessible disease diagnosis.

Therefore, this research addresses these gaps by:

- Developing a labeled dataset of goat and sheep skin disease images collected from real field conditions.
- Designing and evaluating deep learning models (VGG19, DenseNet201, InceptionV3, Custom CNN) specifically for this task
- Identifying the most effective model (DenseNet201) for accurate classification
- Implementing a web-based system using Django for practical deployment and accessibility

Chapter Three

Research Methodology

3.1 Introduction

The chapter explains the research methodology used to develop an improved goat and sheep skin disease classification system based on deep learning algorithms. The chapter documents the general structure that is used to realize the study goals, the design plan, data gathering procedure, data preprocessing, development of models, and evaluation methods**.**

This study uses an experimental research design. This method is quantitative in nature, because it is based on numerical measures of evaluation based on image based datasets and convolutional neural network (CNN) models. The research trains and tests deep learning algorithms systematically to find visual patterns in infected and healthy sheep and goat skin images, and to quantify their performance in classification using standard evaluation measures. Through structured data collection from veterinary clinics in the rural field settings, images of infected and healthy sheep and goats images are collected. Lighting normalization, noise removal, and background normalization are done as initial processing of photos. Deep learning model training and testing on preprocessed dataset is done to allow the classification of diseases accurately.

The methodology described in this chapter provides the basis of the rest of the research work, and the developed model is robust, efficient, and applicable in real-life veterinary diagnosis and livestock management situation.

3.2 Research Design

A research design is a structured plan for data collection and analysis and analysis designed to respond to the research issue. This study aims to develop a deep learning model classification of goat and sheep skin disease. The given study was an experimental research, which is a method that applies algorithms and evaluating the effectiveness and efficiency of those algorithms in an empirical manner by solving real-world problems. The experimental research includes the preparation of the dataset, implementation, and performance analysis.

3.3 System Development Workflow

The system development workflow demonstrates the most important factors, interrelations, and processes within a research study. It is used to illustrate how inputs influence and lead to proposed impacts.

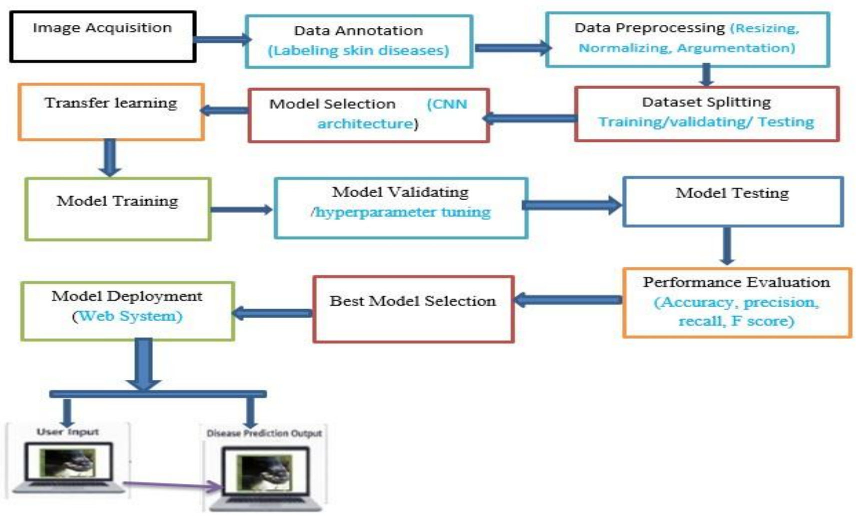


Figure 15 System development workflow diagram

3.4 Data Collection methods

Data source

Primary Source: The dataset was collected from the Southern Zone of the Tigray Region, the Endamoni Wereda animal resource development and healthcare office, veterinary clinics, and field experts for this study. Data were self-collected using a smartphone camera in local veterinary clinics or goat and sheep farms with visible skin diseases (viral, bacterial, parasitic, and healthy).

Secondary sources: Secondary source images obtained from publicly available sources through internet search engines, not open-source data repositories. Google Image Search is an aggregation tool that indexes images from multiple third-party websites; therefore, the original sources of the images vary. Both image sources were carefully reviewed and relabeled under expert supervision (Dr. Haftu Guben, veterinary specialist) into four categories: bacterial, viral, parasitic, and healthy skin conditions. After labeling, both datasets were merged into a single unified dataset. To ensure consistency, all images were resized to 224×224 pixels and normalized using a standard preprocessing pipeline. The combined dataset was randomly split into training, and validation, to preserve class distribution. The testing dataset if from the primary source only.

Due to the aggregation nature of Google Image Search, specific source URLs for individual secondary images cannot be provided. All secondary images were used exclusively for academic research and model training purposes under fair use provisions for non-commercial scientific research. No copyrighted images are redistributed with this thesis. The combined dataset is available for research purposes upon reasonable request to the author.

Data Collection: An image dataset used to achieve the purpose of this research. The model has to be trained, validated, and tested on a set of image data samples collected and labeled. Image data can be collected using images from various veterinary practices, as well as images obtained from the internet for training purposes. The images were taken by using a digital camera in certain situations to test the 18-megapixel images taken by a Samsung phone camera and prepare the dataset [74].

Data Preparation: Images obtained can used in different veterinary clinics and pictures found online to create datasets. The testing was done with photographs of a local veterinary clinic whereas the model training was done with photographs from online sources. These animal skin image files including the healthy and infected ones were manually separated and the infected ones divided into severe and mild cases. The training process uses the data that was formed on the basis of 80% of the dataset, whereas the remaining 20% of the data are utilized in the model validating and testing [75].

Data augmentation

Altering the training data to produce new, artificial training data referred to as Data Augmentation. This is often being used in machine learning and deep learning to enhance the amount of data that is available to train a network without incurring extra data collection costs[76].

3.5 Data Processing Techniques

Data analysis and transformation into significant information assists us in making improved decisions and analysis. This is referred to as Data processing. It is highly significant in businesses, including healthcare. It applies techniques and procedures to ensure that the data is accurate, practical and systematized. There are six stages in the data processing lifecycle:

- **Collection**: We collect the information in such places as farm, veterinary clinic or internet.
- **Preprocessing** Data cleaning. We eliminate noises, scale and structure it for easy analysis.
- **Input**: The structured data were stored in Google Drive. This is possible through Google colab. Use automated methods.
- **Processing**: We perform such tasks as summarizing, combining and sorting the data. This assists us in discovering meanings and trends.
- **Output**: We display the output in forms such as reports, graphs, and dashboards. This assists us in interpreting and decision-making.

3.6 Dataset Preparation

Digital cameras were used to capture images of sheep and goat skin diseases. Images were captured of three different diseases of goats and sheep: viral, bacterial, parasitic, and healthy skin. Since the image, datasets were not of the same size all of the images were resized to 224 × 224 pixels and were saved in the JPG format. Image resolution refers to the method used in this experiment to change the size of the image dataset. For this study, sheep and goat skin disease images classified based on their disease type and stored in Google Drive. In the image acquisition phase, we have taken images of the four skin diseases of goats and sheep (viral, bacterial, parasitic, and healthy).

3.6.1 Split Dataset Ratio

When we say splitting a dataset in a ratio, it means dividing the collection of data into smaller chunks according to a given proportion such as 80:20. The ratio can be broken down into its multiple elements, each of which specifies how many of the available data is to go in some sub-sets typically used for splitting data into training and testing sets in machine learning. For instance, in an 80:20 split 80% of the data used to train a model and the other 20%for validation and test to determine its predictive abilities.

Table 2 Data split and Class Distribution

Types of skin disease	Total image datasets per class	Training dataset	Validating dataset	Testing dataset
Bacterial	564	454	55	55
Healthy	572	456	58	58
Parasitic	555	443	56	56
Viral	650	520	65	65
Total	2341	1873	234	234

3.6.2 Image Resizing

Images are standardized to a uniform resolution. Specifically, a square aspect ratio of **224x224** pixels is adopted, conforming to the input requirements of standard neural network architectures [77].

3.6.3 Normalization

Image normalization was used to provide consistent and effective training of the deep learning models. In the case of the **custom CNN** model, the values of all the input images in the image space were brought to [0, 1] through a division of each pixel value by 255. This guarantees the numerical stability and increased convergence speed in the course of training.

In the case of the pre-trained models (VGG19, InceptionV3, and DenseNet201), the corresponding ImageNet preprocessing functions (preprocess input) were used. These functions do model-specific normalization in line with the ImageNet training distribution:

VGG19: transforms RGB values with ImageNet mean subtraction (channel-wise centering). InceptionV3: pixel values are scaled to the range between -1 and 1.

DenseNet201: uses ImageNet mean and standard deviation normalization.

This normalization approach is a hybrid to be compatible with the pre-trained weights of each of the architectures and enhance the performance of feature extraction.

3.6.4 Sample Dataset

A sample dataset is a subset of the entire dataset. It shows the features of all dataset.

Figure 16 Sample dataset

3.6.5 Data Augmentation

Data augmentation can increase training data without collecting or labeling more data. Instead, it enriches the data distribution by slightly modifying existing data or synthesizing new data. Data augmentation applied to increase dataset diversity and reduce overfitting. The following transformations used such as Rotation (±30 degrees), zoom range (0.2–0.3), horizontal flipping enabled, and width and height shift. Augmentations applied only for the training dataset to improve the generalization capability of the proposed CNN models.

3.7 Design of Goat and Sheep Skin disease Classification Model

We used cutting-edge deep learning models, such as convolutional neural networks, to create a classification model for sheep and goat skin disease types utilizing a deep learning technique. Initially, we were gathering dataset of skin illness images from farms and veterinary clinics. A model for identifying skin diseases was created to assist medical professionals, animal health officers, and veterinarians in precisely analyzing images to identify diseases. The study process of classifying many types of goat and sheep skin diseases and includes data gathering, data splitting, preprocessing, model evaluation, training, testing, validating, model construction. This study proposes a model for classifying goat and sheep skin diseases.

To begin, a collection of images of sheep and goat skin diseases gathered, along with an explanation of each one. A convolutional neural network or another deep learning based image categorization model trained using this dataset. Images of sheep and goat skin diseases and their labels are shown to the model during training. After that, we check how well the model works using a set of data. The model is adjusted to make predictions of sheep and goat skin diseases.

3.7.1 Classification

A Deep learning technique called classification assigns a categorical label or class to an input sample according to predetermined features or attributes of the sample. Using appropriate techniques that match the image pattern with photographs within the database, the images categorized into predetermined groups according to the retrieved features. Accurately predicting a sample class label from its features is the aim of classification. Regarding the classification of goat and sheep skin diseases, the model is designed to classify them into many groups according to characteristics like size, shape, color, surface, and texture. These features extracted from images of skin diseases and then used as input to the classification model.

3.7.2 Proposed Goat and Sheep Skin Disease Classification Model Architecture

The model is built from the input section, where we input raw RGB images of sheep and goat skin. These images resized to 224 x 224 x 3 pixels, show close-up images of animal skin, displaying lesions and diseases. A 3x2 grid of six example images illustrates the various healthy and affected skin conditions, providing the inputs to the whole classification process.

The core model is DenseNet201, which has been pre-trained on the ImageNet dataset, which is the feature extractor. Its convolutional layers learn hierarchical visual patterns such as edges, textures, and colors from the input images. The dense connectivity of DenseNet201, where every layer takes inputs from all prior layers, supports feature reuse and avoids the waning gradient problem. This pre-trained model is connected to a custom classification head designed for the skin disease task.

The custom head starts with global average pooling to reduce the size of features and batch normalization for stable training. It has two dense layers 512 and 256 neurons active with ReLU for non-linearity and dropout layers (50%, 50% and 30% respectively) to prevent overfitting. The output layer uses softmax activation to compute probabilities in four classes Bacterial, Healthy, Parasitic, and Viral. Transfer learning provides an effective way to classify sheep and goat skin by using powerful features from the previous step combined with task-specific knowledge.

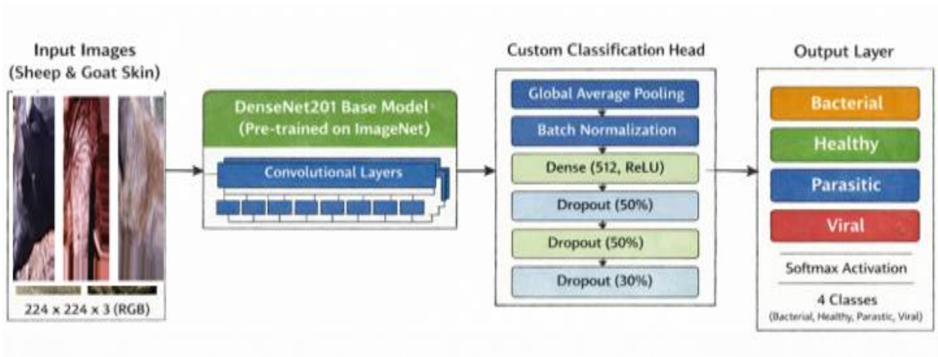


Figure 17 Proposed Goat and Sheep Skin Disease Classification Model

3.8 Transfer learning

Transfer learning is a powerful technique in deep learning that leverages pre-trained models to improve learning efficiency on new, but related tasks is called Transfer learning. Instead of training a deep learning model from scratch, which often requires vast amounts of labeled data and computational resources, transfer learning enables the use of knowledge gained from one problem and applies it to another. This concept mimics the human ability to transfer knowledge from previously learned tasks to new situations, enhancing the speed and accuracy of learning in scenarios with limited data[78].

