FRUIT IDENTIFICATION AND CLASSIFICATION

ABSTRACT

At supermarkets, the most important factor in selecting fresh fruits is fruit identity and quality indication. We can't inspect every fruit since it would take too much time and effort, and we always want to buy the freshest fruits when we go shopping. Fruits can get harmed, rotting, and impacted by their environment. With the aid of image processing and machine learning, we are able to recognise fruits and classify them on the different classes, making it simple for anyone to choose the fresh fruit available. In this project, we offer a useful technique for classifying fruits and indicating them. The type of fruits are determined using image processing and machine learning techniques. One of the fascinating uses of computer vision for both commercial and agricultural applications is the recognition and classification of fruits using deep learning. Nonetheless, because to the similarities in colour, shape, and size of fruits, researchers continue to have difficulty classifying them. By creating a technique for the identification and classification of fruits, this effort aims to address some of the difficulties encountered by the earlier researchers.

1.INTRODUCTION

Fruit identification entails distinguishing between several fruit varieties based on their shape, size, and other attributes. This process is vital for ensuring that fruits are accurately labelled and marketed, and for ensuring that consumers receive the desired product. On the other hand, quality indication entails determining the overall quality of fruits based on a variety of elements, including appearance, texture, taste, and chemical indicators. This technique is vital for ensuring that fruits satisfy market criteria for freshness, ripeness, and overall quality. Accurate quality assessment can also help to identify any defects or issues that may affect the marketability or profitability of the fruit crop. In recent years, technological advancements in spectroscopy, imaging, and other non-destructive methods have provided new tools for more accurate and efficient fruit identification and quality indication. These methods can help to reduce waste, increase efficiency, and improve overall fruit quality. Fruit identification and categorization are significant activities in the domains of horticulture, agriculture, and botany. It's crucial to correctly identify and categorize various fruit varieties for a number of reasons, including comprehending their distinctive traits, figuring out their market value, and informing breeding operations. Fruit identification entails identifying various fruit varieties based on their external and internal traits, such as size, shape, color, texture, and taste. This procedure is crucial to ensuring that fruits are properly marketed and labelled, as well as that customers receive the desired product. Accurate identification can also help to stop misidentification and the spread of invasive species.Computers can now recognise and categorise the items seen in films or photographs with the aid of deep learning algorithms. When a neural network in its early stages of development was able to categorise the object based on edge detection, the field of computer vision was first established in the 1950s. The availability of a wealth of information made possible by the advent of the internet hastened the advancement of this profession. The enormous amount of data created each day has significantly increased these systems' accuracy in just over a decade.The demand for goods is rising across all industries, which has led to an increase in automation, which in turn has increased the usage of computer vision and its applications. Every industry that relies on machines to analyse films, photos, etc. can clearly see the impact of this technology. By tackling the complications, computer vision seeks to improve efficiency while overcoming the limitations of the conventional system.Although computer vision is utilised in many other applications, this study focuses on several deep learning algorithms for classifying fruits.

1.1 PROBLEM STATEMENT

The challenge at hand is to create a system that can correctly recognise different fruit varieties and evaluate their quality based on exterior characteristics like colour, texture, and shape. This system should be able to process data from multiple sources such as images to provide real-time information on the condition of the fruits. Precise fruit identification and quality evaluation can assist growers and distributors in streamlining their supply chains, cutting waste, and offering customers superior goods. This is important for several reasons, including:

1. Food safety: Accurate identification of fruits is important to ensure that they are safe for consumption and free from harmful contaminants.
2. Agriculture: Identifying and classifying fruits can assist in breeding, cultivation, and production of new varieties, which can help to improve crop yields and reduce waste.
3. Marketing: Accurately identifying and classifying fruits can help to improve marketing efforts, as consumers are more likely to purchase fruits that are visually appealing and of high quality.
4. Nutrition: Different types of fruits have varying nutritional value, and accurate identification and classification can help consumers to make informed decisions about the fruits they consume based on their nutritional content.
5. Research: Accurate identification and classification of fruits can aid in research efforts related to their properties, potential uses, and health benefits.

1.2 NEED AND OBJECTIVE

There is an increasing need for precise and effective methods of fruit identification and quality assessment as the fruit industry grows internationally and consumer demand for fresh and high-quality fruits rises. Current manual methods are time-consuming, subjective, and prone to errors, leading to inefficiencies and waste. Therefore, the need for a fruit identification and quality indication system that can accurately identify various types of fruits and assess their quality based on external attributes such as colour, texture, and shape is crucial. A system like this can assist producers and distributors in streamlining their supply chains, cutting waste, and offering better products to customers.

The creation of a fruit identification and quality indication system has the dual goals of enhancing the fruit industry's productivity and effectiveness, as well as providing consumers with high-quality produce. The system should accurately identify different types of fruits and assess their quality in real-time, providing reliable results to farmers, distributors, and retailers. The system should process data from various sources, including images, videos, and sensors, and be effective, scalable, and affordable to implement. Meeting these objectives will assist optimise the supply chain management of fruits, eliminate waste, and meet consumer expectations for high-quality products.

