

Python을 활용한 이미지 분석

4. 이미지 데이터 이해



이미지 데이터 이해

01 파일 경로와 목록

02 RGB 색상

03 이미지 처리 도구



1

파일 경로와 목록

1. 파일 경로와 목록

파일 경로

■ 절대 경로

- ▶ 어떠한 웹페이지나 파일이 가진 고유한 경로
- ▶ 최상위 디렉토리가 포함된 경로

예

`http://bd.kma.go.kr`



`file:///Users/user/Downloads/`



`C:\user\documents`



파일 경로

■ 상대 경로

- ▶ 현재 위치한 곳을 기준으로 하는 경로

/

루트

./

현재 위치

../

현재 위치의 상위 폴더

../..

두 단계 위의 상위 폴더



1. 파일 경로와 목록



이미지 파일 목록

■ os.walk()를 사용한 파일 목록 탐색

- ▶ 현재 경로, 현재 경로의 하위 경로, 파일 목록을 탐색함

코드

```
import os
for dirpath, dirnames, filenames in os.walk('images'):
    print(dirpath, dirnames, filenames)
```

출력

```
images ['cloudy', 'shine', 'sunrise', 'rainy',
'alien_test', 'foggy']
images/cloudy [...]
images/shine [...]
images/sunrise [...]
images/rainy [...]
images/alien_test [...]
images/foggy [...]
```



1. 파일 경로와 목록



이미지 파일 목록

■ glob로 파일 목록 탐색

- ▶ glob는 패턴(유닉스 셸이 사용하는 규칙)을 사용하여 파일을 검색하는 모듈임
- ▶ 현재 디렉터리와 하위 디렉터리의 모든 텍스트파일을 찾아서 출력함

코드

```
import glob
paths = glob.glob('./images/*/*.jpg')
paths
```

출력

```
['./images/cloudy/cloudy126.jpg',
 './images/cloudy/cloudy132.jpg',
 ....
 './images/rainy/rain132.jpg',
 ...]
```



1. 파일 경로와 목록



실습

■ 예제

문제 상황

images 폴더의 파일을 탐색

실습 코드

```
import glob
paths = glob.glob('./images/*/*.jpg')
```

예시 화면

```
# glob은 패턴(유닉스 셸이 사용하는 규칙)을 사용하여 파일을 검색하는 모듈
# 현재 디렉터리와 하위 디렉터리의 모든 텍스트파일을 찾아서 출력
```

```
import glob
paths = glob.glob('./images/*/*.jpg')
paths
```

```
['./images/cloudy/cloudy126.jpg',
 './images/cloudy/cloudy132.jpg',
 './images/cloudy/cloudy23.jpg',
 './images/cloudy/cloudy37.jpg',
 './images/cloudy/cloudy250.jpg',
 './images/cloudy/cloudy244.jpg',
 './images/cloudy/cloudy278.jpg',
```


