PVNET:Pixel-wise Voting Network for 6DoF Pose Estimation

Sida Peng, Yuan Liu, Qixing Huang, Hujun Bao, Xiaowei Zhou

KIST
송명하

Korea Institute of Science and Technology
한국과학기술연구원

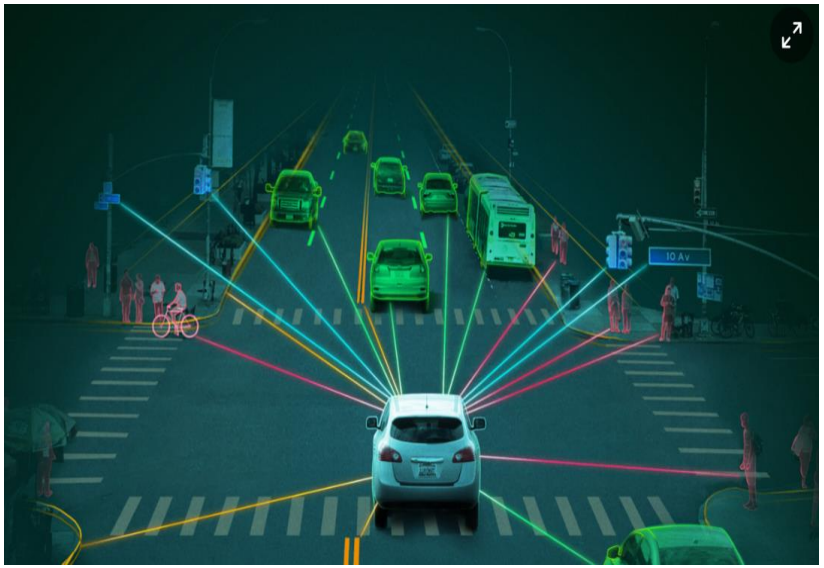
Content

1. Introduction

Object pose estimation 목적 :

detect objects and estimation their orientations and translations relative to a canonical frame.

http://www.aitimes.com/news/articleView.html?idxno=57605

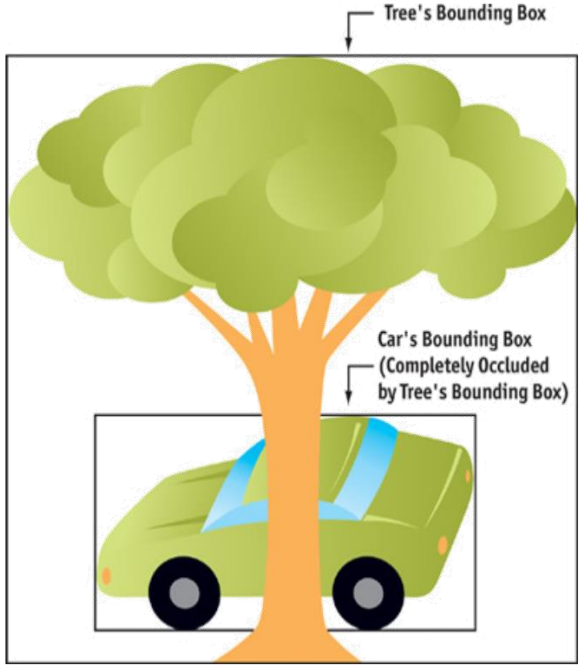


http://www.newstof.com/news/articleView.html?idxno=1134

1. Introduction

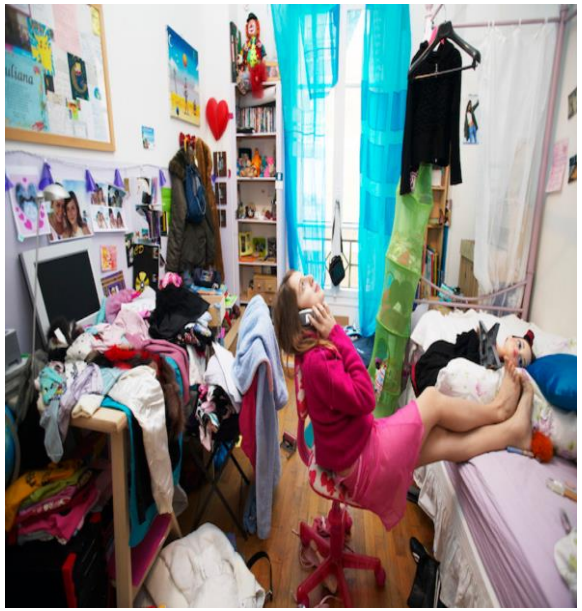
Challenge

Occlusion



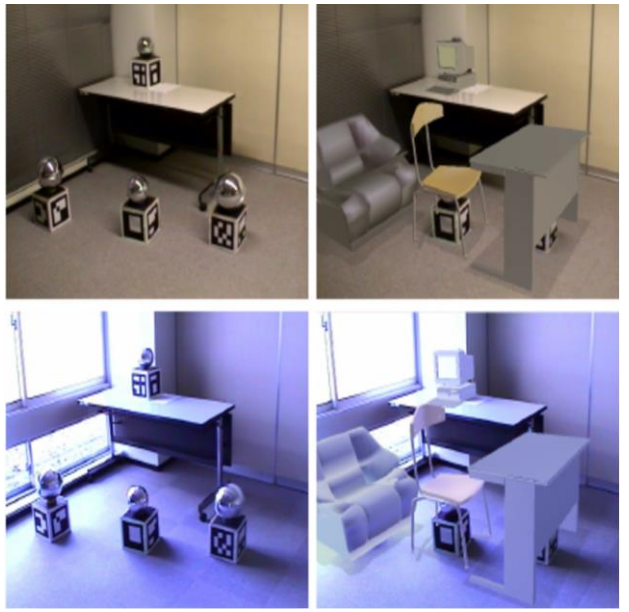
https://developer.nvidia.com/sites/all/modules/custom/gpugems/books/GPUGems/gpugems_ch29.html

Clutter



https://thespinoff.co.nz/society/09-01-2020/im-messy-and-im-done-apologising-for-it/

Lighting condition

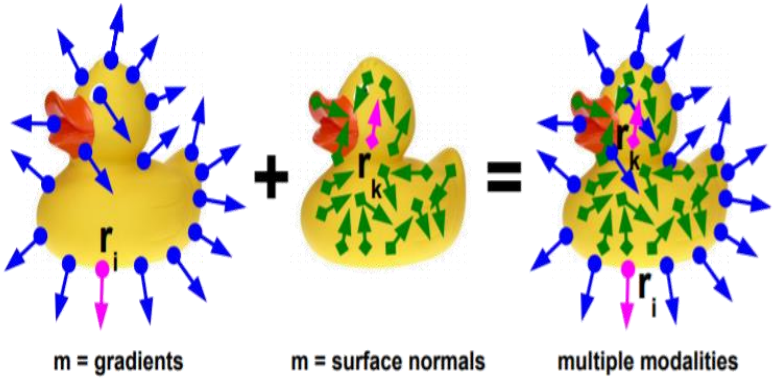


https://www.researchgate.net/figure/Different-Lighting-Condition-for-a-Whole-Room-fluorescent-lamp-top-day-light-from_fig4_221209750

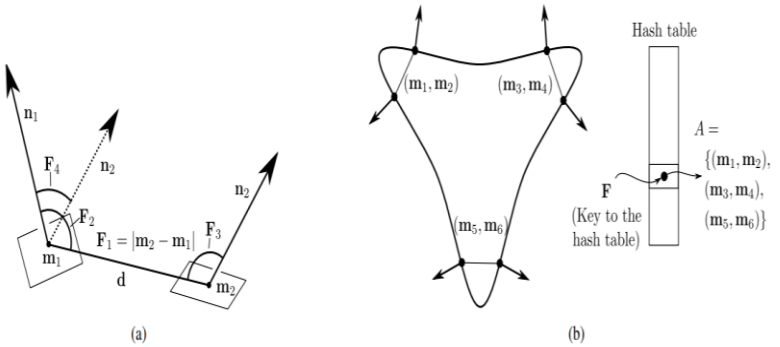
1. Introduction

Challenge

Traditional method



Multimodal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes(ICCV2011)



Model Globally, Match Locally: Efficient and Robust 3D Object Recognition (CVPR2010)

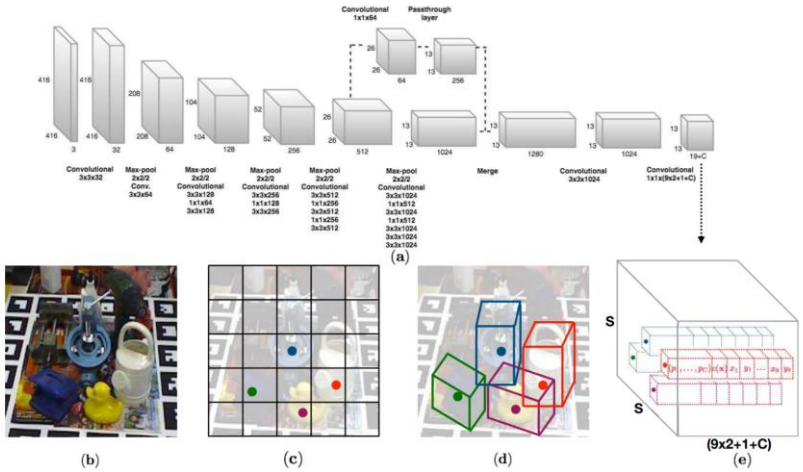
Hand craft feature를 만들어서 문제를 해결하려 함 but 일반화되지 못함.

