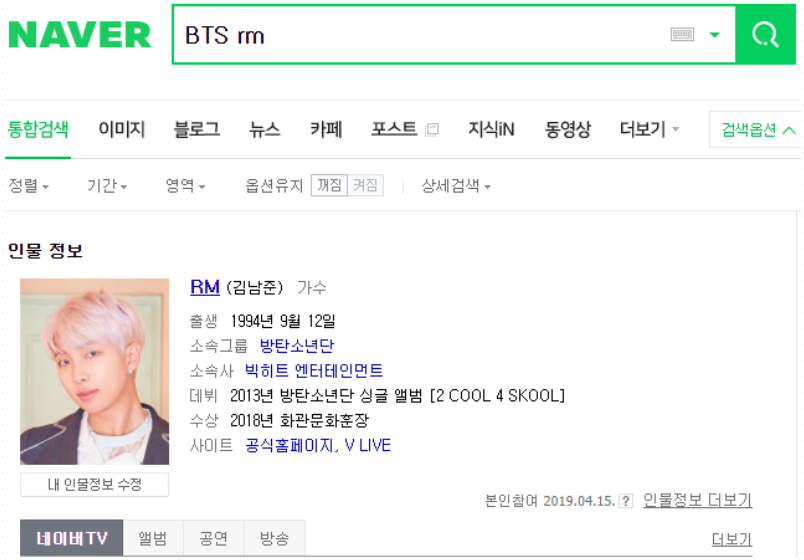

NUGUDEORA...?

NUGUDEORA


(멋 짐 폭 발)
뷔의 트레이드마크 브이 ↑

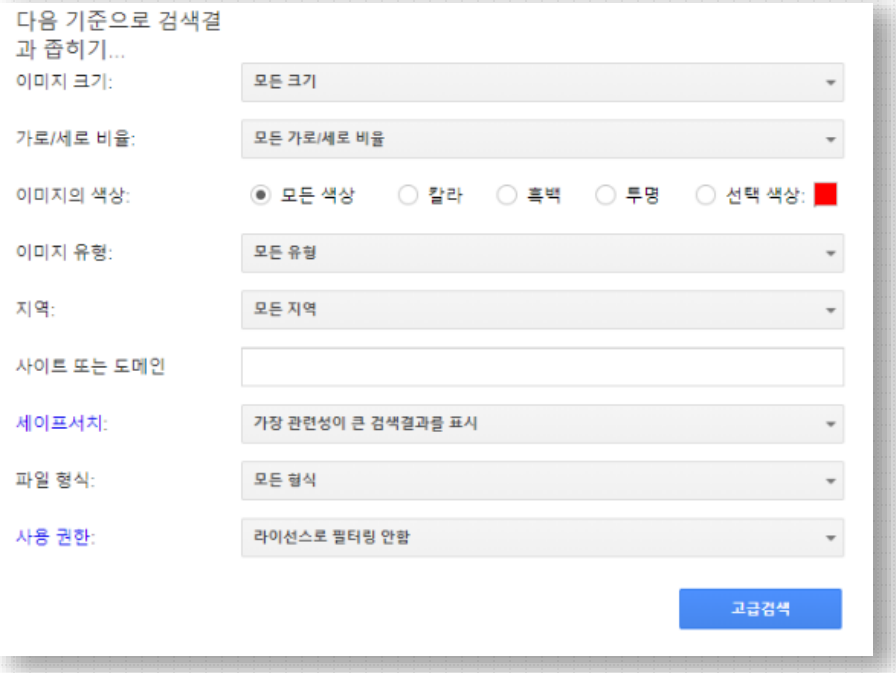

발)
BTS RM
의 트레이드마크 브이 ↑


NAVER
BTS rm
통합검색 이미지 블로그 뉴스 카페 포스트 지식iN 동영상 더보기
검색옵션
정렬 기간 영역 옵션유지 꺼짐 켜짐 상세검색
인물 정보
RM (김남준) 가수
출생 1994년 9월 12일
소속그룹 방탄소년단
소속사 빅히트 엔터테인먼트
데뷔 2013년 방탄소년단 싱글 앨범 [2 COOL 4 SKOOL]
수상 2018년 화관문화훈장
사이트 공식홈페이지, V LIVE
내 인물정보 수정
본인참여 2019.04.15.
인물정보 더보기
네이버TV 앨범 공연 방송
더보기

