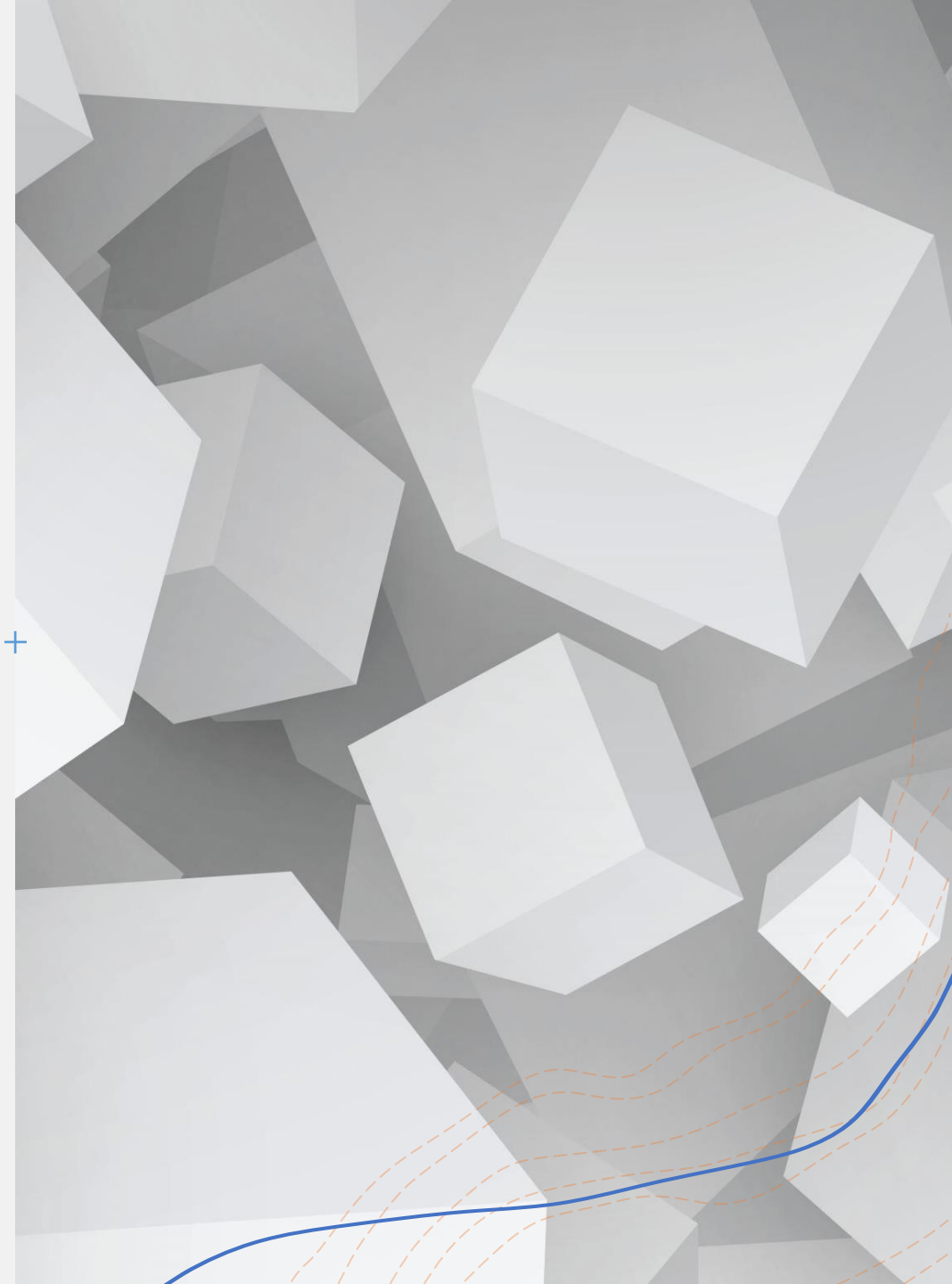
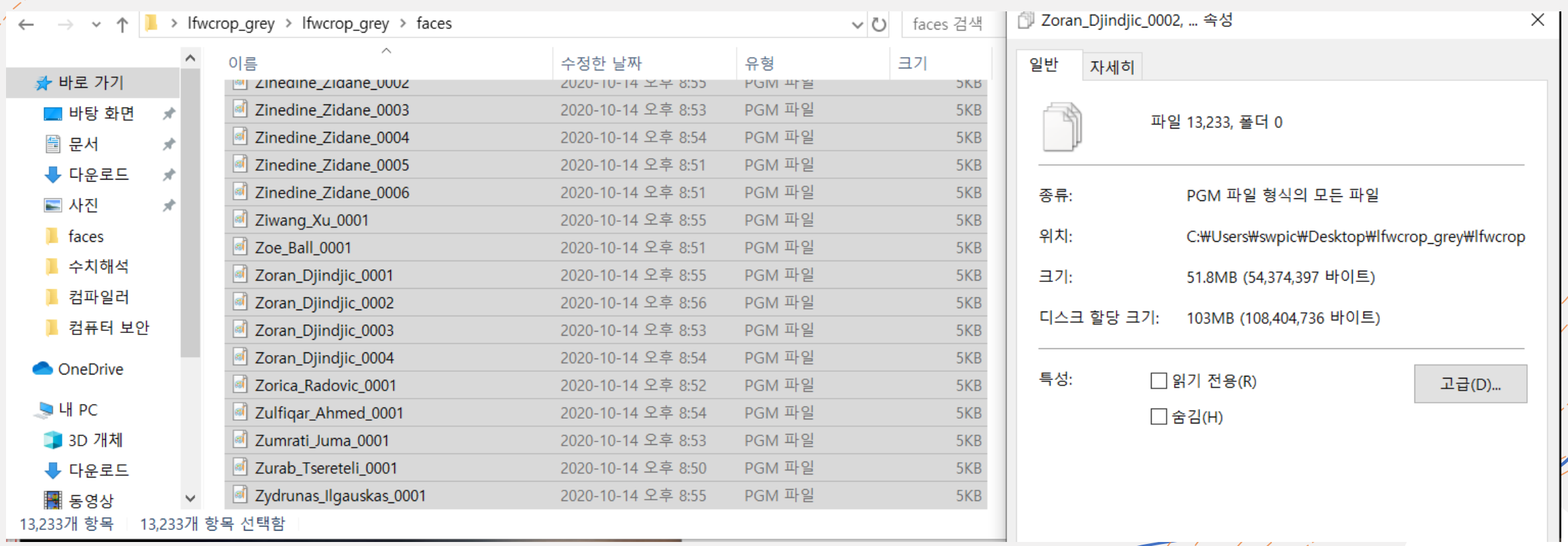
수치해석
Project 1

2018008613 안상욱



1. Collect face image

+ 구글링을 통해 64 X 64 픽셀 4096 – D 사진을 Dimension인 4096보다 많은 13233개를 구했습니다. 이 데이터를 가지고 프로젝트를 진행했습니다.



2. Construct Data Matrix, A

- 먼저 13323개의 사진이 있는 폴더 안의 모든 face image를 하나씩 읽어왔습니다. 그러면 n이라는 변수에 64 X 64의 2차원 배열이 담기게 되는데 ravel 함수를 통해 1 X 4096 행렬로 바꾸어서 13323개의 데이터를 cv_img라는 배열에 append 시켜주면 cv_img 배열은 13323 X 4096 크기를 갖는 배열이 됩니다.

```
import os
import cv2
import glob
import numpy as np

path = glob.glob("lfwcrop_grey/lfwcrop_grey/faces/*.pgm")
cv_img=[]
for img in path:
    n=cv2.imread(img, 0)
    n=n.ravel()
    cv_img.append(n)
```

2. Construct Data Matrix, A

+ 그 이후 열 별로 평균을 구한 값을 담은 1 X 4096 크기의 m이라는 배열을 만들고, 전체 데이터 값에서 이 평균 값을 빼 준 새로운 cv_img 배열을 만듭니다. 그리고 이 cv_img 배열을 Transpose 시킨 4096 X 13323 크기를 갖는 test라는 이름을 가진 A matrix를 생성합니다.

```
m = []
for j in range(len(cv_img[0])):
    colsum = 0
    for i in range(len(cv_img)):
        colsum += cv_img[i][j]
    colsum /= len(cv_img)
    m.append(colsum)
for i in range(len(cv_img)):
    for j in range(len(cv_img[0])):
        cv_img[i][j] -= m[j]

test = np.array(cv_img).T
```

3. Apply SVD

+ 앞에서 만든 A 배열을 통해 SVD 분할하여 U, s, VT에 각각의 분할된 행렬의 값을 담습니다.
+ 이 때, eigen vector는 30개를 사용했습니다.
+ U 행렬에서 eigen vector를 편리하게 사용하기 위해 U를 Transpose시킨 행렬 c를 만들었습니다.
+ 즉, c[0]가 c0 eigen vector, c[1]이 c1 eigen vector …… 로 c[29]까지를 만들어 사용할 수 있도록 했습니다.

```
test = np.array(cv_img).T
U, s, VT = np.linalg.svd(test)

c = U.T
eigen_num = 30
```

Mean vector

+ 다음 코드처럼 mean vector를 64 x 64 array로 만든 뒤, matplotlib API를 이용해 그림을 출력해본 결과 다음과 같은 얼굴이 나옴을 알 수 있었습니다.

```
meanvector = np.array(m)
meanvector = meanvector.reshape(64, 64)
plt.imshow(meanvector, cmap='gray')
plt.show()
```

Eigen face

+ 다음 코드처럼 30개의 eigen vector를 정해서 64 x 64 matrix로 만든 뒤 이미지를 출력해 보았습니다. 출력한 30개의 이미지는 뒷장에 순차적으로 나타냈습니다.

