National University
of Computer & Emerging Sciences

IntelliCART
(Intelligent RehriWala)

“Transforming Roadside Fruit Vendors with AI-Powered Fruit Classification and Customer Convenience”

Department of Computer Science, FAST-NUCES, Karachi, Pakistan

FYP1-EVALUATION

Project Team

Project Supervisor

Dr. Muhammad Farrukh Shahid

Project Members

Abdul Ahad Shaikh
20K-0477

Mohammad Basil Ali Khan
20K-0477

Syed Jodat Ali Naqvi
20K-0155

Introduction

Welcome To The Future Of Urban Fruit’s And Vegs Supply Chain Solution.

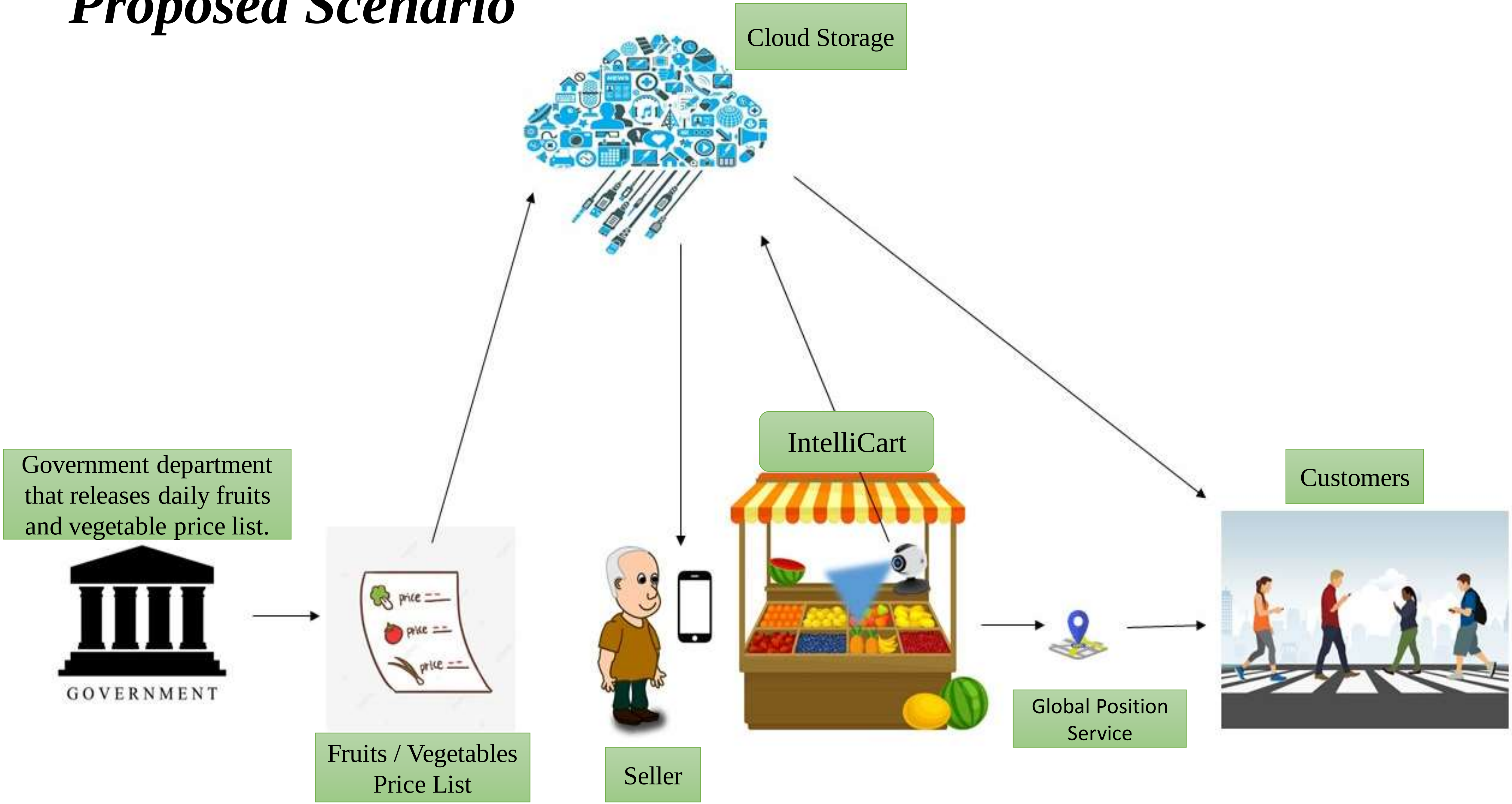
This Project Aims To Revolutionize The Vegetables And Fruits Retail Market, Especially The Roadside Vendors. Harnessing The Power Of AI, This Project Will Enable Customers To Identify Fresh Fruits And Vegs Just From Their Cell Phones And Locate Their Required Fruit.

Upgrading These Rehris Would Provide Convenience To Both Customers And Sellers Through Smooth Retail, Also Reducing Traffic Congestions.

Problem Statement

Fruit vending market is quite unmanaged, neither the prices are controlled by government nor the quality, thus customers face difficulty in finding and classifying healthy fruits, leading to compromised choices. This sector even lacks the proper infrastructure to facilitate customers. Considering these we are stepping up to solve these issues leveraging the power of AI.

Proposed Scenario



FYP-I Deliverables

- Dataset Collection
- Number of classes, Size of Dataset
- Data Annotation
- Impact of Different Models

Dataset Collection

- The dataset for fruits and vegetables was collected online platforms and through data collection drives.

- The following online platforms were used for online data collection:
  1. Kaggle
  2. Zenodo
  3. IEEE Dataport

  (references attached towards the end of presentation)

Dataset Collection - Drive

Classes in Dataset

Fruits	Classes
Banana	Firm
	Heavily Bruised
	Slightly Bruised
Apple	Blotch
	Healthy
	Rotten
	Scab
Orange	Greening
	Healthy
	Damaged

Vegetables	Classes
Tomato	Old
	Ripe
	Unripe
Green Chilli	Dried
	Old
	Ripe
	Damaged

Dataset

Class Name	Quantity			
Apple_blotch	**348**			
Apple_healthy	**652**			
Apple_rotten	**777**			

Dataset

Class Name	Quantity			
Apple_scab	**222**			
Banana_firm	**736**			
Banana_heavilybruised	**678**			

Dataset

Class Name	Quantity			
Banana_slightybruised	793			
GreenChilli_dried	500			
Greenchilli_old	259			

Dataset

Class Name	Quantity			
GreenChilli_ripe	201			
GreenChilli_damaged	138			
Orange_greening	699			

Dataset

Class Name	Quantity			
Orange_healthy	**378**			
Orange_rotten	**1224**			
Tomato_damaged	**421**			

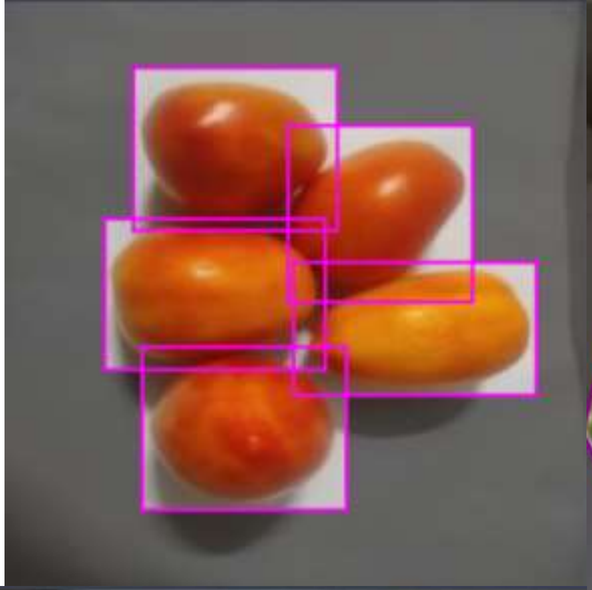
Dataset

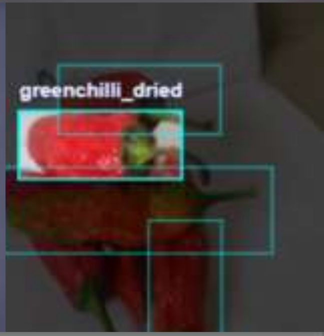
Class Name	Quantity			
Tomato_old	1234			
Tomato_ripe	955			
Tomato_unripe	546			

