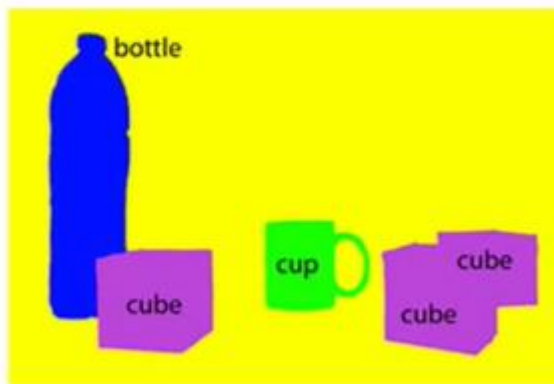
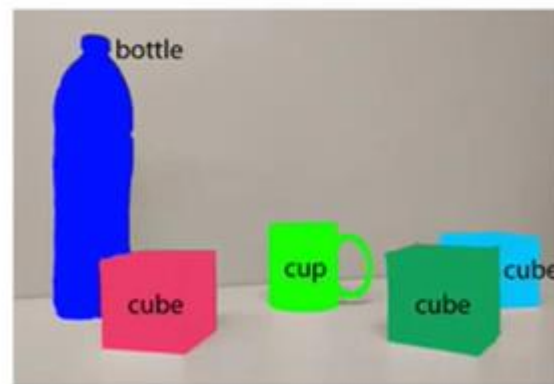


# Segmentation

Semantic Segmentation



Instance Segmentation



**Semantic segmentation – 동일 오브젝트의 경우 동일 형식으로 masking**

**Instance segmentation – 개별 오브젝트 별로 masking**

# Mask Rcn

Mask R CNN  
FCN + Faster RCNN



Input



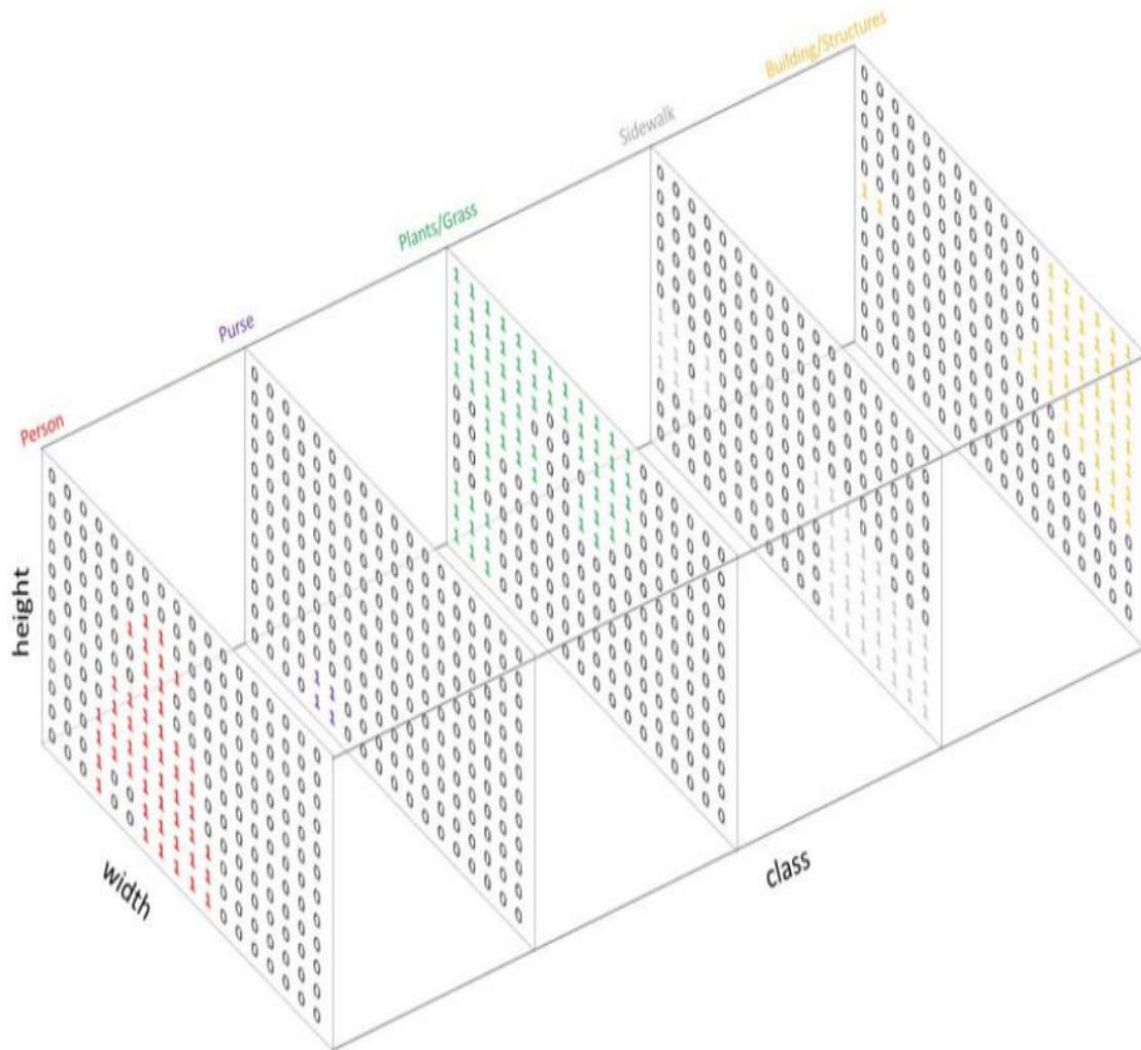
- 1: Person
- 2: Purse
- 3: Plants/Grass
- 4: Sidewalk
- 5: Building/Structures

