Marathwada Shikshan Prasarak Mandal's

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

**Project Report**

**on**

# Detection of Rotten Apples

Submitted By

**Sakshi S. Kherdekar (46014)**

**Sakshi R. Kakde (46024)**

**Rohit D. Paritkar (46066)**

**Dr. Babasaheb Ambedkar Technological University**

**Lonere (M.S.)**


Department of Computer Science and Engineering

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

(2022- 2023)

**Project Report**

**on**

# Detection of Rotten Apples

Submitted By

**Sakshi S. Kherdekar (46014)**

**Sakshi R. Kakde (46024)**

**Rohit D. Paritkar (46066)**

**In partial fulfillment of**

**Bachelor of Technology**

**(Computer Science & Engineering)**

Guided By

**Dr. Padmapani P. Tribhuvan**

Department of Computer Science & Engineering

**Deogiri Institute of Engineering and Management Studies, Aurangabad**

(2022- 2023)

# CERTIFICATE

This is to certify that, the Project entitled "**Detection of Rotten Apples**" submitted by **Sakshi Kherdekar, Sakshi Kakde, and Rohit Paritkar** is a bonafide work completed under my supervision and guidance in partial fulfillment for the award of Bachelor of Technology (Computer Science and Engineering) Degree of Dr. Babasaheb Ambedkar Technological University, Lonere.

Place: Aurangabad
Date:

**Dr. Padmapani P. Tribhuvan**
**Guide**

**Mr. Sanjay B. Kalyankar**
**Head**

**Dr. Ulhas D. Shiurkar**
**Director,**
**Deogiri Institute of Engineering and Management Studies,**
**Aurangabad**

# DECLARATION

This is to certify that, the partial project report entitled, "**Detection of Rotten Apples**" Submitted by **Sakshi Kherdekar, Sakshi Kakde, and Rohit Paritkar** is a bonafide work completed under my supervision and guidance in partial fulfillment for the award of a Bachelor of Technology degree in Computer Science and Engineering of Deogiri Institute of Engineering and Management Studies, Aurangabad under Dr. Babasaheb Ambedkar Technological University, Lonere.

Place: Aurangabad
Date:

Dr. Padmapani P. Tribhuvan

External Examiner

Guide

# Abstract

Detecting rotten apples becomes significant in the Fruit industry. Usually, the classification of fresh and rotten apples carried by humans is not effective. Human beings will become tired after doing the same task multiple times, but machines do not. Thus, the project proposes an approach to reduce human efforts, to reduce the cost and time for production by identifying the defects in the apple in the agricultural industry. If we do not detect those defects, those defective apples may contaminate good fruits. Hence, we proposed a model to avoid the spread of rottenness. The proposed model classifies the fresh apples and rotten apples from the input apple images. A Convolutional Neural Network (CNN) is used for extracting the features from input apple images and classifying the images into fresh and rotten apples. The performance of the proposed model is evaluated on a dataset that is downloaded from Kaggle and produces an accuracy of 98.66%. The results showed that the proposed CNN model can effectively classify fresh fruits and rotten fruits. In the proposed work, we inspected the transfer learning methods in the classification of fresh and rotten apples.

# Contents

## List of Abbreviation

| Sr. No | Acronym | Abbreviation |
|---|---|---|
| 1 | RFD | Rotten Fruit Detection |
| 2 | MDP | Markov Decision Process |
| 3 | CNN | Convolutional Neural Network |
| 4 | NLP | Natural Language Processing |
| 5 | AI | Artificial Intelligence |
| 6 | ML | Machine Learning |
| 7 | ANN | Artificial Neural Network |
| 8 | ROC | Receiver Operating Characteristics |
| 9 | TPR | True Positive Rate |
| 10 | FPR | False Positive Rate |
| 11 | RBM | Restricted Boltzmann Machine |

## List of Figures

## List of Graphs

## List of Tables

## List of Screenshots

# 1. INTRODUCTION

## 1.1. Introduction

The detection of defective apples and the classification of fresh and rotten apples represent the major challenges. Rotten apples may cause damage to the other fresh apples if not classified properly and can also affect productivity. Traditionally this classification is done by men, which was a labor-intensive, time taking, and not efficient procedure. Additionally, it increases the cost of production also. Hence, we need an automated system that can reduce the efforts of humans, and increase production, to reduce the cost of production and time of production. The recent approaches in computer vision, especially in the fields of machine learning and deep learning have improved the efficiency of image classification tasks. Detection of defective apples and the classification of fresh and rotten apples represent one the major challenges in agricultural fields. Rotten apples may cause damage to the other fresh fruits if not classified properly and can also affect productivity. Traditionally this classification is done by men, which was a labor-intensive, time taking, and not efficient procedure. Additionally, it increases the cost of production also. Hence, we need an automated system that can reduce the efforts of humans, and increase production, to reduce the cost of production and time of production.

Computer vision has numerous benefits in the fruit processing industry, enabling the automation of numerous activities. Classification and gradation of fruit and vegetable freshness are crucial for the industrial manufacturing of the highest-quality raw fruits sold in the market. The relevance of fruit safety to the agricultural sector of the global economy is significant. Recently, it has been observed that fruits are sensitive to several infections. This has resulted in global economic pressure on the agricultural industry. The time-consuming manual sorting of many types of fruits and vegetables to assess the quality of fresh and rotting fruits can be minimized by using automatic classification approaches. Therefore, automatic assessment of the quality of fruits and vegetables that enables faster processing of high-quality foods is a rapidly expanding topic of research. Studies have been conducted on using deep neural networks and convolutional neural networks (CNNs) to identify the freshness of fruits and vegetables. Instead of applying typical CNN architectures, this study explores the possibility of transfer learning regarding CNN models for the quality categorization of fruits and vegetables.

### 1.2. Necessity

The human eye can detect or analyze the rottenness of apples, but it is difficult. So, for the easiness of people, we want to develop a mobile application named “Detection of Rotten Apples” (DRA) by using CNN. The purpose of Detection of Rotten Apples is to detect and evaluate the accuracy of the rottenness of apples through images captured through cameras and classify them according to the accuracy of their rottenness.

There are a bulk of apples with the apple vendor so, it is difficult for them to differentiate between rotten apples and fresh apples as well as, for the consumers it is difficult to detect rotten and fresh apples therefore, the main purpose of this application is that it can be used by small scale industries, fruit vendors and the consumers. There is various web application for the detection of rotten apples but every time there is a need to go on the website and then click the image of an apple to make it easier, we have created a mobile application.

### 1.3. Objective

The problem caused by the uncertainty of the rottenness of apples can be overcome by this creative rottenness detector application without any chemical treatment which could damage the nutritional value. The main objective of the Detection of Rotten Apple’s project is that it can be used by anyone from customers to small-scale vendors. This is an easy-to-use mobile application that helps customers, small-scale industries, and fruit vendors to reduce their efforts of detecting rotten apples manually.

### 1.4. Theme of the Project

The objective of the project is to develop a Mobile app that will detect whether an apple is fresh or not. We have made a deep-learning model which will detect the prediction.

The user first needs to open the app and then take an image from the gallery or click the image on the spot. The results will be then fetched from the CNN model. And it will be printed on the Screen. So, this is all about the project.

# 2. LITERATURE SURVEY

## 2.1 Machine Learning and Deep Learning

### 2.1.1 Introduction to Machine Learning

Machine learning is a branch of artificial intelligence that focuses on the use of data and algorithms to imitate the way that humans learn. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, a recommender system, and many more.

With the help of data, machine learning algorithms build a mathematical model that helps in making predictions or decisions without being explicitly programmed. Machine learning constructs or uses algorithms that learn from historical data. The more we will provide the information, the higher will be the performance [8].

