

빅데이터 분석 전문가 양성 과정 15기 최종발표

FIND RICH FACE

당신은 부자가 되고싶조 | 남동우 윤민기 이창수 한하영

1

팀원 소개

2

주제 소개

3

프로젝트 일정

4

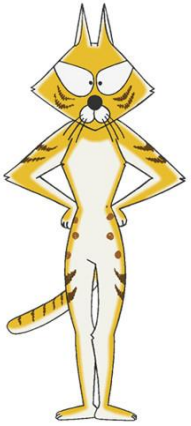
프로젝트 과정

5

결론

01 팀원 소개

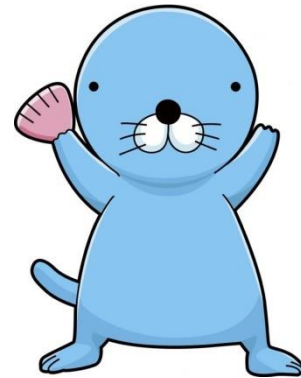
팀원 소개



남동우/수학과
열정



윤민기/경영학과
냉정



이창수/산업공학과
순한맛



한하영/사회복지학과
매운맛

02 주제 소개

2

주제 소개

“관상은 과학이다”



성형하면 인생이 바뀐다?
(관상은 과학이다)
수원점집 수원 ...

올마이티TV

YouTube - 2020. 2. 1.



'관상 is 사이언스?' 박사방 3인방(강훈, 조주빈, 이원호) 얼굴 ...

세계일보

YouTube - 2020. 5. 1.



최현석 관상은 과학~!?
요리조리 왔다갔다~! 맛
집 탈탈 ...

전영주의 미남TV

YouTube - 2020. 1. 21.



2

주제 소개

✓ 관상 관련 연구 사례

① 연구 배경

미국에서 선출된 관료들 얼굴에서의 특성이 그들의 불법 행위를 예측하는지 알아보기 위한 연구 수행

② 연구 방법

실제 미국 정치인 얼굴 사진을 실험자극으로 사용, 이 중 절반은 뇌물 혐의로 기소되었던 정치인들

나머지 절반은 뇌물 관련 전과가 없는 정치인들

사전 지식이 전혀 없는 참가자들이 정치인들의 흑백 얼굴사진을 연달아 본 다음 특성을 유추해보는 과제 수행

③ 연구 결과

참가자들은 비리를 저지른 '부패한' 정치인과 뇌물 혐의가 없는 '청렴한' 정치인들을 얼굴만 가지고 비교적 정확하게 구분
구체적으로 참가자들에 의해 더 부정적인 평가를 받은 사람들일수록 실제 비리 사건에 대해 법적 처벌 받은 것으로 확인

④ 연구 의의

단 시간 동안 타인의 얼굴 정보만을 사용하여 내린 판단이 비교적 정확하다는 점을 보여준다

2

주제 소개

- ✓ 이러한 일련의 연구들은 '관상은 과학이다'라는 주장을 뒷받침
- ✓ 재벌들의 이목구비를 분석해보면 부자가 될 관상을 구별해 볼 수 있지 않을까



03 프로젝트 일정

프로젝트 일정

과제 / 날짜	6/16	6/17	6/18	6/19	6/20	6/21	6/22	6/23	6/24	6/25	6/26	6/27	6/28	6/29	6/30	7/1	7/2	7/3
주제																		
Crawling																		
중간발표																		

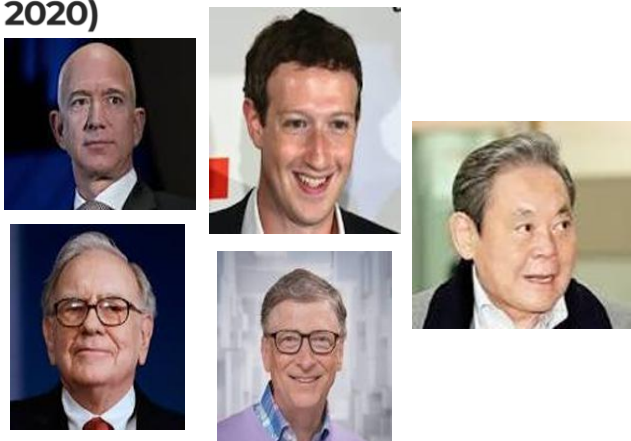
[illegible]

04 프로젝트 과정

4

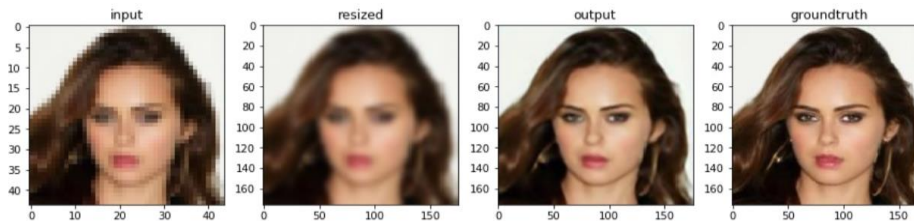
프로젝트 과정

Rich List Index: Top 500
Billionaires In The World (Meet
The Richest People On Earth,
2020)



① Crawling

② 이미지 화질 개선

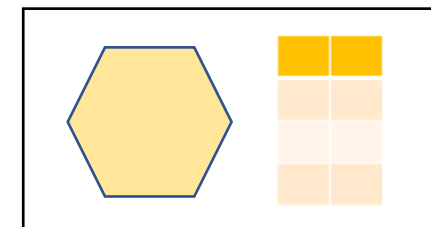


```
class FaceDistanceRatio:
    def __init__(self, dictionary):
        self.dictionary = dictionary

    def distance(self, a, b):
        self.a = self.dictionary[a]
        self.b = self.dictionary[b]
        x1, y1, x2, y2 = self.a[0], self.a[1], self.b[0], self.b[1]
        distance = math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
        return distance
```

③ 비율 계산

④ 시각화



⑤ 어플리케이션

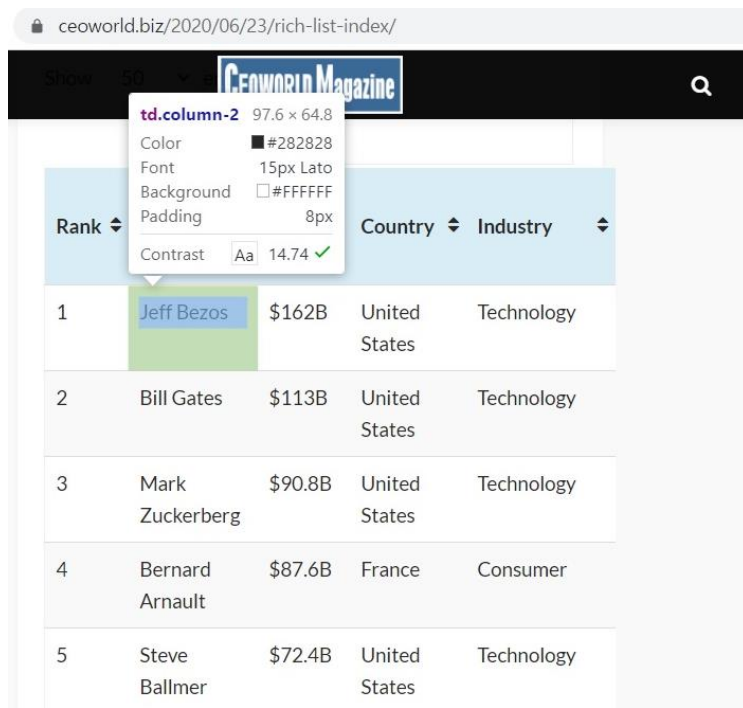
- ① 세계 재벌 500명 명단 → Crawling (이미지 데이터 수집) Package BeautifulSoup, urllib
- ② 이미지 화질 개선 및 저장 (Super Resolution) Package dlib, opencv, keras
- ③ 이목구비 비율 계산 (Face Ratio)
- ④ 시각화 Package Matplotlib, gridspec, GaussianMixture, PCA
- ⑤ 어플리케이션 구현 (kivy → Android Studio)

4

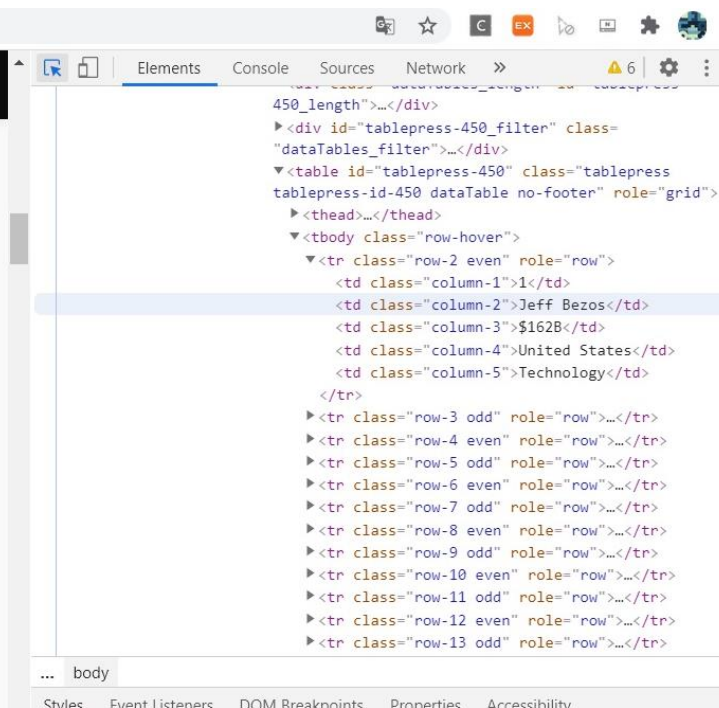
프로젝트 과정

① 데이터 수집

- ✓ <https://ceoworld.biz/2020/06/23/rich-list-index> 사이트에서 세계 재벌 500명 명단 추출
- ✓ Google 이미지에서 부자 1명 당 20장의 사진 Crawling



