DEVELOPMENT OF CONVOLUTIONAL NEURAL NETWORK (CNN) AND ARTIFICIAL NEURAL NETWORK (ANN) MODEL FOR DETECTION OF LEAF SPOT OF GROUNDNUT, ANTHRACNOSE AND POWDERY MILDEW OF MANGO

BY

NALAWADE REVATI RAMESH

M. Sc. (Ag.)

DEPARTMENT OF PLANT PATHOLOGY

FACULTY OF AGRICULTURE
DR. BALASAHEB SAWANT KONKAN KRISHI VIDYAPEETH,
DAPOLI- 415 721, DIST. RATNAGIRI (M.S)

JUNE, 2023

DEVELOPMENT OF CONVOLUTIONAL NEURAL NETWORK (CNN) AND ARTIFICIAL NEURAL NETWORK (ANN) MODEL FOR DETECTION OF LEAF SPOT OF GROUNDNUT, ANTHRACNOSE AND POWDERY MILDEW OF MANGO

A thesis submitted to the

FACULTY OF AGRICULTURE

DR. BALASAHEB SAWANT KONKAN KRISHI VIDYAPEETH, DAPOLI
(AGRICULTURAL UNIVERSITY)

DIST. RATNAGIRI (MAHARASHTRA), INDIA

In partial fulfilment of the requirements for the degree of

Doctor of Philosophy

In

PLANT PATHOLOGY

By

NALWADE REVATI RAMESH

M. Sc. (Ag.)

DEPARTMENT OF PLANT PATHOLOGY
FACULTY OF AGRICULTURE
DR. BALASAHEB SAWANT KONKAN KRISHI VIDYAPEETH,
DAPOLI- 415 721, DIST. RATNAGIRI (M.S)

JUNE, 2023

DEVELOPMENT OF CONVOLUTIONAL NEURAL NETWORK (CNN) AND ARTIFICIAL NEURAL NETWORK (ANN) MODEL FOR DETECTION OF LEAF SPOT OF GROUNDNUT, ANTHRACNOSE AND POWDERY MILDEW OF MANGO

A thesis submitted to the

FACULTY OF AGRICULTURE
DR. BALASAHEB SAWANT KONKAN KRISHI VIDYAPEETH, DAPOLI
(AGRICULTURAL UNIVERSITY)
DIST. RATNAGIRI (MAHARASHTRA), INDIA

In partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In

PLANT PATHOLOGY

By

NALAWADE REVATI RAMESH
M. Sc. (Ag.)

Approved by the Advisory Committee:
Chairman and Research Guide:

(S. D. Sawant)
Vice Chancellor (Rtd.),
Dr. Balasaheb Sawant Konkan Krishi Vidyapeeth,
Dapoli

Members:

(M. S. Joshi)
Head,
Department of Plant Pathology,
College of Agriculture, Dapoli

(V. G. More)
Agrometerologist,
Department of Agronomy,
College of Agriculture, Dapoli

(P. M. Ingle)
Associate Professor,
Department of Irrigation and Drainage
Engineering CAET, Dapoli

(J. S. Dhekale)
Professor-CAS (Rtd.),
Department of Agril. Economics and Statistics,
Dapoli

Dr. S. D. Sawant
M. Sc. (Ag.) Ph. D.
Vice Chancellor (Rtd.),
Dr. Balasaheb Sawant Konkan Krishi Vidyapeeth,
Dapoli, 415 712, Dist. Ratnagiri (M.S.)

CERTIFICATE

This is to certify that the thesis entitled, **"Development of Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) Model for Detection of Leaf Spot of Groundnut, Anthracnose and Powdery Mildew of Mango"** submitted to the Faculty of Agriculture, Dr. Balasaheb Sawant Konkan Krishi Vidyapeeth, Dapoli, Dist. Ratnagiri, Maharashtra State, in the partial fulfilment of the requirements for the degree of ***DOCTOR OF PHILOSOPHY*** in **PLANT PATHOLOGY**, embodies the results of a piece of *bona-fide* research carried out by **Ms. NALAWADE REVATI RAMESH** under my guidance and supervision and that no part of this thesis has been submitted for any other degree or diploma or published in other form. All the assistance and help received during the course of investigation and the sources of literature have been duly acknowledged by her.

Place: Dapoli
Date: 28-12-2023

(S. D. Sawant)
Chairman,
Advisory Committee and
Research Guide

DECLARATION OF STUDENT

I hereby declare that the experimental work and interpretation of the thesis entitled **"DEVELOPMENT OF CONVOLUTIONAL NEURAL NETWORK (CNN) AND ARTIFICIAL NEURAL NETWORK (ANN) MODEL FOR DETECTION OF LEAF SPOT OF GROUNDNUT, ANTHRACNOSE AND POWDERY MILDEW OF MANGO"** or part of thereof has neither been submitted for any other degree or diploma of any University, nor the data have been derived any thesis/ publication of any University or Scientific Organization. Sources and material used and all assistance received during the course of investigation have been duly acknowledged.

RNalawade

Place : Dapoli **(Nalawade Revati Ramesh)**

Date : 28-12-2023 **Enrollment No.- ADPD/19/0318**

Acknowledgment

At the culmination of my Ph.D. thesis on "Development of Convolutional neural networks (CNN) and Artificial neural networks (ANN) for detection of Leaf spot of Groundnut, Anthracnose and Powdery mildew of Mango". I am filled with profound gratitude for the many individuals and organizations who have contributed to my journey in this fascinating field of study. Their support, guidance, and encouragement have been invaluable in shaping my research and personal growth.

I would like to extend my heartfelt appreciation and deepest gratitude to the following individuals who have played pivotal roles in my academic journey and the completion of this Ph.D. thesis.

*First and foremost, I am forever indebted to my parents [**Ramesh Nalawade** and **Shalan Nalawade**]. Their unconditional love, unwavering support, and constant encouragement have been the cornerstone of my success. Their sacrifices, belief in my abilities, and relentless encouragement have shaped me into the person I am today. I am eternally grateful for their endless guidance and the values they instilled in me.*

*To my brothers [**Karan** and **Kunal**] and my sister-in-law [**Komal**], thank you for your unwavering support and encouragement throughout this challenging journey. Your belief in me, words of wisdom, and constant motivation have been a source of strength during times of uncertainty. Your presence has been a reminder that I am never alone in this pursuit.*

*I would like to express my deepest gratitude and appreciation to my supervisor, [**Dr. S. D. Sawant**], for their invaluable guidance, expertise, and unwavering support throughout my Ph.D. journey. His exceptional knowledge, insightful feedback, and constructive criticism have been instrumental in shaping the direction and quality of this thesis. His mentorship and encouragement have been transformative, and I am privileged to have had the opportunity to work under their supervision. I am sincerely grateful for his unwavering support, trust, and belief in my abilities. This thesis would not have been possible without their guidance and support.*

*I would like to express my sincere appreciation and gratitude to **Dr. M. S. Joshi**, Head of Department of Plant Pathology, DBSKKV, Dapoli, for his exceptional leadership, guidance, and support throughout my Ph.D. journey. His insightful advice, constructive feedback, and encouragement have been invaluable in steering my research in the right direction and ensuring its relevance and impact. His mentorship and advocacy have not only enhanced the quality of this thesis but have also inspired me to strive for excellence in all aspects of my academic endeavours.*

*I would like to extend my heartfelt appreciation and gratitude to the members of my thesis committee, [**Dr. V. G. More, Dr. P. M. Ingle** and **Dr. J. S. Dhekale**], for their invaluable contributions and unwavering support throughout the development of this Ph.D. thesis.*

*I would like to acknowledge **Dr. Indu Sawant** for her invaluable assistance in the statistical analysis and result validation of my Ph.D. thesis. Her expertise in statistical methods and data analysis was instrumental in ensuring the accuracy and reliability of the research findings.*

*I extend my heartfelt thanks to the staff and faculty members [**Dr. P. G. Borkar, C. B. Gondhalekar** and **Dr. J. J. Kadam**] at Department at Plant Pathology, DBSKKV, Dapoli, for providing me with an exceptional academic environment. I give my sincerest thanks to **Swapnil Desai sir** for his technical support and expediting official procedures. It is my proud privilege to record my deepest sense of gratitude and cordial thanks to Laboratory Attendant **Shri. Dilip Bhuwad kaka**, and **Smt. Kshirsagar Kaki** for providing timely laboratory facilities.*

*I would like to express my deepest gratitude to **Mr. Sarang Nerkar** for providing a thermal camera for data collection. I would like to express my heartfelt gratitude to **Miss Sanam Kazi** for her invaluable assistance in the development of the VGG-16 model for detection of early leaf spot of groundnut. Her expertise and dedication significantly contributed to the successful implementation of this research project.*

*I would also like to express my heartfelt gratitude to my dearest friends **Sneha, Shivani, Bhagyashree, Sanika didi, Akshay, Hemant, Umesh** and **Pawan**, who have been pillars of support throughout this Ph.D. endeavour. Your unwavering friendship,* *understanding, and encouragement have provided solace during moments of stress and have made this journey truly memorable.*

I would like to express my sincere gratitude to ***Chhatrapati Shahu Maharaj Research, Training and Human Development Institute (SARTHI)****, Pune (An Autonomous Institute of Government of Maharashtra) for their generous financial support throughout my Ph.D. studies. Their funding has been instrumental in enabling the successful completion of this research project and the realization of my academic goals. Their investment in my education has provided me with invaluable resources, opportunities, and the freedom to focus on my research. I am deeply appreciative of their commitment to supporting the pursuit of knowledge and their belief in the potential of aspiring researchers.*

I am very much thankful to all authors and researchers whose articles helped me in organizing my research work on proper line and utilize proper tools for interpretation of the results.

To all those mentioned and those who have supported me behind the scenes, I offer my sincerest appreciation. Your unwavering belief in me, your understanding during times of sacrifice, and your presence in my life have made this achievement possible. I am truly humbled and honoured to have your love and support.

This thesis is dedicated to my parents and my brothers.

Place: Dapoli (Nalawade Revati Ramesh)

Date: Enrollment No. ADPD/19/0318

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PLATES

ABSTRACT

DEPARTMENT OF PLANT PATHOLOGY

COLLEGE OF AGRICULTURE, DAPOLI

Title of thesis	:	**"Development of Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) for detection of Leaf spot of Groundnut, Anthracnose and Powdery Mildew of Mango."**
Name of the student	:	Nalawade Revati Ramesh
Regd. No.	:	ADPD/19/0318
Name and designation of Guide	:	Dr. S. D. Sawant Vice Chancellor (Rtd.), Dr. Balasaheb Sawant Konkan Krishi Vidyapeeth, Dapoli- 415 712.
Year of award of degree	:	2023-24

THESIS ABSTRACT

This Ph.D. thesis presents a comprehensive study on the development and application of advanced machine learning techniques, specifically Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) models, for the detection and diagnosis of leaf spot of groundnut, anthracnose, and powdery mildew of mango.

The objective of this research was to explore the potential of deep learning techniques in plant pathology to improve the accuracy and efficiency of disease detection. Through the integration of image processing, pattern recognition, and machine learning, this thesis aims to address the challenges associated with the early and accurate diagnosis of these devastating diseases.

The research methodology involved the collection of extensive datasets comprising high-resolution RGB and thermal images of infected and healthy leaves of groundnut and mango and mango inflorescence. Pre-processing techniques, including image augmentation, thermal data extraction and data normalization, were employed to enhance the quality and diversity of the image dataset. Subsequently, deep learning models were trained and fine-tuned using state-of-the-art CNN and ANN architectures. In CNN model development, two methods were used. In first method, Teachable machine, a machine learning platform and in second method VGG-16 pre-trained model was used to

develop detection model based on RGB image dataset for leaf spot of groundnut, anthracnose and powdery mildew of mango.

The developed teachable machine models achieved a high accuracy of 98%, 96% and 97% for detection and classification of leaf spot of groundnut, anthracnose, and powdery mildew of mango respectively. The groundnut disease detection model exhibited 97% precision, 98% recall, and 97% F1 score, mango anthracnose detection model exhibited 97% precision, 96% recall, and 96% F1 score and mango powdery mildew model achieved 96% precision, 98% recall, and 97% F1 score, outperforming traditional approaches and showcasing the potential of teachable machine platform in plant disease detection. These models were incorporated into an android mobile application for usage.

These results highlight the potential of Teachable Machine as an effective tool for plant disease detection. The findings contribute to the field of plant pathology by showcasing the feasibility and practicality of using user-friendly machine learning platforms in agricultural applications.

The trained VGG-16 model achieved accuracy of 92.52% in detecting and classifying early leaf spot disease in groundnut surpassing conventional methods and highlighting the potential of deep learning for plant disease detection. The VGG-16 model demonstrated strong generalization capabilities by successfully identifying leaf spot disease in unseen groundnut leaf images. This indicates the model's ability to adapt to variations in leaf appearances and its potential for real-world deployment. The high accuracy, robust generalization, and scalability of the model hold significant implications for plant disease management, facilitating timely interventions and targeted treatments.

The ANN models developed for detection of early leaf spot of groundnut and anthracnose of mango using thermal images of disease infected leaves exhibited exceptional performance with Levenberg marquardt (LM) and Conjugate descent gradient (CM) learning algorithms. among the developed ANN models, the 2-2-1 architecture models with Levenberg Marquardt and Conjugate descent gradient algorithms performed very good as compared to other developed architectures.

The results of this study demonstrate the effectiveness of the developed models in accurately identifying and classifying early leaf spot of groundnut and anthracnose of mango. The trained CNN and ANN models exhibited high accuracy, precision, and recall rates, outperforming traditional image processing methods and manual diagnosis by experts.

Furthermore, the thesis investigates the transferability of the trained models by evaluating their performance on unseen datasets. The models displayed promising generalization capabilities, indicating their potential for real-world deployment and widespread application.

The findings presented in this Ph.D. thesis hold significant implications for plant disease management and agricultural practices. The developed CNN and ANN models offer a non-invasive, rapid, and cost-effective solution for early disease detection, enabling timely interventions and targeted treatments. This research contributes to the broader field of plant pathology and paves the way for further advancements in leveraging deep learning techniques for disease diagnosis and monitoring.

INTRODUCTION

CHAPTER I

INTRODUCTION

The most important oilseed crop in India is Groundnut (*Arachis hypogaea* L.), a self-pollinated annual legume crop, widely grown for its high-quality edible oil and culinary use in the tropical and warm humid regions of the world. India ranks second in groundnut and its oil production after China followed by USA and Nigeria (Tiwari *et al.,* 2018). It contains 48-50% oil and 26-28% protein, and a rich source of nutrients. About 50% of global production is used for edible oil extraction and industrial uses, 35% for food and 15% as seed and animal feed (Birthal *et al*., 2010). In Konkan region groundnut is grown on 20,000 ha area with a productivity of 1800 kg ha^{-1} (Waghmode *et al*., 2017).

The major biotic factors affecting groundnut yield and quality are foliar diseases, *viz*. early leaf spot (*Cercospora arachidicola* Hori.) and late leaf spots (*Phaeoisariopsis personata* Berk. And Curt.). These are the most widely distributed and economically important foliar disease of groundnut causing sever reduction in oil content. Each disease alone can cause substantial yield losses. The combined losses due to both these leaf spots are more than 50% depending on the time of occurrence and congenial weather. The disease damage the plant by reducing the leaf area available for photosynthesis and stimulating the leaflet abscission leading to heavy defoliation. The early leaf spot disease produces characteristic brown lesions surrounded by chlorotic halo on the upper surface of leaves eventually covering the entire leaf. Since groundnut is cultivated mainly during kharif season, the early leaf spot disease spreads rapidly depending on the intensity and distribution of rainfall (Thirumalaisamy *et al.,* 2021).

Mango (*Mangifera indica*) being the king of fruits is the most important fruit in tropical as well as subtropical regions of the world. India contributes a major share in area and production of mango. Mango is the third widely produced fruit crop of the tropics after banana and citrus. Konkan is the major mango producing belt on the west coast of Maharashtra where it occupies an area of 1,11,715 ha with 3,53,066 tons of fruits (Bhattacharyya *et al.,*2019). Konkan region accounts for about 10% of total area under mango in the country out of which almost 90% area is occupied by the choicest mango cultivar 'Alphonso' locally known as 'Hapus' (Raut *et al.,* 2017). Two major diseases hindering mango export and causing drastic yield losses in India are Anthracnose and Powdery mildew caused by *Colletotrichum gloeosporioides* (Penz.) Penz. & Sacc. and *Oidium mangifera* Berthet respectively.

Anthracnose of mango is worldwide in distribution infecting crops at various growth stages. Losses due to this disease was reported to be 60% or higher during the early rainy season. It is the most important field and post-harvest disease of mango resulting in major constraint in the export of mango. The disease causes direct reduction in quantity and quality of the harvested produce (Kumari *et al.*, 2017). Symptoms on leaves appear as irregular shaped black necrotic spots on both sides of the mango leaves. Lesions coalesce and form large necrotic areas, frequently along the leaf margins. Lesions primarily develop on young tissues and conidia are formed in lesions of all ages. Under favourable condition the pathogen can infect the twigs and cause dieback (Felipe Arauz, 2000).

Powdery mildew is a widespread disease of panicles, blossom clusters, fruits, and leaves. The disease can cause yield reduction up to 70% due to its effect on fruit set and development. This is an endemic disease under Konkan conditions; however, it may attain epiphytotic progression due to climatic fluctuations (Karande *et al.,* 2017). The fungus produces white superficial powdery growth on inflorescence with abundant conidia which are borne in chains on conidiophores, which later turns into grey colour. The sepals are more susceptible than the petals. Infected flowers fail to open and fall prematurely. The malformed inflorescence is also infected and the fungus is retained for longer duration in them. Malformed panicles give more favourable micro-climate for longer survival of the pathogen because of its more condensed nature (Misra, 2001).

Plant disease management in modern agriculture needs to be ecofriendly, sustainable, and yet should have high input use efficiency. In commercial and intensive farming, farmer is well informed on general recommended practices specific to crop. In most cases region specific climate based detail crop advisories have been developed in the form of extension literature and is made available to farmers. But to improve and optimize input use efficiency real-time, location and crop stage specific, weather information based advisory mainly on plant protection will be the most ideal input for intensive agriculture. Timely diagnosis of the problem and then appropriate action will be taken based on advisory to avoid increase in proportion of the problem and solve it with minimum use of inputs such as pesticides, labors etc. Such advisory will require large number of crop experts, real-time weather information etc. But with use of computers, and internet in agriculture, it is possible. With introduction of artificial intelligence (AI) need for large number of crop experts can also be minimized. To develop ideal plant protection advisory first and important step is to identify the currently present disease or insect problem correctly. Understanding the need in present study attempts have been made to develop AI based method to identify the crop diseases using sets of digital and field images of plant parts of specific crop plants i.e. ground nut leaves and mango leaves and inflorescence.

In order to improve the recognition rate of disease diagnosis, researchers have studied many techniques using machine learning and pattern recognition such as Convolutional Neural Network, Artificial Neural Network, Back Propagation Neural Network, Support Vector Machine and other image processing methods. During the past few years, the most widely used neural network algorithms for plant disease detection include Convolutional neural network (CNN) and Artificial neural network (ANN).

In this study, the aim was to develop a software system capable of identifying specific plant diseases using RGB and thermal photographs of disease-infected plant leaves or parts. Our approach involved leveraging image analysis techniques to extract relevant features from the images without direct observation of the actual plants. By employing advanced algorithms, a new approach to establish a reliable and automated method for disease detection solely based on visual information captured in the form of digital images was planned.

The aim was to propose a novel concept for plant disease detection. The hypothesis was that by employing RGB images of infected plant parts, along with measuring the temperature difference between healthy and disease-infected areas on the plant part using thermal image, an effective method for identifying specific plant diseases could be developed. Visual information from RGB images and thermal data was combined and exploited to find the potential of this integrated approach for disease identification. This innovative concept opens up new avenues for the field of plant pathology, providing a promising framework for accurate and efficient plant disease diagnosis and monitoring.

In contemporary times, the utilization of multispectral imaging has become prevalent for plant disease image analysis. However, in this research, instead of conventional approach a combination of RGB and thermal images was utilized for the development of Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) models for detection of particular plant diseases under study. By integrating these distinct image modalities, the aim was to explore the potential of these models in effectively discerning and classifying the plant diseases. This departure from the prevailing methodology presents an opportunity to investigate the performance of CNN and ANN models in disease identification using RGB and thermal imaging, shedding light on the applicability and efficacy of alternative techniques in the domain of plant pathology.

In this research, images of disease-infected plant parts were acquired, capturing both RGB and thermal data. Subsequently, image processing and sorting techniques were conducted to pre-process the acquired images. Leveraging these pre-processed images, a Convolutional Neural Network (CNN) model was developed utilizing RGB images and an Artificial Neural Network (ANN) model using thermal images. The objective was to employ these models for the detection of a specific plant disease. To assess the effectiveness and reliability of the models, model validation was performed, subjecting the results to rigorous scrutiny and verification. This process ensures the credibility and robustness of the research findings in the realm of plant disease identification.

In the Convolutional Neural Network (CNN) model developed for detection of groundnut early leaf spot, anthracnose and powdery mildew of mango diseases, RGB images of disease-infected plant parts obtained through a visible range smartphone camera were used as the input data. These images served as the foundation for the development of the model, which aimed to detect the specific plant disease under investigation. By utilizing RGB images captured using accessible smartphone technology, a practical and convenient approach to disease detection was ensured. This methodology demonstrates the potential for exploiting consumer-grade devices and visual data to detect plant diseases in an efficient and cost-effective manner.

Convolutional neural network architecture is specifically used for image recognition and classification. CNN models use visible image patterns and features to make the predictions. CNNs are made up of an input layer, several convolutional layers, along with pooling layers in between them, and finally full connection layer in addition to activation function layers, and output layers (Srikanth Tammina, 2019).

In CNN models image processing is done to process an image by extracting its features, dividing it into different parts based on some characteristics, enhancing its quality and classifying individual objects present in it. It has a big role in plant disease detection, which helps to classify the diseases automatically by providing image of the disease symptoms as input. Therefore, it eliminates the need for engaging experts, doing this monotonous task of monitoring a large field of crops (Moumita and Mantosh, 2019).

The most commonly used CNN architecture used in plant disease detection is VGG-16 (Simonyan and Zisserman, 2014). It is a 16-layer network comprised of convolutional and fully connected layers, using only 3×3 convolutional layers stacked on top of each other for simplicity (Kumar and Kumar, 2023).

In the development of the Artificial Neural Network (ANN) model for detection of groundnut early leaf spot and mango anthracnose diseases, the FlirOne thermal camera was used to acquire thermal images of disease-infected plant parts. Subsequently, utilizing the FlirOne image view application, the temperature values corresponding to both healthy and disease-infected areas of the plant parts were extracted. The resulting temperature difference values between these two regions were then categorized into distinct grades based on predefined difference ranges. These temperature difference values, along with their respective grades, were utilized as input parameters for training the ANN model specifically designed for the detection of the particular plant disease.

Importantly, this novel approach of utilizing temperature difference data from thermal images of infected plant parts has not yet been explored by previous researchers. Therefore, our study contributes a pioneering methodology to detect plant diseases using thermal images, potentially improving the accuracy and efficacy of the existing methods.

The ANN models are well suited under relationships between the input and output variables which are not linear. The ANN models need less data than conventional methods when properly validated and number of variables remain constant from one simulation to another (Ingle and Purohit, 2020). Artificial neural network is a calculation method that builds several processing units based on interconnected connections. The network consists of an arbitrary number of cells or nodes or units or neurons that connect the input set to the output. It is a part of a computer system that mimics how the human brain analyzes and processes data. ANN is an effective tool to model nonlinear systems which may be difficult to present by conventional mathematical equations.

With this view, the present investigation is planned to develop a disease detection model with the help of Convolutional neural network (CNN) and Artificial neural network (ANN) for leaf spot of groundnut, anthracnose and powdery mildew of mango. This will enable for early detection and management of the disease at primary infection stage thereby reducing the number of sprays in the recommended schedule.

The present research work is planned with following objectives:

1. Data collection for dataset creation of Leaf spot Groundnut, Anthracnose and Powdery mildew of Mango.

2. Development of Convolutional Neural Network (CNN) based model for detection of Leaf spot of Groundnut, Anthracnose and Powdery mildew of Mango.

3. Development of Artificial Neural Network (ANN) based model for detection of Leaf spot of Groundnut and Anthracnose of Mango.

REVIEW OF LITERATURE

CHAPTER II

Review of Literature

The necessary step for conducting any scientific research is a review of literature related to the research topic. By knowing the relevant research work carried out by other scientists in the past, we get additional information about our study which helps us to decide outline of the study, research objectives and methods to attain the proposed objectives. The knowledge of previous studies about the selected research topic unable the researcher to approach the problem in a more scientific way and it helps to plan the research in a proper direction. It assists the researcher to justify the research findings and backing reasons of variation. Some of the published literatures related to research topic are reviewed in this chapter.

The literature reviewed for the present study is grouped under following heads:

2.1 Data collection for dataset creation of Leaf spot of groundnut, Anthracnose and Powdery mildew of mango.

2.2 Development of Convolutional Neural Network (CNN) based model for detection of Leaf spot of groundnut, Anthracnose and Powdery mildew of mango.

2.3 Development of Artificial Neural Network (ANN) based model for detection of Leaf spot of groundnut and Anthracnose of mango.

2.1: Data collection for dataset creation of Leaf spot Groundnut, Anthracnose and Powdery mildew of Mango.

Qi *et al.* (2021) used 'Stack Ensemble' algorithm to detect groundnut leaf diseases. Groundnut leaf samples were collected from agronomic experiment base of South China Agricultural University, China. The images for algorithm development were then captured using a mobile phone. About 2000 images collected, after cropping the images into different types, finally 6029 pictures of groundnut leaves were obtained. The database contained 3205 images of healthy leaves (HL), 463 of leaves with scorch disease (SD), 1359 of leaves with rust disease (RD), 346 of leaves with leaf-spot disease (LSD) and 656 of leaves suffering from both rust and scorch disease (RD + SD).

Nikam and Sadavarte (2015) used image processing techniques for mango leaves disease severity measurement. mango leaves infected with different types of diseases such as powdery mildew, anthracnose, bacterial canker and bacterial black spot were selected for the study. About 150 sample images were captured using Sony CIBER shot 12 MP digital camera and preprocessed for noise removal. These different types of infected leaves were collected from different locations in Maharashtra State. They were stored in JEPG format in database folder. The diseased leaf was placed on white background, then leaf was zoomed or cropped so that only the leaf area was taken into image. The background of image was selected white to avoid reflectance.

Vangujare and Tuppad (2019) developed a software using Laplacian filter method to measure the severity of mango leaf diseases. Mango anthracnose and bacterial black spot leaf diseases were selected for the study which causes severe damage to leaves. Diseased leaf samples were collected from various locations in Maharashtra. Theses leaves then placed on white background then the leaf was zoomed, so as to ensure that the image contained only the mango leaves. About 200 images were captured using 12 mega pixel digital camera and stored in JEPG format. These images were then segmented to obtain total healthy leaf pixel area and total diseased leaf pixel area.

Veling *et al.* (2019) proposed an image processing-based system for detection of four mango disease *viz.*, anthracnose, powdery mildew, black banded and red rust and to provide solution over them. This system used Fuzzy C algorithm for model development. Healthy and diseases images of leaves, fruits and stem for analysis were collected using 8 mega pixel digital camera and were stored in JEPG format. About 50 images of each disease were collected for training the model and the images sent and collected from farmers were used for testing the model.

Singh *et al.* (2019) applied Multilayer Convolutional Neural Network for classification of mango leaves infected by anthracnose disease. In the proposed work, two dataset repositories were used, the first one was mango leaves dataset collected manually and second one was multiple plant leaves images downloaded from ‘PlantVillage’ dataset. Total 2200 images were used in this work; 1070 mango leave images were collected manually and 1130 images were downloaded from PlantVillage. These images were categories into four classes *viz.* mango leaves with disease, without disease, multiple plant leaves images with disease and without disease.

Arya and Singh (2019) compared CNN and AlexNet for detection of potato and mango leaf diseases. The potato images were taken from 'PlantVillage' online website. mango leaf images were self-acquired on field at G.B. Pant University of Agri. & Tech. the images were grouped into four classes, two classes represented the diseased leaves and two classes represented the healthy leaves of potato and mango plants. Only RGB (Red, Green, Blue) images were used for the work. The images were resized to 150×150 for CNN and 227×227 for AlexNet architecture to get better feature extraction. Data augmentation was performed using various transformation techniques like affine transformation, perspective transformation, image rotation and intensity transformation (color, brightness, and contrast). Finally, the database containing 4000 images was created.

Rajbongshi *et al.* (2021) evaluated the overall performance of six CNN models: DenseNet201, InceptionResNetV2, InceptionV3, ResNet50, ResNet152V2 and Xception for recognition of mango anthracnose, powdery mildew and rust diseases. Around 1000 images of whole mango leaf were taken over a fourteen-day period of time for creating the dataset. To solve the class imbalance problem and increase the data diversity data augmentation was performed, after which 500 images for each target class were obtained. After augmentation, the dataset was split into 80% for training and 20% for testing operations. All the models were trained at 100 epochs. In performance evaluation matrices, F1 score, accuracy, sensitivity, specificity, FNR (False negative rate) and FPR (False positive rate) were measured. Out of all the six CNN models tested, DenseNet201 performed better with 98.00% accuracy. In DenseNet201, the highest F1 score of 98.33% was for powdery mildew and red rust. The highest precision, and sensitivity of 98.33% was found for anthracnose, powdery mildew, and red rust. The highest specificity observed for anthracnose, gall machi, powdery mildew and red rust, was 99.58%. The lowest FNR and FPR, found for anthracnose, powdery mildew, red rust, was 1.67% and 0.42% as well.

