Mask R-CNN

姓名：徐僖禧

专业：计算机技术

学号：18210240227

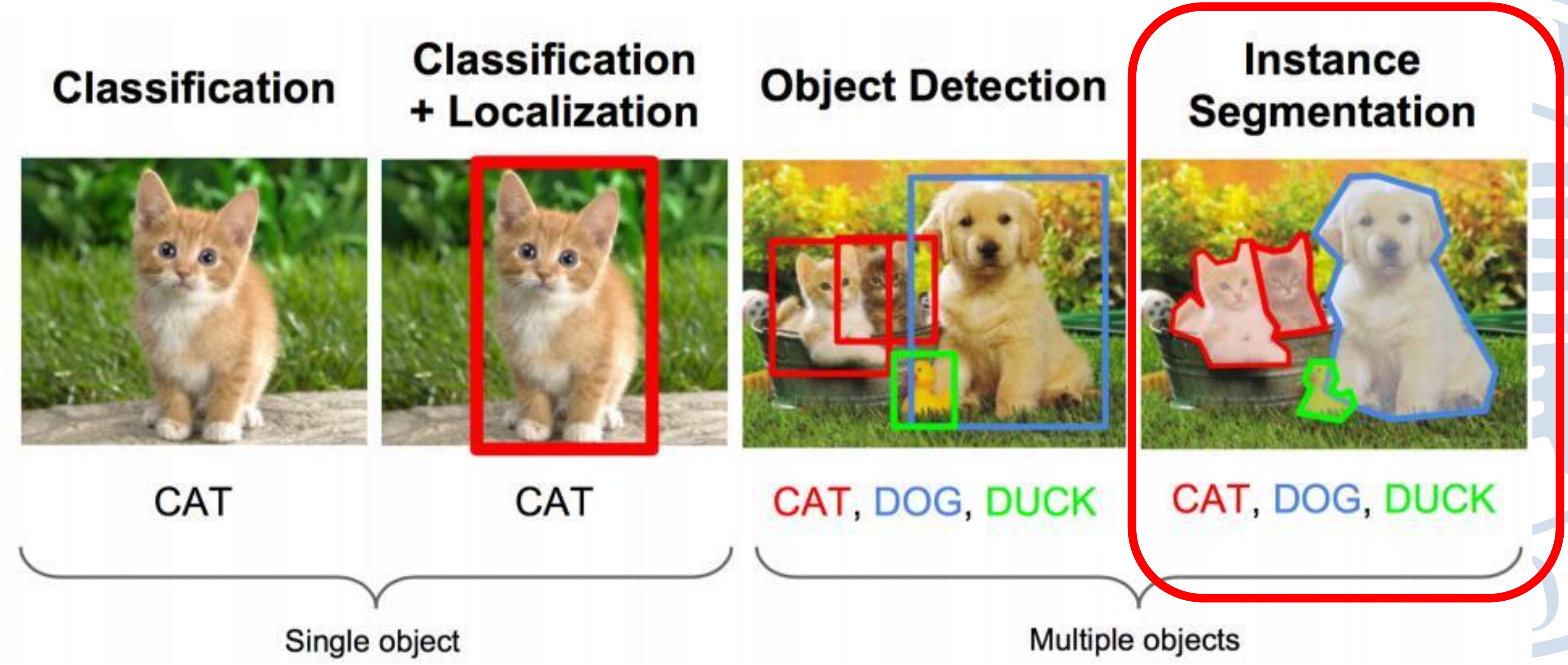
导师：薛向阳

2018.09.30

Content

1 Preliminaries

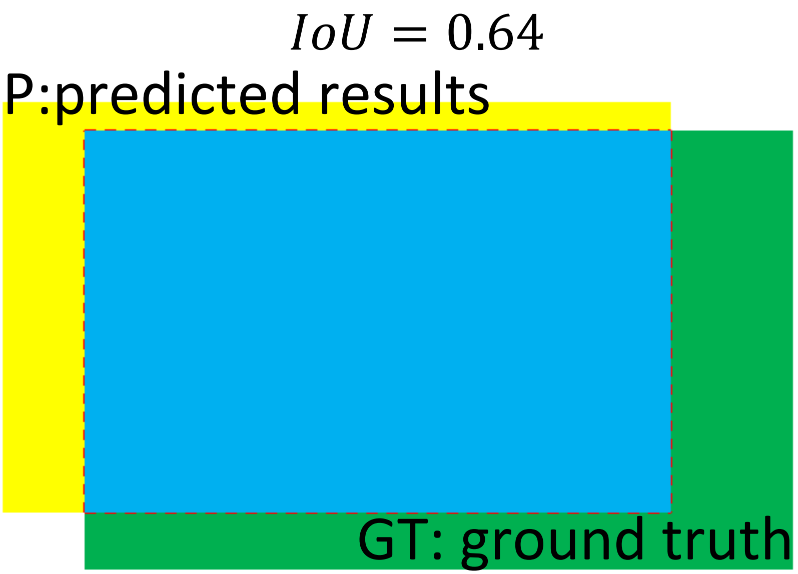
Motivation

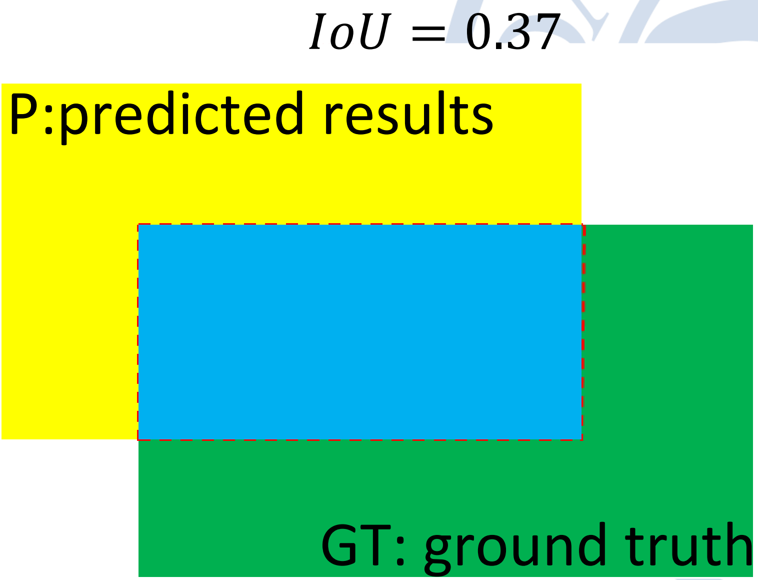

Classification
Classification
+ Localization
Object Detection
Instance
Segmentation
CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK
Single object
Multiple objects

IOU(intersection-over-union)

It refers to the overlap rate between the target window and the original marked window generated by the model.

$$IoU = \frac{DetectionResult \cap GroundTruth}{DetectionResult \cup GroundTruth}$$

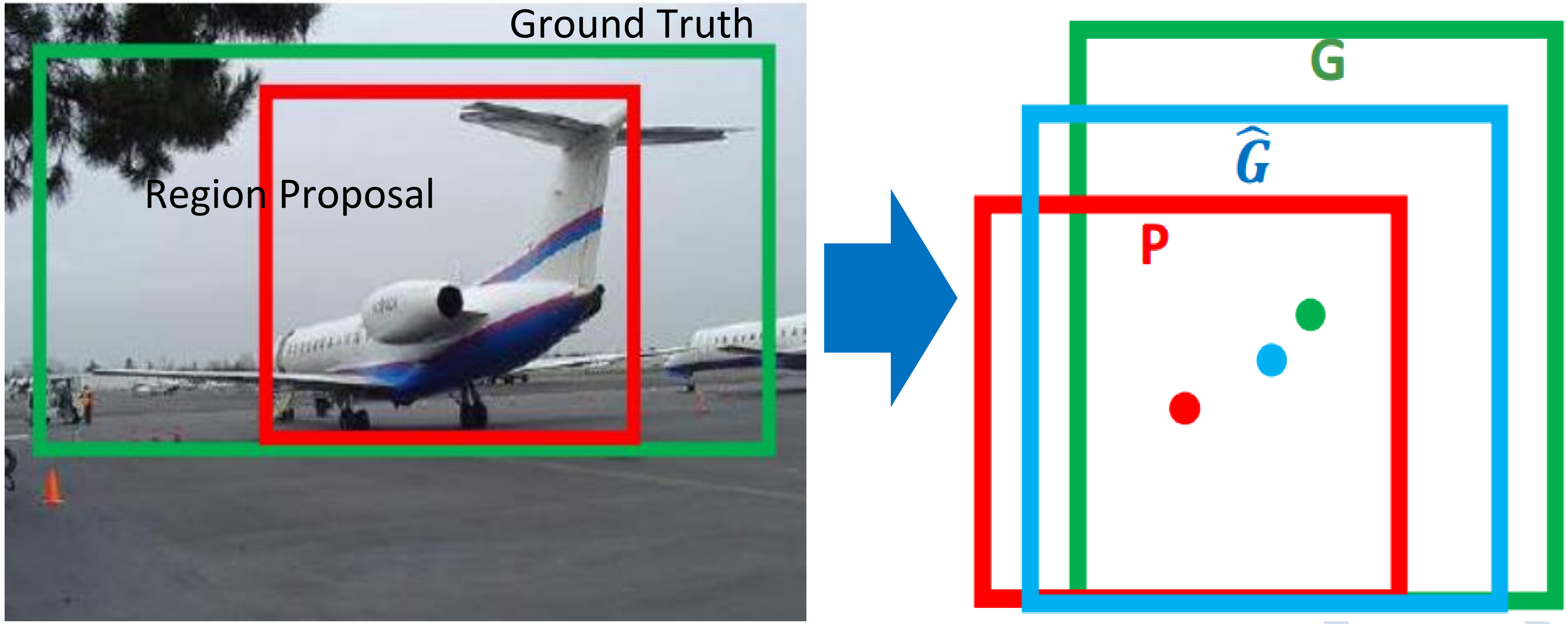




Positive sample threshold: $IoU = 0.5$

Bounding Box Regression

Fine-tune the prediction window to make positioning more accurate.