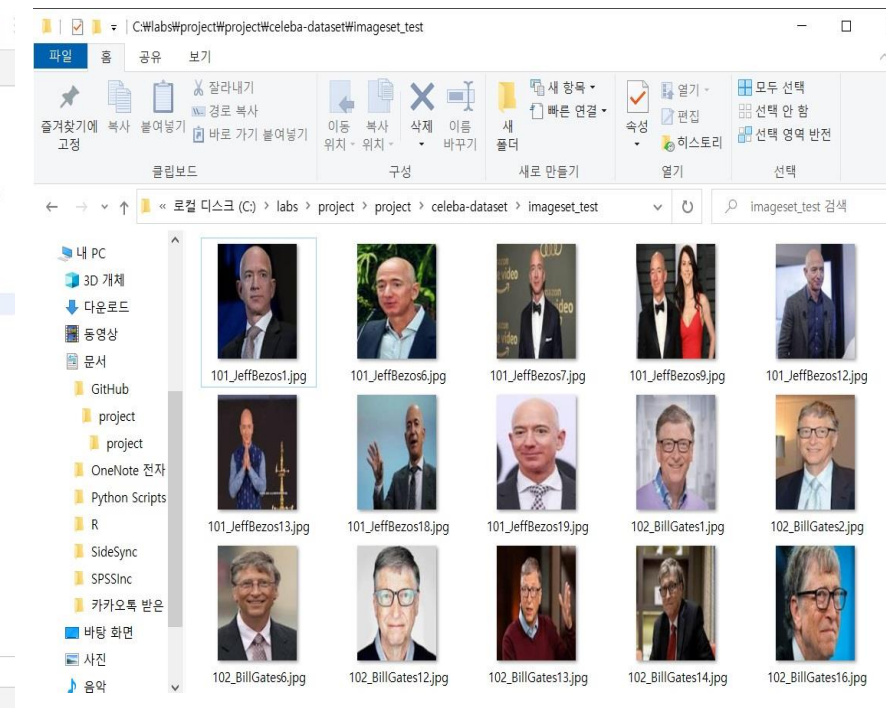
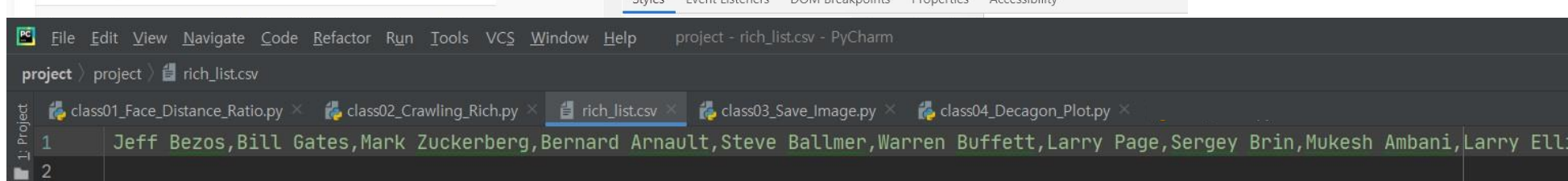
Rank	Name	Net Worth	Country	Industry
1	Jeff Bezos	\$162B	United States	Technology
2	Bill Gates	\$113B	United States	Technology
3	Mark Zuckerberg	\$90.8B	United States	Technology
4	Bernard Arnault	\$87.6B	France	Consumer
5	Steve Ballmer	\$72.4B	United States	Technology



```

450_length">...</div>
<div id="tablepress-450_filter" class="
"dataTable_filter">...</div>
<table id="tablepress-450" class="tablepress
tablepress-id-450 dataTable no-footer" role="grid">
  <thead>...</thead>
  <tbody class="row-hover">
    <tr class="row-2 even" role="row">
      <td class="column-1">1</td>
      <td class="column-2">Jeff Bezos</td>
      <td class="column-3">$162B</td>
      <td class="column-4">United States</td>
      <td class="column-5">Technology</td>
    </tr>
    <tr class="row-3 odd" role="row">...</tr>
    <tr class="row-4 even" role="row">...</tr>
    <tr class="row-5 odd" role="row">...</tr>
    <tr class="row-6 even" role="row">...</tr>
    <tr class="row-7 odd" role="row">...</tr>
    <tr class="row-8 even" role="row">...</tr>
    <tr class="row-9 odd" role="row">...</tr>
    <tr class="row-10 even" role="row">...</tr>
    <tr class="row-11 odd" role="row">...</tr>
    <tr class="row-12 even" role="row">...</tr>
    <tr class="row-13 odd" role="row">...</tr>
  </tbody>
</table>

```

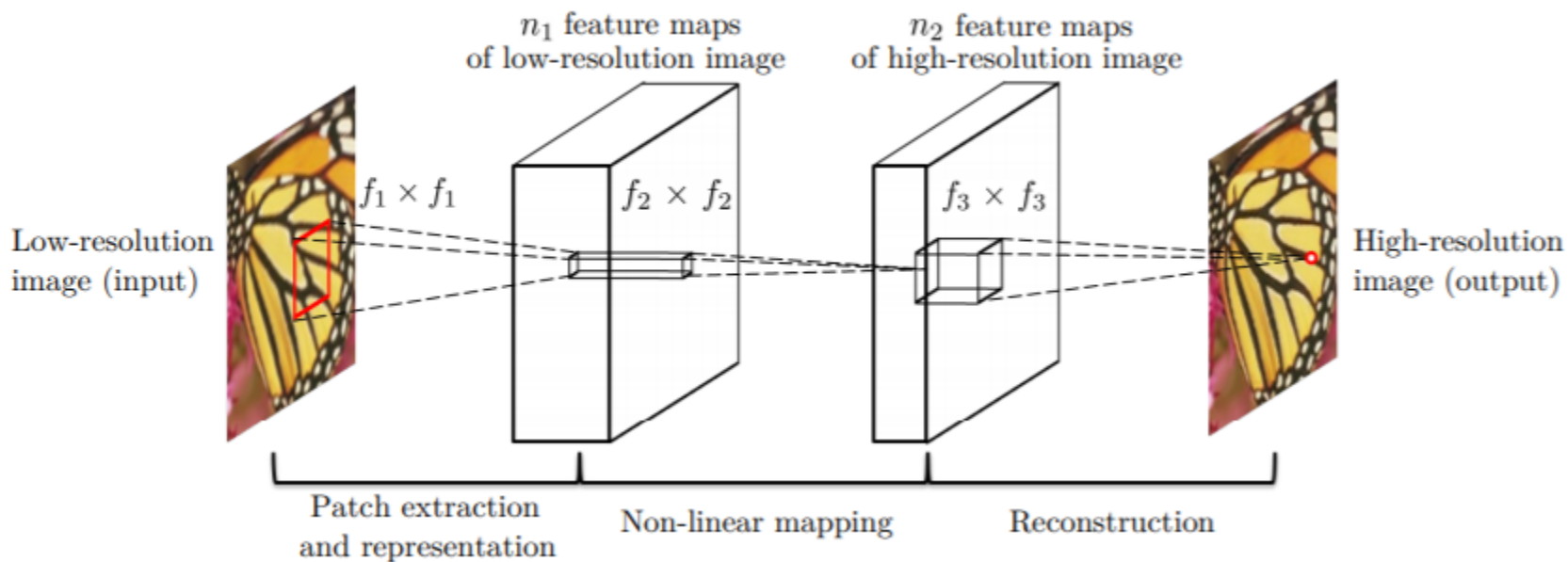
project > project > rich_list.csv

class01_Face_Distance_Ratio.py class02_Crawling_Rich.py rich_list.csv class03_Save_Image.py class04_Decagon_Plot.py

1	Jeff Bezos,Bill Gates,Mark Zuckerberg,Bernard Arnault,Steve Ballmer,Warren Buffett,Larry Page,Sergey Brin,Mukesh Ambani,Larry Elli
2	

▶ 부자 명단.CSV

- ✓ 정확한 결과값을 얻기 위해 Crawling한 이미지들을 딥러닝을 통해 화질 개선
- ✓ Super Resolution기법 : 이미지 혹은 영상 Low-Resolution을 High-Resolution으로 복원하는 기술
- ✓ SRCNN

