Mask R-CNN

Kaiming He et al, ICCV 2017(Oral)

1. Abstract & Introduction

This paper efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance.

The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.

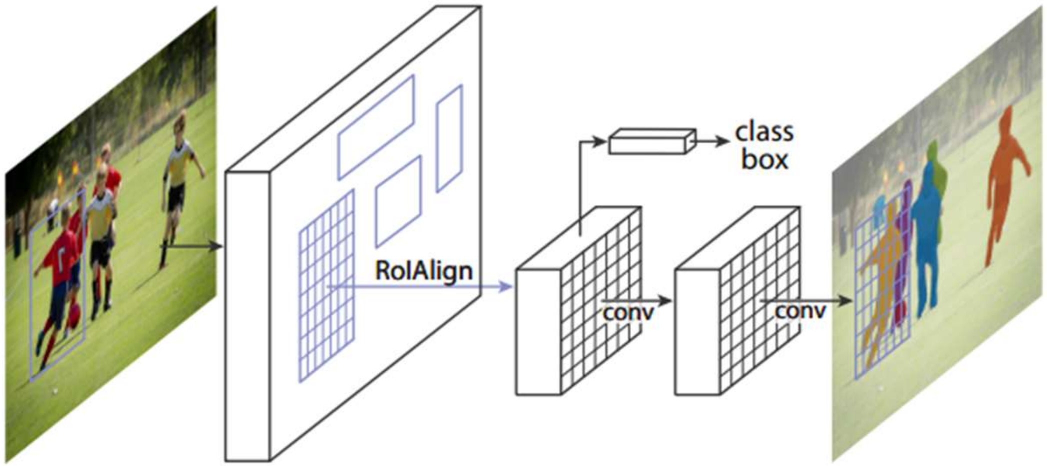


Figure 1. The **Mask R-CNN** framework for instance segmentation.

1. Abstract & Introduction

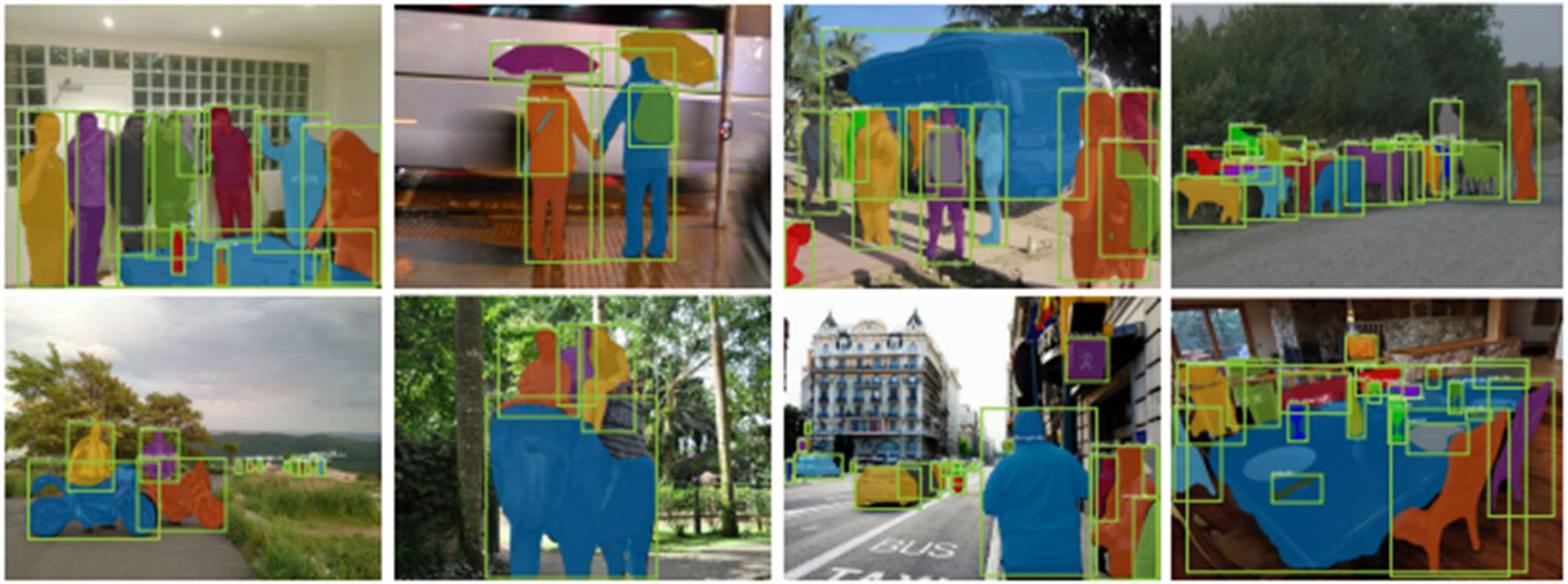
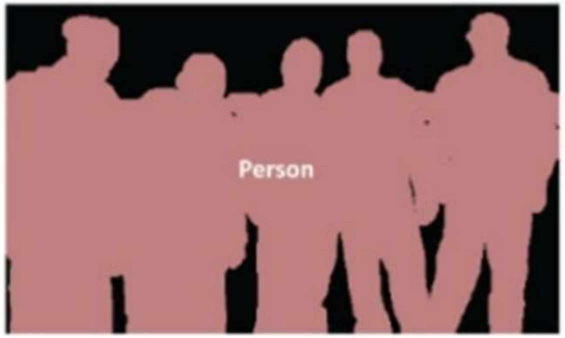


