CATTLE FARMS FOOT-AND-MOUTH DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Madan Beshir Woya

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June, 2024

Adama, Ethiopia

CATTLE FARMS FOOT-AND-MOUTH DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Madan Beshir Woya

Advisor: Dr. Bahiru Shifaw

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June, 2024

Adama, Ethiopia

APPROVAL OF BOARD OF EXAMINER

I, the advisors of the thesis entitled "**CATTLE FARMS FOOT-AND-MOUTH DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK**" and developed by Madan Beshir Woya hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

___________________	_________________________	___________________
Major Advisor/Supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis Madan Beshir Woya have read and evaluated the thesis entitled "**CATTLE FARMS FOOT-AND-MOUTH DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of degree of Master's of Science in Computer Science and Engineering.

_________________________	___________________	___________________
Chairperson	Signature	Date
_________________________	___________________	___________________
Internal Examiner	Signature	Date
_________________________	___________________	___________________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_________________________	___________________	___________________
Department Head	Signature	Date
_________________________	___________________	___________________
School Dean	Signature	Date
_________________________	___________________	___________________
Office of Postgraduate Studies, Dean	Signature	Date

DECLARATION

I hereby declare that this Master Thesis entitled "**CATTLE FARMS FOOT-AND-MOUTH DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Madan Beshir Woya	___________________	_________________
Name of Student	Signature	Date

RECOMMENDATION

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**CATTLE FARMS FOOT-AND-MOUTH DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK**" prepared under my/our guidance by Madan Beshir Woya submitted in partial fulfillment of the requirements for the Degree of Master's in Computer Science and Engineering. Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

Dr. Bahiru Shifaw	____________________	____________________
Major Advisor/Supervisor	Signature	Date

ACKNOWLEDGMENT

First and foremost, I would like to express my gratitude to God for granting me the strength and perseverance throughout my study and to complete this thesis.

I would also like to extend my heartfelt appreciation to Dr. Bahiru Shifaw for his exceptional mentorship, unwavering support, and guidance. His expertise and encouragement have been instrumental in shaping this research project, and I am truly grateful for his dedication to my academic growth.

I would as well like to express my sincere gratitude to the Ministry of Agriculture and College of Veterinary, Bishoftu for their unwavering support, care, and exceptional treatment during my visit to request data. Their willingness to assist me and provide valuable information has been invaluable in the successful completion of my project. I would like to specifically thank Dr. Melkamu Tadesse and Dr. Gashaw Beyene from Disease Prevention and Control Directorate thankful for their generosity and professionalism, which has greatly contributed to the quality of my research.

Furthermore, I would like to thank my parents and all my family members for their continuous support, understanding, and encouragement. Their love and belief in me have been a source of inspiration and motivation during challenging times. My special thanks goes to my friends for their support.

Thank you all to those who contributed directly and indirectly to making this thesis possible. Their insights, feedback, and assistance have been helpful in shaping this research work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SAMPLE CODES

LIST OF ACRONYMS AND ABBREVIATIONS

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area under the ROC Curve
CFT	Complement Fixation Test
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	comma-separated values
DL	Deep Learning
DNA	Deoxyribonucleic acid
EDA	Exploratory Data Analysis
ELIZA	Enzyme Linked Immunosorbent Assay
FAO	Food and Agriculture Organization
FMD	Foot-and-Mouth Disease
FMDV	Foot-and-Mouth Disease Virus
FN	False Negative
FP	False Positive
GB	Giga Byte
GHz	Giga Hertz
GPU	Graphics Processing Unit
HD	Hard Drive
HDD	Hard Disk drive
IEEE	Institute of Electrical and Electronics Engineers

ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LR	Learning Rate
MoA	Ministry of Agriculture
ML	Machine Learning
PC	Personal Computer
PCR	Polymerase Chain Reaction
PLF	Precision Livestock Farming
RBC	Red Blood Cells
ReLu	Rectified Linear Unit
RNA	Ribonucleic acid
ROC	Receiver Operating Characteristic
RT-PCR	Reverse Transcription Polymerase Chain Reaction
SNNP	Southern Nations, Nationalities, and Peoples' Region
TN	True Negative
TP	True Positive
UML	Unified Modeling Language
USD	United States Dollar
VGG	Visual Geometry Group

LIST OF NOTATIONS AND SYMBOLS

Notation	Meaning
Y	Indicates image domain
$\sum\cdot$	Indicates Summation
exp	Indicates exponential
$\approx$	Indicates approximation
N	Indicates total number of training inputs
a	Indicates actual output
w	Indicates weight of neural network
b	Indicates bias of neural network
η	Indicates Learning Rate
δ	Indicates back propagation of network

ABSTRACT

Foot and Mouth Disease (FMD) is a highly contagious and economically significant livestock disease, particularly in countries like Ethiopia where agriculture and livestock farming play a crucial role in the economy. From 2007 to 2021, the predicted pooled prevalence of FMD in cattle in Ethiopia was 21.39% (16.53-26.56%). This suggests that FMD affects more than 1 in 5 cattle in Ethiopia. Ethiopian cattle farms face several challenges in detecting and controlling FMD outbreaks. Limited resources, including trained personnel and diagnostic facilities, make it difficult to identify and respond to outbreaks in a timely manner. This research aims to bridge this gap by developing a Convolutional Neural Network (CNN) system specifically tailored to the detection of FMD in cattle farms. The dataset used for training included images of both FMD infected and healthy cattle. CNN architecture models such as VGG16, Inception V3, and Densenet 201 were utilized, and their accuracies were compared. The research achieved impressive results, with Densenet 201 exhibiting the highest performance with training, validation and testing accuracy of 99.55%, 99.00% and 98.87% respectively. This study has confirmed that Densenet 201 has shown the most significant improvement over other models in terms of evaluation metrics and performance.

Keywords: Foot and Mouth Disease, Convolutional Neural Network, Deep Learning models, CNN architectures

CHAPTER ONE

INTRODUCTION

1.1. BACKGROUND OF THE STUDY

Foot and mouth disease (FMD) is a serious, highly infectious viral illness of animals with major economic consequences. The illness infects cattle, swine, sheep, goats, and other cloven-hoofed ruminants. It is a transboundary animal disease (TAD) that has a significant impact on livestock output while also affecting regional and worldwide traffic in animals and their products (WOAH, 2024).

Ethiopia, one of Africa's prominent livestock rearing regions, faces a significant challenge in combating FMD due to its considerable impact on the country's economy and food security. Additionally the conventional disease detection approaches frequently struggle to precisely detect outbreaks of infection resulting in prolonged distress for the animals and financial losses, for the farmers. FMD was mentioned as one of the diseases that have been a concern for livestock in recent years (Abdela, 2017).

In order to contain an animal illness, livestock owners must incur financial costs. Direct losses and indirect losses are the two categories into which these consequences are divided. While direct losses are linked to death and abortion, indirect losses are connected to the price of preventative measures such as vaccinations and animal isolation fees. According to FAO data, the reemergence of previously eradicated FMD results in annual losses of around USD 1.5 billion, with the projected annual financial burden in endemic countries being USD 6.5 billion. Thus, decreasing FMD in endemic nations through a global and regional coordinated control approach is in everyone's best interests and should be viewed as a global public good (Abdullah, et al., 2020).

A 15-year Global Control Strategy was created in 2012 by the Food and Agriculture Organization (FAO) and the World Organization for Animal Health with the dual goals of maintaining the status of FMD-free countries while reducing the disease in endemic areas. It was acknowledged, therefore, that although the endemic might not be eliminated in 15 years, there would be a major improvement in the burden that FMD causes. According to FAO, Foot and Mouth has been highly prevalent in numerous countries in Africa, the Middle East, Asia, South America, Europe, North and Central America, and the Pacific (The Global foot-and-mouth disease Control Strategy, n.d.).

In the past, professionals have used their experience to identify and diagnose diseases by observing the symptoms and doing visual inspections. However, as observed skilled pathologists and agronomists frequently fail to correctly diagnose the particular illness, leading to incorrect conclusions. (Mohanty, Hughes, & Salathé, 2016) noted that an automated computational system provides agronomists with important support in the identification of diseases.

CNNs have shown remarkable success in various image recognition tasks, including medical image analysis. These deep learning models can learn hierarchical representations of images, enabling them to discern patterns and features that are indicative of disease presence. By training CNNs on large datasets of labeled images, they can become proficient at distinguishing between healthy and diseased samples with high accuracy. Deep learning is a subset of machine learning that uses networks to learn from unstructured or unlabeled data, similar to how the brain functions. Deep learning is also described as a machine learning application that trains a model using complicated algorithms and deep neural networks (IBM, 2024).

Pretrained CNN models, such as VGG16, Inception V3, and DenseNet201, have been trained on vast image datasets for general object recognition tasks. These models have learned rich representations of features from images and have achieved state-of-the-art performance on benchmark datasets. Leveraging pretrained models for specific tasks like disease detection offers several advantages, including faster convergence during training and potentially better generalization to new data. In the context of cattle foot and mouth disease detection, utilizing pretrained CNN models presents an opportunity to automate and enhance the accuracy of disease diagnosis. By feeding cattle foot and mouth disease images into these models, researchers can investigate which pretrained architecture performs best in terms of accuracy, computational efficiency, and robustness to variations in image quality and disease severity.

This study has focused on the application of convolutional neural network, in the early diagnosis of FMD. This technology will lower the social and economic expenses connected with FMD sickness. Smith, (2019) said that artificial intelligence would enable more rapid applications, such as enhancing the accuracy of data regarding what is happening on farms by improving what is discovered and approximated. There is a lot of interest in the application of prediction models for infectious illnesses, since prediction models for livestock disease outbreaks using conventional statistical and mathematical approaches have been widely shown. To this purpose, the use of machine learning (ML) technology to forecast and regulate the spread of FMD is suggested (Smith, 2019).

1.2. MOTIVATION OF THE STUDY

The motivation for studying the detection of Foot-and-Mouth Disease (FMD) using convolutional neural network architectures is driven by the significant impact of FMD on the livestock industry and the economy. Since 2007, FMD outbreaks have occurred annually, affecting the whole nation, including the Amhara and Tigray regions. Regular outbreaks have also been recorded from the Oromia region (Borena pastoral area), where the major route for exporting live animals is located (Tesfaye, et al., 2023). Furthermore, the study aims to contribute to a relatively unexplored area within the field of data science. As of now, there is limited research and development focused on applying deep learning techniques to detect FMD in cattle, specifically within researchers of Ethiopia. By addressing this gap, researchers have the potential to pioneer innovative solutions that can significantly impact disease management and livestock health in the region.

1.3. STATEMENT OF THE PROBLEM

Detecting Foot and Mouth Disease (FMD) in cattle is vital for ensuring animal health and preventing economic losses in the livestock industry. Despite the potential of Convolutional Neural Networks (CNNs) in automating FMD detection, research in this domain remains limited. Previous studies have predominantly relied on classical machine learning techniques and datasets that primarily focus on facial features, overlooking valuable information from other regions of the cattle anatomy. This thesis aims to address these gaps by investigating the effectiveness of pre trained CNN models for FMD detection in cattle. Specifically, the study compares various pre trained CNN architectures to identify the model that not only achieves better accuracy but also requires less computational time. By doing so, this research not only advances the field of automated FMD detection but also provides valuable insights into the underexplored application of CNNs in veterinary medicine.

Taking the aforementioned issues into consideration, this study sought to analyze and answer the following research questions.

1.4. RESEARCH QUESTIONS

The following fundamental research questions will be addressed by the study as it is carried out.

RQ 1: What are the various CNN architectures and techniques used to identify Foot and Mouth Disease (FMD)?

RQ 2: What are the key hyper parameters that significantly impact the performance of CNN models in detecting FMD?

RQ 3: How does the performance of CNN models compare to that of classical machine learning models in detection of FMD?

1.5. OBJECTIVE OF THE STUDY

1.5.1. General Objective

The General Objective of the study is to detect and identify Cattle Farms Foot-And-Mouth Disease Using Convolutional neural network models.

1.5.2. Specific Objectives

- To examine the most suitable and appropriate CNN models for the detection and accurate diagnosis of Foot and Mouth Disease (FMD) in cattle farms.
- To determine the key hyperparameters that significantly impact the performance of CNN models in detecting Foot and Mouth Disease (FMD) in cattle.

- To compare the performance of CNN models with classical machine learning models in detecting Foot and Mouth Disease (FMD) in cattle.
- To evaluate the performance of the proposed system using applicable metrics for cattle detection of foot and mouth Disease (FMD).

1.6. SCOPE AND LIMITATIONS OF THE STUDY

1.6.1. Scope of the Study

The scope of this thesis is to evaluate Convolutional Neural Networks (CNN) for the detection of Foot-and-Mouth Disease (FMD) in cattle using images of both healthy cattle and cattle infected with FMD. The study compares several CNN architectures, such as VGG16, Inception V3, and DenseNet, to establish their accuracy in identifying FMD in livestock. The study includes dataset collecting, data preparation, model selection, feature extraction, training, and evaluation to determine how well each CNN architecture distinguishes between healthy and FMD-infected cattle using image attributes. Ethical issues regarding the use of animal images for research purposes is carefully addressed, ensuring adherence to ethical principles and securing relevant consents. The work seeks to give useful insights into the use of deep learning techniques, notably CNNs, for the early and accurate identification of FMD in cattle, hence contributing to advances in computer-aided disease detection in veterinary medicine.

1.6.2. Limitation of the Study

Despite the valuable insights gained from this study on the detection of Foot and Mouth Disease (FMD) using CNN models, there are certain limitations that should be acknowledged. Firstly, due to the limited availability of data, online datasets were utilized to supplement the collected dataset. Secondly, although the collected data is sufficient to train the model, it should be noted that the dataset itself is not extensive in size. This may impact the generalizability and robustness of the trained model when applied to a larger and more diverse population. Lastly, the scarcity of existing research on FMD detection using CNN models presents a limitation in terms of the ability to compare and validate the findings against a broader body of literature.

1.7. SIGNIFICANCE OF THE STUDY

The significance of the study lies in its potential to revolutionize disease detection and management in livestock farming. By utilizing machine learning techniques for early detection of Foot-and-Mouth Disease (FMD), the study aims to address the following key points:

- Economic Impact: FMD has a significant economic impact on most countries, where livestock farming plays a crucial role in the economy. In crop-livestock mixed systems, the morbidity rate from FMD is 85.2%, whereas in pastoral systems, it is 94.9%. The average economic loss per impacted herd in the pastoral system was $174, or $5.3 per head of cattle. By developing a machine learning system for FMD detection, the study can help reduce economic losses associated with the disease, such as trade bans, high control costs, and reduced animal production.
- Livelihood Improvement: Livestock farming is a major source of livelihood for many people. Detecting and controlling FMD promptly can help improve herd fertility, encourage the use of high-productivity breeds, and ultimately enhance the efficiency of livestock farming, benefiting farmers directly.
- Technological Advancement: By focusing on the application of deep learning techniques in disease detection, the study contributes to the advancement of technology in the agricultural sector. This can lead to the development of innovative tools and solutions for improving livestock health and productivity.
- Research Gap Bridging: The study addresses a gap in existing research by specifically tailoring a deep learning system for the detection of FMD in cattle farms.
- Global Health Impact: Effective disease detection and management in livestock farming not only benefit local economies but also contribute to global health security by preventing the spread of diseases across borders. The study's findings and recommendations can potentially be applied in other regions facing similar challenges with FMD detection and control.

In conclusion, the study's significance lies in its potential to improve livestock health, enhance economic stability, and advance technological solutions for disease detection cattle farming.

1.8. ORGANIZATION OF THE THESIS

The research work is divided into seven chapters which are briefly discussed below.

Chapter One: This chapter summarizes the study and explains how and what will be performed.

Chapter Two: This chapter discusses the relevant literature on deep learning, CNN, transfer learning, and disease detection.

Chapter Three: This chapter discusses research procedures such as data collecting, preprocessing, design tools, development frameworks and platforms, and evaluation techniques.

Chapter Four: This chapter employs block diagrams, flow charts, and mathematical explanations to provide a comprehensive picture of the proposed solution to the problem.

Chapter Five: This chapter discusses how to put the recommended solution to the problem into action. It also offers a code sample for the potential solutions.

Chapter Six: This chapter examines and contrasts the results of the proposed solution using figures, graphs, and tables.

Chapter Seven: This chapter outlines the research results. The study's result is offered, along with recommendations for further research.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

2.1 FOOT-AND-MOUTH DISEASE – AN OVERVIEW

Foot and mouth disease (FMD) is a severe, highly contagious viral sickness in animals that has significant economic effects. The disease affects cattle, swine, sheep, goats, and other cloven-hoofed ruminants. It is a transboundary animal disease (TAD) that has a considerable influence on livestock output, as well as regional and global traffic in animals and their products (WOAH, 2024).

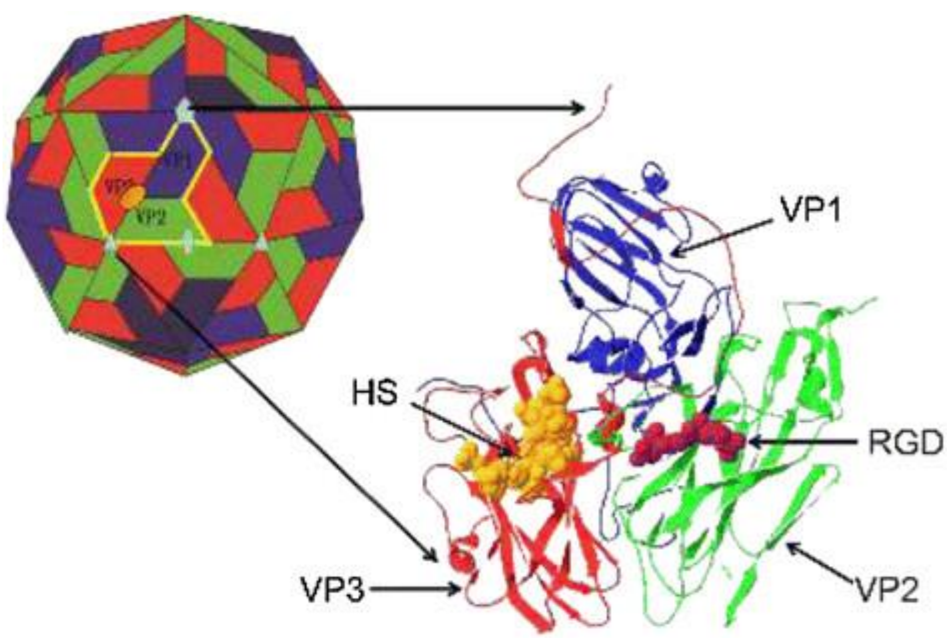


Figure 2.1 *Structure of the Foot and Mouth Virus*

Source (Han, Guo, & Sun, 2014)

The disease is expected to affect 77% of the world's cattle population, mostly in Africa, the Middle East, and Asia, as well as a small portion of South America. Countries that are now free of FMD without immunization have a continual threat of infiltration. Low and lower-middle income nations bear 75 percent of the expenditures associated with FMD prevention and control. Africa and Eurasia are the regions with the highest expenses, accounting for 50% and 33% of overall costs respectively. FMD is caused by an Aphthovirus from the Picornaviridae family; seven strains (A, O, C, SAT1, SAT2, SAT3, and Asia1) are prevalent in various areas throughout the world. Each strain needs a distinct vaccine to confer protection to the inoculated animal. Its prevention is dependent on early detection and warning systems, as well as the adoption of efficient surveillance methods. FMD is the first illness for which WOAH has produced an official list of disease-free nations, which can be formally recognized as disease-free in their totality or in certain zones and compartments (WOAH, 2024).