(a) (b)

$$f(P_x, P_y, P_w, P_h) = (\widehat{G_x}, \widehat{G_y}, \widehat{G_w}, \widehat{G_h})$$
$$(\widehat{G_x}, \widehat{G_y}, \widehat{G_w}, \widehat{G_h}) \approx (G_x, G_y, G_w, G_h)$$

Bounding Box Regression

Fine-tune the prediction window to make positioning more accurate.

$$\widehat{G_x} = P_w d_x(P) + P_x \qquad \widehat{G_w} = P_w \exp(d_w(P))$$

$$\widehat{G_y} = P_h d_y(P) + P_y \qquad \widehat{G_h} = P_h \exp(d_h(P))$$

$$t_x = (G_x - P_x)/P_w \qquad t_w = \log(G_w/P_w)$$

$$t_y = (G_y - P_y)/P_h \qquad t_h = \log(G_h/P_h)$$

$$Loss = \sum_i^N \left(t_*^i - \widehat{w}_*^T \phi_5(P^i)\right)^2$$

$$W_* = \operatorname{argmin}_{w_*} \sum_i^N \left(t_*^i - \widehat{w}_*^T \phi_5(P^i)\right)^2 + \lambda \|\widehat{w}_*\|^2$$

Quantitative evaluation indexs

- **Precision**
  - The ratio of the number of samples **correctly classified** by the classifier to the **total number of samples** for a given test data set.

$$precision = \frac{TP}{TP + FP}$$

- **Recall**
  - It refers to the proportion of items that are **correctly retrieved** (TP) to all items that **should be retrieved** $(TP + FN)$.

$$recall = \frac{TP}{TP + TN}$$

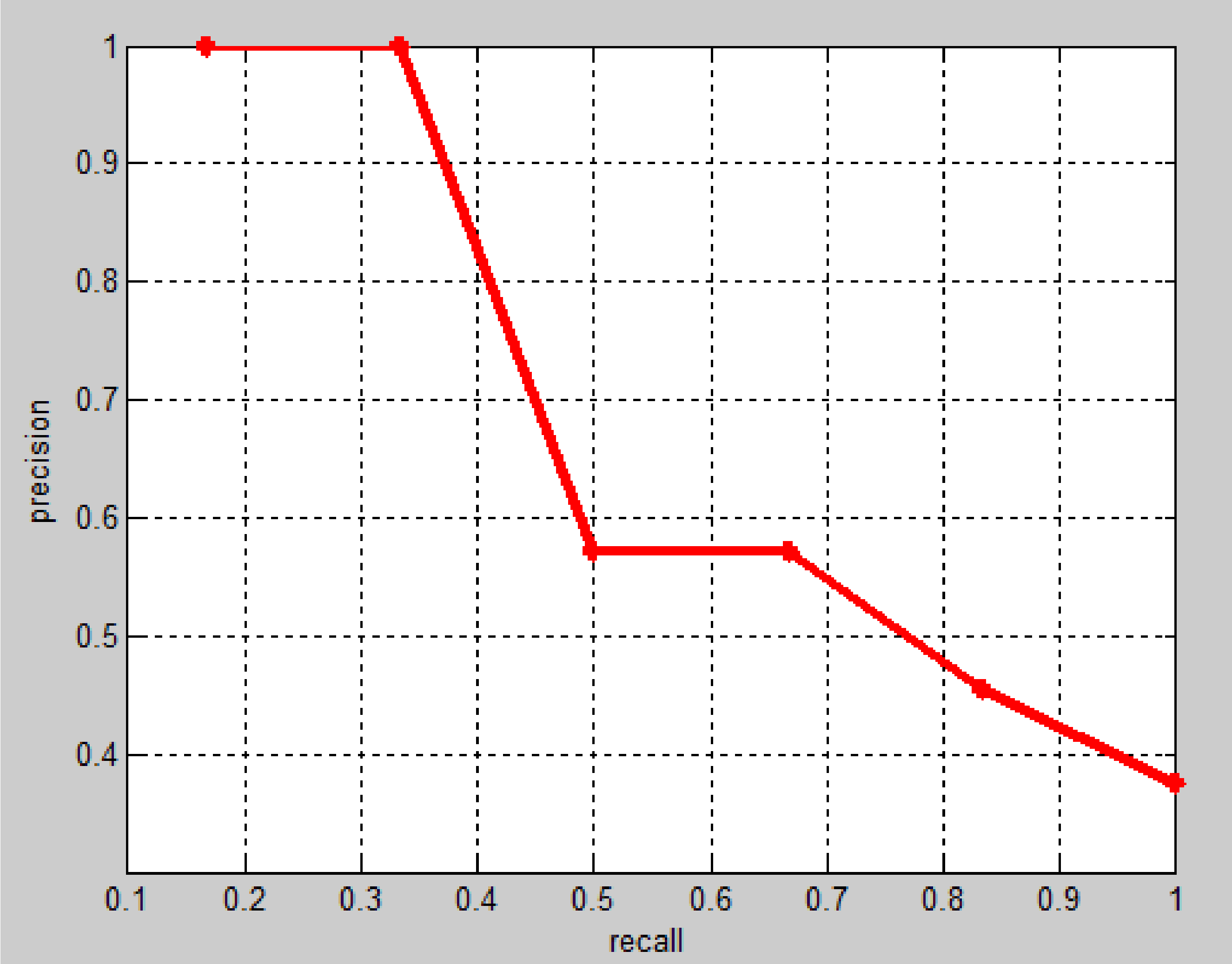
TP: True Positive
FP: False Positive
TN: True Negative

- **F-measure**
  - To evaluate the model

$$F - measure = \frac{2 * precision * recall}{precision + recall}$$

Quantitative evaluation indexs

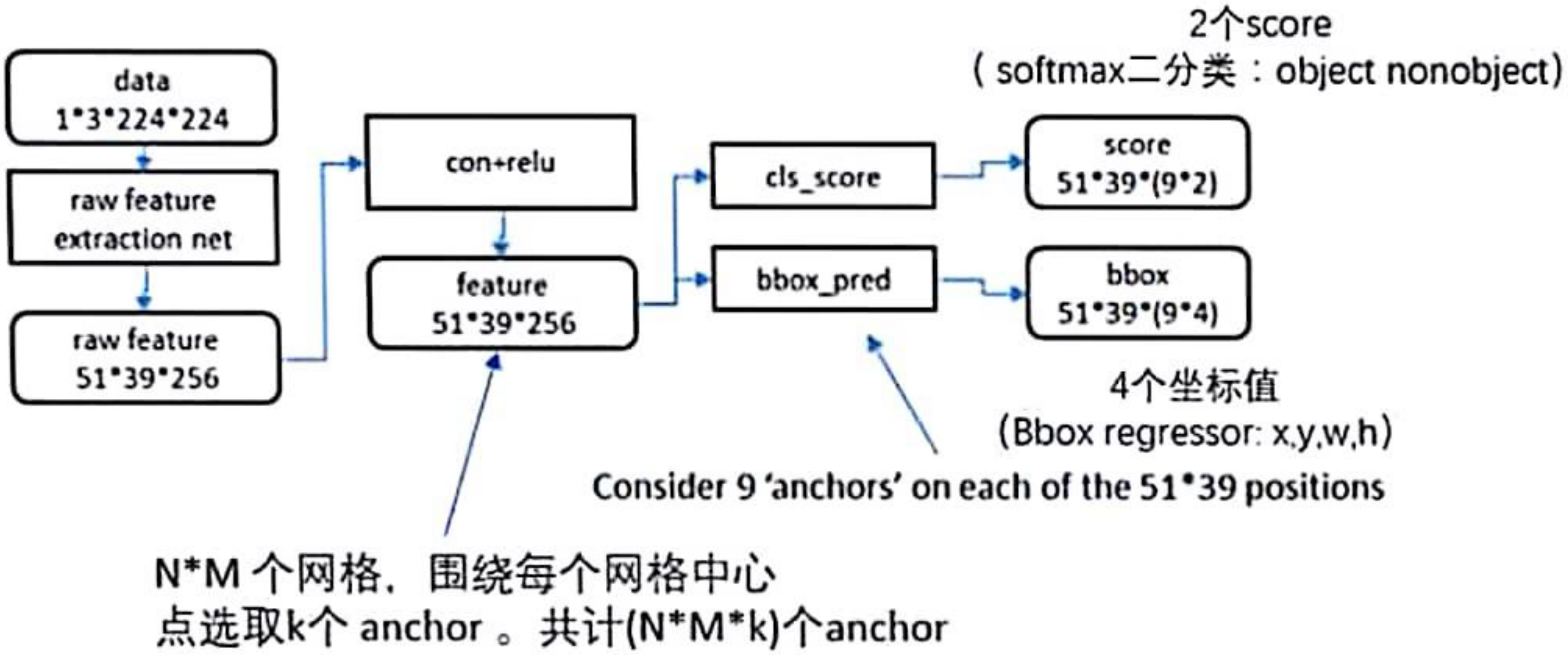
- **AP(average precision over IoU thresholds)**