3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	4	5	5	5
4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4

Semantic Labels

각 클래스의 값들이 배치

# Mask Rcnn

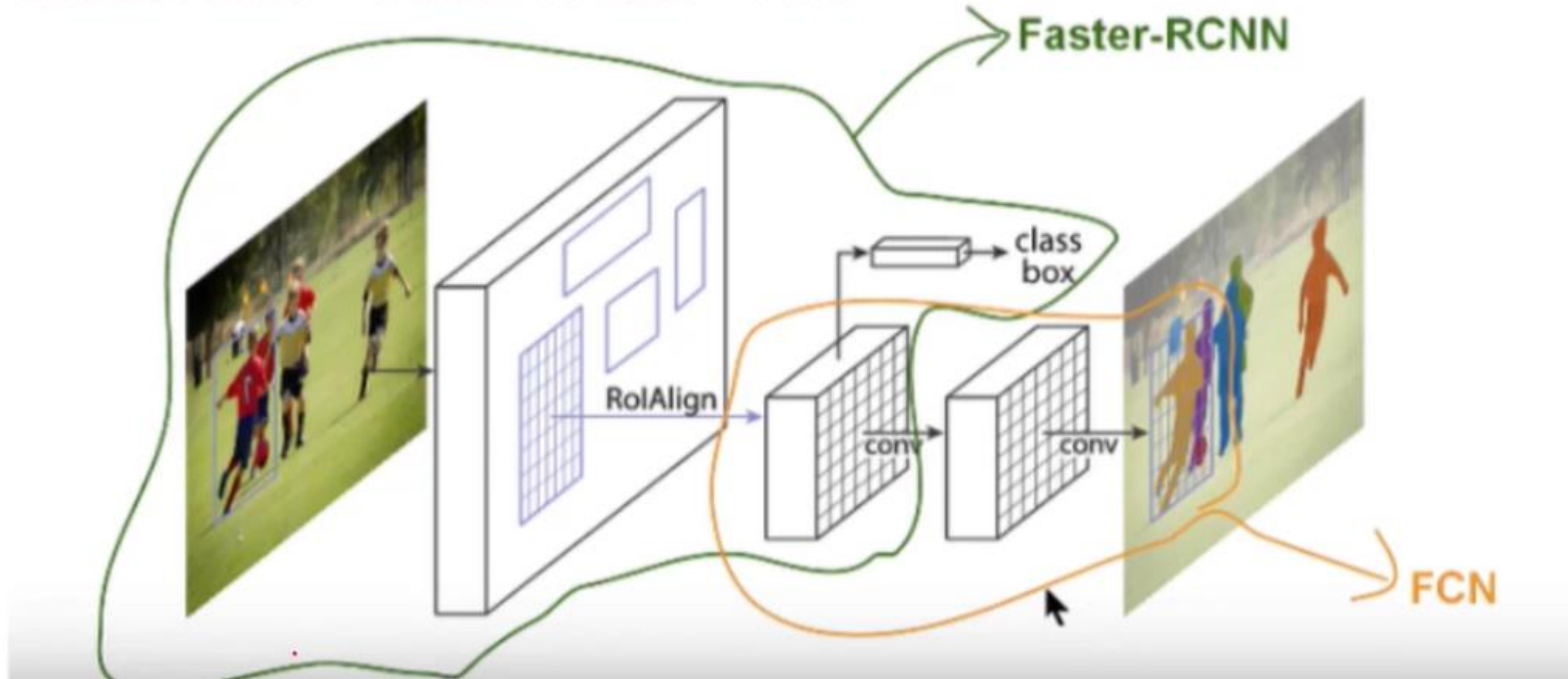


**각 채널에 각 이미지 별로 분류**

# Mask Rcn

Faster Rcn의 경우 RoIpooling을 사용  
Segmentation 의 경우 RoIpooling을 사용하면 정확도가 떨어지게 되어 RoIAlign사용

