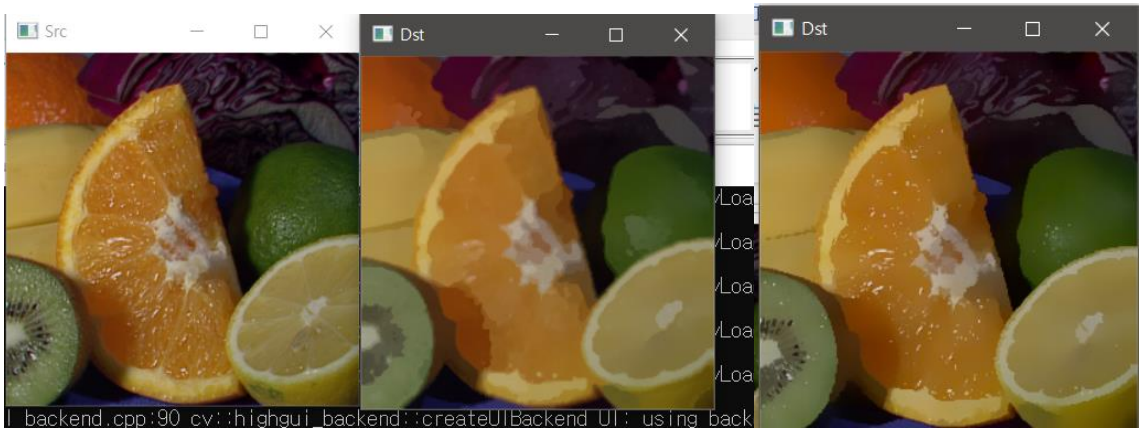
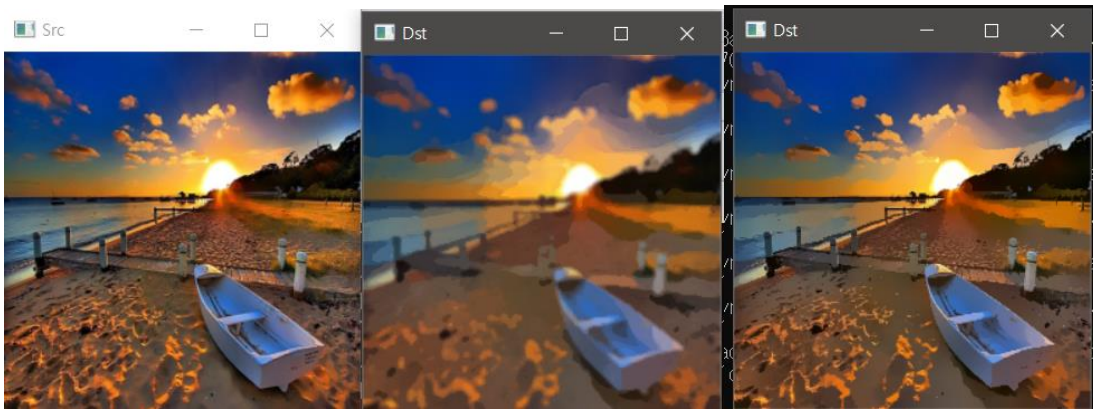
8nd Week Lab Assignment

12201928 이상혁

Task 1

Mean-shift OpenCV와 low-level 구현 2개 적용해보고 결과 분석





가장 맨 왼쪽이 원본사진이며 가운데 사진이 Opencv 에서 내장되있는 함수를 사용한결과, 오른쪽이 low-level 을 통한 구현으로 실행한 결과이다. 결과를 확인하게 되면 직접 구현한 함수가 조금 더 경계가 살아있으며 블러링이 덜 되어 있는 모습을 확인할 수 있다. 즉 노이즈 제거는 opencv 함수가 더 잘 되었으며 부드러운 결과를 지닌다고 볼 수 있다.

객체의 경계는 두 사진다 잘 이루어졌다. 그러나 노이즈를 제거하고 평활화하는 부분은 내장함수가 더 잘된 것 같다. 구현한 함수는 아직 노이즈가 좀 많이 남아있는 것 같다. 블러링이 조금 많이 되있는거 같아 두 함수의 중간정도가 적당하다고 판단된다.

