CATTLE SKIN DISEASES IDENTIFICATION MODEL USING MACHINE LEARNING APPROACH

WORKEE, GETACHEW

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

FACULTY OF COMPUTING

CATTLE SKIN DISEASES IDENTIFICATION MODEL USING MACHINE LEARNING APPROACH

MSC. THESIS

BY

WORKEE GETACHEW MINTESNOT

PROGRAM: COMPUTER SCIENCE

BAHIR DAR, ETHIOPIA

MARCH, 2021

CATTLE SKIN DISEASES IDENTIFICATION MODEL USING MACHINE LEARNING APPROACH

By

Workee Getachew Mentesnot

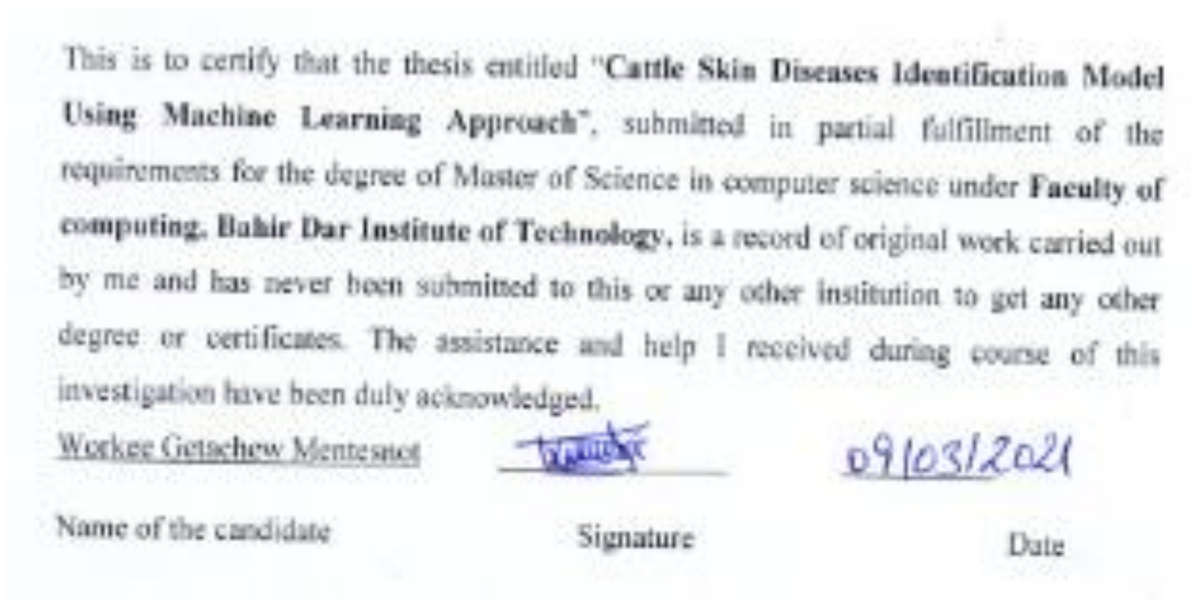
Submitted to Bahir Dar institute of technology in partial fulfillment of the requirements for the degree of masters of Science in the computer science in the computing faculty.

Advisor name: Abraham Debasu (assis prof.)

Bahir Dar, Ethiopia

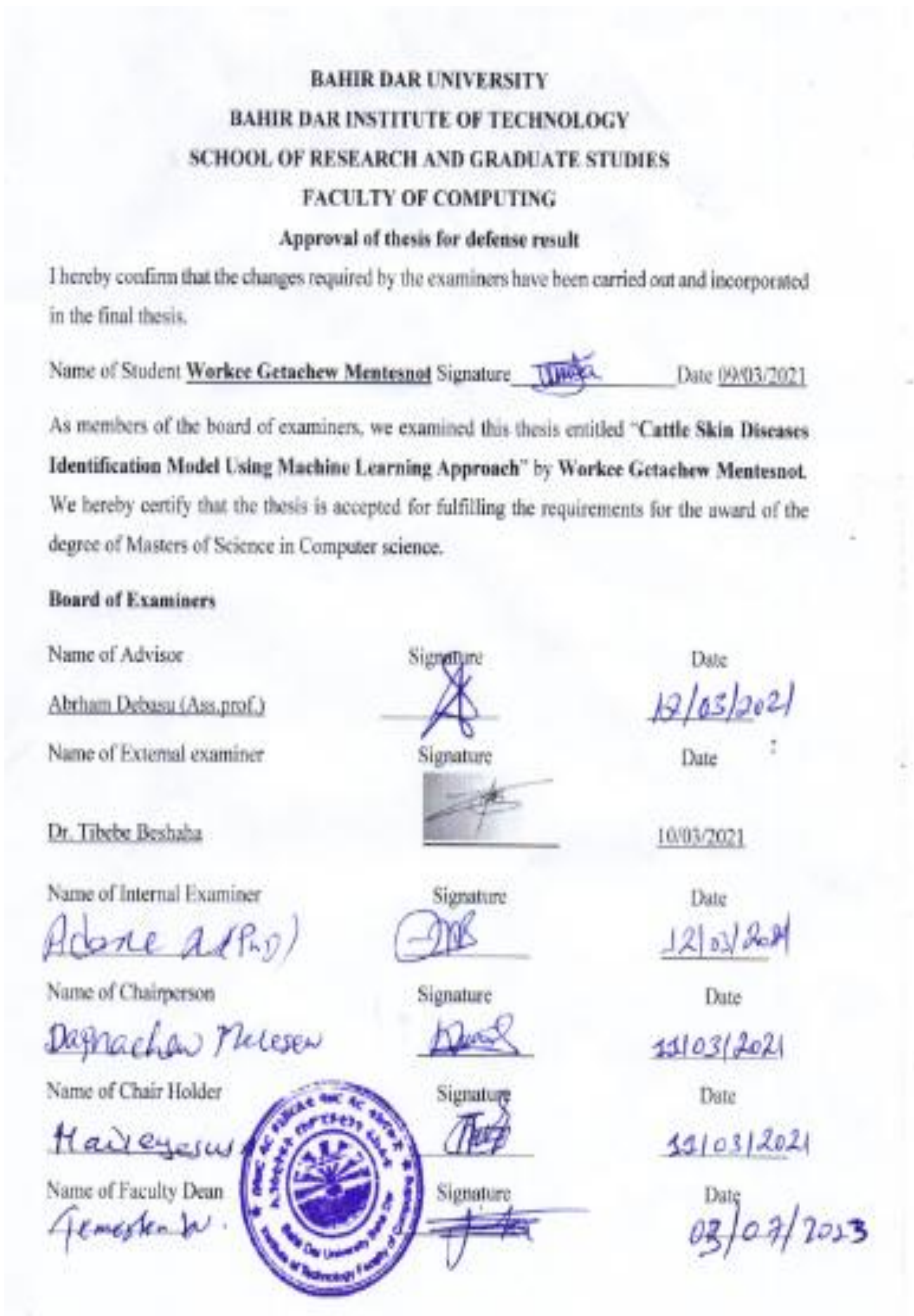
March, 2021

DECLARATION



This is to certify that the thesis entitled "**Cattle Skin Diseases Identification Model Using Machine Learning Approach**", submitted in partial fulfillment of the requirements for the degree of Master of Science in computer science under **Faculty of computing, Bahir Dar Institute of Technology**, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during course of this investigation have been duly acknowledged.

Workee Getachew Mentesnot	[illegible]	09/03/2021
Name of the candidate	Signature	Date



BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND GRADUATE STUDIES

FACULTY OF COMPUTING

Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student **Workee Getachew Mentesnot** Signature [illegible] Date 09/03/2021

As members of the board of examiners, we examined this thesis entitled "**Cattle Skin Diseases Identification Model Using Machine Learning Approach**" by **Workee Getachew Mentesnot**. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science in Computer science.

Board of Examiners

Name of Advisor	Signature	Date
Abrham Debasu (Ass.prof.)	[illegible]	12/03/2021
Name of External examiner	Signature	Date
Dr. Tibebe Beshaha	[illegible]	10/03/2021
Name of Internal Examiner	Signature	Date
Adane a(PhD)	[illegible]	12/03/2021
Name of Chairperson	Signature	Date
Dagnachew Melesew	[illegible]	11/03/2021
Name of Chair Holder	Signature	Date
Haileyesus	[illegible]	11/03/2021
Name of Faculty Dean	Signature	Date
Gemechu W.	[illegible]	03/07/2023

To my lovely family

ACKNOWLEDGEMENT

First, I would thank the Almighty God, for He is the Source of Wisdom and the Beginning of Knowledge. He helps greatly throughout this work.

I would also thank the Ever-Virgin, St. Marry, Mother of our Lord. My deepest gratitude is to my advisor, Abraham Debasu (assistant professor). I am deeply thankful to him for giving his insights and constructive comments. I expect one day I would become as good an advisor as he becomes and guide my students as he has been to me.

I would thank Dr. Mequanint, for his teaching during Research Design and Methodology course. He helped a lot in motivating us to do our work within the schedule.
My sincere thanks also tothe staffs of BahirDar zuria woreda veterinary clinic, who have provided images and identified the diseases properly.

I have also thanked my mother, Mrs. Achash, and my husband, Mr. Belete Baye, for his uninterrupted help infinding the topic for this term paper as well as spelling and logical errors.

ABSTRACT

Nowadays, Skin diseases are become the most infectious diseases occurring in cattle of all ages and often found in humans and animals. Cattle skin diseases are particular kinds of diseases caused by tumors, allergies, parasites, autoimmune, bacteria infectious, and viral diseases. Many cattle skin diseases are very dangerous, mostly if not treated at their first stage. These diseases have imposed economic losses in different aspects especially in day-to-day usage they reduce milk yield, leather quality, and performance in draft cattle. In this regard, the study and investigation of these diseases have been an interesting sector for computing experts. But there is only one paper that has been done using KNN. However, the methods they have used are inefficient. I.e. KNN is a lazy learner Algorithm and always computes the distance metrics due to this it is not applicable for diseases with high correlation. This paper was the second study on four common cattle skin diseases including lumpy, dermatophytosis, dermatophylosis, and wart diseases. In this work, we have developeda cattle skin disease identification model usingthe comparison of three filtering techniques (median, Gaussian, and Gabor filter), threshold segmentation, hybrid feature extraction (CNN, and HOG), and SVM for classifications. We Have to use the HOG technique for the fact that CNN was not invariant for image transformation (rotation and illumination change) while HOG was invariant in image transformation. Anaconda python programming tools have been utilized to complete the overall coding mechanisms. The images were sourced from Bahir Dar zuria woreda veterinary clinic and Bahir Dar university agriculture and environmen tal science campus using a Huawei Y635 model smartphone memory size 8GB and resolution of 840x854. We have collected a total of 765 original images for four classes and CNN needs a large dataset due to this we used augmentation techniques. The total number of augmented images is 2000 images. The images are in a jpg format. We have achieved 96.5% accuracy in CNN, 93% with HOG, and 98.75% using a hybrid feature.

Keywords: *CNN, local feature descriptor, feature extraction, SVM, thresholding.*

Table of Content

LIST OF FIGURE

LIST OF TABLE

ALGORITHM

ABBREVIATIONS

2D-DWT	Two Dimensions Discrete Wavelet Transform
ANN	Artificial Neural Network
BPNN	Back Propagation Neural Network
CBIR	Color Based Image Retrieval
CBR	Case-Based Reasoning
CFS	Correlation Feature Selection
CNN	Conventional Neural Network
FCBF	Fast Correlation-Based Filter
GLCM	Gray Level Co-occurrence Matrix
HOG	Histogram of Oriented Gradient
ICA	Independent Component Analysis
KNN	K-Nearest Neighbors
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
ROI	Region of Interest
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

CHAPTER ONE

1. INTRODUCTION

1.1.Background

In an animal’s body,the skin is the largest and major organthat separates the inner part and protects from the outer environment like allergy, viruses, infection, parasites, bacteria, and controls body temperature. Animals give different purposes including cattle like milk, meat, fiber, fuel, fertilizer, shoemaking, and leather goods. Most of the time, animal skin contributes a lot to the world's economy. The development of the leather industry requires a great measure of materials from various backgrounds, the principal source of which is animal skin. However, in today’s world,Skin diseasesare becomingthe most popular diseases from other diseases and common to all animals with different types of symptoms for the reason that those diseases cantransferfrom one animal to another animal with direct contact and fly bites. They are very dangerous, primarily if not treated at theirfirst stage. Skin diseases not only damage the skin but also have a large effect on the inner part of the body.

Also,these diseasesimpose economic losses, by reducing leather quality, milk production, and performance of draft cattle and causing frequent mortality of cattle. Many people have tried traditional methodsto treatinfections of cattle skin diseases. However, these treatments have notbeen suitable for the problem that causes death. Furthermore, there are difficulties to find the infected part to analyze the skin sickness for dermatologists.Thisthesis work has dealt with methodological and scientific contributions to those problems.Methodologically, there is no existing work available on enabling hybrid features and SVM to find cattle skin diseases to the best of our knowledge.Scientifically, there was only one study that did not consider diseases seem only animals. However, in this work, we have considered those skin diseases only appear in cattle skins considering the target skin diseases of Lumpy, dermatophilosis, dermatophytosis, and wart.

There have been some research works regarding skin disease identification algorithms and techniques. In 2012, (Abdul-rahman & Norhan, 2012)have developed Dermatology Diagnosis with Artificial Neural Network using BPNN through feature selection methods (i.e. CFS and FCBF). CBIR techniques have been utilized as another approach of skin diseases detection method in(Fekri-ershad et al., 2012), andimage processing techniques have been used for human skin disease identification(Amarathunga et al., 2015) focused on three skin diseases. As referenced in (Aruta et al., 2015) the authors proposed mobile-based medical assistance using CBR, image processing and achieved an accuracy of 90%. The authors(Ambad & Shirsat, 2016)presented an image analysis technique with statistical analysis and correlation algorithmsfor automatic skin disease prevention and detection. According to (Yadav, 2016) proposed skin disease diagnosis using various image processing techniques (i.e. filtering, segmentation, feature extraction, image pre-processing, and edge detection).

As presented by (V. B. Kumar, 2016)a dual-stage approach that combines Computer Vision and Machine Learning was utilized for human skin diseases detection. On other hands(M. Kumar & Kumar, 2016) proposed human skin diseases detection mechanisms by using KNN as feature extraction and SVM as a classifier. Moreover, (M. Kumar & Kumar, 2016)suggested the GLCM and SVM approach for the same application, and(Ansari & Sarode, 2017) anticipatedimage processing for diagnosis of skin diseases.

In this work, we have considered four cattle skin diseases including lumpy, dermatophyto sis, dermatophylosis, and warts. The imageswerecaptured by using smartphones from Bahir dar Zuria woreda veterinary clinic, Bahir Dar University agricultural and environmental science campus. The modelwas performed by taking the input image followed by preprocessingtechniques and segmentation of the skin area. The next step was the extraction of the affected area classification with the help of SVM. Furthermore, we have developed identification models that enable users to find cattle skin diseases using hybrid features to carry out the limitations of the techniques andanaconda python software to develop the overall processing of the skin images of the cattle.

1.2.Motivation

Nowadays, Skin diseases are common to all animals with different types of symptoms. It causes different economic problems that decrease production, reproduction, skin, and hide quality (Ambilo & Kebede, 2019). Many animal skin diseases are very dangerous, particularly if they are not treated at their first stage. Those diseases like lumpy are economically discouragingthe growth of animals and results in the reduction of milk products, infertility, abortion, trade limitation, and occasionally death in most African countries including Ethiopia (Feyisa, 2018).For those problems, many researchers tried to overcome those problems by providing vaccination treatments that require continuous monitoring of the diseases, prohibitively expensive and time-consuming in a large livestock industry for short time transmitted diseases.

On the other hand, most people treat infections of cattle skin diseases using traditional methods. However, these treatments are not suitable for skin problemsand it was bad and resulted in death. Those problems cause not only damage to the skin but also have a large effect on the inner part of the body. For dermatologists, many difficulties weredistinguishing the infected area to analyze the skin sickness.

Furthermore, as we know, Ethiopia is the leading country in Africa with a high number of cattle and exporting leather products for foreign countries. For this sector, automatic identification of skin diseases has been a primary focus for researchers.On the other hand,cattlehave given less attention to protection and health concerns onthe contrary.We have to give priority to the highlydemanded leather products and the sector in local as well as international markets due to its application. Even though the leather sector has a great economic role over the country, there was no research work to develop good protection and sanitation habits for those cattlethat are the source of the sector. So thisresearch work was aimed to solve this problem by identifyingcattle skin diseases for the sake of the users, researchers, marketers and the country to be benefited for the fact that cattle skin is the main source of leather industries.Such facts provoked us to work ondeveloping an effective cattle skin diseases identification model.

1.3.Statement of the Problem

In the livestock industry, several animals have been infected by different diseases. Likewise, the cattle have been affected by several diseases (like lumpy, dermatophytosis, dermatophylosis, and wart) which highlyreduce the quality and quantity of the livestock product. Early and accurate identification ofsuch diseases is required for preventing the diseases accordingly before the loss. The diseaserecognition process based on the knowledge of experts is quitetime-consuming, laborious,error-prone, and subjective. The advanced solution is automating the disease identificationand classification process using different algorithms or approaches.There are a lot of researches working on human disease recognition and diagnosis including animals.

The imaging and machine learning approach studied withthe authors (Shah et al., 2019) proposed animal skin disease detection using KNN as feature extraction and SVM for classification. However, the methods they used are inefficient. I.e. KNN is a lazy learner algorithm and always computes the distance metrics because of this it is not applicable for diseases with high correlation(Deng et al., 2016)(M. Panwar et al., 2017).The use of hybrid features, local features like LBP, HOG,SURF,and high-level CNN features perform better than the three individuals in acomputer vision system (Kabbai et al., 2019).Using deep CNN with thousands of datasets takes high computational memory, andhigh processing units like GPU are more powerful. However, this was not the case in most imageprocessing problems. CNN did not extract enough features with a smaller dataset. Whereas,local feature descriptors, like LBP, HOG, SURF, etc., do not require large datasets to learnand are computationally feasible. These feature descriptors have better performanceto analyze the image local patterns but recognition using local patterns is not enough again(Kabbai et al., 2019).

To fill this gap, using the better ofthetwo creations, we have proposed a hybrid of HOG localfeature descriptor and CNN.To the best of our knowledge, the previous animal disease recognition research works emphasizethe uses of lazy learner feature extraction methods to extract image features.Using a hybrid of high-level CNN features and local image features for using the better of the two worlds is not common. Moreover, when researchers used CNN as a featureextractor, they have not applied preprocessing techniques due to that they reason out asCNN requires too little or no preprocessing. In such a model, the CNN identifies and extractsfeatures directly from the original noise image. But applying a preprocessing techniquecan significantly enhance the performance of the CNN (Pitaloka et al., 2017).

To this end, this study was aimed to answer the following research questions:

- ♣ Which preprocessing techniques were suitable for cattle skin disease identification?
- ♣ How to enhance CNN to extract features towardscattle skin disease identification?
- ♣ To what extent local features of an image can improve the classification perfor mance?
- ♣ To what extent the classification accuracy has been registered?

1.4.Objectives of the Study

1.4.1. General Objective

The general objective of this study isto design and developcattle skin disease identification models using a machine learning approach.

1.4.2. Specific Objectives

To meet the general objective, the following specific objectives were identified.

- ♣ To identify preprocessing techniques that are suitable for cattle skin diseases identification.
- ♣ To identify better filtering techniques to enhance CNN feature extraction techniques towards cattle skin disease identification.
- ♣ To checkthe performance of local features descriptors.
- ♣ To analyze the performance of the proposed model.

1.5.Scopeand Delimitation

This study wasfocused on the identification of cattle animal skin diseases. Those skin diseases arelumpy, dermatophilosis, dermatophytosis, and warts) not any other animal skin diseases.

1.6.Methods

To achieve the specified general objective and specific objectives we follow the following
methods, materials, and procedures.

1.6.1. Literature Review

Gather and find relevant documents to indicate the solution of the problems based on the context of the previous literature. Therefore documents related to image processing and computer vision as well as related to skin diseases image classification and recognition are exploredto organize relevant information.

1.6.2. Data Collection

In this context,it wasdefined as the act of retrieving an image from a different source. In this work,an image was captured by Huawei Y635 model smartphone memory size 8GB and resolution of 480x854. The images collected from BahirDar zuria woreda veterinary clinic and BahirDar University agricultural and environmental campus image is in jpg format. During collecting an image the professionals of the organization are helped very well in the occasion of identifying and grouping the cattle skin diseases. We have collected a total of 765 original images and we used augmentation techniques and the augmented images are 2000 for four classes. For each class 500 input images.

1.6.3. Model Design and Experimentation

To design the cattle skin diseases classification model we used preprocessing, segmentation, feature extraction, and classification techniques.In the preprocessing techniques like image resizing, filtering, enhancement, and augmentationalso a thresholdfor segmentation, and CNN fordeep feature extraction. Furthermore,we have used the comparison of (SURF, HOG, and LBP) for local feature descriptors and SVM for classification. Finally, afterdesigning the model we used anaconda python programming language for experimentation ofcattle skin diseases classification model. Python was used for writing the required source codes and Keras (TensorFlow as backend) was used for designing theCNN model because it is powerful(Cholet & Francois, 2018).

1.7.Significance of the Study

This study has the following importance for different areas.

1. For dermatologists

- ♣ Enabling dermatologists to easily findcattle skin diseases that have related but different features.
- ♣ Enabling dermatologists to apply treatments according to the type of cattle skin diseases.
- ♣ To reduce heavy dependencies on dermatologists for the identification of cattle skin diseases.
- ♣ To know different types of cattle skin diseases automatically.
- ♣ Improving the effectiveness of cattle skin disease control mechanisms.

2. Technically for researchers

- ♣ To know the best preprocessing technique.
- ♣ To know the best activation function towards cattle skin disease identification.
- ♣ To know the best local feature descriptor towards cattle skin diseases feature extraction.
- ♣ To recognize the best kernel function towards cattle skin diseases identification.
- ♣ To distinguish the greatest image resizing techniques towards cattle skin diseases identification.
- ♣ To know based on how the model performs towards cattle skin diseases identification.

1.8.Organization of the Study

In this chapter, the general, as well as the specific objectives of the study, are presented. To meet the objective some methods and procedures are discussed in this chapter. The model evaluation techniques, tools that are used based on how this work is implemented arealsopresented. Finally, this chapter showed the scope of the study and the main significance of the study. On the other hand in chapter two literature reviews on general image processing techniques, mechanisms of skin detection and classification such as skin pattern, and color as well as shape recognition and classification, vision systems (human vision and computer vision),and a brief description of skin diseases detection, and classification are also presented.

Generally in this chapter, the overall activities of image processing, segmentation, and fe ature extraction, and classification literature are discussed in detail. Also, the common ga ps of the reviewed works and the way how we fill in the gaps are described. Furthermore, in chapter three the proposed model architecture of the cattle skin diseases identification model, image preprocessing techniques, segmentation, feature extraction, and classificati on algorithms are discussed.

In chapter four,the dataset used, experimental evaluation are described in detail. Finally, the test results are compared with the state-of-the-art models.The evaluated performance of the study is also presented in this chapter based on experiment results.

In chapter five, conclusions and recommendations are discussed. This chapter showed general comments and suggestions for the study.

CHAPTER TWO

2. LITERATURE REVIEW

2.1.Introduction

In this chapter description of cattle skin disease, digital image processing, different prepr ocessing techniques, feature extraction algorithms, and applying different detection and cl assification algorithms are discussed by exploring extensive literature reviews. The previous research on skin recognition, detection, and classifications has also been discussed in the related work section of this chapter.

2.2.The Cattle Skin DiseasesDiagnosis

Skin is the largest organ and the wall between internal organs and the external environment. It isan important check for wounds, swellings, pests, and alopecia. The causes of cattleskin disease are infections, tumors, allergies, parasites, and autoimmune diseases. There are many ways that dermatologists have use for cattle health diagnosis and the different approaches for examination sick cattle.

Many dermatologists used manual examination techniques to identify cattle skin diseases. Skin disease diagnosis is the process of identifying the unique symptoms and signs of thespecific diseases.The cattle have been affected by several diseases (like lumpy, dermatophytosis, dermatophylosis, and wart) which highly reduce the quality and quantity of the livestock product. In the early days, the controlling and analysis of skin diseases wereperformed manually through visualization. This requires experienced individuals andwhich iserror-prone, laborious, time-consuming, and subjective.

Lumpy skin diseases (LSD):characterized by higher fever, enlarged superficial lymph nodes, and multiple nodules on the skin. It is transmitted by fly bites and direct contact and controlled by expert vaccination.
Dermatophylosis: characterized by a moist circular patch
Dermatophytosis: characterized by patches of hair loss on the skin.
Wart: characterized by the presence of pain and deformity

Cattleskin diseases included in this study are lumpy (a), dermatophylosis (b), dermatophytosis (c), and wart (d) are shown in fig 2.1 respectively.

