

AI 모델 개발 및 학습 유효성 검증 보고서

- 데이터8. 대용량 객체 3D 데이터 -

인공지능 학습용 데이터 구축	사업 총괄	
	AI모델 개발	
		
학습 유효성 검증 보고서 작성 (대용량 객체)	과제 책임자	김성흠 조교수
	실무 책임자	장재훈 석사과정
	참여 연구원	최서윤, 길다영, 김채현 학부연구생
	연구/개발 보조	박지영 전임연구원
문서 작성/변경 내용	2021년10월05일	AI 모델 개발 및 샘플 검증 현황 v0.1
	2021년11월19일	AI 모델 개발 및 유효성 검증 현황 v1.0
	2021년12월16일	AI 모델 개발 및 유효성 검증 보고 v1.1

□ 검증 대상

- 영역 : 비전 영역
- 과제명 : 1-29 객체 3D 데이터
- 데이터명 : 데이터7. 실생활 투명 객체 3D 데이터
- 레퍼런스 논문 : Real-Time Seamless Single Shot 6D Pose Prediction, 2018, CVPR
(객체 3D 데이터에 대한 AI모델을 충남대학교와 송실대학교에서 공동 개발)

목 차

1. AI 학습 모델 개발	3
1-1 객체 3D 데이터 구축 배경	3
1-2 기술 요구 사항 분석	3
1-3 AI 학습 모델 개발의 목적	3
1-4 객체 3D 인지 기술 동향	4
1-5 AI 학습 데이터셋 비교 및 분석	5
1-6 AI 학습 모델 개발 및 테스트	6
1-7 AI 학습 모델 배포 및 운영	6
2. 학습 데이터의 유효성 검증	7
2-1 데이터 유효성의 객관성 확보	7
2-2 유효성 검증 지표 및 목표 성능 도출	7
2-3 학습 데이터의 수집 범위	8
2-4. 학습 데이터의 가공 정보	8
2-5 객체 자세 추정 및 평가를 통한 AI 모델 검증	9
2-6 3D 모델 투영과 가공된 객체 영역을 비교하여 AI 모델 검증	9
2-7 AI 학습 모델 검증을 통한 학습 데이터의 유효성 확인	10
3. 검증 결과 및 토의 사항	11
3.1 AI 학습 모델/유효성의 품질 요구 사항 분석	11
3.2 유효성 시험 환경	12
3.3 검증 결과 분석	13
4. 참고 문헌	14
5. 부록	15
[별첨 1] 유효성 검증 환경	17
[별첨 2] 모델 학습 및 검증 조건	18
[별첨 3] 유효성 검증 체크리스트	19
[별첨 4] 도커 환경 배포	21
[별첨 5] 서버 환경 정보 (대용량)	23

1. AI 학습 모델 개발

1-1. 객체 3D 데이터 구축 배경

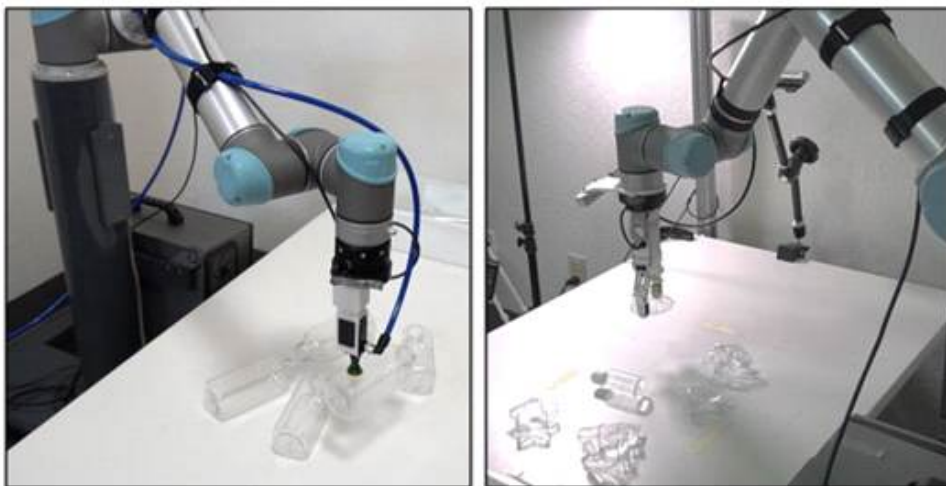
- AI 기술 경쟁력의 핵심인 대규모 학습용 데이터를 단기간에 구축하여 AI 선도 국가로 도약
- 객체 3D 산업의 기술 경쟁력 제고. 일자리 창출과 함께 새로운 성장 동력을 확보하려 함.
- 일상생활에서의 3D 객체를 대용량으로 구축하여, 로봇, 미디어 등의 응용 분야에서 활용

1-2. 기술 요구 사항 분석

- 실감 렌더링 기술, 사람과 객체 상호작용, 로봇 시스템, 게임 기술, VR/AR/MR 분야에 활용
- 원천 데이터로는 객체 3D 이미지를 확보하고, 학습 데이터로는 객체 3D 포인트/메쉬, 세그멘테이션 (객체 영역), 3D 바운딩 박스 등을 가공
- 구축된 학습용 데이터를 활용하여 수행기관 (사업자 등)이 시범 개발해야 하는 모델의 기능과 목표, 성능 등을 작성. 단일 시점에서 객체 3D를 인지하는 딥러닝 모델에 대해 탐색함.

1-3. AI 학습 모델 개발의 목적

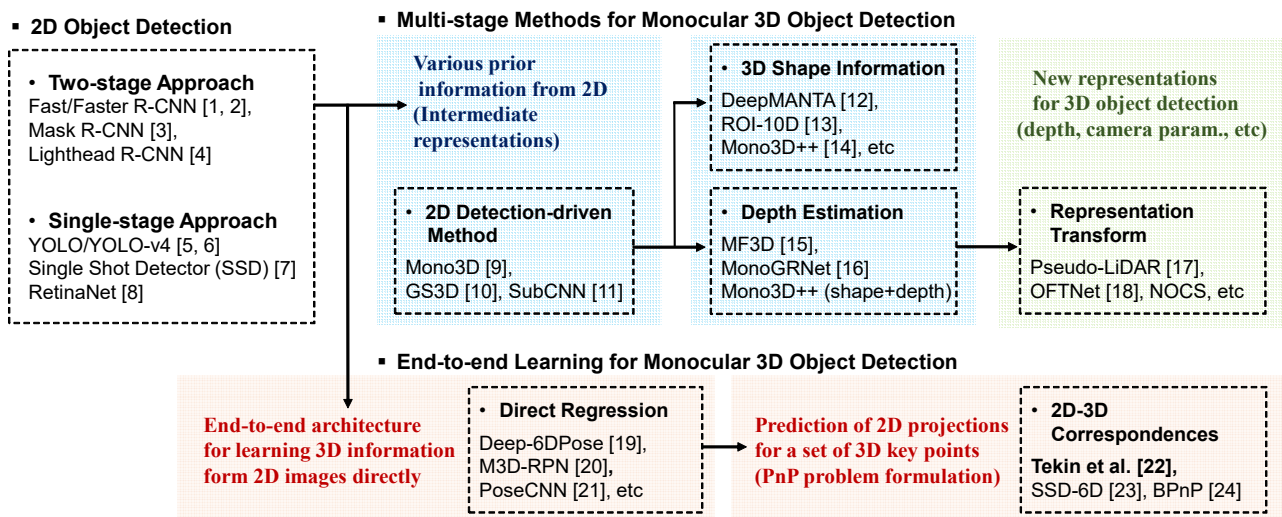
- Pose Estimation이란 카메라에서 보이는 2차원의 물체를 3차원 공간상에서 물체가 어디에 있고 어떤 방향(누워있는지, 앞면을 보이는지, 뒷면을 보이는지 등)으로 놓여 있는지 인식하는 것.
- 주로 로봇 조작(Robotic manipulation)이나 증강현실(Augmented Reality)에서 핵심적인 역할.
- 약 10년 전 Microsoft의 Kinect가 널리 보급되면서 연구가 활발하게 진행. Depth camera를 통해 각 이미지의 pixel 들이 카메라로부터 얼마나 멀리 떨어져 있는지를 알게 되어, 타깃 물체의 3D Model 정보가 있다면 Model의 3차원 포인트들과 현재 카메라로 측정된 포인트들을 계산하여 6D(R,t)를 쉽게 추정하여 비교적 물체를 더욱 정확하게 측정해내는 것이 가능하게 됨.