1.3 SCOPE OF THE PROJECT

- Project entails establishing a system that can properly recognise different types of fruits and grade their quality based on external qualities such as color, texture, and form. The system will use computer vision and deep learning techniques to analyse images or videos of fruits and provide real-time information on their quality. To give a complete picture of fruit quality, the system can also incorporate data from sensors and other sources.
- The project will deal with data integration from various sources as well as the creation of algorithms for fruit identification and quality evaluation. Both common fruits like apples, bananas, and oranges as well as more unusual varieties will be compatible with the system, which will be made to work with a variety of fruits. The project will also include the development of hardware or software solutions to capture and process data in real-time. The final system should be efficient, scalable, and cost-effective to implement, providing a reliable solution for fruit identification and quality indication in the fruit industry.

1.4 PROPOSED SYSTEM

The proposed fruit identification and classification system will use computer vision and machine learning techniques to accurately identify and classify different types of fruits based on their external attributes such as color, texture, and shape. The system will comprise of the following components:

- The system will use cameras or other imaging devices to take pictures of or record videos of fruits.

- Pre-processing: In order to enhance the quality of the images or videos and the precision of the analysis that follows, pre-processing is applied to them.
- Fruit classification: The system will examine the photos or videos and determine the type of fruit using machine learning techniques like computer vision and deep learning.
- Fruit Classification: Based on their outward characteristics, such as colour, texture, and shape, the algorithm will further categorise the identified fruits. The system will use feature extraction and classification techniques to group the fruits into different categories.
- Data Integration: The system will integrate data from sensors and other sources to provide a comprehensive view of fruit quality.
- Output: The system will provide real-time output in the form of visual or audio feedback, enabling farmers, distributors, and retailers to make informed decisions about the types of fruits they are dealing with.

ADVANTAGES OF PROPOSED SYSTEM

- Accuracy: The suggested system accurately identifies and categorises various fruit varieties based on their physical attributes by utilising cutting-edge machine learning algorithms and computer vision technologies. When identifying fruits manually, this can lessen the likelihood of mistakes and discrepancies.
- Efficiency: The proposed method is created to be quick and effective, which can assist in lowering the time and effort necessary for fruit identification and classification. For extensive fruit production and processing operations, this can be extremely advantageous.
- Consistency: The suggested system yields consistent findings, which might aid in ensuring that fruits are correctly identified and categorised over time. In order to maintain product quality and guarantee food safety, this can be very crucial.
- Accessibility: Farmers, food scientists, and consumers may all readily access and utilise the suggested system. This can increase the fruit supply chain's transparency.

2.LITERATURE REVIEW

- External appearance It has been determined that a fruit's external look plays a significant role in determining its quality. For instance, research have revealed that external flaws like blemishes, bruising, or other flaws can drastically lower the market value of fruits (Gonzalez et al., 2018). Additionally, certain fruit varieties have distinctive exterior traits that can be used to identify them, such as the fruit's size, shape, and colour (Dhiman et al., 2020). Fruits have been found to have a variety of textures, which is another key quality indicator. According to studies, the firmness of fruits can be used to determine their level of maturity, with stiffer fruits typically being less ripe (Knoerzer et al., 2020). Similar to how soft spots or deterioration might point to subpar quality and decreased market value (Garcia-Sanchez et al., 2017). Chemical indicators Several chemical parameters, such as brix level, acidity, and pH level, have been employed to evaluate fruit quality. Indicators of fruit quality and ripeness, such as brix level, which quantifies the fruit's sugar content, have been identified (Wang et al., 2020). Furthermore, research has revealed that it is possible to evaluate the freshness and quality of fruits based on their pH and acidity levels (Singh et al., 2019).

- Non-destructive techniques Non-destructive techniques like imaging and spectroscopy have been utilised more and more to identify and rate fruit quality. For instance, studies have demonstrated that near-infrared spectroscopy may be used to precisely assess the sugar content of fruits, which is a major sign of their quality (Xie et al., 2021). Similar to this, imaging methods like hyperspectral imaging have been applied to determine fruit maturity and detect surface flaws (Li et al., 2019). The literature as a whole emphasises the value of carefully evaluating many aspects, such as appearance, texture, and chemical indications to precisely identify and assess the quality of fruits. Moreover, non-destructive techniques like imaging and spectroscopy can be used to do fruit quality assessments that are more precise and effective.

3. REQUIREMENTS

HARDWARE REQUIRMENTS:

Processing unit: Intel I3

8 GB of memory .

LED devices, 15" screen Webcam, mouse, and keyboard are required.

8GB of RAM is the absolute minimum, along with a fast internet connection.

