

실생활 투명 객체 3D 데이터 및 대용량 3D 객체 데이터톤 소개



목차

1. Overview

2. Data Information

3. Baseline Code

4. 성능평가지표

1. Overview

- ◆ 대회 주제 : 투명 객체 3D 및 대용량 3D 객체 탐지 AI 모델 개발
- ◆ 목적 : 제공된 투명 객체 3D 데이터(10종) 및 대용량 3D 객체 데이터(10종)를 이용하여 객체 탐지 및 자세 추정 (6D Pose Estimation) AI 모델 개발
- ◆ 대회방식
 1. 참가팀은 제공된 베이스라인 코드와 투명 객체 3D 데이터셋(10종)과 대용량 3D 객체 데이터셋(10종) 데이터셋을 참고하여 3D 객체 탐지 AI 모델을 개발합니다.
 2. 대회 종료 시 개발된 AI 모델과 규정된 양식의 결과 요약지를 이용하여 개발된 모델에 대한 간략한 설명 및 자체 성능 평가 결과를 제출합니다.
 3. 제출한 AI 모델을 주최측에서 테스트셋으로 성능 평가를 실시하고, 순위에 따라 대상(1팀), 최우수상(1팀), 우수상(1팀)을 선정하여 시상이 진행됩니다.
- ◆ 지원사항

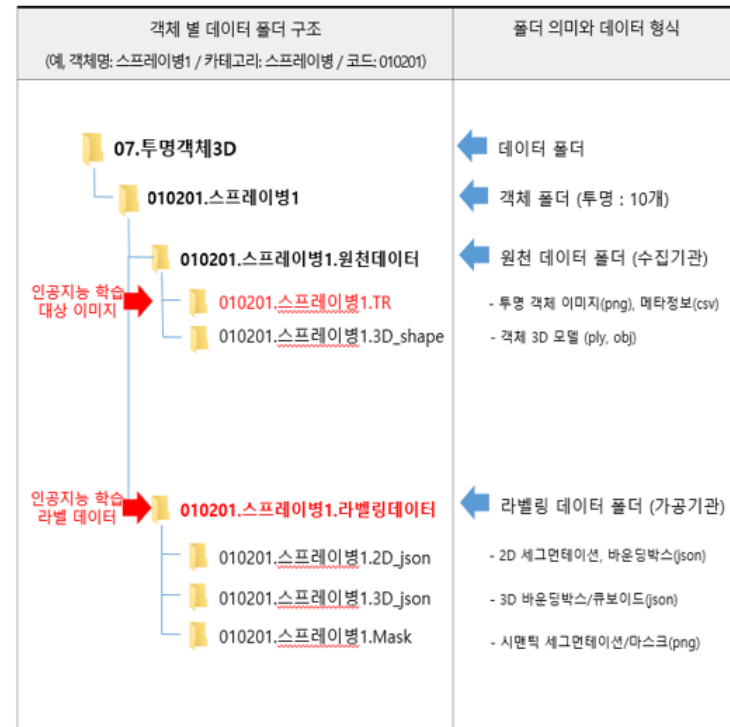
원활한 대회 운영을 위해 네이버 클라우드(nccloud) 환경에서 대회가 진행됩니다.
학습을 위해 각 참가팀별 네이버클라우드 GPU서버가 제공됩니다.
(Nvidia Tesla P40 (2GPU), GPU 메모리 48GB, vCPU 8개, 메모리 60GB, 디스크 100GB SSD)

2. Data Information

데이터는 2가지 타입에 데이터를 제공합니다.

