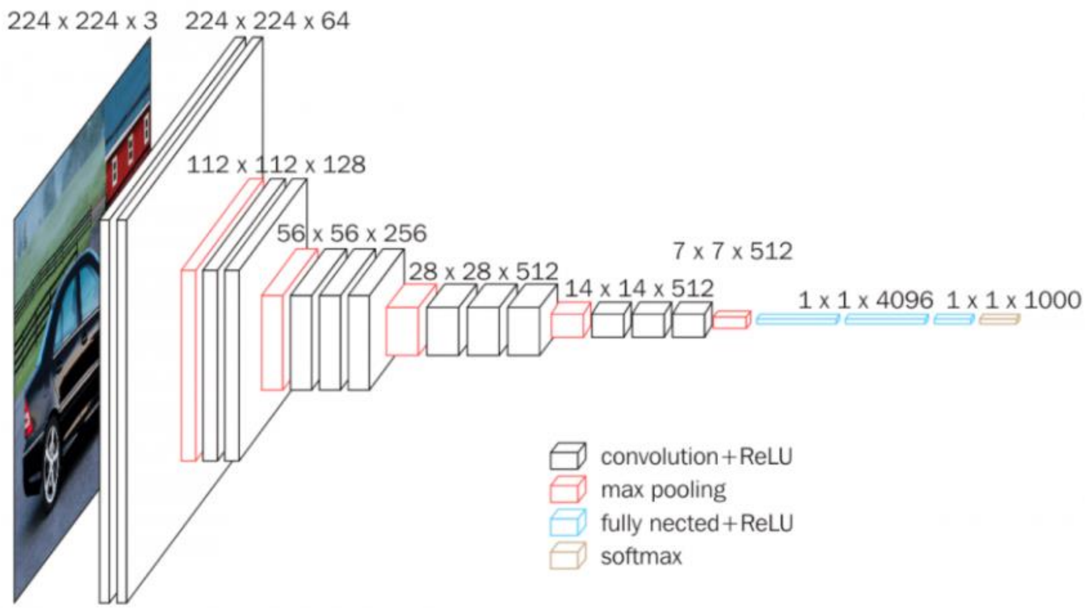
AI for Video Data Processing 3:
R-CNN for Object Detection

Outline

1. R-CNN
2. Fast R-CNN
3. Faster R-CNN
4. Faster R-CNN with FPN
5. Mask R-CNN
6. RetinaNet

1. From CNN to R-CNN

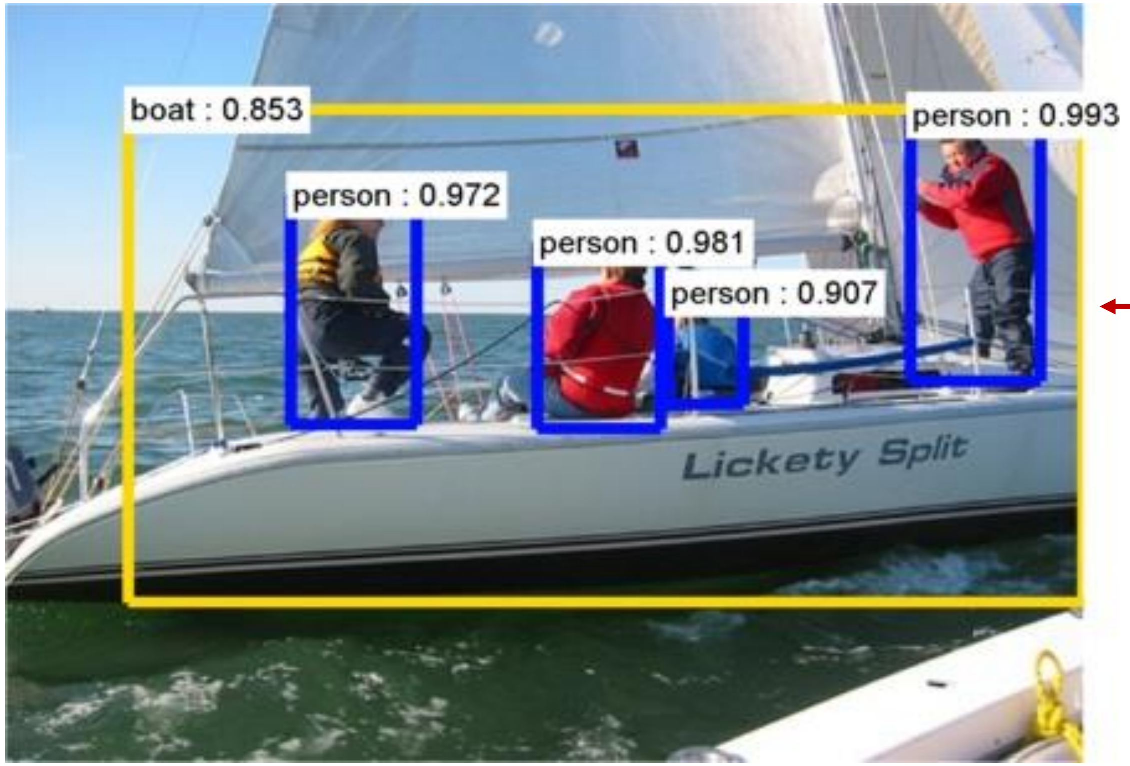
Original CNNs are designed for Image Classification:
LeNet, AlexNet, VGG, ResNet, DenseNet, ...



https://www.jeremyjordan.me/semantic-segmentation/

1. From CNN to R-CNN

If we want to detect objects in an image, then …



What?

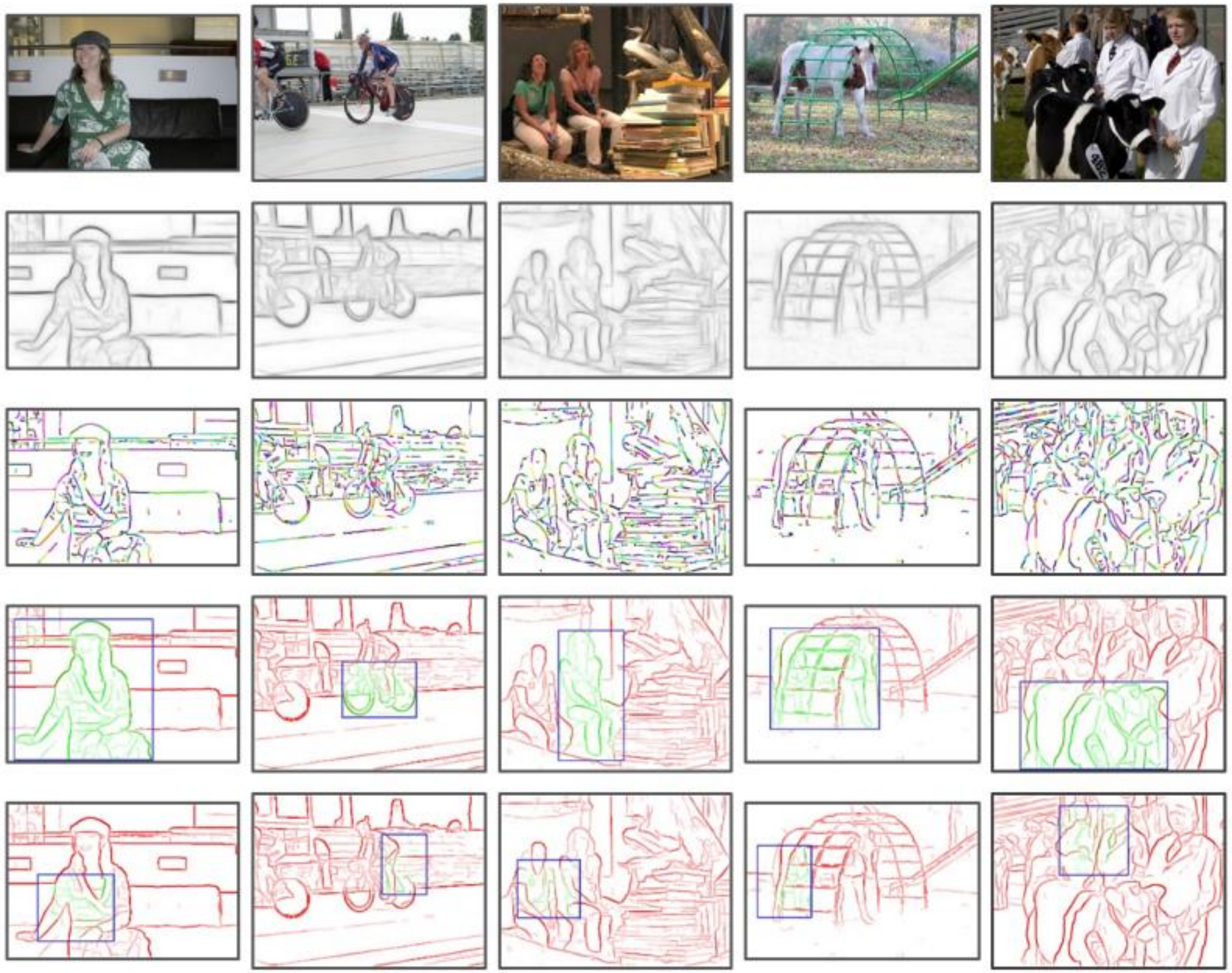
Where?

Object detection with bounding boxes

1. Object Proposal Algorithm in CV

Zitnick, C. Lawrence, and Piotr Dollár. "Edge boxes: Locating object proposals from edges." In *European Conference on Computer Vision*, pp. 391-405. Springer, Cham, 2014.

For example: 2k object proposals for one image

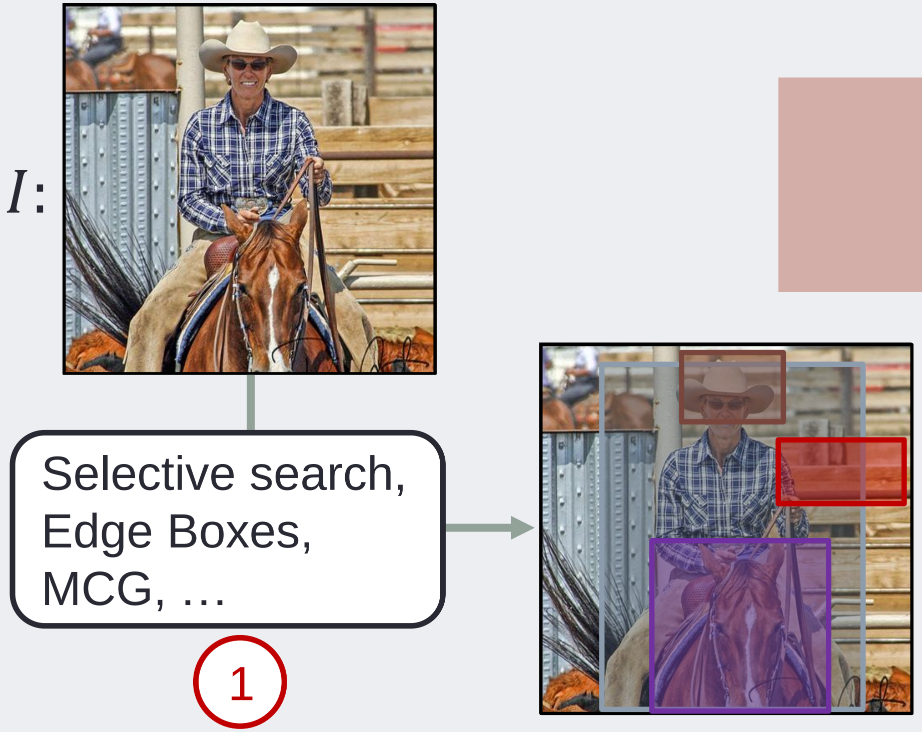


Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

1.R-CNN (Region-based Convolutional Neural Net)

Per-image computation

Per-region computation

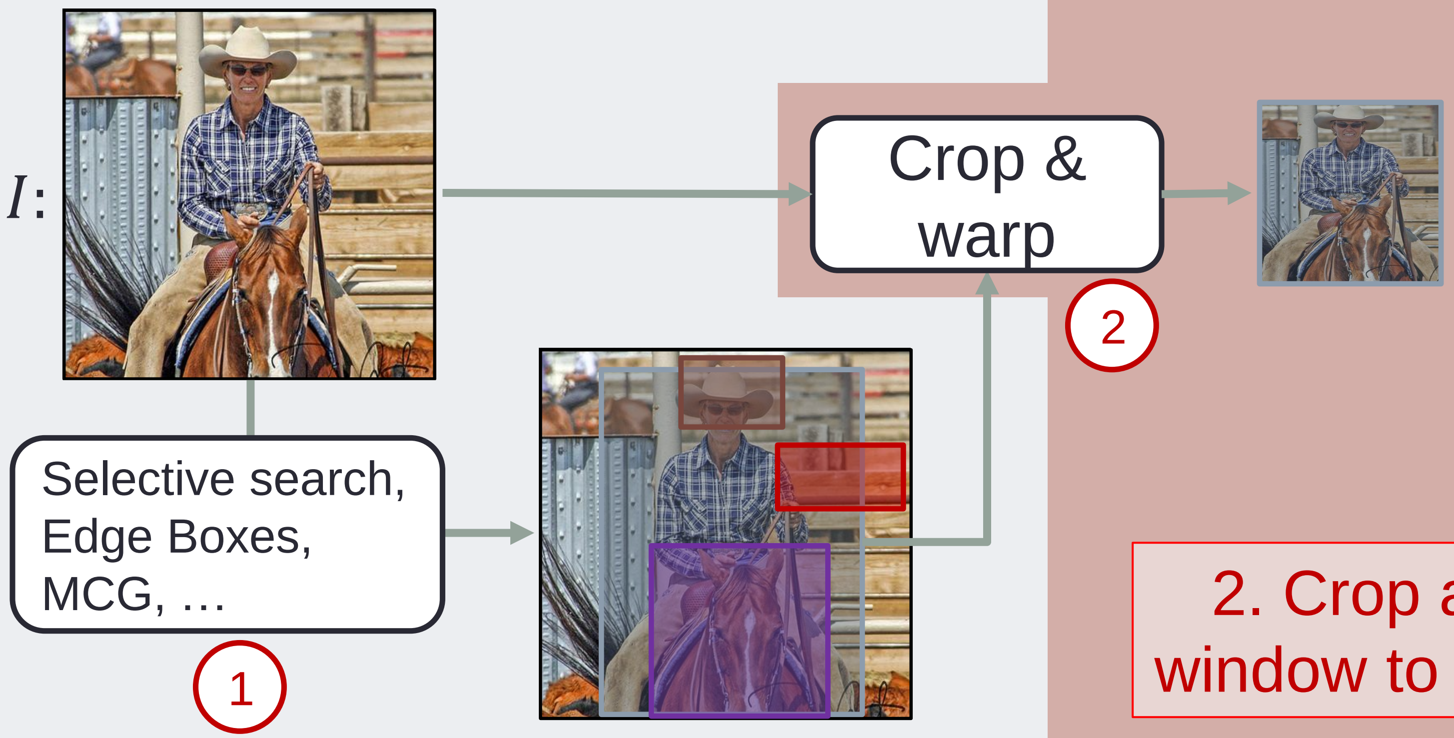


1. Use an off-the-shelf *Region of Interest* (RoI) proposal algorithm (~2k proposals per image)

1. R-CNN

Per-image computation

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

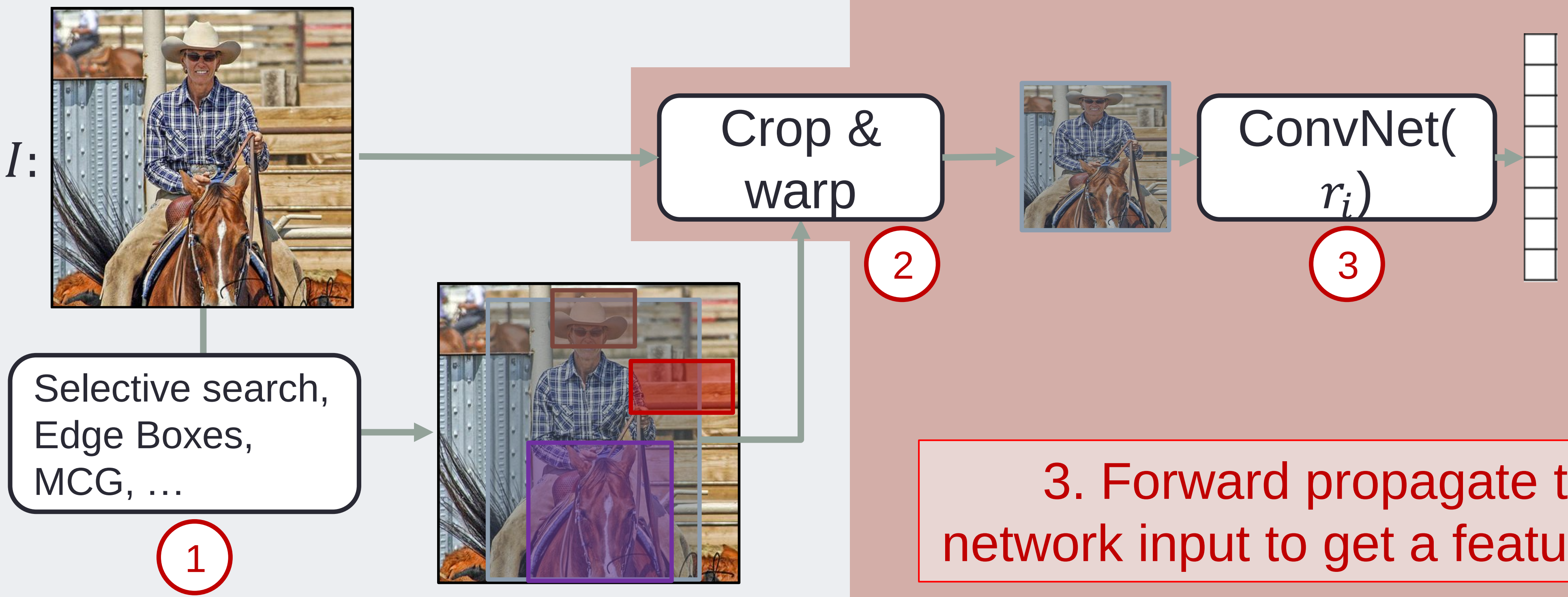


2. Crop and warp each proposal image window to obtain a fixed-size network input

1. R-CNN

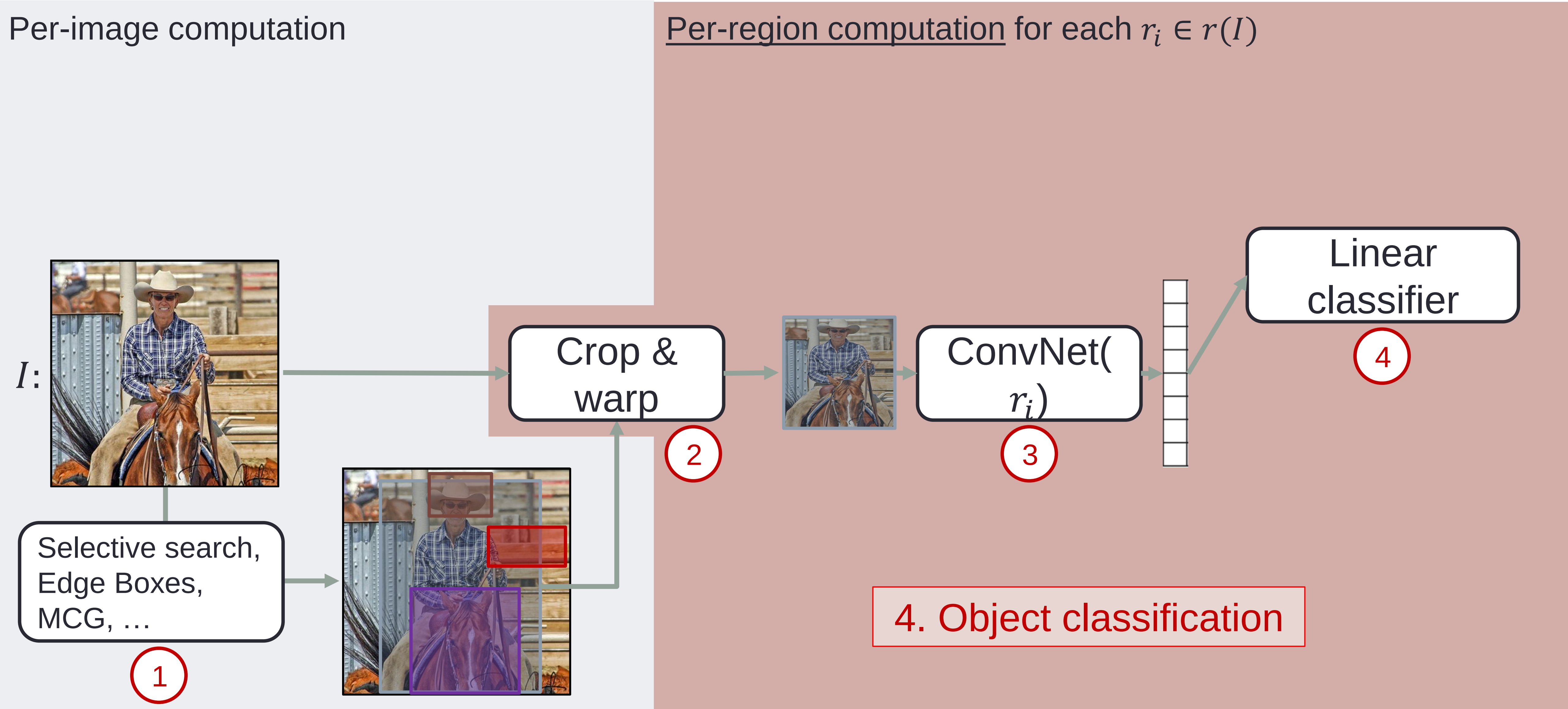
Per-image computation

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

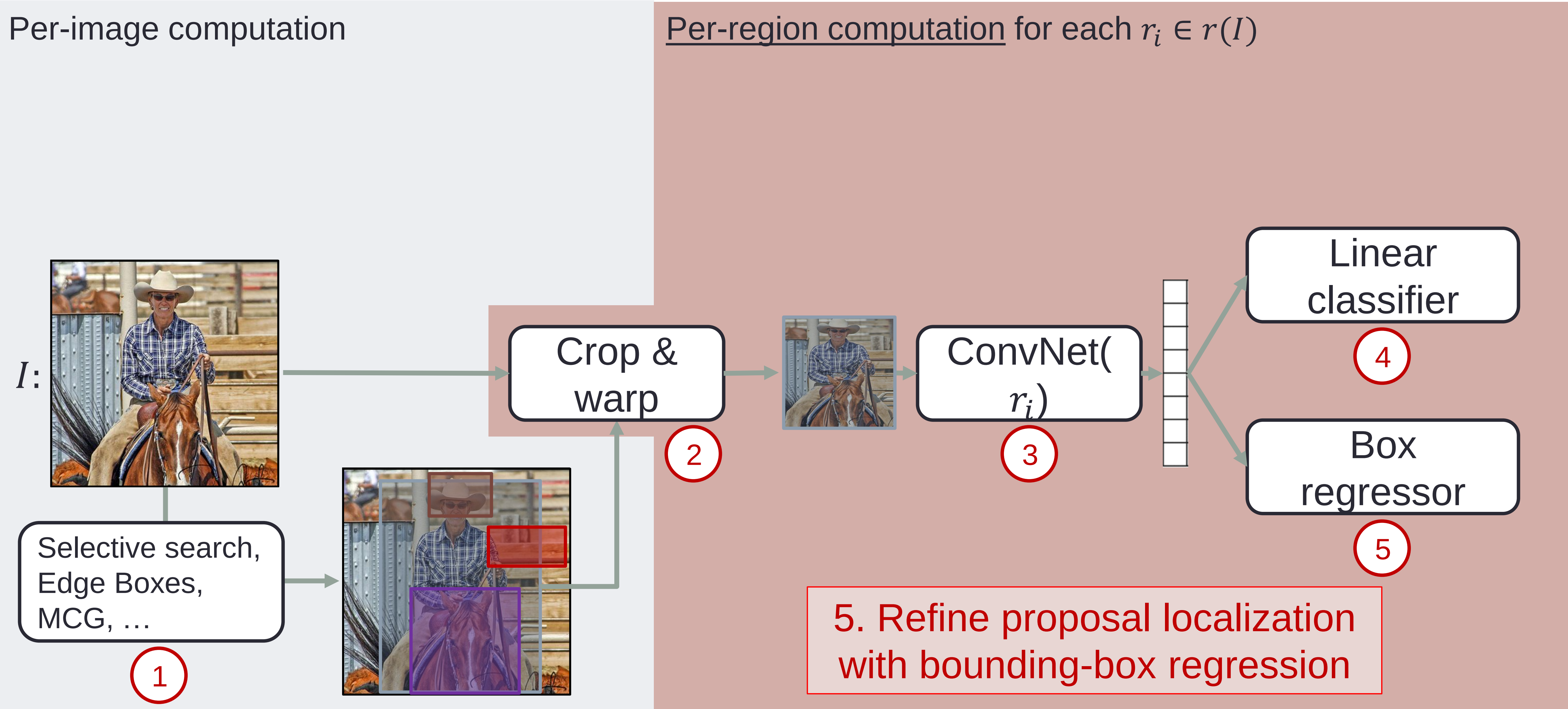


3. Forward propagate the fixed-size network input to get a feature representation

1. 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

1. 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

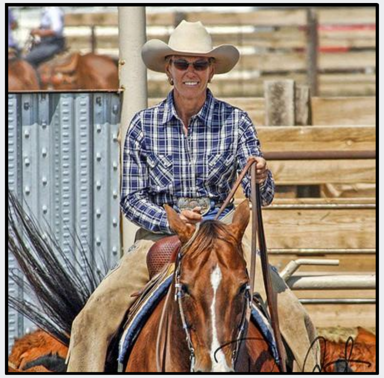
1. From R-CNN to Generalized R-CNN Framework

