



**NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF COMPUTER ENGINEERING**

**IMPLEMENTATION OF PLANT IDENTIFICATION USING DEEP
LEARNING**

M.Sc. THESIS

Neveen salah MUSTAFA

**NICOSIA
April, 2022**

**NEVEEN SALAH
MUSTAFA**

**IMPLEMENTATION OF PLANT
IDENTIFICATION USING MACHINE
LEARNING/DEEP LEARNING**

**MASTER THESIS
2022**

**NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF COMPUTER ENGINEERING**

IMPLEMENTATION OF PLANT IDENTIFICATION USING DEEP LEARNING

M.Sc. THESIS

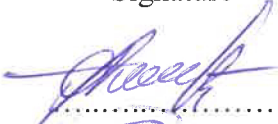


Neveen salah MUSTAFA

**Supervisor
Assoc. Prof. Dr. Boran Şekeroğlu**

**NICOSIA
April, 2022**

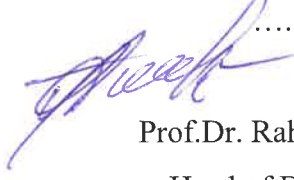
Approval

We certify that we have read the thesis submitted by Neveen Salah Mustafa titled **“INVESTIGATION AND FORECAST OF COMMON CATASTROPHE UTILIZING SPATIAL INFORMATION MINING PROCEDURE”** and that in our combined opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Educational Sciences.

Examining Committee	Name-Surname	Signature
Head of the Committee:	Prof.Dr. Rahib Abiyev	
Committee Member*:	Dr. Ahmet İlhan	
Supervisor:	Assoc. Prof. Dr. Boran Şekeroğlu	

Approved by the Head of the Department


...../...../20...



Prof.Dr. Rahib Abiyev
Head of Department

Approved by the Institute of Graduate Studies

...../...../20...



Prof. Dr. Kemal Hüsnü Can Başer
Head of the Institute

Declaration

I hereby declare that all information, documents, analysis, and results in this thesis have been collected and presented according to the academic rules and ethical guidelines of the Institute of Graduate Studies, Near East University. I also declare that, as required by these rules and conduct, I have fully cited and referenced information and data that are not original to this study.

Neveen Salah Mustafa

Date: 25/04/2022

Acknowledgments

This thesis would not have been possible without the help, support, and patience of my principal supervisor Assoc. Prof. Dr. Boran Şekeroğlu for his constant encouragement and guidance. He has walked me through all the stages of the writing of my thesis. Without his consistent and illuminating instruction, this thesis could not have reached its present form. Above all, my unlimited thanks and heartfelt love would be dedicated to my dearest family for their loyalty and their great confidence in me.

Neveen Salah Mustafa

Abstract**Implementation of Plant Identification Using Deep Learning****Neveen Salah Mustafa****MA, Department of Computer Engineering****April, 2022, 62 pages**

Plant recognition as being one of the most interesting topics in recent times. It has been on the rise in taxonomy, botany, and the agricultural industry. In this era of emerging technology, artificial intelligence techniques, which include machine learning/deep learning, have been effective in the area of prediction. This technology proffers essential models for the identification of plant species. Having the option to identify plant images, particularly its species, has never been so imperatively significant. The people of the florist industry will depend on these species for their livelihood, experimentation, and business. The objective of this thesis is to design a model to identify plants and give a predictable outcome. This thesis aims to apply a machine learning/deep learning strategy known as convolutional neural networks for plant identification. Moreover, different parameters are applied to achieve superior results and PCA. This model will be evaluated and differentiated to perceive the best outcome. The datasets used are gotten from the Swedish Botanic Garden, which has various plant species. It is believed that this work will develop an insightful model with high and strong accuracy to support people, botanists and specialists whenever thought about.

Keywords: Artificial intelligence, plant, convolutional neural networks, florist industry

Özet

Implementation of Plant Identification Using Deep Learning

Neveen Salah Mustafa

MA, Department of Computer Engineering

April, 2022, 62 pages

Bitki tanıma, son zamanların en ilginç konularından biri. Taksonomide, botanikte ve tarım endüstrisinde yükselişte. Bu gelişen teknoloji çağında, makine öğrenmesi/derin öğrenmeyi içeren yapay zeka teknikleri tahmin alanında etkili olmuştur. Bu teknoloji, bitki türlerinin tanımlanması için temel modeller sunar. Bitki görüntülerini, özellikle de türlerini tanımlama seçeneğine sahip olmak hiç bu kadar önemli olmamıştı. Çiçekçilik endüstrisinin insanları, geçimleri, deneyleri ve işleri için bu türlere bağımlı olacaktır. Bu tezin amacı, bitkileri tanımlamak ve tahmin edilebilir bir sonuç vermek için bir model tasarlamaktır. Bunu yapmak için tez, evrişimli sinir ağları olarak bilinen bir makine öğrenimi/derin öğrenme stratejisi toplamayı ve uygulamayı planlamaktadır. Ayrıca modelin doğruluğuna ve genel uygulamasına yardımcı olmak için farklı parametreler ve ayrıca PCA gibi parametre geliştiriciler uygulanacaktır. Bu model değerlendirilecek ve en iyi sonucu algılamak için farklılaştırılacaktır. Kullanılan veri setleri, çeşitli bitki türlerinin bulunduğu İsveç Botanik Bahçesinden alınmıştır. Bu çalışmanın insanlara, botanikçilere ve uzmanlara ne zaman düşünülürse yardımcı olmak için yüksek ve güçlü doğrulukta anlayışlı bir model geliştireceğine inanılıyor.

Anahtar Kelimeler: Yapay zeka, bitki, evrişimli sinir ağları, çiçekçi endüstrisi

Table Of Contents

Approval.....	.ii
Declaration.....	.iii
Acknowledgments.....	.iv
Abstract.....	.v
Öz.....	.vi
Table of Contents.....	.vii
List of Figures.....	.ix
List of Tables.....	.x
List of Abbreviations.....	.xi

CHAPTER I

Introduction.....	1
Background of the study.....	1
Statement of the problem.....	3
Aim and objective of the study.....	4
Significance of the study and contribution	5
Limitation.....	5
Proposal organisation.....	5

CHAPTER II

Literature Review.....	7
Concept of plant identification using deep/machine learning	7
Shape and surface based of plant recognizable proof	8
Plant identification based on deep learning	9
Review of previous works on plant identification using deep learning	9
Related works	11

CHAPTER III

Methodology.....	14
Area of Study and Data Set	14
Informatics and data processing	16
Using the Gaussian Process Latent Variable Model as a starting point	17
Neural Networks with Convolutions and PCA.....ç.....	19
Model Evaluation Using Statistical Methods	23
Latent Characteristics are classified according to their nature	24

CHAPTER IV

Result Presentation and Analysis	25
Model Processing	25
Model Performance	27
Discussion of Results	31

CHAPTER V

Conclusion and Recommendations	34
Conclusion.....	34
Recommendations.....	34
References.....	36
Appendices.....	43
Appendices A: Source Codes.....	43
Appendices B: Ethical and approval document.....	49
Appendices C: Plagiarism and Ethical Approval Form	50

List of Figures

Figure 1: Depicting the raw data before cropping	26
Figure 2: Depicting the data after cropping	26
Figure 3: Gaussian distortion for greyscale imaging	27
Figure 4: The CNN architecture.....	28
Figure 5: Activation maps of the data	28
Figure 6: The patches from the PCA.....	29
Figure 7: Patches graph from the PCA	29
Figure 8: Accuracy of the CNN vs. CNN with PCA for ten epochs.....	31

List of Tables

Table 1: An overview of relevant research on the easily identifiable evidence of plant disease	13
Table 2: The data characteristics	15
Table 3: The resulting experiment of the PCA with CNN (%)	30
Table 4: Experimental result of the used model with others (%)	33

List of Abbreviations

CNN:	Convolutional Neural Network
SVM	Support Vector Machine
KNN:	K-Nearest Neighbors
ML:	Machine Learning
ANN:	Artificial Neural Network
IoT:	Internet of Things
NN:	Neural Network
DL:	Deep Learning
SL:	Supervised Learning
DT:	Decision Tree
AI:	Artificial Intelligence
FIS:	Fuzzy Inference System
ANFIS:	Adaptive Neuro-fuzzy Inference System

CHAPTER I

Introduction

Background of the Study

Today, it is employed in the area of computer vision, where artificial intelligence (AI) is considered to be the most significant element of our life, robotics, digital marketing transformation, medical field, banking, and commercial field (Sekeroglu, & Tuncal. K.; 2021; Adweb et al. 2021; Dogruyol & Sekeroglu 2019). Artificial intelligence (AI) is mainly designed to make machines capable of thinking and acting like humans, and machine learning will be a subsection of artificial intelligence (AI), as well as analysis. Theoretical algorithm analysis and a mathematical model that performs testing without explicit programming, based on hypotheses and models. The three types of machine learning methodologies include supervised learning, unsupervised learning, and reinforcement learning, among others. Output data is included inside supervised learning algorithms; however, unsupervised learning algorithms include only input data and unlabeled entries.

Deep learning delivers outstanding results, even on big data. Deep learning, used in image recognition and computer vision, etc., uses artificial neurons that are identical to human neurons. Deep neural networks, recurrent neural networks, and deep belief networks are all types of neural networks that are used in computer science for speech recognition, language processing, translation software, sound recognition, bioinformatics, and drug development. Plants are indeed an important member of all-natural life (Sapna and Gupta, 2015), and their official naming will ensure that all-natural life is preserved and maintained. Plants are essential for our medicinal purposes, are alternative energy sources such as biofuels, and are also used to meet many of our household needs such as wood, clothing, food, and trapping. The current extinction trend is mainly the product of indirect and overt human activity. Therefore, the generation of the correct information and geographical spread of plant species is crucial for the survival of future ecosystems.

Image recognition strategies have recently begun to emerge in an attempt to simplify this factory inspection process. For the study of plant identification, methods focusing on color

characteristics are often used to establish a method of plant recognition. The interpretation of colors may depend on even though this is not a safe function since there are certain instances in which the temporal precision of the function is compromised, it is useful for determining how colours are distributed in a picture. This functionality is often misused. Adjustments in lighting, leaf movement caused by waves, camera shaking, focus changes, and unexpected changes in camera settings all lead to incorrect estimates of plant species. Based on significant research, it is currently recognised that the kind of vegetation that is dependent on digital photographs is a challenging challenge to solve. This research has concentrated on identifying and classifying individual plant leaves in order to identify and categorise plants (Amala and Sreenkumar, 2017). They were employed in several investigations over the next years to construct models of tree leaf identification systems, which were then published.

According to Gaber et al. (2015), the scholars used MCA to derive visual characteristics from plants and linear discriminant analysis. Multiple researchers also have established the ability of a broad convolution neural network (CNN) to outperform conventional object recognition or detection strategies centered upon ordinary working light, texture, and shape characteristics. Naturally, the CNN systems are used in such bigscale plant recognizing activities which consist of a trait extractor accompanied by a classifier. Despite occlusions, collecting plant crops does not usually do a straightforward job. In fact, some plants may not even be able to see parts of the leaves. On the other hand, there has also been much research related to the use of machine vision approaches to find solutions to these issues. The recognition of aerial images of landscapes using machine learning techniques is an example of recognition computer technologies in agricultural computing, as is the recognition of aerial images of crops.

The automatic recognition of plant images bridging plants is the most promising option currently available in taxonomy, which is receiving considerable interest both in the botanical as well as digital communities. As machine learning technology progresses, more complicated models for autonomous plant identification are being developed and suggested. With the widespread use of smartphones and the rise of mobile applications, applications PlantNet (Joly and Goeau, 2014), millions of photographs of plants have been collected. Automated plant identification on mobile devices is essential for real-world ecological monitoring (Goeau, Bonnet and Joly, 2015) and tracking invasive alien plant species (Goeau, Bonnet and Joly,

2016), popularizing ecological science, Etc. The development of the performance of mobile plant identification models is gaining more and more attention from academics and industry professionals.