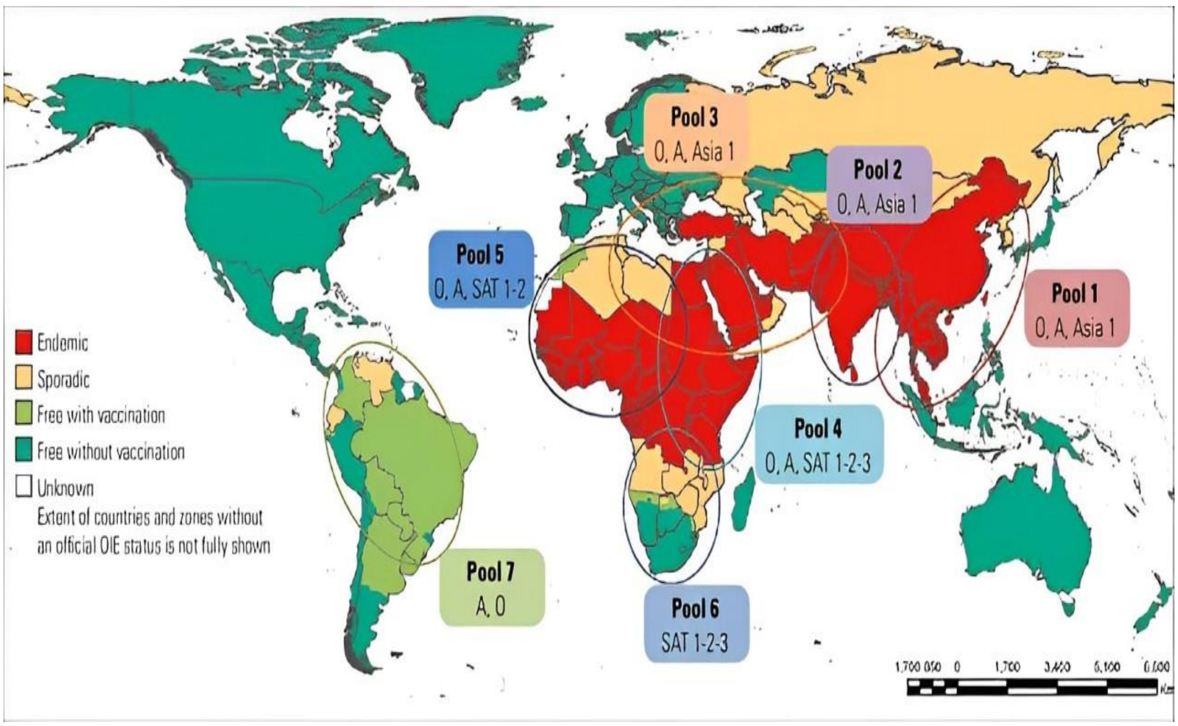


Figure 2.2 *Serotypes classified on pool*

Source: (Woldemariyam et al., 2023)

International trading in animals and animal products is prohibited in countries plagued by foot and mouth disease. Animals, biodiversity, food security, and smallholder farmers' livelihoods are all at risk in emerging nations. In order to mitigate foodborne illness, the Food Agriculture Organization (FAO) and the World Organization for Animal Health devised a 15-year Global Control Strategy in 2012. Despite this, there has been a noticeable decrease in the burden produced by FMD, even if the endemic may not be completely eliminated in the next 15 years.

Foot and Mouth Disease is common in Asia, Africa, and the Middle East. The majority of Latin American countries adopt zoning and are declared FMD-free, whether or not they have received vaccinations. Australia, New Zealand, Indonesia, Central and North America, and Continental Western Europe are currently FMD-free. However, FMD is a transboundary animal disease that may occur periodically in any ordinarily free region (WOAH, 2024).

The FMD epidemic affected almost every developing country on the continents of Asia, Africa, and Europe. The National Veterinary Institute (NVI), the nation's hub for veterinary vaccine production, and the National Animal Health Diagnostic and Investigation Center (NAHDIC), an East African reference laboratory for numerous diseases, have been the primary sites of investigation for Ethiopia's FMD outbreaks since relatively recently. These two agencies, in collaboration with sub-national veterinary laboratories, serve as the cornerstone for the nation's disease prevention and control programs, as well as outbreak investigations. FMD epidemics have shown recurrent patterns in a number of situations. Every year, FMD outbreaks may be found in practically every region of the nation, with demonstrable variations in the severity of the cases and a variety of geographical settings (Wubshet , Dai , Li , & Zhang , 2019).

2.1.1 Clinical Signs of Foot And Mouth

FMDV replicates and spreads quickly within infected animals, among vulnerable animals in touch with them, and through aerosols, making it one of the most infectious illnesses that can affect people or animals. After exposure, disease symptoms may develop two to three days later and persist for seven to ten days (Grubman & Barry , 2004).

Blisters (or vesicles) on the lips, nose, tongue, inside the oral cavity, between the toes, above the hooves, on the teats, and at skin pressure points are the typical clinical symptoms. Blisters that have burst may be extremely limp and unable to move or eat. Blisters usually heal in 7 days, although sometimes they take longer. Issues might occur, such secondary bacterial infections of exposed blisters. Fever, depression, hypersalivation, appetite loss, weight loss, growth retardation, and a reduction in milk production are some typical symptoms that may persist even after recovery. According to reports, animals with chronic illnesses lose 80% of their milk production (WOAH, 2024).

2.1.2 The Transmission of the Disease

All of the afflicted animals' fluids and excretions contain FMD. Additionally, these animals release a significant quantity of aerosolized virus, which can spread orally or through the respiratory system to other animals (WOAH, 2024).

The virus can be found in milk and sperm for up to four days before an animal exhibits clinical indications of sickness. FMD's relevance stems from the virus's ease of transmission through any or all of the following:

- Risk factors include freshly introduced diseased animals,
- Contaminated materials include hay, feed, water, milk, and biologics,
- Contaminated clothing, footwear, and equipment.
- Raw or poorly prepared meat or animal products contaminated with viruses can be given to animals.
- Air currents can transfer contaminated aerosols from an infected premise.

Sometimes, after recovering from a sickness, animals may carry the virus and start fresh outbreaks of the disease.

2.1.3 Diagnosis and Control of FMD

Diagnosis of FMD is based on clinical signs and lab findings (I, 2007). Diagnoses for FMDV infection can be made by looking for a particular antibody response. Viral and serological testing are two methods of diagnosing FMD. The purpose of serological tests, which are diagnostic procedures, is to find out whether antibodies are present in serum or other bodily fluids. These examinations evaluate the immune system's reaction to infections, such as bacteria, viruses, and other microbes. While the isolation of live virus particles or the detection of viral genetic material (DNA or RNA) are common components of virological assays. The outcomes of virological studies are crucial for characterizing viral strains, identifying viral pathogens, and guiding clinical and public health policies for illness management and control. Solid-phase RT-PCR (reverse transcription polymerase chain reaction), CFT (complement fixation test), PCR (polymerase chain reaction), and some non-structural protein antibody tests are also available, such as enzyme-linked immune electro transfer blot assay, are among the frequently used tests (Abdullah, et al., 2020).

Laboratory diagnosis is an additional conventional diagnostic technique. Samples from the suspected animal's epithelium or vesicular fluid are taken in order to diagnose FMD in animals. The preferred cattle sample contains lesions from the mouth, hoof, and foot tissues. Pigs with fluid-filled vesicular wounds from their hooves should have their tongue and nostrils collected (Hirsh, MacLachlan, & Walker, 2004). Sample can be collected in two ways:

1. Clinical Specimens: Samples collected from affected animals typically include epithelial tissue, vesicular fluid, and serum.

2. Environmental Samples: Samples from the environment, such as swabs from surfaces and devices where viral particles may be present, are also collected for testing.

The common and last one is through clinical diagnosis. FMD is a viral disease that spreads quickly and easily among cloven-hoofed animals, both domestic and wild. Infected animals may have fever, appetite loss, weight loss, lameness, drooling, and sadness. FMD also manifests as blister-like lesions on the tongue, teats, and hooves. Additionally, it may result in blisters that burst and eventually heal on the lips, feet, and udder. Because of injury to the skin around the hoof, growth rings may be seen on the hoof wall. The kind of virus, exposure level, animal age and breed, host species, and immunological state are some of the variables that affect the clinical symptoms of foodborne illness (FMD) (Girma , et al., 2023).

The traditional method of detecting Foot-and-Mouth Disease (FMD) in cattle is time-consuming and labor-intensive, often involving visual inspection and laboratory testing. Furthermore, there is currently no specific antiviral treatment for Foot and Mouth Disease (FMD) in animals. The approach to managing FMD primarily involves supportive care and containment measures. This typically includes Isolation of Infected Animals, Pain and Fever Management, Hoof Care and Clinical Monitoring. As a result, the need for more efficient and accurate diagnostic methods is critical for early detection and control of the disease. This research aims to address this gap by leveraging machine learning techniques to develop a faster and more reliable approach for detecting FMD in cattle.

2.2 CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNNs) are a class of deep neural networks particularly well-suited for processing structured grid-like data, such as images. Convolutional, pooling, and fully connected layers are among the several layers of interconnec ted neurons that make up CNNs, which are modeled after the structure of the animal visual cor tex.In order to extract local features through convolution operations and capture patterns such as edges, textures, and forms, the convolutional layers apply filters (kernels) throughout the in put picture.The feature maps are then downsampled by pooling layers in order to save comput ational overhead and spatial dimensions without sacrificing significant information.
Repetitive convolution and pooling processes teach CNNs to learn hierarchical representations of the input data, which helps them recognize complex patterns and relationships in pictures.
By delivering state-of-the-art performance in tasks like image classification, object recognition, and image segmentation, CNNs have revolutionized a number of domains, including computer vision, natural language processing, and medical image analysis. In contemporary machine learning applications, CNNs are essential tools due to their capacity to automatically extract features from unprocessed data and to generalize effectively to new instances (IBM, 2024).

Suppose we have an input image described by tensor **I** of dimension m1 x m2 x mc. Where,

m_1 = height of image (2.1)

m_2 = wifth of image (2.2)

m_c = number of channels (2.3)

We use a filter that is also a tensor of dimensions (n1 x n2 x nc). (The kernel has the same number of channels as the input picture.). The filter goes from left to right across the picture, multiplying that component of I and K and summing those results. The stride sets the magnitude of the step by which the filter moves in response to an image. The output of I and K is another tensor with dimensions (m1 — n1 +1) x (m2 — n2 +1) x 1.

$$\text{dim of } I = m_1 * m_2 * m_c \tag{2.4}$$

$$\text{dim of } K = n_1 * n_2 * n_c \tag{2.5}$$

$$\text{dim of } F = (m_1 - n_1 + 1) * (m_2 - n_2 + 1) * 1 \tag{2.6}$$

And,

$$F[i, j] = (I * K)_{[i,j]} \tag{2.7}$$

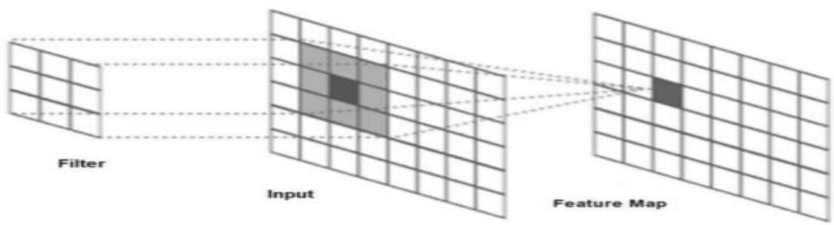


Figure 2.3 *Convolution Operation*

Source: (Wani et al., Basics of Supervised Deep Learning, 2020)

One of the most fascinating deep learning architectures is convolutional neural networks, particularly in light of the model's ability to assist in learning maintain characteristics. When it comes to image identification for robots, automation, and agriculture, CNN has produced impressive results. Certain applications perform difficult tasks like plant disease diagnosis using CNN and computer vision (Medical Imaging Informatics: An Overview, 2019).

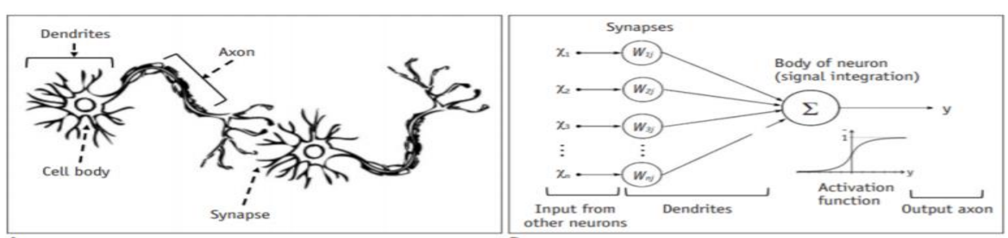


Figure 2.4 *A Real Neuron B Artificial Neuron*

Source: (Medical Imaging Informatics: An Overview, 2019)

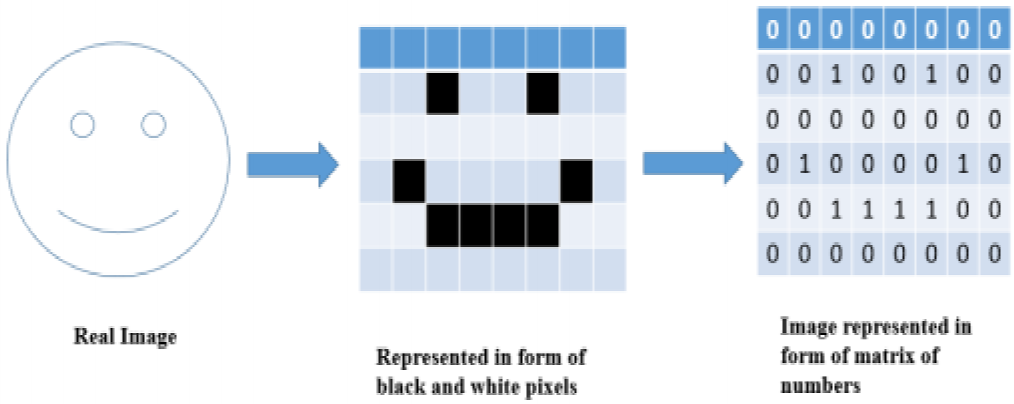


Figure 2.5 *Real image data A, represented in form of an array B and represented in form of pixels of 0's and 1's C*

Source: (Mishra, 2023)

Convolutional neural networks receive pictures as input and image pixels as input in the form of arrays, as seen in Figure 2.5.

The building blocks of basic CNN architecture are the convolution layer, pooling layer, and fully connected layer.

2.2.1 Convolution Layer

A key element of convolutional neural networks (CNNs) is the convolutional layer, which is skilled at extracting features from input data, especially pictures. The layer, which consists of learnable filters, applies these filters by sliding windows over the input picture. Each filter creates a matching feature map by capturing local patterns and features as it passes over the picture. The network gains the ability to automatically recognize significant patterns like edges, textures, and forms during training. This allows the network to learn how to modify the filter weights. Convolutional layers are the foundation of computer vision applications for tasks like object identification, picture segmentation, and image classification because of their sparse connectivity and weight sharing, which enable them to learn hierarchical representations of incoming data effectively (Namita, 2023).

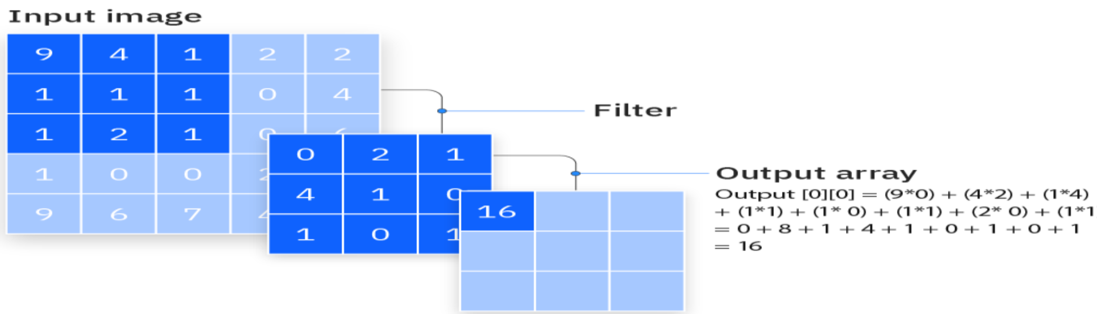


Figure 2.6 *Convolution Operation*
Source: (Namita, 2023)

2.2.2 Pooling Layer

Dimensionality reduction is achieved via pooling layers, sometimes referred to as downsampling, which reduces the number of parameters in an input. Similar to the convolutional layer, the pooling procedure applies a filter to the whole input; however, this filter is weightless. Rather, the output array is filled by the kernel using an aggregation function on the values in the receptive field. There are basically two types of pooling:

- Max pooling: entails choosing the largest value inside each non-overlapping rectangular zone as the output after partitioning the input feature map into those sections. This makes the network more resilient to translation differences by keeping the most salient elements and eliminating unnecessary information.
- Average pooling: The average value in the receptive field is determined as the filter runs over the input and sent to the output array.

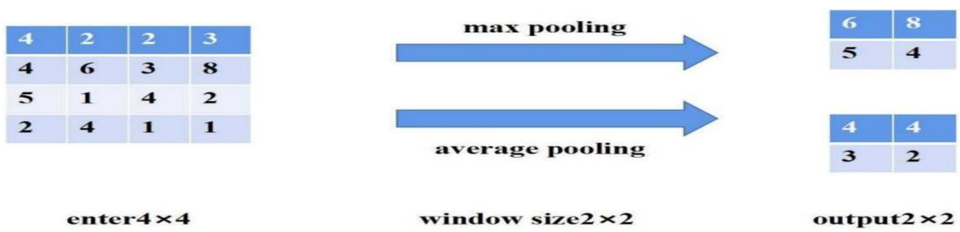


Figure 2.7 *Pooling layer*

Source: (Pei Yin et al., 2022)

2.2.3 Fully Connected Layer

The full-connected layer's name accurately represents its characteristics. In partly linked layers, as previously mentioned, pixel values from the input picture are not directly connected to the output layer. Every output layer node in the fully-connected layer, on the other hand, links directly to every other layer node. Using the features gathered from the previous layers and their different filters, this layer performs classification. While FC layers usually employ a softmax activation function to effectively classify inputs, providing a probability range from 0 to 1, convolutional and pooling layers frequently use ReLu functions (Namita, 2023).

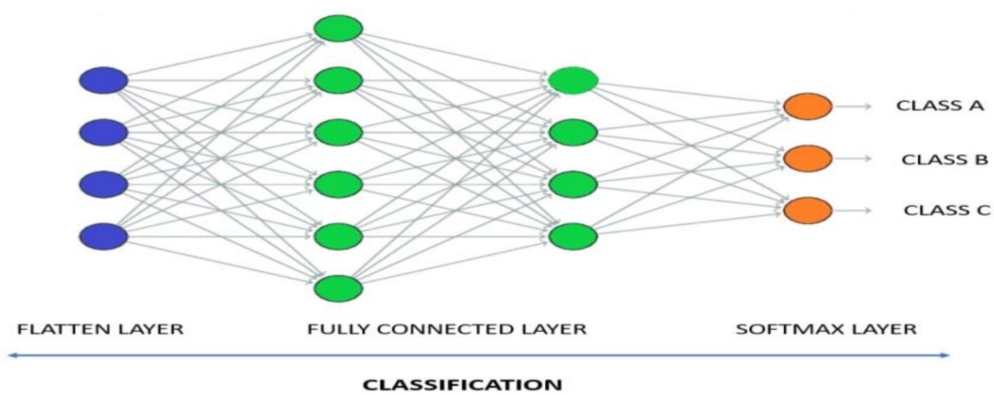


Figure 2.8 *Fully connected layer*
Source: (Namita, 2023)

2.2.4 Activation Function

The activation function determines whether a neuron should be activated or not by computing the weighted sum and applying bias to it. The goal of the activation function is to increase the output of a neuron's nonlinearity. We are aware that the neurons in the neural network operate in line with the activation functions that correspond with weight and bias. Based on the inaccuracy at the output, we would adjust the weights and biases of the neurons in a neural network. The process is referred to as back-propagation. Back-propagation is made possible by activation functions since they provide the gradients and the error for updating the weights and biases (Activation function, n.d.).

2.2.4.1 ***Softmax Activation Function***

The Softmax function is a mixture of many sigmoid functions. The sigmoid function provides values ranging from 0 to 1, which can be interpreted as probability for a certain class of data items.

The softmax function, unlike sigmoid functions used for binary classification, is applicable to multiclass classification issues. The function calculates probability for each data point across all classes (Sharma & Sharma, 2020).

$$s(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{2.7}$$

When creating a network or model for multiple class classification, the output layer should contain the same number of neurons as the target classes.

