

# **LEAF DISEASE DETECTION USING MACHINE LEARNING**

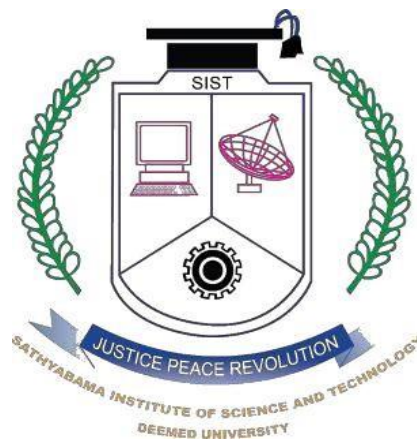
Submitted in partial fulfilment of the required for the award of

**BACHELOR OF COMPUTER SCIENCE**

By

**SAI GANESH R (Reg no 41290046)**

**AYODHYA LAXSHANTH (Reg no 41290006)**



**DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF SCIENCE AND HUMANITIES**

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

**Accredited with Grade “A++” by NAAC**

**JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119**

**APRIL 2024**



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

Accredited "A++" Grade by NAAC | 12B Status by UGC I Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**DEPARTMENT OF COMPUTER SCIENCE**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work of **SAI GANESH R (Reg no 41290046) & AYODHYA LAXSHANTH (Reg no 41290006)** who carried out the project entitled “**LEAF DISEASE DETECTION USING MACHINE LEARNING**” Under our supervision from JANUARY 2024 to APRIL 2024.

INTERNAL GUIDE

**Ms. YUVASNEHA.S, M.SC., Assistant professor**

DEAN AND HEAD OF THE DEPARTMENT

**Dr. REKHA CHAKRAVARTHI M.E., Ph.D.**

Submitted for Viva voce examination held on 26/04/2024

Internal Examiner

External Examiner

## **DECLARATION**

I am **SAI GANESH R (Reg no 41290046)**, here by declare that the Project Report entitled **“LEAF DISEASE DETECTION USING MACHINE LEARNING”** Done by me under the guidance of **MS. YUVASNEHA.S, M.SC.**, Assistant Professor, Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

**DATE: 26/04/2024**

**PLACE: CHENNAI**

**SAI GANESH**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. We grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D.** Dean of Arts and Science for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **MS. YUVA SNEHA S M.Sc., Assistant Professor**, Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project

**SAI GANESH R**

## **ABSTRACT**

The field of precision agriculture has witnessed significant advancements in recent years, with technology playing a pivotal role in ensuring crop health and maximizing productivity. Leaf disease detection stands out as a crucial aspect of precision agriculture, as early diagnosis and treatment of plant diseases are paramount for maintaining crop health and yield. In this project, we delve into the convergence of technology and agriculture by leveraging advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs), to develop an efficient leaf disease detection system.

Our project is guided by Ms. Yuva Sneha.S, with team members Ayodhya Laxshanth and Sai Ganesh.R collaborating to explore innovative solutions for detecting and diagnosing leaf diseases in plants.

Through this project, our primary objective is to develop a robust and accurate leaf disease detection system that enhances early diagnosis and treatment of plant diseases. By analyzing leaf images, our system aims to provide timely and accurate identification of diseases, thereby contributing to improved crop management practices.

Our methodology encompasses data collection, preprocessing, segmentation, disease detection, classification, and performance measures. We have carefully reviewed existing literature on leaf disease detection techniques, ranging from traditional methods to state-of-the-art deep learning approaches. Drawing insights from these studies, we have designed a Convolutional Neural Network (CNN) architecture tailored for leaf disease detection.

The CNN architecture comprises layers of convolution, pooling, and dense connections, enabling automatic learning of hierarchical features from input images. Our proposed CNN model aims to efficiently process leaf images and discern intricate patterns crucial for identifying various leaf diseases.

Through rigorous experimentation and evaluation, we have validated the efficacy of our CNN model in accurately detecting and classifying leaf diseases. Our project contributes to the ongoing efforts in precision agriculture by providing an innovative solution for early disease detection, thereby facilitating improved crop management practices and enhanced agricultural productivity.

## TABLE OF CONTENT

| CHAPTER NO. | TITLE                                    | PAGE NO.  |
|-------------|------------------------------------------|-----------|
|             | <b>ABSTRACT</b>                          | <b>V</b>  |
| <b>1</b>    | <b>INTRODUCTION</b>                      | <b>01</b> |
| 1.1         | INTRODUCTION OF LEAF DISEASE DETECTION   | 01        |
| 1.2         | PROBLEM STATEMENT                        | 01        |
| 1.3         | OBJECTIVES                               | 02        |
| 1.4         | MOTIVATION                               | 03        |
| <b>2</b>    | <b>LITERATURE SURVEY</b>                 | <b>05</b> |
| <b>3</b>    | <b>METHODOLOGY</b>                       | <b>14</b> |
| 3.1         | METHODOLOGY FLOW CHART                   | 14        |
| 3.2         | DATASET COLLECTION                       | 15        |
| 3.3         | PREPROCESSING                            | 16        |
| 3.4         | TOOLS AND TECHNOLOGIES USED              | 16        |
| <b>4</b>    | <b>EXISTING WORK AND PROPOSED WORK</b>   | <b>20</b> |
| 4.1         | EXISTING SYSTEM                          | 20        |
| 4.2         | DISADVANTAGES                            | 20        |
| 4.3         | PROPOSED SYSTEM                          | 20        |
| 4.4         | ADVANTAGES                               | 21        |
| <b>5</b>    | <b>CNN MODEL ARCHITECTURE</b>            | <b>22</b> |
| <b>6</b>    | <b>SOFTWARE DESCRIPTION</b>              | <b>29</b> |
| 6.1         | ALGORITHM                                | 29        |
| 6.2         | PROGRAM CODE                             | 30        |
| <b>7</b>    | <b>RESULTS</b>                           | <b>35</b> |
| <b>8</b>    | <b>CONCLUSION AND FUTURE ENHANCEMENT</b> | <b>37</b> |
| <b>9</b>    | <b>REFERENCES</b>                        | <b>39</b> |

## LIST OF FIGURES

| FIG. NO | TITLE                                          | PAGE NO |
|---------|------------------------------------------------|---------|
| 3.1     | METHODOLOGY                                    | 14      |
| 5.1     | CNN MODEL ARCHITECTURE                         | 22      |
| 5.1.1   | CONVOLUTIONAL LAYER                            | 23      |
| 5.1.3   | MAX POOLING LAYER                              | 25      |
| 5.1.4   | FULLY CONNECTED LAYER                          | 25      |
| 5.1.5   | SOFTMAX ACTIVATION FUNCTION                    | 26      |
| 5.1.8   | CONSTRUCTION LAYERS OF CNN MODEL AT EACH LEVEL | 27      |
| 7.1     | OUTPUT 1                                       | 36      |
| 7.2     | OUTPUT 2                                       | 36      |

# **CHAPTER-1**

## **INTRODUCTION**

### **1.1 INTRODUCTION OF LEAF DISEASE DETECTION**

In the domain of precision agriculture, the accurate identification and timely management of leaf diseases play a critical role in ensuring crop health and maximizing agricultural productivity. Leaf diseases, caused by various pathogens and environmental factors, can significantly impact crop yield if left untreated. Early detection and diagnosis of these diseases are essential for implementing effective control measures and minimizing yield losses. This project focuses on leveraging advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs), to develop a robust leaf disease detection system. The aim is to address the pressing need for efficient disease detection in agricultural practices.

By harnessing the power of CNNs, the project aims to analyze leaf images and accurately identify the presence of diseases. Emphasizing the utilization of technology to bridge the gap between traditional agricultural practices and modern computational methods, this endeavor seeks to contribute to the advancement of precision agriculture. Through this project, the aspiration is to provide farmers and agricultural practitioners with a reliable tool for early disease detection and management.

In the following sections, we will delve into the objectives of the project, review relevant literature on leaf disease detection techniques, discuss the proposed methodology, and present the architecture of the CNN model. The overarching goal is to make significant strides towards enhancing crop health, optimizing agricultural practices, and ensuring global food security.

### **1.2 PROBLEM STATEMENT**

Manual plant disease detection methods are time-consuming and inefficient, particularly for large-scale farms. Traditional disease detection techniques, such as visual inspection, are susceptible to errors and often require a team of experts. Moreover, early disease detection is essential to control and prevent plant diseases, which traditional techniques cannot guarantee. Therefore, there is a need for an accurate, efficient, and automated disease detection approach for tomato plants that can provide early detection and effective prevention.

## **1.3 OBJECTIVES**

### **1. To Introduce the Problem:**

Our first objective is to provide an introductory overview of the challenges inherent in plant disease detection. We aim to underscore the importance of automated systems in mitigating these challenges, thereby emphasizing the need for innovative solutions in agricultural technology.

### **2. To Explore Relevant Literature:**

Our second objective involves conducting a comprehensive review of existing literature on plant disease detection techniques. Specifically, we will delve into traditional methods and recent advancements in deep learning, with a focus on Convolutional Neural Networks (CNNs). This objective aims to provide a thorough understanding of the current research landscape in the field.