1. Generalized R-CNN Framework

Per-image computation

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

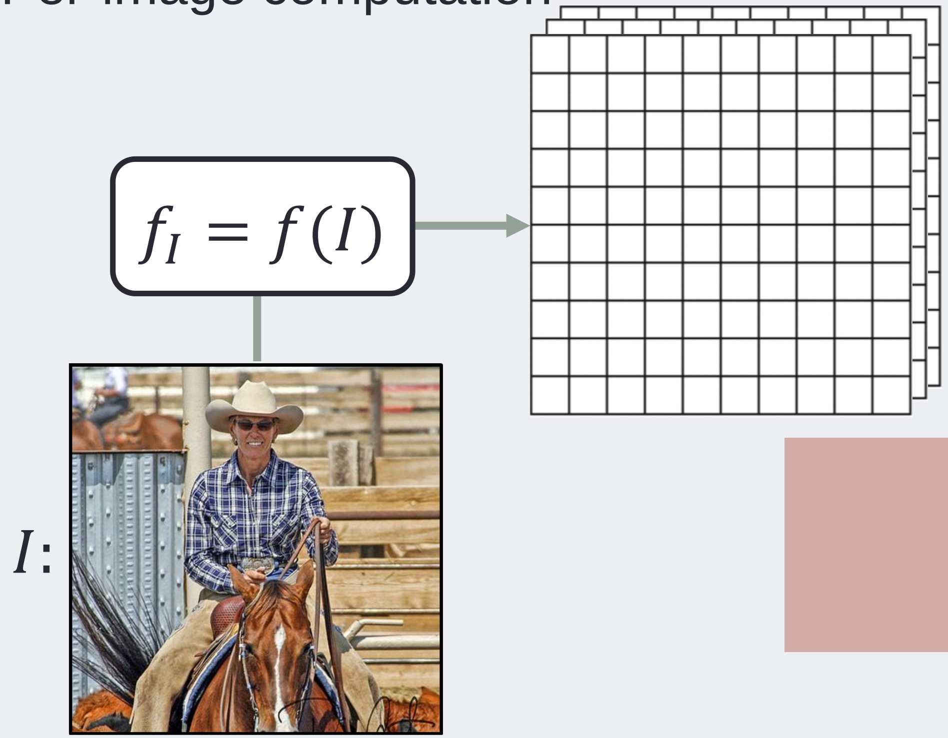
I:



Input image
per-image operations | per-region operations

1. Generalized R-CNN Framework

Per-image computation

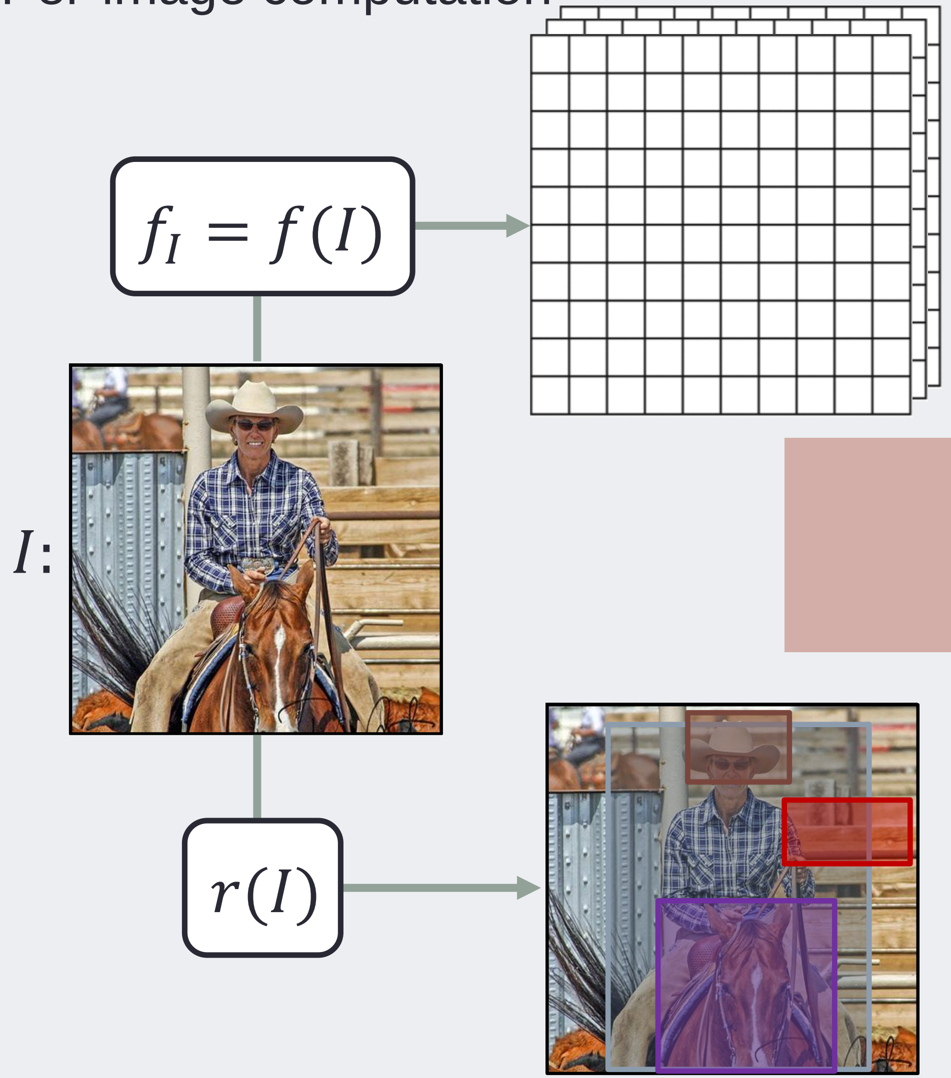


Per-region computation for each $r_i \in r(I)$

Transformation of the input image
into a featurized representation

1. Generalized R-CNN Framework

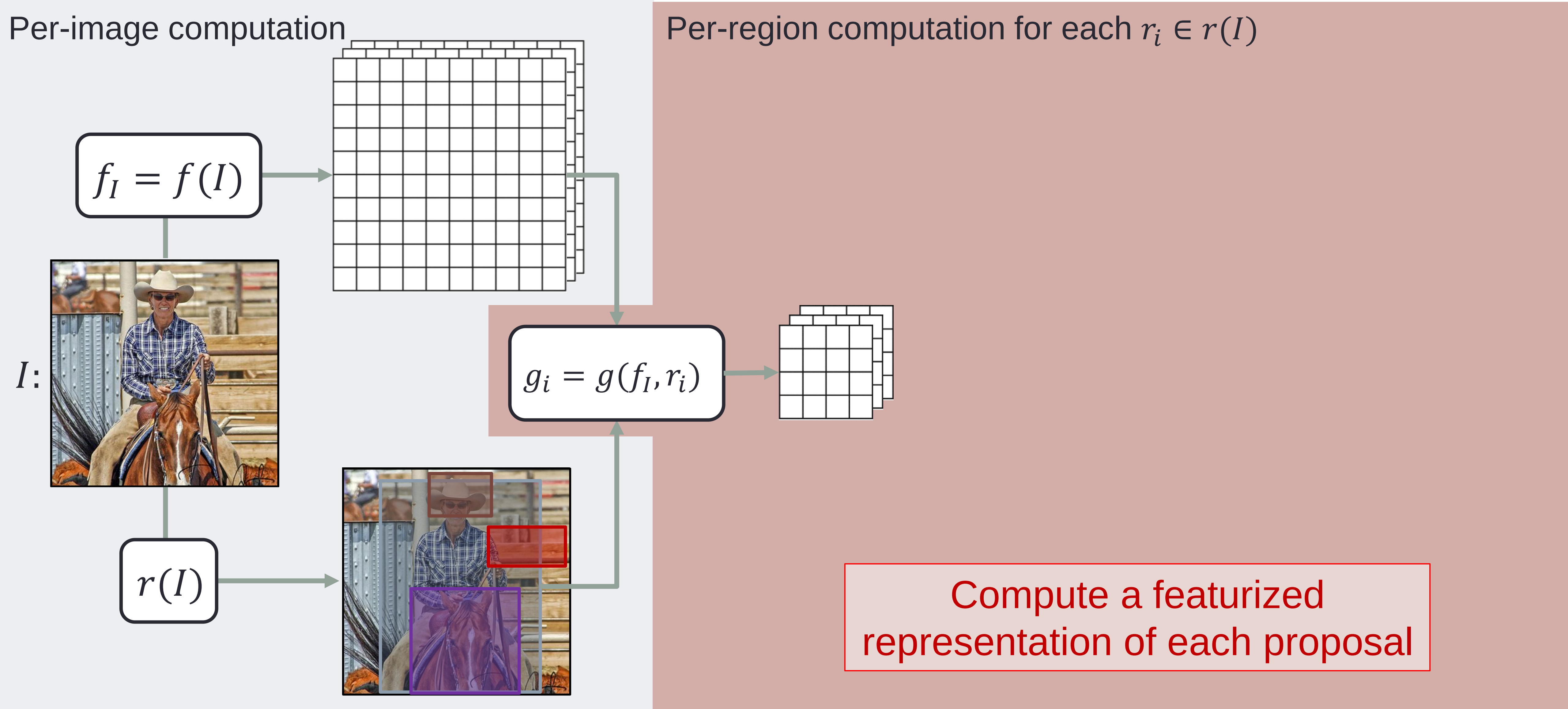
Per-image computation



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

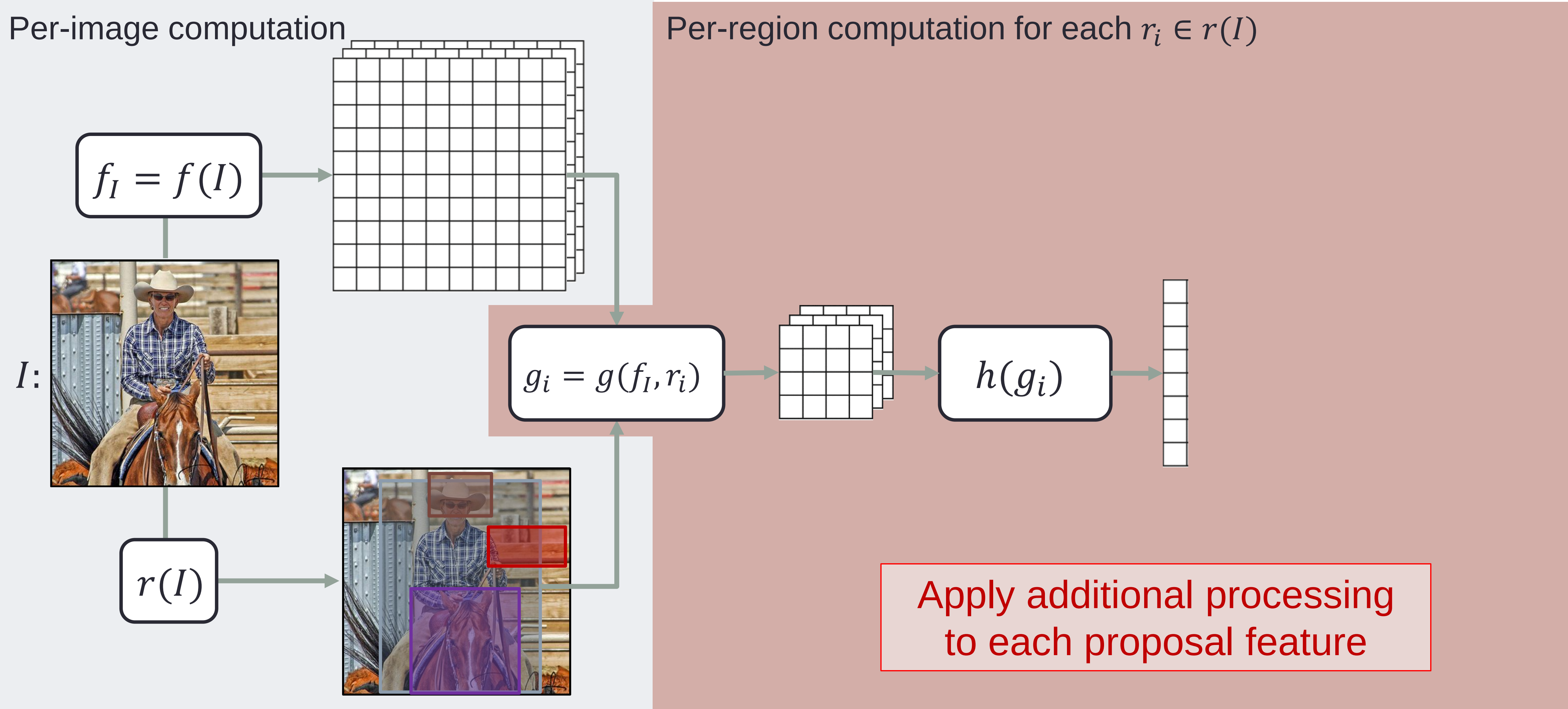
Region of Interest proposals computed for the image

1. Generalized R-CNN Framework



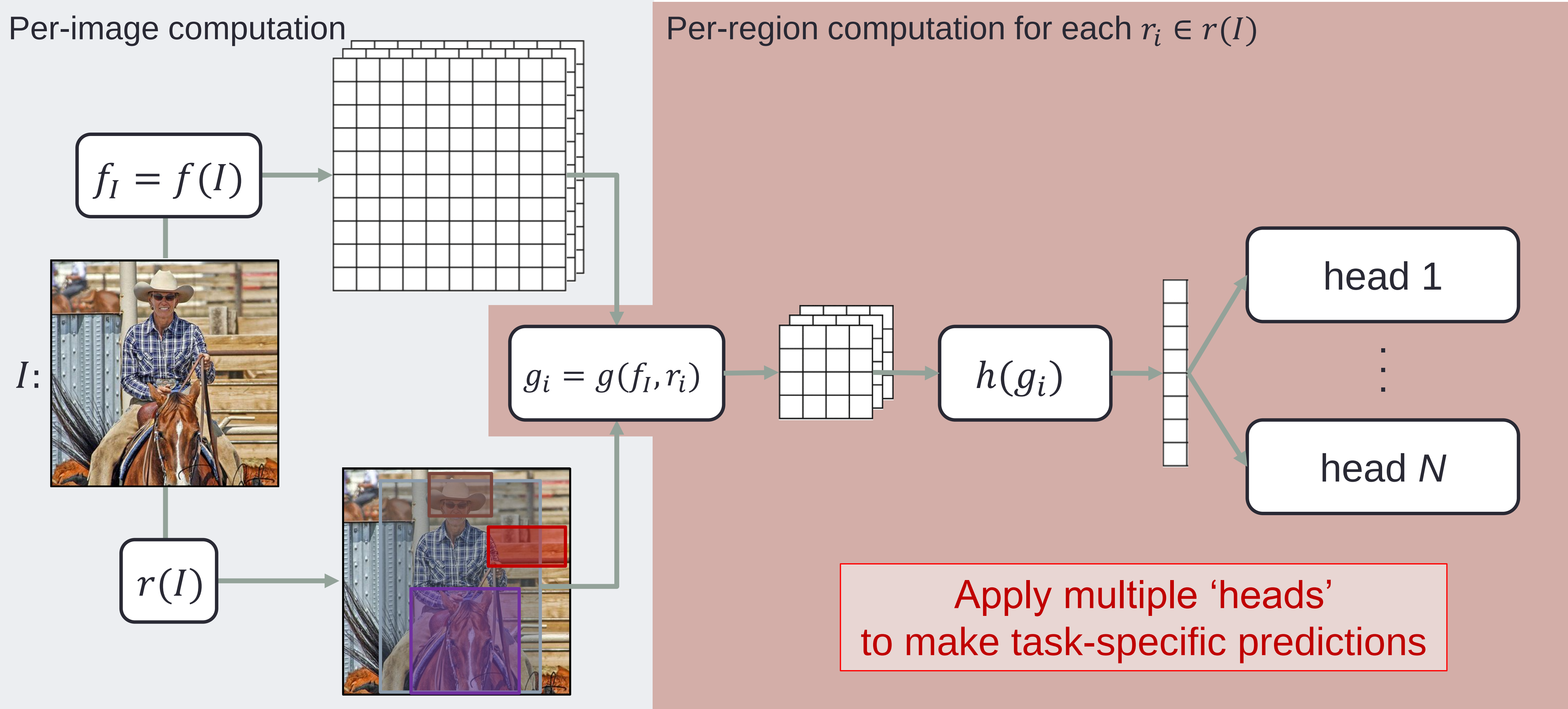
Compute a featurized representation of each proposal

1. Generalized R-CNN Framework



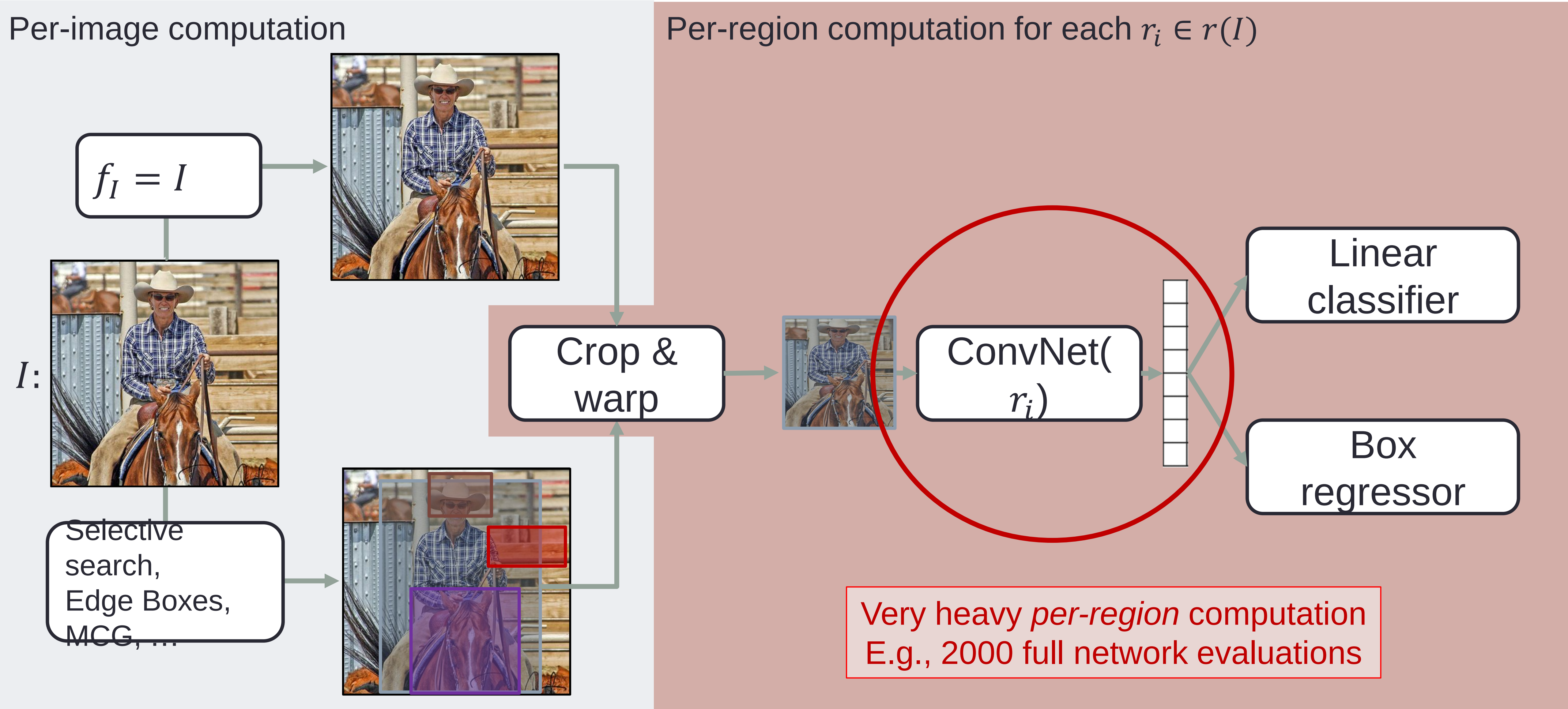
Apply additional processing to each proposal feature

1. Generalized R-CNN Framework



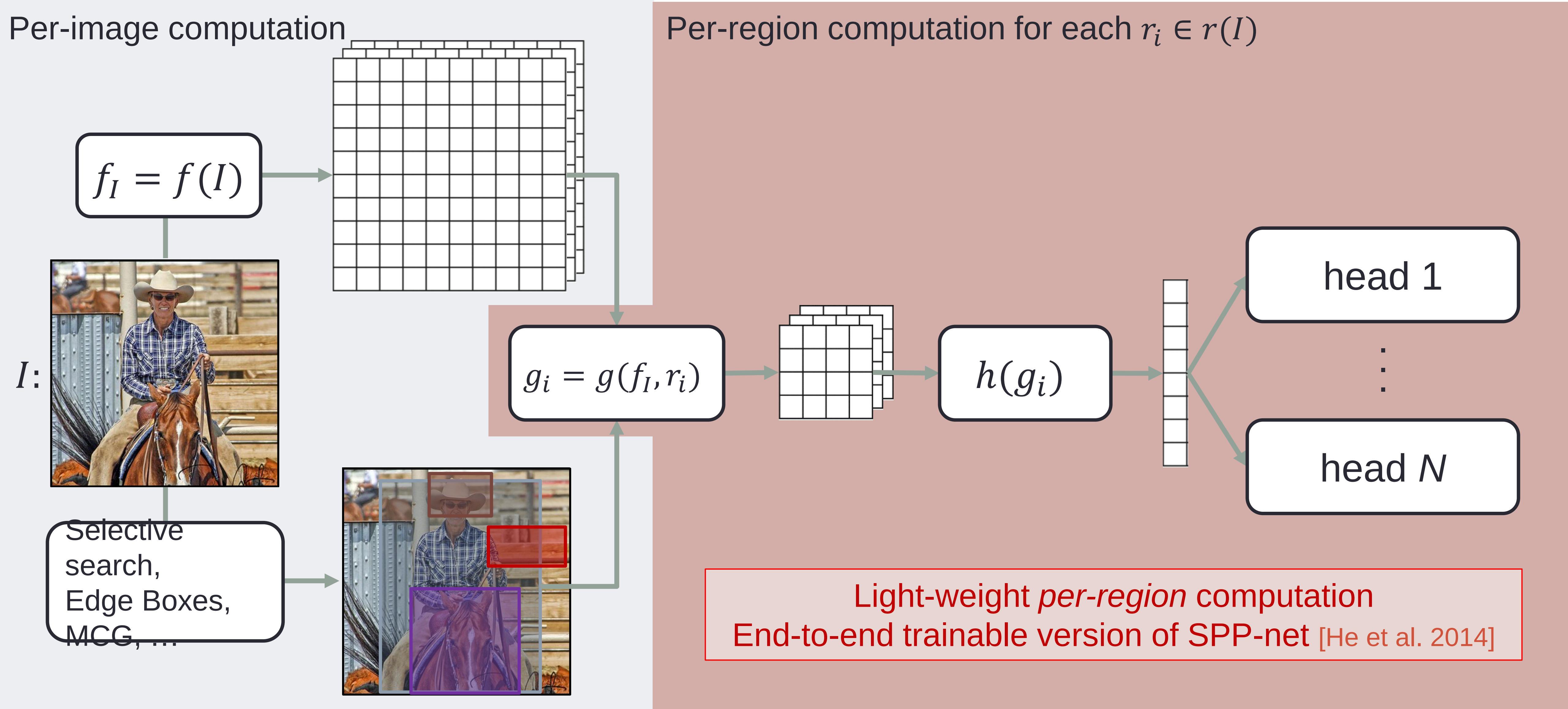
2. Fast R-CNN

"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

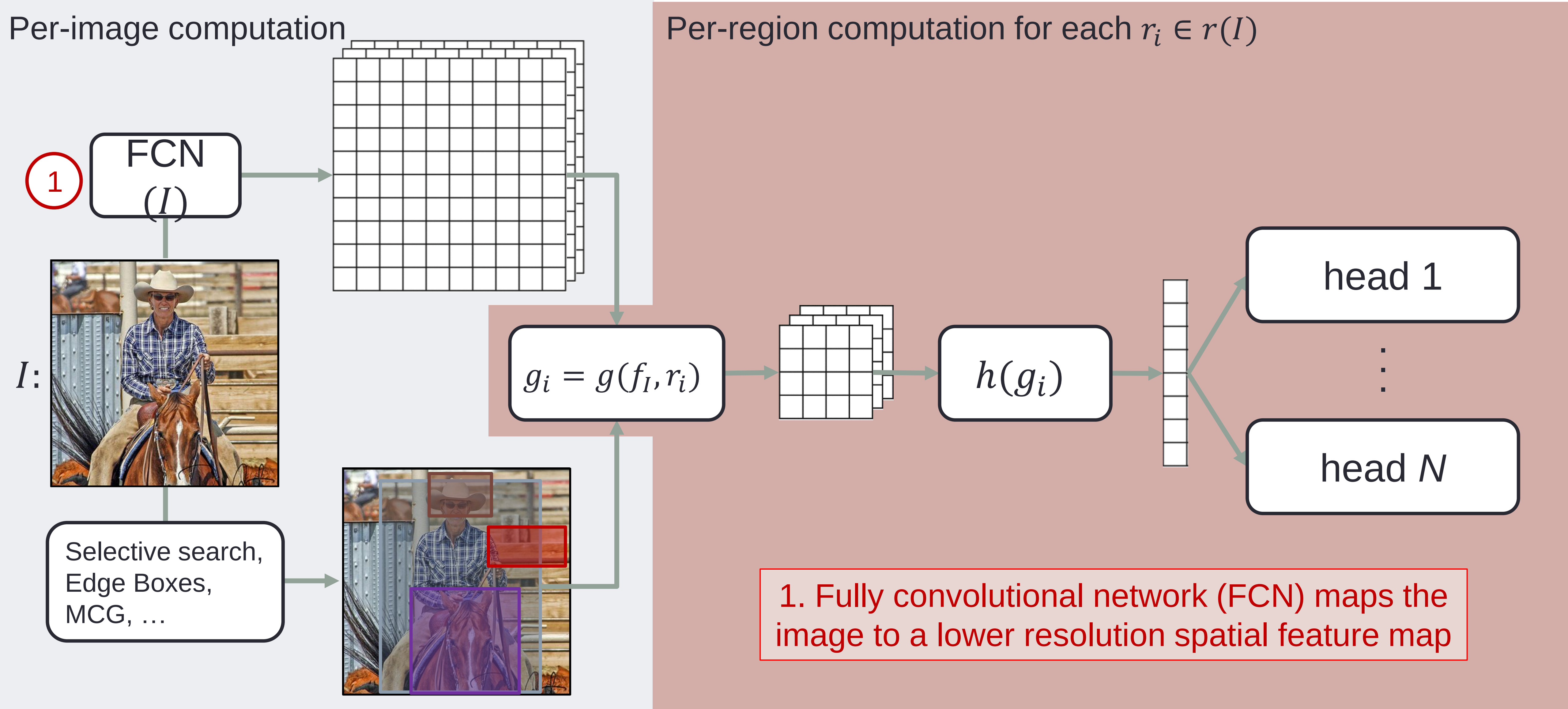
Girshick. Fast R-CNN. ICCV 2015.

