

카카오 API - 비전

0.사전 설정

- 테스트환경 python 3.8 IDLE
- 테스트 전 준비사항

CMD 창에서 `pip install requests`
 `pip install pillow`

먼저 설치해 주어야함.

구현 예제

간단하게 Python을 이용해 비전 API를 활용할 수 있는 예제를 소개합니다. Python 버전은 2.7 또는 3.6 이상을 권장합니다.
설치가 필요한 Python module의 요구사항은 아래와 같습니다.

```
requests==2.14.2  
Pillow==5.0.0
```

작성된 코드들을 .py의 파이썬 파일로 만들어 IDLE에서 돌리면 됩니다.

- API 키 발급은 <https://developers.kakao.com/> -> 내 어플리케이션 을 통해 발급 받으실 수 있습니다.

1. 모자이크 처리하기

- 입력 – 이미지 파일



C:/Users/memen/OneDrive/바탕 화면/
인턴 - 무지개연구소/아이유.jpg

```

import sys
import argparse
import requests
from PIL import Image, ImageFilter

API_URL = 'https://kapi.kakao.com/v1/vision/face/detect'
MYAPP_KEY = '발급받은 키를 입력해주세요!' #발급받은 인증 키

def detect_face(filename):
    headers = {'Authorization': 'KakaoAK {}'.format(MYAPP_KEY)}

    try:
        files = { 'file' : open(filename, 'rb')}
        resp = requests.post(API_URL, headers=headers, files=files)
        resp.raise_for_status()
        return resp.json()
    except Exception as e:
        print(str(e))
        sys.exit(0)

def mosaic(filename, detection_result):
    image = Image.open(filename)

```

```

    for face in detection_result['result']['faces']:
        x = int(face['x']*image.width) #사람 얼굴 중 제일 좌측
        w = int(face['w']*image.width) #사람 얼굴 끝까지 폭
        y = int(face['y']*image.height) #사람 얼굴 상단
        h = int(face['h']*image.height) #사람 얼굴 끝까지의 높이
        box = image.crop((x,y,x+w, y+h))
        box = box.resize((20,20), Image.NEAREST).resize((w,h), Image.NEAREST)
        image.paste(box, (x,y,x+w, y+h))

    return image

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='Mosaic faces.')
    parser.add_argument('image_file', type=str, nargs='?',
        default="C:/Users/memen/OneDrive/바탕 화면/인턴 - 무지개연구소/아이
        유.jpg", help='image file to hide faces') #default에 이미지 file을 넣습니다.

    args = parser.parse_args()

    detection_result = detect_face(args.image_file)
    image = mosaic(args.image_file, detection_result)
    image.show()

```

실행 결과

얼굴 부분만 찾아서 모자이크 됩니다.



2. 상품 추출

- 상품을 사각형으로 마킹 후 검출된 상품 이름출력
- 입력 - 이미지 url

http://t1.daumcdn.net/alvolo/_vision/openapi/r2/images/06.jpg

```

import sys
import argparse
import requests
from PIL import Image, ImageDraw, ImageFont
from io import BytesIO

API_URL = 'https://kapi.kakao.com/v1/vision/product/detect'
MYAPP_KEY = '발급받은 키를 입력해주세요.' # 발급받은 키

def detect_product(image_url):
    headers = {'Authorization': 'KakaoAK {}'.format(MYAPP_KEY)}

    try:
        data = { 'image_url' : image_url}

        resp = requests.post(API_URL, headers=headers, data=data)
        resp.raise_for_status()
        return resp.json()
    except Exception as e:
        print(str(e))
        sys.exit(0)

