Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation

Kiru Park et al. (Vision for Robotics Laboratory, Automation and Control Institute, TU Wien, Austria)

ICCV 2019

2020.08.31

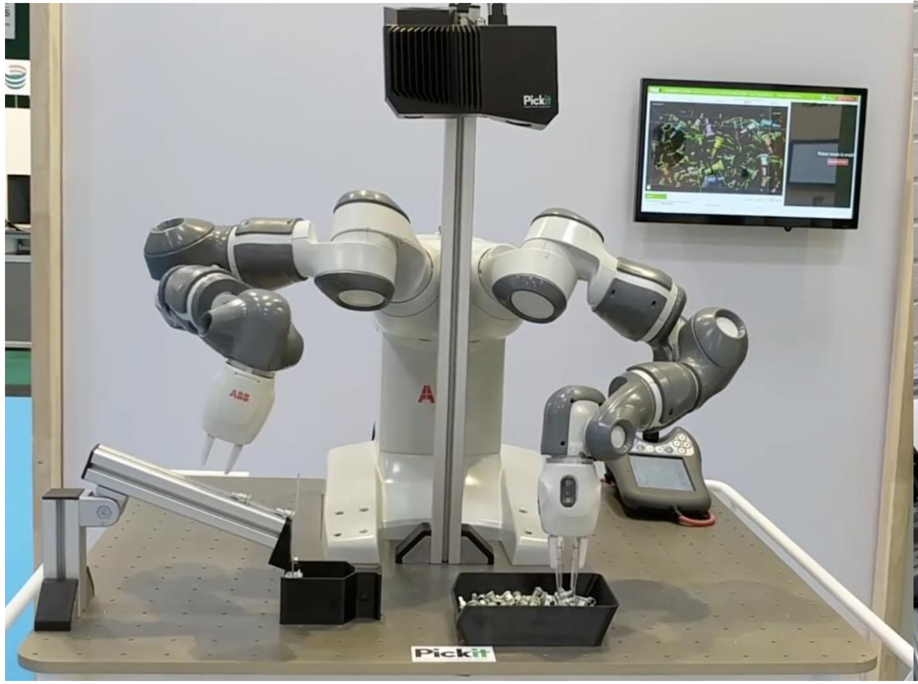
Hanyang univ. AILAB 정지은

Index

1. Introduction

Introduction

- **Bin picking**



https://m.blog.naver.com/PostList.nhn?blogId=rubenautomation

Introduction

- **6D Pose estimation task**

Introduction

- **Challenges**

1) 3D models **with high-quality textures** are required

Special device / manual adjustment

BigBIRD Object Scanning Rig*

Sufficient for
Synthetic rendering

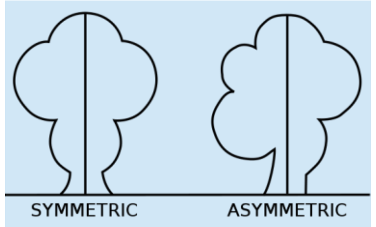
Real environment (e.g., robots)
Noisy odometry, varied lighting, limited viewpoints

3D reconstruction using a mobile robot

Not sufficient!

2) Occlusion

3) Symmetric objects

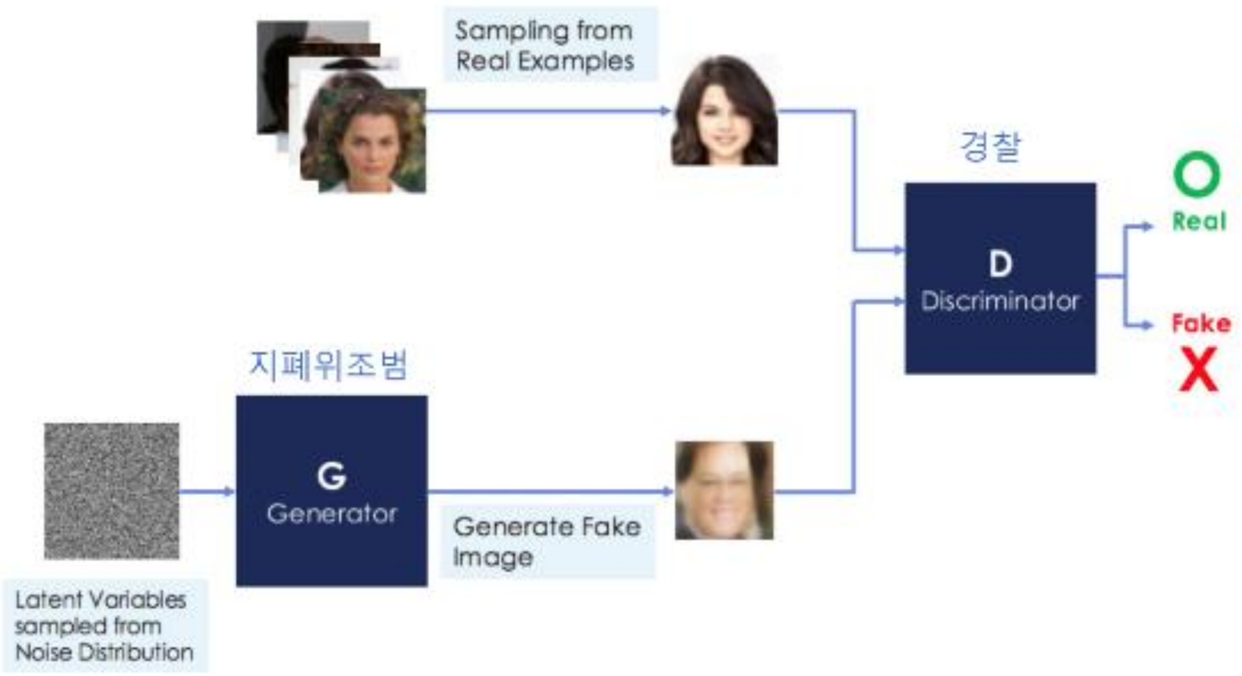


*Calli et al., IJRR (2017)

