인공지능 Project Report

Report 제목: Project #1

PCA를 이용한 얼굴 인식

제출기한:

2022년 10월 07일 (금)

~

2022년 10월 28일 (금)

학 과: 컴퓨터정보공학부

담당교수: 박철수 교수님

수강시간: 금요일 3,4교시

학 번: 2018202065

성 명: 박 철 준

- Introduction

1. 제목

PCA를 이용한 얼굴 인식

2. 목적

본 프로젝트에서는 PCA를 이용하여 얼굴 인식 과정을 시뮬레이션해야 한다. 이 실습에서 사용한 데이터는 LFW(Labeled Faces in the Wild) 얼굴 DB를 이용한다. 얼굴의 주성분들을 시각화 하여 확인해보고, PCA를 적용한 훈련 데이터로 학습시켜서 K-Nearest Neighbor(KNN) 알고리즘을 통해서 정확도를 측정하여야한다. 해당 Dataset을 이용하여 CNN(Convolutional neural network) 모델을 통해서 학습시켜보고 얼굴 이미지를 분류하여 비교해야한다.

과제 요구 사항

1. Dataset load와 데이터셋에 대한 설명한다.
2. PCA을 통해서 고유 얼굴 성분 출력한다.
3. PCA Whitening 사용한다.
4. 주성분의 개수의 차이를 두어 얼굴 이미지 재구성한다.
5. KNN 모델을 생성, 학습시켜 결과 분석한다. (data preprocessing, whitening 차이)
6. CNN 모델 학습, 결과 비교한다.

- Algorithm

1. 가장 작은 샘플 개수 찾는 과정
- people 변수에 lfw 데이터셋 저장
- counts 변수에 각 타깃이 나타난 횟수 저장
- people 변수와 counts 변수를 사용하여 타깃별 이름과 횟수 출력 후 가장 작은 샘플 개수 찾기

2. 데이터 편중방지와 훈련 데이터 및 테스트 데이터 생성
- mask 변수를 이용하여 데이터셋의 편중을 막기 위해 최소개수인 20의 이미지만 선택

3. PCA를 하여 고유 얼굴 성분 출력하는 과정
- x_train, x_test, y_train, y_test변수를 사용하여 테스트/훈련세트 나누기
- import한 PCA를 활용하여 pca 모델 생성 및 적용 이때 변수는 pca로 한다.
- pca의 설정은 주성분 100개, whitening 사용, random_state는 0으로 둔다.
- x_train_pca, x_test_pca 변수를 생성하여 pca 적용
- 고유 얼굴 성분 출력하기

4. KNN 모델을 생성하여 학습 결과를 도출하는 과정
- KNeighborsClassifier을 import하고
- knn 변수에 KNeighborsClassifier을 적용한다.
- x_train_pca, y_train 변수를 fit한다.
- 학습 결과 출력

5. 주성분의 개수의 차이를 두어 얼굴 이미지 재구성 과정
- mglearn.plots.plot_pca_faces 함수를 이용하여 재구성 이미지 출력

6. 원본 LWF 데이터셋을 이용하여 딥러닝 알고리즘인 Convolutional neural network(CNN)를 학습시켜 결과를 비교하는 과정
- people 변수에 lfw 데이터셋 저장
- counts 변수에 각 타깃이 나타난 횟수 저장
- people 변수와 counts 변수를 사용하여 타깃별 이름과 횟수 출력 후 가장 작은 샘플 개수 찾기
- mask 변수를 이용하여 데이터셋의 편중을 막기 위해 최소개수인 20의 이미지만 선택
- x_train, x_test, y_train, y_test변수를 사용하여 테스트/훈련세트 나누기
- CNN 모델 학습 스펙은 아래와 같다.
- conv2d_51 (Conv2D) (None, 85, 63, 32) 320
- max_pooling2d_35 (MaxPooling2D) (None, 42, 31, 32) 0
- conv2d_52 (Conv2D) (None, 40, 29, 64) 18496
- max_pooling2d_36 (MaxPooling2D) (None, 20, 14, 64) 0
- conv2d_53 (Conv2D) (None, 18, 12, 64) 36928
- flatten_17 (Flatten) (None, 13824) 0