1. Introduction

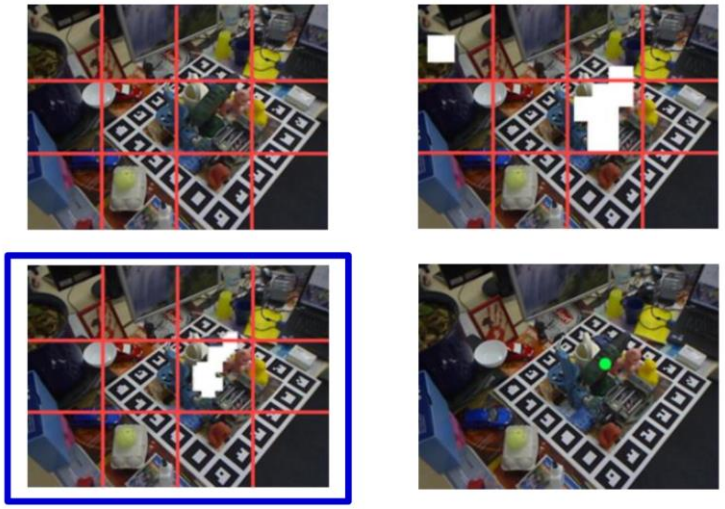
Challenge

Deep Learning based method

Deep learning based methods train end-to-end neural networks that take an image as input and output its corresponding pose.



Real-Time Seamless Single Shot 6D Object Pose Prediction -CVPR 2018

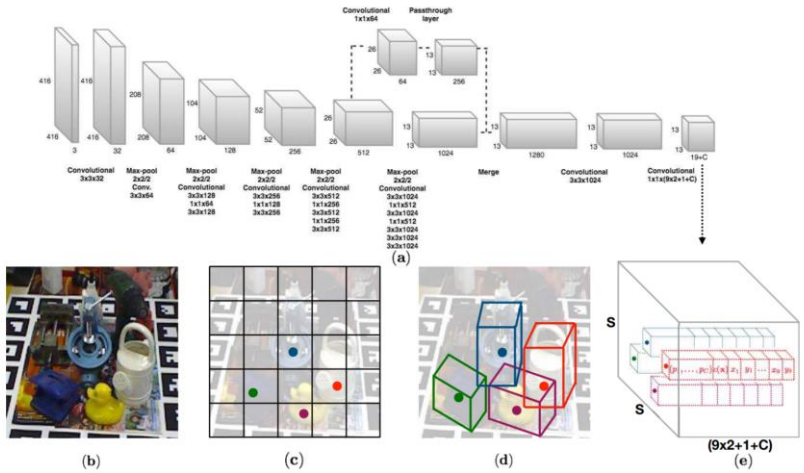


bb8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth - 2017 ICCV

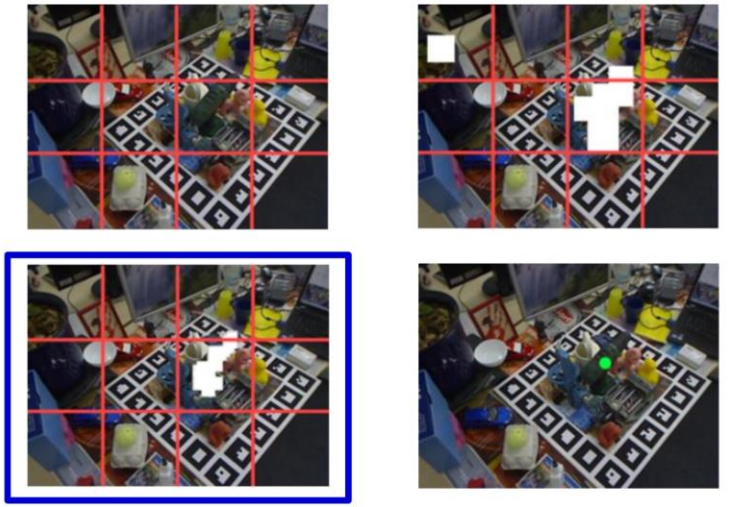
However, generalization remains as an issue, as it is unclear that such end-to-end methods learn sufficient feature representations for pose estimation.

1. Introduction

Deep Learning based method



Real-Time Seamless Single Shot 6D Object Pose Prediction -CVPR 2018



bb8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth - 2017 ICCV

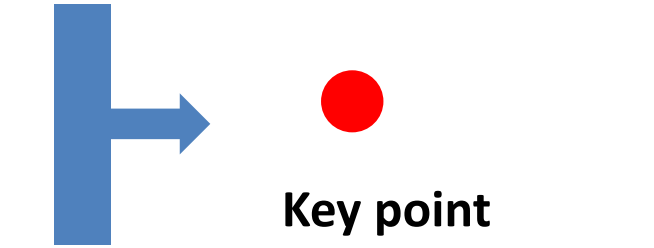
CNN이 unseen keypoints에 대해서 memorizing similar patterns해서 predict할 수 있지만 generalization 하기 어려움

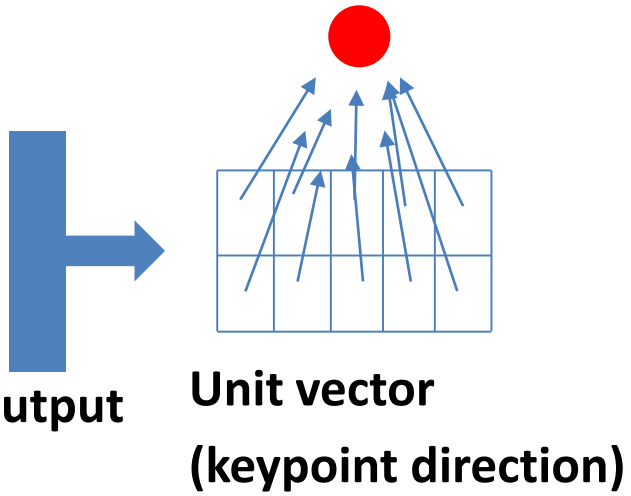
우리는 이런 문제들이 pixel wise 또는 patch wise로 접근한다면 해결가능하다 생각함.

→ 새로운 Network인 PVNet 제안.

1. Introduction

Direct Coordinates Regression ⇒ **Vector field representation**





2가지 이점.

1) Unseen object의 keypoints 를 잘 찾음.
2) Rich한 information을 제공해 PnP 알고리즘 적용 시 좋음.

1. Introduction

Contribution

- We propose a novel framework for 6d pose estimation using a pixel-wise voting network(PVNet), which learns a vector-field representation for robust 2D keypoint localization and naturally deals with occlusion and truncation

- We propose to utilize an uncertainty-driven PnP algorithm to account for uncertainties in 2D keypoint localizations, based on the dense predictions from PVNet

- We demonstrate significant performance improvements of our approach compared to the state of the art on benchmark datasets (ADD:86.3% vs 79% and 40.8% vs. 30.4% on LINEMOD and OCCLUSION,respectively). We also create a new dataset for evaluation on truncated objects

2. Proposed Method

6D Pose 를 찾자.

2stage pipeline

1) Find 2D object keypoints
2) PnP

기존과 다르게 2D keypoints를 새로운 representation을 통해서 구한다.