NUGUDEORA

1. 데이터 수집

GoogleImageCrawler




```python
print("원하는 그림의 키워드를 넣어주세요: ")
kw=input()
```

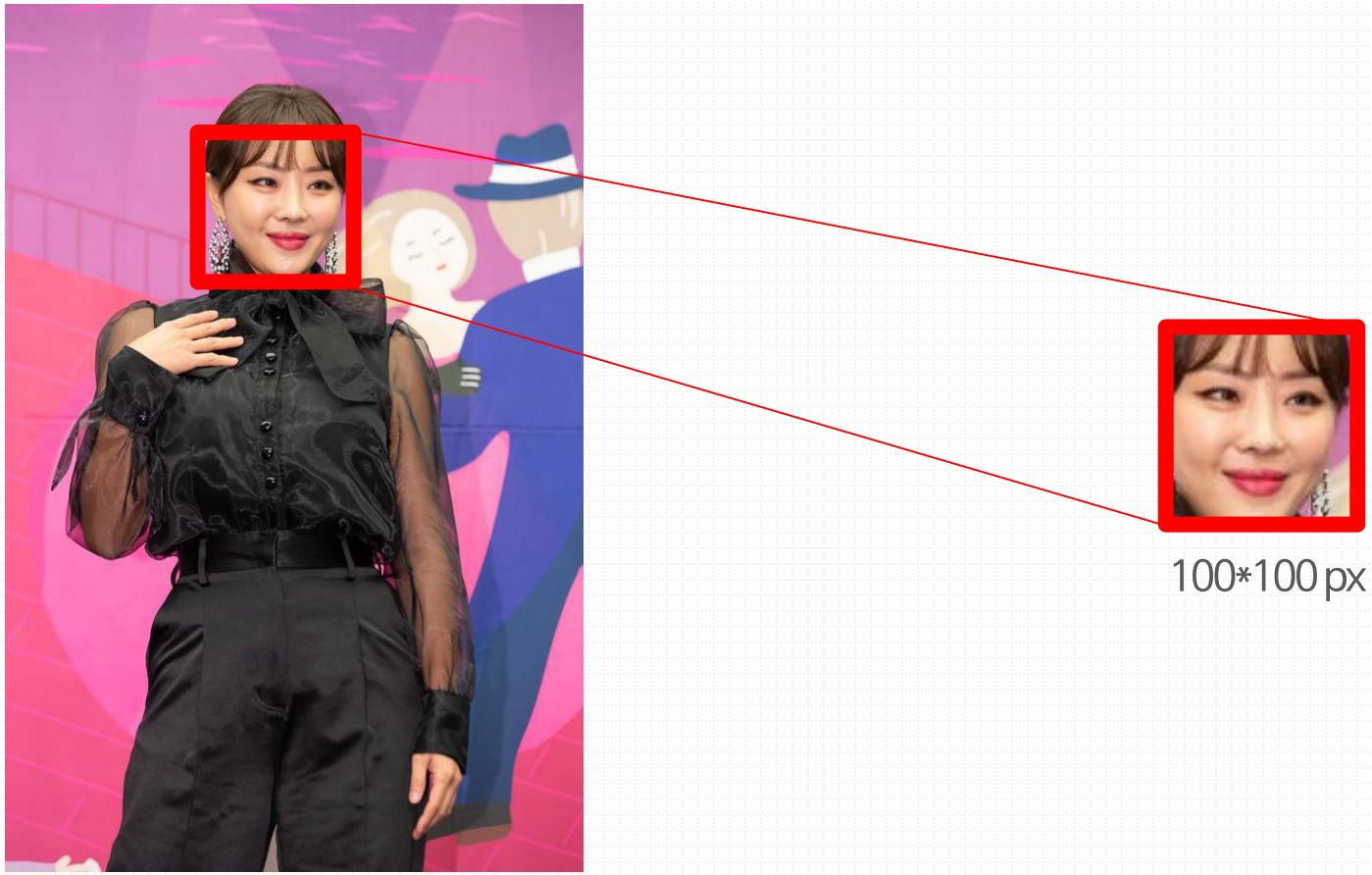
원하는 그림의 키워드를 넣어주세요:

나르샤

```python
google_crawler=GoogleImageCrawler(
    feeder_threads=1,
    parser_threads=1,
    downloader_threads=4,
    storage={'root_dir':download_dir})



filters=dict(size='small', #이미지 크기
            color='color',#이미지의 색깔을 필터링
            license='commercial,modify',
             #검색하고자 하는 날짜의 범위
            date=((2017,1,1),(2017,12,31)))

google_crawler.crawl(keyword=kw1,
                     filters=filters,
                     max_num=50,
                     file_idx_offset=0)
```

1. 데이터 수집

Face ROI



1. 데이터 수집

Face ROI

```python
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
#폴더 내의 이미지 파일명 리스트 반환
file_list = os.listdir(download_dir)

#각 이미지 얼굴만 추출저장
for idx, f_name in enumerate(file
    try:
        print('processing: ' + str(idx))
        #file_type = re.findall('[.]WS+', f_name)[0]
        f_name1 = f_name
        f_name = download_dir + '/' + f_name
        img = my_imread(f_name)
        gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 7)
        if type(faces) == tuple:
            continue
        tmp_px=20
        h_tmp_px=tmp_px//2
        for (x,y,w,h) in faces:
            img = cv2.rectangle(img,(x-h_tmp_px,y-h_tmp_px),(x+w+h_tmp_px,y+h+h_tmp_px),(255,0,0),2)
            roi_gray = gray[y:y+h, x:x+w]
            roi_color = img[y:y+h, x:x+w]
        roi_color=cv2.resize(roi_color,(100,100),interpolation=cv2.INTER_LINEAR)
        save_dir = download_dir + '_face'
        w_name = save_dir + '/' + f_name1
        if not os.path.exists(save_dir):
            os.makedirs(save_dir)
        my_imwrite(w_name, roi_color)
    except:
        pass
print("done")
```

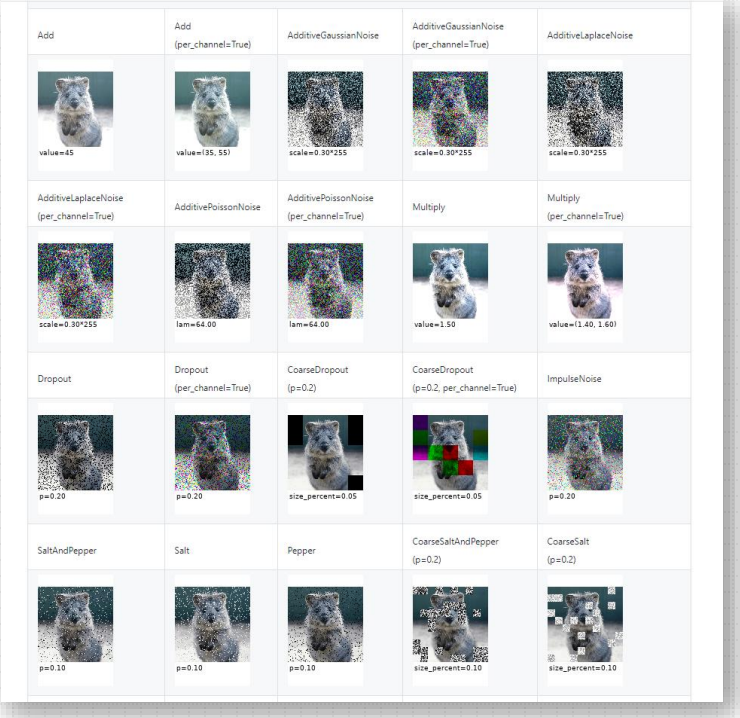
```python
faces = face_cascade.detectMultiScale(gray, 1.3, 7)
```