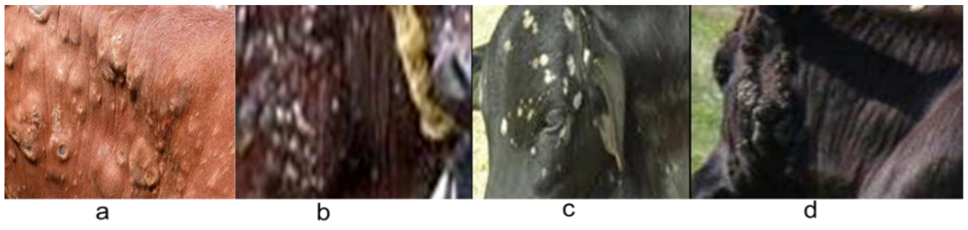


Figure 2.1. The common cattleskin diseases included in this research (Phil Scott, 2017)

2.3.The Digital Image Processing

This refers to the processingof digital imagesusingdigital computers and computer algorithms to perform specific operations on an image to get an enhanced image, by extracting some useful information from the image (Broti et al., 2020). Image processing is composed ofthree stages (Lyubchenko et al., 2016). These stages are low, middle, and high-level process stages. Low-level processes include image preprocessing for the reduction of noise, contrast enhancement, and image sharpening applications. In this process,inputs and outputs are images. On the other hand, mid-level processing focused on the image segmentation process, and higher-level processing emphasis "making sense" for the collective of recognized objects(Broti et al., 2020).

2.4.The image processing

There are common steps in image processing like image acquisition,image preprocessing, segmentation, and feature extraction.

2.4.1. Image Acquisition

The image acquisition mainly ends up with noisy images(Ajay Kumar Boyat, 2019). The sources of noise in image capturing are the proper focus of the camera. Thus, if the camera is not properly focused then we get blurred images. There are also other causes for the occurrence of noise in images. Conditions like foggy environments are causes that can introduce noise into images. Noise hurts digital image processing(Broti et al., 2020).It is the first activityin image processing involved for taking an image using hardware devices appropriate for taking an image unless no processing is possible. Imagesare alsoacquired from a database or another source tailored for research purposes. Most of the time, the unprocessed image has been taken that requires further processing and analysis to be used for specific purposes.

2.4.2. Image Preprocessing

Image preprocessing is the low-level process utilized for removing the noise from a signal of the image (P. Panwar et al., 2016). It isused for detecting an infected part of the imageusinga collection of scanning images (P. Panwar et al., 2016).

2.4.2.1.Image Resizing

It is one part ofpreprocessing performed with the help of different interpolation techniques includinglinear, nearest neighbor, bilinear, and bicubic techniques. Linear interpolationmethods are the simplest image resizing methods with an acceptablequality of the resized image. However, in this technique, the fine details of the images arenot their crucial requirement.Bilinear interpolation mechanism is theextension of linear interpolationin which the value of four nearest pixels has been used to classify unknown pixel values.But it requires more processing time,more complex, and has a longer calculation time as compared to the nearest neighbor interpolation technique(Fadnavis, 2014).

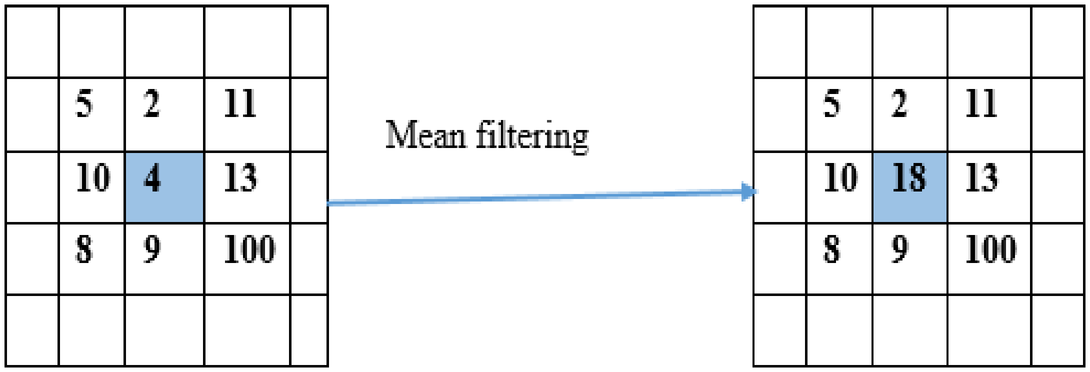
The Nearest Neighbor interpolation: was a very easy resizing mechanism and has requi red less computational time. It has been utilized usingthe pixel value that was very similar to the surrounding coordinates of the intended interpolation point and found the closest comparable pixel in the source imagefor the pixel in the final image(Fadnavis, 2014).

The bicubic interpolationrefers to an extension of the cubic interpolation technique. It works by calculating the estimated or interpolated points of the weighted average of a 4x4 closest neighborhood points, a total of 16 pixels. These pixels are in various distances from the pixel insight and the closer pixels are given higher weight in thecalculation. It was obvious that the bicubic interpolation technique producedsharper images than other interpolation techniques and it provided an ideal combination ofcomputation time and image quality. Due to that, bicubic interpolation was the popularrescaling mechanism in various image editing software like Adobe Photoshop, and printerdrivers.

2.4.2.2.Noise Reduction

In noise removal, there were several noise reduction techniques including mean filter, wiener filter, median filter, Gaussian filter, and Gabor filter(Hambal et al., 2015). The Mean filter technique was a linear filtering processutilized for removing noises from animage by reducing the intensity variation between pixels(G. Kaur & Kaur, 2012). It was
convolution-based filter that replaced the focal incentive pixel in the window with mean or
the average value of all pixels in the window. As shown in table 2.1. The mean value of thepixels in the window of the first table is 18. Therefore, mean filtering replaces the center
value 4 by the mean value 18. The main drawback of the mean filtering algorithm was that asingle pixel unrepresentative of the neighboring pixels can affect the mean value of all those neighborhood pixels.

Table 2.1 A 3x3 window size means filtering example


Mean filtering

	5	2	11	
	10	4	13	
	8	9	100	

	5	2	11	
	10	18	13	
	8	9	100	

On the other hand, the median filter was a nonlinear statistical filtering approach used for removing noises from images better than the mean filter(Tania & Rowaida, 2016). The median filtering method has been worked by moving in the image pixel by pixel and replacing each pixel value with the median value of neighborhood pixels as shown in Figure 2.3. The neighboring pixels together were called windows and moved pixel by pixel all over the image (Rani et al., 2016). Typically, median filtering has been used for removing the salt and pepper noise. Median filtering has kept the edges of the image when it has removed the noise. As a result, the image has not been blurred as other filtering methods.

Table 2.2 A 3x3 window size median filtering example

	9	6	20	
	14	60	13	
	8	3	23	
3, 6, 8, 9 , 13 , 14 , 20 , 23 , 60 Median (the centered value 60 is replaced by 13)				

Median filtering was good for denoising the salt and pepper noises while it was poor for removing Gaussian noises. The main drawback of the median filter was it has removed bothnoise and fine detail of an image.Gaussian filtering has been found as a linear filtering technique that was based on two-dimensions (Reshma & Shan, 2018). It was utilized for removing Gaussian noises though poor for eliminating saltand pepper noises. Wiener Filter was the Mean Square Error (MSE)optimal linear filter for images distortedby the additive noises and blurring (Tania & Rowaida, 2016). It was a frequency domain
filtering techniquecomputed using the Discrete Fourier Transform (DFT).Similarly, the Gabor filter was a linear filter that analyses whether there was any specific frequency content in the image in specific directions in a localized region around the point or region of analysis. Gabor filtering is commonly used for texture analysis

2.4.2.3.Image Enhancement

Image enhancement is the process required for bringing out the details that are concealed and simple to highlight useful features of an image including sharpening an out-of-focus image, improving the image brightness, highlighting edges, etc. Image enhancement has been done after image restoration has been completed(Rishi & Bajpai, 2017). Unlike imagerestoration, image enhancement has beena subjective operation; meaning that image enhancement wasapplied based on the human subjective preferences of what a "good" enhancement resultcomprises(Gonzalez et al., 2018).

There have been various techniques forimage enhancement includingthe most popularly used image enhancement techniques of contrast enhancement, edge enhancement, Intensi ty, Hue, and Saturation Transformations (Reshma et al., 2018).Contrast enhancement has been used to adjust the contrast ratio of an image for resolving the powerand brightness of the entire pixels of images(Rishi & Bajpai, 2017). Thesetechniques are the most widely used image enhancement techniques to produce an imagethat has an optimum contrast ratio by utilizing the entire display medium brightness range. As the state of the art, the most common contrast enhancement method washistogram equalization.Edge enhancement has been utilized as the technique of enhancing or sharpening the edges (discontinuitiesof the image intensity) in the image to improve the visual perception of edges(Zubair et al., 2014). Contrast enhancement adjusts the brightness or contrast of all pixels inthe image whereas edge enhancement works to discriminate the pixels in the edge from theothers pixels. Image edge enhancement has been required when every discontinuity ofimage intensity or edges was not visible. An image edge viewed at a good visualizationor with excellent eyesight typically requires lesser or no edge enhancement.

The Intensity, Hue, and Saturation (IHS) Transformations:

Rishi & Bajpai have stated that image color conversion is one of the preprocessing tasks and involvesthe processes ofconverting images from one color space to another for ease of computation. They have also described RGB (red,green, and blue) coloring system as an additive primary color system that is used in digitalcameras and scanners. Furthermore, they stated the other important coloring approach described as the intensity, hue, andsaturation system (IHS) thathas been useful for the fact that it presented colors almost as the humanobservations perceived them(Rishi & Bajpai, 2017).

The intensity (I) represented the brightnessvariations and ranging from 0 (black) to 255 (white). Hue (H) represented the dominantcolor wavelength. Hue values ranged from 0 to 255. Saturation (S) represented the colorpurity and ranging from 0 (completely impure colors) to 255 (pure and more intensecolors). RGB images are transformed into IHS using equation (2.1)(Rishi & Bajpai, 2017).

$$I = R + G + B;\ H = \frac{G - B}{I - 3B}\ ;\ S = \frac{I - 3B}{I} - - - - - - - - - - - - - - - - 2.1$$

2.4.3. Image Segmentation

P. Panwar et al. havestated Segmentation asthe processof identifying and isolating the surface and regions of the digital imageresembling the structural units depending on various features likecolor or texture contained in the image (P. Panwar et al., 2016). Even if there have been various generic methods for image segmentation some of them are discussed here.

2.4.3.1.Segmentation using Edge Detection

As D. Kaur & Kaurdeclared that Edge detection has been the basic step in image segmentation utilized for dividing imagesusing pixels of an image(D. Kaur & Kaur, 2014). They have divided Edge detection operators into two categories, asfirst-order derivative and second-order derivative operators that second-order derivative operators gave reliable results(D. Kaur & Kaur, 2014).

2.4.3.2.Segmentation using Thresholding

P.Panwar et al. have described that Thresholding segmentation as the simplest approach t
o segment images based on the intensity levels. Threshold segmentation has been implem
ented in two ways (globallyand/ or locally)(P. Panwar et al., 2016). Global thresholding has been utilized for separating objects and background pixels using threshold value chosen, and binary partition. In Local or adaptive thresholding the threshold value

differed in the image depending on the local representatives of the subdivided parts in the image(P. Panwar et al., 2016).

2.4.3.3.Segmentation using Region-based approach

D. Kaur & Kaur, have also described another method of segmentation that focused onsimilarity Based Segmentation approach implemented using pixels related to the same object. They have formed the thresholding segmentation bound with region-based segmentation by primarily identifyingthe change in color and texture followed byedge flow conversion into a vectorwith edges detected for additional segmentation. Region-basedsegmentation was relatively simple and resistant to noise as compared to the edge detection method(D. Kaur & Kaur, 2014).

2.4.3.4.Segmentation using Feature-based Clustering

Furthermore,D. Kaur & Kaur have developed the Clustering-based segmentation involved in the process of combining the groups based on their attributes and contentscontaining a group of similar pixelfixing to a specific region different from other regions. This model has been developed with clustering methods like Hierarchical algorithms, partitional algorithms, K-means algorithms, and fuzzy C-means algorithms. In content-based clustering, the inherited characteristics of the pixels like shape, and texture have been used for grouping(D. Kaur & Kaur, 2014).

2.4.4. Feature Extraction

Di Ruberto & Putzu stated that feature extraction has been the major activity in digital image processing after segmentation processes havebeen ended. They have designedfeatures of imageswith a setof quantifiable attributes describing theobject by which the extracted image features have been used for providingmeaningful information about the image to get betterclassification performance(Di Ruberto & Putzu, 2016). On the other hand,Kabbani et al. statedthat extraction of effective featureswas the important but challenging issue inimage processing for the specified features used for discriminating objects and directlyaffected the efficacy of the classification(Kabbani et al., 2019). The extracted set of featureswere commonly known as feature vectors.Kabbani et al. have broadly classified the feature extraction approaches into low-level and high-levelfeatures(Kabbani et al., 2019). Accordingly, Low-level features weredescribed as the smaller details and visual attributesof images like points, lines, edges, corners, color, etc that provide alimited set of simple, frequent, well localized, and individually identifiable features todescribe the data. These features were extracted automatically without knowing the entireimage’s property with examples of low-level feature descriptors including SIFT, SURF, LBP, and HOG, etc. While high-level features used the low-level features to detect largershapes and objects in the image. The extraction of a certain feature that didn’t significantly vary because of different factors like rotation, illumination, scalevariations, viewpoint changes, occlusion, and noiseswas avery challenging issue.

2.4.4.1.Conventional Neural Network

Khan et al. stated that CNN is a new system designed for both feature extraction andclassification. The convolution and pooling layers of CNN were used for extractingimportant features of images.In which fully connected layers received the extracted featuresfrom all previous neurons as input to map them into the classification layer (Khan et al., 2020). Deep learning algorithms have been foundin high-level feature extractors that learn high-levelsemantic features using the lower layer low-level features. Features extracted using thedeep CNN have not been invariant to any of the affine transformations.Ren et al. stated that the conventional neural network has been recognized as one of the deep neural networks designed for image recognition, detection, and classification (Ren et al., 2016). Therefore, the neuron in a layer was connected to a small region of the layer in the previous layer. Furthermore, CNN has been utilized with element-wise multiplication of matrices followed by submission known as convolution. As Khan et al. clearly stated that CNN was contained with many components and related elements like convolutional
Layers, pooling layers, activation functions, fully connected layers, Los functions, regularizations, and optimizations (Khan et al., 2019).

2.4.4.2.The Building blocks of CNN

Many scholars agreed that the Convolutional layer has been the basic component of CNN designed to learn the feature representation of the data by taking row inputs or feature maps to compute the different featuresin input with different kernels(filters). The new feature maps produced after convolving with kernel filters and applying element-wise multiplication and addition have been the input inthe next layer. The working principle of that convolutional layer was different from other neural network layers for the fact that contains filters that convert images. The convolutional filterhas been the same as feature maps and it was two-dimensional matrices and the feature maps were extracted depending on the convolution filter like 4x4 pixel input image and two convolution filters as shown in table 2.3 below(Hossain & Alam Sajib, 2019).

Table 2.3 4x4 pixel input image and two differ 2x2 filters

4x4 input

2	2	3	3
4	7	5	8
34	0	2	6
0	2	6	8

2X2 filtera)

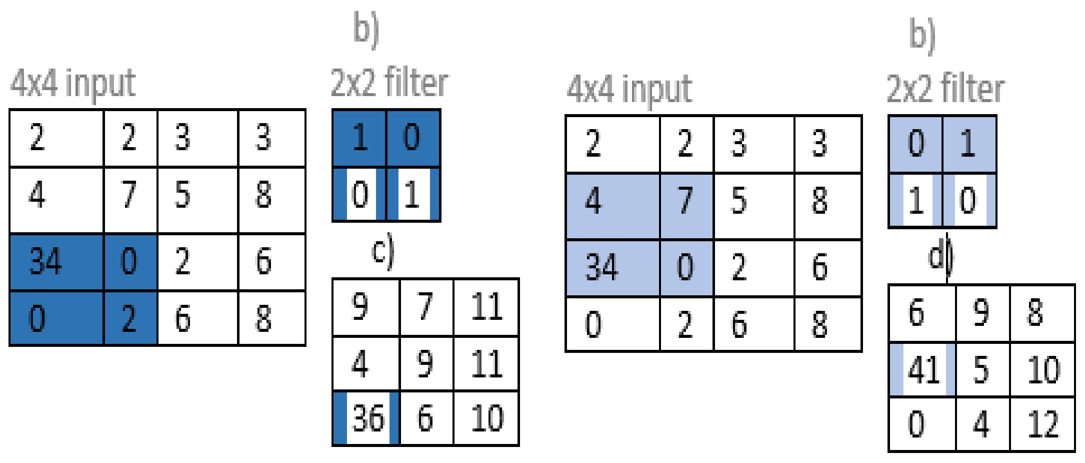
1	0
0	1

b)

1	0
0	1

The result of the 4x4 pixel input image within a 2x2 kernel of two filters a and b were c and d respectively. Both results were 3x3 pixels as shown in table 2.4 below.

Table 2.4. The Result of CNN Convolution Operation on Two Different Filters



2	2	3	3
4	7	5	8
34	0	2	6
0	2	6	8

1	0
0	1

9	7	11
4	9	11
36	6	10

2	2	3	3
4	7	5	8
34	0	2	6
0	2	6	8

0	1
1	0

6	9	8
41	5	10
0	4	12

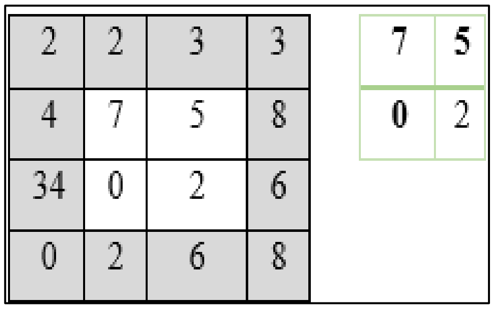
The largest value of result c was 36 and it was generated from the block that has similarities
with the convolutional filter ‘a’ because both had the value at the first and fourth position and zero value at the second and third positions. The convolution operation has a large value when the input block matched the convolutional filter. The largest value of d was 41 using the same 4x4 input, but using the second convolution filter b was also generated from a block that has similarities with the convolutional filter represented in b. so this example hasbeen demonstratedwith the feature map extracted dependent on the convolutional filter. The other parameters controlled the output of the convolutional output like stride number. The stride number indicated that the number of pixels skipped during the convolutional filter or kernels slide over the input vector receptive fields. Receptive fields were some portion of the input vector that has the same dimension as the convolutional kernel. The element-wise dot products of the receptive fields and the kernels produced the feature map or convolved features.

On the other hand, a Padding operation with zero-padding or padding borders of the original image with zero has been applied to a convolution to keep the original image size. Without zero-padding, the spatial dimensions of the input volume have decreased too quickly. The output of applying a 2x2 filter on a zero-padded 4 x 4 image has been utilizedas shown in table 2.3 left while the output of applying a 2x2 filter without zero-padding the original image (right). As described in these examples, zero-padding has been helped to preserve the spatial dimensions of the original 4 x 4 image.

Table 2.6zero-padding (size 1) to the image in table 2.1

0	0	0	0	0	0
0	2	2	3	3	0
0	4	7	5	8	0
0	34	0	2	6	0
0	0	2	6	8	0
0	0	0	0	0	0

Table 2.5 the effect of zero paddings on the output image

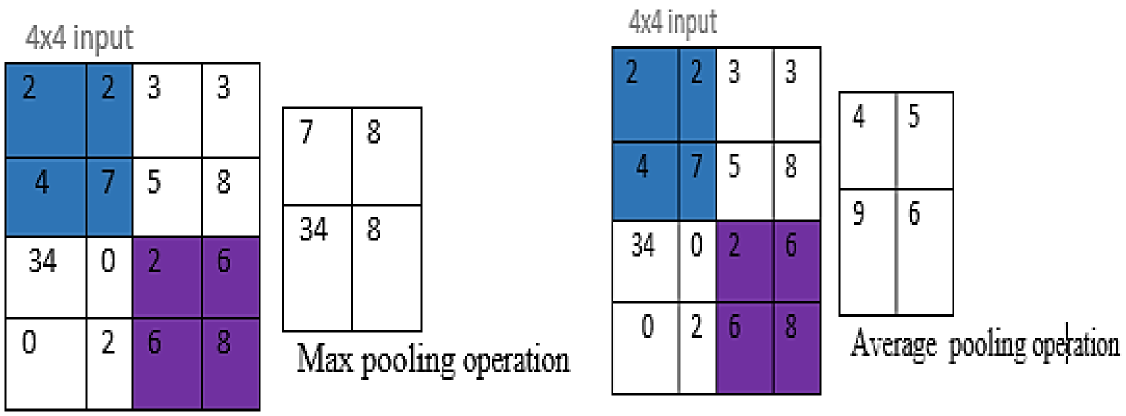


2	2	3	3
4	7	5	8
34	0	2	6
0	2	6	8

7	5
0	2

Besides,Khan et al. have developeda Pooling layerresponsible for reducing the size of the image or feature maps that came from the previous layer. The system hassimple operations to compute with common pooling examples of max pooling and average pooling(Khan et al., 2019) as shown in table 2.7 with 4x4-pixel by applying both max pooling and average pooling operations.

Table 2.7 Max pooling and Average Pooling Operation on 4 x 4 input



2	2	3	3
4	7	5	8
34	0	2	6
0	2	6	8

7	8
34	8

2	2	3	3
4	7	5	8
34	0	2	6
0	2	6	8

4	5
9	6

Max pooling operation was worked selecting the maximum pixel values of the convolution areas and comparing all the pixels of the convolution areas in table2.7was the max value of the first convolution area. On the other hand, average pooling has been worked withadding all the pixel values of the convolution area followed bydivisionwith the number of pixels in the convolution area like$\frac{2+2+4+7}{4} = 3.75 \approx 4$, and all the restwere calculated in the same manner.

Furthermore, Khan et al**.** have developed an activation layer aimed to determine and normalize the output of the network into the same range depending on the activation function(sigmoid, Tanh, and ReLU activation function(Khan et al., 2019)). The sigmoid activation function was utilized to normalize the output of the neuron to the range between one and zero. This type of activation function has a vanishinggradient problem for the fact that the derivative function comes to be close to zero as the input value has been very large or very small. The sigmoid activation function g (z) equation has developedwith the given formula.

$$\text{G (Z)} = \frac{1}{1+e-z} ---------------------------- --(2.2)$$

A similar activation function, the Tanh activation function, has been developed with a range between a positive one and a negative one. Even though the function has a vanishing gradient problem, it was better than the sigmoid activation function for the fact that the latter allowed a negative value. Tanh activation function g (z) was given by the equation:

$$\text{G (Z)} = \frac{2}{1+e-2z} ---------------------------- -\ (2.3)$$

Whilerectified linear unit (ReLU*)* was developedwith a derivative function to overcome the vanishing gradient problem and defined as g (z):

$$\text{G (Z)} = \max\,(0,\text{z}) ----------------------------(2.4)$$

Finally, a fully connected layer (Flatten Layer) used to compute the final output probabilities for each class was developed. It wasapplied at the end of the CNN model before applying the classifier layer (usually Softmax). The Fully-connected Layer was utilized for transforming the feature maps extracted in the final convolution or pooling layer into a one-dimensional (1D) array of numbers. The activation function used in the last fully connected layer for a multiclass classification problem was a soft-max function for normalizing the last fully connected layer output values into target probabilities.

Beyond the activation layers, different Optimization Techniques were developed by different scholars.Those Optimization algorithms wereutilized for decreasing the losses and increasing the accuracy of models. Generally, there have been two optimization algorithms:adaptive optimization algorithms and Non- adaptive optimization algorithms. In the case of non-adaptive optimization algorithms, the learning rate was constant for all features and needed manual selection as indicated in stochastic gradient descent, Momentum, and Nesterov Momentum algorithms. However, in adaptive optimization algorithms, the learning rate was based on the occurrence of features as clearly stated with AdaGrad, Adadelta, and Adam algorithms. Therefore, adaptive optimization algorithms have been selected for solving the problem of manual learning rate schedules in non-adaptive optimization algorithms.