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

2. Reviews – Instance Segmentation

Object Detection



Semantic Segmentation

Instance Segmentation

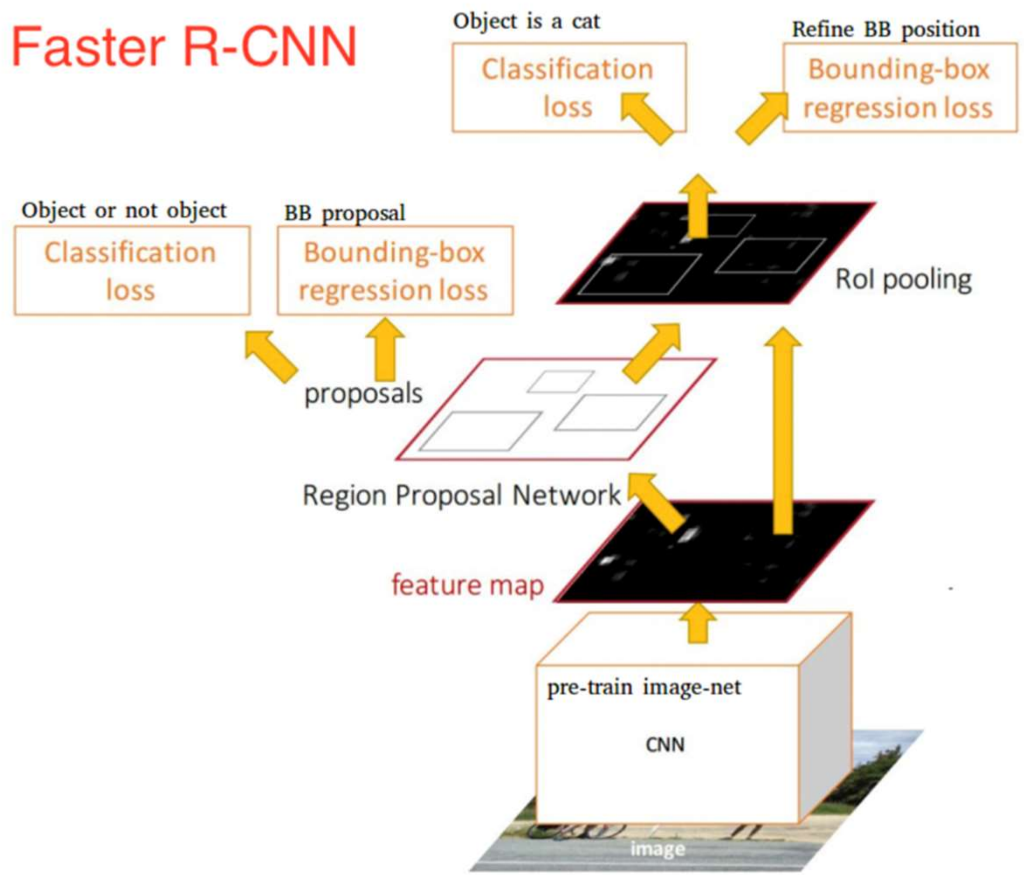
Faster R-CNN

FCN

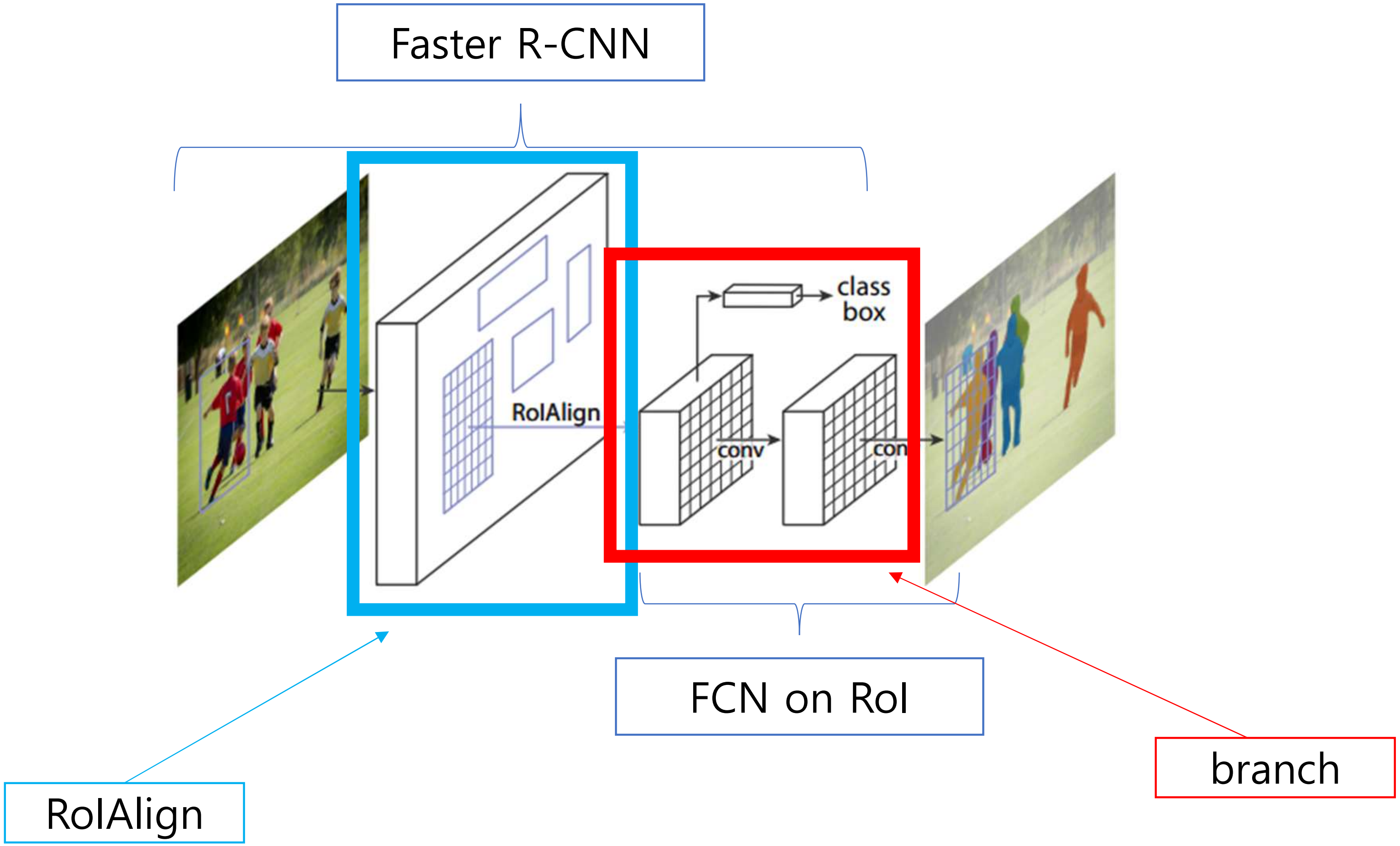
?

BBox Classification + Segmentation Classification = Segmentation in BBox Classification