1. 데이터 수집

Face ROI

2. 데이터 전처리

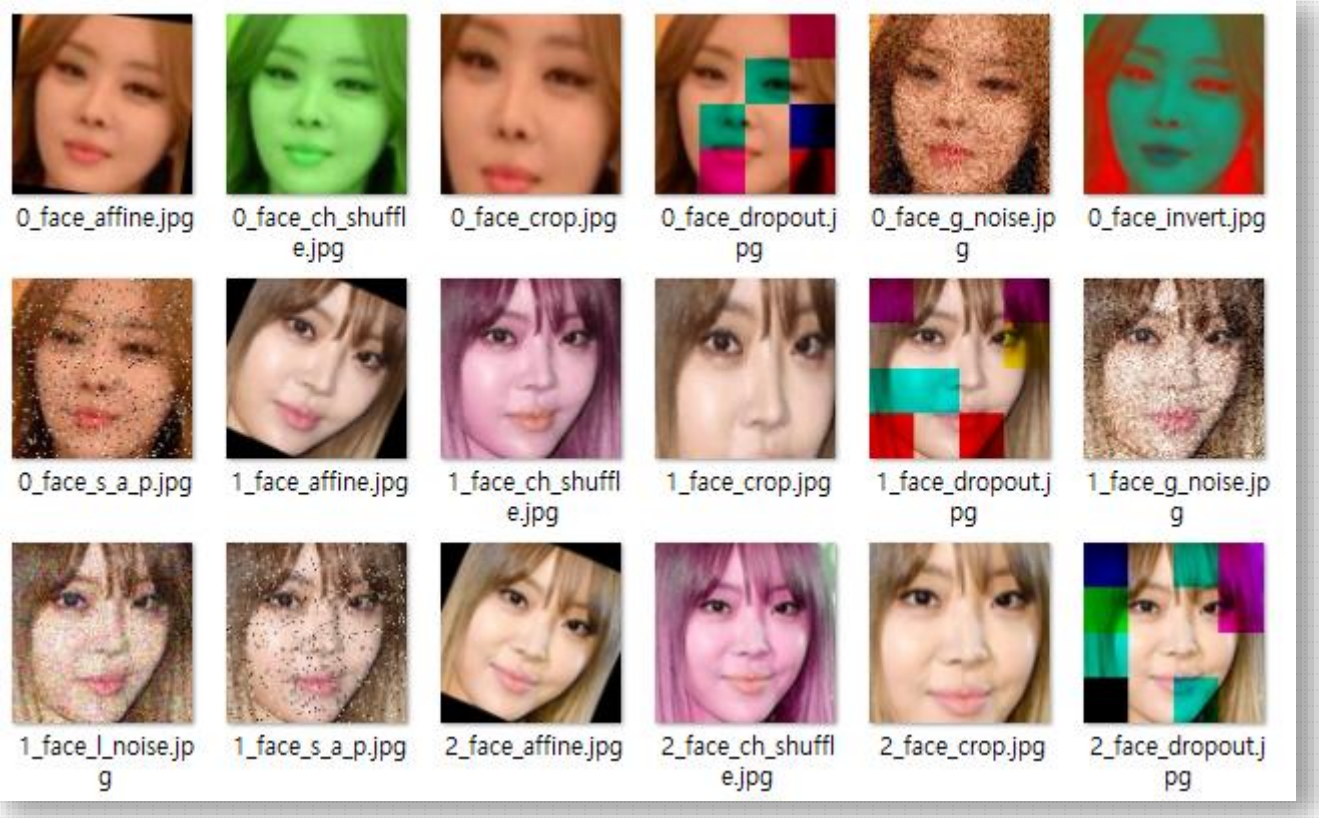
Data Argumentation




```python
# Affine
seq = iaa.Sequential([iaa.Affine(rotate=(-25, 25))], random_order=True)
images_aug.append(seq.augment_image(image))

# Gaussian Noise
seq = iaa.Sequential([iaa.AdditiveGaussianNoise(scale=(30, 50))], random_order=True)
images_aug.append(seq.augment_image(image))

# Crop
seq = iaa.Sequential([iaa.Crop(percent=(0, 0.3))], random_order=True)
images_aug.append(seq.augment_image(image))

# Laplacian Noise
seq = iaa.Sequential([iaa.AdditiveLaplaceNoise(scale=0.1*255, per_channel=True)], random_order=True)
images_aug.append(seq.augment_image(image))

# Coarse Dropout
seq = iaa.Sequential([iaa.CoarseDropout(p=0.2, size_px=1, per_channel=True)], random_order=True)
images_aug.append(seq.augment_image(image))

# Invert
seq = iaa.Sequential([iaa.Invert(per_channel=True, p=0.6)], random_order=True)
images_aug.append(seq.augment_image(image))

# Salt and Pepper
seq = iaa.Sequential([iaa.SaltAndPepper(p=0.1)], random_order=True)
images_aug.append(seq.augment_image(image))

# Channel Shuffle
seq = iaa.Sequential([iaa.ChannelShuffle(p=1.0)], random_order=True)
images_aug.append(seq.augment_image(image))
```