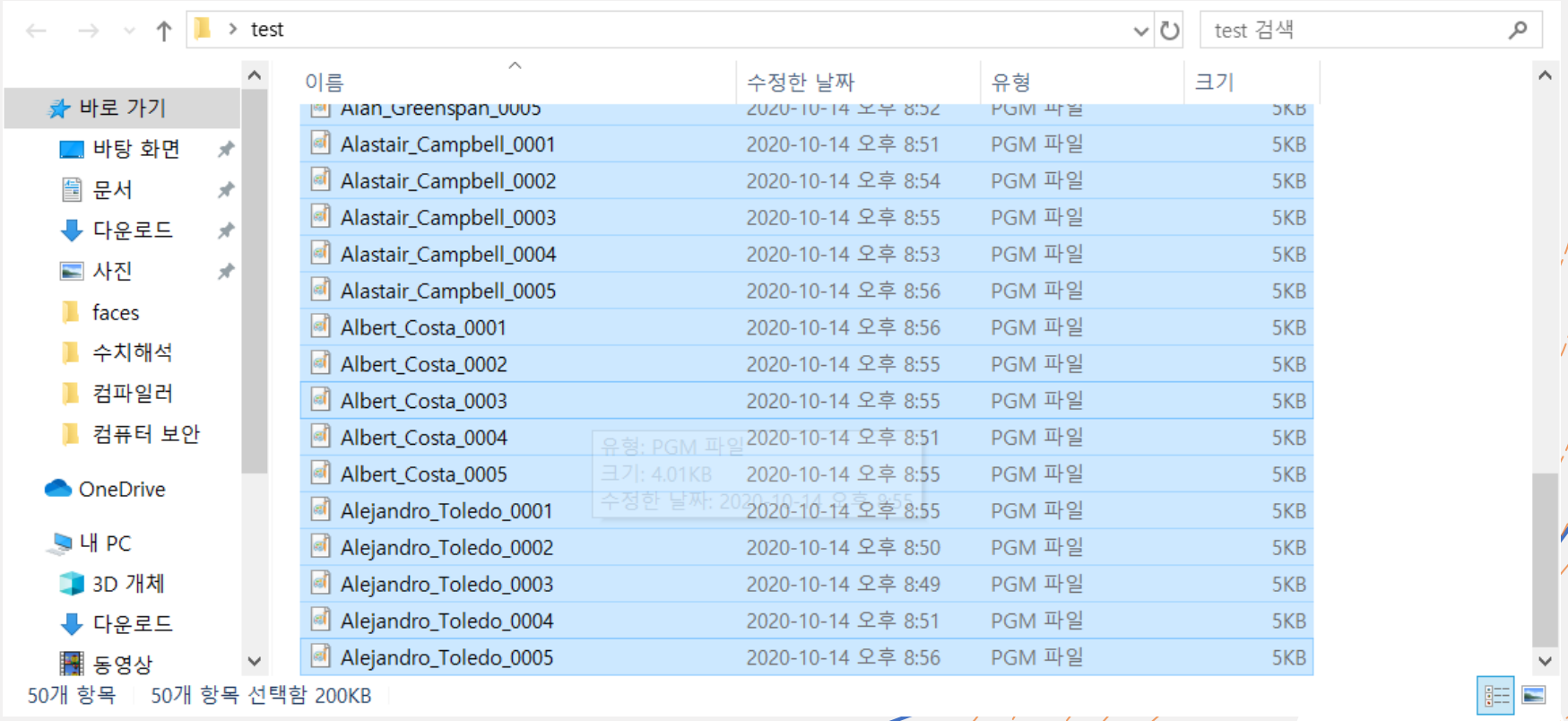
```
c = U.T
eigen_num = 30

meanvector = np.array(m)
meanvector = meanvector.reshape(64, 64)
plt.imshow(meanvector, cmap='gray')
plt.show()
for i in range(eigen_num):
    eigenface = np.array(c[i])
    eigenface = eigenface.reshape(64, 64)
    plt.imshow(eigenface, cmap='gray')
    plt.show()
```

Eigenface

4. Test face recognition

+ 10명의 사람의 사진을 각각 5개씩 뽑아 50개의 사진을 저장한 test라는 폴더를 만들었습니다.



5. Generate face image using eigenfaces

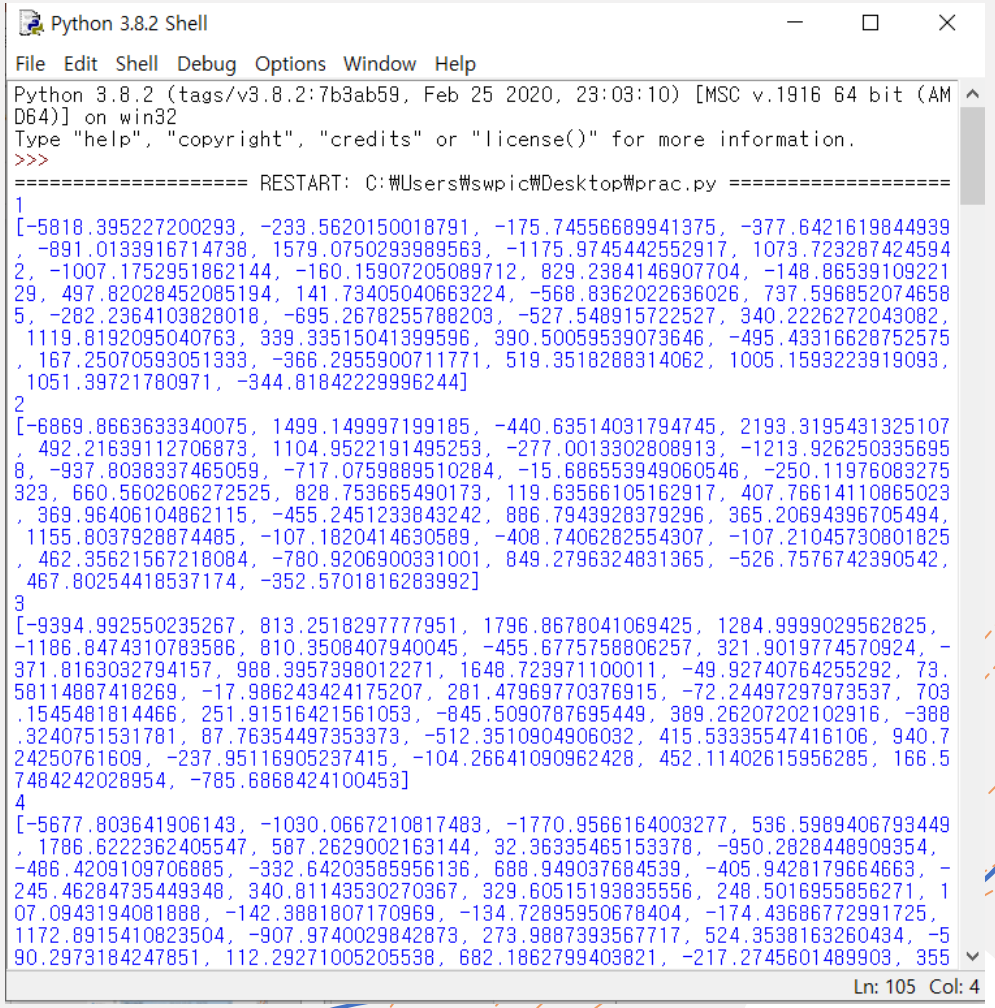
+ 그 후 Test라는 폴더에서 50장의 사진을 하나씩 불러와서 n이라는 1 X 4096 크기를 갖는 행렬을 만들었습니다. 아까 평균을 구했던 m이라는 행렬 값을 빼서 n에 저장한 뒤 이 값을 각각 c0, c1, …. , c29 와 inner product 시키면 이 각각의 c0, c1, … c29 값이 나오므로 이 30개의 값들을 c_arr에 저장한 뒤 이를 50명의 사람 각각에 대해 출력해 보았습니다.

```
c = U.T
eigen_num = 30

path = glob.glob("test/*.pgm")
num = 0
for img in path:
    n=cv2.imread(img, 0)
    n=n.ravel()
    for i in range(4096):
        n[i] -= m[i]
    c_arr = []
    for i in range(eigen_num):
        x = c[i] @ n
        c_arr.append(x)
    print(num + 1)
    print(c_arr)
    num = (num + 1) % 5
```

