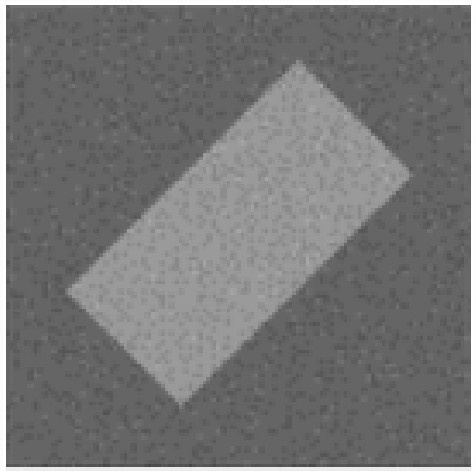
OpenCV를 활용한 이미지 처리3

무엇을 해볼까요?

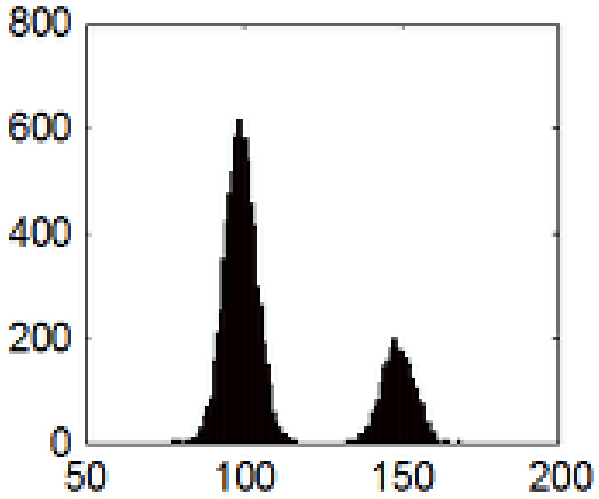
- pixel과 image
- 특정 색상을 찾아라
- **영상 2진화 및 ROI 영역**
- **이미지 Convolution과 이미지 블러링(Blur)**
- **이미지에서 특징 추출(corner, line)**
- Template Matching으로 물체인식
- 숨은 그림 찾기
- 이미지 변환(이동, 회전, Affine, Perspective Transform)

무엇을 해볼까요?

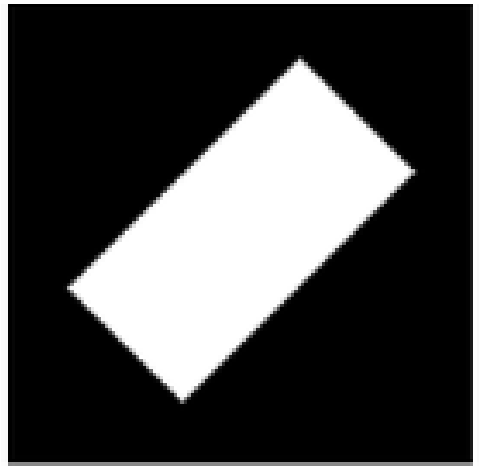
- 영상 2진화
  - 어떤 주어진 임계값(threshold)보다 밝은 픽셀들은 모두 흰색으로, 그렇지 않은 픽셀들은 모두 검은색으로 바꾸는 것을 지칭한다.



(a)



(b)



