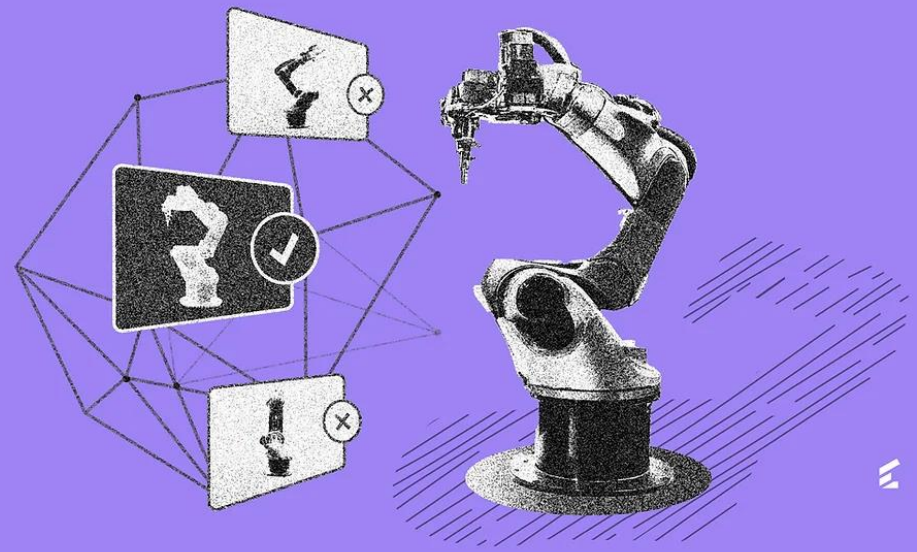
Boostcamp Robotics Study

PWNet: Point Wise Voting Network !

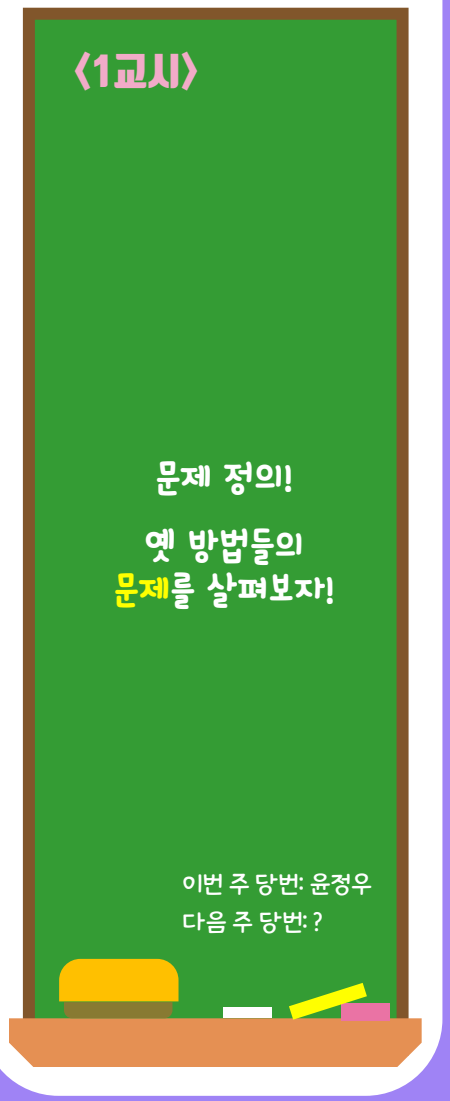
Jungwoo Yoon



Index


ChatGPT는 신이야!

PVNet



INTRO: Problem Statement

1

	전통적 방법	딥러닝 기반 방법	SOTA! PoseCNN
특징	hand-crafted features 기반 객체 이미지 – 객체 모델 간 매칭	CNN 기반 학습	2D 키포인트 회귀 -> PnP 기반 6D Pose 예측
한계	Image variation Background Clutter (복잡한 배경)	일반화 문제	Occlusion (가림) Truncation (잘림)

PVNet

〈1교시〉

새로운 방법!

논문에서 소개하는
새 구조를 잘 살펴보자!

이번 주 당번: 윤정우
다음 주 당번: ?

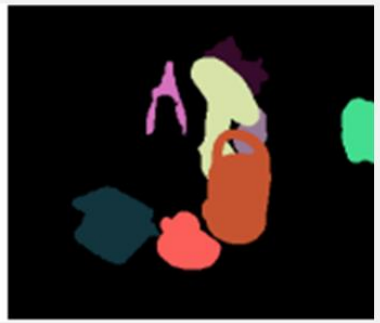
INTRO: Proposed Approach

1

PVNet

1

Dense Prediction



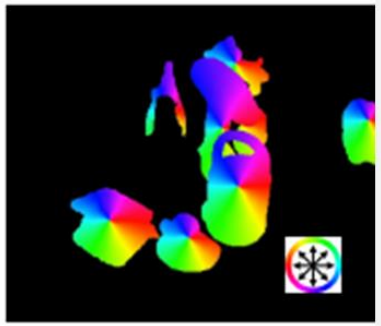
픽셀 혹은 영역별 label 예측

\+

RANSAC

2

Vector-field Prediction



객체의 각 픽셀에서 키포인트 방향
가리키는 벡터 예측

Keypoint Localization

PVNet

〈1교시〉

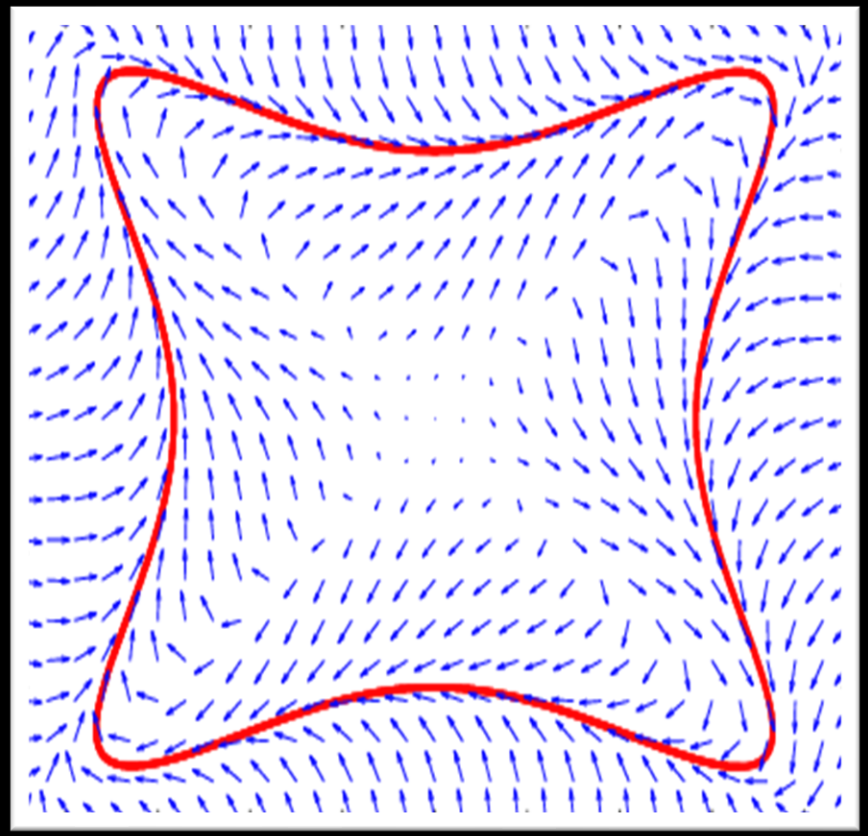
새로운 방법 번외!

