SNJB'S Late Sau. Kantabai Bhavarlalji Jain, College of Engineering, Chandwad

District: Nashik

Accredited with 'A' Grade by NAAC

Department of Computer Engineering

TITLE: Disease detection in grape leaf using Deep learning

Project Guide: **Prof.S.V.Sinha**

Presented by:

1. Gujrathi Siddhant Suhas
2. Lunawat Sujal Kailash
3. Chuttermutha Aatish Sanjay
4. Tatiya Suhani
5. Desai Kaveri

A.Y. 2022-2023

OUR TEAM

Prof.Swati Sinha
Project Guide

Gujarathi Siddhant

Lunawat Sujal

Chuttermutha Atish

Desai Kaveri

Tatiya Suhani

Agenda

01 INTRODUCTION

02 MOTIVATION

03 OBJECTIVE

04 WORKING

05 PROPOSED SYSTEM

06 ADVANTAGES

07 APPLICATION

07 HARDWARE AND SOFTWARE REQUIREMENT

08 REFERENCE

1. Introduction

01 **Agriculture plays a vital role in the world's economy and one of the basic need of human beings .In India it is considered as a major source of employment.**

02 **Nasik district in Maharashtra is a significant contributor to grape production. It is also called the "Grape Capital of India" due to favorable climatic conditions for all types of crop cultivation.**

03 **Grape leaves disease detection refers to the process of identifying and analyse diseases that affect grape leaves.**

04 **Using Advance technologies like CNNs(Convolutional Neural Networks) for the deep learning algorithms which is particularly well-suited for image classification task.**

2. Motivation

01 Grape leaves disease detection involves testing by experts, which is time consuming and expensive. Automated disease detection system using CNNs technology which reduces cost of disease detection, making it more accessible and affordable for farmers.

02 By detecting grape leaves diseases early, farmers can prevent unnecessary use of pesticides and fertilizers. This can help to improve environmental impact of agriculture.

03 This project can contribute to the development of new algorithms and tools that can be applied to other agriculture and industrial applications.

3. Objective

Objective 1

Collection of data from Kaggle website or from Browsers.

Objective 2

CNN Model Building using Tensorflow or Keras libraries .

Objective 3

Using Fast API's , we can load the images in our model.

Objective 4

Create a farmer friendly Application to scan the affected leaf.

Working

Phase:1 Data Collection

Web browsers

Phase:2 Data Pre-Processing

Data Cleaning

Data Resizing

Data augmentation

Phase:3 Model building and Export

CNN

Phase:4 Quantization

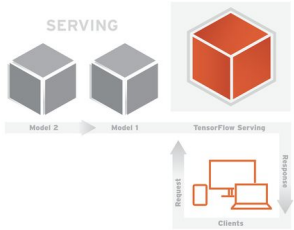
Tf model ⟶ Tf light model

Phase:5 Deployment

Tf light model ⟶

GCP

Phase:6 Deep Learning Operations



⟵ FastAPI

Phase:7 App Development

Flutter

In terms of Technology stack :

Model Building	TensorFlow	CNN (Convolutionary Neural Network)	Data Augmentation	
Backend Server	FastAPI	Tf serving		
Model Optimisation	Quantization	TensorFlow Lite		
Frontend And Deployment	Flutter	Deployment to GCP		

5. Proposed System

Data Cleaning and Preprocessing

Model Building

Tf lite model

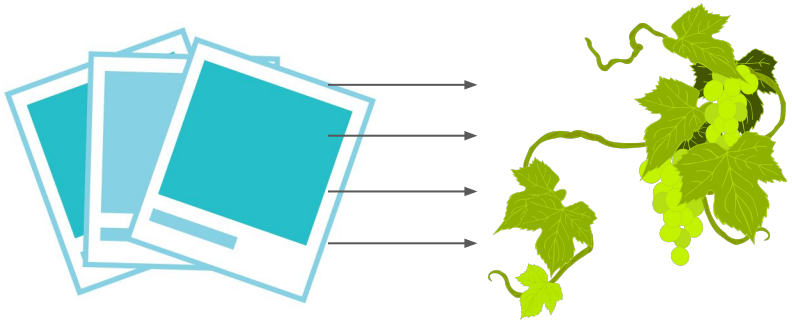
Data Augmentation

CNN

Quantization

Flutter

Google Cloud Functions



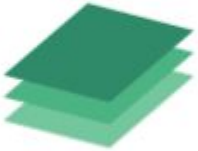
- Using kaggle

- Bing-image-downloader

```
from bing_image_downloader import downloader
downloader.download(query_string, limit=100,
output_dir='dataset')
```