2 Related Work

Region Proposal Network(RPN)

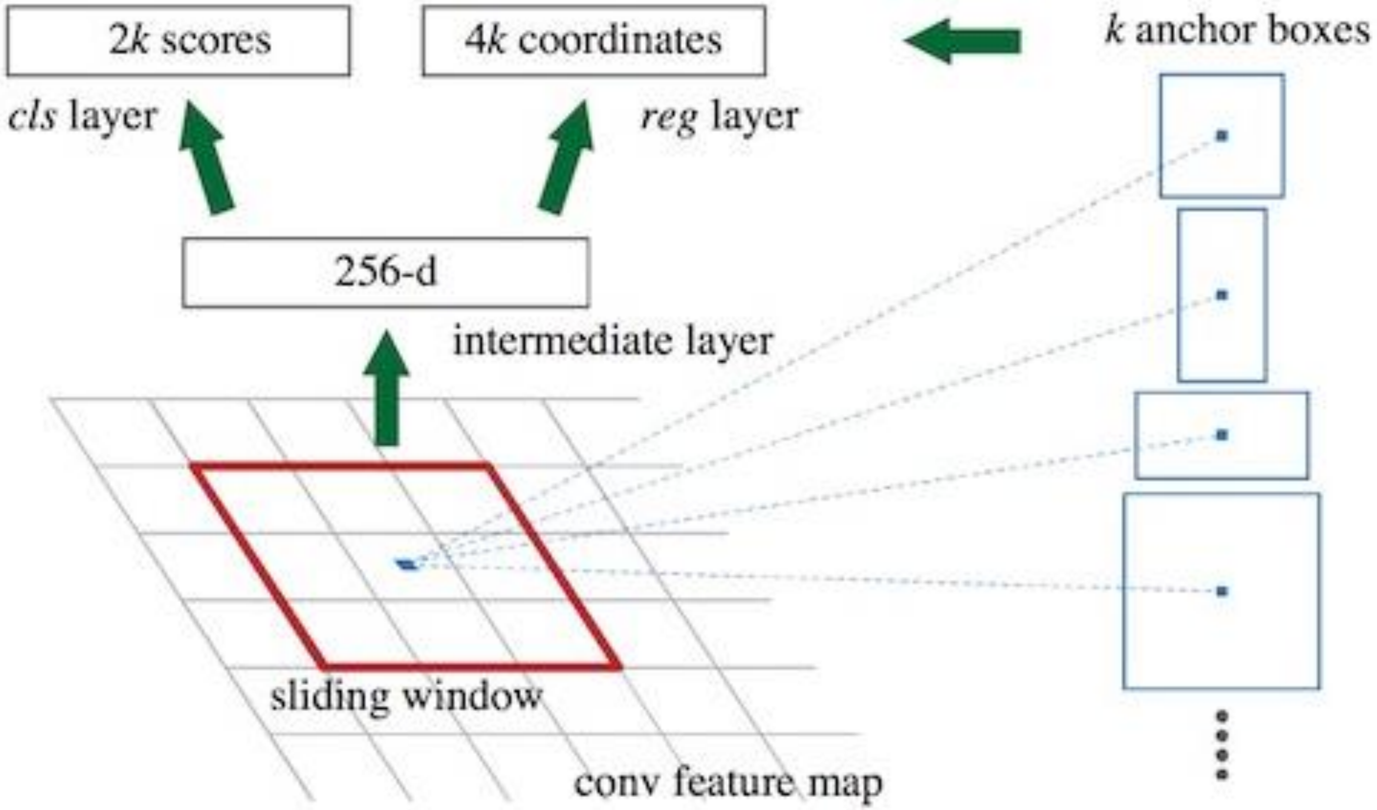
- Region Proposal Network(RPN)
- RPN not only has **no time cost** when extracting proposals, but also improves the **quality** of proposals.



Region Proposal Network(RPN)

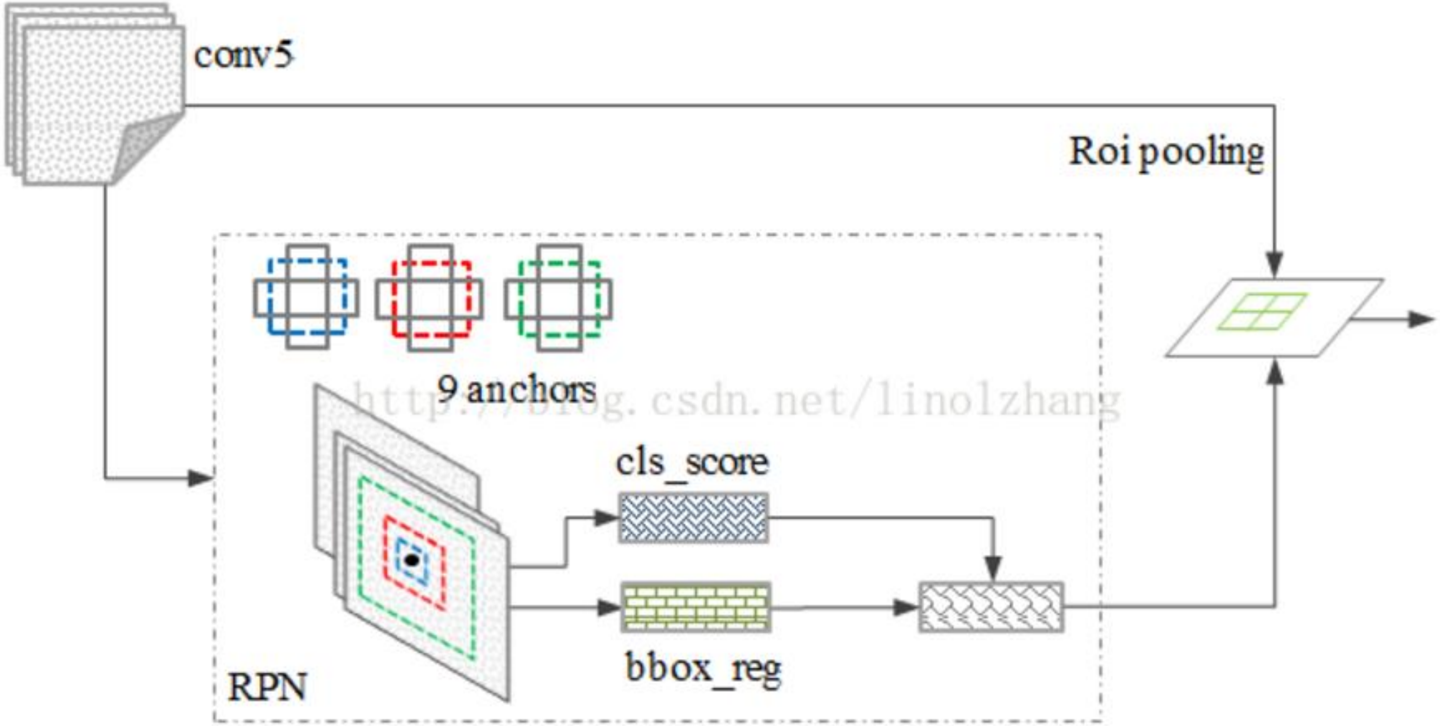
$$L(\{p_i\}\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

Classify obj./not-obj. Regress box locations.



Faster R-CNN

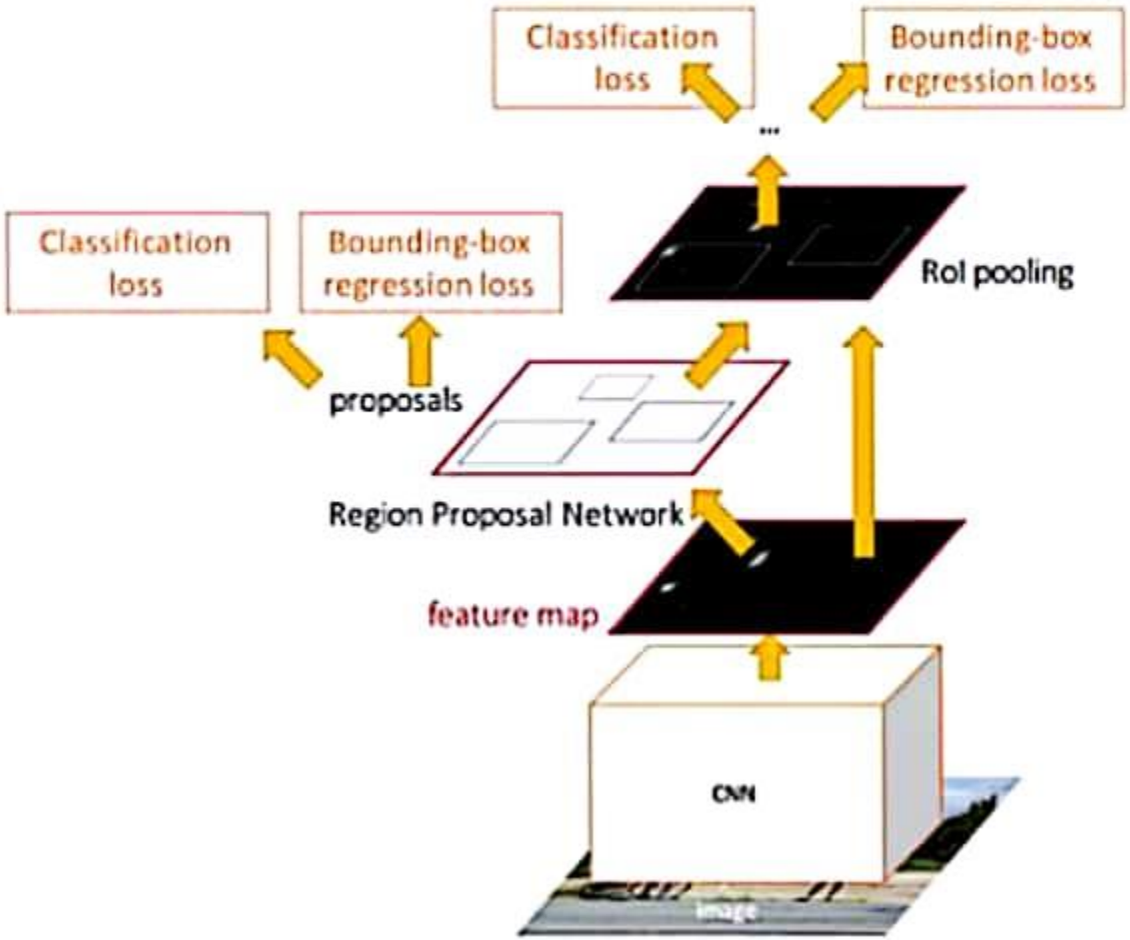
- Whole process:
  - **Step 1：** Input the whole picture to **CNN** and **get feature map**.
  - **Step 2：** **The convolution feature** is input into RPN to get the feature information of the candidate box.