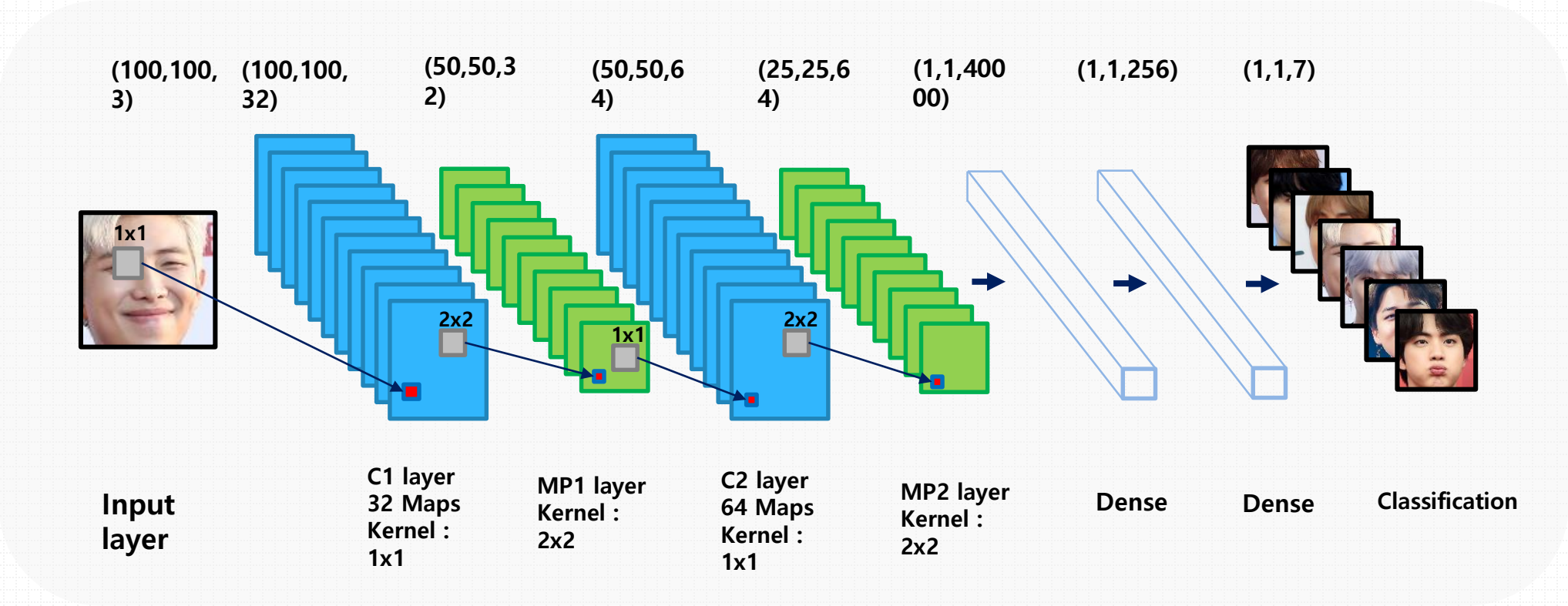
2. 데이터 전처리

Data Argumentation



3. 데이터 학습

Model



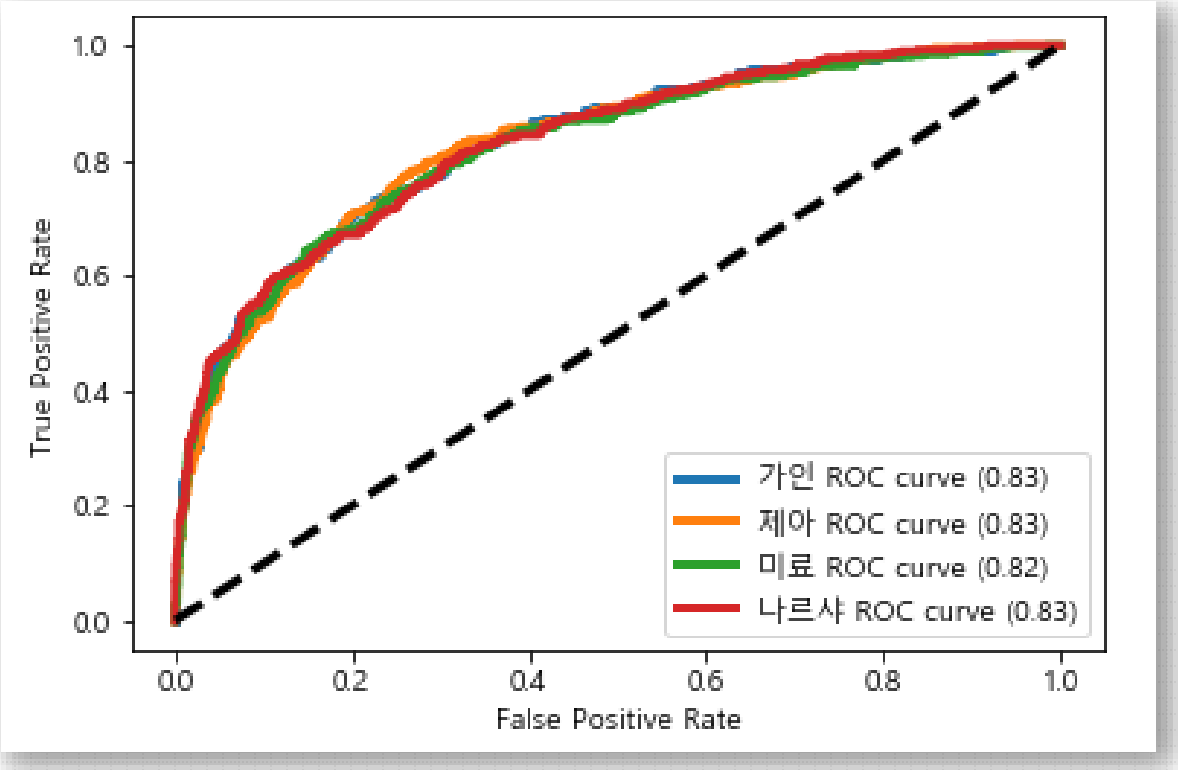
3. 데이터 학습

Sequential

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 100, 100, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 50, 50, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 50, 50, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 50, 50, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 25, 25, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 25, 25, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 40000)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               10240256
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 9)                 2313
=================================================================
Total params: 10,261,961
Trainable params: 10,261,961
Non-trainable params: 0
_________________________________________________________________
```

3. 데이터 학습

ROC curve



4. 모델 적용

저장된 모델 불러오기