Pujari *et al.* (2014) developed a recognition and classification system for produce affected with identically looking powdery mildew disease infecting grape, mango, chilli, wheat, beans and sunflower. The single powdery mildew infected leaf image was captured by digital camera. For image acquisition, a colour camera (DXC-3000A, Sony, Tokyo, Japan) was used. The camera was vertically oriented at 0.5-meter distance while capturing the images. Images were taken 7 days after disease incidence. Total 600 image samples were collected (100 images of each type). 300 images were used for model training and model testing was done on remaining 300 images.

Priya and D’souza (2015) presented a study of feature extraction techniques for the detection of disease of agricultural products. White rot, powdery mildew, scab and sooty blotch diseases of fruits were selected for study. The sample images were captured with the help of digital camera. Images were taken in controlled environment and stored in JEPG format. While taking images infected fruit of Apple was placed on a white background. Artificial light sources were placed at 45^{O} on each side of fruit to avoid reflection and get even light distribution everywhere. The fruits were zoomed at a level that the image contained only the fruit and white background. Around 500 images were collected in the same manner.

Yin and Nay (2018) proposed a methodology for analysis and detection of leaf diseases of chilli, grape, rice, soyabean, wheat, rose, cotton, apple and mango using digital image processing techniques. In this system database, the images collected with digital camera and downloaded from internet were used. Total 560 image samples were captured of four classes *viz.* bacterial blight, leaf spot, powdery mildew and rust found on etc. From these 115 were bacterial blight, 120 were leaf spot, 200 were powdery mildew and 125 were of rust infected leaves.

Taujuddin *et al.* (2020) developed a system capable to detect and identify plant diseases on leaves using Blobs detection. The input images were acquired from various resources such as Plantix.Net website. From this website 15 sample images of healthy leaves, 15 sample images of powdery mildew leaves and 15 images of leaf blight were collected for analysis.

Dandawate and Kokare (2015) developed a decision support system using support vector machine for identification and classification of soyabean plant diseases *viz*. leaf blight, yellow mosaic, grass hopper attack and sun-burn. The proposed system was efficient in identification between healthy and unhealthy soybean leaves. The soyabean image database was created by self-collection of soyabean leaf images using mobile camera having 5 mega pixels resolution. Total 120 images collected consisting of healthy, diseased and pest infected leaves images. These images were stored in JEPG format and the original image size was 1920×2560 pixels.

Akila and Deepan (2018) developed a detection and classification system for plant leaf diseases using deep learning algorithm. Dataset for this system contained images with several diseases in many different plants. In this system some of the cash/commercial crops, cereal crops, vegetable crops and fruit plants were considered. Major crops selected were sugarcane, cotton, potato, carrot, chilli, brinjal, rice, wheat, banana and guava. Diseased and healthy leaves were collected from different sources like some images were downloaded from internet and some collected manually using camera devices or any else from field. After collection the images were sorted into two classes healthy and diseased and a sub class of disease name was given for the later.

Dadrasjavan *et al.* (2019) used UAV- based multispectral imagery for fast citrus greening detection. Low altitude multispectral images were acquired in five discrete bands of R, G, B, Red Edge and NIR by an imaging camera embedded on an unmanned aerial vehicle. The images were collected at an altitude of 25m above ground level. Imaging was carried out in autumn, 2016, between 1:00 p.m. - 2:00 p.m. under natural light condition, sunny sky, no rolling clouds, and wind speed about 10 km/h. to align and geo-reference the captured multispectral images. Total 557 images were collected during this season. Four signalized (40 × 40 cm checkered black and white) targets were placed around the study area as ground control points. Also, 15 citrus greening infected and 10 healthy trees were selected by experts as reference data distributed around the whole orchard.

2.2: Development of Convolutional Neural Network (CNN) based model for detection of Leaf spot of Groundnut, Anthracnose and Powdery mildew of Mango.

Vaishnnave *et al.* (2020) introduced a deep convolutional neural network (DCNN) method for automatic detection of different groundnut diseases. The developed DCNN was able to classify the input images into ten classes as leaf spot, leaf blight, bud necrosis, early and late leaf spot, pepper spot, blight and rust disease. The dataset images contained 6400 groundnut leaf images which were collected from 'PlantVillage' online website, out of which 4500 images were used for training set and 1800 images for the testing process. There were seven layers in the proposed CNN model. Stochastic gradient decent (SGD) momentum method was used for model training. At initial stage the learning rate was fixed at 0.01, maximum epoch was set at 30 with 0.9 momentum value. During training of the CNN, the minibatch size was set at 128, drop period of 10 with factor rate of drop at 0.1. The proposed DCNN contained six combined layers (i.e., maximum pooling and convolutional layer) and optimal number of combined layers was found by using a grid search which was six layers giving a higher accuracy rate. The developed model accuracy was 99.85% for leaf spot, 99.94% for leaf blight, 99.97% for bud necrosis, 100% for both early and late leaf spot, 99.88% for pepper spot, 99.97% for blight and 100% for rust detection. The DCNN model gave comparison accuracy of 94.96%. The accuracy obtained by maximum pooling was 94.95% and average pooling was 94.83%. The overall accuracy performance of DCNN in each class is delivered 99.88% accuracy.

Muthukumaran *et al.* (2021) proposed a convolutional neural network algorithm for diagnosis of early and late leaf spot of groundnut. The collected infected leaf images were pre-processed using image compression techniques which were Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). The collected images were also enhanced using thresholding operations, homomorphic filters and contrast stretching techniques. The database was reconstructed with the output images of each image processing technique. The CNN was trained using each database and was built using ResNet-50 architecture with TensorFlow. Image pre-processing was done using 'Keras' library. To build convolutional layer 32 filters were used, the kernel size (the size of convolutional filter) was 3 and ReLU activation function was used to transform the output of the hidden layer to a linear state. Maximum pooling was used to do the flattening and feature map was adjusted at 64×64×3 with 128 dense units. Output layer of proposed CNN was employed using sigmoid function. The CNN trained using Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) databases classified the leaf spot infected leaves correctly with high accuracy.

Neha and Giridharan (2021) proposed a methodology for prediction of groundnut diseases using CNN models. The models considered for the classification were AlexNet, ANN, KNN and SVM. Early leaf spot, Late leaf spot and Rust disease of groundnut leaves were selected for the study. The leaf images were collected from marker assisted backcross of groundnut at five locations during 2015 rainy season, at Alityar Nagar, Tamil Nadu, India. The total groundnut dataset contained 105 images in total. The optimizers SGDM (Stochastic Descent Gradient), Adam and RMSprop were implemented for the classification. From the results of AlexNet model, it was found that the SGDM optimizer was efficient with the correct classification of diseased (81.56%) and non-diseased (15.26%). The optimizer Adam provided the classification of diseased (50%) and non-diseased (35%). Similarly, the optimizer RMSprop with AlexNet provided the classification of diseased (50%) and non-diseased (35.91%). It was also found that ANN model with the SGDM optimizer was efficient with the correct classification of diseased (79.82%) and non-diseased (18%). On similar lines, the KNN model with the SGDM optimizer was efficient with the correct classification of diseased (75.82%) and non-diseased (21.02%). The SVM model with SGDM optimizer was efficient with the correct classification of diseased (75.82%) and non-diseased (21.02%). This proved that AlexNet was better as compared to other models for prediction of groundnut diseases.

Patayon and Crisostomo (2022) tested pre-trained deep convolutional neural network models for identification of leaf spot of groundnut. They tested eight pre-trained models including Visual Geometry Group–16 (VGG16), Visual Geometry Group–19 (VGG19), InceptionV3, MobileNet, DenseNet, Xception, InceptionResNetV2, and ResNet50 with deep learning optimizers such as Stochastic gradient descent (SGD) with Momentum, Adaptive moment estimation (Adam), Root means square propagation (RMSProp), and Adaptive gradient algorithm (Adagrad). The image dataset contained 1000 images of infected groundnut leaves captured using a mobile camera. Confusion matrix was used to assess the accuracy and precision of the results. The result of the study showed that DenseNet-169 trained using SGD with momentum, Adam, and RMSProp attained the highest accuracy of 98%, while DenseNet-169 trained using RMSProp achieved the highest precision of 98% among pre-trained deep convolutional neural network architectures.

Rakholia *et al.* (2022) proposed a deep learning-based model with progressive resizing for groundnut leaf disease recognition and classification tasks. Five major categories of groundnut leaf diseases *viz*. leaf spot, armyworms effect, wilts, yellow leaf, and healthy leaf were considered for the study. The proposed model was trained with and without progressive resizing while it was validated using cross-entropy loss. The first of its kind dataset used for training and validation purposes was manually created from the Saurashtra region of Gujarat state of India. All major types of symptomatic leaves were plucked manually from the plants and put onto the fixed background to capture the images. Initially, all the images were captured in squared format with the size of 3000x3000 (3 color channels), and then later all the captured images were resized in different sizes of 32x32, 64x64, 128x128, and 256x256 for model development using progressive resizing. The created dataset was divided into a ratio of 80:20 for training and testing. The created dataset was imbalanced in terms of a different number of samples for each category. To handle the imbalanced dataset problem, the extended focal loss function was used. To evaluate the performance of the proposed model, different performance measures including precision, sensitivity, F1-score, and accuracy were applied. The proposed model achieved 96.12% accuracy. The model with progressive resizing performed better than the traditional core neural network-based model built on cross-entropy loss.

Sivasankaran *et al.* (2022) made a comprehensive investigation based on the collected groundnut leaf images in a total of (1950) both diseased and healthy from the various groundnut fields of Villupuram (T.N.), and trained a deep convolutional neural network based on VGG 16 Model with the utilization of Adam optimizer learning algorithm to identify five major groundnut leaf disease classes namely early leaf spot, late leaf spot, bud necrosis, rust and alternaria leaf spot along with the healthy leaf class. The trained model achieved an overall accuracy of 99.82 %. This trained model was compared for its performance and error measures in par with the CNN based VGG16 model with RMSprop optimizer, AlexNet and Inception_V3 models and proved to be outperforming all the compared baseline models in terms of performance metrics.

Arivazhagan and Ligi (2018) proposed a Deep convolutional neural network (DCNN) architecture with three hidden layers for identification of anthracnose, alternaria leaf spot, leaf gall, leaf webber and leaf burn of mango. The dataset contained 1200 images of both diseased and healthy leaves. The dataset was divided into six classes as C1 for Healthy leaf, C2 for anthracnose, C3 for leaf gall, C4 for leaf burn, C5 for alternaria leaf spot and C6 for leaf webber respectively. The proposed CNN architecture included an input layer for image input followed by three hidden layers with different number of filters and the last one was output layer. In the proposed architecture, each hidden layer was composed of a convolutional layer, a batch normalization layer, a rectified linear unit and a maximum pooling layer. Feature extraction was done using convolutional and pooling layers and classification was completed using the fully connected layer. The model images were trained using 100 images per class and another set of 100 images per class was used for testing which resulted in 96.67% testing accuracy.

Arya and Singh (2019) used CNN and AlexNet for comparative analysis for detection of potato and mango leaf diseases using RGB images. Potato leaf image dataset was taken from ‘PlantVillage’ (Open-source website) and mango leaf image dataset was collected manually on field at G.B. Pant University of Agri. & Tech. During pre-processing the images were resized into 150×150 for CNN and 227×227 for AlexNet. To increase the number of images data augmentation was performed, in which transformation techniques like affine transformation, perspective transformation, image rotation and intensity transformation were used. Finally, dataset of 4000 images was created. For training & validation 3523 images were used and for testing 481 images were used. Classification was done using two standard architectures: CNN and AlexNet. The training accuracy of CNN was 93.06% and AlexNet training accuracy was 99.75%. after final testing, AlexNet architecture achieved the most precise testing accuracy 98.33%, while CNN architecture achieved 90.85% accuracy.

Singh *et al.* (2019) developed a Multilayer Convolutional Neural Network (MCNN) for the classification of the mango leaves infected by the anthracnose disease. The developed model was then validated on a real-time dataset consisting 1070 both healthy and infected leaf images of mango leaves. After image acquisition all the images were pre-processed for contrast enhancement and rescaling. Histogram equalization method was used for contrast enhancement and central square crop method was used for rescaling. Four class labels were assigned for leaf images namely C_0 for non-diseased mango leaves, C_1 for diseased mango leaves, C_2 for other plant leaves without disease and C_3 for other plant leaves infected with different diseases. The entire dataset was divided into 80% training set and 20% testing set by randomly splitting the dataset. The proposed model consisted 6 Convolutional layers and 3 maximum pooling layers. It was implemented using an open soured software framework i.e., TensorFlow with Python as programming language. The learning rate was set to 0.01, dropout rate varied from 0.2 to 0.5, and momentum was 0.09 with weight decay of le-6. Training was accomplished in 3 days and testing was done in few minutes. The accuracy of the proposed model was 97.13% along with 2.87% missing report rate and 0% false report rate. The missing report rate was due to the vulnerabilities present in the real-time databases.

Trang *et al.* (2019) presented an image-based diseases identification method using a deep neural network with contrast enhancement and transfer learning for classification of mango leaves infected with anthracnose, powdery mildew and gall midge. The mango disease dataset was collected from ‘An Giang’ province having the largest yields of mango in Vietnam. The dataset consisted of a total of 394 images of mango in four different classes: anthracnose, gall midge, powdery mildew and healthy. The images were captured in the resolution of 3096×3096 pixels with none background. Rescaling of images was done during pre-processing to 256×256 pixels to compress the images. Centre alignment was done by distributing the region of the leaf evenly between the margins. For contrast enhancement firstly, the images in RGB color channel were converted into HSV color channel. Afterwards, the intensity was divided by separator into two sub-parameter which are high and low groups using golden section method. 80% of the dataset was treated as training set uses for training parameters in the convolutional neural network, while the remaining 20% of the dataset was used as a test set for performance evaluation. Besides, in order to compare some popular pre-trained models, the collected image dataset of diseased mango leaves was used to train and test Inception v3, AlexNet and MobileNet v2 models with the same ratio of the training set and test set. The average accuracy of the proposed method reached the highest percentage of 88.46% while the percentage for MobileNet v2 was the second highest at 84.62%. Values 78.48% and 76.92% were the proportion mean accuracy of Inception v3 and AlexNet respectively.

Veling *et al.* (2019) developed a system for recognition of mango diseases based on image processing. The system was developed for anthracnose, powdery mildew, black banded and red rust diseases of mango leaves. The image segmentation to differentiate disease affected area from healthy area was done using fast and robust fuzzy c-means clustering algorithm. Gray-level Co-Occurrence Matrix (GLCM) method was used to extract 9 features from the sample mango leave images which were contrast, correlation, energy, entropy, homogeneity, cluster prominence, cluster shade, variance and dissimilarity. These features then used to train the model using support vector machine classifier on 'MATLAB'. Fifty sample images of each disease type were used to train the model and the images sent by farmers were used to compare the trained database. The processing time of given system was 3 seconds for image segmentation and 0.1 seconds for classification after which results would be displayed. The developed system was able to detect and recognize diseases affected mango within 5 seconds of time proving its efficiency.

Gining *et al.* (2021) utilized Grey Level Co-occurrence Matrix (GLCM) feature extraction technique along with 'Matlab' image processing toolbox to develop anthracnose and bacterial black spot disease recognition system for 'Harumanis' cultivar of mango in Malaysia. The dataset contained 103 healthy mango leaf images and 43 diseased leaf images (anthracnose and bacterial black spot) obtained on field using a smartphone camera. The dataset was divided into 70% training and 30% testing set. 'MATLAB' filtering technique was used to adjust the noise and contrast of images during image pre-processing. The RGB images were then transformed to hue, saturation, and intensity (HIS) format for image segmentation. Finally, GLCM classifier was applied for feature extraction and disease classification. The whole process was implemented in 'MATLAB' image processing toolbox after GLCM feature extraction is done the results were viewed by clicking the view result button. The developed system detected and classified the diseases with an accuracy of 68.89%, which is low due to inadequacy of training images.

Kumar *et al.* (2021) developed a CNN model to detect and classify mango leaves infected with anthracnose disease. The dataset was divided into 80% training images and 20% testing images. To train the CNN, training images were fed to the input layer of the model then the model was evaluated in making prediction. The outcome and error were studied and if the outcome was wrong it was then reinitialized by backpropagation process where it was started again from bottom layer to top layer for making correct prediction. The proposed model was able to detect and classify healthy mango leaves and infected leaves. The proposed model was then implemented using 'TensorFlow' with Python programming language. The training accuracy was 90% and with increase in the number of epochs at 100th epoch it increased to 99.86%. Similarly, the validation accuracy was 16% at epoch1 and at 100th epoch it increased to 96.16%.

Wongshil *et al*. (2021) proposed a machine learning algorithm using convolutional neural network for detection of Mango fruit infected by anthracnose disease. Two databases of diseased and non-diseased mango fruits were created by capturing images using a Charged Couple Device (CCD) camera. The database images were divided into training and testing images of each type producing four class labels namely, S_0 for non-disease mango training, S_1 for disease mango training, S_2 for non-disease mango testing and S_3 for disease mango testing. Histogram equalization method was used for contrast enhancement and center square method was used for rescaling of images during image pre-processing. AlexNet pre-trained model architecture was used for training of the proposed CNN model. The CNN model was trained using TensorFlow open-source software with python programming language. The developed CNN model was able to detect the anthracnose infected mango fruits with more than 70% accuracy.

Rao *et al*. (2021) modified and applied AlexNet a pre-trained CNN architecture for feature extraction and classification of foliar diseases of mango and grape. The image dataset of mango leaves infected with leaf blight, bacterial canker, powdery mildew, scab and Grape leaves infected with black rot and esca (black measles) were acquired using a high-resolution digital camera in RGB format. Noise removal and image clipping methods were used for image pr-processing. The dataset was divided into 80% training and 20% testing dataset. The proposed AlexNet model was implemented on a GPU computer with revised layers of AlexNet. The AlexNet neural network was redesigned by adding a fully connected layer containing 8 neurons, a SoftMax layer and a classifier layer. The training options were set as, Learning rate at 0.0001, Epochs 8, Batch size 32, Iterations per batch 157, Stochastic Descent Gradient as training algorithm and default momentum. 30% of all image samples were used for validation. For mango leaves disease detection, the proposed model performed with the Precision, Recall, and F1 Score accuracy of 0.8950, 0.9050, and 90%. For Grape leaves disease detection, the proposed model performed with the Precision, Recall, and F1 Score accuracy of 0.9890, 0.9850, and 98.85% respectively. TensorFlow an open-source platform was used to implement the API for development of JIT CropFix Android App.

Five Convolutional neural networks namely LeNet, AlexNet, VGG16, VGG19 and ResNet were designed and tested by Deeba and Amutha (2020) for prediction and classification of vegetable leaf diseases. For this work, five main vegetable crops were considered: Potato (early and late bight), Tomato (bacterial spot, early blight, late blight, leaf spot, leaf mold, target spot and mosaic), Corn (rust, northern leaf blight and leaf spot), Brinjal (leaf spot and mosaic) and Chilli (powdery mildew, bacterial leaf spot and leaf spot). Dataset for tomato, potato and corn were extracted form PlantVillage website. The images for brinjal and chilli were collected real-time form agricultural fields of a village named Sathapadi, Salem District, Tamil Nadu, India. Data pre-processing was done to remove noise, and unwanted background, finally the images were resized before augmentation to standard format and uneven light was adjusted. To reduce overfitting of the results data augmentation was performed. The prepared dataset was used to train the CNN models. There were two types of testing done in this work, one testing was done using 'PlantVillage' dataset and second testing was done using the real-time collected dataset from the field. The system was trained at 50 epochs for each work. The results showed that 'ResNet' had the highest training accuracy of 96.6% with validation accuracy on PlantVillage dataset and Real-time images of 94.7% and 91.2% respectively.

AlexNet and Support vector machine convolutional neural networks were compared by Fulari *et al.* (2020) for plant leaf disease detection. The entire dataset was taken form Kaggle website i.e., "new plant diseases dataset". For CNN, 12,949 images were used for training containing healthy and diseased crop leaves such as apple, cherry, corn, grape, peach, pepper bell, potato, strawberry, tomato, etc. The images were collated and labeled with different categories of diseases and healthy. For SVM, 52 images were taken from open-source website. In that 37 images were used for training and 15 images for testing. SVM detected whether the leaf was healthy or infected. Image pre-processing and segmentation was done to remove the unwanted noise and to partition the plant images into diseased portion only. Color, shape and texture features were extracted using gray level co-occurrence matric (GLCM). After training, the proposed CNN (AlexNet) achieved an accuracy of 97.71% which was more than the SVM which had 80% accuracy.

Midhun and Therese (2021) explained Object detection technique based on 'Teachable machine' with a small project to identify different classes of plant diseases. 1000 on field images of each early blight, late blight and healthy leaf of potato were collected using smartphone camera. new image project was created on teachable machine and the image dataset was uploaded with class names as Early Blight, Late Blight and Healthy. Training parameters were set at 16 batch size, 50 epoch and 0.001 learning rate. Model training required about 4 to 5 minutes. The developed teachable machine model performed very well with detection accuracy of 98% for Early blight, 100% for Late blight and 87% for Healthy Potato leaf.

Kumar and Kumar (2023) used VGG-16 model for detection of tomato and potato diseases. The image dataset was taken from PlantVillage website. The developed model was able to detect tomato diseases with four classes *viz.* mosaic, leaf curl, target spot and healthy tomato leaf with 88.6% accuracy. The proposed model was able to detect early blight, late blight and healthy potato leaves with 94.6% accuracy.

2.3: Development of Artificial Neural Network (ANN) based model for detection of Leaf spot of Groundnut and Anthracnose of Mango.

Ramakrishnan and Sahaya (2015) utilized Back propagation algorithm to develop a detection and classification model for groundnut leaf diseases. It is a multilayer feed forward network consisting of two phases as propagation and weight update. The dataset contained 100 images of early leaf spot infected leaves, 100 images of Late leaf spot infected leaves and 100 images of leaf blight infected leaves. 80% images were used for training, 10% were used for testing and remaining 10% for validation purpose. The input RGB images of groundnut leaves were converted into Hue saturation value color image then while image processing color and texture features were extracted. Co-occurrence matrices i.e., a 2D array of C in which both row and column denotes a set of possible image values were used to extract texture features of dataset images. During this process contrast, energy, local homogeneity and entropy texture features were analyzed and extracted for model development. The model was trained using the neural network tool box and diseased leaf images as input. In the first step, the diseased images were fed to input layer, the input layer then created a key for identification which was forwarded to hidden layer of the neural network in the onward propagation phase. This activated the output functions of the network. In the second phase, the output estuary was multiplied and key activated to get the gradient of the weight and final detection results. The Matlab software tool box was used to implement the developed back propagation algorithm which classified groundnut leaf diseases with 97.4l % accuracy.

Kumar and Sowrirajan (2016) proposed an image-processing based approach to automatically classify the normal or diseased leaves (early leaf spot and late leaf spot) and also provide the cure for the same in groundnut. The RGB image samples of groundnut leaves were collected using high resolution camera. During preprocessing stage, the resizing of image to 256x256 pixels, color space conversion and region of interest selection was performed. Color, texture and geometric features of the image were extracted by the HSV conversion, GLCM and Lloyd's clustering respectively. BPN-FF classifier was used for classification based on learning with the training samples and thereby provided the information on the disease (early leaf spot and late leaf spot) as well as the respective control measures.

Vaishnnavel *et al.* (2019) classified four economically important groundnut diseases using KNN classifier. Ealy leaf spot, late leaf spot, rust and blight diseases of groundnut were selected for the study. 250 images of diseased groundnut leaves were collected using a smart phone camera. In the selected 250 images, 45 images were used for model training and 105 images were used for model testing. To increase the quality of the collected images, image pre-processing was done using snipping, smoothing and contrast enhancement methods. The input image was then transformed into binary masked image. This binary masked image was then converted into HSV image for segmentation to make it simple and more meaningful. Color co-occurrence method was used for feature extraction from the input images which produced three color co-occurrence matrix one for each of H, S, V. At the final stage K- Nearest Neighbor (KNN) classifier was applied to classify the diseases.

Devi *et al.* (2020) presented an image processing-based approach that automatically identifies and categorizes the groundnut diseases. The proposed method H2K was a fusion of Harris corner detector, HOG (Histogram on Oriented Gradient) and KNN classifier for accurate detection and classification of groundnut leaf diseases. The created image dataset contained 129 images of groundnut leaves both healthy and diseased taken from the fields of Vridachalam, Tamil Nadu. The diseased leaves included early leaf spot, late leaf spot, rust, bud necrosis and leaf blight symptoms. The images were captured using a digital camera and manually cropped for diseased leave area. These leave images were separated into six different classes that included the above said five diseases and healthy leaves. Histogram equalization technique and Gaussian Filter was used to enhance the quality of the image and to remove the noise or undesired distortion from the image. The RGB image was converted to binary image to find the region of interest for processing. Image segmentation was done by converting the image into HSV color space and H2 (Harris corner detector, and Histogram on Oriented Gradient) was used for feature extraction. Finally, disease classification was done using KNN classifier. The results showed that the proposed model performed well in detecting groundnut diseases with improved detection accuracy of 97.4% for early leaf spot, 100% for late leaf spot, 100% for rust, 90.9% for bud necrosis, 100% for blight and 100% for healthy leaves respectively.

Gowrishankar and Prabha (2020) proposed an image processing model using Artificial neural network (ANN) and GLCM for groundnut leaf disease diagnosis. Ground leave images infected with cercospora, alternaria, anthracnose and bacterial blight were selected for the study. In the proposed work, gray level co-occurrence matrix (GLCM) was used for feature extraction during image processing. These features were calculated by utilizing the threshold image for mapping with components of RGB input image to that image. The values of respective features were stored in feature library after extraction for model training. The features of leaf images extracted were energy, correlation, inertia and entropy. After feature extraction using GLCM, the images were classified using ANN and SVM into diseased and healthy. These classified images along with feature extraction values were fed to ANN and the relevant coefficients like weight and bias in ANN was set with appropriate values. The input and output values fed to ANN. Then support vector machine (SVM) was utilized to analyze the diseases on the groundnut leaves. The ANN model was implemented on 'Matlab' software. The accuracy of proposed ANN system was 98%.

Gowrishankar and Prabha (2020) applied threshold-based color segmentation technique for image segmentation along with Artificial neural network classifier to analyze leaf spot of groundnut. The groundnut leaf images for study were collected from a groundnut farm using high resolution camera, from which 6 set of color images were selected. A number of color image segmentation tests were performed on the selected set of images to obtain the different threshold values for complex, real and low intensity images. The ANN was trained using these threshold values. At the 37th epoch ANN reached the goal with the best performance of training iterations. The classified images by the ANN indicated 51.98% of leaves were affected by leaf spot disease with 98% accuracy.

Pujari *et al.* (2014) developed a recognition and classification system for produce affected by identically looking powdery mildew diseases of grape, mango, chilli, wheat, beans and sunflower. The diseased leave images were collected 7 days after first disease appearance using digital camera in RGB form. Gray level co-occurrence matrix method was used for texture feature extraction and colour features were extracted using RGB and HIS colour models. Knowledge-based and artificial neural network classifiers were adopted in the recognition and classification of images. For ANN based classification the authors used multilayer back propagation neural network (BPNN) with six output nodes and forty-two input nodes related to six categories of powdery mildew infected produce. They used sigmoid activation functions in the hidden layers. All the algorithms were applied using MATLAB 7.10. In knowledge-based classifier the overall average classification accuracy was 71.92 %, 80.60 % and 87.80 % using colour, texture and combined features respectively. In ANN the overall accuracy using colour, texture and combined features was 70.48 %, 70.07 % and 76.61 % respectively.

Suman and Dhruvakumar (2015) classified rice leaf diseases using shape and color features with SVM classifier. The shape and color features from the diseased leaves were extracted and combined using SVM classifier to classify bacterial leaf blight, brown spot, narrow brown spot and blast diseases of paddy. Images of normal and diseased leaf were acquired using digital camera with a white background to avoid reflections while capturing the images. The green color component of the infected portion was extracted and the intensity of the original gray scale image was subtracted by the green value therefore the spot detection was invariant of the brightness and age of the leaves. To remove unnecessary spots median filter was applied, which resulted in a noise free gray scale image. The gray scale images were binarized using thresholding technique. To distinguish between the uninfected and diseased leaves histogram approach was used with 8-connected component analysis image segmentation for segmentation of diseased leaf. Shape and color features were considered for classification of these different paddy diseases. Shape features were extracted using Principal Component Analysis and color features were extracted using Grid based color moments. Finally, SVM classifier was applied for classification of bacterial leaf blight, brown spot, narrow brown spot and rice blast diseases. The proposed technique achieved 70% accuracy for four different diseases *viz.* bacterial leaf blight, brown spot, narrow brown spot and blast.