논문에 나오는
벡터장이 뭘까!

이번 주 당번: 윤정우
다음 주 당번: ?

INTRO: Proposed Approach

1



벡터장 (Vector-field)

- 벡터: (기하학적) 크기와 방향을 함께 가지는 양
- 벡터장: 유클리드 공간의 각 점에 벡터를 대응시킨 것
- 효과: 네트워크가 객체의 local feature와 객체 부분 간 공간적 관계에 집중 유도

  => 객체의 보이는 부분으로 보이지 않는 부분 예측

PVNet

〈2교시〉

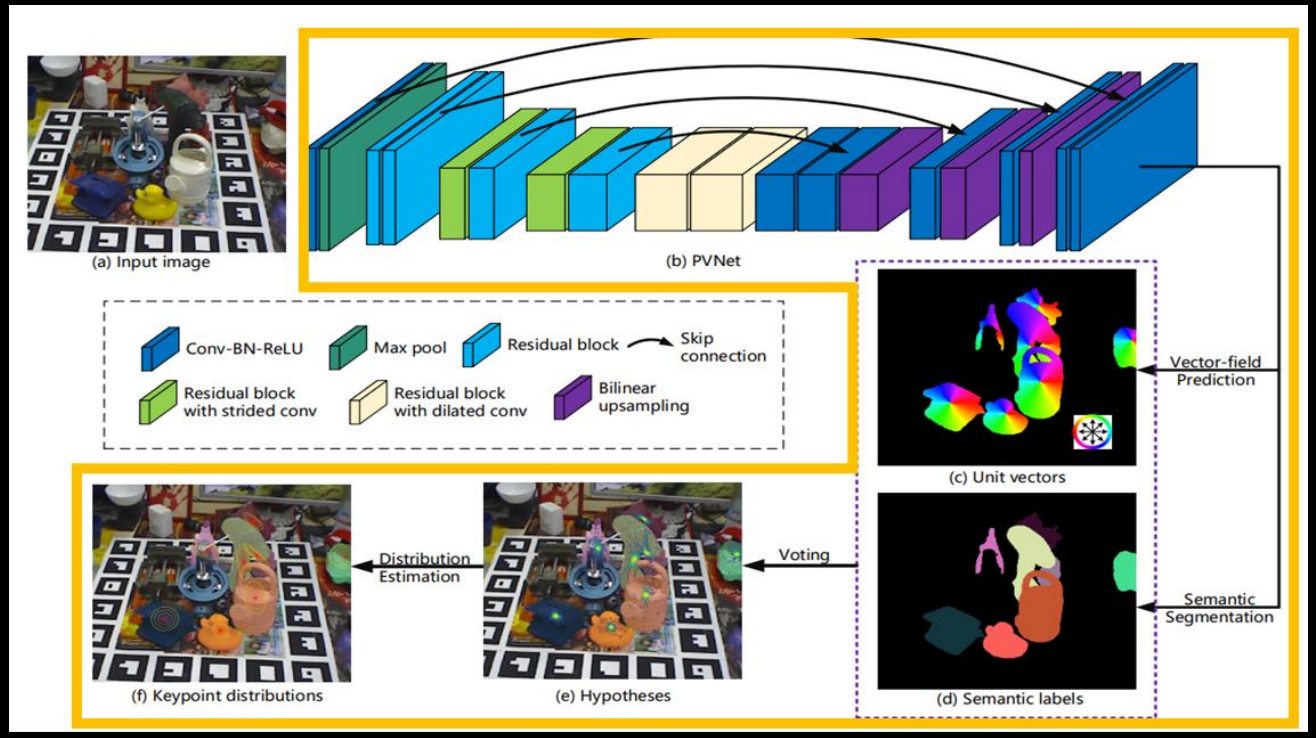
PVNet의 구조!

PVNet은 2단계!

이번 주 당번: 윤정우

다음 주 당번: ?

PVNet: Overall Network Architecture



First Stage

- First Stage : Voting-based Keypoints Localization
- Second Stage : Uncertainty-based PnP

PVNet

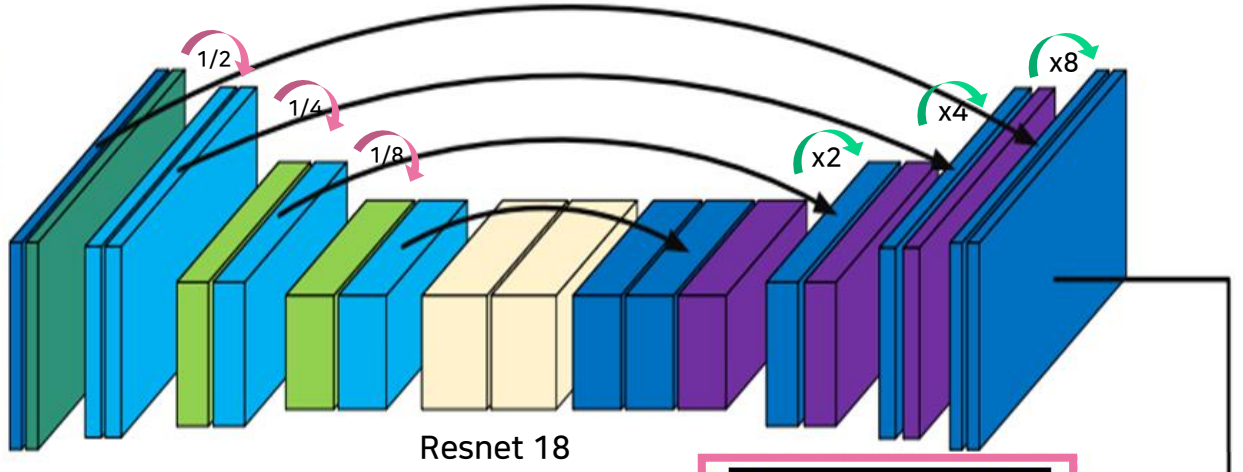
〈2교시〉

PVNet의 1단계!