Many efforts have been made to extract native qualities from leaves, flowers, and fruits in the modern era, with varying degrees of success. To study plants, most researchers utilise changes in leaf characteristics as a comparison tool. There are various foliar datasets, including the Swedish leaf dataset, the Flavia dataset, and the ICL dataset, which are all considered standard references in the field. In (Sodakvist, 2001) used a reversed propagation for a feedforward neural network; the model was able to extract leaf shape and its features from several classes of plants and evaluate them. In (Fu et al., 2003), local contrast and other characteristics were chosen to describe the pixels around the ridge's surrounding pixels. The veins and other parts of the leaves were segmented using artificial neural networks. As a result of their extensive experience, neural networks have shown to be more successful in recognizing vein pictures.

Statement of the problem

The automatic diagnosis of plant diseases from leaves represents a significant step forward in the area of agriculture, according to the researchers. Plant diseases may also have a beneficial influence on crop productivity and quality when they are identified early and promptly (Sethy et al., 2020). Because of the increasing variety of plant products available, even farmers and plant pathologists are sometimes unable to recognize plant diseases just by seeing diseased leaves on the plant. Visual observation, on the other hand, continues to be the primary technique of illness detection in rural regions of developing nations (Chen et al., 2020). In addition, it needs the ongoing monitoring of specialists. Farmers in rural places may be required to travel vast distances to consult with specialists, which is both time-consuming and expensive (Bai et al., 2018; Ramcharan et al., 2017). Farmer and agronomists may benefit from automated computer systems for plant disease detection and diagnosis since they have a high throughput and accuracy.

Researchers have come up with a number of strategies to help them overcome these difficulties. It is possible to utilise several sorts of feature sets in machine learning to categorize plant

diseases in a variety of ways. Traditional manual-based features, as well as deep learning (DL) features, are the most often used of these feature sets. For effective feature extraction, preprocessing operations such as picture enhancement, colour conversion, and segmentation are required (Camargo and Smith, 2009). Following the decomposition of the characteristics, a variety of classifiers may be used.

Although research on automated plant taxa has yielded promising results, it is clear that these models are still far from meeting the needs of a scenario with completely automated ecological monitoring (Li et al. associates, 2005). It is not possible to include mobile-based plant photographs taken in natural environments since they vary greatly in terms of contributors, cameras, areas, and time of year, as well as individual plants and other characteristics. Traditional classification methods depend mostly on preprocessing to eliminate complicated background information and increase the effectiveness of the classification model. Furthermore, manual feature engineering is incapable of dealing with large-scale data sets comprised of unrestricted pictures. With the goal of overcoming the aforementioned difficulties and being inspired by the recent achievement in deep learning picture identification, the study recommends that Convolution Neural Networks be used to create a plant recognition approach.

Aim and objective of the study

The main objective of the study is the implementation of plant identification using machine learning/deep learning. Specifically, the study aims to:

- Explains the intuitive reasons for the classification using CNN with is a deep learning model; this methodology is based on a sample.
- Compared prediction accuracy using 5-fold cross-validation in 10 replicates using redesigned data sets.
- In addition to the prediction accuracy, the study also compares the model uncertainty using mutual information to get more details about the prediction process, while the error rate of the two approaches is compared with CNN using various parameters.

Significance of the Study and Contribution

As a result of this recent research study, many authorities and stakeholders will benefit from its resources. Scholars and researchers around the world will also benefit from the addition of another important reference to their libraries. Significantly, the study would contribute to existing knowledge in the following ways:

- Proposing a novel plant identification scheme through the application of the CNN technique.
- In this work, a CNN-based classifier is constructed for the seedlings class using photos, and the results are shown.
- The study is primarily concerned with recognizing the raw picture data set in order to categorize plant species and their photographs.

Limitation

The limitation of how the application works are that the images must be preprocessed and masked before converting them to the CNN TensorFlow Lite model. To overcome this situation, we can upload the image to a web application that can do the necessary preprocessing and send the resulting image back to the device. Additionally, models can be hosted on Firebase MLKit, which can improve developer efforts, as in-app model pooling seems like a daunting task every time a .tflite model is built. The MLKit can help update the model without requiring the user to manually update the application.

Proposal Organization

Regarding the project organization, the thesis is divided into five chapters, in addition to the references and appendices of the project.

Chapter 1: Provides background research, an overview, and a brief introduction to the research. Statements of matter and objectives, among others, are also provided in the same chapter.

Chapter 2: Presenting a literature review: the review includes various studies conducted on the main research topic. This chapter presents the conceptual framework that forms the core of the proposed study under consideration. Review the existing work of individuals in related fields.

Chapter 3: System Architecture, including a brief overview of the research methodology and a description of the techniques used in the study

Chapter 4: Results and discussion are presented in Section 4 to explore the overall results of the study and describe the data set used.

Chapter 5: Finally, the proposed work is concluded in this area.

CHAPTER II

Literature Review

Concept of Plant Identification Using Deep/Machine Learning

Linear fashions in records inclusive of straight relapse or calculated relapse do have the pick up of being white-container fashions, wherein the informative elements can without difficulty be explored (Freund and Wilson, 2003). Similarly, linear fashions inclusive of the Eigenfaces; Samek et al., (2018), counting on the important thing analysis (PCA), have been utilized in laptop vision. These fashions may be defined in a comparable way as linear regression fashions; however, they are afflicted by a robust linear assumption. However, this interpretability is not to be had for (deep) gadget mastering fashions for photo analysis.

The (deep) machine learning model description depicts the investigation of a prepared show arrangement in an interpretative space such as a heatmap (Montavon, 2018). This study is usually performed after training the model and aims to identify the reasons for individual prediction steps (Samek & Muller, 2019; Montavon et al., 2018). These reasons are usually coded into models that are not visible to the user and should be investigated to validate the black-box model (Samek et al., 2018). There are several strategies for visualizing model-based descriptions. i.e., exploring learned representations, describing individual predictions, describing model behavior, and reviewing explanatory elements based on representative examples (Samek et al., 2018).

A study of these representative training cases was performed by (Koh and Liang, 2017) to interpret the classifier. Similarly, Simonyan et al. (2013) calculated agent illustrations to think about the considered expression. In any case, these illustrations may not be accommodating in interpreting solutions in real-world images (Samek & Muller, 2019). To overcome these limitations, Lapuschkin et al. (2019); Montavon et al. (2018), and Boch et al. (2015) proposed a hierarchical distribution of relevance (Selvaraju et al., 2018) and proposed GradCAM as an elective methodology to imagine the reasons for person CNN expectations. These strategies

can be amplified to account for numerous arrangements and have been utilized to examine the smartHans wonder (Anders et al., 2021).

Be that as it may, the previously mentioned writing addresses the issue of portraying person choices based on profoundly fruitful profound learning models. As previously discussed in Benjo et al. (2013), it is necessary to explore hidden explanatory factors to create a resilient and solid framework. The illustrative strategies described above primarily use individual crystals or representative tests to accumulate data around covered illustrative components. The explanatory elements encoded in the deep learning model are undetectable. To overcome these problems, Markus (2018) proposed unsupervised learning. Inspired by Eigenfaces, Wobe et al. (2021) proposed a strategy based on nonlinear and nonparametric models of the hidden variables of an unsupervised Bayesian Gaussian process (Titsias, 2010). The authors proposed a methodology to overcome the smartHans phenomenon and manually visualize and examine explanatory elements of the model. The explanatory factor analysis is based on behavioral studies of the trained model and renders image regions used for feature extraction (Wober et al., 2021 Wober et al., 2019).

Shape and Surface-Based of Plant Recognizable proof

In Mokhtar et al. (2015), the authors recognized sicknesses in the usage of tomato leaf pictures. They utilized extraordinary geometric and histogram-primarily based capacities completely from portioned unhealthy amounts and carried out an SVM classifier with extraordinary parts for the course. Kaur et al. (2018) recognized three special soybean sicknesses the usage of special colour and texture functions. Babu and Rao (2021) used a feed-ahead neural community and again propagation to become aware of plant clears out and their ailments. Chouhan et al. (2018)] utilized a bacterial scavenging efficiency primarily derived completely characteristic of the radial base community of the nervous system (BRBFNN) fungal diseases in plants as well as the identification of leaves. The different methodologies they use are based on the idea that seed components with similar characteristics may be extracted from leaves using a region-developing set of criteria. Rapid growth and improved route accuracy are achieved by the use of the bacteria-foraging optimization technique (BFO).

Plant Identification based on Deep Learning

Mohanty et al. (2016) used ASPX.Net, and GoogleNet CNN and 26 identifiers are found in these structures distinctive plant diseases. Ferentinos et al. (2018) employed a variety of CNN architectures with their own unique characteristics to become aware of fifty-eight distinctive plant diseases, reaching excessive degrees of category accuracy. To put it another way, in expansion, They looked at the CNN building in real-time. Sladojevic et al. (2016) designate thirteen unique plant diseases as the focus of the DL structure. They utilized the Caffe DL system to carry out CNN training. Kamilaris et al. (2018) a thorough examination of the disadvantages and advantages of various DL techniques in the agricultural sector. In Geetharamani and Pandian (2019), the Nine layers were suggested by the designers. CNN adaptation to end up mindful of plant infections. They used the PlantVillage dataset and records-augmentation methods to build the measure of the record for testing purposes and analyzed execution. The developers claimed a greater degree of precision than is typically associated with machine learning.

Review of Previous works on Plant Identification Using Deep Learning

The literature on deep learning is very wide. This section was carried out via way of means of diverse researchers with insight on the subject of plant identity, and the usage of deep learning is defined in this section. Sharma (2015) used important issue analysis (PCA), Hu's second invariant approach, and morphological functions for class, and they have used 16 special instructions of the leaf. This Matlab measures the circumference via way of means of measuring the distance in every linked range of pixels alongside the area's boundary. Gaber (2015) recommended a plant recommender approach that makes use of 2D visible images of plants. This application used the technique of characteristic fusion and the method of the multilabel class. The experimental findings found that the feature fusion approach's accuracy became a great deal better than different character applications. The assessments confirmed their robustness in offering correct recommendations. Jassmann (2015) has a look at designing a brand new CNN. They examined the use of the newly applied Exponential Linear Unit (ELU) in preference to Rectified Linear Unit (ReLU) as CNN's non-linearity method. Yalcin (2016) counseled the structure of the CNN to perceive the shape of vegetation from the photo sequences acquired from clever agro-stations. The layout is used as a preprocessing degree to

dispose of the photo properties. Configuration of the CNN layout and breadth are critical factors that ought to be highlighted due to the fact they affect the spotting competencies of neural community structure. They used sixteen varieties of vegetation and compared them with different approaches; initial findings display that the CNN targeted approach's class overall performance outranks different approaches.

Sabu (2017) depicts that Universal Leaf Identification is a tough Computer Vision issue. Efficient leaf healing technique for Ayurvedic plants is useful for different components of society which include Medicine and research in Botany. Recognize the snap shots of the leaf. The look at centered on the one-of-a-kind methods and classifications for leaf identification. Lee (2017), amassed one of the images of plant leaves has additionally been mentioned primarily based totally upon the leaf traits used as an enter and convolution neural community is getting used to perceive styles for every plant intensity information. CNN turned into specifically applied right here only for the stepped forward portrayal of the traits, and for powerful research of Leaf organisms, DN (Deconvolutional Network) was used. It permits extra reputation of plant leaves and their populations. Ghazi (2017) applied three fashions of switch gaining knowledge to explain the identification of the numerous plants. The network turned into evaluated the usage of LIFECLEF 2015. These three versions used GoogleNet, VGGNet, and AlexNet for their initiatives.