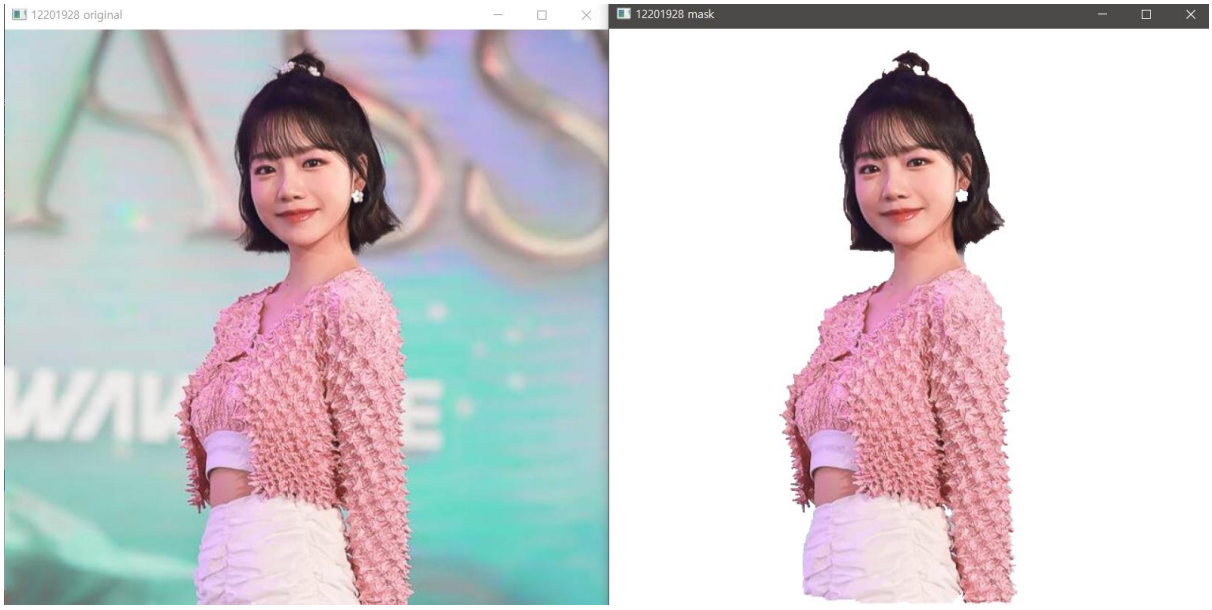
정확히 판별을 하기는 어려울것 같다.

.Meanshift 의 코드는 실습 강의 노트 내용과 동일함으로 생략하겠다.