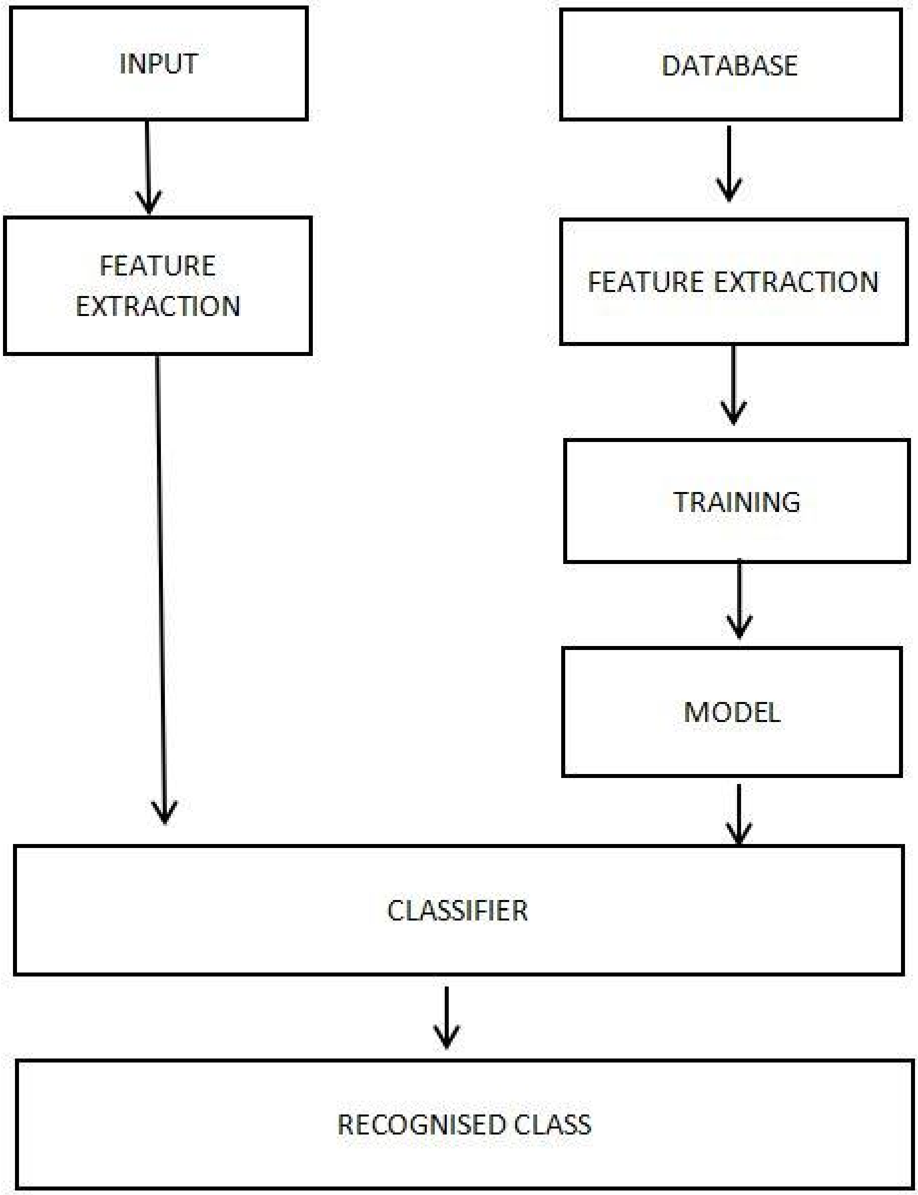
SOFTWARE REQUIRMENTS:

Operating System (OS): Windows 7 and later/UBUNTU

Programming Language (PL): Python Software (Jupyter Notebook)

Libraries TensorFlow, Keras, cv2, Pandas, Numpy, and Matplotlib are required.

4.SYSTEM DESIGN



4.1 USE CASE DIAGRAM

The many tasks carried out in the programme are depicted using diagrams. They also show the many users who can carry out these functions. Because of their focus on the activities carried out and the users (actors) who carry them out, use-case diagrams fall under the category of behaviour diagrams. The application's multiple functions include requesting