2.2.4.2 ***ReLu(Rectified Linear Unit)***

While fixing the vanishing gradients problem, the rectified linear unit (ReLU), also known as the rectifier activation function, adds nonlinearity to a deep learning network. It acknowledges the advantages of its case. In deep learning, it is a frequently utilized activation function. When computing neural network backpropagation, differentiation for the ReLU is rather straightforward. We will just assume that the derivative at 0 is also zero. This is usually not a big concern as it functions well most of the time. The derivative of the function equals the slope. Positive values have a slope of 1.0, whereas negative values have a slope of 0.0 (A Gentle Introduction to the Rectified Linear Unit (ReLU), n.d.).

2.2.5 Optimization Algorithms

Optimization algorithms are a type of algorithm that seeks the optimal solution to a given issue. The purpose of an optimization algorithm is to identify the best solution that minimizes or maximizes a particular objective function. There are several types of optimization algorithms, each with unique strengths and limitations. Some of the most prominent optimization methods include Adam, gradient descent, conjugate gradient, Newton's Method, and Simulated Annealing (Optimization Algorithms in Machine Learning, n.d.).

2.2.5.1 *Adaptive Moment Estimation (Adam)*

Adaptive Moment Estimation, or Adam for short, is a popular optimization approach used in neural network training. It combines the advantages of momentum and RMSprop, two additional well-liked optimization strategies. Adam keeps up a dynamic per-parameter learning rate throughout training, which is dependent on the gradients' first and second moments. In comparison to conventional gradient descent techniques, this adaptive learning rate facilitates the management of various parameter scales and gradient variations, resulting in faster convergence and improved performance. Adam also adds momentum, which accumulates previous gradients and speeds up the optimization process. By directing the optimization process in pertinent directions, these collected gradients help Adam more effectively traverse intricate loss landscapes (Optimization Algorithms in Machine Learning, n.d.).

ADAM combines two stochastic gradient descent methods: adaptive gradients and root mean square propagation. Instead of calculating the gradient using the complete data set, this optimization approach generates a stochastic approximation using a randomly chosen data subset.

2.3 DEEP LEARNING

"Deep learning" is a subfield of machine learning that uses algorithms inspired by the structure of the human brain. The ability of deep learning neural networks to process large volumes of structured input distinguishes them from machine learning. Figure 2.9 illustrates other variances. Deep learning uses neural networks to accomplish this automatically, whereas machine learning employs data scientists to manually extract information.

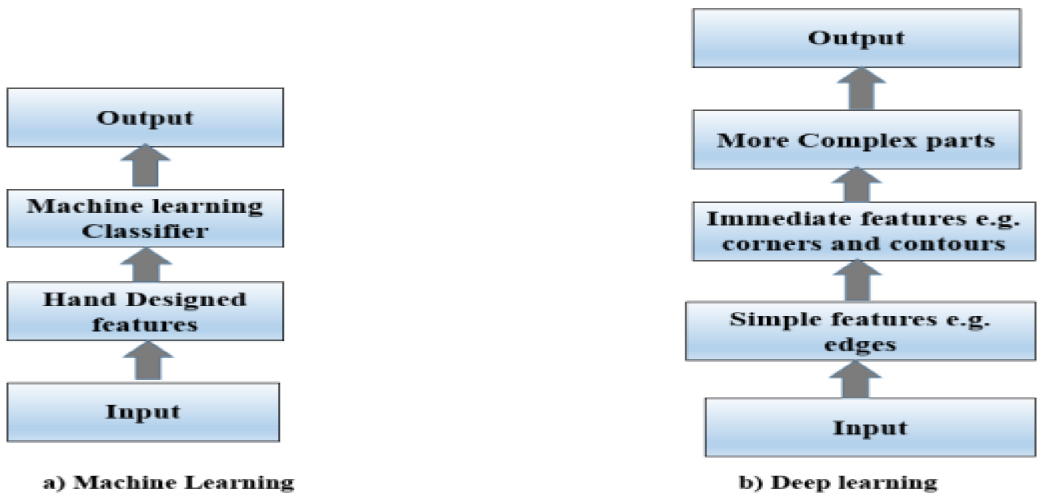


Figure 2.9 *Machine Learning and Deep Learning*

Source: (Wani et al., Advances in Deep Learning, 2020)

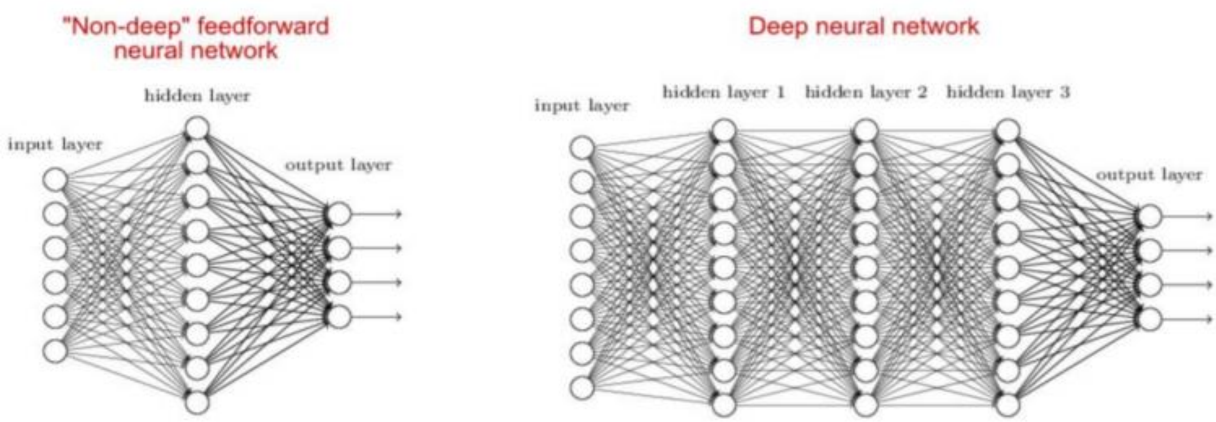


Figure 2.10 *Shallow vs Deep neural network*

Source: (Managwu, 2022)

Models are classified into two groups based on their complexity: shallow and deep. Shallow learning uses basic models with a minimal number of layers that are appropriate for jobs with clear patterns. Examples include linear regression, decision trees, and k-nearest neighbors. Deep learning, on the other hand, makes use of large neural networks with several layers to discover detailed patterns from data. It excels in jobs like picture identification and natural language processing (Managwu, 2022).

Recent research have employed deep convolutional neural networks to identify illnesses in people, animals, and plants. Visual recognition is a difficult task for both people and automated systems due to the diversity of pictures' properties. When shooting a picture, various elements, including position, scale, perspective, and backdrop, can produce differences in light intensity. To accurately and reliably identify the image, the automation system that has to be constructed must be able to consider the previously described views (Lu et al., 2018).

2.3.1 Transfer Learning

A machine learning technique called transfer learning uses a model that was trained on one task to inform a subsequent task. When there is not enough data for the second task or when the second task is similar to the first task, this is advantageous.

Transfer learning is mostly employed in computer vision and natural language processing applications such as sentiment analysis owing to the massive computational power required. Transfer learning is not a machine learning technology, but rather a "design methodology" within the discipline, such as active learning. It is also not a separate component or academic area of machine learning. Nonetheless, it has gained popularity when combined with neural networks, which need massive quantities of data and computer power (What Is Transfer Learning? Definition, Methods, and Applications, n.d.).

A neural network must often be trained with a big quantity of data at first, but access to that data is not always possible. This is where transfer learning comes into play. Because the model has already been pre-trained, transfer learning makes it possible to build a strong machine learning model with relatively little training data. This is particularly helpful in natural language processing, as large-scale labeled data sets creation demands a great degree of skill. Additionally, training time is reduced because it might take a deep neural network days or even weeks to learn from scratch on a difficult task (Built in, 2023).

A large visual database called ImageNet was developed for study purposes related to visual object identification software. Over 14 million photos have been carefully annotated by the project to identify the objects they show, and at least one million of them have bounding boxes. There are more than 20,000 categories in ImageNet; a typical category like "strawberry" or "balloon" has several hundred photos in it. ImageNet provides a publicly accessible database of annotations for third-party picture URLs, but it has no control over the underlying pictures. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which has been held annually since 2010, is a software competition where software programs compete to reliably identify and recognize objects and scenes (Imagenet, 2024).

The most well-prepared classification models, Alexnet, GoogleNet, ResNet, Lenet, Visual Geometry Group (VGG) 16, and VGG19, were trained on the ImageNet-1K dataset. The ImageNet data collection has been used to train deep learning models such as VGG, Google Net, ResNet, Inception, Densnet, and Lenet to recognize pictures. These models can also have extra layers added to them to detect more complex structures.

2.3.2 Pre-trained Models

There are different pre-trained models used for various applications among those we will see Inception V3, VGG16 and Densenet.

2.3.2.1 *VGG16*

VGG16 is a convolutional neural network model presented by K. Simonyan and A. Zisserman of the University of Oxford in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition." The model achieves 92.7% top-5 test accuracy in ImageNet, a dataset of over 14 million pictures categorized into 1000 classes. It was one of the notable models submitted to the ILSVRC-2014. It improves on AlexNet by replacing big kernel-sized filters (11 and 5 in the first and second convolutional layers, respectively) with several 3×3 kernel-sized filters sequentially. VGG16 was trained for weeks using NVIDIA Titan Black GPUs (VGG16 – Convolutional Network for Classification and Detection, 2018).

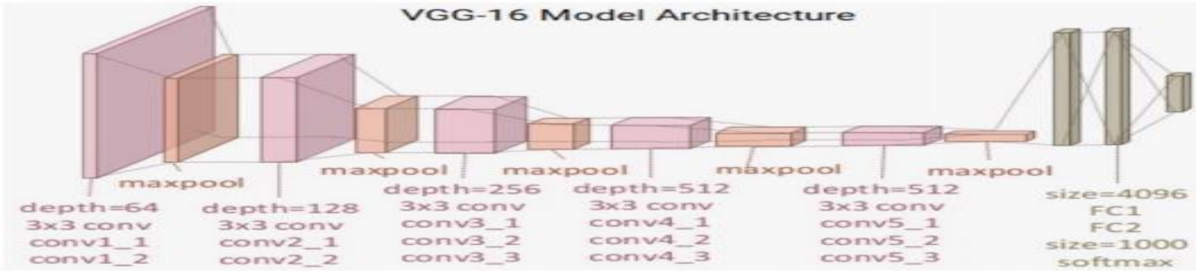


Figure 2.11 *VGG16 Architecture*

Source: (Jahan, 2023)

2.3.2.2 ***DenseNet***

Densely Connected Convolutional Networks (DenseNet) is a feed-forward convolutional neural network (CNN) architecture that connects all layers. This helps the network to learn more effectively by recycling features, which reduces the number of parameters and improves gradient flow when training.The enseNet architecture is based on a simple and fundamental principle: by concatenating the feature maps of all previous layers, a dense block allows each layer to access the features of all preceding levels. In traditional CNNs, each layer only has access to the properties of the layer immediately before it. The architecture of DenseNet is made up of transition layers and dense blocks. Each convolutional layer in a dense block is connected to the other layers in the block. This is performed by connecting the output of each layer to the input of the following layer, resulting in a "shortcut" link. The transition layers reduce the size of feature maps across dense blocks, allowing the network to expand efficiently (Deepchecks, 2023).

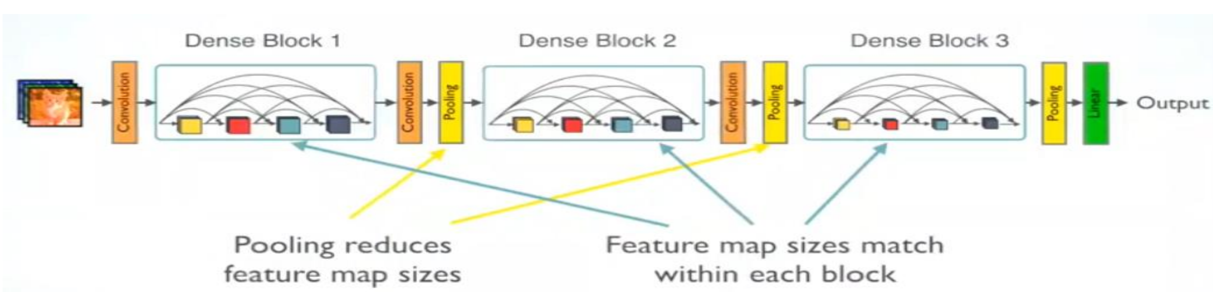


Figure 2.12 *DenseBlocks and Layers*

Source: Towards Data Science

2.3.2.3 ***Inception V3***

A 48-layer convolutional neural network that has been pre-trained with more than a million images from the ImageNet collection is called Inception-v3. Keyboards, mice, pencils, and a variety of animals are among the 1000 object categories that this pre-trained network can identify photographs of. Consequently, the network has acquired comprehensive feature representations for a wide range of images. The size of the network's input image is 299 by 299. The model initially gathers general features from the input photographs, which are subsequently categorized in the second stage (Built in, 2023).

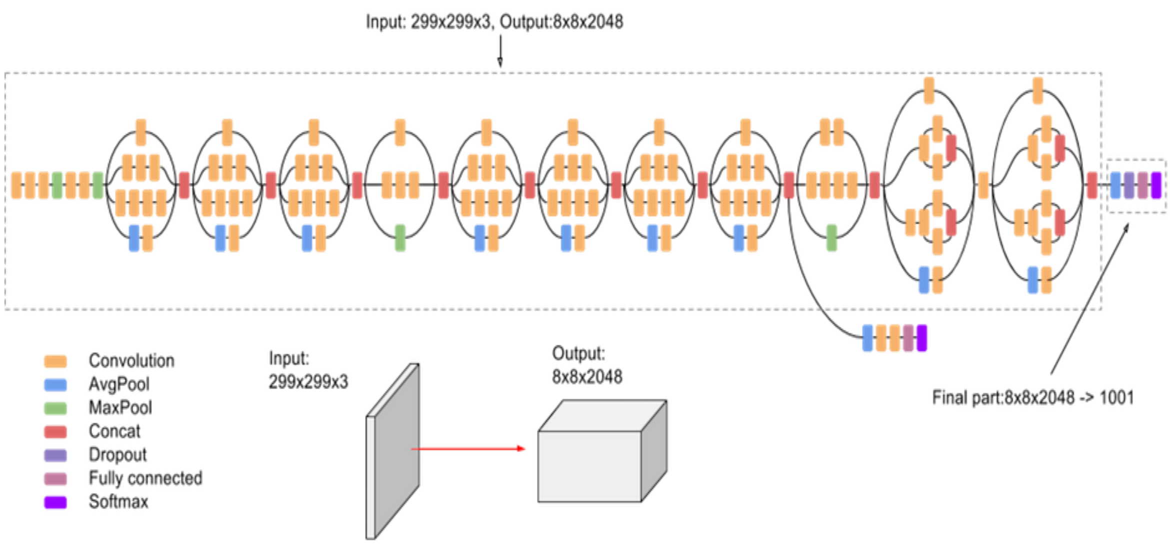


Figure 2.13 *Inception V3 Architecture*
(Wani et al., Advances in Deep Learning, 2020)

2.4 APPLICATION OF MACHINE LEARNING IN DISEASE DETECTION

Machine learning (ML) is a subfield of artificial intelligence that explores the development and implementation of statistical algorithms capable of successfully generalizing and hence completing tasks without explicit instructions. Recent research indicates that generative artificial neural networks surpass many earlier strategies. Large language models, computer vision, audio recognition, email filtering, agriculture, and health are some of the industries that have used machine learning methodologies since developing algorithms to accomplish critical tasks would be too expensive. One approach to machine learning is to develop a model that is trained on training data and then has the ability to understand further data in order to generate predictions. Machine learning systems have been explored and used to a variety of models (Machine Learning, n.d.).

Machine learning has transformed illness detection by automating, accurately, and efficiently analyzing massive volumes of medical data. These algorithms can find trends and forecast illness risk by analyzing patient symptoms, genetic history, demographic data, and medical test results. By comparing a patient's data to vast databases of known instances, machine learning models can give significant insights to healthcare providers, speeding the diagnostic process and lowering the chance of misdiagnosis (Machine Learning, n.d.).

One of the primary benefits of employing machine learning for illness diagnosis is its capacity to handle complicated, high-dimensional data that would be difficult for humans to interpret. Machine learning methods such as support vector machines (SVMs) and neural networks can manage vast amounts of organized and unstructured data, including medical pictures, genetic sequences, and electronic health records. This enables for more extensive and accurate illness prediction, resulting in earlier intervention and better patient outcomes. As machine learning advances, its integration with healthcare will become more seamless, providing doctors with cutting-edge technology to diagnose illnesses in their early stages (Built in, 2023).

Precision livestock management, which includes vital disease diagnosis, vaccination, production management, tracking, and health monitoring, has benefited from the improved machine learning technology. Both human health and animal husbandry are greatly threatened by major animal illnesses. The prediction and analysis of animal illnesses using big data is becoming more and more essential as globalization deepens and data resources increase. The focus of machine learning is to make computers learn how to learn from data and use the learned experience to analyze and predict (Shuwen , Qiang , & Qin, 2020).

ML algorithms are being used for a wider range of tasks in animal and veterinary public health. In addition to broad fields of application similar to those identified in global health, more focused uses of machine learning to handle specific tasks are emerging. In the field of animal health surveillance, ML has been used to construct models that predict which farms are more likely to get infected with certain viruses, based on historical case data and a list of probable risk factors (J , M, Y, & E , 2023).

The diagnosis of cattle foot and mouth disease might be improved in terms of accuracy, speed, and efficacy with the use of machine learning algorithms. These algorithms have the potential to monitor and halt the spread of FMD, therefore mitigating its impact on cow herds and the economy. Machine learning algorithms can be used to identify potential FMD outbreaks by analyzing vast amounts of data on animal health. This covers historical vaccination data, symptom patterns, and animal migration patterns. Machine learning algorithms can identify patterns and correlations in this data to forecast the probability of FMD outbreaks and help implement control measures to stop the disease's future spread. Machine learning has the potential to mitigate the financial impact of outbreaks, safeguard animal herds from injury, and control FMD epidemics early on (Akash, 2023).

The researchers Ekramul Hossain, Ashad_Kabir, Lihong Zheng, Dave L. Swain, Shawn McGrath & Jonathan Medway. (Ekramul, Ashad , Lihong , & Dave, 2022) reviewed existing literature to assess the use of machine learning in cattle identification. The paper retrieved 731 studies from four online scholarly databases, selected 55 articles for in-depth investigation, and highlighted the most commonly used ML and DL models for this purpose. The surveyed report identified support vector machine (SVM), k-nearest neighbor (KNN), and artificial neural network (ANN) as the most frequently utilized ML models for cattle identification. These models have demonstrated effectiveness in pattern recognition and classification tasks, making them suitable for distinguishing individual cattle based on various features. In addition to traditional ML models, the paper highlighted popular DL models in the selected studies, including convolutional neural network (CNN), residual network (ResNet), Inception, You Only Look Once (YOLO), and Faster R-CNN. These DL models have shown promise in image recognition and object detection, offering potential applications in cattle identification through visual data analysis. The paper detailed important factors to consider when choosing a technique or method for cattle identification. This included considerations such as accuracy, efficiency, and scalability of the models. Furthermore, the authors identified major challenges in cattle identification, notably the limited availability of publicly accessible datasets. This promises that machine learning can be used to develop a faster and more reliable approach for detecting FMD in cattle.

Prior research on disease detection using deep learning architectures has set the framework for utilizing advanced computational approaches to identify and forecast pathogenic diseases in animals. Notably, DenseNet-121, known for its effectiveness in deep reading tasks, has emerged as a critical tool for detecting subtle patterns suggestive of sickness. Building on this basis, recent studies have shown that convolutional neural networks (CNNs) can forecast Lumpy Skin Disease Virus (LSDV) based only on visual data. These investigations demonstrated the amazing accuracy of this method in recognizing lumpy skin disorders, which is critical for the early detection and control of LSDV in cattle herds, by dividing pictures into LSDV and Non-LSDV classes using proprietary deep learning models. This body of work underscores the transformative impact of deep learning in veterinary medicine, offering a promising avenue for automated disease detection and surveillance in animals (Mazi et al., 2023).