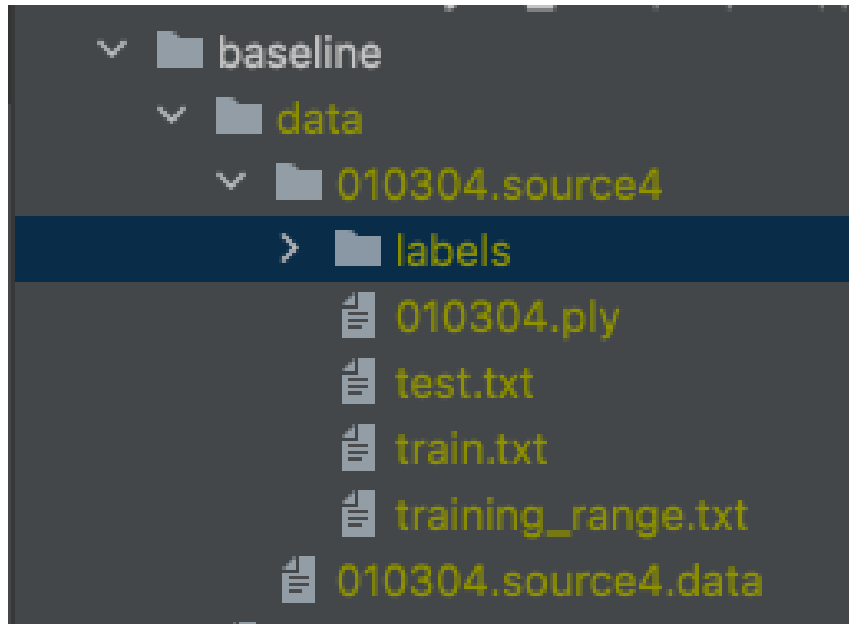
- ◆ 실생활 투명 객체3D 데이터(소분류 10개 종류): 유리/플라스틱 재질. 비정상 형태이거나 액체가 담긴 투명체를 포함합니다.
- ◆ 대용량 3D 객체 데이터(소분류 10개 종류): 다양한 카테고리를 제공 (6페이지 참조)
- ◆ 투명객체3D/대용량3D 데이터셋 구조

1) 폴더 구조



2. Data Information

- ◆ 각 파일에 정의된 속성값에 대한 정보는 [baseline/README.md](#) 내 링크로 제공되는 데이터 정의서 참조
- ◆ 제공되는 학습 데이터셋에 대해 customized 된 구조로 사용 가능하며, baseline에서는 making_txlabels.py를 통해 LINEMOD RGB+D 데이터셋 구조와 유사하게 변환해줍니다.
- ◆ 학습 데이터를 LINEMOD 형태로 변환 한 후 구조



2. Data Information

객체 3D데이터 해커톤 실생활 투명 3D 객체 및 3D 대용량체

3D 데이터 해커톤 제출 객체(투명체 10개 객체, 대용량 10개 객체)

번호	투명객체	갯수	대용량객체	갯수
1	010304.소스용기4	415	030102.쉐이커	1200
2	010305.소스용기5	416	070205.물컵	1200
3	010312.소스용기12	410	070608.각티슈	1200
4	010313.소스용기13	416	070708.휴지통	1200
5	040207.계량컵7	432	100211.무드등	1200
6	040209.계량컵9	416	070710.향초	1200
7	010202.스프레이병2	416	090206.하모니카	1200
8	010204.스프레이병4	416	070702.라이터	1191
9	010321.소스용기21	416	070308.물조리개	1200
10	010201.스프레이병1	416	060201.무우	1200

2. Data Information

객체(투명체, 대용량체) 3D파일형식 설명

파일형식	설명	수집파일	가공파일
csv	수집시 카메라 및 객체 정보	●	
obj, ply	수집 단계 3D Shape 파일	●	
png	Color이미지, Depth 이미지	●	
png(mask)	Binary Mask		●
json(2d)	Bounding, plolygon 라벨링 정보		●
json(3d)	3d 라벨링 정보		●

3. Baseline Code

◆ 객체 3D기반 Microsoft/SingleShotPose 기반으로 작성되었습니다.

- <https://github.com/microsoft/singleshotpose>

◆ 제공하는 학습 데이터셋에 대해 LINEMOD 데이터 형태로 변환됩니다.

◆ Baseline 코드 변경사항

1. making_txtlabels.py 생성 : 제공되는 데이터셋을 LINEMOD 데이터 구조 변환
2. train.py : 멀티 GPU 사용 설정 추가 및 데이터셋에 대한 경로 변경
3. image.py, dataset.py, region_loss.py : 데이터셋에 대한 경로 변경
4. utils.py, valid.py : IoU(Intersection over Union) 관련 연산 함수 추가

4. 성능평가지표

◆ IoU(Intersection of Union)

비교 대상 영역간의 교집합되는 영역의 비율을 나타냅니다. 공식은 아래와 같습니다.



$$\text{IoU} = \frac{\text{Overlapping Region}}{\text{Combined Region}}$$

IoU = 교집합 영역 넓이 / 합집합 영역 넓이

◆ Reprojection Error

포인트 간의 오차를 구하는 것으로 즉 RMS (Root Mean Square) 계산법과 같습니다. 전체 평가를 위해서는 오차 에러를 백분율로 변환

```
eps = 1e-5
acc = len(np.where(np.array(errs_2d) <= px_threshold)[0]) * 100. / (len(errs_2d) + eps)
```

◆ 전체 평가 점수

IoU(%) 값 + reprojection error 백분율 값 = 평가점수

Threshold 범위는 20pixel

감사합니다