2

RGB 색상

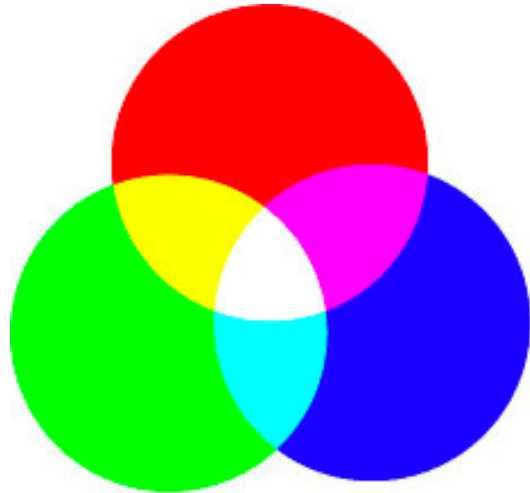
RGB 채널의 이해

■ RGB 색 모형

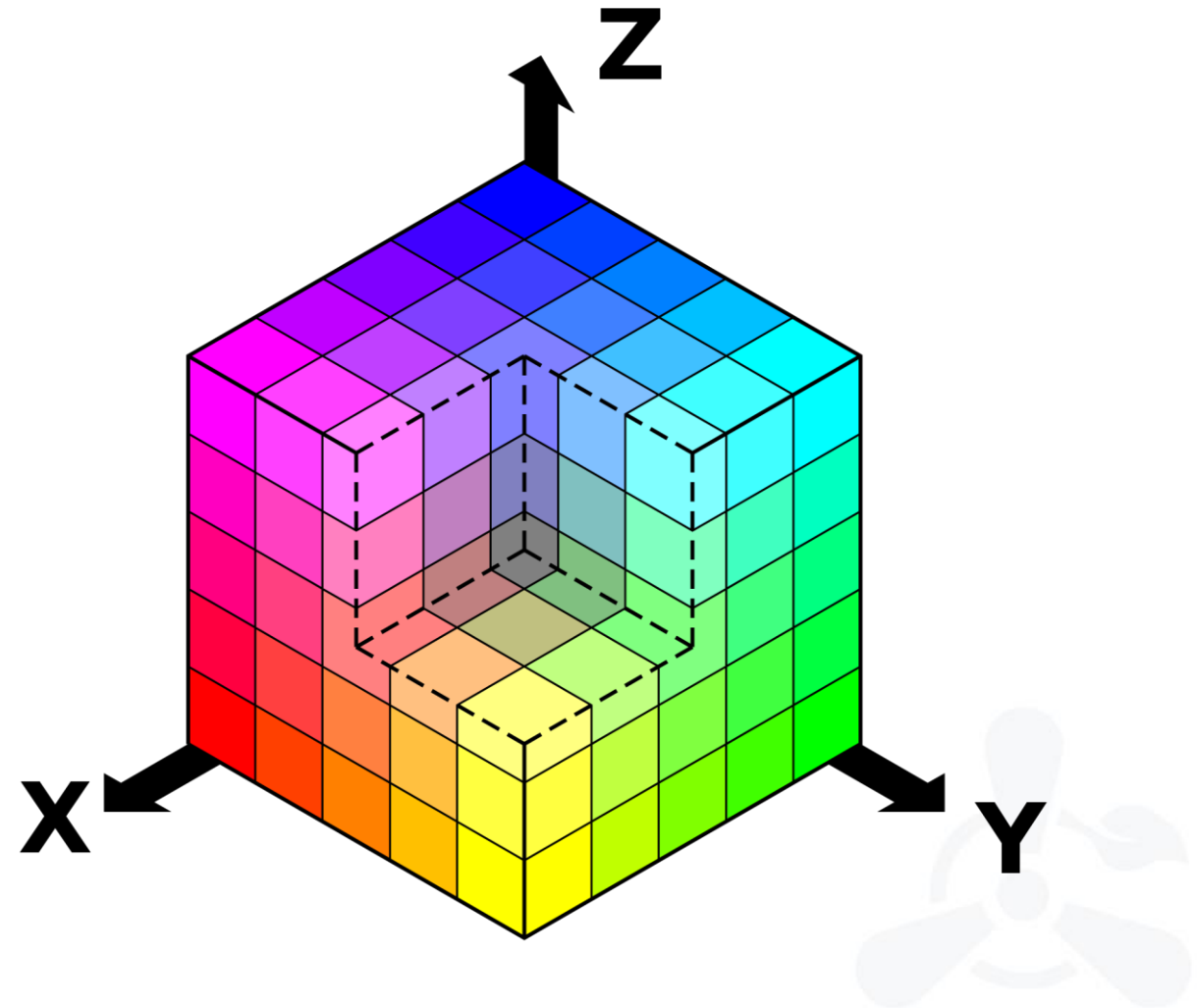
- ▶ 빛의 삼원색을 이용하여 색을 표현하는 방식

가산 혼합

빨강(RED), 초록(GREEN), 파랑(BLUE)
세 종류의 광원을 이용하여 색을 섞을수록
밝아짐



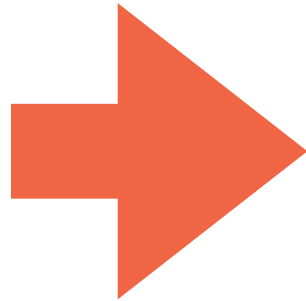
[출처 : 위키피디아. <https://ko.wikipedia.org/wiki/RGB>]



