

DETECTION OF INFECTED MAIZE LEAF USING DEEP LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication Engineering

By

BENISHA J (Reg. No. 40130024)

SUGANYA V (Reg. No. 40130228)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A++" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119**

APRIL 2024



SATHYABAMA

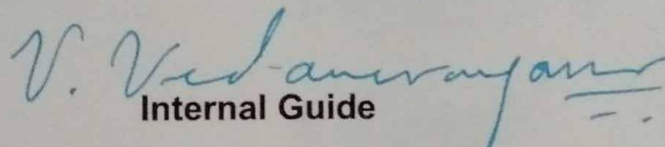
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

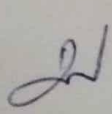
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BENISHA J (40130024), SUGANYA V (40130228)** who have done the project work as a team who carried out the project entitled "**DETECTION OF INFECTED MAIZE LEAF USING DEEP LEARNING**" under my supervision from November 2023 to April 2024.


Internal Guide

Dr. V. VEDANARAYANAN, M.E., Ph.D.,


Head of the Department
Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on 18.04.2024

Internal Examiner

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

External Examiner

DECLARATION

We **BENISHA J** (Reg.No-40130024), **SUGANYA V** (Reg.No- 40130228) hereby declare that the Project Report entitled "**DETECTION OF INFECTED MAIZE LEAF USING DEEP LEARNING**" done by us under the guidance of **Dr.V.VEDANARAYANAN.M.E.,Ph.D.**, Dept. of Electronics and Communication Engineering, SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, CHENNAI is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 18.04.2024

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

1. Benisha J
2. Suganya V.

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Dean, School of Electronics and Communication Engineering** and **Dr. T .RAVI, M.E.,Ph.D., Head of the Department, Dept. of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **DR.V.VEDANARAYANAN.M.E.,Ph.D., Dept. of Electronics and Communication Engineering** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project

TABLE OF CONTENTS

CHAPTER .NO	TITLE	PAGE NO.
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	viii
	ABSTRACT	1
1.	INTRODUCTION	2
	1.1 MAIZE LEAF DISEASE	2
	1.2 MACHINE LEARNING	2
	1.3 LEAF DISEASE	3
	1.4 FUNGAL PATHOGENS	4
	1.5 PLANT HEALTH	4
	1.6 OBJECTIVES	5
2	LITERATURE SURVEY	6-18
3	SYSTEM ANALYSIS	19
	3.1 EXISTING SYSTEM	19
	3.1.1 DRAWBACKS	19
	3.2 PROPOSED SYSTEM	20
	3.2.1 ADVANTAGES	20
	3.3 FEASIBILITY STUDY	21
	3.3.1 TECHNICAL FEASIBILITY	21
	3.3.2 OPERATIONAL FEASIBILITY	22
	3.3.3 ECONOMICAL FEASIBILITY	22
4	SYSTEM SPECIFICATION	23
	4.1 HARDWARE REQUIREMENTS	23
	4.2 SOFTWARE SPECIFICATION	23
5	SOFTWARE DESCRIPTION	24
	5.1 FEATURES OF SOFTWARE	24-28
6	PROJECT DESCRIPTION	29
	6.1 PROBLEM DEFINITION	29

	6.2 MODULE DESCRIPTION	29
	6.2.1 LOAD DATA	29
	6.2.2 DATA PREPROCESSING	29
	6.2.3 FUTURE SELECTION	29
	6.2.4 TRAINING AND TESTING	30
	6.2.5 EVALUATION AND PERFORMANCE	30
	6.3 SYSTEM FLOW DIAGRAM	31
	6.4 INPUT DESIGN	32
	6.5 OUTPUT DESIGN	32
7	SYSTEM TESTING AND IMPLEMENTATION	33
	7.1 SYSTEM TESTING	33
	7.2 SYSTEM IMPLEMENTATION	34
8	SYSTEM MAINTENANCE	35
	8.1 CORRECTIVE MAINTENANCE	36
	8.2 ADAPTIVE MAINTENANCE	36
	8.3 PERFECTIVE MAINTENANCE	36
	8.4 PREVENTIVE MAINTENANCE	37
9	CONCLUSION AND FUTURE ENHANCEMENT	38
10	APPENDICES	39
	10.1 SCREEN SHOTS	39-40
11	REFERENCES	41

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1	MACHINE LEARNING	3
2	INPUT IMAGE 1	39
3	OUTPUT IMAGE 1	39
4	INPUT IMAGE 2	40
5	OUTPUT IMAGE 2	40

LIST OF ABBREVIATIONS

ABBREVIATIONS

CNN	CONVOLUTIONAL NEURAL NETWORK
RNN	RE-CURRENT NEURAL NETWORK
AP	AVERAGE PRECISION
DN	DE CONVOLUTIONAL NETWORK
WRN	WIDE RESIDUAL NETWORK
DFF-ResNet	DEEP FEATURE FUSION RESIDUAL NETWORK
SGD	STOCHASTIC GRADIENT DESCENT
BIM	BASIC IMAGE MANIPULATION
NST	NEURAL STYLE TRANSFER
MGPUs	MULTI-GRAPHICS PROCESSING UNITS
DCNN	DEEP CONVOLUTIONAL NEURAL NETWORK
SVM	SUPPORT VECTOR MACHINE
GPU	GRAPHICS PROCESS UNIT
RGB	RED GREEN BLUE

ABSTRACT

Maize (*Zea mays*), also known as corn in some regions, is one of the most important cereal crops globally, serving as a staple food source for millions of people and a key ingredient in livestock feed and industrial processes. However, the successful cultivation of maize is often challenged by various factors, including pests, environmental stress, and diseases. Among these challenges, maize leaf diseases play a significant role in reducing crop yield and quality, ultimately affecting food security and economic stability in many agricultural communities. We proposed Multi-support vector machine (MSVM) is a machine learning algorithm that can be used to detect maize leaf diseases with high accuracy. MSVM works by finding a hyperplane that separates the data into different categories in a way that maximizes the margin between the data points and the hyperplane. To detect maize leaf diseases using MSVM, a dataset of maize leaf images is first collected and pre-processed. The pre-processed images are then used to train the MSVM model. The trained model can then be used to classify new maize leaf images as either healthy or diseased.

CHAPTER 1

1. INTRODUCTION

1.1 MAIZE LEAF DISEASE

Maize, also known as corn, is one of the world's most vital staple crops, providing sustenance and livelihoods for millions of people. However, like all cultivated plants, maize is susceptible to various diseases that can significantly impact its yield and overall health. Maize leaf diseases are a common concern for farmers and agricultural experts, as they can lead to reduced crop productivity and economic losses. These diseases manifest as various symptoms on the leaves, ranging from discoloration and lesions to wilting and deformities, and they are typically caused by fungal, bacterial, or viral pathogens. Understanding and effectively managing these maize leaf diseases is essential for ensuring food security and sustaining agricultural communities worldwide. In this introduction, we will explore some of the most prevalent maize leaf diseases, their causes, and the importance of proper management strategies to protect maize crops.