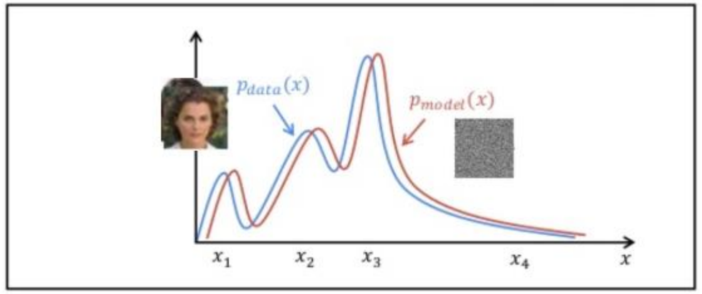
2. Related work

Related work : GAN (Generative Adversarial Network)

- **두개의 모델** (Generator & Discriminator)을 **적대적으로 경쟁**시키면서 서로의 성능을 발전시키는 방식으로 실제 이미지와 비슷한 이미지를 만드는 **생성모델**



The purpose of the GAN



최적화를 통해 서로 다른 확률분포 간의 차이 줄이기

Related work : GAN loss

Discriminator loss function

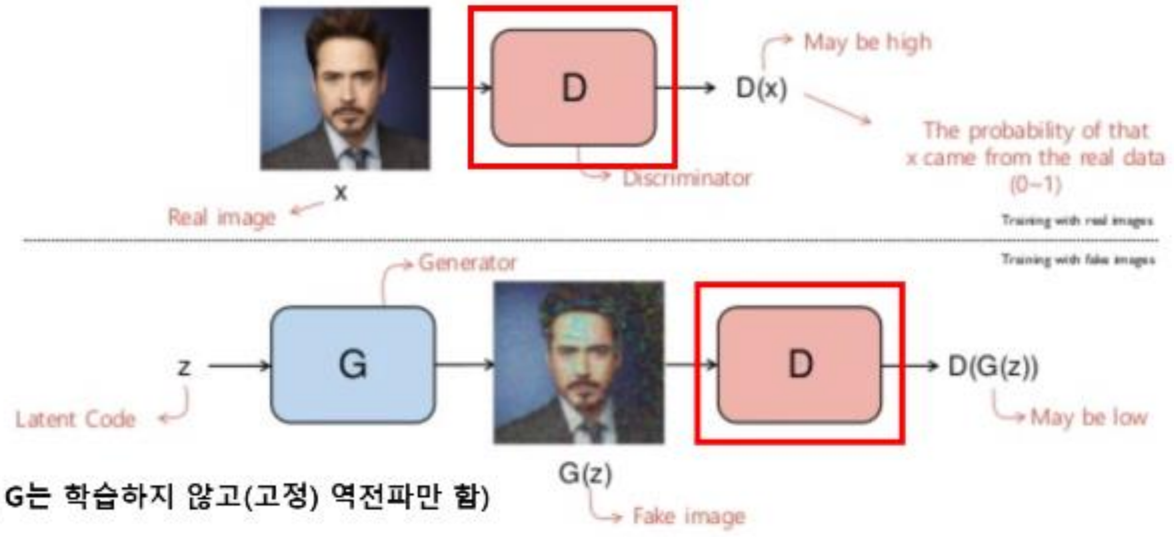
Sample x from real data distribution

Sample z from Gaussian distribution

$$\min_{G} \boxed{\max_{D} V(D,G)} = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

D(x) = 1일때 Maximum

D(G(z)) = 0일때 Maximum



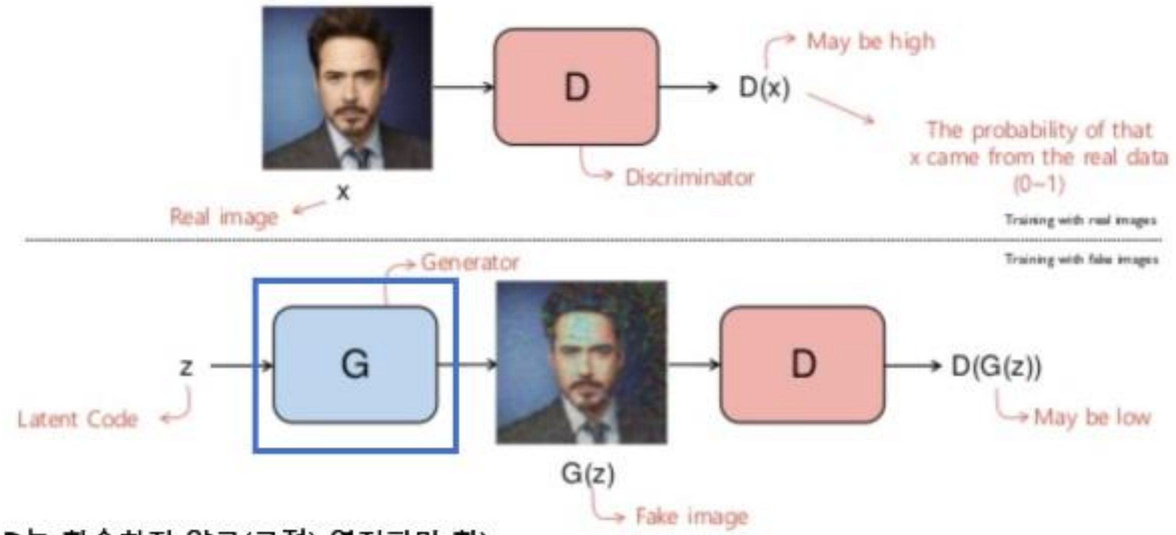
(단, D 학습시에는 G는 학습하지 않고(고정) 역전파만 함)