Backbone으로
2가지 작업을 한대!

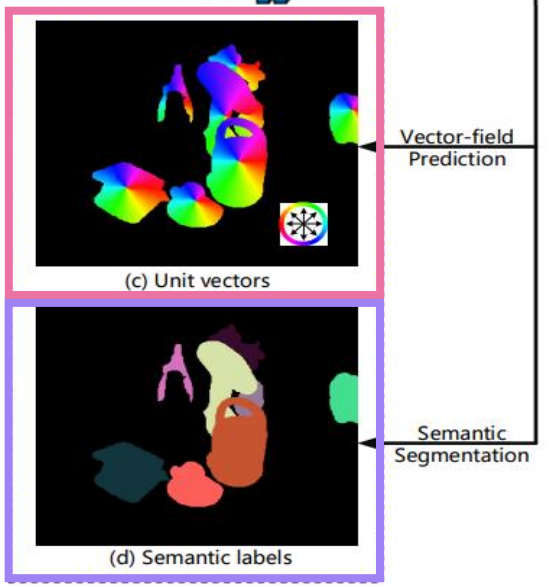
이번 주 당번: 윤정우
다음 주 당번: ?

PVNet: Voting-based Keypoint Localization

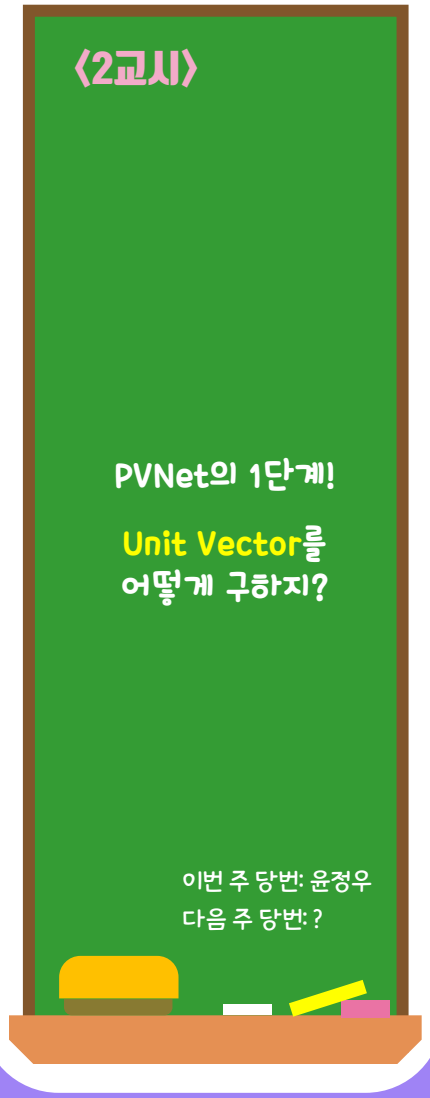


Unit Vectors

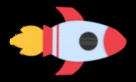
Pixel Label



PVNet



PVNet: Voting-based Keypoints Localization



Unit Vector 수식

$$v_k(p) = \frac{x_k - p}{\|x_k - p\|_2}$$

- $v_k(p)$: Unit Vector
- x_k: 객체 키포인트 좌표
- p: 객체의 한 pixel 좌표
- $x_k - p$: pixel p 에서 키포인트 x_k로의 벡터
- $\|x_k - p\|_2$: pixel p와 키포인트 x_k 간 유클리드 거리

PVNet

〈2교시〉

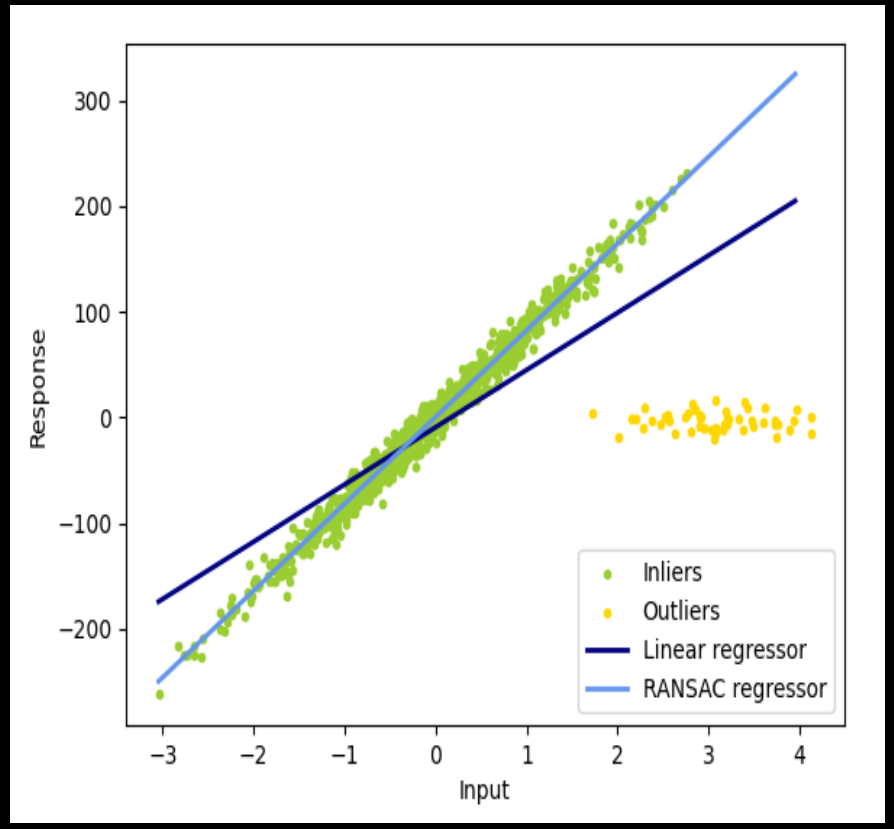
PVNet 1단계 번외!

논문에 나오는
RANSAC이 뭘까!

이번 주 당번: 윤정우
다음 주 당번: ?