Mask R-CNN → Faster R-CNN + FCN

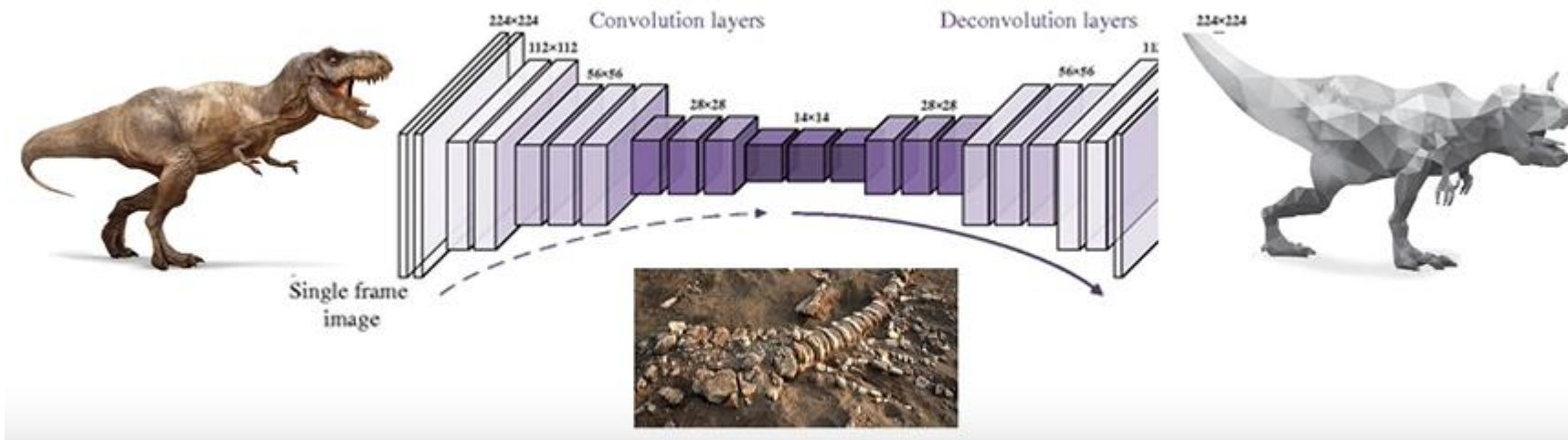


# Mask Rcn - FCN

## FCN - Fully Convolutional Network for Semantic Segmentation

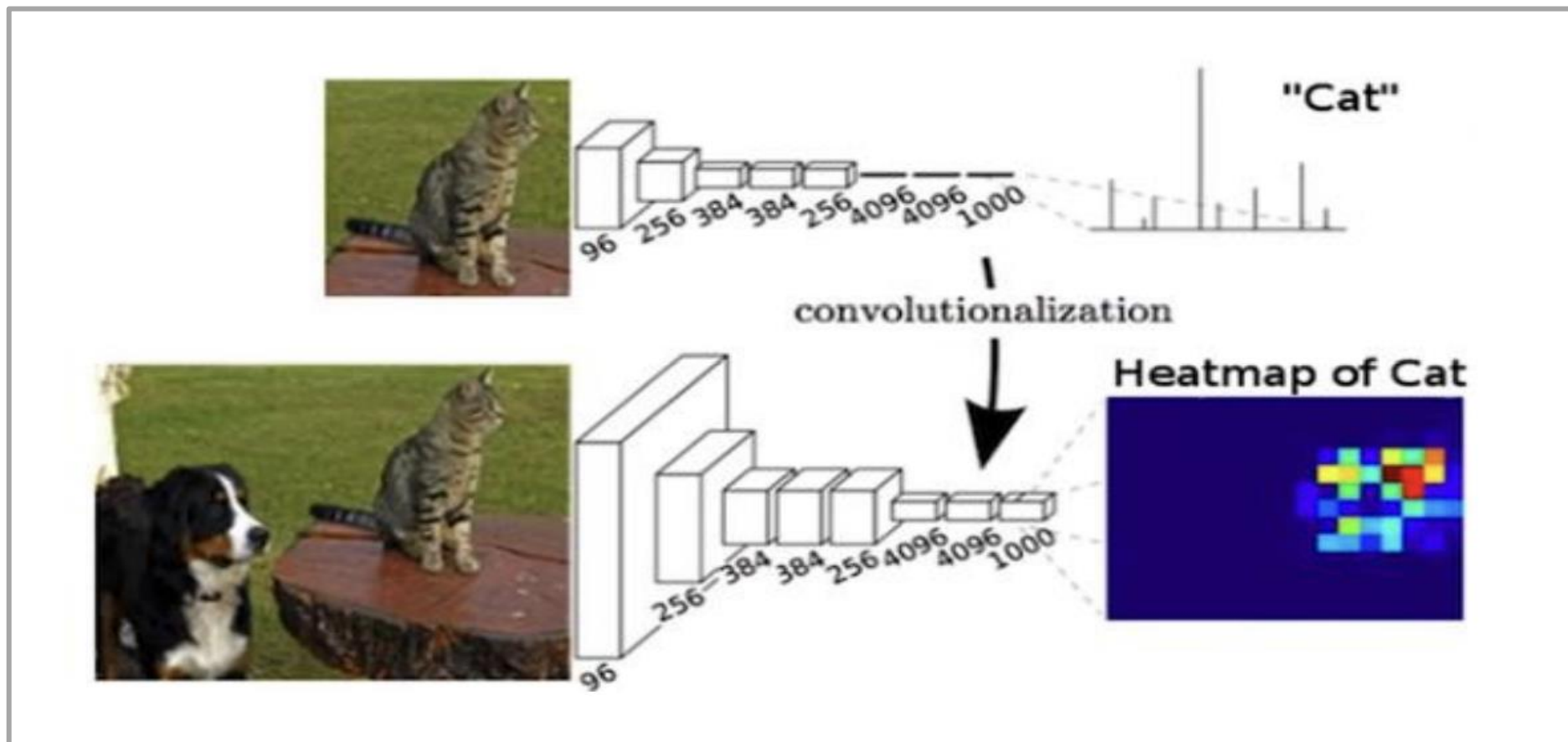
### Semantic Segmentation Encoder-Decoder Model

원본 이미지를 Convolution으로 차원축소하여 응축된 정보를 가지고, 이를 다시 복원하면서 필요한 정보를 학습  
이렇게 학습된 정보를 기반으로 segmentation수행



