2D Object Detection

Renjie Liao

largely reuse materials in slides and papers of Ross Girshick, Kaiming He, et. al.

2D Object Detection

Localization
Where?

Recognition
What?

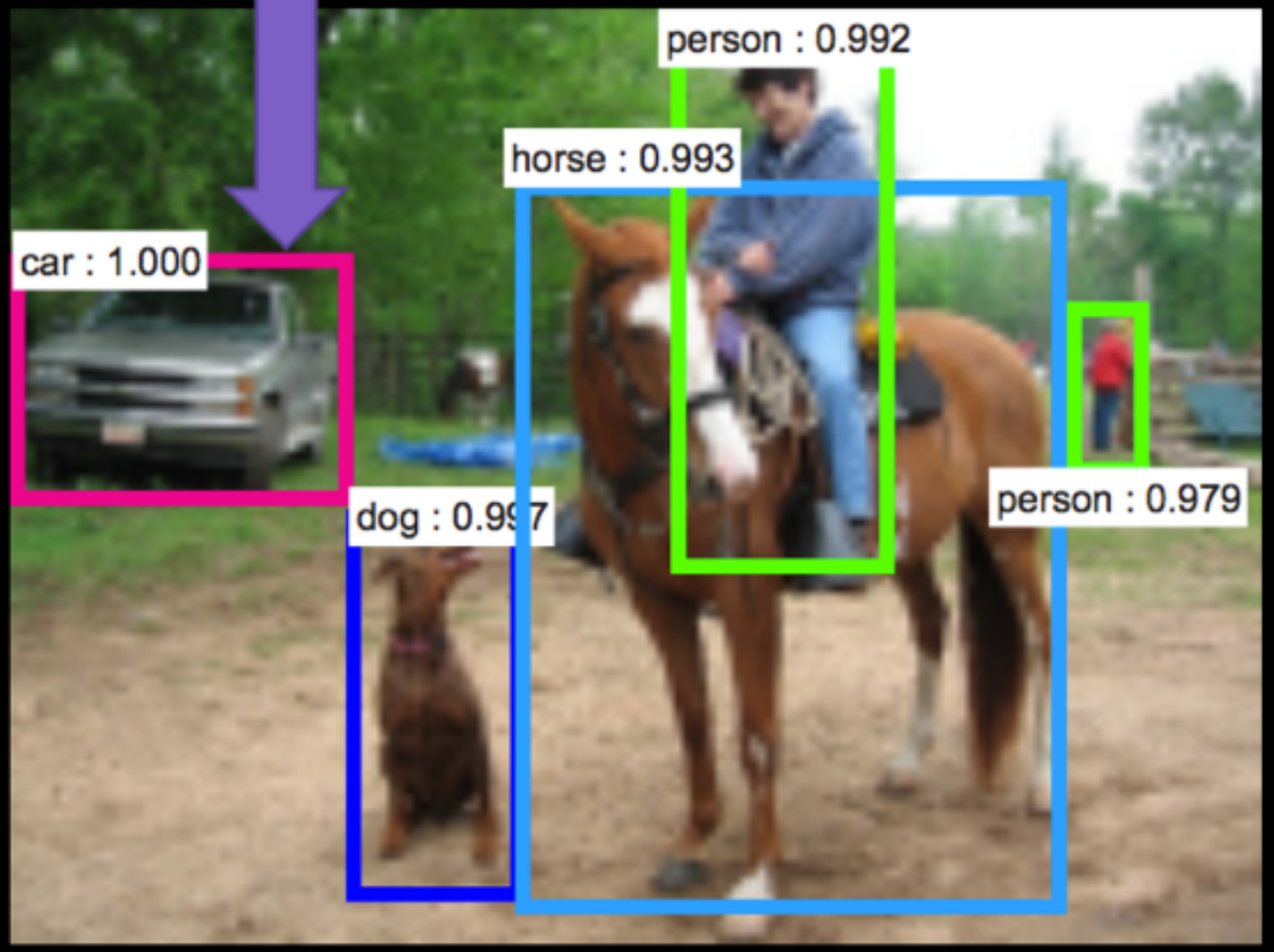
