

Praca Domowa 6

Bartosz Siński

```
In [19]: from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
from sklearn.datasets import fetch_olivetti_faces
np.set_seed = 42
```

Załadowanie danych i rysowanie obrazów

```
In [3]: faces, _ = fetch_olivetti_faces(return_X_y=True, shuffle=True,
                                     random_state=1613)
```

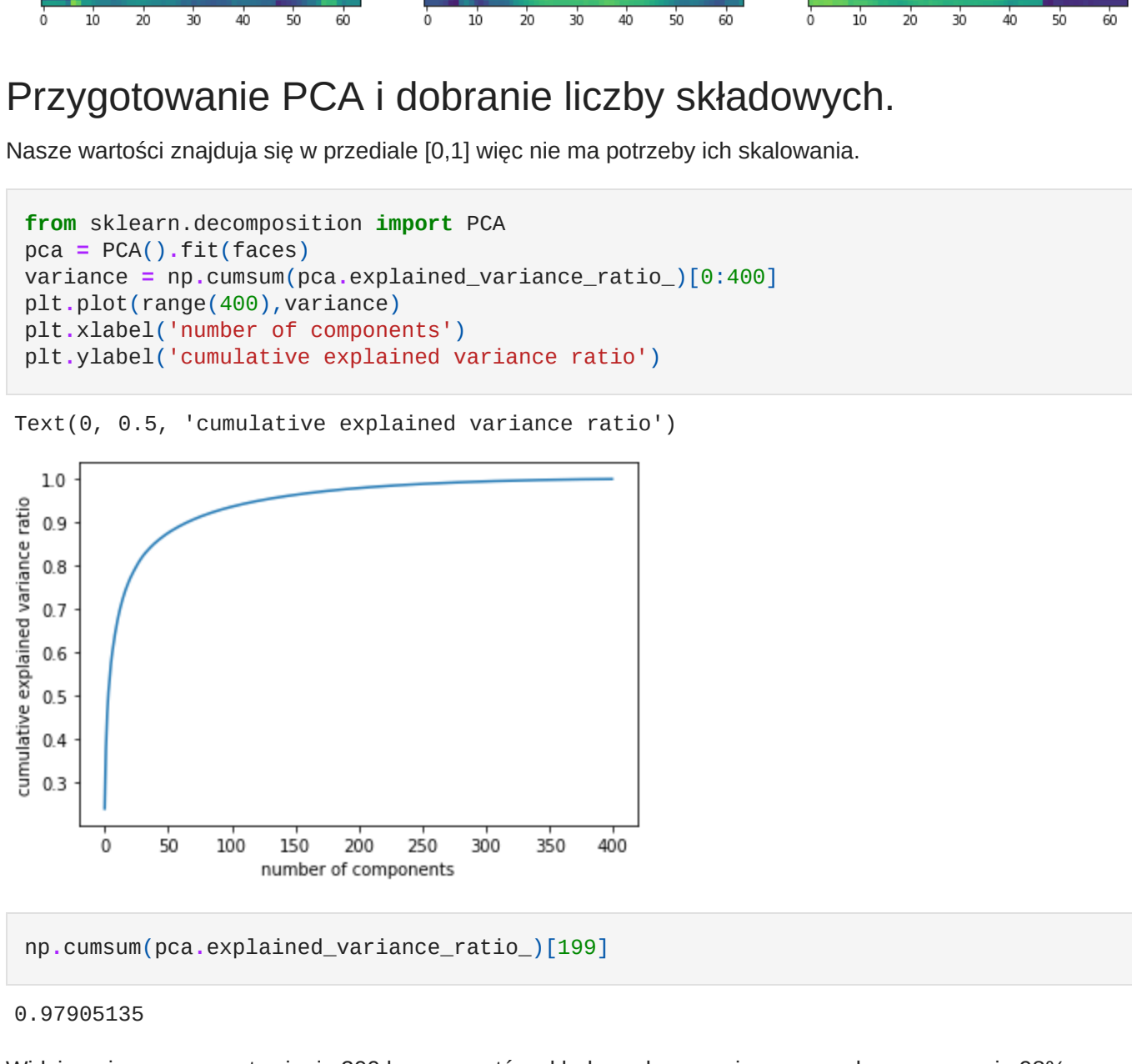
downloading Olivetti faces from https://ndownloader.figshare.com/files/5976027 to C:\Users\komp\Scikit_learn_data

