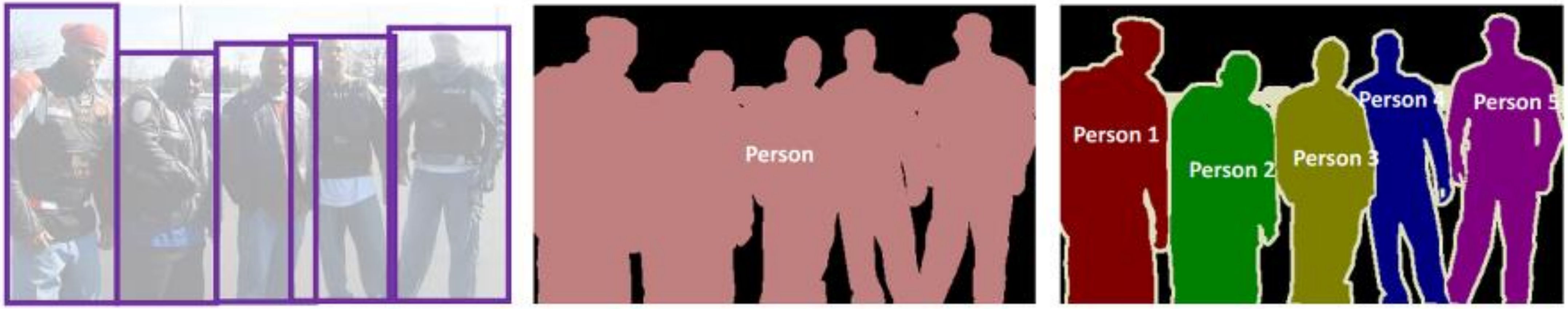
Mask R-CNN

Presenter: Mira Kim

Instance Segmentation



Object Detection | Semantic Segmentation | **Instance Segmentation**

Mask R-CNN

- Framework for *Object Instance Segmentation*
- Extends Faster R-CNN by adding a branch for predicting an object mask

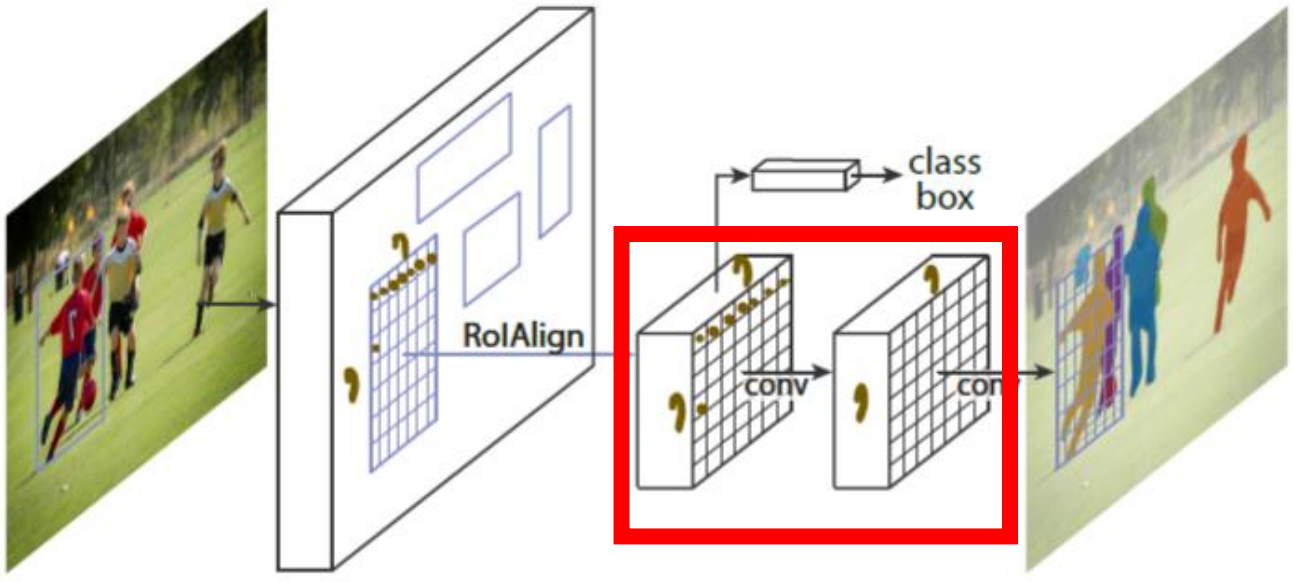
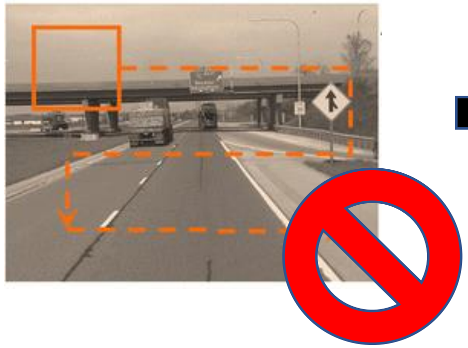
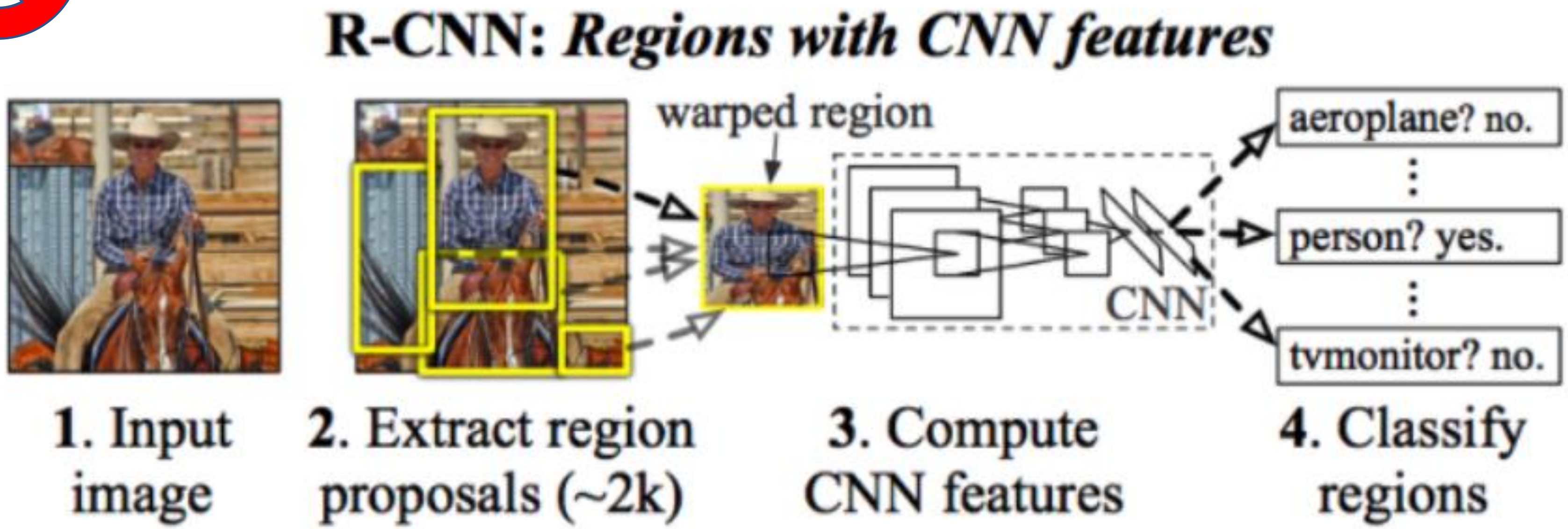


Figure 1. The **Mask R-CNN** framework for instance segmentation.