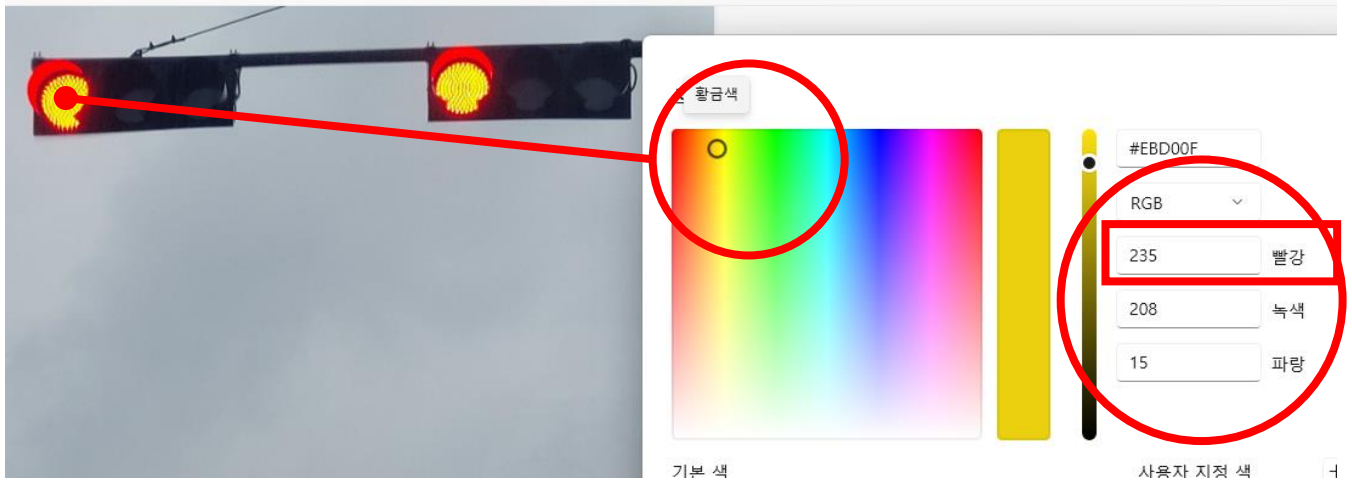
(c)

무엇을 해볼까요?

- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?

무엇을 해볼까요?

- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?



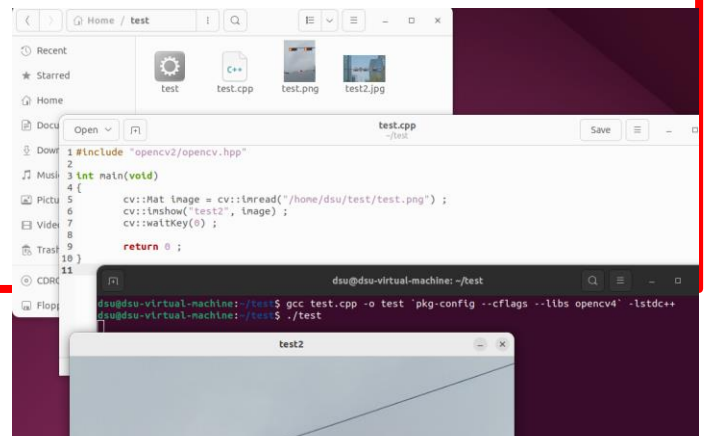
무엇을 해볼까요?

- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?

다시한번 이미지를 읽어 **cv::imshow를 통해 화면에 표시**하자.

```
#include "opencv2/opencv.hpp"

int main(void)
{
        cv::Mat image = cv::imread("/home/dsu/test/test.png") ;
        cv::imshow("test2", image) ;
        cv::waitKey(0) ;

        return 0 ;
}
```

경로 주의



무엇을 해볼까요?

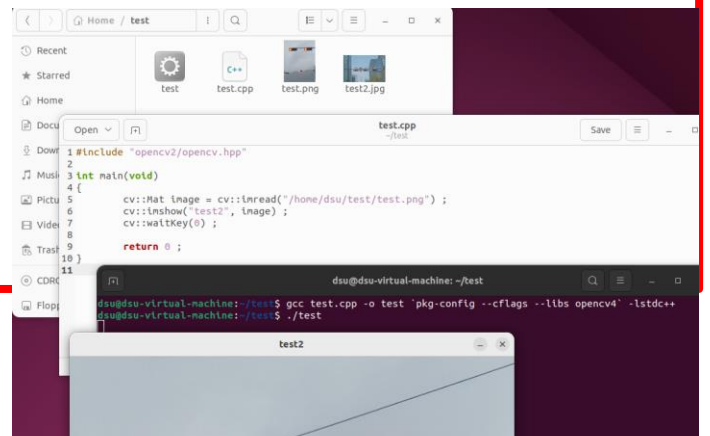
- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?