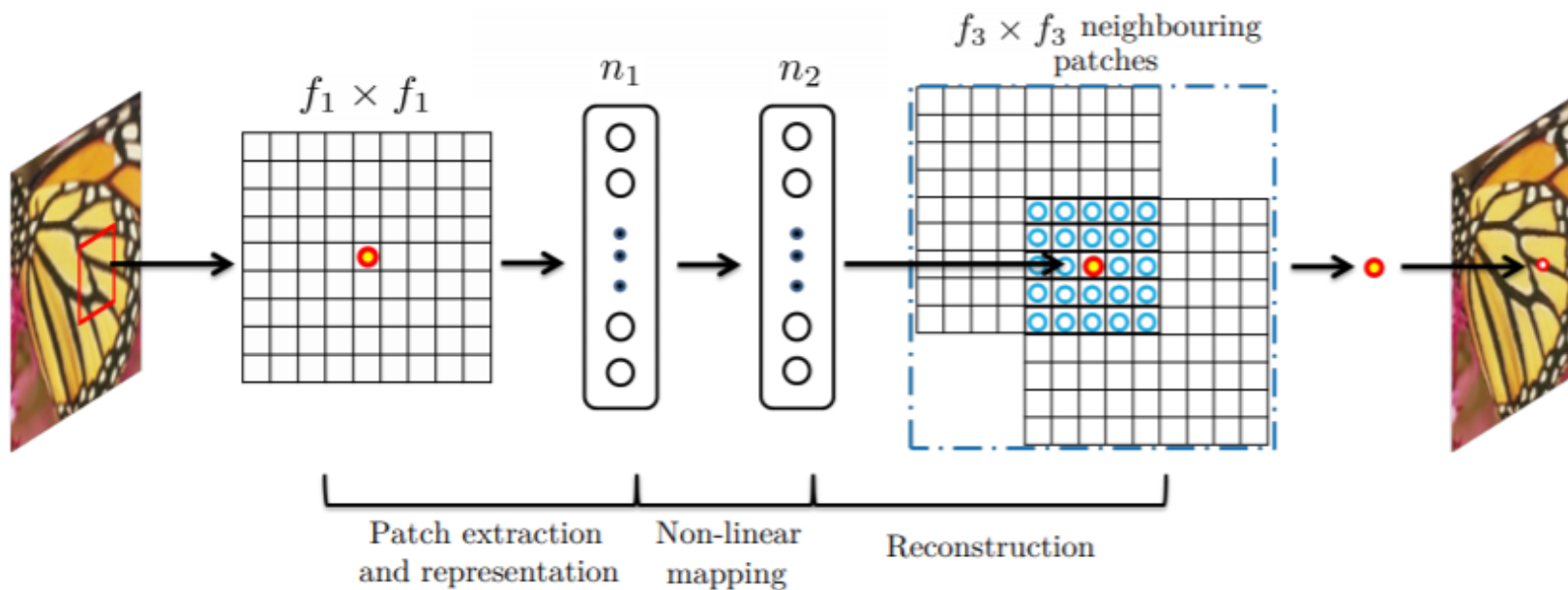


4

프로젝트 과정

② 이미지 화질 개선

- ✓ Low-Resolution Y : Input image
- ✓ High-Resolution F(Y) (restored): Output image
- ✓ Ground truth X : Label image
- ✓ Trained by 'L2 loss'

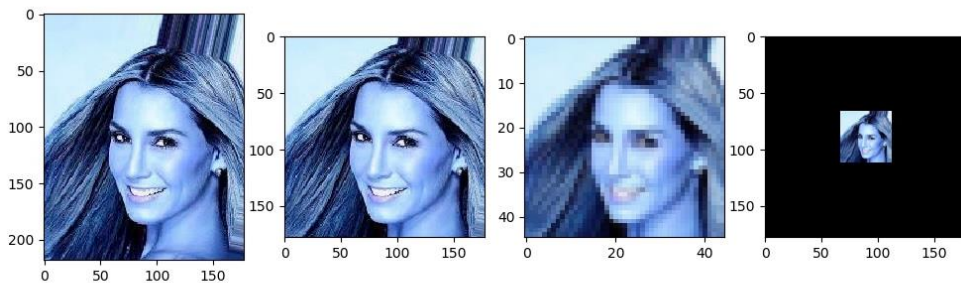
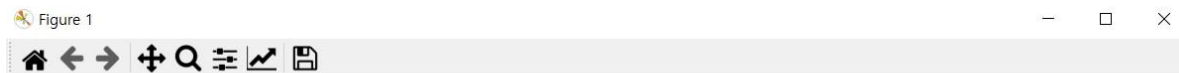


4

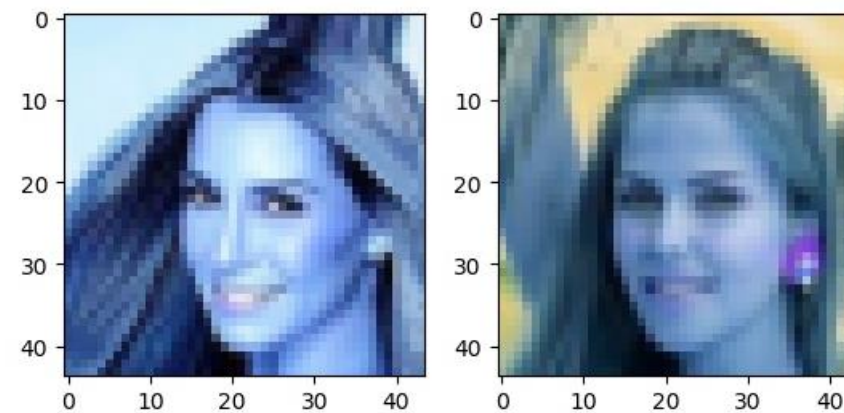
프로젝트 과정

② 이미지 화질 개선

- ✓ 프로젝트 데이터에 이미지 화질 개선 적용 과정
- ✓ SRCNN Image Training



▲ Training Image



▲ Training Image

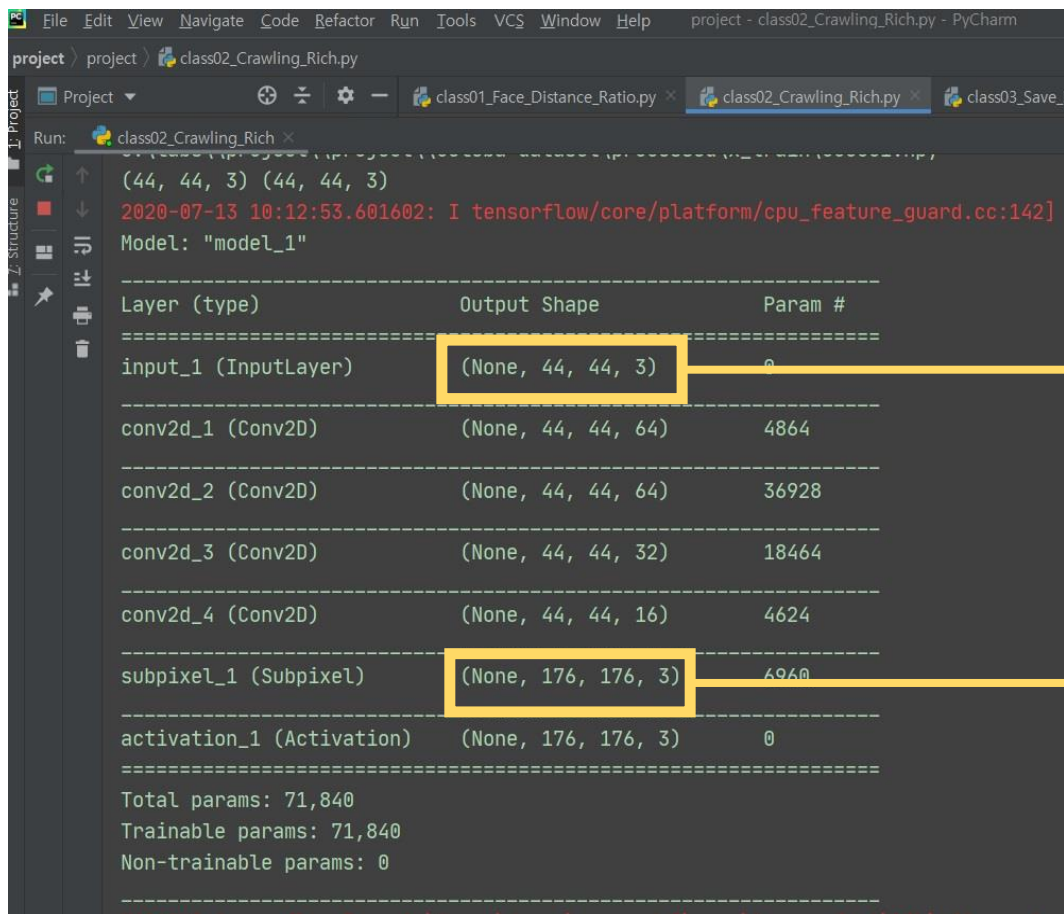
▲ Validation Image

4

프로젝트 과정

② 이미지 화질 개선

✓ SRCNN Image Training



The screenshot shows the PyCharm IDE with a project named 'class02_Crawling_Rich.py'. The Run console displays the output of the SRCNN training process. A table lists the model layers, their output shapes, and the number of parameters. Two specific output shapes are highlighted with yellow boxes and arrows pointing to explanatory text:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 44, 44, 3)	0
conv2d_1 (Conv2D)	(None, 44, 44, 64)	4864
conv2d_2 (Conv2D)	(None, 44, 44, 64)	36928
conv2d_3 (Conv2D)	(None, 44, 44, 32)	18464
conv2d_4 (Conv2D)	(None, 44, 44, 16)	4624
subpixel_1 (Subpixel)	(None, 176, 176, 3)	6960
activation_1 (Activation)	(None, 176, 176, 3)	0

Annotations:

- 기존 사진을 동일한 크기로 사진 crop (44, 44) size
- SRCNN 과정 후 최종 사진 크기 (176, 176) size

Additional output from the console:

```
Model: "model_1"
Total params: 71,840
Trainable params: 71,840
Non-trainable params: 0
```

