The Generalized R-CNN Framework for Object Detection

CVPR 2018 Tutorial

Ross Girshick

Facebook AI Research (FAIR)

Overview of this Tutorial

Topics to cover

- Object detection intro (very brief)
- The Generalized R-CNN framework

Overview of this Tutorial

Topics to cover

- Object detection intro (very brief)
- The Generalized R-CNN framework

Bounding-Box Object Detection

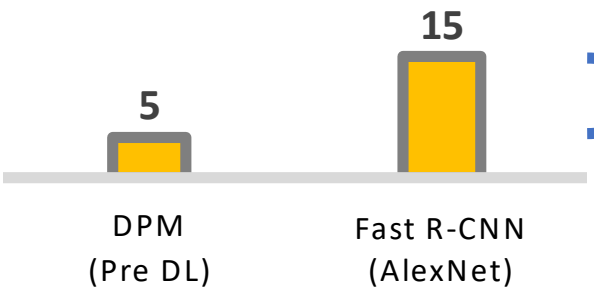
Object detection

How Well are Detectors Working?

COCO Object Detection Average Precision (%)

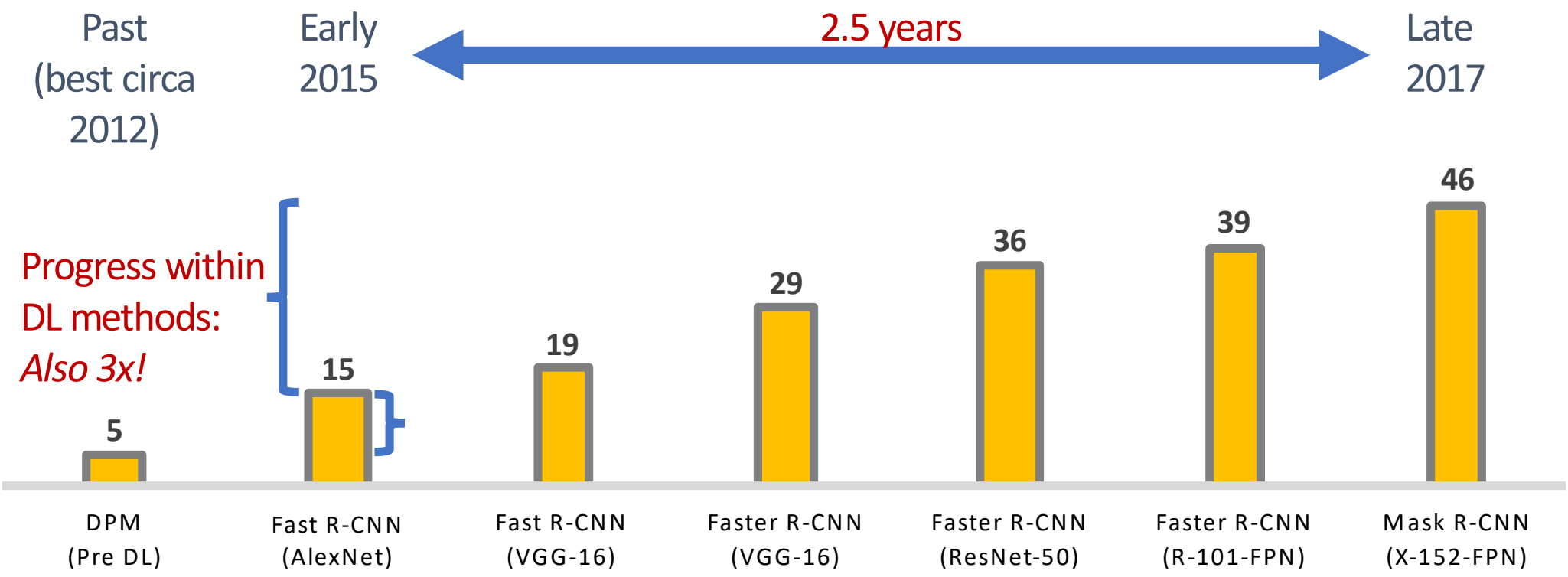
Past (best circa 2012)

Early 2015



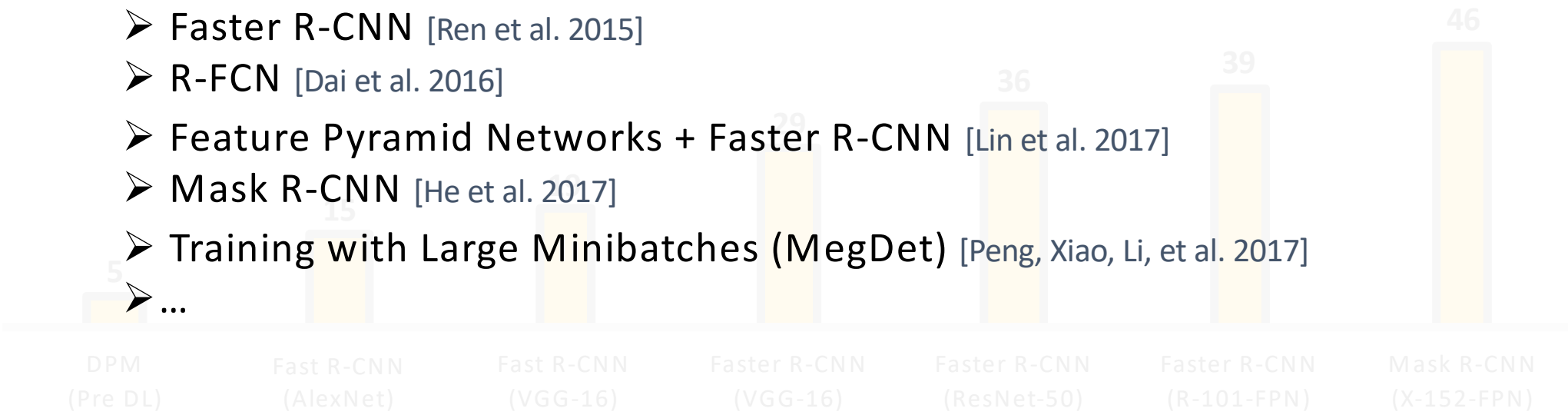
Movement to
Deep Learning methods:
3x improvement in AP

COCO Object Detection Average Precision (%)



Steady Progress

- 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]
- …



Detection Beyond Bounding Boxes

Mask R-CNN

[He, Gkioxari, Dollár, Girshick]

Detection Beyond Bounding Boxes

DensePose: Dense Human Pose Estimation In The Wild
[Güler, Neverova, Kokkinos]

Overview of this Tutorial

Topics to cover

- Object detection intro (very brief)
- The Generalized R-CNN framework

R-CNN

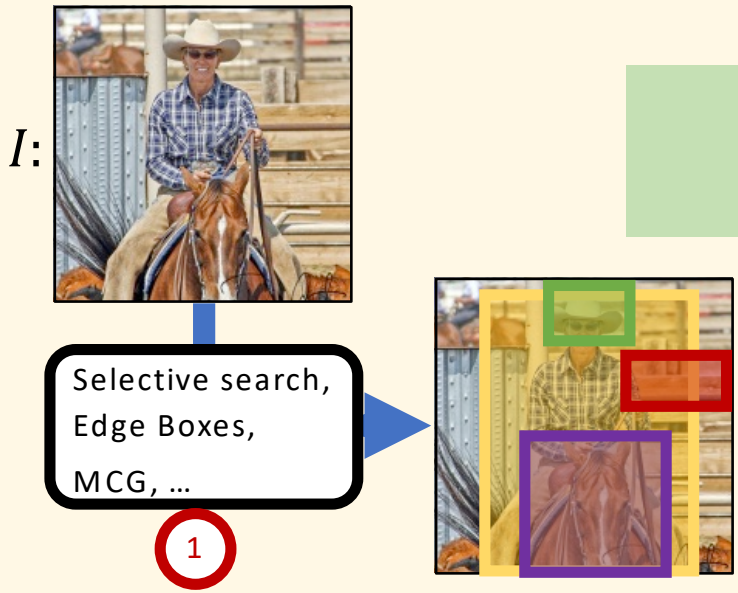
References

- B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. TPAMI, 2012.
- I. Endres and D. Hoiem. Category independent object proposals. In ECCV, 2010.
- J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013.
- A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.
- X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In ICCV, 2013.
- R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

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



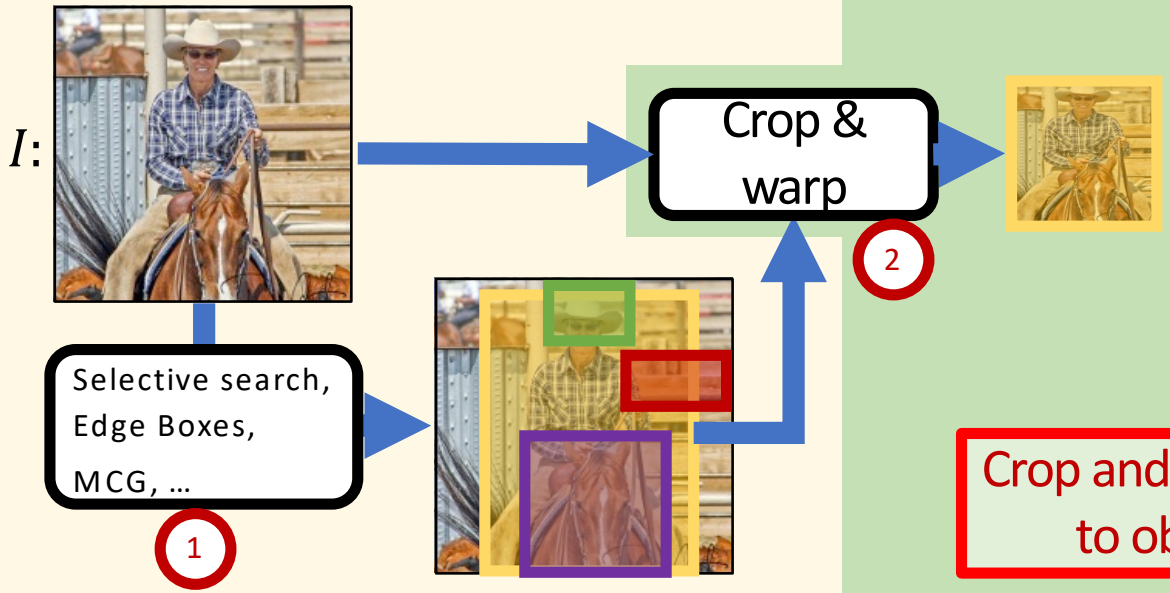
Use an off-the-shelf region/object/detection proposal algorithm (~2k proposals per image)

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

R-CNN

Per-image computation

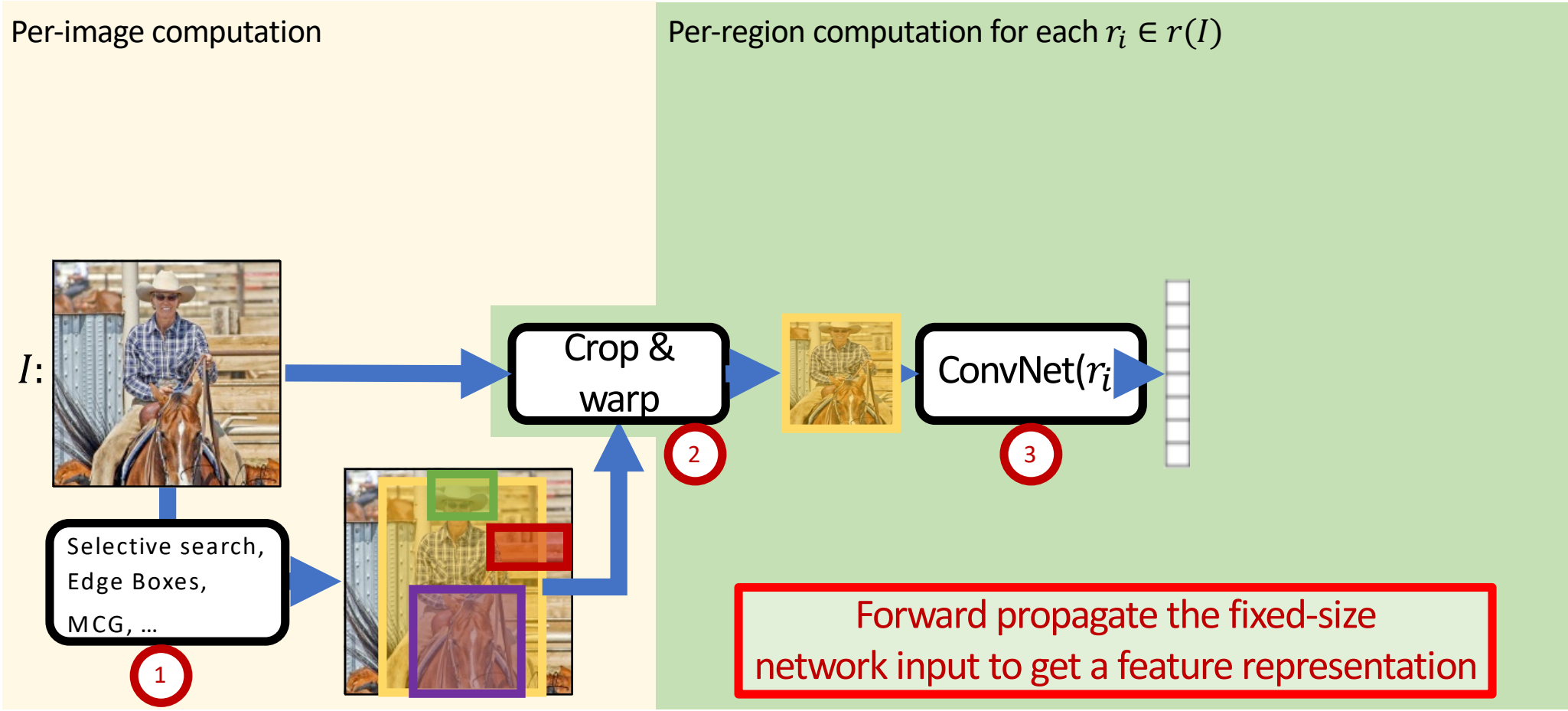
Per-region computation for each $r_i \in r(I)$



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

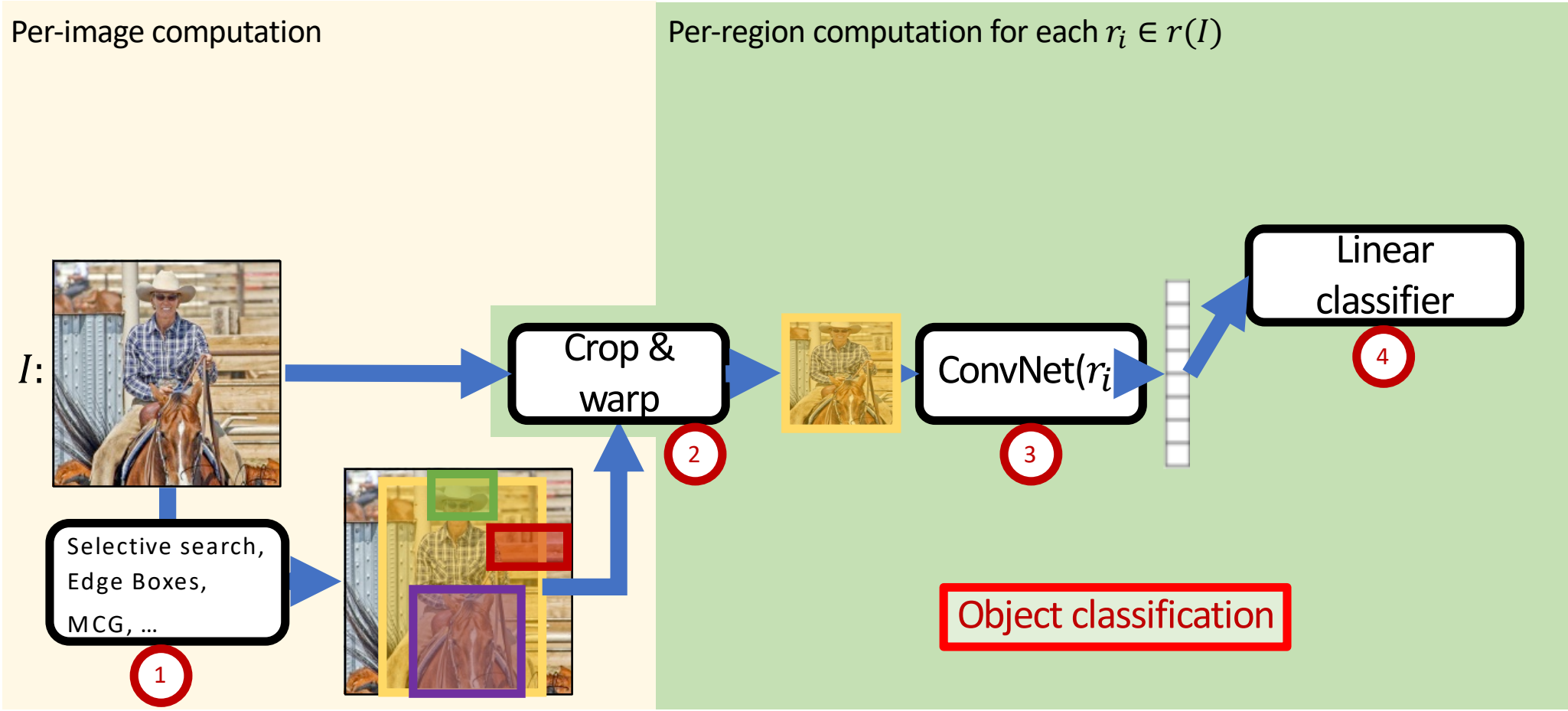
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN



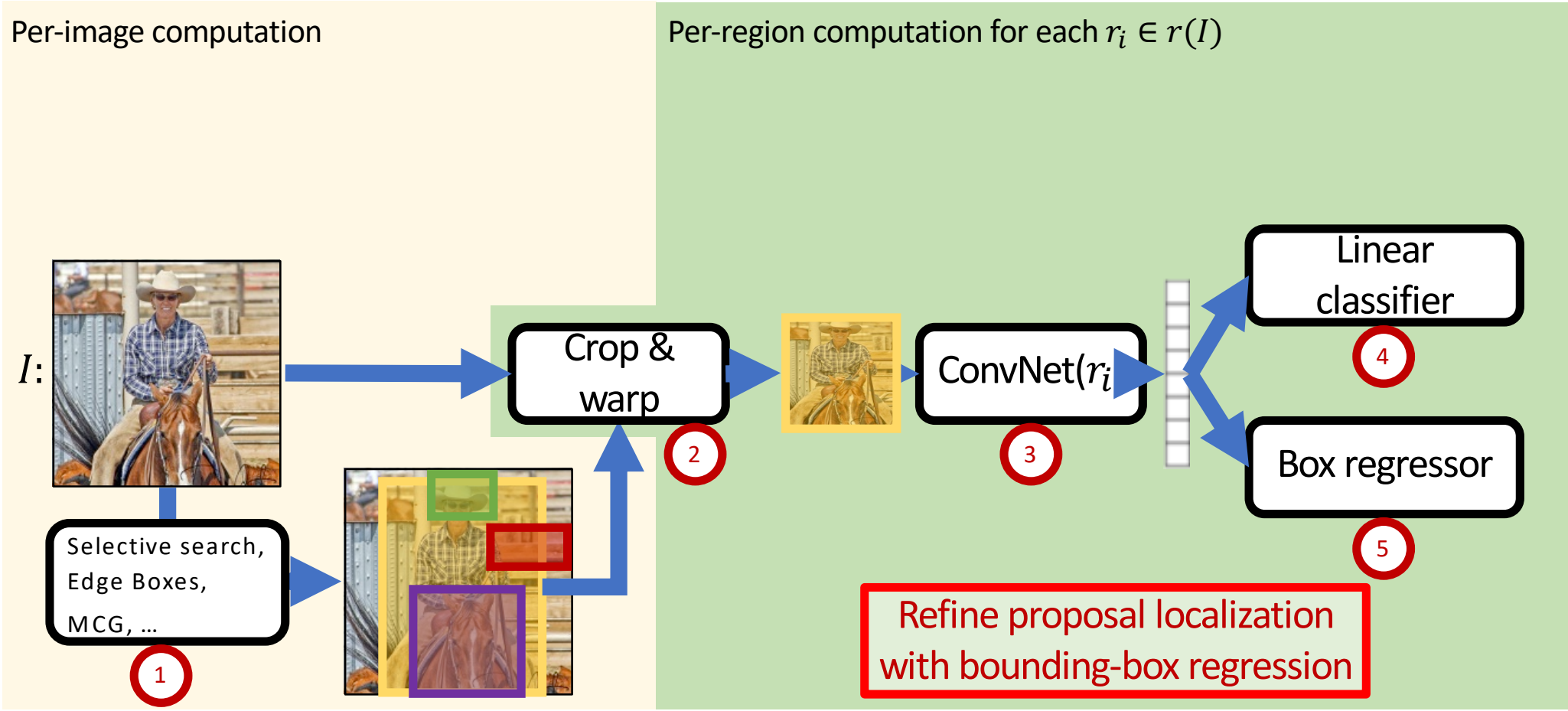
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN



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

R-CNN



Generalized R-CNN Framework

A common framework for understanding

- R-CNN
- Fast R-CNN
- Faster R-CNN
- Feature Pyramid Networks (FPN) + Faster R-CNN
- Mask R-CNN
- … and more (e.g., DensePose)

https://github.com/facebookresearch/Detectron

Generalized R-CNN Framework

Per-image computation

I: 

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

Input image
per-image operations | per-region operations

Generalized R-CNN Framework

Per-image computation

$f_I = f(I)$

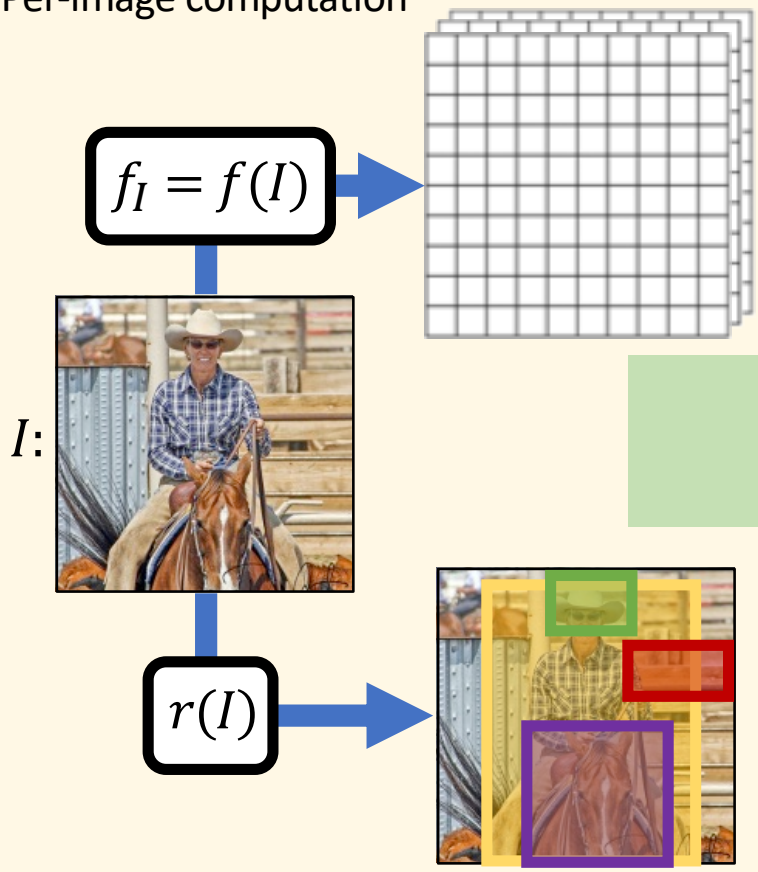
I:

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

Transformation of the input image into a featurized representation

Generalized R-CNN Framework

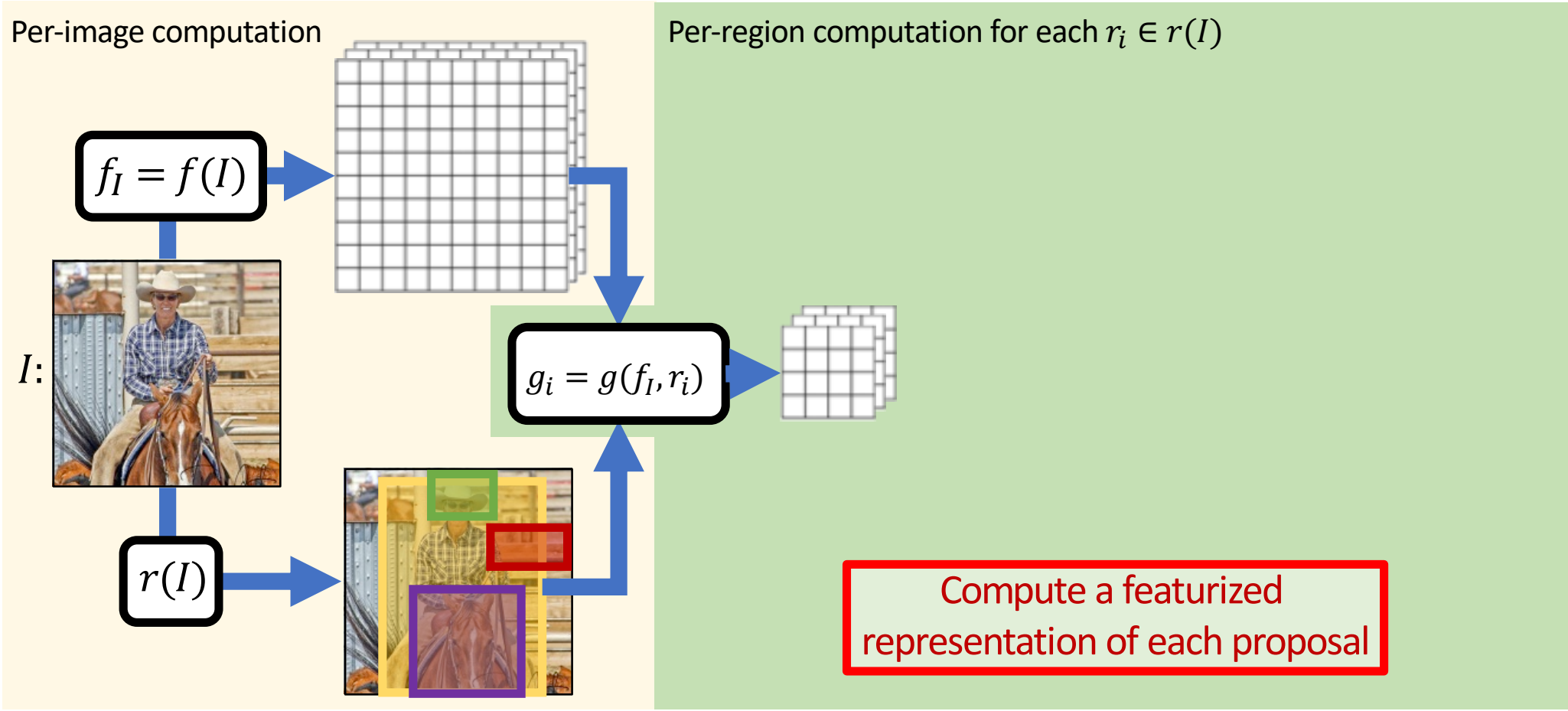
Per-image computation



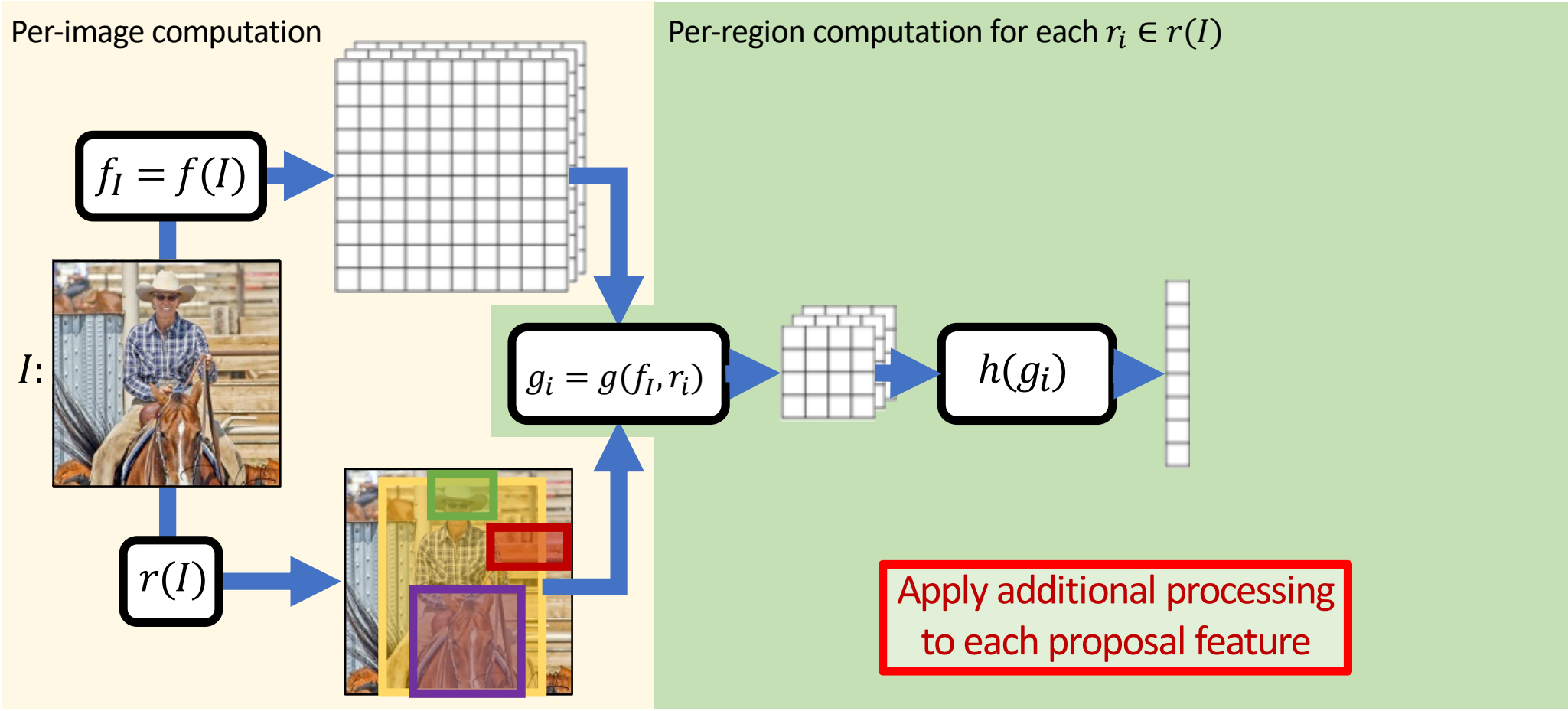
Per-region computation for each $r_i \in r(I)$

Object/region/detection proposals computed for the image

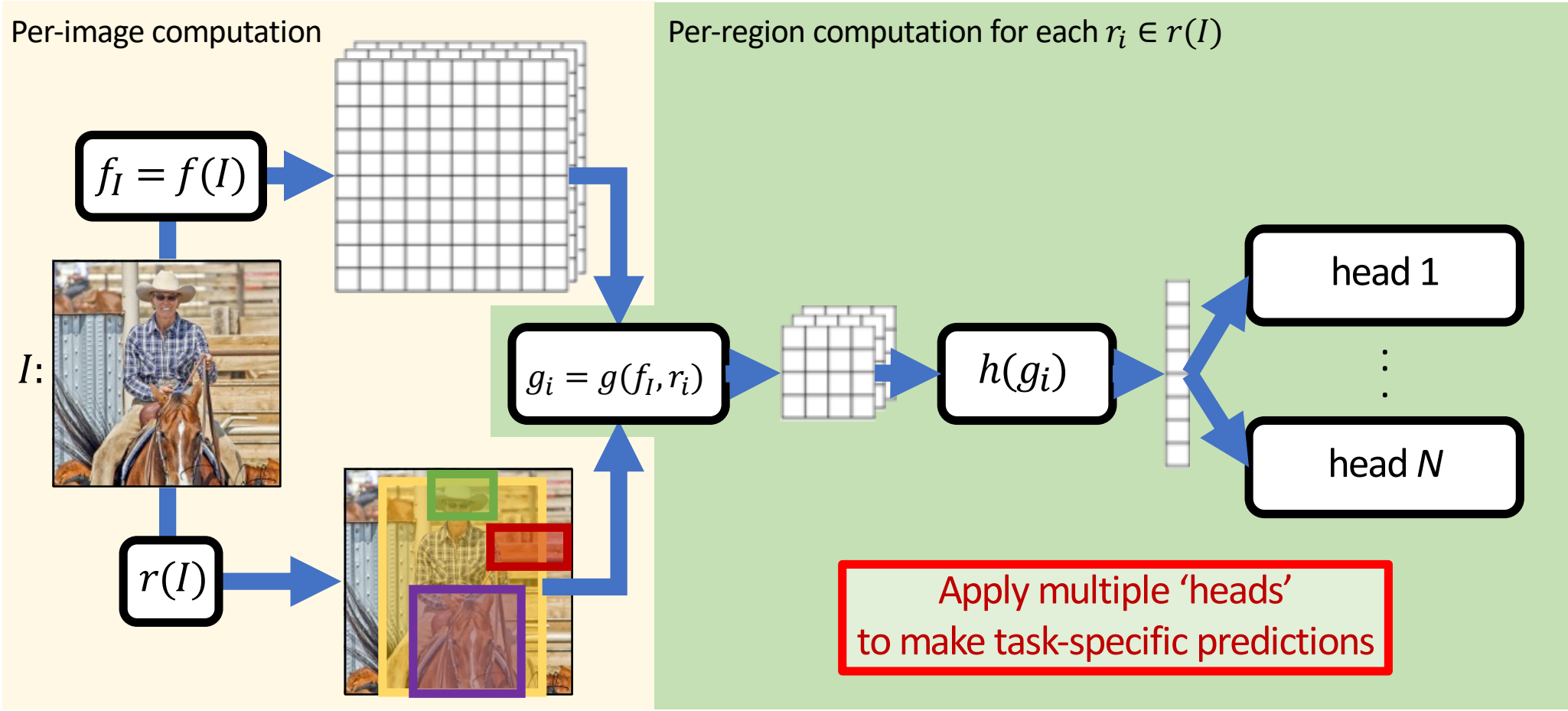
Generalized R-CNN Framework



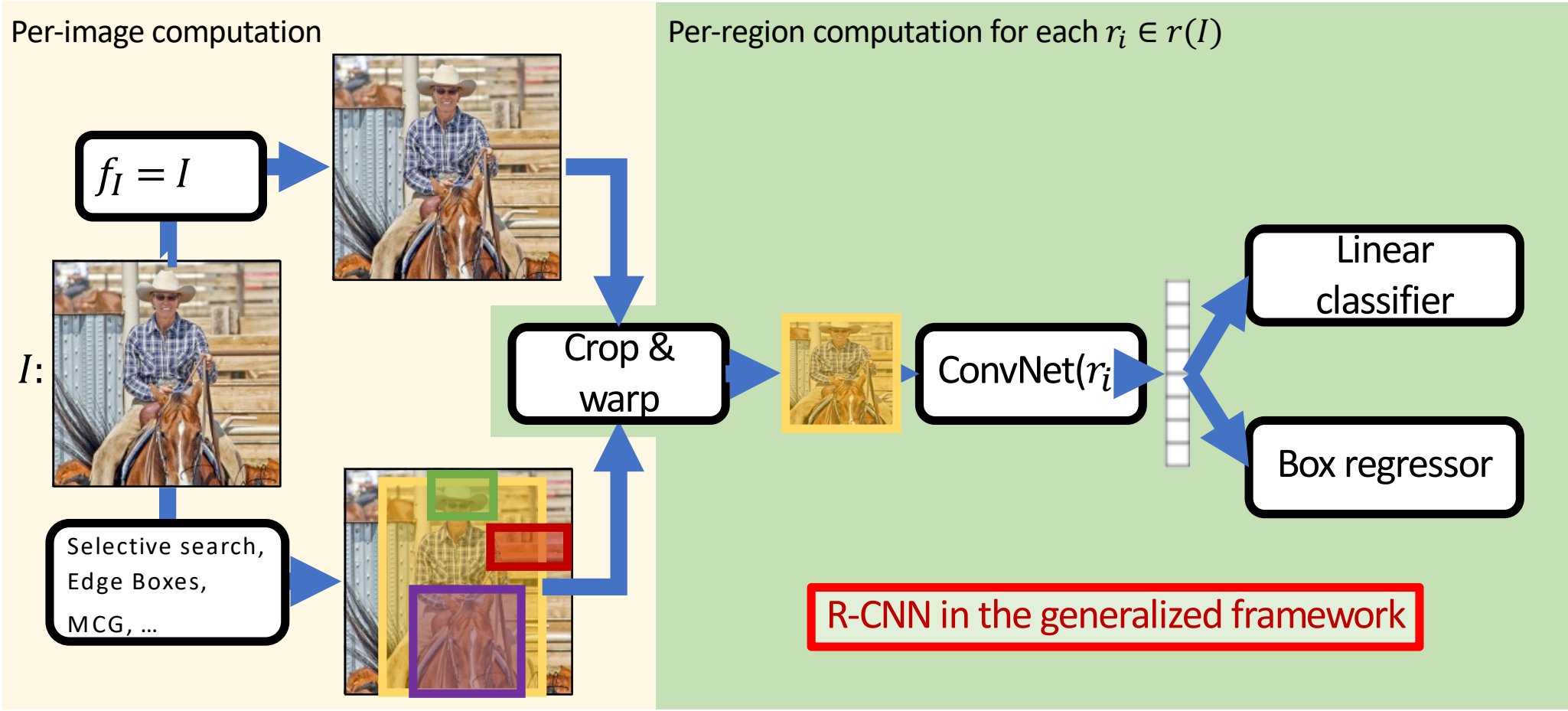
Generalized R-CNN Framework



Generalized R-CNN Framework

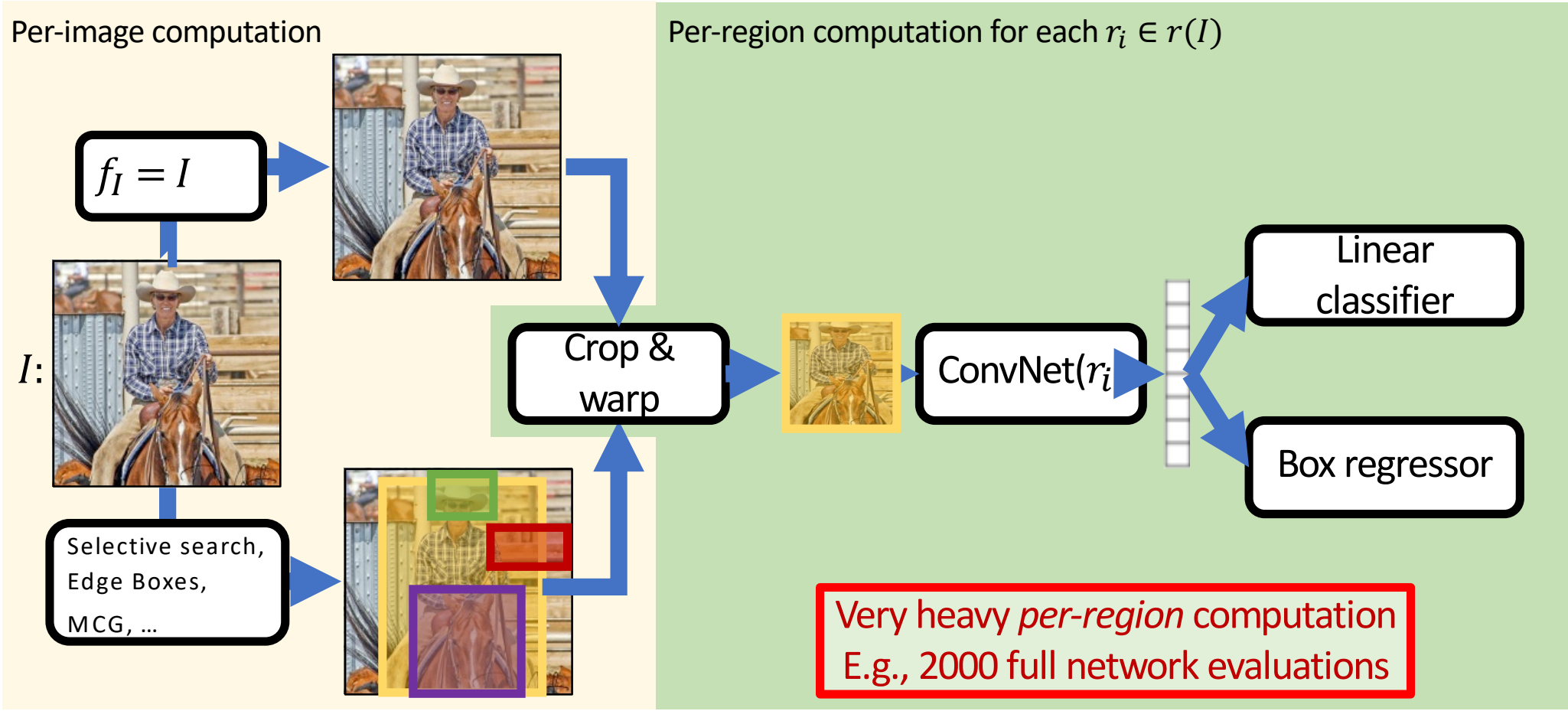


R-CNN in the Generalized Framework



The Problem with 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

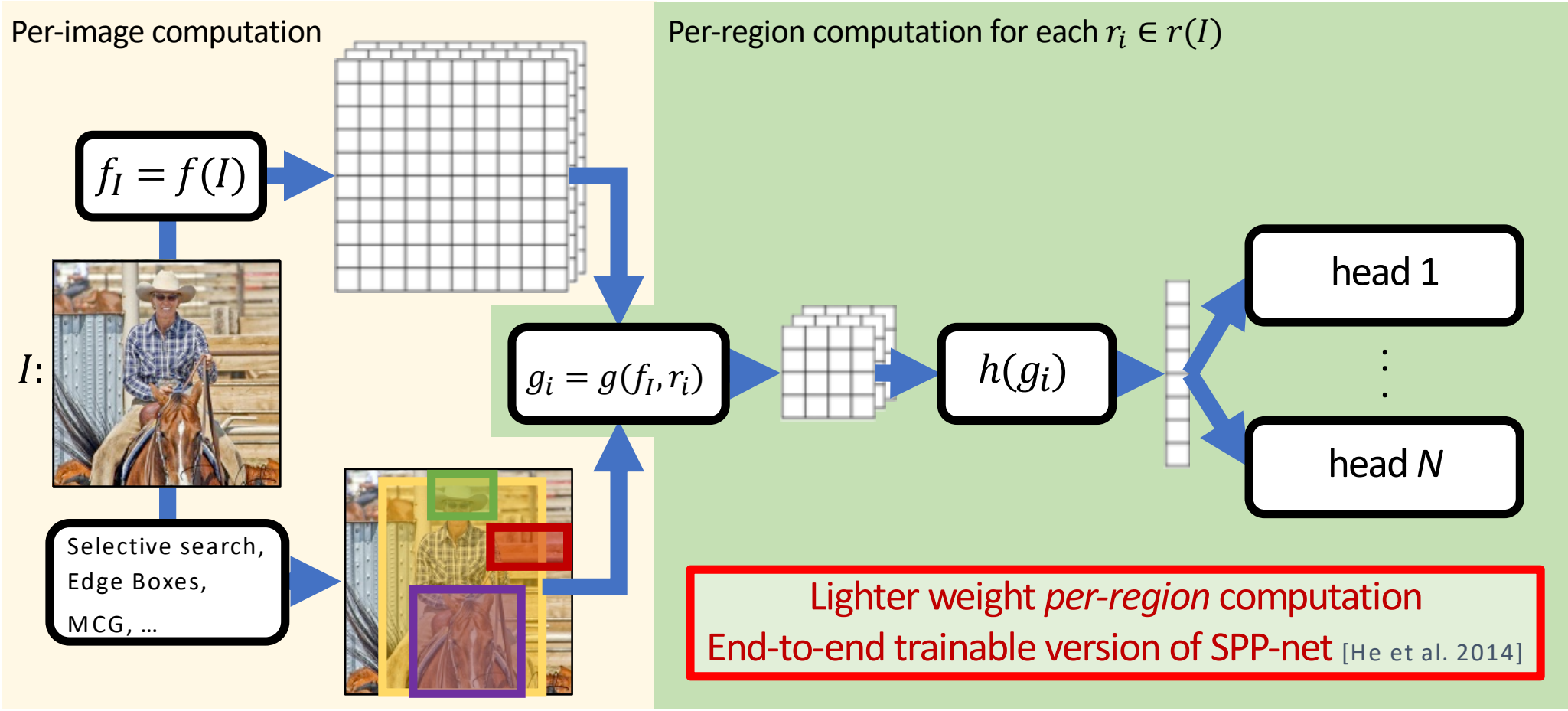
Fast R-CNN

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.

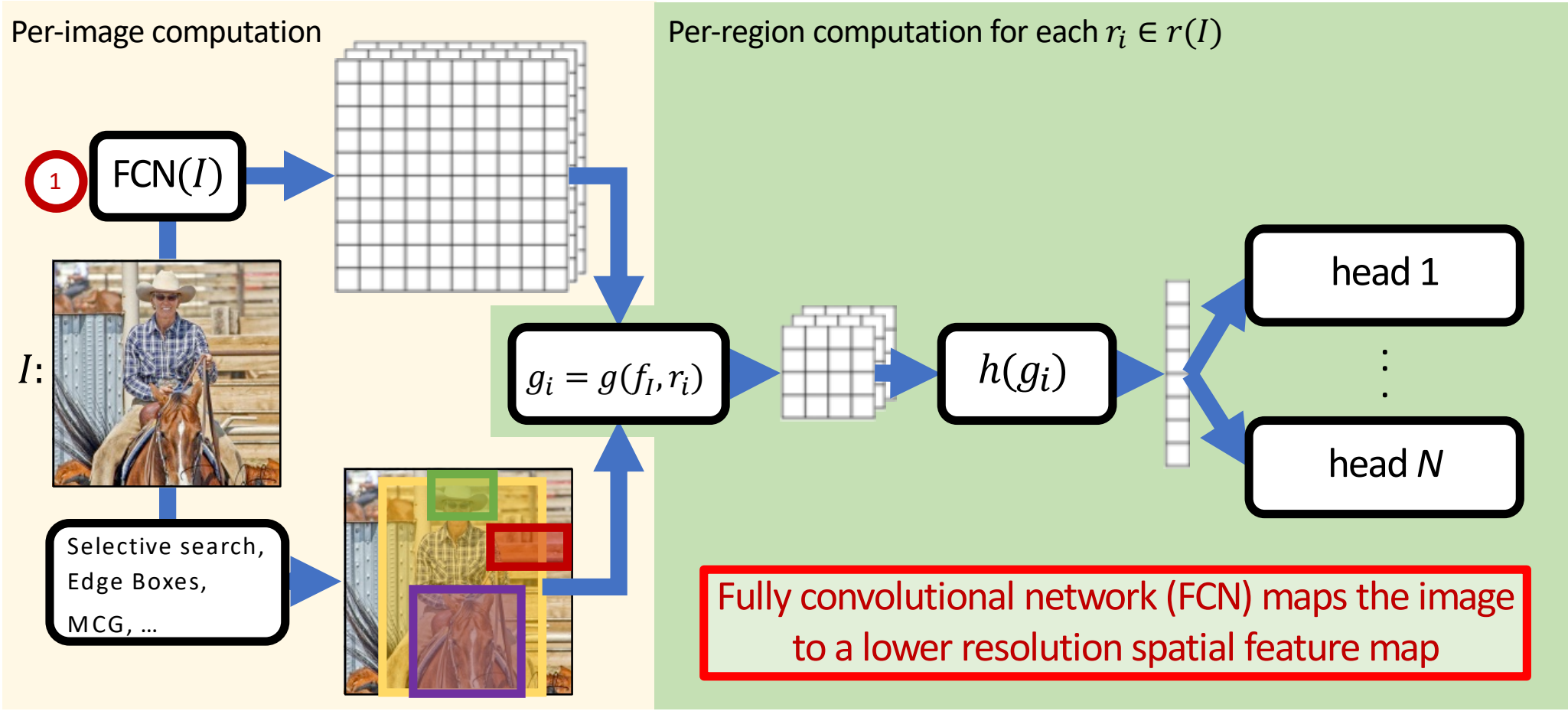
Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN



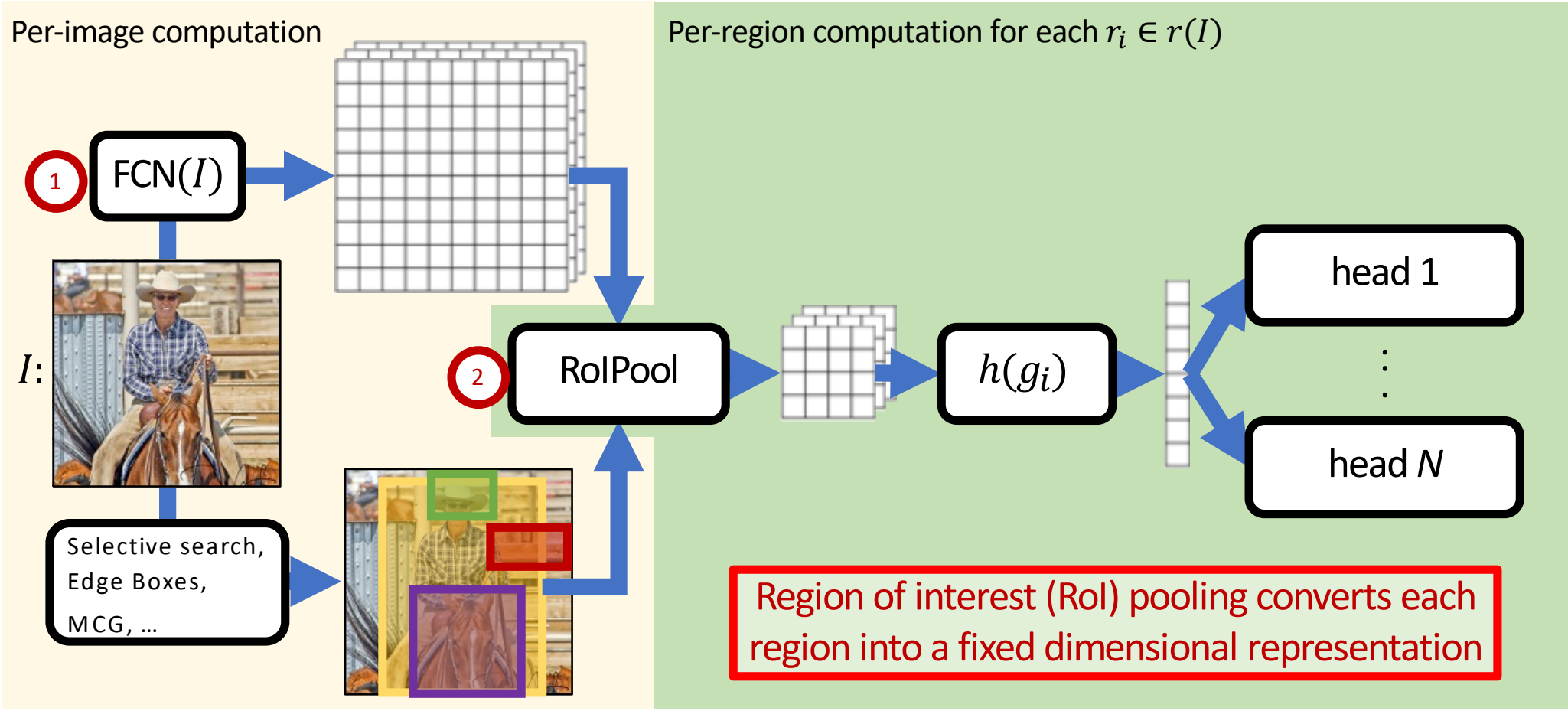
Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN



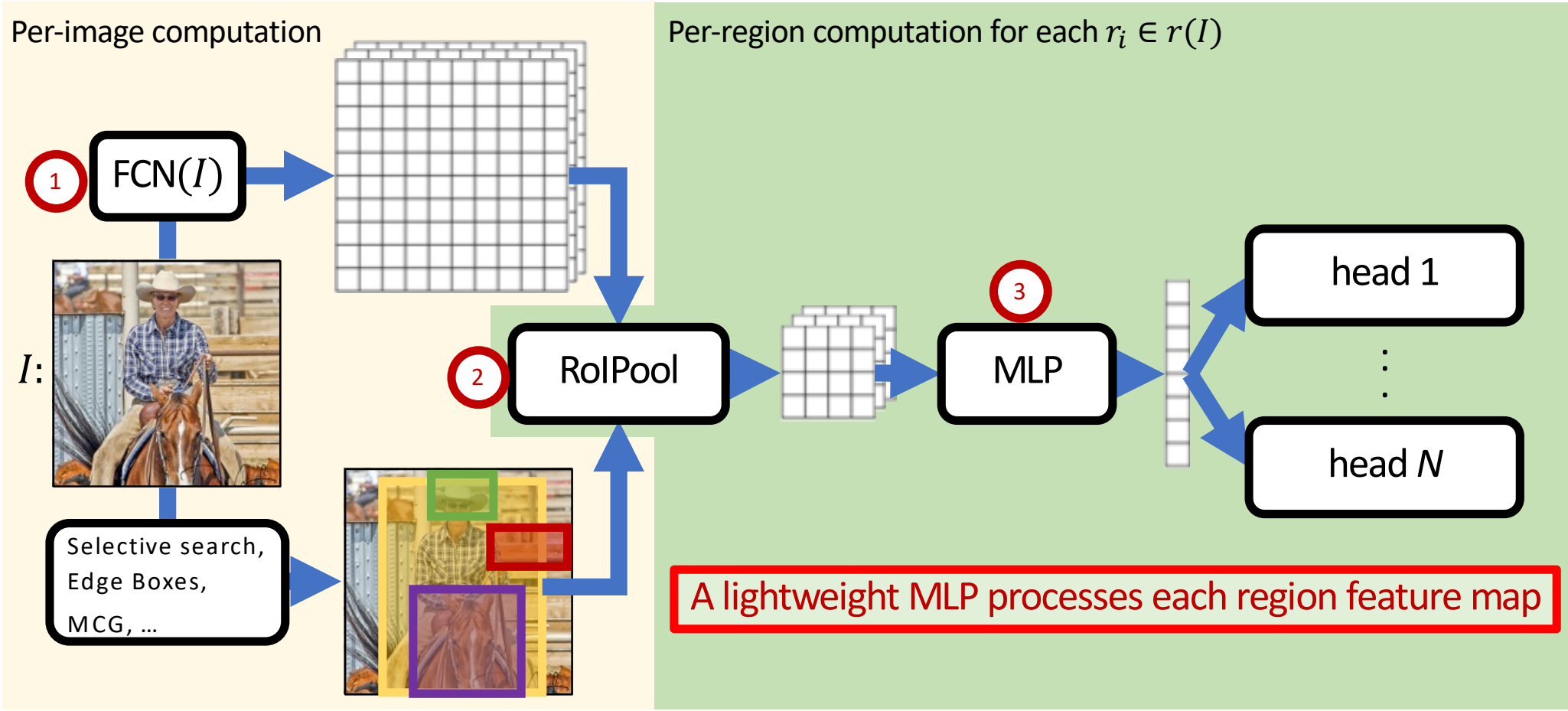
Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN



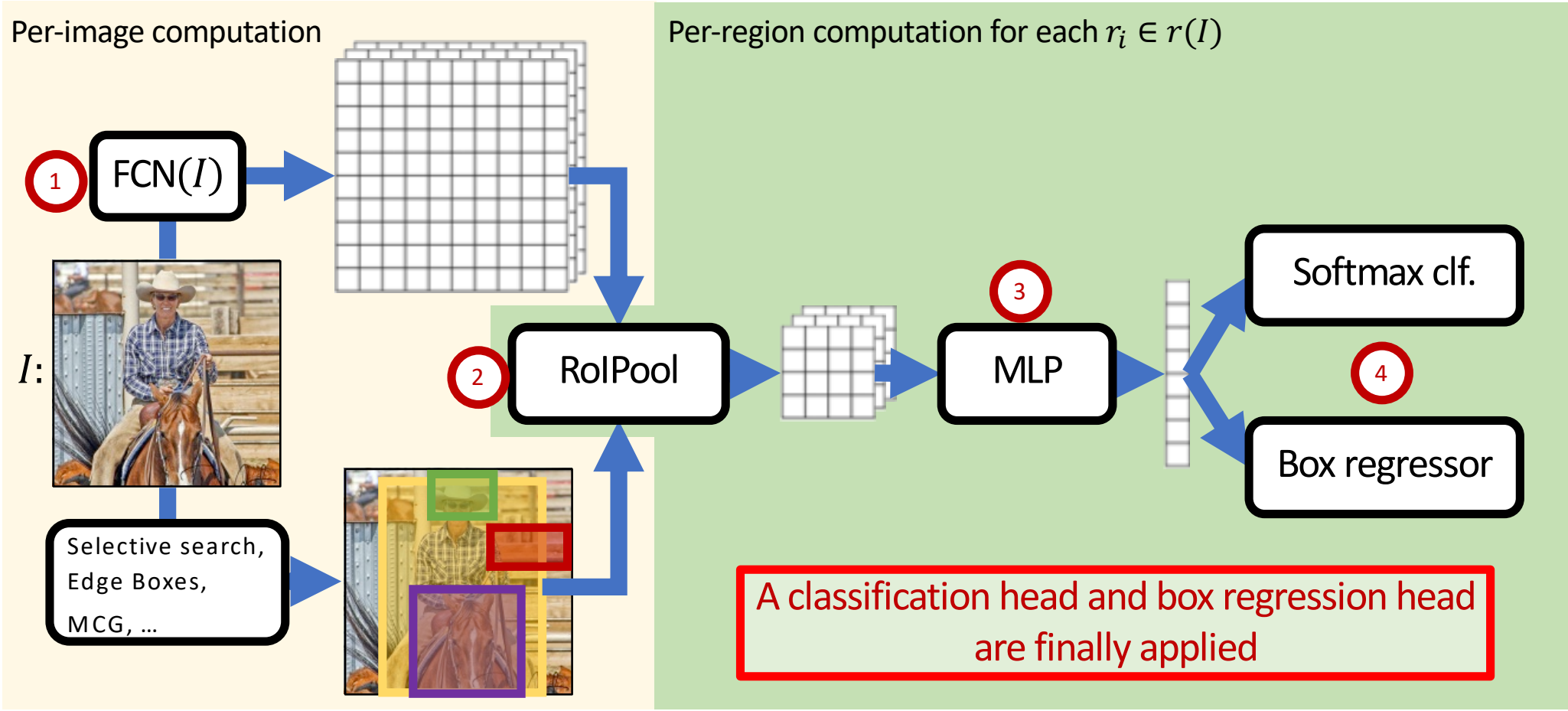
Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN

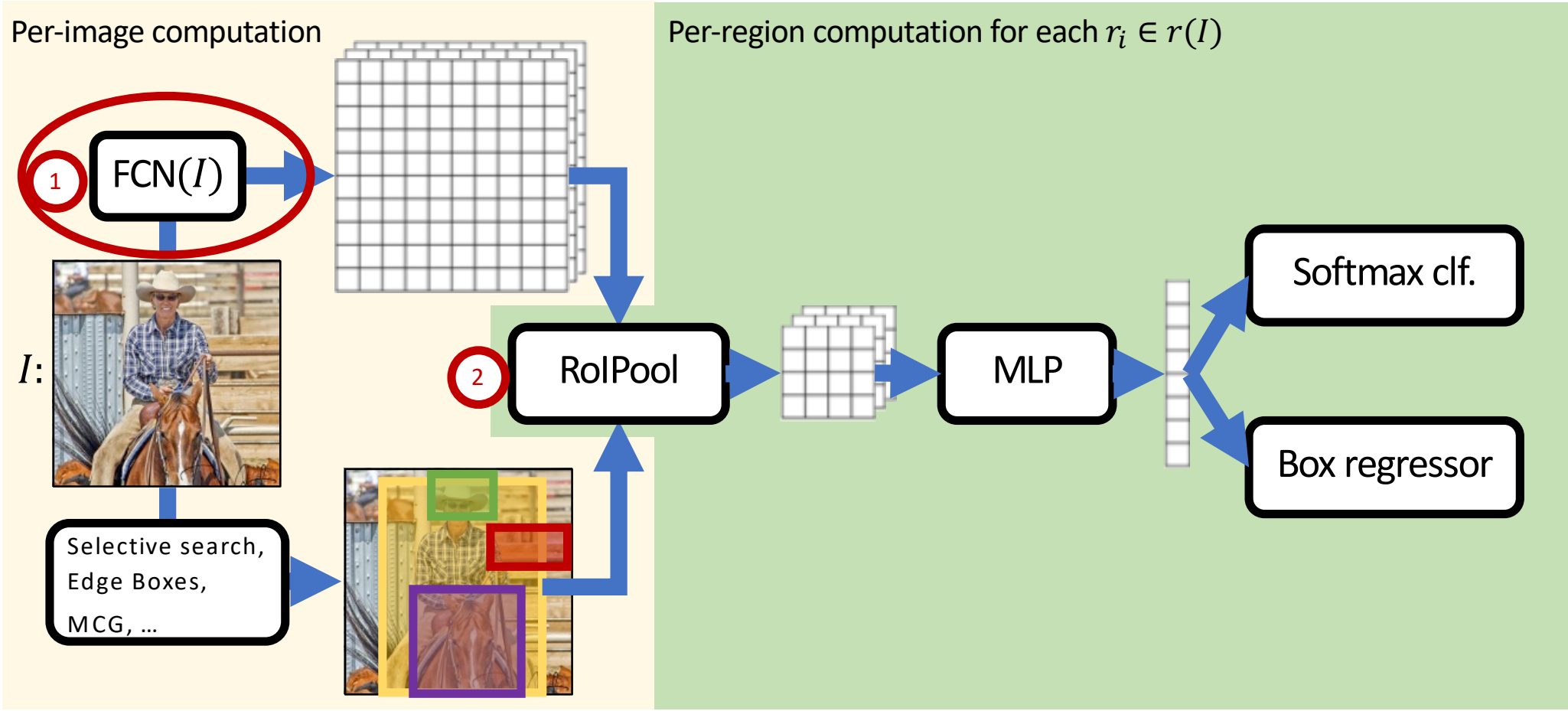


Girshick. Fast R-CNN. ICCV 2015.

Generalized R-CNN → Fast R-CNN



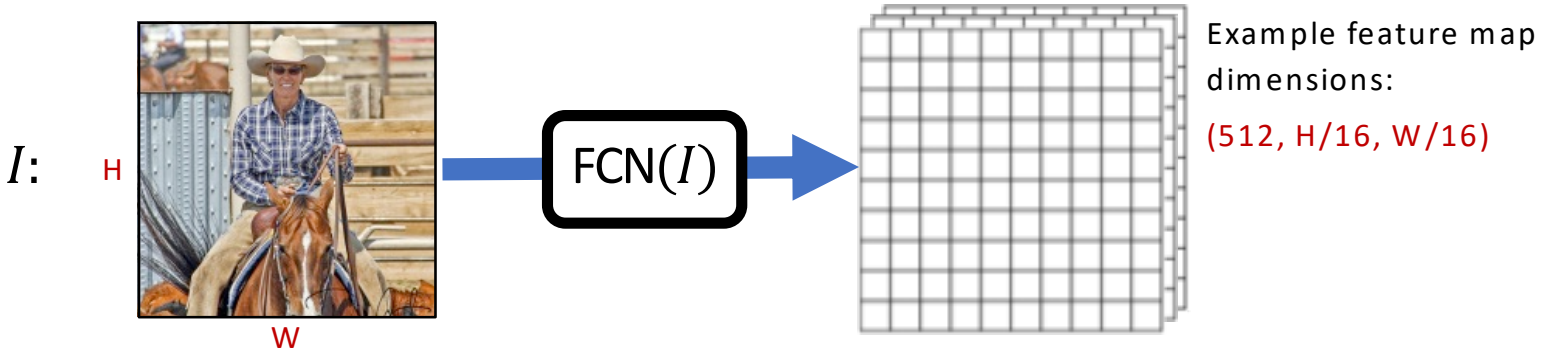
Fast R-CNN


Per-image computation
Per-region computation for each $r_i \in r(I)$
1
FCN(I)
I:
2
RoIPool
MLP
Softmax clf.
Box regressor
Selective search,
Edge Boxes,
MCG, …

Whole-image, Fully Conv. Network (FCN)

Use any standard ConvNet as the "backbone architecture"

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

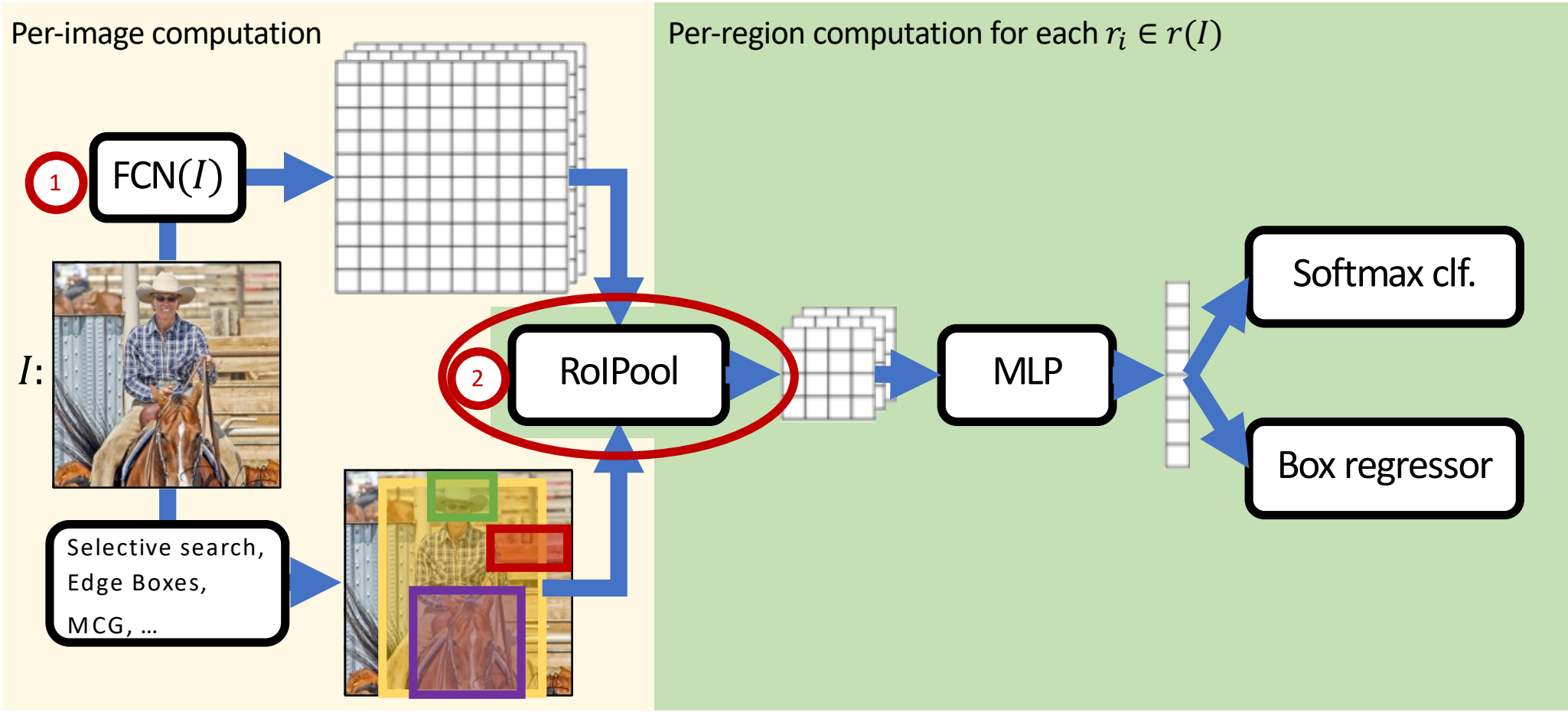


The Engine of Recognition

A good network lift all boats (“features matter”)

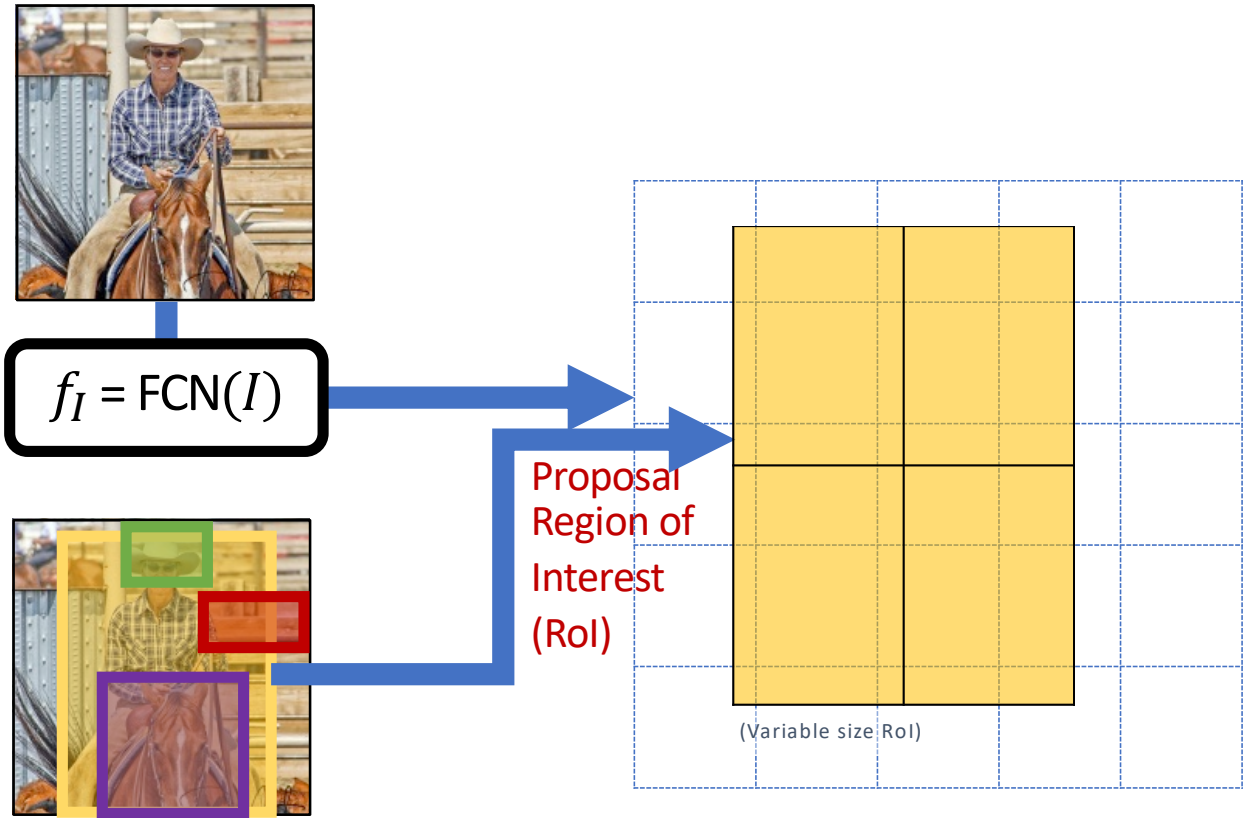
- AlexNet
- VGG
- GoogleNet / Inception
- ResNet
- ResNeXt
- …

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.

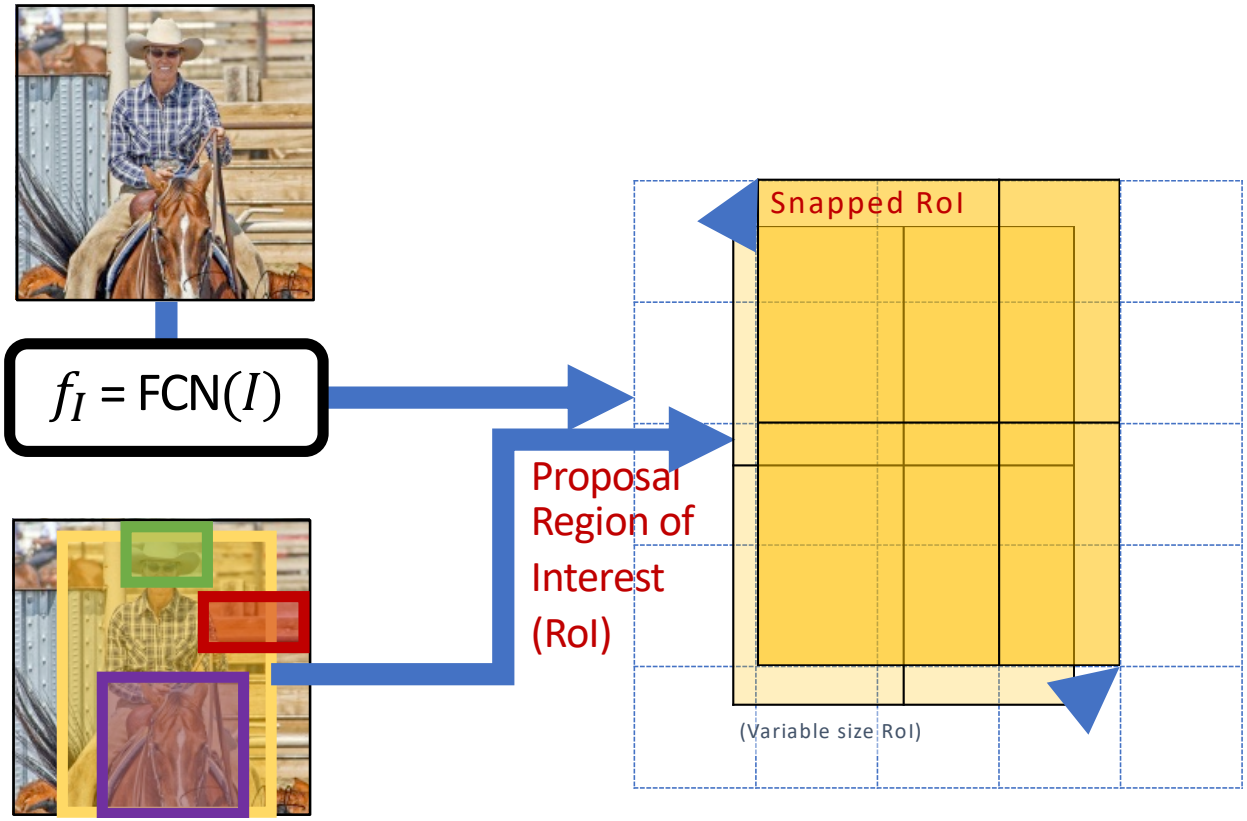
RoIPool (on each Proposal)



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 (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 (on each Proposal)

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

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

Proposal Region of Interest (RoI)

Snapped RoI

(Variable size RoI)

RoIPool transform

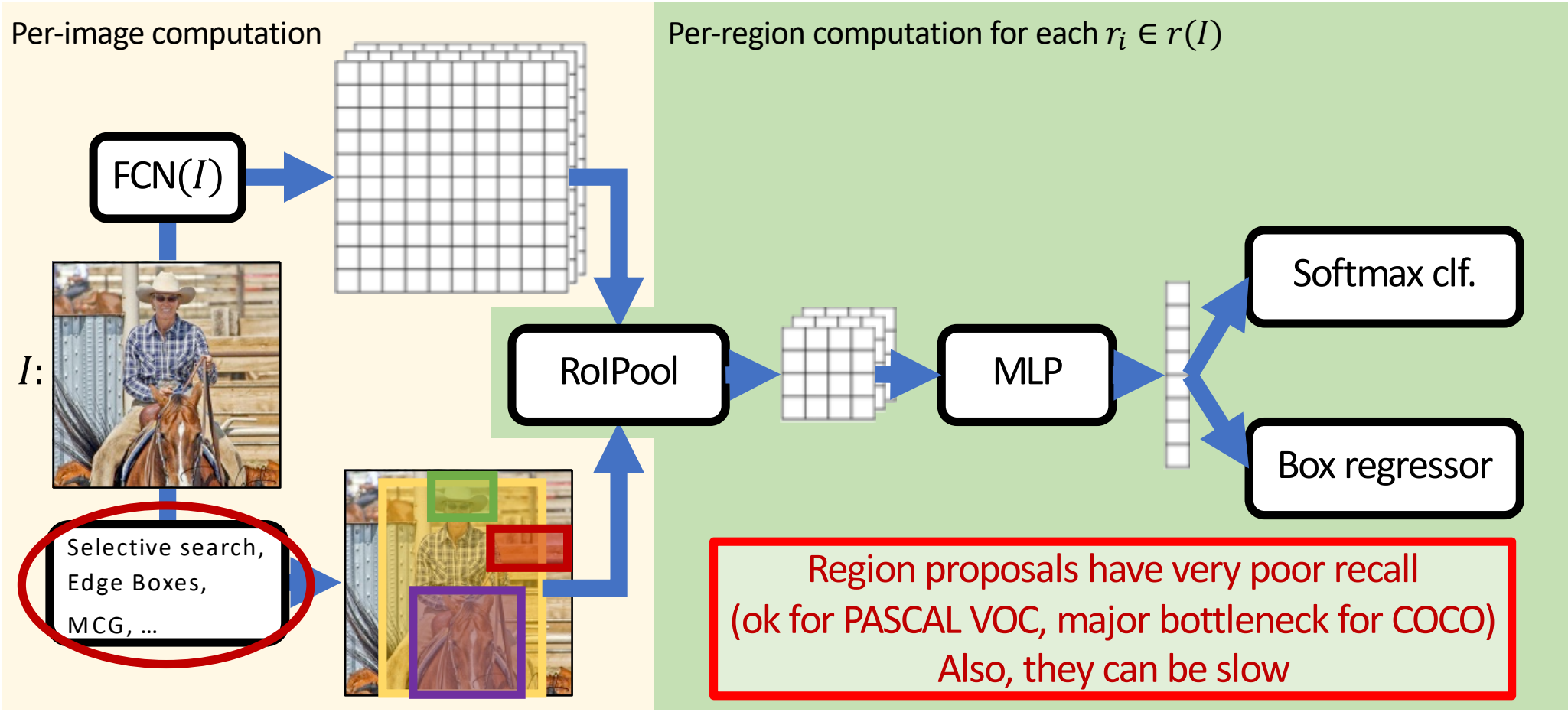
(Fixed dimensional representation)

MLP

Feature value is max over input cells

The Problem with Fast R-CNN

The Problem with Fast R-CNN



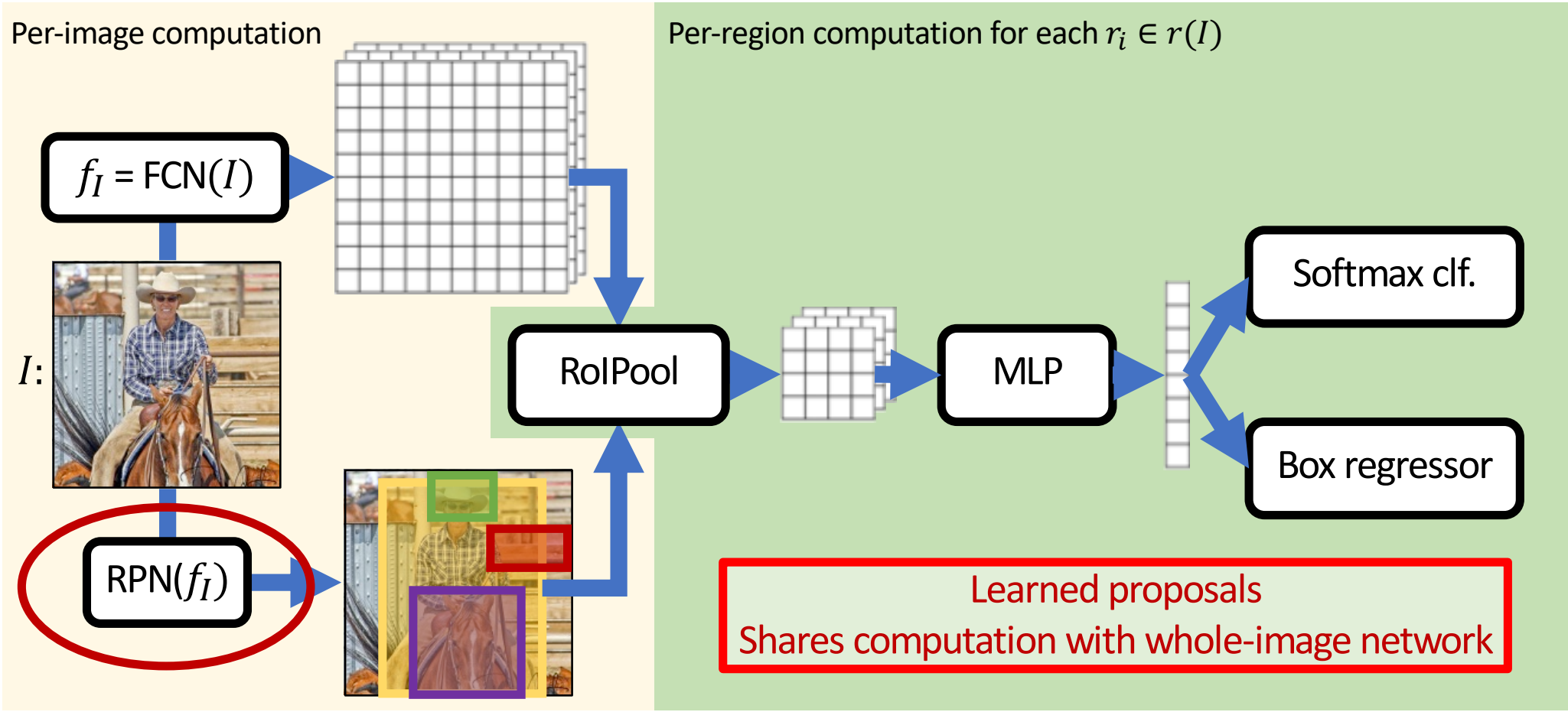
Faster R-CNN

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.

