Image Detection: Past, Present and Future

Presented by: Wei Ji

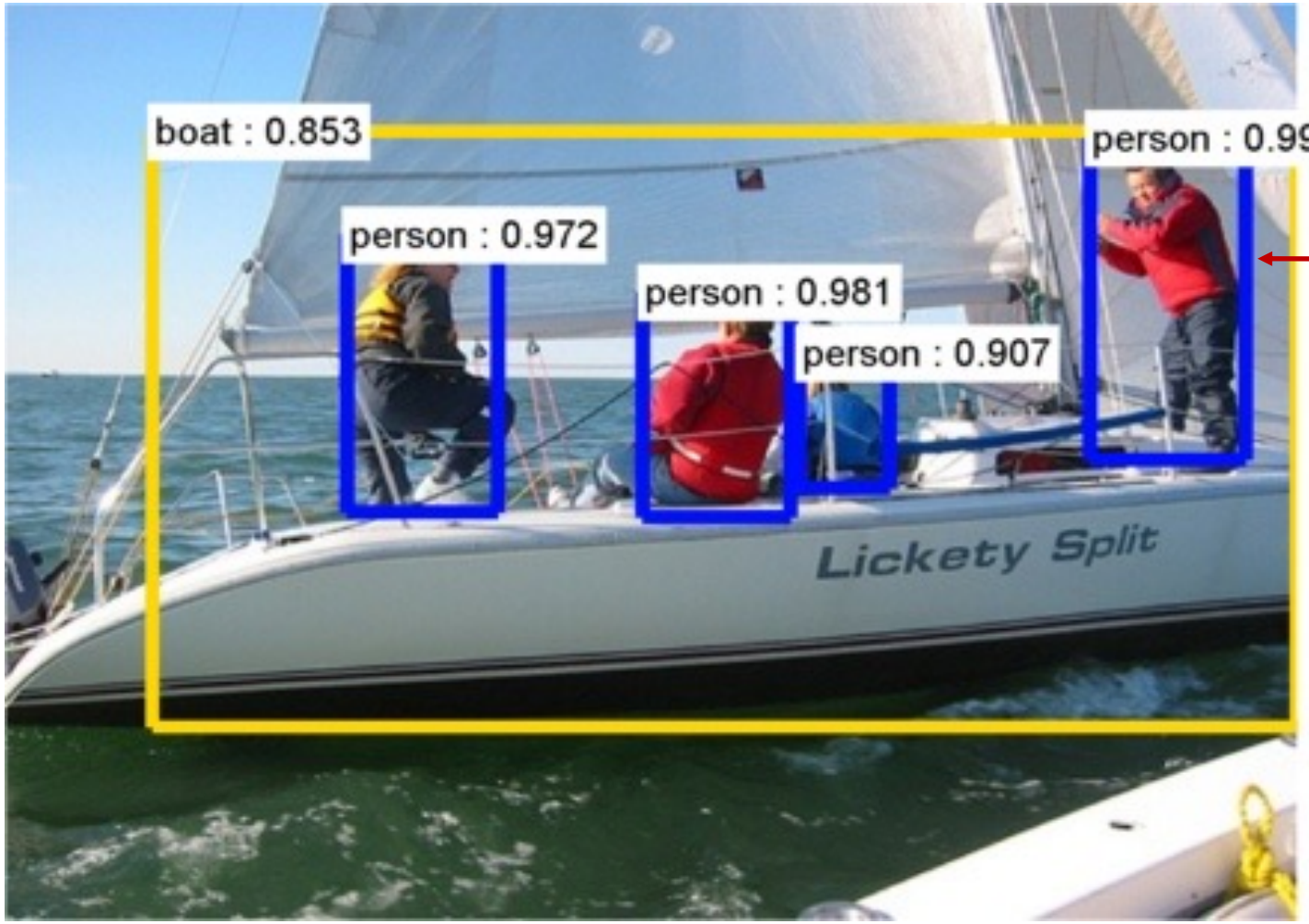
Slides Adapted from Ross Girshick's ICCV 2019 Tutorial on Visual Recognition for Images, Video, and 3D

Overview

Topics to cover

- Object detection intro
- The Generalized R-CNN framework
  - **R-CNN**
  - Fast R-CNN
  - Faster R-CNN
  - **Mask R-CNN**
- Future works + Discussion

Object Detection with Bounding Boxes

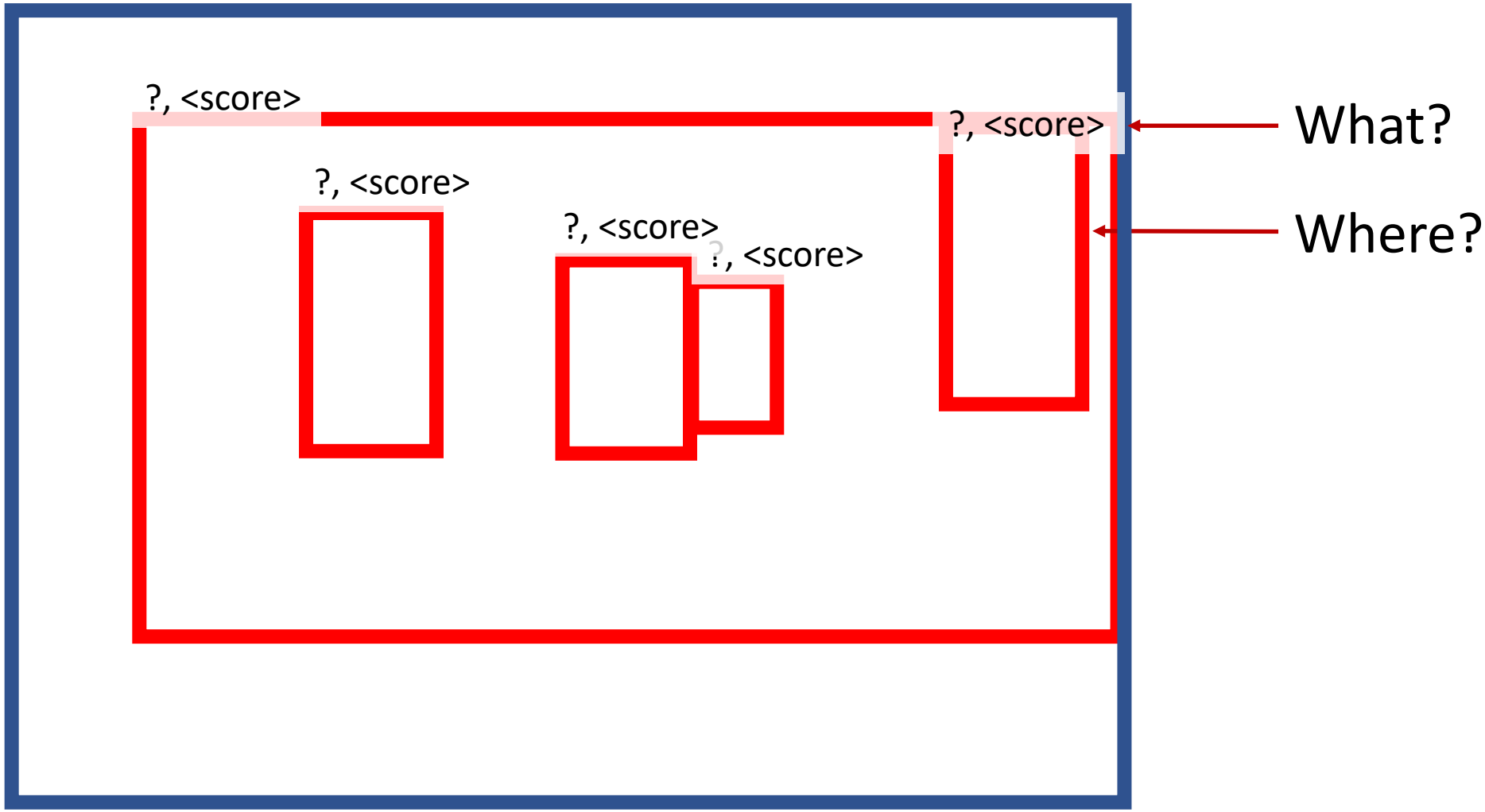


What?

Where?

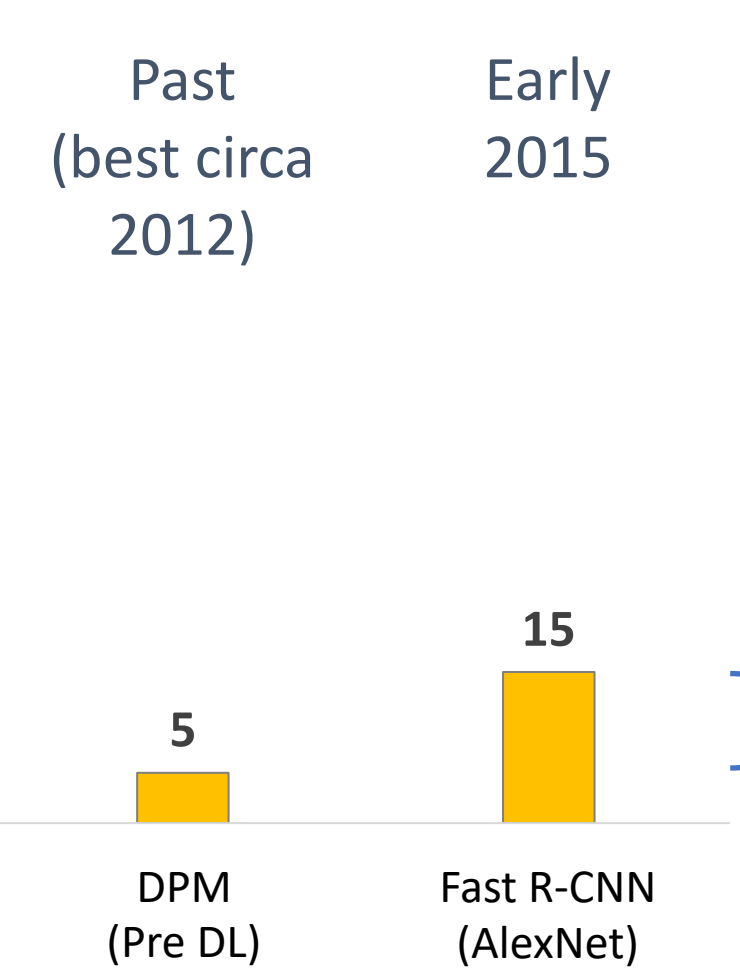
"Object detection"

Object Detection



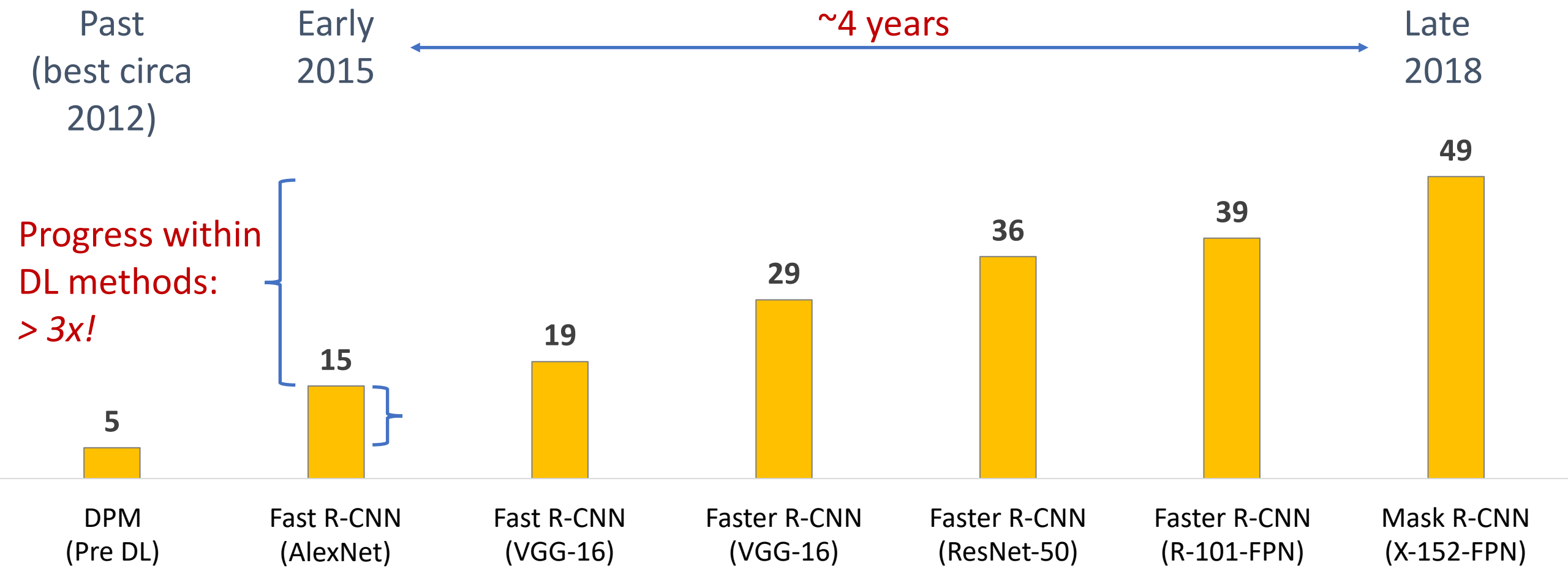
Modern Object Detection: Is this Picture Correct?

COCO Object Detection Average Precision (%)

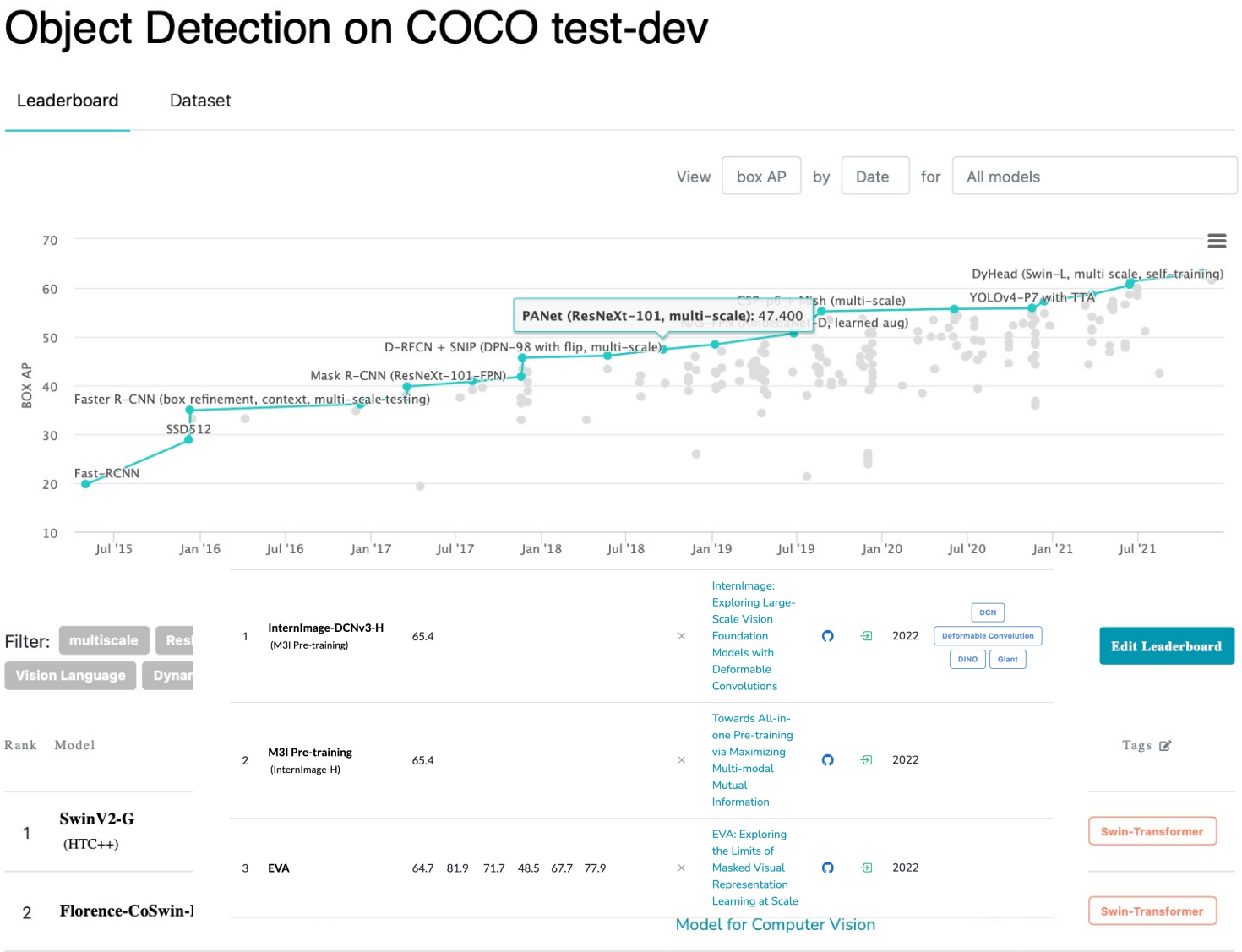


Movement to
Deep Learning methods:
3x improvement in AP

COCO Object Detection Average Precision (%)



COCO Object Detection Average Precision (%)



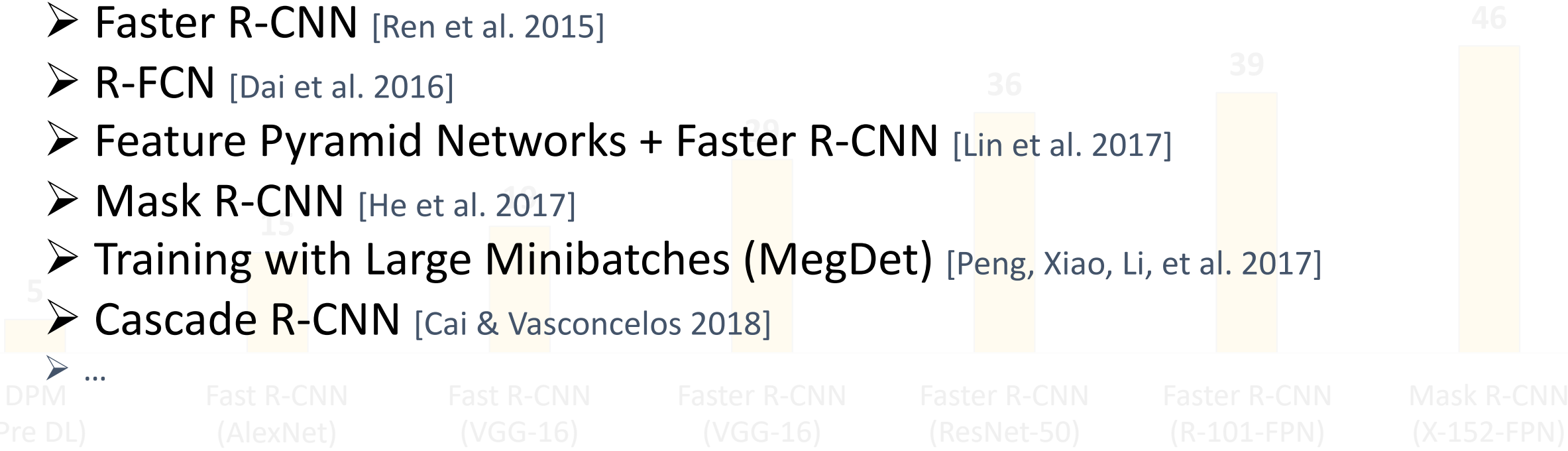
Progress within latest DL methods: *only 65.4%*

Past -> Present

Building
a better
hammer

Steady Progress on Boxes and Masks

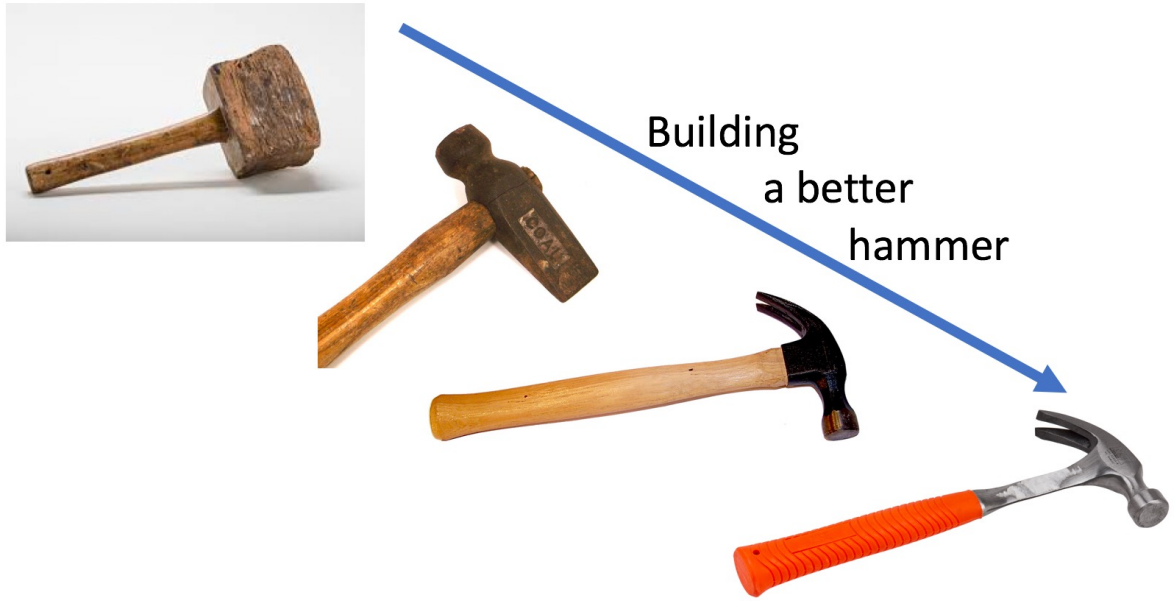
- R-CNN [Girshick et al. 2014]
- SPP-net [He et al. 2014]
- Fast R-CNN [Girshick. 2015]
- Faster R-CNN [Ren et al. 2015]
- R-FCN [Dai et al. 2016]
- Feature Pyramid Networks + Faster R-CNN [Lin et al. 2017]
- Mask R-CNN [He et al. 2017]
- Training with Large Minibatches (MegDet) [Peng, Xiao, Li, et al. 2017]
- Cascade R-CNN [Cai & Vasconcelos 2018]
- …



Overview

Topics to cover

- Object detection intro
- The Generalized R-CNN framework
  - **R-CNN**
  - Fast R-CNN
  - Faster R-CNN
  - **Mask R-CNN**
- Future works
  - **SwinV2**