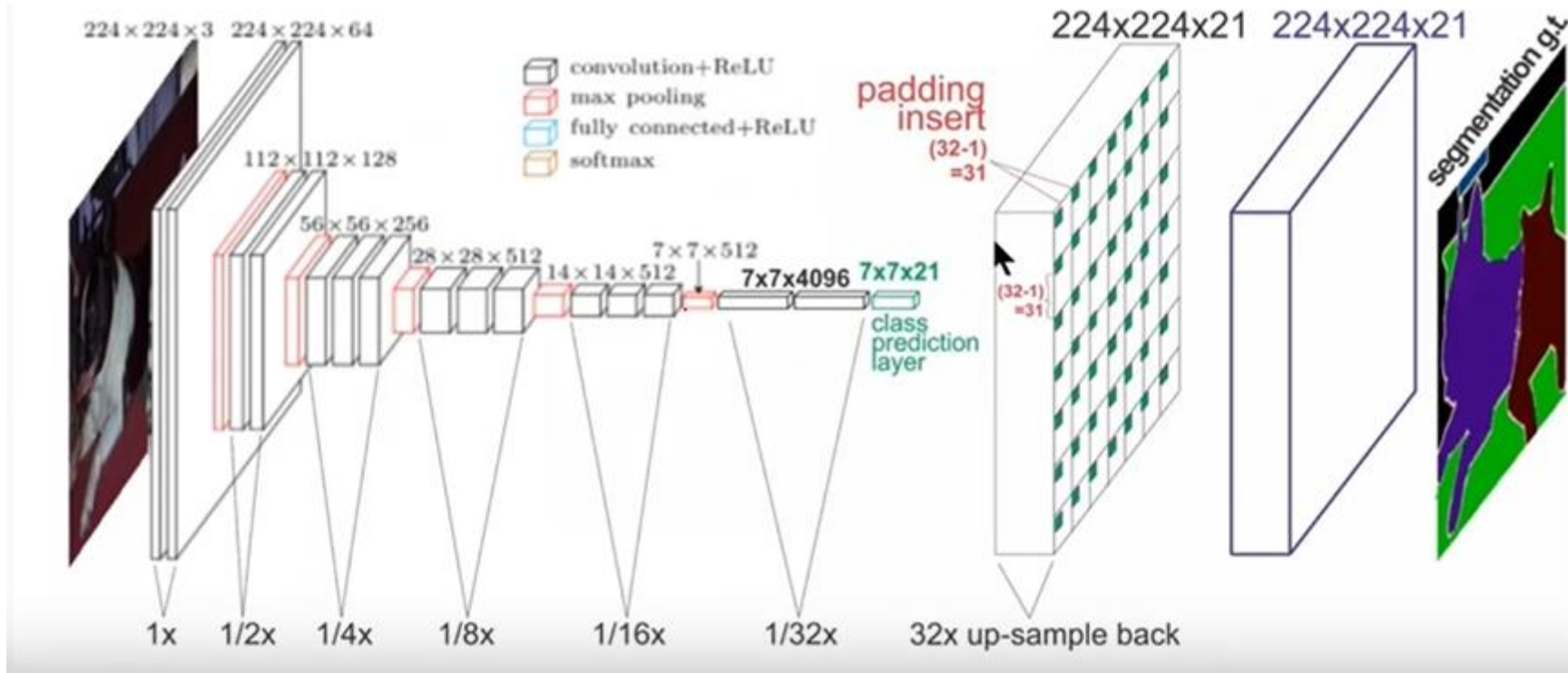


# Mask Rcnn - FCN

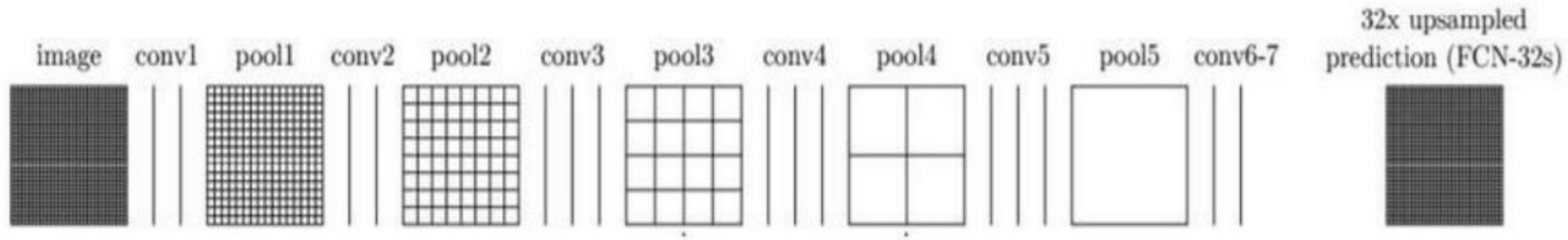


# Mask Rcnn - FCN

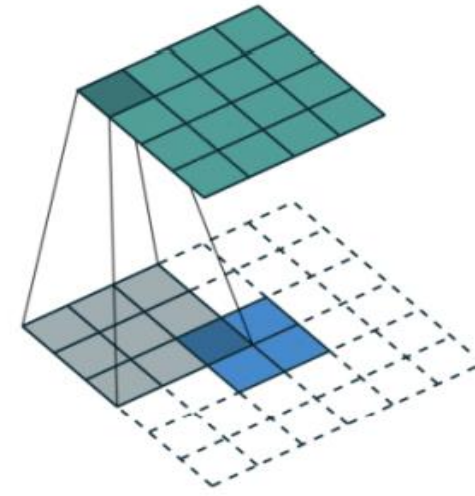
## FCN Down sampling과 Upsampling



# Mask Rcnn - FCN



De-Convolution을 통한  