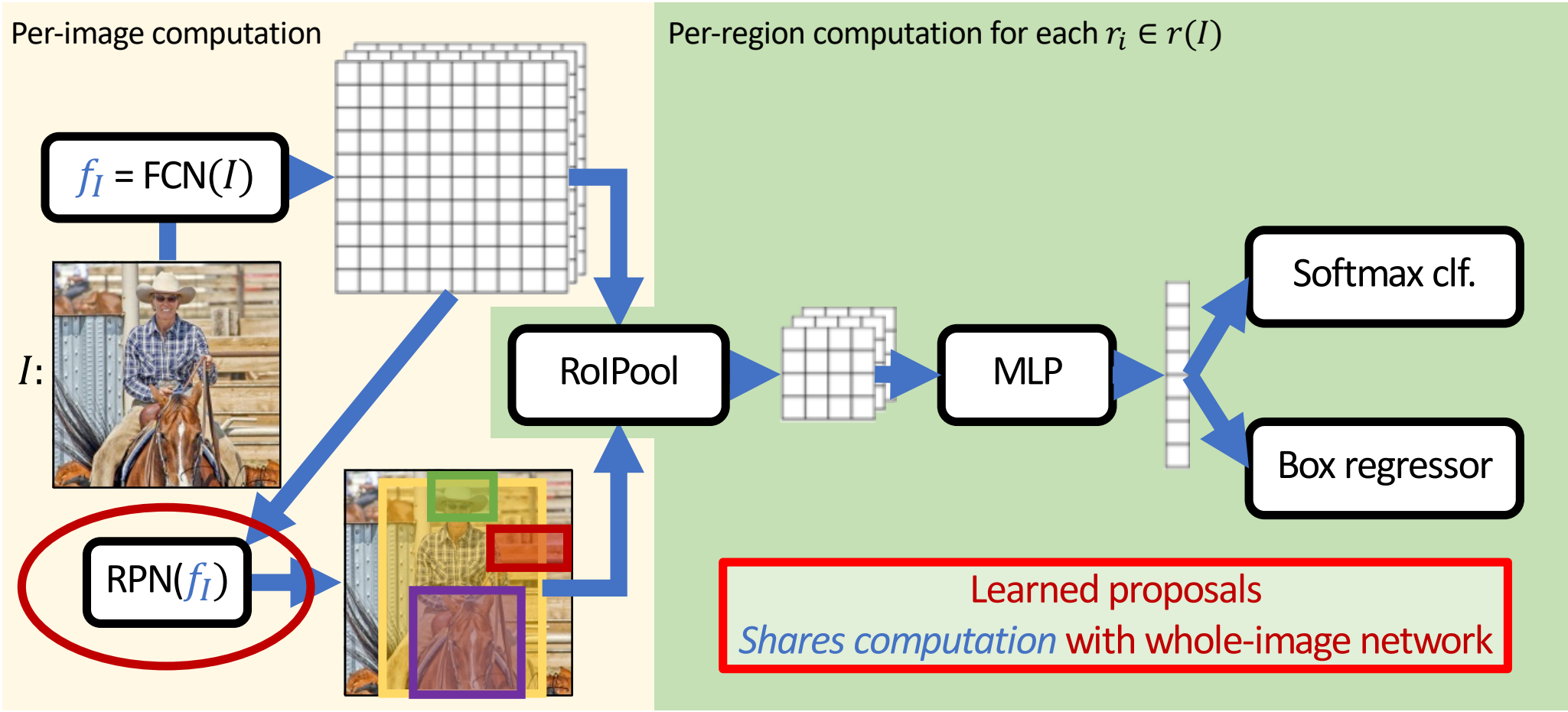
Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN

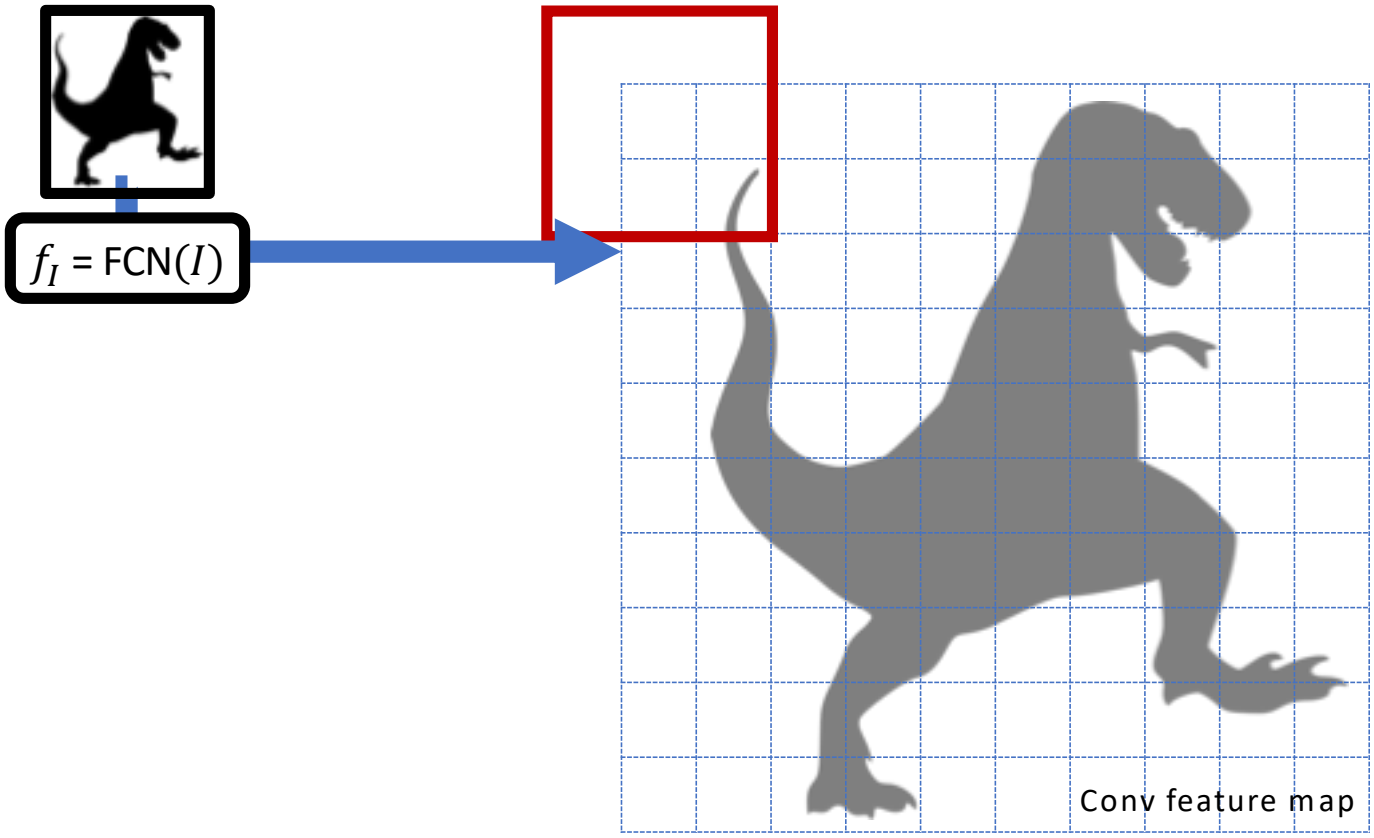


Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

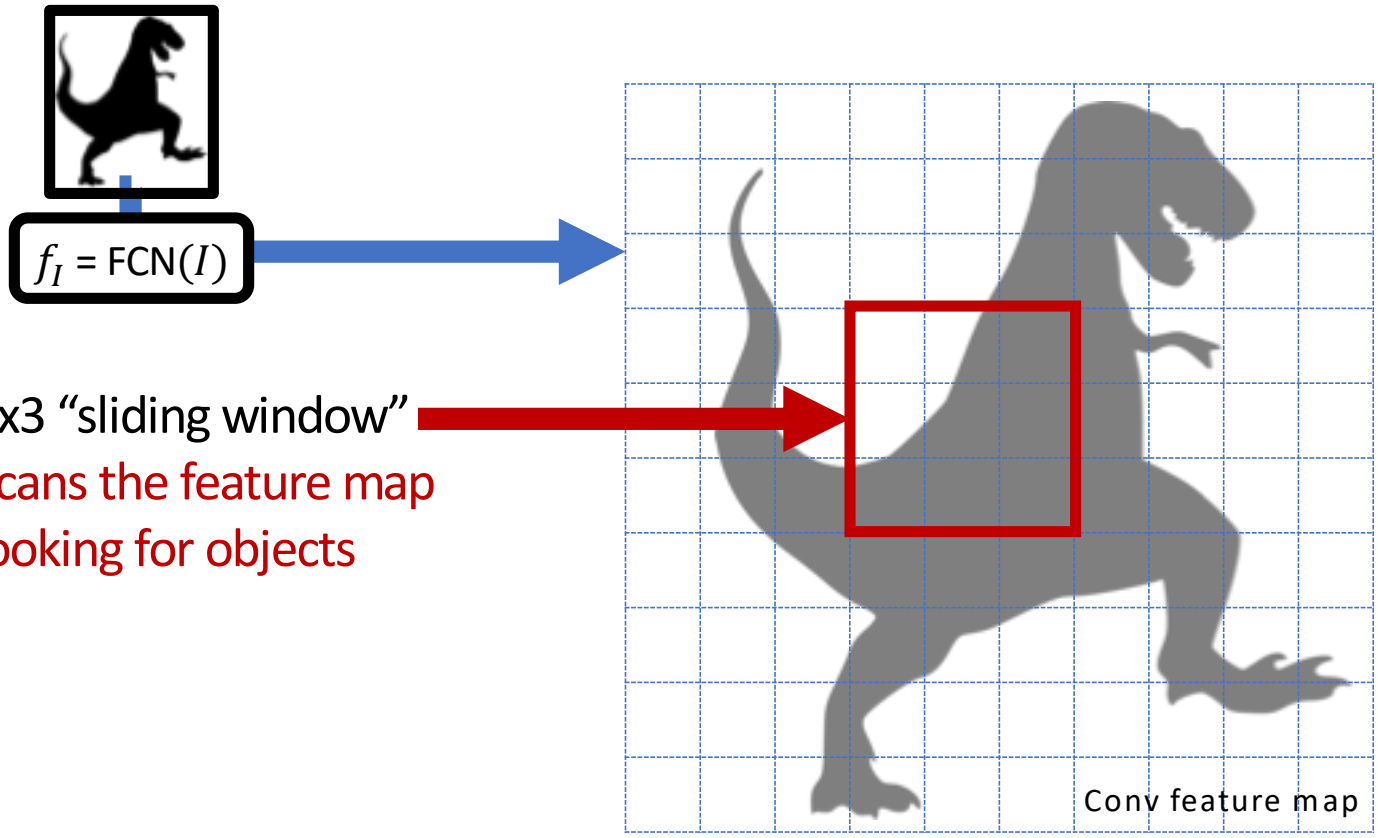
Faster R-CNN



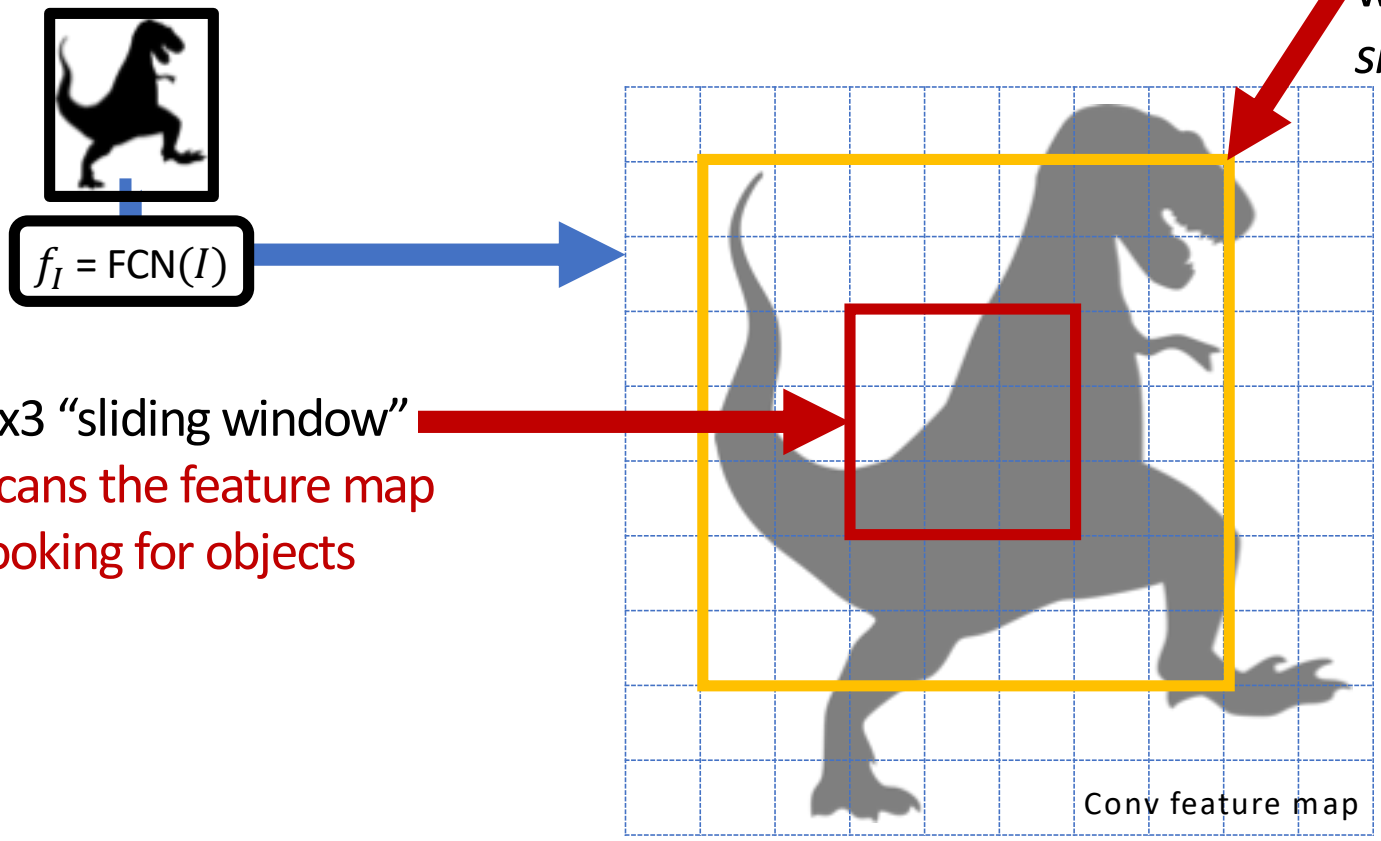
RPN: Region Proposal Network



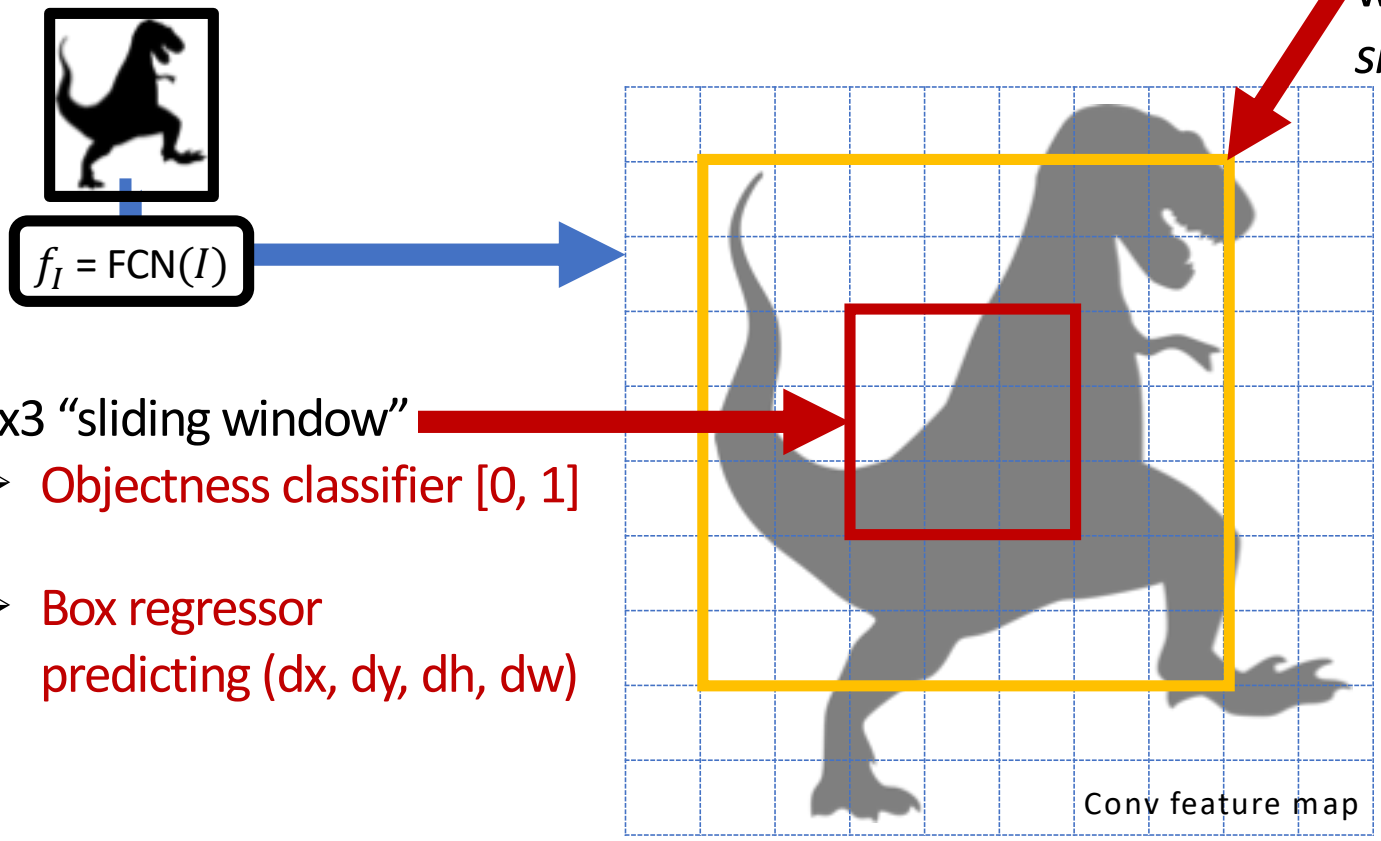
RPN: Region Proposal Network



RPN: Anchor Box



RPN: Anchor Box

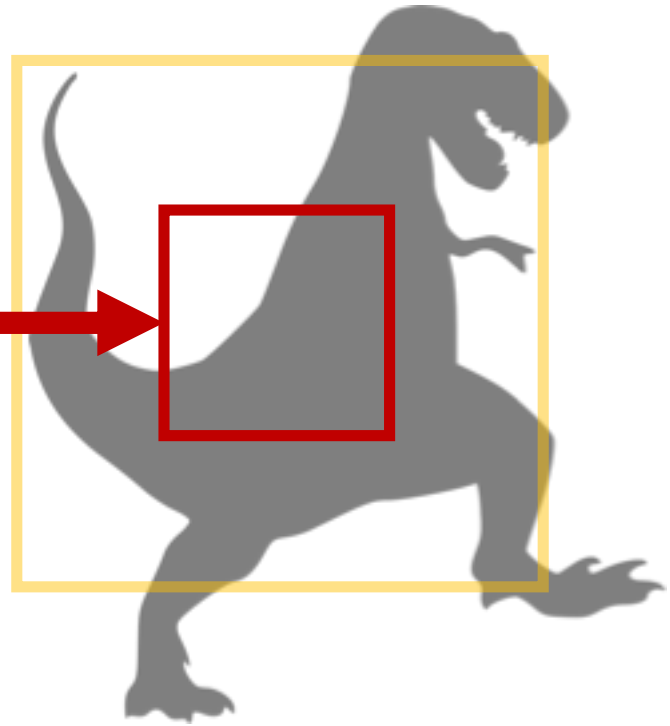


- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

RPN: Prediction (on object)