Overview

Topics to cover

- Object detection intro
- The Generalized R-CNN framework
  - **R-CNN**
  - Fast R-CNN
  - Faster R-CNN
  - **Mask R-CNN**
- Future works

Rolling Pin

Vanilla Solution – Translate to Image Recognition

28 x 28

Hundreds of windows per image

https://github.com/layumi/2015_Face_Detection

Face Detection – Sliding Window is all your need

We return locations that the face classifier predicts high confidence.

https://github.com/layumi/2015_Face_Detection

Overview

Topics to cover

- Object detection intro
- The Generalized R-CNN framework
  - **R-CNN**
  - Fast R-CNN
  - Faster R-CNN
  - **Mask R-CNN**
- Future works

Old-school Hammer

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN (Region-based Convolutional Neural Net)

Per-image computation

I:

Per-region computation

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN (Region-based Convolutional Neural Net)

Per-image computation

I:

Per-region computation

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN (Region-based Convolutional Neural Net)

Per-image computation

I:

Per-region computation

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN (Region-based Convolutional Neural Net)

Per-image computation

Per-region computation

I:

Selective search, Edge Boxes, MCG, …

1

1. Use an off-the-shelf *Region of Interest* (RoI) proposal algorithm (~2k proposals per image)

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

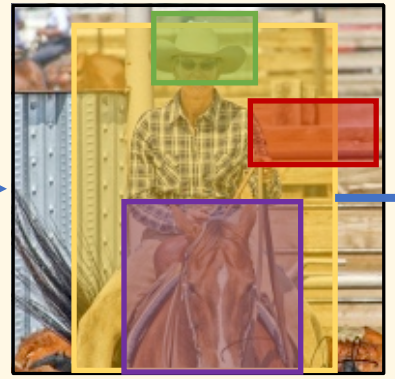
Per-image computation

Per-region computation for each $r_i \in r(I)$

I:

Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

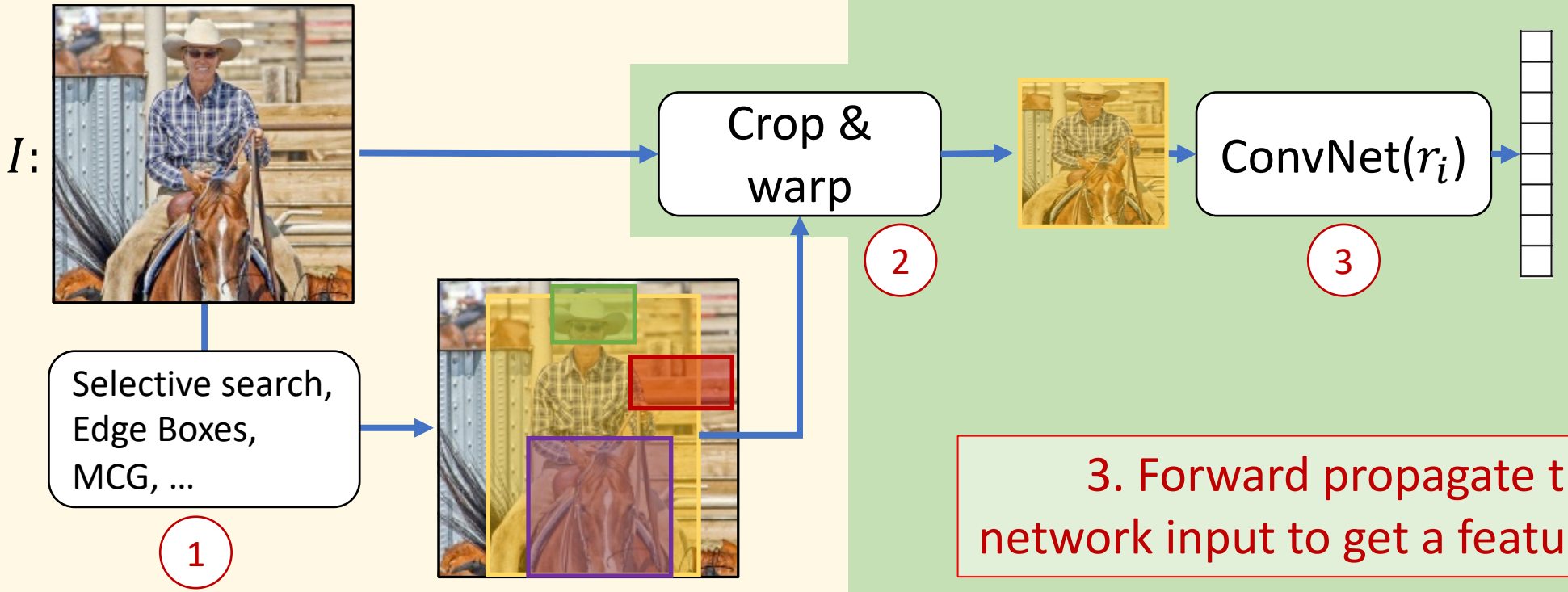
2. Crop and warp each proposal image window to obtain a fixed-size network input

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$



3. Forward propagate the fixed-size network input to get a feature representation

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

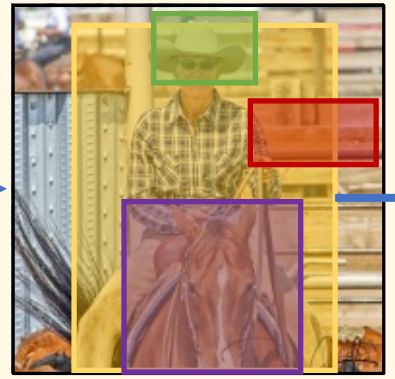
Per-image computation

Per-region computation for each $r_i \in r(I)$

I:

Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

ConvNet(r_i)

3

Linear classifier

4

4. Object classification

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

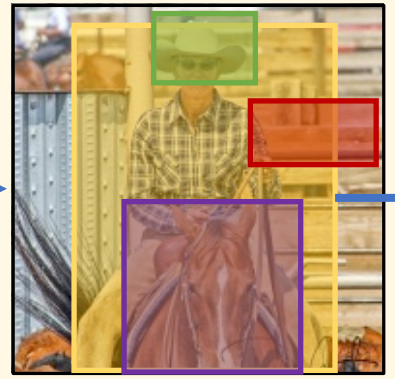
Per-image computation

Per-region computation for each $r_i \in r(I)$

I:

Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

ConvNet(r_i)

3

Linear classifier

4

Box regressor

5

5. Refine proposal localization with bounding-box regression

Generalized R-CNN Framework

Per-image computation

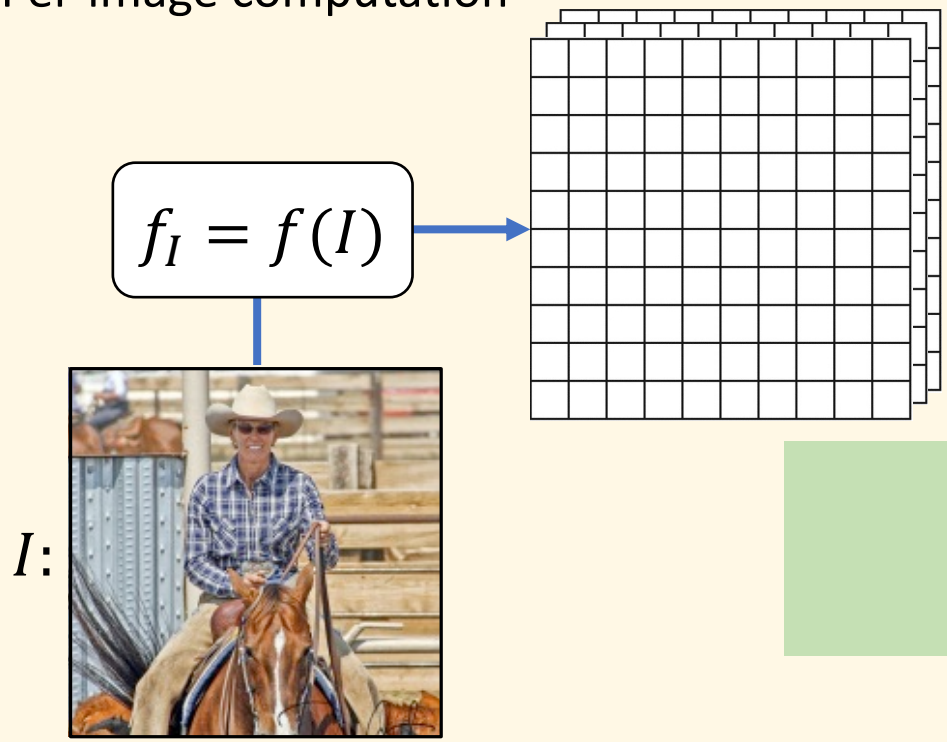
Per-region computation for each $r_i \in r(I)$

I:

Input image
per-image operations | per-region operations

Generalized R-CNN Framework

Per-image computation



Per-region computation for each $r_i \in r(I)$

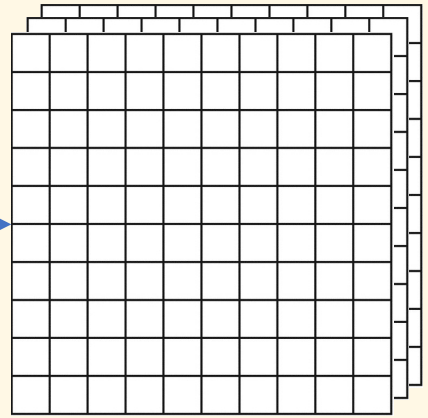
Transformation of the input image into a featurized representation

Generalized R-CNN Framework

Per-image computation

Per-region computation for each $r_i \in r(I)$

$f_I = f(I)$



I:

$r(I)$

Region of Interest proposals computed for the image

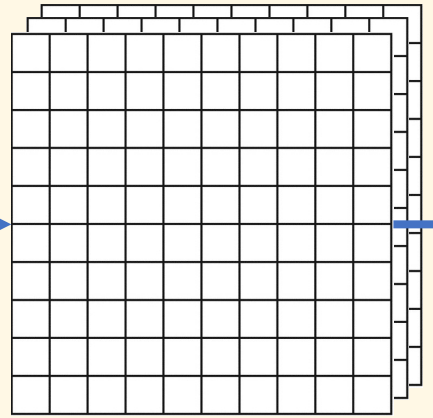
Generalized R-CNN Framework

Per-image computation

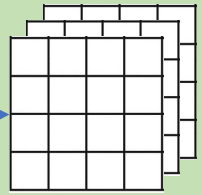
Per-region computation for each $r_i \in r(I)$

$f_I = f(I)$

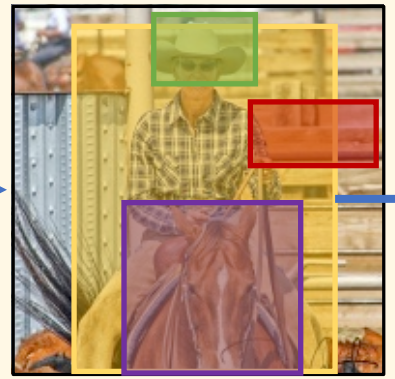
I:



$g_i = g(f_I, r_i)$



$r(I)$



Compute a featurized representation of each proposal

Generalized R-CNN Framework

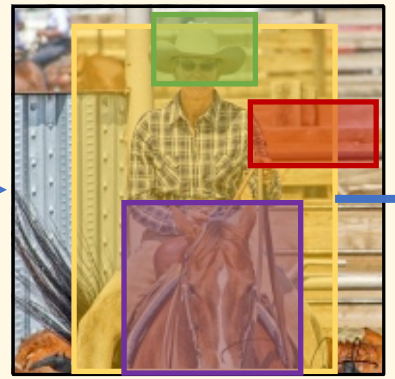
Per-image computation

Per-region computation for each $r_i \in r(I)$

$f_I = f(I)$

I:

$r(I)$



$g_i = g(f_I, r_i)$

$h(g_i)$

Apply additional processing to each proposal feature

Generalized R-CNN Framework

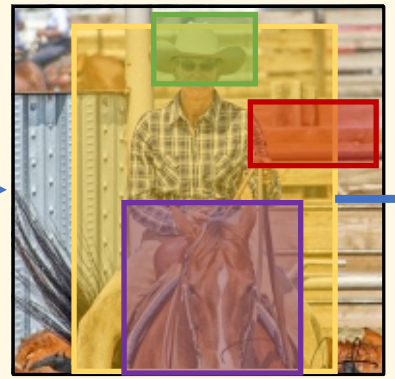
Per-image computation

Per-region computation for each $r_i \in r(I)$

$f_I = f(I)$

I:

$r(I)$



$g_i = g(f_I, r_i)$

$h(g_i)$

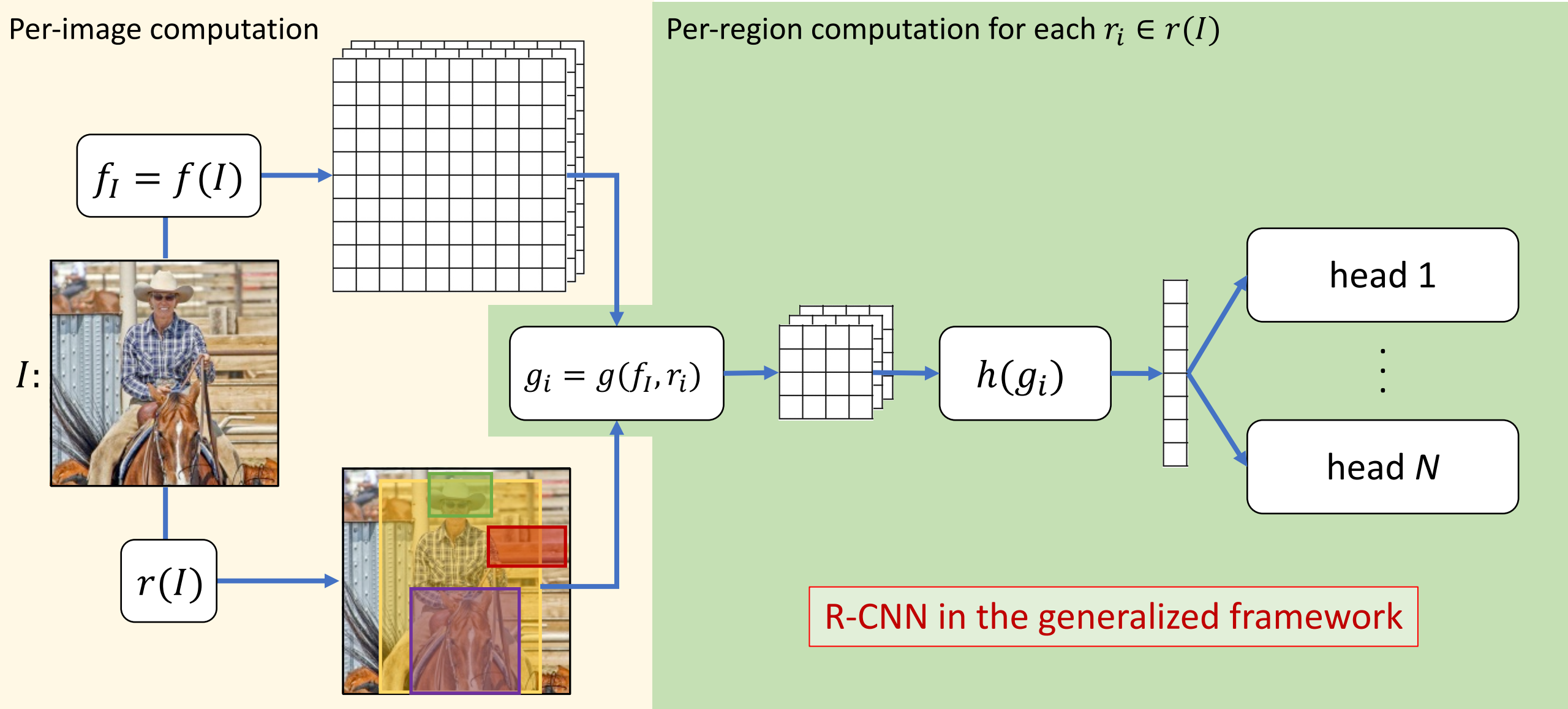
head 1

head *N*

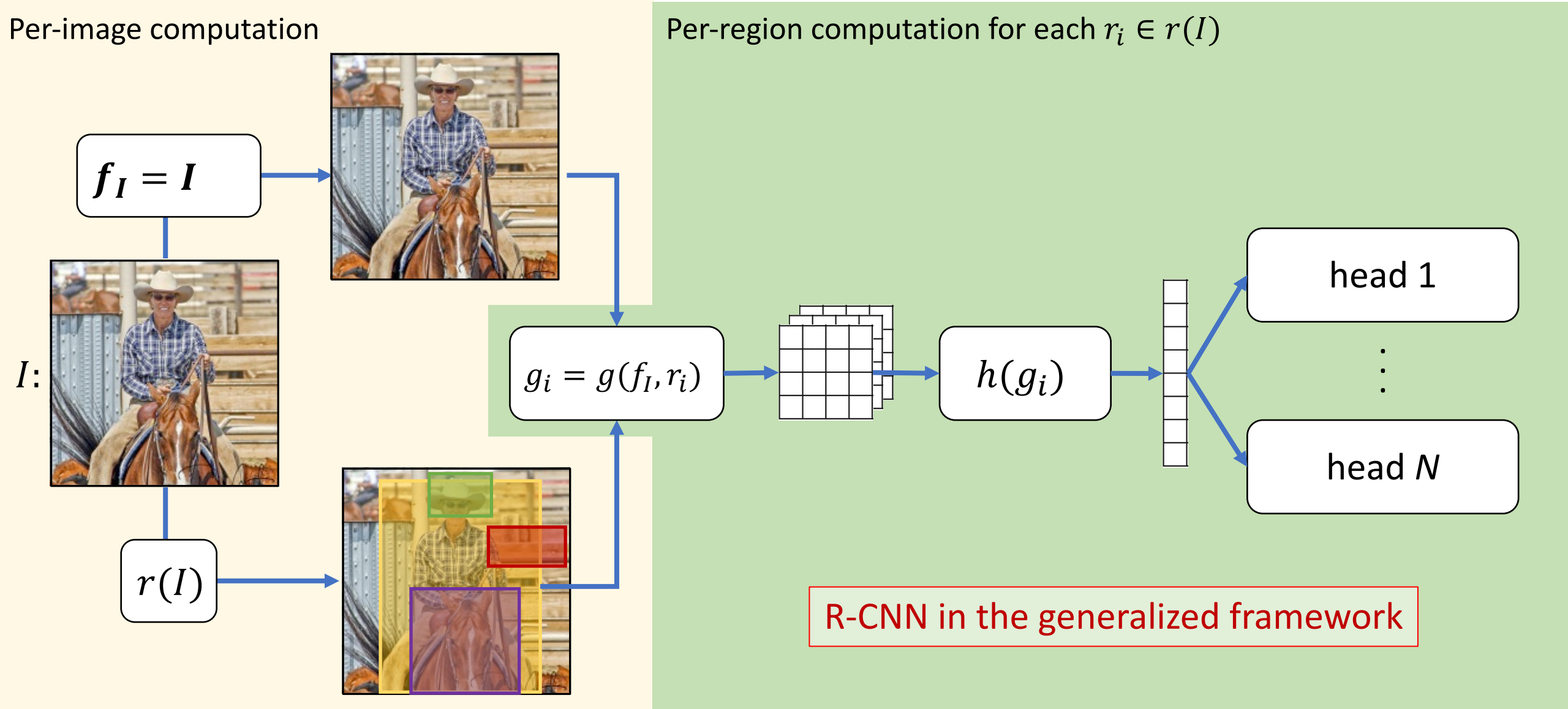
Apply multiple ‘heads’
to make task-specific predictions

What Does R-CNN Look Like in the Generalized Framework?

R-CNN in the Generalized Framework



R-CNN in the Generalized Framework



R-CNN in the Generalized Framework

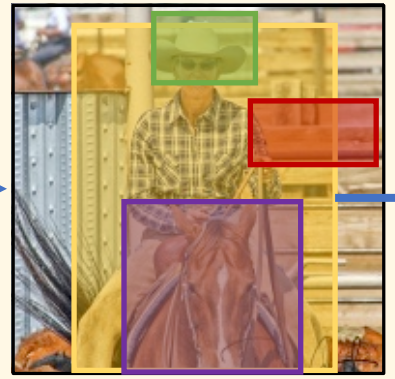
Per-image computation

Per-region computation for each $r_i \in r(I)$

$f_I = I$

I:

$\boldsymbol{r(I)} =$
Selective search,
Edge Boxes,
MCG, …



Crop & warp

$h(g_i)$

head 1

⋮

head *N*

R-CNN in the generalized framework

R-CNN in the Generalized Framework

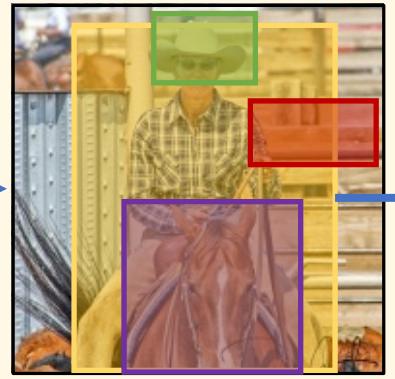
Per-image computation

Per-region computation for each $r_i \in r(I)$

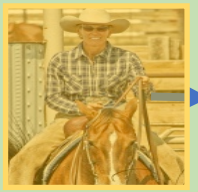
$f_I = I$

I:

$r(I) =$
Selective search,
Edge Boxes,
MCG, …



Crop & warp



ConvNet(r_i)

head 1

⋮

head *N*

R-CNN in the generalized framework

R-CNN in the Generalized Framework

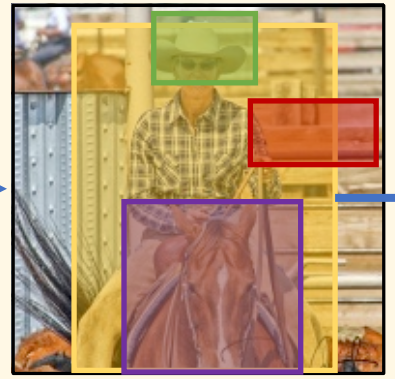
Per-image computation

Per-region computation for each $r_i \in r(I)$

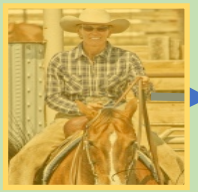
$f_I = I$

I:

$r(I) =$
Selective search,
Edge Boxes,
MCG, …



Crop & warp



ConvNet(r_i)

Linear classifier

Box regressor

R-CNN in the generalized framework

The Problem with R-CNN

R-CNN

Building a better hammer

?