2. Generalized R-CNN → Fast R-CNN



Girshick. Fast R-CNN. ICCV 2015.

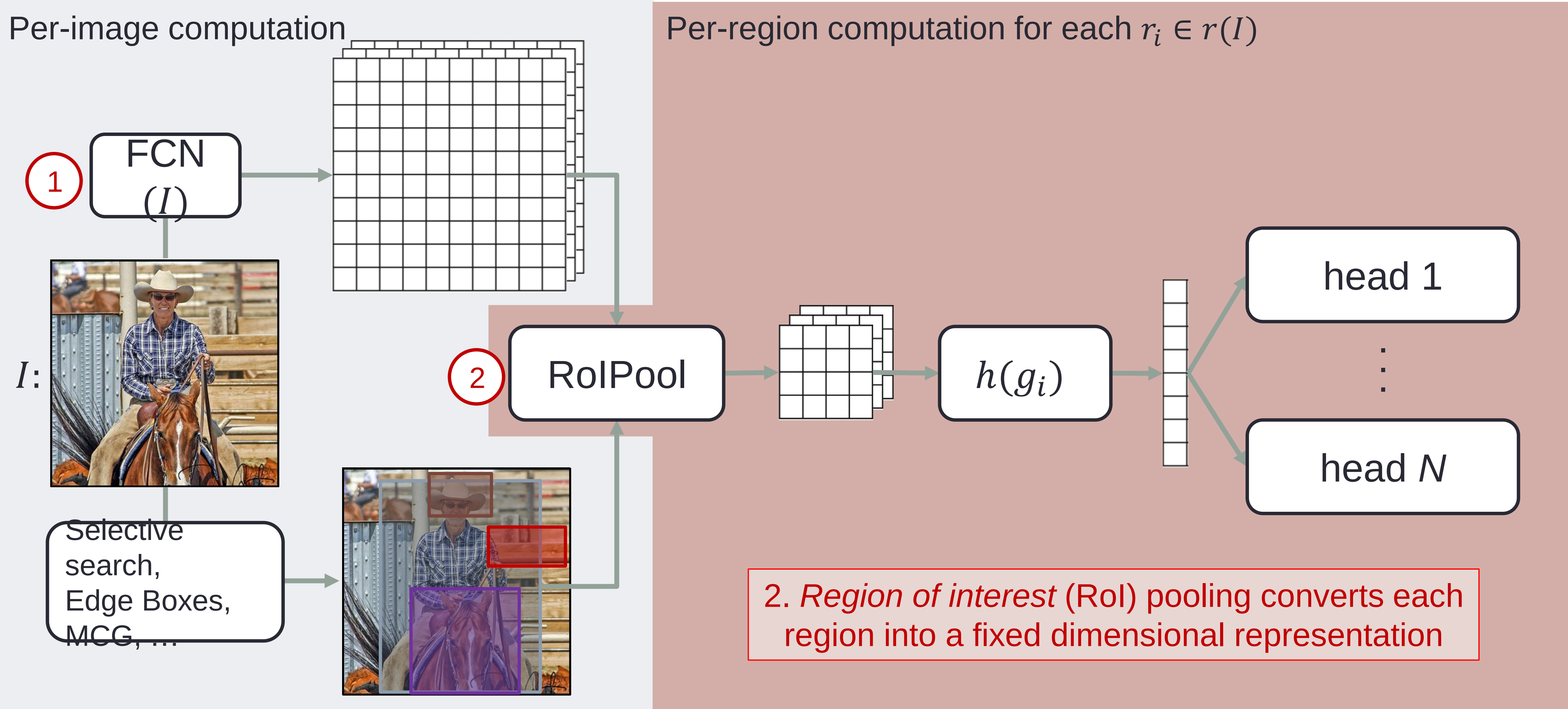
2. Generalized R-CNN → Fast R-CNN



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

Girshick. Fast R-CNN. ICCV 2015.

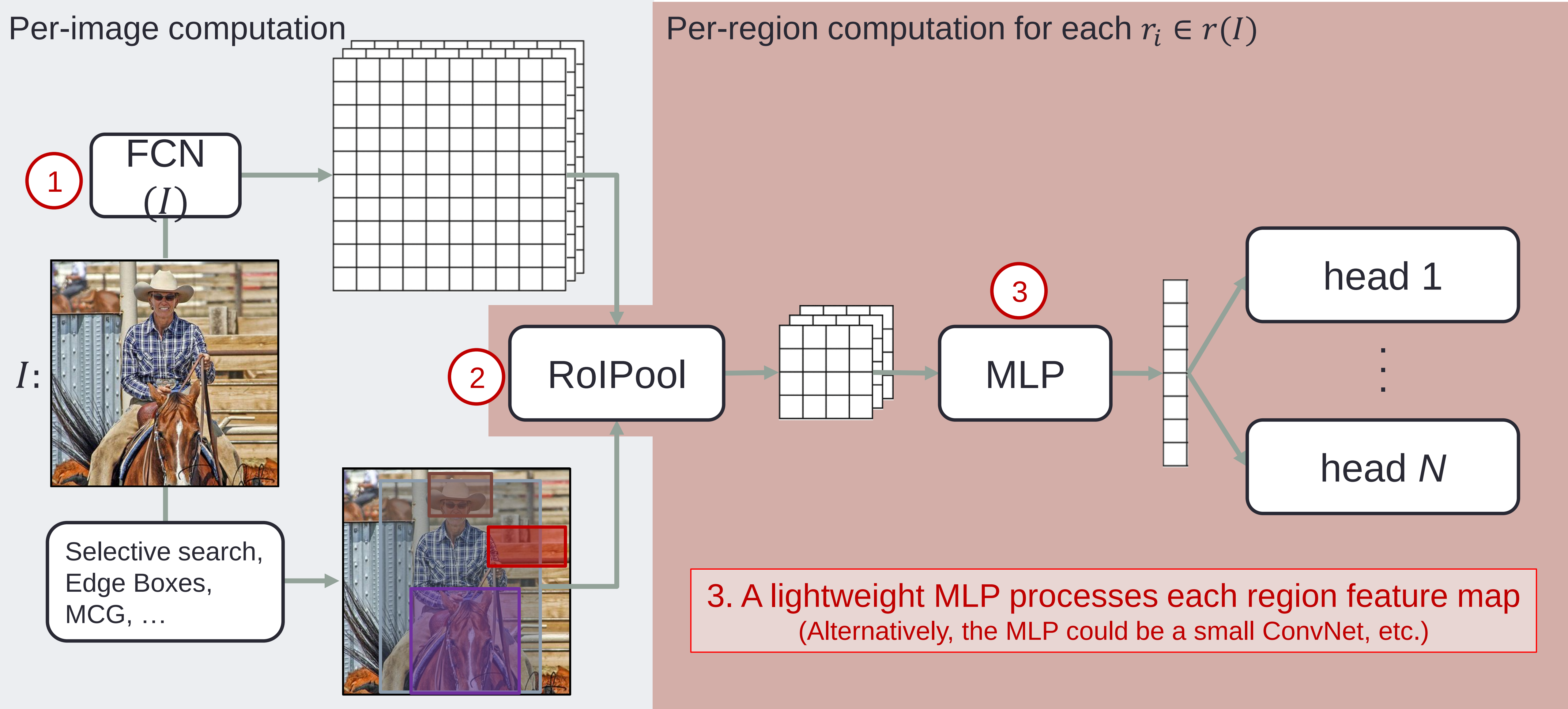
2. Generalized R-CNN → Fast R-CNN



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

Girshick. Fast R-CNN. ICCV 2015.

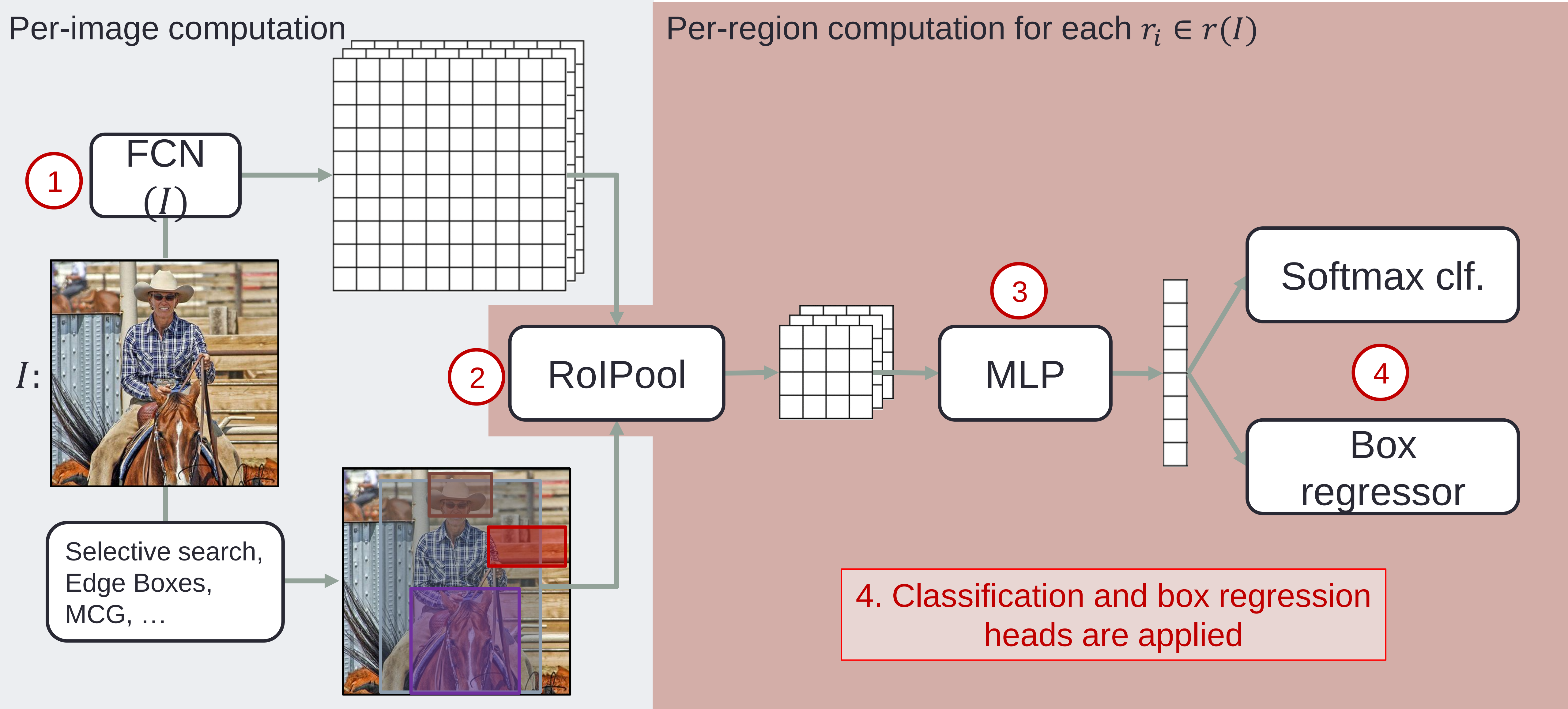
2. Generalized R-CNN → Fast R-CNN



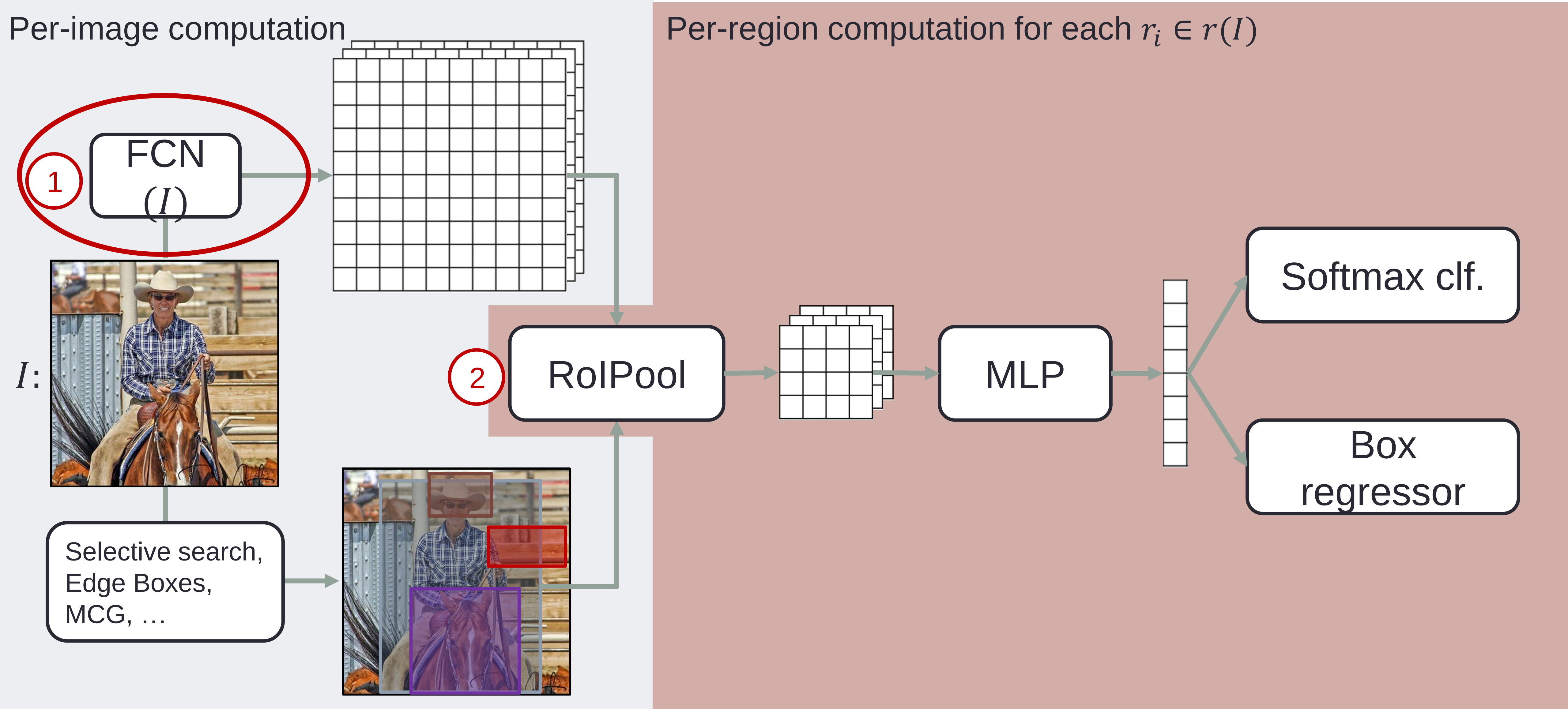
3. A lightweight MLP processes each region feature map
(Alternatively, the MLP could be a small ConvNet, etc.)

Girshick. Fast R-CNN. ICCV 2015.

2. Generalized R-CNN → Fast R-CNN



2. 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

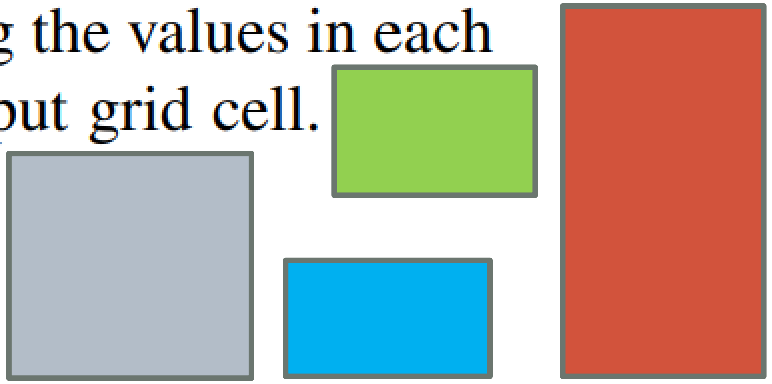
2. Fast R-CNN-RoI Pooling

2 RoIPool

2. Fast R-CNN-RoI Pooling

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., H=7 x W=7)

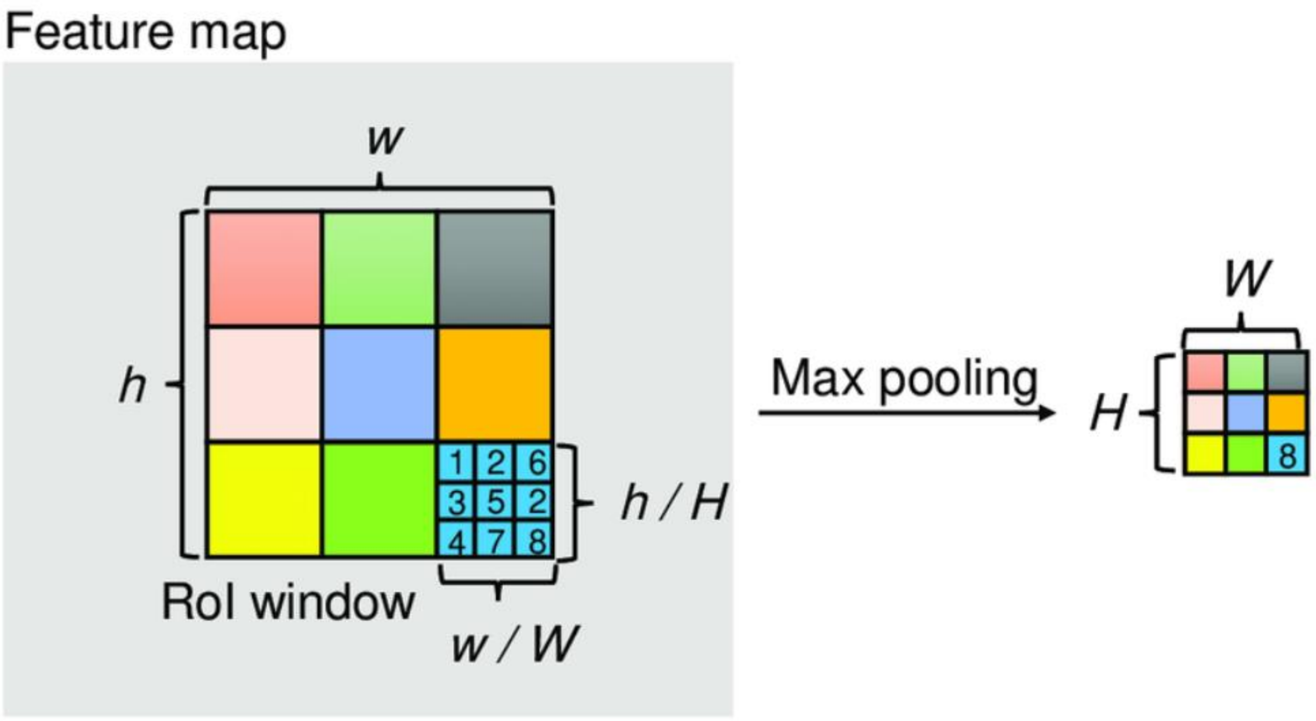
RoI max pooling works by dividing the $h \times w$ RoI window into an $H \times W$ grid of sub-windows of approximate size $h/H \times w/W$ and then max-pooling the values in each sub-window into the corresponding output grid cell.



Girshick. Fast R-CNN. ICCV 2015.