< 로봇 응용을 위한 객체 3D 위치 및 자세 추정 예시 >

1-4. 객체 3D 인지 기술 동향

- 객체 3D 인지를 단일 시점 기반으로 수행하는 기술은 세계적으로도 큰 관심을 받고 있음.
- 특히, 딥러닝 기반 3D 객체를 검출하는 기술은 2D 검출 분야의 성공에 이어서 활발히 연구/개발되고 있으며, 다양한 데이터셋 공개와 함께 최근 큰 발전이 있었음.
- 2D 검출과 유사하게, 3D 객체 검출 네트워크 또한 객체의 3D 방향과 객체 크기를 여러 단계에 걸쳐 추정하는 two phase, multi-stage approach와 해당 정보를 end-to-end로 direct하게 추론하는 single-stage approach 접근으로 나눌 수 있음.



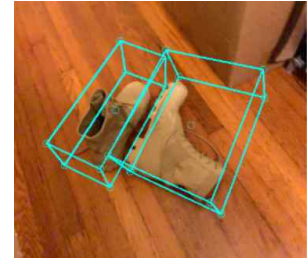
< 단일 시점에서 3D 객체를 검출하는 알고리즘/딥러닝 기술 동향 정리 >

- multi-stage approach에서는 초창기 성공적이었던 2D 객체 검출에 추가적인 prior 정보를 함께 최적화하여 3D 객체의 방향과 크기를 추정하는 방법을 시작됨. 특히, 대상 객체에 대한 3D template 정보를 적극 활용하는 방향으로 발전하고 있음. 다양한 정보를 여러 단계로 최적화하는 것이 특징. two-phase 접근에 대표적인 연구로 학습 기반의 RGB-D 정보로 3D scene을 복원하는 것과 함께 3D 객체 검출을 수행. 또 다른 유용한 표현법은 bird eye's view (이하 BEV)가 있는데, representation transform을 통한 3D 객체 검출 방법론에서는 입력 영상을 BEV로의 변환하는 과정을 집중하고 있으며, 최근 GAN을 통한 변환도 소개됨.
- single stage 접근에서는 객체의 3D 방향과 크기에 대한 GT 정보를 직접적으로 활용하는 regression network를 설계함. 보다 최적화 된 ROI 정보를 위해 반복적인 refinement를 통해 regression 성능을 높이는 방법론도 소개되고 있음. proposal과 refinement 단계에 대해 많은 연구가 이루어지고 있지만, end-to-end 학습이 추가 되는 접근 방법이라 할 수 있음.
- single stage 접근에서 대상 객체의 3D 큐브 지점을 그것이 2D 영상에 투영된 좌표 지점과의 2D-to-3D 대응관계를 regression하고 여기서 추론되는 결과를 통해 객체의 3D 자세/위치, 크기를 계산하는 방법도 비교적 최근에 도입되어, 이를 고도화하는 네트워크들이 계속 소개됨.
- 따라서, 본 과제에서는 2D-3D의 대응 관계를 학습하는 [22]의 연구 결과가 YOLO와 유사한 구조로 재현성/접근성/확장성에 유리하다고 판단. 직관적인 형태이면서 이 분야의 대표성이 있고 범용적으로 활용할 수 있으며 실시간이 가능하여 응용 분야 개척에도 유리하다고 판단.

DB Name	Description	RGB	Depth	Segm.	3D Cube	3D Model	Sim. DB
PASCAL 3D+ [25]	12 rigid categories. ~20,000 images. 3,000 object instances per category.	O	X	X	X	O	X
KITTI 3D [26]	3 categories, 80,256 instances, 14,999 images (outdoor)	O	O	X	O	X	Virtual KITTI2
SUN RGB-D [27]	800 categories, 11,360 instances, 10,355 images	O	O	O	O	X	X
ShapeNet [28]	55 object categories (ShapeNetCore) 51,300 unique 3D models.	X	X	X	X	O	X
Pix3D [29]	9 object categories. 395 3D models. 10,069 image-shape pairs.	O	X	O	X	O	X
ObjectNet3D [30]	100 object classes. 201,888 instances 90,127 images	O	X	O	△	O	O
FAT [31]	21 household objects, 1~10 instances per image. 61,500 simulated images	X	O	O	O	O	O
Mega Depth [32]	Reconstructed 200 3D models (196 locations), 130K valid RGB-D images	O	O	△	X	O	X
LINEMOD [33]	15 texture-less objects with discriminative shapes, sizes, and colors in household environments. 1200 instances per object.	O	X	O	O	O	X
BOP [34]	171 object categories. 302,791 instances in 97,818 real images. 800K simulated images.	O	O	O	O	O	O
Objectron [35]	9 categories, 17,095 inst. (multi-view) 14,819 videos	O	X	X	O	X	X
DexYCB [36]	20 different objects. 582K RGB-D frames over 1,000 sequences	O	O	O	O	X	X



LINEMOD [33]



Objectron [35]

< 객체 3D 학습을 위한 데이터셋 비교 분석. LINEMOD [33]와 Objectron [35] 예시 그림 >

1-5. AI 학습 데이터셋 비교 및 분석

- 본 사업에서 제안하는 학습 데이터셋에 강점 중 하나는 영상에 보이는 객체의 정밀한 3D 모델이 함께 제공하는 것임. 영상에 나타나는 객체의 3D 모델이 주어지는 (image-shape pair) 유사 데이터셋은 PASCAL3D+, Pix3D, ObjectNet3D 등이 있으나, 깊이 정보를 알 수 없음.
- 대표적인 RGB-D 데이터셋인 SUN RGB-D에서는 3D 모델을 가정하지 않음. 자율주행 응용의 KITTI3D, CityScape3D, Synscapes, SYNTHIA-AL 등에서도 가상 영상으로 데이터 양은 많으나, 3D 모델은 주어지지 않음.
- 대표적인 3D 모델 데이터셋인 ShapeNet, 3D Warehouse 등에서는 그러한 3D 모델이 실제로 찍혀 있는 RGB 영상들이 없어, 학습 범위가 한정적임.
- FAT과 같이 사실적 렌더링을 통한 시뮬레이션 영상을 제공하는 방법은 모든 종류 라벨을 계산할 수 있지만, 실사 영상과의 도메인 차이로 가상 DB만으로는 실세계 적용에 한계가 있음.
- MegaDepth에서는 RGB 영상을 SfM+MVS 방법으로 복원하여 깊이 및 3D 모델을 만들어 내지만, 장면 복원 중심. 객체 단위의 위치/자세 추정 학습은 어려움.
- 다양한 데이터셋을 수집하여 정리한 BOP 데이터셋에서는 LM[33], LM-O[37], T-LESS[38], ITODD[39], YCB-V[40], HB[41], RU-APC[42], IC-BIN[43], IC-MI[44] 각각에 따라 일부 라벨이 없는 경우 있음. 자체적으로 수집한 원천데이터는 TUD-L[45], TYO-L[45]이며, 가상 DB를 추가함. 그러나 대부분의 카테고리들이 산업 응용에 초점이 맞추어져, 응용 범위가 대단히 한정적임.
- 비전 분야의 최상위 학회에서 최근 공개된 Objectron, DexYCB 등에서는 비디오 촬영 및 반자동 가공/정제 기법 도입을 통해 학습 DB의 수집 양을 크게 늘림. 세계적으로도 객체 3D를 대용량으로 수집하는 경향을 확인하였으며, 일부 기법은 본 과제에서 활용하는 것과 유사함.
- 특히, LINEMOD[33]와 Objectron[35]을 관심있게 살펴 보았으며, 본 과제의 레퍼런스 모델에서 활용하는 LINEMOD를 참고하고 이를 발전시켜 학계 및 산업계에 도움이 되는 형태로 구축함.

1-6. AI 학습 모델 개발 및 테스트

- 위와 같은 배경에서 본 과제의 레퍼런스 AI 모델을 선정함. Microsoft에서 제공하는 오픈 소스 기반으로 객체 3D 위치/자세를 추정하는 AI 모델을 개발함.
- 기존 공인 데이터셋인 LINEMOD로 테스트하여 AI 모델이 정상적으로 작동하는지 확인함.
- 학계에서 검증된 AI 모델을 누구나 접근/재현 가능하도록 인터페이스를 수정하거나 자동화 함.