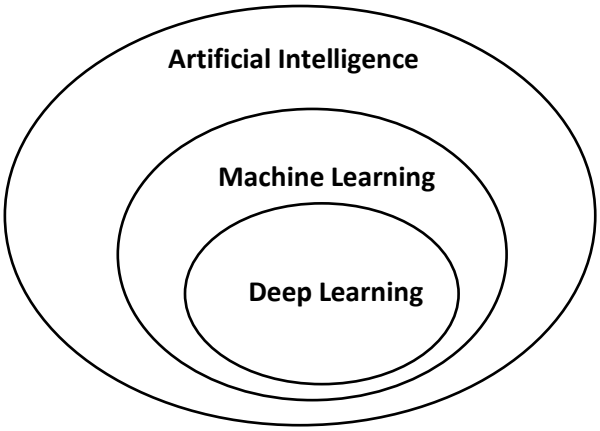



Fig. 2.1 AI, ML, and DL relation diagram

### 2.1.2 Types of Machine Learning

Machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning

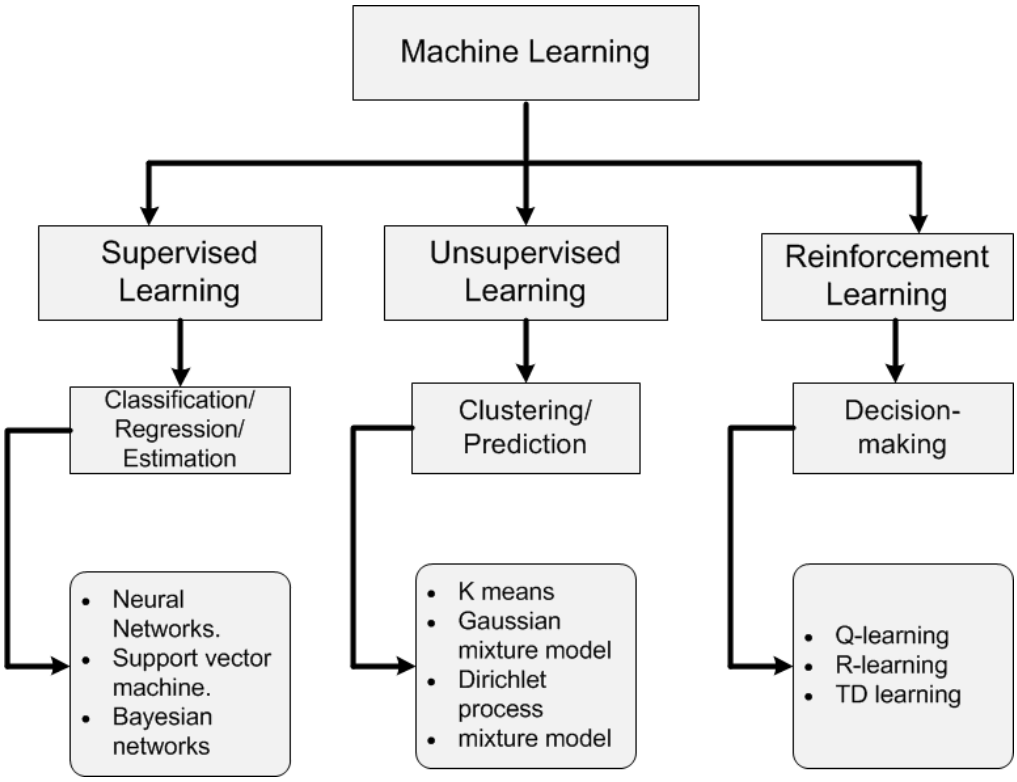



Fig 2.2 Types of Machine Learning

**Supervised Machine Learning**

Supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labeled" dataset, and based on the training, the machine predicts the output. Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, etc.

**Categories of Supervised Machine Learning**

Supervised machine learning can be classified into two types of problems Classification and Regression.

**Classification**

Classification algorithms are used to solve classification problems in which the output variable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc. Some popular classification algorithms are Random Forest Algorithm, Decision Tree Algorithm, Logistic Regression Algorithm, Support Vector Machine Algorithm, etc.

**Regression**

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather predictions, etc. Some popular Regression algorithms are Simple Linear Regression Algorithm, Multivariate Regression Algorithm, Decision Tree Algorithm, Lasso Regression, etc.

**Advantages of Supervised ML:**

1. Since supervised learning works with the labeled dataset so we can have an exact idea about the classes of objects.
2. These algorithms are helpful in predicting the output based on prior experience.

**Disadvantages of Supervised ML:**

1. These algorithms are not able to solve complex tasks.
2. It may predict the wrong output if the test data is different from the training data.
3. It requires lots of computational time to train the algorithm.

**Applications of Supervised Learning:**

Some common applications of Supervised Learning are given below:

1. **Image Segmentation:** Supervised Learning algorithms are used in image segmentation. In this process, image classification is performed on different image data with pre-defined labels.
2. **Medical Diagnosis:** Supervised algorithms are also used in the medical field for diagnosis purposes. It is done by using medical images and past-labeled data with labels for disease conditions. With such a process, the machine can identify a disease for the new patients.
3. **Fraud Detection -** Supervised Learning classification algorithms are used for identifying fraud transactions, fraud customers, etc. It is done by using historic data to identify the patterns that can lead to possible fraud.
4. **Spam detection -** In spam detection & filtering, classification algorithms are used. These algorithms classify an email as spam or not spam. The spam emails are sent to the spam folder.
5. **Speech Recognition -** Supervised learning algorithms are also used in speech recognition. The algorithm is trained with voice data, and various identifications can be done using the same, such as voice-activated passwords, voice commands, etc.

**Unsupervised Machine Learning:**

In unsupervised machine learning, the models are trained with data that is neither classified nor labeled, and the model acts on that data without any supervision. It means, the machine is trained using the unlabeled dataset, and it predicts the output without any supervision.

The main aim of the unsupervised learning algorithm is to group or categorize the unsorted dataset according to the similarities, patterns, and differences**.** Machines are instructed to find the hidden patterns from the input dataset.

**Categories of Unsupervised Machine Learning**

Unsupervised Learning can be further classified into two types, which are Clustering and Association.

**Clustering**

Clustering is a technique used by computer scientists to find the inherent groups from the data. It is a way to group objects into a cluster such that objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. An example of the clustering algorithm is grouping the customers by their purchasing behavior. Some of the popular clustering algorithms are the K-Means Clustering algorithm, Mean-shift algorithm, DBSCAN Algorithm, Principal Component Analysis, Independent Component Analysis.

**Association**

Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit. This algorithm is mainly applied in Market Basket analysis, Web usage mining, continuous production, etc. Some popular algorithms of Association rule learning are Apriori Algorithm, Eclat, FP-growth algorithm.

**Advantages of Unsupervised ML:**

1. These algorithms can be used for complicated tasks compared to supervised ones because these algorithms work on the unlabeled dataset.
2. Unsupervised algorithms are preferable for various tasks as getting the unlabeled dataset is easier as compared to the labeled dataset.

**Disadvantages of Unsupervised ML:**

1. The output of an unsupervised algorithm can be less accurate as the dataset is not labeled, and algorithms are not trained with the exact output in prior.
2. Working with Unsupervised learning is more difficult as it works with the unlabeled dataset that does not map with the output.

**Applications of Unsupervised Learning**

1. **Network Analysis:** Unsupervised learning is used for identifying plagiarism and copyright in document network analysis of text data for scholarly articles.
2. **Recommendation Systems:** Recommendation systems widely use unsupervised learning techniques for building recommendation applications for different web applications and e-commerce websites.
3. **Anomaly Detection:** Anomaly detection is a popular application of unsupervised learning, which can identify unusual data points within the dataset. It is used to discover fraudulent transactions.
4. **Singular Value Decomposition:** Singular Value Decomposition or SVD is used to extract information from the database. For example, extracting information of each user located at a particular location.

**Semi-Supervised Learning:**

Semi-supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It uses a combination of labeled and unlabeled datasets during the training period. As labels are costly, but for corporate purposes, they may have few labels. The main aim of semi-supervised learning is to effectively use all the available data, rather than only labeled data like in supervised learning. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labeled data.

**Advantages:**

1. It is simple and easy to understand the algorithm.
2. It is highly efficient.
3. It is used to solve drawbacks of Supervised and Unsupervised Learning algorithms.

**Disadvantages:**

1. Iterations results may not be stable.
2. We cannot apply these algorithms to network-level data.

**Reinforcement Learning:**

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, acting, learning from experiences, and improving its performance. Agent gets rewarded for each good action and gets punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

In reinforcement learning, there is no labeled data like supervised learning, and agents learn from their experiences only. The reinforcement learning process is like a human being learning various things through experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, and the goal of an agent is to get a high score. Due to its way of working, reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, and multi-agent systems.

A reinforcement learning problem can be formalized using Markov Decision Process (MDP). In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state.

**Categories of Reinforcement Learning**

Reinforcement learning is categorized mainly into two types of methods/algorithms:

1. **Positive Reinforcement Learning:** Positive reinforcement learning specifies increasing the tendency that the required behavior would occur again by adding something. It enhances the strength of the behavior of the agent and positively impacts it.
2. **Negative Reinforcement Learning:** Negative reinforcement learning works exactly opposite to positive RL. It increases the tendency that the specific behavior would occur again by avoiding the negative condition.

**Real-world Use cases of Reinforcement Learning**

1. **Video Games:**

   RL algorithms are much popular in gaming applications. It is used to gain super-human

performance. Some popular games that use RL algorithms are AlphaGO and AlphaGO Zero.

2. **Resource Management:**

   The "Resource Management with Deep Reinforcement Learning" paper showed how to use RL in a computer to automatically learn and schedule resources to wait for different jobs in order to minimize average job slowdown.

3. **Robotics:**

   RL is widely used in Robotics applications. Robots are used in the industrial and manufacturing area, and these robots are made more powerful with reinforcement learning. There are different industries that have their vision of building intelligent robots using AI and Machine learning technology.

