HW1
hpp

```
#include <opencv2/highgui.hpp>
#include <string>
#include <opencv2/opencv.hpp>

using namespace cv;
using namespace std;

class obj {
    private: Mat i,i1;

    public:
    obj(){
        i= imread("/home/david/intern_ws/opencv/day1/src/hw1/src/Lenna.png",IMREAD_COLOR);
        i1= imread("/home/david/intern_ws/opencv/day1/src/hw1/src/Lenna.png",IMREAD_COLOR);
        }
```

Opencv 라이브러리 불러오기(cv_bridge로 Ros에서 가져왔다.)

Cv의 Mat함수로 이미지를 저장하는 변수 생성(i, i1)

생성자 obj

I와 i1에 이미지를 불러온다.(같은 이미지를 부른 이유는 아래에 나온다.)

HW1
hpp

```
    void hsv(){
        cvtColor(i, i, COLOR_BGR2HSV);
        cvtColor(i1, i1, COLOR_BGR2HSV);
    }
    void buler(int a){

            if(a%2==1){
            GaussianBlur(i,i,Size(a,a),0);
            GaussianBlur(i1,i1,Size(a,a),0);
            }
            else{
                cout<<"홀수 or 양수"<<endl;
            }

    }
    void range(int mh, int ms, int mv,int hh, int hs, int hv,int mh1=0, int ms1=0, int mv1=0,int hh1=0, int hs1=0, int hv1=0){
        inRange(i,Scalar(mh,ms,mv),Scalar(hh,hs,hv),i);
        inRange(i1,Scalar(mh1,ms1,mv1),Scalar(hh1,hs1,hv1),i1);
        i+=i1;
    }
    void show(const char name [200] ){
        imshow(name,i);

    }
};
```

변수를 BGR에서 HSV로 변환하는 함수를 hsv 함수로 만든다.

가우시안 블러를 처리하는 변수를 bluer 함수로 만든다.
값 a를 받아 얼마나 블러 처리를 할지 정할 수 있다.

가우시안 블러 특성상 중심이 있어야 하기에 홀수일 경우에만 블러처리가 된다.
블러는 axa 사이즈로 처리된다. 그 이유는 전반적인 노이즈를 제거하기 위해서다.
또한 블러 모드가 0인 이유도 최소한 실제로 존재하는 데이터를 기반으로 테두리 블러처리를 하는 것이 더 성능적으로 이점이 있을 것 같아 사용했다.

예외처리

in range를 이용하여 바이너리 이미지로 만든다.
같은 파일을 2번 만든 이유는 빨간색과 같은 색상은 160~179, 0~20에 분포하여 이를 합쳐서 바이너리를 사용할 수 있도록 만든 것이다.
바이너리는 0과 255만 있기에 +을 이용해서 or연산을 할 수 있기에 이렇게 만들었다.(최댓값 초과시 최댓값으로 맞춰짐)

Show 함수를 이용하면 imshow함수를 이용해 처리된 사진에 창 이름을 정하여 화면에 띄울 수 있게 만들었다.

HW1
cpp

```cpp
#include"../include/hw1/hw1.hpp"


int main(){

    obj row;
    row.show("row");
    row.buler(51);
    row.show("bluer");

    obj ir;
    ir.hsv();
    ir.range(150,50,50,180,255,255,0,50,50,10,255,255);
    ir.show("inrageR");

    obj ig;
    ig.hsv();
    ig.range(30,100,150,70,255,255);
    ig.show("inrageG");

    obj ib;
    ib.hsv();
    ib.range(95,230,150,125,255,255);
    ib.show("inrageB");
```

Row 객체를 생성하여 기존 사진과 블러된 사진을 띄운다.

Ir 객체를 생성해
사진을 hsv로 바꾼 후
바이너리 이미지로 바꾼다.
이때 ir은 빨강이기에 2가지의 영역을 설정하여 합친다.
합친 바이너리를 화면에 띄운다.

초록색 또한 똑같이 한다.

파란색 또한 똑같다.