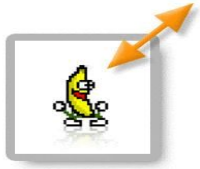
Phase:2 Data Pre-Processing (Creating dataset)

- We load all the images from directory and create a dataset using tensorflow.

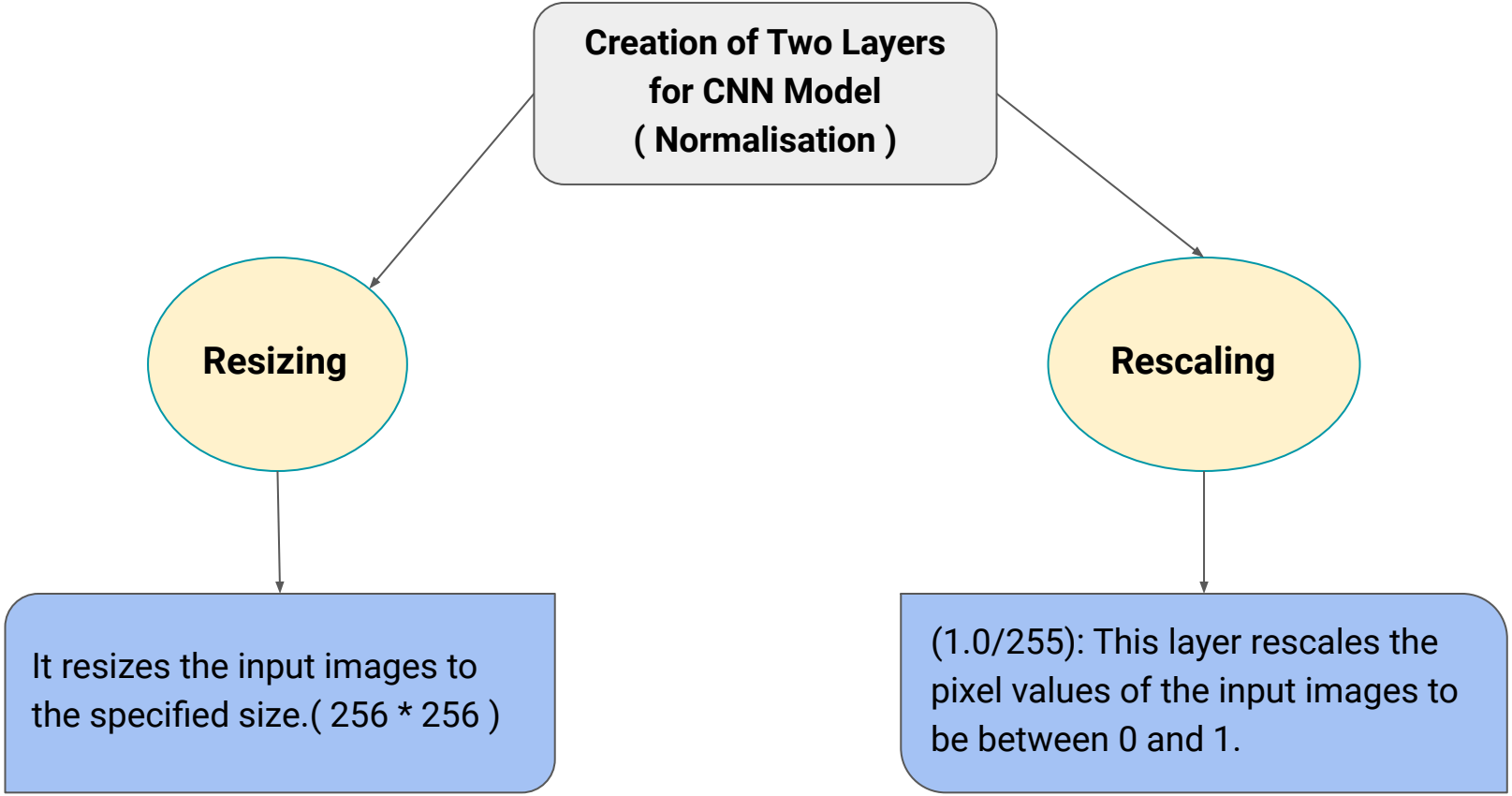


- Then we the shuffled the images randomly before being loaded into the dataset. This helps to avoid any biases in the order of the images.

- Images will be resized to 256 x 256 pixels and then loaded into the dataset.



- We are passing 32 images once into a model by defining batch size to 32.


Creation of Two Layers
for CNN Model
(Normalisation)
Resizing
Rescaling
It resizes the input images to the specified size.(256 * 256)