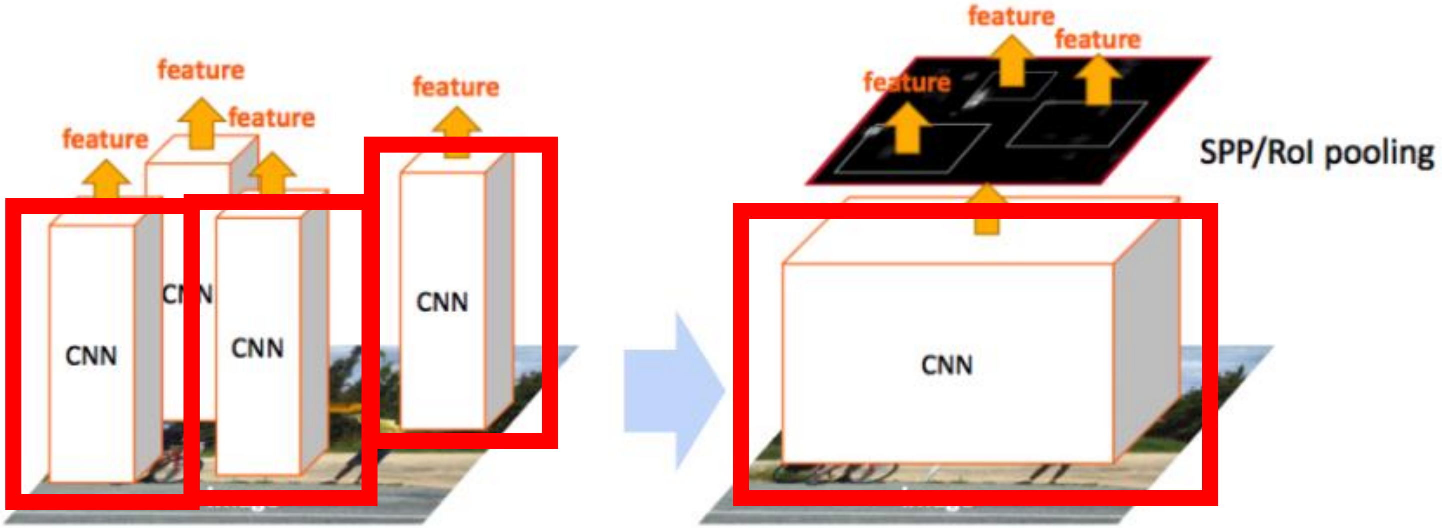
Prerequisite 1. (slow) R-CNN





Multiple Stages
➔ Too Slow and Complex☹

Prerequisite 2. Fast R-CNN



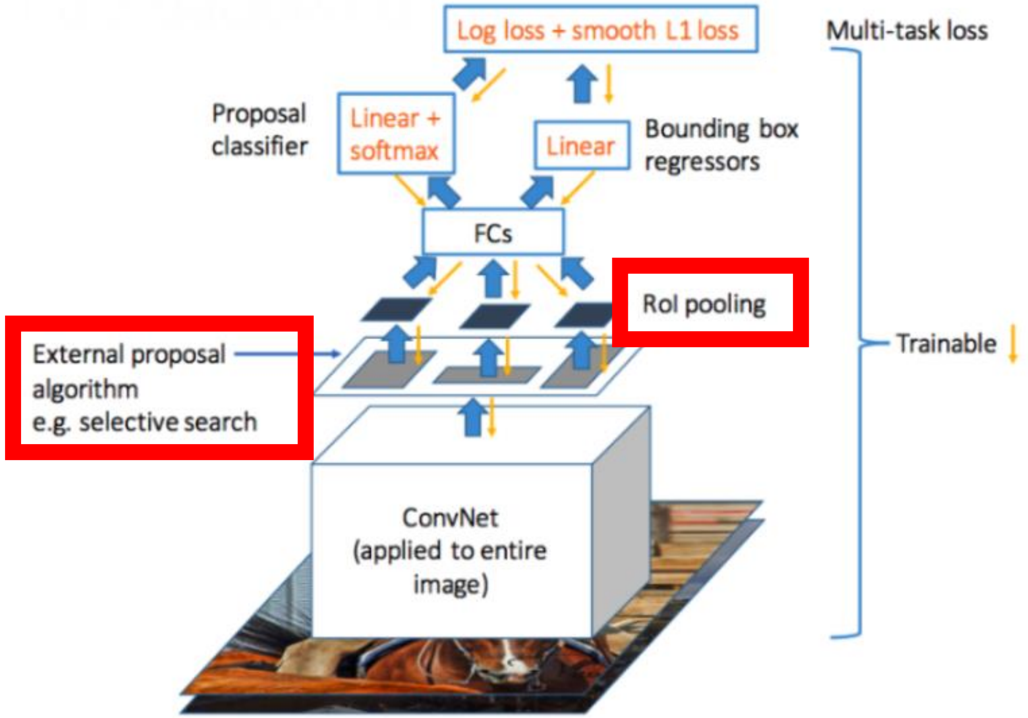
R-CNN

- Extract image regions
- 1 CNN per region (2000 CNNs)
- Classify region-based features
- Complexity: $\sim 224 \times 224 \times 2000$

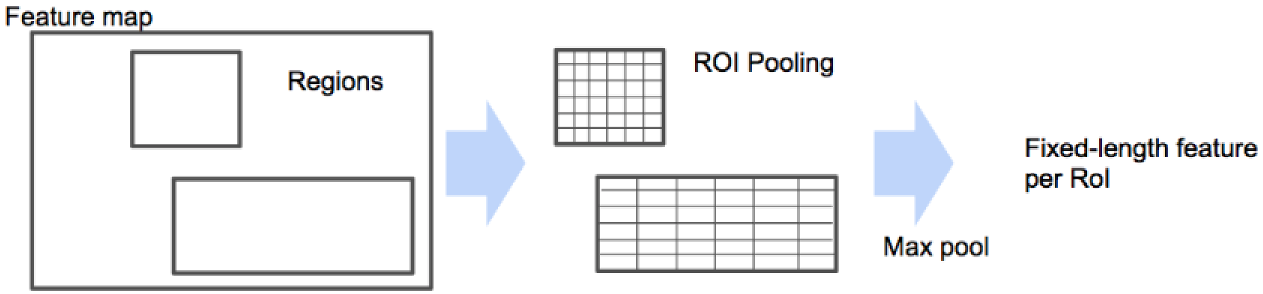
SPP-net & Fast R-CNN (the same forward pipeline)

- 1 CNN on the entire image
- Extract features from feature map regions
- Classify region-based features
- Complexity: $\sim 600 \times 1000 \times \mathbf{1}$
- $\sim$160x **faster** than R-CNN