In recent works on illness classification using segmentation and feature extraction, researchers used experimental techniques based on designed neural network topologies. These studies use picture segmentation and feature extraction to identify patterns indicative of various illnesses. In our experiment, we used a dataset of 1200 photos classified into 15 distinct illness groups for training purposes. Each image, formatted in RGB and measuring 230x230x3, was tagged with an average of 50 points to highlight key elements. Data sequences were created using feature extraction and then input into a convolutional neural network (CNN) for pattern identification. The findings revealed a significant eigen variation in spatial vision, with a remarkable test accuracy of 97.06%. Notably, real-time photographs of ill animals were used to show the effectiveness of the suggested method. This demonstrates the potential of our technology to save substantial time and money while also providing accurate illness forecasts without the need for veterinary knowledge. Such breakthroughs offer great potential for improving livestock disease detection and management procedures, ushering in a new era of efficient and accessible healthcare solutions for animal populations (Mohan, Raju, & Janarthanan, 2019).

Despite significant advances in machine learning and its applications in a variety of disciplines, there has been little study into the identification of cattle illnesses using machine learning methods. In contrast, there is a considerable body of research on the application of machine learning in plant and human health, with machine learning approaches being investigated for illness detection and prediction. The scarcity of research on machine learning-based cattle disease identification emphasizes the need for more study in this field. The development of effective machine learning models for detecting livestock diseases has the potential to greatly enhance animal health and wellbeing, as well as livestock production efficiency. Therefore, it is essential to explore machine learning approaches for detecting and predicting livestock diseases to address the current knowledge gap.

2.5 RELATED WORKS

To gain a thorough understanding of this research topic about foot and mouth disease detection, specifically, using machine learning algorithms, related literature such as journals, articles, and papers were reviewed in addition to the Internet. Despite the increasing application of machine learning algorithms in various domains, there is a noticeable scarcity of research specifically focused on the detection of Foot-and-Mouth Disease (FMD) using these techniques. The limited existing literature underscores the need for more comprehensive investigations into the potential of machine learning for FMD detection, highlighting the critical gap in this area of study. Several studies that have been conducted to implement an intelligent system for detecting FMD, as stated below.

Akash (Akash, 2023) provides a comprehensive overview of a study conducted to develop a machine learning-based approach for detecting Foot-and-Mouth Disease (FMD) in cattle in his paper. The study involved collecting imaging results from cattle with suspected disease, and then pre-processing the data before applying hyperparameter tuning to eight different machine learning algorithms. Whether all predictive factors were included in the model or only meteorological variables were used as predictors, the artificial neural network (ANN) method performed best in terms of the area under the curve (AUC) metric, according to the study. The models achieved a high accuracy of 97%. The researcher also highlighted the limitations and challenges associated with using machine learning algorithms for FMD identification, including the reliance on high-quality data, technical constraints, implementation costs, and the requirement for ongoing maintenance. The researcher emphasized the need for further study to fully realize the potential of machine learning techniques for FMD identification and to address these limitations. One notable gap identified in the literature review is that the study focuses on using machine learning algorithms to forecast the development of lumpy skin disease (LSD) and cow foot and mouth disease (CFMD). As well the study did not given any information on the amount of data used and also no information on the pre-processing techniques used.

The purpose of the article (Zhou, et al., 2022) sought to forecast frequent diseases in dairy cows by employing machine learning algorithms and data gathered from automated monitoring systems. The goal of the research was to create a prediction model that could precisely anticipate when frequent diseases in dairy cows will manifest. Using a dataset of 14 dimensions, including variables like season, days in milking, parity, age at the time of illnesses, milk output, activity, rumination time, and milk conductivity, the study examined 131 sick and 149 healthy cows. The paper utilized eight ML algorithms to analyze the dataset, with the Rpart algorithm demonstrating strong generalization ability and outperforming other algorithms. The study's findings highlight the potential of ML algorithms in effectively predicting dairy cow disorders based on data from automated monitoring systems and milking systems. The use of ML algorithms offers a promising approach to early detection and diagnosis of health issues in dairy cows. The gap in the research lies in the reliance on changes in milk composition as the primary indicator for predicting specific diseases such as Foot-and-Mouth Disease (FMD) in dairy cows. While alterations in milk composition can serve as early indicators of health issues, they may not consistently correlate directly with specific diseases like FMD. Moreover, external factors such as diet and stress can significantly influence milk composition, potentially resulting in false positives. Therefore, depending solely on milk composition may not accurately identify FMD and other specific diseases in cattle.

On the other hand, Harsh J. Shah, Chirag Sharma and Chirag Joshi (Harsh, Chirag, & Chirag, 2022) aimed to surmount the ability of machine learning models to accurately classify and predict various diseases based on input symptoms by capturing images of key areas in cattle and comparing them with a massive dataset of ill and healthy cattle to decide whether the animal is healthy or needs to be quarantined to prevent further spread of infection. By incorporating advanced machine learning libraries such as TensorFlow and Keras, the proposed software system aims to provide an efficient and reliable means of identifying health issues in cattle, particularly focusing on disease prediction, with specific regard to the detection of Foot and Mouth Disease (FMD). The utilization of image capture and analysis through a setup of three strategically placed cameras - targeting the eye, the nose section, and the neck - represents an innovative step in leveraging machine learning for early disease detection, aligned with the unique anatomical signs indicative of FMD. The weakness of the research lies in its reliance solely on symptomatic data for disease detection. As emphasized, the exclusive dependence on visible symptoms poses a significant limitation, as certain animals may not consistently display clear signs of Foot and Mouth Disease (FMD), particularly during the early stages of the disease. This inherent variability in symptom manifestation introduces a potential risk of under diagnosis or delayed detection. Therefore, an approach based solely on symptomatic data may not adequately capture all instances of the disease, especially those characterized by subtle or inconsistent symptoms.

Lastly, (Rony, Barai, Riad, & Hasan, 2021) highlighted the important need of early diagnosis for disease prevention and mentioned that cattle are prone to highly contagious external illnesses as Infectious Bovine Keratoconjunctivitis (IBK), Lumpy Skin Disease (LSD), and Foot and Mouth Disease (FMD). While Convolutional Neural Networks (CNNs) are extensively used in image processing and computer vision, there is a notable absence of deep learning-based systems for detecting cattle diseases within husbandry farms. This study introduces a novel approach leveraging various CNN architectures including conventional deep CNN, Inception-V3, and VGG-16 to detect common external diseases in cattle. The paper meticulously outlines the steps involved, from data collection to processing and outcome analysis. Demonstrating promising results with 95% accuracy, the proposed system aims to mitigate human errors in disease identification, offering valuable support to veterinarians and farmers in disease recognition and control efforts.

2.6 SUMMARY OF RELATED WORKS

Table 2.1 *Summary of Related work*

Researcher/Author	**Contribution**	**Limitation**
(Akash, 2023)	He gave a thorough explanation of how a machine learning-based strategy was developed to identify Foot-and-Mouth Disease (FMD) and Lumpy Skin Disease (LSD) in cattle. This approach included the assessment of various machine learning algorithms and the determination that the artificial neural network (ANN) method was the most successful in detecting FMD.	Only used Randomized SearchCV for hyparameter tuning. No information given on amount of dataset used and pre-processing techniques used.
(Zhou, et al., 2022)	They proposed a machine learning algorithm that can accurately predict common disorders in dairy cows using data from automate monitoring systems.	Depending solely on milk composition may not accurately identify FMD and other specific diseases in cattle.
(Harsh, Chirag, & Chirag, 2022)	They developed a novel software system that utilizes machine learning models and image analysis to accurately identify health issues in	They focused on three parts (eye, nose and neck) which are not major parts to detect the disease.

	cattle, with a specific focus on the early detection of Foot and Mouth Disease (FMD) using strategically placed cameras to capture key anatomical signs.	
(Rony, Barai, Riad, & Hasan, 2021)	By leveraging deep learning techniques, specifically Convolutional Neural Networks (CNNs), the research introduces a novel approach to disease detection within husbandry farms. The proposed model, utilizing various CNN architectures, demonstrates a remarkable 95% accuracy rate in identifying these common cattle ailments.	It does not give any information about the pre-processing technique other than resizing image. Only batch size used to determine the model performance.

Overall, given the evidence presented in the literature, it is clear that FMD is a pressing issue that requires attention and intervention. The literature also underscores the potential of machine learning as a valuable tool for FMD detection in cattle in Ethiopia. By harnessing the wealth of available data and leveraging advanced analytical techniques, the development of accurate and reliable machine learning models holds great promise for enhancing FMD surveillance and control efforts, ultimately contributing to the protection of livestock health and the sustainability of Ethiopia's agricultural sector. This research aims to improve the precision and robustness of illness identification by including a wide range of data points into the analysis. This will improve the system's ability to accurately identify and discriminate FMD. In order to overcome the diversity and complexity of symptoms, particularly in the early stages of infection, this multimodal method shows promise. This will open the door to a more thorough, precise, and timely identification of FMD in animal populations.

CHAPTER THREE

METHODOLOGY

3.1 CHAPTER OVERVIEW

The precise techniques for identifying Foot and Mouth are covered in this chapter. It outlines the actions that must be taken in order to carry out the prediction again, so that it can be enhanced or compared to a different approach by another researcher. Several deep learning architectures (VGG-16, Inception and Densenet) were subjected to transfer learning, and an evaluation that focused on predictive power and performance indicators was conducted. Figure 3.1 illustrates the steps required from image acquisition to its evaluation.

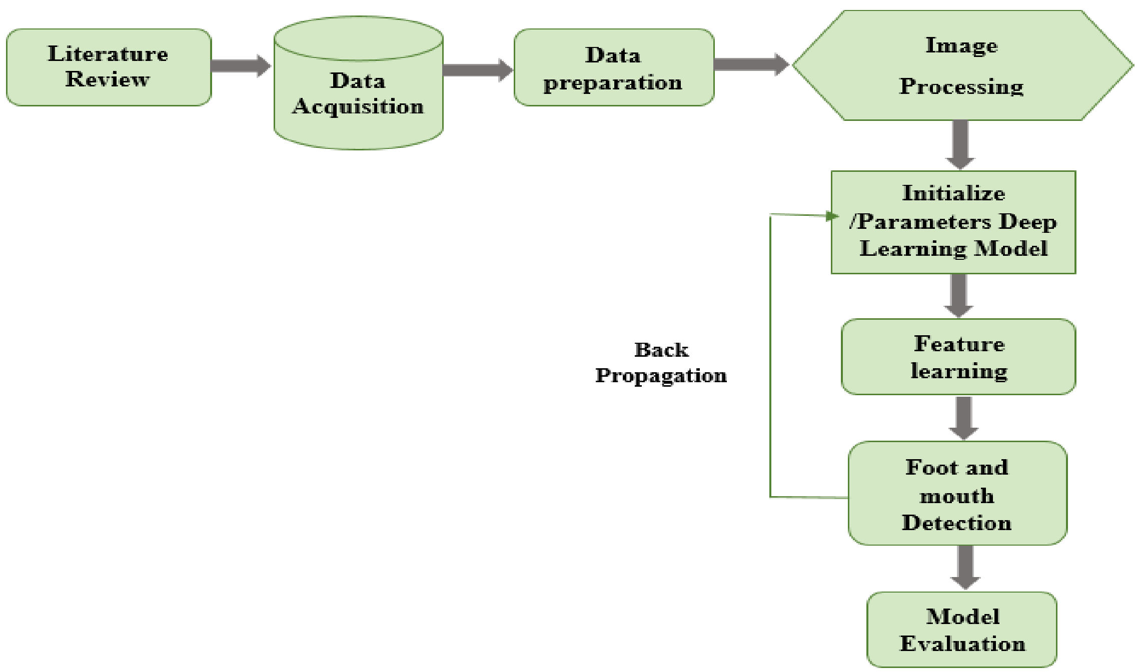


Figure 3.1 *Research Methodology*

3.2 LITERATURE REVIEW

The examination of related literature (Internet, Books, Journals, etc.) will provide insight into image processing and CNN's use in solving linked problems in previous research. This study aim will be achieved by a review of existing literature. During the literature review, gaps are identified in the prior solution to inform our suggested solution. This section includes a thorough analysis of all relevant research references and journals.

The researcher used the internet, books, and journals from Research Gate, Google Scholar, IEEE, and Elsevier to collect content that was not widely available locally. The article discusses deep learning architectures as well as existing methods for identifying foot and mouth. There was also a need to educate veterinary practitioners about foot and mouth disease. The information supplemented the literature review. Journals and books gave useful information on sickness categorization and the most effective approach for identifying foot and mouth.

3.3 DATA ACQUISITION

Image data for this study were collected from multiple sources to ensure a comprehensive dataset. The majority of the images were obtained from the Ministry of Agriculture (MoA) and the College of Veterinary Medicine in Bishoftu, Ethiopia. The MoA collected data from different regions in Ethiopia, primarily Amhara, Tigray, Oromia, and SNNP. These images represent a wide range of FMD lesions observed in field conditions. The College of Veterinary Medicine provided data that was gathered from their own teaching and research activities. These images include a variety of FMD lesions, as well as images of healthy cattles for comparison. From both areas around 3300 images were collected, of which 2000 of it is infected with FMD.

Due of limited local data availability, the study used online databases to detect Foot-and-Mouth Disease (FMD). Given that the symptoms of FMD are comparable across locations, combining locally gathered data with internet sources was considered suitable. This method provided a larger and more thorough dataset, allowing for a more rigorous analysis and increasing the generalizability of the findings. By combining data from multiple sources, we are able to gather a sufficient number of images which is around 7300 images to train and evaluate our deep learning models effectively.

Online datasets relevance and closeness to my original locally obtained data before merging them was carefully evaluated. In order to assess how much the main variables and distributions in the original and supplemental datasets overlapped and were comparable, statistical tests were done. Additionally, I examined the effects of the online datasets inclusion on the combined dataset's general properties, including modifications to the variables' mean, variance, and distribution. In order to find any notable shifts or changes, the distributions of the important variables before and after combining the datasets were examined. To conclude there was no much effect when combining the datasets.

3.4 DATA PREPARATION

In this study, the datasets are divided into three categories: training, validation, and testing. Images in training are classified into two categories: infected images and healthy images, which allows the model to learn to distinguish between sick and healthy states. To ensure a thorough evaluation of the model's performance, the dataset is categorized in three distinct ways for experimental reasons. In the first classification technique, 70% of the images are used to train the model, with 15% left aside for validation and testing. Of the images taken in the second classification technique, 80% are used for training and the remaining 20% are used for validation and testing. 90% of the photos in the final classification method are used for training, while 10% are used for validation and testing. The study uses these three different categorization techniques to evaluate the model's performance with different ratios of training, validation, and testing data. This will help to shed light on how dataset allocation affects the model's ability to detect diseases.

3.5 IMAGE PRE-PROCESSING

3.5.1 Region of Interest

A Region of Interest (ROI) is a portion of an image that is visually picked from a window displaying that picture and where you wish to focus your image analysis. This region can be used to direct future processing.

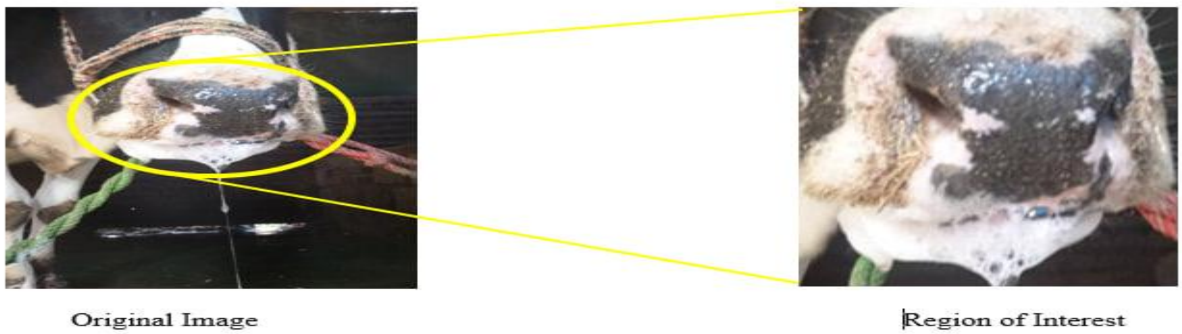


Figure 3.2 *ROI selection*

Source: Author

3.5.2 Image Augmentation

Data augmentation is a technique that allows us to create enormous datasets from tiny ones. We may not have sufficient data for our investigation due to several causes. Deep learning is more successful with huge data sets. Adding effects like cropping, flipping and rotating at different angles can help achieve this. This augmentation procedure generates a large number of alternative data samples from a single data collection. In most image classification studies, the dataset is extremely huge. Following data augmentation techniques, the dataset was significantly enriched, resulting in an expanded collection of approximately 15,000 images.

Data augmentation involves creating additional training data points from the original RGB dataset. It is crucial to enhance the amount of datasets. This allows the network to learn complicated characteristics from data without overfitting. This study applied several data augmentation techniques to the original image collection.

3.5.3 Resizing Images

The image is downsized to allow for the usage of several photos without losing crucial visual elements. Our gathered dataset consists of arbitrary sized RGB images. Neural network models typically assume a square input picture, which is then reshaped to 224x224 pixels with a consistent aspect ratio. Remove low-frequency background noise and normalize intensity.

Image processing included manually cropping each image to a square around the region of interest.

3.6 EVALUATION METHODS

To evaluate the effectiveness and performance of machine learning models, evaluation techniques are crucial. These approaches cover a range of methodologies and metrics intended to assess a model's precision in generalizing to unknown inputs. We use evaluation methods listed below.

3.6.1 Confusion Matrix

One useful method for evaluating the effectiveness of classification models is a confusion matrix. It provides a detailed examination of how the actual labels in the data relate to the predictions made by the model. The confusion matrix is often shown as a square matrix, where cases in a projected class are represented by each column, and examples from an actual class are represented by each row. The matrix's diagonal members show cases that have been successfully classified, whereas the off-diagonal components show examples that have been mistakenly classified. The computation of many performance metrics, including as accuracy, precision, recall, and F1-score, is made possible by the confusion matrix. These metrics provide a comprehensive understanding of the model's advantages and disadvantages, enabling practitioners and researchers to make informed decisions on the modification and choice of models.

Table 3.1 *Confusion matrix*

	(Predicted)	(Predicted)
(True)	**True Positive(TP)**	**False Positive(FP)**
(True)	**False Negative(FN)**	**True Negative(TN)**

There are four possible outcomes.

- True positive (TP): This happens when a positive class instance is accurately predicted by the model to be positive.
- False positive (FP): This happens when a negative class instance is mistakenly predicted as positive by the model.
- True negative (TN): This happens when a positive class instance is mistakenly predicted as negative by the model.
- False negative (FN): predicted to be negative but the actual value is positive

These four results offer information about how well the model performs and may be utilized to compute a number of evaluation approaches, including F1-score, accuracy, precision, and recall.

3.6.2 Evaluation Metrics

- **Accuracy**

The most common metric used to evaluate the effectiveness of a classification model is accuracy. It determines what proportion of all instances are accurately classified. Divide the total number of occurrences by the number of correctly detected instances, then multiply the result by 100 to determine accuracy.

$$\textbf{Classification Accuracy} = \frac{\text{No of samples predicted correctly}}{\text{Total No of sample}}$$

$$\textbf{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

- **Precision**

The percentage of true positives—that is, correctly detected positive cases—among all the positive predictions the model produces is referred to as precision. The calculation involves dividing the total number of true positives and false positives by the number of real positives.

$$\textbf{Precision} = \frac{\text{TP}}{\text{Total True predictions}} = \frac{\text{TP}}{\text{TP+FP}}$$

- **Recall**

Recall is the fraction of genuine positives among all real positive incidents. It is calculated by dividing the total number of false negatives and true positives by the number of true positives.

Recall $=\frac{\text{TP}}{\text{Actual True}} = \frac{\text{TP}}{\text{TP+FN}}$

- **F1 Score**

Accuracy and recall are combined to provide the F1 Score statistic. The harmonic mean of accuracy and recall is used to compute it. Because it takes into account both precision and recall, the F1 Score is a more reliable measure than accuracy.

F1 Score = 2X $\frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$

3.6.3 Receiver Operating Curve

The Receiver Operating Characteristic (ROC) curve is a statistical relationship that assesses the effectiveness of binary classification models. The trade-off between sensitivity and specificity at different thresholds is shown graphically. The true positive rate (TPR) and false positive rate (FPR) are compared to construct the ROC curve. The percentage of correctly identified positive cases is known as the TPR, whereas the percentage of incorrectly classified negative cases is known as the FPR. The ROC curve demonstrates the inherent trade-off between sensitivity and specificity, since increasing sensitivity by traveling along the curve from left to right reduces specificity and vice versa.