Barbedo (2018) mentioned the evaluation of the important thing elements influencing the structure and efficacy of deep neural networks implemented to plant pathology and the intensity take a look at the topic, which illustrates the blessings and disadvantages, will make a contribution to greater concrete findings on plant pathology. Barbedo (2018) explored the implementation of troubles in switch mastering and the usage of deep mastering. They observed that CNN is a technique used to categorise plant biotechnology troubles. Zhu (2018) used CNN (Complex Background) to understand the small objected plant leaves. The designed technique carried out sample-normalization founding V2, which complements the accuracy of Region CNN. For processing, the exceptional pics sub-samples are cut up into 100, and the residual im-a long time are lower back to the very last production. The technique recommended that it can be quicker than traditional vicinity convolutional neural network.

Kaya (2019) counseled the idea of Transfer Learning for Plants Classification targeted on Deep Learning. This paper shows the effect of 4 separate transference schooling fashions on plant category offers to depend on DNN for 4 to be had databases. Finally, their theoretical studies exhibit that Transfer Learning gives a foundation for plant category self-estimating and analysing. They use sure not unusual place codecs such as End-to-End; Fine modulation, Fine modulation Cross Dataset, Deep Integrated Finetuning, Classification through RNN-CNN.

Related Works

The usage of the right strategies to pick out healthful and ailing clears empowers in controlling edit loss and growing efficiency. This phase accommodates exceptional current machine-studying strategies for the identity of plant diseases. A comparison table is given in Table 1.

Pretrained ASPX.Net and GoogleNet were utilized in Jadhav et al. (2020) to stumble on three exclusive soybean illnesses from healthy-leaf pics with changed hyper parameters consisting of how many minibatch and how many epochs max studying fee. Six exclusive prior knowledge networks (AXPS.Net, VGG16, VGG19, GoogLeNet, ResNet101, and DenseNet201) were utilized by Aravind et al.(2020) in order to choose ten distinct ailments in plants, and they achieved an accuracy rate of 97.3%% the usage of GoogleNet. A trained in advance VGG16 because the function extractor, as well as several classes of SVM, had been utilized in Rangarajan and Purushothaman (2020) to categorise exclusive eggplant illnesses. Different color shades have been evaluated using several color schemes (RGB, HSV, YCbCr, and grayscale); the usage of Using RGB colors, the very best category accuracy of 99.4% become a reality. In Arora Agrawal (2020), the researchers categorized maize-leaf sicknesses from wholesome leaves by the usage of deep-woodland techniques. As a whole method, They alter the hyper settings for the deep woods concern wide assortment with regard to trees, woodlands, and especially when contrasted to the traditional influence on the environment machine-studying styles counting SVM, RF, LR, and KNN. Lee et al. particular deep study architectures are compared to each other with inside is known about the disease in plants (Lee et al., 2020). To upgrade the exactness of the method above, Ghazi et al. (2017) utilized a transfer-learning program strategy on the basis of prior experience study designs (Ghazi et al., 2017).

In Li et al. (2020), the creators utilized a shallow CNN with classifiers for SVM and RF data 3 extraordinary varieties of ailmentariosis. As they went about their daily tasks, they particularly as the researchers matched their findings to those of the studying techniques and confirmed that class the usage of Classifiers using SVM and RF extricated capabilities from CNN's weak reporting defeated the set of pre-trained deep models. Self-interested convolutional neural networks are organized (SACNN) changed into utilized in Zeng and Li (2020) to pick out numerous crop diseases. To have a look at the robustness of the demonstration, the creators delivered extraordinary noise stages with interior the collection of images used for testing.

Oyewola et al. (2021) recognized five exceptional cassava-plant illnesses the use of undeniable convolutional neural organize (PCNN), and profound leftover organize (DRNN), and discovered DRNN beaten PCNN through an edge of 9.25%. Ramacharan et al. (2017) identified three different diseases and pests that attack cassava plants using a transfer research technique. After then, the writers published their work and prolonged their paintings on the identity of cassava plant illnesses; the use of a smartphone-primarily based completely on CNN adaptation and executed exactness of 80.6% (Ramcharan, 2019).

Adedaja et al. (2019) used a NASNet-based deep CNN structure to accurately detect leaf disease in plants with an accuracy rate of 93.82 percent. To identify rice and maize leaf diseases, Chen et al. (2020) employed the INC-VGGN approach. VGG19's last convolutional layer was reworked with the introduction of Layered and the addition of a single global common. Li et al. (2020) used the superficial CNN (SCNN) to identify maize, apple, and grape illnesses. In the first place, they took CNN's capabilities and dubbed them as SVM and RF Classifiers. Extreme depth of field was employed by Sethy et al. (2020) to gather knowledge about fashions in order to disentangle capacity and categorize them using an SVM. SVM and ResNet50 were used to get the greatest overall execution accuracy. Zhao et al. (2021) used a VGG16, ResNet, and DenseNet adaption to identify plant diseases from the plant town dataset. They used a two-fold generative opposing organization (DoubleGAN) to construct the dataset estimate, which advanced the overall execution results. The table below shows a similar portrayal of a plant disorder character-based just on leaf images.

Table 1.

An overview of relevant research on the easily identifiable evidence of plant disease

Source	Methodology	Result
Mohanty et al. (2016)	ASPX.Net and GoogleNet	99.57% of ASPX.NET's In GoogleNet, 99.34% of the time
Sladojevic et al. (2016)	Refined the architecture of the CNN	a 96.3% certainty rate
Ramcharan et al. (2017)	Inception V3 is built on GoogleNet	a 93% certainty rate
Fuentes et al. (2017)	Reinforced Convolutional Neural Networks	a VGG accuracy rate of 83% and a precision rate of 99.49%
Ferentinos et al. (2018)	ASPX.NetOWTBn and VGG	ASPX.Net OWTBn, of which 99.53 percent are in VGG
Ramacharan et al. (2019)	MobileNet location and classifier for the SSD model.	80.6% precision on Pictures, 70.4% precision on Video
Geetharamani et al. (2019)	An incredible nine-layer depth CNN	96.75% precision
Chen et al., (2020)	INC VGGN	96.75% precision
Li et al., (2020)	Deep CNN paired with SVM and RFC for a shallow learning	96.75% precision
Oyewola et al., (2021)	Profound leftover neural arrange (DRNN)	96.75% precision

CHAPTER III

Methodology

This study's model evaluation is based on image data from various plant species from the kingdom *Plantae*. Different models were used to process the photos. The following subsection goes over the data sets, processing pipelines for images, and the proposed deep learning technique.

Area of Study and Data Sets

Images of plant species from photosynthetic eukaryotes were acquired and used in this project from the Swedish Botanic Garden. During this research. The dataset added up to 10,031 data sheets of 15 different plant examples (from the species *Sorbus aucuparia*, *Ulmus carpinifolia*, and *Salix sinerea*) were employed, as depicted in Table 2. Most of the specimens were obtained between the years 1971 and 2020, with the majority of them taken between 1996 and 2010. All specimens have been perceived, and explanation labels have been set on the herbarium sheet for each one of these specimens.

Table 2.
The data characteristics

Plant species	No of images
<i>Ulmus carpinifolia</i>	668
<i>Acer</i>	667
<i>Salix aurita</i> :	669
<i>Quercus</i>	668
<i>Alnus incana</i>	667
<i>Betula pubescens</i>	665
<i>Salix alba</i> ' <i>Sericea</i> '	670
<i>Populus tremula</i>	668
<i>Ulmus glabra</i>	670
<i>Sorbus aucuparia</i>	669
<i>Salix sinerea</i>	670
<i>Populus</i>	670
<i>Tilia</i>	670
<i>Sorbus intermedia</i>	670
<i>Fagus silvatica</i>	670

A Canon EOS 5D Mark II digital SLR camera, paired with an EF 16–35 L USM II focal point, was used to photograph the specimens for this investigation. The images were acquired with the Canon EF 16–35mm f/2.8L USM II lens. A computer workstation with Intel® CORE™2 processor, 4 GB RAM, and a Windows 10 Professional (64-bit) operating system was utilized to perform this research. Imaging and feature extraction was carried out with the use of ImageJ (Schneider et al.2012), an open-source picture handling device, and VScode (Visual studio code).

Images of 15 photosynthetic eukaryotes specimens were used to study the impact of complex leaf shapes and architecture. Fifteen plant species were examined and analyzed. A selection of background photos, including other plants, dirt, and stones, was picked at random and included in the data set in addition to the extracted plants. A squared bounding box was used to retrieve

all of the samples. The photosynthetic eukaryotes were represented by 10,031 plant and soil pictures. Taking after the suggestions of Milioto et al. (2018), In order to convert the RGB pictures to excess green, they were transformed into each other and then into a grayscale image by applying the following Equation 1.

$$I_{ExG} = 2.0 \cdot I_G - I_R - I_B \quad (1)$$

The matrices $I_{\{R,G,B\}}$ represent the RGB picture channels and are the abundance of green pictures. In I_{ExG} , Terrain and stones seem to be black, while green pixels appear to be brilliant. This discussion is executed in OpenCV (Bradski, 2000).

Informatics and data processing

The models were chosen based on their performance quest to access, review and make some necessary comparisons on the existing behavioral technique for making a choice in recent surveys of CNN-based models. The study looks into plant identification using machine learning/deep learning. Adopting the proposal which states It is possible that the excellent prediction performance of CNNs is due to image areas that are confounded with the real picture label. In order to prevent this difficulty, the research was conducted as follows: (Wöber et al., 2021); how many studies have used the unsupervised Gaussian handle idle variable demonstrate to produce models that can be understood (Lawrence, 2005). The study also includes extraction execution of the Gaussian handle inactive variable demonstrated was compared to that of a modern profound learning-based comparable, specifically the profound convolutional encoding system. (Dong et al., 2018). Both models were utilized for the extraction of unsupervised plant features in this study. These characteristics were represented graphically and utilized for classification using a support vector machine. (Nayak et al., 2015).

The classification models described above were compared employing a five-fold cross approval procedure, each time using reshuffled picture data sets. This allowed us to make reliable comparisons between the models. For the sake of comparing the CNN and suggesting utilizing a five-fold validation, trials were conducted using the same randomized request of the pictures. This section discusses the models that were employed, as well as the hyperparameter tuning

process. The result of the gaussian process and other image processing techniques will be discussed in the resulting section.

Using the Gaussian Process Latent Variable Model as a starting point

The advice was implemented in order to improve upon the single-staged CNN model, according to the findings of Wöber et al. (2021), as well as the Bayesian Gaussian handle the inactive variable show, which was implemented as proposed in (Titsias et al., 2010) Incorporating an autonomous relevance determination algorithm into the radial base function kernel. The B-GP-LVM formulation (Lawrence, 2005) is a development of the initial GP-LVM detailing that makes use of variational inference (Blei et al., 2017). In this case, the GP-LVM is used to address the challenge of decreasing the D-dimensional matrix of $Y \in \mathbb{R}^{N \times D}$ to $X \in \mathbb{R}^{N \times d}$ Utilizing the probabilistic shown in Equation 2:

$$p(Y/X, \theta) = \prod_{n=1}^N p(Y[n, :] | X, \theta) = \prod_{n=1}^N N(Y[n, :] | 0, K_{NN} + \beta^{-1} I_N) \quad (2)$$

Inferring from equation (2), it can be shown that the kernel matrix influences a set of N Gaussian forms (GPs) mapping from an idle include space to the first picture space. The radial base function kernel (Bishop, 2006) was used in this investigation. The model q's parameter vector is denoted by

Data was used to make this discovery. Every picture in this model is represented as a vector $=Y[j, :]$ in the Y coordinate space.

