Affordance Detection with 2D and 3D data

KIST, 고려대학교 박사과정
이정호

Korea Institute of Science and Technology

한국과학기술연구원

목차

- Affordance 란?
- AffordanceNet (using 2D data)
- PointNet (using 3D data)
- PointFusion (using 2D and 3D data)
- Affordance Detection (using 2D and 3D data)

Affordance

Affordance 란?

행동 유도성. 대상의 어떤 속성이 유기체로 하여금 특정한 행동을 하게끔 유도하거나 특정 행동을 쉽게 하게 하는 성질. 예컨대, 사과의 빨간색은 따 먹고자 하는 행동을 유도하며, 적당한 높이의 받침대는 앉는 행동을 잘 지원한다.

Affordance

Affordance 란?

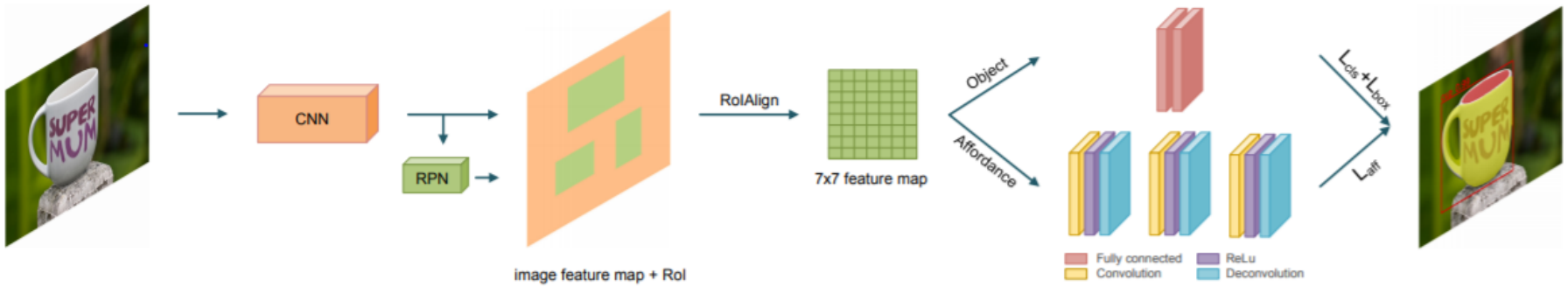
Affordance

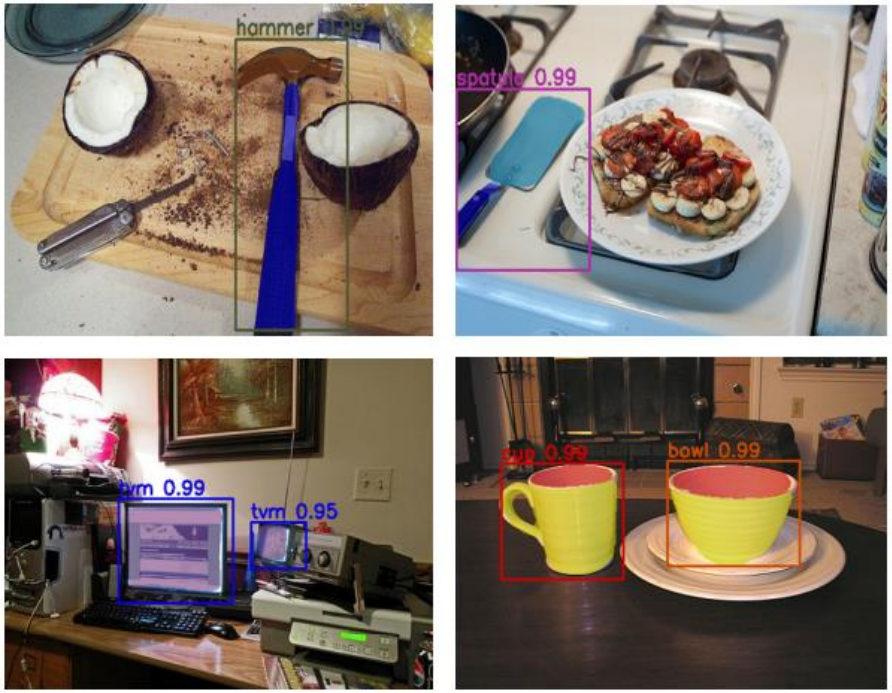
Affordance 란?

