**Tomato Image Classification using Machine Learning Classification Algorithms**

In this project, images of 2 kinds of tomatoes from the market were captured on the smartphone. They were 1) Red tomatoes (ripened) and 2) Green tomatoes (unripen).

The images were then processed with a white background and loaded inside the system having two folders training and testing. The segregation of images was done and place into 2 separate sub-folders.

They were 1) Mobile captured red tomatoes and 2) Mobile captured green tomatoes. These sub-folders are included in both the folders: training and testing.

Then the process of training the machine learning model was carried out on the Jupyter notebook.

**Steps**:

1. Import the necessary libraries, packages of python, pandas, sklearn.
2. Function to get tomatoes from different sub-folders.
3. Initialize the tomatoes for training dataset.
4. Use K-fold for partitioning the dataset into K equal subsets.
5. Scale the data of test and train using StandardScaler.
6. Function to plot the image grid.
7. Use machine learning classification algorithms like Linear Support Vector Machine, Kernel Radial Basis Function (RBF) Support Vector Machine, K-Nearest Neighbors (K-NN) to calculate the performance metrics like confusion matrix, accuracy, precision, recall and F1 score.
8. Perform the dimensionality reduction using Principal Component Analysis (PCA) on all these algorithms and check for the performance metrics.
9. Repeat steps 7 and 8 on the testing dataset.

**Training Dataset**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Performance Metrics** | **Linear Support Vector Machine (SVM)** | **Kernel Radial Basis Function (RBF) Support Vector Machine (SVM)** | **K- Nearest Neighbor**  **(K-NN)** | **PCA on Linear SVM** | **PCA on Kernel RBF SVM** | **PCA on**  **K-NN** |
| 1 | Accuracy score | 100% | 100% | 97% | 96.5% | 96% | 98% |
| 2 | Precision score | 100% | 100% | 97.2% | 96.7% | 96.3% | 98.1% |
| 3 | Recall score | 100% | 100% | 97% | 96.5% | 96% | 98% |
| 4 | F1 score | 100% | 100% | 97% | 96.5% | 96% | 98% |

**Testing Dataset**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Performance Metrics** | **Linear SVM** | **Kernel RBF SVM** | **K-NN** | **PCA on Linear SVM** | **PCA on Kernel RBF SVM** | **PCA on**  **K-NN** |
| 1 | Accuracy score | 100% | 100% | 97% | 96% | 95.5% | 98% |
| 2 | Precision score | 100% | 100% | 97.2% | 96.3% | 95.9% | 98.1% |
| 3 | Recall score | 100% | 100% | 97% | 96% | 95.5% | 98% |
| 4 | F1 score | 100% | 100% | 97% | 96% | 95.5% | 98% |