Faster R-CNN

- Whole process:
  - **Step 3：** A **classifier** is used to determine whether the feature extracted from the candidate box belongs to a **particular class**.

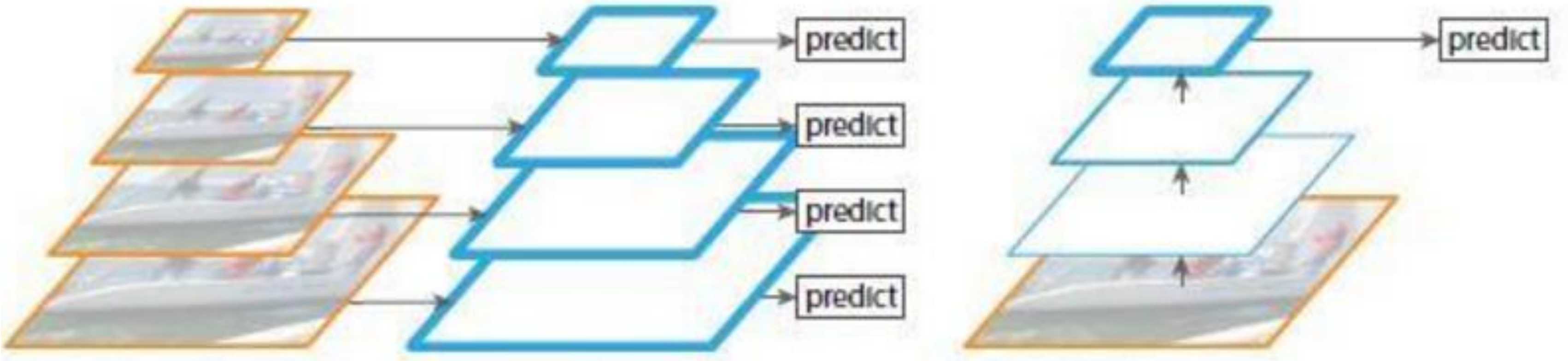
  - **Step 4：** For candidate boxes belonging to a feature, the position of them is further adjusted with **a regression**.