2. Reviews – Faster R-CNN



3. Mask R-CNN - Architecture



3. Mask R-CNN – Architecture

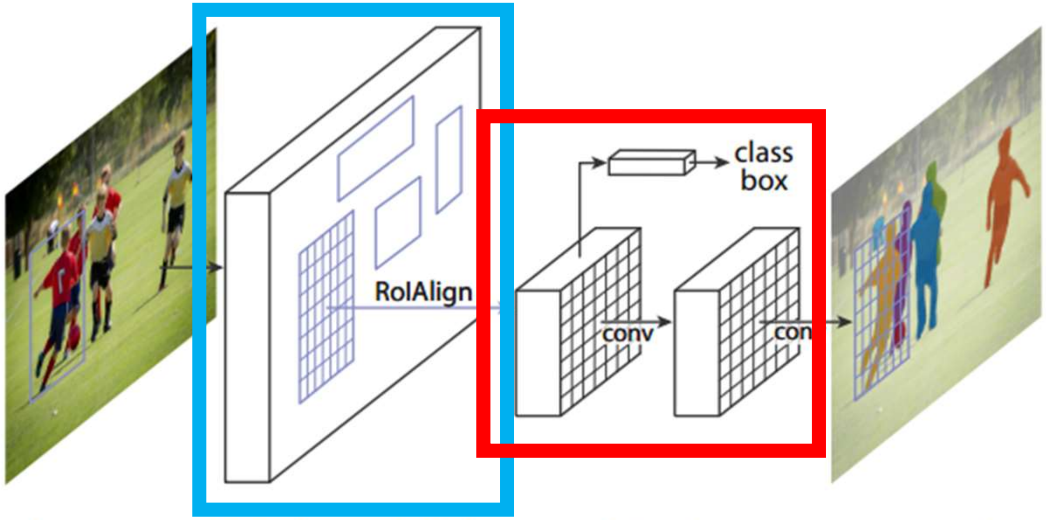
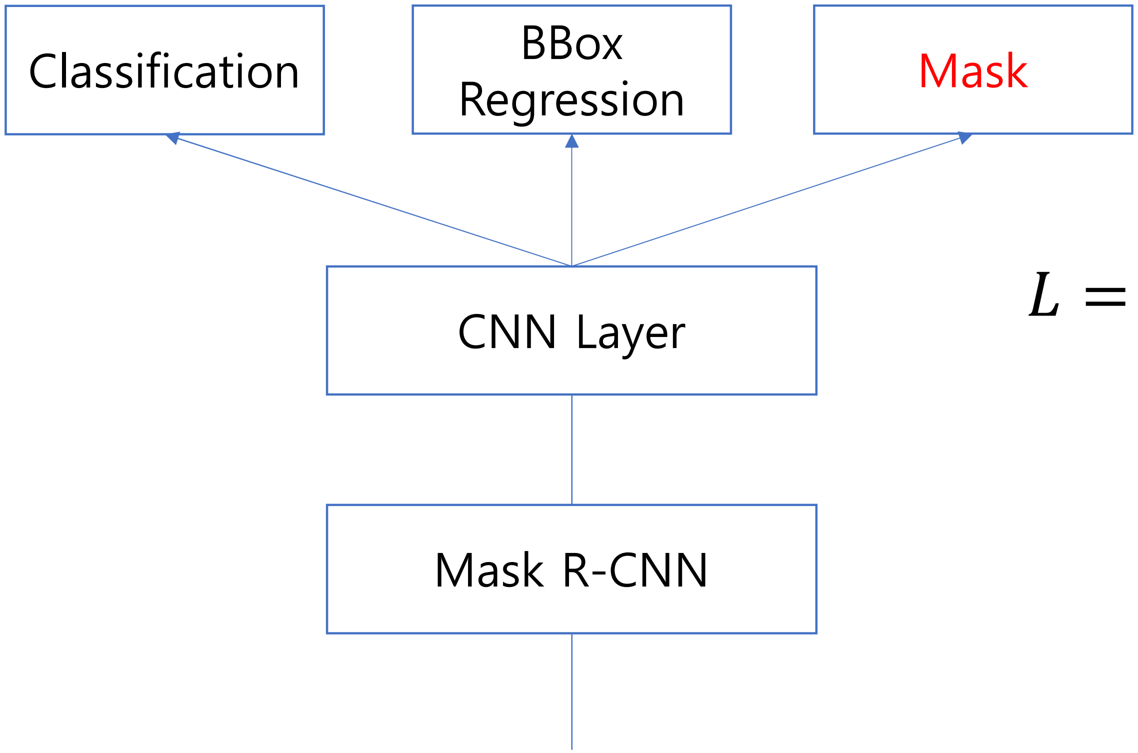


Figure 1. The **Mask R-CNN** framework for instance segmentation.

1. To fix the misalignment, we propose a simple, quantization-free layer called RoIAlign, that faithfully preserves exact spatial locations.

2. Adding a branch for predicting segmentation masks on eah Region of Interest (RoI), in parallel with the existing branch for classification and bbox regression.

3. Mask R-CNN - Loss



$$L = L_{cls} + L_{box} + L_{mask}$$

L_{cls} : Softmax Cross Entropy
L_{cls} : Regression
L_{cls} : Binary Cross Entropy

3. Mask R-CNN - Loss

$$L = L_{cls} + L_{box} + L_{mask}$$

A.

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

p : *Predicted Class*

u : *GT Class*

t^u : *Predicted Bounding Box for class u*

v : *GT Bounding Box*

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

Log loss

$$L_{loc}(t^u,v) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i^u - v_i)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Smooth L1 loss

3. Mask R-CNN - Loss

$$L = L_{cls} + L_{box} \boxed{+ L_{mask}}$$

B.

- $K \cdot (m \times m)$ sigmoid outputs:
  - → pixel-wise binary classification
  - → one mask for each class, no competition
- L_{mask}: mean binary cross-entropy

	AP	AP_{50}	AP_{75}
softmax	24.8	44.1	25.1
sigmoid	**30.3**	**51.2**	**31.5**
	+5.5	*+7.1*	*+6.4*

(b) **Multinomial *vs.* Independent Masks** (ResNet-50-C4): *Decoupling* via per-class binary masks (sigmoid) gives large gains over multinomial masks (softmax).

