Deep Learning for Image Instance Segmentation ----Mask R-CNN

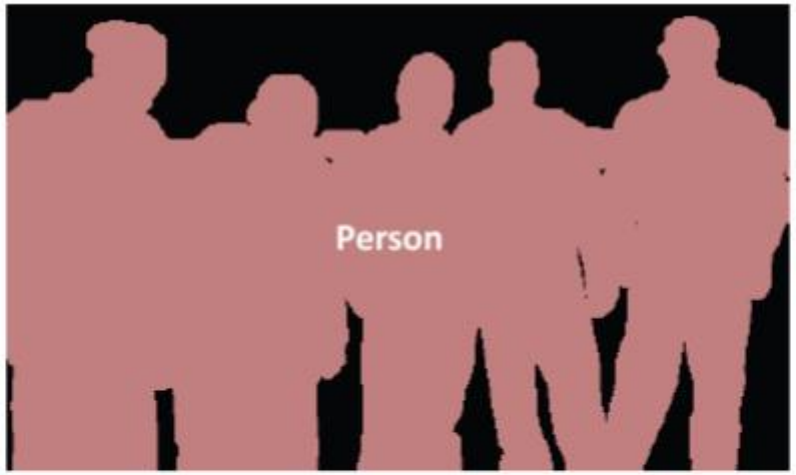
Jianping Fan
Dept of Computer Science
UNC-Charlotte

Course Website:
http://webpages.uncc.edu/jfan/itcs5152.html

Definition of Image Instance Segmentation

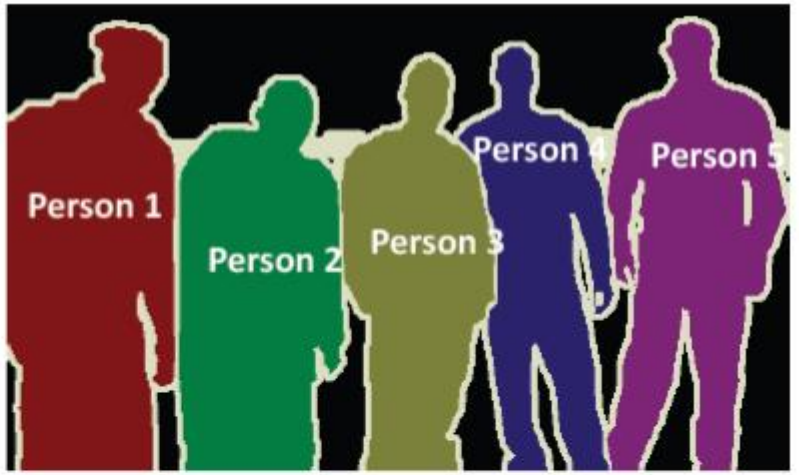
Object Detection

✓



Semantic Segmentation

✓



Instance Segmentation

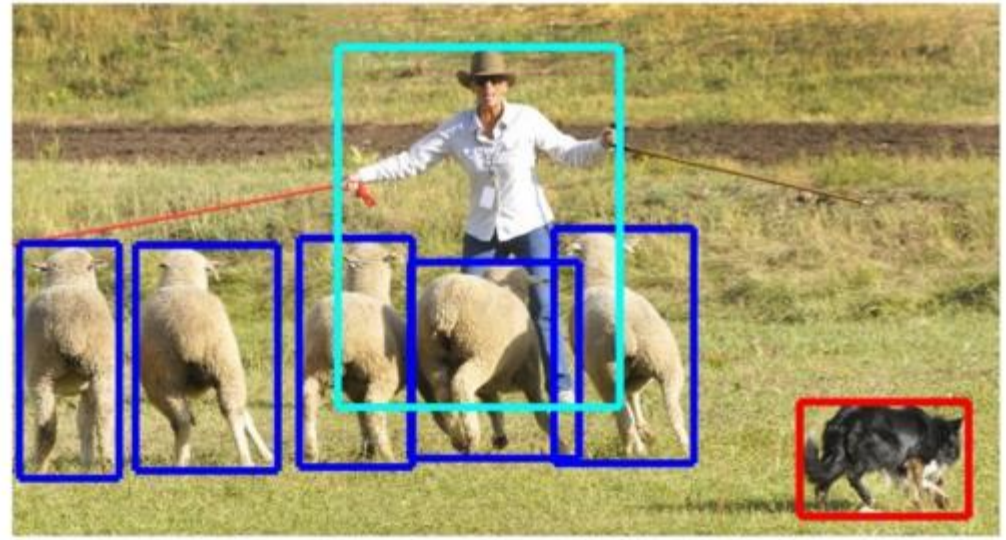
?

Instance segmentation = object detection + semantic segmentation?

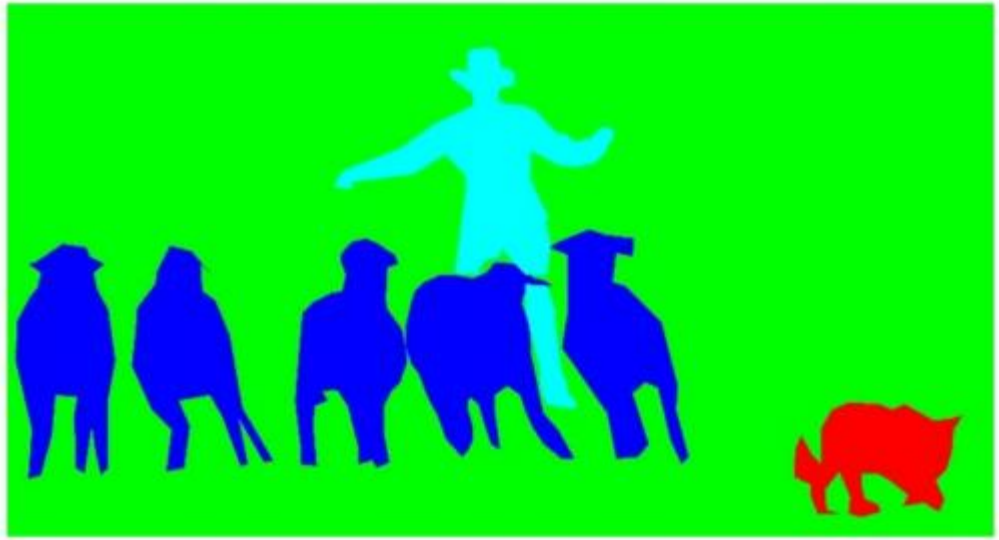
Slide from *Mask R-CNN Tutorial*, K. He. ICCV 2017

Scene understanding

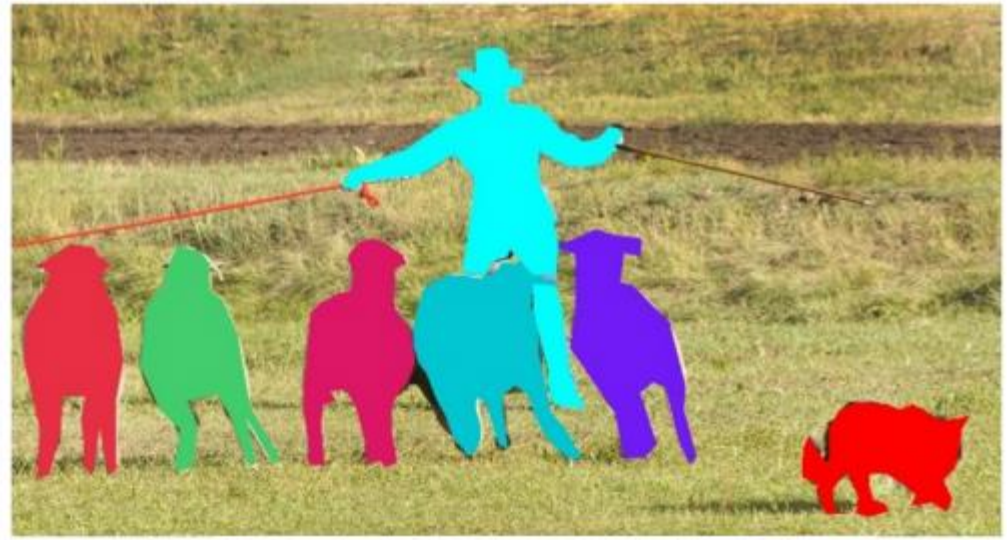
Image classification



Object detection



Semantic segmentation



Instance segmentation

Instance-level Object Understanding Today

He, Gkioxari, Dollár, Girshick. Mask R-CNN. In ICCV 2017

Background Review

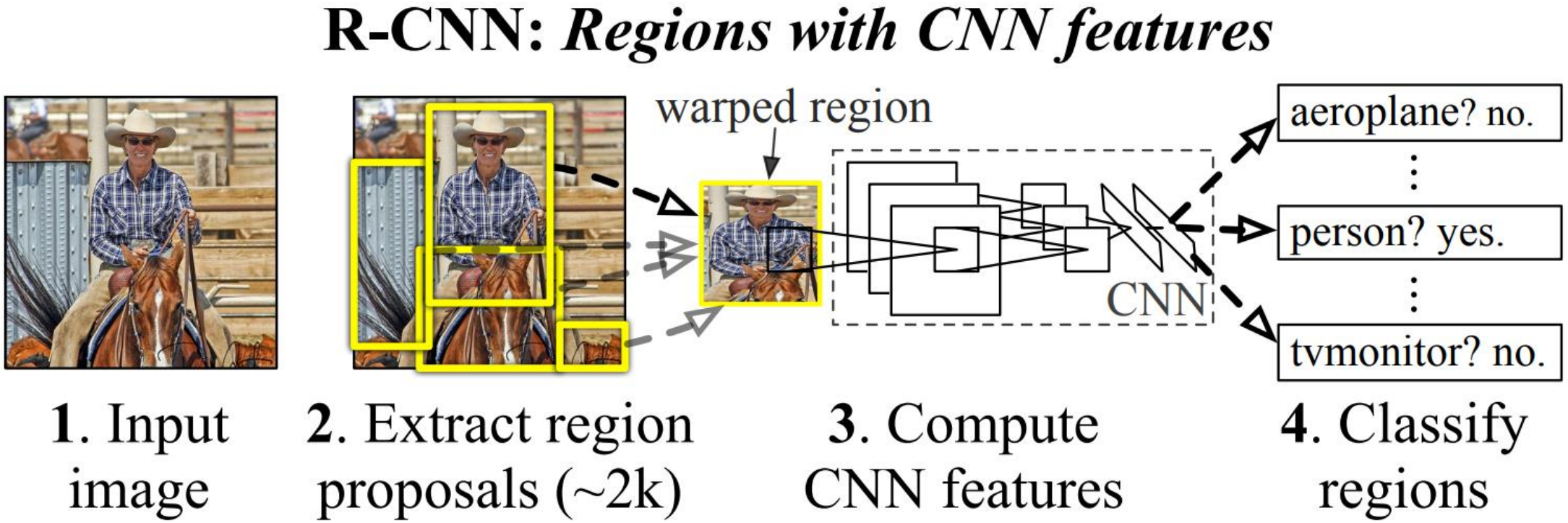
R-CNN

- R-CNN [4]: The Region-based CNN (R-CNN)
  - Replace sliding windows with "**selective search**" region proposals(Uijilings et al. IJCV 2013)
  - Extract rectangles around regions and **resize to 227x227 pixels**
  - Extract features with fine-tuned CNN (that was initialized with network trained on ImageNet before training)
  - Classify last layer of network features with SVM, refine bounding box localization (bbox regression) simultaneously

Slide credit to Ke-Shuan Cheng

R-CNN

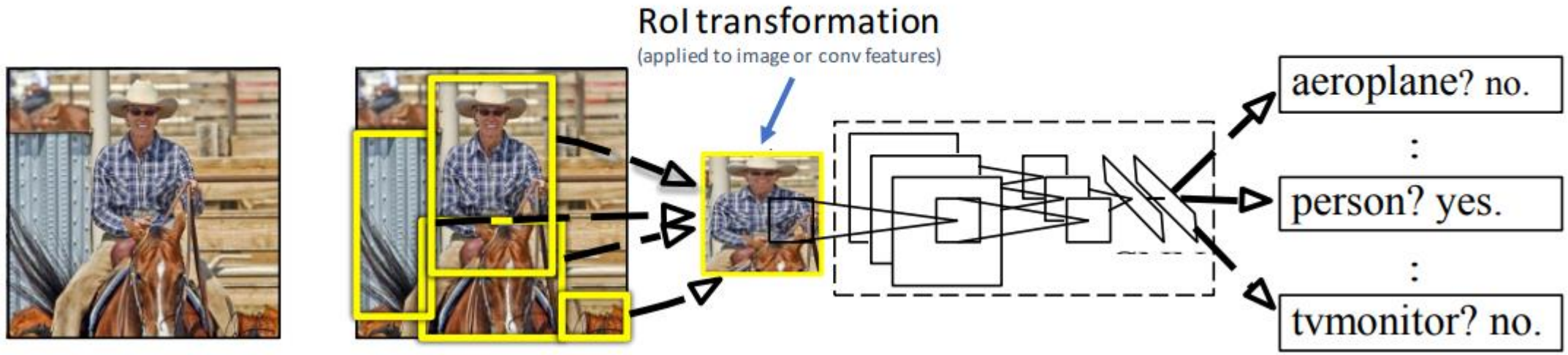
- R-CNN: The Region-based CNN (R-CNN)



Region warping is performed for fixed size features

Slide credit to Ke-Shuan Cheng

R-CNN-style Approach to Object Detection



Input image

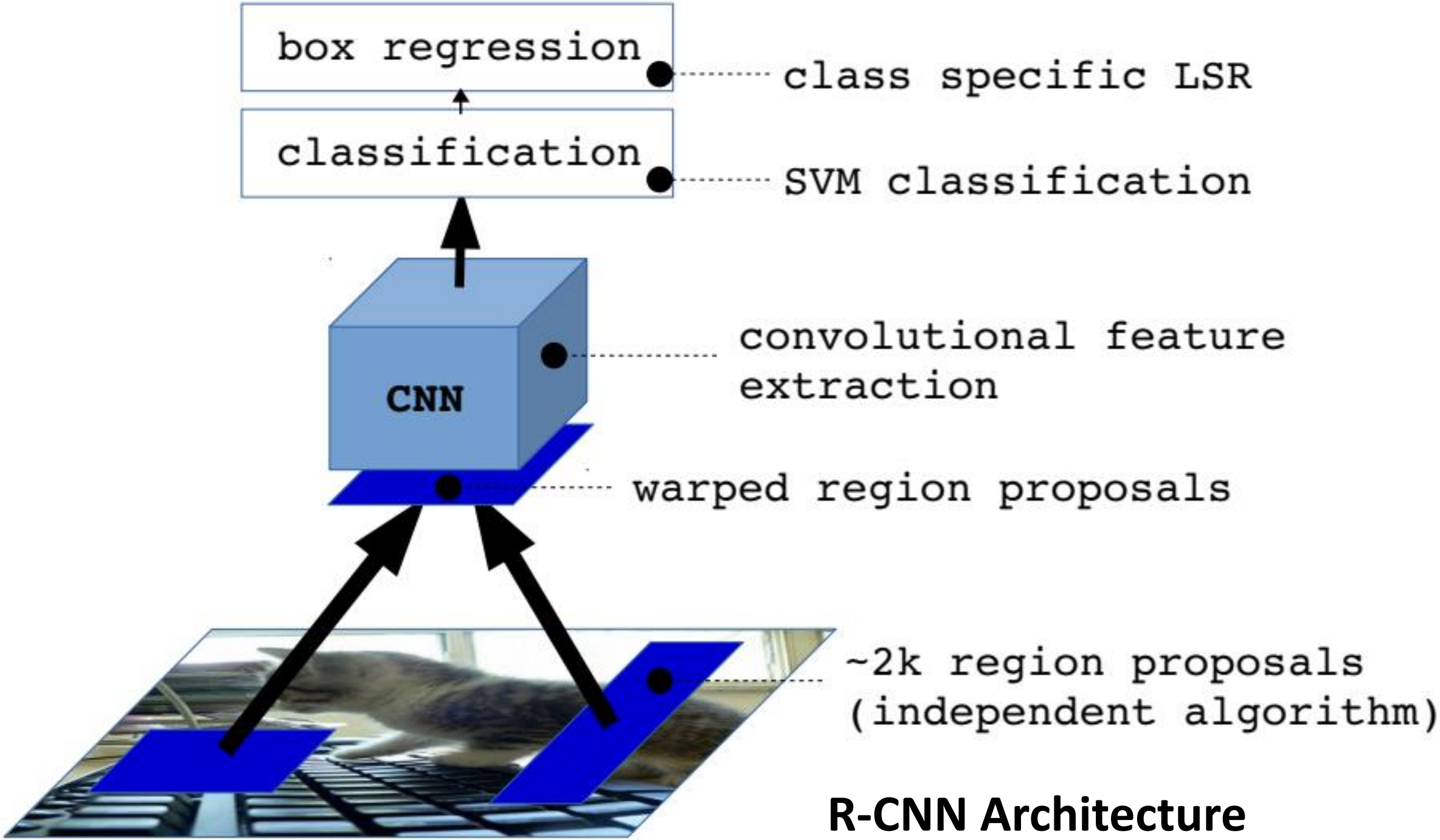
Object / region proposals (RoIs)
(external or internal to network)

Deep Learning-based region classifier

Detection = Region classification + box regression + …

General formula for **R**egion-based **CNN** models

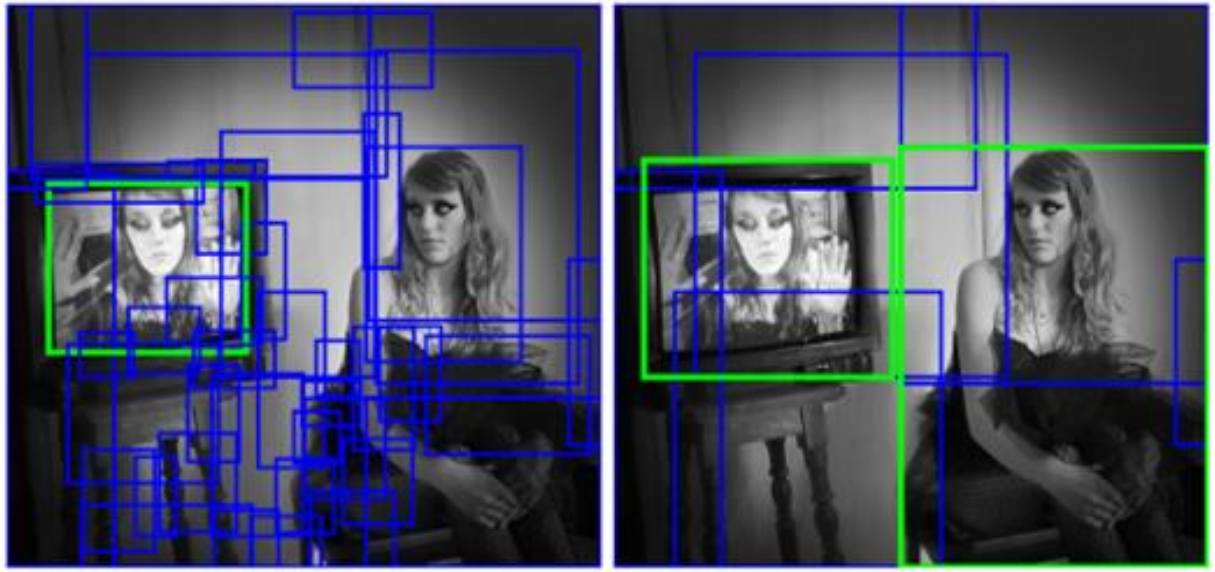
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014
Uijlings, van de Sande, Gevers, Smeulders. Selective Search for Object Recognition. IJCV 2013



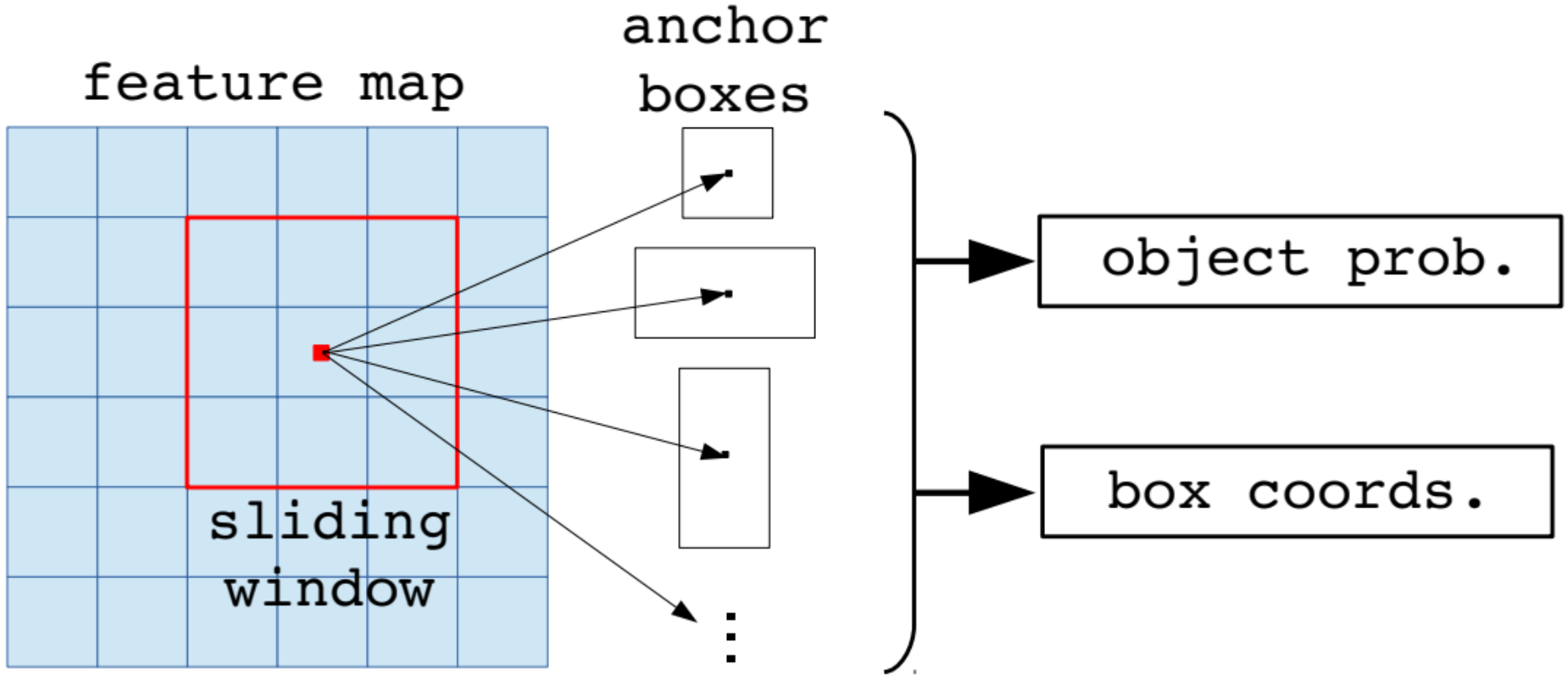
R-CNN Architecture

Regional Proposal Network (RPN)

- Foreground vs Background
- Bounding Box regression
- Feed bounding boxes into Fast RCNN

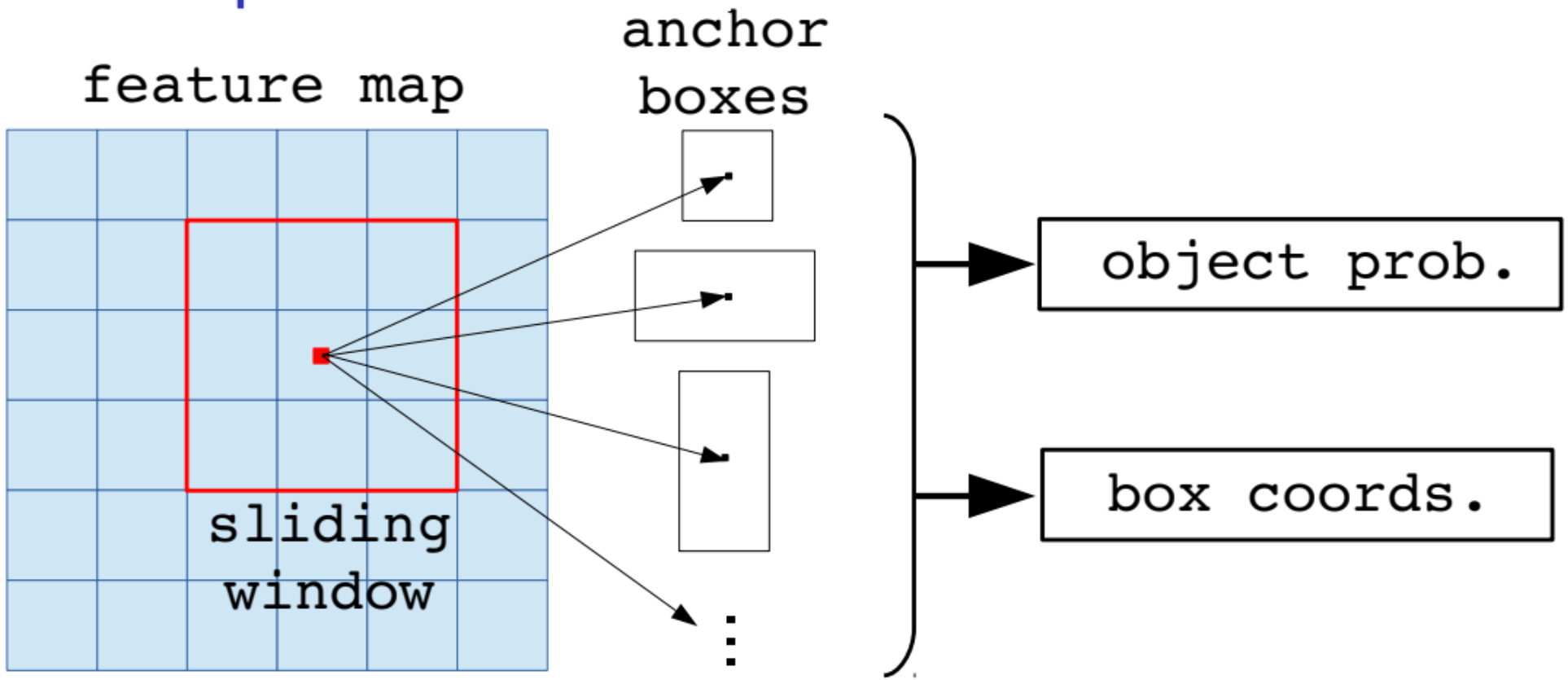


Region Proposal Network



- Shared conv layers with main model
- $n \times n$ sliding window, large receptive field
- Parallel branches:
  - object probability classification
  - box regression

Region Proposal Network

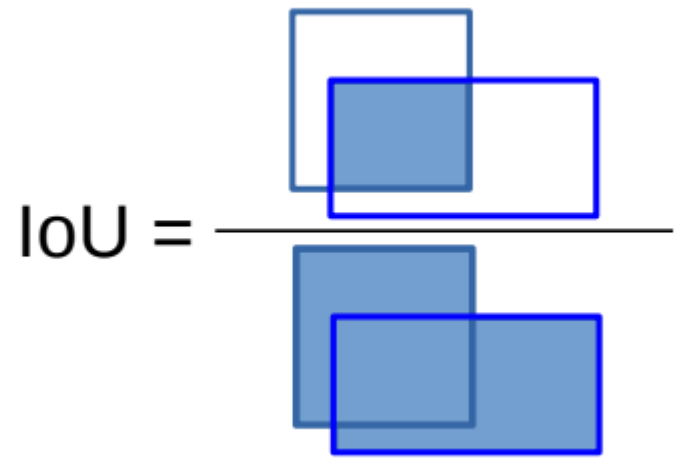


Anchor boxes:

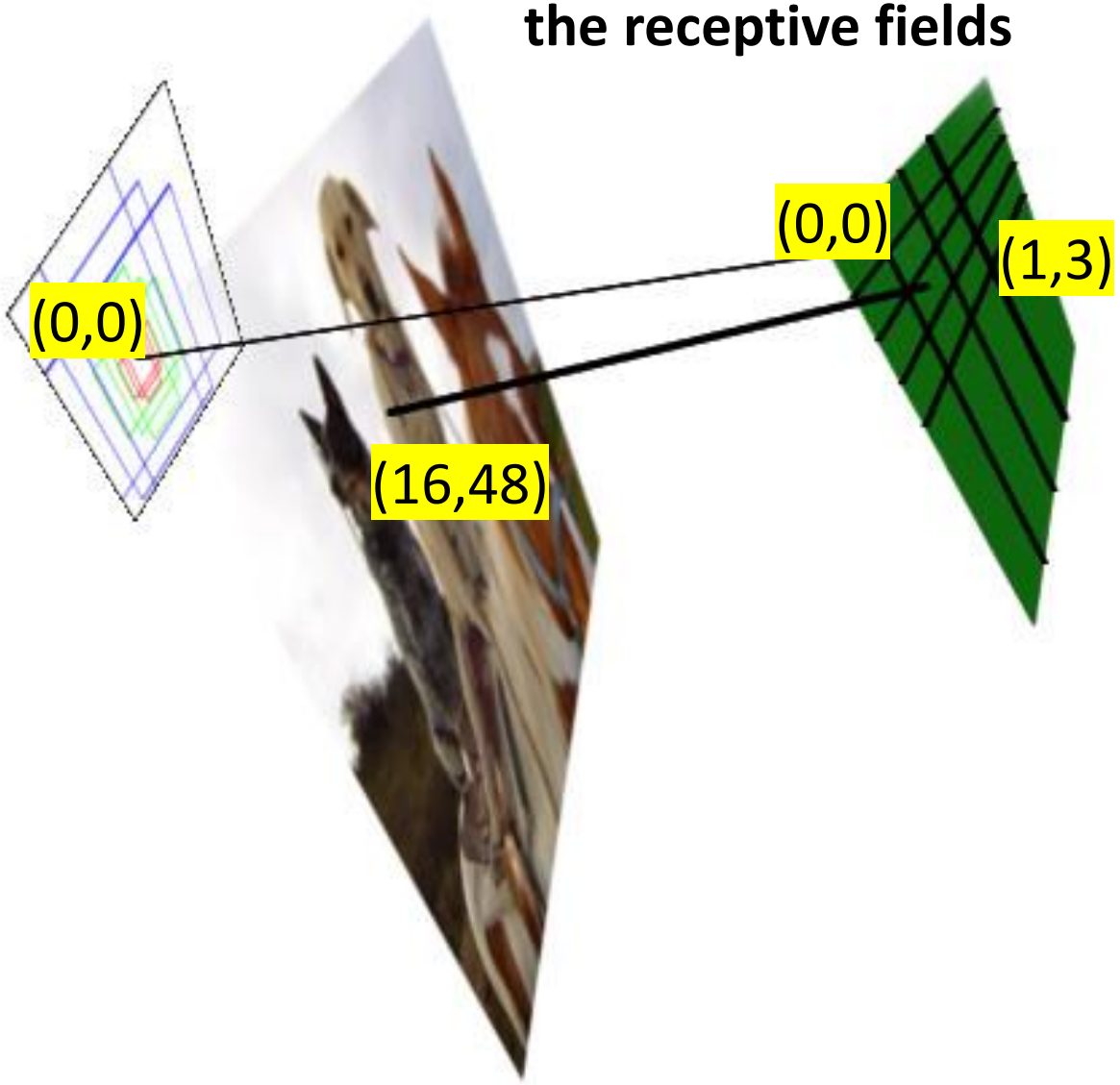
- for every window position, k region prototypes
- multiple scales, e.g. 128^2, 256^2, 512^2
- multiple ratios, e.g. $1:1$, $1:2$, $2:1$

Region Proposal Network

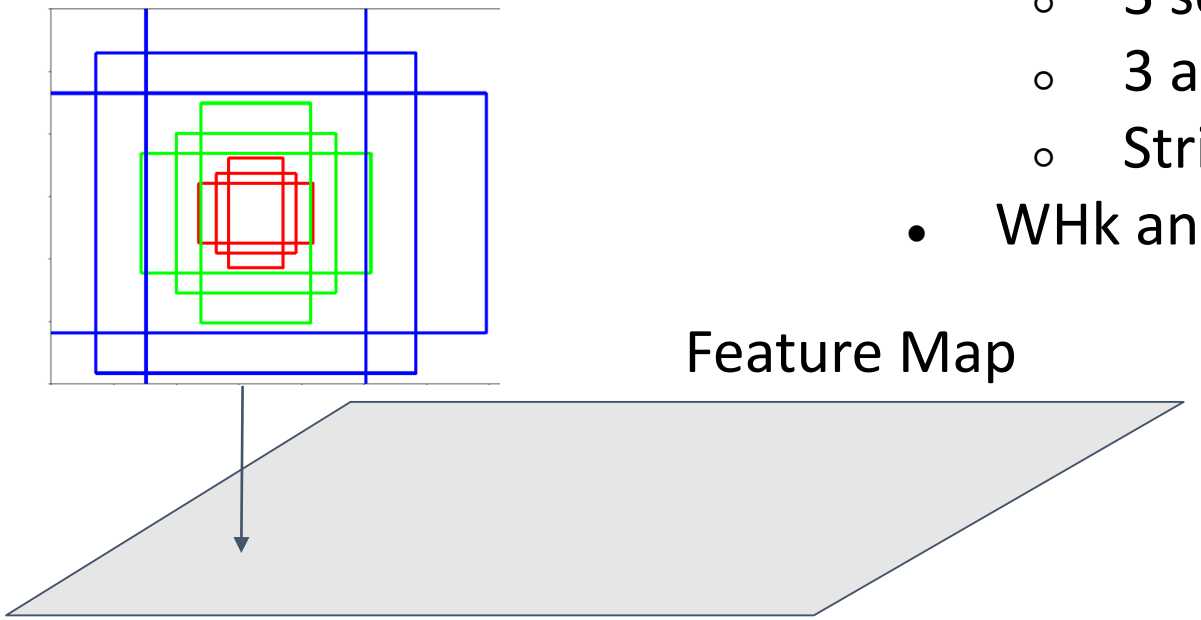
- Multiple anchor scales and ratios $\rightarrow$ single scale images
- Proposal evaluation based on Intersection over Union with ground truth boxes:
  - best regions are kept as positive examples
  - worst (IoU < 0.3) are kept as negatives for training