다시한번 이미지를 읽어 **cv::imshow를 통해 화면에 표시**하자.

```
#include "opencv2/opencv.hpp"

int main(void)
{
        cv::Mat image = cv::imread("/home/dsu/test/test.png") ;
        cv::imshow("test2", image) ;
        cv::waitKey(0) ;

        return 0 ;
}
```

CPP파일 컴파일

```
$ gcc test.cpp -o test `pkg-config --cflags --libs opencv4` -lstdc++
```



무엇을 해볼까요?

- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?

실행

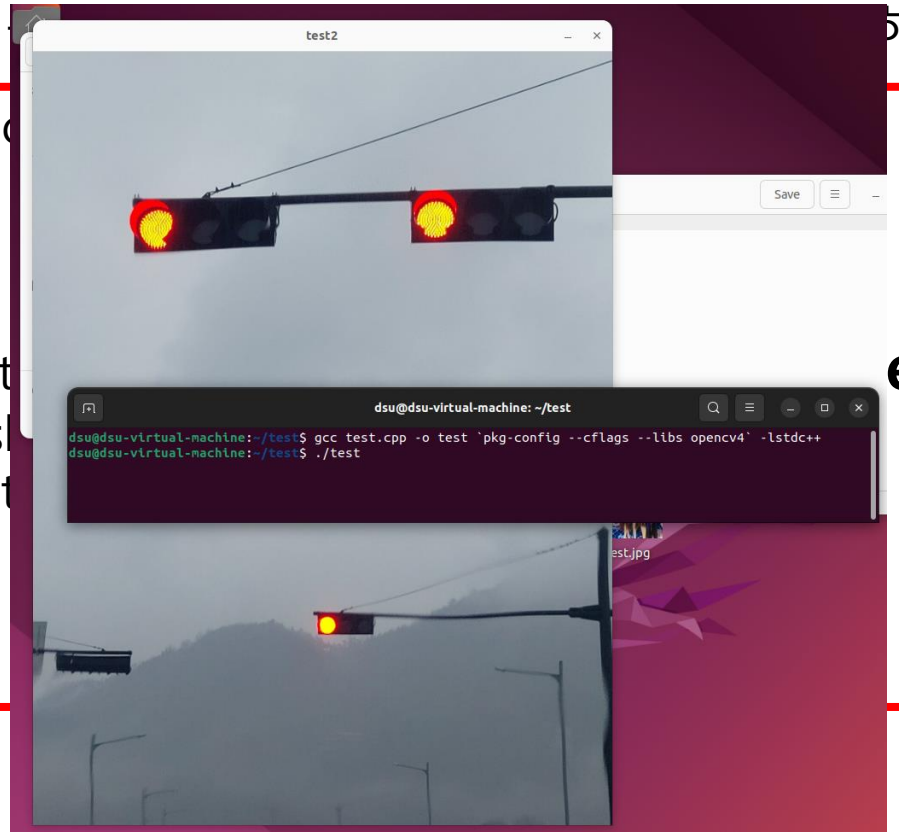
```
$ ./test
```

다시한번 이미지

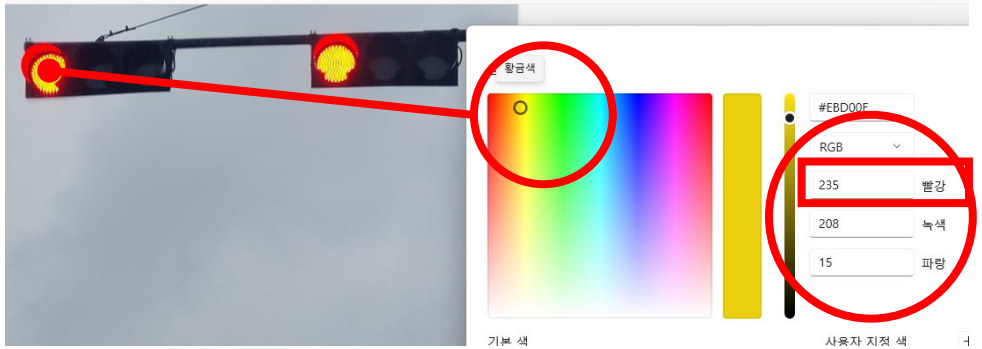
하자.

```
#include "openc

int main(void)
{
        cv::Mat
        cv::ims
        cv::wait

        return
}
```

est/test.png") ;



무엇을 해볼까요?



