

MYZ - 3. Ödev

1. Bu ödevde Temel Bileşen Analizi (TBA-PCA) kullanımını gerçeklemeniz hedeflenmektedir. Bu amaçla Yale veri tabanından seçilerek verilmiş olan gri-düzy kesilmiş –yüz- görüntülerini tek boyutlu vektörler olarak okuyunuz ve TBA analizinde öznitelik (feature) vektörleri olarak kullanınız.

```

> from matplotlib import pyplot as plt
from matplotlib.image import imread
import numpy as np
import os

TOTAL_IMG_FOLDER = "C:/Users/PC/Desktop/myz/odev_3/faces/"

train_set_files = os.listdir(TOTAL_IMG_FOLDER)

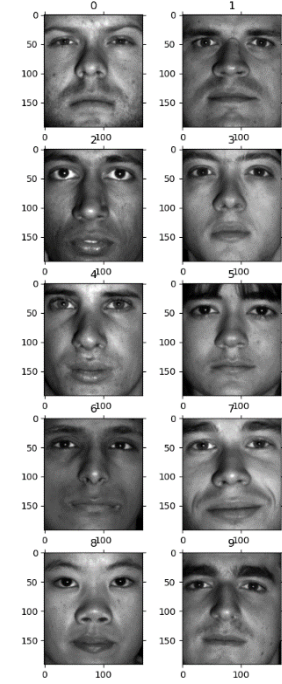
width = 168
height = 192

print('Görseller:')
train_image_names = os.listdir(TOTAL_IMG_FOLDER)
training_tensor = np.ndarray(shape=(len(train_image_names), height*width), dtype=np.float64)

print(training_tensor.shape)
fig, axes = plt.subplots(5, 2, figsize=(5, 13))
for i in range(len(train_image_names)):
    img = plt.imread(TOTAL_IMG_FOLDER + train_image_names[i])
    training_tensor[i, :] = np.array(img, dtype='float64').flatten()
    row = i // 2
    col = i % 2
    axes[row, col].set_title(i)
    axes[row, col].imshow(img, cmap='gray')
    axes[row, col].tick_params(labelleft='off', labelbottom='off', bottom='off', top='off', right='off', left='off', which='both')
plt.show()

... Görseller:
(10, 32256)

```



Yukarıdaki kod parçası sayesinde veri setindeki her bir görseli tek boyutlu vektörler olarak okuyarak bir liste içerisinde birleştirdim. Görseller 192x168 boyutunda olduğu için lineerleştirdiğimde 10 adet 32256 uzunluğunda bir vektör elde ettim her görsel için.

Aynı zamanda da resimleri görselleştirdim.

```

mean_face = np.zeros((1, height*width))

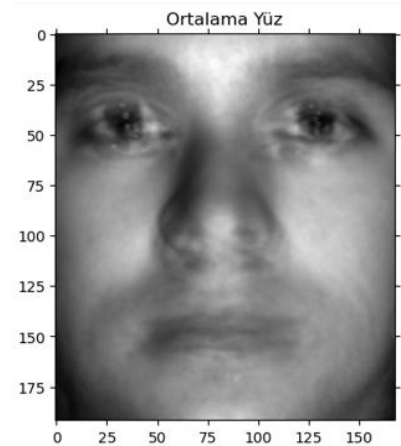
for i in training_tensor:
    mean_face = np.add(mean_face, i)

mean_face = np.divide(mean_face, float(len(train_image_names))).flatten()

plt.imshow(mean_face.reshape(height, width), cmap='gray')
plt.tick_params(labelleft='off', labelbottom='off', bottom='off', top='off', right='off', left='off', which='both')
plt.title("Ortalama Yüz")
plt.show()

```

Ödev içerisinde sorulmamış, fakat merak ettiğimden yukarıdaki kod parçası ile ortalama yüz değerini de oluşturdum.



- a. (15 puan) Öznitelik vektörlerini kullanarak yüz görüntüleri için kovaryans matrisini hesaplayınız.