Syafiqah *et al.* (2015) used Back-propagation algorithm to train a multi-layer perceptron (MLP) model of artificial neural network for classification of healthy and diseased leaves of *Phyllanthus elegans* Wall. Fifty sample images of both healthy and diseased plant leaves were collected using 8-Mega pixel smart phone camera. The process of image processing had three components which were contrast enhancement of images, image segmentation and feature extraction. The Multi-layer perceptron model was trained using back propagation method to classify healthy and diseased leaves. The back propagation network consisted of three layers, including an input layer with two parameters, a hidden layer with ten neurons and an output layer containing a single neuron. The network weights were set randomly in the learning phase. The input parameters were normalized between 0 and 1. Mean square function was used to check the network performance. Two multi-layer feed forward networks were used which were multi-layer perceptron (MLP) and radial basis function (RBF) to select the better performing model. The MLP model attained an accuracy of 90.3% with 9.7% error where the of test samples were more than the training samples. The classification accuracy of RBF model was 99.2% with only 0.8% error where the training samples were more than the test samples. The experimental results showed that the RBF network performed better than MLP network.

Moumita and Mantosh (2019) implemented Back-Propagation Neural Network with particle swarm optimization for identification and classification of leaf diseases of beans. In the proposed work, they took five types of leave images from an online source. The dataset consisted of five types of leaves: anthracnose, bacterial blight two types of leaf spots caused by cercospora and alternaria and healthy leaves of beans. To remove the unnecessary details in the leaf image the images were pre-processed. The images were resized and contrast enhancement was done. The green pixels of the images were masked to highlight the infected area. After that the RGB color image was converted to CIE L*a*b color model to reduce the dimension of image for image segmentation. K-means clustering technique was used for image segmentation. Different texture features: Contrast, Correlation, Entropy, Energy, and Homogeneity were extracted from the images using gray level co-occurrence matrix method. Classification was done using back-propagation and PSO. At first, back-propagation was used to train the feed forward neural network and then NN connection weights were further optimized using particle swarm optimization. For training and testing, the dataset was divided into 75% and 25%, respectively. The neural network was trained for four hundred iterations using back-propagation algorithm and then hundred iterations of PSO were run to optimize the weights. Implementation of the proposed method and other three existing methods have been done using ‘Matlab’. The developed model attained an accuracy of 96.72%.

Padol and Sawant (2019) introduced a fusion classification technique using Support vector machine (SVM) and Artificial neural network (ANN) classifiers to detect downy and powdery mildew of grapes. The images of disease infected leaves were collected from Pune and Nasik area. Few images were taken from internet. Total 137 grape leaf images were used, out of which 75 images were Downy mildew infected and 62 were Powdery mildew infected. For training 60 downy mildew and 50 powdery mildew images were used and for testing 15 downy mildew and 12 powdery mildew images were used. The image pre-processing was done to remove background noise and distortion using Gaussian filtering method. Then before thresholding the images were resized to 300×300 size. Image segmentation was carried out using k-means clustering technique. Both color and texture features of the images were extracted. The first analysis was made by using SVM and ANN then Fusion classification was done, which ensembled classifiers from SVM and ANN to regenerate base classifier for leaf diseases detection in order to improve classification process. The SVM classifier detected the downy mildew images with 93.33% accuracy and powdery mildew images with 83.33% accuracy. The ANN classifier detected the downy mildew images with 86.67% accuracy and powdery mildew images with 91.67% accuracy. The newly generated classifier using fusion classification technique performed best with increased accuracy of 100% for both downy and powdery mildew diseases of grape.

MATERIAL AND METHODS

CHAPTER III

Material and Methods

Following the experimental approach, two distinct approaches were adopted for the development of the CNN and ANN models. In the CNN model development, RGB images of disease-infected plant parts were utilized as input data. These RGB images were captured using standard visible light cameras. Alternatively, the ANN model development was focused on using thermal images of the infected plant parts. These thermal images were acquired using a dedicated thermal camera capable of capturing thermal radiation emitted by the plants. By employing these two different imaging modalities, the intention was to explore and compare the effectiveness of RGB and thermal data in detecting and classifying plant diseases.

This methodological diversity enhances our understanding of the potential applications of both RGB and thermal imaging techniques in the domain of plant pathology for detection of different plant diseases. The material and methodology employed to develop the proposed CNN and ANN models for disease detection is outlined in a stepwise manner below:

3.1 Material:

3.1.1 Hardware used:

The proposed work was based on Neural Networks which utilizes image processing and object detection methods to develop the neural network model capable of detection and classification of diseases under study. So limited no. of hardware tools were required for implementation.

3.1.1.1. FlirOne thermal camera

Weight: 2.75 ounces, ~78 grams, Dimensions: 72 x 26 x 18mm, Battery capacity: 350 mA-h, Visible camera: VGA (used for FLIR®MSX blending), Sensitivity: ability to detect temperature differences as small as 0.18° F (0.1° C), IR resolution- 480 × 640, Device compatibility: Android products containing a micro-USB.

3.1.1.2. Smart phone

Vivo S1 (RAM 4.00GB, 2.0 GHz Octa-core, Front 32MP / Rear 16MP+8MP+2MP camera), 1080 × 2340 pixels resolution.

3.1.1.3. Laptop

ASUS Vivo book 14 (11^{th} Gen, Intel Core i5, 1TB HDD, 256GB SSD).

3.1.2. Software used:

3.1.2.1 Teachable Machine

Teachable machine (teachable.withgoogle.com) is a open source, web-based, Graphical user interface (GUI) tool used to develop customized machine learning classification models without special technical expertise using webcam, images, or sound. It unbales students, teachers, designers, and others to learn about creating CNN model using machine learning for classification, detection and learning purposes. It is a flexible and approachable interface for creating machine learning classification model without coding expertise. It has a set of product design and technical decisions that enable learning and experimentation for new user. It is a good example of how structured learning content surrounding the tool supports users accessing machine learning concepts. Teachable machine uses transfer learning to find patterns and trends within the images or sound samples, and create a simple and easy classification model. With help of teachable machine, a user can add their own data and retrain a model on top of a previously trained base model that has learned a specific domain from a large dataset.

3.1.2.2 Mobiroller

Mobiroller is a free mobile application making platform that allows the users to create mobile applications without coding knowledge. All the applications are native and compatible with Android and iOS. This platform provides a cost-effective application-making platform for small and medium size businesses, content creator, agencies or freelancers. Using teachable machines classification model link, users can create an android or iOS supported mobile application with required contents which can be made available to other users through publication on Google Play store. The users can also monetize it as per their requirement.

3.1.2.3 Google Colaboratory

Google colaboratory is a free online cloud-based Jupyter notebook platform that allows the users to train Machine learning and Deep learning models on CPUs, GPUs, and TPUs. This platform allows the user to run code entirely on the cloud, which helps in training large scale machine learning and deep learning models without access to a powerful machine or a high-speed internet access. It supports both GPU and TPU instances, making it a perfect tool for deep learning and data analytics. It can be accessed remotely from any machine through a browser hence its well suited for commercial purposes as well.

3.1.2.4 Gradio

Gradio is an open-source free Python package used to create easy-to-use, customizable User Interface for Machine learning model, and API, and even an arbitrary Python function using fewer codes. Gradio integrates with the most popular Python libraries, like Scikit-learn, PyTorch, NumPy, Seaborn, Pandas, Tensor Flow and others. Gradio GUI can be integrated directly into the Jupyter notebook or can be shared as a link with anyone. It is the fastest way for demonstration of the machine learning model with a friendly web interface which can be used by anyone.

3.1.2.5 NeuroSolutions

NeuroSolutions is an artificial neural network development software, which combines a modular, component-based network interface with advanced learning algorithms, such as Conjugate Descent Gradient, Levenberg Marquardt and Backpropagation. It is used to design, train and deploy neural networks models to perform different tasks such as data mining, classification and time-series prediction. It is based on the concept that neural networks can be broken down into a fundamental set of neural components, which can be connected together to develop a network capable of solving very complex problems. Neuro Solutions provide three steps to build neural network models, which are Data manager, Neuro Builder and Neural Expert. Data manager helps to import data from input files and load the data directly into a NeuroSolutions software to use the data to perform pre-processing and data analysis to create a new neural network. Neuro Builder contains different learning algorithms (architectures) required to build the neural network. After the problem type and size of the user’s data is provided to the neural expert it selects the neural network size and architecture that will likely produce a good solution to the problem.

3.2 Methods:

The research objectives were pursued and the CNN and ANN models were developed for the detection of groundnut early leaf spot, anthracnose, and powdery mildew of mango using the following methodology:

3.2.1 Data collection for dataset creation of Leaf spot Groundnut, Anthracnose and Powdery mildew of Mango.

3.2.1.1 Selection of plants

Groundnut is an annual legume crop and the incidence of early leaf spot can be found in all seasons damaging the plant by reducing the leaf area available for photosynthesis and causing defoliation. One month old groundnut plants were tagged (100) during Kharif - 2021 for image collection of early leaf spot of groundnut.

Mango grafts at nursery stage are more susceptible to anthracnose disease hence 45 days old seedlings with different stages of leaf development were selected randomly to ensure image collection from initial stage of symptom development at different leaf stages during Kharif - 2021. For image collection of powdery mildew of mango newly sprouted mango inflorescences (100) during January-2020 were selected and tagged.

3.2.1.2 Data collection technique

Two image collection techniques were followed which were as follows.

3.2.1.2.1 Random RGB image collection

RGB images of groundnut leaves infected with early leaf spot, mango leaves infected with anthracnose and mango inflorescence infected with powdery mildew were taken at random using smart phone camera. The Vivo S1 smart phone with 1080 × 2340 pixels resolution was held at a distance of 3 to 4 meter from the plant sample while capturing the image. It was made sure that the captured image covered the complete plant area of interest (Singh *et al.*, 2019). The images were collected in the morning during 10:00 AM – 11:30 AM. The dataset of each disease *viz*. early leaf spot of groundnut, anthracnose and powdery mildew of mango contained images of disease symptoms of various stages.

The collected images contained the disease symptoms of the respective disease only and were free from other insect pest attack. Healthy leave images were also collected in similar manner for groundnut leaves, mango leaves and mango inflorescence. The RGB image dataset contained 700 images of each diseased and healthy plant sample under study (PLATE I & PLATE II), out of which 500 images were used as training dataset for model training and 200 images were used as validation dataset for model validation.

PLATE I

RGB image dataset of Groundnut and Mango crops

Healthy groundnut leaf

Leaf spot infected leaf of groundnut

Healthy mango leaf

Anthracnose infected mango leaf

PLATE II

Healthy mango inflorescence

Powdery mildew infected mango inflorescence

3.2.1.2.2 Tagged Thermal image collection

Healthy groundnut leaves, mango leaves and mango inflorescence (100) were selected and tagged. To avoid wear and tear the tags were coated with wax. Tagging of plants and image collection was started on the same day. For thermal image collection the FlirOne thermal camera was carefully mounted on the smartphone and held at a distance of 3 to 4 meter to capture the image. It was made sure that the image covered the entire leaf area and mango inflorescence. The image collection was continued till full visual symptom development (PLATE III, IV, V, VI, VII & PLATE VIII).

3.2.1.3 Image processing

3.2.1.3.1 RGB image processing

The randomly collected RGB images of early leaf spot of groundnut, anthracnose and powdery mildew of mango using smartphone camera were transferred to the laptop and saved into a single folder for each disease. Each image was checked thoroughly to remove blur images. RGB images which clearly showed specific disease symptoms without any other disease symptoms were selected. The RGB images were saved into a single folder for each disease since they were collected randomly. The RGB image dataset of each disease was renamed as per the disease name and image no.

3.2.1.3.2 Thermal image processing

Data grouping and rename

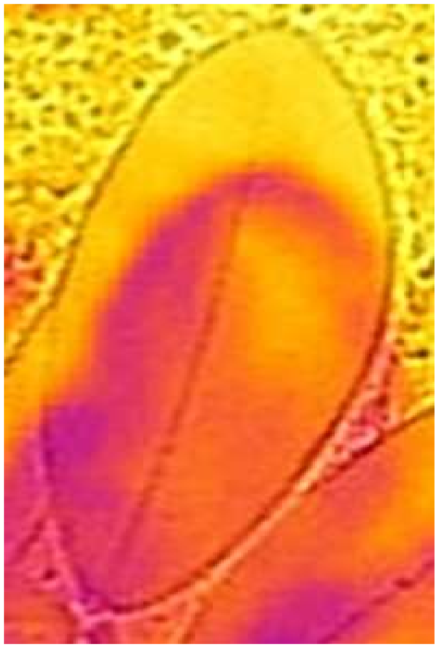
Collected thermal images were segregated and grouped on the basis of no. of observation/tag and observation day. Thermal images were renamed as per date and tag no. and were arranged in sequence.

Temperature difference data extraction from thermal image dataset of groundnut leaf spot and mango anthracnose diseases

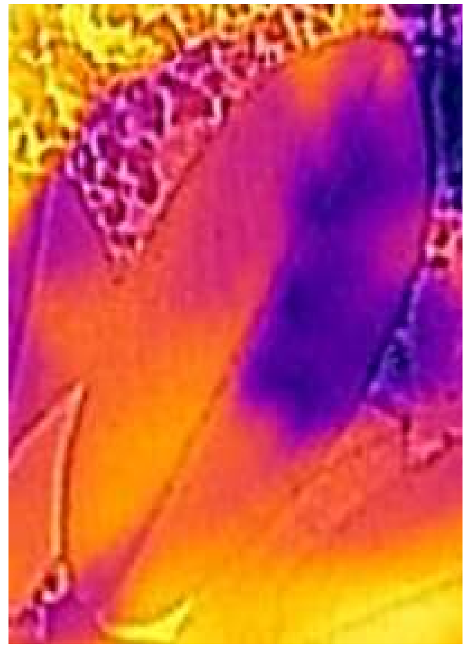
The collected thermal images of groundnut early leaf spot, mango anthracnose and mango inflorescence were stored in computer according to tag no. and date. Each image of a single tag was viewed in FlirOne thermal camera image view application. This application provided a window to view thermal images and extract necessary information from them. The FlirOne application provided information regarding reflectance temperature of plant surface, IR resolution, time and date on which the image was taken and emissivity. The thermal image

PLATE III

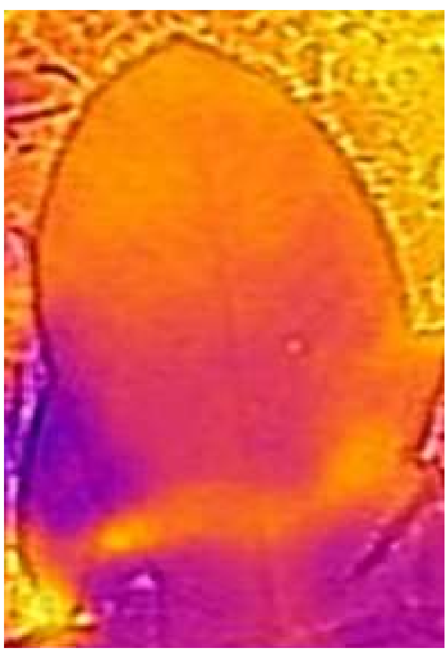
Thermal image dataset of Groundnut Early leaf spot



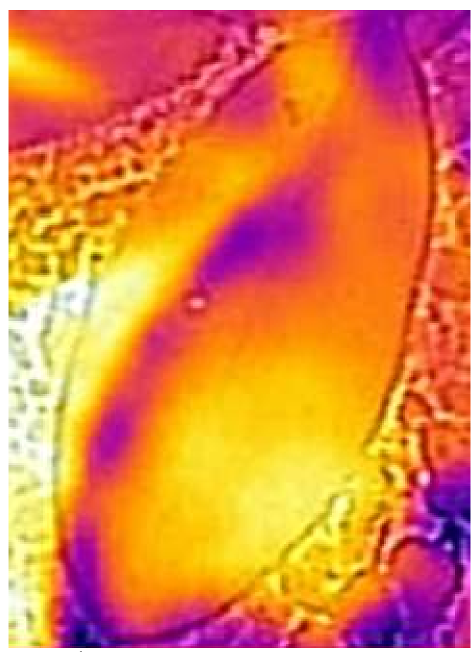
Day 1



1 Day before symptom appearance

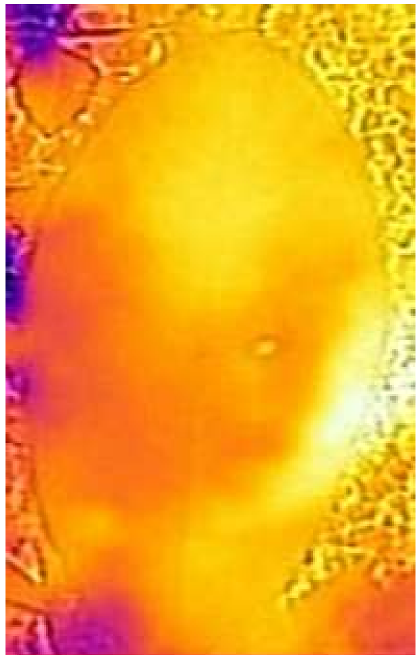


Day of symptom appearance



3rd day of symptom appearance

PLATE IV



5th day of symptom appearance

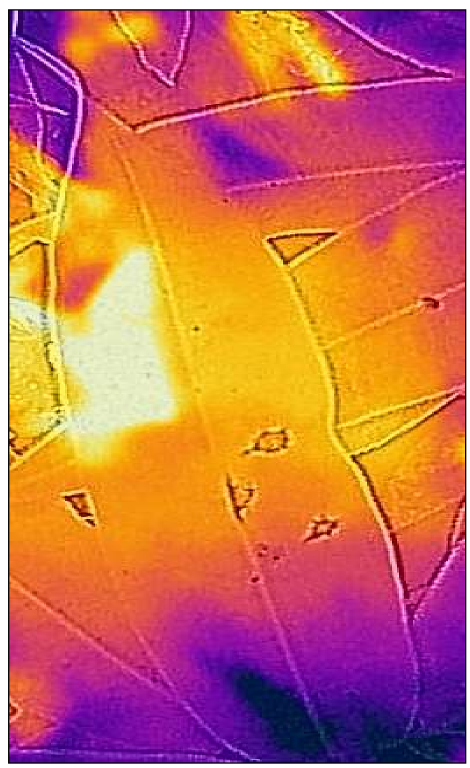
PLATE V

Thermal image dataset of Mango Anthracnose

Day 1

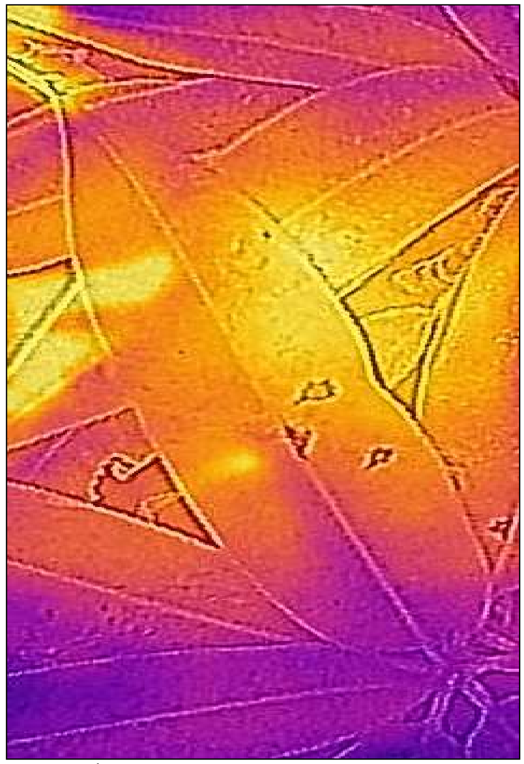
1 Day before symptom appearance

Day of symptom appearance



3rd day of symptom appearance

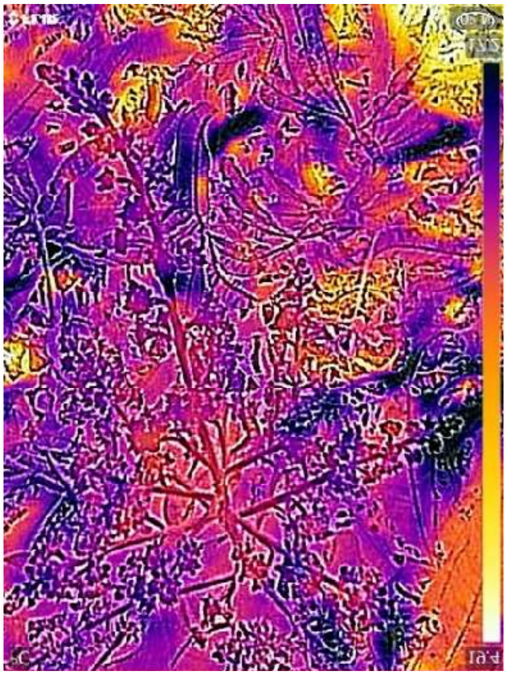
PLATE VI



5th day of symptom appearance

PLATE VII

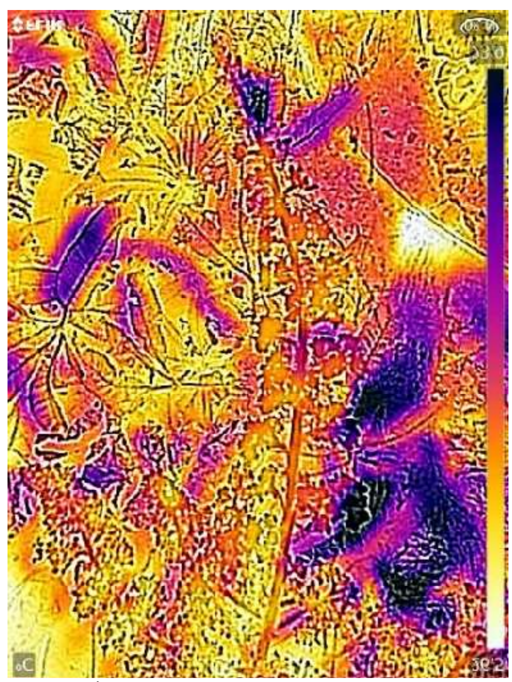
Thermal image dataset of Mango Powdery mildew



Day 1

1 Day before symptom appearance

Day of symptom appearance



3rd day of symptom appearance

PLATE VIII

5th day of symptom appearance

could be viewed into RGB image format using FlirOne thermal camera image view application.

Temperature values of healthy and diseased area of selected thermal images were recorded for both early leaf spot of groundnut and anthracnose of mango. Temperature difference in healthy and diseased leaf area for each tag from day one on which data collection was started, one day prior to symptom appearance, day of symptom appearance 3rd day of symptom appearance and 5th day of symptom appearance were calculated. For day one and the day before symptom appearance two temperature values from the healthy areas were taken. After symptom appearance the temperature value of diseased and healthy leaf area was taken. The temperature values of healthy and diseased leave areas were compared using paired ‘t’ test. Here, both the temperature values were dependent on each other therefore, paired ‘t’ test was selected.

The observations of these three days were classified into temperature difference grade based on minimum and maximum temperature difference and were used to develop artificial neural network model for detection of early leaf spot of groundnut (Table 1) and anthracnose of mango (Table 2).

Table 1: Temperature difference-based grade scale for early leaf spot of groundnut

Grade	Temp. difference in ^{0}C between diseased and healthy leaf area of groundnut
1	0.1 - 0.5
2	0.6 - 1
3	1.1 - 1.5
4	1.6 - 2
5	2.1 - 2.5

Table 2: Temperature difference-based grade scale for anthracnose of mango

Grade	Temp. difference in ^{0}C between diseased and healthy leaf area of mango
1	0.1 - 0.3
2	0.4 - 0.6
3	0.7 - 0.9
4	1 - 1.2
5	1.3 - 1.5
6	1.6 - 1.8

3.2.2 Development of Convolutional Neural Network (CNN) based model for detection of Early Leaf spot of groundnut, Anthracnose and Powdery mildew of mango.

A CNN (Convolutional Neural Network) model is like a smart computer program that can look at an RGB image of an infected plant part and tell us which disease it might have. It first examines the image carefully, looking for different patterns and colors. For example, it looks for spots, discoloration, or unusual shapes in the image of infected plant parts. The model then compares what it sees in the image with patterns it has learned from many other images of healthy and diseased plants which have been used to train the model. By using this training, the model can figure out which patterns and colors in the image are most similar to the ones it has seen before in images of specific diseases. It assigns a higher probability to the disease that matches the patterns it finds in the image. The model continues this process, analyzing the image and checking for different disease patterns. It then gives a prediction of which disease the plant might have based on the patterns it detected.

So, in simpler terms, a CNN model acts like a knowledgeable assistant that looks at the RGB image of the infected plant part, compares it with its knowledge of healthy and diseased plants, and makes an prediction about which disease the plant might be suffering from based on the patterns it finds in the image. This helps plant experts to identify and diagnose plant diseases more efficiently.

In the development of CNN-based detection models for groundnut early leaf spot, anthracnose, and powdery mildew of mango, two distinct approaches were employed. For the detection of groundnut early leaf spot, a CNN-based model was developed using the Teachable Machines website. This approach involved training and fine-tuning the model using groundnut leaf images, enabling it to recognize and classify instances of early leaf spot. An alternative method utilized the VGG-16 pre-trained CNN model. Through adaptation and fine-tuning using groundnut leaf images, this model was specifically tailored to detect and classify cases of early leaf spot.

For the detection of anthracnose and powdery mildew of mango, a single approach relying on the Teachable Machines website was employed. Using this approach, a CNN-based detection model was developed using healthy and diseased mango plant part images. The model was trained to identify and classify instances of anthracnose and powdery mildew accurately. The process of developing these models involved the following steps:

3.2.2.1 Teachable machine model and Android application development for detection of Early Leaf spot of groundnut, Anthracnose and Powdery mildew of mango.

Among the three projects available on teachable machine image project was selected to create the image-based detection model for diseases under study. Three different image project models were prepared for groundnut and mango diseases under study using RGB images. In early leaf spot of groundnut detection model, two classes were created namely healthy leaf and infected leaf. Similarly, in Mango disease detection models, two classes were created for each disease detection model namely, healthy leaf and infected leaf for anthracnose detection model, healthy inflorescence and infected inflorescence for powdery mildew detection.

Two main options were provided for uploading the image dataset as input. In first option, real time webcam could be used to collect the real time image dataset. In second option, images from gallery could be uploaded manually with drag-drop method or copy-paste method and Google drive folder could also be connected directly to access the image dataset online as input. Diseased image dataset collected for groundnut early leaf spot, anthracnose and powdery mildew of Mango were selected from computer to upload as input dataset in respective disease detection models. 500 RGB images of each class type were uploaded manually using drag-drop method for model training and remaining 200 images were used for model validation.

Three training parameters were needed to train the disease detection models. Batch size i.e., number of samples processed before model updated was set to 16, number of epochs at which the model was run for training was set to 50 and 0.001 learning rate was set for both Groundnut and Mango disease detection models. After setting training parameters, the model was trained. The process of model training was completed within 3 to 5 minutes. Training time was varying from model to model based on no. of classes and no. of dataset. After model was trained, model preview window was used to validate the results. To create Teachable machine model link, firstly, the model was uploaded online after which the link was created. These links were used to create android application of groundnut and mango disease detection.

Confusion matrix analysis

The teachable machine disease detection model's performance was tested using confusion matrix analysis of each disease detection model. Each row of the matrix represented the instants in actual class and each column represented instances in predicted class.

Basic terminology used in confusion matrix were true positive, true negative, false positive and false negative. When the disease class was predicted correctly it was considered true positive, if the model predicted no disease i.e., healthy and there was no disease on leaf then it was true negative, when the model predicted disease class and if there was no disease then it was false positive, similarly if the model predicted no disease i.e., healthy then it was considered at false negative. The equations used to calculate Accuracy, Recall, Precision and F1 Score using Confusion matrix given below.

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

Where,

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

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

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

$$\text{F1 score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

3.2.2.2 Android application development

'Mobiroller' open-source website was used to develop the disease detection model using developed teachable machine model links for early leaf spot of groundnut, anthracnose and powdery mildew of mango.

3.2.2.3 VGG-16 CNN model development for detection of Early Leaf spot of groundnut.

VGG-16 is one of the most commonly used CNN architecture for visual object detection procedures. This model was created by Simonyan and Zisserman in 2014. The precise architecture of VGG-16 model shown in Fig.1 as follows:

- The first and second convolutional layers comprised of 64 feature kernel filters with filter size 3×3. As input image (RGB image with depth 3) passed into first and second convolutional layer, dimensions changes to 224×224×64. Then the resulting output passed to max pooling layer with a stride of 2.
- The third and fourth convolutional layers consisted of 124 feature kernel filters with 3×3 size. These two layers were followed by a max pooling layer with stride 2 and the resulting output reduced to 56×56×128.
- The fifth, sixth and seventh convolutional layers had 3×3 kernel size. All three used 256 feature maps. These layers were followed by max pooling layer with stride 2.
- Eighth to thirteen contained two sets of convolutional layers with kernel size 3×3. All these sets of convolutional layers had 512 kernel filters. These layers were followed by max pooling layer with stride 1.
- Fourteen and fifteen layers were fully connected hidden layers of 4096 units followed by a dense output layer (Sixteenth layer).