4

프로젝트 과정

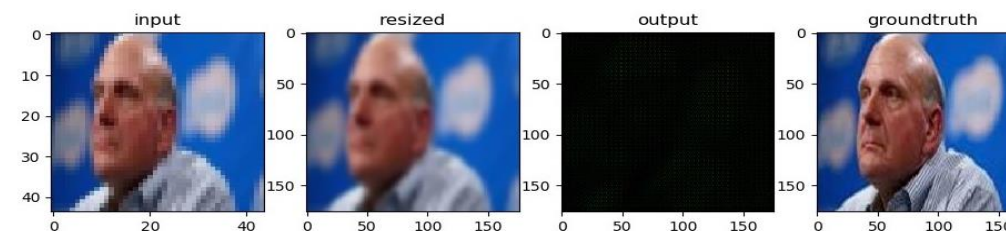
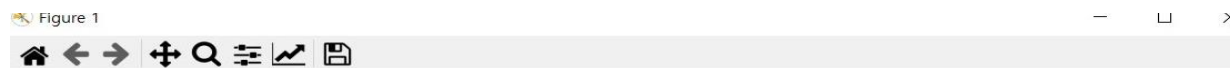
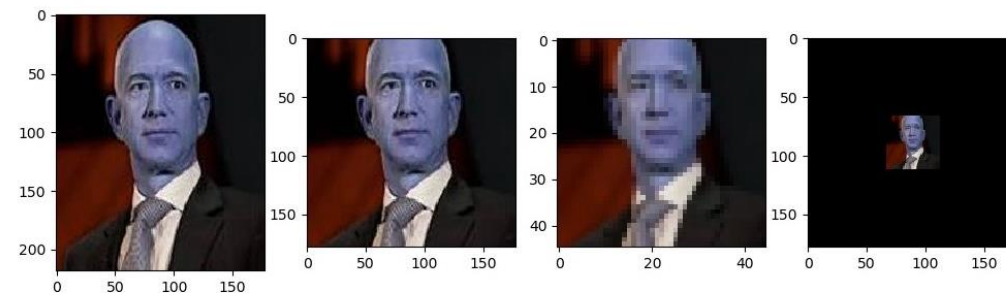
② 이미지 화질 개선

✓ SRCNN Image Training : Epoch 과정

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help project - class02_Crawling_Rich.py - PyCharm
project \ project \ class02_Crawling_Rich.py
Project \ project \ class01_Face_Distance_Ratio.py \ class02_Crawling_Rich.py \ class03_Save_Image.py \ class04_Decagon_Plot.py
Run: class02_Crawling_Rich.py
Epoch 1/10
1/62 [.....] - ETA: 2:48 - loss: 0.2586
2/62 [.....] - ETA: 1:42 - loss: 0.2523
3/62 [>.....] - ETA: 1:20 - loss: 0.2321
4/62 [>.....] - ETA: 1:12 - loss: 0.2134
5/62 [=>.....] - ETA: 1:04 - loss: 0.2032
6/62 [=>.....] - ETA: 1:00 - loss: 0.1959
7/62 [==>.....] - ETA: 57s - loss: 0.1876
8/62 [==>.....] - ETA: 55s - loss: 0.1808
9/62 [===>.....] - ETA: 53s - loss: 0.1763
10/62 [===>.....] - ETA: 51s - loss: 0.1698
11/62 [====>.....] - ETA: 50s - loss: 0.1633
12/62 [====>.....] - ETA: 49s - loss: 0.1553
13/62 [=====>.....] - ETA: 48s - loss: 0.1507
14/62 [=====>.....] - ETA: 46s - loss: 0.1451
15/62 [=====>.....] - ETA: 45s - loss: 0.1399
16/62 [=====>.....] - ETA: 44s - loss: 0.1359
17/62 [=====>.....] - ETA: 44s - loss: 0.1321
18/62 [=====>.....] - ETA: 43s - loss: 0.1278
19/62 [=====>.....] - ETA: 41s - loss: 0.1240
20/62 [=====>.....] - ETA: 40s - loss: 0.1204
21/62 [=====>.....] - ETA: 40s - loss: 0.1166
22/62 [=====>.....] - ETA: 39s - loss: 0.1140

```



Test Set Training 결과 ▲

4

프로젝트 과정

첫번째 Class ▶ CrawlingRich

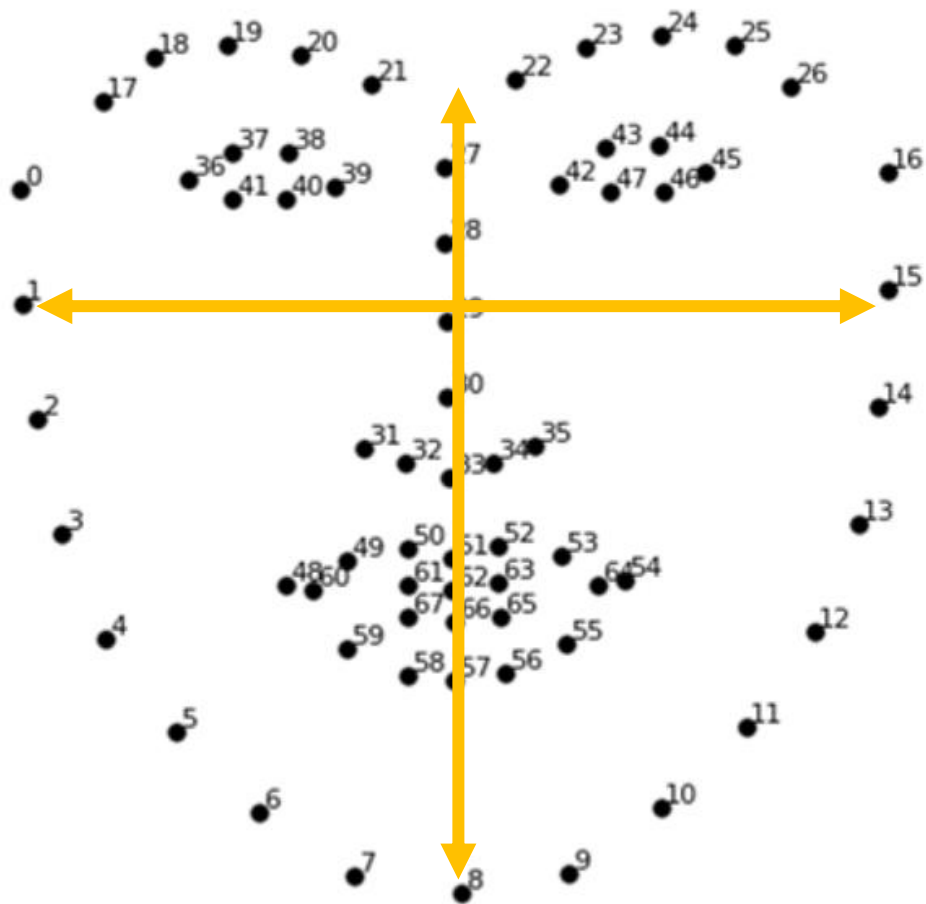
- img_resolution_train : 이미지 crop, 축소 후 이미지 train할 파일 생성
- img_resolution_train2 : 모델에 이미지를 train 후 fitting
- img_resolution_test : 이미지 crop, 축소 후 이미지 test할 파일 생성
- img_resolution_test2 : fitting한 모델로 test 이미지 데이터를 예측
- del_csv : CSV파일을 Dataframe으로 불러와 마지막줄 제거 후 다시 CSV파일로 저장
- Distance : 두 점 사이의 거리
- Radian : 얼굴의 코 각도를 리턴
- facing_67 : 저장된 얼굴사진들을 코각도, 얼굴개수로 걸러내 정보를 JSON파일에 저장
- json_to_dict : JSON파일을 Dictionary
- rich_list : 부자들의 CSV파일을 읽어서 리턴
- dict_to_pkl : Dictionary를 Pickle 파일 형태로 저장
- ✓ rich_total_dataframe : 부자들의 데이터와 부자유형타입, 변동계수를 포함한 최종 데이터프레임 생성
- ✓ crawling : Crawling으로 부자들의 목록과 사진을 저장한 후 전처리과정을 거쳐 부자들의 얼굴정보와 부자유형타입 등을 포함한 최종 데이터프레임 리턴

4

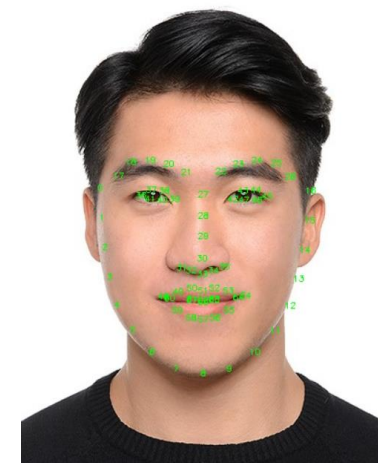
프로젝트 과정

③ 이목구비 비율 계산

✓ 얼굴에 0번부터 67번까지 좌표 생성 → 이목구비 간 거리 계산 가능



- ① 오른쪽 눈 가로 : 36, 39
- ② 오른쪽 눈 세로 : 37, 41
- ③ 오른쪽 눈꼬리 모양 : 39(눈안쪽) 대비 36
- ④ 오른쪽 눈썹 : 17, 18, 19, 20
- ⑤ 왼쪽 눈 가로 : 42, 45
- ⑥ 왼쪽 눈 세로 : 43, 47
- ⑦ 왼쪽 눈꼬리 모양 : 42(눈안쪽) 대비 45
- ⑧ 왼쪽 눈썹 : 22, 23, 24, 25
- ⑨ 미간 길이 : 39, 42
- ⑩ 코 가로 : 31, 35
- ⑪ 코 세로 : 27, 30
- ⑫ 인중 길이 : 33, 51
- ⑬ 입의 가로 : 48, 54
- ⑭ 입의 세로 : 51, 57
- ⑮ 입꼬리 높낮이 : 62, 66
- ⑯ 하관 : 57, 8



↔ 비율 기준 길이
✓ 유의미한 거리값

4

프로젝트 과정

③ 이목구비 비율 계산

- ✓ 화질 개선까지 끝마친 이미지에서 얼굴 이목구비 인덱스와 좌표 추출
- ✓ JSON형식으로 저장 후 최종적으로 PKL형식으로 저장

The image displays the PyCharm IDE interface. The main editor window shows the file `rich_101.json` with JSON data. The data consists of an array of objects, each containing face coordinates for different features. A yellow arrow points from the JSON data to the file explorer on the right.

The JSON data in `rich_101.json` is as follows:

```
[{"0": [52, 69], "1": [52, 78], "2": [53, 87], "3": [54, 97], "4": [57, 106], "5": [62, 114], "6": [70, 121], "7": [70, 121], "8": [70, 121]}, {"0": [52, 69], "1": [52, 78], "2": [53, 87], "3": [54, 97], "4": [57, 106], "5": [62, 114], "6": [70, 121], "7": [70, 121], "8": [70, 121]}, {"0": [69, 51], "1": [69, 57], "2": [70, 64], "3": [71, 69], "4": [75, 75], "5": [79, 80], "6": [84, 84], "7": [84, 84], "8": [84, 84]}, {"0": [51, 62], "1": [51, 66], "2": [52, 70], "3": [53, 75], "4": [55, 79], "5": [57, 83], "6": [59, 87], "7": [59, 87], "8": [59, 87]}, {"0": [111, 63], "1": [111, 67], "2": [112, 72], "3": [113, 76], "4": [113, 76], "5": [113, 76], "6": [113, 76], "7": [113, 76], "8": [113, 76]}, {"0": [34, 77], "1": [35, 87], "2": [36, 97], "3": [39, 107], "4": [39, 107], "5": [39, 107], "6": [39, 107], "7": [39, 107], "8": [39, 107]}, {"0": [45, 85], "1": [45, 92], "2": [47, 100], "3": [49, 108], "4": [49, 108], "5": [49, 108], "6": [49, 108], "7": [49, 108], "8": [49, 108]}, {"0": [72, 55], "1": [71, 60], "2": [72, 65], "3": [74, 71], "4": [74, 71], "5": [74, 71], "6": [74, 71], "7": [74, 71], "8": [74, 71]}, {"0": [100, 44], "1": [100, 48], "2": [100, 53], "3": [101, 58], "4": [101, 58], "5": [101, 58], "6": [101, 58], "7": [101, 58], "8": [101, 58]}]
```

The file explorer on the right shows a folder named `dataframe` containing the following PKL files:

- 101_Jeff Bezos.pkl
- 102_Bill Gates.pkl
- 103_Mark Zuckerberg.pkl
- 104_Bernard Arnault.pkl
- 105_Steve Ballmer.pkl
- 106_Warren Buffett.pkl
- 107_Larry Page.pkl
- 108_Sergey Brin.pkl
- 109_Mukesh Ambani.pkl
- 110_Larry Ellison.pkl

4

프로젝트 과정

③ 이목구비 비율 계산

- ✓ FaceRatio 클래스를 사용하여 부자 이목구비 비율값을 추출하고 DataFrame 형식으로 저장

The screenshot shows the PyCharm IDE with a project named 'project - class02_Crawling_Rich.py'. The 'Run' console displays two DataFrames. The first DataFrame has 9 rows and 6 columns, and the second DataFrame has 8 rows and 6 columns. Both DataFrames contain facial feature ratios for two subjects.

	right_eye_width	right_eye_height	...	mouth_shape	jaw_height
0	0.156205	0.064901	...	0.000000	0.286741
1	0.156205	0.064901	...	0.000000	0.286741
2	0.176666	0.046483	...	-0.046483	0.279867
3	0.156737	0.031246	...	-0.078115	0.218723
4	0.159872	0.057106	...	0.012769	0.282081
5	0.167228	0.047801	...	-0.029875	0.300897
6	0.155443	0.059170	...	-0.014793	0.252775
7	0.132780	0.059973	...	-0.009996	0.260651
8	0.167655	0.071424	...	-0.011904	0.261886

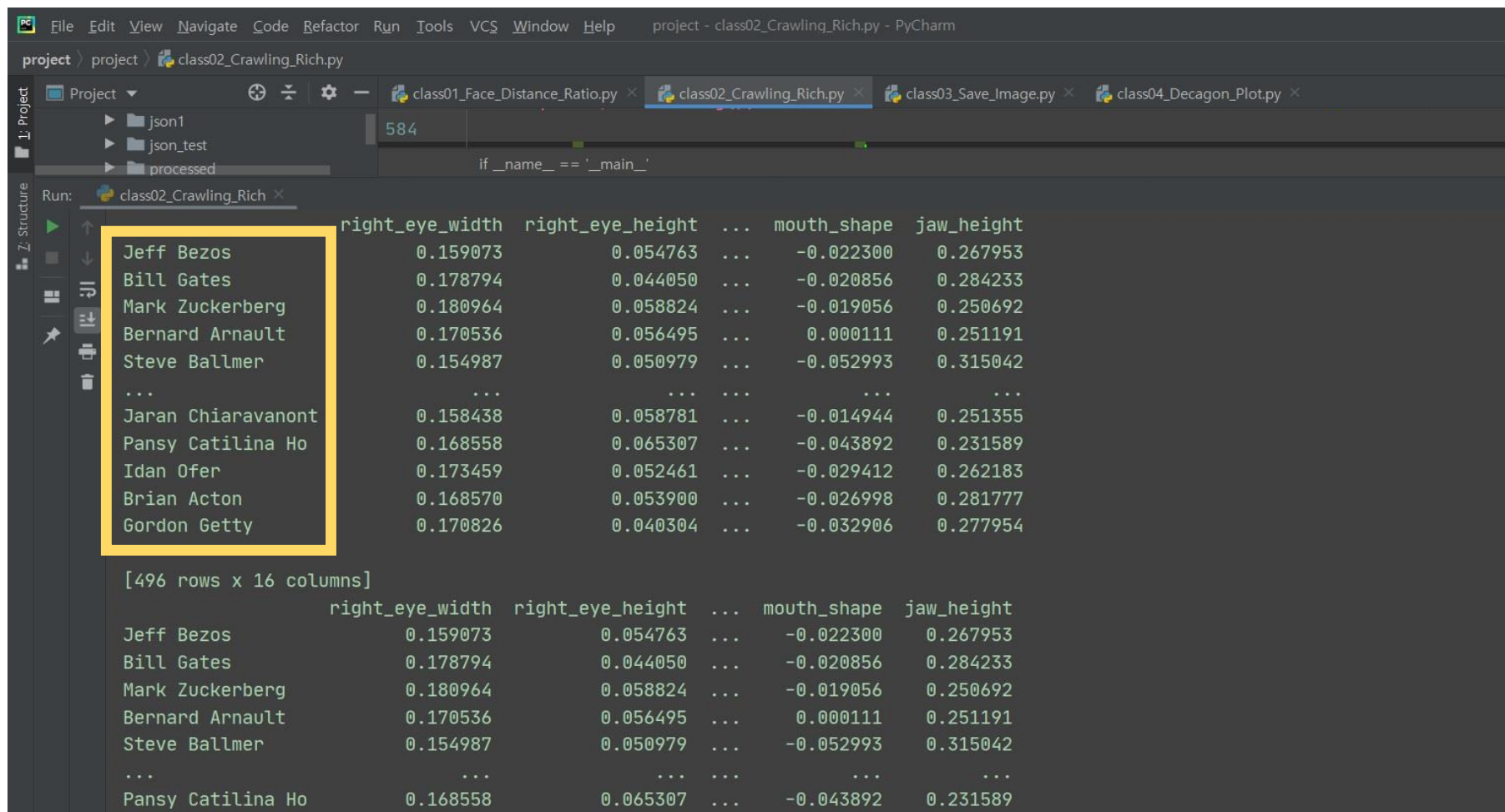
	right_eye_width	right_eye_height	...	mouth_shape	jaw_height
0	0.200407	0.027397	...	-0.061642	0.315061
1	0.171700	0.059536	...	-0.024807	0.278012
2	0.218774	0.050000	...	-0.025000	0.183333
3	0.175675	0.057036	...	0.018036	0.272939
4	0.147242	0.034953	...	0.042387	0.358568
5	0.189618	0.038458	...	-0.044868	0.282028
6	0.144092	0.028750	...	-0.028750	0.302219
7	0.182845	0.056270	...	-0.042202	0.281700

4

프로젝트 과정

③ 이목구비 비율 계산

✓ 각 비율값과 해당 부자이름을 매치한 DataFrame 생성



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help project - class02_Crawling_Rich.py - PyCharm
project > project > class02_Crawling_Rich.py
class01_Face_Distance_Ratio.py class02_Crawling_Rich.py class03_Save_Image.py class04_Decagon_Plot.py
584
if __name__ == '__main__':
    Run: class02_Crawling_Rich.py
    right_eye_width right_eye_height ... mouth_shape jaw_height
    Jeff Bezos 0.159073 0.054763 ... -0.022300 0.267953
    Bill Gates 0.178794 0.044050 ... -0.020856 0.284233
    Mark Zuckerberg 0.180964 0.058824 ... -0.019056 0.250692
    Bernard Arnault 0.170536 0.056495 ... 0.000111 0.251191
    Steve Ballmer 0.154987 0.050979 ... -0.052993 0.315042
    ... ... ... ... ...
    Jaran Chiaravanont 0.158438 0.058781 ... -0.014944 0.251355
    Pansy Catilina Ho 0.168558 0.065307 ... -0.043892 0.231589
    Idan Ofer 0.173459 0.052461 ... -0.029412 0.262183
    Brian Acton 0.168570 0.053900 ... -0.026998 0.281777
    Gordon Getty 0.170826 0.040304 ... -0.032906 0.277954
    [496 rows x 16 columns]
    right_eye_width right_eye_height ... mouth_shape jaw_height
    Jeff Bezos 0.159073 0.054763 ... -0.022300 0.267953
    Bill Gates 0.178794 0.044050 ... -0.020856 0.284233
    Mark Zuckerberg 0.180964 0.058824 ... -0.019056 0.250692
    Bernard Arnault 0.170536 0.056495 ... 0.000111 0.251191
    Steve Ballmer 0.154987 0.050979 ... -0.052993 0.315042
    ... ... ... ... ...
    Pansy Catilina Ho 0.168558 0.065307 ... -0.043892 0.231589
```