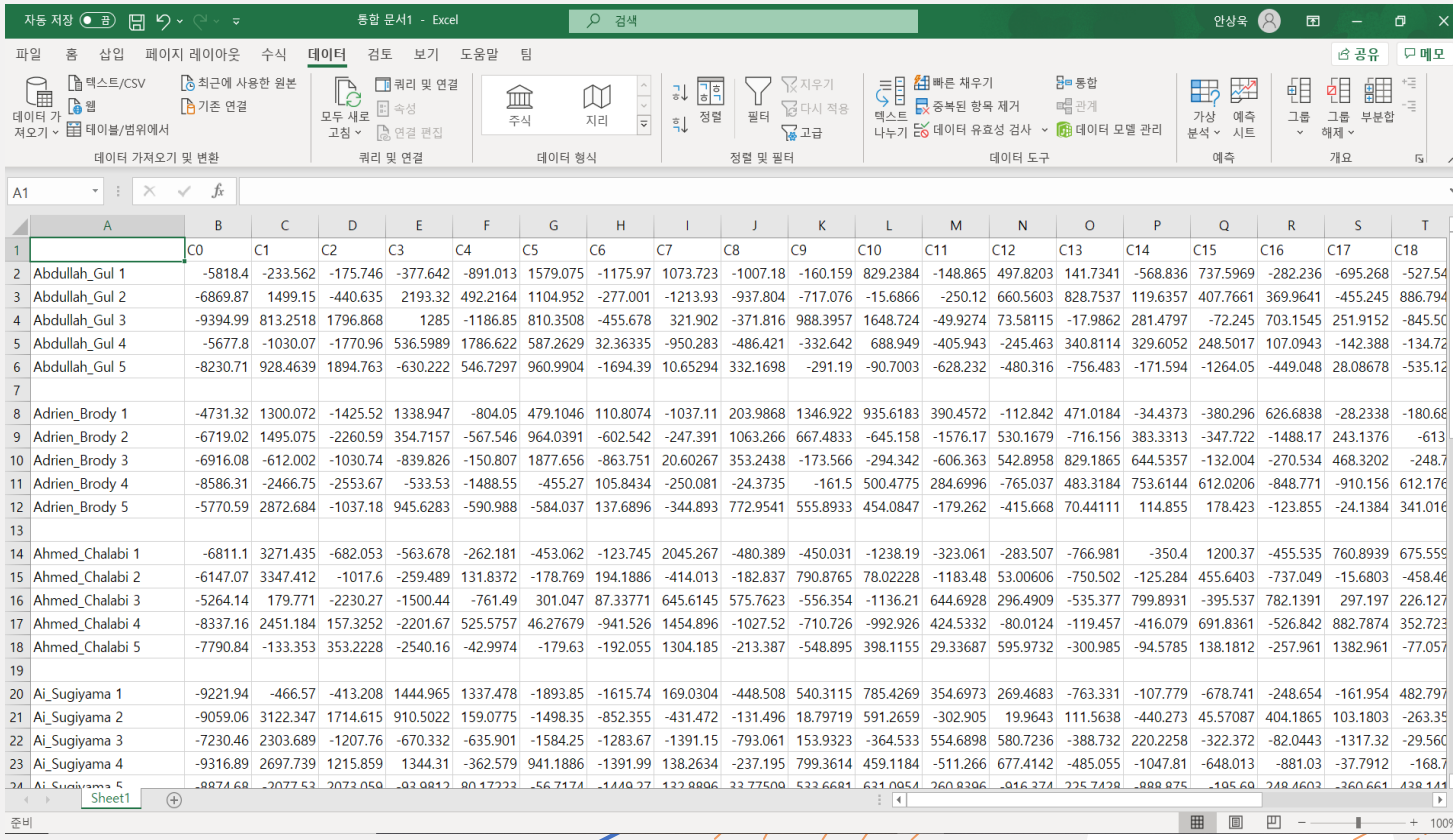
5. Generate face image using eigenfaces

+ 출력 결과입니다.



5. Generate face image using eigenfaces

+ 다음 그림처럼 엑셀로 정리했고, 제출 시 엑셀을 같이 첨부해서 제출하겠습니다.
+ 사람마다 결과를 분석해 보았습니다.



통합 문서1 - Excel

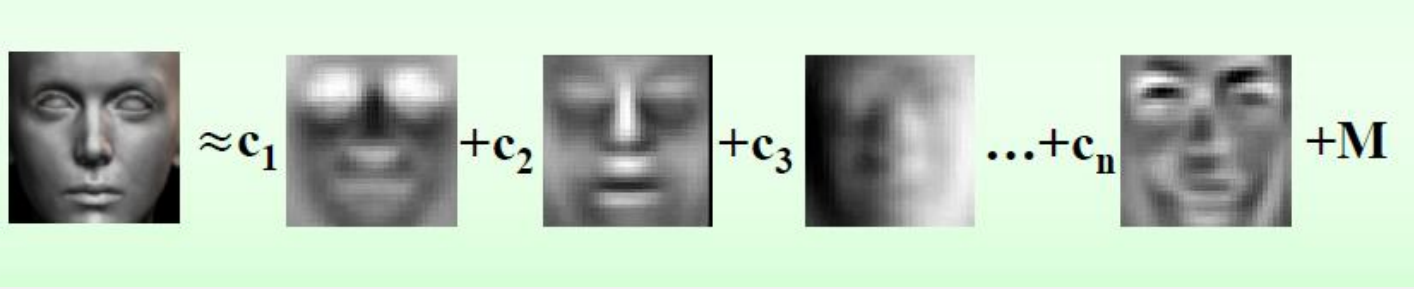
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
2	Abdullah_Gul 1	-5818.4	-233.562	-175.746	-377.642	-891.013	1579.075	-1175.97	1073.723	-1007.18	-160.159	829.2384	-148.865	497.8203	141.7341	-568.836	737.5969	-282.236	-695.268	-527.54
3	Abdullah_Gul 2	-6869.87	1499.15	-440.635	2193.32	492.2164	1104.952	-277.001	-1213.93	-937.804	-717.076	-15.6866	-250.12	660.5603	828.7537	119.6357	407.7661	369.9641	-455.245	886.794
4	Abdullah_Gul 3	-9394.99	813.2518	1796.868	1285	-1186.85	810.3508	-455.678	321.902	-371.816	988.3957	1648.724	-49.9274	73.58115	-17.9862	281.4797	-72.245	703.1545	251.9152	-845.5
5	Abdullah_Gul 4	-5677.8	-1030.07	-1770.96	536.5989	1786.622	587.2629	32.36335	-950.283	-486.421	-332.642	688.949	-405.943	-245.463	340.8114	329.6052	248.5017	107.0943	-142.388	-134.7
6	Abdullah_Gul 5	-8230.71	928.4639	1894.763	-630.222	546.7297	960.9904	-1694.39	10.65294	332.1698	-291.19	-90.7003	-628.232	-480.316	-756.483	-171.594	-1264.05	-449.048	28.08678	-535.1
7																				
8	Adrien_Brody 1	-4731.32	1300.072	-1425.52	1338.947	-804.05	479.1046	110.8074	-1037.11	203.9868	1346.922	935.6183	390.4572	-112.842	471.0184	-34.4373	-380.296	626.6838	-28.2338	-180.6
9	Adrien_Brody 2	-6719.02	1495.075	-2260.59	354.7157	-567.546	964.0391	-602.542	-247.391	1063.266	667.4833	-645.158	-1576.17	530.1679	-716.156	383.3313	-347.722	-1488.17	243.1376	-613
10	Adrien_Brody 3	-6916.08	-612.002	-1030.74	-839.826	-150.807	1877.656	-863.751	20.60267	353.2438	-173.566	-294.342	-606.363	542.8958	829.1865	644.5357	-132.004	-270.534	468.3202	-248.7
11	Adrien_Brody 4	-8586.31	-2466.75	-2553.67	-533.53	-1488.55	-455.27	105.8434	-250.081	-24.3735	-161.5	500.4775	284.6996	-765.037	483.3184	753.6144	612.0206	-848.771	-910.156	612.176
12	Adrien_Brody 5	-5770.59	2872.684	-1037.18	945.6283	-590.988	-584.037	137.6896	-344.893	772.9541	555.8933	454.0847	-179.262	-415.668	70.44111	114.855	178.423	-123.855	-24.1384	341.016
13																				
14	Ahmed_Chalabi 1	-6811.1	3271.435	-682.053	-563.678	-262.181	-453.062	-123.745	2045.267	-480.389	-450.031	-1238.19	-323.061	-283.507	-766.981	-350.4	1200.37	-455.535	760.8939	675.559
15	Ahmed_Chalabi 2	-6147.07	3347.412	-1017.6	-259.489	131.8372	-178.769	194.1886	-414.013	-182.837	790.8765	78.02228	-1183.48	53.00606	-750.502	-125.284	455.6403	-737.049	-15.6803	-458.46
16	Ahmed_Chalabi 3	-5264.14	179.771	-2230.27	-1500.44	-761.49	301.047	87.33771	645.6145	575.7623	-556.354	-1136.21	644.6928	296.4909	-535.377	799.8931	-395.537	782.1391	297.197	226.127
17	Ahmed_Chalabi 4	-8337.16	2451.184	157.3252	-2201.67	525.5757	46.27679	-941.526	1454.896	-1027.52	-710.726	-992.926	424.5332	-80.0124	-119.457	-416.079	691.8361	-526.842	882.7874	352.723
18	Ahmed_Chalabi 5	-7790.84	-133.353	353.2228	-2540.16	-42.9974	-179.63	-192.055	1304.185	-213.387	-548.895	398.1155	29.33687	595.9732	-300.985	-94.5785	138.1812	-257.961	1382.961	-77.057
19																				
20	Ai_Sugiyama 1	-9221.94	-466.57	-413.208	1444.965	1337.478	-1893.85	-1615.74	169.0304	-448.508	540.3115	785.4269	354.6973	269.4683	-763.331	-107.779	-678.741	-248.654	-161.954	482.797
21	Ai_Sugiyama 2	-9059.06	3122.347	1714.615	910.5022	159.0775	-1498.35	-852.355	-431.472	-131.496	18.79719	591.2659	-302.905	19.9643	111.5638	-440.273	45.57087	404.1865	103.1803	-263.35
22	Ai_Sugiyama 3	-7230.46	2303.689	-1207.76	-670.332	-635.901	-1584.25	-1283.67	-1391.15	-793.061	153.9323	-364.533	554.6898	580.7236	-388.732	220.2258	-322.372	-82.0443	-1317.32	-29.560
23	Ai_Sugiyama 4	-9316.89	2697.739	1215.859	1344.31	-362.579	941.1886	-1391.99	138.2634	-237.195	799.3614	459.1184	-511.266	677.4142	-485.055	-1047.81	-648.013	-881.03	-37.7912	-168.7
24	Ai_Sugiyama 5	-8874.68	-2077.53	2073.059	-93.9812	80.17223	-56.7174	-1449.27	132.8896	33.77509	533.6681	631.0954	260.8396	-916.374	225.7428	-888.875	-195.69	248.4603	-360.661	438.141

