

open CV

1. openCV 설치하기 : `pip install opencv-python`

- 파이썬 이미지 처리 모듈
 - PILLOW: 기본모듈, 단순함, 필요하면 이미지를 array로 변환해야함
 - openCV: 강의내용 참조
 - skimg: 흑백2채널을 컬러3채널로 변하는등 데이터분석에 필요한 이미지 전처리에 해당하는 작업이 많음
- 위키정의
 - OpenCV(Open Source Computer Vision)은 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리이다. 원래는 인텔이 개발하였다. 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 인텔 CPU에서 사용되는 경우 속도의 향상을 볼 수 있는 IPP(Intel Performance Primitives)를 지원한다. 이 라이브러리는 윈도우, 리눅스 등에서 사용 가능한 크로스 플랫폼이며 오픈소스 BSD 허가서 하에서 무료로 사용할 수 있다. OpenCV는 TensorFlow, Torch / PyTorch 및 Caffe의 딥러닝 프레임워크를 지원한다.
 - OpenCV는 이미지를 읽을때 배열구조로 읽어 별도의 array변환이 필요없으며,
 - OpenCV 모듈에서 얼굴, 눈, 코등 신체의 일부를 인식하거나, 배경제거등의 컴퓨터비전관련 명령어를 포함하고 있다.
 - 동영상 읽기도 openCV에서 처리한다.

2. openCV로 이미지 자료 읽기

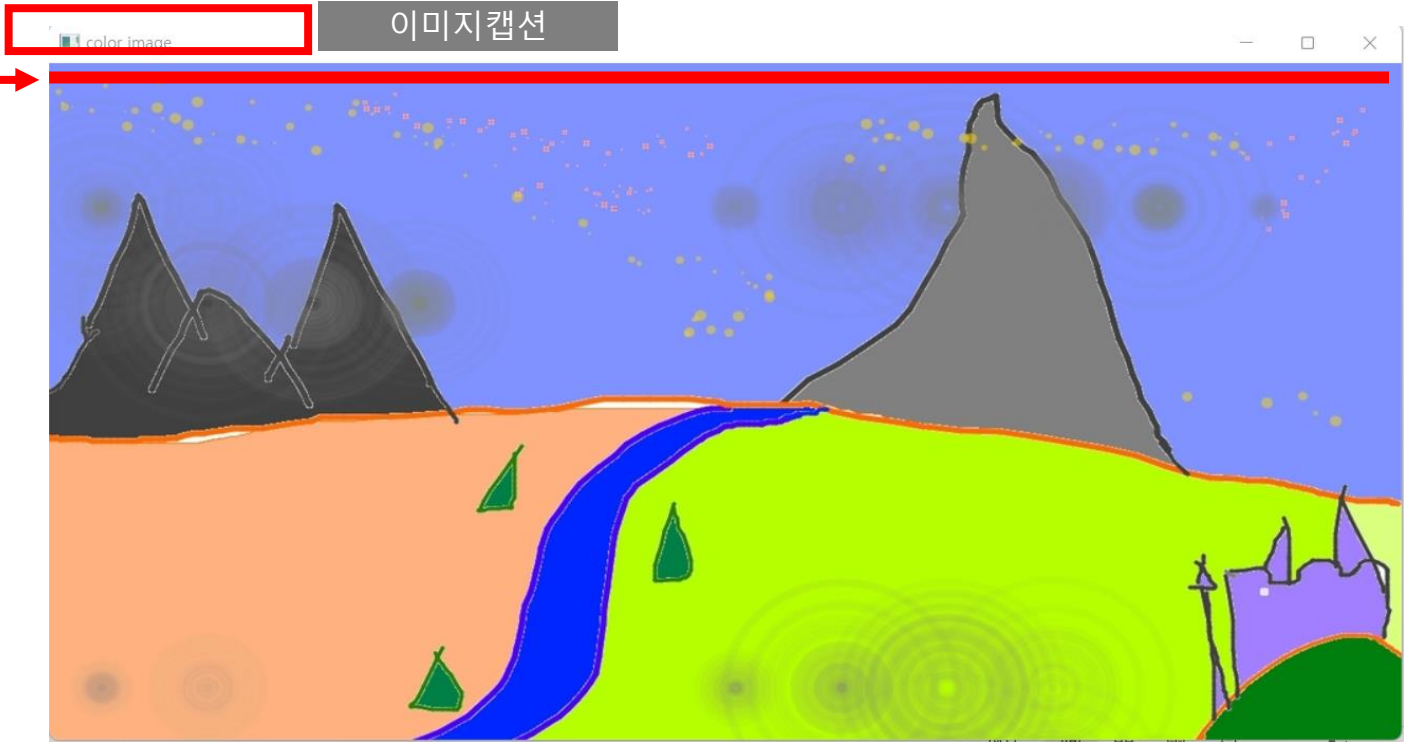
```
1 #####
2 #### 컬러로 읽기
3 #####
4 import cv2
5
6 img_color = cv2.imread('sample.jpg', cv2.IMREAD_COLOR) # img_color = cv2.imread('test.jpg', 1)
7 print(img_color[0], np.shape(img_color))
8 cv2.imshow('color image', img_color) # '이미지캡션', 이미지 파일명
9 cv2.waitKey(0) # 별도의 창에서 뜨기 때문에 종료키가 들어올때까지 잠시 대기
```

컬러는 옵션1로 읽
어도 됨

```
[[255 146 128]
 [255 146 128]
 [255 146 128]
 ...
 [255 146 128]
 [255 146 128]
 [255 146 128]]
```

shape
↓
(512, 1024, 3)

img_color[0]



