

내 얼굴 점수는 몇 점일까?

CNN을 활용한 무료 외모 평가 모델

1팀 일기예보
팀장 : 박영범
팀원 : 권승현, 김예진,
오병훈, 최상준

CNN을 활용한 무료 외모 평가 모델

프로젝트기간: 22.01.07~22.01.14

Work Team & Member

1팀 (일기예보)

팀장 : 박영범

팀원 : 권승현, 김예진, 오병훈, 최상준

Work Schedule

01.07 프로젝트 계획안 발표

01.10 데이터 수집 및 가공 완료

01.11 평가 모델 구축

01.13 오류 수정 및 시각화

01.14 프로젝트 발표

Work Dataset

SCUT-FBP5500 Database

CNN을 활용하여 AI가 평가해주는 나의 외모점수

평가된 데이터셋에서 특징을 추출 후 이미지를 5점 만점으로 평가하는 딥러닝 모델

Skills

Face Dataset



40%

OpenCV



20%

Deep Learning Model



40%



1

개요

주제, 목적

2

진행과정

일정, 역할분담

3

코드

라이브러리, 데이터, 코드

4

마무리

소감, 제한점(써라)



개요

주제

나의 외모 점수를 Ai에게 물어보자!

목적

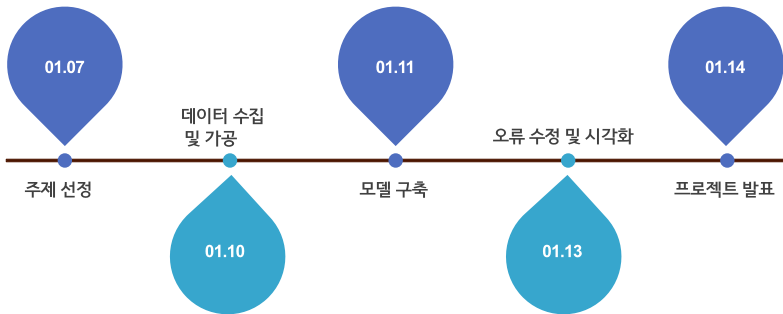
타인에게 직접 외모 평가를 받는 것은 부담스럽기에, Ai를 이용해서 부담 없이 받아보자.



진행과정

일정

22.01.07 ~ 22.01.14





역할분담

박영범
데이터 수집
및 가공

권승현
데이터 수집
및 가공

김예진
모델 구축

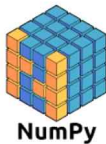
오병훈
오류 수정

최상준
PPT & 발표



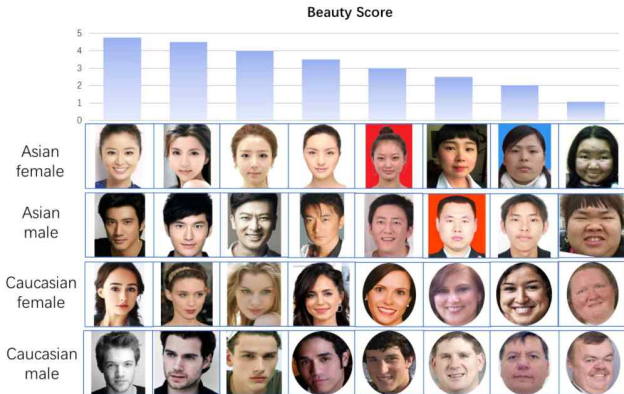
코드

라이브러리



SCUT-FBP5500 Database

- 남중국 공과대학 lab에서 배포한 Dataset
- Dataset은 4000개의 동양 남녀 (AM,AF)의 정면 사진, 1500 백인 남녀(CM,CF)의 정면 사진을 포함함
- Dataset의 모든 이미지를 60명의 자원 봉사자들이 1~5점으로 평가



Read and Analyze Labels

이미지 파일과 평균 점수

labels.head()