2. Proposed Method – 1 Voting-based keypoint localization.

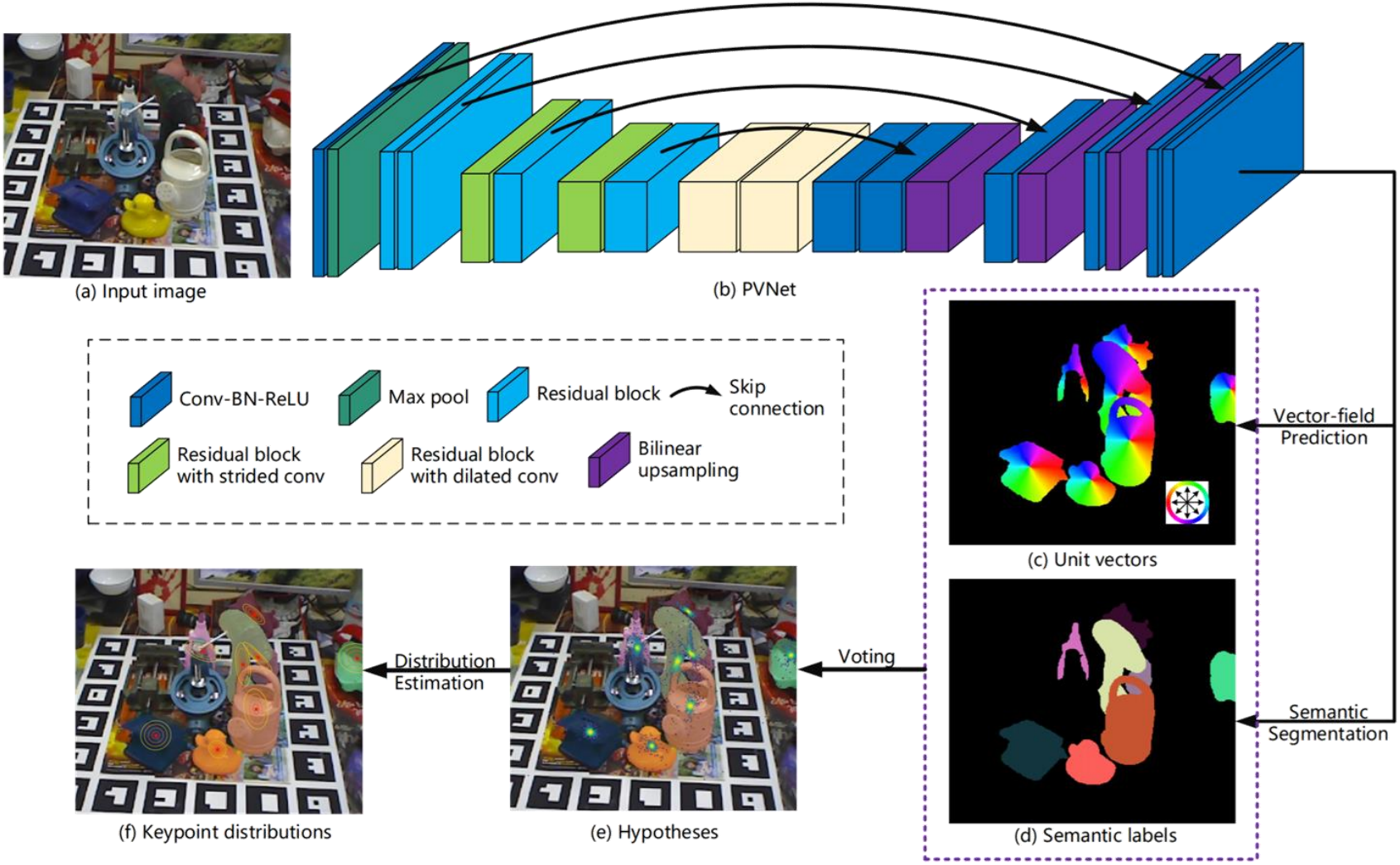
1) Input 으로 RGB image가 주어질 때

2) PVNET은 pixel-wise object labels와 모든 픽셀이 각 keypoint로 향한 direction representation으로 unit vectors를 내뱉음.

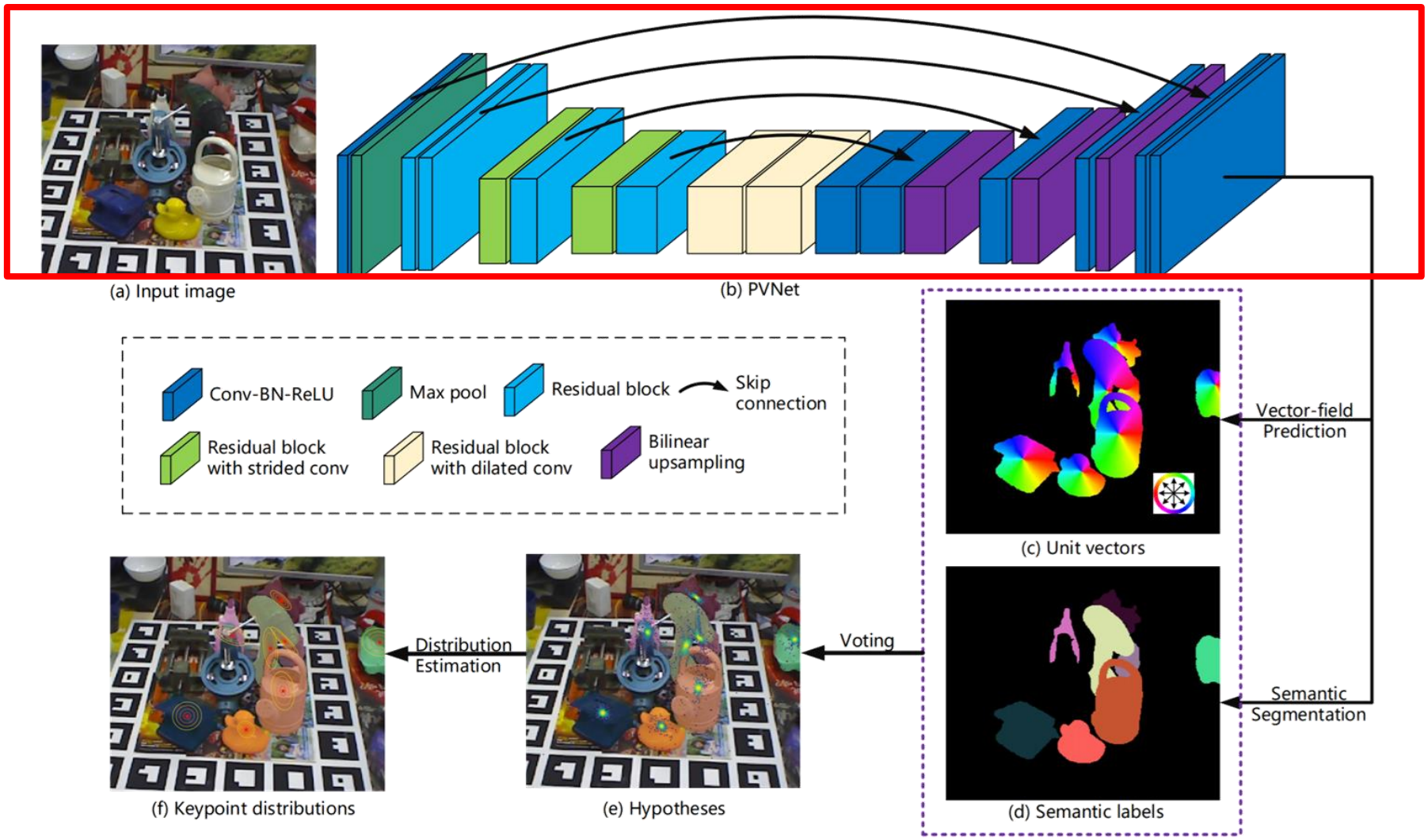
3) 2D keypoint location에 대한 hypothesis 생성하고 ransac 기반 voting을 통해 confidence score를 만듦.

4) Hypotheses에 기반해 keypoints의 mean, covariance of spatial probability distribution을 구함.