3.9 Model Training

Model training is the process of developing the model and achieving the training process on the training dataset using sequential model, CNN-keras, VGG-19, InceptionV3, and DenseNet201 from scratch. In this study, several deep learning models, including VGG19, InceptionV3, DenseNet201, and a custom CNN, were reviewed for their feature extraction capabilities. Each model has unique strengths in capturing image features. After experimental evaluation, DenseNet201 was selected as the final model for this research due to its superior feature reuse capability and better performance on the goat and sheep skin dataset. Its dense connectivity improves gradient flow and enables effective learning of complex patterns compared to the other models. Therefore, DenseNet201 accepted as the primary model for feature extraction and classification in this study.

Model Training Configuration: The proposed deep learning models were trained using a supervised learning approach on labeled image datasets. The training process was carried out using the Keras library with a TensorFlow backend. The models were optimized using the Adam optimizer due to its efficiency in handling large datasets and adaptive learning capabilities. A categorical cross-entropy loss function was used since the problem involves multi-class classification. The training process was conducted for a fixed number of epochs, and performance was monitored using validation data to prevent overfitting. Early stopping was applied to stop training when validation performance stopped improving.

Overfitting Control: To improve generalization and reduce overfitting, dropout layers were used in the fully connected layers. Additionally, data augmentation techniques such as rotation, zooming, and horizontal flipping were applied during training. Early stopping based on validation loss was also implemented to prevent excessive training.

3.10 Validation and Testing Process

To evaluate and improve the model's performance, techniques are employed to ensure the testing data is completely separate from the training data. This allows the classifiers to be validated using unseen datasets. In this study, 10% of the dataset was used for validation, while the testing set also represented 10% of the class distribution. The training set contained the remaining 80% of the data. This approach prevents unfair data from influencing the model during training. The overall dataset was split into a ratio of 80:20 for training and testing/validation respectively.

3.11 Evaluation of the Metrics

We evaluate the performance of the model by assessing the accuracy of the model, which represents the proportion of predictions that are correct from all the inputs used in training and validation. We also use cross-entropy to understand how well the model is performing during training, where it indicates the loss for each epoch. Both accuracy and loss are key metrics to assess the model. The sample data set is balanced across all classes. We also compute accuracy, precision, recall and f1-score to obtain a detailed report on the performance of each of the four classes, allowing us to identify the well-performing and misclassified classes to misclassification [79]. On evaluation measures, the algorithm is compared with custom CNN, InceptionV3, VGG19, and

DenseNet201. These tests are to determine which model performs best developed in the suggestion stage and improved in development and the model for classifying sheep and goat skin disease. Model data is evaluated as accuracy, recall, precision, and F1 score [80]. The confusion matrix places the predictions into four important predictions in a classification context [81].

- **True Positives (TP):** These occur when the model correctly predicts "yes," indicating that the condition (skin disease) is present, and the animal indeed has the condition**.**
- **True Negatives (TN):** These occur when the model correctly predicts "no", meaning the condition is absent, and the person does not have the condition.
- **False Positives (FP):** The occurrence of the model predicts "yes" incorrectly, indicating the condition is present when it is no. This is called a "Type I error".
- **False Negatives (FN):** The occurrence of the model predicts "no" incorrectly, indicating the condition is absent when it is present. This called a "Type II error".

3.11.1 Accuracy

The accuracy of the model is determined by the percentage of true predictions of the top class, the class with the highest probability as determined by the CNN model [82]. The model should assign to the target class as the following evaluation can used to symbolize.

Accuracy is the number of TP and TN divided by the total number of TPs, TNs, FP and FN[81]. The accuracy calculated as in Equation (1).

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

3.11.2 Precision

Precision is a metric used to evaluate the performance of a classification model, particularly in scenarios where the balance between different classes is important. The precision can calculated using the formula presented in Equation 2:

$$Precision = \frac{TP}{TP + FP} \text{ -------------------------------------- } (2)$$

3.11.3 Recall

Recall (also known as Sensitivity or True Positive Rate) is a metric that measures the ability of a model to identify all relevant instances in a dataset. Equation 3) provides the formula for calculating Recall:

$$Recall = \frac{TP}{TP + FN} \text{--} (3)$$

3.11.4 F1 Score

The F1-Score is a way to measure how good a model is by combining precision and recall into one number. It finds a balance, between precision and recall. To calculate the F1-Score we use a formula. The F1-Score equals two time's precision times recall, divided by precision plus recall. This is shown in Equation (4):

$$Recall = 2\,x\,\frac{Presion\ x\ Recall}{Presion + Recall} \text{--------------------------------------} (4)$$

3.12 Implementation Tools and Libraries

Table 3 Implementation Tools and Libraries

Tool / Library	Purpose	Description
Python 3.11	Programming language	High-level, open-source language used for implementing the entire system and model development.
TensorFlow	Deep learning framework	Used for building, training, and optimizing convolutional neural network models.
Keras	Neural network API	High-level API integrated with TensorFlow for easy model design and experimentation.
NumPy	Numerical computing	Used for array operations, matrix computations, and numerical processing of image data.
Matplotlib	Data visualization	Used for plotting graphs such as accuracy/loss curves and model performance visualization.
Google Colab	Development environment	Cloud-based platform used for running Python code, training models, and accessing GPU resources.
Django	Web framework	Used to develop and deploy the web application for real-time skin disease classification.

3.13 CNN Model Hyper-parameter Proposed

Table 4 Proposed CNN Hyperparameters and Configuration

Category	Hyperparameter / Component	Description (Concise)	Proposed Setting / Role
Optimization	Adam Optimizer	Adaptive optimization method using first and second moment of gradients for efficient weight updates	Used for training CNN to minimize loss
Learning Rate	Step size for weight updates	Controls how fast model learns; too high causes instability, too low slows training	Tuned experimentally for best convergence
Loss Function	Multiclass classification loss	Measures error between predicted and true labels	Used with Softmax for four-class classification
Activation (Hidden)	ReLU	Non-linear function: outputs positive values, zeros for negatives	Applied after convolution and pooling layers
Activation (Output)	Softmax	Converts outputs into probability distribution across classes	Used in final layer (four classes)
Input Image Size	Image resolution	Standardized image input size for CNN processing	224 × 224 pixels
Convolution Filters	Feature	Extract image features using kernels	16, 32, 64, 128 filters
Filter Size	Kernel dimension	Local feature extraction window	3 × 3
Epochs	Full dataset passes	Number of times model sees entire dataset	30 epochs (optimal after experiments)
Batch Size	Training samples per update	Number of images processed before weight update	32
Model Output	Prediction summary	Displays training/testing results and model performance	Printed after training completion

3.14 System Deployment

3.14.1 Django Web Framework implementation

After the training process of the deep learning model, the system was implemented as a web application using the Django framework. The trained and evaluated model deployed in a web application to interact with users. Users can upload an image of goat and sheep skin disease captured to the application, and the model predicts the type of skin disease correctly[51]. The Django web application works as a middle platform between the client and the deep learning model. End users system can access by end user can access the system over any web browser by entering the system's URL. The browser sends an HTTP request to the web server, which processes the request and returns an HTTP response to the browser. As shown in Figure 17, the user navigates to the system URL (e.g. http://127.0.0.1:8000/) to access the home page of the system.

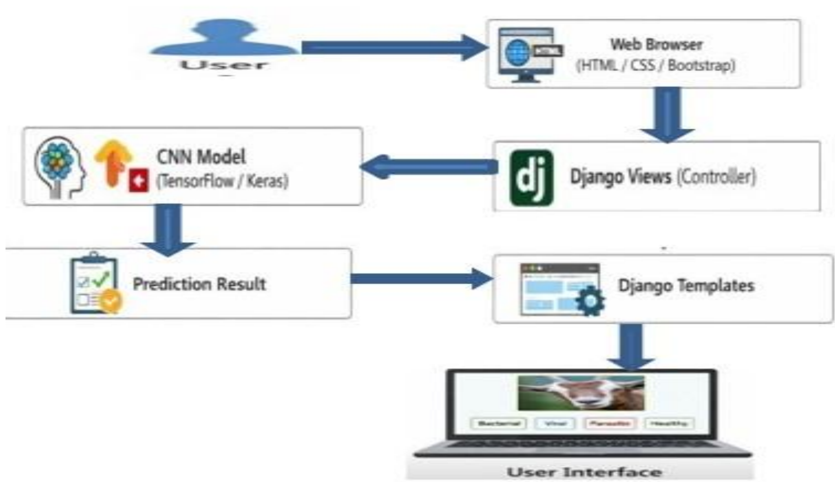


Figure 18 System architecture diagram for a deep learning web application

3.14.2 Model Integration

A trained deep learning model saved in .keras format after the training phase of the CNN architecture. This model was loaded into the Django backend using TensorFlow. The model adjusted once when the server starts to ensure well organized to classify the disease and reduced response time. Once a user uploads an image on the website, the system checks the image to make sure it is healthy or not. The preprocessing steps applied during training, such as normalization and image resizing during extrapolation to maintain consistency.

3.14.3 User Interface Design

Usability is the ease of use and clarity of a web application. User interface plays a very significant role in increasing the usability of application, as it is the medium of human computer interaction. Regardless of the degree to which the application supports the functional requirements, unless the application is efficient, easiness, and handy to the heart of user, the application has to face failure. Since UI gives the abstract view of the entire system to users, the success of the system greatly depends on it[83].

Upload Image UI: The main aspect of the user interface design is to upload image. The system allows users to upload images of either goat or sheep skin using any device from any storage. The image uploaded is also inclusive of validation messages to enhance user experience.

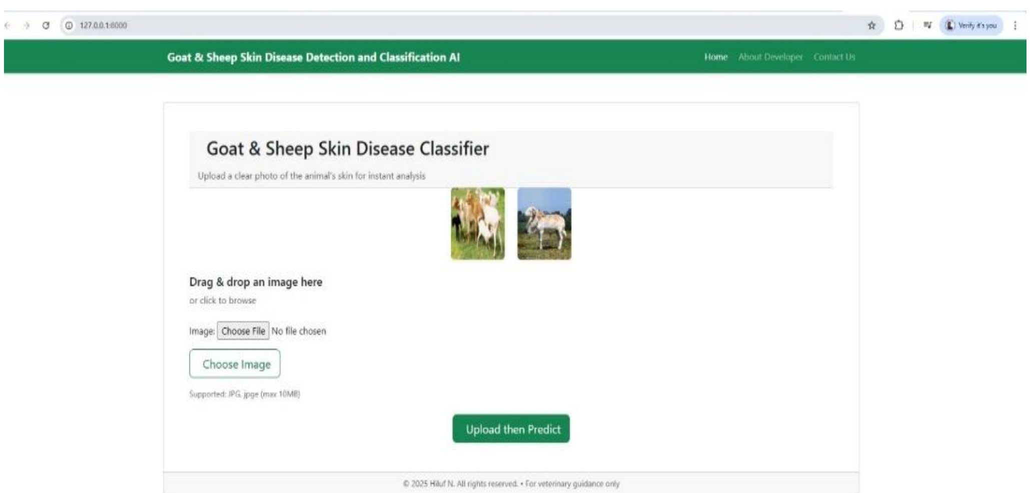
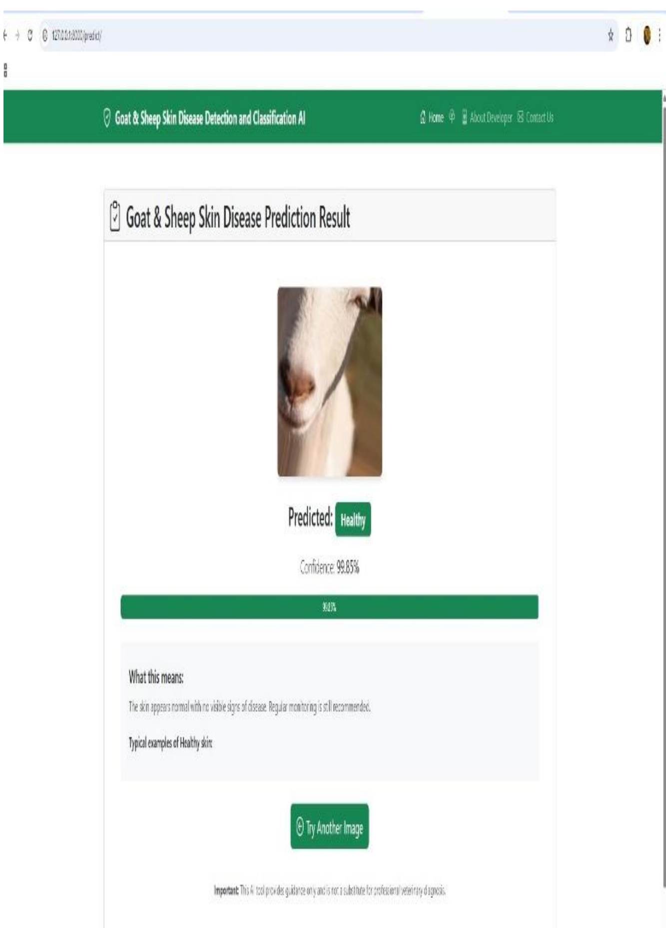


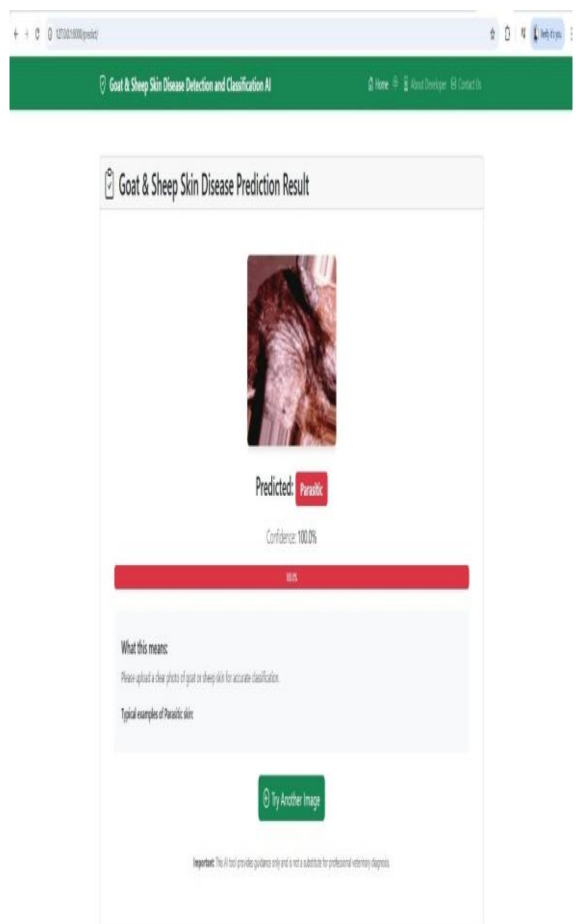
Figure 19 Image upload page

Prediction Results Display Interface

After the image is successfully uploaded, it undergoes image processing and analysis. The prediction results then generated by the trained model and displayed on a separate results page, where users view the classify disease class with relevant confidence scores.