2. openCV로 이미지 자료 읽기

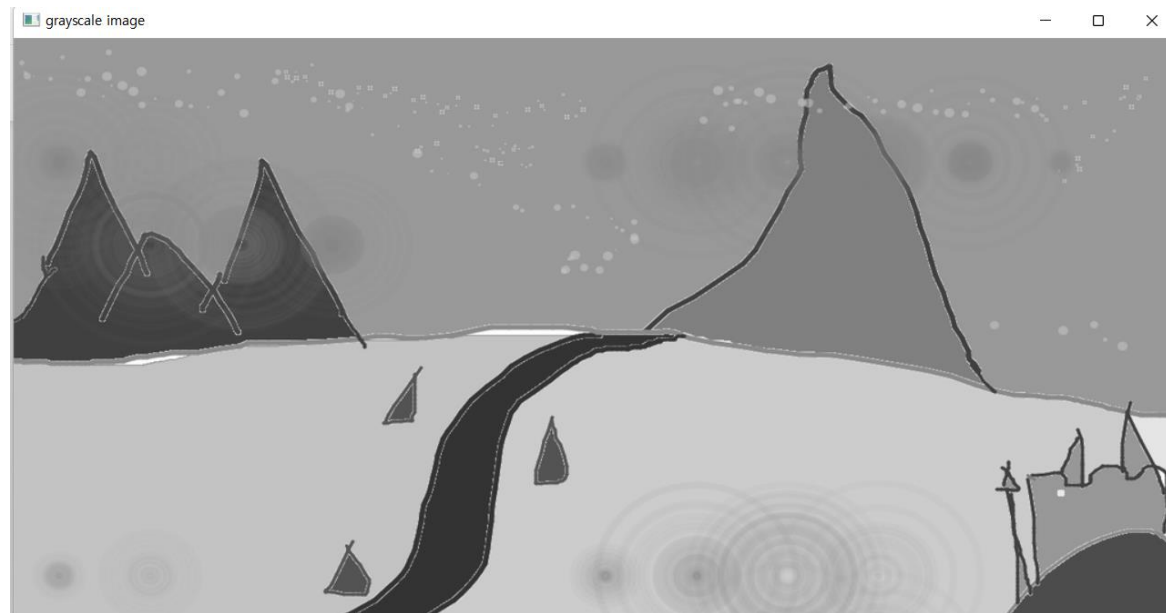
```
1 #####
2 #### 그레이스케일로 읽기
3 #####
4
5 import cv2
6 img_grayscale = cv2.imread('sample.jpg', cv2.IMREAD_GRAYSCALE) # img_grayscale = cv2.imread('test.jpg', 0)
7
8 print(img_grayscale[0], np.shape(img_grayscale))
9
10 cv2.imshow('grayscale image', img_grayscale)
11 cv2.waitKey(0)
12
```

[153 153 153 ... 153 153 153] (512, 1024)

-1

img_color[0]

shape



2. openCV로 이미지 자료 읽기

```
1 #####
2 #### 알파채널까지 읽기 , IMREAD_COLOR로 읽으면 채널이 3으로 읽힘
3 #####
4
5 import cv2
6 import numpy as np
7 img_unchanged = cv2.imread('png.png', cv2.IMREAD_UNCHANGED) # img_unchanged = cv2.imread('test.jpg', -1)
8 |
9 print(img_unchanged[0][0], np.shape(img_unchanged))
10
11 cv2.imshow('grayscale image', img_unchanged)
12 cv2.waitKey(0)
```

[255 255 255 255] (720, 720, 4) shape 4채널

-1

참고: png 에 대하여서는 개념
만 이해합니다.
복잡한 처리는 추후에함

정리 openCV로 이미지 읽기

opencv imread의 flag는 총 3가지가 있다.

| | | |
|----------------------|----|----------------------------------|
| cv2.IMREAD_COLOR | 1 | 이미지 파일을 Color로 읽고, Default값이다. |
| cv2.IMREAD_GRAYSCALE | 0 | 이미지를 Grayscale로 읽는다. |
| cv2.IMREAD_UNCHANGED | -1 | 이미지파일을 alpha channel까지 포함하여 읽는다. |

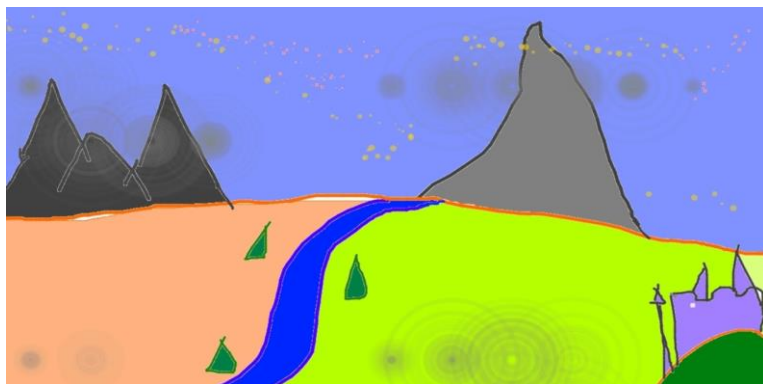
읽어보기

<https://shuka.tistory.com/23>

3. OpenCV로 읽은 BGR 자료 → RGB로 변환

openCV로 읽은 자료는 numpy로 읽히며 RGB가 아닌 BGR로 읽힘으로 별도의 변환 없으면 plt.imshow 에서 색상표가 이상하게 나옴

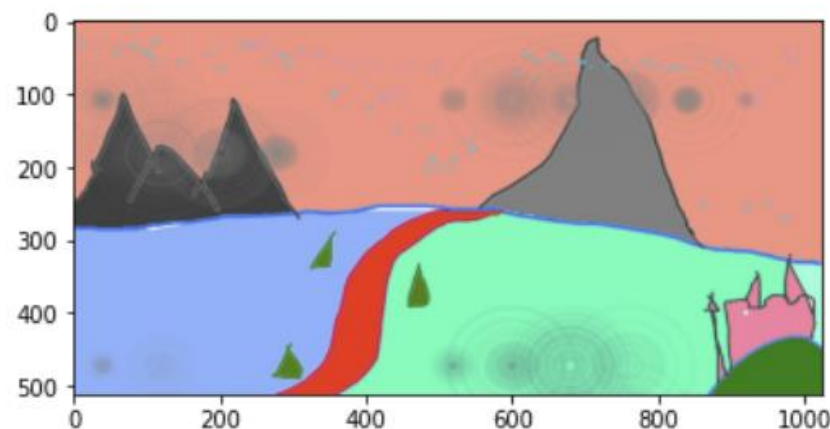
```
1 #####
2 #### 컬러로 읽기
3 #####
4 import cv2
5
6 img_color = cv2.imread('sample.jpg', cv2.IMREAD_COLOR)
7 cv2.imshow('color image', img_color)
8 cv2.waitKey(0) # 별도의 창에서 뜨기 때문에 종료키
```



openCV로 읽은 원본 이미지

```
1 #####
2 #### cv2로 읽을때와 달리, plt에서 확인하면 rgb가
3 #####
4
5 import matplotlib.pyplot as plt
6 plt.imshow(img_color)
```