2. Proposed Method – 1 Voting-based keypoint localization.

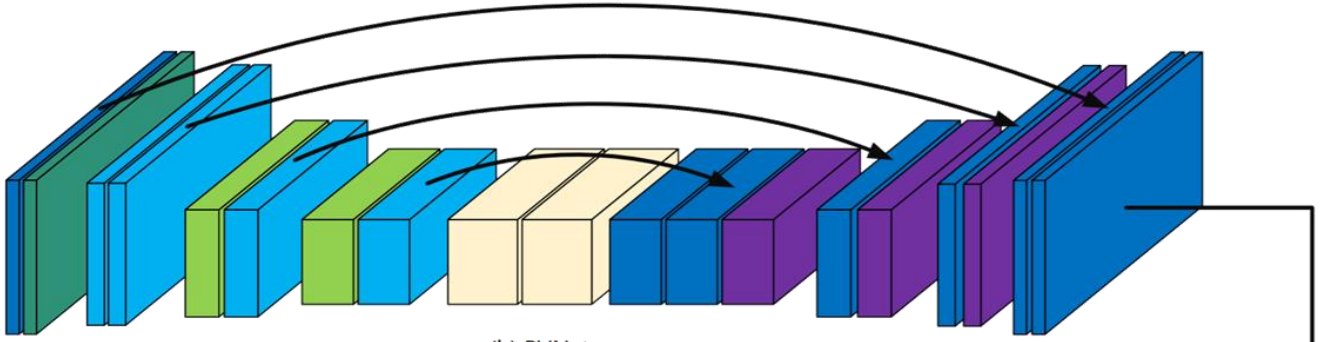


2. Proposed Method – 1 Voting-based keypoint localization.

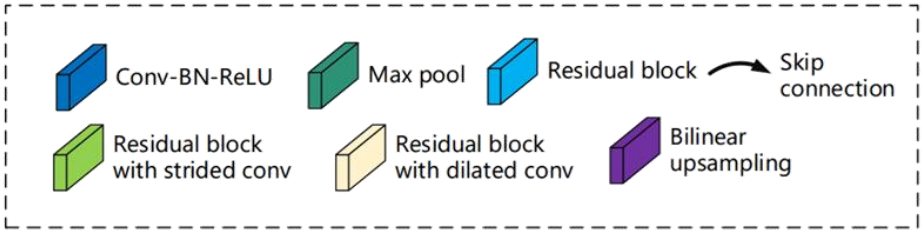


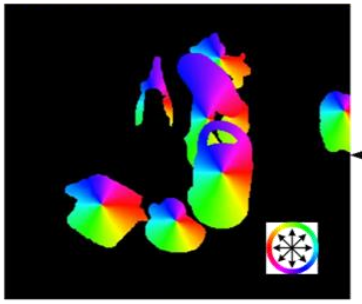
2. Proposed Method – 1 Voting-based keypoint localization.