Data augmentation

The primary RGB dataset of groundnut early leaf spot contained 500 images of healthy and infected leaves. Data augmentation was done to increase the number of dataset

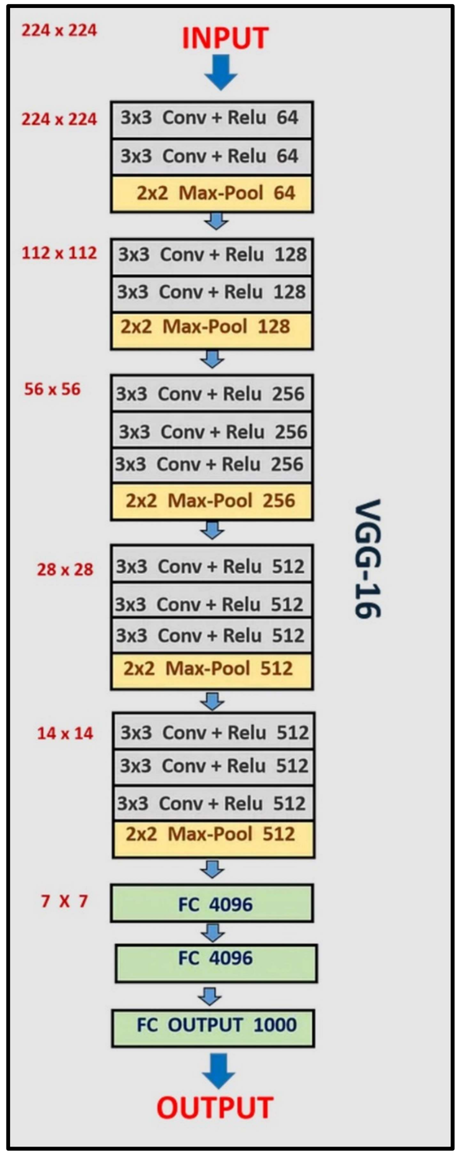


Fig: 1. VGG-16 model architecture

and resize the images into uniform parameters. The dataset was augmented with 0.2 width-shift range, 0.6 height-shift range, 0.3 zoom range, 0.4 shear range and 10 rotation range. After augmentation, the augmented dataset contained 2000 images of both healthy and infected groundnut leaves.

Codes to train VGG-16 model for detection of early leaf spot of groundnut

Step 1: Mounting Google Drive in Google Colab

```
from google.colab import drive

drive.mount('/content/gdrive')
```

This code prompted a link to authenticate google account to mount the google drive in google colab on which the research data is stored. After this, google drive files and folders could be accessed from within colab.

Step 2: Importing required libraries

```
import os

import cv2

import numpy as np

from sklearn.model_selection import train_test_split

import shutil

import matplotlib.pyplot as plt

%matplotlib inline

from tensorflow.keras.preprocessing.image import load_img, img_to_array, array_to_img, ImageDataGenerator

from keras.applications.vgg16 import VGG16

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, InputLayer

from keras.models import Sequential

from keras.layers import BatchNormalization

from keras import optimizers
```

Libraries like TensorFlow for creating deep learning model, ImageDataGeneratore for loading and image augmentation, Numpy for numerical computing, Matplotlib for visualizing the images, and OS for working with directories and file paths were installed.

Step 3: Setting data path

main_path = '/content/gdrive/MyDrive'

groundnut_healthy = main_path + '/Gnut healthy/'

groundnut_leaf_spot = main_path + '/Gnut leaf spot/'

groundnut_healthy_data = [main_path+'/Gnut healthy/'+f for f in os.listdir(groundnut_healthy)]

groundnut_leaf_spot_data = [main_path+'/Gnut leaf spot/'+f for f in os.listdir(groundnut_leaf_spot)]

len(groundnut_healthy_data),len(groundnut_leaf_spot_data)

This code was used to refer or access the folders of google drive in which the image dataset was stored. In this code, the 'os' module was used to manipulate file paths. The 'os.path.join()' function was used to concatenate the 'main_path' variable with a specific file or directory name as 'groundnut healthy' and 'groundnut leaf spot' to form the complete path.

Step 4: Print image

print(groundnut_healthy_data[0:5])

print(groundnut_leaf_spot_data[0:5])

The print function was used to display the output of each selected folder from google drive.

Step 5: Assign labels to the image dataset classes

images = np.array(groundnut_healthy_data + groundnut_leaf_spot_data)

labels= np.array([0]*len(groundnut_healthy_data)+[1]*len(groundnut_leaf_spot_data)).astype('float 32'

The images array was created by concatenating 'groundnut_healthy_data' and 'groundnut_leaf_spot_data' using 'np.array()'. The 'labels' array was created by generating a list of labels. It assigned label ‘0’ to ‘groundnut_healthy_data’ and label ‘1’ to ‘groundnut_leaf_spot_data’. The list comprehension ‘[0]*len(groundnut_healthy_data)’ created a list of zeros with a length equal to the size of ‘groundnut_healthy_data’. Similarly, ‘[1]*len(groundnut_leaf_spot_data)’ created a list of ones with a length equal to the size of ‘groundnut_leaf_spot_data’. These lists were then concatenated using the ‘+’ operator, and the resulting list was converted to a NumPy array. Finally, the ‘astype('float32')’ function was used to convert the data type of the labels to ‘float32’.

Step 6: Dataset split into training, testing and validation dataset

validation_ratio = 0.2

X_train, X_test, y_train, y_test = train_test_split(images,labels,test_size=0.33)

X_train, X_validation, y_train, y_validation = train_test_split(X_train,y_train,test_size=validation_ratio)

X_train.shape,X_test.shape,y_train.shape, y_test.shape

len(X_train),len(X_test),len(y_train),len(y_test)

This code was used to split the groundnut healthy and leaf spot image data into training, testing and validation sets. The variable ‘validation_ratio’ was set to 0.2 to allocate 20% of the data for validation. The ‘train_test_split()’ function was called with images and labels as the input arrays, along with the ‘test_size’ parameter set to 0.33. This splits the data into training and testing sets, where 33% of the data allocated for testing. The resulting arrays were assigned to ‘X_train’, ‘X_test’, ‘y_train’, and ‘y_test’. Another call to ‘train_test_split()’ was made, this time with ‘X_train’ and ‘y_train’ as the input arrays. The ‘test_size’ parameter was set to ‘validation_ratio’, which was 0.2. This splits the training data into training and validation sets, with 20% of the data allocated for validation. The resulting arrays were assigned to ‘X_train’, ‘X_validation’, ‘y_train’, and ‘y_validation’.

‘.shape’ attribute of NumPy arrays was used to obtain the shapes of the ‘X_train’, ‘X_test’, ‘y_train’, and ‘y_test’ arrays,. ‘len()’ function was used to check the number of sample in each class.

Step 7: Import modules/functions from TensorFlow.Keras library for data augmentation

from tensorflow.keras.preprocessing.image import load_img, img_to_array, array_to_img, ImageDataGenerator

The ‘load.img’ function was used to load an image from the given file path for image pre-processing and augmentation. ‘img_to_array’ function was used convert the images into Numpy array before loading for pre-processing. Finally, ImageDataGenerator function was used for data augmentation.

Step 8: Set image dimensions to resize images into uniform size

IMG_WIDTH=224

IMG_HEIGHT=224

IMG_DIM = (IMG_WIDTH, IMG_HEIGHT)

The images were resized to uniform height and width of 224 pixels for data augmentation.

Step 9: Image Pre-Processing

X_train = np.array([img_to_array(load_img(img, target_size=IMG_DIM)).astype('float32')/255 for img in X_train])

X_test = np.array([img_to_array(load_img(img, target_size=IMG_DIM)).astype('float32')/255 for img in X_test])

X_validation = np.array([img_to_array(load_img(img, target_size=IMG_DIM)).astype('float32')/255 for img in X_validation])

This code performed image loading, resizing, conversion to NumPy array, normalization, and replaced the original image paths with the pre-processed image data in the form of NumPy arrays.

The images are resized to the dimensions specified by ‘IMG_DIM’ and then converted to NumPy arrays. The pixel values are also normalized to a range of [0, 1] by dividing by 255. After this pre-processing step, ‘X_train’, ‘X_test’, and ‘X_validation’ contained the processed image data ready to be used for training, testing, and validation set.

Step 10: Preparing the data using ImageDataGenerator

```
dataGen = ImageDataGenerator(width_shift_range=0.2,
                        # 0.1 = 10%
                        height_shift_range=0.6,
                        zoom_range=0.3,  # 0.2 MEANS CAN GO FROM 0.8 TO 1.2
                        shear_range=0.4,  # MAGNITUDE OF SHEAR ANGLE
                        rotation_range=10)  # DEGREES
dataGen.fit(X_train)
batches = dataGen.flow(X_train, y_train,batch_size=20)
X_batch, y_batch = next(batches)
```

The 'width_shift_range' was set to 0.2, used to horizontally shift the images up to 20% of their width in either direction for augmentation. Other parameters like 'height_shift_range' of 0.6 for vertical shift up to 60% of the images height, 'zoom_range' of 0.3 for zooming the images in or out up to 30%, 'shear_range' of 0.4 to shear the images up to 40 degrees and 'rotation_range' of 10 to rotate the images in either direction by up to 10 degrees was used for image data augmentation to increase the diversity and robustness of the training data.

By calling 'dataGen.fit(X_train)', the 'ImageDataGenerator' object 'dataGen' analysed the training data and calculated the necessary statistics or parameters for the specified data augmentation techniques. These statistics or parameters were then used during the training process to generate augmented images on-the-fly. After data augmentation, 'batch' function was used to create batches of augmented data.

Step 11: Convert Numpy array into an image

```
array_to_img(X_train[0])
print(X_train[:10],y_train[-10:])
```

To convert a NumPy array ‘X_train[0]’ back to a PIL image object, ‘array_to_img()’ function was used from the ‘tensorflow.keras.preprocessing.image’ library. This function was used to display the images into their original format.

Step 12: Modify VGG-16 model

```
from keras.models import Model

vgg = VGG16(include_top=False, weights='imagenet', input_shape = (IMG_HEIGHT,IMG_WIDTH,3))

output1 = vgg.layers[-1].output

output2 = Flatten()(output1)

vgg = Model(vgg.input, output2)

for layer in vgg.layers:
    layer.trainable = False

vgg.summary()
```

Keras librarywas used to build a VGG-16 model with the last fully connected layers removed (include_top=False). Then a flatten layer was added to the output of the VGG-16 model. Finally, a new model was created called ‘vgg’ using the input and the flattened output. After creating the model, all layers in the ‘vgg’ model were set as non-trainable, effectively freezing their weights and preventing them from being updated during training.

This code created a modified ‘VGG-16’ model where only the convolutional layers were used for feature extraction from image (Fig.2), and the fully connected layers were removed.

Step 13: Rebuild the model using ‘sequential’ API in keras

```
model = Sequential()

model.add(vgg)

input_shape = (IMG_HEIGHT,IMG_WIDTH,3)

model.add(Flatten())

model.add(Dropout(0.3))
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig: 2. Modified VGG-16 model architecture

Step 15: Compile the model

```
model.compile(loss='binary_crossentropy',
optimizer=optimizers.Adam(learning_rate=0.001), metrics=['accuracy'])

model.summary()
```

The model was configured and compiled with ‘binary cross-entropy’ loss function, ‘Adam’ optimizer with a learning rate of 0.001 and model’s performance was monitored using accuracy function. Fig.3 shows the summary of the model’s architecture, including the layer names, output shapes and number of parameters. The total number of parameters in the model was 27,510,189 with 12,795,501 trainable parameters and 14,714,688 non-trainable parameters.

Step 16: Model training

```
history = model.fit(dataGen.flow(X_train, y_train.reshape((-1,1))),
        epochs=100,
        validation_data=(X_validation,y_validation),
        shuffle=True,
        verbose=1)
```

The model was trained using ‘fit’ function for 100 epochs. Shuffle command was used to shuffling the training data before each epoch, so that the model can learn more effectively. The progress bars during training with a detailed log output were viewed by setting verbosity mode to 1. History command was used to analyze the training progress and visualize the learning curves.

Step 17: Analyze the model performance

```
accuracy = history.history['accuracy']

val_accuracy = history.history['val_accuracy']

loss = history.history['loss']

val_loss = history.history['val_loss']
```

This code allowed to access the recorded metrics during the training process, which were used to analyze the model's performance over epochs, plot learning curves, and compare the training and validation performance of the model.

Step 18: Plot training and validation accuracy over epochs

```
epochs = range(len(accuracy))

plt.plot(epochs, accuracy, label='Training Accuracy')

plt.plot(epochs, val_accuracy, 'b', label='Validation Accuracy')

plt.title('Training and Validation Accuracy')

plt.legend()
```

The code plotted an accuracy graph of training and validation accuracy with ‘x’ axis representing the number of epochs and the ‘y’ axis representing the corresponding accuracy values.

Step 19: Plot training and validation loss over epochs

```
epochs = range(len(loss))

plt.plot(epochs, loss, label='Training loss')

plt.plot(epochs, val_loss, 'b', label='Validation loss')

plt.title('Training and Validation Loss')

plt.legend()
```

The code plotted loss graph of training and validation loss with ‘x’ axis representing the number of epochs and the ‘y’ axis representing the corresponding loss values.

Step 20: Model accuracy on test set

```
score = model.evaluate(X_test,y_test)

print('Test Accuracy: {}'.format(score[1]))
```

The code calculated and displayed the accuracy of the trained model on the test set.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 model (Functional)          (None, 25088)             14714688

 flatten_1 (Flatten)         (None, 25088)             0

 dense (Dense)               (None, 500)               12544500

 dense_1 (Dense)             (None, 500)               250500

 dropout (Dropout)           (None, 500)               0

 dense_2 (Dense)             (None, 1)                 501

=================================================================
Total params: 27,510,189
Trainable params: 12,795,501
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig: 3. Model architecture summary

This code allowed to access the recorded metrics during the training process, which were used to analyze the model's performance over epochs, plot learning curves, and compare the training and validation performance of the model.

Step 18: Plot training and validation accuracy over epochs

```
epochs = range(len(accuracy))

plt.plot(epochs, accuracy, label='Training Accuracy')

plt.plot(epochs, val_accuracy, 'b', label='Validation Accuracy')

plt.title('Training and Validation Accuracy')

plt.legend()
```

The code plotted an accuracy graph of training and validation accuracy with ‘x’ axis representing the number of epochs and the ‘y’ axis representing the corresponding accuracy values.

Step 19: Plot training and validation loss over epochs

```
epochs = range(len(loss))

plt.plot(epochs, loss, label='Training loss')

plt.plot(epochs, val_loss, 'b', label='Validation loss')

plt.title('Training and Validation Loss')

plt.legend()
```

The code plotted loss graph of training and validation loss with ‘x’ axis representing the number of epochs and the ‘y’ axis representing the corresponding loss values.

Step 20: Model accuracy on test set

```
score = model.evaluate(X_test,y_test)

print('Test Accuracy: {}'.format(score[1]))
```

The code calculated and displayed the accuracy of the trained model on the test set.

Step 21: Make predictions on test set

```
predict = model.predict(X_test)

predict
```

Step 22: Save and load model to Google drive using TensorFlow

```
import tensorflow as tf

#model.save('/content/gdrive/MyDrive/CNN/groundnutmodel.h5')

model = tf.keras.models.load_model('/content/gdrive/MyDrive/CNN/groundnutmodel.h5')
```

Step 23: Create a new array 'x' based on the predicted values and print first 10 elements of 'x' array and 'x_test' array

```
x = []

for i in predict:

  if i<=0.5:

    x.append(0)

  else:

    x.append(1)

x = np.array(x).astype('float32')

x[0:10]

X_test[0:10]
```

Step 24: Import accuracy function from 'sklearn.metrics'

```
from sklearn.metrics import accuracy_score
```

Step 25: Calculate the accuracy score between the predicted labels (x) and the true labels (y_test) using the 'accuracy_score' function

```
acc = accuracy_score(x,y_test)

print('Accuracy Score : ',acc*100)
```

Step 26: Install Gradio library

```
!pip install gradio

import gradio as gr

import cv2

import numpy as np
```

Step 27: Make predictions on new input dataset

```
def groundnut_disease(img):

  img = np.asarray(img)

  img = cv2.resize(img,(224,224))

  img = img.reshape(1,224,224,3)

  while True:

    predict = model.predict(img)

    if predict<=0.5:

      return 'Healthy crop'

    elif predict>=0.5:

      return 'Leaf spot'
```

Step 28: Create Gradio interface for the developed model

```
#outputs = gr.output.Textbox()

app = gr.Interface(fn=groundnut_disease,inputs="image",outputs="text",description='This is Ground disease classification model')
```

Step 29: Launch the gradio interface and use the model for detection of early leaf spot of groundnut