In [71]: faces.shape

Out[71]: (400, 4096)

```
In [113]: images_id = [355,145,267,289,375,1]
imshape = (64,64)
```

```
In [171]: fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



Przygotowanie PCA i dobranie liczby składowych.

Nasze wartości znajdują się w przedziale [0,1] więc nie ma potrzeby ich skalowania.

```
In [72]: from sklearn.decomposition import PCA
pca = PCA().fit(faces)
variance = np.cumsum(pca.explained_variance_ratio_)[0:400]
plt.plot(range(400),variance)
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance ratio')
```

Out[72]: Text(0, 0.5, 'cumulative explained variance ratio')

```
In [59]: np.cumsum(pca.explained_variance_ratio_)[199]
```

Out[59]: 0.97905135

Widzimy, że przy pozostawieniu 200 komponentów składowych mamy jeszcze zachowane prawie 98% wariancji więc wybierzemy właśnie tę liczbę komponentów.

```
In [245]: pca200 = PCA(n_components=200).fit(faces)
faces_reduced = pca200.transform(faces)
print('Stopień kompresji: ' + str(faces.shape[1]/faces_reduced.shape[1]))
```

Stopień kompresji: 20.48

```
In [172]: faces_pca = pca200.inverse_transform(faces_reduced)
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_pca[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [247]: from sklearn.metrics import mean_squared_error
for i in range(6):
    print('Błąd rekonstrukcji w postaci RMSE dla obrazka ' + str(i+1) + " = " + str(mse))
print('Średni błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: ' + str(mean_squared_error(faces, faces_reduced)))
```

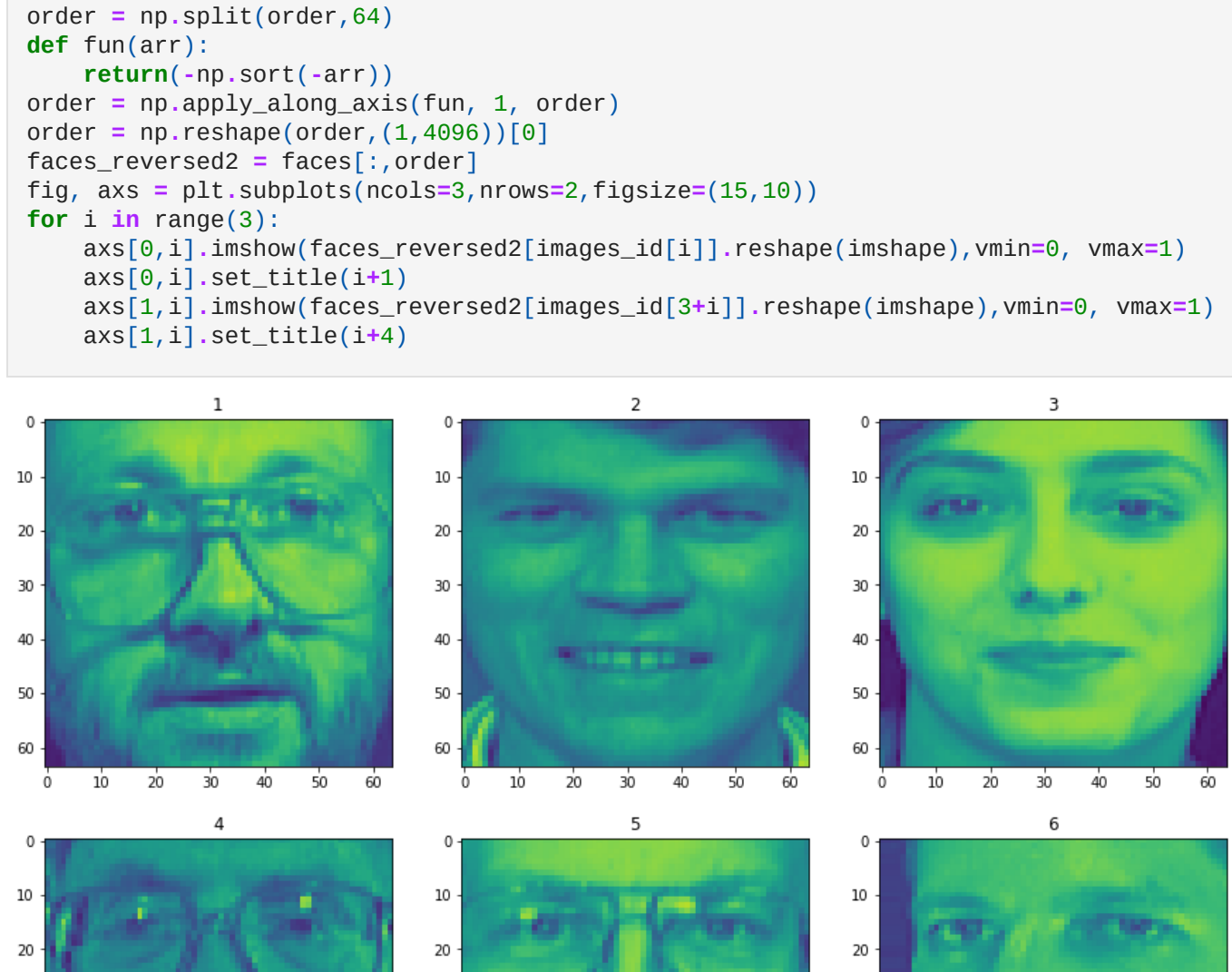
Błąd rekonstrukcji w postaci RMSE dla obrazka 1 = 0.00052746
Błąd rekonstrukcji w postaci RMSE dla obrazka 2 = 0.00039922152