(1.0/255): This layer rescales the pixel values of the input images to be between 0 and 1.

Data Augmentation

It is a technique used to increase the size of the training dataset by applying various transformations to the original images.

It Consists

- *RandomFlip* → Increase the dataset with different orientation images
- *RandomRotation* → Randomly rotate the input images by a specified angle

Phase:3 Model building

Input Shape

Batch Size = 32
Image Size = 256 x 256
Number of Channels = 3 (RGB)

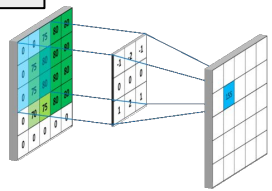
Classes

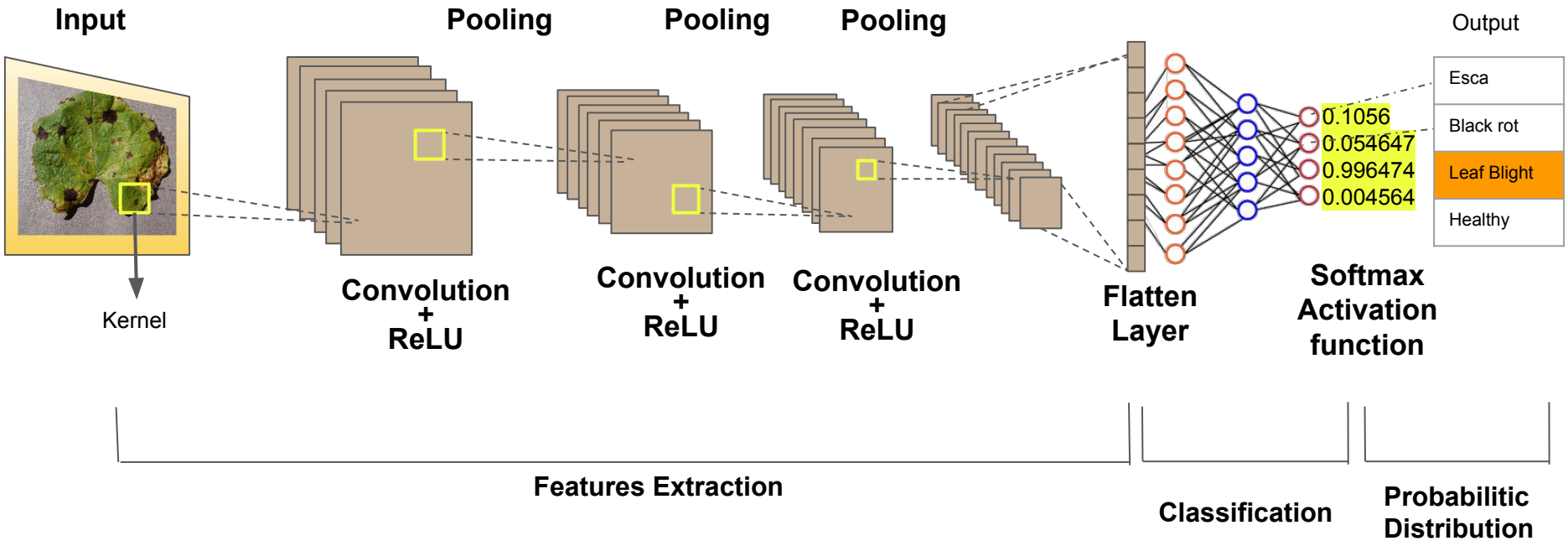
- Grape Black rot
- Grape Esca (Black Measles)
- Grape Leaf blight (Isariopsis Leaf Spot)
- Grape healthy'