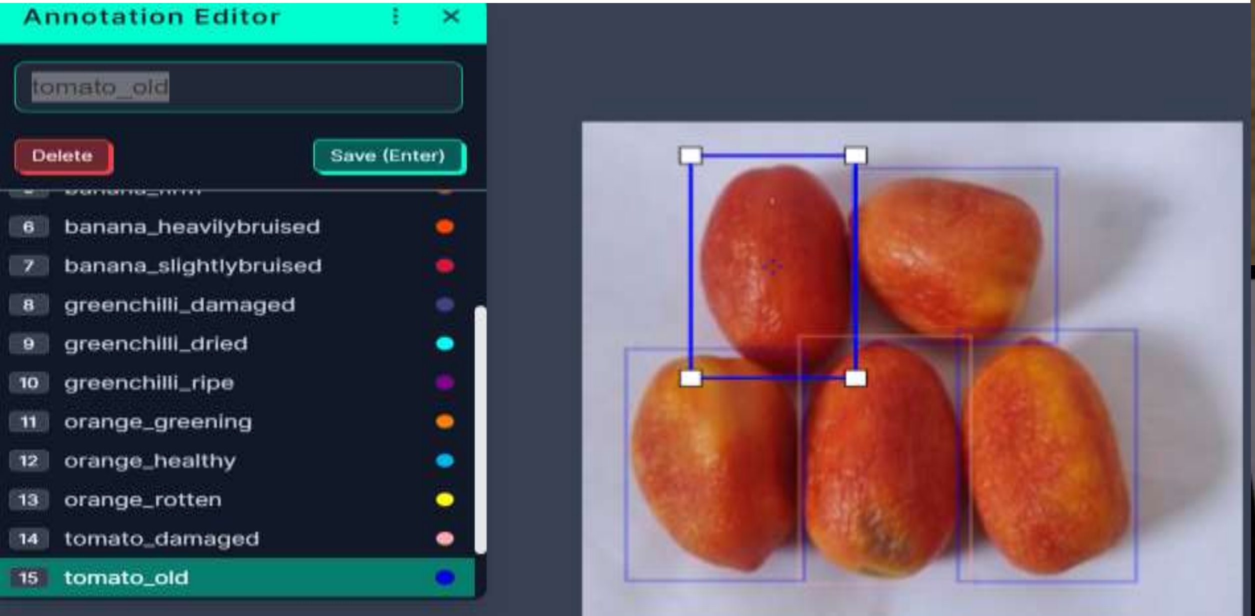
Data Annotation

- Data annotation is the process of labeling or tagging data to make it understandable or usable for machine learning algorithms.

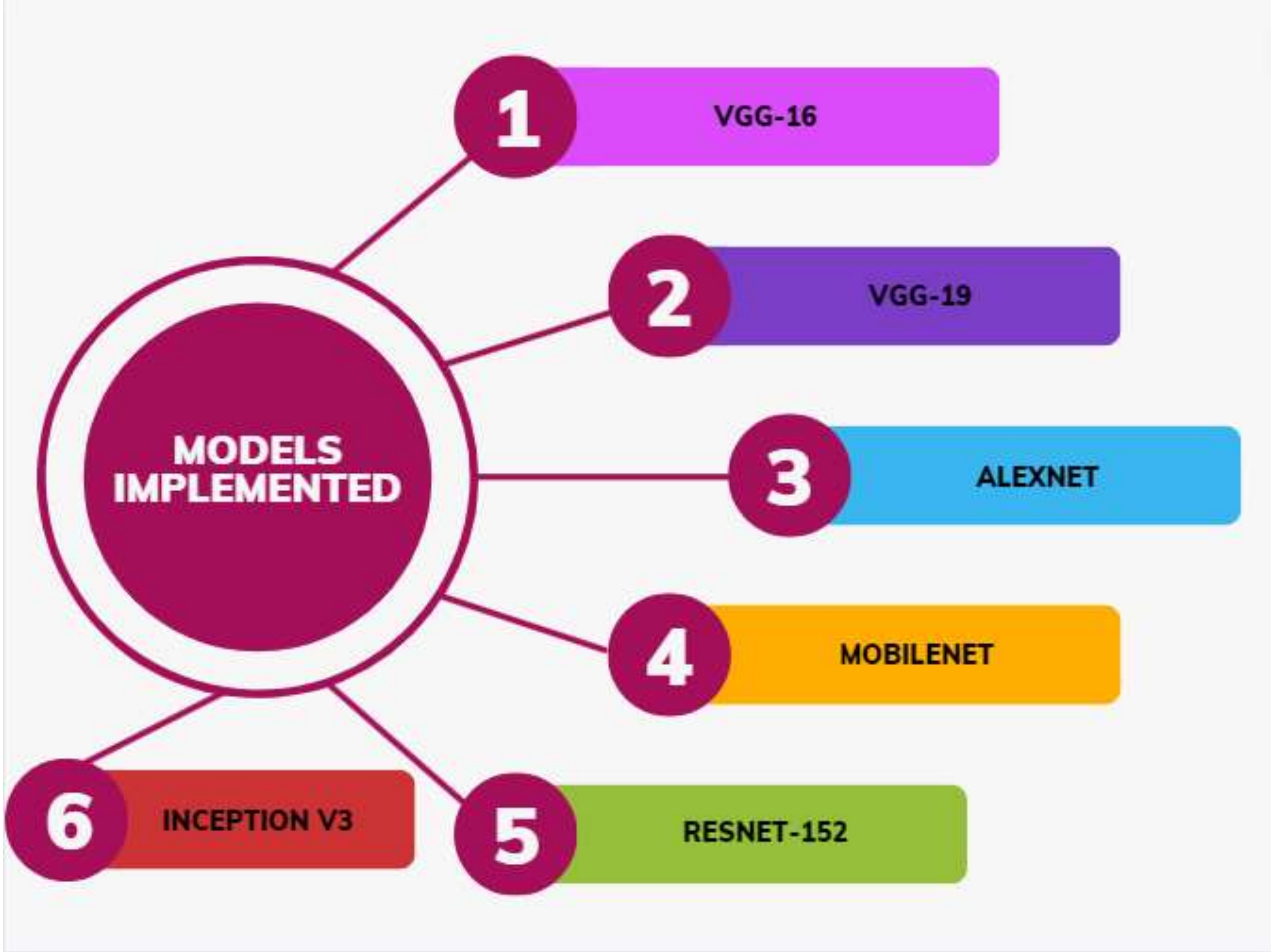
- The following platforms were used to annotate our data:
  - Roboflow
  - Labellmg