<matplotlib.image.AxesImage at 0x2b3fdb8370>



plt.imshow로 보면 그림색이
다름 (rgb가 아닌 bgr로 읽기때문)

```
1 #####
2 #### cv2로 읽을때와 달리, plt에서 확인하면 rgb가 아닌 bgr로 나옴
3 #####
4 import numpy as np
5 print(np.shape(img_color))
6 tmp=np.zeros((512,1024,3))
7 tmp[:, :, 0]=img_color[:, :, 2]
8 tmp[:, :, 1]=img_color[:, :, 1]
9 tmp[:, :, 2]=img_color[:, :, 0]
10
11 tmp=tmp.astype('uint8')
12 print(tmp[0])
13 plt.imshow(tmp)
```

(512, 1024, 3)

[[128 146 255]

[128 146 255]

[128 146 255]

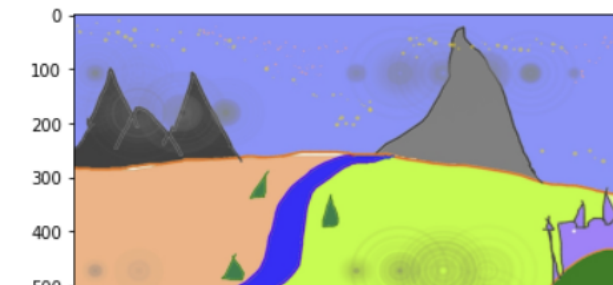
...

[128 146 255]

[128 146 255]

[128 146 255]]

<matplotlib.image.AxesImage at 0x2b3fdc50df0>



bgr를 rgb로 변경해야함

3. OpenCV로 읽은 BGR 자료 → RGB로 변환

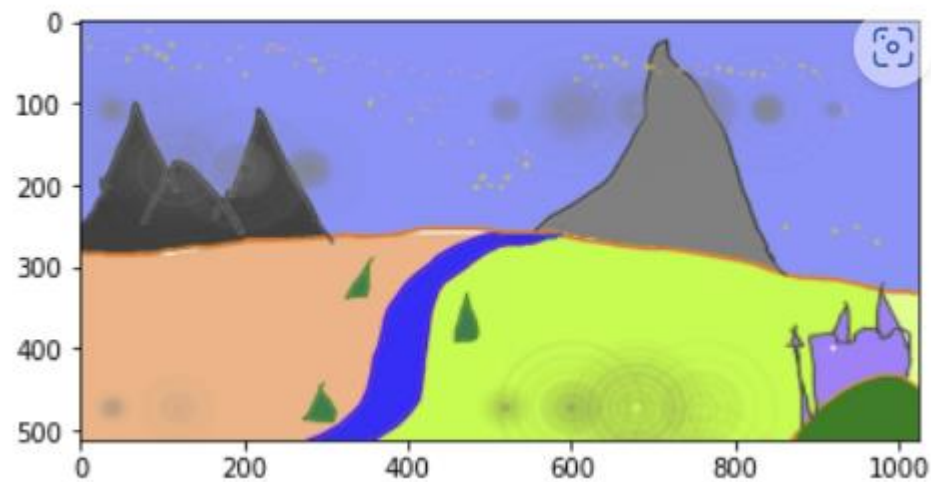
openCV로 읽은 자료는 numpy로 읽히며 RGB가 아닌 BGR로 읽힘으로 별도의 변환 없으면 plt.imshow 에서 색상표가 이상하게 나옴

cv2.COLOR_BGR2RGB

```
1 #####
2 #### 하지만 사용자가 일일이 rgb로 변경할 필요가 없음.
3 #####
4 img_color = cv2.imread('sample.jpg',cv2.IMREAD_COLOR)|
5 img_color=cv2.cvtColor(img_color,cv2.COLOR_BGR2RGB)
6
7 print(img_color[0])
8 plt.imshow(img_color)
```

