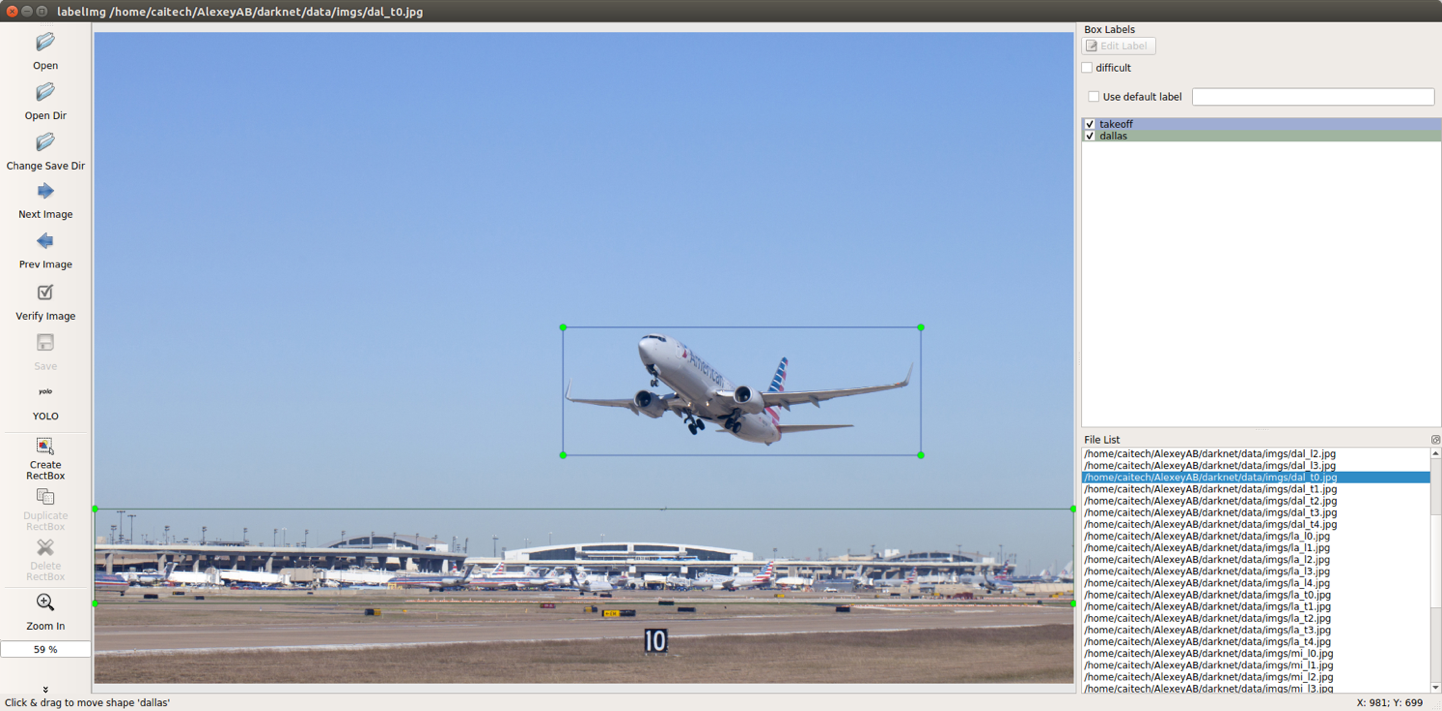
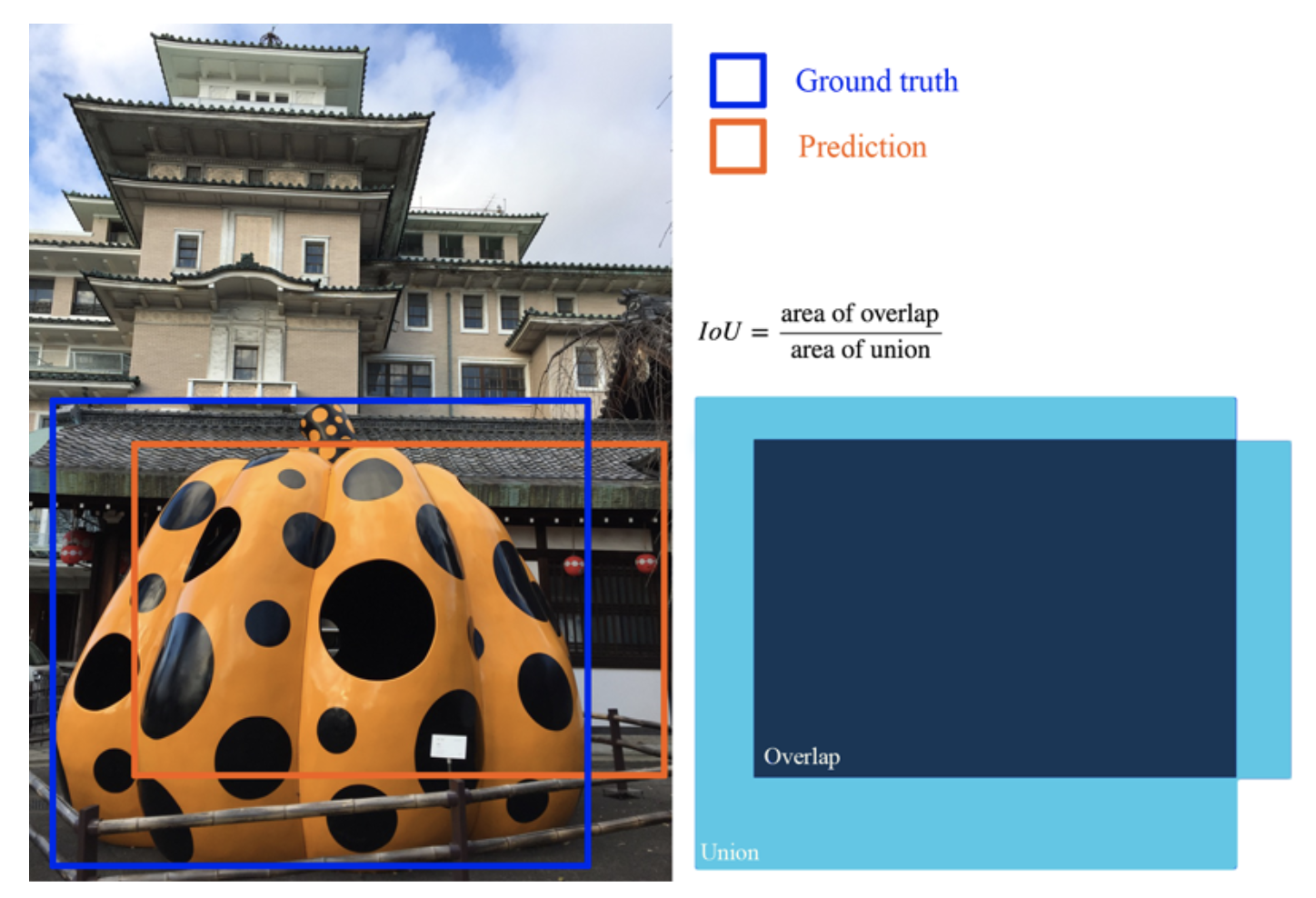
18. 08.06 진행상황