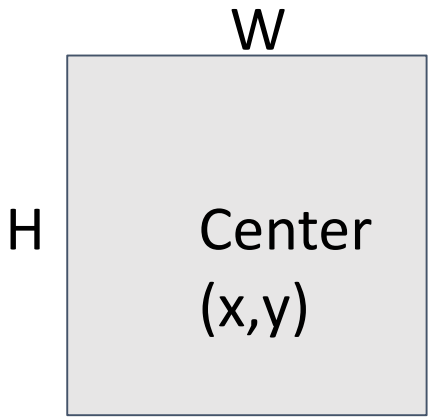
Mapping the center of the receptive fields

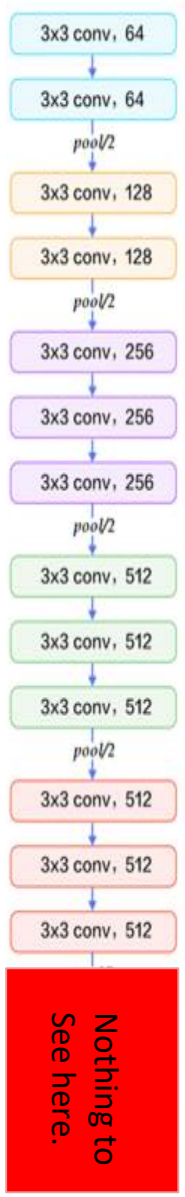


Anchor Boxes

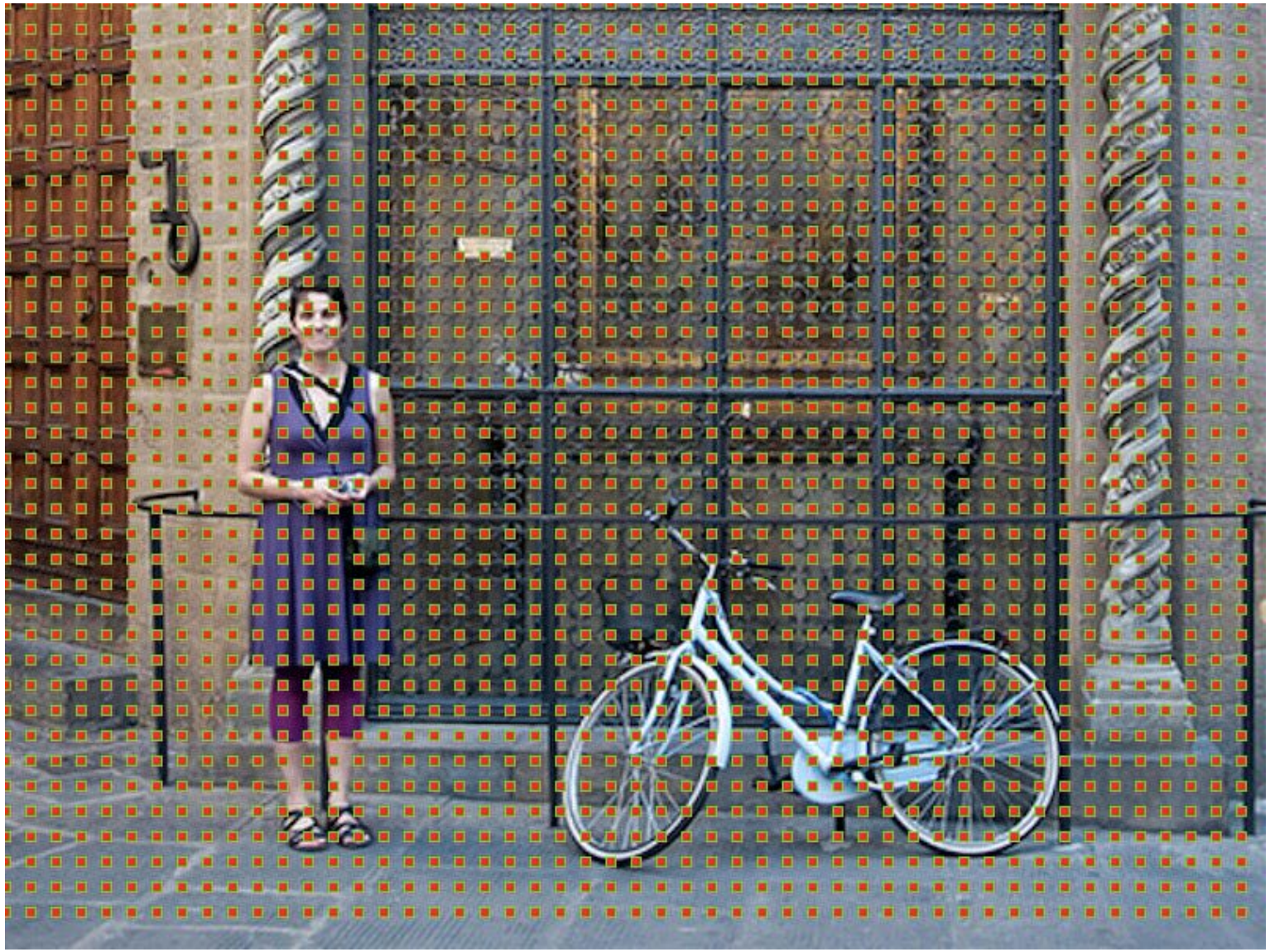


- k anchor boxes
  - 3 scales (8,16, 32)
  - 3 aspect ratios (.5, 1, 2)
  - Stride 16
- WHk anchors



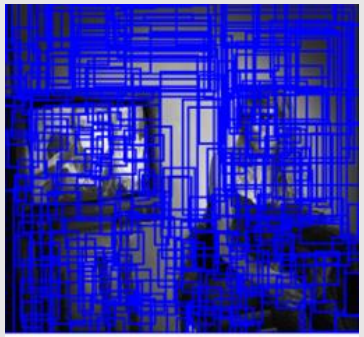


Anchor Boxes

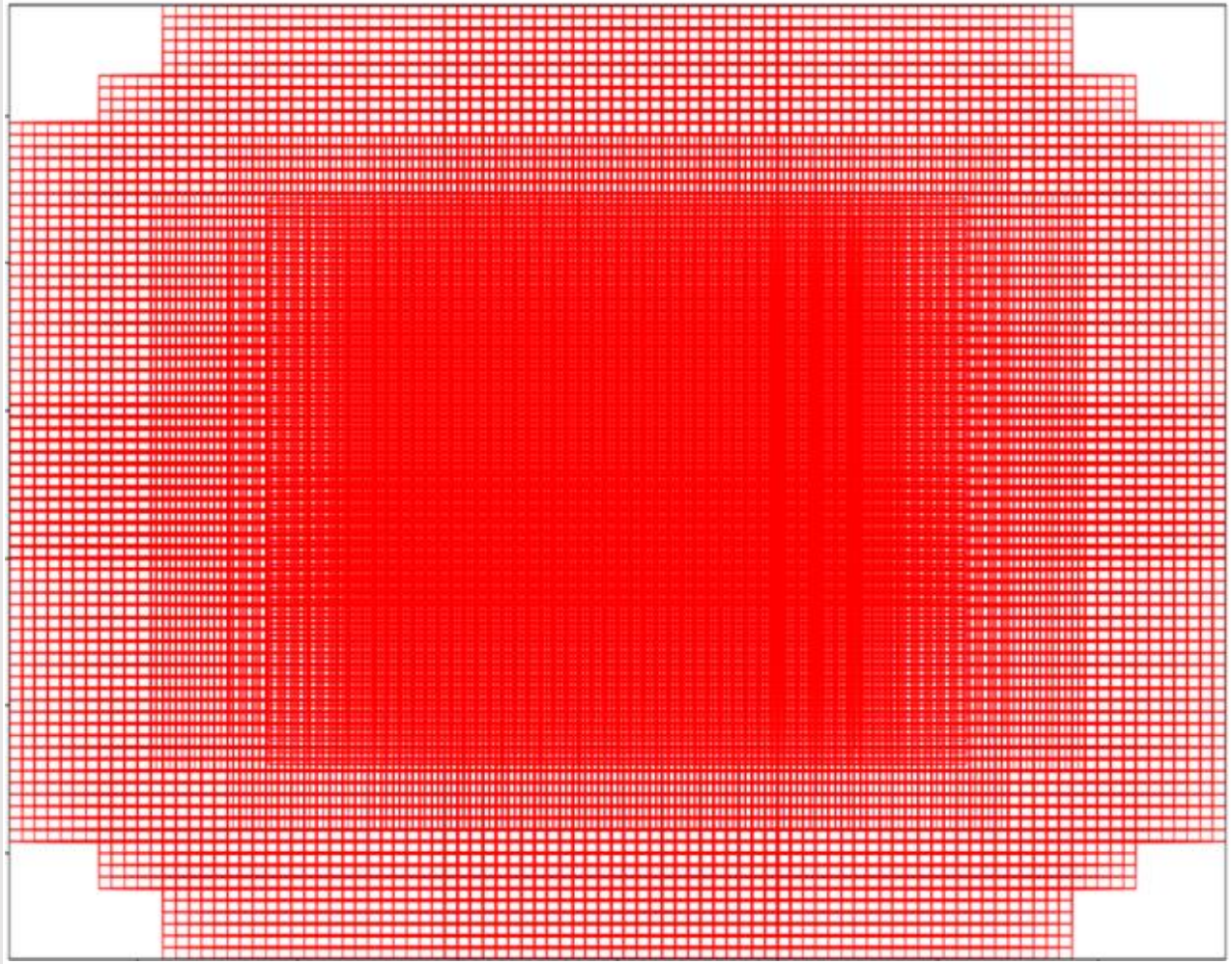


Anchor Boxes

of grid locations: $\frac{800}{16} * \frac{600}{16} = 1900$

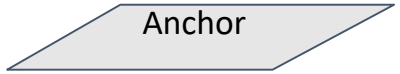


Total number of anchors: 1900*9 = 17100
Some boxes lie outside the image boundary



RPN
(Region Proposal Network)

Object *vs* Not an Object



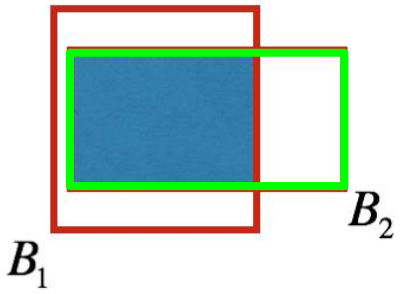
Object = 1 to:

a) Anchors with the highest Intersection-over-Union(IoU)
b) IoU > 0.7 with any ground truth box.

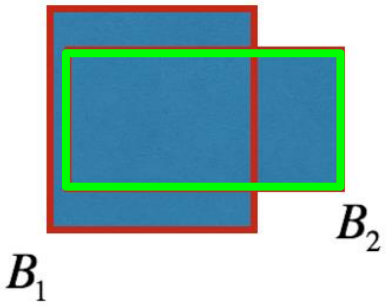
Not object = -1

a) If IoU <0.3

Intersection

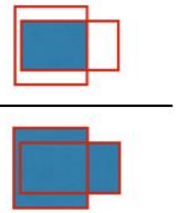


Union



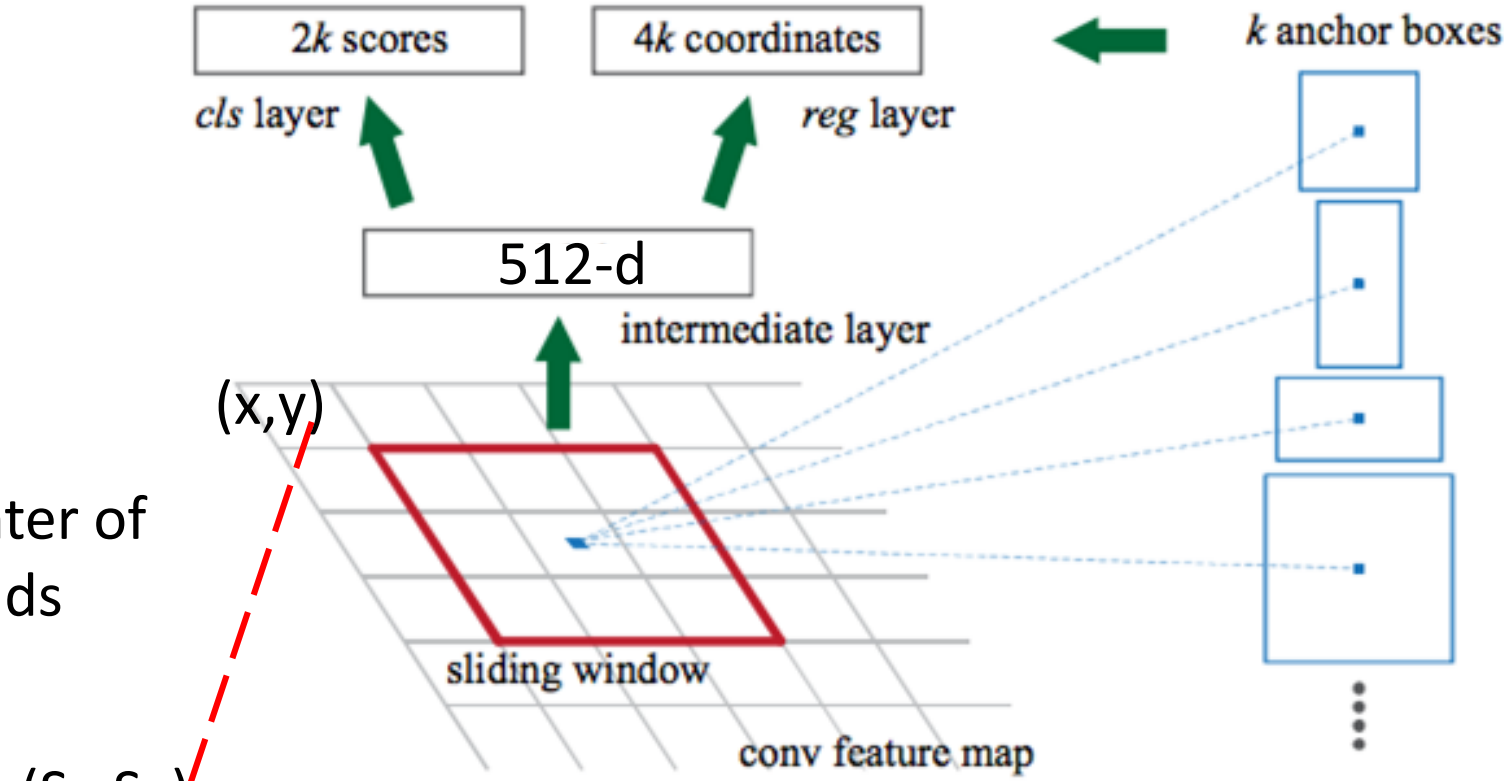
Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} =$$



RPN

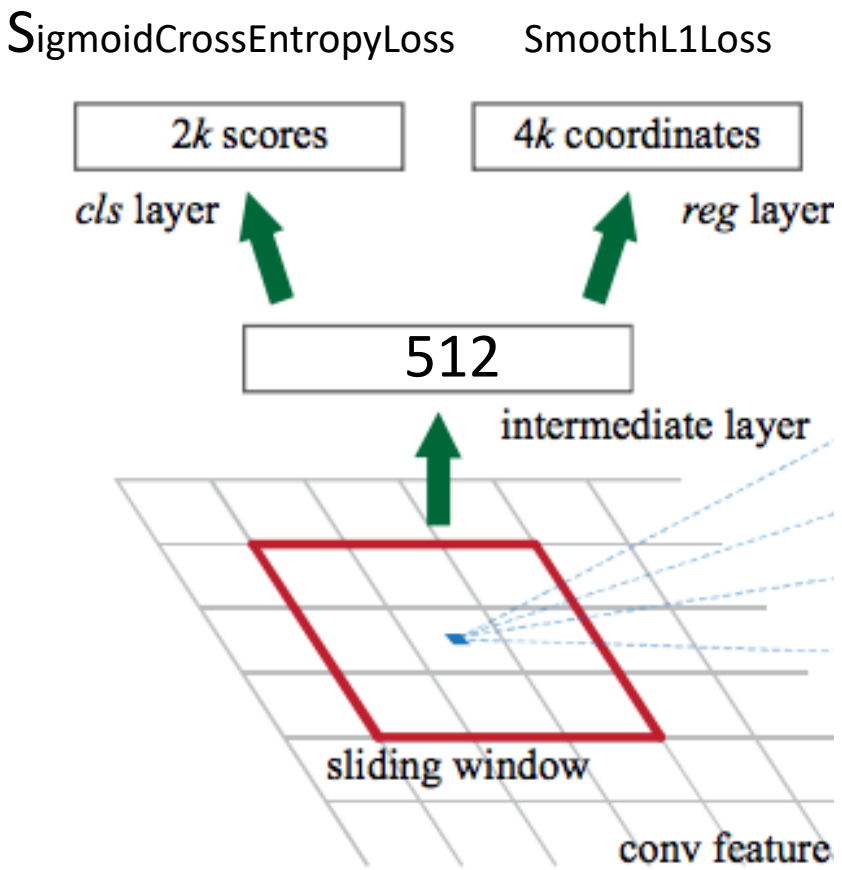
Mapping the center of the receptive fields



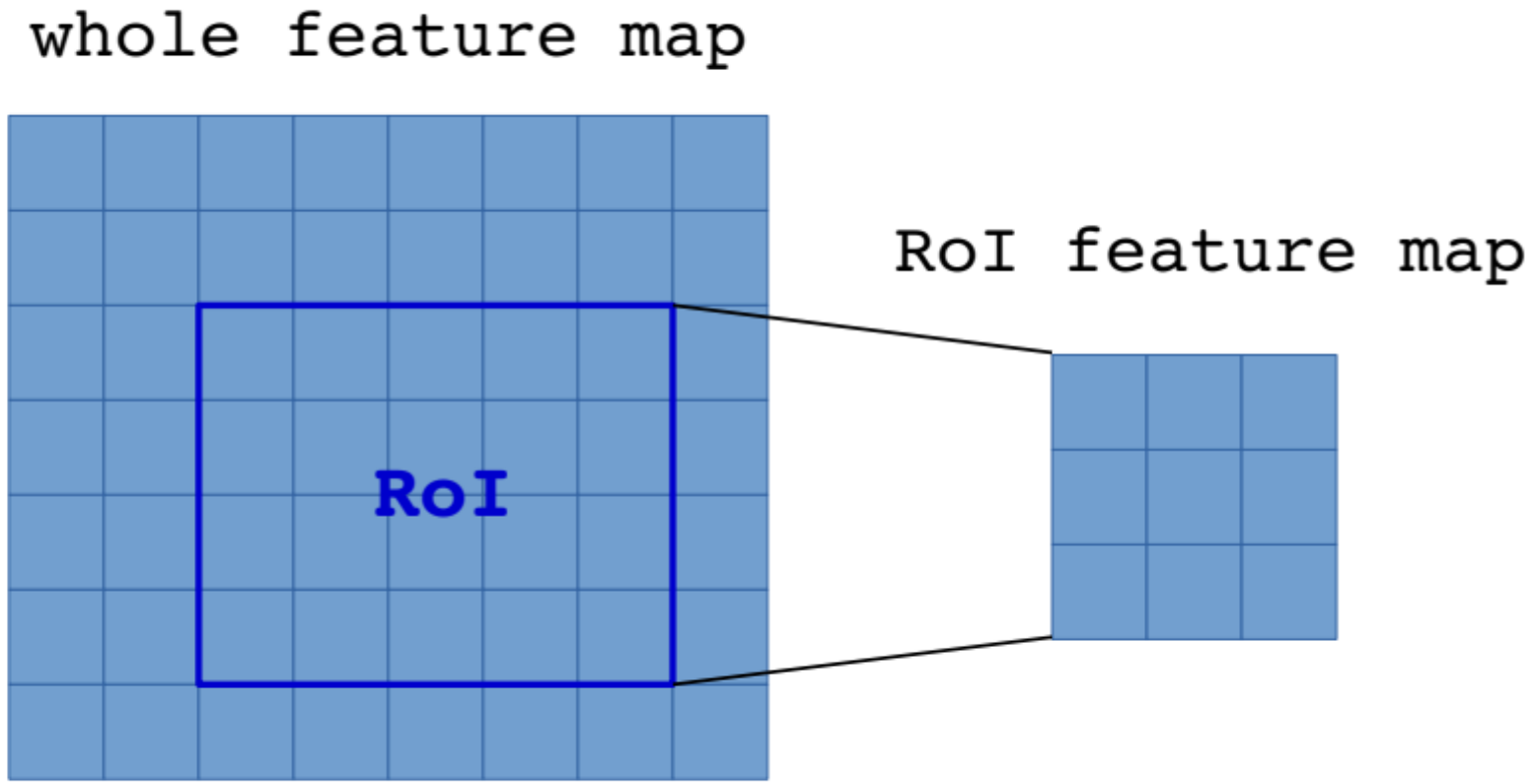
(Sx,Sy)

RPN

(512 × (2 + 4) × 9) parameters for VGG-16)



RoI feature extraction



- RoIPool: quantized bins + pooling

RPN

Multi-task loss:

Mini batch size =256

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

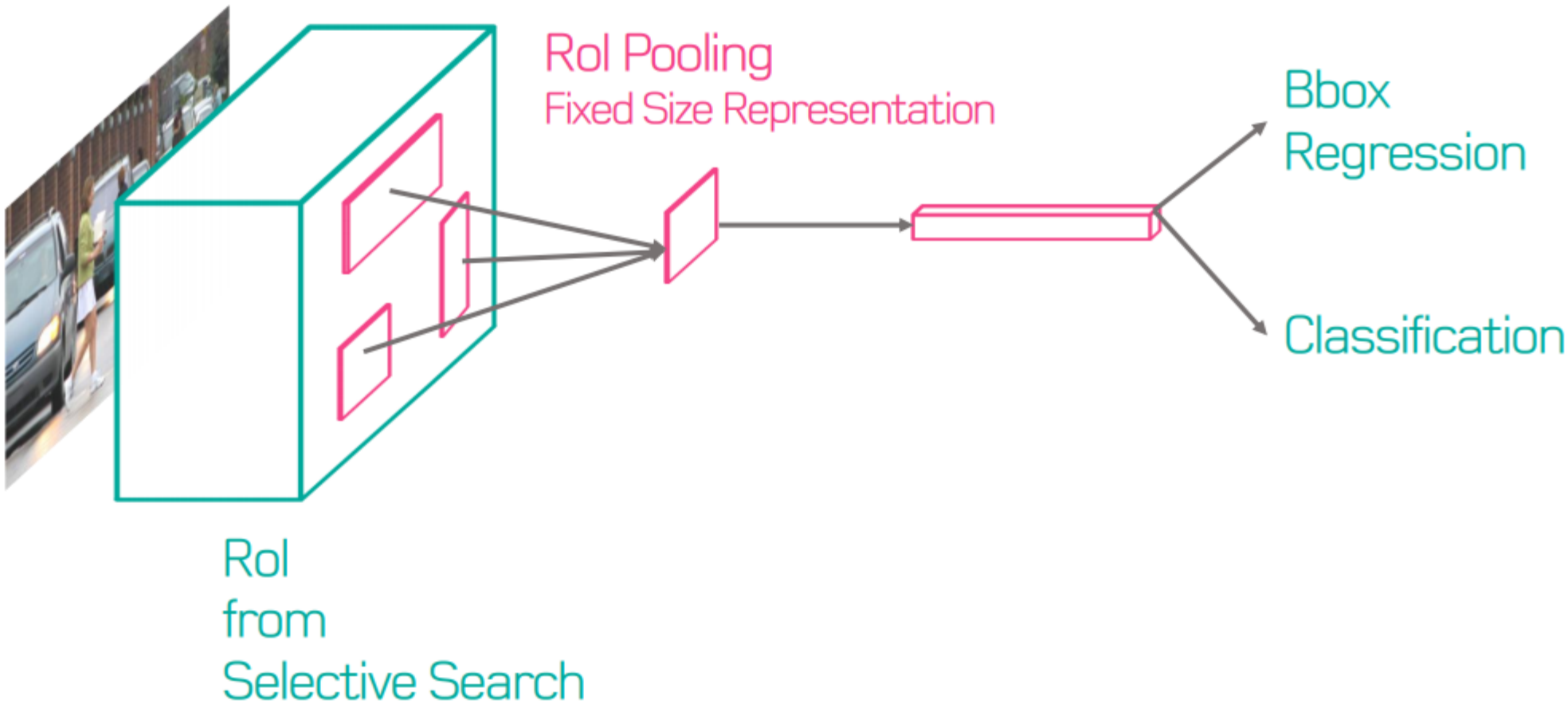
Hyper parameter =10

Only if p*= 1

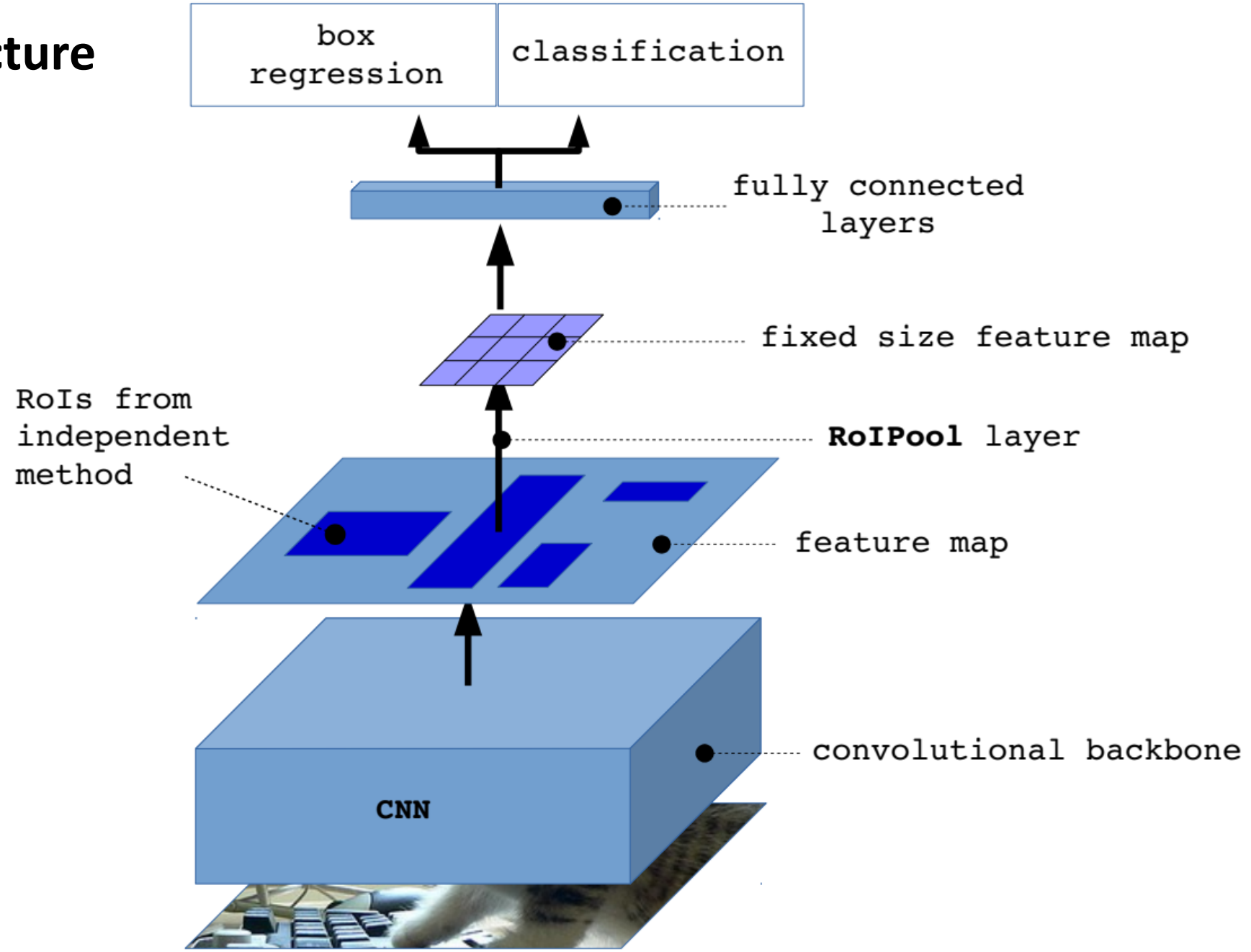
Number of Anchor locations

One net, four losses

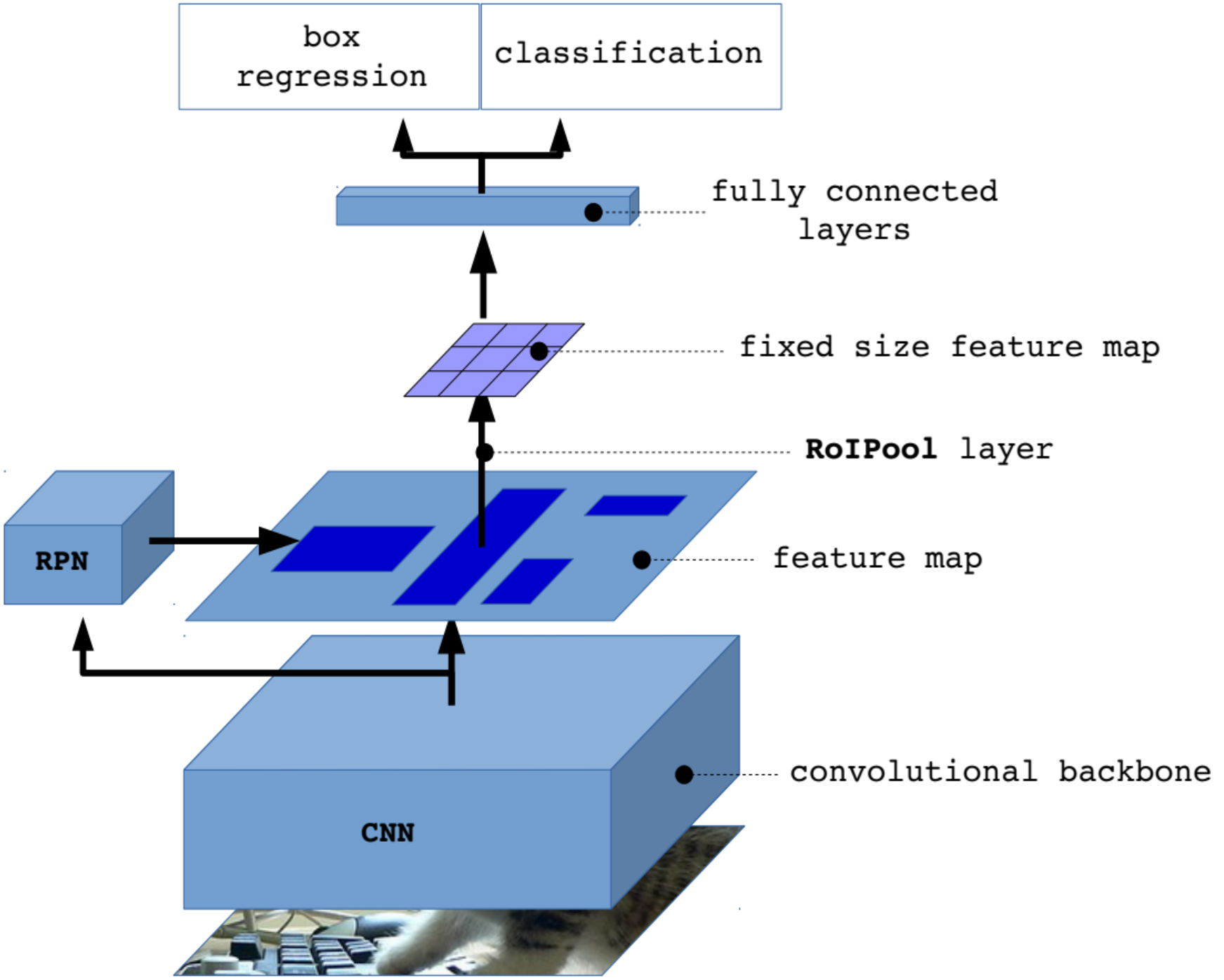
Fast R-CNN


RoI Pooling
Fixed Size Representation
Bbox
Regression
Classification
RoI
from
Selective Search

Fast R-CNN Architecture

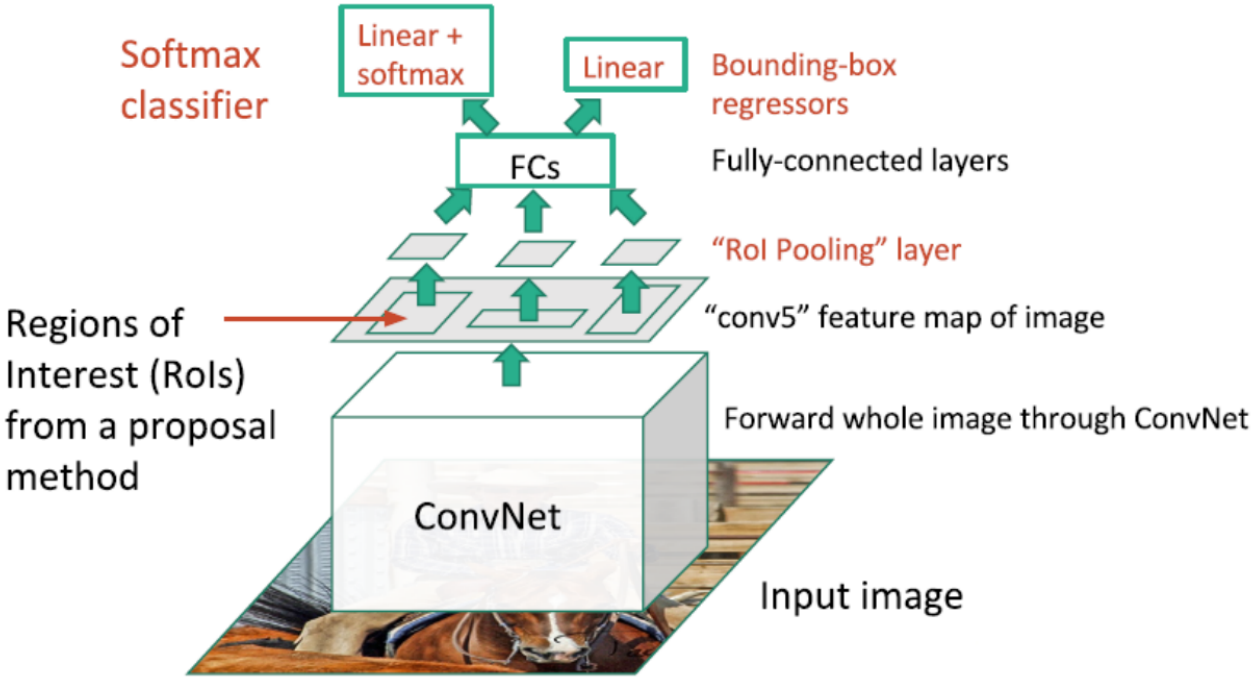


Fast R-CNN Architecture



Fast R-CNN

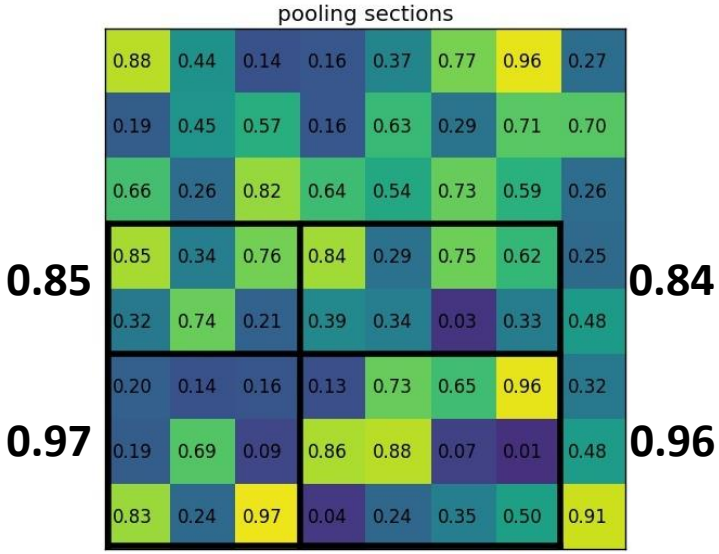
- Fast R-CNN [5]
  - Improvement: It only feed the whole image into CNN only once! Then crop features instead of image itself.