2.4.5. CNN architectures

The architectures of deep convolutional neural networks (CNNs) have evolved for several years and become a crucial issue more on accuracy and speed. For this reason, different scholars have developed different architectural designs at different times. One of the most famously known architectures wasLeNet-5**.** It was introduced by LeCun et al. to classify handwritten digits (Khan et al., 2019). LeNet5 has a total of seven layers, with three convolutional layers, two average pooling layers, onefully connected layer, and one output layer. The sigmoid function was used to incorporatenonlinearity before a pooling operation. The output layer was incorporated withRBF to classify 10 digits. LeNet was the first architecture of CNN with areduced number of parameters that automatically learned features from raw pixels asillustrated in Figure 2.2.

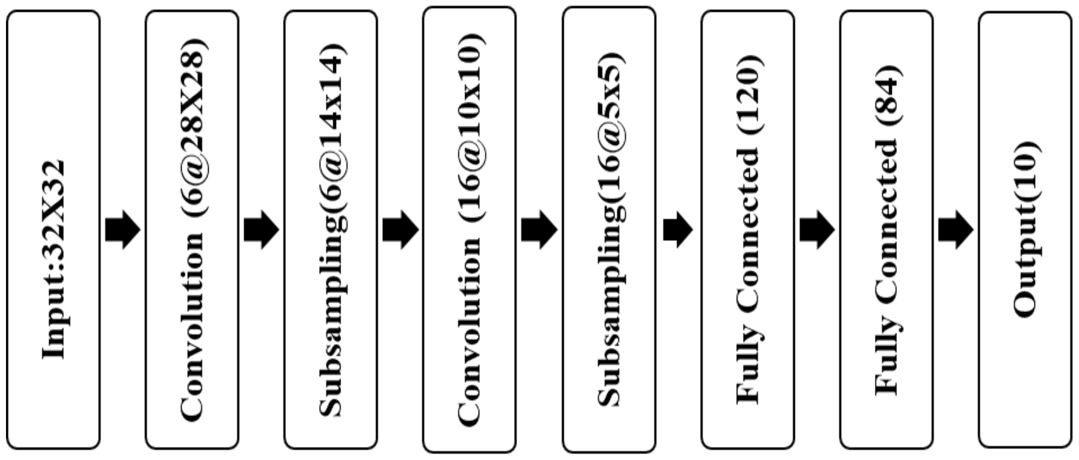
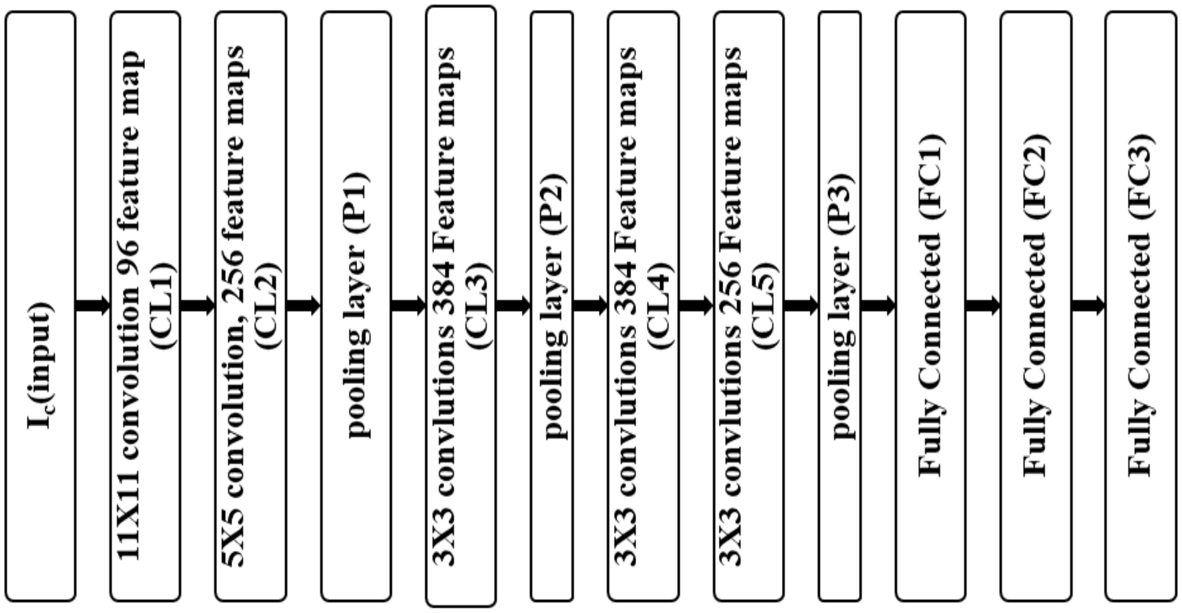


Figure 2.2. The architecture of LeNet5

But,The deep CNN LeNet5 was limited to the handwritten digits classification system and itdidn’t perform well for all image classes (Khan et al., 2019). For this reason, TheAlexNet CNN wasdeveloped and proposed by Krizhevesky et al., to enhancethe learning capacityof the model by creating a deeper layer (with a total of 11 layers) through theapplication of differentoptimization parameters. It consisted of five convolution layers, with three pooling layers and three fully connected layers in between the input layer and the output layer. The input layer was the first layer to define the input dimensionsas shown in Figure 2.12 with 227 x 227 x3 image size and having an output layer as the final layer responsible for classification(Khan et al., 2020)(Khan et al., 2019).



I$_c$(input)
11X11 convolution 96 feature map (CL1)
5X5 convolution, 256 feature maps (CL2)
pooling layer (P1)
3X3 convlutions 384 Feature maps (CL3)
pooling layer (P2)
3X3 convlutions 384 Feature maps (CL4)
3X3 convlutions 256 Feature maps (CL5)
pooling layer (P3)
Fully Connected (FC1)
Fully Connected (FC2)
Fully Connected (FC3)

Figure 2.3The architecture of the AlexNet

In this architecture, each layer was labeled as Cx, Sx, Fx, for x=1, 2, 3…standing for convolutional layer, subsampling/pooling layer, and fully-connected layer sequence number respectively. C1 layer was responsible for performing 11x11 convolutions using stride4 to create 96 feature maps, C2 was responsible for performing 5x5 convolutions using stride 1 to create 256 feature maps and the final three layers(C3, C4, and C5) were responsible for performing 3x3 convolutions using stride 1 to create 384, 384, and 256 feature maps respectively. S1, S2,and S3 layers were utilized with the same 2x2 max pooling without any padding operation. Furthermore,F1 was responsible for performing

256, 13 x 13 featuremaps into a 4096 1D array. AlexNet design was employed with ReLU as a non-linear activation function.

On the other hand, Simonyan et al. proposed an effective CNN architecture design called VGGNet for accelerated architectural design. It was the next best design of the 2014 ILSVRC. The main contribution of the design was showing up the depth of the network as a critical component to achieving better recognition classification accuracy in CNN. The VGGNet architecture has consisted of two convolutional layers; both ofthem used the ReLU activation function. Besides, the single max-pooling layerand the fully connected layers were also integrated using a ReLU activation function. The final layer of the model was theSoftMax layer utilized for classification. The VGGNet architectural design was deeper than the AlexNet design. It has exploited a series of one to more convolutional layers with a similar number of filters having a size of 3×3, astride of 1×1, with the same padding. Thus, the output size was the same as the input size for each filter in each convolutional layer. VGGNet design has used a rectified linear activation function followed by a max-pooling layer with a size of 2×2 and astride of the same dimension. The output feature maps of the final pooling layer were typically flattened; transformed into an array of numbers (or vector), and connected to one or more fully connected layers, also known as dense layers, in that every input was connected to every output by a learnable weight(Khan et al., 2019) as illustrated inFigure 2.4 showing the layers.

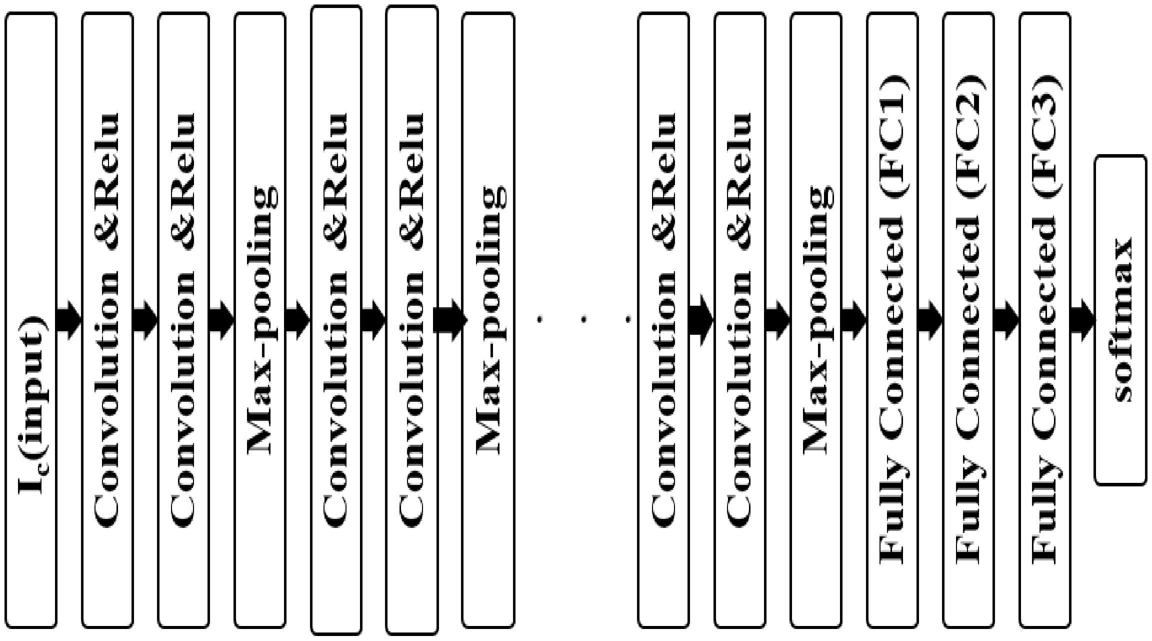


Figure 2.4 VGGNet architecture

But VGGNet was faced withthe main drawback; usage of 138 million trainable parameters, whichcomputationally made it too expensive and difficult to implement on low-resourced systems.

2.4.6. Feature extraction using Local Feature Descriptor

Feature Extraction is the one and the only pillar of machine learning algorithms.There were various feature extraction techniques. Feature extraction using SURF has been built based on the SIFT descriptor. SURF was utilized using an intermediate image representat ion called integral image. It was designed todetect local key points first followed by generating a descriptor for these points. SURF was designed basedon 2D box filters and a fast Hessian blob detector based on the Hessian matrixdeterminant for detecting both scale-spaces and keypoint location. It was designed to approximatethesecond-order Gaussian derivatives of the integral images by applying a set of 2D box filters and scale-spaces to detect blob-like key-point features. The SURF local feature descriptorwas invariant to rotation, scale, illumination, noise, and viewing direction.On the other hand, Scale Invariant Feature Transform **(**SIFT**)** and other local interest point detectors such as SURF and ORB were used for problems to image searching, object matching, with a much lesser degree, image recognition.Another Feature extraction was LBPutilized for binary coding of image pixels using the threshold pixel values. It has been workedthrough a 3x3 kernel considering the central pixel value as a threshold and made a difference between the threshold pixel and the eight surrounding neighborhood pixels. When the neighborhood pixel was greater than the value of the threshold, it was set to '1, unless set to '0'. LBP has been found robust against the change in illumination and it was simple computationally forencoding textures. Since LBP has representedthe discrete patterns, not numerical features, it was hard to combine LBPs with other discriminative features. Furthermore, it has generated higher-dimensional features thatwere sensitive to Gaussian noise in flat regions.

The other local feature descriptorhas been found with HOGthatwas used for extracting local region features using the histogram orientation of edge intensity. The HOG descriptor was designed first to calculate the image gradient using appropriate filter masks like Laplacian and Sobel to extract edge gradients and orientations.

2.4.7. Classification Algorithm

Classification is the process of finding a model that describes and distinguishes data classes with the help of classifiers used to predict categorical class labels. The efficiency and accuracy of classifications have been mainly dependent on the classifier (Uddin et al., 2019). Even though various classification algorithms have been found in machine learning. Only the commonly used classifierswere discussed here below.

2.4.7.1.Naïve Bayes Classifier

Naïve Bayes was a statistical classification algorithm based on Bayes' Theorem with the assumption of independence among predictors(Saiyed et al., 2016). This approach was based on the probability that a given predictor belongs to a class. It has used the Bayes' Theorem (equation 2.5) for calculating the posterior probability p (c|x) of the target/class c particular to the feature/predictor x. The Naïve Bayes has created a directed acyclic graph or tree (Bayesian Network) based on thecomputed conditional probability (L. Dey, 2016). The posterior probability has been given by:

$$= \frac{(x|c)p(c)}{p(x)} ------------------------------- (2.5)$$

Where, the denominator, p(x), was the prior probability of input x, p (x|c), the likelihood probability of the predictor x given in class c, and p(c) was the prior probability of class c. The predictor x was fitted to class c if and only if p (c|x) = max {p (C1|x), p(C2|x) . . . p (Cn|x)}, where c∊C={C1,C2, ..., Cn}.Even thoughthe Naïve Bayes algorithm hasthe main advantage of simplicity, requiringa small dataset,and worked for both linearly separable and inseparable datasets;It was unable to learn relationships among features/predictors because of its feature independence(Ganatra & Patel, 2018).

2.4.7.2.K-nearest neighbor (KNN)

KNN has been worked based on the distance function in case it was an instance-based learning algorithm. The distance function was used to calculate the distance between the new input data and all data in the training data(Hu et al., 2016). Then, k nearest or most correlated data was selectedfrom the training data and the new input data has been assigned the most common class amongstthe K nearest neighbor instances with the help distance functions of The Euclidean distance, Minkowsky distance, Manhattan distance, and Canberra distance. Even though the algorithm was very simple and with norequirements of the datasets to be linearly separable(M. Panwar et al., 2017); it was not robust in the presence of noise and irrelevant parameters within the dataset, more time complex in the case of large training instances for the reason that required calculating the distance between all known instances. The other disadvantage of this algorithm has been found that it was a lazy learner algorithm since it has learned nothing from the training dataset(M. Panwar et al., 2017).

2.4.7.3.Support Vector Machine

SVM has been found as a supervised classification algorithm that has worked based on the norm of margin or decision plane calculation between the classes(A. Dey & Singh, 2016). The margins were calculatedin a way of maximizing the distance between margins and minimizing the classificationerror. The data points that lie on the margin were called support vectors. For the first time, SVM was designed to work on linearly separable datasets usinga hyperplane that divided the two classes. Now, multiclass SVM classifications have been available. These have been worked mapping the input data into a high dimensional (more than two) space H as $R_d \rightarrow H$ and defining a separating hyperplane using a kernel function Φ(x) such that x→ Φ(x). The predefined kernel function Φ(x) has been again used for mapping the new data point into the high dimensional feature space for classification. The kernel function has been the first parameter in multiclass SVM, used for mapping the nonlinear input vector into a set of linearly separable maps. Therefore, the kernel function has changedthe input vector from non-linearly separable data into linearly separable data. Moreover, thekernel function has measured the similarity between two input samples x and y that allowedthe SVM classifiers to make a separation in between with complex boundaries(Amami, 2016). There have been different kernel functions used by SVM like Linear kernel, polynomial kernel, radial basis function (RBF), and sigmoid.The linear kernelwas a kernel function used for a linearly separable dataset. It has workedas of equation (2.6)

$$K(x, y) = x^T y + c ----------------------------(2.6)$$

where c is a constant parameter that needs appropriate adjustment.

The polynomial kernel has been used to transform a given input sample (x and y) called feature vectors in a given feature space into high dimension space with a polynomial of the original variables as shown in equation (2.7).

It was suitable for problems in which all of the training examples were normalized.

$$K(x, y) = (\alpha x^T y + c)d ----------------------------(2.7)$$

where c was a constant, α is a scaling parameter of input samples and d was the degree of the polynomial. On the other hand, the RBF kernelwas found as a commonly used kernel function in various kernellearning algorithms, particularly in SVM. The RBF kernel within two data samples **x** and **y** within some feature space is given by:

$$K(x, y) = \exp\left(-\alpha\left|\left|x - y\right|\right|\alpha\right)\alpha > 0 ---------------------(2.8)$$

Where $\|x-y\|_2$ is the Euclidian distance between x and y, and α is a scaling parameter of this

distance. Also, the sigmoid kernelis the other kernel function that is not positively definite for certain values of its parameters. Therefore, it is mandatory to choose α and c parameters properlyotherwise, the classification becomes drastically wrong. The sigmoid kernel function was poor when compared with RBF and linear kernels. The feature space transformation is done using equation (2.9)

$$K(x, y) = \mathrm{Tanh}(\alpha x^T y + c) ------------------------ (2.9)$$

The other important SVM parameter was the C parameter also called the complexity parameter that was the sum of distances of all points placed on the wrong side of the hyperplane. When the C parameter value was too high then the classification performance was found to become low and vice versa. The third parameter was lambda, the regularization parameter that defines the number of penalties is given for misclassification. It has a controlling role in the trade-off between achieving lowtraining error and low testing error. High values of lambda can minimize the number of misclassified examples. Gamma was the other SVM parameter that defines the influence of a single training example on classification performance. A low gamma value means the training example has low influence and vice versa. SVM is robust enough, even though the training data samples have some noise or distortion. It has good Generalization capability. Although, appropriate kernel function selection is a challenging task in using SVM.

2.4.7.4.Artificial Neural Network (ANN)

Artificial Neural network (ANN) is an artificial intelligence approachthat was derivedfrom the natural or biological concept of neurons(O’Shea & Nash, 2015). ANNhas threeinterconnected layers; input layer, hidden layer, and output layer, as shown in Figure 2.5. The input layer consisted of the input neurons that accept input features, X1, X2… Xiand send them into the hidden layer neurons which again forwards their output into the output layer neurons (figure 2.5). The interconnection between neurons has some weight value W which has beeninitially set with random values. The outputs of each neuron were the function of the sum of weighted inputs. The ANN can be trained with different learning algorithms for instancethe delta learning rule, backpropagationlearning algorithm, etc. In the backpropagation algorithm, the network's predicted output has been compared with the actual output and the output error was calculated. Then, the error fed back into the network to adjust all weights in the net so that the network's output values have been closer to the actual value. After the network converges, the weights of all connections have been fixed and the network has determined the classes of a set of new data.

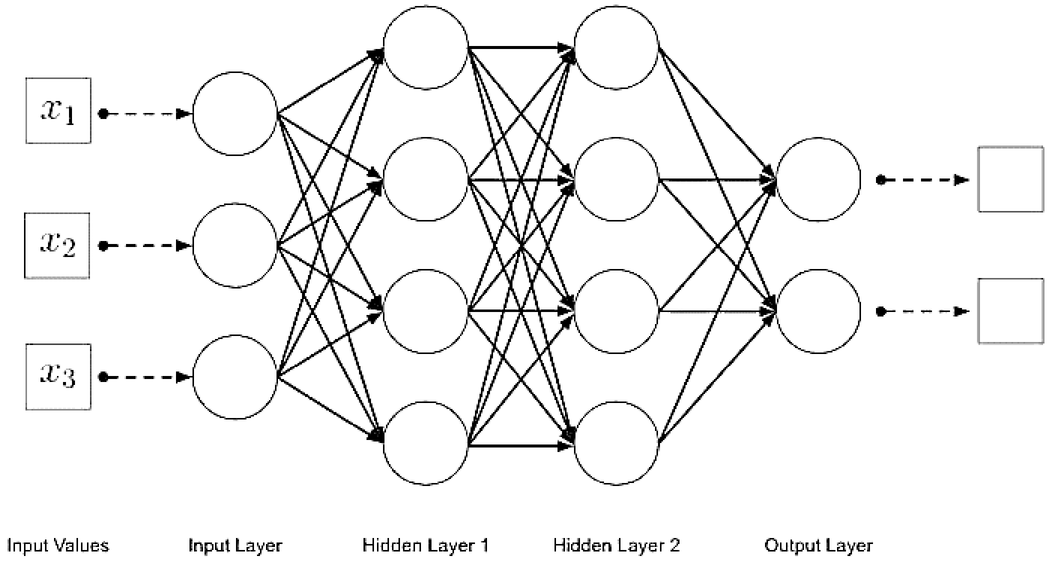


Figure 2.5The basic architecture of ANN(Uddin et al., 2019)

2.4.8. Dimensionality Reduction Techniques

To achieve good accuracy in the classification of large data, identifying and removing unwanted and unnecessary dimensions has been necessary withdimensionality reduction techniques. Data Mininghas become a critical stepused for improving computational efficiency, classification accuracy,and better visualization of dimensions (Saini, 2018). The most widely used dimensionality reduction techniquewas PCAthrough linear data only without real data efficiently for its complexity and high-dimensionality. PCA has worked with the structured and steady dataset by removing unwanted and redundant features(Saini, 2018). On the other hand, ICA was unsupervised, with high computational complexity relatedto data independence measures(Saini, 2018). Moreover, dimensionality reduction using LDA has been employed with issues lacking sample data per class that degraded the classification performancedue to the generalization of decisionswith arbitrary data and noise regularization. It has better Robustness improvement and classification performance in noisy environments (Saini, 2018).

2.4.9. Classification Performance Evaluation Metrics

The performance of a classification model has been evaluated based on statistical measures (accuracy, confusion matrix, precision, recall, and f1-scores)(Vassallo-Barco et al., 2020).

2.4.10. Summary

We have reviewed different literature on the classification and detection of skin diseases based on disease type. Theoretical backgrounds about digital image processing techniquesincluding image pre-processing, segmentation, feature extraction, and classification. Furthermore, we have givena detailed definition and analysis of the building blocks CNN architectures (convolutional layers, activation layers, pooling layers, and dropout layers). Moreover, the characteristics of the layer and the operations of each layer were explained thoroughly. Finally, we have also reviewed and presented different deep CNN architectures with outstanding results in image recognition.

2.5.Related Work

Various digital image processing works for skin disease identifications reported in the literature. The literature review has the chance to show already published papers on related skin diseases.

2.5.1. Human skin diseases identification

Jaiswal et al have proposeda CNN-based model for the Detection and Classification of Skin Diseases(Jaiswal et al., 2020). The authors have used CNN for both feature extraction and classification. Theinput layer of the proposed model was takenas the resized skin diseases image input and theimages passed through the hidden layers. The image features were extracted using theconvolution and pooling layers of the CNN and pulled into the fully connected layer for the classification with 70% accuracy. Increasing the dataset and the number of hidden layers in the CNN model has improvedthe performancein CNN but significantly increased the training time of the CNN. Moreover, theauthors suggested that CNN required little preprocessing of images.

Furthermore, Fekri-Ershad et al. haveproposed a color-based image retrieval technique to detect skin diseases (Fekri-ershad et al., 2012). However, the authors have used traditional feature extraction methods like color, shape, size, and texture features thatwere less effective to generalize and distinguish important features between highlysimilar classes.

As referenced in (Amarathunga et al., 2015)those authors proposed human skin disease detection and classification using image processing with an expert system and focused on three skin diseases. They have used thresholding segmentation for segmentation techniques and data mining techniques to identify the skin disease and to provide recommendations to users. The expert system achieves an accuracy of 85% for Eczema, 95% for Impetigo, and 85% for Melanoma. However,the authors are not using filtering techniques and deep learning approaches, especially CNN, become a powerful machinelearning technique in image processing.

The paper proposed in (Aruta et al., 2015) presents human skin diseases diagnosedusing mobile-based medical assistance with CBR and image processing. The proposed system successfully implemented six different skin diseases with an accuracy of 90%. However,the colors are traditional featuresthat are less effective to generalize and distinguish important features between highlysimilar classes when used alone.

The authors (Ubale & Paikrao, 2019)presented Detection and Classification of Skin Diseases using Different Color Phase Models. They have used KNN for classification and two-color phase models for feature extraction. However, the authors have used traditional feature extraction method colors. The colors were traditional featureswhich are less effective to generalize and distinguish important features between highlysimilar classes when used alone and KNN very limited efficiency. I.e. KNN is a lazy learner algorithm and always it computes the distance metrics due to this it is not applicable for diseases with high correlation

As referenced by (M. Kumar & Kumar, 2016)they have proposed human skin disease identification using the KNN algorithm. They have included two human skin diseases in their work (i.e. acne and psoriasis). However, this model is not effective because the processing time and computational cost become high, KNN is a distance metric since it is a lazy learner.

In (G et al., 2018)have proposed image processing techniques with a database of five common skin diseases (i.e. Acne, eczema, nail psoriasis, tinea corporis, and vitiligo). The proposed system provides skin disease detection accuracy of up to 80%. However, they have not used noise removal techniques.

According to (Wei et al., 2018)they have proposed texture and color features methods for feature extraction, SVM for classification, and GLCM for image segmentation. They have been concerned with three typesof skin diseases (i.e. herpes, dermatitis, and psoriasis) with an accuracy of 90%. However, the authors used traditional feature extraction methods for color and texture features. The color and texture features are traditional featureswhich are less effective to generalize and distinguish important features between highlysimilar classes when used alone(TÜRKOĞLU et al., 2019).