Network Architecture

3

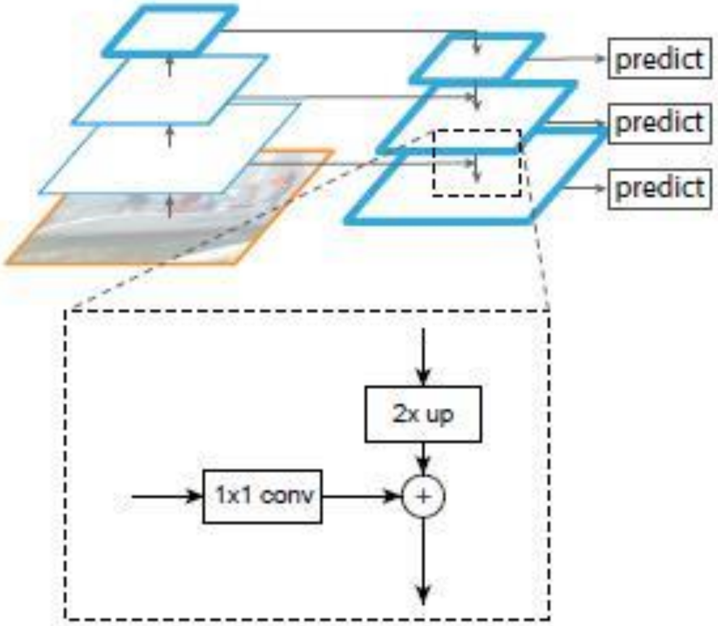
FPN（feature pyramid networks）

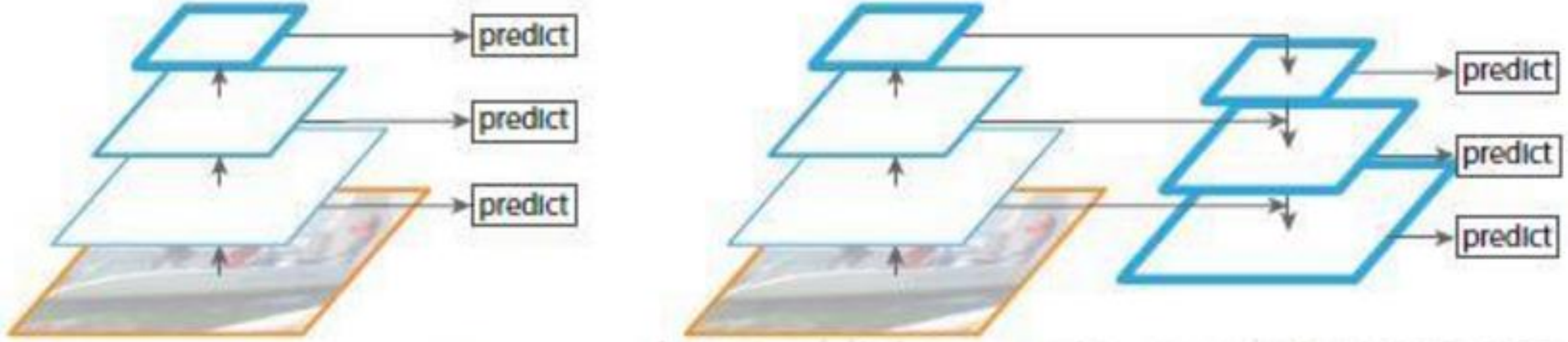


(a) Featurized image pyramid

(b) Single feature map

SPP net， Fast RCNN， Faster RCNN



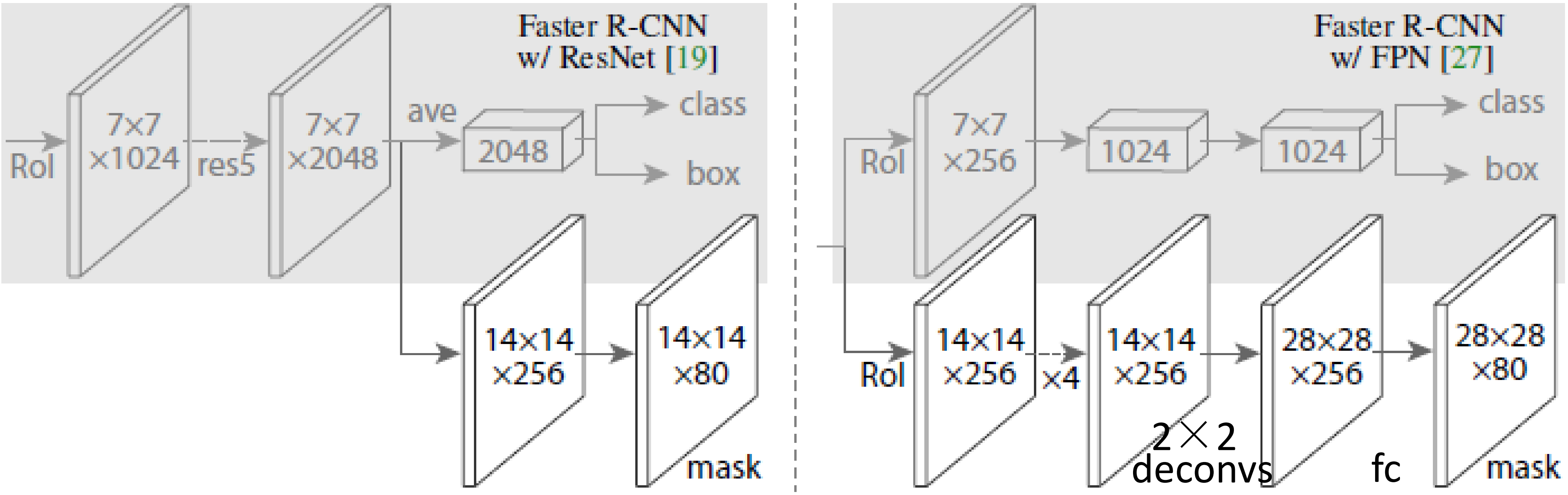


(c) Pyramidal feature hierarchy

(d) Feature Pyramid Network

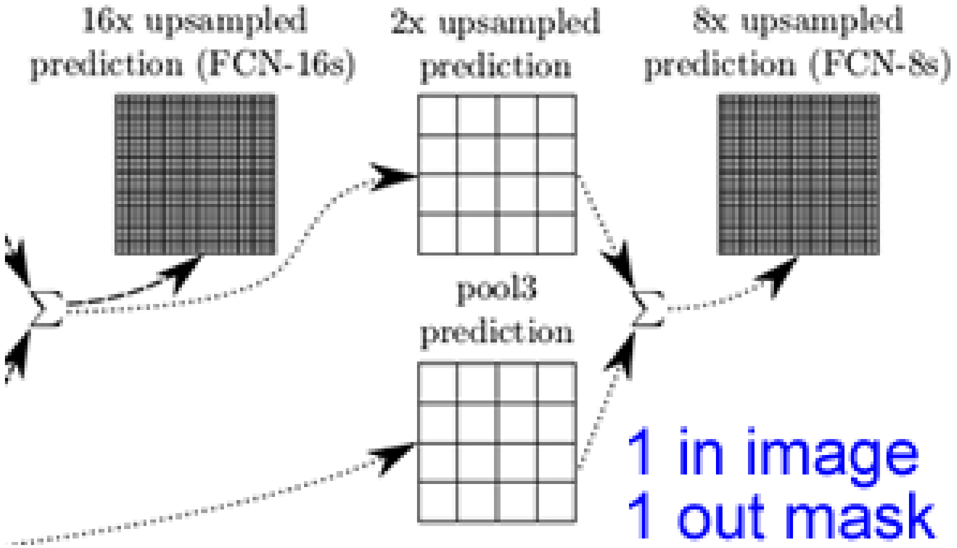
SSD（Single Shot Detector）

Head Architecture



Mask

FCIS

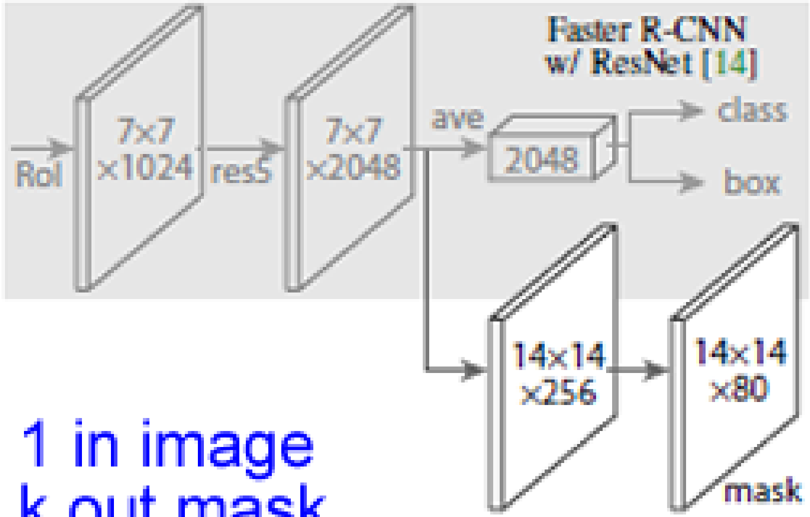


1 in image
1 out mask

多分类的Softmax with entropy loss

VS

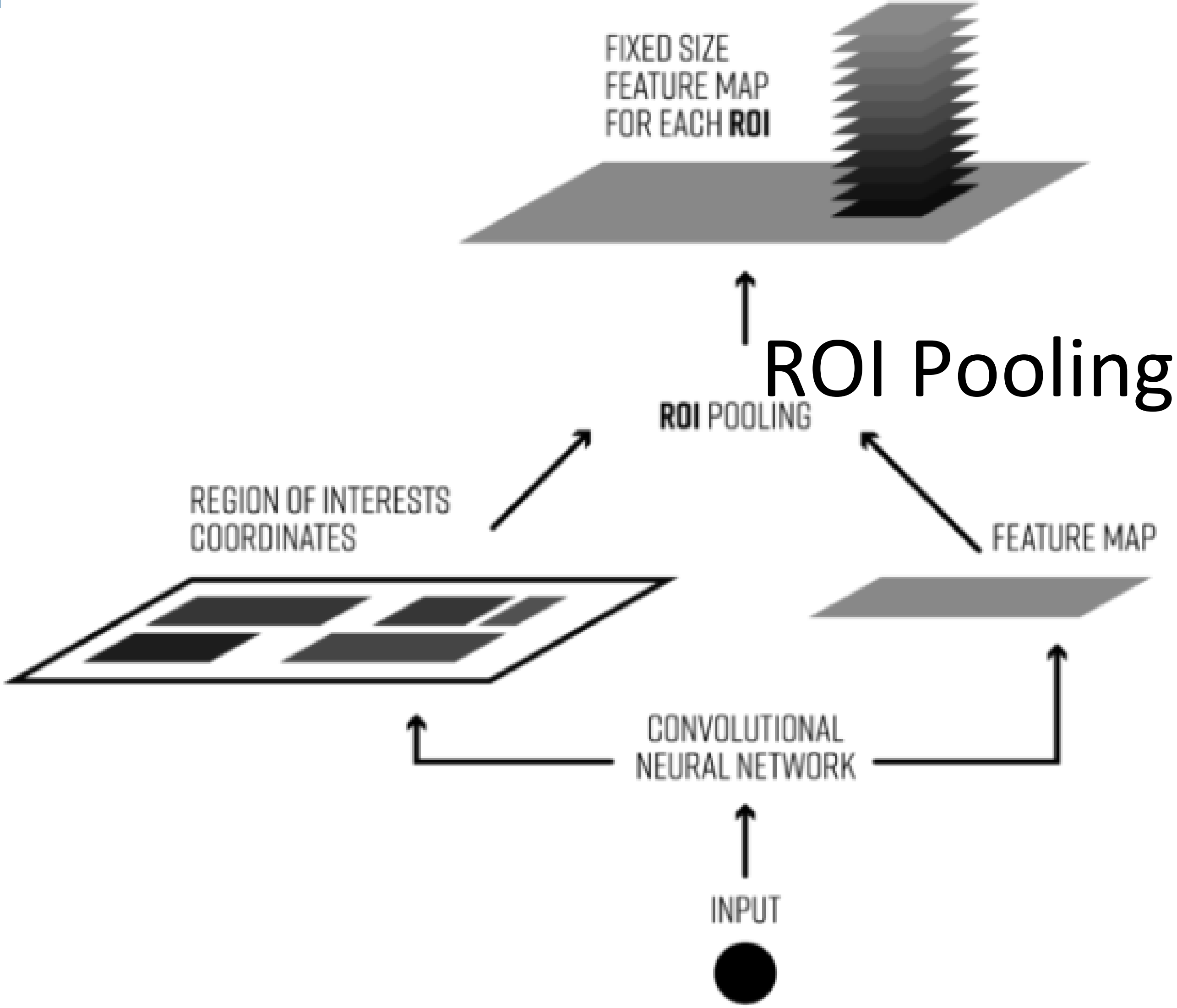
Mask R-CNN



1 in image
k out mask

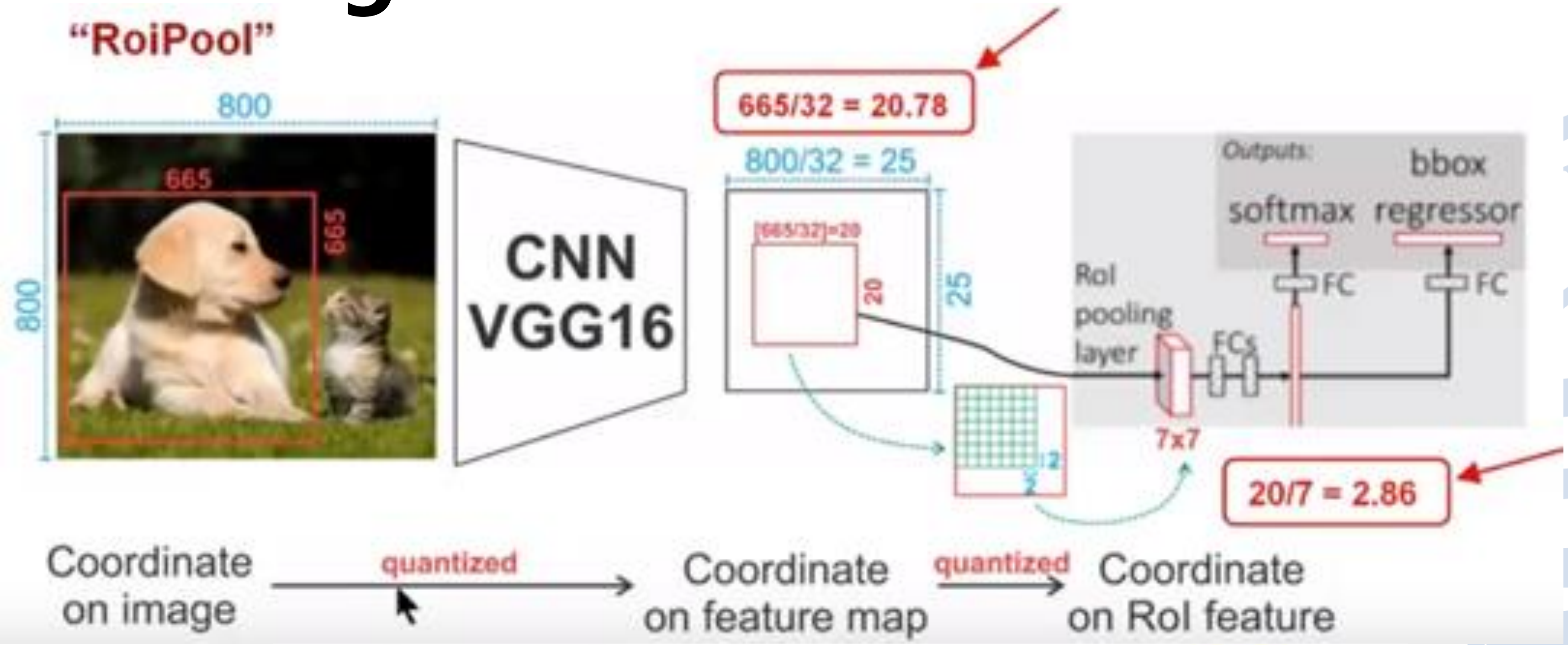
Sigmoid binary cross-entropy loss

FPN（feature pyramid networks）

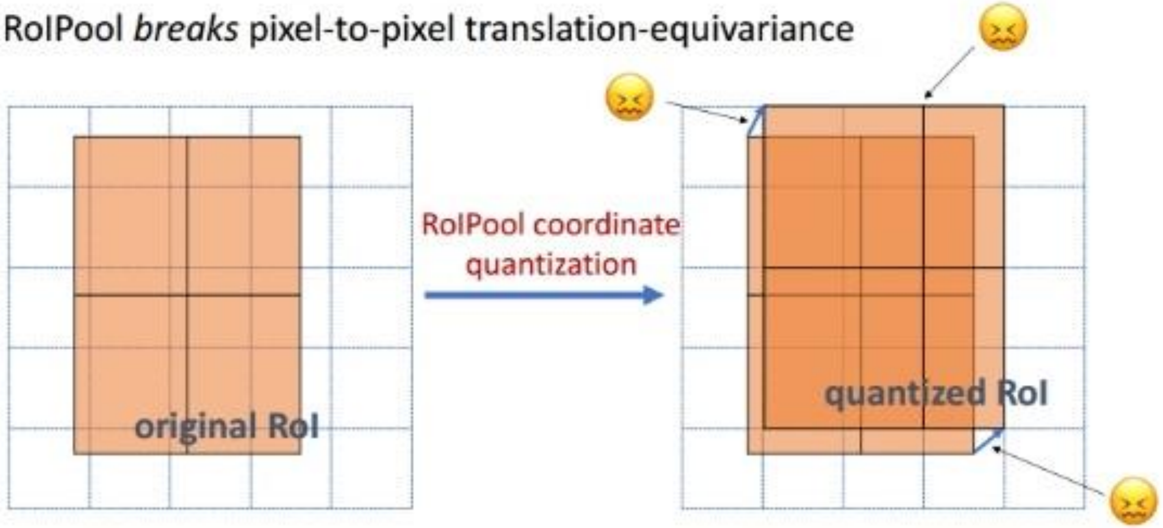


ROI Pooling → ROI Wrap → ROI Align

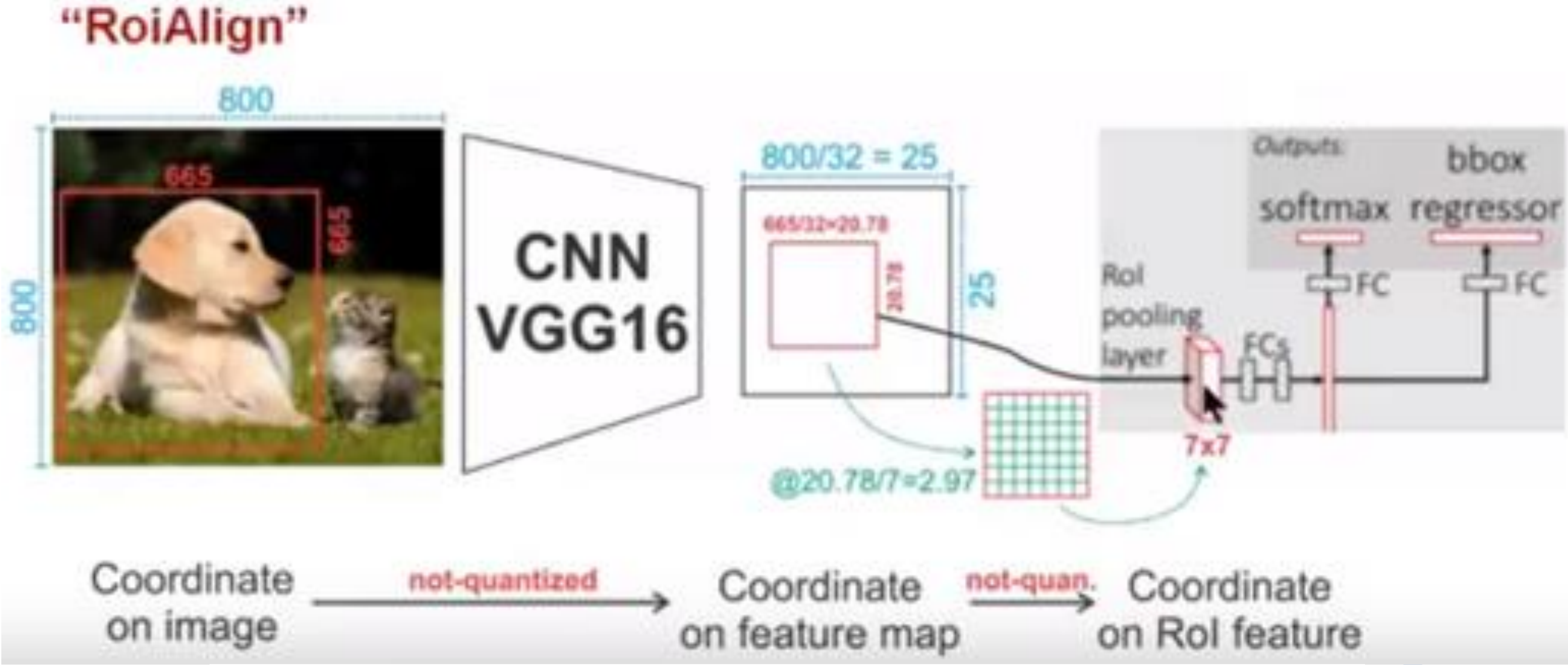
ROI Pooling