1.2 MACHINE LEARNING

Machine learning has emerged as a transformative field of artificial intelligence, revolutionizing the way computers learn and make decisions from data without explicit programming. It empowers systems to recognize patterns, make predictions, and continuously improve their performance through experience. This technology has found applications in various domains, including image and speech recognition, natural language processing, recommendation systems, and much more. At its core, machine learning enables computers to learn from large datasets and extract meaningful insights, allowing them to generalize and perform tasks beyond their explicitly programmed instructions. The process of learning involves adjusting model parameters based on feedback from the data, iteratively refining predictions and improving accuracy over time. Machine learning algorithms are diverse, ranging from classic approaches like linear regression and decision trees to sophisticated deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The advent of deep

learning has significantly advanced the state-of-the-art in numerous applications, especially those involving complex data types like images, audio, and text. The rapid progress and widespread adoption of machine learning have been facilitated by the availability of extensive datasets, powerful computing resources, and open-source libraries and frameworks like Tensor Flow, PyTorch, and scikit-learn.

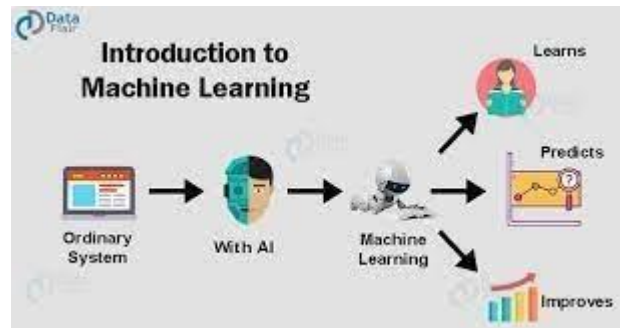


Figure 1 MACHINE LEARNING

1.3 LEAF DISEASE

Leaf diseases are a significant concern in agriculture, impacting a wide range of plant species and posing a substantial threat to crop yields and food security. These diseases, caused by various pathogens such as fungi, bacteria, and viruses, manifest as a variety of symptoms on plant leaves, including discoloration, lesions, and wilting. As leaves play a crucial role in photosynthesis and overall plant health, any affliction that affects them can have a profound effect on the plant's ability to grow and produce. In this introduction, we will delve into the world of leaf diseases, exploring their causes, their impact on plants and agriculture, and the strategies employed to manage and mitigate these detrimental effects. Understanding leaf diseases is essential for farmers and agricultural experts in their quest to protect crops and ensure the sustainable production of food.

1.4 FUNGAL PATHOGENS

Fungal pathogens are a group of microorganisms that play a significant role in the world of plant diseases, posing a considerable threat to agriculture and natural ecosystems. These fungal organisms are responsible for a wide array of plant diseases that can devastate crops, forests, and ornamental plants. Fungal pathogens can infect various plant parts, including leaves, stems, and roots, leading to symptoms such as wilting, discoloration, necrosis, and reduced growth. Understanding the nature of fungal pathogens, their life cycles, and the mechanisms by which they cause diseases is crucial for effective disease management and prevention. In this introduction, we will explore the fascinating and often complex world of fungal pathogens, shedding light on their impact on plant health and the strategies employed to combat these destructive microorganisms.

1.5 PLANT HEALTH

Plant health is a fundamental aspect of agriculture, ecology, and our planet's overall well-being. It encompasses the physical and physiological well-being of plants, ensuring their ability to thrive, reproduce, and contribute to ecosystems and human needs. Healthy plants are not only essential for food production but also play a vital role in maintaining environmental balance. They contribute to air and water purification, provide habitats for wildlife, and serve as a source of inspiration for medicine and industry. In this introduction, we will delve into the significance of plant health, its importance for sustaining life on Earth, and the factors that can threaten it, including pests, diseases, environmental stressors, and human activities. Understanding and promoting plant health is not only crucial for agriculture but also for preserving the biodiversity and ecological stability of our planet.

1.6 OBJECTIVES

- Develop a computer-aided diagnosis system for maize leaf diseases that can accurately identify and classify diseases early on.
- Help farmers to identify and treat diseases early on, before they cause significant damage to crops.
- Reduce the use of pesticides and other chemicals, which can be harmful to the environment and human health.
- Improve the quality and yield of maize crops

CHAPTER 2

2. LITERATURE REVIEW

2.1 DETECTION OF PLANT LEAF DISEASES USING IMAGE SEGMENTATION AND SOFT COMPUTING TECHNIQUES

Vijai Singhet.al. Has proposed in this paper, Agricultural productivity is something on which economy highly depends. This is the one of the reasons that disease detection in plants plays an important role in agriculture field, as having disease in plants are quite natural. If proper care is not taken in this area then it causes serious effects on plants and due to which respective product quality, quantity or productivity is affected. For instance a disease named little leaf disease is a hazardous disease found in pine trees in United States. Detection of plant disease through some automatic technique is beneficial as it reduces a large work of monitoring in big farms of crops, and at very early stage itself it detects the symptoms of diseases i.e. when they appear on plant leaves. This paper presents an algorithm for image segmentation technique which is used for automatic detection and classification of plant leaf diseases. It also covers survey on different diseases classification techniques that can be used for plant leaf disease detection. Image segmentation, which is an important aspect for disease detection in plant leaf disease, is done by using genetic algorithm. The agricultural land mass is more than just being a feeding sourcing in today's world. Indian economy is highly dependent of agricultural productivity. Therefore in field of agriculture, detection of disease in plants plays an important role. To detect a plant disease in very initial stage, use of automatic disease detection technique is beneficial. For instance a disease named little leaf disease is a hazardous disease found in pine trees in United States. The affected tree has a stunted growth and dies within 6 years. Its impact is found in Alabama, Georgia parts of Southern US. In such scenarios early detection could have been fruitful. The existing method for plant disease detection is simply naked eye observation by experts through which identification and detection of plant diseases is done. For doing so, a large team of experts as well as continuous monitoring of plant is required, which costs very high when we do with large farms. At the same time, in some countries, farmers do not have proper

facilities or even idea that they can contact to experts. Due to which consulting experts even cost high as well as time consuming too. In such conditions, the suggested technique proves to be beneficial in monitoring large fields of crops. Automatic detection of the diseases by just seeing the symptoms on the plant leaves makes it easier as well as cheaper. This also supports machine vision to provide image based automatic process control, inspection, and robot guidance. This paper presents the survey on different diseases classification techniques used for plant leaf disease detection and an algorithm for image segmentation technique that can be used for automatic detection as well as classification of plant leaf diseases later. Banana, beans, jackfruit, lemon, mango, potato, tomato, and spot are some of those ten species on which proposed algorithm is tested. Therefore, related diseases for these plants were taken for identification. With very less computational efforts the optimum results were obtained, which also shows the efficiency of proposed algorithm in recognition and classification of the leaf diseases. Another advantage of using this method is that the plant diseases can be identified at early stage or the initial stage. To improve recognition rate in classification process Artificial Neural Network, Bayes classifier, Fuzzy Logic and hybrid algorithms can also be used.

