

POSE

카카오 포즈 API

5조 김현주, 신민경, 길정우

목차

01

기능 소개

02

코드 설명

03

시연

POSE API

[이미지 분석 API]

1. 이미지 내의 사람을 인식

2. 사람의 코, 눈, 귀, 관절(어깨, 골반 발목 등)

총 17개의 키 포인트 추출

3. 사람의 자세(Pose)를 분석

02 코드 설명

```
Anaconda Prompt (anaconda3)

(base) C:\Users\SEM>pip uninstall opencv-python
WARNING: Skipping opencv-python as it is not installed.

(base) C:\Users\SEM>pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.5.1.48-cp38-cp38-win_amd64.whl (34.9 MB)
    |████████████████████| 34.9 MB 3.3 MB/s
Requirement already satisfied: numpy>=1.17.3 in d:\anaconda3\lib\site-packages (from opencv-python) (1.19.2)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.1.48

(base) C:\Users\SEM>_
```

[Anacinda Install]

1. pip uninstall opencv-python
2. pip install opencv-python
3. pip install pycocotools-windows

```
#다차원 배열 처리
import numpy as np
```

```
#데이터 시각화
```

```
from matplotlib import image as mpimg
import matplotlib.pyplot as plt
```

```
#물체 감지 : 객체 인식 및 검출에 사용
from pycocotools.coco import COCO
```

```
from requests import Session
```

```
#영상 처리 라이브러리
import cv2
```

```
#파일 주소
```

```
IMAGE_FILE_PATH = 'D:\workspace_python\pose\pose\kim02.jpg'
result = inference(IMAGE_FILE_PATH) #위의 파일을 inference에 매개변수로 줌
visualize(IMAGE_FILE_PATH, result) #프즈 구현한 사진 보여주기
```

```
APP_KEY = '6c034887e3f423e07a8108edcfcf1094' #kakao developers에서 발급받은 REST API키
#입력한 APP_KEY를 카카오 서버에 전달 후 사용자 식별
session = Session()
session.headers.update({'Authorization': 'KakaoAK ' + APP_KEY})
```

```
def inference(filename): #파일 받음
    with open(filename, 'rb') as f:
        #받은 파일을 카카오 디비로 가져감
        response = session.post('https://cv-api.kakaobrain.com/pose', files={'file': f})
        response.raise_for_status() #에러인지 아닌지 판별
        return response.json()
```



```

def visualize(filename, annotations, threshold=0.2):
    # Annotation : 이미지에 있는 사람/사람의 분할 영역과 카테고리 등의 정보
    for annotation in annotations:

        # keypoints : 이미지에서 검출된 17개의 키 포인트의 좌표(x, y)와 정확도(score)
        keypoints = np.asarray(annotation['keypoints']).reshape(-1, 3)

        #정확도가 낮은 점을 0으로 변환
        low_confidence = keypoints[:, -1] < threshold
        keypoints[low_confidence, :] = [0, 0, 0]

        annotation['keypoints'] = keypoints.reshape(-1).tolist()

    #데이터 시각화
    plt.imshow(mping.imread(filename))
    #x,y축 범위를 설정하지 않음
    plt.axis('off')
    coco = COCO()
    coco.dataset = {
        "categories": [
            {
                #supercategory, id, name은 고정
                "supercategory": "person",
                "id": 1,
                "name": "person",
                #이미지에서 검출된 17개의 키 포인트의 좌표(x, y)와 정확도(score)
                "keypoints": ["nose", "left_eye", "right_eye", "left_ear", "right_ear", "left_shoulder",
                             "right_shoulder", "left_elbow", "right_elbow", "left_wrist", "right_wrist", "left_hip",
                             "right_hip", "left_knee", "right_knee", "left_ankle", "right_ankle"],
                #skeleton : 연결된 두 키 포인트 ID들을 담은 목록 예: [1,2]는 코와 왼쪽 귀를 연결한 선을 의미함
                "skeleton": [[1, 2], [1, 3], [2, 3], [2, 4], [3, 5], [4, 6], [5, 7], [6, 7], [6, 8], [6, 12], [7, 9],
                             [7, 13], [8, 10], [9, 11], [12, 13], [14, 12], [15, 13], [16, 14], [17, 15]]
            }
        ]
    }
    coco.createIndex()
    coco.showAnns(annotations)

    #구현한 파일을 이미지로 저장
    plt.savefig('kim02.jpg')
    #이미지를 읽음
    img = cv2.imread('kim02.jpg', cv2.IMREAD_UNCHANGED)
    #이미지를 화면에 보임
    cv2.imshow('Result', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

```

THANK YOU