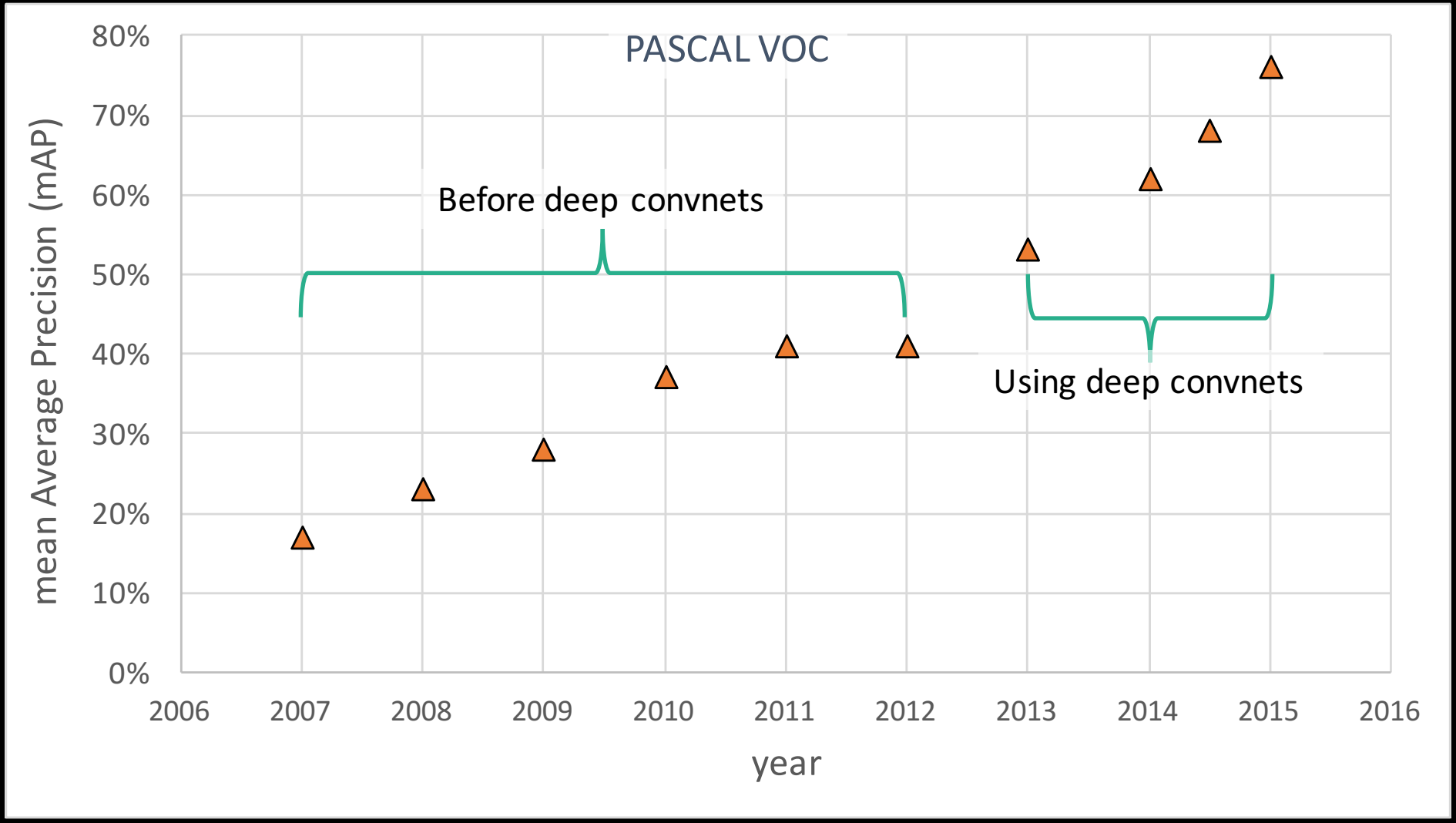
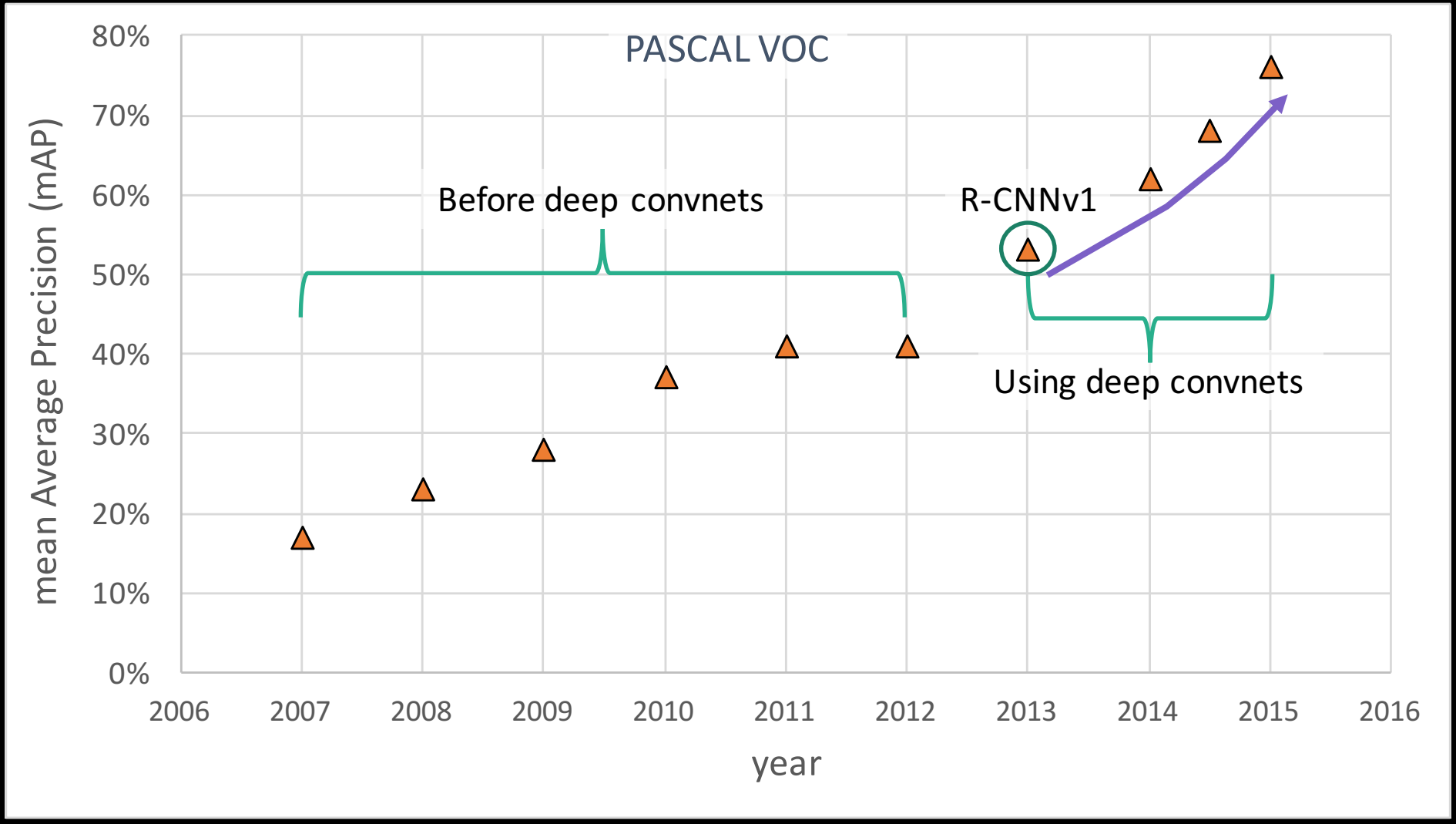