- flatten_17 (Flatten) (None, 13824) 0
- dense_34 (Dense) (None, 64) 884800
- dense_35 (Dense) (None, 5) 325
- 손실과 정확도 계산 후 출력

- Result

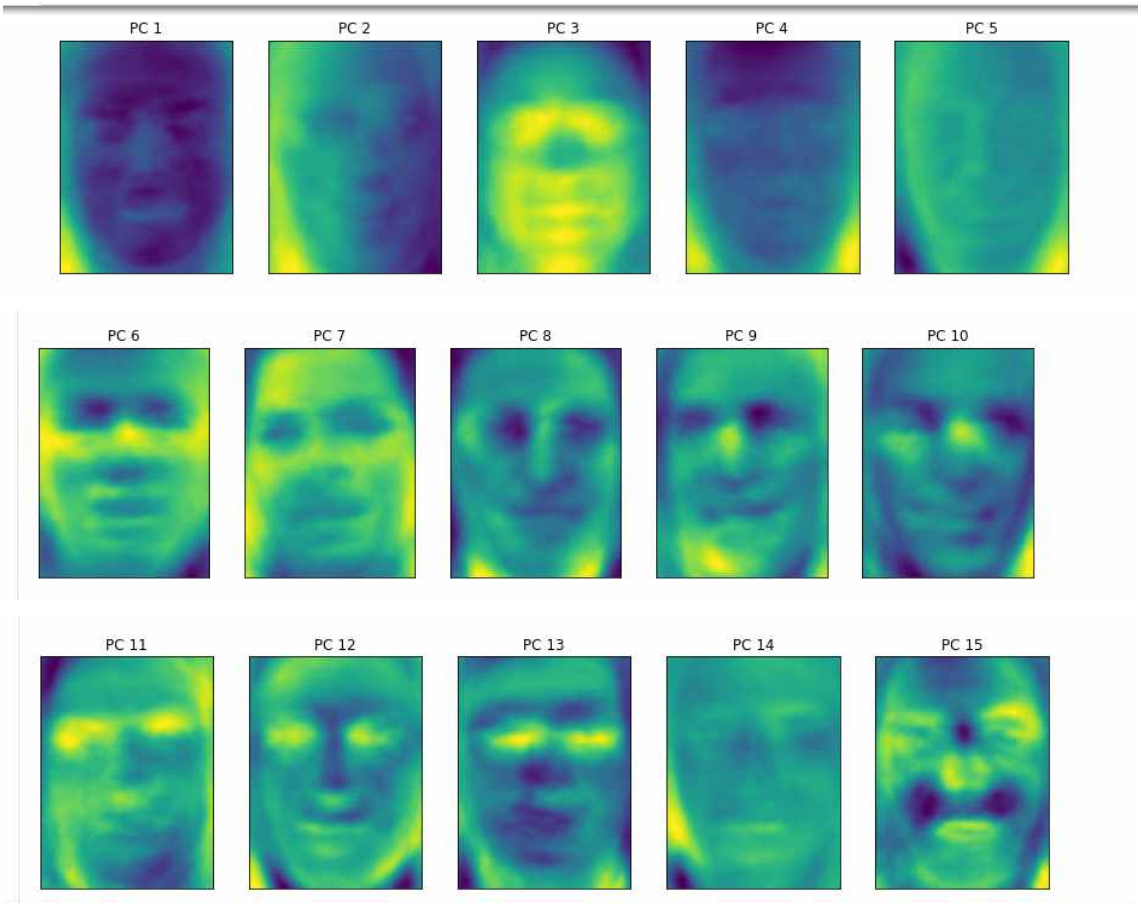
1. LFW(Labeled Faces in the Wild) Dataset에 대한 설명


```
Output exceeds the size limit. Open the full output data in a text editor
people.images.shape : (3023, 87, 65)
클래스 개수 : 62
Alejandro Toledo                39
Alvaro Uribe                    35
Amelie Mauresmo                 21

Andre Agassi                    36
Angelina Jolie                  20
Ariel Sharon                    77

Arnold Schwarzenegger           42
Atal Bihari Vajpayee            24
Bill Clinton                    29

Carlos Menem                    21
Colin Powell                   236
David Beckham                   31

Donald Rumsfeld                121
George Robertson                22
George W Bush                  530

Gerhard Schroeder              109
Gloria Macapagal Arroyo         44
Gray Davis                      26
...
Vicente Fox                     32

Vladimir Putin                  49
Winona Ryder                    24
```

LFW 데이터셋을 로드하고 사진을 출력함을 통해 잘로드했음을 알 수 있다. 62개의 클래스 데이터 3023개가 존재하고 이미지 사이즈는 87X65이다.