4. **Text Mining:**

   Text-mining, one of the great applications of NLP, is now being implemented with the help of Reinforcement Learning by Salesforce company.

**Advantages of Reinforcement ML:**

1. It helps in solving complex real-world problems which are difficult to be solved by general techniques.
2. The learning model of RL is similar to the learning of human beings; hence most accurate results can be found.
3. Helps in achieving long-term results.

**Disadvantage of Reinforcement ML:**

1. RL algorithms are not preferred for simple problems.
2. RL algorithms require huge data and computations.—
3. Too much reinforcement learning can lead to an overload of states which can weaken the results.

### 2.1.3 Introduction to Deep Learning

Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence. Basically, it is a machine learning class that makes use of numerous nonlinear processing units to perform feature extraction as well as transformation. The output from each preceding layer is taken as input by each one of the successive layers.

Deep learning is implemented with the help of Neural Networks, and the inspiration behind artificial neural networks is the biological neurons, which are nothing but brain cells. Since deep learning has been evolved by machine learning to mimic human behavior, so same is "the idea of deep learning to build a such algorithm that can mimic the brain".

So basically, deep learning is implemented with the help of deep networks, which are nothing but neural networks with multiple hidden layers.

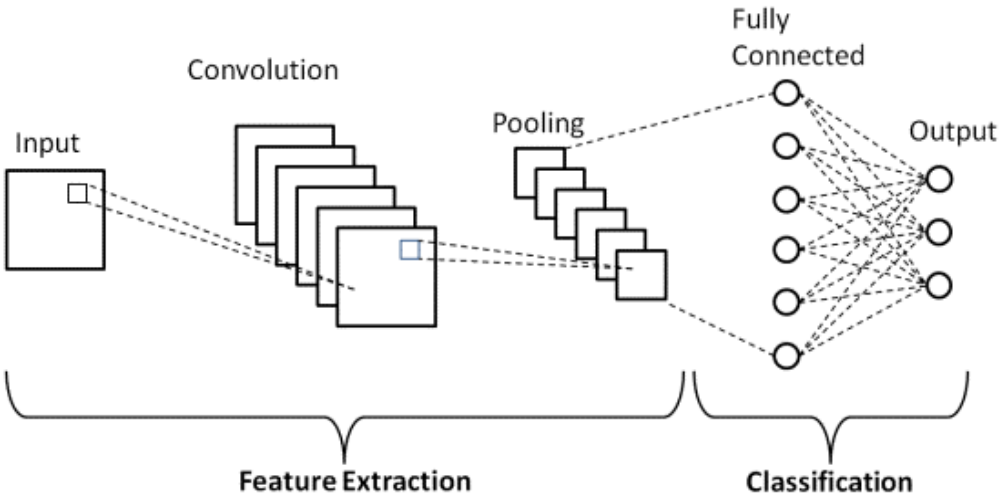



Fig 2.3 CNN layer's structure

Images are sent to layers that determine the patterns of local contrast, which means it will differentiate on the basis of colors, luminosity, etc. The 1st hidden layer will determine the face feature, i.e., it will fixate on the eyes, nose, lips, etc. and then, it will apply those features on the correct face template. More hidden layers can be added to solve more complex problems such as whether a particular kind of face has large or light complexions.

**Architectures**

1. **Deep Neural Networks**: Deep neural networks are highly proficient in model and process non-linear associations.
2. **Deep Belief Networks**: A deep belief network is a class of Deep Neural Networks that comprises multi-layer belief networks

**Steps to perform DBN:**

I. With the help of the Contrastive Divergence algorithm, a layer of features is learned from perceptible units.
II. Next, the formerly trained features are treated as visible units, which perform the learning of features.
III. Lastly, when the learning of the final hidden layer is accomplished, then the whole DBN is trained.

3. **Recurrent Neural Networks**: It permits parallel as well as sequential computation, and it is exactly like that of the human brain (large feedback network of connected neurons). Since they are capable enough to reminisce all the imperative things related to the input they have received, so they are more precise.

### 2.1.4 Types of Deep Learning

Based on the methods and way of learning, deep learning is divided into mainly five types, which are:

1. Feed Forward Neural Network
2. Recurrent Neural Network
3. Convolutional Neural Network
4. Restricted Boltzmann Machine
5. Autoencoders

**Feed Forward Neural Network**

A feed-forward neural network is an Artificial Neural Network, which ensures that the nodes do not form a cycle. In this kind of neural network, all the perceptrons are organized within layers, such that the input layer takes the input and the output layer generates the output. Since the hidden layers do not link with the outside world, it is named hidden layers. Applications of Feed Forward Neural Networks are Data Compression, Pattern Recognition, Computer Vision, Sonar Target Recognition, Speech Recognition, Handwritten Characters Recognition, etc.

**Recurrent Neural Network**

Recurrent neural networks are yet another variation of feed-forward networks. Here each of the neurons present in the hidden layers receives an input with a specific delay in time. The Recurrent neural network mainly accesses the preceding info of existing iterations. It not only processes the inputs but also shares the length as well as weights crossways time. Applications are Machine Translation, Robot Control, Time Series Prediction, Speech Recognition, Speech Synthesis, Speech Synthesis, Rhythm Learning, Music Composition, etc.

**Convolutional Neural Network**

Convolutional Neural Networks are a special kind of neural network mainly used f image classification, clustering of images and object recognition. DNNs enable the unsupervised construction of hierarchical image representations. To achieve the best accuracy, deep convolutional neural networks are preferred more than any other neural network. Applications of CNN are Identify Faces, Street Signs, and Tumours, Image Recognition, Video Analysis, NLP, Anomaly Detection, Drug Discovery, Checkers Game, and Time Series Forecasting.

**Restricted Boltzmann Machine**

Boltzmann Machines (BMs) are models in which the neurons present in the input layer and the hidden layer encompass symmetric connections amid each other, but there is no internal association between them. These restrictions in BMs help the model to train efficiently. In contrast, Boltzmann machines encompass internal connections inside the respective layer. Applications are Filtering, Feature Learning, Classification, Risk Detection, Business, and Economic analysis, etc.

**Autoencoders**

An autoencoder neural network is another kind of unsupervised machine learning algorithm. Autoencoders are mainly used for the smaller representation of the input. It helps in the reconstruction of the original data from compressed data. This algorithm is comparatively simple as it only necessitates the output identical to the inputs.

Encoder: Convert input data to lower dimensions.

Decoder: Reconstruct the compressed data.

Applications of Autoencoders are Classification, Clustering, and Feature Compression.

**Deep learning applications**

1. Self-Driving Cars: In self-driven cars, it is able to capture the images around it by processing a huge amount of data, and then it will decide which actions should be incorporated to take a left or right or should it stop. So, accordingly, it will decide what actions it should take, which will further reduce the accidents that happen every year.
2. Voice-Controlled Assistance: When we talk about voice control assistance, then Siri is the one thing that comes into our mind. So, you can tell Siri whatever you want it to do it for you, and it will search it for you and display it for you.
3. Automatic Image Caption Generation: Whatever image you upload, the algorithm will work in such a way that it will generate captions accordingly. If you say blue-colored eye, it will display a blue-colored eye with a caption at the bottom of the image.
4. Automatic Machine Translation: With the help of automatic machine translation, we can convert one language into another with the help of deep learning.

**Advantages**

1. Lessens the need for feature engineering.
2. It eradicates all those costs that are needless.
3. It easily identifies difficult defects.
4. It results in the best-in-class performance on problems.

**Disadvantages**

1. It requires an ample amount of data.
2. It is quite expensive to train.
3. It does not have strong theoretical groundwork.
4. It only learns through observations.
5. It comprises biases issues.

## 2.2 Classification

Classification is the process of identifying and grouping objects or ideas into predetermined categories. In data management, classification enables the separation and sorting of data according to set requirements for various business or personal objectives.

### 2.2.1 Classification in Machine Learning

The classification algorithm is a Supervised Learning technique that is used to identify the category of new observations based on training data. In Classification, a program learns from the given dataset or observations and then classifies new observations into several classes or groups. Classes can be called targets/labels or categories. Unlike regression, the output variable of Classification is a category, not a value.

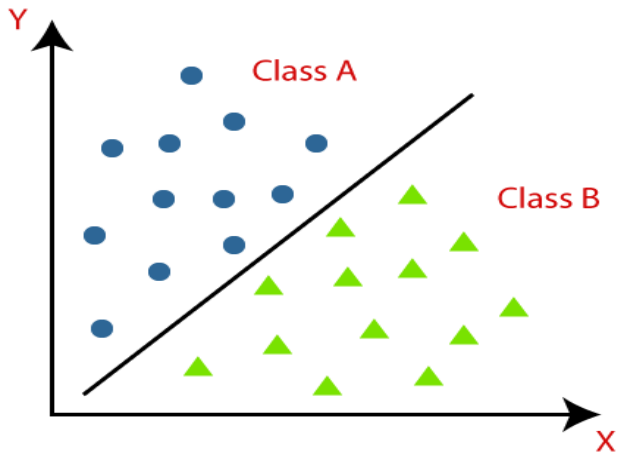
In the classification algorithm, a discrete output function(y) is mapped to the input variable(x).

$$y=f(x), \text{ where } y = \text{categorical output}$$

Classification algorithms are mainly used to predict the output for categorical data. In the below diagram, there are two classes, class A and Class B. These classes have features that are like each other and dissimilar to other classes. The best example of an ML classification algorithm is Email Spam Detector.