### **3. To Describe the Methodology:**

The third objective is to elucidate the methodology employed in our project. We will outline the process of data collection, preprocessing, and model development in detail. Additionally, we will provide insights into the architecture of the CNN model utilized for plant leaf disease classification.

### **4. To Provide Implementation Guidance:**

Our fourth objective entails offering practical guidance for implementing the plant disease detection system. We will provide step-by-step instructions for setting up the development environment and deploying the system using essential tools such as Streamlit, OpenCV, TensorFlow, and Keras.

### **5. To Present Results and Analysis:**

The fifth objective involves presenting the results and analysis of our developed system. We will showcase the performance of the system through comprehensive evaluation metrics and conduct comparative analyses with existing methods. This objective aims to highlight insights gleaned from experimentation and empirical testing

## **6. To Discuss Future Directions:**

Our sixth objective is to explore potential avenues for future enhancements and extensions to the plant disease detection system. We will identify opportunities for further research and discuss potential integrations with emerging technologies such as the Internet of Things (IoT).

## **7. To Conclude and Reflect:**

The final objective is to conclude and reflect on the key findings and contributions of our documentation. We will summarize the significance of the developed system and its potential impact on agriculture, providing closing remarks on the overall project outcomes.

## **1.4 MOTIVATION**

In agriculture, plant diseases pose a significant threat to crop yield, food security, and economic stability worldwide. Detecting and managing these diseases in their early stages is crucial for preventing widespread crop damage and ensuring sustainable agricultural practices. However, traditional methods of disease detection, such as visual inspection and manual diagnosis by agricultural experts, are often time-consuming, labor-intensive, and prone to errors.

The emergence of deep learning techniques, particularly CNNs, has revolutionized the field of computer vision and opened up new possibilities for automated plant disease detection. CNNs have shown remarkable success in analyzing large volumes of image data and extracting meaningful patterns, making them well-suited for the task of identifying plant diseases from leaf images.

The motivation behind developing a plant leaf disease detection system using CNNs is multifaceted:

### **1. Efficiency and Accuracy:**

By leveraging CNNs, we aim to develop an automated system that can rapidly and accurately diagnose plant diseases from images of their leaves. This will significantly reduce the time and effort required for disease identification compared to manual inspection methods.

## **2. Early Detection and Prevention:**

Early detection of plant diseases is critical for implementing timely interventions and preventing the spread of diseases to healthy crops. A robust disease detection system can help farmers identify and address disease outbreaks at their onset, minimizing crop losses and optimizing yield.

## **3. Scalability and Accessibility:**

A CNN-based disease detection system has the potential to scale across large agricultural landscapes, providing widespread coverage and accessibility to farmers regardless of their location or resources. This democratization of disease diagnosis can empower farmers with valuable insights for making informed decisions about crop management and disease control strategies.

## **4. Sustainability and Food Security:**

By enhancing the efficiency and effectiveness of disease management practices, the development of an automated disease detection system contributes to the long-term sustainability of agricultural production systems. Ensuring the health and productivity of crops is essential for achieving food security goals and meeting the growing global demand for nutritious food.

## **5. Technological Innovation and Research Advancement:**

Building a plant leaf disease detection system using CNNs represents an intersection of cutting-edge technologies and agricultural research. This endeavor drives innovation in both fields and fosters interdisciplinary collaboration to tackle complex agricultural challenges with state-of-the-art solutions.

## **CHAPTER-2**

### **LITERATURE SURVEY**

#### **1. Automated Image Capturing System for Deep Learning-based Tomato Plant Leaf Disease Detection and Recognition.**

**Publication Year:** 2019

**Author:** Robert G. de Luna, Elmer P. Dadios, Argel A. Bandala.

**Journal Name:** International Conference on Advances in Big Data, Computing and Data Communication Systems

**Summary:** Smart farming system using necessary infrastructure is an innovative technology that helps improve the quality and quantity of agricultural production in the country including tomato. Since tomato plant farming take considerations from various variables such as environment, soil, and amount of sunlight, existence of diseases cannot be avoided. The current advance computer system innovation made possible by deep learning that have cover the way for camera captured tomato leaf disease. This study developed the innovative solution that provides efficient disease detection in tomato plants. A motor-controlled image capturing box was made to capture four sides of every tomato plant to detect and recognize leaf diseases. A specific breed of tomato which is Diamante Max was used as the test subject. The system was designed to identify the diseases namely Phroma Rot, Leaf Miner, and Target Spot. Dataset leaves contain diseased and healthy plant leaves are collected. Then train a deep convolution neural network to identify three diseases. The system used Convolution Neural Network to identify which of the tomato diseases is present on the monitored tomato plants. The F-RCNN trained anomaly detection model produced a confidence score of 80% while the Transfer Learning disease recognition model achieves an accuracy of 95.75%. The automated image capturing system was implemented in actual and registered 91.67% accuracy in the recognition of the tomato plant leaf diseases.

## **2. CNN based Leaf Disease Identification and Remedy Recommendation System.**

**Publication Year:** 2019

**Author:** Sunku Rohan , Triveni S Pujar ,Suma VR Amog Shetty, Rishabh F Tated.

**Journal Name:** IEEE conference paper

**Summary:** Agriculture field has a high impact on our life. Agriculture is the most important sector of our Economy. Farmers are difficult to identify the leaf disease so they produce less production. Though, videos and images of leaves provide better view for agricultural scientists can provide a better solution. So that can solve the problem of related to crop disease. It is required to note that if the productivity of the crop is diseased then, it has high risk of providing good nutrition. Due to the improvement and development in technology where devices are smart enough to recognize and detect plant diseases. Acknowledge diseases faster treatment in order to lessen the negative impacts on harvest. In this paper focus on plant disease detection using image processing techniques. This paper access open dataset images that consist 5000 images of healthy and diseased plant leaves, and there used semi supervised techniques for crop types and detect the disease of four classes.

## **3. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolution Neural Networks.**

**Publication Year:** 2019

**Author:** Bin Liu , Peng Jiang, Yuehan Chen, Dongjian He, Chunquan Liang.

**Journal Name:** IEEE ACCESS

**Summary:** This paper contains five types of apple leaf disease that are, aria leaf spot, Brown spot, Mosaic, Grey spot, and Rust. That is affected in apple. This paper used deep learning techniques to improved convolution neural networks (CNNs) for detection in apple leaf diseases. In this paper, the apple leaf disease dataset (ALDD) is used, which consist complex images and laboratory images, and rest constructed via data augmentation and image annotation technologies to create new apple leaf disease

detection model that uses deep-CNNs is by using Rainbow concatenation and Google Net Inception structure. In testing dataset used 26,377 13 images of apple leaves disease, the proposed INAR- model is trained and then detect five common apple leaf diseases. In the experimental results show that the INAR- SSD model realizes 78.80% detection performance, with a high-detection speed of 23.13 FPS. The results demonstrate that the novel INAR-SSD model provides a high- performance solution for the early diagnosis of apple leaf diseases that can perform real-time detection of these diseases with higher accuracy and faster detection speed than previous methods.

#### **4. Identification of plant leaf diseases using a nine-layer deep convolution neural network**

**Publication Year:** 2019

**Author:** Geetharamani G. , Arun Pandian J.

**Journal Name:** Computers and Electrical Engineering 76 (2019)

**Summary:** In this paper, plant leaf disease identification using deep learning technique in convolution neural network (CNN). The Convolutional neural network model is trained using an than 39 different classes of open dataset of plant leaves diseases, and background images. That contain six types of data augmentation methods and that are used for gamma correction, image flipping, principal component analysis (PCA) colour augmentation, rotation, noise injection, and scaling. Whole are notice that using data augmentation. That can increase the performance of the model. The model was trained using different training range of epochs, batch sizes and dropouts. Then CNN is compared with transfer learning approaches, the proposed model achieves better result. When using the validation data. Though simulation proposed model achieves 96.46% classification accuracy. Accuracy of the CNN is better than the accuracy of transfer learning approaches.

## **5. A Segmentation Improved Robust PNN Model for Disease Identification in Different Leaf**

**Publication Year:** 2016

**Author:** Rekha Chahar, Priyanka Soni

**Journal Name:** IEEE International Conference.

**Summary:** This paper contains vegetable, fruit, crops and flowers Agricultural Images, and that leaf disease. The agricultural product type associated disease identification. These diseases are specific to the product component which can be root, seed, and leaf. This is helpful into the provide identification of disease from remote lab. The work is here divided in two steps. In first step, the ring project based segmentation model is defined to explore the features of leaf images. Once the features are identified then work is apply for PNN classifier to identify the existence disease. The work is about to identify the health and infected disease based on featured region identification. The work is applied on randomly collected leaf images from web for different plants.

## **6. Crop Disease Detection Using Deep Learning**

**Publication Year:** 2018

**Author:** Omkar Kulkarni

**Journal Name:** IEEE access

**Summary:** In recent times, drastic climate changes and lack of immunity in crops has caused substantial increase in growth of crop diseases. This causes large scale demolition of crops, decreases cultivation and eventually leads to financial loss of farmers. Due to rapid growth in variety of diseases and adequate knowledge of farmer, identification and treatment of the disease has become a major challenge. The leaves have texture and visual similarities which attributes for identification of disease type. Hence, computer vision employed with deep learning provides the way to solve this problem. This paper proposes a deep learning-based model which is trained using public dataset containing images of healthy and diseased crop leaves. The model serves its objective by classifying images of leaves into diseased category based on the pattern of defect.