3.7 DEVELOPMENT TOOLS

To design and implement the thesis work, a variety of development tools are employed. Prototyping development platforms, UML modeling tools, and more research-related tools are among the tools. The different development tools are briefly explained in the sections that follow. This research use UML to develop a suggested model and create figures in several areas. The study document's figures were created using the UML tool.

3.7.1 Design Tools

Design tools enable the creation, presentation, and interpretation of design concepts. The system is designed mostly using Enterprise Architect. This tool creates professional-looking flowcharts, network diagrams, and other design concepts while being lightweight and powerful.

3.7.2 Hardware Tools

To execute machine learning and deep learning on any dataset, the software/program requires a computer system strong enough to manage the computational power required. The following hardware considerations were made for this study shown in Table 3.2.

Table 3.2 *Hardware Tools*

No.	Tools	Specification	Used for
1	GPU	NVIDIA GeForce GT 740M	To enhance computation and speed up training.
2	Hard Drive(HDD)	WD 1TB USB 3.0 Gen 4	Used to store vast amounts of data.
3	RAM	8GB RAM	For the computer speed and performance.

3.7.3 Software Tools

Various writing software and coding tools will be utilized to accomplish the study through coding, and their explanations are provided below.

Table 3.3 *Software Tools*

No.	Tools	Version	Specification
1	Python	3.11.3	Python is a high-level, general-purpose programming language with an abundance of libraries and frameworks that make coding easier and save development time.
2	Scikit-learn	0.32.2	It is an open-source software library in Python that includes machine learning methods.
3	Anaconda	2020.11	It is a free and open-source Python distribution that primarily focuses on machine learning and data science applications. The distribution includes a Python interpreter and various packages.
4	Jupyter Notebook	6.1.4	It is an open-source web application that includes coding and real-time visualization.
5	TensorFlow	2.16.1	It is an end-to-end open-source, flexible ecosystem of machine learning platform tools.
6	Keras	3.3.3	It is an open-source framework that offers a Python interface to neural networks and acts as a link between Python and TensorFlow.

CHAPTER FOUR

PROPOSED SOLUTION

4.1 CHAPTER OVERVIEW

This chapter focuses on the proposal's design and experimental parameters. The suggested model's construction, feature extraction, and classification utilizing the training from scratch approach are briefly presented.

4.2 PRE-PROCESSING

The suggested model's preprocessing step follows the generation of a digital image. The digitized picture is examined for skewing and pre-processed to remove noise. FMD detection systems require pre-processing to decrease background noise, highlight the region of focus, and distinguish between foreground and background. Python's openCV package was utilized for pre-processing. In this study, we employ certain preprocessing approaches. We use several procedures on the dataset, including scaling, converting to grayscale, normalizing pixel values, and data augmentation.

4.2.1 Resize Image

The image is downsized to allow for the usage of several photos without losing crucial visual elements. Our gathered dataset consists of arbitrary sized RGB images. Neural network models typically assume a square input picture, which is then reshaped to 224x224 pixels with a consistent aspect ratio. Remove low-frequency background noise and normalize intensity. Image processing included manually cropping each image to a square around the region of interest.

4.2.2 RGB to Grayscale Conversion

Grayscale is the simplest color model, using just one component to specify colors which is lightness. Brightness is defined as a value between 0 (black) and 255 (white). The collection's images are all in RGB color format. Comparing grayscale to RGB color photographs reduces computational complexity. Grayscale input photographs are necessary for our proposed model, hence the source images must be converted to this format. This was followed by applying additional preprocessing steps like soft focus to enhance specific aspects of the image. This sequential approach allows for the extraction of relevant features while reducing computational complexity.

4.2.3 Normalization

This image processing procedure changes the range of pixel intensity values. It converts a picture into pixel values. "Normalization" is the process of changing photographs to a standard size. This approach is employed at the data pre-processing step. For data sets with varying ranges, numerical values are converted to a common scale of [0,1]. The input image is divided by 255. The output values will vary from 0 to 1. Normalization reduces calculations, speeds up forward propagation, and ensures correct results.

4.3 MODEL SELECTION

After a review of the literature on computer vision, particularly image classification, CNN, a deep learning technique, was selected. CNNs are one possible approach to processing adaptive images. The method trains, tests, and assesses the accuracy of the model in addition to extracting and classifying information. CNNs are capable of handling data without the requirement for additional feature extraction or pre-processing steps. Moreover, a unified framework unifies feature extraction and categorization.

Convolutional Neural Networks (CNNs) were selected for FMD detection due to their ability to automatically learn hierarchical features from image data, making them more automated than classical machine learning algorithms. Transfer learning involves applying previously trained models to a specific case. In this study, we used models from the "ImageNet" contests and shared by the Keras library, including VGG16, Densenet201, and Inception V3.

The selection of VGG16, InceptionV3, and DenseNet201 for the identification of foot and mouth disease (FMD) is supported by the advantages of their respective architectures, track records of effectiveness, and task-specific fit. The simplicity and depth of VGG16—which uses 16 weight layers with tiny convolution filters for simpler implementation and fine-tuning—make it popular. It is a dependable option for FMD detection due to its potent feature extraction capabilities, which have proven to perform well in a variety of image classification applications, including medical imaging. Using inception modules that collect multi-scale elements in images—a critical component for distinguishing the various FMD manifestations—InceptionV3 is chosen for its efficiency and depth. It can be trained with little computer resources as it drastically cuts the number of parameters while keeping a deep structure. DenseNet201 is distinguished by its dense connection, in which all layers are interconnected to maximize information flow and enhance feature reuse. Because of its great parameter efficiency, this design is perfect for smaller datasets, which FMD data frequently contain. It also reduces overfitting. The capacity of DenseNet201 to capture the fine-grained information required for identifying FMD symptoms is further demonstrated by its state-of-the-art performance in a variety of picture classification problems. Together, the strengths of these models—the depth and simplicity of VGG16, the multi-scale feature extraction of InceptionV3, and the dense connectivity and parameter efficiency of DenseNet201—provide a strong and dependable FMD detection method. Together, these models provide a thorough method for identifying a variety of patterns and characteristics in FMD pictures, which eventually improves detection accuracy.

To determine the most suitable CNN architecture for our task, we compared the accuracy of three different models: VGG16, Densenet201, and Inception V3. Each of these models has its own unique strengths and weaknesses. Inception is known for its ability to capture multi-scale features, VGG-16 is a classic CNN architecture that has been widely used for image classification tasks and DenseNet is known for its dense connectivity, which allows for efficient feature reuse. By comparing the accuracy of these different models, we were able to select the best model for our FMD detection task.

4.4 MODEL DESIGN

A. VGG16 Architecture

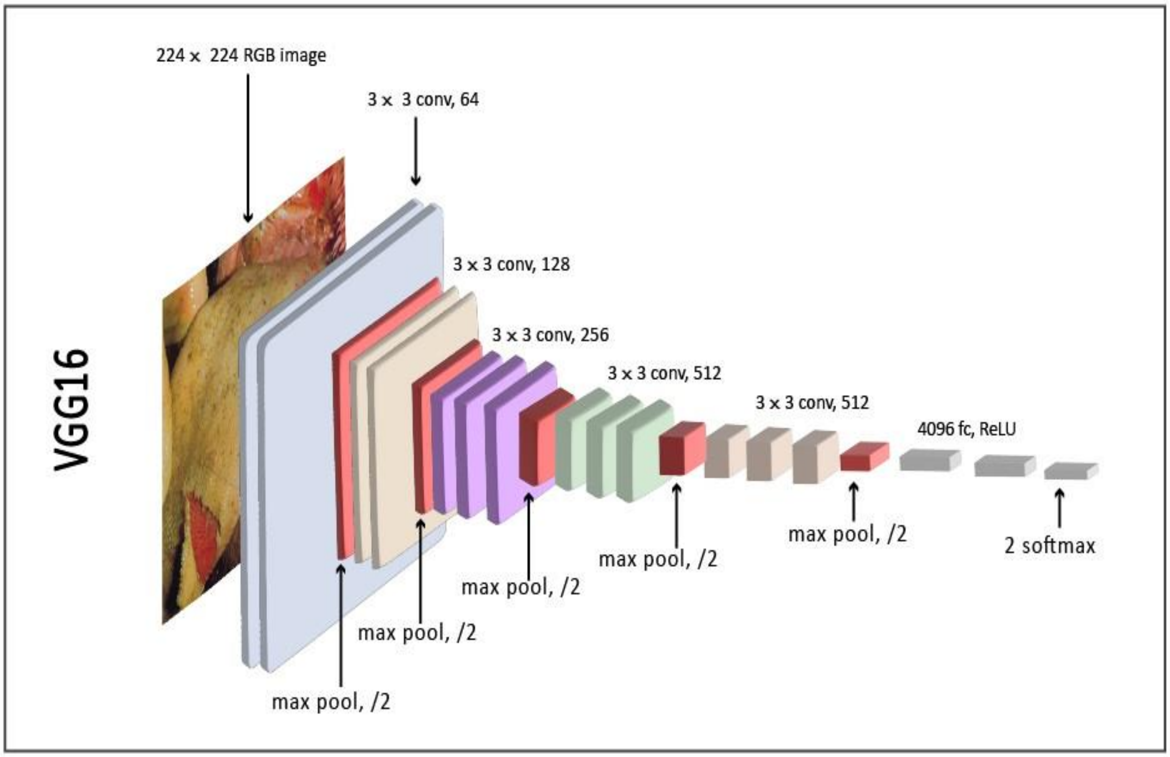


Figure 4.1 *VGG16 architecture*

Source: Author

In the provided VGG16 architecture, there are a total of 16 layers, including convolutional layers, pooling layers, and fully connected layers.

There are 13 convolutional layers, each followed by a rectified linear unit (ReLU) activation function. These layers perform feature extraction by convolving input images with learnable filters to detect patterns and features at different scales and complexities.

There are 5 max-pooling layers, each applied after every two consecutive convolutional layers. Max-pooling layers down sample the feature maps, reducing their spatial dimensions while retaining the most relevant information.

There is 1 fully connected layer with 4096 units and ReLU activation. This layer, commonly referred to as a dense layer, learns high-level features from the flattened feature maps generated by the preceding convolutional and pooling layers. This layer allows the network to learn complex patterns and relationships in the data, leading to better classification performance.

The input image size is 224 × 224 x 3, which represents the input image's RGB channels. The first two convolutional layers consist of 3 x 3 convolutions with 64 filters each. The convolutional layers apply filters to extract features from the input image. Following each pair of convolutional layers, a max-pooling layer with a 2 x 2 window size reduces the spatial dimensions of the feature maps. The next two convolutional layers have 3 x 3 convolutions with 128 filters each. These layers further extract higher-level features from the input. Similar to the previous layers, max-pooling with a 2 x 2 window size is applied after the convolutional layers. Three consecutive convolutional layers with 3 x 3 convolutions and 256 filters each follow the pattern of feature extraction. Max pooling with a 2 x 2 window size is performed after the set of convolutional layers. Another set of three convolutional layers with 3 x 3 convolutions and 512 filters each for deeper feature learning. Max pooling with a 2 x 2 window size is applied to reduce spatial dimensions. Three more convolutional layers with 3 x 3 convolutions and 512 filters each for further feature extraction. Max pooling with a 2 x 2 window size is performed to downsample the feature maps.

The layer known as Global Average Pooling combines features from the whole feature map while reducing the spatial dimensions to 1 x 1. There is a completely connected layer with 4096 neurons and a ReLU activation mechanism that follows after the Global Average Pooling layer. The class probabilities for "FMD" and "Healthy" are provided by the last layer using the Softmax activation function.

These layers collectively form the architecture of the VGG16 model, enabling it to extract hierarchical features from input images and make predictions based on the learned features.

B. InceptionV3 Architecture

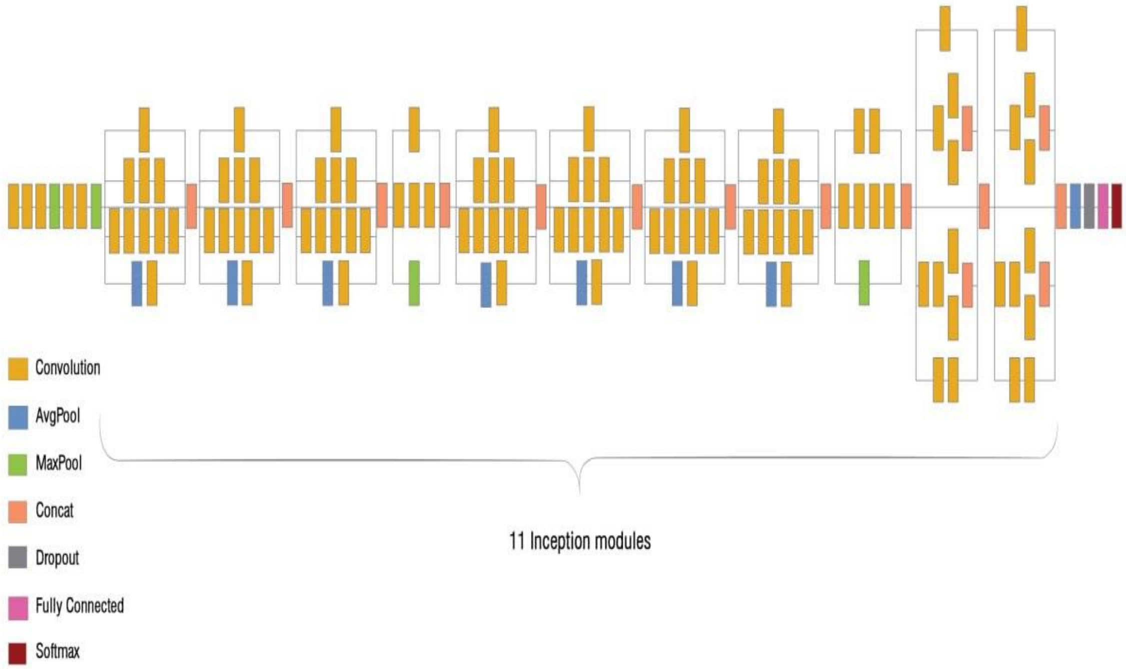


Figure 4.2 *Inception V3 architecture*

Source: Author

The input image size is 229 x 229 x 3, representing the dimensions of the input image with RGB channels.

Inception V3 consists of five convolutional layers that extract features from the input image using various filters to capture different levels of information. The model includes two max-pooling layers that downsample the feature maps to reduce spatial dimensions and extract dominant features. Inception V3 comprises 11 inception modules, which are building blocks with multiple parallel convolutional operations of different sizes to capture diverse features. Following the inception modules, there is an average pooling layer that aggregates features globally to further condense information.

A fully connected layer with 2048 neurons is added to the model after the convolutional and pooling layers in order to enable it to learn high-level characteristics unique to the dataset. To provide a normalized distribution for classification, the last layer uses the Softmax activation function to generate class probabilities for the two output classes.

C. DenseNet201 Architecture

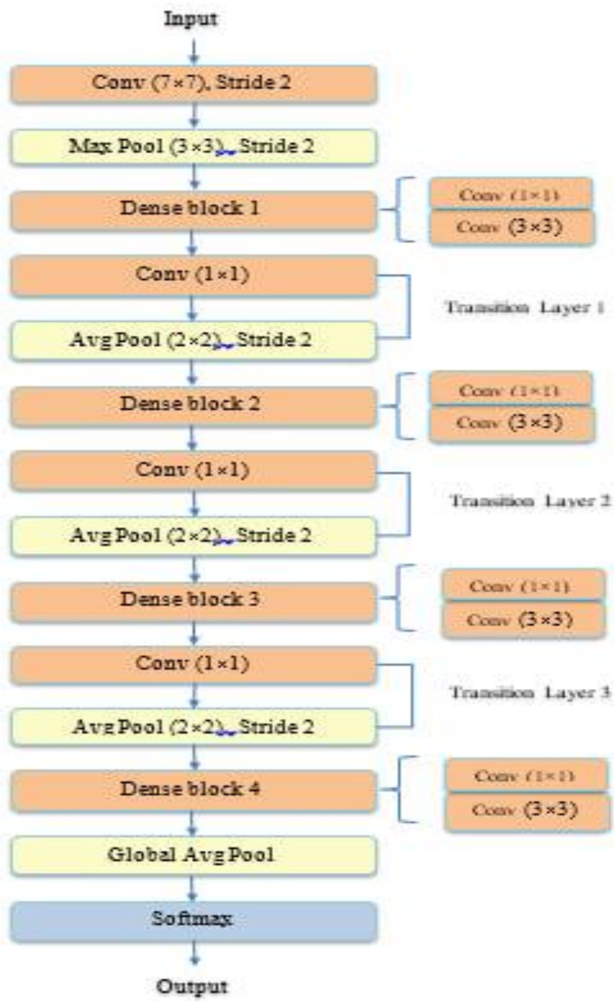


Figure 4.3 *Densenet architecture*

Source: Author

The first convolutional layer has a 7x7 kernel with a stride of 2 and 64 filters, followed by a 3x3 max pooling layer with a stride of 2. This aids in extracting fundamental characteristics from input photos and reducing the spatial dimensions of the data. The architecture thus consists of four desne blocks, each with a unique number of layers. The first dense block contains six layers; the second has twelve, the third has twenty-four, and the fourth has sixteen. Each layer in the dense blocks has a 3x3 kernel with a stride of 1 and a number of filters that rises with the block's depth. Following each dense block, there is an average pooling layer with a 2x2 kernel and stride of 2. This helps to downsample the feature maps and enhance the amount of features extracted. Transition layers are used to minimize the spatial dimensions of feature maps while increasing the amount of features extracted. Each transition layer is made up of a 1x1 convolutional layer with a stride of 1, followed by a 2x2 average pooling layer with a stride of 2. Finally, a global average pooling layer averages the feature maps across spatial dimensions, followed by a fully linked layer using a softmax activation function. This layer produces a probability distribution over the classes.

4.5 FEATURE EXTRACTION

The first stage in classifying images is feature extraction. The CNN algorithm collects crucial characteristics for picture classification. This phase extracts and defines several aspects of the image, including its height, horizontal lines, widths, circles, pixels, and vertical arcs. Moreover, it is the procedure used to separate the most discriminating information from unprocessed data.

It identifies the precise and unique factors that characterize an image's form in tiny dimensions. The purpose of is to extract a collection of characteristics that enhance acceptance rates with fewer occurrences and provide a consistent feature set across diverse pictures. During this stage, key characteristics are extracted from photos to create feature vectors. A fully connected network classifies input photos based on feature vectors.

4.5.1 Algorithm for Optimization

To reduce errors, the model is trained with a gradient descent optimization approach. Weights are then updated using the back-propagation process. Gradient descent is the most often utilized optimization approach in deep learning research. It is a widely used optimization technique for fine-tuning pre-trained models. Modern Deep Learning libraries, like Keras and Tensorflow, include gradient descent optimization methods. In order to minimize the loss function, it optimizes the model's weight and parameters. Gradient descent is optimized via Adaptive Moment Estimation (Adam). When training deep neural networks, Adam is the most used optimization method for figuring out adaptive learning rates. This method is easy to implement, computationally efficient, and useful for managing huge datasets and parameters. The algorithm adjusts the learning rate using squared gradients and the gradient's moving average.

4.5.2 Activation Function

Different forms of activation functions affect our neuron's output, including sigmoid, tanh, softmax, and Relu (rectified linear unit). The model's output layer uses softmax activation, and ReLU is best suited for convolution and fully linked layers.

4.5.3 L1 Regularization

In order to prevent overfitting in machine learning models, L1 regularization, sometimes referred to as Lasso regularization, adds a penalty term to the loss function that is dependent on the absolute value of the model's coefficients. The main idea underlying L1 regularization is to encourage model sparsity by setting some of the coefficients to absolutely zero. This efficiently executes feature selection by identifying and retaining just the most significant traits while eliminating irrelevant ones.

Mathematically, L1 regularization adds a term proportional to the weights' absolute values to the loss function:

$$L_{new}(w) = L_{original}(w) + \lambda \sum_{i=1}^{m} |wi| \tag{4.1}$$

where lambda is the regularization strength hyperparameter that determines the trade-off between minimizing the original loss and minimizing the absolute weights.