Models Implemented

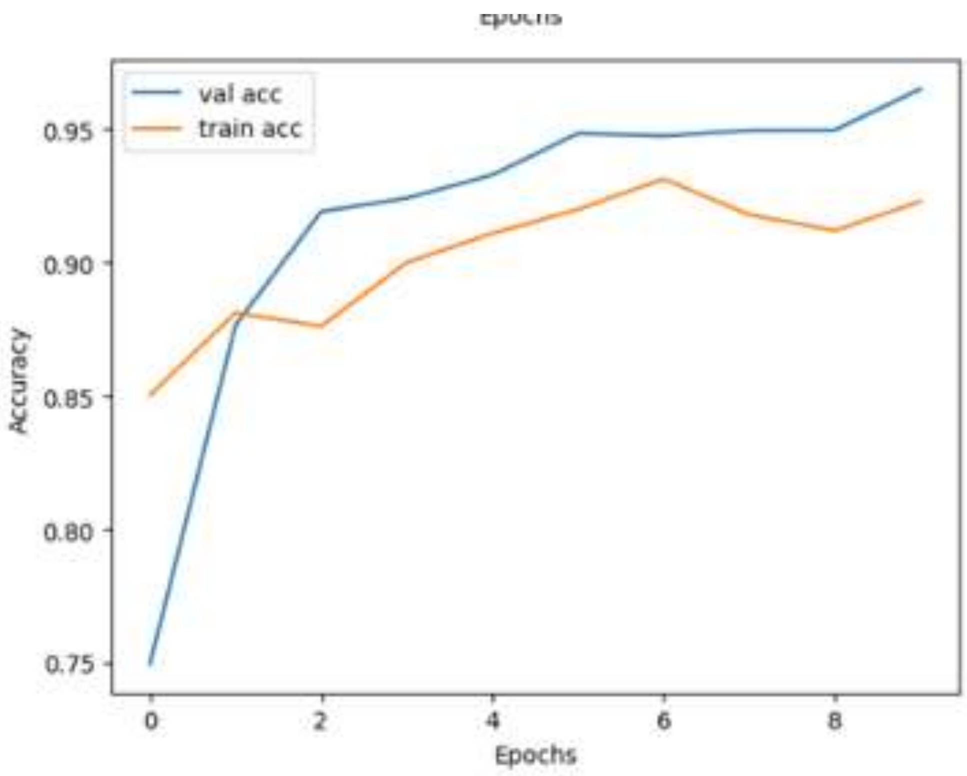


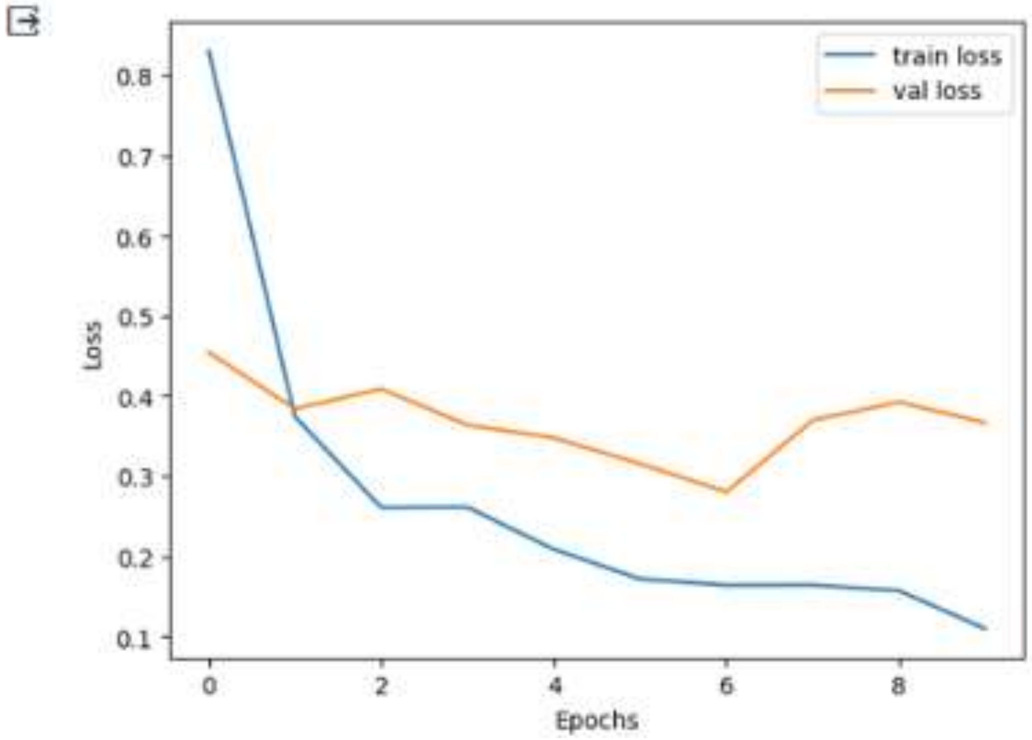
VGG-16

- ➢ VGG-16 is a convolutional neural network that is 16 layers deep and has an image input size of 224 by 224.
- ➢ VGG-16 contains 16 layers, 13 convolutional layers, 5 max-pooling layers and 3 fully connected layers.
- ➢ **We implemented VGG-16 on our dataset and achieved an accuracy of 92%.**

	precision	recall	f1-score	support
Apple_blotch	0.79	0.64	0.71	64
Apple_healthy	0.88	0.90	0.89	137
Apple_rotten	0.95	0.93	0.94	136
Apple_scab	1.00	0.95	0.98	42
Banana_firm	0.97	0.98	0.98	148
Banana_heavilybruised	0.94	0.92	0.93	136
Banana_slightlybruised	0.91	0.94	0.93	159
GreenChilli_damaged	1.00	1.00	1.00	27
GreenChilli_dried	0.99	0.94	0.96	100
GreenChilli_old	0.83	0.94	0.88	52
GreenChilli_ripe	0.69	0.88	0.77	40
GreenChilli_unripe	0.93	0.62	0.75	45
Orange_greening	1.00	0.81	0.89	136
Orange_healthy	0.86	0.94	0.90	71
Orange_rotten	0.88	0.99	0.93	247
Tomato_damaged	0.00	0.00	0.00	4
Tomato_old	0.93	0.98	0.96	247
Tomato_ripe	0.97	0.94	0.95	190
Tomato_unripe	1.00	0.97	0.98	29
accuracy			0.92	2010
macro avg	0.87	0.86	0.86	2010
weighted avg	0.92	0.92	0.92	2010