"Slow" R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

$f_I = I$

I:

Selective search, Edge Boxes, MCG, …

Crop & warp

ConvNet(r_i)

Linear classifier

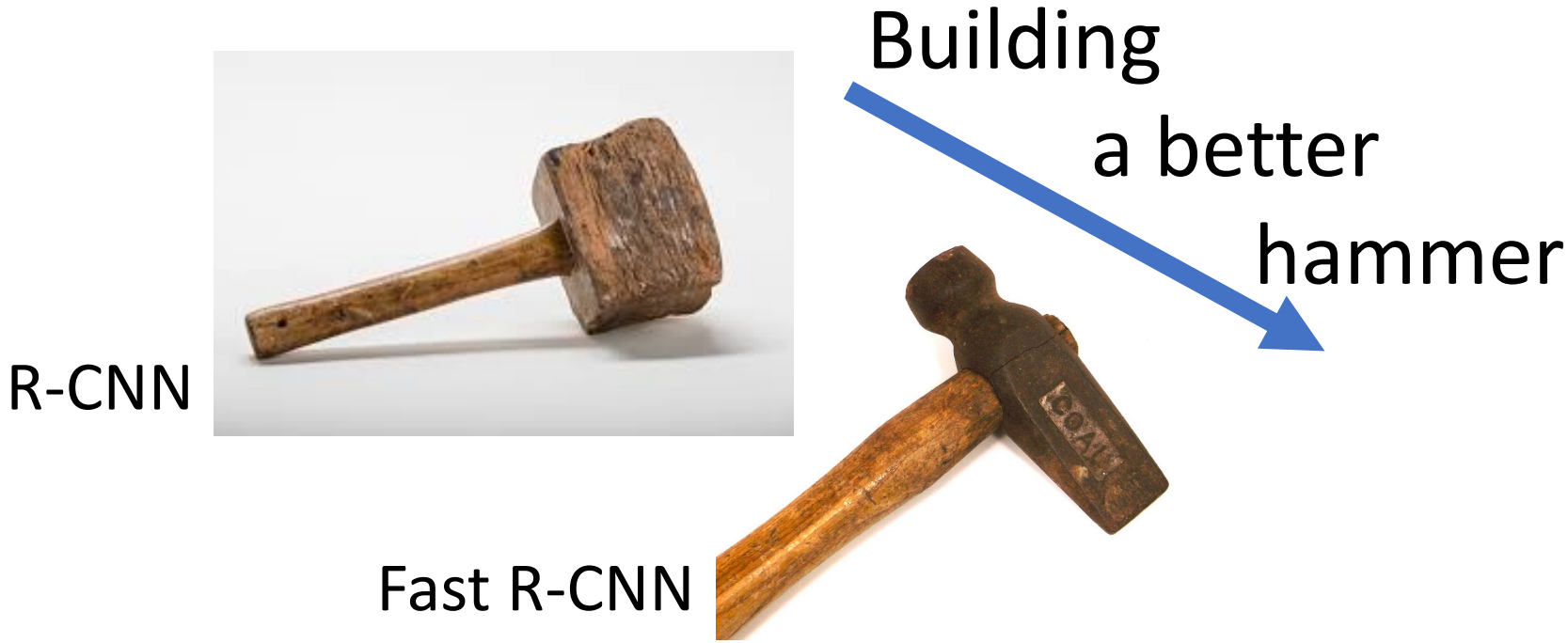
Box regressor

Very heavy *per-region* computation
E.g., 2000 full network evaluations

Fast R-CNN

References

- K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.
- R. Girshick. Fast R-CNN. In ICCV, 2015.



Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

$f_I = f(I)$

I:

Selective search,
Edge Boxes,
MCG, …

$g_i = g(f_I, r_i)$

$h(g_i)$

head 1

head *N*

Light-weight *per-region* computation
End-to-end trainable version of SPP-net [He et al. 2014]

Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search, Edge Boxes, MCG, …

$g_i = g(f_I, r_i)$

$h(g_i)$

head 1

head *N*

1. Fully convolutional network (FCN) maps the image to a lower resolution spatial feature map

Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

1 FCN(I)

I:

Selective search, Edge Boxes, MCG, …

2 RoIPool

$h(g_i)$

head 1

head *N*

2. *Region of interest* (RoI) pooling converts each region into a fixed dimensional representation

Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …

2

RoIPool

3

MLP

head 1

head *N*

3. A lightweight MLP processes each region feature map

(Alternatively, the MLP could be a small ConvNet, etc.)

Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …

2

RoIPool

3

MLP

Softmax clf.

4

Box regressor

4. Classification and box regression heads are applied

Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …

2

RoIPool

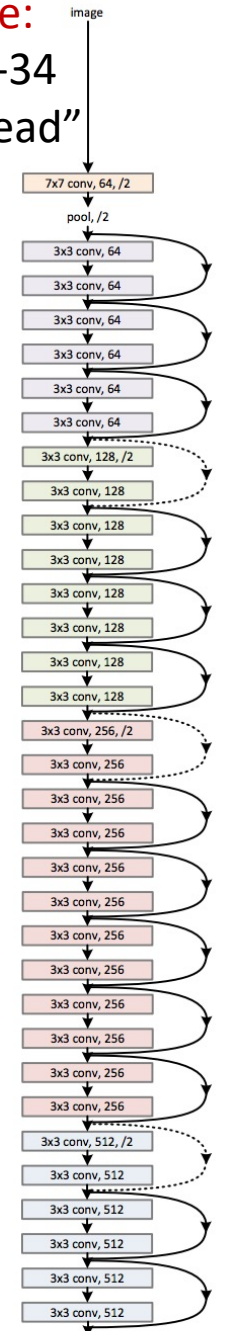
MLP

Softmax clf.

Box regressor

Whole-image, Fully Conv. Network (FCN)

Example:
ResNet-34
w/o “head”

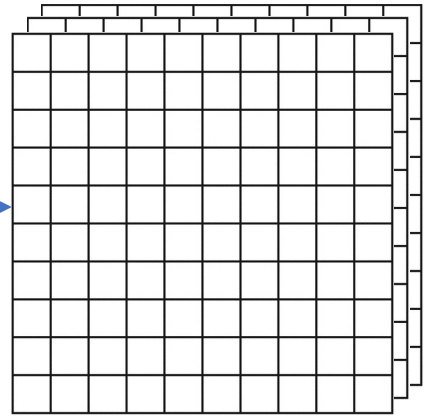


Use any standard ConvNet as the “backbone architecture”

- AlexNet, VGG, ResNet, Inception, Inception-ResNet, ResNeXt, DenseNet, NAS*, …
- Remove global pooling / FC
- Output spatial dims are proportional to input spatial dims

I:

FCN(I)



Example feature map dimensions:
(512, H/16, W/16)

Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …

2

RoIPool

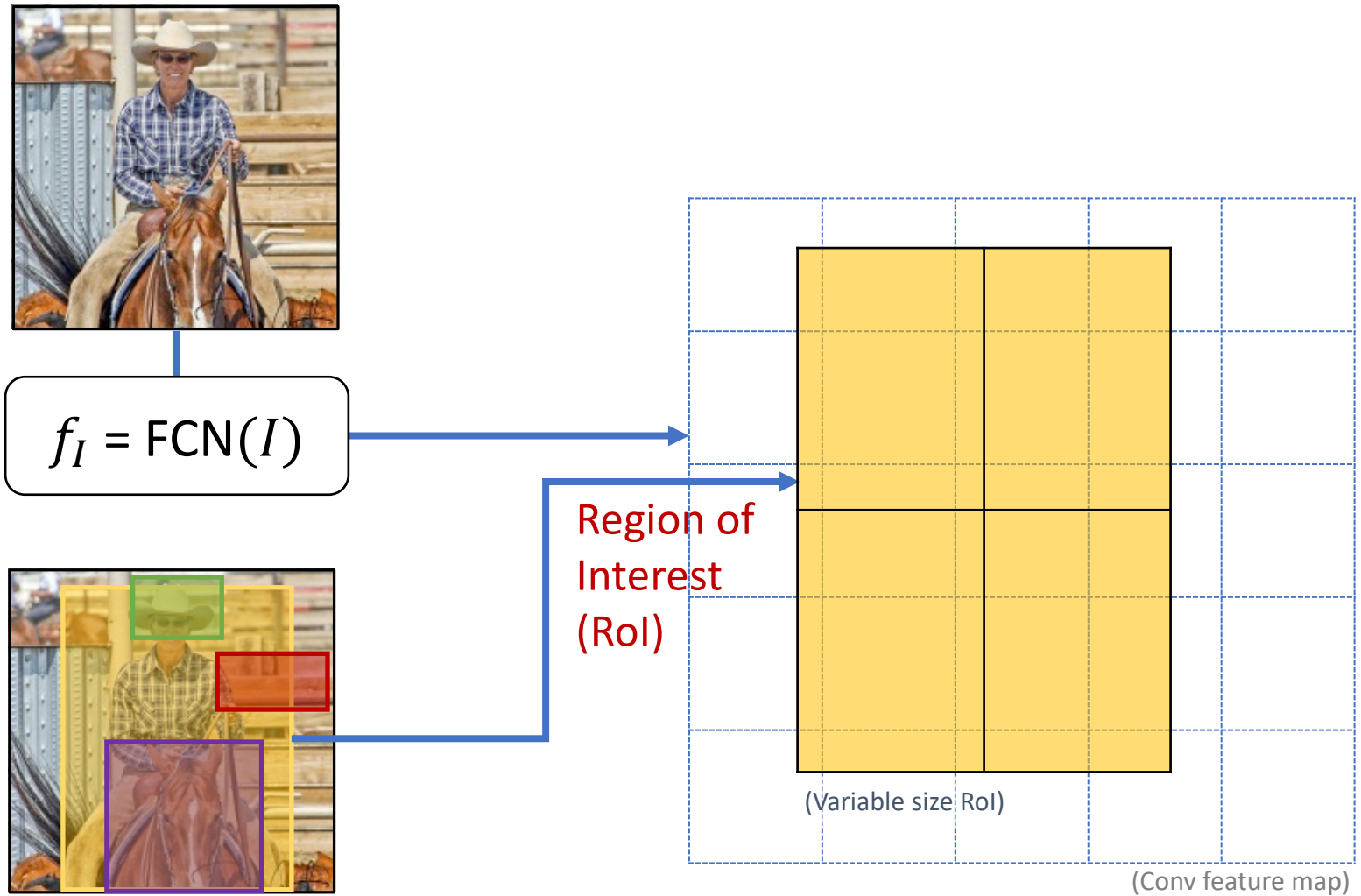
MLP

Softmax clf.

Box regressor

He, Zhang, Ren, Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

RoIPool Operation (on each Proposal)

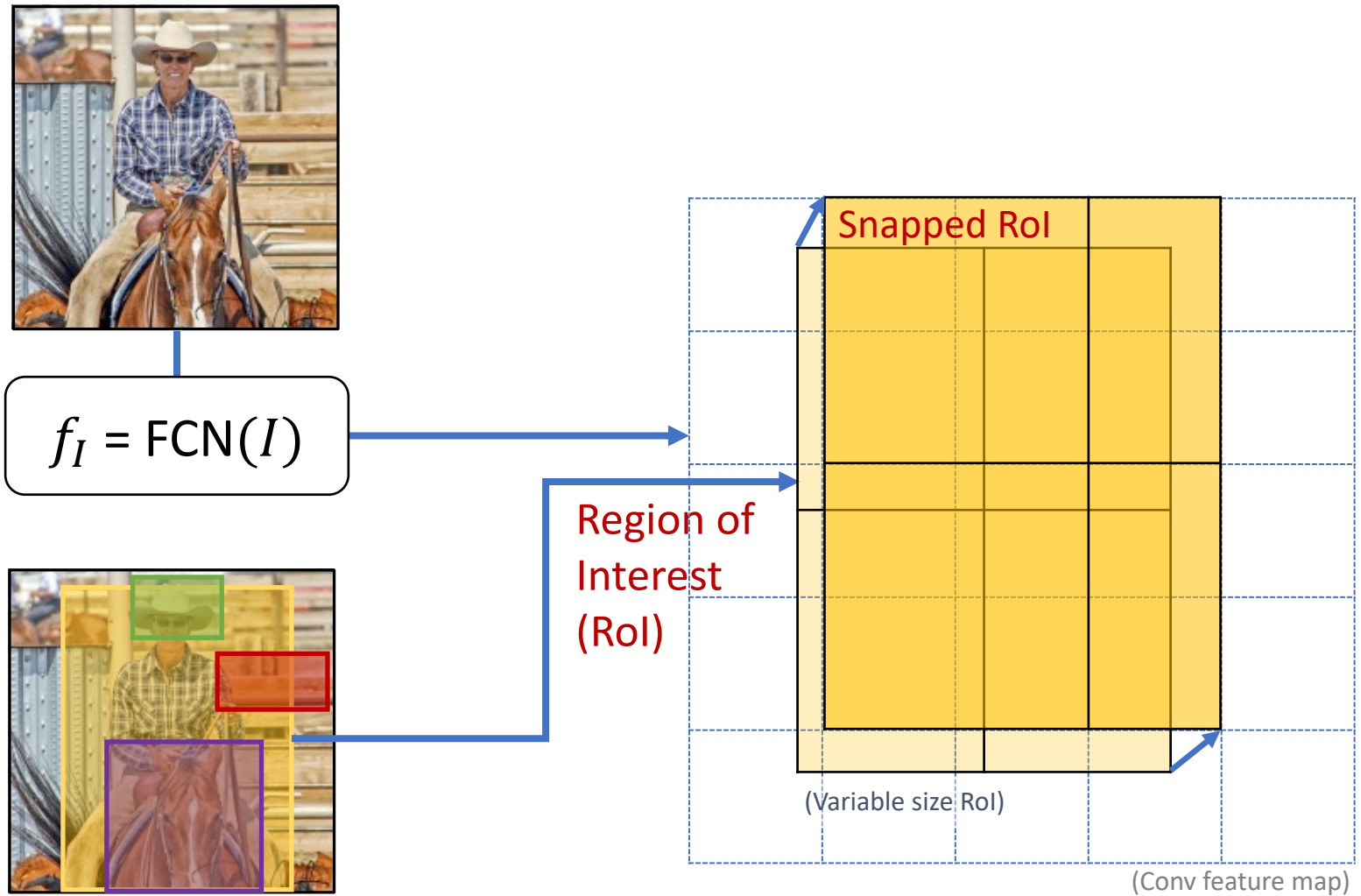


Key innovation in SPP-net

[He et al. 2014]

He, Zhang, Ren, Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

RoIPool Operation (on each Proposal)



Key innovation in SPP-net

[He et al. 2014]

He, Zhang, Ren, Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

RoIPool Operation (on each Proposal)

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., 2x2)

f_I = FCN(I)

Proposal Region of Interest (RoI)

Snapped RoI

(Variable size RoI)

(Conv feature map)

RoIPool transform

(Fixed dimensional representation)

MLP

Feature value is max over input cells

Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

	Fast R-CNN			R-CNN			SPPnet
	S	**M**	**L**	**S**	**M**	**L**	†**L**
train time (h)	**1.2**	2.0	9.5	22	28	84	25
train speedup	**18.3**×	14.0×	8.8×	1×	1×	1×	3.4×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0	2.3
▷ with SVD	**0.06**	0.08	0.22	-	-	-	-
test speedup	98×	80×	146×	1×	1×	1×	20×
▷ with SVD	169×	150×	**213**×	-	-	-	-
VOC07 mAP	57.1	59.2	**66.9**	58.5	60.2	66.0	63.1
▷ with SVD	56.5	58.7	66.6	-	-	-	-

The Problem with Fast R-CNN

R-CNN

Building a better hammer

Fast R-CNN

?

The Problem with Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …

RoIPool

MLP

Softmax clf.

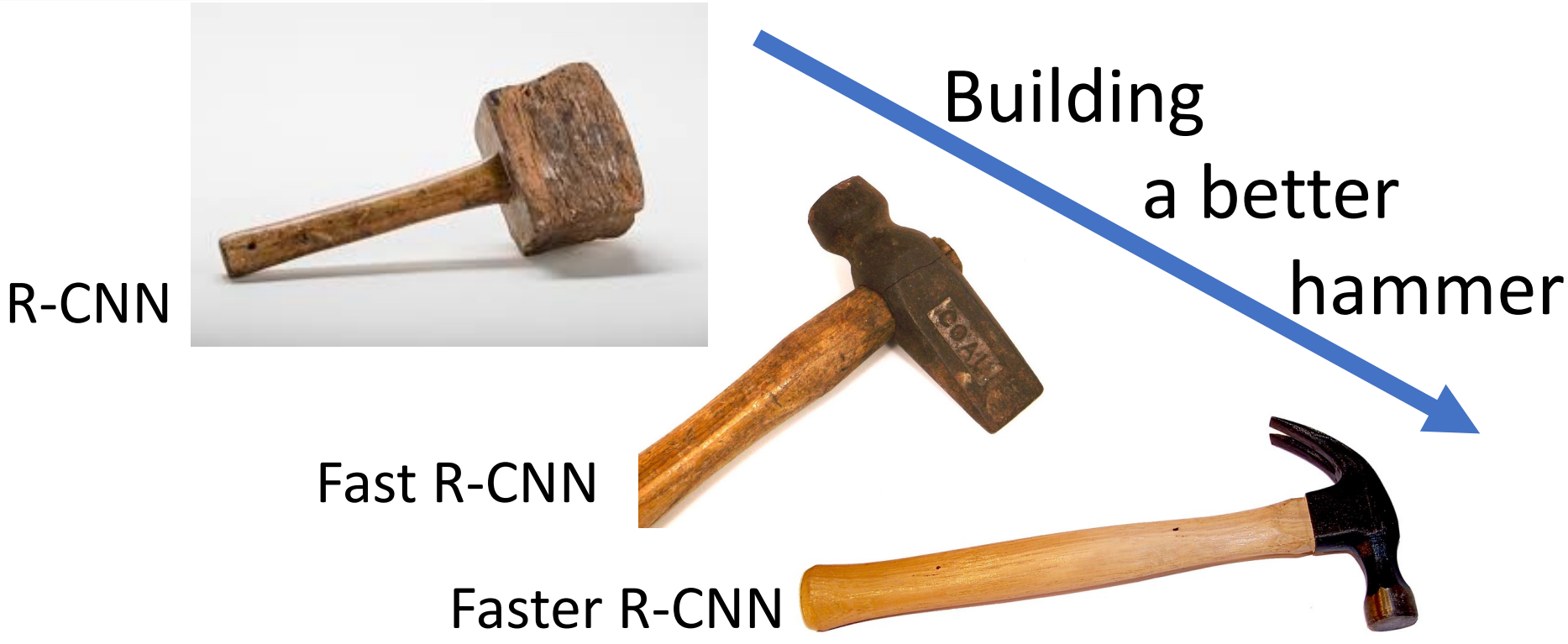
Box regressor

Region proposals have very poor recall
(ok for PASCAL VOC, major bottleneck for COCO)
Also, they can be slow

Faster R-CNN

References

- D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In CVPR, 2014.
- P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In NIPS, 2015.
- S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.



Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FCN(I)

I:

RPN(f_I)

RoIPool

MLP

Softmax clf.

Box regressor

Learned proposals
Shares computation with whole-image network

Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

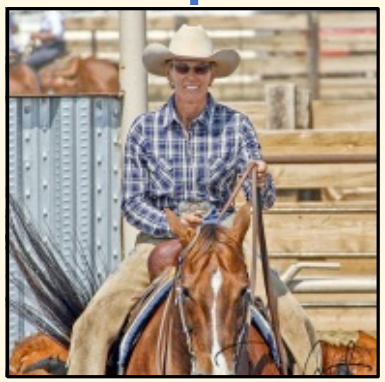
Faster R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FCN(I)

I:



RPN(f_I)

RoIPool

MLP

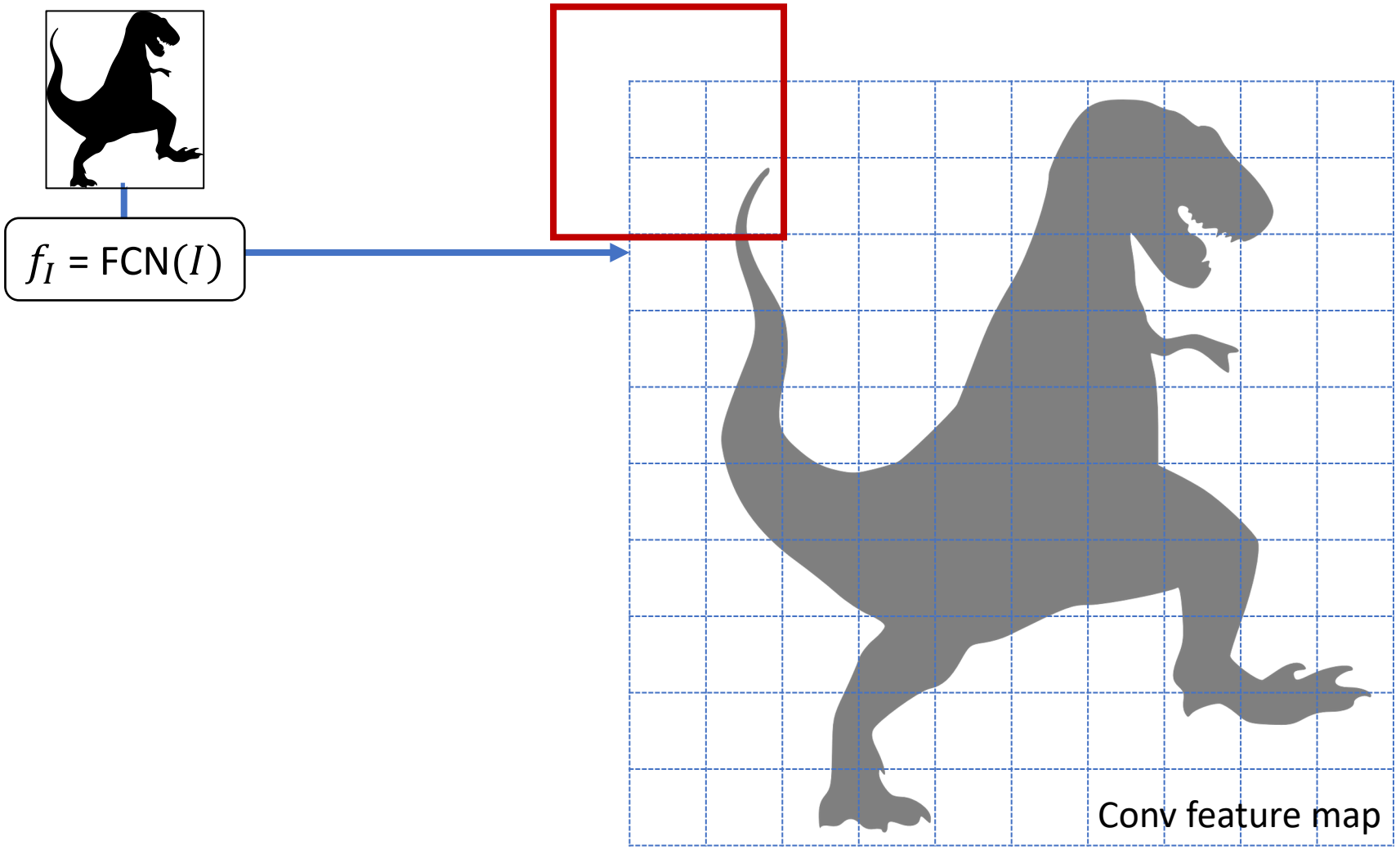
Softmax clf.