```

```

def show_products(image_url, detection_result):
    try:
        image_resp = requests.get(image_url)
        image_resp.raise_for_status()
        file_jpgdata = BytesIO(image_resp.content)
        image = Image.open(file_jpgdata)
    except Exception as e:
        print(str(e))
        sys.exit(0)

    draw = ImageDraw.Draw(image)
    for obj in detection_result['result']['objects']:
        x1 = int(obj['x1']*image.width) #사각형의 좌표
        y1 = int(obj['y1']*image.height)
        x2 = int(obj['x2']*image.width)
        y2 = int(obj['y2']*image.height)
        draw.rectangle([(x1,y1), (x2, y2)], fill=None, outline=(255,0,0,255))
        draw.text((x1+5,y1+5), obj['class'], (255,0,0))
    del draw

    return image

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='Detect Products.')
    parser.add_argument('image_url', type=str, nargs='?',
                        default="http://t1.daumcdn.net/alvolo/_vision/openapi/r2/images/06.jpg",
                        help='image url to show product's rect')
    #이미지 url
    args = parser.parse_args()

    detection_result = detect_product(args.image_url)
    image = show_products(args.image_url, detection_result)
    image.show()

```


실행결과



2. 상품 추출 2

- 입력 - 이미지 url

<https://i.pinimg.com/originals/3e/5f/d5/3e5fd59309bb4f79626bd23c63edb738.png>



3. 썸네일 생성

- 지정한 width /height 비율에 맞게 이미지를 축소시킵니다.
- 입력 - 이미지 파일

```

import sys

import argparse

import requests

from PIL import Image, ImageFilter


API_URL = 'https://kapi.kakao.com/v1/vision/thumbnaill/detect'

MYAPP_KEY = '발급받은 키를 입력해주세요.' #발급받은 키

def detect_thumbnail(filename, width, height):

    headers = {'Authorization': 'KakaoAK {}'.format(MYAPP_KEY)}

    try:
        files = { 'file' : open(filename, 'rb')}

        params = {'width': width, 'height': height}

        resp = requests.post(API_URL, headers=headers, data=params, files=files)

        resp.raise_for_status()

        return resp.json()

    except Exception as e:

        print(str(e))

        sys.exit(0)

```

```

def show_thumbnail(filename, detection_result, width, height):
    image = Image.open(filename)
    rect = detection_result['result']['thumbnail']
    thumbnail = image.crop((rect['x'], rect['y'], rect['x'] + rect['width'], rect['y'] + rect['height'])) #이미지 자르기
    thumbnail = thumbnail.resize((width, height)) #이미지 리사이징

    return thumbnail


if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='Make a thumbnail.')
    parser.add_argument('image_file', type=str, nargs='?', default="C:/Users/memen/OneDrive/바탕 화면/인턴 - 무지개연구소/아이유.jpg", #이미지 파일 경로
                        help='image file to make a thumbnail')
    parser.add_argument('width', type=int, nargs='?', default=150,
                        help='thumbnail width')
    parser.add_argument('height', type=int, nargs='?', default=200,
                        help='thumbnail height')

    args = parser.parse_args()

    detection_result = detect_thumbnail(args.image_file, args.width, args.height)
    image = show_thumbnail(args.image_file, detection_result, args.width, args.height)
    image.show()

```

실행결과 : 좌 - 원본 / 우 - 생성 썸네일



4. 태그 추출

이미지를 설명 할 수 있는 태그 여러 개를 출력합니다.

입력 – 이미지 url

<http://blog.fursys.com/wp-content/uploads/2016/11/16.png>

```
import sys

import argparse

import requests

from PIL import Image, ImageDraw, ImageFont

from io import BytesIO
```

```
API_URL = 'https://kapi.kakao.com/v1/vision/multitag/generate'
```

```
MYAPP_KEY = '발급받은 키를 입력해주세요.' #발급받은 키
```

```
def generate_tag(image_url):
```

```
    headers = {'Authorization': 'KakaoAK {}'.format(MYAPP_KEY)}
```

```
    try:
```

```
        data = { 'image_url' : image_url}
```

```
        resp = requests.post(API_URL, headers=headers, data=data)
```

```
        resp.raise_for_status()
```

```
        result = resp.json()['result']
```

```
        if len(result['label_kr']) > 0:
```

```
            if type(result['label_kr'][0]) != str:
```

```
                result['label_kr'] = map(lambda x: str(x.encode("utf-8")), result['label_kr']) #한국어 라벨
```

```
            print("이미지를 대표하는 태그는 {}".format(','.join(result['label_kr'])))
```

```
        else:
```

```
            print("이미지로부터 태그를 생성하지 못했습니다.")
```

```
    except Exception as e:
```

```
        print(str(e))
```

```
        sys.exit(0)
```

```
if __name__ == "__main__":
```

```
    parser = argparse.ArgumentParser(description='Classify Tags')
```

```
    parser.add_argument('image_url', type=str, nargs='?',
```

```
                        default="http://blog.fursys.com/wp-content/uploads/2016/11/16.png",
```

```
                        help='image url to classify') #이미지 url
```

```
    args = parser.parse_args()
```

```
    generate_tag(args.image_url)
```