PVNet: Voting-based Keypoints Localization

2

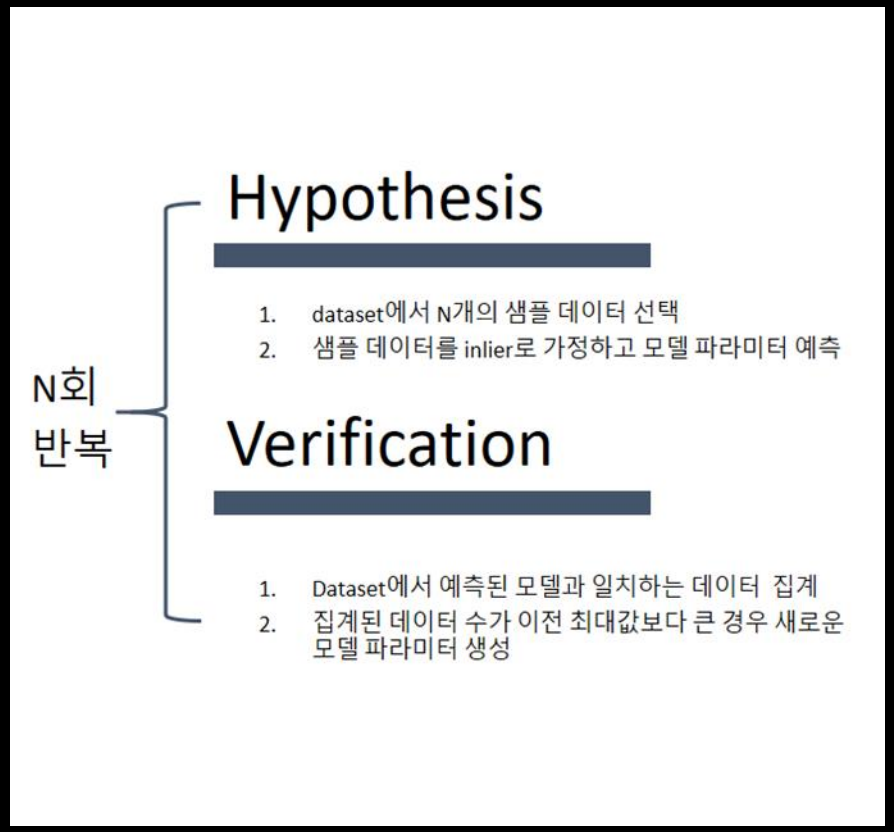


RANSAC (RANdom SAmple consensus)

- 정의: 특정 임계값 이상의 outlier 데이터를 무시하고, 샘플을 무작위 추출해 최대 데이터가 일치하는 이상적인 모델을 추출하는 알고리즘

PVNet

PVNet: Voting-based Keypoints Localization



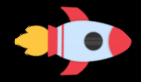
RANSAC (RANdom SAmple consensus)

$$p = 1 - (1 - \alpha^m)^N$$

- P: inlier로만 이뤄진 샘플을 획득할 확률
- alpha: dataset에서 inlier의 비율
- m: 회당 추출하는 데이터 수
- N: 알고리즘 반복 회수
- T: inlier / outlier 구분 기준

PVNet

PVNet: Voting-based Keypoints Localization



Voting (Confidence) Score 수식

$$w_{k,i} = \sum_{p \in O} I\left(\frac{(h_{k,i} - p)^T}{\|h_{k,i} - p\|_2} v_k(p) \geq \theta \right)$$

주요 변수

- $h_{k,i}$: k번째 키포인트에 대한 i번째 가설
- $w_{k,i}$: 가설 $h_{k,i}$의 투표 (신뢰) 점수
- O: 객체의 모든 픽셀 p의 집합 = 투표에 참여하는 픽셀들
- p: 객체의 한 pixel 좌표
- $v_k(p)$: pixel p 에서 예측한 unit vector
- θ: threshold 값
- I: indicator 함수 -> 괄호 안 조건이 참이면 1, 거짓일 경우 0 반환

PVNet

PVNet: Voting-based Keypoints Localization

2

Voting (Confidence) Score 수식

$$w_{k,i} = \sum_{p \in O} I\left(\frac{(h_{k,i} - p)^T}{\|h_{k,i} - p\|_2} v_k(p) \geq \theta \right)$$

주요 수식

- $v_k(p)$: pixel p 에서 키포인트 $h_{k,i}$로의 예측 unit vector
- $\frac{(h_{k,i}-p)}{\|h_{k,i}-p\|_2}$: 가설 $h_{k,i}$에서 픽셀 p로 향하는 실제 unit vector
- $\frac{(h_{k,i}-p)^T}{\|h_{k,i}-p\|_2} v_k(p)$: 내적 값 = 실제 unit vector와 예측 unit vector 간 일치도 (방향 유사성) 계산
- $I(\frac{(h_{k,i}-p)^T}{\|h_{k,i}-p\|_2} v_k(p) \geq \theta)$: 일치도가 threshold 이상일 경우 1, 미만일 경우 0 반환
- $w_{k,i}$: 가설 $h_{k,i}$의 최종 투표 점수

PVNet

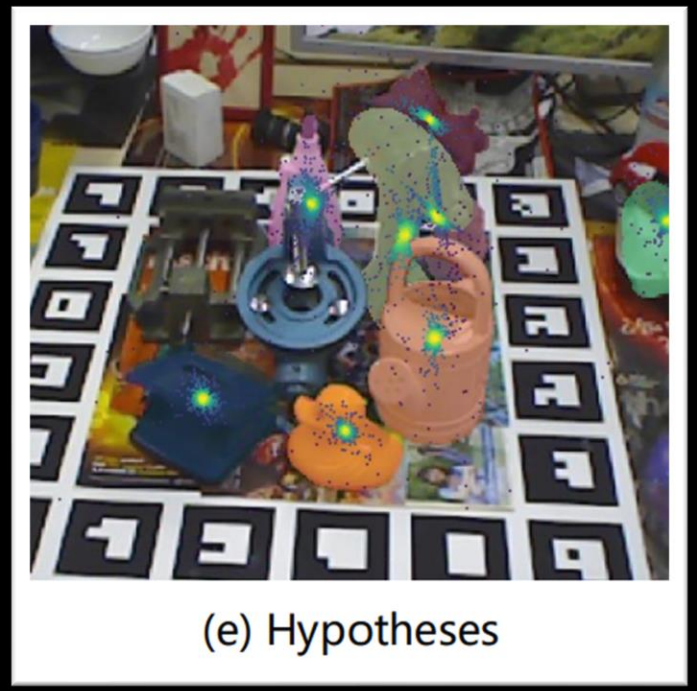
〈2교시〉

PVNet 1단계!

2D Keypoint
공간적 확률분포 Get!

이번 주 당번: 윤정우
다음 주 당번: ?

PVNet: Voting-based Keypoints Localization



(e) Hypotheses

예측한 2D Keypoints의 공간적 확률분포 get!