Objectness score

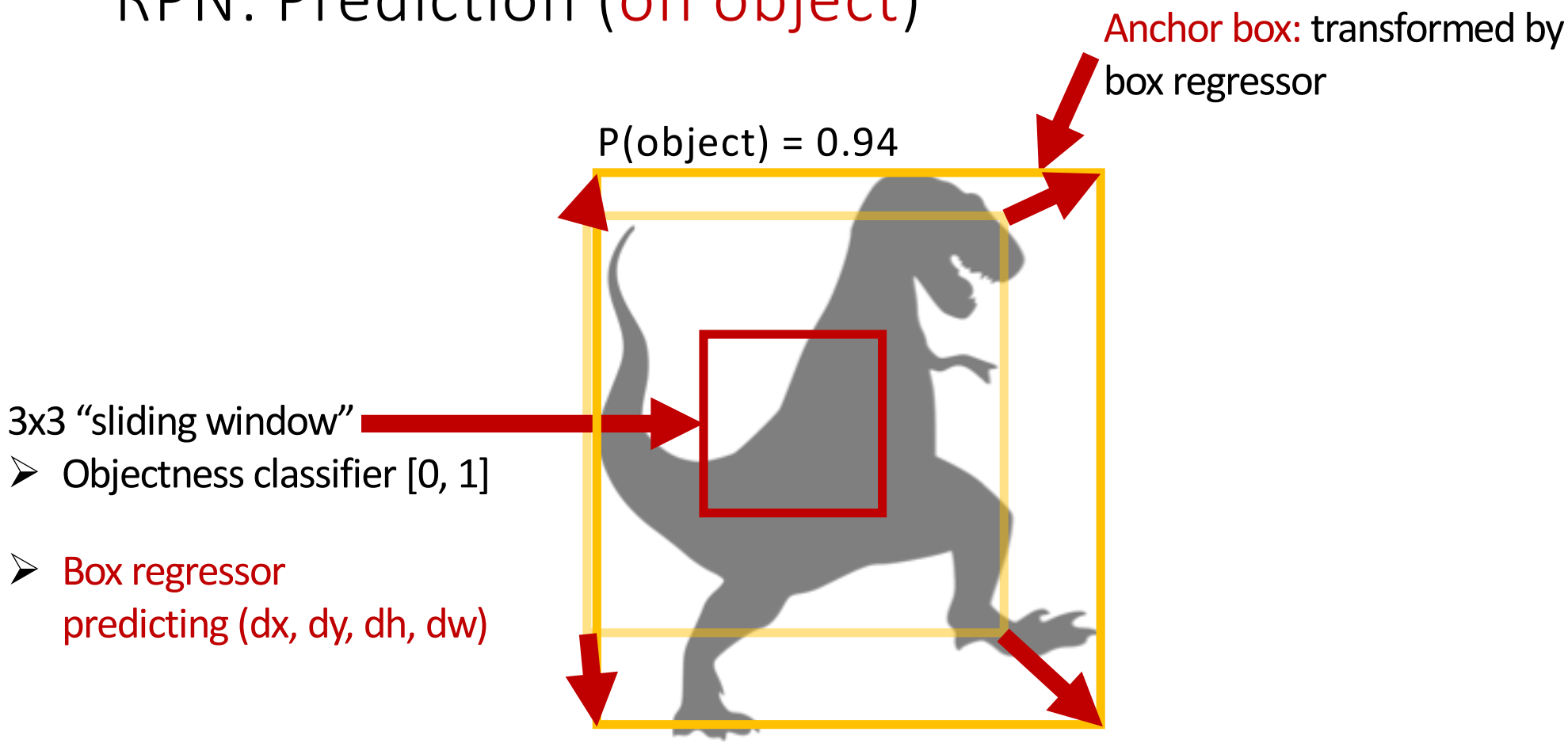
P(object) = 0.94



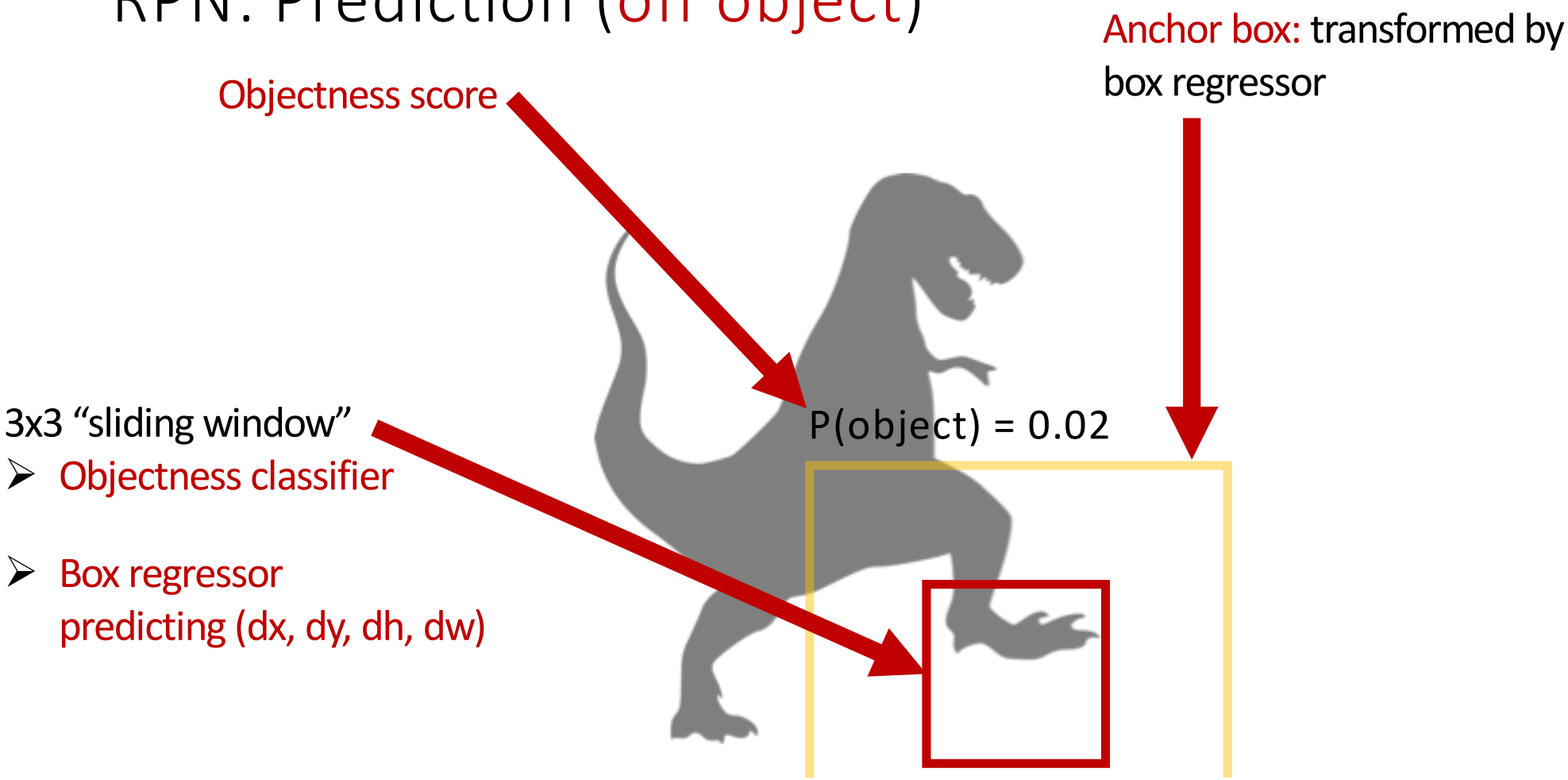
3x3 “sliding window”

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

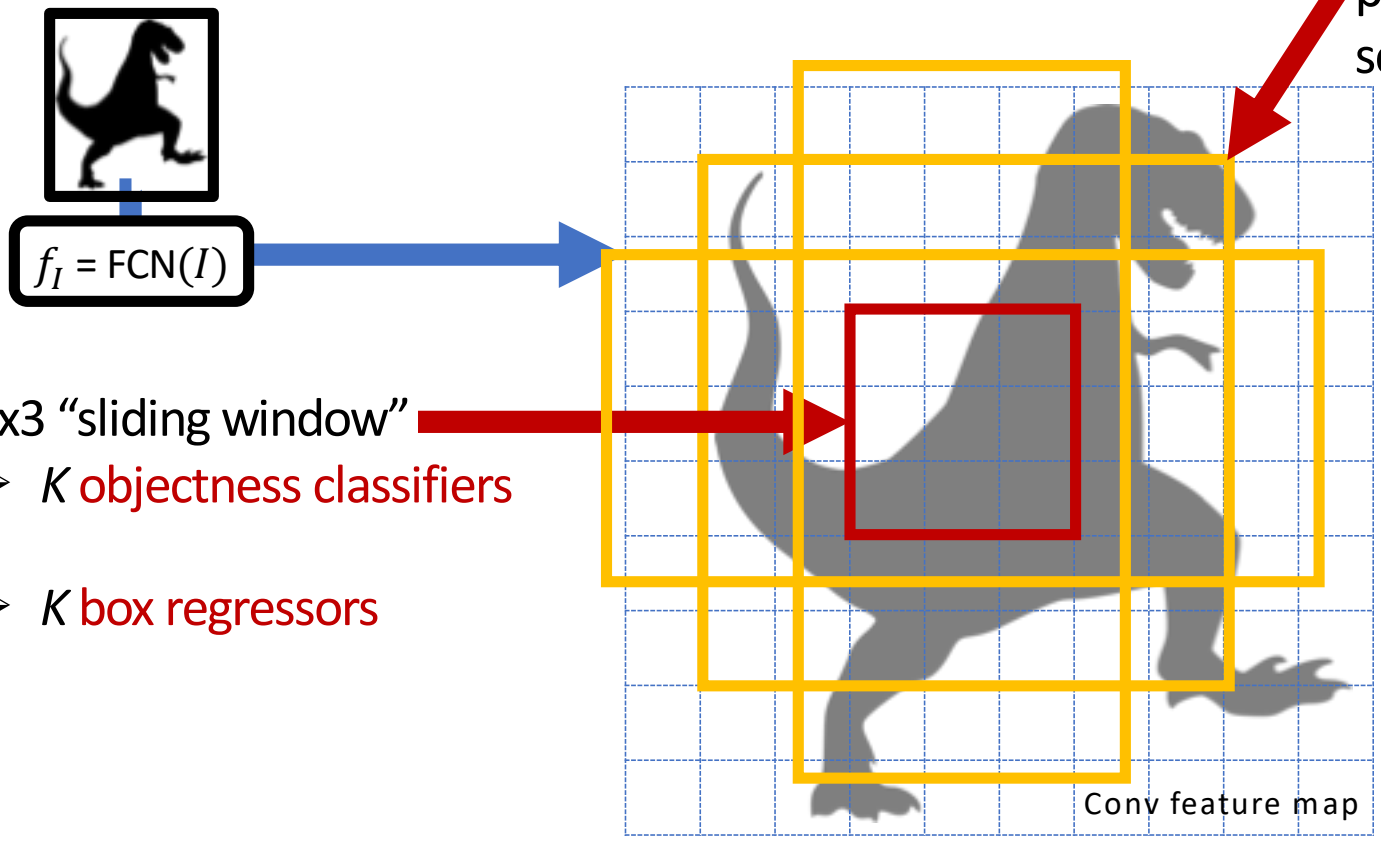
RPN: Prediction (on object)



RPN: Prediction (off object)

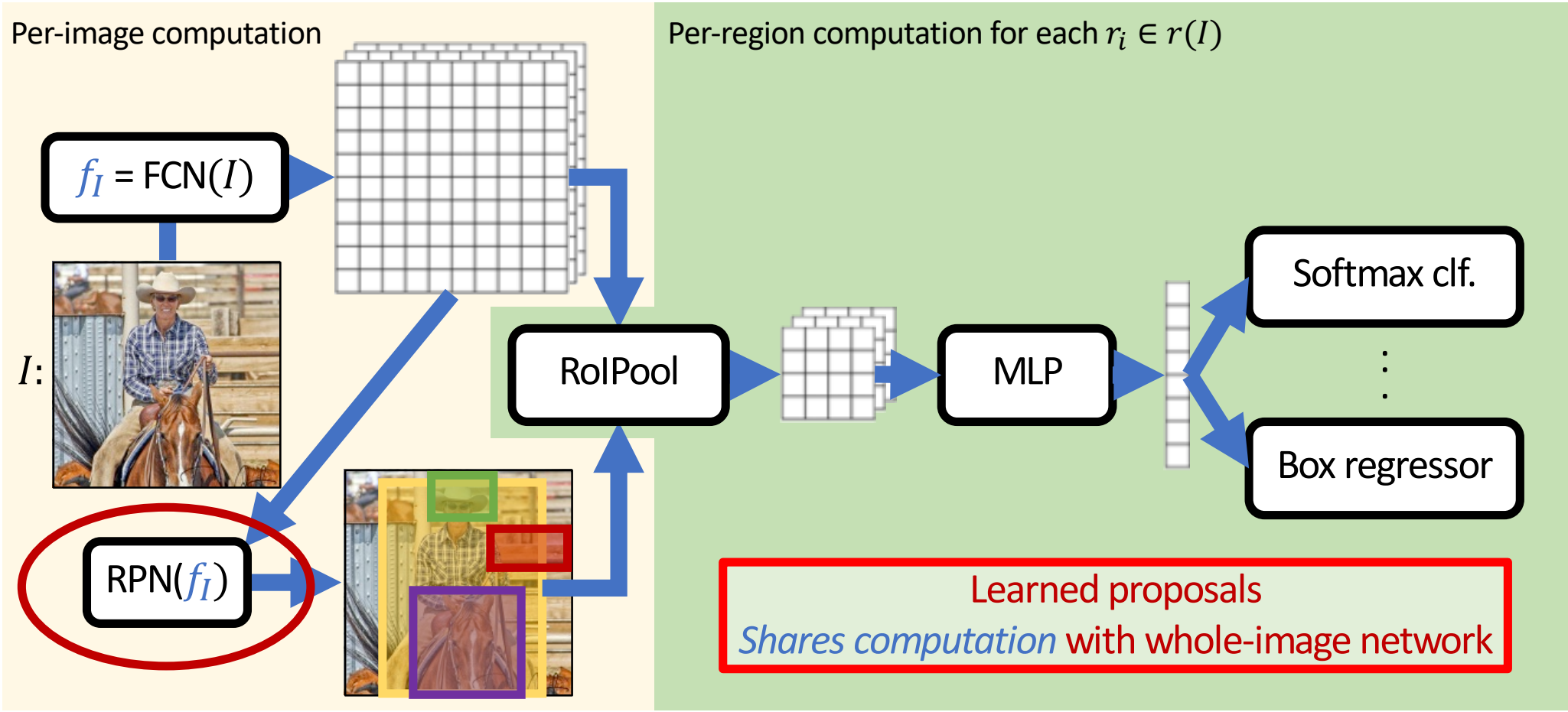


RPN: Multiple Anchors



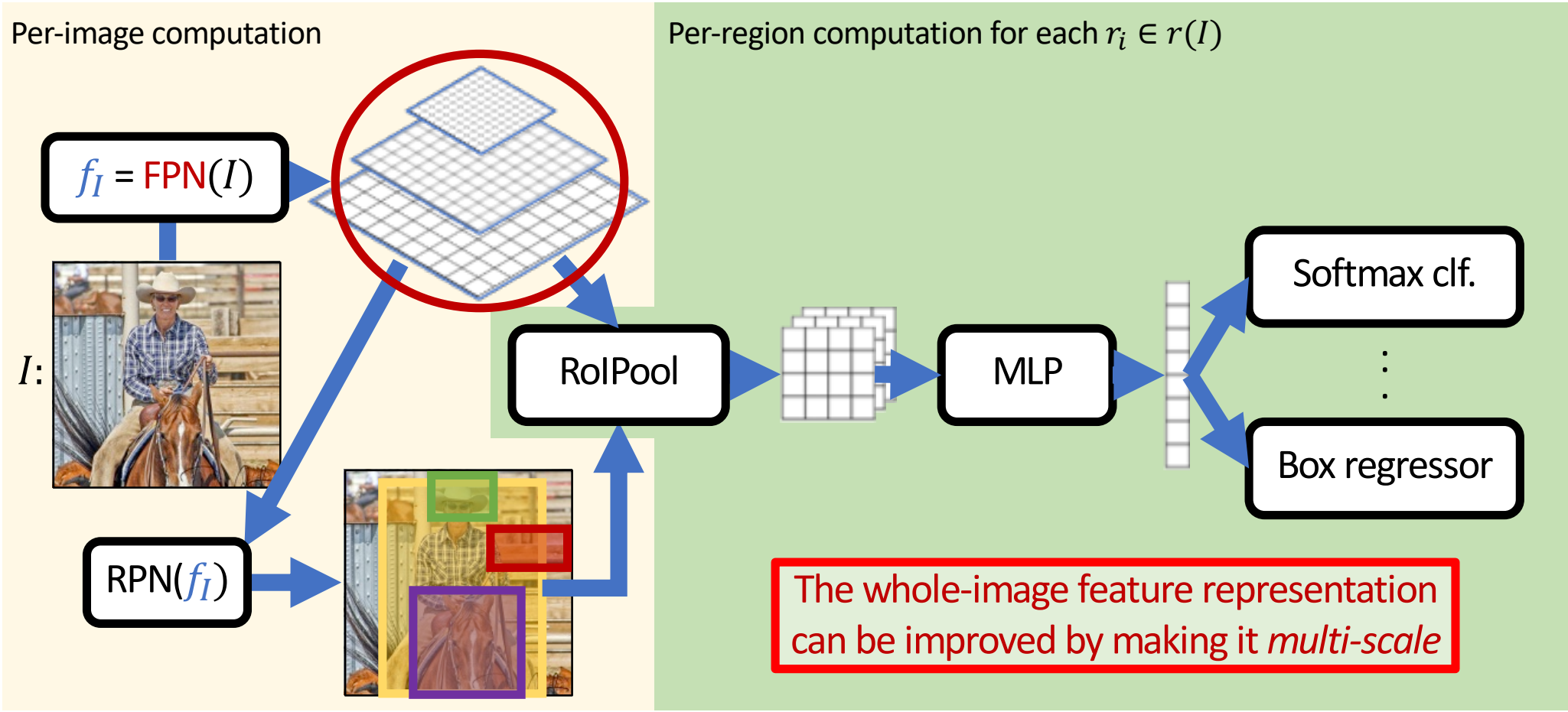
Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN



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

Faster R-CNN with a Feature Pyramid Network

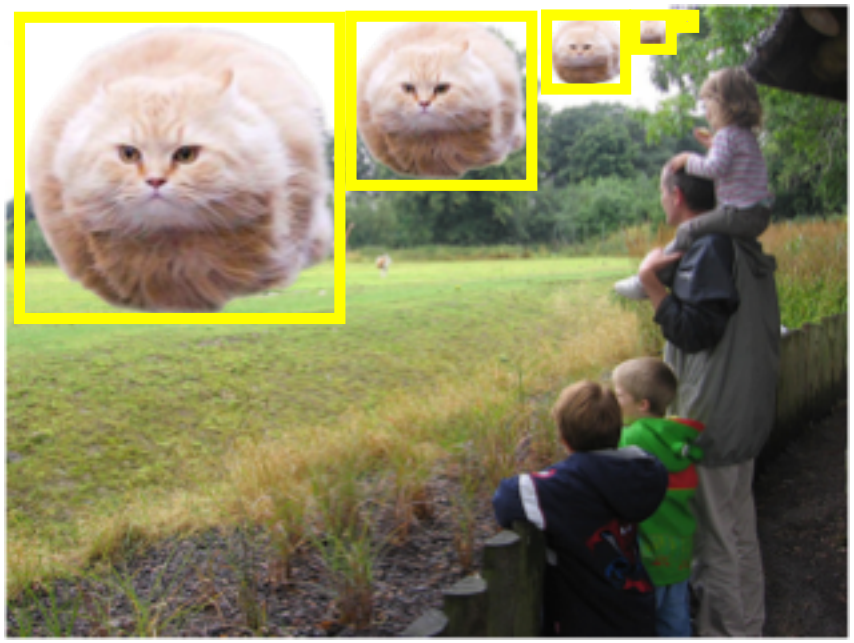


Feature Pyramid Network (FPN)

References

- O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In MIC- CAI, 2015.
- P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In ECCV, 2016.
- W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. SSD: Single shot multibox detector. In ECCV, 2016.
- A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. arXiv:1612.06851, 2016.
- T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie. Feature Pyramid Networks for Object Detection. In CVPR, 2017.

FPN: Improving Scale Invariance and Equivariance



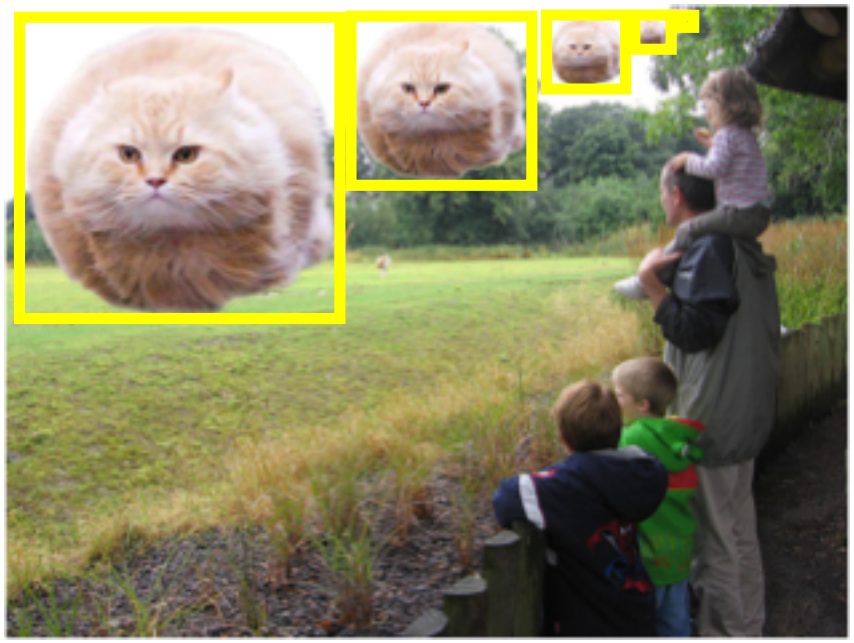
Detectors need to

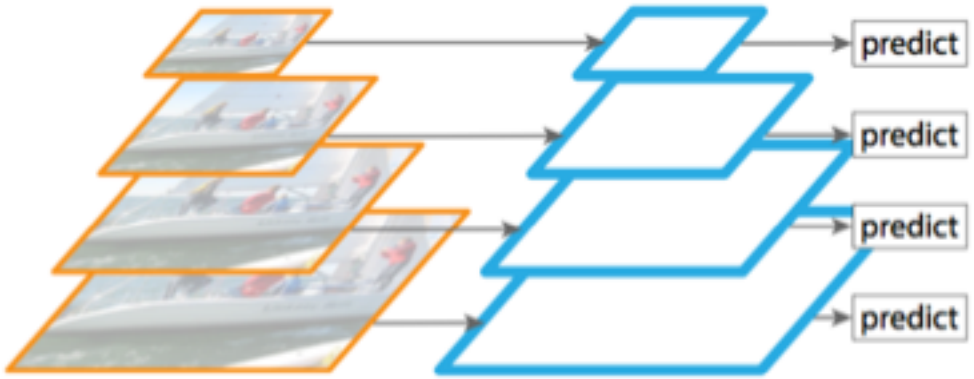
1. classify and
2. localize

objects over a wide range of scales

FPN improves this ability

Strategy 1: Image Pyramid



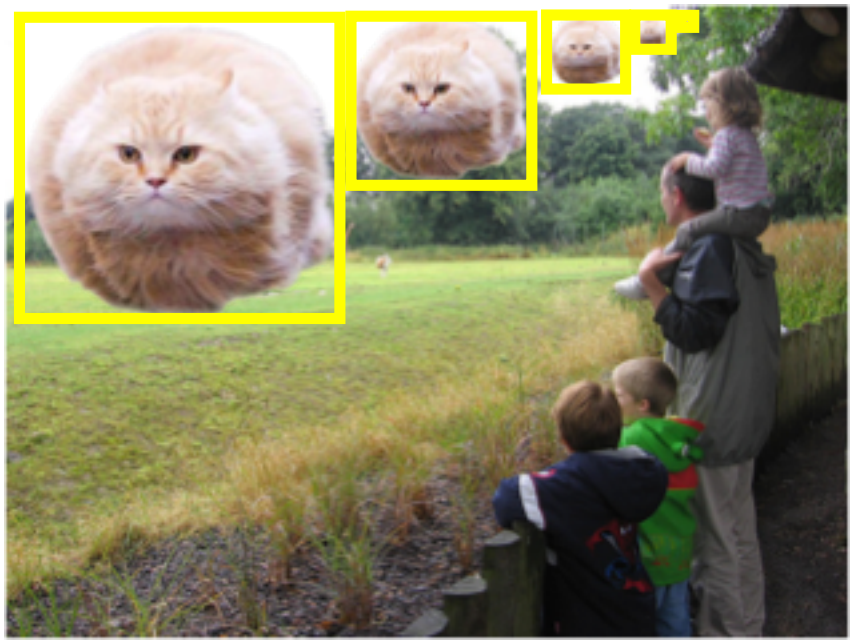


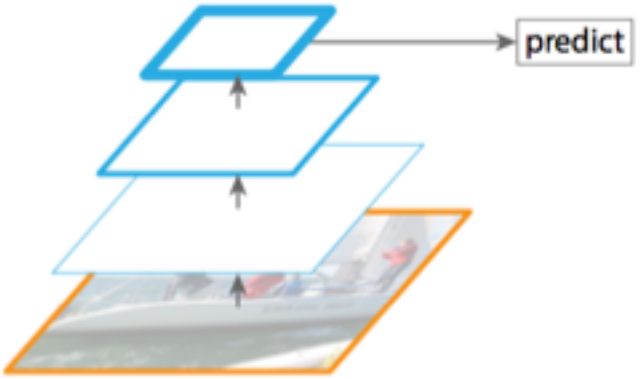
(a) Featurized image pyramid

Standard solution – *slow!*

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

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



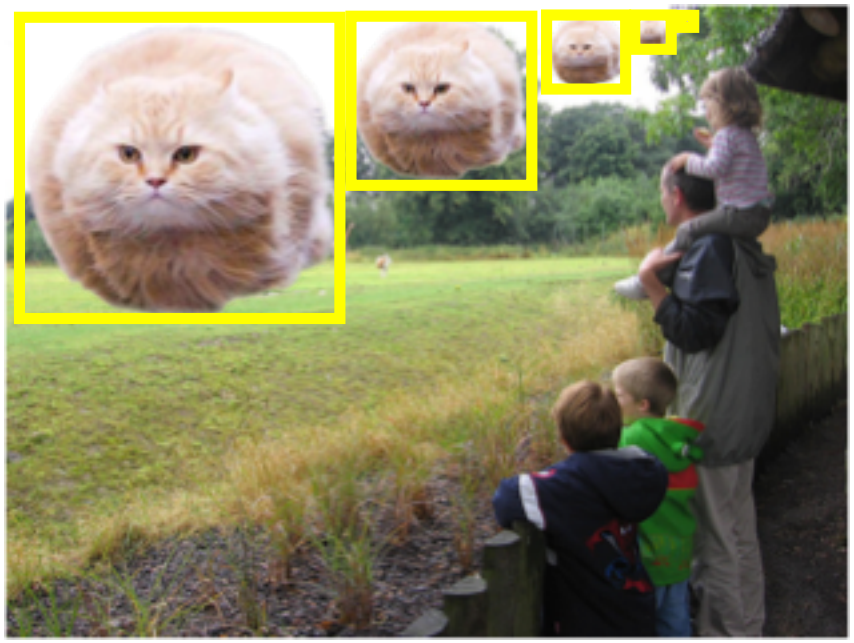


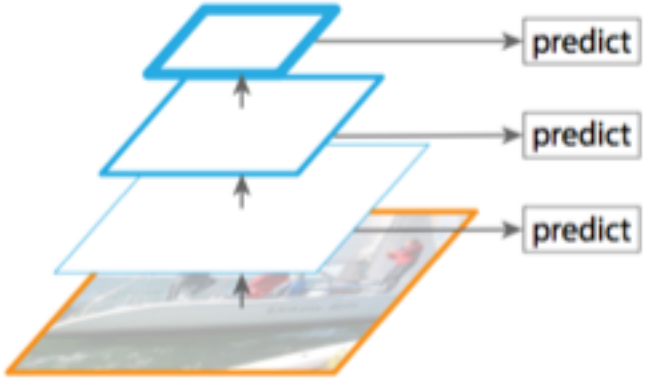
(b) Single feature map

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

(E.g., Fast/er R-CNN, YOLO, …)