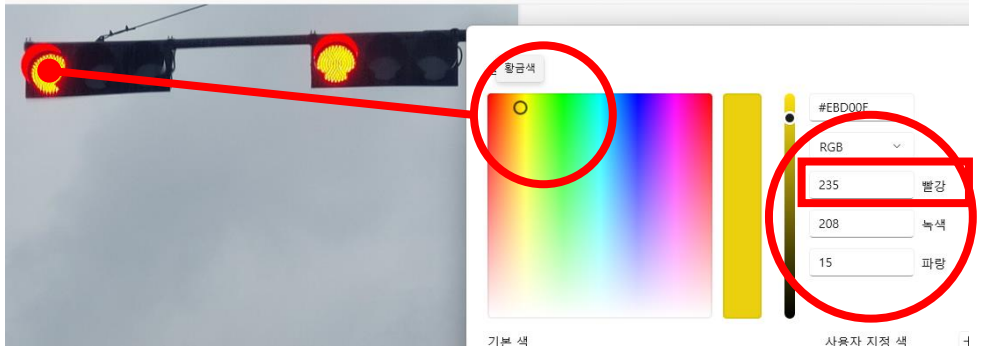
- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?

Red값이 큰 값을 갖는 pixel만 남기고 나머지는 모두 RGB(0,0,0) 으로 만들자

```
if( red_value > 220  )
{
        image.data[red_index] = 255 ;   //red
        image.data[green_index] = 255 ;         //green
        image.data[blue_index] = 255 ;  //blue
}
else
{
        image.data[red_index] = 0 ;     //red
        image.data[green_index] = 0 ;   //green
        image.data[blue_index] = 0 ;    //blue
}
```

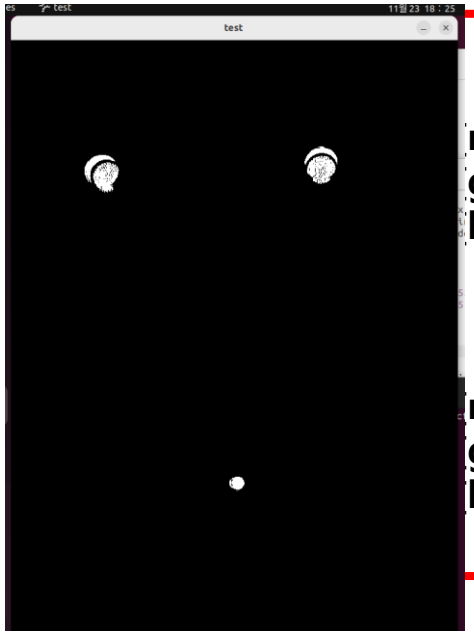
예제) 09_opencv/07_red_bin.cpp

무엇을 해볼까요?



- 영상 2진화
  - 아래의 이미지에서 신호등이 정지 신호인지 어떻게 알 수 있지?

Red값이 큰 값을 갖는 pixel만 남기고 나머지는 모두 RGB(0,0,0) 으로 만들자




```
[red_index] = 255 ;   //red
[green_index] = 255 ;          //green
[blue_index] = 255 ;  //blue


[red_index] = 0 ;     //red
[green_index] = 0 ;   //green
[blue_index] = 0 ;    //blue
```