The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

1. Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary Classifier. Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT, or DOG, etc.
2. Multi-class Classifier: If a classification problem has more than two outcomes, then it is called a Multi-class Classifier. Example: Classifications of types of crops, Classification of types of music.




Graph. 2.1 Classification

**Learners in Classification Problems:**

In the classification problems, there are two types of learners:

1. Lazy Learners: Lazy Learner first stores the training dataset and waits until it receives the test dataset. In the Lazy learner case, classification is done on the basis of the most related data stored in the training dataset. It takes less time in training but more time for predictions. Example: K-NN algorithm, Case-based reasoning
2. Eager Learners: Eager Learners develop a classification model based on a training dataset before receiving a test dataset. Opposite to Lazy learners, an Eager Learner takes more time in learning, and less time in prediction. Example: Decision Trees, Naïve Bayes, ANN.

**Types of ML Classification Algorithms:**

Classification Algorithms can be further divided into the Mainly two category:

**Linear Models**

1. Logistic Regression
2. Support Vector Machines

**Non-linear Models**

1. K-Nearest Neighbors
2. Kernel SVM
3. Naïve Bayes
4. Decision Tree Classification
5. Random Forest Classification

**Evaluating a Classification model:**

Once our model is completed, it is necessary to evaluate its performance; either it is a Classification or Regression model. So, for evaluating a Classification model, we have the following ways:

**Log Loss or Cross-Entropy Loss:**

1. It is used for evaluating the performance of a classifier, whose output is a probability value between 0 and 1.
2. For a good binary Classification model, the value of log loss should be near to 0.
3. The value of log loss increases if the predicted value deviates from the actual value.
4. The lower log loss represents the higher accuracy of the model.

**Confusion Matrix:**

1. The confusion matrix provides us with a matrix/table as output and describes the performance of the model.

2. It is also known as the error matrix.

The matrix consists of predictions that result in a summarized form, which has a total number of correct predictions and incorrect predictions. The matrix looks like anthes below table:

| | **Actual Positive** | **Actual Negative** |
|---|---|---|
| Predicted Positive | True Positive | False Positive |
| Predicted Negative | False Negative | True Negative |

Table 2.1 Prediction in CNN

Accuracy = (TP + TN) / Total Population

**AUC-ROC curve:**

ROC curve stands for Receiver Operating Characteristics Curve and AUC stands for Area Under the Curve.

1. It is a graph that shows the performance of the classification model at different thresholds.
2. To visualize the performance of the multi-class classification model, we use the AUC-ROC Curve.
3. The ROC curve is plotted with TPR and FPR, where TPR (True Positive Rate) on Y-axis and FPR (False Positive Rate) is on X-axis.

**Use cases of Classification Algorithms**

Classification algorithms can be used in different places. Below are some popular use cases of Classification Algorithms Email Spam Detection, Speech Recognition, Identification of Cancer tumor cells, Drugs Classification, Biometric Identification, etc.

## 2.3 Convolutional Neural Network

### 2.3.1 Introduction

Convolutional neural networks are used to do image classification and image recognition in neural networks. Scene labeling, objects detections, face recognition, etc. are some of the areas where convolutional Neural Networks are widely used.

A computer sees an image as an array of pixels and depends on the resolution of the image. An RGB image is a 6 * 6 * 3 array of the matrix, and a grayscale image is 4 * 4 * 1 array. Based on image resolution, it will see as h * w * d, where h= height w= width, and d= dimension.

In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, and filters. After that, we apply the Soft-max function to classify an object with probabilistic values 0 and 1.

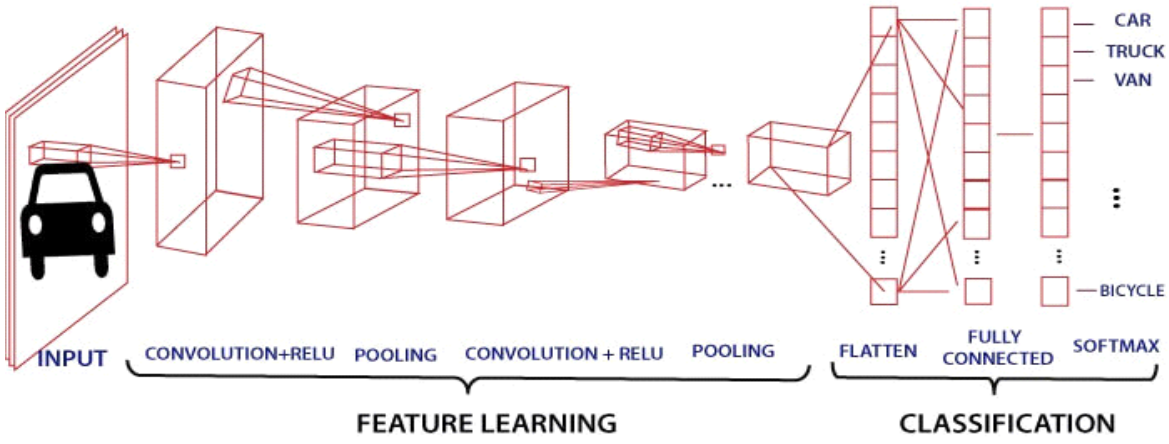



Fig 2.4 Feature Learning and Classification in CNN

### 2.3.2 Working

**Convolution Layer**

Convolution is a mathematical operation that takes two inputs such as an image matrix and a kernel or filter. The convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. The dimension of the image matrix is **h×w×d**. The dimension of the filter is $\mathbf{f_h \times f_w \times d}$ and the dimension of the output is $\mathbf{(h-f_h+1) \times (w-f_w+1) \times 1}$.

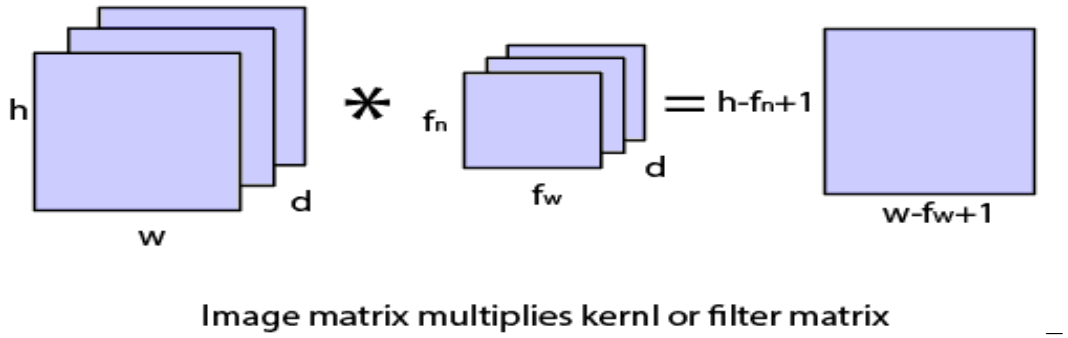



Fig. 2.5 Image matrix multiplies Kernel or filter matrix

Let's start with consideration a 5*5 image whose pixel values are 0, 1, and filter matrix 3*3 as:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$5 \times 5$ – Image Matrix  $3 \times 3$ – Filter Matrix

Fig. 2.6 Matrix Multiplication

The convolution of 5*5 image matrix multiplies with 3*3 filter matrix is called "Features Map" and show as an output.

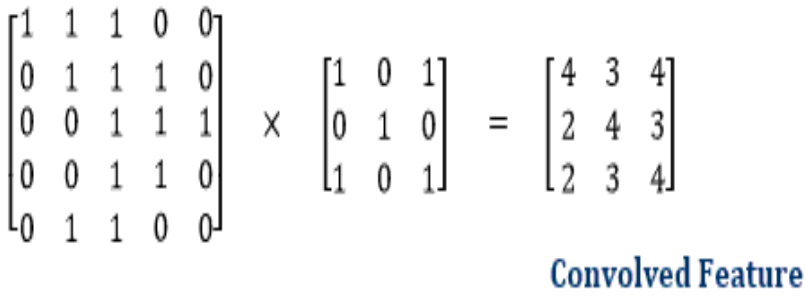



Fig. 2.7 Matrix Multiplication Convolved Feature

Convolution of an image with different filters can perform an operation such as blur, sharpen, and edge detection by applying filters.