* AI 학습 모델 개발

인공지능 학습 모델 개요 (객체 3D 데이터)	
개발 언어	Python 3.6
프레임워크	PyTorch 1.8.0, CUDA 11.1, OpenCV 4.5.3.56, SciPy 1.2.0, Pillow 8.2.2
학습 알고리즘	Real-Time Seamless Single Shot 6D Object Pose Prediction (CVPR18) 설치 및 테스트: https://github.com/seongheum-ssu/nia_singleshotpose
입력 정보	학습/평가 데이터: .jpg (영상), .txt (자세), .png (영역), .ply (3D 데이터)
출력 정보	객체 당 큐브의 꼭지점을 예측. 이를 통해 3D 위치/자세 값을 계산.
테스트 방법	공인 LINEMOD 데이터셋: https://paperswithcode.com/dataset/linemod-1

- 핵심이 되는 딥러닝 네트워크는 CVPR18에서 공개한 버전 그대로이며 임의 편집하지 않음.
- 해당 모델과 데이터셋은 객체 3D 분야에서 매우 유명한 사례이며, 잘 이해되고 있음.
- 개발한 코드와 데이터셋은 Github 페이지를 통해 주관기관 및 관련 참여기관과 공유함 (7/22).
- 한편, 객체의 3D 위치/자세를 추정하는 것은 로봇과 미디어 응용에 핵심이 되는 기술임. 3D 공간 상의 객체의 위치와 자세를 이해하는 것은 로봇 gripping이나 장애물 회피를 위한 path planning에 반드시 필요한 요소이며, 가상 객체를 3D 공간 상에 등록하거나 사실적 렌더링 콘텐츠를 제작할 때도 중요함.
- 일상 생활 객체의 3D 위치/자세를 추정하는 것은 구축된 3D 모델과 함께 주어진 현실 공간을 가상 객체들로 재구성할 수 있게 함. 재구성 된 공간에서는 객체 위치/자세가 변경되며 증강 학습을 위한 데이터가 무제한으로 생성될 수 있음. 본 과제에서 개발된 모델과 결과물들은 사실적 렌더링 기법과 함께 향후 중요한 self-training 기법으로도 활용될 수 있음.
- 구축된 객체 3D 데이터를 활용하는 딥러닝 모델들을 AI 전문 기업들과 공유하여, 가치 있는 응용 서비스를 발굴하고 새로운 BM이 도출될 수 있기를 기대함.

1-7. AI 학습 모델 배포 및 운영

- AI 모델을 돌리기 위한 실험 환경, 코드 등의 자료는 부록으로 정리함 [별첨 1].
- Github를 통해 코드 및 샘플 데이터셋을 테스트할 수 있으며, 도커 환경을 제공함.
- 카테고리별로 학습된 모델 파일 및 유효성 검증 관련 정보들을 업데이트할 예정.

2. 학습 데이터의 유효성 검증

2-1. 데이터 유효성의 객관성 확보

- 제안되는 신규 데이터셋이 딥러닝 학습 데이터로 유효한지 여부에 대해 객관성을 확보함.
- 공인된 레퍼런스 모델이 공인된 데이터셋에서 보이는 성능을 근거하여 목표를 설정하고, 이 목표가 달성되는 신규 데이터셋은 baseline과 대등한 수준으로 구축되었음을 공인할 수 있음.
- 구축되는 데이터셋으로 객체의 3D 위치/자세가 얼마나 잘 추정될 수 있는지 알 수 있는 성능 지표를 집중적으로 검증함. ground truth로 제시한 라벨링 값이 실제로도 유효한지를 확인함.

* 목표 성능 도출 근거 1. 학계에 보고된 성능 [22]

method	ape	benchvise	cam	can	cat	driller	duck	eggbox	glue	lamp	average
rep. acc.	92.10	95.06	93.24	97.44	97.41	79.41	94.65	90.33	96.53	76.87	91.30%
IoU acc.	99.81	99.9	100	99.81	99.9	100	100	99.91	99.81	100	99.91%
rank	#7	#4	#6	#1	#2	#9	#5	#8	#3	#10	

* 목표 성능 도출 근거 2. 공개된 코드로 재현한 성능 (<https://github.com/microsoft/singleshotpose>)

method	ape	benchvise	cam	can	cat	driller	duck	eggbox	glue	lamp	average
rep. acc.	87.43	92.44	86.18	90.78	91.04	69.97	89.58	86.29	89.00	72.84	85.36%
IoU acc.	98.29	99.9	100	100	99.9	99.9	97.84	99.44	99.81	100	99.51%
rank	#6	#1	#8	#3	#2	#10	#5	#7	#4	#9	

2-2. 유효성 검증 지표 및 목표 성능 도출

- 본 과제에서 선정한 레퍼런스 모델은 LINEMOD 데이터셋에 대해 표 2와 같이 성능이 보고됨. 논문에서 제시한 Table 1, Table 4의 reprojection 기반, IoU 기반 정확도에 대해 정리함.
- 저자가 공개한 코드를 그대로 활용하여 가려짐이 크지 않은 10개의 객체에 대해 학습 및 검증을 수행하고 표 3과 같이 정리함. 일반적으로 논문의 일부 implementation details은 공개되지 않아 100% 재현은 어렵지만, 유사한 경향의 결과는 재현할 수 있음. 이를 목표 성능으로 설정.
- 모든 test case의 batch size는 8로 고정. 수렴되는 epoch는 데이터 특성에 따라 다를 수 있음. 그 밖에 파라미터는 공개된 코드의 초기값을 그대로 사용 (e.g. learning rate: 0.001).

2-3. 학습 데이터의 수집 범위

- 대용량 객체 3D 데이터는 10개의 대분류에 대한 32개의 중분류 카테고리 구성되어 있으며, 300개 이상의 객체 인스턴스를 수집함. 사람 외 31종에 대한 세부 정보는 아래와 같음.
- 실생활의 다양한 객체 3D 데이터, 2D/3D 영역과 큐브 정보를 가공하여 3D 공간상에서 객체의

위치와 자세를 추정하는 AI 모델을 학습할 수 있음.

- 객체의 3D 위치 및 자세 추정을 활용하는 로봇, 미디어 응용 시나리오에서 수집된 데이터가 유용하게 사용될 수 있을 것으로 기대함.

*** 대용량 객체 3D 데이터 수집**

대분류명	중분류명	카테고리 ID	인스턴스명	카테고리별 분포
사람	남자	0101XX	20대 외 4건	1.62%
	여자	0102XX	20대 외 4건	1.62%
디지털/가전	생활가전	0301XX	로봇청소기 외 5건	1.94%
	컴퓨터/주변기기	0302XX	게임컨트롤러 외 5건	1.94%
	디지털기기/용품	0303XX	마이크 외 8건	2.91%
패션잡화	신발	0403XX	로퍼 외 10건	3.56%
	모자	0404XX	중절모 외 5건	1.94%
스포츠/레저	스포츠용품	0501XX	럭비공 외 11건	3.88%
	레저/캠핑	0502XX	부탄가스 외 11건	3.88%
	홈트레이닝	0503XX	벨런스보드 외 11건	3.88%
식품	과일	0601XX	석류 외 10건	3.56%
	야채	0602XX	무 외 10건	3.56%
	제과/제빵	0603XX	마카롱 외 8건	2.91%
일상용품	영유아용품	0701XX	장난감자동차 외 11건	3.88%
	주방용품	0702XX	식칼 외 11건	3.88%
	청소용품	0703XX	분무기 외 9건	3.24%
	미용	0704XX	헤어드라이어 외 14건	4.85%
	공구	0705XX	전선탄피기 외 10건	3.56%
	위생용품	0706XX	일반면도기 외 12건	4.21%
	생활잡화	0707XX	옷걸이 외 11건	3.88%
	애완용품	0708XX	애완용밥그릇 외 9건	3.24%
	자동차용품	0709XX	소화기 외 10건	3.56%
	세탁용품	0710XX	가루세제 외 7건	2.59%
		0711XX		
가공식품	즉석/편의식품	0801XX	컵라면 외 8건	2.91%
교육/문화용품	문구/사무용품	0901XX	다이어리 외 6건	2.27%
	악기	0902XX	리코더 외 10건	3.56%
	미술용품	0903XX	팔레트 외 9건	3.24%
	게임용품	0904XX	악어이빨보드게임 외 6건	2.27%
	이벤트용품	0905XX	LED장식초 외 6건	2.27%
가구/인테리어	가구	1001XX	접이식상 외 5건	1.94%
	인테리어	1002XX	사각형벽시계 외 9건	3.24%
의약품/의료기기	의료기기	1101XX	바르는모기약용기 외 12건	4.21%