1.Training Fruit Images

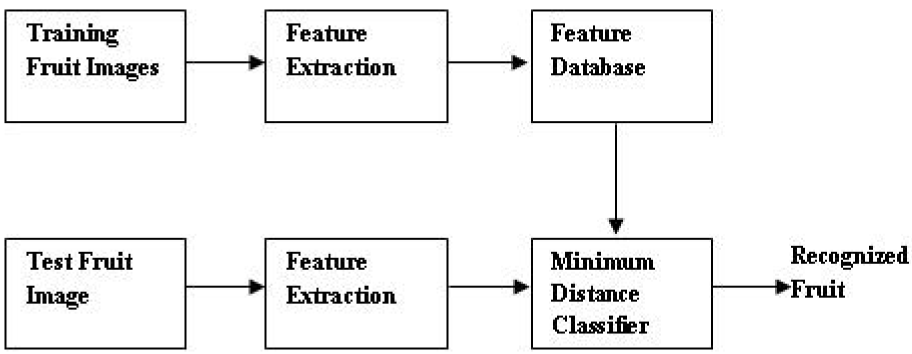
2.Feature Extraction

3.Feature Database

4.Test Fruit Image

5.Feature Extraction

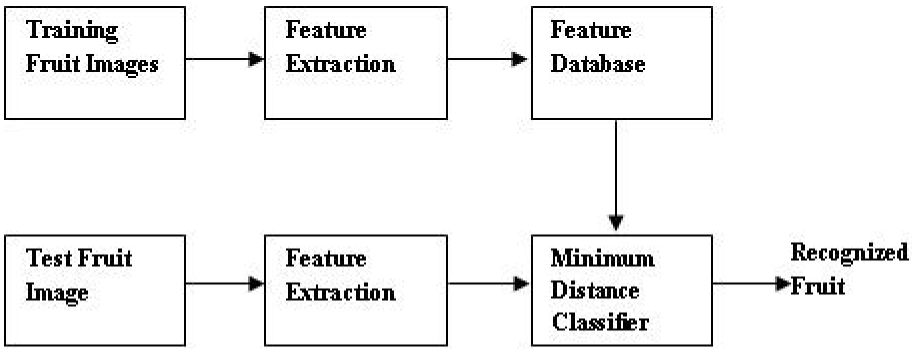
6.Minimum Distance Classifier



4.2 ACTIVITY DIAGRAM

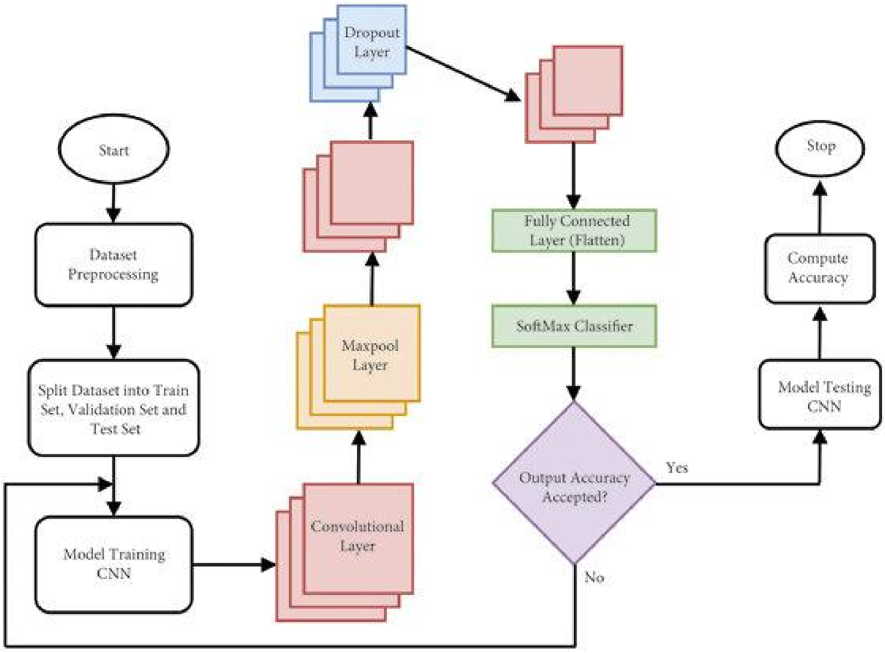
An application's control flow or sequence is shown in an activity diagram. In fact, we may utilise the activity diagram to validate each action depicted in the use-case diagram. It also shows the process steps. These essentially show how the software application's workflows are done. It also places emphasis on the order in which activities should be completed and the prerequisites that must be satisfied for each activity to be successful. In this manner, it informs us about the factors that influence a specific job. As a result, we can now see the application at a high level. The primary goals are to:

- Show the activity flow
- Explain the order (branched or sequential)

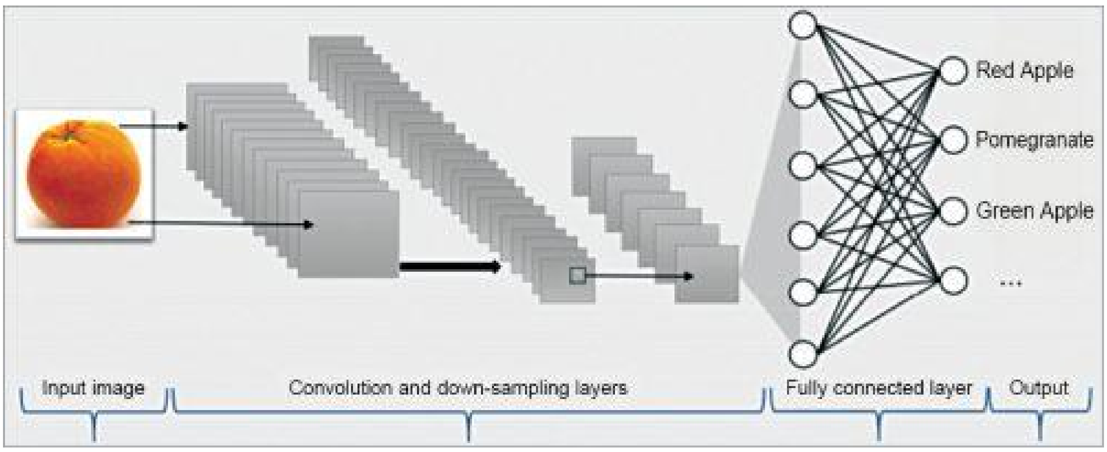


6. METHODOLOGY

The framework uses a camera to take a picture of the fruit, and the first phase uses a small neural network in TensorFlow to determine whether the image is of a natural product. The TensorFlow CNN neural system learning on a Linux server is then given the image to perform further grouping.