Box regressor

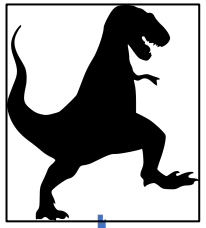
Learned proposals

Shares computation with whole-image network

RPN: Region Proposal Network



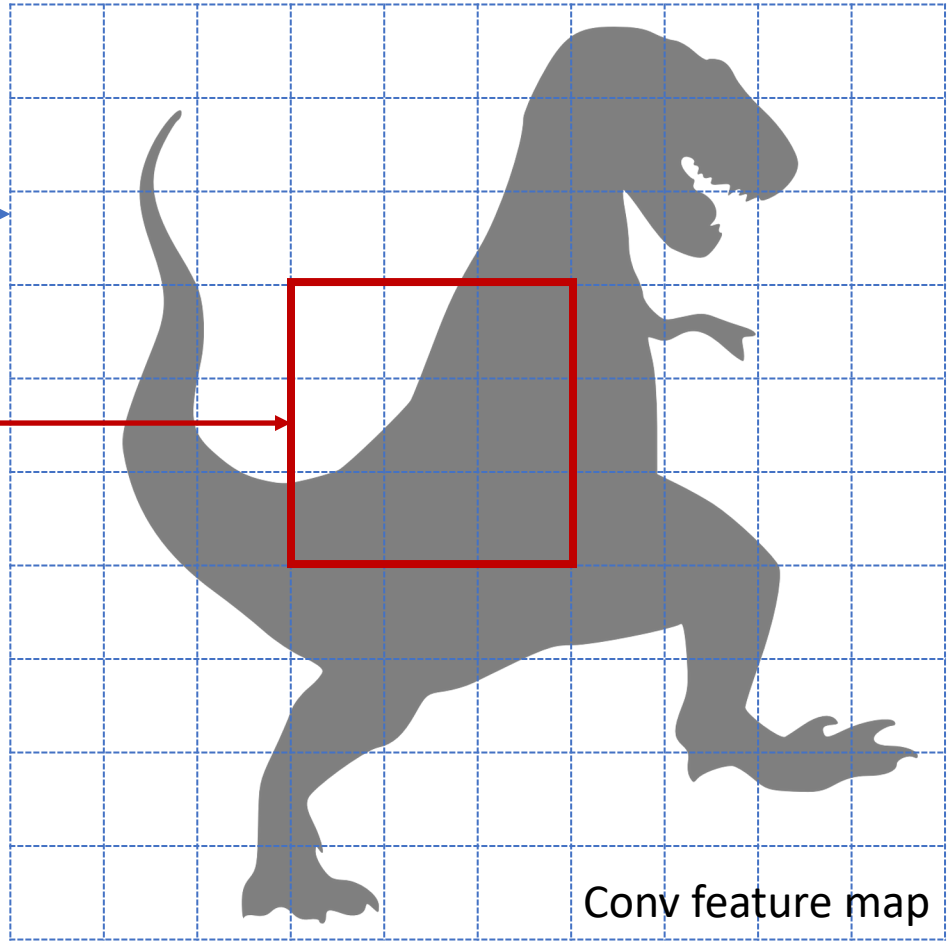
RPN: Region Proposal Network



f_I = FCN(I)

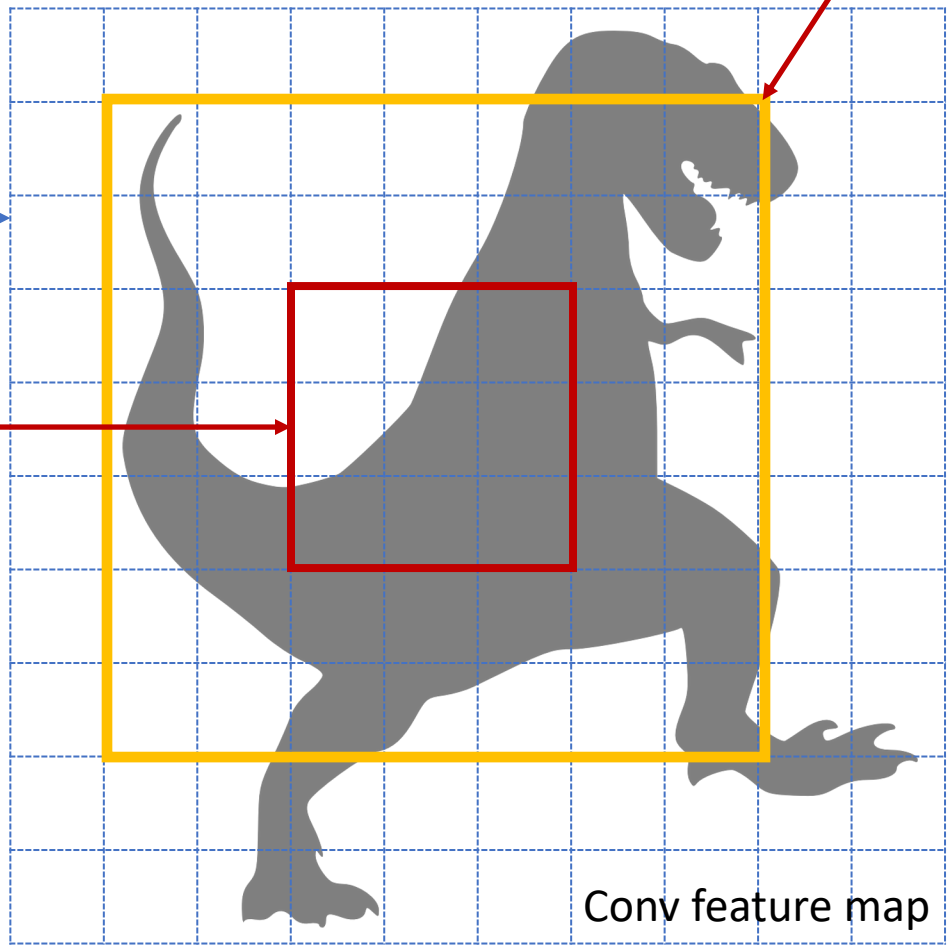
3x3 “sliding window”

Scans the feature map looking for objects



RPN: Anchor Box

f_I = FCN(I)



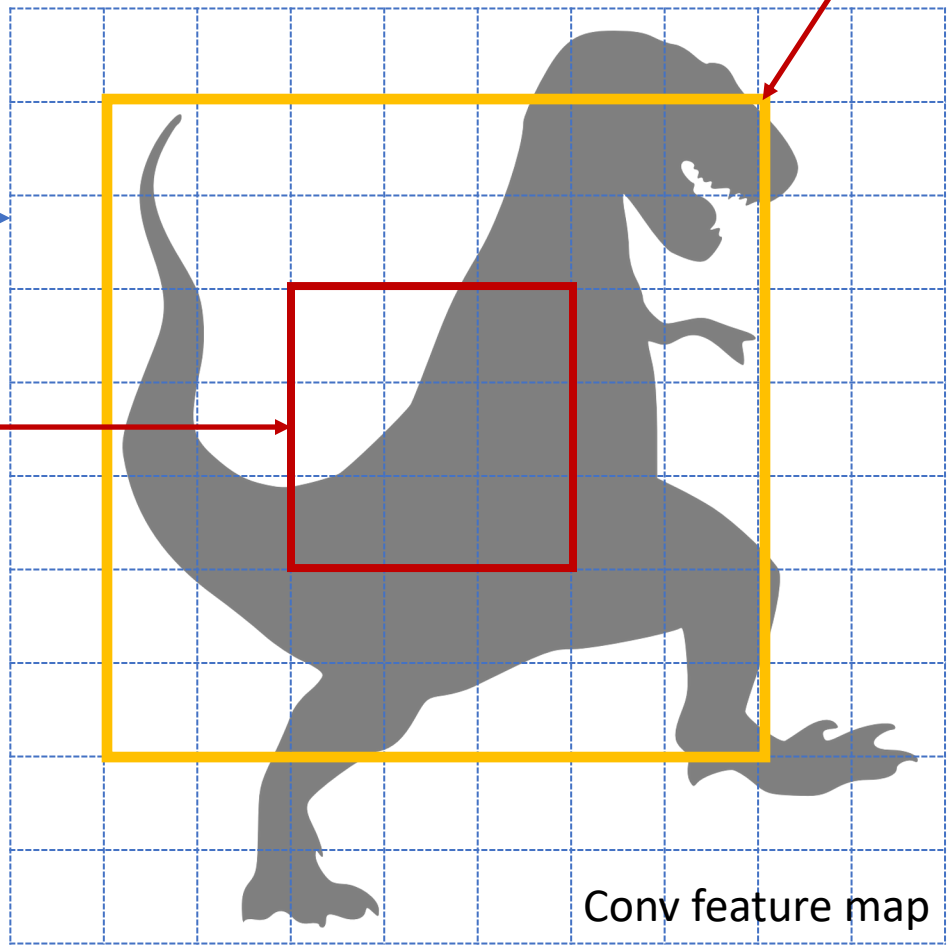
Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*

3x3 “sliding window”

Scans the feature map looking for objects

RPN: Anchor Box

f_I = FCN(I)



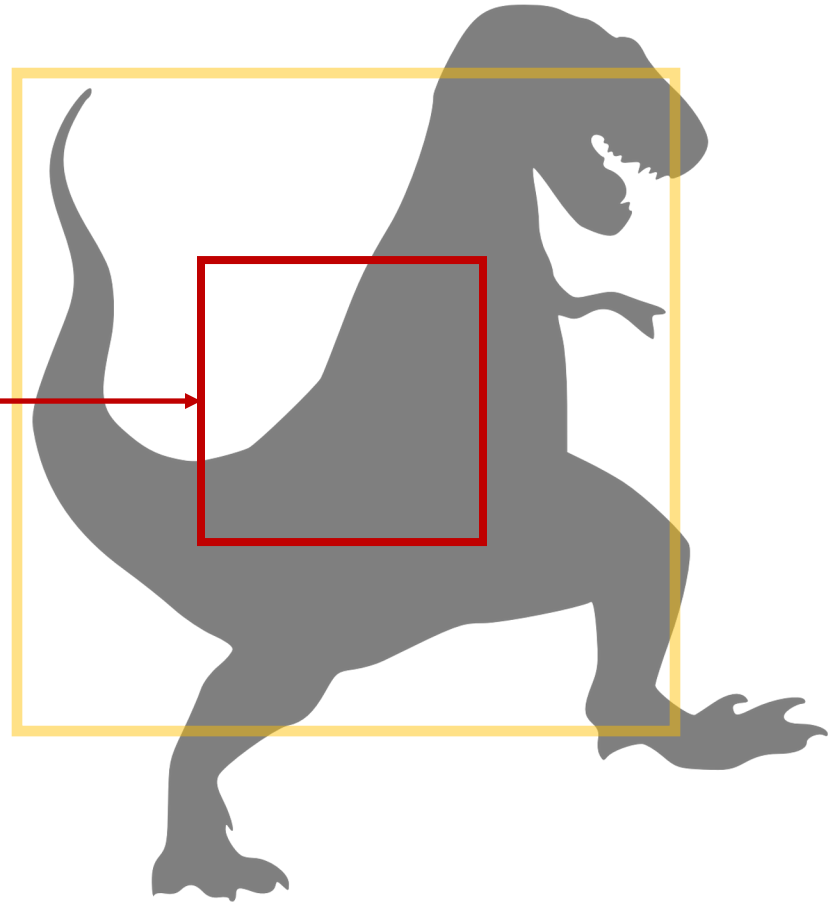
Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*

3x3 “sliding window”

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

RPN: Prediction (on object)

Objectness score → P(object) = 0.94



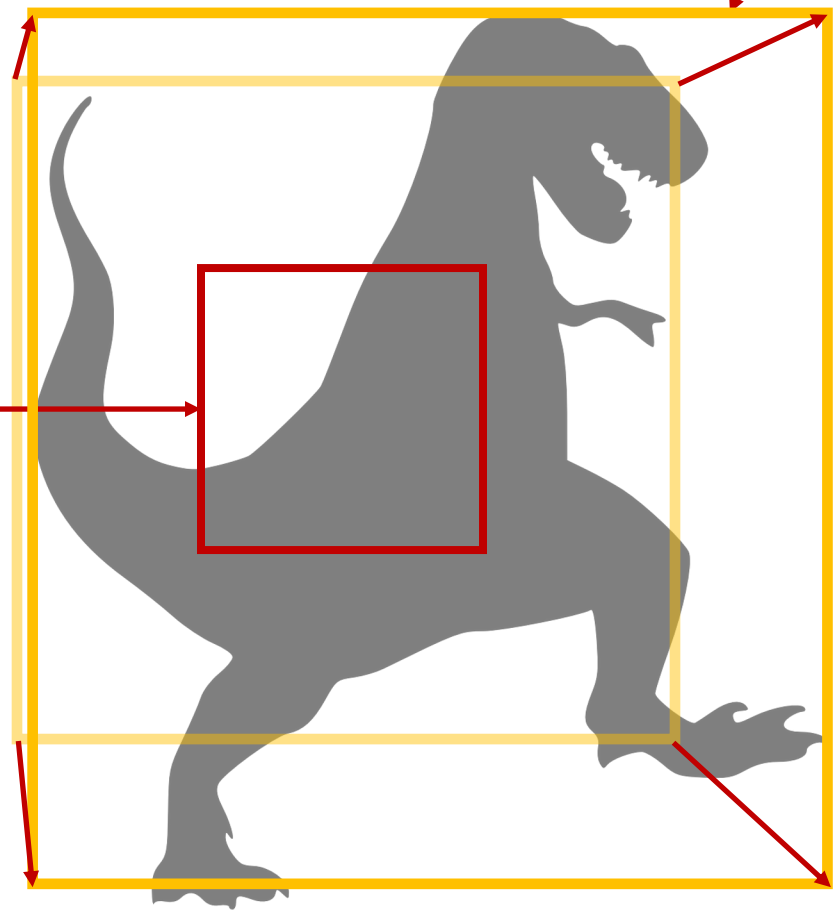
3x3 “sliding window” →

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

RPN: Prediction (on object)

Anchor box: transformed by box regressor

P(object) = 0.94



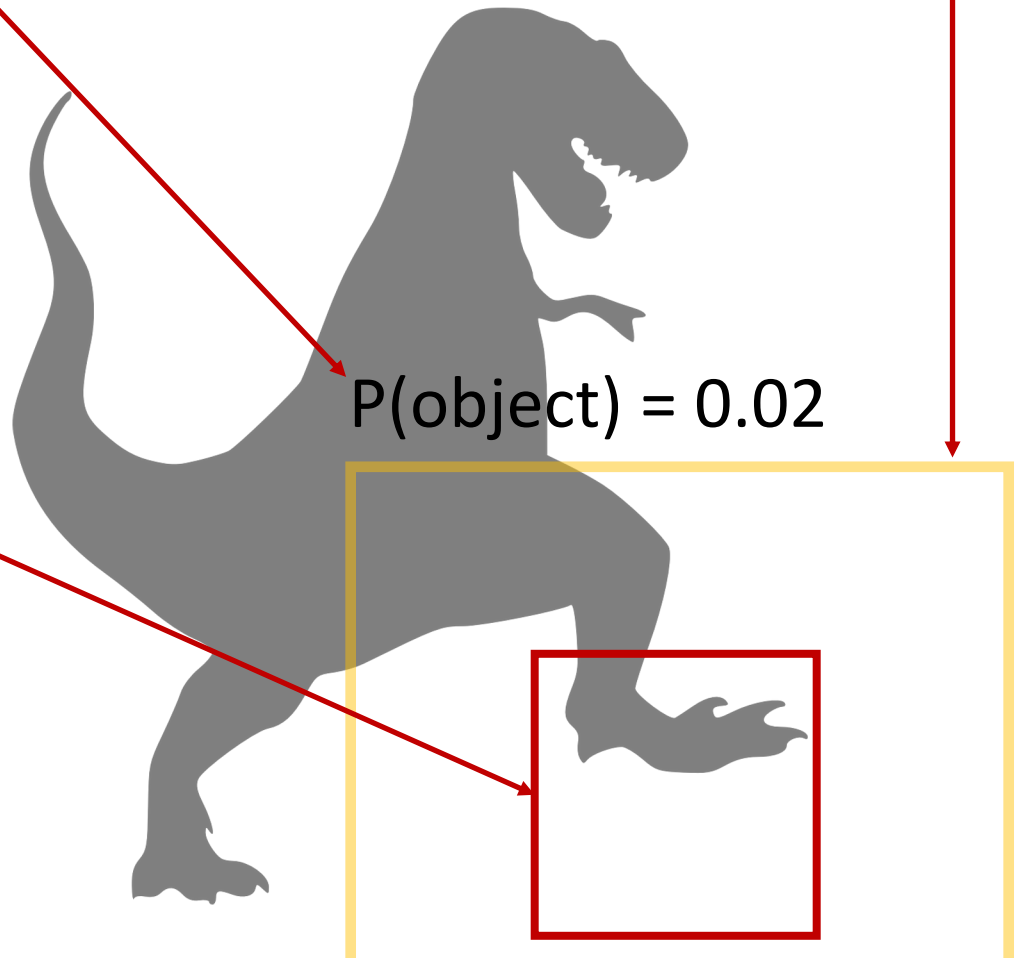
3x3 “sliding window”

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

RPN: Prediction (off object)

Objectness score

Anchor box: transformed by box regressor



3x3 “sliding window”

- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)

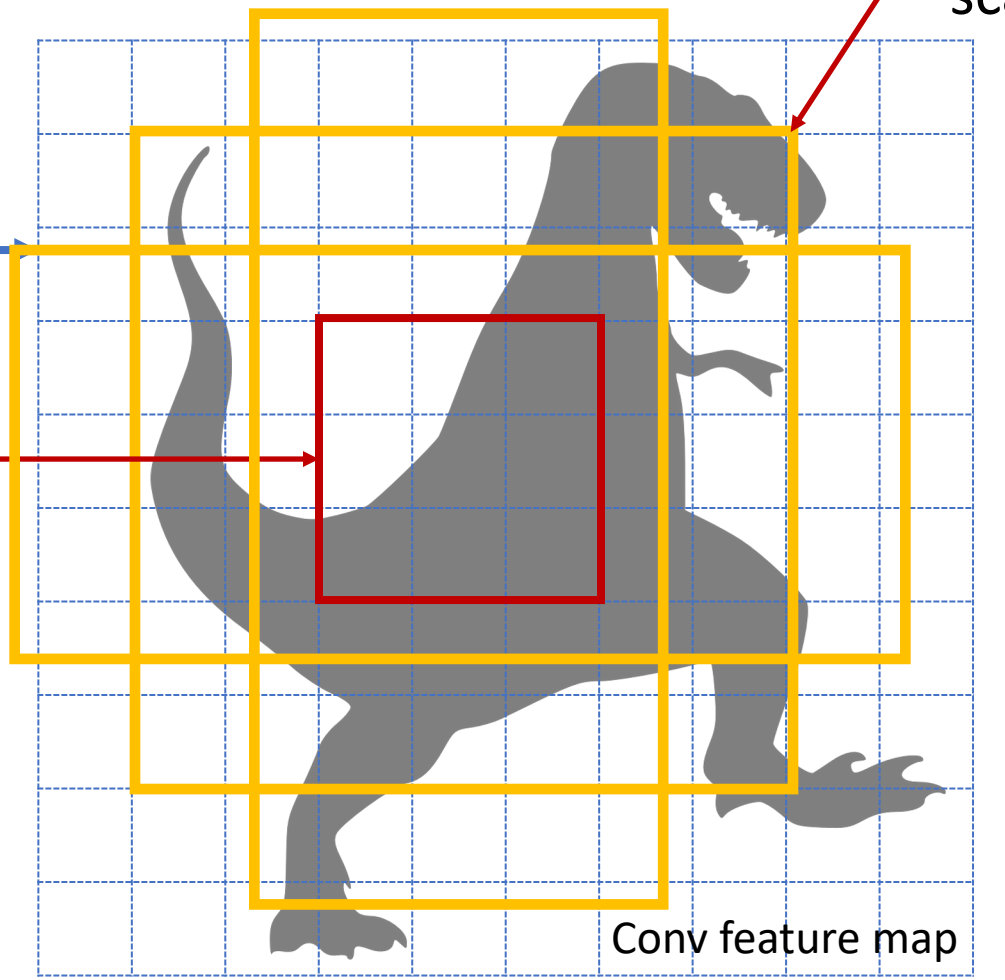
RPN: Multiple Anchors

f_I = FCN(I)

Anchor boxes: *K* anchors per location with different scales and aspect ratios

3x3 “sliding window”

- *K* objectness classifiers
- *K* box regressors



Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

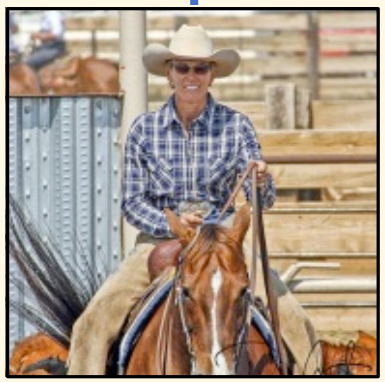
Faster R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FCN(I)

I:



RPN(f_I)

RoIPool

MLP

Softmax clf.