2-4. 학습 데이터의 가공 정보

- 객체 3D 데이터로는 인스턴스별로 watertight한 mesh 모델과 texture를 복원하고 정제. 인스턴스별로 3D 데이터가 기준이 되는 위치와 자세로 정렬되어 있어야 함. 정렬된 3D 객체는 공간상에서 min/max 2점을 통해 3D 큐브가 자동으로 정의됨
- 관심 객체가 RGB 및 깊이 영상에 투영될 때, 해당하는 전경 영역을 그렇지 않은 배경과 분리.
- 객체는 모두 rigid한 것으로 가정하고, 3D 큐브를 통해 공간상의 위치와 자세를 정의할 수 있음. 영상에 맺힌 관심 객체는 이동 및 회전 가능한 가상 3D 큐브를 통해 위치/자세를 좌표로 기록. 큐브의 각 꼭지점 8개에 대한 2D 좌표는 공간 상에서 자동으로 계산된 3D 좌표와 대응 관계.

* 객체 3D 데이터의 가공 정보 (경진대회용 데이터 예시: 070308)

복원/정제 및 정렬된 3D 데이터	영상에서 큐브 가공 (위치/자세)	관심 객체 영역 분리 (배경증강)

2-5. 객체 자세 추정 및 평가를 통한 AI 모델 검증

- 로봇 응용 분야에서는 공간상의 객체가 어디에, 어떤 자세로 놓여 있는지 예측하는 것이 중요.
- 정렬된 3D 모델과 보정된 카메라 영상의 큐브가 가공되면 2D-3D 대응 관계를 활용하여, 공간상의 위치/자세가 유일하게 결정됨. 위와 같이 가공된 큐브는 기준이 되는 3D 모델이 roll, pitch, yaw 각각 121.7933, -58.4431, 30.2407 deg. 회전하고, 0.83, -0.74, 6.57 이동된 것으로 해석됨.
- 이와 같이, 객체 3D 데이터의 바운딩 큐브의 8개의 꼭지점과 중심점이 영상에 투영되는 2D 픽셀 좌표셋 $p_{gt, k=1 \dots 9}^{2D}$ 에 대해 라벨링되면 해당 객체의 6자유도 위치/자세 정보인 R_{gt}, t_{gt} 가 계산됨. 학습 후 예측되는 $p_{pr, k=1 \dots 9}^{2D}$ 로부터 추정되는 R_{pr}, t_{pr} 의 정확도 유효성은 $p_{gt, k=1 \dots 9}^{2D}(R_{gt}, t_{gt})$ 와 $p_{pr, k=1 \dots 9}^{2D}(R_{pr}, t_{pr})$ 의 L2 norm 평균이 10 pixels 안에 들어오는지 여부로 판단 [22].

* 객체 3D 데이터의 AI 모델 검증 (큐브 꼭지점에 대한 reprojection error 분석으로 정확도 측정)









객체 자세 추정 예시	가려짐으로 인한 성능 저하	가공 데이터의 유효성 판단

2-6. 3D 모델 투영과 가공된 객체 영역을 비교하여 AI 모델 검증

- VR/AR/MR 응용 분야에서는 가상 3D 객체를 공간 상에 투영하는 것으로 실감 콘텐츠를 생성.
- 복원/정제된 3D 모델을 예측한 회전/이동 정보로 영상에 투영. 예측 및 투영된 영역과 사람이 실제 가공한 객체 영역을 비교하여, 겹쳐지는 영역이 일정치 이상이 된다면 유효한 것으로 판단.

- 즉, 테스트 영상에서 학습한 모델을 통해 추정된 R_{pr}, t_{pr} 로 객체 3D 모델을 투영한 예측 영역과 라벨링 된 영역이 잘 일치하는지 여부를 Intersection of Union (IoU: $\frac{A_{gt} \cap A_{pr}}{A_{gt} \cup A_{pr}}$)로 확인. 논문에서는 예측 영역과 라벨링 영역의 교집합과 합집합 비율이 50% 이상이 되어야 참으로 판단 [22].

*** 객체 3D 데이터의 AI 모델 검증 (투영된 3D 모델 영역과 GT 영역의 IoU 분석으로 정확도 검증)**

 <p>81.63 > 0.5 (True)</p>		
IoU 측정 기반 판단	추정된 위치/자세로 모델 투영 (예측)	배경에서 분리된 객체 영역 (가공)
 <p>79.79 > 0.5 (True)</p>		
IoU 측정 기반 판단	추정된 위치/자세로 모델 투영 (예측)	배경에서 분리된 객체 영역 (가공)
 <p>70.46 > 0.5 (True)</p>		
IoU 측정 기반 판단	추정된 위치/자세로 모델 투영 (예측)	배경에서 분리된 객체 영역 (가공)

2-7. AI 학습 모델 검증을 통한 학습 데이터의 유효성 확인

- 객체 3D 데이터의 위치 및 자세 추정 정확도는 논문에서 제시한 두 가지 방법으로 확인 [22].
- 큐브 꼭지점에 대한 reprojection error 분석으로 가공된 큐브의 품질을 검증 가능.
- 투영된 큐브 영역과 객체 영역에 대한 IoU 분석으로 가공된 객체 3D 데이터 및 영역 검증 가능.

3. 검증 결과 및 토의 사항

3-1. AI 학습 모델/유효성의 품질 요구 사항 분석

- 이미지 분석 분야에 최적화된 3D 객체 위치/자세 추정 알고리즘을 적용하여, 구축된 학습 데이터셋을 활용할 수 있는 베이스라인 AI 모델을 제공.
- 3D 객체 자세 추정 정확도를 reprojection error 기반으로 분석. 카테고리별 정확도(Accuracy) 평균 수치는 82.5%로 설정
- 3D 객체 위치 추정 정확도를 IoU (Intersection of Union) 기반으로 분석. 카테고리별 정확도(Accuracy) 평균 수치는 87.5%로 설정

* 유효성 검증 항목

품질특성	항목명	측정 지표	정량 목표	지표 및 목표 설정 근거
유효성	객체 3D 자세 학습 모델	reprojection error 기반 정확도 (%)	카테고리별 정확도 평균 수치 82.5% 이상	Real-Time Seamless Single Shot 6D Pose Prediction (CVPR18) [22]
	객체 3D 위치 학습 모델	IoU 분석 기반 정확도 (%)	카테고리별 정확도 평균 수치 87.5% 이상	Real-Time Seamless Single Shot 6D Pose Prediction (CVPR18) [22]

※ 작성된 품질 요구사항 정의서는 품질검증 전문기관(TTA 등)과 협의를 거쳐 조정될 수 있음

* 객체 3D 자세 추정 정확도 성능

- reprojection error 기반 분석: $\frac{1}{N} \sum_{k=1}^N \|p_{gt,k}^{2D}(R_{gt}, t_{gt}) - p_{pr,k}^{2D}(R_{pr}, t_{gt})\|_2$
- R: 정렬된 3D 모델과의 상대적 회전
- t: 정렬된 3D 모델과의 상대적 위치 이동
- p: 재투영된 큐브의 각 꼭지점 좌표 (gt: 가공된 큐브, pr: 예측한 큐브)