The predictive performance illustrated in **Figure 20** provides a qualitative assessment of the model’s classification capability. To ensure experimental transparency and address potential concerns regarding selected results, it is critical to define the provenance of these samples. The four instances shown displaying confidence scores ranging from **98.2% to 100%** were drawn exclusively from the held out test set, comprising images that the model did not encounter during the training or validation phases. While these specific screenshots represent high-confidence classifications, they are statistically representative of the model's broader performance on the test data and any goat and sheep image out of dataset is work well. In test dataset achieved an overall mean confidence of **99%.** These samples were selected to demonstrate the model’s robustness across the primary disease categories (Bacterial, Viral, and Parasitic) under varying field conditions in Endamohoni wereda collection site.





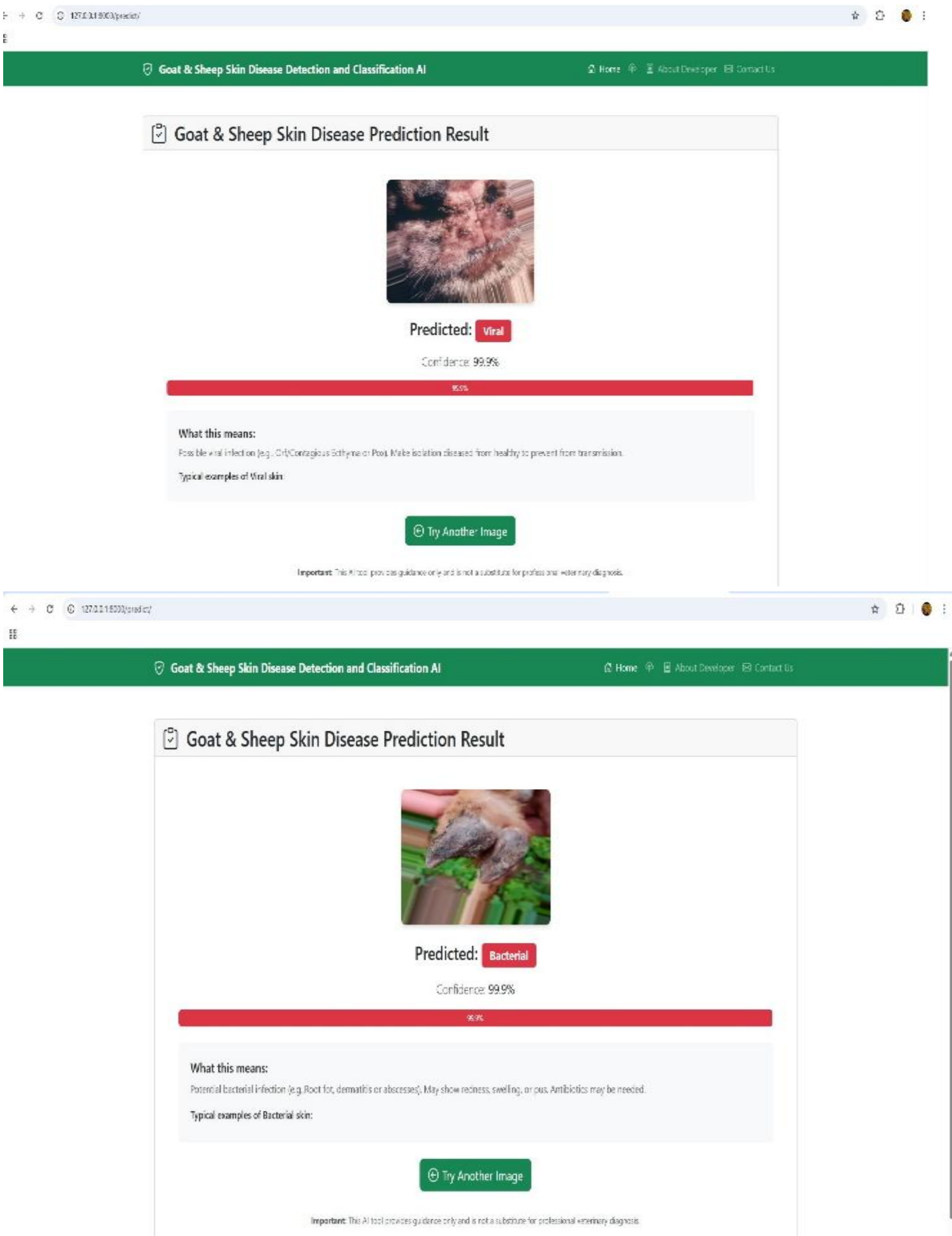


Figure 20 Prediction Results Display Interface

Chapter Four

Results and Discussion

4.1 Introduction

In this chapter, an experimental evaluation an experimental evaluation of the proposed model described in detail. The experimental evaluation confirmed the effectiveness of the proposed model. The dataset used and the implementation of the proposed model were described carefully. The effects of different filtering techniques are evaluated and compared with other state-of-the-art models are described in detail.

4.2 Dataset distribution

Dataset distribution is the pattern of data points within a dataset, showing how values spread out. It visualized using graphs to show the frequency of each value of each class. Understanding the distribution helps to identify patterns like central tendency, variability, and shape, which is important for selecting appropriate analytical methods and identifying trends or outliers.

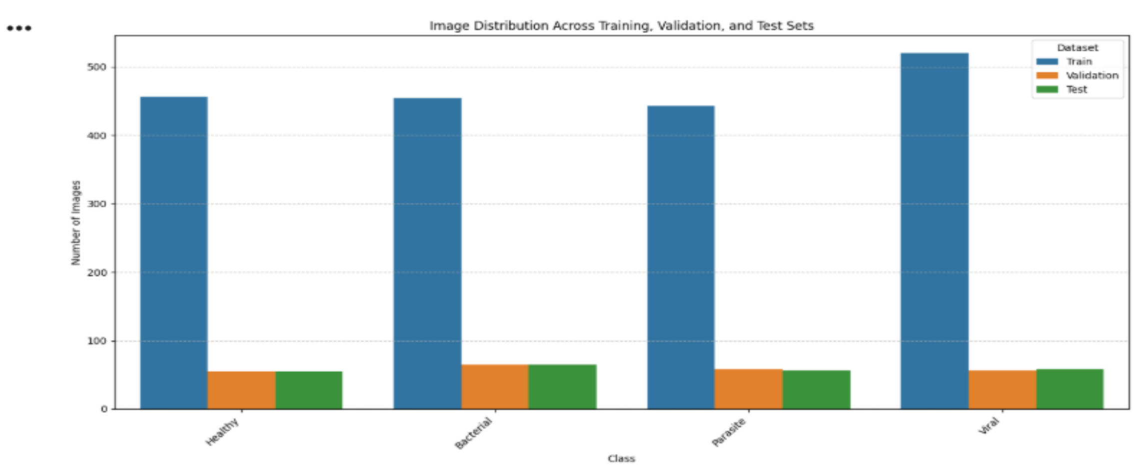


Figure 21 Data Distributions for validating dataset

4.3 Experimental Setup

A model developed and tested during this phase of the study to classify sheep and goat skin disease images, several scripts were developed and executed using a variety of models. These include Python 3.11.3, Keras, the Jupyter Notebook, transfer learning model and other libraries. Training and testing of the dataset performed using the following CNN architecture models: VGG19, CNN keras, InceptionV3, and DenseNet201. The proposed models were trained on 2341 skin disease images.

```
Summary of Datasets Split
Found 1873 images belonging to 4 classes.
Found 234 images belonging to 4 classes.
Found 234 images belonging to 4 classes.
Training images: 1873
Validation images: 234
Test images: 234
```

Figure 22 summary of the image counts in Dataset split

4.4 Experimental Results

To compare and select the most effective CNN architecture among trained and sequential models with improved dataset the following parameters are considered in each of the trials and adjusted as follows. These parameters consist of epochs, batch size, loss and learning rate. The image dataset's color system needs to be understood classification of the skin. RGB color space was used to understand recognized skin diseases. To test and assess the model's performance in classification, several models and classification algorithms used in this experiment. The following models were used for classification: CNN-keras, VGG19, InceptionV3, and DenseNet201. The weights were adjusted based on the dataset loaded into the classifier during training, with the image dataset being moved to the classifier. Unseen data were used to determine which model performs best data that does not previously seen and evaluate the classification model's effectiveness

1. IMPORT LIBRARIES

```
# ============================================================
# 1. IMPORT LIBRARIES
# ============================================================
import os
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.keras.applications import VGG19
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import confusion_matrix, classification_report
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.io as pio
pio.templates.default = "plotly_white"
import warnings
warnings.filterwarnings("ignore")
import tensorflow as tf
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Figure 23 import libraries connect to google colab drive

4.4.1 Experiment_1: VGG19 Performance Evaluation

The VGG19 model trained in this experiment on the dataset achieved a training accuracy of 90.61%, with training loss of 0.2731, and a validation accuracy of 95.33%, with validation loss of 0.167. The model demonstrated strong generalization performance, achieving a classification accuracy of 95%. Figure 25 illustrates the training and validation accuracy and loss across epochs for the VGG19 model.

```
80134624/80134624 ━━━━━━━━━━━━━━━━━━━━ 6s 0us/step
Model: "functional"
```

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 512)	12,845,568
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 4)	2,052

```
Total params: 32,872,004 (125.40 MB)
Trainable params: 12,847,620 (49.01 MB)
Non-trainable params: 20,024,384 (76.39 MB)
```

Figure 24 model Summary of VGG19 Architecture

```
Epoch 1/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 666s 10s/step - accuracy: 0.3808 - loss: 1.4811 - val_accuracy: 0.5642 - val_loss: 1.0570
Epoch 2/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 568ms/step - accuracy: 0.6268 - loss: 0.9113 - val_accuracy: 0.6651 - val_loss: 0.8527
Epoch 3/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 37s 573ms/step - accuracy: 0.6638 - loss: 0.8209 - val_accuracy: 0.7018 - val_loss: 0.8051
Epoch 4/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 548ms/step - accuracy: 0.7200 - loss: 0.7128 - val_accuracy: 0.7385 - val_loss: 0.7299
Epoch 5/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 564ms/step - accuracy: 0.7248 - loss: 0.7235 - val_accuracy: 0.7523 - val_loss: 0.6569
Epoch 6/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 563ms/step - accuracy: 0.7473 - loss: 0.6489 - val_accuracy: 0.7385 - val_loss: 0.6251
Epoch 7/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 41s 558ms/step - accuracy: 0.7328 - loss: 0.6665 - val_accuracy: 0.7844 - val_loss: 0.5938
Epoch 8/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 41s 558ms/step - accuracy: 0.7808 - loss: 0.6030 - val_accuracy: 0.7706 - val_loss: 0.5740
Epoch 9/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 546ms/step - accuracy: 0.8260 - loss: 0.4939 - val_accuracy: 0.8303 - val_loss: 0.4985
Epoch 10/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 565ms/step - accuracy: 0.8083 - loss: 0.5487 - val_accuracy: 0.7890 - val_loss: 0.5777
Epoch 11/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 544ms/step - accuracy: 0.8072 - loss: 0.5187 - val_accuracy: 0.8257 - val_loss: 0.5215
Epoch 12/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 550ms/step - accuracy: 0.8127 - loss: 0.4968 - val_accuracy: 0.8486 - val_loss: 0.4407
Epoch 13/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 550ms/step - accuracy: 0.8603 - loss: 0.4245 - val_accuracy: 0.8945 - val_loss: 0.4050
Epoch 14/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 541ms/step - accuracy: 0.8155 - loss: 0.4919 - val_accuracy: 0.8303 - val_loss: 0.4513
Epoch 15/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 560ms/step - accuracy: 0.8332 - loss: 0.4439 - val_accuracy: 0.8440 - val_loss: 0.4014
Epoch 16/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 561ms/step - accuracy: 0.8581 - loss: 0.4179 - val_accuracy: 0.8165 - val_loss: 0.4709
Epoch 17/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 534ms/step - accuracy: 0.8511 - loss: 0.4164 - val_accuracy: 0.8807 - val_loss: 0.3606
Epoch 18/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 550ms/step - accuracy: 0.8490 - loss: 0.4085 - val_accuracy: 0.9037 - val_loss: 0.3004
Epoch 19/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 530ms/step - accuracy: 0.8587 - loss: 0.3823 - val_accuracy: 0.8532 - val_loss: 0.3646
Epoch 20/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 548ms/step - accuracy: 0.8599 - loss: 0.3774 - val_accuracy: 0.8394 - val_loss: 0.3992
Epoch 21/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 549ms/step - accuracy: 0.8752 - loss: 0.3653 - val_accuracy: 0.8624 - val_loss: 0.3821
Epoch 22/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 535ms/step - accuracy: 0.8670 - loss: 0.3677 - val_accuracy: 0.9266 - val_loss: 0.2544
Epoch 23/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 560ms/step - accuracy: 0.8797 - loss: 0.3336 - val_accuracy: 0.8807 - val_loss: 0.3275
Epoch 24/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 535ms/step - accuracy: 0.8803 - loss: 0.3470 - val_accuracy: 0.9312 - val_loss: 0.2394
Epoch 25/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 545ms/step - accuracy: 0.8914 - loss: 0.3148 - val_accuracy: 0.8991 - val_loss: 0.2911
Epoch 26/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 562ms/step - accuracy: 0.8851 - loss: 0.3084 - val_accuracy: 0.8991 - val_loss: 0.3353
Epoch 27/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 564ms/step - accuracy: 0.8945 - loss: 0.3032 - val_accuracy: 0.9128 - val_loss: 0.2671
Epoch 28/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 39s 541ms/step - accuracy: 0.8896 - loss: 0.3152 - val_accuracy: 0.9037 - val_loss: 0.2636
Epoch 29/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 35s 543ms/step - accuracy: 0.8920 - loss: 0.2982 - val_accuracy: 0.8761 - val_loss: 0.2930
Epoch 30/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 36s 563ms/step - accuracy: 0.9061 - loss: 0.2734 - val_accuracy: 0.9541 - val_loss: 0.1846
Total training time: 28.89 minutes
```