3. Mask R-CNN - RoIAlign

We propose an RoIAlign layer that removes the harsh quantization of RoIPool, properly aligning the extracated features with the input.

RoIAlign improves mask accuracy by relative 10% to 50%, showing bigger gains under stricter localization metrics.

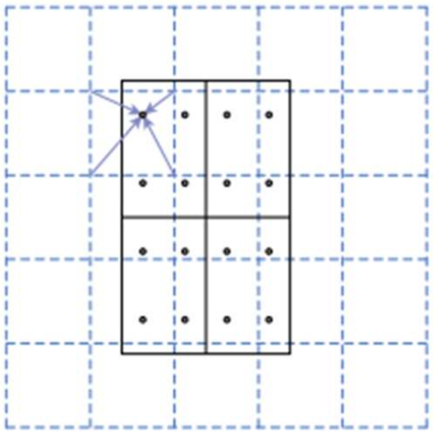
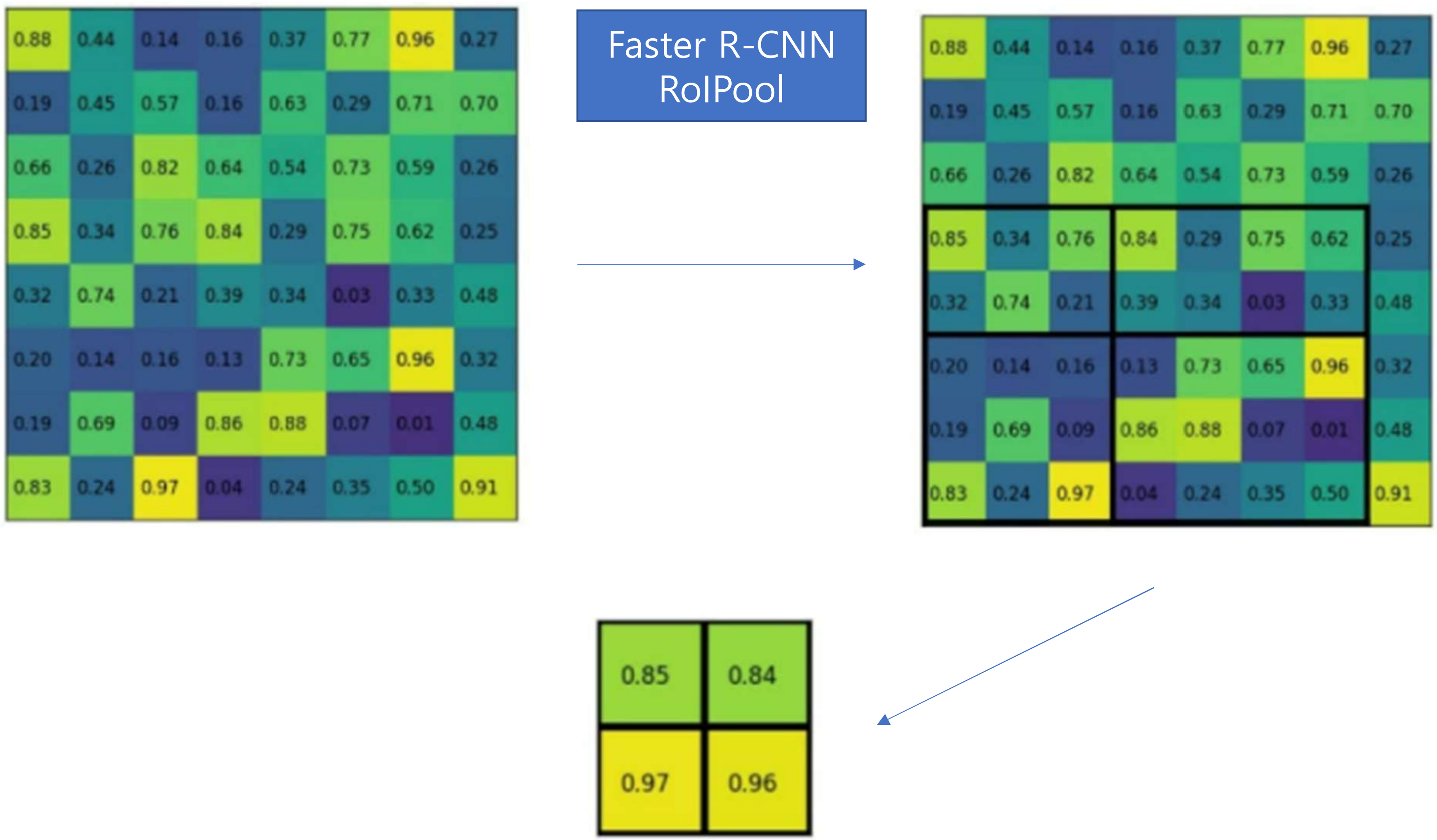
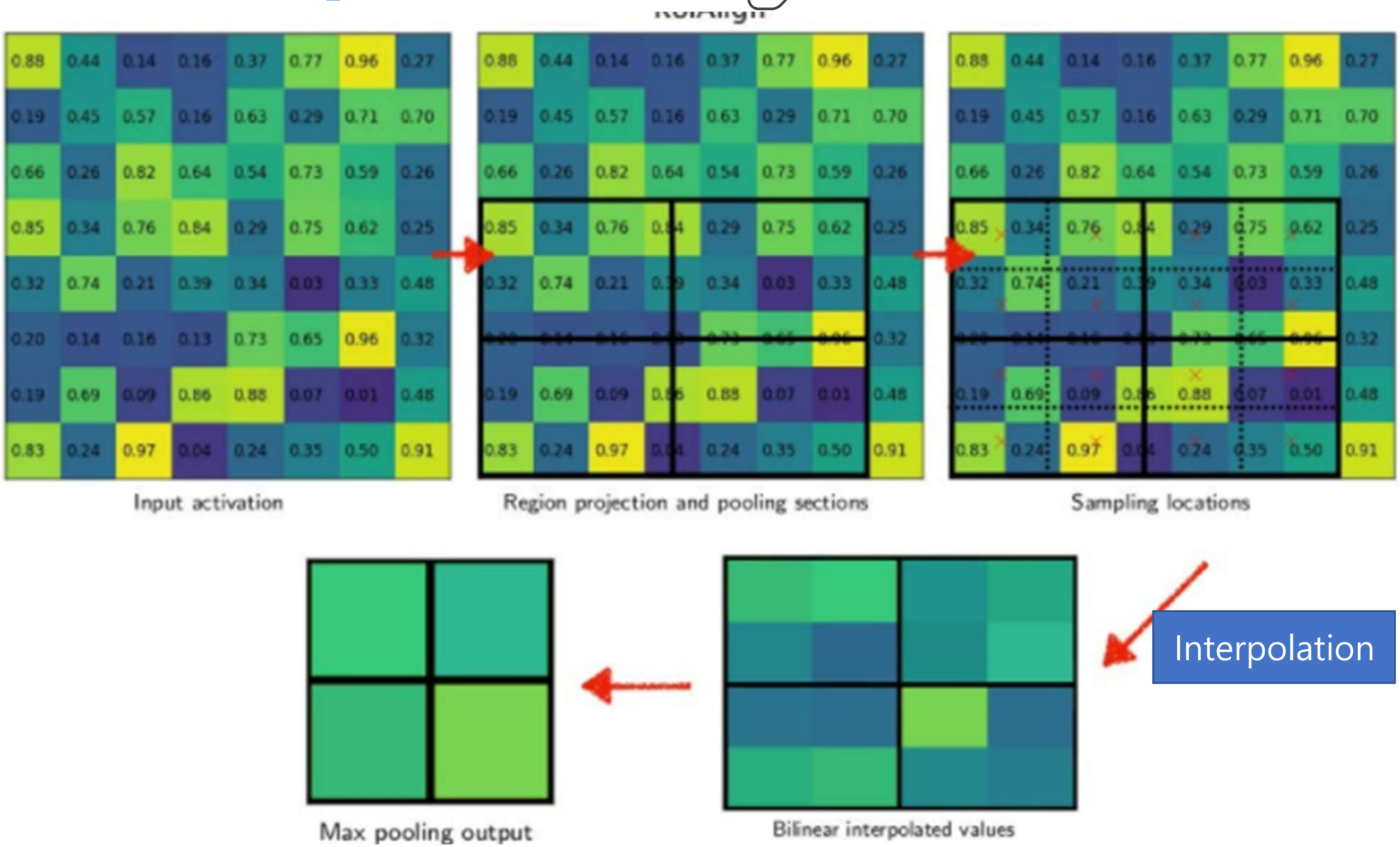


Figure 3. **RoIAlign:** The dashed grid represents a feature map, the solid lines an RoI (with 2×2 bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points.