4

프로젝트 과정

③ 이목구비 비율 계산

✓ 부자 500명의 이목구비별 비율 평균값 추가

The screenshot shows a PyCharm IDE with a project named 'project - class02_Crawling_Rich.py'. The main editor displays a pandas DataFrame with 496 rows and 16 columns. The columns include 'right_eye_width', 'right_eye_height', 'mouth_shape', and 'jaw_height'. The DataFrame is sorted by 'right_eye_width' in descending order. The first row is for 'Gordon Getty' with values 0.170826, 0.040304, ..., -0.032906, 0.277954. The last row is for 'rich_mean' with values 0.162557, 0.052153, ..., -0.022534, 0.268134. The 'rich_mean' row is highlighted with a yellow border. Below the first DataFrame, there is another DataFrame with 497 rows and 16 columns, which is a duplicate of the first one, starting with 'Jeff Bezos' and ending with 'Mark Zuckerberg'.

```
Run: class02_Crawling_Rich.py
Gordon Getty      0.170826      0.040304      ...      -0.032906      0.277954

[496 rows x 16 columns]
      right_eye_width  right_eye_height  ...  mouth_shape  jaw_height
Jeff Bezos          0.159073          0.054763  ...  -0.022300      0.267953
Bill Gates          0.178794          0.044050  ...  -0.020856      0.284233
Mark Zuckerberg      0.180964          0.058824  ...  -0.019056      0.250692
Bernard Arnault     0.170536          0.056495  ...   0.000111      0.251191
Steve Ballmer       0.154987          0.050979  ...  -0.052993      0.315042
...                ...                ...      ...      ...
Pansy Catilina Ho   0.168558          0.065307  ...  -0.043892      0.231589
Idan Ofer           0.173459          0.052461  ...  -0.029412      0.262183
Brian Acton         0.168570          0.053900  ...  -0.026998      0.281777
Gordon Getty        0.170826          0.040304  ...  -0.032906      0.277954
rich_mean           0.162557          0.052153  ...  -0.022534      0.268134

[497 rows x 16 columns]
      right_eye_width  right_eye_height  ...  mouth_shape  jaw_height
Jeff Bezos          0.159073          0.054763  ...  -0.022300      0.267953
Bill Gates          0.178794          0.044050  ...  -0.020856      0.284233
Mark Zuckerberg      0.180964          0.058824  ...  -0.019056      0.250692
```


4

프로젝트 과정

③ 이목구비 비율 계산

✓ 부자 500명의 이목구비별 비율 변동계수 추가

project - class02_Crawling_Rich.py - PyCharm

project > project > class02_Crawling_Rich.py

584

if __name__ == '__main__':

Run: class02_Crawling_Rich

Pansy Catilina Ho	0.168558	0.065307	...	-0.043892	0.231589
Idan Ofer	0.173459	0.052461	...	-0.029412	0.262183
Brian Acton	0.168570	0.053900	...	-0.026998	0.281777
Gordon Getty	0.170826	0.040304	...	-0.032906	0.277954
rich_mean	0.162557	0.052153	...	-0.022534	0.268134

[497 rows x 16 columns]

	right_eye_width	right_eye_height	...	mouth_shape	jaw_height
Jeff Bezos	0.159073	0.054763	...	-0.022300	0.267953
Bill Gates	0.178794	0.044050	...	-0.020856	0.284233
Mark Zuckerberg	0.180964	0.058824	...	-0.019056	0.250692
Bernard Arnault	0.170536	0.056495	...	0.000111	0.251191
Steve Ballmer	0.154987	0.050979	...	-0.052993	0.315042
...
Idan Ofer	0.173459	0.052461	...	-0.029412	0.262183
Brian Acton	0.168570	0.053900	...	-0.026998	0.281777
Gordon Getty	0.170826	0.040304	...	-0.032906	0.277954
rich_mean	0.162557	0.052153	...	-0.022534	0.268134
rich_cv	0.064889	0.168981	...	-0.769212	0.091070

✓ 변동계수 = 표준편차/평균값

- 표준편차는 평균값이 큰 데이터 쪽이 커지는 경향
- 따라서 다른 평균값을 가진 데이터를 비교하는 경우 평균값의 차이를 조정하여 비교하기 위함

4

프로젝트 과정

③ 이목구비 비율 계산

✓ 변동계수 값이 작은 순으로 정렬, 컬럼 축소 및 타입 추가

PyCharm interface showing a project named 'project - class02_Crawling_Rich.py'. The main editor displays a table of data with columns: name, eye_between, left_eye_width, right_eye_width, left_eyebrow, right_eyebrow, nose_width, nose_height, jaw_height, mouth_width, philtrum, and type. The table is sorted by the 'eye_between' column in ascending order. A yellow box highlights the first few rows of the table, and a yellow callout box lists the features being calculated.

	eye_between	left_eye_width	right_eye_width	left_eyebrow	right_eyebrow	nose_width	nose_height	jaw_height	mouth_width	philtrum	type
Bernard Arnault	0.170550	0.050475	...	0.000111	0.251191						
Steve Ballmer	0.154987	0.050979	...	-0.052993	0.315042						
...						
Idan Ofer	0.173459	0.052461	...	-0.029412	0.262183						
Brian Acton	0.168570	0.053900	...	-0.026998	0.281777						
Gordon Getty	0.170826	0.040304	...	-0.032906	0.277954						
rich_mean	0.162557	0.052153	...	-0.022534	0.268134						
rich_cv	0.064889	0.168981	...	-0.769212	0.091070						

[498 rows x 16 columns]

eye_between left_eye_width ... philtrum type

- eye_between
- left_eye_width
- right_eye_width
- left_eyebrow
- nose_width
- right_eyebrow
- nose_height
- jaw_height
- mouth_width
- Philtrum
- type

두번째 Class ▶ Face Ratio

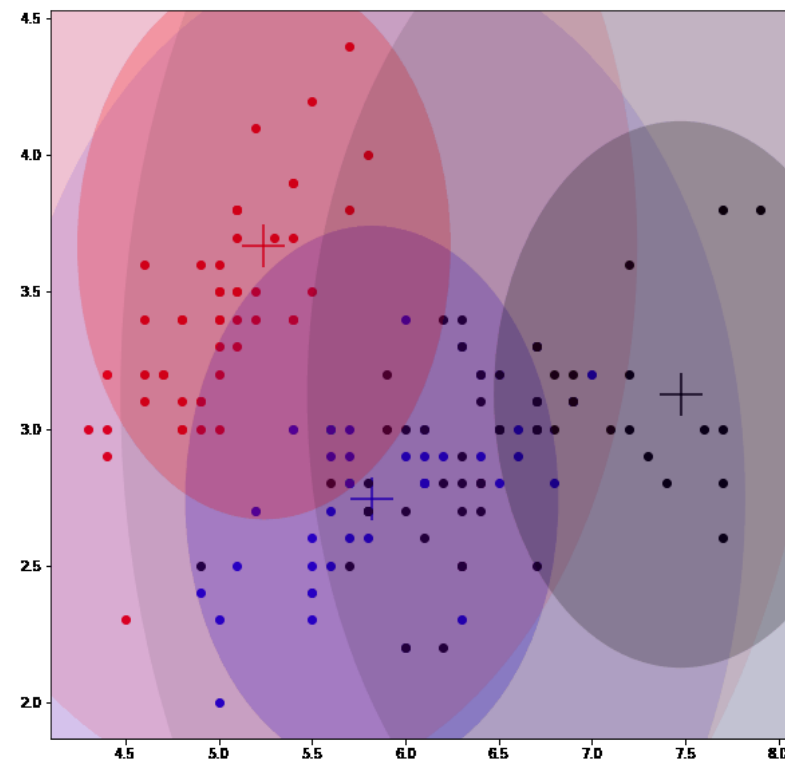
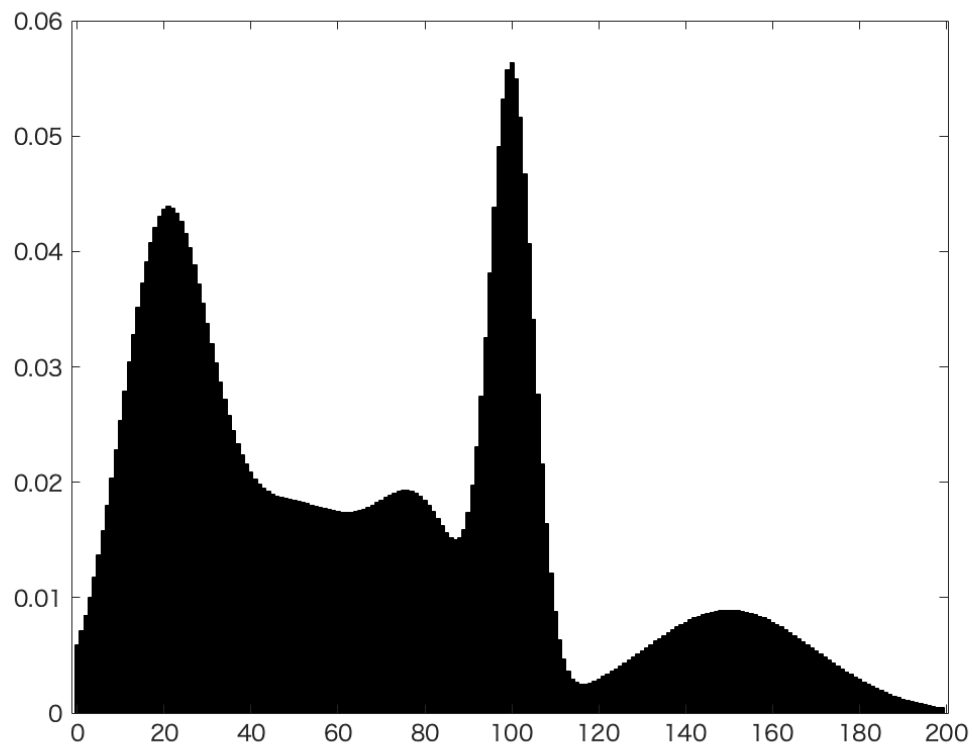
- distance : 두 점사이의 거리
- face_height : 얼굴 세로 길이
- face_width : 얼굴 가로 길이(광대에서 광대까지 거리)
- right_eye_width : 얼굴 가로 길이 대비 오른쪽 눈의 가로 길이 비
- right_eye_height : 얼굴 세로 길이 대비 오른쪽 눈의 세로 비
- left_eye_width : 얼굴 가로 길이 대비 오른쪽 눈의 가로 길이 비
- left_eye_height : 얼굴 세로 길이 대비 오른쪽 눈의 세로 비
- eye_between : 얼굴 가로 길이 대비 미간 길이의 비
- nose_width : 코의 가로 길이 비
- nose_height : 코의 세로 길이 비
- philtrum : 인중의 길이 비
- mouth_width : 입의 가로 길이 비
- mouth_height : 입의 세로 길이 비
- jaw_height : 하관의 길이 비
- right_eyebrow : 오른쪽 눈썹의 길이 비
- left_eyebrow : 왼쪽 눈썹의 길이 비
- right_eye_shape : 오른쪽 눈꼬리의 높낮이
- left_eye_shape : 왼쪽 눈꼬리의 높낮이
- mouth_shape : 입꼬리 높낮이
- info : 사진에서 구할 수 있는 모든 값들의 정보
- ✓ dict_to_df : Dictionary를 얼굴 수치 정보를 가진 DataFrame으로 변환
- ✓ dict_to_series : Dictionary를 Series로 변환