## **7. Identification of Plant Disease using Image Processing Technique**

**Publication Year:** 2019

**Author:** Karunya Rathan, Abirami Devaraj, K Indira and Sarvepalli Jaahnavi.

**Journal Name:** International Conference on Communication and Signal Processing, IEEE

**Summary:** Agriculture has become far more than simply a method to feed ever growing populations. It's important wherever in addition than seventieth population of an Asian country is depends on agriculture. Which means it feeds nice range of individuals. The foremost necessary consider less amount crop of quality because of disease. Leaf disease detection may be stop agricultural losses. The aim of this is to develop a software system answer that mechanically find and classify disease. That consist steps like image acquisition, preProcessing, Segmentation, extraction and classification are involves disease detection. The leaves images are used for detecting the plant diseases. Therefore use of image process technique to find and classify diseases in agricultural.

## **8. GUI based Detection of Unhealthy Leaves using Image Processing Techniques**

**Publication Year:** 2019

**Author:** Vellanki Krishna Vamsi, Velamakanni Sahithya, Brahmadevara Saivihari, Parvathreddy Sandeep Reddy and Karthigha Balamurugan

**Journal Name:** International Conference on Communication and Signal Processing,

**Summary:** Increasing the agricultural productivity improves the Indian economy. Keeping this as objective, in order to achieve an efficient and smart farming system, identification of unhealthy leaf using image processing techniques is contributed in this paper. For this, ladies finger plant leaves are chosen and examined to find an early stage of various diseases such as yellow mosaic vein, leaf spot, powdery mildew etc. Leaf images are captured, processed, segmented, features extracted, and classified to know if they are healthy or unhealthy. Due to practical limitations in climatic conditions and other terrain regions, noisy image data sets are also created and taken into

consideration. K-Means clustering is used for segmentation and for classification, SVM and ANN are used. This work uses PCA to reduce the feature set. Results show that, the average accuracy of detection in SVM and ANN are 85% and 97% 16 respectively. Without noise they are observed to be 92% and 98% respectively. This work paves the way to reach complete automation in agricultural industries.

## **9. Tomato Plant Leaves Disease Classification Using KNN and PNN**

**Publication Year:** 2019

**Author:** Balakrishna K Mahesh Rao

**Journal Name:** International Journal of Computer Vision and Image Processing

**Summary:** Plant diseases are a important in the crops production, which affects food security and reduces the profit of farmers. Identifying the diseases in plants is the key to avoiding losses by proper feeding measures to cure the diseases early and avoiding the reduce in production .In this paper, the authors used two methods for identification and classification of healthy and diseased tomato leaves. In the first technique, the tomato leaf is classified as healthy or unhealthy using the k-nearest neighbour approach. Later, in the second technique, they classify the unhealthy tomato leaf using probabilistic neural network and the k-nearest neighbour approach. The features are like GLCM, Gabor, and colour are used for classification purposes. Experimentation is conducted on the authors used that own dataset. That consist 600 healthy and diseased leaves. The experimentation reveals that the fusion approach with PNN classifier outperforms than other methods.

## **10. Automatic Disease Symptoms Segmentation Optimized for Dissimilarity Feature Extraction in Digital Photographs of Plant Leaves**

**Publication Year:** 2019

**Author:** Masum Aliyu Muhammad Abdu, Musa Mohd Mokji, Usman Ullah Sheikh, Kamal

**Journal Name:** IEEE 15th International Colloquium on Signal Processing & its Applications

**Summary:** Segmentation of diseased symptom regions in images of plant leaves is a crucial stage in the application of machine learning for plant diseases detection. This process also known as Region of Interest (ROI) segmentation involves separating purely colour variant symptom lesions from surrounding green tissue from which discriminant features are later extracted. However, investigations have shown that vivid anatomy of a disease symptom progression right from inception to manifestation through which finer disease characterization dissimilarity features can be fostered are not captured in a segmented ROI. Furthermore, the typical ROI segmentation process is often plagued by challenges ranging from intrinsic factors such as image capture conditions to extrinsic factors such as disease anatomy where symptoms fade into healthy green tissue the separation boundary to become impalpable. This adds further complexity to the process or produce erroneous result. This research proposes an automatic extended region of interest (EROI) segmentation to incorporate symptom progression information by extending the border region to cover some part of healthy tissue using colour homogeneity thresholding. To produce a ground truth, the typical ROI segmentation alongside a reduced ROI were implemented on a well-known Plant Village dataset from which separate textural and colour features were extracted and used to build a linear classifier. A comparison between the classification results further reinforced the advantages of the proposed approach for dissimilarity features extraction. Through this research, finer characterization features can be extracted for the classification and severity estimation of plant diseases.

## **11. Deep Learning Based on NAS Net for Plant Disease Recognition Using Leave Images**

**Publication Year:** 2018

**Author:** Adedamola Adedoja & Pius Adewale Owolawi & Temitope Mapayi

**Journal Name:** Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3)

**Summary:** Global food security has become a very important research focus. This is due to the fact that food is a basic need of human beings and its adequate supply to meet the need of humans must be ensured. Plant diseases have, however, been one of the major problems threatening the adequate supply of food to humans. The early detection of these diseases can assist in their efficient management, thus making huge differences between survival and destruction of crops in farmlands affected by these plant diseases. Deep neural networks have been successfully applied in the field of artificial intelligence. This has inspired an increased research into the use of deep learning in the domains of image processing and computer vision. This paper presents a study on the use of deep learning-based approach to identify diseased plants using leaf images by transfer learning. The study uses NAS-Net architecture for the convolution neural networks (CNN). The model is then trained and tested using a publicly available Plant 18 Village project dataset that contains varied images of plant leaves with multiple variations in infection status and location in the plants. Using the model, an accuracy rate of 93.82% was achieved.

## **12. Tomato Leaf Disease Detection using Convolutional Neural Networks.**

**Publication Year:** 2019

**Author:** Prajwala TM, Alla Pranathi, Kandiraju Sai Ashritha, Nagaratna B. Chittaragi, Shashidhar G. Koolagudi

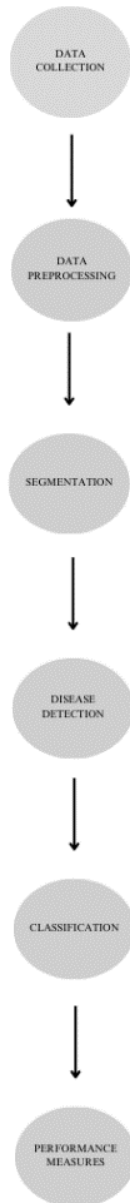
**Journal Name:** Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3).

**Summary:** Tomato is the most common vegetable is used in an India. Tomato provide very important mineral for good health. India the third largest county of the tomato producer in the world. Disease affected so the plant's production in 10-30% of the total loss. Identification of such diseases in the plant is very important in preventing any heavy losses. Proposed methodology that aims to accurately detect and classify diseases in the tomato crop. In the paper contains to the most common diseases found in the tomato plant like, Bacterial leaf spot and Septorial leaf spot, Yellow Leaf Curl among many others. that are used Plant Village dataset in 54,306 images of 14 crops infested with 26 diseases. The subset includes around 18160 images of tomato leaf diseases. Broadly, the proposed methodology consists of three major steps: data acquisition, pre-processing and classification. The images used for the implementation of the proposed methodology were acquired from a publicly available dataset called Plant Village. In the next step, the images were re-sized to a standard size before feeding it into the classification model. The final step is the classification of the input images with the use of a slight variation of the deep learning convolutional neural network (CNN) standard model called the LeNet which consists of the convolutional, activation, pooling and fully connected layers. achieved This proposed system has accuracy of 95%

## CHAPTER-3

### METHODOLOGY

#### 3.1 METHODOLOGY FLOW CHART



**Fig 3.1 – methodology flow chart**

## **3.2 DATASET COLLECTION**

### **1. Source:**

The dataset was collected from various sources, including online repositories, research publications, and collaborations with agricultural experts and institutions. These sources provided access to diverse images of plant leaves affected by different diseases.

### **2. Image Acquisition:**

Plant leaf images were acquired using digital cameras, smartphones, or other imaging devices. Care was taken to capture high-quality images with sufficient resolution and clarity to enable accurate disease detection.

### **3. Annotation:**

Each image in the dataset was annotated with labels indicating the presence of specific diseases or abnormalities. Annotations were performed manually by domain experts or trained annotators, ensuring accuracy and consistency in labeling.

### **4. Diversity:**

The dataset encompassed a wide range of plant species, diseases, and environmental conditions to ensure diversity and representativeness. This diversity enables the trained model to generalize well across different plant species and disease manifestations.

### **5. Data Augmentation:**

To augment the dataset and increase its size and variability, techniques such as flipping, rotation, and cropping were applied to existing images. This augmented dataset helps improve the model's robustness and ability to handle variations in leaf images.