The Google-provided TensorFlow code modifies the union, pooling, and arranging configuration of the Inception-v3 model to coordinate the number of classes and classes of pixels in the image with minor changes to the last layer. Between the info and yield layers, in the concealed layer of CNN, there is a convolutional layer and a pooling layer.

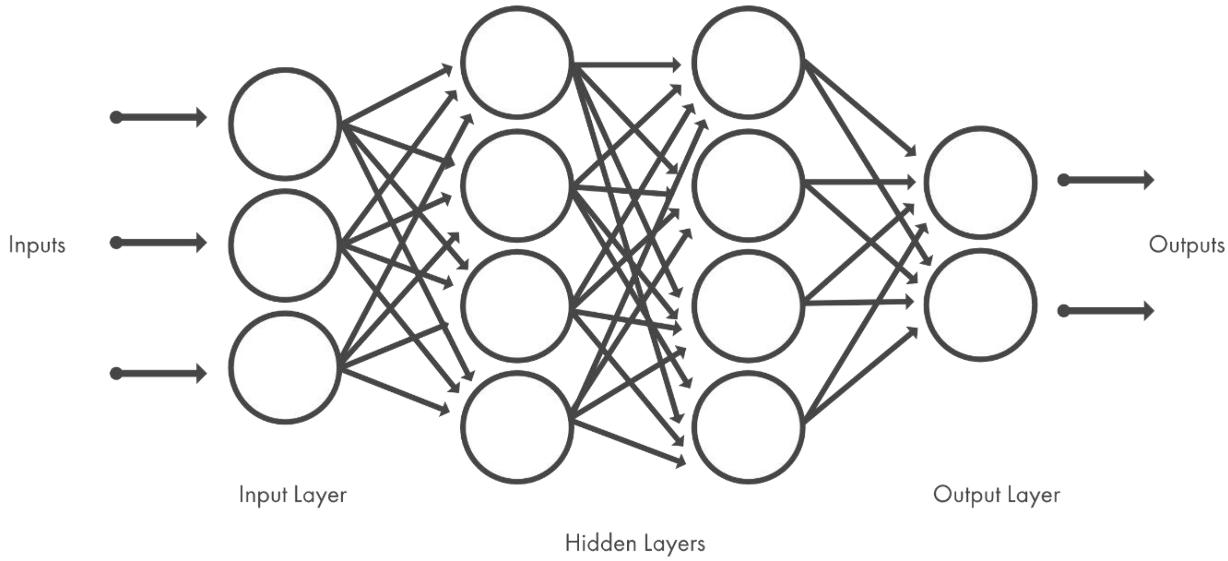


The method for bringing down or testing the picture's objectives is repeated in these two tiers. A portion of the information image that can be relevant for grouping is given a weighted channel by the convolution layer, creating an element map. By sub-testing the most important portion of the component map that it acquired from the convolutional layer, the pooling layer reduces the element map.

It decreases the size of the information while maintaining the qualities, preventing the difference in the information caused by the area change and enhancing the neural system's display by doing so. The categorization is done with these deleted highlights in mind.

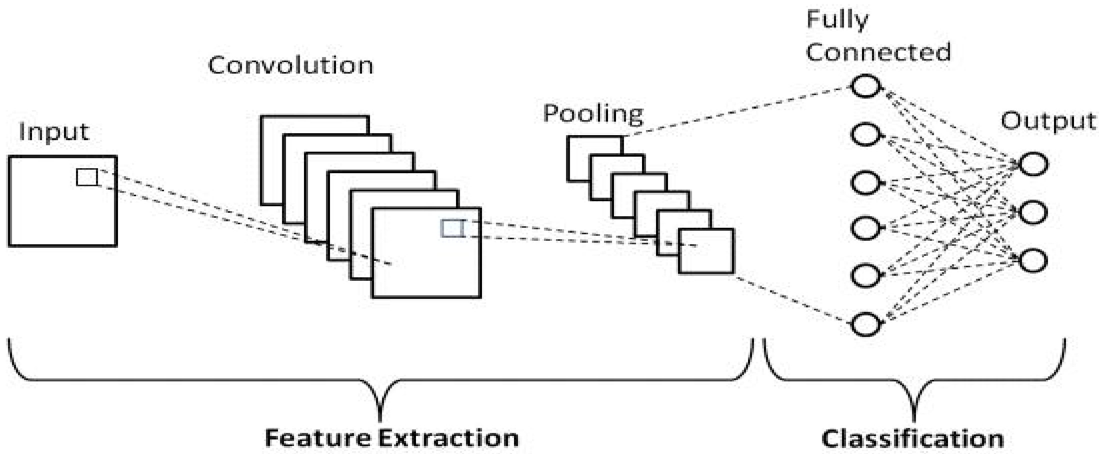
Deep Learning:

A Wing of ML called "deep learning" is totally supported by ANN. Deep learning may be thought of as a form of mimic of the human brain because neural networks are made to resemble it. With deep learning, not everything needs to be explicitly programmed. DL is not a fresh idea. It’s there in existence from some time. Because we did not have as much data and processing power back then, it's more common now. 14 million photos may be accessed by ImageNet, a well-known deep learning tool for image recognition, in its dataset-driven algorithms. For deep learning algorithms that use photos as their training dataset.



CNN:

CNNs, a kind of Deep Neural Network, are often used for image analysis and are effective at recognising and categorising certain characteristics in images. Image classification, image analysis for medical use, picture and video recognition, computer vision, and natural language processing are some of its uses. CNN is beneficial for picture recognition because of its high degree of accuracy. Several industries, including phone, security, medical image analysis, and recommendation systems, use image recognition.



CNN LAYERS:

1.Convolutional Layer

• The layer serves as the first step in the process of extracting different attributes from the input photos. In this layer, a filter with a size of MxM is mathematically convolutioned with the input picture. The dot product between the input picture's components and the filter.

• Convo is performed on the input by the CNN convolution layer, which then transfers the output to the subsequent layer. Convolutional layers in CNN are quite helpful since they preserve the spatial link between the pixels.

2. Pooling Layer

• A convolutional layer is frequently followed by a pooling layer. This layer's main objective is to scale down the convolved feature map in order to save on computing costs. By reducing the connections between layers and focusing solely on each feature map, this is achieved. Pooling activities may be categorised into a number of different groups depending on how they are carried out. It perfectly encapsulates the characteristics that a convolution layer generates.

3.Fully Connected Layer

At this point, the processing of categorisation begins here. Two layers are linked because they function better together than they do separately. The necessity for human monitoring is lessened by a number of levels in CNN.

4.DROPOUT

• When every attribute is linked to the FC layer, the training dataset is vulnerable to overfitting. Overfitting occurs when a given model performs so well on training data that it has a detrimental effect on the model's performance when applied to fresh data.

This issue is resolved by using a dropout layer, which reduces the size of the model by eliminating a few neurons from the neural network during training. When a dropout of 0.3 is reached, 30% of the nodes in the neural network are randomly removed. By lowering overfitting and network complexity, dropout improves the performance of machine learning models. Neurons are removed from neural networks during training.

5.ACTIVATION FUNCTION

The CNN model's last component, the activation function, is crucial. They are applied to the learning and estimation of any type of complex continuous network interaction between network variables. Simply said, it determines which model information has to be communicated to the network's end and which doesn't.

6. IMPLEMENTATION

6.1 Database Setup

We used fruits 360 data baseset from Kaggle which contains

Apples (different varieties: Golden, Golden-Red, Granny Smith, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red), Cactus fruit, Cantaloupe (2 varieties), Carambula, Cherry (different varieties, Rainier), Clementine, Cocos, Dates, Granadilla, Grape (Pink, White, White2), Grapefruit (Pink, White), Guava, Huckleberry, Kiwi, Kaki, Kumsquats, Lemon (normal, Meyer), Lime, Litchi, Mandarine, Mango, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine, Orange, Papaya, Passion fruit, Peach, Pepino, Pear (different varieties, Abate, Monster, Williams), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum, Pomegranate, Quince, Raspberry, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo.

6.2 Implementation steps

Implementation involves following steps:

1.Importing the required Libraries

2. Data loading

3.Labeling the data

4. Data Pre-processing

5.Visualizing the data with labels

6.Data Splitting

7. Building CNN Model

8.Traing model

9. Evaluating Model

1.Importing the required Libraries

We will need opencv, pandas, numpy, tensorflow, matplotlib, sklearn.

```
import os
import cv2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow import keras
import seaborn as sns
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

2. Data loading

Here we have to load the data from the directory.

```
train='data/fruits-360/Training'
file_names=os.listdir(train)
pd.DataFrame(file_names,columns=['Names'])
```

	Names
0	Apple Braeburn
1	Apple Golden 1
2	Apple Golden 2
3	Apple Golden 3
4	Apple Granny Smith
...	...
65	Salak
66	Strawberry
67	Strawberry Wedge
68	Tamarillo
69	Tangelo

70 rows × 1 columns

3. Data Labeling

In this step since our data do not have labels we use the directory names as labels and indicate the all the classes images we loaded with the directory names.