* 객체 3D 위치 정합 정확도 성능

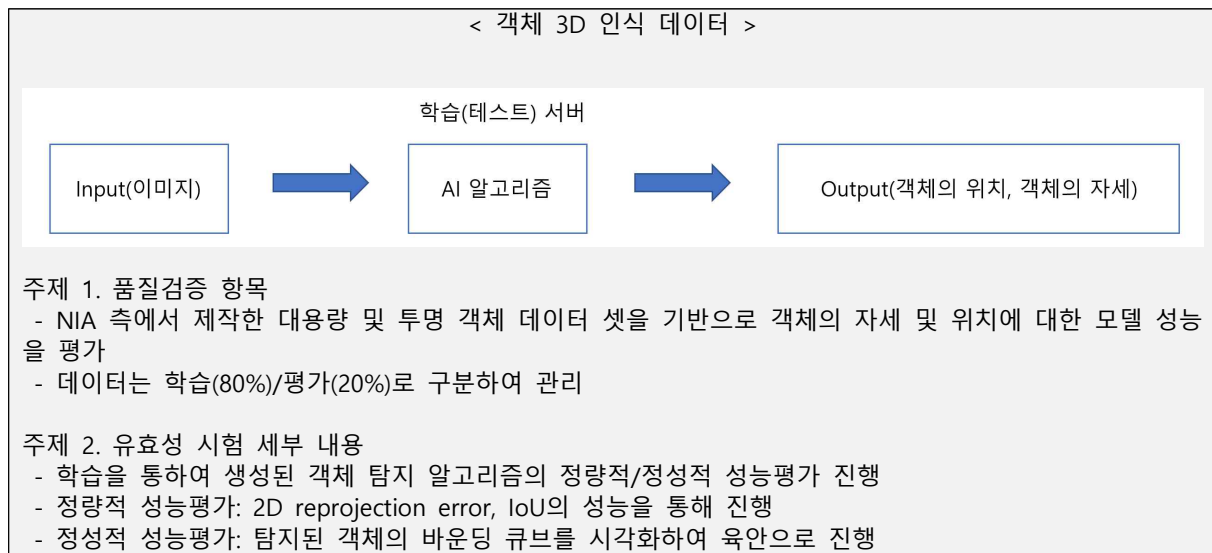
- intersection of union (IoU) score 기반 분석: $\frac{A_{gt} \cap A_{pr}}{A_{gt} \cup A_{pr}}$
- A: 투영된 영역 (gt: 가공된 영역, pr: 예측한 큐브)
- 3D 데이터의 품질이 떨어지거나, 객체 영역이 부정확한 경우 성능이 저하됨



3-2. 유효성 시험 환경

- 아래 표처럼 유효성 시험 환경을 구성하였으며, 모델 정보 및 실행 명령어를 정리함.

* 유효성 시험 환경 구성도



* 유효성 검증을 위한 시스템 환경

CPU	Intel i9-10700K / 10900K
GPU	NVIDIA GeForce RTX 3080 / 3090 GPU
Memory	Samsung DDR4-3200 32GB 이상
Storage	SSD 500GB 이상
OS	Windows 10 / Ubuntu 18.04

* 유효성 검증을 위한 모델 (single-shot pose) 학습 정보

AI 학습 모델	Real-Time Seamless Single Shot 6D Pose Prediction [22]
입력 해상도	1280X720
최대 epoch	5000
학습 조기 종료 구간	1000
배치 크기(batch_size)	8
학습(training) 이미지 수 (80%)	960
평가(test) 이미지 수 (20%)	240

* 성능 평가를 위한 명령어

```

# 입력: 학습/평가 자료 정보 (가공 데이터), AI 학습 모델 정보
# 출력: 학습된 모델 또는 예측한 위치/자세 추정 결과

# 학습
python train.py --datacfg data/[카테고리ID+XX].data --modelcfg cfg/yolo-pose.cfg --initweightfile
cfg/darknet19_448.conv.23 --pretrain_num_epochs 15

# 평가
python valid.py --datacfg data/[카테고리ID+XX].data --modelcfg cfg/yolo-pose.cfg --weightfile
data/[카테고리ID+XX]/model.weights
  
```

* 실행 결과 및 성능 평가 예시 (시각화)

 <p>AI 학습 모델 결과 가공된 큐브 (GT)</p>	 <p>AI 학습 모델 결과 (투영) 가공된 객체 3D 데이터</p>	 <p>AI 학습 모델 결과 가공된 영역 (GT) IoU 분석 (교집합)</p>
<p>큐브 추정 및 가공된 큐브와 비교 (reprojection err. 기반 정확도)</p>	<p>3D 모델을 2D 영상으로 투영 (추정된 큐브의 R, t 정보 활용)</p>	<p>투영된 영역과 가공된 영역 비교 (IoU score 기반 정확도)</p>

3-3. 검증 결과 분석

- 실행 결과 예시. 추정치(파란색)과 정답(초록색) 사이의 차이를 정성적/정량적으로 확인 가능.
- 카테고리별 AI 학습 모델 성능 분석 결과, 설정된 목표를 초과 달성. 평균 성능 뿐 아니라 카테고리별로도 기준치를 크게 미달된 건들은 추가적으로 성능 저하 요인을 분석 중. 현재 가공 및 분석이 진행 중인 사람 외에 카테고리 내 객체 3D 데이터들도 유효성 검증 실시 예정.
- 큐브 라벨 시에 3D 모델 바운딩 큐브의 각 꼭지점의 순서대로 좌표가 지정되어야 함. 3D 모델 정렬 방향에 대해 라벨러가 인지하고 큐브 꼭지점이 순서대로 정의되어야 함. 이 부분이 흐트러지면 회전/자세 해석에 크게 영향을 줌. 일관된 정면 방향 유지가 가장 중요함.

* 학습 유효성 검증 현황 (대용량)

카테고리별 정확도 (%) 검증			
카테고리명	reproj. 기반 정확도 (%)	IoU 기반 정확도 (%)	비고
레저/캠핑 (0502)	99.79%	100%	
홈트레이닝 (0503)	96.49%	100%	
과일 (0601)	100%	100%	
야채 (0602)	93.33%	94.17%	
주방용품 (0702)	89.17%	100%	
청소용품 (0703)	75.83%	28.33%	기준치 크게 미달된 건 분석
미용 (0704)	96.40%	97.47%	
위생용품 (0706)	91.80%	98.18%	
생활잡화 (0707)	94.48%	99.16%	
자동차용품 (0709)	81.93%	78.57%	
악기 (0902)	99.17%	100%	
인테리어 (1002)	78.33%	99.17	
카테고리별 평균(avg.)	89.87%	90.90%	사람 외 가공/분석 진행중
AI 모델 학습이 가능한 test case 28건 (2021.12.17.까지 전달된 목록)에 대한 결과 정리: https://docs.google.com/spreadsheets/d/1hBpDfIhX-XMNNJh3YIAXCMkUz_eo2dUqgDcSYLNc3c/edit#gid=1043349205			

4. 참고 문헌

- [1] Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448
- [2] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* 2016, 39, 1137–1149.
- [3] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
- [4] Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-head R-CNN: In Defense of Two-stage Object Detector. *arXiv* 2017, arXiv:1711.07264.
- [5] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- [6] Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal Speed and Accuracy of Object Detection. *arXiv* 2020, arXiv:2004.10934.
- [7] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
- [8] Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988
- [9] Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. Monocular 3D Object Detection for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2147–2156.
- [10] Li, B.; Ouyang, W.; Sheng, L.; Zeng, X.; Wang, X. GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1019–1028.
- [11] Xiang, Y.; Choi, W.; Lin, Y.; Savarese, S. Subcategory-aware Convolutional Neural Networks for Object Proposals and Detection. In Proceedings of the Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 924–933.
- [12] Chabot, F.; Chaouch, M.; Rabarisoa, J.; Teuliere, C.; Chateau, T. Deep MANTA: A Coarse-to-fine Many-task Network for Joint 2D and 3D Vehicle Analysis from Monocular Image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2040–2049.
- [13] Manhardt, F.; Kehl, W.; Gaidon, A. ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2069–2078.
- [14] He, T.; Soatto, S. Mono3D++: Monocular 3D Vehicle Detection with Two-scale 3D Hypotheses and Task Priors. In Proceedings of the AAAI, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8409–8416
- [15] Xu, B.; Chen, Z. Multi-level Fusion based 3D Object Detection from Monocular Images. In

- Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 2345–2353.
- [16] Qin, Z.; Wang, J.; Lu, Y. MonoGRNet: A Geometric Reasoning Network for Monocular 3D Object Localization. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8851–8858.
 - [17] Wang, Y.; Chao, W.L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 8445–8453.
 - [18] Roddick, T.; Kendall, A.; Cipolla, R. Orthographic Feature Transform for Monocular 3D Object Detection. In Proceedings of the British Machine Vision Conference (BMVC), Cardiff, UK, 9–12 September 2019, pp. 1–13.
 - [19] Do, T.T.; Cai, M.; Pham, T.; Reid, I. Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image. arXiv 2018, arXiv:1802.10367.
 - [20] Brazil, G.; Liu, X. M3D-RPN: Monocular 3D Region Proposal Network for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9287–9296.
 - [21] Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018; pp. 1–10, doi:10.15607/RSS.2018.XIV.019.
 - [22] Tekin, B.; Sinha, S.N.; Fua, P. Real-time Seamless Single Shot 6D Object Pose Prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 292–301.
 - [23] Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. SSD-6D: Making RGB-based 3D Detection and 6D Pose Estimation Great Again. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1521–1529.
 - [24] Chen, B.; Parra, A.; Cao, J.; Li, N.; Chin, T.J. End-to-end Learnable Geometric Vision by Back-propagating PnP Optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8100–8109.
 - [25] Y. Xiang et al., "Beyond Pascal: A Benchmark for 3D Object Detection in the Wild," WACV 2014
 - [26] A. Geiger et al., "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," CVPR 2012
 - [27] S. Song et al., "SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite," CVPR 2015
 - [28] A. X. Chang et al., "ShapeNet: An Information-rich 3D Model Repository," arXiv 2015, arXiv:1512.03012.
 - [29] X. Sun et al., "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling," CVPR 2018
 - [30] Y. Xiang et al., "ObjectNet3D: A Large scale Database for 3D Object Recognition," ECCV 2016
 - [31] J. Tremblay et al., "Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation," CVPRW 2018
 - [32] Z. Li. Et al., "MegaDepth: Learning Single-View Depth Prediction from Internet Photos," CVPR 2018
 - [33] S. Hinterstoisser et al., "Model based Training, Detection and Pose Estimation of Texture-less 3D Objects in Heavily Cluttered Scenes," ACCV 2012
 - [34] T. Hodan et al., BOP Challenge 2020 on 6D Object Localization," ECCV 2020

- [35] A. Ahmadyan et al., "Objectron: A Large Scale Dataset of Object-Centric Videos in the Wild with Pose Annotations," CVPR 2021
- [36] Y.-W. Chao et al., "DexYCB: A Benchmark for Capturing Hand Grasping of Objects," CVPR 2021
- [37] E. Brachmann et al., "Learning 6D Object Pose Estimation using 3D Object Coordinates," ECCV 2014
- [38] T. Hodan et al., "T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects," WACV 2017
- [39] B. Drost et al., "Introducing MVTec ITODD-A Dataset for 3D Object Recognition in Industry," ICCVW 2017
- [40] Y. Xiang et al., "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," RSS 2018
- [41] R. Kaskman et al., "HomebrewedDB: RGB-D Dataset for 6D Pose Estimation of 3D Objects," ICCVW 2019
- [42] C. Rennie et al., "A Dataset for Improved RGBD-based Object Detection and Pose Estimation for Warehouse Pick-and-place," RAL 2016
- [43] A. Doumanoglou et al., "Recovering 6D Object Pose and Predicting Next-best-view in the Crowd," CVPR 2016
- [44] A. Tejani et al., "Latent-class Hough Forests for 3D Object Detection and Pose Estimation," ECCV 2014
- [45] T. Hodan et al., "BOP: Benchmark for 6D Object Pose Estimation," ECCV 2018

5. 부록

[별첨 1] 유효성 검증 환경

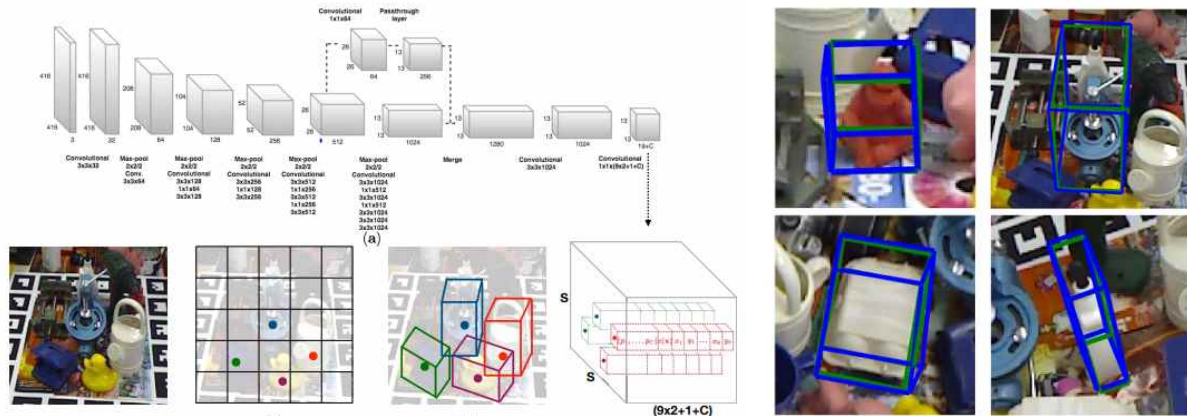
- 본 과제의 3D 객체 위치/자세 평가를 위해 baseline 인공지능 모델로 "Real-Time Seamless Single Shot 6D Object Pose Prediction (CVPR2018)"을 선정함.
- 실시간 객체 자세 추정을 YOLO와 유사한 구조로 제시하였음. 3D 객체 자세 추정 분야의 검증된 baseline 모델로 널리 인용되고, 최상위 학회에서의 후속 연구 역시 활발함.
- 인공지능 유효성 검증 환경 및 학습조건은 다음과 같음.

* 유효성 검증 환경 및 내용 정리

유효성 검증 항목	
항목명	객체 3D 학습 모델
검증 방법	도커 이미지 또는 현장 점검
목적	객체 3D 위치/자세 추정
지표	- 2D reprojection error, IoU metric에 따른 3D 위치/자세 정확도(%) - 레퍼런스 모델의 논문에서 설명한 방법과 동일하게 진행 (CVPR18)
측정 산식	<p>- 2D reprojection 오차 계산식:</p> $\frac{1}{N} \sum_{k=1}^N \ p_{gt,k}^{2D}(R_{gt}, t_{gt}) - p_{pr,k}^{2D}(R_{pr}, t_{pr}) \ _2$ <p>객체 3D 데이터의 바운딩 큐브의 8개의 꼭지점과 중심점이 영상에 투영되는 2D 픽셀 좌표셋 $p_{gt,k=1...9}^{2D}$에 대해 라벨링되면 해당 객체의 6자유도 위치/자세 정보인 R_{gt}, t_{gt}가 계산된다.</p> <p>학습 후 예측되는 $p_{pr,k=1...9}^{2D}$로부터 추정되는 R_{pr}, t_{pr}의 정확도 유효성은 $p_{gt,k=1...9}^{2D}(R_{gt}, t_{gt})$와 $p_{pr,k=1...9}^{2D}(R_{pr}, t_{pr})$의 L2 norm 평균이 10 pixels 안에 들어오는지 여부로 판단할 수 있다.</p> <p>- intersection over union (IoU) 오차 계산식: $\frac{A_{gt} \cap A_{pr}}{A_{gt} \cup A_{pr}}$</p> <p>객체 3D 데이터가 영상에 투영된 segmentation 마스크, A_{gt}를 라벨링한다. 테스트 영상에서는 학습한 모델을 통해 추정된 R_{pr}, t_{pr}로 객체 3D 모델을 투영한 예측 영역과 라벨링 된 영역이 잘 일치하는지 여부로 유효성을 판단할 수 있다. 레퍼런스 논문에서는 예측 영역과 라벨링 된 영역의 교집합과 합집합 비율이 50% 이상이 되어야 참값으로 판단하였다.</p>
실행 파일명	objectID/valid.py
도커 이미지	objectID/valid.py
유효성 검증 환경	
CPU	Intel i9-10700K / 10900K
Memory	NVIDIA GeForce RTX 3080 / 3090 GPU
GPU	Samsung DDR4-3200 32GB 이상
Storage	SSD 500GB 이상
OS	Windows 10 / Ubuntu 18.04

[별첨 2] 모델 학습 및 검증 조건

- 수집되는 3D 모델 정보, 3D 큐브, 개체 수준의 세그멘테이션에 대해 유효성 검증.
- 성능 분석은 논문에서 제시한 IoU score와 reprojection error에 기반 정확도로 판단. LINEMOD 데이터셋 Hinterstoisser, Stefan et.al. "Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes." ACCV 2012에 대한 재현 성능치를 근거.
- 인공지능 모델 학습 및 검증 조건은 다음과 같음.



