DEVELOPMENT OF ANALYTICAL AND PREDICTIVE TECHNIQUES FOR MACHINE LEARNING

THESIS

Submitted in fulfillment of the requirement of Degree of

DOCTOR OF PHILOSOPHY

to

The Faculty of Engineering and Technology

by

SURABHI LINGWAL

(Registration No.: 17-YMCA-902003)

Under the Joint Supervision of

PROF. KOMAL KUMAR BHATIA
Supervisor

PROF. MANJEET SINGH
Co-Supervisor

Department of Computer Engineering

J.C. Bose University of Science and Technology, YMCA, Faridabad

Sector-6, Mathura Road, Faridabad-121006, Haryana, INDIA

July 2023

DEDICATION

I dedicate this thesis to the almighty GOD and my family

who

gave me strength and patience to complete this research work.

DECLARATION

I hereby declare that this thesis entitled **"DEVELOPMENT OF ANALYTICAL AND PREDICTIVE TECHNIQUES FOR MACHINE LEARNING"** by **SURABHI LINGWAL**, being submitted in fulfillment of requirements for the Degree of Doctor of Philosophy in the Department of Computer Engineering under Faculty of Informatics and Computing of J.C. Bose University of Science and Technology YMCA, Faridabad, during the academic year 2022-23, is a bona fide record of my original work carried out under the guidance and supervision of **DR. KOMAL KUMAR BHATIA, PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING & DR. MANJEET SINGH, PROFESSOR, DEPARTMENT OF COMPUTER APPLICATIONS** and has not been presented elsewhere.

I further declare that the thesis does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University.

SURABHI LINGWAL

Registration No. 17-YMCA-902003

CERTIFICATE

This is to certify that this thesis entitled **"DEVELOPMENT OF ANALYTICAL AND PREDICTIVE TECHNIQUES FOR MACHINE LEARNING"** by **SURABHI LINGWAL** submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy in DEPARTMENT OF COMPUTER ENGINEERING, under Faculty of Informatics and Computing of J.C. Bose University of Science and Technology, YMCA, Faridabad, during the academic year 2022-23, is a bona fide record of work carried out under my guidance and supervision.

I further declare that to the best of my knowledge, the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

Dr. Komal Kumar Bhatia
Professor
Department of Computer Engineering
Faculty of Informatics & Computing
J.C. Bose University of Science & Technology, YMCA, Faridabad

Dr. Manjeet Singh
Professor
Department of Computer Applications
Faculty of Informatics & Computing
J.C. Bose University of Science & Technology, YMCA, Faridabad

Dated:

ACKNOWLEDGMENT

It gives me great pleasure to convey my appreciation and gratitude to my Ph.D. supervisor **Dr. Komal Kumar Bhatia**, Professor, Department of Computer Engineering, and my co-supervisor **Dr. Manjeet Singh**, Professor, Department of Computer Applications, Faculty of Informatics & Computing, J. C. Bose University of Science & Technology, YMCA, Faridabad, for their insight, vision, knowledge, direction, enthusiastic involvement, and constant support throughout the process. Furthermore, I appreciate their arduous efforts in carefully reviewing and revising the manuscripts; without them, this task would not have been possible.

I owe a debt of gratitude to **Dr. Atul Mishra**, Chairman of the Department of Computer Engineering, **Dr. Neelam Duhan** & **Dr. Ashlesha**, Ph.D. Coordinator, Department of Computer Engineering and all the other esteemed faculty members of **J. C. Bose University of Science & Technology, YMCA, Faridabad** for providing the necessary resources, assistance, and support for conducting the research.

The moral support, love, encouragement, and blessings of my parents, **Mr. K. L. Lingwal** and **Mrs. Deepmala Lingwal**, as well as my in-laws, **Mr. R. S. Negi** and **Mrs. Savitri Negi**, allowed me to successfully finish this work. I want to thank my daughter **Riyanshika Negi** and my husband **Mr. Rajeev Negi** in particular for their support, love, and understanding throughout this journey.

I wish to thank my research fellow **Ms. Dimple** and my colleague **Mr. Banit Negi** for their support and encouragement as I worked on my project. I also want to express my heartfelt gratitude to all of my friends, co-workers, and other family members−whose names are not listed here−who have accompanied me in both good and difficult times.

I owe thanks to every reviewer for their insightful comments on my research articles. My research effort was positively shaped by their feedback and ideas. A special thanks go out to the DRC members for their helpful advice, encouragement, and support. Ad-

ditionally, I am grateful in particular to the personnel of the department of computer engineering for their prompt assistance and ongoing cooperation with the investigation.

Last but not least, I owe and thank the Almighty for supporting me in this endeavor.

Surabhi Lingwal

Registration No. 17-YMCA-902003

ABSTRACT

Though an important sector of the Indian economy, agriculture is still lagging in technological advancement. It is facing several challenges in its different fields. Varieties of crops and diseases and their manual identification and classification require much time and labor. These all issues must be automated to manage food production, distribution and consumption correctly. With the development of artificial intelligence and its subfields like machine learning, computer vision, pattern recognition and deep learning, new approaches for automating many agricultural industry activities are being developed to address long-standing issues. These technologies are helping farmers to improve their productivity in the face of unfavorable environmental conditions. Machine learning techniques are growing and getting employed for detecting land boundaries, which facilitates agricultural land recognition and land resource management.

The thesis has proposed various techniques emphasizing different agricultural challenges that require automation to meet the demand of the general population and food industry. Countries like India are one of the largest producers of several crops, but they still lack automation for these farming practices such as disease detection. Therefore, one of the proposed techniques in the thesis focused on detecting and identifying food crops and diseases. A model "Agri-CNN" has been proposed and structured based on deep learning which performs the automatic classification and detection of various crops and diseases which can be used to protect the crops from severe damage and risk by avoiding mitigation for the entire cropland. Based on visual characteristics and attributes passed as input to the deep network, the technique has carried out classification through supervised learning.

While evaluating the accuracy of classification and prediction for crops and diseases, it is sometimes found that the result may deteriorate with irrelevant features and attributes. This issue was elucidated in the thesis by proposing a new dimensionality reduction model that helps to identify and remove the features that do not contribute to improving accuracy. This hybrid "Info_PCA" model is developed for the selection and extraction of attributes based on combining the properties of both PCA and Shannon entropy. It is found that the reduced feature set through this model enhances the performance for

classification and prediction.

Crop yield prediction is another area covered by the thesis concerning agriculture. For predicting crop yield based on meteorological and agricultural conditions, a machine learning "RaNN" model has been developed. This model integrates the functionalities of Random Forest and ANN, along with employing feature sampling and majority voting techniques for dimensionality reduction, to produce the model that most accurately predicts crop yield. Additionally, the suggested model is contrasted with various machine learning techniques.

Agriculture cropland mapping is another area that requires attention to determine the amount of agricultural land still under cultivation. The various crops grown on agricultural land must be identified to determine the crop or agricultural area occupied over time. Therefore, research of different image segmentation algorithms using thresholding and machine learning models is carried out to detect and identify the agricultural areas. The results given by the implementation are analyzed and compared, and it is found that the agriculture cropland mapping also helps to identify and estimate the area under cultivation from satellite imagery. In general, these algorithms extract the features from the pixels to identify the similar and dissimilar attributes to detect the region of interest for farmland mapping through semantic segmentation.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Title** **Page No.**

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
AIS	Artificial Immune System
ANN	Artificial Neural Network
CART	Classification and Regression Tree
C-GAN	Conditional Generative Adversarial Network
CLARA	Clustering Large Applications
CNN	Convolutional Neural Network
DBNs	Deep Belief Networks
DBSCAN	Density-based spatial clustering of applications with noise
EL	Ensemble Learner
FP-Growth	Frequent Pattern-Growth
GLCM	Gray Level Co-occurrence Marix
GPS	Global Positioning System
HMAX	Highest Maximum temperature
IG	Information Gain
IoU	Intersection over Union
kNN	k Nearest Neighbour
LDA	Linear Discriminant Analysis
LMIN	Lowest Minimum Temperature
LSTMs	Long Short Term Memory Networks
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MEVP	Mean Evaporation
MLP	Multilayer Perceptron
MLR	Multiple Linear Regression

MMAX	Mean Maximum Temperature
MMIN	Mean Minimum Temperature
MPPE	Mean Percentage Prediction Error
MSSH	Duration of Sunshine
NVDI	Normalized Difference Vegetation Index
MWS	Mean Wind Speed
PAM	Partition around Medoids
PCA	Principal Component Analysis
R^2	Coefficient of Determination
RBFNs	Radial Basis Function Networks
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RMSE	Root Mean Square Error
RN	Number of Rainy Days
SAR	Synthetic Aperture Radar
SETM	Sort and merge scan
SRC	Sparse Representation Classifier
SVM	Support Vector Machine
TMRF	Total Rainfall in months
UAVs	Unmanned Aerial Vehicles
VGG	Visual Graphics Group

CHAPTER I

INTRODUCTION

CHAPTER 1

INTRODUCTION

This chapter introduces the driving forces behind the basic hypothesis of research. The chapter also describes the technologies and research fields used to address the challenges.

1.1 GENERAL

The rise in population and income level of the urban and rural people increases the demand for food products which drives the agriculture sector to export the food supply. Though agriculture is engaging 40% of Indian labor and sustaining 70% of the rural population, it is still contributing insufficiently to the Indian economy [1]. Agriculture is recognized as a source of livelihood that directly or indirectly benefits the majority of the population. The majority of rural residents in India depend on agriculture, which involves several activities such as crop cultivation, harvesting and monitoring [2]. It contributes a considerable sum to most developing nations' income. Besides producing food, the sector is also responsible for providing fodder for domestic animals. Products like tea, rice, spices and coffee are the export items that rely on agriculture which could be enhanced with technological development in the agriculture sector. Aside from producing edible and non-edible oils, cotton and jute textiles, sugar refineries and other industries, agriculture also provides raw materials for many other related work areas. The agricultural policies and technologies need to be re-evaluated and reframed, covering factors of production like land, labor, protection against crops, marketing, etc. [3]. This development could contribute to a marketable surplus. The internal trade through agriculture is responsible for the significant revenue to the Government [4]. The agriculture industry suffers from the issues of food waste and labor shortage which raises

the demand for automation in agriculture, specifically after the pandemic COVID-19. The Indian food industry is growing as one of the largest industries, reaching nearly 35% of the Indian food market supply [1]. It also raises its contribution to world food consumption daily. Food security is the primary concern in India that strongly depends on the production of cereal crops, vegetables and fruits. The agricultural stability of any country is responsible for the nation's food security. To satisfy the growing demand of people, the agriculture sector needs to collaborate with the competitive and automated environment. India is one of the biggest manufacturers of several pulses, rice, wheat, and a wide variety of vegetables and fruits in the world. Thus, there is a rising need for managing and monitoring the sustainable production of these crops. The evolution and management of the agriculture sector are required to maintain the country's economy and food supply for the population to make a country socially and economically strong. Agriculture is responsible for human survival which satisfies the living need and provides employment to a large section of the world's population. Despite the need for automation in agriculture, most farmers are still involved in traditional farming practices, which produces low revenue. Moreover, the classical practices with a lack of automation generate problems like inefficient decision-making in crop selection and crop cultivation and improper monitoring and tracking of croplands due to poor prediction of weather and soil conditions. Agriculture practice by the farmers is mainly categorized into these significant tasks as given in Figure 1.1, which require sufficient labor and time when executed manually.

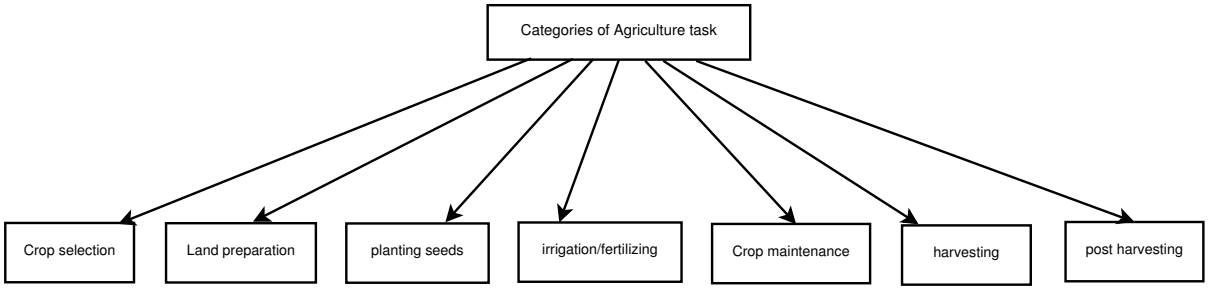


Figure 1.1: Categories of agriculture tasks

1.2 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

The software and hardware industries are being forced by automation, AI, and machine learning to create and introduce new products that can imitate the capabilities of the human mind. AI is virtually getting involved in every technology to make smarter decisions for different tasks and areas. AI assists humans in making correct decisions by digesting massive amounts of data. Automation can be called the core of AI since it helps in executing a large tedious task in less amount of time and thus, helping the workers to deviate their effort to more useful work. AI has a long list of benefits, such as it helps in making things automated, being competent in processing a huge amount of data in less time, and predicting output using trends and patterns [5]. AI is helping a large number of industries like retail, healthcare, finance, agriculture, education, insurance, etc. AI functions in four modes: assessment, selection, creating the platform and managing AI plans. Assessment deals with the assessment of strategy, skills and data, selection deals with the selection of software, platform and cost, the platform is created for processing the model, and last, will manage and modify the ongoing process.

1.2.1 Artificial Intelligence

Artificial intelligence is responsible for human-machine interaction that enables a machine to collect requests, perform computation and generate a conclusion. For example, online shopping from any merchant site recommends other products that were earlier purchased together. Thus, AI and Machine Learning are responsible for future prediction and classification based on the previous learning of data to make intelligent decisions as humans [6]. AI was initially introduced for military science and statistics to make computers more intelligent for equivalent reasoning and thinking like humans. Artificial intelligence prototypes the human brain's capabilities to learn and perceive patterns and process them for outcomes [7]. It develops a system that can perform tasks compared to human intelligence, such as perceiving visuals, decision making and language understanding. It is the process of building intelligent computers with human intelligence. AI has different sub-fields such as machine learning, natural language processing, computer vision and deep learning. AI allows the system to work smartly for real-world problems.

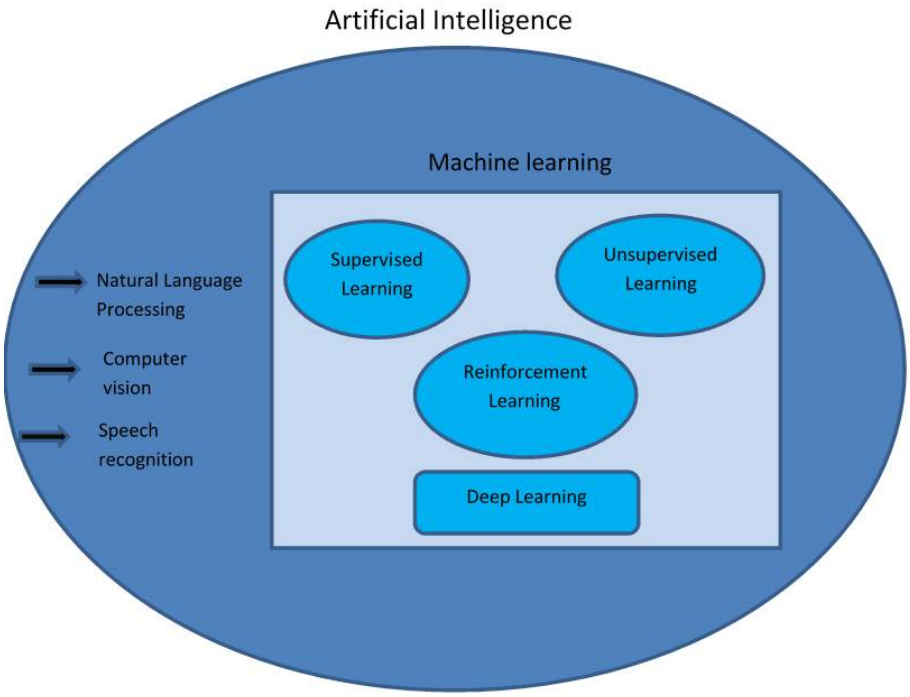


Figure 1.2: Correlation between Artificial Intelligence, Machine learning and Deep Learning

1.2.2 Machine Learning

Machine Learning, a subfield of AI as shown diagrammatically in Figure 1.2, is an experiential "learning" process that mimics human intelligence. It uses a computational algorithm to learn and update the analysis [8]. Machine learning involves a large set of data inputs effectively learned to recognize the patterns and train the model for effective decision-making. This iterative learning process modifies the model to accurately predict the output categories or clusters etc. In supervised learning, to verify the model's conclusion and forecast future data, the anticipated output is then contrasted with the actual output. "Deep Neural Networks" or Deep learning uses a complex hierarchical network of neurons consisting of several hidden layers. These several hidden layers help the neural network to implement any universal function and hence, can solve any practical problem. This makes the network deeper and more complex to categorize the input features into output classes. During deep learning, the focus is given to each neuron in hidden layers to get an accurate prediction. These predictive abilities of machine learning are being employed in various areas like health, agriculture, defense, robotics, etc. The base of machine learning is the idea of the development of a model, which derives significance from statistics and neural networks.

Machine learning employs various techniques to construct models that learn from data over several iterations to achieve better outcomes. These models can later be used for the prediction of unseen data. For a given model, machine learning is in fact, an effort to modify and adapt the actions of the computer through continuous learning to get more accuracy, which is verified by how far the chosen action matches the correct one [6]. Based on particular objectives and required activities, machine learning approaches can be divided into various types, as given below:

Supervised learning: In supervised learning, target labels are given to the training set to create a learned model that uses an algorithm to accurately respond to all of the incoming data [6]. The method is capable of dealing with noises which are the small inaccuracies in the data that can bring large variations in the actual results. Supervised learning can be further classified as regression and classification. While the regression method is based on continuous values, the classification process uses discrete data, where each data point corresponds to exactly one class.

Unsupervised learning: A method based on training data without target labels is used to prepare the model. Using the commonalities between the input data points, the data are divided into various groups [6].

Reinforcement learning: Reinforcement learning is learning between supervised and unsupervised, where the learning happens on the reward given on every right answer [6]. The method informs about the wrong answer but does not tell anything about how to correct the answer.

Evolutionary learning: Evolutionary learning studies Biological evolution. This illustrates the learning process of biological organisms that adapts to the changing environment to improve the method of evolution and survival [6]. It works on the model of survival of the fittest that will achieve a score based on how far the solution is correct.

The machine learning process is carried out through several phases. These phases are listed as:

Data Collection and Preparation: Depending upon the problem and area, the data can be collected from the standard repositories or developed and assembled from scratch. The collected data may contain several features irrelevant to the output. The entire collected data is analyzed and prepared. Initially, the collected data sometimes contain

noises, missing data and errors. Supervised learning requires the data to have target labels; an additional task called labeling of data. Deep learning, in particular, requires an enormous amount of clean data because the noise could affect the correctness of the results. The size of the data also increases the computational cost.

Feature Selection: Finding valuable features and excluding useless ones is a crucial responsibility for the machine learning process. Irrelevant features may sometimes degrade the accuracy rather than improve it. The feature extraction process also lowers the computational time and improves the computational speed after removing unnecessary features.

Algorithm: This step is the identification of an appropriate algorithm to design a model as per the requirement and area of the problem.

Model selection and parameter tuning: The data and algorithm combine to form a model with a repetitive learning process. The model's parameters are adjusted to create the best model possible for a particular issue.

Training: Training is learning the input data based on the algorithm to produce a model that could be employed to predict the output. This process requires the usage of computational resources.

Evaluation: The model training is followed by the model evaluation where the model is tested and cross-validated on the data, on which it was not trained earlier. The variation between the predicted and the actual results are identified and compared.

Several well-known machine learning methods, including Nave Bayes, KNN, SVM, Random Forest and ANN, have been used largely for classification and prediction tasks. While kNN is a density-based classifier that establishes the conventional nearest classes, Naive Bayes utilizes the Bayes theorem, which establishes conditional probability. The SVM technique is used for high dimensional data based on kernel functions; Random Forest is a classification technique that combines multiple weak classifiers. ANN is the neural network technique that classifies the data based on learning through different hidden layers. Deep Learning is one of the sub-field of artificial intelligence based on neural networks that combine many hidden layers to enhance the training and learning of neurons for various complex patterns to improve outcomes for a large amount of data. Computer Vision is the other sub-field of AI, as shown in Figure 1.2, responsible for deriving valuable and meaningful information from images, video, and any other

visual information that is further deployed for other computation. Natural Language Processing is another sub-field of AI that allows the computer to understand text and speech produced by humans for better communication between machines and man.

1.3 AUTOMATION OF DIFFERENT AGRICULTURE PRACTICES FOR IMPROVED PRODUCTIVITY AND GROWTH

Smart farming or agriculture automation involves new technologies and innovations in the traditional farming approach for the overall growth in production and quality of crops. For example, farming practices are incorporating drones and robots for agriculture automation. These agribots and agritractors could be employed for different labor-intensive tasks in farming, such as harvesting, watering, seeding, etc., built-in with different light detection and GPS systems. Various industries are working on the production of tools for farm automation. Agriculture automation also eases the process of monitoring and analysis of farmlands using satellite images.
In general, agriculture automation is increasingly getting adopted due to the reasons like:

- Agriculture automation is helping to quickly meet the consumer demands for fresh food supply, which in turn saves time and money.
- Agriculture automation is also helpful in managing the labor shortage by performing several labor operations.
- Agriculture automation helps maintain an eco-friendly environment.

Agriculture practices are divided into three parts: pre-harvesting, harvesting and post-harvesting. AI and machine learning are progressively being deployed to automate all these three farming processes. The automation practices in agriculture combine the knowledge and experience of farmers with new technologies and tools. This automation will help to lead to a sustainable environment for agricultural growth with the advancement of technologies and software. Consumer choices and priorities are diverting towards organic and sustainable products. Agriculture automation makes it possible while produce and deliver products much more fresh and faster to consumers. This will eventually increase the yield and production, thereby reducing the cost to

consumers. Artificial Intelligence is progressively being deployed for agriculture automation in crop production, disease identification, efficient monitoring of farmlands, product quality management with reduced cost and efforts, cropland segmentation, crop selection, etc. Machine Learning, Deep Learning, Computer Vision and Data Science are the most commonly used approaches under AI for agriculture automation. Machine Learning is progressively being applied to different site-specific agricultural problems. It helps to manipulate multidimensional non-linear data through analytical approaches with reduced cost and effort. Advancement in Machine Learning helps in the growth of agriculture by automating different processes that provide the inside details of vegetation. These advancement reduces human manpower and increases agriculture productivity through different predictive and monitoring techniques. These techniques help in acquiring the data, processing the data, extracting the features and classifying the data values. The shifting of taste from artificial to natural food is raising the demand for farm-based crops. Also, the focus on the urban lifestyle leads to a shortage of trained labor. Moreover, the eco-friendly approach to agriculture productivity avoids the usage of pesticides which allows for agriculture automation. Farm automation will be better. Thus, the automation of agriculture could lead to better yield prediction of crops based on weather conditions and also help in suitable crop selection and cultivation.

Machine learning techniques can be employed to learn different farming factors, like temperature, soil, weather data, water usage, etc., to determine the correct time and condition to plant seeds and identify the correct crop choices. AI systems play a significant role in precision agriculture, improving crops' quality and quantity while harvesting. They are helping in identifying the diseases in plants and crops and factors responsible for the poor nutrition of farms. AI-enabled sensors can detect weeds and then decides on the herbicide to apply to the detected region, thus, helpful in reducing the cost and herbicide wastage. Various bots have been developed using AI for chemical sprays on crops. One of the farmers' most significant concerns is the crop price fluctuation that hampers farmers' efficiency in planning a particular production pattern. These issues are getting resolved with AI and Machine learning that helps real-time farmlands monitoring using satellite images and weather forecasts. These technologies can help the farmers and government forecast future price trends of a type of crop sown in a particular season that could produce maximum benefit. Various startups in the field of

agriculture are getting established using the technologies of AI and machine learning. Moreover, technologies like computer vision, artificial intelligence, machine learning and mechatronics are progressively developing remote sensing methods. These methods will overall monitor the issues of farming like weeds, diseases, plants, soil, water, etc. and avoid wastage of herbicides and water, thus, improving the quality and production. Particularly, AI and machine learning possess various applications in the agriculture sector, these are:

Crop gene selection: Machine learning could be employed to develop a probability model that can identify the specific gene that could survive a particular climatic condition. The method will make better use of water and nutrients available in a particular soil and make it less susceptible to diseases. Thus, producing a beneficial trait of a plant.

Species Detection: Machine learning could be deployed to identify and detect the plant species by analyzing a large number of features of leaves rather than just color and shape as was done earlier in the manual recognition process [9].

Soil management: Machine learning can be deployed to understand the activities of the ecosystem that could help identify the insights of climate change [10]. This will help them in the process of soil management.

Water Management: Machine learning possesses the strength to collect and predict daily, weekly or monthly precipitation, temperature and rainfall for building better weather forecasting and irrigation system [11].

Yield Prediction: Machine learning, AI, and computer vision could be employed for precision agriculture which deals with crop yield mapping and estimation, maintaining and managing the supply and demand of cultivated crops [12].

Crop Quality: Machine learning and AI have the potential to identify and classify crop quality, which helps in managing the crop price and quality while reducing wastage [13].

Disease and weed Detection: For the early detection and identification of plant diseases and the identification of weeds and their separation from crops, machine learning and computer vision could be employed [14]. This will aid in regulating crop quality and productivity. Also, they are helpful for the proper deployment of pesticides and herbicides as per the requirement of the infected disease to avoid wastage.

Livestock production: Machine learning is also helpful for predicting and estimating the parameters for the proper management of livestock, such as cattle and eggs produc-

tion [15].

Farmer helper: AI and machine learning are developing apps to help farmers to explain their issues through chats and talks with experts [16].

1.4 MOTIVATION FOR THE RESEARCH WORK

The agriculture sector is facing several challenges that need to be resolved for the welfare of rural people and the overall development of the Indian economy. These are:

- Data is increasing rapidly in every area but still lacks data analysis and prediction.
- Lack of data utilization for decision making.
- Pattern discovery needs to be exploited to meet market demand.
- The agriculture productivity per unit of land needs to be raised.
- Food security needs to be focused on gaining self-sufficiency in food grains which is possible by increasing productivity and yield.
- A particular technique of machine learning seems to be limited in providing accurate results to a particular dataset.

Despite the world's population growing, a substantial reduction in agricultural labor has been observed in recent years. In addition, the demand for sustainable and organic food brought new terminology into the farming industry. Technology advancements and improved farming methods are required to meet the rising demand for higher-quality crops in larger numbers. Agriculture sectors lack proper infrastructure and operation. The focus should be given to encouraging farmers to diversify to higher-value commodities and minimize consumer prices for the overall growth of the agriculture sector. There is a need to incorporate automation with agriculture for better productivity and economic growth. Automation needs to be carried out in agriculture because people's tastes are shifting from unhealthy fast and packed food to natural food, raising the demand for farm-based crops. Also, the focus on the urban lifestyle leads to a shortage of trained labor. Moreover, the eco-friendly approach to agriculture productivity avoids the usage of pesticides which allows for agriculture automation. Thus, the automation of agriculture could lead to better yield prediction of crops based on weather conditions and will

also help in suitable crop selection and cultivation.

The agriculture sector is shifting from traditional tools to modern practices implementing advanced tool usage. New techniques, such as smart farming, need to evolve to meet the challenges of the modern world. This process of involving new technologies to reform and automate the farming process is called farm automation which is required to ease the laborious and time-consuming task of traditional agricultural farming. This agricultural automation will eventually help the farmers to devote their saved time to increasing production. The innovation of drones and robots for farming has helped farmers to improve their production. New technologies are needed for irrigation, planting, analyzing and monitoring farmlands, harvesting, weeding, etc. The automation of farming practices leads to various advantages for agriculture as well as the industry:

- Addresses the growing needs of the expanding population, smart agricultural farming practices are required.
- Traditional farming practices were labor intensive that generally faced the labor shortage problem, which the farm could resolve by involving robots, drones, etc.
- The process of deforestation has also been reduced through smart farming by improving the cultivation process in the same space, ultimately helping the environment.

1.5 PROBLEM STATEMENT

In the majority of nations, agriculture is one of the industries with the highest employment rates. However, with the growing population, it is coming up with burning issues of land scarcity to satisfy the basic food requirements of the people. These issues will eventually burden farmers to grow more on less. However, traditional farming methods are not enough to manage this massive demand for food. This is propelling farmers and agro-industries to find new alternatives for raising production and reducing cost and wastage. The agriculture business sector, which lagged earlier for several reasons, has been significantly impacted by technological improvement in farming in recent years. This need is making the scope for Artificial Intelligence steadily emerge as an option for agriculture companies for their technological innovations. These AI-based tools will

help the farmers improve their efficiency and the standard and quantity of food production. The present research mainly focuses on the automation of different agriculture practices by proposing novel AI and machine learning techniques that could improve crops' and vegetation's performance and productivity. This automation will eventually help to raise the economic growth of the country.

1.6 RESEARCH OBJECTIVES

In the light of above problem definition, the following research objectives are identified:

- To identify strengths and weaknesses of existing popular Machine learning techniques through empirical analysis.
- To develop some robust Machine learning techniques for analysis and prediction by considering an appropriate number of parameters and other factors such as the multidimensional and non-linear separable nature of data.
- To evaluate, examine, and contrast the effectiveness of the suggested procedures with the techniques already in use.

In general, to achieve these objectives we have developed different models for prediction, classification and segmentation which will be discussed in detail in the following chapters.

1.7 RESEARCH METHODOLOGY

The research is being performed using a quantitative research methodology, where the data is collected and analyzed to explain or predict the control variables of interest. Applying quantitative research objective is to recognize and test the causal relationship among variables, perform prediction and classification and thereby develop a system that the wider population can universally adopt. This research work uses an experimental research approach and adopts a longitudinal time horizon for its completion. It focused on collecting, listing and measuring numerical and image data from a large sample space, where the data has been collected from observations, databases, documents and records. Once the data gets collected, data analysis is performed using descriptive and inferential analysis. The research has conducted various experiments on the collected data to derive

valid results. In most of the experiments, the method of probability sampling is adopted to select random samples from the sample space. Figure 1.3 provides a summary of the approach used for this research study. For various research proposals, the data has been collected from three sources: images, statistics and satellites. These data and the algorithms are used to develop models for classification, prediction and segmentation. The developed models then execute experiments and produce results with different hyper tuning, and later, these models are evaluated and validated on the unknown data.

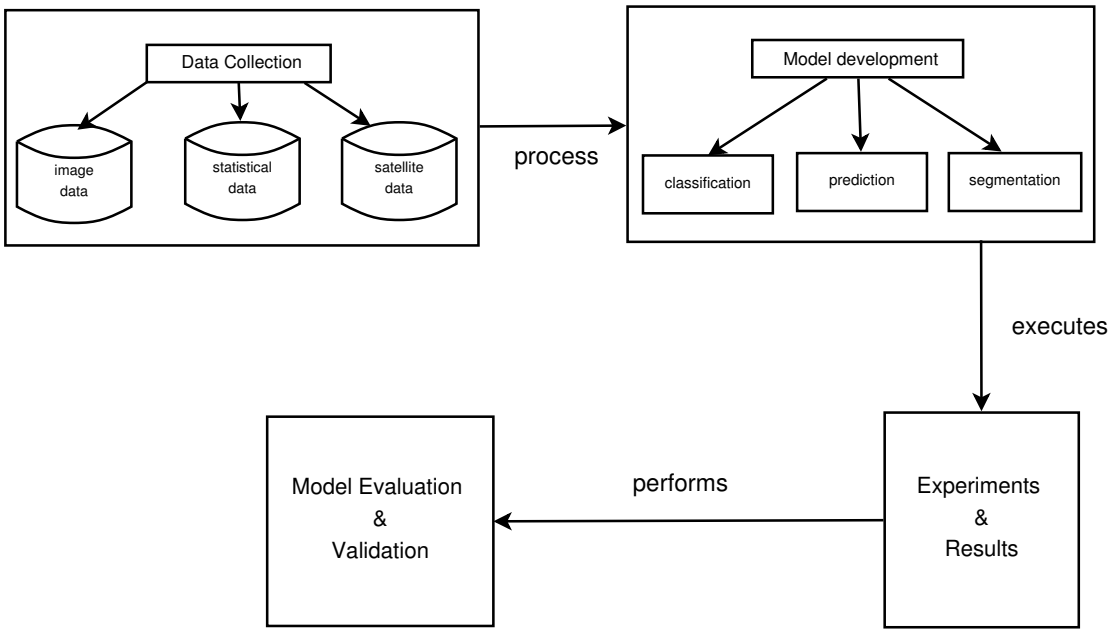


Figure 1.3: A block structure of the research work

1.8 SYSTEM CONFIGURATION

The research has been conducted on the system with:

Hardware configuration: Windows 10, 16 GB RAM, 64-bit operating system, x64-based processor and NVIDIA GeForce GTX 1650 4GB Graphics.

Software configuration: Python (Spyder 3.7), TensorFlow 2.8, OpenCV, Keras, Scikit-learn, NumPy and CuDNN libraries.

1.9 ORGANIZATION OF THE THESIS

The seven chapters that make up the arrangement and organization of the research effort covered in the thesis are briefly explained as follows:

i) **Chapter I** briefly explains the scope and subject of this research work. It summarizes the main highlights of the work. The chapter states the motivation and the problem statement of the research conducted. It also identifies the research objectives and presents a research methodology adopted for the work. The chapter concludes by specifying the system configuration on which the implementation was executed.

ii) **Chapter II** discuss the literature survey concerning different methodologies that have been adopted to automate the different agriculture practices. It gives a survey of several techniques employed by different researchers in the given research area. The chapter broadly defines the issues considered in three main categories: classification, prediction and segmentation, where every category is separately explained and defined. The chapter finally explains the research objectives in detail based on the entire literature survey and the gaps identified.

iii) **Chapter III** performs neural network-based pattern detection on three different datasets for two types of problems. This initially performs the disease identification based on the patterns and features of diseases to classify the type of disease on tomato and potato leaves using transfer learning. Based on the findings of the earlier models, a new model is proposed "Agri-CNN" for wheat grains classification to identify wheat varieties based on the patterns identified.

iv) **Chapter IV** emphasizes on the dimensionality reduction techniques. It makes a comparative study of the effectiveness of PCA and Entropy-based method for dimensionality reduction. The chapter also proposed a hybrid algorithm "Info_PCA" for dimensionality reduction, which is compared with the other methods and verified by the results.

v) **Chapter V** focused on crop yield prediction, another vital issue of agriculture automation. A hybrid algorithm, "RaNN", has been proposed to predict rice yield, which is compared with several other machine learning algorithms. The result section includes illustrations of how the proposed model has performed.

vi) **Chapter VI** deals with image processing to perform semantic segmentation of croplands. The chapter explains two different approaches to semantic segmenta-

tion. One is thresholding based approaches and the other is machine learning-based approaches. These techniques are used to identify the region of interest for cropland mapping. The work has also estimated the area covered under vegetation.

vii) **Chapter VII** concludes the thesis with the overall experiments and methods proposed under the research work for classification and prediction. Moreover, it also states the future work that could be conducted and performed shortly.

CHAPTER II

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

This chapter presents a thorough review of the literature on the most popular machine learning and deep learning models applied to solving diverse agricultural challenges.

2.1 MACHINE LEARNING

Machine Learning is a sub-field of Artificial Intelligence, used to create models that aid in emulating the visual reasoning and prediction capabilities of the human brain [17]. It utilizes the combination of hardware and software to analyze and visualize statistical and image data for prediction and classification and to provide decision-making capabilities. Machine learning has been applied to autonomous vehicles, healthcare, business intelligence, retail and agriculture. To produce results that are more precise and optimized, machine learning undertakes in-depth learning sometimes through thousands of iterations [18]. It requires the training data as input along with the learning algorithms to develop a specific model. This developed model is then applied to predict the test data where the results are validated by identifying the variance with actual values (in the case of supervised learning). The larger the variance between the predicted and actual results, the more will be the error which is reduced by providing feedback to the learning algorithms to provide the necessary modifications in the developed model to get more accurate results. This entire process of Machine learning involving various steps is depicted in Figure 2.1.

Since Machine learning possesses the characteristic of learning patterns from the input data using an iterative process therefore we can conclude that the critical aspect of machine learning is the input data which may be quantitative and qualitative for accurate

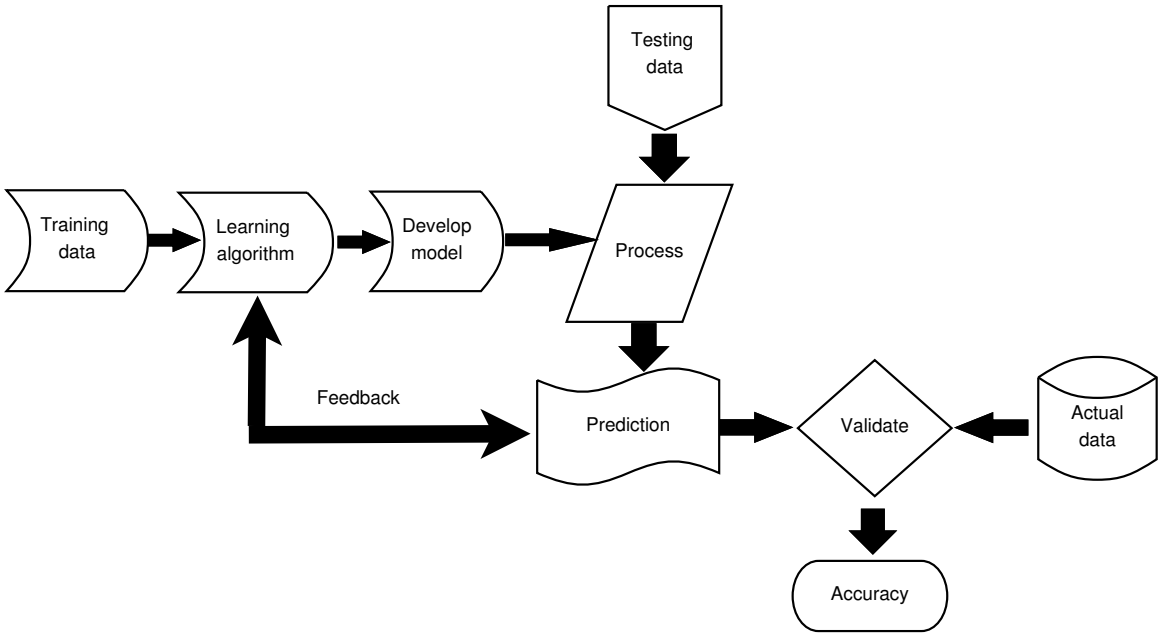


Figure 2.1: Steps of machine learning process

predictions. Normally, to build a model, the input data is split into training and test data. The training of the model is done through training data whereas the accuracy of the learning is determined through the test data. Machine learning is categorized into four different types of learning, which include:

Supervised learning: A model using supervised learning is trained on input data with target output, called labeled data. The generated model is then used/tested to get output (class or regression values). For instance, a classification model to accurately categorize the photos of cats and dogs would learn the characteristics or patterns of both classes from the labels provided through an iterative process.

Unsupervised learning: This type of learning is performed using non-labeled data. For instance, in clustering, the data points make the clusters using some kind of similarity. For example, the images of various animals like cats, dogs, cows and lions are grouped in several clusters depending on the features and patterns without using the target labels.

Semi-supervised learning: Both labeled and unlabelled data can be used by semi-supervised learning to build a model, with labeled data being used for model training whereas, unlabelled data are used for separating the classes by defining boundaries.

Reinforcement Learning: Reinforcement learning is based on situations and actions taken. A reward is mapped to every correct action or prediction while penalized for a wrong answer or action. The algorithm learns based on the maximization of rewards by

following various trials and errors.

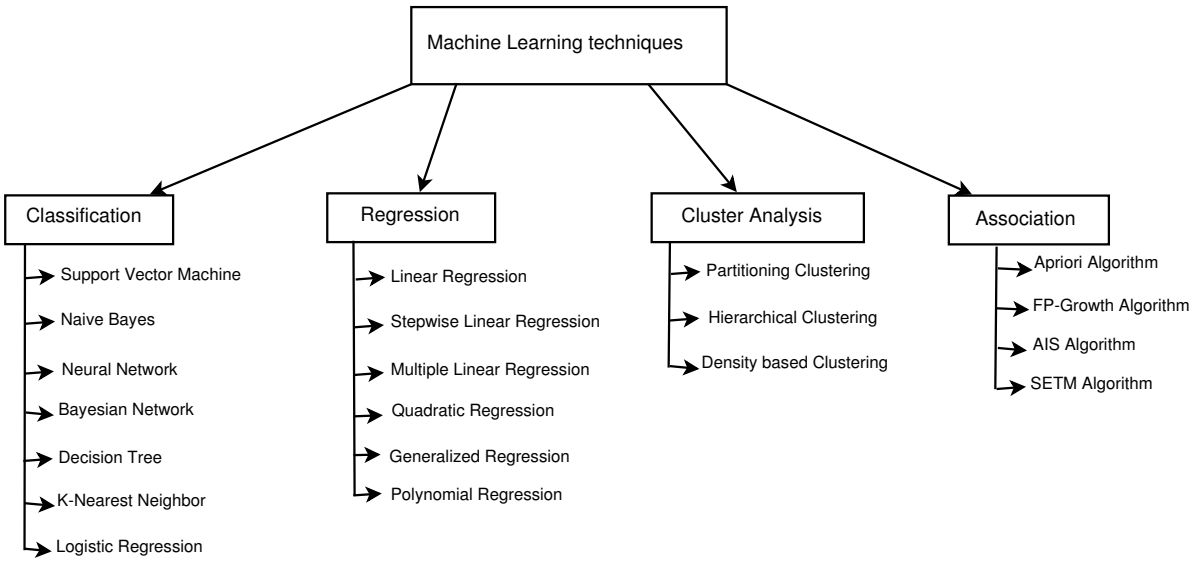


Figure 2.2: Various machine learning techniques

Machine Learning-based predictive analysis is needed in different areas [19] such as Market analysis, Biosciences, E-commerce data analysis, Weather prediction, Customer service management, Disease prediction, Agriculture data analysis, Medical data analysis, etc. Machine learning involves various techniques for solving different types of issues. These techniques are categorized as classification, regression, cluster analysis and association [18–20] as shown in Figure 2.2. A brief introduction of these areas is as follows:

Classification: Classification is to predict the category or class of data or items. It is a supervised approach [21] where the target label is already set, and the items need to be labeled and classified to the target class e.g., classification of soil for any particular crop as good, bad and average.

Regression: Regression is again a supervised technique where the task is to predict a numerical value for the item rather than class e.g., the amount of rainfall in a particular year.

Cluster analysis: Cluster analysis is to organize the items into different groups as per the similarity between items. It is an unsupervised approach [14] where the items are unclassified, and there is no target label and prior knowledge e.g., clustering of states of India based upon different crop cultivation.

Association: Creating rules for discovering patterns and associations between various things in a database is another unsupervised approach known as an association.

Today every aspect of technology involves some part of machine learning concepts. From the viewpoint of Artificial Intelligence, Machine learning was developed to introduce intelligence in the machine, initially, using data mining techniques. As we have already stated, AI performs the machine learning operation that learns the patterns from history to adapt to new instances and data with the correct results. AI, through Computer Vision, helps visualize and perceive objects while moving and managing things through Robotics. The various areas of computer science like Artificial Intelligence, Statistics, Data Mining, Pattern Recognition, Computer Vision and Deep Learning are closely associated with Machine learning. Their correlation is shown through the Venn diagram as shown in Figure 2.3.

Brief introduction to all these disciplines is given in the coming subsections.

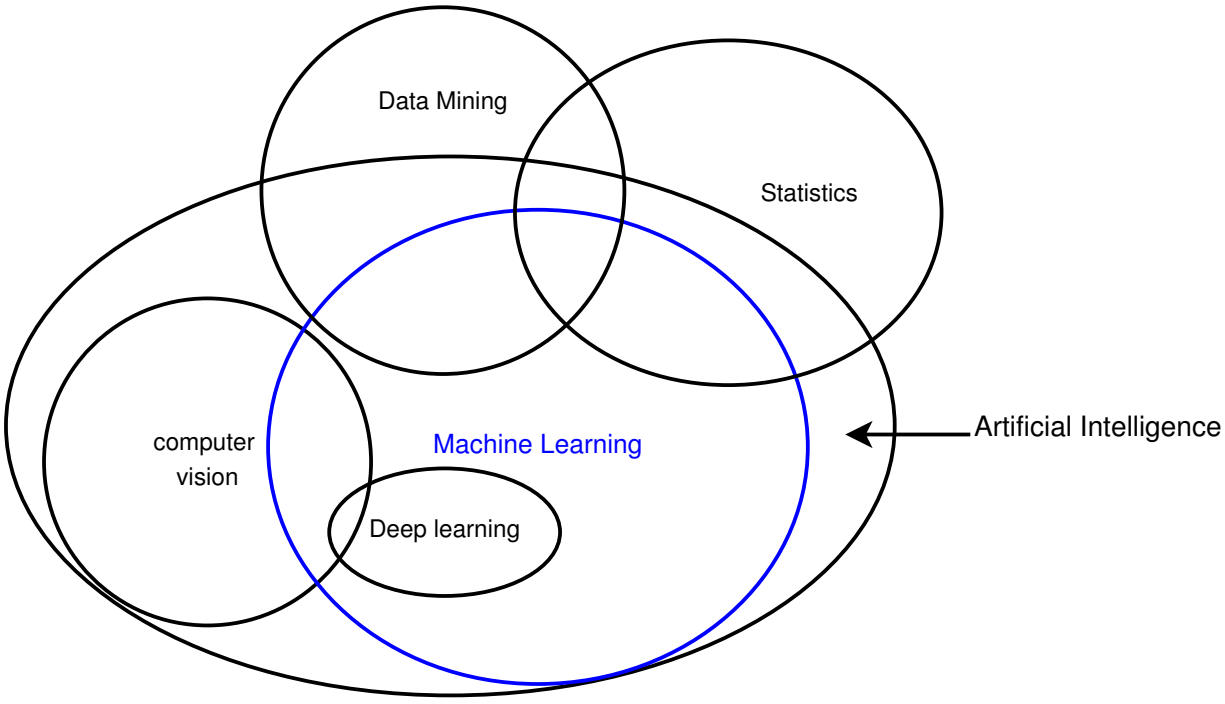


Figure 2.3: Correlation of Machine Learning with other related branches of Computer Science

2.1.1 Artificial Intelligence

The objective of developing AI is to model human intelligence [7] and to develop intelligent artifacts. It will comprise human abilities like reasoning, speech and language processing, visual and audio grasping, learning capabilities, planning and a problem-solving approach.

2.1.2 Deep Learning

Deep Learning [22] is a subfield of machine learning which allows the computer to learn with experience and data with a significant deal of power and flexibility. It represents a learned nested hierarchy of concepts through various fully connected hidden layers of feedforward back propagation neural network. These nested hierarchies of concepts or the connected hidden layers are capable of mapping any universal functions. Automatic feature extraction is a key property of deep learning. As per Arel et al. [23], deep learning emphasizes data representation rather than a task. Deep learning is widely applied to self-driving cars, fraud detection, speech recognition, E-commerce, Natural language processing, supercomputers, customer relationship management, healthcare for disease prediction and diagnosis, etc. Deep learning techniques are in demand due to their high-performance computing ability achieved through the implementation of deep neural networks where the hidden layer extracts some set of features and passes it to the next layer [24]. The earliest layers are in charge of extracting the low-level features, those are integrated with the features taken from the upper levels to provide the whole sequence of features. Its design philosophy is inspired by the fact that a complex pattern is a combination of simpler ones. For example, the image of a person is formed with simple concepts of contour and corners, which are further formed with the edges. As per Lecun et al. [25], Deep learning extracts the properties of AI and machine learning based on how humans understand things and extract knowledge. It's a part of data science that comprises statistics and predictive modelling [26]. It automates the prediction task by learning through the hierarchy of rising complexity and abstraction, unlike machine learning, which is a linear process [27]. For example, repeatedly learning the features of humans and animals through many images in several iterations enabled the machine to automatically classify the unknown image into human or animal categories. These days Deep learning is very popular as deep learning models need a large number of training

data (both organized and unstructured), as well as processing capacity, which is now accessible due to the growth of big data and computing platforms like cloud computing. Strong models based on deep learning could be created by adjusting different parameters as given below:

Learning rate: Learning rate sets a condition for the speed of learning to develop a model. Too high a learning rate makes the system unstable, while too low a learning rate increases the training time. Thus, the learning rate during training needs to be adjusted for the correct model.

Transfer learning: The practice of leveraging previously trained models obtained on a common dataset for another classification process is called transfer learning. The trained model is trained on new data set till the desired performance is achieved. This helps in performing new tasks with different output classes or the same as the previous one to reduce the overall learning time.

Dropout: By randomly removing some of the neurons and their associated connections from the network can handle the problem of overfitting during learning for some of the applications such as speech recognition and document classification.

Hidden Layers: The model, based on a neural network, can be trained with different hidden layers to check for the accuracy of the output. Convolution, ReLU, max-pooling, and dropout are added before the hidden layer to boost its effectiveness.

Number of Neurons in a hidden layer: The number of neurons in a hidden layer, is another parameter that can be adjusted for better output. It is assumed that a hidden layer with more neurons, creates a stronger model.

Activation Functions: The activation functions are the functions to control the behavior of the neurons in a layer for a given set of inputs by activating the relevant pathways and suppressing the irrelevant pathways in the network.