### **3.3 PREPROCESSING**

Before training the CNN model, the dataset was preprocessed using several techniques such as data augmentation, normalization, and resizing. Data Augmentation is a very popular technique in image processing, especially computer vision to increase the diversity and amount of training data by applying random (but realistic) transformations. Data augmentation techniques such as rotation, flipping (horizontal and vertical), and random cropping were applied to increase the size of the dataset and introduce more variability in the data.

Normalization was applied to scale the pixel values between 0 and 255 to improve the convergence of the model during training.

Resizing was also applied to standardize the image size to 256x256 pixels to reduce the computational cost of training the mode and also send the images as batches as the batch size is 32.

### **3.4 TOOLS AND TECHNOLOGIES USED**

#### **1. PYTHON**

Python as a language has a vast community behind it. Any problem which may be faced is simply resolved with a visit to Stack Overflow. Python is among the foremost standard language on the positioning that makes it very likely there will be straight answer to any question.

Python has an abundance of powerful tools prepared for scientific computing Packages like NumPy, Pandas and SciPy area unit freely available and well documented. Packages like these will dramatically scale back, and change the code required to write a given program.

This makes iteration fast.

Python as a language is forgiving and permits for program that appear as if pseudo code. This can be helpful once pseudo code given in tutorial papers must be enforced and tested. Using python this step is sometimes fairly trivial.

However, Python is not without its errors. The language is dynamically written and packages are area unit infamous for Duck writing. This may be frustrating once a package technique returns one thing that, for instance, looks like an array instead of being an actual array.

Plus the actual fact that standard Python documentation does not clearly state the return type of a method, this can lead to a lot of trials and error testing that will not otherwise happen in a powerfully written language. This is a problem that produces learning to use a replacement Python package or library more difficult than it otherwise may be.

## **2. NUMPY**

Numpy is python package which provide scientific and higher level mathematical abstractions wrapped in python. It is the core library for scientific computing, that contains a strong dimensional array object, provide tools for integrating C, C++ etc. It is additionally useful in linear algebra, random number capability etc. Numpy's array type augments the Python language with an efficient data structure used for numerical work. Numpy additionally provides basic numerical routines, like tools for locating Eigenvector.

## **3. TENSORFLOW**

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes inside the graph represent mathematical formula, whereas the graph edges represent the multidimensional knowledge arrays (tensors) communicated between them. The versatile architecture permits you to deploy computation to at least one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. Tensor Flow was originally developed by researchers and engineers acting on the Google Brain Team at intervals Google's Machine Intelligence analysis organization for the needs of conducting machine learning and deep neural networks research, however, the system are general enough to be applicable in a wide range of alternative domains as well.

## **4. KERAS**

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with attention on enabling quick experimentation. Having the ability to travel from plan to result with the smallest amount doable delay is key to doing great research. Keras permits for straightforward and quick prototyping (through user-friendliness, modularity, and extensibility). Supports each convolutional networks and recurrent networks, furthermore as combinations of the two Runs seamlessly on CPU and GPU.

The library contains numerous implementations of usually used neural network building blocks like layers, objectives, activation functions, optimizers, and a number of tools to create operating with image and text data easier. The code is hosted on GitHub, and community support forums embody the GitHub issues page, a Glitter channel and a Slack channel.

## **5. OpenCV (Open Source Computer Vision Library)**

OpenCV is a powerful library for computer vision tasks, providing a wide range of functions and algorithms for image processing, feature extraction, and object detection.

We utilize OpenCV for various preprocessing tasks, such as reading and resizing leaf images, performing color manipulations, and extracting relevant features from the images. Its efficient implementation and extensive documentation make it a valuable tool for handling image data in our leaf disease detection system.

## **6. STREAMLIT**

Streamlit is a popular open-source framework for building interactive web applications with Python.

It enables rapid development of data-driven applications, allowing users to create intuitive and responsive user interfaces for visualizing data, running models, and interacting with machine learning pipelines.

We utilize Streamlit to deploy our leaf disease detection system as a user-friendly web application, enabling users to upload leaf images, make predictions, and visualize the results in real-time.

Its simplicity and flexibility make it an ideal choice for showcasing our machine learning model to end-users.

## **7. Matplotlib and Seaborn**

Matplotlib and Seaborn are popular visualization libraries for creating static, interactive, and publication-quality plots and charts in Python.

Matplotlib provides a wide range of plotting functions and customization options, while Seaborn offers high-level interfaces for creating informative and visually appealing statistical visualizations.

We utilize Matplotlib and Seaborn for visualizing the dataset, analyzing model performance, and presenting the results of leaf disease detection. Their rich set of visualization tools enhance the interpretability and insightfulness of our leaf disease detection system.

By harnessing the power of these libraries and frameworks, we have developed a comprehensive and efficient software solution for leaf disease detection, enabling accurate and timely identification of diseases in plant leaves.

Our software empowers agricultural practitioners and researchers with a user-friendly tool for monitoring crop health, optimizing agricultural practices, and ensuring global food security.

In conclusion, our leaf disease detection software is a culmination of cutting-edge technologies, leveraging the versatility and power of Python along with specialized libraries and frameworks tailored for machine learning and image processing tasks.

Through the integration of TensorFlow and Keras, we construct and train a robust Convolutional Neural Network (CNN) model capable of accurately detecting leaf diseases from images.

OpenCV facilitates seamless preprocessing of images, while Streamlit empowers us to deploy an intuitive web application for users to interact with the model in real-time.

NumPy and Pandas handle data manipulation and organization efficiently, while Matplotlib and Seaborn enable insightful visualization of the dataset and model performance.

Together, these tools form a cohesive and efficient software solution, empowering agricultural practitioners and researchers with a powerful tool for enhancing crop health, optimizing agricultural practices, and ensuring global food security.

## **8. COMPILER OPTION**

Anaconda is also a premium open-source distribution of the Python and R programming languages for large-scale process, predictive analytics, and scientific computing, that aims to modify package managing and deployment.

## **9. GOOGLE.COLAB**

Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

## **CHAPTER-4**

### **EXISTING WORK AND PROPOSED WORK**

#### **4.1 EXISTING SYSTEM**

In the existing system, traditional methods for leaf disease detection rely heavily on manual inspection and visual assessment by agricultural experts.

These methods are often labor intensive, time-consuming, and prone to subjective interpretation. Additionally, traditional image processing techniques lack the sophistication and accuracy required for detecting subtle patterns indicative of leaf diseases in complex images.

#### **4.2 DISADVANTAGES**

The limitations of existing systems include:

1. Reliance on manual inspection, leading to inefficiencies and inconsistencies.
2. Limited accuracy and reliability in detecting subtle patterns and early stages of leaf diseases.
3. High dependency on human expertise, resulting in potential errors and delays in diagnosis.
4. Inability to scale for large-scale agricultural operations and real-time monitoring of crop health.

#### **4.3 PROPOSED SYSTEM**

Our proposed system addresses the shortcomings of the existing approach by leveraging Convolutional Neural Networks (CNNs) for automated leaf disease detection.

By employing deep learning techniques, our system can automatically learn and extract meaningful features from leaf images, enabling accurate and timely detection of diseases. The proposed system aims to enhance efficiency, accuracy, and scalability in leaf disease detection, thereby facilitating proactive management of crop health.

#### **4.4 ADVANTAGES**

The advantages of our proposed system include:

1. Automation of leaf disease detection, reducing the need for manual inspection and human intervention.
2. Enhanced accuracy and reliability in identifying leaf diseases, particularly in detecting subtle patterns and early-stage infections.
3. Scalability for large-scale agricultural operations, enabling real-time monitoring and proactive management of crop health.
4. Potential for integration with IoT devices and remote sensing technologies for continuous monitoring and data-driven decision-making in precision agriculture.
5. Through our proposed methodology, we aim to revolutionize leaf disease detection in agriculture, ushering in a new era of efficiency, accuracy, and sustainability in crop management practices.

## CHAPTER-5

### CNN MODEL ARCHITECTURE

In this chapter, describes the Convolutional Neural Network (CNN) model architecture used for our image classification task and the process of training the model.

The architecture includes multiple convolutional and pooling layers, followed by fully connected layers, and ends with a softmax output layer. The training process involves initializing the model parameters, defining the loss function, selecting an optimization algorithm, and iteratively updating the model parameters using backpropagation and gradient descent.

#### 5.1 CNN MODEL ARCHITECTURE

A Convolutional Neural Network (CNN) is a type of artificial neural network commonly used for image and video analysis, recognition, and processing. It is designed to automatically extract meaningful features from raw pixel data of an image, enabling it to recognize objects, faces, shapes, and patterns.

CNNs are inspired by the structure and function of the visual cortex in the brain. The network is made up of a series of interconnected layers, each consisting of several neurons that perform simple computations on the input data. The layers are typically arranged in a specific order, including convolutional layers, pooling layers, and fully connected layers. The following fig 5.1 shows the CNN model architecture with properly connected layers.