The vector $=X[j, :]$ represents the extracted characteristics of the object under consideration. The factorised prior is used by the GP-LVM. The factorized approximation via the optimization method is used to solve the inference issue in this example (Titsias et al., 2010).

Then it is given as Equation 3,

$$F(Q) = \int Q(X) \log\left(\frac{p(X)p(X|Y)}{Q(X)}\right) dX \quad (3)$$

This approximation is based on the variational inference method developed by Blei and colleagues (2017). In comparison to Titsias (2009), this approximation relies on a negligible number of assistant focuses and makes use of sparse generalized parabolics (GPs). The features for this research are based on the desire the highlight conveyance, as given in condition 1. (2). The previous demonstration is based on the assurance of the measure of the idle space, as well as the number of assistants focuses used by the sparsed GPs, which are both given in the

previous section. This study, however, estimated both hyper parameters using the following procedure (Wöber et al., 2021):

1. To access the first estimate of the latent space's size, the study The principal component analysis (PCA) was used to examine the picture vector-matrix Y (Bishop, 2006). Initially, the number of main components explaining 78 percent of variations was utilised to make an educated approximation about d , the number of latent plant characteristics. In addition, the principal component analysis implementation from `sklearn` was employed in the investigation (Pedregosa et al., 2011).
2. In order to determine the ideal number of assistant focuses in a given extent of values, the researchers employed the greatest marginal log-likelihood method. $f_{55, 75, 100, 125, 150g}$ respectively. It also It was determined that the bare minimum of auxiliary points was employed above 95 percent of the most extreme negligible log probability.
3. In estimating the Latent dimension, the study estimated the latent dimensions using the above-mentioned in a range of $f_{15, 25, 55, 75, 100, \text{ and } 150g}$ latent dimensions, the optimal number of assistant focuses, and the most prominent negligible log probability was determined.

The estimated hyper-parameters were utilised to fit a B-GP-LVM using the estimated hyper-parameters. When implementing GPy with default settings, the research relied on the Gaussian Process Framework (GPy) implementation. (GPy, 2021).

Furthermore, and in expansion to the manual include examination carried out by (Wöber et al., 2021), the study chosen highlights for classification by evaluating their relevance values from the kernel (Raamussen et al., 2005) and noteworthiness from the p-values of the Spearman's rank relationship test. At first, the researchers selected the characteristics according to their importance, and then they calculated the increase in relevance for each feature. Uninformative features are those whose value falls below the median rise in significance over a certain period of time. It then sorts the features according to their relevancy and disregards any features that appear after the first uninformative feature does so.

Following that, we conducted a Spearman rank relationship test to see on the off chance that there's a statistically significant relationship between the feature j and the label for each feature. It is possible to reject the null hypothesis in the case when the p-value of a highlight is less than

or equal to 0.01. This test was carried out using the Spearman (2014) package under the cran R environment (RCore Team, 2021).

In addition to this report, the study, however, statistical feature research was used instead of human feature selection in this case as proposed by (Wöber et al., 2021). Finally, for feature visualization, the study also adopted the principles of (Wöber et al., 2021 and Wöber & Mohamed, 2019), using samples in order to view significant picture areas. The Gaussian distributed feature space was used to get this result. In order to do this, the researchers developed a vector for each inactive measurement and test within the ponder. There are a total of four entries in this vector, and each entry is set to the cruel of the anticipated values of the assessed highlights. In order to assess the feature dimension, the study created and changed the value of the entry to the actual expected value of the samples in the vectors. The collection of feature vectors for the plants that were developed was projected onto the picture space. Visualization and interpretation of the pixel-wise change of this projection are given, as well as the informative components recouped by the B-GP-LVM. Saliency maps are used to depict the relationships between the explanatory elements. It was determined that the significant picture areas of interest with p values less than 0.001 were obtained using the statistical test described in (Genaev and colleagues, 2021) for pixel-wise and p value-based threshold-holding of the saliency maps.

Convolutional Neural Networks and PCA

The PCA (principle component analysis) analyzes the observed data which is usually described by the input data. So, PCA is a dimensionality decrease procedure that changes a bunch of highlights in a dataset into fewer elements called principal components while simultaneously attempting to hold however much data in the first dataset as could reasonably be expected. The vital point of PCA is to lessen the quantity of highlights of an informational index, while protecting however much data as could reasonably be expected. The vectors are the principal components, but are not randomly chosen. The initial component chosen is orthogonal to the second, and it defines the number of variance left after the second principal component. The other principal component is computed so that it defines the number of variance in the original features. PCA permits the use of further steps and the data is represented as a linear combinations of principal components. Here the original data can be depicted as

vectors feature. Getting principal components is similar to the data a linear transformation from the feature1 x feature2 axis to a patch1 x patch2 axis.

CNN also known as ConvNet works on the basis of several hidden layers as proposed by Simonyan and Zisserman (Simonyan and Zisserman, 2015). The goal is to extract the important information from the data and to express this information as a set of new orthogonal output called the principal components. A completely associated layer of 1024 neurons with a ReLu actuation work was introduced to the VGG16.0 model utilised in this investigation. (Goodfellow et al., 2016), two neurons in a softmax output layer, then a dropout layer with a 50% dropout rate. One of the many ImageNets (Deng et al., 2009) initialization was the method that was employed. For a last step, Keras information enlargement was employed with the taking after parameters: rescale = 1/255, rotation range = 20, width shift range = 0.2, height shift range = 0.2, horizontal flip = True, and last but not least, fill mode = "nearest".

During the optimization process, the category cross-entropy was employed as a misfortune. Employing a clump estimate of 8, 50 ages and the Keras RMSProp optimizer, this loss was reduced to its minimum. (Keras, 2021). In addition, the research utilised the default settings for all other parameters, including the learning rate of the optimizer, which was set at 10⁻⁴. The extra green pictures were resized to [64x64x3] pixels and reshaped. This implementation was accepted and utilised in this investigation in accordance with the proposal in Keras (2021).

Hidden layers are partitioned into pooling layers, convolutional layers, classification layers, and fully connected layers. CNN can investigate the ailment dataset as pictures; during the pre-preparing stage, it utilizes ConvNet to examine the distinctive set of information, and utilizing a few channels, the ConvNet can catch the various elements of pictures. CNN is the blend of every one of these layers.

Assume f be a CNN with N layers sequential structure ($f_1; f_2; \dots; f_N$). A CNN comprises of multi-sub-testing, sub-inspecting, nonlinear, and altogether interconnected layers. Mappings will be carried out among input (w) and yield (u) as demonstrated in Equation (4) below:

$$u = f(w; X_1, X_2, X_3, \dots, X_N) = f_1(w; X_1) \text{ of } f_2(w; X_2) \dots \text{of } f_{N-1}(w; X_{N-1}) \text{ of } f_N(w; X_N) \quad (4)$$

Where X_N indicates inclination and weight vector for the n th layers f_N . f_N has been assigned to do spatial convolution or nonlinear enactment or classification. As indicated by the scope of η preparing information $\{(w(i), u(i))\}_{i=1}^{\eta}$, vectors $(X_1, X_2, X_3, \dots, X_N)$ can be decided as Equation (5):

$$\arg \min_{X_1, X_2, X_3, \dots, X_N} \frac{1}{\eta} \sum_{i=1}^{\eta} f_{Loss}(f(w^{(i)}; X_1, X_2, X_3, X_N), u^{(i)}) \quad (5)$$

Stochastic decrement and reverse engendering methodologies can achieve Equation (6). Where f_{Loss} infers loss work. The element map FM_m^h the Equation at the m level is depicted in Equation (6). In the calculation of a component map, convolutional layers commonly use convolutional channels.

$$FM_m^h = f(\alpha_m^h + \sum_j FM_j^{h-1} * G_{jm}^h) \quad (6)$$

Predispositions and bits individually are G_{jm}^h and α_m^h . FM_{in}^{h-1} and FM_{out}^h are a few qualities of input and output. Two segments make highlight maps for every convolution layer. The advantages of this technique are the capacity to gauge the picture size of the information and to make a positive distinction in neighboring districts. The nearby responsive territory is the main component, and shared weights are the next part. The subsequent Equation is utilized to decide the capacity as indicated in Equation (7)

$$\psi_j = \max(\psi_i^{n*n} z(n, n)) \quad (7)$$

Rectified Linear Units (ReLUs) will be utilized as an instrument for enactment and drop slope as indicated in Equation (8). This is the input picture; z shows the function of the window, and n recorded just the size of the input fix.

$$q(r) = \max(0, r) \quad (8)$$

Where q portrays the yield segment of the system with the r data, every layer has a similar size for information and yield. The choice tree additionally applies to choice hubs as inside branch hubs, indicating them with D . The dynamic component f_d has dispensed for every choice hub $0 \in Df_d(X; \Theta); X \rightarrow [0, 1]$. Additionally, hubs of expectation are proposed by P as terminal hubs. On the off chance that the reference $x \in X$ enters hub of judgment d , it will proliferate to one side or the left of the foundation, dependent on a $f_d(X; \Theta)$. The probability dissemination

π_p over Y is accessible in every $p \in P$ projection hub. The end-product for test x of the tree T with choices defined with swaying is indicated by the below Equation (9)

$$P_T[y | x, \Theta, \pi] = \sum_{p \in P} \pi_{py} \mu_p(X | \Theta) \quad (9)$$

At the point where $x \in X$, $\sum_p \mu_p(X | \Theta) = 1$. In this situation, π_{py} and $\pi = (\pi_p)_{p \in P}$ is a probability that the example would enter leaf p on class y and recognize by $\mu_p(X | \Theta)$. Choice hubs depend on the stochastic schedule and have been portrayed as Equation (10):

$$f_d(x; \Theta) = \sigma(f_r(x; \Theta)) \quad (10)$$

The sigmoid capacity $\sigma(x)$ for this situation is set to $\sigma(x) = \frac{1}{(1+e^{-x})}$ the choice timberland is known as a gathering of dynamic trees and is characterized by the subsequent Equation (11)

$$F = \{T_1, T_2, \dots, T_z\} \quad (11)$$

Let I be $a = I_1, I_2, I_3, \dots, I_Q$, where Q shows a bunch of pixels and I_Q implies the size of the dark degree of a pixel L , $K = (K_1, K_2, K_3, \dots, K_Q$ where $K_Q \in LL = \{0, 1\}$ can be summed up to a set with positive names as described in Equation (12).

$$K^* = \arg \min_k \{Y(I | K, \Theta) Y(K)\} \quad (12)$$

$Y(K)$ is the dissemination of Gibbs. Equation (13) can be composed as in the Expect-Maximization computation.

$$K^* = \arg \min_{K \in k} \{U(I | K, \Theta) U(K)\} \quad (13)$$

Here U relates to urinary potential or energy of possibility and is shown by Equation (14)

$$U(I | K, \Theta) = \sum_Q \left[\frac{(I_Q - \mu_{KQ})^2}{2\sigma_K^2} + \ln \sigma_K \right] \quad (14)$$

This hypothesis cannot show occasions, all things considered. (Gamma Mixture Model) GMM is, subsequently, the ideal decision for engineers for dynamic conveyance. As per the hypothesis, it expects a Gaussian appropriation with boundaries $\sigma_{xi} = \mu_{xi}$, σ_{xi} follows the divided region strength. The following are the counts of a GMM with c components as described in Equation (15)

$$\sigma_i = \{(\mu_{i,1}, \sigma_{i,1}, W_{i,1}), \dots, (\mu_{i,c}, \sigma_{i,c}, W_{i,c})\} \quad (15)$$