Slide credit to Ke-Shuan Cheng

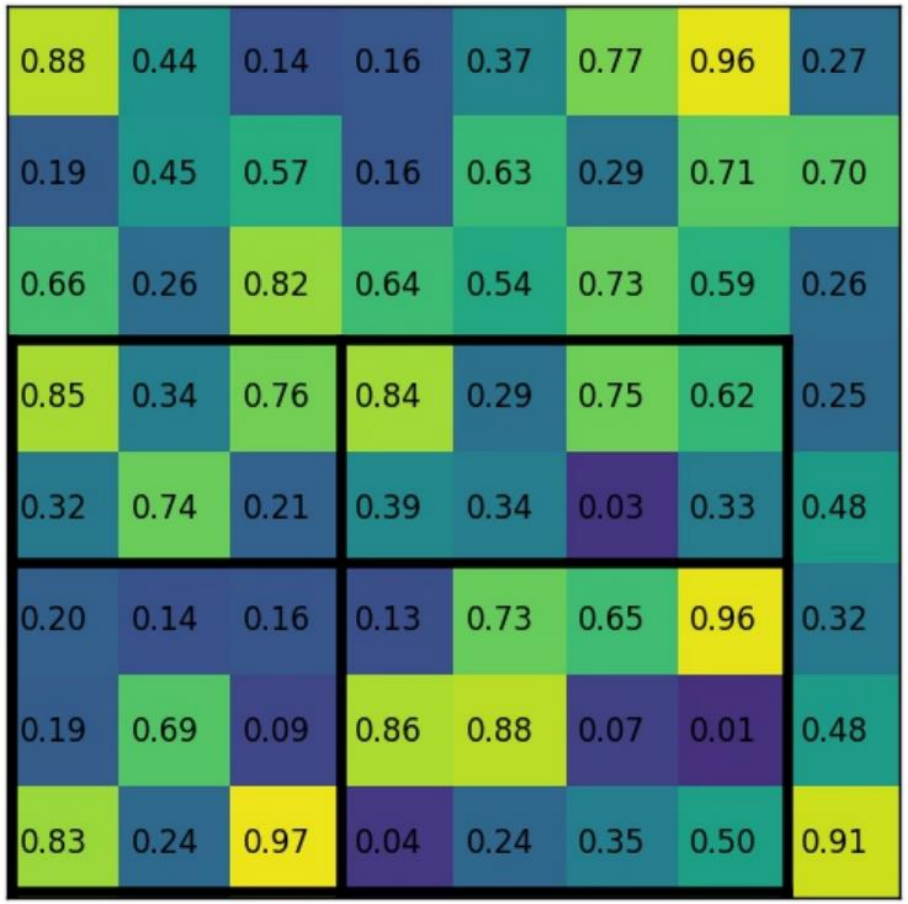
Fast R-CNN

- RoI Pooling
  - The RoI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent of H × W. (e.g., 2*2)



Slide credit to Ke-Shuan Cheng

RoI Pooling in Fast R-CNN



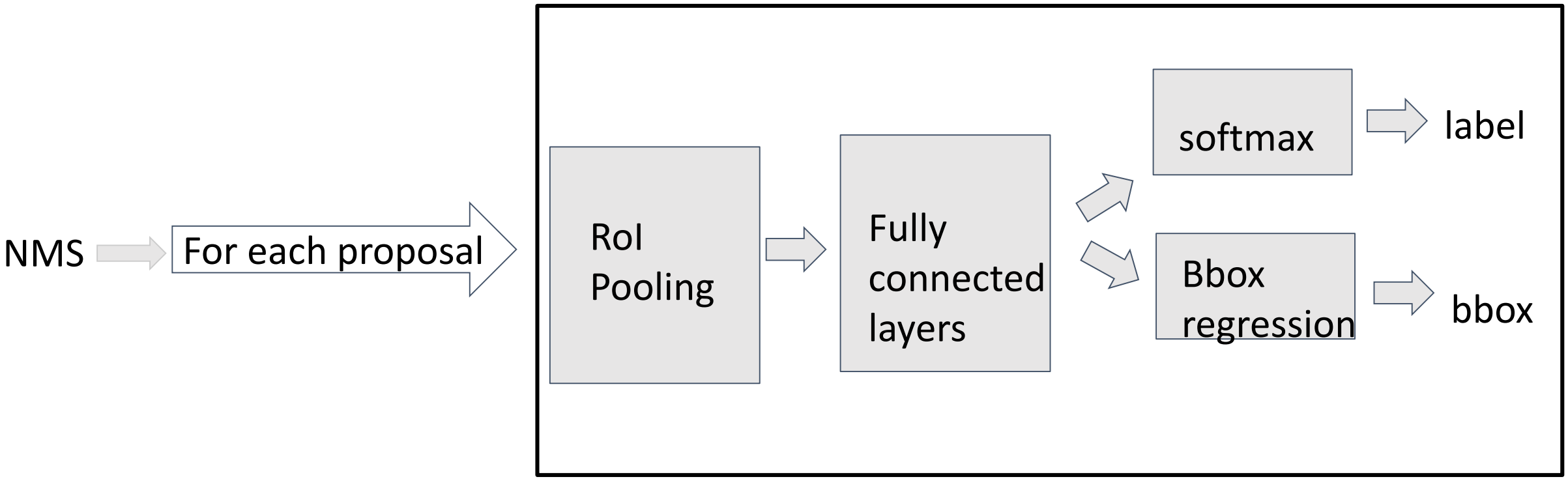
Region projection and pooling sections

RoI Pooling

Max pooling output

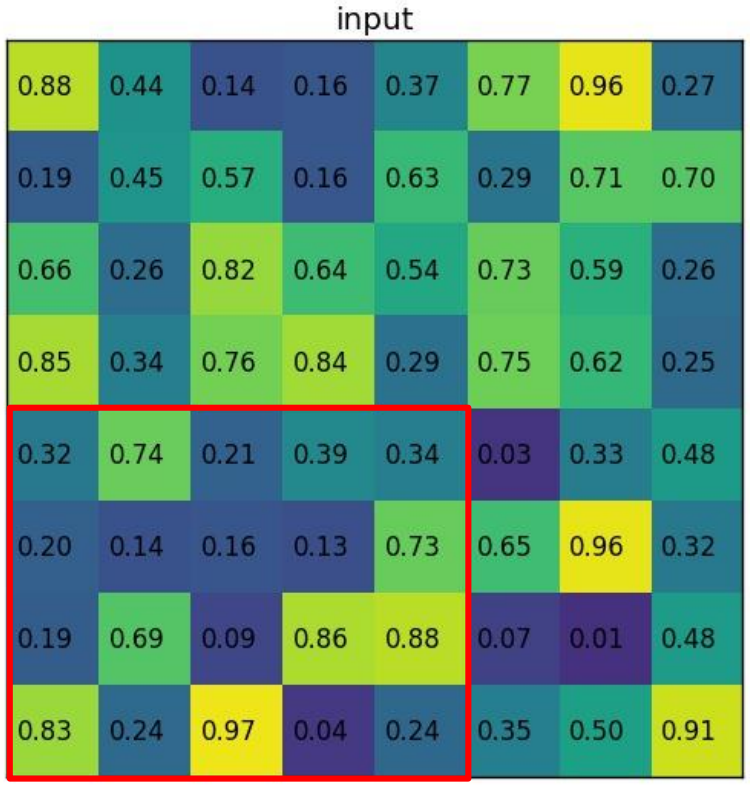
Slide credit to Silvio Galesso

Fast R-CNN


NMS
For each proposal
RoI Pooling
Fully connected layers
softmax
label
Bbox regression
bbox

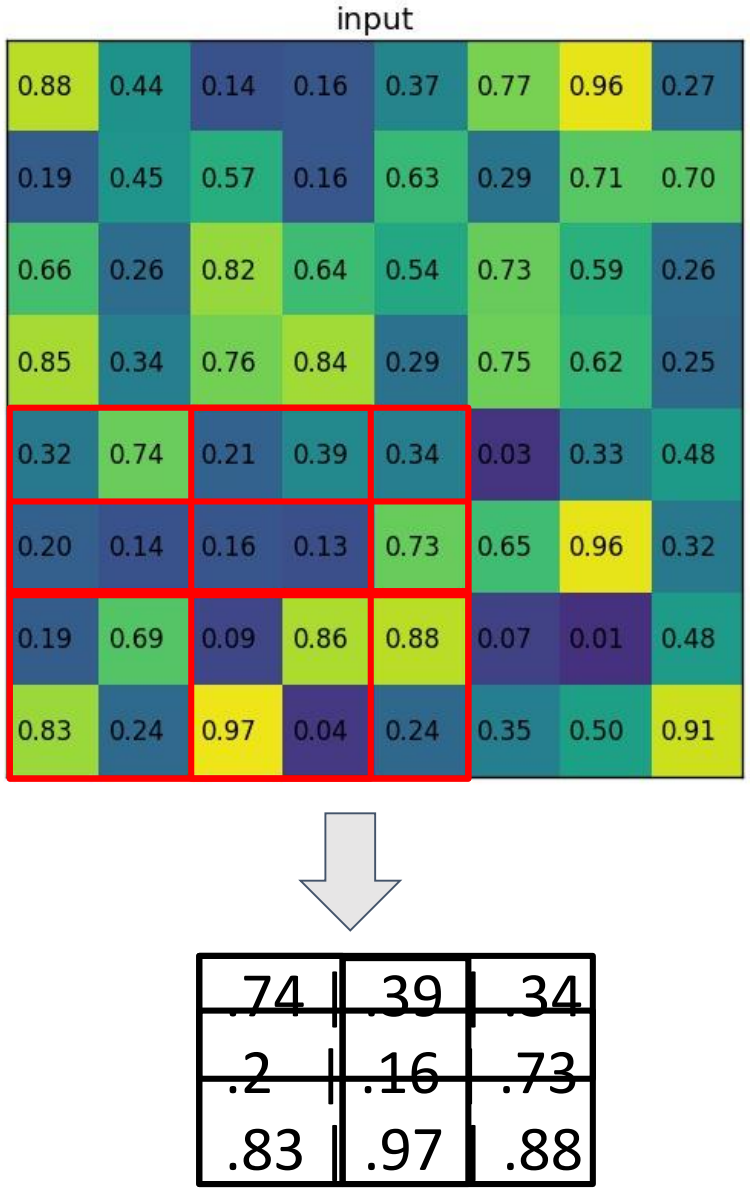
Fast R-CNN

Region of Interest (RoI):



Fast R-CNN

Region of Interest (RoI): 3X3 RoI pooling

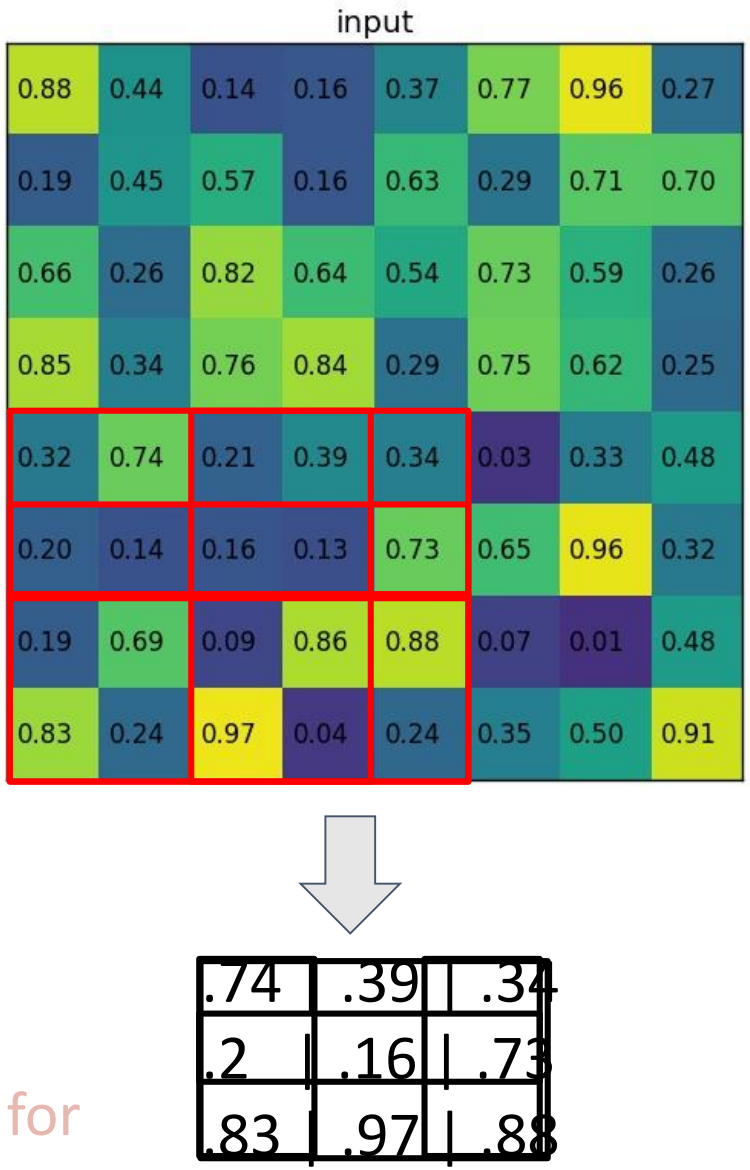


.74	.39	.34
.2	.16	.73
.83	.97	.88

Fast R-CNN

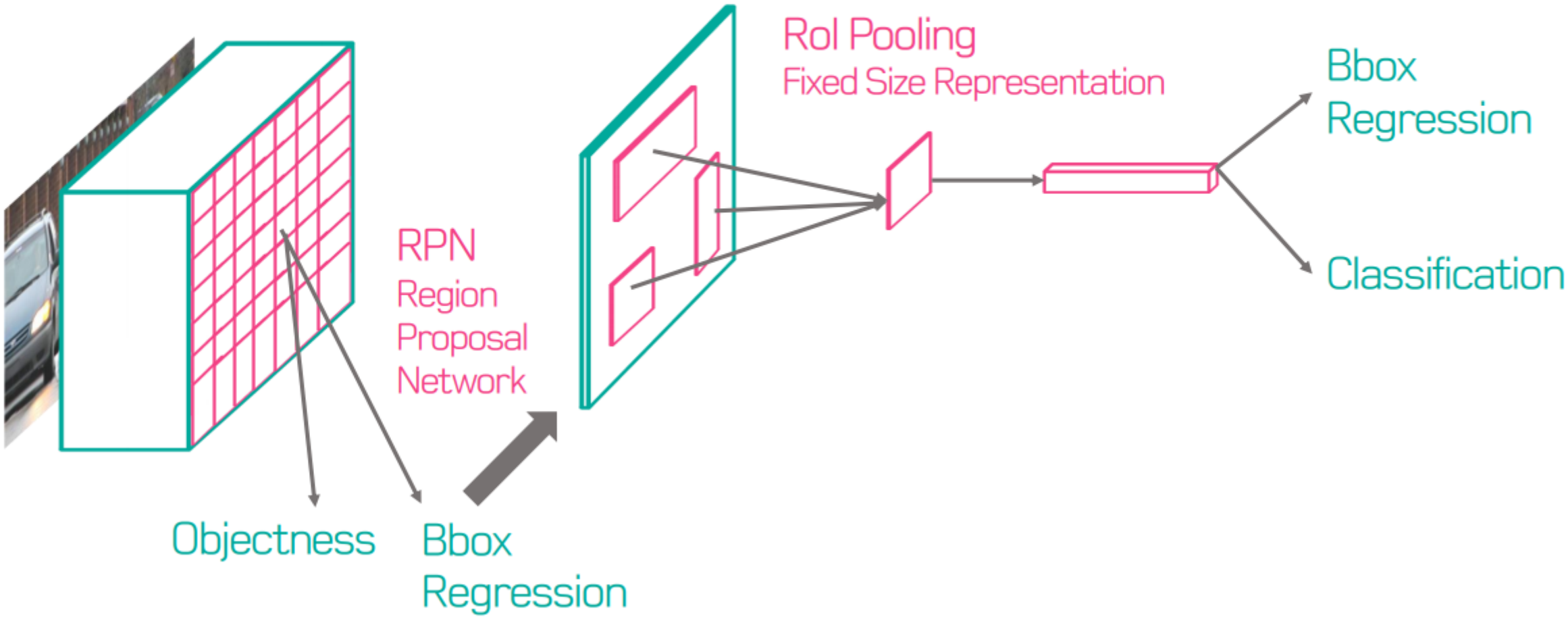
Region of Interest (RoI):

7X7 RoI pooling
Per proposal



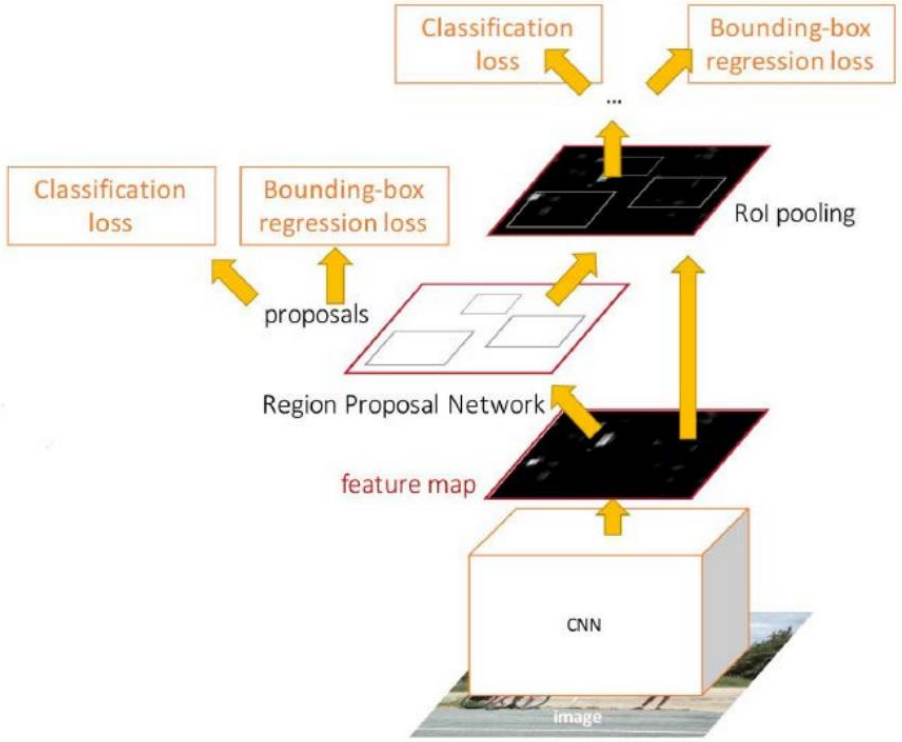
Only a problem for segmentation

Faster R-CNN


RoI Pooling
Fixed Size Representation
Bbox
Regression
Classification
RPN
Region
Proposal
Network
Objectness
Bbox
Regression

Faster R-CNN

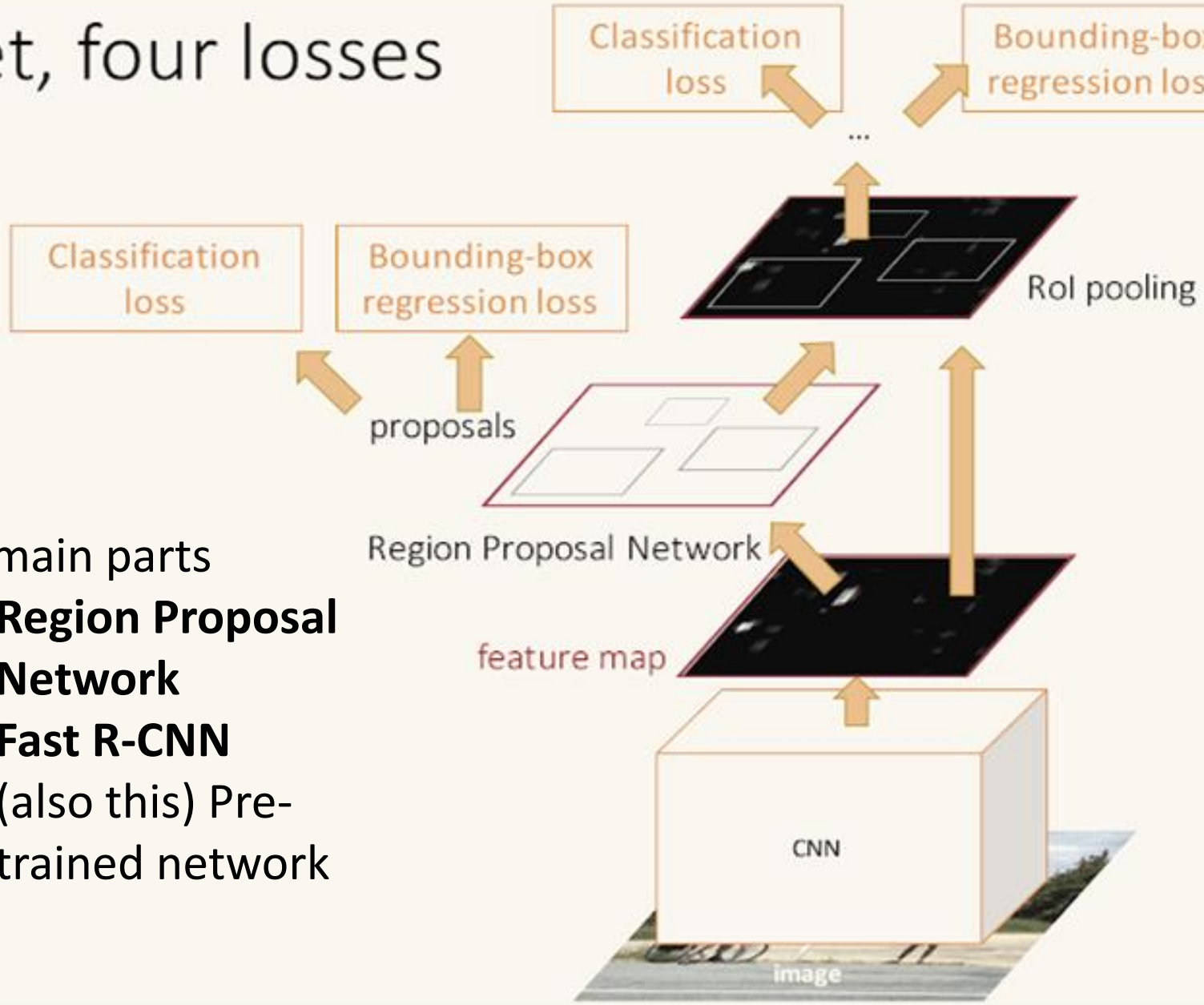
- Faster R-CNN [6]
  - Improvement: Generate RoI by Region Proposal Network.



Slide credit to Ke-Shuan Cheng

One net, four losses

facebook

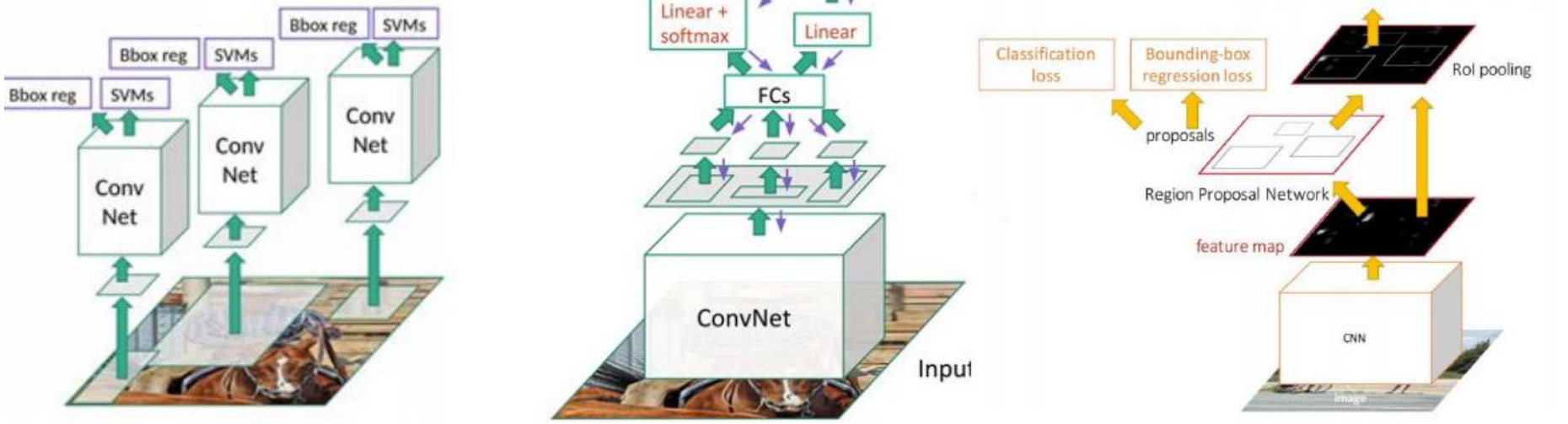


- Two main parts
  - **Region Proposal Network**
  - **Fast R-CNN**
  - (also this) Pre-trained network

Comparison

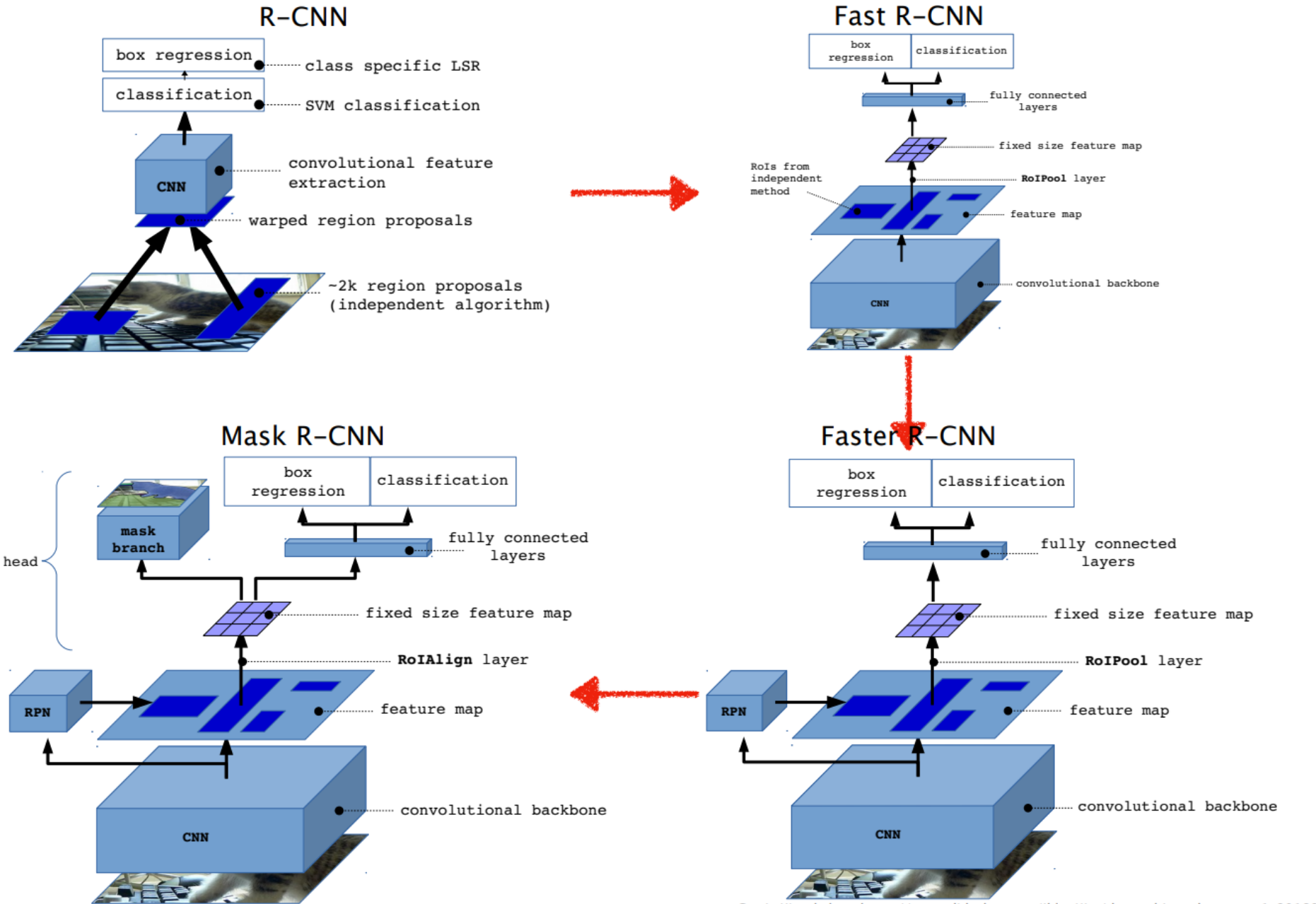
- **Compare with 3 model**

R-CNN, Fast R-CNN, & Fast*er* R-CNN



Slide credit to Ke-Shuan Cheng

Fast R-CNN & Faster R-CNN



Curtis Kim, kakao, https://www.slideshare.net/IldooKim/deep-object-detectors-1-20166

Mask-RCNN

Mask R-CNN

- Mask RCNN is a simple, flexible, and general framework for object **instance segmentation**.
- Mask R-CNN, extends **Faster R-CNN** by **adding a branch for predicting segmentation masks** on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression.
- Mask R-CNN is simple to trained and adds only a **small overhead** to Faster R-CNN.
- The mask branch is a small **Fully Convolutional Network (FCN)** applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner.

