

Chapter

1

Keras를 이용한 사진분류

- Python + Google Custom Search API
- Python + OpenCV
- Python + Keras
- Python + Django

← → ↺ ⓘ 127.0.0.1:8000

앱 MyEtherWallet.com Free Stock Charts,... IntelliJ IDEA を使っ... 【5分でわかる】heroku... 동아비즈니스리뷰 Facebook Messeng...

keras를 이용한 딥러닝

좋아하는 연예인을 분류해 보자.

판정할 이미지를 선택해 주세요.

파일 선택

선택된 파일 없음

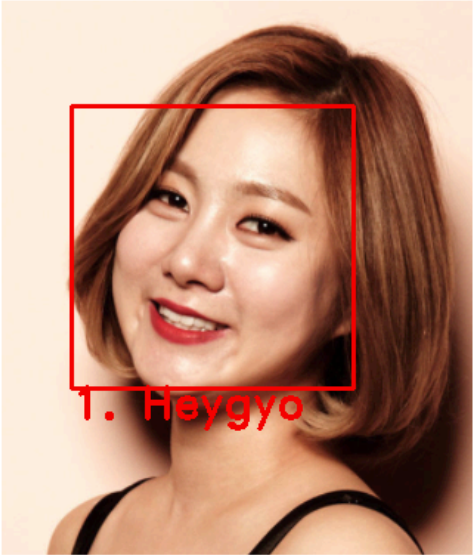
분석

판정 이미지

- predict_2.jpg

판정 결과

1. 전지현일 가능성:0.000% / 송혜교일 가능성:100.000%



1. Heygyo

Python과 Google Custom Search API

이미지 다운로드



Google Search API 사용

- Google Cloud Platform (<https://console.cloud.google.com>)
 - 신규 프로젝트 생성
 - API 및 서비스
 - Custom Search API
 - 인증 등록 (API Key)

Google Cloud Platform console interface showing the 'API 및 서비스' (APIs & Services) page.

API 및 서비스

- 대시보드
- 라이브러리
- 사용자 인증 정보**
- OAuth 동의 화면
- 도메인 확인
- 페이지 사용 동의

사용자 인증 정보 + 사용자 인증 정보 만들기 삭제

사용 설정한 API에 액세스하려면 사용자 인증 정보를 만드세요. [자세히 알아보기](#)

⚠ 나와 내 사용자가 보호받으려면 동의 화면과 애플리케이션이 Google의 확인을 거쳐야 합니다. [자세히 알아보기](#)

API 키

<input type="checkbox"/>	이름	생성일 ↓	제한사항	키
<input type="checkbox"/>	⚠ API 키 1	2020. 1. 20.	없음	AIzaSyCHqp...na7Y9oyIcw

Google Search API 사용

- Custom Search Engine 추가 (<https://cse.google.com/cse/all>)
 - 검색 엔진 추가
 - 검색 할 사이트 등록 → www.google.co.kr
 - 검색 엔진 ID
 - 이미지 검색, 전체 웹 검색 → ON

Custom Search

New search engine

▼ Edit search engine

All

► Help

Visit Help Forum (Ask a question)

Send Feedback

Edit search engines

Add Delete

Search engines	Edition	Is owner?	Public URL
<input type="checkbox"/> Google	Free	Yes	GO

```
(venv)> pip install pylint
(venv)> pip install requests
(venv)> pip install python-dotenv
(venv)> pip install google-api-python-client
```

- API_KEY=*** 
CUSTOM_SEARCH_ENGINE= 

이미지 다운로드 프로그램

- img_download_gcs.py

```
# 지정한 키워드로 검색한 이미지의 URL 얻기
def get_image_urls(keyword, total_num):
```

```
# 이미지 URL을 기본으로 해서 이미지 다운로드
def get_image_files(dir_path, keyword_count, image_urls):
```

```
# 이미지 다운로드
def download_image(url):
```

```
# 이미지를 파일로 저장
def save_image(filename, image):
```

```
# 디렉토리, 디렉토리 내 파일 삭제
def delete_dir(dir_path, is_delete_top_dir=True):
```

```
def main():
    print("=====")
    print("Image Downloader - Google Customr Search API")
    print("지정한 키워드로 검색된 이미지 파일을 다운로드")
    print("=====")
```

이미지 다운로드 프로그램

- settings.py

```
import os
from os.path import join, dirname
from dotenv import load_dotenv

dotenv_path = join(dirname(__file__), 'settings_1.env')
load_dotenv(dotenv_path)

API_KEY = os.environ.get("API_KEY")
CUSTOM_SEARCH_ENGINE = os.environ.get("CUSTOM_SEARCH_ENGINE")
```

- 프로그램 실행

```
(venv)> python img_download_gcs.py "다운로드할 이미지"
```


Python과 OpenCV

이미지파일에서 얼굴검출



얼굴검출 프로그램

- 라이브러리 설치

```
(venv)> pip install python-dotenv  
(venv)> pip install opencv-python
```

- 설정 파일 추가 → settings.env

CASCADE_FILE_PATH="{작업경로}\haarcascades\haarcascade_frontalface_default.xml"

얼굴검출 프로그램

- ***Haar Cascade***

- 머신러닝 기반 오브젝트 검출 알고리즘
- 2001년 논문 "Rapid Object Detection using a Boosted Cascade of Simple Features"에서 Paul Viola와 Michael Jones가 제안한 특징(feature)을 기반으로 비디오 또는 이미지에서 오브젝트를 검출하기 위해 사용되는 알고리즘
- 사각형 영역으로 구성되는 특징을 사용하기 때문에 픽셀을 직접 사용할 때 보다 동작 속도가 빠름

- 1) Haar Feature Selection
- 2) Creating Integral Images
- 3) Adaboost Training
- 4) Cascading Classifiers

얼굴검출 프로그램

- img_face_detect.py

```
def load_name_images(image_path_pattern):
```

```
def detect_image_face(file_path, image, cascade_filepath):
```

```
def delete_dir(dir_path, is_delete_top_dir=True):
```

```
def main():  
    print("=====")  
    print("이미지 얼굴인식 OpenCV 이용")  
    print("지정한 이미지 파일의 정면얼굴을 인식하고, 64x64 사이즈로 변경")  
    print("=====")
```

얼굴검출 프로그램

- settings.py

```
import os
from os.path import join, dirname
from dotenv import load_dotenv

dotenv_path = join(dirname(__file__), 'settings_2.env')
load_dotenv(dotenv_path)

CASCADE_FILE_PATH = os.environ.get("CASCADE_FILE_PATH")
```

- 프로그램 실행

```
(venv)> python img_face_detect.py
```

Python과 OpenCV

이미지파일 수 증가



얼굴검출 프로그램

- img_data_generator.py

```
def load_name_images(image_path_pattern):
```

```
def scratch_image(image, use_flip=True, use_threshold=True, use_filter=True):
```

```
def delete_dir(dir_path, is_delete_top_dir=True):
```

```
def main():  
    print("=====")  
    print("이미지 증가를 위한 OpenCV 이용")  
    print("지정한 이미지 파일의 수를 증가 (Flip, 임계값 등의 작업으로 8배 증가)")  
    print("=====")
```

- 프로그램 실행

```
(venv)> python img_data_generator.py
```