(a) Input image



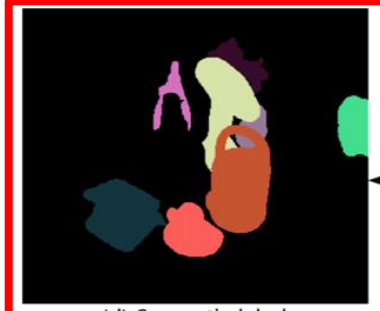
(b) PVNet





(c) Unit vectors

Vector-field Prediction



(d) Semantic labels

Semantic Segmentation

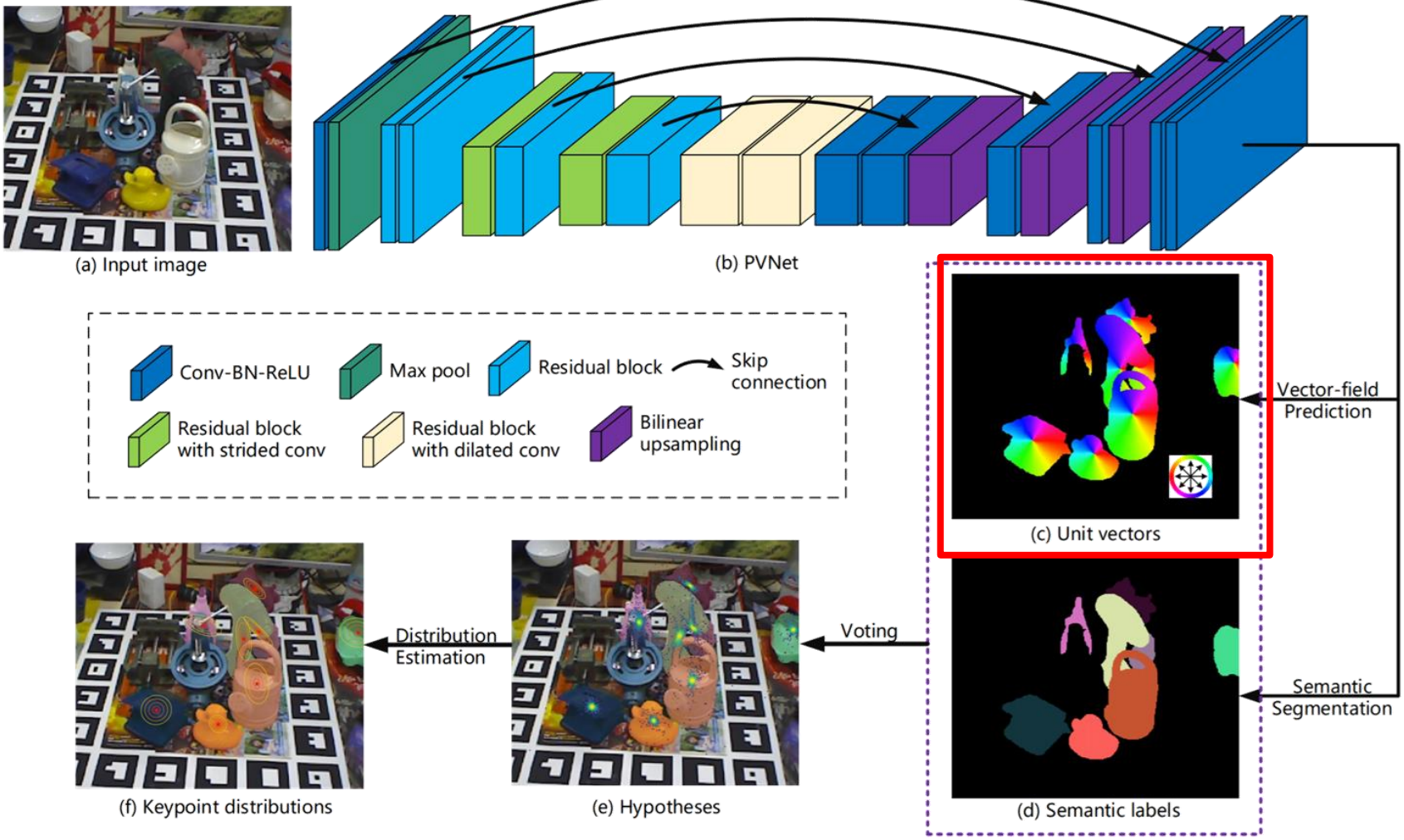
Voting

(e) Hypotheses

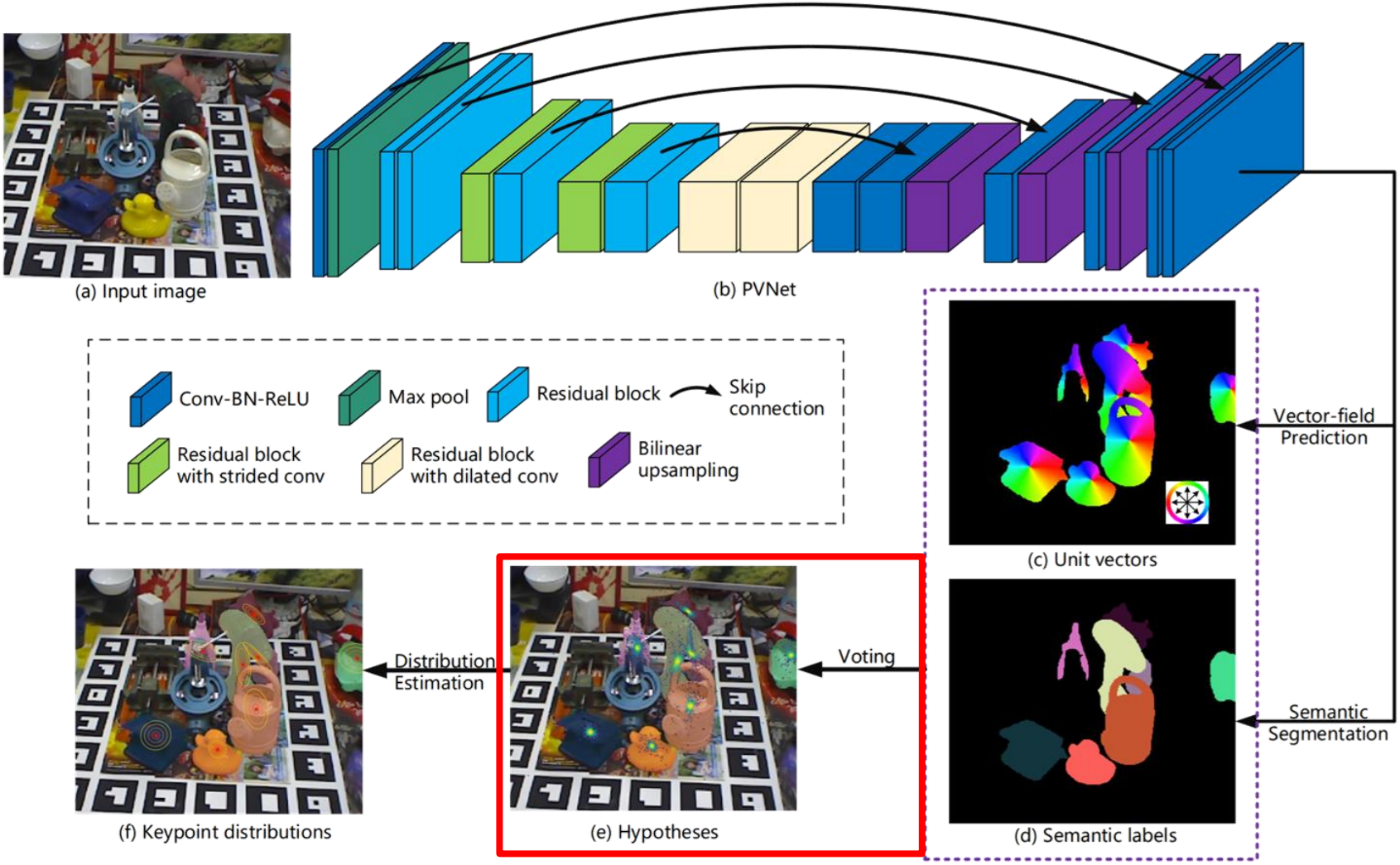
Distribution Estimation

(f) Keypoint distributions

2. Proposed Method – 1 Voting-based keypoint localization.



2. Proposed Method – 1 Voting-based keypoint localization.



2. Proposed Method – 1 Voting-based keypoint localization.

$$\mathbf{v}_k(\mathbf{p}) = \frac{\mathbf{x}_k - \mathbf{p}}{\|\mathbf{x}_k - \mathbf{p}\|_2}. \qquad (1)$$

Pixel p는 semantic label의 output.

Vector는 각 pixel p로부터 2d keypoint x_k까지의 direction으로 나타냄.

수식은 아래와 같이 정의된다.

2. Proposed Method – 1 Voting-based keypoint localization.

Semantic labels과 unit vectors가 주어지면

우리는 keypoint 가설을 ransac based voting scheme으로 만들음.

1) Semantic labels을 이용해서 target pixel을 찾는다.
2) 랜덤으로 2개의 pixel을 고르고 그것들의 intersection 를 xk에 대한 가설로 만든다.
3) 이 과정은 N번 반복되고 이 가설은 가능한 keypoints의 위치 N개의 가설 세트를 만들음.
4) 최종적으로 모든 물체의 pixel들은 그들의 가설에 vote함.

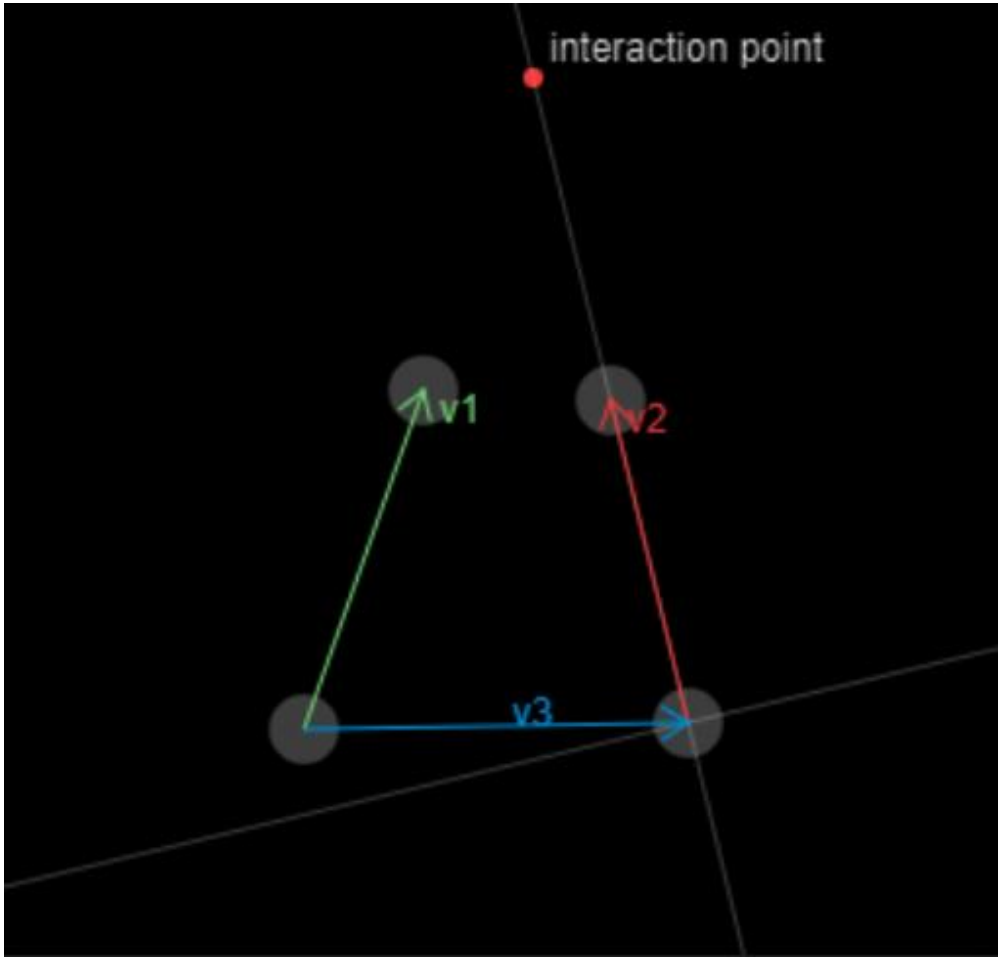
2. Proposed Method – 1 Voting-based keypoint localization.

Semantic labels과 unit vectors가 주어지면

우리는 keypoint 가설을 ransac based voting scheme으로 만들음.

1) Semantic labels을 이용해서 target pixel을 찾는다.
2) 랜덤으로 2개의 pixel을 고르고 그것들의 intersection 를 xk에 대한 가설로 만든다.
3) 이 과정은 N번 반복되고 이 가설은 가능한 keypoints의 위치 N개의 가설 세트를 만들음.
4) 최종적으로 모든 물체의 pixel들은 그들의 가설에 vote함.

2. Proposed Method – 1 Voting-based keypoint localization.



2. Proposed Method – 1 Voting-based keypoint localization.

Semantic labels과 unit vectors가 주어지면

우리는 keypoint 가설을 ransac based voting scheme으로 만들음.

1) Semantic labels을 이용해서 target pixel을 찾는다.
2) 랜덤으로 2개의 pixel을 고르고 그것들의 intersection 를 xk에 대한 가설로 만든다.
3) 이 과정은 N번 반복되고 이 가설은 가능한 keypoints의 위치 N개의 가설 세트를 만들음.
4) 최종적으로 모든 물체의 pixel들은 그들의 가설에 vote함.