Sheet1

준비

5. Generate face image using eigenfaces

+ 각 사람마다 c0~c29까지의 계수를 구한 뒤 이를 이용해서 다시 그 사람의 face를 복원하고 이를 이미지로 출력해서 원본 사진과 얼마나 유사한가 비교해보았습니다.




```
for img in path:
    n=cv2.imread(img, 0)
    n=n.ravel()
    for i in range(4096):
        n[i] -= m[i]
    c_arr = []
    for i in range(eigen_num):
        x = c[i] @ n
        c_arr.append(x)
    print(num + 1)
    print(c_arr)
    num = (num + 1) % 5
    facevector=c[0] * c_arr[0]
    for j in range(1,30):
        facevector += c[j] * c_arr[j]
    facevector += m
    facevector = np.array(facevector)
    facevector = facevector.reshape(64, 64)
    plt.imshow(facevector, cmap='gray')
    plt.show()
```

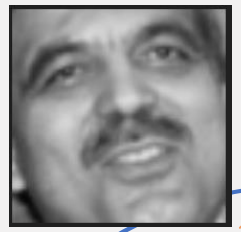
1. Abdullah Gul

- 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
- 2, 3번 사진이 비슷하게 찍혔고 1, 4, 5번 사진이 비슷하게 찍힌 것을 확인할 수 있습니다.
- 비교 결과 3 4 6 10 12 14 16 19 22 24 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
Abdullah_Gul 1	-5818.4	-233.562	-175.746	-377.642	-891.013	1579.075	-1175.97	1073.723	-1007.18	-160.159	829.2384	-148.865	497.8203	141.7341	-568.836	737.5969	-282.236	-695.268	-527.54
Abdullah_Gul 2	-6869.87	1499.15	-440.635	2193.32	492.2164	1104.952	-277.001	-1213.93	-937.804	-717.076	-15.6866	-250.12	660.5603	828.7537	119.6357	407.7661	369.9641	-455.245	886.794
Abdullah_Gul 3	-9394.99	813.2518	1796.868	1285	-1186.85	810.3508	-455.678	321.902	-371.816	988.3957	1648.724	-49.9274	73.58115	-17.9862	281.4797	-72.245	703.1545	251.9152	-845.50
Abdullah_Gul 4	-5677.8	-1030.07	-1770.96	536.5989	1786.622	587.2629	32.36335	-950.283	-486.421	-332.642	688.949	-405.943	-245.463	340.8114	329.6052	248.5017	107.0943	-142.388	-134.72
Abdullah_Gul 5	-8230.71	928.4639	1894.763	-630.222	546.7297	960.9904	-1694.39	10.65294	332.1698	-291.19	-90.7003	-628.232	-480.316	-756.483	-171.594	-1264.05	-449.048	28.08678	-535.12

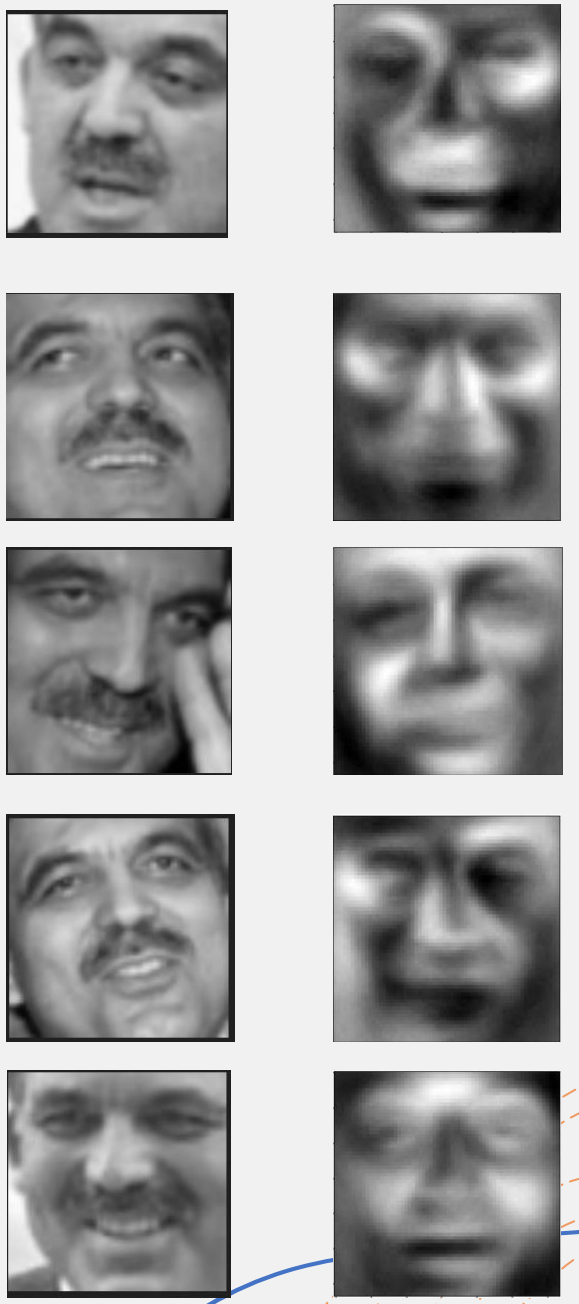
C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29
340.2226	1119.819	339.3352	390.5006	-495.433	167.2507	-366.296	519.3518	1005.159	1051.397	-344.818
365.2069	1155.804	-107.182	-408.741	-107.21	462.3562	-780.921	849.2796	-526.758	467.8025	-352.57
389.2621	-388.324	87.76354	-512.351	415.5334	940.7243	-237.951	-104.266	452.114	166.5748	-785.687
-174.437	1172.892	-907.974	273.9887	524.3538	-590.297	112.2927	682.1863	-217.275	355.7749	-4.23683
-130.735	1298.296	-651.033	735.954	439.4711	143.7602	-247.71	74.396	0.051064	-616.091	866.9985




1. Abdullah Gul

+ 복원 결과 다음과 같은 결과를 가짐을 확인할 수 있었습니다.