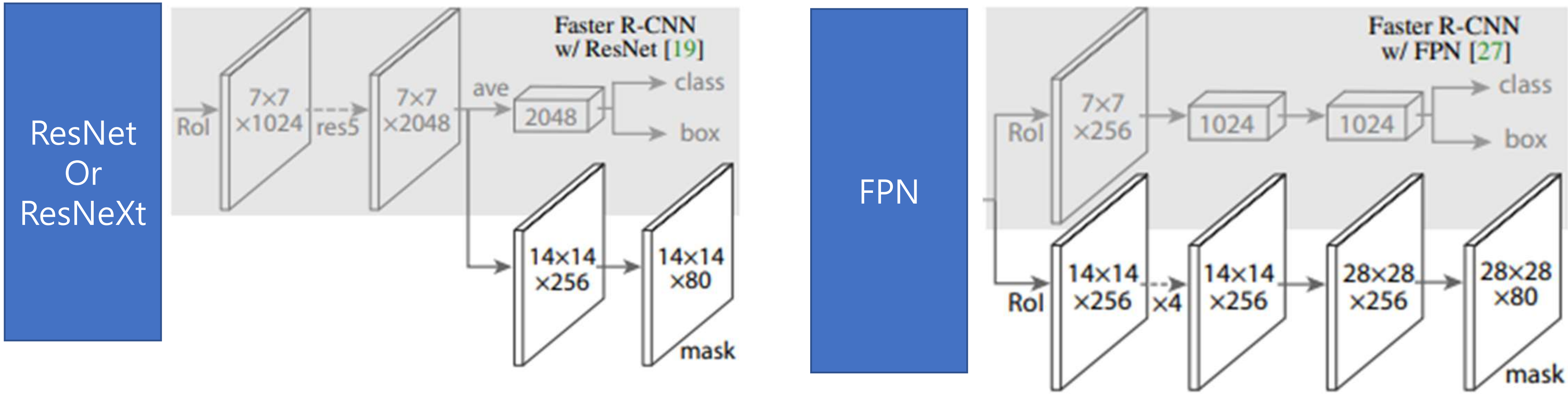
3. Mask R-CNN - RoIAlign



3. Mask R-CNN - RoIAlign



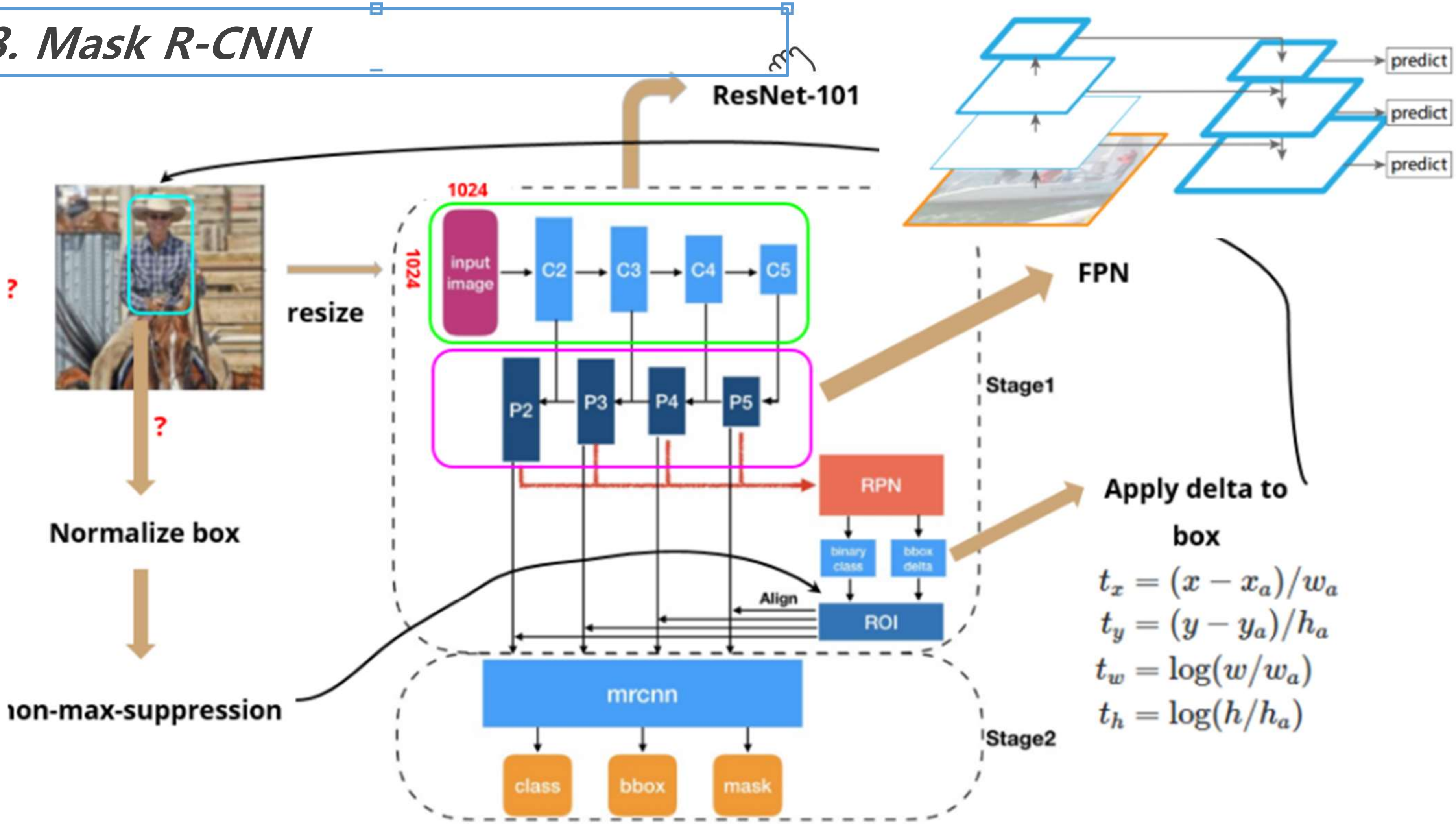
3. Mask R-CNN - Backbone



3. Mask R-CNN - Backbone

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

3. Mask R-CNN



Mask R-CNN architecture

$$t_x = (x - x_a)/w_a$$
$$t_y = (y - y_a)/h_a$$
$$t_w = \log(w/w_a)$$
$$t_h = \log(h/h_a)$$

4. Result

net-depth-features	AP	AP_{50}	AP_{75}
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	**36.7**	**59.5**	**38.9**

(a) **Backbone Architecture**: Better backbones bring expected gains: deeper networks do better, FPN outperforms C4 features, and ResNeXt improves on ResNet.

	AP	AP_{50}	AP_{75}
softmax	24.8	44.1	25.1
sigmoid	**30.3**	**51.2**	**31.5**
	+5.5	*+7.1*	*+6.4*

(b) **Multinomial *vs.* Independent Masks** (ResNet-50-C4): *Decoupling* via per-class binary masks (sigmoid) gives large gains over multinomial masks (softmax).

	align?	bilinear?	agg.	AP	AP_{50}	AP_{75}
RoIPool [12]			max	26.9	48.8	26.4
RoIWarp [10]		✓	max	27.2	49.2	27.1
		✓	ave	27.1	48.9	27.1
RoIAlign	✓	✓	max	**30.2**	**51.0**	**31.8**
	✓	✓	ave	**30.3**	**51.2**	**31.5**

(c) **RoIAlign** (ResNet-50-C4): Mask results with various Ro layers. Our RoIAlign layer improves AP by ~3 points an AP_{75} by ~5 points. Using proper alignment is the only fac tor that contributes to the large gap between RoI layers.