HW1
cpp

```
    obj ibr;
    ibr.hsv();
    ibr.buler(51);
    ibr.range(150,50,50,180,255,255,0,50,50,10,255,255);
    ibr.show("inragebR");

    obj ibg;
    ibg.hsv();
    ibg.buler(51);
    ibg.range(30,100,150,70,255,255);
    ibg.show("inragebG");

    obj ibb;
    ibb.hsv();
    ibb.buler(51);
    ibb.range(95,230,150,125,255,255);
    ibb.show("inragebB");

    waitKey(0);
}
```

Ibr 객체또한 ir과 똑같이 hsv로 바꾼다.
하지만 가우시안 블러에 51 크기 만큼 적용하여 흐리게 만든 후

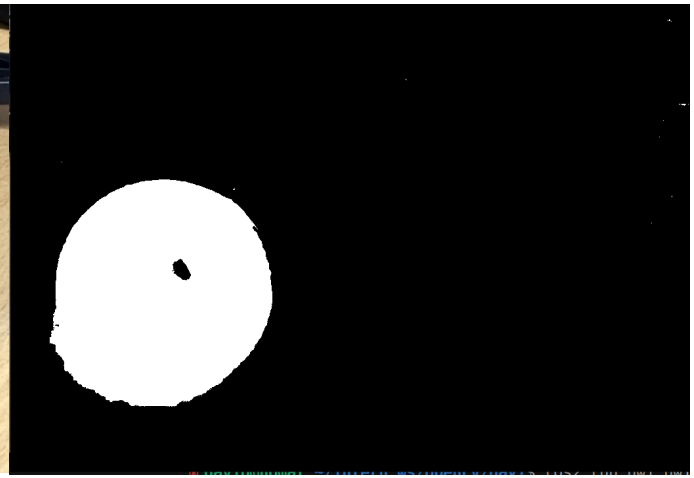
바이너리 이미지로 변환하고
화면에 띄운다.

이후 초록, 파랑도 똑같이 한다.

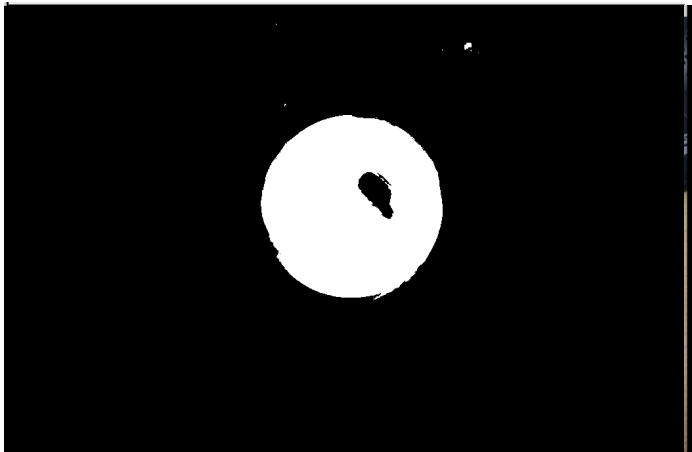
키가 입력되기 전까지 무한정 기다려
화면에 원본, 빨강 바이너리, 초록 바이너리, 파랑 바이너리와
가우시안 블러된 원본/ 빨강 ,초록, 파랑 바이너리 이미지를 띄운다.

