Mask R-CNN

Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick
Facebook AI Research (FAIR)

박태우

Introduction

- 기존의 framework에서 instance segmentation이 가능하도록 목표로 함.

- Faster R-CNN에서 추가적으로 segmentation mask를 예측하는 branch를 추가.

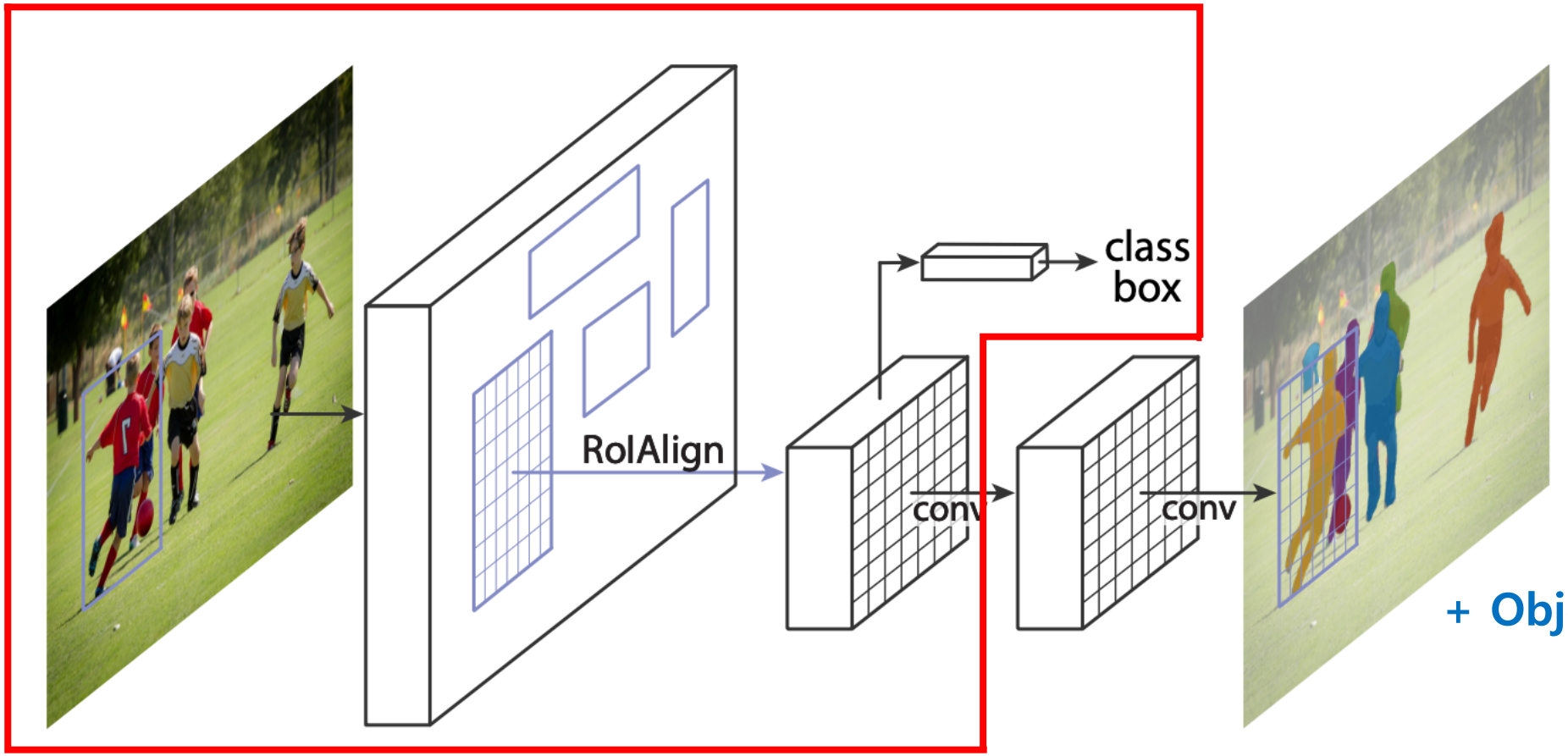
- 기존 RoIPooling의 quantization 문제를 해결하기 위해 RoIAlign을 도입함.

Mask R-CNN - Architecture

- 기존의 Faster R-CNN은 output으로 classification, bbox regression을 도출 했음.