컵이라고 인식 못하는 컵을 위해 Affordance라는 개념을 도입

AffordanceNet

AffordanceNet



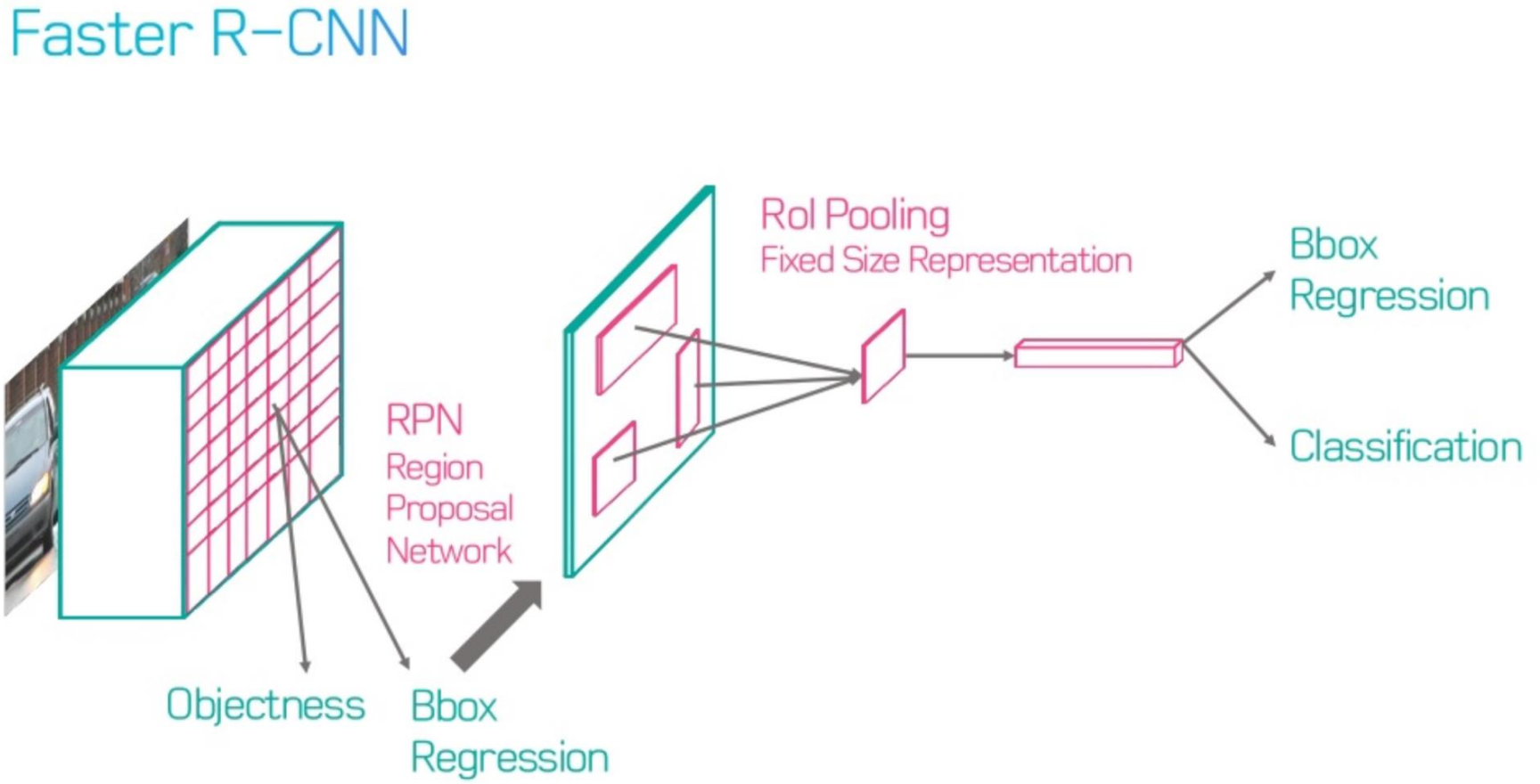


주요 특징

1. ROI Align
2. Deconvolution for High Affordance Mask
3. A Multi-task Loss Function
4. Use only RGB data

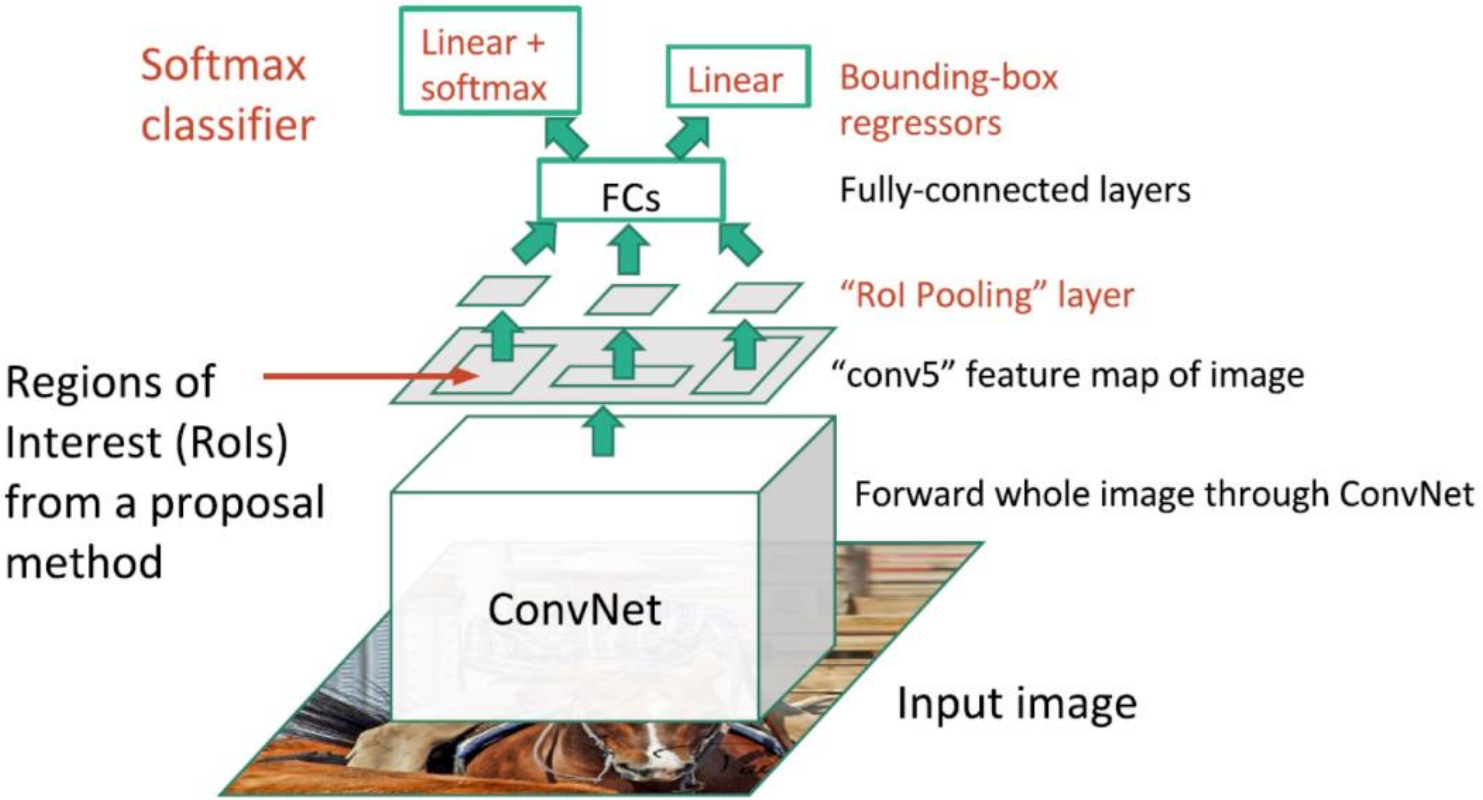
[1] AffordanceNet : Thanh-Toan Do, Anh Nguyen, Ian Reid, ICRA 2018