Prerequisite 2. Fast R-CNN

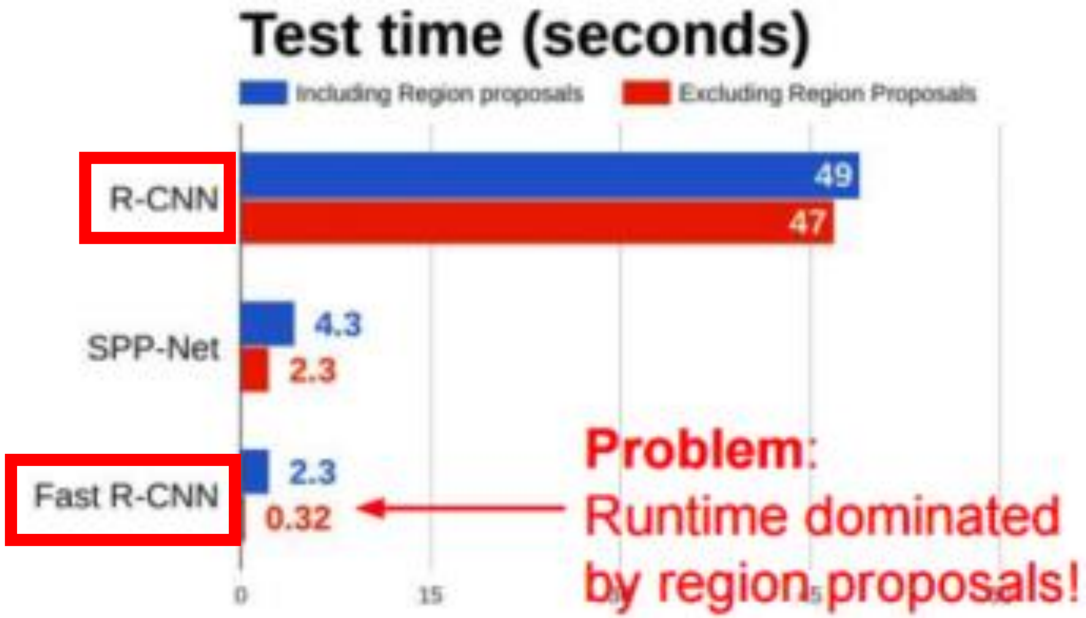


Crop and
R-CNN: Warping ☹
Fast F-CNN: RoI-Pooling ☺

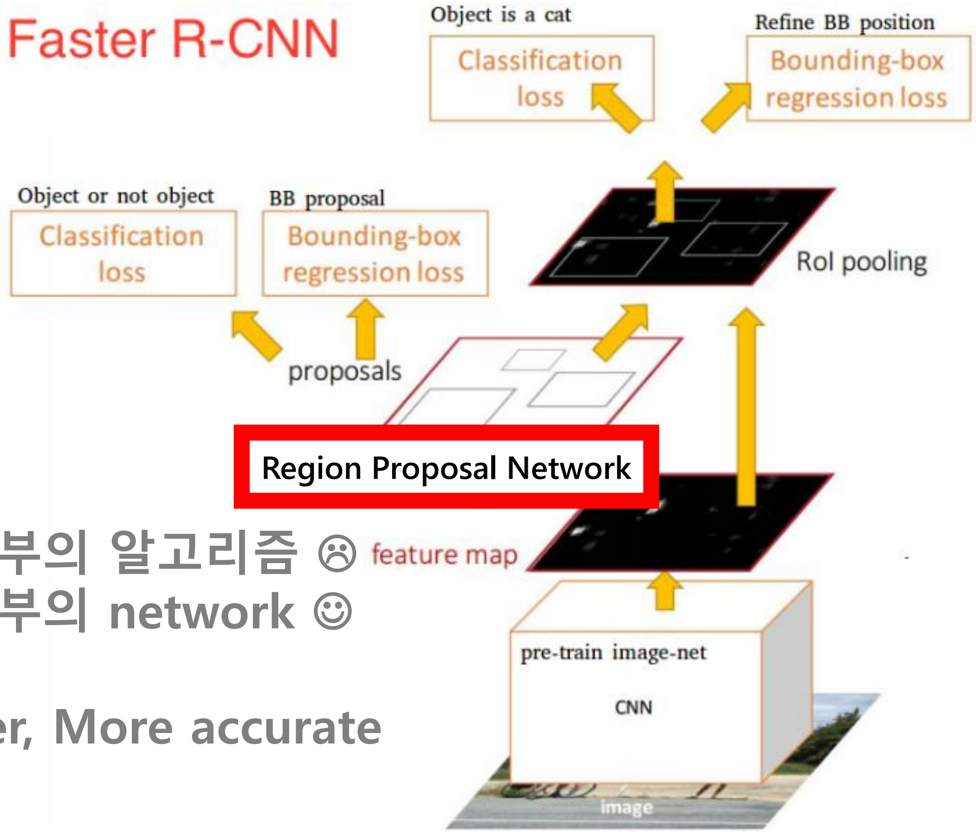


RoI pooling layer의 pooling 과정

Prerequisite 2. Fast R-CNN



Prerequisite 3. Faster R-CNN