2. PCA whitening & PCA

- PCA를 하여 고유 얼굴 성분을 출력



- KNN 모델을 학습시키는 과정에서 PCA Whitening의 여부를 통한 정확도의 차이를 비교하여라.

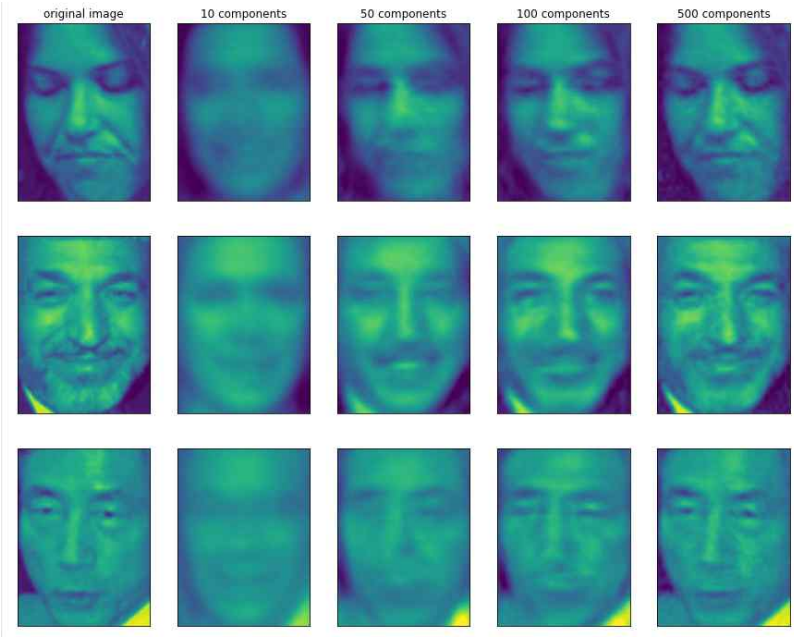
PCA Whitening = True

```
정확도: 0.24
```

PCA Whitening = False

```
정확도: 0.20
```

-



- 주성분의 개수의 차이를 두어 얼굴 이미지를 재구성하여 그리고 결과를 설명

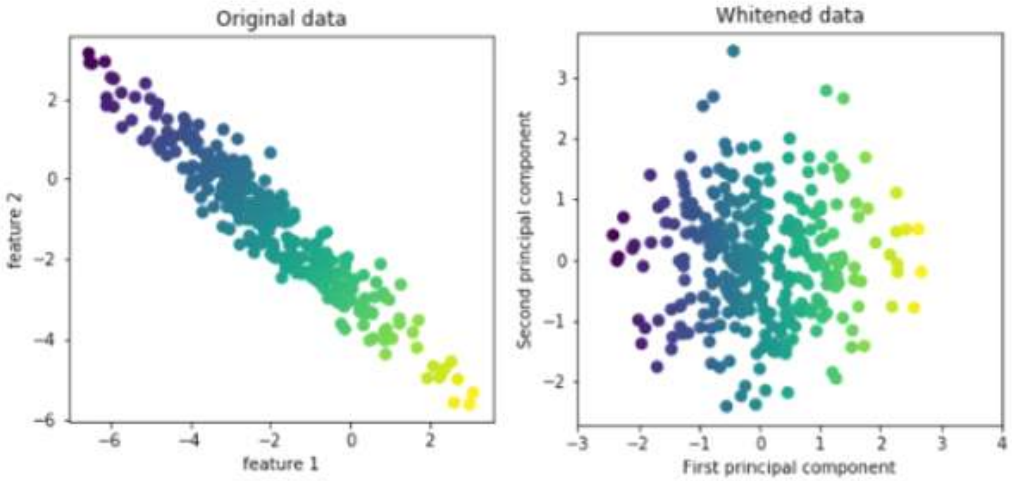
주성분이 많아질수록 원본이미지에 가까워진다. 해당 방식은 mglearn.plots.plot_pca_faces함수를 사용하였고 아래는 사용자의 입력에 따른 주성분에 대한 재구성한 사진이다.