Figure adapted from Kaiming He

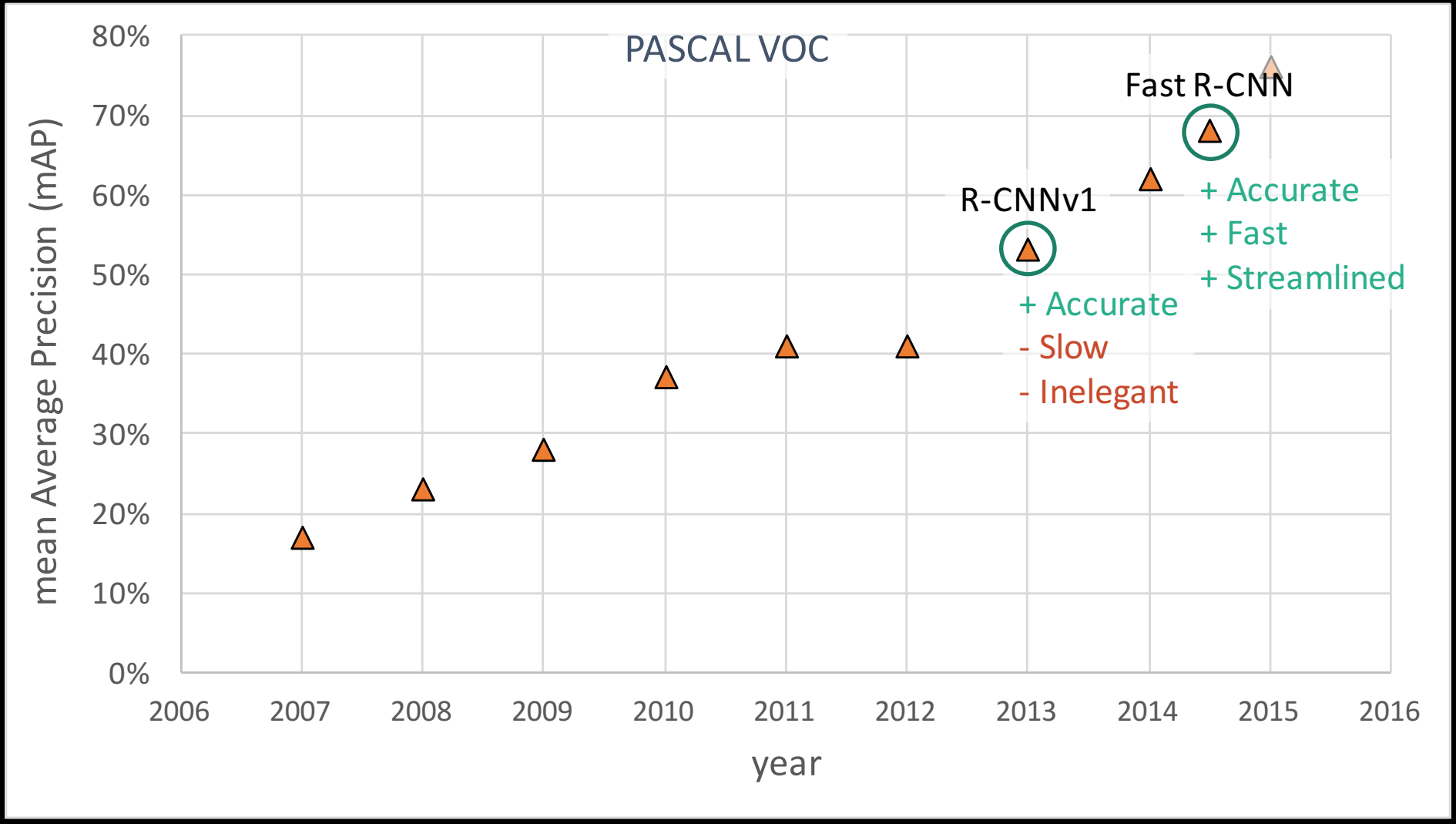
Object detection renaissance (2013-present)



Object detection renaissance (2013-present)

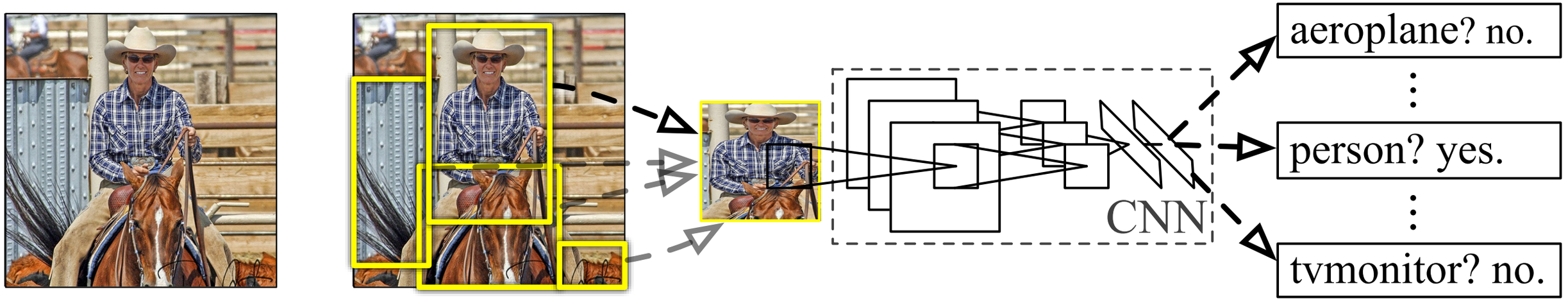


Object detection renaissance (2013-present)