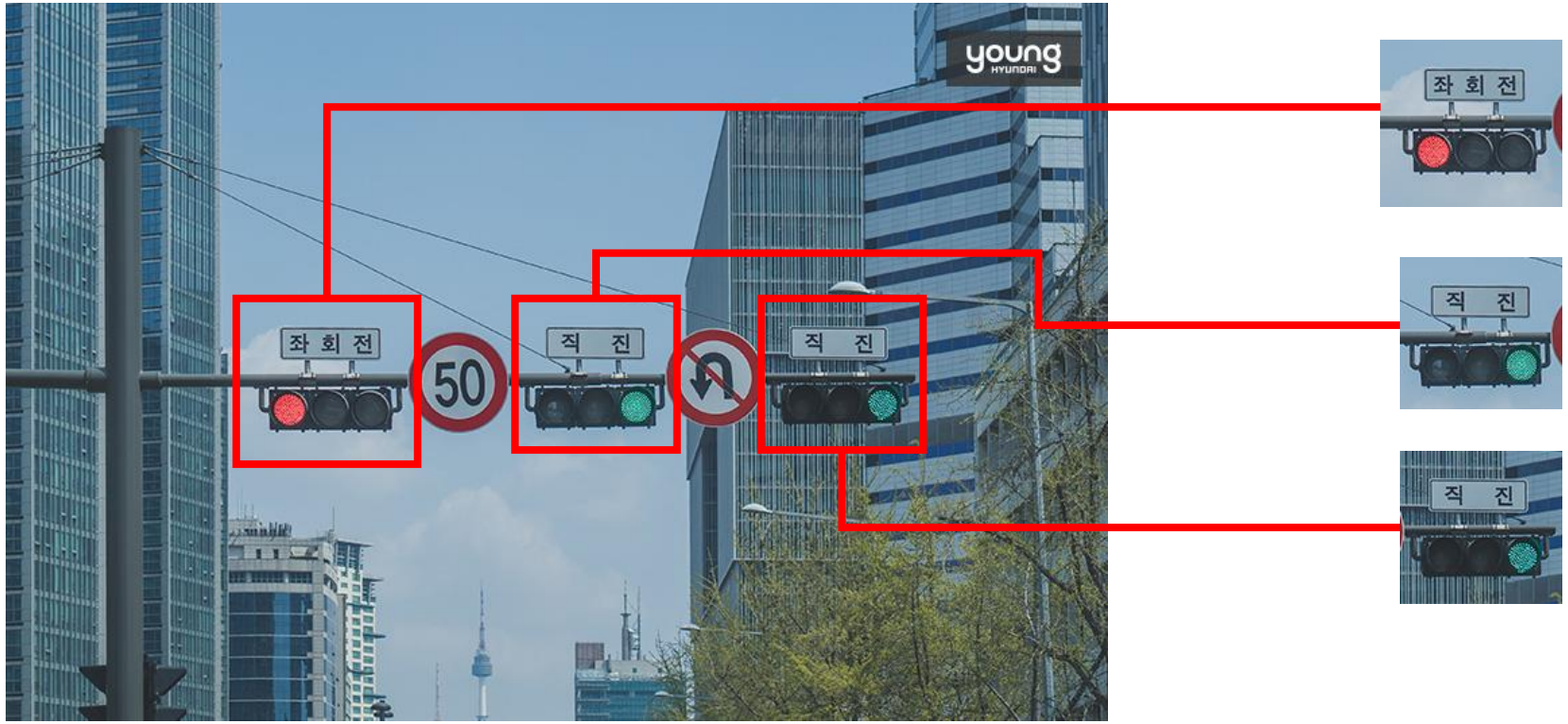
예제) 09_opencv/07_red_bin.cpp

무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역)
  - ROI 란? 말 그대로 **영상 내에서 관심이 있는 영역**을 뜻합니다.

무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역)
  - ROI 란? 말 그대로 **영상 내에서 관심이 있는 영역**을 뜻합니다.