2. Proposed Method – 1 Voting-based keypoint localization.

hypotheses 대한 voting score

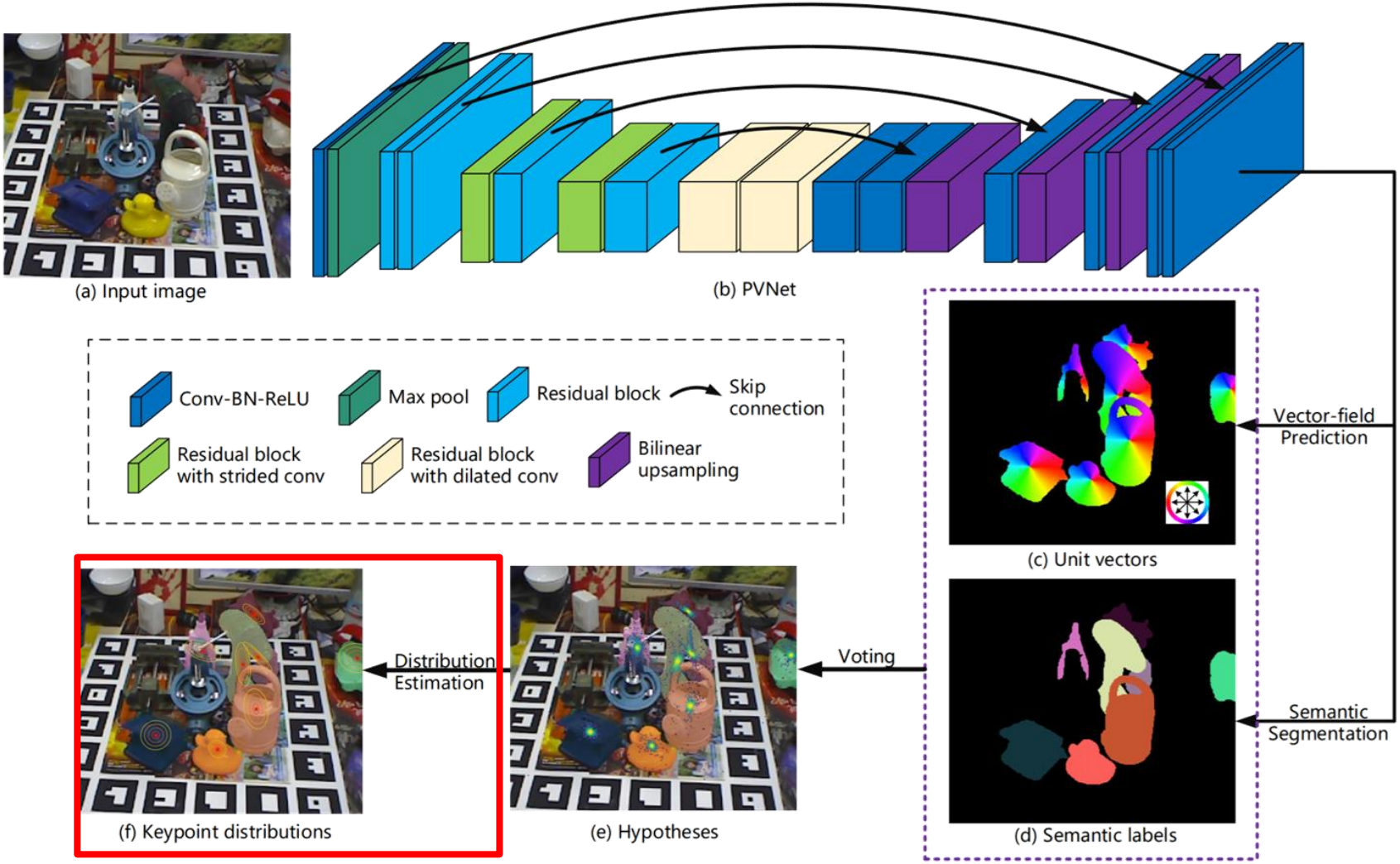
$$w_{k,i} = \sum_{\mathbf{p} \in O} \mathbb{I}\left(\frac{(\mathbf{h}_{k,i} - \mathbf{p})^T}{\|\mathbf{h}_{k,i} - \mathbf{p}\|_2} \mathbf{v}_k(\mathbf{p}) \geq \theta \right), \qquad (2)$$

여기서 $\mathbb{I}$ 는 indicator function

θ는 threshold 0.99.

직관적으로 투표 점수가 높을수록 예측된 방향과 일치하므로 가설이 더 확실해진다.

2. Proposed Method – 1 Voting-based keypoint localization.



2. Proposed Method – 1 Voting-based keypoint localization.

가설의 결과는 spatial probability distribution of a keypoints in the image로 나타남.

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} w_{k,i}\mathbf{h}_{k,i}}{\sum_{i=1}^{N} w_{k,i}}, \tag{3}$$

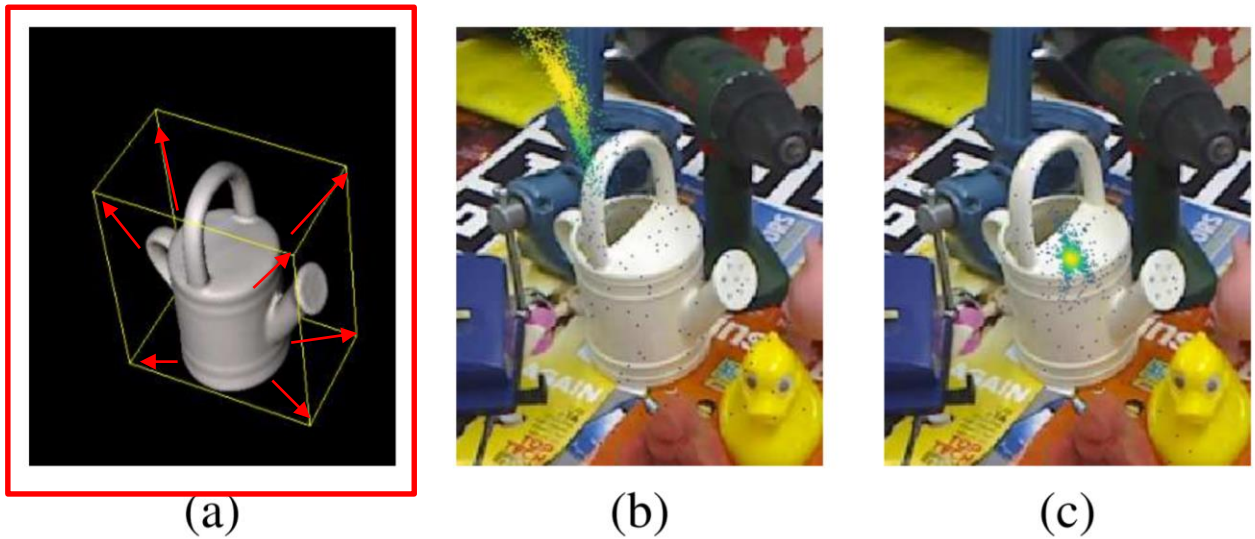
$$\boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{N} w_{k,i}(\mathbf{h}_{k,i} - \boldsymbol{\mu}_k)(\mathbf{h}_{k,i} - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} w_{k,i}}, \tag{4}$$

나중에 uncertainty driven PnP described 함.

2. Proposed Method – 1 Voting-based keypoint localization.

Keypoint selection

대부분의 keypoint based method 들은 keypoints를 3D bounding box 의 corner점을 설정.



(a) (b) (c)

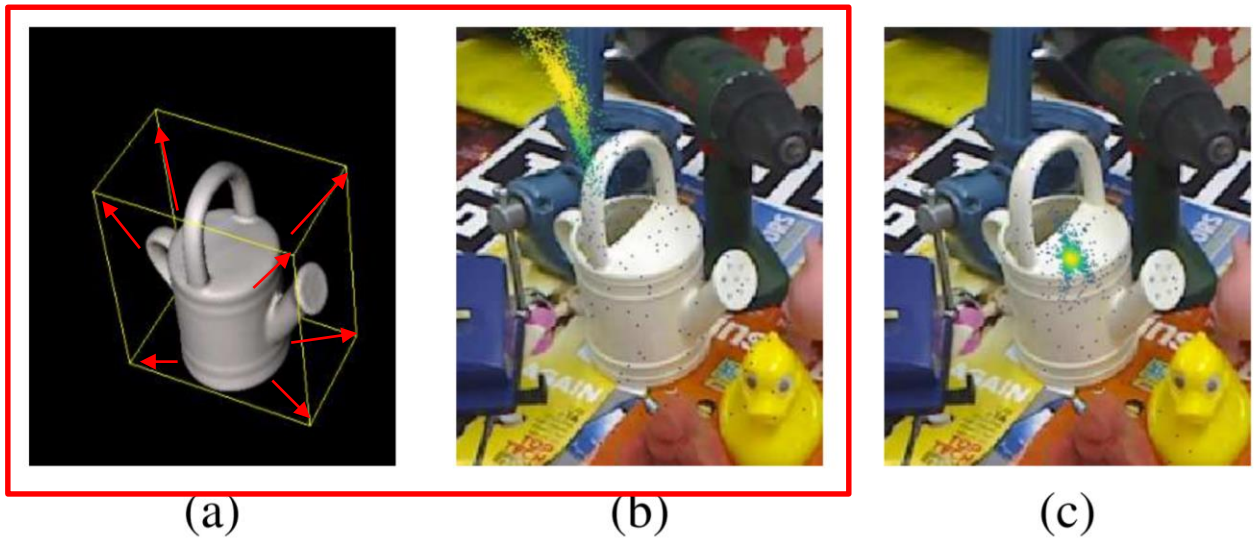
PVNet의 output은 물체위에서 direction으로 나타나지만 Corner는 물체로부터 거리가 멀다.

Localization Error 발생.

2. Proposed Method – 1 Voting-based keypoint localization.

Keypoint selection

대부분의 keypoint based method 들은 keypoints를 3D bounding box 의 corner점을 설정.