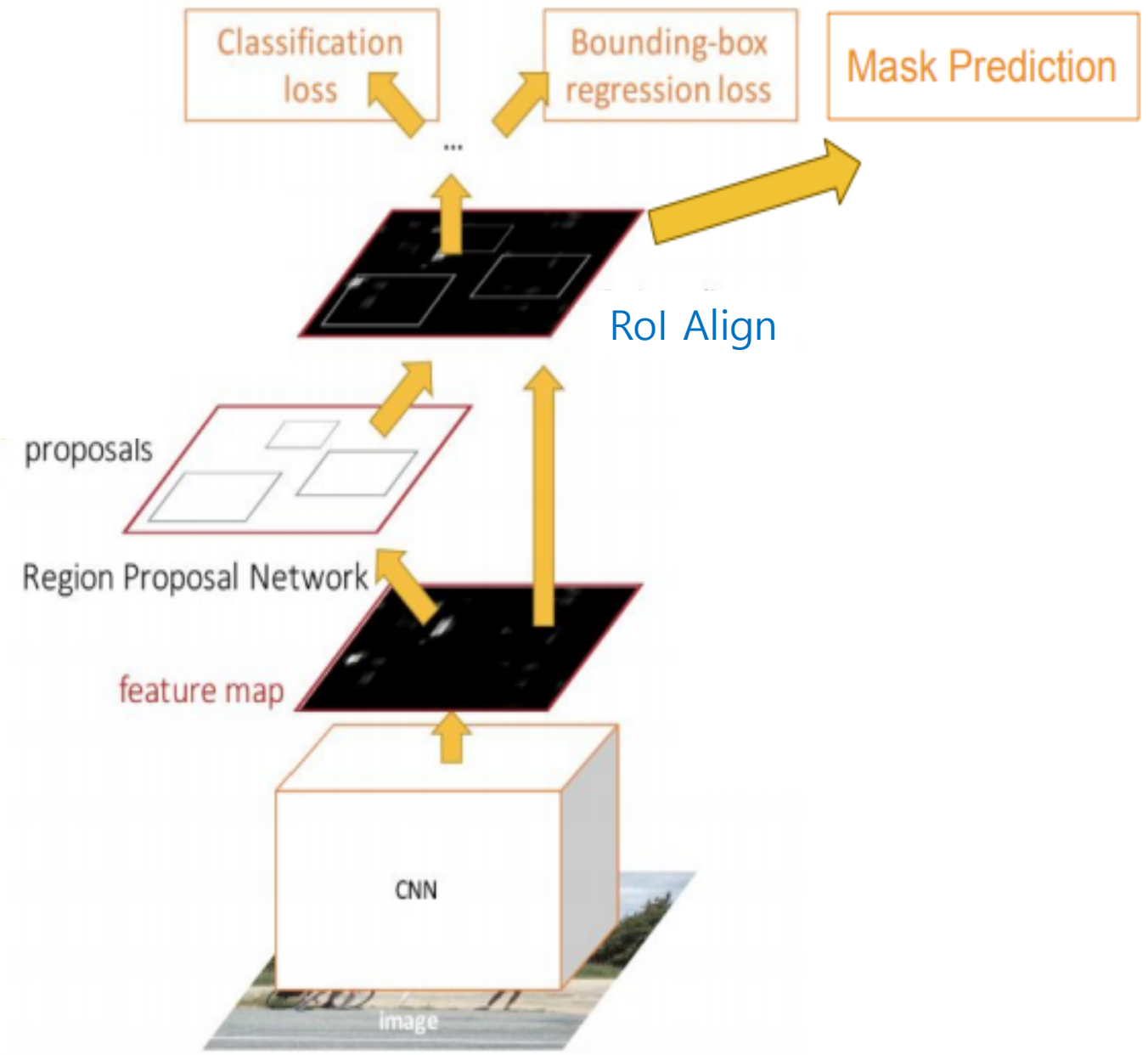
- Mask R-CNN은 Faster R-CNN에서 추가적으로 object mask를 추가함.