Box regressor

Learned proposals

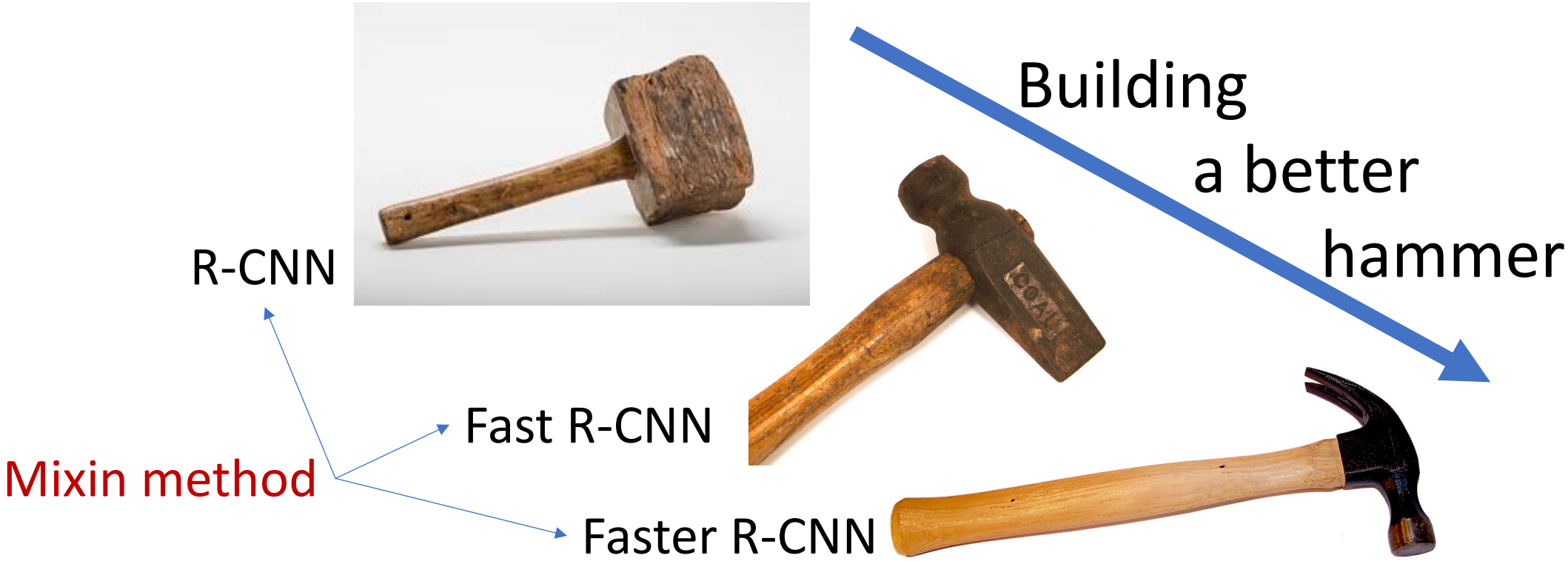
Shares computation with whole-image network

Faster R-CNN

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Improvements from "Mixin Methods"

- Largely orthogonal to the base detector
- Can be added to many different detectors as a "mixin"
  (A better backbone network is one example)



Faster R-CNN with a Feature Pyramid Network

Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FPN(I)

I:

RPN(f_I)

RoIPool

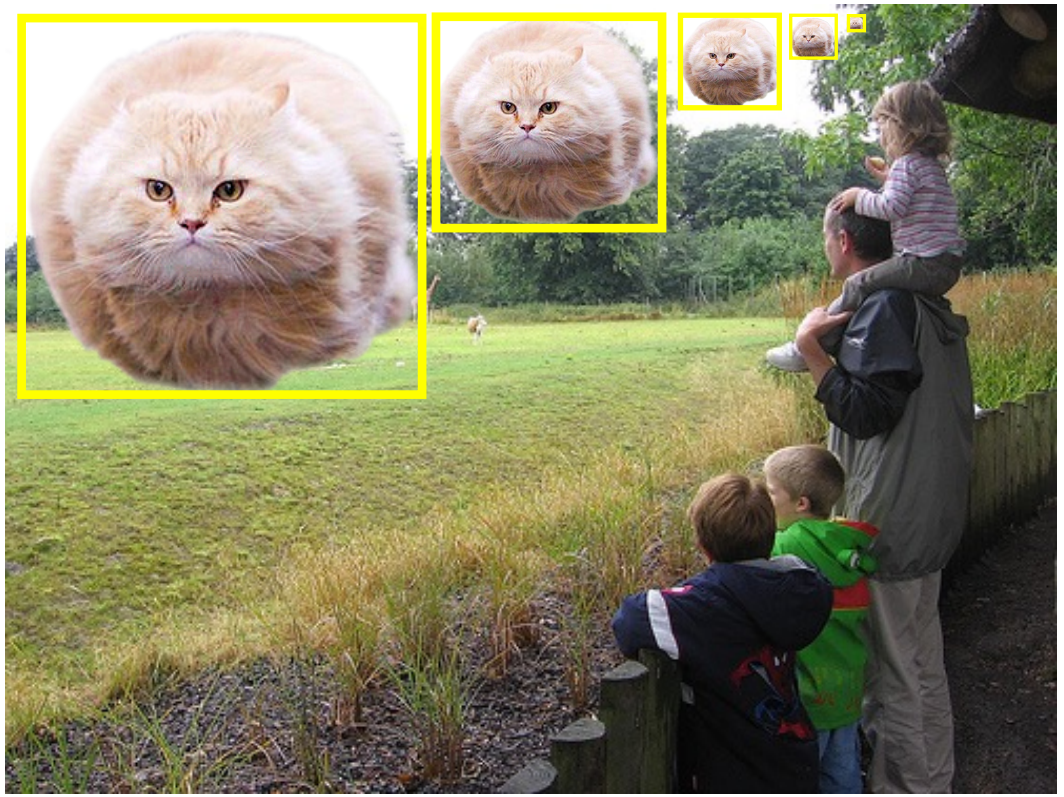
MLP

Softmax clf.

Box regressor

The whole-image feature representation can be improved by making it *multi-scale*

FPN: Improving Scale Invariance and Equivariance



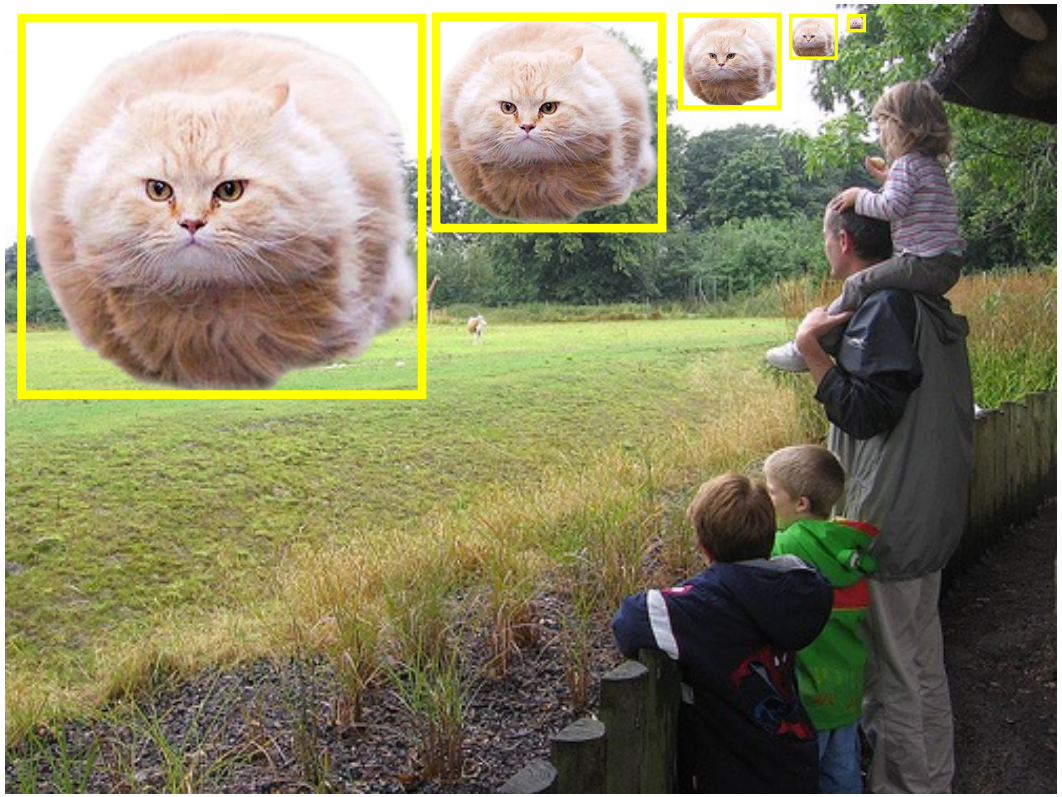
Detectors need to

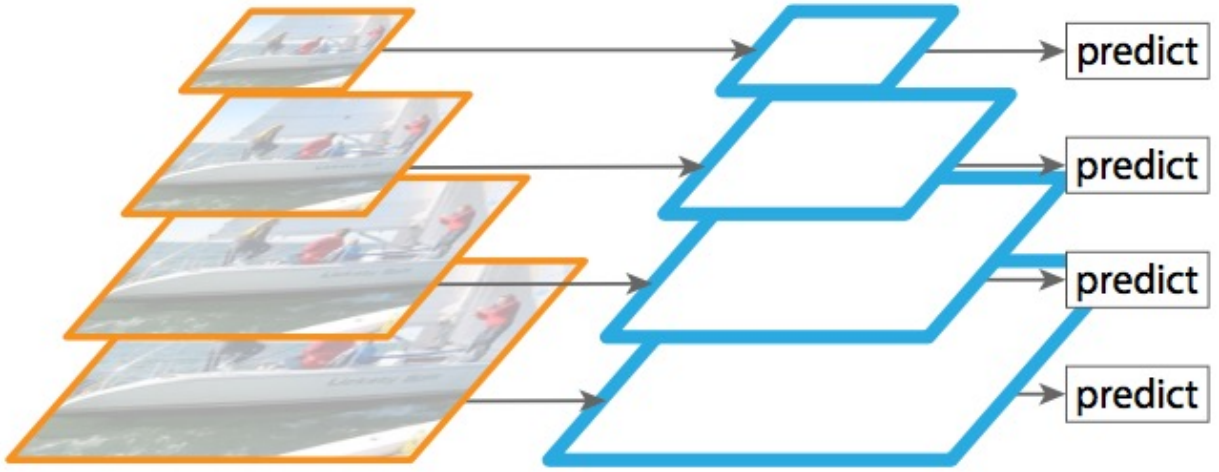
1. Classify (invariance) and
2. Localize (equivariance)

objects over a wide range of scales

FPN improves this ability

Strategy 1: Image Pyramid



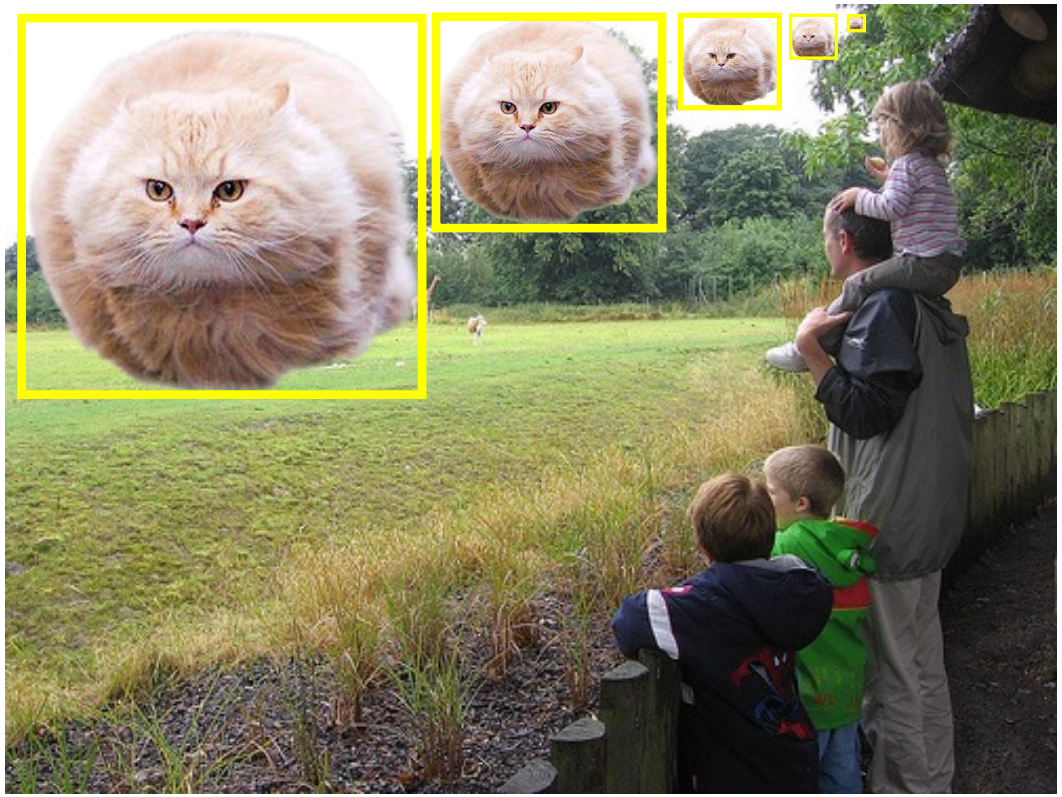


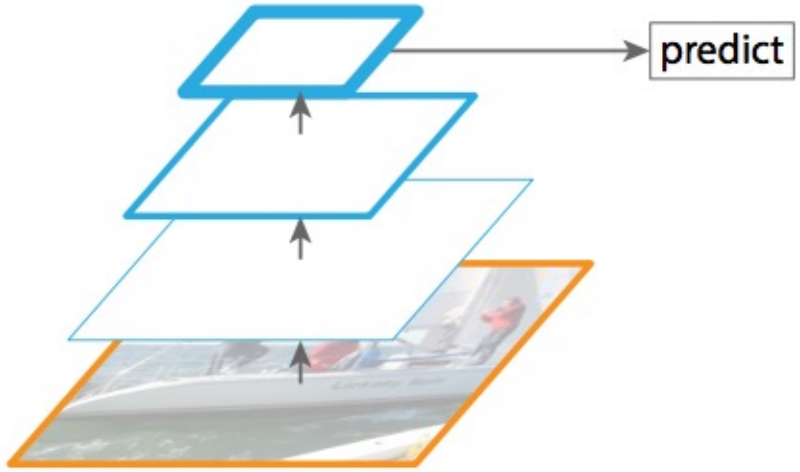
(a) Featurized image pyramid

Standard solution – *slow!*

(E.g., Viola & Jones, HOG, DPM, SPP-net, multi-scale Fast R-CNN, …)

Strategy 2: Multi-scale Features (Single-scale Map)



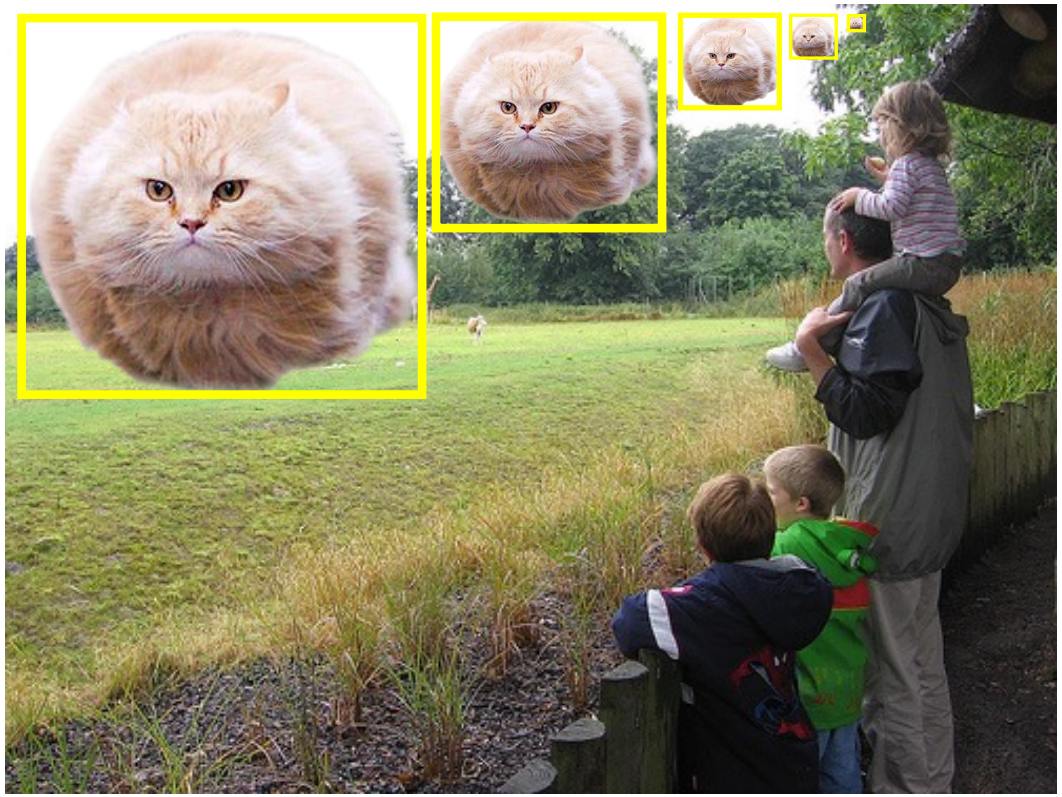


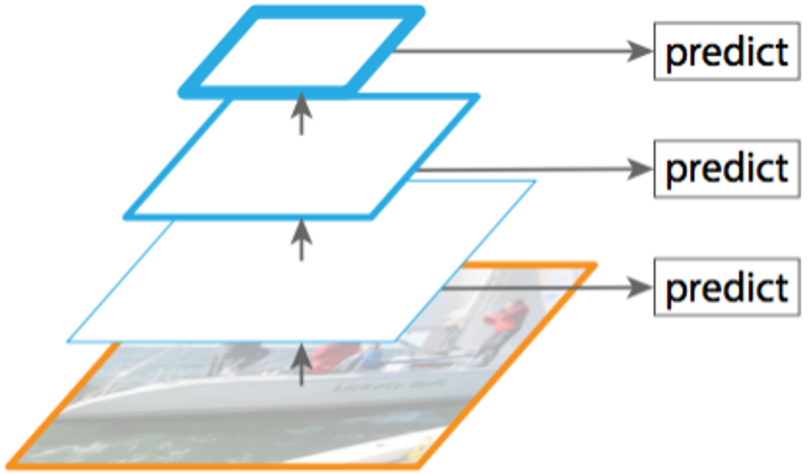
(b) Single feature map

Leave it all to the features – *fast, suboptimal*

(E.g., Fast/er R-CNN, YOLO, …)

Strategy 3: Naïve In-network Pyramid



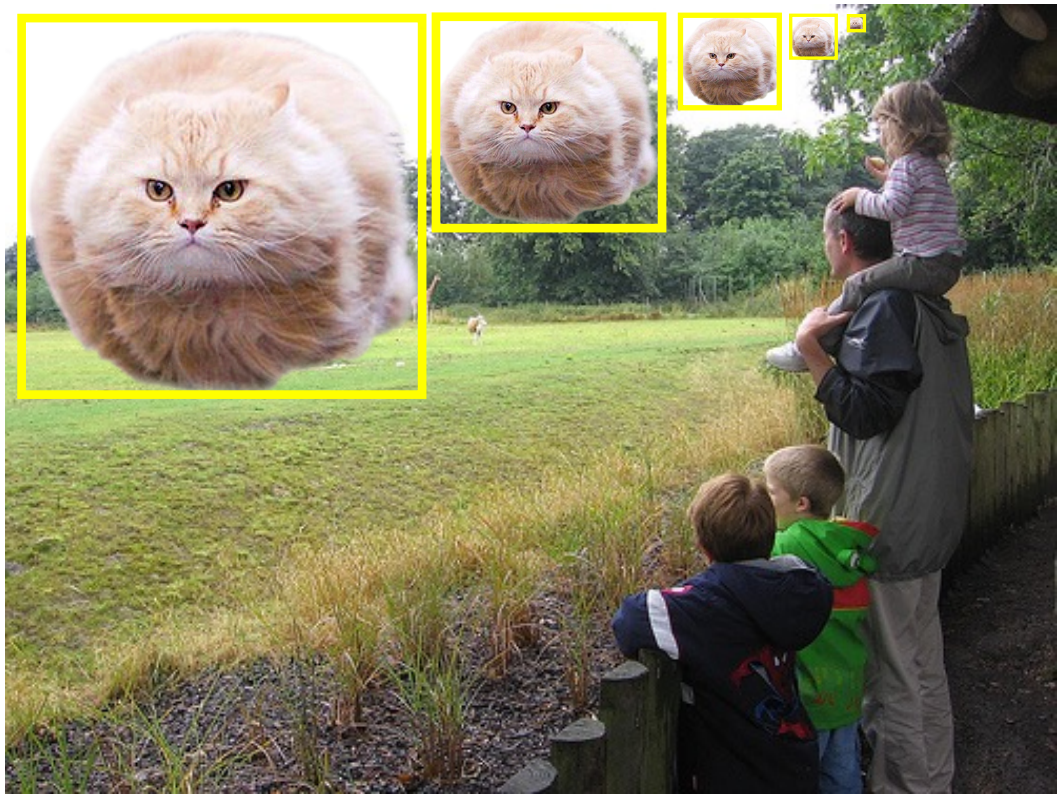


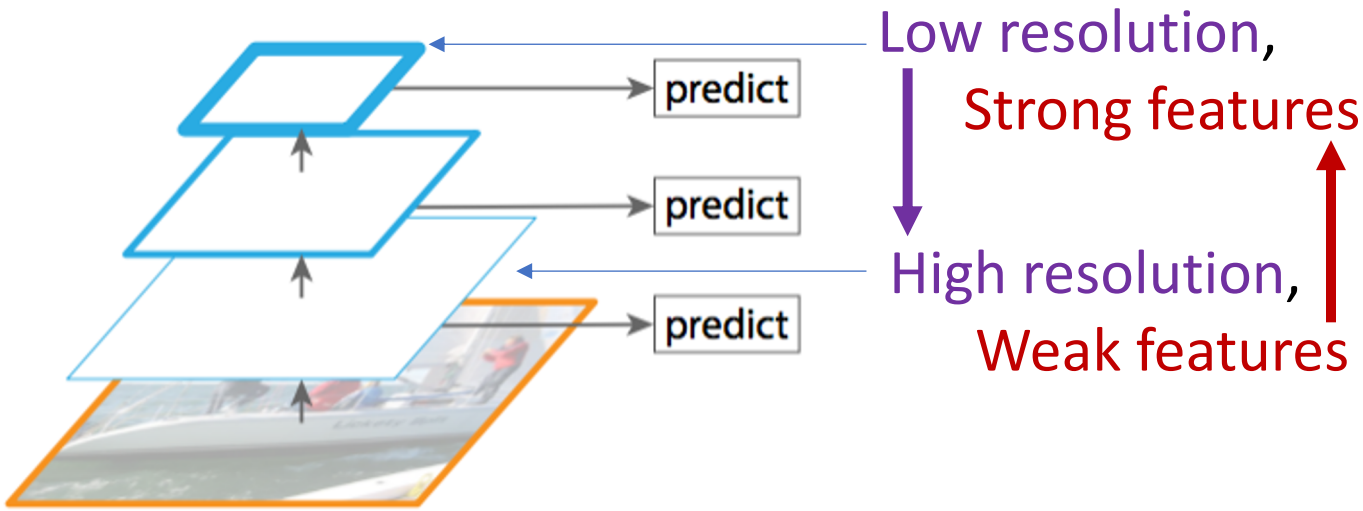
(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 3: Naïve In-network Pyramid