Generator loss function

$$\boxed{\min_G}\max_D V(D,G) = \cancel{E_{x\sim p_{data}(x)}[\log D(x)]} + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

G is independent

D(G(z)) = 1일때 Minimum



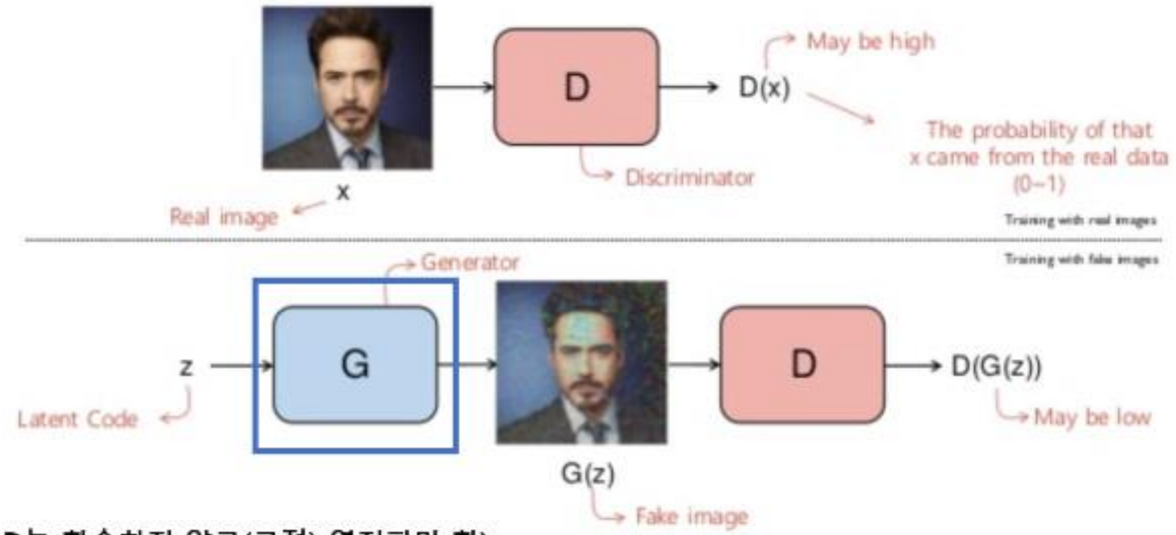
(단, G 학습시에는 D는 학습하지 않고(고정) 역전파만 함)

Generator loss function

$$\boxed{\min_G}\max_D V(D,G) = \cancel{E_{x\sim p_{data}(x)}[\log D(x)]} + E_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

G is independent

D(G(z)) = 1일때 Minimum

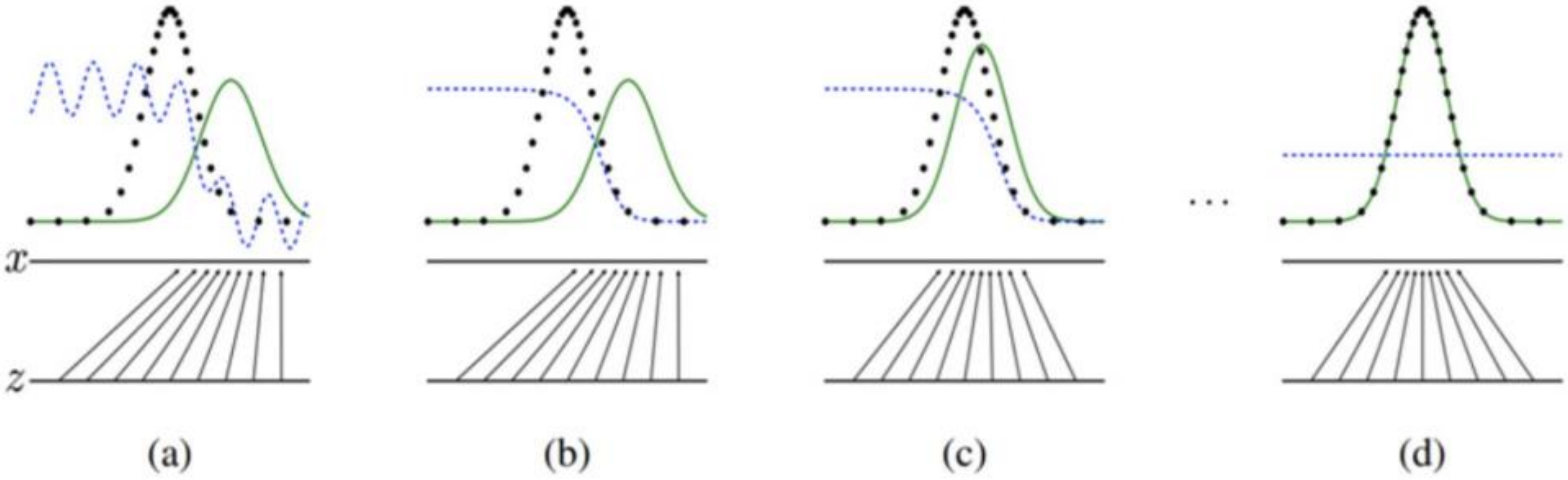


(단, G 학습시에는 D는 학습하지 않고(고정) 역전파만 함)