AffordanceNet



AffordanceNet

Faster R-CNN



AffordanceNet

ROI Pooling - ROI Align

RoI Pooling (Fast R-CNN)
- Input: Each RoI
- Output: 7x7 Pooled Feature

RoI Align (Mask R-CNN)
- Input: Each RoI
- Output: 7x7 Pooled Feature

- 7x7 Feature for FCN
- Too Small to Segment
- Need Precise Pooled Feature

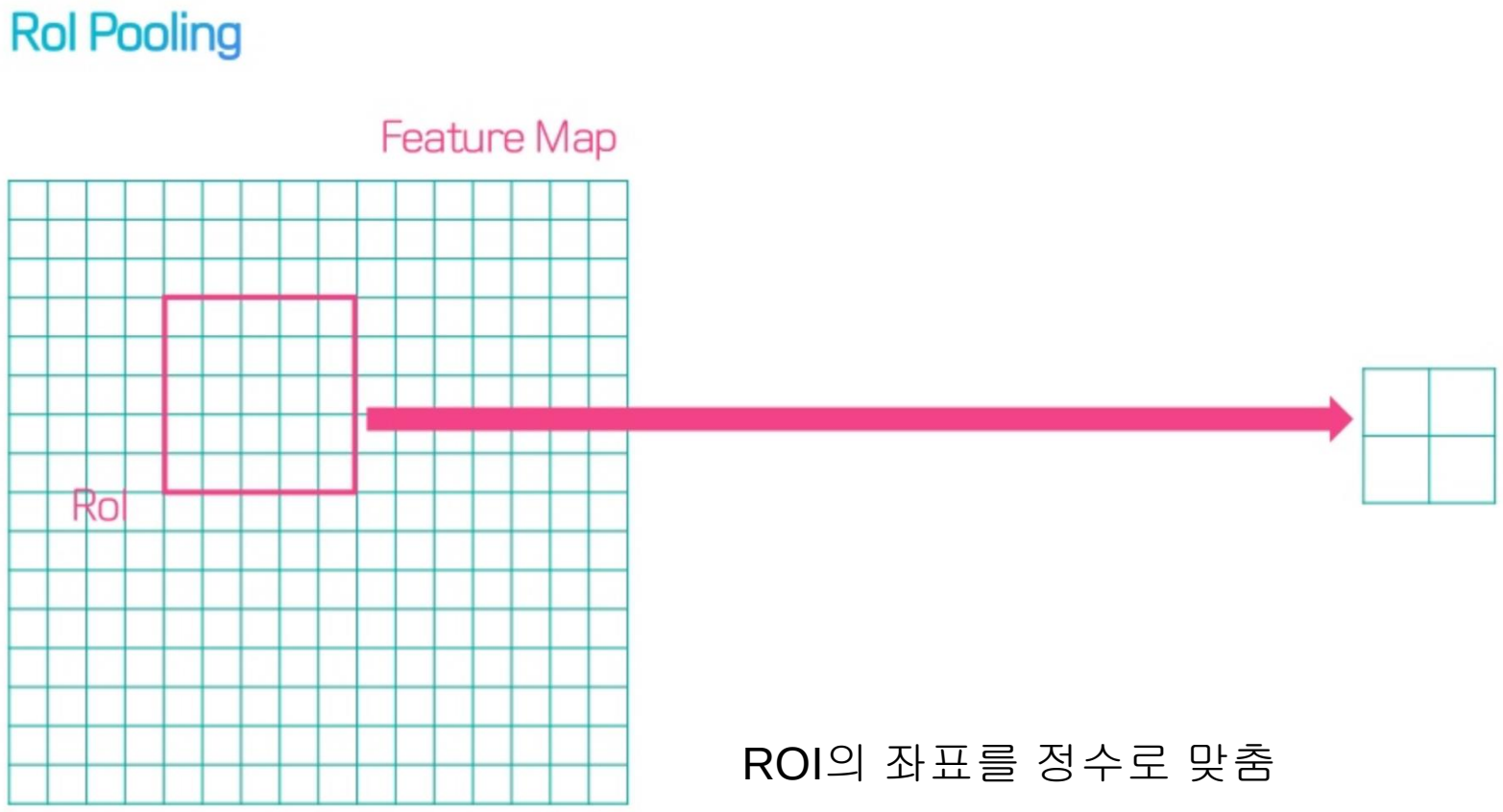
AffordanceNet

ROI Pooling

Feature Map 상에서 ROI (Region of Interest) 의 좌표를 소수점으로 표시