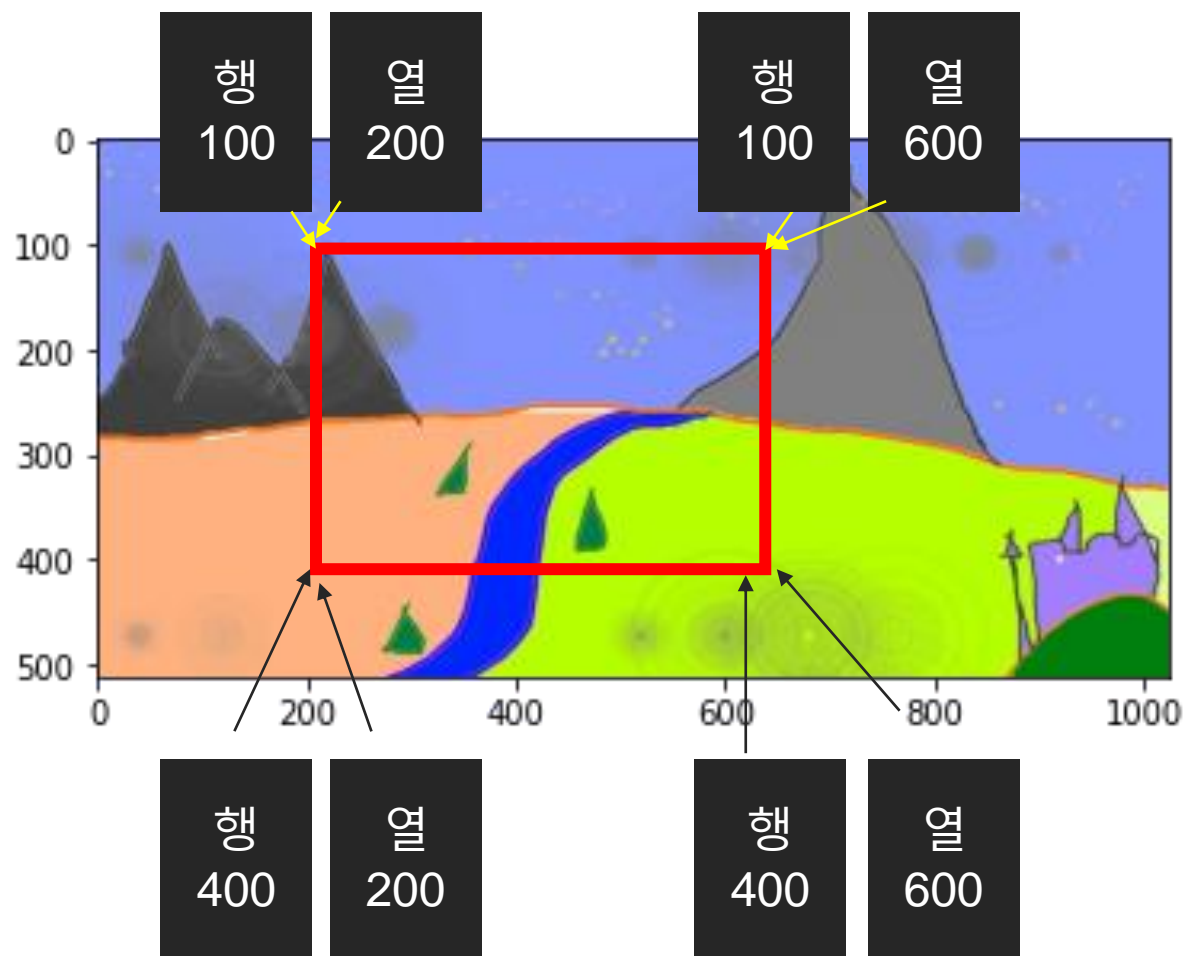
```
[[128 146 255]
 [128 146 255]
 [128 146 255]
 ...
 [128 146 255]
 [128 146 255]
 [128 146 255]]
```

<matplotlib.image.AxesImage at 0x235e8488eb0>



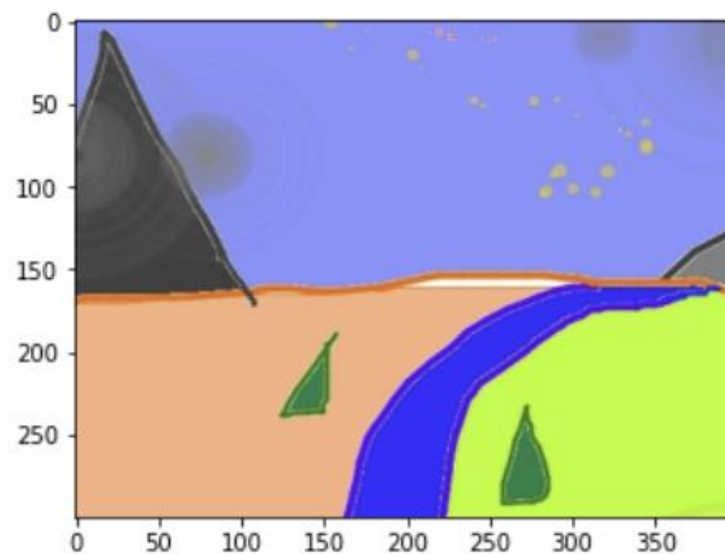
4. 이미지의 일정영역만 띄어내기

이미지의 일정영역은 `img[행시작:행종료, 열시작:열종료]` 로 지정하면 됨.
단 이 영역은 마우스로 드래그할때는 x,y좌표값으로 들어감으로 이해가 필요함



```
1 img_color = cv2.imread('sample.jpg', cv2.IMREAD_COLOR)
2 img_color = cv2.cvtColor(img_color, cv2.COLOR_BGR2RGB)
3 tmp = img_color[100:400, 200:600]
4 plt.imshow(tmp)
```

