Homework5\18-661-HW5-Spring-2025\HW5\pca.py

```python
from scipy.io import loadmat
from matplotlib import pyplot as plt
import numpy as np

def pca_fun(input_data, target_d):
    # Step 1: Compute the mean of the input data
    mean_data = input_data.mean(axis=0)

    # Step 2: Center the data by subtracting the mean
    centered_data = input_data - mean_data

    # Step 3: Compute the covariance matrix
    covariance_matrix = np.cov(centered_data, rowvar=False)

    # Step 4: Compute eigenvalues and eigenvectors of the covariance matrix
    eigenvalues, eigenvectors = np.linalg.eigh(covariance_matrix)

    # Step 5: Sort eigenvectors by descending eigenvalues
    sorted_indices = np.argsort(eigenvalues)[::-1]
    sorted_eigenvectors = eigenvectors[:, sorted_indices]

    # Step 6: Select the top target_d eigenvectors
    P = sorted_eigenvectors[:, :target_d]

    # P: d x target_d matrix containing target_d eigenvectors
    return P

# Load the data
data = loadmat('face_data.mat')
images = data['image'][0]

# Vectorize each 50x50 image into a 2500-dimensional vector
vectorized_images = np.array([img.flatten() for img in images])

print(vectorized_images.shape)
# Perform PCA to compute eigenfaces
d = 200
eigenfaces = pca_fun(vectorized_images, d)

# Display the top 5 eigenfaces as images
for i in range(5):
    eigenface_image = eigenfaces[:, i].reshape(50, 50)
    plt.imshow(eigenface_image, cmap='gray')
    plt.title(f'Eigenface {i + 1}')
    plt.show()

# ### Data loading and plotting the image ###
# data = loadmat('face_data.mat')
# image = data['image'][0]
# person_id = data['personID'][0]

```

```
# plt.imshow(image[0], cmap='gray')
# plt.show()
```