Several variants of deep neural networks are available like multilayer perceptron, convolutional neural networks, Radial Basis Function Networks (RBFNs), Deep Belief Networks (DBNs), Long Short Term Memory Networks (LSTMs), Autoencoders, recurrent neural networks, etc. All of which are needed for various specialized applications. As the unstructured data is increasing in large volume and with the introduction of big data analytics, the requirement for deep neural networks is rising sharply. Deep learning methods include image recognition, chatbox formation, natural language processing,

healthcare, satellite object detection, industrial automation, agriculture automation and computer vision. Despite, many benefits, deep learning also has certain limitations:

- requirement of a large amount of data for better accuracy
- issue of bias in data
- results may get affected by the learning rate
- requirement of GPU-based system for larger processing

2.1.3 Computer Vision

The component of AI known as computer vision [28]−allows a computer to process object images and videos like that of a human vision system. Computer Vision, combined with AI, Deep learning and huge data, can train and compute complex problems related to images and videos, such as detecting and labeling objects. Computer Vision [29] deals with pattern recognition to identify the patterns available in images or videos. The process allows understanding the video in terms of the collection of labeled images learned through different iterations to identify the patterns.

As per Kurtulmucs et al. [30], Computer vision extracts meaningful knowledge and information from visual data or signals. Computer vision has various applications, like object recognition, object detection, object verification, object segmentation, automatic face recognition, gesture identification, visual information of vehicles, medical image analysis, robotics designing for medical and agriculture, optical character recognition, formation of intelligent offices, object tracking, biometric identification, self-driving cars, augmented and mixed reality, healthcare, agriculture, collision detection and many others. In general, computer vision is an AI-based system that develops a visual machine to solve several AI problems and visually represent the various aspects of world issues. These visual signals get converted into symbolic representations to detect the patterns in the data. With the dramatic increase in the number of photos on the web, computer vision has various applications, including:

Special effects: By recording and creating the shapes and motions of characters that humans play as in the movie 'avatar', computer vision can be used to create animated films.

3D urban modeling: Computer vision can be used on drones to capture the 3D model

of a location and also helps in combining various pictures in one frame.

Scene recognition: Computer vision can be applied to identify a picture location, where a location is compared to several other pictures to identify the best match.

Facial features detection: Computer Vision is being progressively used for face detection like in cameras, smile detection that will capture the image only when a person smile, and face recognition as done by social networking sites. Computer vision has also been used in the iris and biometric identification.

Optical Character Recognition: Computer vision has been used to recognize characters and numbers, such as by reading zip codes and number plates.

Visual search: Computer Vision can be applied to search an image on google using mobile.

Self-driving cars: Computer vision is being applied to develop fully autonomous cars such as self-driving cars from Tesla.

Augmented Reality and Virtual reality: Computer vision is being applied for augmented reality and virtual reality using algorithms to detect the position of a person or any other surrounding object. It needs to identify and update the position as the user moves to show the dynamic and realistic view.

Satellite Monitoring: Computer Vision applies high-performance models to extract information from large satellite images to generate better prediction and accuracy concerning environmental changes, urban and rural population, poverty, etc.

Control System: Computer Vision is being employed for Unmanned Aerial Vehicles (UAVs). It provides the ability to detect moving and non-moving objects for tracking and targeting.

Cosmology Study: Computer Vision is also used for Cosmological study to extract the features from various astronomical and cosmological objects.

2.1.4 Data Mining

Data mining is the process of digging through a vast amount of data to find interesting patterns and knowledge. Databases, data warehouses, the internet and other dynamic data repositories can all be used as data sources. Data Mining is one of the fields of Data Science for finding out the frequent important and unknown patterns from databases for efficient decision-making [20, 31]. Contrarily, machine learning allows the machine to

learn the patterns of the collected data to develop models based on algorithms to make predictions for future data [17]. Data Mining needs human intervention for processing, while most techniques in Machine Learning are self-learned by developing models using learning algorithms.

2.1.5 Statistics

Statistics is another area related to computer science. Some of the concepts or methods such as mean, mode, standard deviations, variance, and co-variance are applied in some of the machine learning techniques. Statistics concludes the data sample while Machine learning identifies the patterns out of data for generalized prediction [32].

2.1.6 Pattern Recognition

Pattern recognition is the field of computer science that deals with identifying the patterns within data and matching the extracted information with the stored data of the memory. It applies the machine learning algorithms to extract the patterns [33].

2.2 MACHINE LEARNING AND AGRICULTURE

Modern agriculture faces various challenges like food scarcity with population increase, changes in pattern and type of food consumption, depletion of natural resources, change in climate and issues related to health and safety. These issues place an overburden on the agriculture sector, which should be reduced by improvising agricultural practices such as the introduction of precision agriculture. The digital evolution of agriculture opens the scope for artificial intelligence to extract meaningful data from a large volume of information generated from numerous sources. This will focus on global issues like optimal use of natural resources, development of automatic services, ecosystem conservation and sensor deployment. Vegetation, livestock management, water and soil are the four divisions of agricultural management. Crop management focuses on crop identification, yield projection, disease and weed detection and quality monitoring of crops. Water management focuses on the optimal use of water resources, soil management works for soil monitoring and protection and Livestock management works for animal welfare and livestock production. Knowledge-based farming systems can address many

issues owing to machine learning. Machine learning is utilizing its exponential computational power to resolve various issues in agriculture [6, 34]. Machine learning is actively working on different agricultural issues, like species recognition, soil and water monitoring, yield forecasting, quality monitoring of crops, crop management, disease identification, cattle welfare, etc. SVM, ANN and Deep learning are the most popularly used techniques of machine learning. ANN utilizes the concept of the human brain can be used for computing complex functions like pattern recognition, object identification and decision making. These techniques are employed for classification and regression tasks of crop management, disease and weed identification and classification. To categorize data and forecast crop quality and yield, a linear separating hyperplane is produced by the binary classifier SVM. Various sensors have been established to capture the entire farmlands and agriculture through satellite images. AI-enabled farms are getting developed using machine learning techniques to produce a knowledge-based agricultural system that will, in turn, enhance agricultural production and quality. Machine Learning techniques are being employed for different Agricultural issues. The Classification, Prediction and Segmentation methodologies have been extensively studied in this chapter to identify problems related to different agricultural issues discussed earlier and it is decided that the research work, in the following chapters will focus on the problems, which are broadly categorized into three categories, as given below:

- Classification
- Prediction
- Segmentation

Therefore, in the following subsections, we are providing a review of the earlier research work related to each of the problem areas.

2.2.1 Review of Classification techniques

Classification of data is the primary task performed through Machine Learning techniques. However, data classification through Machine Learning techniques requires feature extraction [35] that raises the processing time and complexity. The introduction of deep learning overcomes this issue by automatic feature extraction followed by learning through various hidden layers [23, 24, 26] to examine a huge volume of data [22].

Machine Learning and Deep Learning are being utilized in agriculture for various classification issues, including the disease categorization of leaves, stems, fruits and grains, among others, in conjunction with computer vision. India is a leading cultivator of various grains and pulses, which are also exported to other countries. However, the lack of tools and techniques causes the cultivation of grains in massive amounts without the demand and quality requirement of a particular variety in the market. This leads to a considerable wastage of cost and production. The food industries and sectors require various grains for different purposes like noodle production, bread production, etc. [1] which requires proper grain cultivation and classification. The correct grain classification will lower the production cost while improving the quality; on the other hand [36]. Earlier, the experts manually executed the classification processes by grasping the visual features, which was considered a laborious and time taking process [37, 38]. Machine Learning is used to develop appropriate methods to help automate the grain classification process. Computer vision is also essential for extracting, identifying and categorizing visual elements from images [7,39,40]. The blending of Machine learning and computer vision is developing high computational power for different agriculture-related issues that is overall reducing the production cost [41]. Several Machine Learning classifiers for grain classifications have been considered in the literature. Some of these are briefly explained below:

Through image analysis and textual feature extraction utilizing Gray Level Co-occurrence Matrix and Linear Binary patterns, E. Olcay et al. [37] suggested a method to automatically identify the grain varieties farmed in Turkey where k-NN classifiers are used to perform the classification. This technique has classified the wheat varieties grown in Turkey. The adopted statistical technique GLCM uses the spatial relationship of pixels to give the intensity variation at a particular pixel point. Another method adopted for texture analysis is LBP which calculates the thresholding of the pixel neighborhood to label the particular pixel of an image as a binary number. The method is robust for monotonic grey level changes due to different lighting conditions, making it suitable for real-time settings. Different k values are specified, and the results are assessed while lightning is uncontrolled to determine the classification accuracy of k-NN.

Applying image analysis and sparse representation-based classification, T. Kuo et al. [38] applied a locally developed SRC classifier to identify 30 types of rice grains. The method

makes advantage of the morphological, visual and texture characteristics of grain, sterile lemma and brush. The SRC technique is a machine learning approach utilized in this study for high-dimensional problems. It encodes training data features as atoms in an encoded dictionary. Each fresh sample of data is coded as a sparse collection of atoms, and the class with the fewest coding errors is given the assignment. The entire work of the said paper is divided into three parts- developing a microscopic imaging system to capture grain images, extraction of features like morphological, color and texture from the grain body and constructing a locally constrained SRC classifier to classify different varieties of rice grain.

To determine the species and subspecies of Acacia seeds, V. Shivkumar et al. [9] does the classification. Based on the physical features of seeds, such as size and shape, categorization is carried out using discriminant analysis. Considering the physical characteristics such as 2D surface area, aspect ratio, perimeter, fullness ratio, roundness, length and width, the seeds were analyzed. It concludes that species are discriminated against based on seed size and shape. Stepwise discriminant analysis was used to differentiate the species.

Iran's rice grains were classified into five classes by A. Pazoki et al. [42] using morphological and chromatic characteristics. Four shape features, 11 morphological features and 24 color features that were produced from rice grain images were used in the classification procedure. Once these properties have been recovered, multilayer perceptron and neuro-fuzzy neural networks are utilized to classify the rice grains. The MLP model's structure consists of an input layer with 39 neurons, two hidden layers and a resultant layer with five neurons, whereas the neuro-fuzzy neural network's structure is identical but has 60 rules added to it. This paper has compared the accuracy of MLP and neuro-fuzzy neural network before and after feature selection for classifying rice grains. The paper has also identified the best set of external features responsible for classification.

Using a neuro-fuzzy inference system, K. Sabanci [43] has used a computer vision-based approach to categorize wheat grains into drum and bread classes. The image classification is performed on five dimension features, three color features and five texture features extracted by processing the image. The model is evaluated on 20 grains after being trained on 180 grain's visual characteristics.

Using a predictive model, F. Kurtulmus et al. [30] have classified rapeseed varieties using an ideal collection of attributes discovered with various feature sets, models and learning classifiers. The classification was based on characteristics like texture and color. To minimize overfitting, the initial 420 features were condensed to just 29 features. The best feature extraction methods were found to be SVM classification, stepwise discriminant analysis and recursive feature removal. To prevent overfitting, grid-search and k-fold cross-validation are exercised for model validation.

K. Sabanci et al. [36] have applied a feed-forward backpropagation ANN model to categorize wheat grains into bread and drum varieties by extracting an optimal set of textual and color features. Image processing is performed with visual features including four-dimensional, three-color and five-texture features to reproduce 21 features. These selected attributes are passed to the ANN classification model to train on 180 samples and tested on 20 samples. A total of seven features are identified which are affecting the model accuracy.

P. Zapotoczny [4] proposed a method to classify eleven wheat grain varieties by adopting an optimal set of textual and color features. The full features were reduced to 49 using genetic and class ranker algorithms. These features are then subjected to multidimensional analysis and classification using Bayes, lazy, meta, decision tree and discriminatory analyses.

The majority of the literature has a common issue in that they suffer from the limitation of the manual feature extraction process that requires expert knowledge. These methods also fall short in their ability to distinguish between nearly similar characteristics of two distinct variations of a single species, producing inaccurate classification accuracy scores. Convolutional Neural networks, a most potent technique of Deep Learning, rapidly get involved in several agricultural issues such as disease classification of plants, vegetation identification, monitoring of agricultural lands, etc. [44–48]

CNN is also used for identifying plant diseases in conjunction with transfer learning [49, 50]. A TL DCNN model was suggested by R. Thangaraj et al. [50] to identify illnesses in tomato leaves. It has detected and classified ten different tomato plant diseases from the images. It has opted for adam, stochastic gradient descent and RMsprop optimizer for disease classification, where adam optimizer performs best. The paper has used transfer learning to give better results on the unbalanced dataset. It identifies that

the Modified-Xception model with adam optimizer performs better than other optimizers. Results were assessed using precision, recall, and F1 scores.

M. Brahimi et al. [24] have improved performance by applying deep learning models over machine learning for plant disease classification. The model performs automatic feature extraction, unlike machine learning. It has made use of transfer learning to perform the classification of a larger dataset. The diseases are detected by focusing on the localization of the infected region by automatic extraction of features.

A. K. Rangarajan et al. [51] performed the classification of seven tomato leaf diseases using transfer learning models such as AlexNet and VGG16 net on the data taken from the PlantVillage dataset. VGG16 and AlexNet's classification precision were initially tested on 13,262 pictures; later, the models were tuned by changing the number of images, learning rate, weights, bias and batch size. It has been observed that a change in performance by changing the number of images is distinctly visible. The paper has compared the performance of two different transfer learning models, which have shown different results on different parameter tuning.

In addition, several additional articles have demonstrated how deep learning models can classify plant diseases. Disease categorization has also been performed using KNN and C5.0 [52]. Tomato leaf disease has been classified using DenseNet-121 with the application of C-GAN (Conditional Generative Adversarial Network) that produces synthetic images [53]. Plant disease categorization has also made use of machine learning methods including K-NN, SVM and Naive Bayes [54]. CNN has employed various transfer learning like MobileNet [55], SVM [27], EfficientNet [54], etc., for plant disease classification. A big dataset, required for categorization in the deep learning process, is present during the transfer learning phase. To retain the level of accuracy even when using smaller datasets, the transfer learning method swaps out the resultant layer used in the standard dataset with the output classes of the user dataset [48, 50, 53, 56–60].

A summarized comparative analysis of various techniques employed for classification is also illustrated in Table 2.1, mentioning the techniques, their derived accuracy on different datasets, and their strength and weakness. Thus, the properties of CNN of learning with deep layers and automatic feature extraction could be utilized for various classification problems related to agriculture.

Table 2.1: Comparison of methods used for classification of crops

Paper	No. of Classes	Varieties	No. of Images	Classification techniques	Accuracy	Strength	Weakness
[36]	2	Wheat	200	ANN	99.9%	dimension, color, texture feature;Use Otsu and GLCM method	limited dataset; feature detection
[38]	30	Rice	1500	SRC	89.1%	morphological, color, texture; k-means for hue-saturation	mis- classification; dictionary for each variety
[61]	3	Soyabean	866	SVM (Gaussian kernel)	89.57%	analysis of vein morphological;	discard shape, texture, color; expensive & time consuming
		Red bean		SVM (linear kernel)	89.77%	adaptive threshold	
		White bean		Random forests	87.87%		
				PDA	89.97%		

Continued on next page

Table 2.1 – *Continued from previous page*

Paper	No. of Classes	Varieties	No. of Images	Classification techniques	Accuracy	Strength	Weakness
[9]	8	Acacia	200	Discriminant Analysis	79.6%	morphological traits; statistical & discriminant analysis	limited data; feature extraction
[42]	5	Rice		Multi-layer perceptron, Neuro-fuzzy neural networks	98.40% 99.73%	use of histogram; low time and cost	limited dataset; only color & morphological feature
[43]	2	Wheat	200	Adaptive neuro-fuzzy inference system (ANFIS)	99.46%	color, texture feature	limited dataset; manual feature extraction
[30]	7	Rapeseed	525	SVM,KNN, SGD KNN SGD	100% 92.40% 94.30%	color, texture feature; SVM classifier	limited data size; linear dependent features

Continued on next page

Table 2.1 – *Continued from previous page*

Paper	**No. of Classes**	**Varieties**	**No. of Images**	**Classification techniques**	**Accuracy**	**Strength**	**Weakness**
[62]	9	Tomato	14828	CNN, AlexNet, GoogLeNet, InceptionV3	99.18%	localization of infected region; automatic feature extraction	large computation time; inappropriate for small dataset
[50]	14	Tomato	54306	D-CNN	99.5%	Automatic feature extraction; pre-trained weights	complex structure; large computational time
[49]	14	Tomato	50000	CNN,VGG16, InceptionV3, MobileNet	91%	deep architecture; automatic feature extraction	computational heavy; unbalanced data
[51]	7	Tomato	13262	AlexNet, VGG-16	91.2%	mini batch sizes; automatic feature extraction	complex model; not suitable for unbalanced data

2.2.2 Review of Prediction techniques

By repeatedly learning data that aids in better prediction and classification, artificial intelligence and machine learning are successfully automating the agriculture sector. The difficulties that farmers and consumers encountered in traditional agriculture can also be solved by AI and machine learning. It is helping in various agriculture issues like crop yield prediction [63, 64], prediction of climatic changes [65], soil condition monitoring [10], plant disease prediction [66], prediction of droughts [67], etc. By detecting the data features of global meteorological variables like moisture, wind speed, rainfall, precipitation, wind speed, temperature, etc., AI and machine learning are also helping with annual crop planning [11] for crop productivity improvement.

AI is helping agricultural farmers in a timely decision-making process for better yield and production and protection against crop-related issues through data analysis that will facilitate forecast and prediction [68]. This relationship between weather and crop yield needs to be identified for early warning and risk management. Various studies have been performed in the literature for agriculture data analysis and prediction. To maximize crop output and productivity, machine learning approaches are utilized to identify the best parameters through data analysis [69]. Crop yield prediction has been carried out on various types of crops such as rice, tea, cotton, maize and wheat through different machine learning techniques [63, 64, 70–72] such as Multiple Linear Regression [73], Artificial Neural Network [74–76], Random Forest [12], Hybrid MLR-ANN [77], Naive Bayes and Bayesian Network [78, 79] and Support Vector Machine [80]. Crop production generally gets influenced directly and indirectly by some weather conditions that provide noisy and biased information [81]. Several other methodologies are being used for crop yield prediction like prediction with Convolutional Neural Network [82] using image data [83], ecological distance algorithm [84], multidimensional model [85], satellite-based SIF models [86], etc. Machine learning models assist in identifying the variables influencing agricultural yield forecasts like various natural resources covering soil pH, chemical composition, water pH [87], moisture, rainfall [88], cropland conditions, climatic conditions and social factors [72, 89, 90]. Some of the papers on crop production and yield estimation are discussed below:

Majumdar et al. [91] have analyzed the data from agriculture to identify the optimal parameters for improving crop production. Five different crops have been worked on:

cotton, wheat, groundnuts, jowar and rice. It has used five mining techniques for data, including PAM, CLARA, DBSCAN, K-NN and multiple linear regression, to find the ideal parameters and important factors for forecasting the yield. The author applied the modified DBSCAN to cluster districts with similar temperatures while PAM and CLARA clustered the districts based on maximum crop production. Later multiple linear regression was applied to estimate the crop yield. The analysis is limited to 5 crops and two parameters which are rainfall and temperature. The techniques are compared based on precision, recall and rand index results.

N. Gandhi et al. [92] used data visualization and association rule mining to identify the impact of seasonal rainfall distribution on the rice yield of Rajasthan. The rainfall from June to November of the Kharif crop production season is divided into three different sets of months: the first is June-July, the second is Aug-Sep, and the last is Oct-Nov. The impact of rainfall differences in the different sets of months is visualized to identify some patterns. The limitation of the work is the less amount of parameters as rainfall is one of the external factors; there may be other factors that may largely influence the rice crop production.

S. Veenadhari et al. [93] created a system named "Crop Advisor" to provide predictions about how climatic elements will affect crop yield. It is a user-friendly web page that would estimate the crop yield before harvesting to help farmers and policymakers act accordingly. The research does not cover the other input factors like irrigation, fertilizers, pesticides, etc., because of their dependency on the respective field. However, they are also one of the significant factors in yield prediction.

To determine whether a crop can be successfully grown in a specific location given the availability of natural resources such as soil pH, moisture, nitrogen, phosphorus, potassium and water pH, A. Anitha et al. [87] proposed a hybrid approach that relies on a rough set based on fuzzy approximation space and ANN. The input layer of ANN is fed with pre-processed data to identify the missing values class and forecast crop suitability. This process is time-consuming.

To give a significant qualitative and quantitative data analysis and prediction, W. W. Guo et al. [69] presented a hybrid strategy that uses the benefit of statistical analysis and neural network while obscuring both approaches' shortcomings. The significant features that contributed to the prediction were found using the statistical method. The

Multilayer perceptron model then uses these identified features for prediction, where the accuracy is checked through mean absolute error. The research has considered only three parameters for wheat yield prediction, but there may be other parameters that may have also substantially impacted the yield prediction.

R.D. Baruah et al. [94] suggested a method to predict tea yield using data mining techniques in different regions of Assam and Cachar based on climatic variations. The author applied the Multiple Linear Regression technique for yield prediction, which generates a linear relationship to link dependent and independent features, thereby removing the non-significant attributes based on the p-test. This work can be extended to include two significant tea yield parameters: soil and irrigation.

S.Hira et al. [85] developed a multidimensional data model, which performs the data correlation using multidimensional and statistical analysis. These co-related data are then analyzed using association rule mining. These all applied techniques help in extracting useful information from the model. The work was based on multiple parameters of the agriculture dataset gathered from the World Bank website.

For predicting the yield of crops like wheat, maize and cotton, A.Shastry et al. [95] used a variety of regression techniques, including quadratic, pure-quadratic, interactions and polynomial. Based on various metrics values, like Root Mean Square Error (RMSE), R^2 and Mean Percentage Prediction error, an ideal model for yield prediction (MPPE) was identified. It also provides a gap to find a correlation between variables important for prediction.

J. Pant et al. [63] has developed a trained model for forecasting four crop yields using machine learning techniques. The crops considered were maize, potatoes, rice and wheat, and the data was chosen from FAOSTAT and the World Bank. To predict agricultural yield, four machine learning methods—a gradient boosting regressor, a random forest regressor, SVM and decision tree regressor [63]—were used and compared. The better technique was found with the R-squared value. The best prediction accuracy was identified for the potato crop by decision tree regressor.

Y. Liu et al. [73] performed the crop yield prediction using stepwise linear regression. It has collected a large dataset of 38 years covering 31 provinces of china with agricultural, economic and environmental impacting the Spatio-temporal variants of various yields. The paper has identified the correlation between the independent factors and the

combined yield of different crops. It has analyzed the space-time patterns of yield and built an empirical model to identify the yield using stepwise linear regression and also analyzed the temporal and spatial patterns of regression residuals [96].

N.R. Prasad et al. [12] applied random forest, a machine learning model to predict the cotton yield in Maharashtra at three different times before the harvest. To anticipate the production at the last of September, December and February, the agrometeorological and spectral variables were gathered from the satellites with multi-sensor capabilities. The co-linearity between the independent and dependent variables is analyzed and calculated, which is validated through with Random forest model. These features are selected using the CART decision tree and the recursive feature elimination method.

A machine learning approach for forecasting agricultural yield was put forth by H. Khan et al. [80]. The data was collected for the state of Chhattisgarh, covering crop production and rainfall. The data initially undergoes feature extraction and selection through the dimensionality reduction techniques such as PCA and k-medoid clustering to remove the features that hamper the prediction accuracy. The final crop yield estimation was performed with the support vector machine technique.

E. Khosla et al. [74] have performed the prediction of a significant Kharif crop in Visakhapatnam. It initially performed the prediction of the degree of rainfall in the district during monsoon weather through a modular artificial neural network. This is followed by important feature selection influencing the prediction. Finally, the crop yield prediction is executed with factors like area and rainfall using Support vector regression. The paper has also compared the results of the applied technique with the other machine learning techniques. The crops chosen for prediction were bajra, maize, rice and ragi.

Besides this, various other papers have also proposed the prediction techniques and applied machine learning methods for crop productivity prediction. A. Haghverdi et al. [75] forecast the Tennessee cotton lint yield using remote sensing technologies and ANN to perform the prediction. To estimate the BLC and JNE genotypes of two different maize strains, D. Seka et al., used two probabilistic methods: Gaussian Naive Bayes and Logistic Regression [78]. Based on the accuracy and AUC curve, the findings are contrasted. Using UAVs and remote sensing, P. Nevuori et al. [82] employed CNN to predict agricultural productivity. The NDVI and RGB data are passed to the CNN

model, which consists of six convolutional layers and L2 regularization. The accuracy is estimated based on MAE and MAPE.

A summarized comparative analysis of various techniques employed for prediction is also illustrated in Table 2.2, mentioning the techniques, their derived accuracy on different datasets, and their strength and weakness.

Table 2.2: Comparison of feature reduction and prediction algorithms

Paper	Data	Time	No. of features	Techniques	Accuracy	Strength	Weakness
[63]	Maize, Potato, Rice, wheat	26 years	7	boosting regression, random forest, SVR, decision tree	96%	feature importance is identified through probability	less techniques compared, small no.of features
[71]	wheat	29 years	1	grid based simulation, point based simulation, statistical regressions	95%	quantify method and model uncertainty	single factor; complex structure
[73]	12 crops	38 years	27	stepwise multiple linear regression	95%	focus on spatial patterns and temporal trends; identically distributed Guassian variable	important explana-tory variables are excluded; consider only linear relation-ship

Continued on next page

Table 2.2 – *Continued from previous page*

Paper	Data	Time	No. of features	Techniques	Accuracy	Strength	Weakness
[12]	cotton	16 years	8	Random forest	68%	use long term agromet-spectral variables	chances of overfitting; less data leads to over or under estmations
[74]	rice, ragi, maize, bajra	5 years	3	support vector regression; ANN	85%	prediction performed in two steps	only five years data is considered; small set of features
[75]	cotton lint	2 years	4	ANN	86%	focus on spatial variability; remotely sensed crop indices	limited features; discard texture similarity
[78]	maize	1 year	2	Gaussian Naive Bayes, Logistic regression	87%	prediction of two genotypes; prediction based on parents phenotypic traits	limited independent traits; results with different subsets differ

Continued on next page

Table 2.2 – *Continued from previous page*

Paper	Data	Time	No. of features	Techniques	Accuracy	Strength	Weakness
[79]	rice	4 year	7	Bayes Net, Naive Bayes	97%	model uncertainity of parameters; probabilistic relationship among variables	less no. of features; shows only casual relationship
[82]	wheat, barley	1 year	2	CNN	95%	large spatial scales; remote sensing based	focus only linear relationship; minimum pre-processing
[95]	wheat, maize, cotton	10 years	9	quadratic, pure-quadratic interaction, polynomial	80%	linear relationship;	limited set of features; limited data
[97]	-	-	1003	CNN	90.2%	compared before and after optimization	application after extraction takes long

Continued on next page

Table 2.2 – *Continued from previous page*

Paper	Data	Time	No. of features	Techniques	Accuracy	Strength	Weakness
[8]	Grid world	-		Relevance Assignation Feature Selection	90%	mutual information theory; feature importance with redundancy	more calculation time; complex structure

2.2.3 Review of Segmentation techniques

Semantic image segmentation, a process of computer vision, divides the object into several sub-objects and labels them with different categories to extract the semantic information. Appropriate, suitable and accurate mapping of cropland boundaries for long distances is a complex process for agricultural monitoring yet needed for food security concerns. The already available global datasets related to croplands lack spatial resolution information for small-area distribution, which obstructs their processing at regional and local scales. However, with the rise of high-resolution images, the mapping and extraction of croplands from heterogeneous and scattered landscapes become easier. The increase in population and urbanization is resulting in the reduction of available productive agricultural land, which needs to be effectively utilized and managed [98]. Better yield prediction and improved food systems are possible with proper cropland mapping. With the evolution of remote sensing, machine learning and computer vision and the availability of high-resolution imagery from Google satellites, agricultural processes are becoming more automated. Therefore, to meet the population's growing demand for food, the identification of uncultivated and cultivated croplands is essential [13, 99]. For the implementation of sustainable agricultural practices and routine crop monitoring, appropriate systems for mapping farmland are required. The image segmentation process is helping in the identification of such abandoned agricultural lands [100, 101]. The image segmentation process separates the object of interest from the background

based on homogeneity and similarity. This segmentation is performed by splitting an image into nonoverlapping regions [102]. Machine Learning is employed to analyze the different cropland layers and learn the different high and low-level features that help in grasping the edge information of object boundaries. Image segmentation algorithms are being applied for cropland mapping and agriculture land identification. To lift the living standard of people and for the global security of food, a reliable and sustainable cropland mapping [103–105] is needed for the identification of crop yield [106–108]. Cropland mapping can be performed with land cover image segmentation, which is helpful for crop type mapping [109–111]. The cropland mapping has applications in crop yield prediction, area estimation of agriculture, soil quality estimation and monitoring, crop damage monitoring [108,112,113] and controls over the agriculture practices [114]. The process of cropland mapping to identify the regions under agricultural farms requires a computer vision model for appropriate monitoring and management. Cropland mapping has been performed in the literature using different methods. Some of the papers are discussed below:

A deeply constructed CNN, which can automatically learn the distinctive properties, was used by D. Zhang et al. [115] to map croplands. For better prediction at a higher resolution using the modified pyramid scene parsing network, the author combined the expanded covering characteristics with local shadow features (PSPNet). Concerning four regions of China with various agro-systems, it has gathered high spatial resolution satellite pictures from multiple sources. The edge information relating to item boundaries was covered by local features, whereas long-range features gathered the long spatial dependencies to identify farmland patches. This collection helps to identify precise boundaries and develop a better classification model. The outputs are compared with the traditional and other methods for cropland mapping.

M.F.A. Vogels [116] presented a semi-automatic method for cropland mapping using B&W photography. The study was conducted in Ethiopia and Netherlands. The dataset contains features like texture, spectral and neighbor information, shape and slope. Geographic image analysis and random forest classifiers based on objects are used to evaluate and categorize these attributes. This method helps to map the historical cropland expansion using the brightness values in the B&W photography. The method has adopted slope as a covariate for cropland mapping and classification.

To develop an urban functional zone mapping system using high-resolution remote sensing photos, S. Du et al. [117] employed multi-scale segmentation and conditional random field. A multi-scale semantic segmentation network with pixel-wised prediction was proposed [117] to use the multi-scale contextual data to predict urban functions for geographic objects. A conditional random field is then used to organize the objects into UFZs and create the final UFZ map [117].

M. Belgiu et al. [118] have classified various crop varieties from three distinct regions—România, Italy, and the USA—using both pixel-based and object-based methods. It has used the Sentinel-2 time series to apply a dynamic temporal warping model weighted on time for categorization. Results of object-based and pixel classification are contrasted with those of the Random forest classifier. On training samples, it was found that TWDTW based on objects gave better outcomes than TWDTW based on pixels.

The classification process varies from Semantic segmentation in classifying the entire image to one of the given classes. In contrast, the latter classifies every image pixel to the given labels of an image which is needed for the different problems of artificial intelligence. The semantic segmentation process has been applied to different areas such as disaster management [119], object detection [120], agriculture [14, 121, 122], semantic text segmentation [123], etc. Semantic segmentation can be used for cropland boundary identification [124, 125].

J. Adrian et al. have implemented the 3D U-Net, a deep neural network that makes use of the spectral and texture data from Sentinel-1's multi-temporal SAR data [109]. This integration was made to correct the classification results of different crop types. The results are compared to SegNet, 2D U-Net and Random Forest.

M. Belgiu et al. [103] presented Spatio-temporal image fusion methods related to remote sensing. These are reconstruction-based, learning-based and unmixing-based methods [103]. These methods blend the spare-resolution images into coarse-resolution images. It mainly emphasizes the fusion of microwave data from remote sensing to optical data under different environmental conditions collected from the different sensors. The fusion is performed by CNN, capable enough to fuse different spectral values. The reconstruction-based method for fusion had shown the best results.

The multi-task semantic segmentation was done by D. Jiang et al. [103]. This was accomplished by utilizing RGB-D image data based on the faster-RCNN model [103] in

a complicated interior setting. The model is capable of many simultaneous identifications, including semantic indoor scene segmentation, target classification, and detection of numerous visual tasks [103]. It increases the model's effectiveness and the information provided by the fusion picture feature information. It has applied a model for the fusion of color and depth of the image.

A. Agapiou et al. [126] performed the classification of colorized images using a greyscale satellite to reflect land cover. The study used CORONA satellite images from Larnaca city in Cyprus. The CORONA photos were first colored using a deep learning technique, and then the colored CORONA images were compared to the original greyscale images using a different quality measure method. Finally, the SVM classifier is used to classify the photos into five basic land covers: cultivation, land region, water, lake and urban areas.

On the multi-band input dataset compiled from individual and multi-sensor layers from ASTER, Landsat 8 OLI, Sentinel-1, and Sentinel-2A [127], S. Shayeganpour et al. [127] performed an object-based categorization. The data consist of medium-resolution imagery. The classification was executed by machine learning techniques like k-NN, SVM, Naive Bayes and Random forest, whose results are evaluated and compared. The RF and SVM algorithms have shown maximum accuracy.

The MFANet that accomplished deep feature extraction, as well as the fusing of upsampling features, was proposed by B. Chen et al. [128]. To maximize the effectiveness of the learned knowledge, its Channel feature compression module extracts the important features and eliminates the superfluous channel data [128]. The Multi-level feature aggregation up sample module [128] then completes its task, which is utilized to position the restoration of high-resolution remote sensing images. The guiding principle behind this technique is that high-level features direct low-level features. The high-resolution feature maps that have been restored are then refined using the Channel ladder refinement module.

With the rise in satellite data, these images could be utilized for identifying the precise view of land covers that ultimately could be employed for purposes like agriculture monitoring [109], damages in land cover regions [104, 129, 130], cropland monitoring, agriculture area identification and estimation, etc. Image segmentation and classification have been performed using various techniques and algorithms like ANN, SVM, Random

Forest, CNN, Autoencoders, etc. Random Forest is used to segmenting the input images with the learning of filters through an iterative optimization algorithm [131] that could be employed for agricultural land classification [132,133]. With the aid of a local homogeneity method that will capture the discriminative spectral-spatial properties, SVM is employed to recognize the information about land cover [120]. SVM and object-based image analysis is being employed for land cover classification [117, 126, 127]. Image Segmentation is being performed for cropland mapping [134], to find the region of interest where these techniques employ pixel-wise semantic segmentation.

A summarized comparative analysis of various techniques employed for segmentation is illustrated in Table 2.3 also, mentioning the techniques, their derived accuracy on different datasets, and their strength and weakness.

Table 2.3: Comparison of methods for semantic segmentation of land covers

Article	**Algorithm**	**Dataset**	**Maximum Accuracy**	**Strength**	**Weakness**
[131]	Random Forest	Road scene; Indore scene	93%	represent weights, shapes and sparsity; iterative optimization algorithm	restricted performance; less learning
[135]	CNN	satellite images (building, roads, agricul-ture products)	95%	use both spectral & spatial information; pixel based classification	complex model; more computa-tional time and space

Continued on next page

Table 2.3 – *Continued from previous page*

Article	**Algorithm**	**Dataset**	**Maximum Accuracy**	**Strength**	**Weakness**
[136]	Semantic Texton Forest frame-work	CamVid and MSRC-v2	93%	class specific image segmentation; exploit com-plementary features	accuracy differ with different frequencies; limitation with large dataset
[127]	RF, SVM, Naive Bayes, k-NN	land covers	90%	object based image analysis; multi spectral satellite images	complex structure; restricted geographical settings
[126]	SVM	land covers (CORONA images)	85%	greyscale satellite sources; vector machine classification vector machine classification	no focus on texture analysis; false positive classification
[128]	MFANet	land cover	86%	deep feature extraction; upsampling feature fusion	insufficient feature extraction; unextracted low level features

Continued on next page

Table 2.3 – *Continued from previous page*

Article	Algorithm	Dataset	Maximum Accuracy	Strength	Weakness
[137]	FCN, DCNN	land cover	88%	comparison of DL with ML; utilizes object surrounding information	low accuracy for small data; disregard label of objects
[134]	ANN	wetland	89%	automate complex wetland mapping; correlation between the spectral values	weak with images with different spatial information; disregard contextual information
[138]	histogram based approach	Disease region	98%	histogram & validating algorithms; use of mutual information	manual retrieval of features; not appropriate for all regions
[139]	LDA classifier	Agro-ecological zones (landset images)	85%	model of one location can be applied to other regions; decrease between scene variability	not cover all pixel points of all regions; different climatic factors affects the classification algorithm

2.3 RESEARCH OBJECTIVES REVISTED

Utilizing AI, Machine Learning, and Deep Learning algorithms to create a machine that could automate the activities to bring technological improvement to agriculture. Furthermore, appropriate algorithms with various parameters based on the characteristics of the data are employed to enhance performance. The aim of the thesis is to provide machine learning analytical and predictive methods for the agricultural sector. The limitations of the individual method of AI and machine learning for a domain give the scope of developing hybrid and novel techniques for data analysis and the prediction that should possibly cover more parameters and uncertainties inherently available in a dataset efficiently. This research has mainly focussed on some of the issues of agriculture automation taking some of the crops under consideration, briefly illustrated in Figure 2.4. These issues are broadly divided into classification, prediction, and segmentation, as shown at Level 1 of the tree and as given earlier. The Classification category of level 1 further emphasized some specific issues as shown in Level 2, like grain and disease classification, which will further research on specific grain classifications like wheat and specific disease classification like tomato and potato. Similarly, the prediction category of level 1 further focuses on dimensionality reduction, as shown at level 2, which will improve the prediction process like the rice crop yield. Likewise, the segmentation category of level 1 is utilized to identify the area under the cropland. It has applied threshold-based and machine learning-based segmentation techniques, shown at level 2.

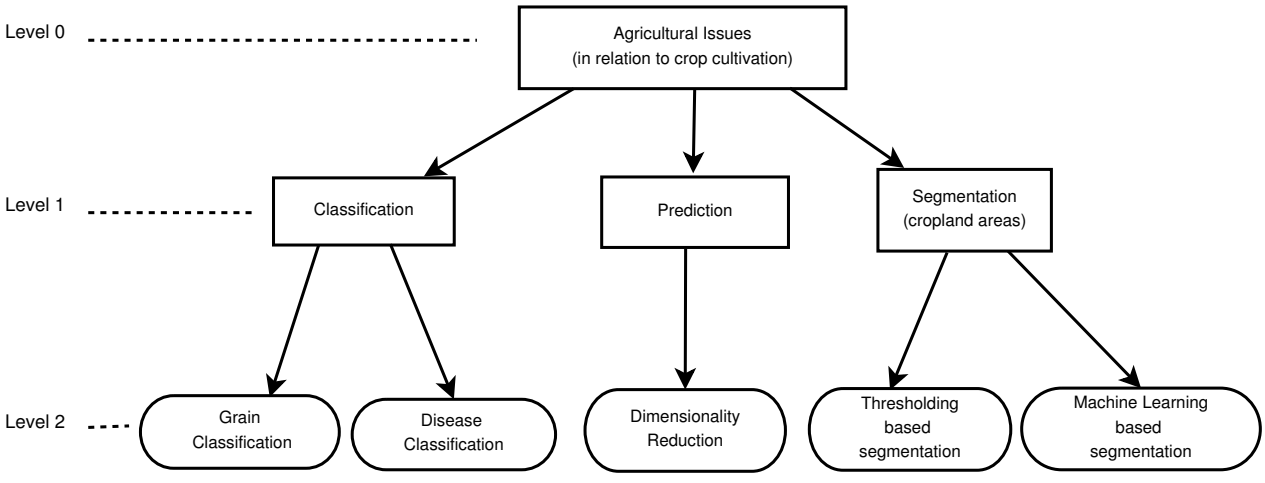


Figure 2.4: The problems and domain considered for the research work

The literature study under these issues has identified the different gaps and limitations of the related work. Thus, we have designed and categorized the objectives of our research work, considering the earlier gaps and limitations, broadly into three points:

- To identify strengths and weaknesses of existing popular Machine learning techniques through empirical analysis.
- To develop some robust Machine learning techniques for analysis and prediction by considering an appropriate number of parameters and other factors such as the multidimensional and non-linear separable nature of data.
- To assess, examine and compare the effectiveness of the suggested procedures with the techniques already in use.

The research's aforementioned goals and how they are met is being briefly discussed as follows:

i) **To identify strengths and weaknesses of existing popular Machine learning techniques through empirical analysis**

To accomplish this goal, we have produced a thorough literature review that highlights the advantages and disadvantages of several machine learning algorithms used in the relevant research work. Several techniques related to analytical and prediction have been chosen and discarded based on their strength and weakness adopted for the different agricultural issues. It has been determined and demonstrated that deep learning methods, such as CNN, are more effective at resolving classification-related problems than conventional Machine Learning techniques based on labor-intensive manual feature extraction procedures. Thus, to increase the classification accuracy, a new model based on CNN has been proposed. The literature review also noted that the classification accuracy gets improved by reducing the dimensions in case of original data having too many features. This research has also studied and focused on different dimensionality reduction techniques, which are compared considering the limitations of the available techniques. A hybrid model to reduce dimensionality has also been proposed. In addition to this, the research work has thoroughly studied different prediction algorithms based on statistical data, identified their limitations, and developed a hybrid algorithm for

predicting crop yield. This research has also studied an important agriculture issue which is agricultural cropland mapping. This mapping is performed to identify the area covered under the cultivation by applying segmentation techniques on the land which can be cultivated or barren.

ii) **To develop some robust Machine learning techniques for analysis and prediction by considering appropriate number of parameters and other factors such as multidimensional and non-linear separable nature of data**

The research has been performed on multidimensional and non-linear separable agriculture data related to various varieties of wheat grains to achieve the objective. Various transfer learning models are adopted to check the effectiveness of the model and validate the results for plant disease and grain classification. To fill the gap, a model named "Agri-CNN" has been proposed for wheat grains classification. The proposed model is hyper-tuned to identify the best parameters for classifying multidimensional data to attain better accuracy and results. When a dataset has numerous parameters and features, overfitting can lead to inaccurate conclusions. Thus, the number of important parameters (particularly dropout) is estimated and identified for better classification results by applying a novel dimensionality reduction technique called "Info_PCA". The result obtained from the developed hybrid model is compared with the conventional feature selection and extraction methods, and it is found that the proposed model is giving better results. A novel machine learning algorithm, "RaNN", has been developed to predict crop yield that outperforms the earlier machine learning algorithms for multidimensional non-linear data. This research has also considered several Machine learning algorithms related to the semantic segmentation of land covers using satellite imagery. The semantic segmentation has also been carried out using another approach wherein different thresholding-based techniques are applied. The results from these two methodologies are then contrasted and examined further. The segmented areas under the croplands are also estimated to identify the region covered under agricultural cultivation.

iii) **To evaluate, analyze and compare the performance of the proposed techniques with the existing techniques**

All of the proposed methods in this research work are implemented and compared with the techniques already available by using certain metrics such as precision, recall and F1-score. Moreover, using the confusion matrix, the results of the classification are cross-validated. The proposed prediction model, which is based on statistical data, has been assessed using the Mean Absolute Error, Root Mean Square Error and Coefficient of Determination (R^2). Histograms are used to test and validate the techniques that are being researched for the semantic segmentation of farmland images. To examine and compare various machine learning techniques for semantic segmentation, the measures like Training accuracy, Test accuracy, Mean IoU and Kappa Score are used.

2.4 CHAPTER SUMMARY

To classify, predict, and segment statistical data and images about diverse agricultural challenges, this chapter gives a survey of several machine learning and deep learning algorithms. The survey identified the limitations of the earlier techniques involved in various fields, it is found that there is scope for developing a certain hybrid and novel technique for data analysis and the prediction that should cover more parameters and uncertainties inherently available in a dataset efficiently.

The literature study has revealed that the classification and identification of large varieties of grains and crop diseases were conducted based on hand-crafted features and hence they were less accurate. Particularly, these earlier works have shown that crop classification suffers from small and unbalanced data limitations, manual feature extraction, complex structure, and significant computational time. Consequently, a classification model has been put forth in this research work to overcome the aforementioned restrictions.

The study also demonstrated that the preceding methods were ineffective in separating and identifying the closely related features of two varieties of a single species. Thus, a hybrid model is proposed for the dimensionality reduction of data to improve the result of prediction. The research also focused on the limitation of the prediction techniques for statistical data due to the large set of irrelevant features that may degrade the accuracy of an algorithm. In light of the shortcomings of prediction algorithms, a hybrid model for agricultural yield prediction has been developed in this thesis. The model has demonstrated improved and more accurate output based on machine learning

techniques.

The literature study has also focused on another issue related to the segmentation of croplands which is used to identify the cultivated and uncultivated regions. The earlier work experienced difficulty in categorizing texture features and false-positive classification. Thus, the two semantic segmentation approaches based on image pixels have been studied for cropland mapping. The two segmentation approaches based on thresholding and machine learning have been applied to identify the croplands. The result obtained is used for cropland area estimation. The model has successfully identified the region of interest, despite the region's complexity due to similar texture and orientation.

In the coming chapters of this thesis, the architecture and design of proposed techniques related to different agriculture issues are presented with a detailed description.

CHAPTER III

PROPOSED NEURAL NETWORK BASED PATTERN DETECTION TECHNIQUES FOR CLASSIFICATION

CHAPTER 3

PROPOSED NEURAL NETWORK BASED PATTERN DETECTION TECHNIQUES FOR CLASSIFICATION

This chapter has initially, demonstrated the classification of crop diseases using transfer learning. Later, considering the performance of the initial process a novel "Agri-CNN" model based on CNN has been proposed to better classify the plant diseases along with the classification of grain varieties.

3.1 INTRODUCTION

The agriculture sector of India is a producer of different types of grains consumed by most of the Indian population and exported to other countries, such as wheat, rice, barley, maize, etc. The possibility of mismanagement, deterioration in quality, and an increase in production costs may affect the production and availability of grains regardless of the market's requirements for quality and variety [1, 36]. An accurate grain classification is required for the proper supply of grains to different end users for their consumption and production. Earlier, the people manually carried out this analysis and classification process, which requires a lot of labor, time and expertise [37, 38]. Hence, the correct classification of grains needs to be automated to save labor costs and time.

Moreover, certain crops are severely damaged by diseases which results in low production and financial trouble for the farmers [67]. Therefore, the problem of detection and categorization of plant diseases with the help of machines is one of the important actions to be taken for the automation of agriculture-related activities.

In general, the identification and classification is a pattern recognition task requiring

Deep Learning and Computer Vision for the solution [140]. Due to their tremendous computing capacity and low processing costs, Deep Learning and Computer Vision are combined to handle agricultural problems involving images [41]. ANN is the foundation of Deep learning with many hidden layers to discover patterns across many iterations on a massive amount of data [22]. Deep learning is favored over machine learning [141] because of a deep network of hidden layers that automatically extract even minute sets of features from large sets of images [23, 24, 26]. Since Deep Learning [142] is showing a promising solution to several agricultural problems, therefore it has been applied in this work to create a model that can identify and categorize different grain types and plant diseases. Hyper-parameters of CNN are tuned to produce a more accurate model. In this chapter, mainly two works are presented. One related to the plant disease classification using tomato and potato datasets was chosen from the Plant Village dataset. This classification task used the approach of transfer learning. The objective of this work is not only restricted to just developing a classification system to detect diseases of plants but also to study the effectiveness of the transfer learning model used on small datasets. Once the effectiveness of the models is determined, the most effective model is used to classify the wheat grain varieties. It gets clear from the result that transfer learning-based systems are not so effective for classification where the training dataset is small. It is also found from the study that out of all the transfer models, the model based on VGG-16 performs well. To overcome the issue of small data size, a self-generated dataset of the wheat grain is developed where the data size is 15000 images covering 15 varieties of wheat. The transfer learning is also performed on this self-generated dataset by the most effective transfer model called VGG-16. A model called Agri-CNN is also developed from scratch for the classification of wheat grain variety and the result of this proposed model is contrasted with VGG-16 centered on transfer learning, and it has been found that the proposed model (Agri-CNN) outperforms the transfer model VGG-16 even on large datasets.

The two primary approaches, mentioned above, employ Convolutional Neural Network and Transfer learning as their base technology or approach. Therefore, the coming sections, first introduce the basic architecture and functioning of CNN followed by the introduction of transfer learning.

3.2 CONVOLUTIONAL NEURAL NETWORK

A deep neural network made up of several layers followed by a fully connected feed-forward neural network is called a convolutional neural network [53]. It identifies the correct classes of input data irrespective of their close similarities. CNN is one of the deeply engineered neural networks which can process structured data arrays like photos and videos. Normally, it starts with an input image, applies weights and biases, and runs the information through several triads of layers called convolutional, ReLU and Maxpooling layers. The hidden attributes in the input data are extracted with the help of these layers. These extracted attributes are later passed to a fully connected back-propagation neural network, consisting of several hidden layers (sometimes up to 30 or more [120]) to generate the output classes [59]. CNN works efficiently in identifying patterns like lines, gradients, circles, eyes and faces from input images. They are also capable of performing on raw images as well, even without pre-processing. A typical CNN is shown in Figure 3.1.

