

Computer Vision

Assignment 1

precision: 94.319%

recall: 95.142%

황준선

201321140

저는 예제 코드를 기반으로, 성능을 높이기 위한 방법을 다음과 같이 총 2 가지에 초점을 맞춰서 코딩해 나갔습니다. 첫 째는 나뭇잎과 같은 노이즈를 제거하기 위해 (precision 을 높이기 위해) 노력하였으며, 둘 째는 추출해내야하는 물체를 정확히 파악하고, 물체의 안(inside)을 채우는 것(recall 을 높이기 위해)입니다.

노이즈를 제거하기 위해, 총 두개의 필터를 사용하였습니다. 또한, 노이즈 제거 절차는 이미지를 불러올 때와 이미지를 출력 및 저장하기 전에 각 이미지 당 2 번 시행하였습니다. 이미지를 불러올 때 Gaussian Filter(filter size: 7*7)를 사용해서 노이즈를 줄이는 전처리 하였으며, 이미지를 출력 및 저장하기 전에 시행할 때는 Median Filter(filter size: 5*5) + Gaussian Filter(filter size: 7*7) + Binary Threshold(pixel value: 200)를 사용하여 후처리 하였습니다.

추출해내야하는 물체를 정확히 파악하기 위해서 하이퍼 파라미터(threshold, filter size 등)를 조절해가면서 우측 상단 차가 작게 보이는 부분에서부터 잘 추출되도록 노력하였으며, 전처리와 후처리 중간에 물체의 안을 채우기 위해 총 2 가지 알고리즘을 사용하였습니다.

물체를 채우기 위해 사용한 2 개의 알고리즘 중 하나는 flood fill 입니다. flood fill 은 간단하게 말하여 연결 여부를 검사하여 색을 채워나가는 알고리즘입니다. 모두 검은색(pixel 값: 0)인 원본 이미지보다 2pixel 만큼 높이와 너비가 큰 mask 를 만들고, (0, 0)을 기준으로 시작해서 마스크와 색이 겹쳐 연결된 픽셀 값들은 모두 흰색으로 바꿨습니다. 그 후, 해당 이미지를 반전시켜서 배경색을 다시 검은색으로, 물체를 흰색으로 바꾸고, 원본 이미지와 bitwise or 연산을 하여 물체의 안을 채우려고 하였습니다.

하지만, 이렇게 진행해도 자동차 안이 깔끔하게 다 채워지지 않아서 morphology 를 사용하였습니다. morphology 연산은 크게 두 가지로 침식(erosion)연산, 팽창(dilation)연산이 있습니다. 팽창 연산 이 후, 침식 연산을 하면 닫힘(close) 연산이 됩니다. 이는 직관적으로 팽창 연산으로 떨어지거나 비어있는 선이나 공간을 채우고 난 뒤 물체들의 부풀어 오를 부분을 침식 연산으로 제거합니다. 그리고, 침식 연산 이 후, 팽창 연산을 하게 되면 열림(open) 연산이 됩니다. 이는 직관적으로 침식 연산으로 작은 노이즈를 없애고 물체들의 줄어든 부분을 팽창 연산으로 원상복귀 시킵니다. 저의 경우, 물체 안의 빈 공간을 채우기 위하여 닫힘 연산을 사용하였습니다.