Slide credit to Ke-Shuan Cheng

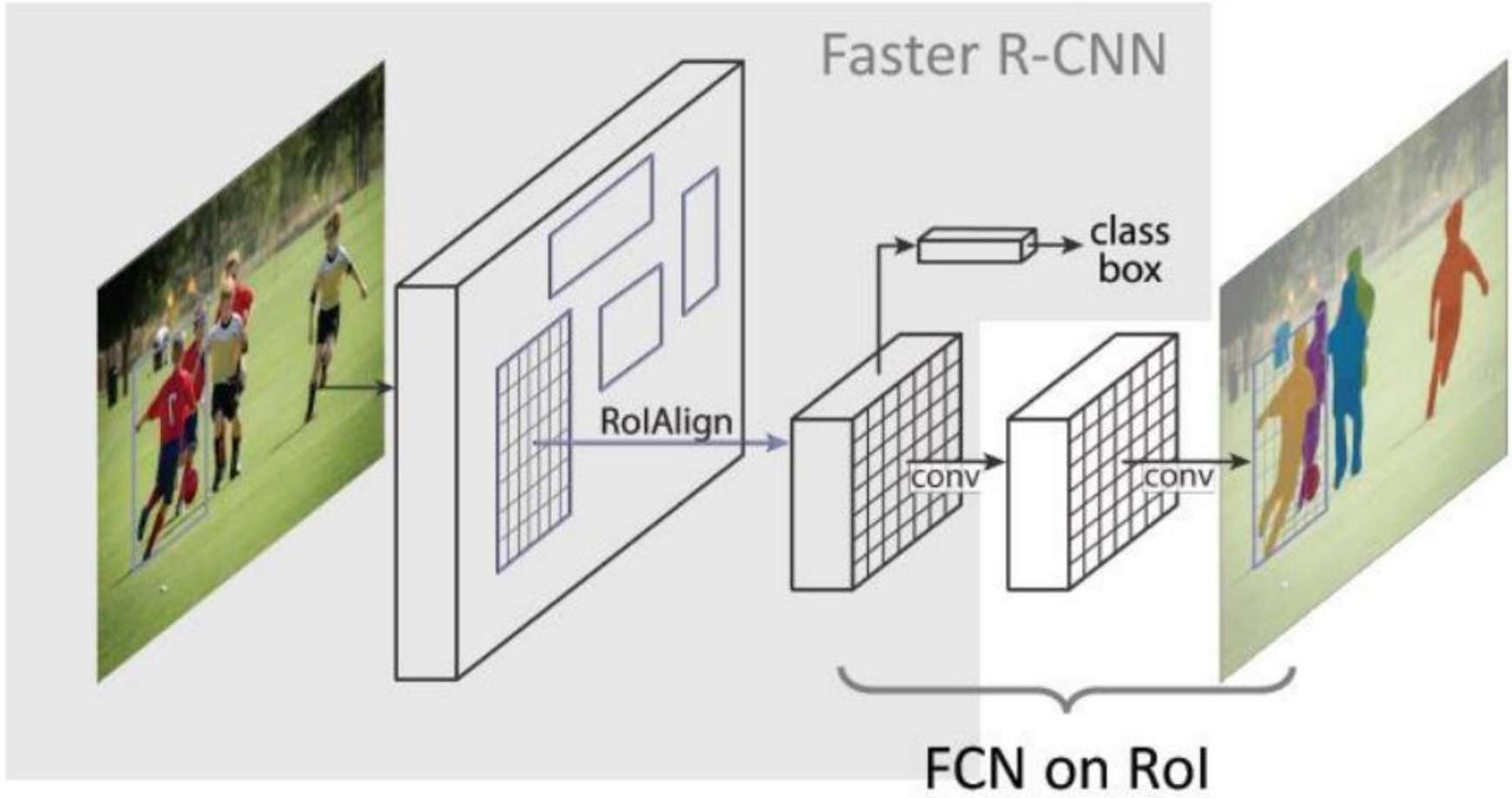
Mask R-CNN: Model Overview

R-CNN-style detection system

1. Backbone architecture
2. Feature Pyramid Network (FPN)
3. Region Proposal Network (RPN)
4. Region of interest feature alignment (RoIAlign)
5. Multi-task network head
   - Box classifier
   - Box regressor
   - Mask predictor
   - Keypoint predictor

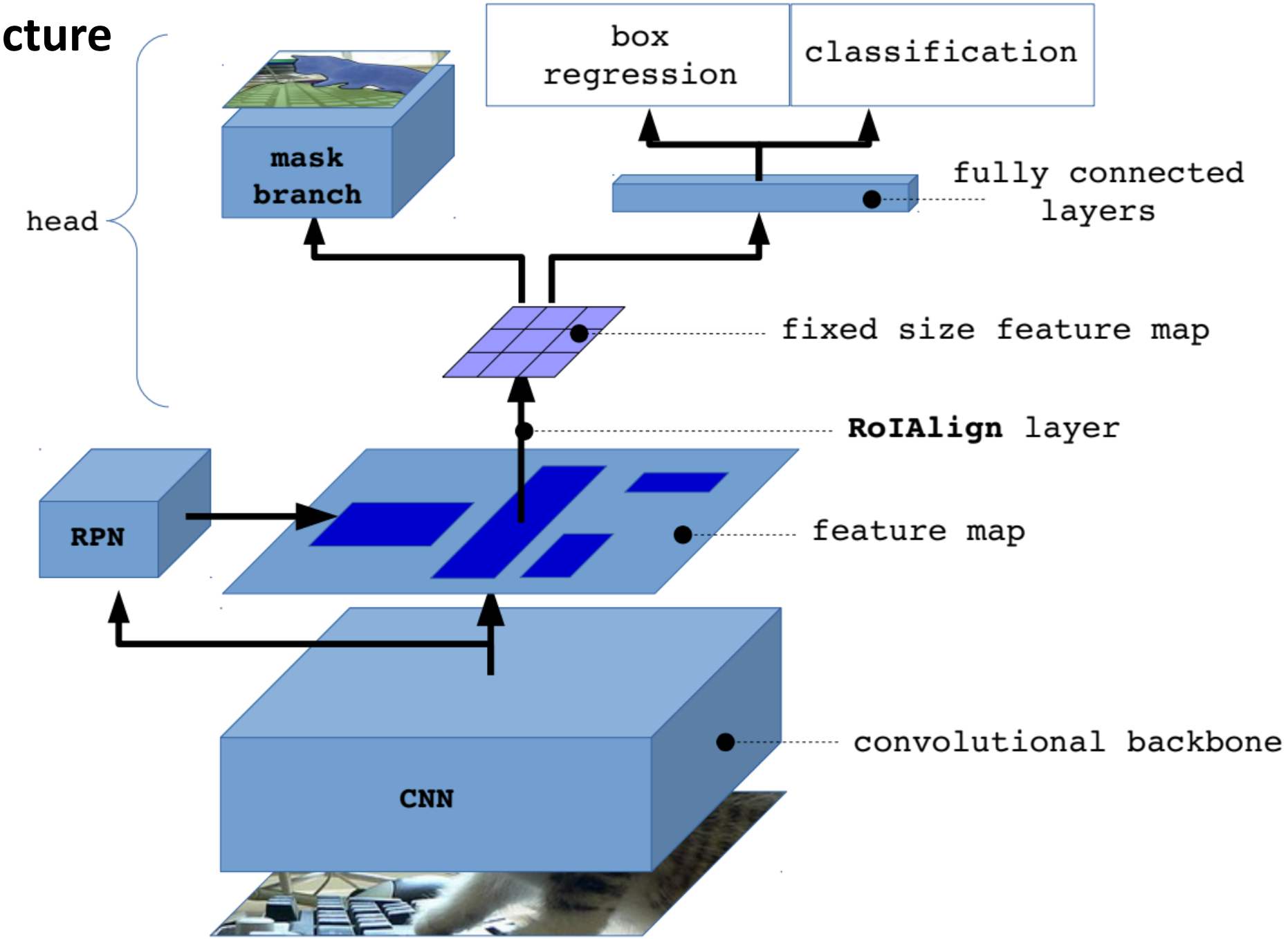
Modular composition of many recent ideas

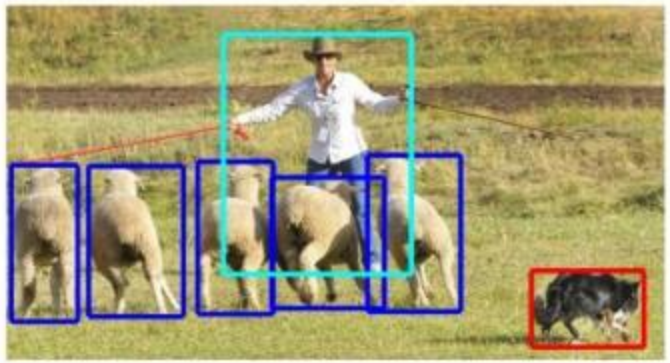
- Mask R-CNN = **Faster R-CNN** with **FCN** on RoIs



Slide from *Mask R-CNN Tutorial*, K. He. ICCV 2017

Mask R-CNN Architecture





BBox
Classification

+

Segmentation
Classification

=

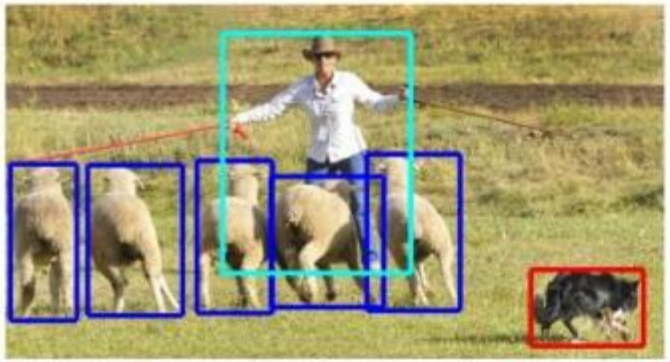
Segmentation
in BBox
Classification

Can Separate
Cannot Segment

Cannot Separate
Can Segment

Faster R-CNN

FCN



BBox
Classification

+

Segmentation
Classification

=

Segmentation
in BBox
Classification

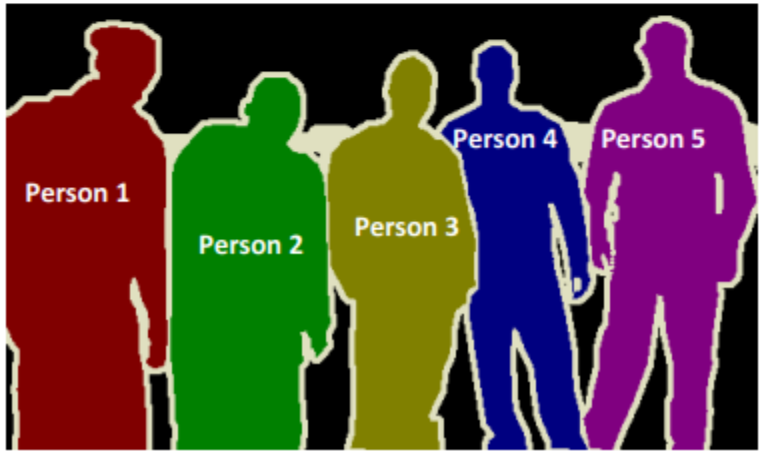
Can Separate
Cannot Segment

Cannot Separate
Can Segment

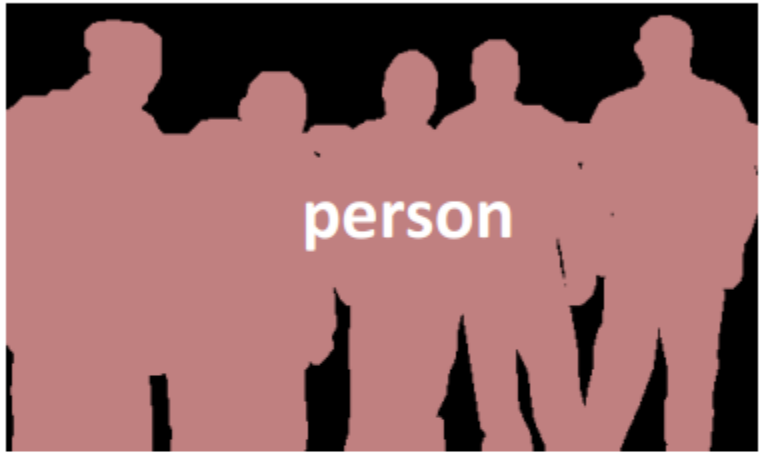
Faster R-CNN + FCN = FCN
on BBOX !

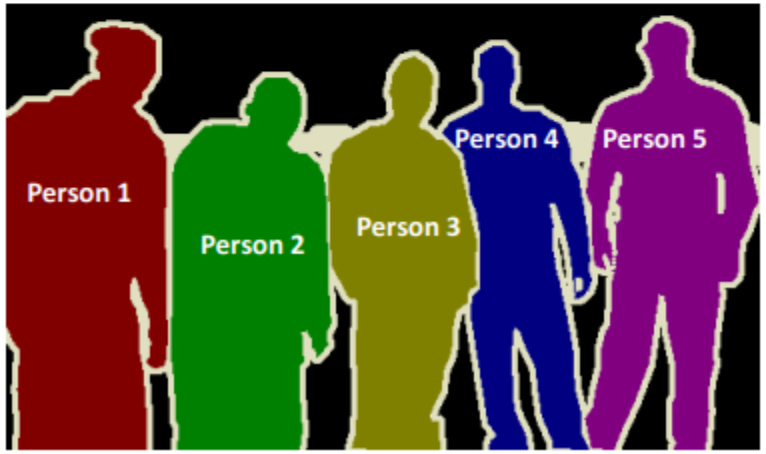
Instance Segmentation Methods

R-CNN driven



FCN driven

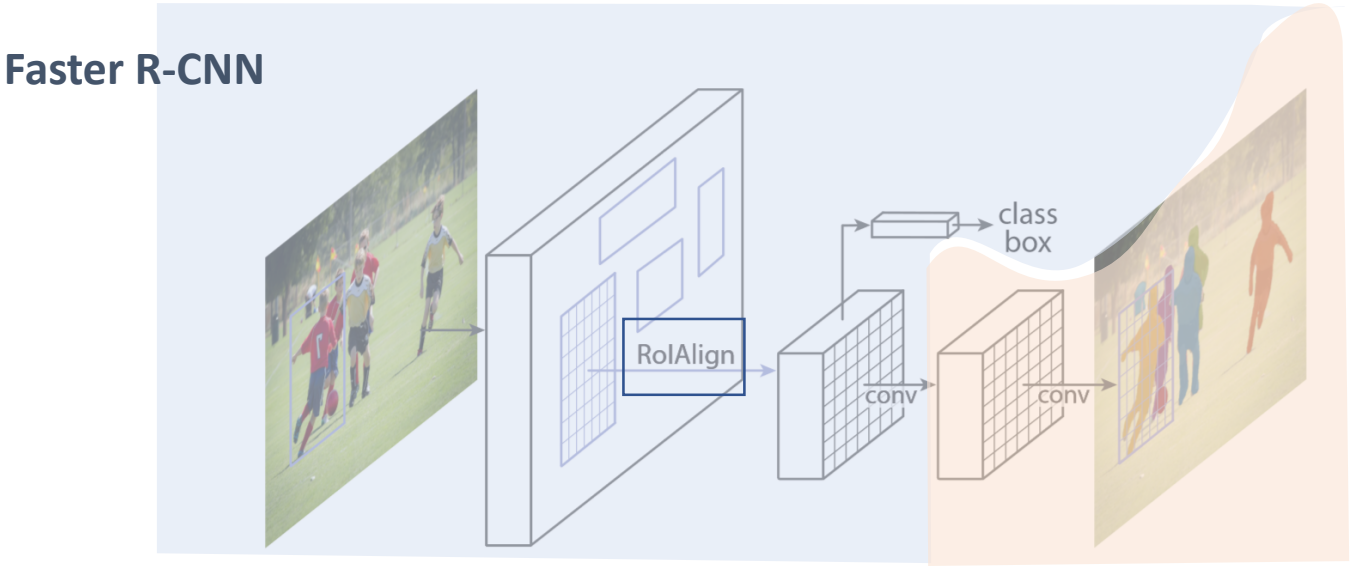




Kaiming He, ICCV 2017 Tutorial

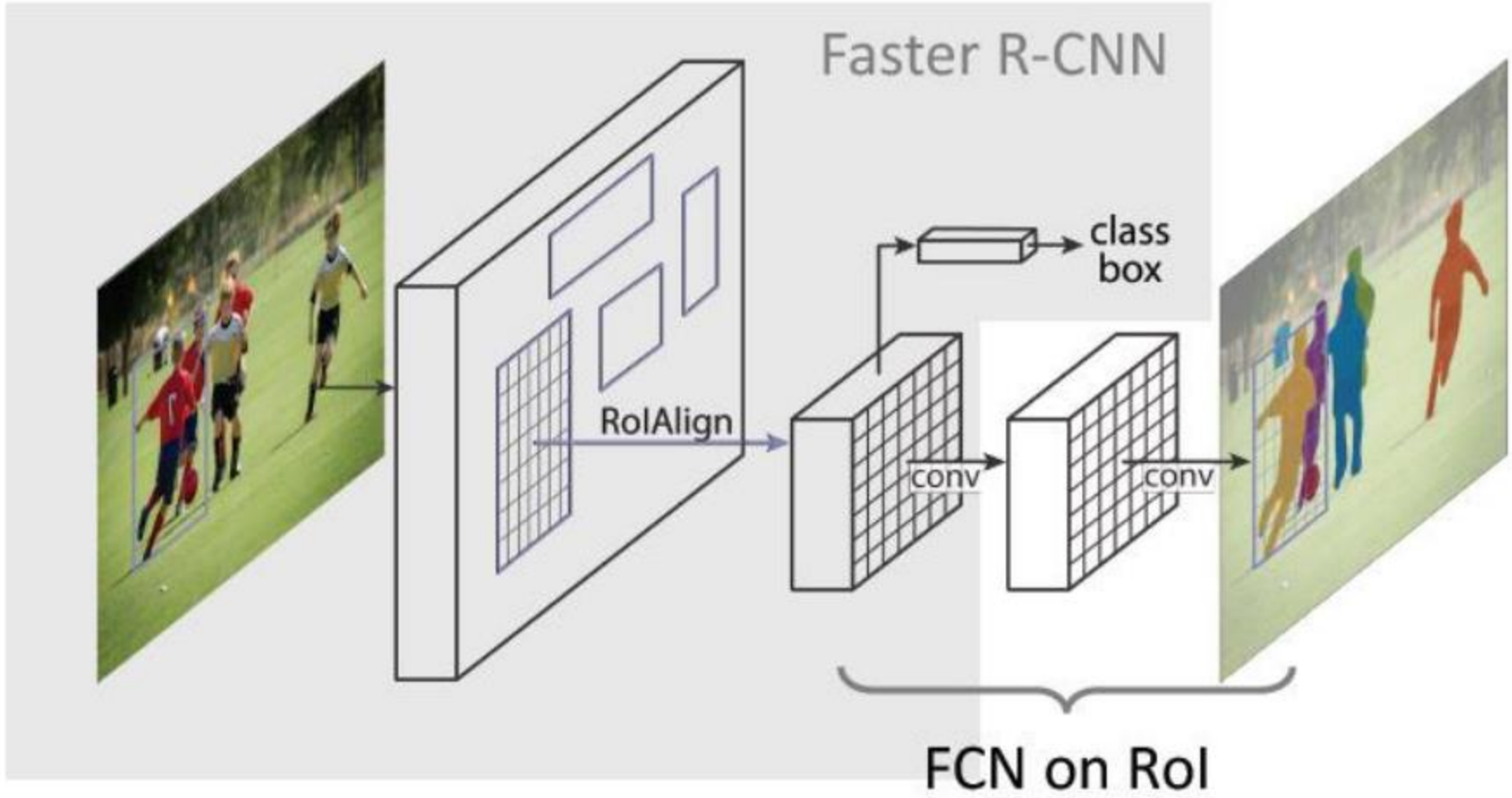
Mask R-CNN

- **Mask R-CNN [3]** is conceptually simple: Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; to this R-CNN added a **third branch that outputs the object mask.**



Slide credit to Ke-Shuan Cheng

- Mask R-CNN = **Faster R-CNN** with **FCN** on RoIs



Slide from *Mask R-CNN Tutorial*, K. He. ICCV 2017

Network architecture

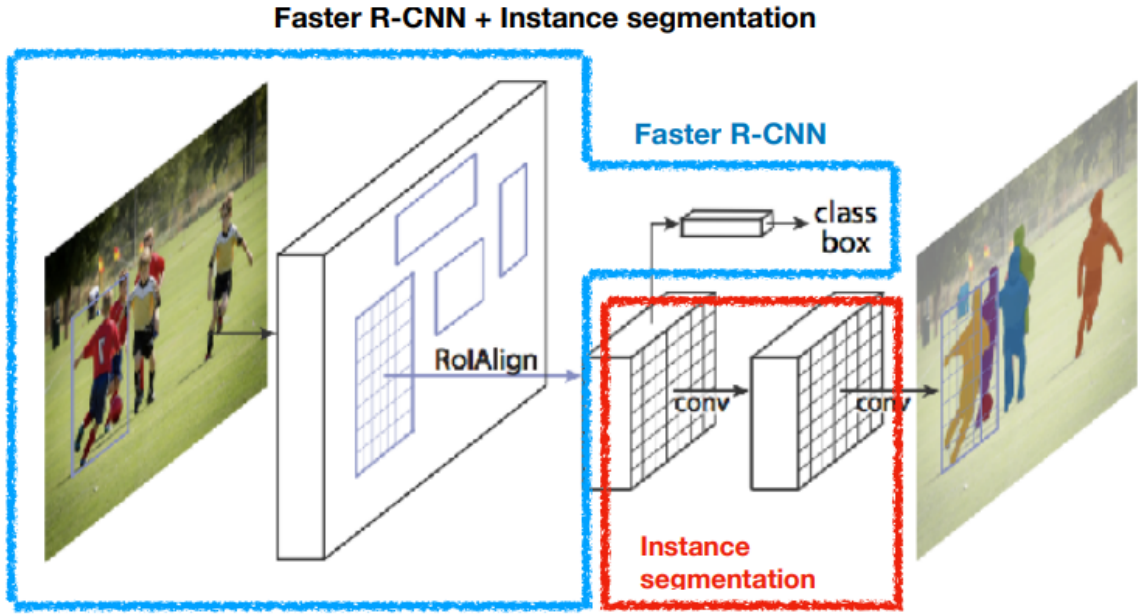
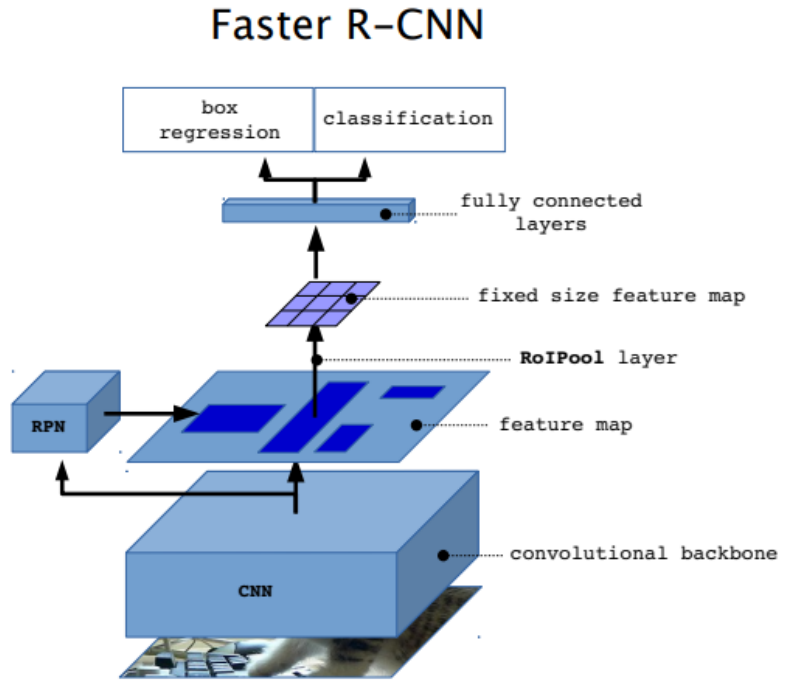
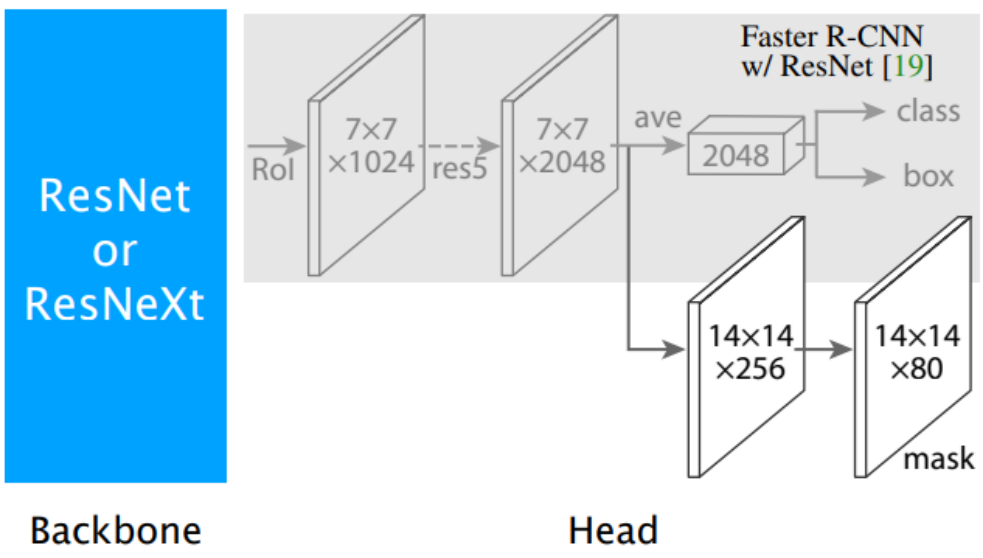
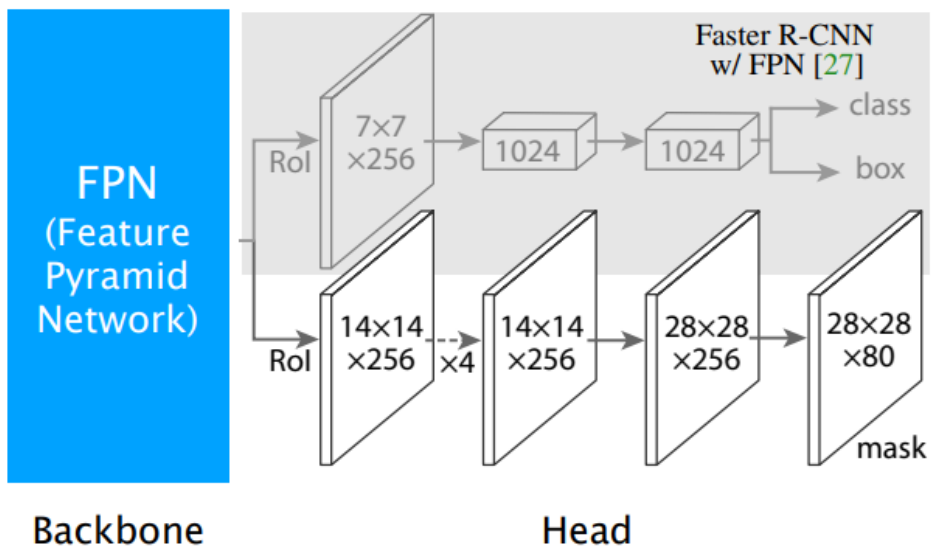


Figure 1. The **Mask R-CNN** framework for instance segmentation.

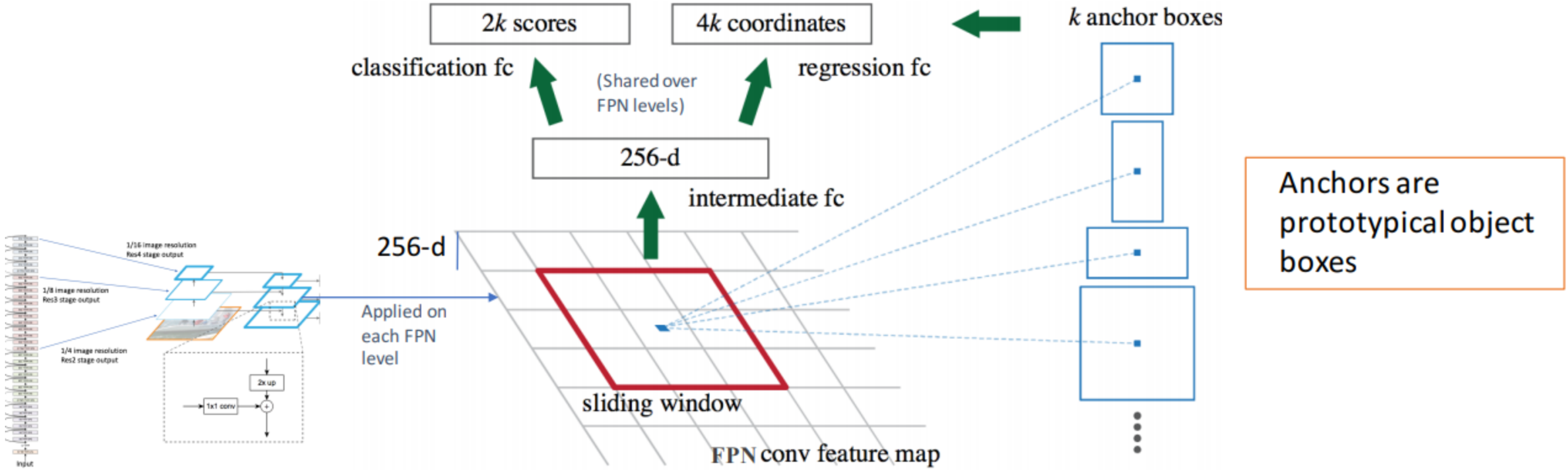






Region Proposal Network (RPN)

Proposals = sliding window object/not-object classifier + box regression
inside the same network

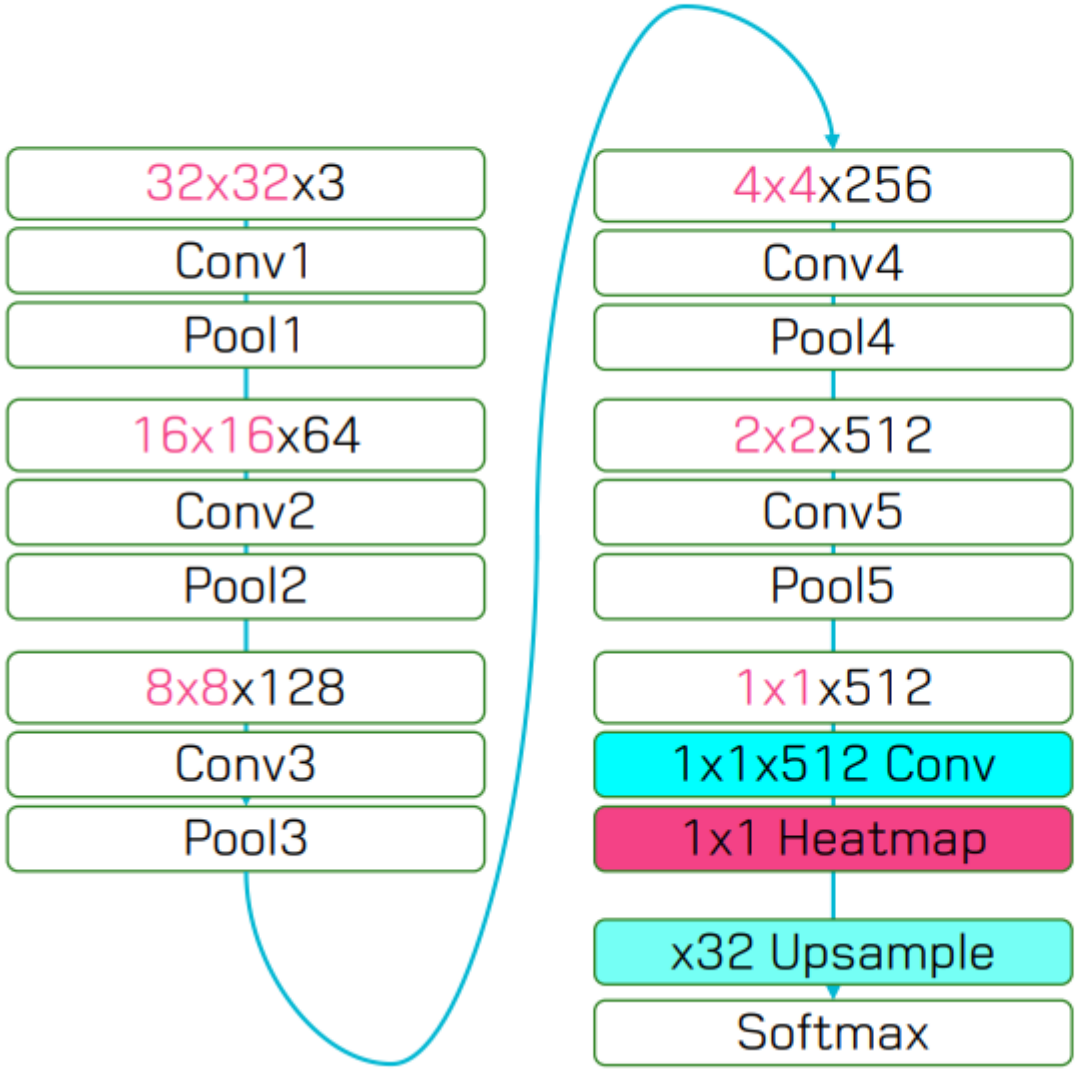


Anchors are prototypical object boxes

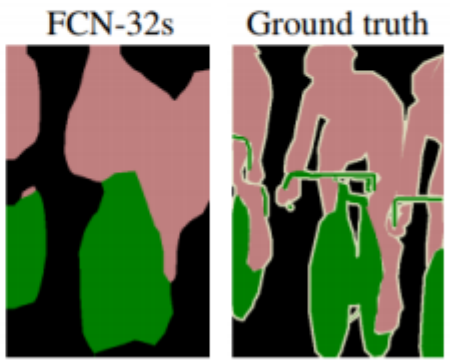
Ren et al. Faster R-CNN: Region Proposal Networks for Real-Time Object Detection. In NIPS 2015

Slide credit to Ross Girshick

Fully Convolutional Networks



Remove Pooling
1x1 Conv for Heatmap Output



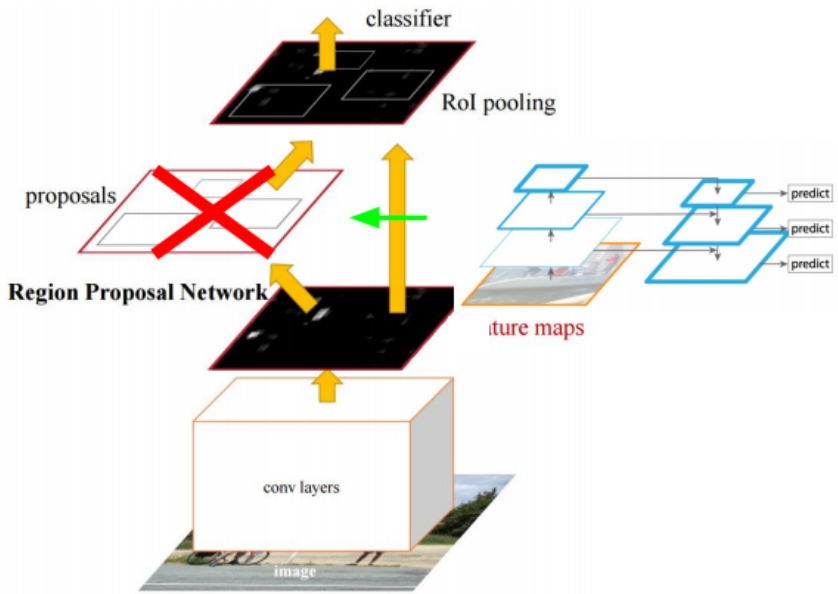
Network architecture

- Backbone – ResNet

ResNet-50-C4 ResNet-101-C4

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

- Backbone – FPN



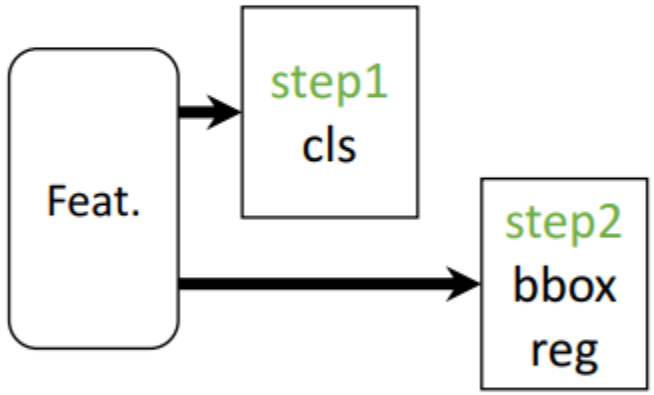
– FPN exploits the inherent hierarchy of CNNs to compute multi-scale features:

– Replace single scale feature map with FPN.