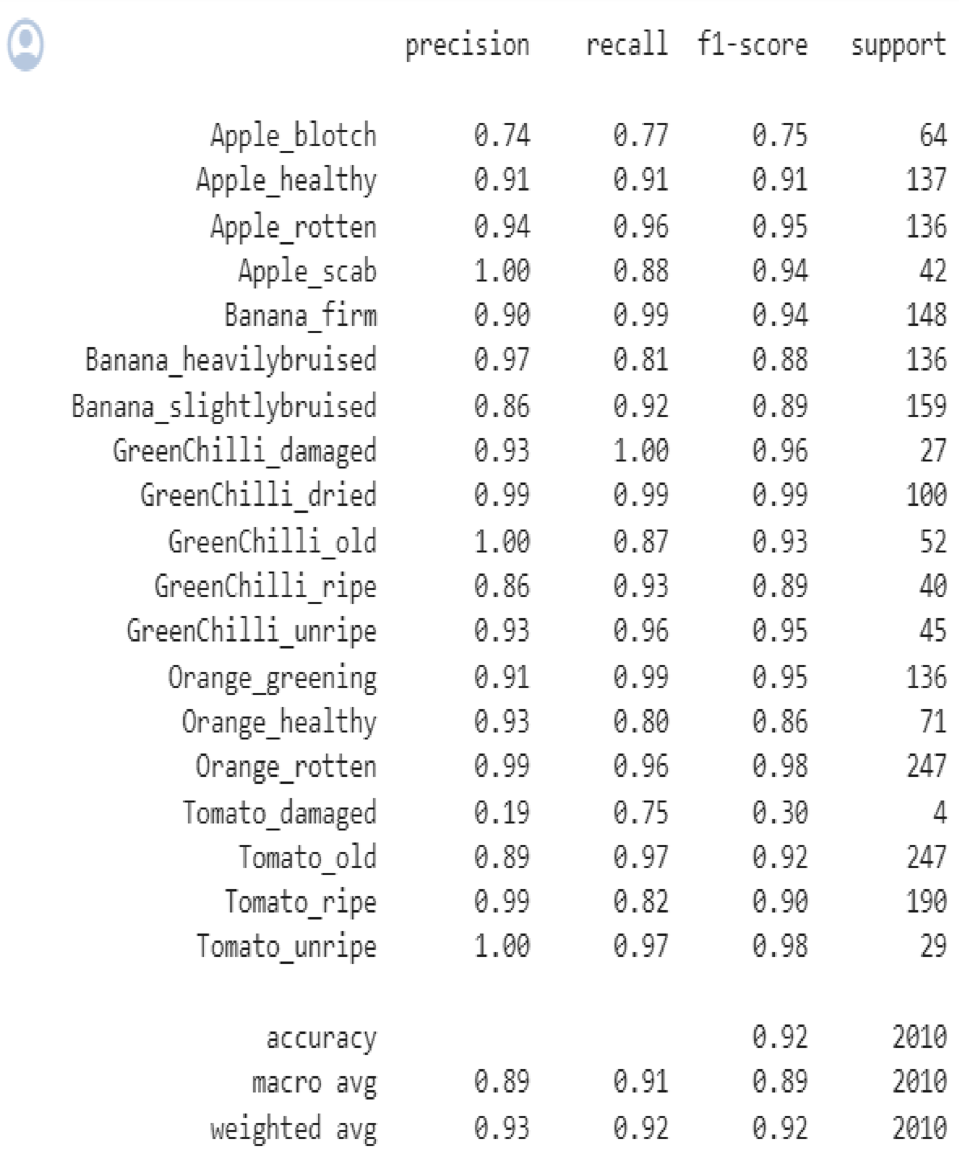
VGG-16 Accuracy and Loss





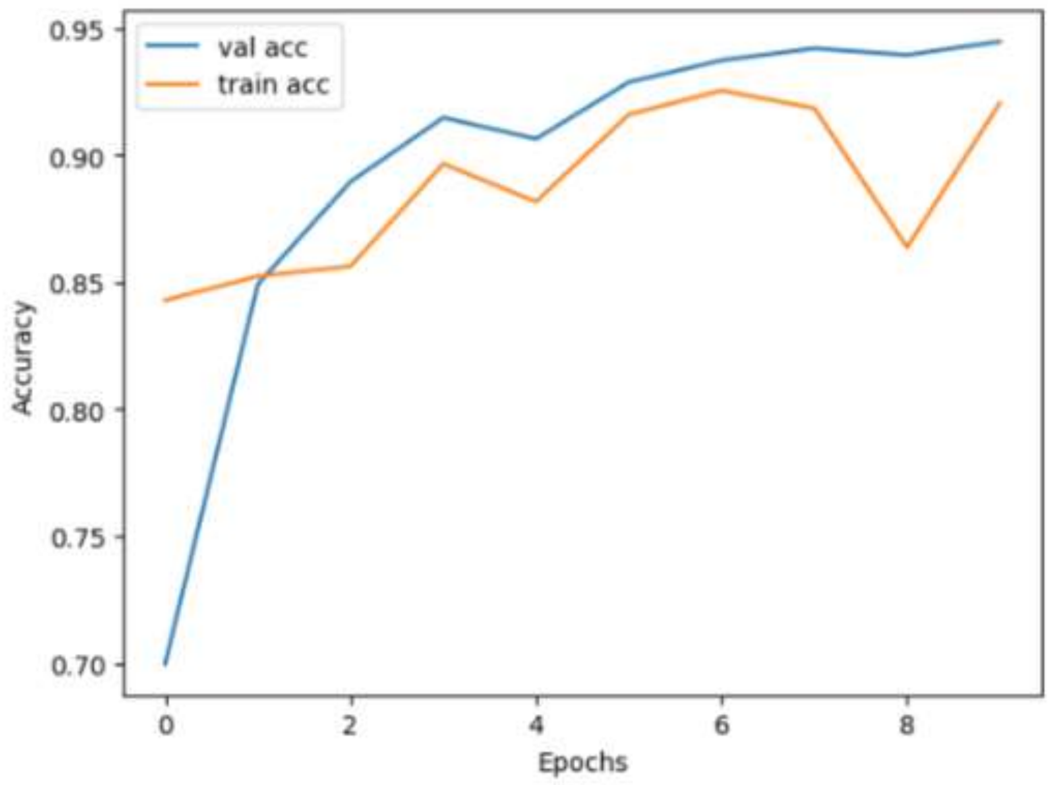
VGG-19

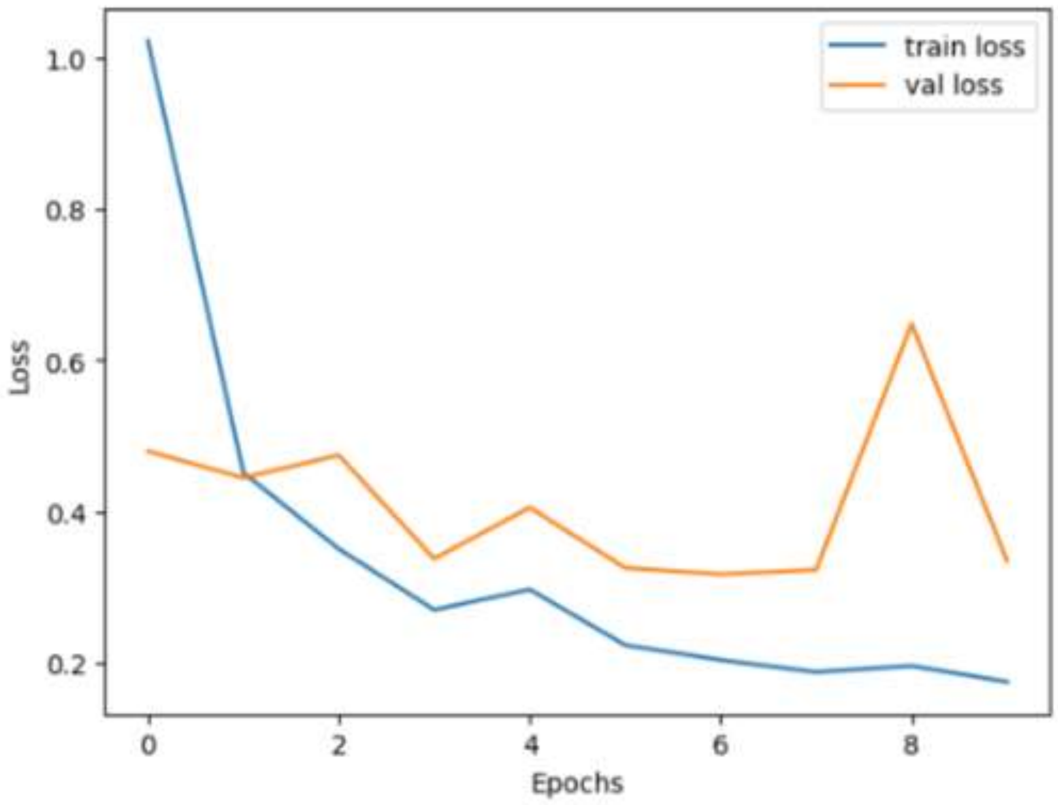
- VGG-16 is a convolutional neural network used for image classification that is 19 layers deep and has an image input size of 224 by 224.
- VGG-19 consists of 19 layers with 16 convolutional layers, 5 max-pooling layers and 3 fully connected layers.
- **We implemented VGG-19 on our dataset and achieved an accuracy of 92%.**



	precision	recall	f1-score	support
Apple_blotch	0.74	0.77	0.75	64
Apple_healthy	0.91	0.91	0.91	137
Apple_rotten	0.94	0.96	0.95	136
Apple_scab	1.00	0.88	0.94	42
Banana_firm	0.90	0.99	0.94	148
Banana_heavilybruised	0.97	0.81	0.88	136
Banana_slightlybruised	0.86	0.92	0.89	159
GreenChilli_damaged	0.93	1.00	0.96	27
GreenChilli_dried	0.99	0.99	0.99	100
GreenChilli_old	1.00	0.87	0.93	52
GreenChilli_ripe	0.86	0.93	0.89	40
GreenChilli_unripe	0.93	0.96	0.95	45
Orange_greening	0.91	0.99	0.95	136
Orange_healthy	0.93	0.80	0.86	71
Orange_rotten	0.99	0.96	0.98	247
Tomato_damaged	0.19	0.75	0.30	4
Tomato_old	0.89	0.97	0.92	247
Tomato_ripe	0.99	0.82	0.90	190
Tomato_unripe	1.00	0.97	0.98	29
accuracy			0.92	2010
macro avg	0.89	0.91	0.89	2010
weighted avg	0.93	0.92	0.92	2010