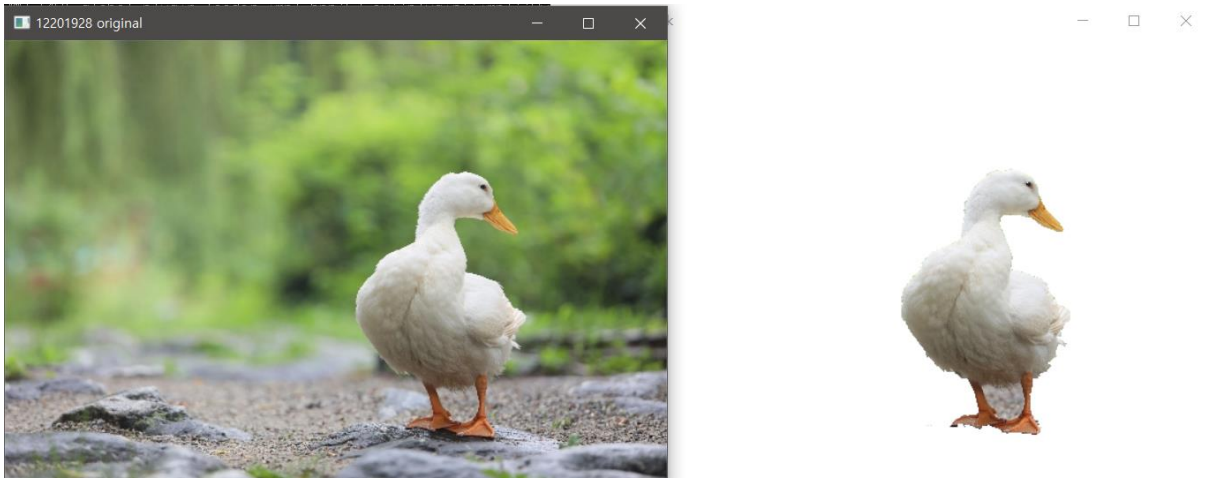
Task 2

Grab cut 의 처리가 잘 되는 영상과 잘 되지 않는 영상을 찾아 실험해보고

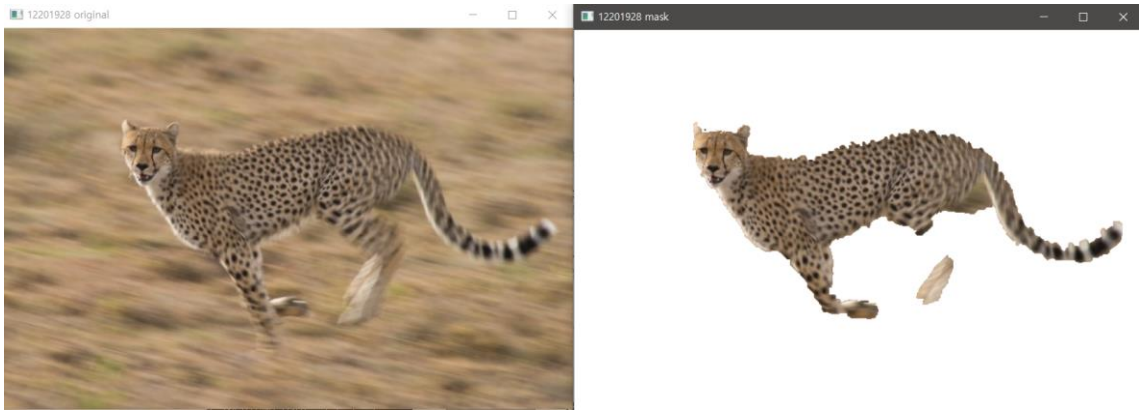
그 이유를 서술할 것(성의 없는 분석, 완전히 틀리거나 의미 없는 분석은 감점)



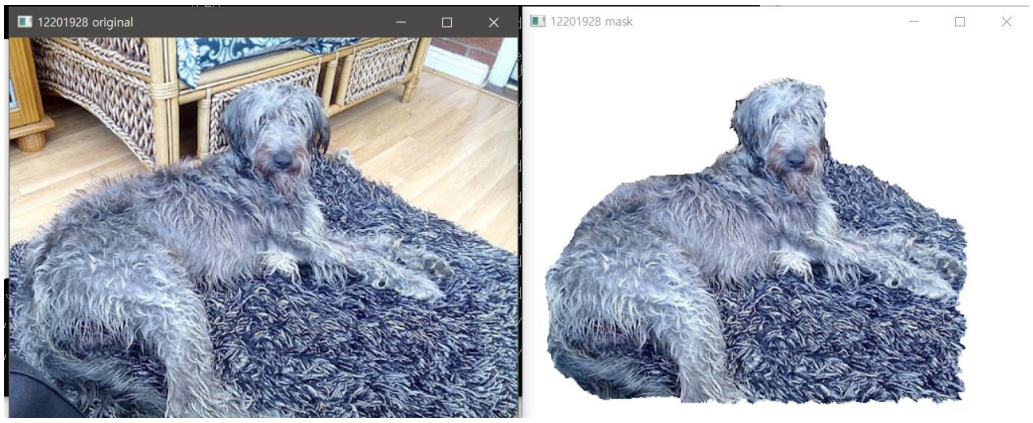
그랩컷이 잘된 에제이다. 이 같은 경우 인식하고자 하는 사람의 모습 중 색상이 주변 배경의 색상과 겹치는 것이 없고 경계가 분명하기에 잘되었다고 판단이 된다.



다리사이의 배경을 제외한 인덕이의 모습은 아주 잘 되었다. 이도 위 이유와 동일하게 확실한 경계와 색상에 의해 잘됨을 확인할 수 있었다.