```
app.launch(debug=True)
```

After the gradio interface was launched, the final code script of groundnut early leaf spot disease detection model was saved in the jupyter notebook as 'VGG-16 groundnut model' for further use.

3.2.3 Development of Artificial Neural Network (ANN) based model for detection of Early Leaf spot of groundnut and Anthracnose of mango.

An Artificial Neural Network (ANN) model can be thought of as a computer program that mimics the functioning of the human brain to help us detect and identify plant diseases. The ANN model is trained on a large dataset that includes examples of different plant diseases after which it learns to recognize specific patterns or signals associated with each disease.

When we provide the ANN model with the temperature difference data and grade assigned to it obtained from a thermal image of an infected plant part, it carefully examines this information. It looks for variations between the temperature difference values and temperature grade assigned to it. Using its knowledge of patterns acquired during training, the ANN model compares the temperature difference data and grade patterns it observes with the ones it has seen before. It looks for similarities and tries to match the observed patterns with those associated with specific diseases. Based on these comparisons, the ANN model predicts the given temperature difference data belongs to which temperature grade. It assigns a higher probability to the temperature grade that best matches the observed temperature difference patterns. The general procedure required to develop ANN disease detection model was as follows (Fig.4).

3.2.3.1 Normalization of Data

Data normalization was essential to limit the data range within 0 to 1 interval before analysing the data using ANN. Data normalization ensured that the data series had normalized value closer to 0.5 and within 0 to 1 range. The following equation was used for normalizing the dataset (Kumar, 2011).

$$X_{norm} = 0.5\left(\frac{x_0 - x_{mean}}{x_{max} - x_{min}}\right) + 0.5$$

Where,

X_{norm} = normalized value, x_0 = original value, x_{mean} = mean value, x_{max} = maximum value and x_{min} = minimum value.

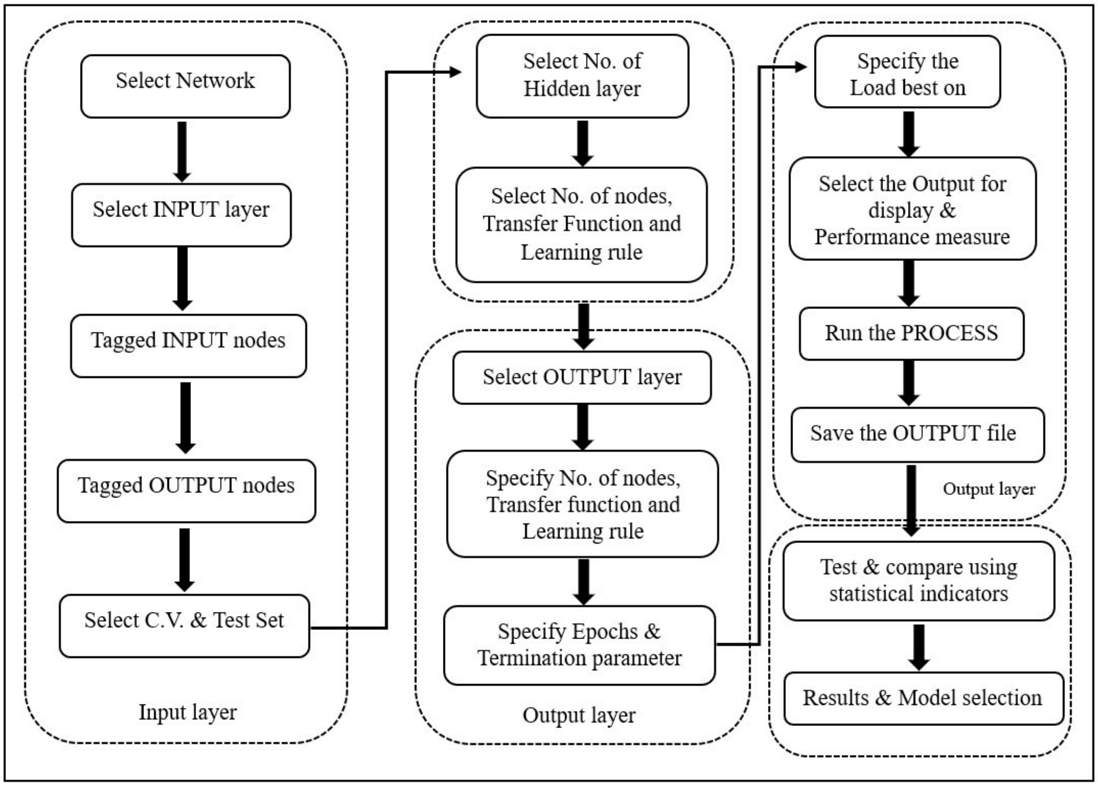


Fig: 4. Execution of Artificial neural network model

3.2.3.2 ANN architecture development

ANN architecture development involved creation of network topology and network training under various combinations of nodes in hidden layers, number of hidden layers, training cycles and parameters of training function. The performance of each combination was evaluated on the basis of statistical indicators. Table 3 represents the parameters used to develop the artificial neural network model.

Table 3: Parameters for development of Artificial Neural Network

Parameters	Selected type
Neural network model	Multilayer Perceptron neural network
Input layer	1
Input layer nodes	2
Hidden layer	1
Number of nodes in Hidden layers	2, 4, 6 (2n+2)
Output layer	1
Output layer node	1
training: validation ratio	70:30
Learning Algorithms	Levenberg Marquardt (LM) Conjugate Descent Gradient (CDG)
Transfer function	Sigmoid Axon
Epochs	200, 400, 600, 800, 1000
Model termination	Min. MSE (Mean Square Error)

3.2.3.3 Number of Nodes in Input layer

The number of nodes in input layer depended on number of parameters under study. In present study, two input parameters, namely, temperature difference in healthy and diseased area and grade assigned to the temperature difference were considered as nodes in input layers.

3.2.3.4 Number of Nodes in Hidden layer

The number of hidden nodes depends on several factors such as number of input and output nodes, number of training cycles, amount of noise in the targets, complexity of function or classification to be done, architecture, type of hidden unit activation function and training algorithm. The network with minimum nodes in hidden layer providing statistically accurate estimation was considered as optimal number of nodes in hidden layer required to train the model. A few hidden nodes resulted in a high training and generalization errors due to under-fitting, whereas too many hidden nodes resulted in low training error but higher estimation error due to over-fitting (Kumar, 2011). In present study, three different combinations of hidden layer nodes were performed 2, 4 and 6 to avoid the under-fitting and over-fitting problem while model training (Table 4).

Table 4: No. of Nodes in different layers of ANN model architecture

Sr. No.	Input layer nodes	Hidden layer nodes	Output layer nodes	ANN model
1	2	2	1	2-2-1
2	2	4	1	2-4-1
3	2	6	1	2-6-1

3.2.3.5 Number of Hidden layers

The number of hidden layers in the network architecture depends on the non-linearity of function to be learned. One hidden layer was enough to create the model architecture and train the model (Devi *et al*., 2020).

3.2.3.6 Transfer Function

The transfer function was utilized for limiting the amplitude of output to some finite values. Sigmoid Axon transfer function was most commonly used in the hidden layers of ANN network (Pujari *et al*., 2014).

3.2.3.7 Number of Node in Output layer

The number of nodes in the output layer depended on number of target variables. In present study, the output layer had a single node corresponding to the Grade in which the temperature difference belongs.

3.2.3.8 Number of Training Cycles

The number of training cycles was used to determine how many times the patterns should be learned to achieve minimized or stabilized training error. The number of training cycles depended on the algorithm employed in the training and non-linearity between input and output. The system error decreased with increasing number of training cycles during training but remained unchanged and in some cases increased during validation process. In present study, all ANN architectures were trained at 200, 400, 600, 800 and 1000 epochs and with goal of mean squared error of 0.01 during both training and validation.

3.2.3.9 Learning Algorithm

Learning algorithm optimized error function in order to modify the link weight. Two types of learning algorithms were employed in ANN architecture development for detection of early leaf spot of groundnut and anthracnose of mango.

3.2.3.9.1 Levenberg-Marquardt learning algorithm

The Levenberg-Marquardt (LM), a second order optimization technique was employed in thermal modelling using artificial neural network. LM algorithm was modification of Gauss-Newton method of approximation and presented in detail by Kisi (2007). The LM algorithm claimed to be superior to standard back-propagation in terms of fast convergence and thus needed lesser learning cycles (Zanetti *et al.*, 2007).

3.2.3.9.2 Conjugate gradient decent learning algorithm

The conjugate gradient descent method was applied to general unconstrained optimization problems by Fletcher and Reeves (1964). Unlike back-propagation, the conjugate gradient method processed in direction orthogonal to one in previous step. This prevented future steps from influencing the minimization achieved during current step. Fletcher and Reeves (1964) showed that any minimization method developed by conjugate gradient algorithm was quadratically convergent.

3.2.3.10 Performance Evaluation of ANN model

The performance evaluation of ANN models was done suing statistical analysis. The selected statistical indicators were as follows.

3.2.3.10.1 Root Mean Square Error (RMSE)

The root mean square error measured the average difference. Smaller the RMSE values, better the model performance. The optimum value for RMSE should be zero $\leq$ RMSE (Vazquez and Feyan, 2003). The RMSE represented by

$$\mathrm{RMSE} = \left[\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{P_i} - \mathrm{O_i})^2\right]^{0.5}$$

Where,

P_i = Predicted reference temperature difference grade for i^{th} observation

O_i = Targeted reference temperature difference grade for i^{th} observation

N = Number of Observations

3.2.3.10.2 Mean Bias Error (MBE)

The mean bias error was used to measure model bias. It provided general biasness but not average error that could be expected. The positive MBE value indicated overestimation and negative value indicated underestimation. The absolute value was indicator of model performance (Dehghani Sanij *et al.* 2004). With the optimal value of zero the biasness lies between -∞ to +∞ ($-\infty <$ bias $\leq +\infty$). The MBE given by

$$\mathrm{MBE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{P_i} - \mathrm{O_i})$$

3.2.3.10.3 Index of Agreement (I.A)

Index of agreement provides a relative measure of error allowing cross comparison of the model (Berengena and Gavilan, 2005). The model performance was good, when value of degree of index of agreement d $\geq$ 0.95 with optimal value one. Willmott (1982) expressed index of agreement as

$$d = 1 - \left[\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{P_i} - \mathrm{O_i})^2}{\sum_{\mathrm{i}=1}^{\mathrm{N}}((P'_{\mathrm{i}}| - |\mathrm{O'_i})^2)}\right]$$

Where,

N = number of Observations

$P'_i = P_i - \bar{O}$

$O'_i = O_i - \bar{O}$

3.2.3.10.4 Coefficient of correlation (r)

The correlation coefficient showed the degree of influence of independent variable on dependent variable. The value of correlation coefficient, if greater than 0.9 showed high level of correlation and 0.5 showed low correlation and between 0.5-0.9 represented moderate correlation. The correlation coefficient (r) measured strength of linear relationships between two variables. The correlation coefficient was calculated using equation

$$r = \frac{\sum PO - \frac{\sum P \sum O}{N}}{\sqrt{\left(\sum P^2 - \frac{(\sum P)^2}{N}\right)\left(\sum O^2 - \frac{(\sum O)^2}{N}\right)}}$$

RESULT AND DISCUSSION

CHAPTER IV

RESULT AND DISCUSSION

4.1 Early Leaf spot of groundnut disease detection model

4.1.1 Performance of CNN model developed for detection of early leaf spot of groundnut

The confusion matrix of early leaf spot disease detection model (Fig.5) proved that groundnut early leaf spot disease detection model performed well with 98% accuracy, 98% recall and 97% precision i.e., out of all diseased and healthy leaf samples tested 98% samples were classified correctly with 97% precision. F1 Score for groundnut disease detection model was 97% which considered both precision and recall and accounted for both false positive and false negative values. Teachable machine model preview developed for detection of early leaf spot of groundnut presented in PLATE IX.

In calculation of accuracy only true positive and true negative values were considered. The gap between training and testing accuracy in accuracy curve was very low (Fig.6), so there were no issues of model over fitting and therefore it is concluded that the model learned correctly.

Many researchers have used the similar method to develop different purpose models and have achieved similar results.

Aqil *et al.* (2021) reported that teachable machine model developed for detection of off-type plants based on male, female and contamination tassels types in maize was successful to classify maize tassel types with 80.7% accuracy. In addition, the classification error was 19.3%, a relatively lower value as compared to large testing datasets. Several mis-classification were found particularly at similar tassel shape. The off-type plant detection in maize model accuracy i.e., 80.7% was less than the accuracy of developed model for groundnut early leaf spot detection which was 98% with 2% classification error as compared to 19.3% in maize model. This is particularly due to large testing datasets which increased the mis-classification rate of the model.

Developed Teachable Machine model link for detection of early leaf spot of groundnut: “https://teachablemachine.withgoogle.com/models/X6J8sBZJk/”

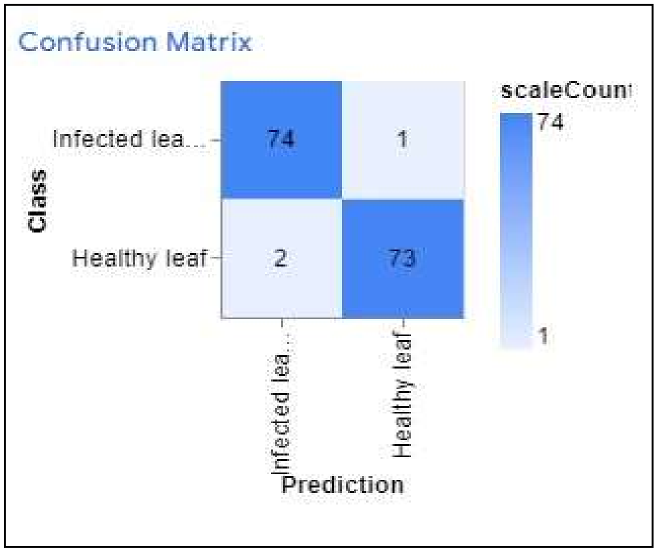
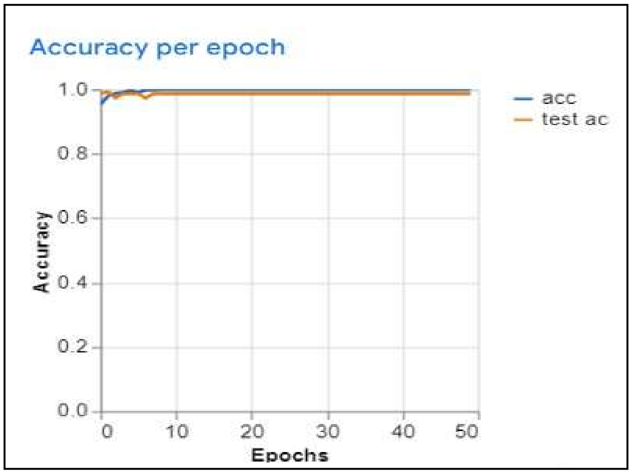
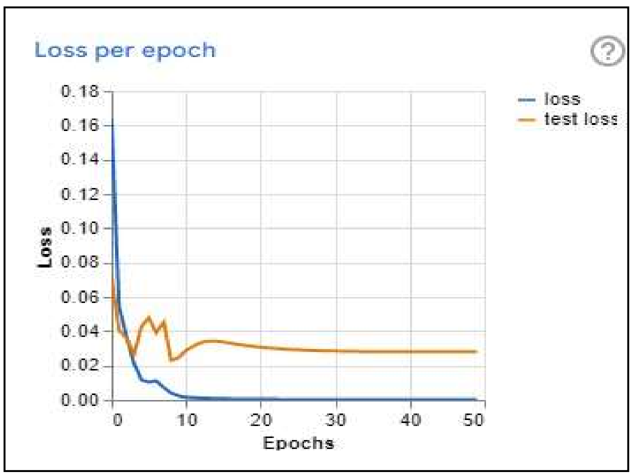


Fig: 5 Confusion matrix of teachable machine model developed for detection of groundnut leaf spot disease



Accuracy graph



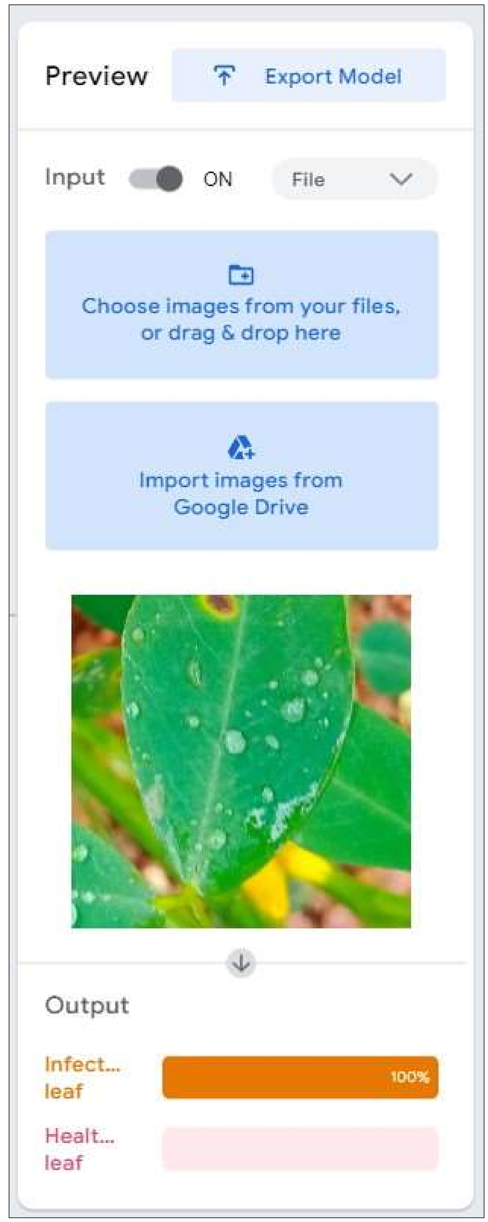
Loss graph

Fig: 6. Accuracy and loss of teachable machine model developed for detection of groundnut leaf spot disease

PLATE IX

Teachable machine model previews developed for detection of early leaf spot groundnut

Healthy groundnut leaf



Leaf spot infected groundnut leaf

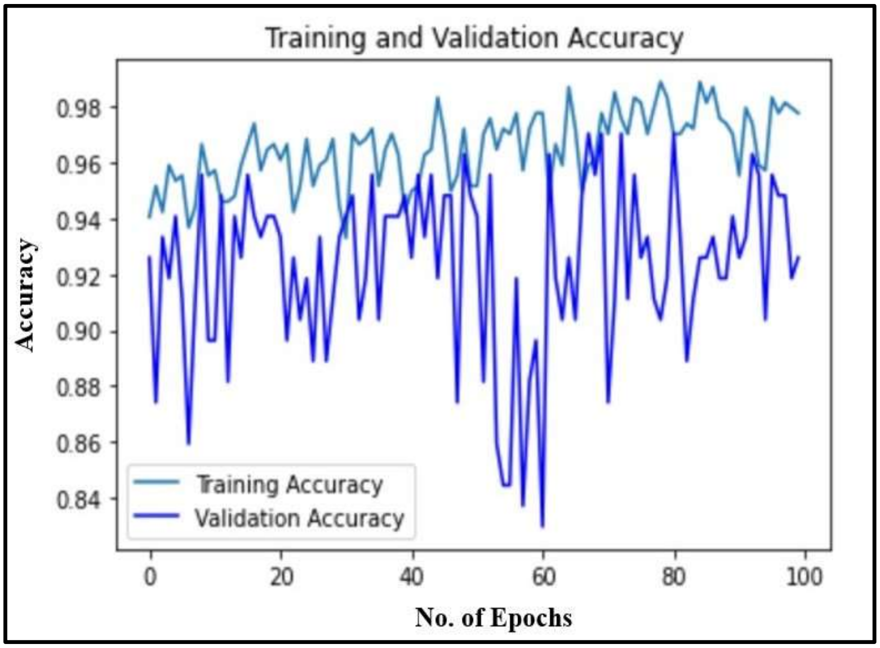
Performance of VGG-16 model developed for detection of early leaf spot of groundnut

The performance of developed VGG-16 model for groundnut early leaf spot detection was evaluated in terms of accuracy and loss function, which was measured during the training and validation of developed model (PLATE X). The Fig.7 shows the model performance in terms of accuracy and loss during training and validation phase. In the training phase the developed VGG-16 model achieved 97.77% accuracy. In validation step, the model accuracy was 92.59%.

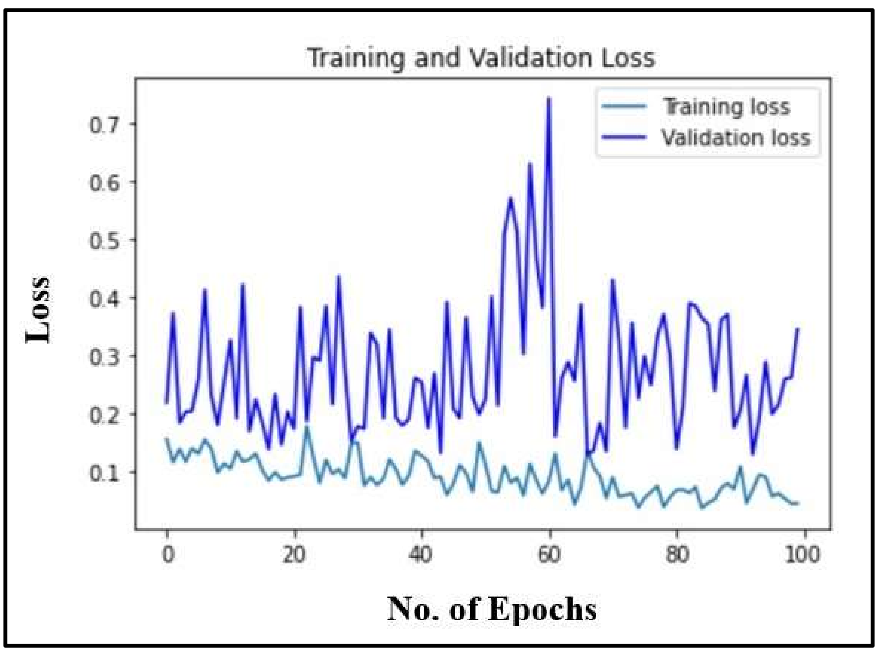
The VGG-16 model have been modified and tested by many researchers for detection of different plant diseases. Deeba and Amutha (2020) stated that five convolutional neural networks *viz.* LeNet, AlexNet, VGG-16, VGG-19 and ResNet when compared for detection of vegetable leaf diseases. The dataset contained images of potato (healthy, early blight and late blight), tomato (healthy, bacterial spot, early blight, late blight, leaf mold, septorial leaf spot, spite mites, target spot, mosaic and yellow leaf curl), corn (healthy, rust, northern leaf blight and leaf spot), brinjal (healthy, leaf spot and mosaic), and chilli (powdery mildew, bacterial leaf spot and cercospora leaf spot). The results showed that ResNet model achieved the highest detection accuracy of 96.6% as compared to VGG-16 model with 85.4% accuracy. In contrast to this, the developed model in this study for groundnut early leaf spot detection achieved 92.59% accuracy which was 8% higher than the previous findings.

Similarly, Patayon and Crisostomo (2022) tested eight pre-trained models including VGG-16, VGG-19, InceptionV3, MobileNet, DenseNet, Xception, InceptionResNetV2 and ResNet50.the results indicated that DenseNet model attained highest accuracy of 98% as compared to VGG-16 model with 92% accuracy. All the seven models except ResNet50 performed very good in detection of groundnut early leaf spot diseases with more than 90% accuracy.

In the same way, Sivasankaran *et al.* (2022) trained VGG-16 model based on deep convolutional neural networks with Adam optimizer performed with 99.82% accuracy for detection of five groundnut diseases *viz*. early leaf spot, late leaf spot, alternaria leaf spot, rust and bud necrosis. The results of VGG-16 models were compared with similar models *viz*. VGG-16 with RMSprop optimizer, AlexNet and Inception_V3 and proved to be outperforming all compared models in terms of accuracy. The accuracy of proposed model



Accuracy graph

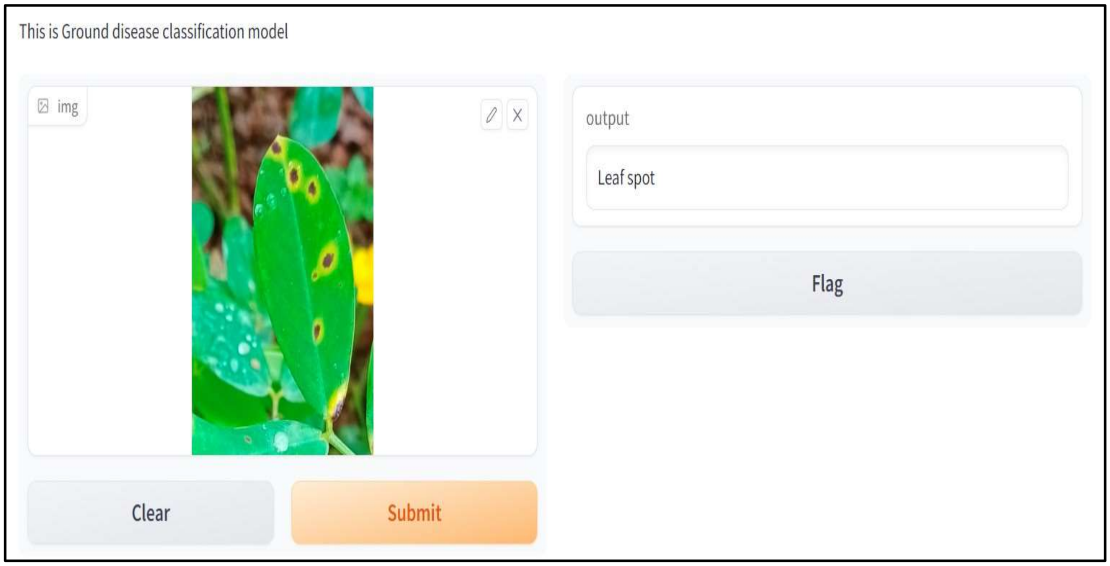


Loss graph

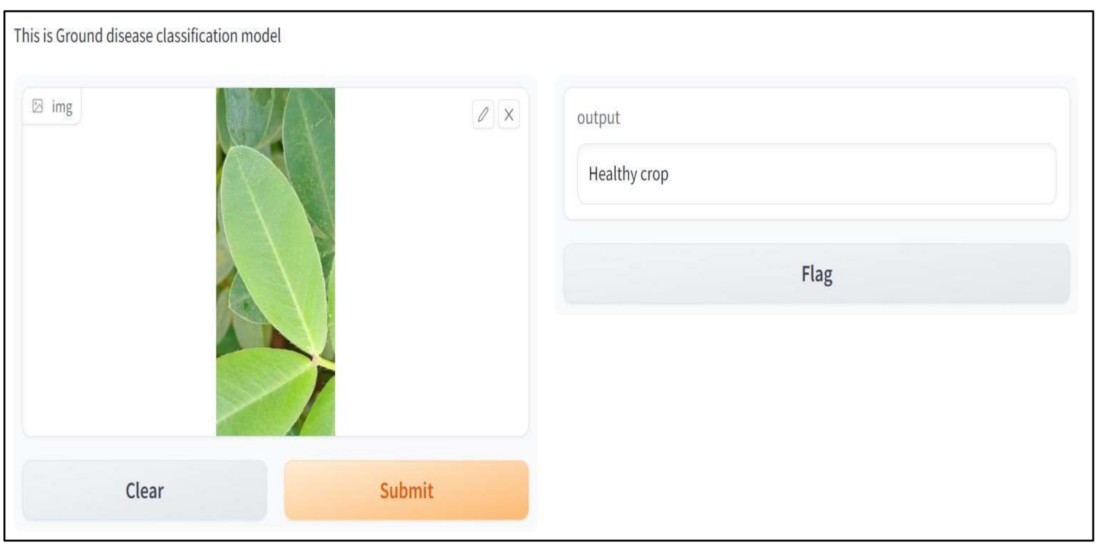
Fig: 7. Accuracy and Loss graph of VGG-16 groundnut leaf spot disease detection model

PLATE X

VGG-16 Gradio interface



Leaf spot of groundnut detection



Healthy groundnut leaf detection

was 6.23% higher than the VGG-16 model developed in this study for groundnut early leaf spot detection.

Kumar and Kumar (2023) also observed that VGG-16 model used for detection of tomato and potato diseases achieved 88.6% accuracy for four classes of tomato (mosaic, leaf curl, target spot and healthy leaves) with 94.6% detection accuracy for early blight, late blight and healthy potato leaves.

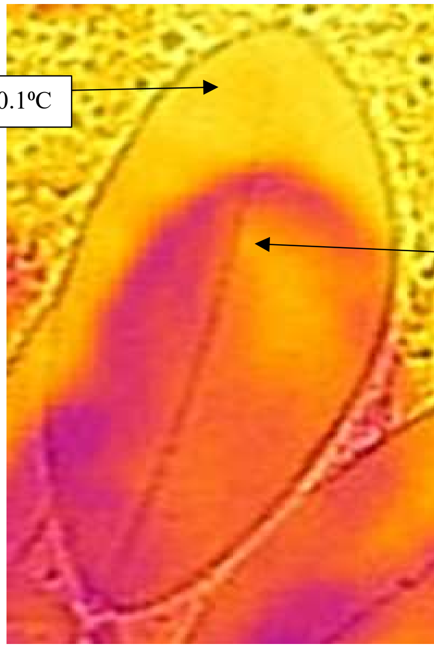
Likewise, Progressive groundnut convolutional neural network (PGCNN) model was developed by Anna A. and Shanthini A. (2023) to detect four diseases of groundnut. The model was trained and validated using self-collected dataset of early spot, late spot, rust and rosette diseases from various locations around the village located nearby Pudukkottai district, Tamil Nadu, India. The developed model was compared with AlexNet, VGG11, VGG13, VGG16 and VGG19. The PGCNN model excelled in terms of 99.39% training accuracy, 97.58% validation accuracy and continued with an overall accuracy of 97.58% outperforming the VGG16 model which achieved 69.49% training accuracy and 62.29% validation accuracy. As compared to these results, the developed VGG-16 model for detection of early leaf spot of groundnut higher accuracy of 92.59% outperforming other models.

4.1.2 Artificial Neural Network model development for detection of early leaf spot of groundnut

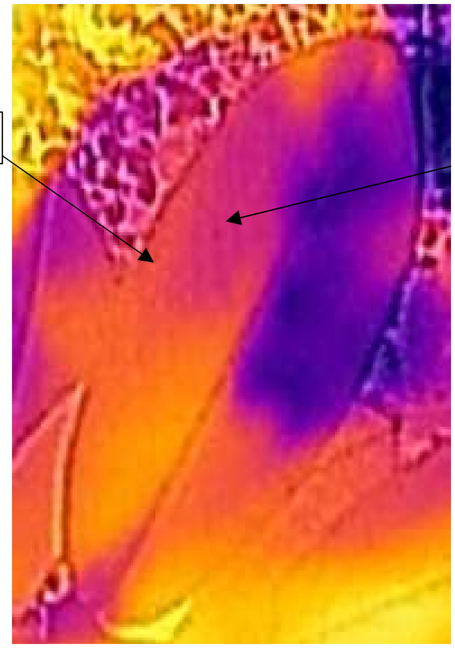
The ‘t’ test proved that there was no significant difference between temperature values of day one and the day prior to symptom development. The ‘t’ test was found significant for the temperature difference of the first day of symptom development, 3rd and 5th day of symptom development for early leaf spot of groundnut (PLATE XI & XII) (Table 5). The observations of these three days were classified into temperature difference grade (Table 1) based on minimum and maximum temperature difference and were used to develop artificial neural network model for detection of early leaf spot of groundnut.

PLATE XI

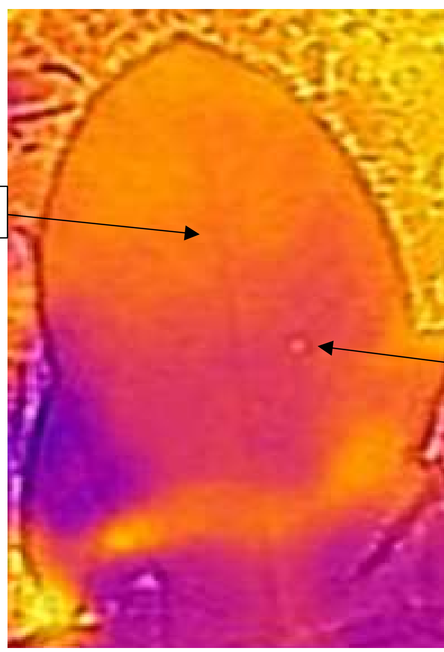
Temperature difference of Groundnut leaf spot dataset



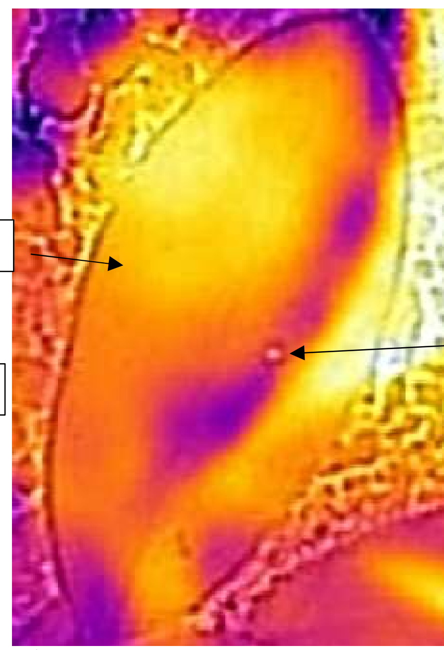
Day 1



1 Day before symptom appearance

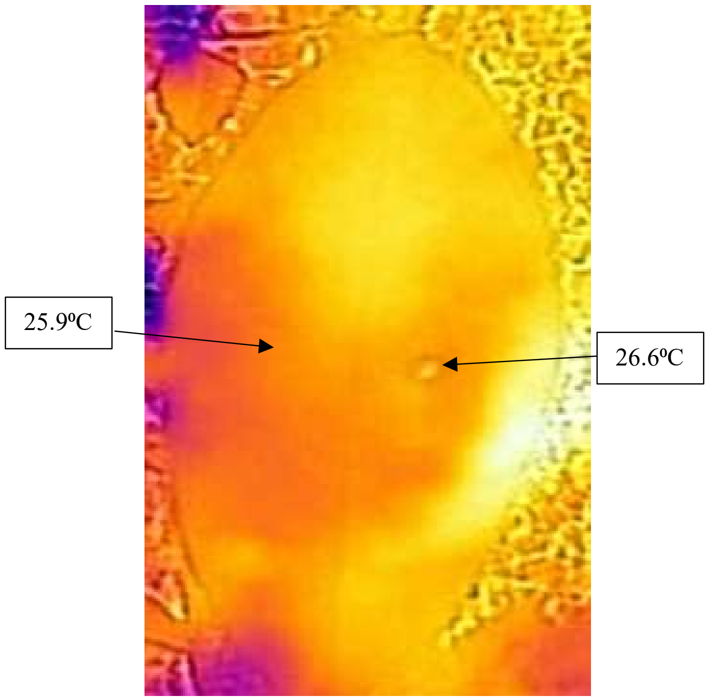


Day of symptom appearance



3rd day of symptom appearance

PLATE XII



5th day of symptom appearance

Table 5: Statistics of Groundnut early leaf spot thermal image data of healthy and diseased leaf area

	Healthy leaf	1 day before symptom appearance	Day of symptom appearance	3rd day of symptom appearance	5th day of symptom appearance
Days	1	6	7	9	11
Min. temp (^{0} c)	0.10	0.10	0.10	0.10	0.60
Max. temp (^{0} c)	0.80	0.80	0.50	0.60	2.50
Avg. temp (^{0} c)	0.24	0.28	0.24	0.29	1.06
St. Dev.	0.16	0.17	0.13	0.12	0.47
'p' value	Non-significant	Non-significant	Extremely significant	Extremely significant	Extremely significant

The ANN architecture 2-2-1 with LM algorithm trained with 1000 epochs performed well with r= 0.96, I.A.= 0.97, MBE= -0.0080 and RMSE= 0.05 for training and r= 0.97, I.A.=0.97, MBE= -0.0080 and RMSE=0.04 for cross validation sets. The correlation coefficient was 0.97 for both training and cross validation sets with close agreement between the training and cross validation datasets. The MBE (MBE= -0.0090 & -0.0080) and RMSE (RMSE = 0.05 and 0.04) showed very less error in prediction of early leaf spot of groundnut for training and cross validation sets.

The ANN architecture 2-4-1 with LM algorithm trained at 1000 epochs performed with r= 0.95, I.A.= 0.96, MBE= -0.0118 and RMSE = 0.06 for training set and r= 0.97, I.A.= 0.98, MBE= - 0.0097 and RMSE= 0.03 for cross validation set. The correlation coefficient for training and cross validation set was more than 0.95. The I.A. indicated close agreement between training (0.96) and cross validation (0.98) sets. The MBE of -0.0118; -0.0097 and RMSE of 0.06 and 0.03 indicated very less error in prediction of early leaf spot of groundnut for both training and cross validation sets.