VGG-19 Accuracy and Loss





AlexNet

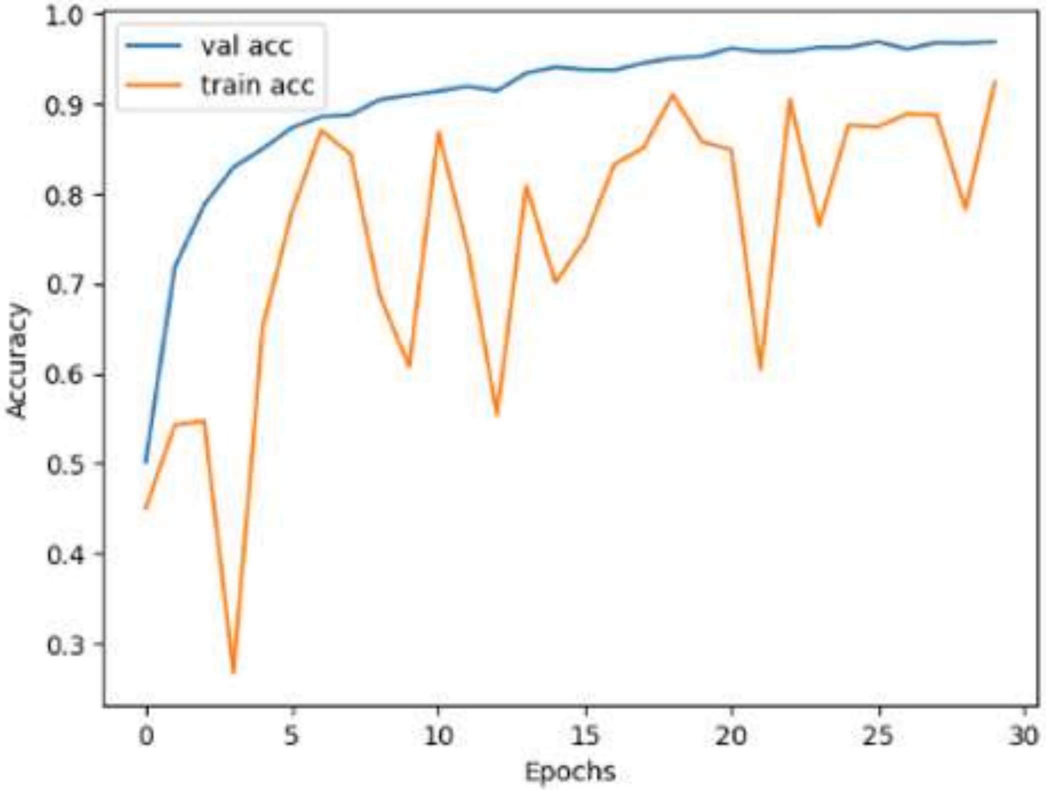
- AlexNet is a convolutional neural network that is 8 layers deep and has an image input size of 227 by 227.

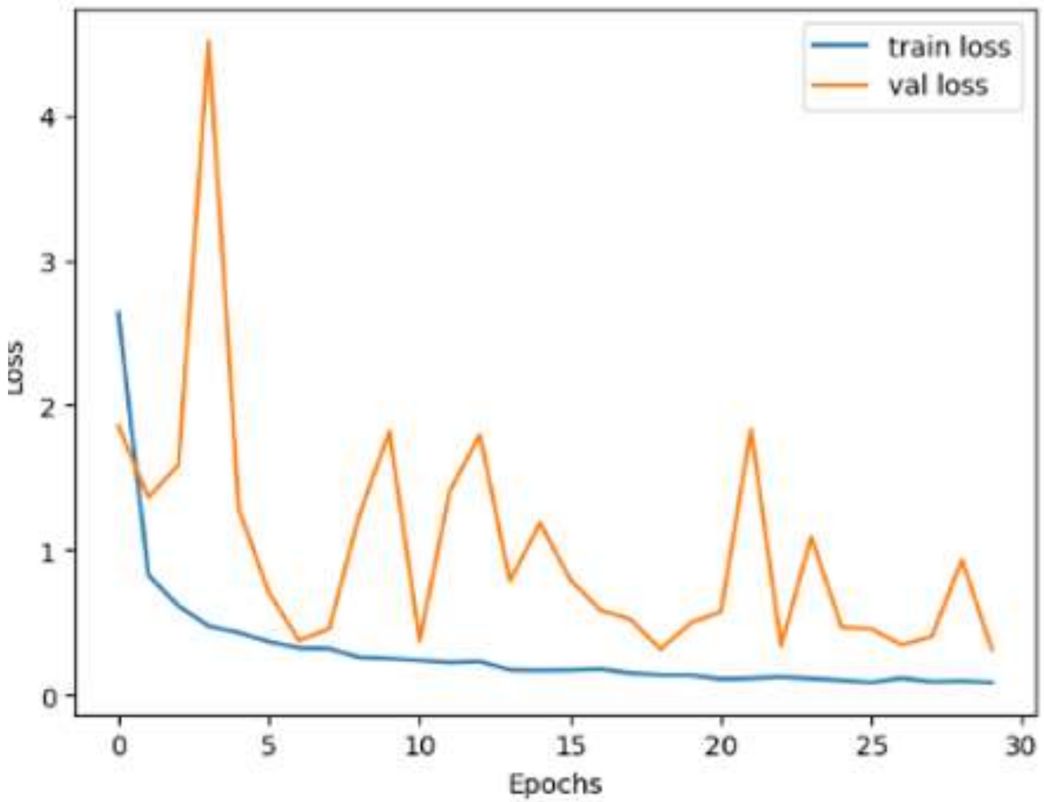
- AlexNet contained eight layers; the first five were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers. It uses the ReLU activation function,

- **We implemented AlexNet on our dataset and achieved an accuracy of 92%.**

	precision	recall	f1-score	support
Apple_blotch	0.71	0.64	0.67	64
Apple_healthy	0.84	0.91	0.88	137
Apple_rotten	0.99	0.92	0.95	136
Apple_scab	0.97	0.93	0.95	42
Banana_firm	0.94	0.97	0.95	148
Banana_heavilybruised	0.90	0.93	0.92	136
Banana_slightlybruised	0.92	0.86	0.89	159
GreenChilli_damaged	0.84	1.00	0.92	27
GreenChilli_dried	0.91	0.99	0.95	100
GreenChilli_old	0.98	0.81	0.88	52
GreenChilli_ripe	0.80	0.90	0.85	40
GreenChilli_unripe	0.93	0.82	0.87	45
Orange_greening	0.90	0.96	0.93	136
Orange_healthy	0.93	0.77	0.85	71
Orange_rotten	0.96	0.98	0.97	247
Tomato_damaged	0.00	0.00	0.00	4
Tomato_old	0.95	0.98	0.96	247
Tomato_ripe	0.98	0.95	0.97	190
Tomato_unripe	0.97	1.00	0.98	29
accuracy			0.92	2010
macro avg	0.86	0.86	0.86	2010
weighted avg	0.92	0.92	0.92	2010