Upsampling



Ground truth



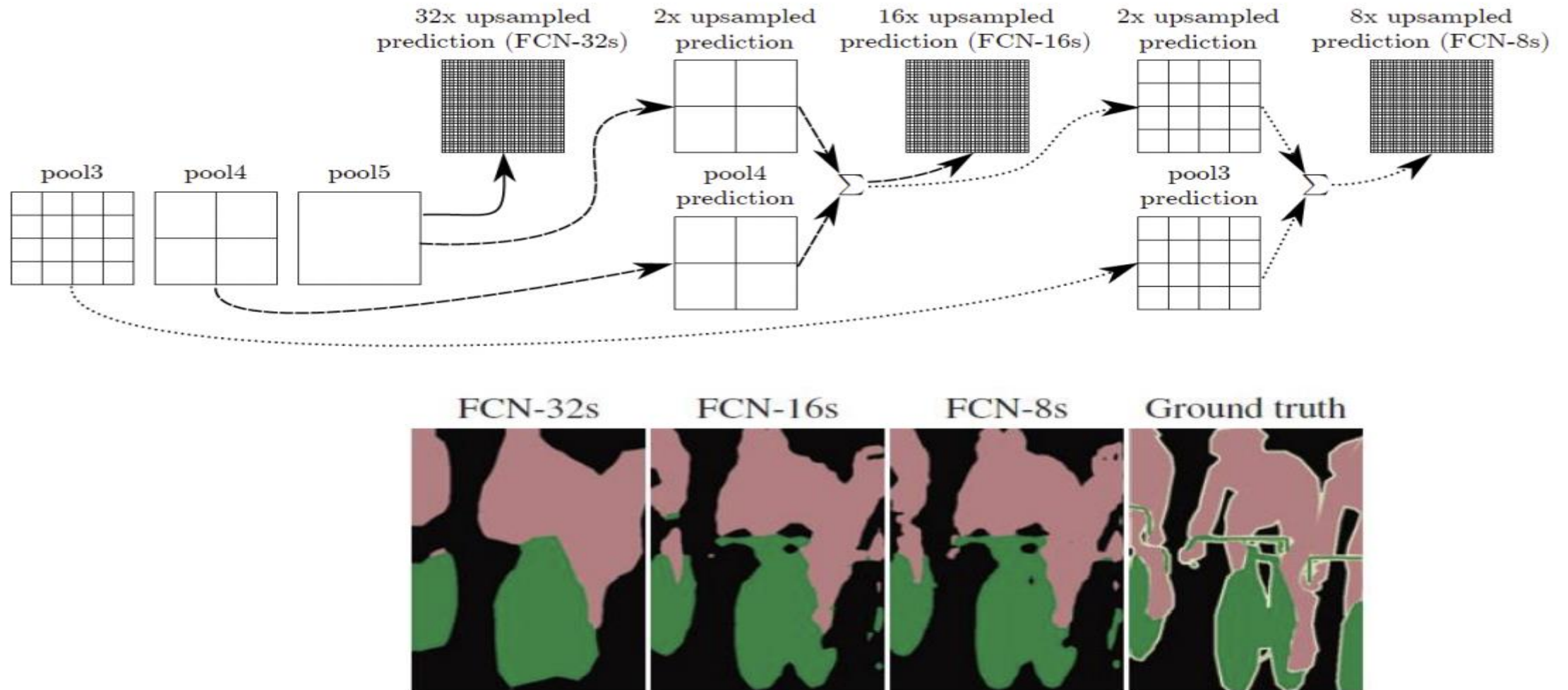
FCN-32s



**FCN32를 바로 적용할 경우 Segmentation이 정확하게는 이루어지지 않는다.**



# Mask Rcnn - FCN



32 - 16 - 8FCN의 결과는 위와 같다  
FCN8이 Ground Truth와 가장 유사하게 Segmentation이 되고 있다.