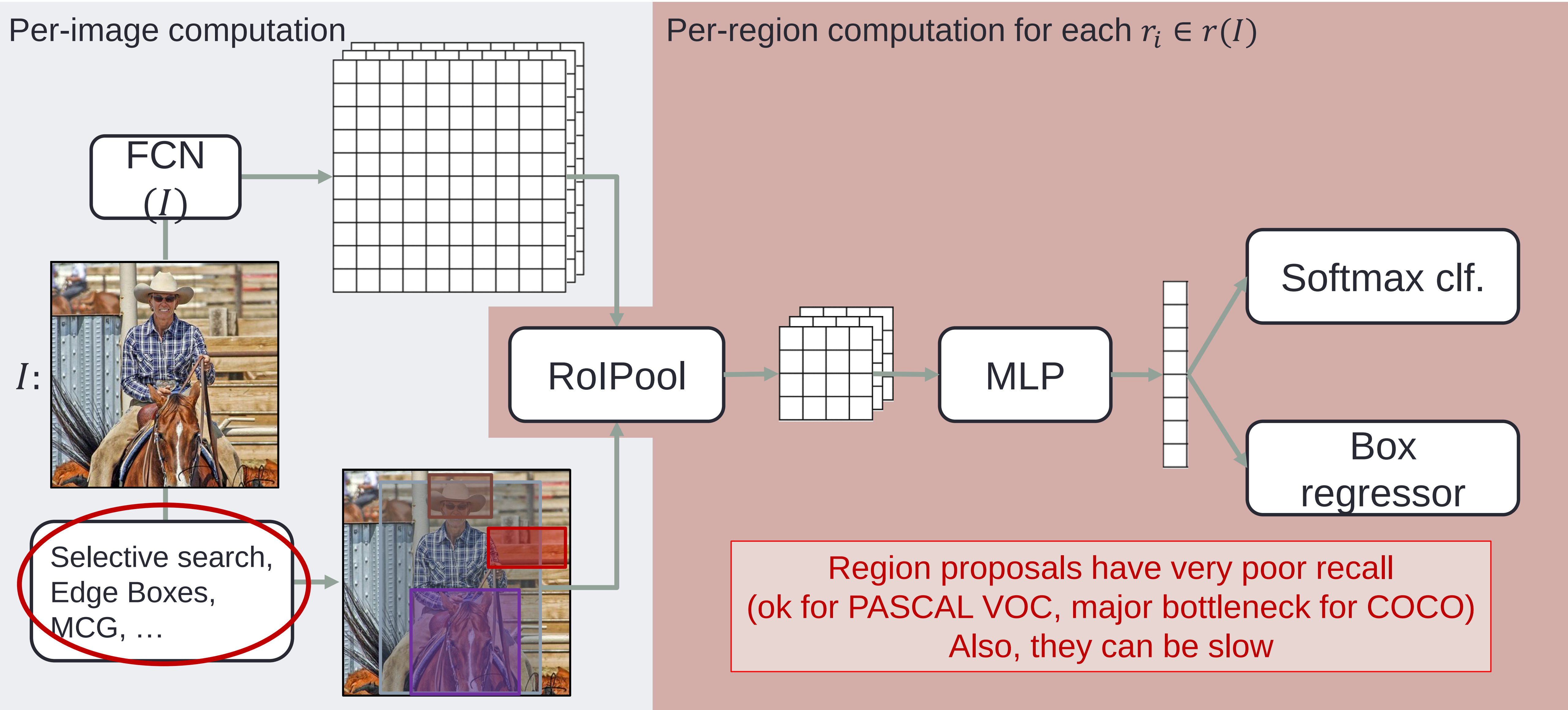
2. Fast R-CNN-RoI Pooling

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., H=7 x W=7)



https://www.researchgate.net/figure/Illustration-of-the-RoI-pooling-operation_fig4_333521857

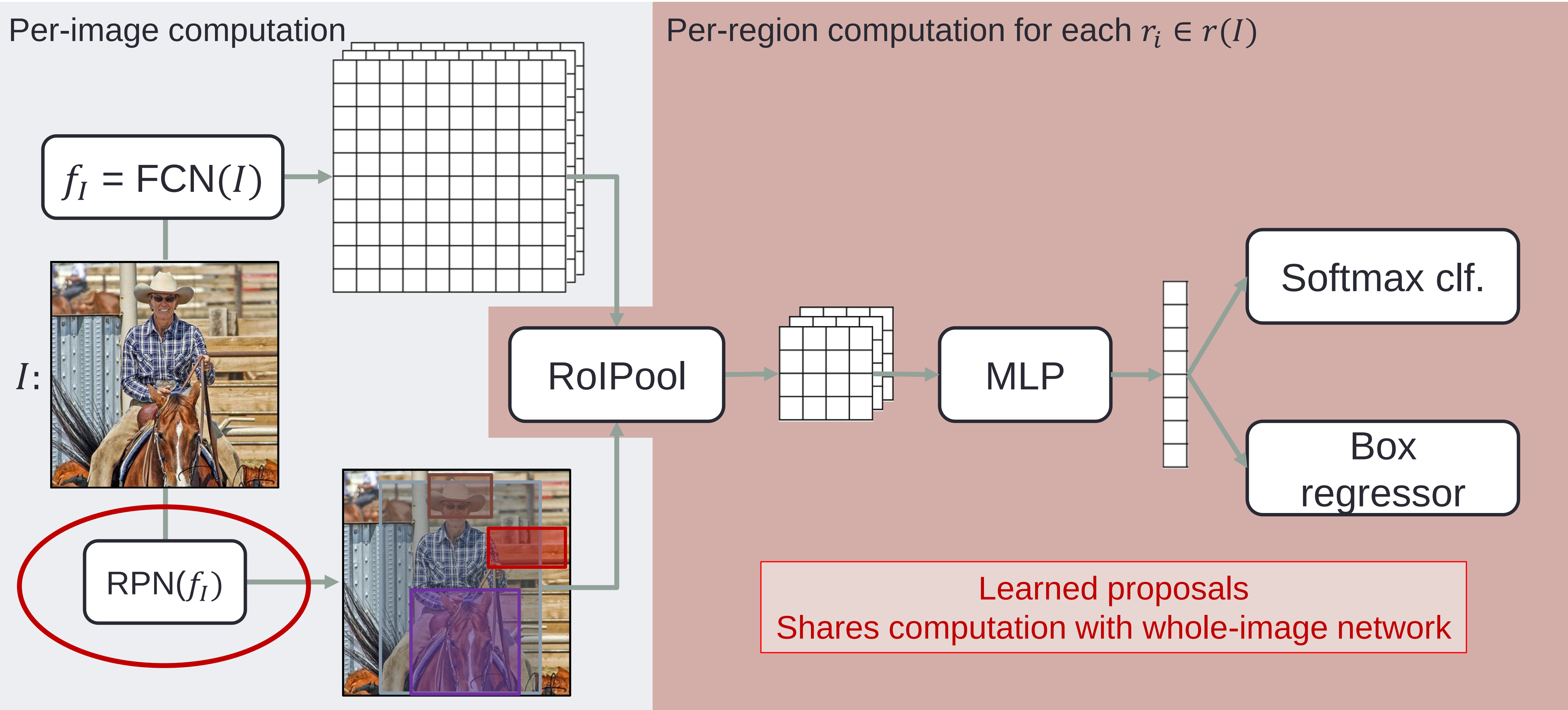
2. The Problem with Fast R-CNN



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

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

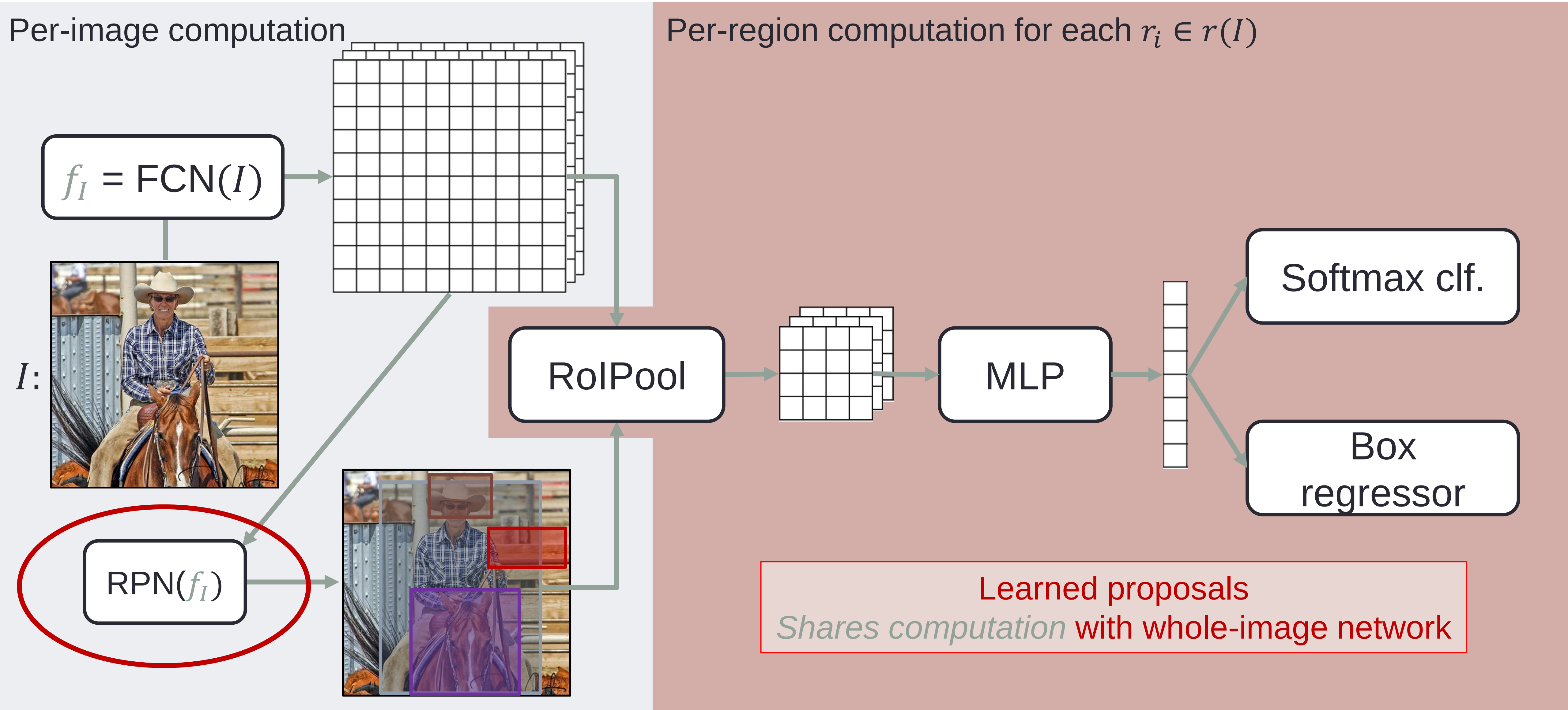
2. Faster R-CNN



Learned proposals
Shares computation with whole-image network

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

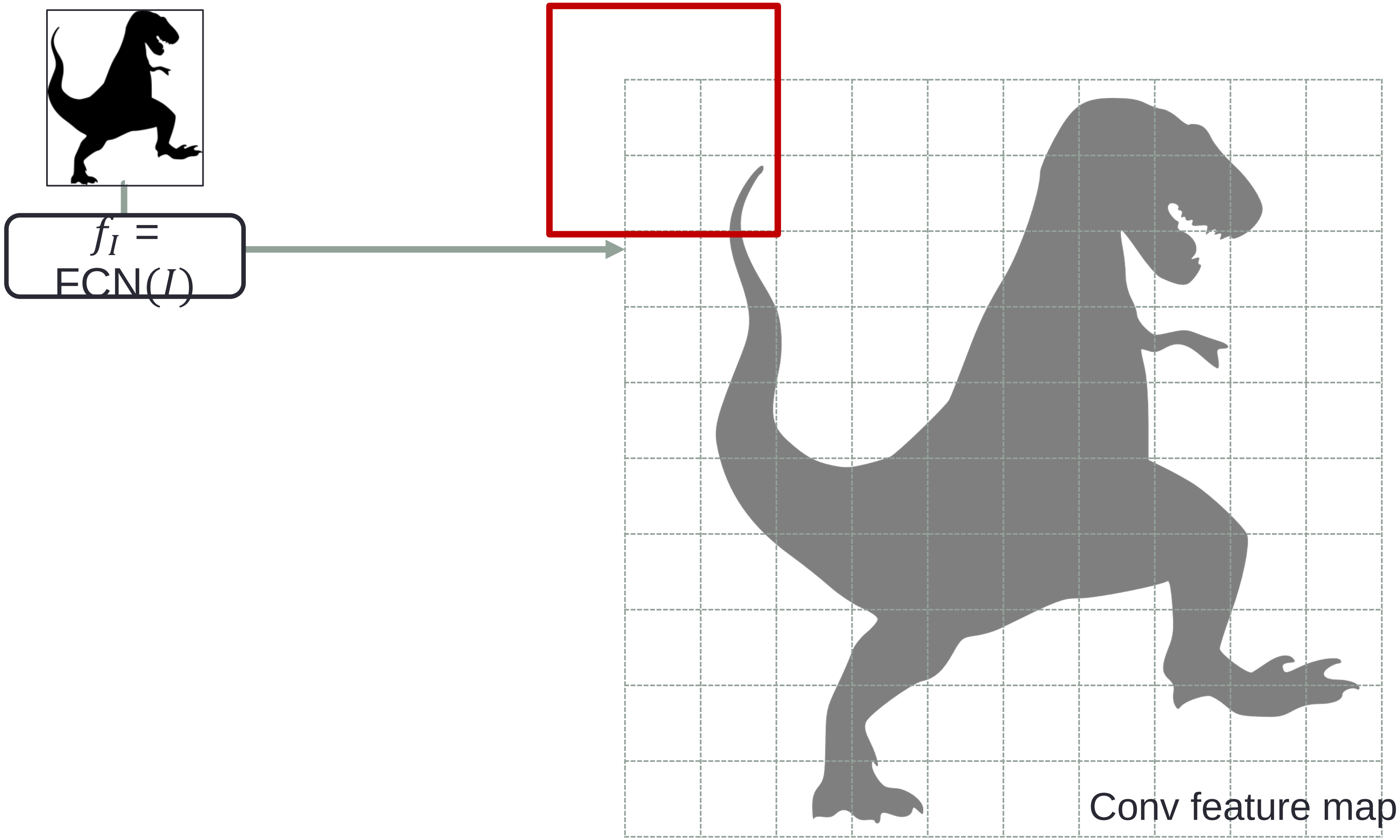
3. Faster R-CNN



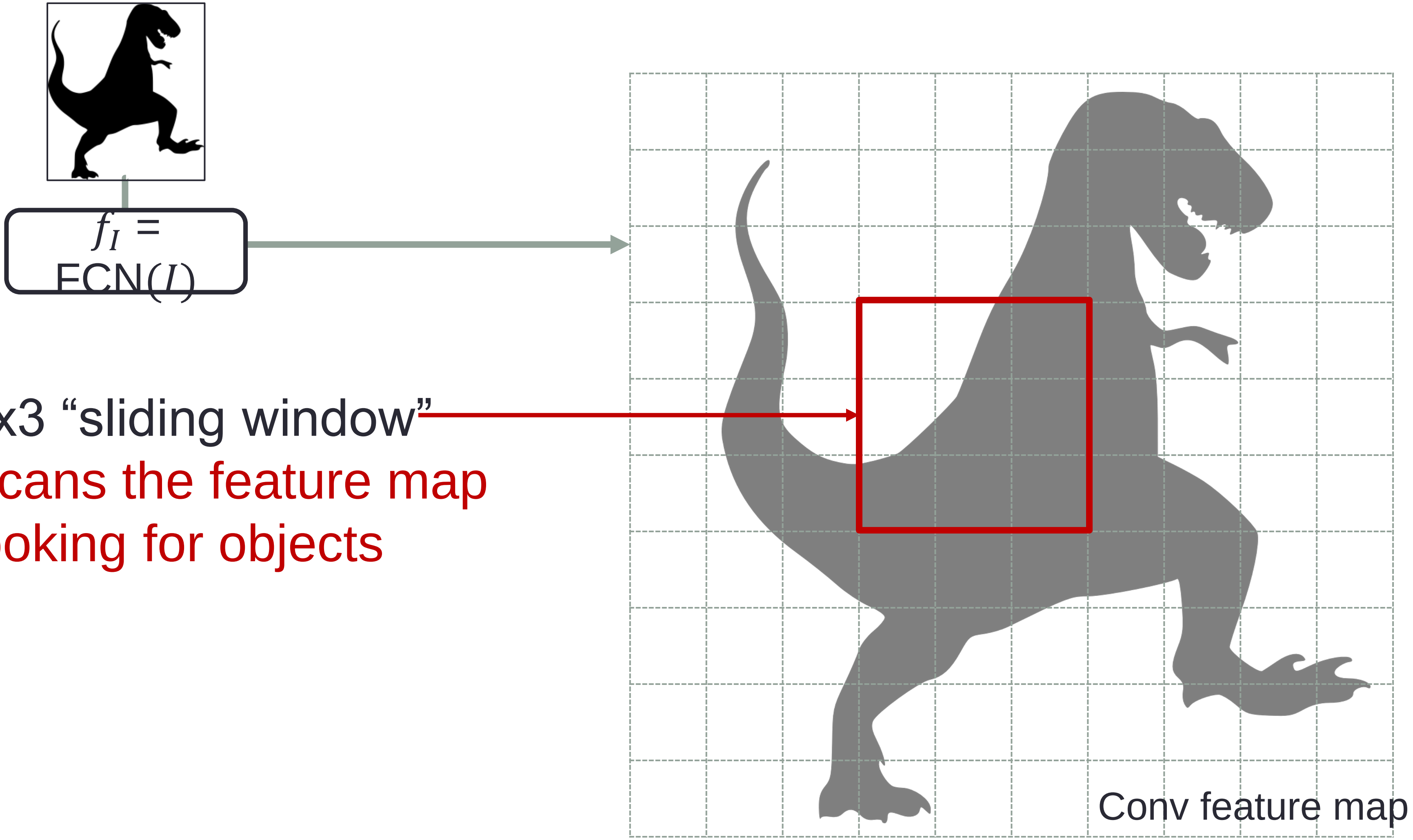
Learned proposals

Shares computation with whole-image network

3. RPN: Region Proposal Network



3. RPN: Region Proposal Network

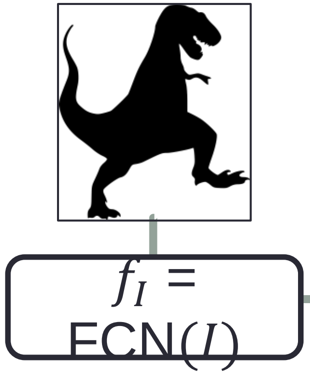


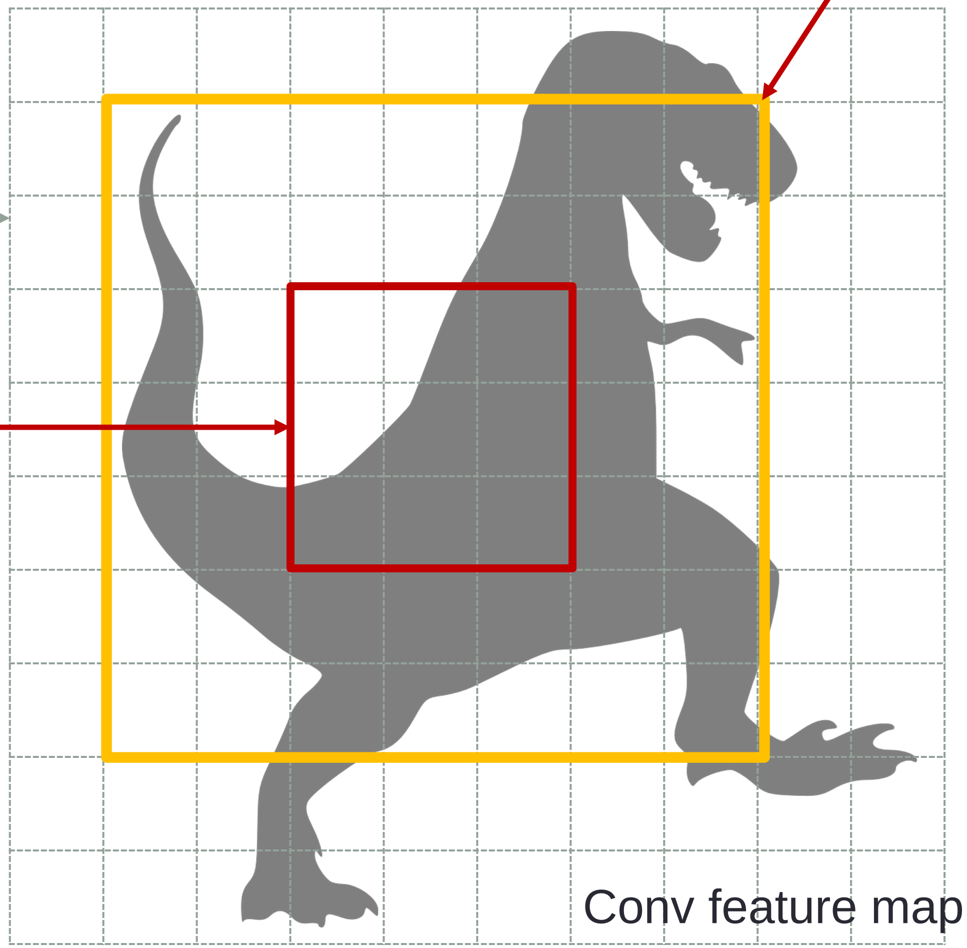
3x3 "sliding window"

Scans the feature map looking for objects

3. RPN: Anchor Box

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



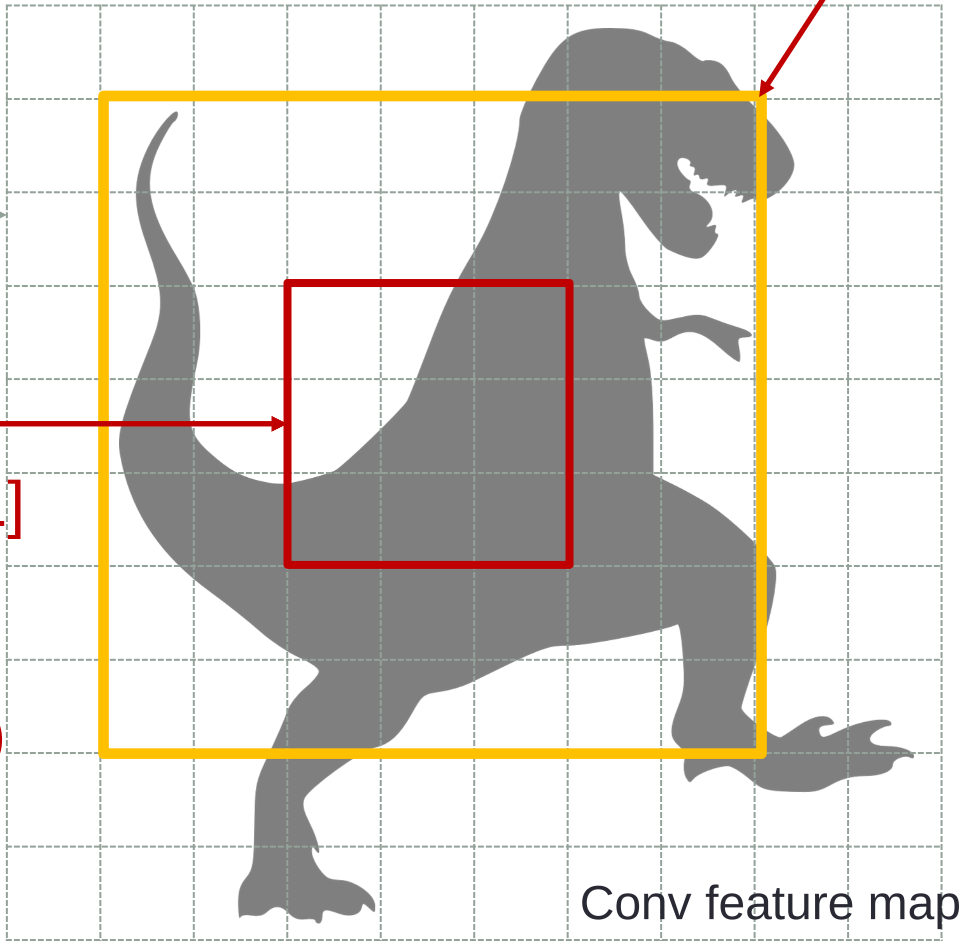


3x3 "sliding window"

Scans the feature map looking for objects

3. RPN: Anchor Box

f_I = FCN(I)



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

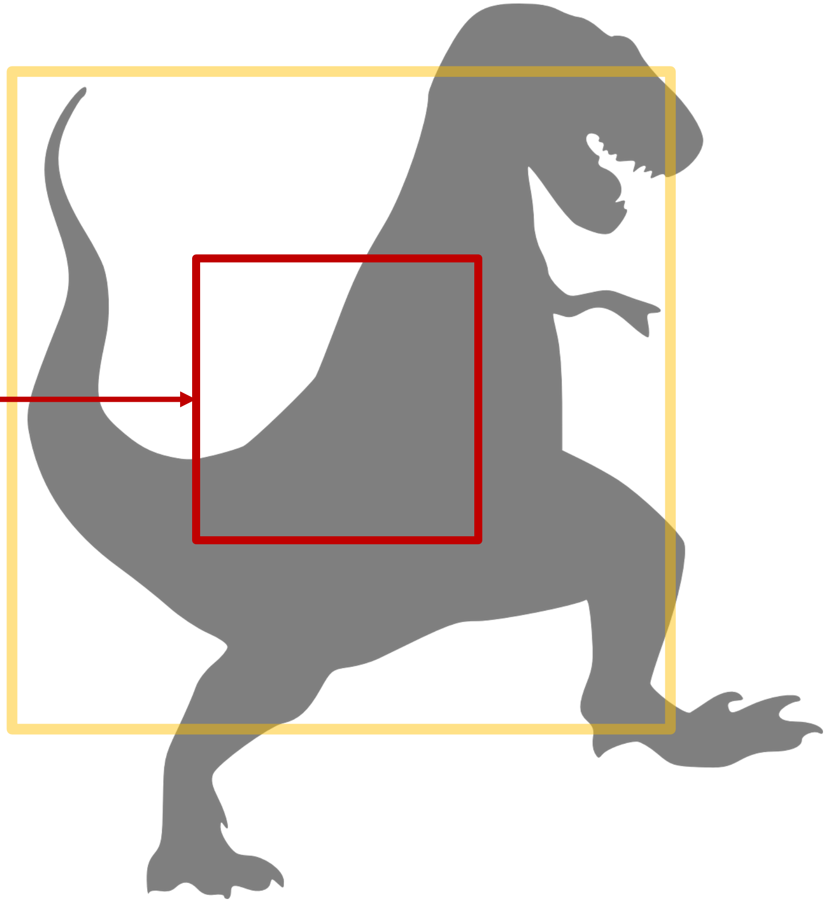
3x3 “sliding window”

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

3. RPN: Prediction (on object)

Objectness score

P(object) = 0.94



3x3 “sliding window”