AlexNet Accuracy and Loss





MobileNet

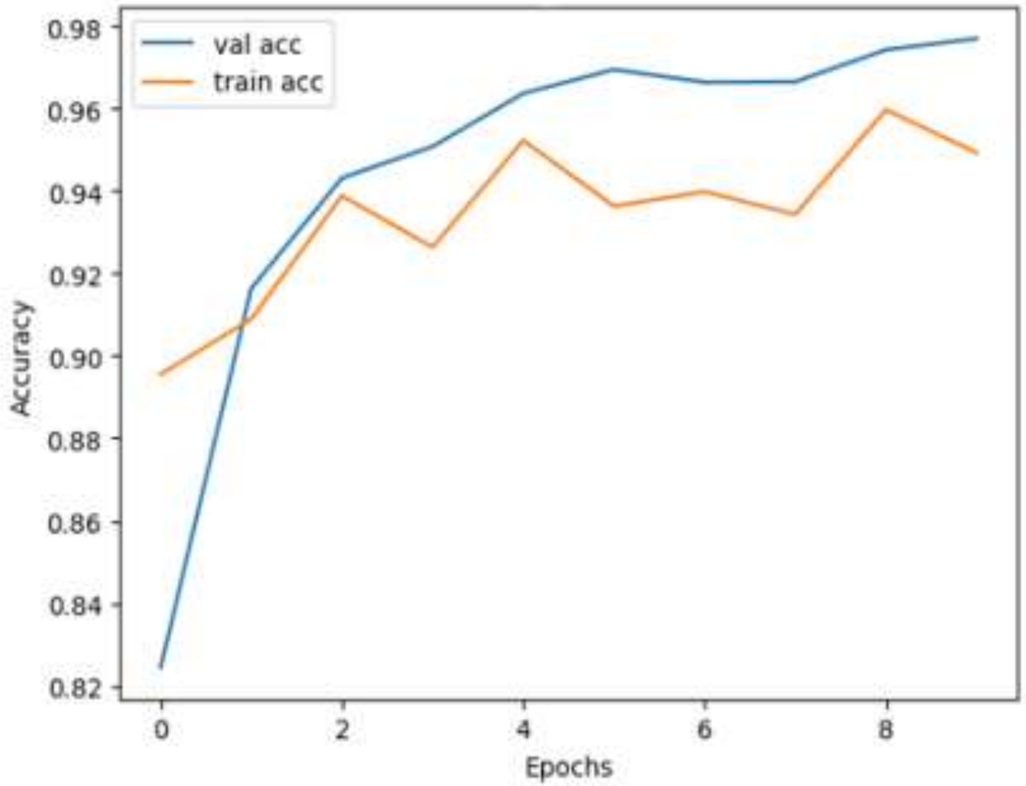
- MobileNet is a convolutional neural network that uses depthwise separable convolutions and has an input size of 224 by 224.

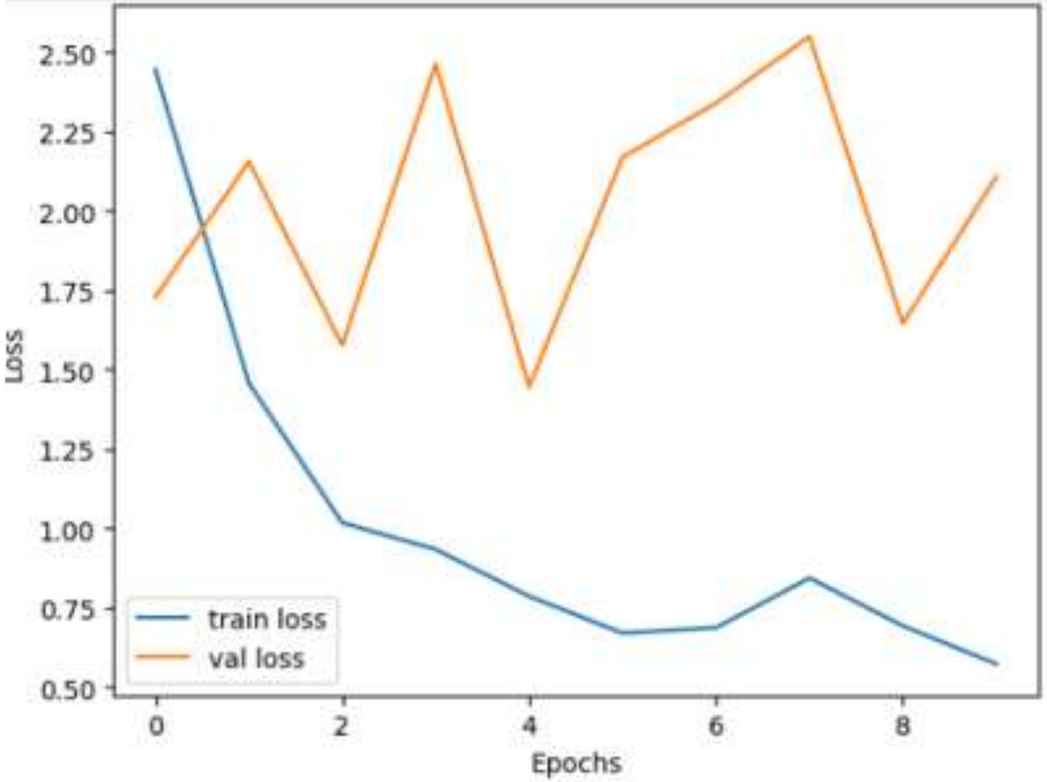
- Mobile Net split the convolution into a **3x3 depth-wise convolution** and a **1x1 pointwise convolution**, as shown in the figure.

- **We implemented MobileNet on our dataset and achieved an accuracy of 95%.**

	precision	recall	f1-score	support
Apple_blotch	0.82	0.70	0.76	64
Apple_healthy	0.91	0.93	0.92	137
Apple_rotten	0.96	0.99	0.97	136
Apple_scab	1.00	1.00	1.00	42
Banana_firm	0.90	1.00	0.95	148
Banana_heavilybruised	0.94	0.93	0.93	136
Banana_slightlybruised	0.95	0.88	0.91	159
GreenChilli_damaged	1.00	1.00	1.00	27
GreenChilli_dried	1.00	0.95	0.97	100
GreenChilli_old	0.98	0.98	0.98	52
GreenChilli_ripe	0.84	0.93	0.88	40
GreenChilli_unripe	0.93	0.91	0.92	45
Orange_greening	0.89	0.99	0.94	136
Orange_healthy	0.98	0.87	0.93	71
Orange_rotten	1.00	0.96	0.98	247
Tomato_damaged	0.50	0.25	0.33	4
Tomato_old	0.98	0.98	0.98	247
Tomato_ripe	0.96	0.99	0.98	190
Tomato_unripe	1.00	0.97	0.98	29
accuracy			0.95	2010
macro avg	0.92	0.91	0.91	2010
weighted avg	0.95	0.95	0.95	2010