<matplotlib.image.AxesImage at 0x235e81eba00>



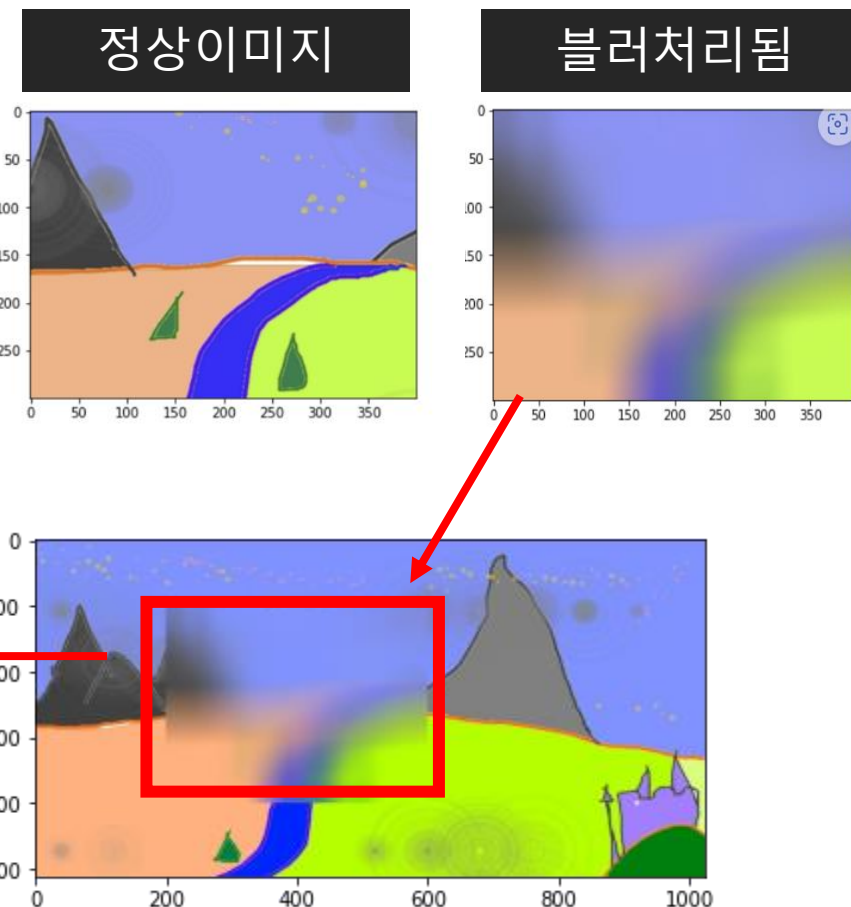
4. 이미지의 일정영역만 띄어내기

이작업후 마우스로 드래그하여서 이미지의 일정영역을 블러처리하는 코드로 실습함.

```
#####  
####  이미지의 일정영역을 뿌옇게 변하게 하는 블러효과를 넣음  
#### https://076923.github.io/posts/Python-opencv-13/  
#####  
dst = cv2.blur(tmp, (90, 90), anchor=(-1, -1), borderType=cv2.BORDER_DEFAULT)  
plt.imshow(dst)
```

```
#####  
#### 블러효과 자료를 원래 이미지에 넣음  
#####  
dst = cv2.blur(tmp, (90, 90), anchor=(-1, -1),  
borderType=cv2.BORDER_DEFAULT)  
plt.imshow(dst)
```

```
img_color[100:400,200:600]=dst  
plt.imshow(img_color)
```

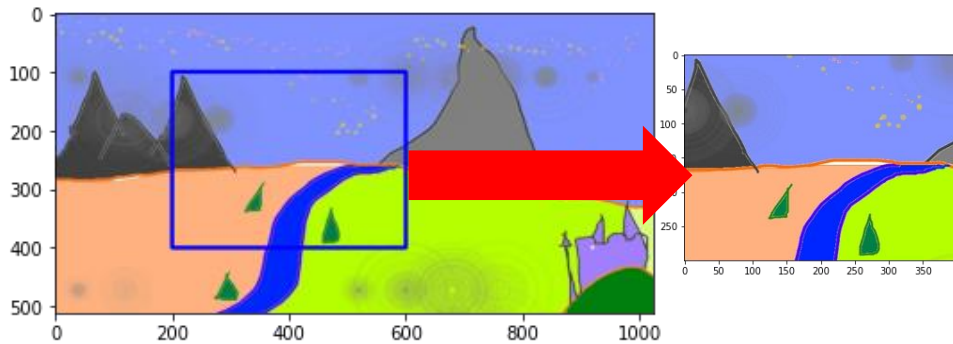


5. 다른 이미지 자료 합성하기

이작업후 마우스로 드래그하여서 이미지의 일정영역을 블러처리하는 코드로 실습함.

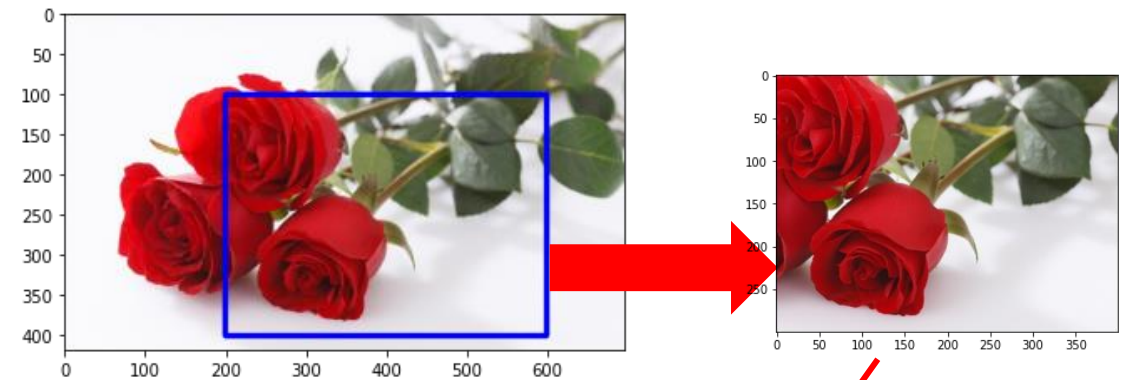
```
img_color = cv2.imread('sample.jpg',cv2.IMREAD_COLOR)  
img_color=cv2.cvtColor(img_color,cv2.COLOR_BGR2RGB)  
tmp1=img_color[100:400,200:600]
```

```
plt.imshow(tmp1)
```

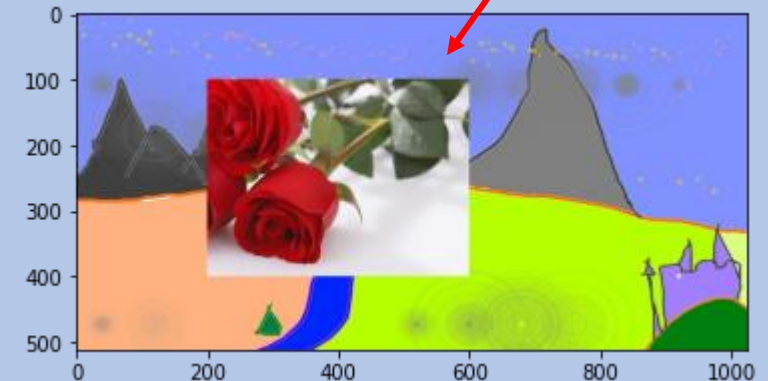


```
img_color2 = cv2.imread('flower.jpg',cv2.IMREAD_COLOR)  
img_color2=cv2.cvtColor(img_color2,cv2.COLOR_BGR2RGB)  
tmp2=img_color2[100:400,200:600]
```

```
plt.imshow(tmp2)
```



```
tmp1=tmp2  
img_color[100:400,200:600]=tmp1  
plt.imshow(img_color)
```