평균 $\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} w_{k,i}\mathbf{h}_{k,i}}{\sum_{i=1}^{N} w_{k,i}}$, 공분산 $\boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{N} w_{k,i}(\mathbf{h}_{k,i} - \boldsymbol{\mu}_k)(\mathbf{h}_{k,i} - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} w_{k,i}}$,

PVNet

〈2교시〉

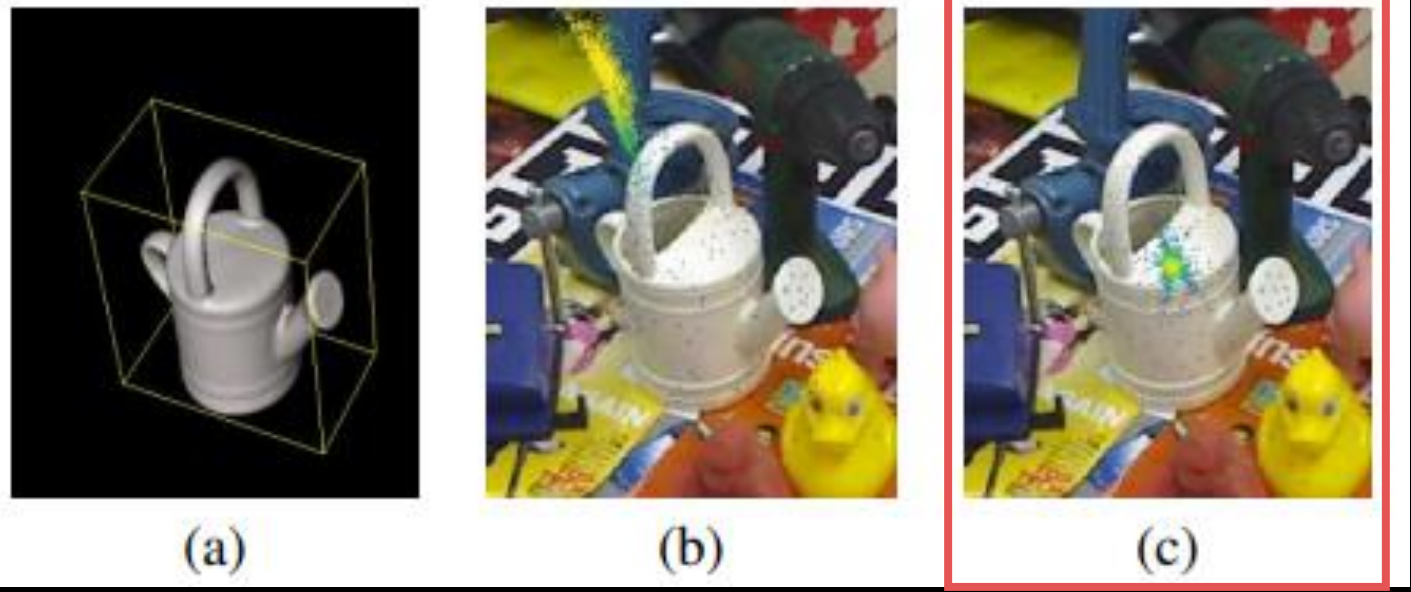
PVNet 1단계!

3D Keypoint는
어떻게 구할까?

이번 주 당번: 윤정우
다음 주 당번: ?

PVNet: Voting-based Keypoints Localization

3D Keypoints Selection



객체 표면에서 FPS 알고리즘으로 K개의 3D keypoints get!

PVNet

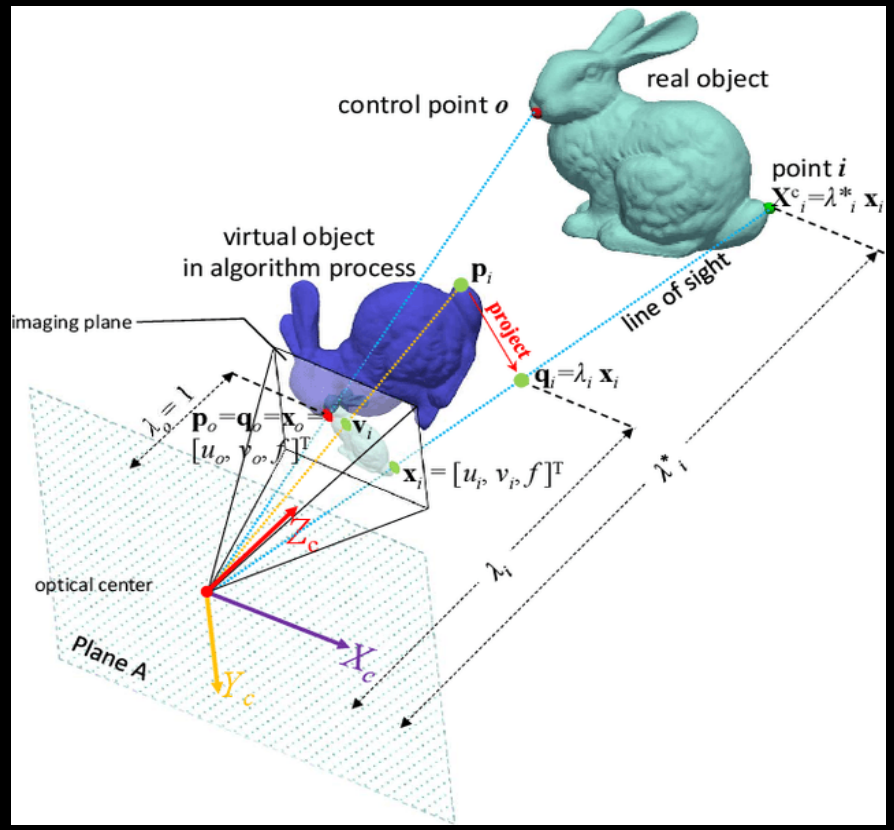
〈3교시〉

PVNet의 두 번째 단계!

그런데 잠깐?!
PnP가 뭐였더라?

이번 주 당번: 윤정우
다음 주 당번: ?