AffordanceNet

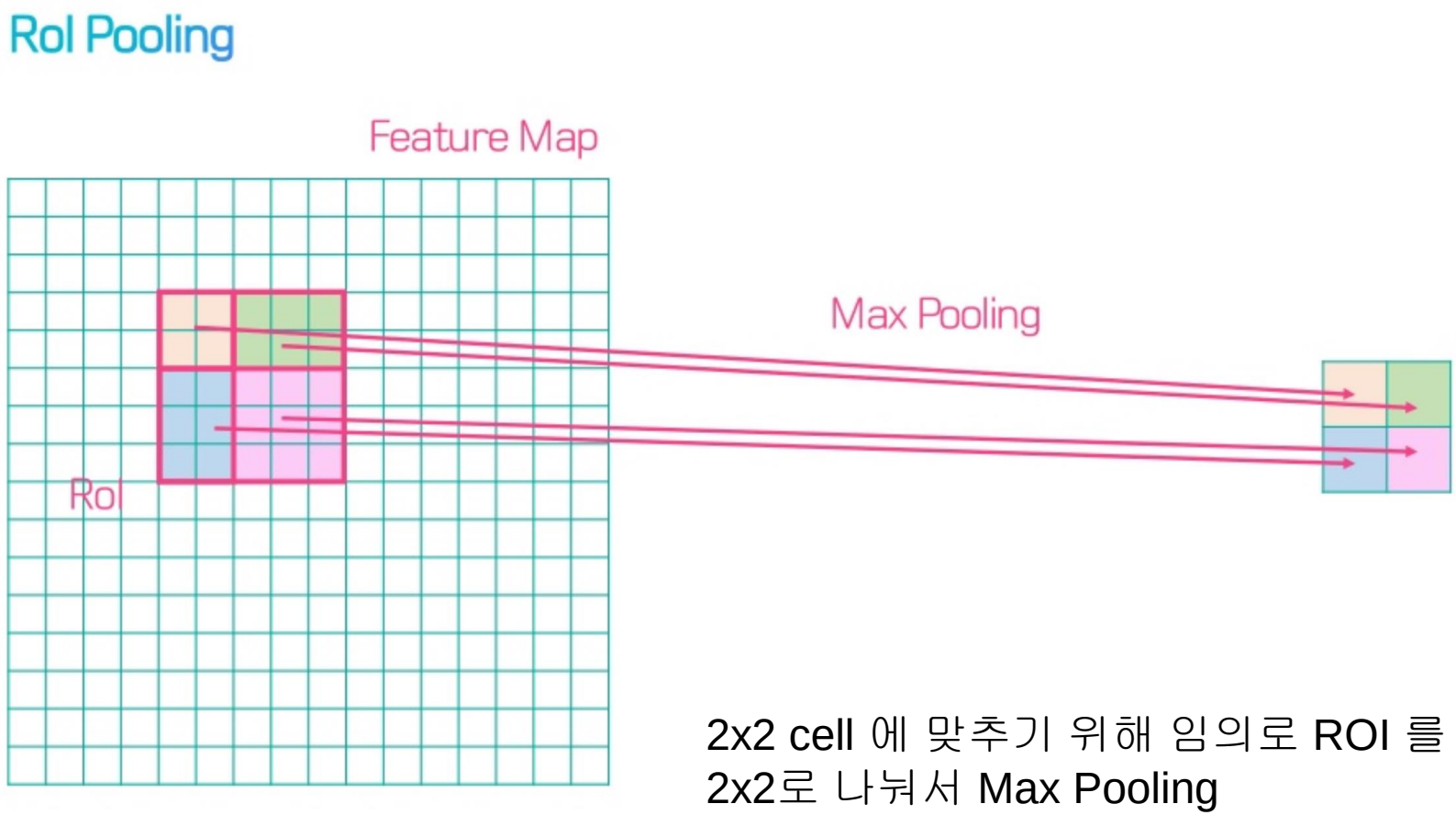
ROI Pooling



ROI의 좌표를 정수로 맞춤

AffordanceNet

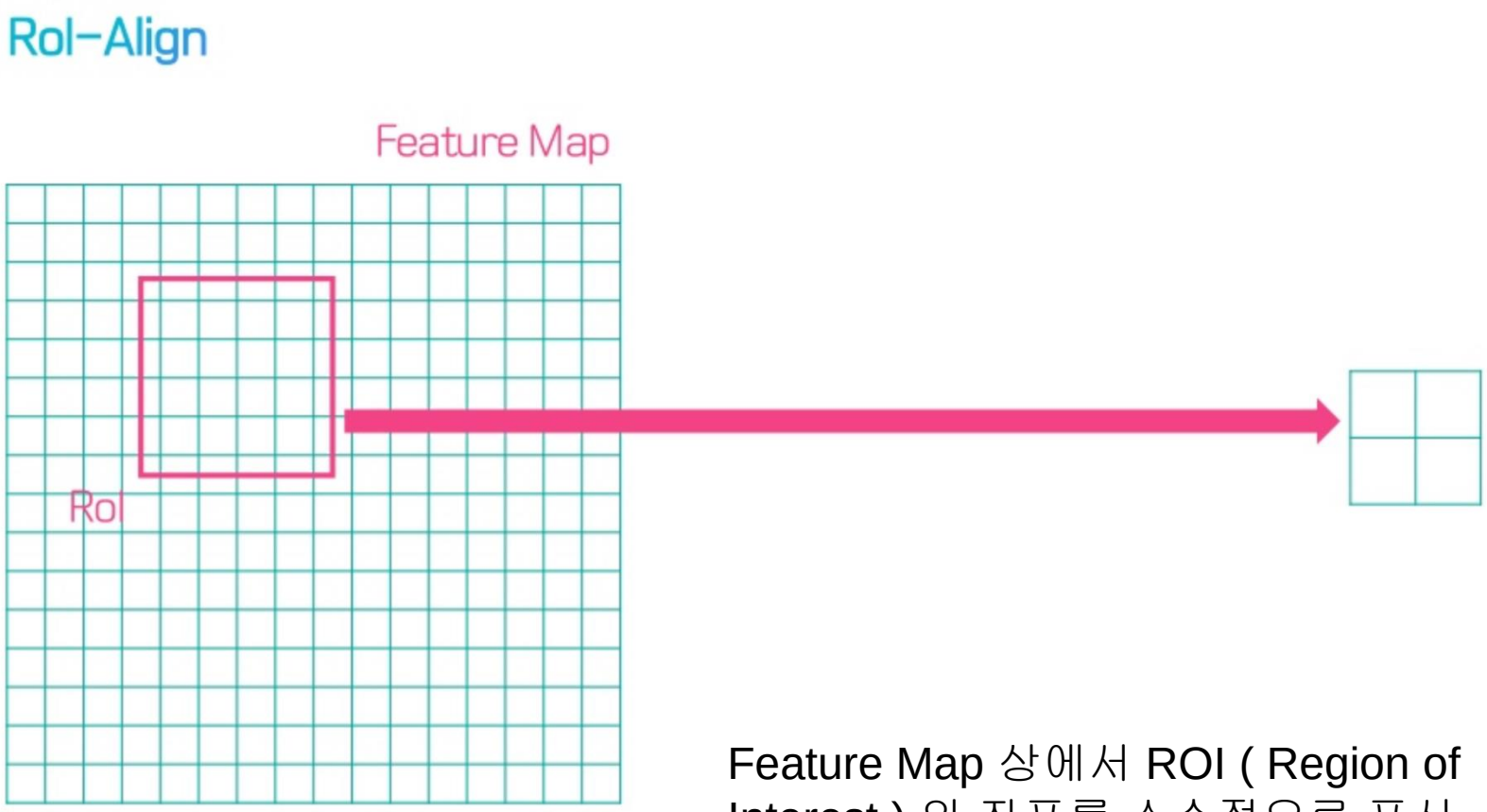
ROI Pooling



2x2 cell 에 맞추기 위해 임의로 ROI 를 2x2로 나눠서 Max Pooling

AffordanceNet

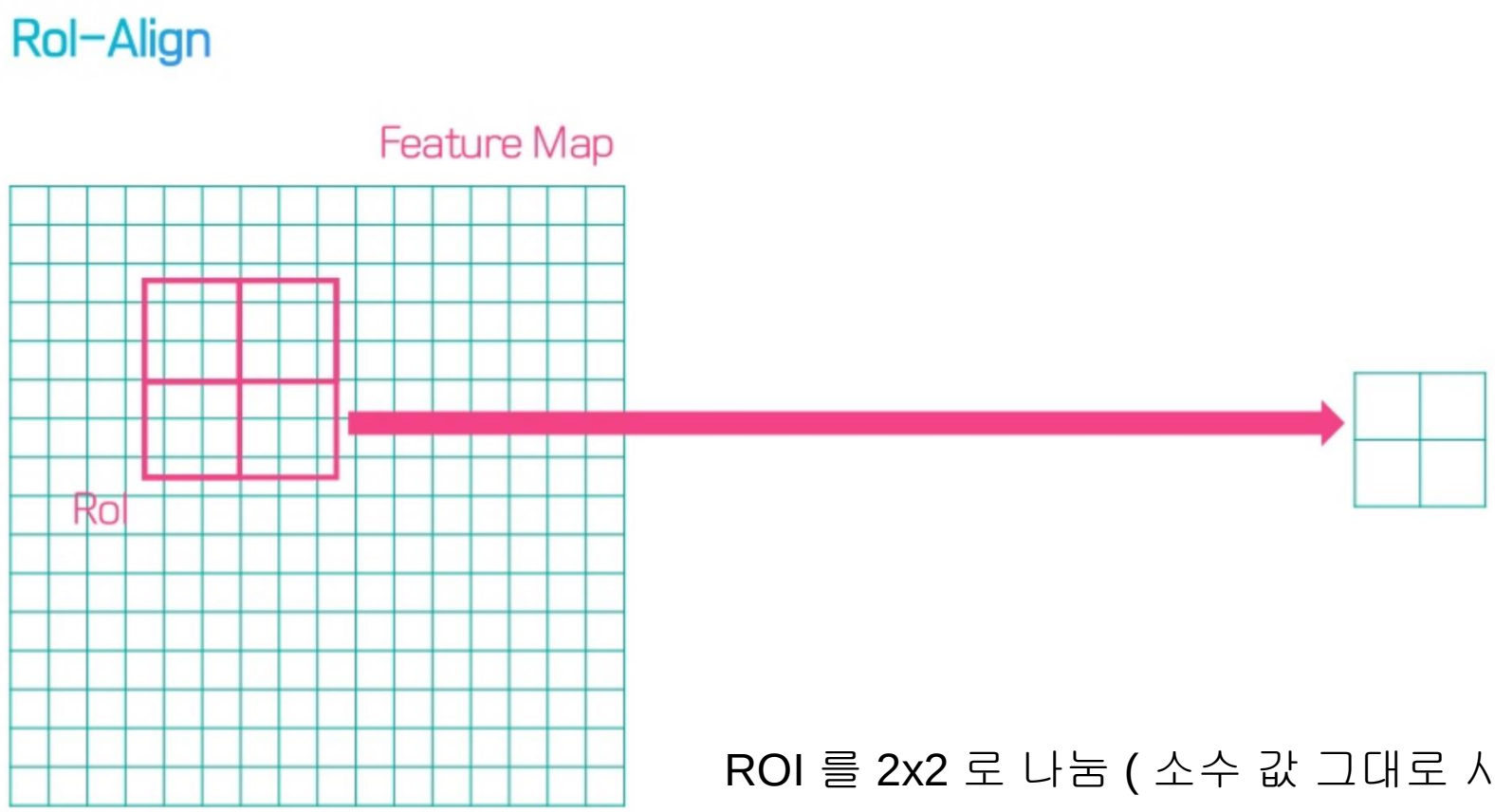
ROI Align



Feature Map 상에서 ROI (Region of Interest) 의 좌표를 소수점으로 표시

AffordanceNet

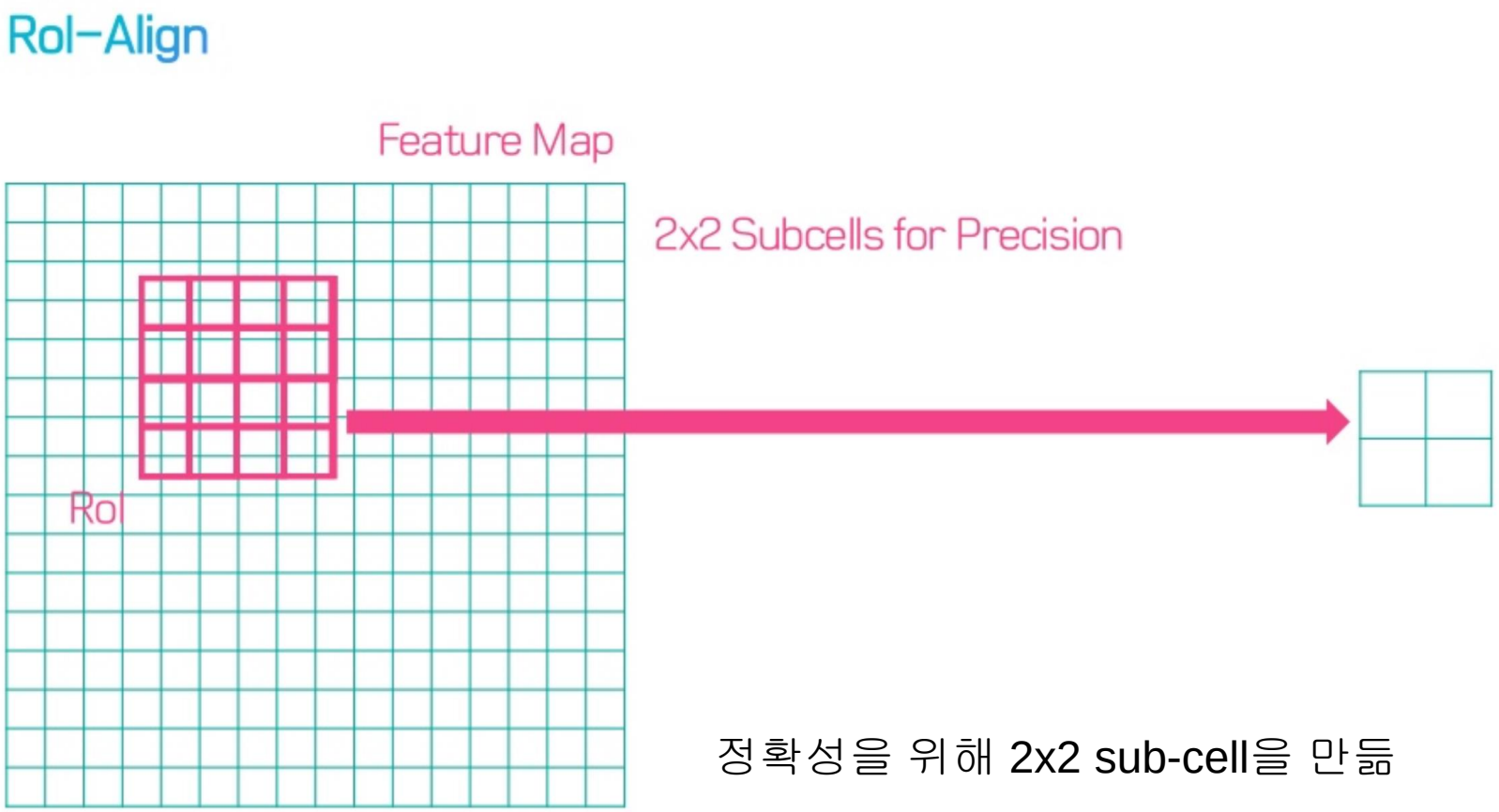
ROI Align



ROI 를 2x2 로 나눔 (소수 값 그대로 사용)