```
... 8/8 ━━━━━━━━━━━━━━━━━━━━ 58s 8s/step - accuracy: 0.9522 - loss: 0.1620
     TEST EVALUATION:
    Test Loss: 0.19486278295516968
    Test Accuracy: 0.9529914259910583
```

Figure 25 Epoch Result of VGG19 Training and validating Model

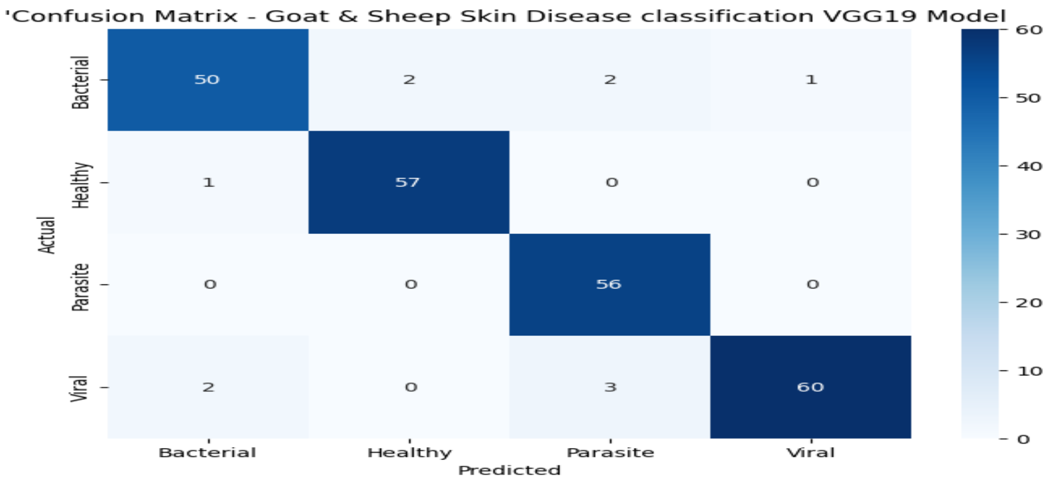


Figure 26 Confusion Matrix of VGG19 Model.

```
Classification Report:

              precision    recall  f1-score   support

   Bacterial       0.94      0.91      0.93        55
     Healthy       0.97      0.98      0.97        58
    Parasite       0.92      1.00      0.96        56
       Viral       0.98      0.92      0.95        65

    accuracy                           0.95       234
   macro avg       0.95      0.95      0.95       234
weighted avg       0.95      0.95      0.95       234
```

Figure 27 Classification report of VGG19 Model.

Figure 28 VGG19 Training, Validation Test performance curve or Plot

4.4.2 Experiment_2: Custom CNN-keras Performance Evaluation

The custom CNN-Keras model trained in this experiment on the dataset achieved a training accuracy of 90.81%, with training loss of 0.284, and a validation accuracy of 94.95%, with validation loss of 0.2626. The model demonstrated strong generalization performance, achieving a classification accuracy of 95%. Figure 30 illustrates the training and validation of accuracy and loss across epochs for the CNN-keras model.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
batch_normalization (BatchNormalization)	(None, 222, 222, 32)	128
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 109, 109, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 52, 52, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
conv2d_3 (Conv2D)	(None, 24, 24, 256)	295,168
batch_normalization_3 (BatchNormalization)	(None, 24, 24, 256)	1,024
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 256)	0
flatten (Flatten)	(None, 36864)	0
dense (Dense)	(None, 256)	9,437,440
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 4)	1,028

Total params: 9,828,804 (37.49 MB)
Trainable params: 9,827,844 (37.49 MB)
Non-trainable params: 960 (3.75 KB)

Figure 29 model Summary of custom CNN Architecture

```
Epoch 1/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 466s 7s/step - accuracy: 0.4475 - loss: 2.2866 - val_accuracy: 0.2523 - val_loss: 3.1157
Epoch 2/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 481ms/step - accuracy: 0.6248 - loss: 0.9434 - val_accuracy: 0.2523 - val_loss: 7.0409
Epoch 3/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 476ms/step - accuracy: 0.6429 - loss: 0.8607 - val_accuracy: 0.2523 - val_loss: 7.6821
Epoch 4/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 489ms/step - accuracy: 0.7159 - loss: 0.7431 - val_accuracy: 0.2569 - val_loss: 5.4326
Epoch 5/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 474ms/step - accuracy: 0.7166 - loss: 0.7160 - val_accuracy: 0.2477 - val_loss: 5.0114
Epoch 6/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 480ms/step - accuracy: 0.7325 - loss: 0.7040 - val_accuracy: 0.2339 - val_loss: 4.0954
Epoch 7/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 492ms/step - accuracy: 0.7600 - loss: 0.6275 - val_accuracy: 0.2844 - val_loss: 2.9467
Epoch 8/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 481ms/step - accuracy: 0.7648 - loss: 0.6219 - val_accuracy: 0.5642 - val_loss: 1.1888
Epoch 9/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 477ms/step - accuracy: 0.7965 - loss: 0.5627 - val_accuracy: 0.7202 - val_loss: 0.6318
Epoch 10/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 490ms/step - accuracy: 0.8034 - loss: 0.5294 - val_accuracy: 0.7936 - val_loss: 0.5420
Epoch 11/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 477ms/step - accuracy: 0.8216 - loss: 0.4902 - val_accuracy: 0.8440 - val_loss: 0.4201
Epoch 12/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 488ms/step - accuracy: 0.8335 - loss: 0.4526 - val_accuracy: 0.9128 - val_loss: 0.2755
Epoch 13/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 480ms/step - accuracy: 0.8484 - loss: 0.4776 - val_accuracy: 0.9083 - val_loss: 0.2939
Epoch 14/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 479ms/step - accuracy: 0.8308 - loss: 0.4504 - val_accuracy: 0.8945 - val_loss: 0.3075
Epoch 15/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 484ms/step - accuracy: 0.8469 - loss: 0.4222 - val_accuracy: 0.8670 - val_loss: 0.3929
Epoch 16/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 483ms/step - accuracy: 0.8403 - loss: 0.4455 - val_accuracy: 0.8349 - val_loss: 0.3656
Epoch 17/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 478ms/step - accuracy: 0.8471 - loss: 0.4190 - val_accuracy: 0.8853 - val_loss: 0.3444
Epoch 18/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 475ms/step - accuracy: 0.8534 - loss: 0.3756 - val_accuracy: 0.9358 - val_loss: 0.2390
Epoch 19/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 483ms/step - accuracy: 0.8731 - loss: 0.3651 - val_accuracy: 0.9220 - val_loss: 0.2322
Epoch 20/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 474ms/step - accuracy: 0.8635 - loss: 0.3658 - val_accuracy: 0.9083 - val_loss: 0.2577
Epoch 21/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 477ms/step - accuracy: 0.9017 - loss: 0.2814 - val_accuracy: 0.9771 - val_loss: 0.1565
Epoch 22/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 486ms/step - accuracy: 0.8942 - loss: 0.3051 - val_accuracy: 0.9174 - val_loss: 0.2416
Epoch 23/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 475ms/step - accuracy: 0.8716 - loss: 0.3306 - val_accuracy: 0.9725 - val_loss: 0.1783
Epoch 24/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 478ms/step - accuracy: 0.8889 - loss: 0.3166 - val_accuracy: 0.9633 - val_loss: 0.1566
Epoch 25/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 488ms/step - accuracy: 0.8964 - loss: 0.2772 - val_accuracy: 0.9404 - val_loss: 0.2023
Epoch 26/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 473ms/step - accuracy: 0.9009 - loss: 0.2827 - val_accuracy: 0.9633 - val_loss: 0.1619
Epoch 27/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 475ms/step - accuracy: 0.8902 - loss: 0.2970 - val_accuracy: 0.9633 - val_loss: 0.1307
Epoch 28/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 31s 488ms/step - accuracy: 0.8911 - loss: 0.2973 - val_accuracy: 0.9725 - val_loss: 0.1559
Epoch 29/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 477ms/step - accuracy: 0.8847 - loss: 0.3152 - val_accuracy: 0.9541 - val_loss: 0.1678
Epoch 30/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 30s 478ms/step - accuracy: 0.9081 - loss: 0.2626 - val_accuracy: 0.9495 - val_loss: 0.1367
Total training time: 22.99 minutes

8/8 ━━━━━━━━━━━━━━━━━━━━ 36s 5s/step - accuracy: 0.9436 - loss: 0.1690
 TEST EVALUATION:
Test Loss: 0.14695312082767487
Test Accuracy: 0.9487179517745972
```

Figure 30 Training epoch of custom CNN-keras model

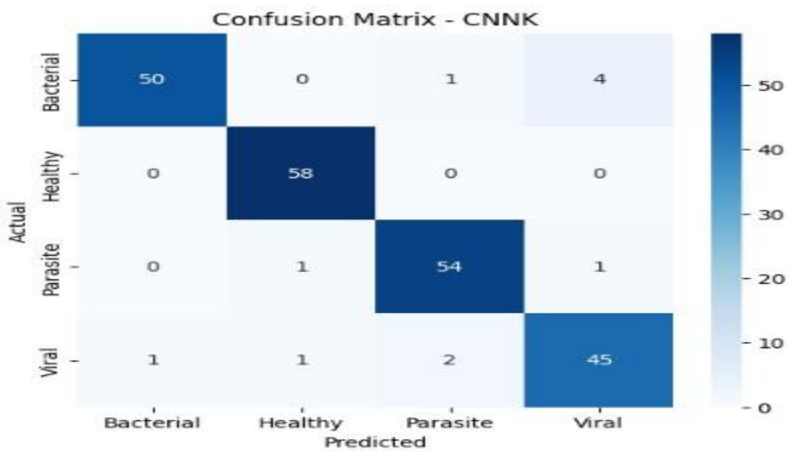


Figure 31 Confusion matrix of custom CNN-keras

Classification Report:

	precision	recall	f1-score	support
Bacterial	0.98	0.91	0.94	55
Healthy	0.97	1.00	0.98	58
Parasite	0.95	0.96	0.96	56
Viral	0.90	0.92	0.91	49
accuracy			0.95	218
macro avg	0.95	0.95	0.95	218
weighted avg	0.95	0.95	0.95	218

Figure 32 classification report of custom CNN-keras

...

Figure 33 Model accuracy and loss curve of custom CNN-keras

4.4.3 Experiment_3: DenseNet201 Performance Evaluation

During the training phase, the DenseNet201 model achieved accuracy of 95.04% and loss of 0.1524 while the validation accuracy was 97.71% and loss 0.1251. After the entire training and model selection process is finished the final performance evaluation is 97.44%. The model demonstrated strong generalization model was highly generalized with an accuracy of 97% in classification.