```
# 학습된 모델 불러오기 (****수정하세요****)
model = joblib.load('BTS_classification_model2.pkl')

# (****categoties만 수정하세요****)
categories = ["브아걸_제아", "브아걸_나르샤", "브아걸_미료", "브아걸_가인"]
roi_color = [(255, 255, 255), (0, 255, 255), (255, 0, 255), (255, 255, 0),
             (0, 0, 255), (0, 255, 0), (255, 0, 0)]

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
fontpath = "fonts/gulim.ttc"
font = ImageFont.truetype(fontpath, 20)

count=0
frame_count=0

# (****경로 수정하세요****)
cap=cv2.VideoCapture('img/Brown_Eyed_Grils_mv.mp4')
```

4. 모델 적용

지정한 영상에 모델 적용

```
# 학습된 모델 불러오기 (****수정하세요****)
model = joblib.load('BTS_classification_model2.pkl')

# (****categoties만 수정하세요****)
categories = ["브아걸_제아", "브아걸_나르샤", "브아걸_
roi_color = [(255, 255, 255), (0, 255, 255), (2
             (0, 0, 255), (0, 255, 0), (255, 0,

face_cascade = cv2.CascadeClassifier('haarcasca
fontpath = "fonts/gulim.ttc"
font = ImageFont.truetype(fontpath, 20)

count=0
frame_count=0

# (****경로 수정하세요****)
cap=cv2.VideoCapture('img/Brown_Eyed_Grils_mv.m
```

```
tmp_px=20
h_tmp_px=tmp_px//2
while True:
    is_ok, frame=cap.read()
    roi=[]
    if not is_ok:
        break
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    if len(faces)!=0:
        faces_n,_=faces.shape
        for (x,y,w,h) in faces:
            #frame = draw_face(frame, model, w, y, w, h)
            #roi_gray = gray[y:y+h, x:x+w]
            roi = frame[y:y+h, x:x+w]
            roi=cv2.resize(roi,(100,100),interpolation=cv2.INTER_LINEAR)
            pred_face=roi.reshape(1,100,100,3)
            i = np.argmax(model.predict(pred_face))