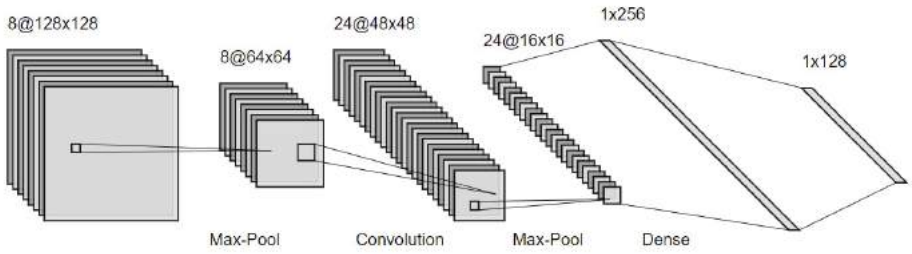


Figure 3.1: Diagrammatic Visualization of CNN

The primary power of CNN, comes with an initial convolutional layer. These layers can be many, which are stacked one on top of each other, where every layer is capable of identifying more features. The organization of CNN is similar to that of the human visual cortex, where several layers or neurons process an image and identify several complicated elements to determine an image's pattern [143]. The multi-layered design in a sequential fashion allows for extracting the hierarchal features. Convolutional layers make up some hidden levels, followed by activation and pooling layers. A convolutional neural network is most commonly adopted for images due to its advantages:

- elimination of manual feature extraction process, as CNN automatically learns the

features.

- results are much more accurate.
- can be applied in the form of transfer learning to serve as a pre-trained model for a new task.

3.2.1 Applications of CNN

CNN is being employed for various applications such as:

Medical Imaging: CNN architecture is being used in medical disease diagnosis. It helps in the visual detection of various infected cells in images.

Audio Processing: CNN can be used for audio processing like keyword detection adopted in android and Windows phones, where a device with a microphone detects a certain word or phrase. CNN can more accurately perform this task by dropping all the words that occurred in the sequence.

Object Detection: CNN is capable of detecting objects. It can identify and detect a particular object from images or videos such as YOLO.

Artificial image generator: CNN can be used to construct an image from the previous image using autoencoders and corrections.

Facial Recognition: CNN has been employed for recognizing different facial structures using different features like eyes, nose, mouth, cheeks, etc. to identify a person. This is needed in different security areas.

Plant disease detection: CNN can be employed for plant disease detection by feeding the collected pictures of plant diseases to the developed model to get the results.

Land cover classification: CNN can be employed for land cover classification with the use of remote sensing data to identify various areas like grassland, cultivated lands, agricultural crop areas, wetlands, etc.

Crop type classification: CNN is being used for the classification of various crops grown across a region based on various textures and colored features to assure food security and sustainable agricultural development.

Weeds identification: CNN can be employed to classify weeds among the crop species based on their shape and color features to avoid their impact on crop cultivation and growth.

3.2.2 Functioning of CNN

CNN acts as a feed-forward neural network that assesses visual images by processing data stored in a matrix of pixels. Convolutional Neural Network works through different layers. These are given below as:

Convolutional layer: The first and most significant layer of CNN is the convolutional layer, which performs most of the computation. This computation requires input data, filters and feature maps. A feature detector moves across different pixels of an image in convolution to identify and check the presence of a feature. A feature detector is a 2D array of weights illustrating an image's part. The filter is generally a 3*3 matrix. According to the formula, an output feature map in a convolution is made up of numerous input feature maps (x) multiplied by kernels and added with bias (3.2.1):

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l) \tag{3.2.1}$$

where, l is the lth layer, whose output is still being formed based on the inputs or outputs of the preceding $l-1$ layers, M_j is the collection of input feature maps (x), k_{ij} are the filters or kernels connected to the input feature map and b_j is the bias.

Each output value in the feature map only needs a connection with the image pixels, where the filter is applied rather than connecting to each image pixel. That is why these layers are also called "partially connected" layers. Some parameters affect the results of the neural network:

- quantity of filters determines the depth of the output
- filter's stride is the number of pixels or the distance it travels over the input matrix
- padding is used to fit the input image

ReLU function: ReLU function adds the non-linearity on the feature map obtained through the convolutional layer by replacing negative values with 0. ReLU is a non-linear function that learns the complex data relationships. ReLU unit refers to the node that implements this activation function. The equation for ReLU for a pixel value (y) of a feature map (x) is represented in (3.2.2).

$$f(y) = max(y, 0) \tag{3.2.2}$$

$$f(y) = \begin{cases} 0, & \text{if } y < 1 \\ y, & \text{if } y \geq 0 \end{cases}$$

Pooling layer: A pooling layer is employed to minimize the feature map's dimensionality by moving a window over the filtered images to capture the strongest activation value. It is sometimes referred to as downsampling, which lowers dimensionality by reducing the number of input parameters. The filter (2*2, 3*3) is spread throughout the full input via the pooling procedures. Pooling can be split into two kinds: Max and average pooling. The filter is moved over the input during max pooling while choosing the pixel having the highest value, to be included in the resultant matrix. Average pooling estimates the average value for the output array during the filter movement across the input. The pooling layer helps in reducing complexity, improving performance and reducing overfitting.

Flatten: The pooled featured map is then transformed to a resultant array forming a single long continuous linear vector. The feed-forward backpropagation neural network is then fed this flattened output after each training iteration. The model is trained to recognize the top-level and low-level features used by the softmax function to categorize images into several output classes. This training is performed under several epochs. Each hidden layer in the neural network is fully connected to its preceding and following hidden layers, as shown in Figure 3.2.

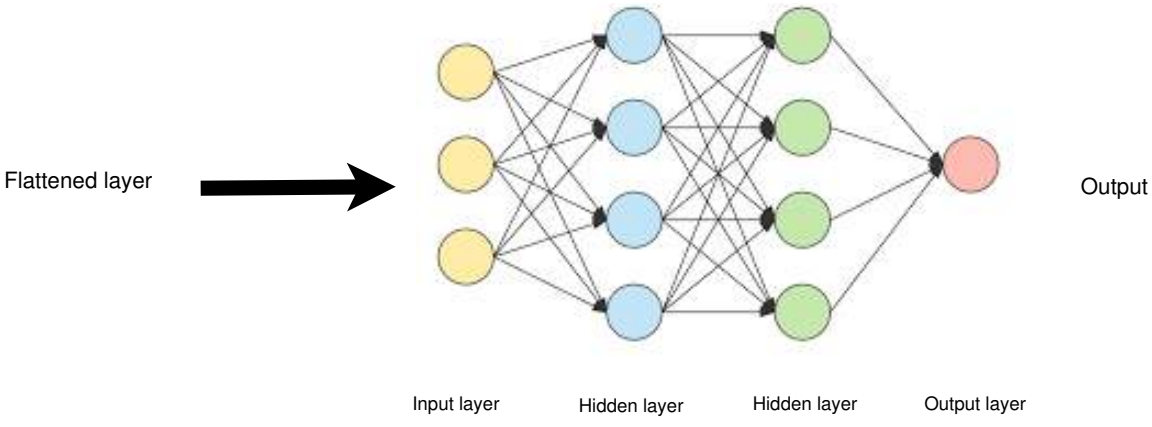


Figure 3.2: Different hidden layers connected to form a fully connected network

These completely interconnected layers are in charge of categorizing images based on the properties discovered in the preceding layers. The softmax function is employed by the resultant layer for accurate categorization of images to the output classes by generating

the probability from 0 to 1. This approach is thought to be simple for identifying non-linear combinations of top-level features that are produced by the convolutional layer.

3.3 TRANSFER LEARNING

Transfer learning is a machine learning technique that uses a model that has already performed learning using a large set of training data. This trained model is then optimized by retraining using a new and small set of training data. This way, it requires less amount of training data and time and results in higher accuracy as compared to the base model [143]. ImageNet, AlexNet, DenseNet, GoogleNet and Inception are some transfer learning models currently available.

Traditional machine learning techniques develop a model from expensive scratch, requiring extensive data and time-consuming practice. These techniques develop a model independently for a particular task without previous information and dependency. Transfer learning methods, however, are computationally effective that can acquire sound output even for small training data [53] as models already understand the features of previously trained models.

3.3.1 Categories of Transfer Learning

Transfer learning can be categorized based on the similarity of the task area. These are:

Homogenous transfer learning: This approach is applied when the areas in which the task is performed belong to the same feature space.

Instance transfer: In this approach, both the areas differ with slight distributions; the source area includes a large number of labeled inputs while the target space has limited inputs.

Parameter transfer: This approach transfers the data through the shared parameters of the trained model of the source and target space. This can be possible with the use of ensemble learning. This approach is proper when the source and target objectives are similar; then, the pre-trained source model with a good build structure can be used for the target space.

Feature-representation transfer: This approach is used to transfer the features from the original to the new one. This can be the asymmetric and symmetric approaches.

The asymmetric approach collects the attributes through the source region. It places them in the target attribute space. At the same time, the symmetric approach identifies a new standard latent features zone and converts both the original and new features into this common feature space.

Relational-knowledge transfer: This approach learns the logical relationships between the source and target areas, derive some conclusion out of it and use the extracted knowledge in the current scenario.

Heterogeneous Transfer Learning: This method is employed to find and fix problems that lie among the source and target regions, such as discrepancies between the feature space, data distributions, and label space of the source and target.

Therefore, transfer learning is a machine learning technique where a model is constructed for the same or different purposes.

3.4 PLANT DISEASE CLASSIFICATION USING TRANSFER LEARNING

To accurately identify and categorize plant diseases, this calls for an automatic plant disease recognition system. The process of identifying diseases has become simpler because of deep learning and the way it processes images using tools like CNN [144]. Therefore, in this section, we are developing several transfer models based on homogenous and parameter transfer learning for classification by using two distinct leaf diseases of potato and tomato plants. The process has been carried out by importing the transfer models and training them on standard datasets taken from Plant Village datasets and where the output classes of the standard transfer model get replaced with the desired output classes. While training the transfer model the features such as color, shape and texture from plant images are taken into consideration automatically.

3.4.1 The Transfer Learning Process

The transfer learning process applied to our dataset involving six steps is explained below and depicted in Figure 3.3.

1. **Get a pre-trained model**: The transfer learning process starts with adopting a pre-trained model as per the requirement to construct a base model. This process requires compatibility between the source model and the target work area by building a strong correlation. The pre-trained models that are used in our work according to the

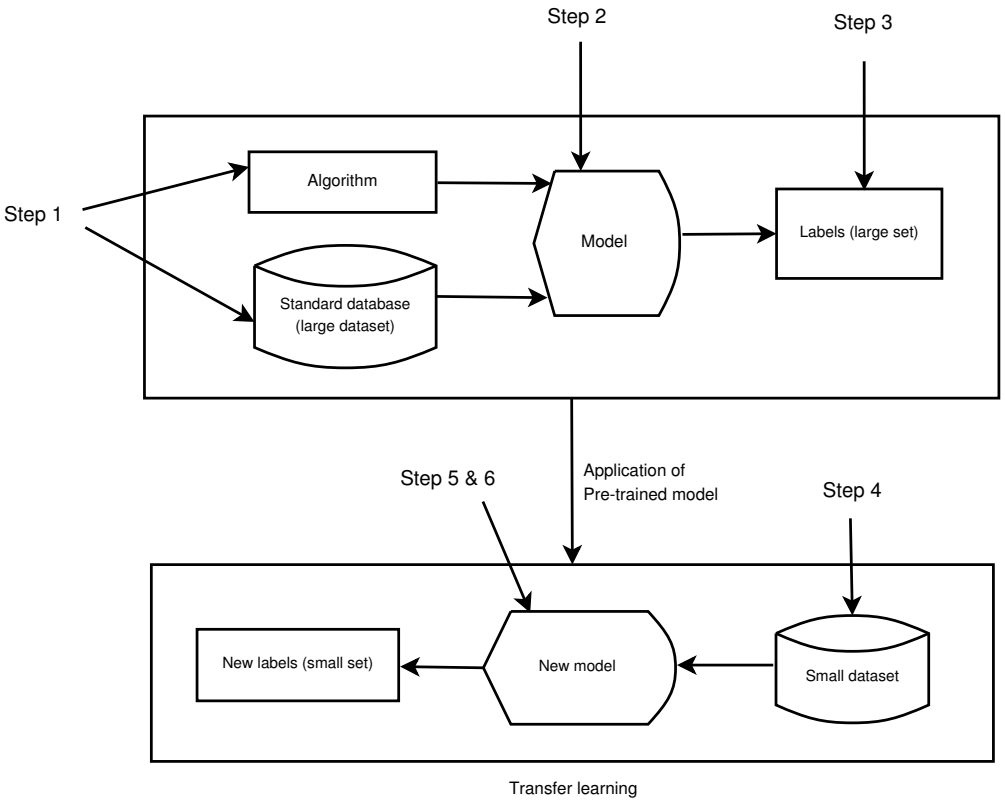


Figure 3.3: Process of Transfer Learning

requirement and literature study, are listed below:

- VGG-16
- Inception V3
- DenseNet
- ResNet
- GoogLeNet

2. Construct a base model: A base model, such as ImageNet or Inception-V3 selected in the first step, will be employed for the new task. These models could be used as network weights to save the additional training time needed for the new task or can be chosen as the network design to develop a system from scratch. There are situations when the source model's output layer has more neurons than the task's necessary labels; then, in those cases, the output layer is modified as per the requirement.

3. Freeze layers: The initial layers of the pre-trained model are frozen to allow one to skip the extra work of learning the model's fundamental features. Otherwise, it will

drop the learning that has already been done and start the process from scratch.

4. Add new trainable layers: The pre-trained model with its starting layers allows for feature extraction. These frozen layer needs to be added with an additional layer that will predict the output classes. This added layer is the last output layer containing the neurons as per the number of classes required as output. Concerning our dataset, the output layer of the pre-trained model is replaced with 10 classes for the tomato leaf dataset and 4 classes for the potato leaf dataset.

5. Train the new layers: The last output layer of the pre-trained model typically differs from the output layer required for the new job, for example, the 1000 output classes of the pre-trained model's resultant layer for ImageNet. Thus, the model developed for the new task should be trained with the newly added output layer per the need of the new job.

6. Model's fine-tuning: With fine-tuning, the achievement of the model created through the transfer model can be enhanced. This will occasionally unfreeze a portion of the previously learned model and retrain the complete model at a slow learning rate. This will enhance the model's performance while preventing overfitting. The transfer models used for our dataset are tuned with a 0.001 learning rate, which generally provides satisfactory results.

In addition to computer vision, the transfer models are also used for Natural language processing, audio and speech recognition, etc.

3.4.2 Different Transfer learning models based on CNN

The different transfer learning models applied in our work are discussed below.

VGG: The most common transfer learning technique used for image categorization is the VGG-16 model, which consists of three convolutional layers of increasing depth stacked on top of one another. The ImageNet dataset, which contains billions of images with 1000 distinct classes, was used to train the VGG-16. It was created by Oxford's Visual Graphics Group (VGG) [145, 146]. The main highlight of this 16-layered VGG-16 model is the increased depth. This allows 224*224 RGB images to pass through a network of convolutional layers, where every block is composed of a 3*3 filter. The data features are padded in the convolutional layer to preserve the spatial resolution with the stride of 1. The max pooling layers separate the blocks with a stride of 2 over 2*2

windows. Three fully connected layers are coupled with five blocks of convolutional layers. The softmax function, applied in the last level, divides the input data into several class probabilities.

DenseNet: A densely connected Convolutional Neural Network architecture called DenseNet [24] connects all of the hidden layers as the feed-forward fashion to maximize the flow of knowledge. DenseNet has the benefit of minimizing the vanishing gradient problem and reducing the parameters. A vanishing gradient is an issue in deep networks where there are several layers passes through input and output and the information vanishes before reaching the endpoint. DenseNet helps in reducing such problems and thus, improves accuracy. It is found as the best model for feature propagation. There will be n(n+1)/2 connections in DenseNet if n is the number of network levels. In comparison to other models, DenseNet contains fewer layers, making it easier to train more than 100 layers. DenseNet is called parameter efficient as it works with limited parameters or kernels.

Inception: The inception [143] model belongs to GoogLeNet architecture that performs the work on ImageNet data. The convolutions are made of various sizes, such as 1×1, 3×3 and 5×5. It makes use of max-pooling. The outputs derived from the layers are combined and put into the next layer. 1×1 convolutions which enable dimensionality reduction and reduce the amount of computing needed, are a key feature of the Inception design. The Inception architecture is made with the repetition of inception modules. The model building emphasizes a large deep neural network consisting of many layers and units within the layers. It helps to extract features at different scales to cover broader perception of objects like human biological visuals, which extracts patterns at varying scales. This allows for multi-scale covnet to extract and learn wider. This structure resolves the overfitting issue due to the large network, which will ultimately manage the cost and complexity. The initial version of the Inception model was named as GoogleNet or Inception-v1.

ResNet: The development of the Deep ResNet model [110], which is a component of the residual network, was made possible by the non-linear layer's failure to recognize identity mapping and degradation issues. It is eight times deeper than the VGG model and has 34 plain layer networks, hence it is called Deep. Various building pieces make up the network, each consisting of a unique residual unit. The problem of training

an intense network consisting of several layers has been solved by introducing several blocks called residual blocks. It allows skipping connections that skip some model layers to make a direct connection. The dimensions for the skip connection are increased by padding with zero and added with a 1*1 convolutional layer to match the dimension. These skip connections help solve the vanishing gradient problem by providing skip connections for the gradient to flow through. These connections are also managed by regularization.

3.4.3 "Agri-CNN"(model built from scratch)

Model building from scratch is a tedious task. The architecture is designed from the base where different layers, activation functions and parameters are combined, making different possible combinations that will ultimately increase the accuracy. The weight and bias are also adjusted to make the model more efficient. The model does not use pre-trained learning; it learns and builds an entire model from the initial stage, avoiding unnecessary layers and complexity. A model "Agri-CNN" has been proposed in this work for disease and grain classification, which is developed from scratch. It consists of three convolutional layers, along with two max-pooling layers with a 0.25 dropout value and 0.001 learning rate, two dense layers with softmax function are used to generate the resultant classes. This proposed model "Agri-CNN" developed in this work, is explained further in detail in Section 3.5.

3.4.4 Methodology

The work and effort have demonstrated the application of CNN using information gathered through the PlantVillage dataset [24]. The tomato and potato leaves in the dataset are both diseased and uninfected. The data consists of colored images of various sizes that have later been shrunk and normalized into 224*224. Figure 3.4 shows the entire process for recognizing and categorizing plant diseases using transfer learning based on the image data. The PlantVillage dataset served as the source for this dataset. Transforming, filtering, resizing, rotating, translating and enlarging the data were the initial pre-processing steps. This is followed by the labeling of data based on expert knowledge. These labeled pictures are partitioned into training, test and validation data. The training images are employed for the different transfer learners like−ResNet 50, DenseNet 121,

VGG 16 and Inception-v3 to generate the output classes.

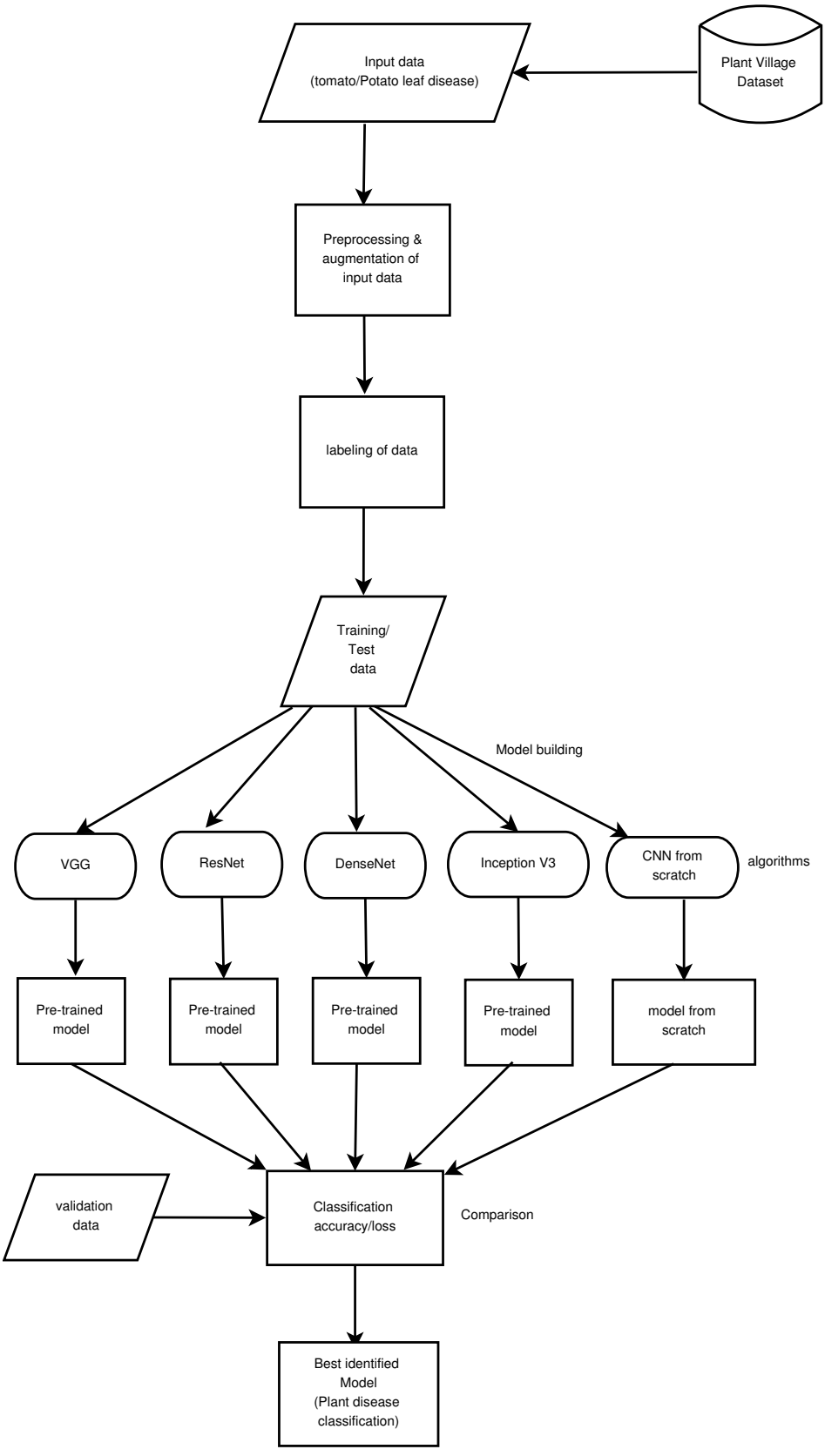
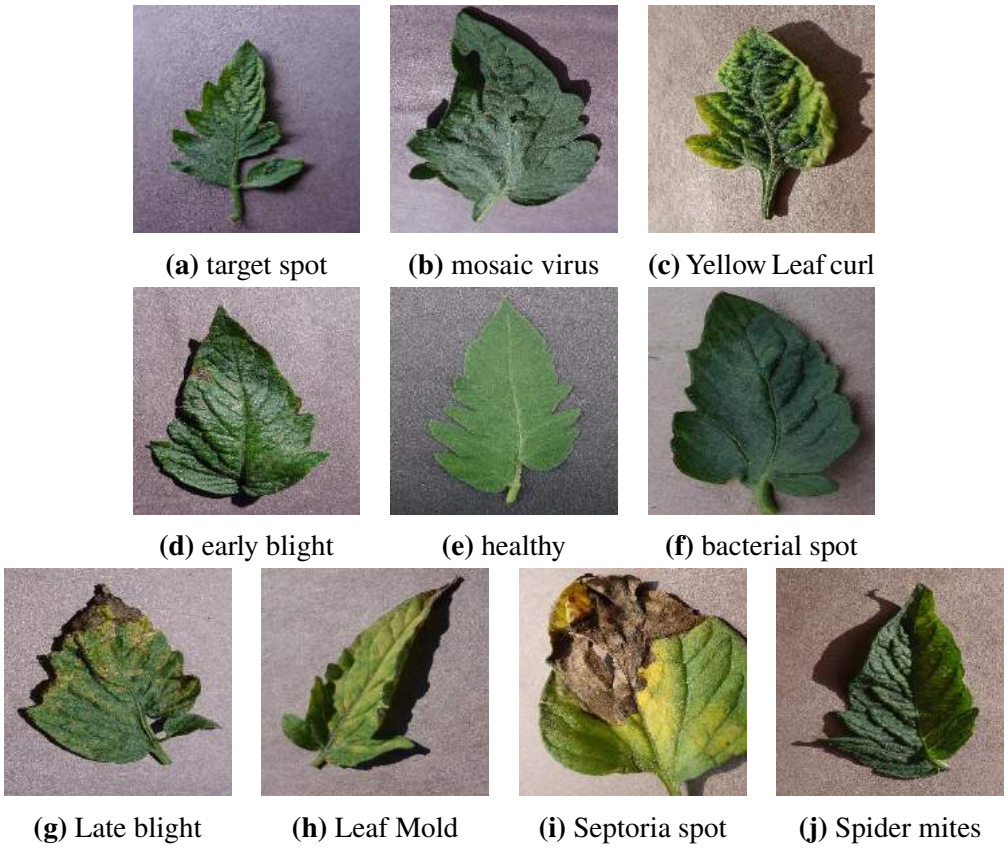


Figure 3.4: Flow chart of the proposed work

The resulting model's classification accuracy is verified by the application of the model with the test data. The trained model is further verified using the validation data. Finally, the accuracy and loss of all tested models are compared on the given dataset.

3.4.5 Gathering of Data

This study makes use of the PlantVillage dataset [48] to classify tomato and potato diseases. Nine distinct diseases and one set of healthy tomato leaves make up the dataset for tomato diseases. The ratio of the collected photos is 70:20:10, where 70% data is set aside for training, testing uses 20% data and 10% for validation. The dataset is unbalanced for a different set of classes. Figure 3.5 displays pictures of the tomato leaf diseases under the various classes. Three groups-two sick and one healthy-make up the dataset of potato diseases. The training set acquires 75% data, while the test set makes use of 25% data. The pictures have been scaled down to 224 x 224 pixels. In Figure 3.6, a few sample pictures of potato leaves are displayed.



(a) target spot **(b)** mosaic virus **(c)** Yellow Leaf curl

(d) early blight **(e)** healthy **(f)** bacterial spot

(g) Late blight **(h)** Leaf Mold **(i)** Septoria spot **(j)** Spider mites

Figure 3.5: Sample of different tomato leaf diseases

(a) early blight **(b)** late blight **(c)** healthy

Figure 3.6: Sample of different potato leaf diseases

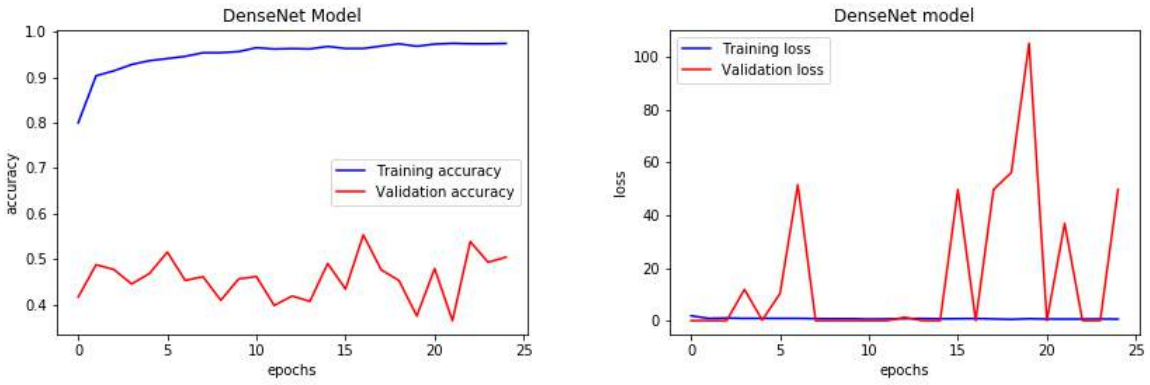
Table 3.1: Evaluation Metrics of transfer learning models for tomato leaves

No. of epochs	Deep Learning Models	Training loss	Training Accuracy	Validation loss	Validation Accuracy
10	Densenet	0.7183	0.9566	0.00000	0.4724
	Inception V3	1.7048	0.8529	30.3588	0.5727
	ResNet	1.1237	0.9697	51.0649	0.0590
	VGG 16	0.1494	0.9465	0.02110	0.8841
	Agri-CNN	0.9488	0.6724	0.81470	0.6830
20	Densenet	0.5959	0.9725	16.2734	0.5456
	Inception V3	1.6666	0.8827	19.4069	0.5311
	ResNet	0.7606	0.9812	143.521	0.0659
	VGG 16	0.1075	0.9644	0.00410	0.9049
	Agri-CNN	0.6377	0.7796	0.71850	0.8308
25	Densenet	0.5712	0.9748	67.6174	0.4408
	Inception V3	1.5943	0.8940	0.00000	0.4714
	ResNet	0.7837	0.9829	169.440	0.0659
	VGG 16	0.0886	0.9694	0.06190	0.9084
	Agri-CNN	0.0742	0.9531	0.05120	0.8992

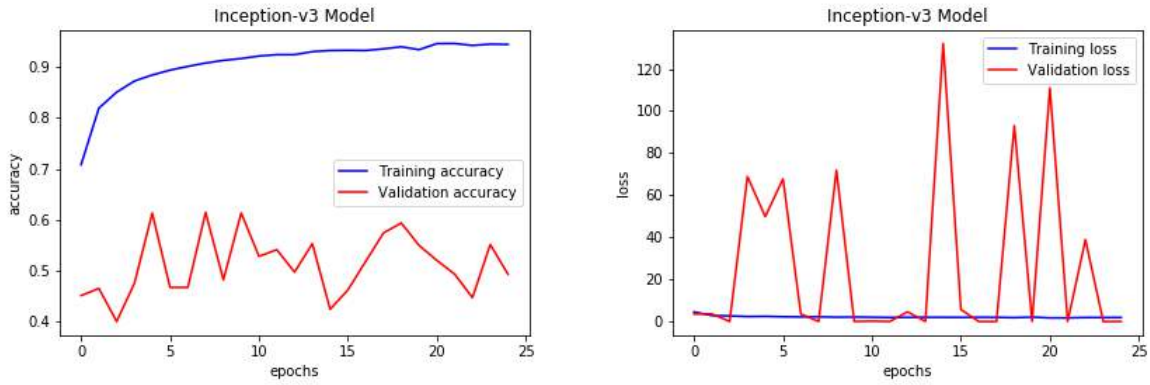
3.4.6 Experimentation and Results

The process is executed using python 3.7 with NVIDIA GPU processor along with OpenCV, Keras and CuDNN libraries. The experiments are conducted with four transfer learning models along with the "Agri-CNN" model built from scratch. The process takes care of morphological, texture and color features while performing the classification. Table 3.1, list the various model's assessment metrics as per accuracy and loss during training and testing which illustrates the evaluation results for tomato leaf diseases. For tomato leaves, the results are examined for 25 epochs. Similarly, for potato leaf diseases, evaluation is done for 30 epochs because after that the results get saturated. With a batch size of 32 and a learning rate of 0.001, the models are trained for both datasets.

The experimentation has been performed through five deep learning models: DenseNet, Inception-v3, ResNet50, VGG-16 and Agri-CNN. Initially, the models get pre-trained weights of learning through ImageNet, the output layer of the transfer model is then replaced with the output classes of our dataset. There are ten output classes in the tomato leaf disease dataset and three output classes in the potato leaf disease dataset. The training and test results produced under the different models experimented with tomato leaf disease are shown graphically in Figure 3.7 and Figure 3.8. The accuracy and loss obtained during training and validation with the DenseNet model reveal poor validation accuracy and validation loss, as shown in the first two graphs (a) and (b) of Figure 3.7. The Inception-v3 model also produces low accuracy and loss. It has been found from Figure 3.8 and Table 3.1, that the proposed "Agri-CNN" and VGG-16 models have produced better results than the other three models.

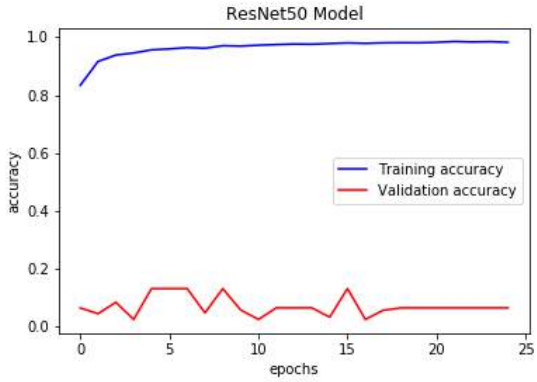


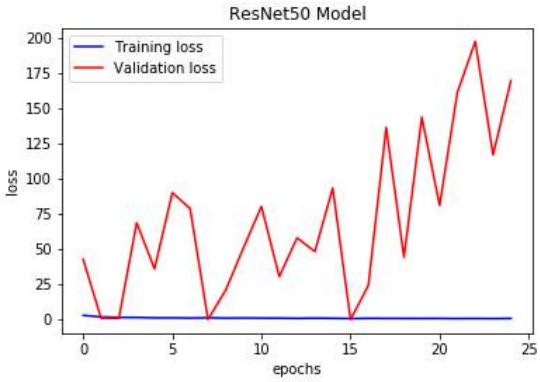
(a)Accuracy during training and validation (DenseNet) (b)Loss during training and validation (DenseNet)



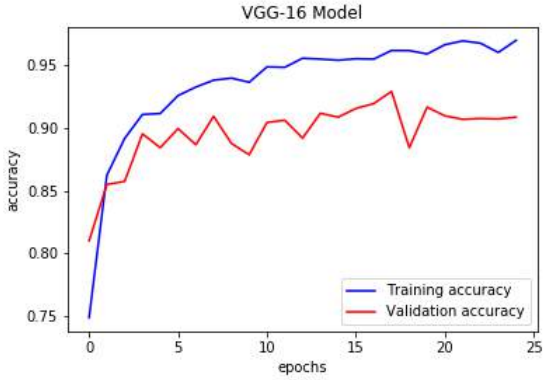
(a)Accuracy during training and validation (Inception-v3) (b)Loss during training and validation (Inception-v3)

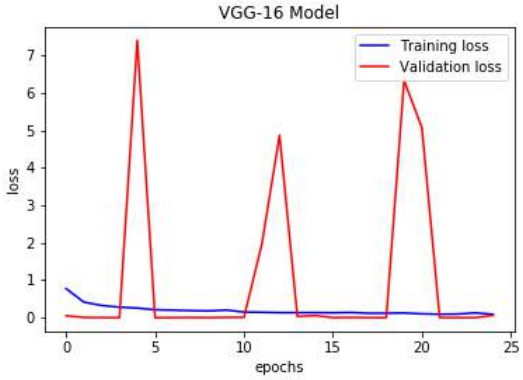
Figure 3.7: Accuracy and loss graph during training and testing under DenseNet and Inception-v3 model for tomato leaf diseases.



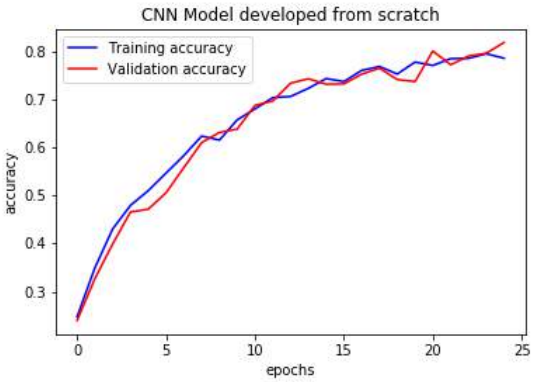


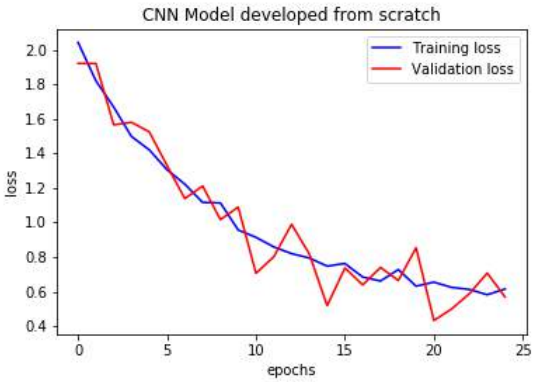
(a)Accuracy during training and validation (ResNet50) (b)Loss during training and validation (ResNet50)





(a)Accuracy during training and validation (VGG-16) (b)Loss during training and validation (VGG-16)





(a)Accuracy during training and validation (Agri-CNN) (b)Loss during training and validation(Agri-CNN)

Figure 3.8: Accuracy and loss graph during training and testing under ResNet50, VGG-16 and Agri-CNN model for tomato leaf diseases.

VGG-16 gave the highest validation accuracy and minimal validation loss followed by "Agri-CNN" whereas, DenseNet121, Inception-v3 and ResNet50 performed poorly on the dataset. The Confusion Matrix is further computed to validate the results.

	0	1	2	3	4	5	6	7	8	9
0	2	0	0	0	57	19	0	1	0	0
1	0	3	1	1	17	9	0	1	2	0
2	0	0	28	0	11	22	0	3	0	0
3	0	0	1	6	8	4	0	2	5	0
4	0	0	0	0	50	4	0	0	1	0
5	0	0	0	0	2	55	0	0	0	0
6	0	1	0	0	9	20	1	1	1	0
7	0	0	0	0	1	6	0	83	1	0
8	0	0	0	0	0	0	0	0	12	0
9	0	0	0	0	11	27	1	1	2	7

(a) Confusion matrix for DenseNet121 model

	0	1	2	3	4	5	6	7	8	9
0	17	3	1	0	3	21	0	1	15	5
1	1	8	3	0	1	14	0	1	7	0
2	1	5	35	0	1	12	1	1	1	2
3	0	2	6	2	2	9	0	0	8	0
4	2	2	6	0	14	14	0	4	9	1
5	0	0	0	0	0	59	0	0	0	0
6	0	2	0	0	0	45	1	1	5	3
7	1	0	0	0	0	8	0	76	12	0
8	0	0	0	0	0	2	0	0	8	0
9	0	0	0	0	0	16	0	0	1	19

(b) Confusion matrix for Inception-v3 model

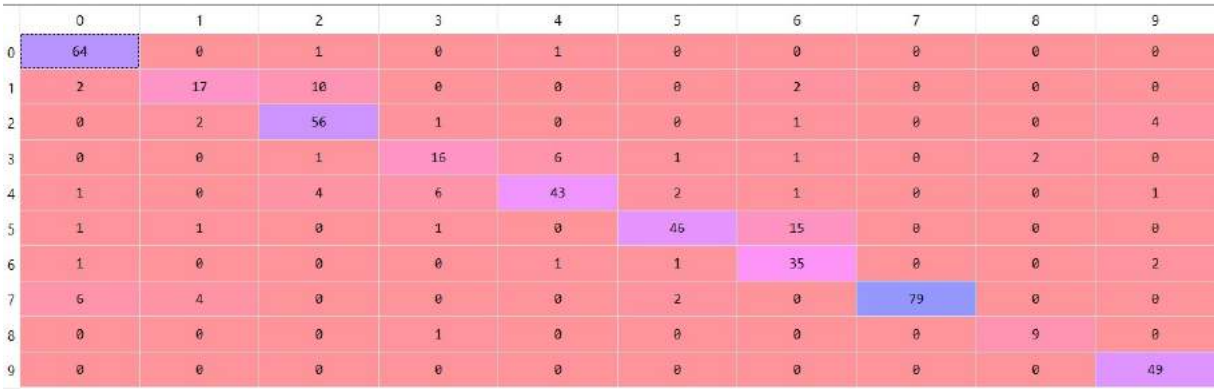
	0	1	2	3	4	5	6	7	8	9
0	0	0	0	69	0	0	0	0	0	0
1	0	0	0	30	0	0	0	0	0	0
2	0	0	0	56	0	0	0	0	0	0
3	0	0	0	41	0	0	0	0	0	0
4	0	0	0	51	0	0	0	0	0	0
5	0	0	0	49	0	0	0	0	0	0
6	0	0	0	39	0	0	0	0	0	0
7	0	0	0	110	0	0	0	0	0	0
8	0	0	0	10	0	0	0	0	0	0
9	0	0	0	45	0	0	0	0	0	0

(c) Confusion matrix for ResNet50 model

	0	1	2	3	4	5	6	7	8	9
0	69	0	0	0	1	0	0	0	0	0
1	0	29	0	0	0	0	3	3	0	0
2	1	2	56	1	0	0	2	0	0	1
3	1	2	0	22	0	0	1	3	0	0
4	0	2	1	1	43	0	3	0	0	0
5	0	0	0	0	0	41	7	1	0	0
6	0	0	0	0	0	0	42	0	0	0
7	1	0	0	0	0	0	1	92	0	0
8	0	0	0	1	2	0	1	0	12	0
9	0	0	0	0	0	0	2	0	0	50

(d) Confusion matrix for VGG-16 model

Figure 3.9: Test set confusion matrix for tomato leaf diseases with DenseNet, Inception-v3, ResNet50 and VGG-16 based on training. The top row and the first column signify the different output classes.



	0	1	2	3	4	5	6	7	8	9
0	64	0	1	0	1	0	0	0	0	0
1	2	17	10	0	0	0	2	0	0	0
2	0	2	56	1	0	0	1	0	0	4
3	0	0	1	16	6	1	1	0	2	0
4	1	0	4	6	43	2	1	0	0	1
5	1	1	0	1	0	46	15	0	0	0
6	1	0	0	0	1	1	35	0	0	2
7	6	4	0	0	0	2	0	79	0	0
8	0	0	0	1	0	0	0	0	9	0
9	0	0	0	0	0	0	0	0	0	49

(a) Confusion matrix for Agri-CNN

Figure 3.10: Test set confusion matrix for tomato leaf diseases with Agri-CNN model based on training. The top row and the first column signify the different output classes.

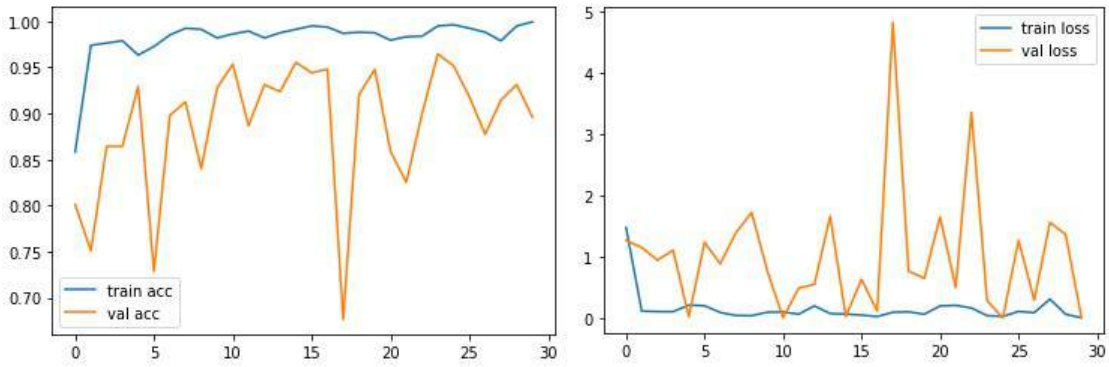
Figures 3.9 (a), (b), (c) and (d) depicts the confusion matrix for the test set on tomato disease based on the predicted output and actual output generated under the DenseNet, Inception-v3, ResNet and VGG-16 models, respectively. Figures 3.10 (a) and (b) show the confusion matrices generated with Agri-CNN, where the top row and leftmost column show the output classes. It can be observed from the five confusion matrices that VGG-16 predicts the highest correct classes resulting in maximum accuracy.

The results of the experiments performed on potato leaf disease are shown in Table 3.2.

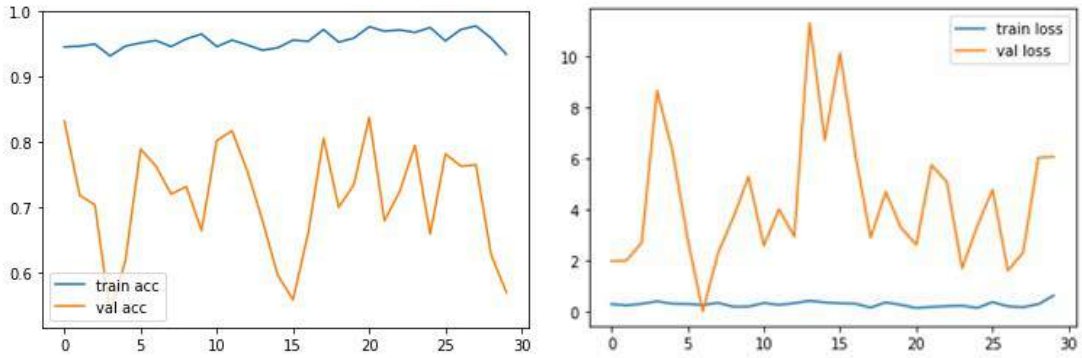
Table 3.2: Evaluation Metrics of transfer learning models for potato leaves

No. of epochs	Deep Learning Models	Training accuracy	Training loss	Validation accuracy	Validation loss
10	InceptionV3	0.9659	0.1970	0.6654	5.2816
	VGG 16	0.9882	0.0425	0.9610	0.0488
	ResNet 50	0.9913	0.1097	0.4647	30.955
	DenseNet 121	0.9820	0.0942	0.9275	0.7710
	Agri-CNN	0.9712	0.0589	0.9524	0.0512
20	InceptionV3	0.9597	0.2754	0.7361	3.2753
	VGG 16	0.9969	0.0167	0.9480	0.1903
	ResNet 50	0.9938	0.1961	0.4647	49.026
	DenseNet 121	0.9876	0.0612	0.9480	0.6481
	Agri-CNN	0.9869	0.0367	0.9280	0.2303
30	InceptionV3	0.9349	0.6228	0.5706	6.0523
	VGG 16	0.9994	0.0080	0.9535	0.2544
	ResNet 50	0.9913	0.1611	0.4647	32.607
	DenseNet 121	0.9994	0.0057	0.8959	0.0144
	Agri-CNN	0.9763	0.1321	0.9232	0.3241

Figure 3.11 and 3.12 shows the accuracy and categorical cross-entropy loss for each model. ResNet 50 and Inception-v3 models performed poorly while DenseNet 121, VGG 16 and "Agri-CNN" performs comparatively well on the given data. The result of learning on the test data reveals the poor performance of ResNet50 and Inception-v3(unstable performance). It can be identified from the output that the proposed model "Agri-CNN" also performs well in accuracy and loss, nearly comparable to that of VGG-16. Overall, it can be stated that VGG-16 performance is best among the all, with the greatest accuracy on the training and test set, followed by Agri-CNN and DenseNet, as shown in Table 3.2. Figures 3.11 and 3.12 show that the findings are not balanced because the dataset contains only three classes and 2152 photos in the case of potato diseases. The model is trained up to the 30th epoch only because for VGG16, training beyond this results in saturation of the training and test accuracy. It is also noted that the test loss ranges from 0.01 to 0.25, considered a respectable score for such a small dataset.

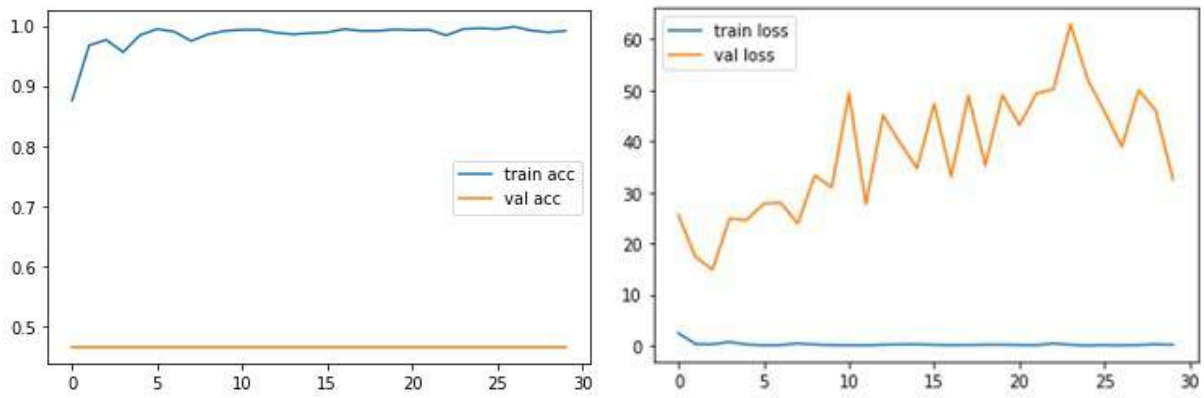


(a)Accuracy during training and validation (DenseNet) (b)Loss during training and validation (DenseNet)

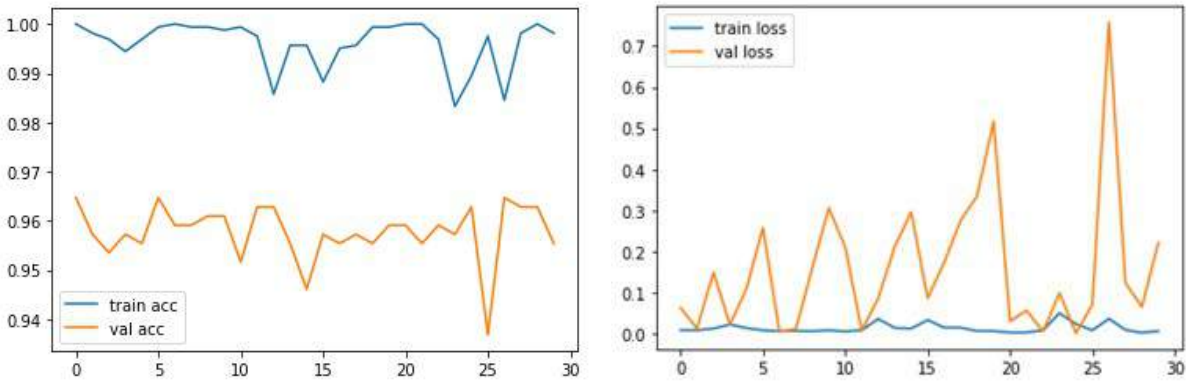


(a)Accuracy during training and validation (Inception-v3) (b)Loss during training and validation (Inception-v3)

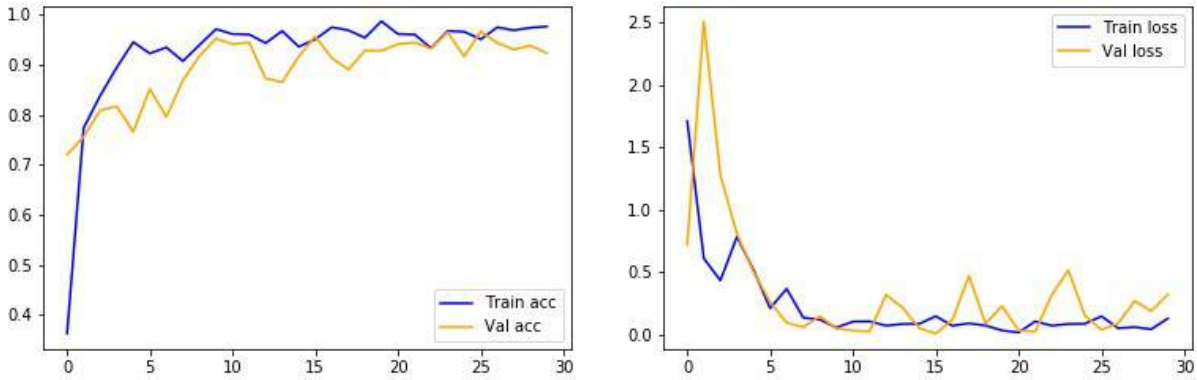
Figure 3.11: Accuracy and loss graph during training and testing under DenseNet, Inception-v3 models for potato leaf diseases.