The Gaussian boundary appropriation can be shown as Equation (16)

$$G(Z, \alpha_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(Z-\mu_i)^2}{2\sigma_i^2}\right) \quad (16)$$

To get the assessed likelihood by consolidating Equation (16) with Equation (17) and it is portrayed below as

$$G_{mix}(Z, \alpha_i) = \sum_{c=1}^h W_{i,c} G(Z, \mu_{i,c}, \sigma_{i,c}) \quad (17)$$

The pixel force is a three-dimensional vector for an RGB signal. Thus, the GMM has its standards, and it is depicted in Equation (3.18). The pixel force is a three-dimensional vector for an RGB signal.

$$\alpha_{xi} = (Z, \alpha_i) = (\mu_{i,1}, \sigma_{i,1}, W_{i,1}) \dots (\mu_{i,c}, \sigma_{i,c}, W_{i,c}) \quad (18)$$

As indicated by Equation (3.14) with Equation (3.18), the energy list is calculated below as in Equation (3.19),

$$U(I | K, \Theta) = \sum_Q \left[\frac{1}{2} (I_Q - \mu_{KQ})^T \Sigma_{KQ}^{-1} (I_Q - \mu_{KQ}) + \ln |\Sigma_{KQ}| + \frac{1}{2} \right] \quad (19)$$

Model Evaluation Using Statistical Methods

The layer-wise engendering of pertinence (LPR) algorithm introduced by Bach et al. (2015), as well as idle representations of input characteristics such as the generalized partial latent variable model (GP-LVM), are used. (Lawrence, 2004; Lawrence, 2005 and Titsias & Lawrence, 2010) and cAE provided the relevant metrics which suitably coded the significance of input characteristics in the decision-making handle of machine learning algorithms is discussed in detail below. LPR, GradCAM, and GP-LVM variance characteristics are often shown in so-called saliency maps in image analysis, and this is a typical technique. Even if raw metric data are beneficial for identifying critical picture areas, displaying them as pictures place the weight of elucidation on the individual utilizing the calculations.

Lawrence et al. (2021) offered an orderly examination of saliency maps combined with a centrality test in order to strengthen the objectivity in determining the most relevant elements.

According to Lapuschkin et al. (2019), saliency maps were utilised for studies similar to those discovered in Montavon et al. (2018a) and Wöber et al. (2021). These saliency maps share a few properties with each other. First and first, saliency maps are sparse, and second, relevant elements emerge in saliency maps as connected zones with positive metric values, which indicates that they are significant. When attempting an objective recognizable proof of significant locations in saliency maps, it is necessary to articulate this finding numerically. Following the logic of (Lawrence et al., 2021), features appear as stamped spatial point forms, while irrelevant areas are defined by the invalid theory that metric values are inferred from a checked homogenous spatial Poisson handle. The null hypothesis is tested against a one-sided alternative in which the metric values are taken from a designated generic spatial point process. The result is used to determine whether an outline area has a place to include or not.

These assumptions enable (Montavon et al., 2018) to use a nonparametric approach to get site-specific p-values, which are described in more detail below. The saliency map is smoothed by convolving with a k dimensional Gaussian part, resulting in a smoothed saliency outline esteem $s(x,y)_{alt}$ for each position on the saliency map. $S(x, y)_{null}$ was used to create samples for the null hypothesis, which was achieved by randomising the placements of saliency outline sections and smoothing with the same part to create $s(x, y)_{null}$. A positive result in support of the invalid speculation versus the one-sided elective happens if the $s(x,y)_{alt}$ function returns the value $s(x,y)_{null}$. Through the use of N repetitions of the randomization process, as well as smoothing and comparison of data, we may get the value $n(x,y)_{null}$, which represents the number of cases that bolster the invalid speculation. Using an appropriately large number of samples, p-values for particular locations (psm) may be calculated (x, y) . When $psm(x, y)$ is set to a sufficiently modest value ($p_{thrs} = a$), the positions of essential saliency map features may be reliably determined in a repeatable way. In this investigation, the numbers $N = 104$, $k = 5$, and $a = 0.001$ are used.

Latent Characteristics

For categorization, the evaluated characteristics of the cAE and the B-GP-LVM were employed in conjunction with each other. N support vector machines, together with the RBF kernel, are used to accomplish the classification process (Lapuschkin et al., 2019). In a manner comparable

to the CNN pipeline, the analysts utilized ten emphases of five-fold cross approval based on reshuffled information sets. A global Bayesian optimization approach (Genaev et al., 2021) is used to optimise the hyperparameters of the part work and the SVM, which relies on a total of 10 initialization points and ten optimization points. This improvement is accomplished by the use of (Pedregosa et al., 2011). Prediction probabilities were obtained by the usage of the logistic regression-based extension, which was based on the e1071 R SVM implementation described in (Genaev et al., 2021).

CHAPTER IV

Result Presentation and Analysis

We first outline the model processing technique for the image dataset before presenting our findings. The forecast performance is then displayed.

Model Processing

According to Rasmussen and Williams (2006), Gaussian processes (GPs) are useful for image manipulation to obtain more accurate results. For inference of extremely nonlinear latent functions from observable data, GPs provide a Bayesian nonparametric framework. They've gained a lot of attraction in machine learning for challenges like nonlinear regression and classification. The Gaussian process can be applied to process the image in order to get a clearer image. Several other techniques were also applied, like image cropping. The image dataset was of different dimensions and scales, so each image was resized to get a better image process and to produce a better result. The results gotten will be contrasted with various assessment metrics to show how efficient our model is. VScode was used, which comprises different libraries for this task. As follows, the out of the models will be explained in the resulting segment. Each model utilized a comparative area of instructive assortments. In Figure 1, we depict a segment of the raw utilized data. This imaging data shows the need for processing to be done in order to get the right data. Figure 2 shows that the raw image was later processed in order to get a better result. Figure 3 shows the image after Gaussian processing; this method helps the proposed technique to work accurately.

Figure 1

Depicting the raw data before cropping



Figure 2

Depicting the data after cropping

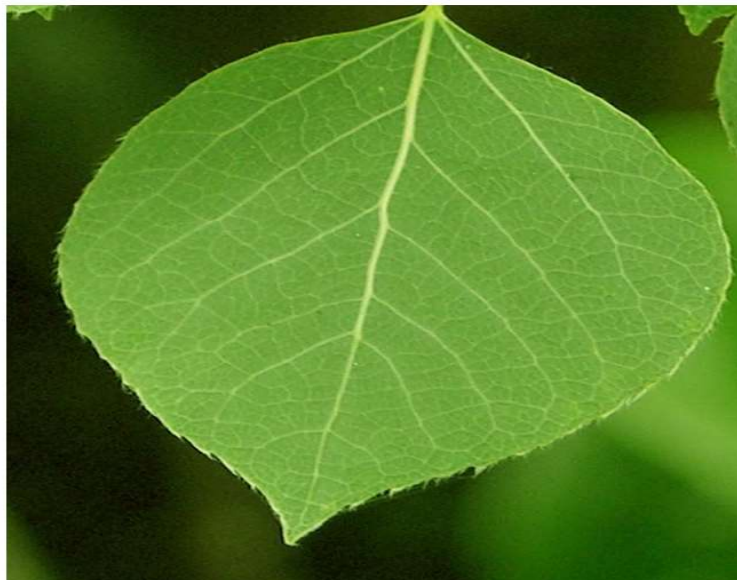
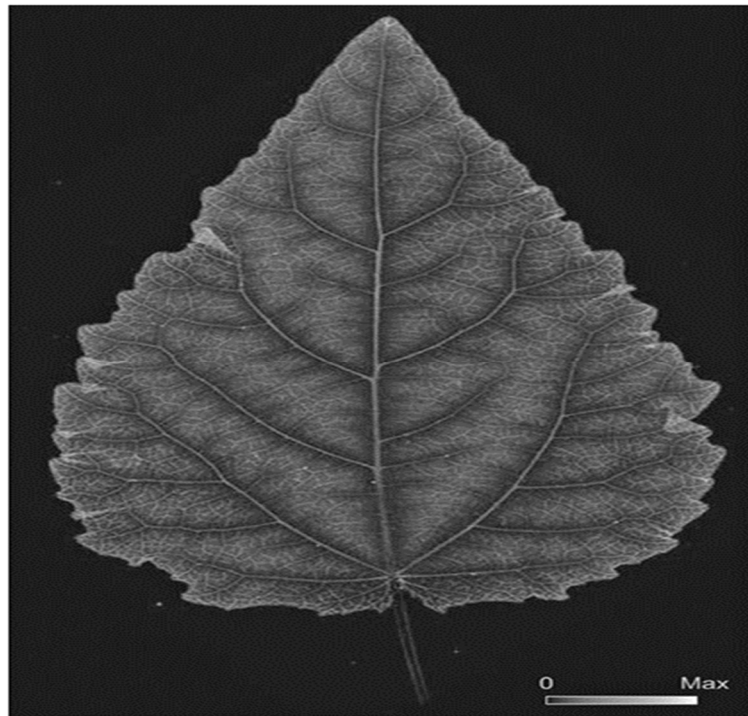


Figure 3

Gaussian distortion for greyscale imaging



Model Performance

The model architecture is given in Figure 4. The model is split in an 80 to 20 test-split ratio after several trials. The data added up to 10,031 images of 15 different plant examples, and the test split was applied to it. To process the given attributes, the experiment was done on the normal CNN technique; afterward, a PCA was approached in order to get better accuracy. The ReLU activation function was used, and the application of adam optimizer was also applied. The result of using Relu as a faster training approach than others. The cross-entropy approach was also applied to get a lesser loss function. Figure 5 shows the activation map of some data due to the initialization of random weights. This depicts exactly how the filter works after implementation. For the application of the PCA and the CCN, convolutional weights are changed, but the other settings are the same. The weights were changed using an initializer custom function.

Figure 4

The CNN architecture

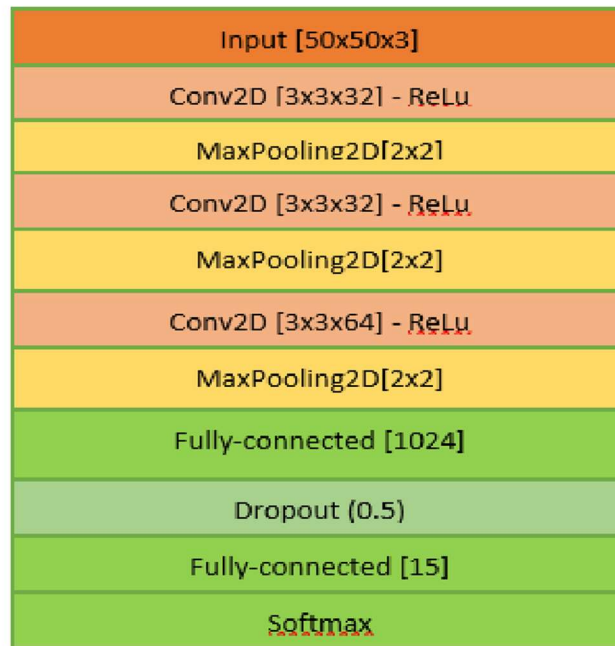


Figure 5

Activation maps of the data

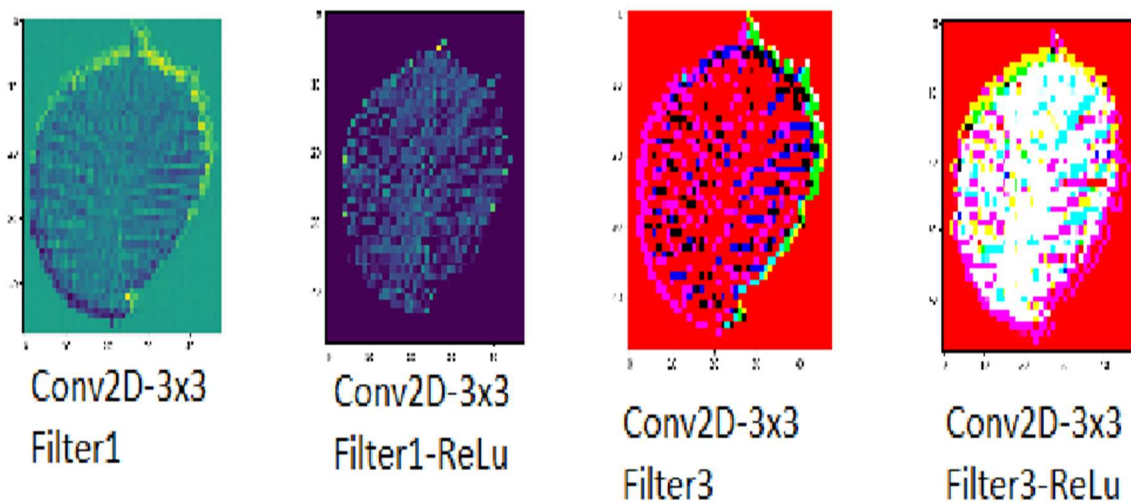


Figure 6 shows the reshaped size of the data after applying the PCA. The PCA weights are reshaped to give basic image patches. Utilizing all the patches propels a variance of up to 100%.