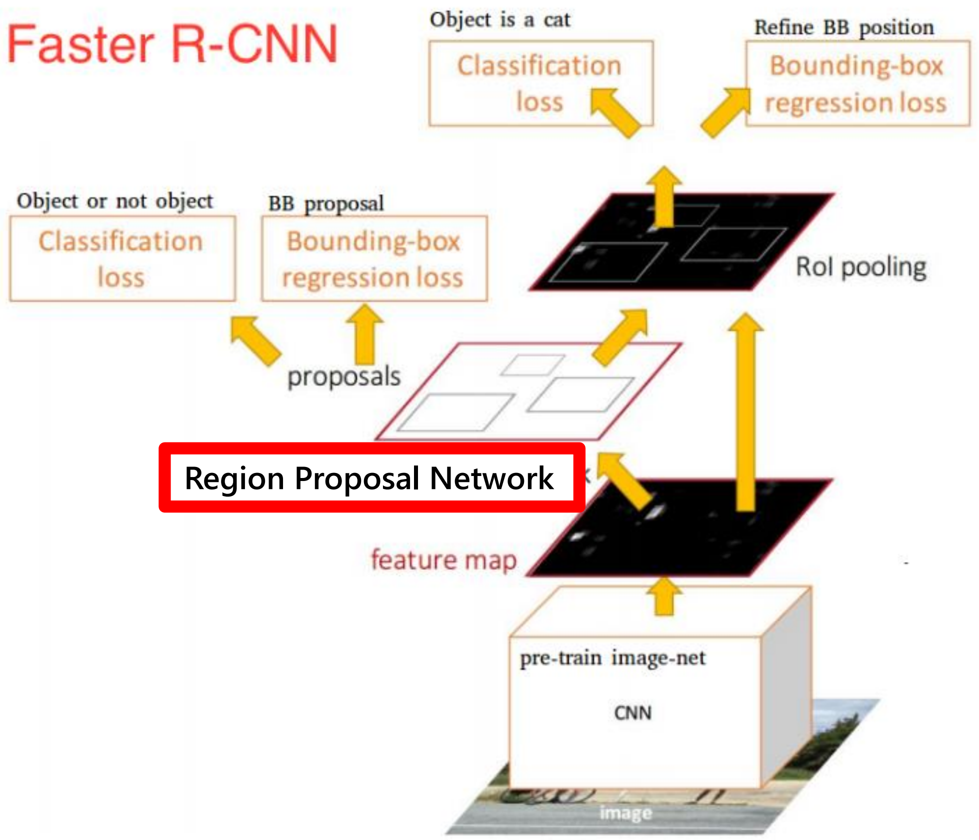


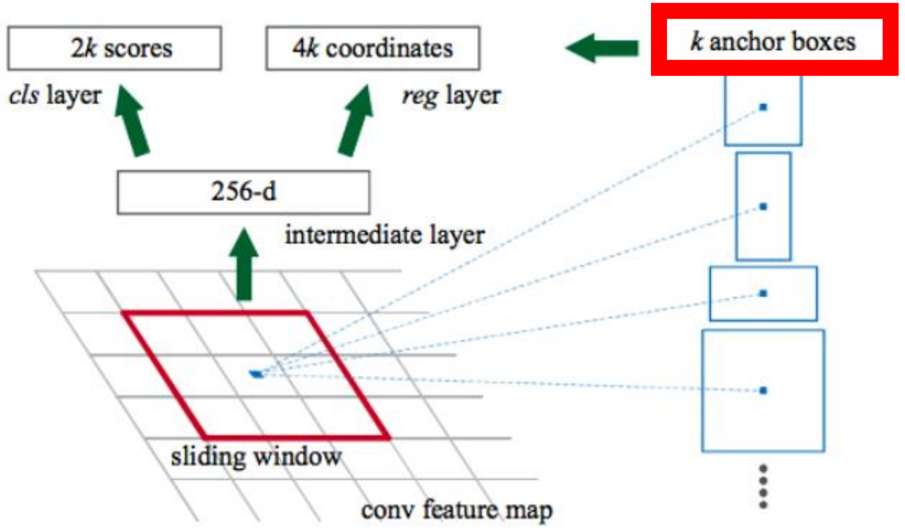
CNN 외부의 알고리즘 ☹
CNN 내부의 network ☺

➔ Faster, More accurate

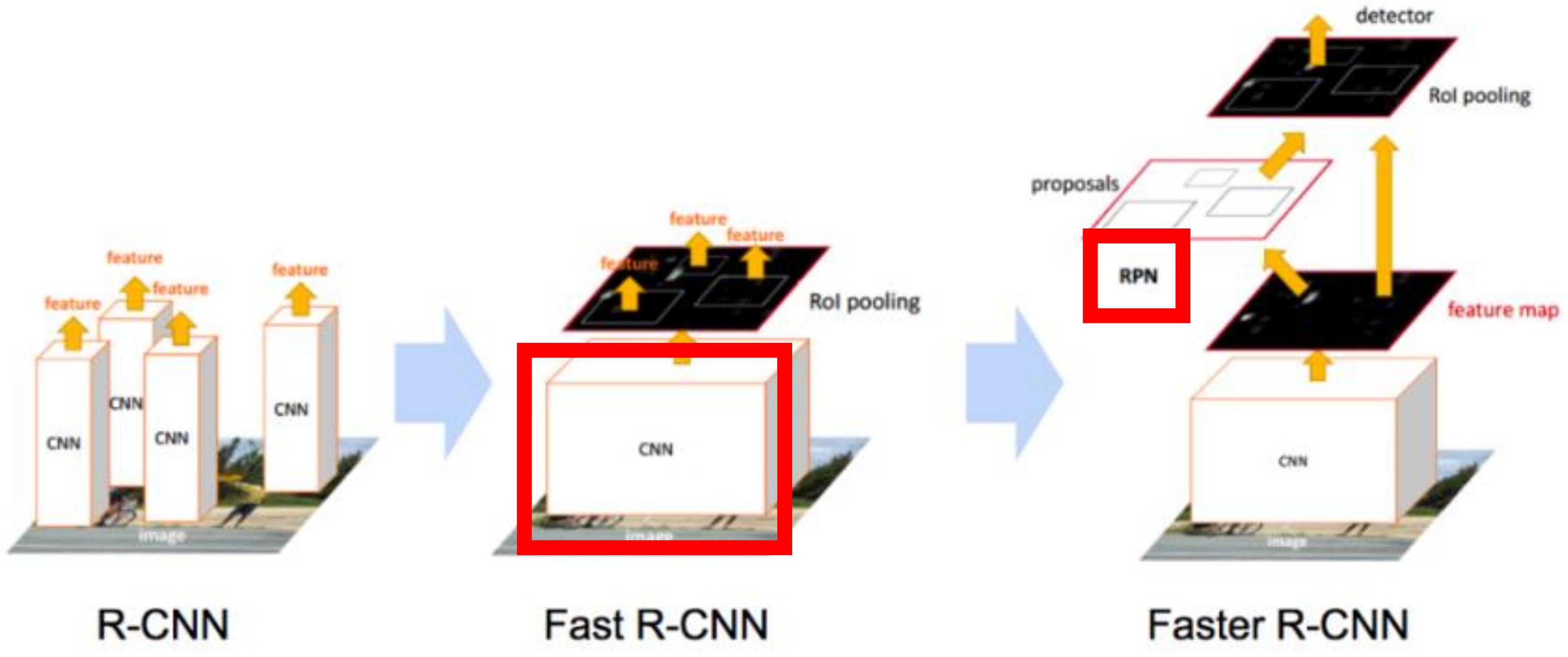
ZFNet[8] 논문에서 보여준 feature map activation 시각화