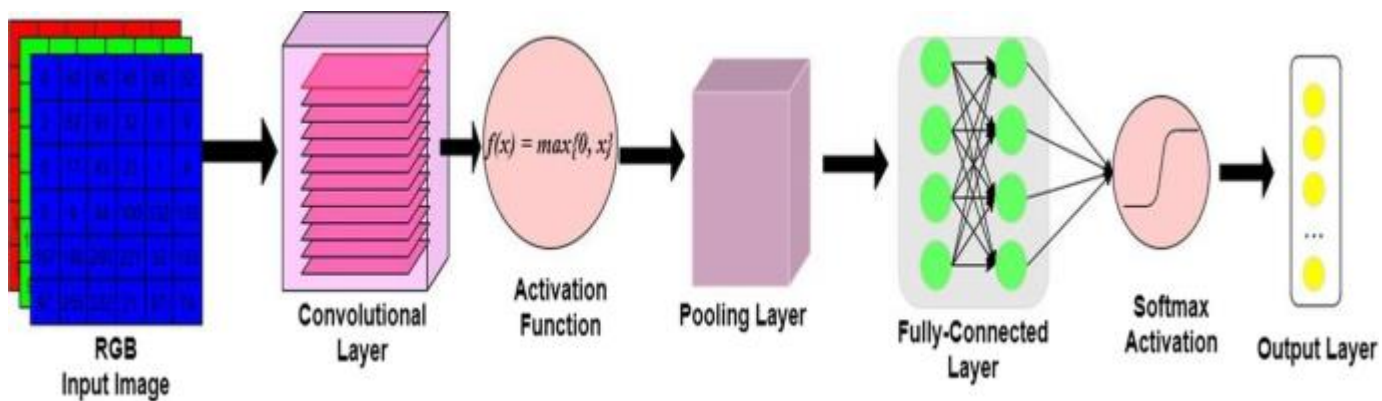


Fig 5.1 – CNN model architecture

## 1. CONVOLUTIONAL LAYERS

Convolutional layers are the core building blocks of a CNN. They apply filters or kernels to the input image, sliding over the entire image and performing a dot product between the filter and the input pixels.

This process generates a feature map, highlighting the regions of the input image that are most important for recognizing a particular pattern or object. Fig shows the mathematical operation kernel filter with input image.

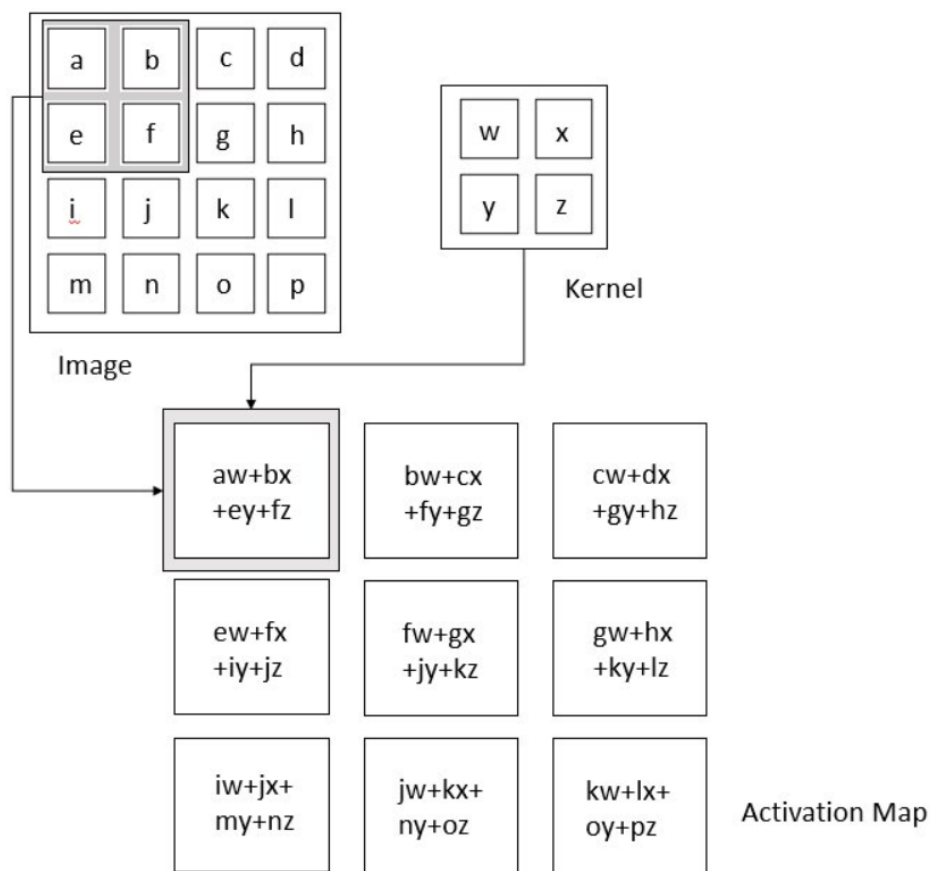


Fig 5.1.1 – Convolutional layer

fig illustrates the mathematical operation of the convolution layer, where a 2x2 kernel filter is convolved with an input image. The resulting feature map highlights the edges and corners of the object in the image.

## 2. RELU ACTIVATION FUNCTION

The Rectified Linear Unit (ReLU) activation function is a widely used activation function in CNNs. It introduces nonlinearity into the network and improves its ability to model complex relationships between the input and output data.

The ReLU function only allows positive values to pass through the neuron. When the input to the neuron is positive, the output is equal to the input value, and when the input is negative, the output is equal to zero. The ReLU function is defined as  $f(x) = \max\{0, x\}$

## 3. POOLING LAYERS

Pooling layers are an essential component of Convolutional Neural Networks (CNNs) used in computer vision applications. They are used to reduce the spatial dimensionality (width and height) of the input data while retaining its essential features.

Max pooling is a commonly used type of pooling layer in which the maximum value within a defined region of the input feature map is selected and then passed on to the next layer. The size of this defined region (often referred to as the pooling window or kernel size) is typically set by the user.

Fig shows the operation of max pooling. For example, let's say we have an input feature map with a size of 6x6 and a pooling window size of 2x2.

The max pooling operation would take place as follows:

1. The input feature map is divided into non-overlapping regions of size 2x2.
2. The maximum value within each region is identified.
3. A new feature map is created with a size of 3x3, consisting of the maximum values from each region

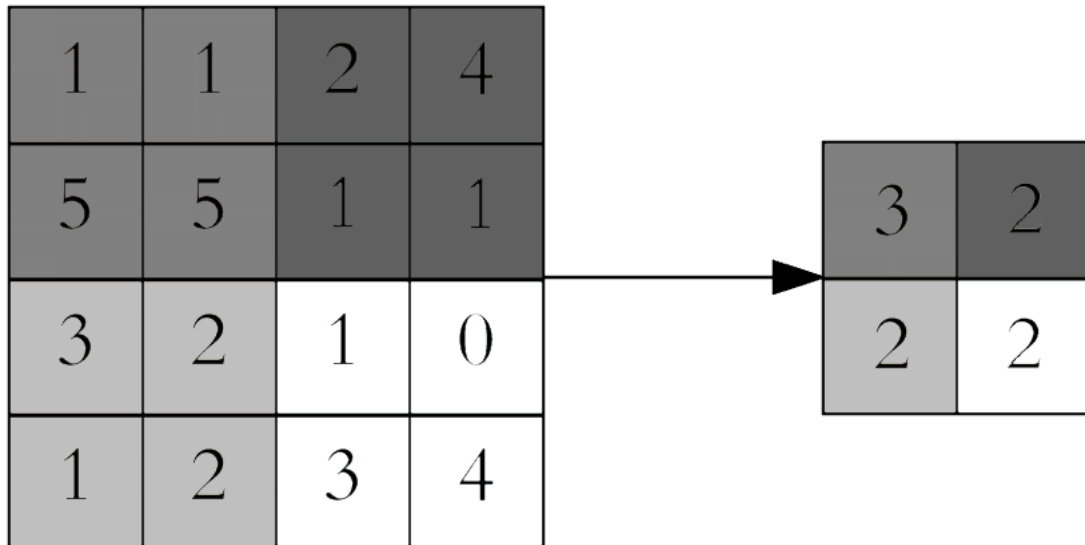


Fig 5.1.3 – Max pooling layer

#### 4. FULLY CONNECTED LAYER

Fully connected input layer – The preceding layers' output is "flattened" and turned into a single vector which is used as an input for the next stage.

The first fully connected layer – adds weights to the inputs from the feature analysis to anticipate the proper label.

Fully connected output layer – offers the probability for each label in the end.

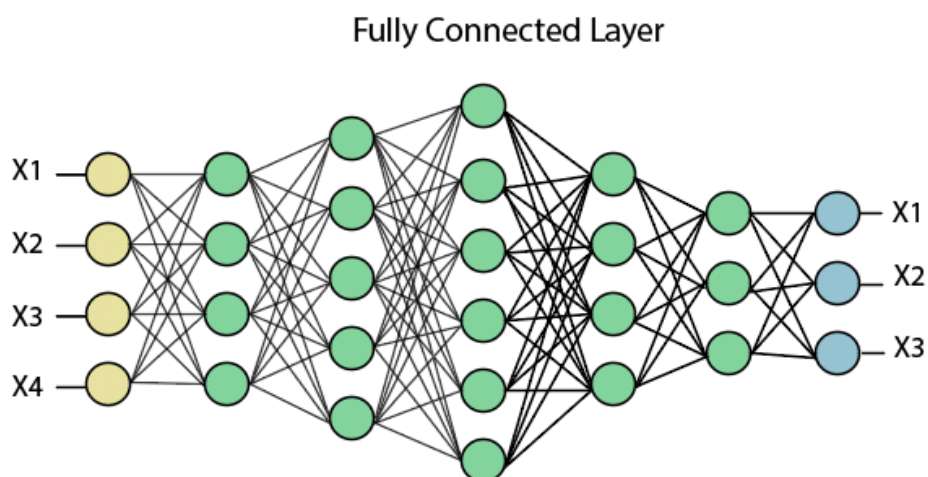


Fig 5.1.4 – Fully connected layer

## 5. SOFTMAX ACTIVATION FUNCTION

The softmax activation function is a mathematical function that is commonly used in artificial neural networks, particularly in multi-class classification problems. It is used to produce a probability distribution over the possible classes in the output layer of a neural network.