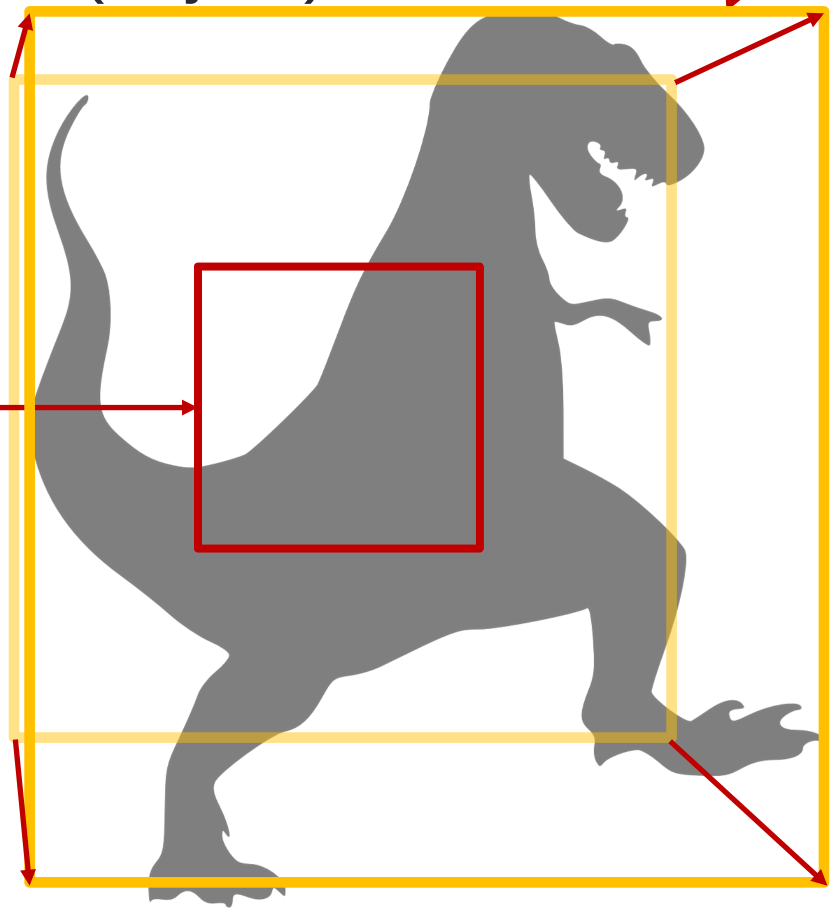
- Objectness classifier [0, 1]

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

3. 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)

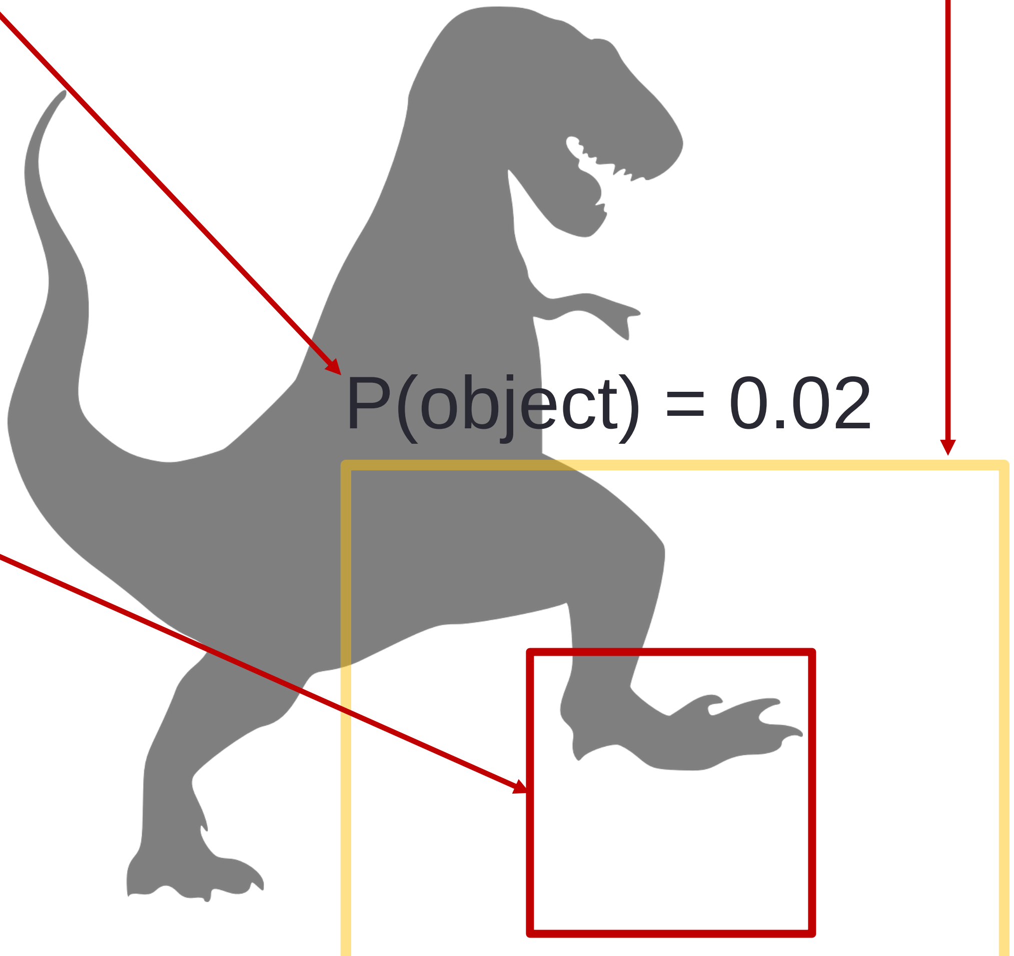
3. RPN: Prediction (off object)

Anchor box: transformed by box regressor

Objectness score

3x3 “sliding window”

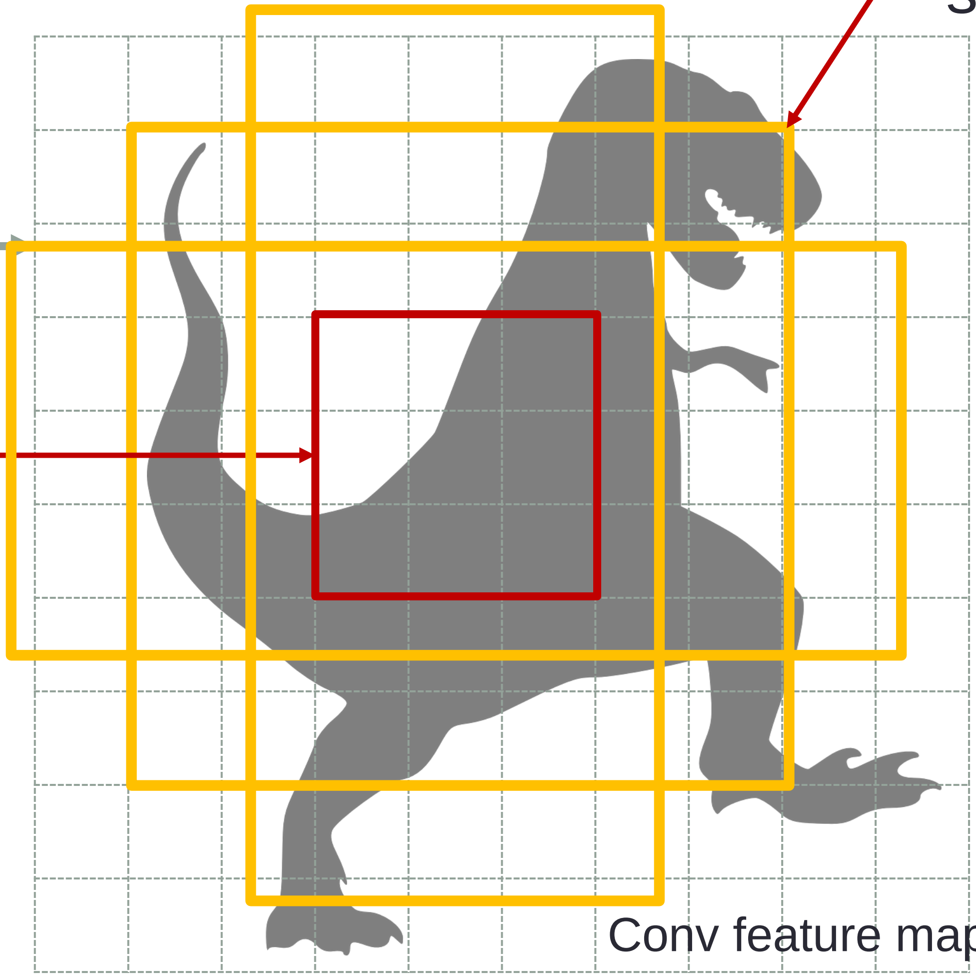
- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)



3. RPN: Multiple Anchors

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

$f_I = \mathrm{FCN}(I)$

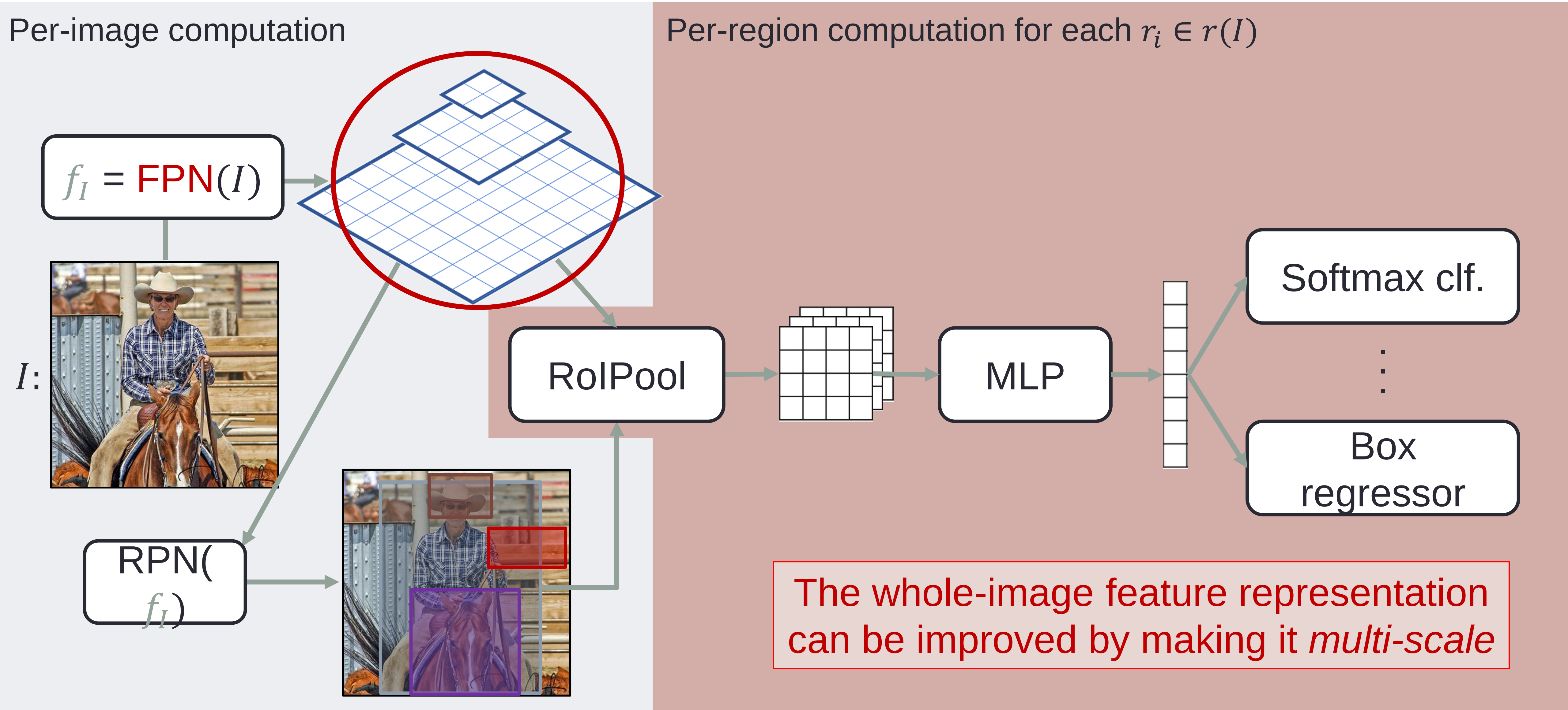


3x3 “sliding window”

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

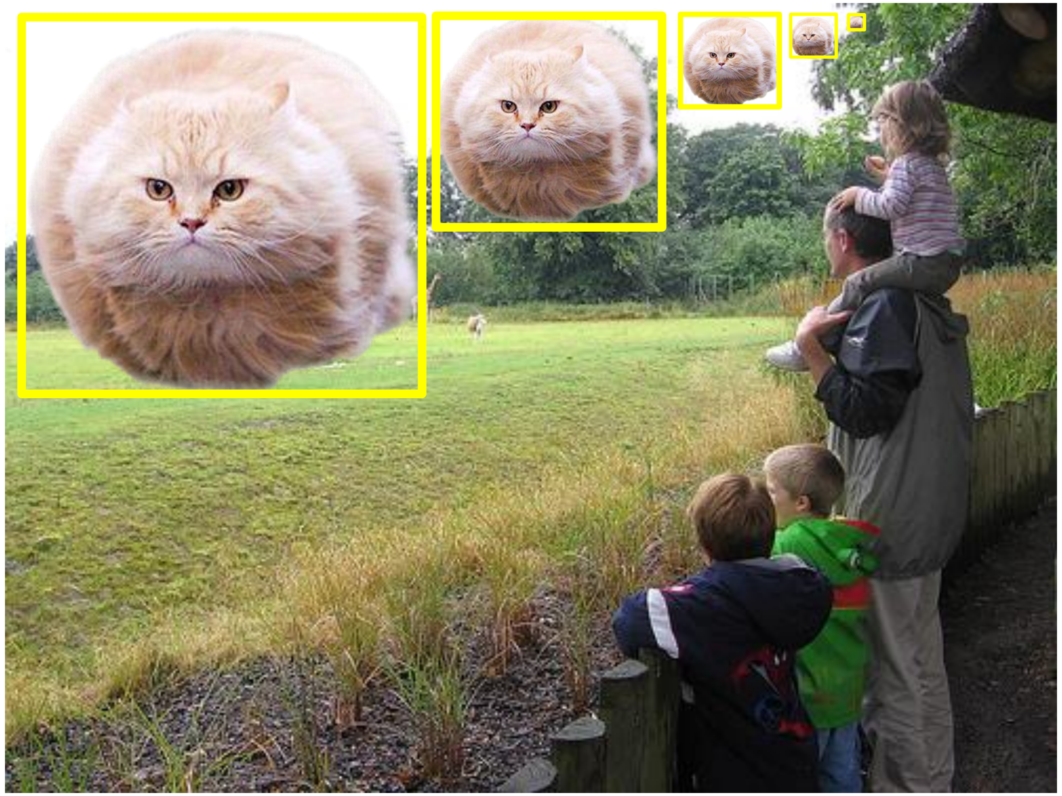
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

4. Faster R-CNN with a Feature Pyramid Network



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

4. FPN: Improving Scale Invariance and Equivariance



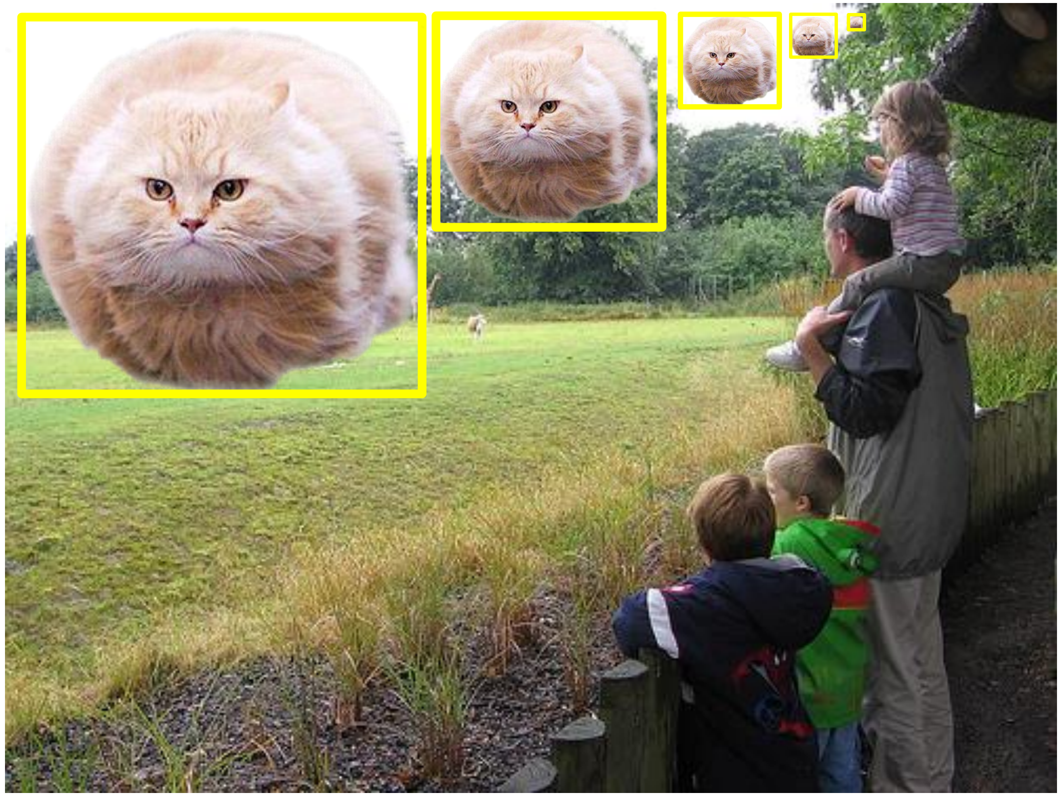
Detectors need to

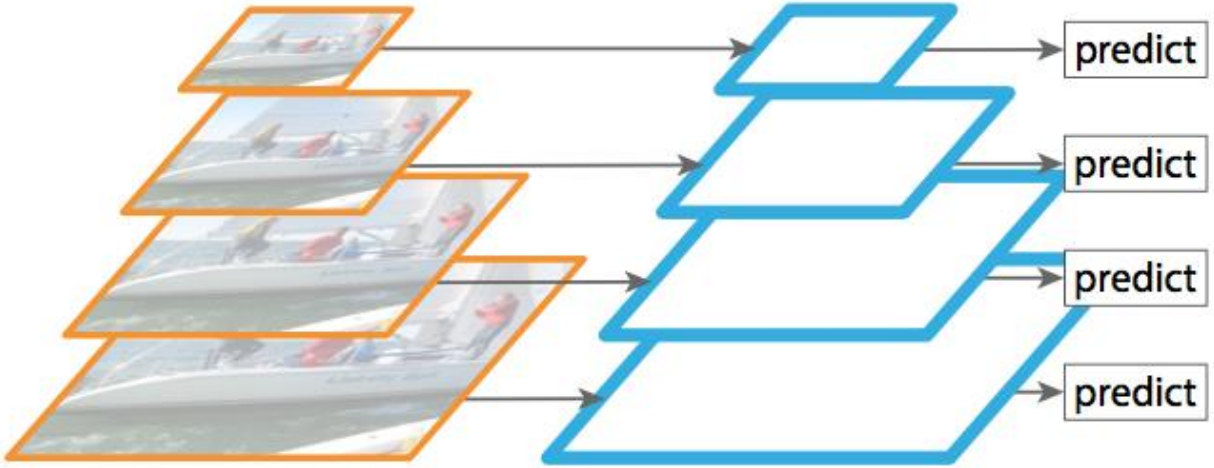
1. classify and
2. localize

objects over a wide range of scales

FPN improves this ability

Strategy 1: Image Pyramid



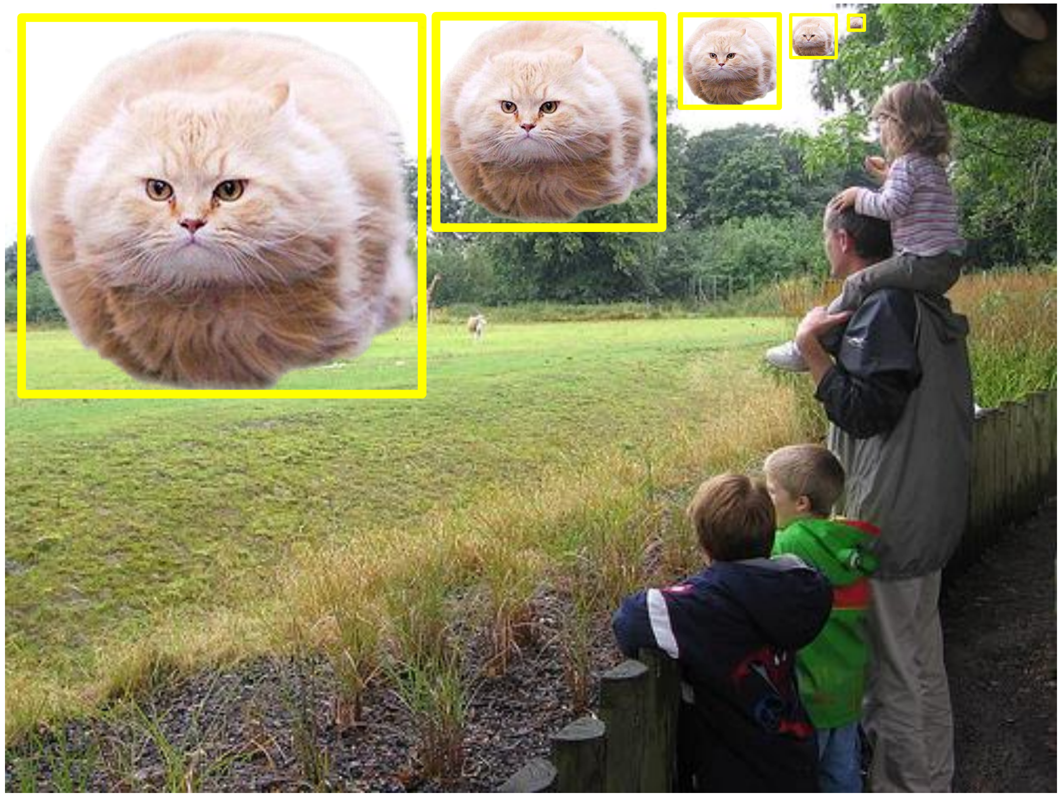


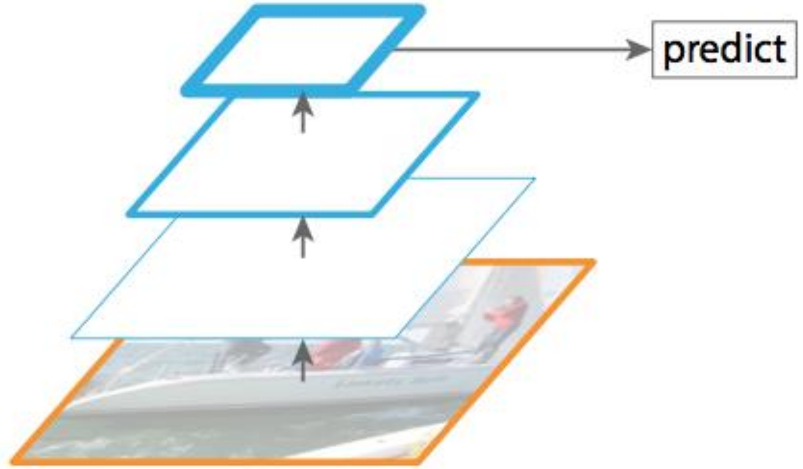
(a) Featurized image pyramid

Standard solution – *slow!*

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

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



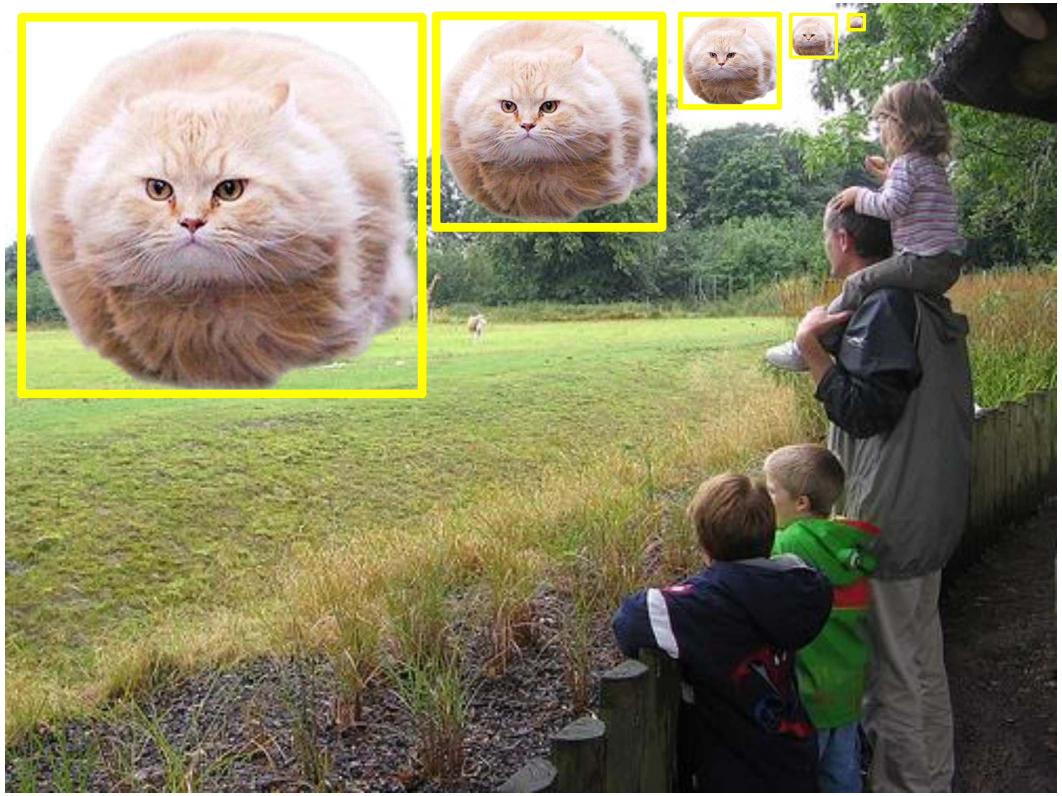


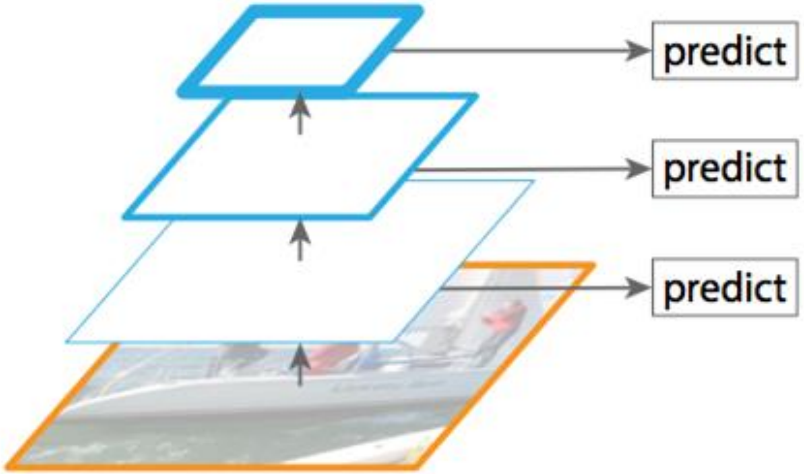
(b) Single feature map

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

(E.g., Fast/er R-CNN, YOLO, …)