# Mask Rcnn - FCN-32s/16s/8s 별 성능

FCN-32s



FCN-16s



FCN-8s

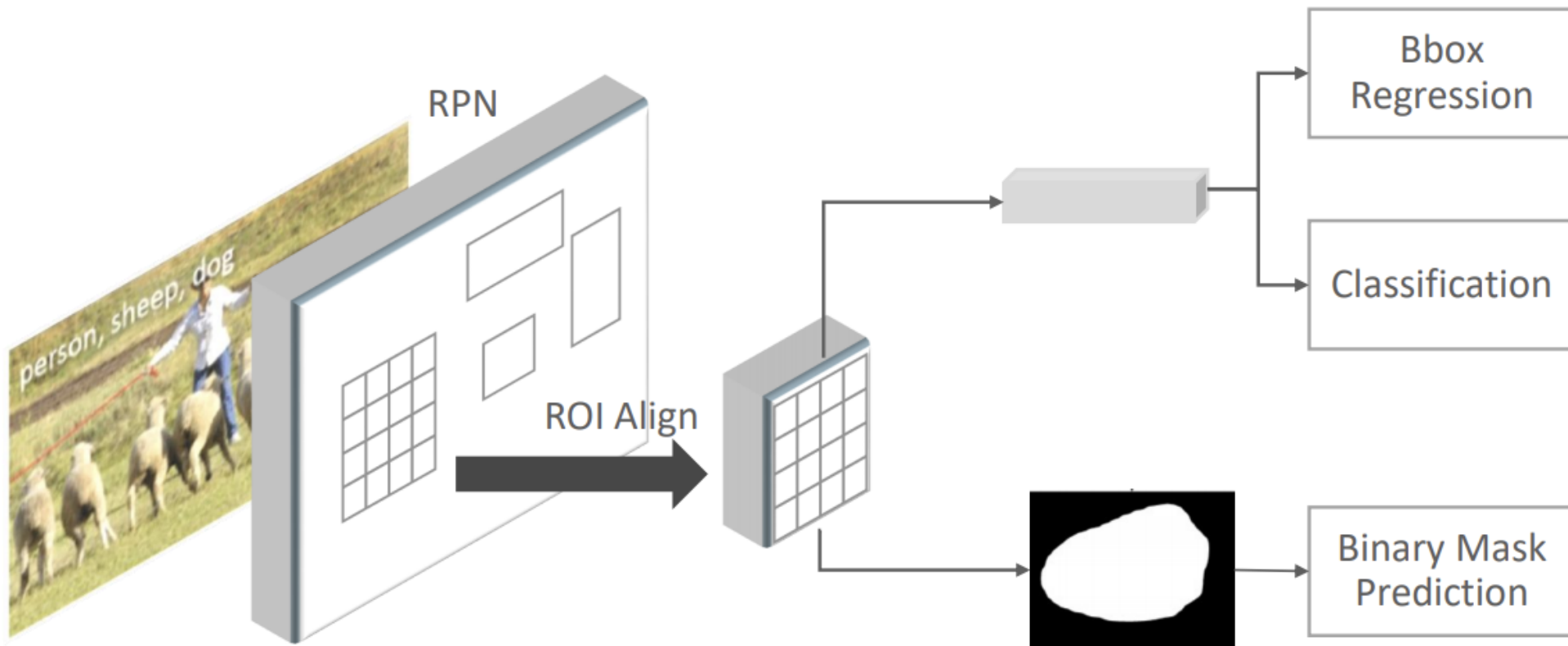


정답(Ground truth)