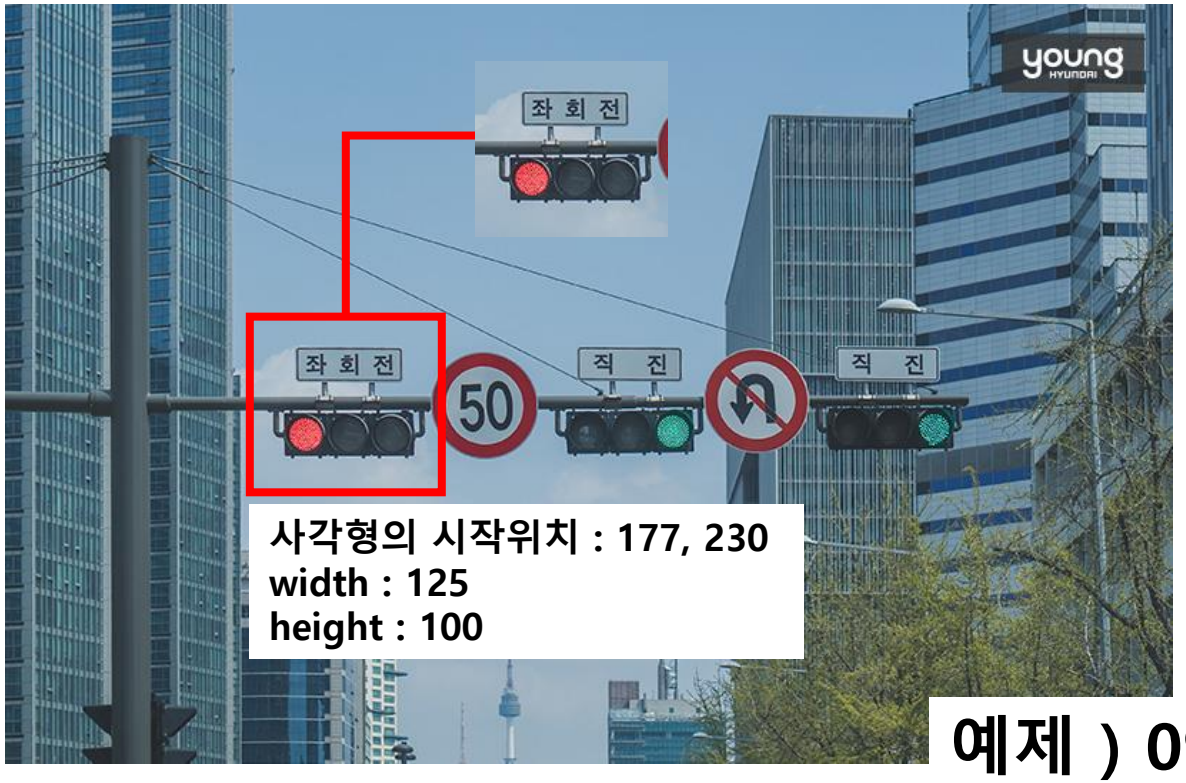


무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역)
  - 이미지에서 신호등 이미지만 잘라보자.

무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역)
  - 이미지에서 신호등 이미지만 잘라보자.

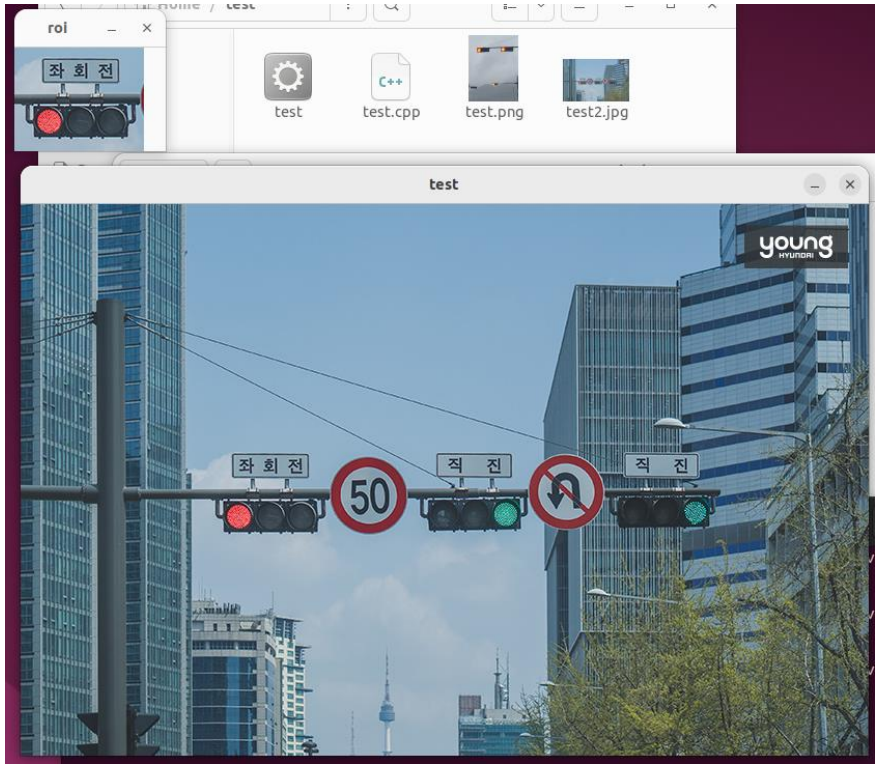



```
cv::Rect roi ;
roi.x = 177 ;
roi.y = 230 ;
roi.width = 125 ;
roi.height = 100 ;

cv::Mat roi_image ;
image(roi).copyTo(roi_image) ;
```

예제) 09_opencv/08_roi_image.cpp

무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역)
  - 이미지에서 신호등 이미지만 잘라보자.