As referenced in (Handge et al., 2019)proposed image processing techniques using morphological features. The extracted features help to identify the malignant lesions from the non-malignant ones and can reach an accuracy of 80%. However, the authors used traditional feature extraction methods for color, shape, and texture features. The color, shape, and texture features are traditional featuresthat are less effective to generalize and distinguish important features between highlysimilar classes when used alone (TÜRKOĞLU et al., 2019)

Albawi et al. have proposed a skin disease identification model for three skin diseases (i.e. Melanoma, Nevus, and Atypical). Here adaptive filtering and adaptive region growing techniques have been utilized for noise removal, and segmentation, and feature extraction processes. A hybrid feature extraction method composed of 2D-DWT, geometric and texture features,with CNN for classification and prediction of skin disease were employed and reached an accuracy of 96.768%(Albawi et al., 2019). However, they have used handcrafted feature extraction that wasis less effective to generalize and distinguish important features between highlysimilar classes when used alone and CNN takes high computational time.

2.5.2. Animal skin diseases identification

Shah et al. have studiedanimal skin disease detection by using image processing operations. In this work, the prototype wasperformed based on six common skin diseases with help of KNN as feature extraction and SVM as a classifier(Shah et al., 2019). Even though the work has focused on animal skin diseases, the employed method was very limited and inefficient for the fact that. KNN wasa lazy learner algorithm and computed the distance metrics and not applicable for diseases with high correlation(Deng et al., 2016).

2.5.3. Summary

We have already presented the most recent related works for automating skin disease recognition systems to show and explain the performance and limitations of these works. Further, we have tried to show how the gaps of these related works have to been filled out. To the best of our knowledge, there was no research workconducted for recognizing skin diseases by using a hybrid of Local features and CNN. So in this thesis work, we have designed a hybrid of HOG local feature and CNN applied to recognize lumpy, dermatophylosis, dermatophytosis, and wart and fill the gaps in the previous related works (Table 2.8).

Table 2.8 Summary of related work analysis

Authors	Title	Method	Accuracy	Limitation
Jaiswal et al., 2020	Detection and classification of skin diseases	CNN for both feature extraction and classification	70%	Using CNN asa classifier increased the computational time of the model.Even though increasing the dataset and the number of hidden layers in the CNN model improvesthe

				performance of the CNN, significantly increases the training time of the CNN. For faster convergence, CNN requiresGraphics Processing Unit (GPU) to incuran additional cost
(Ubale & Paikrao, 2019	Detection and Cl assification of Skin Diseases using Different Color Phase Models	Colors for feature extraction KNN for classification	81%	The authors used traditional feature extraction methods colors. The colors have been traditional featuresthat were less efficient to generalize anddistinguis h important features between highlysimilar classes and KNN'svery limited efficiency.
Aruta et al., 2015	Skin Diseases De tection Models using Image Processing	CBR and Image processing	90.%	They have utilizedtraditional and less effective features (color, shape, size, and texture) to generalize and distinguish important features between highlysimilar classes when used alone
(Shah et al., 2019	animal skin diseases detection using image processing	KNN as feature extraction and S VM as a classifie r	91%	The method they considered was the inefficient andlazy learner (KNN algorithm)not applicable for diseases with high correlation.

CHAPTER THREE

3. METHODOLOGY (METHODS AND MATERIALS)

3.1.Introduction

The proposed cattle skin diseases identification model development possesses different image processing stages starting from the cattle skin image acquisition up-to-the diseases type identification (image acquisition, preprocessing, segmentation, feature extraction and classification).In this chapter, the detailed description of the proposed model
architecture for cattle skin diseases identification is presented. Moreover, the materials and methods used for achieving the general and specific objectives of the thesis are discussed.

3.2.CattleSkin Diseases Identification Model

The proposed cattle skin diseases identification model architecture is depictedin Figure 3.1. In general, the architecture tells the overall steps followed to identify thecorresponding class of a particular input image. The proposed system involves thefollowing major tasks of image processing. These are image acquisition, preprocessing, segmentation, feature extraction,
and classification. Image acquisition is the first step in every image processing system.

As shown in Figure 3.1, there are two phases in the system. The training phase and the testing phase. The training phase is started before the testing phase for training the model. The testing phase follows the training phase for testing the performance of the trained a model with a new image dataset.

The proposed system starts by taking the whole cattle skin image dataset for preprocessing.In the preprocessing phase, resizing, noise removal by Gaussian filtering, median filtering, and Gabor filter, and contrast enhancement by histogram equalization are applied. Afterpreprocessing, image augmentation is followed. Image augmentation is required forincreasing the dataset to prevent overfitting. Then, the augmented images are subdividedinto train dataset and test dataset. Next, both the train dataset and test dataset are segmentedusing threshold segmentation technique. And in parallel, the local features of theimages in both datasets are detected using HOG. The next step is feature extraction, usingCNN, which is used to extract the useful high-level features of the cattle skin image. Lastly,in the training phase, the high-level features and local features are fused together and feedin to the classifier to train the SVM classifier. While in the test set, similarly both featuresare fused together and feed to the previously trained SVM model. The detail of eachphase of the proposed system is presented in the following section.

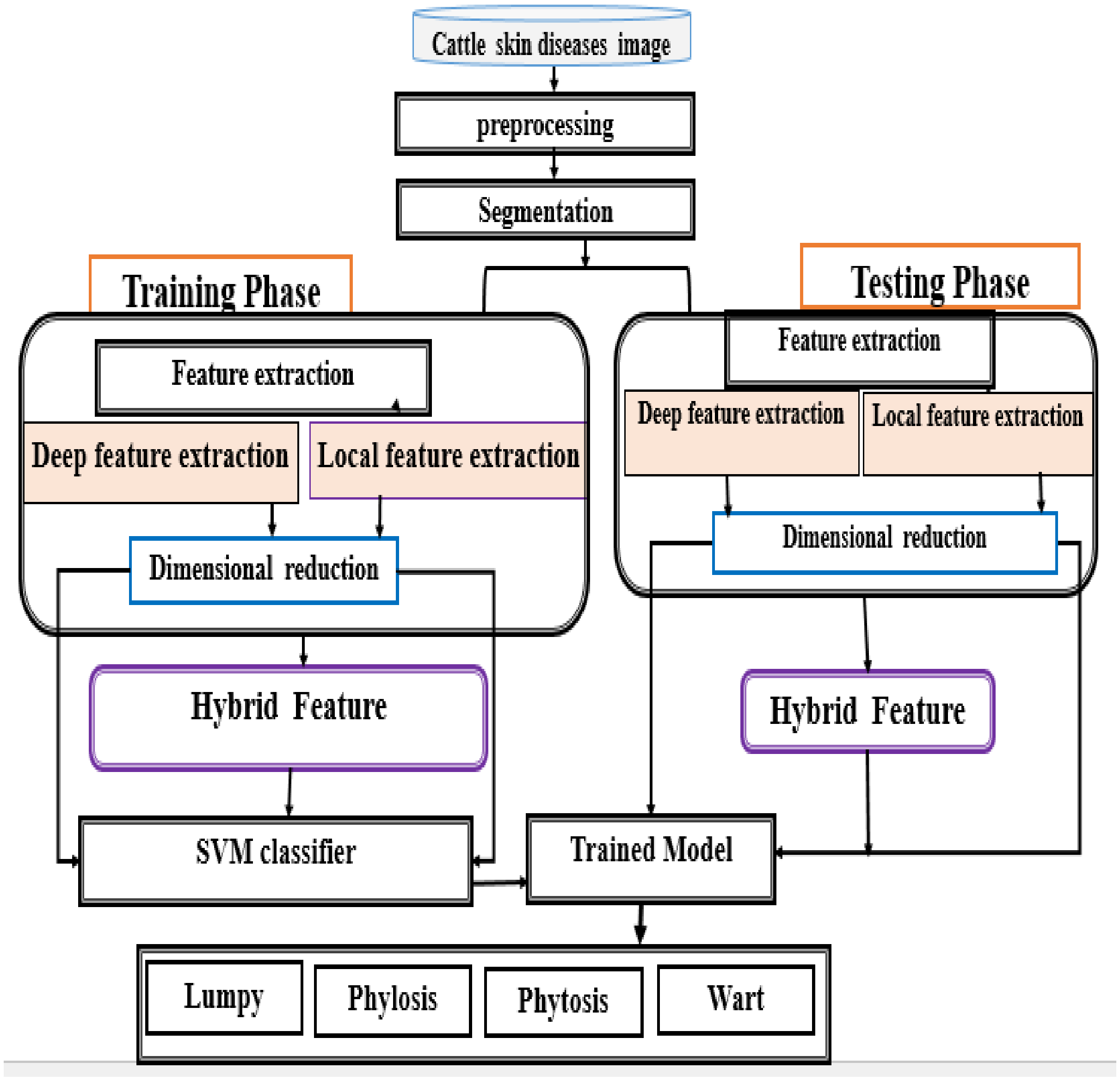


Figure 3.1 The proposed model architecture

3.2.1. Image acquisition

In this study, we use a Huawei Y635 model smartphone memory size 8GB and a resolution of 480x854 for capturing cattle skin diseased images in the form of JPG (Joint Photographic Group) file format. During image acquisition, the proper focus of the camera was essential. Thus, when the camera was not properly focused, we have got blurred images.Therefore to manage the camera we have to manage the camera shutter speed based on the object movement (speed) and place (i.e. crowded or not) and when we capture the images, we fix the smartphone on a stand to reduce the hand movement. Based on the work done by(Mengistu et al., 2016), we capture the image with a 130mm distance from the cattle skin up to the camera for better image quality. The cattle skin diseased images are sampled from BahirDar zuria woreda veterinary clinic and BahirDar university agricultural and environmental campus. The labeling process is done by experts found in the two organizations because that some of the diseases look similar depending on the infection status. The total number of original images collected for this study is 765 and we have used 2000 augmented images 500 input images for each class.

3.2.2. Image Preprocessing

Different artifacts and/or noises may exist in our collected image dataset. They are not suitable for the identification of necessary features for the classification. Preprocessing the stage takes the raw image as input and producesan enhanced image as an output. In this thesisresearch, in the preprocessing stage, we resize the image; we remove noises using filteringtechniques; and we enhance the contrast of the images with histogram equalization. Thealgorithms for implementing these preprocessing processes are described in the followingsub-sections.

3.2.2.1.Image resizing

The collected images are varied in size. Image resizing is the process of converting the size
of the original image into a constant image size. We have selected 224 x 224 image size because of that we compare different image sizes and we have achieved a better result in 224x224, whichwas initially designed to work on ImageNet dataset of size 224 x

224. This standard image sizeis used throughout the whole image processing. To resize an image, we are using the OpenCVpython library. The OpenCV has cv2.imread () function to read an image in NumPy arrayformat that the Keras works on and cv2.resize () for image resizing.

There are different interpolation techniques for resizing/rescaling an image by preserving the important image content as discussed in section 2.4.2.1. Some of them are nearest neighbor interpolation, bilinear interpolation, and BiCubic interpolation. Among these, Bicubic interpolation provides an ideal computational time and output image quality than the other; thus, it is popular in many real application areas like Adobe Photoshop and printerdrivers(Parsania & Virparia, 2016). The algorithm that we followed for image resizing is depicted in Algorithm 3.1.

Algorithm 3. 1: Bicubic image resizing algorithm

Input image
Im = image. Open (input image)
Width = ()
Height =()
Img =im.resize((width, height), image.BICUBIC)
Img.save(Img)
Output image
Where Im and Img are variables

3.2.2.2.Color change

Though the input images were RGB, we have used both the RGB and grayscale colors sothat changing the color from RGB to gray wasmandatory. The relationship between the RGB and grayscale color has been performed here as shown below in algorithm 3.2 and figure 3.2.

Algorithm 3. 2:RGB to Grayscale color change

Input image
Img=image. Open (input image)
gray =image .convert('L')
Gray. Save()saved path
Where img is variable

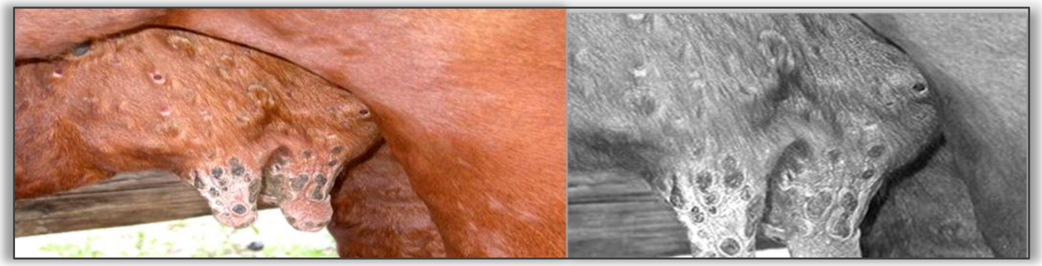


Figure 3.2 RGB to the gray color change of animal skin diseases

3.2.2.3.Image enhancement techniques: histogram equalization

The principal objective of image enhancement in this work was to modify attributes of cattle skin diseases of an image to make it more suitable for segmentation tasks. Histogram equalization has been an image enhancement techniqueused to adjust the image contrast using the image's histogram. This method has beenused toequally distribute various pixel intensities over the entire image allowing lower localcontrast areas to gain a higher contrast. We have implemented the histogram equalizationmethod on the previously resized image using the OpenCVcv2.equalizeHist () module onYUV color space.

Algorithm 3. 3:Algorithm of histogram equalization

Read the filtered image img
Convert the image color RGB to YUV color space using cv2.cvtColor()
Equalize the image histogram using cv2.equalizeHist()
Convert the image color back to RGB

Return equalized image img

3.2.2.4.Noise reduction

Different types of image noises make images unreadable and cause a barrier in many applications of image processing. Noise removal is an important task of image processing which affects the classification accuracy. In most cases, digital camera images are distorted
with Gaussian noise and salt and pepper noise(Navjeet et al., 2016)(Reshma & Shan, 2018). Thus, we areproposed to compare median filtering (remove impulsive noises like salt andpepper noise and smooth other noises), Gaussian filtering (remove Gaussian noise), andGabor filter to observe the suitable image filtering technique.

Gaussian filter

Gaussian filtering is one of thenoise reduction technique that usually used for smoothing Gaussian noises. It is a linear convolution filtering that works by convolving the 2D Gaussian kernel over the entire image. It takes the histogram equalized image and distributes all pixels of the image as a Gaussian distribution by convolving a 2D 5 x 5 Gaussian kernel over the entire image. The Python OpenCV library has cv2.GaussianBlur () function was used for implementing Gaussian filtering and the algorithm is shown in Algorithm 3.4.

Algorithm 3. 4:Gaussian filter algorithm

Read the resized image img
Select kernel size k
Compute kxk 2D Gaussian kernel using the 2D Gaussian function
Convolve the 2D Gaussian kernel on the entire image
Return filtered image img

Median filter: is a nonlinear statistical filtering approach used for removing noises from an image. The median filter is better than the mean filter. The median filtering works by moving in the image pixel by pixel and replacing each pixel value with the median value of neighborhood pixels. The neighboring pixels together are called windows and it moves pixel by pixel all over the image. Typically, median filtering can be used for removing the salt and pepper noise. Median filtering keeps the edges of the image when it removes the noise. As a result, the image is not blurred as other filtering methods.

Algorithm 3. 5:Algorithm of a median filter

Step1: Read the resized image img
Step2: Select a 2D kernel k of size 5x5
Step3: Compare the central pixel at coordinates (x,y) with the neighboring pixels within the kernel
Step4: If the central pixel found to be unique from neighboring pixels go to step 5 otherwise, go to step 6
Step5: compute the median value Kmed of the pixels within the kernel
Step6: Replace the central pixel at (x,y)with the median value kmed
Step7: if all pixels within the image is not processed go to step2, otherwise, go to step8
Step8: return filtered image img

Gabor filter

Gabor filter is the linear filter thatanalyses whether there is any specific frequency content in the image in specific directions in a localized region around the point or region of analysis.Gabor filtering is commonly usedfor texture analysis.Algorithm 3.6 below shows how the Gabor filter works in short.

Algorithm 3. 6: Algorithm of Gabor Filter

Step1: read the resized image
Step2: get gabor kernel k size (ksize,ksize), theta,ktype,lambda
Step3: append the parameter in step2
Step4: return filters

3.2.3. Image augmentation

Image augmentation is one of the image processing technique to increase the dataset so that to overcome the overfitting problem and train the proposed model very well. Since the original four cattle skin diseases image collected for this thesis is not sufficient to trainthe model very well, we used the augmentation technique for increasing the dataset. There are a lot of augmentation techniques like zooming, cropping, flipping, rotation, shifting, and others. Among these techniques, we used rotation, zooming, flipping, and shifting because other techniques work by removing some part of the image to augment it.

Algorithm 3. 7:Algorithm of image data augmentation

Step1: read the enhanced and filtered image
Step2: a select range of rotation, zooming, shift, crop, flip horizontal
Step3: write augmented image in the new path
Step4: return augmented image

3.2.4. Image Segmentation

Image segmentation is an important phase and it dividesimage categories, which correspond to parts of objects.Segmentationis used to identify objects and relevant information from digital images. The methods we have used threshold segmentation to segment the diseased part from the normal skin. This segmentation aims to reduce the information to enable easy analysis.

3.2.4.1.Segmentation using threshold

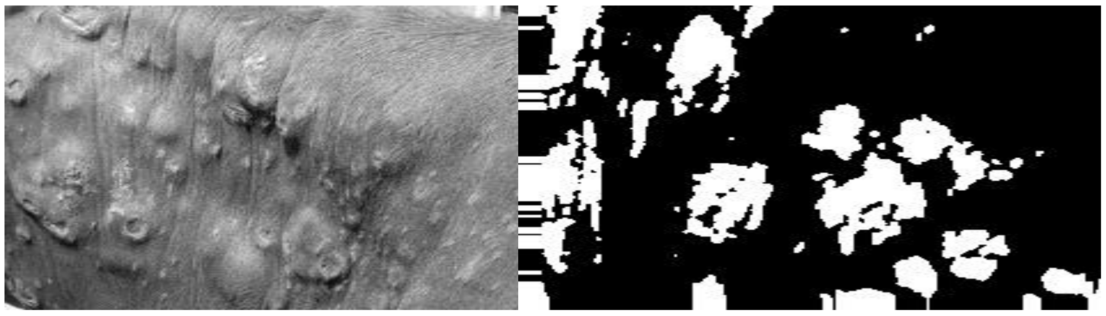


Figure 3.3 The normal and segmented image using threshold segmentation

3.2.5. Dimensionality Reduction with PCA

Dimensionality reduction in large-scale image research plays an important role for their performance in different applications. In this paper, we have used Principal Component Analysis as a dimensionality reduction method.

Algorithm 3. 8: Algorithm of PCA

Step1: get the extracted feature data
Step2: give the data structure
Step3: standardize the data
Step4: get covariance of Z
Step5: calculate Eigenvectors and Eigenvalues
Step6: sort the Eigenvalues
Step7: calculate the new features
Step8: drop unimportant features from the new set

3.2.6. Feature Extraction using CNN

Feature extraction describesthe objects to reduce a form suitable for computer processing. The weight inthe convolutional layer is used to control the capacity and to decrease the model complexity. In a detailed discussion, the CNN has three common layers, these are:

The Convolutional Layer: in CNN architecture convolution is the first layer since, in this layer,the convolution operation is performed and weights are a set of learnable filters created randomly and learned through the back-propagation algorithm. The outcome every convolved is a feature map.

Arrangement of filter size: In the proposed CNN model we have test five models, the first model is 3x3, 5x5, and 1x1, the second model 3x3, 5x5, 3x3, and 1x1, the third model 3x3, 3x3, 5x5, and 7x7, the fourth model 3x3,5x5, 7x7,3x3, and 1x1, the fifth model 3x3, 5x5, 7x7, and 3x3 filter sizes have been tested.Wehave used filter sizes of 3x3, 5x5, 7x7, and 3x3. This filter size is selected after testing each filter size in the model and we achieved better accuracy using a combination of these filter sizes.

Selection of Stride Number: we have tested the model, by analyzing the effects of stride number in each layer. So,in this work,we have used stride number = 2 to skip two pixels to reduce the dimension of feature vectors when the size of the feature vector is large. We also use normal stride number = 1(one) to skip one pixel at once during convolution operation.

Activation function layer: It is a widely used activation function and chiefly implemented in hidden layers of neural networks. In this research we have used the ReLu functionwhich makes the Equation *A(x) = max (0, x)*, the output is x where x is positive including zero. The ReLU function is the range between **[**0, inf) and nonlinear, which means we can backpropagate the errors easily and multiple layers of neurons being activated.Compared with other functions ReLu is less computationally expensive and faster,due to simpler mathematical operations.

The Pooling operation Layer: which implements a downsampling operation to decrease the size of convolutional layers. Using a determined pooling mask and pooling operation that appliedon the pooling layer. When we have developed the CNN model, we have testedthe effects of average pooling and maximum pooling during experimentation. We haveused a max-pooling size of 3x3 after the convolution operation is performed. This reducesthe dimension of the data by selecting the maximum values. The operation of the max-pooling layer is performed by selecting important features and reducing otherfeatures. In our CNN feature extraction model, we also used the average pooling size of 2x2 to reducethe dimension of the feature vector by taking the average of the feature vectors. In ourmodel using maximum poling and average pooling achieves better accuracy rather thanusing them individually.

Fully connected layers: which used the extracted features in the preceding layers to do the classification task. The result of the last convolutional or pooling layer is fed to the fully connected layers like in an original neural network. The number of convolution l ayer, type of the pooling operation, number of filters, and the
filter size is selected by experiment.

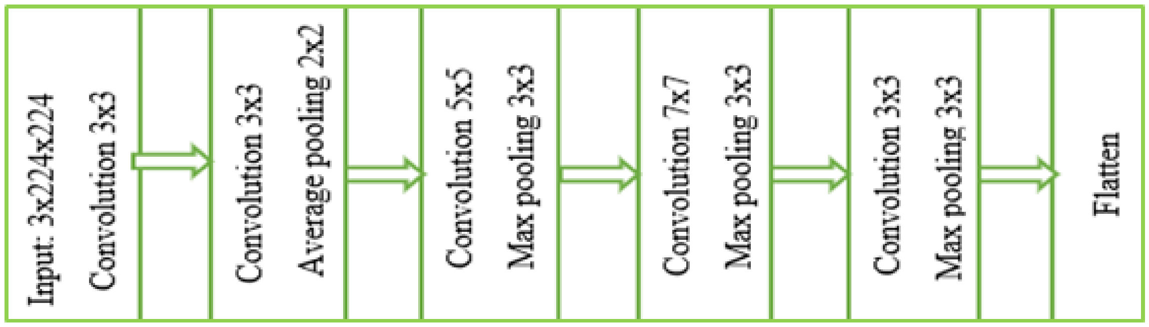


Figure 3.4. The proposed CNN Model for Feature Extraction of cattle skin diseases

Algorithm 3. 9: The process of extracting features vectors of cattle skin diseases in CNN

```
Initialization:  pre-processed image =img,  number of filters=nb_filters, img_rows,
img_cols, pool size =nb_pool, filter size= fs,  padding=p, image channel= img_channel
dropout probabilt= dp, Number of class=numclass, stride number=sn
start:
// first block of the model
Enter pre-processed image
Apply convolution operation(nb_filters, fs, sn, p, img_rows, img_cols, img_channel)
Apply activation function, ReLU on output
// second block of the model
Apply convolution operation(nb_filters, fs, sn, p)
Apply average pooling operation(nb_pool, sn)
Apply activation function, ReLU
Apply dropout operation (p)
// third block of the model
Apply convolution operation (nb_filters, fs, sn, p)
Apply max-pooling operation (nb_pool, sn)
Apply activation function, ReLU on output
Apply dropout operation (p)
//forth block of the model
Apply convolution operation (nb_filters, fs, sn, p)
Apply max-pooling operation (nb_pool, sn)
Apply activation function, ReLU on output
Apply dropout operation (p)
// fifth block of the model
```

```
Apply convolution operation (nb_filters, fs, sn, p)
Apply max pooling operation(nb_pool, sn)
Apply activation function,  ReLU on output
Apply dropout operation(p) Add flatten layer// used to change multidimensional feature
vector to one dimension
Apply fully connected layer (numclass) // which takes the number of class which is
directly entered to a classifier
Return model. Predict
Save the extracted vector
End
```

As it is described in the above algorithm there are many operations performed to extract the feature vector of the cattle skin diseases. After the convolution operation and pooling operation performed matrix feature vector entered in to flatten layer of our model. The flatten layer changes the matrix feature vector into a one-dimensional array. This feature vector is the input for the classification layer.