Faster R-CNN



+ Object mask

Mask R-CNN - Architecture



Mask R-CNN - Architecture

CNN
+RPN

RoI Align

Classification Scores: C
Box coordinates (per class): 4 * C

Fully Connected Layer

Conv

Conv

256 x 14 x 14

256 x 14 x 14

Fully Conv Layer

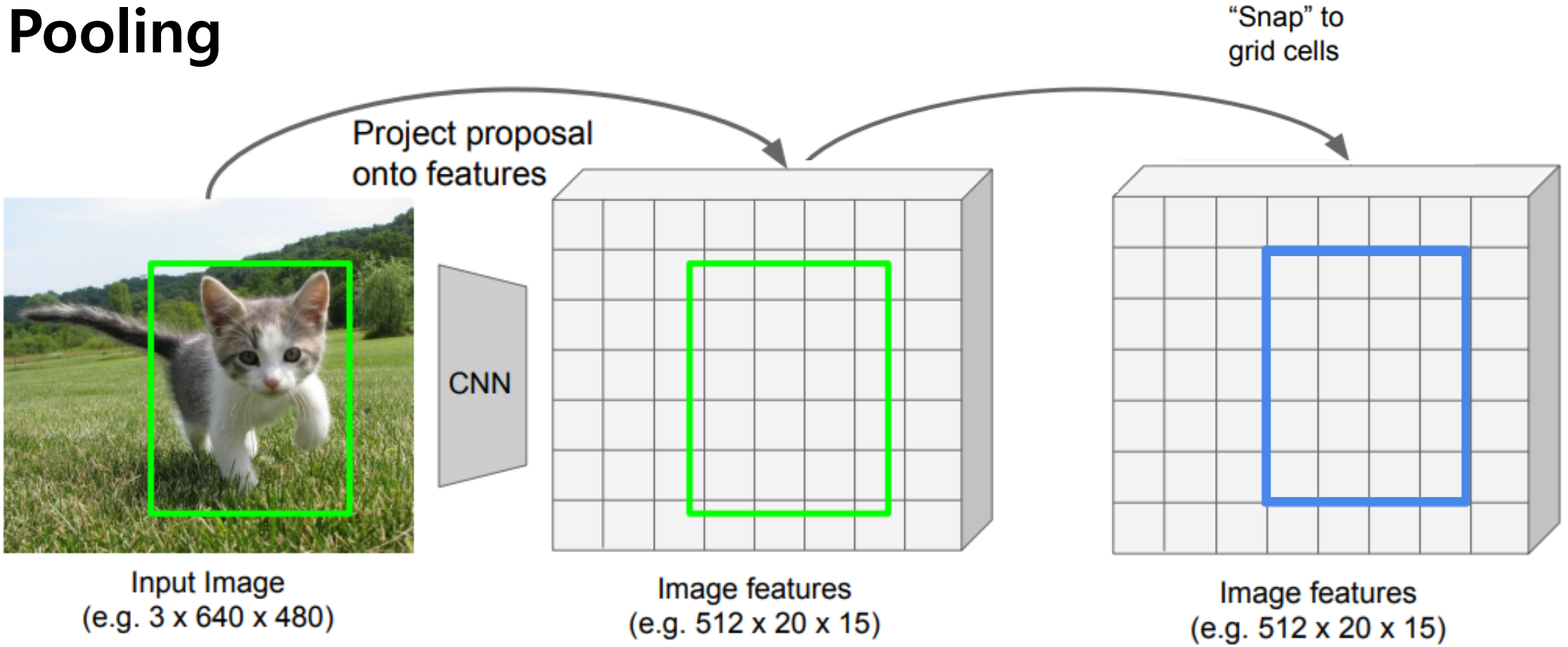
Predict a mask for each of C classes

C x 28 x 28

Class별로 mask 도출

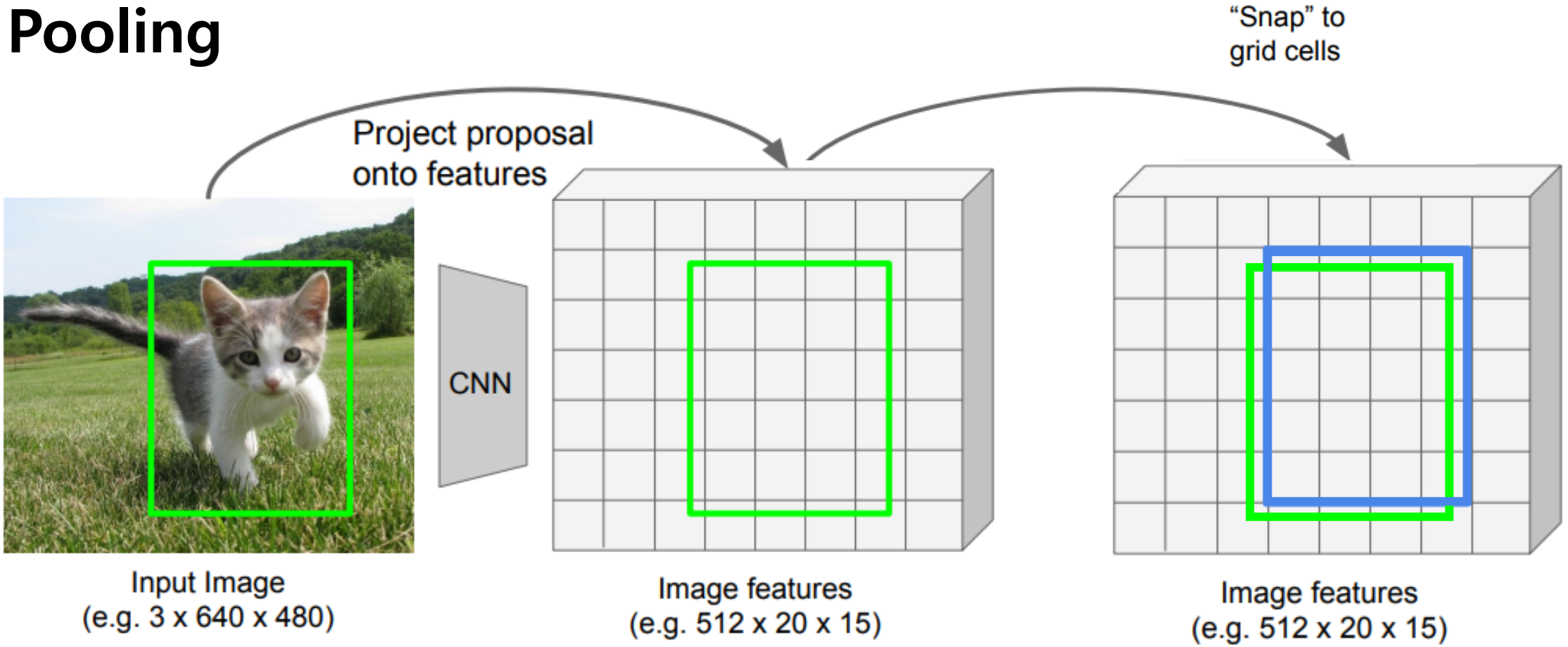
Mask R-CNN – RoI Align

RoI Pooling



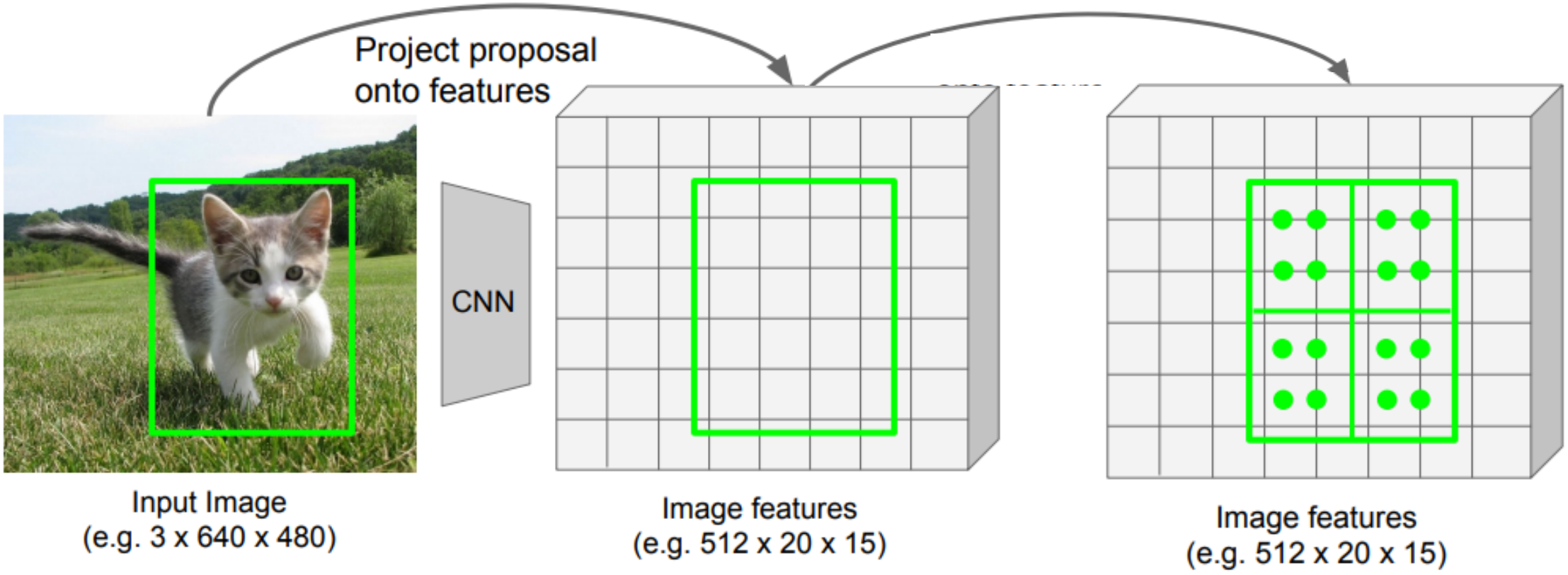
Mask R-CNN – RoI Align

