SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Ćwiczenie Nr 5 Data 17.03.2024 Temat: "Analiza "eigenfaces"" Wariant 9	Dominik Katana Informatyka II stopień, stacjonarne, II semestr, gr.1

Link do repozytorium:

https://github.com/Dominowy/MK

1. Polecenie: wariant 9 zadania

Zadanie polega na wykorzystaniu analizy głównych składowych (PCA) do redukcji wymiarowości obrazu twarzy i zachowania określonej części informacji wyrażonej jako procent (k%) oryginalnej informacji. Następnie należy przedstawić zredukowane zdjęcie z użyciem odpowiedniej liczby "eigenfaces".

Wariant zadania (k):

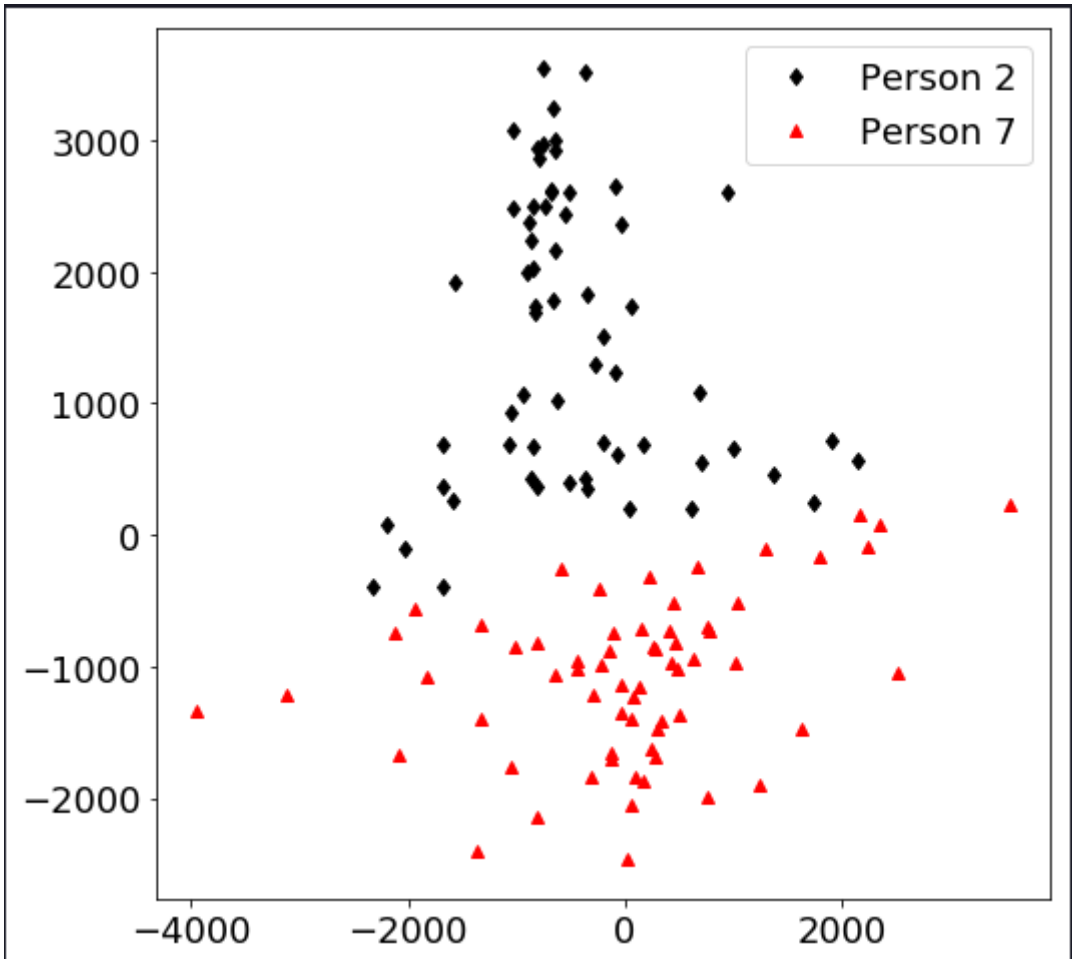
9. 90

2. Opis programu opracowanego (kody źródłowe, zrzuty ekranu)

```
mport matplotlib.pyplot as plt
import numpy as np
import os
import scipy.io

plt.rcParams['figure.figsize'] = [8, 8]
plt.rcParams.update({'font.size': 18})

try:
    mat_contents = scipy.io.loadmat('allFaces.mat')
except FileNotFoundError:
    print("Nie można znaleźć pliku allFaces.mat")
except Exception as e:
    print(f"Wystąpił problem podczas wczytywania pliku: {e}")
```

```
    exit()

faces = mat_contents['faces']
m = int(mat_contents['m'])
n = int(mat_contents['n'])
nfaces = np.ndarray.flatten(mat_contents['nfaces'])

# We use the first 36 people for training data
trainingFaces = faces[:, :np.sum(nfaces[:36])]
avgFace = np.mean(trainingFaces, axis=1)  # size n*m by 1

# Compute eigenfaces on mean-subtracted training data
X = trainingFaces - np.tile(avgFace, (trainingFaces.shape[1], 1)).T
U, S, VT = np.linalg.svd(X, full_matrices=0)

# Plot singular values
plt.figure(1)
plt.semilogy(S)
plt.title('Singular Values')
plt.show()

plt.figure(2)
plt.plot(np.cumsum(S) / np.sum(S))
plt.title('Singular Values: Cumulative Sum')
plt.show()

# Plot average face and the first eigenface
fig1 = plt.figure()
ax1 = fig1.add_subplot(121)
img_avg = ax1.imshow(np.reshape(avgFace, (m, n)).T)
img_avg.set_cmap('gray')
plt.axis('off')