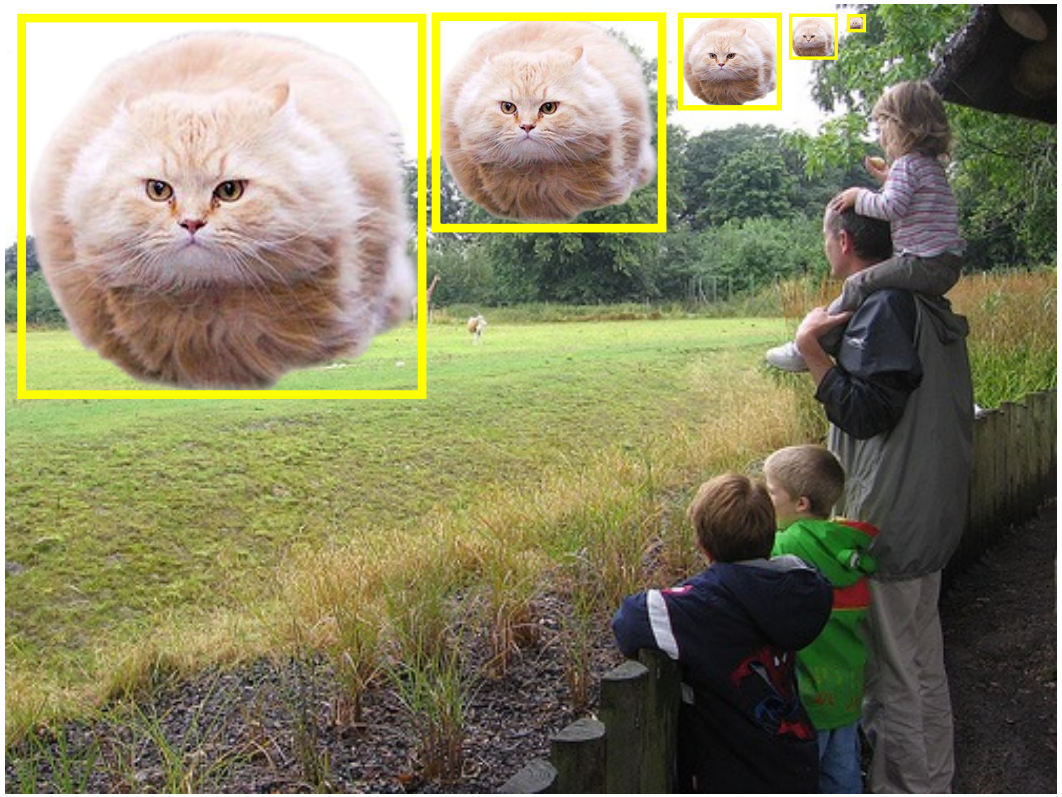


(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 4: Feature Pyramid Network

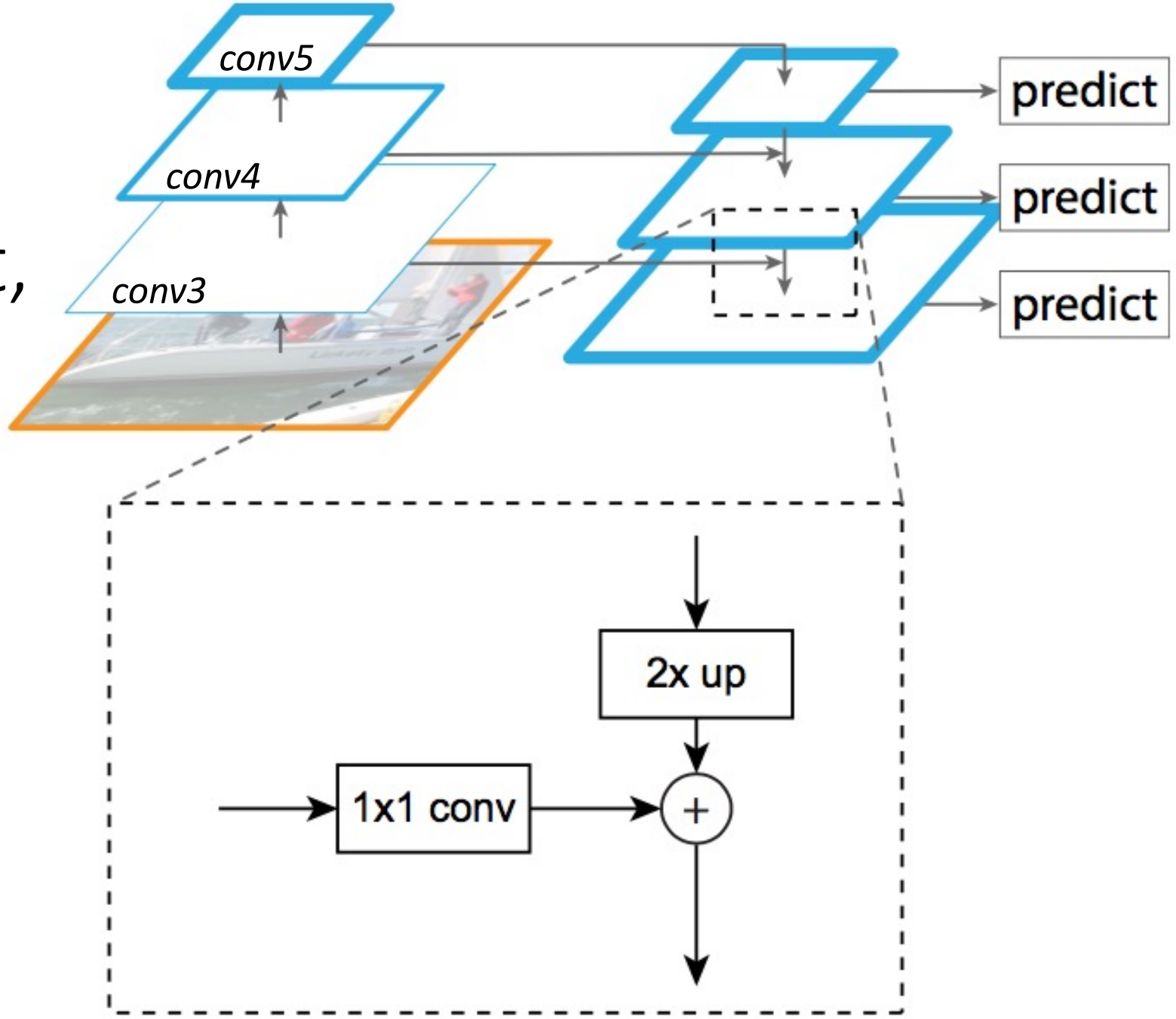


(d) Feature Pyramid Network

Top-down enrichment of high-res features – *fast, less suboptimal*

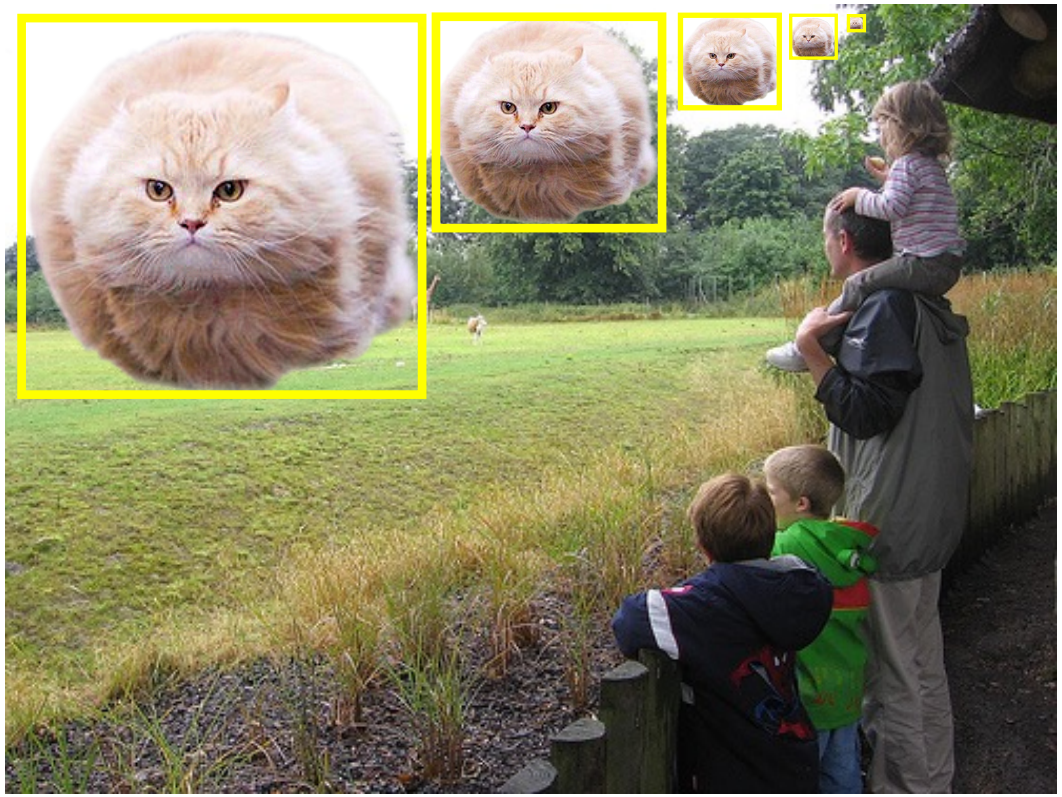
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

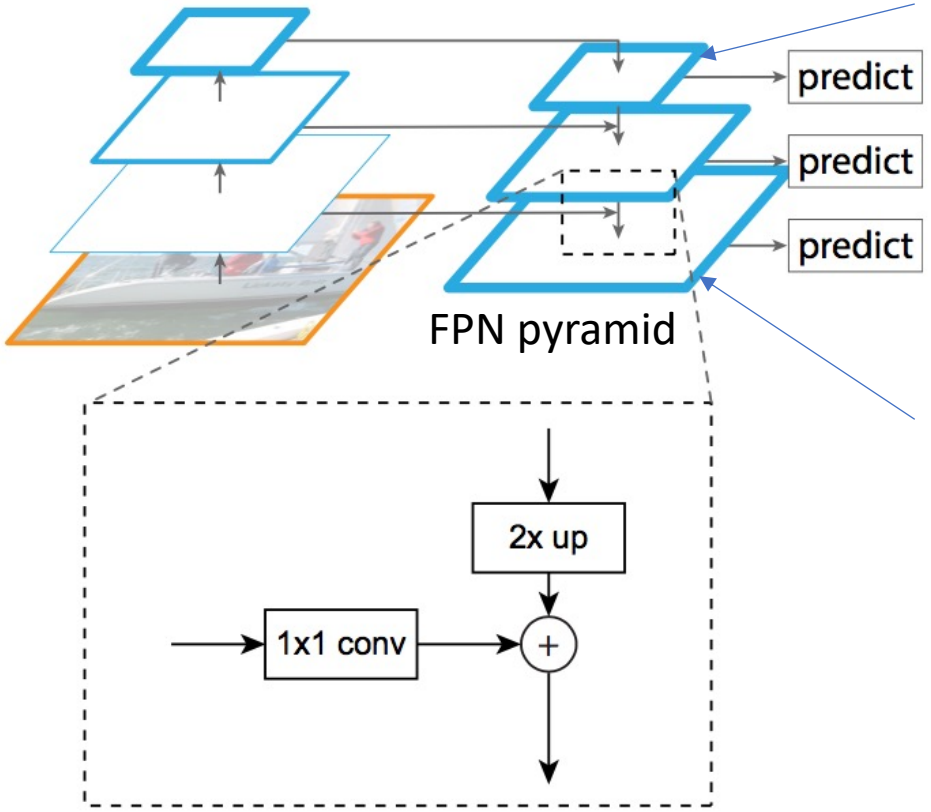
FPN: Light-weight, Top-down Refinement Module



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

No Compromise on Feature Quality, Still Fast



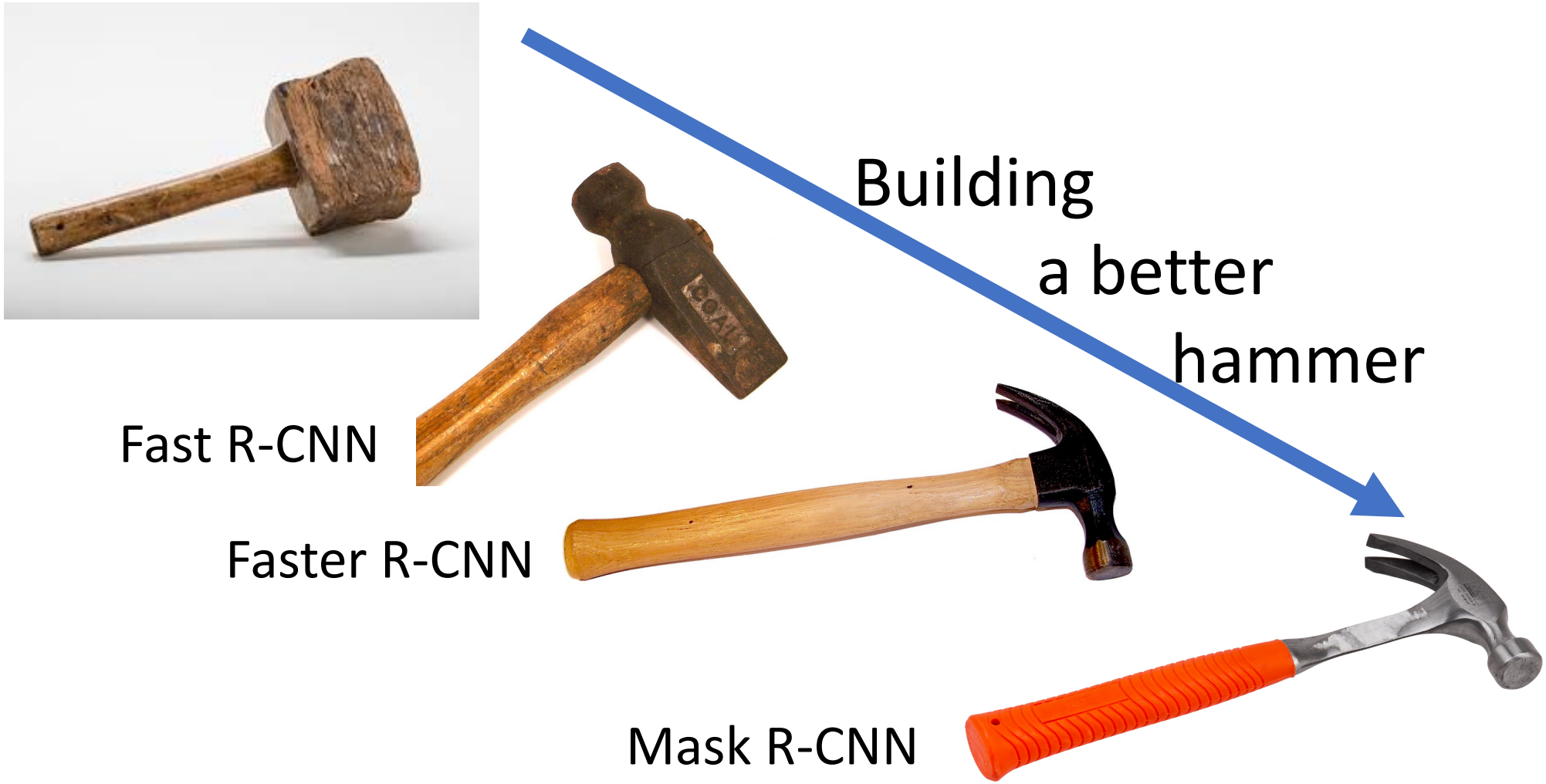


Low resolution,
Strong features

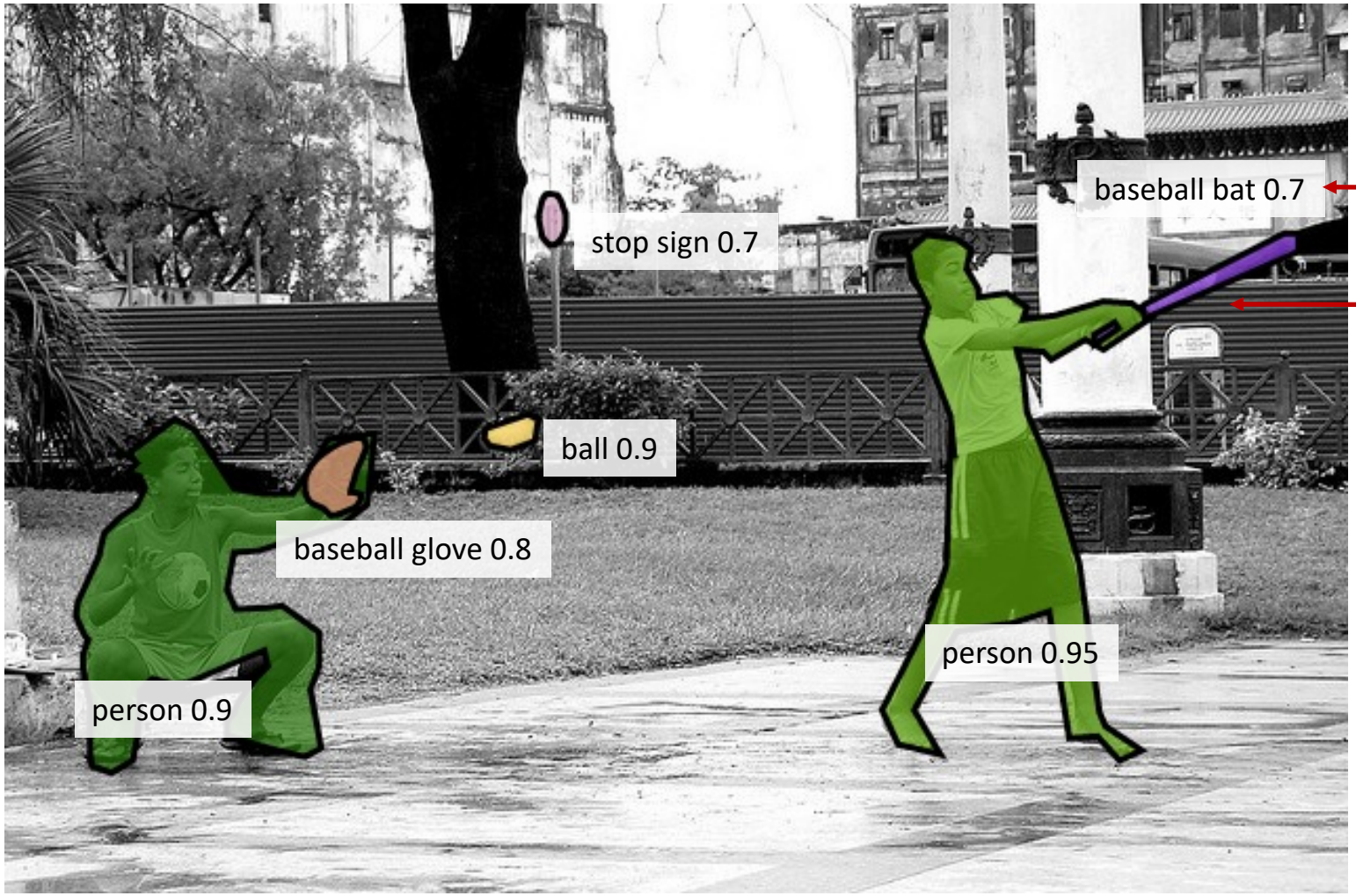
High resolution,
Strong features

Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017. See also: Shrivastava's TDM.

Mask R-CNN: The Final Hammer



Object Detection with Segmentation Masks



"Instance segmentation"

He et al. Mask R-CNN. ICCV 2017.

Mask R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FPN(I)

I:

RPN(f_I)

RoIAlign

MLP

Softmax clf.

Box regressor

Cascaded heads (inference only)

FCN

Masks

An additional head is added to predict instance-level segmentation masks

He et al. Mask R-CNN. ICCV 2017.