RoI Pooling



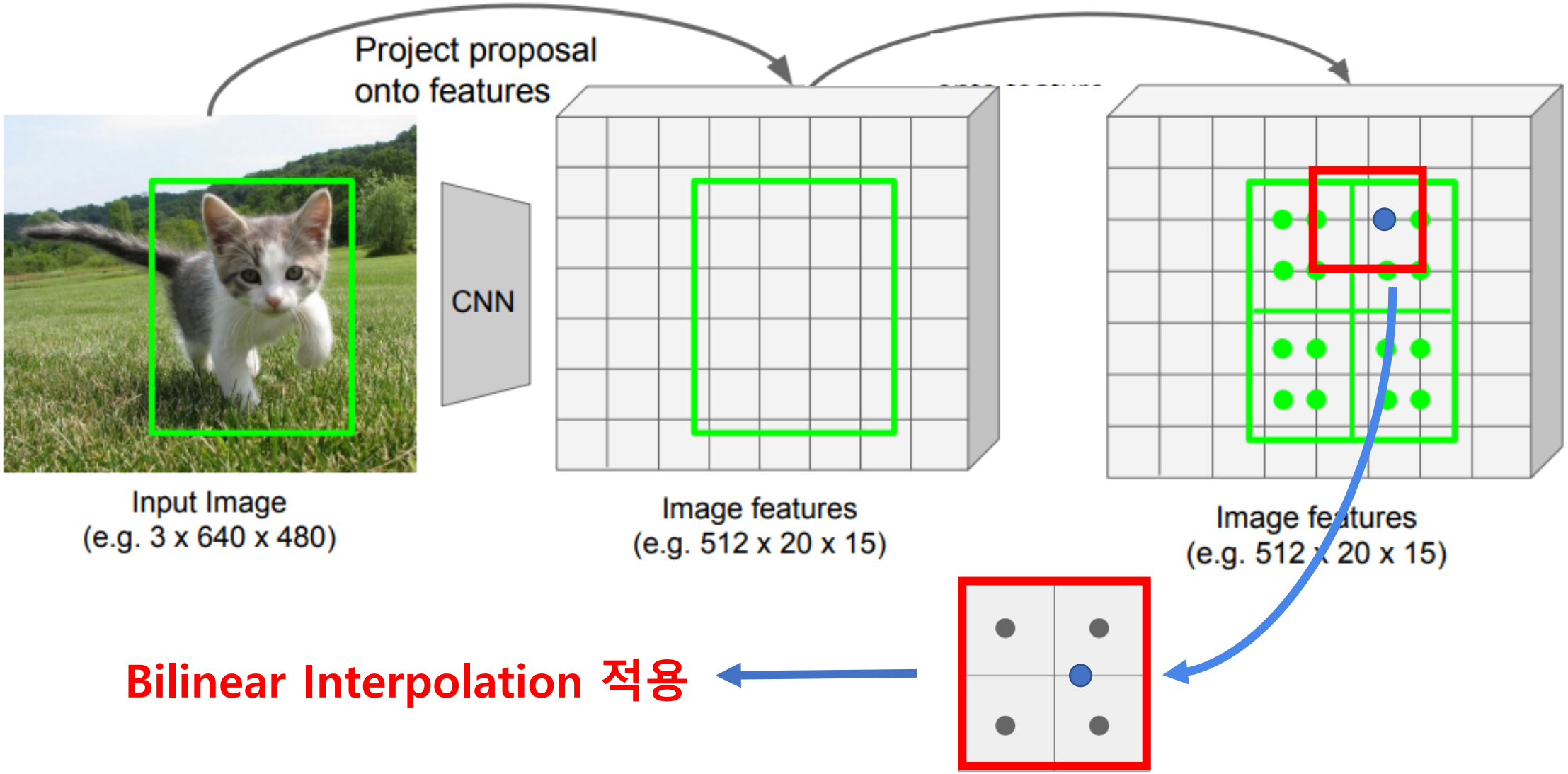
Mask R-CNN – RoI Align

RoI Align



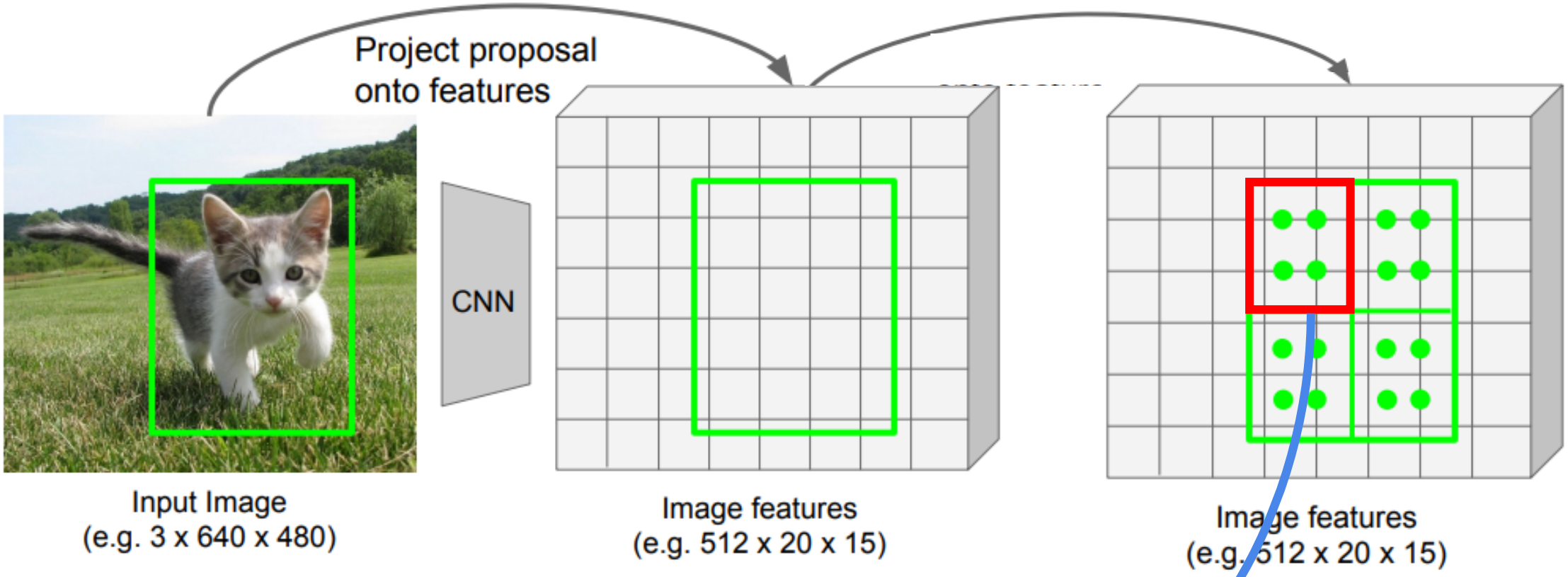
Mask R-CNN – RoI Align

RoI Align



Mask R-CNN – RoI Align

RoI Align



Mask R-CNN – Mask resize