The reduced number of patches is given by the graph as shown in Figure 7; it depicts the captured patches of up to 90% of the variance. After various experiments depicting the number of utilized filters and adjusted kernels, the application of the PCA for optimizing the results shows how the features are correlated. The intensity of the color shows how close the images are and gives an insight into the appropriate weights to be utilized than randomizing the weights.

Figure 6

The patches from the PCA

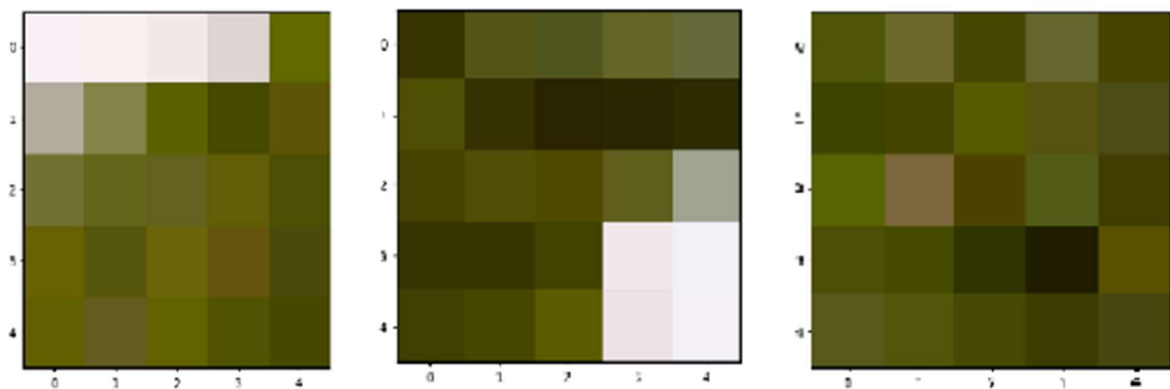
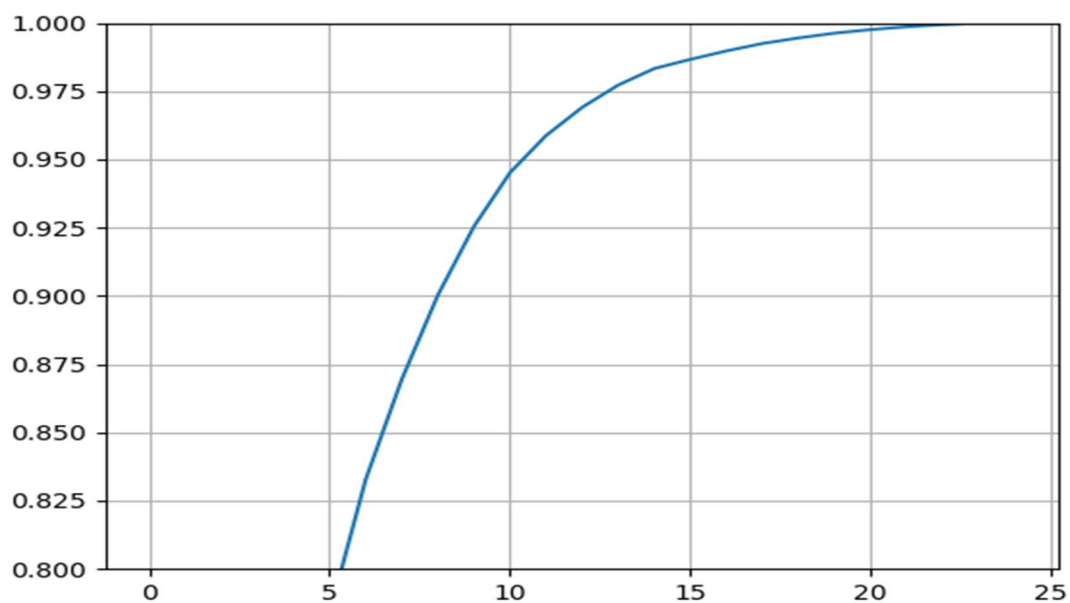


Figure 7

Patches graph from the PCA



The prediction accuracy is one solid parameter to show how accurate the model is. The result on various parameters is given in Table 3. The accuracy of the model is given at 0.89%, the loss function is given at 0.48%, F1-score 0.92%, Recall 0.73% and Precision 0.86% for CNN with PCA. On the other hand, the research found biologically incomprehensible justifications for CNN's decision. These uninterpretable areas could be at fault for CNN's significant categorizing contrast. This exhibits that these picture locales were utilized to accomplish high forecast accuracy during the training process. These discoveries show that pixels from different plants are utilized to accurately recognize the pictures data. Furthermore, the data imply that the CNN's result of the accuracy categorization is based on visual areas associated with picture labeling and identification (Demilew, F. & Sekeroglu. B. 2019).

Unsupervised learning is used in the suggested methodology to try to solve this challenge by extracting features using statistical analysis. In the initial stage, image processing analysis isolates picture areas with high inconstancy and chooses feature highlights. It makes saliency maps by examining the component space and assessing change in the picture space. The study, however, offers to visualize the learned process after picking attributes using a Spearman's rank correlation test. For further justification, the experimental epochs are given below, and the accuracy is in Figure 8.

Test accuracy: 0.8984771370887756 Test loss: 0.48781031370162964

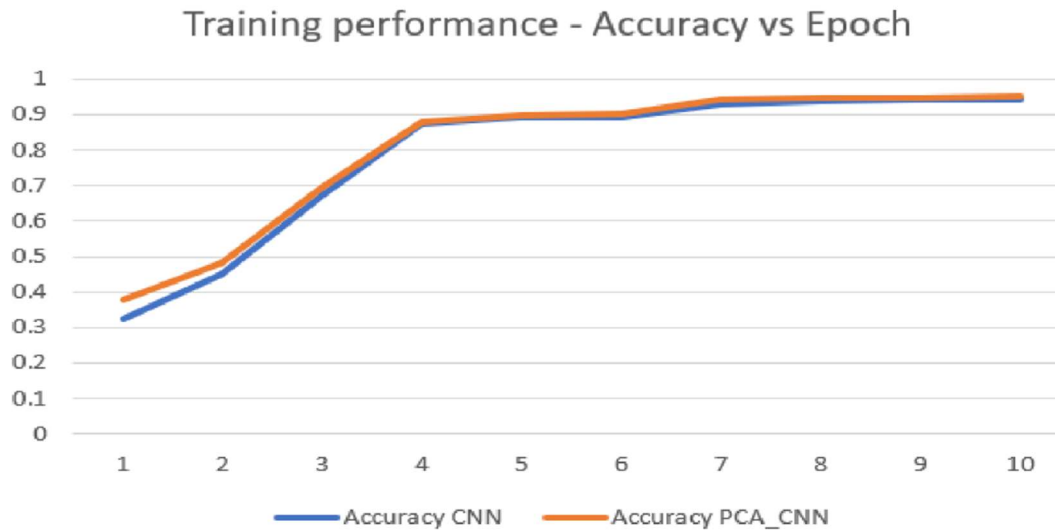
Table 3.

The resulting experiment of the PCA with CNN (%)

	F1-score	Precision	Recall	Epochs
	0.92	0.86	0.73	20
Weighted average	0.89	0.91	0.89	20
Macro average	0.88	0.92	0.86	20
Accuracy	0.89			20

Figure 8

Accuracy of the CNN vs. CNN with PCA for ten epochs



Discussion of Results

The study's findings show that the PCA with CNN outperforms other contrasted techniques in terms of prediction and identification of accuracy and F1 score got from (Montavon et al., 2018; Bach et al., 2015). Nonetheless, there was no significant variation in CNN prediction performance for the data set. Concerning the utilized data, the difference in error rate prediction was significant. The big disparity is considered to be related to the leaf structure and the best CNN architecture for pictures. Unfortunately, we noticed that the PCA with CNN gained higher accuracy by utilizing image sections that biological systems could not understand and optimizing it by using various patches. Similar to prior findings for animal photos, the study suggests that the PCA with CNN's higher prediction accuracy is due to the clever-Hans mode of predictors (Montavon et al., 2018, Bach et al., 2015), as clever Hans states that the learning strategy must be solid and there must be no bias. In this instance, the image areas that are related to the image label account for the high PCA with CNN prediction accuracy. The effect was identified in photographs when pixels near the plant's real location were used to categorize it.

Unsupervised techniques are known to be able to extract image regions that are interpretable, unlike CNN, which requires a per-sample model explanation. We were able to discover statistically viable properties for these models using Spearman's rank correlation test. Our findings suggest that the PCA with CNN architecture is capable of recognizing complex coherences in visual material. Regardless, assessing an overall assertion for the hidden explanatory elements is trying because of the sample-based decision. The suggested unsupervised learning models, on the other hand, generate saliency maps that depict the classification zones. Unlike CNN, the total explanatory parameters required for categorization are known beforehand and apparent. Features that aren't necessary may be eliminated.

Utilizing an informational index with 15 species, the result depicted that the illustrative pieces of the CNN choice in certain circumstances were uninterpretable. This is particularly significant while deep learning/machine learning is utilized for factual examination and oddity recognizable proof as opposed to in-field application arrangement. The utilization of an unsupervised process, then again, prompts the plant to distinguish methods having reasonable components. In any case, our discoveries show that the PCA with CNN utilized produces ends in view of biologically significant image districts. These explainable qualities are embedded in the unsupervised process and are related to the image label in a substantial way. Individual predictions are used in deep learning algorithms to illustrate their explainability (Montavon et al., 2018; Bach et al., 2015).

We conclude that PCA with CNN should be utilized with caution, especially when the biological context is important. Unlike other plant categorization algorithms (Bach et al., 2015; Bergstra et al., 2011), our method selects regions of interest initially from the pictures. Our method enables biological interpretation, which has long been regarded as a critical part of plant identification. The approaches available, however, are confined to classification. Recent deep learning architectures, on the other hand, can be utilized for more complicated tasks, just like pixel-wise flower segmentation. Where logical qualities are a higher priority than expectation precision, however, the provided method could be used instead of deep learning classification applications. The comparison table of our model with others is given below in Table 4.

Table 4.