2.2 HOW DEEP LEARNING EXTRACTS AND LEARNS LEAF FEATURES FOR PLANT CLASSIFICATION

Sue Han Lee et al. has proposed in this paper Plant identification systems developed by computer vision researchers have helped botanists to recognize and identify unknown plant species more rapidly. Hitherto, numerous studies have focused on procedures or algorithms that maximize the use of leaf databases for plant predictive modeling, but this results in leaf features which are liable to change with different leaf data and feature extraction techniques. In this paper, we learn useful leaf features directly from the raw representations of input data using Convolutional Neural Networks (CNN), and gain intuition of the chosen features based on a De convolutional Network (DN) approach. We report somewhat unexpected results different orders of venation are the best representative features compared to those of outline shape, and we observe multi-level representation in leaf data, demonstrating the hierarchical transformation of features from lower-level to higher-level abstraction, corresponding to species classes. We show that

these findings fit with the hierarchical botanical definitions of leaf characters. Through these findings, we gained insights into the design of new hybrid feature extraction models which are able to further improve the discriminative power of plant classification systems. Computational botany consists of applying innovative computational methods to help progress on an age-old problem, i.e. the identification of the estimated 400,000 species of plants on Earth. This interdisciplinary approach combines botanical data and species concepts with computational solutions for classification of plants or parts thereof and focuses on the design of novel recognition methods. These are modelled using botanical data, but are extendable to other large repositories and application domains. Plant species identification is a subject of great importance in many fields of human endeavor, including such areas as agronomy, conservation, environmental impact, natural product and drug discovery and other applied areas. This paper investigated the use of deep learning to harvest discriminatory features from leaf images by learning, and apply them as classifiers for plant identification. Our experimental results demonstrate that learning the features using CNNs can provide better feature representations of leaf images as compared to using hand-crafted features. We also quantified the features that most efficiently represent the leaves for the purpose of species identification, using a DN approach. In the first experiment we show that venation structure is a very important feature for identification especially when shape feature alone is inadequate.

2.3 DETECTION AND COUNTING OF FLOWERS ON APPLE TREES FOR BETTER CHEMICAL THINNING DECISIONS

Guy Farjon et al. has proposed in this system, Accurate chemical thinning of apple trees requires estimation of their blooming intensity, and determination of the blooming peak date. Performing this task, as of today, requires human experts to be present in the orchards for the entire blossom period or extrapolate using a single observation. Since experts are rare and in high demand, there is a need to automate this process. The system presented in this paper is able to estimate the blooming intensity and the blooming peak date from a sequence of tree images, with close-to-human accuracy. For this purpose, a two years dataset was collected in 2014–2015, partially tagged for the flowers location and completely annotated for blooming intensity. Using this dataset, an algorithm

was developed and trained with three stages: a visual flower detector based on a deep convolutional neural network, followed by a blooming level estimator, and a peak blooming day finding algorithm. Despite the challenging conditions, the trained detector was able to detect flowers on trees with an Average Precision (AP) score of 0.68, which is on a par with contemporary results of other objects in detection benchmarks. The blooming estimator was based on a linear regression component, which used the number of flowers detected and related statistics to estimate the blooming intensity. The Pearson correlation between the algorithm blooming estimation and human judgments of several experts indicated high agreement levels (0.78–0.93) which were similar to the correlations measured among the human experts. Moreover, the developed estimator was relatively stable across multiple years. The developed peak date finding algorithm identified correctly the orchard's blooming peak date, which was used to determine the thinning date in the current practice (the entire orchard is thinned in the same day). Experiments testing the algorithm's ability to find a blooming peak date for each tree independently showed encouraging results, which may lead upon refinement to a more precise practice for tree-specific thinning. The agricultural sector faces an immense challenge—the world population is exponentially growing, and with it, the demand for agricultural yields. The need to satisfy this demand led to significant progress in precision agriculture applications, using advanced technologies including computer vision, satellite navigation systems, remote sensing, geographic information systems, and many others. This work faces the challenge of estimating blooming intensity of apple trees, a part of the chemical thinning process, using advanced computer vision tools. In agricultural sciences, thinning refers to the removal of a fraction of the trees' flowers or fruits, to improve the growth of others. The link between thinning and fruits quality has been well established in the literature. During their blossom period, apple trees produce many flowers, which later transform into apples. Each such flower requires resource allocation from the tree in order to grow properly and turn into an apple. However, when the number of flowers is too high there is an abundance of apples, which hence grow to be small, with insufficient quality and may be unworthy for sale. When this number is too low, the trees produce a small amount of apples, which are worthy for sale but with a low revenue per acre. In order to

get the optimal number of apples, which depends on cultivar, growing site and market price structure, trees must undergo a thinning process.

2.4 USING DEEP LEARNING FOR IMAGE-BASED PLANT DISEASE DETECTION

Sharada Prasanna Mohanty et al. has proposed in this system Crop diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. The combination of increasing global smartphone penetration and recent advances in computer vision made possible by deep learning has paved the way for smartphone-assisted disease diagnosis. Using a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions, we train a deep convolutional neural network to identify 14 crop species and 26 diseases (or absence thereof). The trained model achieves an accuracy of 99.35% on a held-out test set, demonstrating the feasibility of this approach. When testing the model on a set of images collected from trusted online sources - i.e. taken under conditions different from the images used for training - the model still achieves an accuracy of 31.4%. While this accuracy is much higher than the one based on random selection (2.6%), a more diverse set of training data is needed to improve the general accuracy. Overall, the approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path towards smartphone-assisted crop disease diagnosis on a massive global scale. Modern technologies have given human society the ability to produce enough food to meet the demand of more than 7 billion people. However, food security remains threatened by a number of factors including climate change, the decline in pollinators, plant diseases, and others. Plant diseases are not only a threat to food security at the global scale, but can also have disastrous consequences for small holder farmers whose livelihoods depend on healthy crops. In the developing world, more than 80 percent of the agricultural production is generated by smallholder farmers, and reports of yield loss of more than 50% due to pests and diseases are common. Furthermore, the largest fraction of hungry people (50%) live in smallholder farming households, making smallholder farmers a group that's particularly vulnerable to pathogen-derived disruptions in food supply. Smartphones in particular To address the issue of over-fitting, we vary the test set to train set ratio and observe that even in the

extreme case of training on only 20% of the data and testing the trained model on the rest 80% of the data, the model achieves an overall accuracy of 98.21% (mean F1-Score of 0.9820) in the case of Google Net::Transfer Learning::Color::20-80. As expected, the overall performance of both Alex Net and Google Net do degrade if we keep increasing the test set to train set ratio, but the decrease in performance is not as drastic as we would expect if the model was indeed over-fitting. also shows that there is no divergence between the validation loss and the training loss, confirming that over fitting is not a contributor to the results we obtain across all our experiments.

2.5 DFF-RESNET: AN INSECT PEST RECOGNITION MODEL BASED ON RESIDUAL NETWORKS

Wenjie Liu et al. has proposed in this paper : Insect pest control is considered as a significant factor in the yield of commercial crops. Thus, to avoid economic losses, we need a valid method for insect pest recognition. In this paper, we proposed a feature fusion residual block to perform the insect pest recognition task. Based on the original residual block, we fused the feature from a previous layer between two convolution layers in a residual signal branch to improve the capacity of the block. Furthermore, we explored the contribution of each residual group to the model performance. We found that adding the residual blocks of earlier residual groups promotes the model performance significantly, which improves the capacity of generalization of the model. By stacking the feature fusion residual block, we constructed the Deep Feature Fusion Residual Network (DFF-ResNet). To prove the validity and adaptivity of our approach, we constructed it with two common residual networks (Pre-ResNet and Wide Residual Network (WRN)) and validated these models on the Canadian Institute For Advanced Research (CIFAR) and Street View House Number (SVHN) benchmark datasets. The experimental results indicate that our models have a lower test error than those of baseline models. Then, we applied our models to recognize insect pests and obtained validity on the IP102 benchmark dataset. The experimental results show that our models outperform the original ResNet and other state-of-the-art methods. Insect pest control has always been a crucial problem for commercially important crops. Early detection of the insect pest species helps to decrease the damage they cause, which is significant for a stable