Strategy 3: Naïve In-network Pyramid



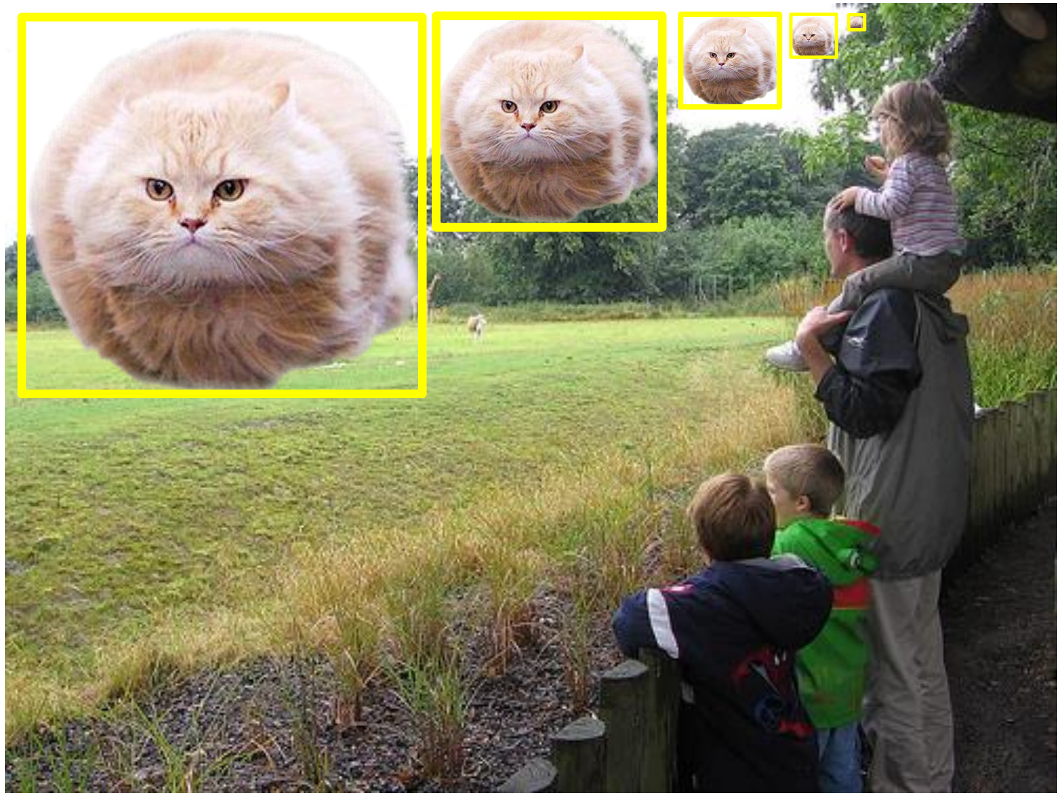


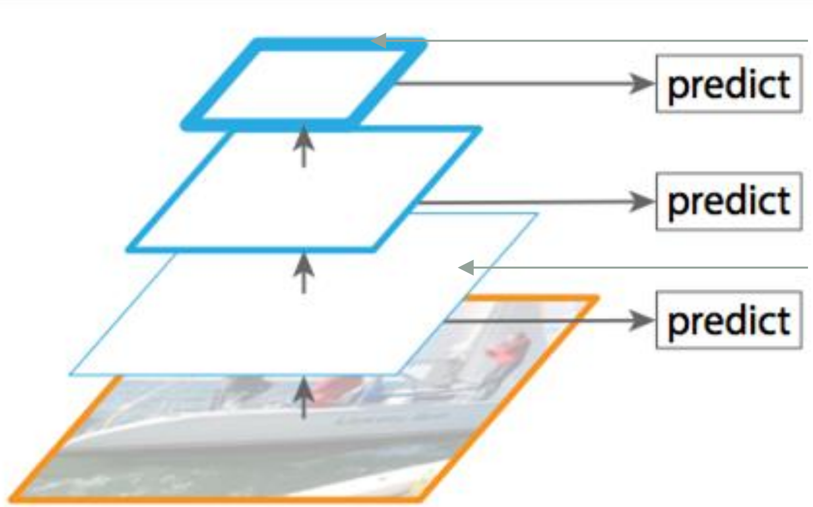
(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 3: Naïve In-network Pyramid





(c) Pyramidal feature hierarchy

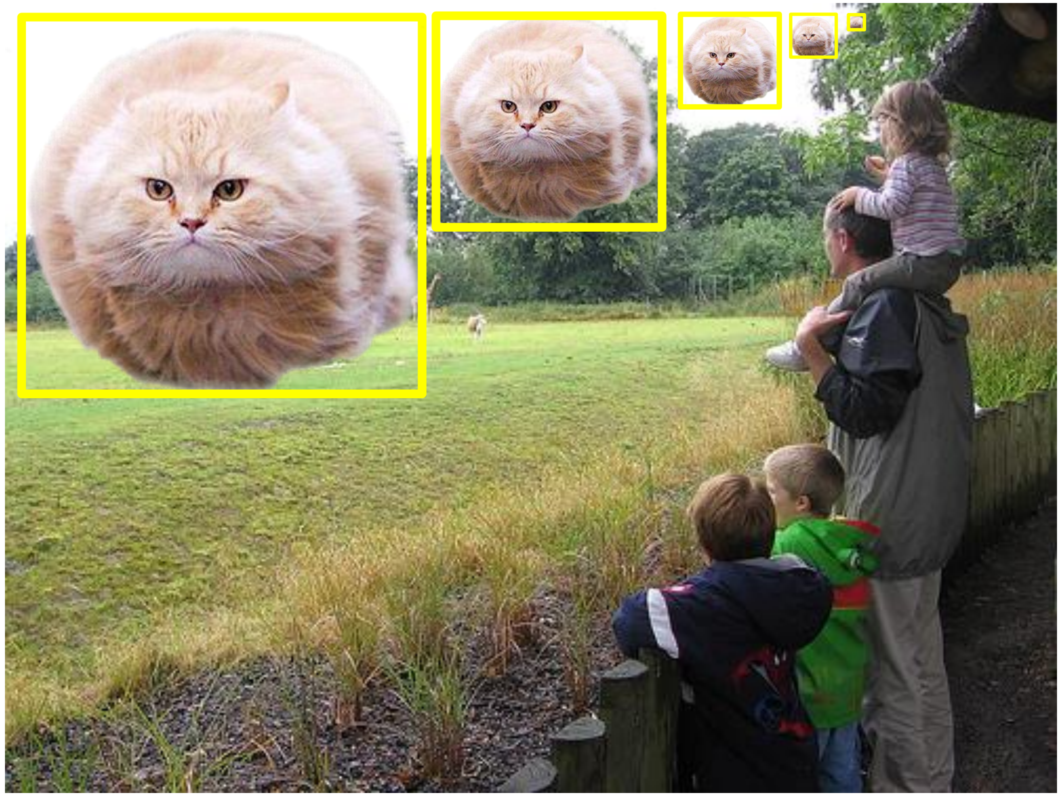
Low resolution,
Strong features

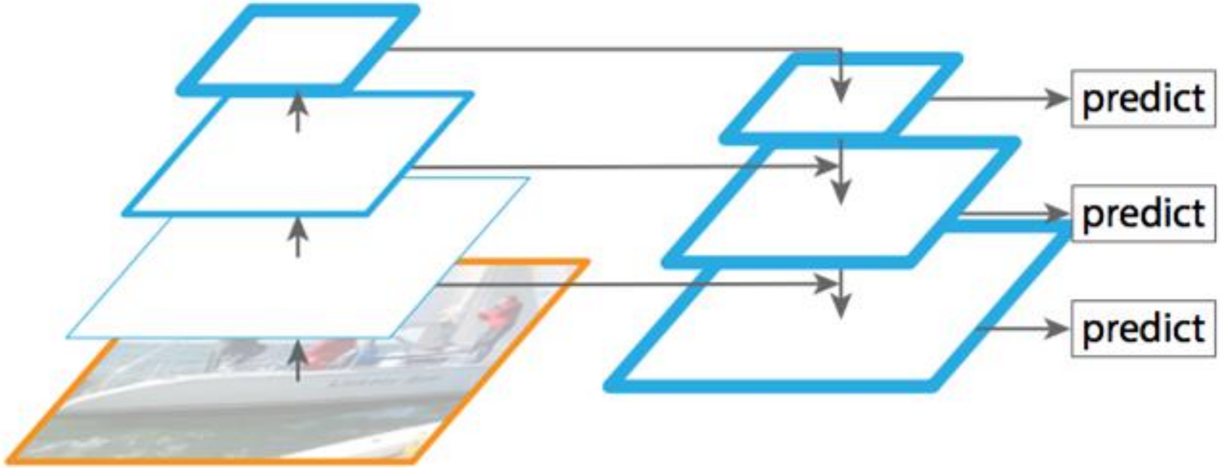
High resolution,
Weak features

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 4: Feature Pyramid Network



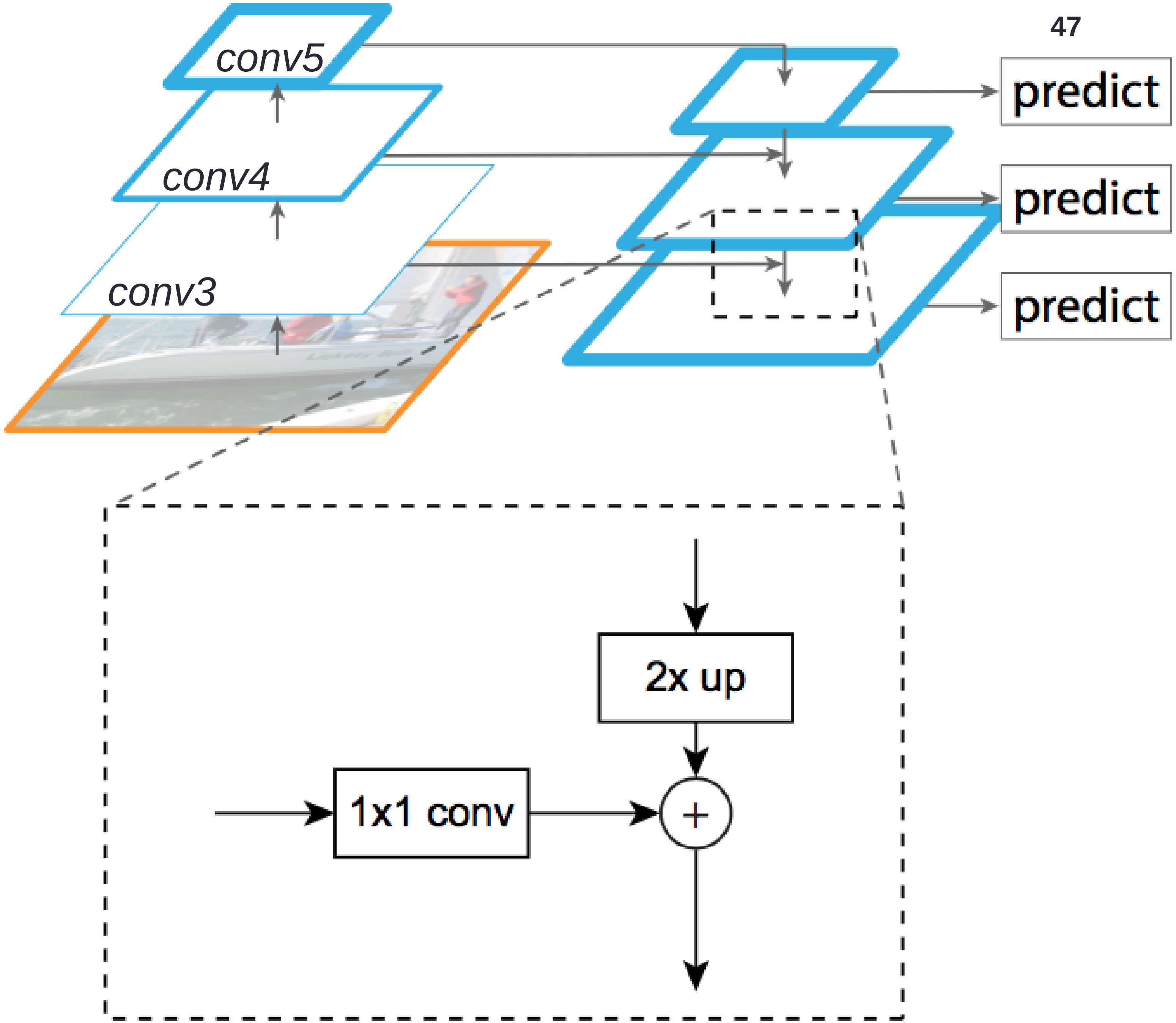


(d) Feature Pyramid Network

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

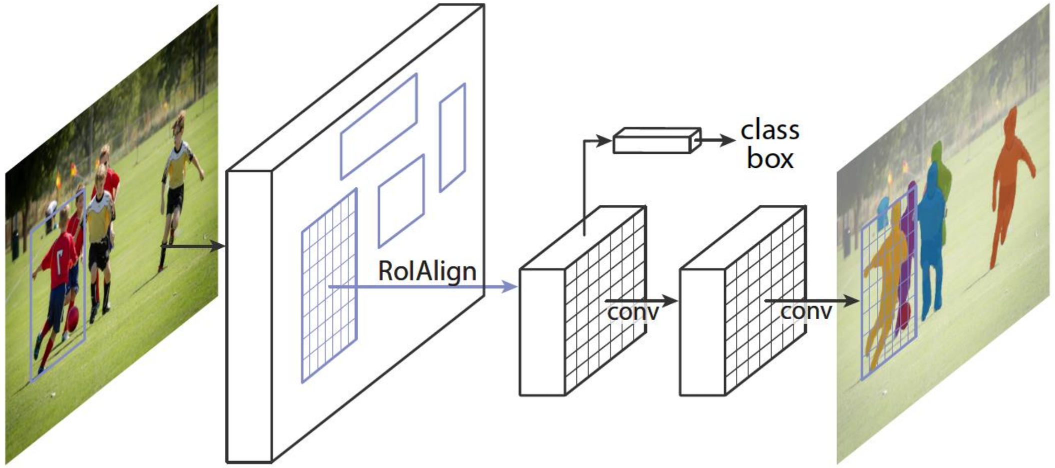
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

FPN: Light-weight, Top-down Refinement Module

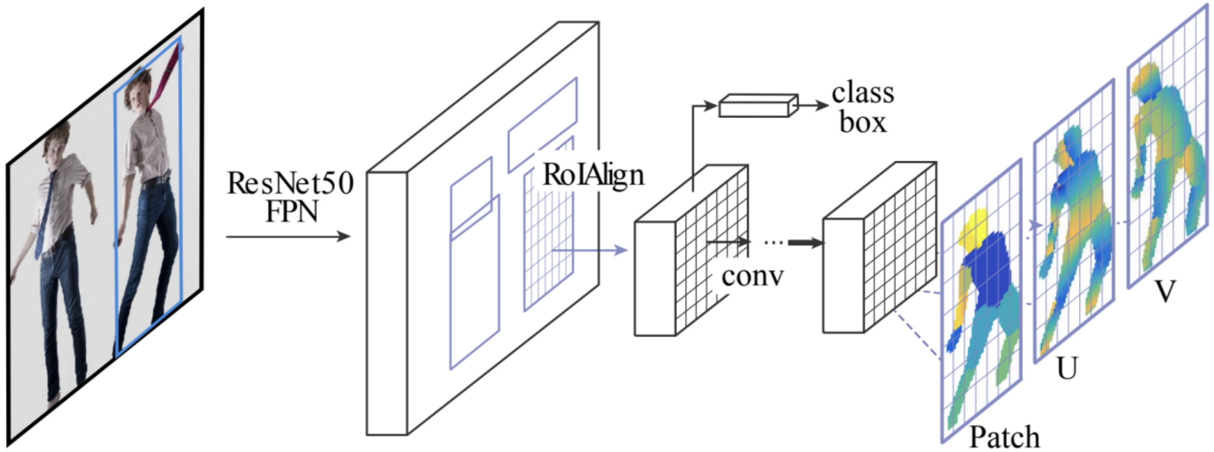


Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

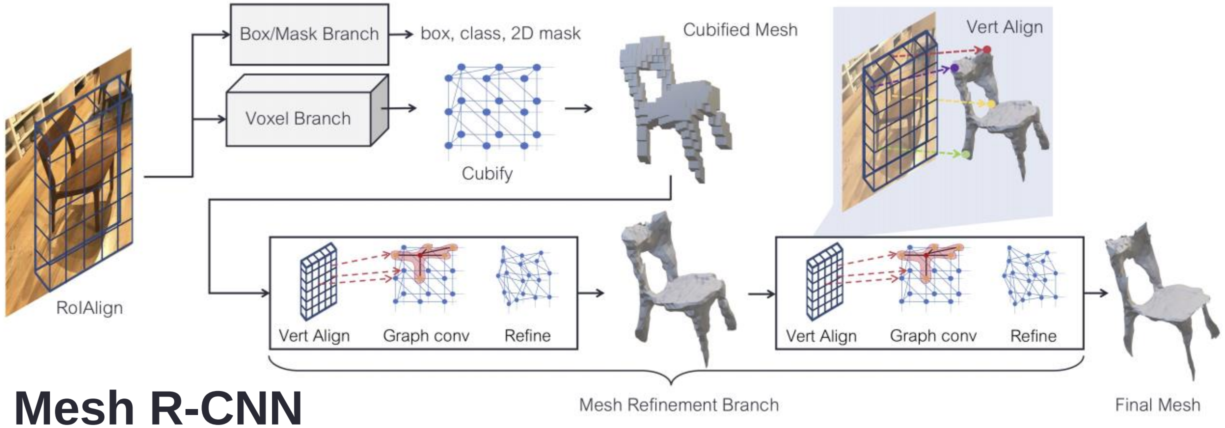
5. Generalized R-CNN: Adding More Heads



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

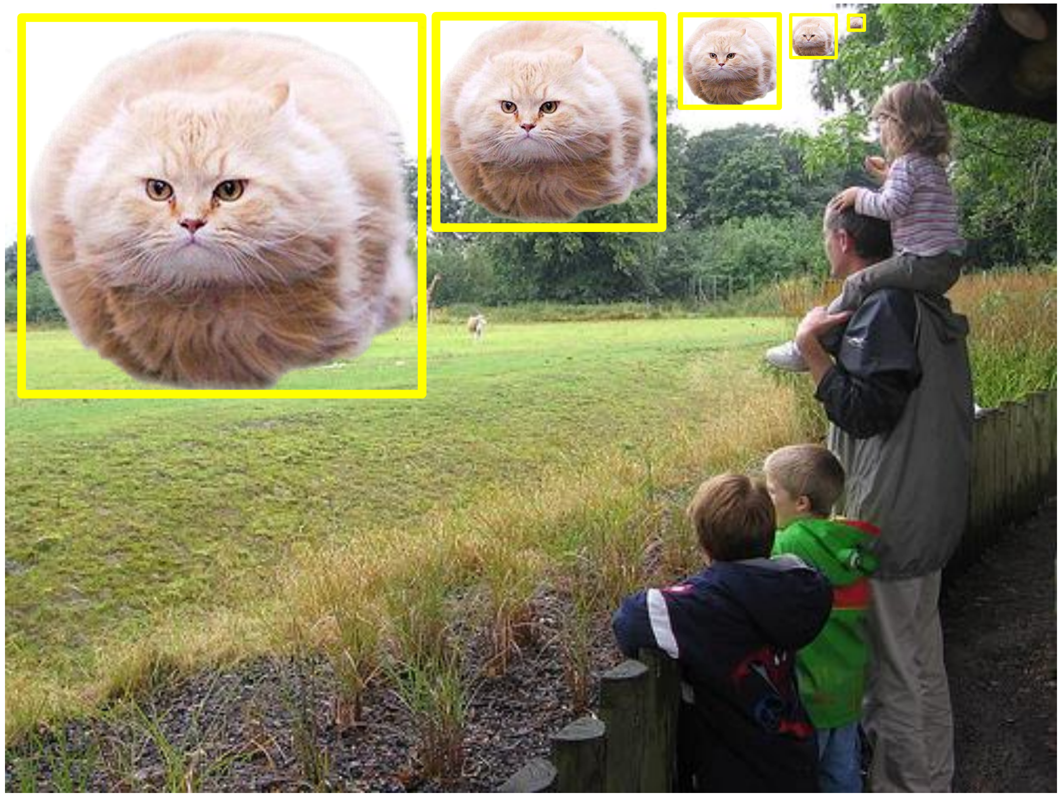


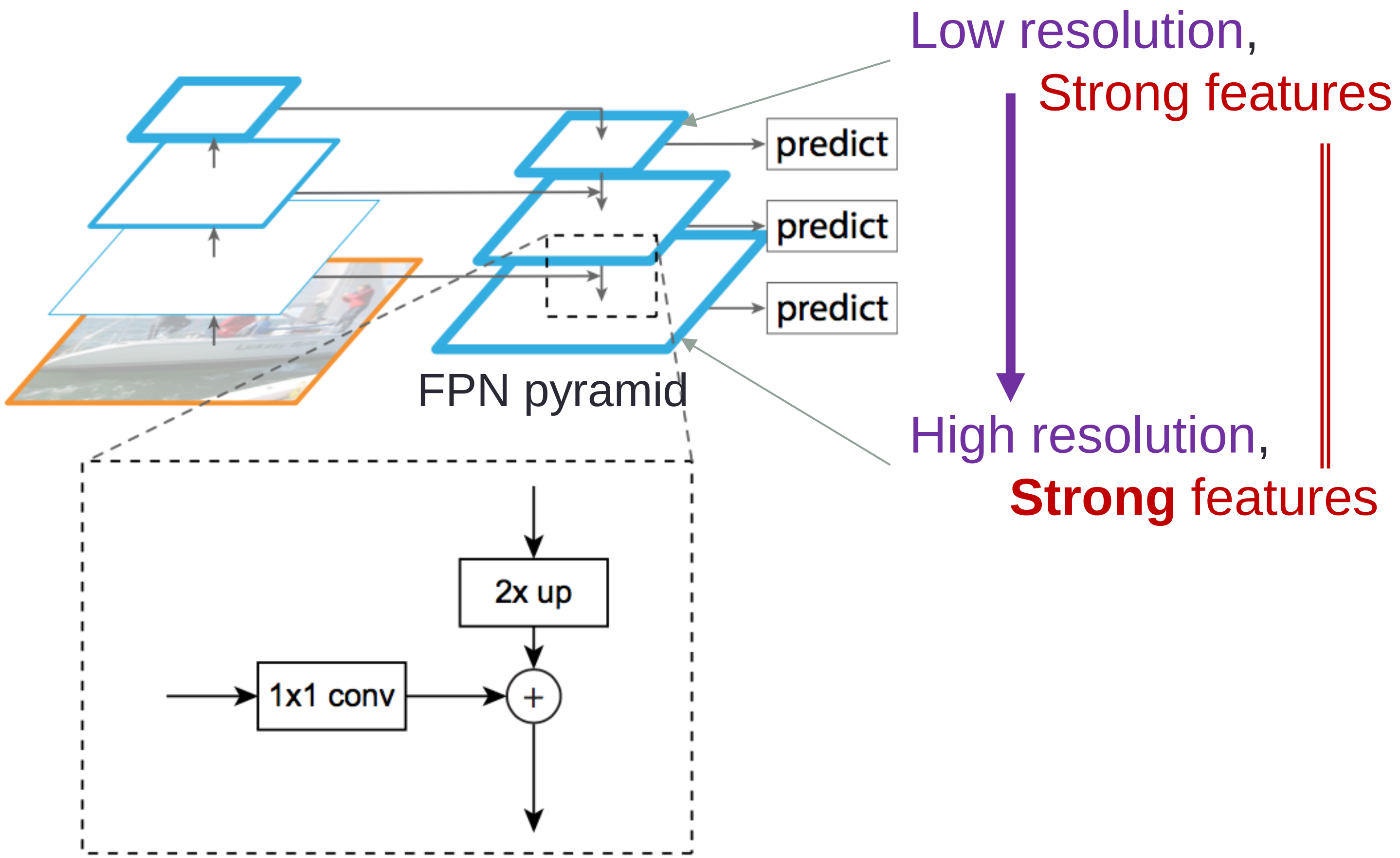
DensePose
[Güler, Neverova, Kokkinos]



