Project Report

Overview:

Using the Yale face dataset, the code (project) applies Principal Component Analysis (PCA) to create a basic face recognition system. The system comprises several functions that load and preprocess face images, expressions, split the dataset, perform PCA on training images, project test images onto the PCA basis, back project projections, calculate loss, and finally predict the label based on the minimum loss. The face recognition system is then further used to recognize facial expressions. It introduces functions for loading images related to different facial expressions, splitting the dataset for expression recognition, performing PCA for each expression, and predicting expressions for test images. Furthermore, we have glasses/no glasses recognition function as well that differentiates and tells us whether the person in the given image is wearing glasses or not.

Functions:

Load_images:

The load_images function loads face images from a specified path (main folder path), resizes them (to 50x50 for faster computing), and flattens them into a 1D array.

Split_dataset:

This function divides the dataset into two separate sets (training set and testing set).

Perform_pca:

This function calculates the principal components using PCA on the training images.

Project_images:

this function projects test images onto the PCA basis obtained from the training set.

Back_project:

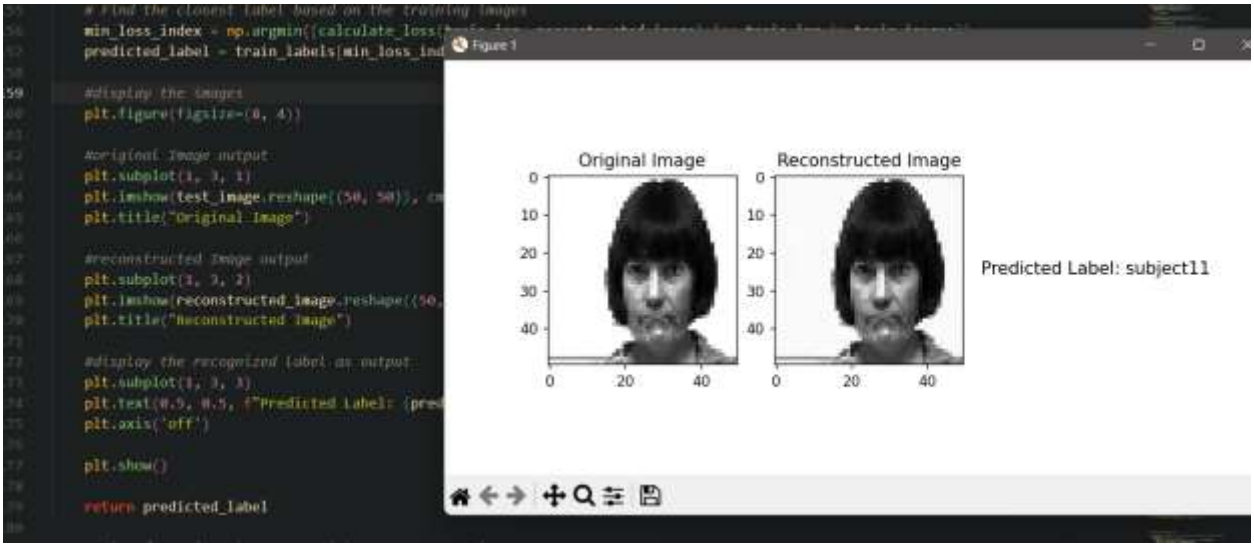
This function reverses the projection to obtain reconstructed images.

Calculate_loss:

This function measures the mean squared error between original and reconstructed images.

Face_recognition_for_single_image:

This function applies the face recognition pipeline to a single test image, returning the predicted label.




```
Task 1 Starting!!


Predicted Label: subject11
```

Load_images_by_expression:

This function loads face images associated with different expressions.

Split_expression_dataset:

This function divides the dataset for expressions into two separate sets (training set and testing set).

Perform_expression_pca:

The code performs PCA (Principal Component Analysis) separately for each facial expression using this function.

Recognize_expression:

This function projects test images onto PCA (Principal Component Analysis) bases and predicts expressions on human faces (happy, sad, wink etc.).

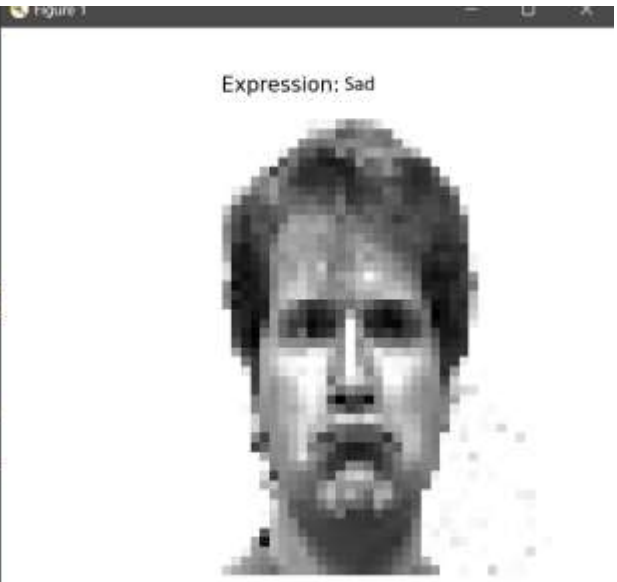
Test_single_image:

This function tests a single image for facial expression recognition, returning predictions (happy, sad, wink etc.).

```
def display_single_expression_image(image, expression):
    plt.figure(figsize=(5, 5))
    plt.imshow(image.reshape((50, 50)), cmap='gray')
    plt.title(f"Expression: {expression}")
    plt.axis('off')
    plt.show()

# Example of testing a single image
test_image_path = "D:/BSCS22122-LA-Project/salefaces/expression
result = test_single_image(test_image_path, expressions, pca_ba

if result is not None:
    predicted_expression = result[0][0]
    print(f"Predicted expression for {test_image_path}: {predic
    # Display the single expression image
    display_single_expression_image(load_test_image(test_image_
else:
    print("Test failed.")
```



Expression: happy

Load_Images_from_folder:
Loads images from glasses and no glasses folders.

Perform_PCA_glasses_no_glasses:
The perform_pca_glasses_no_glasses function applies perform_pca separately to images with glasses and without glasses, providing the mean faces and respective principal components for each category.

Test_glasses_recognition:
This function evaluates and prints whether each test image is predicted to have glasses or not based on the comparison of reconstruction losses using PCA for images with and without glasses. It iterates through test images, calculates losses, and determines the predicted status, printing the result for each image.

```
Test Image8: With Glasses
Test Image9: Without Glasses
Test Image10: Without Glasses
Test Image11: Without Glasses
Test Image12: With Glasses
Test Image13: With Glasses
Test Image14: Without Glasses
Test Image15: With Glasses
```