**Strides**

Stride is the number of pixels that are shifted over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly if it is equaling to 2 or more, we move them twice as fast. The following figure shows that the convolution would work with stride no 2

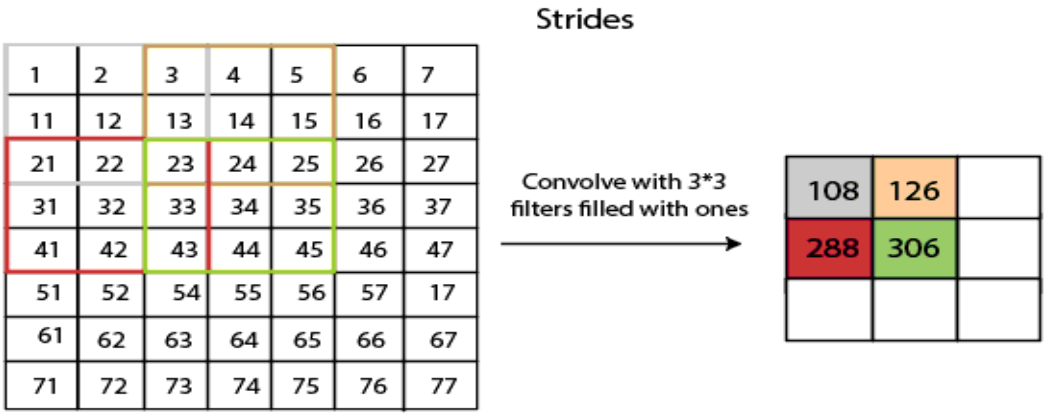



Fig. 2.8 Convolution work with a stride of 2

**Padding**

Padding plays a crucial role in building the convolutional neural network. If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end. "Padding is an additional layer which can add to the border of an image."

**Pooling Layer**

Pooling is the "downscaling" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called down sampling or subsampling. It reduces the dimensionality of each map but retains the important information.

**Max Pooling**

Its main objective is to downscale an input representation, reducing its dimensionality. It allows for the assumption to be made about features contained in the sub-region binned. Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

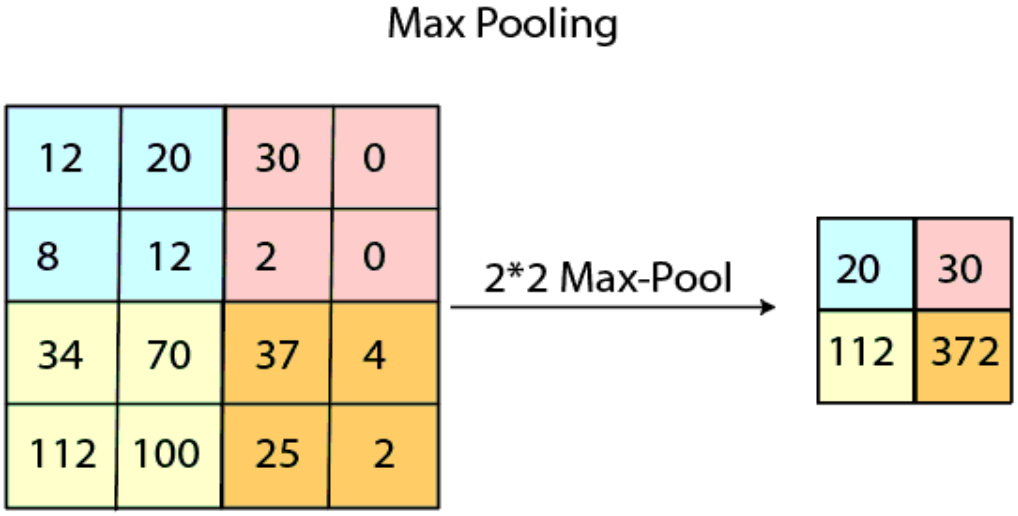



Fig 2.9 Max Pooling Calculations in CNN

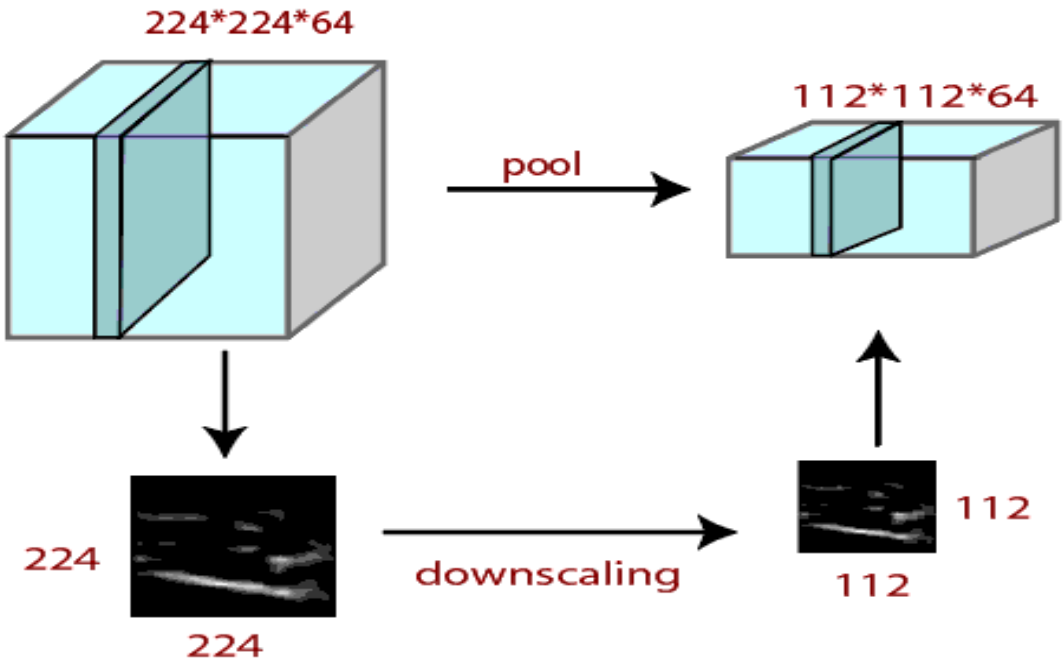



Fig 2.10 Average Pooling Calculations in CNN

**Average Pooling**

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax:

layer = averagePooling2dLayer(pool Size)

layer = averagePooling2dLayer(pool Size, Name, Value)

**Sum Pooling**

The sub-region for sum pooling or mean pooling is set exactly the same as for max pooling but instead of using the max function we use sum or mean.

**Fully Connected Layer**

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

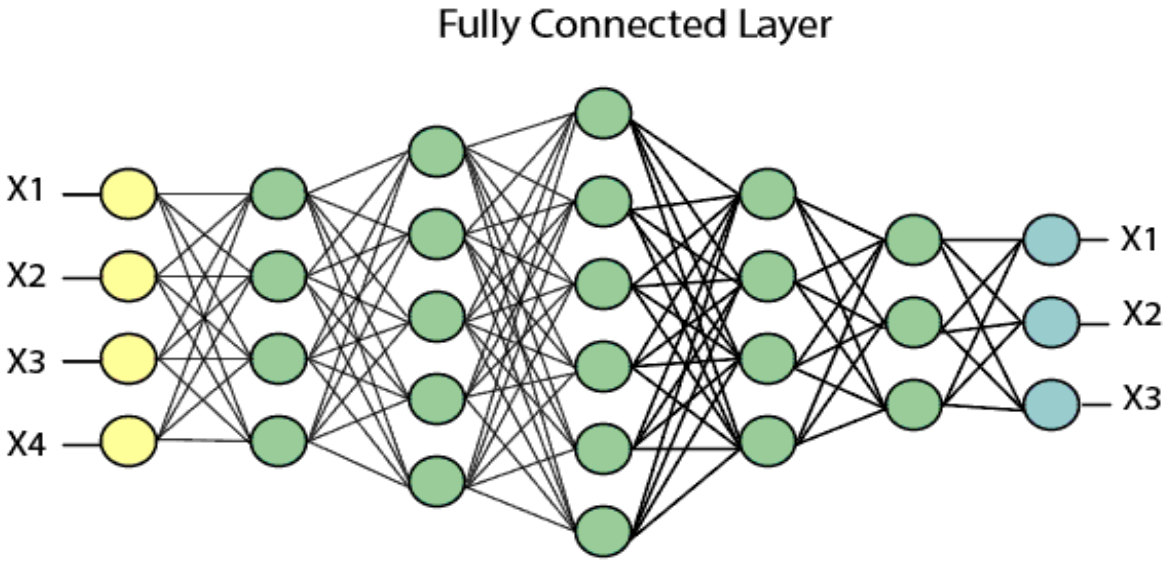



Fig. 2.11 Fully Connected Layer in CNN

In the above diagram, the feature map matrix will be converted into vectors such as $X_1$, $X_2$, $X_3$... $X_n$ with the help of fully connected layers. We will combine features to create a model and apply the activation function such as SoftMax or sigmoid to classify the outputs as a car, dog, truck, etc.