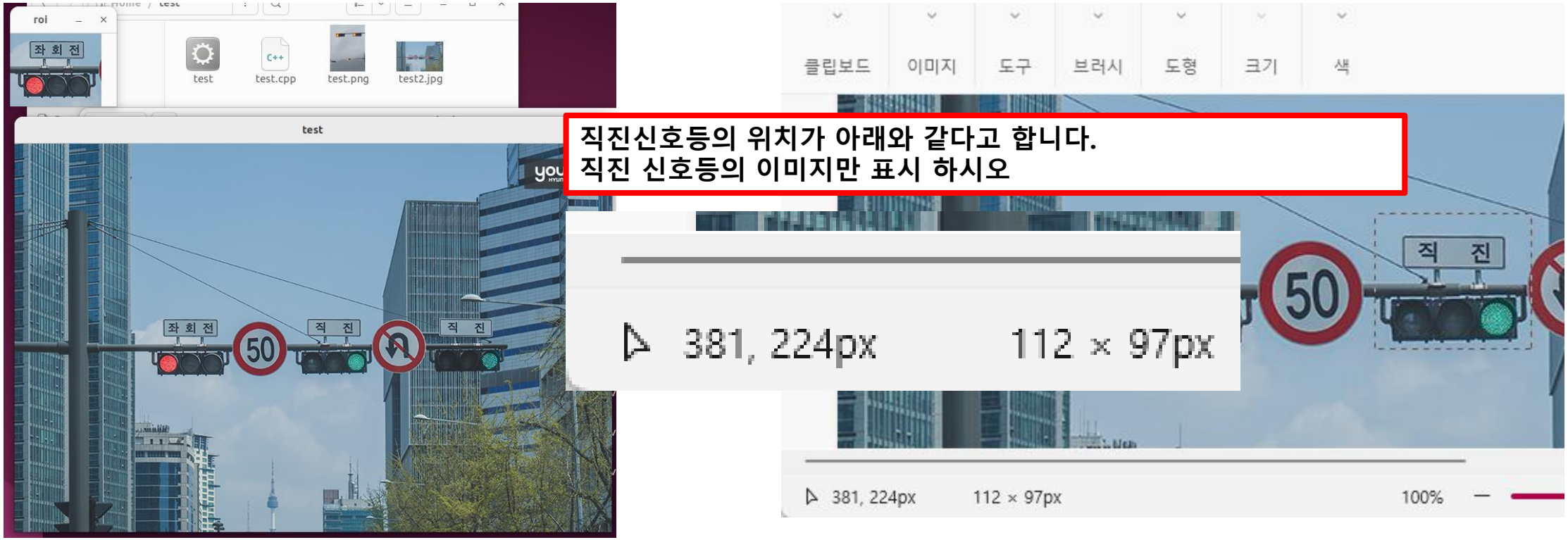



```
cv::Rect roi ;
roi.x = 177 ;
roi.y = 230 ;
roi.width = 125 ;
roi.height = 100 ;

cv::Mat roi_image ;
image(roi).copyTo(roi_image) ;
```

예제) 09_opencv/08_roi_image.cpp

무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역)
  - QUIZ) 이미지에서 직진 신호등 이미지만 잘라보자.



무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역) 이미지에서 영상처리
  - ROI 이미지에서 신호등 색상을 분석하자.

관심 영역 안에서만 처리하기 때문에
전체 영역을 검색하는 것이 아니라 속도가 빠르고 정확도가 높다.

무엇을 해볼까요?