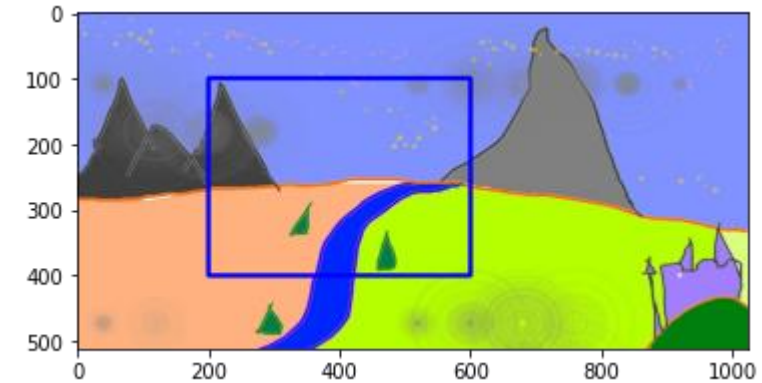
6. 네모박스 그리기

이작업후 마우스로 드래그하여서 이미지의 일정영역을 블러처리하는 코드로 실습함.

```
import cv2
import matplotlib.pyplot as plt

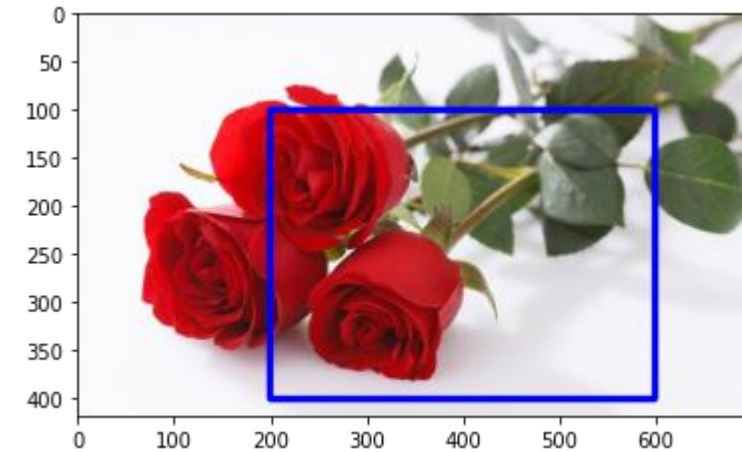
img_color1 = cv2.imread('sample.jpg',cv2.IMREAD_COLOR)
img_color1=cv2.cvtColor(img_color1,cv2.COLOR_BGR2RGB)
tmp1=img_color1[100:400,200:600]

rectangle=cv2.rectangle(img_color1,(200,100),(600,400),(0,0,255),5)
plt.imshow(rectangle)
```



```
img_color2 = cv2.imread('flower.jpg',cv2.IMREAD_COLOR)
img_color2=cv2.cvtColor(img_color2,cv2.COLOR_BGR2RGB)
tmp2=img_color2[100:400,200:600]

rectangle=cv2.rectangle(img_color2,(200,100),(600,400),(0,0,255),5)
plt.imshow(rectangle)
```