agricultural economy and food security. In conventional recognition approaches, early detection requires many trained experts to recognize insect pests which is expensive and less efficient. Therefore, we need a valid detection method to accelerate the process of insect pest recognition. Deep learning technology, as an emerging hot technology, has increasingly attracted attention, and many advanced methods depending on this technology have been applied in every field including insect pest recognition. Deep learning has gained considerable attention in various domains, e.g., computer vision, natural language processing, emotion computing, etc. Especially for computer vision, many excellent approaches have been proposed to daily problems and achieved a state-of-the-art performance. Residual Networks (ResNets) made breakthroughs in several visual tasks. In our work, to improve the model performance, the DFF-ResNet is proposed. The central idea of our model focuses on the fusion feature from previous layers and making the model become deeper than Pre ResNet. To further improve the model performance, we explored the influence of the number of residual blocks of earlier residual groups and found that it could benefit our models. To verify the validity and adaptively of our approach, we applied it to different residual networks and evaluated the model performance on the CIFAR and SVHN benchmark datasets with different widths and depths. The experimental results indicated that our models achieved a better performance than baseline models. Moreover, for high-resolution image classification tasks, we applied our model to recognize insect pests and evaluated the IP102 benchmark dataset. As the testing performance showed, our models achieve a better test accuracy performance than ResNet model and other state-of-the-art approaches. Thus, based on above studies. It demonstrates the validity and adaptively of our approach, which is convenient for embedding into other common residual networks.

2.6 IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Ilya Sutskever et al. has proposed in this paper, We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-

5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way soft max. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small — on the order of tens of thousands of images, and . Simple recognition tasks can be solved quite well with datasets of this size, especially if they are augmented with label-preserving transformations. For example, the current best error rate on the MNIST digit-recognition task to learn about thousands of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don’t have. Convolutional neural networks (CNNs) constitute one such class of models . Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse. Our results show that a large, deep convolutional neural network is capable of achieving record breaking results on a highly challenging dataset using purely supervised learning. It is notable that our network’s performance degrades if a single convolutional layer is removed. For example, removing any of the middle layers

results in a loss of about 2% for the top-1 performance of the network. So the depth really is important for achieving our results.

2.7 GOING DEEPER WITH CONVOLUTIONS

Pierre Sermanet et al. has proposed in this paper, We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection. In the last three years, mainly due to the advances of deep learning, more concretely convolutional networks, the quality of image recognition and object detection has been progressing at a dramatic pace. One encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures. No new data sources were used, for example, by the top entries in the ILSVRC 2014 competition besides the classification dataset of the same competition for detection purposes. Our GoogLeNet submission to ILSVRC 2014 actually uses 12× fewer parameters than the winning architecture of Krizhevsky et al from two years ago, while being significantly more accurate. Another notable factor is that with the ongoing traction of mobile and embedded computing, the efficiency of our algorithms – especially their power and memory use – gains importance. It is noteworthy that the considerations leading to the design of the deep architecture presented in this paper included this factor rather than having a sheer fixation on accuracy numbers. For most of the experiments, the models were designed to keep a computational budget of 1.5 billion multiply-adds at inference time, so that they do not end up to be a purely academic curiosity, but could be put to real world use, even on large datasets, at a reasonable cost. Our results seem

to yield a solid evidence that approximating the expected optimal sparse structure by readily available dense building blocks is a viable method for improving neural networks for computer vision. The main advantage of this method is a significant quality gain at a modest increase of computational requirements compared to shallower and less wide networks. Also note that our detection work was competitive despite of neither utilizing context nor performing bounding box regression and this fact provides further evidence of the strength of the Inception architecture. Although it is expected that similar quality of result can be achieved by much more expensive networks of similar depth and width, our approach yields solid evidence that moving to sparser architectures is feasible and useful idea in general.

2.8 VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan et al. has proposed in this paper In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localization and classification tracks respectively. We also show that our representations generalize well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision. With ConvNets becoming more of a commodity in the computer vision field, a number of attempts have been made to improve the original architecture of Krizhevsky et al. (2012) in a bid to achieve better accuracy. For instance, the best-performing submissions to the ILSVRC 2013 (Zeiler & Fergus, 2013; Sermanet et al., 2014) utilised smaller receptive window size and smaller stride of the first convolutional layer. Another line of improvements dealt with training and testing the networks densely over the whole image and over multiple scales (Sermanet et al., 2014; Howard, 2014). In this paper, we address another

important aspect of ConvNet architecture design – its depth. To this end, we fix other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3×3) convolution filters in all layers. The initialization of the network weights is important, since bad initialization can stall learning due to the instability of gradient in deep nets. To circumvent this problem, we began with training the configuration A (Table 1), shallow enough to be trained with random initialization. Then, when training deeper architectures, we initialized the first four convolutional layers and the last three fully connected layers with the layers of net A (the intermediate layers were initialized randomly). We did not decrease the learning rate for the pre-initialized layers, allowing them to change during learning. For random initialization (where applicable), we sampled the weights from a normal distribution with the zero mean and 10^{-2} variance. The biases were initialized with zero. It is worth noting that after the paper submission we found that it is possible to initialize the weights without pre-training by using the random initialization procedure of Glorot & Bengio (2010). Our representation achieves the state of art on the VOC action classification task even without using the provided bounding boxes, and the results are further improved when using both images and bounding boxes. Unlike other approaches, we did not incorporate any task-specific heuristics, but relied on the representation power of very deep convolutional features.

2.9 DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION

Kaiming He et al. has proposed in this paper. Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers— $8\times$ deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10

with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. Deep convolutional neural networks have led to a series of breakthroughs for image classification. Deep networks naturally integrate low/mid/high level features and classifiers in an end-to-end multilayer fashion, and the “levels” of features can be enriched by the number of stacked layers (depth). Recent evidence reveals that network depth is of crucial importance, and the leading results on the challenging ImageNet dataset all exploit “very deep” models, with a depth of sixteen to thirty. Many other non trivial visual recognition tasks have also greatly benefited from very deep models. Driven by the significance of depth, a question arises: Is learning better networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradient, which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization and intermediate normalization layers, which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation. We argue that this is because of overfitting. The 1202-layer network may be unnecessarily large (19.4M) for this small dataset. Strong regularization such as max out or dropout is applied to obtain the best results on this dataset. In this paper, we use no maxout/dropout and just simply impose regularization via deep and thin architectures by design, without distracting from the focus on the difficulties of optimization. But combining with stronger regularization may improve results, which we will study in the future.