Mask R-CNN

Per-image computation

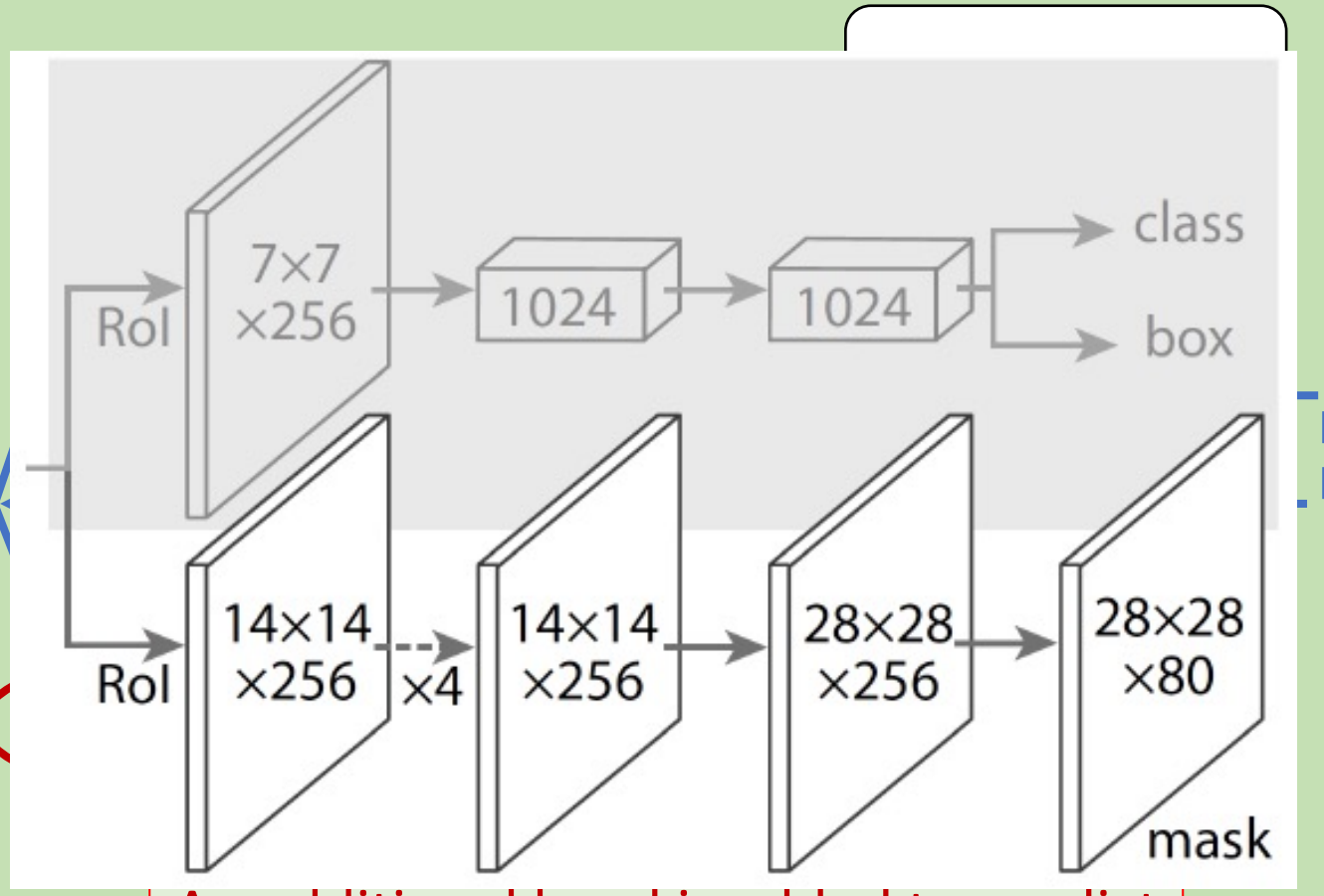
f_I = FPN(I)

I:

RPN(f_I)

RoIAlign

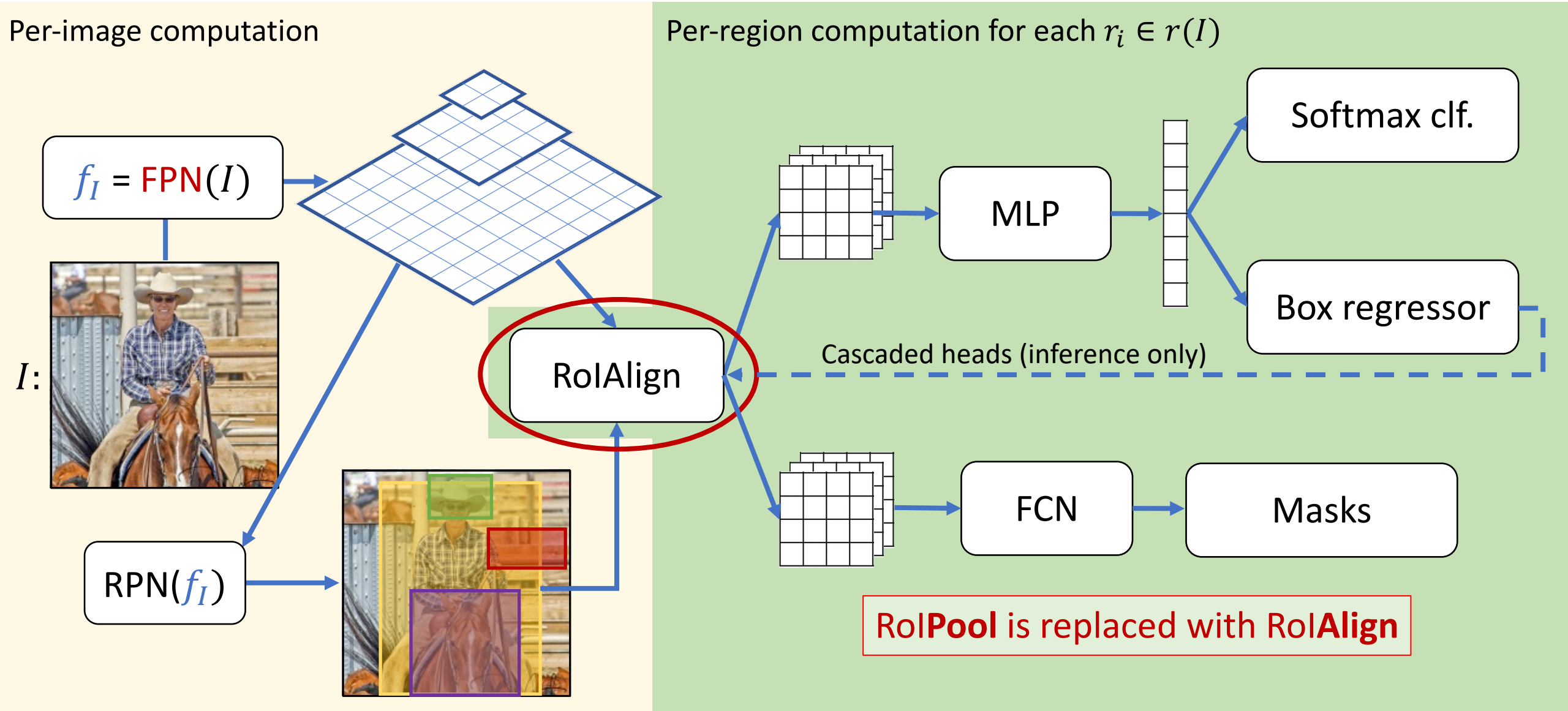
Per-region computation for each $r_i \in r(I)$



An additional head is added to predict instance-level segmentation masks

He et al. Mask R-CNN. ICCV 2017.

Mask R-CNN



He et al. Mask R-CNN. ICCV 2017.

RoIAlign Operation (on each Proposal)

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., 2x2)

f_I = FCN(I)

Region of Interest (RoI)

Grid of bilinear interpolation points

(Variable size RoI)

(Conv feature map)

RoIAlign transform

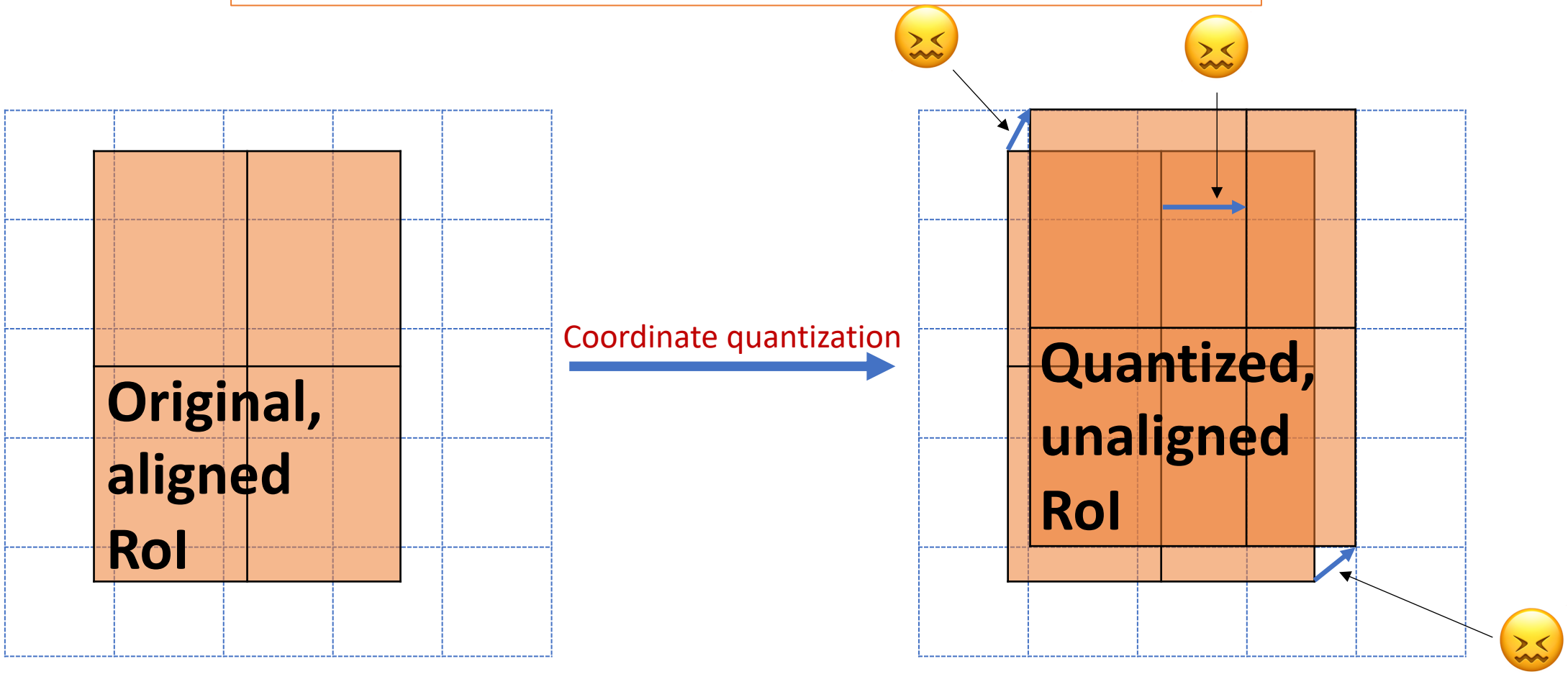
(Fixed dimensional representation)

MLP/FCN

Feature value is average of interpolated values

Compare to RoIPool and RoIWarp

Quantization breaks pixel-to-pixel **alignment** between input and output

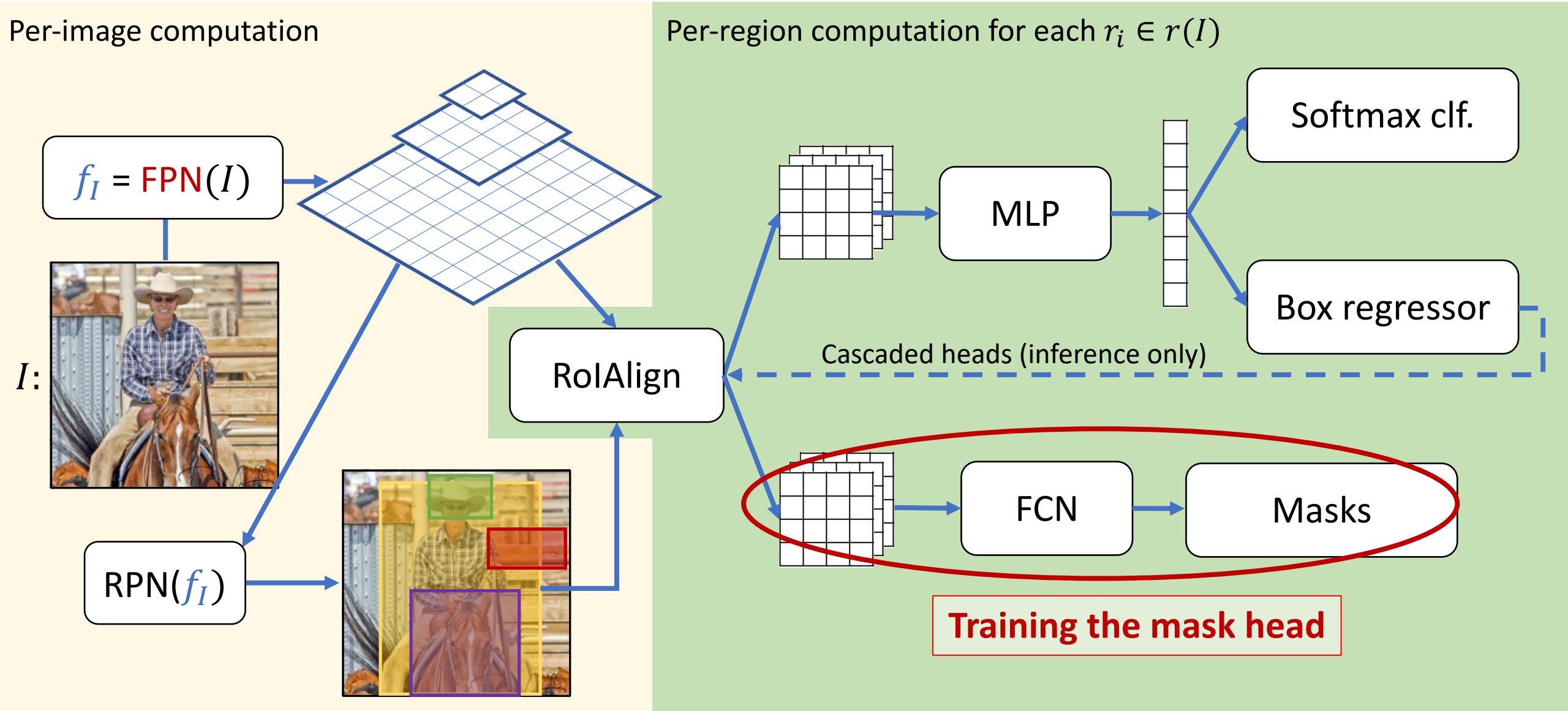


Effects of RoIAlign Operation

	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*

He et al. Mask R-CNN. ICCV 2017.

Mask R-CNN



Mask R-CNN: Inference

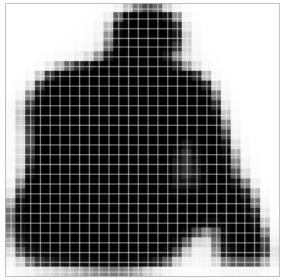
1. Perform Faster R-CNN inference
   - Generate proposals (RPN)
   - Score the proposals
   - Regress from proposals to refined detection boxes
   - Apply NMS and take the top *K* (= 100, e.g.)

2. Run RoIAlign and mask head on top-*K* refined, post-NMS boxes
   - Fast (only compute masks for top-*K* detections)
   - Improves accuracy (uses *refined* detection boxes, not proposals)

Mask Prediction

Validation image with box detection shown in red

28x28 soft prediction from Mask R-CNN
(enlarged)



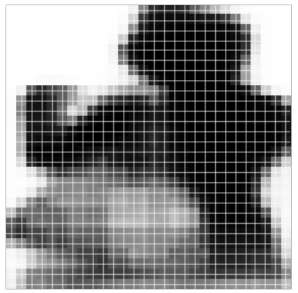
Soft prediction resampled to image coordinates
(bilinear and bicubic interpolation work equally well)

Final prediction (threshold at 0.5)

Mask Prediction

Validation image with box detection shown in red

28x28 soft prediction



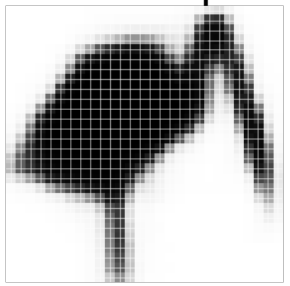
Resized soft prediction

Final mask

Mask Prediction

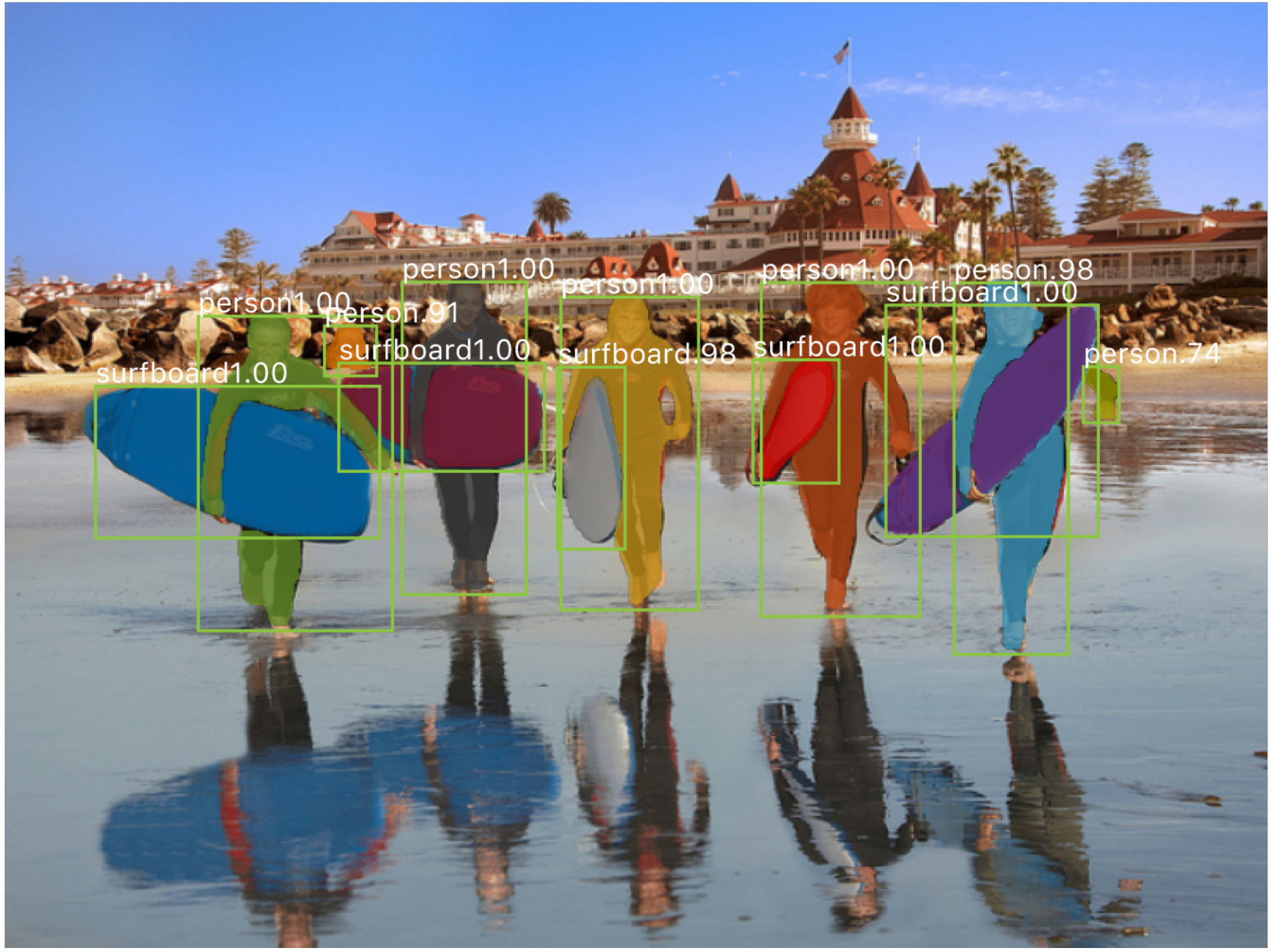
Validation image with box detection shown in red

28x28 soft prediction



Resized Soft prediction

Final mask


person1.00
person1.00
person1.00
person.98
surfboard1.00
person1.00
person.91
surfboard1.00
surfboard.98
surfboard1.00
person.74
surfboard1.00


person.88
person1.00
tv.98
tv.84
person1.00
person1.00
bottle.97
wine glass.99
dining table.95
wine glass1.00
wine glass1.00

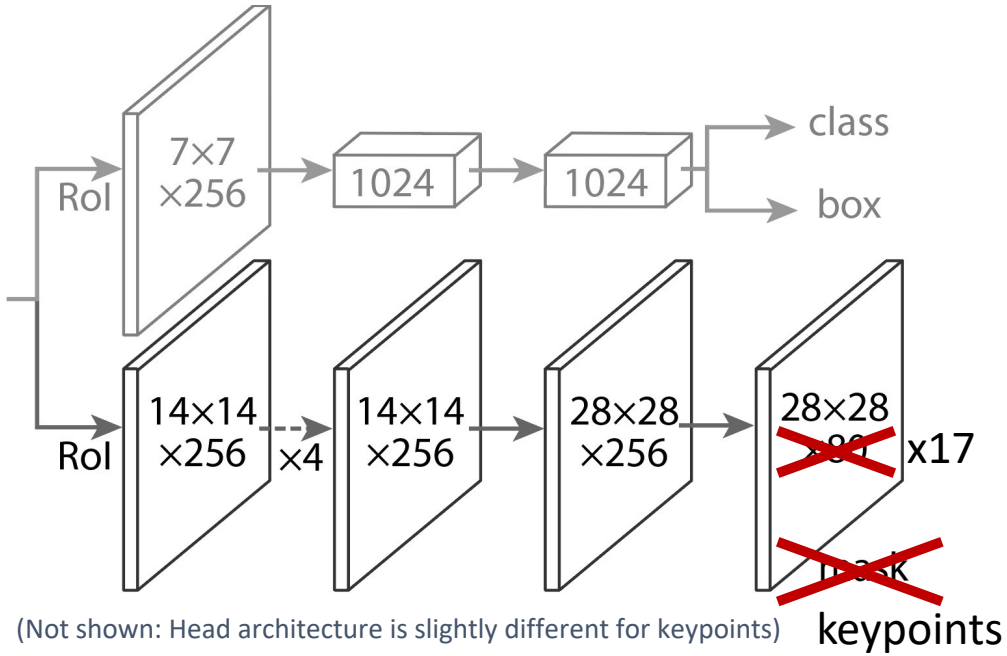
He et al. Mask R-CNN. ICCV 2017.