The ANN architecture 2-6-1, with LM algorithm was developed for five different epochs i.e., 200, 400, 600, 800 and 1000. The statistical indicators of 2-6-1 architecture with LM algorithm trained at 1000 epochs were r= 0.94, I.A.= 0.96, MBE= -0.0073 and RMSE =0.06 for training set and r = 0.96, I.A.=0.97, MBE= -0.0086, RMSE= 0.04 for cross validation set. These statistical indicators showed that 2-6-1 architecture with LM algorithm trained at 1000 epochs performed better than other developed models with different range of epochs.

The ANN architecture development for groundnut early leaf spot detection also tested for Conjugate descent gradient learning algorithm. The different architectures i.e., 2-2-1, 2-4-1 and 2-6-1 at 200, 400, 600, 800 and 1000 epochs were tested as presented in table no.6.

The 2-2-1 ANN architecture with CDG learning algorithm was trained at different epochs and tested with statistical indicators. The statistical indicators showed that, the ANN architecture 2-2-1 with CDG algorithm trained at 1000 epochs performed well in terms of r= 0.93, IA.=0.96, MBE= - 0.0158, RMSE=0.06 for training set and r= 0.96, I.A.=0.97, MBE= -.0164 and RMSE= 0.04 for cross validation set. The correlation coefficient was more than 0.9 for both training (0.93) and cross validation (0.96) with high Index of Agreement (I.A.) between both training (0.96) and cross validation (0.97) sets. The MBE and RMSE indicated less error in prediction of early leaf spot of groundnut for both training (MBE= -0.0158 and RMSE= 0.06) and cross validation (MBE= -0.0164 and RMSE= 0.04) sets.

The 2-4-1 architectures with CDG learning algorithm trained at different epochs (i.e., 200, 400, 600, 800 and 100) the 2-4-1 architecture with CDG trained at 1000 epochs was suitable based on statistical indicators. The training dataset showed correlation coefficient of 0.76, I.A.= 0.79, MBE= - 0.0178 and RMSE= 0.11. The correlation coefficient for cross validation set was 0.79 with 0.89 I.A., MBE= -0.00057 and RMSE= 0.08. The statistical indicators for both training and cross validation showed close agreement with less error in prediction of early leaf spot of groundnut with field observations.

The ANN 2-6-1 architecture with CDG algorithm at 200 epochs performed well with r= 0.79, I.A.= 0.80, MBE= -0.0117 and RMSE= 0.11 for training set and r=0.81, I.A.=0.84, MBE= 0.0013 and RMSE=0.08 for cross validation set. The statistical indicators for 2-6-1 architecture with CDG trained at 200 epochs were at par with the same architecture at 400 epochs. The correlation coefficient and I.A. values of 2-6-1 architecture with CDG algorithm trained at 400 epochs were increased to 0.83 and 0.82 for training and decreased to 0.31 and 0.54 for cross validation sets. Increase in error was observed for both training and cross validation sets for 2-6-1 architecture with CDG algorithm trained at 400 epochs. Table no. 6 represents the statistical analysis results of all trained ANN architecture combinations developed for detection of early leaf spot of groundnut.

Table 6: Statistical analysis of ANN architecture 2-2-1, 2-4-1 and 2-6-1 developed at 200, 400, 600, 800 and 1000 epochs with Levenberg Marquardt and Conjugate descent gradient algorithms developed for detection of early leaf spot of groundnut

Algorithm	ANN arch	Node	Epoch	Statistical Indicators							
				Training				C.V.			
				r	I.A.	MBE	RMSE	r	I.A.	MBE	RMSE
LM	2-2-1	2	200	0.81	0.81	-0.0156	0.11	0.85	0.85	-0.0045	0.08
			400	0.84	0.82	-0.0267	0.10	0.91	0.86	-0.0216	0.08
			600	0.82	0.81	-0.0324	0.11	0.86	0.82	-0.0321	0.09
			800	0.80	0.81	-0.0167	0.11	0.84	0.85	-0.0057	0.08
			1000	0.96	0.97	-0.0090	0.05	0.97	0.97	-0.0080	0.04
	2-4-1	4	200	0.80	0.80	-0.0163	0.11	0.84	0.85	-0.0050	0.08
			400	0.83	0.82	-0.0222	0.10	0.89	0.87	-0.0146	0.08
			600	0.83	0.82	-0.0227	0.10	0.90	0.87	-0.0155	0.08
			800	0.82	0.82	-0.0197	0.10	0.87	0.86	-0.0105	0.08
			1000	0.95	0.96	-0.0118	0.06	0.97	0.98	-0.0097	0.03
	2-6-1	6	200	0.79	0.80	-0.0117	0.11	0.81	0.84	-0.0012	0.08
			400	0.79	0.80	-0.0120	0.11	0.82	0.84	-0.0007	0.08
			600	0.82	0.81	-0.0202	0.10	0.87	0.86	-0.0109	0.08

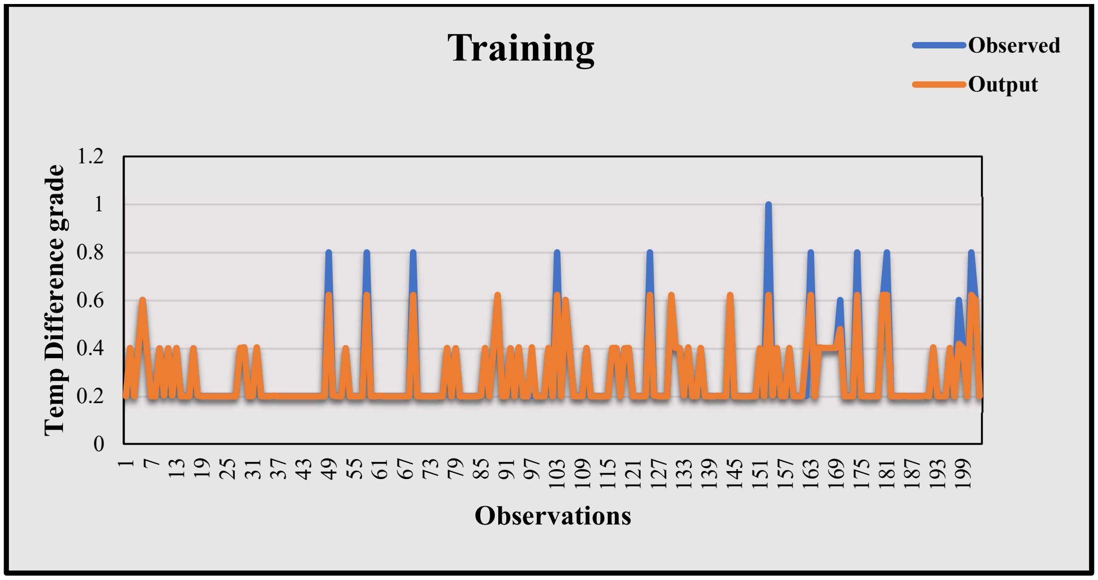
			800	0.83	0.82	-0.0225	0.10	0.89	0.87	-0.0148	0.08
			1000	0.94	0.96	-0.0073	0.06	0.96	0.97	-0.0086	0.04
CDG	2-2-1	2	200	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
			400	0.93	0.35	-0.0578	0.14	0.95	0.31	-0.0524	0.14
			600	0.25	0.35	-0.0585	0.14	0.56	0.31	-0.0530	0.14
			800	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
			1000	0.87	0.91	-0.0366	0.07	0.88	0.85	-0.0497	0.09
	2-4-1	4	200	-0.09	0.34	-0.0590	0.14	0.31	0.31	-0.0531	0.14
			400	-0.23	0.34	-0.0592	0.14	0.14	0.31	-0.0532	0.14
			600	0.20	0.35	-0.0587	0.14	0.96	0.32	-0.0520	0.14
			800	-0.87	0.34	-0.0590	0.14	-0.91	0.30	-0.0535	0.14
			1000	0.89	0.92	-0.0205	0.06	0.90	0.86	-0.0326	0.08
	2-6-1	6	200	-0.10	0.35	-0.0602	0.14	-0.91	0.30	-0.0534	0.14
			400	0.89	0.35	-0.0584	0.14	0.90	0.31	-0.0529	0.14
			600	-0.95	0.34	-0.0588	0.14	-0.96	0.31	-0.0532	0.14
			800	-0.89	0.34	-0.0588	0.14	-0.90	0.31	-0.0532	0.14
			1000	0.89	0.92	-0.0201	0.06	0.90	0.86	-0.0324	0.08

The perusal of table no. 7 reveals that the three ANN architecture 2-2-1, 2-4-1 and 2-6-1 with two learning algorithms CDG and LM performed very well. The results also claimed that the ANN architecture 2-2-1 with LM (Fig.8) and CDG (Fig.9) learning algorithms trained with 1000 epochs was found best for prediction of early leaf spot of groundnut with minimum error and high degree of association.

Table 7: Best ANN architecture observed from the performed ANN architecture combinations for detection of early leaf spot of groundnut

Algorithm	ANN arch	Epoch	Statistical Indicators							
			Training				C.V.			
			r	I.A.	MBE	RMSE	r	I.A.	MBE	RMSE
LM	2-2-1	1000	0.96	0.97	-0.0090	0.05	0.97	0.97	-0.0080	0.04
	2-4-1	1000	0.95	0.96	-0.0118	0.06	0.97	0.98	-0.0097	0.03
	2-6-1	1000	0.94	0.96	-0.0073	0.06	0.96	0.97	-0.0086	0.04
CDG	2-2-1	1000	0.93	0.96	-0.0158	0.06	0.96	0.97	-0.0164	0.04
	2-4-1	1000	0.76	0.79	-0.0178	0.11	0.79	0.84	-0.0057	0.08
	2-6-1	200	0.79	0.80	-0.0117	0.11	0.81	0.84	0.0013	0.08

The present findings are in agreement with Paul and Munkvold (2004), who used regression and Back propagation neural network (BPNN) modelling method for prediction of gray leaf spot of maize. The duration of favourable temperatures, relative humidity, and surface wetness were calculated for four periods during the growing season relative to the R1 (silking) plant growth stage. These periods were (i) 45 days before R1 until 15 days after R1, (ii) 15 days before until 15 days after R1, (iii) 30 days before R1 until R1, and (iv) 45 days before R1 until 15 days before R1. BPNN model A2, with seven input variables LON (longitude), GLSR (gray leaf spot resistance rating based on a 1-to-9 scale), SR (maize residue on soil surface), PD (planting date), CDT4 (cumulative daily temperature between 22 and 30^{o}C for period 4), AVNT2 (mean nightly temperature for period 2), and NRH904 (cumulative hours of nightly relative humidity $\geq$90% for period 4), was the most superior. Model performance was evaluated based on correlation co-efficient of determination (r) and mean square error (MSE) for the validation data set. The best model A2 had ‘r’ ranging from 0.70 to 0.75 which represented moderately strong



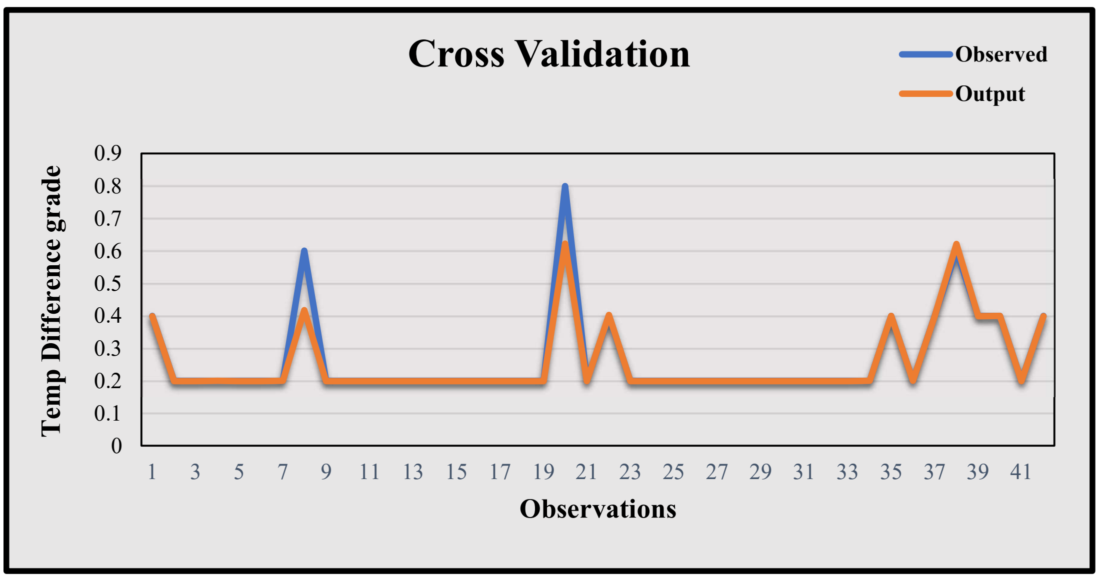
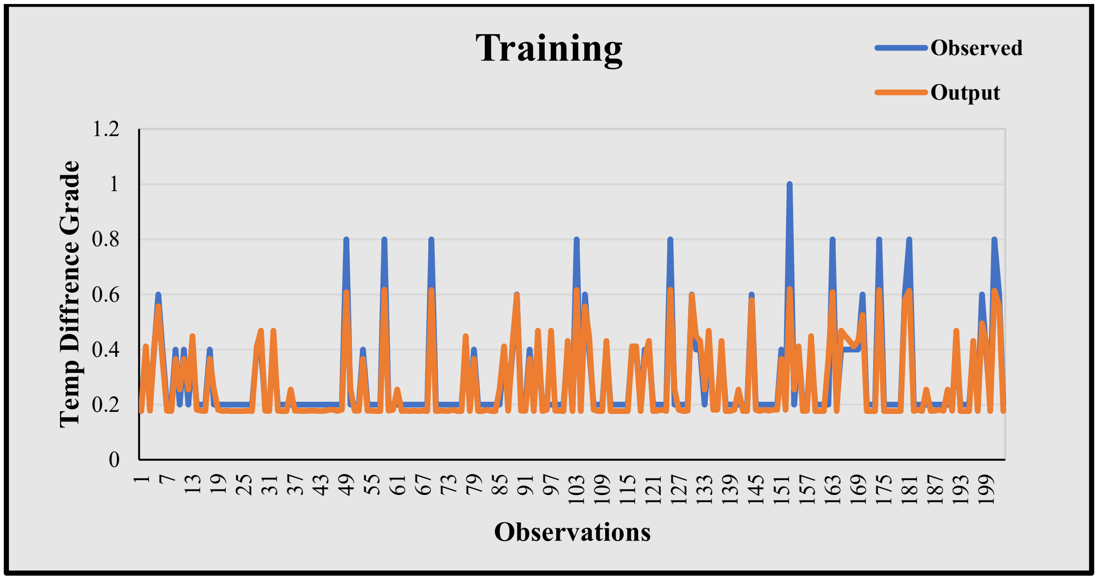


Fig: 8. Training and cross validation graphs of ANN architecture 2-2-1 with Levenberg Marquardt algorithm developed for detection of leaf spot of groundnut at 1000 epochs



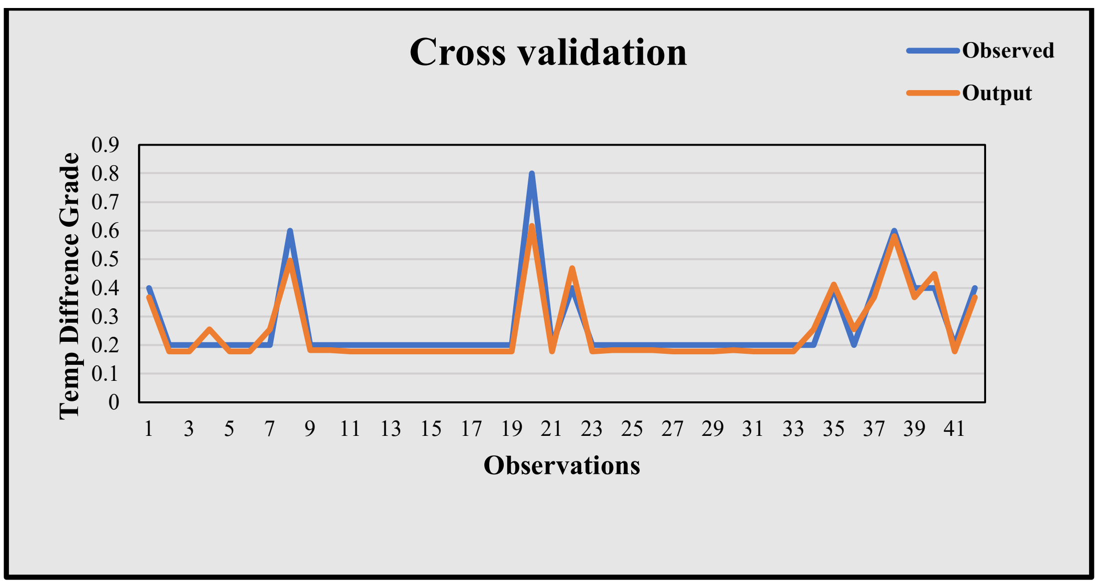


Fig: 9. Training and cross validation graphs of ANN architecture 2-2-1 with Conjugate Descent gradient algorithm developed for detection of leaf spot of groundnut at 1000

positive relationship between the study variables as compared to correlation coefficient of 0.96 of LM model and 0.93 of CDG model indicating very strong positive relationship between the study variables of groundnut early leaf spot disease detect model.

Similarly, Ramakrishnan and Sahaya (2015), used a multilayer feed forward back propagation neural network for detection and classification of groundnut leaf diseases. The dataset contained 100 images of early leaf spot infected leaves, 100 images of late leaf spot infected leaves and 100 images of leaf blight infected leaves. 80% images were used for training, 10% were used for testing and remaining 10% for validation purpose. Co-occurrence matrices were used to extract texture features of dataset images. The model was trained using the neural network tool box and diseased leaf images as input. The Matlab software tool box was used to implement the developed back propagation algorithm which classified groundnut leaf diseases with 97.4l % accuracy.

Devi *et al.* (2020) combined Harris corner detector, HOG (Histogram on Oriented Gradient) and KNN classifier in a method called H2K for accurate detection and classification of groundnut leaf diseases. The created image dataset contained 129 images of groundnut leaves both healthy and diseased with early leaf spot, late leaf spot, rust, bud necrosis and leaf blight symptoms. The results showed that the proposed model performed well in detecting groundnut diseases with improved detection accuracy of 97.4% for early leaf spot, 100% for late leaf spot, 100% for rust, 90.9% for bud necrosis, 100% for blight and 100% for healthy leaves respectively.

Gowrishankar and Prabha (2020) with Artificial neural network classifier applied threshold-based color segmentation technique for image segmentation to analyze early leaf spot of groundnut. The ANN was trained using threshold values of collected images. At the 37th epoch ANN 20 reached the goal with the best performance of training iterations. The classified images by the ANN indicated 51.98% of leaves were affected by early leaf spot disease with 98% accuracy.

Further, they have also combined Artificial neural network (ANN) and GLCM for groundnut leaf disease diagnosis. Ground leave images infected with early and late leaf spot were selected for the study. In the proposed work, gray level co-occurrence matrix (GLCM) was used for feature extraction during image processing. After feature extraction using GLCM, the images were classified using ANN and SVM into diseased and healthy.

The ANN model was implemented on ‘Matlab’ software. The accuracy of proposed ANN system was 98%.

Vaishnnavel *et al.* (2019) classified four economically important groundnut diseases (early leaf spot, late leaf spot, rust and blight) using KNN classifier. Color co-occurrence method was used for feature extraction from the input images. The K- Nearest Neighbor (KNN) classifier achieved 95% classification accuracy for groundnut diseases.

The results of the ANN multi-layer perceptron model developed in this research work proved that the model is very efficient and precise in detection of groundnut early leaf spot disease based on thermal imaging as compared to previously developed ANN models.

4.2 Mango anthracnose disease detection model

4.2.1 Performance of CNN model developed for detection of anthracnose of mango

Based on confusion matrix (Fig.10), accuracy of mango anthracnose disease detection model was 96% with 96% recall. The Precision and F1 score values were also in range with 97% precision and 96% F1 Score. The accuracy and loss curve of mango anthracnose disease detection model is presented in Fig.11. The developed mango anthracnose detection model preview shown in PLATE XIII.

These results were similar with the findings of Patil and Nagpure (2022). They have achieved 98% detection accuracy for northern leaf blight infected leaves and 99% detection accuracy for healthy leaves of corn with corn disease detection model developed using teachable machine. The model was trained using 1090 healthy leaf images and 1762 leaf blight infected leaf images of corn.

Developed Teachable Machine model link for detection of anthracnose of mango:

“https://teachablemachine.withgoogle.com/models/MxX9zWd7C/”

4.2.2. Artificial Neural Network model development for detection of anthracnose of mango

Results of ‘t’ test for mango anthracnose thermal images (Table 8) indicated that there was no significant difference between temperature values of first day of image collection and the day prior to symptom development. Therefore, the temperature difference values from first day of symptom development, 3rd and 5th day of symptom

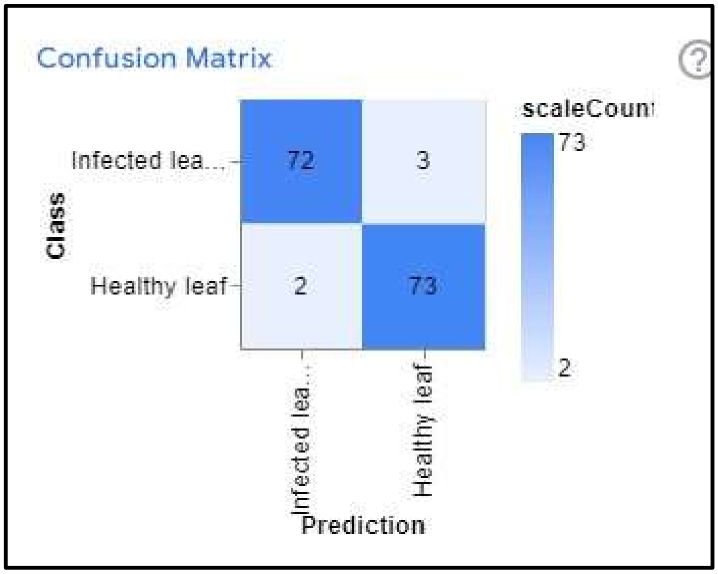
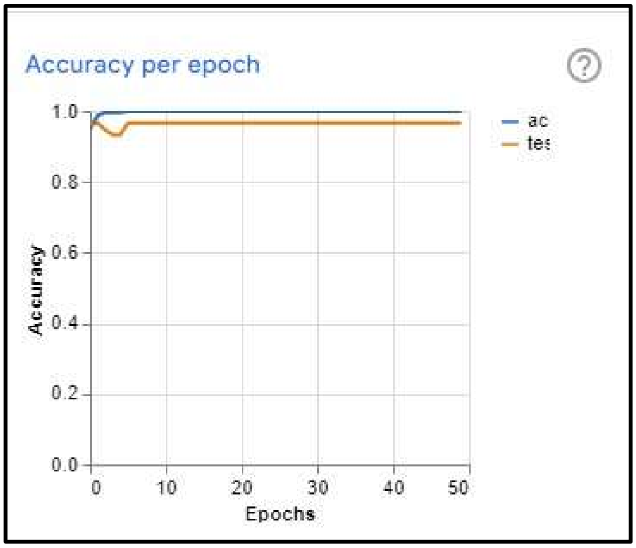
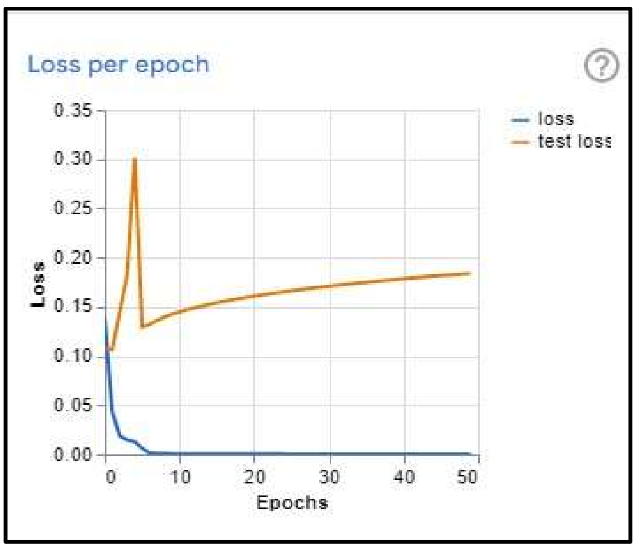


Fig: 10. Confusion matrix of teachable machine model developed for detection of mango anthracnose disease



Accuracy graph



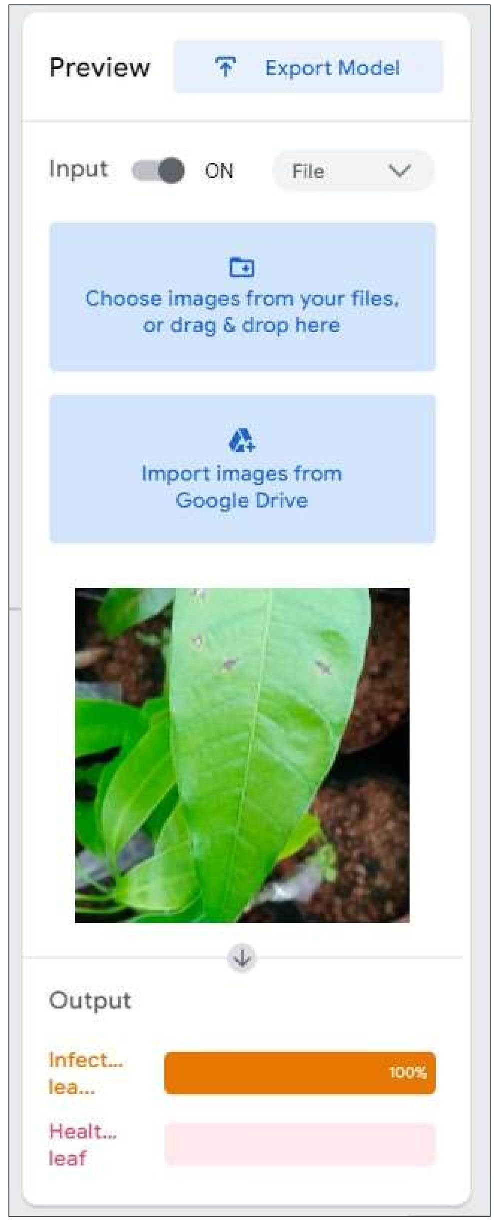
Loss graph

Fig: 11. Accuracy and loss of teachable machine model developed for detection of mango anthracnose disease

PLATE XIII

Teachable machine model previews developed for detection of anthracnose of mango

Healthy mango leaf



Anthracnose infected mango leaf

development (PLATE XIV & PLATE XV) were classified in given temperature grades (Table 2) and were used to develop the ANN model for detection of mango anthracnose disease.

Table 8: Statistics of Mango anthracnose thermal image data of healthy and diseased leaf area

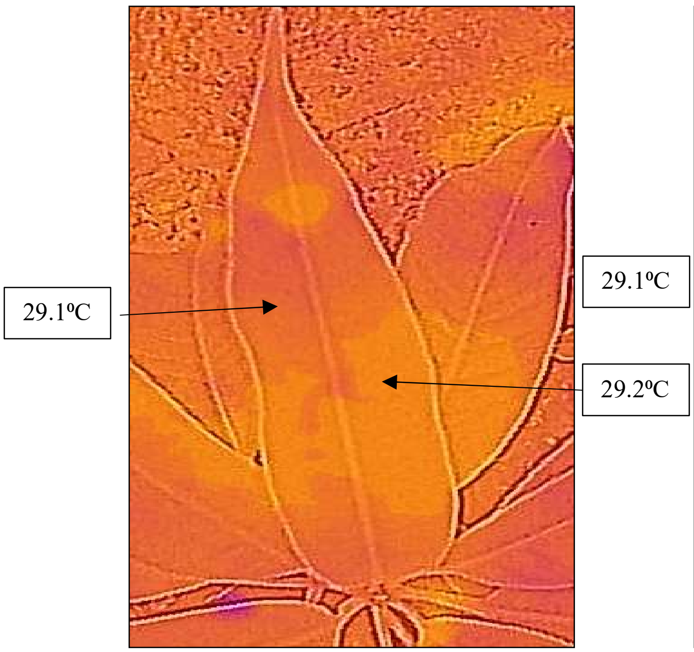
	Healthy leaf	1 day before symptom appearance	Day of symptom appearance	3rd day of symptom appearance	5th day of symptom appearance
Days	1	5	6	8	10
Min. temp (^{0} c)	0.10	0.10	0.10	0.10	0.30
Max. temp (^{0} c)	0.50	0.40	0.60	0.90	1.80
Average	0.17	0.17	0.21	0.19	0.62
St. Dev.	0.08	0.08	0.10	0.10	0.30
'p' value	Non-significant	Non-significant	Extremely significant	Extremely significant	Extremely significant

For identification of mango anthracnose disease images of infected mango leaves were collected using FlirOne thermal camera and classified as per the temperature difference grade. The different ANN architectures such as 2-2-1, 2-4-1 and 2-6-1 were analyzed with two learning algorithms *viz.* Levenberg Marquardt and Conjugate Descent Gradient. The developed models were tested using training and testing sets with the help of statistical indicators such as Correlation of co-efficient (r), Index of Agreement (I.A.), Mean Bias Error (MBE) and Root Mean Square Error (RMSE).

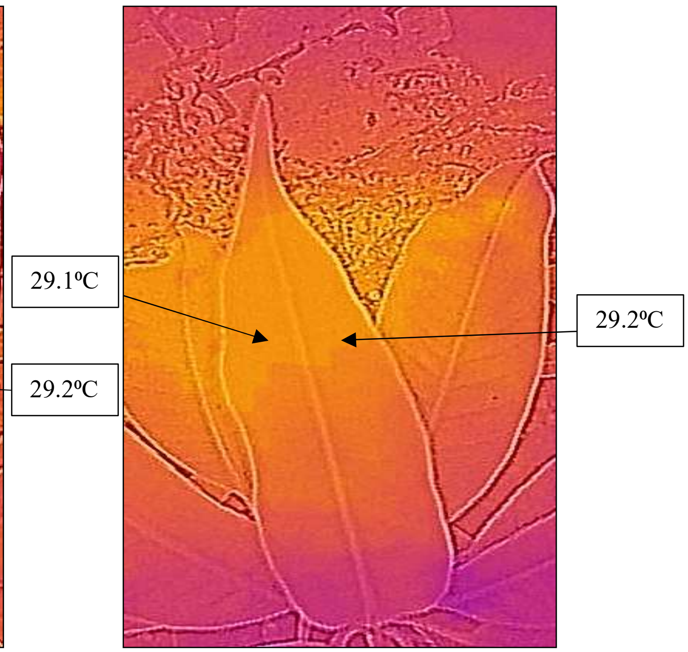
The ANN architecture 2-2-1 with LM algorithm trained with 1000 epochs (Table 9) performed well in terms of r=0.88, I.A.=0.92, MBE= -0.0210, RMSE= 0.07 for training set and r=0.89, I.A.=0.86, MBE= -0.0329, RMSE= 0.08 for cross validation set. The statistical indicators values indicated that 2-2-1 architecture with LM at 1000 epochs found better in terms of correlation coefficient of 0.88 and 0.89 for training and cross validation set with close agreement between both training (0.92) and cross validation (0.86) sets. The MBE (-0.0210 and -0.0329) and RMSE (0.07 and 0.08) for 2-2-1 with LM trained at 1000 epochs resulted in less error for testing and cross validation sets.

PLATE XIV

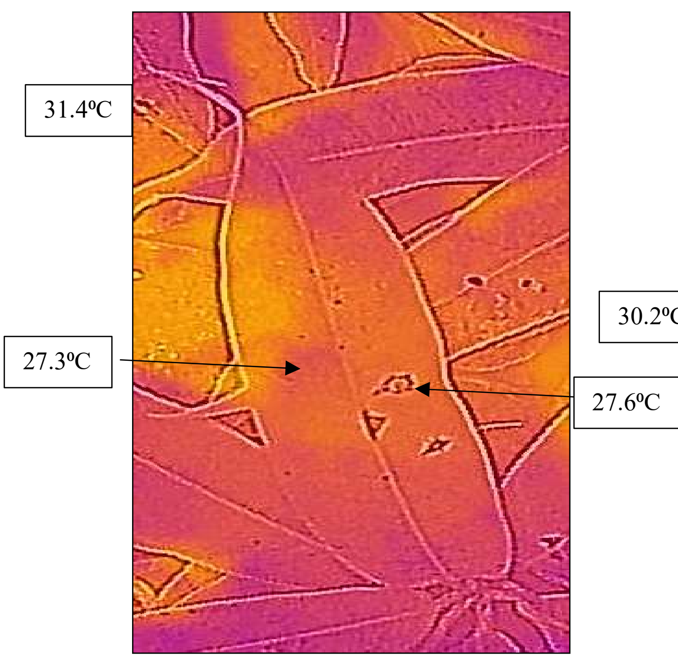
Temperature difference of Mango anthracnose dataset



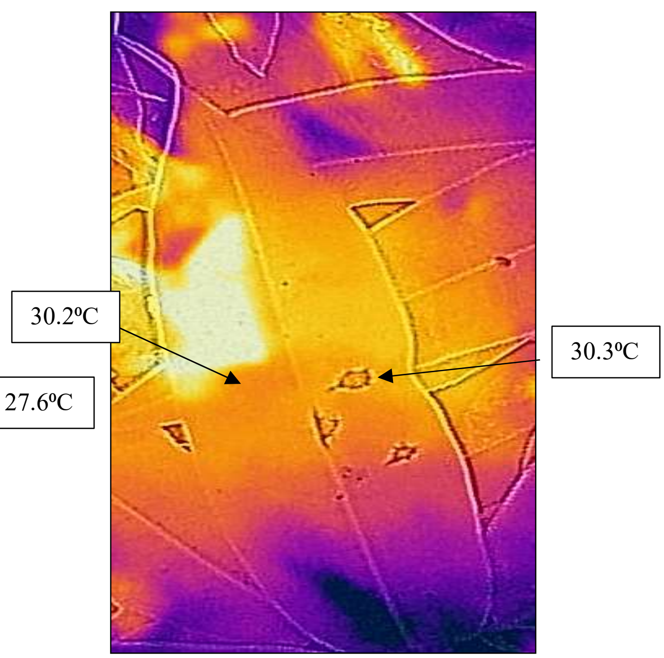
Day 1



1 Day before symptom appearance

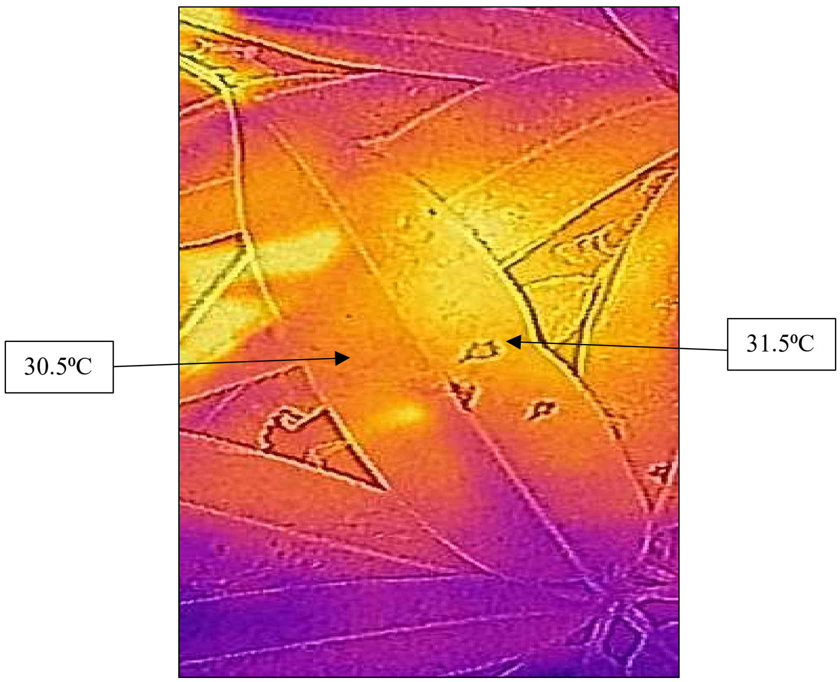


Day of symptom appearance



3rd day of symptom appearance

PLATE XV



5th day of symptom appearance

The statistical indicators for 2-4-1 ANN architecture with LM algorithm showed that 2-4-1 architecture trained with 1000 epochs performed well with r=0.89, I.A.=0.91, MBE=-0.0205, and RMSE=0.06 for training dataset and r=0.90, I.A.=0.86, MBE= -0.0326 and RMSE=0.08 for cross validation sets. The correlation coefficient was nearly equal to 0.9 for both training and cross validation with close agreement between both sets. The mean bias error near to zero (MBE= -0.0205 and -0.0326) with less error (RMSE= 0.06 and 0.08) was found for both training and testing sets.

The ANN architecture 2-6-1 with LM algorithm analyzed with 6 nodes in hidden layer was trained in similar manner with 200, 400, 600, 800 and 1000 epochs. The correlation coefficient for 2-6-1 architecture with LM at 1000 epochs was nearly equal to 1.0 for training (0.89) and cross validation (0.90). the I.A. showed close agreement between training (0.92) and cross validation (0.86) sets. Mean Bias Error was near to zero (i.e., -0.0201 and -0.0324) for both training and cross validation sets. The RMSE of 0.06 and 0.08 resulted in less error for prediction for training and cross validation sets. Like 2-2-1 and 2-4-1, 2-6-1 trained at 1000 epochs with LM algorithm proved suitable based on statistical indicators.

The ANN architecture for detection of anthracnose of mango was also developed using Conjugate Descent Gradient (CDG) learning algorithm for 200, 400, 600, 800 and 1000 epochs with 2, 4 and 6 nodes in hidden layer and tested with statistical indicators for selection of most suitable model with optimum data.

The statistical indicators showed that ANN architecture 2-2-1 with conjugate descent gradient learning algorithm trained with 200 epochs found suitable in terms of r=0.88, I.A.=0.92, MBE= -0.0210 and RMSE= 0.07 for training set. The cross validation set also showed similar trend i.e., r=0.89, I.A.=0.86, MBE=-0.0330 and RMSE= 0.08.

The value of correlation co-efficient was nearly equal to 0.9 for both training and cross validation sets. The Index of agreement (I.A.) showed close agreement (i.e., 0.92 and 0.86) between training and cross validation dataset. The mean bias error (MBE) represents the underestimation of predicted values with observed values. The MBE values near to zero also pointed out less error in prediction of anthracnose infection with observed values for both training and cross validation set (MBE= -0.021 and -0.033). The Root mean square error (RMSE) for 2-2-1 architecture with Conjugate descent gradient learning algorithm trained at 200 epochs resulted in less error of (0.07 and 0.08) for training and cross validation set.

The ANN architecture 2-2-1(CDG) was also trained with increment of 200 up to 1000 epochs resulted in decrease in correlation coefficient and Index of Agreement (less than one) and increase in MBE and RMSE (more than zero) as shown in (Table 9). Therefore, ANN architecture 2-2-1 (CDG) with 200 epochs was found suitable for predictions of mango anthracnose disease.

The ANN architecture having 4 nodes i.e., 2-4-1 with Conjugate descent gradient algorithm was trained at 200, 400, 600, 800 and 1000 epochs to find the suitable architecture. The statistical indicators indicated that 2-4-1 architecture with Conjugate descent gradient algorithm trained at 1000 epochs performed better in terms of r=0.89, I.A.= 0.92, MBE= -0.0205, RMSE= 0.06 for training dataset and r=0.90, I.A.=0.86, MBE= -0.0326, RMSE= 0.08 for cross validation dataset. The index of agreement showed close agreement between training (0.92) and cross validation (0.086) datasets with 0.89 and 0.90 correlation coefficient for training and cross validation sets. The MBE close to zero was observed showing less prediction error with observed values for both training (-0.0205) and cross validation (-0.0326) datasets. RMSE for training and cross validation sets was 0.06 and 0.08 much less than models trained at 200, 400, 600 and 800 epochs.

The ANN architecture 2-4-1 was trained up to 1000 epochs after which with increase in epochs resulted in decrease in correlation co-efficient and index of agreement, and increase in error (MBE & RMSE). Hence, ANN architecture 2-4-1 (CDG) trained at 1000 epochs was found suitable among other developed models.