```
train_image=[]
label=[]
for file in file_names:
    path=os.path.join(train,file)
    for img in os.listdir(path):
        image=cv2.imread(os.path.join(path,img))
        train_image.append(image)
        label.append(file)
pd.DataFrame(label,columns=['label'])
```

	label
0	Apple Braeburn
1	Apple Braeburn
2	Apple Braeburn
3	Apple Braeburn
4	Apple Braeburn
...	...
35128	Tangelo
35129	Tangelo
35130	Tangelo
35131	Tangelo
35132	Tangelo

35133 rows × 1 columns

4. Data Pre-processing

Here we pre process the data with the images and the labels.

```
train_image=np.array(train_image)
label=np.array(label)
```

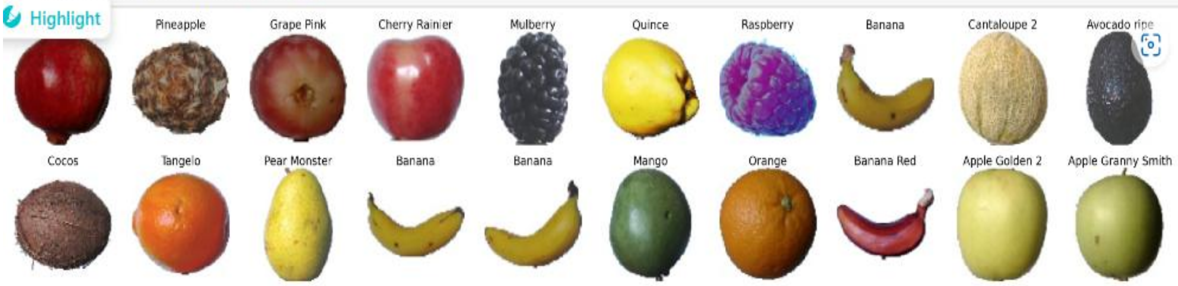
```
code={}
label_uniqe=list(pd.unique(label))
for i in range(70):
    code[label_uniqe[i]]=i
code
```

```
def get_Name(N):
    for x,y in code.items():
          if y==N:
                return x
label2=[]
for i in label:
    label2.append(code[i])
label2=np.array(label2)
pd.DataFrame(label2)
```

5. Visualizing the Data with labels

Here we visualize the input data with the labels.

```python
plt.figure(figsize=(50,50))
for n,i in enumerate(np.random.randint(0,len(train_image),100)):
    plt.subplot(10,10,n+1)
    plt.imshow(cv2.cvtColor(train_image[i], cv2.COLOR_BGR2RGB))
    plt.axis('off')
    plt.title(label[i],fontsize=25)
```



6. Data splitting

Here we split the data for training and evaluation purpose.

```python
X_train, X_test, y_train, y_test = train_test_split(train_image, label2, test_size=0.1, random_state=44, shuffle =True)
print('X_train shape is ' , X_train.shape)
print('X_test shape is ' , X_test.shape)
print('y_train shape is ' , y_train.shape)
print('y_test shape is ' , y_test.shape)
```

```
X_train shape is  (31619, 100, 100, 3)
X_test shape is  (3514, 100, 100, 3)
y_train shape is  (31619,)
y_test shape is  (3514,)
```

7. Building CNN Model

Here we build a Sequential model with keras, with class size of 131.

```python
shape=(100,100,3)
num_class=131
model=keras.models.Sequential()
model.add(keras.layers.Conv2D(filters=32, kernel_size=(3,3),activation=tf.nn.relu,input_shape=shape))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.MaxPool2D((3,3)))
model.add(keras.layers.Dropout(.3))
model.add(keras.layers.Conv2D(filters=64, kernel_size=(3,3),activation=tf.nn.relu))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.MaxPool2D((3,3)))
model.add(keras.layers.Dropout(.3))
model.add(keras.layers.Conv2D(filters=128, kernel_size=(3,3),activation=tf.nn.relu))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.MaxPool2D((3,3)))
model.add(keras.layers.Dropout(.3))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512,activation=tf.nn.relu))
model.add(keras.layers.Dropout(.3))
model.add(keras.layers.Dense(128,activation=tf.nn.relu))
model.add(keras.layers.Dropout(.3))
model.add(keras.layers.Dense(num_class,activation=tf.nn.softmax))
```

8. Traing Model

In this step we compile the model with loss function as sparse categorical crossentropy.and train the model with traing data and check the loss and accuracy of the model.

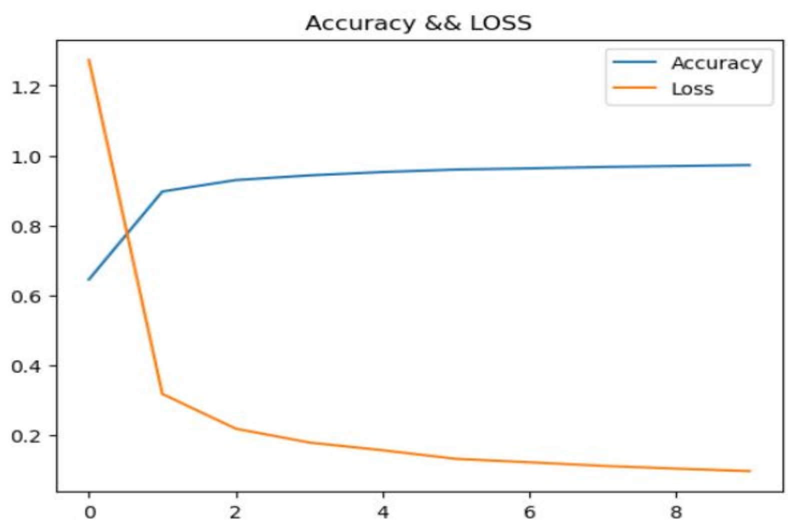
```
model.compile(optimizer ='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

```
hist=model.fit(X_train,y_train,epochs=5)
```

```
Epoch 1/5
989/989 [==============================] - 322s 326ms/step - loss: 0.2766 - accuracy: 0.9076
Epoch 2/5
989/989 [==============================] - 318s 322ms/step - loss: 0.1779 - accuracy: 0.9403
Epoch 3/5
989/989 [==============================] - 317s 320ms/step - loss: 0.1357 - accuracy: 0.9541
Epoch 4/5
989/989 [==============================] - 317s 321ms/step - loss: 0.1274 - accuracy: 0.9587
Epoch 5/5
989/989 [==============================] - 320s 324ms/step - loss: 0.1185 - accuracy: 0.9611
```

```
hist_=pd.DataFrame(hist.history)
hist_
```