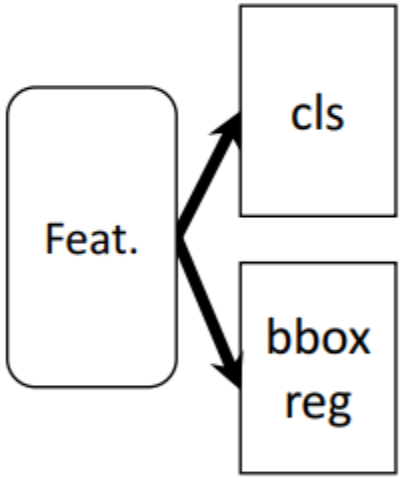
Source: Lin et al., Feature Pyramid Networks for Object Detection

Parallel Heads

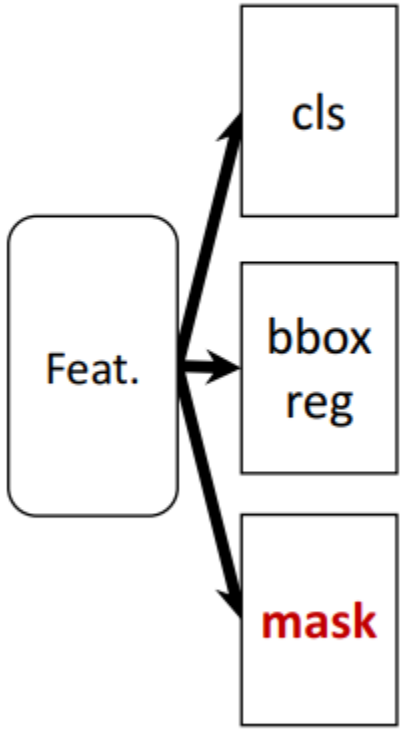
- Easy, fast to implement and train



(slow) R-CNN

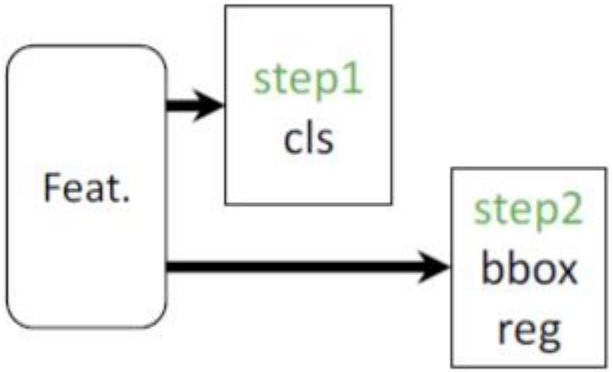


Fast/er R-CNN

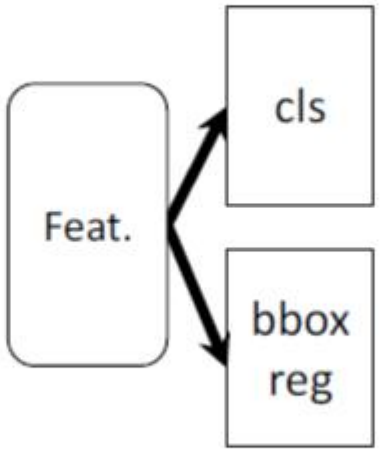


Mask R-CNN

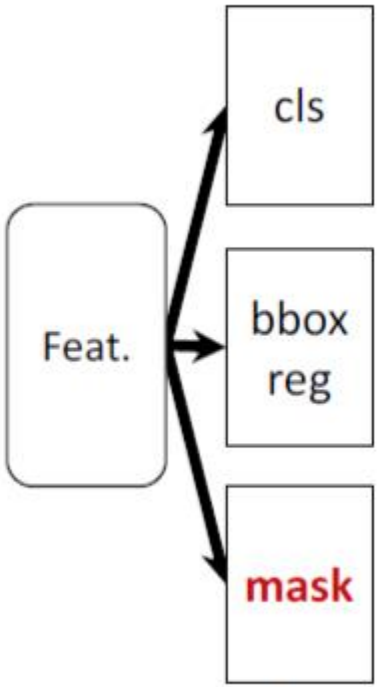
Mask Head on Faster R-CNN



(slow) R-CNN



Fast/er R-CNN

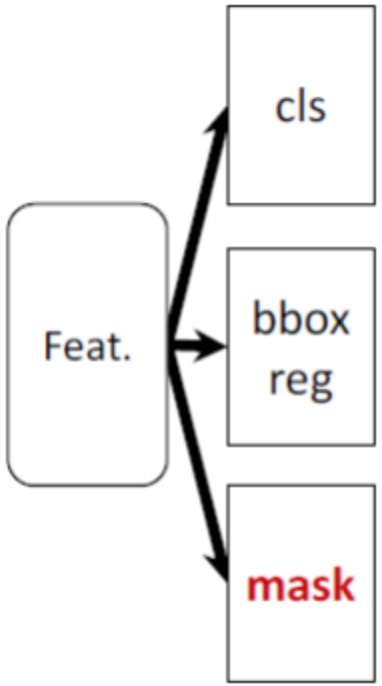


Mask R-CNN

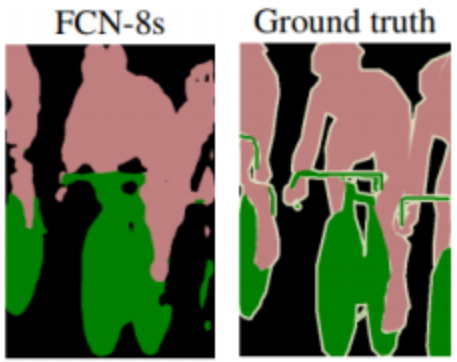
- Fully connected branch:
  - $K + 1$ softmax for classification
  - $4 \cdot K$ box regression targets: $\mathbf{t}^k = (t_x^k, t_y^k, t_w^k, t_h^k)$

Slide credit to T. Kim

Mask Head on Faster R-CNN



Mask R-CNN

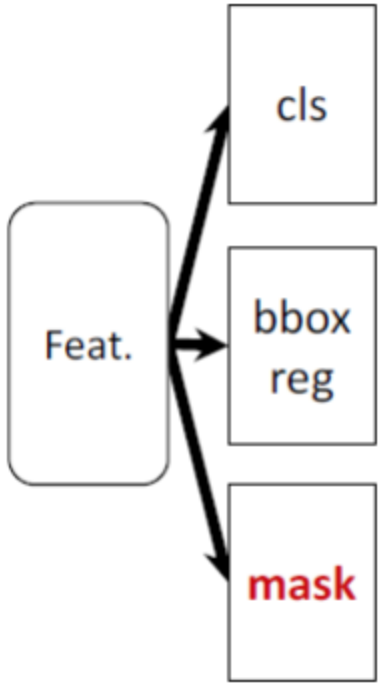


FCN

- Pixel-level Classification
- Per Pixel Softmax (Multinomial)
- Multi Instance

Slide credit to T. Kim

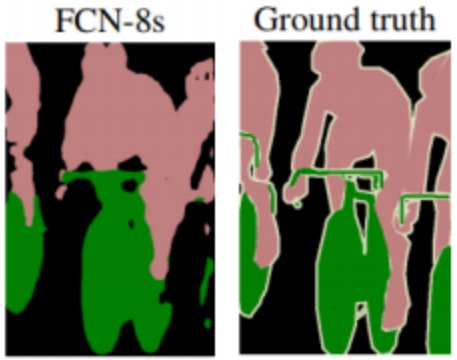
Mask Head on Faster R-CNN



Mask R-CNN

Faster R-CNN

- Classification
- Instance Level RoI

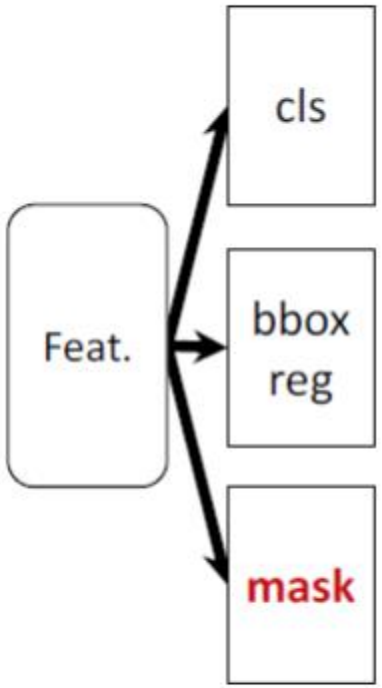


FCN

- Pixel-level Classification
- Per Pixel Softmax (Multinomial)
- Multi Instance

Slide credit to T. Kim

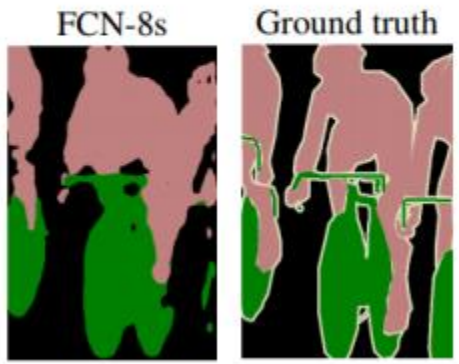
Mask Head on Faster R-CNN



Mask R-CNN

Faster R-CNN

- Classification
- Instance Level RoI

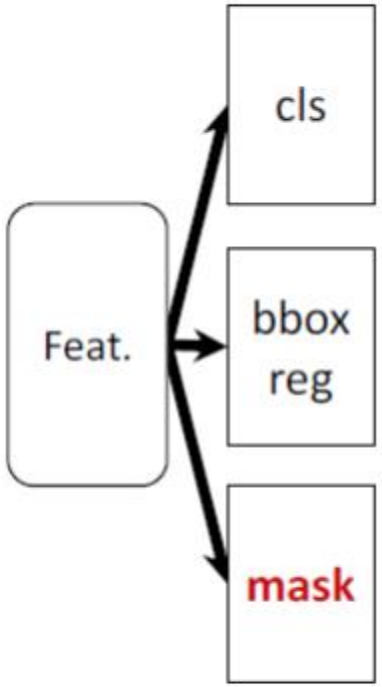


FCN

- ~~Pixel-level Classification~~
- Per Pixel Softmax (Multinomial)
- Multi Instance

Slide credit to T. Kim

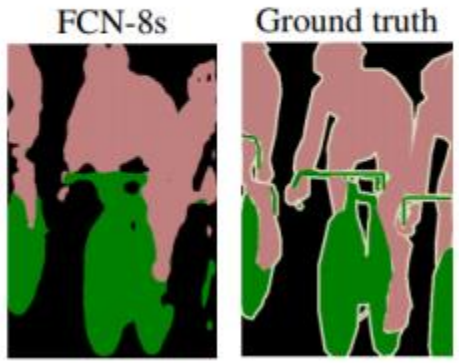
Mask Head on Faster R-CNN



Mask R-CNN

Faster R-CNN

- Classification
- Instance Level RoI

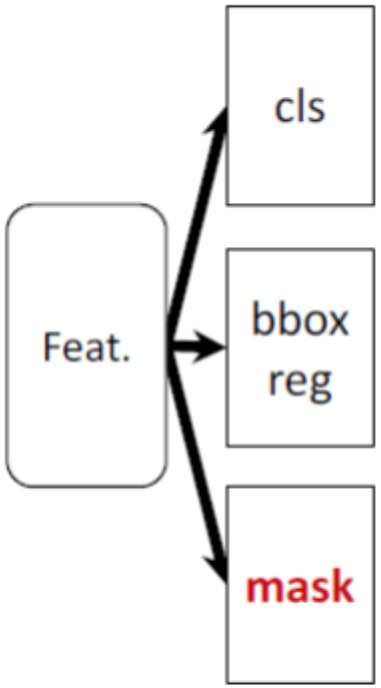


FCN

- ~~Pixel-level Classification~~
- Per Pixel ~~Softmax~~ → Sigmoid (Binary)
- Multi Instance

Slide credit to T. Kim

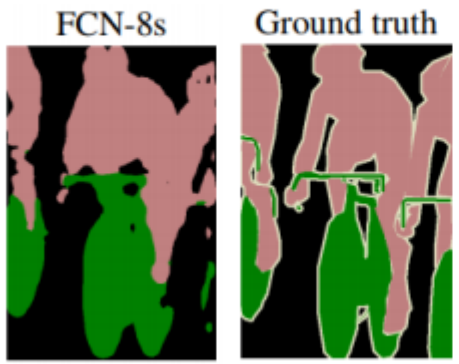
Mask Head on Faster R-CNN



Mask R-CNN

Faster R-CNN

- Classification
- Instance Level RoI

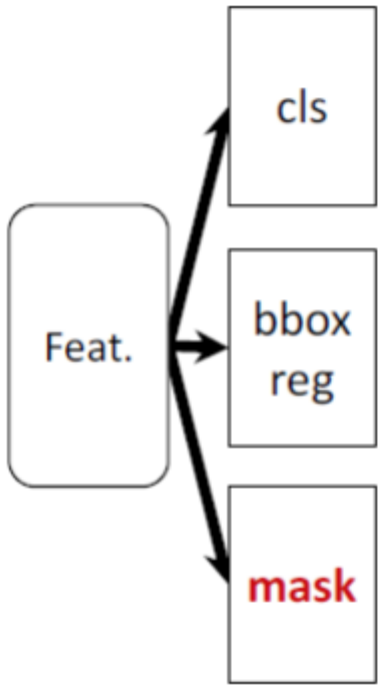


FCN

- ~~Pixel-level Classification~~
- Per Pixel ~~Softmax~~ → Sigmoid (Binary)
- ~~Multi-Instance~~

Slide credit to T. Kim

Mask Head on Faster R-CNN



Mask R-CNN

DB

BBox + Class + Mask

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

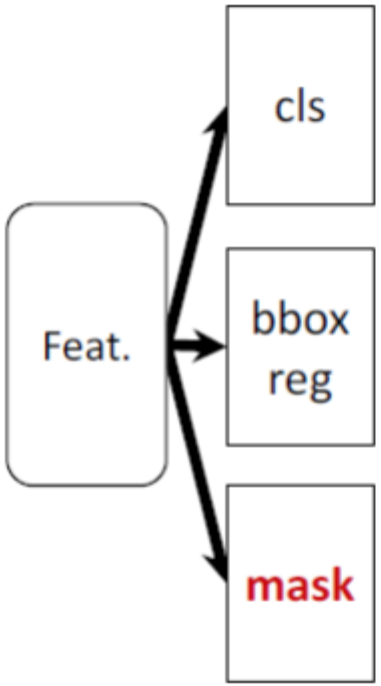
L_{cls}: Softmax Cross Entropy

L_{box}: Regression

L_{mask}: Binary Cross Entropy

Slide credit to T. Kim

Mask Head on Faster R-CNN



Mask R-CNN

Training Phase

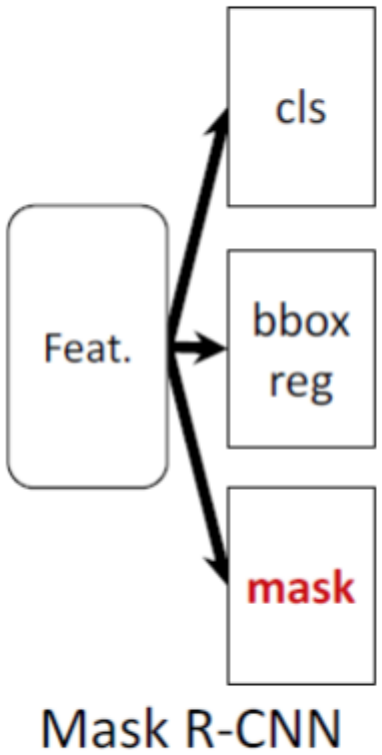
$$L_{mask} = L_{c1} + L_{c2} + \cdots + L_{ck}$$

if) GT Class is 3

$$L_{mask} = L_{c3}$$

Slide credit to T. Kim

Mask Head on Faster R-CNN



Mask R-CNN

Training Phase

$$L_{mask} = L_{c1} + L_{c2} + \cdots + L_{ck}$$

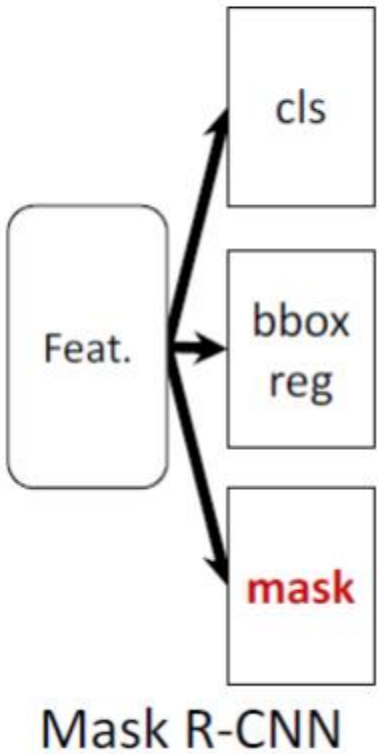
if) GT Class is 3

$$L_{mask} = L_{c3}$$

Mask Branch Only Learns How to Mask *independent of Class*

Slide credit to T. Kim

Mask Head on Faster R-CNN



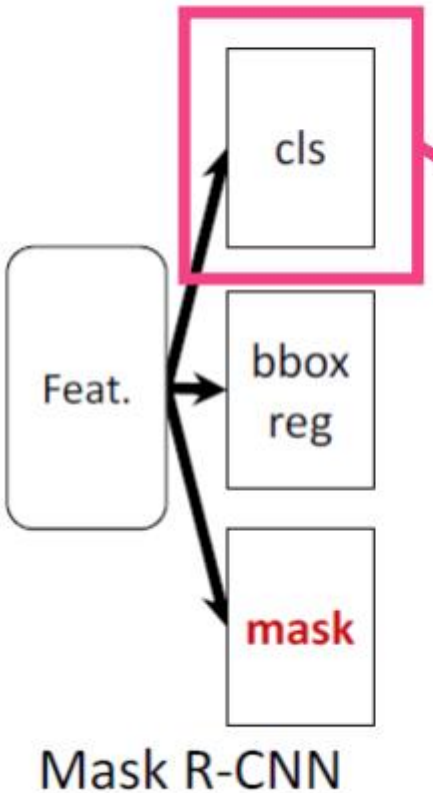
Mask R-CNN

Test Phase

Predicts Human Mask
Predicts Car Mask
Predicts Horse Mask
Predicts ...

Slide credit to T. Kim

Mask Head on Faster R-CNN



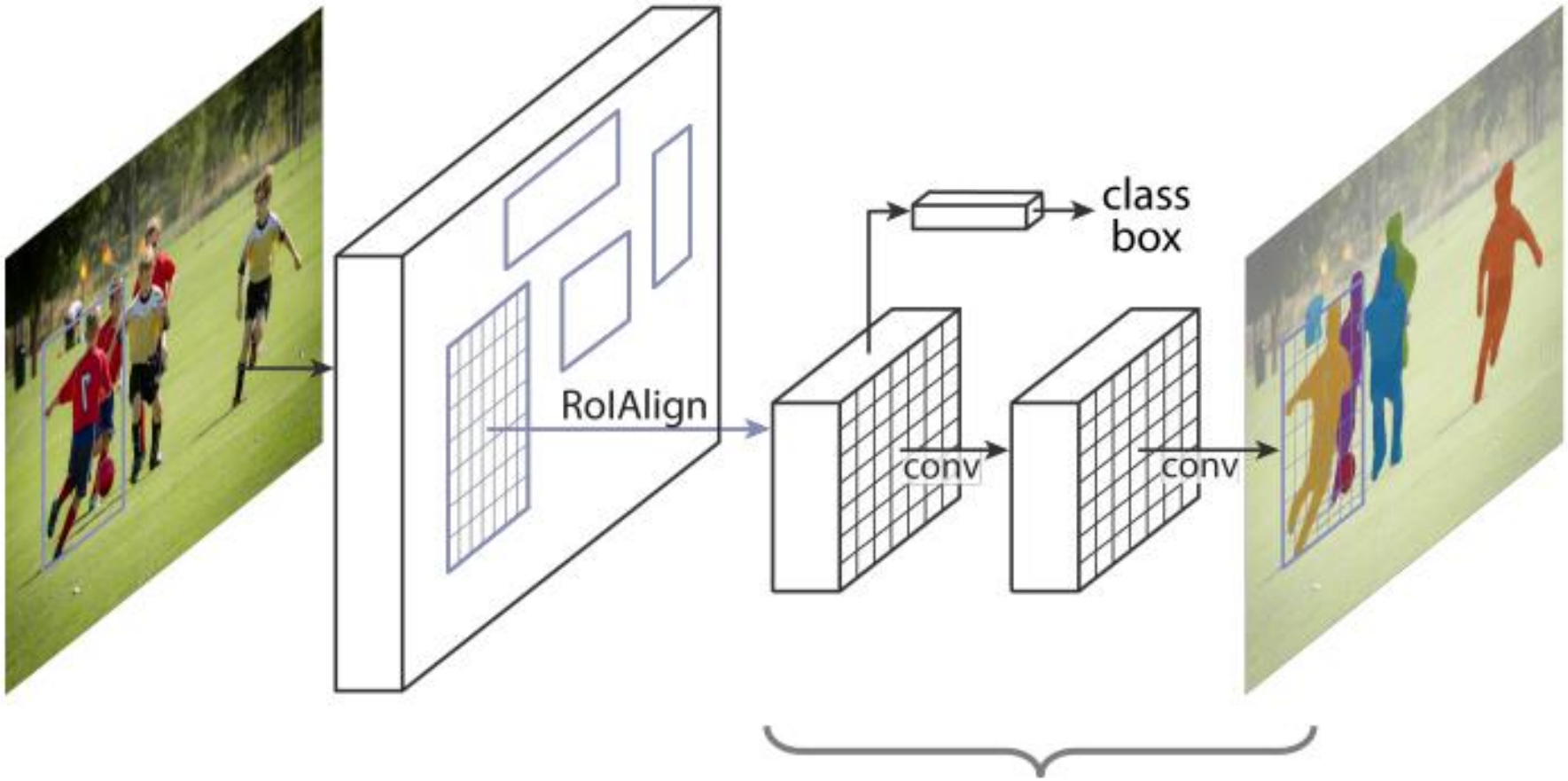
Mask R-CNN

Test Phase

Predicts Human Mask
Predicts Car Mask
Predicts Horse Mask
Predicts ...

Winner Takes All

Slide credit to T. Kim

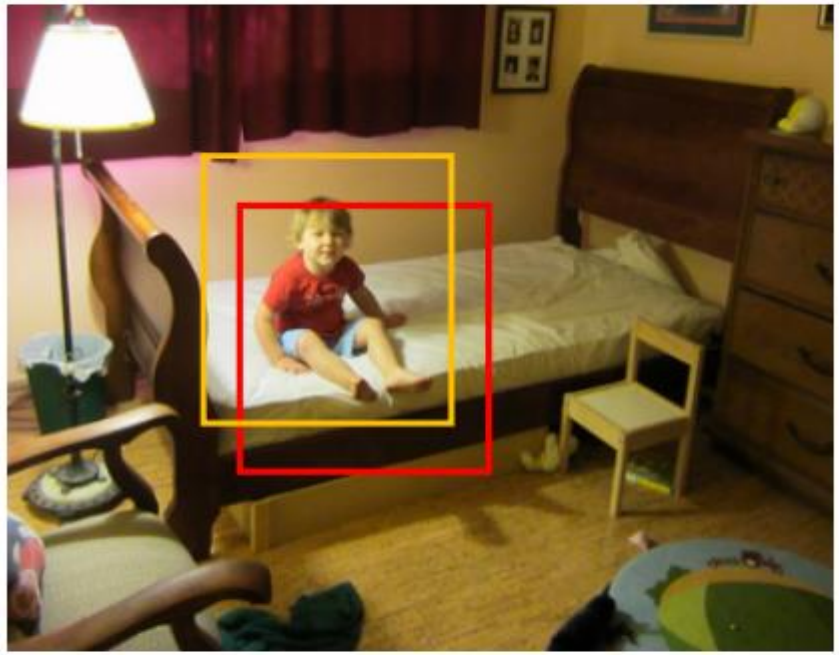


2. Fully-Conv on RoI:

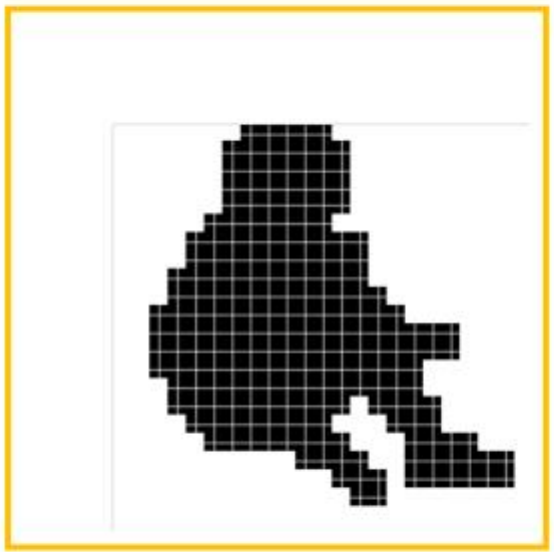
equivariant to translation within RoI

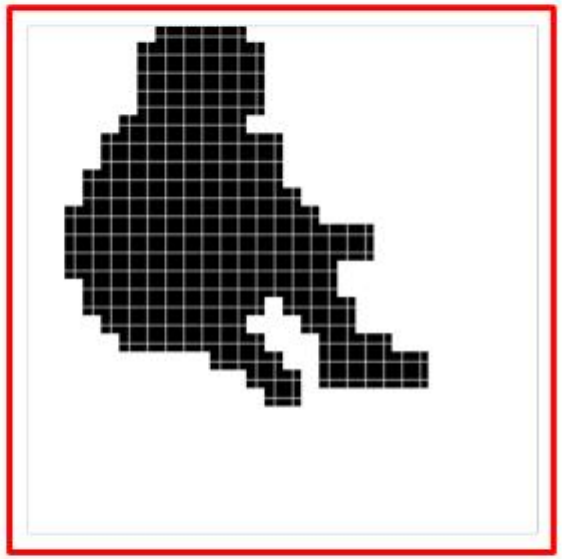
Kaiming He, ICCV 2017 Tutorial

Fully-Conv on RoI



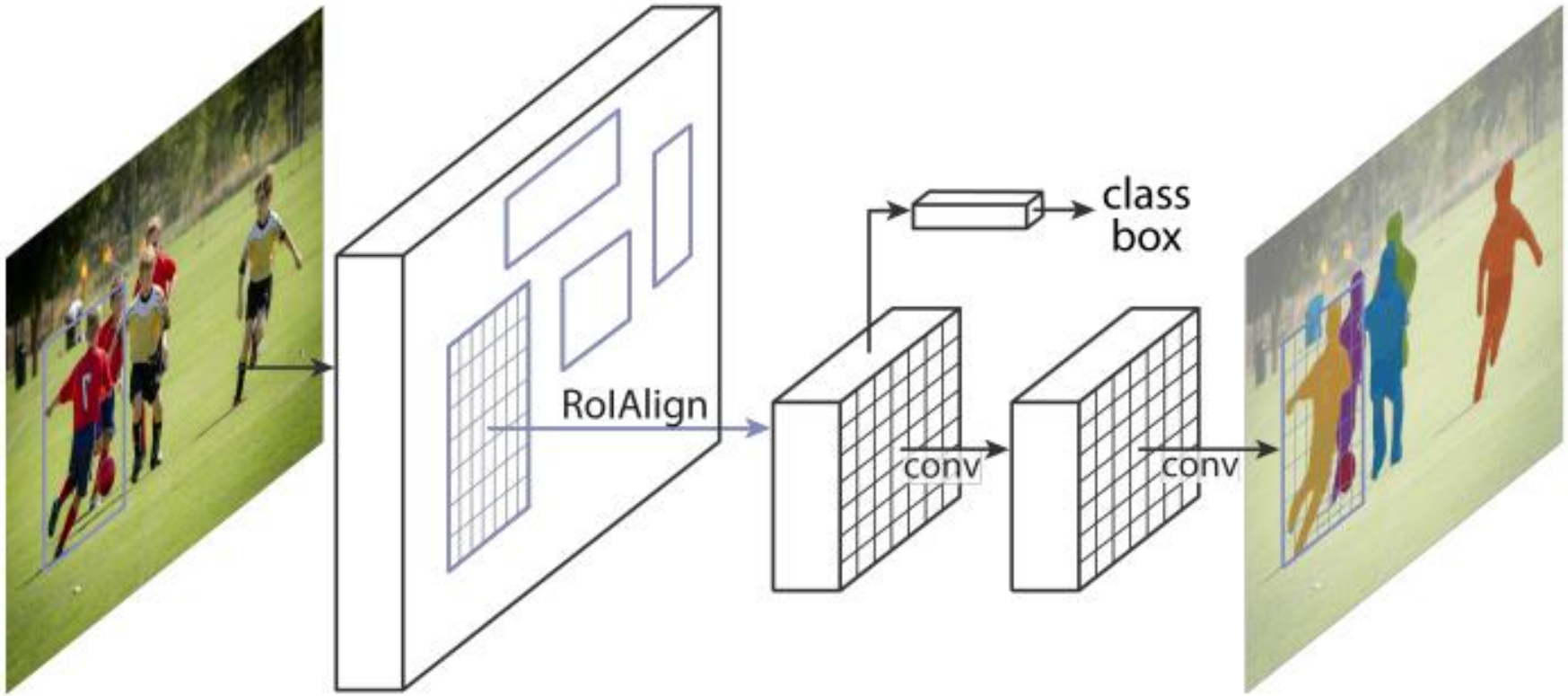
target masks on RoIs





Translation of object in RoI => Same translation of mask in RoI

- Equivariant to small translation of RoIs
- More robust to RoI's localization imperfection