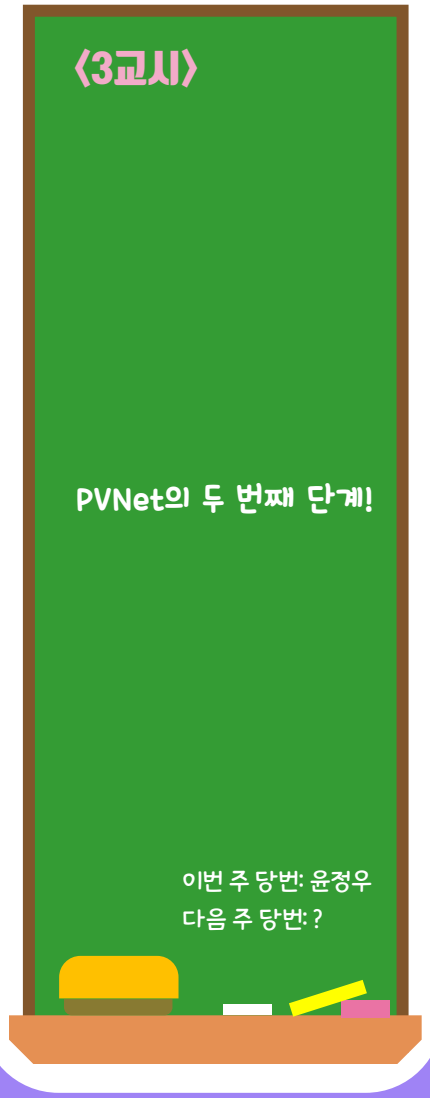
PVNet: Uncertainty-based PnP



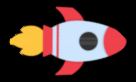
PnP (Perspective-n-Point)

- 정의: 이미지의 3D points 집합과
  이에 대응하는2D 투영이 주어졌을 때,
  카메라의 6DoF Pose를 추정하는 알고리즘

PVNet



PVNet: Uncertainty-based PnP



PVNet PnP 수식

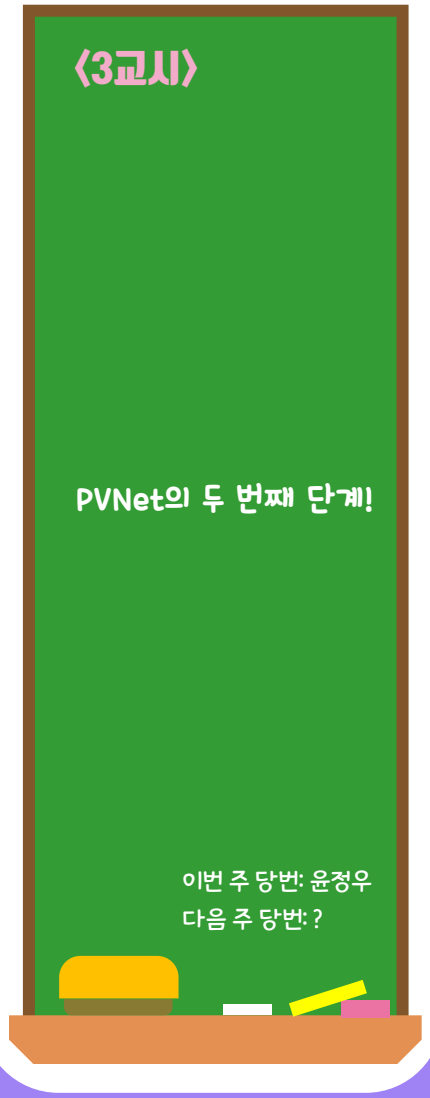
$$\underset{R,\mathbf{t}}{\text{minimize}} \sum_{k=1}^{K} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k),$$

$$\tilde{\mathbf{x}}_k = \pi(R\mathbf{X}_k + \mathbf{t}),$$

주요 변수

- X_k: 3D 모델의 k번째 keypoint 위치
- $\tilde{x}_k$: X_k의 2D 투영 keypoint 위치
- R, t: 카메라의 rotation / translation 좌표 = 6DoF pose parameter
- π: perspective projection (원근 투영) 함수
- μ_k: 예측한 2D keypoints의 위치
  = 예측한 2D keypoints의 공간적 확률분포 평균
- $\sum_k$: 예측한 2D keypoint의 공분산 행렬 = 해당 keypoint 위치 추정의 불확실 정도
  -> $\sum_k^{-1}$: 공분산 행렬의 분포를 반영한 가중치

PVNet



PVNet: Uncertainty-based PnP

3

PVNet PnP 수식

$$\underset{R,\mathbf{t}}{\text{minimize}} \sum_{k=1}^{K} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k),$$

$$\tilde{\mathbf{x}}_k = \pi(R\mathbf{X}_k + \mathbf{t}),$$

주요 수식

- R, t: (초기값) EPnP 기법으로 산출
  -> 공분산 행렬에서 불확실성이 가장 작은 4개의 키포인트를 기준으로 초기 6D Pose 추정
- $(\tilde{x}_k - \mu_k)^T \Sigma_k^{-1} (\tilde{x}_k - \mu_k)$: Mahalanobos 거리 = 두 벡터 간 차이를 공분산 행렬을 반영하여 계산
  = 가중치가 반영된 재투영 오차
- $\sum_{k=1}^{K} \quad (\tilde{x}_k - \mu_k)^T \Sigma_k^{-1} (\tilde{x}_k - \mu_k)$: 키포인트별 가중치가 반영된 재투영 오차 합계
- $minimize_{R,t}$: Levenberg-Marquardt 알고리즘을 사용하여 R, t 최적화 -> 6D pose 추정

PVNet

〈3교시〉

PVNet의 두 번째 단계!

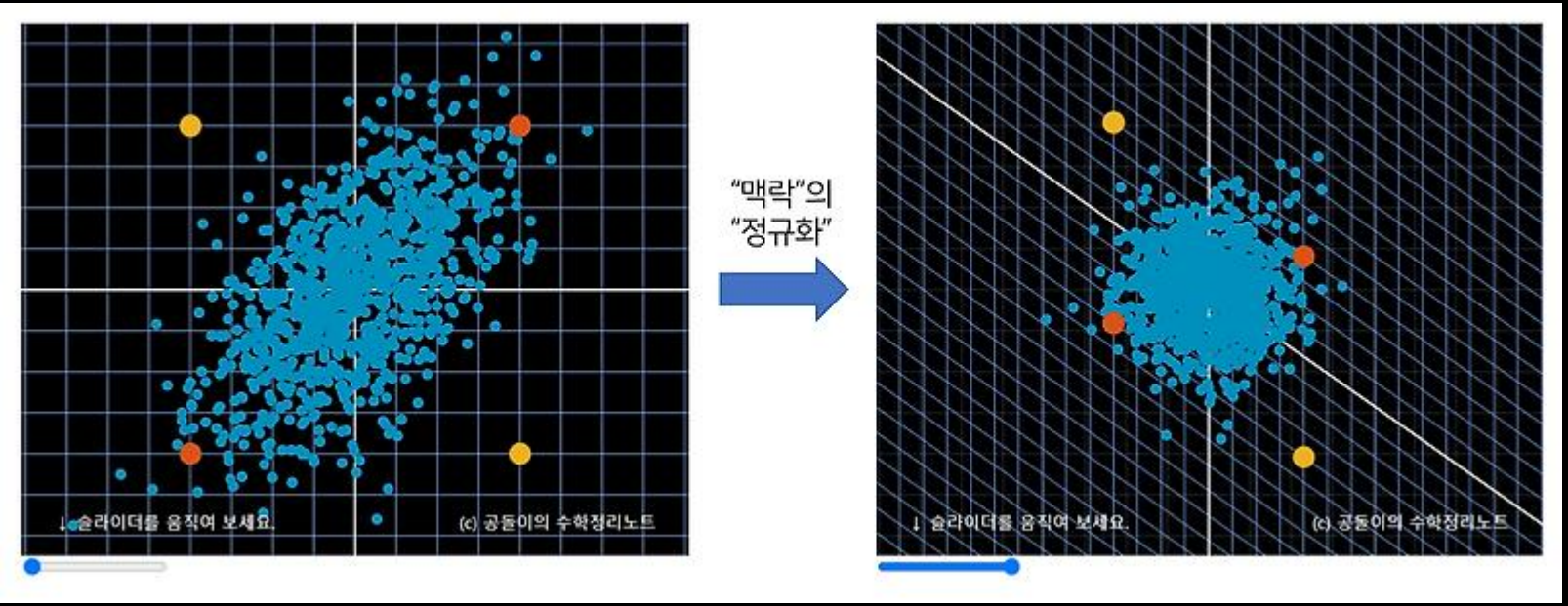
그런데 잠깐?!
PnP가 뭐였더라?