AffordanceNet

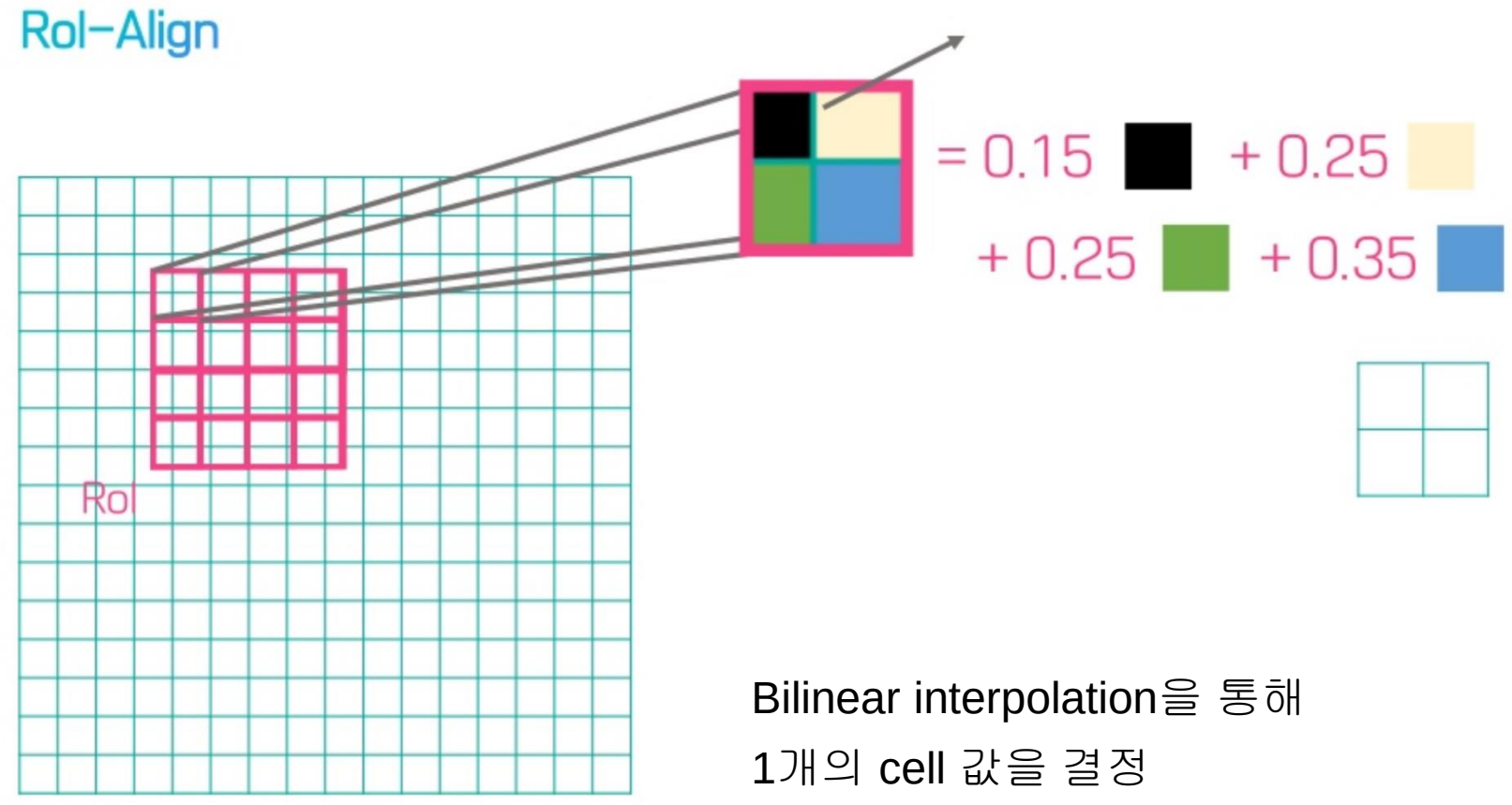
ROI Align



정확성을 위해 2x2 sub-cell을 만듦

AffordanceNet

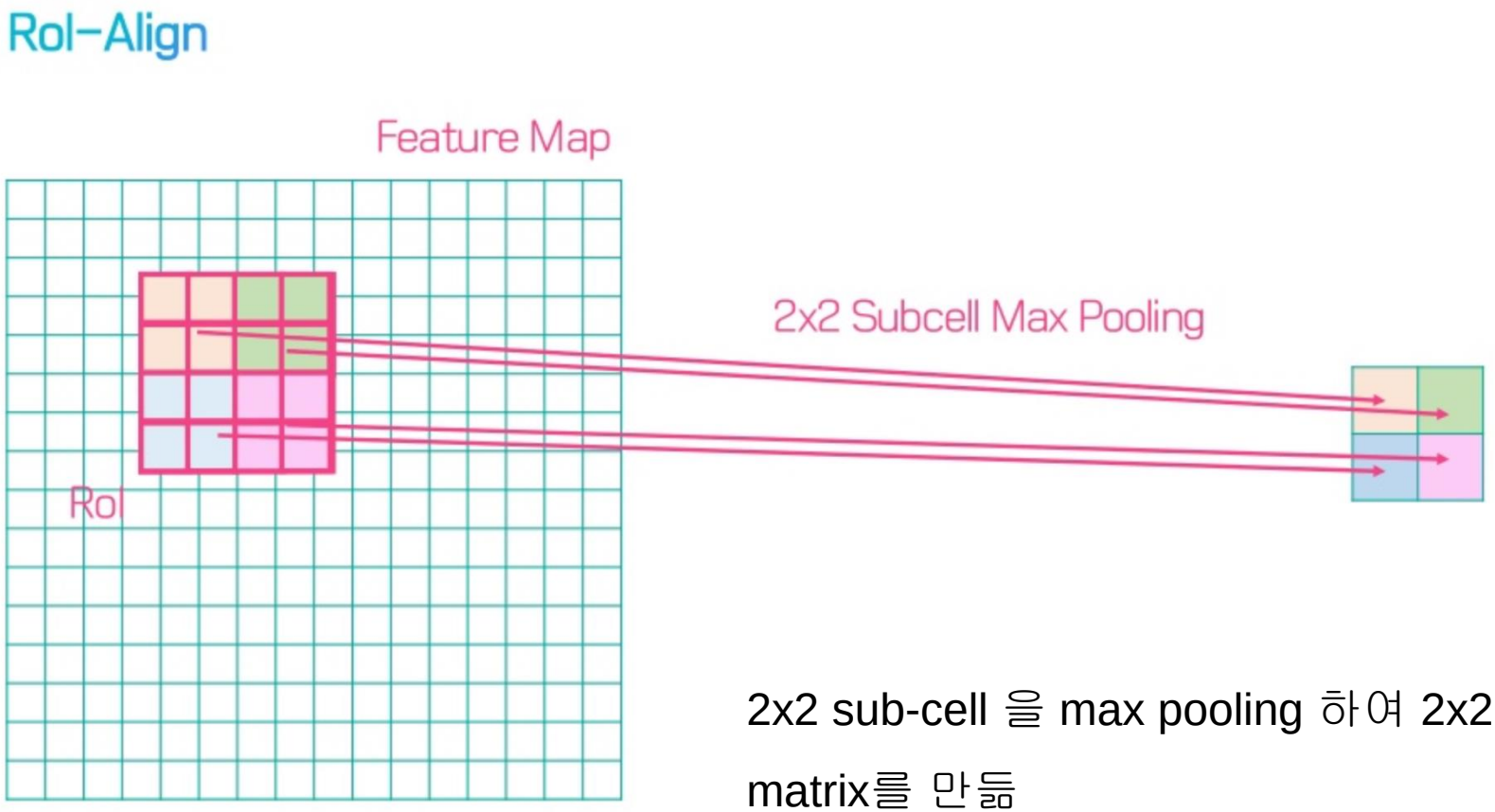
ROI Align



Bilinear interpolation을 통해
1개의 cell 값을 결정

AffordanceNet

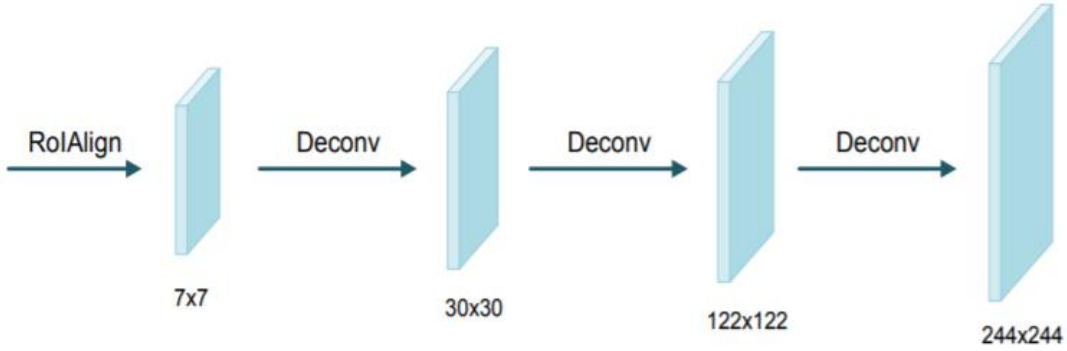
ROI Align

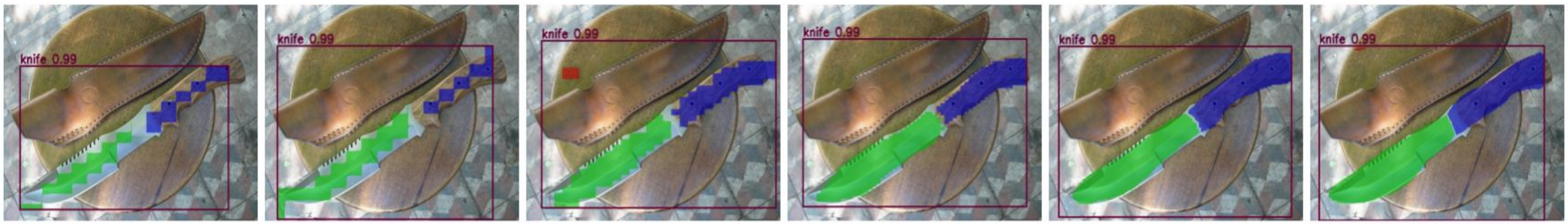


2x2 sub-cell 을 max pooling 하여 2x2 의 matrix를 만듦

AffordanceNet

A Sequence of Deconv layers





AffordanceNet14 AffordanceNet14_6conv AffordanceNet28 AffordanceNet56 AffordanceNet112 AffordanceNet244

- Instance Segmentation 방법인 Mask R-CNN는 Deconv를 사용 X
- 경험적으로 affordance detection 에는 Deconv를 사용해야 함
  (part segmentation 개념이므로 High resolution이 필요)
- Deconv 앞에 Conv는 Deconv에 사용될 feature를 learning

AffordanceNet

A Multi-task Loss Function

$$L = L_{cls} + L_{loc} + L_{aff}$$

$$L(p, u, t^u, v, m, s) = L_{cls}(p, u) + I[u \geq 1] L_{loc}(t^u, v) + I[u \geq 1] L_{aff}(m, s)$$

$$L_{cls}(p, u) = -log(p_u)$$

$$L_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h\}} Smooth_{L1}(t_i^u - v_i)$$

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x - 0.5| & \text{otherwise} \end{cases}$$

$$L_{aff}(m, s) = \frac{-1}{N} \sum_{i \in RoI} log(m_{s_i}^i)$$

u : GT class

v : GT bbox

s : aff mask

p_u : softmax class

t^u: regressed box offset

m_{si}^i : softmax aff label at pixel i

N : nPixel in the ROI

I[u≥1] : u ≥ 1 이면 1, otherwise 0

AffordanceNet

Experiments

PERFORMANCE ON IIT-AFF DATASET

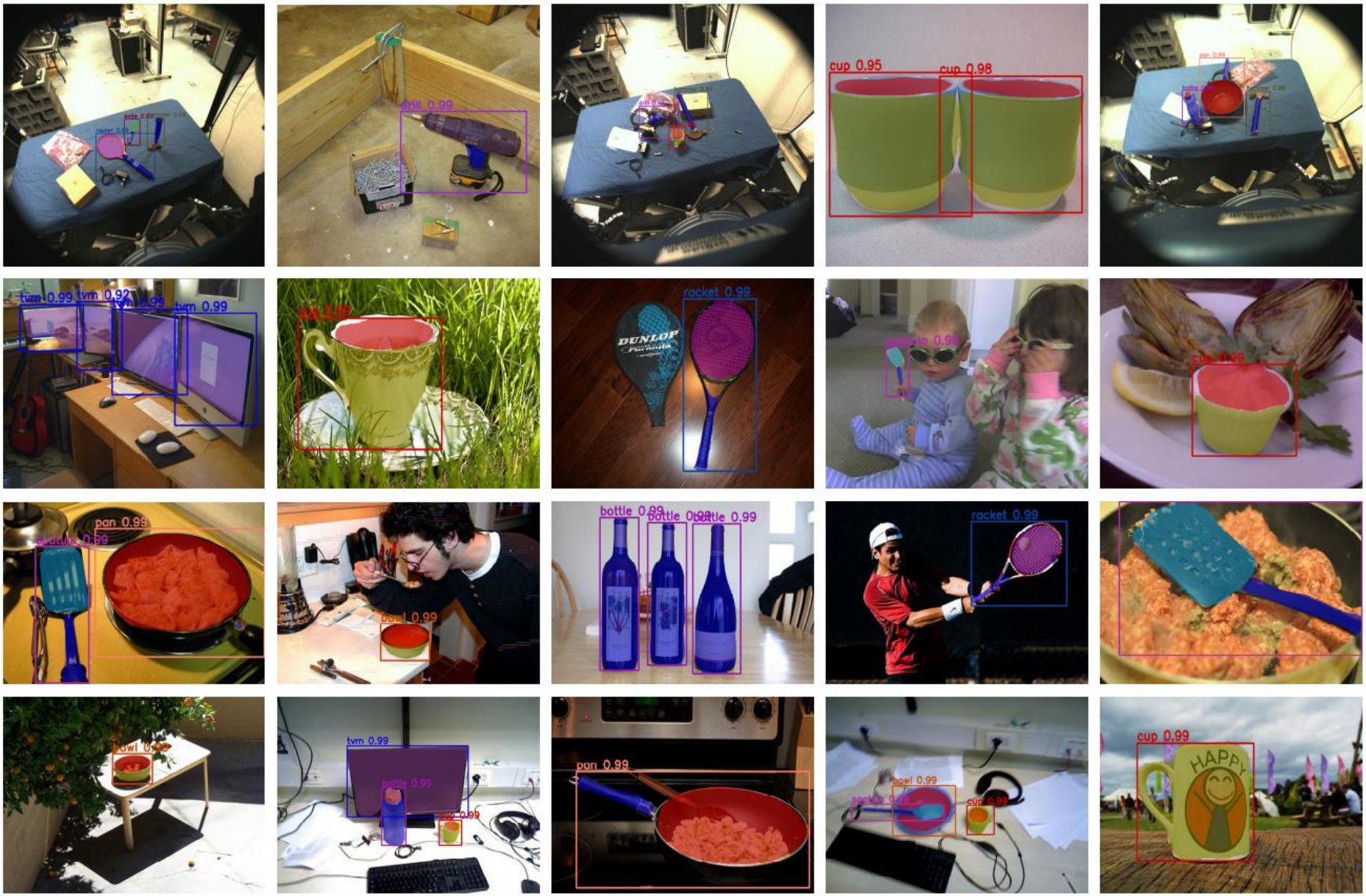
	ED-RGB [16]	ED-RGBD [16]	DeepLab [24]	DeepLab-CRF [24]	BB-CNN [6]	BB-CNN-CRF [6]	AffordanceNet (ours)
contain	66.38	66.00	68.84	69.68	75.60	75.84	**79.61**
cut	60.66	60.20	55.23	56.39	69.87	71.95	**75.68**
display	55.38	55.11	61.00	62.63	72.04	73.68	**77.81**
engine	56.29	56.04	63.05	65.11	72.84	74.36	**77.50**
grasp	58.96	58.59	54.31	56.24	63.72	64.26	**68.48**
hit	60.81	60.47	58.43	60.17	66.56	67.07	**70.75**
pound	54.26	54.01	54.25	55.45	64.11	64.86	**69.57**
support	55.38	55.08	54.28	55.62	65.01	66.12	**69.81**
w-grasp	50.66	50.42	56.01	57.47	67.34	68.41	**70.98**
Average	57.64	57.32	58.38	59.86	68.57	69.62	**73.35**

Affordance 종류	Affordance 설명
Contain (빨강)	담을 수 있는 부분
Hit (맑은 파랑)	때리는 부분 (ex. 테니스 넷)
Cut (맑은 초록)	자르는 부분
Display (하늘색)	정보를 보여주는 부분
Engine (남색)	도구의 엔진 부분
Support (노란색)	물체를 받쳐주는 평평한 부분
Pound (진한 하늘색)	두드리는 부분 (ex. 망치 머리)
grasp (진한 노란색)	잡을 수 있는 부분
w-grasp (보라색)	손바닥으로 감싸서 잡는 부분