Prerequisite 3. Faster R-CNN

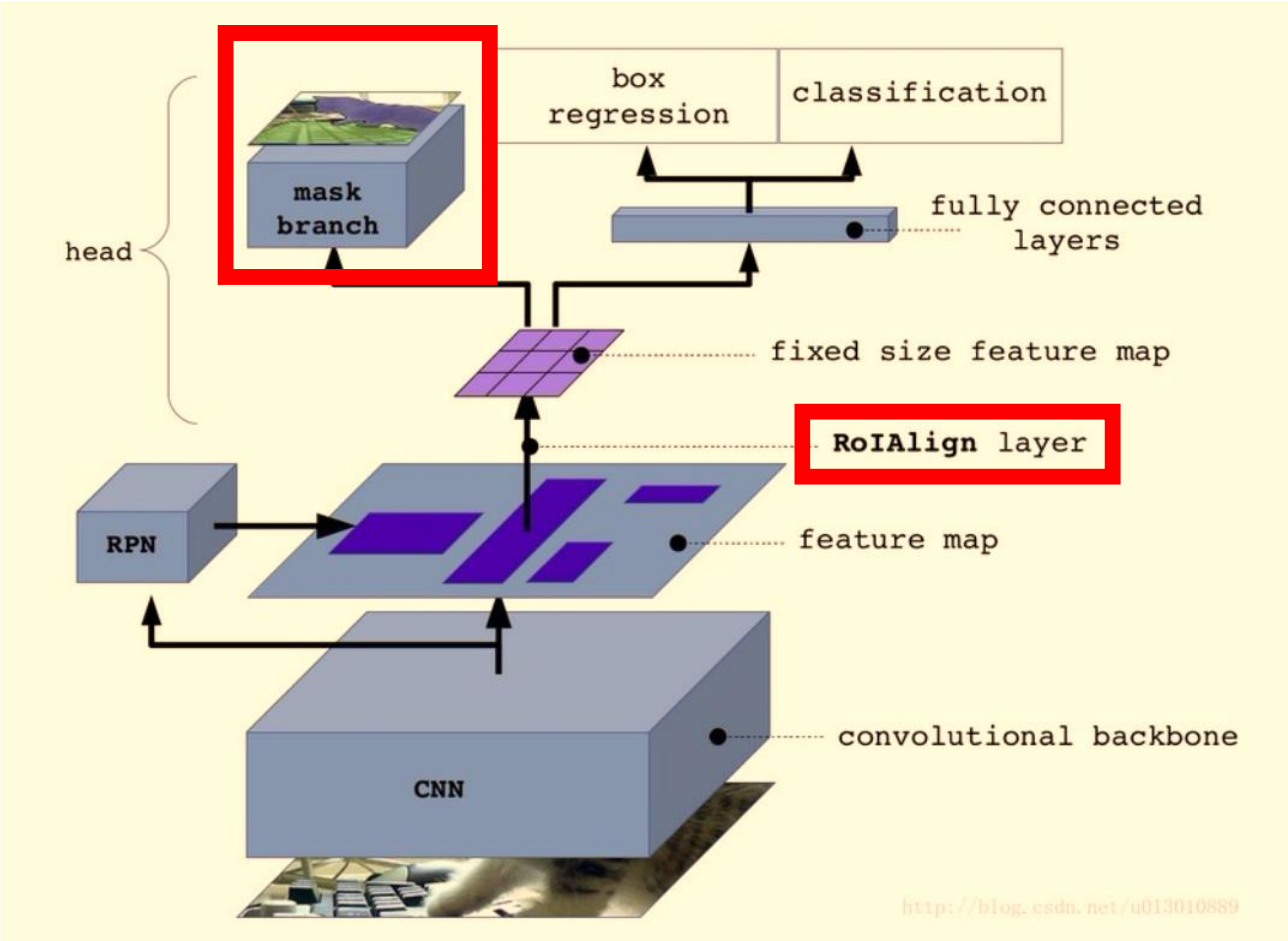


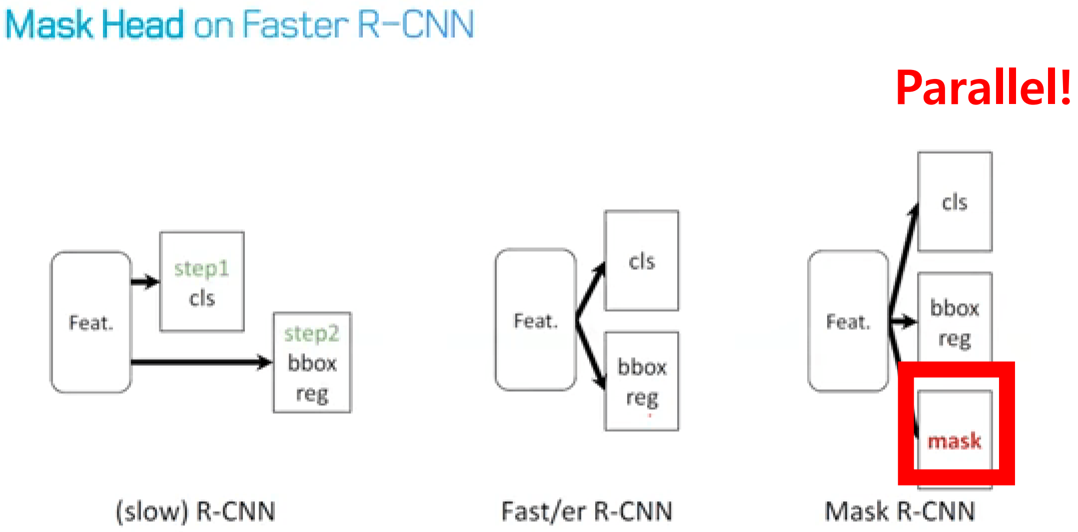


Prerequisites Summary

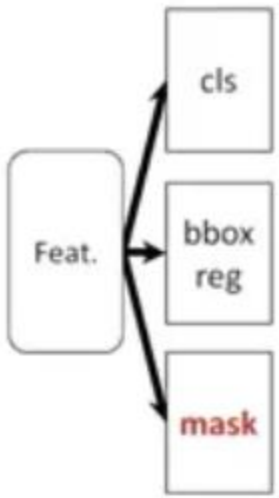


Mask R-CNN 1) Mask Branch





Mask R-CNN 1) Mask Branch



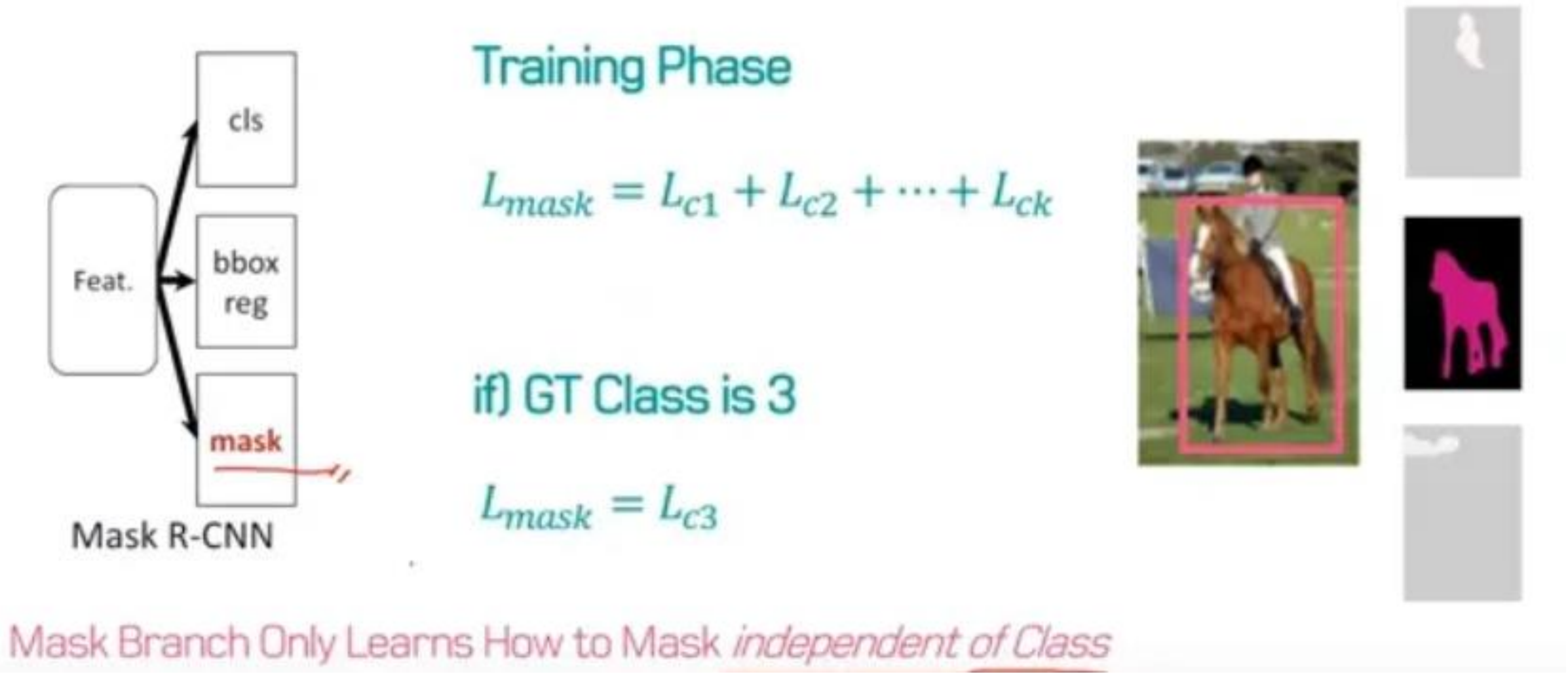
Mask R-CNN

DB
BBox + Class + Mask

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

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

Mask R-CNN 1) Mask Branch



# of class = K,
= mask branch는 RoI마다 K개의 mask를 만든다.
➔ Output of mask branch = Km^2 for each RoI
(K개의 m^2 사이즈의 binary mask)

Mask R-CNN

Segmentation First Strategy (*e.g. FCN Outputs*)
1) Per pixel-classification result
2) Cut the pixels of the same category into different instances

➔ Slow, Less accurate

Mask R-CNN

1) Instance First Strategy
2) Parallel prediction of masks and class labels

➔ Simpler, Faster

Ablation Experiments

2) Multinomial vs **Independent Masks**

	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).