	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

	mask branch	AP	AP_{50}	AP_{75}
MLP	fc: $1024 \rightarrow 1024 \rightarrow 80 \cdot 28^2$	31.5	53.7	32.8
MLP	fc: $1024 \rightarrow 1024 \rightarrow 1024 \rightarrow 80 \cdot 28^2$	31.5	54.0	32.6
FCN	conv: $256 \rightarrow 256 \rightarrow 256 \rightarrow 256 \rightarrow 256 \rightarrow 80$	**33.6**	**55.2**	**35.3**

4. Result

	training data	AP [val]	AP	AP_{50}	person	rider	car	truck	bus	train	mcycle	bicycle
InstanceCut [23]	fine + coarse	15.8	13.0	27.9	10.0	8.0	23.7	14.0	19.5	15.2	9.3	4.7
DWT [4]	fine	19.8	15.6	30.0	15.1	11.7	32.9	17.1	20.4	15.0	7.9	4.9
SAIS [17]	fine	-	17.4	36.7	14.6	12.9	35.7	16.0	23.2	19.0	10.3	7.8
DIN [3]	fine + coarse	-	20.0	38.8	16.5	16.7	25.7	20.6	30.0	23.4	17.1	10.1
SGN [29]	fine + coarse	29.2	25.0	44.9	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4
Mask R-CNN	fine	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
Mask R-CNN	fine + COCO	**36.4**	**32.0**	**58.1**	**34.8**	**27.0**	**49.1**	**30.1**	**40.9**	**30.9**	**24.1**	**18.7**

Reference

https://www.slideshare.net/windmdk/mask-rcnn
https://niniit.tistory.com/32
https://velog.io/@kimkj38/Point-Review-Fast-R-CNN
https://www.slideshare.net/TaeohKim4/pr057-mask-rcnn