왼쪽이 현재 주성분을 100개로 두어 재구성한 결과이고 오른쪽은 원본이미지 이다.

3. KNN & CNN

- KNN 모델을 생성, 학습시켜 결과 분석(data preprocessing, whitening 차이)

분류하려는 얼굴과 가장 비슷한 얼굴이미지를 찾는 1-최근접 이웃 분류기를 사용하는데 data preprocessing을 위해 차원감소에 활용되는 PCA와 밀접하게 연관된 전처리 과정인 whitening을 사용하여 input들의 feature들을 uncorrelated 하게 만들어주고, 각각의 variance를 1로 만들어주는 작업을 하여 결과를 그래프로 나타내면 아래와 같다.



결과적으로 whitening을 적용하여 나타낸 결과의 비교는 아래와 같다.

PCA Whitening = True

정확도: 0.24

PCA Whitening = False

정확도: 0.20

whitening을 적용한 것이 더 높은 결과를 나타냄을 알 수 있다.

- CNN 모델 학습, 결과 비교

```
Model: "sequential_19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_54 (Conv2D)          (None, 85, 63, 32)        320

 max_pooling2d_37 (MaxPoolin  (None, 42, 31, 32)       0
 g2D)

 conv2d_55 (Conv2D)          (None, 40, 29, 64)        18496

 max_pooling2d_38 (MaxPoolin  (None, 20, 14, 64)       0
 g2D)

 conv2d_56 (Conv2D)          (None, 18, 12, 64)        36928

 flatten_18 (Flatten)        (None, 13824)             0

 dense_36 (Dense)            (None, 64)                884800

 dense_37 (Dense)            (None, 5)                 325

=================================================================
Total params: 940,869
Trainable params: 940,869
Non-trainable params: 0
```

손실과 정확도는 아래와 같다.

```
12/12 [------------------------------]
Test loss: 2.7868828773498535
Test accuracy: 0.47999998927116394
```

CNN학습을 통한 결과가 PCA,KNN을 사용한것보다 월등히 좋다.

- Consideration

이번 과제를 진행하면서 이론에서 배운 Princple Component Analysis를 통하여 직접 구현하여 보았다. 추상적인 이해로 공분산을 구하고 고유벡터를 구해 basis function으로서 사용하여 사진 각각의 고유 수치를 구하는 과정을 직접 코딩을 통해 이미지 데이터를 픽셀 단위에서 어떠한 식으로 코딩하여 적용해야 하며 각각의 추상적인 과정이 직접적으로 어떠한 형태로 저장해야 하며 벡터로 표현할 시 어떤 부분을 고려해야할지 에 대해 생각해 볼 수 있었다. 또한 KNN(1-최근접 이웃 분류기를 사용)을 통해 해당 개념을 숙지 후 PCA에 실 적용을 할 수 있던 것 또한 개념적 접근에 이어 실제 구현함에 있어 KNN을 지원하는 파이썬 함수를 어떠한 식으로 설계하여야 할지 아키텍처를 생각해 이를 적용하는 과정을 거치게 되었다. CNN을 구현하는 과정에서는 백프로게이션을 하기 위한 레이어 구성을 어떠한 식으로 하여야 할지 구상하였고 약 0.2의 정확도에서 약 0.47의 정확도까지 끌어 올리 수 있었다. 하지만 이때 발생한 이슈는 min_faces_per_person을 20으로 두고 CNN을 실행하였을 시 오류가 발생하여 적절한 값을 넣으면서 찾던 중 100값을 입력하여 실행하였다. 만약 20으로 두고 fit적용부분의 어떤부분을 만져야 돌아갈지에 대한 생각이 부족하여 해결하지 못한 아쉬움이 남는다.