원본

빨강



초록



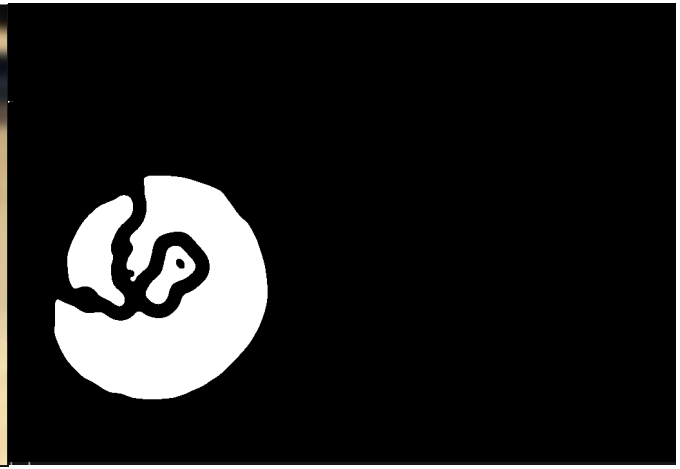
파랑

HW1
가우시안 X

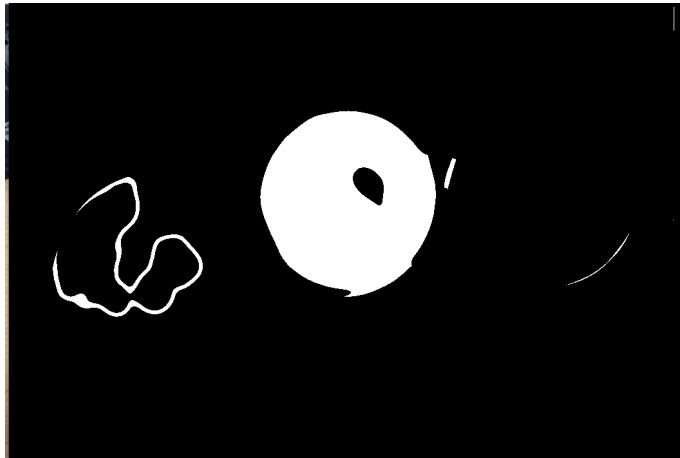
가우시안을 사용하지 않은 바이너리는
대체적으로 감지영역이 거칠며
세밀한 바이너리영역 설정이 없으면
다른 물체의 일부도 바이너리에
포함되어 점형식의 노이즈가
껴지는 것을 알 수 있다.