Experimental result of the used model with others (%)

Authors	Recall	Precision	F1 score	Accuracy
Arad et al. (2020)	-	-	-	0.81
Yang & Wei (2019)	-	0.86	0.85	0.86
Kierdorf et al. (2019)	-	-	-	0.85
Weyler et al (2021)	0.72	0.86	0.85	0.87
Duong et al. (2019)	0.71	0.83	0.81	0.86
This thesis	0.73	0.86	0.92	0.89

CHAPTER V

Conclusion And Recommendation

Conclusion

The study utilized the convolutional neural networks to classify and identify images based on several features that could not be understood biologically in order to study plant identification through deep learning. Furthermore, the study believed that several learning techniques like the unsupervised technique perform comparably to supervising learning-based methodologies while avoiding the black-box model problem. Images of plant species from photosynthetic eukaryotes were acquired and used in this project from the Swedish Botanic Garden. To analyze and categorize various plant species adequately. We looked at prediction accuracy and mutual information in our experiments. In addition, the PCA process was utilized to compare the performance for prediction of the suggested unsupervised technique to that of the chosen CNN-based models that can be optimized to perform profoundly. Using unsupervised machine learning models, we retrieved and displayed characteristics. The characteristics were represented using the methods of (Montavon et al., 2018).

Finally, the study looked at the presentation of the utilized classifiers regarding prediction and identification. When it comes to accuracy prediction and similar information, PCA with CNN surpasses the other techniques. The PCA test, however, revealed significant variations in performance for the data set as its accuracy was increased to 89%. However, the study finds that CNNs must be used with caution as it requires an efficient learning strategy in order to procure a good result. Unsupervised learning-based techniques, in which all employed image areas can be observed before categorizing, were found to be capable of solving the classification problem but performed poorly in terms of prediction accuracy. In any case, we can be sure that the unsupervised process utilizes no elements that are baffling.

Recommendations

Through convolutional neural networks, automated plant disease detection systems offer the categorization and identification of diseases in plants, combining the expertise of

phytopathologists with the ability to extract symptomatic information. It is challenging to semantically classify the data in representative sets with a sufficient number of labeled samples due to the diversity of difficulties and the specificities of real-world settings. This issue becomes a relevant and difficult obstacle in making machine learning approaches more practical. Our findings show that the application of CNN in plant disease prediction has progressed significantly. On this basis, we recommend that traditional structures be paired with optimization and customization methods, notwithstanding the complexity of the data set, which is constituted of photos recorded in real-world crop situations, to achieve appropriate accuracy.

The popularity of possible approaches that suggest new CNN structures based on the process of recognizing plant illnesses is increasing, despite the fact that they have poorer accuracy than methods that employ classic designs. However, we recommend that crops with significant food and financial representativeness, such as grains and cereals, be properly examined rather than being disregarded by methodologies. We also call attention to the small number of methods for identifying or classifying diseases caused by worms. The analysis of the works presented and the challenges they propose to answer, as well as new developments in convolutional neural network architecture, with the goal of identifying and classifying plant illnesses. Based on this, the study suggests that some techniques, such as multispectral and hyperspectral pictures, can lead to novel agricultural solutions that increase productivity while reducing the harm caused by biotic and abiotic agents.

Finally, on the grounds that safeguarding crops in natural cultivating is a troublesome process, the review recommends having a full comprehension of the harvest being cultivated as well as possible vermin, microorganisms, and weeds. The previously mentioned framework can be extended to a constant video passage framework, taking into consideration unattended plant care. Another profound learning model can be deduced in view of a unique structural convolution network that was built in this work to recognize plant ailments utilizing photographs of various plant leaves. Plant management has been displayed in investigations and research to upgrade yields by up to half. A system that treats and analyzes plant components is one more component that can be acquainted for future research.

References

- Adedoja, A., Owolawi, P.A., & Mapayi, T. (2019). Deep learning based on NASNet for plant disease recognition using leave images. *In Proceedings of the 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Winterton, South Africa, 5–6*, pp. 1–5.
- Adweb, K. M. A., Cavus N., & Sekeroglu. B. (2021). Cervical Cancer Diagnosis Using Very Deep Networks Over Different Activation Functions, *IEEE Access*, vol. 9, pp. 46612–46625, 2021, doi: 10.1109/ACCESS.2021.3067195.
- Amala Sabu, & Sreekumar K, (2017). Literature Review of Image Features and Classifiers Used in Leaf Based Plant Recognition Through Image Analysis Approach, *International Conference on Inventive Communication and Computational Technologies(ICICCT)*, DOI: 10.1109/ICICCT.2017.7975176.
- Anders, C.J., Weber, L., Neumann, D., Samek, W., Müller, K.R., & Lapuschkin, S. (2022). Finding and removing Clever Hans: Using explanation methods to debug and improve deep models. *Inf. Fusion*, 77, 261–295. [CrossRef]
- Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., & Tielen, T. (2020). Development of a sweet pepper harvesting robot. *Journal of Field Robotics*, 37(6):1027–1039.
- Babu, M.P., & Rao, B.S. (2021). Leaves Recognition Using Back Propagation Neural Network-Advice for Pest and Disease Control on Crops. Available online: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.3632&rep=rep1&type=pdf>.

- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W. (2015). On Pixel-Wise Explanations for Nonlinear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS ONE*, 10, e0130140. [[CrossRef](#)]
- Bai, X., Cao, Z., Zhao, L., Zhang, J., Lv, C.; Li, C., & Xie, J. (2018). Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. *Agric. For. Meteorol.*, 259, 260–270. [[CrossRef](#)]
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35, 1798–1828.
- Camargo, A., & Smith, J. (2009). An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosyst. Eng.*, 102, 9–21.
- Camargo, A. & Smith, J. (2009). Image pattern classification for the identification of disease causing agents in plants. *Comput. Electron. Agric.*, 66, 121–125.
- Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanekaran, Y.A. (2020). Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.*, 173, 105393.
- Chouhan, SS, Kaul, A., Singh, U.P., & Jain, S. (2018). Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology. *IEEE Access*, 6, 8852–8863. [[CrossRef](#)]
- Demilew, F. & Sekeroglu. B. (2019). Ancient Geez script recognition using deep learning. *SN Applied Sciences* Vol.1(11): 1-7. DOI: 10.1007/s42452-019-1340-4
- Dogruyol K., & Sekeroglu B., (2019). Absenteeism Prediction: A Comparative Study Using Machine Learning Models", In: Aliev R., Kacprzyk J., Pedrycz W., Jamshidi M., Babanli M., Sadikoglu F. (eds) 10th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions – ICSCCW

2019. ICSCCW 2019. *Advances in Intelligent Systems and Computing*, vol 1095. Springer, Cham DOI: https://doi.org/10.1007/978-3-030-35249-3_94
- Duong-Trung, N., Quach, L.-D., Nguyen, M.-H., & Nguyen, C.-N. (2019). Classification of grain discoloration via transfer learning and convolutional neural networks. *In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, pages 27–32.
- Ferentinos, K.P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.*, 145, 311–318.
- Geetharamani, G., & Pandian, A. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.*, 76, 323–338.
- Ghazi, M.M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing*, 235, 228–235. [[CrossRef](#)]
- Fu, H. Chi, Z. Chang, J. & Fu, C. (2003). Extraction of leaf vein features based on artificial neural network—Studies on the living plant identification I, *Chinese Bulletin of Botany*, vol. 21, pp. 429–436.
- Gaber, T. Tharwat, A. Snasel, V. & Hassanien A. E. (2015). Plant Identification: Two Dimensional-Based Vs One Dimensional- Based Feature Extraction Methods", *International Conference on Soft Computing Models in Industrial and Environmental Applications*, Springer, Cham, DOI: <http://doi-org443.webvpn.fjmu.edu.cn/10.1007/978-3-319-19719-733>.
- Goeau, H. Bonnet, P. & Joly, A. (2015). LifeCLEF plant identification task 2015," in *Proceedings of the Conference and Labs of the Evaluation Forum (CLEF' 15)*.

- Goeau, H., Bonnet, P. & Joly, A. (2016). Plant identification in an open-world (lifeclef 2016)," in *Proceedings of the CLEF working notes*, vol. 2016,
- Hulya Yalcin, & Salar Razavi, (2016). Plant Classification using Convolutional Neural Networks, *IEEE International Conference on Agro-Geoinformatics*, Tianjin, China, DOI: 10.1109/ Agro- Geoinformatics.2016.7577698.
- Jadhav, SB, Udipi, VR, & Patil, S.B. (2020). Identification of plant diseases using convolutional neural networks. *Int. J. Inf. Technol.*, 1–10. [[CrossRef](#)]
- Jassmann, T. J. Tashakkori, R. & Parry, R. M. (2015). Leaf classification utilizing a convolutional neural network, *IEEE Southeastcon, Fort Lauderdale, FL, USA*, DOI: 10.1109/ SECON.2015.7132978.
- Joly, A., Goeau, H. & Bonnet P., (2014). Interactive plant identification based on social image data, *Ecological Informatics*, vol. 23, pp. 22–34.
- Kamilaris, A., & Prenafeta-Boldú, F.X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric.*, 147, 70–90. [[CrossRef](#)]
- Kaur, S., Pandey, S., & Goel, S. (2018). Semi-automatic Leaf Disease Detection and Classification System for Soybean Culture. *IET Image Process.*, 12, 1038–1048. [[CrossRef](#)]
- Kierdorf, J., Zabawa, L., Lucks, L., Klingbeil, L., & Kuhlmann, H. (2019). Detection and counting of wheat ears by means of ground-based image acquisition. *Bornimer Agrartechnische Berichte*, (102):158–167.
- Koh, P.W., & Liang, P. Understanding Black-box Predictions via Influence Functions. *In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*, 6–11 August 2017; Volume 70, pp. 1885–1894.

- Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K.R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nat. Commun.*, 10, 1096. [[CrossRef](#)] [[PubMed](#)]
- Lee, S.H., Goëau, H., Bonnet, P., & Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. *Comput. Electron. Agric.*, 170, 105220. [[CrossRef](#)]
- Li, Y., Nie, J., & Chao, X. (2020). Do we really need deep CNN for plant diseases identification? *Comput. Electron. Agric.*, 178, 105803. [[CrossRef](#)]
- Li, Y. Zhu, Q. Cao, Y. & Wang, C. (2005). A leaf vein extraction method based on snakes technique," in *Proceedings of the International Conference on Neural Networks and Brain (ICNN&B' 05)*, pp. 885–888.
- Marcus, G. (2018). Deep Learning: A Critical Appraisal. arXiv, arXiv:1801.00631.
- Mohanty, S.P., Hughes, D.P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.*, 7, 1419. [[CrossRef](#)]
- Mokhtar, U., Ali, M.A., Hassanien, A.E., & Hefny, H. (2015). Identifying two of tomatoes leaf viruses using support vector machine. In *Information Systems Design and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, pp. 771–782.
- Montavon, G., Samek, W., & Müller, K.R. (2018). Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, 73, 1–15. [[CrossRef](#)]
- Oyewola, D.O., Dada, E.G., Misra, S., & Damaševičius, R. (2021). Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. *PeerJ Comput. Sci.*, 7, e352. [[CrossRef](#)] [[PubMed](#)]

- Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D.P. (2017). Deep Learning for Image-Based Cassava Disease Detection. *Front. Plant Sci.*, 8, 1852.
- Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinyi, N., Mrisho, L., Ndalaha, M., Legg, J., & Hughes, D.P. (2019). A mobile-based deep learning model for cassava disease diagnosis. *Front. Plant Sci.*, 10, 272.
- Rangarajan Aravind, K., & Raja, P. (2020). Automated disease classification in (Selected) agricultural crops using transfer learning. *Autom. Casopis Autom. Mjer. Elektron. Računarstvo Komun.*, 61, 260–272.
- Rangarajan, AK, & Purushothaman, R. (2020). Disease classification in eggplant using pre-trained vgg16 and MSVM. *Sci. Rep.*, 10, 1–11.
- Samek, W., Wiegand, T., & Müller, K.R. (2018). Explainable Artificial Intelligence: Understanding, Visualizing, and Interpreting Deep Learning Models. *ITU J. ICT Discov.*, 1, 49–58.
- Sapna Sharma, & Chitvan Gupta. (2015). A Review of Plant Recognition Methods and Algorithms, *IJIRAE - International Journal of Innovative Research in Advanced Engineering*, Vol. 2, Issue no. 6.
- Sekeroglu, B. (2004). Classification of Sonar Images Using Back Propagation Neural Network, *IEEE IGARSS*, (5), 3092-3095.
- Sekeroglu, B. & Tuncal. K. (2021). Prediction of cancer incidence rates for the European continent using machine learning models, *Health Informatics Journal*, <https://doi.org/10.1177/1460458220983878>, 2021.
- Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. arXiv **2016**, arXiv:1610.02391.