2.10 DENSELY CONNECTED CONVOLUTIONAL NETWORKS

Gao Huanget.al. Has been proposed this paper. Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce the Dense Convolutional Network (Dense Net), which connects each layer to every other layer in a

feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections—one between each layer and its subsequent layer—our network has $L(L+1)/2$ direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. Dense Nets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. We evaluate our proposed architecture on four highly competitive object recognition benchmark tasks (CIFAR-10, CIFAR-100, SVHN, and ImageNet). Convolutional neural networks (CNNs) have become the dominant machine learning approach for visual object recognition. Although they were originally introduced over 20 years ago, improvements in computer hardware and network structure have enabled the training of truly deep CNNs only recently. The original LeNet5 consisted of 5 layers, VGG featured 19, and only last year Highway Networks and Residual Networks (ResNets) have surpassed the 100-layer barrier. As CNNs become increasingly deep, a new research problem emerges: as information about the input or gradient passes through many layers, it can vanish and “wash out” by the time it reaches the end (or beginning) of the network. Many recent publications address this or related problems. ResNets and Highway Networks pass signal from one layer to the next via identity connections. Stochastic depth shortens ResNets by randomly dropping layers during training to allow better information and gradient flow. We proposed a new convolutional network architecture, which we refer to as Dense Convolutional Network (Dense Net). It introduces direct connections between any two layers with the same feature-map size. We showed that Dense Nets scale naturally to hundreds of layers, while exhibiting no optimization difficulties. In our experiments, Dense Nets tend to yield consistent improvement in accuracy with growing number of parameters, without any signs of performance degradation or overfitting. Under multiple settings, it achieved state-of-the-art results across several highly competitive datasets. Moreover, DenseNets require substantially fewer parameters and less computation to achieve state-of-the-art performances. Because we adopted hyperparameter settings optimized for residual networks in our study, we believe that further gains in accuracy of DenseNets may be obtained by more detailed tuning of hyperparameters and learning rate schedules.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the field of agricultural information, the automatic identification and diagnosis of maize leaf diseases is highly desired. To improve the identification accuracy of maize leaf diseases and reduce the number of network parameters, the improved Google Net and Cifar10 models based on deep learning are proposed for leaf disease recognition in this paper. Two improved models that are used to train and test nine kinds of maize leaf images are obtained by adjusting the parameters, changing the pooling combinations, adding dropout operations and rectified linear unit functions, and reducing the number of classifiers. In addition, the number of parameters of the improved models is significantly smaller than that of the VGG and Alex Net structures.

3.1.1 DRAWBACKS

- As with any deep learning model, there is a risk of overfitting, where the model learns to perform well on the training data but does not generalize well to new data. This can be mitigated by using techniques such as data augmentation and regularization.
- Deep learning models can be computationally expensive to train and deploy. This may be a limitation for farmers and other agricultural stakeholders with limited resources.
- It requires large amounts of data to train effectively. This may be a challenge for maize leaf disease recognition, as there is a limited amount of publicly available data.

3.2 PROPOSED SYSTEM

Maize leaf diseases are caused by various pathogens, including fungi, bacteria, and viruses. These pathogens can lead to leaf discoloration, necrosis, and reduced photosynthetic efficiency, which in turn can hinder the plant's growth and development. Furthermore, untreated or poorly managed diseases can lead to crop loss, decreased farmer income, and food scarcity. The proposed system for maize leaf disease detection using multi-support vector machine (MSVM) is a computer-aided diagnosis system that can accurately identify and classify maize leaf diseases early on. The system works by first collecting a dataset of maize leaf images. The dataset should contain images of both healthy and diseased maize leaves. The images are then pre-processed, which may involve resizing the images, converting them to grayscale, and extracting features from the images. The pre-processed images are then used to train the MSVM model. The trained model can then be used to classify new maize leaf images as either healthy or diseased.

3.2.1 ADVANTAGES

- MSVM is a powerful machine learning algorithm that is well-suited for maize leaf disease detection. It is able to learn complex patterns in the data and generalize to new data, which is important for developing a robust disease detection system.
- This is important for maize leaf disease detection, as farmers need to be able to identify and treat diseases quickly to minimize crop losses.
- The MSVM-based maize leaf disease detection systems can be developed and deployed using relatively inexpensive hardware and software. This makes them accessible to farmers in developing countries, where resources may be limited.
- The maize leaf disease detection systems can be designed to be easy to use by farmers.

3.3 FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

3.3.1 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at DB2 Database. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and

are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.3.2 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

3.3.3 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS:

Processor Type : AMD RYZEN 7

Speed : 4.40GHZ

RAM :16 GB RAM

☐ Hard disk : 1 TB

☐ Keyboard : 101/102 Standard Keys

☐ Mouse : Optical Mouse

4.2 SOFTWARE SPECIFICATION

Operating System : Windows 10

Front End : MATLAB

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 FEATURES OF SOFTWARE

ABOUT MATLAB

Key Features

- ✓ Numeric Computation
- ✓ Data Analysis and Visualization
- ✓ Programming and Algorithm Development
- ✓ Application Development and Deployment
- ✓ Contact Sales
- ✓ Product Trial
- ✓ Pricing and Licensing

KEY FEATURES

High-level language for numerical computation, visualization, and application development Interactive environment for iterative exploration, design, and problem solving Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations Built-in graphics for visualizing data and tools for creating custom plots Development tools for improving code quality and maintainability and maximizing performance Tools for building applications with custom graphical interfaces Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel® Analyzing and visualizing data using the MATLAB desktop. Enlarge Analyzing and visualizing data using the MATLAB desktop. The MATLAB environment also lets you write programs and develop algorithms and applications.

NUMERIC COMPUTATION

MATLAB provides a range of numerical computation methods for analyzing data, developing algorithms, and creating models. The MATLAB language includes mathematical functions that support common engineering and science operations. Core math functions use processor-optimized libraries to provide fast execution of vector and matrix calculations.

AVAILABLE METHODS INCLUDE:

- ✓ Interpolation and regression
- ✓ Differentiation and integration
- ✓ Linear systems of equations
- ✓ Fourier analysis
- ✓ Eigenvalues and singular values
- ✓ Ordinary differential equations (ODEs)
- ✓ Sparse matrices

MATLAB add-on products provide functions in specialized areas such as statistics, optimization, signal analysis, and machine learning. Refinement of gridded data using 2-D cubic interpolation. Enlarge Refinement of gridded data using 2-D cubic interpolation. Data Analysis and Visualization MATLAB provides tools to acquire, analyze, and visualize data, enabling you to gain insight into your data in a fraction of the time it would take using spreadsheets or traditional programming languages. You can also document and share your results through plots and reports or as published MATLAB code.

ACQUIRING DATA

MATLAB lets you access data from files, other applications, databases, and external devices. You can read data from popular file formats such as Microsoft Excel; text or binary files; image, sound, and video files; and scientific files such as netCDF and HDF. File I/O functions let you work with data files in any format. Using MATLAB with add-on products, you can acquire data from hardware devices, such as your computer's serial

port or sound card, as well as stream live, measured data directly into MATLAB for analysis and visualization. You can also communicate with instruments such as oscilloscopes, function generators, and signal analyzers. A mixed numeric and text file for import into MATLAB using the Import Tool.

ENLARGE

A mixed numeric and text file for import into MATLAB using the Import Tool. MATLAB automatically generates a script or function to import the file programmatically.

ANALYZING DATA

MATLAB lets you manage, filter, and preprocess your data. You can perform exploratory data analysis to uncover trends, test assumptions, and build descriptive models. MATLAB provides functions for filtering and smoothing, interpolation, convolution, and fast Fourier transforms (FFTs). Add-on products provide capabilities for curve and surface fitting, multivariate statistics, spectral analysis, image analysis, system identification, and other analysis tasks. Fitting a surface to data with a custom model using MATLAB and Curve Fitting Toolbox. Enlarge Fitting a surface to data with a custom model using MATLAB and Curve Fitting Toolbox.

VISUALIZING DATA

MATLAB provides built-in 2-D and 3-D plotting functions, as well as volume visualization functions. You can use these functions to visualize and understand data and communicate results. Plots can be customized either interactively or programmatically. The MATLAB plot gallery provides examples of many ways to display data graphically in MATLAB. For each example, you can view and download source code to use in your MATLAB application.

