



ICE4027 디지털영상처리설계

실습 9주차

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.
2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.
3. 나는 보고서의 내용을 조작하지 않겠습니다.
4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.
5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2023년 05월 03일

학부 정보통신공학

학년 3

성명 김동한

학번 12191727

1. 개요

Homework

실습 및 과제

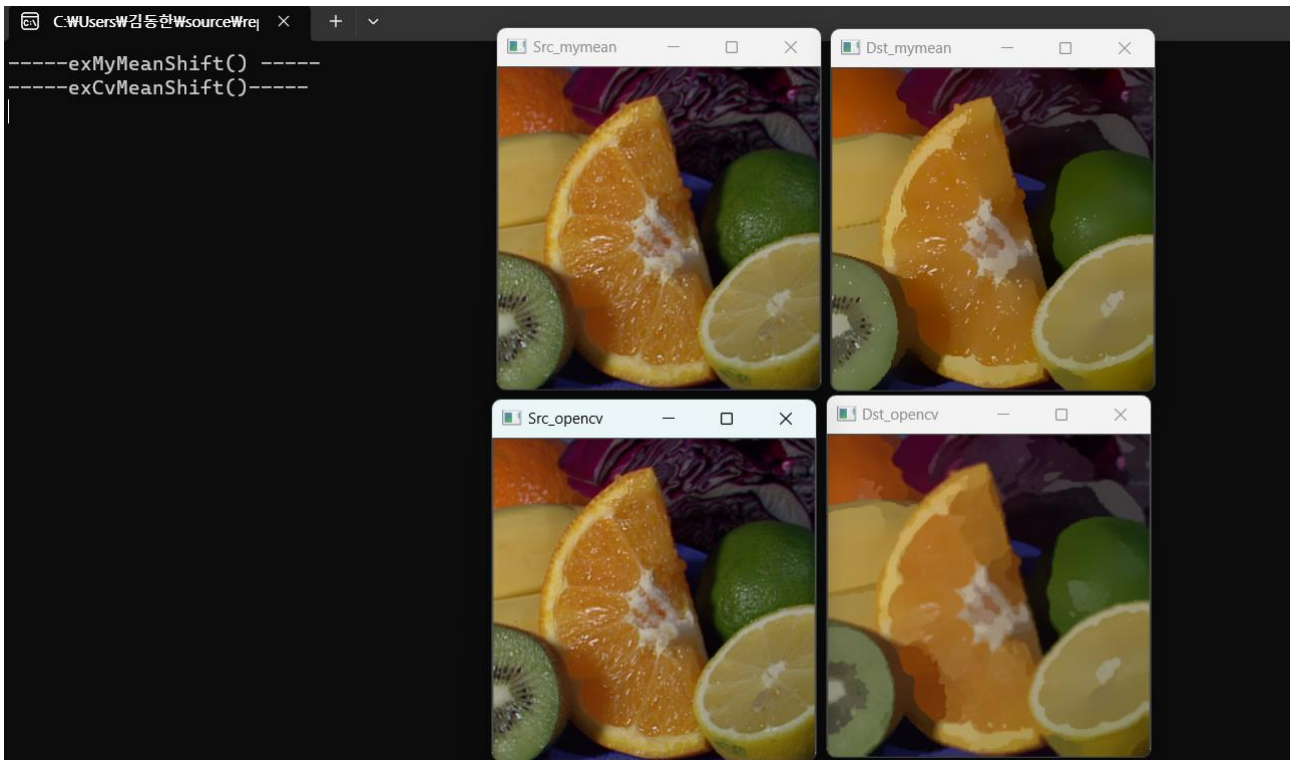
1. Mean-shift OpenCV와 low-level 구현 2개 적용해보고 결과 분석
2. Grab cut의 처리가 **잘 되는 영상과 잘 되지 않는 영상을 찾아 실험**해보고 그 이유를 서술할 것(성의 없는 분석, 완전히 틀리거나 의미 없는 분석은 감점)

1. 구현과정과 결과 분석을 반드시 포함할 것
2. 보고서에도 코드와 실험결과 사진을 첨부할 것
3. 반드시 바로 실행가능한 코드(.cpp)를 첨부할 것



2. 상세 설계 내용

#1



low-level로 구현한 것이 더 잘 분류된것처럼 보이는 이유는 간단하다. opencv를 사용한 예제는 kernel 정한뒤, mean shift를 수행하는데, low-level로 구현한 것은 픽셀 값으로 접근하기 때문에 더 자세하게 segmentation이 이루어진 것이다. 코드는 강의노트의 코드와 동일하여 따로 보고서에 첨부하지않았다.