3.2.7. Feature Extraction using Local feature descriptor

There are different local feature descriptors for image processing since the most common are SIFT, SURF, HOG, and LBP(Kabbai et al., 2019). But SURF is the enhancement of SIFT due to this we used SURF. In this work, we have compared those feature extraction techniques to extract the local feature of cattle skin diseases. As referenced by (Kabbai et al., 2019) those local features much easier and quicker to compute than other feature descriptors. Furthermore, those descriptors were achieved better performance for the smaller datasetand those are invariant to affine transformation(Kabbai et al., 2019).

3.2.8. Feature Vector concatenation

After extracting the features of cattle skin diseases using CNN and HOG we have combined them into one feature vector by concatenating them into one feature set. The combined feature vector extracted using HOG and feature vectors extracted with CNN is a hybrid feature vector of the cattle skin diseases feature. The following algorithm 3.10 below shows feature vector concatenation of HOG feature vectors and CNN-based feature vectors of cattle skin diseases.

Algorithm 3. 10: Combination of Feature vectors extracted by CNN and HOG

For each image in the dataset
CNNFV = model. Predict(image)
HOGFV = hog. Compute(image)
HFV = concatenate(CNNFV, HOGFV)
HFV =hstack(HFV)
Return HFV
Where: CNNFV =feature vector extracted by CNN
HOGFV =feature vector extracted by HOG
HFV = hybrid feature vector

3.2.9. Classification using SoftMax classifier

We have addeda dense layer with the Softmax activation function of the CNN for classifying the input images based on their features extracted in the final fully-connected layer of the proposed CNN. The Softmax first learns important features of each category in the training phase from the training set. Then, it classifies the new testing dataset accordingly. Choosing proper loss and optimization function increases the power of the classification of the Softmax classifier. The algorithm we followed to implement the Softmax classifier is shown in Algorithm 10 below.

Algorithm 3. 11: Algorithm for SoftMax classifier

Step1: get the extracted features f
Step2: split f into train and test set
Step3: initialize the loss, learning rate, optimization function, and epoch
Step4: train the softmax with the train set in a specified epoch
Step5: evaluate the softmax with the test set
Step6: return class label

3.2.10. Classification using SVM classifier

SVM classifier is a binary classifier, for multi-class classification,we usedthe two strategies (i.e. one versus rest (all), and one versus one) and kernel functions due to that forcattle skindiseases identification we have a one versus one strategy and compared those kernel functions. From that SVM with kernel function RBF achieved a good classification accuracy. Therefore, we used the RBF function throughout theexperimentto classify the animal skin diseases into four classes, these are dermatophylosis, dermatophytosis, lumpy, and wart.

Algorithm 3. 12: Algorithm of an SVM classifier

Step1: get the extracted feature vector f
Step2: split f into train and test set
Step3: Initialize the kernel, gamma, and other SVM parameters
Step4: train the SVM with the train set
Step5: evaluate the SVM with the test set
Step6: return class label

3.2.11. Summary

In this chapter, we have discussed the design of the convolutional neural network for the automatic identification of cattle skin diseases thoroughly. The components that compose the model (preprocessing, segmentation, feature extraction using convolutional neural network and HOG, classification using SVM), the relationship among the components, and the responsibility of each component are described in detail. The classification was done in two phases (the training phase, and the testing phase). The training phase has useda training dataset to form the learning model and the testing phase has used a testing dataset passed via the same procedures as in the training phases.

CHAPTER FOUR

4. RESULTS AND DISCUSSION

4.1.Introduction

In this chapter, an experimental evaluation of the proposed model for automatic detection and classification of cattle skin diseases was described in detail. Experimental evaluationapproved the realization of the proposed model. The dataset used and theimplementation of the proposed model were described thoroughly. The effect of the comparison filtering techniques evaluated and compared with other state-of-the-art models was discussed in detail.

4.2.Dataset preparation

For this study, cattle skin disease images classified basedon their disease type taken from the collected image stored in ASDD&C our local folder. In the image acquisition phase, we have taken images of the four skin diseases of cattle (lumpy, wart, dermatophylosis, and dermatophytosis).

Table 4.1cattle skin diseases data description

Serial No.	Types of disease	Total No. of the original image	Total No. Of augmented image		Spatial Resolution	Image Format
1	Lumpy	200	500		224x224	Jpg
2	Phylosis	160	500		224x224	Jpg
3	Phytosis	240	500		224x224	Jpg
4	Wart	165	500		224x224	Jpg
Total	-	765	2000		224x224	Jpg

4.3.Implementation

Experiments were done based on the prototype developed with Keras (TensorFlow as a backend) on Intel Core ™ i5-6200 CPU, and 8 GB of RAM. The model is trained for 50 epochs, a batch size of 32, and a starting or initial learning rate of 0.001 (1e-3). The total dataset (2000)was first partitioned into training and testing dataset 80/20 which means that 80 percent of the data(1600) is assigned for training the model and 20 percent of the data (400) is allocated for testing the trained model.Also, the training dataset splits into80/20 which means 20 presents(320) from the training for validation data.

4.4.An experiment on CNN Using Softmax Classifier

We have tested the effects of image size in CNN models because different image sizes were selected in different literature. Then, we have conducted a different experiment on the proposed end-to-end CNN model. In the CNN model, there is no common standard to select the parameter, since we have selected the parameters by trial and error method.

4.4.1. The Effects of Different Image Size Using SoftMax Classifier

There was no fixed standard image size for CNN, so we have tested an experiment on different input image size. In this experiment we have compared 200x200, 224x224, 300x300, and 360x360 input image sizes. The following table showed a comparison of input image size.

Table 4.2 The effect of different image size

Image size	200x200	224x224	300x300	360x360
Accuracy	94.75%	96.5%	90%	97%
Epoch /time taking	240sec	180sec	360sec	400sec

Image size with 360×360 shows better performance and high training time. However, 224x224 relatively better training accuracy and speed, so in this experiment, we have selected image size with 224×224 since it has better training timebecause in the health sector a few seconds has a large effect on health.

4.4.2. The Effects of Different Activation Function in CNN model

The Activation Functions in a neural network can be mainly divided into two types: thoseare Linear and Non-linear Activation Functions. Non-linear Activation Functions consist of Soft sign, sigmoid, Tanh, and ReLu. In the experiment, we have seen the effect of those activation functions on the CNN model.

Table 4.3 the effect of activation functions on our CNN model

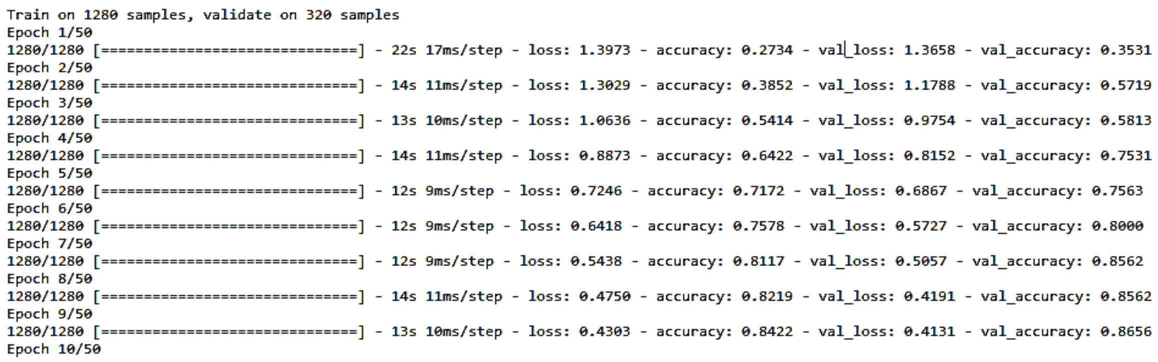
Activation function		Tanh	Softsign	ReLU
Accuracy	Training	97%	98%	98.75%
	Testing	94.25%	95%	96.5%
Time taking/epoch		360sec	300sec	180sec

4.4.3. CNN Model Training and Validation without filtering technique

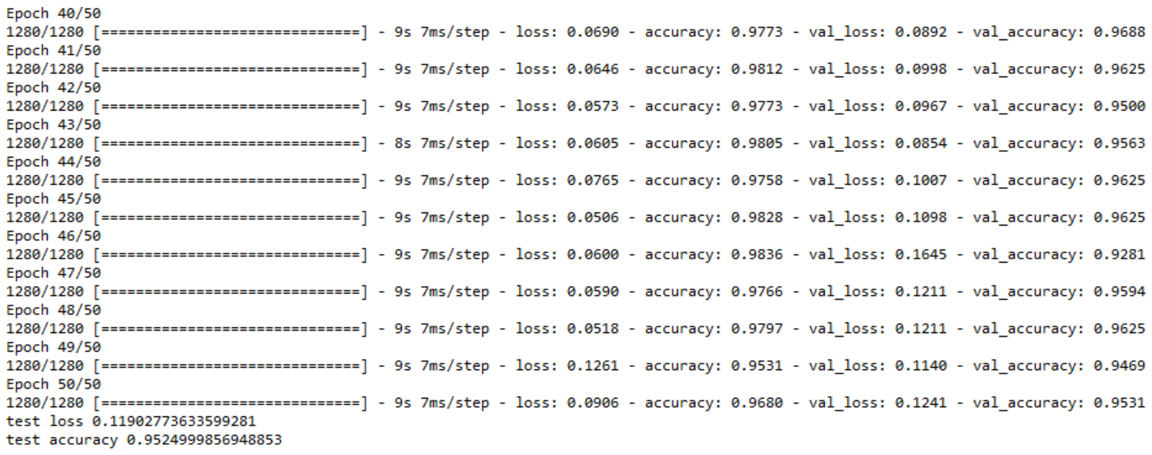
CNN feature extraction without filtering technique in cattle skin diseases image attained 95.3% training accuracy and 90.5% testing accuracy. This classification accuracy was obtained when the model was trained without filtering technique.

4.4.4. The model Training and Validation Phase with Gaussian Filter

As clearly depicted inthe figure below, end-to-end CNN, obtained 97.4% training accuracy and 95% testing accuracy. This classification accuracy was obtained when the model was trained with batch normalization, data augmentation, Gaussian filter, and dropout at each stage. Applying batch normalization after each activation layer in the network gavea more stable loss curve. We have also got better accuracy after batch normalization.



```
Train on 1280 samples, validate on 320 samples
Epoch 1/50
1280/1280 [==============================] - 22s 17ms/step - loss: 1.3973 - accuracy: 0.2734 - val_loss: 1.3658 - val_accuracy: 0.3531
Epoch 2/50
1280/1280 [==============================] - 14s 11ms/step - loss: 1.3029 - accuracy: 0.3852 - val_loss: 1.1788 - val_accuracy: 0.5719
Epoch 3/50
1280/1280 [==============================] - 13s 10ms/step - loss: 1.0636 - accuracy: 0.5414 - val_loss: 0.9754 - val_accuracy: 0.5813
Epoch 4/50
1280/1280 [==============================] - 14s 11ms/step - loss: 0.8873 - accuracy: 0.6422 - val_loss: 0.8152 - val_accuracy: 0.7531
Epoch 5/50
1280/1280 [==============================] - 12s 9ms/step - loss: 0.7246 - accuracy: 0.7172 - val_loss: 0.6867 - val_accuracy: 0.7563
Epoch 6/50
1280/1280 [==============================] - 12s 9ms/step - loss: 0.6418 - accuracy: 0.7578 - val_loss: 0.5727 - val_accuracy: 0.8000
Epoch 7/50
1280/1280 [==============================] - 12s 9ms/step - loss: 0.5438 - accuracy: 0.8117 - val_loss: 0.5057 - val_accuracy: 0.8562
Epoch 8/50
1280/1280 [==============================] - 14s 11ms/step - loss: 0.4750 - accuracy: 0.8219 - val_loss: 0.4191 - val_accuracy: 0.8562
Epoch 9/50
1280/1280 [==============================] - 13s 10ms/step - loss: 0.4303 - accuracy: 0.8422 - val_loss: 0.4131 - val_accuracy: 0.8656
Epoch 10/50
```



```
Epoch 40/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0690 - accuracy: 0.9773 - val_loss: 0.0892 - val_accuracy: 0.9688
Epoch 41/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0646 - accuracy: 0.9812 - val_loss: 0.0998 - val_accuracy: 0.9625
Epoch 42/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0573 - accuracy: 0.9773 - val_loss: 0.0967 - val_accuracy: 0.9500
Epoch 43/50
1280/1280 [==============================] - 8s 7ms/step - loss: 0.0605 - accuracy: 0.9805 - val_loss: 0.0854 - val_accuracy: 0.9563
Epoch 44/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0765 - accuracy: 0.9758 - val_loss: 0.1007 - val_accuracy: 0.9625
Epoch 45/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0506 - accuracy: 0.9828 - val_loss: 0.1098 - val_accuracy: 0.9625
Epoch 46/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0600 - accuracy: 0.9836 - val_loss: 0.1645 - val_accuracy: 0.9281
Epoch 47/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0590 - accuracy: 0.9766 - val_loss: 0.1211 - val_accuracy: 0.9594
Epoch 48/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0518 - accuracy: 0.9797 - val_loss: 0.1211 - val_accuracy: 0.9625
Epoch 49/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.1261 - accuracy: 0.9531 - val_loss: 0.1140 - val_accuracy: 0.9469
Epoch 50/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0906 - accuracy: 0.9680 - val_loss: 0.1241 - val_accuracy: 0.9531
test loss 0.11902773633599281
test accuracy 0.9524999856948853
```

Figure 4.1. Training and validation accuracy of the CNN model with Gaussian filter

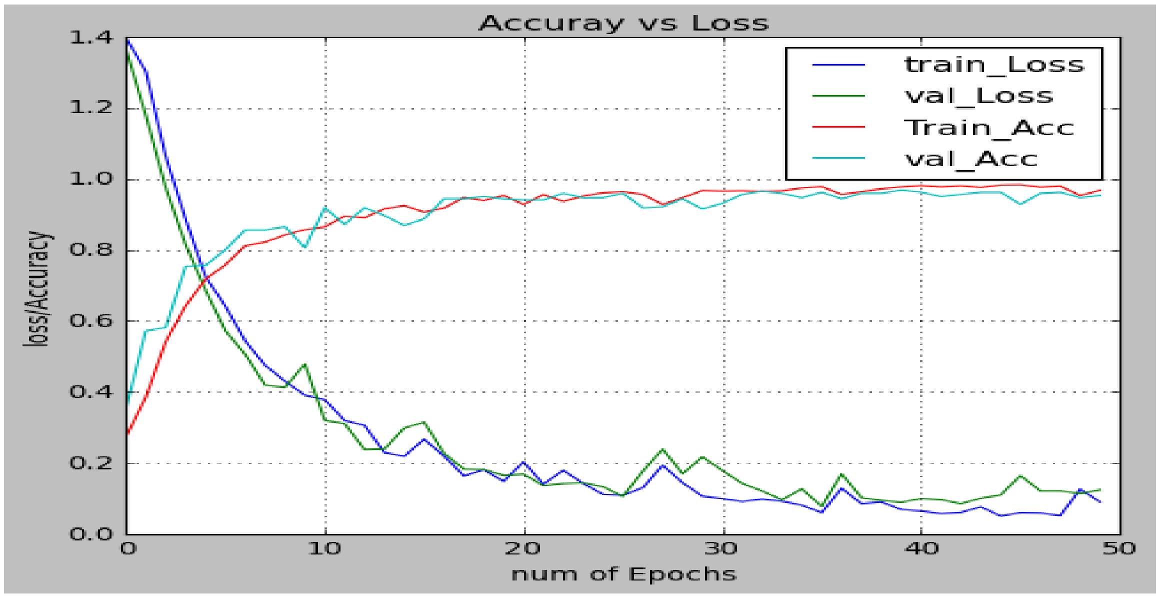
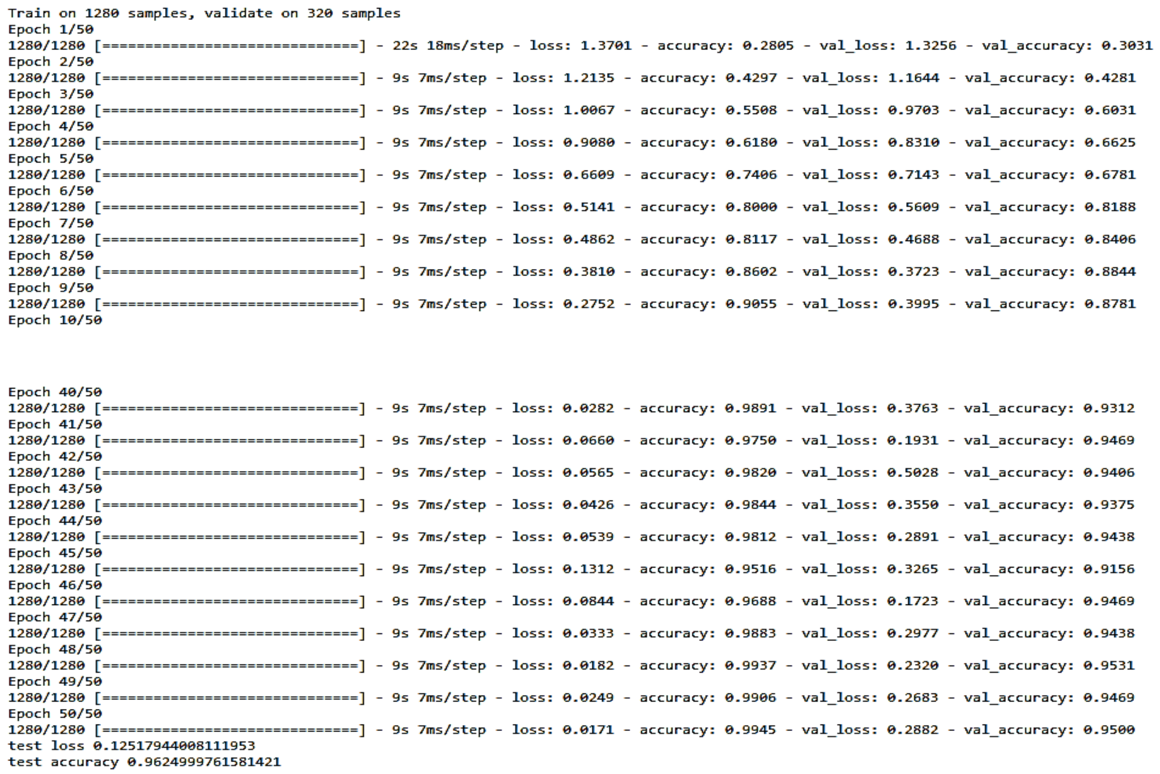


Figure 4.2. Accuracy and loss curve of end-to-end CNN model with Gaussian filter

4.4.5. The Model during Training and Validation Phase: with Gabor filter

As clearly depicted in the figure below, the model has obtained 98.75 % training accuracy and 96.5% testing accuracy. This classification accuracy was obtained when the model was trained with batch normalization, data augmentation, Gabor filter, and dropout at initial stages. We have got a more stable loss curve through the application of batch normalization after each activation layer in the network and better accuracy after batch normalization.



```
Train on 1280 samples, validate on 320 samples
Epoch 1/50
1280/1280 [==============================] - 22s 18ms/step - loss: 1.3701 - accuracy: 0.2805 - val_loss: 1.3256 - val_accuracy: 0.3031
Epoch 2/50
1280/1280 [==============================] - 9s 7ms/step - loss: 1.2135 - accuracy: 0.4297 - val_loss: 1.1644 - val_accuracy: 0.4281
Epoch 3/50
1280/1280 [==============================] - 9s 7ms/step - loss: 1.0067 - accuracy: 0.5508 - val_loss: 0.9703 - val_accuracy: 0.6031
Epoch 4/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.9080 - accuracy: 0.6180 - val_loss: 0.8310 - val_accuracy: 0.6625
Epoch 5/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.6609 - accuracy: 0.7406 - val_loss: 0.7143 - val_accuracy: 0.6781
Epoch 6/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.5141 - accuracy: 0.8000 - val_loss: 0.5609 - val_accuracy: 0.8188
Epoch 7/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.4862 - accuracy: 0.8117 - val_loss: 0.4688 - val_accuracy: 0.8406
Epoch 8/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.3810 - accuracy: 0.8602 - val_loss: 0.3723 - val_accuracy: 0.8844
Epoch 9/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.2752 - accuracy: 0.9055 - val_loss: 0.3995 - val_accuracy: 0.8781
Epoch 10/50


Epoch 40/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0282 - accuracy: 0.9891 - val_loss: 0.3763 - val_accuracy: 0.9312
Epoch 41/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0660 - accuracy: 0.9750 - val_loss: 0.1931 - val_accuracy: 0.9469
Epoch 42/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0565 - accuracy: 0.9820 - val_loss: 0.5028 - val_accuracy: 0.9406
Epoch 43/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0426 - accuracy: 0.9844 - val_loss: 0.3550 - val_accuracy: 0.9375
Epoch 44/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0539 - accuracy: 0.9812 - val_loss: 0.2891 - val_accuracy: 0.9438
Epoch 45/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.1312 - accuracy: 0.9516 - val_loss: 0.3265 - val_accuracy: 0.9156
Epoch 46/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0844 - accuracy: 0.9688 - val_loss: 0.1723 - val_accuracy: 0.9469
Epoch 47/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0333 - accuracy: 0.9883 - val_loss: 0.2977 - val_accuracy: 0.9438
Epoch 48/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0182 - accuracy: 0.9937 - val_loss: 0.2320 - val_accuracy: 0.9531
Epoch 49/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0249 - accuracy: 0.9906 - val_loss: 0.2683 - val_accuracy: 0.9469
Epoch 50/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0171 - accuracy: 0.9945 - val_loss: 0.2882 - val_accuracy: 0.9500
test loss 0.12517944008111953
test accuracy 0.9624999761581421
```

Figure 4.3. Training and validation accuracy of the CNN model applying Gabor filter

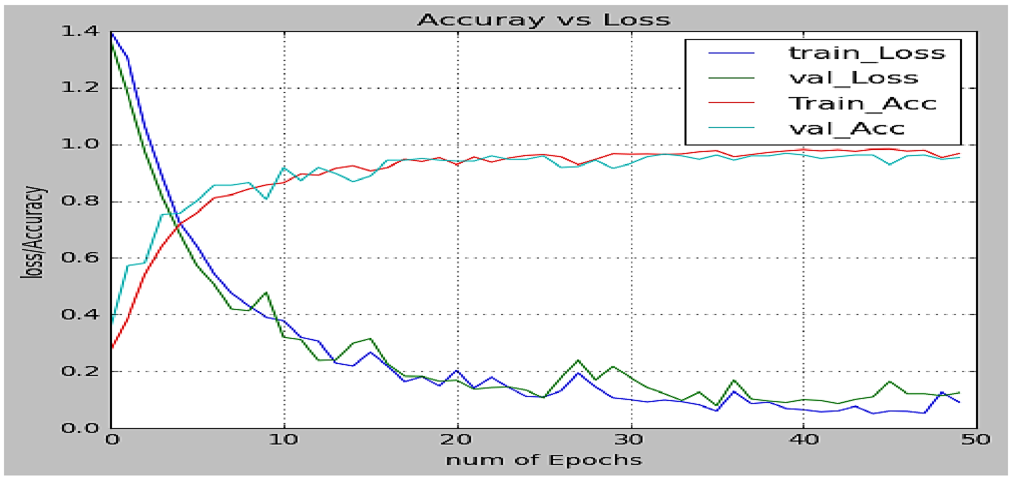
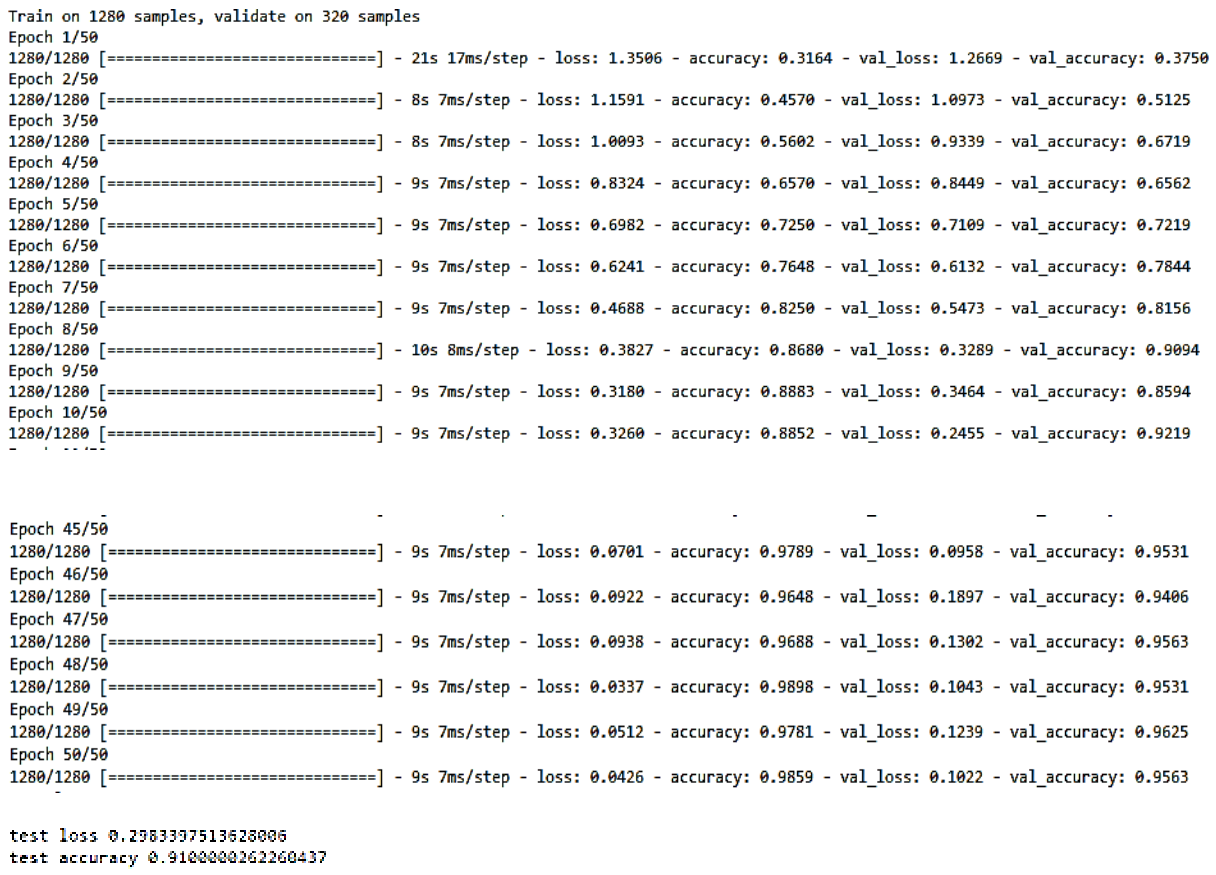