Kovaryans matrisini oluşturduğum 10x32256 şeklinde olan training_tensor ile np.cov(matrix) komutu ile oluşturarak 10x10 boyutunda bir matrix elde ettim. Kod ve kovaryans matrisi aşağıda belirtilmiştir.

```
cov_matrix=np.cov(training_tensor)

print('Covariance Matrix Shape:', cov_matrix.shape)

✓ 0.0s

Covariance Matrix Shape: (10, 10)
```

```
[[1416.25735111 568.10190202 731.88579292 899.7391961 950.24536766
 762.36179494 555.15156806 882.26556686 839.04786997 551.04867764]
 [ 568.10190202 759.17595188 334.43233888 624.75227412 556.3327408
 495.64510399 318.25338835 530.11002108 344.95788131 436.0445504 ]
 [ 731.88579292 334.43233888 1471.05048123 973.13608089 1065.52855177
 755.84401598 492.05552957 877.65621035 701.97792132 391.0278583 ]
 [ 899.7391961 624.75227412 973.13608089 1603.23662997 1129.45133217
 1070.94116849 482.19871945 929.79671592 618.72277043 584.75315014]
 [ 950.24536766 556.3327408 1065.52855177 1129.45133217 1823.52824561
 944.3684463 587.08866268 972.67814489 789.47524731 733.99382353]
 [ 762.36179494 495.64510399 755.84401598 1070.94116849 944.3684463
 1864.85023238 579.58371398 990.3754344 821.16314712 538.7677501 ]
 [ 555.15156806 318.25338835 492.05552957 482.19871945 587.08866268
 579.58371398 684.17031353 522.14434624 516.48325739 402.09499029]
 [ 882.26556686 530.11002108 877.65621035 929.79671592 972.67814489
 990.3754344 522.14434624 1660.9309702 806.8953302 434.51874716]
 [ 839.04786997 344.95788131 701.97792132 618.72277043 789.47524731
 821.16314712 516.48325739 806.8953302 1450.67959615 396.55639271]
 [ 551.04867764 436.0445504 391.0278583 584.75315014 733.99382353
 538.7677501 402.09499029 434.51874716 396.55639271 1332.16749339]]
```

- b. (25 puan) Kovaryans matrisinin öz değerlerini ve bunlara karşı düşen öz vektörleri hesaplayınız. En yüksek öz değere karşı düşen öz vektörden başlayarak 10 tane öz vektör seçiniz. Seçilen ilk 5 öz vektörü (öz yüz) görüntü olarak kaydederek raporlayınız.

Özdeğer ve Özvektörleri numpy kütüphanesindeki lineer cebir alt kütüphanesi olan linalg'da bulunan eig komutu ile oluşturabildim. Sırası ile eigenvalue ve eigenvector olmak üzer sonuçları aşağıdaki görselde gösterilmiştir.

```
eigenvalues, eigenvectors, = np.linalg.eig(cov_matrix)
print('Özdeğer boyutları: {} Özvektör boyutları: {}'.format(eigenvalues.shape, eigenvectors.shape))

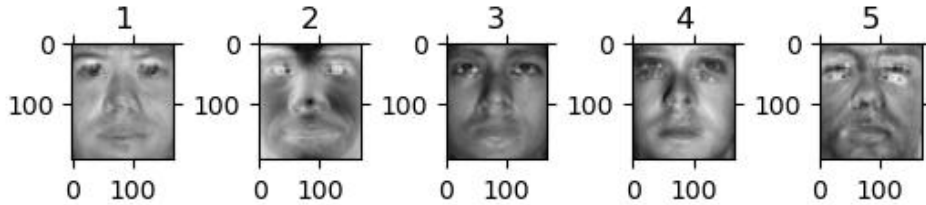
print("Özdeğerler:\n", eigenvalues, "\n")
print("Özvektörler:\n", eigenvectors)
```

```
Özdeğer boyutları: (10,) Özvektör boyutları: (10, 10)

Özdeğerler:
[7865.34977518 1182.02133222 990.38729992 1029.56242569 763.6354239
 648.25725594 277.0216401 364.94676379 417.60376188 518.46780265]

Özvektörler:
[[ 0.33333188 -0.03767364 -0.38789166 -0.04385294 -0.37126552 0.44958567
 -0.14037141 0.43713145 0.4293401 0.00110012]
 [ 0.19590095 -0.20928126 0.02233181 -0.12531673 -0.41555627 0.16971051
 -0.46332797 -0.67604635 -0.16320189 -0.04226327]
 [ 0.32663567 -0.14826655 0.02961859 0.5696421 0.29576947 -0.09953608
 -0.34916078 -0.10314312 0.19385646 0.52767196]
 [ 0.37266824 -0.10222196 0.41583535 0.18050091 -0.175984 0.39581756
 0.45249298 0.1072553 -0.43269371 0.23248672]
 [ 0.39917583 -0.24869891 -0.0083792 0.38790673 0.25423159 -0.06407518
 -0.00162302 -0.00162006 -0.00972933 -0.74794001]
 [ 0.36807502 0.25066532 0.54159047 0.553575 0.28865077 0.04522384
 -0.21990045 0.08663108 0.21163551 -0.09177528]
 [ 0.20432589 -0.02565471 -0.14054109 -0.11463469 0.08120666 -0.01204898
 0.61490902 -0.51956612 0.5131001 0.08140267]
 [ 0.35923792 0.32954454 -0.01555093 0.00250023 -0.52305601 -0.67046006
 0.10253729 0.10274944 -0.07663419 -0.03550562]
 [ 0.29765535 0.34285912 -0.50952497 -0.24787966 0.34395102 0.11813649
 0.0313546 -0.07969213 -0.49392375 0.03621803]
 [ 0.22389823 -0.75250407 -0.11830571 -0.30300933 0.15680426 -0.34918736
 -0.03147641 0.18807026 -0.07898492 0.29778703]]
```

Daha sonrasında ise özdeğerleri sıralayarak en büyük 5 özdeğere karşılık düşen özvektörlerden "Eigenface"leri oluşturdum. Adımlar biraz uzun olduğundan sadece sonuç olarak oluşturduğum görselleri koyuyorum.