(a)Accuracy during training and validation (ResNet50) (b)Loss during training and validation (ResNet50)



(a)Training and validation accuracy (VGG-16) (b)Training and validation loss (VGG-16)



(a)Training and validation accuracy(Agri-CNN) (b)Training and validation loss(Agri-CNN)

Figure 3.12: Accuracy and loss graph during training and testing under ResNet50, VGG-16 and Agri-CNN models for potato leaf diseases.

Since the data appear to exhibit the same graph structure even after the 30th epoch therefore the epochs are not raised further. It has been identified that by expanding

the dataset and adding more classes, the results can be balanced. Thus, it can be concluded from the two tables that VGG-16 performs better than any other model for the two datasets, whereas Agri-CNN has also performed well for the same small and unbalanced data.

The results of this work are also contrasted with the earlier techniques discussed in the literature. This comparison is shown in Table 3.3.

Table 3.3: Comparison of results with other transfer learning models

Paper	No. of Classes	Classification	Best techniques	Model Accuracy
[62]	9	CNN, AlexNet, GoogLeNet, InceptionV3	InceptionV3	99.18%
[50]	9	D-CNN(Transfer learning based CNN)	D-CNN	99.5%
[49]	10	CNN,VGG16, InceptionV3 and MobileNet	CNN	91%
[51]	7	AlexNet, VGG-16	AlexNet	91.2%
Models employed in this work	10	Agri-CNN, DenseNet121, ResNet50, Inception-V3, VGG-16	VGG-16	90.84%

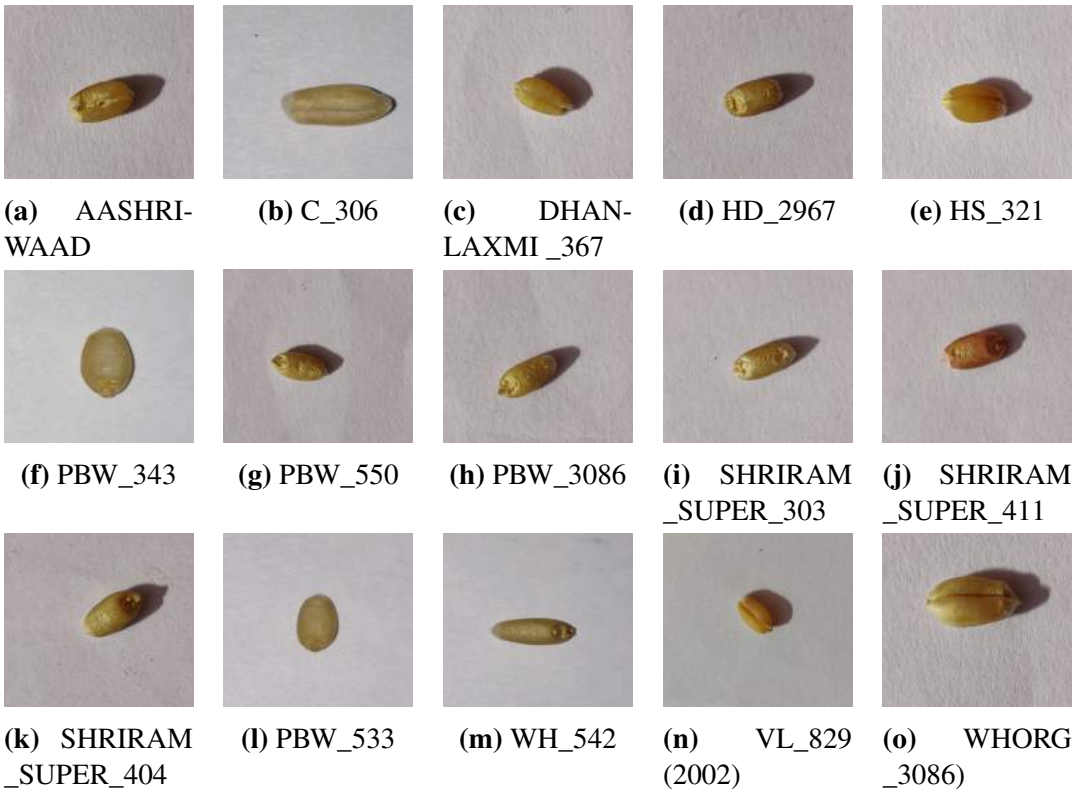
3.5 WHEAT GRAIN CLASSIFICATION

The proposed model "Agri-CNN" as discussed earlier was developed for the classification of large varieties of grains. It is used to classify the grain varieties such as wheat, considering the different visual and textual features. The work has shown different model variations with hyper-tuning to identify the accurate model which can classify the wheat grain varieties with maximum accuracy and minimum cross-entropy loss. The developed model is also contrasted with the earlier identified best transfer learning model – VGG-16.

3.5.1 Proposed Model

Convolutional Neural Networks, a common Deep Learning method for image data, is used to create an "Agri-CNN" model for categorizing the grain types. The model

successfully identified the correct classes of grains irrespective of the close similarities between the classes. Before CNN [147,148], the classification process relied on Machine Learning methods, which call for a manually created feature extraction process. Earlier methods were also based on Transfer learning, where the Deep learning models are learned on a standard dataset. For a vast volume of data like ImageNet, our work suggested a model which is created from scratch with hyper-parameter tuning to generate more accurate results. The research has demonstrated several model variants with hyper-tuning to find the correct model which will accurately and efficiently categorize the diverse wheat grain types.



(a) AASHRI-WAAD **(b)** C_306 **(c)** DHAN-LAXMI _367 **(d)** HD_2967 **(e)** HS_321

(f) PBW_343 **(g)** PBW_550 **(h)** PBW_3086 **(i)** SHRIRAM _SUPER_303 **(j)** SHRIRAM _SUPER_411

(k) SHRIRAM _SUPER_404 **(l)** PBW_533 **(m)** WH_542 **(n)** VL_829 (2002) **(o)** WHORG _3086)

Figure 3.13: Sample images of Wheat grain varieties

3.5.2 Gathering and Preparing Data

Grain categorization used to be handled by professionals in the agricultural sector, but this method is time- and money-consuming and sometimes inaccurate. Thus, it motivates us to develop such a model that will perform this automatic classification of grains. The dataset consists of 15000 images, some of the sample images are shown in Figure 3.13. Fifteen different wheat classes were collected from Haryana, India's corn exchange offices. Due to the necessity for a versatile and user-friendly system that the

majority of people can understand, the photographs were taken using a mobile phone's 16-megapixel primary camera. Data acquisition is performed to display a single grain of wheat over a white background followed by data augmentation [25] to standardize the photos. These scaled photos are given to the deep network's input layer for classification. Texture, color and morphological features are used in the classification process to assign the photos to the appropriate output classes.

Classes of Wheat

The study has gathered fifteen types of wheat that are grown in different parts of India and are recognized scientifically as "Triticum Aestivum." Nearly 15000 samples are formed by collecting 1000 pictures of each variety. Table 3.4 lists the name of the wheat variety, how many samples there are of each type, and where in India it is grown.

Table 3.4: Wheat varietals, sample size and cultivation location

Indian Name	No. of Samples	Indian Agricultural locations
AASHRIWAAD	1000	Central Zone
C_306	1000	Central Zone
DHANLAXMI_367	1000	Central, North Western, North Eastern Plains Zone
HD_2967	1000	North Western Plains Zone
HS_321	1000	Central, North Western Plains Zone
PBW_343	1000	North Western Plains Zone
PBW_550	1000	North Western Plains Zone
PBW_3086	1000	Central, North Western, North Eastern Plains Zone
SHRIRAM_SUPER_303	1000	Central, North Western Plains Zone
SHRIRAM_SUPER_411	1000	Central, North Western Plains Zone
SHRIRAM_SUPER_404	1000	Central, North Western Plains Zone
PBW_533	1000	Peninsular Zone
WH_542	1000	North Western Plains Zone
VL_829(2002)	1000	Northern Hills Zone
WHORG_3086	1000	North Western Plains Zone

Dataset Split Ratio

The 15000 wheat variety photos are divided into three groups, training data having 70%(10,500 images) of total data, 20%(3,000 images) being set aside for the test set, while the remaining 10%(3,000 images) being used for the validation set. Random images are assigned to different groups.

Software

Python 3.7 and TensorFlow 2.0 with Keras embedded together are used to experiment. For matrix calculations and classifications, Scikit-learn and NumPy libraries are employed. Using an Nvidia GPU processor, the investigations are carried out.

3.5.3 Experimentation

To find the optimal model that can successfully and accurately categorize photos of many varieties of wheat, various variants of the CNN model have been constructed and tested using parameter tuning. Through the use of accuracy and loss during training and validation, the outcomes are generated with hyper-parameter adjustment. However, few hyper-parameters are maintained constant in the generated models for network training and testing, as introduced below:

Adam's optimizer: A stochastic gradient descent extension called Adam's optimizer [149] determines an adaptive learning rate for a variety of parameters. As the optimizer is computationally economical and noise-resistant, it has opted for the classification over the other optimizers.

Softmax function: For multiclass classification, the CNN model's output layer applies the Softmax function. It is regarded as superior to the sigmoid function because it performs output mapping with a probability distribution in the range [0,1] such that the probability of each class is added together to give the value, if it equals 1 then the predicted output will be the class with a high probability [22,150]. The sigmoid function also computes probability in the range [0,1] but applies to binary classification. The definition of the softmax function $\sigma: R^K -> R^K$ is given by the formula (3.5.1):

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e_k{}^z} \tag{3.5.1}$$

where, j= 1,2,.....,k, and $z = (z_1, z_2, ..., z_k) \in R^k$

The function will assess the likelihood of each element in the input vector, z_j, overall potential target classes to compute the total of the components, $\sigma(z)$, which will result in an output of 1. The y-labeled images are divided into K classes by the softmax function,

which is expressed as (3.5.2):

$$y^{(j)} \in 1, 2, 3, ..., K \tag{3.5.2}$$

The training data set is made up of the following pairs:
$(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, ..., $(x^{(n)}, y^{(n)})$ where n is the number of labelled data samples, $x^{(j)}$ are the input image features, and $y^{(j)}$ are the labels.

Categorical accuracy: Any model's accuracy for the prediction is calculated using the categorical accuracy metric. As it slants in the direction of 1, the model becomes more precise. In a case involving a multiclass classification, the predicted class and the labeled class must coincide. According to the research, categorical accuracy is the foundation for training and test accuracy. Following is the formula for category accuracy (3.5.3):

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{3.5.3}$$

Categorical cross-entropy loss: In a situation where the result of the classification model is represented as a probability value between 0 and 1, the function assesses the cross-entropy loss. The loss value rises as the projected probability moves farther from the actual label. By maximizing the weights and filters, the function aims to lower the cost function. Cross-entropy is utilized for multiclass classification rather than mean square error since the former is used for classification and the latter is used for numerical and regression issues. Based on categorical cross-entropy loss, the loss that occurred during training and validation is reported in the work. The following is the cross-entropy formula for multiclass classification (3.5.4):

$$CE = -\sum y^{(j)} log(\sigma(z_j)) \tag{3.5.4}$$

Figure 3.14 depicts the architecture of the proposed Agri-CNN model. The model is created from scratch. Here, the solid boxes show the fixed parameters, and the dotted boxes show the regions used for tuning the hyper-parameters.

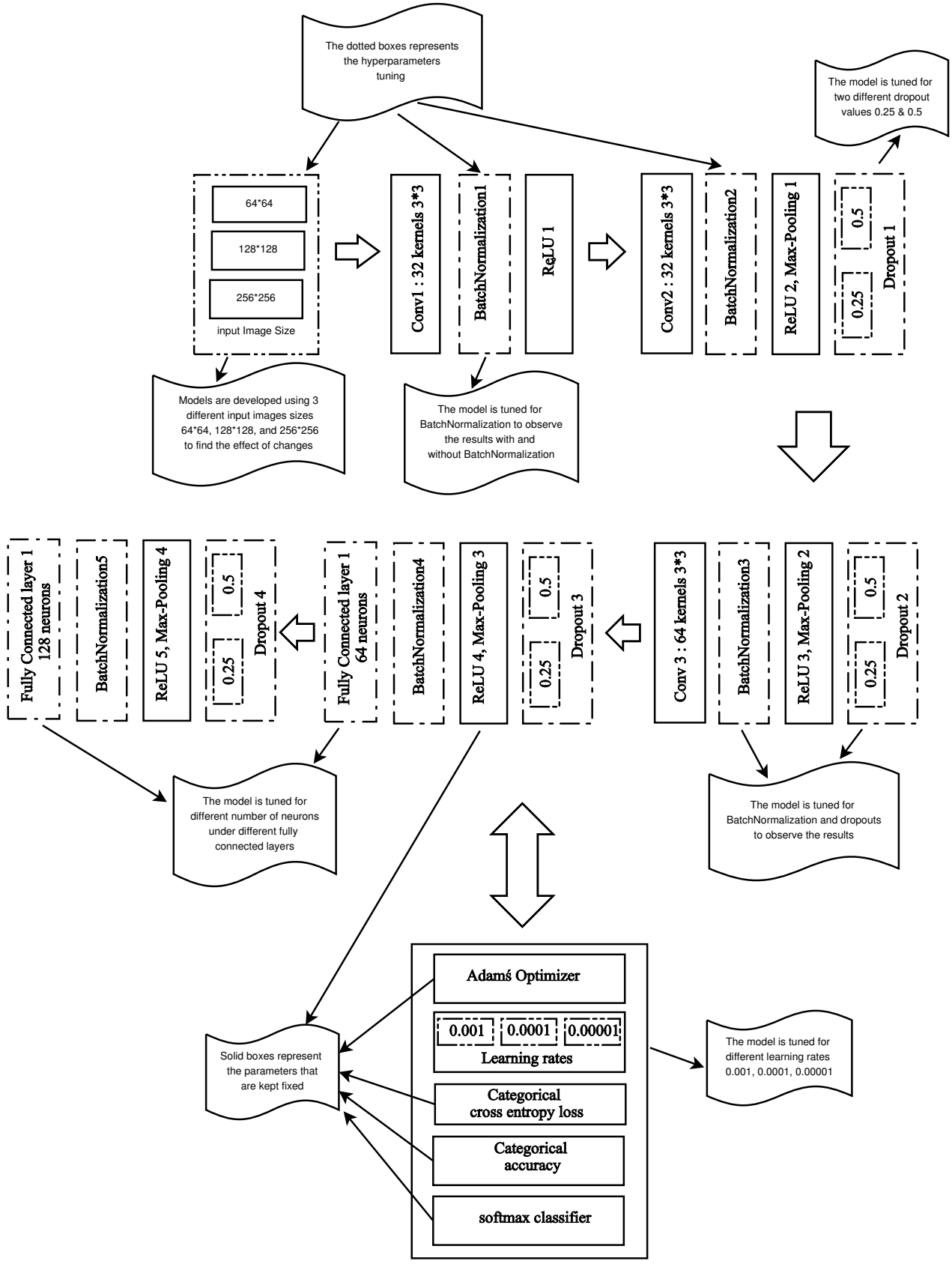


Figure 3.14: Architecture of the "Agri-CNN" model

Table 3.5 lists the hyper-parameters that were adjusted during model building. These hyper-parameters are briefly explained below.

Table 3.5: Hyper-parameters adjusted for experiments

S. No	Hyperparameter	Values
1	Epochs	5,10,15,20,25,30,35,40,45,50
2	Image Size	64*64, 128*128, 256*256
3	Dropouts	0.25, 0.5
4	Hidden Layer	3 conv, 3 ReLU, 2 max-pool
		4 conv, 4 ReLU, 3 max-pool
5	Dense Layer	Single layer of 64 or 128 neurons
		Two dense layer of 64 and 128 neurons
6	Batch-Normalization	With and without batch normalization
7	Learning Rate	0.001, 0.0001, 0.00001 with Adam's optimizer
8	Batch Size	16, 32

Number of Epochs- To provide more significant fluctuations and changes, the epochs are multiplied by 5. The number of iterations for every epoch determines the learning data sample size or batch size. Only the first 50 epochs are taken into account, after which the validation accuracy becomes saturated or occasionally worsens.

Dropout-Dropout causes a neuron to cease operation by removing half of the feature detectors randomly, which helps to reduce overfitting when training. The two dropout levels used in the training are 0.5 and 0.25.

Hidden layer- Convolutional, ReLU, max-pooling and dropout are components of the hidden layer. To determine the impact of changes, these layers can be altered.

Batch Normalization- To prevent biasing in the output layer that could result from the existence of bigger weights, batch normalisation [151] is used to normalize the output of one layer before applying the activation function. The subsequent layer receives these normalized findings. To determine how batch normalization affects categorization, the results of this work are compared with and without it.

Learning rate- Network weights are managed by the learning rate following the loss function to reduce the loss value. Adam optimizer with three different learning rates 0.001, 0.0001 and 0.00001, has therefore been taken into consideration in the work.

Batch size- With two possible sizes: 16 and 32, batch size describes the number of samples that are passed to the network during training via each iteration. Batch sizes

greater than 32 are not taken into account for training because it depletes the GPU's memory.

The model building based on CNN via tuning of hyper-parameters is performed under different parameters. The technique was put forth to provide the ideal model for classifying wheat grains. As indicated in Table 3.5, some of the characteristics are adjusted for categorization while others are maintained constant. These model buildings with the tuning of hyper-parameters are discussed in the following sections.

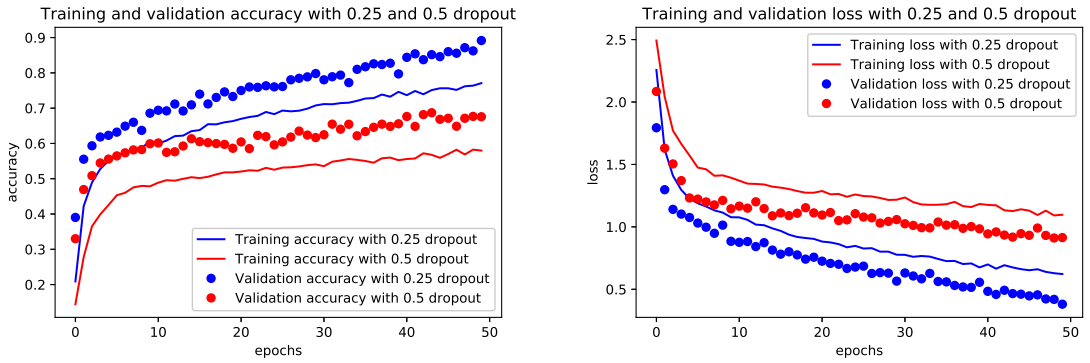
Table 3.6: Accuracy and loss measures with 0.25 and 0.5 dropout

No. of Epochs	0.25				0.5			
	Training accuracy	Training loss	Validation accuracy	Validation loss	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.5512	1.2238	0.6230	1.0750	0.4255	1.5738	0.5557	1.2316
10	0.6051	1.0786	0.6857	0.8842	0.4785	1.3924	0.5990	1.1448
15	0.6348	0.9880	0.7093	0.8140	0.5043	1.3230	0.6133	1.0883
20	0.6633	0.9052	0.7330	0.7570	0.5180	1.2744	0.5863	1.1119
25	0.6830	0.8472	0.7603	0.6779	0.5255	1.2601	0.5960	1.1055
30	0.7077	0.7767	0.7980	0.5654	0.5406	1.2167	0.6167	1.0581
35	0.7207	0.7421	0.8100	0.5625	0.5534	1.1765	0.6217	1.0388
40	0.7463	0.6749	0.7970	0.5561	0.5525	1.1838	0.6557	0.9841
45	0.7563	0.6597	0.8460	0.4607	0.5592	1.1406	0.6683	0.9455
50	0.7709	0.6221	0.8917	0.3805	0.5797	1.0975	0.6757	0.9141

Dropouts

CNN is used to build two models with dropout rates of 0.5 and 0.25. The model is initially trained using 50 epochs with a dropout rate of 0.25 and an image size of 64*64. With a 0.001 learning rate and 32 batch size, Adam's optimizer was used. By doing this, they were able to get the highest accuracy possible; above 50, the validation accuracy and loss start to decline. Table 3.6 displays the training and validation accuracy and loss for the two models with 0.25 and 0.5 dropouts for the various epochs. The graphs in Figure 3.15 (a) and (b) show the accuracy and loss for the two models with various dropout values, and they draw the conclusion that the accuracy and loss reached with 0.25 dropout is better than 0.5 dropouts with the increase in epochs. The graphs also show that the validation accuracy is always higher than the training accuracy when dropout is used to reduce overfitting. The validation loss, however, is always smaller than the training loss. It is clear from this, that the model with a 0.25 dropout is currently

the best classifier model and will be utilized for other hyper-parameter adjustments over the course of the same 50 epochs.



(a) (b)

Figure 3.15: (a) Graph of training and validation accuracy for 0.5 and 0.25 dropout (b) Graph of training and validation loss for 0.5 and 0.25 dropout

Varying densities of neurons in different dense layer

To find the optimum model, various fully linked network layers with various numbers of neurons are explored.

Single layer with 64 neurons: To create a model, experiments are conducted with a single layer of a network with 64 completely linked neurons. In Table 3.7, the accuracy and loss obtained during training and validation are listed.

Table 3.7: Accuracy and loss measures with 64 neurons in a single fully connected layer

No. of Epochs	64 neurons			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.5512	1.2238	0.6230	1.0750
10	0.6051	1.0786	0.6857	0.8842
15	0.6348	0.9880	0.7093	0.8140
20	0.6633	0.9052	0.7330	0.7570
25	0.6830	0.8472	0.7603	0.6779
30	0.7077	0.7767	0.7980	0.5654
35	0.7207	0.7421	0.8100	0.5625
40	0.7463	0.6749	0.7970	0.5561
45	0.7563	0.6597	0.8460	0.4607
50	0.7709	0.6221	**0.8917**	**0.3805**

Single layer with 128 neurons: A fully connected network including 128 neurons on a single dense layer is experimented with to produce a model. The accuracy and loss produced with different epochs are shown in Table 3.8, illustrating results that are better than the model with 64 neurons.

Table 3.8: Accuracy and loss measures with 128 neurons in a single fully connected layer

No. of Epochs	128 neurons			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.6307	1.0232	0.6713	0.9158
10	0.7100	0.7982	0.7710	0.6632
15	0.7510	0.6766	0.7820	0.6009
20	0.7829	0.5880	0.8640	0.4244
25	0.8102	0.5014	0.8653	0.3842
30	0.8382	0.4305	0.9043	0.2880
35	0.8507	0.4065	0.9263	0.2486
40	0.8607	0.3729	0.9293	0.2166
45	0.8767	0.3375	0.9373	0.2024
50	0.8843	0.3204	**0.9450**	**0.1680**

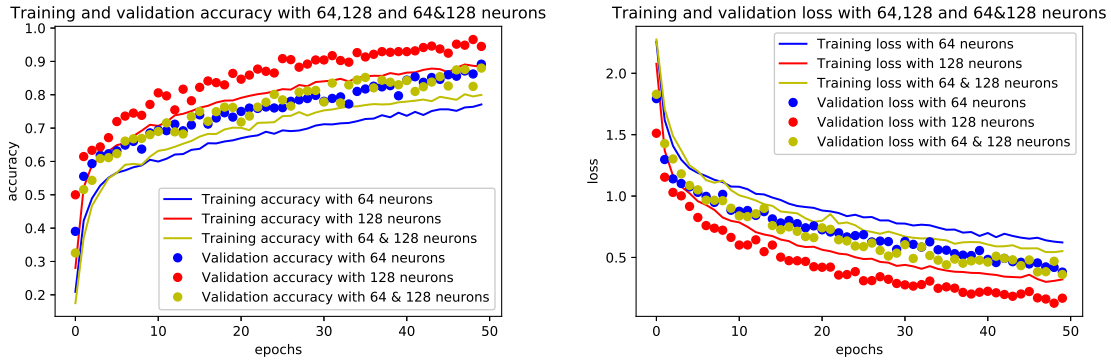
Table 3.9: Accuracy and loss measures with 64 and 128 neurons in a single fully connected layer

No. of Epochs	64 and 128 neurons			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.5494	1.2464	0.6113	1.0864
10	0.6143	1.0492	0.6803	0.9010
15	0.6614	0.9218	0.7343	0.7597
20	0.7012	0.7897	0.7620	0.6615
25	0.7177	0.7643	0.8010	0.5929
30	0.7468	0.6963	0.8130	0.5349
35	0.7652	0.6460	0.8443	0.4779
40	0.7773	0.6226	0.8300	0.4776
45	0.7939	0.5747	0.8543	0.4334
50	0.7996	0.5505	**0.8800**	**0.3615**

Two layers with 64 and 128 neurons: To create a model, experiments are conducted with 64 and 128 neurons over two layers of a fully connected network. On increasing

the epochs, accuracy and loss are seen to increase and decrease, respectively, as shown in Table 3.9. The accuracy and loss of the results are better with a single layer of a fully connected network of 128 neurons, though.

In comparison to the other two models, the model with a single layer of a fully connected network comprising 128 neurons and 50 epochs has achieved greater accuracy and minimal loss, as shown by the three tables and graphs in Figure 3.16 (a) and (b).



(a) (b)

Figure 3.16: (a) Graph of training and validation accuracy with 64, 128 and 64 and 128 neurons in a single fully connected layer (b) Graph of training and validation loss with 64, 128 and 64 and 128 neurons in a single fully connected layer

Based on the aforementioned findings, the model with a single fully connected layer of 128 neurons, dropout of 0.25, 64*64 picture size, a learning rate of 0.001 using Adam's optimizer, batch size of 32, and less than 50 epochs will undergo the following hyper-parameter tuning.

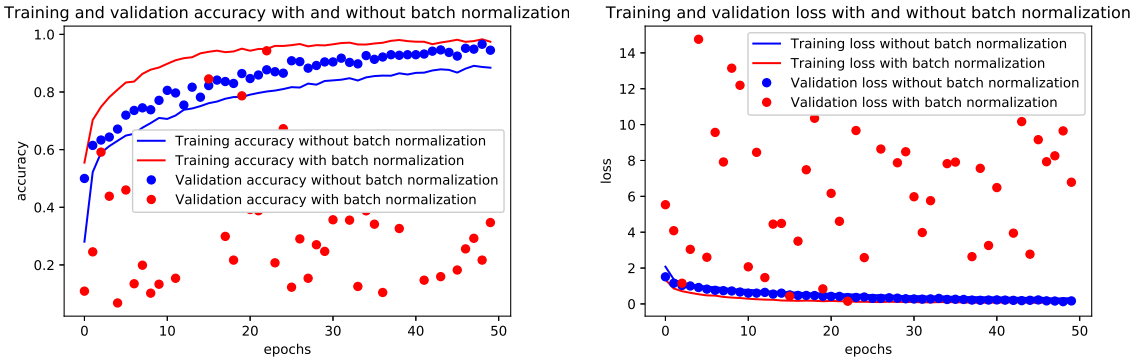
Batch Normalization

The above-mentioned model was subjected to batch normalization after the convolutional layer and before the ReLU activation function. Results are provided in Table 3.10 for accuracy and loss. The table shows that the model's loss and training accuracy change with the number of epochs. According to Figures 3.17 (a) and (b), validation accuracy and loss are still out of balance and poor. Because of its unstabilized nature, the model is still not much better than the preceding one.

As a result, taking into account the aforementioned outcome, subsequent hyper-parameter tuning will be done on an identical model with 128 neurons and 0.25 dropout.

Table 3.10: Accuracy and loss measures with batch-normalization

No. of Epochs	Batch Normalization			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.8081	0.5436	0.0680	14.7580
10	0.8850	0.3324	0.1330	12.1920
15	0.9342	0.1933	0.4477	4.4896
20	0.9505	0.1511	0.7867	0.8404
25	0.9597	0.1260	0.6727	2.5830
30	0.9617	0.1168	0.2470	8.4864
35	0.9647	0.1055	0.3877	7.8231
40	0.9765	0.0709	0.5733	3.2589
45	0.9760	0.0753	0.6203	2.7733
50	**0.9744**	**0.0897**	0.3470	6.7862



(a) (b)

Figure 3.17: (a) Graph of training and validation accuracy with and without batch normalisation (b) Graph of training and validation loss with and without batch normalisation

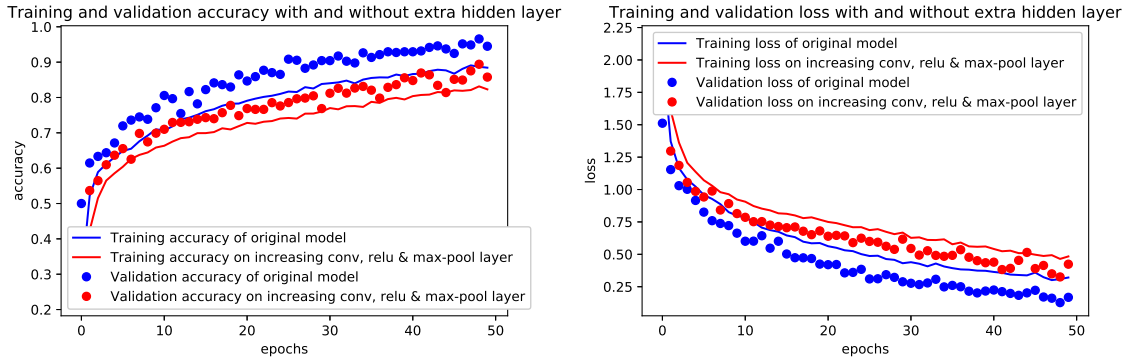
Extra Hidden layer

Convolution, ReLU, max-pooling, and dropout are added to the hidden layer to boost its effectiveness. Table 3.11 lists the results from training and validation after this enhancement. The table demonstrates that the model's accuracy and loss during training and validation are worse than those of the previous best-identified model. Figure 3.18 (a) and (b) provide a comparison between the earlier model with a single fully connected layer of 128 neurons, involving 0.25 dropout and less than 50 epochs, and illustrates that adding more convolution, max-pooling, and dropout layers do not result in enhanced accuracy and loss.

Based on the results discussed above, the prior model that was chosen as the best will undergo additional hyper-parameter tuning.

Table 3.11: Accuracy and loss measures with one additional layer of conv, ReLU, and max-pooling

No. of Epochs	Increasing Hidden layer			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.5856	1.1384	0.6367	0.9859
10	0.6584	0.9241	0.6990	0.8145
15	0.6991	0.8157	0.7383	0.7146
20	0.7190	0.7675	0.7487	0.6806
25	0.7404	0.7084	0.7760	0.6244
30	0.7601	0.6672	0.7687	0.6167
35	0.7733	0.6156	0.8313	0.4846
40	0.7944	0.5637	0.8557	0.4351
45	0.8140	0.5080	0.8140	0.5147
50	0.8230	0.4858	**0.8577**	**0.4238**



(a) (b)

Figure 3.18: (a) Graph of training and validation accuracy with and without the additional hidden layer (b) Graph of training and validation loss with and without the additional hidden layer

Image Size

The classifier accuracy is based on the image resolution. During training, three distinct image sizes: 64*64, 128*128, and 256*256-are used for experimentation. The training up until this point used the 64*64 image size, which was then tested with 128*128 and 256*256. After this, the resolution is not increased due to the need for more training time and the high computational demand on the CPU.

128*128 Image size: Table 3.12 provides the outcomes of a model trained using a 128*128 image size alteration. In comparison to the previous best-found model, the validation accuracy and loss are essentially comparable.

Table 3.12: Accuracy and loss measures with 128*128 image resolution

No. of Epochs	image size 128*128			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.6644	0.9238	0.7183	0.8425
10	0.7492	0.6946	0.7953	0.6172
15	0.8004	0.5392	0.8500	0.4671
20	0.8291	0.4615	0.8913	0.3641
25	0.8546	0.3884	0.9033	0.3209
30	0.8797	0.3265	0.9200	0.2546
35	0.8919	0.2955	0.9390	0.2143
40	0.9021	0.2634	0.9517	0.1689
45	0.9107	0.2499	0.9543	0.1662
50	0.9160	0.2270	**0.9430**	**0.1792**

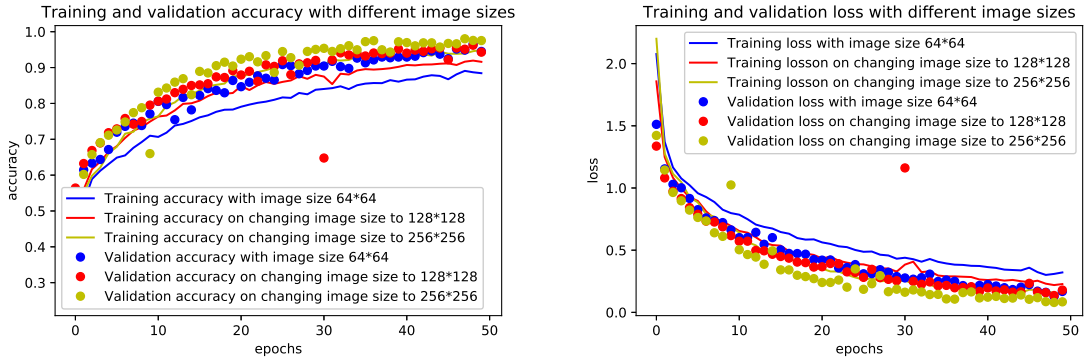
Table 3.13: Accuracy and loss measures with 256*256 image resolution

No. of Epochs	image size 256*256			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.6613	0.9245	0.7113	0.8222
10	0.7575	0.6847	0.6600	1.0245
15	0.8259	0.4718	0.8243	0.4946
20	0.8761	0.3492	0.9043	0.2712
25	0.8993	0.2800	0.8863	0.3494
30	0.9134	0.2441	0.9507	0.1860
35	0.9309	0.1927	0.9497	0.1444
40	0.9310	0.2010	0.9720	0.1158
45	0.9444	0.1585	0.9573	0.1397
50	0.9488	0.1504	**0.9753**	**0.0846**

256*256 Image size: Table 3.13 provides the findings of a model that was trained by varying the 256*256 picture size. Tables 3.8, 3.12, and 3.13 show that this model's validation accuracy and loss are superior to those of the other two models.

Figures 3.19 (a) and (b) provide a comparison of the accuracy and loss for the three

models with image sizes of 64*64, 128*128, and 256*256. According to the provided graphs, the three models' accuracy and loss under 256*256 image resolution are the best. This model is therefore considered to be an improved model and will be looked at for additional hyper-tuning.



(a) (b)

Figure 3.19: (a) Graph of training and validation accuracy with 64*64, 128*128 and 256*256 image pixel resolutions (b) Graph of training and validation loss with 64*64, 128*128 and 256*256 image pixel resolutions

Learning Rates

With a 0.001 learning rate, Adam's optimizer is initially used to create the model. By using 0.0001 and 0.00001 in place of the original learning rate, two additional new models are created. These are the typical learning rates used by various optimizers.

Learning Rate 0.0001 Table 3.14 shows the accuracy and loss that the model with a learning rate of 0.0001 achieved. The learning rate of 0.0001 is determined to be inferior in accuracy and loss under the 50 epochs listed in the table, including Adam's optimizer.

Learning Rate 0.00001 Table 3.15 displays the accuracy and loss of the model with a 0.00001 learning rate. The table below shows the model accuracy and loss with 50 epochs, Adam's optimizer, and a learning rate of 0.00001; the accuracy is lower and the loss is higher compared to the prior model with a 0.0001 learning rate.

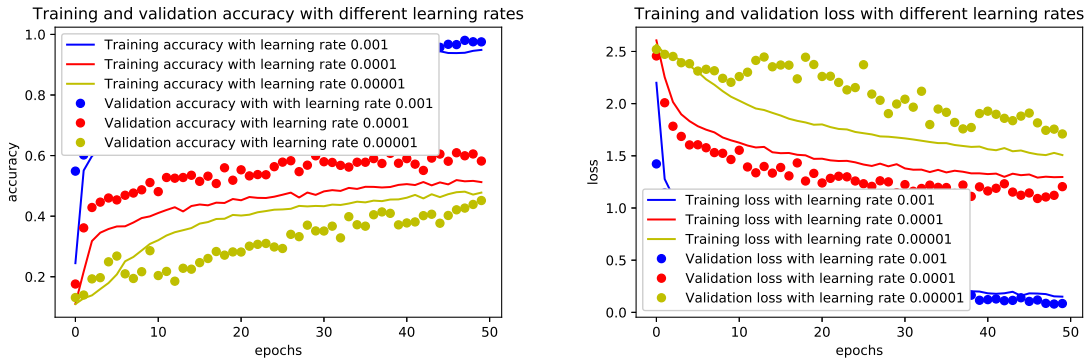
Table 3.14: Accuracy and loss measures with learning rate 0.0001 employing Adam's optimizer

No. of Epochs	0.0001 learning rate			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.3577	1.8334	0.4600	1.6044
10	0.3994	1.6474	0.5110	1.4647
15	0.4335	1.5348	0.5350	1.3344
20	0.4566	1.4716	0.5187	1.3316
25	0.4643	1.4377	0.5633	1.2587
30	0.4703	1.4195	0.5993	1.1463
35	0.4900	1.3682	0.5780	1.1977
40	0.5032	1.3317	0.5747	1.2025
45	0.5115	1.3141	0.6050	1.1220
50	0.5125	1.2960	0.5820	1.2046

Table 3.15: Accuracy and loss measures with 0.00001 learning rate employing Adam's optimizer

No. of Epochs	0.00001 learning rate			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.1794	2.3654	0.2493	2.3829
10	0.3085	2.0670	0.2860	2.2045
15	0.3607	1.9129	0.2250	2.3531
20	0.4041	1.7975	0.2820	2.3749
25	0.4214	1.7291	0.2983	2.1540
30	0.4338	1.6761	0.3513	1.9965
35	0.4478	1.6244	0.3713	1.9480
40	0.4554	1.5877	0.3713	1.9048
45	0.4628	1.5515	0.3767	1.8804
50	0.4778	1.5074	0.4517	1.7096

The three models with learning rates of 0.001, 0.0001 and 0.00001 are represented by the graphs in Figures 3.20(a) and (b), respectively, which show the accuracy and loss acquired under each model. During training and validation, with 0.001 learning rate model gives improved results in terms of accuracy and loss. Based on the aforementioned findings, the model described above will undergo additional hyper-parameter tuning with a learning rate of 0.001, an image size of 256*256, a batch size of 32 and under 50 epochs.



(a) (b)

Figure 3.20: (a) Graph of training and validation accuracy for learning rates 0.001, 0.0001 and 0.00001 (b) Graph of training and validation loss for learning rates 0.001, 0.0001 and 0.00001

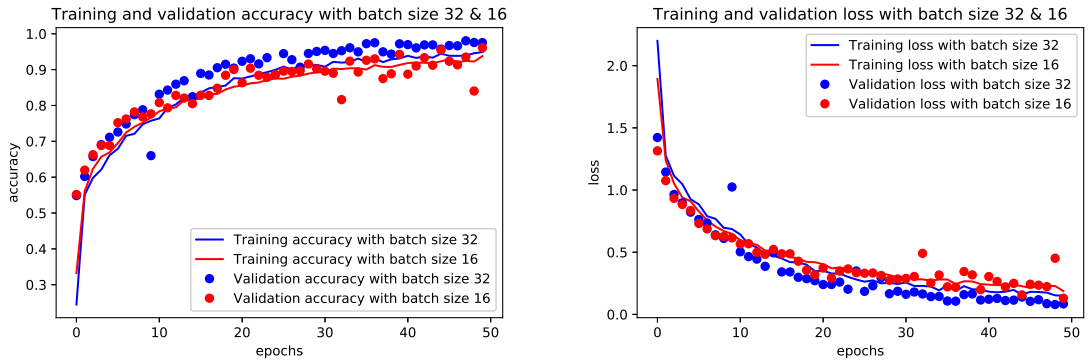
Batch size

Table 3.16 displays the accuracy and loss that were obtained when the batch size was changed to 16. The table shows that batch size 16 has superior accuracy and loss than the other models, with a validation accuracy of around 96%. However, compared to batch size 32, which is provided in the best-found model, validation accuracy and loss are relatively modest. 16 photos are processed in an iteration with an enhanced number of epochs when the batch size is 16.

Table 3.16: Accuracy and loss measures with batch size 16

No. of Epochs	batch size of 16			
	Training accuracy	Training loss	Validation accuracy	Validation loss
5	0.6690	0.9099	0.6883	0.8354
10	0.7667	0.6469	0.7767	0.6152
15	0.8151	0.5127	0.8050	0.5221
20	0.8523	0.4195	0.9003	0.3184
25	0.8753	0.3476	0.8860	0.3355
30	0.8926	0.2983	0.9020	0.2848
35	0.9036	0.2817	0.8933	0.3170
40	0.9132	0.2435	0.9427	0.1989
45	0.9231	0.2262	0.9550	0.1569
50	0.9376	0.1867	**0.9610**	**0.1301**

Figure 3.21 (a) and (b) compare the accuracy and loss achieved during training and validation for the two models with batch sizes 16 and 32, respectively. The accuracy achieved under batch size 16 is slightly less than that achieved under batch size 32, as seen by the two graphs.



(a) (b)

Figure 3.21: (a) Graph of training and validation accuracy with 16 and 32 batch sizes (b) Graph of training and validation loss with 16 and 32 batch sizes

As a result of all the aforementioned hyper-parameter tuning, it has been determined that the best model has 3 convolutions, 3 ReLU, 2 max-pooling, 1 fully connected layer with 128 neurons, dropout of 0.25, batch size 32, a learning rate of 0.001 and input image size of 256*256. This model achieves the highest accuracy and the least amount of loss. Table 3.13 and Figures 3.19 (a) and (b) provide illustrations of the accuracy and loss of the best model.

Figure 3.22 provides a diagrammatic illustration of the suggested approach that is most effective for classifying wheat varieties. The flowchart for the proposed Agri-CNN shows how a wheat input image with a size of 1800*1650 is reduced to 256*256 and sent to the first layer of the CNN model as input. A complete plot of the model for wheat varietal categorization that was determined to be the best and most effective throughout experimentation is displayed in Figure 3.23 architecture. It demonstrates how the scale of the image changes when you transition between layers, including convolutional, max-pooling, dropout, and dense layers. The graphic describes each layer's input and output based on image size, the number of filters, and the number of neurons.

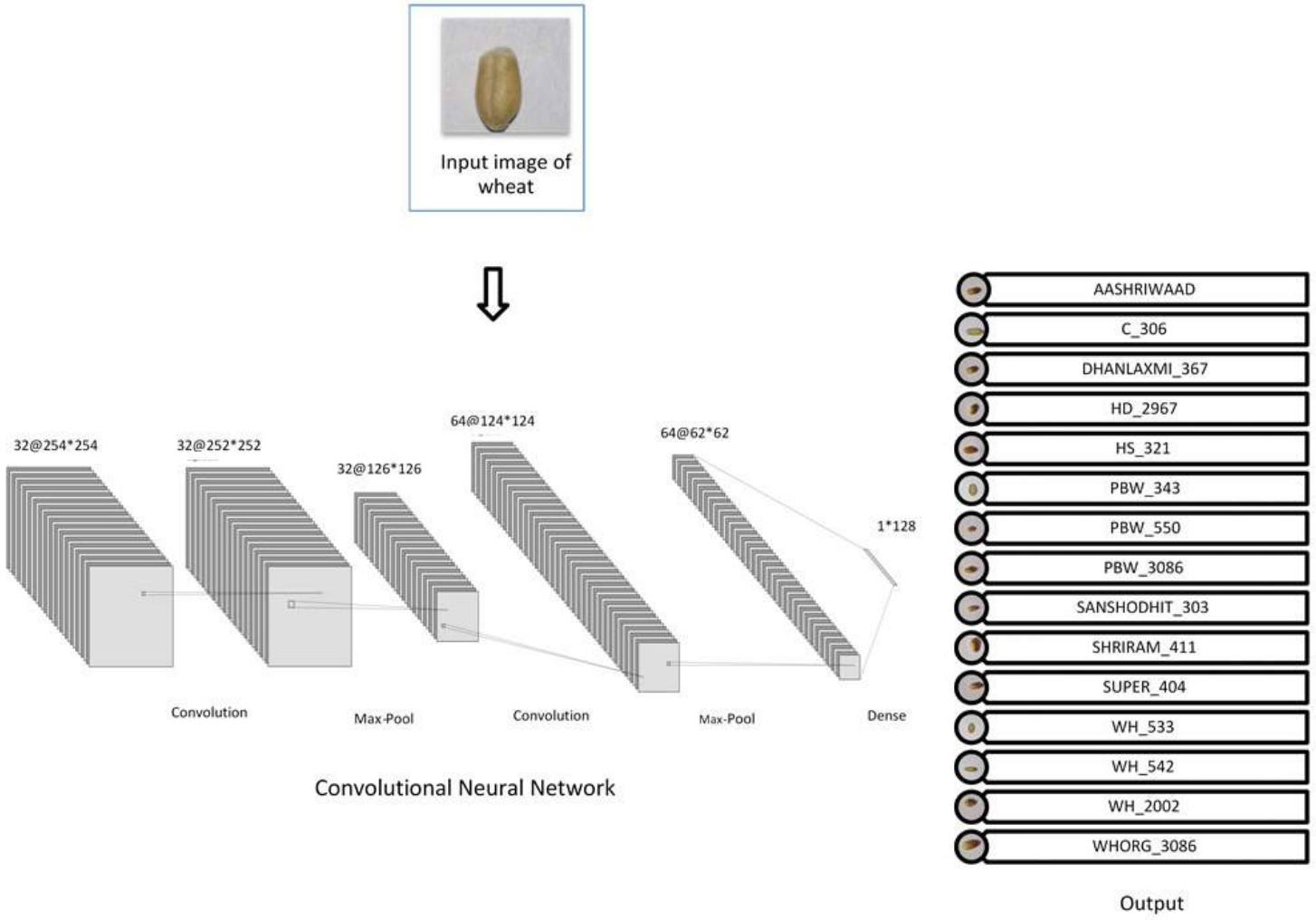


Figure 3.22: Diagrammatic representation of "Agri-CNN" model

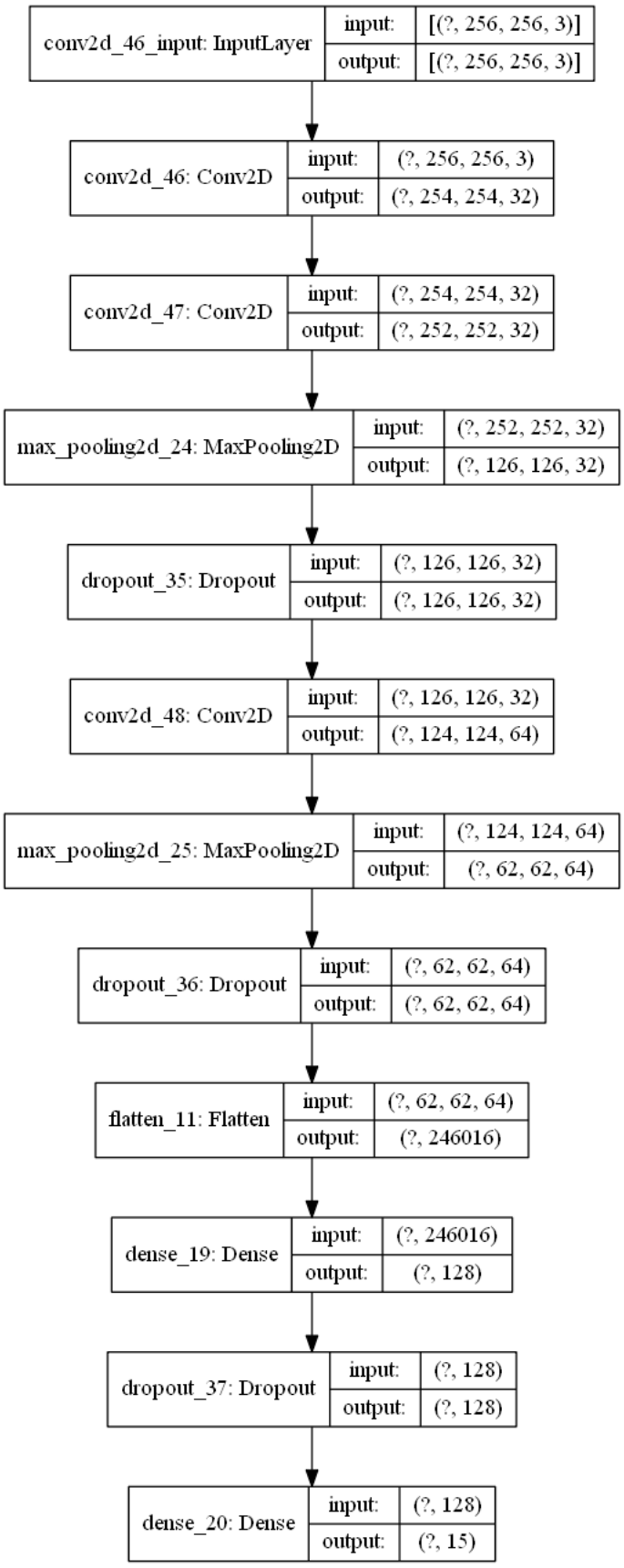


Figure 3.23: Layered structure of the proposed "Agri-CNN" model

3.5.4 Model Evaluation and Results

Figures 3.22 and 3.23 show the best CNN-based model for classifying wheat varieties after all tuning of hyper-parameters. Table 3.13 and Fig. 3.19 (a) and (b) provide the model accuracy and loss information. The model is then further developed and tested for accuracy by using it to predict the right class of image out of fifteen possible classes for the validation set of 1500 photos. A comparison of the actual class of validation images and the anticipated class is shown in Figure 3.24. Figure 3.25 shows the metrics report for the 1500 images in the validation set, including precision, recall, F1-score, and support. The resulting accuracy is 97%.