실행 결과



```
==== RESTART: C:/Users/memen/OneDrive/바탕 화면/인턴 - 무지개연구소/kakao_vision  
_태그생성.py ===  
이미지를 대표하는 태그는 "사람,여러사람,가구,남성"입니다.
```

4. 태그 추출 2

입력 - 이미지 url (아까 상품 추출에서 사용한 블랙핑크 지수
공항패션 사진)

<https://i.pinimg.com/originals/3e/5f/d5/3e5fd59309bb4f79626bd23c63edb738.png>



= RESTART: C:\Users\memen\OneDrive\바탕 화면\인턴 - 무지개연구소\카카오\비전\kak
ao_vision_태그생성.py
이미지를 대표하는 태그는 "사람, 한 사람, 여성"입니다.

5. 성인 이미지 / 노출 이미지 판단

- 성인 이미지 ,노출 포함 여부를 점수로 보여주는 코드입니다.

- 입력 – 이미지 url

http://t1.daumcdn.net/alvolo/_vision/openapi/r2/images/10.jpg

```
import sys
import argparse
import requests
from PIL import Image, ImageDraw, ImageFont
from io import BytesIO
```

```
API_URL = 'https://kapi.kakao.com/v1/vision/adult/detect'
```

```
MYAPP_KEY = '발급받은 키를 입력해주세요.' #발급받은 키 정보
```

```
def detect_adult(image_url):
    headers = {'Authorization': 'KakaoAK {}'.format(MYAPP_KEY)}

    try:
        data = { 'image_url' : image_url}
        resp = requests.post(API_URL, headers=headers, data=data)
        resp.raise_for_status()
        result = resp.json()['result'] #분석 결과가 adult ~ soft ~ normal에 따라 텍스트 다르게 출력
        if result['adult'] > result['normal'] and result['adult'] > result['soft']:
            print("성인 이미지일 확률이 {}% 입니다.".format(result['adult']*100))
        elif result['soft'] > result['normal'] and result['soft'] > result['adult']:
            print("노출이 포함된 이미지일 확률이 {}% 입니다.".format(result['soft']*100))
        else :
            print("일반적인 이미지일 확률이 {}% 입니다.".format(result['normal']*100))
```

```
    except Exception as e:
        print(str(e))
        sys.exit(0)
```

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='Classify adult image.')
    parser.add_argument('image_url', type=str, nargs='?',
        default="http://t1.daumcdn.net/alvolo/_vision/openapi/r2/images/10.jpg",
        help='image url to classify') #이미지 url

    args = parser.parse_args()

    detect_adult(args.image_url)
```

실행 결과



```
==== RESTART: C:/Users/memen/OneDrive/바탕 화면/인턴 - 무지개연구소/kakao_vision  
_노출판단.py ===  
노출이 포함된 이미지일 확률이 88.0% 입니다.
```

5. 성인 이미지 / 노출 이미지 판단 2

- 이번엔 건전한 이미지를 돌려볼 때 결과입니다.

- 입력 – 이미지 url

https://post-phinf.pstatic.net/MjAxNzA5MjBfMjl1/MDAxNTA1ODg0NjMyNDgw.9GzMHmgvTppZHJAb2BOq1pdQJQqX79NTv003csX-Q6cg.k88VnGh5SQbQiZkDIIPRYrmX0kWu1kd0u3CsqWPkfrsg.PNG/20170920_134618.png?type=w800_q75