Mask R-CNN

:

decouples mask and class prediction

Box branch predicts class label

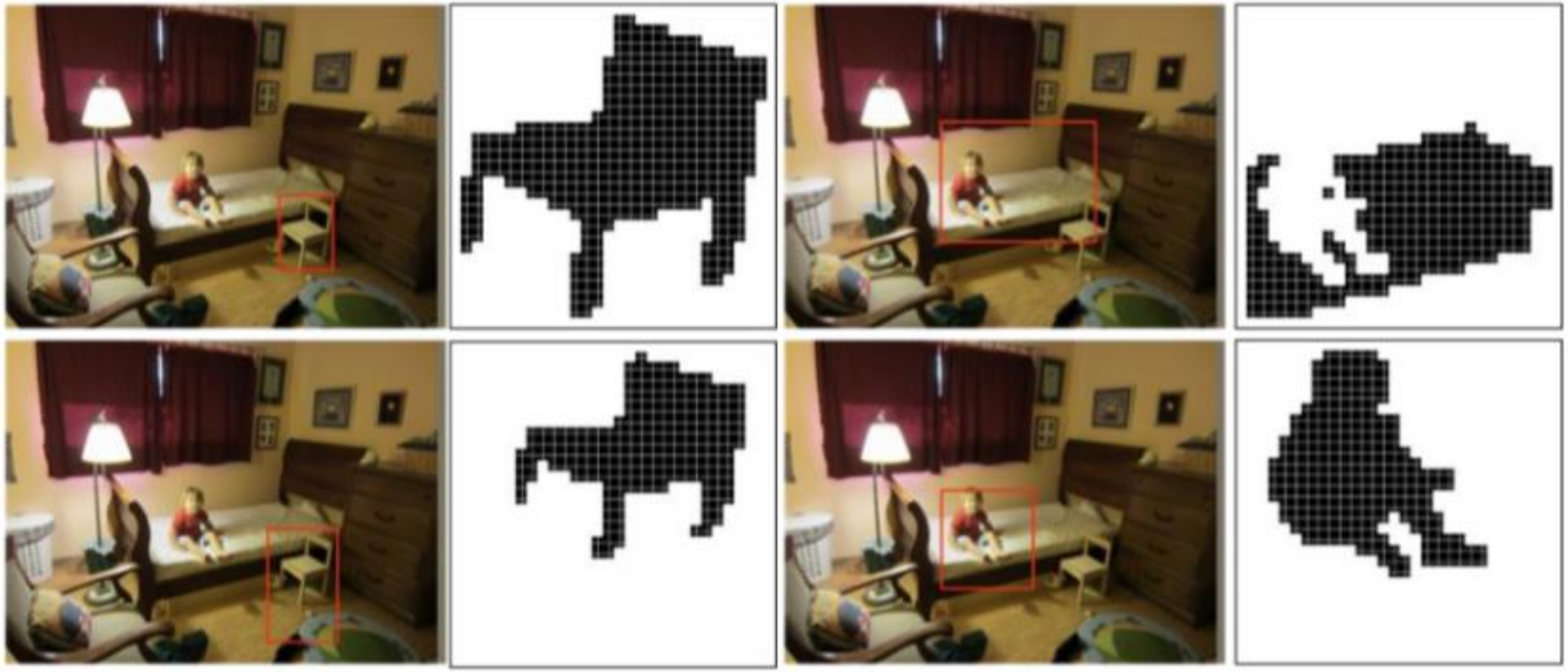
Without competition, each class has its own mask

➔ Per-pixel sigmoid, binary loss

VS

Per-pixel softmax, multinomial loss(FCNs)

Mask R-CNN 1) Mask Branch



Ablation Experiments

3) **Class-Specific** vs. Class-Agnostic Masks

	Mask AP
Class-specific mask	30.3
Class-agnostic mask	29.7

Mask R-CNN: class-specific mask

Interestingly,
Class-agnostic mask is nearly **effective**

Ablation Experiments

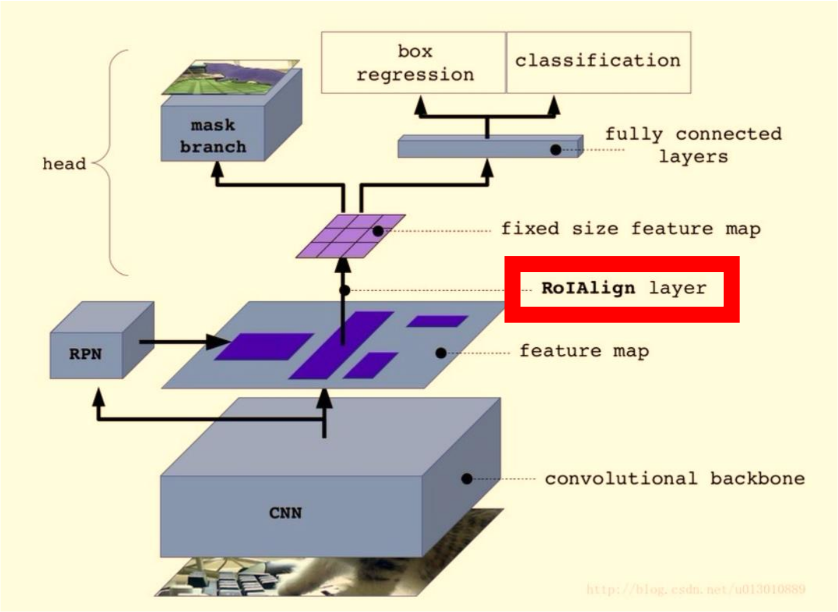
5) Mask Branch

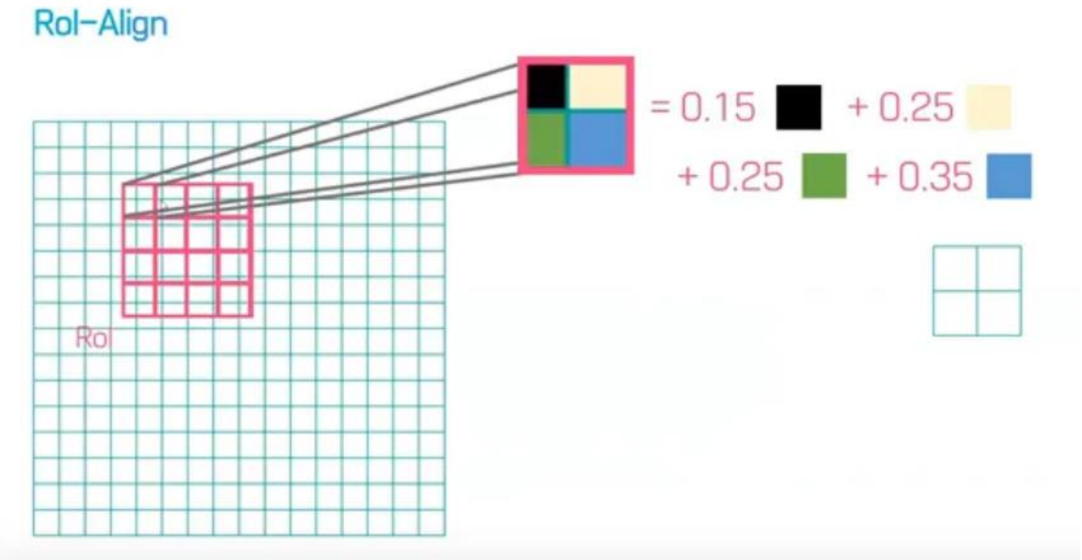
Class label, bounding box offset ➔ fc layers ➔ vector type
Mask representation ➔ pixel to pixel ➔ extract spatial structure

	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**

(e) **Mask Branch** (ResNet-50-FPN): Fully convolutional networks (FCN) *vs.* multi-layer perceptrons (MLP, fully-connected) for mask prediction. FCNs improve results as they take advantage of explicitly encoding spatial layout.

