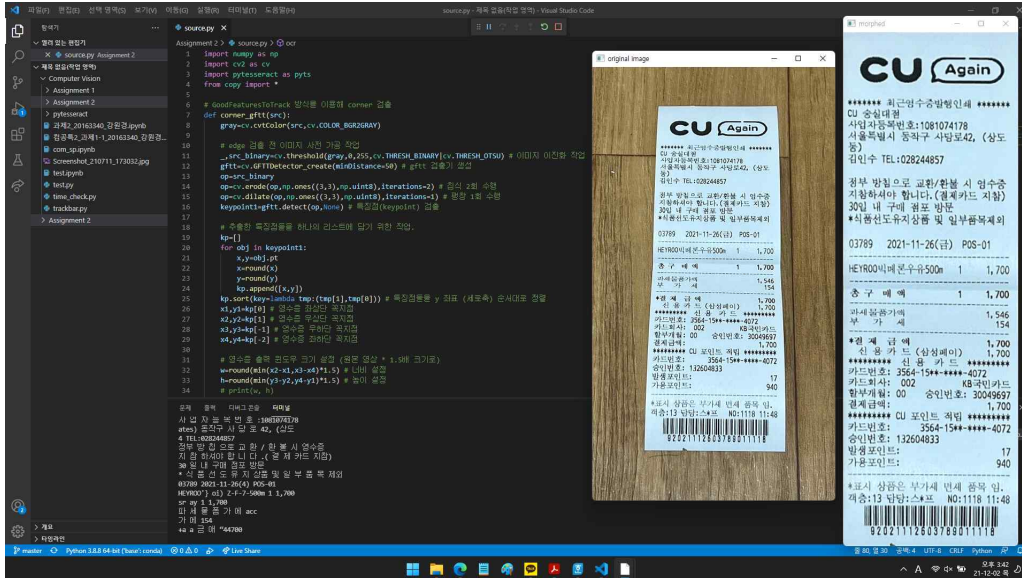


문제 1



#12~16 GoodFeaturesToTrack 방식을 이용해 Corner를 검출합니다.

왜 GoodFeaturesToTrack을 사용했는가?

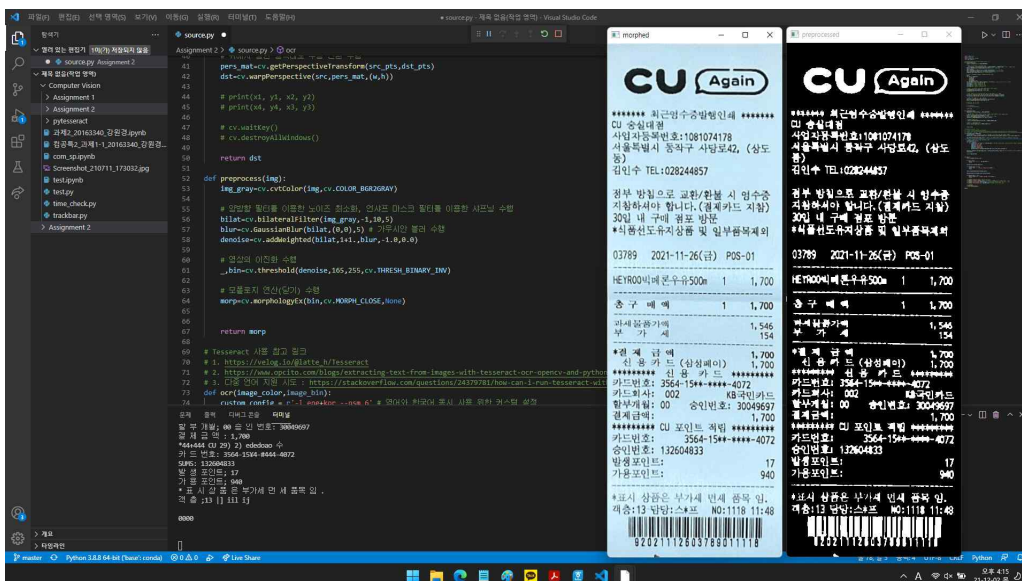
1. 특징점 검출 기법이기는 하나, 내부적으로는 Corner 검출 기법을 사용하므로 목적에 크게 어긋나지 않습니다.
2. 추가적인 비최대 억제 함수 정의 없이 파라미터값 만으로 간편하게 비최대 억제 작업을 수행할 수 있습니다.
3. 이 이미지에서는 위 방식이 허프변환 검출보다 더 적절한 Corner를 검출해주었습니다.

#25 GFTT 방식으로 추출한 특징점들을 y좌표 높이대로 정렬합니다.