# 3. SYSTEM DEVELOPMENT

## 3.1 Requirement Specification

### 3.1.1 User Interface

1. The user interface is kept simple so that the user can easily use the complete functionality of Android, it makes it easy for the user to use this app as it is user-friendly, and the users do not have to reveal their identity.
2. The user must open the app and give access to the gallery and the camera so that the user can select the images from the gallery and click photos of the apples from the camera.
3. After the user is done with giving the access then the user can select the option getting started.
4. Then the user has two choices either to select an image from the gallery or to click the photo after this the user can either predict that the apple is rotten or not or can reset and click the image again.

### 3.1.2 Hardware Requirements

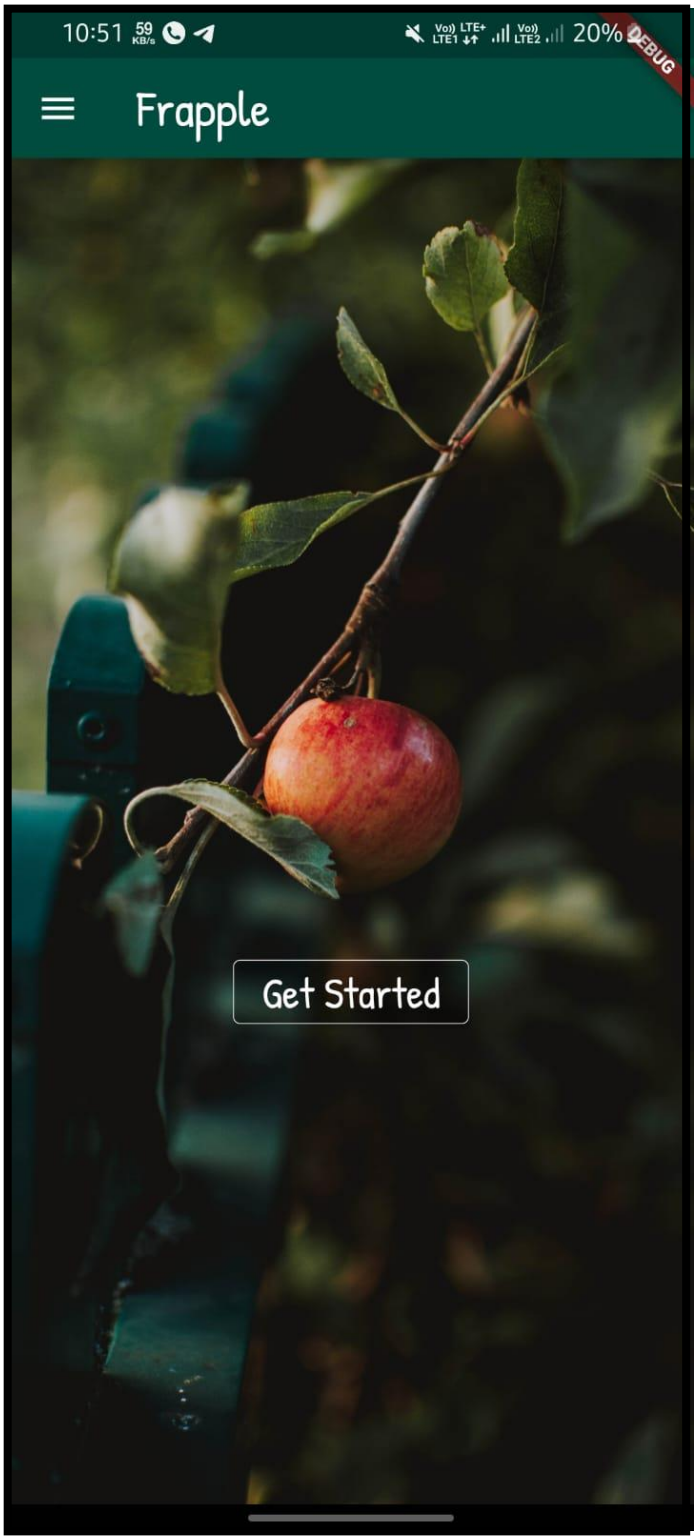
1. Intel Pentium 4 (2.8 GHz) Processor and Above
2. RAM 1 GB and Above
3. System Type 32-bit and above
4. HDD 40 GB Hard Disk Space and Above

### 3.1.3 Software Requirements

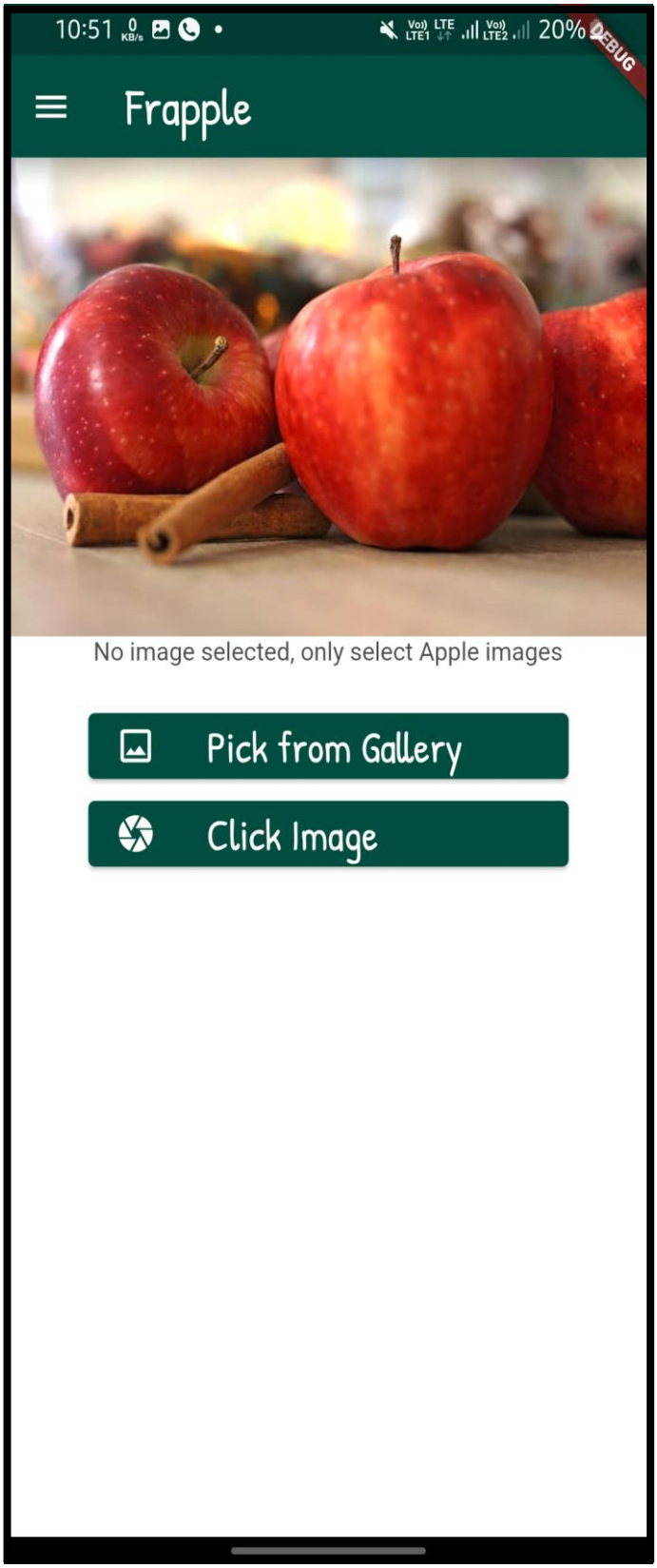
1. Operating System: The operating system used in developing mobile applications of the Detection of Rotten Apples is Windows 11. The mobile application is developed in Flutter. So, for this project, we have used Windows 11.

2. Dataset Storage: For dataset storage of this mobile application Google Drive is used. The machine learning model for the Detection of Rotten Apples is executed on the IDE named Google Colab through which we can directly get connected to Google Drive to access the dataset.
3. Android Studio: It is a platform used for the development of mobile applications.
4. Flutter: Flutter is a framework used for the development of mobile applications.
5. Google Colab: It is used for training and testing the machine learning model using CNN.