1. 구글 이미지에서 공항 전경과 비행기가 한 프레임에 담겨 있는 이미지 위주로 수집.
   1. 5 x 2 x 4 = 40장 이미지 수집(5개의 공항 x 이/착륙 x 각 비행기 상태에 따른 이미지 4장씩)
      1. dallas/chicago/losangeles/newyork/miami airport
      2. takeoff/land flight
2. 라벨링 도구를 이용하여 각 이미지에 공항의 종류와 비행기 이착륙 상태를 직접 라벨링.  
   
3. yolo\_v3 알고리즘을 이용하여 이미지 학습: epoch(학습 반복 횟수): 23,000번
4. 테스트셋에대한 평가  
   

|  |  |  |
| --- | --- | --- |
| class\_id = 0 | name = dallas | ap = 0.00 % |
| class\_id = 1 | name = chicago | ap = 64.65 % |
| class\_id = 2 | name = losangeles | ap = 9.09 % |
| class\_id = 3 | name = newyork | ap = 0.00 % |
| class\_id = 4 | name = miami | ap = 77.27 % |
| class\_id = 5 | name = takeoff | ap = 78.61 % |
| class\_id = 6 | name = land | ap = 80.62 % |
| for thresh = 0.25, precision = 0.48, recall = 0.47, F1-score = 0.48  for thresh = 0.25, TP = 15, FP = 16, FN = 17, **average IoU = 38.65 %**  **mAP(mean average precision) = 0.443203, or 44.32 %**  Total Detection Time: 1.000000 Seconds | | |

1. 결과 성능 지표  
   IoU와 mAP에 대해서는 Appendix에 설명되어 있음.
2. 결과 해석
   1. 공항의 종류를 판단하는 것은 잘 수행하지 못함
      1. 위의 표를 보면 공항에 대한 ap(average precision)는 각 공항마다 성능이 잘 나오는 것도 있고 아예 탐지 못하는 것도 있음.
      2. 복합적인 원인이 있어 보임. 공항 라벨링의 일관성 부족 + 공항마다의 특징을 나타내는 이미지를 구하기 쉽지 않음.
   2. 그러나 이륙과 착륙 여부를 판별하는 것은 공항 종류를 판별하는 것에 비해서 좋은 결과를 보임.(비행기의 shape가 어느정도 잘 학습 된 것으로 판단 됨.)
3. Appendix.
   1. object detection에서는 성능지표로 주로 **IoU**나 **mAP**라는 지표를 사용한다.
   2. IoU(Intersect of Union)  
      
   3. mAP(mean Average Precision): 각 클래스별 precision을 평균낸 것 0~1사이의 값을 가짐.
4. 결론
   1. 원하는 이미지로 yolo 학습이 가능함을 확인 함.
   2. 저번 면담 때 동영상의 경우에 프레임 마다 탐지된 개체를 텍스트 파일로 저장 가능한지 확인해 보라고 하셨는데, 조사 해본 결과 직접 C++로 작성된 코드를 수정해야해서 현재로선 조금 힘들어 보임. 그러나 이미지의 경우 탐지된 개체(결과 값)를 텍스트 파일로 저장 가능함.