원본

빨강



초록

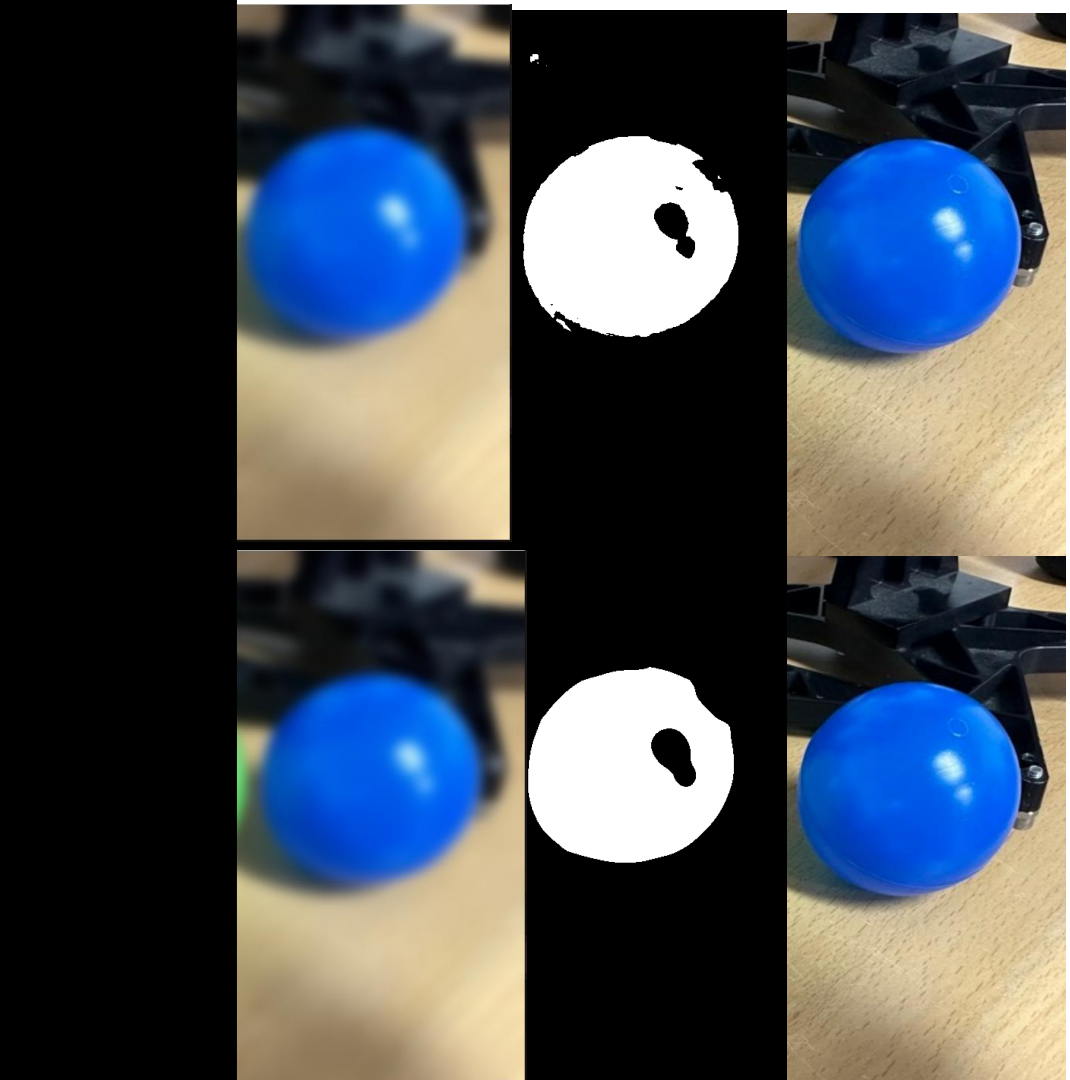


파랑

HW1
가우시안 O

가우시안을 사용한 바이너리는
영역 경계가 대체적으로 매끄러우며
가우시안 없는 빨강에 경우 바닥의
일부가 영역에 포함되었지만
가우시안 적용 후엔 해당 영역이
매우 줄어들었다.
또한 전반적인 외부 물체에 대한
노이즈도 거의 제거되었다.

하지만 가우시안을 적용하여 오히려
노이즈가 생긴 경우도 보인다.

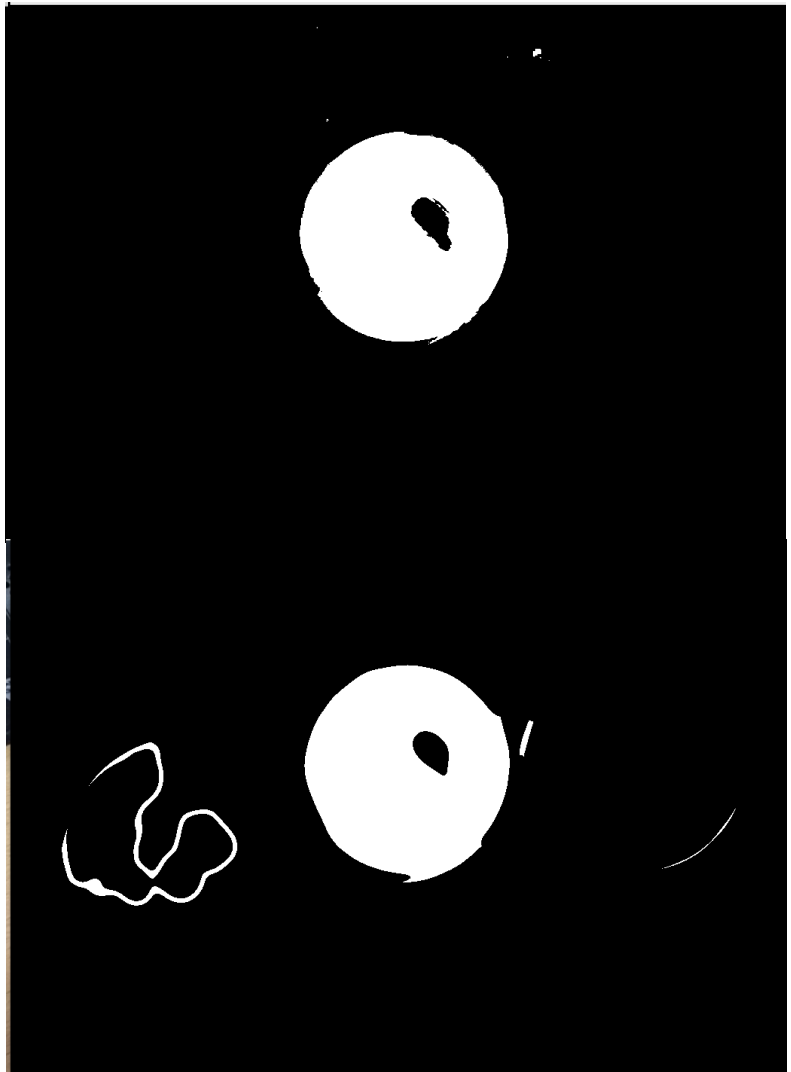


HW1
가우시안 장점

가우시안 적용 후 가장 좋은 결과를 보여준 색은 파란색이다.