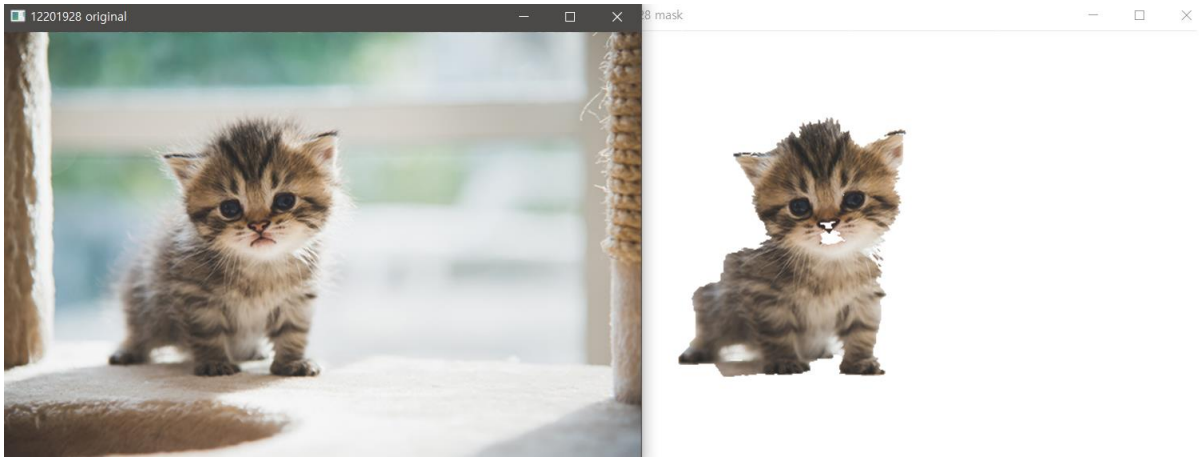


치타의 모습 중 다리 부분이 잘되지않았다. 배경과의 색이 비슷하며 사진에서 약간의 모션블러가 일어났기 때문에 그랩컷이 원할하게 실행되지 않았기 때문이다. 따라서 꼬리부분에서도 약간의 짤림이 확인된다.



바깥의 카페트모습과 같이 컷되었다. 강아지의 털색, 패턴이 외부의 카페트와 매우 유사하기에 동일한 객체로 인식하였기 때문에 이러한 결과가 나왔다.

저해상도의 고양이 사진도 외부의 물체가 동시에 인식되며 몸이 절반잘린 형태를 보인다.



위와 동일하게 그랩컷이 원활하게 이루어지지 않음을 확인할 수 있다. 약간의 털 노이즈로 인해 삐죽삐죽한 모습도 보인다.

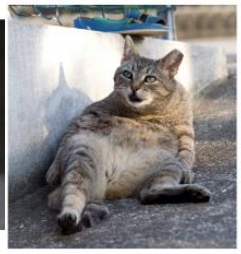
```
//lest 1
//Rect rect = Rect(Point(113,22), Point(509, 608)); //bounding box
//Mat result, bg_model, fg_model;
//Mat img = imread("8/jo.jpg", 1);
Rect rect = Rect(Point(296, 114), Point(471, 360)); //bounding box
Mat result, bg_model, fg_model;
Mat img = imread("8/induk.jpg", 1);
//Rect rect = Rect(Point(126, 92), Point(645, 389)); //bounding box
//Mat result, bg_model, fg_model;
//Mat img = imread("8/cheeta.jpg", 1);
//Rect rect = Rect(Point(23, 41), Point(448, 370)); //bounding box
//Mat result, bg_model, fg_model;
//Mat img = imread("8/hide.jpg", 1);
//Rect rect = Rect(Point(12, 6), Point(346, 463)); //bounding box
//Mat result, bg_model, fg_model;
//Mat img = imread("8/dowm.jpg", 1);
//Rect rect = Rect(Point(80,79), Point(369, 354)); //bounding box
//Mat result, bg_model, fg_model;
//Mat img = imread("8/cat.jpg", 1);
imshow("12201928 original", img);
grabCut(img, result, rect, bg_model, fg_model, 5, GC_INIT_WITH_RECT); //segmentation
compare(result, GC_PR_FGD, result, CMP_EQ);

Mat mask(img.size(), CV_8UC3, cv::Scalar(255, 255, 255));
img.copyTo(mask, result);
imshow("12201928 result", result);
imshow("12201928 mask", mask);
waitKey(0);
destroyAllWindows();
```

이 코드는 위 테스트를 진행한 내용이다.

결론을 내리자면 객체와 배경의 경계가 명확하며 겹치는 색상이 없어야 되며 해상도가 높은사진 일수록 그랩컷이 잘됨을 판단할 수 있다. 잘되지않는 사진을 대체로 분석해보면 명확한 구분이 없거나, 저해상도 이거나, 색이 겹치는 경우기 때문이다. 따라서 이는 컴퓨터비전에 Segmatatio 이 잘되지않는 경우와 매우 비슷함을 확인할 수 있다. 아래는 CS231n 에서 발췌한 내용이다. 이와 같은 경우일때는 잘 되지 않는다.

Challenges: Illumination

Challenges: Deformation



Challenges: Occlusion

Challenges: Background Clutter