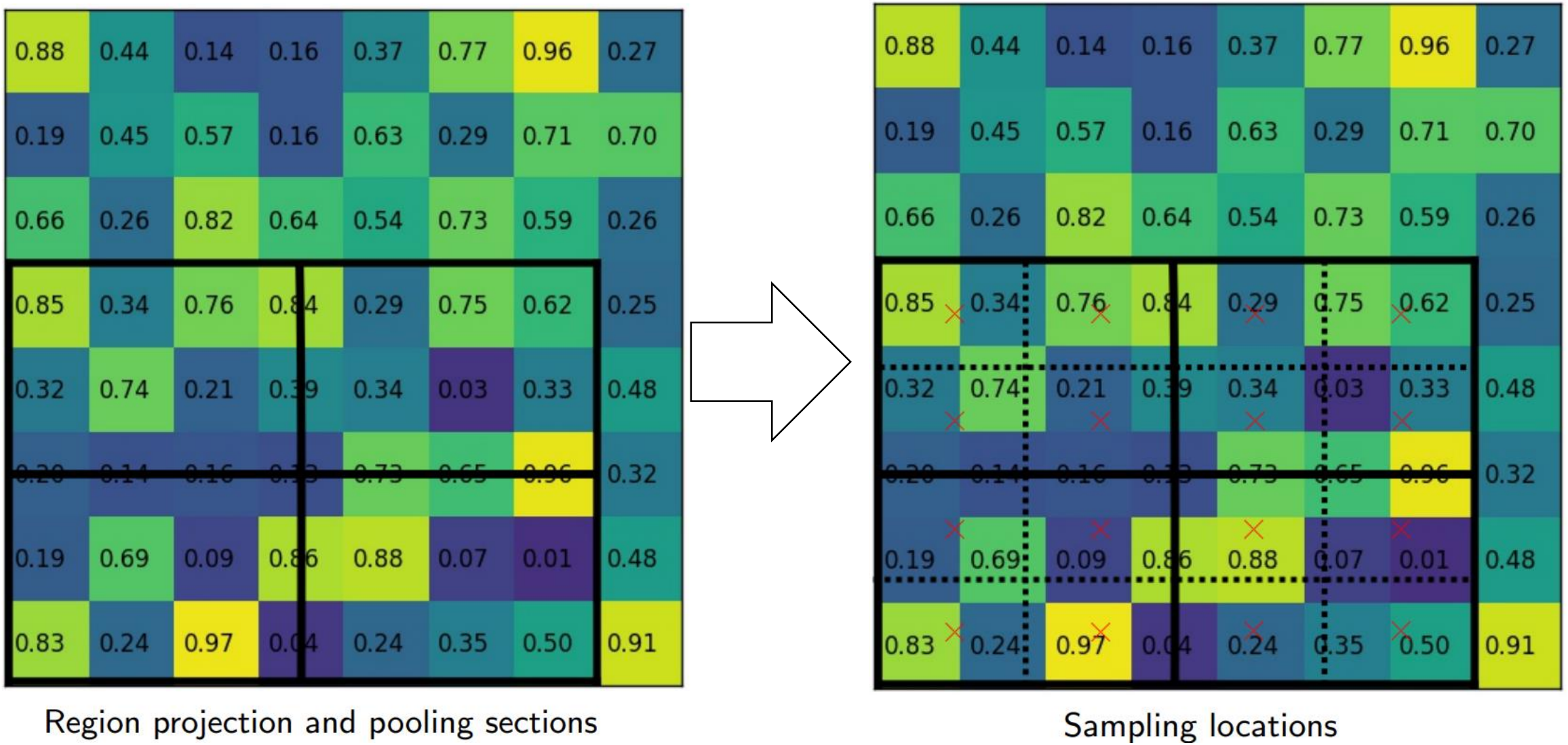


3. RoIAlign:

3a. maintain translation-equivariance before/after RoI

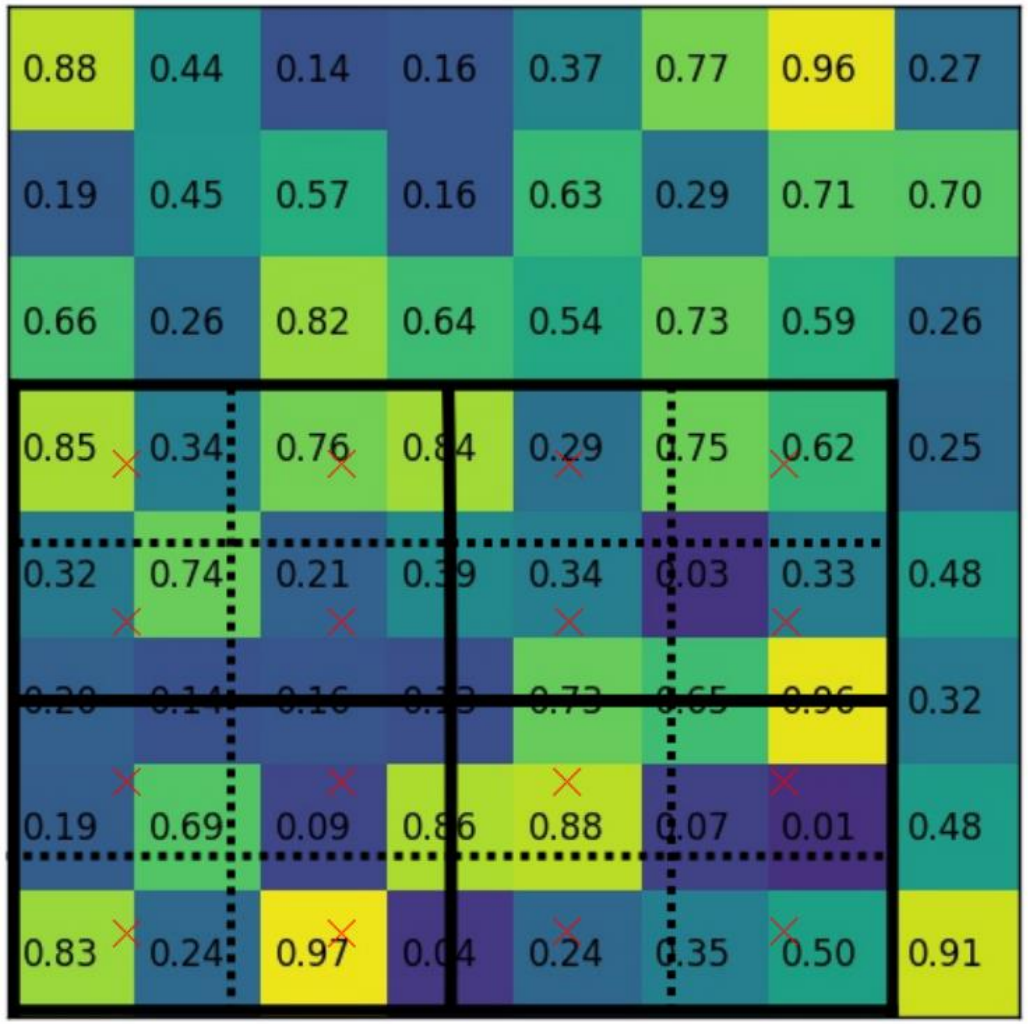
Kaiming He, ICCV 2017 Tutorial

RoI Align in Mask R-CNN



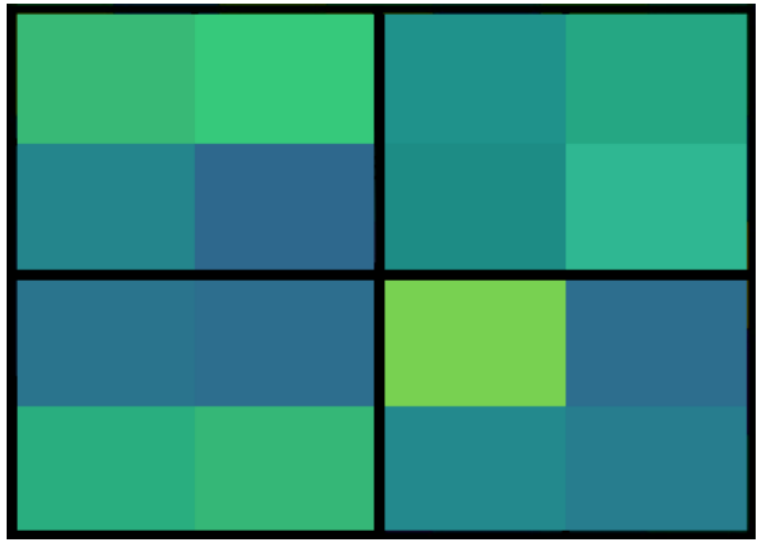
Slide credit to Silvio Galesso

RoI Align in Mask R-CNN



Sampling locations

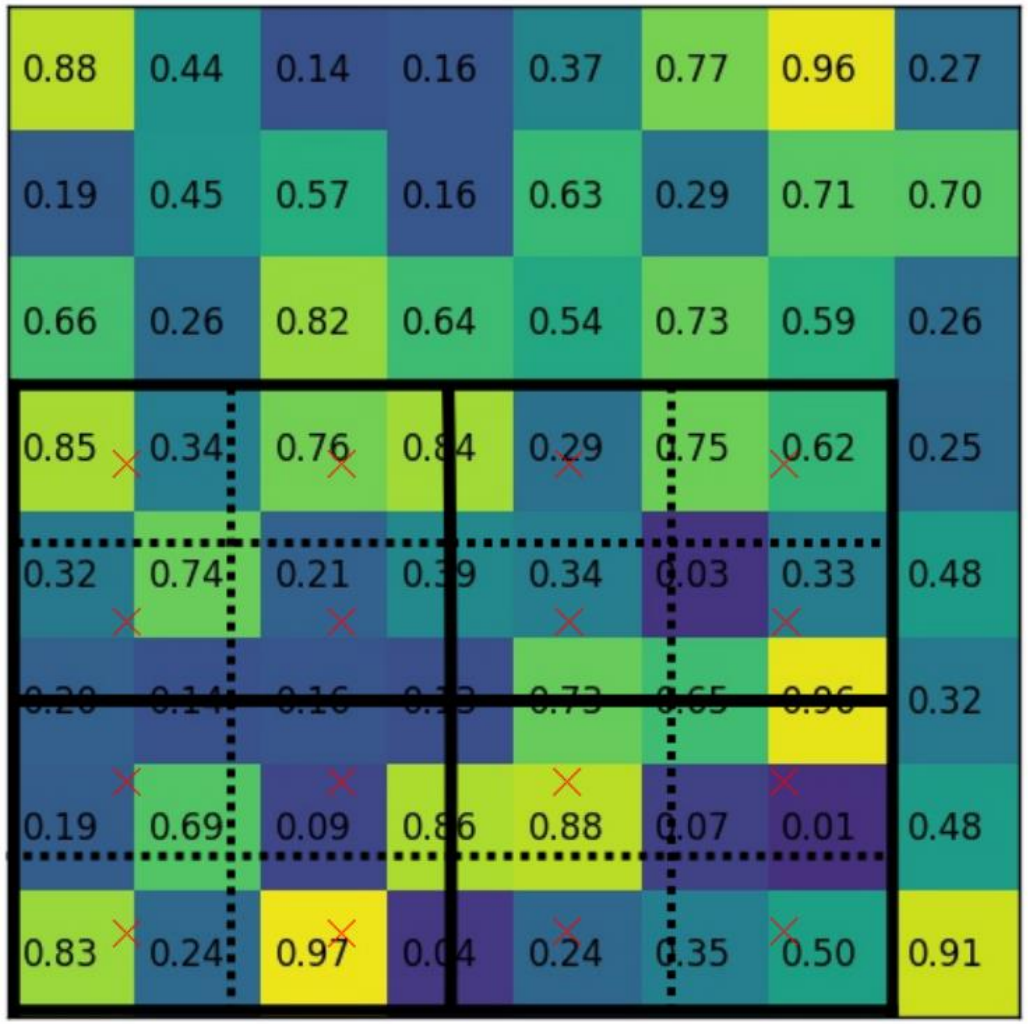
RoI Align



Bilinear interpolated values

Slide credit to Silvio Galesso

RoI Align in Mask R-CNN



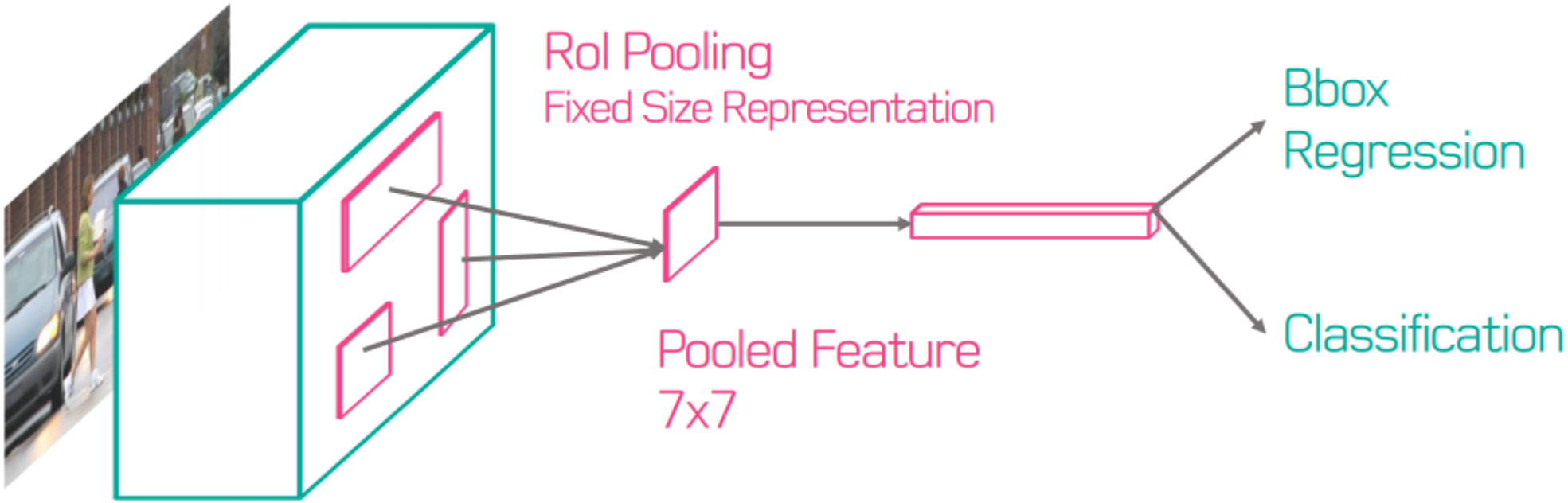
Sampling locations

RoI Align

Max pooling output

Slide credit to Silvio Galesso

RoI–Align



Slide credit to T. Kim

RoI Pooling (Fast R-CNN)

- Input: Each RoI
- Output: 7x7 Pooled Feature

RoI Align (Mask R-CNN)

- Input: Each RoI
- Output: 7x7 Pooled Feature

RoI Pooling (Fast R-CNN)

- Input: Each RoI
- Output: 7x7 Pooled Feature

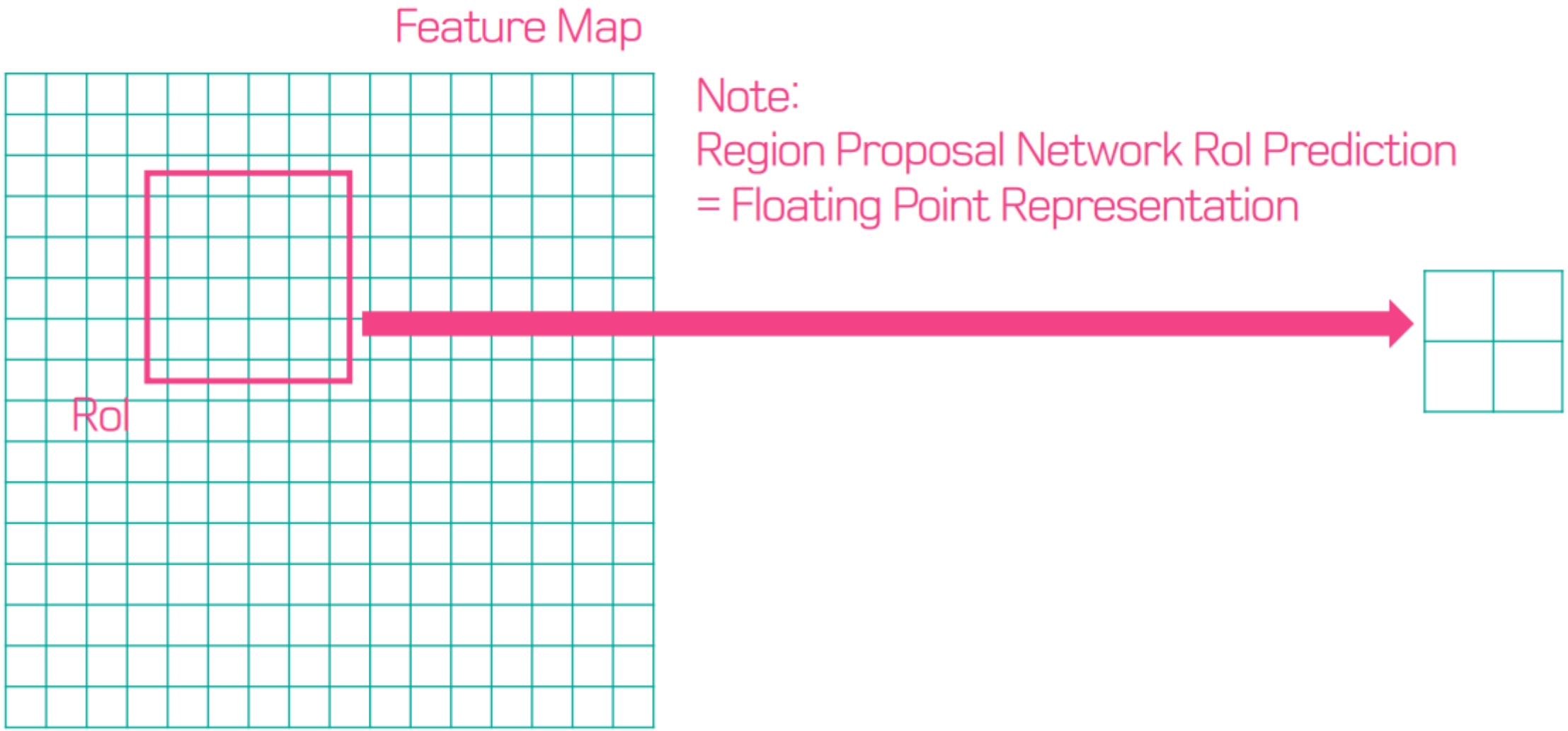
- 7x7 Feature for FCN
- Too Small to Segment
- Need Precise Pooled Feature

RoI Align (Mask R-CNN)

