

0. 개요

기본적으로 <https://universe.roboflow.com/nata/my-first-project-2>를 바탕으로 학습한 모델로 건물을 검출합니다. 엠파이어 스테이트 빌딩이 아닌데 검출하는 경우가 굉장히 잦아, 신뢰도가 85% 이상으로 매우 높을 때에만 엠파이어 스테이트 빌딩이 검출된 것으로 처리합니다. 그렇지 않은 경우에는 특징점 검출 및 매칭을 사용하여 엠파이어 스테이트 빌딩을 검출합니다. 특징점 검출은 엠파이어 스테이트 빌딩임에도 불구하고, 검출하지 못하는 경우가 빈번하게 있습니다. 그에 비해, 엠파이어 스테이트 빌딩이 아닌 객체에 대해서는 매우 강인하게 (검출하지 않음) 동작합니다. 그래서, 일차적으로 모델(AI)을 기반으로 엠파이어 스테이트 빌딩 검출을 시도한 후, 그럼에도 엠파이어 스테이트 빌딩이 검출되지 않으면 특징점 검출 및 매칭을 사용하여 검출을 다시 한번 시도합니다.

특징점 검출 방법을 사용한 검출 기법은 다음 장부터 자세히 적어두었습니다.

1. 아이디어

전반적으로는 SIFT 특징점 검출 및 매칭을 사용합니다. 이외 부분에 대해 어떻게 처리하였는지 작성합니다. 먼저, 이미지에 양방향 필터와 가우시안 블러를 적용합니다. 양방향 필터와 가우시안 블러로 인해 웬만한 잡음이 사라집니다. 특히, 가우시안 블러는 해상도가 다른 오브젝트인 경우에서도 매칭이 잘 이루어지도록 하는 결과를 보였습니다. 특징점 사이의 거리는 0.7 미만이어야지 받아들입니다(accept). 이렇게 accept된 특징점들을 matched_points라고 정의합니다.

matched_points의 개수가 4개 미만이라면 투시 변환을 수행하지 못하므로 건물이 없다고 판별합니다. findHomography 함수를 사용해 가장 잘 매칭할 수 있는 변환 행렬 M을 구합니다. 템플릿 영상의 끝 네 점에 투시 변환 행렬 M을 적용하여 결과 영상에서는 어떤 좌표(result)에 매칭되는지 구합니다. 이때, 올바르게 볼록 사각형인지 판단하기 위해 ccw를 사용하여 반시계 방향으로 잘 구성되어 있는지 확인합니다. 올바르게 사각형이 그려지지 않는 매칭이라면 건물이 없다고 판별합니다.

이제, 결과 영상에 그려진 사각형(직사각형은 아닐 수 있음) 내부의 매칭된 특징점들의 개수와 비율을 구합니다. 템플릿 영상은 오직 empire state building만 포함하므로, 결과 영상에서 매칭된 특징점들의 비율은 매우 높아야 합니다. 또한, 매칭된 영역은 충분히 넓어야 하며, 세로 방향으로 긴 직사각형이어야 합니다. 이런 조건을 하나라도 만족하지 않으면 건물이 없다고 판별합니다.

모든 경우를 다 통과하면 건물이 있다고 판별하고, 앞서 구한 result를 바탕으로 사각형을 그립니다.

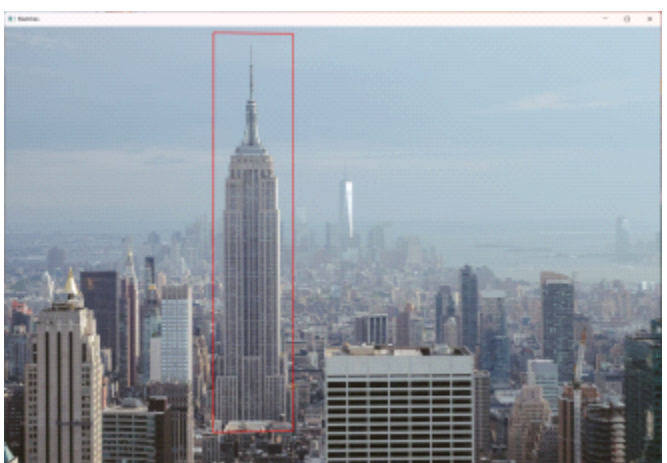
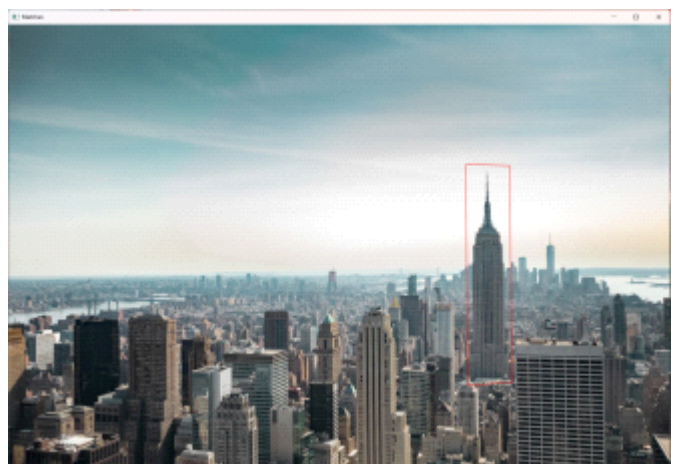
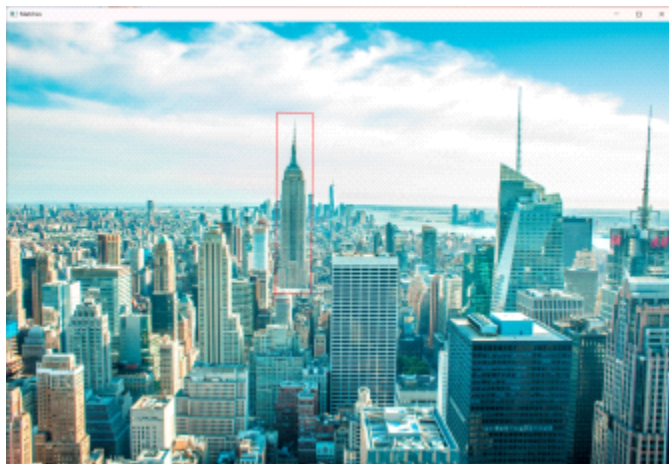
하나의 입력 영상이 들어오면 사전에 구성한 6개의 사진과 비교를 실시합니다. 이때, **건물이 포함되어 있지 않은 데이터인 경우, 모든 템플릿(모델) 영상과의 비교를 수행하기 때문에 실행 시간이 매우 길어질 수 있습니다!! (길어도 1분 내에는 동작합니다.)**