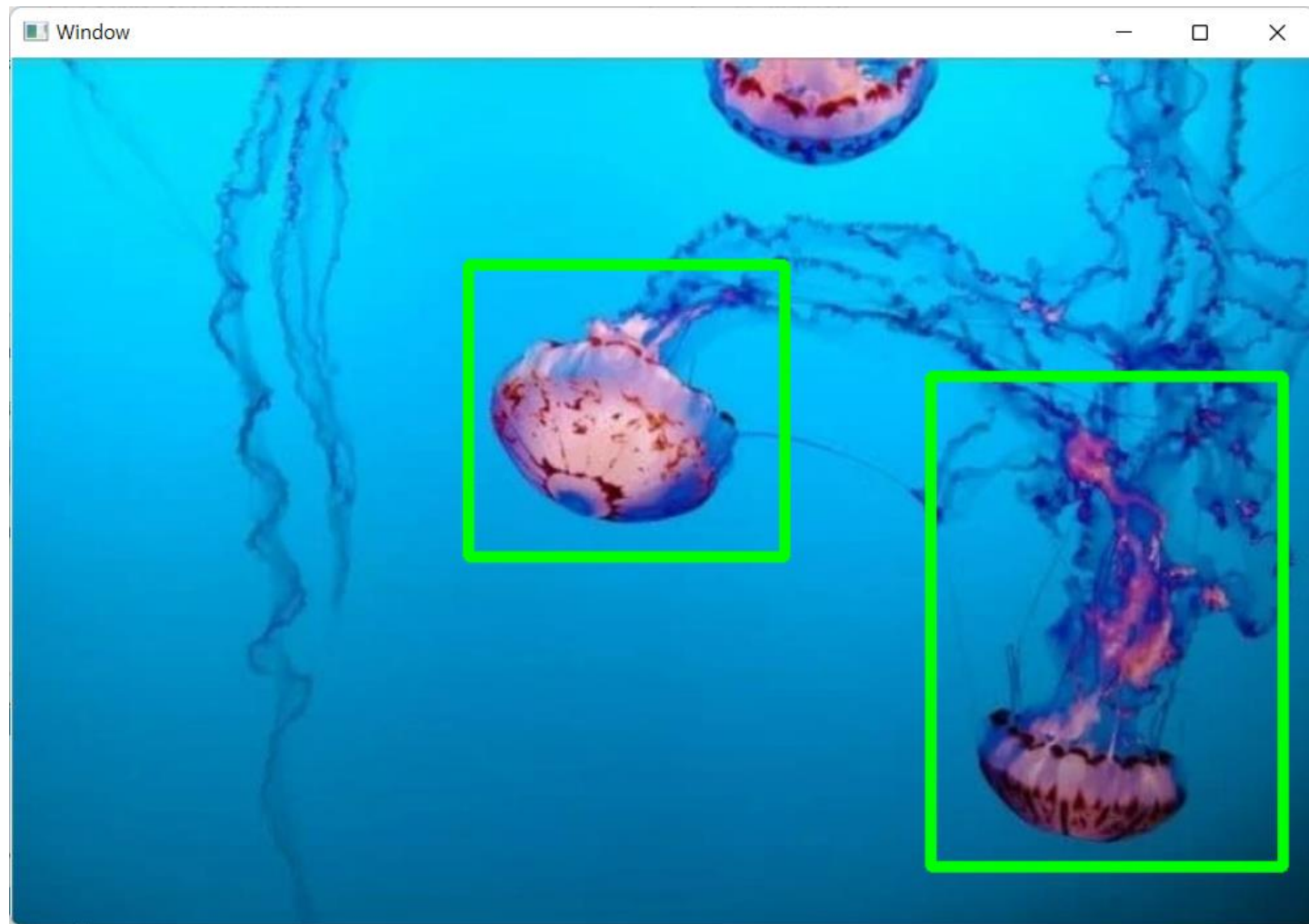


<https://copycoding.tistory.com/146?category=1042125>

[마우스로 드래그한 영역 편집]