ax2 = fig1.add_subplot(122)
img_u1 = ax2.imshow(np.reshape(U[:, 0], (m, n)).T)
img_u1.set_cmap('gray')
plt.axis('off')

plt.show()

# Now show eigenface reconstruction of image that was omitted from
test set
```

```
testFace = faces[:, np.sum(nfaces[:36])]  # First face of person 37
plt.imshow(np.reshape(testFace, (m, n)).T, cmap='gray')
plt.title('Original Image')
plt.axis('off')
plt.show()

testFaceMS = testFace - avgFace

# Reconstruction for different values of r
r_list = [25, 50, 100, 200, 400, 800, 1600]

for r in r_list:
    reconFace = avgFace + U[:, :r] @ U[:, :r].T @ testFaceMS
    plt.imshow(np.reshape(reconFace, (m, n)).T, cmap='gray')
    plt.title(f'r = {r}')
    plt.axis('off')
    plt.show()

# Project person 2 and 7 onto PC5 and PC6
P1num = 2  # Person number 2
P2num = 7  # Person number 7

P1 = faces[:, np.sum(nfaces[:(P1num - 1)]):np.sum(nfaces[:P1num])]
P2 = faces[:, np.sum(nfaces[:(P2num - 1)]):np.sum(nfaces[:P2num])]

P1 = P1 - np.tile(avgFace, (P1.shape[1], 1)).T
P2 = P2 - np.tile(avgFace, (P2.shape[1], 1)).T

PCAmodes = [5, 6]  # Project onto PCA modes 5 and 6
PCACoordsP1 = U[:, PCAmodes - 1].T @ P1
PCACoordsP2 = U[:, PCAmodes - 1].T @ P2

plt.plot(PCACoordsP1[0, :], PCACoordsP1[1, :], 'd', color='k',
label='Person 2')
plt.plot(PCACoordsP2[0, :], PCACoordsP2[1, :], '^', color='r',
label='Person 7')

plt.legend()
plt.show()
```


Person 2
Person 7
3000
2000
1000
0
−1000
−2000
−4000
−2000
0
2000

3. Wnioski

W analizie danych twarzy za pomocą PCA udało się zidentyfikować istotne składowe informacji oraz zrekonstruować twarze na podstawie ich średniej i pierwszych składowych własnych. Zaobserwowane singular values oraz kumulatywne sumy singular values pomagają zrozumieć, jak dużo informacji jest zachowywane przy różnych poziomach redukcji wymiarów. Projektowanie osób na przestrzeni PCA pokazuje, jak można różnicować osoby na podstawie ich cech twarzy w abstrakcyjnym, numerycznym kontekście.