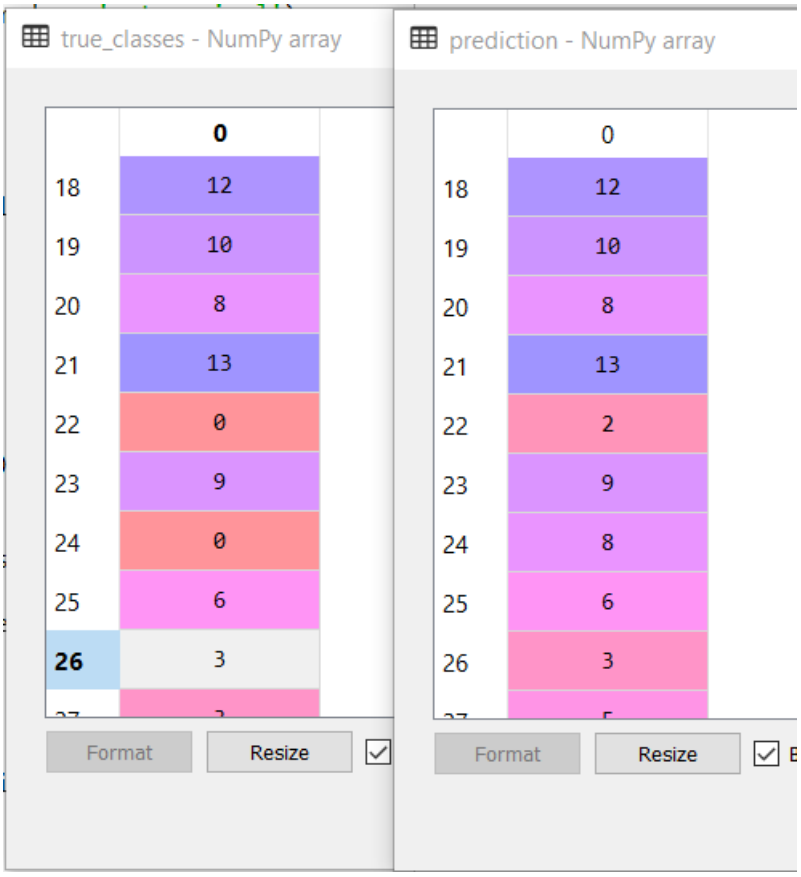


Figure 3.24: Prediction result of 1500 images of validation set

Confusion matrix: The confusion matrix or error matrix, is presented as a table that compares the predicted classes of validation data against the true classes for the same input data to assess the performance of a classification model. The confusion matrix for our validation data of 1500 randomly selected images is displayed in Figure 3.26 below. The first column of the matrix contains the predicted labels for the wheat class, whereas

Text editor - report

	precision	recall	f1-score	support
AASHRIWAAD	0.98	0.85	0.91	100
C_306	1.00	1.00	1.00	100
DHANLAXMI_367	0.91	0.99	0.95	100
HD_2967	0.97	0.94	0.95	100
HS_321	0.89	0.96	0.92	100
PBW_3086	0.98	0.99	0.99	100
PBW_343	1.00	1.00	1.00	100
PBW_550	0.97	0.99	0.98	100
SANSHODHIT_303	0.95	0.90	0.92	100
SHRIRAM_411	1.00	1.00	1.00	100
SUPER_404	0.96	0.97	0.97	100
WHORG_3086	0.99	0.97	0.98	100
WH_2002	0.96	0.98	0.97	100
WH_533	1.00	1.00	1.00	100
WH_542	1.00	1.00	1.00	100
accuracy			0.97	1500
macro avg	0.97	0.97	0.97	1500
weighted avg	0.97	0.97	0.97	1500

Figure 3.25: Model performance of test data

the first row of the matrix has the actual labels from 0 to 14. Table 3.17 displays the labels for the various classes of wheat. As a result, given the validation data, the model accuracy is found to be close to 97 percent.

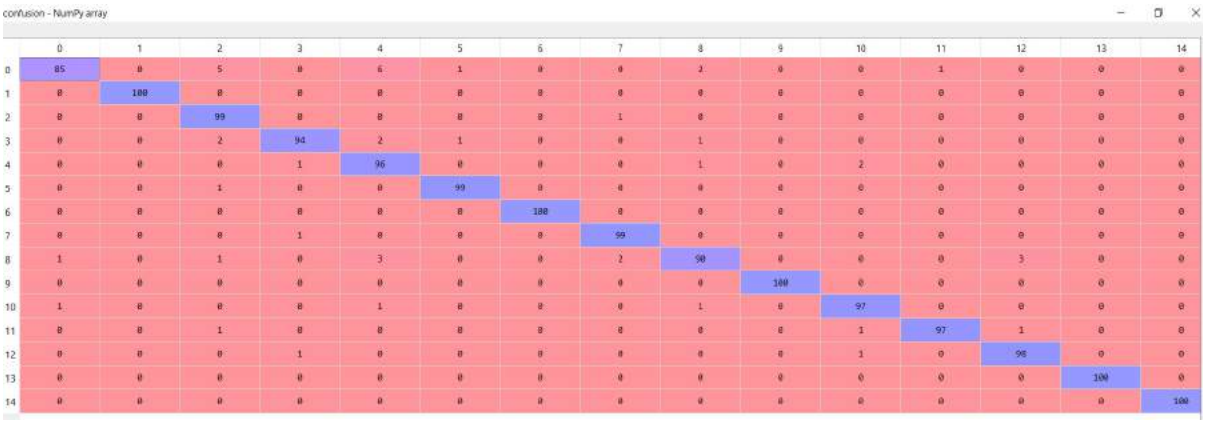


Figure 3.26: Confusion matrix of wheat classes on validation set

3.5.5 Performance Evaluation

The suggested Agri-CNN model's performance is compared with the VGG-16 transfer learning model, which has produced better results earlier for plant disease identification. It has been identified that our Agri-CNN model has given better results for grain clas-

Table 3.17: Classes of wheat with label

Class of wheat grains	Label Number
AASHRIWAAD	0
C_306	1
DHANLAXMI_367	2
HD_2967	3
HS_321	4
PBW_343	5
PBW_550	6
PBW_3086	7
SHRIRAM_SUPER_303	8
SHRIRAM_SUPER_411	9
SHRIRAM_SUPER_404	10
PBW_533	11
WH_542	12
VL_829(2002)	13
WHORG_3086	14

sification than VGG-16, as shown in Table 3.18. The Agri-CNN has performed better with 97% accuracy for large image datasets with reduced complexity, unlike VGG-16, which has shown only 93% accuracy. Furthermore, based on the various methods used to categorize the grains, the achievements of the proposed "Agri-CNN" is compared with the other models, as shown in the literature review. Table 3.19 gives a comparison of the techniques. As a result, our suggested model correctly categorized fifteen wheat types with 97.53% test accuracy and 97% validation accuracy. It can be seen from the table that the proposed "Agri-CNN" model performs better than the majority of the techniques of literature. In contrast to manually designed feature extraction which requires expertise and labor, our model performs automatic feature extraction, which makes the classification task simpler.

Table 3.18: Comparison of accuracy and loss generated through proposed "Agri-CNN" and VGG-16 for wheat grains classification

No. of epoch	Training accuracy	Training loss	Validation accuracy	Validation loss
Proposed Agri-CNN	0.9488	0.1504	0.9753	0.0846
VGG-16	0.9212	0.3214	0.9345	0.1214

Table 3.19: Comparative analysis of various grain classification methods

Paper	No. of classes	Varieties	No. of images	Feature extraction techniques	Classification techniques	Accuracy
[36]	2	Wheat	200	Hand-crafted feature extraction	ANN	99.9%
[38]	30	Rice	1500	Hand-crafted feature extraction	SRC	89.1%
[61]	3	Soybean leaves, Red bean leaves, White bean leaves	866	Hand-crafted feature extraction	SVM (Gaussian kernel); SVM (linear kernel); Random forests; PDA	89.57%; 89.77%; 87.87%; 89.97%
[9]	8	Acacia seeds	200	Hand-crafted feature extraction	Discriminant Analysis	79.6%
[42]	5	Rice		Hand-crafted feature extraction	Multi-layer perceptron; Neuro-fuzzy neural networks	98.40%; 99.73%
[43]	2	Wheat	200	Hand-crafted feature extraction	Adaptive neuro-fuzzy inference system (ANFIS)	99.46%

Continued on next page

Table 3.19 – *Continued from previous page*

Paper	No. of classes	Varieties	No. of images	Feature extraction techniques	classification techniques	Accuracy
[30]	7	Rapeseeds	525	Hand-crafted feature extraction	SVM,KNN,SGD; KNN; SGD	100%; 92.40%; 94.30%
Agri-CNN	15	Wheat	15000	Automatic feature extraction	CNN	98%

3.6 CHAPTER SUMMARY

The convolutional neural network, one of the most promising deep learning techniques, has been used in this work as the framework for the development of additional models. The process for creating an image identification and classification model is covered in this chapter. This chapter has presented two types of the classification process. One performs the classification of two plant diseases-tomatoes and potatoes, where data was collected from the PlantVillage repository. To estimate the precision of the classification model for the two datasets, the study used transfer learning models based on CNN. This work aims to develop a plant disease classification system and also to study the effectiveness of the transfer learning model used on small and unbalanced datasets. ResNet 50, DenseNet 121, VGG 16 and Inception-v3 are the four transfer learning models applied to the dataset. VGG-16 is identified as the most effective model among all and is then used to classify the wheat grain varieties. It is found that the transfer learning based models are not so effective for classification where the training dataset is small. This limited and unbalanced data condition sometimes leads to overfitting and reduce precision. To overcome the problem of small data size, a self-generated dataset of wheat grains is developed where the data size is 15000 images covering 15 varieties of wheat. This self-generated data is then subjected to transfer learning by the most effective transfer model VGG-16. A model called "Agri-CNN" is also developed from scratch for the classification of wheat grain varieties and the

output of "Agri-CNN" is later contrasted with the VGG-16 transfer learning model. It has been found that the proposed model (Agri-CNN) outperforms the transfer model VGG-16 with 98% accuracy and a minimum loss of 0.0846 even for large datasets. The findings have also shown that the image resolution also significantly affects classification accuracy because, occasionally, photos with higher resolution and input size impose a considerable computational load on the network's initial layers, resulting in a long and expensive training process. Furthermore, handling huge batch sizes is again a challenging task.

CHAPTER IV

DESIGN OF A HYBRID TECHNIQUE BASED ON INFORMATION GAIN FOR DIMENSIONALITY REDUCTION

CHAPTER 4

DESIGN OF A HYBRID TECHNIQUE BASED ON INFORMATION GAIN FOR DIMENSIONALITY REDUCTION

To prevent overfitting and thereby improve classification accuracy, it is necessary to identify and choose the right number of features for the correct classification of grain varieties and plant diseases.

4.1 INTRODUCTION

It is seen at times that the performance of a model can occasionally get negatively impacted by the presence of too many irrelevant features [152], while on the other side, a substandard model could do better in terms of the availability of a less but good set of features [153]. Therefore, considering such issues, feature extraction and feature selection are considered the key process to be applied as a pre-processing step for the classification and regression methods. Feature selection focuses on eliminating undesirable features and selecting a subset of the original features, whereas feature extraction is a method of extracting a compressed collection of features from the actual data by transforming, gathering and structuring the input features into newly derived features [154, 155]. Wrapper and filter methods [156] are the two majorly employed supervised learning techniques for feature selection that uses functionalities of Shannon Entropy and the Chi-Square formula [157]. To extract the features, principal component analysis is generally used in a variety of domains, including data mining, pattern recognition, machine learning and artificial intelligence as a feature extraction process [156].

By storing the features with more significant information and variation and disregarding the features with less information and variance, PCA reduces the relevance of the characteristics. These contributing features are identified later from a linear combination of original features [153].

Since the data is rising sharply, it becomes challenging to analyze and visualize the large volume of data for valid inferences. Thus, this work has developed a hybrid model "Info_PCA" for dimensionality reduction and better data analysis and visualization with limited space requirements. This "Info_PCA" model has combined the mutual information gain produced through Shannon entropy with PCA to fuse the functionalities of both feature selection and feature extraction process, considering the features that were shown to have the greatest separability and the least correlation between the classes. To check the effectiveness of this proposed scheme of dimensionality reduction, the reduced data is then used for classification. The proposed classification model is evaluated on the two common datasets selected from the UCI machine learning repository [158], namely the iris and leaf data set [159], where the iris dataset consists of three class label−Setosa, Versicolor and Virginia, while the leaf dataset includes shape and texture attributes classified into 30 different categories. To further check the efficiency and strength of the Info_PCA model, its classification accuracy is compared with three other methodologies which are also structured to classify the given datasets. These methodologies are defined as:

- The first method involves just the simple application of an ANN classifier to classify the two datasets.
- The second method initially performs the feature selection on the dataset through mutual information gain, which is followed by an ANN classifier to classify the data.
- The third method performs the classification of data through an ANN classifier after the process of feature extraction using PCA.
- The fourth method is itself the "Info_PCA" model applied on the dataset for dimensionality reduction which is later followed by the ANN classifier for classification.

During the entire training and learning process, it has been identified that the proposed Info_PCA has outperformed in terms of classification accuracy compared to the other three models.

The reduction process in the dimensionality of data makes use of various techniques that consist of feature selection and feature extraction processes. Therefore, in the following sections, some of the dimensionality reduction techniques and their functionalities are introduced.

4.2 OVERVIEW OF THE DIMENSIONALITY REDUCTION TECHNIQUES

The method for reducing the quantity of training input variables is called dimensionality reduction. The process of dimensionality reduction transfers data from a high-dimensional feature space to one with fewer dimensions without eliminating any of the information [4]. Several variables called features are typically used to encompass procedures like classification and regression. The complexity of model creation increases as the number of characteristics increases; this is known as the "curse of dimensionality." Additionally, the dataset becomes noisier due to the redundant nature of the features, which raises the complexity that needs to be reduced. Contrary to this, the presence of fewer dimensions makes a simpler machine model with few parameters called the degree of freedom, which eventually results in the overfitting of data and inaccurate performance for unknown and new data. Therefore, the dimensionality reduction technique must be carried out on the training and test or validation data before developing a model as a pre-processing step, executed after cleaning and scaling of data. Dimensionality reduction is often used for data visualization purposes to analyze and simplify the data [126]. The dimensionality reduction process can be classified into different categories, as shown in Figure 4.1. These reduction techniques are broadly categorized as feature selection and feature extraction. The following section will give a brief introduction to the functionalities of these two techniques.

4.2.1 Feature Selection

Feature selection is detecting and identifying appropriate features from the dataset. Sometimes, the collected feature set includes a large number of features where some of them are found irrelevant in affecting the dependent variable [154]. This set of features

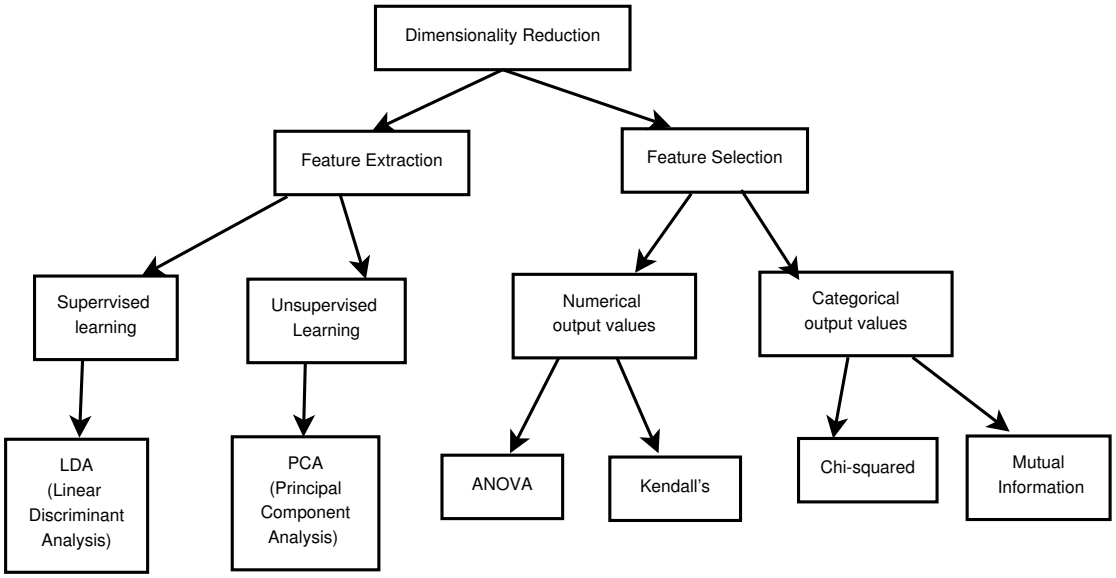


Figure 4.1: Dimensionality reduction techniques

needs to be detected and removed from the dataset through the process called feature selection. Heatmap is one of the feature selection tools that relate the correlation between features by plotting them against the target variable [152]. Some methods for feature selection are Variance Threshold and Univariate selection. Variance Threshold works by dropping all those features where the variance along the column lies below a given threshold value. Univariate Feature Selection gives the observation for a single attribute and feature. The process selects the features using a statistical test. Some statistical tests involved under univariate feature selection are Pearson Correlation, ANOVA, Chi-square, maximal information coefficient and Distance correlation. A method like ANOVA is preferred for continuous numerical values, while Chi-square is used for categorical values. The feature selection techniques can also be categorized based on the data it is subjected to. These are classified as numerical values and categorical values. Numerical output values include numerical values like height and weight and categorical values like labels. The numerical output values are used for regression purposes. It includes methods like ANOVA and Kendall's. Categorical output values include labels like 1,2,3, and yes or no, which makes it categorical data. The categorical output values are used for classification purposes. It includes feature selection methods like Chi-squared and Mutual Information.

4.2.2 Feature Extraction

A new, smaller feature subspace is created by the feature extraction procedure, which involves taking information from the original feature space. The process compresses the information in the data to extract the most relevant out of it [155, 160]. This helps in reducing the model complexity, absolute error and overfitting of data and makes the system more efficient. There are supervised and unsupervised methods of feature extraction. Supervised learning measures the statistical relationship between independent variables using a labeled training dataset. Classification comes under supervised learning, where the classifier predicts the output class of a new data point. One of the effective methods under supervised learning is Linear Discriminant Analysis. LDA becomes the best classifier when the size of the data sample is large and the dimensionality of the feature space is low [161]. LDA [160] is based on the idea that the best classifier employs both mean and variance when the data has a gaussian distribution, which contains all the information in mean and variance. Unsupervised methods are typically used to reduce the dimensions when the number of variables is usually high. Unsupervised methods compress the features and improve efficiency while reducing overfitting. The goal of this learning strategy which excludes labels is to find the hidden structure in the unlabelled input data. One of the common techniques under the unsupervised learning technique is PCA. To create a set of linearly uncorrelated features known as principle components, the Principal Component Analysis (PCA) transforms the values of correlated features using an orthogonal transformation [97, 161]. It helps in extracting and compressing a large set of feature values to a smaller set, making a linear combination of original features. It generates various principal components where the first component gives the maximum variance.

The following section will discuss the two datasets used in the work for dimensionality reduction.

4.3 DATA

This work has involved two datasets collected from the UCI machine learning repository. These are the iris dataset and leaf dataset. Both datasets fall under the agricultural category of plants and vegetation.

Leaf dataset: The dataset includes illustrations of 40 different plant species' leaves, together with information about their shape and texture. The dataset has 16 characteristics, including the following: Class (Species), Specimen Number, Eccentricity, Aspect Ratio, Elongation, Solidity, Stochastic Convexity, Isoperimetric Factor, Maximal Indentation Depth, Lobedness, Average Intensity, Average Contrast, Smoothness, Third moment, Uniformity and Entropy. Pedro F.B. Silva, Andre R.S. Marcal and Rubim M. Almeida da Silva were the original developers of the dataset. The dataset is thought to be suitable for classifying values from real data.

Iris dataset: One of the well-known datasets that are most frequently used for pattern recognition and classification is the Iris dataset. The dataset consists of three classes with 50 occurrences of each: Setosa, Versicolor, and Virginica. Four features-petal lengths, petal width, sepal length and sepal width help to identify the data. Each class of iris plant belongs to a category where each class differs from the others. Data was generated by R.A. Fisher.

4.4 METHODOLOGY FOR DEVELOPING INFO_PCA

The entire work performed under this section is to design a new dimensionality reduction scheme, called "Info_PCA". Thus, to present the effectiveness of this proposed scheme, four different methodologies are used to classify the iris and leaf dataset. Figure 4.2 illustrates the entire procedure of four models. As can be seen from the diagram, the data is initially retrieved from the UCI machine learning repository and first subjected to data pre-processing. Later these pre-processed data proceeded further to construct the four models. These methodologies are explained separately in the following sections. Section 4.4.1 applies data directly to machine learning classifiers called Artificial Neural Networks (ANN), Section 4.4.2 works on feature selection before the ANN using Information gain, Section 4.4.3 processes feature extraction before the ANN classifier using PCA, and Section 4.5.4 implements the hybrid classifier Info_PCA. The classification accuracy obtained through these four methods is compared and it is found that Info_PCA outperforms the rest of the proposals. It may be noted that during all these implementations, the hyper-parameters of ANN are kept constant like the learning rate of 0.01, the number of hidden layers as 3, number of epochs as 500. The results derived

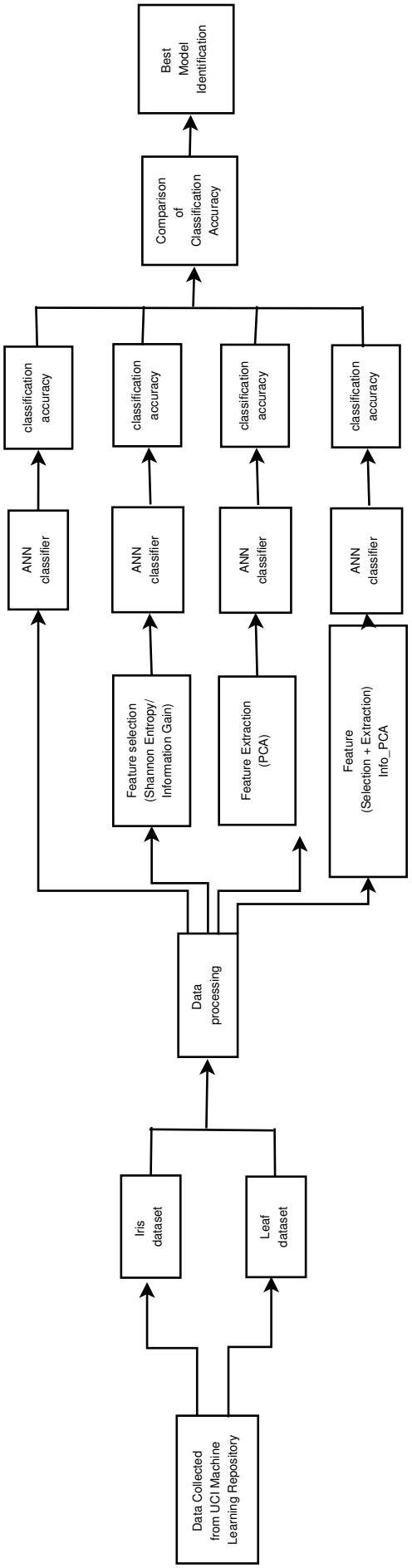


Figure 4.2: Different methodologies structured for classification

from each methodology are evaluated and compared as given in Sections 4.5 and 4.6. The following subsections will introduce the different models constructed for evaluating and checking the classification accuracy produced on the iris and leaf dataset. The proposed Info_PCA model is also presented and implemented in the coming subsection.

4.4.1 Simple application of ANN for classification

In this approach, the classification of the Iris and Leaf dataset is performed using artificial neural network (ANN) [162] without performing any feature selection or feature extraction scheme. An ANN is a feedforward backpropagation neural network containing many hidden layers in between the input and output layer [163]. The transfer function at each neuron in the hidden layer will receive the activation signal from its previous layers to produce non-linearity. In general, deep and practical learning and noise immunity are the two characteristics of ANN.

4.4.2 Information gain followed by ANN classifier

The objective of applying feature selection is to determine the ideal group of features for creating an appropriate model [8]. Since the filter methods are quicker and more affordable computationally, therefore they are used for feature selection. The filter method uses the Chi-square and Information Gain algorithms for the categorical dataset [164]. To identify the inherent characteristics of the features, univariate statistics are used.
In this section, the information gain for each variable influencing the dependent variable is estimated, and Shannon entropy-based information gain is used to find the independent features that contribute the most. The significant features that were found through information gain when applied to the leaf dataset are eccentricity, aspect_ratio, elongation, solidity, stochastic_convexity, isoperimetric_factor, maximum_indentation_depth, average_intensity, average_contrast, smoothness and third_moment. Petal_length, petal _width and sepal_length were the pertinent features identified for the Iris dataset.

4.4.3 PCA followed by ANN classifier

The relevancy of multiple features is compressed during feature extraction, which has a considerable effect on the dependent variable [165]. It reduces the number of features by creating a new set of features from the original data, by determining the key charac-

teristics of each feature. This will facilitate quicker training and avoid data overfitting and gives better data visualization and accuracy.

Principal Component Analysis [97], a linear dimensionality reduction technique is utilized in this work to extract features, which identifies the most important features from the complete dataset and compress them into several main components. To boost variance and reduce reconstruction error, PCA takes into account pair-wised distances. The data is exposed to a series of orthogonal axes, with values assigned to each axis depending on their relative importance. PCA is an unsupervised learning approach that places a strong emphasis on variance.

4.4.4 Info_PCA followed by ANN classifier

A hybrid technique called Info_PCA is proposed in this work for dimensionality reduction which will be employed to improve the classification accuracy for both training and test datasets. This method combines the qualities and abilities of Information gain and Principal Component Analysis. The model first passes the dataset to the filter method for feature selection, which involves mutual information gain to identify the level of relevancy of a feature for a dependent variable. These selected feature vectors with maximum information gain (IG) are then passed to Principal Component Analysis to generate the principal components, which are fed as inputs to the input layer of the ANN. The parameters of the ANN model are then set for the classification where the weights and bias are adjusted to produce the weighted sum of the output, iteratively to minimize the error. This entire work is defined in Algorithm 1 given below and the architecture of the proposed hybrid model "Info_PCA" is shown diagrammatically in Figure 4.3.

The following sections will discuss the implementation and experiments performed on the two datasets along with an analysis of the result generated.

4.5 EXPERIMENTS

The experiment is run on the leaf and iris datasets, collected from the UCI machine learning repository which falls under the category of plants and vegetation. Initially, the data is divided into training and test set, where 80% of data is passed as training while the rest is reserved for testing. Four distinct training models are applied to the two datasets for processing. The first model is created through the simple application of the ANN

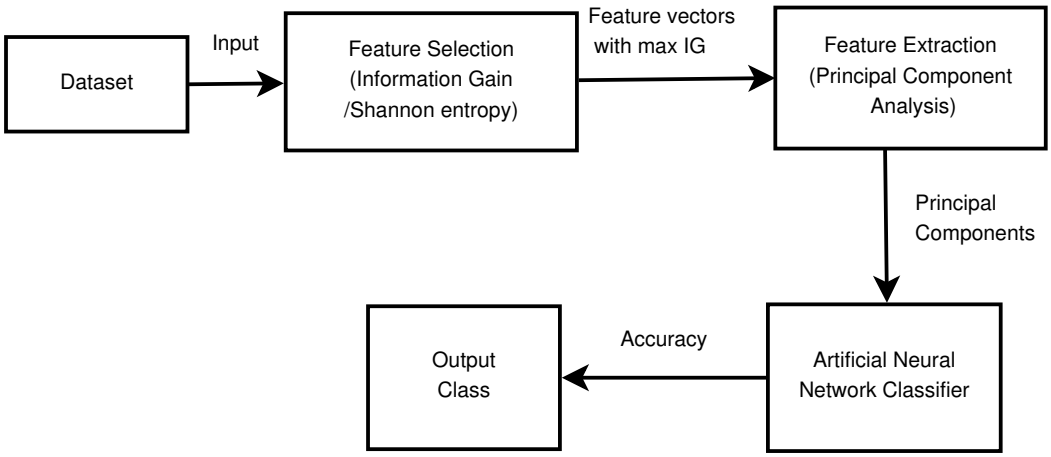


Figure 4.3: Block diagram of Hybrid model "Info_PCA"

Algorithm 1 Algorithm for Info_PCA model

1. Start.
2. Read the dataset.
3. Input the data to feature selection process through Information gain.
4. Pass the selected features to next phase.
5. Extract the compressed attributes from selected features through PCA.
6. Generate principal components.
7. Pass the principal components to ANN classifier.
   (a) Input the principal components to the ANN model
   (b) Set the learning rate.
   (c) Set the number of hidden layers.
   (d) Set the number of epochs.
   (e) Set the bias and weights of the input layer.
   (f) Calculate the weighted sum of the output layer.
   (g) Calculate the error.
   (h) Move to step 7(e).
   (i) Back propagate followed by feed forward until the error is minimized.
8. Generate the result.
9. Calculate the accuracy.
10. End.

model. The second model goes for feature selection through Information gain which ANN later follows for classification. The third model performs the feature extraction process using PCA and then classification by ANN. The fourth model applies our proposed hybrid model, Info_PCA, for dimensionality reduction and then classification is carried out through ANN. The accuracy and loss achieved through the four different models while training on the two datasets are shown in Table 4.1(a) and (b). It can be seen, from the tables that while training, our proposed hybrid model, "Info_PCA," performs more accurately than the other three models.

Table 4.1: Training accuracy and loss of two datasets

(a) leaf dataset

Techniques	Training Accuracy	Training loss
Simple ANN	0.7721	0.6522
IG followed by ANN	0.7878	0.5095
PCA followed by ANN	0.7868	0.6410
Info_PCA	0.8088	0.4834

(b) iris dataset

Techniques	Training Accuracy	Training loss
Simple ANN	0.8833	0.2875
IG followed by ANN	0.9250	0.2384
PCA followed by ANN	0.9333	0.2202
Info_PCA	0.9834	0.1356

4.6 RESULT ANALYSIS

Using test data from the iris and leaf dataset, the four distinct models are explored. The outcomes are assessed and validated using metrics including the confusion matrix, accuracy score, mean absolute error, classification report, precision, recall and F1 score. Table 4.2 and Table 4.3 shows the result of iris data on applying the four models, which were trained for 50 epochs using three classes. Table 4.2 (a), (b), (c) and (d) gives the confusion matrix for 30 test samples, which identifies the relevancy between the actual class and the predicted class, whereas the Table 4.3 (a), (b), (c) and (d) provides the test's overall accuracy score, mean absolute error and F1-score value along with the precision and recall achieved on iris data.

Table 4.4, Table 4.5 and Table 4.6 display the outcomes of the leaf dataset trained for 300 epochs with several classes. The results for 68 test samples utilizing the simple

Table 4.2: Confusion matrix for test samples of iris data

(a) Confusion matrix with Simple application of ANN

		act_class		
		Setosa	versicolor	virginica
Pred_class	Setosa	11	0	0
	versicolor	1	8	4
	virginica	0	1	5

(b) Confusion matrix using IG followed by ANN

		act_class		
		Setosa	versicolor	virginica
Pred_class	Setosa	11	0	0
	versicolor	1	9	3
	virginica	0	1	5

(c) Confusion matrix using PCA followed by ANN

		act_class		
		Setosa	versicolor	virginica
Pred_class	Setosa	11	0	0
	versicolor	1	10	2
	virginica	1	0	5

(d) Confusion matrix with Info_PCA

		act_class		
		Setosa	versicolor	virginica
Pred_class	Setosa	11	0	0
	versicolor	0	13	0
	virginica	0	0	6

application of ANN and Information gain followed by ANN are given by the confusion matrix of Table 4.4(a) and (b) which gives the relevance between the actual and predicted classes. Table 4.5(a) and (b) on the other hand, describe the confusion matrix produces by the application of PCA followed by ANN and the proposed Info_PCA model. Table 4.6 (a), (b), (c) and (d) provide the mean absolute error and overall accuracy score for the four models, along with identifying precision, recall and F1-score on the leaf data.

Thus, it can be identified from the estimated results given in the different tables for the iris and leaf datasets, that our proposed model Info_PCA has outperformed the other models in terms of accuracy and loss considering the different number of data instances and classes. It can be observed from the tables that the proposed Info_PCA has generated classification accuracy of 62.76% for leaf data and 100% for Iris data. The classification performed through ANN after the application of the Info_PCA model

Table 4.3: Accuracy evaluation of test samples of iris data

(a) Measures of test data accuracy using Simple ANN

	Precision	Recall	F1-score	Acc _Sc	Mean _Abs_Err
Setosa	1.00	1.00	1.00	24	0.122
versicolor	0.89	0.62	0.73		
virginica	0.56	0.83	0.67		

(b) Measures of test data accuracy using IG followed by ANN

	Precision	Recall	F1-score	Acc _Sc	Mean _Abs_Err
Setosa	1.00	1.00	1.00	25	0.099
versicolor	0.90	0.69	0.78		
virginica	0.62	0.83	0.71		

(c) Measures of test data accuracy using PCA followed by ANN

	Precision	Recall	F1-score	Acc _Sc	Mean _Abs_Err
Setosa	1.00	1.00	1.00	26	0.066
versicolor	1.00	0.77	0.87		
virginica	0.71	0.83	0.77		

(d) Measures of test data accuracy using Info_PCA

	Precision	Recall	F1-score	Acc _Sc	Mean _Abs_Err
Setosa	1.00	1.00	1.00	30	0.000
versicolor	1.00	1.00	1.00		
virginica	1.00	1.00	1.00		

Table 4.4: Confusion Matrix of Leaf Data Using Simple ANN and IG followed by ANN

(a) Test data confusion matrix with Simple ANN

Pred_class \ actual_class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0
27	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b) Test data confusion matrix with IG followed by ANN

Pred_class \ actual_class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
3	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
19	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0
22	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0
27	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.5: Confusion Matrix of Leaf Data using PCA followed by ANN and Info_PCA model

(a) Test data confusion matrix with PCA followed by ANN

Pred_class \ actual_class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	0	1	0	2	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
25	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
26	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b) Test data confusion matrix with Info_PCA

Pred_class \ actual_class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0
25	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.6: Accuracy evaluation of test samples of leaf data

(a) Measures of test data accuracy using Simple ANN

	Precision	Recall	F1-score	Acc_Sc	Mean_Abs_Err
1	0.33	0.50	0.40		
2	0.50	0.50	0.50		
3	0.20	0.50	0.29		
4	0.00	0.00	0.00		
5	0.00	0.00	0.00		
6	0.75	0.60	0.67		
7	1.00	0.33	0.50		
8	1.00	1.00	1.00		
9	0.50	0.50	0.50		
10	1.00	0.25	0.40		
11	0.00	0.00	0.00		
12	0.80	1.00	0.89		
13	0.60	0.60	0.60		
14	1.00	0.50	0.67		
15	1.00	0.50	0.67	33	0.03431373
16	1.00	0.75	0.86		
17	0.00	0.00	0.00		
18	0.67	0.50	0.57		
19	0.00	0.00	0.00		
20	0.00	0.00	0.00		
21	0.00	0.00	0.00		
22	0.00	0.00	0.00		
23	1.00	0.50	0.67		
24	1.00	0.50	0.67		
25	0.67	0.67	0.67		
26	0.50	0.25	0.33		
27	0.00	0.00	0.00		
28	0.50	0.50	0.50		
29	0.00	0.00	0.00		
30	0.00	0.00	0.00		

(b) Measures of test data accuracy using IG followed by ANN

	Precision	Recall	F1-score	Acc_Sc	Mean_Abs_Err
1	0.00	0.00	0.00		
2	0.00	0.00	0.00		
3	0.50	0.50	0.50		
4	0.50	1.00	0.67		
5	0.50	1.00	0.67		
6	1.00	0.60	0.75		
7	0.80	0.67	0.73		
8	1.00	1.00	1.00		
9	0.67	1.00	0.80		
10	0.75	0.75	0.75		
11	0.00	0.00	0.00		
12	0.50	0.50	0.50		
13	0.71	1.00	0.83		
14	0.00	0.00	0.00		
15	1.00	0.50	0.67	40	0.0027450984
16	0.50	0.50	0.50		
17	0.00	0.00	0.00		
18	0.50	0.25	0.33		
19	0.00	0.00	0.00		
20	0.00	0.00	0.00		
21	0.25	1.00	0.40		
22	0.00	0.00	0.00		
23	1.00	1.00	1.00		
24	1.00	0.50	0.67		
25	0.67	0.67	0.67		
26	0.33	0.50	0.40		
27	1.00	1.00	1.00		
28	0.50	0.50	0.50		
29	0.50	1.00	0.67		
30	0.00	0.00	0.00		

(c) Measures of test data accuracy using PCA followed by ANN

	Precision	Recall	F1-score	Acc_Sc	Mean_Abs_Err
1	0.33	0.50	0.40		
2	0.33	0.50	0.40		
3	0.50	0.50	0.50		
4	0.00	0.00	0.00		
5	0.50	1.00	0.67		
6	0.50	0.20	0.29		
7	1.00	0.33	0.50		
8	1.00	1.00	1.00		
9	0.00	0.00	0.00		
10	0.75	0.75	0.75		
11	0.00	0.00	0.00		
12	0.67	0.50	0.57		
13	1.00	0.40	0.57		
14	0.33	0.50	0.40		
15	1.00	0.50	0.67	32	0.03529412
16	0.50	0.50	0.50		
17	0.00	0.00	0.00		
18	0.75	0.75	0.75		
19	0.00	0.00	0.00		
20	0.00	0.00	0.00		
21	0.14	1.00	0.25		
22	0.00	0.00	0.00		
23	1.00	1.00	1.00		
24	.50	0.50	0.50		
25	0.50	0.33	0.40		
26	0.50	0.25	0.33		
27	.50	1.00	0.67		
28	0.50	0.50	0.50		
29	0.00	0.00	0.00		
30	0.00	0.00	0.00		

(d) Measures of test data accuracy using Info_PCA

	Precision	Recall	F1-score	Acc_Sc	Mean_Abs_Err
1	0.67	1.00	0.80		
2	0.00	0.00	0.00		
3	0.67	1.00	0.80		
4	0.00	0.00	0.00		
5	1.00	1.00	1.00		
6	1.00	0.80	0.89		
7	1.00	0.83	0.91		
8	1.00	1.00	1.00		
9	1.00	1.00	1.00		
10	1.00	0.50	0.67		
11	0.00	0.00	0.00		
12	0.67	0.50	0.57		
13	0.75	0.60	0.67		
14	0.50	0.50	0.50		
15	1.00	1.00	1.00	42	0.025490198
16	0.50	0.50	0.50		
17	0.00	0.00	0.00		
18	0.50	0.50	0.50		
19	0.00	0.00	0.00		
20	0.00	0.00	0.00		
21	0.17	1.00	0.29		
22	0.00	0.00	0.00		
23	1.00	1.00	1.00		
24	0.50	0.50	0.50		
25	0.67	0.67	0.67		
26	0.50	0.50	0.50		
27	0.00	0.00	0.00		
28	0.50	0.50	0.50		
29	0.00	0.00	0.00		
30	0.00	0.00	0.00		

has shown the maximum accuracy and least mean absolute error as compared to the other three models. The confusion matrix of Info_PCA revealed that 42 out of 68 test samples from the leaf dataset and 30 out of 30 test samples from the iris dataset were accurately predicted.

4.7 CHAPTER SUMMARY

This work developed and presented a hybrid model "Info_PCA" for dimensionality reduction. In this model, the information gain generated through Shannon entropy and PCA is blended to use the advantages and functionalities of both feature selection and feature extraction. The developed hybrid model determines the features that were shown to have the greatest separability and the least correlation between the classes. The reduced dataset from the Info_PCA is further used to classify the output classes by using the ANN classifier on datasets of iris and leaf data, selected from the UCI machine learning repository. This work has also made a comparative analysis to identify the effectiveness of the developed Info_PCA by structuring three other methodologies to classify the given datasets. The first method involves just the simple application of an ANN classifier to classify the given dataset. The second method initially performs the feature selection on the dataset through mutual information gain, which is followed by an ANN classifier to classify the data. The third method performs the classification of data through an ANN classifier after the process of feature extraction using PCA. Finally, the fourth method initially applied the "Info_PCA" model on the dataset for dimensionality reduction which is later followed by an ANN classifier for classification. During the entire training and learning process, it has been identified that the proposed Info_PCA has outperformed in terms of classification accuracy (which is 62.76% for leaf data and 100% for Iris data) as compared to the other three models. It has been found from the results of test data of all four models that the classification performed through ANN after the application of the Info_PCA model has shown the maximum accuracy and least mean absolute error as compared to the other three models. The confusion matrix of Info_PCA revealed that 42 out of 68 test samples from the leaf dataset and 30 out of 30 test samples from the iris dataset were accurately predicted.

Thus we can conclude, the presence of large sets of irrelevant features sometimes may degrade the performance rather than improve the model. This problem of irrelevant features affecting the model's accuracy can be solved with the proposed Info_PCA method to reduce the dimensionality of data which otherwise leads to inaccuracy and overfitting in the case of classification and prediction.

CHAPTER V

CROP YIELD PREDICTION USING THE PROPOSED "RaNN" MODEL

CHAPTER 5

CROP YIELD PREDICTION USING THE PROPOSED "RaNN" MODEL

This chapter deals with crop yield prediction, a significant problem related to crops based on different weather and agricultural factors.

5.1 INTRODUCTION

Artificial intelligence and machine learning are being used for giving scientific solutions to handle agriculture sectors like the prediction of change in climatic conditions [65], soil quality checking [10], prediction of crop yield [63,64], plant diseases detection [66,166], management of farmlands [3], etc. Besides, AI and Machine Learning methods are helping for annual planning of crops [11] to increase productivity and yield by understanding various factors responsible for global climate variations such as rainfall, moisture, heat degrees, wind speed and many others [68].

In this chapter, a hybrid model "RaNN" has been proposed and implemented to enhance the accuracy of crop yield prediction by considering certain parameters related to weather conditions and agricultural production. This predictive model makes use of capabilities and functionalities of the two most popular approaches of machine learning-Random forest and ANN, applied for the prediction problems. The work has adopted various other machine learning methods like Ensemble Learner, ANN, SVM Regression, Boosting Regression, Decision Tree, Random Forest and Multiple Linear Regression for the prediction of rice yield. A comparison of the evaluated results of these machine learning models is also performed with the proposed hybrid "RaNN" model to iden-

tify the strength and effectiveness of the "RaNN" model in comparison with the others. Based on analysis and prediction, the "RaNN" model, as opposed to other tested models, delivers better prediction with the highest accuracy, highest correlation, and lowest error on prediction results. The experiments are performed on the agricultural produce data concerning various districts of Punjab between the year 1997 to 2017. The independent variables in the data influencing the response variable "yield" are Precipitation, Temperature, Moisture Content, Area, Production, etc. This proposed model can be adopted as an application for decision-making for yield prediction and hence can be used by both farmers and government bodies.

5.2 METHODOLOGY

As mentioned above, the main objective of this research work is to forecast rice yield using various Machine learning techniques. The different climatic and agricultural factors presented in statistical and numerical data are analyzed for rice yield prediction, which is collected from the two sources: Indian Meteorological Department Pune and Punjab Envis Centre. The gathered data is pre-processed for cleaning to resolve the inconsistency, noises and missing values. The data after pre-processing is passed to the feature selection, which finds the important attributes out of various independent factors impacting the dependent variable called yield. The identified attributes are passed to different machine learning models to predict the yield. The proposed hybrid "RaNN" model for predicting rice yield combining Random Forest and Multilayer Feedforward Neural Network has also been presented in the chapter. This proposed model has demonstrated improved accuracy and correlation between the true and projected results. In comparison to the other tested models, the proposed RaNN model also exhibited the highest prediction accuracy and minimum loss. The entire workflow is diagrammatically represented in Figure 5.1.
The following sections will introduce the data and different feature selection algorithms applied in this work.

5.2.1 Data

The study was performed on the croplands of the Punjab state of India. Data on rice production for 21 years was collected for research from the Pune Meteorological De-

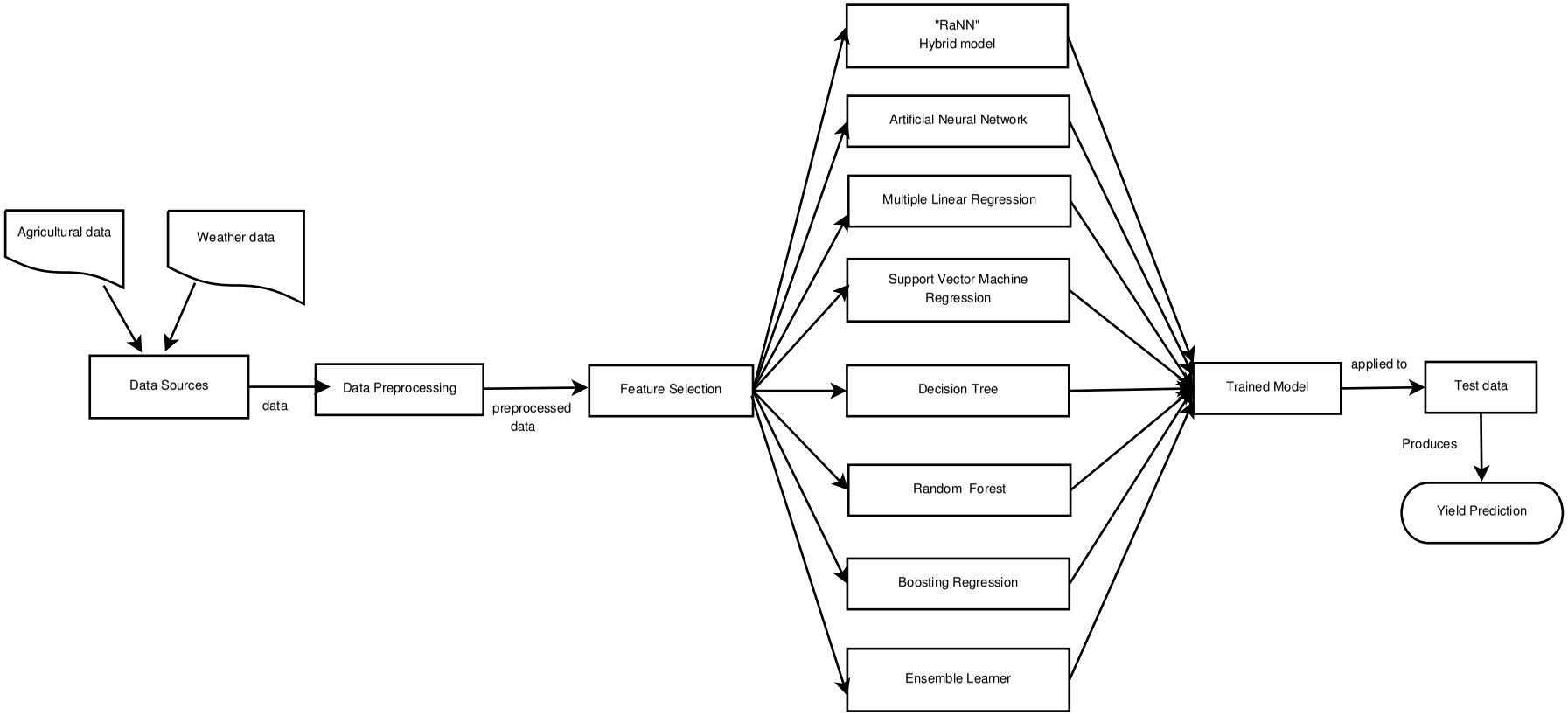


Figure 5.1: Architecture of the methodology designed for the proposed work

partment and the Punjab Department of Agriculture. The GPS coordinates considered for the analysis are 31.1471° N to 75.34124° E. The dataset includes weather and agricultural data for 15 attributes covering the production of rice in 17 different districts of Punjab, including Amritsar, Bhatinda, Fatehgarh, Firozpur, Gurdaspur, Hoshiarpur, Jalandhar, Kapurthala, Ludhiana, Mansa, Moga, Muktsar, Nawanshahr, Patiala, Rupnagar and Sangrur. The data spans the year 1997 to 2017 and Figure 5.2 depicts the state of Punjab as well as its districts. Area, Production and Chemical Fertilizers (Chem Fert) are the aspects of agriculture that were examined. The weather information includes Mean Maximum Temperature in Degrees Celsius (MMAX), Highest Maximum Temperature in Degrees Celsius (HMAX), Mean Minimum Temperature in Degrees Celsius (MMIN), Lowest Minimum Temperature (LMIN), Total Monthly Rainfall in Millimetres (TMRF), RD (No. of Rainy Days), Mean Wind Speed in kmph (MWS), Mean Evaporation in Millimetres (MEVP), Duration of Sunshine in Hours (MSSH), No. of days with precipitation (0.1 - 0.2 mm) P1, Number of days with precipitation (greater equal to 0.3 mm) P2 and Rainfall. Before feature selection, pre-processing of data is performed to manage missing values and detected outliers. The data values are scaled and normalized to remove biasing.

