- Name :

2021BEC0015: Adusumilli Laahiri

- Image Processing Task :

Plant Disease Detection in Apple (Scab Disease Detection)

- Details of Dataset :

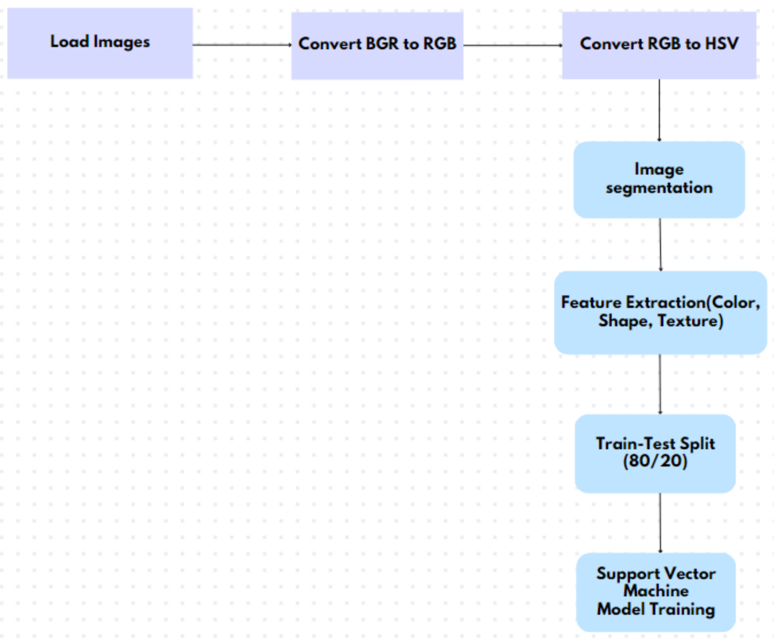
PlantDoc is a dataset for visual plant disease detection. The dataset contains 2,598 data points in total across 13 plant species and up to 17 classes of diseases

We are considering ML modeling for classification of healthy and unhealthy (Scab) for Apple leaf images.
Apple Scab images = 630
Healthy Apple leaf images = 800

Symptoms of Apple Scab on Leaves

- Small Spots: Small, brown or olive-green spots with indistinct margins appear on the underside of young leaves or on either surface of older leaves.
- Raised Tissue: The tissue beneath the lesions may become raised, resulting in a blistered appearance on the leaves.
- Curled Leaves: Leaves that are heavily infected may curl, shrivel, and eventually fall from the tree.
- Cupping: Lesions on older leaves may cause the underside of the leaf to cup.

These symptoms are indicative of Apple Scab, a common fungal disease affecting apple trees.



❖ Image Pre-processing :

▪ **Conversion of Image from BGR to RGB**

▪ The image is initially in BGR format.

▪ Since OpenCV (a popular Python library for image processing) accepts images in BGR format, conversion is necessary to ensure that colors are represented accurately in subsequent processing as RGB.

▪ **Conversion of Image from RGB to HSV**

▪ HSV separates the color information (Hue) from the intensity (Value), making it easier to process and analyze images based on color.

▪ This separation is beneficial for several reasons: Robustness to Lighting Changes, Shadow, Intuitive Color Manipulation.

▪ Apple scab symptoms are primarily characterized by specific colors (such as dark spots and yellowing), which can be influenced by lighting conditions. By converting images to the HSV color space, the model can focus on the hue (color) rather than the intensity (brightness), making it more robust to variations in lighting.

- **Image Segmentation**

▪ Image segmentation is performed to extract specific colors from the image (e.g., the green of healthy leaves and the brown of diseased leaves).

▪ It is useful for feature extraction, noise reduction and to isolate region of interest

❖ Feature Extraction :

Each of these functions extracts different types of features from the segmented image:

•**Hu Moments**: These are shape descriptors that capture the geometrical properties of the segmented areas, making them invariant to transformations (translation, rotation, scaling). (7 features)

•**Haralick Features**: These are texture descriptors derived from the Gray Level Co-occurrence Matrix (GLCM), providing information about the texture of the segmented areas. (13 features)

•**Histogram Features**: These capture the distribution of pixel intensities in the image, giving insight into the overall color composition. (8X8X8 =512 features)

•**Feature Scaling:** MinMaxScaler: This initializes the scaler to scale the features to the range [0, 1]. Many algorithms, including SVM and logistic regression, assume that all features contribute equally to the result. Scaling helps in ensuring that the model is not biased toward features with larger values.

After performing feature extraction and scaling, the final data consists of:

- **Rescaled Features (rescaled_features):**

A numpy array where each row represents a sample (an image of a plant leaf), and each column represents a feature (such as Hu moments, Haralick features, and histogram features).

The shape of rescaled_features is (1430, 532), meaning there are 1,430 samples, each with 532 features.

- **Encoded Labels (target):**

A numpy array containing the encoded labels for each sample. The shape of target is (1430,), indicating the label for each of the 1,430 samples.
The labels have been encoded to numerical values (e.g., 0 for healthy, 1 for diseased) using a label encoder, making it suitable for machine learning algorithms.

❖ Model Training :

- **Test_train_split:**

The dataset is split into training and testing sets. This is done using a stratified split to maintain the proportion of classes in both sets, ensuring that the model is trained and tested on a balanced dataset.

- **Model Training:**

The SVM classifier is trained using the training dataset. During this process, the model learns to associate the features extracted from the images with their corresponding labels.

- **Hyperparameter Tuning with PCA:**

To check which kernel to choose for the SVM, Principal Component Analysis (PCA) is employed for dimensionality reduction. PCA transforms the feature space to capture the most variance while reducing the number of features. By applying PCA, you can visualize the data and assess how well different kernels (like linear, polynomial, or radial basis function) separate the classes in the reduced feature space.

- **Model Evaluation:**

All the models are evaluated on the test set to assess its performance on unseen data, allowing to measure its accuracy, precision, recall, F1score, confusion matrix.

Kernel Type	Accuracy	Precision [0]	Precision [1]	Recall [0]	Recall [1]	F1 Score [0]	F1 Score [1]
Linear	94.06%	0.92	0.96	0.94	0.94	0.93	0.95
RBF	94.06%	0.92	0.96	0.95	0.93	0.93	0.95
Polynomial (degree=4)	93.36%	0.92	0.94	0.93	0.94	0.92	0.94
Sigmoid	91.61%	0.92	0.91	0.89	0.94	0.90	0.93