            frame = cv2.rectangle(frame,(x-h_tmp_px,y-h_tmp_px),
                                  (x+w+h_tmp_px,y+h+h_tmp_px),roi_color[i],2)
            frame_pil = Image.fromarray(frame)
            draw = ImageDraw.Draw(frame_pil)
            draw.text((x,y+h+40), categories[i], font=font, fill=roi_color[i])
            frame = np.array(frame_pil)
#             frame=cv2.putText(frame,categories[i],(x,y+h+40)
#                     ,cv2.FONT_HERSHEY_PLAIN,2,roi_color[i],4)
```

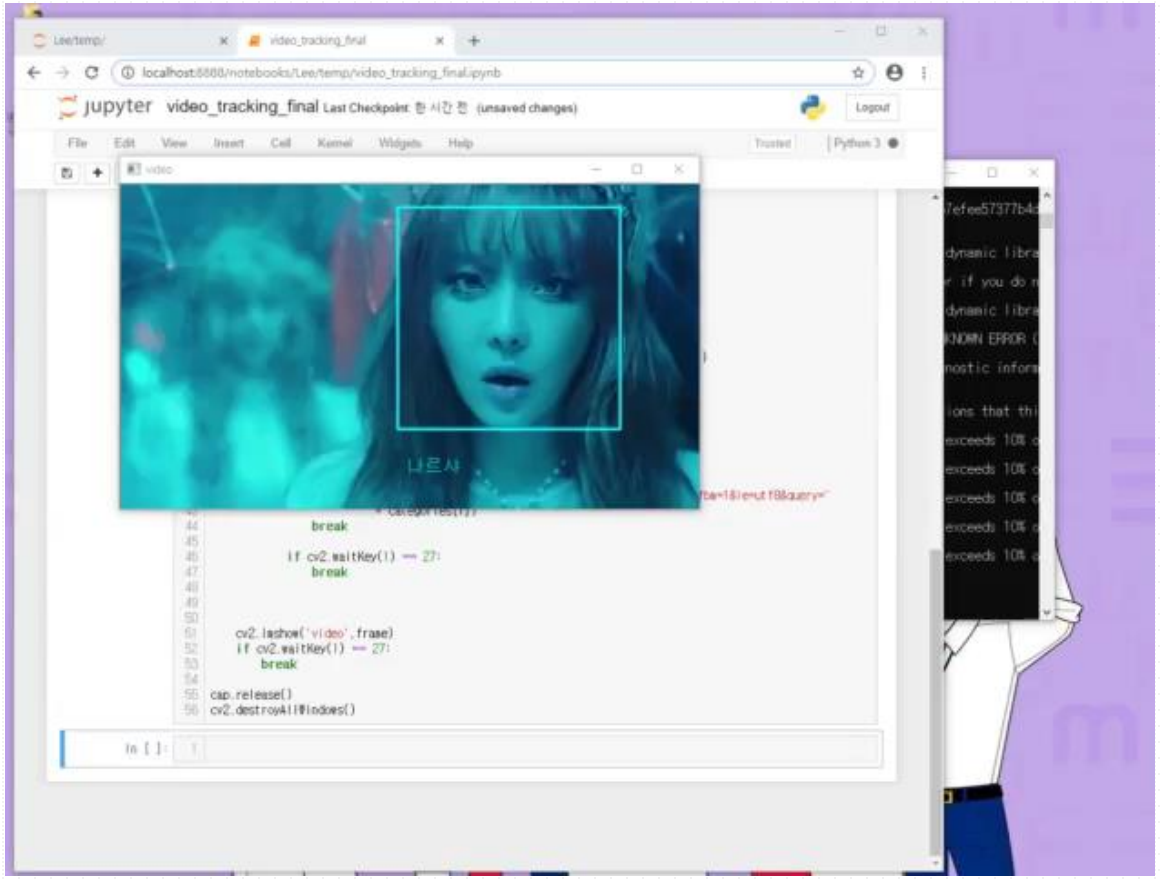
4. 모델 적용

'Enter' 키를 통해 원하는 인물 검색

```
key = cv2.waitKey(1)
if key == 13: # enter
    driver = webdriver.Chrome('chromedriver.exe')
    driver.implicitly_wait(1)
    driver.get('https://search.naver.com/search.naver?sm=top_hty&fbm=1&ie=utf8&query='
               + categories[i])
    break

if cv2.waitKey(1) == 27:
    break
```

4. 모델 적용

'Enter' 키를 통해 원하는 인물 검색



4. 모델 적용

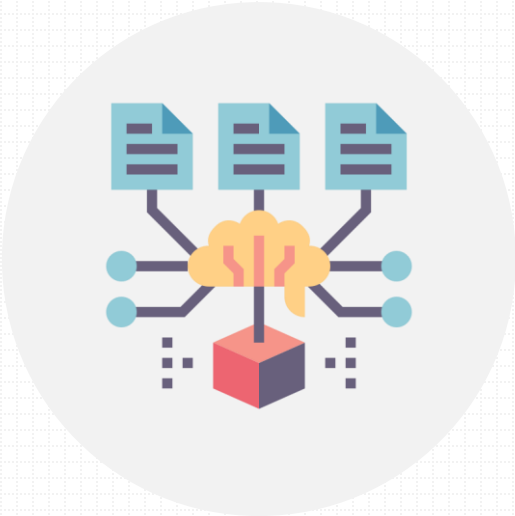
'Enter' 키를 통해 원하는 인물 검색

5. 개선점

개선해야 할 2가지 문제

이미지 데이터 수집

다양한 표정을 가진 사진 데이터가
학습시키기에 부족함



최적화된 모델 선정

임의의 모델을 만들어서 적용했기 때문에
최적화된 모델을 찾아야 함

감사합니다.