[그림] 객체 3D 데이터 학습의 레퍼런스 모델 및 예상 결과

* 유효성 검증 모델 학습 및 검증 조건 정리

유효성 검증 모델 학습 및 검증 조건	
개발 언어	Python 3.6
프레임워크	PyTorch 1.8.0, CUDA 11.1, OpenCV 4.5.3.56, SciPy 1.2.0, Pillow 8.2.2
학습 알고리즘	Real-Time Seamless Single Shot 6D Object Pose Prediction (CVPR18) https://github.com/seongheum-ssu/nia_singleshotpose -->레퍼런스모델
학습 조건	optimizer: SGD, learning rate: 0.001, epochs: 5,000
파일 형식	학습/평가 데이터: .jpg (입력), .txt (자세), .png (영역), .ply (3D 데이터)
전체 구축 데이터 대비 모델에 적용되는 비율	시모델 사용 이미지 비율 (대용량 수량) - 30만장 이상 (클래스 당 인스턴스 1,000건 이상으로 100% 활용) ※ 유효성 검증은 구축된 데이터 전체를 적용하며, 변경이 필요한 경우 TTA 담당자와 협의한다.
모델 학습 과정별 데이터 분류 및 비율 정보	- Training 비율: 클래스별 인스턴스의 80% - Validation, Test 비율: 클래스별 인스턴스의 20%
제한사항	- 초과로 수집되는 가상/증강 영상을 통해 지표 개선 여부 검토 중. - 대용량 객체의 정면 방향을 정의할 수 있는 카테고리로 제한.

[별첨 3] 유효성 검증 체크리스트

- AI 모델 성능을 2D reprojection error 분석과 IoU score 분석 기반으로 accuracy 확인 및 평가
- 카테고리별 성능을 평균하여, 최종적으로 각각 82.5%와 87.5% 이상이 달성되도록 함.
- 품질 지표 및 목표 설정은 다음과 같음.

* 유효성 품질 지표 정리 (대용량)

품질특성	번호	항목명	측정 지표	정량 목표
유효성	4-1	객체 3D 자세 학습 모델	2D reprojection error 분석 기반 acc.	평균적으로 82.5% 이상
지표 및 목표 설정 근거				
<p>■ 품질측정 지표/목표 설정 근거</p> <p>- Real-Time Seamless Single Shot 6D Object Pose Prediction (CVPR18) 참고</p> <p>■ 품질측정 항목 관련 상세 설명</p> <p>* 대용량: 2D reproj. error 분석 기반 정확도 82.5% 이상</p> <p>- 2D reprojection 오차 계산식: $\frac{1}{N} \sum_{k=1}^N \ p_{gt,k}^{2D}(R_{gt}, t_{gt}) - p_{pr,k}^{2D}(R_{pr}, t_{gt}) \ _2$</p> <p>객체 3D 데이터의 바운딩 큐브의 8개의 꼭지점과 중심점이 영상에 투영되는 2D 픽셀 좌표셋 $p_{gt,k=1...9}^{2D}$에 대해 라벨링되면 해당 객체의 6자유도 위치/자세 정보인 R_{gt}, t_{gt}가 계산된다. 학습 후 예측되는 $p_{pr,k=1...9}^{2D}$로부터 추정되는 R_{pr}, t_{pr}의 정확도 유효성은 $p_{gt,k=1...9}^{2D}(R_{gt}, t_{gt})$와 $p_{pr,k=1...9}^{2D}(R_{pr}, t_{pr})$의 L2 norm 평균이 10 pixels 안에 들어오는지 여부로 판단할 수 있다.</p>				

품질특성	번호	항목명	측정 지표	정량 목표
유효성	4-2	객체 3D 위치 학습 모델	IoU 분석 기반 acc.	평균적으로 87.5% 이상
지표 및 목표 설정 근거				
<p>■ 품질측정 지표/목표 설정 근거</p> <p>- Real-Time Seamless Single Shot 6D Object Pose Prediction (CVPR18) 참고</p> <p>■ 품질측정 항목 관련 상세 설명</p> <p>* 대용량: IoU 기반 87.5% 이상</p> <p>- intersection over union (IoU) 오차 계산식: $\frac{A_{gt} \cap A_{pr}}{A_{gt} \cup A_{pr}}$</p> <p>객체 3D 데이터가 영상에 투영된 segmentation 마스크, A_{gt}를 라벨링한다. 테스트 영상에서는 학습한 모델을 통해 추정된 R_{pr}, t_{pr}로 객체 3D 모델을 투영한 예측 영역과 라벨링 된 영역이 잘 일치하는지 여부로 유효성을 판단할 수 있다. 레퍼런스 논문에서는 예측 영역과 라벨링 된 영역의 교집합과 합집합 비율이 50% 이상이 되어야 참값으로 판단하였다.</p>				

※ 작성된 품질 요구사항 정의서는 품질검증 전문기관(TTA 등)과 협의를 거쳐 조정될 수 있음

- 유효성 검증 체크리스트 정보는 다음과 같음.

★ 체크리스트 내용 정리 (대용량)

체크리스트		
1. 로그파일	시험 환경 로그	시험 환경 (CPU, GPU, RAM, HDD, OS)
	평가 수행 로그	1) 수행 조건 (학습 조건, 코드 실행 시간 등) 및 실행 명령어 관련 정보 출력
		2) 개별 결과값 (카테고리별 위치/자세 추정 정확도)
		3) 최종 결과값: 3-1. 대용량: 카테고리별 2D reproj. 기반 정확도가 평균적으로 82.5% 이상, IoU 기반 87.5% 이상.
		4) 최종 결과 계산 시 사용된 값 4-1. 2D reproj. 기반 정확도: 자세추정 reprojection error가 10 pixels 미만인 경우만 참인 것으로 판단. 4-2. IoU 기반 정확도: 예측한 자세 (R,t)로 3D 모델을 영상으로 투영하여 해당 객체 영역과 overlap이 50% 이상이어야 함.
2. 테스트셋	1) 모집단의 분포를 따르도록 선정한 유효성 검증에 사용된 random test set.	
	2) 구축 데이터셋에서 클래스별 인스턴스의 20%	

※ 작성된 품질 요구사항 정의서는 품질검증 전문기관(TTA 등)과 협의를 거쳐 조정될 수 있음

[별첨 4] 도커 환경 배포

- Docker image build 방법을 아래와 같이 정리. 레퍼런스 모델 빌드 예시.

* 도커 설치법

1. Install Docker Engine on Ubuntu

<https://docs.docker.com/engine/install/> --> server platform 선택하여 필요 버전 설치

- Optional post-installation steps 등 이후 필요 과정 진행

2. Install Docker Compose

<https://docs.docker.com/compose/install/> --> OS 선택하여 필요 과정 진행하여 설치

* 실행 방법

방법 1. docker build & run

1. build : 필요시 태그명 수정

`$source01_build.sh`

2. run : build 태그명 수정하였을 경우 최하단 태그명 동일하게 수정 필요

호스트 볼륨 경로 사용자 환경에 맞게 수정 필요, dropbox 다운로드 한 데이터 BG, cfg, data, LINEMOD 저장 폴더 지정

-v `/data/ssp/BG:/ssp/BG`

-v `/data/ssp/cfg:/ssp/cfg`

-v `/data/ssp/data:/ssp/data`

-v `/data/ssp/LINEMOD:/ssp/LINEMOD`

`$source02_run.sh`

방법 2. docker-compose

필요 부분 수정

`services:`

`NIA-SSP :`

`build:`

`context:./`

`dockerfile:./Dockerfile`

`image : nia-ssp:0.1`

`runtime:nvidia`

`container_name : NIA-SSP`

`volumes:`

- `/data/ssp/BG:/ssp/BG`

- `/data/ssp/cfg:/ssp/cfg`

- `/data/ssp/data:/ssp/data`

- `/data/ssp/LINEMOD:/ssp/LINEMOD`

docker build 후 `$ source 03_list_gpu.sh` 실행하여 gpu 정보 확인 및 테스트 가능.