By reducing this regularized loss function during training, the model learns weights that strike a compromise between minimizing the original loss and keeping the weights relatively modest. This promotes several weights to go absolutely zero, thus removing their respective characteristics from the model.

4.5.4 Epochs

One thorough run of the full dataset during the training phase of a machine learning model is referred to as an epoch. In other words, the model iterates over all of the training instances once during a single epoch, modifying its parameters (weights and biases) in response to the mistakes that are seen in order to minimize the loss function. The model was trained using distinct epochs for each architecture. Determine the number of epochs required for the model's peak performance. This was done to determine the best possible optimum epoch.

4.5.5 Batch Size

The number of training examples used in a single training cycle is referred to as the batch size. Training is usually done on batches of data rather than adjusting the model's parameters after processing each individual training example (which would be computationally inefficient). The amount of samples processed prior to the model's parameters being changed is determined by the batch size. The 32-batch size and 64-batch size was utilized to train this model. Small batch sizes are optimal for great performance, but they incur large computational costs.

4.5.6 Learning Rate

In a machine learning model, the size of parameter updates during training is determined by the learning rate. It defines how much weight to update during backpropagation. Choosing the optimum learning rate during the experiment was the most difficult part. Lower learning rates require more training time than higher ones. Two learning rates, 0.001 and 0.01 are examined in the experiment to assess how they affected the model's performance and training dynamics.

CHAPTER FIVE

IMPLEMENTATION OF THE PROPOSD SOLUTION

5.1 CHAPTER OVERVIEW

The process of implementation is covered in this chapter. This thorough tutorial covers every aspect, including training and data processing as well as workspace setup. Code samples from the implementation are also provided and thoroughly examined.

5.2 WORKING ENVIRONMENTS

The working environment includes both hardware and software facilities that is used for this work.

5.2.1 Environmental Setup

Tensorflow: This machine learning platform allows for model development and deployment in client contexts, similar to real-world applications. TensorFlow serves as the backend for Keras, a toolkit that trains and models deep neural network (DNN) classifiers for prediction on testing sets. The version of Tensorflow used in this work is 2.16.1.

Keras: The objective is to make deep learning model training easier for research and development by employing models that are well-suited to deep learning. Keras is a high-level neural networks API written in Python, capable of running on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit. It allows for easy and fast experimentation with deep learning models. The module is currently placed in TensorFlow. The framework is easy to use, adaptable, and flexible. Keras' current functioning version is 3.3.3.

Python: - Python programming with anaconda cooperation will use for the implementation of the study. The python version is 3.11.3.

Jupyter Notebook: Interactive notebook papers with real-time code, equations, graphics, video, and other computational outputs can be created with Jupyter Notebook.

5.3 PROPOSED SOLUTION IMPLEMENTATION

5.3.1 Implementation for Data Preprocessing

5.3.1.1 *Image Data Augmentation*

Data augmentation is creating new data variants from old data in order to increase its volume. This can be accomplished by making tiny modifications to the original data or training a model to generate new data. Image augmentation is increasingly used in deep learning models due to the need for huge datasets and improved model performance. The model's speed can be improved by applying it to existing photographs rather than adding new ones. This allows the model to learn picture variety and imitate real-world data.

```
# Create an ImageDataGenerator object
datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Snippet Code 5.1 *Data Augmentation Code*

The ImageDataGenerator class has a number of parameters that can be used to control the type and amount of augmentation that is applied to the data. In the code snippet above, the following parameters are used:

• **rotation_range**: The range of rotation degrees that will be applied to the images is specified by this parameter. When the value is 40 degrees, the rotation of the photos will be random, ranging from -40 degrees to 40 degrees.

• **width_shift_range:** The range of horizontal shifts that will be applied to the images is specified by this parameter. When the value is 0.2, the pictures will be randomly displaced by a percentage ranging from -20% to 20% of their width.

• **height_shift_range**: The range of vertical shifts that will be applied to the images is specified by this parameter. When the number is 0.2, the pictures will be randomly displaced by a percentage ranging from -20% to 20% of their original height.

• **shear_range**: The range of shear angles that will be applied to the images is specified by this option. When the parameter is 0.2, a random angle between -20 and 20 degrees will be used to shear the photos.

• **zoom_range**: The range of zoom factors that will be applied to the images is specified by this option. A value of 0.2 indicates that a random factor between 0.8 and 1.2 will be used to zoom the images.

• **horizontal_flip**: Whether or whether the images will be randomly flipped horizontally is indicated by this option. A value of True means that the images will be flipped horizontally with a 50% probability.

The ImageDataGenerator class can be used to generate a variety of different types of augmented data. The types of augmentation that are generated depend on the parameters that are specified. In the code snippet above, the ImageDataGenerator class is used to generate a batch of augmented images that have been rotated, shifted, sheared, zoomed, and flipped horizontally. The augmented images are then used to train the model. This helps to improve the model's performance on unseen data by exposing it to a wider variety of data.

5.3.1.2 **Rescale**

```
# Create a Rescaling layer to normalize the RGB channels
normalization_layer = tf.keras.layers.Rescaling(1./255)

# Apply the normalization layer to the training dataset
normalized_ds = train_ds.map(lambda x, y: (normalization_layer(x), y))
```

Snippet Code 5.2 *Normalization Code*

This code snippet creates a Rescaling layer that normalizes the RGB channels of the images in the training dataset. The Rescaling layer scales the pixel values in each channel by 1/255.0. This ensures that all of the pixel values are between 0 and 1. The map() function is used to apply the normalization layer to each element in the training dataset. The map() function takes a lambda function as its argument. The lambda function takes two arguments: the input data (x) and the target data (y). The lambda function applies the normalization layer to the input data and returns the normalized data and the target data.

5.3.2 Model Implementation

The research examines Keras applications with pre-trained weights for deep learning models. Three models were implemented based on empirical results, as shown below.

A. VGG16 Implementation

Snippet Code 5.3's code snippet imports a VGG16 model that has already been pre-trained on the ImageNet dataset, omitting the classification layers. In order to reduce spatial dimensions, a Global Average Pooling layer is implemented. Next, a 4096-unit fully connected layer with ReLU activation is added for feature learning. For group classification, a final Dense layer with Softmax activation is implemented. The Adam optimizer was utilized in the construction of this model, along with an accuracy metric and a categorical cross-entropy loss function.

By averaging the values in each channel over the whole image, the global average pooling layer lowers the dimensionality of the input data. As a result, the input data is compressed from 224 × 224 x 3 to 1 x 1 x 3. A complicated connection between the input data and the output labels is learned by the fully connected layer. The probability distribution across the output classes is calculated using the softmax layer. The model will predict one of the two classes because the softmax layer has two output classes.

The categorical cross entropy loss function and the Adam optimizer are used to construct the model. One well-liked optimizer for neural network training is the Adam optimizer. One popular loss function for classification issues is the categorical cross entropy loss function.

```
# Load the pre-trained VGG16 model
vgg_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Add a global average pooling layer
x = vgg_model.output
x1 = GlobalAveragePooling2D()(x)

# Add a fully connected layer
x2 = Dense(4096, activation='relu')(x1)

# Add a softmax layer for classification
predictions = Dense(2, activation='softmax')(x2)

# Create the model
vggmodel = Model(inputs=vgg_model.input, outputs=predictions)

# Compile the model
vggmodel.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])
```

Snippet Code 5.3 *VGG16 Implementation*

B. InceptionV3 Implementation

```
# Load the pre-trained InceptionV3 model
inc_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Add a global average pooling layer
y = inc_model.output
y1 = GlobalAveragePooling2D()(y)

# Add a fully connected layer
y2 = Dense(2048, activation='relu')(y1)

# Add a softmax layer for classification
predictions = Dense(2, activation='softmax')(y2)

# Create the model
incmodel = Model(inputs=inc_model.input, outputs=predictions)

# Compile the model
incmodel.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])
```

Snippet Code 5.4 *Inception V3 Implementation*

C. Densnet201 Implementation

```
# Load the pre-trained DenseNet201 model
dense_model = DenseNet201(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Add a global average pooling layer
z = dense_model.output
z1 = GlobalAveragePooling2D()(z1)

# Add a fully connected layer
z2 = Dense(2048, activation='relu')(z1)

# Add a softmax layer for classification
predictions = Dense(2, activation='softmax')(z2)

# Create the model
densemodel = Model(inputs=dense_model.input, outputs=predictions)

# Compile the model
densemodel.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])
```

Snippet Code 5.5 *Densnet201 Implementation*

5.3.3 L1 regularization implementation

```
l1_regularizer = regularizers.L1(l1=0.01)  # Adjust the regularization strength 

for layer in vgg_model.layers:
    if hasattr(layer, 'kernel_regularizer'):
        layer.kernel_regularizer = l1_regularizer

for layer in inc_model.layers:
    if hasattr(layer, 'kernel_regularizer'):
        layer.kernel_regularizer = l1_regularizer

for layer in dense_model.layers:
    if hasattr(layer, 'kernel_regularizer'):
        layer.kernel_regularizer = l1_regularizer
```

Snippet Code 5.6 *L1 Regularization Implementation*

5.3.4 Early stopping

```
# Split data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.05, random_s

# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weig

# Train the models with early stopping
fin_vgg16 = vggmodel.fit(X_train, y_train, epochs=100, batch_size=32, validation_
fin_inc = incmodel.fit(X_train, y_train, epochs=100, batch_size=32, validation_da
fin_dense = densemodel.fit(X_train, y_train, epochs=100, batch_size=32, validati
```

Snippet Code 5.7 *Early Stopping Implementation*

5.3.5 Hyper-parameter Tuning

Hyperparameters are those that control other parameters. Hyper-parameter tuning is the process of optimizing values to achieve the best results. Proper parameter selection is crucial for a model's performance. The AUTOTUNE API allows for more flexible and efficient input pipelines. Additional hyperparameter tunings were also applied to additional parameters.

```
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

Snippet Code 5.8 *Hyper-parameter Tuning Code*

CHAPTER SIX

RESULTS AND DISCUSSION

6.1 CHAPTER OVERVIEW

The experimental results for this study are described in depth in this chapter, and a commentary follows. The findings of the experiments, which include InceptionV3, VGG16, and Densenet201, are provided in detail. Image conversion is also provided.

6.2 EXPERIMENTAL RESULTS

The performance of each model from the experimental investigation is described in depth in this section. Before going into the experimental findings for each model, let's have a look at how an RGB image is converted to grayscale and what happens when we apply soft focus.

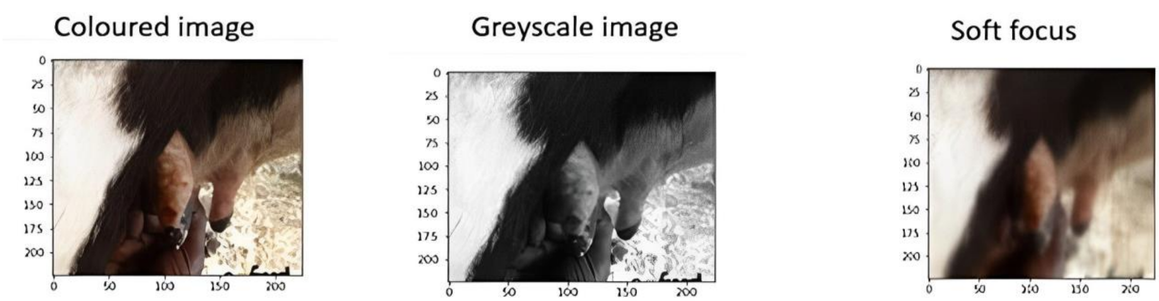


Figure 6.1 *Color Conversion*

6.2.1 Accuracy and Loss Performance

Most popular libraries in Python as Tensor Flow and Keras were used to calculate the accuracies for the mentioned CNN architecture models. Training and validation accuracy were measured for different epochs and same batch size as stated in Table 6.1 below. Figures 6.2, 6.4 and 6.6 shows the accuracy of the models and Figures 6.3, 6.5 and 6.7 shows the loss of the performance of the models.

Table 6.1 *Performance of each models*

Models	Batch Size	Minimum Training Accuracy		Minimum Validation Accuracy		Maximum Training Accuracy		Maximum Validation Accuracy		Overall Test Accuracy
		Epoch	Accuracy (%)	Epoch	Accuracy (%)	Epoch	Accuracy (%)	Epoch	Accuracy (%)	
VGG16	32	1	73.55	1	66.25	28	93.95	22	94.55	81.69%
InceptionV3	32	1	71.22	1	84.75	12	99.20	14	95.05	96.87%
Densnet201	32	1	77.82	13	92.00	16	99.55	20	99.00	98.87%

6.2.1.1 ***Vgg16 result***

Overall test accuracy for the performance is 81.69%. At epoch 1, the minimum training accuracy is 73.55%, whereas the minimum validation accuracy is 66.25%. The greatest validation accuracy is 94.55% at epoch 22, whereas the best training accuracy is 93.95% at epoch 28. An estimate of the overall model loss is 0.2892.

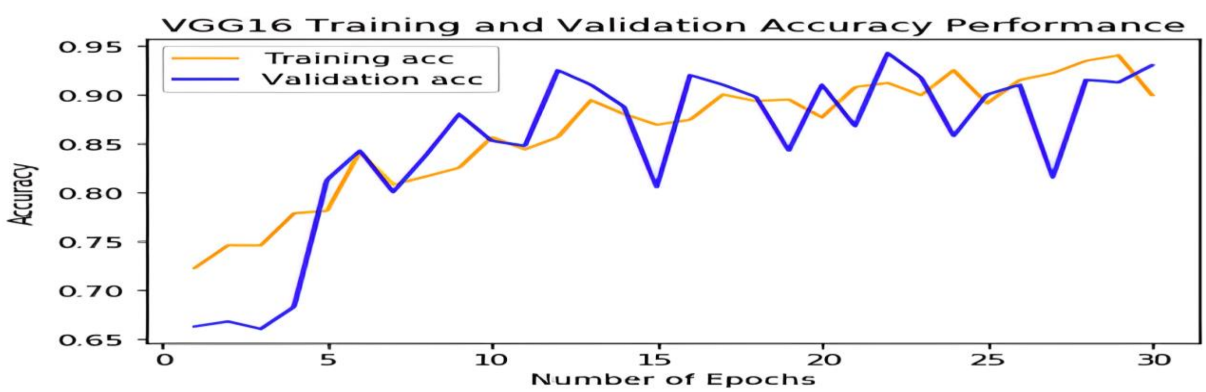


Figure 6.2 *VGG16 Training and Validation Accuracy Performance*

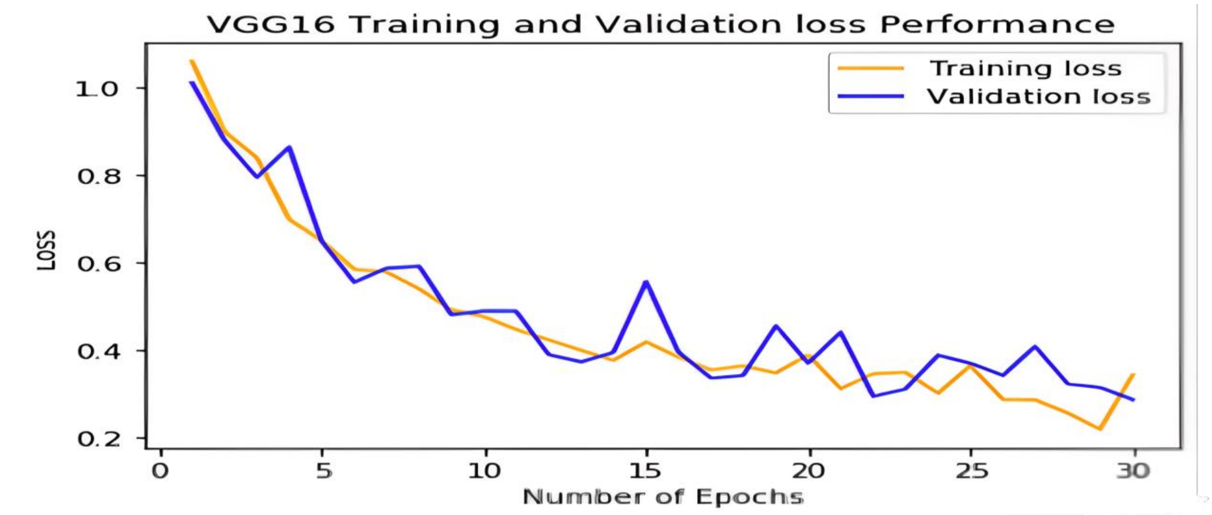


Figure 6.3 *VGG16 Training and Validation Loss Performance*

6.2.1.2 ***InceptionV3 Result***

The performance's total test accuracy is 96.87%. At epoch 1, the least training accuracy is 71.22%, while the minimum validation accuracy is 84.75%. The maximum training accuracy is 99.20% at epoch 12, whereas the maximum validation accuracy is 95.05% at epoch 14. An estimate of the overall model loss is 0.274.

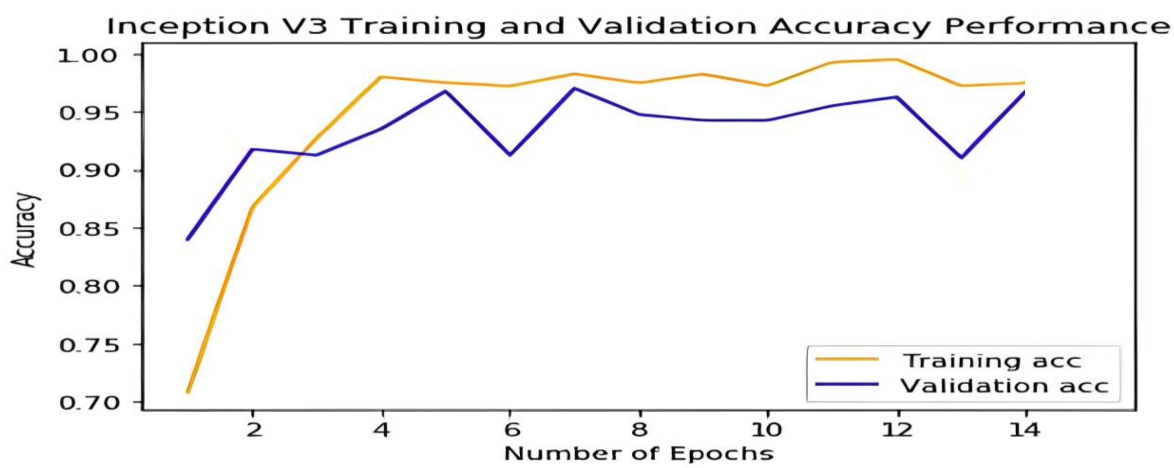


Figure 6.4 *Inception V3 Training and Validation Accuracy Performance*

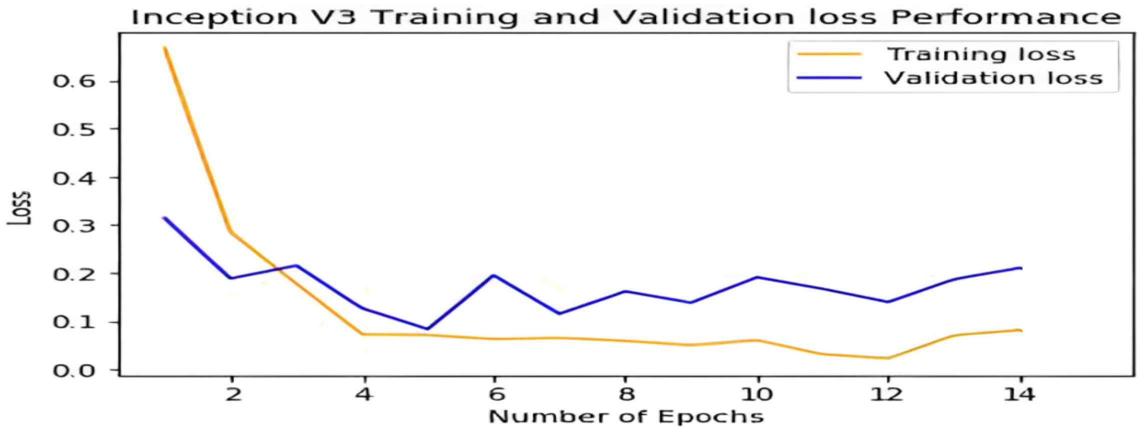


Figure 6.5 *Inception V3 Training and Validation Loss Performance*

6.2.1.3 ***Densenet201 Result***

The performance has a 98.87% test accuracy overall. The lowest training accuracy is 77.82% at epoch 1, while the lowest validation accuracy is 92.00% at epoch 13. The maximum training accuracy is 99.55% at epoch 16, whereas the maximum validation accuracy is 99.00% at epoch 20. An estimate of the overall model loss is 0.224.