1 epoch 완료

Related work : GAN (Generative Adversarial Network)

The purpose of the GAN



※ **검은 점선**: 원 데이터의 확률분포, **녹색 점선**: GAN이 만들어 내는 확률분포, **파란 점선**: 분류자의 확률분포

위로 뻗은 화살표 : x = G(z)의 mapping

<GAN에서 학습을 통해 확률분포를 맞추어 나가는 과정>

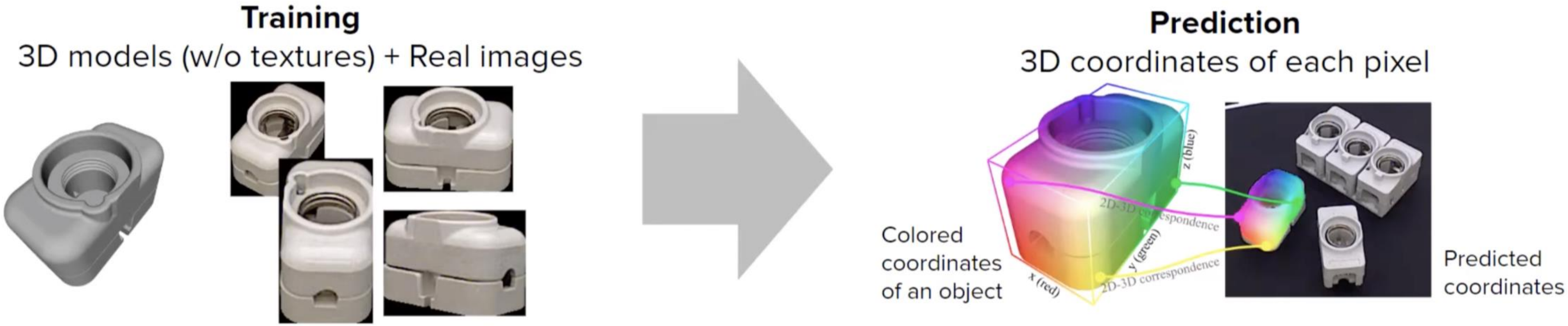
3. Method

Method : Pix2Pose

- **기본 아이디어 :**

  GAN의 image-to-image translation 방식처럼 물체의 가려진 부분을 복원하면서

  이미지 -> 좌표값 으로 translation



- Annotation = 6D 포즈 좌표값
- Ground truth = CAD 모델에 6D 좌표값을 대응시킨 3D Color 이미지

- 3D 좌표값이 색상으로 렌더링된 2D 이미지가 됨

Method : Pix2Pose

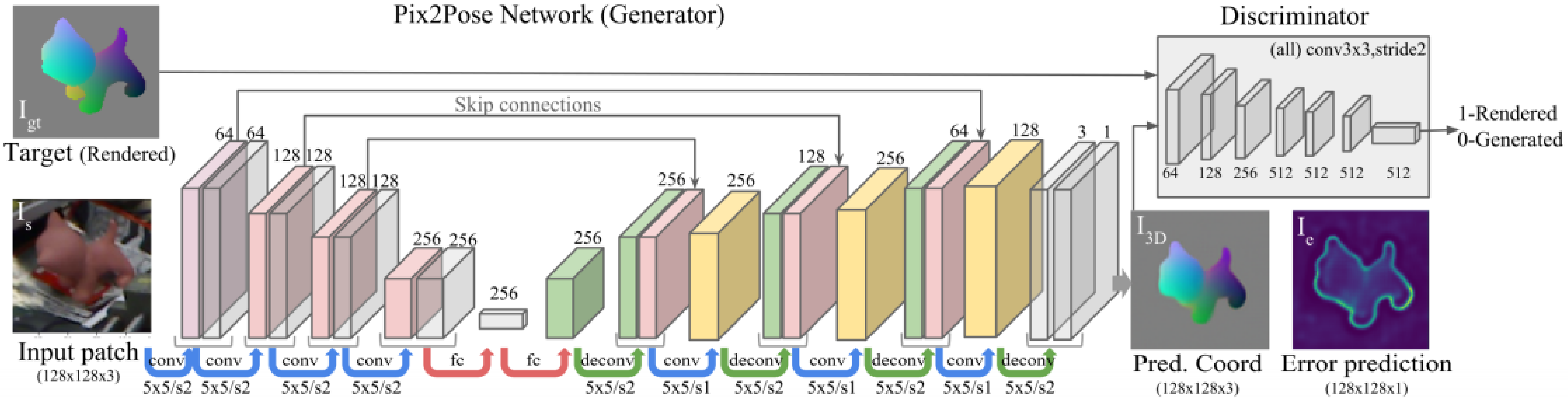


Figure 2. An overview of the architecture of Pix2Pose and the training pipeline.