2. RGB 색상

RGB 채널의 이해

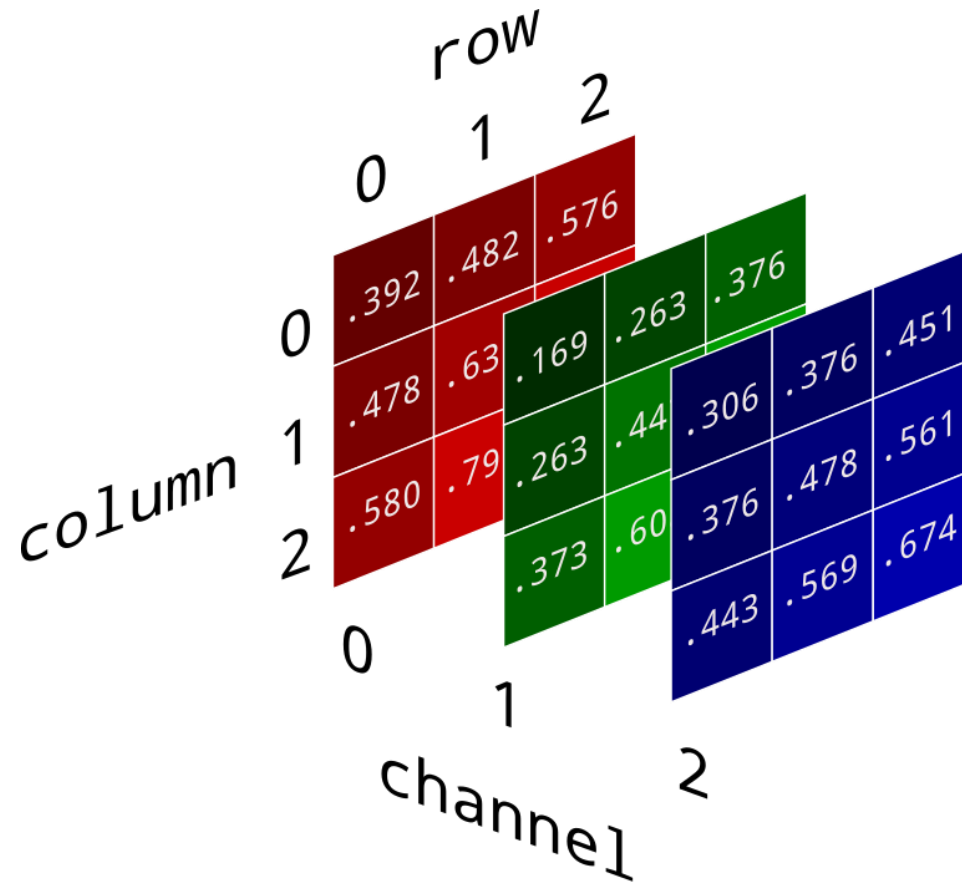
■ RGB 색 모형



2. RGB 색상

RGB 채널의 이해

■ RGB 삼원색



[출처 : https://e2eml.school/convert_rgb_to_grayscale.html]

2. RGB 색상

RGB 채널의 이해

■ RGB의 분포도

(0, 0, 0)

검정

(0, 0, 255)

파랑

(255, 255, 255)

하양

(255, 255, 0)

노랑

(255, 0, 0)

빨강

(0, 255, 255)