Błąd rekonstrukcji w postaci RMSE dla obrazka 3 = 0.00040842284
Błąd rekonstrukcji w postaci RMSE dla obrazka 4 = 0.0005473286
Błąd rekonstrukcji w postaci RMSE dla obrazka 5 = 0.000515138
Błąd rekonstrukcji w postaci RMSE dla obrazka 6 = 0.0004932576
Średni błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: 0.00046721206

Przed wszystkim zdjęcia wydają się być mniej ostre. Dodatkowo wśród osób bez okularów pojawiły się obramowania wokół oczu. Najmniejszy błąd przy rekonstrukcji był dla zdjęcia 3, na którym znajduje się kobieta o gładkich rysach bez okularów, a największy dla zdjęcia 4 gdzie znajduje się starszy pan w okularach.

Zmodyfikowane obrazy na wcześniej wyuczonym modelu

Obrazy obrócone

```
In [173]: order = -np.sort(-np.arange(4096))
faces_reversed = faces[:,order]
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_reversed[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_reversed[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [248]: faces_r = pca200.transform(faces_reversed)
faces_r = pca200.inverse_transform(faces_r)
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_r[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_r[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [249]: print('Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: ' + str(mean_squared_error(faces, faces_r)))
```

Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: 0.00662068

Z PCA nauczonym na normalnych obrazach, odwrócone obrazy po rekonstrukcji w żaden sposób nie przypominają oryginałów. Widac że to jeszcze bardziej "dorysowywanie" okularów przy rekonstrukcji. Błędy RMSE są także ponad 10 razy większe niż poprzednio.

Obrazy odwrócone w poziomie