Strategy 3: Naïve In-network Pyramid



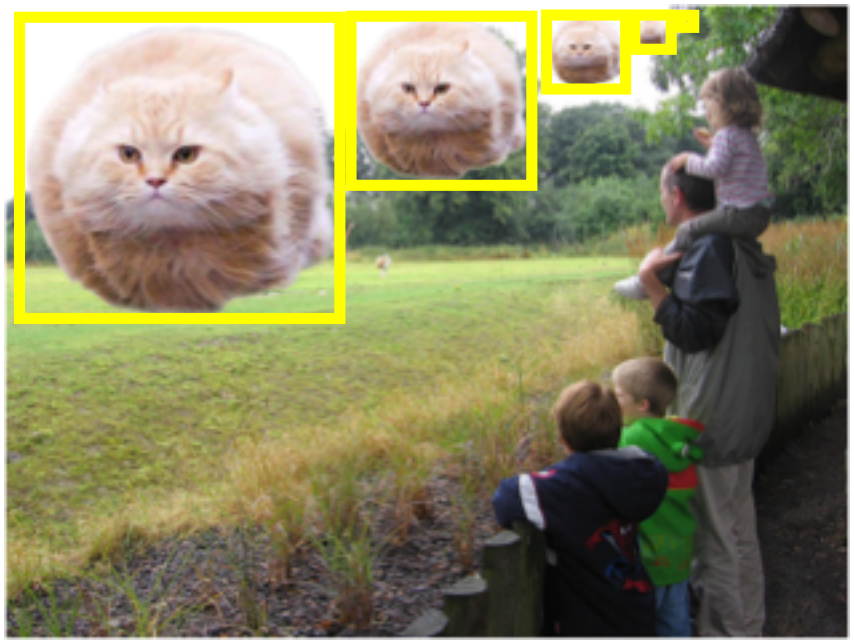


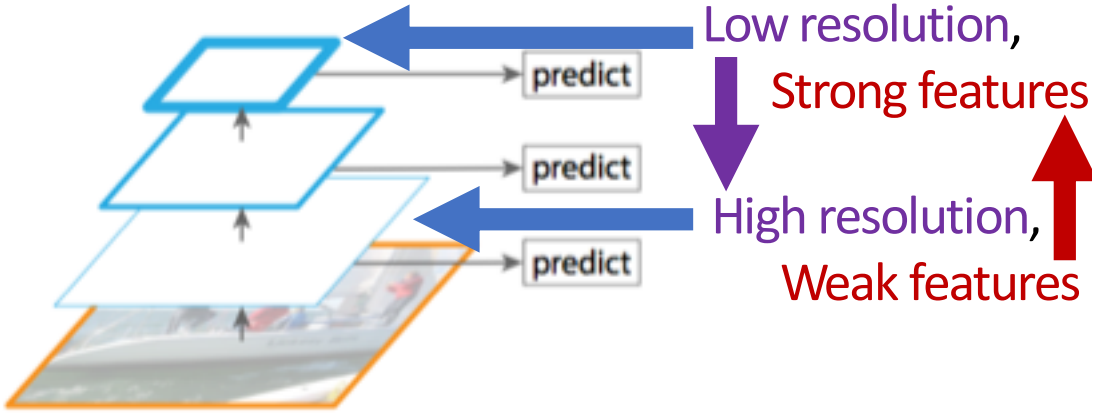
(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 3: Naïve In-network Pyramid



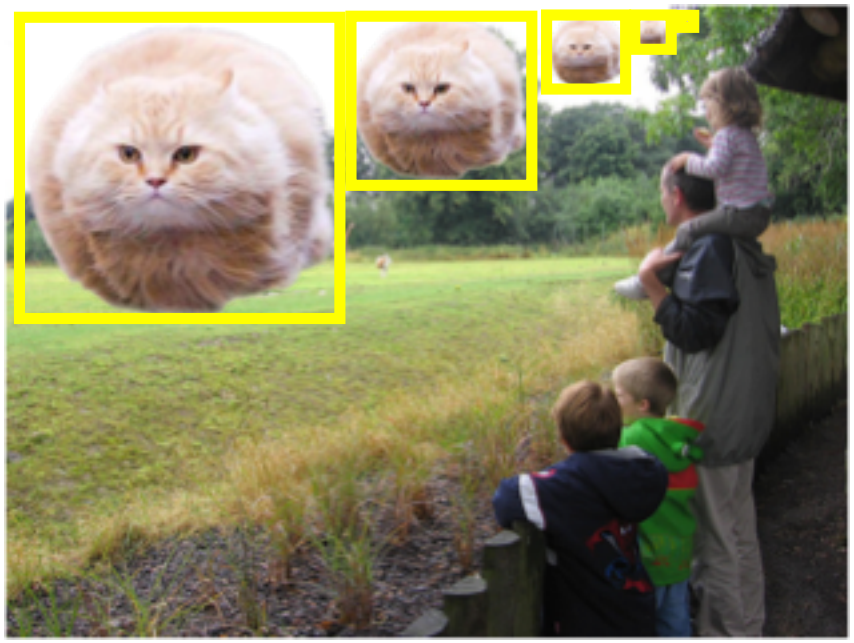


(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 4: Feature Pyramid Network

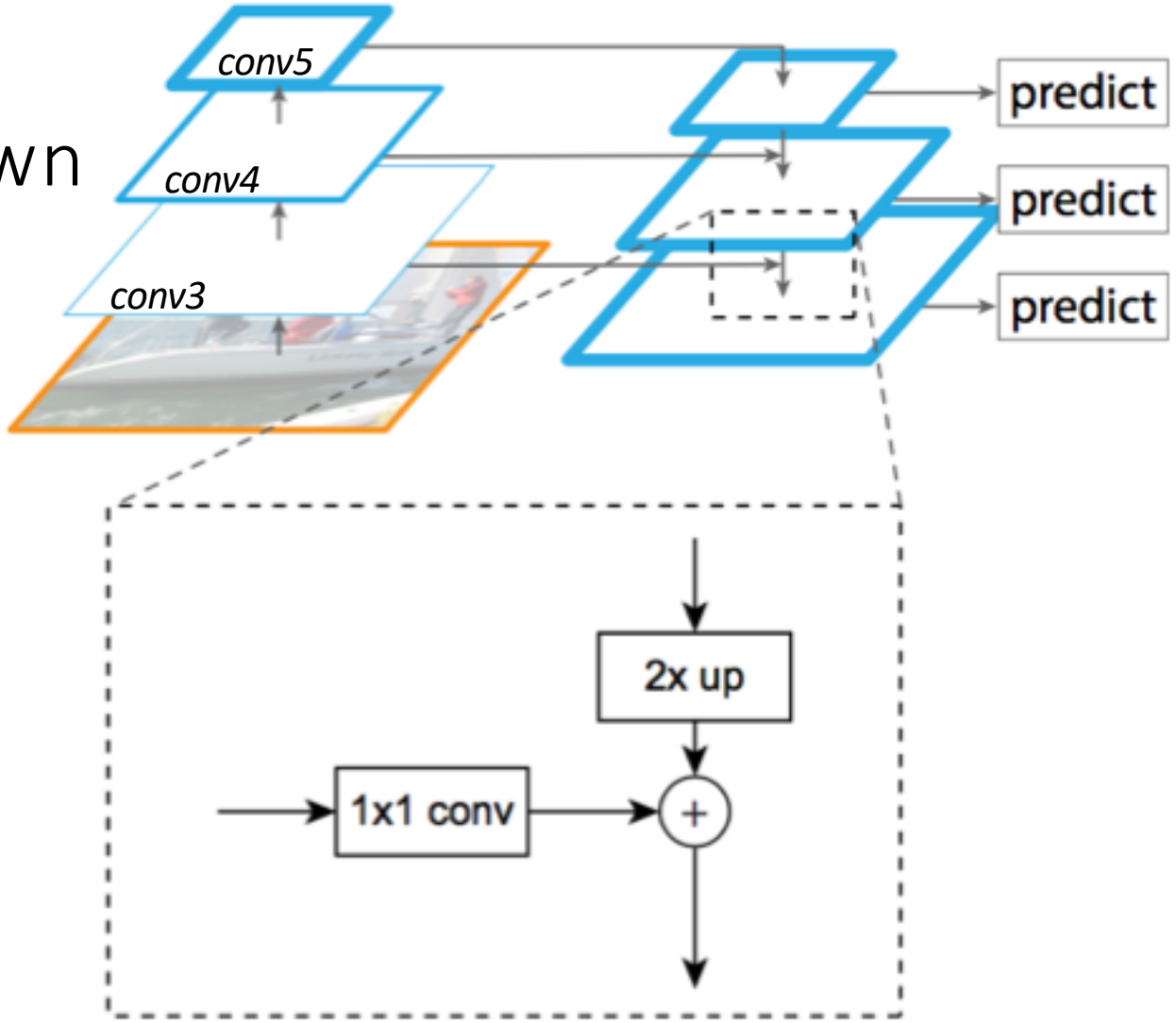


(d) Feature Pyramid Network

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

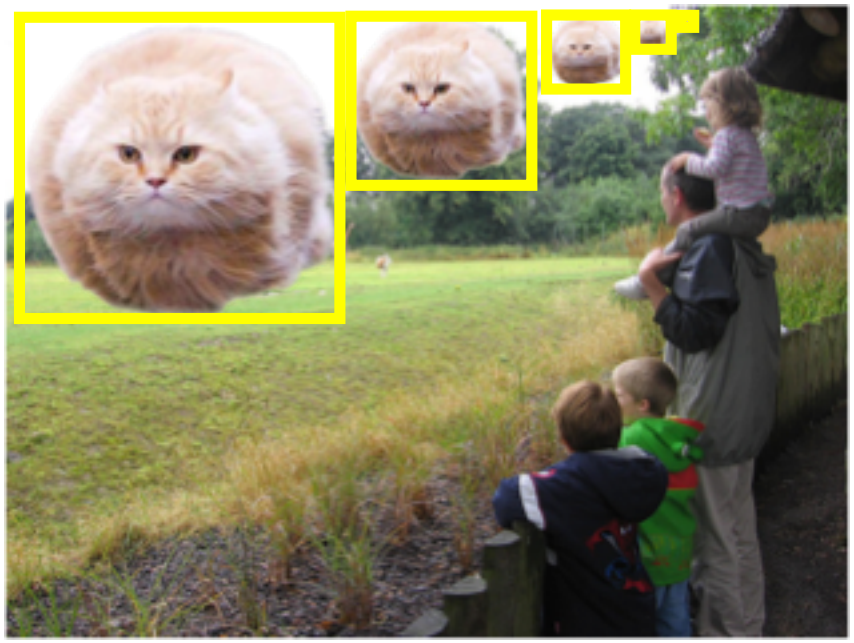
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

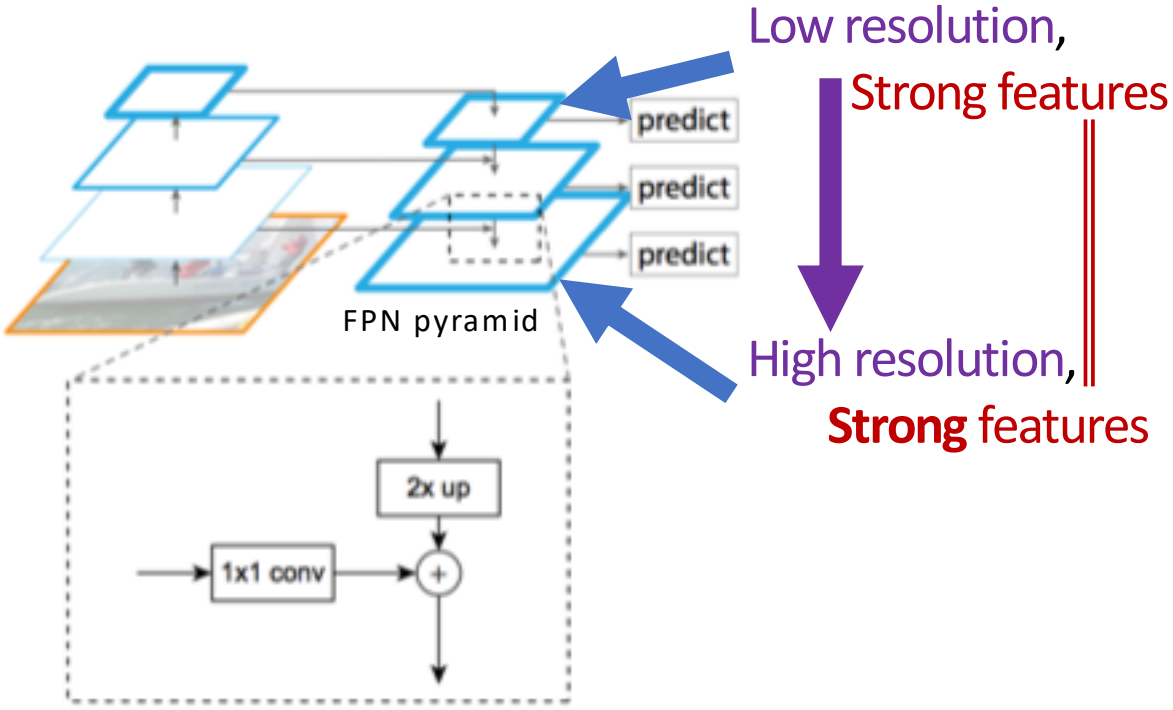
FPN Top-down Refinement Module



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

No Compromise on Feature Quality, still Fast





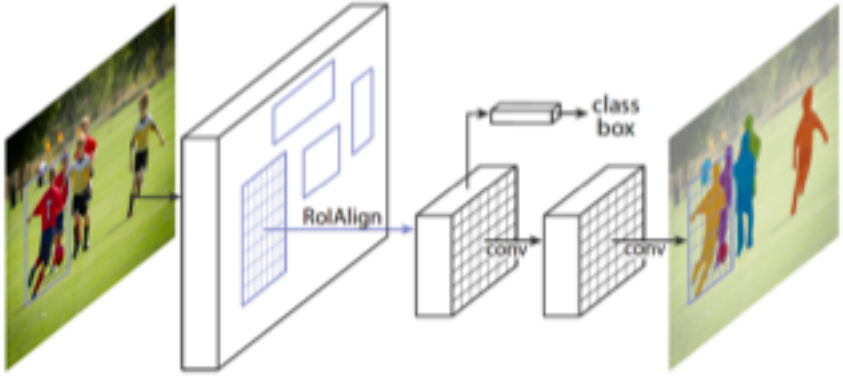
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017. See also: Shrivastava's TDM.

FPN – A Generic Backbone Modification

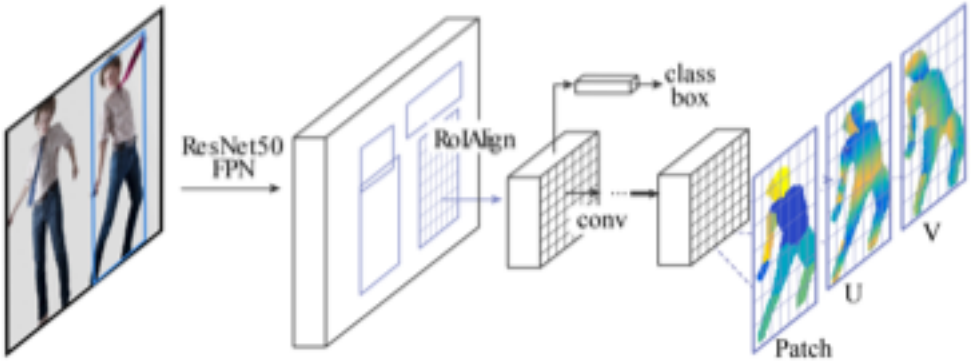
Generates a feature pyramid—*useful in many applications!*

- RPN
- Fast/er R-CNN
- Mask R-CNN
- RetinaNet
- FPN-based semantic segmentation FCN

Generalized R-CNN: Adding More Heads

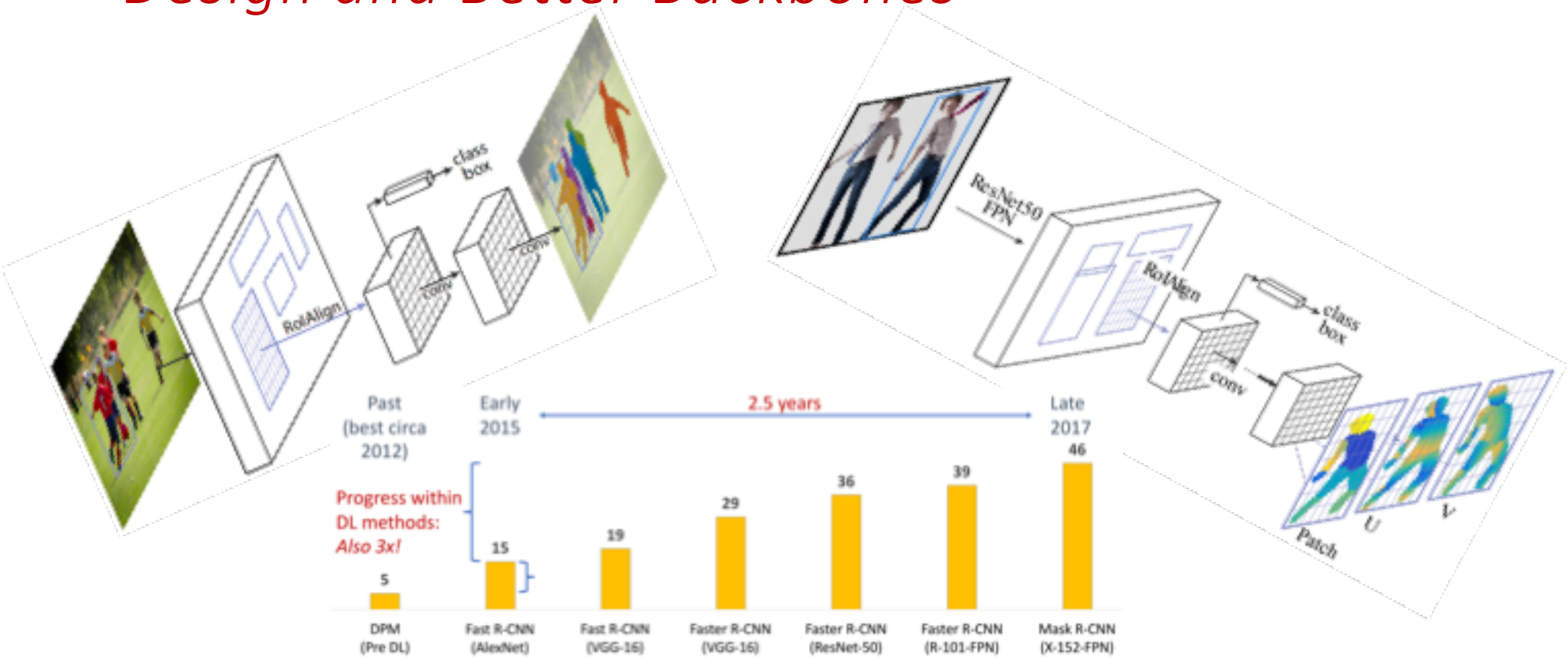


Mask R-CNN
[He, Gkioxari, Dollár, Girshick]



DensePose
[Güler, Neverova, Kokkinos]

Summary: Progress through Better Detector Design and Better Backbones



Takeaway Points

1. Detection is not solved – *datasets are just a model of the world*
2. Features matter – *the backbone net is the engine of recognition*
3. Detector design matters – *progress from domain knowledge*
4. Flexible / extensible framework with OSS code

https://github.com/facebookresearch/Detectron

Additional Material Not Covered in 2018

Topics to cover

- Object detection intro (very brief)
- The Generalized R-CNN framework
- One-stage vs. multi-stage detectors & speed/accuracy tradeoffs

Stages: What and Why?

Detection output space: $N = H \times W$ pixel image has $O(N^2)$ boxes

Output space is HUGE, even for small images it's *billions of boxes*

Number of target objects is small, say 10

Massive foreground / background class imbalance

How do we Deal with Class Imbalance?

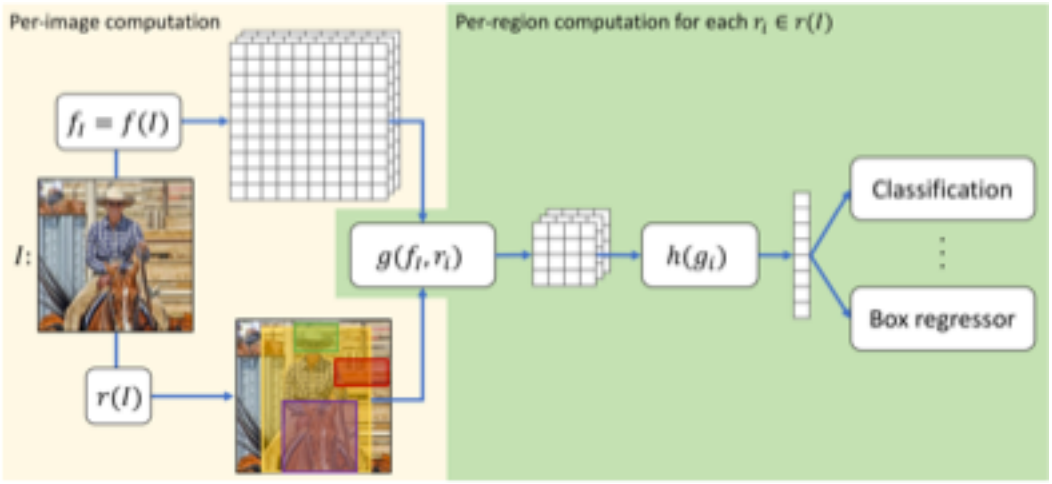
1. Subsample the output space while maintaining *high recall*

- Sliding window
- Object proposals

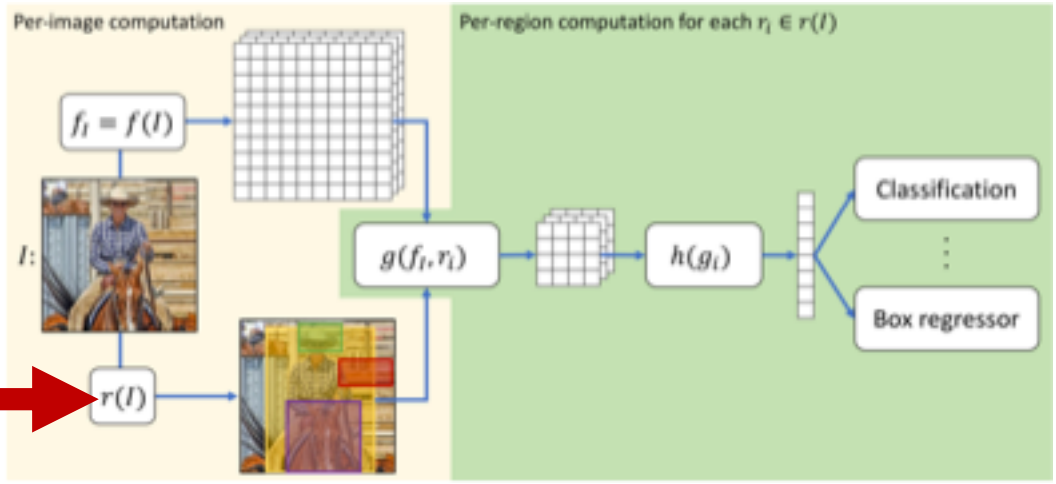
2. Classify boxes in a *cascade of stages*

- Most famous example: Viola-Jones detector
- Typically combined with subsampling
- *Note that subsampling is already an implicit stage*

Example of two stages using proposals: Generalized R-CNN Framework

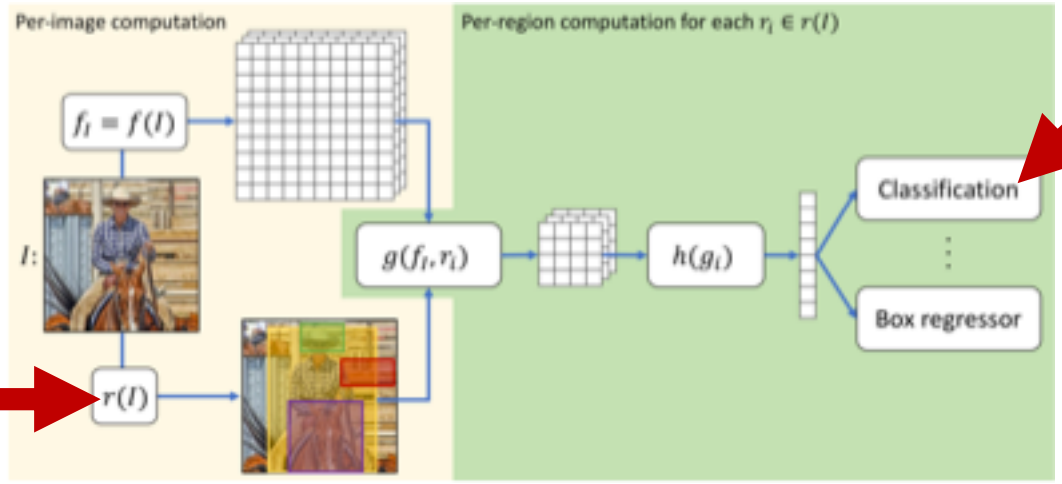


Example of two stages using proposals: Generalized R-CNN Framework



1. Proposals eliminate most of the output space

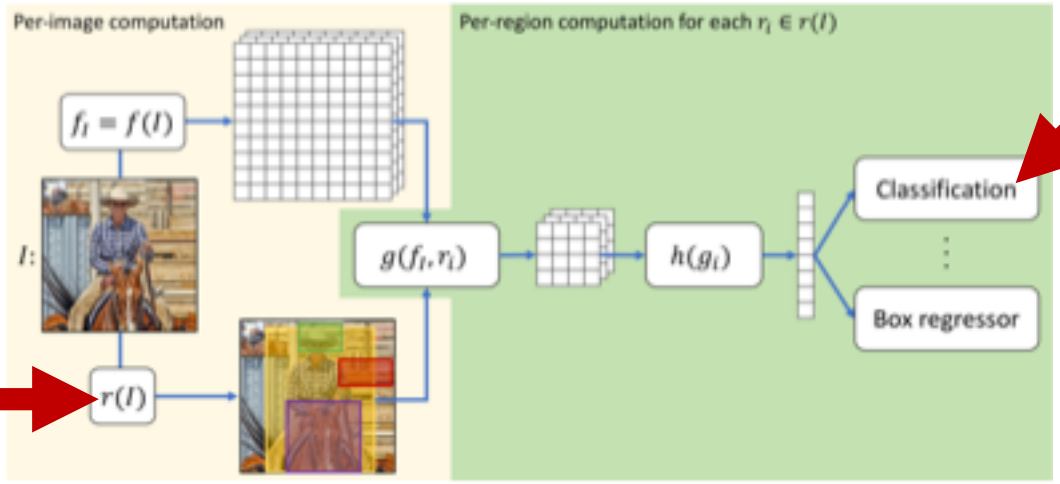
Example of two stages using proposals: Generalized R-CNN Framework



1. Proposals eliminate most of the output space

2. Classification on small subset of output space

Example of two stages using proposals: Generalized R-CNN Framework



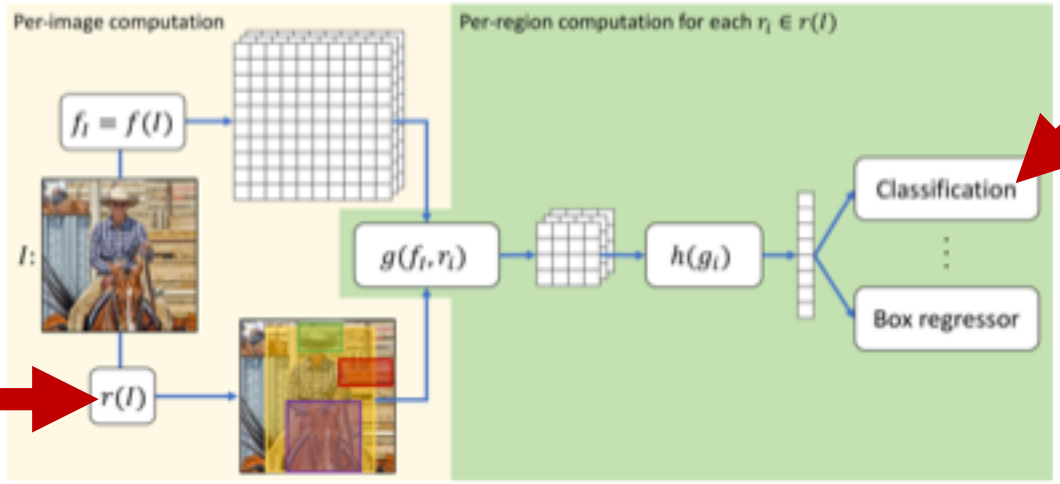
1. Proposals eliminate most of the output space

2. Classification on small subset of output space

Example one stage using *dramatic* subsampling

Redmond et al. You Only Look Once:
Unified Real-time Object Detection. CVPR 2016.