```
Epoch 1/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 172s 3s/step - accuracy: 0.6658 - loss: 0.8689 - val_accuracy: 0.7569 - val_loss: 0.7829
Epoch 2/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 514ms/step - accuracy: 0.7611 - loss: 0.6734 - val_accuracy: 0.7752 - val_loss: 0.6859
Epoch 3/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 528ms/step - accuracy: 0.7853 - loss: 0.5846 - val_accuracy: 0.7752 - val_loss: 0.6088
Epoch 4/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 504ms/step - accuracy: 0.8348 - loss: 0.4993 - val_accuracy: 0.8165 - val_loss: 0.5616
Epoch 5/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 522ms/step - accuracy: 0.8341 - loss: 0.4664 - val_accuracy: 0.8257 - val_loss: 0.5080
Epoch 6/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 506ms/step - accuracy: 0.8568 - loss: 0.4299 - val_accuracy: 0.8303 - val_loss: 0.4844
Epoch 7/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 537ms/step - accuracy: 0.8672 - loss: 0.4044 - val_accuracy: 0.8578 - val_loss: 0.4278
Epoch 8/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 512ms/step - accuracy: 0.8796 - loss: 0.3768 - val_accuracy: 0.8670 - val_loss: 0.4017
Epoch 9/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 527ms/step - accuracy: 0.8836 - loss: 0.3453 - val_accuracy: 0.8807 - val_loss: 0.3731
Epoch 10/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 507ms/step - accuracy: 0.8631 - loss: 0.3826 - val_accuracy: 0.8670 - val_loss: 0.3940
Epoch 11/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 528ms/step - accuracy: 0.8979 - loss: 0.2984 - val_accuracy: 0.8991 - val_loss: 0.3355
Epoch 12/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 511ms/step - accuracy: 0.8867 - loss: 0.3200 - val_accuracy: 0.8624 - val_loss: 0.3358
Epoch 13/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 528ms/step - accuracy: 0.9015 - loss: 0.2935 - val_accuracy: 0.9128 - val_loss: 0.2755
Epoch 14/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 514ms/step - accuracy: 0.8900 - loss: 0.3129 - val_accuracy: 0.8991 - val_loss: 0.2837
Epoch 15/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 532ms/step - accuracy: 0.9041 - loss: 0.2719 - val_accuracy: 0.9220 - val_loss: 0.2602
Epoch 16/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 504ms/step - accuracy: 0.9154 - loss: 0.2427 - val_accuracy: 0.9083 - val_loss: 0.2520
Epoch 17/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 516ms/step - accuracy: 0.9304 - loss: 0.2449 - val_accuracy: 0.9174 - val_loss: 0.2415
Epoch 18/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 507ms/step - accuracy: 0.9158 - loss: 0.2413 - val_accuracy: 0.9128 - val_loss: 0.2201
Epoch 19/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 34s 523ms/step - accuracy: 0.9210 - loss: 0.2316 - val_accuracy: 0.9312 - val_loss: 0.2057
Epoch 20/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 40s 510ms/step - accuracy: 0.9278 - loss: 0.2452 - val_accuracy: 0.9220 - val_loss: 0.2032
Epoch 21/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 491ms/step - accuracy: 0.9281 - loss: 0.2160 - val_accuracy: 0.9450 - val_loss: 0.1961
Epoch 22/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 495ms/step - accuracy: 0.9438 - loss: 0.2052 - val_accuracy: 0.9450 - val_loss: 0.1869
Epoch 23/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 513ms/step - accuracy: 0.9434 - loss: 0.1895 - val_accuracy: 0.9679 - val_loss: 0.1581
Epoch 24/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 495ms/step - accuracy: 0.9327 - loss: 0.2206 - val_accuracy: 0.9725 - val_loss: 0.1518
Epoch 25/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 513ms/step - accuracy: 0.9320 - loss: 0.2147 - val_accuracy: 0.9450 - val_loss: 0.1763
Epoch 26/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 496ms/step - accuracy: 0.9389 - loss: 0.1922 - val_accuracy: 0.9633 - val_loss: 0.1587
Epoch 27/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 507ms/step - accuracy: 0.9581 - loss: 0.1501 - val_accuracy: 0.9679 - val_loss: 0.1523
Epoch 28/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 496ms/step - accuracy: 0.9418 - loss: 0.1851 - val_accuracy: 0.9725 - val_loss: 0.1366
Epoch 29/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 32s 494ms/step - accuracy: 0.9533 - loss: 0.1639 - val_accuracy: 0.9679 - val_loss: 0.1275
Epoch 30/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 33s 512ms/step - accuracy: 0.9504 - loss: 0.1594 - val_accuracy: 0.9771 - val_loss: 0.1215
Total training time for DenseNet201: 18.89 minutes

8/8 ━━━━━━━━━━━━━━━━━━━━ 130s 18s/step - accuracy: 0.9765 - loss: 0.1291
Test Accuracy: 97.44%
8/8 ━━━━━━━━━━━━━━━━━━━━ 51s 3s/step
```

Figure 34 Epoch Result of DenseNet201 Training Model

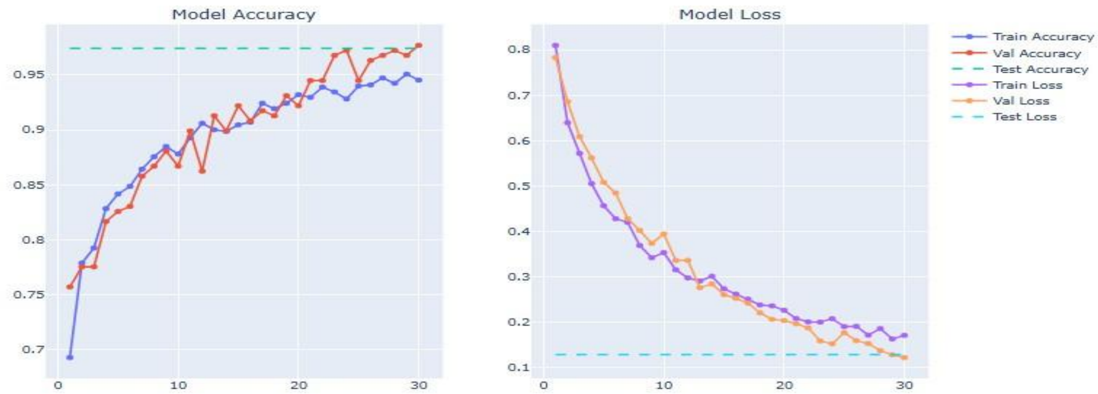


Figure 35 Model accuracy and loss curve of DenseNet201 Model

The following figure provides detailed insight into how well a model performs during training by showing the number of correct and incorrect predictions for each class in the DenseNet201 model.

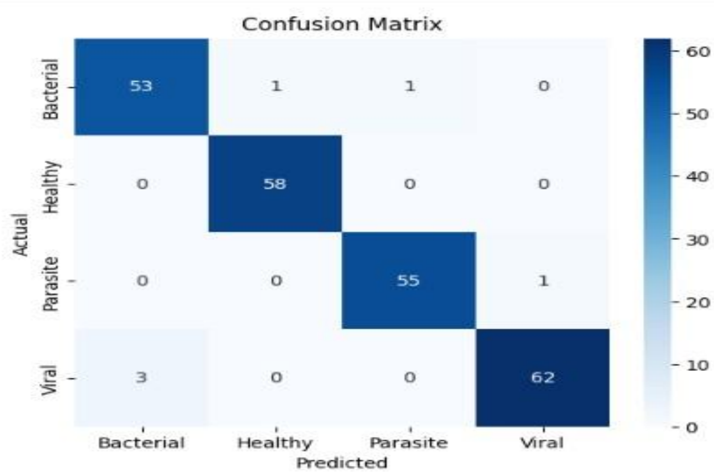


Figure 36 Confusion Matrix of Test the Model

Confusion Matrix Explanation: The confusion matrix described here allows you to evaluate the classification model by comparing the actual labels to predicted labels.

True Labels: These rows represent the classes of the test dataset.

Predicted Labels: The columns represent the predicted classes by the model.

Diagonal Elements: These diagonal numbers show the number of correct classes for each class.

Off-Diagonal Elements: These represent misclassifications (off-diagonal values).

Table 5 Confusion matrix of the best-performing model

True Label	Predicted Bacterial	Predicted Healthy	Predicted Parasitic	Predicted Viral
Bacterial	53	1	1	0
Healthy	0	58	0	0
Parasitic	0	0	55	1
Viral	3	0	0	62

Step 1: True Positives (TP): TP is the count of correct predictions for a class (diagonal element).

Bacterial TP = 53

Healthy TP = 58

Parasitic TP = 55

Viral TP = 62

Step 2: False Negatives (FN): FN is when the true class was not predicted correctly.

Bacterial FN = 1 + 1 + 0 = 2

Healthy FN = 0 + 0 + 0 = 0

Parasitic FN = 0 + 0 + 1 = 1

Viral FN = 3 + 0 + 0 = 3

Step 3: False Positives (FP): FP is when other classes are incorrectly predicted as this class.

Bacterial FP = 0 + 0 + 3 = 3

Healthy FP = 1 + 0 + 0 = 1

Parasitic FP = 1 + 0 + 0 = 1

Viral FP = 0 + 0 + 1 = 1

Step 4: True Negatives (TN)

TN is all remaining correctly classified negative cases.

Total samples = 53 + 1 + 1 + 0 + 0 + 58 + 0 + 0 + 0 + 0 + 55 + 1 + 3 + 0 + 0 + 62 = **234**

1. Bacterial: TP = 53, FN = 2, FP = 3, TN = 176, Total = 234

Accuracy = (53 + 176) / 234 = 229 / 234 =0.9786

Precision = 53 / (53 + 3) = 53 / 56 = 0.9464 Recall = 53 / (53 + 2) = 53 / 55 = 0.9636

F1-score = 2 × (0.9464 × 0.9636) / (0.9464 + 0.9636) = 0.9550

2. Healthy: TP = 58, FN = 0, FP = 1, TN = 175

Accuracy = (58 + 175) / 234 = 233 / 234 = 0.9957 Precision = 58 / (58 + 1) = 58 / 59 = 0.9831

Recall = 58 / (58 + 0) = 58 / 58 = 1.0 F1-score = 2 × (0.9831 × 1.0) / (0.9831 + 1.0) = 0.9915

3. Parasitic: TP = 55, FN = 1, FP = 1, TN = 177

Accuracy = (55 + 177) / 234 = 232 / 234 = 0.9915 Precision = 55 / (55 + 1) = 55 / 56 = 0.9821

Recall = 55 / (55 + 1) = 55 / 56 = 0.9821 F1-score = 2 × (0.9821 × 0.9821) / (0.9821 + 0.9821) = 0.9821

4. Viral: TP = 62, FN = 3, FP = 1, TN = 168

Accuracy = (62 + 168) / 234 = 230 / 234 = 0.9829 Precision = 62 / (62 + 1) = 62 / 63 = 0.9841

Recall = 62 / (62 + 3) = 62 / 65 = 0.9538 F1-score = 2 × (0.9841 × 0.9538) / (0.9841 + 0.9538) = 0.9688

Overall accuracy: Total correct predictions = sum of TPs = 53 + 58 + 55 + 62 = 228, Total samples = 234 Overall Accuracy=228/234=0.9744

The figure below characterizes classification reports of the DenseNet201 model metrics.

Classification Report:

	precision	recall	f1-score	support
Bacterial	0.95	0.96	0.95	55
Healthy	0.98	1.00	0.99	58
Parasite	0.98	0.98	0.98	56
Viral	0.98	0.95	0.97	65
accuracy			0.97	234
macro avg	0.97	0.97	0.97	234
weighted avg	0.97	0.97	0.97	234

Figure 37 Classification Reports of the DenseNet201 Model

4.4.4 Experiment_4: InceptionV3 Efficient Performance Evaluation

During the training phase, the InceptionV3 model achieved accuracy of 92.68% and loss of 0.2202 concurrently the validation accuracy of 89.91% with validation loss of 0.2615. The model demonstrated strong generalization performance, achieving a classification report accuracy of 90%. Figure 37 illustrates training and validation accuracy and loss across epochs for the InceptionV3 model.