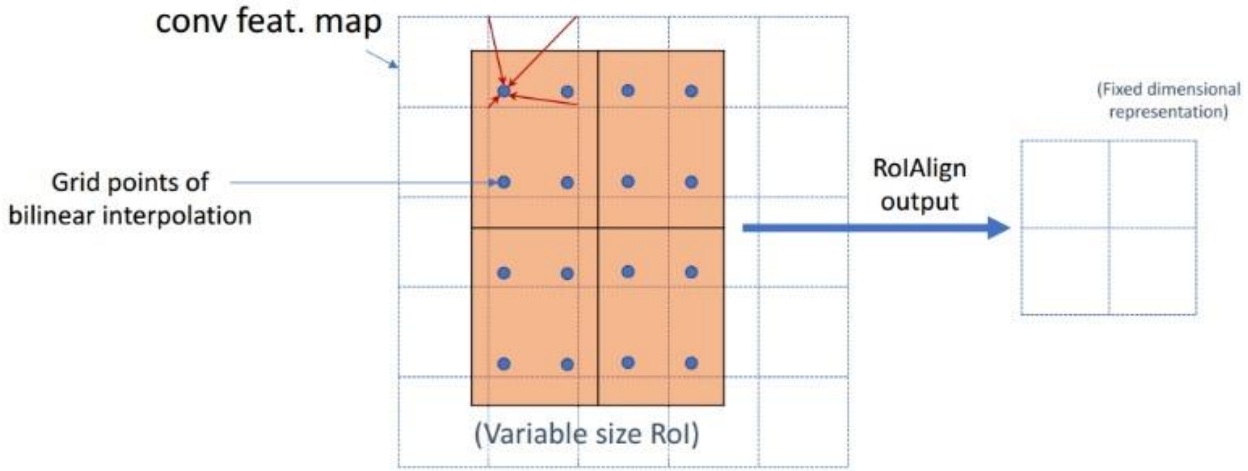


- RoIPool *breaks* pixel-to-pixel translation-equivariance

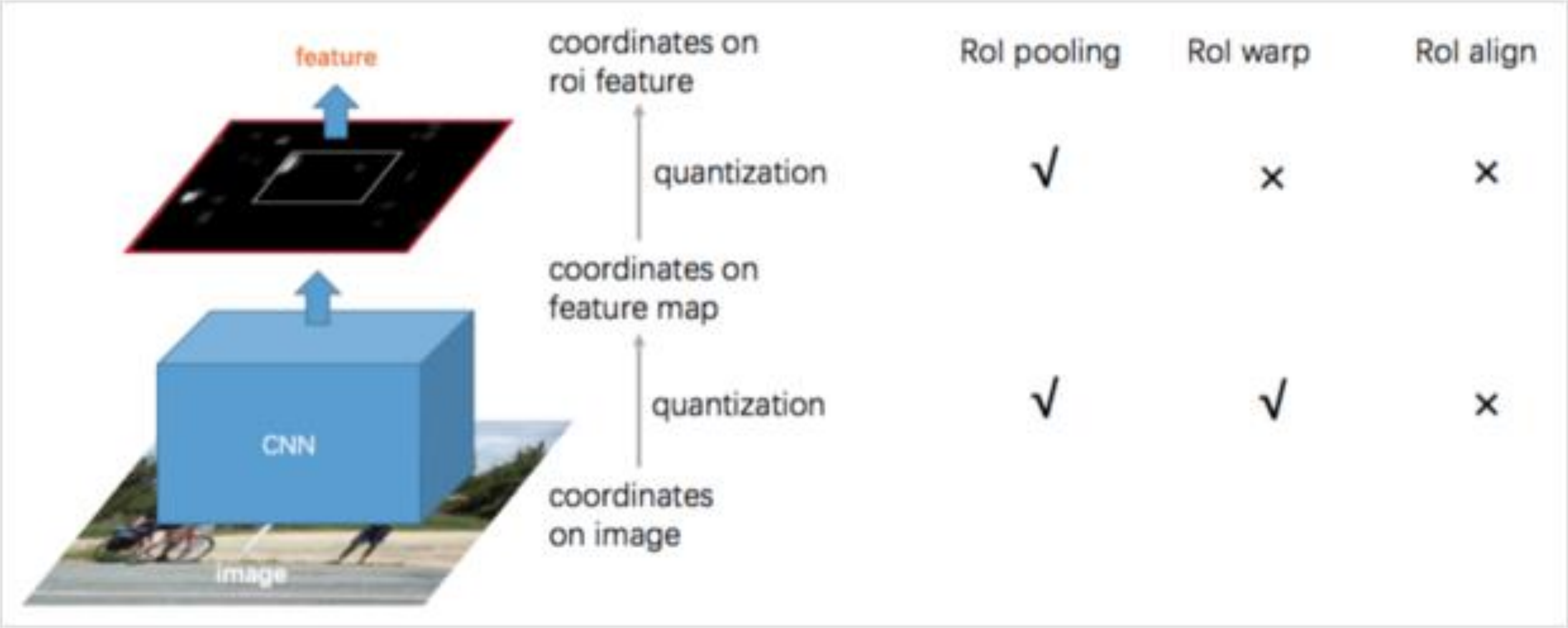


Network Architecture

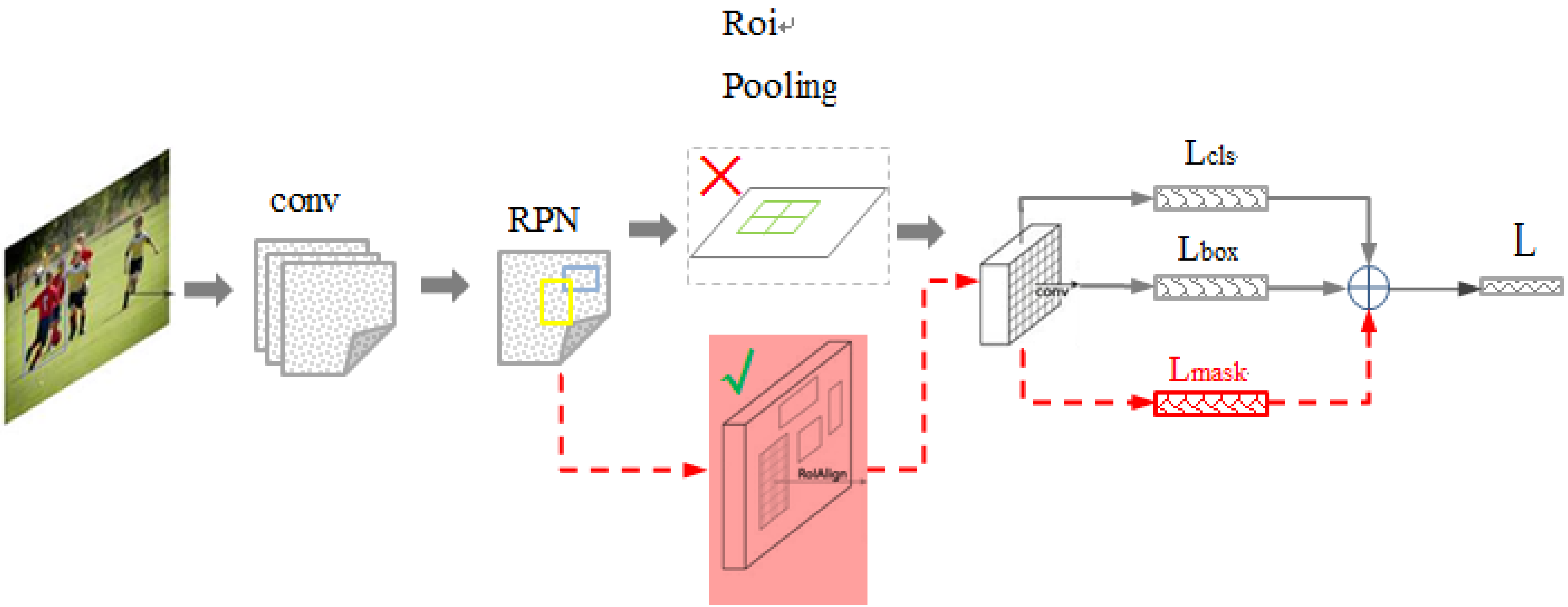




Network Architecture



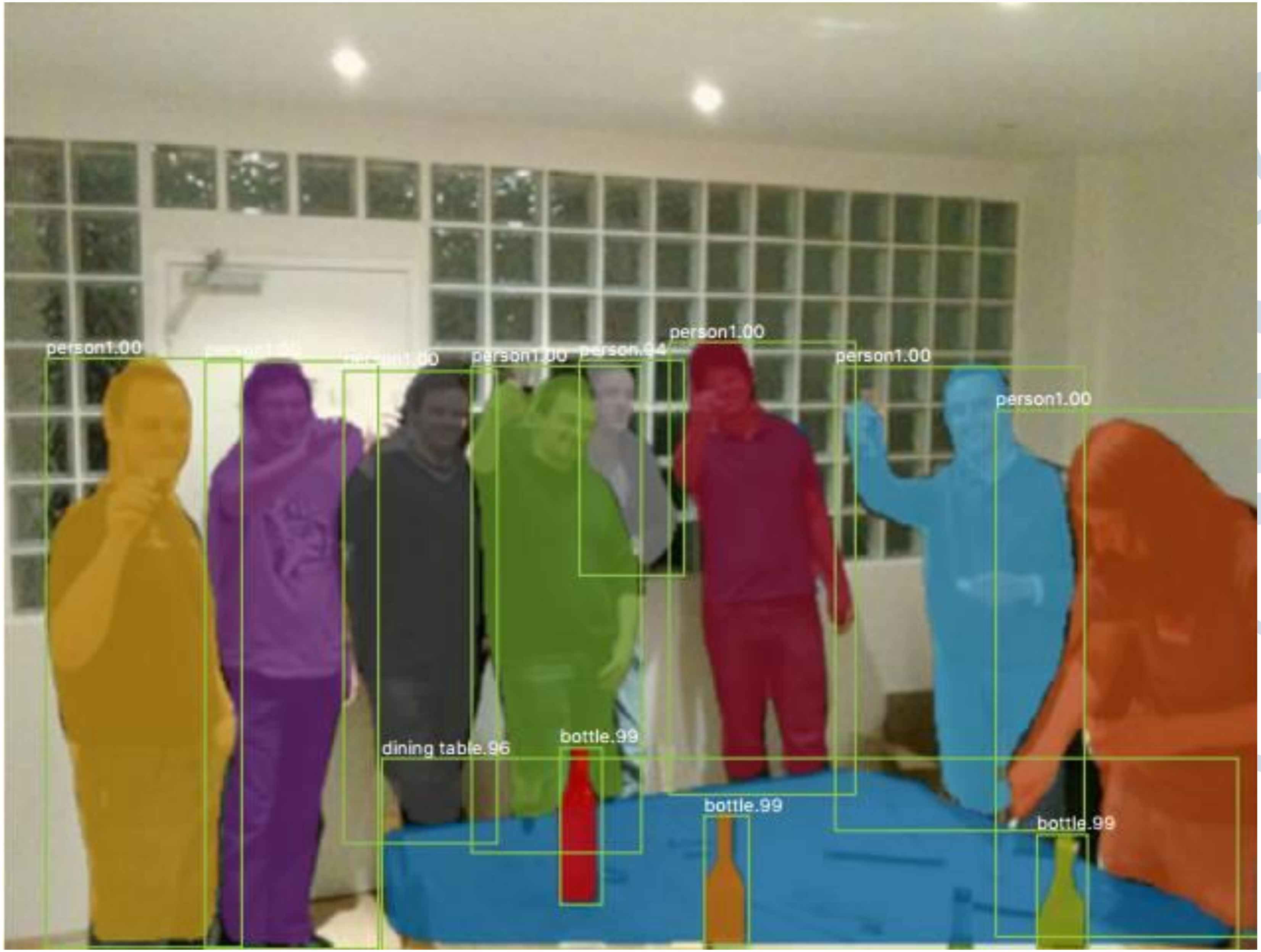
Network Architecture



Experiments

4

Experiments



Results

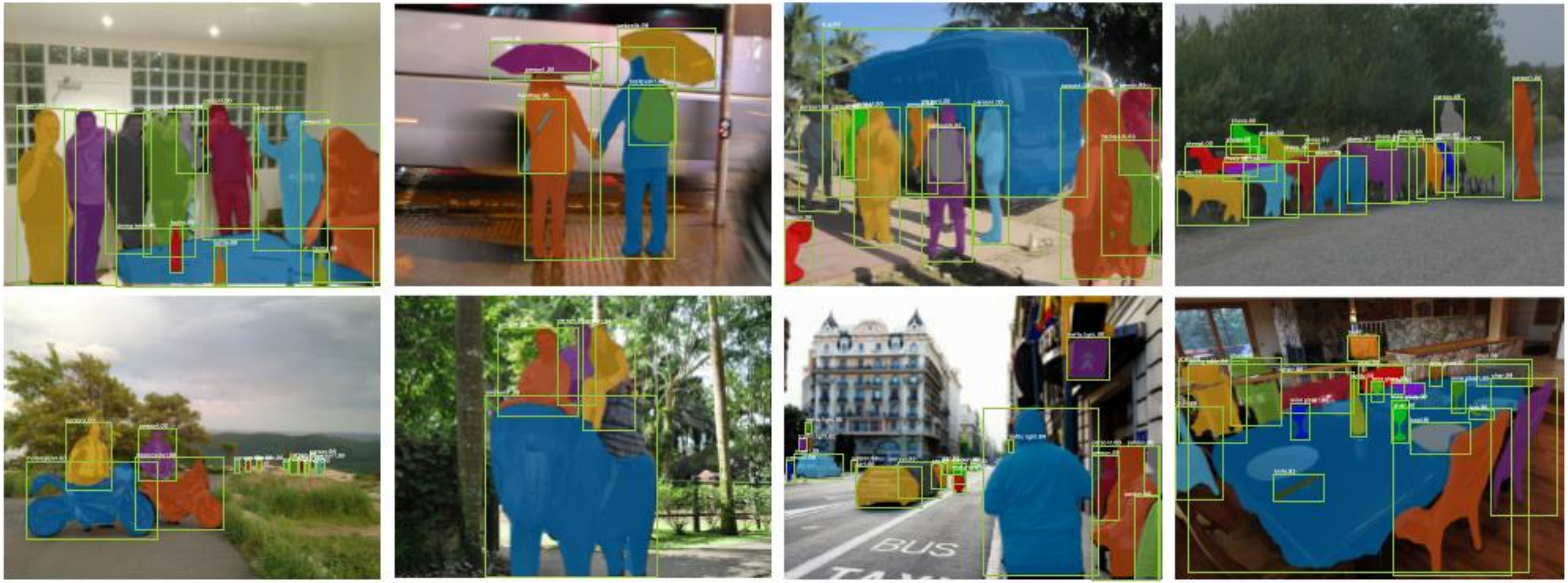


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

Results

Figure 4. 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).

Results

	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 [35]. All entries are *single-model* results.

Experiments

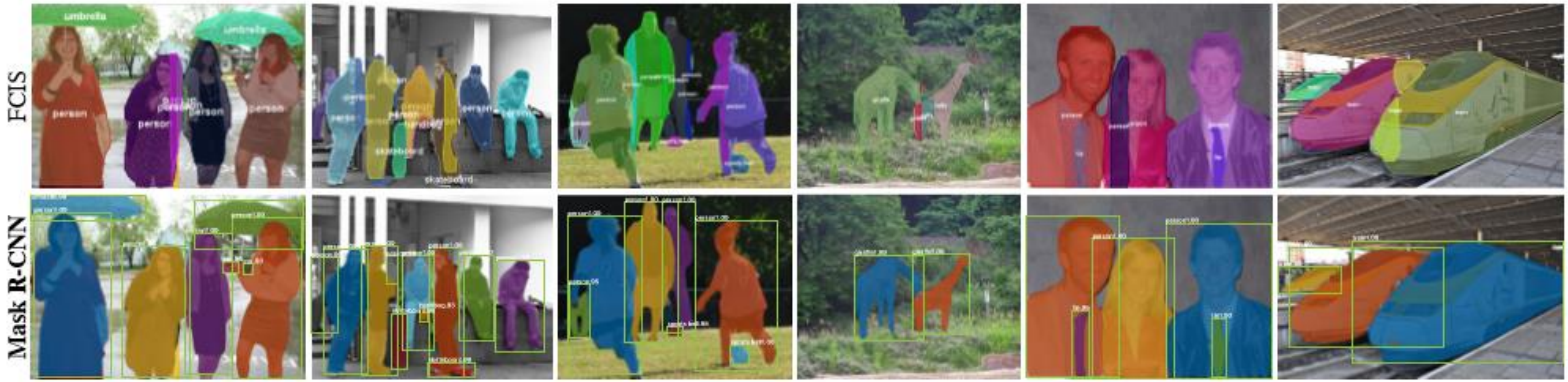


Figure 5. FCIS+++ [26] (top) *vs*. Mask R-CNN (bottom, ResNet-101-FPN). FCIS exhibits systematic artifacts on overlapping objects.

Experiments