2. Adrien Brody

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 2, 3번 사진이 비슷하게 찍혔고 1, 4, 5번 사진이 비슷하게 찍힌 것을 확인할 수 있습니다.
+ 비교 결과 3 7 9 13 16 21 25 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Adrien_Brody 1	-4731.32	1300.072	-1425.52	1338.947	-804.05	479.1046	110.8074	-1037.11	203.9868	1346.922	935.6183	390.4572	-112.842	471.0184	-34.4373	-380.296	626.6838	-28.2338	-180.68
Adrien_Brody 2	-6719.02	1495.075	-2260.59	354.7157	-567.546	964.0391	-602.542	-247.391	1063.266	667.4833	-645.158	-1576.17	530.1679	-716.156	383.3313	-347.722	-1488.17	243.1376	-613
Adrien_Brody 3	-6916.08	-612.002	-1030.74	-839.826	-150.807	1877.656	-863.751	20.60267	353.2438	-173.566	-294.342	-606.363	542.8958	829.1865	644.5357	-132.004	-270.534	468.3202	-248.7
Adrien_Brody 4	-8586.31	-2466.75	-2553.67	-533.53	-1488.55	-455.27	105.8434	-250.081	-24.3735	-161.5	500.4775	284.6996	-765.037	483.3184	753.6144	612.0206	-848.771	-910.156	612.176
Adrien_Brody 5	-5770.59	2872.684	-1037.18	945.6283	-590.988	-584.037	137.6896	-344.893	772.9541	555.8933	454.0847	-179.262	-415.668	70.44111	114.855	178.423	-123.855	-24.1384	341.016

457.9367	-895.245	442.0808	159.0691	-433.579	-145.172	-494.878	-301.69	-178.351	-686.763	-67.7601
112.6513	70.29245	796.2889	-571.204	260.8787	-423.427	-44.0178	-53.8401	-58.2096	168.5869	373.1587
549.2986	242.1289	70.93392	376.7537	594.5782	-260.471	-356.623	-221.674	137.155	294.6523	105.3491
512.7454	544.6304	-594.426	167.4564	-463.853	-129.796	470.6502	-452.997	-247.034	-220.078	-222.23
-288.786	-491.253	452.2874	344.452	-63.349	405.2188	88.11288	21.65181	-892.634	302.2932	478.7281





2. Adrien Brody

+ 복원 결과 다음과 같은 결과를 가짐을 알 수 있었습니다.

3. Ahmed Chalabi

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 모든 사진이 유사하게 나온 것을 확인할 수 있었습니다.
+ 비교 결과 7 11 20 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Ahmed_Chalabi 1	-6811.1	3271.435	-682.053	-563.678	-262.181	-453.062	-123.745	2045.267	-480.389	-450.031	-1238.19	-323.061	-283.507	-766.981	-350.4	1200.37	-455.535	760.8939	675.559
Ahmed_Chalabi 2	-6147.07	3347.412	-1017.6	-259.489	131.8372	-178.769	194.1886	-414.013	-182.837	790.8765	78.02228	-1183.48	53.00606	-750.502	-125.284	455.6403	-737.049	-15.6803	-458.46
Ahmed_Chalabi 3	-5264.14	179.771	-2230.27	-1500.44	-761.49	301.047	87.33771	645.6145	575.7623	-556.354	-1136.21	644.6928	296.4909	-535.377	799.8931	-395.537	782.1391	297.197	226.127
Ahmed_Chalabi 4	-8337.16	2451.184	157.3252	-2201.67	525.5757	46.27679	-941.526	1454.896	-1027.52	-710.726	-992.926	424.5332	-80.0124	-119.457	-416.079	691.8361	-526.842	882.7874	352.723
Ahmed_Chalabi 5	-7790.84	-133.353	353.2228	-2540.16	-42.9974	-179.63	-192.055	1304.185	-213.387	-548.895	398.1155	29.33687	595.9732	-300.985	-94.5785	138.1812	-257.961	1382.961	-77.057

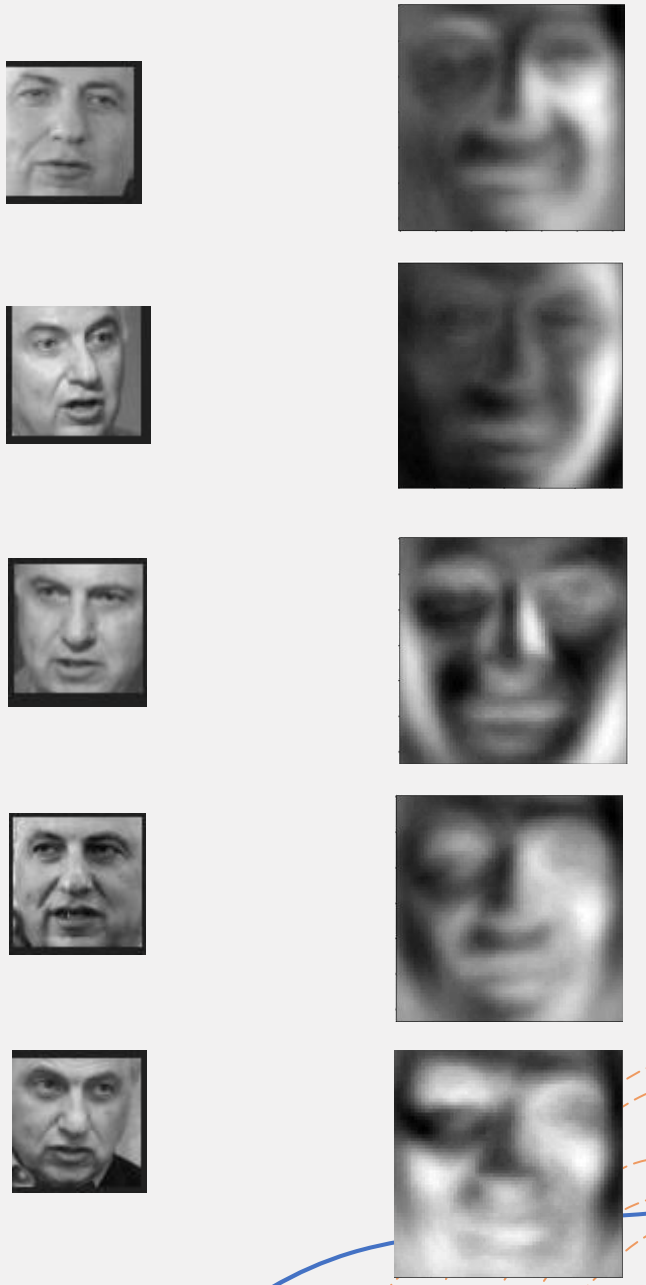
627.6047	279.4792	137.3003	361.5631	601.1205	274.1056	-126.44	34.04875	149.6372	-43.636	118.669
-1076.22	176.2305	300.4019	-148.145	-111.954	407.295	141.728	-421.909	23.06585	-42.0896	48.87779
486.575	1178.823	296.587	-520.786	-82.0643	-232.683	878.7759	-83.5795	468.9387	-44.4735	-289.896
511.4286	-827.176	307.3122	18.34865	166.1452	91.60022	16.86933	60.88436	-562.384	-233.419	486.9013
773.2469	-632.381	111.2111	134.4165	-227.623	245.3374	430.0766	128.2922	-247.391	207.6868	19.69419





3. Ahmed Chalabi

+ 복원 결과 다음과 같은 결과가 나옴을 알 수 있었습니다.



4. Ai Sugiyama

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 모든 사진이 유사하게 나온 것을 확인할 수 있었습니다.
+ 비교 결과 12 17 25 27 28 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Ai_Sugiyama 1	-9221.94	-466.57	-413.208	1444.965	1337.478	-1893.85	-1615.74	169.0304	-448.508	540.3115	785.4269	354.6973	269.4683	-763.331	-107.779	-678.741	-248.654	-161.954	482.797
Ai_Sugiyama 2	-9059.06	3122.347	1714.615	910.5022	159.0775	-1498.35	-852.355	-431.472	-131.496	18.79719	591.2659	-302.905	19.9643	111.5638	-440.273	45.57087	404.1865	103.1803	-263.35
Ai_Sugiyama 3	-7230.46	2303.689	-1207.76	-670.332	-635.901	-1584.25	-1283.67	-1391.15	-793.061	153.9323	-364.533	554.6898	580.7236	-388.732	220.2258	-322.372	-82.0443	-1317.32	-29.560
Ai_Sugiyama 4	-9316.89	2697.739	1215.859	1344.31	-362.579	941.1886	-1391.99	138.2634	-237.195	799.3614	459.1184	-511.266	677.4142	-485.055	-1047.81	-648.013	-881.03	-37.7912	-168.7
Ai_Sugiyama 5	-8874.68	-2077.53	2073.059	-93.9812	80.17223	-56.7174	-1449.27	132.8896	33.77509	533.6681	631.0954	260.8396	-916.374	225.7428	-888.875	-195.69	248.4603	-360.661	438.141