PERFORMANCE ON UMD DATASET

	HMP [4]	SRF [4]	DeepLab [24]	ED-RGB [16]	ED-RGBD [16]	ED-RGB HHA [16]	AffordanceNet (ours)
grasp	0.367	0.314	0.620	0.719	0.714	0.673	**0.731**
w-grasp	0.373	0.285	0.730	0.769	0.767	0.652	**0.814**
cut	0.415	0.412	0.600	0.737	0.723	0.685	**0.762**
contain	0.810	0.635	**0.900**	0.817	0.819	0.716	0.833
support	0.643	0.429	0.600	0.780	0.803	0.663	**0.821**
scoop	0.524	0.481	**0.800**	0.744	0.757	0.635	0.793
pound	0.767	0.666	**0.880**	0.794	0.806	0.701	0.836
Average	0.557	0.460	0.733	0.766	0.770	0.675	**0.799**

AffordanceNet



PointNet

1. Introduction

- 기존 3D classification : 3D voxel 을 주로 사용 (conv 를 사용하기 위해)
- 이 논문에서는 input을 point cloud로 받고, output으로 cls, seg, part seg를 보냄
- Symmetric function 인 max pooling 사용 : 더 정보가 많은 point들의 최적화 된 영역을 배움
- Global feature 추출 : classification 에 사용
- Point feature 추출 : global feature 와 함께 segmentation 에 사용
- Rigid or affine transformation 적용 : data를 정규화 시도를 하는 data에 의존적인 STN 을 적용

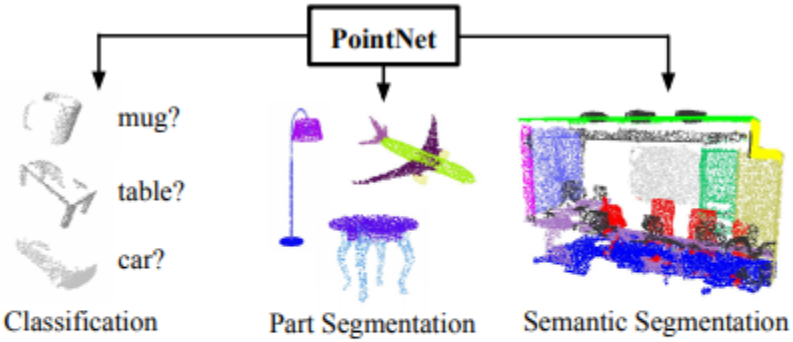


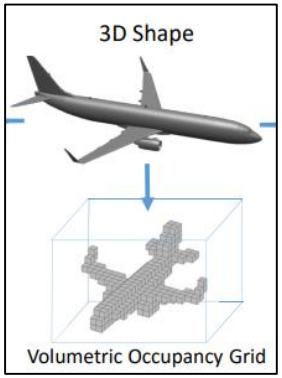
Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

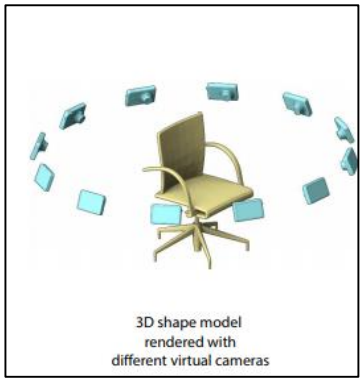
[2] PointNet : Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas, CVPR 2017

PointNet

2. Related Work

- Point Cloud Features
  - 대부분의 point cloud feature 사용하는 논문들은 handcraft 가 많음
  - Point feature 를 종종 통계적 특성으로 encode 하기도 함
- Deep Learning on 3D Data
  - Volumetric CNNs : voxel로 변환
  - Multiview CNNs : 2D 영상을 여러 장 찍어 3D 로 rendering
  - Spectral CNNs : mesh에 spectral CNN 사용. Manifold mesh 에 제한
  - Feature-based DNNs : 3D data를 vector로 변환. feature 추출에 제한
- Deep Learning on Unordered Sets
  - Attention 을 사용한 연구가 있으나 NLP 분야라서 geometry 역할이 없음





PointNet

3. Problem Statement

- Input으로 (x,y,z) 좌표만 사용
- Classification 에는 k 개의 class에 k개의 score가 있음
- Segmentation 에는 n x m score가 있음 (n : point cloud 개수, m : sub-category)

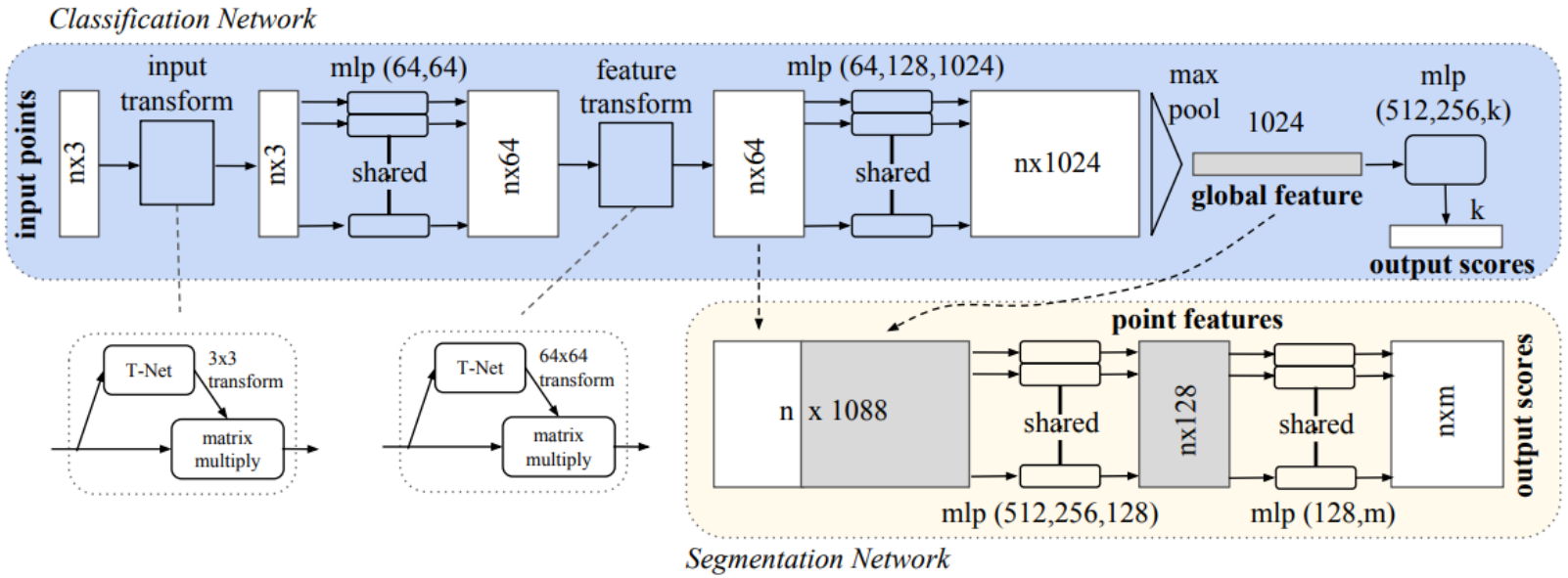


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

PointNet

4. Deep Learning on Point Sets

4.1. Properties of Point Sets in R^n

- Unordered : input 에 N! 의 치환에도 invariant
- Interaction among Points : point 간의 거리가 존재. 즉, isolated 는 아님.
  Local structure 를 capture 해야 함
- Invariance under transformations : point set의 learned representation은 반드시 어떠한 transformation에도 invariant 해야 함

4. Deep Learning on Point Sets

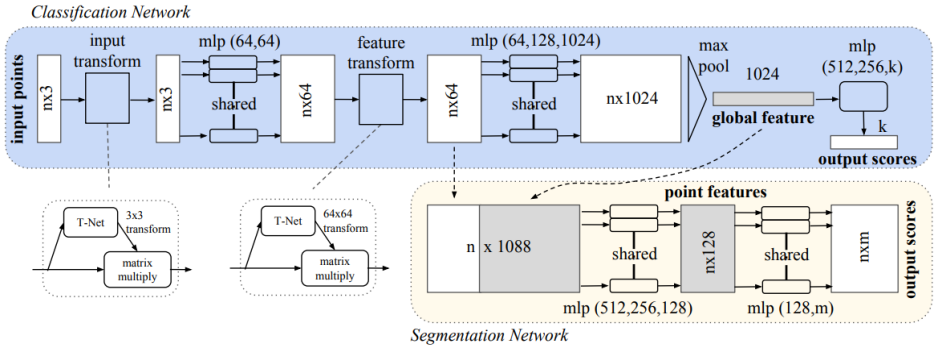
4.1. Properties of Point Sets in R^n

- Unordered : input 에 N! 의 치환에도 invariant
- Interaction among Points : point 간의 거리가 존재. 즉, isolated 는 아님.
  Local structure 를 capture 해야 함
- Invariance under transformations : point set의 learned representation은 반드시 어떠한 transformation에도 invariant 해야 함

PointNet

4. Deep Learning on Point Sets

4.2. PointNet Architecture



- 3가지 key module
  1. The max pooling
  2. Local and global information
  3. Two joint alignment networks
- Symmetry Function for Unordered Input (1. for the max pooling)
  - Input의 permutation에 invariant 하게 만들기 위해 3가지 전략
    - ① sort input into a canonical
    - ② Treat the input as a sequence to train an RNN
    - ③ Use a simple symmetric function to aggregate the information from each point

$$f(x_1, x_2, \ldots, x_n) = f(x_2, x_1, \ldots, x_n) = f(x_3, x_1, \ldots, x_n, x_{n-1})$$

PointNet

4. Deep Learning on Point Sets

4.2. PointNet Architecture

① sort input into a canonical

- High dimensional space there does not exist an ordering that is stable w.r.t. point perturbations in the general sense.