The ANN architecture (2-6-1) with Conjugate descent gradient algorithm was trained with 6 nodes in hidden layer with 200, 400, 600, 800 and 1000 epochs. The 2-6-1 architecture with CDG with 1000 epochs performed well in terms of statistical indicators such as r=0.89, I.A.=0.92, MBE= -0.0201, RMSE=0.06 for training set and r=0.90, I.A.=0.86, MBE= -0.0324, RMSE= 0.08 for cross validation sets. The value of correlation coefficient was nearly equal to 0.9 for both training (0.89) and cross validation (0.90) with close agreement (I.A.) of 0.92 and 0.86 for both sets. The MBE and RMSE for training and cross validation sets showed less error in prediction of Anthracnose with observed values. The developed model (2-6-1, CDG, 1000 epochs) had less errors in prediction when compared with observed set. Similar results also given in table no. 9.

Table 9: Statistical analysis of ANN architecture 2-2-1, 2-4-1 and 2-6-1 developed at 200, 400, 600, 800 and 1000 epochs with Levenberg Marquardt and Conjugate descent gradient algorithms developed for detection of anthracnose of mango

Algorithm	ANN arch	Node	Epoch	Statistical Indicators							
				Training				C.V.			
				r	I.A.	MBE	RMSE	r	I.A.	MBE	RMSE
LM	2-2-1	2	200	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
			400	0.93	0.35	-0.0578	0.14	0.95	0.31	-0.0524	0.14
			600	0.25	0.35	-0.0585	0.14	0.56	0.31	-0.0530	0.14
			800	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
			1000	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0329	0.08
	2-4-1	4	200	-0.09	0.34	-0.0590	0.14	0.31	0.31	-0.0532	0.14
			400	-0.23	0.34	-0.0592	0.14	0.40	0.31	-0.0532	0.14
			600	0.20	0.35	-0.0587	0.14	0.96	0.32	-0.0520	0.14
			800	-0.87	0.34	-0.0590	0.14	-0.91	0.30	-0.0535	0.14
			1000	0.89	0.92	-0.0205	0.06	0.90	0.86	-0.0326	0.08
	2-6-1	6	200	-0.10	0.35	-0.0602	0.14	-0.91	0.30	-0.0534	0.14
			400	0.89	0.35	-0.0584	0.14	0.90	0.31	-0.0529	0.14
			600	-0.95	0.34	-0.0588	0.14	-0.96	0.31	-0.0532	0.14

			800	-0.89	0.34	-0.0588	0.14	-0.90	0.31	-0.0532	0.14
			1000	0.89	0.92	-0.0201	0.06	0.90	0.86	-0.0324	0.08
CDG	2-2-1	2	200	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
			400	0.93	0.35	-0.0578	0.14	0.95	0.31	-0.0524	0.14
			600	0.25	0.35	-0.0585	0.14	0.56	0.31	-0.0530	0.14
			800	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
			1000	0.87	0.91	-0.0366	0.07	0.88	0.85	-0.0497	0.09
	2-4-1	4	200	-0.09	0.34	-0.0590	0.14	0.31	0.31	-0.0531	0.14
			400	-0.23	0.34	-0.0592	0.14	0.14	0.31	-0.0532	0.14
			600	0.20	0.35	-0.0587	0.14	0.96	0.32	-0.0520	0.14
			800	-0.87	0.34	-0.0590	0.14	-0.91	0.30	-0.0535	0.14
			1000	0.89	0.92	-0.0205	0.06	0.90	0.86	-0.0326	0.08
	2-6-1	6	200	-0.10	0.35	-0.0602	0.14	-0.91	0.30	-0.0534	0.14
			400	0.89	0.35	-0.0584	0.14	0.90	0.31	-0.0529	0.14
			600	-0.95	0.34	-0.0588	0.14	-0.96	0.31	-0.0532	0.14
			800	-0.89	0.34	-0.0588	0.14	-0.90	0.31	-0.0532	0.14
			1000	0.89	0.92	-0.0201	0.06	0.90	0.86	-0.0324	0.08

Based upon statistical analysis, table no. 10 represents the best ANN architecture for anthracnose disease prediction. Among which, 2-2-1 with 1000 epochs was found suitable for LM learning algorithms and for CDG learning algorithms 2-2-1 trained at 200 epochs was suitable for prediction of anthracnose disease of mango. Fig. 12 and 13 represents the training and validation graphs of 2-2-1 architectures trained with LM and CDG learning algorithms.

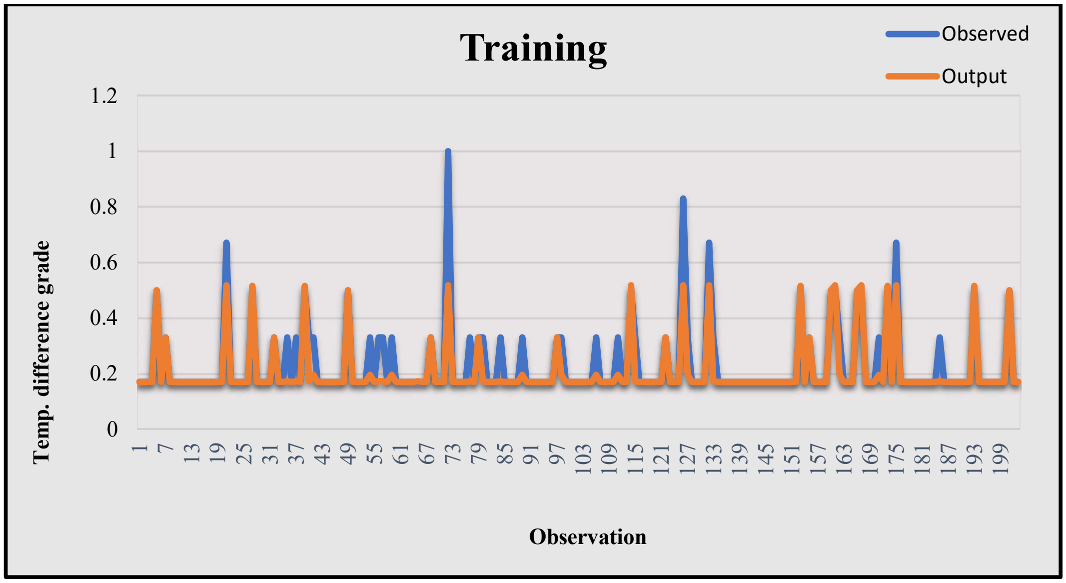
Table 10: Best ANN architecture observed from the performed ANN architecture combinations for detection of anthracnose of mango

Algorithm	ANN arch	Epoch	Statistical Indicators							
			Training				C.V.			
			r	I.A.	MBE	RMSE	r	I.A.	MBE	RMSE
LM	2-2-1	1000	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0329	0.08
	2-4-1	1000	0.89	0.92	-0.0205	0.06	0.90	0.86	-0.0326	0.08
	2-6-1	1000	0.89	0.92	-0.0201	0.06	0.90	0.86	-0.0324	0.08
CDG	2-2-1	200	0.88	0.92	-0.0210	0.07	0.89	0.86	-0.0330	0.08
	2-4-1	1000	0.89	0.92	-0.0205	0.06	0.90	0.86	-0.0326	0.08
	2-6-1	1000	0.89	0.92	-0.0201	0.06	0.90	0.86	-0.0324	0.08

The results of developed ANN model for detection of mango anthracnose were very similar to other ANN models developed for detection of other crop diseases by researchers.

Pujari *et al.* (2014) used multilayer back propagation neural network (BPNN) for ANN based classification of produce affected by identically looking powdery mildew diseases of mango, grape, chilli, wheat, beans and sunflower. They used sigmoid activation functions in the hidden layers. All the algorithms were applied using MATLAB 7.10. In ANN, the overall accuracy of these six categories of disease affected produce using colour, texture and combined features was 70.48 %, 70.07 % and 76.61 % respectively which was less than the results obtained in the developed ANN model for detection of mango anthracnose.

Rice leaf diseases (bacterial leaf blight, brown spot, narrow brown spot and blast) were classified using shape and color features with SVM classifier by Suman and Dhruvakumar (2015). Images of normal and diseased leaf were acquired using digital



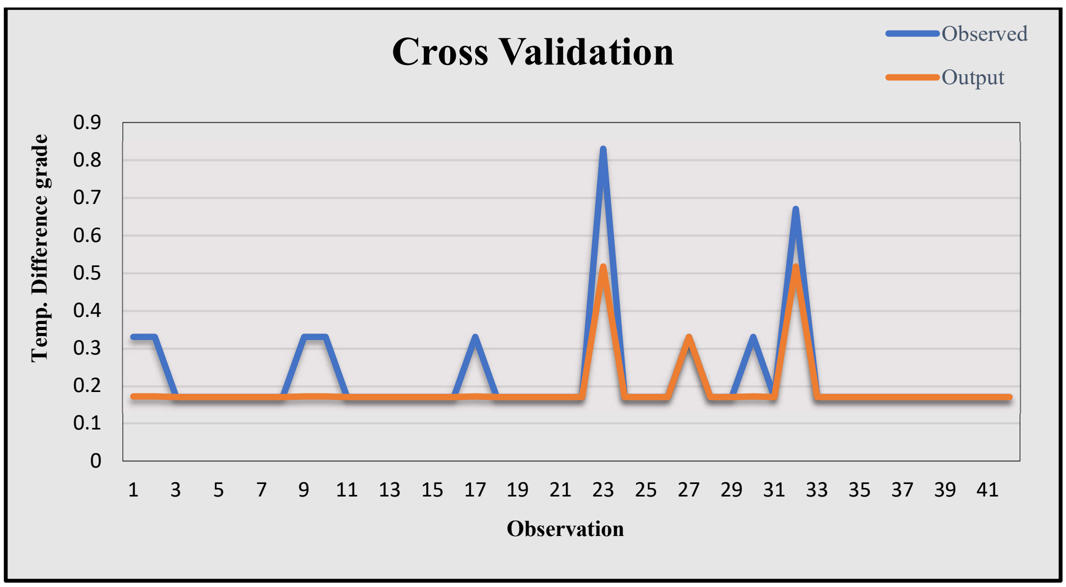


Fig: 12. Training and cross validation graphs of ANN architecture 2-2-1 with Levenberg Marquardt algorithm for developed for detection of anthracnose of mango at 1000 epochs

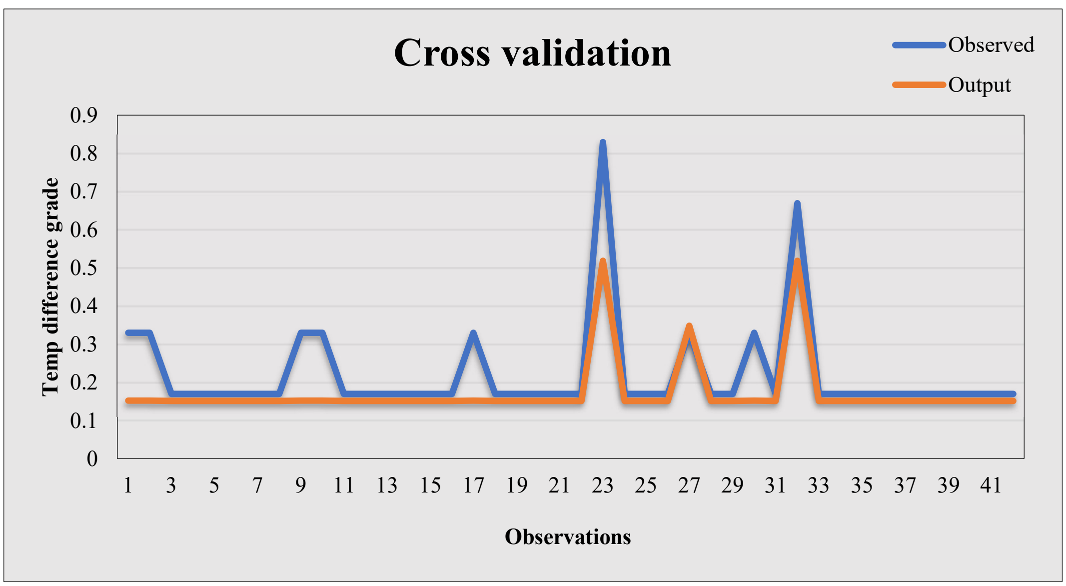


Fig: 13. Training and cross validation graphs of ANN architecture 2-2-1 with Conjugate descent gradient algorithm for developed for detection of anthracnose of mango at 200 epochs

camera with a white background to avoid reflections while capturing the images. The proposed technique achieved 70% accuracy for classification of rice disease which was very less than the accuracy percentage achieved in the developed ANN model for mango anthracnose disease.

A multi-layer perceptron (MLP) model was trained using Back-propagation algorithm of artificial neural network (ANN) for classification of healthy and diseased leaves of beans (Syafiqah *et al.,* 2015). Two multi-layer feed forward networks were used which were multi-layer perceptron (MLP) and radial basis function (RBF) to select the better performing model. The MLP model attained an accuracy of 90.3% with 9.7% error where the of test samples were more than the training samples. The classification accuracy of RBF model was 99.2% with only 0.8% error where the training samples were more than the test samples. The experimental results showed that the RBF network performed better than MLP network.

Moumita and Mantosh (2019) implemented Back-propagation neural network (BPNN) with particle swarm optimization (PSO) for identification and classification of four leaf diseases and healthy leaves of beans *viz.* anthracnose, bacterial blight, two types of leaf spots caused by cercospora and alternaria and healthy leaves. Classification was done using back-propagation and PSO. At first, back-propagation was used to train the feed forward neural network and then NN connection weights were further optimized using particle swarm optimization. For training and testing, the dataset was divided into 75% and 25%, respectively. The neural network was trained for four hundred iterations using back-propagation algorithm and then hundred iterations of PSO were run to optimize the model weights. The model was implemented using 'Matlab'. The developed model attained an accuracy of 96.72%.

The results proved that ANN model for anthracnose disease detection of mango works better than previously used ANN classification techniques.

4.3 Mango powdery mildew disease detection model

4.3.1 Performance of CNN model developed for detection of powdery mildew of mango

The confusion matrix (Fig.14) analysis of mango powdery mildew disease detection model (PLATE XVI) predicted that the developed model achieved 97% accuracy, 98% recall and 96% precision with 97% F1 score. Fig.15 represents the accuracy

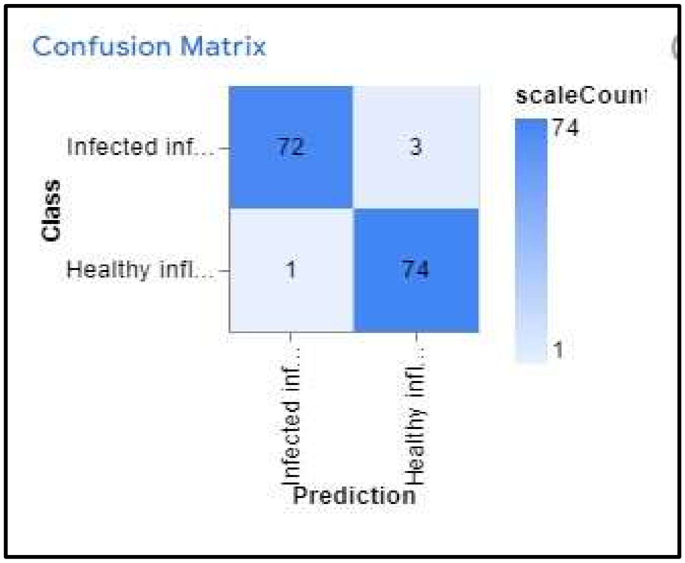
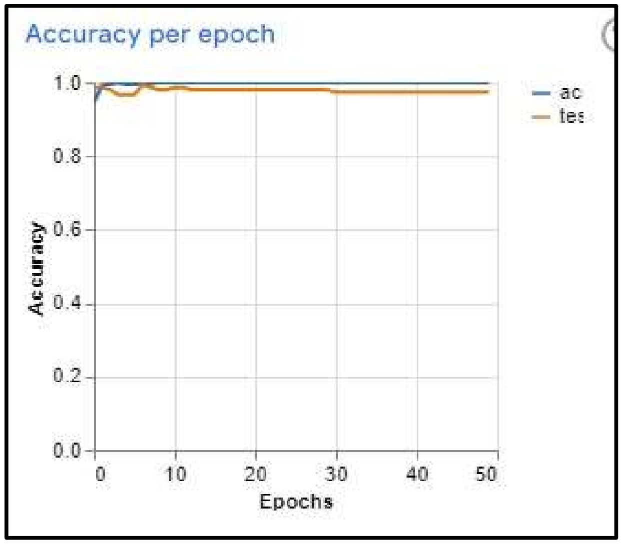
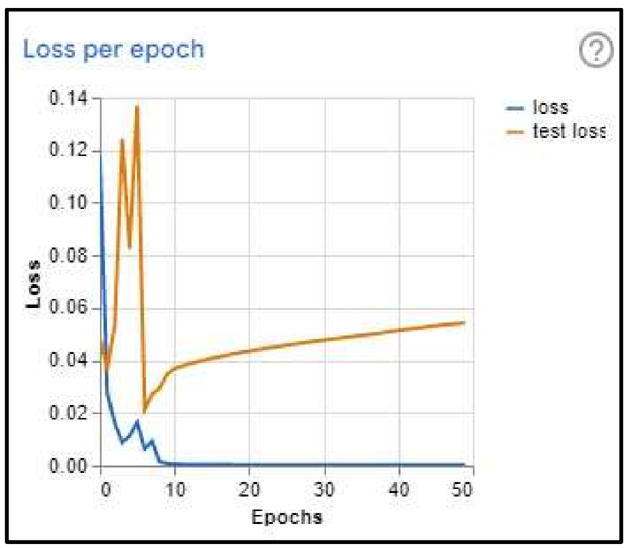


Fig: 14. Confusion matrix of teachable machine model developed for detection of mango powdery mildew disease



Accuracy graph

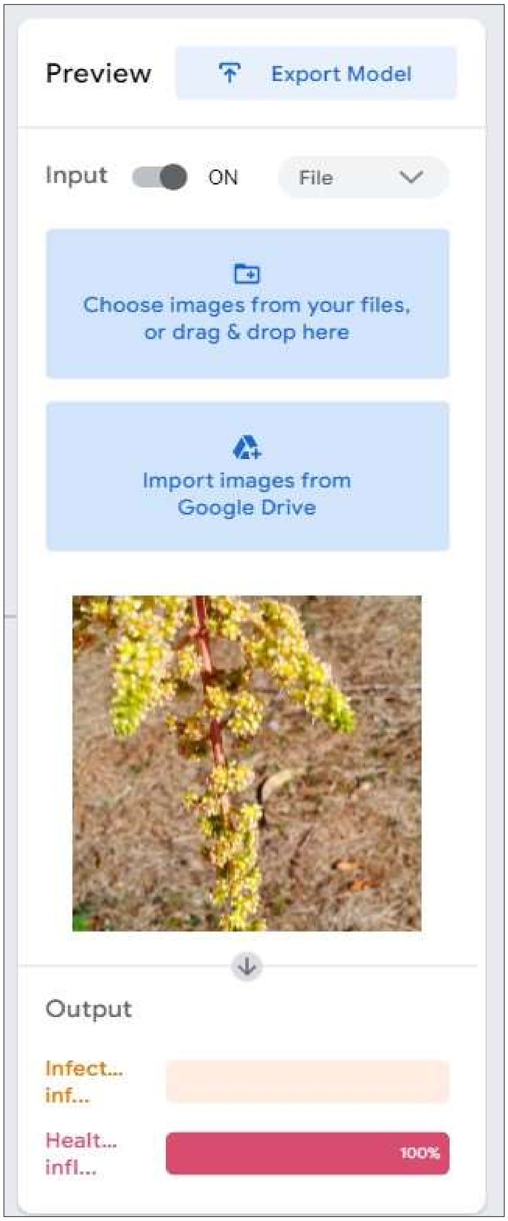


Loss graph

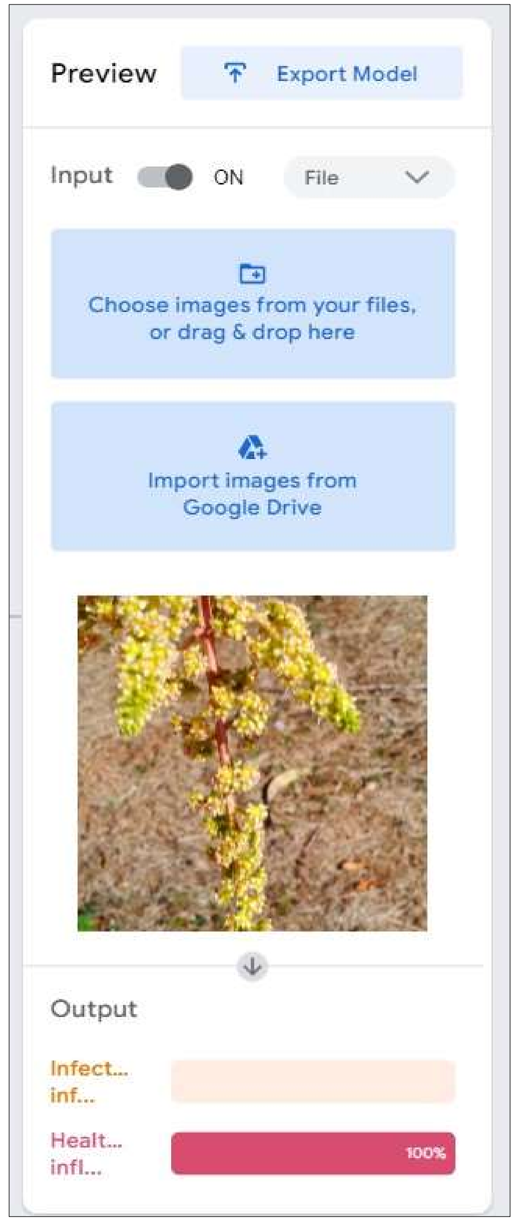
Fig: 15. Accuracy and loss of teachable machine model developed for detection of mango powdery mildew disease

PLATE XVI

Teachable machine model previews developed for detection of powdery mildew of mango



Healthy mango inflorescence



Powdery mildew infected mango inflorescence

and loss curve for mango powdery mildew disease detection model. The gap between training and testing accuracy of powdery mildew disease detection model was very low (Fig.15), so there was no issue of model over fitting and concluded as the model learned correctly.

The results of developed model were parallel to the findings of Mathew and Therese (2021). The researchers developed a model with similar methodology for detection of early blight, late blight and healthy leaf of potato which attained detection accuracy of 98% for early blight, 100% for late blight and 87% for healthy potato leaf. The model was trained on 1000 on field images of each class type.

Developed Teachable Machine model link for detection of powdery mildew of mango:

"https://teachablemachine.withgoogle.com/models/7uul5NqqX/"

4.3.2 Performance of ANN model developed for detection of powdery mildew of mango

The thermal images of mango inflorescence infected with powdery mildew disease showed no significant temperature difference to detect the powdery mildew infection. Therefore, it was not possible to develop the ANN model for detection of powdery mildew of mango using thermal images temperature difference data.

4.4 Android application developed using teachable machine model links for early leaf spot of groundnut, anthracnose and powdery mildew of mango

A free trial account was created on 'Mobiroller' website with one-month free trail programme using email-id to create android application using developed teachable machine disease detection model links. Manually created template with groundnut early leaf spot, anthracnose and powdery mildew of mango images was used as android application preview and background image.

Three application contents were selected *viz*., website link, standard application content and contact information. In website link content, the teachable machine model links of groundnut and mango disease detection models were uploaded separately. In standard content, the detailed information about groundnut early leaf spot disease, anthracnose and powdery mildew diseases of mango was uploaded along with causal organism, symptoms and control measure. Contact information about the information centre *i.e.,* Department of Plant Pathology, College of Agriculture, Dr. Balasaheb Sawant Konkan Krishi Vidyapeeth, Dapoli was provided in the contact content of the application. In application appearance option, general application settings were updated. After which, the completed application dashboard was previewed to check each content was updated successfully. Android Package Kit (APK) file format was used to generate the application and publish on Google Play. Necessary permissions were granted while generating the application. The android application link was forwarded to the registered email-ID after the application was created. The android application preview given in PLATE XVII.

4.5 Mobile application and ANN architecture files storage

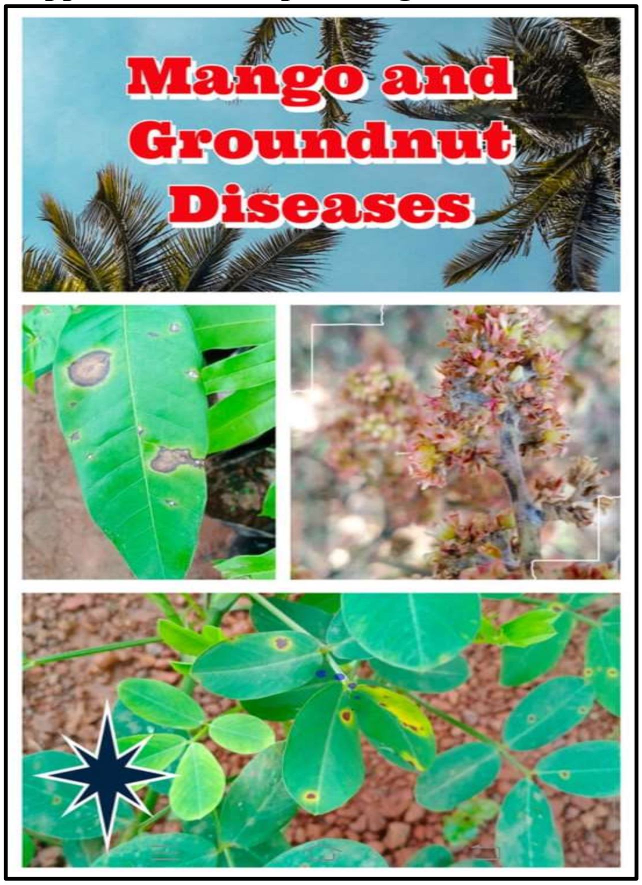
I have included the developed disease detection application file in APK format, along with the ANN architectures, on the CD/Hard drive provided with the thesis hard copy. The application file is designed for installation on Android devices, facilitating easy access and utilization of the disease detection system on compatible smartphones and tablets.

Furthermore, the included ANN architectures have been implemented using NeuroSolutions software, offering a powerful tool for neural network analysis and experimentation. Researchers and practitioners can utilize the ANN architectures within NeuroSolutions to study their performance and adapt them to different scenarios. I have provided the ANN architectures in a format suitable for incorporation with Internet of Things (IoT) systems. This feature allows researchers and developers to integrate the disease detection models into IoT devices, thereby creating a smart disease detection system capable of real-time monitoring and remote diagnostics.

The CD/Hard drive accompanying this thesis serves as an essential repository of supplementary materials, providing access to the disease detection application and ANN architectures. The inclusion of these resources aims to support the reproducibility and further exploration of the developed models.

PLATE XVII

Android application developed using teachable machine models



Application screen

Application content

SUMMARY AND CONCLUSION

CHAPTER V

SUMMARY AND CONCLUSION

This chapter provides a summary of the key findings and contributions of the Ph.D. thesis entitled "Development of Convolutional neural network and Artificial neural network models for detection of leaf spot of groundnut, anthracnose, and powdery mildew of mango." It concludes the research journey by highlighting the achievements, implications, and potential future directions of the study.

In this research, we focused on addressing the challenges associated with the early and accurate detection of early leaf spot of groundnut, anthracnose, and powdery mildew of mango. The research investigated the potential of advanced machine learning techniques, specifically Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) models, to improve the accuracy and efficiency of disease diagnosis. The following key findings were obtained:

- Extensive datasets of high-resolution RGB images and thermal images of infected and healthy groundnut and mango plant leaves and mango inflorescence were collected, pre-processed, and augmented to enhance the diversity and quality of the dataset.
- CNN and ANN models were designed, trained, and fine-tuned using state-of-the-art architectures to accurately identify and classify early leaf spot of groundnut, anthracnose, and powdery mildew of mango.
- Two types of CNN models *viz*. Teachable machine and VGG-16 were used for groundnut early leaf spot disease detection.
- Teachable machine model developed for detection of early leaf spot of groundnut achieved 98% accuracy with 97% precision, 98% recall and 97% F1 score.
- The VGG-16 early leaf spot of groundnut detection model achieved 92.52% accuracy.
- The 2-2-1 ANN architectures trained with Levenberg marquardt and Conjugate descent gradient algorithm both trained at 1000 epochs performed best among all other architectures trained for detection of early leaf spot of groundnut.

- Mango anthracnose and powdery mildew disease detection CNN model were developed using teachable machine method.
- Teachable machine model developed for detection of mango anthracnose attained 96% accuracy, 97% precision, 96% recall, 96% with F1 score and powdery mildew detection model achieved 97% accuracy, 96% precision, 98% recall with 97% F1score.
- The ANN model development for mango disease detection was only possible for mango anthracnose disease. The 2-2-1 architecture with Levenberg marquardt algorithm trained at 1000 epochs and Conjugate descent gradient algorithm trained at 200 epochs performed excellent than other ANN architectures trained.
- Thermal images of powdery mildew infected mango inflorescence showed no significant temperature difference in healthy and infected area of mango inflorescence. Therefore, it was not possible to develop ANN model for detection of powdery mildew of mango using thermal image dataset.
- The developed models outperformed traditional image processing methods and manual diagnosis by experts, demonstrating their effectiveness in disease detection.
- The trained models displayed promising generalization capabilities, showcasing their potential for real-world deployment and widespread application in different geographical locations and environmental conditions.

Conclusion:

This research contributes to the advancement of deep learning techniques in plant pathology by designing and implementing robust CNN and ANN models specifically tailored for the detection of leaf spot of groundnut, anthracnose, and powdery mildew of mango.

The developed models surpass traditional methods and manual diagnosis, offering a non-invasive, rapid, and accurate solution for disease detection, enabling timely interventions and targeted treatments. The models demonstrated strong generalization capabilities across diverse datasets, showcasing their potential for real-world application and scalability in different agricultural settings.

The CNN and ANN models developed in this research can significantly contribute to early disease detection, allowing farmers and agricultural experts to implement timely and targeted measures to control and manage leaf spot of groundnut, anthracnose, and powdery mildew of mango. The accurate disease identification provided by the models enables farmers to adopt precision agriculture practices, reducing the need for indiscriminate pesticide use and optimizing resource allocation.

The developed models can be integrated into smart farming systems, such as android mobile application developed in this research using the developed teachable machine model links, Internet of Things (IoT) devices, to provide real-time disease monitoring and decision support for farmers.

Future research directions that can build upon this work include:

- The developed models can be extended to detect and classify additional plant diseases to provide a comprehensive disease diagnosis system.
- Data from multiple sensors, such as hyperspectral imaging or thermal imaging can be incorporated into these models, to enhance the accuracy and reliability of disease detection.
- Collaboration with agricultural stakeholders and industry can help deploy the developed models in real-world scenarios.

LITERATURE CITED

LITERATURE CITED

Akila M and Deepan P. 2018. Detection and classification of Plant Leaf Diseases by using Deep Learning Algorithm. *Int. J. Eng. Res*. **6(7)**:1-5.

Alatawi A. A., Alomani S. M., Najd I. A. and Ayaz M. 2022. Plant disease detection using AI based VGG-16 model. *IJACSA*. **13(4)**:718-727.

Anna A and Shanthini A. 2023. Leaf diseases identification and classification of self-collected dataset on groundnut crop using progressive convolutional neural network (PGCNN). *Int. J. Adv. Com. Sci. App.* **14(2)**:364-373.

Aqil M, Tabri F, Andayani N N, Panikkai S, Suwardi, Efendi R, Bunyamin Z, Azrai M and Ratule T, 2021. 2nd IOS Conf. Series: Earth and Environmental Science. Pp:1-6.

Arivazhagan S and Ligi S V. 2018. Mango leaf diseases identification using Convolutional Neural Network. *International Journal of Pure and Applied Mathematics*. **120(6)**: 11067-11079.

Arya S and Singh R. 2019. A comparative study of CNN and Alexnet for detection of disease in Potato and Mango leaf. Proceedings of International Conference on Issues and Challenges in Intelligent Computing Techniques.

Berengena J, Bakis R and Koc C. 2016. Mapping of precipitation, temperature and evaporation in Seydisuyu basin with the help of distance related estimation methods. *Journal of Geographic Information System*. **8**:224-237.