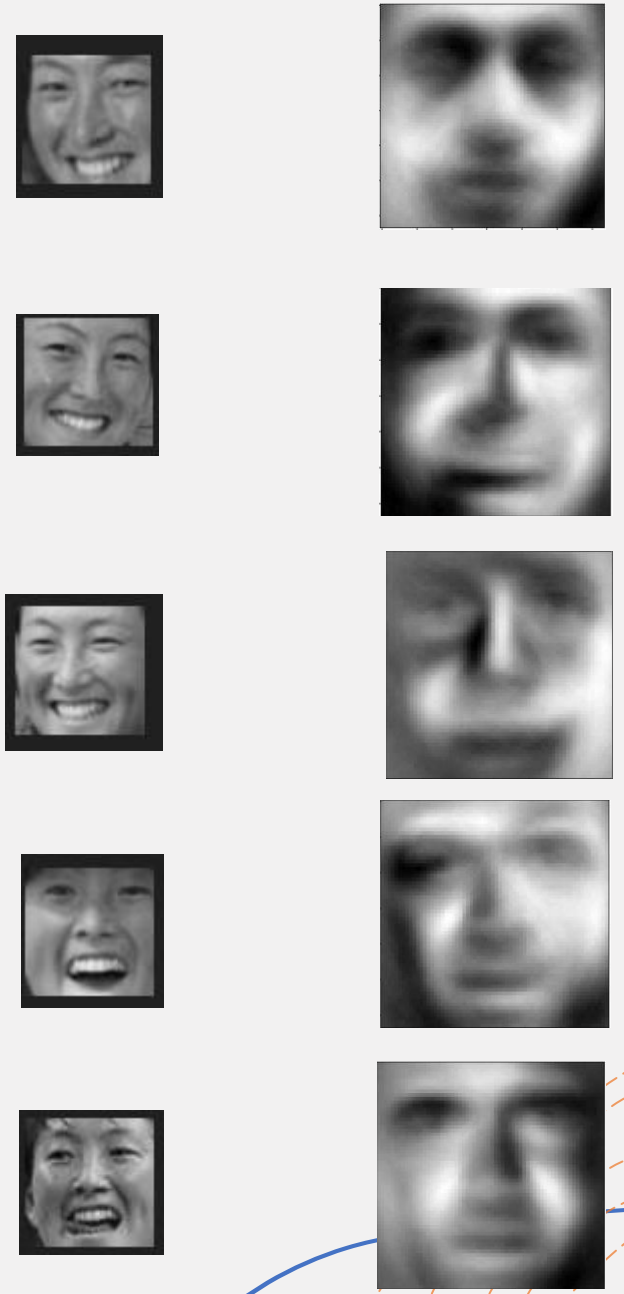
-565.069	252.4041	-35.4202	-117.642	222.0623	-211.307	-656.34	132.2989	-397.434	-301.158	326.5384
-560.356	537.529	431.6331	-973.667	981.1187	583.1491	204.0477	-199.153	837.1749	218.0605	-783.454
236.8466	-981.821	14.80377	532.2458	-130.719	75.78728	-2.6691	-9.72202	-241.482	519.6538	-700.655
-123.035	155.1809	-679.07	310.9606	-226.723	-432.524	-187.253	-1059.13	328.6176	-883.87	41.28899
-223.677	283.1913	-536.72	297.5118	435.7308	-487.34	131.6776	-911.832	-11.2738	-64.3275	124.8054





4. Ai Sugiyama

+ 복원 결과 다음과 같은 결과가 나옴을 알 수 있었습니다.



5. Al Gore

- 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
- 4, 5번 사진이 유사하게 나온 것을 확인할 수 있었습니다.
- 비교 결과 2 4 9 10 14 18 20 25 26 28 29 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Al_Gore 1	-8697.89	3087.257	-598.645	551.7912	708.1192	-267.823	-568.708	310.1984	1271.128	-658.923	1314.742	234.8741	-889.776	932.2865	52.14992	126.8653	-146.282	-397.032	104.365
Al_Gore 2	-11419	-280.298	114.5424	-629.368	-409.619	-1290.94	-104.563	-213.346	-262.374	902.9176	301.222	-587.466	-451.099	287.0495	-1113.84	197.2179	-31.6111	149.3019	-590.02
Al_Gore 3	-7391.36	2591.645	-1654.25	1068.12	1006.012	-434.998	-960.95	-558.255	404.5417	-932.499	-21.6585	573.377	824.0225	243.6681	-1268.01	162.9276	91.76987	45.12225	928.496
Al_Gore 4	-9756.27	3023.693	1421.81	-250.407	-900.624	28.93337	-236.846	508.3511	-480.192	522.8309	261.1595	3.934756	662.2859	-481.198	-794.227	-114.044	857.34	805.78	-1075.7
Al_Gore 5	-9846.85	-236.606	-246.981	362.8199	196.1924	-1403.94	-784.636	742.6839	-321.356	-274.723	-414.449	-22.7938	-187.368	122.4397	554.2698	-54.2104	250.3277	-164.372	166.697

-505.83	57.51355	197.244	-78.9954	-211.183	384.3872	20.44764	-139.034	516.4633	-328.029	203.3774
34.93057	489.7831	-695.97	873.6946	-616.163	-2.98561	291.3759	-1083.91	712.6494	-21.9761	-1098
-221.524	363.2944	526.8273	-14.4786	-410.371	-269.24	-202.167	148.4516	812.544	482.708	-280.348
-243.886	314.077	-189.788	726.6798	-523.153	-44.5116	-88.2965	-258.612	588.1445	208.3213	-692.122
299.3607	-837.979	-138.071	755.3559	-255.2	-396.21	-1026.66	404.7612	-281.361	-462.417	327.3998





5. Al Gore

+ 복원 결과 다음과 같은 결과가 나옴을 알 수 있었습니다.

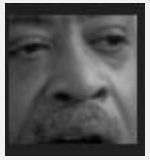
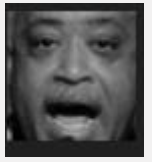
6. Al Sharpton

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 2, 3번 사진이 유사하게 나온 것을 확인할 수 있었습니다.
+ 비교 결과 2 4 5 9 13 15 20 21 28 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Al_Sharpton 1	-9627.87	-1624.26	3233.881	362.4621	-323.571	888.5918	615.4673	-859.022	391.7362	85.36997	1109.534	621.4217	-545.859	251.1172	15.64431	-271.368	32.01702	814.4489	-533.15
Al_Sharpton 2	-5511.46	1492.228	-1002.95	228.013	-1018.37	-1088.52	747.5643	-1410.3	986.3981	-275.971	-320.861	775.2568	-692.778	-654.284	-159.148	489.3363	-1027.69	176.3622	-266.06
Al_Sharpton 3	-4777.93	364.3947	648.821	-610.776	39.78079	1689.234	35.66631	-57.4393	711.5706	79.99848	492.5209	549.6752	-305.213	-151.381	-261.887	-1170.86	148.2924	172.539	-335.64
Al_Sharpton 4	-7773.61	2258.073	592.5809	1599.702	623.2629	312.3784	1095.799	-1531.55	-688.86	-1309.05	-226.43	487.7993	-1228.63	980.3081	-292.113	98.09995	509.9553	-54.9438	-900.0
Al_Sharpton 5	-6505.24	1186.667	-770.862	-675.447	-866.524	-219.104	201.8142	-478.007	241.402	79.89706	-271.494	-340.998	-44.1988	176.048	255.6514	-4.19768	66.12139	-370.843	-313.51

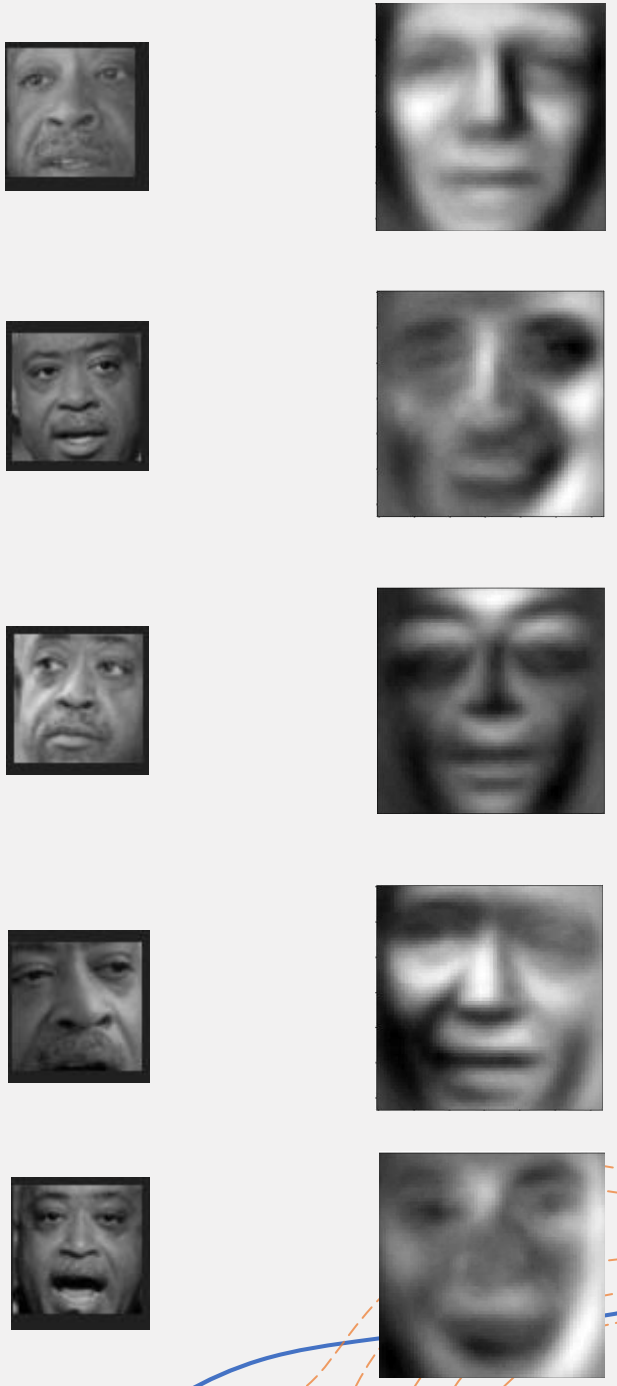
286.6923	-152.865	-67.5703	1294.866	-180.829	665.1401	313.5647	44.67572	-7.83292	196.7607	454.5153
-162.347	-617.737	-1172.64	-1121.12	-367.212	312.4271	200.5719	-491.072	-316.967	1074.585	-181.221
803.4811	727.2859	149.2848	-697.866	-249.847	-84.821	-119.645	791.2476	86.66568	-255.316	-249.049
564.0842	19.46164	815.2009	520.0588	109.198	8.770508	-137.485	-538.887	-386.21	162.2551	630.9764
-168.165	259.7684	-91.6497	-235.828	-295.137	-153.135	108.9102	153.888	76.09536	203.1035	33.45655



6. Al Sharpton

+ 복원 결과 다음과 같은 결과가 나옴을 알 수 있었습니다.