Figure 4.4. Accuracy and loss of end-to-end CNN model with Gabor filter

4.4.6. A CNNduring Training and Validation Phase Applying Median filter

As clearly depicted in the figure belowthe model has obtained 95.9% training accuracy and 91% testing accuracy. This classification accuracy was obtained when the model was trained with batch normalization, data augmentation, the median filter, and dropout at each stage. We have got a more stable loss curve through the application of batch normalization after each activation layer in the network and better accuracy after batch normalization.



```
Train on 1280 samples, validate on 320 samples
Epoch 1/50
1280/1280 [==============================] - 21s 17ms/step - loss: 1.3506 - accuracy: 0.3164 - val_loss: 1.2669 - val_accuracy: 0.3750
Epoch 2/50
1280/1280 [==============================] - 8s 7ms/step - loss: 1.1591 - accuracy: 0.4570 - val_loss: 1.0973 - val_accuracy: 0.5125
Epoch 3/50
1280/1280 [==============================] - 8s 7ms/step - loss: 1.0093 - accuracy: 0.5602 - val_loss: 0.9339 - val_accuracy: 0.6719
Epoch 4/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.8324 - accuracy: 0.6570 - val_loss: 0.8449 - val_accuracy: 0.6562
Epoch 5/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.6982 - accuracy: 0.7250 - val_loss: 0.7109 - val_accuracy: 0.7219
Epoch 6/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.6241 - accuracy: 0.7648 - val_loss: 0.6132 - val_accuracy: 0.7844
Epoch 7/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.4688 - accuracy: 0.8250 - val_loss: 0.5473 - val_accuracy: 0.8156
Epoch 8/50
1280/1280 [==============================] - 10s 8ms/step - loss: 0.3827 - accuracy: 0.8680 - val_loss: 0.3289 - val_accuracy: 0.9094
Epoch 9/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.3180 - accuracy: 0.8883 - val_loss: 0.3464 - val_accuracy: 0.8594
Epoch 10/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.3260 - accuracy: 0.8852 - val_loss: 0.2455 - val_accuracy: 0.9219

Epoch 45/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0701 - accuracy: 0.9789 - val_loss: 0.0958 - val_accuracy: 0.9531
Epoch 46/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0922 - accuracy: 0.9648 - val_loss: 0.1897 - val_accuracy: 0.9406
Epoch 47/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0938 - accuracy: 0.9688 - val_loss: 0.1302 - val_accuracy: 0.9563
Epoch 48/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0337 - accuracy: 0.9898 - val_loss: 0.1043 - val_accuracy: 0.9531
Epoch 49/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0512 - accuracy: 0.9781 - val_loss: 0.1239 - val_accuracy: 0.9625
Epoch 50/50
1280/1280 [==============================] - 9s 7ms/step - loss: 0.0426 - accuracy: 0.9859 - val_loss: 0.1022 - val_accuracy: 0.9563

test loss 0.2983397513628006
test accuracy 0.9100000262260437
```

Figure 4.5. Training and testing accuracy of the CNN model applying a median filter

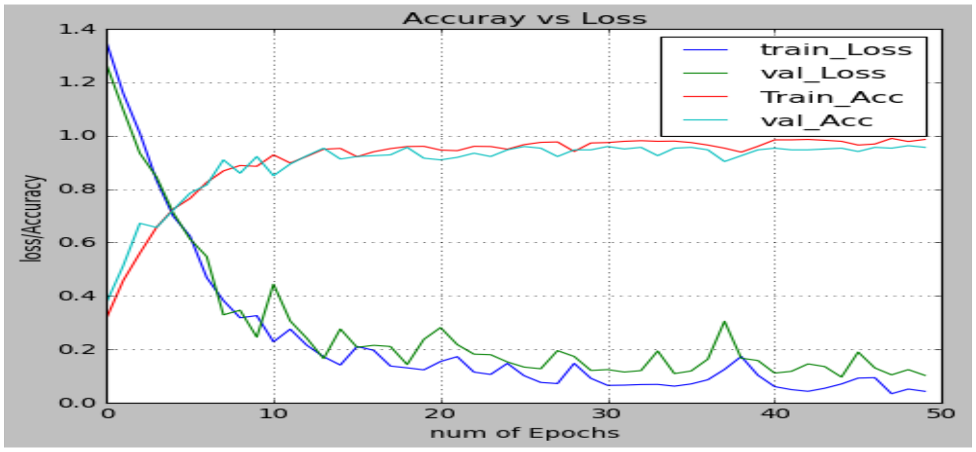


Figure 4.6. Accuracy vs loss and epoch of end-to-end CNN model with a median filter

4.5.Evaluation of The Proposed Model

We have evaluatedtheproposed model using the confusion matrix, accuracy, recall, precision, and f1 score. The correct and incorrect predictions were count values and broken down to each class. We have used the Confusion matrixto analyze the performance of how the classification model was confused when performing predictions.It was a clear and explicit way to present the prediction results of a classifierfor each class. It has beenrepresented by a matrix of the true class labels versus thepredicted class labels containing the number of truly predicted instances and misclassifiedinstances for each class.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} ------------------------ 4.1$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} ---------------------------- 4.2$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} --------------------------4.3$$

$$\text{F1} - \text{score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} -------------------- 4.4$$

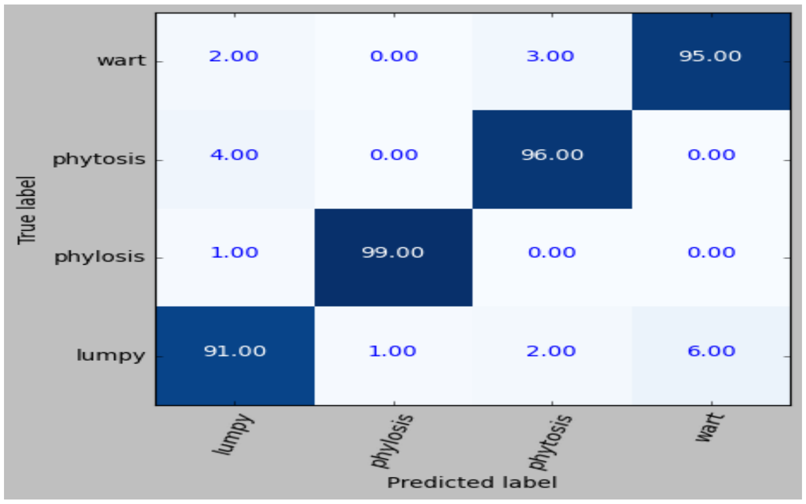
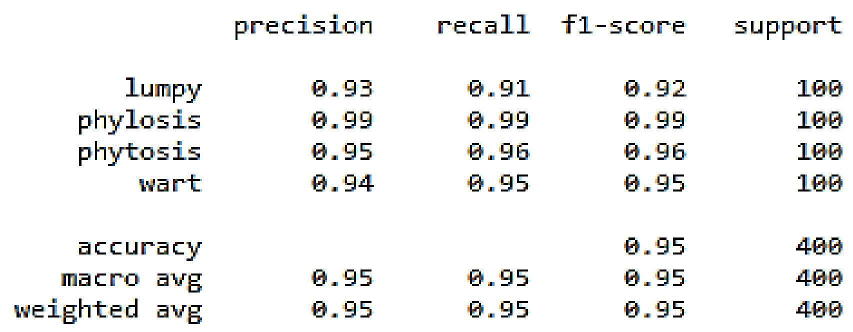


Figure 4.7. Confusion matrix end-to-end CNN model using Gaussian filter



```
              precision    recall  f1-score   support

       lumpy       0.93      0.91      0.92       100
    phylosis       0.99      0.99      0.99       100
    phytosis       0.95      0.96      0.96       100
        wart       0.94      0.95      0.95       100

    accuracy                           0.95       400
   macro avg       0.95      0.95      0.95       400
weighted avg       0.95      0.95      0.95       400
```

Figure 4.8. Precision, recall, and F1-score of CNN model using Gaussian filter

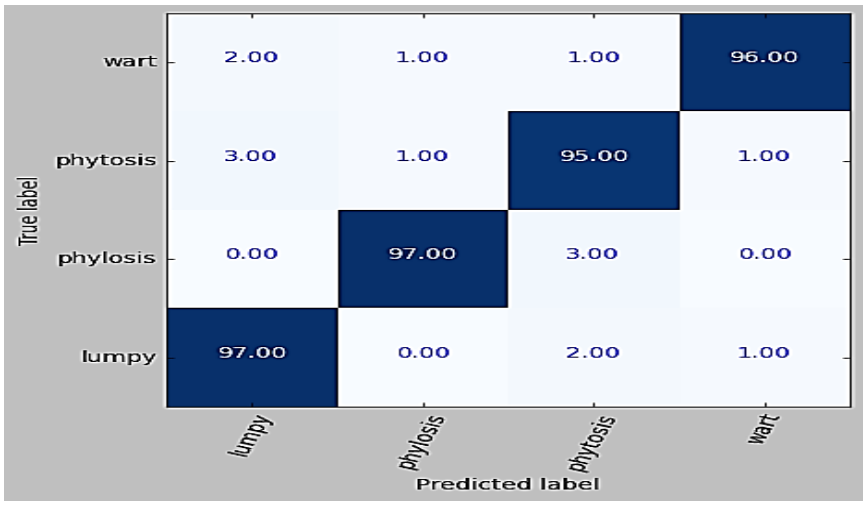
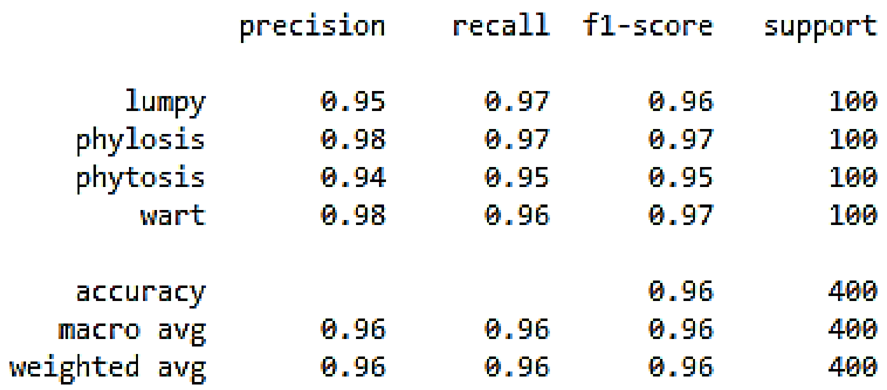


Figure 4.9. Confusion matrix of end-to-end CNN model using Gabor filter



```
              precision    recall  f1-score   support

       lumpy       0.95      0.97      0.96       100
    phylosis       0.98      0.97      0.97       100
    phytosis       0.94      0.95      0.95       100
        wart       0.98      0.96      0.97       100

    accuracy                           0.96       400
   macro avg       0.96      0.96      0.96       400
weighted avg       0.96      0.96      0.96       400
```

Figure 4.10. Precision, recall, and F1-score of CNN model using Gabor filter

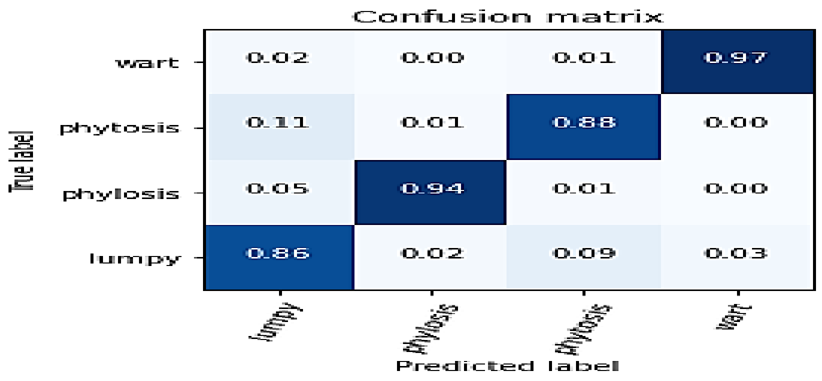


Figure 4.11. Confusion matrix of end-to-end CNN model usinga median filter

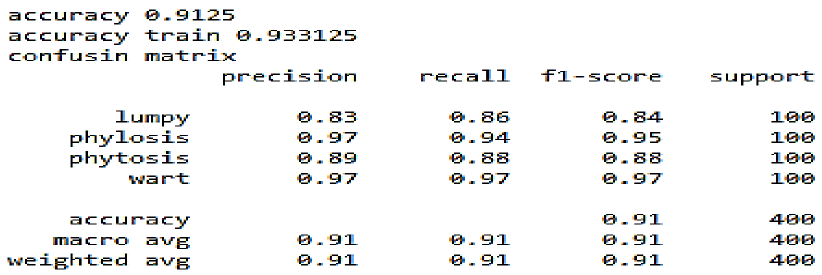

```
accuracy 0.9125
accuracy train 0.933125
confusin matrix
              precision    recall  f1-score   support

       lumpy       0.83      0.86      0.84       100
    phylosis       0.97      0.94      0.95       100
    phytosis       0.89      0.88      0.88       100
        wart       0.97      0.97      0.97       100

    accuracy                           0.91       400
   macro avg       0.91      0.91      0.91       400
weighted avg       0.91      0.91      0.91       400
```

Figure 4.12. Precision, recall, and F1-score of CNN model using a median filter

4.6.Experiments on SVM Classifier

In this study, feature vectors of CNN, HOG, and the hybrid of the two features (CNN and HOG) were tested using the SVM classifier and nonlinear kernel functions. First, we have compared thepolynomial, radial basis function (RBF),and linear kernel functions. Due to this experiment RBF has performed good accuracy as shown in table 4.4 below.

Table 4.4 Comparison of SVM kernel functions.

Kernel function	Feature vector		
	CNN feature vector	HOG feature vector	Combined feature vector
Poly	93.5%	80%	97%
RBF	96.25%	89%	98%
Linear	92.25%	79.25%	95.75%

It was observed from the result that SVM with RBF showed a better result in all feature vectors. Therefore, we have used the RBF kernel throughout the experiment.

Table 4.5: The comparison result of the three local feature descriptors

Feature descriptor	HOG	SURF	LBP
Accuracy	94.255%	90.25%	82.5%
Training time	180sec	120sec	300sec

We have compared the three state-of-the-art local feature descriptors (HOG, SURF, and LBP); the HOG feature has achieved better accuracy. As a result, we have used the HOG feature combined with CNN and got the final model.

4.6.1. HOG feature extraction using Gaussian filter and SVM classification

In this experiment, we used the HOG feature descriptor for feature extraction, Gaussian filter for filtering, and SVM for classification. As depicted inthe figure below, the HOG descriptor using a Gaussian filter was obtained with 98% training 84% testing classification accuracy.

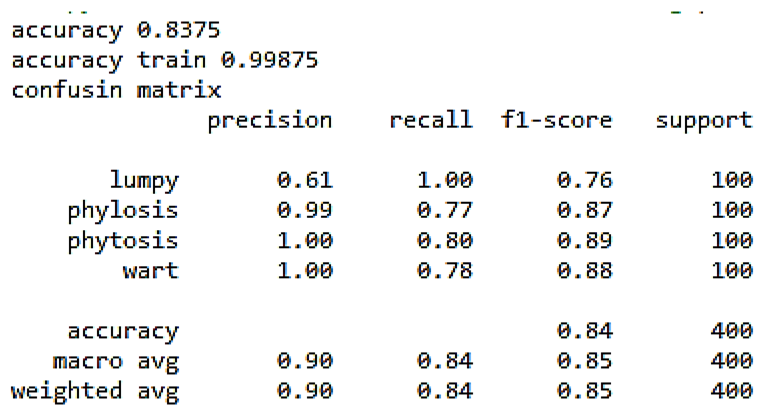
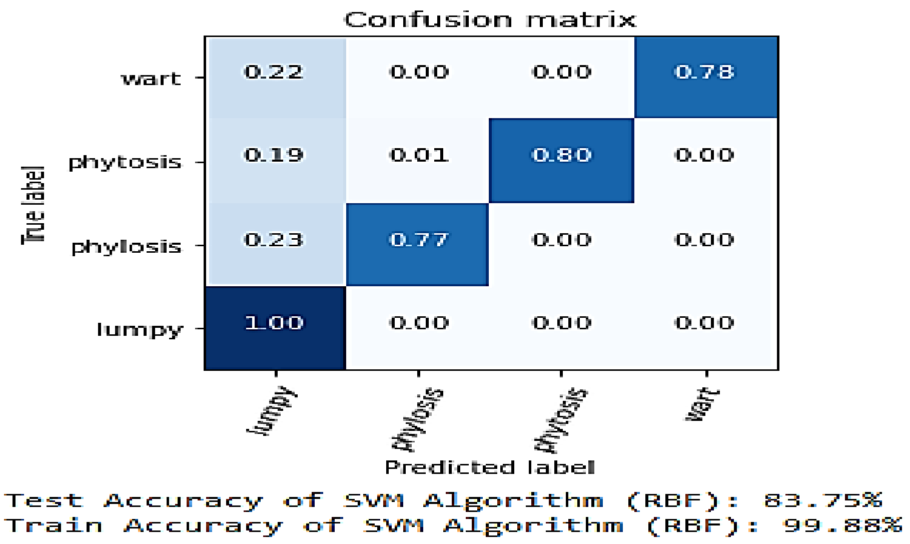

```
accuracy 0.8375
accuracy train 0.99875
confusin matrix
             precision    recall  f1-score   support

       lumpy       0.61      1.00      0.76       100
    phylosis       0.99      0.77      0.87       100
    phytosis       1.00      0.80      0.89       100
        wart       1.00      0.78      0.88       100

    accuracy                           0.84       400
   macro avg       0.90      0.84      0.85       400
weighted avg       0.90      0.84      0.85       400
```

Figure 4.13. Precision, recall,and F1-scoreof the above experiment

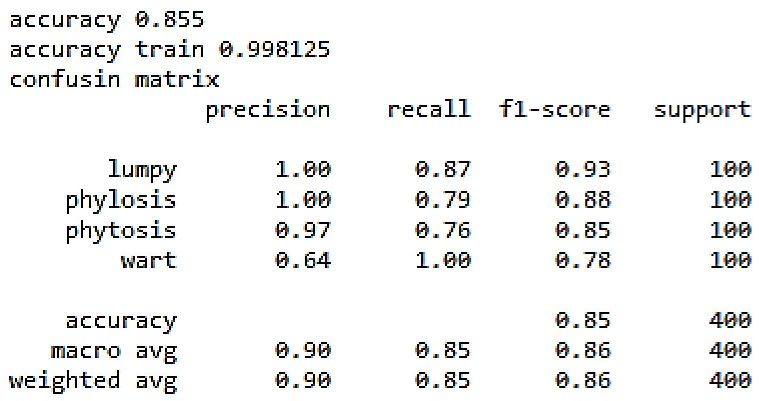


```
Test Accuracy of SVM Algorithm (RBF): 83.75%
Train Accuracy of SVM Algorithm (RBF): 99.88%
```

Figure 4.14. Confusion matrix of the above experiment

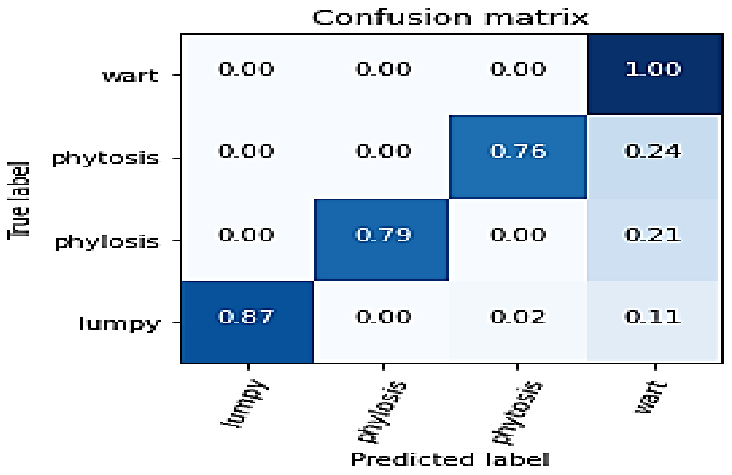
4.6.2. HOG feature extraction using Gabor filter and SVM classification

In this experiment, we have used the HOG feature descriptor for feature extraction, Gabor filter for filtering, and SVM for classification. As depicted in the figure below, the HOG descriptor using a Gabor filter was obtained with 98.5% training 85.5% testing classification accuracy.



```
accuracy 0.855
accuracy train 0.998125
confusin matrix
             precision    recall  f1-score   support

       lumpy       1.00      0.87      0.93       100
    phylosis       1.00      0.79      0.88       100
    phytosis       0.97      0.76      0.85       100
        wart       0.64      1.00      0.78       100

    accuracy                           0.85       400
   macro avg       0.90      0.85      0.86       400
weighted avg       0.90      0.85      0.86       400
```

Figure 4.15. Precision, recall, and F1-score of the above experiment



```
Test Accuracy of SVM Algorithm (RBF): 85.50%
Train Accuracy of SVM Algorithm (RBF): 99.81%
```

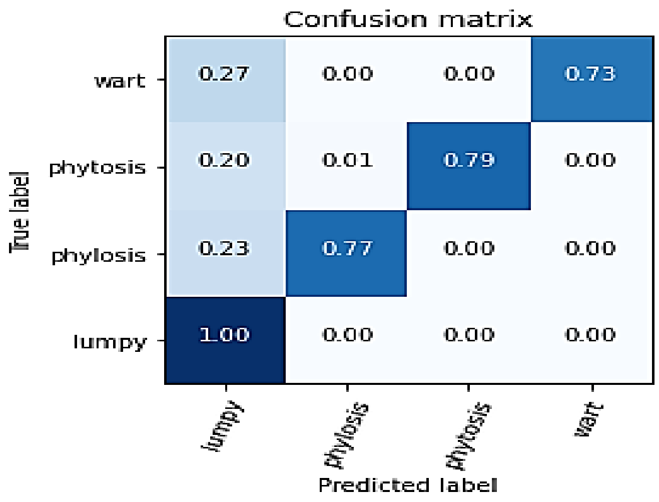
Figure 4.16. Confusion matrix of the above experiment

4.6.3. HOG feature extraction using median filter and SVM classification

In this experiment, we have used the HOG feature descriptor for feature extraction, median filter for filtering, and SVM for classification. As depicted below in the figure, the HOG descriptor usingthe median filter was obtained with 99% training 82.25% testing classification accuracy

```
accuracy 0.8225
accuracy train 0.999375
confusin matrix
              precision    recall  f1-score   support

       lumpy       0.59      1.00      0.74       100
    phylosis       0.99      0.77      0.87       100
    phytosis       1.00      0.79      0.88       100
        wart       1.00      0.73      0.84       100

    accuracy                           0.82       400
   macro avg       0.89      0.82      0.83       400
weighted avg       0.89      0.82      0.83       400
```