Top score Class

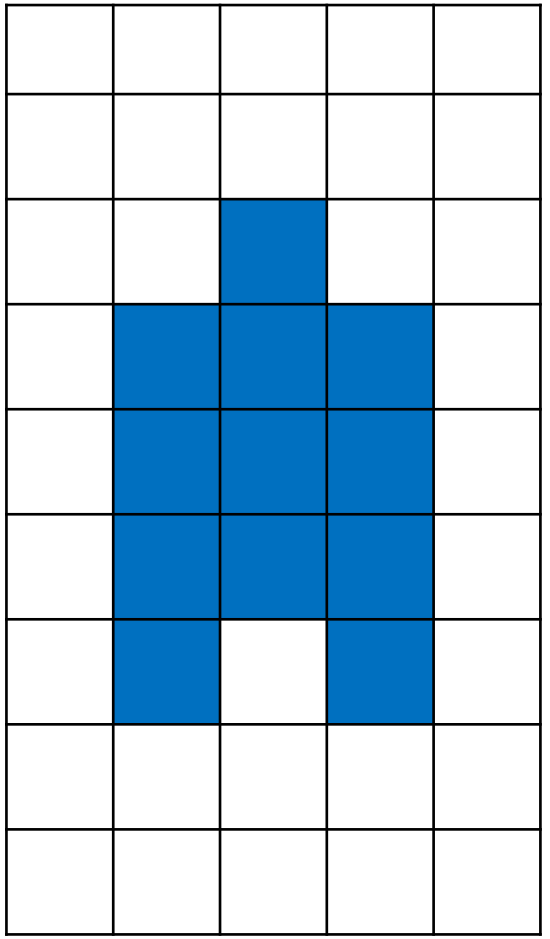
0.1	0.1	0	0.2	0.2	0.1	0.1
0.1	0.2	0.3	0.8	0.3	0.1	0.1
0.2	0.5	0.8	0.7	0.6	0.2	0.1
0.3	0.4	0.9	0.6	0.7	0.3	0.1
0.2	0.1	0.8	0.8	0.9	0.5	0.1
0.1	0.2	0.3	0.5	0.2	0.3	0.1
0.1	0.1	0.1	0	0	0.1	0.1