- **Input** : **A cropped image** Is using a bounding box of a detected object class
- **Output** : **Normalized 3D coordinates** of each pixel I3D in the object coordinate and estimated errors

Network training : Loss

$$G^* = \arg\min_G \max_D \mathcal{L}_{\mathrm{GAN}}(G, D) + \lambda_1 \mathcal{L}_{\mathrm{3D}}(G) + \lambda_2 \mathcal{L}_{\mathrm{e}}(G)$$

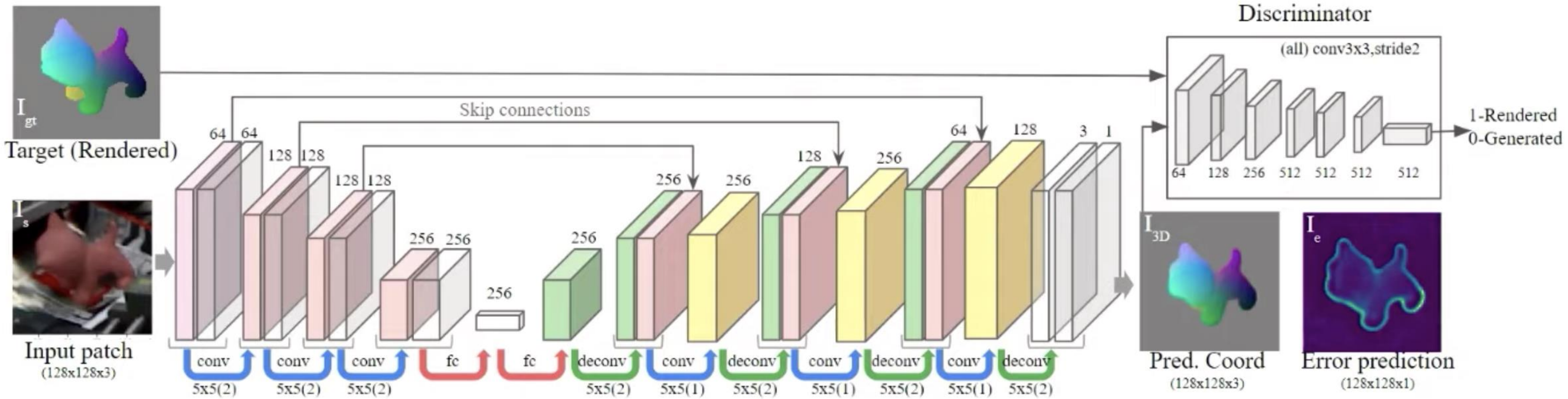
- 전체 손실함수는 3가지로 구성되어 있음

(1) GAN loss : Occlusion 문제 해결

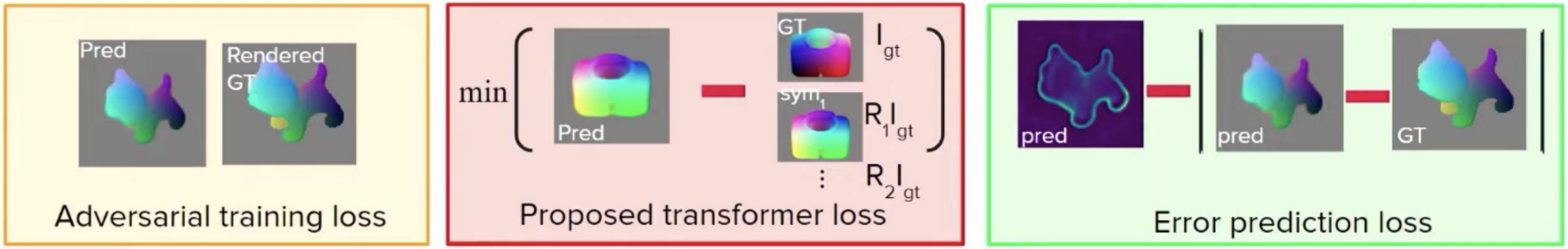
(2) Transformer loss : Symmetric objects 문제 해결

(3) Error loss : <u>외곽부분</u>을 보완적으로 보정해서 정밀한 이미지 생성

Network training : Loss



Training objective: $G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda_1 \mathcal{L}_{3D}(G) + \lambda_2 \mathcal{L}_e(G)$



For occlusion objects For symmetric objects For Inliers and outliers

Network training : Loss

(1) **GAN loss**: 이미지의 가려진 부분을 복원해서 알맹이 만들기 (Occlusion 문제 해결)

$$\mathcal{L}_{\mathrm{GAN}} = \log D(I_{gt}) + \log(1 - D(G(I_{\mathrm{src}})))$$

- G : 가짜 이미지 생성
- D : 가짜 생성 이미지와 GT 이미지 판별
- I_src : source img(input)
- I_gt : GT