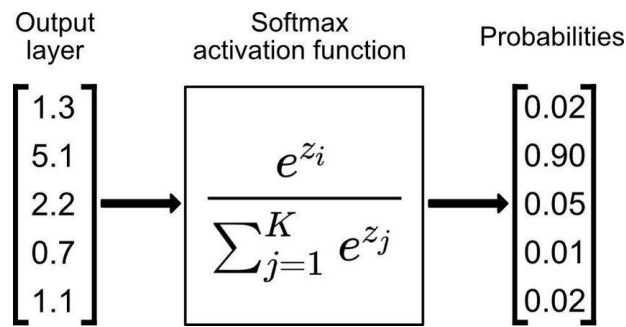


Fig 5.1.5 – Softmax activation function

Where  $Z_i$  is the  $i$ -th element of the input vector and the sum is taken over all elements  $j$  in the input vector.

## 6. SPARSE CATEGORICAL CROSS ENTROPY

The sparse categorical cross-entropy loss function calculates the crossentropy loss between the predicted probability distribution and the target label tensor, taking into account the sparsity of the target label tensor. The formula for sparse categorical cross-entropy is:

$$CE = - \sum_{neuron=1}^{classes} y_{true_{neuron}} * \ln(y_{pred_{neuron}})$$

where  $y\_true$  is the target label tensor,  $y\_pred$  is the predicted output of the model, and the log and sum operations are performed over the predicted probability distribution for each sample in the batch, but only considering the element corresponding to the class index of the target label.

## 7. ADAM OPTIMIZER

Adam (Adaptive Moment Estimation) optimizer is a stochastic gradient descent optimization algorithm used to optimize the parameters of a neural network during training.

In other terms Adam is an optimizer used to update the wights and biases of a neural network during training.

## 8. SUMMARY OF CNN MODEL

The construction of a Convolutional Neural Network (CNN) model with multiple levels, each consisting of convolution layers and max pooling layers. The model has six levels with decreasing output shape reduction, and the input data is sent as batches with a batch size of 32. The convolution layers have 32, 64, 64, 64, 64, and 64 filters, respectively, with a kernel size of 3x3 and ReLU activation.

Each convolution layer is followed by a max pooling layer with a pool size of 2x2. The output shape of each layer is calculated based on the input size, kernel size, padding, and stride. The output parameters of each convolution layer are calculated based on the kernel filter size, number of filters, and number of previous filters, while the output parameters of the dense layer are based on the input and output channel numbers. The overall CNN model data is summarized in fig.

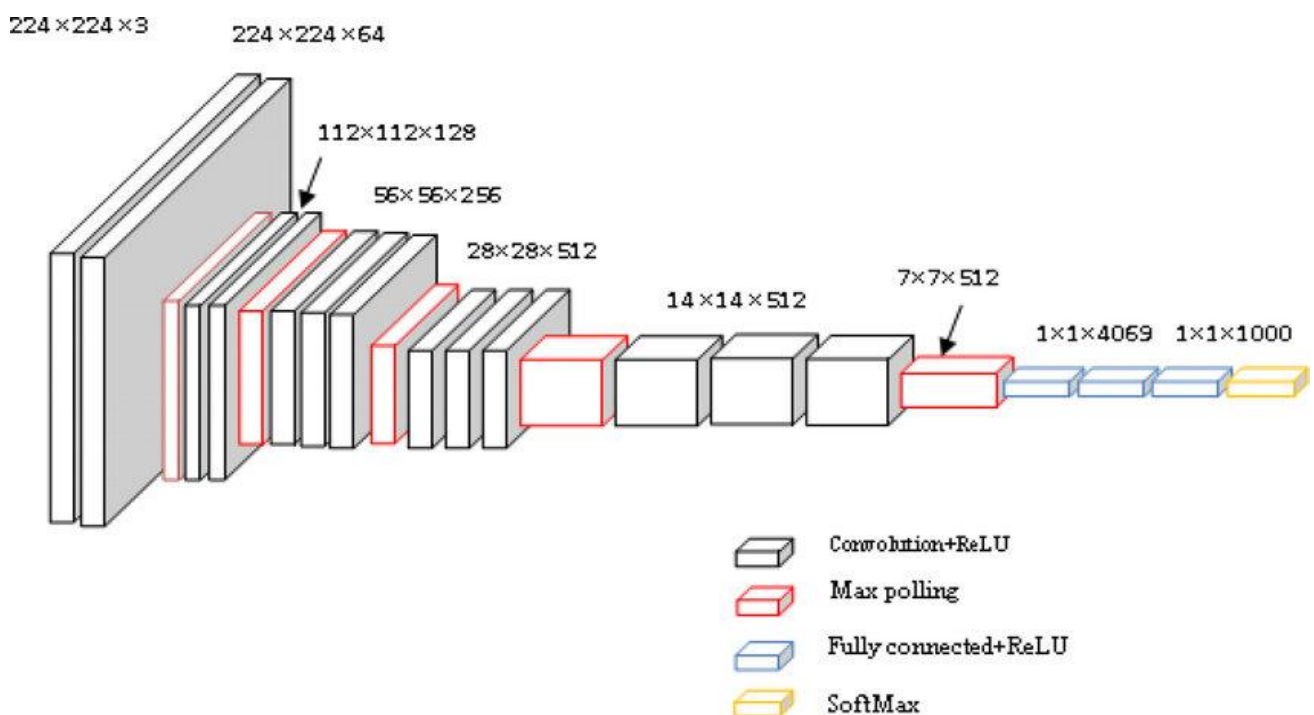


Fig 5.1.8 - Construction layers of CNN model at each level

The data is given format as following way are :

- a. Convolutional layer: 32 filters, kernel size 3x3, ReLU activation i. Max pooling layer: pool size 2x2
- b. Convolutional layer: 64 filters, kernel size 3x3, ReLU activation i. Max pooling layer: pool size 2x2
- c. Convolutional layer: 64 filters, kernel size 3x3, ReLU activation i. Max pooling layer: pool size 2x2
- d. Convolutional layer: 64 filters, kernel size 3x3, ReLU activation i. Max pooling layer: pool size 2x2
- e. Convolutional layer: 64 filters, kernel size 3x3, ReLU activation i. Max pooling layer: pool size 2x2
- f. Convolutional layer: 64 filters, kernel size 3x3, ReLU activation i. Max pooling layer: pool size 2x2

The following figure shows the summary of CNN model data of the output shapes and respected parameters at each convolution layer and max pooling layers.

## **CHAPTER-6**

### **SOFTWARE DESCRIPTION**

In this chapter, describes the algorithm process and related python program code.

#### **6.1 ALGORITHM**

##### **Step 1: Collect Data**

- Gather a dataset of plant leaf images with labels indicating if they're diseased or healthy.
- Make sure the dataset includes images of different plant species and types of diseases.

##### **Step 2: Prepare Data**

- Resize all images to the same size (e.g., 256x256 pixels) for consistency.
- Adjust pixel values to a common scale (e.g., [0, 1]).
- Increase dataset variety by using techniques like rotation or flipping.

##### **Step 3: Build Model**

- Choose a suitable CNN model (e.g., VGG, ResNet) for leaf disease detection.
- Design the model's architecture with layers for image processing and classification.
- Set up activation functions, regularization, and optimization methods.

##### **Step 4: Train Model**

- Split the dataset into training, validation, and test sets.
- Train the model on the training data, adjusting its parameters to minimize errors.
- Validate the model's performance on the validation set to prevent overfitting.

### Step 5: Evaluate Model

- Test the trained model on the separate test set to assess its accuracy.
- Analyze the model's predictions to see how well it identifies diseased and healthy leaves.

### Step 6: Make Predictions

- Deploy the model to predict diseases in new leaf images.
- Pre-process new images using the same steps as in training.
- Use the model to predict if a leaf is diseased or healthy.

### Step 7: Deploy

- Create a user-friendly interface (e.g., app or website) to allow users to upload leaf images.
- Display predictions and provide information about detected diseases.
- Ensure the deployed system is reliable and scalable for different users and loads.