Mask Prediction

resize

0.1	0.1	0.2	0.1	0.1
0	0.2	0.3	0.1	0.1
0.2	0.3	0.7	0.4	0.1
0.1	0.6	0.9	0.7	0.3
0.2	0.9	0.7	0.7	0.4
0.4	0.8	0.6	0.6	0.2
0.3	0.5	0.3	0.8	0.3
0.1	0.2	0.1	0.2	0.1
0.1	0.1	0.1	0	0.1

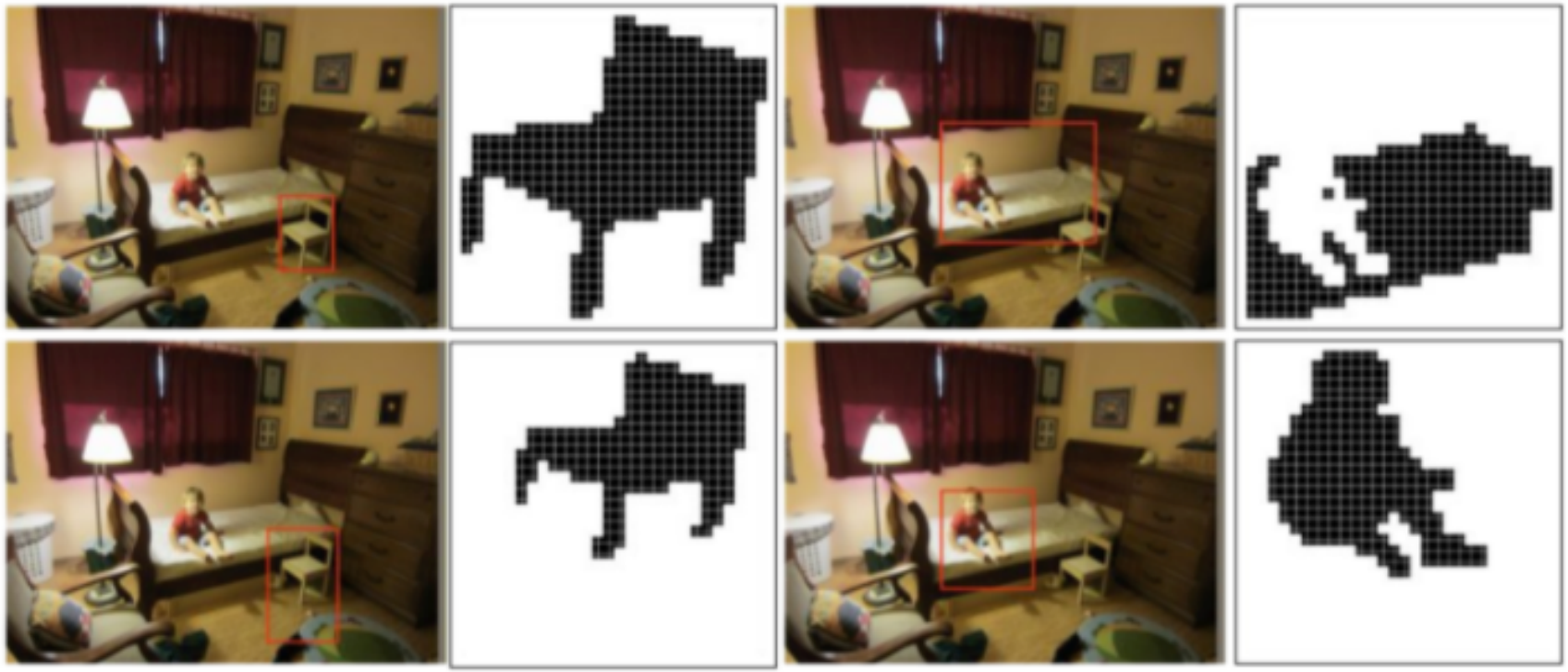
Threshold
0.5



Mask of Input Image

Mask R-CNN

- 최근 연구들에서는 classification이 mask prediction에 영향을 줌.
- 하지만 Mask R-CNN의 경우 mask prediction은 classification과 독립적으로 수행됨.
- Mask prediction은 해당 픽셀에 object가 존재하는지 존재하지 않는지만 나타낸다.