7. Alan Greenspan

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 모든 사진이 유사하게 나온 것을 확인할 수 있었습니다.
+ 비교 결과 12 13 16 21 26 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Alan_Greenspan 1	-12651	-276.948	557.9315	-43.1135	-44.1118	-584.691	-79.3884	462.1636	87.49516	-62.0029	-181.747	865.46	-567.558	-370.516	87.07219	-19.3071	-767.143	234.2338	314.982
Alan_Greenspan 2	-12630.1	-307.818	808.3192	307.7407	-79.4306	-186.283	468.9496	-172.487	412.6061	103.9872	-474.301	327.8527	131.2302	233.6662	278.729	44.61455	-20.2004	239.8039	-166.23
Alan_Greenspan 3	-10604.8	-1438.21	2085.608	177.4264	677.1679	313.856	280.2453	-181.695	-478.015	178.4413	-229.74	-339.338	-762.524	-643.772	688.0785	238.3356	-228.276	309.7984	-140.46
Alan_Greenspan 4	-12572.3	277.0282	761.847	-546.312	540.897	-391.326	183.4869	-294.307	305.7079	-749.363	791.9834	636.6959	664.808	285.4089	38.58885	-274.7	157.5375	656.2419	476.838
Alan_Greenspan 5	-12740.8	744.5046	382.7537	450.347	-155.692	-535.874	-108.135	-9.43918	-90.6444	301.494	-638.159	105.3643	-369.655	-222.639	416.3973	133.0279	231.1006	-143.885	-368.25

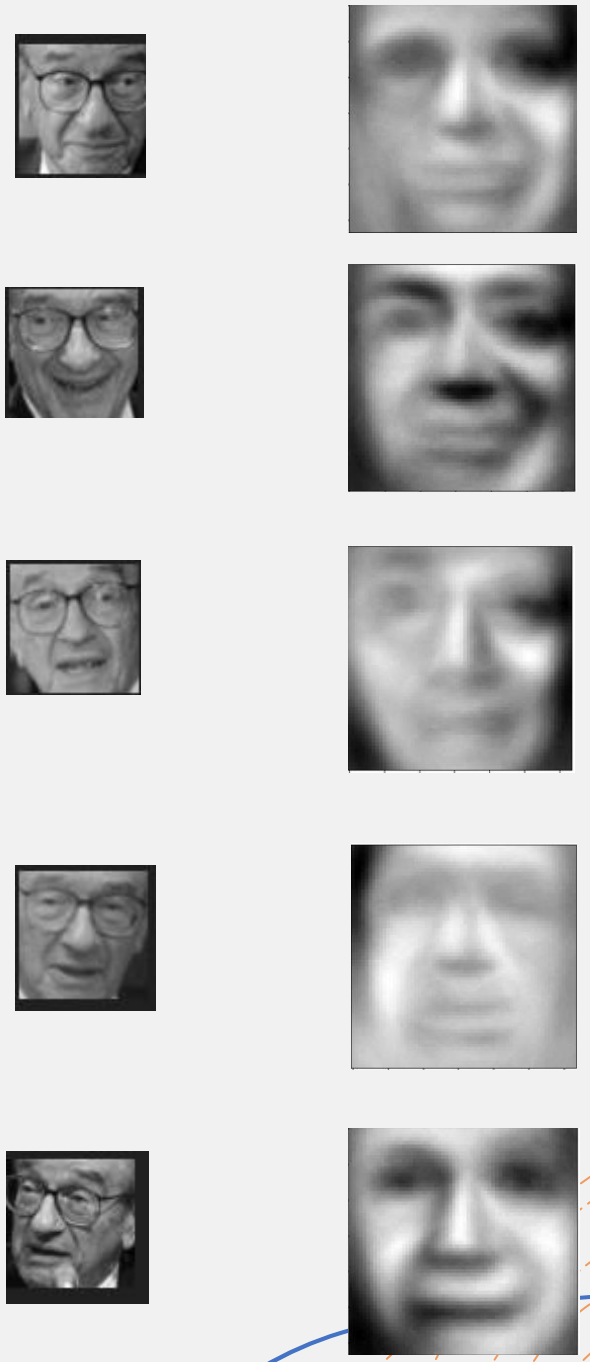
267.795	187.158	-48.0636	160.5372	-270.923	330.2601	-504.707	-625.454	-148.735	325.4271	-808.935
282.063	-266.47	424.7492	-699.831	-224.865	961.8223	118.6447	-110.653	-167.253	178.4924	-579.042
624.1017	-461.618	-692.032	-131.316	-112.269	437.5095	443.7056	-326.424	-121.106	230.6046	-217.4
-384.353	136.7433	396.5002	39.36336	200.5006	371.5895	-46.2087	20.76478	-172.116	304.9756	-116.494
113.0258	-131.427	113.6119	215.9783	302.1788	170.3174	-474.245	-233.16	-63.5453	38.72768	-429.811





7. Alan Greenspan

+ 복원 결과 다음과 같은 결과를 가짐을 확인할 수 있었습니다.



8. Alastair Campbell

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 1, 3번 사진이 유사하게 나온 것을 확인할 수 있었습니다.
+ 비교 결과 5 6 12 22 23 24 26 27 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Alastair_Campbell 1	-9157.26	294.6199	-1260.57	-939.704	611.5881	-1829.37	2022.112	149.8663	186.5304	366.0508	-1119.65	244.9473	-550.821	100.0535	-1315.66	-98.3023	-45.6226	518.6693	-1239
Alastair_Campbell 2	-3494.13	250.4553	-196.912	356.6752	-1086.36	202.9958	158.7727	-582.99	529.9278	-106.273	203.9807	771.311	-782.893	352.6096	450.7496	-250.05	324.3177	823.7641	292.002
Alastair_Campbell 3	-9024.27	2586.777	-104.807	-1415.8	304.7313	689.7556	-117.717	702.4468	-542.97	-220.154	173.3122	234.2618	929.5301	-229.624	-338.998	17.0058	-60.7265	689.4293	-633.1
Alastair_Campbell 4	-11366.4	90.59483	-902.153	689.358	-417.366	-634.495	530.788	577.3728	542.3188	285.9084	-936.827	-37.4103	-391.633	760.1797	-1328.8	507.3893	-1076.38	229.618	-594.79
Alastair_Campbell 5	-9095.93	-1255.68	199.6217	1120.599	-1564.46	-1455.91	-230.009	-868.992	1402.713	-452.897	-486.9	-303.197	-157.803	717.3408	31.95828	-63.5894	394.5725	0.608262	215.006

-328.842	329.0232	-15.1077	-450.417	540.1758	-568.408	101.0305	-1419.02	-418.344	220.2685	-219.157
412.4539	-492.66	359.0462	219.906	-33.4909	11.30401	-506.718	-105.566	34.23749	252.0346	176.3901
-189.099	66.53656	158.9651	1067.55	83.42883	203.5838	391.1989	134.2886	159.1417	361.2548	-260.07
-154.099	210.2114	-305.267	-99.738	-347.151	-1200.68	-494.088	-34.0385	515.4158	193.0243	129.5253
-263.625	-10.7864	-234.271	-1113.66	-60.6374	577.9434	-491.796	175.444	262.4664	-347.857	-560.799





8. Alastair Campbell

+ 복원 결과 다음과 같은 결과가 나옴을 알 수 있었습니다.