MobileNet Accuracy and Loss





ResNet-152

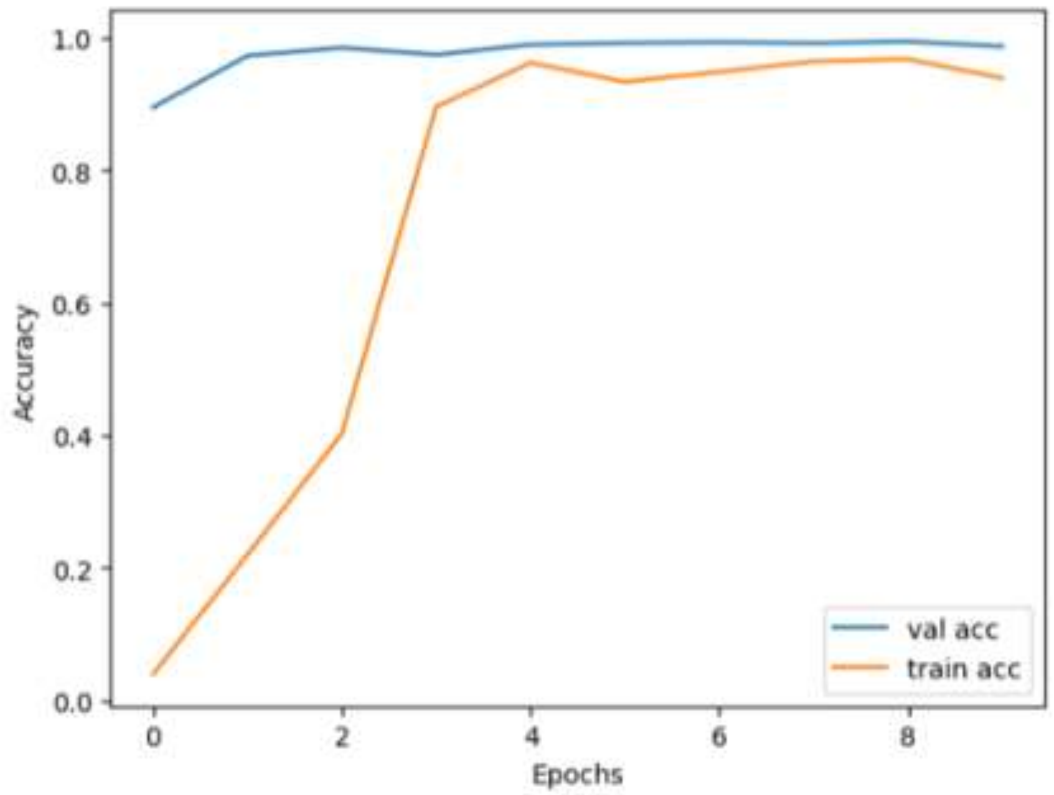
- ResNet-152 is a convolutional neural network that is 152 layers deep and has an image input size of 224 by 224 or 229 by 229.

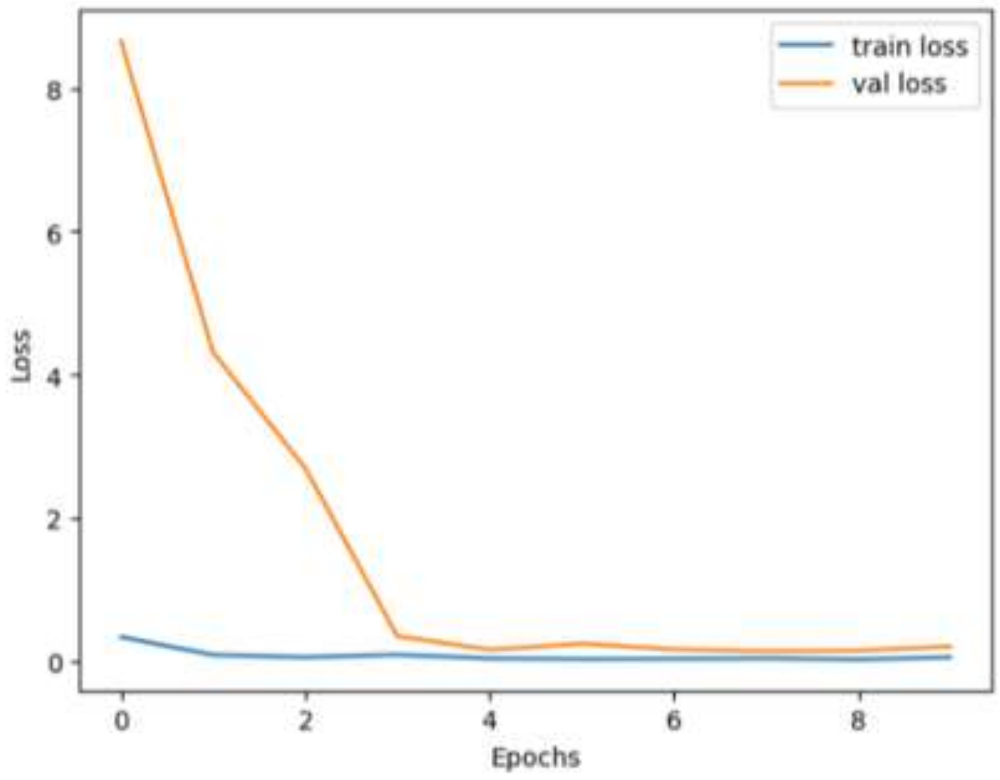
- ResNet-152 contained 152 layers; the first 151 were convolutional layers, there are no Max-pool layers, down sampling is achieved through stride of 2 with certain layers, followed by a global average pooling layer and a fully connected layer.

- **We implemented ResNet-152 on our dataset and achieved an accuracy of 94%.**

	precision	recall	f1-score	support
Apple_blotch	0.96	0.67	0.79	64
Apple_healthy	0.87	0.99	0.92	137
Apple_rotten	0.81	0.99	0.89	136
Apple_scab	1.00	0.83	0.91	42
Banana_firm	0.98	0.98	0.98	148
Banana_heavilybruised	0.96	0.94	0.95	136
Banana_slightlybruised	0.93	0.94	0.94	159
GreenChilli_damaged	1.00	0.93	0.96	27
GreenChilli_dried	1.00	1.00	1.00	100
GreenChilli_old	0.98	0.96	0.97	52
GreenChilli_ripe	0.88	0.95	0.92	40
GreenChilli_unripe	0.96	0.96	0.96	45
Orange_greening	0.87	1.00	0.93	136
Orange_healthy	0.94	0.87	0.91	71
Orange_rotten	0.99	0.83	0.90	247
Tomato_damaged	0.22	0.50	0.31	4
Tomato_old	0.98	0.97	0.97	247
Tomato_ripe	0.99	0.99	0.99	190
Tomato_unripe	1.00	1.00	1.00	29
accuracy			0.94	2010
macro avg	0.91	0.91	0.90	2010
weighted avg	0.95	0.94	0.94	2010

ResNet-152 Accuracy and Loss





Inception-V3

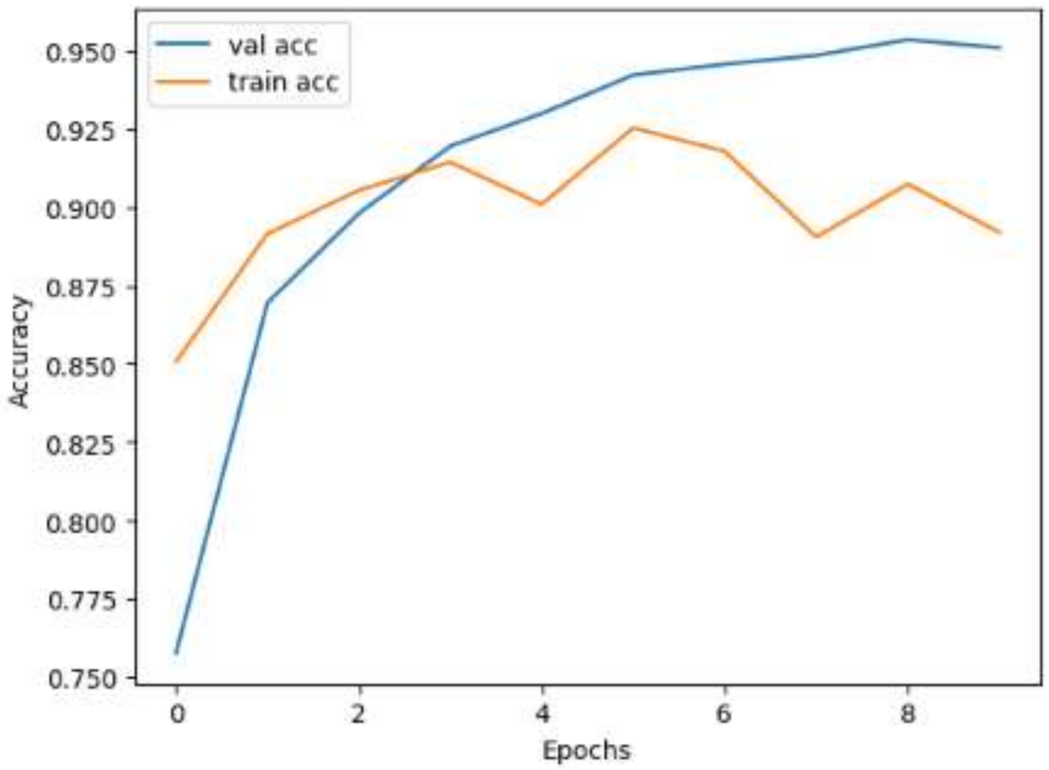
- ➢ Inception-V3 is a convolutional neural network used for image classification that is 48 layers deep and has an image input size of 229 by 229.