Mesh R-CNN
[Gkioxari, Malik, Johnson arXiv 2019]

No Compromise on Feature Quality, Still Fast

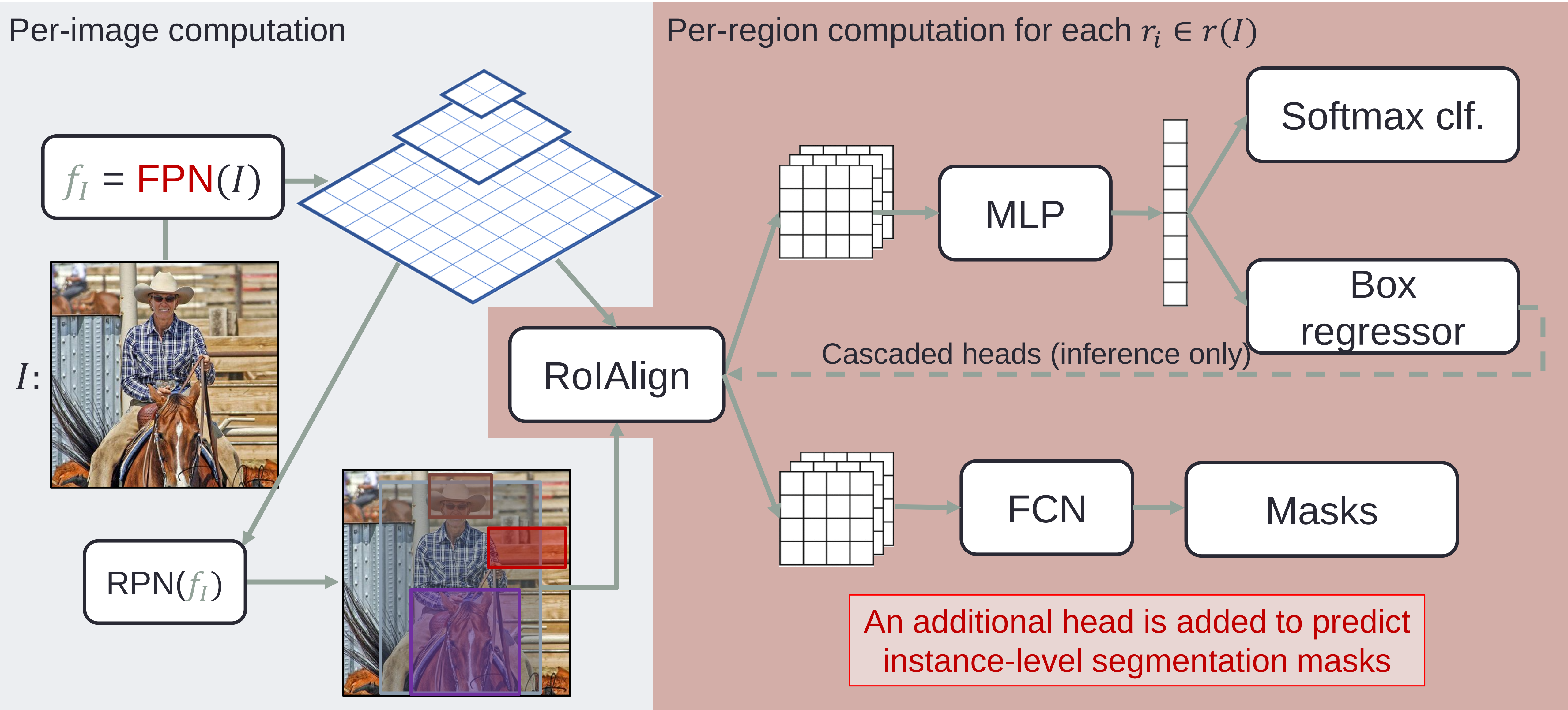




Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017. See also: Shrivastava’s TDM.

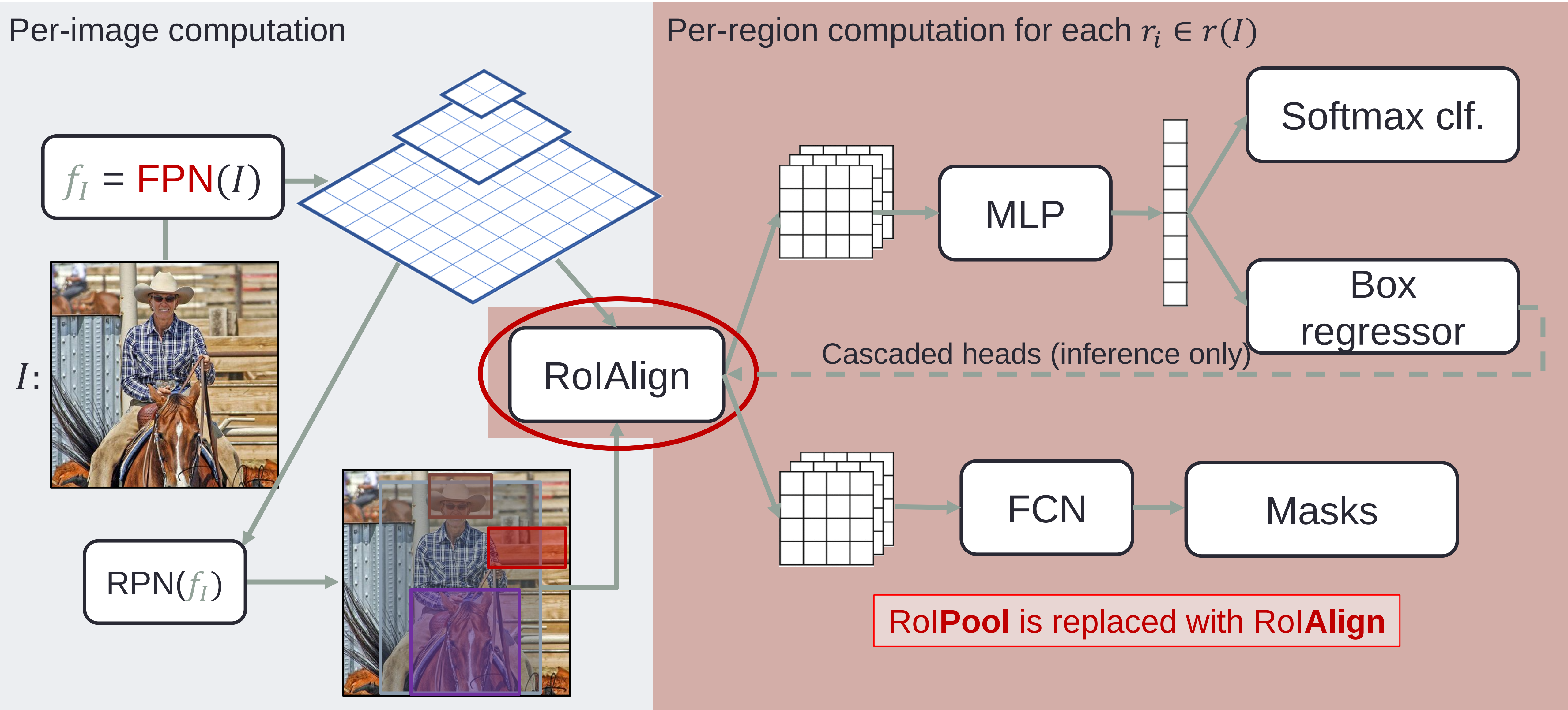
He et al. Mask R-CNN. ICCV 2017.
Best Paper Award

5. Mask R-CNN



He et al. Mask R-CNN. ICCV 2017.
Best Paper Award

5. Mask R-CNN

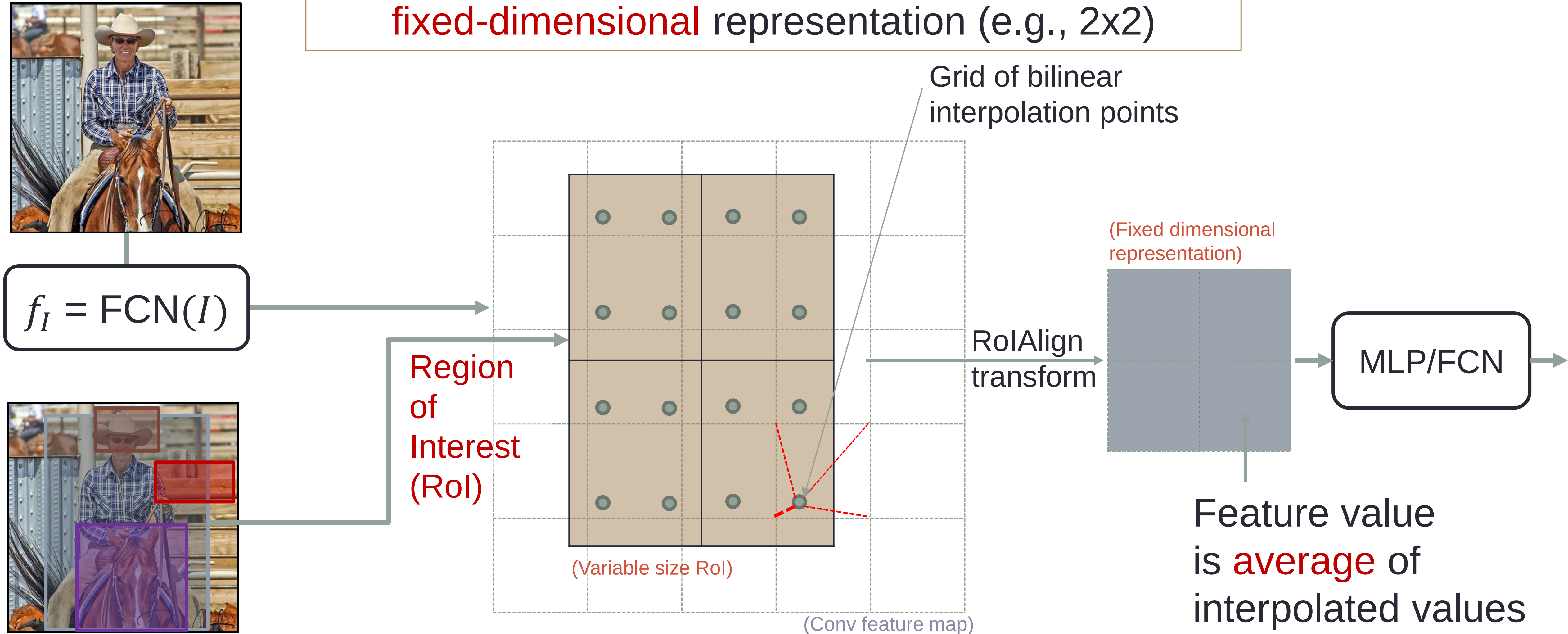


RoI**Pool** is replaced with RoI**Align**

He et al. Mask R-CNN. ICCV 2017.
Best Paper Award

5. RoIAlign Operation (on each Proposal)

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

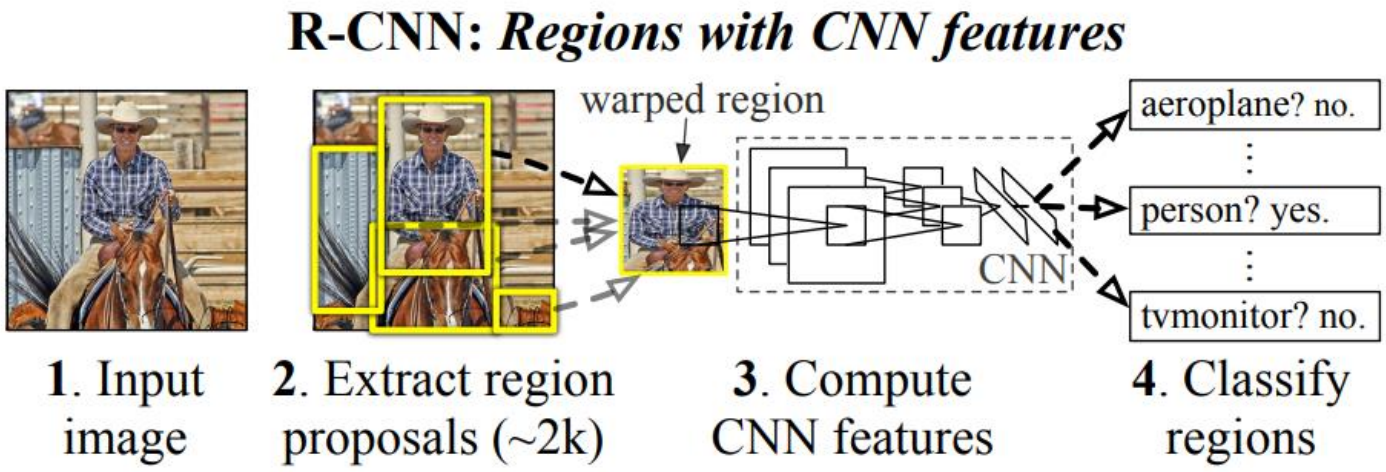


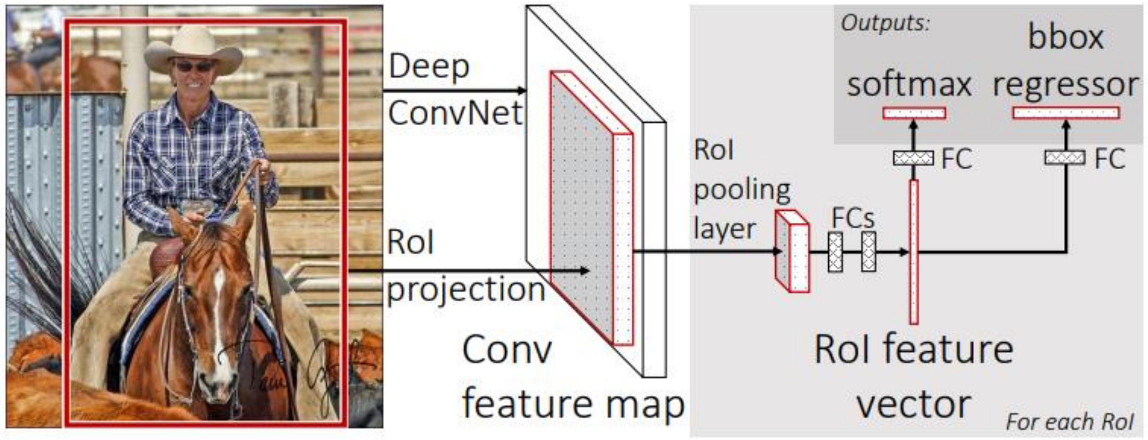

person1.00
person1.00
person1.00
person1.00
person.98
person.91
person.74
surfboard1.00
surfboard1.00
surfboard.98
surfboard1.00
surfboard1.00

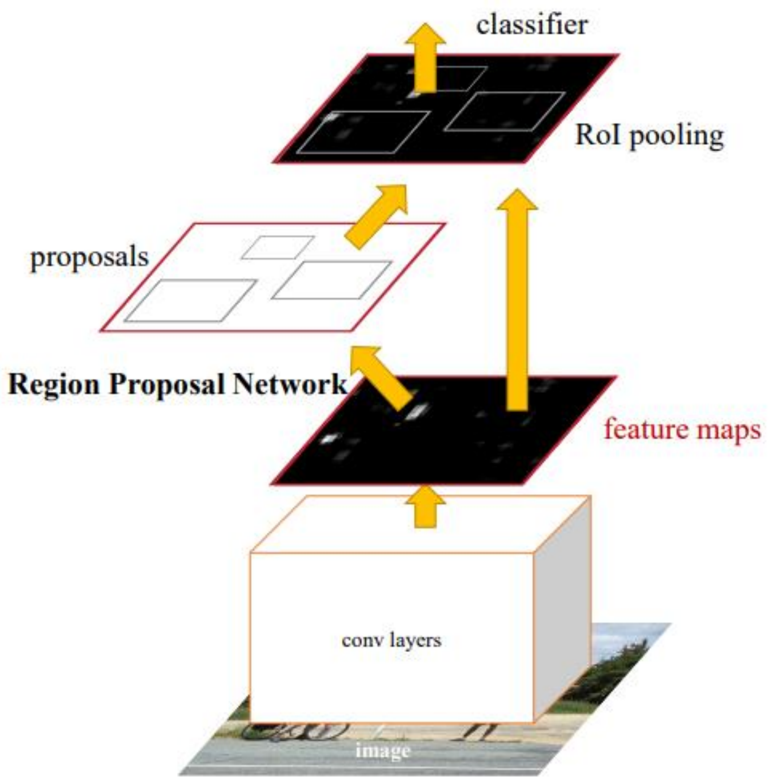

person1.00
person.88
tv.98
tv.84
person1.00
person1.00
bottle.97
wine glass.99
dining table.95
wine glass1.00
wine glass1.00


www.tfrancis.co.uk

Summary of 1-5







Summary of 1-5

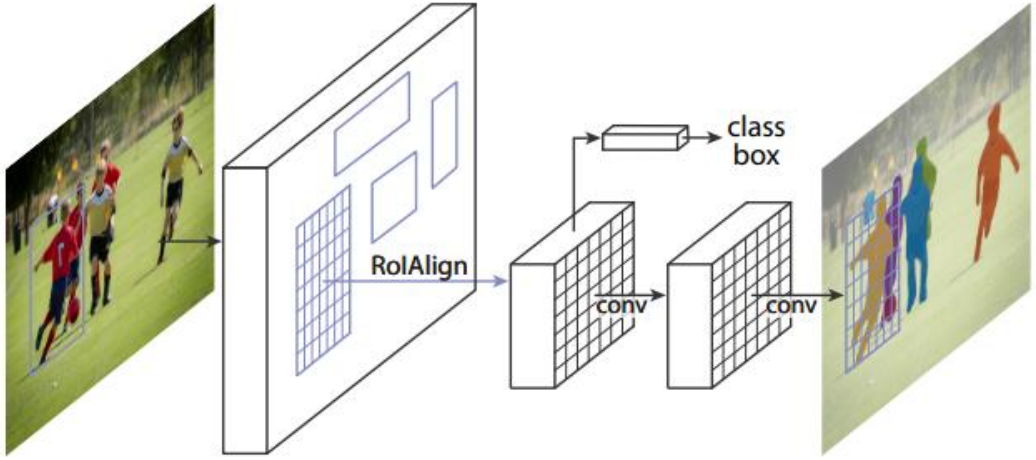
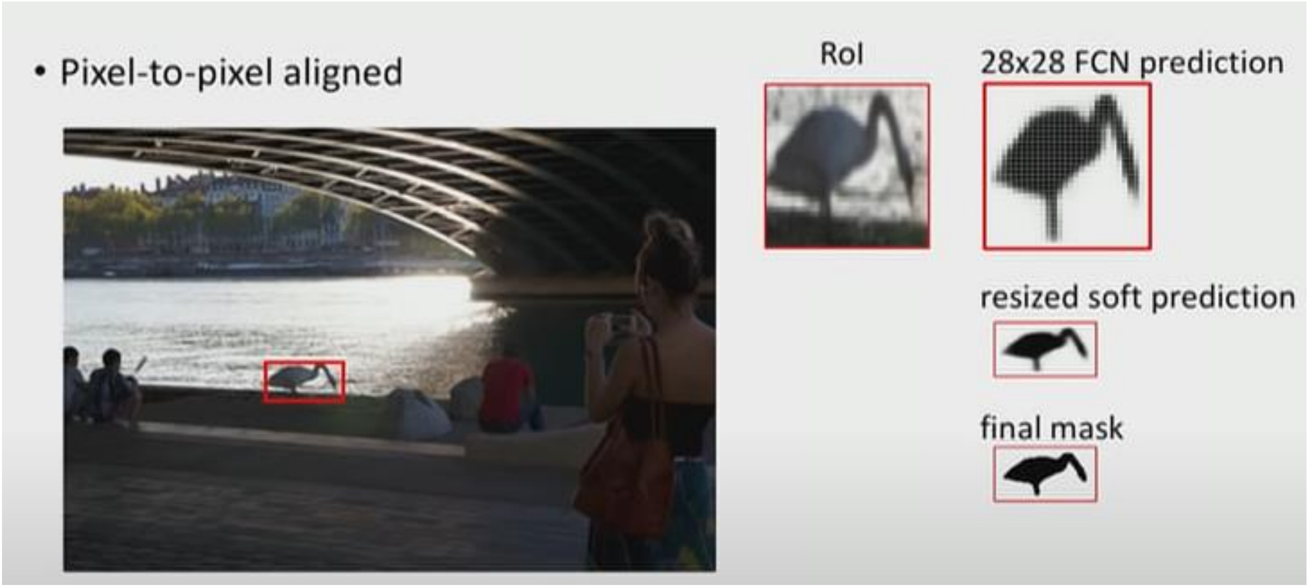
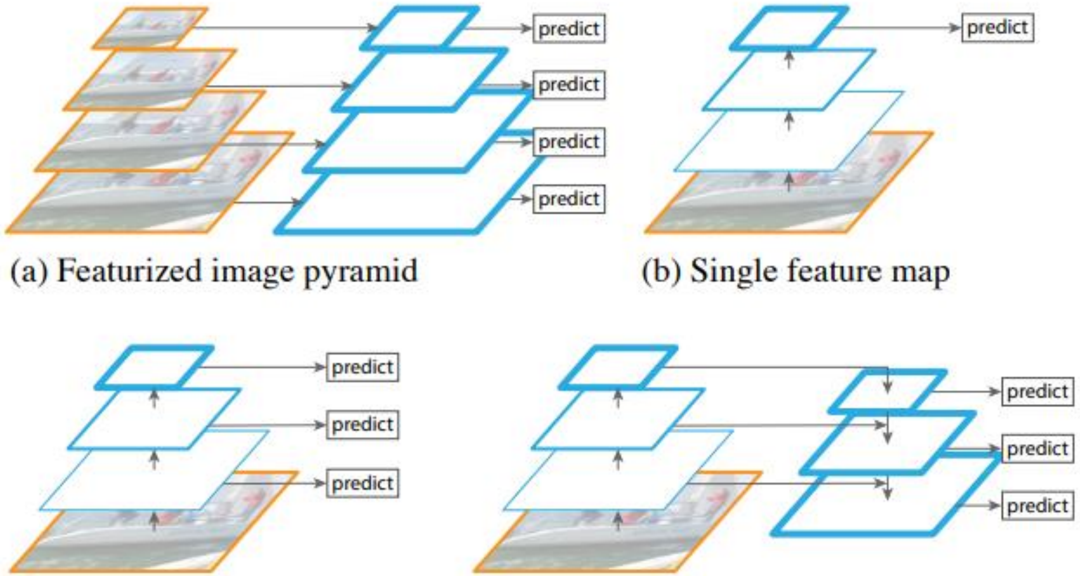


Figure 1. The **Mask R-CNN** framework for instance segmentation.

