

OPEN_CV DAY1

과제 1번 보고서

2023741024
로봇학부 박건후

목차

- (1) 코드 설명
- (2) 필터 적용 후 달라지는 점과 왜
그 필터를 사용하는지

(1) 코드 설명

```
#include <opencv2/opencv.hpp>
using namespace cv;
```

위 부분은 hpp파일의 헤더파일에서 인클루드하는 부분입니다. open_cv 헤더 파일을 인클루드하면 cv라는 네임스페이스에 정의되어 있는 라이브러리를 사용하게 되는데 이 때 cv를 매번 쓰지 않기 위해 using namespace를 사용했습니다.

```
class image_compare{
```

헤더와 구현, 그리고 main파일로 각각 분리하기 위해 위 사진처럼 매우 구조가 간단한 클래스를 만들었습니다. 그 안에는 필터를 적용할 때와 하지 않을 때에 대해 만들어야 할 모든 객체를 private에 정의하고 public으로 생성자와 필터를 적용하여 imshow를 하는 것, 필터를 적용하지 않고 imshow하는 각각의 메서드(no_filter 메서드, yes_filter메서드)를 만들었습니다. 초록색, 파란색, 빨간색 공에 대해 각각 바이너리 이미지를 만들고자 하였습니다.

```
image_compare::image_compare():lower_green(35, 50, 50), upper_green(85, 255, 255), lower_red_1(0, 50, 50)
upper_red_1(10, 255, 255), lower_blue(100, 50, 50), upper_blue(130, 255, 255){
```

위 부분은 생성자 구현부입니다. Scalar로 범위를 생성하기 위해 r,g,b에 대한 각각의 Scalar 객체에 hsv값을 쉽게 넣어주기 위해 멤버 초기화 리스트를 통해 Scalar의 생성자를 호출하였습니다.

```
img = imread(image_path, IMREAD_COLOR); //imread() 메서드를 통해 사진 객체 불러옴
```

imread()함수를 통해 컬러로 이미지를 읽도록 플래그 변수를 작성해준 뒤 파일 경로를 string으로 받기 때문에 절대경로를 string으로 작성하고 이를 인수에 넣어줘서 원본 이미지 객체를 받아왔습니다.

```
cvtColor(img, img_hsv, COLOR_BGR2HSV);
```

cvtColor를 통해 rgb사진을 hsv로 바꾸는 작업을 했습니다. 그래야 색이 더 잘 인식되기 때문입니다. 이 때 COLOR_BGR2HSV라는 플래그 변수로 설정을 하여 img_hsv 이미지 객체를 받아왔습니다.

```
inRange(img_hsv, lower_green, upper_green, green_mask);
```

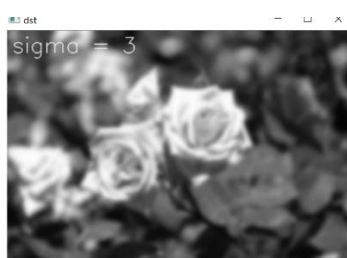
Scalar로 만든 범위인 lower_green, upper_green을 만들어서 초록 범위를 만들고 이 범위 내에 있는 색을 가져와 바이너리 이미지로 만든 부분입니다. 이를 빨강, 파랑에 대해서도 적용해주어 각각의 색에 대한 바이너리 이미지를 만들고 이를 가우시안 필터를 적용한 것과 비교하고자 했습니다.

```
imshow("Green Mask", green_mask);
```

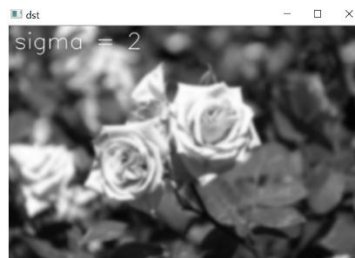
그리고 이렇게 만든 바이너리 이미지를 imshow를 창에 띄우도록 했습니다. 이 작업을 하는 메서드가 no_filter 메서드입니다.

```
GaussianBlur(img, img_blur, Size(19,19), (double)5); //가우시안blur 필터 사용
```

그리고 yes_filter라는 2번째 메서드는 no_filter에서 했던 작업에서 가우시안 필터를 적용하는 작업만 추가된 것입니다. 가우시안 필터를 적용할 때는 커널과 커널을 적용할 표준편차를 작성하게 되는데 커널은 무조건 홀수로만 작성해야 합니다. 또한 인터넷 블로그 (<https://deep-learning-study.tistory.com/144>)를 통해 표준편차가 2일 때와 3일 때의 차이를 아래 사진(좌:시그마 3, 우: 시그마 2)과 같이 경험하였기에 최대한 커널의 범위를 넓게 잡고 많이 흐리도록 표준 편차를 잡아서 작성해주었습니다. 그렇게 한 이유는 흐리게 하면 할수록 노이즈가 덜 낄 것이라고 예상했기 때문입니다.