(a) (b) (c)

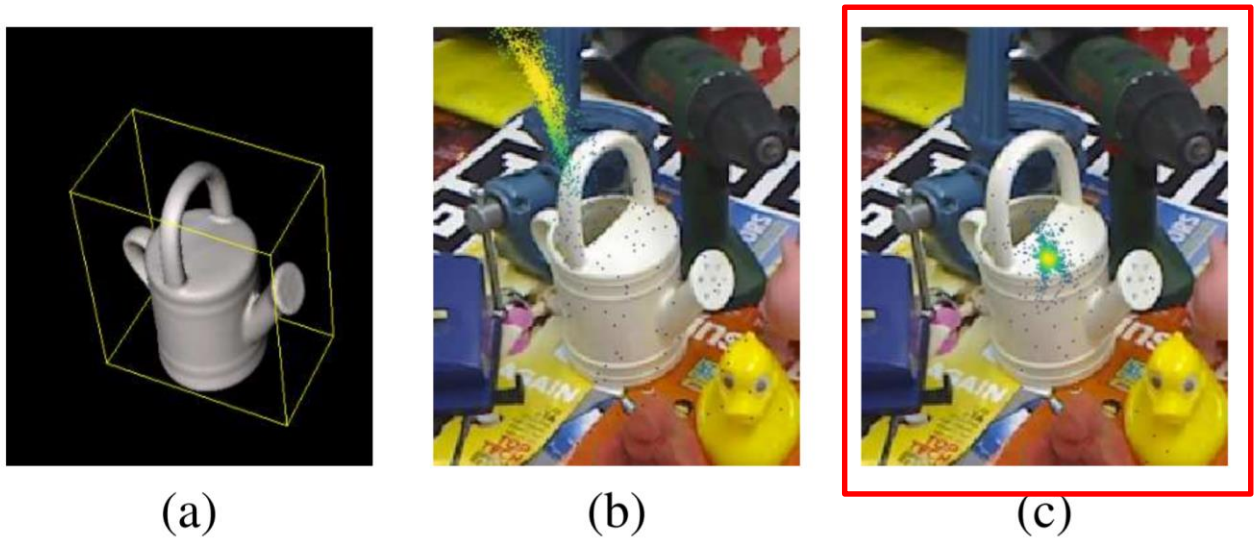
PVNet의 output은 물체위에서 direction으로 나타나지만 Corner는 물체로부터 거리가 멀다.

Localization Error 발생.

2. Proposed Method – 1 Voting-based keypoint localization.

Keypoint selection

대부분의 keypoint based method 들은 keypoints를 3D bounding box 의 corner점을 설정.



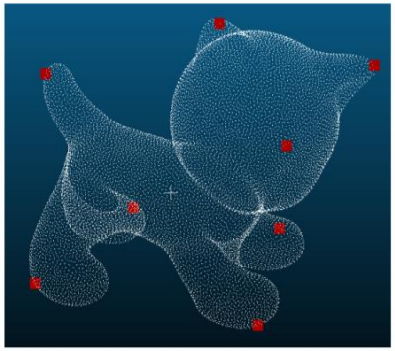
(a) (b) (c)

keypoint는 object surface 위에 있어야 localization variance가 작아짐.
따라서 FPS 알고리즘(farthest point sampling)을 통해서 K개의 keypoint를 설정.

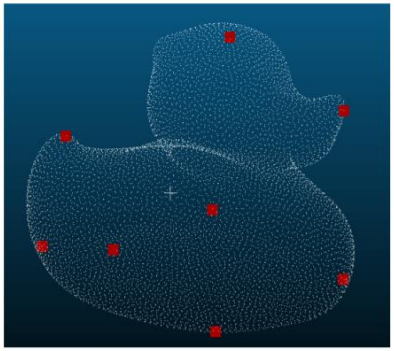
2. Proposed Method – 1 Voting-based keypoint localization.

Keypoint selection

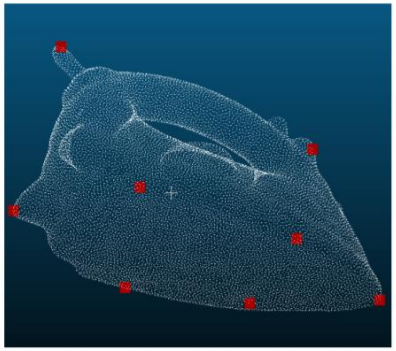
FPS 알고리즘을 이용해서 선택된 8개의 Keypoint



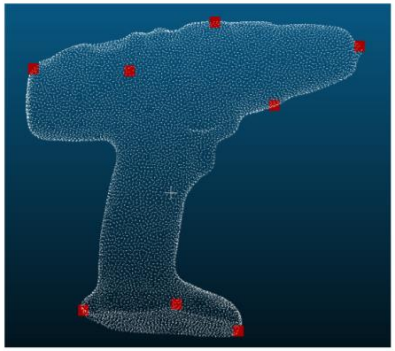
cat



duck



iron



driller

2. Proposed Method – 2 Uncertainty-driven PnP

2D keypoint location이 주어졌을 때 EPnP을 이용해서 해결함.

그러나 대부분의 방법들은 서로 다른 keypoints는 다른 confidences와 uncertainty patterns을 다룬다는 걸 간과함 이것들은 PnP문제를 풀때 고려되어야하는 문제임.

우리의 voting based method 는 각각 keypoint에 대해서 spatial probability distribution를 고려해서 풀음.

Mean 와 Covariance가 주어졌을 때 6D Pose를 Mahalanobis distance를 최소화시키면서 구할 수 있음.

2. Proposed Method – 2 Uncertainty-driven PnP

most of them ignore the fact that different keypoints may have different confidences and uncertainty patterns, which should be considered when solving the PnP problem.

$$\underset{R,\mathbf{t}}{\text{minimize}} \sum_{k=1}^{K} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\tilde{\mathbf{x}}_k - \boldsymbol{\mu}_k), \qquad (5)$$

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

X_k : 3D keypoint (model) **$\tilde{x}_k$: X_k의 2D projection된 값.**

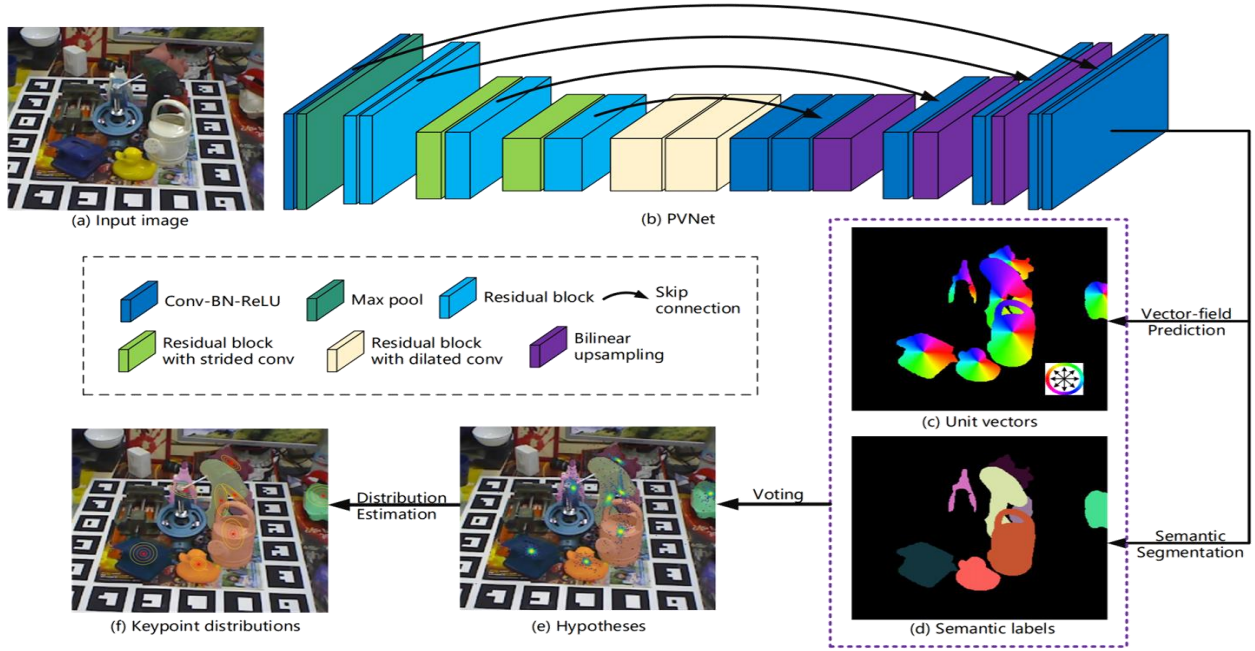
π: projection function

covariance matrics는 가장 작은 trace를 갖음.

Levenberg Marquardt algorithm 으로 풀음.(?)

directly minimize the reprojection errors.

3. Implementation detail



1) Resnet18에서 이미지 output size를 H/8 x W/8로 고정

2) Receptive fields를 바꾸지않고 연속된 conv대신에 안정적으로 dilated conv로 바꿈.

3) Fully connected layers 대신에 conv1x1로 바꿈.

4) 그 뒤에 Upsampling을 계속해서 HxW사이즈로 맞춰줌. 그뒤 1x1 conv final conv로 각각의 값을 만듦.

