Computer Vision Project Report

Plant Classification and Disease Recognition

#Team - SC_11

Name	Id
محمد محسن السيد عبد الرحمن طرابيه	**20191700570**
محمد محمود رجب محمد	**20201700734**
كيرلس مجدي فايز عبد النور	**20201700615**
محمد مدحت عبد اللطيف عبد الفتّاح	**2021170487**
يوسف محمد احمد كمال الدين ابراهيم	**20191700783**
عبد العزيز أحمد عبد العزيز عبد الحميد	**20191700374**

Outline:

- Data Reading & Preprocessing.
- Feature Extraction and Normalization.
- Classification Model & Accuracy.
- Verification Model & Accuracy.
- Validation Screenshots.

Data Reading & Preprocessing:

After loading the images folder:

Split the files name on (___) to get class name.

```
Classes in the LabelEncoder: ['Apple' 'Cherry_(including_sour)' 'Corn_(maize)' 'Grape' 'Peach'
 'Pepper,_bell' 'Potato' 'Strawberry' 'Tomato']
```

Convert to Grayscale:

Converts the image to grayscale to simplify the data and reduce the complexity of color channels.

Resize image to 244*244:

Ensures all images are of the same size (224x224 pixels) for consistency.

Blurring:

Used to reduce noise and smooth the image with Gaussian blur using window 5*5.

Label Encoding:

Converts the plant species and disease names into numeric labels. This transformation is necessary because machine learning models can only process numerical data.

```
Apple -> 0
Cherry_(including_sour) -> 1
Corn_(maize) -> 2
Grape -> 3
Peach -> 4
Pepper,_bell -> 5
Potato -> 6
Strawberry -> 7
Tomato -> 8
```

Shuffling for image:

Ensures randomness in the dataset, which helps avoid overfitting and ensures the model generalizes better by not learning patterns based on the order of the data.

Feature Extraction and Normalization:

Flattening Images:

Reshapes the images from 2D (height x width) to 1D (a flat array of pixels), which is necessary for inputting the images into the SVM model.

Standardization:

Scales the pixel values to have a mean of 0 and a standard deviation of 1. This is important because SVM is sensitive to the scale of input data.

SVM Accuracy & Time:

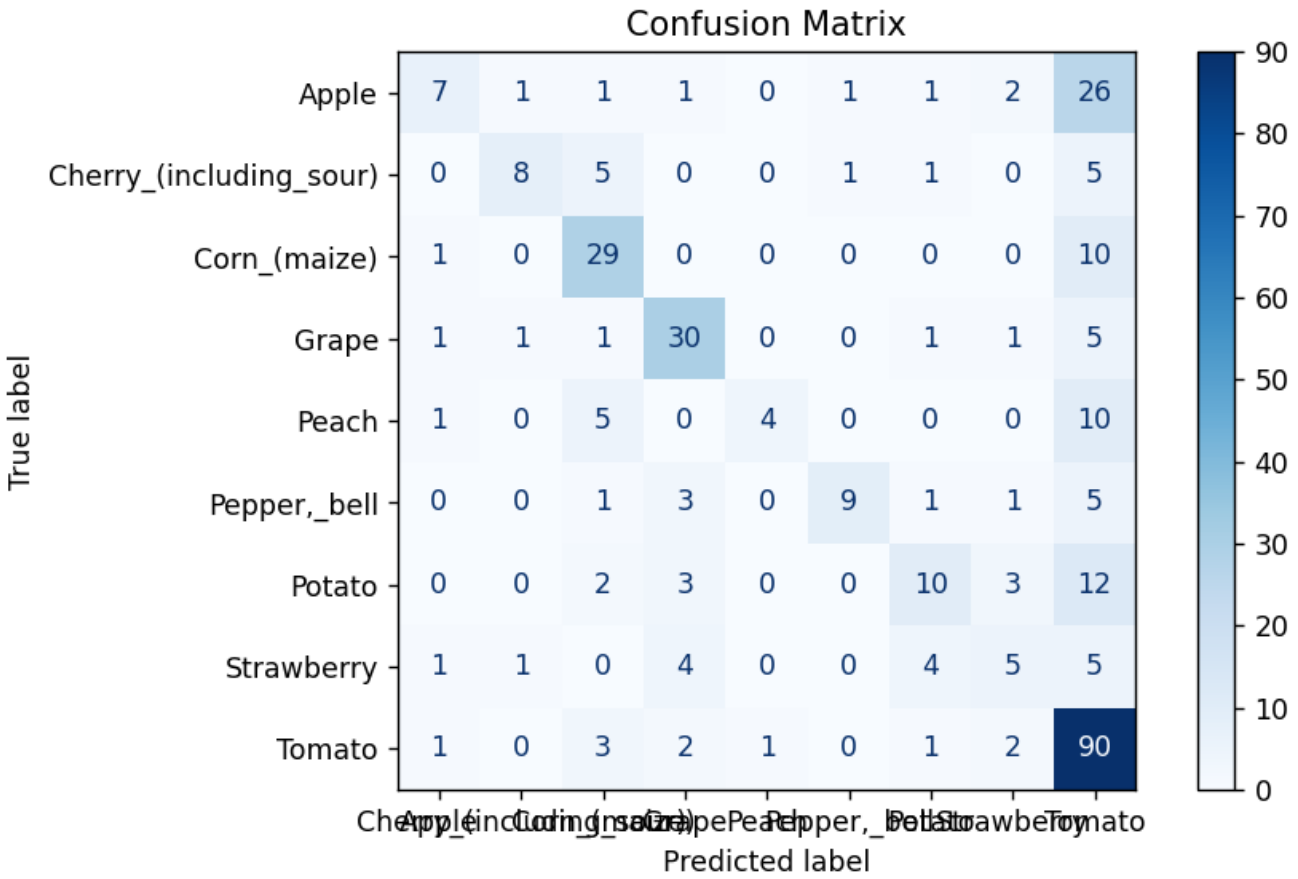
Support Vector Machine (SVM): A supervised learning algorithm used for classification tasks. It works by finding the optimal hyperplane that maximally separates data points of different classes. In this case, the SVM is used with an RBF kernel, which is effective for non-linear data, and the model is trained on the preprocessed plant images.

- RBF Kernel: Used to map the input data to a higher-dimensional space, making it easier to find a hyperplane that separates the classes.
- Hyperparameter Tuning (C): The C parameter in SVM controls the trade-off between achieving a low error on the training set and minimizing the complexity of the decision boundary. A value of 1 is commonly used as a starting point.

SVM Achieved accuracy reached to: **58.2 %**

With Training Time: **148.93** sec.

& Testing Time: **62.30** sec.



```
Training the SVM model...
Evaluating the SVM model...

Test Accuracy: 58.5 %

Training Time: 45.91 seconds
Evaluation Time: 31.93 seconds

Confusion Matrix:
<Figure size 800x600 with 0 Axes>
```

Save model using: Joblib:

A Python library used to save the trained model to disk, allows you to load these components later.

Verification: -

The Preparation:

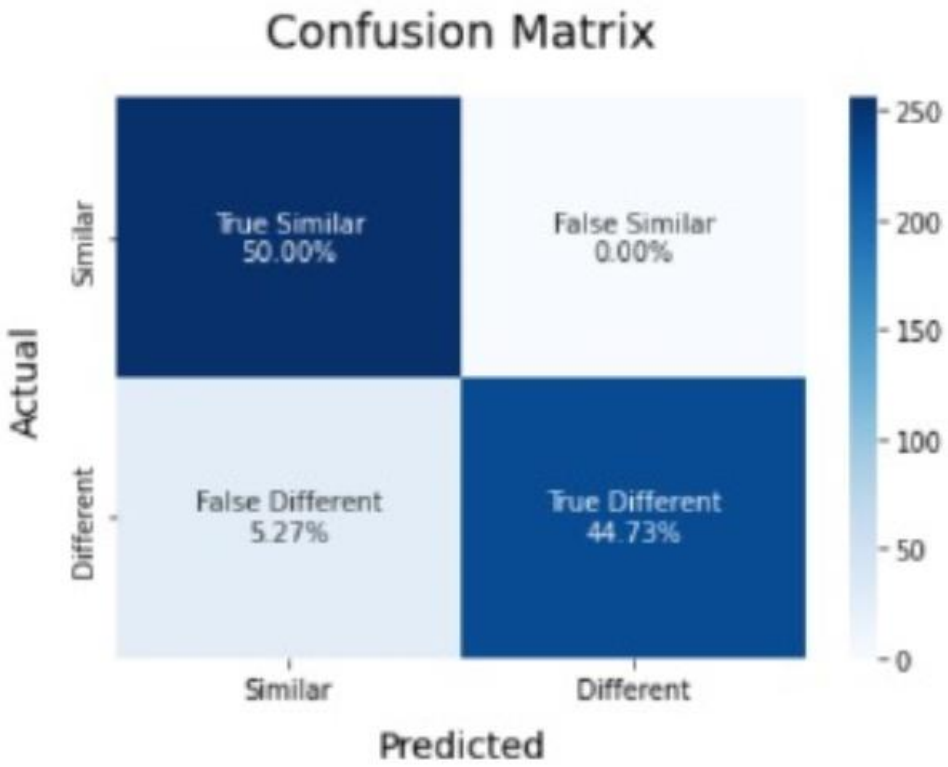
- The preprocessing is built-in as the preprocessing of the inception_v3.

- Building triplets of the anchor and the positive and negative images (positive is of the same class, negative is from any difference class).

The model:

The model is fine tuning for pre-trained Exception model with the weights of 'ImageNet', with only the last 27 layer as trainable and with additional 2 dense layers to give the model the chance if need to learn additional something.

Fine tuning achieved Accuracy reached to: **94.7 %**.



Evaluation:

we use the validation triplets for evaluation as we see the prediction with the positive image and the negative one and compare them and if the prediction with the positive is less than the one with the negative then it's a good prediction and calculate with the true predictions.

Preprocessing the test image...

Prediction Results:
Predicted Encoded Label: 5
Predicted Class Name: Peach

Predicted: Peach

Preprocessing the test image...

Prediction Results:
Predicted Encoded Label: 2
Predicted Class Name: Potato

Predicted: Potato

Preprocessing the test image...

Prediction Results:
Predicted Encoded Label: 1
Predicted Class Name: Grape

Predicted: Grape

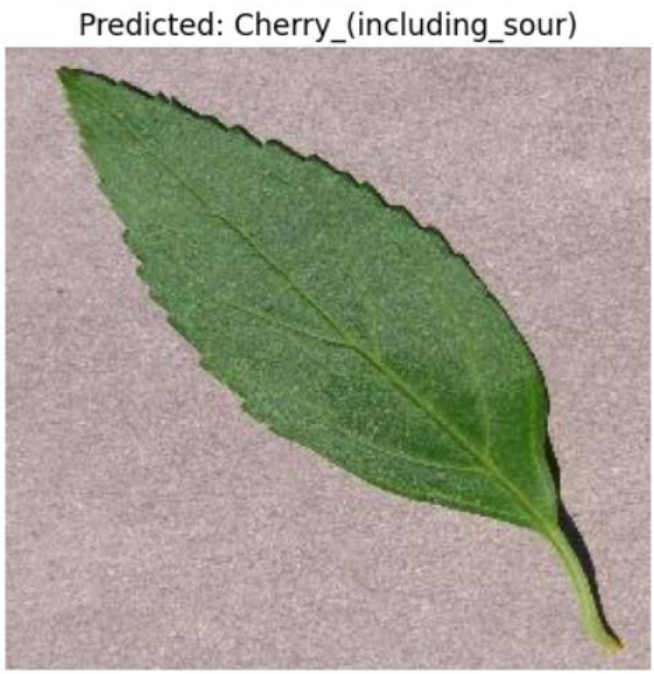
Preprocessing the test image...

Prediction Results:
Predicted Encoded Label: 6
Predicted Class Name: Apple

Predicted: Apple

```
Preprocessing the test image...

Prediction Results:
Predicted Encoded Label: 7
Predicted Class Name: Cherry_(including_sour)
```




```
Preprocessing the test image...

Prediction Results:
Predicted Encoded Label: 3
Predicted Class Name: Corn_(maize)
```