5.2.2 Feature Selection

The data set includes fifteen independent features subjected to a feature selection technique to find the important attributes responsible for accurately predicting the rice yield. To determine the significant features influencing the yield prediction, three distinct feature selection methods are employed, which are briefly introduced as:

Regularized random forest algorithm: The selection of features with the single ensemble is performed by Regularized random forest algorithm [167]. The data features are evaluated at each node during training using regularized random forest while applying a greedy approach. It builds a summarized feature subspace by selecting the essential features using the tree regularization framework of Random forest. It uses regularized information gain.

Correlation-based feature selection: Correlation-based technique [168] is a filter method that does not affect the final classification model. It selects the features based on maximum correlations between features and dependent class to maintain high predictive

Figure 5.2: Punjab state's land cover for predicting the rice yield (Image downloaded from https://diligentias.com/)

power considering the intrinsic properties of data while maintaining minimum feature correlation to avoid redundancy.

Recursive feature elimination algorithm: Recursive algorithm of feature elimination [169] is a method to identify the significant features of data required for predicting the response variable. Features are ranked based on criteria that recursively eliminate features on every iteration. It aims to remove the dependency and collinearity that stays

in the model.

Identified essential features are then passed to regression models of machine learning to generate better prediction accuracy. The features identified as relevant are listed in Table 5.1 and shown as a box plot diagram in Figure 5.3. The green boxplots illustrate the essential features, the yellow boxplots are preliminary, and the blue boxplots exhibit the randomness included in the data set with the shifting of features. The values of a shadow attribute that is not included in the record attributes are shadowmax, shadowmean and shadowmin. Only the top 10 scores with the highest values are taken into consideration from these 13 green box plots, which reveals that the significance of attributes declines as one proceeds from right to left.

Table 5.1: Selected significant features

S.No.	Name of the feature
1	Production
2	Area
3	Chemical Fertilizers
4	MMIN
5	MSSH
6	TMRF
7	Rainfall
8	MEVP
9	P2
10	LMIN

The succeeding section presents the detailed implementation of the developed model and its comparison with the other machine learning techniques.

5.3 EXPERIMENTATION

The chosen set of data features, which are organized into training and test sets and contain 75% (1420 tuples) of the data for training and 25% (356 tuples) for the test, are used in the experiment. Multiple Linear Regression, Decision Tree, Random Forest, SVM Regression, Gradient Boosting Regression, Ensemble Learner and Artificial Neural Network, including the proposed hybrid "RaNN" model, are the machine learning models used for rice yield prediction. These systems are initially worked using 75% of the training data, and their performance is then assessed using the test data. The

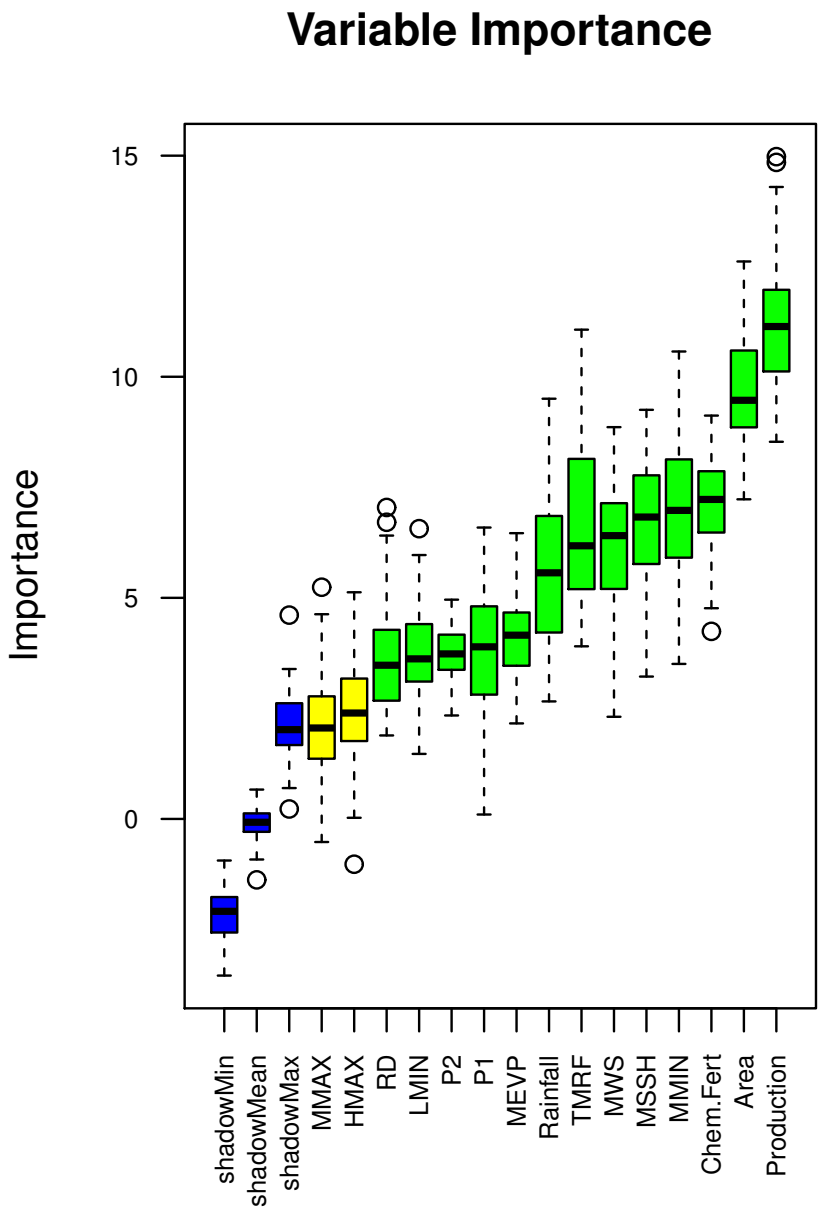


Figure 5.3: Variable Importance Plot

comparative findings are presented in terms of MAE, RMSE and R^2 [77] values generated during the testing of the model. By estimating the error using the indicators given below, the model's performance is assessed during the regression analysis.

Mean Absolute Error: The phrase "Mean Absolute Error" (MAE) refers to the total of the absolute difference in error divided by the number of observations, also known as the mean error, as in equation (5.3.1), and it expresses the difference between the true

and anticipated value of the data in terms of error.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \tag{5.3.1}$$

where y_i is the actual value and $\hat{y}$ is the predicted value, and N denotes the total number of observations. The model is typically assessed using the MAE measure when using regression. For the test data, the mean absolute error of the model is the mean of the predicted absolute error scores for each instance of the test set. In other words, it is the arithmetic mean of all the absolute errors.

Root Mean Squared Error: The measurement of an error's standard deviation is called root mean squared error (RMSE). It is a standard method to estimate the model error for predicting quantitative data. RMSE is a normalized value or gap between an instance's predicted value and target value. According to equation (5.3.2), the instance standard deviation of the relationship between the predicted output and the true data is known as residuals.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \tag{5.3.2}$$

A small RMSE value means better prediction by the model, while a considerable RMSE value loses various essential and relevant features impacting the output. RMSE value is a heuristic for training models and deriving accuracy. RMSE is generally used over standardized data and preferred for supervised learning applications as it requires labels in the form of actual estimation for every instance of data prediction.

R^2(Coefficient of determination): The R^2 value displayed in the range of 0 and 1, gives the percentage similarity between the expected and true value. With rising value, the model's accuracy rises. It shows how the dependent variable might vary depending on the chosen response factors. Equation (5.3.3) gives the formula:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \tag{5.3.3}$$

where $\hat{y}$ is the expected value of y and $\bar{y}$ is the average value of y. The coefficient of determination R^2 establishes a model's effectiveness and precision in predicting the outcome of the dependent variable. The minimum value of R^2 is considered 0, while the maximum value is 1. A model is said to be accurate if its accuracy lies closer to 1.

The part of the variance shared between independent and dependent variables is called the coefficient of determination R^2.
The different machine learning techniques experimented with in the work will be presented in the following sections.

5.3.1 Multiple Linear Regression

Multiple Linear Regression [73, 94] is a statistical method that employs various independent features to predict the response variable. It simulates how the independent and dependent variables are related linearly. The experiment is conducted on normalized data to remove biasing because of massive values. The equation of MLR is given as (5.3.4):

$$P = \beta_0 + \beta_1 y_1 + \beta_2 y_2 ..... \beta_n y_n \tag{5.3.4}$$

Here β_0 represents the bias while $\beta_1, \beta_2 ..., \beta_n$ are determined as the coefficients of independent factors measured by learning the data. The coefficient plot shown in Figure 5.4 shows the influence of a single independent variable on the response variable. The two dependent variables, production and area, demonstrate that the coefficient value going away from 0 has a greater impact on yield. The result of applying the learned MLR model to the test data is shown in Table 5.2, which provides the yield prediction's MAE, RMSE and R^2 values. The R^2 score of the test samples, as given in Table 5.2, indicates that there is a nearly 61% correlation between the actual and anticipated yields.

Table 5.2: Results of Multiple Linear Regression

MAE	RMSE	R^2
0.3049	0.6161	0.6191

Multiple linear regression establishes a mathematical relationship between different random variables. The model produces a straight line that best approximates the maximum data points.

5.3.2 Decision Tree

Decision trees [170–172] are used to choose an outcome by creating a tree of potential decisions for classification and regression issues. The decision tree is a hierarchical

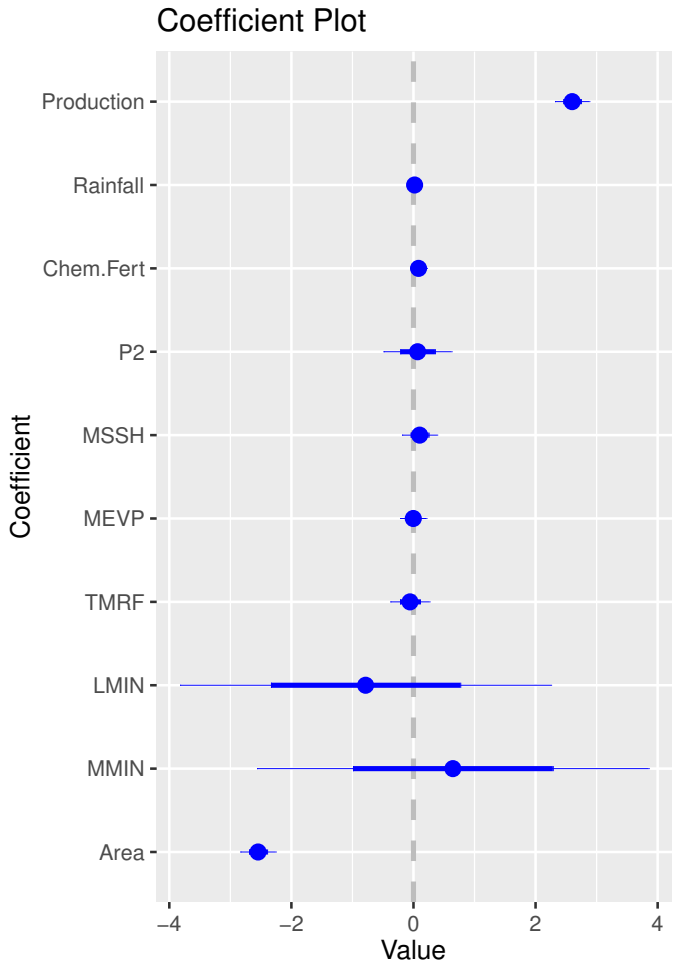


Figure 5.4: Coefficient Plot for MLR

structure in the form of a flowchart, where every internal node shows a condition to be tested on a feature, each branch gives the test result, and each leaf node determines the output labels. A decision tree approach known as a classification and regression tree predicts an outcome by using the mean value of the leaf features containing the training data. The prediction is then computed using the test data. The decision tree performs an attribute value test on each internal node for learning by parting the source set into subsets. This partitioning is recursively performed for learning by splitting the sets. This process of partitioning ends when further splitting does not add value to the prediction results or when all the subset at a node shows the same value of the target variable. The decision tree classifier is best suited for exploratory knowledge discovery as it requires no parameter setting and domain information. It is competent in handling multi-dimensional data. It is called an inductive method to classify objects based on learning and knowledge.

The impurity of a node was tested using the decision tree, and the deviation and Gini metrics were employed to quantify impurity. Table 5.3 shows the yield prediction results produced by the two approaches using the test data in the forms of MAE, RMSE and R^2. Table 5.3 depicts that for test data, there is minimum relevancy between the true and expected yield. The Gini technique's outcome, however, is superior to the deviance approach in terms of correlation and MAE.

Table 5.3: Results of Decision tree

	deviance			gini	
MAE	RMSE	R^2	MAE	RMSE	R^2
0.4823904	0.7623889	0.4251414	0.3728905	0.694024	0.5240611

5.3.3 Random Forest

Random forest [173, 174] forms a group of the tree constructed from different groups of training data chosen from the total training data, including complete feature elements. Using multiple decision trees based on randomly chosen data attributes and sample data training size, random forest regression calculates the predicted value for any specific data point. The mean of all those predicted values is then calculated for a different data point. Random forest arranged three primary hyper-parameters: node size, the number of trees and attributes before training. Thus, the random forest can be adopted for regression and classification problems. A sample dataset known as a bootstrap sample that was created using a replacement from the training data makes up each tree in the group. The dataset is more diversified by introducing some amount of randomness through feature bagging, which will help minimize the correlation among decision trees. Depending on the task that needs to be completed, a random forest uses a different prediction process. For classification, the majority voting technique is applied, choosing the frequent categorical variable as the predicted class, while the average of individual decision trees is calculated for regression. The results are validated using the test samples.

The experiment in this research has been executed using Quantile regression random forest. The Random Forest of Quantile Regression provides a nonparametric and precise method of calculating conditional quantiles required for high dimensional independent features. The model's output, which includes the MAE, RMSE and R^2 values calculated

for the test data, is displayed in Table 5.4. Table 5.4 shows that a regression using Random Forest on test data produced a higher R^2 value between the actual and predicted yield. It gains 89% correlation and 0.12 MAE value finer than the prior experimented models.

Table 5.4: Results of Random Forest

RMSE	MAE	R^2
0.3285428	0.1249495	0.8922962

5.3.4 SVM Regression

SVM [65, 174] determines the line or plane that divides the classes based on the largest margin to represent complex relationships. SVM can be used for predictions as well as classification. An SVM is used for regression when the response variable is numerical or quantitative. The SVM algorithm works on the concept of estimating the best-fit line, a hyperplane covering a larger set of points under a threshold value. The margin linking the hyperplane and boundary line constructs the threshold value. The functionality of SVR is centered only on the part of training data because a large amount of samples gets ignored by the cost function whose prediction results are close to the target value. SVR is easy to implement and robust to outliers and can generalize with good prediction accuracy.

The experiments utilized the SVM model with two different versions- eps-regression and nu-regression [175] on the data set. These two versions differ in the size of control permitted over the errors and the data vectors chosen as support vectors. The eps-regression provides no control, while the nu-regression provides an upper and lower limit for the error and support vectors, respectively. The SVM regression is executed with 10-fold cross-validation on the data with the linear kernel. Table 5.5 displays the result of this strategy for yield prediction using the test data and the two SVM regression models (eps-regression and nu-regression). The predicted output is depicted using the metrics, MAE, RMSE and R^2 values. The MAE and RMSE statistics providing error for both eps regression and nu regression can thus be observed in Table 5.5 to be nearly identical. On the test data, SVM eps-regression had a little better R^2 value for yield prediction than nu-regression. 69% correlation between the true and predicted yield is

obtained using the SVM eps-regression.

Table 5.5: Results of SVM eps-regression and nu-regression with linear kernel

	SVM eps-regression	SVM nu-regression
MAE	0.1816969	0.1823238
RMSE	0.3162596	0.3170481
R^2	0.6913411	0.6896243

5.3.5 Boosting Regression

An additive model called gradient boosting is used to maximize a differentiable loss function [176]. Problems involving classification and regression can both be solved using gradient boosting. The process makes use of gradient descent to reduce the loss. Gradient boosting is a regressor model used to predict age, weight, cost, etc. At each level, a regression tree is fixed to the loss function's negative gradient. Reducing overfitting and limiting loss are the main goals. It calculates the residuals or the distance between the expected and original values. This is followed by training a weak model by relating features to the residual. This residual identified by the weak model is put into the input of the original existing model to identify the correct target. The process is iterated several times to build the correct prediction.

Gradient boosting is mostly preferred for tabular datasets. Any non-linear relationship between response and independent variables can be detected through it. Gradient boosting is good at dealing with high cardinality categorical scores along with missing values and outliers. Gradient boosting regression trees utilizes the concept of ensemble technique originating from decision trees. This approach employs many decision trees as opposed to a single tree, similar to a random forest because a single tree can occasionally result in overfitting. Gradient boosting works by continually adding decision trees, which causes the next decision tree to correct the residual of the previous tree. The results of gradient boosting are sensitive to parameter settings. To minimize the loss, the experiment uses two regularisation functions, L1 (Lasso regression) and L2 (Ridge regression) [177] to fit the linear regression using different loss functions.

- **L1 Regression:** As a penalty term to the loss function, L1 regression adds the absolute value of the magnitude of the coefficients. For the less significant

attributes, it reduces the coefficient to zero and eliminates some variables. The Laplace() family owns it.

- **L2 Regression:** The least square error is decreased using L2 regression (aggregation of the squares of the distance between the estimated and true values). It helps prevent overfitting by summing the squared magnitude of the coefficient as a penalty term to the loss function. It is a member of the Gaussian family ().

Gradient boosting optimizes the loss functions using component-wise linear models. On the training set of data, the two boosting regression models (L1 and L2) are initially trained independently. On the test set of data, the learned models produce prediction accuracy. Table 5.6 displays the results of L1 and L2 regression models on the test data using the metrics MAE, RMSE and R^2. Table 5.6 demonstrates that both L1 regression and L2 regression have poor performance on the test data. However, compared to the L1 regression, the accuracy produced by the L2 regression is 46% which is better than L1.

Table 5.6: Results of Boosting Regression

	L1 regression	L2 regression
MAE	0.4934394	0.5308277
RMSE	0.8593229	0.845262
R^2	0.2754489	0.463634

5.3.6 Ensemble Learner

Ensemble learning [178,179] is the process of combining different models of machine learning to allow for better and more precise prediction. It is a process of combining several low-level models to develop an optimal predictive model. It is a technique to merge a vast set of learners to enhance the model's strength and prediction capabilities. It combines the prediction power of various base learners to create an overall improved structure. Ensemble learners were developed to overcome statistical, computational and representational problems. Ensemble learning is not about getting utterly accurate base models but producing a base model with dissimilar training errors. Ensemble learning can be performed using three methods Bagging, boosting and stacking.

- **Bagging:** Bagging is a way to calculate the prediction accuracy from the average of forecasts from different models.

- **Boosting:** Boosting is an iterative method that improves prediction accuracy by varying the weights of an input based on the most recent categorization.

- **Stacking:** Stacking integrates the output of several models by piling layers on top of one another to generate a better prediction.

The regression models used in EL are those that were developed using a complete training set and integrate several regression models with a meta-regressor. In contrast, the meta-regressor learns from the output of regression models taking them as variables. The stacking method is used for ensemble learning in the experiments because it underlines the gaps in base learning models and significantly reduces bias or variance error. The models that have been used as base-level models for ensembling are multiple linear regression (MLR), support vector machine regression (SVM), and random forest (RF) while the model that has been used as meta-regressor is stochastic gradient boosting (GBM) which is built on top of the base-level models. While the meta regressor model GBM learns using the attributes provided by base-level models, all of the base-level models, including MLR, SVM and RF, are initially trained on the entire training set. Table 5.7 displays the results produced as a result of using the trained ensemble model on the test data to forecast yield as well as the values of MAE, RMSE and R^2. Table 5.7 shows that MLR, RF and SVM each have a unique correlation value of 61%, 85% and 69%, respectively. At the same time, the meta-learner or the ensemble model achieves an overall correlation of nearly 81% that gives the accuracy between the true and the predicted yield. However, compared to the models presented in sections 5.3.1, 5.3.2, 5.3.4 and 5.3.5, ensemble learning has higher model accuracy. However, it is still less than the result of Random Forest when applied separately as given in Section 5.3.3.

Table 5.7: Results of Ensemble learner

	GBM	MLR	RF	SVM
MAE	0.2533316	0.2251682	0.2904947	0.1624851
RMSE	0.4098472	0.3816772	0.4668050	0.3076801
R^2	0.8171344	0.6191635	0.8555569	0.6950808

5.3.7 Artificial Neural Network

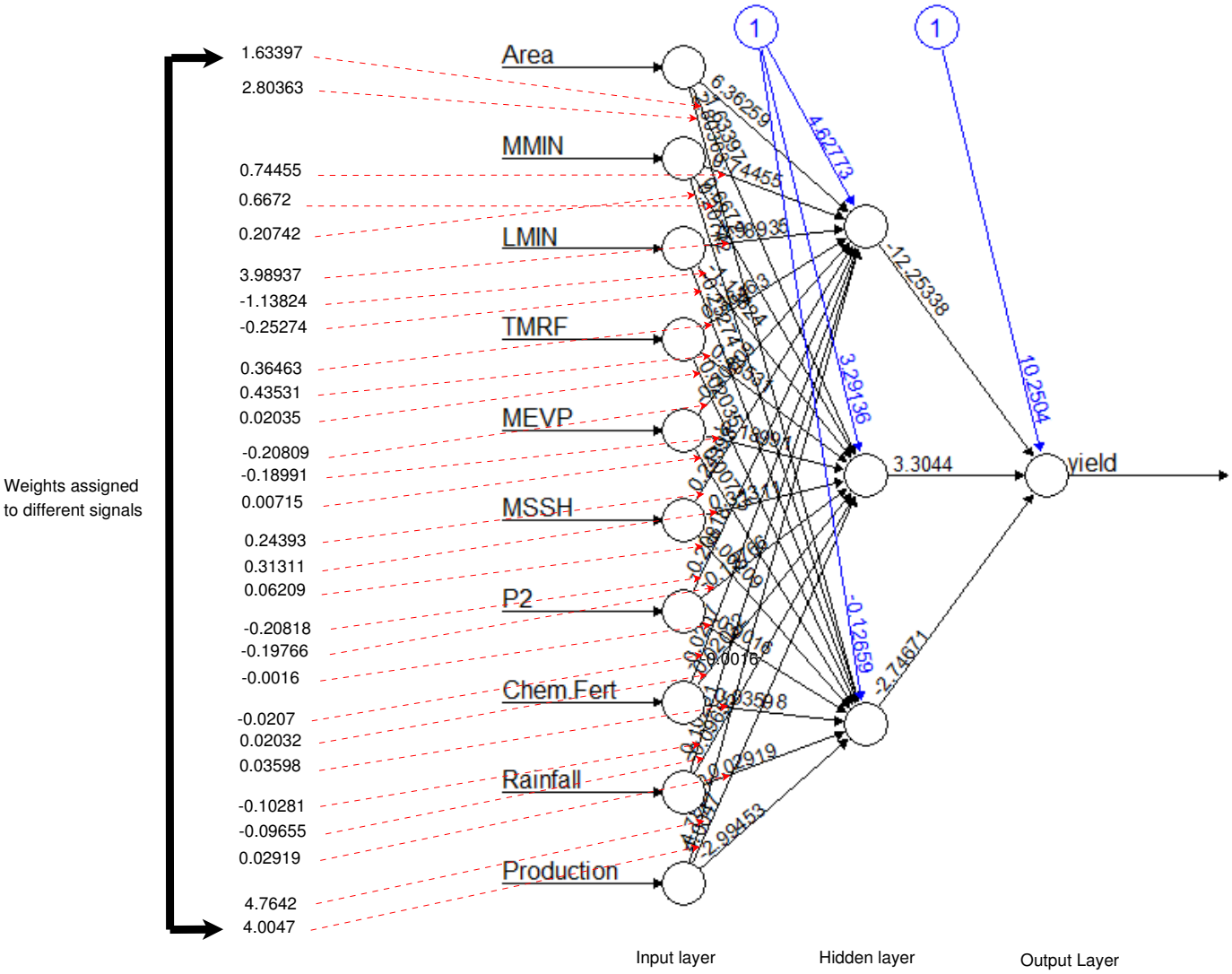


Figure 5.5: Diagrammatic representation of Artificial Neural network

ANN performs regression and classification by developing a relationship between input signals (independent variables) and an output (response variable) based on historical data [180]. It takes input and, using bias introduced by a transfer function, creates the weighted sum of the inputs. If any neuron's output rises above the threshold value, that neuron gets triggered and transfers the signal to the next layer. Otherwise, it remains deactivated. These weights determine the significance of any variable. The larger the weights more significantly the variable contributes to the output. This will produce the summation of the inputs multiplied by their respective weights. It regulates the total amount of weight that is delivered as an input to an activation function to produce the

results specified in equation 5.3.5:

$$Output = \sum_{i=1}^{N} W_i * X_i + b \tag{5.3.5}$$

where N is the number of inputs sent to the network, X_i is the number of inputs, W_i is the weight assigned to the input, and b is the bias.

Table 5.8: Results of Artificial Neural Network

	MAE	RMSE	R^2
3 Hidden layer	0.1808797	0.3663678	0.8107579

This research has applied a multilayer perceptron for crop yield prediction [181, 182] ([181,182]). This work predicts the rice yield using the features listed in Table 5.1 as its input. The input and the output are connected using three fully connected hidden layers. The output yield is obtained by transferring the signal using the sigmoid activation function from the input to the output layer via the hidden layers. Table 5.8 provides an example of the projected result in MAE, RMSE and R^2 values. Table 5.8 shows that although the mean absolute error (MAE) score is small, the R^2 value does not outperform the score of Random Forest, as shown in Section 5.3.4. As can be seen from Figure 5.5, the ANN model thus generates a correlation of about 81% between the actual and expected yields.

5.3.8 RaNN Network (Hybrid RF-ANN model)

Based on the results of all the models of machine learning experimented for rice yield prediction, it is identified that Random Forest and ANN are performing best among all the other models. Therefore, considering these results a hybrid model has been proposed in this research work to enhance the prediction results. The proposed "RaNN" model has combined the capabilities and functionalities of Random forest and ANN, where it selects the feature vectors using the majoring voting technique through Random Forest and passes these feature vectors to ANN as an input. The parameters of ANN are tuned further to correctly predict the rice yield. The entire details of the proposed model are given below.

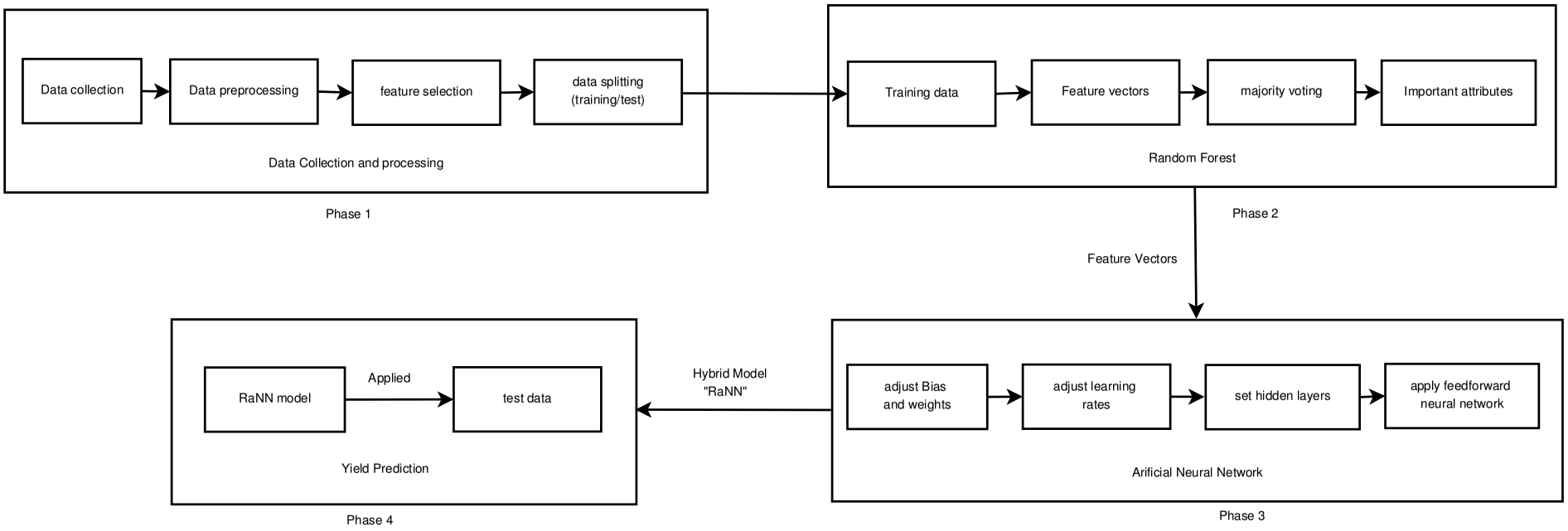


Figure 5.6: Design of the RaNN model

Our hybrid model "RaNN" illustrated in Figure 5.6, combines Random forest and ANN. The architecture includes different phases.

- **Phase 1 of RaNN:** The data is initially collected from various sources, as described in Section 5.2.1, and is then examined for any noises or inconsistencies through pre-processing before being normalized to prevent bias. Following that, the features are chosen using the procedure described in Section 5.2.2.

- **Phase 2 of RaNN:** A training set and a test set are created from the dataset. The training data covering 75% of the dataset undergo the extraction of feature vectors using the decisions of various trees centered on the majority voting technique. Based on the highest vote or suggestions from the other feature extraction models, the majority voting technique is used to extract the best feature vectors. These extracted features are passed as input to the ANN input layer since they are highly significant.

- **Phase 3 of the RaNN model:** The extracted features are fed to the hidden layers of the ANN to forecast the rice yield along with applying appropriate weights and bias. Three learning rates of 0.1, 0.01 and 0.001, are also examined to derive better accuracy.

- **Phase 4 of the RaNN model:** It is responsible for making a prediction which is later employed for the test data.

Algorithm 2 gives the procedure for the whole RaNN model. Initially, the data is checked for noises and missing values, which are later normalized to avoid biasing. The cleaned data is then fed to feature selection algorithms to detect the relevant features. The identified data is split into training and testing where the Random Forest algorithm is applied to training data to identify the feature vectors using the majority voting technique. It identifies the contributing feature which is further passed to the input layer of the ANN model. The parameters like learning rate, epochs, weight and bias of the ANN model are adjusted to give the weighted sum of the output. The model is then checked for MAE and RMSE values. The ANN model follows the backpropagation and feedforward till the error get minimized. The developed RaNN model is implemented on the test data.

Algorithm 2 Algorithm for RaNN model

1. Start.
2. Read the dataset.
3. Check for noises and missing data.
4. Normalized the data.
5. Apply the feature selection algorithm
6. Split the dataset into training and test data.
7. Apply Random Forest on the training dataset.
   (a) Use majority voting technique for decisions on feature vectors.
   (b) Find contributing feature vectors.
8. Use the ANN model.
   (a) Input the feature vectors to the ANN model
   (b) Set the learning rate as 0.1.
   (c) Set the number of hidden layers as 3.
   (d) Set the number of epochs as 500.
   (e) Set the bias and weights of the input layer.
   (f) Calculate the weighted sum of the output layer.
   $Output = \sum_{i=1}^{N} W_i * X_i + b$
   (g) Calculate the MAE and RMSE.
   (h) Move to step 8(e).
   (i) Back propagate followed by feed forward until the error is minimized.
9. Apply the trained hybrid "RaNN" Model to the test data.
10. Generate the result.
11. Identify the MAE, RMSE and R^2 values.
12. End.

Table 5.9 shows the outcome of RaNN prediction in terms of MAE, RMSE and R^2 value. It can be seen from the table that in comparison to all other models tested previously, the value of MAE and RMSE generated through RaNN is substantially lower. Additionally, the R^2 value of the RaNN model is higher than that of the earlier evaluated models. Compared to the other tested models, it obtains a true-to-predicted yield correlation of about 98%.

Table 5.9: Results of RaNN model

	MAE	RMSE	R^2
3 Hidden Layer	0.05490451	0.09056766	0.9880095

5.4 RESULT ANALYSIS

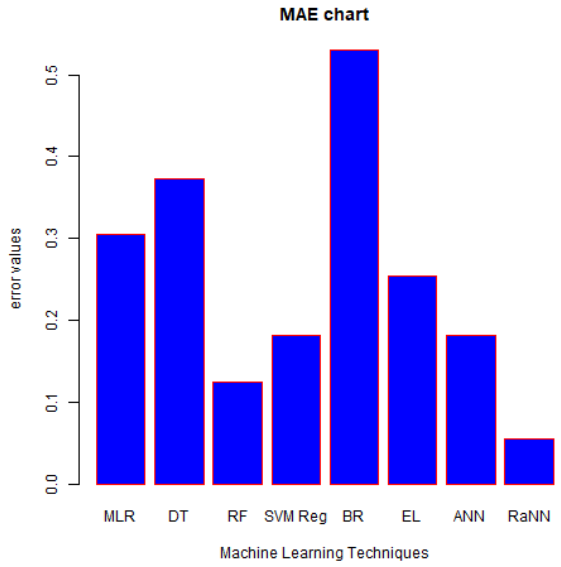
This work was carried out to predict the rice yield using R programming, originally considering 15 different attributes that affect the yield. The work has implemented eight learning models on the dataset to generate a correct rice yield prediction. The initial 15 attributes, including the cultivation and climate features, are transferred to a feature selection method that selects the ten essential features. The work uses three feature selection algorithms to detect valuable features: regularized random forest algorithm, correlation-based feature selection and recursive feature elimination algorithm. This feature selection procedure is required to exclude undesirable features that either have a negative correlation with the response variable or contribute very little to it. Production, area, chemical fertilizers, MMIN, MSSH, TMRF, Rainfall, MEVP, P2 and LMIN are the crucial parameters that are ultimately chosen and which have an impact on the output. The chosen data is then divided into training and test sets, with 75% of the data being utilized for training and 25% being saved for yield prediction tests. As indicated in Tables 5.2 to 5.9, three metrics-MAE, RMSE and R^2-have been used to quantify and examine the prediction results of various machine learning techniques, including "RaNN". The implementation result of the entire work for rice yield prediction is collectively shown in Table 5.10. Table 5.10 is the collective result of all the Tables from 5.2 to 5.9 that exhibit the best results under every individual model. This can be explained by the fact that Table 5.3 compares the performance of deviation and the Gini approach of decision trees, where the Gini model gives a better result. Thus, Table 5.10 presents the optimal decision tree outcome using the Gini model. Similar findings are seen in Table 5.10 for SVM and Boosting Regression. In terms of prediction, the hybrid model "RaNN" performs better than the other tested models. The RaNN model's correlation value is roughly 98%, which is the highest of all the other tested models, presenting the similarity between the actual and anticipated results. In addition, the mean absolute error, which is 0.054, is the lowest of all the models that have been tested. In addition to

the "RaNN" model, Random Forest, Multilayer Perceptron and Ensemble Learner have all demonstrated better accuracy on test data in terms of R^2 value; nonetheless, they still fall short of the "RaNN" model, which provides a 98% correlation between the expected and the true yield.

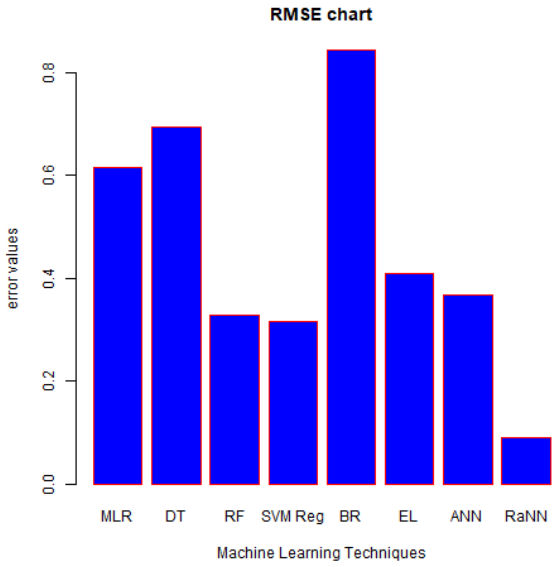
Table 5.10: Collective result of different Machine Learning techniques

Methods	MAE	RMSE	R^2
MLR	0.3049000	0.6161000	0.6191000
Decision Tree	0.3728905	0.6940240	0.5240611
Random Forest	0.3285428	0.1249495	0.8922962
SVM Regression	0.1816969	0.3162596	0.6913411
Boosting Regression	0.5308277	0.8452620	0.4636340
Ensemble Learner	0.2533316	0.4098472	0.8171344
ANN	0.1808797	0.3663678	0.8107579
RaNN model	0.0549045	0.0905676	0.9880095

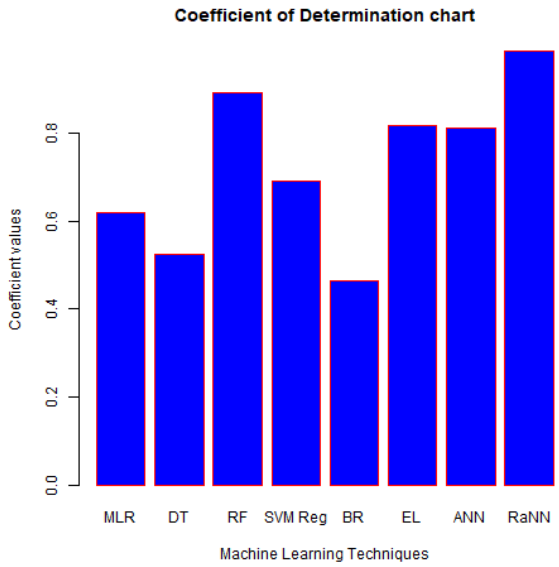
The result of all the eight models in terms of MAE, RMSE and R^2 values are represented as a bar plot given in Figures 5.7 (a), (b) and (c), respectively. Here, the x-axis defines different machine learning models experimented with in the research. In contrast, the y-axis represents the MAE error value in Figure 5.7(a), the RMSE error value in Figure 5.7(b), and the R^2 value in Figure 5.7(c) obtained while predicting the test data. These bar graphs are the representation of Table 5.10, which indicates the value attained by a certain model when it applies the training models to the test data. Table 5.10 and Figure 5.7 (a), (b) and (c) shows that the RaNN model, Random Forest, ANN and Ensemble Learner identifies the top four best-predicted accuracies or R^2 values obtained on the test data. Figures 5.8 (a), (b), (c) and (d) illustrate the predicted and actual test data values for the four best-identified models, where the x-axis indicates the number of samples included and the y-axis indicates the yield obtained using each model. The red dotted line represents the expected yield, whereas the blue line represents the true yield as indicated by the test results. Figure 5.8 (d) shows the high correlation between the true and expected yield where the blue and red dots on the blue and red dotted lines are virtually touching each other. In addition, Figure 5.8(b) displays a strong association in comparison to Figures 5.8 (a) and (c). However, Figures 5.8 (a) and (b) show the test samples' results, producing a perplexing graph. Therefore, to make the Figures clear to



(a) MAE



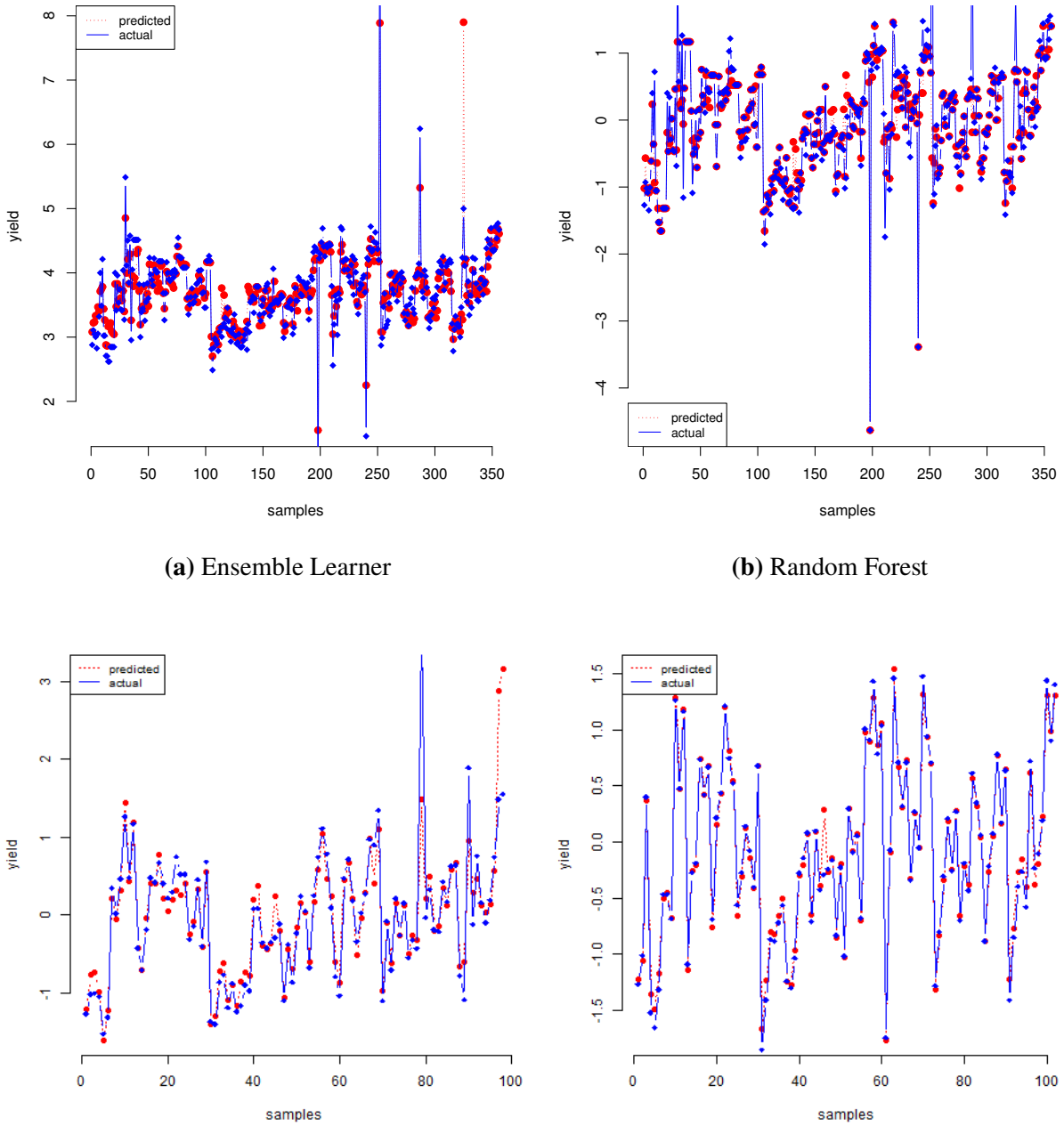
(b) RMSE



(c) R^2

Figure 5.7: Bar plot representation for MAE, RMSE and R^2 values generated through different models

understand, Figures 5.8 (c) and (d) are exhibited only for 100 test instances. Thus, it has been identified from the four graphs that the "RaNN" model depicts better accuracy and correlation not only among the other three models but in fact among all the tested models.



(a) Ensemble Learner

(b) Random Forest

(c) ANN

(d) RaNN

Figure 5.8: Visualization of the test data's true and expected yield using different models (a) Ensemble Learner, (b) Random Forest, (c) ANN and (d) RaNN models

5.4.1 Validation

An additional 100 randomly chosen tuples from the data having ten attributes were used to validate the "RaNN" model. The obtained result is depicted in Table 5.11. The model operates satisfactorily on the validation data as well, as can be seen from the table, which shows a 96% percent correlation and a minimum error between the true and predicted

yield. In Table 5.12, a comparison between our proposed RaNN model and other models

Table 5.11: Result of RaNN model on validation data

	MAE	RMSE	R^2
3 Hidden Layer	0.08490451	0.17056766	0.96548305

discussed in the literature for predicting various crop yields has been carried out. The table shows that, in comparison to the previous models, our proposed model RaNN performs considerably well with the highest correlation and minimal RMSE.

Table 5.12: Comparison of RaNN and other machine learning methods for crop yield prediction

Reference	Technique	Correlation	RMSE
[87]	rough set on fuzzy approximation using neural network	0.93	0.55
[183]	extreme learning machine (ELM)	0.89	0.8
[184]	support vector machine	0.94	0.255
[185]	Random Forest	0.92	0.42
[79]	Naive Bayes, BayesNet	0.84, 0.97	0.29, 0.144
[186]	Support Vector Machine Regression	0.968	0.548
[187]	BPNN	0.88	0.253
[80]	SVM	0.97	0.34
[12]	RF	0.85	0.62
[73]	MLR	0.95	0.75
[75]	ANN	0.86	0.188
proposed RaNN model	hybrid RF-ANN	0.98	0.09

5.5 CHAPTER SUMMARY

In this chapter, we've seen that the correct crop yield prediction is a challenging issue in agriculture that requires advanced agriculture technological capabilities. Thus, considering such issues, a hybrid model "RaNN" has been developed in this work to enhance the accuracy of crop yield prediction, considering the weather and agriculture factors affecting the yield and production of crops. This model is a hybrid model which merges the functionalities of Random forest and ANN machine learning techniques, generally employed for prediction problems. Moreover, the work has also examined seven other machine learning techniques like Ensemble Learner, ANN, SVM Regression, Boosting Regression, Decision Tree, Random Forest and Multiple Linear Regression on the same

dataset for predicting the rice yield. Finally, a comparison of the evaluated accuracy of these machine learning models is performed with the proposed hybrid "RaNN" model to identify the strength and effectiveness of the proposed model. Based on analysis and prediction findings of the RaNN model, as opposed to other tested models, delivers better predictions by showing the highest accuracy with a 98% correlation between the true and the expected output and the lowest error on prediction results. The models are evaluated on the rice yield dataset of different districts of Punjab from the year 1997 to 2017 covering 15 different agricultural and climatic parameters.

CHAPTER VI

CROPLAND AREA ESTIMATION

CHAPTER 6

CROPLAND AREA ESTIMATION

This chapter demonstrates agricultural cropland mapping using different threshold-based image segmentation algorithms and later with machine learning algorithms for cropland area estimation.

6.1 INTRODUCTION

Besides urbanization and industrial progress, the rise in population reduces the availability of productive agricultural lands [98]. This demands the effective utilization and management of these leftover lands for the progress of a country [188]. These abandoned lands could be used for other agricultural purposes and production [13, 99]. A data-driven agricultural process has advanced primarily to the expansion of data and high-resolution imagery, such as that provided by Google satellite, which might be used for sophisticated cropland mapping using remote sensing, machine learning and computer vision algorithms. These data could be utilized for efficient cropland mapping [103–105] to emphasize the production [106–108] needed to maintain the global food security and quality of life. Cropland mapping also has applications in crop type mapping [109–111] that could be employed for crop yield prediction, estimation of cropland area, keeping track of farming practices [114], quality checking of soil, identification and estimation of vegetation damage [108, 112, 113], damage detection in land cover [104, 129, 130], etc.

The image segmentation techniques can be used to detect such abandoned agricultural lands [100]. To separate the relevant area from the background, image segmentation divides a picture into several non-overlapping sections based on characteristics like ho-

mogeneity and similarity [102]. It will learn the top and low-level attributes of different layers of farmlands to grasp the edge information of subject boundaries and long-distance spatial dependencies to detect the croplands [124, 125]. Agriculture [14, 121, 122], object identification [120], disaster management [119], semantic text segmentation [123], and other fields have all used the image segmentation techniques. The classification method assigns a single label to the entire image at once, but the semantic segmentation process assigns a label to each image pixel. Different data from the same class are treated as a single unit by the semantic segmentation procedure [14]. As the satellite images facilitate identifying the concise view of different land cover regions, therefore, the research has chosen satellite data consisting of images of general landscape covering croplands.

The work in this chapter studied and presented agricultural cropland mapping using two different approaches-thresholding based algorithms and machine learning algorithms. These techniques perform the semantic segmentation of croplands to recognize the region of interest (area under cultivation). Initially, the semantic segmentation process is carried out through four thresholding-based techniques such as histogram-based segmentation [138, 189], Multi-Otsu-based segmentation [190, 191], K-means-based segmentation [14] and Random Walker segmentation [192], to find the regions of cropland against uncultivated or urbanized areas. It has been identified, that out of all the four methods of thresholding-based algorithms, the k-means method results in better segmentation. But due to the static approach followed by the thresholding techniques, the segmented results are not found to be correctly segmented. As of this, machine learning models are being utilized for the segmentation of croplands. This process is carried out initially by extracting the relevant set of features through the feature extraction process, which is followed by the application of three machine learning techniques-Random Forest, SVM and ANN. The derived results are compared to identify the technique producing the best segmentation, where Random forest produces the best results. Despite the textural complexity and pixel proximity beneath different locations, the method successfully divided the cultivation area through multiple iterations. The segmented areas under the croplands are also estimated to identify the area covered under agricultural cultivation. This approach can be applied commercially to segment cultivated areas related to a particular crop and hence, identify the total yield and production.