② Treat the input as a sequence to train an RNN

- It's hard to scale to thousands of input elements.

③ Use a simple symmetric function to aggregate the information from each point

- h by a mlp
- g by a composition of a single variable function and a max pooling function
- 써보니 잘됨 ㅋ

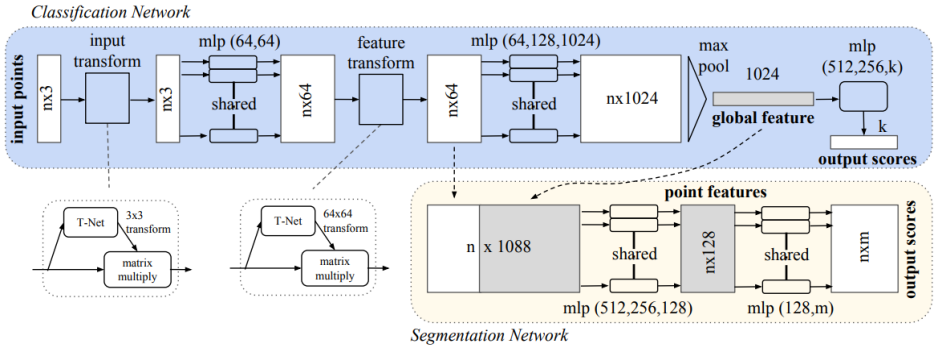
$$f(\{x_1, \ldots, x_n\}) \approx g(h(x_1), \ldots, h(x_n)), \qquad (1)$$

where $f : 2^{\mathbb{R}^N} \rightarrow \mathbb{R}$, $h : \mathbb{R}^N \rightarrow \mathbb{R}^K$ and $g : \underbrace{\mathbb{R}^K \times \cdots \times \mathbb{R}^K}_{n} \rightarrow \mathbb{R}$ is a symmetric function.

PointNet

4. Deep Learning on Point Sets

4.2. PointNet Architecture



- 3가지 key module
  1. The max pooling
  2. Local and global information
  3. Two joint alignment networks
- Local and Global Information Aggregation (for 2.)
  - Global feature 와 local feature 를 concat 해서 새로운 per point feature 를 추출 (이는 global 과 local information 을 갖고 있음)
- Joint Alignment Network
  - Point cloud 가 어떠한 geometric transformation을 겪더라도 invariant 해야 함
  - 그래서 feature extraction 하기 전에 canonical space로 align 해줌
  - STN 보다 쉬운 T-Net이라는 걸 만듬. T-Net은 이 PointNet과 비슷하게 생김
  - Feature space의 transformation matrix는 더 higher dimension 이기 때문에 최적화 하기 어려움 → Loss 에 regularization term 추가
  - A : feature alignment matrix.

$$L_{reg} = \|I - AA^T\|_F^2$$

PointNet

5. Experiments

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [27]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [29]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**	91.5	**85.9**	80.8	95.3	65.2	**93.0**	81.2	**57.9**	**72.8**	**80.6**

Table 2. **Segmentation results on ShapeNet part dataset.** Metric is mIoU(%) on points. We compare with two traditional methods [27] and [29] and a 3D fully convolutional network baseline proposed by us. Our PointNet method achieved the state-of-the-art in mIoU.

	mean IoU	overall accuracy
Ours baseline	20.12	53.19
Ours PointNet	**47.71**	**78.62**

Table 3. **Results on semantic segmentation in scenes.** Metric is average IoU over 13 classes (structural and furniture elements plus clutter) and classification accuracy calculated on points.

	table	chair	sofa	board	mean
# instance	455	1363	55	137	
Armeni et al. [1]	46.02	16.15	**6.78**	3.91	18.22
Ours	**46.67**	**33.80**	4.76	**11.72**	**24.24**

Table 4. **Results on 3D object detection in scenes.** Metric is average precision with threshold IoU 0.5 computed in 3D volumes.

Figure 4. **Qualitative results for semantic segmentation.** Top row is input point cloud with color. Bottom row is output semantic segmentation result (on points) displayed in the same camera viewpoint as input.

PointFusion

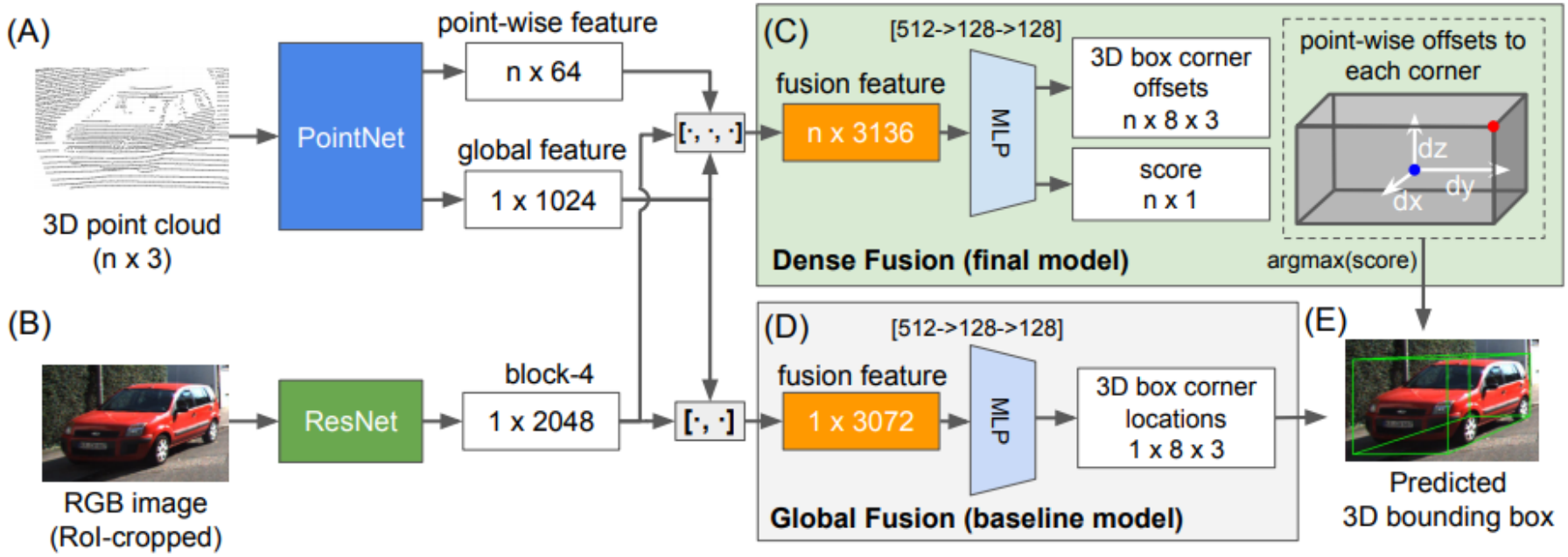
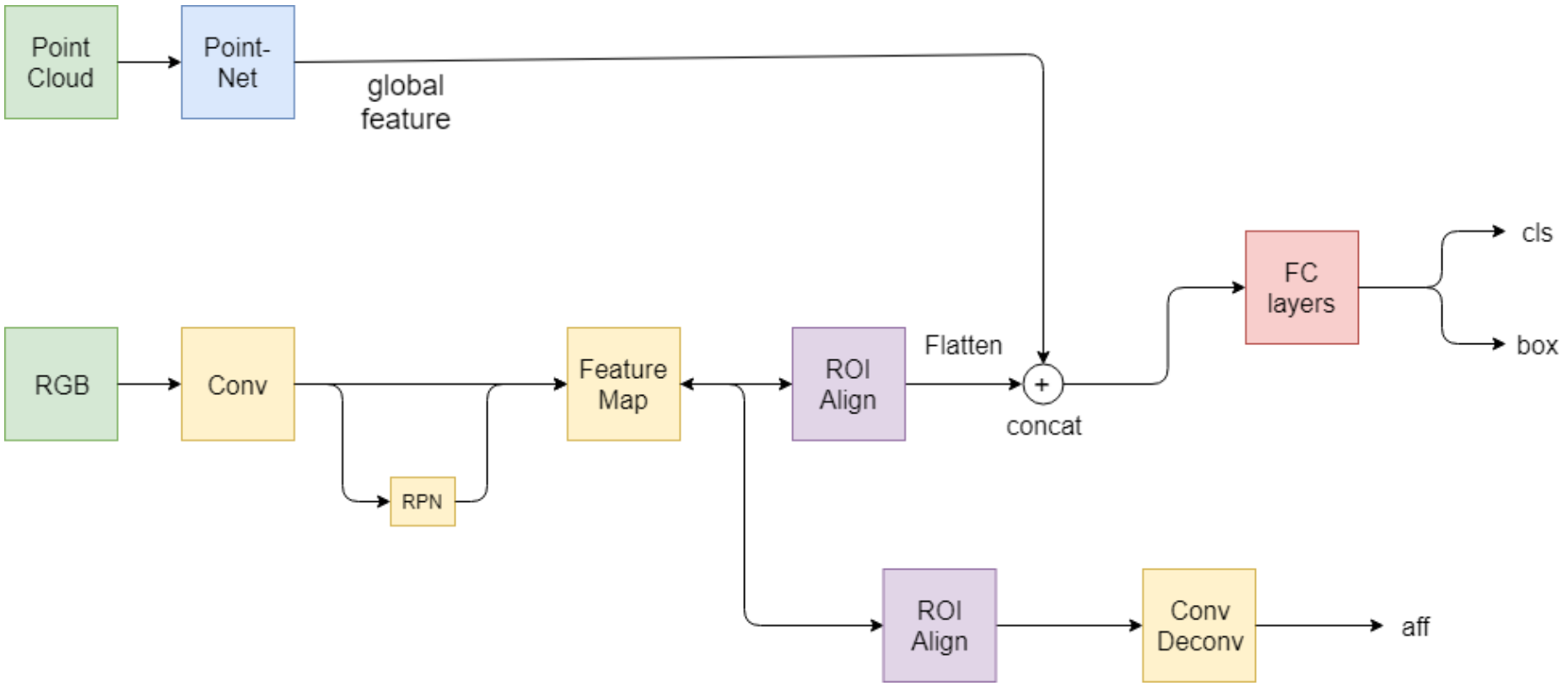


Figure 2. An overview of the dense PointFusion architecture. PointFusion has two feature extractors: a PointNet variant that processes raw point cloud data (A), and a CNN that extracts visual features from an input image (B). We present two fusion network formulations: a vanilla *global* architecture that directly regresses the box corner locations (D), and a novel *dense* architecture that predicts the spatial offset of each of the 8 corners relative to an input point, as illustrated in (C): for each input point, the network predicts the spatial offset (white arrows) from a corner (red dot) to the input point (blue), and selects the prediction with the highest score as the final prediction (E).