표면이 매끄러워졌으며
파란색 이외의 영역에서 탐지되던 영역도 모두 제거되었다.

물론 구체가 기존에 비해 작아져 보이지만
오히려 형태가 구형에 가까워지며
객체 분별하기가 조금이나마 쉬워졌다.



HW1
가우시안 한계

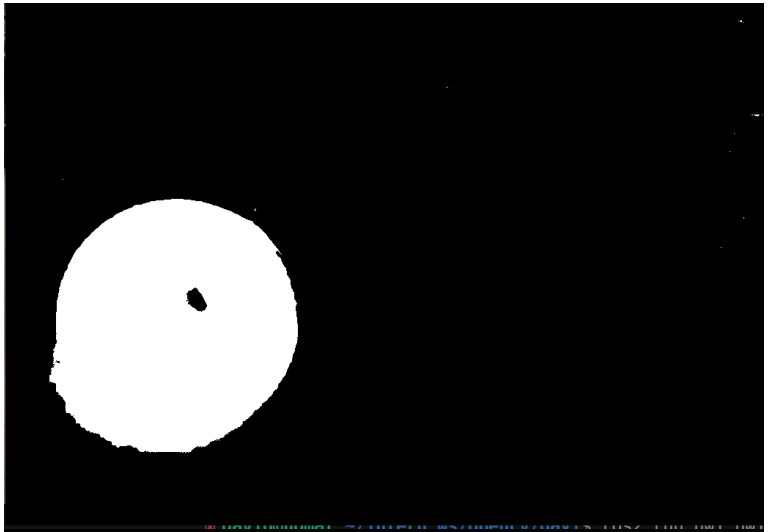
하지만 오히려 안좋아지는 경우도 나왔다.

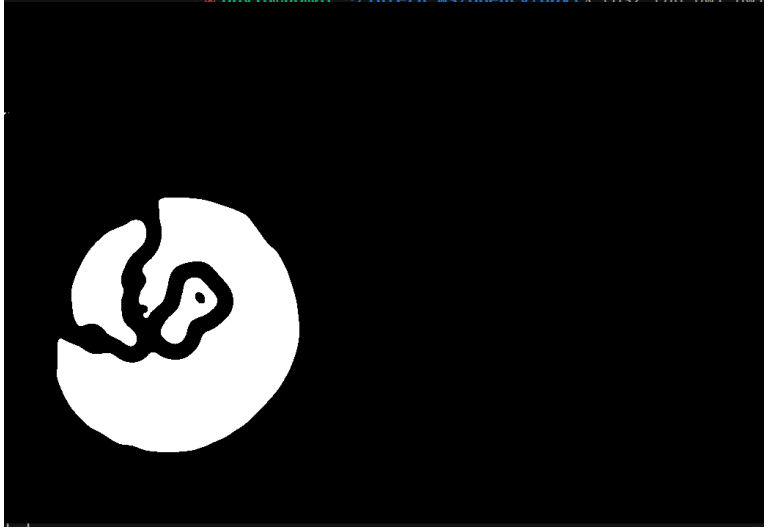
초록색의 경우
빨간색과 유사하게 경계가 매끄러워지며
일부 다른 영역 탐지도 줄어들었지만
다른 공들에서의 영역 탐지가 늘기도 했다.

해당 문제는 바이너리를 조정하면 일부 해결되지만
완전히 해결을 할 수가 없다.

이러한 문제가 발생하는 이유를 생각해 보았을 때
빨간색이라고 해도 현실세계에서 난반사로 인해
빛이 주로 빨간색을 반사하지만
일부 빨간색이 아닌 색도 반사되어있는 사진에서
가우시안 블러로 인해 해당 색상이 흰색과 섞여
다른 색에 영역에 탐지가 되는 것 같다.

그렇게 생각한 이유에 대한 근거가 빨간색과 연결된다.