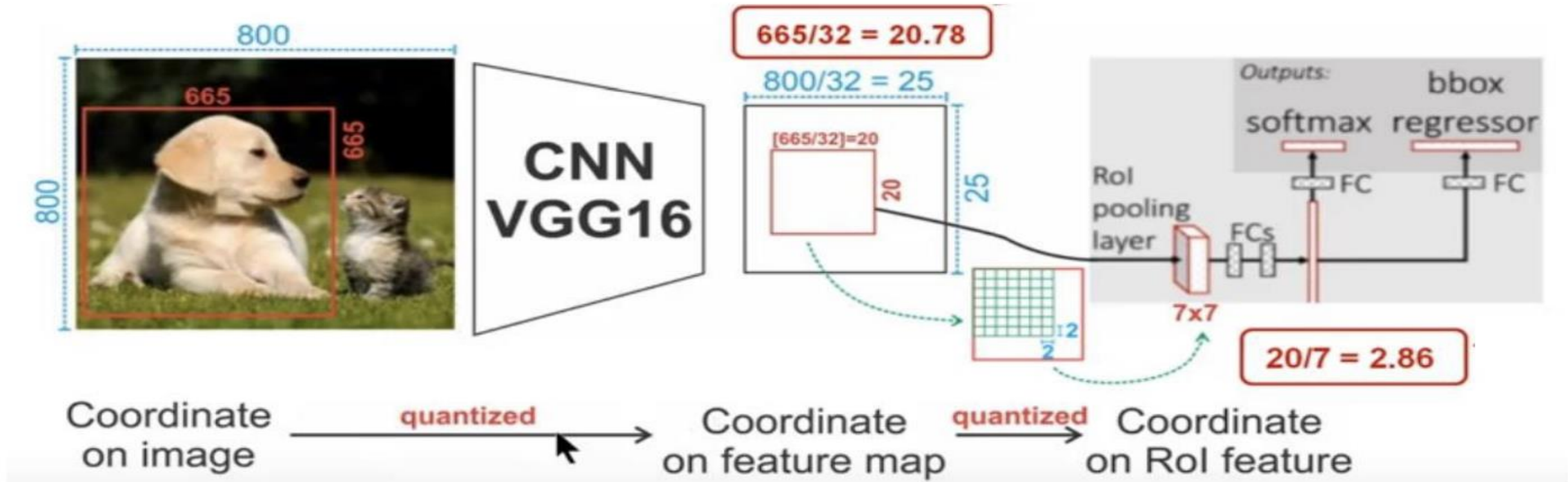
	FCN-32s	FCN-16s	FCN-8s
IoU (Intersection over Union)	59.4(%)	62.4(%)	<b>62.7(%)</b>

# Mask Rcnn 구조



새로 등장

# Segmentation 에서 ROI Pooling 문제점



기존 Faster Rcn에서 ROI Pooling을 사용하여 Detection을 실행  
Segmentation의 경우 좀 더 정확성을 요구하므로 ROI Align을 사용

# Segmentation 에서 ROI Pooling 문제점

Feature Map

0.3	0.4	0.2	0.1
0.5	0.1	0.9	0.7
0.3	0.6	0.2	0.2
0.1	0.7	0.9	0.1

## ROI Pooling

ROI 영역

0.3	0.4	0.2	0.1
0.5	0.1	0.9	0.7
0.3	0.6	0.2	0.2
0.1	0.7	0.9	0.1

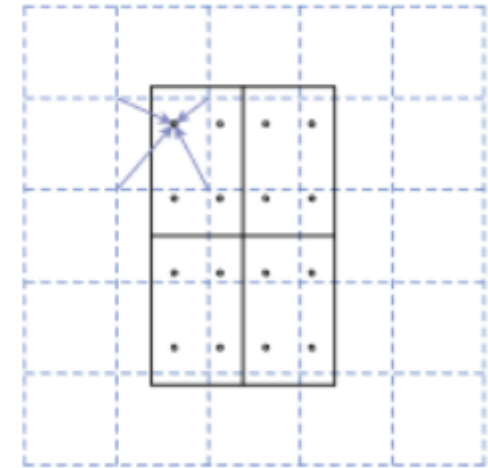
2x2

0.3	0.4	0.2	0.1
0.5	0.1	0.9	0.7
0.3	0.6	0.2	0.2
0.1	0.7	0.9	0.1

2X2 ROI (Max) Pooling

0.9	0.7
0.9	0.1

## ROI-Align



**ROI Pooling의 경우 위와 같은 경우 2X2를 적용할 경우 정확하게 나누어 구하기 어렵다.**

**ROI Align의 경우 기존 그리드에 구애받지 않고 소수점 그대로 매핑**



**~ing**