- Input: Each RoI
- Output: 7x7 Pooled Feature

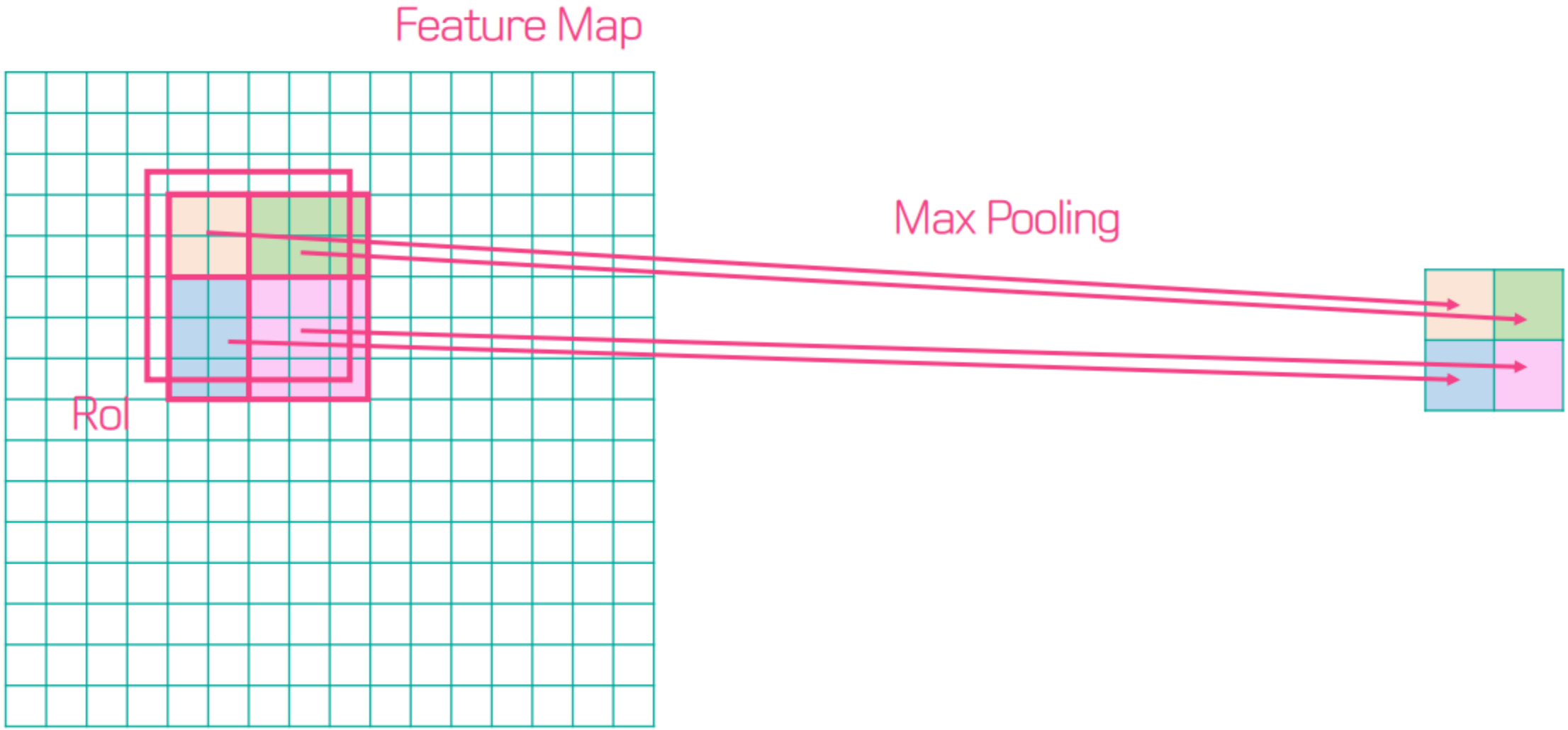
Slide credit to T. Kim

RoI Pooling



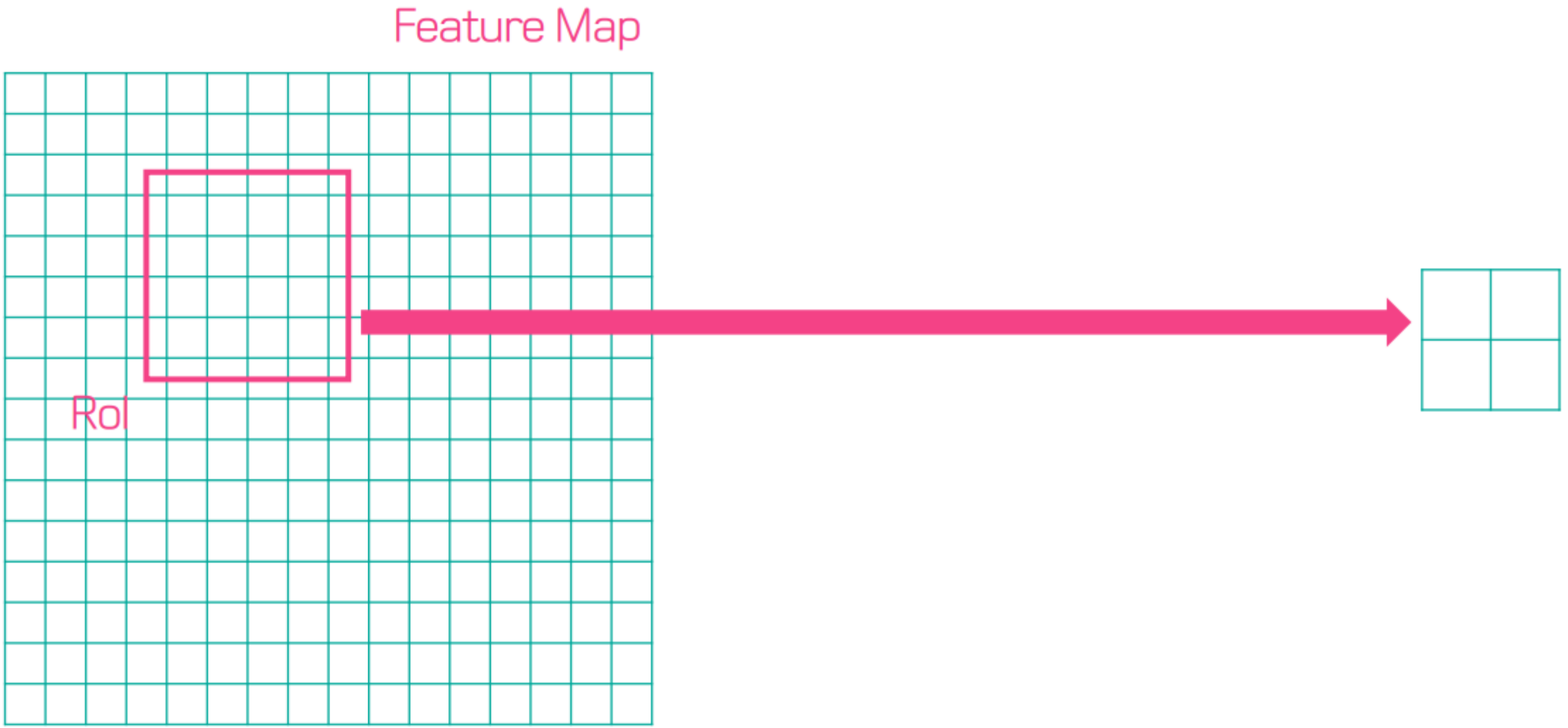
Slide credit to T. Kim

RoI Pooling



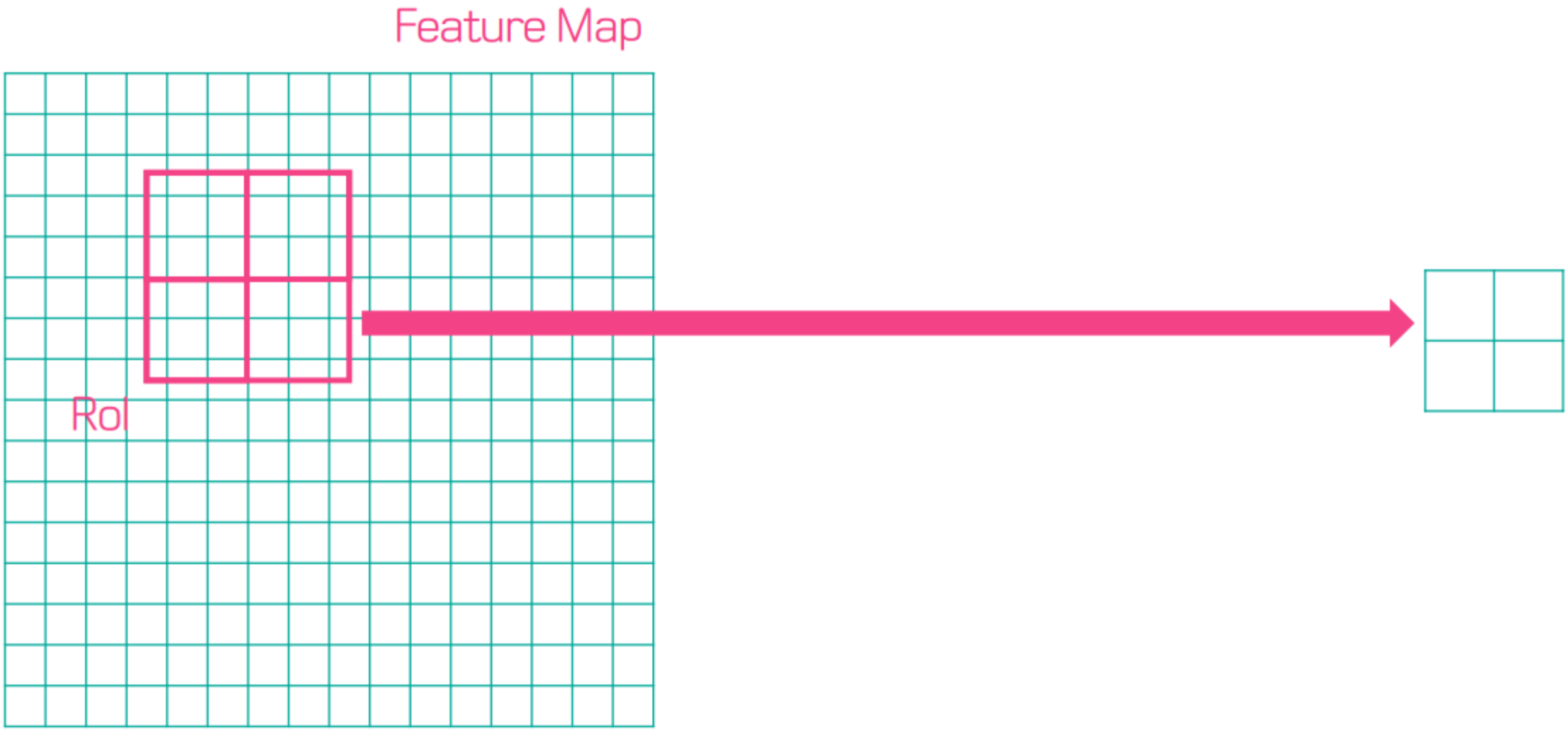
Slide credit to T. Kim

RoI-Align



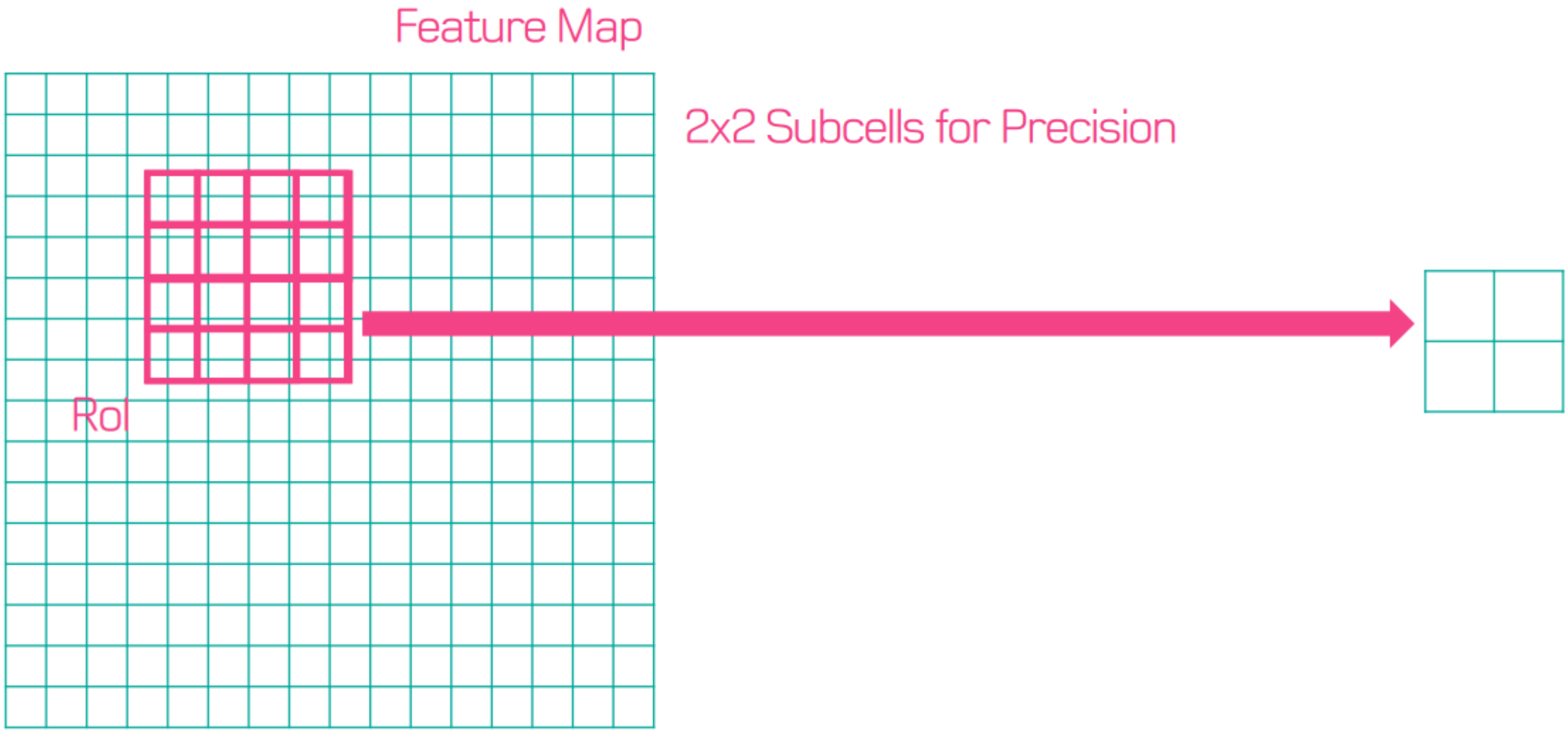
Slide credit to T. Kim

RoI-Align



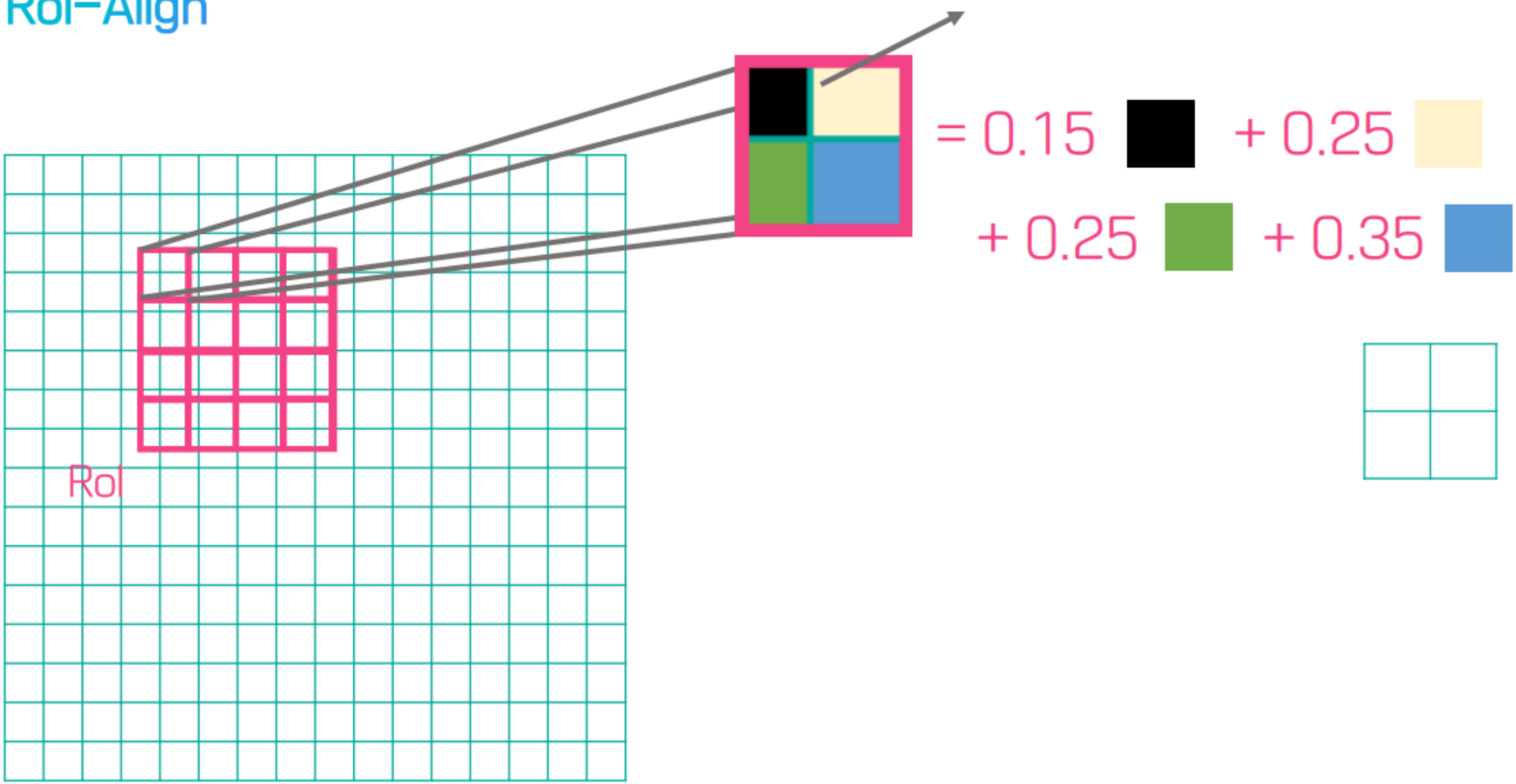
Slide credit to T. Kim

RoI-Align



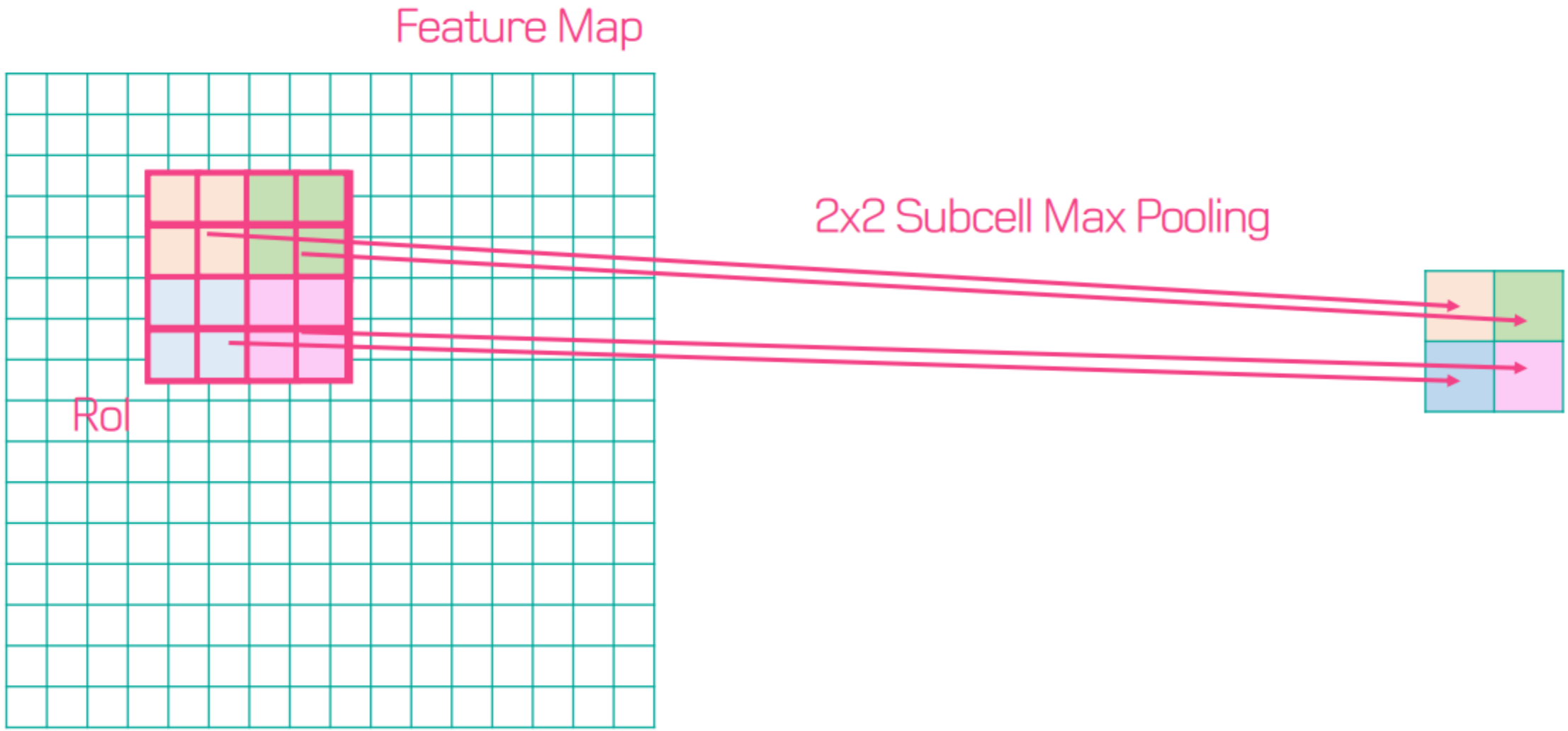
Slide credit to T. Kim

RoI-Align



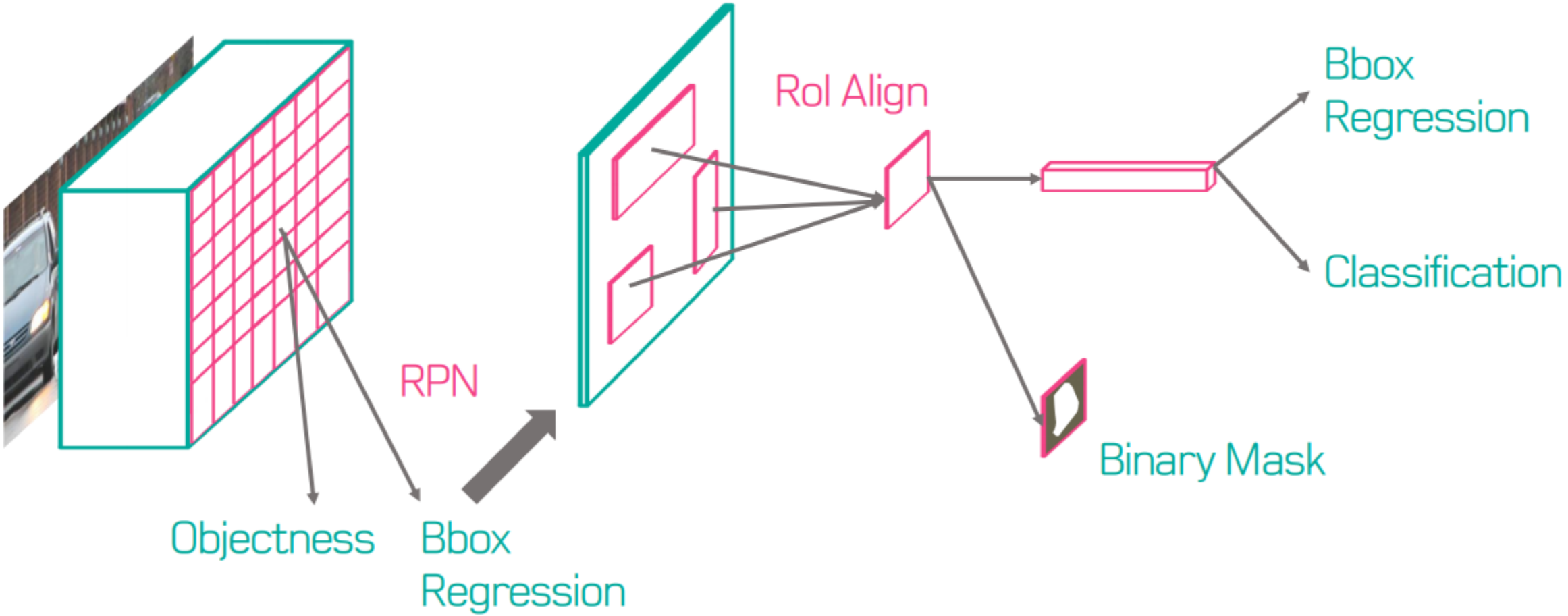
Slide credit to T. Kim

RoI-Align



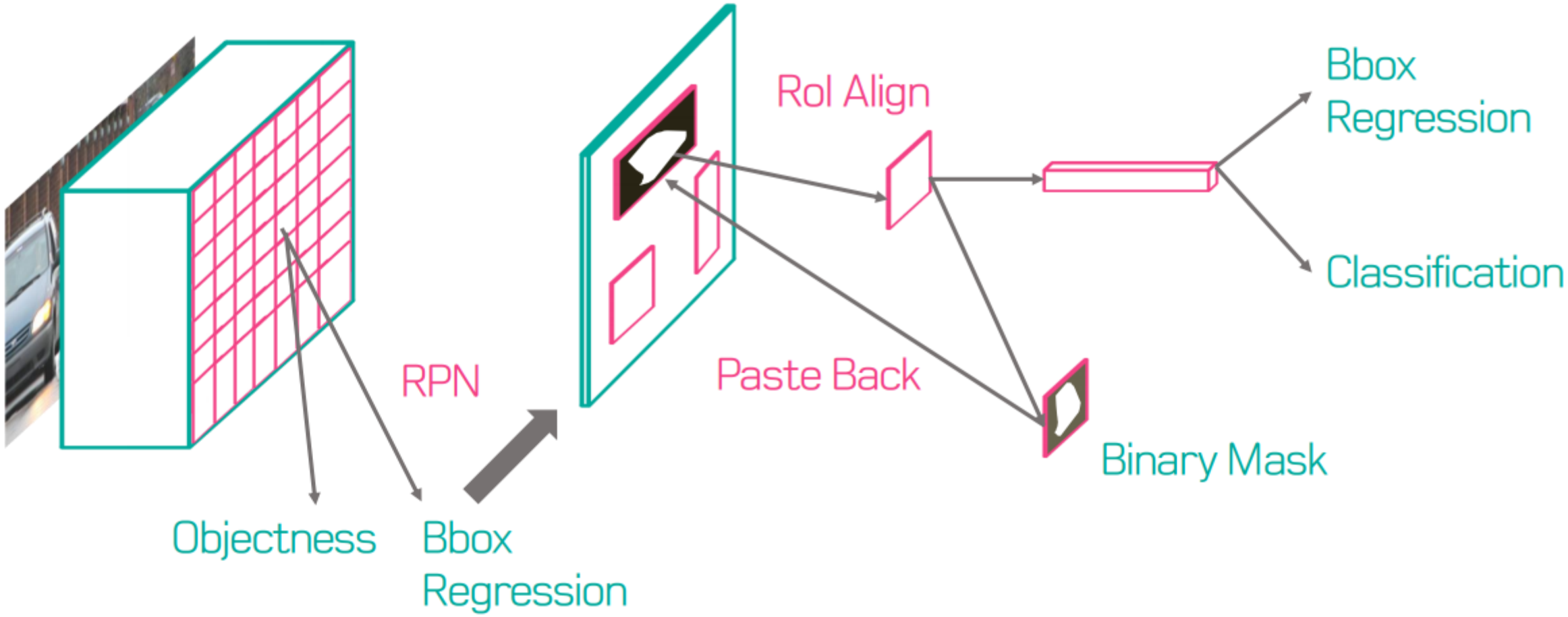
Slide credit to T. Kim

RoI-Align



Slide credit to T. Kim

RoI–Align Paste Back



Slide credit to T. Kim

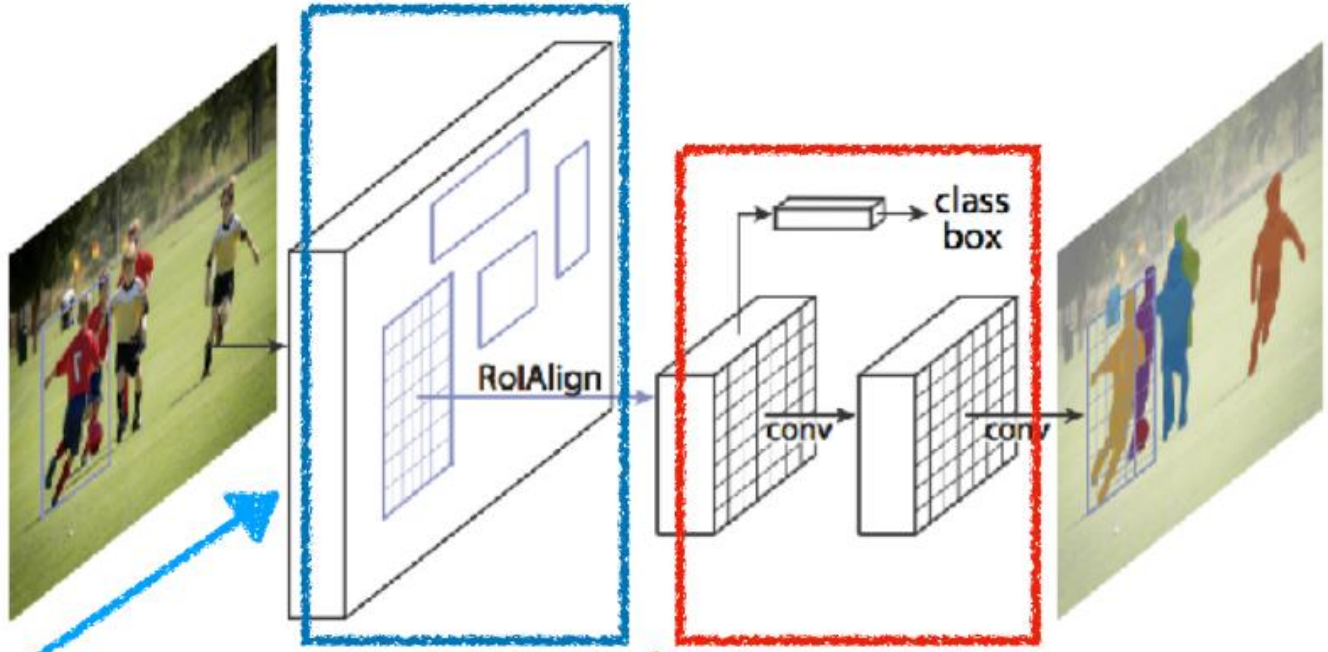


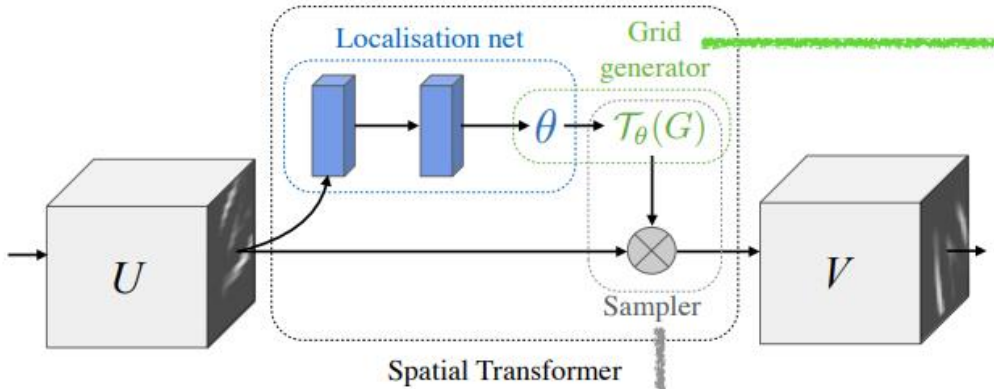
Figure 1. The **Mask R-CNN** framework for instance segmentation.

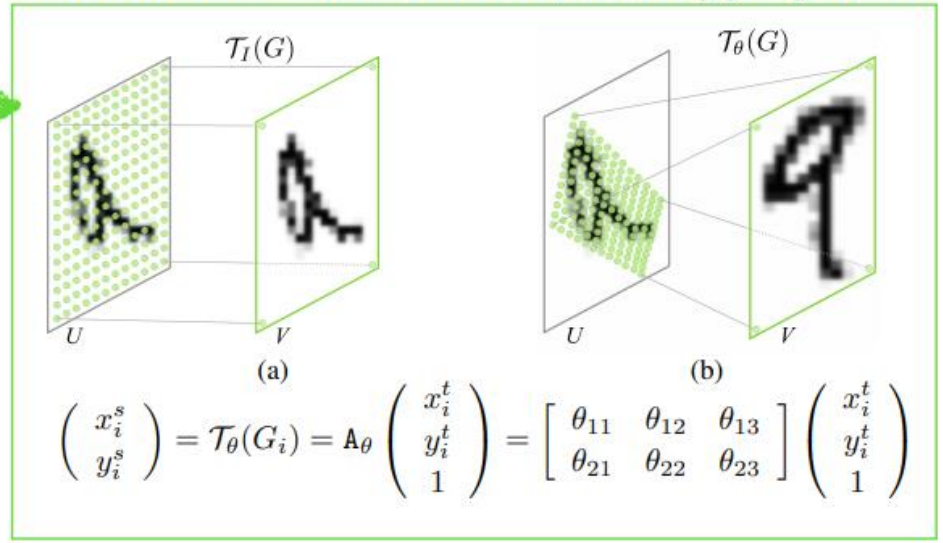
1. To fix the misalignment, we propose a simple, quantization-free layer, called RoIAlign, that faithfully preserves exact spatial locations.

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

RoIAlign

We use bilinear interpolation(Spatial transformer networks, Jaderberg, et al.) to compute the exact values of the input features at four regularly sampled locations in each RoI bin, and aggregate the result (using max or average).





Input feature value at location (n,m) in channel c

Sampling kernel

Kernel parameters

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \quad \forall i \in [1 \dots H'W'] \quad \forall c \in [1 \dots C]$$

Target feature value at location i in channel c

Sampling coordinate

bilinear sampling kernel

Copying the value at the nearest pixel to (x^s_i , y_i^s) to the output location (x^t_i , y_i^t).

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

RoIAlign

Source: https://deepsense.io/region-of-interest-pooling-explained/

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

Input activation

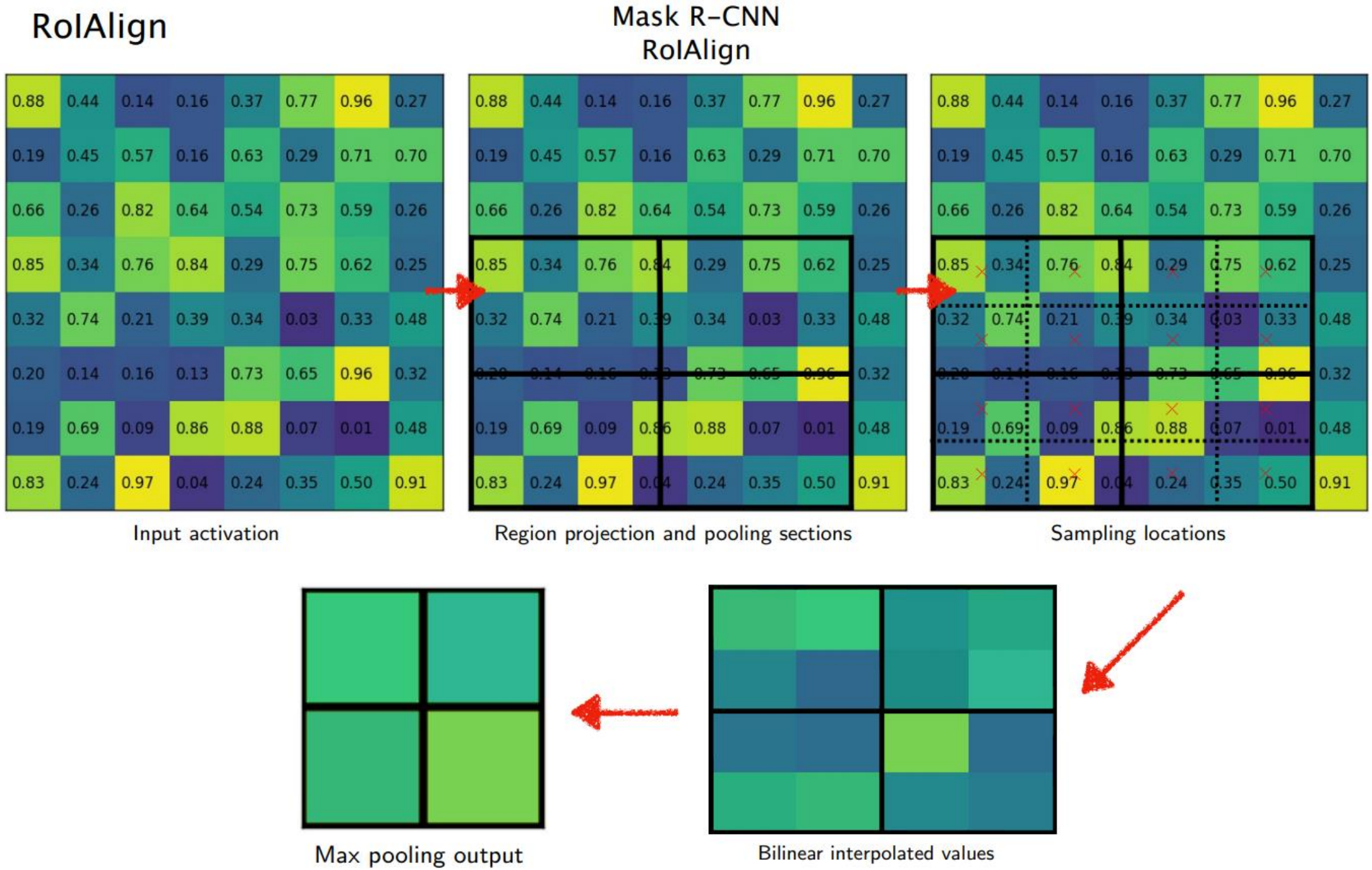
Faster R-CNN
RoIPool

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

Region projection and pooling sections

0.85	0.84
0.97	0.96

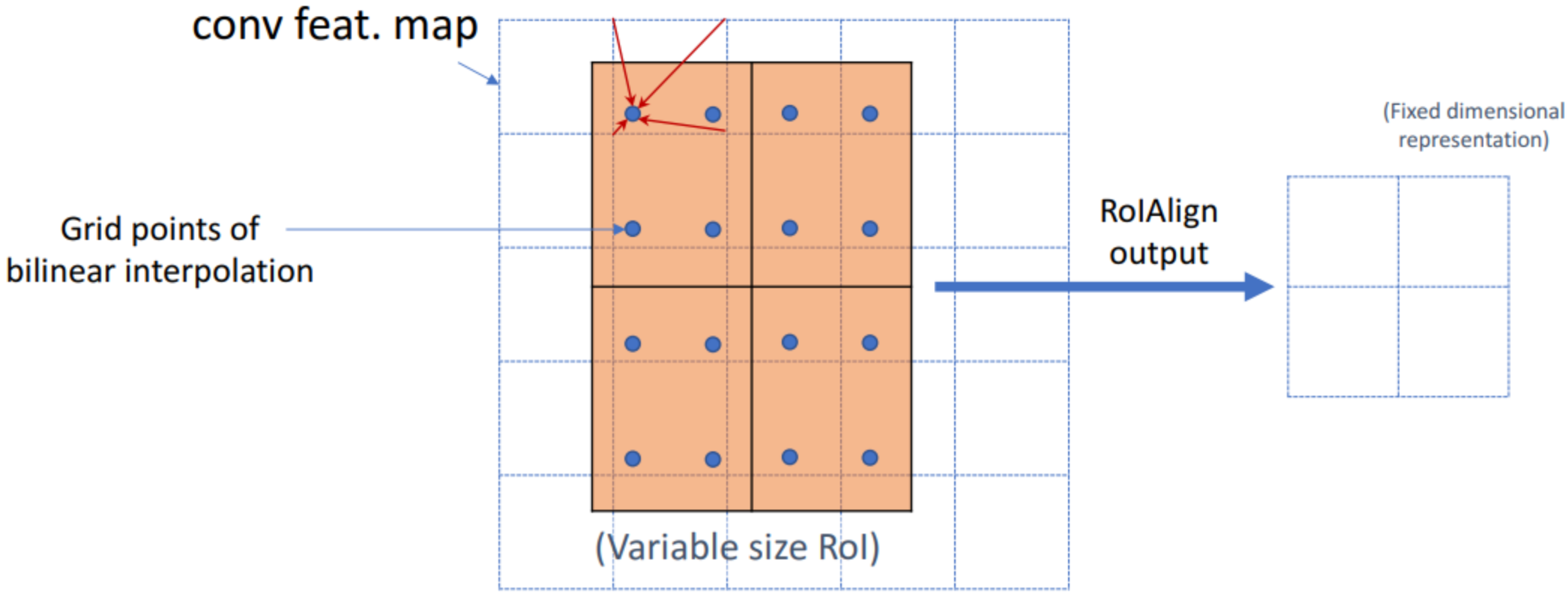
Max pooling output


RoIAlign
Mask R-CNN
RoIAlign
0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91
Input activation
Region projection and pooling sections
Sampling locations
Max pooling output
Bilinear interpolated values

RoIAlign

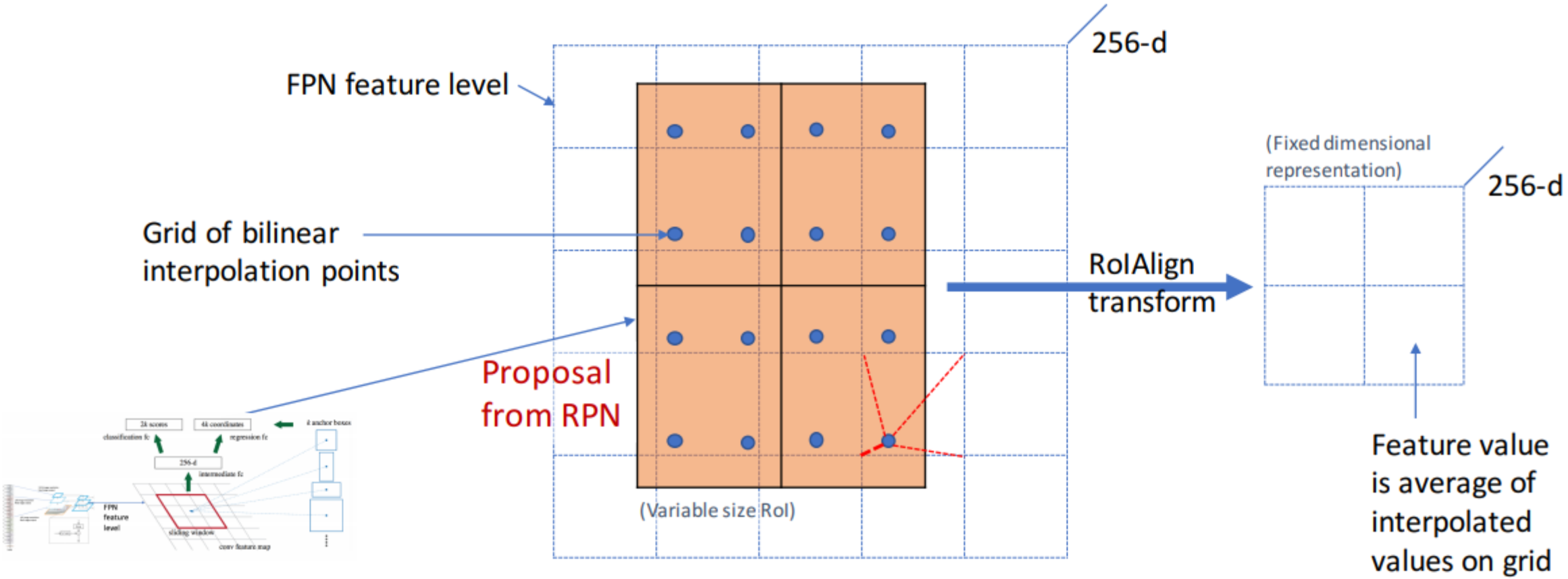
FAQs: how to sample grid points within a cell?

- 4 regular points in 2x2 sub-cells
- other implementation could work



RoIAlign Transform (on each Proposal)

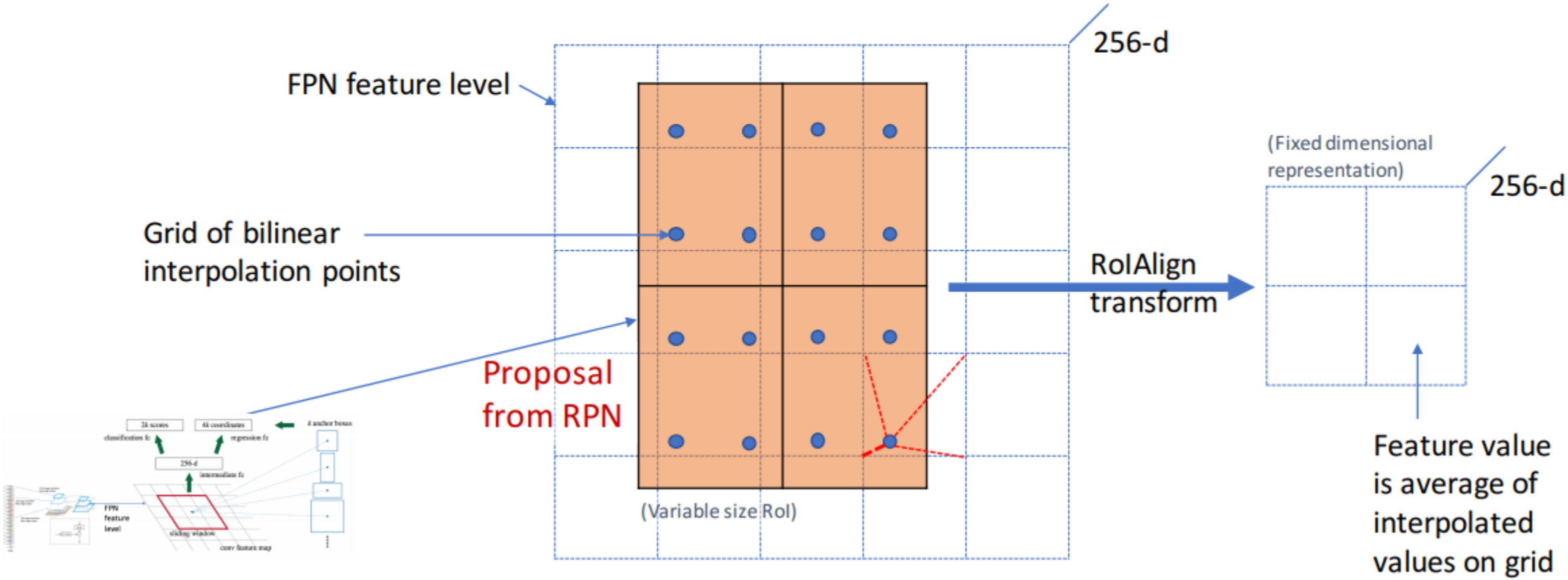
Smoothly normalize features and predictions into coordinate frame
free of scale and aspect ratio



Slide credit to Ross Girshick

RoIAlign Transform (on each Proposal)

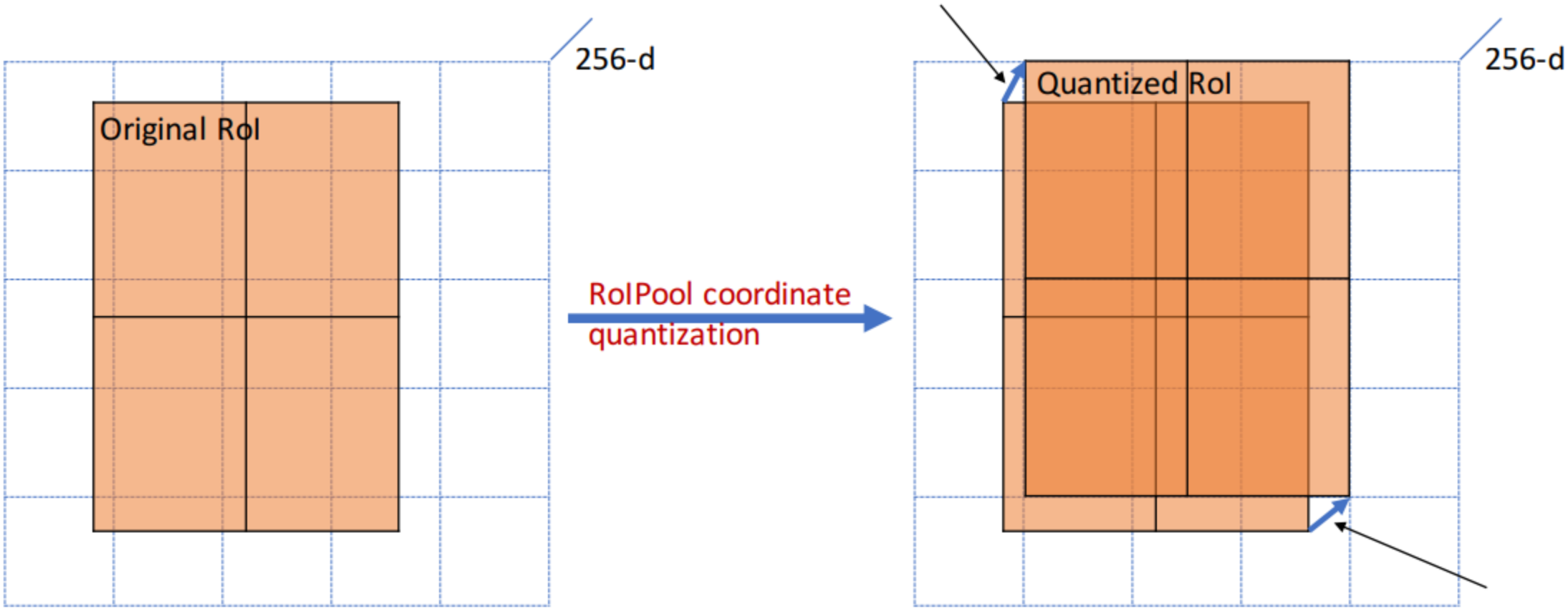
Key: No coordinate quantization (cf. RoIPool in Fast R-CNN, etc.)