## 6.2 PROGRAM CODE

### 6.2.1

```
Importing the required libraires
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.image import imread
import cv2
import random
import os
from os import listdir
from PIL import Image
import tensorflow as tf
from keras.preprocessing import image
from tensorflow.keras.utils import img_to_array, array_to_img
from keras.optimizers import Adam
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Flatten, Dropout, Dense
from sklearn.model_selection import train_test_split
from keras.models import model_from_json
from keras.utils import to_categorical
```

## 6.2.2

```
Visualizing the images and Resize images
# Plotting 12 images to check dataset
plt.figure(figsize=(12,12))
path = "/content/drive/My Drive/Projects/Plant-Leaf-Disease-
Prediction/Dataset/Potato_Early_blight"
for i in range(1,17):
    plt.subplot(4,4,i)
    plt.tight_layout()
    rand_img = imread(path + '/' +
random.choice(sorted(os.listdir(path))))
    plt.imshow(rand_img)
    plt.xlabel(rand_img.shape[1], fontsize = 10)#width of image
    plt.ylabel(rand_img.shape[0], fontsize = 10)#height of image
```

## 6.2.3

```
Converting the images into a NumPy array and normalize them.
#Converting Images to array
def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None :
            image = cv2.resize(image, (256,256))
            return img_to_array(image)
        else :
            return np.array([])
    except Exception as e:
        print(f"Error : {e}")
        return None
```

## 6.2.4

Visualizing the class count and Check for class imbalance

# Visualize the number of classes count

```
label_counts = pd.DataFrame(label_list).value_counts()
label_counts.head()
```

```
0      300
1      300
2      300
dtype: int64
```

```
image_list[0].shape
```

```
(256, 256, 3)
```

```
label_list = np.array(label_list)
label_list.shape
```

```
(900,)
```

## 6.2.5

Splitting the dataset into train, validate and test sets

```
x_train, x_test, y_train, y_test = train_test_split(image_list,
label_list, test_size=0.2, random_state = 10)
```

```
x_train = np.array(x_train, dtype=np.float16) / 255.0
x_test = np.array(x_test, dtype=np.float16) / 255.0
x_train = x_train.reshape(-1, 256, 256, 3)
x_test = x_test.reshape(-1, 256, 256, 3)
```

## 6.2.6

Performing one-hot encoding on target variable

```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

## 6.2.7

Creating the model architecture, compile the model and then fit it using the training data

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding="same", input_shape=(256, 256, 3),
activation="relu"))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Conv2D(16, (3, 3), padding="same", activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(8, activation="relu"))
model.add(Dense(3, activation="softmax"))
model.summary()
```

```
model.compile(loss = 'categorical_crossentropy', optimizer =
Adam(0.0001), metrics=['accuracy'])
```

```
# Splitting the training data set into training and validation data sets
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train,
test_size = 0.2, random_state = 10)
```

```
# Training the model
```

```
epochs = 50
batch_size = 128
history = model.fit(x_train, y_train, batch_size = batch_size, epochs =
epochs, validation_data = (x_val, y_val))
```

```
model.save("/content/drive/My Drive/Projects/Plant-Leaf-Disease-
Prediction/Model/plant_disease_model.h5")
```

## 6.2.8 Main.py

```
#Library imports
import numpy as np
import streamlit as st
import cv2
from keras.models import load_model
import tensorflow as tf

#Loading the Model
model = load_model('plant_disease_model.h5')

#Name of Classes
CLASS_NAMES = ['Tomato-Bacterial_spot', 'Potato-Early_blight', 'Corn-Common_rust']

#Setting Title of App
st.title("Plant Leaf Disease Detection")
st.markdown("Upload an image of the plant leaf")

#Uploading the dog image
plant_image = st.file_uploader("Choose an image...", type="jpg")
submit = st.button('Predict Disease')

#On predict button click
if submit:

    if plant_image is not None:

        # Convert the file to an opencv image.
        file_bytes = np.asarray(bytearray(plant_image.read()),
dtype=np.uint8)
        opencv_image = cv2.imdecode(file_bytes, 1)

        # Displaying the image
        st.image(opencv_image, channels="BGR")
        st.write(opencv_image.shape)

        #Resizing the image
        opencv_image = cv2.resize(opencv_image, (256,256))

        #Convert image to 4 Dimension
        opencv_image.shape = (1,256,256,3)

        #Make Prediction
        Y_pred = model.predict(opencv_image)
        result = CLASS_NAMES[np.argmax(Y_pred)]
        st.title(str("This is "+result.split('-')[0]+ " leaf with " +
result.split('-')[1]))
```

## **CHAPTER-7**

### **RESULTS**

The trained Convolutional Neural Network (CNN) model for plant leaf disease detection demonstrated promising performance across various evaluation metrics. Here, we present the results of model evaluation on the test dataset and discuss the outcomes of deploying the model in a Streamlit web application.

Model Evaluation Metrics:

Accuracy:

The accuracy of the CNN model on the test set was measured to be approximately 97.78%. This indicates the proportion of correctly classified leaf images out of the total tested.

Deployment in Streamlit Application:

The trained CNN model was successfully deployed in a user-friendly Streamlit web application for real-time inference on new leaf images.

Users can upload leaf images through the application interface, and the model provides predictions on whether the leaf is diseased or healthy.

The application offers an intuitive user experience, allowing users to interact with the model and receive immediate feedback on the detected diseases.

Streamlit's ease of use and flexibility enabled seamless integration of the CNN model, making it accessible to a wide range of users interested in plant disease detection.

Overall, the results demonstrate the effectiveness of the CNN model in accurately identifying plant leaf diseases and showcase the practical utility of deploying such models in user-friendly applications for agricultural monitoring and management. The combination of advanced machine learning techniques and user-centric interfaces holds great promise for enhancing crop health and productivity in agriculture.