Bhattacharyya T, B R Salvi, P M Haldankar and N V Dalvi. 2019. Growing Alphonso mango on Konkan Laterites, Maharashtra. *Indian Journal of Fertilizers*. **15(8)**: 878- 885

Birthal P S, Parthasarathy R P, Nigam S N, Bantila M S and Bhagavatula S. 2010. Groundnut and Soyabean economics in Asia: Facts, Trends and Outlook. ICRISAT report. 92 pp.

Dadrasjavan F, Samadzadegan F, Seyed and Pourazar S H. 2019. UAV-based multispectral imagery for fast Citrus Greening detection. *J. Plant Dis Prot* 126:307–318.

Dandawate Y and Kokare R. 2015. An automated approach for classification of plant diseases towards development of futuristic decision support system in Indian perspective. *Int. Conf. Adv. Comput*.

Deeba K and Amutha B. 2020. ResNet – Deep Neural Network architecture for leaf disease classification. *Microprocessors and Microsystems*.

DehghaniSanij, Hossein Y and Velu R. 2004. Assessment of evapotranspiration estimation models for use in semi-arid environments. *Agric. Water Management*. **64**:91-106.

Devi K S, Srinivasan P and Bandhopadhyay S. 2020. H2K- A robust and optimum approach for detection and classification of groundnut leaf diseases. *Computers and Electronics in Agriculture.* **178**:1-8.

Felipe Arauz. 2000. Mango Anthracnose: Economic impact and current options of Integrated management. *Plant Disease*. **84(6)**:600-611.

Fletcher R and Reeves C M. 1964. Function minimization by conjugate gradient. *Comp. J.* **7**:149-154.

Fulari U N, Shastri R K and Fulari A N. 2020. Leaf disease detection using machine learning. *Journal of Seybold report*. **15(9)**:1828-1832.

Gining R A J M, Fauzi S S M, Yusoff N M, Razak T R, Ismail M H, Zaki N A and Abdullah F. 2021. Harumanis mango leaf disease recognition system using image processing technique. *Indonesian J Elec Eng & Comp Sci*. **23(1)**:378-386.

Gowrishankar K and S Lakshmi Prabha. 2020. An integrated image processing approach for diagnosis of groundnut plant leaf disease using ANN and GLCM. *J Sci Ind Res*. **79**: 372-376.

Gowrishankar K and S Lakshmi Prabha. 2020. Diagnosis of groundnut plat leaf diseases using threshold-based color segmentation and artificial neural network. *IJERT*. **8(3)**:1-4.

Hanson A, M G Joel, Joy Annette and Francis Jerin. 2017. Plant Leaf Disease detection using Deep learning and Convolutional Neural Network. *Int. J. Eng. Comput. Sci*. **7(3)**: 5324-5328.

Ingle P. M. and Purohit R. C. 2020. Assessment of reference evapotranspiration using ANN at Mulde, Maharashtra. *Journal of Agrometeoroloy*. **22(2)**:220-225.

Karande R A, M S Joshi, N R Phutankar, J J Kadam and P G Borkar. 2017. *In vitro* and field evaluation of Bio organics against powdery mildew of mango caused by *Oidium mangiferae* Berthe. *Int. J. Chem. Stud*. **5(3)**:167-169.

Kisi O. 2007. Evapotranspiration modelling from climatic data using a neural computing technique. *Hydrol Process*. **21**:1925-1934.

Kumar M. 2011. Artificial neural networks approach in evapotranspiration modelling: A review. *Irrig. Sci.* **29**:11-25.

Kumar P, Ashtekar S, Jayakrishna S S, Bharath K P, Vanathi P T and Kumar R M. 2021. Proceedings of the Fifth International Conference on Computing Methodologies and Communication.

Kumar A and Kumar A. 2023. Plant disease detection using VGG16. *IJCRT.* **11(1)**:770-775.

Kumari P and Singh R. 2017. Anthracnose of Mango Incited by *Colletotrichum Gloeosporioides*: A Comprehensive Review. *Int. J. App. Biosci.* **5(1)**:48-56.

Kumar S R and Sowrirajan S. 2016. Automatic leaf disease detection and classification using hybrid features and supervised classifier. *IJAREEIE*. **5(6)**:4556-4563.

Mathew M and Therese Y M. 2021. Object detection based on Teachable machine. *Journal of VLSI design and Signal Processing.* **7(2)**:20-26.

Mayuri K P and Vani Priya. 2018. Role of image processing and machine learning techniques in disease recognition, diagnosis and yield prediction of crops: A review. *Int. J. Adv. Res. Com. Sci.* **9(2)**:788-795.

Midhun P M and Therese Y M. 2021. Object Detection based on Teachable Machine. *Journal of VLSI Design and Signal Processing*. Vol.**7(2):**20-26.

Misra A K. 2001. Powdery mildew – A serious disease of mango. *J. Appl. Hort.* **3(1):**63-68.

Moumita C and Mantosh B. 2019. Plant disease identification and classification using Back-Propagation neural network with Particle swarm optimization. Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019).

Muthukumaran S, Geetha P and Ramraj E. 2021. Leaf spot disease image classification for Groundnut crop using Deep Convolutional neural network. *Turkish Journal of Computer and Mathematics Education.* **12(14)**:3166-3179.

Neha S and Giridharan A. 2021. Predicting groundnut disease using CNN models. *Journal of University of Shanghai for Science and Technology*. **23(6)**:756-766.

Nikam R and Sadavarte M. 2015. Application of image processing technique in mango leaves disease severity measurement. Proceedings of National Conference on Emerging Trends in Computer, Electrical and Electronics.

Padol P B and Sawant S D. 2016. Fusion classification techniques used to detect downy and powdery mildew of grape leaf diseases. Proceedings of International conference on Global Trends in Signal Processing, Information Computing and Communication.

Patayon U B and Crisostomo R V. 2022. Peanut leaf spot disease identification using pre-trained deep convolutional neural network. *Int. J. Elec & Comp Eng*. **12(3)**:3005-3012.

Patil L and Nagpure T. 2022. Detection of Northern Leaf Blight Disease of Maize Based on Teachable Machine (AI). AgriCos e-Newsletter. Vol.**3(8):**105-107.

Paul P A and Munkvold G P. 2004. Regression and Artificial neural network modelling for the prediction of gray leaf spot of maize. *Phytopathology*. **95(4)**:388-396.

Priya P and D'souza A D. 2015. Study of feature extraction techniques for the detection of diseases of agricultural products. *Int. J. Inov. Res. Electr. Electron. Instrum. Control Eng*. **3(1)**:4-8.

Pujari J D, Yakkundimath R and Byadgi A S. 2014. Recognition and classification of produce affected by identically looking powdery mildew disease. *Acta Technol. Agric*. 2:29-34.

Qi Haixia, Yu Liang, Quanchen Ding and Jun Zou. 2021. Automatic identification of Peanut-leaf diseases based on Stack Ensemble. *Appl. Sci*. 11:1-14.

Rajbongshi A, Khan T, Rahman M, Pramanik A, Siddiquee S M T and Chakraborty N R. 2021. Recognition of mango leaf disease using convolutional neural network models: a transfer learning approach. *Indonesian J.Elec. Eng &Comp Sci*. **23(3)**:1681-1688.

Rakholi R M, Tailor J H, Saini J R, Kaur J and Pahuja H. 2022. Groundnuts leaf disease recognition using neural network with progressive resizing. *Int. Jr. of Adv. Com. Sci. & App.* **13(6)**:83-88.

Ramakrishnan M and Sahaya A N. 2015. Groundnut leaf disease detection and classification by using back propagation algorithm. *Proc. IEEE Int. Conf. Commun. Signal Process*.

Rangarajan A K, Purushothaman R, Prabhakar M and Szczepanski C. 2021. Crop identification and disease classification using traditional machine learning and deep learning approaches. *J. Eng. Res.* Pp:1-41.

Rao U S, Swathi R, Sanjana V, Arpitha L, Chandrasekhar K and Chinmayi. 2021. Deep learning precision farming: Grapes and Mango leaf disease detection by Transfer learning. Global Transitions Proceedings 2.

Raut R A, Patil P D, Haldavnekar P C, Munj A Y, Baviskar S B and Sanas M P. 2017. Epidemiological studies on mango powdery mildew (OidiummangiferaeBerthe) cv. Alphanso. *Int. J. Chem. Stud*. **5(3):**525-526.

Shelar N, Shinde S, Sawant S, Dhumal S and Fakir K. 2022. Plant disease detection using CNN. Proceedings of Information Technology Management Web Conferences-2022.

Simonyan K and Zisserman A. 2014. Very deep convolutional networks for large scale image recognition. Proceedings of ICLR.

Singh U P, Chouhan S S, Jain S and Jain S. 2019. Multilayer convolutional neural network for the classification of mango leavese infected by Anthracnose disease. *Proc IEEE Inst Electr Electron Eng* P. 7: 43721-43729.

Sivasankaran S, Mohan K J and Nazer G M. 2022. A comparative CNN based deep learning model investigation for identifying and classifying the leaf diseases of *Arachis hypogea* (Groundnut crop) grown in the semi-arid landscapes of Villupuram district of Tamil Nadu. *Journal of Algebraic Statistics*. **13(2)**:232-249.

Srikanth T. 2019. Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *Int. J. Sci. Res. Pub.* **9(10)**:2019.

Subrahmanyam P, Mehan V K, Nevil D J and McDonald D. 1980. Research on fungal diseases of groundnut at ICRISAT. Pp:193-198. *IJEEE*. **7(1)**:239-250.

Suman T and Dhruvakumar T. 2015. Classification of paddy leaf diseases using shape and color features.

Syafiqah I, Mohd H F R, Siti N A, Mohd K and Hashim S. 2015. Leaf disease classification using Artificial neural network. *Jurnal Teknologi (Sciences & Engineering)* **77(17):** 109-114.

Taujuddin N S A M, Mazlan A I A, Ibrahim R, Sari S, Ghani A R A, Senan N and Muda W H N W. 2020. *Int. J. Adv. Comput. Sci. Appl*. **11(8):**407-411.

Thirumalaisamy P P, Bhat M N, Nisar S, Yadav S K, Chandra P and Vennila S. 2021. Status of groundnut early leaf spot (Cercospora arachidicola) at hot semi-arid ecoregion of Gujarat (India): past, present and future. *Acta Scientific Agriculture* **5(8):**48-52.

Tiwari S, Kumar N, Pramanik A, Joshi E, Sasode D, Sikarwar R S, Tomar R S, Tripathi M K, Kandalkar V S and Singh A K. 2018. Breeding for foliar disease resistance in groundnut usingconventional and molecular approaches. Proceedings of National Conference on Current Trends in Plant Science and Molecular Biology for Food security and Climate Resilient Agriculture.

Trang K L, TonThat N, Thao M G and Tran Ta Thi N. 2019. Mango Disease Identification by a Deep Neural Residual Network with Contrast Enhancement and Transfer

Learning. IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies. Pp:138-142.

Vaishnnave M P, Devi K S, Srinivasan P and Arutperumjothi G. 2019. Detection and classification of Groundnut leaf diseases using KNN classifier. Proceedings of International Conference on Systems Computation Automation and Networking.

Vaishnnave M P, Devi K S and Ganeshkumar P. 2020. Automatic method for classification of Groundnut diseases using Deep Convolutional neural network. *Soft Computing*. **24**:16347-16360.

Vangujare D A and Tuppad S G. 2019. Mango leaf disease severity measurement using Image processing techniques. *J. Inf. Comput. Sci*. **6(7):**75-83.

Vanquez R F and Feyen J. 2003. Effect of potential evapotranspiration estimates on effective parameters and performance of the MIKE SHE-code applied to as medium-size catchment. *J. Hydro*. **270**:309-327.

Veling Shripad S, Kalekar R S, Ajgaonkar L V, Mestry N V and Gawade N N. 2019. Mango disease detection by using Image processing. *Int. J. Res. Appl. Sci. Eng. Technol*. **7(4):**3717-3726.

Waghmode B D, Kamble A, Navhale V and Mahadkar U V. 2017. Groundnut research in Konkan: A new look. *J. Adv. Agric. Technol*. **1(2):**183-194.

Willmott C. J. 1982. Some comments on the Evaluation of Model Performance. *Bulletin American Meteorological Society*. **63(11)**:1309-1313.

Wongshil S, Chantrasri P and Sureephong P. 2021. Machine learning algorithm development for detection of Mango infected by Anthracnose disease. Proceedings of Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering.

Yin Min Oo and Nay Chi Htum. 2018. Plant leaf disease detection and classification using Image processing. *Int. J. Eng. Res.* **5(9):** 516-523.

Zanetti S S, Oliveira V P S, Almeida F T and Bernardo S. 2007. Estimating evapotranspiration using artificial neural network and minimum climatological data. *J. Irrig. Drain. Eng. ASCR.* **133**:83-89.

APPENDICES

APPENDIX – I

ABBREVIATIONS USED

%	:	Per cent
/	:	Per
^{0}C	:	Degree centigrade
AM	:	Anti Meridiem
ANN	:	Artificial Neural Network
APK	:	Android Package
BPNN-FF	:	Back Propagation Neural Network – Feed Forward
CDG	:	Conjugate descent gradient
cm		Centimetre
CNN	:	Convolutional Neural Network
CPU	:	Central Processing Unit
DCNN	:	Deep Convolutional Neural Network
DCT	:	Discrete Cosine Transform
DFT	:	Discrete Fourier Transform
Dist.	:	District
DWT	:	Discrete Wavelet Transform
et al.	:	And others
etc.	:	Etcetera
Fig.	:	Figure
FNR	:	False Negative Rate
FPR	:	False Positive Rate
GB	:	Gigabit
Gen	:	Generation

GLCM	:	Gray-level Co-Occurrence Matrix
GUI	:	Graphical User Interface
ha	:	Hectare
HIS	:	Hue, Intensity, and Saturation
HL	:	Healthy Leaves
HOG	:	Histogram on Oriented Gradient
I.A.	:	Index of Agreement
i.e.	:	That is
iOS	:	i-Phone Operating System
km/h	:	Kilometre per hour
KNN	:	K-nearest neighbour classifier
LM	:	Levenberg Marquardt
LSD	:	Leaf spot disease
mt	:	Metre
mA-h	:	Megahertz
mm	:	Millimetre
MBE	:	Mean Bias Error
MLP	:	Multi-layer Perceptron
MP	:	Megapixels’
Min. temp.	:	Minimum Temperature
Max. temp.	:	Maximum Temperature
PM	:	Post Meridiem
PSO	:	Particle Swarm Optimization
r	:	Correlation coefficient
RBF	:	Radial Basis Function
RD	:	Rust disease

RD + SD	:	Rust and Scorch disease
RGB	:	Red, Green, Blue
RMSE	:	Root Mean Square Error
SD	:	Scorch disease
SGD	:	Stochastic Gradient Descent
S.E.	:	Standard error
St. Dev.	:	Standard Deviation
SVM	:	Support Vector Machine
TPU	:	Tensor Processing Units
VGG-16	:	Visual Geometry Group- 16
viz.	:	Namely
hrs	:	Hours

APPENDIX II

Terms used

1. Artificial intelligence

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to perform tasks that typically require human intelligence.

2. Artificial neural network

It is a network of interconnected artificial neurons, also known as nodes or units, that work together to process and transmit information.

3. Batch size

Batch size refers to the number of training examples or data points that are processed together in a single forward and backward pass during the training of a neural network.

4. Binary image

A binary image is a type of digital image that consists of only two possible pixel values or colors, typically represented as black and white.

5. Confusion matrix

A confusion matrix, also known as an error matrix, is a table that provides a comprehensive evaluation of the performance of a classification model. It summarizes the predicted and actual classes of a set of data points, enabling the analysis of the model's accuracy and error rates.

6. Convolutional neural network

A Convolutional Neural Network (CNN) is a specialized type of neural network that is particularly effective in analyzing visual data, such as images and videos. CNNs are designed to automatically learn and extract relevant features from input data through the application of multiple convolutional layers.

7. Conjugate descent gradient

The Conjugate Gradient (CG) method is an iterative optimization algorithm used to solve systems of linear equations and perform unconstrained optimization. It is particularly efficient for solving large-scale linear systems that are symmetric and positive definite.

8. Correlation of coefficient

The correlation coefficient is a statistical measure that quantifies the strength and direction of the linear relationship between two variables. It provides a numerical value that ranges between -1 and +1, indicating the degree of correlation between the variables.

9. Epochs

Epochs refer to the number of times the entire training dataset is passed forward and backward through the neural network during the training process.

10. F1 score

The F1 score is a commonly used performance metric in binary classification tasks, particularly when the dataset is imbalanced. It combines precision and recall into a single value that provides an overall assessment of a model's accuracy.

11. Feature extraction

Feature extraction is a fundamental process in machine learning and pattern recognition that involves transforming raw input data into a set of meaningful and representative features.

12. Hidden layer

A hidden layer refers to one or more layers of artificial neurons or nodes that come between the input layer and the output layer. These layers are called "hidden" because their nodes do not directly interact with the external environment or provide outputs to the outside world.

13. Image processing

Image processing involves applying various algorithms and methods to images to extract meaningful information, improve visual quality, or facilitate further analysis and interpretation.

14. Image segmentation

Image segmentation divides the image into distinct regions based on similarities in color, texture, intensity, or other visual properties.

15. Index of Agreement

The Index of Agreement (I.A.), also known as the Index of Concordance or Willmott's Index, is a statistical measure used to assess the accuracy or agreement between observed and predicted values in a predictive modelling or forecasting context.

16. Interface

An interface refers to a programming construct that defines a contract or a set of methods, properties, and events that a class or object must implement. The purpose of an interface in software development is to enable code reusability, modularity, and maintainability.

17. Kernel filter

A kernel filter, also known as a convolution kernel or a filter mask, refers to a small matrix of coefficients or weights that is applied to an image through a mathematical operation known as convolution.

18. Kernel size

It specifies the width and height of the square or rectangular matrix of coefficients that defines the filter.

19. Learning algorithm

Learning algorithm is a set of rules or procedures used to train a machine learning model.

20. Learning rate

The learning rate is a hyperparameter that determines the step size or the rate at which a model's parameters are updated during the training process.

21. Levenberg Marquardt

Levenberg-Marquardt (LM) is an optimization algorithm commonly used for solving nonlinear least squares problems.

22. Machine learning

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed.

23. Maximum pooling layer

It is a type of pooling layer used for down sampling and spatial dimension reduction.

24. Output layer

The output layer of a neural network is the final layer in the network that produces the network's output or prediction.

25. Pattern recognition

Pattern recognition is the process of identifying and classifying patterns or regularities in data.

26. Precision

Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives plus false positives). It quantifies the model's ability to minimize false positives.

27. Programming language

A programming language is a formal language used to write computer programs or software.

28. Python library

A Python library is a collection of pre-written code modules or packages that provide a set of functions, classes, and methods to extend the functionality of Python programming language.

29. Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives plus false negatives). It quantifies the model's ability to minimize false negatives.

30. Root Mean Square Error

It measures the average magnitude of the differences between predicted values and the corresponding true values.

31. Transfer function

It refers to the activation function applied to the output of each neuron in a neural network layer. The transfer function determines the output of a neuron based on the weighted sum of its inputs.