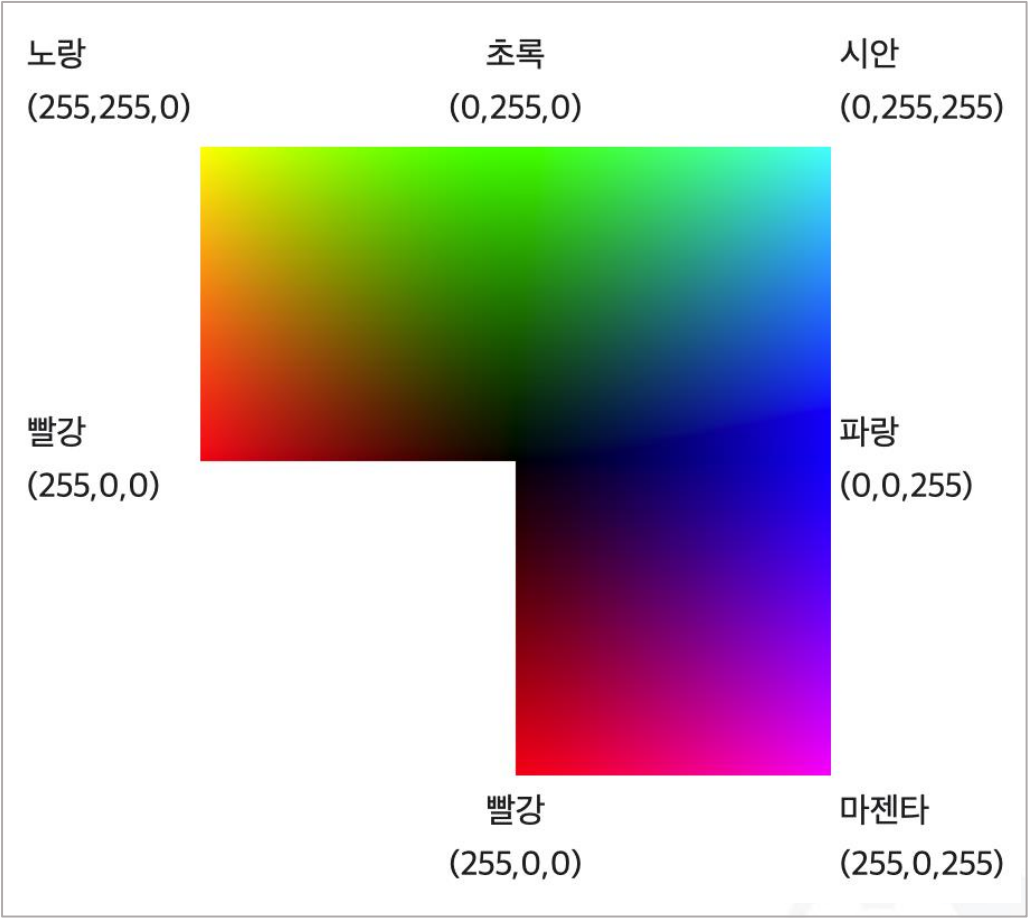
시안

(0, 255, 0)

초록

(255, 0, 255)

마젠타





실습

예제

문제 상황

이미지 데이터의 RGB 값 이해

실습 코드

```
import matplotlib.pyplot as plt  
plt.imread()
```

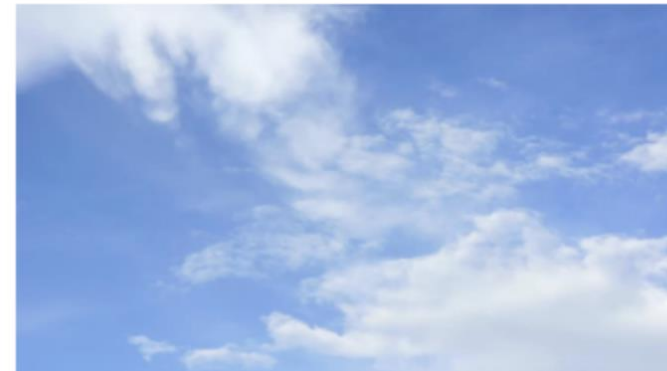
예시 화면

```
import matplotlib.pyplot as plt  
img = plt.imread('./images/cloudy/cloudy126.jpg')  
print(img.shape)  
img
```

(480, 852, 3)

```
array([[[207, 220, 239],  
       [207, 220, 239],  
       [207, 220, 239],  
       ...,  
       [ 90, 132, 206],  
       [ 90, 132, 206],  
       [ 90, 132, 206]]],  
      dtype=uint8)
```

```
plt.imshow(img)  
plt.axis("off")  
  
(-0.5, 851.5, 479.5, -0.5)
```