#2

main)

```
int main() {
    exMyMeanShift();
    exCvMeansShift();

    destroyWindow("Dst_opencv");
    destroyWindow("Src_opencv");

    destroyWindow("Src_mymean");
    destroyWindow("Dst_mymean");

    //#####1 Grabcut 잘 되지않는 이미지
    Mat result_1, bg_model_1, fg_model_1;
    Mat img_1; //src_img
    img_1 = imread("C:\\WWW\\images\\src\\img.jpg", 1);

    Rect rect(Point(160, 150), Point(460, 719));
    grabCut(img_1, result_1,
            rect, bg_model_1, fg_model_1,
            5,
```

```

GC_INIT_WITH_RECT);

compare(result_1, GC_PR_FGD, result_1, CMP_EQ);
//GC_PR_FGD: Grabcut class foreground 픽셀
//CMP_EQ: compare 옵션(equal)

Mat mask(img_1.size(), CV_8UC3, cv::Scalar(255, 255, 255));
img_1.copyTo(mask, result_1);

imshow("grabcut_srcimg_1", img_1);
waitKey(0);
imshow("grabcut_result_1", mask);
waitKey(0);
imshow("grabcut_mask_1", result_1);
waitKey(0);

destroyWindow("grabcut_srcimg_1");
destroyWindow("grabcut_result_1");
destroyWindow("grabcut_mask_1");

//#2 Grabcut 잘 되는 이미지_1
Mat result_2, bg_model_2, fg_model_2;
Mat img_2; //src_img
img_2 = imread("C:\\images\\grabcutex.jpg", 1);
Rect rect_2(Point(60, 65), Point(195, 206));
grabCut(img_2, result_2,
        rect_2, bg_model_2, fg_model_2,
        5,
        GC_INIT_WITH_RECT);

compare(result_2, GC_PR_FGD, result_2, CMP_EQ);
Mat mask_2(img_2.size(), CV_8UC3, cv::Scalar(255, 255, 255));
img_2.copyTo(mask_2, result_2);

imshow("grabcut_srcimg_2", img_2);
waitKey(0);
imshow("grabcut_result_2", mask_2);
waitKey(0);
imshow("grabcut_mask_2", result_2);
waitKey(0);

destroyWindow("grabcut_srcimg_2");
destroyWindow("grabcut_result_2");
destroyWindow("grabcut_mask_2");

///#3 Grabcut 잘 되는 이미지_2
Mat result_3, bg_model_3, fg_model_3;
Mat img_3;
img_3 = imread("C:\\images\\diving.jpg", 1);
Rect rect_3(Point(30, 40), Point(490, 300));
grabCut(img_3, result_3,
        rect_3, bg_model_3, fg_model_3,
        5,
        GC_INIT_WITH_RECT);

compare(result_3, GC_PR_FGD, result_3, CMP_EQ);
Mat mask_3(img_3.size(), CV_8UC3, cv::Scalar(255, 255, 255));
img_3.copyTo(mask_3, result_3);

```

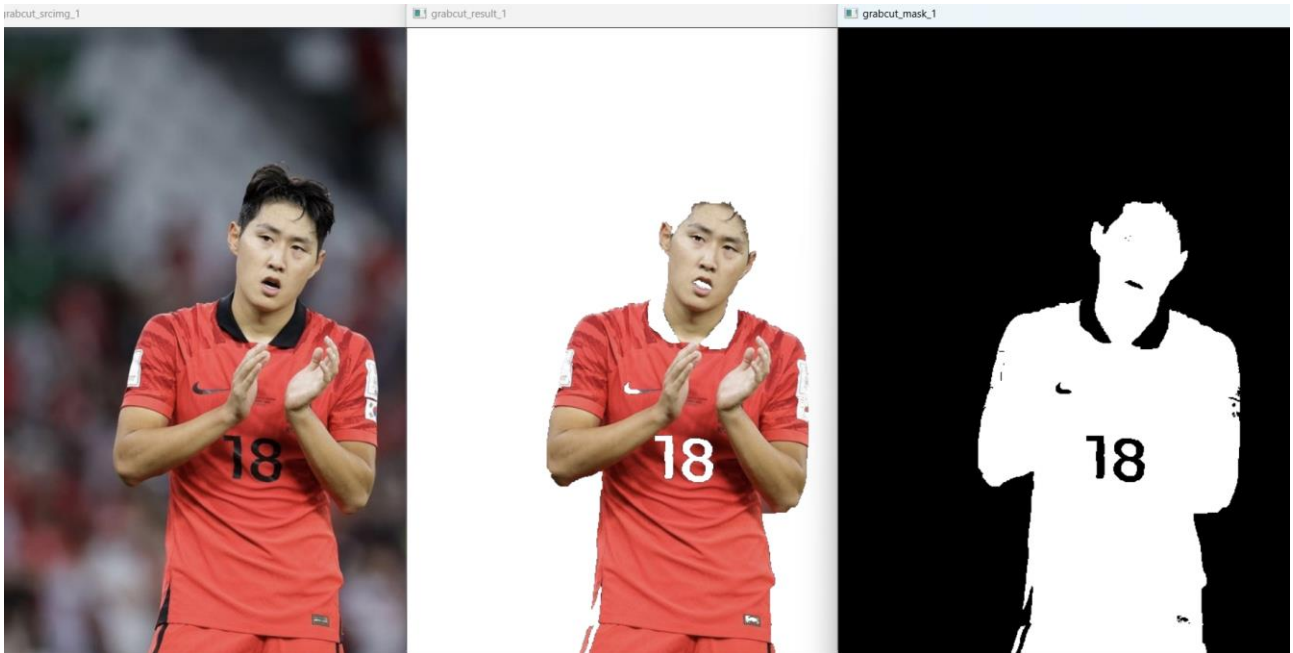
```

imshow("grabcut_srcimg_3", img_3);
waitKey(0);
imshow("grabcut_result_3", mask_3);
waitKey(0);
imshow("grabcut_mask_3", result_3);
waitKey(0);

destroyWindow("grabcut_srcimg_3");
destroyWindow("grabcut_result_3");
destroyWindow("grabcut_mask_3");

}

```



위의 예시는 grabcut이 잘 이루어지지 않은 예시이다. 이강인 선수의 머리색이 배경과 유사한 픽셀값을 가지기 때문에 같이 처리된 것이다. grabcut 알고리즘은 이미지의 전경과 배경을 구분하는 윤곽선이 명확하지 않으면 불분명하게 나오게되는것이다. 아래의 두 영상은 전경과 배경의 구분이 확실한 case이기 때문에 grabcut이 문제없이 이루어진 것을 확인 할 수 있다.