The two semantic segmentation techniques which will be discussed in Section 6.3 and Section 6.4 are evaluated through certain metrics. These metrics are briefly introduced in the following section.

6.2 MODEL'S EVALUATION METRICS

Accuracy, Mean IoU and Kappa Coefficient are the metrics for measuring the model's performance concerning training and test data.

Accuracy: Accuracy is one of the metrics used to assess the classification results and hence the performance of the model. The accuracy measure is used to determine, up to what level the output produced by a model is identified as accurate. The model's effectiveness is assessed across all equally important classes. This accuracy score for multiclass categorization ranges from 0 to 1, where 1 denotes the highest level of accuracy. The match score between the actual and predicted classes determines this value. The formula (6.2.1) provides accuracy:

$$Accuracy = \frac{\text{number of correctly predicted events}}{\text{sum of all predictions}} \tag{6.2.1}$$

The accuracy metric identifies the frequency of correctness of an algorithm in classifying the data point correctly. The total of true positives and negatives divided by the total of true positives, false positives, true positives, and true negatives identify the accuracy. A data point is called a true positive or true negative if they are correctly classified as true or false, respectively. Data points that are incorrectly classified are referred to as false positives or false negatives. This is shown in the formula given below (6.2.2):

$$Accuracy = \frac{true_{positive} + true_{negative}}{true_{positive} + false_{positive} + true_{negative} + false_{negative}} \tag{6.2.2}$$

Mean Intersection-over-Union: Image segmentation is also verified using the metric mean intersection-over-union. It first calculates the IoU value for each semantic class before estimating the average across all the classes. The confusion matrix with a sample weight that is used to generate the MeanIoU value is used to represent the predicted and

real classes. The formula (6.2.3) provides the answer:

$$IoU = \frac{true\_positive}{true\_positive + false\_positive + false\_negative} \tag{6.2.3}$$

where false_positive refers to erroneous pixels that have been mistakenly detected as correct pixels while false_negative refers to correct pixels that have been mistakenly detected as incorrect pixels. Apart from this true_positive refers to the correct pixels and true_negative are the incorrect pixels. IoU gives a number that computes the overlap between the ground truth and predicted area in the form of bounding boxes covering the object identified.

Cohen's kappa Coefficient: A quantitative metric known as Cohen's kappa coefficient (k) is used to measure the level of agreement among the two raters, which will group I items into D distinct categories. It assesses the consistency of ratings made by two different raters, who randomly agree on the same item. It's stated as (6.2.4):

$$k = \frac{P_0 - P_e}{1 - P_e} \tag{6.2.4}$$

where P_e is the fictitious likelihood of agreement using observed data and P_0 is the observed relative agreements between raters. If all raters agree, then the value of k will be 1, and if there is no consensus, then it can only happen by chance, leaving k=0. Interrater reliability is analyzed using Cohen's kappa coefficient. The metric is used to identify the extent to which data gathered for the task appropriately matches the selected variable. Interrater reliability is evaluated by the range up to which raters give the same score to the identical variable.

The two different approaches-thresholding based techniques and machine learning techniques, applied for the semantic segmentation of croplands are introduced and presented in the following sections.

6.3 EFFECTIVENESS OF THRESHOLDING BASED IMAGE SEGMENTATION ALGORITHMS FOR AGRICULTURE CROPLAND MAPPING

This section will present the different thresholding techniques used for producing the semantic segmentation of croplands, which will help with agricultural cropland mapping.

These techniques are further compared to identify the strength and effectiveness of each algorithm. It has been identified that the k-means thresholding technique produces the best result of all.

6.3.1 Methodology

Thresholding-based image segmentation is an easy and functional way of detecting regions of interest holding different intensities and colors of an image by fixing an optimal threshold value. The thresholding method helps in image analysis by changing the pixels of an image with white or black color based upon the particular pixel intensity [193]. The image pixels get changed from color or greyscale to a binary image with the low-intensity pixel replaced with black, whereas the high-intensity pixels get changed with the white color. The thresholding method helps identify the region of interest by rejecting the unwanted areas. The four thresholding-based semantic segmentation algorithms applied for the task include the Histogram approach, Multi-Otsu thresholding, Random Walker and k-means thresholding-based segmentation, which are computer vision image segmentation approaches. These experimented algorithms are compared based on the segmentation accuracy of croplands, which is discussed at the end of this section.
The four thresholding-based algorithms implemented in this work, are briefly discussed below.

Histogram based segmentation

The histogram is a graph representing several poles. It is a graphical representation that sums up a variable's distribution when the variable's value is numeric. The value range of a variable is partitioned into separate successive subranges called bins or buckets. These buckets carry the subset of values of any variable distribution. The distance between two buckets is called bucket width. Histograms are used for exhibiting the patterns lying inside the data along with the dispensation of a variable. Histograms have a unique feature called the number of modes or peaks in the distribution, which are visibly distinguishable. The moment when the distribution fall and rises creates a peak [39, 138]. Histograms can be classified as unimodal, bimodal and multimodal. A unimodal shows a single peak in the variable distribution; a bimodal illustrates two peaks, while a multimodal shows three or more peaks [189]. An image histogram

depicts the relative frequency of different color (greyscale) instances in an image. The histogram is shown as a discrete function for a digital image with a number N of grey levels, given as (6.3.1):

$$h(c_i) = \frac{n_i}{N}, i = 0, 1, ..., N-1 \tag{6.3.1}$$

provide the number of pixels n_i with color c_i as a ratio of the total number of pixels N. This work has utilized a bi-modal histogram where the image histogram is portioned through a single global threshold value V. By detecting the threshold at the bottom of the valley, the histogram employs peaks and valleys to divide the relevant areas from the background. The factors that may impact the results are image noises, objects and background-size, uniformity of the illumination and reflectance.

Multi-Otsu based segmentation

The Multi-Otsu thresholding algorithm splits the pixels into separate classes depending on the intensity of the greyscale of an image. It evaluates the various values of the threshold based on the required number of classes [190]. The Multi-Otsu algorithm has three default classes, with two threshold values given by two red lines in the histogram. Otsu thresholding is considered a ubiquitous thresholding algorithm in image segmentation. It uses binarization algorithms that iterate over all threshold values that estimate the variance of pixel level on either side of the threshold. Multi-otsu algorithm uses the pixel's spatial information, mean and median of the neighborhood to perform segmentation with better results and good noise immunity than 1D and 2D Otsu. However, multi-otsu suffers from more time complexity compared to others. When a picture should include the two foreground and background classes of pixels that create a bi-modal histogram, Otsu's method [39] can be utilized to compress the greyscale image in the binary image automatically using cluster-based image thresholding. The technique then assesses the best threshold to split the classes so that the least intra-class variance referred to as a weighted sum of the variance of the two classes given in (6.3.2) is maintained:

$$\gamma_x^2(t) = x_0(t)\gamma_0^t(t) + x_1(t)\gamma_1^t(t) \tag{6.3.2}$$

where γ_0^2 and γ_1^2 are variances of the two classes, and x_0 and x_1 are the probabilities of the two classes parted with a threshold t. When used for multi-level thresholding, this process is called Multi Otsu's method [194]. This is identified as a fast algorithm for multi-level thresholding.

Random walker segmentation

Sometimes histograms find difficulty in properly segmenting the image; then, a Random walker segmentation algorithm can be used for such cases. Random Walker is a semi-automatic image segmentation method [192] that gives the benefit of dividing an image into an arbitrary number of objects as per the need rather than splitting the image into background and region of interest. The algorithm is centered on the concept of providing a few pixels with user-defined labels called seeds which are used to label the unseeded pixels. This is performed by solving the probability that a random walker initiating from each unlabelled pixel will first reach the seeds. The random walker thus identifies the final segmentation by choosing the most probable seed position for each unseeded pixel. A random walker can give a quality segmentation if the appropriate seeds are positioned manually. The segmentation utilizes an anisotropic diffusion equation to calculate the local diffusivity coefficient with markers to select similar and unknown pixel values. Random Walker considers the image as a graph where a node is represented by a vertex v_i which is a group of pixels, and e_i edge shows the connection between the two vertexes v_i and v_j and in turn between the pixels. The vertices are grouped as labeled vertices marked as connected to objects as seeds, and the other group is unlabelled vertices. Based on the intensity of the greyscale or color scale of the two pixels, weights are applied to the edges of the pixels w(v_i,v_j). Let g_i be the intensity level of pixel v_i be it grey or color scale and likewise, g_j be the intensity of v_j, then the weight on edges w(v_i,v_j) can be given as (6.3.3):

$$w(v_i, v_j) = exp(-\beta(g_i - g_j)^2) \tag{6.3.3}$$

These weights, edges, and vertices can be utilized to produce the graph. This structure makes them flexible for transmitting information via the pixels throughout the image.

K-means thresholding for segmentation

This segmentation process does not depend on histogram formation, which reduces the possibility of noise generation. The K-means works on splitting the data into k classes that should have minimum inter-similarity among classes and maximum intra-similarity across a cluster. The algorithm allocates the random seeds and assigns the various data points to these seeds or centroids till the fixing of centroid [195–197]. K-means use the concept of variance and assign the data points to the appropriate cluster for better segmentation. Let the K-means split the N data points into k classes or clusters such as $C_1, C_2, C_3, ...C_k$ by managing the condition of having a minimum sum of the square distance between the cluster which is represented as (6.3.4):

$$D = \sum_{d=1}^{k} \sum_{n \in C_d} (y_n - T_d)^2 \tag{6.3.4}$$

where y_n is the value representing n^{th} data point, d is the index value of classes between $1 < d < k$, T_d is supposed as the centroid of the data points in C_d.

(a) picture 1 **(b)** picture 2

Figure 6.1: (a) and (b) are the sample of two cropland pictures used in the work

6.3.2 Experimentation and Results

The experiment has been performed on 25 satellite images adopted from www.google.com. Two sample images from the data used in the work are depicted in Figure 6.1(a) and (b). Initially, the image is cleaned for noises and converted from color to grey level while the image pixel values are converted from integer to floating values. The different adopted algorithms are performed in the way listed below.

- Histogram uses the greyscale images to divide them into various grey-level areas. The image's sigma value is estimated for each pixel. The histogram is used to

select the ranges for various segments.

- The Multi-Otsu method is used for the automatic segmentation of images. The numbers of classes are used to give minus one level of thresholding to partition the area of interest.

- Random Walker maximizes the histogram pattern by extending the range of the histogram. It provides two labels spotted as dark and white; anything below a point marks the darker face while anything above it marks the brighter face to segment the image. A random walker covers every pixel in the image and gives each unassigned pixel either a label 1 or 2 using the initial seeds.

- In K-means, the pixels are flattened for clustering purposes where the process is carried out till they reach an eps value and max iteration value. The algorithm will revert the compactness (total of squared distances linking the points and the centroid), labels and cluster centers as output.

Based on the four thresholding techniques used, the semantic segmentation outcome for the sample image of Figure 6.1 (a) (Picture 1) is shown in Figure 6.2 and Figure 6.1 (b) (Picture 2) is in Figure 6.3. The results are illustrated in terms of the applied segmentation technique, their corresponding greyscale image, the histogram built illustrating the expansion of pixels, and the segmented image generated using the different techniques. The segmented result of the croplands derived from the different techniques as given in label (d) of Figure 6.2 and Figure 6.3 is used to check the effectiveness of the technique through the evaluated accuracy. The bar chart given in Figure 6.4 illustrates the performance accuracy of different techniques on the given data. It can be identified from the graph that the k-means thresholding method has shown the best performance among others for both pictures 1 and 2 with 90% and 93% accuracy, respectively. This is followed by the random walker approach with 85% and 83% accuracy. Similarly, the procedure is repeated for each of the 25 pictures, and the outputs are displayed in Table 6.1, where the k-means thresholding method performs best in overall accuracy, Mean IoU and kappa score. The k-means algorithm has shown better performance because it does not require histograms, which makes it appropriate for 2D and 3D multi-level thresholding. But due to the static approach followed by the thresholding techniques the segmented results are not found to be correctly segmented. Thus, in the other section of

this work, machine learning techniques are applied to perform semantic segmentation through an iterative approach.

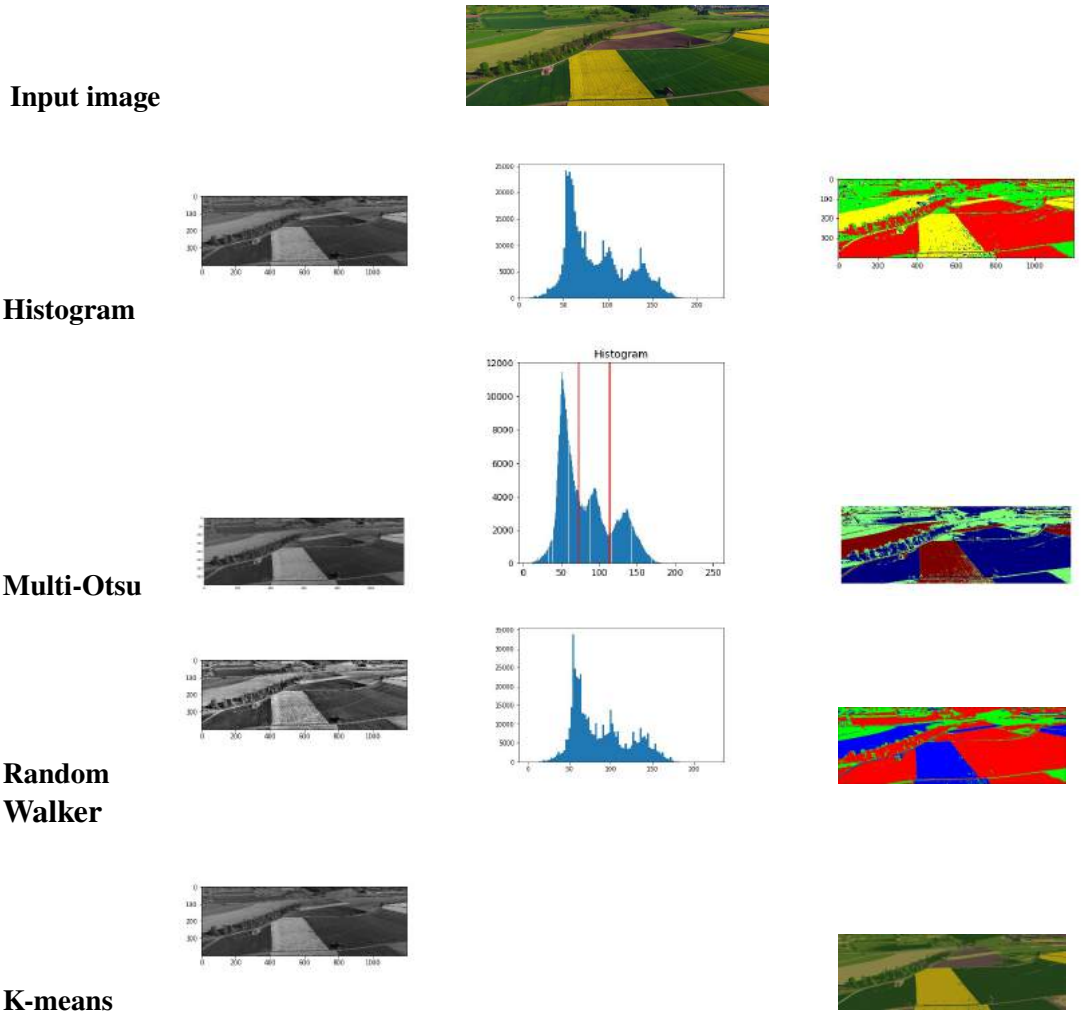
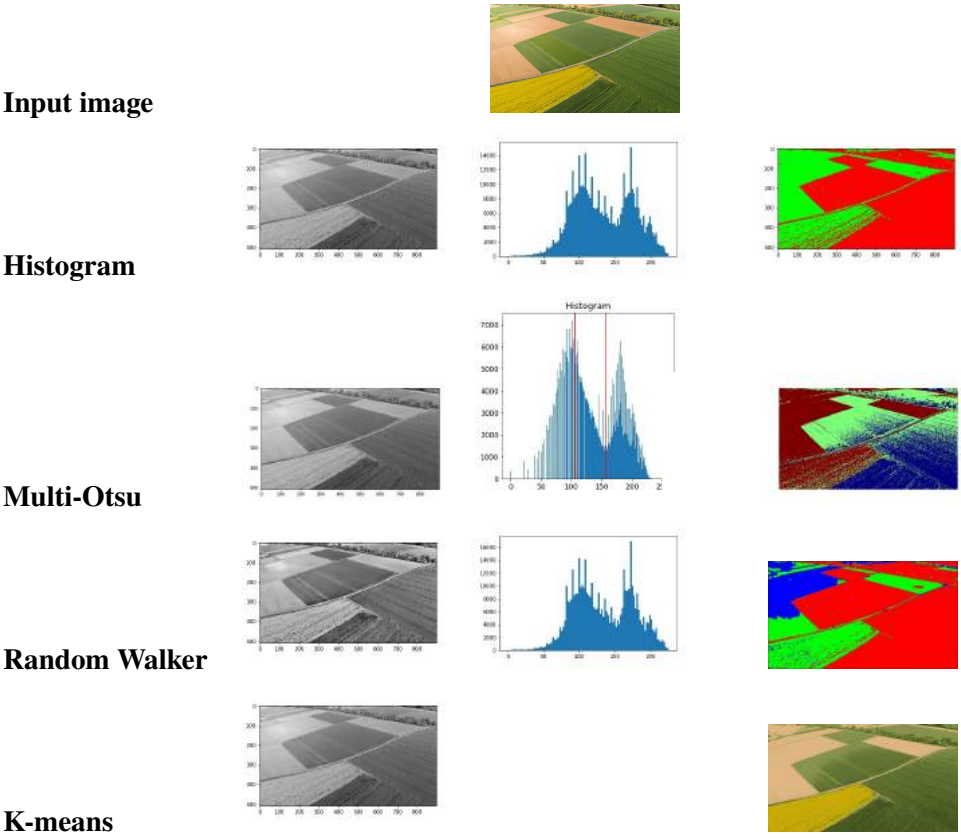


Figure 6.2: Semantic image segmentation of input image (picture 1). (a) Name of different techniques applied on the image for segmentation. (b) Corresponding grey level images for input image (c) Histogram of pixel distribution under different techniques (d) Final segmented image using different techniques



(a) Segmentation Technique	(b) Grey-level Image	(c) Histogram	(d) Segmented Image

Figure 6.3: Semantic image segmentation of input image (Picture 2) (a) Name of different techniques applied on the image for segmentation. (b) Corresponding grey level images for input image (c) Histogram of pixel distribution under different techniques (d) Final segmented image using different techniques

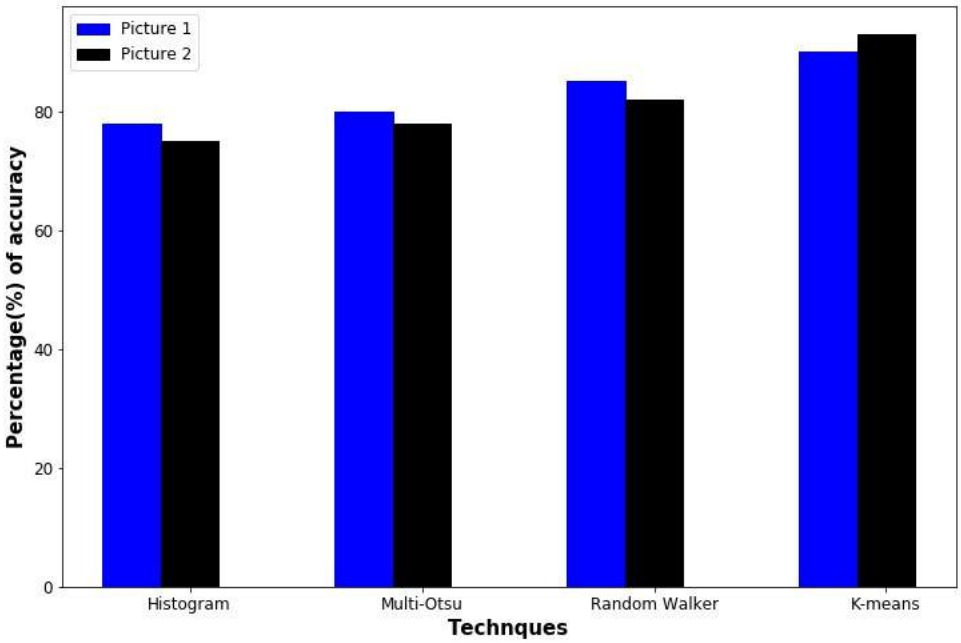


Figure 6.4: Performance measurement of different techniques

Table 6.1: Semantic segmentation results of the applied algorithms

Pictures	Histogram			Multi-Otsu			Random Walker			k-means		
	Acc	Mean IoU	kappa score	Acc	Mean IoU	kappa score	Acc	Mean IoU	kappa score	Acc	Mean IoU	kappa score
Picture 1	0.6934	0.7254	0.4432	0.7234	0.7436	0.4378	0.7998	0.8166	0.5254	0.9034	0.9318	0.6632
Picture 2	0.7012	0.7315	0.4557	0.6954	0.7183	0.4312	0.7642	0.7832	0.5132	0.8978	0.9239	0.6954
Picture 3	0.7342	0.7583	0.4847	0.7732	0.7938	0.4783	0.8341	0.8468	0.6172	0.8864	0.9133	0.7183
Picture 4	0.7186	0.7335	0.4658	0.7546	0.7722	0.4831	0.8688	0.8724	0.6346	0.9438	0.9742	0.7414
Picture 5	0.7684	0.7789	0.4925	0.7812	0.7892	0.4933	0.7914	0.8125	0.5344	0.8934	0.9364	0.6855
Picture 6	0.7114	0.7385	0.4544	0.7377	0.7532	0.4783	0.8386	0.8563	0.5966	0.8714	0.9217	0.6752
Picture 7	0.7534	0.7872	0.5188	0.7612	0.7877	0.4978	0.8564	0.8679	0.6142	0.9086	0.9457	0.7343
Picture 8	0.7642	0.7765	0.4977	0.8012	0.8167	0.5342	0.8412	0.8528	0.5738	0.9134	0.9483	0.7258
Picture 9	0.7118	0.7383	0.4832	0.7288	0.7572	0.4637	0.8343	0.8492	0.5842	0.9388	0.9617	0.7366
Picture 10	0.7345	0.7412	0.4953	0.7714	0.7897	0.4939	0.8177	0.8273	0.5167	0.8765	0.9186	0.6897
Picture 11	0.6683	0.6881	0.4398	0.7379	0.7572	0.4652	0.7934	0.8178	0.5372	0.8834	0.9211	0.6763
Picture 12	0.7823	0.7748	0.4711	0.8086	0.8129	0.5121	0.8586	0.8774	0.5798	0.9010	0.9315	0.6851
Picture 13	0.6884	0.7077	0.4783	0.7642	0.7833	0.4865	0.8239	0.8562	0.5532	0.9034	0.9217	0.6766
Picture 14	0.6966	0.7142	0.4487	0.7392	0.7536	0.4673	0.8134	0.8416	0.5612	0.8768	0.9146	0.6532
Picture 15	0.7623	0.7848	0.4511	0.7886	0.8029	0.4721	0.8386	0.8474	0.5398	0.9310	0.9615	0.7251
Picture 16	0.7319	0.7562	0.4618	0.7734	0.7817	0.4617	0.8513	0.8617	0.5477	0.9112	0.9432	0.7018
Picture 17	0.6862	0.7081	0.4317	0.7312	0.7533	0.4531	0.8034	0.8233	0.5692	0.8964	0.9318	0.6977
Picture 18	0.6944	0.7286	0.4572	0.7832	0.8011	0.4843	0.8432	0.8723	0.5518	0.9412	0.9788	0.7384
Picture 19	0.6834	0.7037	0.4372	0.7134	0.7432	0.4397	0.8113	0.8439	0.5832	0.8816	0.9233	0.7116
Picture 20	0.7177	0.7392	0.4532	0.7586	0.7634	0.4846	0.8839	0.8912	0.5642	0.9511	0.9813	0.7532
Picture 21	0.6534	0.6784	0.4324	0.7612	0.7894	0.4952	0.8239	0.8314	0.5837	0.9223	0.9566	0.7136
Picture 22	0.7138	0.7355	0.4567	0.7314	0.7456	0.4362	0.8103	0.8346	0.5462	0.8813	0.9133	0.6837
Picture 23	0.7512	0.7792	0.4639	0.7534	0.7762	0.4566	0.7913	0.8167	0.5313	0.8914	0.9263	0.7034
Picture 24	0.6534	0.6714	0.4263	0.7386	0.7587	0.4519	0.8478	0.8538	0.5742	0.9314	0.9656	0.7335
Picture 25	0.7436	0.7673	0.4793	0.7794	0.7876	0.4837	0.8417	0.8566	0.5992	0.9158	0.9437	0.7277

6.4 SEMANTIC SEGMENTATION OF CROPLAND THROUGH MACHINE LEARNING AND ESTIMATION OF CROP AREA

This section implemented and presents the semantic segmentation of land cover areas through feature extraction and machine learning techniques. The segmented regions are further processed to identify the area covered under the different regions of land cover.

6.4.1 Data Gathering and Preparation

The region of study is the Indian state of Madhya Pradesh, as shown in Figure 6.5. Twenty Google satellite pictures from the QGIS 3.18 Zurich application make up the collection. The dataset is limited to just twenty photos only because .tiff images require a lot of computing effort and speed. Figure 6.6 displays a few of the land cover photos that were selected for the objective. These pictures are annotated using "http://www.apeer.com" to produce four labels: croplands, rivers, buildings and uncultivated regions represented as the background. This creates a mask or ground truth. The dataset is split into training and test portions, where training receives 80% and testing receives 20% of the data. This work is performed using an Nvidia GPU processor, Python 3.7 with TensorFlow 2.0 and Keras. Sci-kit, Open CV and NumPy libraries are the additional libraries used here for image processing.

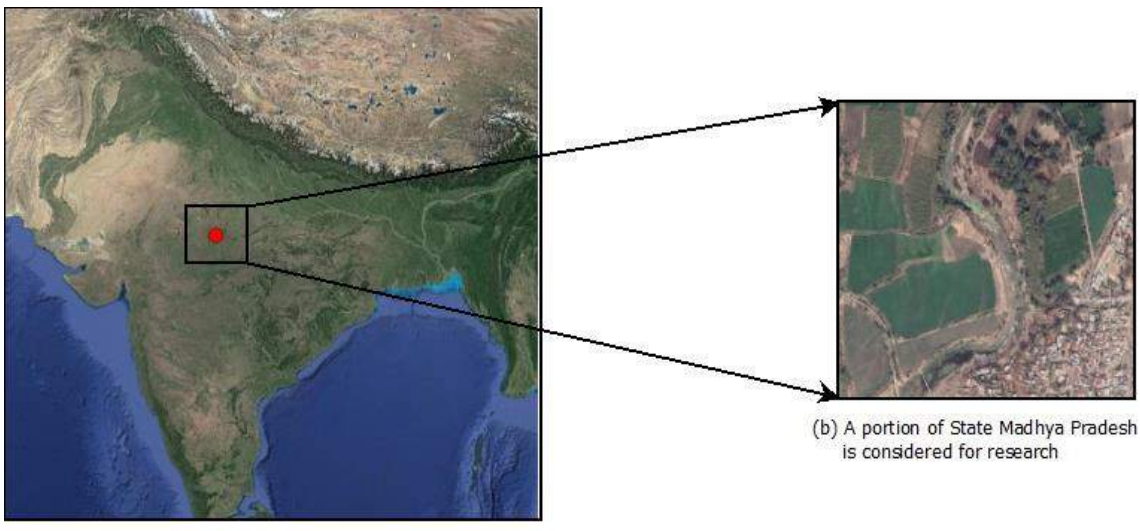
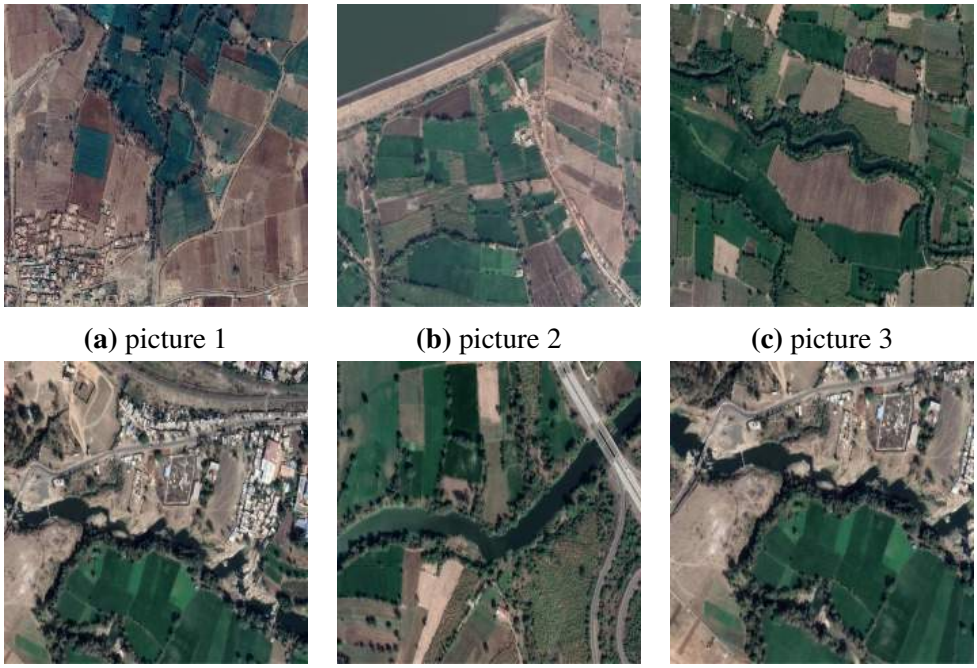


Figure 6.5: Land cover area under study (a) A map of India (b) India's Madhya Pradesh region opted for the study



(a) picture 1 **(b)** picture 2 **(c)** picture 3

(d) picture 4 **(e)** picture 5 **(f)** picture 6

Figure 6.6: Images of the opted land covers

6.4.2 Materials and Methods

With the aim of mapping cropland, a method was developed to accomplish the semantic segmentation of images using machine learning techniques followed by area estimation of distinct regions. Initially, the annotated masks are created for all the pictures which are divided into four labels: croplands, rivers, buildings and uncultivated regions. Based on the pixel value, the pixels are divided into light and dark pixels, with anything above a specific value being seen as a bright and important pixel while lower pixel values, such as 0 to 40, are used as background pixels. These images are then forwarded to the feature extraction process, which identifies the important features needed for segmentation. Gabor, Robert Edge, Sobel, Scharr, Prewitt, Gaussian and Median filters are among the seven types of filters or kernels that the method has chosen for feature extraction. To develop a full collection of feature values for an image, this generates various feature values for the image data collected in a data frame. The remaining features are removed from the data frame, and a list of important features is found that majorly contribute to the feature space. The retrieved features with annotated masks are sent into the Machine learning models for semantic segmentation. Three learning models-Random Forest, Support Vector Machine and Artificial Neural Network-have been used in this research.

The MeanIoU and Kappa values are used to compare, test and validate each of the three models for correctness. Lastly, the area covered by the various regions is estimated in terms of pixels for agricultural mapping. Figure 6.7 provides a detailed structure of the entire methodology.

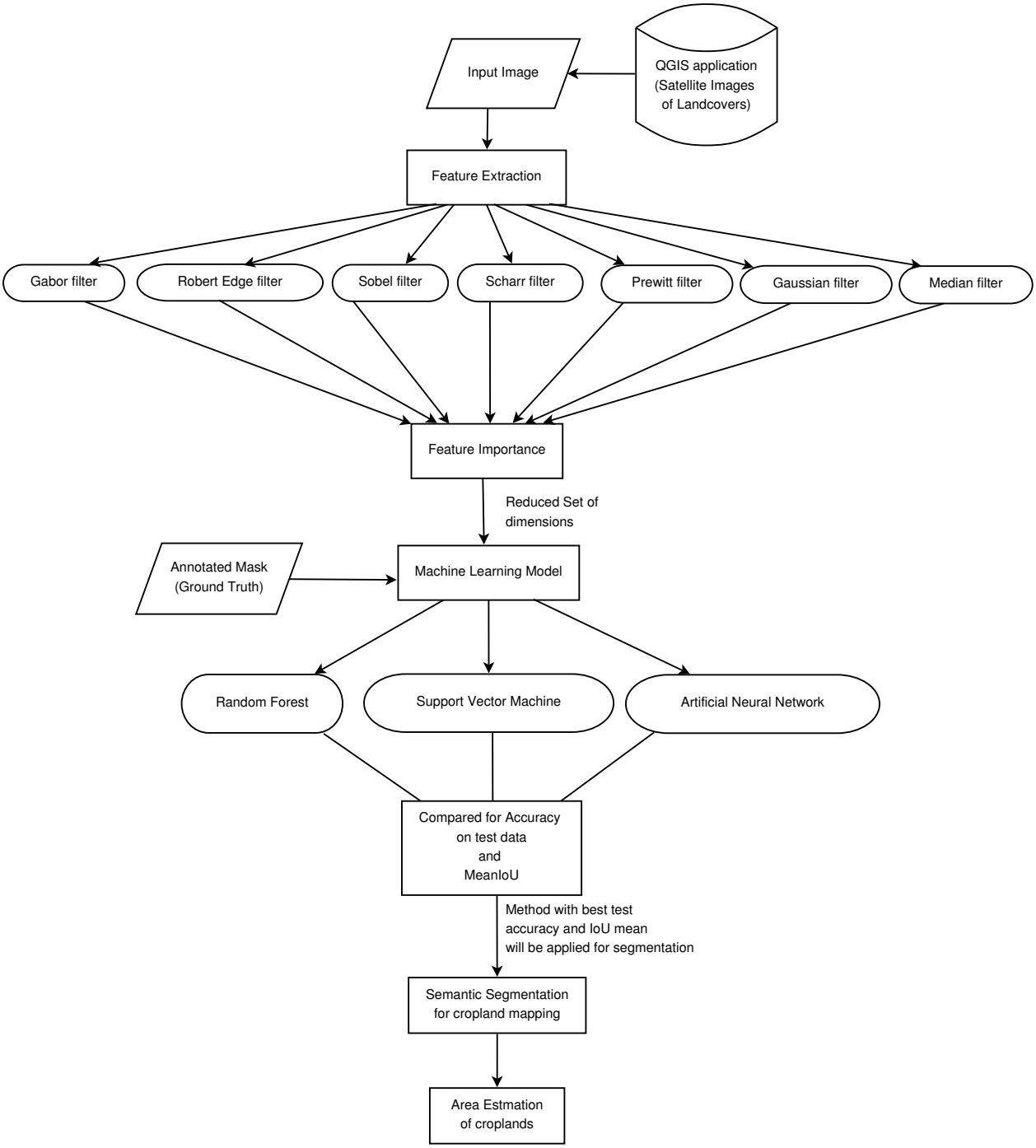


Figure 6.7: Flow chart of the proposed model

6.4.3 Pre-processing of data

The work has used seven different types of feature extraction and edge detection kernels. The result of these filters is compared to determine which filter set has the best values that would ultimately influence image segmentation. These different filters are introduced briefly as follows:

Gabor filter: Gabor filters are used in image processing to extract features and detect edges [198, 199]. They are used for texture-based picture segmentation because they have the best localization properties in the spatial and frequency domains. They are regarded as a type of bandpass filter that accepts a particular range of frequencies while rejecting another range. The Gaussian envelope function is added to a complex oscillation that reduces time and space confusion in these filters to provide an impulsive response [200]. Combining Gaussian and Sinusoidal filters creates a Gabor filter. The Gabor filter is depicted as (6.4.1)) in two dimensions.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2})cos(2\pi\frac{x'}{\lambda} + \psi) \tag{6.4.1}$$

where, $x' = xcos(\theta) + ysin(\theta)$ and $y' = -xsin(\theta) + ycos(\theta)$.
In the equation above, λ represents the wavelength of the sinusoidal factor, θ represents the direction of the parallel line's normal concerning a Gabor function, ψ represents the phase offset, σ represents the standard deviation of the Gaussian filters, and γ designates the spatial aspect ratio that supports the ellipticity of the Gabor function. Here, the Sinusoidal filters display the directionality while the Gaussian filters explain the weights.

Robert Edge filter: The Robert edge filter, which is used for edge detection, is a non-linear filter that successively identifies edges using horizontal and vertical filters before summing the results. The method allows switching from light to dark pixels [201, 202] and is quick enough due to the minimal amount of calculations. The two formulae provided by Robert Edge Filter to highlight the change in intensity diagonally are as follows: (6.4.2) and (6.4.3):

$$b_{i,j} = \sqrt{a_{i,j}} \tag{6.4.2}$$

$$c_{i,j} = \sqrt{(b_{i,j} - b_{i+1,j+1})^2 + (b_{i+1,j} - b_{i,j+1})^2} \tag{6.4.3}$$

where c is the determined derivative, a is the actual image intensity value, and i, j indicates the picture's location.

Sobel filter: By determining the absolute gradient magnitude at each pixel of an image, the Sobel edge detectors highlight high spatial frequency [203, 204]. The filter utilizes the original image convolved with two 3*3 kernels to determine the horizontal and vertical deflection.

Scharr filter: The Scharr method employs the first derivative to determine the gradient edges from the image pixels. It results in the detection of gradients across the x-axis and y-axis separately [205, 206].

Prewitt filter: Prewitt filters use two separate 3*3 kernels to distinguish between edges on the horizontal and vertical axes. For edge calculation, differences in the image's pixel intensities are used. To assess the derivatives, these kernels are convolved with the images [207, 208].

Gaussian filter: Gaussian filtering helps in blurring images to remove noise and features where sigma is the standard deviation that finds an important role in producing the variations in the behavior of Gaussian distribution. The Gaussian filters with sigma ratings of 3 and 7 are used in this chapter. Larger sigma values produce more blurring results. The filter's Gaussian character is maintained by growing the kernel size by raising the sigma value. The Gaussian kernel coefficient [209, 210], is identified by the sigma value. The Gaussian filter is represented by the equation given as (6.4.4):

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6.4.4}$$

where σ is the gaussian distribution's standard deviation, x is the margin from the origin to the horizontal distance, and y is the margin from the origin to the vertical axis.

Median filter: Median filters are used to reduce image noises. Each pixel value in the image is identified by the median filter, which modifies it using the surrounding pixel's median values. Every pixel value is checked against its surroundings. The scores of the surrounding pixels are altered following the value of the central pixel and are then arranged numerically [211, 212]. In Figure 6.8, a chart is used to display the

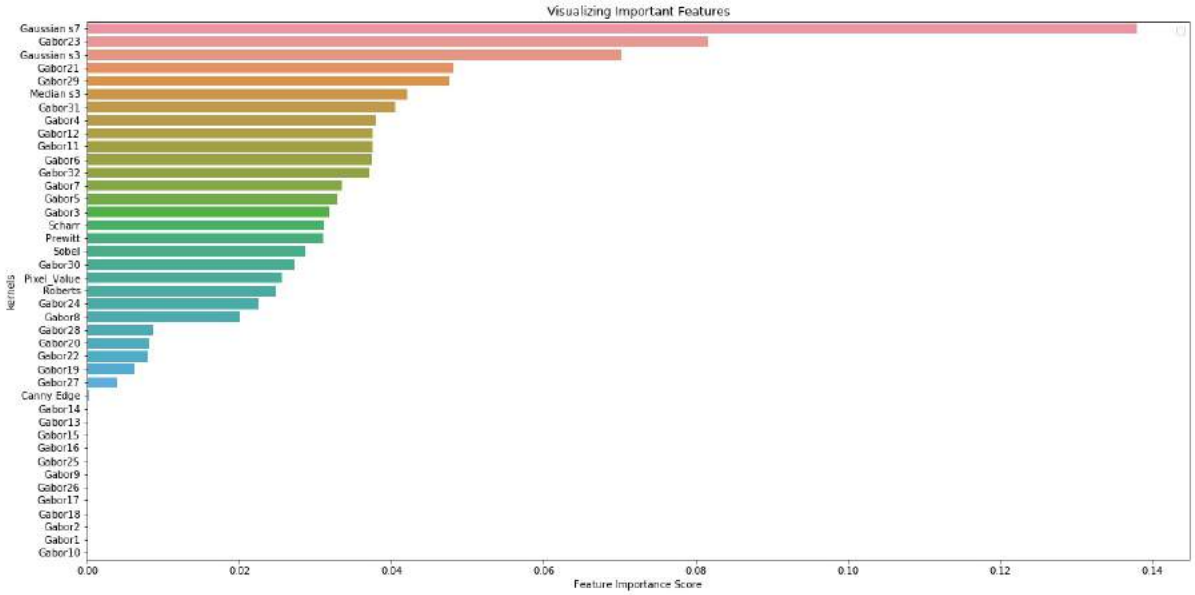


Figure 6.8: Feature Importance of different filters

feature relevance that each filter offers, where the y-axis lists the various kernels or filters employed and the x-axis displays the feature importance score attained by the various filters. The graph shows that certain filters have a noticeable impact, some have a minor impact, and the other filters have no impact on extracting features from the image data. Removing the filter values having a negligible impact helps to decrease the dimensionality.

6.4.4 Machine Learning Models

For land cover semantic segmentation, three machine learning techniques are employed in this work. These are Artificial Neural Networks, Support Vector Machines and Random Forest. A brief introduction to these learning techniques is given below.

Random Forest: Random Forest is used to handle classification and regression issues by combining various classifiers to provide a solution. For overfitting and precision, it is a more advantageous option than a decision tree. The random forest is utilized for semantic segmentation of pictures for a variety of applications, including medical imaging, land cover, plant diseases, geographical characteristics, etc. [135, 136, 213]. Random forest [173, 174] forms a group of trees built from different groups of training data chosen from the total training data, including total feature elements. Each tree in the collection comprises a sample dataset called a bootstrap sample derived from the training data using replacement. The dataset is more diversified by introducing some randomness through feature bagging, which will help minimize the correlation among decision trees. Random forest arranged three main hyper-parameters: node size, number of trees and features before training. In classification, the frequent categorical variable is selected as the predicted class using the majority vote method, but in regression, the average of each decision tree is determined.

Support Vector Machine: The objective of SVM is to locate the ideal hyperplane, a decision boundary that can accurately divide the n-dimensional space into different classes. To perform the task, SVM chooses the extreme points as support vectors for constructing the hyperplane. SVM can be classified as linear and non-linear. A dataset is referred to as linearly separable data or Linear SVM if it can be divided into two distinct groups using just one straight line. On the other hand, non-linear separable data, or non-linear SVM, refers to data that cannot be divided into multiple groups using a single straight line. The SVM kernel performs this problem of data classification. SVM kernel uses the low dimensional data space to convert it into higher dimensional space so that the non-linear data can easily get separated in high dimensional space based on the labels provided. An SVM model is effective for data with multiple features. The method is used for satellite pictures, farmland mapping, land cover identification, disease identification and other applications of image segmentation [128, 214–217].

Artificial neural networks: ANNs, which draw inspiration from biological systems, are networks that can be employed to solve a vast set of problems, covering classification, overfitting, data pre-processing, etc. ANN is also getting utilized for the semantic

segmentation process as well [214,218–220]. The input layer, one or more hidden layers and the output layer are all enclosed within a succession of layers in an ANN. Each node in an ANN is linked to another node through a threshold value, bias and weight. A neuron's output is activated and sends the signal to the following layer if it exceeds the threshold value; else, it is deactivated. These weights determine the significance of any variable; the larger the weights more significantly the variable contributes to the output. This will produce the summation of the inputs multiplied by their respective weights.

6.4.5 Experimentation and Results

The reduced set of feature vectors extracted with the application of different filters as illustrated in Figure 6.8 is forwarded to the Random Forest, SVM and ANN models along with the annotated masks. The segmentation results, thus produced from the machine learning techniques are given in Table 6.2. These results are assessed using the mean IoU, kappa coefficient value, the training and test accuracy. The table is divided into two columns. The metrics obtained for the three models by using only the Gabor filter are shown in one column, while the metrics obtained after applying the reduced set of filters which includes Gabor as well are shown in the other column. It has been identified, that the reduced set of filters exhibits noticeably better outcomes than the case where only the Gabor filter is applied. These results of applying filters are shown in Figure 6.9, which shows the original image, the annotated mask used as the ground truth, the segmented images after the application of the Gabor filter, and the segmented images after all the filters were applied. It is found that the result produced after the application of all filters with lower dimensionality has shown better and clearer segmentation on the test images.

Table 6.2: Measures of a model's effectiveness

	utilising simply gabor filters			applying all dimensionality-reduced filters		
Measures	Random Forest	SVM	ANN	Random Forest	SVM	ANN
Training Accuracy	0.9543	0.5525	0.8352	0.9995	0.6926	0.8656
Test Accuracy	0.8323	0.3486	0.7586	0.9010	0.6928	0.8654
Mean IoU	0.7548	0.1229	0.4574	0.9315	0.3023	0.5884
Kappa score	0.6111	0.1421	0.5998	0.6851	0.3104	0.6280

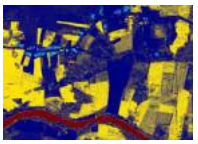
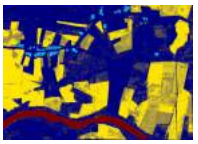
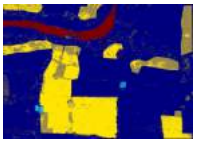
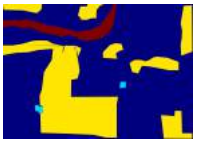
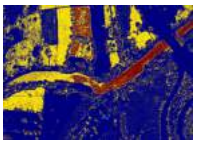
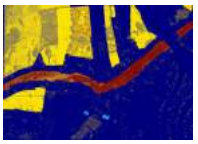
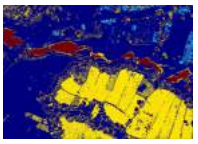
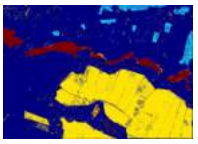
S.No.	Actual Image	Ground Truth	Segmented Image with Only Gabor Filters	Segmented Image all filters with using reduced dimensions
1.				
2.				
3.				
4.				

Figure 6.9: Result of applying filters for semantic segmentation of images

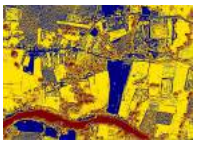
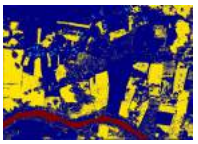
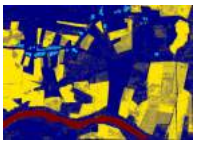
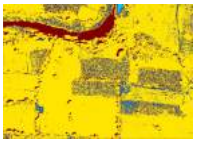
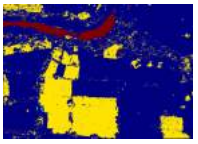
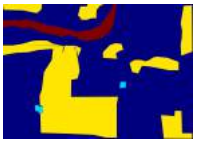
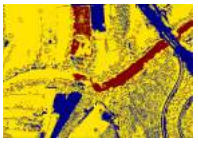
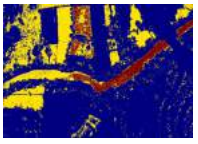
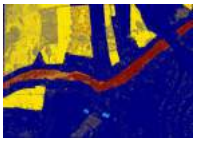
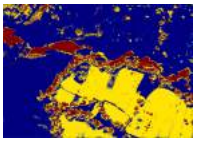
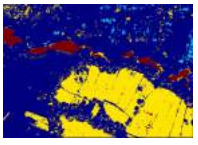
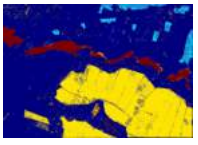
S.No.	Orginal Image	Ground Truth	Segmented Image using SVM	Segmented Image using ANN	Segmented Image using Random Forest
1.					
2.					
3.					
4.					

Figure 6.10: Results of SVM, ANN and Random Forest for semantic segmentation

These conclusions are rather near to the essential truth: light blue regions denote buildings, yellow regions depict farmland, red regions show water, and dark blue regions give the background or empty spaces. Later these images after the application of filters are fed to three different Machine learning techniques. The results of applying machine learning techniques to these images can be seen in Figure 6.10, which compares and displays the results of segmentation on the sample images. It has been found that Random Forest outperforms the other two Machine Learning models in test accuracy, Mean IoU value and kappa score. It achieves a maximum MeanIoU value of roughly 93% and a test accuracy of 90% better than the other two techniques. Thus, choosing a Random forest for semantic segmentation of land covers can be a better option. Finally, these segmented regions are estimated to determine the region covered within different regions, where the area is divided into the three specified regions-buildings, farms, and water. Area estimated in terms of a number of pixels can be mapped to actual segmentation by using the ratio of image resolution to land area. The area estimated under the different regions is shown in Table 6.3. The table displays the labels, the names of the associated regions, and the area covered under different regions for different pictures.

Table 6.3: Region-specific area estimation (pixel value)

		area identified under different regions			
Labels	Region Name	picture 1	picture 2	picture 3	Picture 4
1	Buildings	360024	88275	44784	831792
2	Croplands	9642186	7200786	7082988	6897957
3	Water	1086315	1090239	1878324	1469664

6.5 CHAPTER SUMMARY

Proper land cultivation is necessary to meet the rising need for food production and supply. The solution to this issue is to identify arid and uncultivated terrain. The chapter has studied and presented the process for segmenting the land cover area acquired from Google satellite pictures into several regions for agricultural mapping. The adopted images are divided into four land cover areas−croplands, rivers, buildings and uncultivated fields shown as background.