FEATURES OF MATLAB

Following are the basic features of MATLAB –

- It is a high-level language for numerical computation, visualization and application development.
- It also provides an interactive environment for iterative exploration, design and problem solving.
- It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
- It provides built-in graphics for visualizing data and tools for creating custom plots.
- MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.
- It provides tools for building applications with custom graphical interfaces.
- It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

USES OF MATLAB

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including –

- Signal Processing and Communications
- Image and Video Processing
- Control Systems
- Test and Measurement
- Computational Finance

- Computational Biology

Structures

MATLAB has structure data types.^[15] Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.). Unfortunately, MATLAB JIT does not support MATLAB structures, therefore just a simple bundling of various variables into a structure will come at a cost.

Functions

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. Functions are also often case sensitive.

Function handles

MATLAB supports elements of lambda calculus by introducing function handles, or function references, which are implemented either in .m files or anonymous/nested functions.

Classes and object-oriented programming

MATLAB supports object-oriented programming including classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics. However, the syntax and calling conventions are significantly different from other languages. MATLAB has value classes and reference classes, depending on whether the class has *handle* as a super-class (for reference classes) or not (for value classes).

CHAPTER 6

PROJECT DESCRIPTION

6.1 PROBLEM DEFINITION

Maize leaf diseases are a major threat to maize production, causing significant yield losses. Early detection and diagnosis of maize leaf diseases is essential for effective disease management. However, traditional methods of maize leaf disease diagnosis are time-consuming, labor-intensive, and require expertise. Deep learning-based approaches have the potential to automate the diagnosis of maize leaf diseases, making it more efficient and accessible. However, existing deep learning models for maize leaf disease recognition are often complex and require large amounts of data to train. This can make them difficult to deploy in real-world settings.

6.2 MODULE DESCRIPTION

6.2.1 LOAD DATA

In this step, we'll gather and import the data that we'll use for our maize leaf disease detection project. This data will typically include images of maize leaves, and each image should be labelled or categorized based on the type of disease it represents. Loading data is a crucial initial step as our model's performance relies heavily on the quality and quantity of the data we use.

6.2.2 DATA PREPROCESSING

Data pre-processing involves cleaning and preparing the data for further analysis. This step might include tasks such as resizing images, normalizing pixel values, removing noise, and augmenting the data if necessary. Data pre-processing ensures that your data is in a suitable format for training your machine learning model.

6.2.3 FEATURE SELECTION

In machine learning, feature selection involves choosing the most relevant attributes or features from your data that will be used to make predictions. For image-based

classification tasks like maize leaf disease detection, you might use techniques like feature extraction, or we could leverage deep learning models to automatically extract meaningful features from the images.

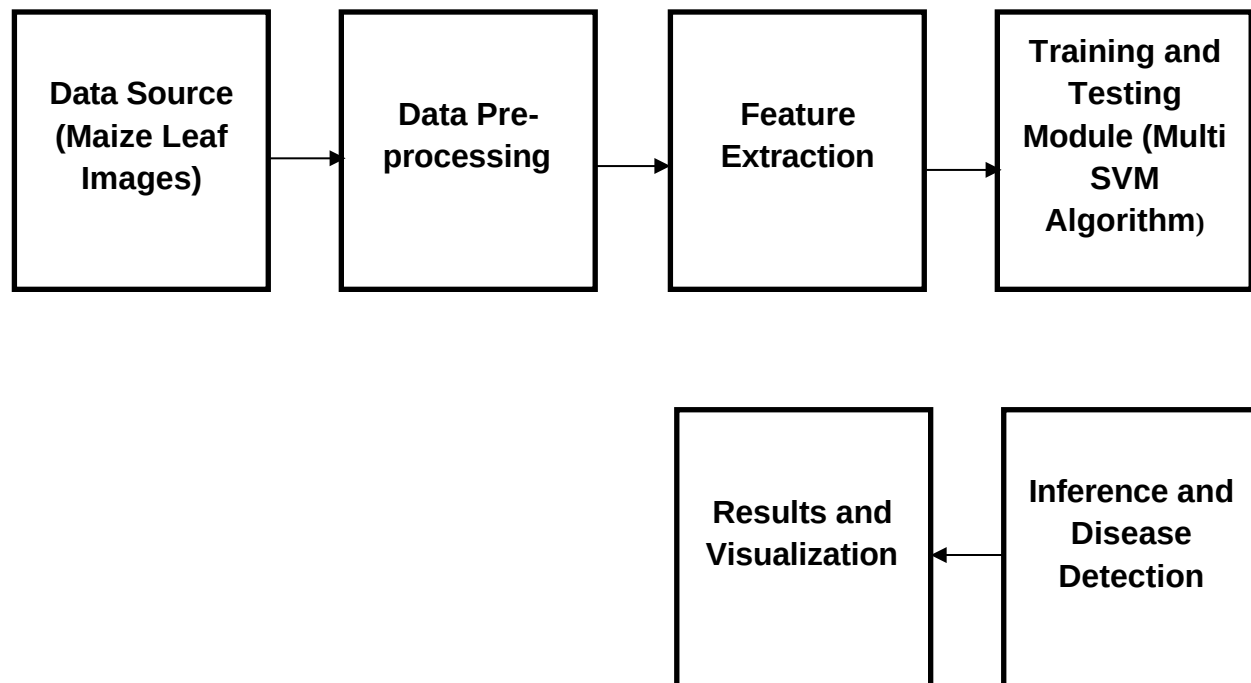
6.2.4 TRAINING AND TESTING

This step involves splitting our pre-processed data into training and testing sets. The training set is used to train our Multi Support Vector Machine (SVM) model to learn the patterns in the data. The testing set is used to evaluate how well your model generalizes to new, unseen data. Training and testing are essential for model development and ensuring that your model performs well on real-world examples.

6.2.5 EVALUATION AND PERFORMANCE

In this phase, we assess the model's performance using various metrics such as accuracy, precision, recall, and F1 score. The evaluation helps you understand how well your Multi SVM algorithm is at detecting maize leaf diseases. we may also perform fine-tuning and optimization in this stage to improve our model's performance.

6.3 SYSTEM FLOW DIAGRAM



6.4 INPUT DESIGN

The input design for a maize leaf disease detection system using multi-support vector machine (MSVM) should consider the following factors:

- Image quality: The system should be able to handle images of varying quality, including images that are blurry, noisy, or have poor lighting.
- Image size: The system should be able to handle images of different sizes, but it is important to standardize the image size before feeding it to the MSVM model.
- Image format: The system should be able to handle images in a variety of formats, such as JPEG, PNG, and TIFF.
- Feature extraction: The system should extract relevant features from the images that can be used to distinguish between healthy and diseased maize leaves.

6.5 OUTPUT DESIGN

The output design for a maize leaf disease detection system using multi-support vector machine (MSVM) should consider the following factors:

- Accuracy: The system should be able to accurately identify and classify maize leaf diseases.
- Speed: The system should be able to classify maize leaf images quickly and efficiently.
- Interpretability: The system should be able to provide interpretable results that can be understood by users.

CHAPTER 7

7. SYSTEM TESTING AND IMPLEMENTATION

7.1 SYSTEM TESTING

System testing for a maize leaf disease detection system using multi-support vector machine (MSVM) should be conducted to ensure that the system meets the following requirements:

- Accuracy: The system should be able to accurately identify and classify maize leaf diseases.
- Speed: The system should be able to classify maize leaf images quickly and efficiently.
- Reliability: The system should be able to operate reliably in a variety of conditions.
- Usability: The system should be easy to use and understand.
- Security: The system should be secure and protect user data.