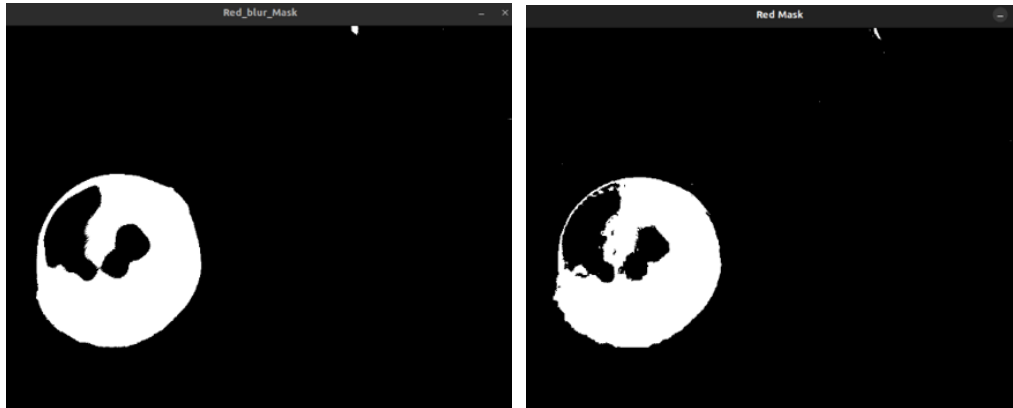


표준편차 = 3



표준편차 = 2

(2) 필터를 적용하는 이유와 왜 그 필터를 사용하는지

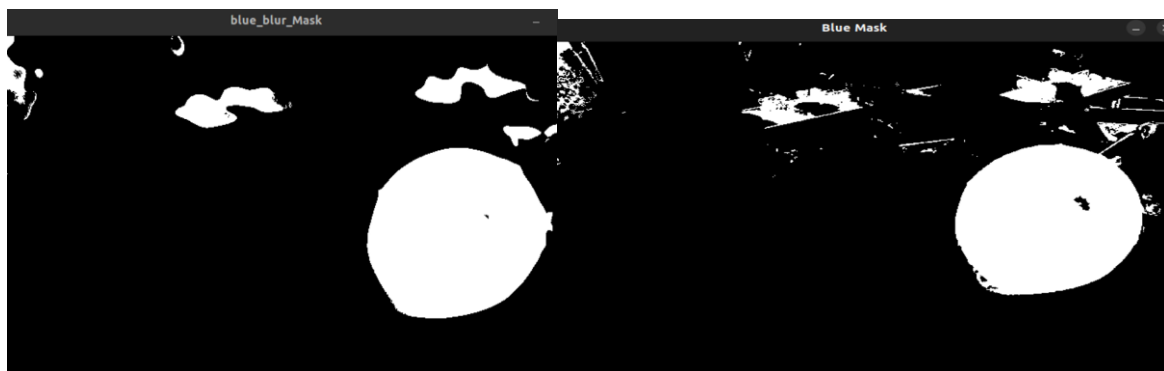


이 부분은 빨간색에 대해 가우시안 필터를 적용했을 때와 안 했을 때의 사진을 나타낸 것입니다. 왼쪽이 필터를 적용한 것이고 오른쪽이 필터를 적용하지 않은 사진입니다. 가우시안 필터를 적용했을 경우에는 경계가 더 부드러워지고 내부가 매끄럽게 채워진 것을 확인할 수 있었습니다. 또한 내부의 점들이 사라진 것을 확인할 수 있었습니다. 이렇게 공 속에 있는 일부 노이즈를 제거할 수 있었습니다.

이렇게 노이즈를 제거해야 나중에 모폴로지 연산이나 다른 후처리 작업을 진행할 때도 안정성을 높여줄 수 있을 것 같습니다.



이번에는 초록색 부분입니다. 이번에도 왼쪽 사진이 훨씬 매끄럽고 우하단의 노이즈가 줄어들며 하얀색 노이즈의 길이 또한 줄어든 것을 볼 수 있었습니다. 또한 공의 내부도 조금 더 채워졌습니다.



마지막으로 파란색에 대한 것입니다. 필터를 적용하지 않은 바이너리 파일에서는 노이즈들이 다양한 곳에 분포했지만 필터를 적용한 것에서는 노이즈가 많이 줄은 것을 확인할 수 있었습니다. 그리고 공 내부 또한 조금 더 채워졌습니다. 감사합니다.