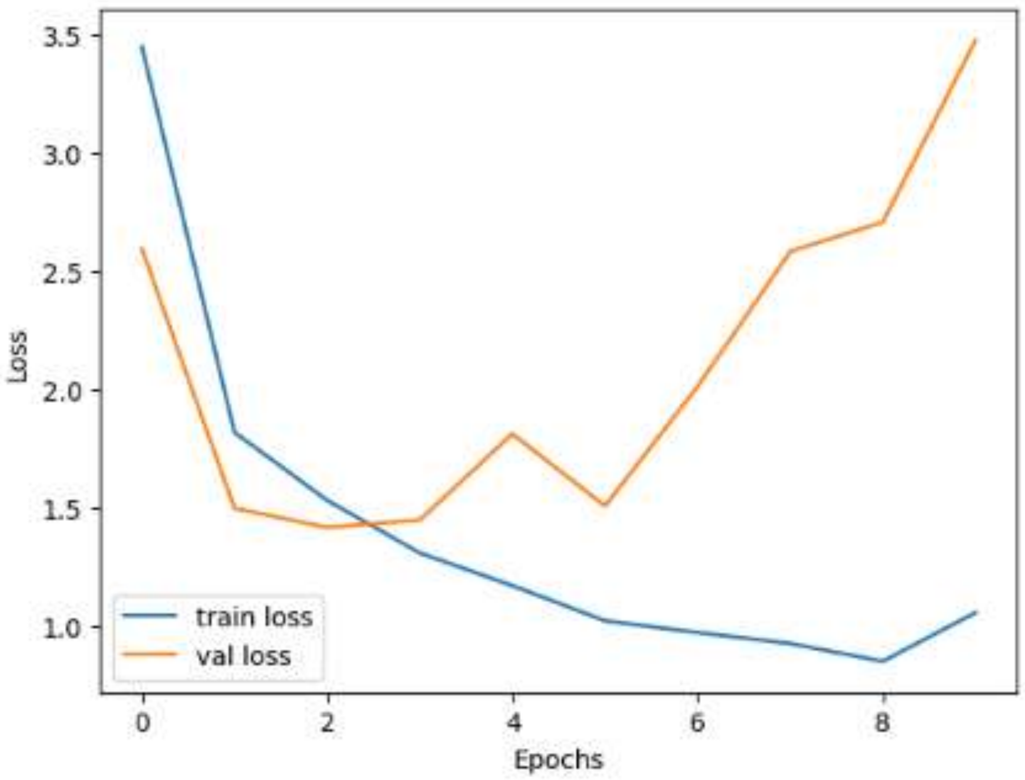
- ➢ Inception-V3 has better accuracy and less computational cost compared to previous inception versions.

- ➢ **We implemented Inception-V3 on our dataset and achieved an accuracy of 89%.**

	precision	recall	f1-score	support
Apple_blotch	0.86	0.39	0.54	64
Apple_healthy	0.83	0.97	0.89	137
Apple_rotten	0.92	0.93	0.93	136
Apple_scab	0.98	0.98	0.98	42
Banana_firm	1.00	0.89	0.94	148
Banana_heavilybruised	0.86	0.96	0.91	136
Banana_slightlybruised	0.94	0.92	0.93	159
GreenChilli_damaged	1.00	0.78	0.88	27
GreenChilli_dried	1.00	0.96	0.98	100
GreenChilli_old	1.00	0.73	0.84	52
GreenChilli_ripe	0.85	0.88	0.86	40
GreenChilli_unripe	0.66	1.00	0.80	45
Orange_greening	0.95	0.87	0.91	136
Orange_healthy	0.89	0.72	0.80	71
Orange_rotten	0.86	1.00	0.93	247
Tomato_damaged	0.17	0.50	0.25	4
Tomato_old	0.99	0.76	0.86	247
Tomato_ripe	0.79	0.99	0.88	190
Tomato_unripe	1.00	0.97	0.98	29
accuracy			0.89	2010
macro avg	0.87	0.85	0.85	2010
weighted avg	0.91	0.89	0.89	2010

Inception-V3 Accuracy and Loss





Why we chose MobileNet? (Best Accuracy)

Many real-life classification applications, such as robotics, autonomous driving, smartphone, etc, the classification task is highly constrained by the computational resources that are available.
The problem thus becomes to pursue the optimal accuracy subject to a limited computational budget (i.e. memory and/or MFLOPs).

MobileNet is a type of convolutional neural network (CNN) that has been specifically designed for use on mobile devices with limited computational resources. It has the following advantages:

1. **Lightweight architecture**: MobileNet has a smaller number of parameters compared to other deep learning models, which makes it ideal for deployment on mobile devices and other embedded systems with limited resources.

2. **Speed**: Because MobileNet has fewer parameters, it is faster to train and run than other deep learning models. This makes it an ideal choice for real-time applications such as object detection and image recognition.

3. **Accuracy**: Despite its lightweight architecture, MobileNet can achieve high accuracy on a variety of tasks, including image classification and object detection. It has been shown to perform well on benchmark datasets such as ImageNet.

4. **Compatibility**: MobileNet is compatible with a wide range of deep learning frameworks, including TensorFlow, Keras, and PyTorch. This makes it easy to integrate MobileNet into existing deep learning workflows and frameworks.

MobileNet: Inter Quartile range

In deep learning, the interquartile range (IQR) plays a role in data preprocessing and outlier detection. During data preprocessing, which is a common step in deep learning, input data is often normalized or scaled. The interquartile range can be used as a measure of the spread or variability within the data. By considering the IQR, one can guide the normalization or scaling operations, ensuring that the range of values in different features or samples is consistent. This helps in preventing any feature from dominating the learning process due to its wider range.

In the evaluation of a binary classification model, several metrics provide insights into its performance. These metrics include True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), and False Negative Rate (FNR).

TPR average	0.9058942945130736
FPR average	0.002858193302300445
TNR average	0.9971418066976995
FNR average	0.0941057054869264

The average TPR value is 0.905, indicating that, on average, the model accurately classifies around 90.5% of positive instances. The average FPR value is 0.0028, indicating that, on average, the model misclassifies approximately 0.28% of negative instances as positive. With an average TNR of 0.997, the model correctly identifies approximately 99.70% of negative instances, on average. The average FNR value is 0.094, indicating that, on average, the model misclassifies around 0.94% of positive instances as negative.

References (Dataset)

1. Good and Bad Quality Fruits: https://www.kaggle.com/datasets/shashwatwork/fruitnet-indian-fruits-dataset-with-quality

2. Fresh and Rotten Fruits: https://zenodo.org/records/4788775

3. Fresh, Mild and Rotten Fruits: https://zenodo.org/records/4788775

4. Top Indian Fruits with their Qualities: https://ieee-dataport.org/open-access/fruitsgb-top-indian-fruits-quality#

THANK YOU

ANY QUESTION?