3. Implementation detail – 1. training strategy

$$\ell(\mathbf{w}) = \sum_{k=1}^{K} \sum_{\mathbf{p} \in O} \ell_1(\Delta \mathbf{v}_k(\mathbf{p}; \mathbf{w})|_x) + \ell_1(\Delta \mathbf{v}_k(\mathbf{p}; \mathbf{w})|_y),$$

$$\Delta \mathbf{v}_k(\mathbf{p}; \mathbf{w}) = \tilde{\mathbf{v}}_k(\mathbf{p}; \mathbf{w}) - \mathbf{v}_k(\mathbf{p}), \qquad (6)$$

Smooth l1 loss를 사용해서 unit vector를 learning 함.
여기서 w는 PVNet parameters $\tilde{v}_k$는 predicted vector

Overfitting 막기 위해 2만개 synthetic data동원

4. Experiments – 1. Datasets

LINEMOD [15] is a standard benchmark for 6D object pose estimation. This dataset exhibits many challenges for pose estimation: cluttered scenes, texture-less objects, and lighting condition variations.

Occlusion LINEMOD [2] was created by additionally annotating a subset of the LINEMOD images. Each image contains multiple annotated objects, and these objects are heavily occluded.

Truncation LINEMOD To fully evaluate our method on truncated objects, we create this dataset by randomly cropping images in the LINEMOD dataset. After cropping, only 40% to 60% of the area of a target object remains in the image. Some examples are shown in Figure 6. On LINEMOD, we use exactly the same training-test split as in [36], while the Occlusion LINEMOD and Truncation LINEMOD are used for testing only.

YCB-Video [40] is a recently proposed dataset. The images are collected from the YCB object set [5]. This dataset is challenging due to the varying lighting conditions, significant image noise and occlusions.

4. Experiments – 2. Evaluation metrics

We evaluate our method using two standard metrics:

1) 2D projection metric

2) average 3D distance of model points (ADD) metric

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].

4. Experiments – 3. Ablation studies

Vector type에 대한 효과를 평가함.

methods	Tekin [36]	BBox 8	Offset 8	FPS 4	FPS 8	FPS 12	FPS 8 + Un
ape	2.48	6.50	12.99	5.31	**17.44**	15.1	15.81
can	17.48	65.04	**69.10**	18.81	63.21	64.87	63.30
cat	0.67	15.00	**26.12**	16.01	17.35	16.68	16.68
duck	1.14	15.95	14.55	13.85	**26.12**	24.89	25.24
driller	7.66	55.60	65.24	12.19	62.19	64.17	**65.65**
eggbox	-	35.23	41.62	36.77	44.96	41.53	**50.17**
glue	10.08	42.64	**55.48**	24.81	47.32	51.94	49.62
holepuncher	5.45	35.06	32.22	15.98	39.50	**40.16**	39.67
average	6.42	33.88	39.66	17.96	39.76	39.92	**40.77**

비교해보면 bbox8로 했을 때보다 surface(fps) + center로 한게 좋음.
개수 늘려서 해봤는데 12일 때가 제일 정확하긴 함.

Uncertainties in pnp를 하기 위해서 EPnP를 사용해서 평가해보니까 FPS8 + Un가 제일 좋았다.

4. Experiments – 4. Comparison with the state-of-the-art methods

RGB images를 input으로 받는 6d pose estimation방법들과 비교를 해봄.

Performance on the LINEMOD dataset.

	w/o refinement			w/ refinement
methods	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

4. Experiments – 4. Comparison with the state-of-the-art methods

	w/o refinement				w/ refinement	
methods	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

ADD 값을 가지고 비교해봤는데 제일 잘함.

4. Experiments – 4. Comparison with the state-of-the-art methods

Robustness to occlusion(Occluded Linemod)

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**

2D Projection 지수 제일 잘함.

4. Experiments – 4. Comparison with the state-of-the-art methods

Robustness to occlusion

4. Experiments – 4. Comparison with the state-of-the-art methods

Robustness to occlusion(Occluded Linemod)

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**

4. Experiments – 4. Comparison with the state-of-the-art methods

Robustness to truncation

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

4. Experiments – 4. Comparison with the state-of-the-art methods

Robustness to truncation

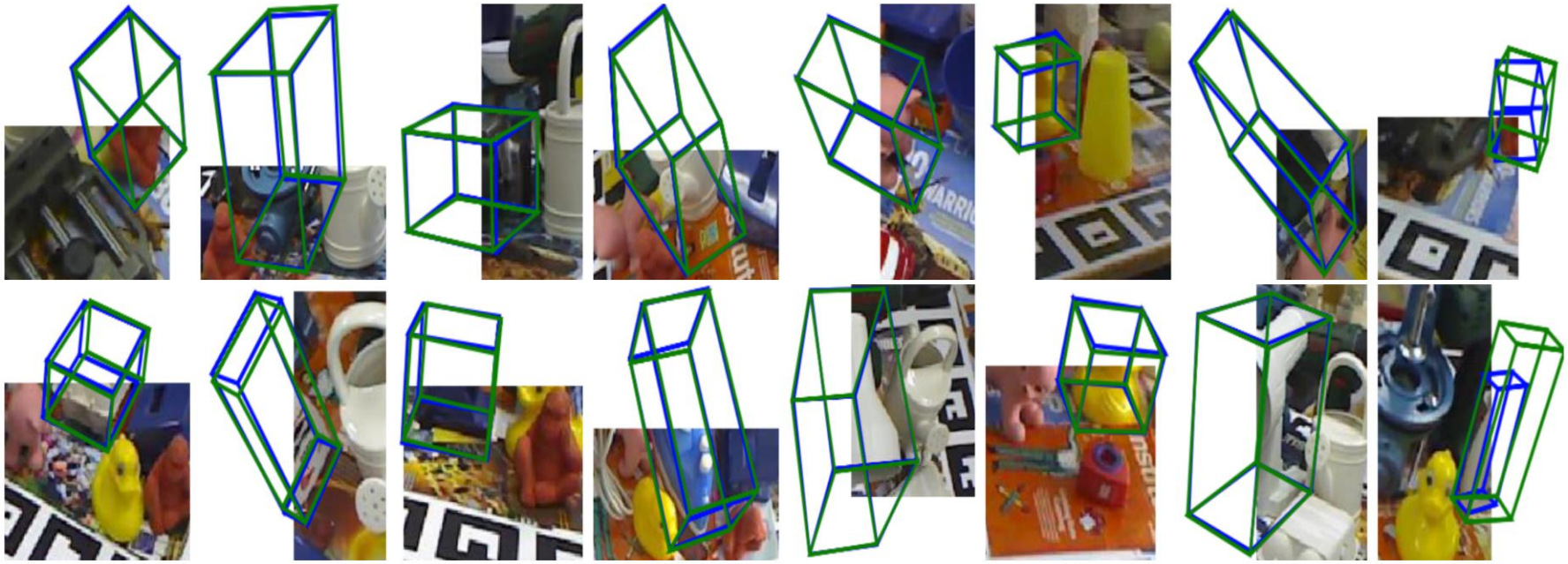
	w/o refinement			w/ refinement
methods	BB8 [30]	Tekin [36]	OURS	BB8 [30]
ape	95.3	92.10	**99.23**	96.6
benchwise	80.0	95.06	**99.81**	90.1
cam	80.9	93.24	**99.21**	86.0
can	84.1	97.44	**99.90**	91.2
cat	97.0	97.41	**99.30**	98.8
driller	74.1	79.41	**96.92**	80.9
duck	81.2	94.65	**98.02**	92.2
eggbox	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

	w/o refinement				w/ refinement	
methods	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

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

4. Experiments – 4. Comparison with the state-of-the-art methods

Robustness to truncation



4. Experiments – 4. Comparison with the state-of-the-art methods

Performance on the YCB_Video dataset.

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

4. Experiments – 5. Running time

480 X 640 input image

25 fps (i7 3.7GHz CPU, 1080ti)

5. Conclusion

1. We introduced a novel framework for 6DoF object pose estimation, which consists of the pixel-wise voting network(PVNet) for keypoint localization and the uncertainty-driven PnP for final pose estimation

2. We showed that predicting the vector fields followed by RANSAC-based voting for keypoint localization gained a superior performance than direct regression of keypoint coordinates, especially for occluded or truncated object

3. We showed that considering the uncertainties of predicted keypoint locations in solving the PnP problem further improved pose estimation.

4. We reported the state-of-the-art performances on all three widely-used benchmark datasets and demonstrated the robustness of the proposed approach on a new dataset of truncated objects

End

2. Proposed Method – 1 Voting-based keypoint localization.

Direct로 image patch로부터 keypoint location을 regression하는 방법과 대조적으로 pixel wise predictiong task는 network가 조금 더 object의 local feature를 강화시키고, background cutter의 효과를 완화시킨다.

이 방법의 다른 이점은 keypoint의 표현능력인데 occlude이거나 outside the image임.

비록 keypoint가 invisibl하더라도 정확하게 located 함.