Network training : Loss

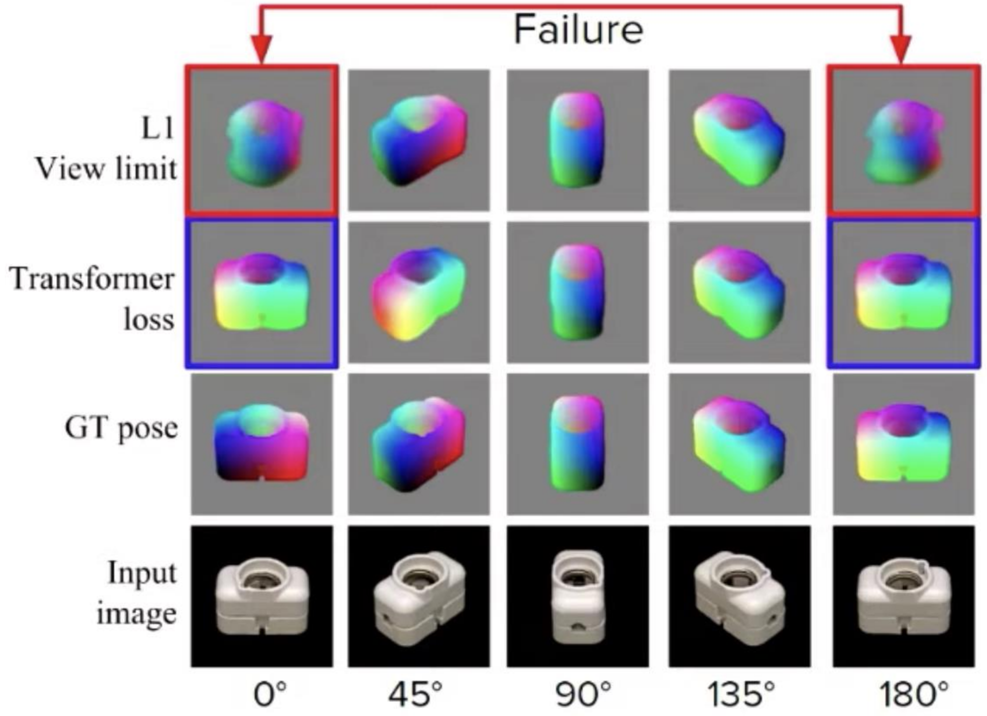
(2) **Standard loss**: 기본 L1 로스

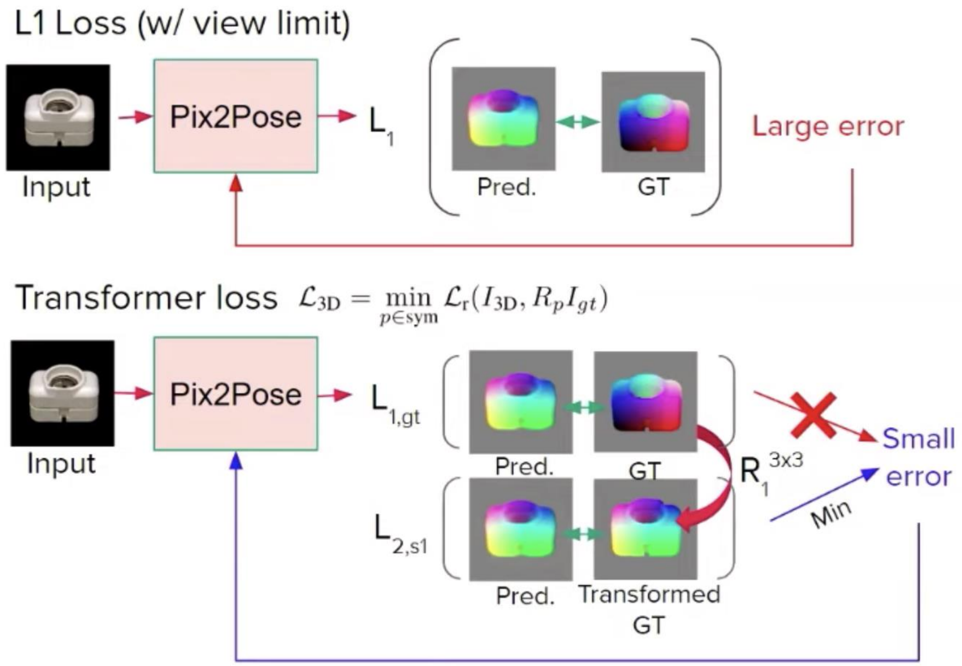
$$\mathcal{L}_{\mathrm{r}} = \frac{1}{n}\Big[\beta \sum_{i \in M} ||I^i_{\mathrm{3D}} - I^i_{\mathrm{gt}}||_1 + \sum_{i \notin M} ||I^i_{\mathrm{3D}} - I^i_{\mathrm{gt}}||_1\Big]$$

- n : 이미지 전체 픽셀의 개수
- M : GT 이미지의 오브젝트 마스크

Network training : Loss

(2) **Standard loss**: 대칭 물체에서 문제 발생

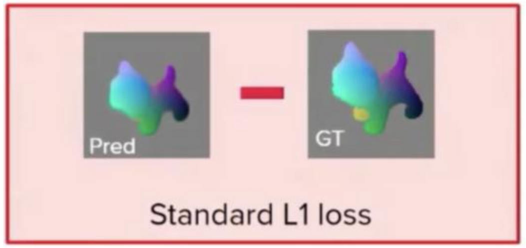




Network training : Loss

(2) Standard loss: 기본 L1 로스

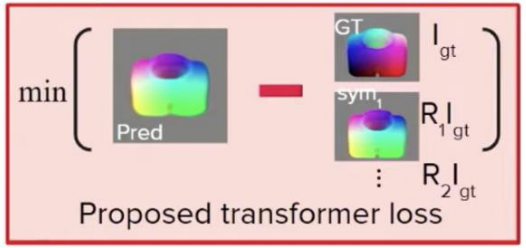
$$\boxed{\mathcal{L}_{\mathrm{r}}} = \frac{1}{n}\left[\beta \sum_{i \in M} ||I^i_{\mathrm{3D}} - I^i_{\mathrm{gt}}||_1 + \sum_{i \notin M} ||I^i_{\mathrm{3D}} - I^i_{\mathrm{gt}}||_1\right]$$