이번 주 당번: 윤정우
다음 주 당번: ?

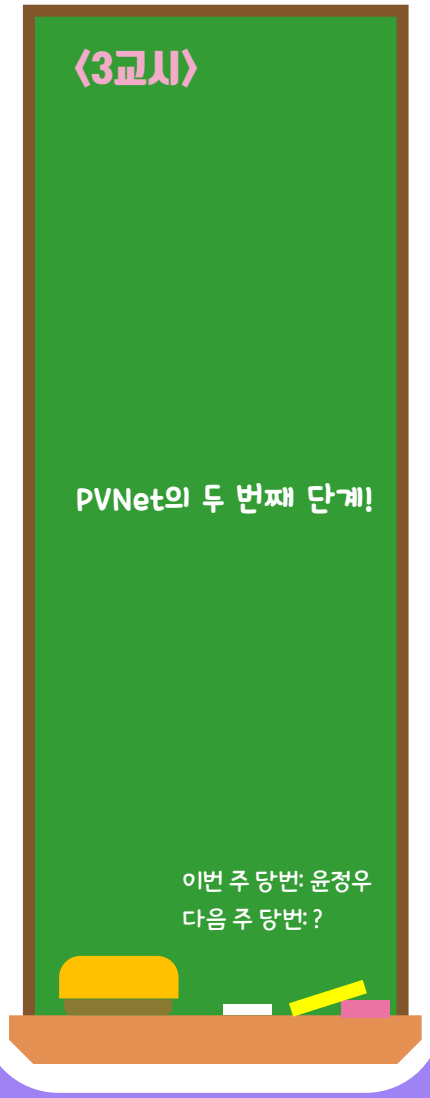
PVNet: Uncertainty-based PnP

Mahalanobis Distance

- 정의: 데이터들의 분포를 통해 맥락을 조사하고,
  이를 정규화 한 뒤에 유클리드 거리를 계산
- 특징: 분포(맥락) 기반의 상대적인 거리



PVNet



PVNet: Uncertainty-based PnP

3

PVNet PnP 수식

$$\underset{R,\mathbf{t}}{\text{minimize}} \sum_{k=1}^{K} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k),$$

$$\tilde{\mathbf{x}}_k = \pi(R\mathbf{X}_k + \mathbf{t}),$$

주요 수식

- R, t: (초기값) EPnP 기법으로 산출
  -> 공분산 행렬에서 불확실성이 가장 작은 4개의 키포인트를 기준으로 초기 6D Pose 추정
- $(\tilde{x}_k - \mu_k)^T \Sigma_k^{-1} (\tilde{x}_k - \mu_k)$: Mahalanobos 거리 = 두 벡터 간 차이를 공분산 행렬을 반영하여 계산
  = 가중치가 반영된 재투영 오차
- $\sum_{k=1}^{K} \quad (\tilde{x}_k - \mu_k)^T \Sigma_k^{-1} (\tilde{x}_k - \mu_k)$: 키포인트별 가중치가 반영된 재투영 오차 합계
- $minimize_{R,t}$: Levenberg-Marquardt 알고리즘을 사용하여 R, t 최적화 -> 6D pose 추정

PVNet

〈4교시〉 실습시간~

PVNet으로 실험해보기!

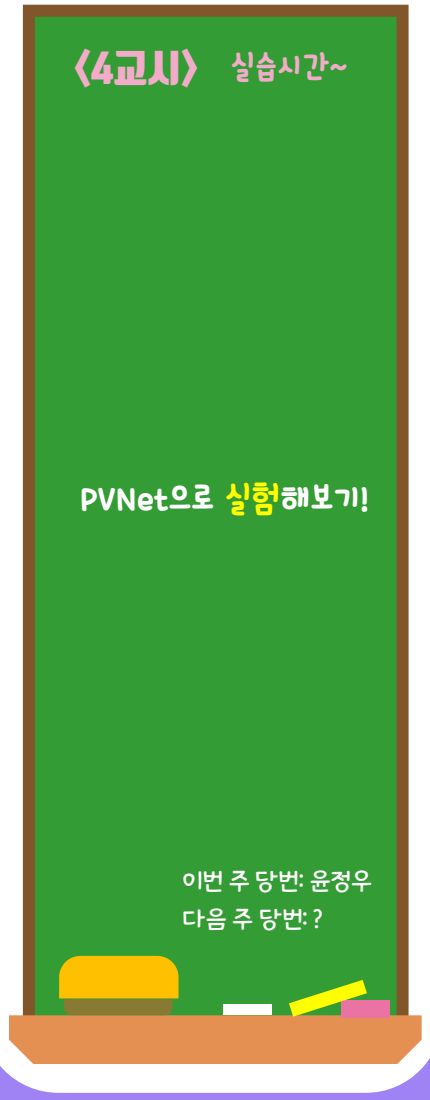
이번 주 당번: 윤정우
다음 주 당번: ?

Experiment: Performance Evaluation

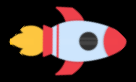
2D Projection metric. This metric computes the mean distance between the projections of 3D model points given the estimated pose and the ground-truth pose. A pose is considered as correct if the distance is less than 5 pixels.

