

인공지능Lab. Project #1

- PCA를 이용한 얼굴 인식

본 프로젝트에서는 PCA를 이용하여 얼굴 인식 과정을 시뮬레이션해 본다. 이 실습에서 사용한 얼굴 영상 DB는 LFW(Labeled Faces in the World), 얼굴 DB를 이용한다. 53명의 얼굴을 찍은 이미지를 담고 있다. 얼굴의 주성분들을 시각화 하여 확인해보고, PCA를 적용한 훈련 데이터로 학습시켜서 K-Nearest Neighbor(KNN) 알고리즘을 통해서 정확도를 측정해보자.

해당 Dataset을 이용하여 CNN(Convolutional neural network) 모델을 통해서 학습시켜보고 얼굴 이미지를 분류하여 비교해보자.



그림 1. 학습 얼굴 영상 집합

☐ Program implementation

1. LFW(Labeled Faces in the World) Dataset

LFW Dataset은 53개의 class의 데이터 2769로 구성되어 있고 각각의 얼굴 이미지는 87x65로 구성되어 있다. 다음 그림2와 같은 방식으로 Dataset을 load할 수 있다.

```
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_lfw_people

people = fetch_lfw_people(min_faces_per_person=20, resize=0.7)
image_shape = people.images[0].shape
```

그림 2. Dataset load Example

```
people.images.shape : (2769, 87, 65)
Class : 53
```

그림 3. Data shape

Alejandro Toledo	39
Alvaro Uribe	35
Amelie Mauresmo	21
Andre Agassi	36
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
Hamid Karzai	22
Hans Blix	39
Hugo Chavez	71
Igor Ivanov	20

• • •

그림 4. 각각의 인물에 대한 이미지의 개수

데이터셋에서 편중된 데이터가 있음을 확인하고 가장 작은 샘플 개수에 맞춰서 선택해서 사용해주시오. 그리고 데이터셋에 대한 설명을 해주세요

2. PCA whitening & PCA

픽셀을 비교할 때 얼굴 위치가 한 픽셀만 오른쪽으로 이동해 큰 차이를 만들어 다른 얼굴로 인식하게 됩니다. 주성분으로 변환하여 거리를 계산하여 정확도를 높일 수 있다. 화이트닝 옵션은 PCA 변환할 때 표준편차로 나누어 적용합니다. 또한, PCA 변환에서 데이터의 평균을 빼서 평균을 0으로 만들어 줍니다 ($\frac{x-mean}{std}$).

```
from sklearn.decomposition import PCA

# PCA 모델 생성 및 적용
pca = PCA(n_components=100, whiten=True, random_state=0) # 주성분 갯수, whitening option, 랜덤상태
```

그림 5. PCA library 사용 및 옵션 조정 Example

다음과 같은 방식으로 PCA 화이트닝 옵션을 적용하여 scatter plot을 그려 데이터의 조정할 수 있습니다. PCA를 하여 고유 얼굴 성분을 출력하여 보고 이후 KNN 모델을 학습시키는 과정에서 PCA Whitening의 여부를 통한 정확도의 차이를 비교하여 주세요. 그림 5에서처럼 옵션을 조정하여 주성분의 개수의 차이를 두어 얼굴 이미지를 재구성하여 그리고 결과를 설명해주세요.

3. KNN & CNN

LWF Dataset을 train과 test 데이터셋을 분류하여 train과 test 데이터에 PCA를 적용하여 KNN 모델을 생성하여 학습시켜 결과를 설명해주세요. (그림 6, 7, 8 참조).

```
# 전처리한 데이터를 분할
x_train, x_test, y_train, y_test = train_test_split(x_people_scaled, y_people, # 분할할 데이터
                                                    stratify=y_people, random_state=0) # 그룹화할 데이터, 랜덤상태
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split

import numpy as np
```

그림 6. Dataset split & KNN library load Example

그림 7. Dataset split Example

```
# 머신 러닝 라이브러리 import
knn = KNeighborsClassifier(n_neighbors=1) # 이웃의 수
knn.fit(x_train, y_train) # 모델 학습
```

그림 8. KNN Example

이 결과 비교를 위해 원본 LWF 데이터셋을 이용하여 딥러닝 알고리즘인 Convolutional neural network(CNN)를 학습시켜 결과를 비교해주세요. 임의의 CNN Network를 구성하여 그림 9, 10을 참고하여 학습하여 결과를 만들어 PCA를 통한 Feature를 추출한 방법과 성능을 비교하여 설명해주세요

```
import tensorflow as tf
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
import numpy as np
```

그림 9. Convolutional neural network Example

```

input_shape = (87, 65, 1)

batch_size = 128
num_classes = 53
epochs = 100

model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1, 1), padding='same', activation='sigmoid', input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(64, (2, 2), activation='sigmoid', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(32, activation='sigmoid'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='sigmoid'))
model.summary()

```

Model: "sequential_26"

Layer (type)	Output Shape	Param #
conv2d_45 (Conv2D)	(None, 87, 65, 32)	832
max_pooling2d_41 (MaxPooling)	(None, 43, 32, 32)	0
conv2d_46 (Conv2D)	(None, 43, 32, 64)	8256
max_pooling2d_42 (MaxPooling)	(None, 21, 16, 64)	0
dropout_40 (Dropout)	(None, 21, 16, 64)	0
flatten_20 (Flatten)	(None, 21504)	0
dense_39 (Dense)	(None, 32)	688160
dropout_41 (Dropout)	(None, 32)	0
dense_40 (Dense)	(None, 53)	1749
Total params: 698,997		
Trainable params: 698,997		
Non-trainable params: 0		

그림 9. Convolutional neural network Architecture Example

```

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
hist = model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, verbose=1)

score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])

```

그림 10. Convolutional neural network learning Example

□ 채점 기준

항목	점수
Dataset load와 데이터셋에 대한 설명	1
PCA를 통해서 고유 얼굴 성분 출력	2
PCA Whitening 사용	2
주성분의 개수의 차이를 두어 얼굴 이미지 재구성	2
KNN 모델을 생성, 학습시켜 결과 분석(data preprocessing, whitening 차이)	2
CNN 모델 학습, 결과 비교	1

□ 제출기한 및 제출방법

✓ 제출기한

- 2022년 10월 28일 23:59:58 까지 제출

✓ 제출 방법

- 소스코드와 보고서 파일(pdf)을 함께 압축하여 제출
- KLAS -> 과제 제출 -> 압축 파일 제출

✓ 제출 형식

- 파일 이름 : 학번_AI_project1

✓ 보고서 작성 형식

- 보고서 내용은 한글로 작성
- 보고서에는 소스코드를 포함하지 않음
- 채점 기준에 항목은 보고서 내용에 모두 포함되어야 합니다.
- 아래 각 항목을 모두 포함하여 작성
 - Introduction : 프로젝트 내용에 대한 설명
 - Algorithm : 프로젝트에서 이용된 Algorithms과 Method의 동작을 설명
 - Result : 결과 화면(그래프, 이미지)을 캡처하고 동작을 설명
 - Consideration : 고찰 작성