Mask R-CNN – Trianing Loss

- Training 과정에서 각 RoI마다 다음의 Loss를 가짐.

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

- 각 픽셀마다 sigmoid를 수행, L_{mask} 는 binary cross-entropy loss의 평균으로 계산.

- Mask는 각 class마다 생성, L_{mask} 는 해당되는 class의 mask만 계산에 참여함.

Mask R-CNN

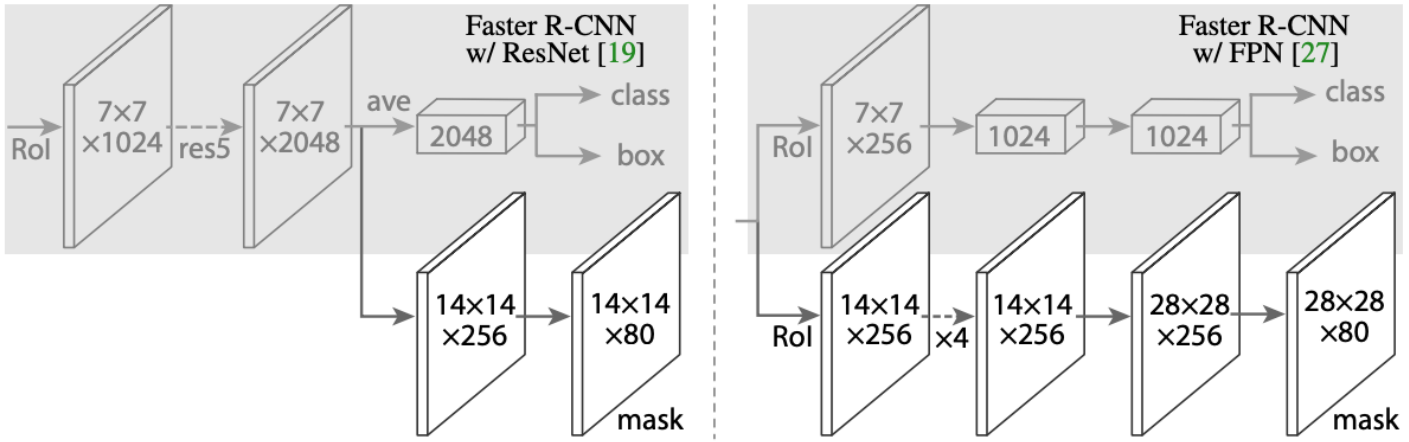


Figure 4. **Head Architecture**: We extend two existing Faster R-CNN heads [19, 27]. Left/Right panels show the heads for the ResNet C4 and FPN backbones, from [19] and [27], respectively, to which a mask branch is added. Numbers denote spatial resolution and channels. Arrows denote either conv, deconv, or *fc* layers as can be inferred from context (conv preserves spatial dimension while deconv increases it). All convs are 3×3, except the output conv which is 1×1, deconvs are 2×2 with stride 2, and we use ReLU [31] in hidden layers. *Left*: 'res5' denotes ResNet's fifth stage, which for simplicity we altered so that the first conv operates on a 7×7 RoI with stride 1 (instead of 14×14 / stride 2 as in [19]). *Right*: '×4' denotes a stack of four consecutive convs.

Result

Figure 5. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

Result

	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**

Table 1. **Instance segmentation** *mask* AP on COCO `test-dev`. MNC [10] and FCIS [26] are the winners of the COCO 2015 and 2016 segmentation challenges, respectively. Without bells and whistles, Mask R-CNN outperforms the more complex FCIS+++, which includes multi-scale train/test, horizontal flip test, and OHEM [38]. All entries are *single-model* results.

Result

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [41]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [39]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

Table 3. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on `test-dev`. Mask R-CNN using ResNet-101-FPN outperforms the base variants of all previous state-of-the-art models (the mask output is ignored in these experiments). The gains of Mask R-CNN over [27] come from using RoIAlign (+1.1 AP^{bb}), multitask training (+0.9 AP^{bb}), and ResNeXt-101 (+1.6 AP^{bb}).

참고 링크

http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture12.pdf