Figure 4.17. Precision, recall, and F1-score of the above experiment

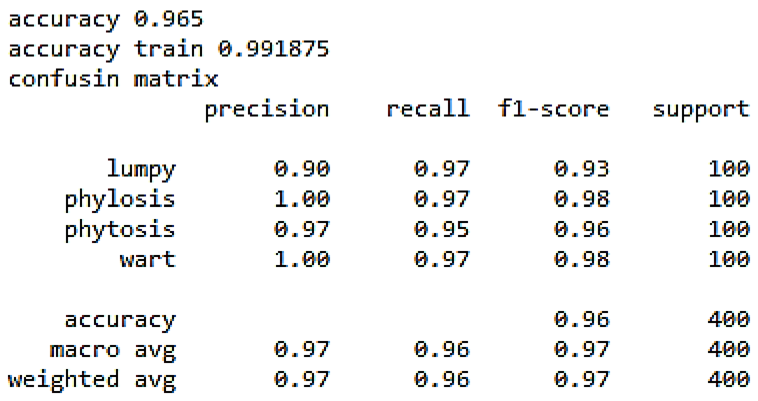


```
Test Accuracy of SVM Algorithm (RBF): 82.25%
Train Accuracy of SVM Algorithm (RBF): 99.94%
```

Figure 4.18. Confusion matrix of the above experiment

4.6.4. CNN feature extraction using Gaussian filter and SVM classification

Here, we have used CNN for feature extraction, Gaussian filter for filtering, and SVM for classification. As depicted below in the figure, the CNN using a Gaussian filter was obtained with 99.5% training 95.75% testing classification accuracy



```
accuracy 0.965
accuracy train 0.991875
confusin matrix
              precision    recall  f1-score   support

       lumpy       0.90      0.97      0.93       100
    phylosis       1.00      0.97      0.98       100
    phytosis       0.97      0.95      0.96       100
        wart       1.00      0.97      0.98       100

    accuracy                           0.96       400
   macro avg       0.97      0.96      0.97       400
weighted avg       0.97      0.96      0.97       400
```

Figure 4:19. Precision, recall, and F1-score of the above experiment

```
Test Accuracy of SVM Algorithm (RBF): 95.75%
Train Accuracy of SVM Algorithm (RBF): 99.50%
```

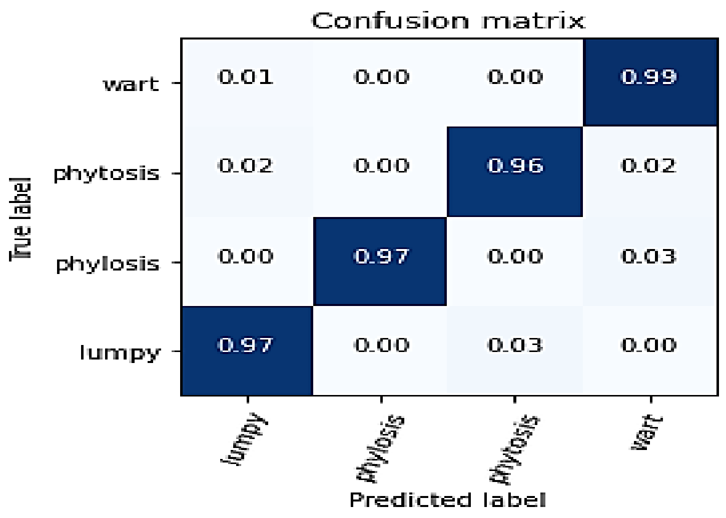
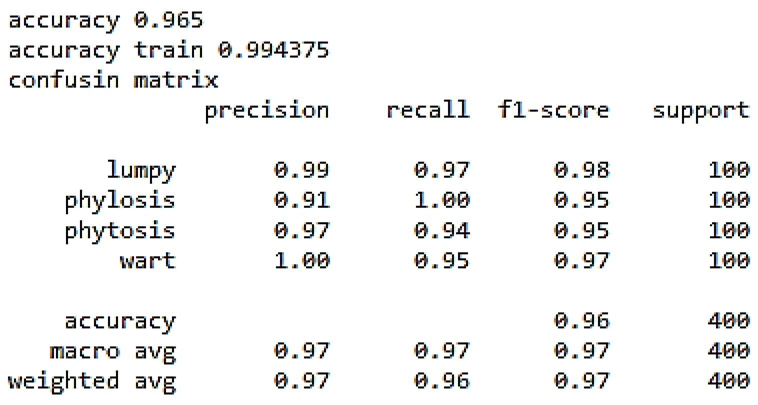


Figure 4.19. Confusion matrix of the above experiment

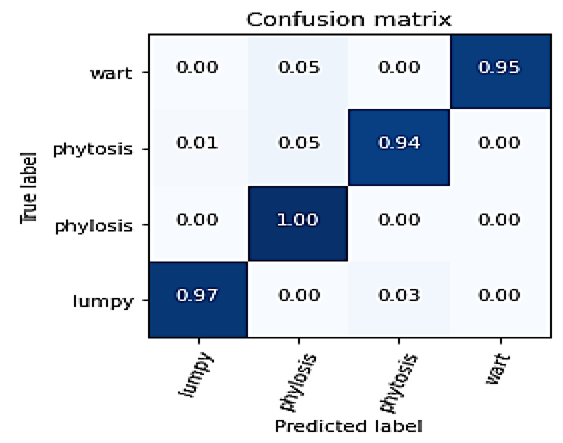
4.6.5. CNN feature extraction using Gabor filter and SVM classification

In this experiment, we have employed CNN for feature extraction, Gabor filter for filtering, and SVM for classification. As depicted below in the figure,the CNN feature extractor using a Gabor filter is obtained 99.8% training 996.5% testing classification accuracy



```
accuracy 0.965
accuracy train 0.994375
confusin matrix
                precision    recall  f1-score   support

       lumpy       0.99      0.97      0.98       100
    phylosis       0.91      1.00      0.95       100
    phytosis       0.97      0.94      0.95       100
        wart       1.00      0.95      0.97       100

    accuracy                           0.96       400
   macro avg       0.97      0.97      0.97       400
weighted avg       0.97      0.96      0.97       400
```

Figure 4.20. Precision, recall, and F1-score of the above experiment

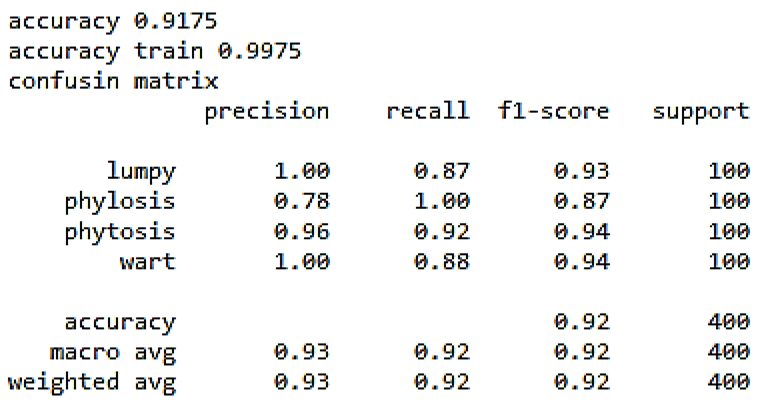


```
Test Accuracy of SVM Algorithm (RBF): 96.50%
Train Accuracy of SVM Algorithm (RBF): 99.44%
```

Figure 4.21. Confusion matrix of the above experiment

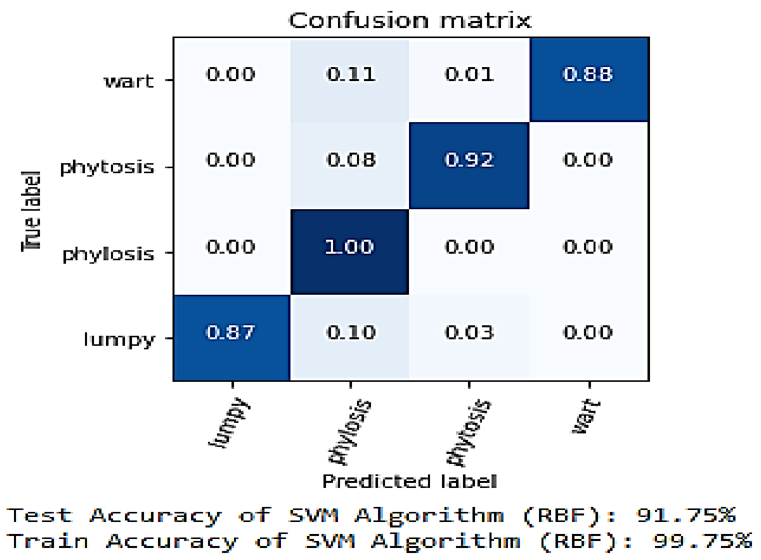
4.6.6. CNN feature extraction using median filter and SVM classification

In this experiment, we have worked with CNN for feature extraction, median filter for filtering, and SVM for classification. As depicted in the figure below, the CNN feature extractionwithina median filter was obtained 99.8% training 91.75% testing classification accuracy



```
accuracy 0.9175
accuracy train 0.9975
confusin matrix
              precision    recall  f1-score   support

       lumpy       1.00      0.87      0.93       100
    phylosis       0.78      1.00      0.87       100
    phytosis       0.96      0.92      0.94       100
        wart       1.00      0.88      0.94       100

    accuracy                           0.92       400
   macro avg       0.93      0.92      0.92       400
weighted avg       0.93      0.92      0.92       400
```

Figure 4.22. Precision, recall, support, and F1-score of the above experiment



```
Test Accuracy of SVM Algorithm (RBF): 91.75%
Train Accuracy of SVM Algorithm (RBF): 99.75%
```

Figure 4.23. Confusion matrix of the above experiment

4.6.7. Comparison summary of Gaussian, Gabor, and median filter

Most of the time training accuracy is greater than testing accuracy. In this experiment, the training loss was smaller than the testing loss throughout the curve. Therefore, using the Gabor filter,the image has a better performance compared to using a Gaussian filter and median filter in the model.As clearly shown in table 4.5 below, the Gabor filter has enhanced the CNN feature with an accuracy of 6% compared with the raw data.

Table 4.5. Comparison Summary of Gaussian, median, and Gabor filter

Filtering techniques	HOG feature		CNN + SoftMax		CNN+SVM
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing accuracy	Testing accuracy
On raw data	….	…..	91.25%	89.5%	90.5%
Gaussian filter	84.1%	83.25%	98.4%	95%	95.7%
Median filter	84.37%	79.75%	95.9%	91%	91.75%
Gabor filter	93.56%	89.7%	98.5%	96%	96.5%

4.6.8. Comparison Summary of Optimization Techniques

The optimization algorithms are important to minimize the loss function within global minima. In this experiment, we have compared the three popular optimization techniques (SGD, Adam, and RMSprop) shown in table 4.6 below.

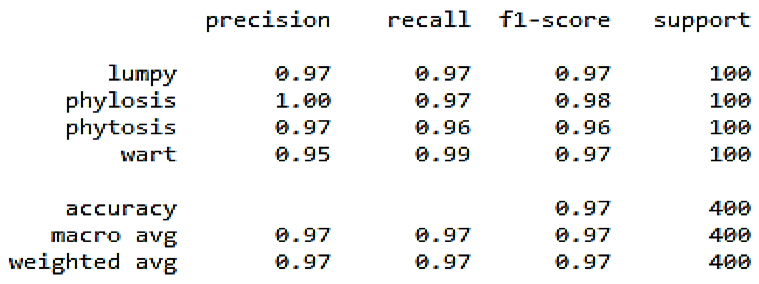
Table 4.6 Comparison of optimization techniques compiling CNN model

Optimization techniques	Accuracy	
	Testing accuracy	Training accuracy's
Adam	96.5%	98%
SGD	75%	89%
RMS prop	95%	96%

The Adam optimization technique has achieved a better result in the experiment compared with the current techniques.

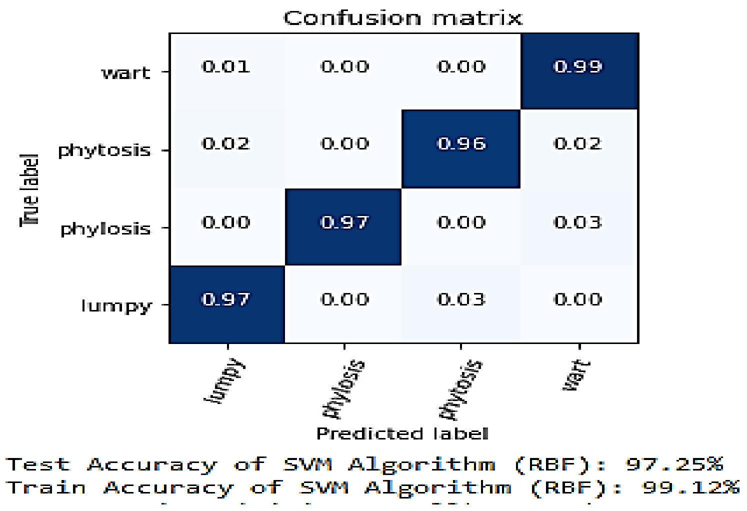
4.6.9. CNN Feature Extraction with PCA and SVM Classification

In this experiment, we have used CNN feature extraction with PCA dimensionality reduction, Gabor filter for filtering, and SVM for classification. As illustrated below in the figure, the CNN feature extraction with PCA using a Gabor filter was obtained with 99.8% training and 97.25% testing classification accuracy. Because PCA reduces the computational time of the classification processes.



```
              precision    recall  f1-score   support

       lumpy       0.97      0.97      0.97       100
   phylosis        1.00      0.97      0.98       100
   phytosis        0.97      0.96      0.96       100
        wart       0.95      0.99      0.97       100

    accuracy                           0.97       400
   macro avg       0.97      0.97      0.97       400
weighted avg       0.97      0.97      0.97       400
```

Figure 4.24. Precision, recall, support, and F1-score

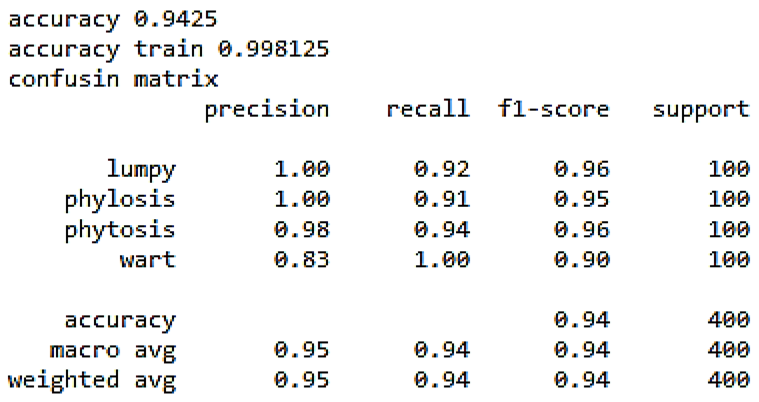


```
Test Accuracy of SVM Algorithm (RBF): 97.25%
Train Accuracy of SVM Algorithm (RBF): 99.12%
```

Figure 4.25. Confusion matrix of the above experiment

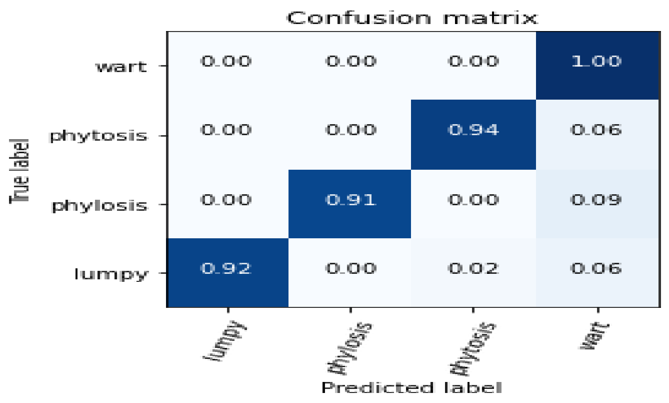
4.6.10. HOG Feature Extraction with PCA and SVM Classification

In this experiment, we have practicedwith the HOG feature descriptor, PCA dimensionality reduction, Gabor filter for noise filtering,and SVM for classification. As described below in the figure, HOG with PCA feature extractor using Gabor filter was obtained with 98.8% training 94.25% testing classification accuracy. Here, we have higher classification accuracy performance and good computational time than without using PCA.



```
accuracy 0.9425
accuracy train 0.998125
confusin matrix
              precision    recall  f1-score   support

       lumpy       1.00      0.92      0.96       100
    phylosis       1.00      0.91      0.95       100
    phytosis       0.98      0.94      0.96       100
        wart       0.83      1.00      0.90       100

    accuracy                           0.94       400
   macro avg       0.95      0.94      0.94       400
weighted avg       0.95      0.94      0.94       400
```

Figure 4.26. Precision, recall, and F1-score of the above experiment

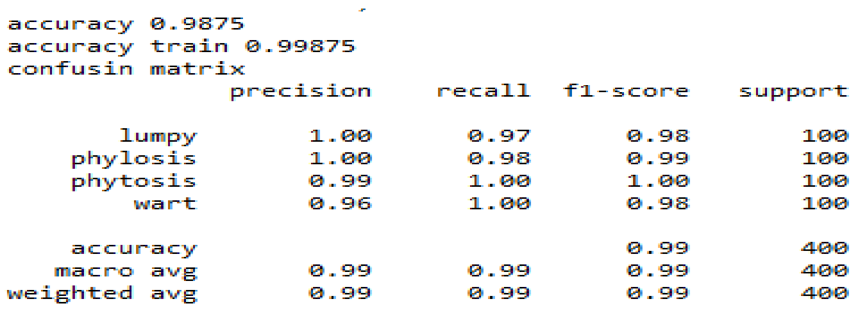


```
Test Accuracy of SVM Algorithm (RBF): 94.25%
Train Accuracy of SVM Algorithm (RBF): 99.81%
```

Figure 4.27. Confusion matrix of the above experiment

4.6.11. CNN and HOG for Feature Extraction and SVM Classification

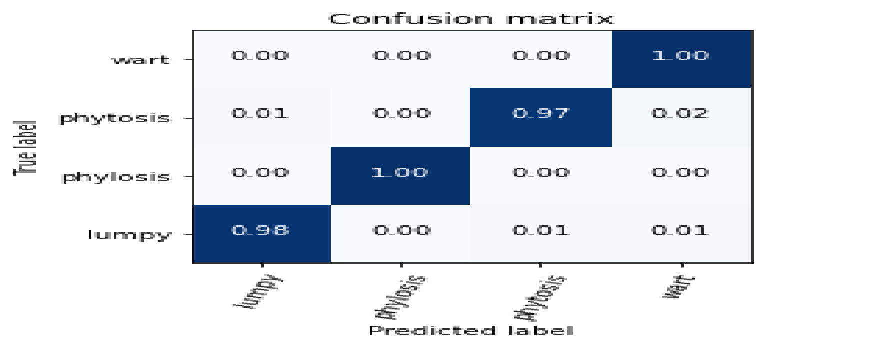
In this experiment, we have used the hybrid feature extraction (HOG feature descriptor and CNN feature extraction) and SVM for classification. We have concatenated the feature vector extracted using CNN and HOG, and then given it to the SVM classifier. We have achieved 99.8% training accuracy and 98.75% testing accuracy using the hybrid features.



```
accuracy 0.9875
accuracy train 0.99875
confusin matrix
              precision    recall  f1-score   support

       lumpy       1.00      0.97      0.98       100
    phylosis       1.00      0.98      0.99       100
    phytosis       0.99      1.00      1.00       100
        wart       0.96      1.00      0.98       100

    accuracy                           0.99       400
   macro avg       0.99      0.99      0.99       400
weighted avg       0.99      0.99      0.99       400
```

Figure 4.28. Precision, recall, and F1-score of the above experiment

In the hybrid feature vector, we have achieved 98.75% testingaccuracy and it performs better than the individual feature vector an experiment described in the above section. The following figure 4.29 below shows the confusion matrix prediction of this experiment.



```
Test Accuracy of SVM Algorithm (RBF): 98.75%
Train Accuracy of SVM Algorithm (RBF): 99.81%
```

Figure 4.29. Confusion matrix of the above experiment

In this experiment, we have achieved better prediction value in phylosis and wartcattle skin diseases. The prediction valuewas increasedcompared tothe previous experiment. In general, the classification accuracy of the model was better in thehybrid feature compared with the individual one.

4.7.Bar Graph of the Proposed Models

As we have described previously, an experiment conducted using the two classifiers (Softmax and SVM), CNN and HOG feature extraction, and dimensional reductionPCA. The performance of these models is described in figure 4.30 below.

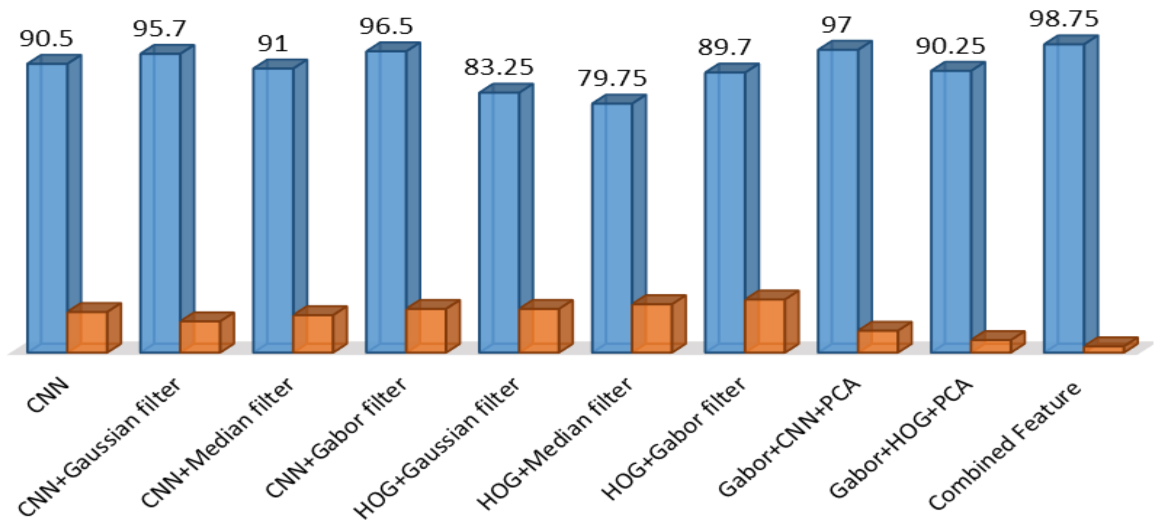


Figure 4.30. Performance of proposed models

4.8.Comparison of Proposed Model with AlexNet, LeNet, and VGGNet

We have comparedthe proposed model with the current state-of-the-art models AlexNet LeNet, and VGGNet based on the two criteria (accuracy and training time) using two classifiers (SoftMax and SVM). As described in the table below 4.7 VGGNet takes a high training time approximately in both classifiers while in SoftMax. Therefore, we obtained low performance and high training time in the AlexNet, LeNet, and VGGNet model compared withthe proposed model.

Table 4.7 Comparison summary of the model with AlexNet, LeNet, and VGGNet

Item	Proposed model	Alex Net		LeNet model		VGGNet	
Classifier	SVM	Softmax	SVM	softmax	SVM	softmax	SVM
Accuracy	98.75%	91.25%	93.5%	90.25%	92.75%	93%	93.75%
Training time	80sec	540sec	180sec	480sec	120sec	20hrs	300sec

4.8.1. SUMMARY

In this chapter, an experimental evaluation of the proposed model for the automatic identification of cattle skin diseases is described in detail. The dataset used and the implementation of the proposed model, the experimental comparison of the three filtering techniques (Gaussian, Gabor, and median) filters discussed in detail. The application of applying PCA dimension reduction in the two features (CNN and HOG feature) are discussed. The experimental result comparisons of the three local feature descriptors (SURF, HOG, and LBP) are discussed and the HOG descriptor achieves a good result. Finally, the result of CNN, HOG, and the hybrid featureis discussed in detail.

CHAPTER FIVE

5. CONCLUSIONS AND RECOMMENDATIONS

5.1.Conclusion

Becausecattle skin is the backbone of the Ethiopian economy, we proposed a cattle skin disease identification model for detecting lumpy, dermatophylosis, dermatophytosis, and wart. In this study, we follow experimental research, which involves data set preparation for training and testing the proposed model. We have prepared a total of 2000 images (500 per class). Images are initially pre-processed using Gaussian filter, Median filter, and Gabor filter for comparing their effect. We compared the three local feature descriptors and take the best one. We proposed new feature extraction techniques using HOG, CNN, and the combination of the two. After we extracted the features, the classification model is built using SVM and SoftMax classifier. Experimental results show that the combination of local features by HOG and deep features by CNN with the SVM classifiers can achieve better classification performance with an accuracy of 98.75%. We also compared the proposed cattle skin diseases identification model with the current state-of-the-art models AlexNet, LeNet, and VGGNet and the model has better performance accuracy than AlexNet, LeNet, and VGGNet. This research work identified the hybrid model (CNN with the local feature) asa major contributor and recent research area for the automatic identification of the four cattle skin diseases. CNNis not a local invariant so that it can be used to detect an image as having a particular sign of disease regardless of its spatial location. We have also identified that different approaches such as image processing, machine learning, and deep learning techniques are highly applied for the detection of cattle skin diseases.