Instance Segmentation Results on COCO

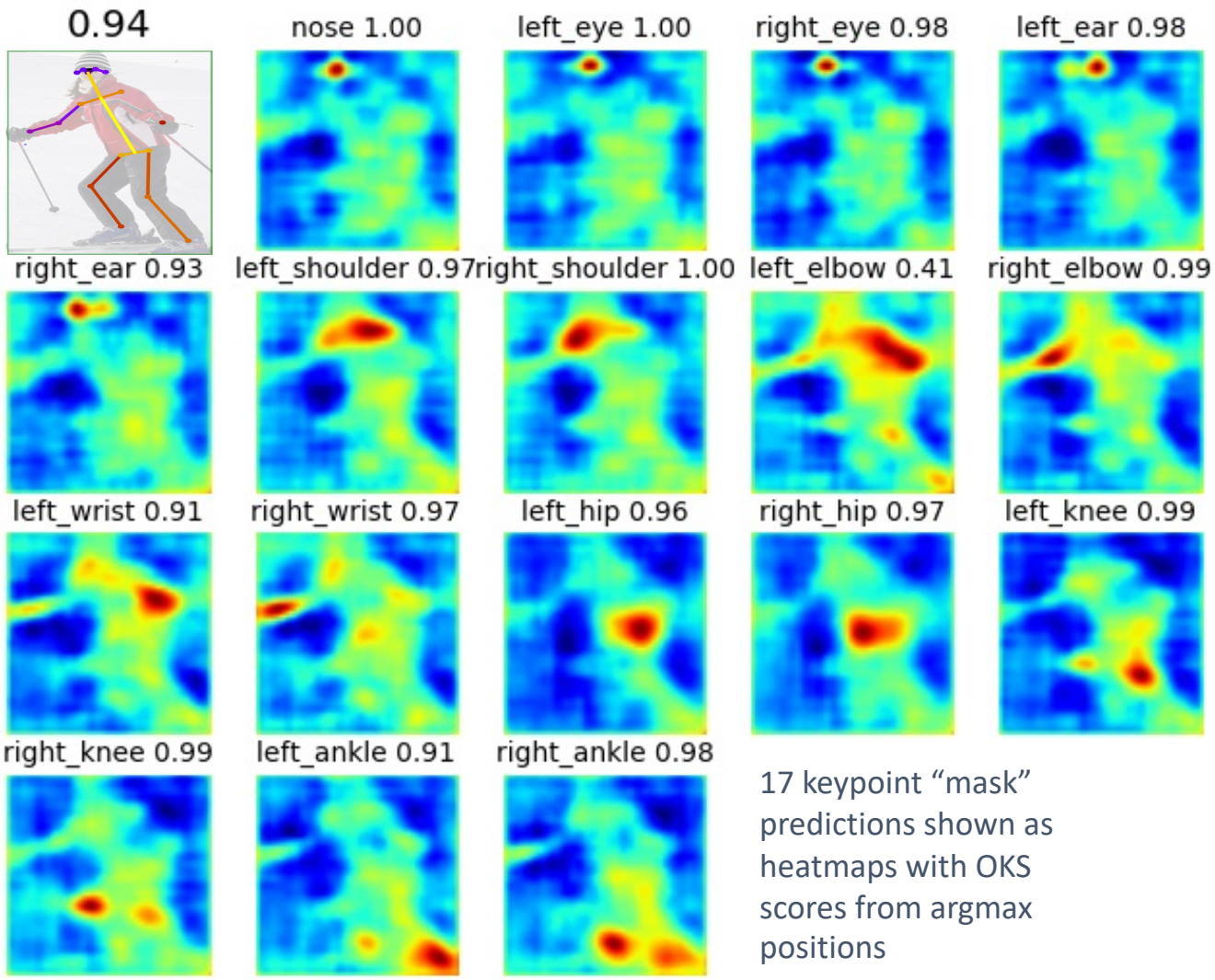
	backbone	AP	AP_{50}
MNC [7]	ResNet-101-C4	24.6	44.3
FCIS [20] +OHEM	ResNet-101-C5-dilated	29.2	49.5
FCIS+++ [20] +OHEM	ResNet-101-C5-dilated	33.6	54.5
Mask R-CNN	ResNet-101-C4	33.1	54.9
Mask R-CNN	ResNet-101-FPN	35.7	58.0

- without bells and whistles, **2 AP better** than 2016 winner
- **200ms / img**

Human Pose



(Not shown: Head architecture is slightly different for keypoints)



17 keypoint "mask" predictions shown as heatmaps with OKS scores from argmax positions

- Add keypoint head (28x28x17)
- Predict one "mask" for each keypoint
- Softmax over spatial locations (encodes one keypoint per mask "prior")

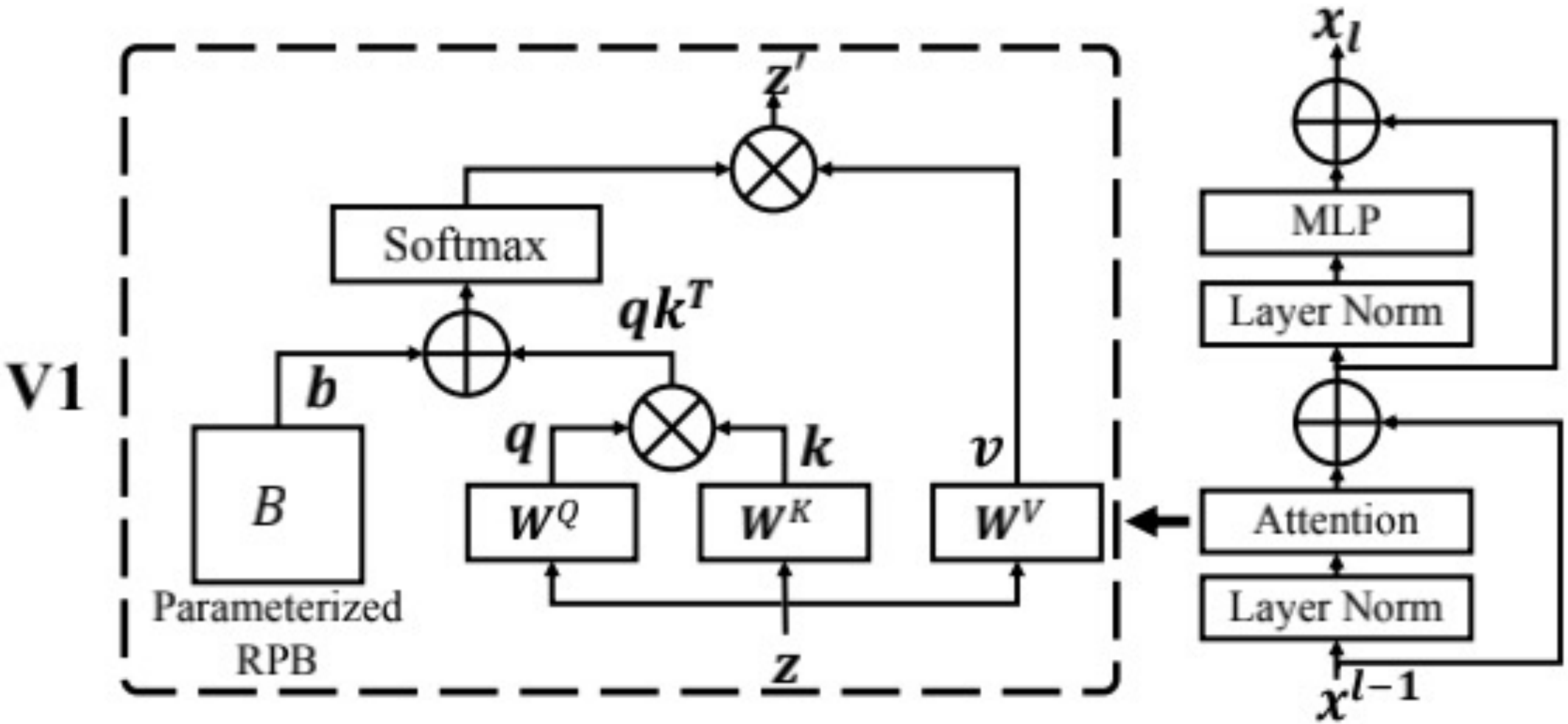

www.tfrancis.co.uk

Overview

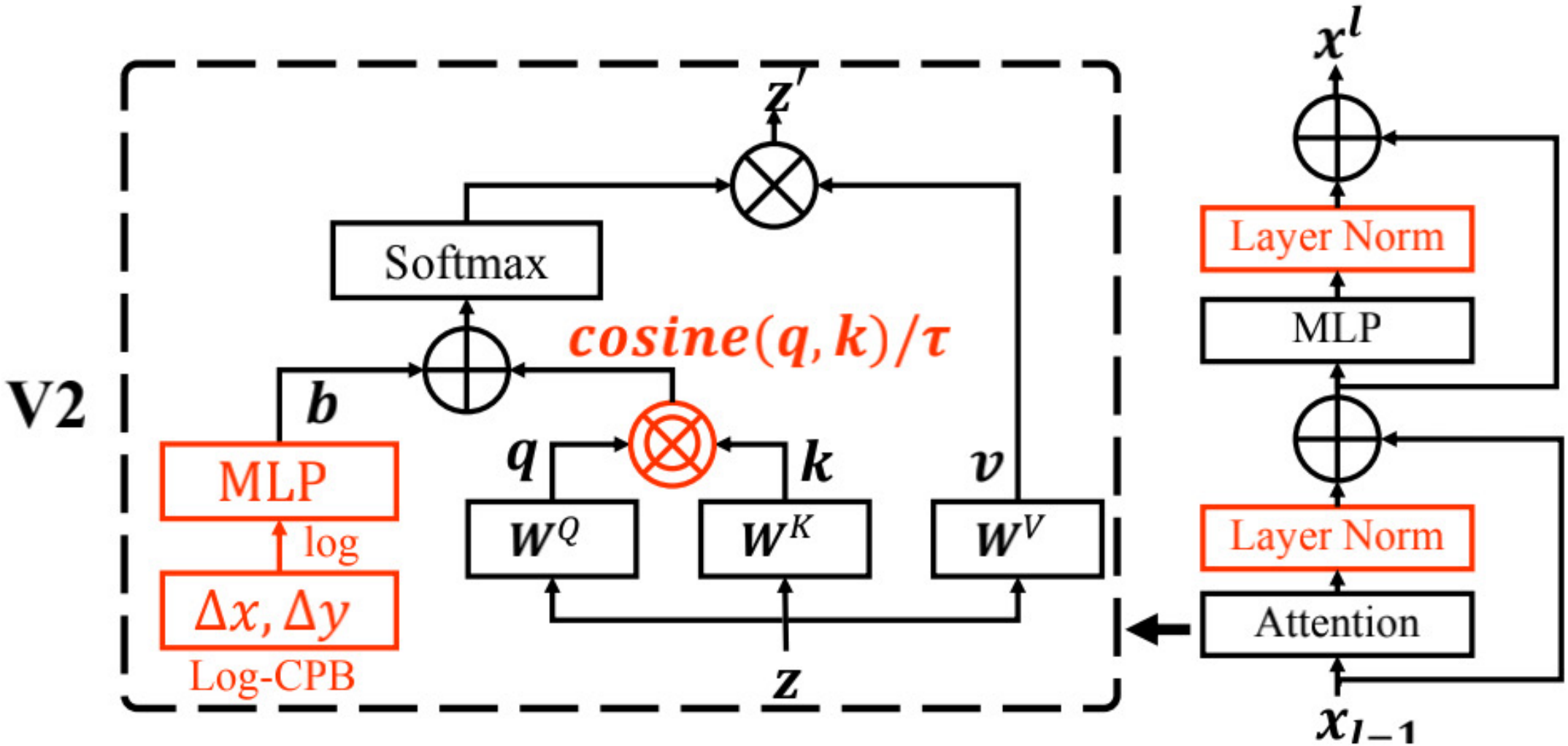
Topics to cover

- Object detection intro
- The Generalized R-CNN framework
  - **R-CNN**
  - Fast R-CNN
  - Faster R-CNN
  - **Mask R-CNN**
- Future works + Discussion
  - **SwinV2**

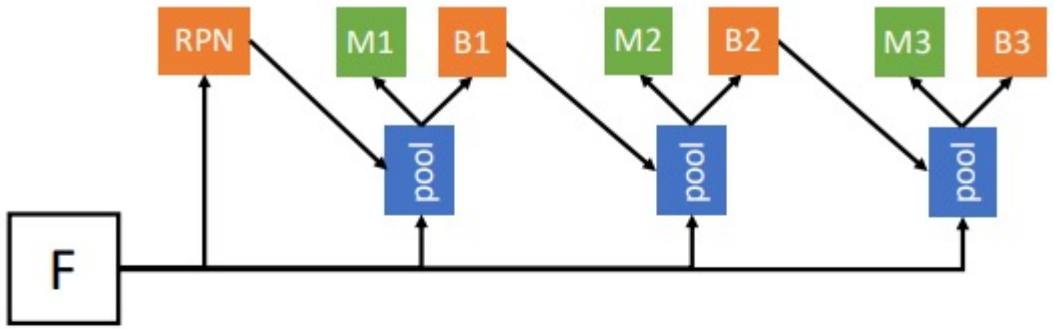
Swin


V1
z'
Softmax
qk^T
b
q
k
v
B
W^Q
W^K
W^V
Parameterized
RPB
z
x_l
MLP
Layer Norm
Attention
Layer Norm
x^{l-1}

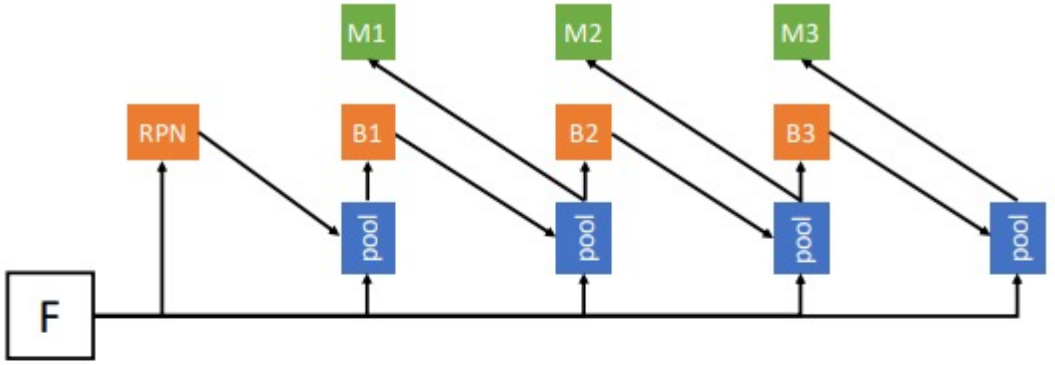
SwinV2


V2
z'
Softmax
$cosine(q,k)/\tau$
b
MLP
log
$\Delta x, \Delta y$
Log-CPB
q
k
v
W^Q
W^K
W^V
z
x^l
Layer Norm
MLP
Layer Norm
Attention
x_{l-1}

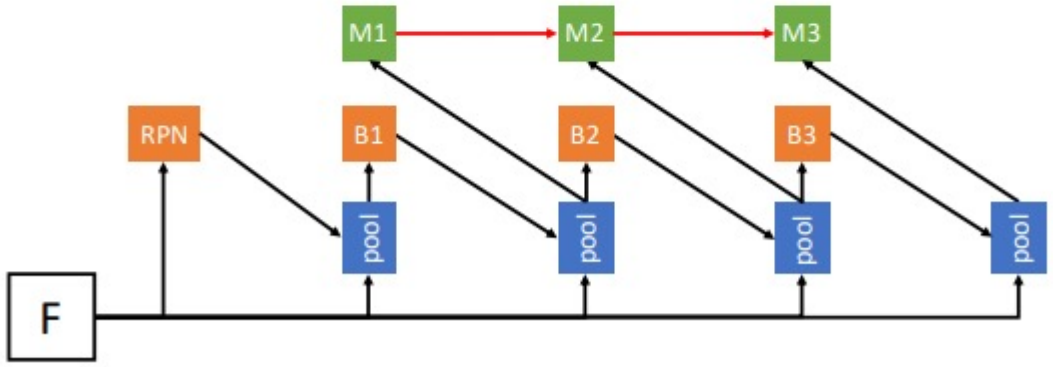
Hybrid Task Cascade



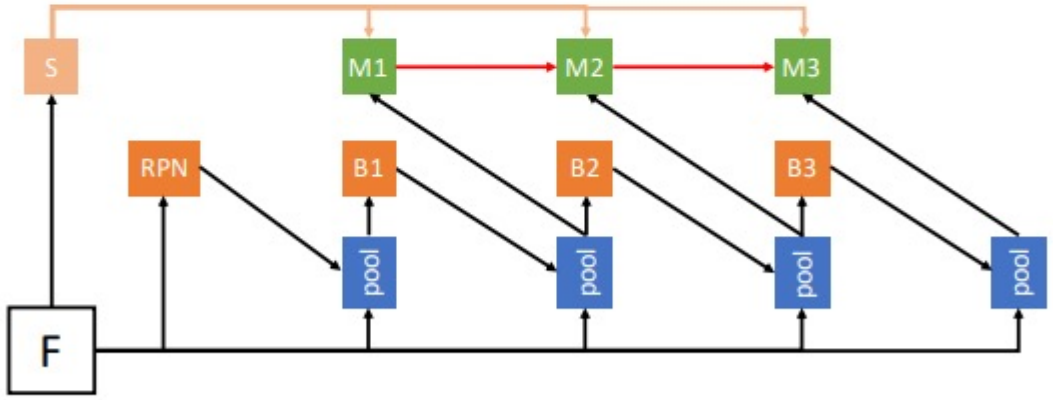
(a) Cascade Mask R-CNN



(b) Interleaved execution

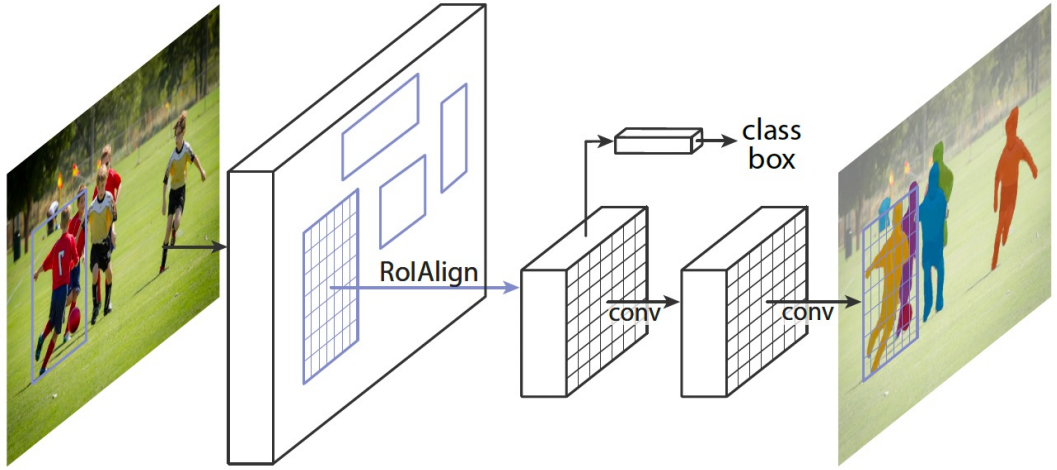


(c) Mask information flow

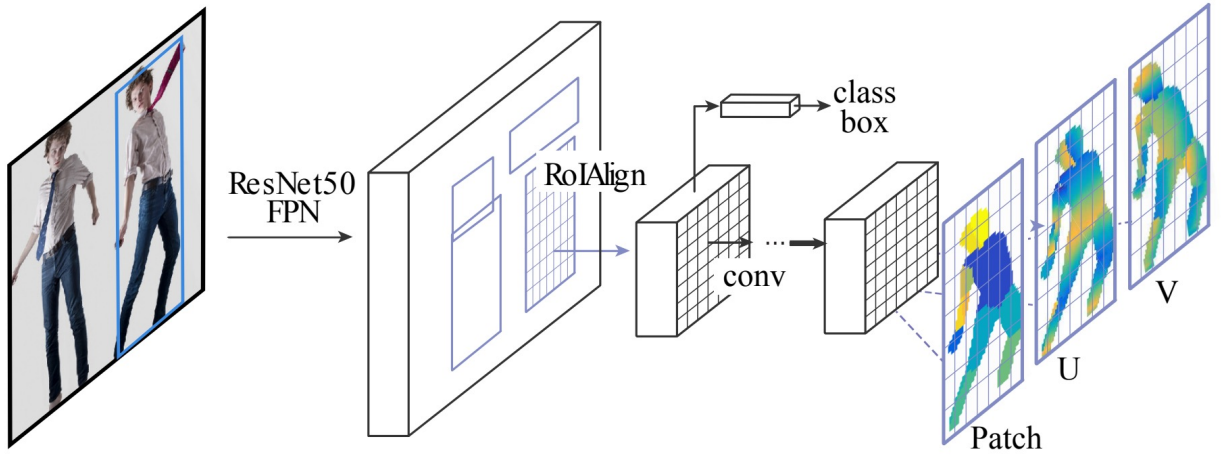


(d) Hybrid Task Cascade (semantic feature fusion with box branches is not shown on the figure for neat presentation.)

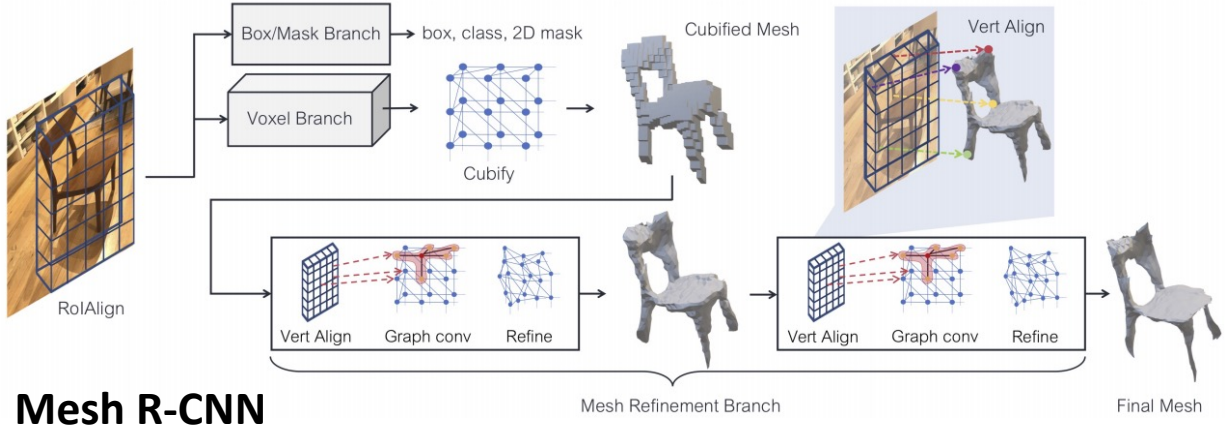
Generalized R-CNN: Adding More Heads



Mask R-CNN
[He, Gkioxari, Dollár, Girshick]



DensePose
[Güler, Neverova, Kokkinos]



Mesh R-CNN
[Gkioxari, Malik, Johnson. ICCV 2019]

Size

Rene Magritte, *The Listening Room*

Open Question Time

~

Interposition

Position, Probability, Size

Elephant in the Fridge

Thank You!