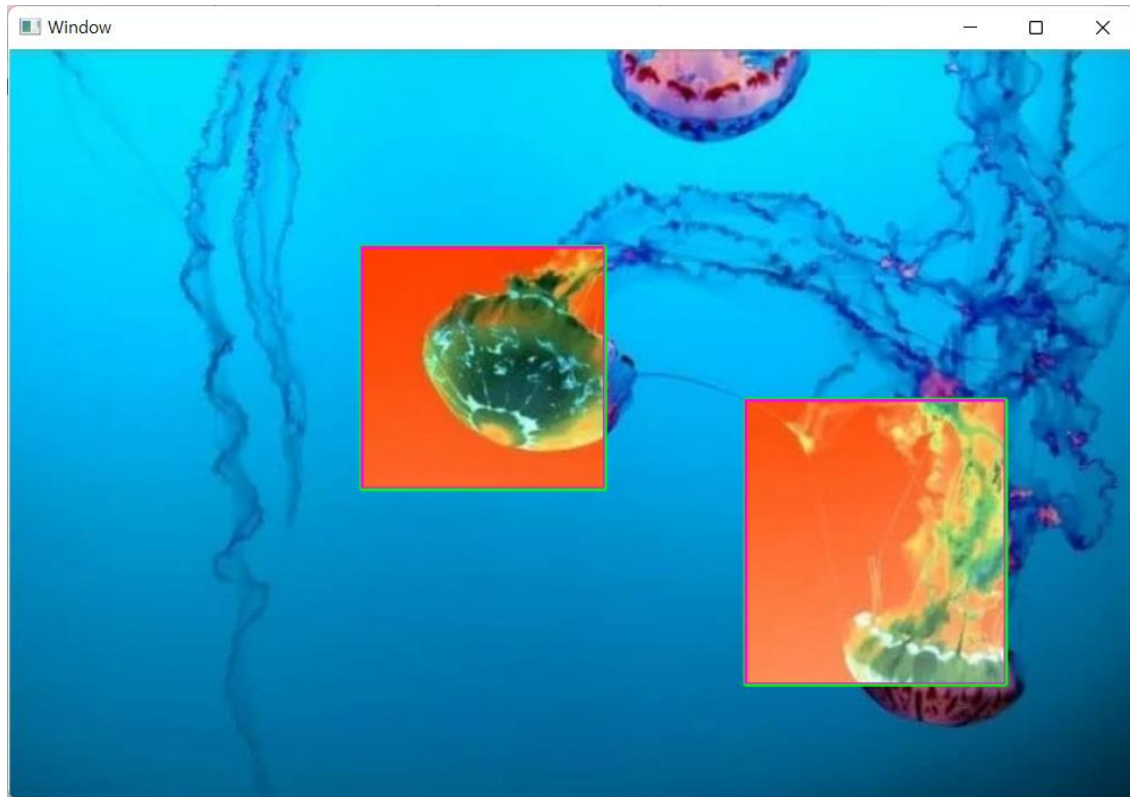
01 마우스로 사각형 그리기.ipynb

1. 마우스로 사각형 그리고
2. 영역지정 위치 csv로 저장

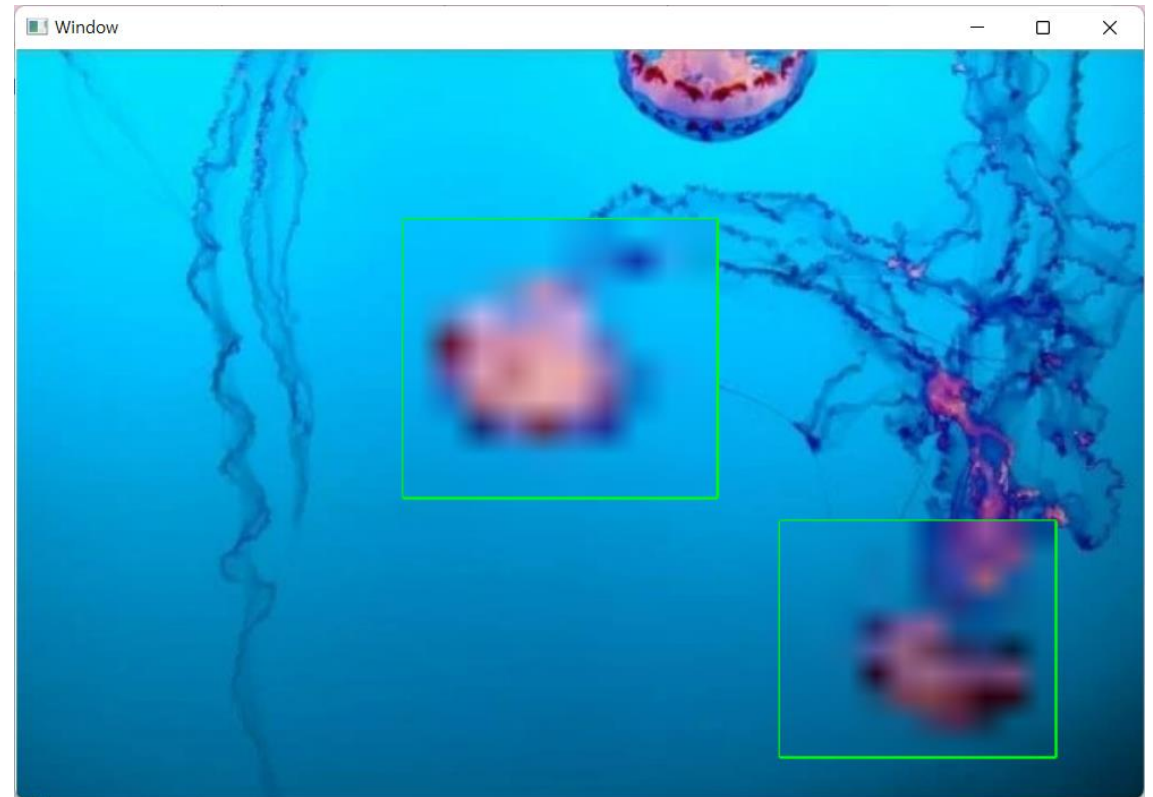


[마우스로 드래그한 영역 편집]

영역지정해서 필터 적용하기



[영역지정해서 모자이크 처리]



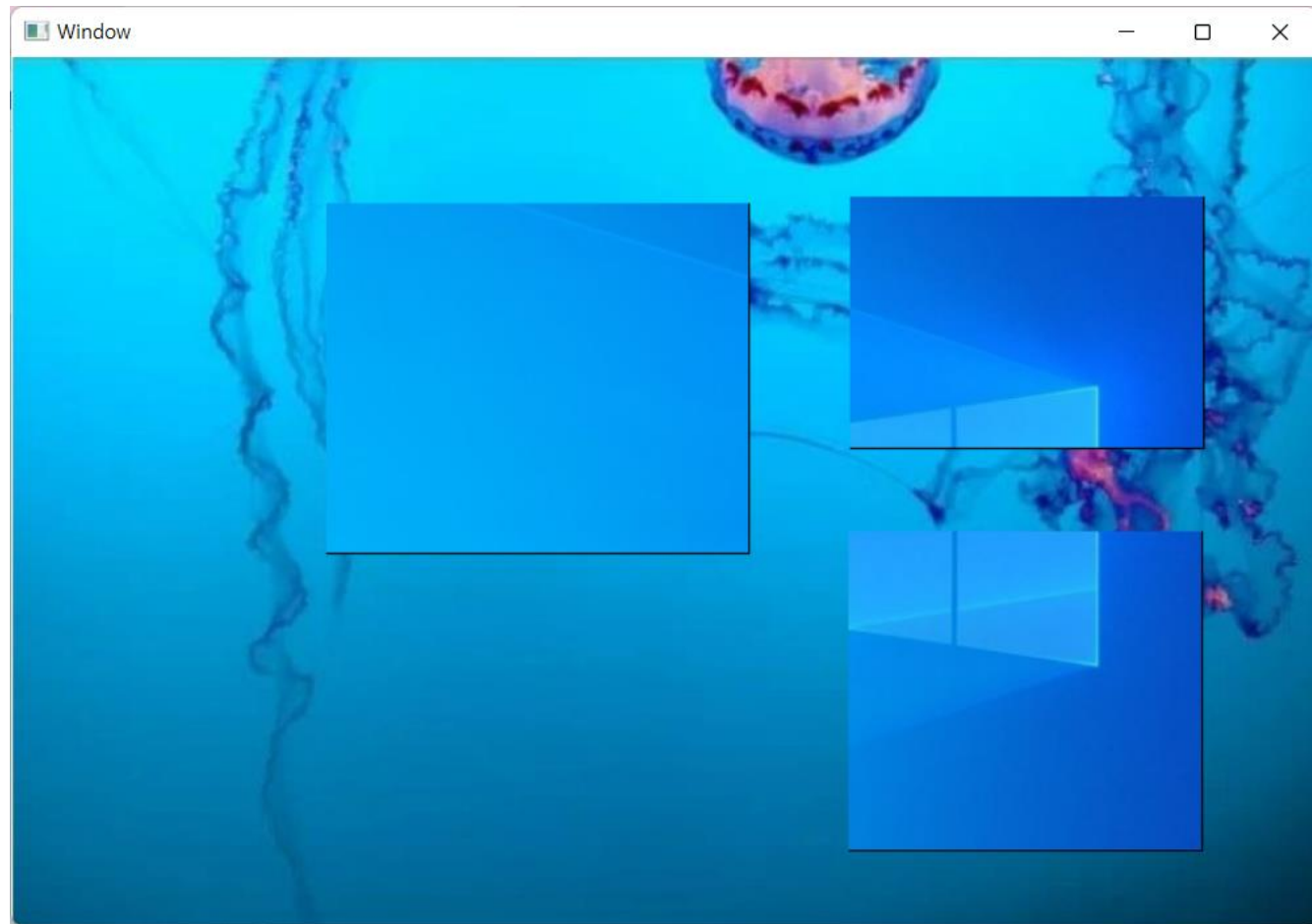
[마우스로 드래그한 영역 편집]

같은 위치에 다른 이미지로 채우기

이미지1



이미지2



[마우스로 드래그한 영역 편집]

이미지 복원