```
Epoch 1/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 127s 2s/step - accuracy: 0.3296 - loss: 1.5281 - val_accuracy: 0.6468 - val_loss: 1.0369
Epoch 2/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 831ms/step - accuracy: 0.6421 - loss: 0.9374 - val_accuracy: 0.7385 - val_loss: 0.8435
Epoch 3/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 52s 817ms/step - accuracy: 0.7208 - loss: 0.7399 - val_accuracy: 0.7477 - val_loss: 0.7174
Epoch 4/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 814ms/step - accuracy: 0.7615 - loss: 0.6426 - val_accuracy: 0.7706 - val_loss: 0.6918
Epoch 5/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 83s 836ms/step - accuracy: 0.7865 - loss: 0.6186 - val_accuracy: 0.7890 - val_loss: 0.6311
Epoch 6/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 835ms/step - accuracy: 0.7951 - loss: 0.5589 - val_accuracy: 0.8303 - val_loss: 0.5430
Epoch 7/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 828ms/step - accuracy: 0.8282 - loss: 0.5129 - val_accuracy: 0.8394 - val_loss: 0.5323
Epoch 8/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 833ms/step - accuracy: 0.8347 - loss: 0.4826 - val_accuracy: 0.8119 - val_loss: 0.5479
Epoch 9/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 830ms/step - accuracy: 0.8660 - loss: 0.4229 - val_accuracy: 0.8165 - val_loss: 0.5440
Epoch 10/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 52s 815ms/step - accuracy: 0.8466 - loss: 0.4407 - val_accuracy: 0.8486 - val_loss: 0.4766
Epoch 11/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 52s 812ms/step - accuracy: 0.8627 - loss: 0.3941 - val_accuracy: 0.8165 - val_loss: 0.5270
Epoch 12/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 83s 830ms/step - accuracy: 0.8759 - loss: 0.3934 - val_accuracy: 0.8440 - val_loss: 0.4340
Epoch 13/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 823ms/step - accuracy: 0.8820 - loss: 0.3627 - val_accuracy: 0.8486 - val_loss: 0.4375
Epoch 14/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 52s 819ms/step - accuracy: 0.8782 - loss: 0.3661 - val_accuracy: 0.8624 - val_loss: 0.3883
Epoch 15/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 52s 815ms/step - accuracy: 0.8734 - loss: 0.3636 - val_accuracy: 0.8716 - val_loss: 0.3769
Epoch 16/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 52s 808ms/step - accuracy: 0.8977 - loss: 0.3146 - val_accuracy: 0.8486 - val_loss: 0.4206
Epoch 17/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 819ms/step - accuracy: 0.9055 - loss: 0.3096 - val_accuracy: 0.8670 - val_loss: 0.3592
Epoch 18/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 815ms/step - accuracy: 0.8793 - loss: 0.3542 - val_accuracy: 0.8761 - val_loss: 0.3412
Epoch 19/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 834ms/step - accuracy: 0.8804 - loss: 0.3561 - val_accuracy: 0.8761 - val_loss: 0.3582
Epoch 20/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 831ms/step - accuracy: 0.9046 - loss: 0.3138 - val_accuracy: 0.8945 - val_loss: 0.3236
Epoch 21/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 831ms/step - accuracy: 0.9026 - loss: 0.2913 - val_accuracy: 0.8991 - val_loss: 0.2962
Epoch 22/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 837ms/step - accuracy: 0.9064 - loss: 0.2716 - val_accuracy: 0.8991 - val_loss: 0.3110
Epoch 23/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 54s 844ms/step - accuracy: 0.9173 - loss: 0.2527 - val_accuracy: 0.8761 - val_loss: 0.2937
Epoch 24/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 54s 842ms/step - accuracy: 0.9036 - loss: 0.2868 - val_accuracy: 0.9083 - val_loss: 0.2609
Epoch 25/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 832ms/step - accuracy: 0.8877 - loss: 0.3103 - val_accuracy: 0.9037 - val_loss: 0.2487
Epoch 26/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 835ms/step - accuracy: 0.9052 - loss: 0.2702 - val_accuracy: 0.8991 - val_loss: 0.2431
Epoch 27/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 835ms/step - accuracy: 0.9205 - loss: 0.2422 - val_accuracy: 0.9083 - val_loss: 0.2493
Epoch 28/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 828ms/step - accuracy: 0.9126 - loss: 0.2749 - val_accuracy: 0.8991 - val_loss: 0.2509
Epoch 29/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 831ms/step - accuracy: 0.9184 - loss: 0.2470 - val_accuracy: 0.9128 - val_loss: 0.2208
Epoch 30/30
64/64 ━━━━━━━━━━━━━━━━━━━━ 53s 821ms/step - accuracy: 0.9289 - loss: 0.2202 - val_accuracy: 0.8991 - val_loss: 0.2615
Total training time: 28.75 minutes
  self._warn_if_super_not_called()
8/8 ━━━━━━━━━━━━━━━━━━━━ 6s 758ms/step - accuracy: 0.8663 - loss: 0.3330
Test Accuracy: 90.17%
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 144ms/step
```

Figure 38 Epoch Result of InceptionV3 Model

The following figure shows the InceptionV3 result training, validation accuracy training, and validation loss.

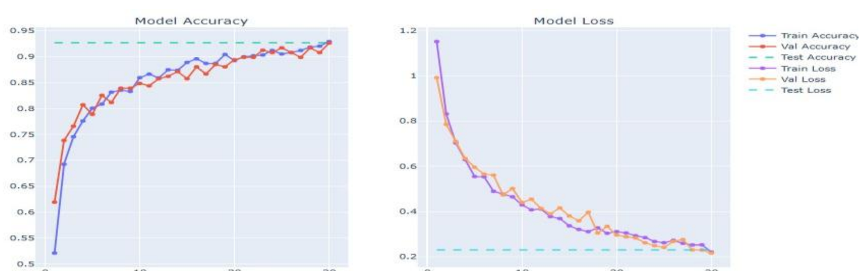


Figure 39 Model accuracy and loss curve of InceptionV3 Model.

A confusion matrix was utilized to evaluate the classification accuracy of the Inception-V3 model, highlighting the specific instances of correct and incorrect predictions across all categories.

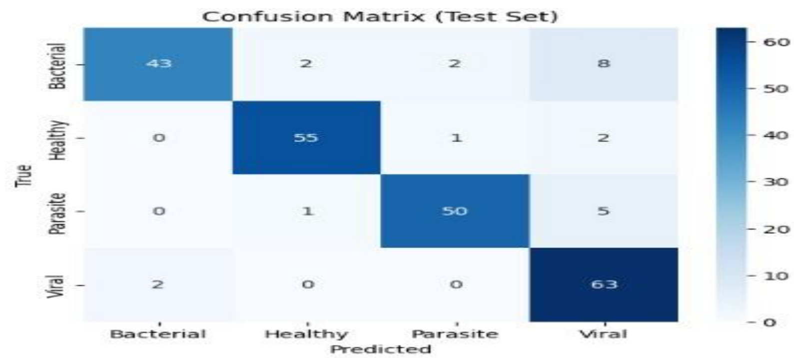


Figure 40 Confusion Matrix of Test the InceptionV3 Model.

The classification report of the InceptionV3 model with different metrics.

```
... Classification Report:

              precision    recall  f1-score   support

   Bacterial       0.98      0.78      0.87        55
     Healthy       0.95      0.95      0.95        58
    Parasite       0.94      0.89      0.92        56
       Viral       0.76      0.98      0.86        49