ADD metric. We compute the mean distance between two transformed model points using the estimated pose and the ground-truth pose. When the distance is less than 10% of the model's diameter, it is claimed that the estimated pose is correct. For symmetric objects, we use the ADD-S metric [40], where the mean distance is computed based on the closest point distance. When evaluating on the YCB-Video dataset, we compute the ADD(-S) AUC proposed in [40].

PVNet



Experiment: Performance Evaluation



methods	w/o refinement			w/ refinement
	BB8 [30]	Tekin [36]	OURS	BB8 [30]
ape	95.3	92.10	**99.23**	96.6
benchwise	80.0	95.06	**99.81**	90.1
cam	80.9	93.24	**99.21**	86.0
can	84.1	97.44	**99.90**	91.2
cat	97.0	97.41	**99.30**	98.8
driller	74.1	79.41	**96.92**	80.9
duck	81.2	94.65	**98.02**	92.2
eggbox	87.9	90.33	**99.34**	91.0
glue	89.0	96.53	**98.45**	92.3
holepuncher	90.5	92.86	**100.0**	95.3
iron	78.9	82.94	**99.18**	84.8
lamp	74.4	76.87	**98.27**	75.8
phone	77.6	86.07	**99.42**	85.3
average	83.9	90.37	**99.00**	89.3

methods	w/o refinement				w/ refinement	
	BB8 [30]	SSD-6D [17]	Tekin [36]	OURS	BB8 [30]	SSD-6D [17]
ape	27.9	0.00	21.62	43.62	40.4	**65**
benchwise	62.0	0.18	81.80	**99.90**	91.8	80
cam	40.1	0.41	36.57	**86.86**	55.7	78
can	48.1	1.35	68.80	**95.47**	64.1	86
cat	45.2	0.51	41.82	**79.34**	62.6	70
driller	58.6	2.58	63.51	**96.43**	74.4	73
duck	32.8	0.00	27.23	52.58	44.30	**66**
eggbox	40.0	8.90	69.58	99.15	57.8	**100**
glue	27.0	0.00	80.02	95.66	41.2	**100**
holepuncher	42.4	0.30	42.63	**81.92**	67.20	49
iron	67.0	8.86	74.97	**98.88**	84.7	78
lamp	39.9	8.20	71.11	**99.33**	76.5	73
phone	35.2	0.18	47.74	**92.41**	54.0	79
average	43.6	2.42	55.95	**86.27**	62.7	79

PVNet

〈4교시〉 실습시간~

PVNet으로 실험해보기!

이번 주 당번: 윤정우
다음 주 당번: ?

Experiment: Performance Evaluation

methods	Tekin [36]	PoseCNN [40]	Oberweger [27]	OURS
ape	7.01	34.6	**69.6**	69.14
can	11.20	15.1	82.6	**86.09**
cat	3.62	10.4	65.1	**65.12**
duck	5.07	31.8	61.4	**61.44**
driller	1.40	7.4	**73.8**	73.06
eggbox	-	1.9	**13.1**	8.43
glue	4.70	13.8	54.9	**55.37**
holepuncher	8.26	23.1	66.4	**69.84**
average	6.16	17.2	60.9	**61.06**

Table 4. The accuracies of our method and the baseline methods on the **Occlusion LINEMOD** dataset in terms of **2D projection**.

methods	Tekin [36]	PoseCNN [40]	Oberweger [27]	OURS
ape	2.48	9.6	**17.6**	15.81
can	17.48	45.2	53.9	**63.30**
cat	0.67	0.93	3.31	**16.68**
duck	1.14	19.6	19.2	**25.24**
driller	7.66	41.4	62.4	**65.65**
eggbox	-	22	25.9	**50.17**
glue	10.08	38.5	39.6	**49.62**
holepuncher	5.45	22.1	21.3	**39.67**
average	6.42	24.9	30.4	**40.77**

Table 5. The accuracies of our method and the baseline methods on the **Occlusion LINEMOD** dataset in terms of the **ADD(-S)** metric, where glue and eggbox are considered as symmetric objects.

PVNet

〈4교시〉 실습시간~

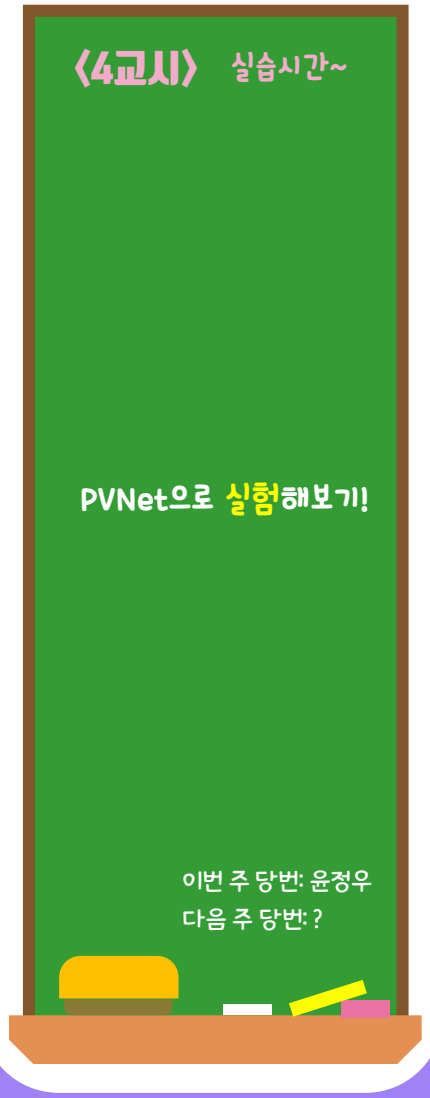
PVNet으로 실험해보기!

이번 주 당번: 윤정우
다음 주 당번: ?

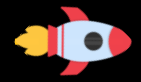
Experiment: Performance Evaluation

objects	ape	benc-hvise	cam	can	cat	driller	duck
2D Projection	52.59	58.19	54.87	57.44	61.66	43.27	54.23
ADD(-S)	12.78	42.80	27.73	32.94	25.19	37.04	12.36
objects	eggbox	glue	holep-uncher	iron	lamp	phone	avg
2D Projection	87.23	86.64	53.84	46.53	46.94	51.35	58.06
ADD(-S)	44.13	38.11	22.39	42.01	40.91	30.86	31.48

PVNet

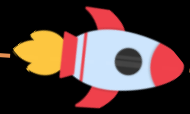


Experiment: Performance Evaluation



methods	PoseCNN [40]	Oberweger [27]	OURS
2D Projection	3.72	39.4	47.4
ADD(-S) AUC	61.0	72.8	73.4

PVNet



The END

FoundationPose