3

이미지 처리 도구

PIL(Python Imaging Library)

■ 개념

- ▶ 파이썬 인터프리터에 다양한 이미지 파일 형식을 지원함
- ▶ 강력한 이미지 처리와 그래픽 기능을 제공함
- ▶ Pillow라는 후속 프로젝트가 PIL 저장소에서 갈려져 나와 Python 3.x 지원이 추가됨

→ PIL을 대체하기 위해서 채택됨

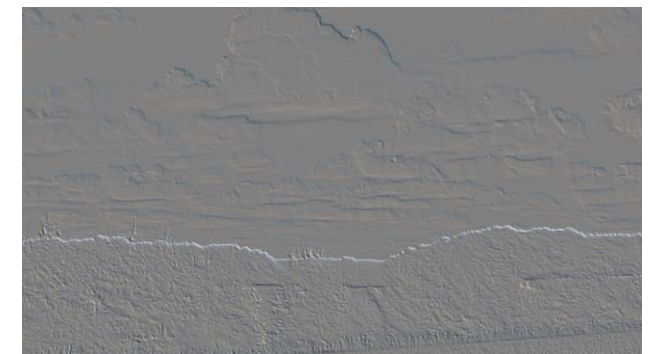
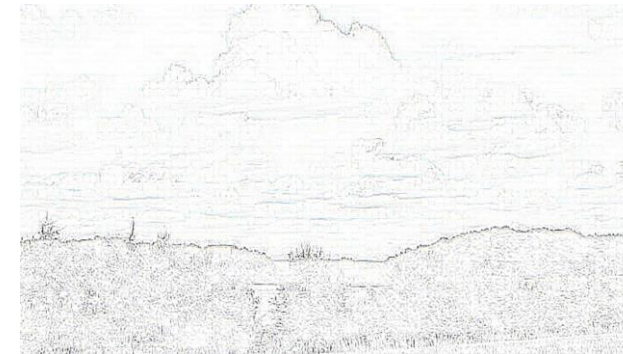


3. 이미지 처리 도구

PIL(Python Imaging Library)

■ 기능

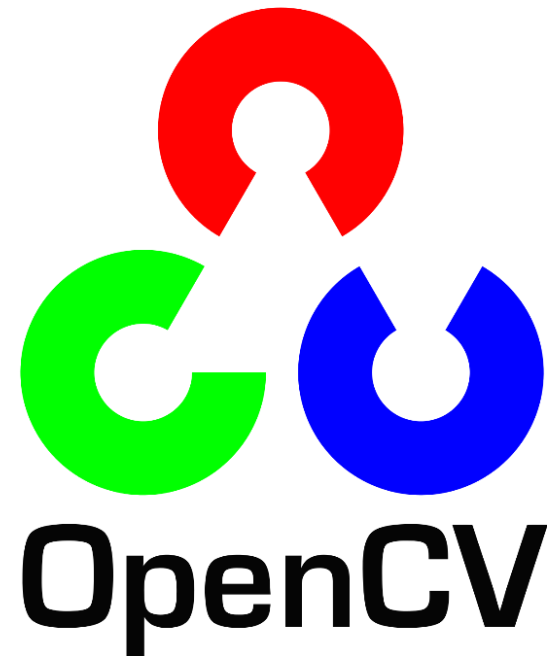
- 1 픽셀 단위의 조작
- 2 마스킹 및 투명도 제어
- 3 흐림, 윤곽 보정, 윤곽 검출 등의 이미지 필터
- 4 선명하게, 밝기 보정, 명암 보정, 색 보정 등의 화상 조정
- 5 이미지에 텍스트 추가 등



OpenCV(Open Source Computer Vision)

■ 개념

- ▶ 실시간 컴퓨터 비전을 목적으로 인텔이 개발한 프로그래밍 라이브러리
- ▶ 윈도우, 리눅스 등에서 사용 가능한 크로스 플랫폼 오픈소스 도구
- ▶ TensorFlow, Torch / PyTorch 및 Caffe의 딥러닝 프레임워크를 지원