4

프로젝트 과정

④ 시각화

- ✓ GaussianMixture
- ✓ 전체 집단의 하위 집단의 확률분포가 가우시안 분포를 따르는 경우 사용
- ✓ 가우시안 혼합 모델은 비지도학습의 한 종류로, 데이터 클러스터링 (clustering)에 사용

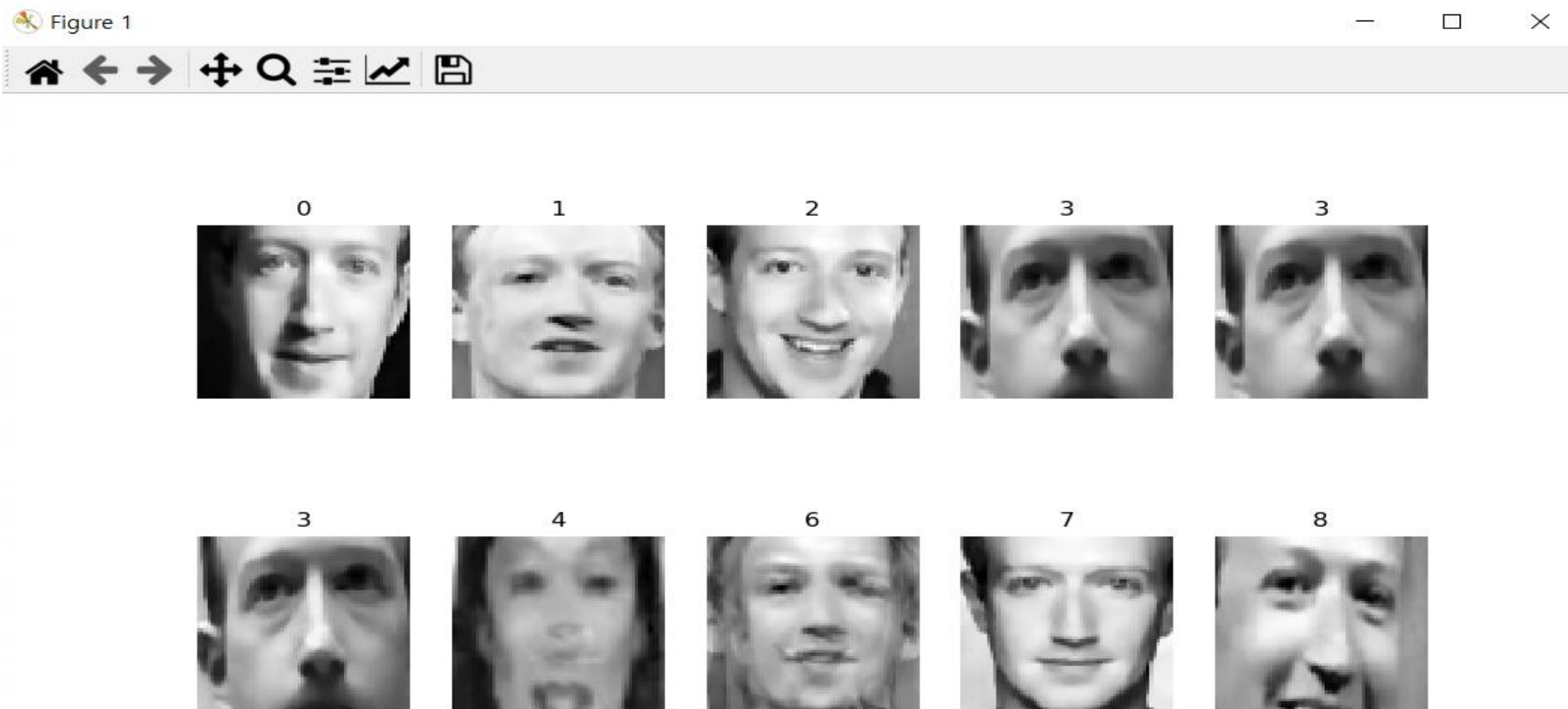


4

프로젝트 과정

④ 시각화

- ✓ Crawling한 부자 사진 n장에 GMM(Gaussian Mixture Model)을 적용한 결과 예시 (마크 주커버그)



4

프로젝트 과정

④ 시각화

- ✓ Test 이미지가 부자와 얼마나 닮았는지 그 결과를 퍼센트 값으로 출력
- ✓ 500명의 부자 중 가장 유사한 부자의 이름과 그 부자의 GMM (Gaussian Mixture Model) 이미지를 함께 출력

```
Run: class04_Decagon_Plot (1) ×
C:\Users\dong_\anaconda3\envs\teamproject\python.exe C:/labs/project/project/class04_Decagon_Plot.p
Using TensorFlow backend.
      위치   퍼센트(%)
0      미간    96.53
1  좌안 너비    92.30
2  우안 너비    96.07
3  왼눈썹 너비    57.72
4      코 너비    92.17
5  오른눈썹 너비    92.74
6      코 길이    99.78
7      턱 길이    89.06
8      입 너비    90.48
9  인중 길이    67.44
1
당신과 가장 닮은 부자는 Lakshmi Mittal입니다.
```

✓ 500명 부자들의 성공유형을 9가지로 분류하여 함께 출력

0) 자수성가형, 뼈빠지게 고생하는 타입인 당신! 언젠간 포브스 500에 당신의 이름이 실릴 날이...

1) 금수저형, 팡! 다음 생애에...

2) 투자의귀재, 보는 눈이 있는 당신! 손대는 것마다 투자성공!

3) 또라이형, 모 아니면 도! 포브스와 정신병원의 갈림길에 서 있는 당신!

4) 자퇴형, 일단 자퇴를 해라. 그러면 성공할 것이다.

5) 결혼형, 배우자 복이 있는 당신! 행복하세요!

6) 시인형, 당신은 부자의 관상은 아닙니다.. 하지만! 17세기 최고의 시인이 될 수 있는 관상! 돈보다 문학을 선택한 당신! 화이팅!

7) UN특사형, 당신은 부자의 관상은 아닙니다.. 하지만! 국제무대의 비둘기가 될 수 있는 관상! UN특사에 도전해보는 것은 어떨까요?!

8) 정치인형, 당신은 부자의 관상은 아닙니다.. 하지만! 뒷주머니로 챙기기 나름! 정치로 세계 500대 부자에 도전해 보세요!

9) Professor Type, 당신은 부자의 관상이 아닙니다.. 하지만! 대학원 5년을 견디면 교수가 될, 수도 있는 당신! 화이팅!