- Docker image 테스트 예시.

```

nomad@devBOX-GPU ~/git/NIA_SSP source 02_run.sh
layer filters size input output
0 conv 32 3 x 3 / 1 416 x 416 x 3 -> 416 x 416 x 32
1 max 2 x 2 / 2 416 x 416 x 32 -> 208 x 208 x 32
2 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64
3 max 2 x 2 / 2 208 x 208 x 64 -> 104 x 104 x 64
4 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128
5 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64
6 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128
7 max 2 x 2 / 2 104 x 104 x 128 -> 52 x 52 x 128
8 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256
9 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128
10 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256
11 max 2 x 2 / 2 52 x 52 x 256 -> 26 x 26 x 256
12 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512
13 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256
14 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512
15 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256
16 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512
17 max 2 x 2 / 2 26 x 26 x 512 -> 13 x 13 x 512
18 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024
19 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512
20 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024
21 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512
22 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024
23 conv 1024 3 x 3 / 1 13 x 13 x1024 -> 13 x 13 x1024
24 conv 1024 3 x 3 / 1 13 x 13 x1024 -> 13 x 13 x1024
25 route 16
26 conv 64 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 64
27 reorg / 2 26 x 26 x 64 -> 13 x 13 x 256
28 route 27 24
29 conv 1024 3 x 3 / 1 13 x 13 x1280 -> 13 x 13 x1024
30 conv 20 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 20
31 detection
2021-09-16 11:30:29 Testing ape...
2021-09-16 11:30:29 Number of test samples: 1050
-----
tensor to cuda : 0.000521
forward pass : 0.005555
get_region_boxes : 0.009625
prediction time : 0.015702
eval : 0.003493
-----
2021-09-16 11:30:52 Results of ape
2021-09-16 11:30:52 Acc using 5 px 2D Projection = 93.24%
2021-09-16 11:30:52 Acc using 10% threshold - 0.010209801197378918 vx 3D Transformation = 21.81%
2021-09-16 11:30:52 Acc using 5 cm 5 degree metric = 41.81%
2021-09-16 11:30:52 Mean 2D pixel error is 5.299963, Mean vertex error is 0.036064, mean corner error is 3.905215
2021-09-16 11:30:52 Translation error: 0.035785 m, angle error: 6.353973 degree, pixel error: 5.299963 pix
valid.py:114: UserWarning: volatile was removed and now has no effect. Use `with torch.no_grad():` instead.
data = Variable(data, volatile=True)
/ssp/utils.py:242: UserWarning: Implicit dimension choice for softmax has been deprecated. Change the call to include dim=X as an argument.
cls_confs = torch.nn.Softmax()(Variable(output[2*num_keypoints+1:2*num_keypoints+1+num_classes].transpose(0,1))).data

```

※ 레퍼런스 AI 모델 자체 검증을 위한 샘플 데이터 테스트 예시

[별첨 5] 서버 환경 정보 (대용량)

- 인공지능 학습 데이터 유효성 검증을 위해 GPU 서버 클러스터를 구축.

* 학습 데이터 유효성 검증을 위한 GPU 서버 클러스터 (대용량)



* 학습 데이터 유효성 검증을 위한 GPU 서버 클러스터 상세 내용

도입기관		기관구분	참여	기관명	송실대학교 산학협력단
장 비 명	(국문)	학습 데이터 유효성 검증을 위한 GPU 서버 클러스터			
	(영문)	GPU Clusters for High-Performance Computing			
모 델 명		-			
제작국가		한국		제작사	DELL, 조립
구입방법		구매(O) 임차()		구입구분	내자() 외자(O)
구입일자(예정)		2021.09.30.		구입금액(예정)	52,790,000원
공동활용 여부		단독활용(), 기관내부활용(O), 기관외부활용(), 협의회 결정()			
장비 개요	필요성 및 규모 적합성	객체 3D 학습 데이터의 유효성 검증을 위한 GPU 컴퓨팅 자원 확보. 대용량 및 투명체 검증의 category별 검증 case마다 0.5 day의 시간 비용이 발생함에 따라, 다수의 GPU 클러스터를 도입하여 이를 병렬적으로 처리. 구축한 데이터가 실제로 학습에 유효한지 판단함.			
	원리 및 특징	총 12대의 NVIDIA GeForce RTX 3090로 구성되었고, CPU 및 SSD 자원 활용과 함께 최대 15kW의 전력을 수용할 수 있도록 플랫폼 환경이 최적화 됨. 병렬적으로 검증 cases를 처리할 수 있도록 설계. 가상이미지 생성을 위한 렌더팜으로 고성능 컴퓨팅 자원 활용 가능.			
	주요사양	<ul style="list-style-type: none"> • GPU 구성 : NVIDIA GeForce RTX 3090 12대 • CPU 구성 : Intel CPU 10700K 및 10900K 조합 • 동시 처리 가능한 검증 test case 수: 12건 • 인스턴스별 검증 시간: 0.5 day • 가상 데이터 생성 및 강화 학습을 위한 렌더팜 지원 가능 • 대용량 데이터를 저장 및 관리할 수 있는 고속 SSD 자원 • 검수 완료 데이터를 수집하여 관리하는 NAS 시스템과 연동 			
	사용용도 (활용방안)	객체 3D 데이터의 3D 모델, 3D 큐브, 및 객체 영역이 실제로 학계에서 인용할 수 있는 품질을 갖추게 되는지를 확인. Microsoft사의 레퍼런스 모델을 통해 category별로 약 1200장 단위를 1개의 test case로 병렬적으로 검증 가능. 유효성 검증 결과를 신속하게 피드백.			
활용 계획	수요예측 및 활용빈도	시스템 구축 및 전기 공사 완료 후 10월부터 활용 가능. 가공 및 검수 완료된 데이터가 빠르게 축적되는 11월~12월에 집중적으로 활용될 것으로 예측함. 저성능 임대 GPU 보다 월등한 처리 속도로 상시적으로 GPU를 활용할 수 있을 것으로 예상. 객체 3D 데이터를 활용한 응용서비스 지원에도 본 플랫폼이 계속 활용되며 기여할 예정.			
	운영비 확보 방안	송실대학교에서는 GPU 클러스터를 과제 기간에 계속적으로 활용할 수 있도록 공간을 대여함. 전기료 및 인터넷 사용에 따른 비용 지원.			
	전담인력 확보 방안	석사급 연구인력 1명이 GPU 서버 클러스터를 전담하여 관리할 예정. 학사급 연구인력 3명이 이를 보조해 학습 유효성 검증을 진행.			
	유지보수 방안	공개 입찰을 통해 (주)원퓨트가 협력 업체로 선정됨. 국내 GPU 공급 이슈로 10대분을 미국 현지에서 조립된 형태로 수입하게 됨. (주)원퓨트에서는 DELL사로부터 2년 이상의 품질 책임서를 발급 받음.			
	기타	사업 계획에 있어, 고성능 및 학습용 GPU로 분리하여 구매가 추진되었으나, 약 5천만원 정도의 단일 AI 플랫폼을 초기 단계부터 계획함. 국내/해외 공급 이슈로 자재 공급 및 가격 변동성에 적절히 대응함. 특히, 최근 GPU 가격이 급격히 상승하여 수량 조정이 있었음.			

- GPU 단가 상승 및 공급 부족으로 인해 계획 대비 수량은 확보하지 못하였으나, 방학 중 AI 실습실을 활용하여 학습 유효성 검증 진행 가능.
- 학교에 동의를 구하여, NVIDIA GeForce RTX 3080 / 3090을 탑재한 45대 컴퓨터를 1월말까지 NIA 과제를 위해 독점적으로 사용할 수 있음.
- 모든 컴퓨팅 자원을 가용할 시, 대용량의 모든 test case의 유효성 검증을 3~4일 안으로 수행 가능. 투명체의 경우도 약 1주일 안에 모든 test case의 유효성 검증 진행 가능.

*** 학습 데이터 유효성 검증을 위한 AI 실습실 자원 대여 (추가 자원 확보 부분)**



(2022년 1월말까지 객체 3D 데이터 유효성 검증을 위해 실습실 대여, NIA 과제를 위해서만 사용 예정)