9. Albert Costa

- 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
- 모든 사진이 유사하게 나온 것을 확인할 수 있었습니다.
- 비교 결과 3 4 7 11 19 22 23 27 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Albert_Costa 1	-10094.9	-702.467	1552.466	-65.8392	1754.695	-522.182	728.279	130.91	-614.832	1281.691	-234.24	1247.913	479.9184	-199.826	541.4415	-509.142	379.2709	-715.541	-1179
Albert_Costa 2	-6094.19	-3335.56	1415.421	1276.84	-1547.63	-836.656	179.9157	-56.2666	694.3762	709.5111	-79.8654	81.41851	1160.895	-392.46	424.68	-727.036	672.8191	-76.4216	396.727
Albert_Costa 3	-4455.05	-129.336	-388.994	48.46054	-115.035	681.5753	273.7493	224.8568	-475.17	434.4433	-38.0515	153.9157	-859.846	48.21335	-465.315	56.13321	-365.966	146.6078	300.834
Albert_Costa 4	-3306	-349.682	-818.304	-752.137	433.7164	317.129	867.843	923.7927	176.6977	-448.539	839.821	442.863	-555.303	37.22952	-377.077	-136.458	-110.144	20.14081	-99.702
Albert_Costa 5	-3338.51	253.4267	-607.454	19.35121	-450.492	395.9102	-100.212	-360.705	98.64766	-376.259	-29.2594	299.5967	281.4734	37.66848	57.71375	436.8671	70.26	-152.168	258.007

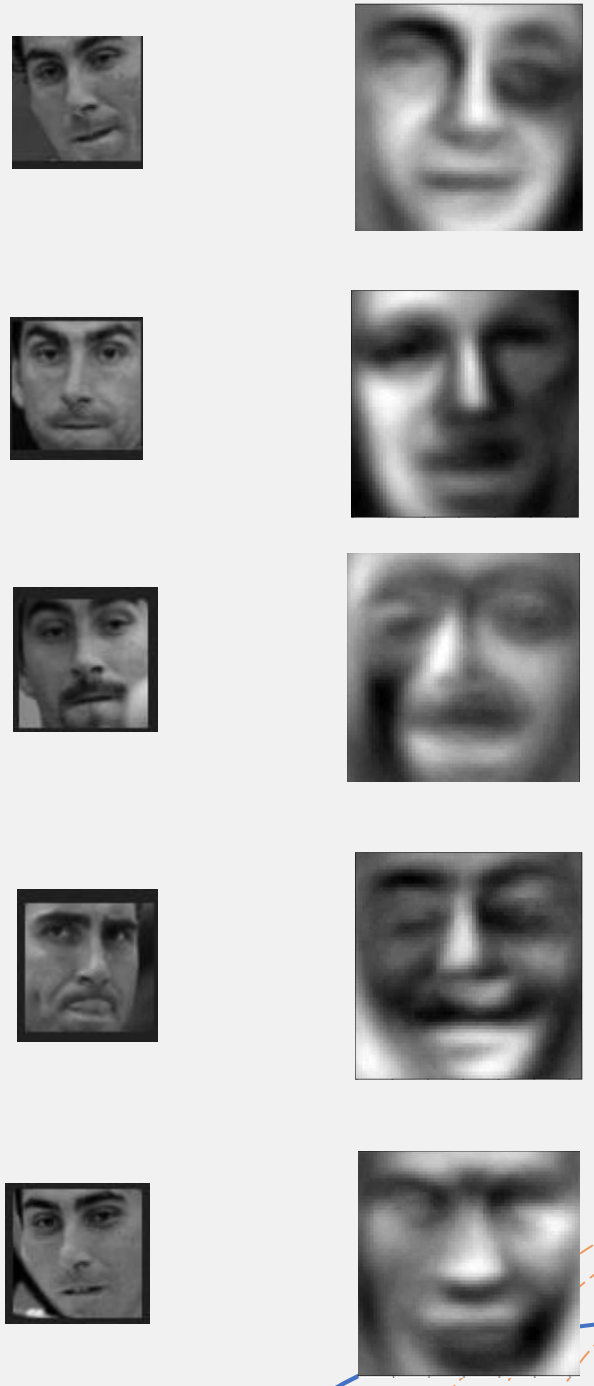
-1425.81	-362.108	288.5515	-553.75	691.7054	58.35175	241.3627	88.87303	-453.779	-78.2897	218.5828
-347.617	-749.845	217.9711	480.5239	-467.411	148.3275	59.32953	-534.766	431.7741	-156.607	990.3029
286.5356	11.3639	-1.05838	-486.336	-94.6091	-253.078	113.5373	-56.4097	-164.663	-327.627	112.3002
-57.9662	-469.857	293.0364	-177.004	-200.418	181.0303	-338.676	325.5024	233.5668	-282.329	225.2952
386.4585	83.11989	109.1573	127.0544	249.1209	-16.0998	-42.0628	304.9689	-127.149	115.2882	371.2967





9. Albert Costa

+ 복원 결과 다음과 같은 결과를 가짐을 알 수 있었습니다.



10. Alejandro Toledo

+ 왼쪽부터 1, 2, 3, 4, 5번 사진입니다.
+ 모든 사진이 유사하게 나온 것을 확인할 수 있었습니다.
+ 비교 결과 3 6 8 11 12 14 16 21 24 26 eigenvector에서 유사한 사진의 값이 다르게 나오고, 나머지 eigenvector에서는 유사한 사진의 값이 비슷하게 나오는 것을 확인할 수 있었습니다.

Alejandro_Toledo 1	-11208.1	1660.134	-960.861	625.3875	1192.809	-1239.54	568.1313	194.8208	-443.738	681.3973	-694.509	632.9201	258.2662	258.5741	-352.517	509.2551	905.9727	-878.952	-537.39
Alejandro_Toledo 2	-12800.1	-97.1806	230.2203	977.6093	-266.908	-677.175	-576.012	-139.64	-19.0513	229.6075	128.2036	-95.8675	57.39671	266.469	523.8812	359.7138	143.8739	-23.6732	240.91
Alejandro_Toledo 3	-9250.97	3039.32	-120.823	182.2813	-247.834	-638.749	1207.28	615.0592	1216.537	794.9029	335.0297	228.3906	1008.554	-809.342	231.6261	1416.48	35.11973	-895.372	345.64
Alejandro_Toledo 4	-10987.5	-1750.79	258.8966	-479.9	276.9011	1143.585	-64.2149	606.1831	219.3699	224.4444	643.8148	-88.0982	-394.684	-49.6221	907.0226	266.3578	-69.3598	-216.546	44.9924
Alejandro_Toledo 5	-12254.4	350.2482	70.16313	1102.119	-6.40296	-681.108	-556.534	-492.483	249.3046	255.0159	-77.1319	-94.8647	153.051	704.0985	299.8548	-12.3468	-353.233	-344.85	16.8604

318.6584	-434.326	184.2406	237.2844	311.5449	21.70077	519.8903	787.5969	-136.112	262.125	520.7178
-218.675	-2.8917	-220.081	-11.3972	-40.3386	-227.884	-234.735	-16.5514	327.3134	-46.0109	-100.256
375.4398	56.05402	216.6023	-23.3334	404.1925	-526.104	223.1964	40.02124	383.6134	175.9945	427.8562
-1154.43	-27.04	20.06085	43.88077	216.8102	1065.822	670.3424	35.53759	342.9753	486.4978	534.0919
-293.31	-63.7498	-365.195	-297.863	-26.1721	-125.879	98.90779	213.1109	59.66808	75.74272	293.9519





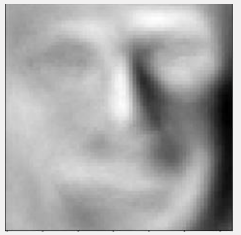
10. Alejandro Toledo

+ 복원 결과 다음과 같은 결과가 나옴을 알 수 있었습니다.







결과 분석

- 분석 결과 3 4 6 7 11 14 26번 eigen vector가 같은 사람의 얼굴이라도 상이한 값의 상수를 가지는 경향을 가짐을 알 수 있었습니다. 그 외의 eigen vector들은 같은 얼굴에 대해서 비슷한 값의 상수를 가짐을 알 수 있었습니다.
- C0 부터 c29까지의 eigen vector들은 python에서 결과를 도출해낼 때 eigen value 값이 큰 순서대로 정렬되어 있었습니다. 즉, c0가 가장 큰 eigen value를 가지고 c29가 가장 작은 eigen value를 가짐을 알 수 있었습니다. 분석 결과 대체적으로 eigen value가 큰 eigen vector가 가지는 상수값의 절댓값이 큰 경우가 많은 것을 확인해 볼 수 있었습니다.
- 그리고 c0, c1등 앞쪽에 분포한 eigen vector들의 상수값은 서로 다른 사람일 때 값의 차이가 큰 것을 확인할 수 있었습니다. 가장 큰 eigen value를 갖는 c0에 대한 상수값은 같은 사람일 때 거의 유사한 값을 가짐을 확인할 수 있었습니다. C29, c28 등 뒤쪽에 분포한 eigen vector들의 상수값은 서로 다른 사람이라도 그렇게 많은 차이를 갖지 않음을 확인해볼 수 있었습니다.
- 그리고 원본 사진을 복원해본 결과 4096개의 eigen vector 중 30개의 eigen face만을 이용해 복원을 했기 때문에 원본 사진과 똑같지는 않지만 그래도 어느정도 유사한 형태를 가지는 것을 확인해 볼 수 있었습니다.