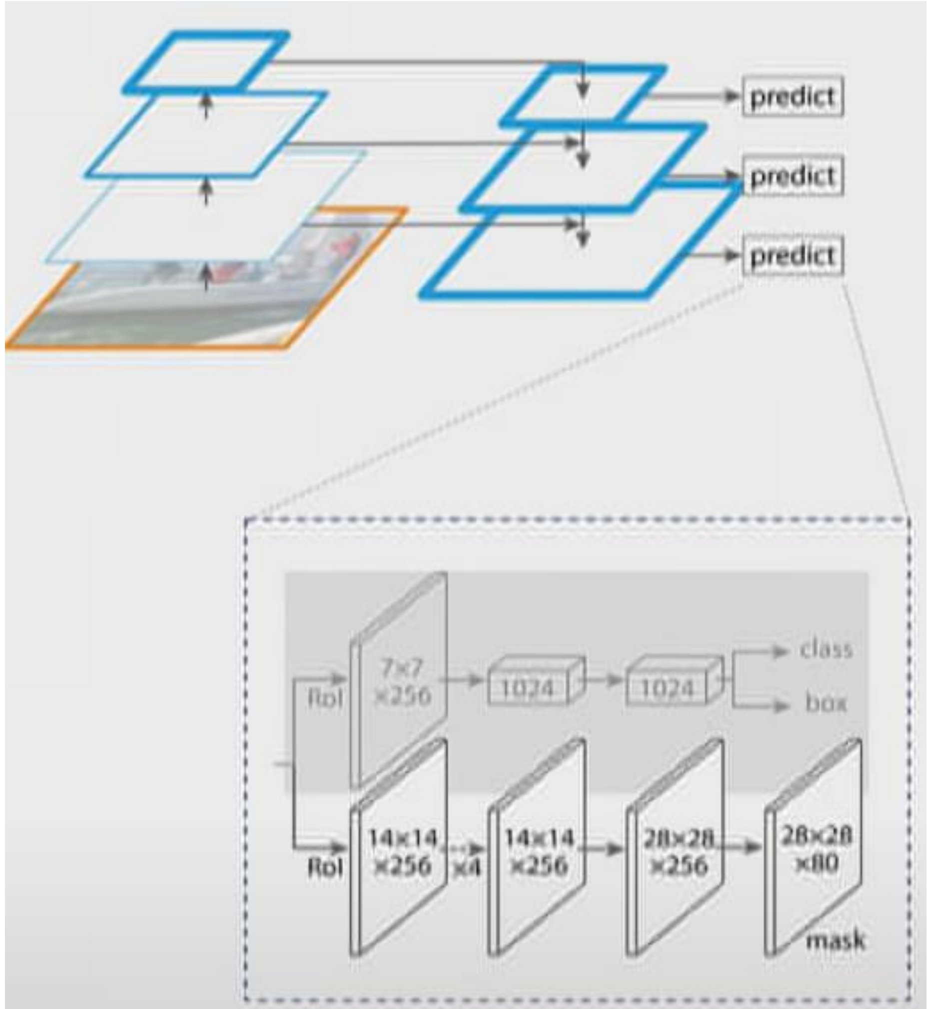


Summary of 1-5



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

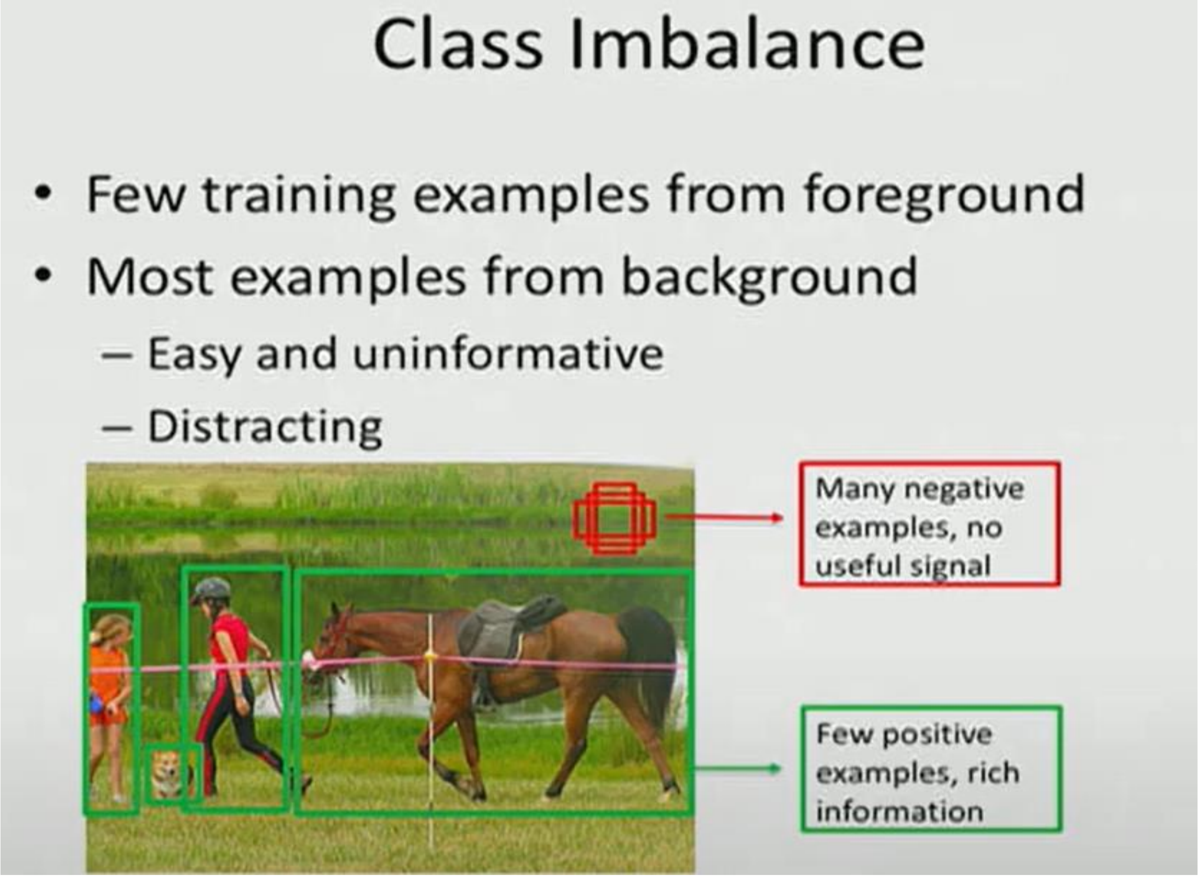


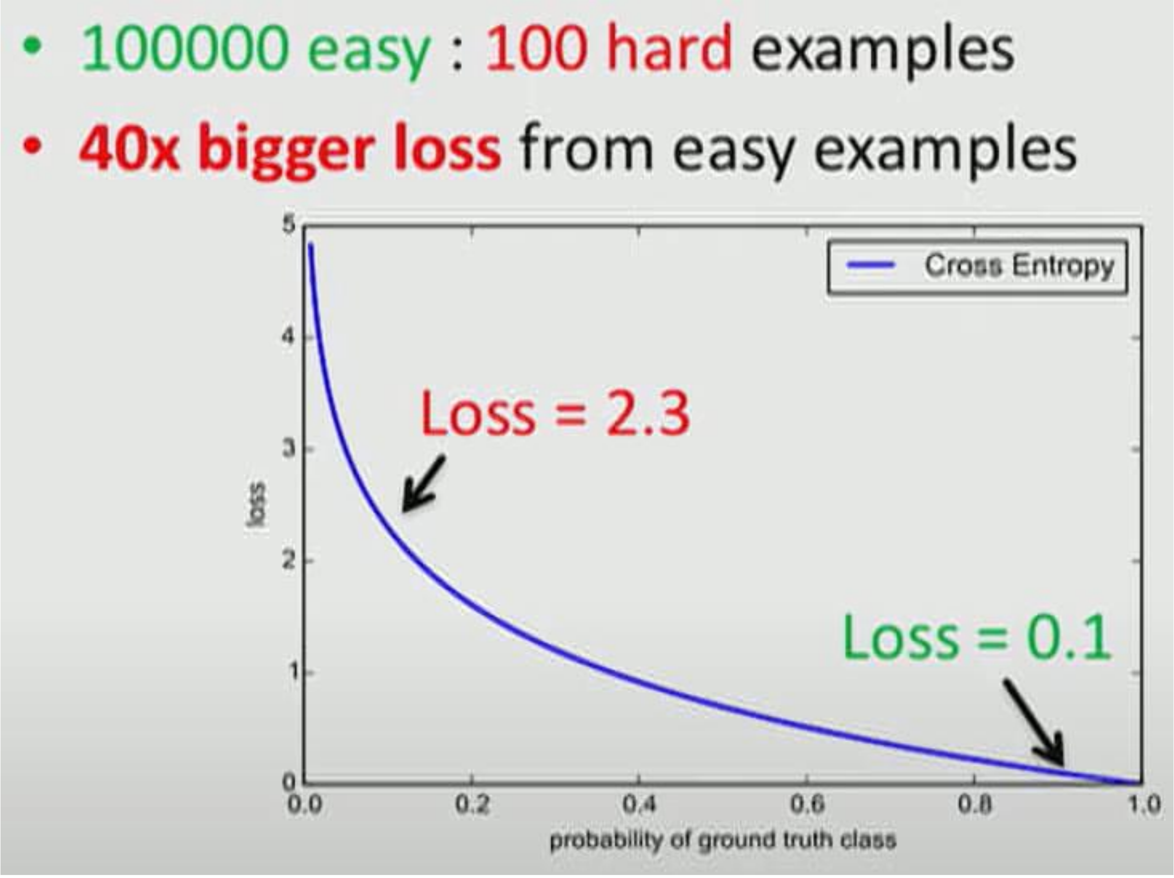
6. RetinaNet

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." In *IEEE International Conference on Computer Vision*, pp. 2980-2988. 2017.
Best Student Paper Award

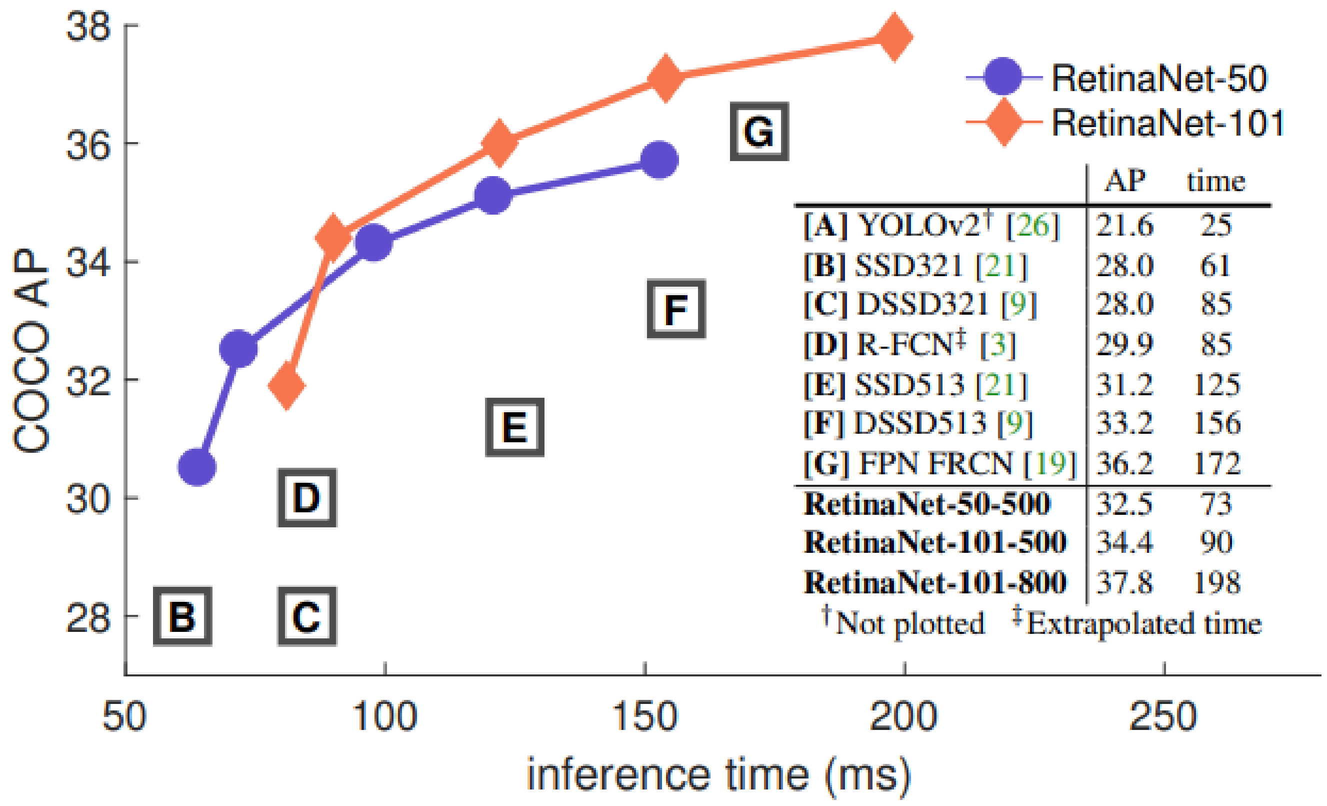
- Identify **class imbalance** is the major issue for training one-stage dense detector
- Propose **Focal Loss** to address class imbalance
- Achieve state-of-the-art **accuracy** and **speed**

6. RetinaNet


Class Imbalance
• Few training examples from foreground
• Most examples from background
– Easy and uninformative
– Distracting
Many negative examples, no useful signal
Few positive examples, rich information


• 100000 easy : 100 hard examples
• 40x bigger loss from easy examples
Cross Entropy
Loss = 2.3
Loss = 0.1
loss
probability of ground truth class
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5

6. RetinaNet

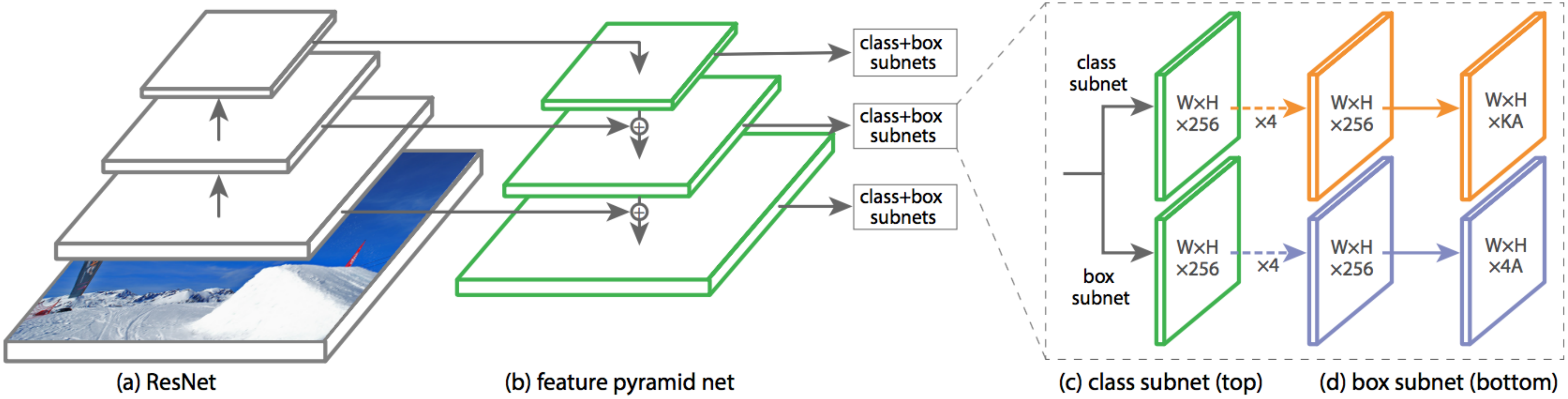


	AP	time
[A] YOLOv2† [26]	21.6	25
[B] SSD321 [21]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [21]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [19]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

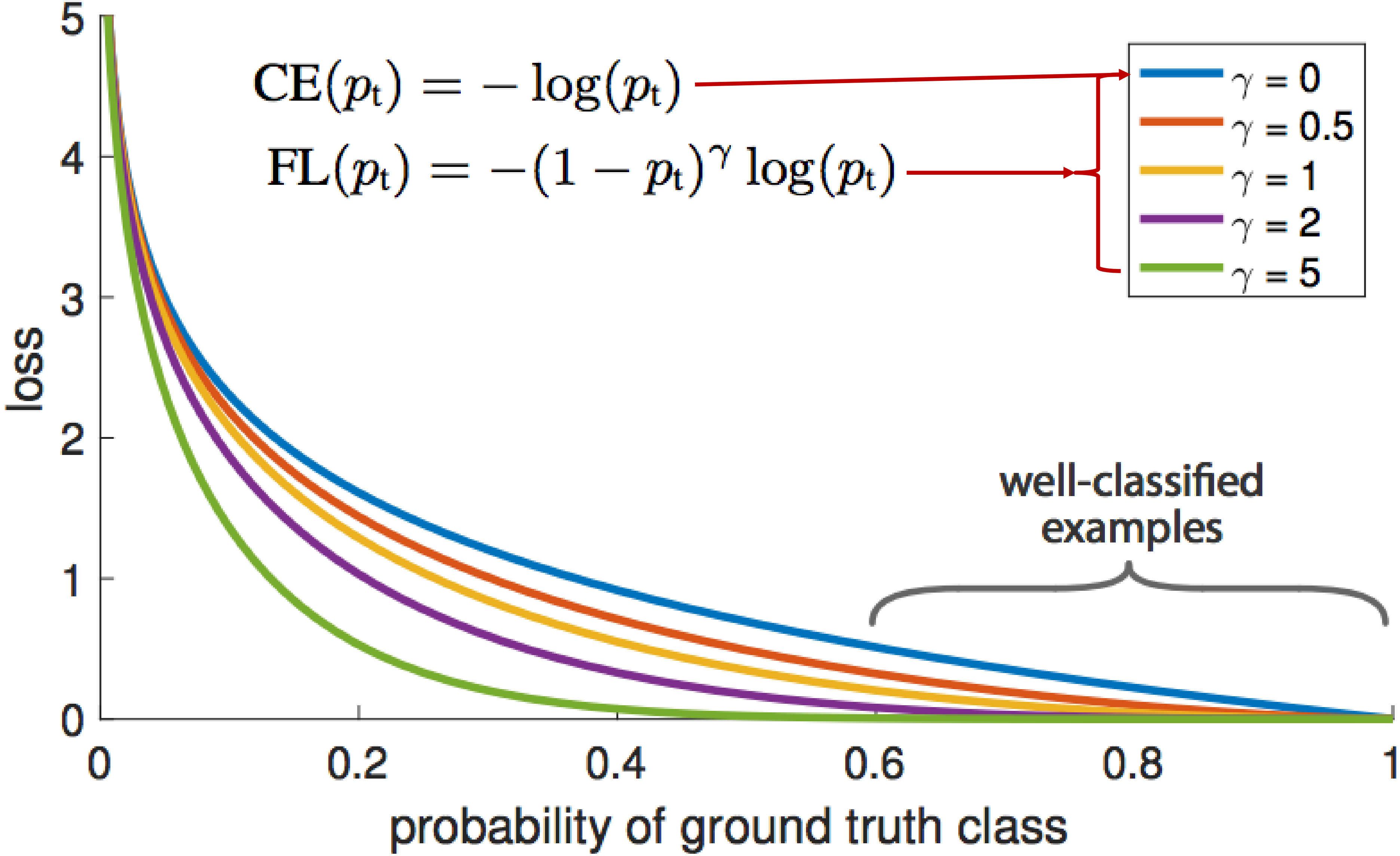
The one-stage RetinaNet detector outperforms all previous one-stage and two-stage detectors

6. RetinaNet Model Description



- Backbone with FPN
- 6 anchors per location (2 scales × 3 aspect ratios)
- 100 – 200k anchor boxes to classify per image → "dense" detection

6. Focal Loss: larger Gamma focuses more on Hard-Example



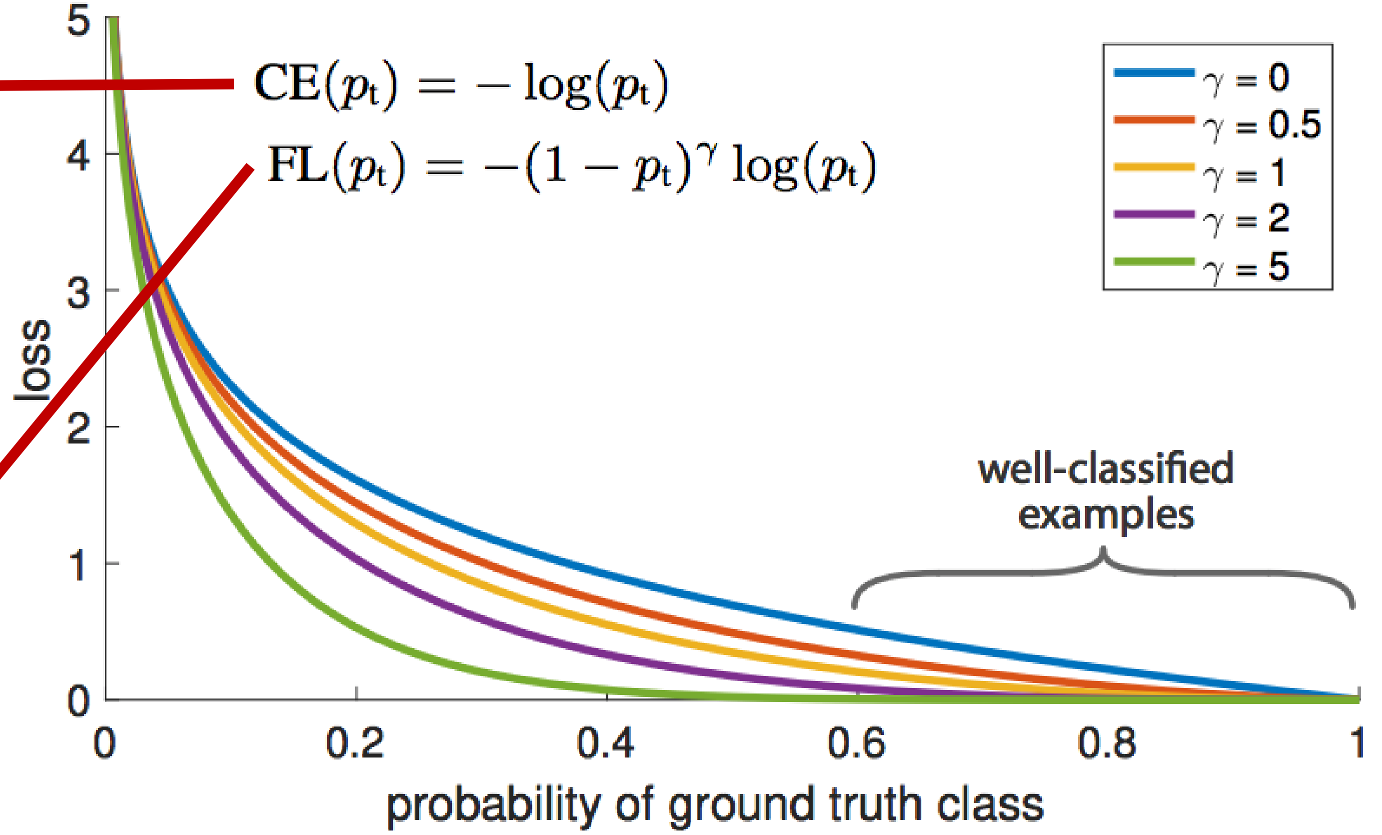
6. Impact of Focal Loss (FL)

$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t).$

γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

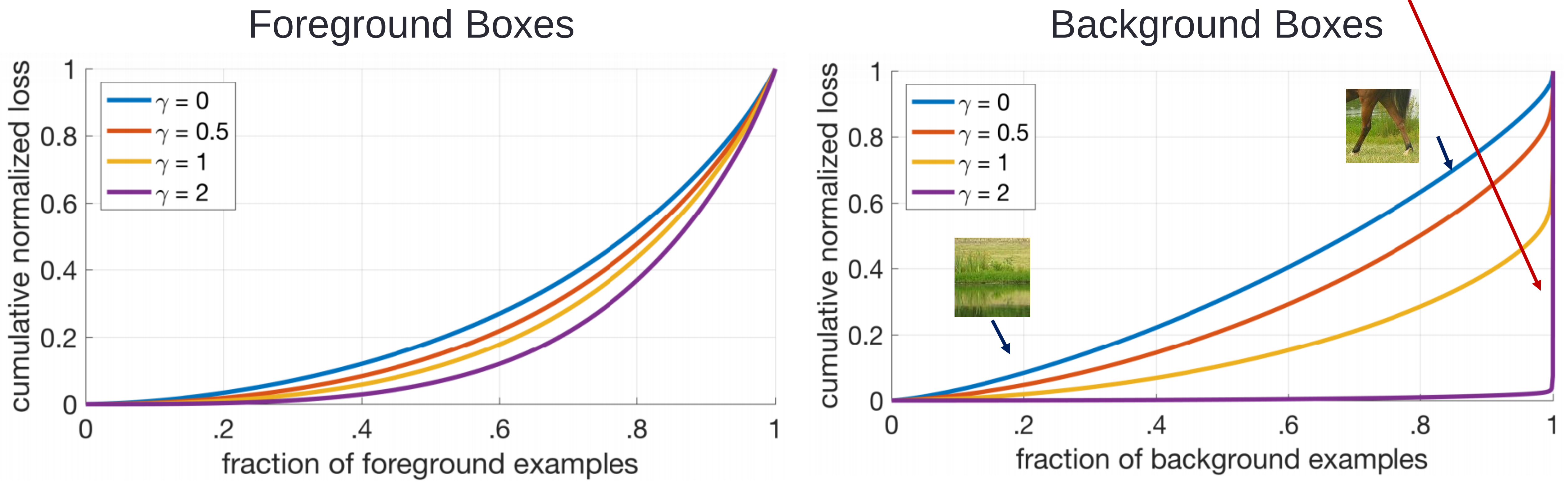
(b) **Varying γ for FL** (w. optimal α)

(ResNet-50-FPN 600px input image)



6. Loss Distribution under Focal Loss

Hard mining behavior



Reference:

1. Dr. Ross Girshick, Facebook AI Research, CVPR 2019 Tutorial on Visual Recognition and Beyond
https://www.dropbox.com/s/rakgo35h6b8p7uv/cvpr2019_tutorial_ross_girshick.pptx?dl=0

2. Dr. Kaiming He, ICCV 2017 Oral Talk for the Mask R-CNN paper.
https://www.youtube.com/watch?v=g7z4mkfRjI4&ab_channel=ComputerVisionFoundationVideos

3. Dr. Tsung-Yi Lin, ICCV 2017 Oral Talk for the Focal Loss paper.
https://www.youtube.com/watch?v=44tlnmmt3h0&ab_channel=ComputerVisionFoundationVideos