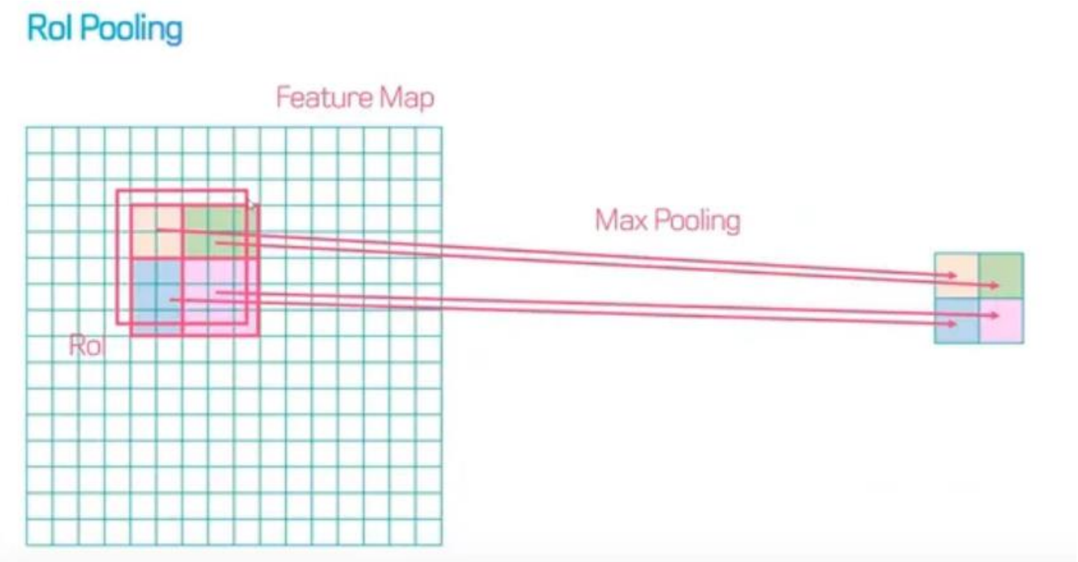
Mask R-CNN 2) RoI Align



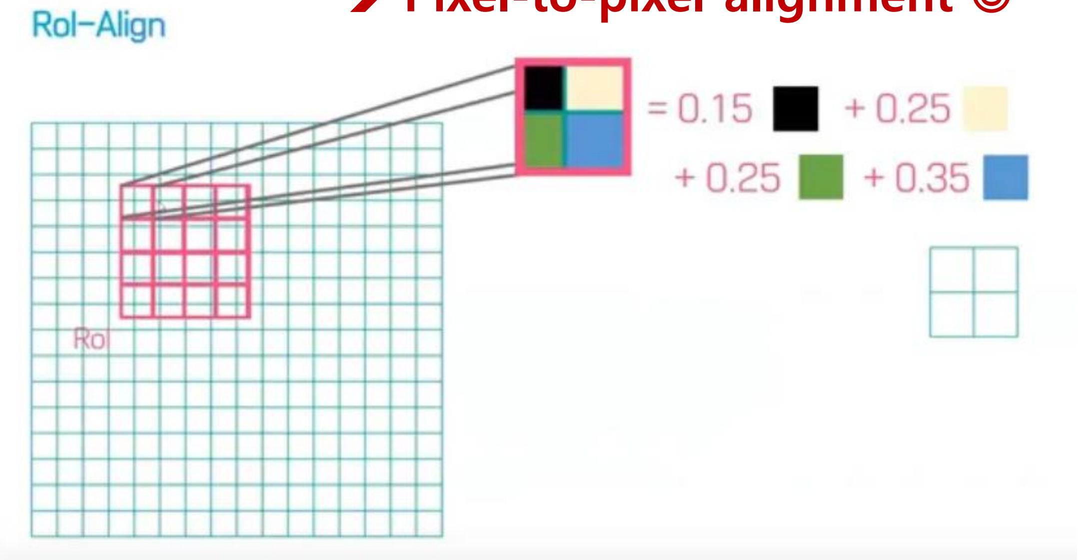


RoI Pooling vs. RoI Align

Quantization ☹



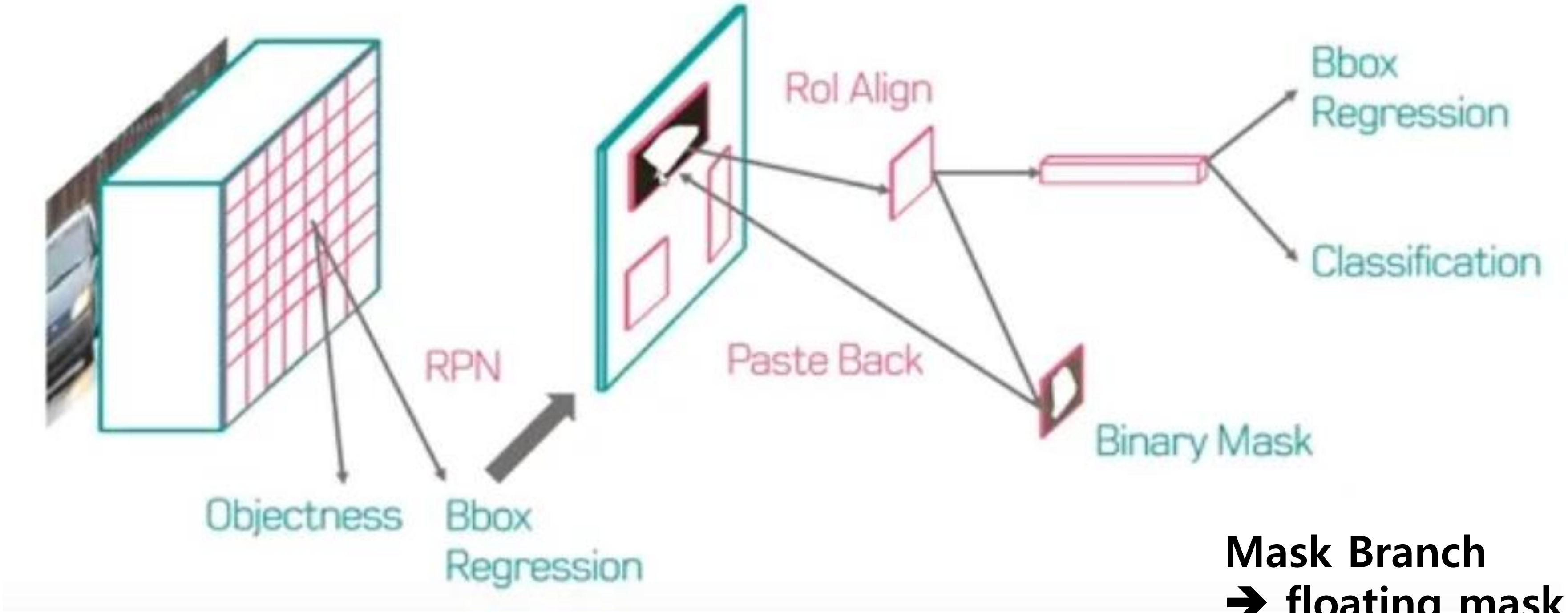
Bilinear interpolation ☺
➔ Pixel-to-pixel alignment ☺



Mask R-CNN

Stage 1

Stage 2



Mask Branch
➔ floating mask output
➔ RoI size로 resized
➔ 0.5 threshold로 binarized

Ablation Experiments

4) **RoI Align**(ResNet-50-C4 stride 16, ResNet-50-C5 stride 32)

	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 RoI layers. Our RoIAlign layer improves AP by ~3 points and AP_{75} by ~5 points. Using proper alignment is the only factor that contributes to the large gap between RoI layers.