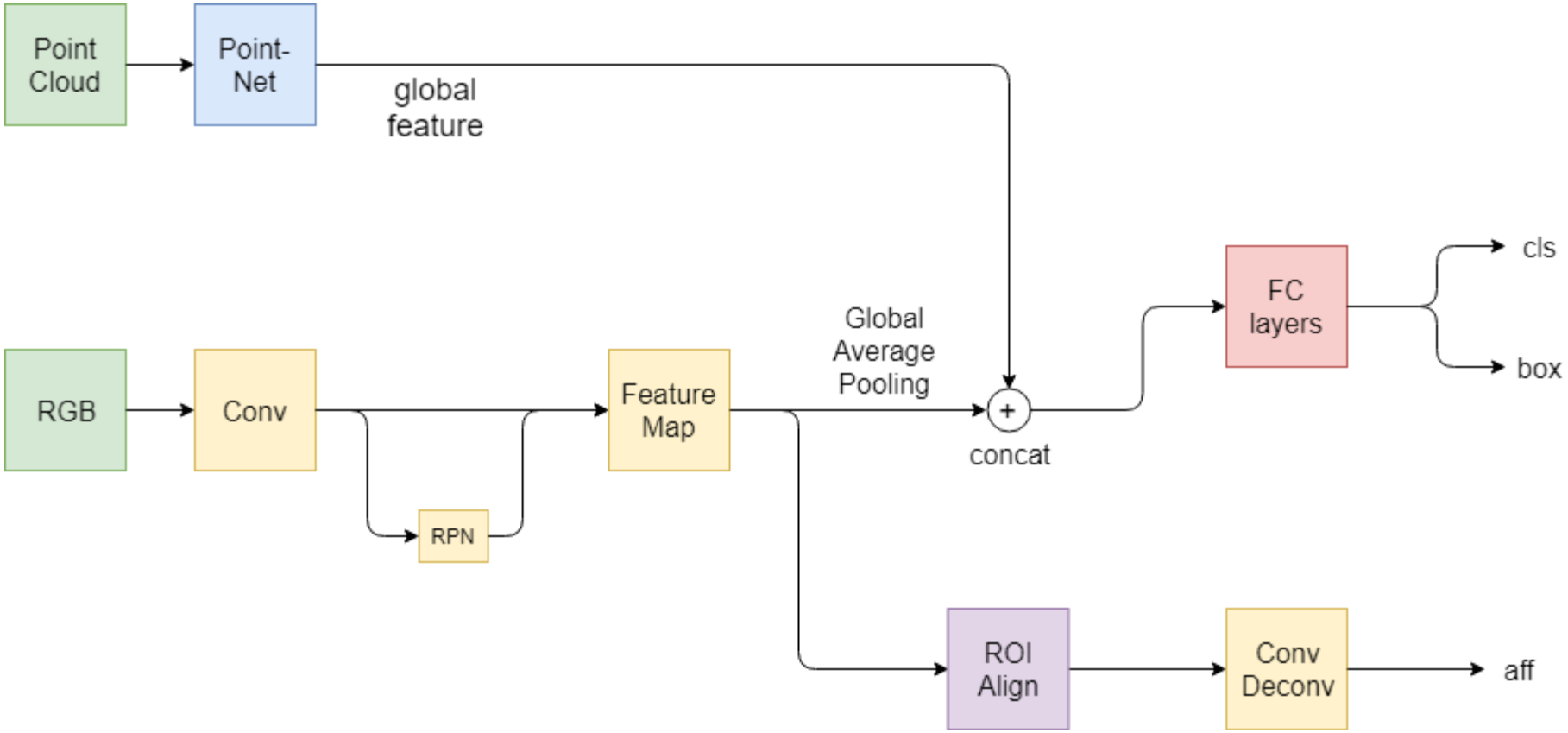
[3] PointFusion : Danfei Xu, Dragomir Anguelov, Ashesh Jain, CVPR 2018

Aff Detection with 2D and 3D data



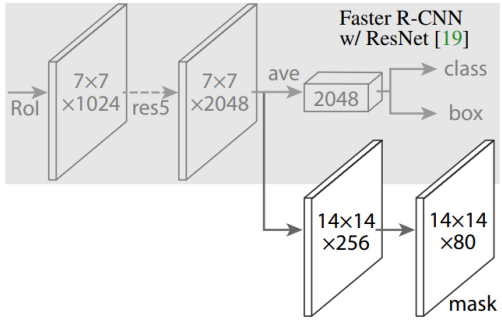
Proposal Network (RoI - Flatten)

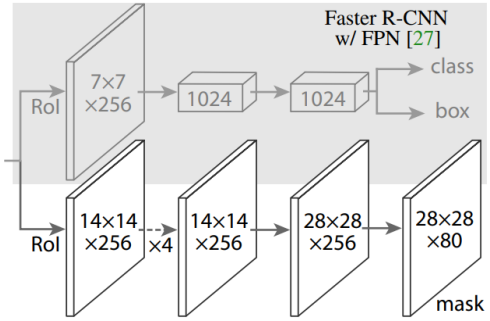
Aff Detection with 2D and 3D data

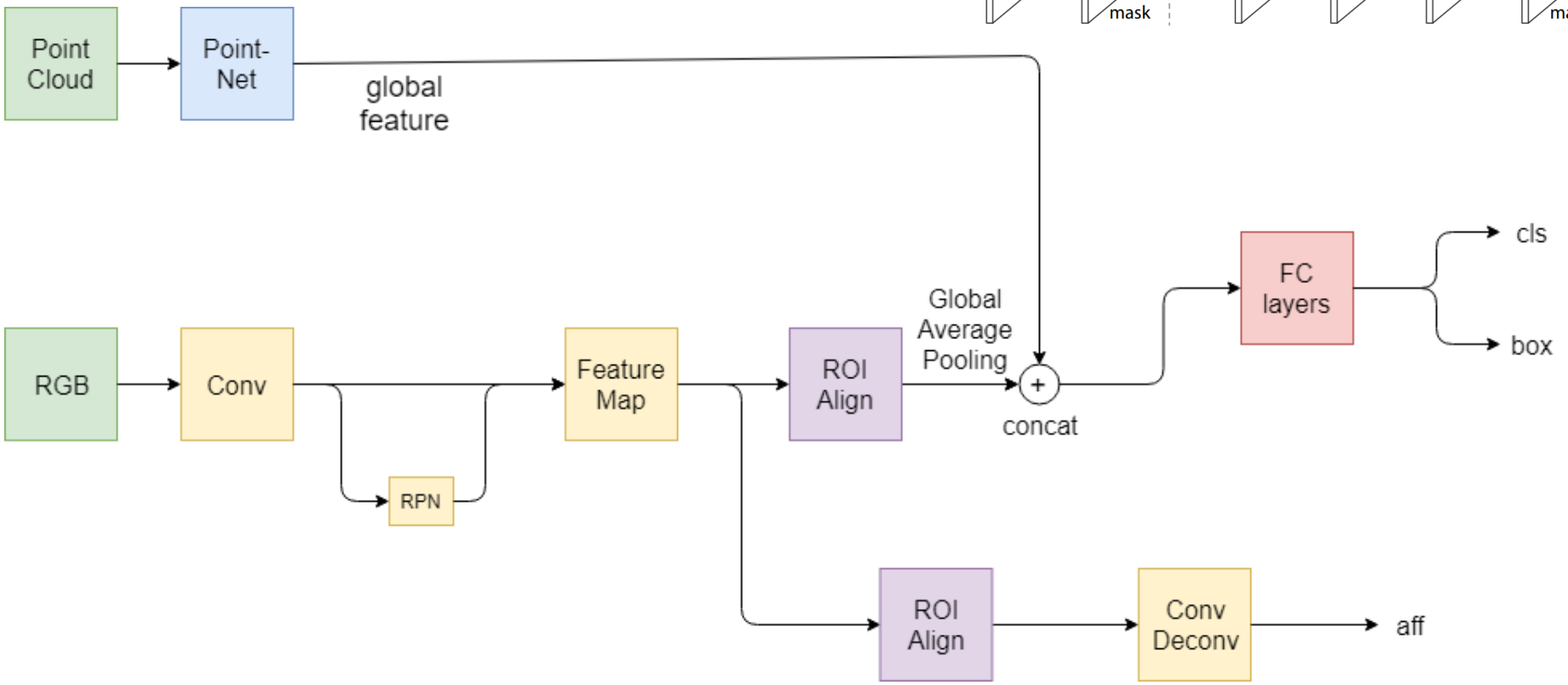


Proposal Network (Feature Map - GAP)

Aff Detection with 2D and 3D data







Proposal Network (RoI - GAP)

Aff Detection with 2D and 3D data

Conclusion

1. Object Affordance 의 정의
2. Aff Detection using RGB data
3. 3D segmentation using point cloud (3d data)
4. Mixing RGB and point cloud features
5. Applied Object Aff Detection

Q & A