		0	1
0	CF437.jpg	2.883333	
1	AM1384.jpg	2.466667	
2	AM1234.jpg	2.150000	
3	AM1774.jpg	3.750000	
4	CF215.jpg	3.033333	

labels.describe()

	1
count	5500.000000
mean	2.990891
std	0.688112
min	1.016667
25%	2.500000
50%	2.833333
75%	3.533333
max	4.750000

Split Train and Validation Dataset

Train : Test = 6 : 4

```
1 x_train, x_val, y_train, y_val = train_test_split(imgs, labels_np[:, 1], test_size=0.4)
2 np.save('x_train.npy', x_train)
3 np.save('x_val.npy', x_val)
4 np.save('y_train.npy', y_train)
5 np.save('y_val.npy', y_val)
6 print(x_train.shape, x_val.shape)
7 print(y_train.shape, y_val.shape)
```



Plot Sample

```
plt.figure(figsize=(16, 6))  
for i, img in enumerate(x_train[:10]):  
    plt.subplot(2, 5, i+1)  
    plt.axis('off')  
    plt.title('%2f' % y_train[i])  
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
    plt.imshow(img)
```



Data Augmentation

```
train_datagen = ImageDataGenerator(  
    rescale=1./255,  
    rotation_range=30,  
    shear_range=0.2,  
    zoom_range=0.4,  
    horizontal_flip=True  
)  
  
val_datagen = ImageDataGenerator(  
    rescale=1./255  
)  
  
train_generator = train_datagen.flow(  
    x=x_train, y=y_train,  
    batch_size=32,  
    shuffle=True  
)  
  
val_generator = val_datagen.flow(  
    x=x_val, y=y_val,  
    batch_size=32,  
    shuffle=False  
)
```





Create Model

```
inputs = Input(shape=(350, 350, 3))

net = Conv2D(32, kernel_size=3, strides=1, padding='same')(inputs)
net = LeakyReLU()(net)
net = MaxPooling2D(pool_size=2)(net)

net = Conv2D(32, kernel_size=3, strides=1, padding='same')(net)
net = LeakyReLU()(net)
net = MaxPooling2D(pool_size=2)(net)

net = Conv2D(32, kernel_size=3, strides=1, padding='same')(net)
net = LeakyReLU()(net)
net = MaxPooling2D(pool_size=2)(net)

net = Conv2D(32, kernel_size=3, strides=1, padding='same')(net)
net = LeakyReLU()(net)
net = MaxPooling2D(pool_size=2)(net)

net = Flatten()(net)
```

```
net = Dense(256)(net)
net = Activation('relu')(net)
net = Dense(256)(net)
net = Activation('relu')(net)
net = Dense(128)(net)
net = Activation('relu')(net)
net = Dense(64)(net)
net = Activation('relu')(net)
net = Dense(32)(net)
net = Activation('relu')(net)
net = Dense(16)(net)
net = Activation('relu')(net)
net = Dense(1)(net)
outputs = Activation('linear')(net)
```

```
model = Model(inputs=inputs, outputs=outputs)
```

```
model.compile(optimizer='adam', loss='mae')
```

```
model.summary()
```

Train

```
import h5py

start_time = datetime.datetime.now().strftime('%Y_%m_%d_%H_%M_%S')

model.fit_generator(
    train_generator, epochs=10, validation_data=val_generator,
    callbacks=[
        ModelCheckpoint('models/%s.h5' % (start_time), monitor='val_loss')
    ]
)
```