#26~29 이 정렬 결과에 따라 상위 2개의 점, 하위 2개의 점을 추출하였습니다.

#41~42 이 4개의 점을 바탕으로 영수증의 영역을 특정하여 getPerspectiveTransform과 warpPerspective를 이용해 장방향으로 출력하였습니다.

문제 2



1번 과정에서 추출한 이미지를 parameter로 받습니다.

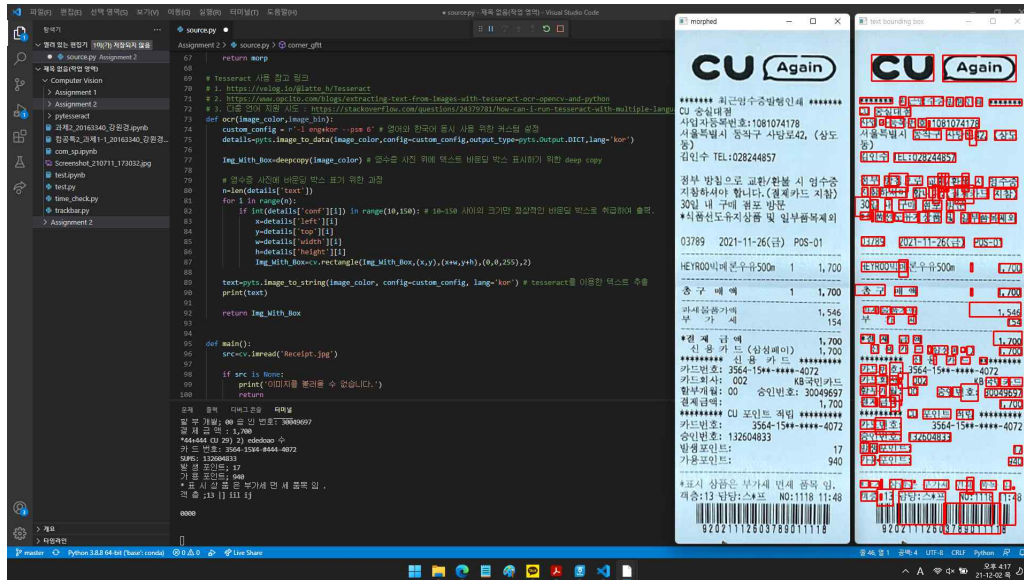
#56~58 양방향 필터를 이용해 1차적으로 노이즈를 제거합니다.

연사프 마스크 필터의 개념을 이용해 영상에 샤프닝을 수행합니다.

#61 이진화는 OTSU에서 출력된 threshold 값(153)과 TRIANGLE에서 출력된 threshold 값(205)을 적절히 혼합한 165라는 값을 하드코딩으로 사용했습니다.

#64 모폴로지 연산(닫기)을 수행해 이진화 후의 노이즈를 최소화 했습니다.

문제 3



이 코드에서는 장방형으로 가공한 Color 이미지를 그대로 입력으로 사용했습니다.

왜 이진화+morphology 연산을 수행한 이미지를 사용하지 않았는가?

1. 이진화+morphology 연산을 수행한 결과가 만족스럽지 않았습니다.

(열기, 닫기, 침식2회후 팽창, 팽창 1회 후 침식 2회 등등 여러 가지를 수행해보았지만, 만족스러운 결과가 전혀 나오지 않았습니다.)

2. 이 원본 이미지를 이용해 이진화+morphology 연산을 수행한 이미지를 입력으로 사용한 것보다 가공되기 전의 장방형 영수증 이미지를 그대로 넣었을 때의 수행 결과가 압도적으로 좋았습니다.

#74 한글+영어+숫자를 모두 출력하기 위한 커스텀 설정 코드입니다.

#80~87 영수증 사진에 바운딩 박스를 표기하기 위한 과정입니다. 이 때, 일정 범위 내의 바운딩 박스만 정상적인 텍스트 바운딩 박스로 처리합니다.

- 이 범위를 주지 않을 경우, 영수증 전체, 또는 이중으로 바운딩 박스가 표기됩니다.

#89 바운딩 박스가 표시된 이미지를 jpg 이미지 파일 형태로 저장합니다.

#91 영수증 이미지로부터 텍스트를 추출합니다.

메인 함수에서 위에 작성한 모든 함수들을 수행하고 각각의 반환된 값을 출력하거나 다음 함수의 parameter로 넘겨주면 정상적으로 동작합니다.

과제 보고서에 더 자세하게 코드에 대한 설명을 적을 수도 있었지만, 공간의 제약 상 간략하게 적었고 더 자세한 내용은 소스코드의 주석으로 첨부하였습니다.