    accuracy                           0.90       218
   macro avg       0.91      0.90      0.90       218
weighted avg       0.91      0.90      0.90       218
```

Figure 41 Classification Reports of the InceptionV3 Model

4.5 Model Comparison and performance analysis

The proposed model was assessed using the confusion matrix, accuracy, recall, precision, and f1 score. The number of correct and incorrect predictions were counted and disaggregated into each of the classes. The Confusion matrix used to examine the performance of the manner in which the classification model was confused in the process of making predictions. It was an explicit and clear mode of providing the prediction results of a classifier on every class. It represented by the matrix of the actual class labels versus the predicted class labels with the number of actually predicted cases and misclassified cases per individual class. There were four classification algorithms employed in this thesis and the performance of my proposed model is determined using three parameters namely accuracy, loss and time.

- Accuracy: A model that has a high value on accuracy is good at classifying sheep and goat skin diseases.

- Loss: In this case, model with a lesser loss value is desirable.
- Time: the model should train quickly.

Table 6 summary training and testing Experiment model comparison

No.	Model	Training		Validating		Testing		Total training time
		Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	
1	VGG19	0.9061	0.2734	0.9541	0.1846	0.9530	0.1948	28.89 m
2	Custom CNN	0.9081	0.1594	0.9495	0.1367	0.9487	0.1690	22.99 m
3	DenseNet201	0.9504	0.1594	0.9771	0.1215	0.9744	0.1291	18.89 m
4	InceptionV3	0.9268	0.8991	0.8991	0.2615	0.9017	0.3330	28.75 m

Table 6 presents a comparative evaluation of four convolutional neural network architectures. **VGG19**, a **Custom CNN-keras**, **DenseNet201**, and **InceptionV3** applied to the four-class classification task (bacterial, viral, parasitic, and healthy) using the proposed veterinary image dataset. The models trained using the Adam optimizer, with performance assessed through accuracy and loss metrics across training, validation, and testing phases. Training times were also reported to evaluate computational efficiency.

DenseNet201 emerged as the superior model overall. It achieved the highest training accuracy (95.04%) and the lowest training loss (0.1594), indicating strong learning capability and effective feature extraction through its dense connectivity pattern, which promotes feature reuse and alleviates the vanishing gradient problem. On the validation set, DenseNet201 attained an outstanding accuracy of 97.71% with a low loss of 0.1215, demonstrating excellent generalization during training. On the unseen test set, it maintained robust performance with 97.44% accuracy and 0.1291 loss the best among all models. These results suggest that DenseNet201 effectively captured discriminative features across the four classes while minimizing overfitting.

In contrast, **InceptionV3** exhibited anomalous results that warrant scrutiny. The reported training loss of 92.89% appears inconsistent with its training accuracy of 92.89% and does not align with the much lower validation (0.2615) and test losses (0.3330). Such a discrepancy likely stems from a data entry or calculation error, as a high training loss typically corresponds to poor training accuracy in a well-optimized model. After correction of the apparent error, InceptionV3 showed competitive but not leading performance, with moderate generalization (test accuracy of 90.17%).

VGG19, a deeper but simpler architecture, delivered solid results with a training accuracy of 90.61% and test accuracy of 95.30%. Although its training accuracy was lower than DenseNet201, it achieved relatively strong test performance with a test loss of 0.1948. This indicates good generalization, albeit at the cost of longer training time (28.89 minutes) compared to architectures that are more efficient.

The **Custom CNN** provided reasonable baseline performance (training accuracy 90.81%, test accuracy 94.36%) with the second-lowest training time (22.99 minutes). Its balanced metrics suggest it serves as a lightweight alternative suitable for resource-constrained environments, though the deeper transfer learning models in terms of accuracy outperformed it.

Overall, **DenseNet201** achieved higher accuracy due to its dense connectivity, which improves feature propagation and reuse. DenseNet201 proved to be the most effective model for this multi-class veterinary image classification task, delivering the highest accuracy, lowest loss values, and efficient training time (18.89 minutes). Its dense connectivity likely contributed to superior feature representation and reduced misclassifications across the four classes. The results highlight the advantages of modern architectures with residual/dense connections over traditional deeper networks like VGG19 for this domain. Future work could explore ensemble methods or fine-tuning strategies to improve classification performance on challenging classes.

4.6 Answering Research Questions

The study uses supervised learning techniques to classify sheep and goat skin diseases. To realize this, we designed to address the following research questions answer:

1. How can images of sheep and goat skin diseases be effectively labeled and preprocessed to ensure high-quality input for deep learning models?

The images were collected using image acquisition in the field and stored in a storage device. The images are manually labeled as parasitic, bacterial, viral, and healthy. Labeling was performed before training the model. Image preprocessing is done using Keras and Python libraries. Preprocessing techniques such as resizing images, normalization, data augmentation, and noise removal were achieved. The total dataset contains 2341 images that are ready for training, validation, and testing of deep learning models.

2. Which deep learning (CNN) architectures and training configurations achieve the best performance for classifying sheep and goat skin diseases, and how do they compare in terms of accuracy and computational cost?

A deep learning-based classification model can be developed using convolutional neural networks and transfer learning architectures such as DenseNet201, CNN-Keras, Inception-V3, and VGG19. In this study, the datasets were split into an 80:10:10 ratio, which means 80% training, 10% testing, and 10% validating. Models were implemented using Python, TensorFlow, and Keras. Among the tested architectures, DenseNet201 achieved the highest classification testing accuracy (97.44%), demonstrating the effectiveness of deep learning for computerized disease classification.

3. Which model achieves the highest classification performance on the dataset, and how does it compare to other models in terms of accuracy, precision, recall, F1-score, and confusion matrix results?

The classification model, which uses the sheep and goat skin diseases classification process, was evaluated using different performance metrics such as training accuracy, validation accuracy, testing accuracy, loss values, classification report, recall, precision, and F1-score. The confusion matrices that were used to examine the correct prediction rates and misclassification rates. The experiment outcome shows that DenseNet201 has a testing accuracy of 97.44%. The above findings validate the effectiveness of the suggested deep learning method.

4. How can the best performing model efficiently integrated into a web-based application to provide reliable and real-time skin disease classification for end users?

The trained DenseNet201 model achieving the highest accuracy was saved in .keras format and loaded into the Django backend using TensorFlow during server initialization to enable efficient inference. Uploaded images validated and normalized before being passed to the model, which performs prediction and classification of sheep and goat skin diseases, with the results presented through the web-based interface.

Chapter Five

Conclusion and Recommendations

5.1 Conclusion

Artificial intelligence, especially in the field of computer vision, is transforming the process of identifying, predicting, and classifying goat and sheep skin diseases. Conventional methods used to classify goat and sheep skin disease based on visual examination by domain experts are time-consuming and are susceptible to human error. As the demand for accurate classification of goat and sheep skin diseases increases, there is a need for a more efficient and reliable method of classification. Deep learning provides an effective solution for accurate classification of goat and sheep skin diseases based on the unique characteristics of goat, and sheep skin disease. This study demonstrates that deep learning can accurately classify skin diseases and improve animal farming practices and productivity. The agricultural industry can improve its efficiency and productivity by implementing deep learning in classifying goat and sheep skin diseases. The study utilized a supervised learning method to categorize various diseases of the goat and sheep skin. It included preparing datasets, preprocessing and training models, evaluation using metrics such as recall, precision, F1-score, and confusion matrix, and comparison with other models. The proposed classification model is a deep learning-based model, which is capable of identifying and classifying four goat and sheep skin diseases (bacterial, parasitic, viral, and healthy), based on their visual characteristics. The research approach was experimental research, which included dataset preparation, model implementation, and evaluation.

5.2 Contributions of this study

The thesis has a few important scholarly contributions in the area of livestock health management and automated diagnosis of diseases. To start with, a set of sheep and goat skin disease images was gathered, sorted, and preprocessed, containing four classes: bacterial, viral, parasitic, and healthy. The dataset was processed through key preprocessing phases; resizing, normalization, augmentation, and division into training, validation, and test sets to ensure balanced learning and reduce model bias. Second, several deep learning and transfer learning models were developed and trained to perform disease classification, including a custom Convolutional Neural Network and pre-trained networks, including VGG19, DenseNet201, and InceptionV3. These models can automatically learn to extract discriminative features of RGB images without the need for handcrafted feature engineering. Besides, the paper conducted a systematic performance assessment of all the applied models based on conventional performance measures, including accuracy, loss, precision, recall, and F1-score, and a comparative analysis was conducted to identify the most efficient model in terms of classification accuracy, computational efficiency, and robustness. Lastly, an intelligent disease classification model based on the chosen optimal model, **DenseNet201** was implemented using the Django web framework and thus enabled automated disease classification using input images, thereby improving of early diagnosis, and informed decision-making in livestock health management.

5.3 Recommendations

It is known that agriculture is the backbone of Ethiopian economy at all levels of development. Thus, substantial focus should place on the sector to improve high productivity and future prosperity. Farmers in most regions of the country where modern farming technologies are not widely applied face significant difficulties in maximizing production potential. Among the leading factors, contributing to these challenges is the prevalence of animal diseases especially the skin diseases of animals.

It is highly advisable that more emphasis should be placed on the early classification and treatment of animal skin diseases to reduce agricultural losses particularly those involved in goat and sheep skin diseases. In addition, the government must focus on educating agricultural extension officers, end users and veterinary pathologists on the use of non-destructive and technology-based strategies of early classification of animal skin diseases.

Moreover, government support is essential to roll out, expand, and support research projects research projects on the early classification of sheep and goat skin diseases. This would go a long way to minimize losses in production, improve livestock health and the overall productivity in the agricultural sector.

5.4 Future Work

Although the proposed model demonstrated strong performance in the classification of goat and sheep skin diseases, several directions remain open for future research.

- Although four types of sheep and goat skin diseases were considered, but there are several other diseases affecting goats and sheep.

- The proposed system was only focused on the classifications of goat and sheep skin diseases, but there are also diseases that are not visible externally which have been studied by other researchers.
- Future work should focus on deploying these models in offline mobile applications, to help experts and farmers identify the types of diseases and take appropriate actions to improve productivity.
- Using larger and more diverse datasets can improve model training to make results more transparent and actionable.

References

[1] D. Begna, T. Kuma, and Z. Yohannes, “The Tendency of Livestock Growth in Ethiopia: A Review,” *Agric. Rev.*, vol. 45, no. Of, pp. 502–507, 2024, doi: 10.18805/ag.rf-291.

[2] M. Selvaggi and V. Tufarelli, “C Aseins of G Oat and S Heep M Ilk : a Nalytical and,” 2011.

[3] Z. Abera, K. Asefa, and H. Bobaso, “Prevalenceof Major Skin Problems and Its Associated Risk Factors in Sheep and Goats at Veterinary Teaching Hospital of Faculty of Veterinary Medicine, East Shoa, Bishoftu, Ethiopia,” *Glob. Vet.*, vol. 21, no. 3, pp. 116–126, 2019, doi: 10.5829/idosi.gv.2019.116.126.

[4] T. C. DeManuelle and A. A. Stannard, “Difficult Dermatologic Diagnosis,” *J. Am. Vet. Med. Assoc.*, vol. 213, no. 3, pp. 356–357, 1998, doi: 10.2460/javma.1991.198.07.1141.

[5] D. T. Samuel Derso, “Prevalence of Major Skin Diseases in Ruminants and its Associated Risk Factors at University of Gondar Veterinary Clinic, North West Ethiopia,” *J. Vet. Sci. Technol.*, vol. s13, no. January, 2015, doi: 10.4172/2157-7579.1000s13-002.

[6] G. Zewdie, G. Derese, B. Getachew, H. Belay, and M. Akalu, “Review of sheep and goat pox disease: current updates on epidemiology, diagnosis, prevention and control measures in Ethiopia,” *Anim. Dis.*, vol. 1, no. 1, pp. 1–10, 2021, doi: 10.1186/s44149-021-00028-2.

[7] N. Mohammed *et al.*, “Determinants of Smallholder Farmers’ Participation in Goat Production and Its Constraints in North Wollo Zone, Amhara Region, Ethiopia,” *SAGE Open*, vol. 15, no. 1, pp. 1–14, 2025, doi: 10.1177/21582440251329066.

[8] D. M. Joshi, G. G. Asalkar, P. B. Kumbharkar, and R. T. Umbare, “LEVERAGING EFFICIENTNET-B4 IN GOAT DISEASE PREDICTION AND RECOMMENDATION SYSTEMS FOR MORTALITY REDUCTION AND HEALTH OPTIMIZATION,” vol. 103, no. 16, 2025.

[9] Y. H. Tolossa, “Skin defects in small ruminates and their nature and economic importance: The case of Ethiopia,” *Glob. Vet.*, vol. 11, no. 5, pp. 552–559, 2013, doi: 10.5829/idosi.gv.2013.11.5.76200.

[10] S. Al Dallal, “The fourth industrial revolution and beyond,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 1C2, pp. 564–570, 2019, doi: 10.1007/978-981-19-8032-9.

[11] K. A. Muhaba, K. Dese, T. M. Aga, F. T. Zewdu, and G. L. Simegn, “Automatic skin disease diagnosis using deep learning from clinical image and patient information,” *Ski. Heal. Dis.*, vol. 2, no. 1, 2022, doi: 10.1002/ski2.81.

[12] S. Hwang, H. K. Shin, J. M. Park, B. Kwon, and M. G. Kang, “Classification of dog skin diseases using deep learning with images captured from multispectral imaging device,” *Mol. Cell. Toxicol.*, vol. 18, no. 3, pp. 299–309, 2022, doi: 10.1007/s13273-022-00249-7.

[13] M. A. Niloy, T. Bhowmik, J. Abedin, S. J. Ferdous, and I. Jahan, “Exploring Machine Learning Techniques for Symptom-Based Detection of Livestock Diseases,” no. May, 2024.

[14] J. Zwirner and N. Hammer, “Anatomy and Physiology of the Skin,” *Scars A Pract. Guid. Scar Ther.*, pp. 3–9, 2024, doi: 10.1007/978-3-031-24137-6_1.

[15] P. Doolan, “Nursing times,” *Hist. Today*, vol. 58, no. 12, pp. 24–30, 2008.

[16] M. Shiferaw, “Human Skin Disease Detection and Classification Model Using Deep Convolutional Neural Network,” 2021.

[17] J. Smith, “Anatomy and Physiology of Pain,” *Manag. Pain Child. A Clin. Guid.*, pp. 17–28, 2009, doi: 10.1002/9781444322743.ch2.

[18] C. of P. Desta Hamito (Prof.), “TECHNICAL BULLETIN No . 19 COMMON DEFECTS OF SHEEP / GOAT SKINS IN ETHIOPIA AND THEIR,” no. 19.