- n : 이미지 전체 픽셀의 개수
- M : GT 이미지의 오브젝트 마스크

(2-1) Transformer loss: 회전시킨 GT들과의 가장 작은 L1 Loss (Symmetric objects 문제 해결)

$$\mathcal{L}_{\mathrm{3D}} = \min_{p \in \mathrm{sym}} \mathcal{L}_{\mathrm{r}}(I_{\mathrm{3D}}, R_p I_{gt}),$$

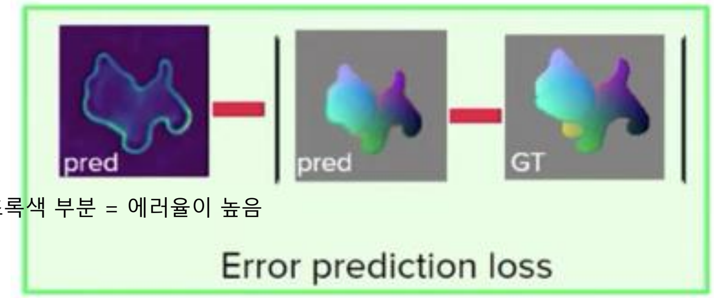


- R_p : transformed matrix
- Sym : symmetric pool

Network training : Loss

(3) **Error loss**: 외곽부분을 따로 잡아주는 손실함수(더욱 정밀한 pose 근사가 가능하도록 함)

$$\mathcal{L}_{\mathrm{e}} = \frac{1}{n}\sum_{i} ||I_{\mathrm{e}}^{i} - \min[\mathcal{L}_{\mathrm{r}}^{i}, 1]||_{2}^{2}, \beta = 1.$$



초록색 부분 = 에러율이 높음

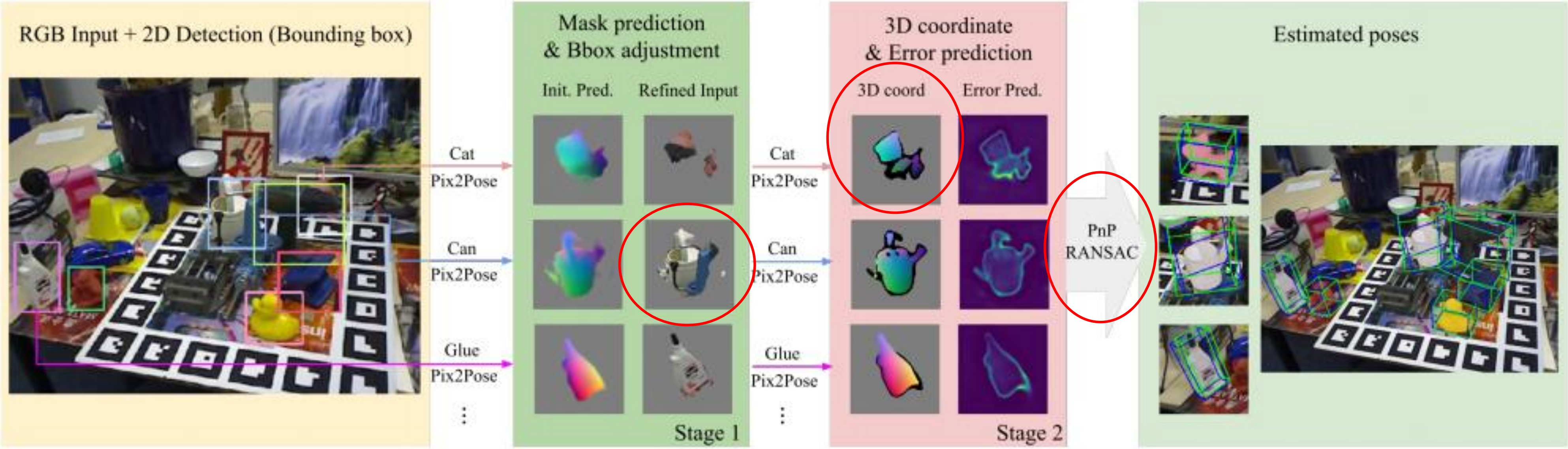
- I_e : predicted error
- min 은 Lr을 0~1 사이로 정규화 하는 효과

$$\begin{array}{|c|c|c|c|c|}\hline \; & \; & \; & \; & \; \\ \hline \; & \; & 1 & \; & \; \\ \hline 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} - \left| \begin{array}{|c|c|c|c|c|}\hline \; & \; & \; & \; & \; \\ \hline \; & \; & 5 & \; & \; \\ \hline \; & 4 & 2 & 3 & \; \\ \hline \; & 9 & 9 & 8 & \; \\ \hline \; & \; & \; & \; & \; \\ \hline \end{array} - \begin{array}{|c|c|c|c|c|}\hline \; & \; & \; & \; & \; \\ \hline \; & \; & 4 & \; & \; \\ \hline 7 & 4 & 2 & 3 & -7 \\ \hline 7 & 9 & 9 & 8 & -7 \\ \hline 7 & 7 & 7 & 7 & -7 \\ \hline \end{array} \right| = \begin{array}{|c|c|c|c|c|}\hline \; & \; & \; & \; & \; \\ \hline \; & \; & 1 & \; & \; \\ \hline 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Ex) 극단적으로 외곽을 잘 예측한 경우