7.2 SYSTEM IMPLEMENTATION

To implement a maize leaf disease detection system using multi-support vector machine (MSVM), the following steps can be followed:

1. Collect a dataset of maize leaf images. The dataset should contain images of both healthy and diseased maize leaves. The images can be collected from various sources, such as field surveys, agricultural research stations, and online databases.
2. Pre-process the images. This may involve resizing the images, converting them to grayscale, and extracting features from the images. There are a variety of image processing techniques that can be used for this purpose.

3. Train the MSVM model. The pre-processed images are then used to train the MSVM model. This involves feeding the images to the model and providing the corresponding labels (healthy or diseased).
4. Implement the system. The trained MSVM model can be implemented as a software application or a web service. The application should provide users with a way to upload images of maize leaves and receive the disease classification results.

CHAPTER 8

SYSTEM MAINTENANCE

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In this project the process can be added without affecting other parts of the system. Maintenance plays a vital role. The system is liable to accept any modification after its implementation. This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy.

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environment. It has been seen that there are always some errors found in the system that must be noted and corrected. It also means the review of the system from time to time.

The review of the system is done for:

- Knowing the full capabilities of the system.
- Knowing the required changes or the additional requirements.
- Studying the performance.

TYPES OF MAINTENANCE:

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

8.1 CORRECTIVE MAINTENANCE

Changes made to a system to repair flaws in its design coding or implementation. The design of the software will be changed. The corrective maintenance is applied to correct the errors that occur during that operation time. The user may enter invalid file type while submitting the information in the particular field, then the corrective maintenance will display the error message to the user in order to rectify the error.

Maintenance is a major income source. Nevertheless, even today many organizations assign maintenance to unsupervised beginners, and less competent programmers.

The user's problems are often caused by the individuals who developed the product, not the maintainer. The code itself may be badly written maintenance is despised by many software developers unless good maintenance service is provided, the client will take future development business elsewhere. Maintenance is the most important phase of software production, the most difficult and most thankless.

8.2 ADAPTIVE MAINTENANCE:

It means changes made to system to evolve its functionalities to change business needs or technologies. If any modification in the modules the software will adopt those modifications. If the user changes the server then the project will adapt those changes. The modification server work as the existing is performed.

8.3 PERFECTIVE MAINTENANCE:

Perfective maintenance means made to a system to add new features or improve performance. The perfective maintenance is done to take some perfect measures to maintain the special features. It means enhancing the performance or modifying the programs to respond to the users need or changing needs. This proposed system could be added with additional functionalities easily. In this project, if the user wants to improve the performance further then this software can be easily upgraded.

8.4 PREVENTIVE MAINTENANCE:

Preventive maintenance involves changes made to a system to reduce the changes of features system failure. The possible occurrence of error that might occur are forecasted and prevented with suitable preventive problems. If the user wants to improve the performance of any process then the new features can be added to the system for this project.

CHAPTER 9

9. CONCLUSION

Maize leaf diseases can cause significant crop losses and reduce farmer income. Early detection and classification of these diseases is essential for effective crop management. A maize leaf disease detection system using multi-support vector machine (MSVM) is a promising approach for early detection and classification of maize leaf diseases. The MSVM model is trained on a dataset of maize leaf images that have been pre-processed to extract relevant features. Once the model is trained, it can be used to classify new maize leaf images as either healthy or diseased. The MSVM model is well-suited for maize leaf disease detection because it is able to learn complex relationships between the features of the images and the corresponding disease labels. A maize leaf disease detection system using MSVM can be implemented as a software application or a web service. The system should provide users with a way to upload images of maize leaves and receive the disease classification results. The system should be easy to use and maintain, and it should provide users with the information they need to make informed decisions about crop management.

FUTURE WORK

Improve the accuracy of the MSVM model. This can be done by using a larger and more diverse dataset of maize leaf images, or by using more advanced machine learning techniques. Develop a real-time maize leaf disease detection system. This would allow farmers to detect diseases in their crops early on, before they cause significant damage. Integrate the MSVM model with other crop management systems. This would allow farmers to manage their crops more effectively and efficiently. Develop a mobile app for maize leaf disease detection. This would make the technology more accessible to farmers in developing countries.