```
= RESTART: C:\Users\mmemen\OneDrive\바탕 화면\인턴 - 무지개연구소\카카오비전\kak  
ao_vision_노출판단.py  
일반적인 이미지일 확률이 96.8% 입니다.  
>>> |
```


6. 문자 영역 감지 & 문자 인식

카카오 예제 python 코드 에러로 인해 Rest API로 진행합니다.
순서는 문자 영역 감지 -> 문자 인식 입니다.

1. 문자 영역 - 필드 영역에 대한 이해

> Request

URL

```
POST /v1/vision/text/detect HTTP/1.1
Host: kapi.kakao.com
Authorization: KakaoAK {app_key}
Content-Type: multipart/form-data
```

Parameter

Name	Type	Description	Required
file	Binary	이미지 파일	O

> Response

Key

Name	Type	Description
result	Result	추출한 문자 영역에 대한 좌표 정보

result

Name	Type	Description
boxes	List<number[]>	추출한 문자영역에 대한 x,y 좌표 정보 하나의 box에서 꼭지점 순서는 [[좌상단x, 좌상단y], [우상단x, 우상단y], [우하단x, 우하단y], [좌하단x, 좌하단y]] *여기서 상단/하단은 모니터의 위아래를 가리킵니다. x 좌표는 왼쪽에서 오른쪽으로 갈수록 증가하며, y 좌표는 위에서 아래로 갈수록 증가합니다.

6-1. 문자 영역 감지

카카오 예제 python 코드 에러로 인해 Rest API로 진행합니다.
순서는 문자 영역 감지 -> 문자 인식 입니다.

1. 입력 이미지



6-1. 문자 영역 감지

2. 실행 코드 – 실행 툴은 chrome의 확장 프로그램인 rest client를 사용하였습니다.

- 1) 헤더에 Authorization 부분에 발급받은 키를 넣고
- 2) content – Type을 multipart/form-data로 맞춥니다.
- 3) 그 후, Body에 filed name을 file로 맞추고, CHOOSE FILE을 통해 이미지를 선택 후 SEND

Method POST Request URL https://kapi.kakao.com/v1/vision/text/detect

Parameters ^

Headers Body Variables

Toggle source mode + Insert headers set

Header name	Header value	
Authorization	KakaoAK [redacted]	X ✎ ?
Content-Type	multipart/form-data	X ✎ ?

Method POST Request URL https://kapi.kakao.com/v1/vision/text/detect

Parameters ^

Headers Body Variables

Body content type multipart/form-data

Field name file

CHOOSE FILE wakeup.jpg (64892 bytes) X

문자 영역 감지 결과

문자 영역에 해당하는 부분이 boxes 배열 안에 숫자로 들어가 있습니다.

```
{  
  "result":{  
    "boxes":[[[30,393],[208,393],[208,457],[30,457]],[[229,404],[307,404],[307,462],[  
229,462]],[[324,396],[568,394],[568,457],[324,458]]]  
  }  
}
```

이제 이 값을 이용하여 문자 인식을 실시합니다.

6-2. 문자 인식

이제 문자 인식 입니다.

1. 문자 인식 – 필드 영역에 대한 이해

Request

URL