5.2.Contribution

This research work has both methodological and scientific contributions.

The Scientific contributions:

- ♣ To the best of our knowledge, this study is the second investigation on cattleskin diseases including lumpy, dermatophylosis, dermatophytosis, and wartidentification.
- ♣ The enhanced CNN feature extraction is suitable for cattle skin diseaseidentification.
- ♣ It has an important contribution for identifying the limitation of CNN feature extraction towards image processing.

The methodological contribution

♣ It has a varied role for Comparison of filtering techniques: CNNs were trained with an input of raw (image) data. We have changed this trend by the application of compare filtering techniques. This enables the proposed model to have higher training and testing accuracy than CNN applied on the raw image data.

♣ It has a better starring role for Feature extraction: in this study, we proposed a newhybrid system of the two feature extraction techniques, as far as overall skin disease identification techniques.

♣ It is very imperative to work for comparison of a local feature: in this study, we have compared the three feature descriptors (HOG, SURF, and LBP). Selecting the best feature descriptor is important based on how the model gets better accuracy.

5.3.Recommendation

This research has constructive importance for areas of skin disease recognition, and classification. Therefore, those areas like leather industries and animal care centers are beneficial from the application of this model. That’s why we proposed this model for cattle skin disease classification so that many leather industries in Ethiopia shall apply the model.

5.4.Future Work

There are many research areas on different animals to protect from skin loss. In this study, we have used the four cattle skin diseases only used in lumpy, dermatophylosis, dermatophytosis, and wart. Therefore, future work will be focusedon all animal skin diseases by improving the model inthe best way. Furthermore, we will find appropriate filtering techniques for removing animal hair byusing deep learning-based segmentation techniques to fragment the most important part of the diseased part headed for increasing feature discrimination ability of algorithms on behalf of animal skin diseases.

REFERENCES

Abdul-rahman, S., & Norhan, A. K. (2012). *Dermatology Diagnosis with Feature Selection Methods and Artificial Neural Network*. *December*, 371–376.

Ajay Kumar Boyat, B. K. J. (2019). *TECHNIQUES-A REVIEW*. *6*(9), 56–61.

Albawi, S., Abbas, Y. A., & Almadanie, Y. (2019). *Robust skin diseases detection and classification using deep neural networks*. *July*. https://doi.org/10.14419/ijet.v7i4.24178

Amami, R. (2016). *Incorporating Belief Function in SVM for Phoneme Recognition Incorporating Belief Function for*. *September*. https://doi.org/10.1007/978-3-319-07617-1

Amarathunga, A. A. L. C., Ellawala, E. P. W. C., Abeysekara, G. N., & Amalraj, C. R. J. (2015). Expert System For Diagnosis Of Skin Diseases. *International Journal of Scientific & Technology Research*, *4*(1), 174–178.

Ambad, P. S., & Shirsat, A. S. (2016). An Image Analysis System to Detect Skin Diseases. *IOSR Journal of VLSI and Signal Processing*, *06*(05), 17–25. https://doi.org/10.9790/4200-0605011725

Ambilo, A., & Kebede, A. (2019). *Prevalence of major skin diseases of cattle in and*. 116–122. https://doi.org/10.34297/AJBSR.2019.02.000586

Ansari, U. B., & Sarode, T. (2017). *Skin Cancer Detection Using Image Processing*. 2875–2881.

Aruta, C. L., Calaguas, C. R., Gameng, J. K., Prudentino, M. V., Anthony, A., & Lubaton, C. J. (2015). *Mobile-based Medical Assistance for Diagnosing Different Types of Skin Diseases Using Case-based Reasoning with Image Processing*. *3*, 115–118.

Broti, T., Siddika, A., Rituparna, S., Hossain, N., & Sakib, N. (2020). *Medical Image Analysis System for Segmenting Skin Diseases using Digital Image Processing Technology*. *12*(28), 7–15.

Chollet, & Francois. (2018). Introduction to Keras. *Deep Learning with Python*, 97–111. http://link.springer.com/10.1007/978-1-4842-2766-4_7

Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient kNN classification algorithm for big data. *Neurocomputing*, *195*(February), 143–148. https://doi.org/10.1016/j.neucom.2015.08.112

Dey, A., & Singh, J. (2016). *Analysis of Supervised Machine Learning Algorithms for Heart Disease Analysis of Supervised Machine Learning Algorithms for Heart Disease Prediction with Reduced Number of Attributes using Principal Component Analysis*. *April*. https://doi.org/10.5120/ijca2016909231

Dey, L. (2016). *Sentiment Analysis of Review Datasets using Naïve Bayes ' and K -NN Classifier*.

Di Ruberto, C., & Putzu, L. (2016). A Feature Learning Framework for Histology Images Classification. *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology: Systems and Applications*, 37–48. https://doi.org/10.1016/B978-0-12-804203-8.00003-1

Fadnavis, S. (2014). Image Interpolation Techniques in Digital Image Processing: An Overview. *Journal of Engineering Research and Applications*, *4*(10), 70–73. www.ijera.com

Fekri-ershad, S., Saberi, M., & Tajeripour, F. (2012). *AN INNOVATIVE SKIN DETECTION A APPROACH USING COLOR B ASED IMAGE R ETRIEVAL*. *4*(3), 57–65.

Feyisa, A. F. (2018). *Journal of Veterinary Science & A Case Report on Clinical Management of Lumpy Skin Disease in Bull*. *9*(3), 9–10. https://doi.org/10.4172/2157-7579.1000538

G, S. R., Murthy, P. H. S., & Jena, S. (2018). *Digital Dermatology - Skin Disease Detection*. 8201–8211. https://doi.org/10.15680/IJIRSET.2018.0707007

Ganatra, N., & Patel, A. (2018). A Survey on Diseases Detection and Classification of Agriculture Products using Image Processing and Machine Learning. *International Journal of Computer Applications*, *180*(13), 7–12. https://doi.org/10.5120/ijca2018916249

Gonzalez, Iwasokun, & Gabriel. (2018). Image Enhancement Methods: A Review. *British Journal of Mathematics & Computer Science*, *4*(16), 2251–2277. https://doi.org/10.9734/bjmcs/2014/10332

Hambal, A. M., Pei, Z., & Libent Ishabailu, F. (2015). Image Noise Reduction and

Filtering Techniques. *International Journal of Science and Research*, *6*(3), 2319–7064. https://doi.org/10.21275/25031706

Handge, M. P. T., Khalkar, M. A. S., Randhe, M. S. K. S., & Patil, M. P. G. (2019). *Skin Disease Diagnosis System Using Image Processing*. *2*, 81–83.

Hossain, M. A., & Alam Sajib, M. S. (2019). Classification of Image using Convolutional Neural Network (CNN). *Global Journal of Computer Science and Technology*, *19*(2), 13–18. https://doi.org/10.34257/gjcstdvol19is2pg13

Hu, L. Y., Huang, M. W., Ke, S. W., & Tsai, C. F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, *5*(1). https://doi.org/10.1186/s40064-016-2941-7

Jaiswal, A. K., Bindushree, G. L., & A, P. L. (2020). *Detection and Classification of Skin Diseases*. *May*, 3658–3663.

Kabbai, L., Abdellaoui, M., & Douik, A. (2019). Image classification by combining local and global features. *Visual Computer*, *35*(5), 679–693. https://doi.org/10.1007/s00371-018-1503-0

Kaur, D., & Kaur, Y. (2014). Various Image Segmentation Techniques: A Review. *International Journal of Computer Science and Mobile Computing*, *3*(5), 809–814.

Kaur, G., & Kaur, R. (2012). *IMAGE DE-NOISING USING WAVELET TRANSFORM*. *2*(2), 15–21.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2019). *A Survey of the Recent Architectures of Deep Convolutional Neural Networks 1 Introduction*. *w*, 1–70.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*(8), 5455–5516. https://doi.org/10.1007/s10462-020-09825-6

Kumar, M., & Kumar, R. (2016). An intelligent system to diagnosis skin disease. *ARPN Journal of Engineering and Applied Sciences*, *11*(19), 11368–11373.

Kumar, V. B. (2016). *Dermatological Disease Detection Using Image Processing and Machine Learning*. 88–93.

Lyubchenko, V., Matarneh, R., Kobylin, O., & Lyashenko, V. (2016). Digital Image Processing. *The Electrical Engineering Handbook, Vol. 6(7)*, 79-83. https://doi.org/10.1016/B978-012170960-0/50064-5

Mengistu, A. D., Alemayehu, D. M., & Mengistu, S. G. (2016). Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques. *International Journal of Database Theory and Application*, *9*(4), 79–88. https://doi.org/10.14257/ijdta.2016.9.4.07

Navjeet, K., Punetha, D., & Ehiagwina, F. (2016). Preserving the Edges of a Digital Image Using Various Filtering Algorithms and Tools. *International Journal of Signal Processing, Image Processing, and Pattern Recognition*, *9*(12), 11–18. https://doi.org/10.14257/ijsip.2016.9.12.02

O'Shea, K., & Nash, and R. (2015). *An Introduction to Convolutional Neural Networks An Introduction to Convolutional Neural Networks*. *December*.

Panwar, M., Acharyya, A., Shafik, R. A., & Biswas, D. (2017). K-nearest neighbor-based methodology for accurate diagnosis of diabetes mellitus. *Proceedings - 2016 6th International Symposium on Embedded Computing and System Design, ISED 2016*, *November 2018*, 132–136. https://doi.org/10.1109/ISED.2016.7977069

Panwar, P., Gopal, G., & Kumar, R. (2016). *SEGMENTATION TECHNIQUES IN IMAGE PROCESSING*. *7*(12), 304–308.

Parsania, M. P. S., & Virparia, D. P. V. (2016). A Comparative Analysis of Image Interpolation Algorithms. *Pearce*, *5*(1), 29–34. https://doi.org/10.17148/ijarcce.2016.5107

Phil Scott. (2017). common Cattle skin diseases. *Book*.

Pitaloka, D. A., Wulandari, A., Basaruddin, T., & Liliana, D. Y. (2017). Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition. *Procedia Computer Science*, *116*, 523–529. https://doi.org/10.1016/j.procs.2017.10.038

Ren, X., Guo, H., He, G., Xu, X., Di, C., & Li, S. (2016). *Component Analysis Initialization for Image Classification*. https://doi.org/10.1109/DSC.2016.18

Reshma, & Shan. (2018). The Melanoma Skin Cancer Detection and Feature Extraction through Image Processing Techniques. *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, *5*(4), 106–112.

Reshma, Shan, Shukla, Kuldeep, & Narayan. (2018). Image Enhancement Techniques. *International Journal of Engineering and Applied Computer Science*, *02*(07), 232–235. https://doi.org/10.24032/ijeacs/0207/05

Rishi, S., & Bajpai, K. (2017). Analysis of Image Enhancement Techniques Used in Remote Sensing Satellite Imagery. *International Journal of Computer Applications*, *169*(10), 1–11. https://doi.org/10.5120/ijca2017914884

Saini, O. and P. S. S. (2018). A Review on Dimension Reduction Techniques in Data Mining. *Computer Engineering and Intelligent Systems*, *9*(1), 7–14.

Saiyed, S., Bhatt, N., & Ganatra, A. P. (2016). *A Survey on Naive Bayes Based Prediction of Heart Disease Using Risk Factors Research in Engineering A Survey on Naive Bayes Based Prediction of Heart Disease Using Risk Factors*. *January*.

Shah, M., Patel, T., Joshi, M., & Parmar, P. N. (2019). *To Identify Animal Skin Disease Model Using Image Processing*. *5*(2), 1087–1092.

Tania, & Rowaida. (2016). Comparative analysis of filtering techniques. *Journal of King Saud University - Computer and Information Sciences*, *8*, 15. https://doi.org/10.1016/j.jksuci.2020.03.007

TÜRKOĞLU, HANBAY, Satpute, & R, M. (2019). *Color, Size, Volume, Shape and Texture Feature Extraction Techniques for Fruits : A Review*. *2016*.

Ubale, A. V, & Paikrao, P. L. (2019). *Detection and Classification of Skin Diseases using Different Color Phase Models*. *July* 1331–1335.

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, *19*(1), 1–16. https://doi.org/10.1186/s12911-019-1004-8

Vassallo-Barco, Hossin, & Sulaiman. (2020). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Wei, L. S., Gan, Q., & Ji, T. (2018). Skin Disease Recognition Method Based on Image Color and Texture Features. *Computational and Mathematical Methods in Medicine*, *2018*. https://doi.org/10.1155/2018/8145713

Yadav, N. (2016). *Skin Diseases Detection Models using Image Processing : A Survey*. *137*(12), 34–39.

Zubair, Rasak, A., & Fakolujo. (2014). Image Edge Detection and Image Edge Enhancement : Numerical Experiment on High Pass Spatial Filtering. *International*

Journal of Computer and Information Technology, *03*(04), 772–781.

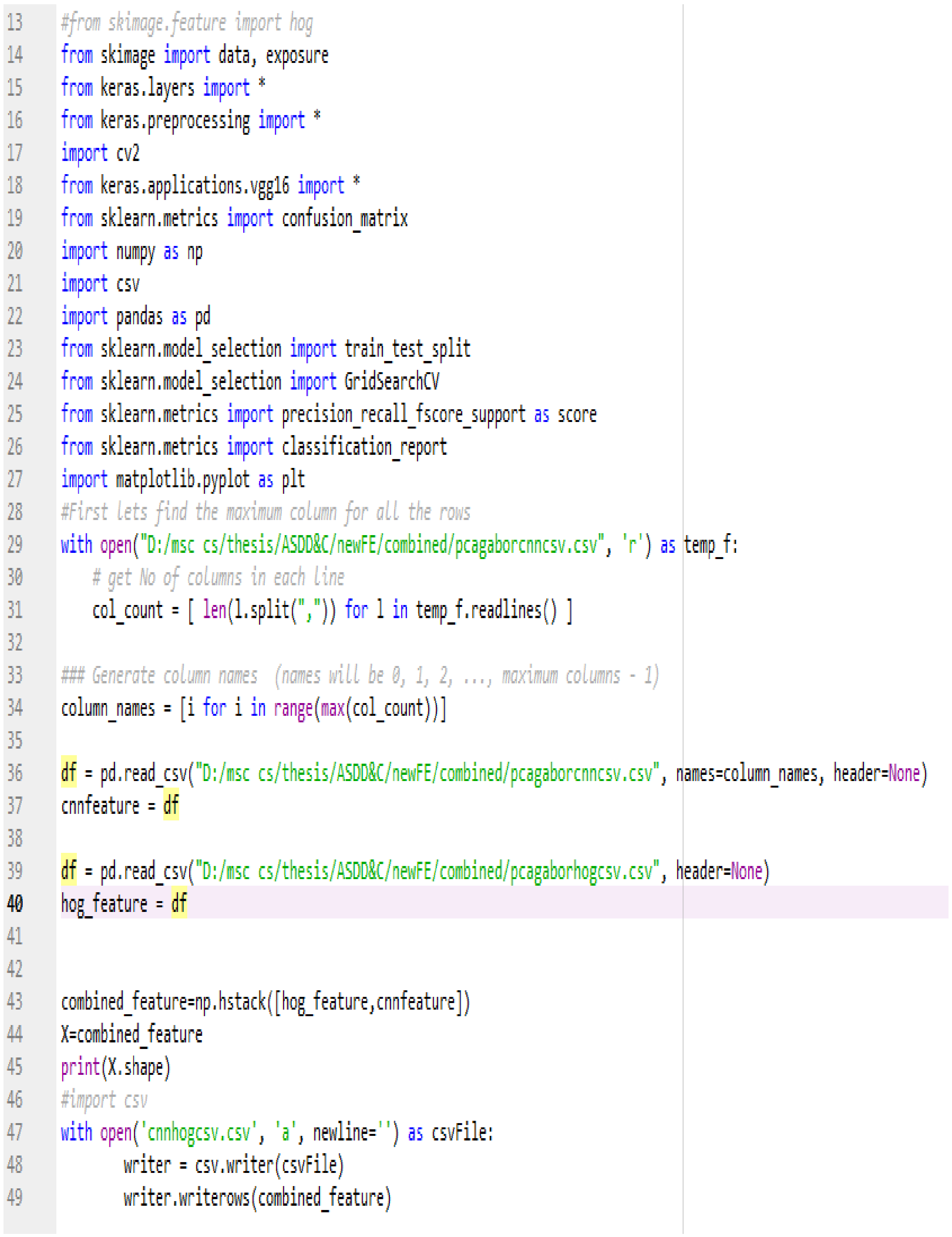
APPENDIX

A. Description of the Proposed CNN Model

Model: "sequential"

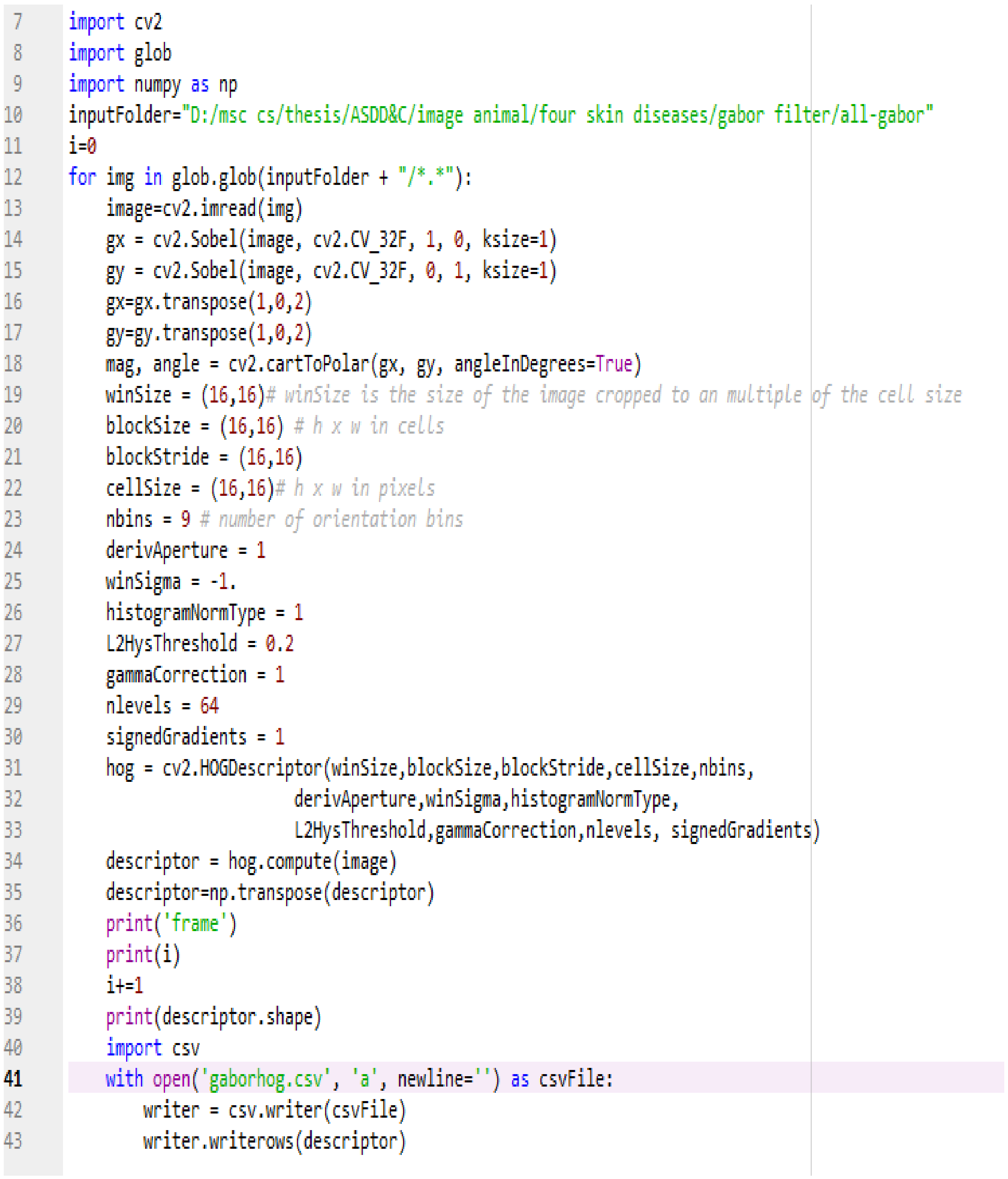
Operations	Kernel/stride	Filter size	Output Shape	Parameters
conv2d_1 (Conv2D)	3x3/1	32	224x224x32	64544
max_pooling2d_1 (MaxPooling2	-------	32	75x75x32	0
activation_1 (Activation)	------	32	75x75x32	0
dropout_1(Dropout)	------	32	75x75x32	0
conv2d_2 (Conv2D)	5x5/1	32	75x75x32	25632
max_pooling2d_2 (MaxPooling2	--------	32	25x25x32	0
average_pooling2d_1 (Average	--------	32	13x13x32	0
activation_2 (Activation)	--------	32	13x13x32	0
dropout_2 (Dropout)	---------	32	13x13x32	0
conv2d_3 (Conv2D)	7x7/1	64	13x13x64	100416
max_pooling2d_3 (MaxPooling2	-------	64	5x5x64	0
activation_3 (Activation)	-------	64	5x5x64	0
dropout_3 (Dropout)	---------	64	5x5x64	0
conv2d_4 (Conv2D)	3x3/1	64	5x5x64	36928
flatten_1 (Flatten)	--------	320	320	0
dense_1 (Dense)	--------		320	1284

B. HOG and CNN feature concatenation sample code



```
#from skimage.feature import hog
from skimage import data, exposure
from keras.layers import *
from keras.preprocessing import *
import cv2
from keras.applications.vgg16 import *
from sklearn.metrics import confusion_matrix
import numpy as np
import csv
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import precision_recall_fscore_support as score
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
#First lets find the maximum column for all the rows
with open("D:/msc cs/thesis/ASDD&C/newFE/combined/pcagaborcnncsv.csv", 'r') as temp_f:
    # get No of columns in each line
    col_count = [ len(l.split(",")) for l in temp_f.readlines() ]

### Generate column names  (names will be 0, 1, 2, ..., maximum columns - 1)
column_names = [i for i in range(max(col_count))]

df = pd.read_csv("D:/msc cs/thesis/ASDD&C/newFE/combined/pcagaborcnncsv.csv", names=column_names, header=None)
cnnfeature = df

df = pd.read_csv("D:/msc cs/thesis/ASDD&C/newFE/combined/pcagaborhogcsv.csv", header=None)
hog_feature = df


combined_feature=np.hstack([hog_feature,cnnfeature])
X=combined_feature
print(X.shape)
#import csv
with open('cnnhogcsv.csv', 'a', newline='') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerows(combined_feature)
```

C. HOG local feature descriptor sample code



```python
import cv2
import glob
import numpy as np
inputFolder="D:/msc cs/thesis/ASDD&C/image animal/four skin diseases/gabor filter/all-gabor"
i=0
for img in glob.glob(inputFolder + "/*.*"):
    image=cv2.imread(img)
    gx = cv2.Sobel(image, cv2.CV_32F, 1, 0, ksize=1)
    gy = cv2.Sobel(image, cv2.CV_32F, 0, 1, ksize=1)
    gx=gx.transpose(1,0,2)
    gy=gy.transpose(1,0,2)
    mag, angle = cv2.cartToPolar(gx, gy, angleInDegrees=True)
    winSize = (16,16)# winSize is the size of the image cropped to an multiple of the cell size
    blockSize = (16,16) # h x w in cells
    blockStride = (16,16)
    cellSize = (16,16)# h x w in pixels
    nbins = 9 # number of orientation bins
    derivAperture = 1
    winSigma = -1.
    histogramNormType = 1
    L2HysThreshold = 0.2
    gammaCorrection = 1
    nlevels = 64
    signedGradients = 1
    hog = cv2.HOGDescriptor(winSize,blockSize,blockStride,cellSize,nbins,
                    derivAperture,winSigma,histogramNormType,
                    L2HysThreshold,gammaCorrection,nlevels, signedGradients)
    descriptor = hog.compute(image)
    descriptor=np.transpose(descriptor)
    print('frame')
    print(i)
    i+=1
    print(descriptor.shape)
    import csv
    with open('gaborhog.csv', 'a', newline='') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerows(descriptor)
```