Slide credit to Ross Girshick

Compare to RoIPool, RoIWarp, and others

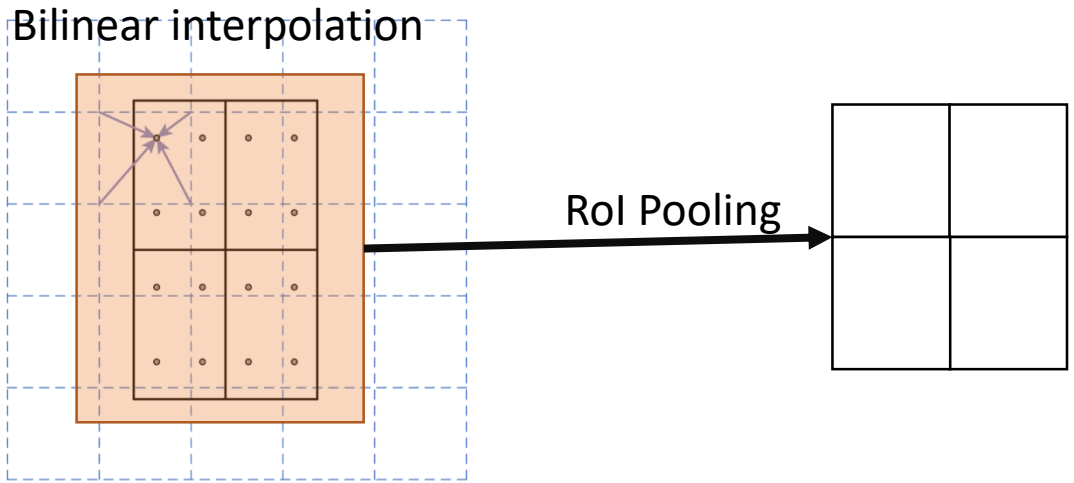
Quantization breaks pixel-to-pixel alignment



Slide credit to Ross Girshick

Mask R-CNN

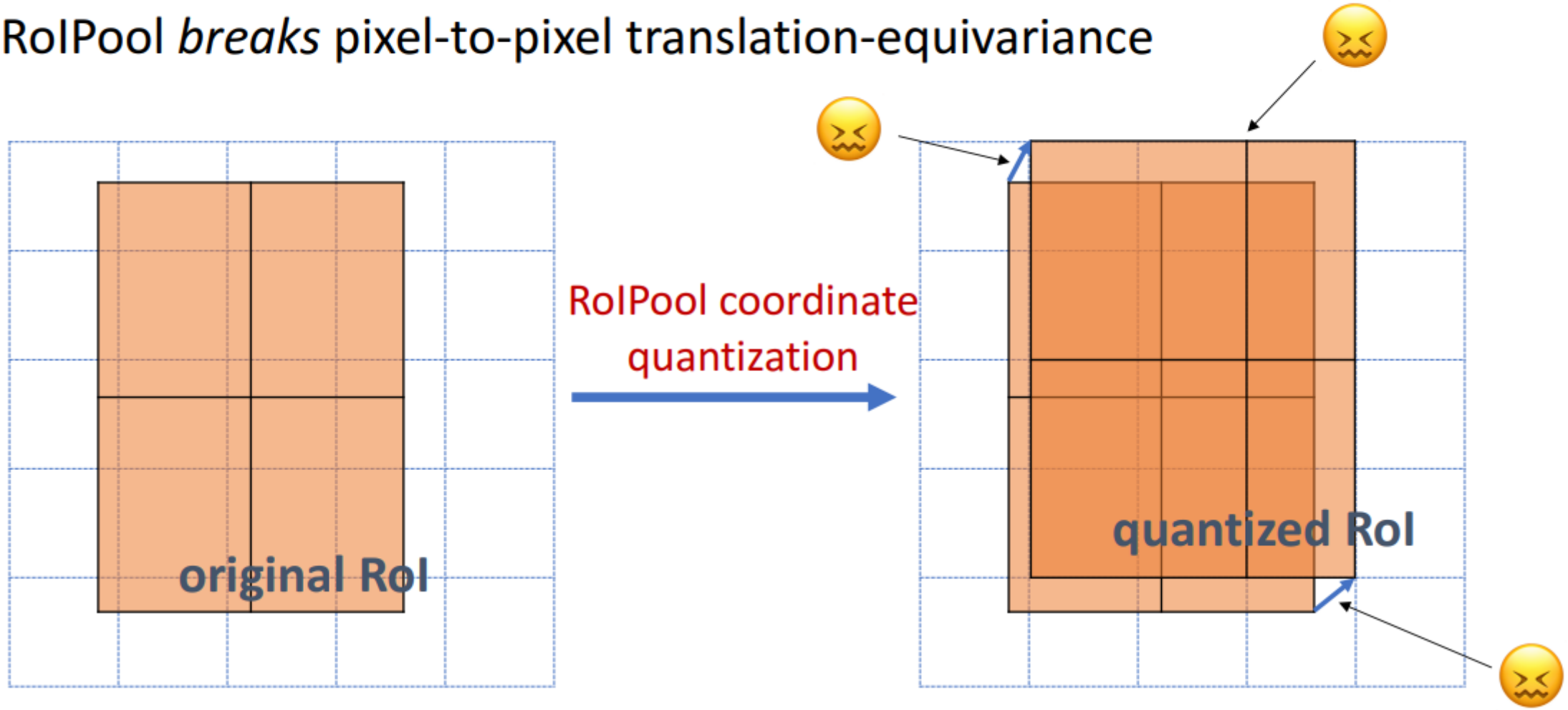
- **RoI Align** is Improves *miss-align problems* of RoI pooling
  - RoI Align use **bilinear interpolation** to generate new feature map.
  - Do RoI Pooling with *aligned* feature map

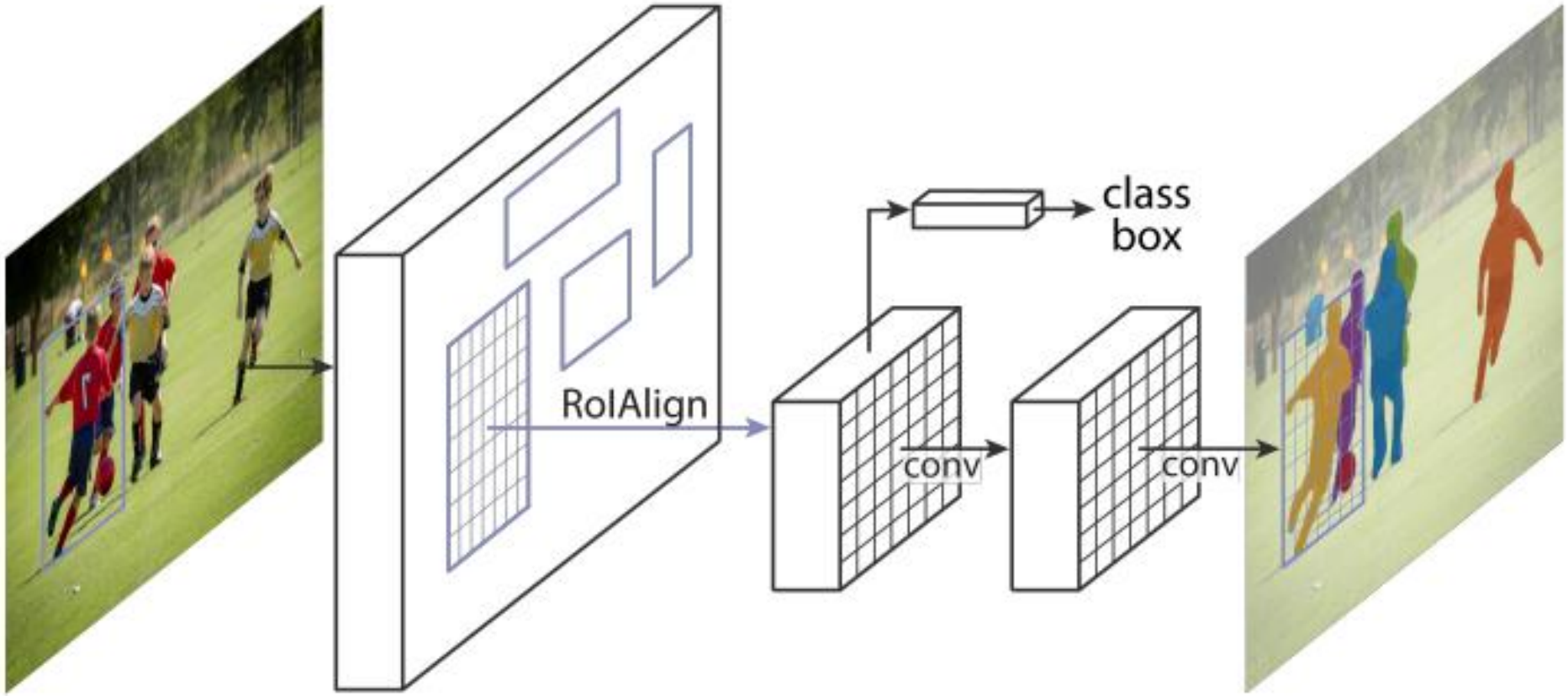


Slide credit to Ke-Shuan Cheng

RoIAlign vs. RoIPool

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



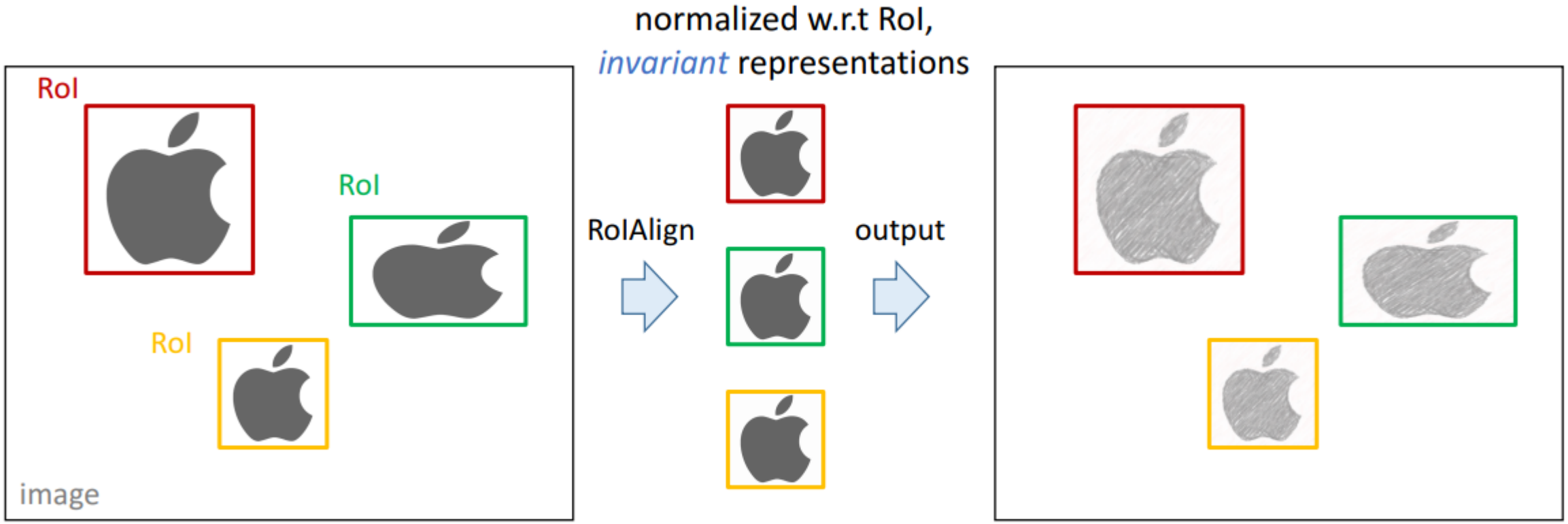


3. RoIAlign:

3b. Scale-equivariant (and aspect-ratio-equivariant)

Kaiming He, ICCV 2017 Tutorial

RoIAlign: Scale-Equivariance



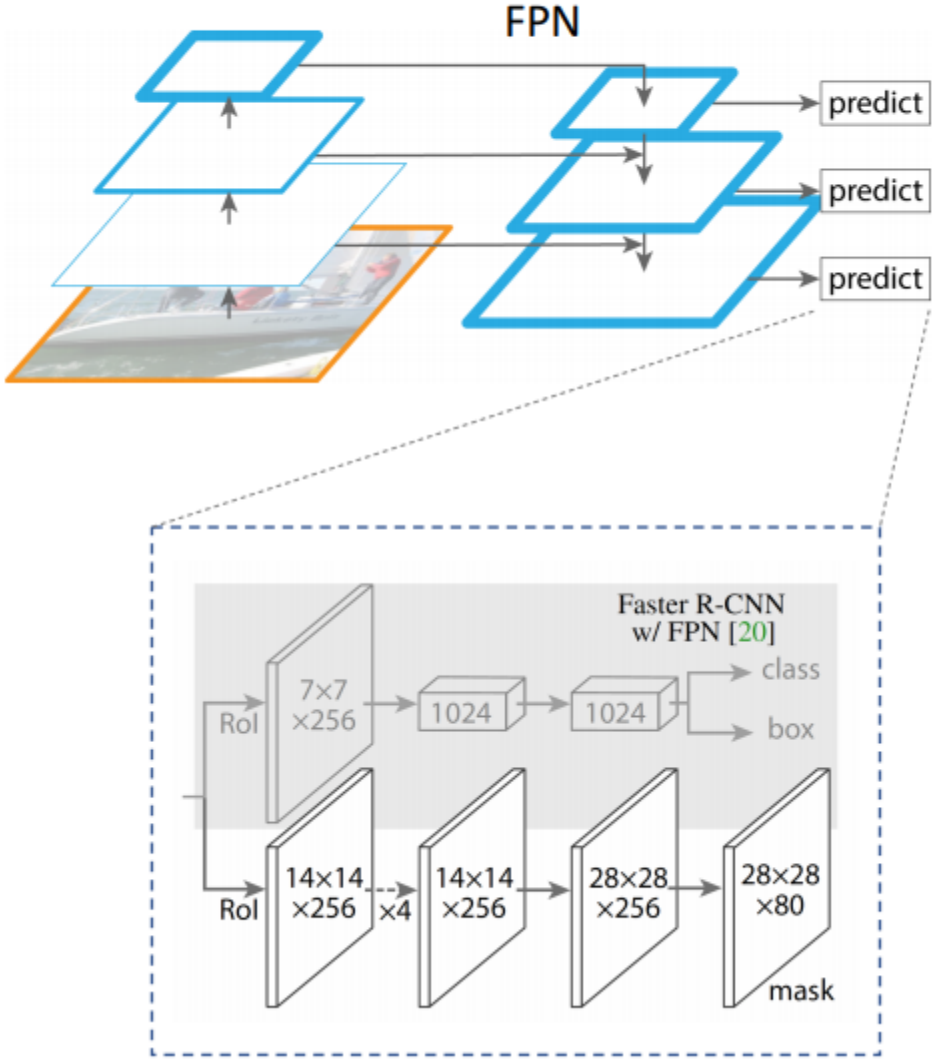
- RoIAlign creates *scale-invariant* representations
- RoIAlign + "output pasted back" provides *scale-equivariance*

More about Scale-Equivariance: FPN

- RoIAlign is scale-invariant if on raw pixels:
  - = (slow) R-CNN: crops and warps RoIs
- RoIAlign is scale-invariant if on scale-invariant feature maps

- Feature Pyramid Network (FPN) [Lin et al. CVPR'17] creates approx. scale-invariant features

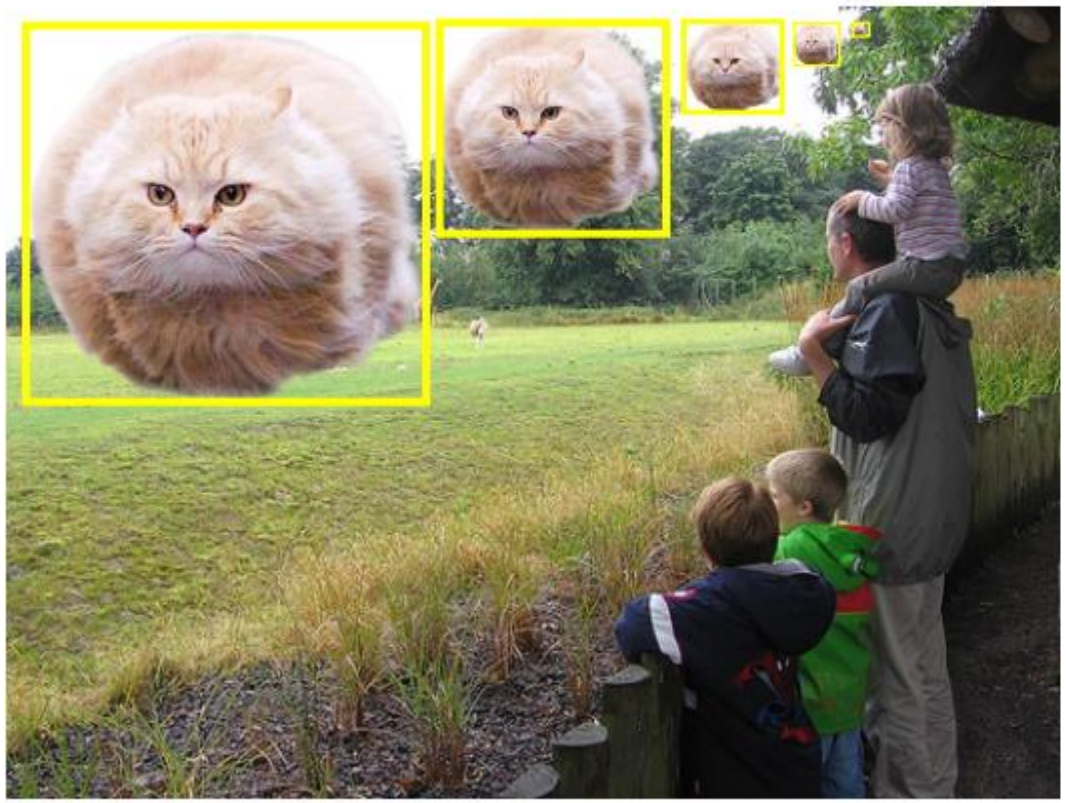
FPN (Feature Pyramid Network)

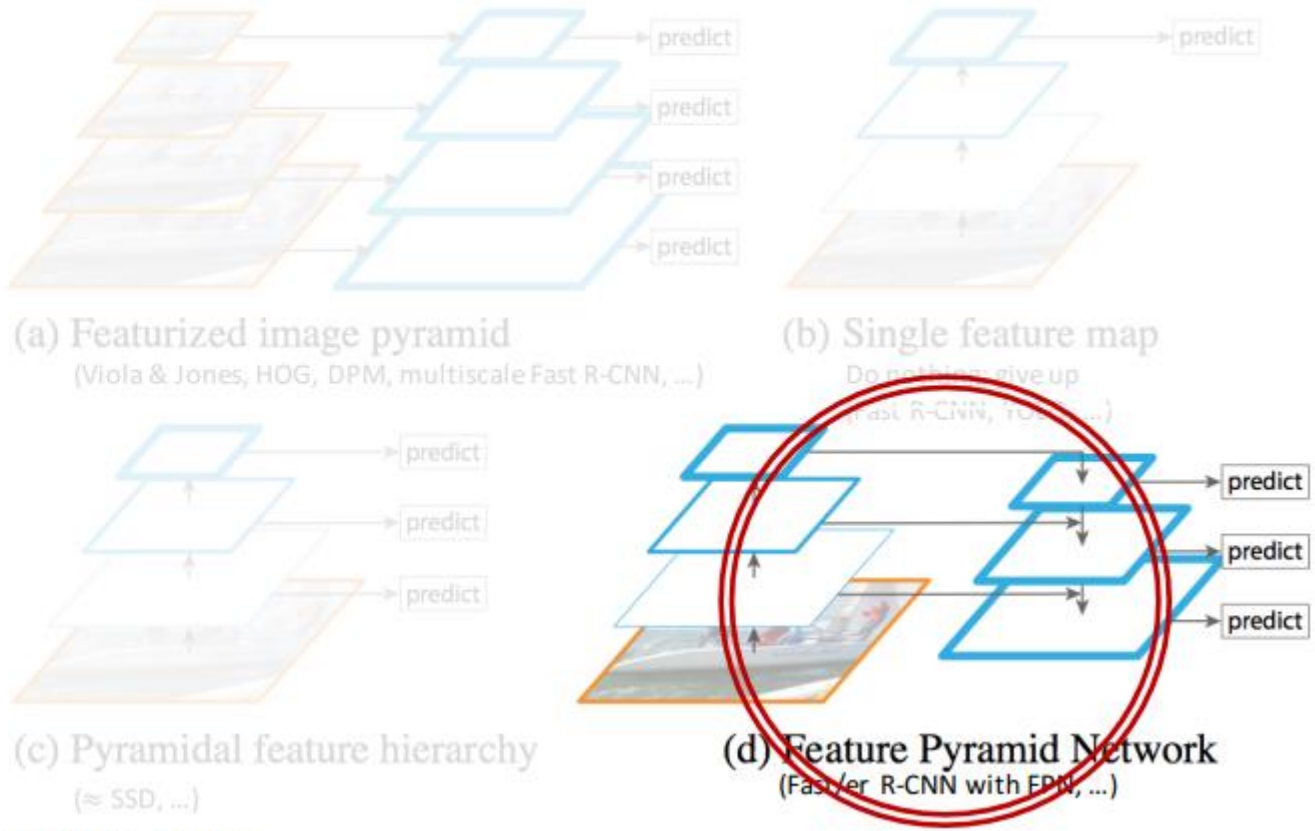


Kaiming He, ICCV 2017 Tutorial

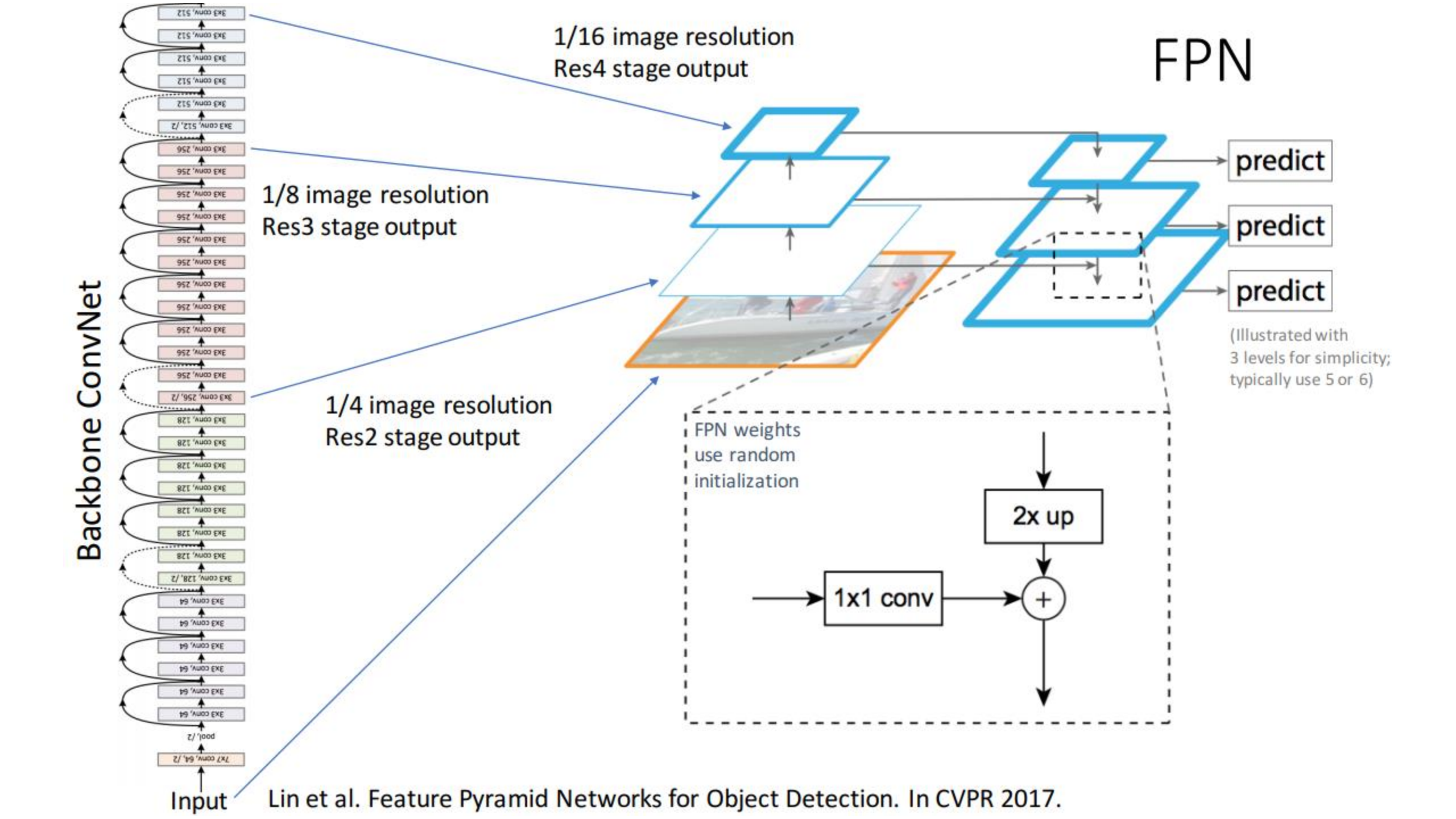
Feature Pyramid Network (FPN)

[Optional, but recommended]





Lin et al. Feature Pyramid Networks for Object Detection. In CVPR 2017.



Lin et al. Feature Pyramid Networks for Object Detection. In CVPR 2017.

Equivariance in Mask R-CNN: Summary

- Translation-equivariant
  - FCN features
  - FCN mask head
  - RoIAlign (pixel-to-pixel behavior)

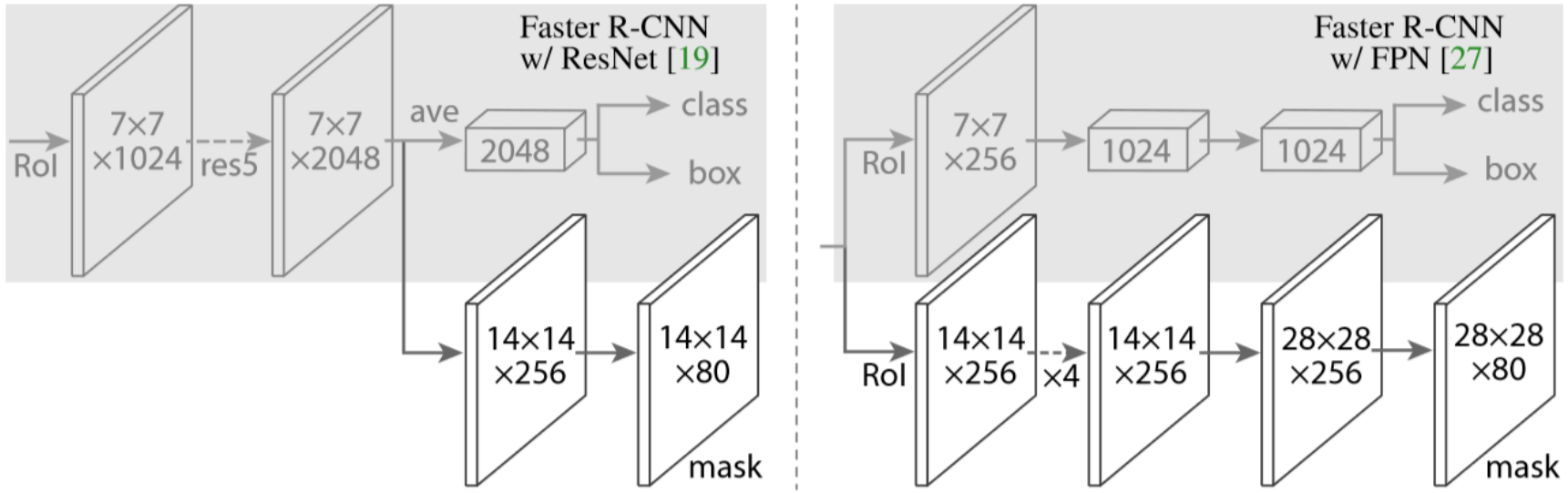
- Scale-equivariant (and aspect-ratio-equivariant)
  - RoIAlign (warping and normalization behavior) + paste-back
  - FPN features

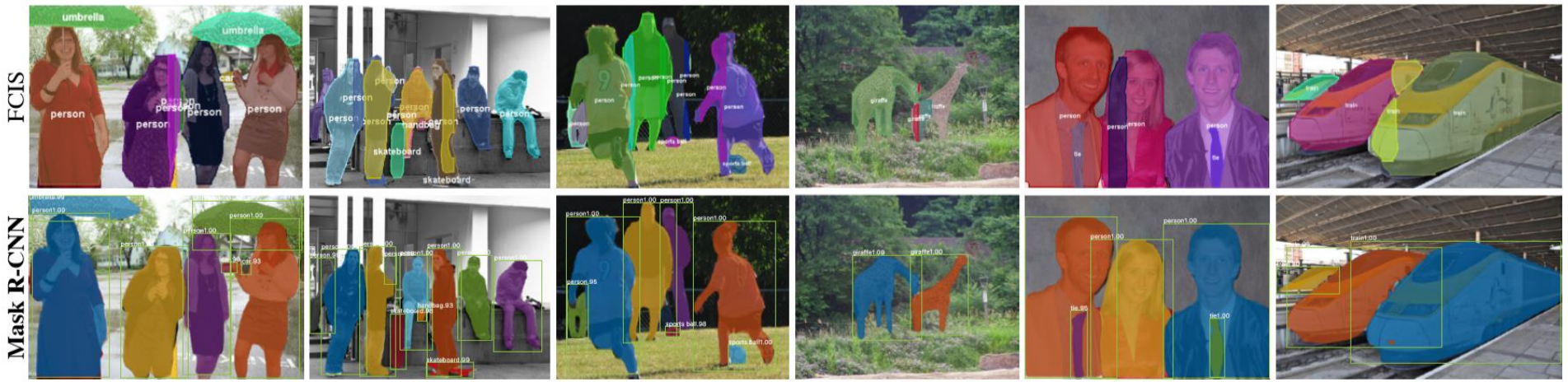
Mask R-CNN

- They proposed 2 architecture and compared they for Object Mask branch
  - ResNet
    - Branch from last Convolutional layer
  - Feature Pyramid Network(FPN)
    - Branch from RoI

Slide credit to Ke-Shuan Cheng

Mask R-CNN





Slide credit to Ke-Shuan Cheng

Mask R-CNN

- **Loss function** is defined as below:

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

  - L_{cls}: Cross-Entropy
  - L_{box}: IoU

$$\frac{Area\ of\ OverLap}{Area\ of\ Union}$$

  - L_{mask}: Cross-Entropy between pixel-to-pixel

Slide credit to Ke-Shuan Cheng

Network architecture

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

A. Fast R-CNN B.

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

B.

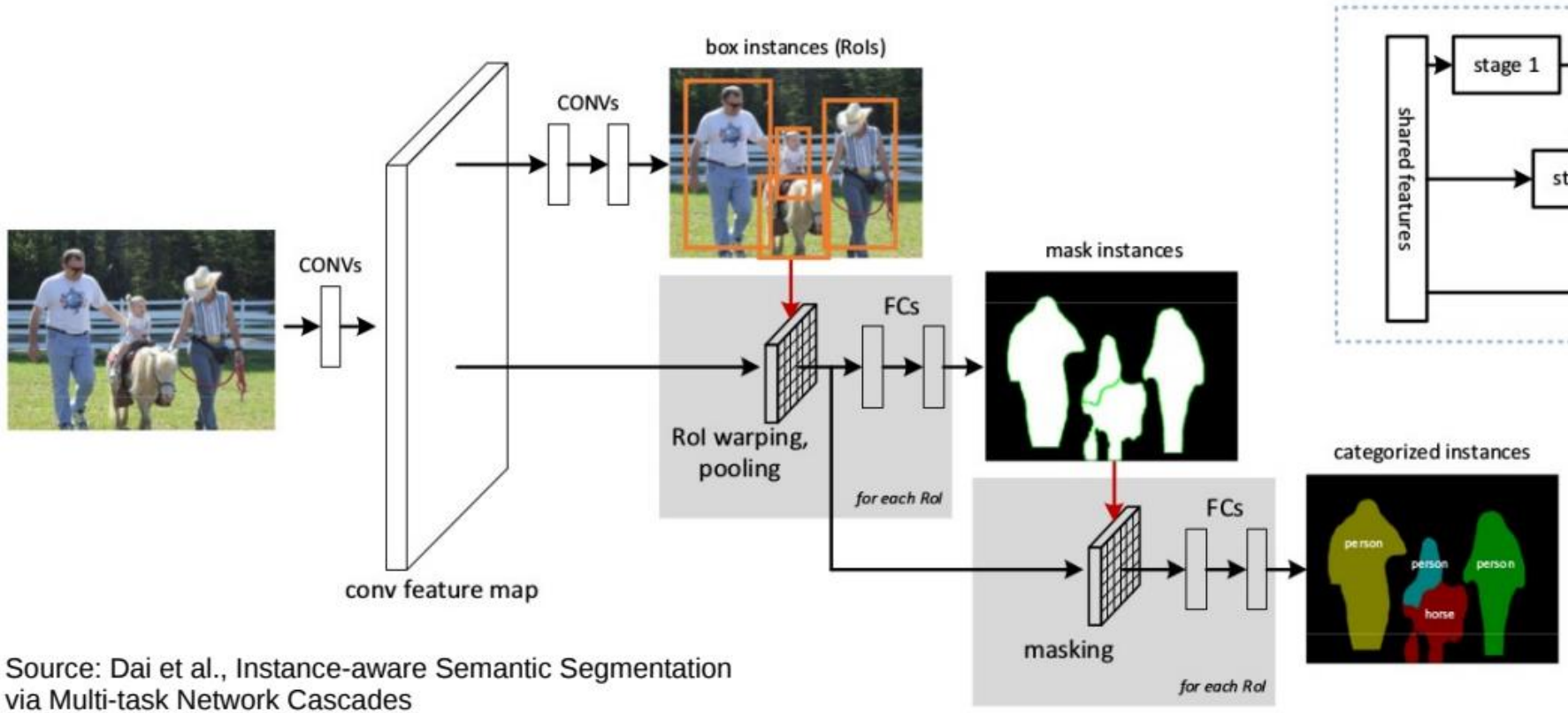
- $K \cdot (m \times m)$ sigmoid outputs:
  - → pixel-wise binary classification
  - → one mask for each class, no competition
- L_{mask}: mean binary cross-entropy

Mask R-CNN

- Result

Slide credit to Ke-Shuan Cheng

Related methods: MNC

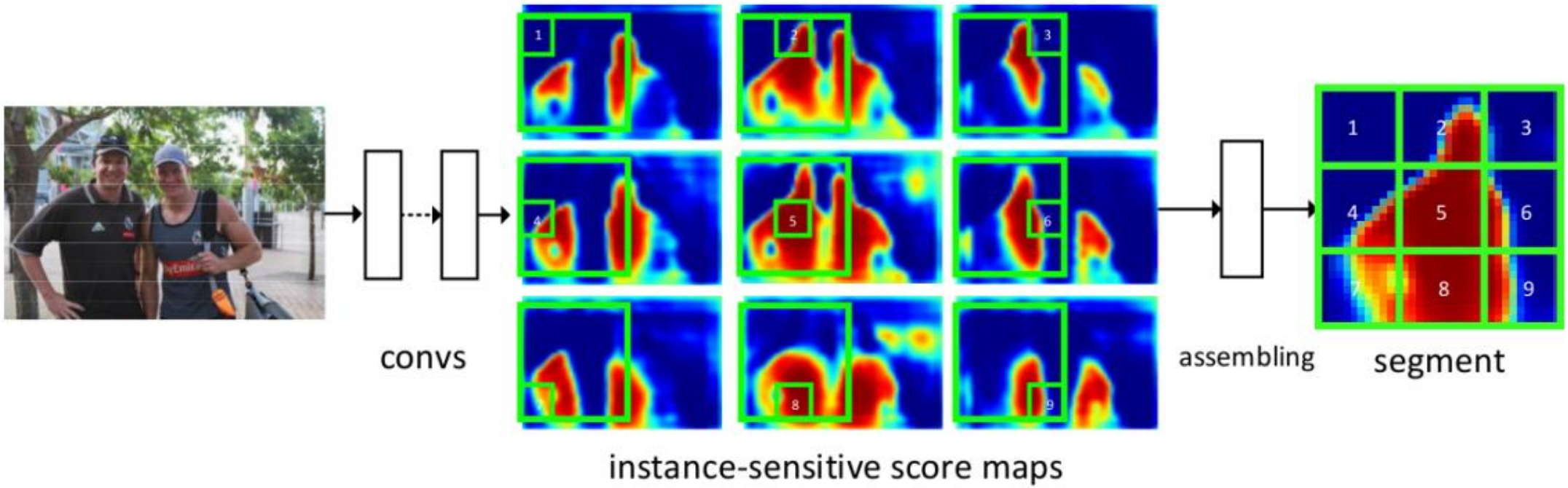


Source: Dai et al., Instance-aware Semantic Segmentation via Multi-task Network Cascades

Multi-task **N**etwork **C**ascade:

- bounding box regression
- mask estimation
- classification

Related methods: FCIS



Source: Dai et al., Instance-sensitive Fully Convolutional Networks

- **F**ully **C**onvolutional **I**nstance **S**egmentation
- Challenge: translation invariance $\rightarrow$ no instance awareness
- Proposed solution: positional aware sliding masks

References

https://arxiv.org/pdf/1506.01497.pdf (Faster R-CNN)
https://arxiv.org/pdf/1504.08083.pdf (Fast R-CNN)
https://arxiv.org/pdf/1506.06981.pdf (R-CNN minus R)
https://koen.me/research/pub/uijlings-ijcv2013-draft.pdf (Selective Search for Object Detection)
https://arxiv.org/pdf/1703.06870.pdf (Mask R-CNN)
http://host.robots.ox.ac.uk/pascal/VOC/
https://www.dropbox.com/s/xtr4yd4i5e0vw8g/iccv15_tutorial_training_rbg.pdf?dl=0
http://kaiminghe.com/iccv15tutorial/iccv2015_tutorial_convolutional_feature_maps_kaiminghe.pdf
https://lovesnowbest.site/2018/02/27/Intro-to-Object-Detection/
https://blog.deepsense.ai/region-of-interest-pooling-explained/
https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/