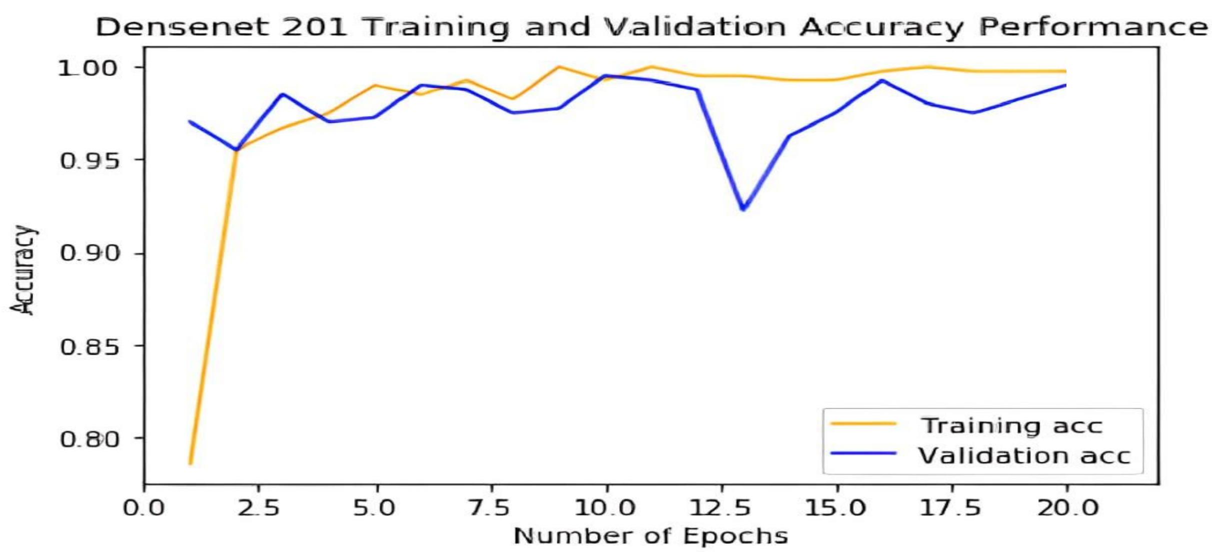


Figure 6.6 *Densnet201 Training and Validation Accuracy Performance*

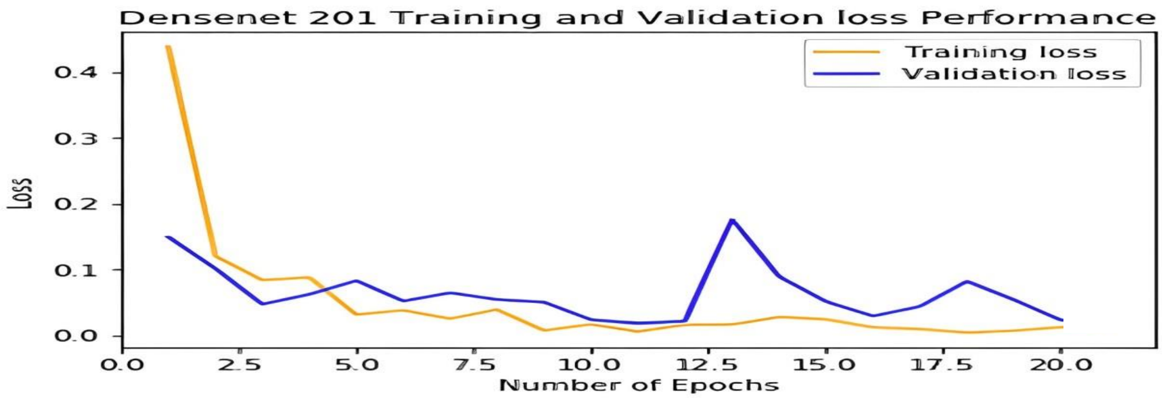
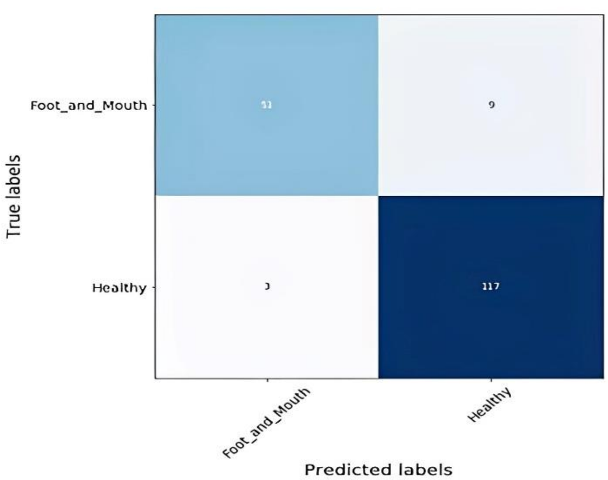


Figure 6.7 *Densnet201 Training and Validation Loss Performance*

6.2.2 Evaluation Matrix Result

Based on the accuracy and loss performance, Densnet201 has the highest accuracy. Next to Densnet201 InceptionV3 has the highest while VGG16 got a low accuracy. Here we compare the models using evaluation metrics as accuracy, precision, recall and f1-score.

6.2.2.1 *VGG16 Classification Report Result*



VGG16 Classification Report

	precision	recall	f1-score	support
Foot_and_Mouth	0.94	0.85	0.89	60
Healthy	0.93	0.97	0.95	120
accuracy			0.93	180
macro avg	0.94	0.91	0.92	180
weighted avg	0.93	0.93	0.93	180

Figure 6.8 *VGG16 Classification result*

Table 6.2VGG16 Confusion Matrix

Model	TP	FP	FN	TN
VGG16	51	9	3	117

6.2.2.2 *InceptionV3 Classification Report Result*

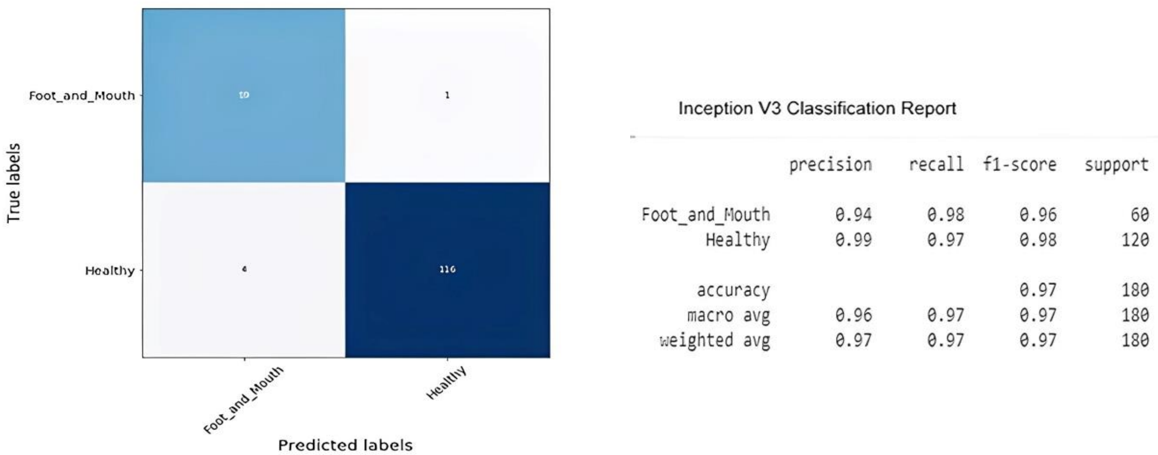
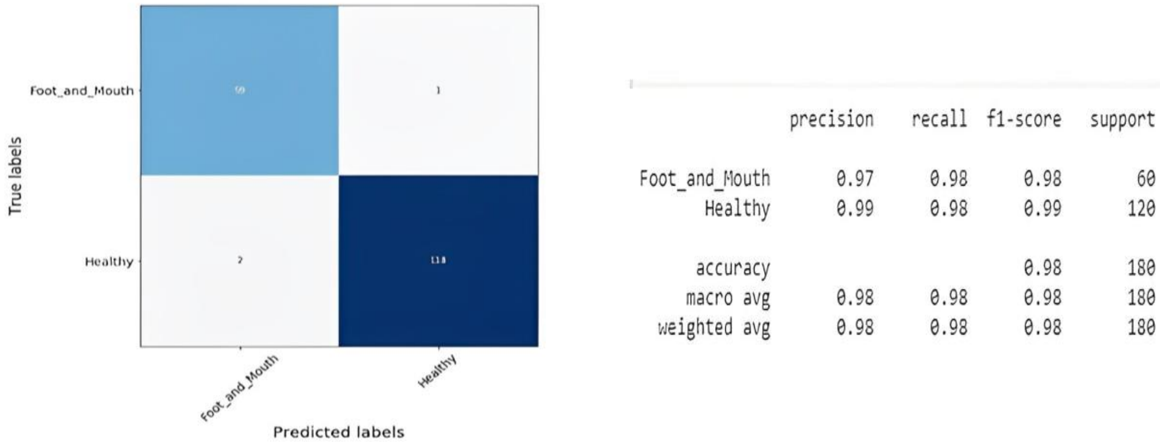


Figure 6.9 *InceptionV3 Classification result*

Table 6.3 *VGG16 Confusion Matrix*

Model	TP	FP	FN	TN
InceptionV3	59	1	4	116

6.2.2.3 ***Densnenet201 Classification Report Result***



	precision	recall	f1-score	support
Foot_and_Mouth	0.97	0.98	0.98	60
Healthy	0.99	0.98	0.99	120
accuracy			0.98	180
macro avg	0.98	0.98	0.98	180
weighted avg	0.98	0.98	0.98	180

Figure 6.10 *Densnet201 Classification Report*

Table 6.4 *Densenet201 Confusion Matrix*

Model	TP	FP	FN	TN
Densenet201	59	1	2	118

Densnet201 has the highest accuracy with 98%. It had highest accuracy on the accuracy performance and minimum loss on loss performance. On the test of evaluation matrix as well Densnet201 has highest accuracy.

6.2.3 ROC and AUC Graphs

When assessing a binary classifier's performance at different probability thresholds, the Receiver Operating Curve (ROC) is most frequently utilized. At each probability threshold, the ROC measures the proportion of true positives to erroneous negatives. The classification were determined using the area under the ROC curve.

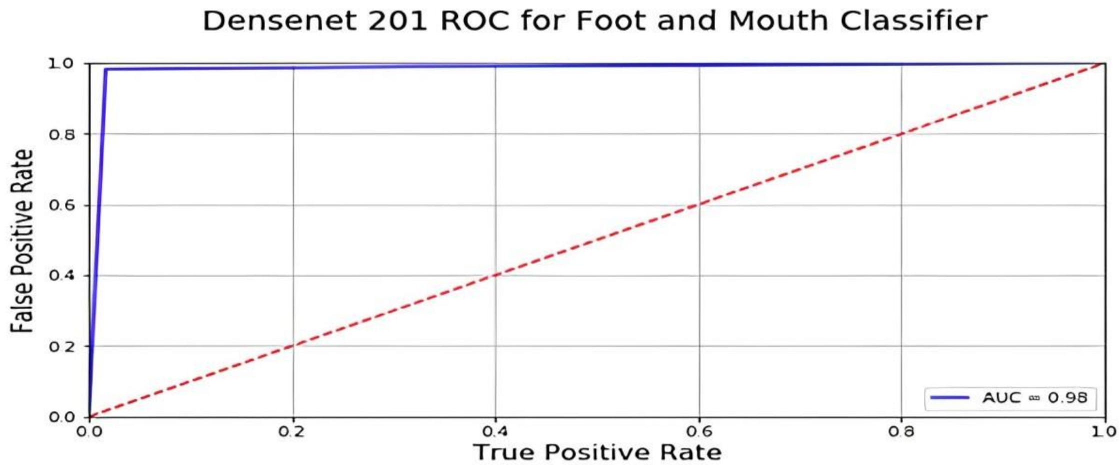


Figure 6.11 *Densnet201 ROC*

Densnet201 has showed a good rate when using ROC graph. We can now generalize by stating that Densnet201 has a better accuracy and outperforms other deep learning models compared in this study.

6.2.4 Result of experiment done on rgb images

In the study, the original images were initially in RGB format but were subsequently converted to grayscale, and soft focus techniques were applied for experimental purposes. Throughout the experimentation, a diverse array of image types was systematically employed to comprehensively evaluate performance. RGB images consistently demonstrated higher accuracy compared to their grayscale counterparts and those manipulated with soft focus. This discrepancy underscores the pivotal role of color information in enhancing the efficacy of image analysis and underscores the significance of utilizing RGB images for optimal results.

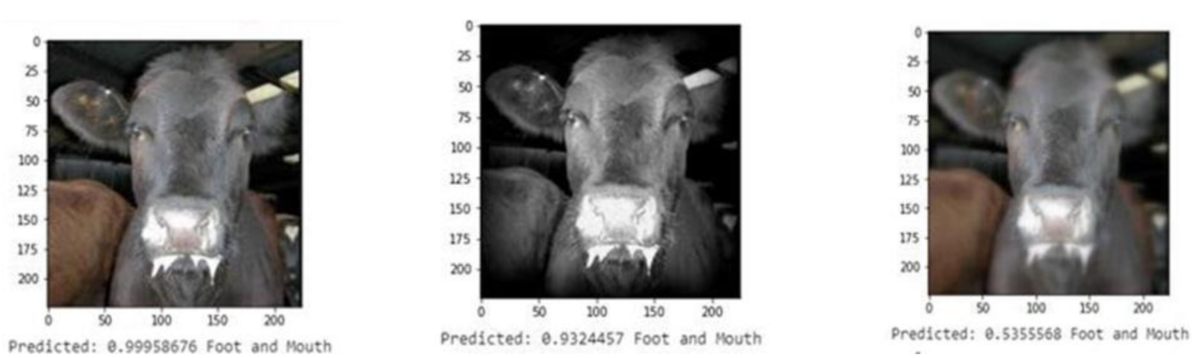


Figure 6.12 *Prediction result on RGB, gray scale and soft focus images*

6.2.5 Experimental Result Using Different Dataset Spliting Ratios

In the experimental setup, the dataset underwent three different splitting configurations for comparative analysis. The original dataset of 7300 images was augmented to 15000 images, with augmentation applied solely to the training data before splitting. The dataset was then divided into three sets using varying ratios: 80% training, 20% validation; 70% training, 30% validation; and 90% training, 10% validation. Each split aimed to assess the impact of different training-validation proportions on model performance. The 80%-20% split allocated the largest portion to training, followed by a 70%-30% split with a slightly smaller training set, and a 90%-10% split with the smallest training set. By comparing the performance of models trained on these different splits, insights are gained into the influence of training-validation data distribution on the effectiveness of the models in handling the task at hand.

The total original dataset consists of 7300 images. To increase the size and diversity of the training data, data augmentation techniques are applied, resulting in a total of 15000 images. It is important to note that the augmentation process is performed only on the training data, and not on the validation or test sets. When splitting the dataset, an 80:10:10 ratio is used, where 80% of the data is allocated for training, 10% for validation, and the remaining 10% for testing. Applying this ratio to the augmented dataset of 15000 images, the training directory ended up with 12000 images, while both the validation and test directories contained 1500 images each.

For the second comparison, a 70:15:15 ratio was used, where 70% of the data was allocated for training, 15% for validation, and the remaining 15% for testing. Applying this ratio to the augmented dataset of 15000 images, the training directory ended up with 10500 images, while both the validation and test directories contained 2250 images each.

Lastly, a 90:5:5 ratio, where 90% of the data was allocated for training, 5% for validation, and 5% for testing. This means that the training directory will contain 13500 images, the validation directory will contain 750 images, and the test directory will contain 750 images. This approach ensures that the validation and test sets remain representative of the original, unaugmented data, allowing for an unbiased evaluation of the model's performance on new, unseen data.

Table 6.5Experimental Result using different split ratios

Models	Training Samples	Validation and Test samples	Loss	Accuracy
VGG16	70%	30%	0.56	84.15%
	80%	20%	0.2892	93.95%
	90%	10%	0.324	89.99%
InceptionV3	70%	30%	0.311	92.32%
	80%	20%	0.274	99.20%
	90%	10%	0.299	97.35%
Densenet201	70%	30%	0.29	93.00%
	80%	20%	0.224	99.55%
	90%	10%	0.243	98.88%

6.2.6 Experimental Result Using Different Activation Function and Learning Rate

The learning rate determines the magnitude of parameter updates during training in a machine learning model. A low number might cause a stalled training process, whereas a high value can lead to learning suboptimal weights too soon or instability. The learning rate is a hyper parameter used to train neural networks. It has a moderate positive value, often between 0.0 and 1.0. The experiment was conducted using learning rates of 0.01 and 0.001.

Table 6.6 *Experiment using different learning rates*

Models	Learning Rate	Accuracy		Loss	
		Training	**Validation**	**Training**	**Validation**
VGG16	0.01	86.74%	84.33%	20%	20.6%
	0.001	90.17%	89.10%	17.9%	13.3%
InceptionV3	0.01	94.55%	93.13%	18.66%	13.4%

	0.001	98.45%	97.17%	4.9%	8.3%
Densenet201	0.01	96.54%	94.33%	9.66%	13.4%
	0.001	99.61%	99.10%	2.9%	4.3%

6.2.7 Experimental Result Using Batch Size and Activation Function

Batch size is an important hyperparameter in deep learning models since it controls the amount of samples utilized to update the model's parameters during training. A bigger batch size can result in more stable training and better convergence, but it also raises computing demands and memory consumption. For this study, we will examine our model's performance across two batch sizes: 16 and 32. This will enable us to assess the effect of batch size on the model's accuracy and training time.

Softmax is a method that converts non-normalized data into probability distributions for output classes. The sigmoid function is a common activation function in neural network output layers for binary classification applications. It assures that the output is a probability between 0 and 1, which may be translated as the chance of the input belonging to the positive class. For this study softmax activation function and sigmoid activation function are compared.

Table 6.7 *Experimental result using batch size and activation function*

Models	**Optimizer**	**Batch Size**	**Activation Function**	**Accuracy**		**Loss**
				Training	**Validation**	
VGG16	ADAM	16	Sigmoid	92.95	93.00	0.34
			Softmax	93.05	93.55	0.299
		32	Sigmoid	93.99	94.87	0.265
			Softmax	93.95	94.55	0.2892
InceptionV3	ADAM	16	Sigmoid	98.25%	93.25%	0.285
			Softmax	98.26%	93.17%	0.324
		32	Sigmoid	99.25%	95.75%	0.263

			Softmax	99.20%	95.05%	0.274
Densenet201	ADAM	16	Sigmoid	98.88%	97.13%	0.293
			Softmax	98.83%	98.01%	0.284
		32	Sigmoid	99.57%	99.06%	0.22
			Softmax	99.55%	99.00%	0.224

6.2.8 Result Using Machine Learning Models

The study by (Akash, 2023) employed the following methodology: A substantial dataset comprising images of both infected and healthy cattle was collected. Subsequently, the data underwent preprocessing to ensure optimal quality and consistency. Feature extraction techniques were then applied to extract relevant information from the images. Following this, a model was trained utilizing the extracted features and subsequently evaluated for its performance. In order to fine-tune the model's hyperparameters, Randomized SearchCV was employed, enabling an efficient exploration of the hyperparameter space. This approach facilitated the identification of the most suitable hyperparameter configurations, thereby enhancing the model's overall effectiveness and robustness.

Table 6.8 *Result using machine learning models*

Models	**Accuracy**
Logistic Regression	73.5%
Support Vector Machine	81.6%
Descion Tree	65.3%
Random Forests	85.0%
AdaBoost	53.3%
Baging	86.4%
XGBoost	74.8%
Artificial Neural Network	97.4%

While these standard approaches achieved respectable accuracy, they fell short of capturing the complex visual patterns and subtleties associated with FMD. In contrast, among the deep learning models evaluated in this current study, DenseNet201, consistently outperformed the classical machine learning algorithms reported in the Akash (2023) benchmark.