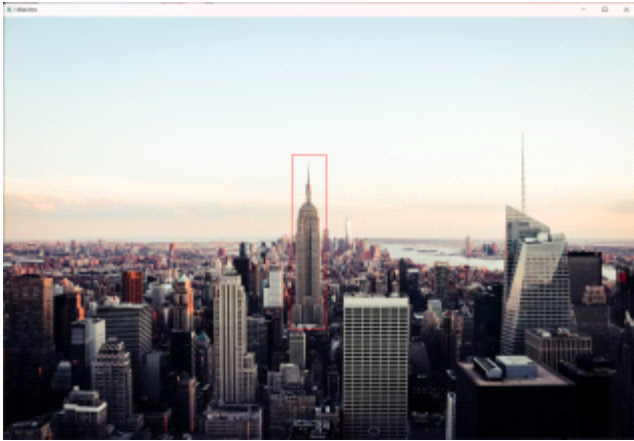
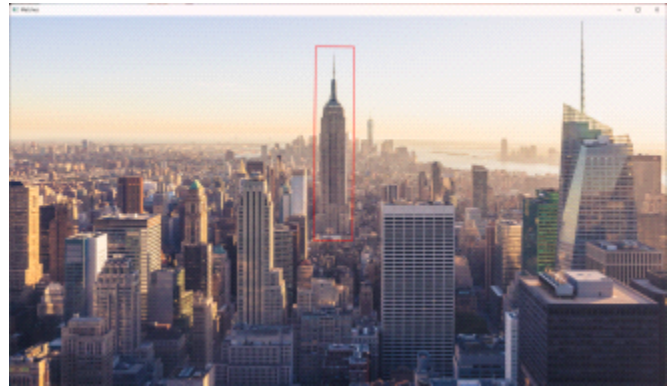
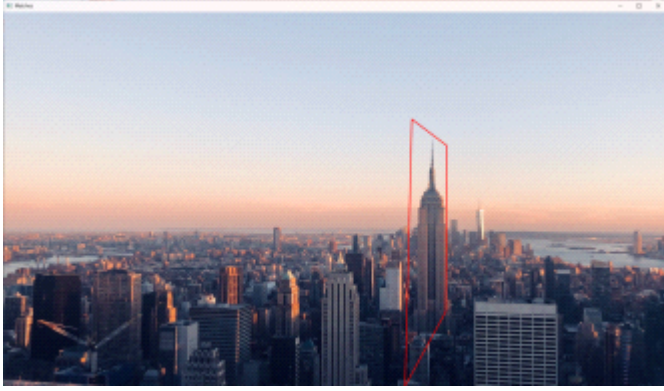
2. 과제 실행 결과

- 건물이 존재하지 않는 경우 (과제 2-1)

```
Run run1 x
C:\Users\Chansol\AppData\Local\Programs\Python\Python312\python.exe C:\Users\Chansol\Desktop\ssu\cv\hw02\run1.py
0
False
1
False
2
False
3
False
4
False
5
False
6
False
Process finished with exit code 0
```

- 건물이 존재하는 경우 (과제 2-1, 2-2)





각 실행 결과는 `python3 run1.py` 혹은 `run2.py`를 실행하면 직접 확인하실 수 있습니다. 테스트 도구와 샘플 데이터는 아래에 있습니다.

3. 테스트 도구 실행 방법

과제 메인 폴더에서,

- 과제 1 : `python3 run1.py`
- 과제 2 : `python3 run2.py`

샘플 데이터 (각 17개씩)

- 건물 포함 데이터 : `tests_yes/`
- 건물 미포함 데이터 : `tests_no/`

템플릿 데이터 (모델 데이터 6개)

- `models/`