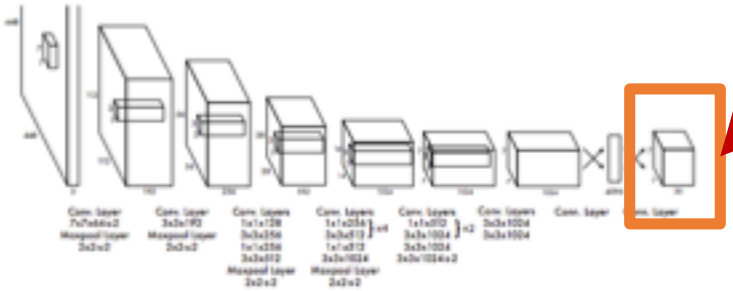
Example of two stages using proposals: Generalized R-CNN Framework



1. Proposals eliminate most of the output space

2. Classification on small subset of output space

Example one stage using *dramatic* subsampling



Redmond et al. You Only Look Once: Unified Real-time Object Detection. CVPR 2016.

1. Consider a *tiny* subset of the output space by design; directly classify this small set of boxes

"You only look once" = "Single shot" = "One stage"

Subsampled Output Space [+ Hard Example Mining]

Central challenge for one-stage detectors: class imbalance

- YOLOv1 – 98 boxes
- YOLOv2 – ~ 1k
- OverFeat – ~ 1-2k
- SSD – ~ 8–26k (hard-example mining)

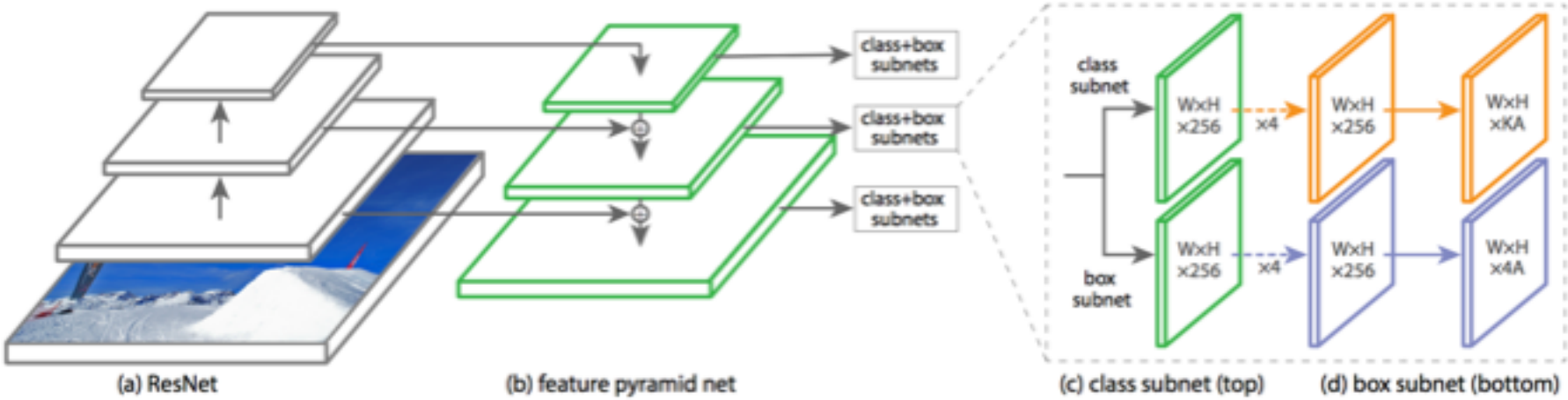
- RetinaNet – ~ 100-200k ("soft" hard-example mining)

RetinaNet

References

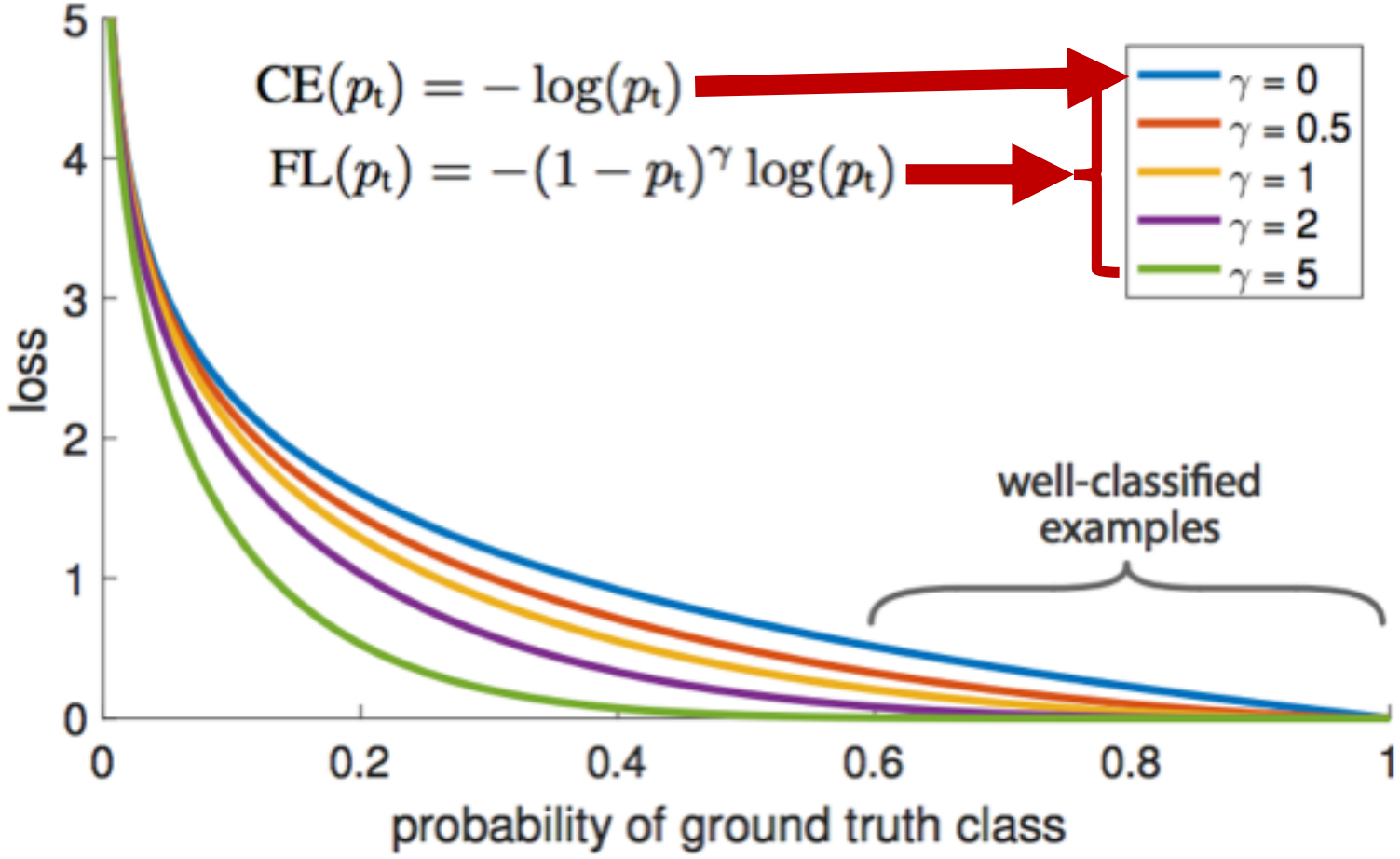
- P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.
- A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In CVPR, 2016.
- P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In NIPS, 2015.

RetinaNet Model Description



- Backbone with FPN + class-specific RPN (final detections)
- 6 anchors per location (2 scales × 3 aspect ratios)
- 100 – 200k anchor boxes to classify per image → “dense” detection

Focal Loss: “Soft” Hard-Example Mining

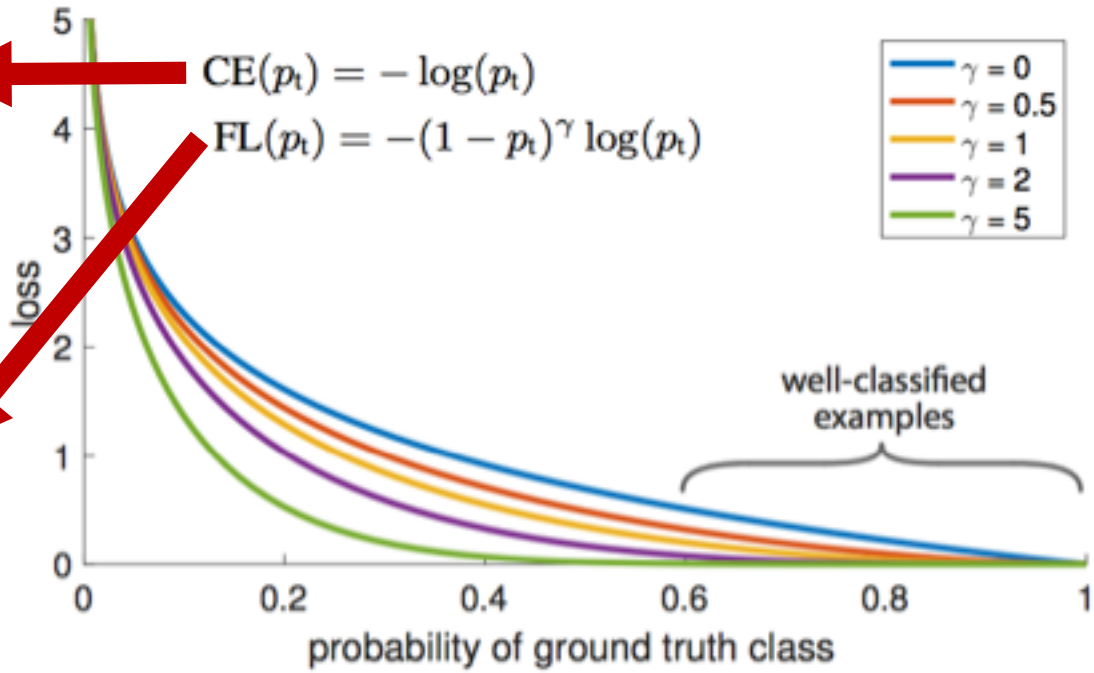


Impact of Focal Loss (FL)

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)

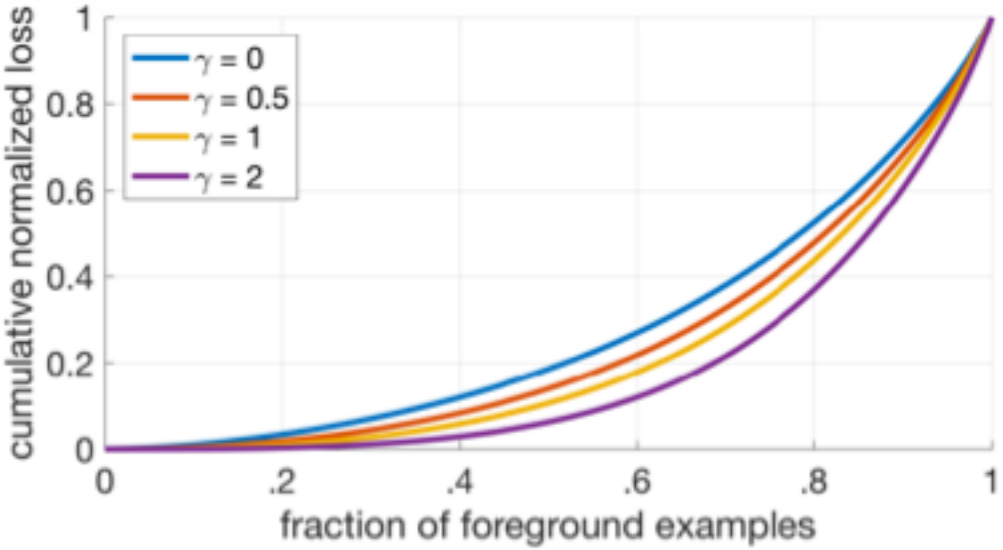
(ResNet-50-FPN 600px input image)