Kernel Size

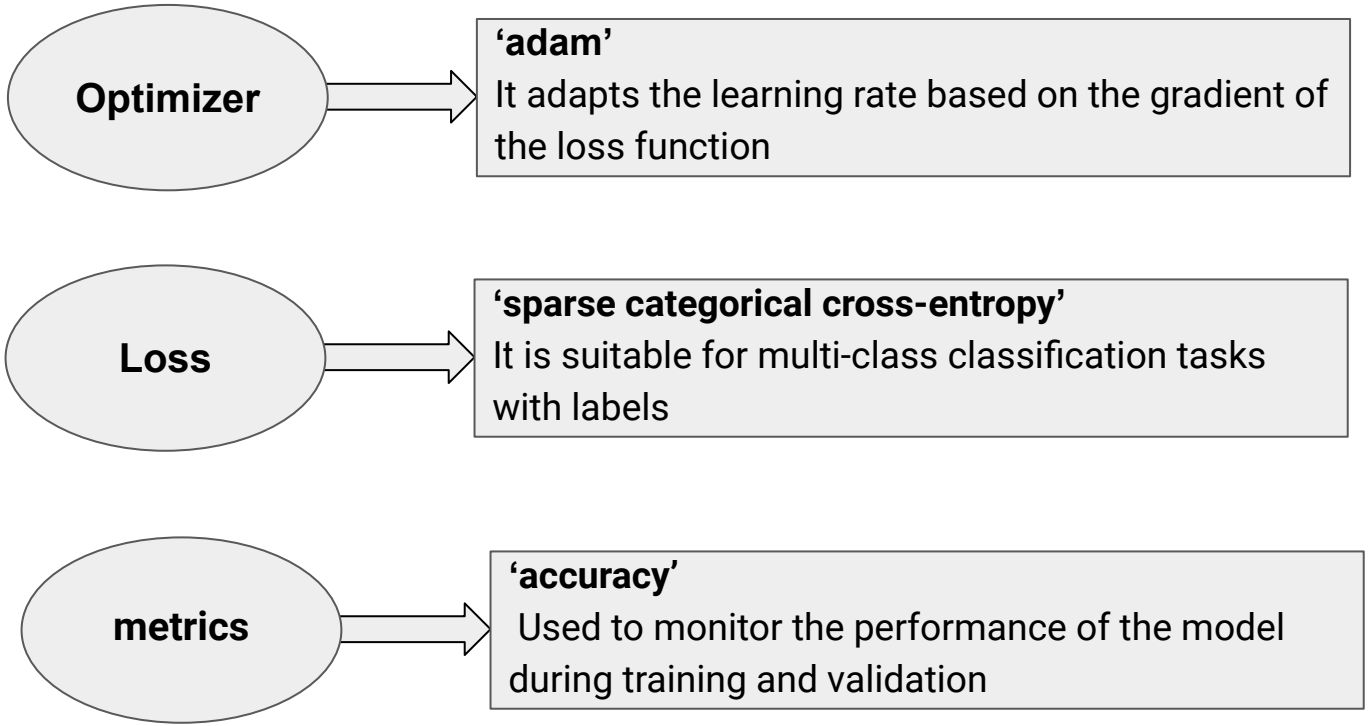
= 3 x 3

The size of the kernel determines the receptive field of the filter, or the region of the input image that the filter is sensitive to.