This work has performed the semantic segmentation of agricultural lands using aerial or satellite images through two different approaches namely thresholding-based technique

and machine learning techniques to detect the region of interest or area under cultivation. Initially, the semantic segmentation process is carried out through four thresholding-based techniques like histogram-based segmentation, Multi-Otsu-based segmentation, K-means-based segmentation and Random Walker segmentation, where the k-means algorithm performed best to estimate the regions of cropland against uncultivated or urbanized areas. K-means help in reducing the variance between classes by finding the close similarity in texture. It is also found that the methods like Histogram, Multi-Otsu and Random Walker need a grey-level histogram before segmentation, which sometimes produces noises, whereas, K-means thresholding does not require histograms, which makes it appropriate for 2D and 3D multi-level thresholding. But due to the static approach followed by the thresholding techniques the segmented results are not found to be correctly segmented. As of this, another approach employing machine learning is getting utilized for the segmentation of croplands. This process is carried out initially by extracting the relevant set of features through the seven different types of filters or kernels, which is followed by the application of three Machine learning approaches−Random Forest, SVM, and ANN. The analysis and findings show that Random Forest, with 99% training accuracy, 90% test accuracy, 93% MeanIoU value and with roughly 69% kappa score produces the best result for cropland segmentation. Despite the textural complexity and pixel proximity beneath different locations, the method successfully divided the cultivation area through multiple iterations. Finally, the segmented areas under the croplands are estimated to identify the area covered under the cultivation. Area estimated in terms of a number of pixels can be mapped to actual segmentation by using the ratio of image resolution to land area.

CHAPTER VII

CONCLUSION AND FUTURE DIRECTIONS

CHAPTER 7

CONCLUSION AND FUTURE DIRECTIONS

The research work performed in this thesis emphasizes the automation of agricultural practices through artificial intelligence and machine learning. This research has covered various research questions related to crop cultivation and production and successfully developed suitable solutions for them.

7.1 CONCLUSION

The research work incorporated in this thesis has addressed the issues in the automation of agricultural practices through AI and Machine learning. The development and use of analytical and predictive machine learning techniques for a variety of agricultural concerns are the focus of this thesis. An extensive study of existing methods and techniques related to agricultural problems has been performed based on which the different limitations and gaps were identified and worked on.
The proposed work achieves the accompanying objectives:

- **Classification result:** The classification of agricultural illnesses and grains has previously been done using various machine learning techniques. However, the classification results of techniques along with transfer learning are limited to the structure of data. The proposed "Agri-CNN" model has shown much better and more promising results for classifying agriculture grains and diseases than the earlier machine learning techniques.

- **Dimensionality Reduction:** The proposed dimensionality reduction "Info_PCA" technique has performed well for feature selection and extraction for the vast

dataset. The extensive set of features may sometimes degrade the performance of the classification of prediction technique rather than improving it. The proposed model results are substantially better than the other dimensionality reduction technique in overall identifying the classification and prediction accuracy.

- **Crop yield prediction:** To forecast agricultural yield, the hybrid "RaNN" model has been developed. Comparing the proposed method to other machine learning techniques previously used for similar problems, it has demonstrated superior accuracy and minimal loss.

- **Land cover mapping and area estimation:** A method is proposed for semantic segmentation of land areas gathered from the Google satellite imagery into several regions for farmland mapping, with the land cover including farmland, rivers, buildings and vacant land as background. These discovered brownfields could be utilized for other productive reasons like planting forests to maintain the ecological balance. It has also proposed a method for calculating the area within the various identified regions to detect the area covered under the occupied and uncultivated lands. No such method is found in the literature that has performed the semantic segmentation of the land region and area calculation of the land identified together.

Comparing the output of the proposed models to the prior machine models reveals the following conclusions:

- The proposed model for image classification is identified as better than the transfer learning techniques due to the reduced network size and thereby reduces the learning time.

- The developed "Agri-CNN" model is hyper-tuned by changing the parameters to improve its performance for grain classification. It generates the best parameters for multi-dimensional data classification to achieve better results.

- A hybrid dimensionality reduction technique "Info_PCA" has been proposed for significant feature selection to reduce overfitting.

- A "RaNN" model has been proposed for predicting statistical data that has shown promising results over the traditional methods.

- The land cover mapping and the calculation of the area occupied by each region are both accurately carried out by the semantic segmentation methods that have been studied.

7.2 SCOPE FOR FUTURE STUDY

Due to the shortage of resources and time, every research has some limitations, so the case with this research work. There are several issues in the research work which can be resolved and extended in future studies.

- The classification of several other grains with various variations and complexities, such as the presence or lack of complicated backgrounds, can be performed using the current study on wheat grains.
- To detect and distinguish between good and low-quality grains, the process of grain classification can be improved.
- The issues like the degree or severity of the disease in a given crop can be addressed by using the infected leaf area and segmentation techniques.
- The estimated yield can be used to determine the land cover used for agriculture to further the study effort on segmenting farmland regions.
- The semantic segmentation with thresholding techniques can be expanded with edge detection and region-based algorithms.
- The semantic segmentation of satellite images can be extended with the deep learning technique involving a large dataset.
- The estimated land area can be used for yield estimation provided certain authentic data mapping is available.
- To estimate the growth of plants and trees based on earlier observations, to predict their yield and identify their water needs.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. Uthayakumaran, I. L. Batey, and C. W. Wrigley, "Variety identification to suit the needs of industry," *Cereals 2005*, 2005.

[2] K. Jha, A. Doshi, P. Patel, and M. Shah, "A comprehensive review on automation in agriculture using artificial intelligence," *Artificial Intelligence in Agriculture*, vol. 2, pp. 1–12, 2019.

[3] I. Rakhmatulin, "Neural network for automatic farm control," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 1, pp. 147–160, 2021.

[4] P. Zapotoczny, "Discrimination of wheat grain varieties using image analysis and multidimensional analysis texture of grain mass," *International Journal of food properties*, vol. 17, no. 1, pp. 139–151, 2014.

[5] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[6] S. Marsland, *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

[7] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Computers and electronics in agriculture*, vol. 153, pp. 69–81, 2018.

[8] L. Gao and W. Wu, "Relevance assignation feature selection method based on mutual information for machine learning," *Knowledge-Based Systems*, vol. 209, p. 106439, 2020.

[9] V. Sivakumar, R. Anandalakshmi, R. R. Warrier, B. G. Singh, M. Tigabu, and B. Nagarajan, "Discrimination of acacia seeds at species and subspecies levels using an image analyzer," *Forest Science and Practice*, vol. 15, no. 4, pp. 253–260, 2013.

[10] M. D. Nguyen, B. T. Pham, L. S. Ho, H.-B. Ly, T.-T. Le, C. Qi, V. M. Le, L. M. Le, I. Prakash, D. T. Bui *et al.*, "Soft-computing techniques for prediction of soils consolidation coefficient," *Catena*, vol. 195, p. 104802, 2020.

[11] L. Del Vasto-Terrientes, V. Kumar, T. Chao, and A. Valls, "A decision support system to find the best water allocation strategies in a mediterranean river basin in future scenarios of global change," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 1-2, pp. 331–350, 2016.

[12] N. Prasad, N. Patel, and A. Danodia, "Crop yield prediction in cotton for regional level using random forest approach," *Spatial Information Research*, vol. 29, no. 2, pp. 195–206, 2021.

[13] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[14] S. G. Sodjinou, V. Mohammadi, A. T. S. Mahama, and P. Gouton, "A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images," *Information Processing in Agriculture*, 2021.

[15] R. Garcia, J. Aguilar, M. Toro, A. Pinto, and P. Rodriguez, "A systematic literature review on the use of machine learning in precision livestock farming," *Computers and Electronics in Agriculture*, vol. 179, p. 105826, 2020.

[16] S. Godara and D. Toshniwal, "Deep learning-based query-count forecasting using farmers' helpline data," *Computers and Electronics in Agriculture*, vol. 196, p. 106875, 2022.

[17] E. M. El Houby, "A survey on applying machine learning techniques for management of diseases," *Journal of Applied Biomedicine*, vol. 16, no. 3, pp. 165–174, 2018.

[18] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine learning and data mining methods in diabetes research," *Computational and structural biotechnology journal*, vol. 15, pp. 104–116, 2017.

[19] H. Wu, S. Yang, Z. Huang, J. He, and X. Wang, "Type 2 diabetes mellitus prediction model based on data mining," *Informatics in Medicine Unlocked*, vol. 10, pp. 100–107, 2018.

[20] A. Mucherino, P. Papajorgji, and P. M. Pardalos, "A survey of data mining techniques applied to agriculture," *Operational Research*, vol. 9, no. 2, pp. 121–140, 2009.

[21] J. Fürnkranz, D. Gamberger, and N. Lavrač, "Machine learning and data mining," in *Foundations of Rule Learning*. Springer, 2012, pp. 1–17.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[23] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *IEEE computational intelligence magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[24] M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, "Deep learning for plant diseases: detection and saliency map visualisation," in *Human and machine learning*. Springer, 2018, pp. 93–117.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[26] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.

[27] K. Thaiyalnayaki and C. Joseph, "Classification of plant disease using svm and deep learning," *Materials Today: Proceedings*, 2021.

[28] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[29] D. Forsyth and J. Ponce, *Computer vision: A modern approach.* Prentice hall, 2011.

[30] F. Kurtulmuş and H. I. Ünal, "Discriminating rapeseed varieties using computer vision and machine learning," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1880–1891, 2015.

[31] J. Han, M. Kamber, and J. Pei, "Data mining: Concepts and techniques third edition, third edit," 2012.

[32] N. U. Moroff, E. Kurt, and J. Kamphues, "Machine learning and statistics: A study for assessing innovative demand forecasting models," *Procedia Computer Science*, vol. 180, pp. 40–49, 2021.

[33] D. Di Caprio and F. J. Santos-Arteaga, "Enhancing the pattern recognition capacity of machine learning techniques: The importance of feature positioning," *Machine Learning with Applications*, vol. 7, p. 100196, 2022.

[34] N. Gupta, M. Khosravy, S. Gupta, N. Dey, and R. G. Crespo, "Lightweight artificial intelligence technology for health diagnosis of agriculture vehicles: parallel evolving artificial neural networks by genetic algorithm," *International Journal of Parallel Programming*, pp. 1–26, 2022.

[35] Y. Kaya and L. Kayci, "Application of artificial neural network for automatic detection of butterfly species using color and texture features," *The visual computer*, vol. 30, no. 1, pp. 71–79, 2014.

[36] K. Sabanci, A. Kayabasi, and A. Toktas, "Computer vision-based method for classification of wheat grains using artificial neural network," *Journal of the Science of Food and Agriculture*, vol. 97, no. 8, pp. 2588–2593, 2017.

[37] E. O. Güneş, S. Aygün, M. Kırcı, A. Kalateh, and Y. Çakır, "Determination of the varieties and characteristics of wheat seeds grown in turkey using image processing techniques," in *2014 The Third International Conference on Agro-Geoinformatics*. IEEE, 2014, pp. 1–4.

[38] T.-Y. Kuo, C.-L. Chung, S.-Y. Chen, H.-A. Lin, and Y.-F. Kuo, "Identifying rice grains using image analysis and sparse-representation-based classification," *Computers and Electronics in Agriculture*, vol. 127, pp. 716–725, 2016.

[39] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[40] X. Liu, Q. Xu, and N. Wang, "A survey on deep neural network-based image captioning," *The Visual Computer*, vol. 35, no. 3, pp. 445–470, 2019.

[41] H. Zareiforoush, S. Minaei, M. R. Alizadeh, and A. Banakar, "Potential applications of computer vision in quality inspection of rice: a review," *Food engineering reviews*, vol. 7, no. 3, pp. 321–345, 2015.

[42] A. Pazoki, F. Farokhi, and Z. Pazoki, "Classification of rice grain varieties using two artificial neural networks (mlp and neuro-fuzzy)," *The Journal of Animal & Plant Sciences*, vol. 24, no. 1, pp. 336–343, 2014.

[43] K. Sabanci, A. Toktas, and A. Kayabasi, "Grain classifier with computer vision using adaptive neuro-fuzzy inference system," *Journal of the Science of Food and Agriculture*, vol. 97, no. 12, pp. 3994–4000, 2017.

[44] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic investigation on a robust and practical plant diagnostic system," in *2016 15th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2016, pp. 989–992.

[45] E. L. Mique Jr and T. D. Palaoag, "Rice pest and disease detection using convolutional neural network," in *Proceedings of the 2018 International Conference on Information Science and System*, 2018, pp. 147–151.

[46] N. Petrellis, "Mobile application for plant disease classification based on symptom signatures," in *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, 2017, pp. 1–6.

[47] J. Wang, L. Chen, J. Zhang, Y. Yuan, M. Li, and W. Zeng, "Cnn transfer learning for automatic image-based classification of crop disease," in *Chinese Conference on Image and Graphics Technologies*. Springer, 2018, pp. 319–329.

[48] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and electronics in agriculture*, vol. 153, pp. 46–53, 2018.

[49] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "Toled: Tomato leaf disease detection using convolution neural network," *Procedia Computer Science*, vol. 167, pp. 293–301, 2020.

[50] R. Thangaraj, S. Anandamurugan, and V. K. Kaliappan, "Automated tomato leaf disease classification using transfer learning-based deep convolution neural network," *Journal of Plant Diseases and Protection*, vol. 128, no. 1, pp. 73–86, 2021.

[51] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia computer science*, vol. 133, pp. 1040–1047, 2018.

[52] C. Xie, C. Yang, and Y. He, "Hyperspectral imaging for classification of healthy and gray mold diseased tomato leaves with different infection severities," *Computers and electronics in agriculture*, vol. 135, pp. 154–162, 2017.

[53] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with c-gan synthetic images," *Computers and Electronics in Agriculture*, vol. 187, p. 106279, 2021.

[54] A. Bhatia, A. Chug, A. P. Singh, R. P. Singh, and D. Singh, "A machine learning-based spray prediction model for tomato powdery mildew disease," *Indian Phytopathology*, pp. 1–6, 2021.

[55] S. Ashwinkumar, S. Rajagopal, V. Manimaran, and B. Jegajothi, "Automated plant leaf disease detection and classification using optimal mobilenet based convolutional neural networks," *Materials Today: Proceedings*, 2021.

[56] V. Sravan, K. Swaraj, K. Meenakshi, and P. Kora, "A deep learning based crop disease classification using transfer learning," *Materials Today: Proceedings*, 2021.

[57] T. Subetha, R. Khilar, and M. S. Christo, "A comparative analysis on plant pathology classification using deep learning architecture–resnet and vgg19," *Materials Today: Proceedings*, 2021.

[58] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.

[59] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning cnn models for disease detection in plants using image segmentation," *Information Processing in Agriculture*, vol. 7, no. 4, pp. 566–574, 2020.

[60] K. Thenmozhi and U. S. Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Computers and Electronics in Agriculture*, vol. 164, p. 104906, 2019.

[61] M. G. Larese, R. Namías, R. M. Craviotto, M. R. Arango, C. Gallo, and P. M. Granitto, "Automatic classification of legumes using leaf vein image features," *Pattern Recognition*, vol. 47, no. 1, pp. 158–168, 2014.

[62] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.

[63] J. Pant, R. Pant, M. K. Singh, D. P. Singh, and H. Pant, "Analysis of agricultural crop yield prediction using statistical techniques of machine learning," *Materials Today: Proceedings*, 2021.

[64] T. Van Klompenburg, A. Kassahun, and C. Catal, "Crop yield prediction using machine learning: A systematic literature review," *Computers and Electronics in Agriculture*, vol. 177, p. 105709, 2020.

[65] Y. Guo, Y. Fu, F. Hao, X. Zhang, W. Wu, X. Jin, C. R. Bryant, and J. Senthilnath, "Integrated phenology and climate in rice yields prediction using machine learning methods," *Ecological Indicators*, vol. 120, p. 106935, 2021.

[66] A. S. Abade, P. A. Ferreira, and F. d. B. Vidal, "Plant diseases recognition on images using convolutional neural networks: A systematic review," *arXiv preprint arXiv:2009.04365*, 2020.

[67] A. Kaur and S. K. Sood, "Cloud-fog based framework for drought prediction and forecasting using artificial neural network and genetic algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 32, no. 2, pp. 273–289, 2020.

[68] B. Basso, C. Fiorentino, D. Cammarano, and U. Schulthess, "Variable rate nitrogen fertilizer response in wheat using remote sensing," *Precision agriculture*, vol. 17, no. 2, pp. 168–182, 2016.

[69] W. W. Guo and H. Xue, "An incorporative statistic and neural approach for crop yield modelling and forecasting," *Neural Computing and Applications*, vol. 21, no. 1, pp. 109–117, 2012.

[70] N. R. Peralta, Y. Assefa, J. Du, C. J. Barden, and I. A. Ciampitti, "Mid-season high-resolution satellite imagery for forecasting site-specific corn yield," *Remote Sensing*, vol. 8, no. 10, p. 848, 2016.

[71] B. Liu, S. Asseng, C. Müller, F. Ewert, J. Elliott, D. B. Lobell, P. Martre, A. C. Ruane, D. Wallach, J. W. Jones *et al.*, "Similar estimates of temperature impacts on global wheat yield by three independent methods," *Nature Climate Change*, vol. 6, no. 12, pp. 1130–1136, 2016.

[72] M. Safa and S. Samarasinghe, "Determination and modelling of energy consumption in wheat production using neural networks:"a case study in canterbury province, new zealand"," *Energy*, vol. 36, no. 8, pp. 5140–5147, 2011.

[73] Y. Liu, G. B. Heuvelink, Z. Bai, P. He, X. Xu, W. Ding, and S. Huang, "Analysis of spatio-temporal variation of crop yield in china using stepwise multiple linear regression," *Field Crops Research*, vol. 264, p. 108098, 2021.

[74] E. Khosla, R. Dharavath, and R. Priya, "Crop yield prediction using aggregated rainfall-based modular artificial neural networks and support vector regression," *Environment, Development and Sustainability*, pp. 1–22, 2019.

[75] A. Haghverdi, R. A. Washington-Allen, and B. G. Leib, "Prediction of cotton lint yield from phenology of crop indices using artificial neural networks," *Computers and Electronics in Agriculture*, vol. 152, pp. 186–197, 2018.

[76] R. J. Schalkoff, *Artificial neural networks*. McGraw-Hill Higher Education, 1997.

[77] P. M. Gopal and R. Bhargavi, "A novel approach for efficient crop yield prediction," *Computers and Electronics in Agriculture*, vol. 165, p. 104968, 2019.

[78] D. Seka, B. Bonny, A. N. Yoboué, S. Sié, and B. Adopo-Gourène, "Identification of maize (zea mays l.) progeny genotypes based on two probabilistic approaches: Logistic regression and naïve bayes," *Artificial Intelligence in Agriculture*, vol. 1, pp. 9–13, 2019.

[79] N. Gandhi, L. J. Armstrong, and O. Petkar, "Predicting rice crop yield using bayesian networks," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 795–799.

[80] H. Khan and S. Ghosh, "Crop yield prediction from meteorological data using efficient machine learning model," in *Proceedings of International Conference on Wireless Communication*. Springer, 2020, pp. 565–574.

[81] K. Gibert, M. Sànchez-Marrè, and J. Izquierdo, "A survey on pre-processing techniques: Relevant issues in the context of environmental data mining," *AI Communications*, vol. 29, no. 6, pp. 627–663, 2016.

[82] P. Nevavuori, N. Narra, and T. Lipping, "Crop yield prediction with deep convolutional neural networks," *Computers and electronics in agriculture*, vol. 163, p. 104859, 2019.

[83] C. Zhao, B. Liu, S. Piao, X. Wang, D. B. Lobell, Y. Huang, M. Huang, Y. Yao, S. Bassu, P. Ciais *et al.*, "Temperature increase reduces global yields of major crops in four independent estimates," *Proceedings of the National Academy of Sciences*, vol. 114, no. 35, pp. 9326–9331, 2017.

[84] L. Tian, C. Wang, H. Li, and H. Sun, "Yield prediction model of rice and wheat crops based on ecological distance algorithm," *Environmental Technology & Innovation*, vol. 20, p. 101132, 2020.

[85] S. Hira and P. Deshpande, "Data analysis using multidimensional modeling, statistical analysis and data mining on agriculture parameters," *Procedia Computer Science*, vol. 54, pp. 431–439, 2015.

[86] B. Peng, K. Guan, W. Zhou, C. Jiang, C. Frankenberg, Y. Sun, L. He, and P. Köhler, "Assessing the benefit of satellite-based solar-induced chlorophyll

fluorescence in crop yield prediction," *International Journal of Applied Earth Observation and Geoinformation*, vol. 90, p. 102126, 2020.

[87] A. Anitha and D. P. Acharjya, "Crop suitability prediction in vellore district using rough set on fuzzy approximation space and neural network," *Neural Computing and Applications*, vol. 30, no. 12, pp. 3633–3650, 2018.

[88] N. Gandhi and L. J. Armstrong, "A review of the application of data mining techniques for decision making in agriculture," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, pp. 1–6.

[89] M. Trigui, K. Gabsi, I. E. Amri, A. N. Helal, and S. Barrington, "Modular feed forward networks to predict sugar diffusivity from date pulp part i. model validation," *International Journal of food properties*, vol. 14, no. 2, pp. 356–370, 2011.

[90] S. Shidnal, M. V. Latte, and A. Kapoor, "Crop yield prediction: two-tiered machine learning model approach," *International Journal of Information Technology*, pp. 1–9, 2019.

[91] J. Majumdar, S. Naraseeyappa, and S. Ankalaki, "Analysis of agriculture data using data mining techniques: application of big data," *Journal of Big data*, vol. 4, no. 1, pp. 1–15, 2017.

[92] N. Gandhi and L. J. Armstrong, "Assessing impact of seasonal rainfall on rice crop yield of rajasthan, india using association rule mining," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 1021–1024.

[93] S. Veenadhari, B. Misra, and C. Singh, "Machine learning approach for forecasting crop yield based on climatic parameters," in *2014 International Conference on Computer Communication and Informatics*. IEEE, 2014, pp. 1–5.

[94] R. D. Baruah, S. Roy, R. Bhagat, and L. N. Sethi, "Use of data mining technique for prediction of tea yield in the face of climate change of assam, india," in *2016 International Conference on Information Technology (ICIT)*. IEEE, 2016, pp. 265–269.

[95] A. Shastry, H. Sanjay, and E. Bhanusree, "Prediction of crop yield using regression techniques," *International Journal of Soft Computing*, vol. 12, no. 2, pp. 96–102, 2017.

[96] Y. Liu, "Statistical analysis and modelling of crop yield and nitrogen use efficiency in china," Ph.D. dissertation, Wageningen University, 2022.

[97] J. Ma and Y. Yuan, "Dimension reduction of image deep feature using pca," *Journal of Visual Communication and Image Representation*, vol. 63, p. 102578, 2019.

[98] C. F. Bowdren and R. Santo, "Sustainable diets for a food-secure future," in *Environmental Nutrition*. Elsevier, 2019, pp. 285–303.

[99] E. Ndikumana, D. Ho Tong Minh, N. Baghdadi, D. Courault, and L. Hossard, "Deep recurrent neural network for agricultural classification using multitemporal sar sentinel-1 for camargue, france," *Remote Sensing*, vol. 10, no. 8, p. 1217, 2018.

[100] S. Kumar and P. Jayagopal, "Delineation of field boundary from multispectral satellite images through u-net segmentation and template matching," *Ecological Informatics*, vol. 64, p. 101370, 2021.

[101] Y. H. Robinson, S. Vimal, M. Khari, F. C. L. Hernández, and R. G. Crespo, "Tree-based convolutional neural networks for object classification in segmented satellite images," *The International Journal of High Performance Computing Applications*, p. 1094342020945026, 2020.

[102] U. Verma, A. Chauhan, M. P. MM, and R. Pai, "Deeprivwidth: Deep learning based semantic segmentation approach for river identification and width measurement in sar images of coastal karnataka," *Computers & Geosciences*, vol. 154, p. 104805, 2021.

[103] M. Belgiu and A. Stein, "Spatiotemporal image fusion in remote sensing," *Remote sensing*, vol. 11, no. 7, p. 818, 2019.

[104] Z. Du, J. Yang, C. Ou, and T. Zhang, "Smallholder crop area mapped with a semantic segmentation deep learning method," *Remote Sensing*, vol. 11, no. 7, p. 888, 2019.

[105] S. Wei, H. Zhang, C. Wang, Y. Wang, and L. Xu, "Multi-temporal sar data large-scale crop mapping based on u-net model," *Remote Sensing*, vol. 11, no. 1, p. 68, 2019.

[106] A. Mercier, J. Betbeder, J. Baudry, V. Le Roux, F. Spicher, J. Lacoux, D. Roger, and L. Hubert-Moy, "Evaluation of sentinel-1 & 2 time series for predicting wheat and rapeseed phenological stages," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 163, pp. 231–256, 2020.

[107] P. Sidike, V. Sagan, M. Maimaitijiang, M. Maimaitiyiming, N. Shakoor, J. Burken, T. Mockler, and F. B. Fritschi, "dpen: Deep progressively expanded network for mapping heterogeneous agricultural landscape using worldview-3 satellite imagery," *Remote sensing of environment*, vol. 221, pp. 756–772, 2019.

[108] K. Van Tricht, A. Gobin, S. Gilliams, and I. Piccard, "Synergistic use of radar sentinel-1 and optical sentinel-2 imagery for crop mapping: a case study for belgium," *Remote Sensing*, vol. 10, no. 10, p. 1642, 2018.

[109] J. Adrian, V. Sagan, and M. Maimaitijiang, "Sentinel sar-optical fusion for crop type mapping using deep learning and google earth engine," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 175, pp. 215–235, 2021.

[110] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[111] N. Torbick, X. Huang, B. Ziniti, D. Johnson, J. Masek, and M. Reba, "Fusion of moderate resolution earth observations for operational crop type mapping," *Remote Sensing*, vol. 10, no. 7, p. 1058, 2018.

[112] C. Buckley and P. Carney, "The potential to reduce the risk of diffuse pollution from agriculture while improving economic performance at farm level," *Environmental science & policy*, vol. 25, pp. 118–126, 2013.

[113] Y. Guo, X. Jia, D. Paull, and J. A. Benediktsson, "Nomination-favoured opinion pool for optical-sar-synergistic rice mapping in face of weakened flooding

signals," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 155, pp. 187–205, 2019.

[114] S. Fritz, L. See, J. C. L. Bayas, F. Waldner, D. Jacques, I. Becker-Reshef, A. Whitcraft, B. Baruth, R. Bonifacio, J. Crutchfield *et al.*, "A comparison of global agricultural monitoring systems and current gaps," *Agricultural systems*, vol. 168, pp. 258–272, 2019.

[115] D. Zhang, Y. Pan, J. Zhang, T. Hu, J. Zhao, N. Li, and Q. Chen, "A generalized approach based on convolutional neural networks for large area cropland mapping at very high resolution," *Remote Sensing of Environment*, vol. 247, p. 111912, 2020.

[116] M. F. Vogels, S. M. de Jong, G. Sterk, and E. A. Addink, "Agricultural cropland mapping using black-and-white aerial photography, object-based image analysis and random forests," *International journal of applied earth observation and geoinformation*, vol. 54, pp. 114–123, 2017.

[117] S. Du, S. Du, B. Liu, and X. Zhang, "Mapping large-scale and fine-grained urban functional zones from vhr images using a multi-scale semantic segmentation network and object based approach," *Remote Sensing of Environment*, vol. 261, p. 112480, 2021.

[118] M. Belgiu and O. Csillik, "Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis," *Remote sensing of environment*, vol. 204, pp. 509–523, 2018.

[119] B. Du, Z. Zhao, X. Hu, G. Wu, L. Han, L. Sun, and Q. Gao, "Landslide susceptibility prediction based on image semantic segmentation," *Computers & Geosciences*, p. 104860, 2021.

[120] D. Jiang, G. Li, C. Tan, L. Huang, Y. Sun, and J. Kong, "Semantic segmentation for multiscale target based on object recognition using the improved faster-rcnn model," *Future Generation Computer Systems*, vol. 123, pp. 94–104, 2021.

[121] R. Barth, J. IJsselmuiden, J. Hemming, and E. J. Van Henten, "Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation," *Computers and Electronics in Agriculture*, vol. 161, pp. 291–304, 2019.

[122] W. Zhao, Y. Fu, X. Wei, and H. Wang, "An improved image semantic segmentation method based on superpixels and conditional random fields," *Applied Sciences*, vol. 8, no. 5, p. 837, 2018.

[123] M. Villamizar, O. Canévet, and J.-M. Odobez, "Multi-scale sequential network for semantic text segmentation and localization," *Pattern Recognition Letters*, vol. 129, pp. 63–69, 2020.

[124] N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Asian conference on computer vision*. Springer, 2016, pp. 180–196.

[125] J. Li, B. Xi, Q. Du, R. Song, Y. Li, and G. Ren, "Deep kernel extreme-learning machine for the spectral–spatial classification of hyperspectral imagery," *Remote Sensing*, vol. 10, no. 12, p. 2036, 2018.

[126] A. Agapiou, "Land cover mapping from colorized corona archived greyscale satellite data and feature extraction classification," *Land*, vol. 10, no. 8, p. 771, 2021.

[127] S. Shayeganpour, M. H. Tangestani, and P. V. Gorsevski, "Machine learning and multi-sensor data fusion for mapping lithology: A case study of kowli-kosh area, sw iran," *Advances in Space Research*, vol. 68, no. 10, pp. 3992–4015, 2021.

[128] B. Chen, M. Xia, and J. Huang, "Mfanet: a multi-level feature aggregation network for semantic segmentation of land cover," *Remote Sensing*, vol. 13, no. 4, p. 731, 2021.

[129] M. S. Moran, Y. Inoue, and E. Barnes, "Opportunities and limitations for image-based remote sensing in precision crop management," *Remote sensing of Environment*, vol. 61, no. 3, pp. 319–346, 1997.

[130] A. Orynbaikyzy, U. Gessner, and C. Conrad, "Crop type classification using a combination of optical and radar remote sensing data: a review," *international journal of remote sensing*, vol. 40, no. 17, pp. 6553–6595, 2019.

[131] B. Kang and T. Q. Nguyen, "Random forest with learned representations for semantic segmentation," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3542–3555, 2019.

[132] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[133] A. Smith, "Image segmentation scale parameter optimization and land cover classification using the random forest algorithm," *Journal of Spatial Science*, vol. 55, no. 1, pp. 69–79, 2010.

[134] J. Mallick, S. Talukdar, S. Pal, A. Rahman *et al.*, "A novel classifier for improving wetland mapping by integrating image fusion techniques and ensemble machine learning classifiers," *Ecological Informatics*, vol. 65, p. 101426, 2021.

[135] E. Saralioglu and O. Gungor, "Semantic segmentation of land cover from high resolution multispectral satellite images by spectral-spatial convolutional neural network," *Geocarto International*, pp. 1–21, 2020.

[136] D. Ravì, M. Bober, G. M. Farinella, M. Guarnera, and S. Battiato, "Semantic segmentation of images exploiting dct based features and random forest," *Pattern Recognition*, vol. 52, pp. 260–273, 2016.

[137] T. Liu, A. Abd-Elrahman, J. Morton, and V. L. Wilhelm, "Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system," *GIScience & remote sensing*, vol. 55, no. 2, pp. 243–264, 2018.

[138] S. Kalaivani, S. Shantharajah, and T. Padma, "Agricultural leaf blight disease segmentation using indices based histogram intensity segmentation approach," *Multimedia Tools and Applications*, vol. 79, no. 13, pp. 9145–9159, 2020.

[139] A. R. Phalke and M. Özdoğan, "Large area cropland extent mapping with landsat data and a generalized classifier," *Remote sensing of environment*, vol. 219, pp. 180–195, 2018.

[140] V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features: a machine learning paradigm," *Journal of Plant Pathology*, vol. 103, no. 1, pp. 17–26, 2021.

[141] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *2015 IEEE international conference on image processing (ICIP)*. IEEE, 2015, pp. 452–456.

[142] K. Kamal, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, p. 104948, 2019.

[143] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Computers and electronics in agriculture*, vol. 158, pp. 226–240, 2019.

[144] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems engineering*, vol. 144, pp. 52–60, 2016.

[145] R. Girshick and R. Fast, "Ieee int. conf. comput. vis," *Santiago, Chile, December*, pp. 7–13, 2015.

[146] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[147] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[148] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 120–147, 2018.

[149] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization. arxiv: 14126980," *Retrieved from the arXiv database*, 2014.

[150] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv preprint arXiv:1704.00805*, 2017.

[151] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[152] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzande, "Review of swarm intelligence-based feature selection methods," *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104210, 2021.

[153] Z. J. Viharos, K. B. Kis, Á. Fodor, and M. I. Büki, "Adaptive, hybrid feature selection (ahfs)," *Pattern Recognition*, vol. 116, p. 107932, 2021.

[154] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[155] S. Ding, H. Zhu, W. Jia, and C. Su, "A survey on feature extraction for pattern recognition," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 169–180, 2012.

[156] D. Effrosynidis and A. Arampatzis, "An evaluation of feature selection methods for environmental data," *Ecological Informatics*, vol. 61, p. 101224, 2021.

[157] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[158] D. Dua, C. Graff *et al.*, "Uci machine learning repository," 2017.

[159] P. F. Silva, A. R. Marcal, and R. M. A. da Silva, "Evaluation of features for leaf discrimination," in *International Conference Image Analysis and Recognition*. Springer, 2013, pp. 197–204.

[160] J. Xu, "A weighted linear discriminant analysis framework for multi-label feature extraction," *Neurocomputing*, vol. 275, pp. 107–120, 2018.

[161] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne)," *Computer Science Review*, vol. 40, p. 100378, 2021.

[162] F.-P. Schilling and T. Stadelmann, *Artificial Neural Networks in Pattern Recognition: 9th IAPR TC3 Workshop, ANNPR 2020, Winterthur, Switzerland, September 2-4, 2020, Proceedings*. Springer Nature, 2020, vol. 12294.

[163] P. S. Sajja, "Examples and applications on artificial neural networks," in *Illustrated Computational Intelligence*. Springer, 2021, pp. 101–153.

[164] M. Al-Kasassbeh, S. Mohammed, M. Alauthman, and A. Almomani, "Feature selection using a machine learning to classify a malware," in *Handbook of computer networks and cyber security*. Springer, 2020, pp. 889–904.

[165] A. M. Jade, B. Srikanth, V. K. Jayaraman, B. Kulkarni, J. Jog, and L. Priya, "Feature extraction and denoising using kernel pca," *Chemical Engineering Science*, vol. 58, no. 19, pp. 4441–4448, 2003.

[166] K. Prabha, "Disease sniffing robots to apps fixing plant diseases: applications of artificial intelligence in plant pathology—a mini review," *Indian Phytopathology*, vol. 74, no. 1, pp. 13–20, 2021.

[167] H. Deng and G. Runger, "Gene selection with guided regularized random forest," *Pattern Recognition*, vol. 46, no. 12, pp. 3483–3489, 2013.

[168] A. M. Kowshalya, R. Madhumathi, and N. Gopika, "Correlation based feature selection algorithms for varying datasets of different dimensionality," *Wireless Personal Communications*, vol. 108, no. 3, pp. 1977–1993, 2019.

[169] B. F. Darst, K. C. Malecki, and C. D. Engelman, "Using recursive feature elimination in random forest to account for correlated variables in high dimensional data," *BMC genetics*, vol. 19, no. 1, pp. 1–6, 2018.

[170] M. Rakhra, P. Soniya, D. Tanwar, P. Singh, D. Bordoloi, P. Agarwal, S. Takkar, K. Jairath, and N. Verma, "Crop price prediction using random forest and decision tree regression:-a review," *Materials Today: Proceedings*, 2021.

[171] D. Perez-Alonso, S. Peña-Tejedor, M. Navarro, C. Rad, Á. Arnaiz-González, and J.-F. Díez-Pastor, "Decision trees for the prediction of environmental and agronomic effects of the use of compost of sewage slugde (css)," *Sustainable Production and Consumption*, vol. 12, pp. 119–133, 2017.

[172] T. Elomaa* and T. Malinen, "On look-ahead and pathology in decision tree learning," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 1-2, pp. 19–33, 2005.

[173] T. E. Franz, S. Pokal, J. P. Gibson, Y. Zhou, H. Gholizadeh, F. A. Tenorio, D. Rudnick, D. Heeren, M. McCabe, M. Ziliani *et al.*, "The role of topography, soil, and remotely sensed vegetation condition towards predicting crop yield," *Field Crops Research*, vol. 252, p. 107788, 2020.

[174] S. Srinivasagopalan, J. Barry, V. Gurupur, and S. Thankachan, "A deep learning approach for diagnosing schizophrenic patients," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 6, pp. 803–816, 2019.

[175] J. Langhammer and J. Česák, "Applicability of a nu-support vector regression model for the completion of missing data in hydrological time series," *Water*, vol. 8, no. 12, p. 560, 2016.

[176] J. Cai, K. Xu, Y. Zhu, F. Hu, and L. Li, "Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest," *Applied energy*, vol. 262, p. 114566, 2020.

[177] X. Luo, X. Chang, and X. Ban, "Regression and classification using extreme learning machine based on l1-norm and l2-norm," *Neurocomputing*, vol. 174, pp. 179–186, 2016.

[178] D. Devi, S. K. Biswas, and B. Purkayastha, "Correlation-based oversampling aided cost sensitive ensemble learning technique for treatment of class imbalance," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–33, 2021.

[179] D. G. Mayer, K. A. Chandra, and J. R. Burnett, "Improved crop forecasts for the australian macadamia industry from ensemble models," *Agricultural Systems*, vol. 173, pp. 519–523, 2019.

[180] N. Karasekreter, F. Başçiftçi, and U. Fidan, "A new suggestion for an irrigation schedule with an artificial neural network," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 1, pp. 93–104, 2013.

[181] S. S. Panda, D. P. Ames, and S. Panigrahi, "Application of vegetation indices for agricultural crop yield prediction using neural network techniques," *Remote Sensing*, vol. 2, no. 3, pp. 673–696, 2010.

[182] A. Khoshroo, A. Emrouznejad, A. Ghaffarizadeh, M. Kasraei, and M. Omid, "Sensitivity analysis of energy inputs in crop production using artificial neural networks," *Journal of cleaner production*, vol. 197, pp. 992–998, 2018.

[183] P. Taherei Ghazvinei, H. Hassanpour Darvishi, A. Mosavi, K. b. W. Yusof, M. Alizamir, S. Shamshirband, and K.-w. Chau, "Sugarcane growth prediction based on meteorological parameters using extreme learning machine and artificial neural network," *Engineering Applications of Computational Fluid Mechanics*, vol. 12, no. 1, pp. 738–749, 2018.

[184] I. Ahmad, U. Saeed, M. Fahad, A. Ullah, M. Habib ur Rahman, A. Ahmad, and J. Judge, "Yield forecasting of spring maize using remote sensing and crop modeling in faisalabad-punjab pakistan," *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 10, pp. 1701–1711, 2018.

[185] P. Filippi, E. J. Jones, N. S. Wimalathunge, P. D. Somarathna, L. E. Pozza, S. U. Ugbaje, T. G. Jephcott, S. E. Paterson, B. M. Whelan, and T. F. Bishop, "An approach to forecast grain crop yield using multi-layered, multi-farm data sets and machine learning," *Precision Agriculture*, vol. 20, no. 5, pp. 1015–1029, 2019.

[186] A. Shah, A. Dubey, V. Hemnani, D. Gala, and D. Kalbande, "Smart farming system: Crop yield prediction using regression techniques," in *Proceedings of international conference on wireless communication*. Springer, 2018, pp. 49–56.

[187] H. Cheng, L. Damerow, Y. Sun, and M. Blanke, "Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks," *Journal of Imaging*, vol. 3, no. 1, p. 6, 2017.

[188] J. S. Visockiene, E. Tumeliene, and V. Maliene, "Analysis and identification of abandoned agricultural land using remote sensing methodology," *Land Use Policy*, vol. 82, pp. 709–715, 2019.

[189] S. Kumar, M. Pant, M. Kumar, and A. Dutt, "Colour image segmentation with histogram and homogeneity histogram difference using evolutionary algorithms," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 1, pp. 163–183, 2018.

[190] A. K. Bhandari and I. V. Kumar, "A context sensitive energy thresholding based 3d otsu function for image segmentation using human learning optimization," *Applied Soft Computing*, vol. 82, p. 105570, 2019.

[191] S. Pare, A. Kumar, G. Singh, and V. Bajaj, "Image segmentation using multilevel thresholding: a research review," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 1, pp. 1–29, 2020.

[192] S. K. Jha, P. Bannerjee, and S. Banik, "Random walks based image segmentation using color space graphs," *Procedia Technology*, vol. 10, pp. 271–278, 2013.

[193] M. Á. Castillo-Martínez, F. J. Gallegos-Funes, B. E. Carvajal-Gámez, G. Urriolagoitia-Sosa, and A. J. Rosales-Silva, "Color index based thresholding method for background and foreground segmentation of plant images," *Computers and Electronics in Agriculture*, vol. 178, p. 105783, 2020.

[194] S. Kotte, P. R. Kumar, and S. K. Injeti, "An efficient approach for optimal multilevel thresholding selection for gray scale images based on improved differential search algorithm," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 1043–1067, 2018.

[195] A. Jose, S. Ravi, and M. Sambath, "Brain tumor segmentation using k-means clustering and fuzzy c-means algorithms and its area calculation," *International*

Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 3, 2014.

[196] H. Zhang and Q. Peng, "Pso and k-means-based semantic segmentation toward agricultural products," *Future Generation Computer Systems*, vol. 126, pp. 82–87, 2022.

[197] K. Tian, J. Li, J. Zeng, A. Evans, and L. Zhang, "Segmentation of tomato leaf images based on adaptive clustering number of k-means algorithm," *Computers and Electronics in Agriculture*, vol. 165, p. 104962, 2019.

[198] R. Singh, A. Goel, and D. Raghuvanshi, "Computer-aided diagnostic network for brain tumor classification employing modulated gabor filter banks," *The Visual Computer*, pp. 1–15, 2020.

[199] T. Vijayan, M. Sangeetha, A. Kumaravel, and B. Karthik, "Gabor filter and machine learning based diabetic retinopathy analysis and detection," *Microprocessors and Microsystems*, p. 103353, 2020.

[200] S. S. More, B. Narain, and B. Jadhav, "Role of modified gabor filter algorithm in multimodal biometric images," in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2019, pp. 1–5.

[201] D. Chowdhury, S. Mishra, S. Paul, S. Bagchi, H. Ahmed, T. Das, and C. Mandal, "Robust and atomic technique for edge detection to identify edges present in digital image using dscths method," in *Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020*. Springer Singapore, 2021, pp. 219–227.

[202] B. K. Shah, V. Kedia, R. Raut, S. Ansari, and A. Shroff, "Evaluation and comparative study of edge detection techniques," *IOSR Journal of Computer Engineering*, vol. 22, no. 5, pp. 6–15, 2020.

[203] D. Sharifrazi, R. Alizadehsani, M. Roshanzamir, J. H. Joloudari, A. Shoeibi, M. Jafari, S. Hussain, Z. A. Sani, F. Hasanzadeh, F. Khozeimeh *et al.*, "Fusion of convolution neural network, support vector machine and sobel filter for accurate

detection of covid-19 patients using x-ray images," *Biomedical Signal Processing and Control*, vol. 68, p. 102622, 2021.

[204] T. Sanida, A. Sideris, and M. Dasygenis, "A heterogeneous implementation of the sobel edge detection filter using opencl," in *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2020, pp. 1–4.

[205] R. Garg, A. Kumar, M. Prateek, K. Pandey, and S. Kumar, "Land cover classification of spaceborne multifrequency sar and optical multispectral data using machine learning," *Advances in Space Research*, 2021.

[206] D. K. Matci and U. Avdan, "Optimization-based automated unsupervised classification method: A novel approach," *Expert Systems with Applications*, vol. 160, p. 113735, 2020.

[207] C. Orhei, V. Bogdan, C. Bonchis, and R. Vasiu, "Dilated filters for edge-detection algorithms," *Applied Sciences*, vol. 11, no. 22, p. 10716, 2021.

[208] P. Topno and G. Murmu, "An improved edge detection method based on median filter," in *2019 Devices for Integrated Circuit (DevIC)*. IEEE, 2019, pp. 378–381.

[209] S. Das, D. Roy, and P. Das, "Disease feature extraction and disease detection from paddy crops using image processing and deep learning technique," *Computational Intelligence in Pattern Recognition. Advances in Intelligent Systems and Computing*, vol. 1120, pp. 443–449, 2020.

[210] T. Yun, K. Jiang, G. Li, M. P. Eichhorn, J. Fan, F. Liu, B. Chen, F. An, and L. Cao, "Individual tree crown segmentation from airborne lidar data using a novel gaussian filter and energy function minimization-based approach," *Remote Sensing of Environment*, vol. 256, p. 112307, 2021.

[211] A. Sarkar, M. Maniruzzaman, M. S. Ahsan, M. Ahmad, M. I. Kadir, and S. T. Islam, "Identification and classification of brain tumor from mri with feature extraction by support vector machine," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–4.

[212] S. I. T. Joseph, J. Sasikala, and D. S. Juliet, "Detection of ship from satellite images using deep convolutional neural networks with improved median filter,"

in *Artificial Intelligence Techniques for Satellite Image Analysis*. Springer, 2020, pp. 69–82.

[213] A. Bihani, H. Daigle, J. E. Santos, C. Landry, M. Prodanovic, and K. Milliken, "Mudrocknet: Semantic segmentation of mudrock sem images through deep learning," *arXiv preprint arXiv:2102.03393*, 2021.

[214] T. Meraj, A. Hassan, S. Zahoor, H. T. Rauf, M. I. Lali, L. Ali, and S. A. C. Bukhari, "Lungs nodule detection using semantic segmentation and classification with optimal features," 2019.

[215] A. Abdalla, H. Cen, L. Wan, R. Rashid, H. Weng, W. Zhou, and Y. He, "Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure," *Computers and Electronics in Agriculture*, vol. 167, p. 105091, 2019.

[216] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, p. 105527, 2020.

[217] W. Boulila, "A top-down approach for semantic segmentation of big remote sensing images," *Earth Science Informatics*, vol. 12, no. 3, pp. 295–306, 2019.

[218] M. Maity, A. Jaiswal, K. Gantait, J. Chatterjee, and A. Mukherjee, "Quantification of malaria parasitaemia using trainable semantic segmentation and capsnet," *Pattern Recognition Letters*, vol. 138, pp. 88–94, 2020.

[219] J. Balado, J. Martínez-Sánchez, P. Arias, and A. Novo, "Road environment semantic segmentation with deep learning from mls point cloud data," *Sensors*, vol. 19, no. 16, p. 3466, 2019.

[220] Y. Sun, X. Zhang, Q. Xin, and J. Huang, "Developing a multi-filter convolutional neural network for semantic segmentation using high-resolution aerial imagery and lidar data," *ISPRS journal of photogrammetry and remote sensing*, vol. 143, pp. 3–14, 2018.

BRIEF PROFILE OF THE RESEARCH SCHOLAR

Surabhi Lingwal did her B.Tech (Computer Science & Engineering) in 2011 from Govind Ballabh Pant Engineering College, Uttarakhand and M.Tech (Computer Science & Engineering) from Govind Ballabh Pant Engineering College, Uttarakhand, in 2013. Presently, she is serving as an Assistant Professor in the Department of Computer Science & Engineering at Govind Ballabh Pant Institute of Engineering & Technology, Pauri Garhwal, Uttarakhand. She has a total of 7 years of teaching experience. Her research area includes Artificial Intelligence, Machine Learning, Deep Learning and Computer Vision. She has published several papers in various reputed international and national journals.

LIST OF PUBLICATIONS

List of Published Papers

Sl. No.	Title of Paper	Name of Journal where published	DOI No.	Volume & Issue	Year	Pages
1	"Image-based wheat grain classification using convolutional neural network"	Multimedia Tools and Applications, Springer, **SCI-indexed**	https://doi.org /10.1007/s11042 -020-10174-3	vol. 80	2021	pages 1-25
2	"Info_PCA: A Hybrid Technique to Improve Accuracy by Dimensionality Reduction"	IT in Industry, Science Research Society (SRS), **ESCI-indexed**	https://doi.org /10.17762/ itii.v9i2.370	vol. 9	2021	pages 1-9
3	"A Novel Machine Learning Approach for Rice Yield Estimation"	Journal of Experimental & Theoretical Artificial Intelligence, Taylor & Francis, **SCI-indexed**	https://doi.org/ /10.1080/ 0952813X.2022 .2062458	–	2022	pages 1-20
4	"Semantic segmentation of landcover for cropland mapping and area estimation using Machine Learning techniques"	Data Intelligence, MIT Press, **ESCI& Scopus -indexed**	https://doi.org /10.1162/ dint_a_00145	vol.5(2)	2022	pages 1-18

List of Conference Papers

Sl. No.	Title of Paper	Name of Conference	Place	Year
1	"Comparative Analysis of Machine Learning Techniques for Plant Disease"	IEEE & Springer International Virtual Conference on Integrated Intelligent Enable Networks & Computing(IIENC-2020)	Uttarakhand	2020
2	"Deep Convolutional Neural Network approach for Tomato leaf disease Classification"	Springer 3^{rd} International conference on Machine Learning, Image Processing, Network security and Data Science (MIND-2021)	NIT Raipur	2021

List of Communicated Papers

Sl. No.	Title of Paper	Name of Journal	Present Status	Year
1	"Thresholding-based image semantic segmentation for agriculture cropland mapping"	Sensing and Imaging **Scopus-indexed**	Under Review	2022