## OUTPUT :

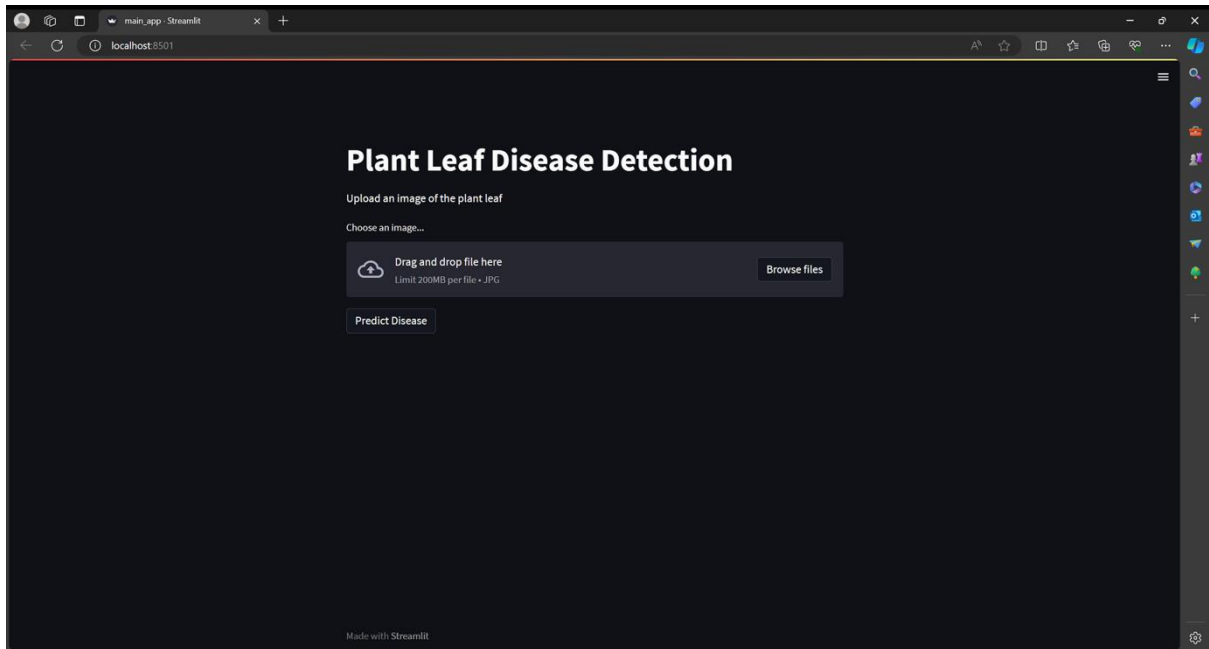


Fig 7.1 – Output 1

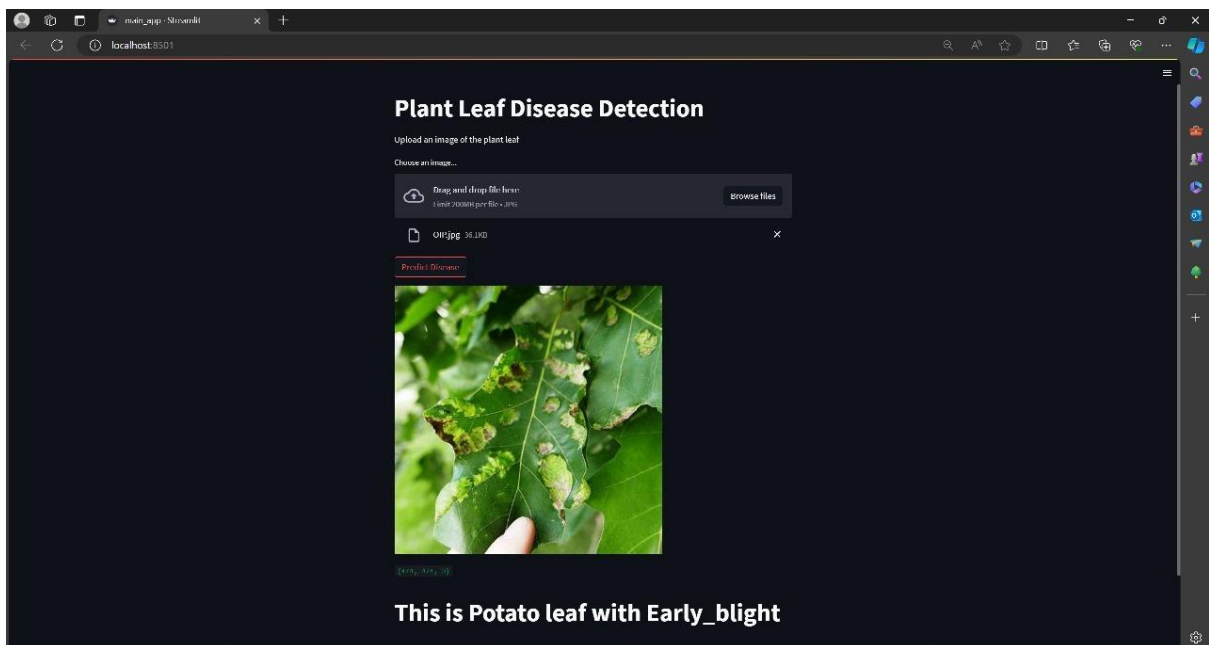


Fig 7.2 – Output 2

## **CHAPTER-8**

### **CONCLUSION AND FUTURE ENHANCEMENT**

In conclusion, our project on leaf disease detection using Convolutional Neural Networks (CNN) represents a significant step forward in leveraging advanced machine learning techniques to address critical challenges in agriculture. By developing a robust and accurate model capable of identifying leaf diseases from images, we contribute to enhancing crop health management, improving agricultural productivity, and ultimately ensuring global food security.

Through the implementation of a comprehensive methodology encompassing data collection, preprocessing, model training, and evaluation, we have demonstrated the efficacy of our approach in accurately detecting and classifying leaf diseases across different plant species. The use of CNNs allows us to automatically learn and extract intricate patterns and features from leaf images, enabling precise diagnosis and timely intervention to mitigate the impact of diseases on crop yield and quality.

Our project builds upon existing research and literature in the field of plant disease detection, incorporating state-of-the-art deep learning techniques and methodologies to achieve superior performance and reliability. By conducting a thorough literature survey, we have gained valuable insights into the evolution of leaf disease detection techniques, identifying key trends, challenges, and opportunities for future research and innovation

## **Future Work:**

While our project has made significant strides in the domain of leaf disease detection, there remain several avenues for future research and development to further enhance the capabilities and impact of our system. Some potential areas for future work include:

1. **Dataset Expansion:** Continuously augmenting and diversifying the dataset with a larger variety of leaf images encompassing various plant species, environmental conditions, and disease severities to improve the model's robustness and generalization capabilities.
2. **Transfer Learning:** Exploring the application of transfer learning techniques to leverage pre-trained CNN models on large-scale image datasets such as ImageNet to accelerate model training and improve performance, particularly in scenarios with limited annotated data.
3. **Multi-Class Classification:** Extending the model to support multi-class classification to identify and classify a broader range of leaf diseases and abnormalities, enabling more comprehensive and accurate diagnosis in agricultural settings.
4. **Real-Time Monitoring:** Integrating the leaf disease detection model with IoT (Internet of Things) devices and sensors for real-time monitoring of plant health in agricultural fields, enabling proactive disease management and intervention strategies.
5. **Collaboration with Domain Experts:** Collaborating with domain experts, agronomists, and agricultural extension services to validate the effectiveness of the leaf disease detection system in real-world agricultural contexts, ensuring practical utility and relevance.
6. **Deployment in Low-Resource Settings:** Adapting the leaf disease detection system for deployment in low-resource settings and developing countries, considering factors such as limited internet connectivity, computational resources, and technical expertise.

By pursuing these avenues for future work, we can further advance the field of plant disease detection, contribute to sustainable agriculture, and address the evolving challenges facing global food production and security. Our project lays the foundation for continued innovation and collaboration in leveraging machine learning for agricultural applications, with the ultimate goal of ensuring a healthy and resilient food supply for future generations.

## **CHAPTER-9**

### **REFERENCES**

- [1] Robert G. de Luna, Elmer P. Dadios, Argel A. Bandala, “Automated Image Capturing System for Deep Learning-based Tomato Plant Leaf Disease Detection and Recognition,” International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD) 2019.
- [2] Suma VR Amog Shetty, Rishab F Tated, Sunku Rohan, Triveni S Pujar, “CNN based Leaf Disease Identification and Remedy Recommendation System,” IEEE conference paper 2019.
- [3] Peng Jiang, Yuehan Chen, Bin Liu, Dongjian He, Chunquan Liang, “Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolution Neural Networks,” IEEE ACCESS 2019.
- [4] Geetharamani, Arun Pandian, “Identification of plant leaf diseases using a nine-layer deep convolution neural network,” Computers and Electrical Engineering 76 (2019).
- [5] Robert G. de Luna, Elmer P. Dadios, Argel A. Bandala, “Automated Image Capturing System for Deep Learning-based Tomato Plant Leaf Disease Detection and Recognition,” Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference.
- [6] Omkar Kulkarni, “Crop Disease Detection Using Deep Learning,” IEEE access 2018.
- [7] Abirami Devaraj, Karunya Rathan, Sarvepalli Jaahnavi and K Indira, “Identification of Plant Disease using Image Processing Technique,” International Conference on Communication and Signal Processing, IEEE 2019.
- [8] Velamakanni Sahithya, Brahmadevara Saivihari, Vellanki Krishna Vamsi, Parvathreddy Sandeep Reddy and Karthigha Balamurugan, “GUI based Detection of Unhealthy Leaves using Image Processing Techniques,” International Conference on Communication and Signal Processing 2019.

- [9] Balakrishna K Mahesh Rao, "Tomato Plant Leaves Disease Classification Using KNN and PNN," International Journal of Computer Vision and Image Processing 2019.
- [10] Masum Aliyu Muhammad Abdu, Musa Mohd Mokji, Usman Ullah Sheikh, Kamal Khalil, "Automatic Disease Symptoms Segmentation Optimized for Dissimilarity Feature extraction in Digital Photographs of Plant Leaves," IEEE 15th International Colloquium on Signal Processing & its Applications 2019.
- [11] <http://dx.doi.org/10.17632/tywbtsjrjv/1> (J & GOPAL, 2019), A. P., & GOPAL, G. (2019), <https://data.mendeley.com/datasets/tywbtsjrjv/1>. Retrieved, <http://dx.doi.org/10.17632/tywbtsjrjv/1>.
- [12] Suja Radha, "Leaf Disease Detection using Image Processing," Article in Journal of Chemical and Pharmaceutical Sciences, March 2017.
- [13] Sneha Patel, U.K Jaliya, Pranay Patel, "A Survey on Plant Leaf Disease Detection," International Journal for Modern Trends in Science and Technology, April 2020.
- [14] Priyanka Soni ,Rekha Chahar, "A Segmentation Improved Robust PNN Model for Disease Identification in Different Leaf Images," 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016).
- [15] S. Arivazhagan, R. Newlin Shebiah S. Ananthi, S. Vishnu Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," CIGR Journal, March 2013.
- [16] Adedamola Adedaja & Pius Adewale Owolawi & Temitope Mapayi, "Deep Learning Based on NAS Net for Plant Disease Recognition Using Leave Images," 2018.
- [17] Prajwala TM, Alla Pranathi, Kandiraju Sai Ashritha, Nagaratna B. Chittaragi, Shashidhar G. Koolagudi, "Tomato Leaf Disease Detection using Convolutional Neural Networks," Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3), 2018.
- [18] VGG16 - Convolutional Network for Classification and Detection, 21 Nov, 2018 <https://neurohive.io/en/popular-networks/vgg16/>

- [19] “Python Numpy Tutorial | Learn Numpy Arrays with Examples,” Edureka, Jul. 14, 2017. <https://www.edureka.co/blog/python-numpy-tutorial/> (accessed Apr. 21, 2020).
- [20] J. Han and C. Moraga, “The Influence of the Sigmoid Function Parameters on the Speed of Back propagation Learning,” in Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation, Berlin, Heidelberg, Jun. 1995, pp. 195–201, Accessed: Apr. 13, 2020. [Online].
- [21] “Scikit-learn Tutorial: Machine Learning in Python – Dataquest.” <https://www.dataquest.io/blog/sci-kit-learn-tutorial/> (accessed Apr. 21, 2020).
- [22] J. Brownlee, “Introduction to the Python Deep Learning Library TensorFlow,” Machine Learning Mastery, May 04, 2016. <https://machinelearningmastery.com/introductionpython-deep-learning-library-tensorflow/> (accessed Apr. 21, 2020).
- [23] “Project Jupyter” <https://www.jupyter.org> (accessed Apr. 19, 2020).