	loss	accuracy
0	0.276647	0.907556
1	0.177926	0.940289
2	0.135715	0.954110
3	0.127411	0.958664
4	0.118538	0.961099



9. Evaluating model

After traing the model with traing data we now evaluate the trained model with the evaluation data and check the loss and accuracy of the evaluation.

```
score, acc = model.evaluate(X_test, y_test)
print('Test Loss =', score)
print('Test Accuracy =', acc)
```

```
110/110 [==============================] - 9s 75ms/step - loss: 0.4477 - accuracy: 0.9385
Test Loss = 0.4477287828922272
Test Accuracy = 0.9385315775871277
```

1. RESULTS

```
y_pred=model.predict(X_test)
y_pred
```

```
110/110 [==============================] - 8s 71ms/step

array([[2.8618351e-17, 2.8970579e-34, 6.6088787e-27, ..., 8.0366358e-31,
        3.2923751e-21, 2.6017429e-28],
       [9.7316791e-16, 7.9403722e-01, 1.0950502e-10, ..., 3.0835635e-18,
        3.6886296e-17, 1.5043847e-20],
       [9.2267943e-28, 4.3883767e-20, 1.3878387e-22, ..., 6.5784575e-23,
        9.1823050e-21, 3.8175409e-24],
       ...,
       [1.2580797e-17, 2.1756892e-19, 5.6350622e-29, ..., 5.8043360e-24,
        8.6107602e-20, 2.5458880e-21],
       [2.3388514e-20, 2.6739471e-09, 3.4058732e-12, ..., 8.6563525e-17,
        8.0822657e-14, 5.4019760e-20],
       [1.2712874e-19, 5.2954378e-18, 5.9624064e-20, ..., 4.8833500e-21,
        2.4230241e-17, 3.6571309e-18]], dtype=float32)
```

```
pred_Name=[]
pred_number=[]
for row in y_pred:
    N=np.argmax(row)
    pred_Name.append(get_Name(N))
    pred_number.append(N)
pd.DataFrame(pred_Name,columns=['pred Names'])
```

	pred Names
0	Apricot
1	Apple Golden 1
2	Melon Piel de Sapo
3	Cantaloupe 2
4	Tamarillo
...	...
3509	Apricot
3510	Lemon
3511	Mandarine
3512	Cantaloupe 2
3513	Kiwi

3514 rows × 1 columns

```
plt.figure(figsize=(50,50))
n=1
for i in range(10):
    plt.subplot(20,5,n)
    plt.imshow(cv2.cvtColor(X_test[i], cv2.COLOR_BGR2RGB))
    plt.axis('off')
    ti=get_Name(y_test[i])+', prediction: '+pred_Name[i]
    plt.title(ti,fontsize=25)
    n+=1
```

8.CONCLUSION

The classification and identification of fruits is a critical undertaking in the agricultural, food-processing, and retail sectors. Fruits should be properly identified and categorised to assist assure consumer satisfaction, quality control, and food safety. With the development of technology, a variety of procedures and techniques, including optical inspection, chemical analysis, molecular analysis, texture analysis, and AI/ML algorithms, are now accessible for the identification and classification of fruits. These techniques can be employed singly or in tandem to provide precise and effective outcomes. Modern methods and technology for fruit categorization and identification can greatly increase the accuracy and efficiency of this task. Accurate fruit identification and categorization are crucial for guaranteeing consumer safety and satisfaction given the rising demand for high-quality and safe food items. We can keep raising the calibre and security of the food we eat by developing and improving these techniques.

10. REFERENCES