### 3.2 User Interface




Screenshot 3.1 Front Screen of Frapple App




Screenshot 3.2 Second Screen of Frapple App


Screenshot 3.3 Detecting Fresh Apple


Screenshot 3.4 Detecting Rotten Apple

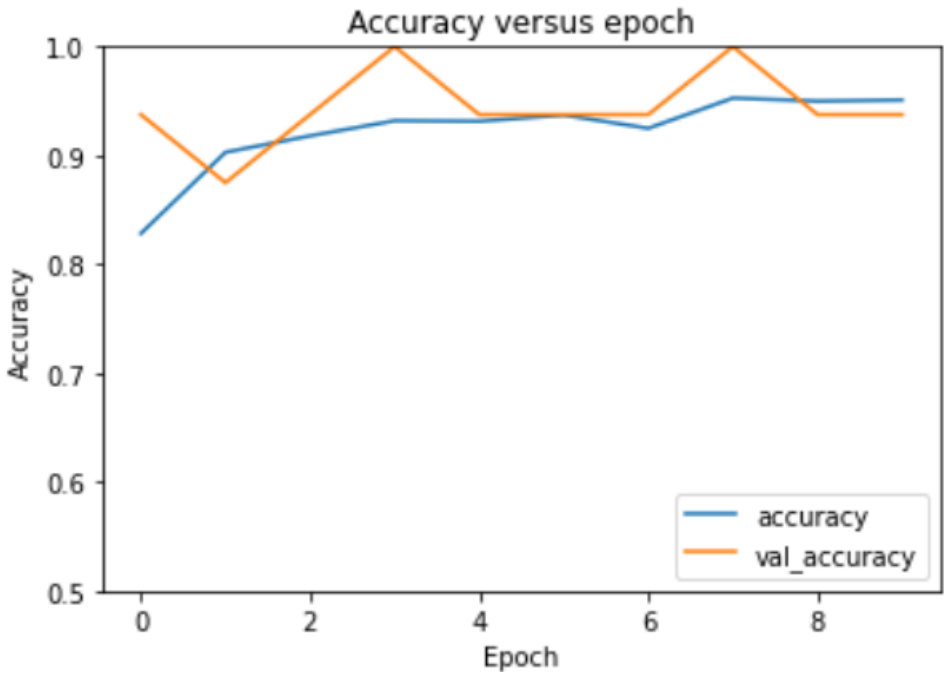
### 3.3 Dataset Details

| Training | Testing |
|---|---|
| Fresh apples – 30 | Fresh Apples - 20 |
| Rotten apples – 120 | Rotten Apples - 100 |

Table 3.1 Dataset detail

# 4. PERFORMANCE EVALUATION

We have calculated the performance of our CNN model by plotting the graph of Accuracy against epoch. Our CNN model is giving accuracy in the range of 90 to 96%. The below graph shows for each epoch what accuracy we have got. The accuracy is in the range of 84 to 96 out of 100.




Graph.4.2 Plot of Accuracy vs Epoch

For calculating the performance of our model on the Android App we have taken 10 images of Apples. We have checked the prediction of our CNN model on the app and with the help of manual testing, we have measured the performance of our Model.

Out of 10 images output for one image is not predicted correctly, but the rest of the 9 Images are giving correct output.

The below table shows the expected output for the apple image and the output given by the CNN Model:

| Sr. No. | Apple Photo | Expected Output | Output Given By CNN Model |
|---|---|---|---|
| 1 |  | Fresh Apple | Fresh Apple |
| 2 |  | Fresh Apple | Fresh Apple |
| 3 |  | Rotten Apple | Rotten Apple |
| 4 |  | Fresh Apple | Fresh Apple |
| 5 |  | Rotten Apple | Rotten Apple |
| 6 |  | Rotten Apple | Fresh Apple |
| 7 |  | Fresh Apple | Fresh Apple |
| 8 |  | Fresh Apple | Fresh Apple |
| 9 |  | Fresh Apple | Fresh Apple |
| 10 |  | Fresh Apple | Rotten Apple |

Table 4.3 Performance of the model

**Performance Analysis**

**Pre-condition:**

1. Tester should know how to handle mobile phones and their functioning of it.
2. Mobile phones should be in working condition. i.e. fully charged and a SIM card should be inserted in it.

**Test Cases:**

| **Step no.** | **Test cases** | **Expected result** | **Actual Result** | **Status (pass/fail)** |
|---|---|---|---|---|
| 1 | Check Weather app is open or not. | Yes | Yes | Pass |
| 2 | Check whether the start button is navigating to the next page or not. | Yes | Yes | Pass |
| 3 | Check whether the app is taking images correctly or not. | Yes | Yes | Pass |
| 4 | Check whether the app is loading the results after clicking the image or not. | Yes | No | Fail |
| 5 | Check whether the app is giving correct results or not. | Yes | Yes | Pass |
| 6 | Check whether the app is able to click images from the Camera. | Yes | Yes | Pass |
| 7 | Check whether the app is able to select images from the gallery or not. | Yes | Yes | Pass |
| 8 | Check whether the buttons are working properly or not. | Yes | Yes | Pass |

Table 4.4 Test cases

# 5. CONCLUSION AND FUTURE SCOPE

The classification of fresh and rotten apples is very important in agricultural fields. In our work, we introduced a model based on CNN and concentrated on building transfer learning models for the task of classification of fresh and rotten apples. The effects of different hyper-parameters i.e., batch size, number of epochs, optimizer, and learning rate are interrogated in this work. The results proved that the CNN model proposed can classify fresh and rotten apples firmly and produced better accuracy than transfer learning models. Thus, the proposed CNN model can automate the process of the human brain in classifying fresh and rotten fruits with the help of the proposed convolutional neural network model and thus reduce human errors while classifying fresh and rotten fruits. An accuracy of 97.82% is attained` for the proposed CNN model.

This project can be easily changed or upgraded in the future as the versions used are mostly compatible with all devices. By checking the rottenness of apples, we can also predict the same for different fruits by training the model with different images.

# References

[1] Zhang, Baohua, Shuxiang Fan, Jiangbo Li, Wenqian Huang, Chunjiang Zhao, Man Qian, and Ling Zheng. "Detection of early rottenness on apples by using hyperspectral imaging combined with spectral analysis and image processing." *Food Analytical Methods* 8, no. 8 (2015): 2075-2086.

[2] Palakodati, Sai Sudha Sonali, Venkata Rami Reddy Chirra, Dasari Yakobu, and Suneetha Bulla. "Fresh and Rotten Fruits Classification Using CNN and Transfer Learning." *Rev. d'Intelligence Artif.* 34, no. 5 (2020): 617-622.

[3] Anwar, Iqra. "How to Detect Rotten Fruits using Image Processing in Python?."

[4] Singh, Shiksha, and Nagendra Pratap Singh. "Machine learning-based classification of good and rotten apple." In *Recent trends in communication, computing, and electronics*, pp. 377-386. Springer, Singapore, 2019.

[5] El-aziz, Abd, A. Atrab, Ashraf Darwish, Diego Oliva, and Aboul Ella Hassanien. "Machine learning for apple fruit diseases classification system." In *The International Conference on*

[6] Payne, Rap. "Developing in Flutter." In *Beginning App Development with Flutter*, pp. 9-27. Apress, Berkeley, CA, 2019.

[7] https://www.javatpoint.com/flutter-buttons (20.11.22, 1:30 PM)

[8] https://www.youtube.com/watch?v=2n0ja7ALyDY (15.12.22, 9:00 PM)

[9] https://www.javatpoint.com/machine-learning (24.12.22 ,10:30 AM)

# APPENDIX

## CNN Model

```
from google.colab import drive

drive.mount('/content/drive')
```

**Importing Libraries**

```
import tensorflow as tf

from keras.preprocessing.image import ImageDataGenerator

import pandas as pd

import keras

import matplotlib.pyplot as plt

import matplotlib.image as mpimg


train_datagen = ImageDataGenerator(
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)
training_set = train_datagen.flow_from_directory(
        directory = r"/content/drive/MyDrive/DatasetFYP/Train/",
        target_size=(64 , 64),
        batch_size=32,
```

```
        class_mode='binary')


test_datagen = ImageDataGenerator(rescale=1./255)
test_set = test_datagen.flow_from_directory(
        directory = r"/content/drive/MyDrive/DatasetFYP/Test/",
        target_size=(64 , 64),
        batch_size=32,
        class_mode='binary')
validation = ImageDataGenerator(rescale=1./255)
vall = test_datagen.flow_from_directory(
        directory = r"/content/drive/MyDrive/DatasetFYP/Val/",
        target_size=(64 , 64),
        batch_size=32,
        class_mode='binary')
```

**Intiallize CNN**

```
cnn = tf.keras.models.Sequential()
```

**Step1:- Convolution**

```
cnn.add(tf.keras.layers.Conv2D(filters = 32 , kernel_size = 3 , activation = 'relu' ))
cnn.add(keras.Input(shape=(64, 64, 3)))
```

**Pooling**

```
cnn.add(tf.keras.layers.MaxPool2D( pool_size = 2 , strides = 2))
```

**Flattening**

```
cnn.add(tf.keras.layers.Flatten())
```

**Fully connection**

```
cnn.add(tf.keras.layers.Dense(units = 128 , activation = 'relu'))
```

**Adding the output layer**

```
cnn.add(tf.keras.layers.Dense(units = 1 , activation = 'sigmoid'))
```

**Compile the model**

```
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

**Fitting the model**

```
history=cnn.fit(x = training_set , validation_data = vall , epochs = 10)
```

**Evaluation**