```
POST /v1/vision/text/recognize HTTP/1.1
Host: kapi.kakao.com
Authorization: KakaoAK {app_key}
Content-Type: multipart/form-data
```

Parameter

Name	Type	Description	Required
file	Binary	이미지 파일	O
boxes	List of number[]	문자의 box 영역의 좌표 값 문자 영역 감지 API 응답의 result 하위에서 boxes 값을 추출한 좌표값을 그대로 넣어줘야 합니다.	O

Response

Key

Name	Type	Description
result	Result	추출한 글자 정보

result

Name	Type	Description
recognition_words	String[]	추출한 글자 정보. box 순으로 정렬

6-2. 문자 인식

2. 실행 코드 – 실행 툴은 chrome의 확장 프로그램인 rest client를 사용하였습니다.

- 1) 헤더에 Authorization 부분에 발급받은 키를 넣고
- 2) content – Type을 multipart/form-data로 맞춥니다.
- 3) 그 후, Body에 filed name을 file로 맞추고, CHOOSE FILE을 통해 이미지를 선택합니다.
- 4) 마지막으로 Body에 text filed를 추가하여 앞서 받았던 boxes의 값인 배열을 그대로 집어넣습니다.

Method POST Request URL <https://kapi.kakao.com/v1/vision/text/recognize>

Parameters ^

Headers

Header name	Header value
Authorization	KakaoAK [REDACTED]
Content-Type	multipart/form-data

Body

Toggle source mode + Insert headers set

Method POST Request URL <https://kapi.kakao.com/v1/vision/text/recognize> SEND

Parameters ^

Headers

Body content type multipart/form-data

Body

Field name	file	CHOOSE FILE	wakeup.jpg (64892 bytes)	X	
Content type (Optional)					
boxes	[[[30,393],[208,393],[208,457],[30,457]],[[229,404],[307,404],[307,462],[229,462]],[[324,396],[568,396],[568,457],[324,457]]]]				X

문자 인식 결과

200 OK 184.21 ms



```
{
  -"result": {
    -"recognition_words": [Array(3)]
      0: "Wake",
      1: "up",
      2: "humang"
    ],
  }
}
```

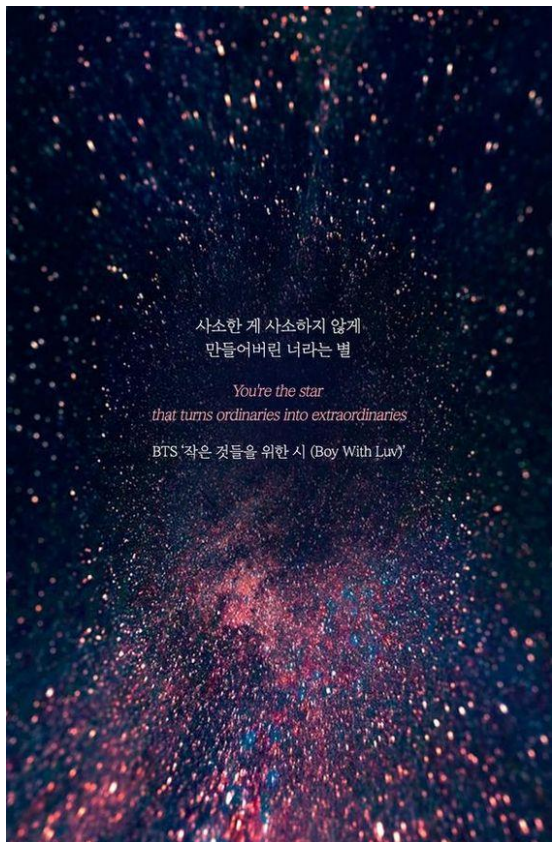


조금 틀렸긴 하지만 비슷하게 나옵니다.
human!을 humang으로 인식하였음..

6. 문자 영역 감지 & 문자 인식

이번엔 영어 문자 두 줄이 있는 이미지를 돌렸을 때 어떠한 결과가 나오는지 보겠습니다.

정상적으로 잘 인식되는 것을 알 수 있습니다.



```
[[[196,321],[241,321],[241,337],[196,337]],[[245,321],[366,321],[366,341],[245,341]],[[205,347],[356,347],[356,362],[205,362]],[[233,388],[325,388],[325,403],[233,403]],[[149,415],[409,415],[409,429],[149,429]],[[149,452],[209,452],[209,468],[149,468]],[[213,451],[256,451],[256,468],[213,468]],[[260,451],[343,451],[343,468],[260,468]],[[347,452],[407,452],[407,466],[347,466]]]
```

200 OK 197.92 ms



```
{
  -"result": {
    -"recognition_words": [Array[9]
      0: "사소한",
      1: "게 사소하지 않게",
      2: "만들어버린 너라는 별",
      3: "You're the star",
      4: "that turns ordinaries into extraordinarie",
      5: "BTS '작은",
      6: "것들을",
      7: "위한 시 (Boy",
      8: "With Luy"
    ],
  },
}
```