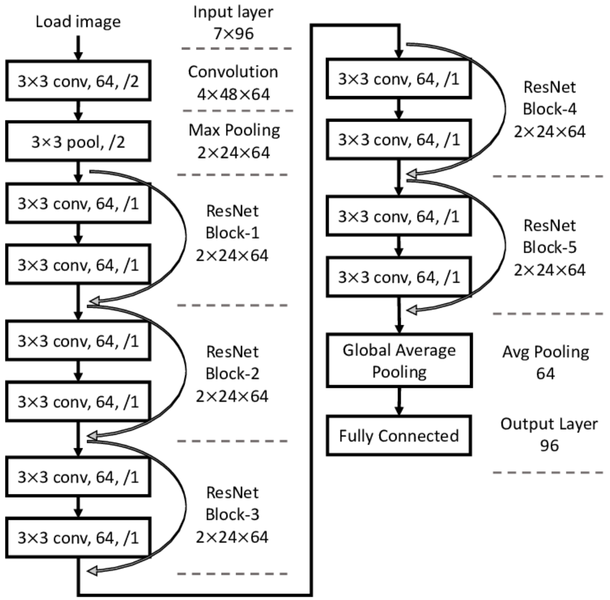
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

(d) **RoIAlign** (ResNet-50-**C5**, *stride 32*): Mask-level and box-level AP using *large-stride* features. Misalignments are more severe than with stride-16 features (Table 2c), resulting in big accuracy gaps.

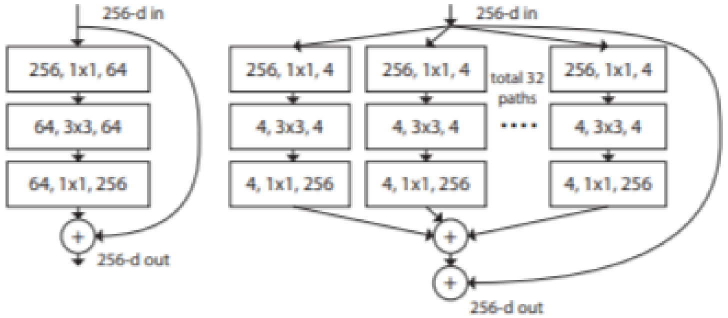
1) Higher IoU ➔ Higher AP gap
2) More accurate with ResNet-50-C5, stride 32 than C4, stride 16
➔ Solves the accuracy problem with larger stride in detection and segmentation problem

Mask R-CNN Backbone

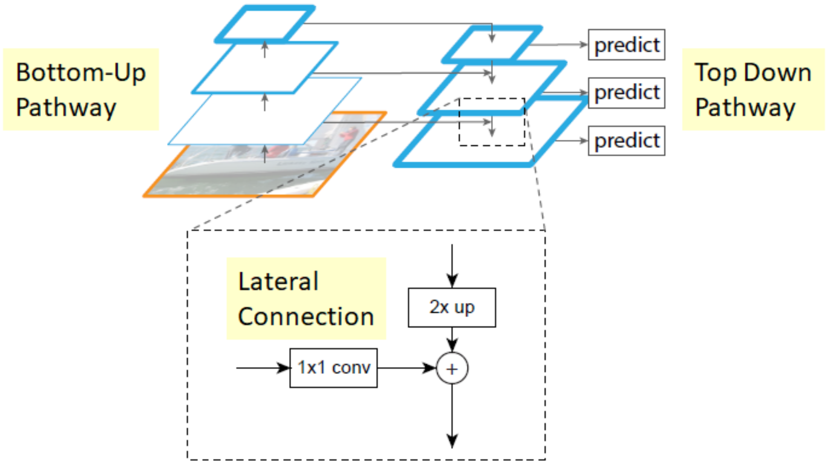
ResNet



ResNeXt



FPN(Feature Pyramid Network)



➔ **ResNeXt-101-FPN**

Ablation Experiments

1) Architecture

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.

Mask AP

C4 < FPN
ResNet < ResNeXt

Mask R-CNN Training & Inference

Training

1) RoI = positive if IoU > 0.5
2) L mask only in positive RoI
3) Mask Target = Intersection between RoI and groundtruth mask
4) Minibatch size = 16
5) # of sampled RoI:
   C4: 64
   FPN: 512
6) RPN = 5 scale, 3 aspect ratio

Inference

1) # of sampled RoI:
   C4: 300
   FPN: 1000

2) box prediction branch + NMS on each RoI

3) Top 100 biggest scoring detection box
➔ mask branch
(inference 단계에서는 parallel X)

➔ fewer & more accurate RoI 사용
➔ speed up inference time
+ small overhead

Bounding Box Detection Results

(mask output ignored)

	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}).

	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.

Mask R-CNN for Human Pose Estimation

Figure 7. Keypoint detection results on COCO test using Mask R-CNN (ResNet-50-FPN), with person segmentation masks predicted from the same model. This model has a keypoint AP of 63.1 and runs at 5 fps.

Keypoint location = one-hot m * m binary mask
# of Keypoints = K ➔ K binary masks (only one pixel is labeled as a foreground)

Mask R-CNN for Human Pose Estimation

	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
CMU-Pose+++ [6]	61.8	84.9	67.5	57.1	68.2
G-RMI [32]†	62.4	84.0	68.5	**59.1**	68.1
Mask R-CNN, keypoint-only	62.7	87.0	68.4	57.4	71.1
Mask R-CNN, keypoint & mask	**63.1**	**87.3**	**68.7**	57.8	**71.4**

Table 4. **Keypoint detection** AP on COCO `test-dev`. Ours is a single model (ResNet-50-FPN) that runs at 5 fps. CMU-Pose+++ [6] is the 2016 competition winner that uses multi-scale testing, post-processing with CPM [44], and filtering with an object detector, adding a cumulative ~5 points (clarified in personal communication). †: G-RMI was trained on COCO *plus* MPII [1] (25k images), using two models (Inception-ResNet-v2 for bounding box detection and ResNet-101 for keypoints).

	AP^{bb}_{person}	AP^{mask}_{person}	AP^{kp}
Faster R-CNN	52.5	-	-
Mask R-CNN, mask-only	**53.6**	**45.8**	-
Mask R-CNN, keypoint-only	50.7	-	64.2
Mask R-CNN, keypoint & mask	52.0	45.1	**64.7**

Table 5. **Multi-task learning** of box, mask, and keypoint about the *person* category, evaluated on `minival`. All entries are trained on the same data for fair comparisons. The backbone is ResNet-50-FPN. The entries with 64.2 and 64.7 AP on `minival` have `test-dev` AP of 62.7 and 63.1, respectively (see Table 4).

	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
RoIPool	59.8	86.2	66.7	55.1	67.4
RoIAlign	**64.2**	**86.6**	**69.7**	**58.7**	**73.0**

Table 6. **RoIAlign *vs.* RoIPool** for keypoint detection on `minival`. The backbone is ResNet-50-FPN.

Keypoint detection benefits
from **multitask training**,
While not helping other tasks.

Q & A

Presenter: Mira Kim