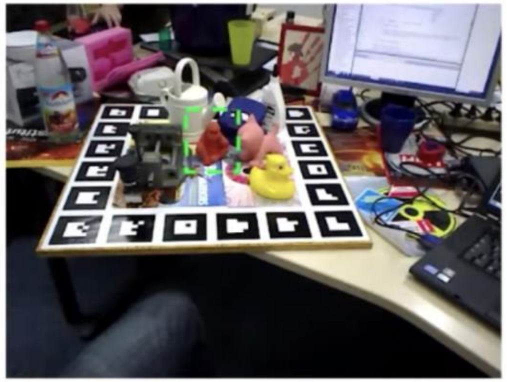
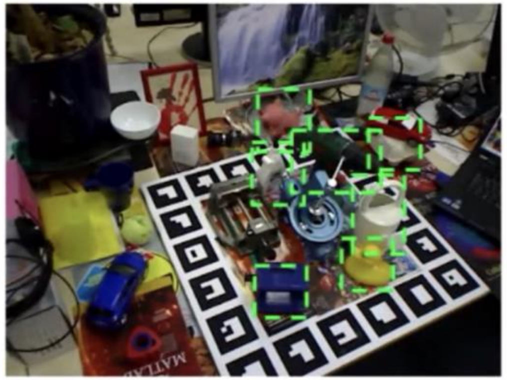
Pose prediction

- A single network is trained and used for each object class.



4. Experiments

Dataset

	LineMOD	LineMOD Occlusion	T-Less
			
	13 Objects	8 Objects	30 Objects
Training images	Sub-sampled **real images**	**Real images** in LineMOD	**Real training** images
Symmetric objects	2 Objects	2 Objects	**All**
Occlusion	No	Yes	**Yes**

Experiments : LineMOD

	ape	bvise	cam	can	cat	driller	duck	e.box*	glue*	holep	iron	lamp	phone	avg
Pix2Pose	**58.1**	**91.0**	**60.9**	**84.4**	**65.0**	**76.3**	**43.8**	**96.8**	79.4	**74.8**	**83.4**	**82.0**	45.0	**72.4**
Tekin [30]	21.6	81.8	36.6	68.8	41.8	63.5	27.2	69.6	**80.0**	42.6	75.0	71.1	**47.7**	56.0
Brachmann [2]	33.2	64.8	38.4	62.9	42.7	61.9	30.2	49.9	31.2	52.8	80.0	67.0	38.1	50.2
BB8 [25]	27.9	62.0	40.1	48.1	45.2	58.6	32.8	40.0	27.0	42.4	67.0	39.9	35.2	43.6
Lienet$^{30\%}$ [4]	38.8	71.2	52.5	86.1	66.2	82.3	32.5	79.4	63.7	56.4	65.1	89.4	65.0	65.2
BB8^{ref} [25]	40.4	91.8	55.7	64.1	62.6	74.4	44.3	57.8	41.2	67.2	84.7	76.5	54.0	62.7
Implicit^{syn} [29]	4.0	20.9	30.5	35.9	17.9	24.0	4.9	81.0	45.5	17.6	32.0	60.5	33.8	31.4
SSD-6D$^{syn/ref}$ [15]	65	80	78	86	70	73	66	100	100	49	78	73	79	76.7
Rad$^{syn/ref}$ [26]	-	-	-	-	-	-	-	-	-	-	-	-	-	***78.7***

Experiments : LineMOD Occlusion

Method	**Pix2Pose**	Oberweger† [23]	PoseCNN† [33]	Tekin [30]
ape	**22.0**	17.6	9.6	2.48
can	44.7	**53.9**	45.2	17.48
cat	**22.7**	3.31	0.93	0.67
driller	44.7	**62.4**	41.4	7.66
duck	15.0	19.2	**19.6**	1.14
eggbox*	25.2	**25.9**	22.0	-
glue*	32.4	**39.6**	38.5	10.08
holep	**49.5**	21.3	22.1	5.45
Avg	**32.0**	30.4	24.9	6.42

- Texture 정보 사용
- Pose variation 많음

Experiments : T-Less

Input	RGB only		RGB-D	
Method	**Pix2Pose**	Implicit [29]	Kehl [16]	Brachmann [2]
Avg	**29.5**	18.4	24.6	17.8

5. Conclusion

Conclusion

- **Pix2Pose : RGB 이미지를 이용한 물체의 6D Pose estimation model**
  - Texture정보 필요 없음
  - GAN 학습방식 사용 : Occlusion에 강건
  - Transformer Loss 제안 : 유한개의 대칭 포즈를 가진 물체에 대한 문제 해결

Thank You

Reference

- Paper : https://arxiv.org/pdf/1908.07433.pdf
- ICCV 2019 Oral presentation : https://www.youtube.com/watch?v=zem03fZWLrQ
- GAN : https://github.com/HYU-AILAB/ai-seminar/blob/master/season_06/11.%20Generative%20Adversarial%20Network/GAN_190408_%EC%A0%95%EC%A7%80%EC%9D%80.pdf