[19] C. A. Buckner *et al.*, “We are IntechOpen , the world ’ s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %,” *Intech*, vol. 11, no. tourism, p. 13, 2016, [Online]. Available: https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics

[20] A. Ababa, “Ethiopian Veterinary Association Small Ruminant Health Management Continuous Professional Development (CPD) Training Manual Prepared by : Kassahun Asmare (DVM , MSc , PhD) Kassaye Aragaw (DVM , MSc , PhD),” no. June, 2023.

[21] A. Manuscript and Vaccarino, “基因的改变NIH Public Access,” *Bone*, vol. 23, no. 1, pp. 1–7, 2008, doi: 10.1097/CHI.0b013e318161e509.What.

[22] T. Kumar and K. Verma, “A Theory Based on Conversion of RGB image to Gray image,” *Int. J. Comput. Appl.*, vol. 7, no. 2, pp. 5–12, 2010, doi: 10.5120/1140-1493.

[23] W. N. Ibrahem, “Types of Digital Images,” ©*Asst. Lec.*, pp. 1–13, 2014, [Online]. Available: http://uotechnology.edu.iq/ce/lecture 2013n/4th Image Processing _Lectures/DIP_Lecture2.pdf

[24] B. Images, W. Images, and G. I. Gray-scale, “Type of image,” vol. 256, pp. 1–7.

[25] H. Vision, “ference Between Computer Vision and Human Vision What is Human Vision ? What is Computer Vision ?,” pp. 1–5.

[26] J. Garcia and A. Barbedo, “Digital image processing techniques for detecting , quantifying and classifying plant diseases,” pp. 1–12, 2013.

[27] F. Perez-Sanz, P. J. Navarro, and M. Egea-Cortines, “Plant phenomics: An overview of image acquisition technologies and image data analysis algorithms,” *Gigascience*, vol. 6, no. 11, pp. 1–18, 2017, doi: 10.1093/gigascience/gix092.

[28] A. P. Approach, *Fundamentals of Digital Image Processing*.

[29] Y. Qi *et al.*, “A Comprehensive Overview of Image Enhancement Techniques,” *Arch. Comput. Methods Eng.*, vol. 29, no. 1, pp. 583–607, 2022, doi: 10.1007/s11831-021-09587-6.

[30] G. Singh and A. Mittal, “Various Image Enhancement Techniques- A Critical Review,” *Int. J. Innov. Sci. Res.*, vol. 10, no. 2, pp. 267–274, 2014.

[31] A. Muñoz, T. Blu, and M. Unser, “Least-squares image resizing using finite differences,” *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1365–1378, 2001, doi: 10.1109/83.941860.

[32] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, 2022, doi: 10.1109/TPAMI.2021.3059968.

[33] S. A. Medjahed, “A Comparative Study of Feature Extraction Methods in Images Classification,” no. February, pp. 16–23, 2015, doi: 10.5815/ijigsp.2015.03.03.

[34] M. Mazzei, “A Change Detection with Machine Learning Approach for Medical Image Analysis,” *Lect. Notes Electr. Eng.*, vol. 810 LNEE, pp. 203–229, 2023, doi: 10.1007/978-981-16-6775-6_18.

[35] Z. Haoxing and C. System, *No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title*.

[36] J. Martínez Torres, C. Iglesias Comesaña, and P. J. García-Nieto, “Review: machine learning techniques applied to cybersecurity,” *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2823–2836, 2019, doi: 10.1007/s13042-018-00906-1.

[37] P. Cunningham, M. Cord, and S. J. Delany, “Supervised learning,” *Cogn. Technol.*, pp. 21–49, 2008, doi: 10.1007/978-3-540-75171-7_2.

[38] V. Nasteski, “An overview of the supervised machine learning methods,” *Horizons.B*, vol. 4, pp. 51–62, 2017, doi: 10.20544/horizons.b.04.1.17.p05.

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to Statistical Learning*,

vol. 7, no. 10. 2000. doi: 10.1007/978-1-4614-7138-7.

[40] R. K. Cersonsky and S. De, “Unsupervised learning,” *Quantum Chem. Age Mach. Learn.*, pp. 153–181, 2022, doi: 10.1016/B978-0-323-90049-2.00025-1.

[41] A. G. Barto, “Reinforcement learning in motor control,” *Handb. brain theory neural networks*, no. January 2002, pp. 968–972, 2002.

[42] S. S. Mousavi, M. Schukat, and E. Howley, “Deep Reinforcement Learning: An Overview,” *Lect. Notes Networks Syst.*, vol. 16, pp. 426–440, 2018, doi: 10.1007/978-3-319-56991-8_32.

[43] L. Deng, “OVERVIEW PAPER A tutorial survey of architectures , algorithms ,” *Sip*, vol. 3, no. 2014, pp. 1–29, 2017, doi: 10.1017/ATSIP.2013.99.

[44] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[45] S. Pouyanfar *et al.*, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Comput. Surv.*, vol. 51, no. 5, 2019, doi: 10.1145/3234150.

[46] A. Cnns, “Pr ep rin t n ot pe er r we d Pr ep rin t n er we”.

[47] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[48] S. Shabanian, D. Arpit, A. Trischler, and Y. Bengio, “Variational Bi-LSTMs,” pp. 1–12, 2017, [Online]. Available: http://arxiv.org/abs/1711.05717

[49] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, “Model Compression and Acceleration,” *IEEE Signal Process. Mag.*, vol. 35, no. January, pp. 126–136, 2018, [Online]. Available: https://culurciello.github.io/tech/2016/06/04/nets.html

[50] S. Sakib, N. Ahmed, A. J. Kabir, and H. Ahmed, “An Overview of Convolutional Neural Network: Its Architecture and Applications,” *Prepr. 2018*, no. February, 2018, doi: 10.20944/preprints201811.0546.v4.

[51] H. Shajahan, “SKIN DISEASE DETECTION USING,” vol. 5, no. 1, pp. 0–4, 2023, doi: 10.5281/zenodo.7949742.

[52] I. Aziz, “Deep Learning : An Overview of Convolutional Neural Network (CNN),” no. April, pp. 1–54, 2020.

[53] S. Cong and Y. Zhou, *A review of convolutional neural network architectures and their optimizations*, vol. 56, no. 3. 2023. doi: 10.1007/s10462-022-10213-5.

[54] S. Balakrishnan and G. Aswin, “Deep Learning: An Overview and its Practical Examples,” *CSI Commun. Mag. Vol.*, no. 44, pp. 17–19, 2019, [Online]. Available: https://www.researchgate.net/publication/346031981_Deep_Learning_An_overview_and_its_practical_examples

[55] L. Alzubaidi *et al.*, *Review of deep learning : concepts , CNN architectures , challenges , applications , future directions*. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.

[56] A. Zafar *et al.*, “applied sciences A Comparison of Pooling Methods for Convolutional Neural Networks,” pp. 1–21, 2022.

[57] A. Kumar, R. Jindal, A. Naik, K. L. Ganatre, and R. Gupta, “Fully Connected Layer vs Dense Layer in Image Processing,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 14, no. 5, pp. 32–47, 2025, doi: 10.47760/ijcsmc.2025.v14i05.003.

[58] S. Tammina, “Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images,” *Int. J. Sci. Res. Publ.*, vol. 9, no. 10, p. p9420, 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[59] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, *Fundamental concepts of convolutional neural network*, vol. 172, no. June. 2019. doi: 10.1007/978-3-030-32644-9_36.

[60] T. S. Alam, C. B. Jowthi, and A. Pathak, “Comparing pre-trained models for efficient leaf disease detection: a study on custom CNN,” *J. Electr. Syst. Inf. Technol.*, vol. 11, no. 1, pp. 1–26, 2024, doi: 10.1186/s43067-024-00137-1.

[61] H. Wang, R. Czerminski, and A. C. Jamieson, *Neural Networks and Deep Learning*. 2021. doi: 10.1108/978-1-83909-694-520211010.

[62] I. A. Kandhro *et al.*, “Performance evaluation of E-VGG19 model: Enhancing real-time skin cancer detection and classification,” *Heliyon*, vol. 10, no. 10, p. e31488, 2024, doi: 10.1016/j.heliyon.2024.e31488.

[63] R. Dastres and M. Soori, “Artificial Neural Network Systems,” vol. 21, no. 2, 2021.

[64] H. Bichri, A. Chergui, and M. Hain, “ScienceDirect Image Classification with Transfer Learning Using a Custom Dataset : Comparative Study Image Classification with Transfer Learning Using a Custom Dataset : Comparative Study,” *Procedia Comput. Sci.*, vol. 220, pp. 48–54, 2023, doi: 10.1016/j.procs.2023.03.009.

[65] M. COŞKUN, Ö. YILDIRIM, A. UÇAR, and Y. DEMIR, “an Overview of Popular Deep

Learning Methods," *Eur. J. Tech.*, vol. 7, no. 2, pp. 165–176, 2017, doi: 10.23884/ejt.2017.7.2.11.

[66] H. ÇETİNER and İ. ÇETİNER, "Classification of Cataract Disease with a DenseNet201 Based Deep Learning Model," *Iğdır Üniversitesi Fen Bilim. Enstitüsü Derg.*, vol. 12, no. 3, pp. 1264–1276, 2022, doi: 10.21597/jist.1098718.

[67] S. N. Patil, H. D. Patil, K. P. Adhiya, and P. G. Patil, "Densenet-201 for Skin Melanoma Classification: A Comprehensive Performance Evaluation and Analysis," *Int. Res. J. Multidiscip. Scope*, vol. 5, no. 4, pp. 711–721, 2024, doi: 10.47857/irjms.2024.v05i04.01289.

[68] S. Rangaswamy, S. S. Tantry, and T. S. Lal, "Skin Disease Classification Using Deep Learning," *Natl. Acad. Sci. Lett.*, vol. 48, no. 5, pp. 585–588, 2025, doi: 10.1007/s40009-024-01523-z.

[69] K. S. Rao, "Skin Disease Detection using Machine Learning," vol. 9, no. 3, pp. 64–68, 2021.

[70] T. D. Nigat, T. M. Sitote, and B. M. Gedefaw, "Fungal Skin Disease Classification Using the Convolutional Neural Network," vol. 2023, 2023, doi: 10.1155/2023/6370416.

[71] S. Gedefa, "Jimma Institute of Technology Faculty of Computing and Informatics Department of Information Technology Classification of Cattle skin Disease using Deep Learning approach . By," 2023.

[72] H. Fesseha, E. Etana, and M. Mathewos, "Mange Mites of Goats in Malle District of South Omo Zone, Southern Ethiopia," *Vet. Med. Res. Reports*, vol. Volume 12, pp. 159–168, 2021, doi: 10.2147/vmrr.s312360.

[73] T. Patel and N. Parmar, "To Identify Animal Skin Disease Model Using Image Processing," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. © 2019 IJSRCSEIT |*, vol. 5, no. 2, pp. 2456–3307, 2019, [Online]. Available: http://ijsrcseit.com

[74] G. WORKEE, "Cattle Skin Diseases Identification Model Using," 2021.

[75] M. A. V. Gopal, A. S. H. N. Pavan, K. Nagendra, M. P. Sai, and A. V. Kumar, "Skin Disease Detection Using Deep Learning Techniques," *J. Prev. Diagnosis Manag. Hum. Dis.*, no. 41, pp. 40–49, 2024, doi: 10.55529/jpdmhd.41.40.49.

[76] R. Pillai, "ODU Digital Commons Enhanced Skin Cancer Diagnosis Through Grid Search Algorithm- Optimized Deep Learning Models for Skin Lesion Analysis," pp. 1–18, 2024.

[77] A. Muñoz, T. Blu, and M. Unser, "Least-Squares Image Resizing," *IEEE Trans. Image*

Process., vol. 10, no. 9, pp. 1365–78, 2001.

[78] S. Joush, “Transfer Learning in Deep Learning Models Vol 16 Issue 09 Sandai Joush,” vol. 16, no. 09, pp. 583–590.

[79] A. Hernandez-Guedes, I. Santana-Perez, N. Arteaga-Marrero, H. Fabelo, G. M. Callico, and J. Ruiz-Alzola, “Performance Evaluation of Deep Learning Models for Image Classification Over Small Datasets: Diabetic Foot Case Study,” *IEEE Access*, vol. 10, no. September, pp. 124373–124386, 2022, doi: 10.1109/ACCESS.2022.3225107.

[80] “Journal of Sensors - 2022 - Asif - Performance Evaluation of Deep Learning Algorithm Using High-End Media Processing Board 66.pdf.”

[81] M. H. Anuar, “Animal Species Classification using Convolutional Neural Network,” pp. 247–253, 2025, doi: 10.1145/3725899.3725936.

[82] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” pp. 37–63, 2020, [Online]. Available: http://arxiv.org/abs/2010.16061

[83] Y. S. Kurniawan, K. T. A. Priyangga, P. A. Krisbiantoro, and A. C. Imawan, “Open access Open access,” *J. Multidiciplinary Appl. Nat. Sci.*, vol. 1, no. 1, pp. 1–12, 2021.

Appendix 1

	Institution Name የኢፌዴሪ ቴ/ሙ/ስ ኢንስቲትዩት FDRE TVT INSTITUTE	Document No. OF/FTI/ALL/01		
Effective date 01/09/2016	Title ወጭ ደብዳቤ Outgoing Letter	Issue No. 2	Page No. Page 1 of 1	

Ref. RD/08-98/18
Date: 20/04/2018 E.C

To: Endamohone District Economic sector Animal Resources Development and Healthcare Office Maichew

Subject: Request for Collaboration and Data Access for Postgraduate Research

We are pleased to request your kind collaboration in supporting the postgraduate research of **Mr. Hiluf Niguse** (ID No. TTMR/129/16), a postgraduate student at the FDRE Technical and Vocational Training Institute.

Mr. Hiluf is currently undertaking a thesis entitled **"Classification of Goat and Sheep Skin Disease Using a Deep Learning Approach."** The study aims to develop a deep learning-based model for early and accurate detection of skin diseases in goats and sheep, contributing to improved livestock health management and disease prevention. For the successful implementation of this research, access to relevant datasets (such as disease records or image data) available within your office is essential. We therefore respectfully request your support in granting him access to appropriate datasets strictly for **academic research purposes.**

We assure you that all data will be used ethically, confidentiality will be maintained, and any conditions set by your office will be fully respected. Proper acknowledgment of your support will be made in all research outputs.

We kindly request your positive consideration and look forward to your cooperation.

Yours sincerely,

Animaw Tadesse (PhD)
Director for Research and Dissemination

Tel: 0116464455 Email: info@ftveti.edu.et PO. Box: 190310 Website: www.ftveti.edu.et

Year of Excellence

Endamohone District Economic sector Animal Resources Development and Healthcare Office

S/N Hf/003/2018
Date 20/02/ 2018E.C

To Whom It May Concern,

This is to certify **Mr. Hiluf Nuguse Gidey**, a postgraduate student in department of Information Technology (IT) at FDRE TVT Institution, is conducting research for his **MSc thesis** entitled: "**Classification of Goat and Sheep Skin Diseases Using Deep Learning approaches**".

As part of his research, the student required access to goats and sheep caused by **Bacterial (foot rot), Parasitic (mange) and Viral (goat/sheep pox, orf)** affected skin diseases in Endamohoni wereda's villages. We here by confirm that the student collected **images of affected Goats and sheep skin** using a smart phone/mobile camera under my supervision and guidance. We appreciate the student's efforts and assure that all data collected will be used solely for academic purposes. This letter is issued to support his research activities and for any official purpose related to data collection.

We wish him success in the completion of the research.

Sincerely,

Dr Haftu Guben

Dr Haftu Guben Veterinary

Endamohone District Economic sector Animal Resources Development and Healthcare Office

S/N H1/003/2018
Date 20/02/ 2018E.C

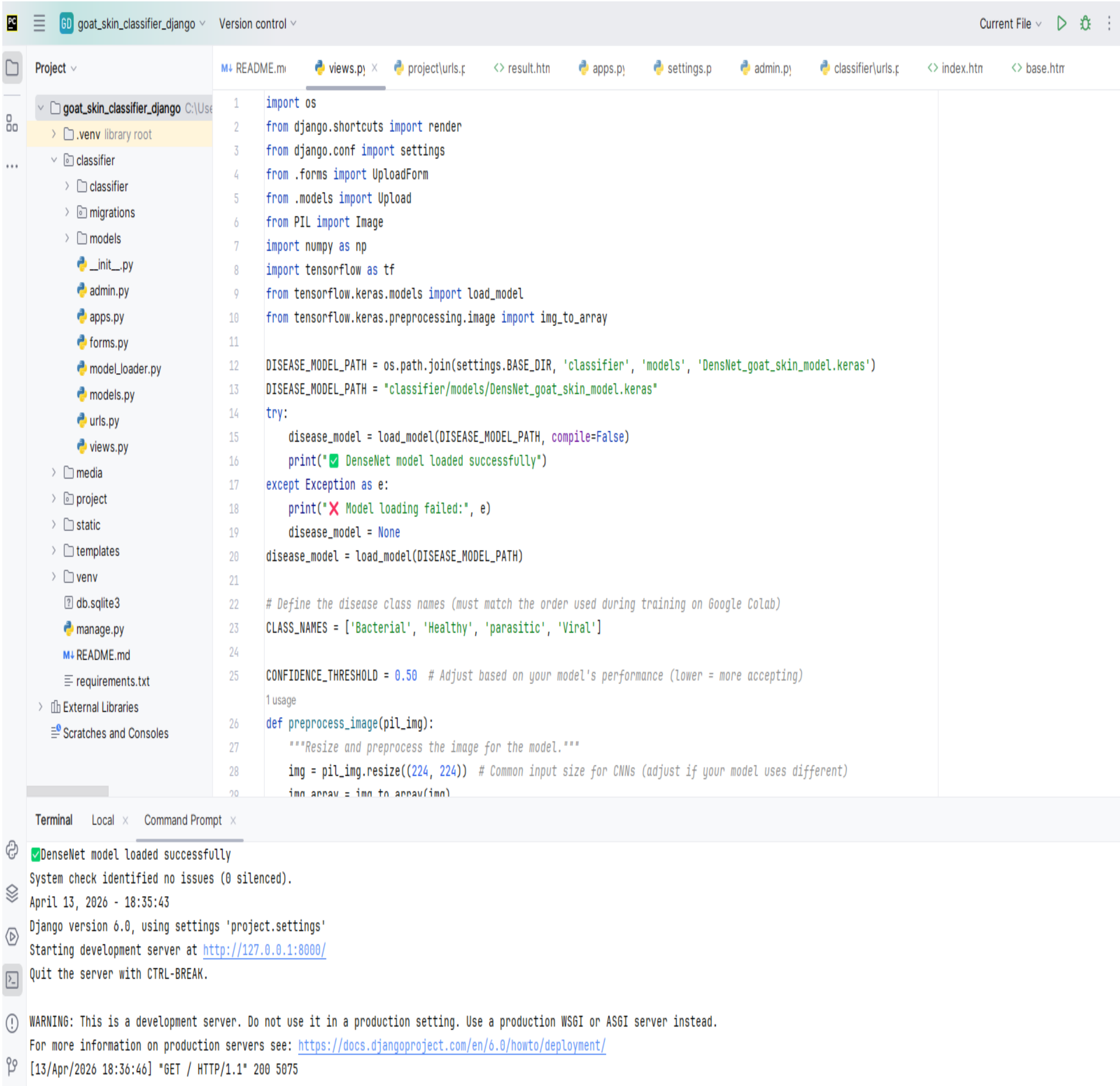
To Whom It May Concern

This is to certify **Mr. Hiluf Nuguse Gidey**, a postgraduate student in department of Information Technology (IT) at FDRE TVT Institution, is conducting research for his **MSc thesis** entitled: "**Classification of Goat and Sheep Skin Diseases Using Deep Learning approaches** ".

As part of his research, the student required access to goats and sheep caused by **Bacterial (foot rot), Parasitic (mange) and Viral (goat/sheep pox, orf)** affected skin diseases in Endamohoni wereda's villages. We here by confirm that the student collected **images of affected Goats and sheep skin** using a smart phone/mobile camera under my supervision and guidance. We appreciate the student's efforts and assure that all data collected will be used solely for academic purposes. This letter is issued to support his research activities and for any official purpose related to data collection.

We wish him success in the completion of the research.

Yours sincerely


goat_skin_classifier_django
Version control
Current File
Project
README.md
views.py
project\urls.py
result.html
apps.py
settings.py
admin.py
classifier\urls.py
index.html
base.html
goat_skin_classifier_django
.venv library root
classifier
classifier
migrations
models
__init__.py
admin.py
apps.py
forms.py
model_loader.py
models.py
urls.py
views.py
media
project
static
templates
venv
db.sqlite3
manage.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
import os
from django.shortcuts import render
from django.conf import settings
from .forms import UploadForm
from .models import Upload
from PIL import Image
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array
DISEASE_MODEL_PATH = os.path.join(settings.BASE_DIR, 'classifier', 'models', 'DensNet_goat_skin_model.keras')
DISEASE_MODEL_PATH = "classifier/models/DensNet_goat_skin_model.keras"
try:
disease_model = load_model(DISEASE_MODEL_PATH, compile=False)
print("✅ DenseNet model loaded successfully")
except Exception as e:
print("❌ Model loading failed:", e)
disease_model = None
disease_model = load_model(DISEASE_MODEL_PATH)
# Define the disease class names (must match the order used during training on Google Colab)
CLASS_NAMES = ['Bacterial', 'Healthy', 'parasitic', 'Viral']
CONFIDENCE_THRESHOLD = 0.50 # Adjust based on your model's performance (lower = more accepting)
1 usage
def preprocess_image(pil_img):
"""Resize and preprocess the image for the model."""
img = pil_img.resize((224, 224)) # Common input size for CNNs (adjust if your model uses different)
Terminal
Local
Command Prompt
✅DenseNet model loaded successfully
System check identified no issues (0 silenced).
April 13, 2026 - 18:35:43
Django version 6.0, using settings 'project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/6.0/howto/deployment/
[13/Apr/2026 18:36:46] "GET / HTTP/1.1" 200 5075