```
In [237]: order = np.arange(0, 4096)
order = np.split(order,64)
def fun(arr):
    return(-np.sort(-arr))
order = np.apply_along_axis(fun, 1, order)
order = np.reshape(order,(1, 4096))[0]
faces_reversed2 = faces[:,order]
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_reversed2[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_reversed2[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [250]: faces_r2 = pca200.transform(faces_reversed2)
faces_r2 = pca200.inverse_transform(faces_r2)
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_r2[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_r2[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



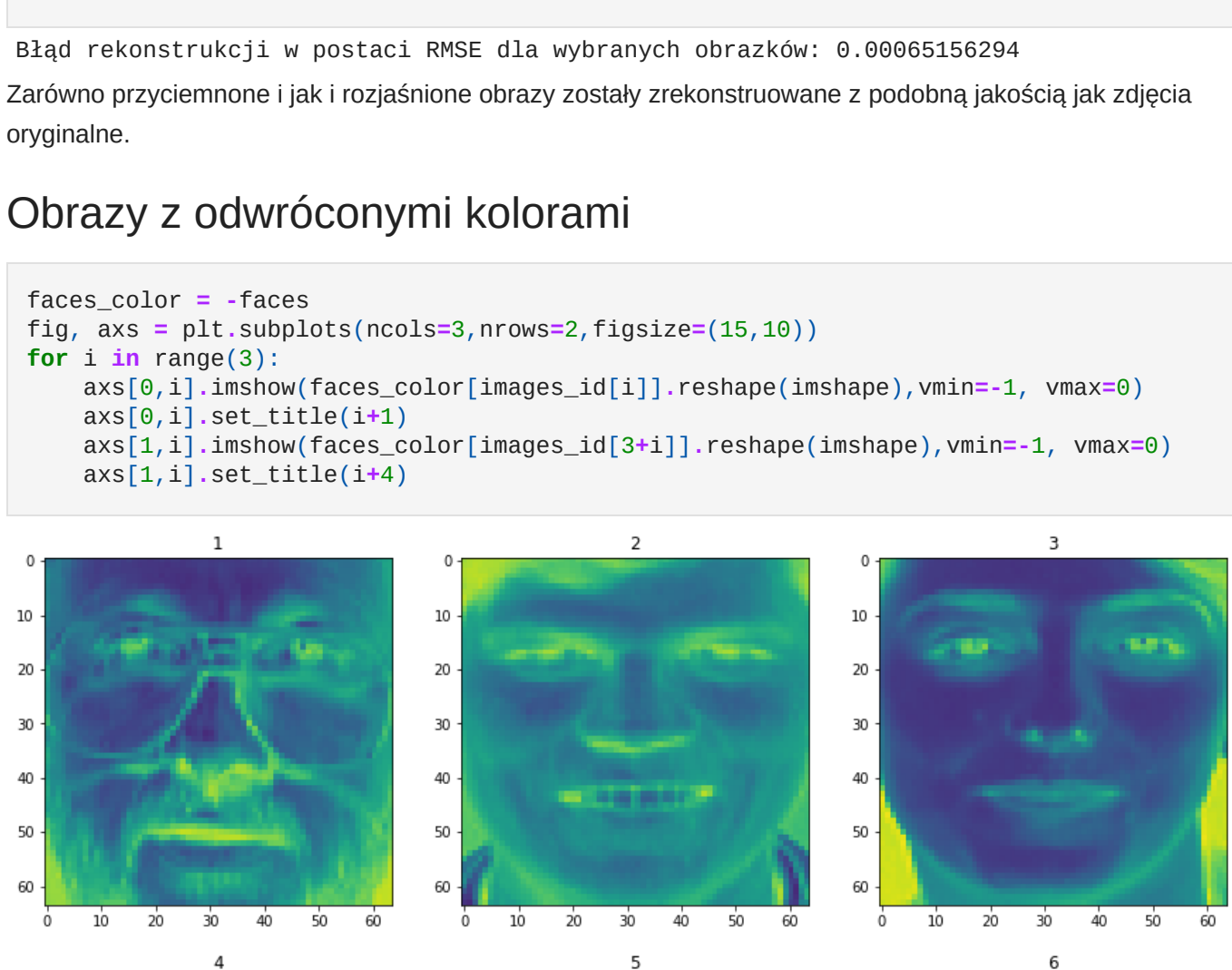
```
In [251]: print('Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: ' + str(mean_squared_error(faces, faces_r2)))
```

Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: 0.0029547855

Błędy rekonstrukcji są mniejsze niż w przypadku obrazów odwróconych o 180 stopni jednak nadal są ponad 5 razy większe niż w przypadku oryginalnych zdjęć. Obrazy mocno różnią się od oryginałów.

Obrazy przyciemnione

```
In [176]: faces_dimmed = faces * 0.2
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_dimmed[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_dimmed[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [252]: faces_d = pca200.transform(faces_dimmed)
faces_d = pca200.inverse_transform(faces_d)
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_d[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_d[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [253]: print('Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: ' + str(mean_squared_error(faces, faces_d)))
```

Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: 0.0064626674

Obrazy rozjaśnione

```
In [254]: faces_bright = faces * 0.2
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_bright[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_bright[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [255]: faces_b = pca200.transform(faces_bright)
faces_b = pca200.inverse_transform(faces_b)
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_b[images_id[i]].reshape(imshape),vmin=0, vmax=1)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_b[images_id[3+i]].reshape(imshape),vmin=0, vmax=1)
    axs[1,i].set_title(4+i)
```



```
In [256]: print('Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: ' + str(mean_squared_error(faces, faces_b)))
```

Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: 0.00065156294

Również przyciemnione i jask i rozjaśnione obrazy zostały zrekonstruowane z podobną jakością jak zdjęcia oryginalne.

Obrazy z odwróconymi kolorami

```
In [190]: faces_color = -faces
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_color[images_id[i]].reshape(imshape),vmin=-1, vmax=0)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_color[images_id[3+i]].reshape(imshape),vmin=-1, vmax=0)
    axs[1,i].set_title(4+i)
```



```
In [257]: faces_c = pca200.transform(faces_color)
faces_c = pca200.inverse_transform(faces_c)
fig, axs = plt.subplots(ncols=3,rows=2,figsize=(15,10))
for i in range(3):
    axs[0,i].imshow(faces_c[images_id[i]].reshape(imshape),vmin=-1, vmax=0)
    axs[0,i].set_title(i+1)
    axs[1,i].imshow(faces_c[images_id[3+i]].reshape(imshape),vmin=-1, vmax=0)
    axs[1,i].set_title(4+i)
```



```
In [258]: print('Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: ' + str(mean_squared_error(faces, faces_c)))
```

Błąd rekonstrukcji w postaci RMSE dla wybranych obrazków: 0.0052294135

Odwrócenie kolorów zdecydowanie obniżyło jakość rekonstrukcji obrazów.

Wnioski

Rezultaty transformacji odwrotnej obrazów z 200 komponentów do 4096 okazały się bardzo zadowalające. W przypadku większości obrazów oryginalnych, do których fitowaliśmy PCA, ciężko było znaleźć różnice pomiędzy oryginałami, a zrekonstruowanymi obrazami. Zaskakujące okazało się to, że obrazy pociemnione i rozjaśnione zostały zrekonstruowane prawie tak samo dobrze jak oryginały. W przypadku bardziej zniekształcających obraz transformacji (np. obrót obrazu o 180 stopni) wyniki rekonstrukcji były zdecydowanie gorsze. Nasza metoda odtwarzania była wrażliwa na okulary i na większości obrazów było dorysowywanie. PCA mogłoby być posłużyc do określenia w prosty sposób czy wśród naszych obrazów są zdjęcia ludzi w okularach.