CHAPTER 10

APPENDICES

10.1 SCREEN SHOTS

INPUT IMAGE 1

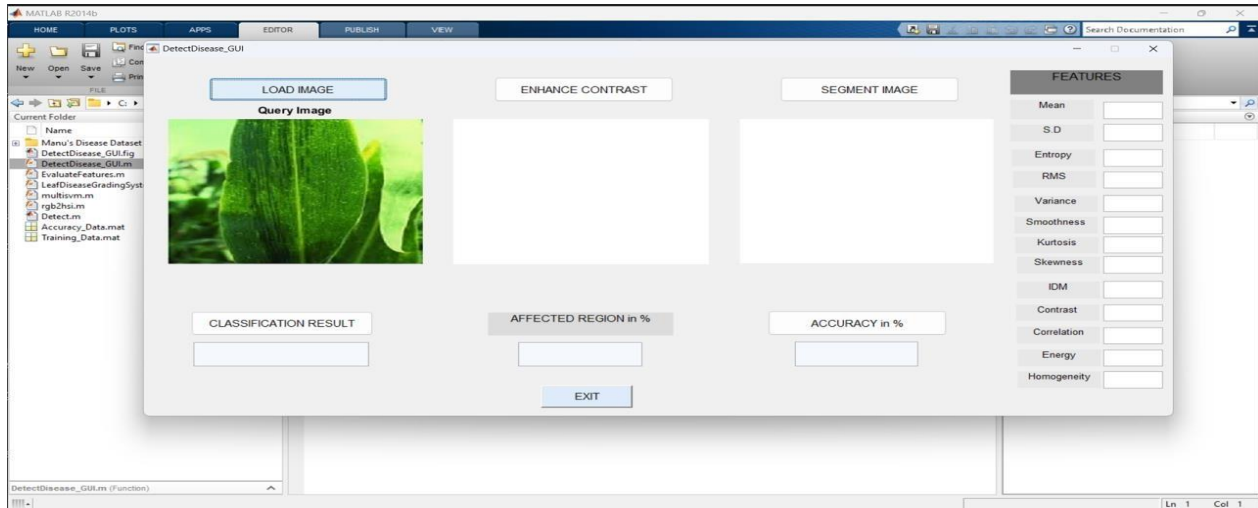


Figure 2 Input Image 1

OUTPUT IMAGE 1

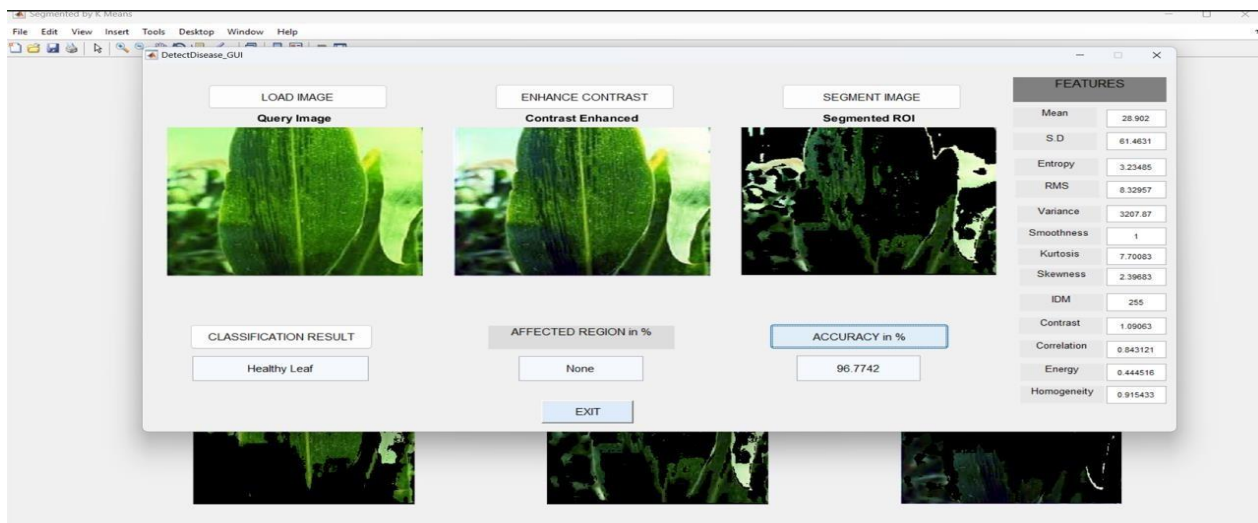


Figure 3 Output Image 1

INPUT IMAGE 2

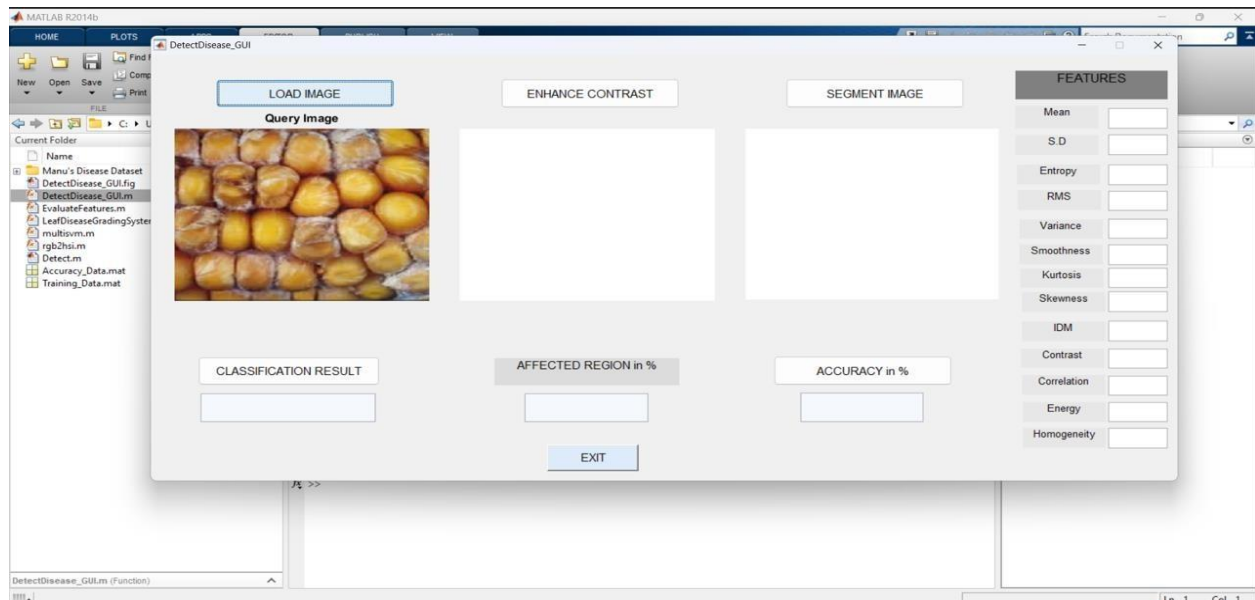


Figure 4 Input Image 2

OUTPUT IMAGE 2

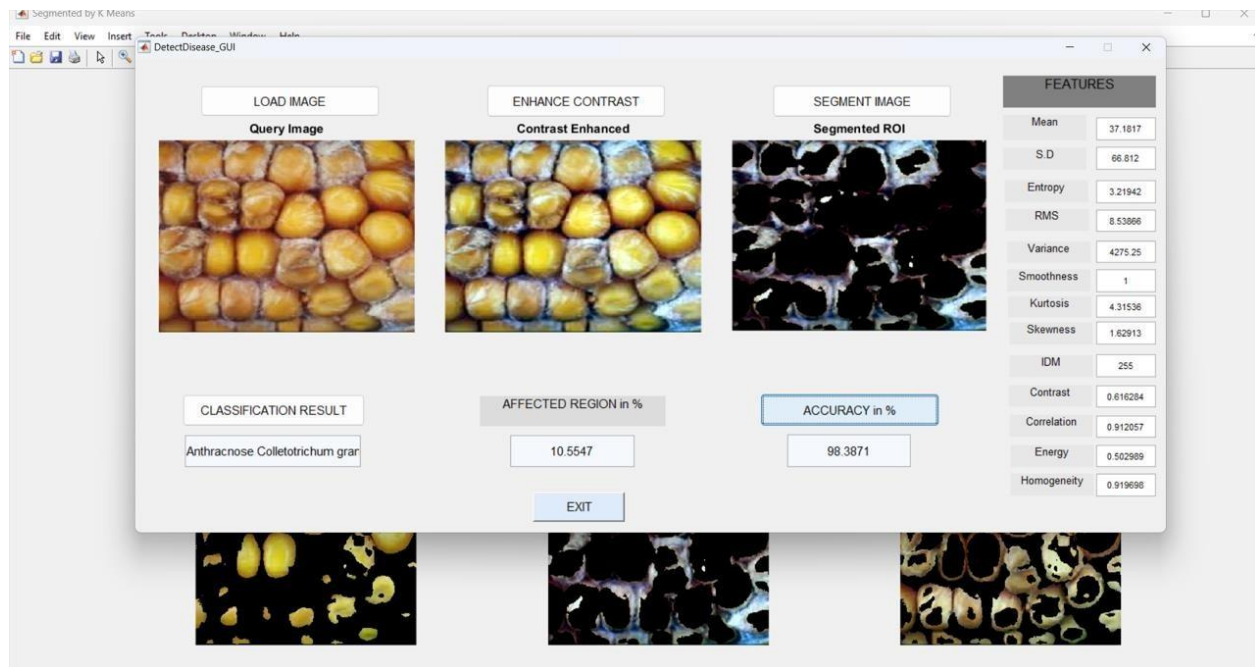


Figure 5 Output Image 2

CHAPTER 11

11. REFERENCES

- [1] V. Singh and A. K. Misra, “Detection of plant leaf diseases using image segmentation and soft computing techniques,” *Inf. Process. Agricult.*, vol. 4, pp. 41–49, Mar. 2017.
- [2] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, “How deep learning extracts and learns leaf features for plant classification,” *Pattern Recognit.*, vol. 71, pp. 1–13, Nov. 2017
- [3] G. Farjon, O. Krikeb, A. B. Hillel, and V. Alchanatis, “Detection and counting of flowers on apple trees for better chemical thinning decisions,” *Precis. Agricult.*, vol. 21, pp. 1–19, Aug. 2019.
- [4] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016
- [5] F. Ren, W. Liu, and G. Wu, “Feature reuse residual networks for insect pest recognition,” *IEEE Access*, vol. 7, pp. 122758–122768, 2019
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.
- [7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556
- [9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778