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 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 massive accuracy gaps.

	mask branch	AP	AP_{50}	AP_{75}
MLP	fc: 1024→1024→80·28^2	31.5	53.7	32.8
MLP	fc: 1024→1024→1024→80·28^2	31.5	54.0	32.6
FCN	conv: 256→256→256→256→256→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.

Table 2. **Ablations** for Mask R-CNN. We train on `trainval35k`, test on `minival`, and report *mask* AP unless otherwise noted.

Experiments

	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 [37]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [36]	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}).

Human Pose Estimation

Figure 6. 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.

	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 [31]†	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**

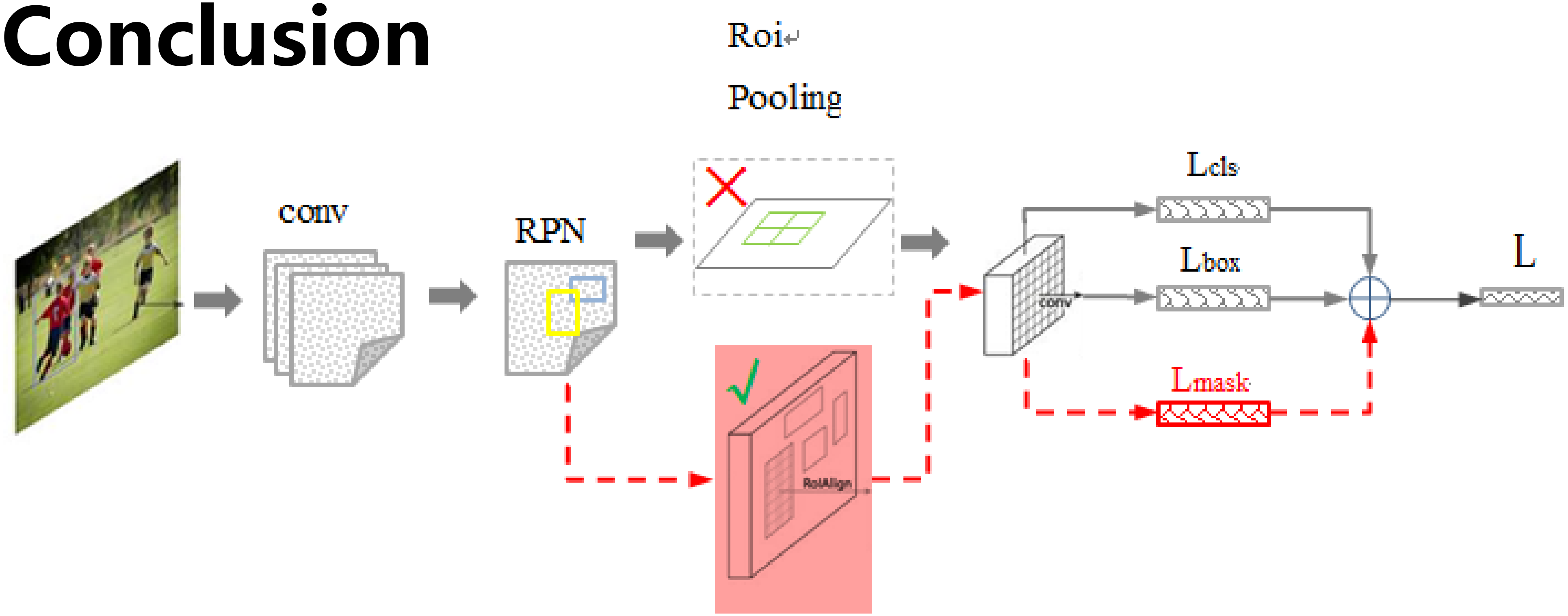
Cityscapes

	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
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**

Table 7. Results on Cityscapes val ('AP [val]' column) and test (remaining columns) sets. Our method uses ResNet-50-FPN.

Figure 7. Mask R-CNN results on Cityscapes test (32.0 AP). The bottom-right image shows a failure prediction.

Conclusion



- ➢ Advantages：
- Faster
- Accuracy
- Simple
- Flexible

- ➢ Improvements：
- Replace ROI Pooling with ROI Align
- Add a mask branch

Reference

[1] He K, Gkioxari G, Dollár P, et al. Mask R-CNN[J]. 2017.
[2] Uijlings J R R, Sande K E A V D, Gevers T, et al. Selective Search for Object Recognition[J]. International Journal of Computer Vision, 2013, 104(2):154-171.
[3] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection[J]. 2016.
[4] Ren S, Girshick R, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137.

THANKS

请各位评委老师批评指正