6.3 RESEARCH QUESTION DISCUSSION

Answer for RQ1: Detecting Foot and Mouth Disease (FMD) in animals is critical for avoiding its spread and reducing its impact on agricultural economies. Convolutional Neural Network (CNN) architectures have emerged as useful tools in this sector, thanks to their capacity to automatically learn discriminative features from images, making them ideal for illness diagnosis. In this paper, the effectiveness of multiple CNN architectures is evaluated, including VGG16, InceptionV3, and DenseNet201, for detecting FMD. These designs range in complexity and feature extraction capabilities, allowing us to compare their performance at different levels of model sophistication. The first step of our inquiry comprised data gathering, which was a difficult task. Next substantial preprocessing is performed on the acquired images, including techniques such as scaling, color conversion, normalization and other methods. These preprocessing methods were critical for improving the quality and consistency of the input data, allowing for more effective learning by CNN models. Using these cutting-edge designs and preprocessing approaches, we hoped to create robust and accurate FMD detection systems with potential applications in this study.

Answer for RQ2: The following hyperparameters have a substantial influence on CNN models' performance in identifying Foot and Mouth Disease (FMD): learning rate, batch size, optimizer, epochs, and activation function. The learning rate, or how rapidly the model learns from the data, was discovered to have a considerable influence on the model's performance. A low learning rate of 0.001 proved to be more effective than a greater learning rate of 0.01. The batch size, which determines the amount of samples required to update the model's parameters, was also discovered to be significant. A batch size of 32 proved to be more successful than a batch size of 16.

The optimizer, which controls how the model changes its parameters during training, was found to be critical in producing successful results. The ADAM optimizer was employed in this investigation, and it has already been proved to be successful in other cattle disease studies. The number of epochs, which governs how many times the model views the whole dataset during training, was also discovered to be significant. Finally, the activation function, which defines the model's output, was discovered to have a considerable effect on its performance. Both sigmoid and softmax activation functions were utilized, and the results were almost identical, with sigmoid yielding slightly higher values. To prevent overfitting, we employed early stopping, a technique that monitors the validation loss and stops training when the loss stops improving. By using early stopping, we were able to train our models to an optimal number of epochs, avoiding both underfitting and overfitting. The findings of this study highlight the importance of carefully tuning the hyperparameters of CNN models for optimal performance in FMD detection. By experimenting with different learning rates, batch sizes, and regularization techniques, we were able to develop efficient and accurate CNN models that can reliably identify FMD cases. These insights can inform the development of future CNN-based systems for FMD detection, contributing to improved disease surveillance and control efforts.

Answer for RQ3: The findings of this study demonstrate the better performance of deep learning, particularly Convolutional Neural Network (CNN) models, in detecting Foot-and-Mouth Disease (FMD) in cattle compared to traditional machine learning approaches. This is evidenced by the benchmark provided in the study by (Akash, 2023), which utilized classical machine learning algorithms for cattle disease detection. In the (Akash, 2023) study, the researchers used standard machine learning approaches including Support Vector Machines (SVMs), Random Forests, and Logistic Regression to categorize several cattle illnesses, including FMD. While these standard approaches achieved respectable accuracy, they fell short of capturing the complex visual patterns and subtleties associated with FMD. In contrast, the deep learning models evaluated in this current study, including VGG16, InceptionV3, and DenseNet201, consistently outperformed the classical machine learning algorithms reported in the Akash (2023) benchmark. The CNN architectures demonstrated significantly higher accuracy in correctly identifying the cases, showcasing their superior ability to learn discriminative features directly from the image data. The hierarchical structure of CNNs, with multiple convolutional and pooling layers, allows these models to capture intricate visual cues and patterns indicative of FMD at different scales. This is a key advantage over traditional machine learning approaches, which often rely on manually engineered features that may not fully capture the complexity of the disease manifestation in cattle.

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

In conclusion, this study focused on the detection of Foot and Mouth Disease (FMD) using pre-trained models of Convolutional Neural Networks (CNNs). The aim of this research was to develop reliable FMD detection systems with possible uses. To that end, data was gathered and images were preprocessed utilizing methods like scaling, color conversion, and normalization to increase data quality and consistency. While previous research mainly utilized classic machine learning methods, only a few incorporated deep learning techniques such as CNNs. Moreover, most existing studies only detected FMD on specific parts of the animal's body, neglecting the fact that FMD symptoms, such as lesions and ruptures, primarily occur on the tongue, gum, mouth, teat, and hooves. In contrast, this study trained the model to detect FMD based most body parts where lesions are commonly found including other symptoms as drooling. According to the study, a number of hyperparameters significantly affected how well CNN models identified Foot and Mouth Disease (FMD). It was shown that a learning rate of 0.001 was more efficient than a learning rate of 0.01. It was found that a batch size of 32 worked better than a batch size of 16. This study used the ADAM optimizer, which has been shown to be successful in previous research on cow diseases. The activation functions sigmoid and softmax were both used, and the results were about the same, with sigmoid producing somewhat higher values. The results show that deep learning—and CNN models in particular—performs better than conventional machine learning techniques at identifying FMD in cattle. This is demonstrated by the benchmark given in the research by Akash (2023), which used traditional machine learning methods to identify diseases in cattle. In order to improve disease monitoring and control in livestock, our study demonstrates the potential of cutting-edge CNN architectures like VGG16, InceptionV3, and DenseNet201 in developing precise and dependable FMD detection systems. By utilizing pre-trained models such as VGG16, Inception V3, and DenseNet201, this research achieved impressive results, including a training accuracy of 99.55%, a validation accuracy of 99.00%, and a test accuracy of 98.87%.

In summary, this study presented a novel approach to the detection of Foot and Mouth Disease by leveraging pre-trained CNN models. By considering all the body parts where FMD lesions predominantly occur, the model achieved remarkable accuracy in training, validation, and testing. These findings highlight the potential of deep learning techniques in accurately identifying FMD, which can contribute to early detection and prevention of the disease in livestock. Further research and validation on larger datasets can potentially enhance the performance and applicability of the proposed model in real-world scenarios.

7.2 FUTURE WORK

Although this thesis covers a wide range of topics, there are certain notions that might be enhanced or added. One important aspect is the expansion of the dataset used for detection. While image-based datasets provide valuable visual information, incorporating a text-based dataset alongside it could offer a more comprehensive understanding of FMD. This text dataset could include symptoms that are not visually evident, such as symptoms that cannot be seen with eyes, as well as additional information like vaccination history, climate conditions, and other features that may serve as indicators of FMD. By combining image and text datasets, researchers can potentially develop a more robust and accurate detection system, enabling early identification of FMD cases and facilitating effective preventive measures. This integrated approach could significantly improve the overall performance and reliability of the detection system for FMD.

REFERENCES

A Gentle Introduction to the Rectified Linear Unit (ReLU). (n.d.). Retrieved from Machine Learning Mastery.

Abdela, Nejash. (2017). Sero-prevalence, risk factors and distribution of foot and mouth disease in Ethiopia. *Acta Trop*.

Abdullah Azeem, Immad Rashid, Hassan Muhammad, Muhammad Asad, Ghazala Kaukab, Amna Tehseen, & Sabita . (2020). A review on foot and mouth disease in dairy animals, etiology, pathogenesis and clinical findings. *Pure and Applied Biology*.

Activation function. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Activation_function

Akash. (2023). CATTLE DISEASE DETECTION USING MACHINE . *International Research Journal of Modernization in Engineering Technology and Science*.

Alexanderson , S., Zhang, Z., Donaldson, A., & Garland, A. (2003). The pathogenesis and diagnosis of foot-and-mouth disease. *J Comp Pathol*, 1-36.

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. *Series of the Gesellschaft fur Informatik.* Lecture Notes in Informatics (LNI).

Built in. (2023).

Chollet, F. (2018). Deep Learning with Python.

Deepchecks. (2023). Retrieved from Densnenet.

Ekramul Hossain, Ashad Kabir, Lihong Zheng, & Dave, L. (2022). A Systematic Review of Machine Learning Techniques for Cattle Identification: Datasets, Methods and Future Directions. *Artificial Intelligence in Agriculture*.

Girma Zewdie, Mirtneh Akalu, Wondwossen Tolossa, Hassen Belay, Getaw , D., Mariamawit Zekarias, & Yeneneh Tesfaye. (2023). A review of foot-and-mouth disease in Ethiopia: epidemiological aspects, economic implications, and control strategies. *Virology Journal*.

Grubman Marvin, & Barry Baxt (2004). Foot-and-mouth disease. *Clin Microbiol Rev*.

Han Shi Chong, Guo Hui Chen, & Sun Shi Qi (2014). Three-dimensional structure of foot-and-mouth disease virus and its biological functions. *Archives of Virology*, 1-16.

Harsh J. Shah, Chirag Sharma, & Chirag Joshi. (2022). Cattle medical diagnosis and prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)*.

Hirsh Dwight. C., MacLachlan, N. James, & Walker, Richard. L. (2004). Vetrinary Microbiology. *Wiley-Blackwell.*

Hofstra Gerben, Van Abeelen, Duindam Marleen. Houbeen Bas, Kuijpers Joris, Arendsen Tim, & Petie, Ronald. (2023). Automated monitoring and detection of disease using a generic facial feature scoring system–A case study on FMD infected cows. *Preventive Veterinary Medicine*.

Huang, G., Zhuang, L., van der Maaten, L., & Weinberger, K. (2017). Densely Connected Convolutional Networks. *CVPR.*

I, D. A. (2007). Diseases of Sheep. *Edinburgh: Blackwell Publishing*.

IBM. (2024).

Imagenet. (2024). Retrieved from Wikipedia.

Infection. (2018). Contact Challenge of Cattle with Foot-and-Mouth Disease Virus Validates the Role of the Nasopharyngeal Epithelium as the Site of Primary and Persistent Infection. 1-18.

J Guitian, M Arnold, Y Chang, & E L Snary. (2023). Applications of machine learning in animal and veterinary public health surveillance. *Revue scientifique et technique (International Office of Epizootics)*.

Jemberu, W., Mourtis, M., Sahle, M., Siraw, B., Vernooij, J., & Hogeveen , H. (2016). Epidemiology of Foot and Mouth Disease in Ethiopia: a Retrospective Analysis of District Level Outbreaks 2007-2012. *Transbound Emerg Dis*.

Jennifer, M., Dinu, G., Yakir , M., Teddy, L., Anna, Z., & Amber, A. P. (2023). Machine learning approaches to predict and detect early-onset of digital dermatitis in dairy cows using sensor data. *Frontiers Veterinary Science.*

John, G., Ebert, K., Gubbin, S., Bashiruddin, J., & Paton , D. J. (2011). Normal variation in thermal radiated temperature in cattle: implications for foot-and-mouth disease detection. *BMC Veterinary Research*.

Lee June-Goo, Jun Sanghoon, Cho Young-Won., Lee Hyunna, Kim Guk Bae, Seo Joon Beom, & Kim Namkug. (2017). Deep Learning in Medical Imaging : General Overview. *PubMed*, 570-584.

Longjam, N., Deb, R., Sarmah, A., Awachat, V., & Saxena, V. (2011). A Brief Review on Diagnosis of Foot-and-Mouth Disease of Livestock: Conventional to Molecular Tools. *Vet Med Int.*

Machine Learning. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Machine_learning

Managwu Chidozie. (2022). APPLICATION OF OBJECT DETECTION IN THE BROWSER TO DETECT AND ISOLATE PLASTIC BOTTLES.

Mazi Toyal, Rakesh Bollepally, Suram Tahrun & Nimma Chandu (2023). A DEEP LEARNING METHOD TO IDENTIFY LUMPY SKIN. *International Research Journal of Modernization in Engineering Technology and Science* .

Medical Imaging Informatics: An Overview. (2019). In E. Seeram, *Digital Radiography* (pp. 165-183). Singapore: Springer.

Mohan, A., Raju, D. R., & Janarthanan, P. (2019). Animal Disease Diagnosis Expert System using Convolutional Neural Networks. *International Conference on Intelligent Sustainable Systems (ICISS).*

Namita. (2023). *Medium*.

Optimization Algorithms in Machine Learning. (n.d.). Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/optimization-algorithms-in-machine-learning/

Pei Yin, Jun Zhang, Han Yan, Jun Zaho & Jing Wang (2022). Privacy Risk Perception of Online Medical Community Users Based on Deep Neural Network. *Human-Media Interaction*.

Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinty, N., Mrisho, L., Ndalahwa, M., . . . P.Hughes, D. (2019). A mobile-based deep learning model for cassava disease diagnosis. *Frontiers in Plant Science*.

Rony, M., Barai, D., Riad, & Hasan, Z. (2021). Cattle External Disease Classification Using Deep Learning Techniques. *International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1-7.

Schaefer, A., Cook, N. J., Tessaro, S., & Deregt, D. (2004). Early detection and prediction of infection using infrared thermography. *The Canadian veterinary journal*.

Sharma, S., & Sharma, S. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*.

Shuwen , Z., Qiang , S., & Qin, C. (2020). Application of Machine Learning in Animal Disease Analysis and Prediction. *Current Bioinformatics*.

Sinkala , Y., Simuunza, M., Pfeiffer, D., Munang'andu, H., Mulumba, M., Kasanga, C., & Mweene, A. (2014). Challenges and economic implications in the control of foot and mouth disease in sub-saharan Africa: lessons from the zambian experience. *Vet Med Int*.

Smith, Matthew J. (2019). Getting value from artificial intelligence in agriculture. *Animal Production Science*, 46–54. doi: 10.1071/AN18522.

Tekleghiorghis, T., Moormann, R. J., Weerdmeester, k., & Dekker, A. (2016). Foot-and-mouth Disease Transmission in Africa: Implications for Control, a Review. *Transboundary and Emerging Diseases*, 136-151.

Tesfaye, A. B., Assefa, G. A., Shishaye, L. B., Abera, B. M., Gebreanenya, N. T., Gebregiorgis, G. L., & Dürr, S. (2023). Tesfaye, Outbreak investigation of foot-and-mouth disease in cattle in Tigray region, Northern Ethiopia. *Frontiers in Veterinary Science*.

The Global foot-and-mouth disease Control Strategy. (n.d.). Retrieved from Food and Agriculture Organization of the United Nations (FAO): https://www.fao.org/eufmd/global-situation/global-fmd-control-strategy/en/

Tm, P., Pranathi, A., SaiAshritha, K., Chittara, N. B., & Koolagudi, S. G. (2018). Tomato Leaf Disease Detection Using Convolutional Neural Networks. *Eleventh International Conference on Contemporary Computing (IC3)*.

VGG16 – Convolutional Network for Classification and Detection. (2018, November 20). Retrieved from Neurohive: https://neurohive.io/en/popular-networks/vgg16/

Wallelign, S. (2017). Soybean Plant Disease Identification Using Convolutional Neural Network. 146-151.

Wani et al. (2020). Basics of Supervised Deep Learning.

What Is Transfer Learning? Definition, Methods, and Applications. (n.d.). Retrieved from Spice works.

WOAH. (2024). Retrieved from Foot and Mouth Disease: https://www.woah.org/en/disease/foot-and-mouth-disease/

Woldemariyam Fanos, Joseph Kamu, Kris Clercq & David Lefebre (2023). Epidemiological Dynamics of Foot-and-Mouth Disease in the Horn of Africa: The Role of Virus Diversity and Animal Movement. *Viruses*.

Wubshet Ashenafi, Dai Junfei, Li Qian, & Zhang Jie (2019). Review on Outbreak Dynamics, the Endemic Serotypes, and Diversified Topotypic Profiles of Foot and Mouth Disease Virus Isolates in Ethiopia from 2008 to 2018. *Viruses*.

Zhou Xiaojing, Xu Chuang, Wang Hao, Xu Wei, Zhao Zixuan, Chen, M., & Huang, Bin. (2022). The Early Prediction of Common Disorders in Dairy Cows Monitored by Automatic Systems with Machine Learning Algorithms. *Animals*.

APPENDIXES

Appendix I: Importing Libraries

```
import tensorflow as tf
import os
import tensorflow.keras as keras
from tensorflow.keras.applications import VGG16, InceptionV3, DenseNet201
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Input
from keras.metrics import categorical_crossentropy, categorical_accuracy
from tensorflow.keras.models import Model
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l1
from keras.applications import imagenet_utils
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, TensorBoar
from sklearn.metrics import accuracy_score, confusion_matrix, roc_curve, auc
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.models import Sequential
```

Appendix II: Reading Images

```
train_dir = 'fmdimages/train_data'
validation_dir = 'fmdimages/validation_data'
test_dir = 'fmdimages/test_data'
```

Appendix III: Data Augmentation

```
# Create an ImageDataGenerator object
datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

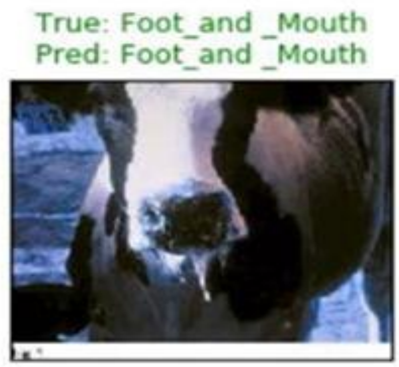
Appendix IV: Rescaling

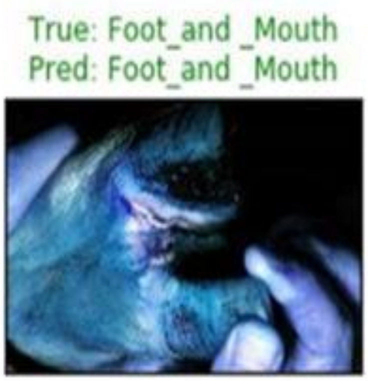
```
# Create a Rescaling layer to normalize the RGB channels
normalization_layer = tf.keras.layers.Rescaling(1./255)

# Apply the normalization layer to the training dataset
normalized_ds = train_ds.map(lambda x, y: (normalization_layer(x), y))
```

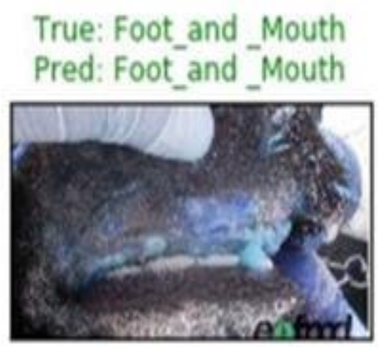
Appendix V: Batch Testing

```
def batch_testing(model, images, true_labels, class_names):
    predictions = model.predict(images)
    num_images = len(images)

    plt.figure(figsize=(15, 10))
    for i in range(num_images):
        plt.subplot(3, 4, i+1)
        plt.imshow(images[i])
        plt.axis('off')
        true_label = class_names[true_labels[i]]
        predicted_label = class_names[np.argmax(predictions[i])]
        plt.title(f'True: {true_label}\nPred: {predicted_label}', fontsize=10)

    plt.tight_layout()
    plt.show()
```

Appendix VI: Batch test result







Appendix VII: Agreement with MoA

ግብርና ሚኒስቴር
MINISTRY OF AGRICULTURE

AHVPH Executive
Epidemiology Desk

DATA SHARING AGREEMENT

Purpose of Agreement

The purpose of this agreement is to ensure proper use of surveillance data and dissemination of the outputs.

Description of Data

The data generated from surveillance system: disease outbreak and vaccination is shared with researchers whose investigation could help improve the livestock production and health. Data transfer could be through electronic means preferably using email.

Data Security and sharing

The recipient of data should be responsible for proper management of the data including storing in secured storage by restricting access to other peoples. The data should be used for the intended purpose and should not be shared for third party. Consent of the ministry should be obtained for publication. Epidemiology Dept should be involved to review the document.

The ministry should be communicated for publication of the outputs from the data. The final report or manuscript should be sent to the ministry.

Payment

Describe if and how payment for data will be provided between the Providing Organization and Requesting Organization.

Data recipient/researcher

I Madan Beshir confirm that I have received data on FMD Disease outbreak and vaccination from 2022 to 2023 year for the purpose of my research entitled "CATTLE FARMS FMD Detection using machine Learning" and I hereby signed to use the data for the intended purpose and to share the report after completing the research.

Signature: ____________ Date: Mar, 5 2024

Email: madanbeshir496@gmail.com Phone No. 0968073824