- c. (20 puan) Seçilen öz vektörleri baz vektörü olarak kullanarak yüz görüntülerini yeni vektör uzayına izdüşürünüz. İzdüşürülen görüntüleri raporlayınız.

```
k = 10

eigenfaces = np.dot(normalised_training_tensor.T, eigvectors_sort[:k].T)

for i in range(eigenfaces.shape[1]):
    eigenfaces[:, i] /= np.linalg.norm(eigenfaces[:, i])

W = eigenfaces
```

Sıraladığım özvektörlerin 10 tanesini seçerek yüz görüntülerini vektör uzayına izdüşürdüm. Bunun sonucunda ise resimleri tekrardan oluşturmayı başardım. Aynı fonksiyon içerisinde de aynı zamanda Öklid uzaklıklarını bulmayı başardım.

```
reconstructed_images = [] # Yeniden yapılandırılmış görüntüleri saklamak için liste
reconstruction_errors = []
num_images = train_projections.shape[0]

# Her bir görüntü için yeniden yapılandırma işlemi:
for i in range(num_images):

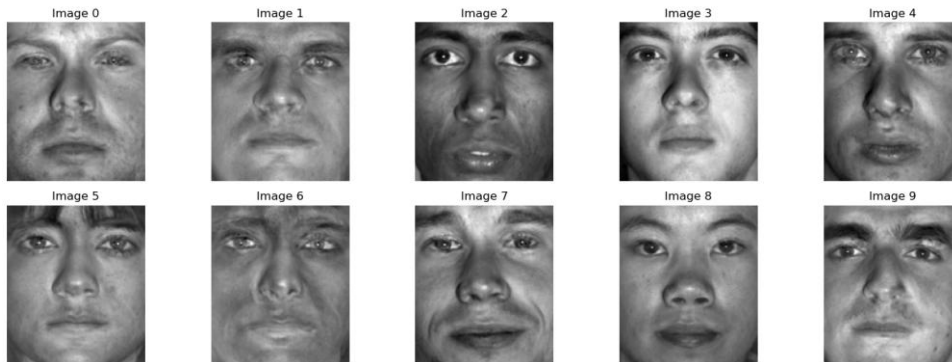
    # Proje katsayıları (shape: (k,))
    proj_coeff = train_projections[i]

    recon_vector = mean_face + np.dot(W, proj_coeff)
    recon_image = recon_vector.reshape(height, width)
    reconstructed_images.append(recon_image)

    error = np.linalg.norm(training_tensor[i] - recon_vector)
    reconstruction_errors.append(error)

# Görselleştirme (örneğin 2 satır 5 sütunluk grid şeklinde)
fig, axes = plt.subplots(2, 5, figsize=(15, 6))
axes = axes.flatten()
for i, ax in enumerate(axes):
    if i < num_images:
        ax.imshow(reconstructed_images[i], cmap='gray')
        ax.set_title(f"Image {i}")
        ax.axis('off')
plt.suptitle("Yeniden Yapılandırılmış Görüntüler")
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

Yeniden Yapılandırılmış Görüntüler



d. (20 puan) (c) de hesaplanan izdüşürülmüş yüzler ile orijinal yüzler arasındaki Euclidean uzaklığı hesaplayarak raporlayınız.

Öklid uzaklıklarını c şıkında bahsettiğim algoritma içerisinde elde ederek orijinal resimlere göre aşağıda belirttiğim şekilde buldum.

[8098.306258001109, 4150.792848123523, 3371.824131233844, 1405.5784688237307, 5040.248870054061, 6225.909598406936, 9975.40388748064, 9018.081813486822, 3130.7024786204047, 3047.0141857002523]

e. (20 puan) (c) deki değerlere baktığınızda, (b) de seçmiş olduğunuz öz yüzlerin elinizdeki 10 adet yüz görüntüsünü modellemekte yeterli olup olmadığı hakkında görüşünüzü yazınız. Yeterli / yeterli değil ise ne önerirsiniz?

Seçmiş olduğumuz öz yüzler resimleri mükemmel seviyede tekrar oluşturmakta yeterli olamamaktadır, aynı zamanda 2. Görselde başarılı bir recreation oluşturabilirsek de, 6. Görsel gibi örneklerde daha başarısız sonuçlar doğurarak dengesiz sonuçlar elde edilmiştir. Bu durum bazı yüzlerde daha iyi genelleme yapabilirken diğerlerine yetersiz kaldığını göstermiştir.

Daha fazla öz yüz kullanılması, eğitime daha fazla yüz eklenmesi ile veri setinin genişlemesi gibi öneriler yüzlerin tekrar oluşturulmasında faydalı olabilir.