Evaluate Validation Dataset

```
model = load_model('/content/drive/MyDrive/beauty_score')

val_data = val_generator.__getitem__(0)

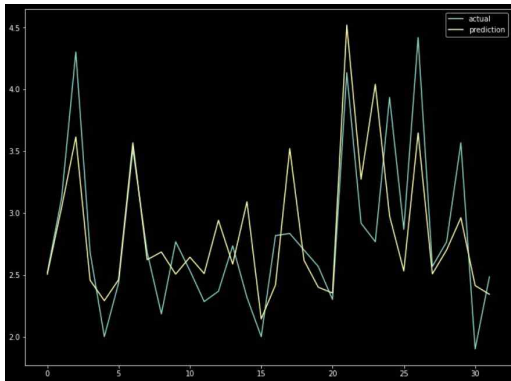
preds = model.predict(val_data[0])

plt.figure(figsize=(16, 8))
for i, img in enumerate(val_data[0]):
    plt.subplot(4, 8, i+1)
    plt.title('%0.2f / %0.2f' % (preds[i], val_data[1][i]))
    plt.axis('off')
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(img.squeeze())
```

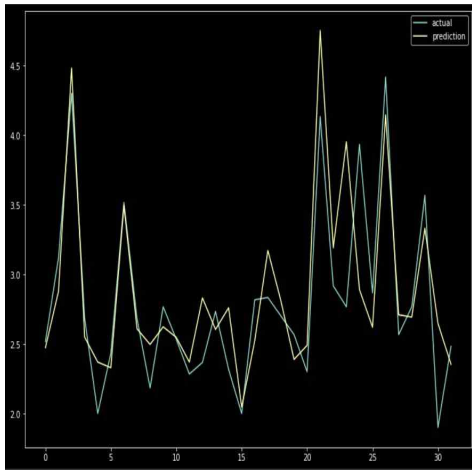


Visualize

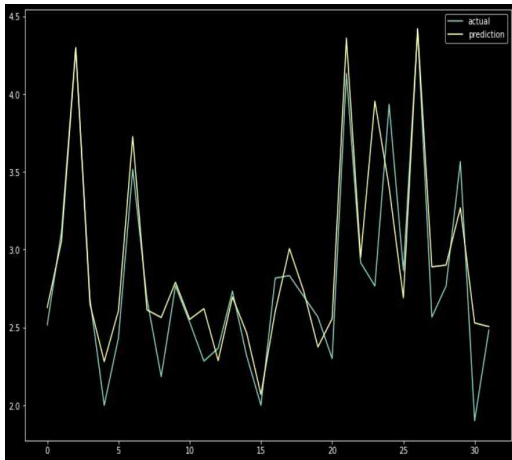
```
1 plt.figure(figsize=(12, 9))
2 plt.plot(np.asarray(y_val)[:32], label='actual')
3 plt.plot(preds, label='prediction')
4 plt.legend()
5 plt.show()
```



2차수정 모델 시각화 그래프



최종 모델 시각화 그래프



3.76



3.72



2.62



2.82



3.13



2.84



2.81



4.31



2.80



4.03



4.08



3.10



The background is a dark, grayscale photograph of a Ferris wheel, likely the London Eye, with its spokes and capsules visible. In the foreground, there are silhouettes of trees and a structure that looks like a pagoda or a traditional Korean roof. Overlaid on this is a large, light blue diamond shape with a white border. Inside the diamond, the Korean word '마무리' is written in white.

마무리

소감

저번 프로젝트에 비해 주제가 재밌어서 즐겁게 참여할 수 있었다. 항상 열심히 해 주시는 팀원분들께 진심으로 감사하다. 다음 프로젝트 역시 활기차게 진행할 수 있으면 좋겠다.

저번 프로젝트와는 다르게 이미지 분석을 활용 하면서 CNN에 대해 더 공부하고 잘 알게 되었다. 항상 의지 되는 팀원들에게 고맙다.

차후에 있을 프로젝트의 사전체험으로 생각하며 팀원들과의 소통과 협업 능력이 향상되었다고 생각하며 유익한 시간이었던 것 같다.

박영범

권승현

김예진

오병훈

최상준

팀 프로젝트로 얻은 것이 있다면 아마도 여러분이 아닐까 싶다. 협동심은 한계점이 없다! Thank you!

처음에는 딥러닝 관련 기술이 다소 어려웠지만, 조금씩 알아가는 것 같아서 뿌듯하다.



Thank you.