**EXPT NO: 6 A python program to implement face recognition**

**DATE: 24/10/2024 using Support Vector Machine.**

**AIM:**

To write a python program to implement face recognition using SVM.

**PROCEDURE:**

Implementing face recognition using svm involves the following steps:

**Step 1: Import Necessary Libraries**

First, import the libraries that are essential for data manipulation, visualization, and model building.

from sklearn.datasets import fetch\_lfw\_people import matplotlib.pyplot as plt from sklearn.model\_selection import train\_test\_split from sklearn.svm import SVC from sklearn.pipeline import make\_pipeline from sklearn.decomposition import PCA as RandomizedPCA from sklearn.metrics import accuracy\_score

**Step 2: Load the Dataset**

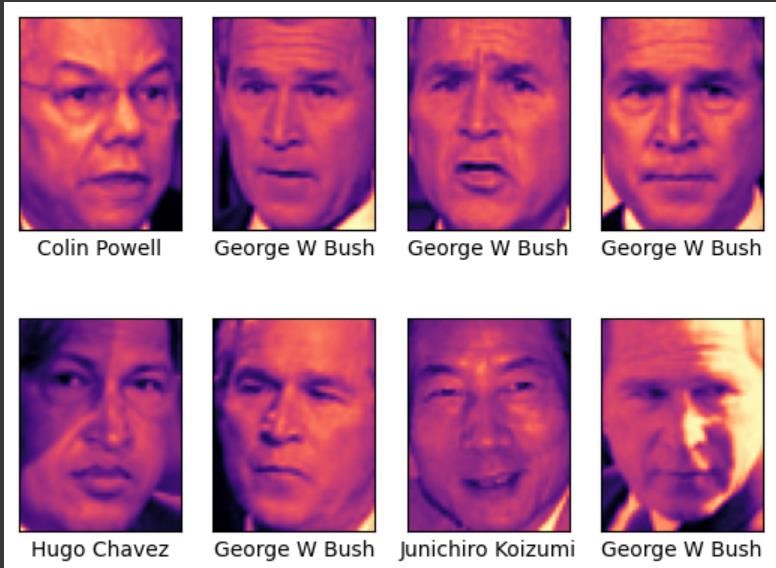
The dataset can be loaded and display the first few faces of the dataset.

faces = fetch\_lfw\_people(min\_faces\_per\_person=60) fig, splts = plt.subplots(2, 4)

for i, splts in enumerate(splts.flat): splts.imshow(faces.images[i], cmap='magma') splts.set(xticks=[], yticks=[],

xlabel=faces.target\_names[faces.target[i]])

**OUTPUT:**



\

**Step 4: Split the Data**

Split the data into training and testing sets.

Fit the dataset to the model.

X = faces.data y = faces.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=42)

**Step 5:**

**Dimensionality Reduction**

Reduce the dimension using Principal Component Analysis (PCA) Fit the model with SVC.

pca = RandomizedPCA(n\_components=150, whiten=True, random\_state=42) svc = SVC(kernel='rbf', class\_weight='balanced') model = make\_pipeline(pca, svc) model.fit(X\_train, y\_train)

**Step 6: Make Predictions**

Use the model to make predictions on the test data.

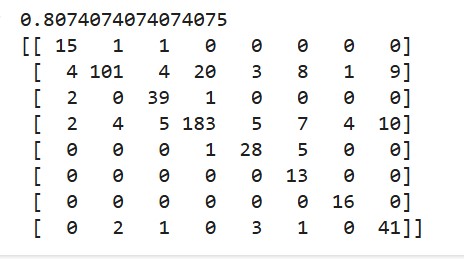
predictions = model.predict(X\_test)

**Step 7: Evaluate the Model**

Evaluate the model performance using metrics like Accuracy Score and confusion\_matrix

**OUTPUT :**

predictions = model.predict(X\_test) accuracy = accuracy\_score(predictions, y\_test) print(accuracy) matrix = confusion\_matrix(predictions, y\_test) print(matrix)



**RESULT:**

This step-by-step process will help us to implement face recognition using SVM and analyzed their performance.