- 이미지 ROI(Region of interest 관심 영역) 이미지에서 영상처리
  - ROI 이미지에서 신호등 색상을 분석하자.


```
cv::Rect roi ;
roi.x = 177 ;
roi.y = 230 ;
roi.width = 125 ;
roi.height = 100 ;

cv::Mat roi_image = image(roi) ;

if( red_value > 220 )
{
        roi_image.data[red_index] = 255 ;       //red
        roi_image.data[green_index] = 255 ;     //gree
        roi_image.data[blue_index] = 255 ;      //blue
}
else
{
        roi_image.data[red_index] = 0 ;         //red
        roi_image.data[green_index] = 0 ;       //green
        roi_image.data[blue_index] = 0 ;        //blue
}
```

예제) 09_opencv/09_roi_image_bin.cpp

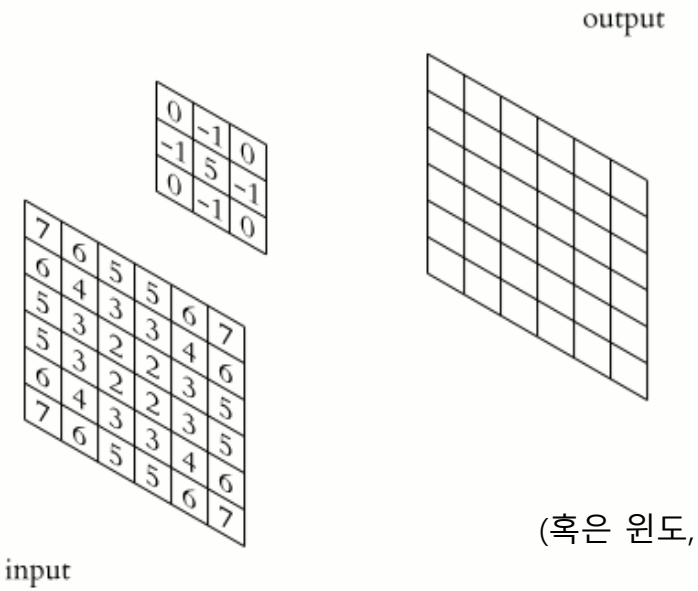
무엇을 해볼까요?

- ROI는 그럼 어떻게 알수있지?? : 물체 검출 알고리즘을 이용
  - YOLO

https://pjreddie.com/darknet/yolo/

무엇을 해볼까요?

- 이미지 Convolution
  - Filter(필터)와 Convolution(컨볼루션)
  - 컨볼루션 연산은 공간 영역 필터링을 위한 핵심 연산 방법
  - 커널을 이용한 콘볼루션 계산을 통해 새로운 정보로 만드는 방법

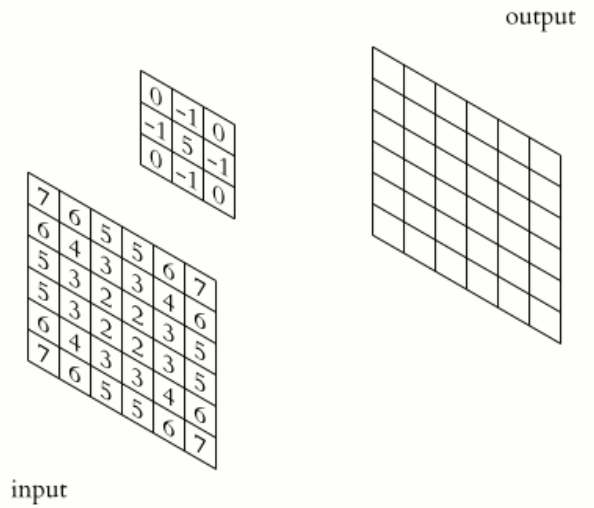


$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

커널(kernel)
(혹은 윈도, 필터, 마스크라고도 부름)

무엇을 해볼까요?

- 이미지 Convolution
  - Filter(필터)와 Convolution(컨볼루션)
  - 평균 필터를 적용해 보자



$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

$\frac{1}{9}\times$

1	1	1
1	1	1
1	1	1

$\frac{1}{25}\times$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

평균 커널(kernel)
(혹은 윈도, 필터, 마스크라고도 부름)