HW1
가우시안 한계

초록색과 반대로 빨간색의 경우

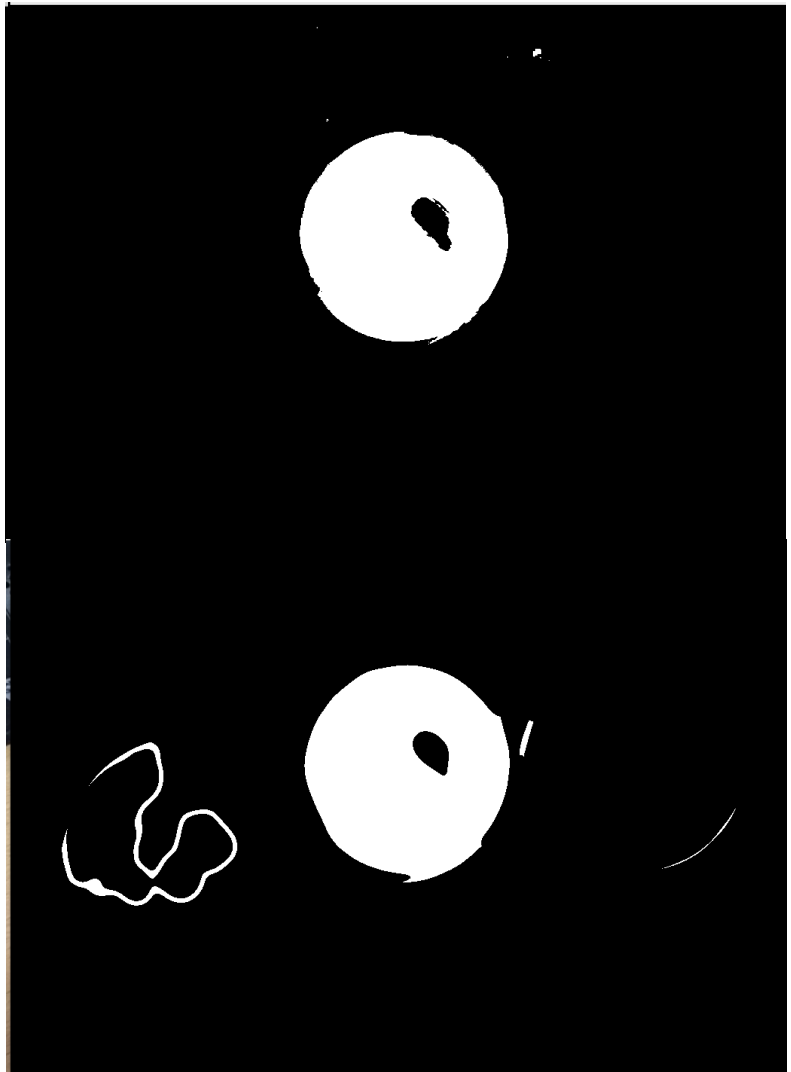
가우시안을 쓰지 않았을 때가 가우시안을 썼을 때보다
더 적은 빨간색 영역탐지가 되는 경향이 나왔다.

가우시안으로 인한 빨간색의 삭제 영역과
초록색 추가 영역을 대조하면 일치한 형상이라는 것을 알 수 있다.

이 뜻은 가우시안이 해당 영역을 빨간색에서 초록색 방향으로
색조를 변형했다는 것으로 해석할 수 있다.

그렇다고 색조 범위를 늘린다면 주황색과 유사한 바닥도
같이 탐지 되어버려 오히려 정확도가 떨어지게 되어
Hsv 조정으로 가우시안의 한계를 보정하는데는 역부족이다.

따라서 이 문제를 해결하려면
모폴로지의 팽창이나 침식이 필요할 것이다.



HW1
가우시안 한계

추가로 빨간색에 비해 파란색이 덜 번지는 이유는
파란색이 반사하는 파장이 초록에 비해 상대적으로 짧다.

따라서 빨강색인 경우 흰색이 섞여 새로 만들어진 색에서
더 높은 색조를 가진 색이 초록색 영역에 탐지가 될 경우가 높지만

파랑과 흰색이 섞여서 새로 만들어진 색은 파랑보다
더 높은 색조가 나오지 그보다 낮은 색조가 나오기 힘들기에
초록색 영역에 덜 탐지되는 것 같다.