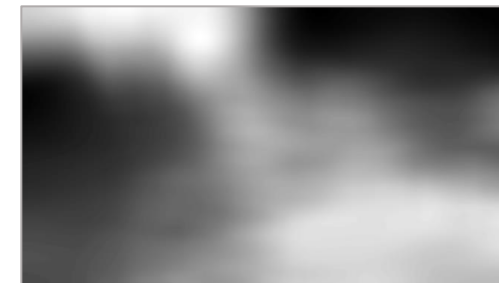
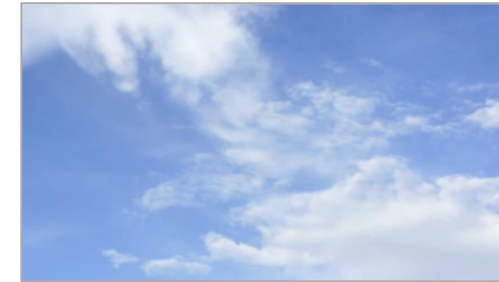


3. 이미지 처리 도구

OpenCV(Open Source Computer Vision)

■ 기능

- 1 비디오에서 컬러 개체를 추적
- 2 이미지의 기하학적 변환 - 회전, 이동 등
- 3 형태 변형, 이미지 그라디언트, Edge Detection
- 4 이미지 혼합, 윤곽선 찾기
- 5 이미지 변환 - 푸리에, 코사인 변환 등
- 6 이미지에서 개체 검색, 원 등의 감지, 이미지 분할
- 7 전경 추출



3. 이미지 처리 도구



실습

■ 예제

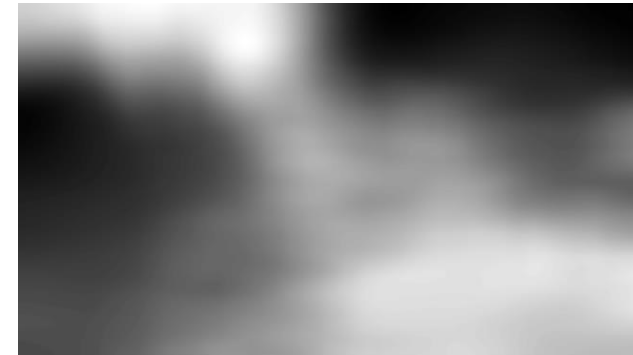
문제 상황

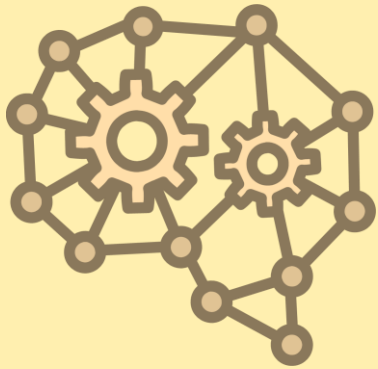
Pillow, OpenCV를 통한 이미지 변환

실습 코드

```
from PIL import Image, ImageFilter
```

예시 화면





KEY POINT

▶ 파일 경로와 목록

- 절대 경로 : 어떠한 웹페이지나 파일이 가진 고유한 경로
- 상대 경로 : 현재 위치한 곳을 기준으로 하는 경로
- `os.walk()`를 사용한 파일 목록 탐색
- `glob`는 유닉스 셸이 사용하는 규칙을 사용하여 파일을 검색할 때 사용하는 모듈로 파일을 탐색

▶ RGB 색상

- RGB 색 모형 : 빛의 삼원색을 이용하여 색을 표현하는 방식
- 각 색상(Hue)마다 256가지의 명도, 선명도를 결정

▶ 이미지 처리 도구

- PIL(Python Imaging Library)
: 강력한 이미지 처리와 그래픽 기능을 제공
- OpenCV(Open Source Computer Vision)
: 실시간 이미지 프로세싱에 중점을 둔 라이브러리





Python을 활용한 이미지 분석

4. 이미지 데이터 이해

“이번 시간을 모두 마치셨습니다.
수고하셨습니다.”