- Sethy, P.K., Barpanda, N.K., Rath, A.K., & Behera, SK. (2020). Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.*, 175, 105527.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv, arXiv:1312.6034
- Singh, J. & Kaur, H. (2018). Plant disease detection based on region-based segmentation and KNN classifier. In *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018, Palladam, India*, 16–17; pp. 1667–1675.
- Titsias, M.K., & Lawrence, N.D. (2010). Bayesian Gaussian Process Latent Variable Model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Chia Laguna Resort, Sardinia, Italy*, 13–15, pp. 844–851.
- Weyler, J., Milioto, A., Falck, T., Behley, J., & Stachniss, C. (2021). Joint plant instance detection and leaf count estimation for in-field plant phenotyping. *IEEE Robotics and Automation Letters*, 6(2):3599–3606.
- Wöber, W., Curto, M., Tibihika, P., Meulenbroek, P., Alemayehu, E., Mehnen, L., Meimberg, H.; & Sykacek, P. (2021). Identifying geographically differentiated features of Ethiopian Nile tilapia (*Oreochromis niloticus*) morphology with machine learning. *PLoS ONE*, 16, e0249593.
- Yang, Chengzhuan & Wei, Hui. (2019). Plant Species Recognition Using Triangle-Distance Representation. *IEEE Access*. 7. 178108-178120. 10.1109/ACCESS.2019.2958416.
- Zeng, W., & Li, M. (2020). Crop leaf disease recognition based on Self-Attention convolutional neural network. *Comput. Electron. Agric.*, 172, 105341. [[CrossRef](#)]

Appendices A

Source Codes

- CCN module

```

#Importing all the required libraries for this project
import pandas as pd
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split

# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras

import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import KFold
from sklearn.metrics import log_loss
import os
import sys
import pickle
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras import backend as K
from keras.utils.vis_utils import plot_model
from keras.utils import np_utils
from sklearn.decomposition import PCA
from sklearn.feature_extraction.image import extract_patches_2d

K.set_image_dim_ordering('tf')
dir_prefix = "../data/all_data/"
data_file_dir = ["leaf1", "leaf2", "leaf3", "leaf4", "leaf5", "leaf6", "leaf7", "leaf8",
"leaf9", "leaf10", "leaf11", "leaf12", "leaf13", "leaf14", "leaf15"]
#classes_dict = {"leaf1" : "leaf2", "leaf3", "leaf4", "leaf5", "leaf6", "leaf7", "leaf8",
"leaf9", "leaf10", "leaf11", "leaf12", "leaf13", "leaf14", "leaf15"}
test_split = 0.35

```



```

n_folds = 5 #k-fold cross validation
pca_weights = None

def read_preprocess_data():
    X_data = []
    X_classes = []
    print("Reading Input Data: ")
    i = 0
    for dir in data_file_dir:
        for file in os.listdir(dir_prefix + dir):
            img = Image.open(dir_prefix + dir + "/" + file)
            width, height = img.size # Get dimensions
            left = width/10
            top = height/10
            right = 9 * width/10
            bottom = 9 * height/10
            #img = img.crop((left, top, right, bottom))
            img = img.resize((50,50))
            arr = np.array(img)
            #normalizing the pixel values to the scale [0,1]
            arr = arr / 255
            X_data.append(arr)
            #cropping 'leaf1' to int('leaf1')
            X_classes.append(i)
            sys.stdout.write("->")
            sys.stdout.flush()
            i = i + 1
    X_data = np.array(X_data)
    X_classes = np.array(X_classes)
    print("len(X_data): ", len(X_data))
    print("X_data.shape: ", X_data.shape)
    train_data, test_data, train_classes, test_classes = train_test_split(X_data,
X_classes, test_size = test_split, random_state=42)
    print("Train Data, Test Data, Train Classes, Test Classes:")
    print(type(train_data), type(test_data), type(train_classes), type(test_classes))
    print(len(train_data), len(test_data), len(train_classes), len(test_classes))
    print(train_data.shape, test_data.shape, train_classes.shape, test_classes.shape)
    print("Test Classes: ", test_classes)
    return train_data, test_data, train_classes, test_classes

def create_model():
    model = Sequential()
    #model.add(Conv2D(25, (5, 5), input_shape=(50, 50, 3)))
    model.add(Conv2D(25, (5, 5), input_shape=(50, 50, 3)))

```

```

#print("**** Model Initialized weights: ", len(model.get_weights()),
model.get_weights()[0].shape)
#print(model.get_weights())
#model.add(Conv2D(25, (5, 5), input_shape=(50, 50, 3), kernel_initializer=my_init))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

#model.add(Conv2D(32, (3, 3)))
#model.add(Activation('relu'))
#model.add(MaxPooling2D(pool_size=(2, 2)))

#model.add(Conv2D(64, (3, 3)))
#model.add(Activation('relu'))
#model.add(MaxPooling2D(pool_size=(2, 2)))
# the model so far outputs 3D feature maps (height, width, features)
model.add(Flatten()) # this converts our 3D feature maps to 1D feature vectors
model.add(Dense(1024))
model.add(Activation('relu'))
#model.add(Dropout(0.5))
model.add(Dense(15))
model.add(Activation('softmax'))
plot_model(model, to_file='model_plot.png', show_shapes=True,
show_layer_names=True)
model.compile(loss='categorical_crossentropy',
              optimizer = tf.train.AdamOptimizer(),
              metrics=['categorical_accuracy'])
model.summary()
'''
model = VGG16()
model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])
'''
return model

def train_evaluate_model(model, train_data, train_classes, validate_data,
validate_classes):
    model.fit(train_data, train_classes, epochs=10)
    test_loss, test_acc = model.evaluate(validate_data, validate_classes)
    predictions = model.predict(validate_data)
    print('Test accuracy:', test_acc, "Test loss: ", test_loss)
    print("Validate Classes, Predictions:")
    #print(validate_classes, predictions)

```

```

print("Length of predictions: ", len(predictions), "Length of validate classes: ",
len(validate_classes))
#llog_loss = log_loss(validate_classes, predictions)
#print("Log Loss: ", llog_loss)

def plot_image(i, predictions_array, true_label, img):
    #predictions_array, true_label, img = predictions_array[i], true_label[i], img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

    plt.imshow(img, cmap=plt.cm.binary)

    predicted_label = np.argmax(predictions_array)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'red'

    """
    plt.xlabel("{} {:.2f}% ({}).format(class_names[predicted_label],
                                     100*np.max(predictions_array),
                                     class_names[true_label]),
               color=color)
    """

    plt.show()

def save_array(arr , file):
    output = open(file, 'wb')
    pickle.dump(arr, output)
    output.close()

def load_array(file):
    pkl_file = open(file, 'rb')
    data = pickle.load(pkl_file)
    return data

def my_init(shape, dtype=None):
    print("Calling my initializer: shape: ", shape, "dtype: ", dtype)
    ker = np.zeros(shape, dtype=dtype)
    #ker = np.ones()

```

```

    print("Kernel Shape to return: ",pca_weights.shape)
    return pca_weights

print("Tensorflow version: ",tf.__version__)
'''
all_train_data, test_data, all_train_classes, test_classes = read_preprocess_data()
save_array(all_train_data, "all_train_data.pk1")
save_array(all_train_classes, "all_train_classes.pk1")
save_array(test_data, "test_data.pk1")
save_array(test_classes, "test_classes.pk1")
'''

all_train_data = load_array("all_train_data.pk1")
all_train_classes = load_array("all_train_classes.pk1")
all_train_classes = np_utils.to_categorical(all_train_classes)
test_data = load_array("test_data.pk1")
test_classes = load_array("test_classes.pk1")
#print(test_classes)
test_classes = np_utils.to_categorical(test_classes)
#print(test_classes[0])

print("Running PCA on random patches: ")
print("All train data shape: ", all_train_data.shape)
all_train_data_pca = all_train_data
all_train_data_pca = all_train_data_pca.reshape(-1,2500)
print("After all_train_data reshape: ", all_train_data_pca.shape)
patches = extract_patches_2d(all_train_data_pca, (5,5), max_patches=75)
print("patches.shape: ", patches.shape)
patches = patches.reshape(-1,75)
print("after patches re-shape: ", patches.shape)
pca = PCA(n_components=25, random_state=0, svd_solver='randomized')
pca.fit(patches)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.ylim(0.8, 1.0)
plt.grid()
plt.show()
print("pca.components_.shape",pca.components_.shape)
print("pca.components_[0].shape",pca.components_[0].shape)
pca_weights = np.reshape(pca.components_, (5,5,3,25))
print("pca_weights.shape: ", pca_weights.shape)
print("End PCA! ")
model = create_model()
train_evaluate_model(model, all_train_data, all_train_classes, test_data, test_classes)

#Now test with the unseen data

```

```
#model.fit(train_data, train_classes, epochs=5)  
#test_loss, test_acc = model.evaluate(test_data, test_classes)
```

Appendices B
ETHICAL APPROVAL DOCUMENT



ETHICAL APPROVAL DOCUMENT


Date: 29/04/2022

To the Institute of Graduate Studies;

For the thesis project entitled "IMPLEMENTATION OF PLANT IDENTIFICATION USING DEEP LEARNING," the researchers declare that they did not collect any data from human/animal or any other subjects. Therefore, this project does not need to go through the ethics committee evaluation.

Title: Assoc. Prof. Dr.

Name Surname: Boran Şekeroğlu

Signature: 

Role in the Research Project: Supervisor

Appendices C

Similarity Report

turnitin.com/t_inbox.asp?i=84497899558383&svr=45&lang=en_us&aid=120565620

Boran Şekeröglü | User Info | Messages | Instructor | English | Community | Help | Logout

turnitin



Assignments | Students | Grade Book | Libraries | Calendar | Discussion | Preferences

NOW VIEWING: HOME > NEVEEN SALAH MUSTAFA > NEVEEN SALAH MUSTAFA MSc. THESIS

About this page
This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

Neveen Salah Mustafa MSc. Thesis
INBOX | NOW VIEWING: NEW PAPERS

Submit File Online Grading Report | Edit assignment settings | Email non-submitters

<input type="checkbox"/>	AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
<input type="checkbox"/>	Neveen Salah Mustafa	All Thesis	14% 	--	--		182363254	29-Apr-2022

Copyright © 1996 - 2022 Turnitin, LLC. All rights reserved.

Privacy Policy | Privacy Pledge | Terms of Service | EU Data Protection | Compliance | Copyright Protection | Legal FAQs | Helpdesk | Research Resources


Assoc. Prof. Dr. Boran Şekeröglü
Supervisor