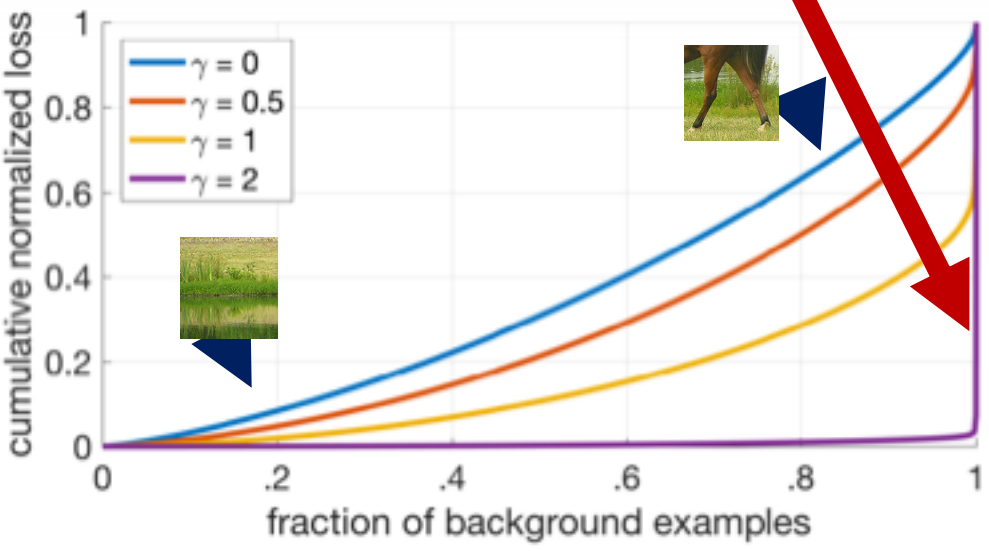
Loss Distribution under Focal Loss

Hard mining behavior

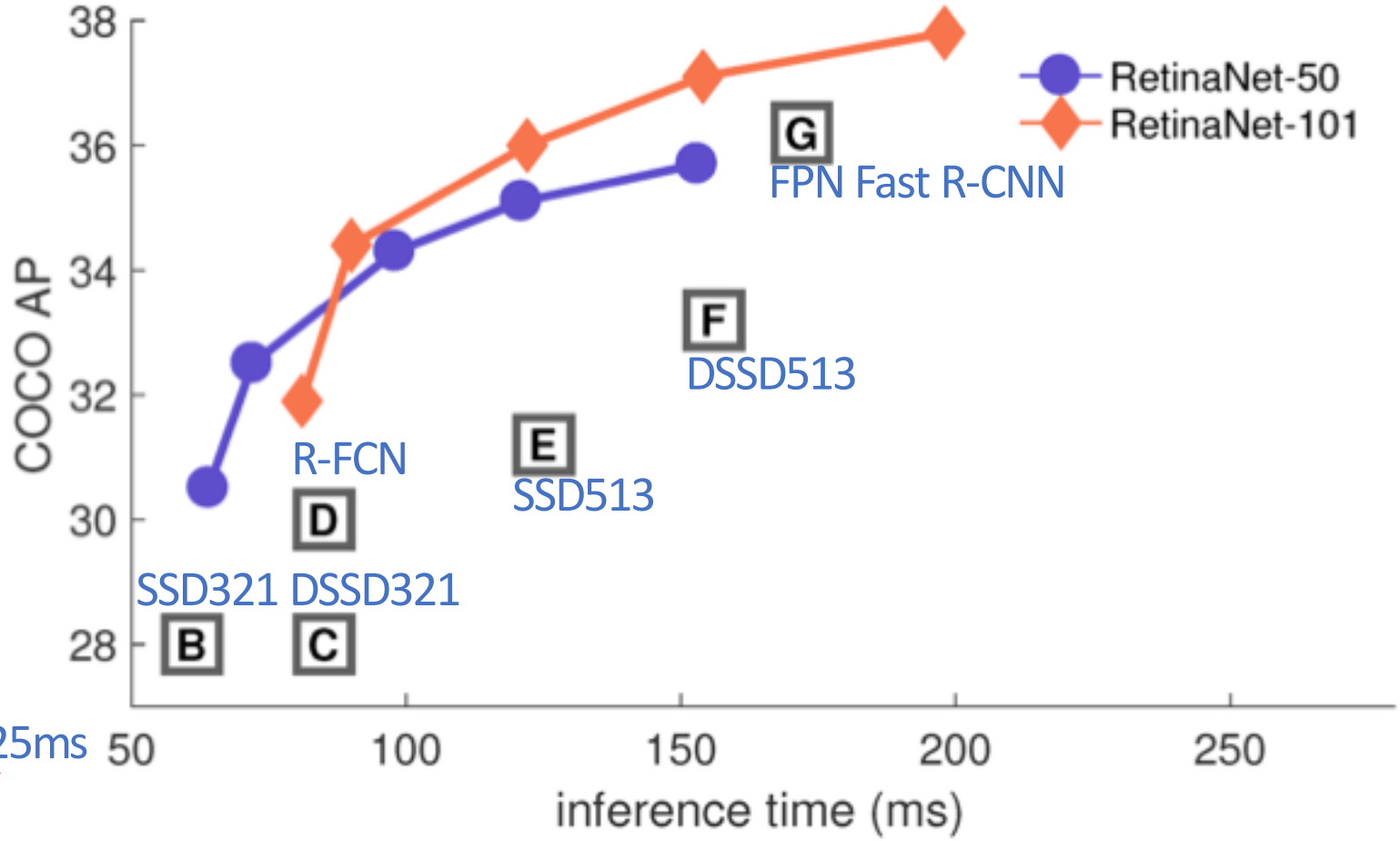
Foreground Boxes



Background Boxes

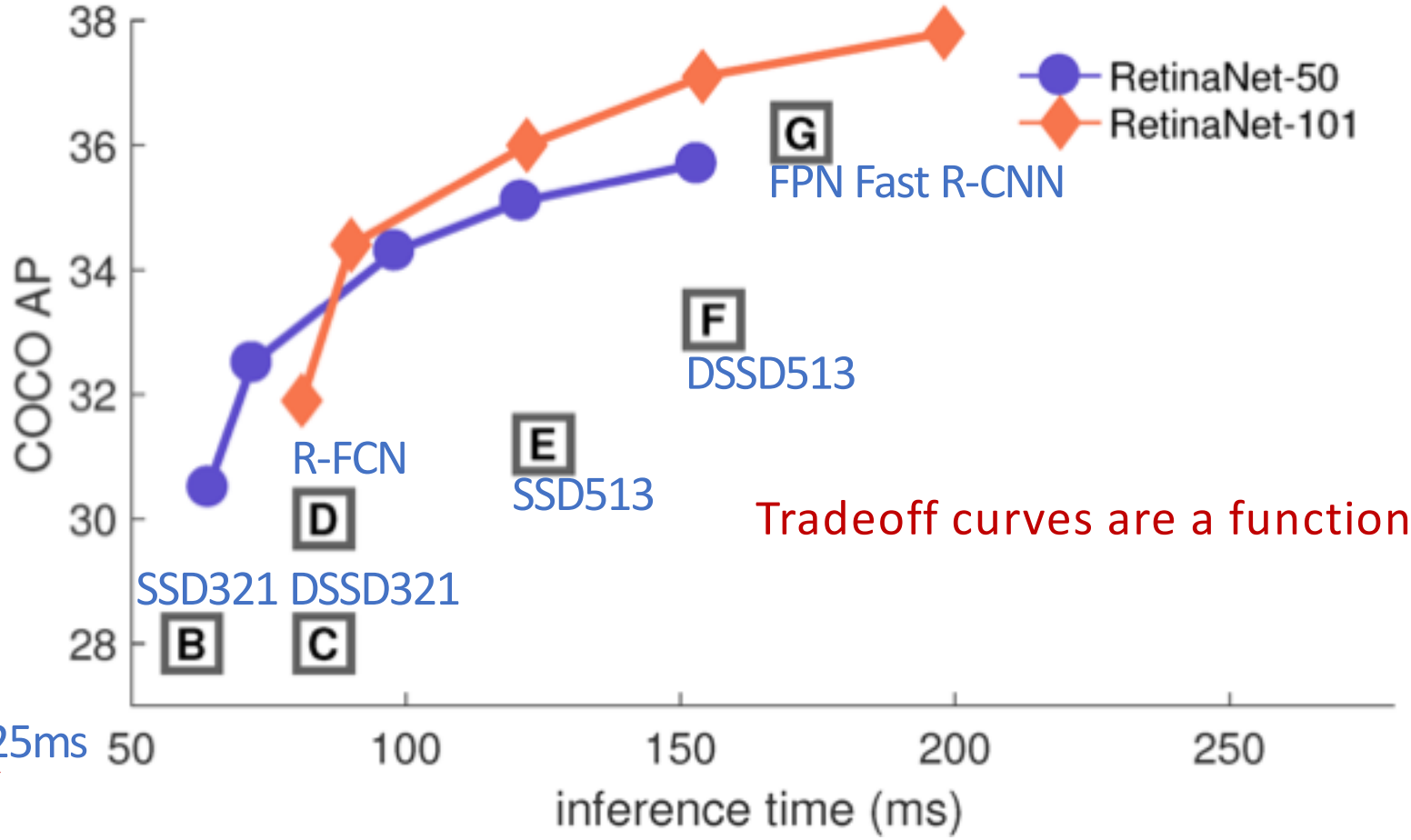


Speed/Accuracy Tradeoff



YOLOv2
AP 22 @ 25ms

Speed/Accuracy Tradeoff



Tradeoff curves are a function of FLOPs

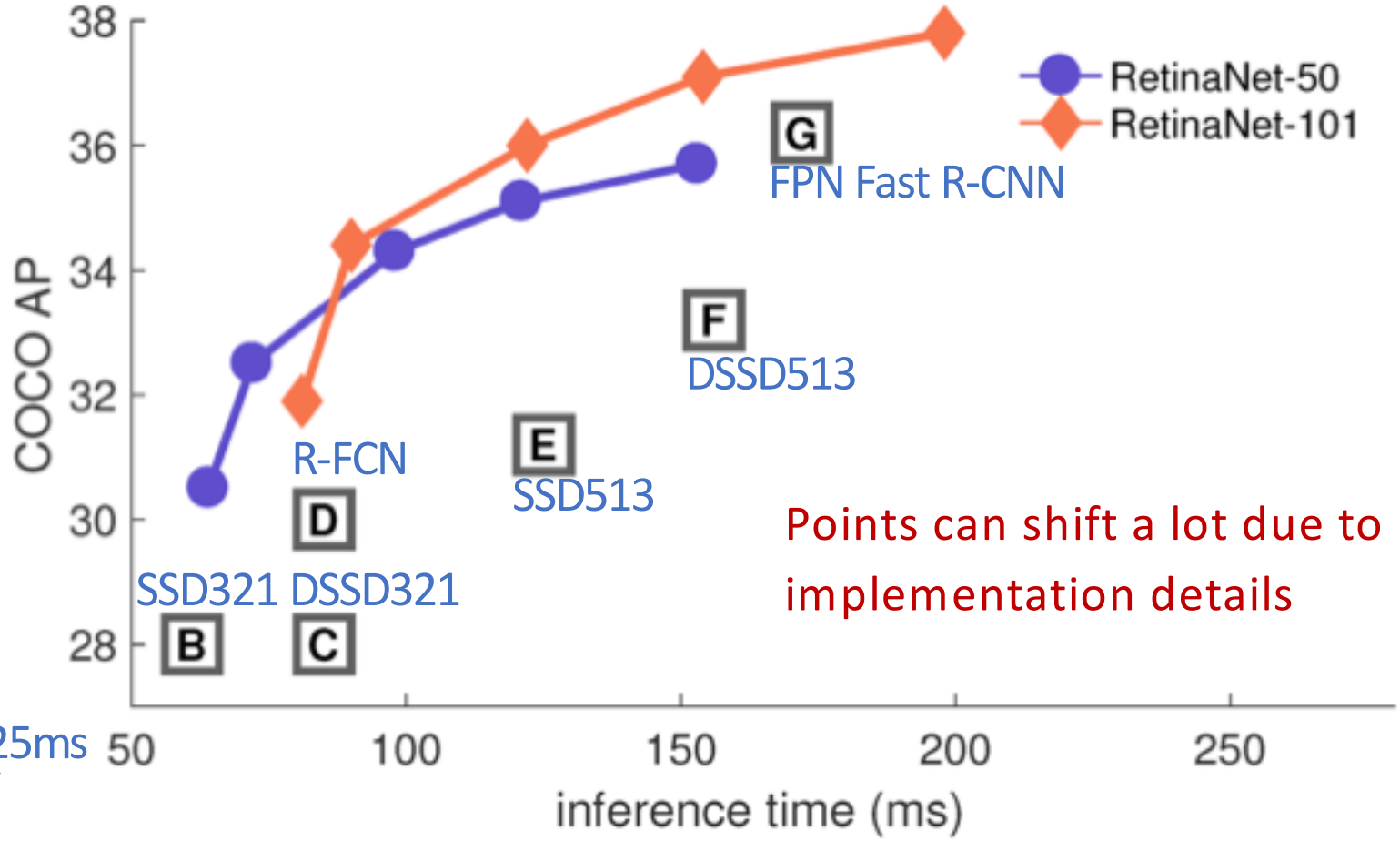
YOLOv2
AP 22 @ 25ms

Warning about Speed/Accuracy Tradeoffs

Comparisons across publications are *completely uncontrolled*

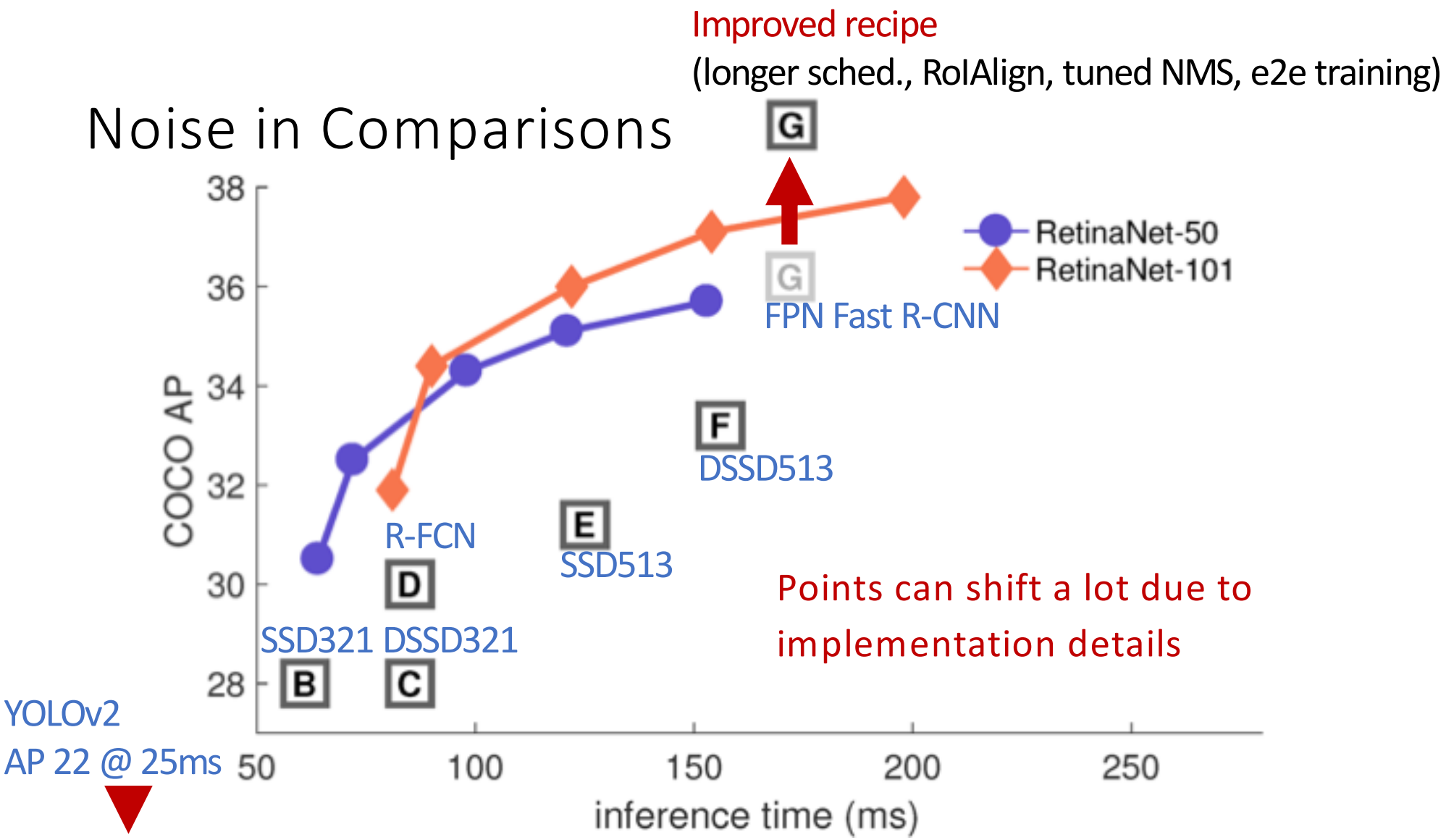
- Results should be taken with a grain of salt
- Accuracy varies with hyper-parameters ('recipe')
- Speed varies with low-level optimization (perf tuning)
- Speed varies with 'tricks' (e.g., batching during inference)

Noise in Comparisons



YOLOv2
AP 22 @ 25ms

Noise in Comparisons



Fast Detectors

Key design factors (*it's all about the FLOPs*)

- Low resolution input
- Lightweight backbone network
- Top-*k* proposals with small *k* = 10-50 (if applicable)

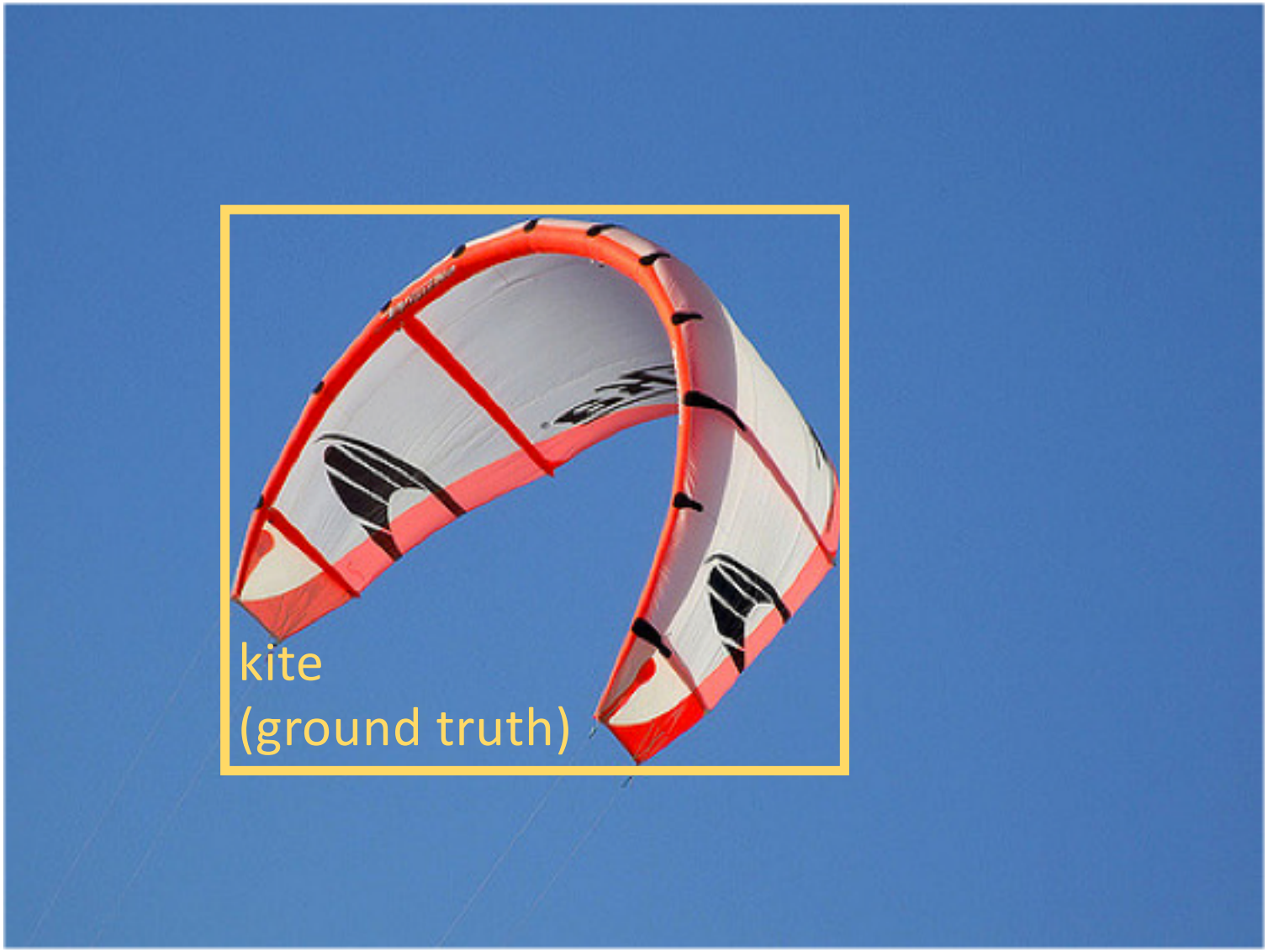
"Anti-factors"

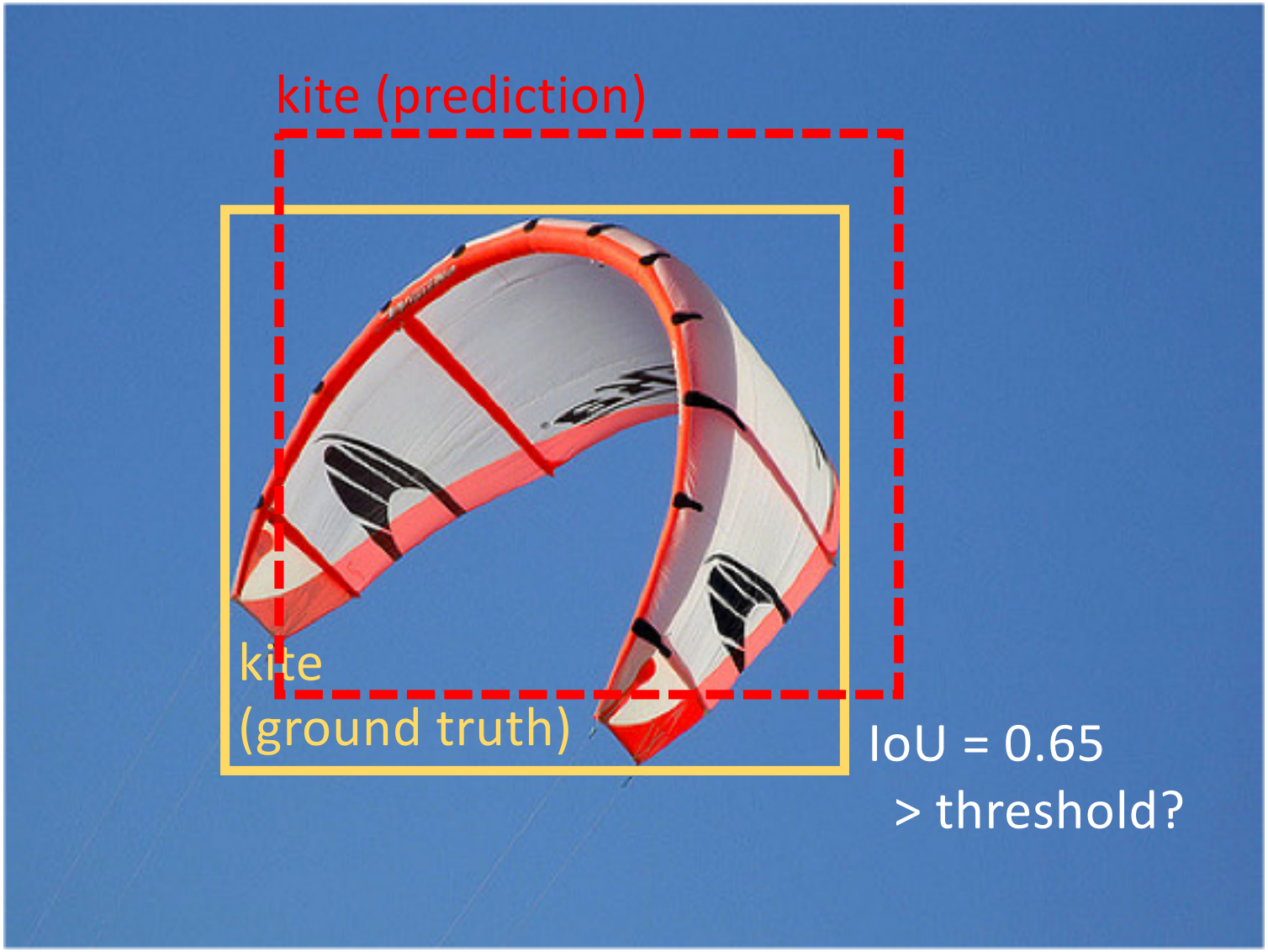
- One-stage instead of two-stage

In Conclusion, Four Takeaway Points

1. Detection is not solved – *datasets are just a model of the world*
2. Features matter – *the backbone net is the engine of recognition*
3. Detector design matters – *progress from domain knowledge*
4. Speed/accuracy tradeoff is *not* a point – *it's a curve/surface*

End of Slides


kite
(ground truth)

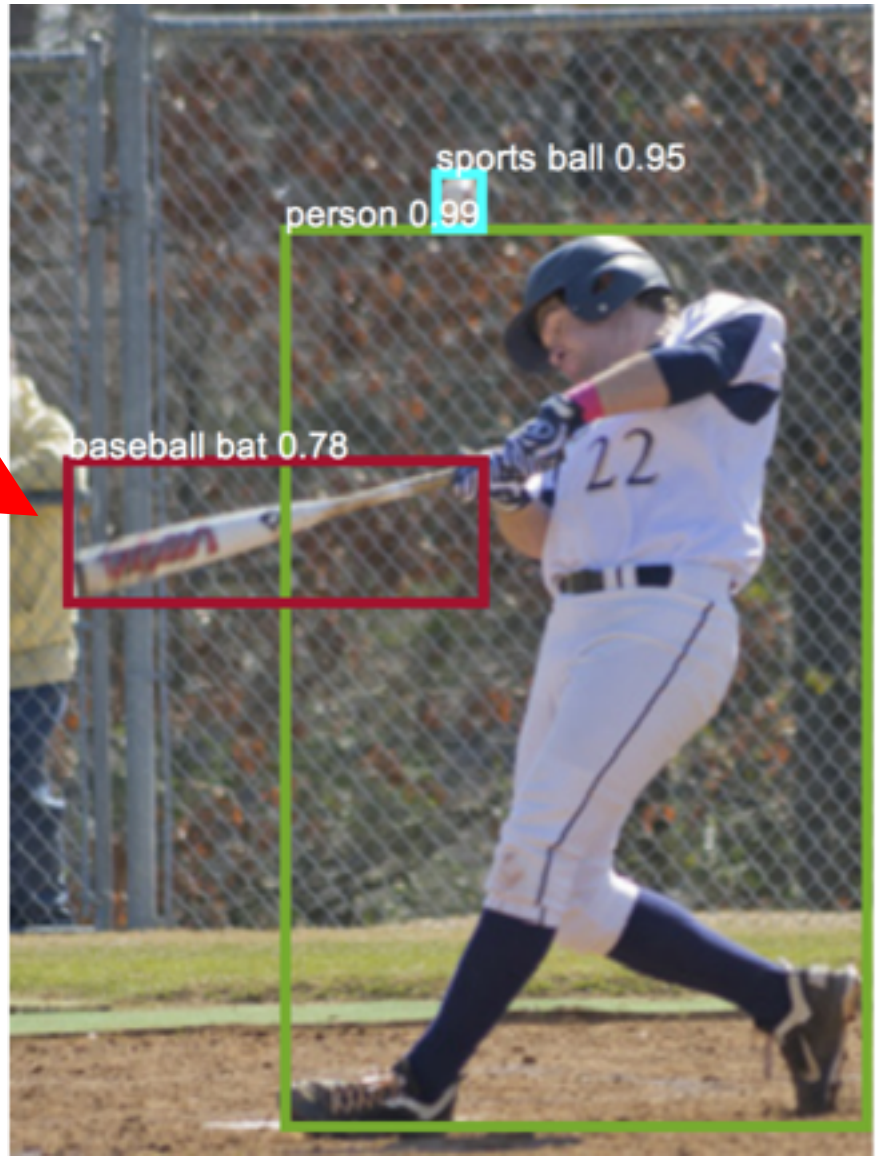

kite (prediction)
kite
(ground truth)
IoU = 0.65
> threshold?

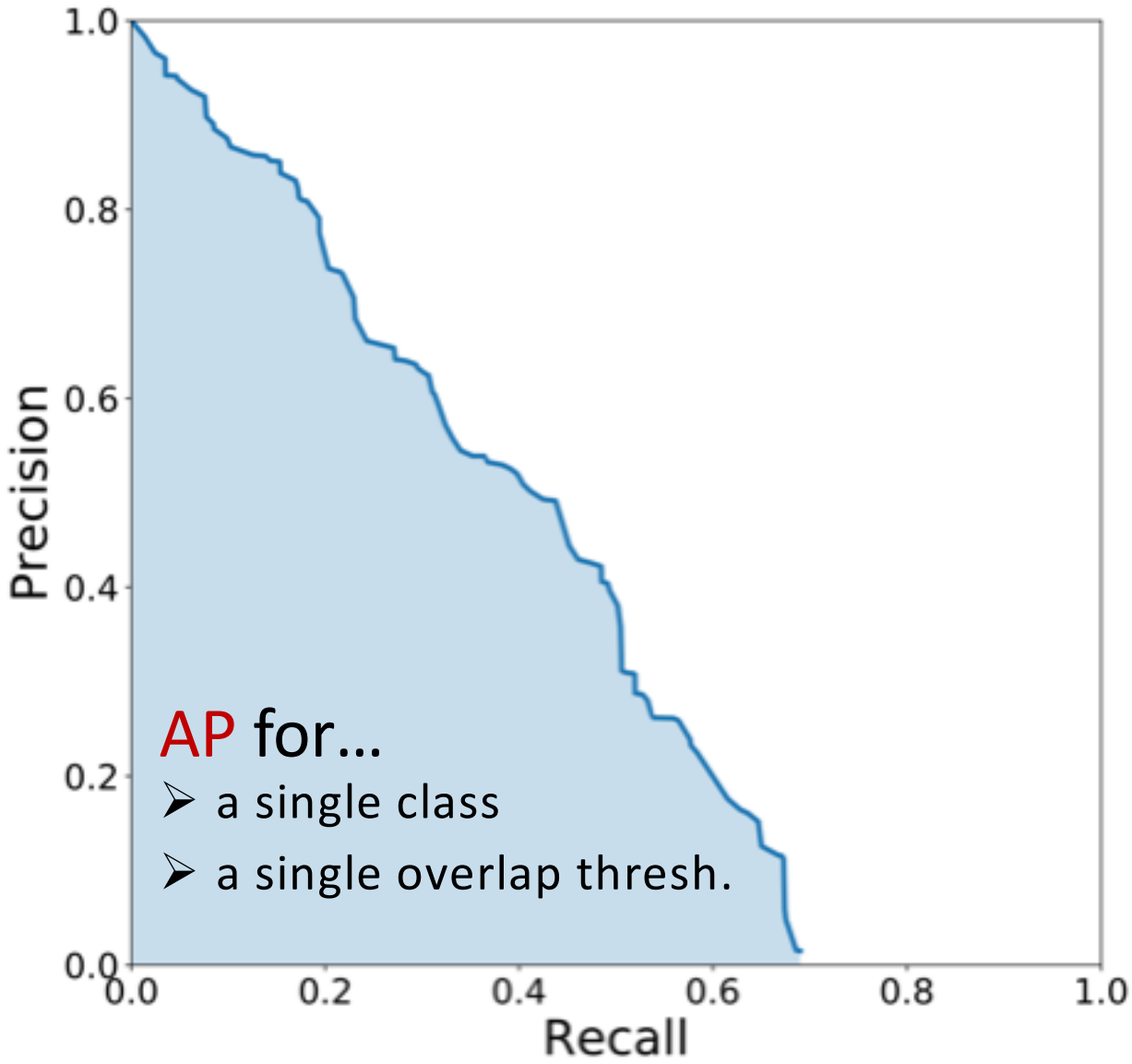
False Positive

IoU too low

(or duplicate)

False Negative
(missing detection)




1.0
0.8
0.6
0.4
0.2
0.0
Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
AP for…
➢ a single class
➢ a single overlap thresh.

COCO Average Precision (%)

AP averaged over…

- Object categories
- True positive overlap (IoU) requirement (from 0.5 to 0.95)

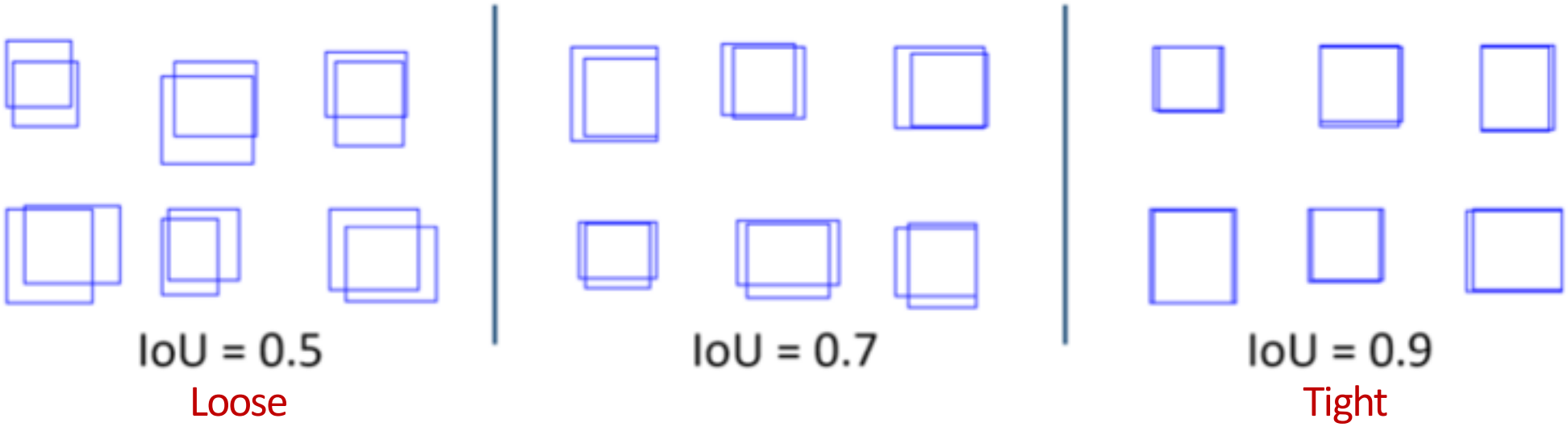
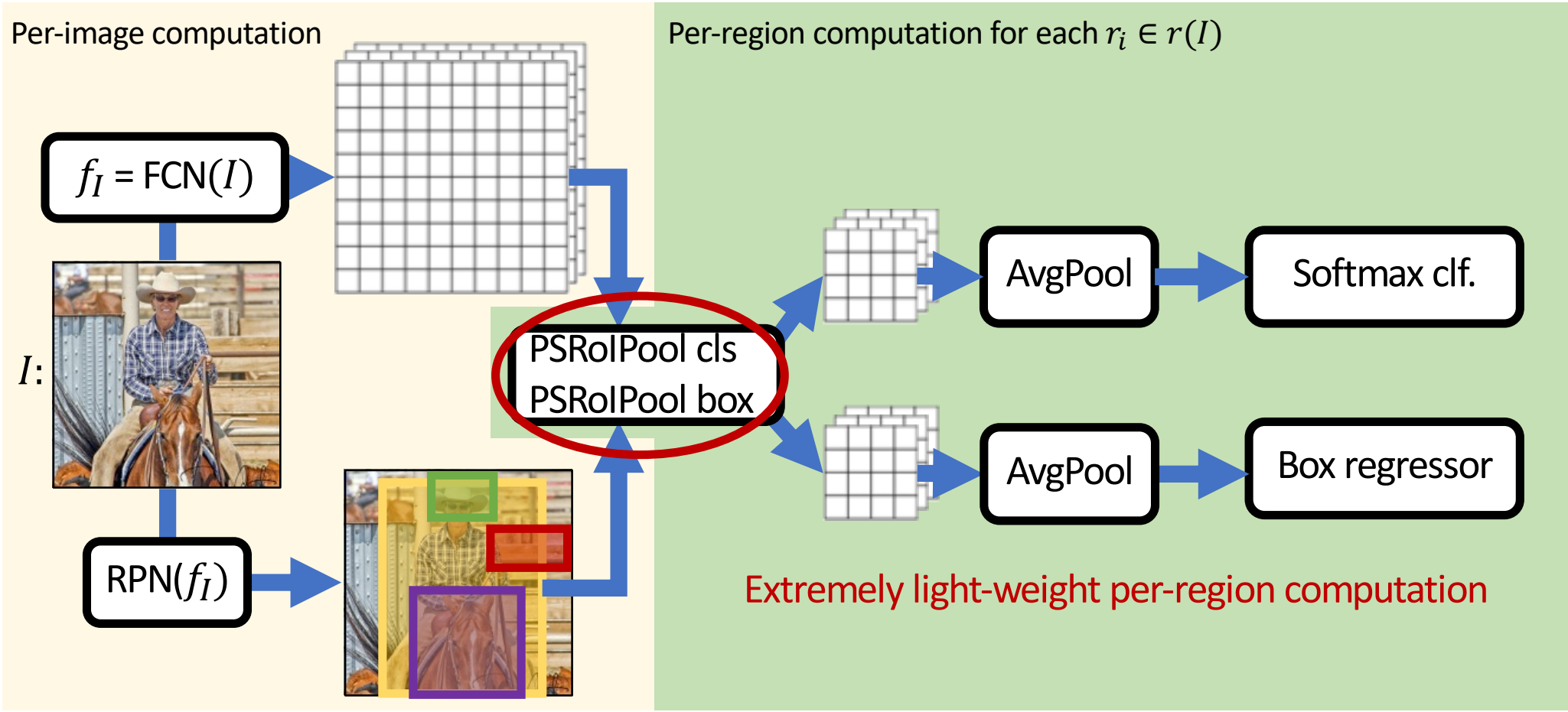


Figure credits: Dollár and Zitnick

Dai, Li, He, Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. NIPS 2016.

R-FCN



Slow R-CNN in the Generalized Framework