```
test_loss, test_acc = cnn.evaluate(test_set, verbose=2)

print("The accuracy of the model is:")

print(test_acc)
```

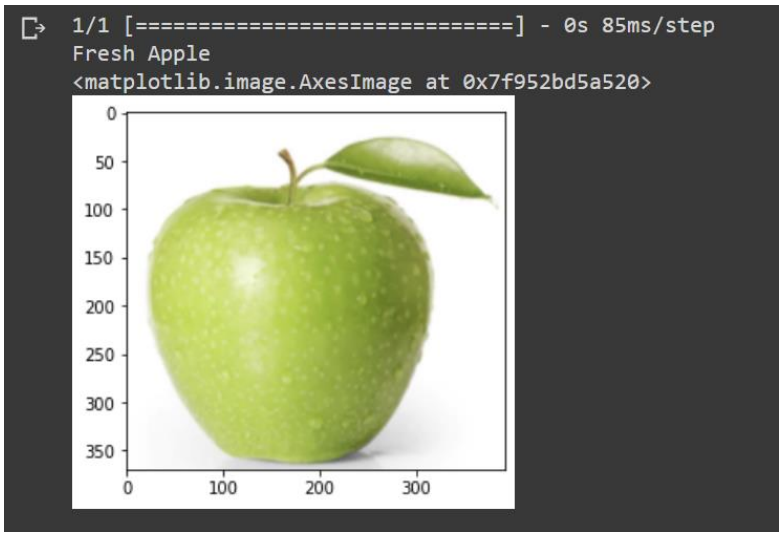
**Plot the accuracy**

```
plt.title("Accuracy versus epoch")

plt.plot(history.history['accuracy'], label='accuracy')

plt.plot(history.history['val_accuracy'], label = 'val_accuracy')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.ylim([0.5, 1])

plt.legend(loc='lower right')
```

**Predicting**

```
import numpy as np
```

```
from keras_preprocessing import image

test_image =
tf.keras.utils.load_img('/content/drive/MyDrive/DatasetFYP/Test/freshapples/Screen Shot 2018-06-08 at 5.07.26 PM.png', target_size = (64, 64))

test_image = image.img_to_array(test_image)

test_image = np.expand_dims(test_image, axis = 0)

result = cnn.predict(test_image)

training_set.class_indices

if result[0][0] == 0:

  prediction = 'Fresh Apple'

else:

  prediction = 'Rotten Apple

print(prediction)

plt.imshow(mpimg.imread('/content/drive/MyDrive/DatasetFYP/Test/freshapples/Screen Shot 2018-06-08 at 5.07.26 PM.png'))
```



**Flutter Code For App Development**

**Main.dart File in Flutter**

```
import 'package:flutter/material.dart';
import 'package:tflite_image_classification/TfliteModel.dart';
import 'TfliteModel.dart';


void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Frapple',


      theme: ThemeData(
        primarySwatch: Colors.teal,),
      home: Homepage(),);}
}
class Homepage extends StatefulWidget{
  @override
  _HomepageState createState() => _HomepageState();
}
```

```
class _HomepageState extends State<Homepage>{
  @override
  Widget build(BuildContext context){
    return Scaffold(
      appBar: AppBar(
        leading: GestureDetector(
          onTap: () {},
          child: Icon(
            Icons.menu,),),
        title: const Text("Frapple",
          style: TextStyle(
            fontFamily: 'PatrickHand-Regular',
            fontSize: 28,),),
        elevation: 10,backgroundColor: Colors.teal[900],
      ),
      body: Center(
        child: Stack(
          children: <Widget>[
            Container(
              alignment: Alignment.center,
              child:
              Image.asset('assets/images/apple_backgroud(1).jpg',
                fit:BoxFit.cover,
```

```
        width: double.infinity,),),
      Container(
        alignment: Alignment.center,
       child: ElevatedButton(
            onPressed: (){
              Navigator.of(context).push(MaterialPageRoute(
                builder: (context)=> TfliteModel(),),);},
            style: ElevatedButton.styleFrom(
              primary: Colors.black12,
              side: BorderSide(
                width: 0.5,
                color: Colors.white,),),
              child: Text(
                'Get Started',
                style: TextStyle(
                fontFamily: 'PatrickHand-Regular',
                fontSize: 24,),),),
       margin: EdgeInsets.only(left:50.0,right: 50.0,top: 180.0),),],),
    ),
   );
  }
}
```

**TfliteModel.dart File in Flutter**

```
import 'dart:io';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:tflite/tflite.dart';

class TfliteModel extends StatefulWidget {
  const TfliteModel({Key? key}) : super(key: key);
  @override
  _TfliteModelState createState() => _TfliteModelState();
}
class _TfliteModelState extends State<TfliteModel> {
  late File _image;
  late List _results;
  bool imageSelect = false;
  @override
  void initState() {
    super.initState();
    loadModel();
  }
  Future loadModel() async {
    Tflite.close();
    String res;
    res = (await Tflite.loadModel(
        model: "assets/CnnModel.tflite", labels: "assets/labels.txt"))!;
    print("Models loading status: $res");
  }
  Future imageClassification(File image) async {
    final List? recognitions = await Tflite.runModelOnImage(
```

```
    path: image.path,
    numResults: 6,
    threshold: 0.05,
    imageMean: 127.5,
    imageStd: 127.5,);
  setState(() {
    _results = recognitions!;
    _image = image;
    imageSelect = true;});
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    // backgroundColor: Colors.black,
    appBar: AppBar(
      leading: GestureDetector(
        onTap: () {},
        child: Icon(
          Icons.menu,),),
      title: const Text("Frapple",
        style: TextStyle(
          fontFamily: 'PatrickHand-Regular',
          fontSize: 28,),),
      elevation: 10,backgroundColor: Colors.teal[900],),
    body: ListView(
      children: [
        Image.asset('assets/images/appletop.png',
          fit:BoxFit.cover,),
        (imageSelect) ? Container(
          height: 200,
          width: 200,
```

```
  // margin: const EdgeInsets.all(20),
  margin: EdgeInsets.only(left:10.0,right: 10.0,top: 20,bottom: 0),
  child: Image.file(_image),
) : Container(
  // margin: const EdgeInsets.all(10),
  child: const Opacity(
    opacity: 0.8,
    child: Center(
      child: Text("No image selected, only select Apple images"),),),),
SizedBox(height: 20,),
SingleChildScrollView(
  child: Column(
    children: (imageSelect) ? _results.map((result) {
      return Card(
        child: Container(
          margin: EdgeInsets.all(15),
          child: Text(
            "${result['label']}",
            style: const TextStyle(color: Colors.red,
                // backgroundColor: Colors.lightGreen,
                fontFamily: 'PatrickHand-Regular',
                fontSize: 33,
              fontWeight: FontWeight.bold,),),),);
    }).toList() : [],),),
CustomButton(
  title: 'Pick from Gallery',
  icon: Icons.image_outlined,
  onClick: pickImage,),
CustomButton(
  title: 'Click Image',
  icon: Icons.camera,
```

```
            onClick: () => pickImagefromcamera(),),],),);}
  Future pickImage() async {
    final ImagePicker _picker = ImagePicker();
    final XFile? pickedFile = await _picker.pickImage(
      source: ImageSource.gallery,);
    File image = File(pickedFile!.path);
    imageClassification(image);}
  Future pickImagefromcamera() async {
    final ImagePicker _picker = ImagePicker();
    final XFile? pickedFile = await _picker.pickImage(
      source: ImageSource.camera,);
    File image = File(pickedFile!.path);
    imageClassification(image);}}
Widget CustomButton({
  required String title,
  required IconData icon,
  required VoidCallback onClick,}){
  return Container(
    width:80,
    child:ElevatedButton(
      onPressed: onClick,
      style: ElevatedButton.styleFrom(
        primary: Colors.teal[900],),
      child: Row(
        children: [
          Icon(icon),
          SizedBox(
            width:20,),
          Container(
              width: 10,),
          Text(title,style: TextStyle(
```

```
            fontFamily: 'PatrickHand-Regular',
            fontSize: 24,),)],),),
    margin: EdgeInsets.only(left:50.0,right: 50.0),);
}
```

# ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to Mr. Sanjay. B. Kalyankar, HOD-Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his generous guidance, help and useful suggestions.

We express our sincere gratitude to Dr. Padmapani P. Tribhuvan, Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We are extremely thankful to Dr. Ulhas Shiurkar, Director, Deogiri Institute of Engineering, and management Studies Aurangabad, for providing me infrastructural facilities to work in, without which this work would not have been possible.

**Signatures of Students**

Sakshi Sagar Kherdekar

Sakshi Rajesh Kakde

Rohit Dattatraya Paritkar