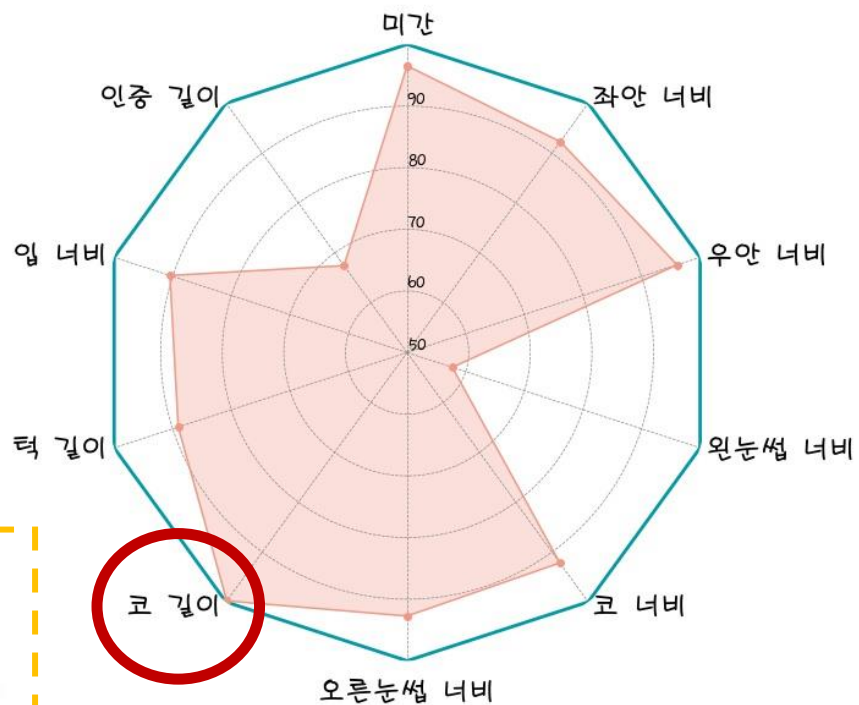
4

프로젝트 과정

④ 시각화

✓ 최종 결과

Similarity : 88.2564 %



Test 이미지



금수저형!
짱! 다음 생애에... (ex. 이건희)

어디가 가장 닮았을까?

위치	퍼센트 (%)
미간	96.53
좌안 너비	92.3
우안 너비	96.07
원눈썹 너비	57.72
코 너비	92.17
오른눈썹 너비	92.74
코 길이	99.78
턱 길이	89.06
입 너비	90.48
인중 길이	67.44

당신과 가장 닮은 부자는 Lakshmi Mittal입니다.



Gaussian Mixture Model

4

프로젝트 과정

④ 시각화

세번째 Class ▶ SaveImage

- facing_67_test : 자신의 이미지 파일 경로로 설정한 facing_67함수
- test_image_df_recall : 새로운 이미지 파일을 디렉토리에 저장하게 되면 해당 파일에 대한 최종데이터프레임 생성

네번째 Class ▶ DecagonPlot


- return_type : 최종데이터프레임의 부자유형타입에 따라 문구를 리턴
- plot_radar_chart : 디렉토리에 저장된 이미지에 대한 정보를 다각형, 표, 문구 등으로 출력

4

프로젝트 과정

⑤ 오류처리

- ✓ NoneType Error : Test 하려는 이미지가 없음을 나타냄
- ✓ 얼굴 인식이 안되어서 사진이 걸러지고 삭제됐을 때 발생하는 오류



```
Run: class03_Save_Image x
/home/minky/Downloads/tion/envs/project/bin/python /home/minky/project/project/class03_Save_Image.py
Using TensorFlow backend.
Traceback (most recent call last):
  File "/home/minky/project/project/class03_Save_Image.py", line 438, in <module>
    si.last_image_save()
  File "/home/minky/project/project/class03_Save_Image.py", line 394, in last_image_save
    self.img_resolution_test()
  File "/home/minky/project/project/class03_Save_Image.py", line 169, in img_resolution_test
    h, w, _ = img_sample.shape
AttributeError: 'NoneType' object has no attribute 'shape'

Process finished with exit code 1
```

★ 2: Favorites

▶ 4: Run ↩ 9: Git 📄 Python Console 📄 R Console 📄 Terminal 📄 6: TODO

Invalid VCS root mappings: The following directories are registered as VCS roots, but they are not: // <Project>/KivyMD // <Proj...rm/python-for-android // Configure... (a minute ago)

4

프로젝트 과정

⑤ 오류처리

- ✓ 해결방안 : Try-Except Error를 이용하여 오류 발생 시 출력문을 프린트하여 사용자가 디렉토리에서 이미지를 확인할 수 있도록 함

```

213         except AttributeError:
214             print("*****경로에 사진이 삭제되어있습니다. 디렉토리를 확인하여 주세요*****", flush=True)
215
216         return None
217
SavImage > img_resolution_test() > except AttributeError
class03_Save_Image x
/home/minky/Downloads/tion/envs/project/bin/python /home/minky/project/project/class03_Save_Image.py
Using TensorFlow backend.
*****경로에 사진이 삭제되어있습니다. 디렉토리를 확인하여 주세요*****
../project/celeba-dataset/processed_test_test/x_test/101_JeffBezos1.npy
../project/celeba-dataset/processed_test_test/y_test/101_JeffBezos1.npy
2020-07-14 11:04:10.095648: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use
2020-07-14 11:04:10.116716: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2095820000 Hz
2020-07-14 11:04:10.117273: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x55ebac91c750 initialized for platform Host (this does not guarantee th
2020-07-14 11:04:10.117302: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2020-07-14 11:04:10.117442: I tensorflow/core/common_runtime/process_util.cc:147] Creating new thread pool with default inter op setting: 2. Tune using inter_op_p
Process finished with exit code 0

```

4

프로젝트 과정

⑥ 어플리케이션 구현

- ✓ 파이썬 gui kivi모듈로 java를 활용하지 않은 어플리케이션 구현하려 했으나 크나큰 한계를 느끼고 도전 중
- ✓ 안드로이드 스튜디오로 개발 예정 (7월 중)



프로젝트 과정



얼굴 좌표를 이용한 비율 계산 클래스 생성
수집된 이미지를 JSON → DF로 저장
수집된 데이터 시각화 (Radar Chart)
재사용 가능한 클래스 생성

데이터 수집을 위한 crawling 함수 생성
수집된 이미지들의 이목구비 좌표 추출
좌표 추출된 이미지 사이즈 일괄화
어플리케이션 개발



역할분담



수집된 이미지들의 이목구비 좌표 추출
이미지 필터링 및 CSV 삭제 함수 생성
함수 내 오류 디버깅
재사용 가능한 클래스 생성

필터링 된 이미지 삭제 함수 생성
이미지 화질 개선 (Super Resolution)
수집된 데이터 시각화 (Radar Chart)
함수 내 오류 디버깅



05 결론

- ✓ 사람은 80% 이상 비슷하게 생겼다. (80% 이상의 유사도를 갖는다)
- ✓ 일반적으로 남자와 여자의 관상이 다를 것이라고 생각하지만,
본 프로젝트 모델에서는 남자라고 해서 높은 값, 여자라고 해서 낮은 값이 출력되지 않는다.
(데이터의 대부분이 남자임에도 불구하고 여자 얼굴을 테스트로 넣었을 때 80% 이상의 유사도가 출력된다)
- ✓ 또한 동양인과 서양인의 관상 역시 유사도의 차이가 크지 않다.
- ✓ 즉 우리 프로젝트 모델에서의 성별, 인종 등의 변수는 유의하지 않다.
- ✓ 반면, 아이는 골격이 작아서 이목구비를 인식하지만 비율이 성인과 차이가 난다.
- ✓ 따라서 나이에 따른 변수는 유의하다.

✓ 1000명의 부자를 50장씩 Crawling 할 때 34일 소요 예상

> 처음에는 1장당 30초~1분 소요, 코드 수정 후 15초 소요

> 그러나 저장사진이 많아질수록 다시 1장당 1분씩 소요

> 중간중간 수동으로 리셋 진행

대안책 : 사진을 저장하고 나서 list_eval_partition_test.csv 파일 마지막줄을 삭제하여 시간 단축

✓ 저장된 사진들 중 동명이인, 그림, 2명 중 1명만 인식되는 사진 등 목적에 맞지 않는 사진 출력

> 부자는 아니지만 유명한 동명이인의 사진이 추출

> 부자 명단의 뒤쪽으로 갈수록 해당 부자가 아닌 다른 사람의 사진 추출

대안책 : 정확한 사진 추출을 위한 대안이 없어서 이러한 경우 이상치로 남겨둠

- ✓ 한 장도 출력되지 않는 사진 존재 (중국 부자의 경우 고대 학자들 사진으로 인한 얼굴 인식 불가)
 - > 226 - 116 : " Zhang Yiming "
 - > 403 - 303 : "Zeng Fangqin"
 - > 위 두 인물은 사진 저장 실패대안책 : If문 내에서 이미지 파일이 없으면 Pass 하도록 코드 작성

- ✓ 모든 결과값(부자와 닮은 정도)이 80% 이상 나오고 그 미만의 값은 출력이 되지 않음
 - > 비율로 결과값을 도출하기 때문에 이목구비를 인식하기만 하면 비슷한 결과값이 나오는 한계 발생

✓ 참고문헌

Lin, C., Adolphs, R., & Alvarez, R. M.,

"Inferring Whether Officials Are Corruptible From Looking at Their Faces",

Psychological Science, 2018, 29, 1807-1823

Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaoou Tang, Fellow, IEEE,

"Image Super-Resolution Using Deep Convolutional Networks", arXiv:1501.00092v3, 2015

Bishop, Christopher M., "Pattern Recognition and Machine Learning", Springer-Verlag Berlin, Heidelberg, 2006

Murphy, Kevin P., "Machine Learning: A Probabilistic Perspective", MIT Press, Cambridge, Mass, 2012

✓ 참고 사이트

내 삶의 심리학 mind (<http://www.mind-journal.com>)

Rich List (<https://ceoworld.biz/2020/06/23/rich-list-index/>)

이미지 딥러닝 (<https://github.com/kairess>, <https://hoya012.github.io/blog/Single-Image-Super-Resolution-Overview/>)

Q & A

THANK YOU