Input
Pooling
Pooling
Pooling
Output
Esca
Black rot
Leaf Blight
Healthy
0.1056
0.054647
0.996474
0.004564
Kernel
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Flatten Layer
Softmax Activation function
Features Extraction
Classification
Probabilitic Distribution

Phase:4 Model Compilation



Phase:5 Training The compiled model

TensorFlow

epochs	50 It specifies the number of times the entire training dataset is used for training the model.
Batch Size	32 specifies the number of samples that are propagated through the network at once.
Verbose	1 progress bars will be displayed during training
Validation Data	1 used for evaluating the performance of the model during training.

Phase:3 Model Export

TensorFlow

It is very easy to save the tf compiled model as..

- **Tflite**
- **.pb5**
- **saved_model.**

Phase:5 Deployment

- Deployment in the context of Google Cloud Platform (GCP) refers to the process of making application or services available and accessible for use on the cloud infrastructure provided by Google.
- GCP offers a wide range of services and tools for deploying various types of applications, ranging from simple web applications to complex, multi-tiered applications with distributed systems.
- Some common methods for deploying applications on GCP include:

  1.App Engine.

  2.Compute Engine

Phase 6: FASTAPI

1. FastAPI is a modern, high-performance web framework for building APIs with Python.

2.Easy to use and efficient.

3.Provides automatic response validation, dependency injection ,high-performance execution.

4.Supports authentication, authorization, data serialization, and integrates well with other Python libraries.

5.Automatically generates interactive API documentation, making it a popular choice for developers building APIs, microservices, and backend applications.

Phase:7 App Development : Flutter

1.App Development is one of the most important part of our project .

2.Flutter is used for the App development.

3.Flutter is an open-source UI software development kit created by Google

4.Used to develop cross-platform applications for Android, iOS, Linux, macOS, Windows, Google2.Should be simple and farmer friendly .

5.Should be simple and farmer friendly.

Advantages

1..Accuracy: CNNs have shown to be highly accurate in detecting diseases in images, including grape diseases. This is because CNNs are capable of detecting patterns in images that may not be visible to the human eye.

2.Speed: CNN can process large amount of images data quickly and accurately making it an efficient way to detect grape disease

3.Automation : CNN can automate the process of grape disease detection ,making possible detect disease in large vineyards and orchards without the need for human invention

Advantages

4.Early Detection : CNNs can help to detect disease early on,enabling farmers to take appropriate measures to containand treat the disease before it spreads

5 Cost Effective : CNN can reduce the cost of detecting grape diseases by automating the process and reducing the need for human labour

Applications

1.Early detection and prevention of diseases.

2.Precision farming:-
By using grape disease detection technologies, farmers can apply targeted treatments only where needed. This reduces the amount of pesticides and other treatments required, leading to more sustainable and environmentally-friendly farming practices.

3.Grape disease detection can help increase yields in the vineyard.
By detecting and treating diseases early, grape disease detection can help increase yields in the vineyard. This is because diseases can cause significant damage to the grapes, resulting in a lower yield.

List of Required Hardware, Software

Software Requirement :

1. Operating System : Windows 8 and above.
2. Anaconda Navigator
3. Python Libraries : Pandas,numpy,matplotlib,tenserflow,keras,opencv
4. Android Studio
5. flutter
6. Pycharm
7. Google collab

Hardware Requirement :

1. Smart Phone
2. Intel core i3 or more
3. Hard disk : 256GB
4. RAM : 4GB

Date : 11/5/23

References

1) https://vikaspedia.in/agriculture/crop-production/integrated-pest-managment/ipm-for-fruit-crops/ipm-strategies-for-grapes/grapes-diseases-and-symptoms
2) https://www.frontiersin.org/articles/10.3389/fpls.2020.00751/full
3) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7285655/
4) https://www.sciencedirect.com/science/article/pii/S2214317319301003
5) https://www.mdpi.com/2073-4395/11/11/2234
6) https://youtube.com/playlist?list=PLuhqtP7jdD8CD6rOWy20INGM44kULvrHu