R-CNN

R-CNN: Regions with CNN features



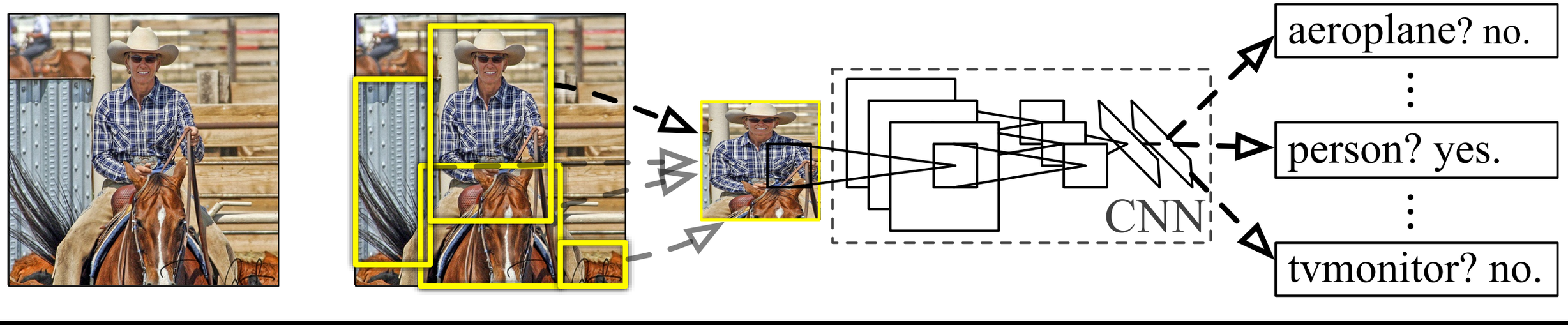
Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions (linear SVM)

R-CNN at test time: Step 1



Input image

Extract region proposals (~2k / image)

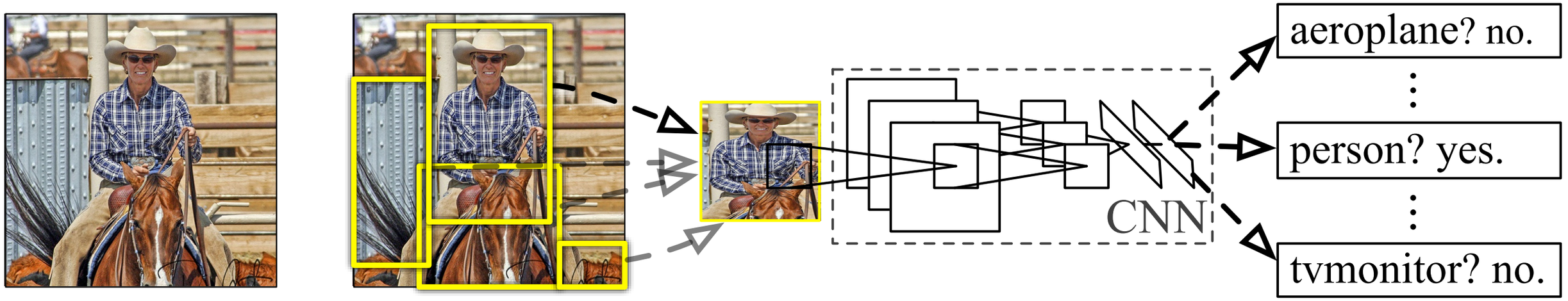
Proposal-method agnostic, many choices

- Selective Search [van de Sande, Uijlings et al.] (Used in this work)
- Objectness [Alexe et al.]
- Category independent object proposals [Endres & Hoiem]
- CPMC [Carreira & Sminchisescu]

Active area, at this CVPR

- BING [Ming et al.] – *fast*
- MCG [Arbelaez et al.] – *high-quality segmentation*

R-CNN at test time: Step 2

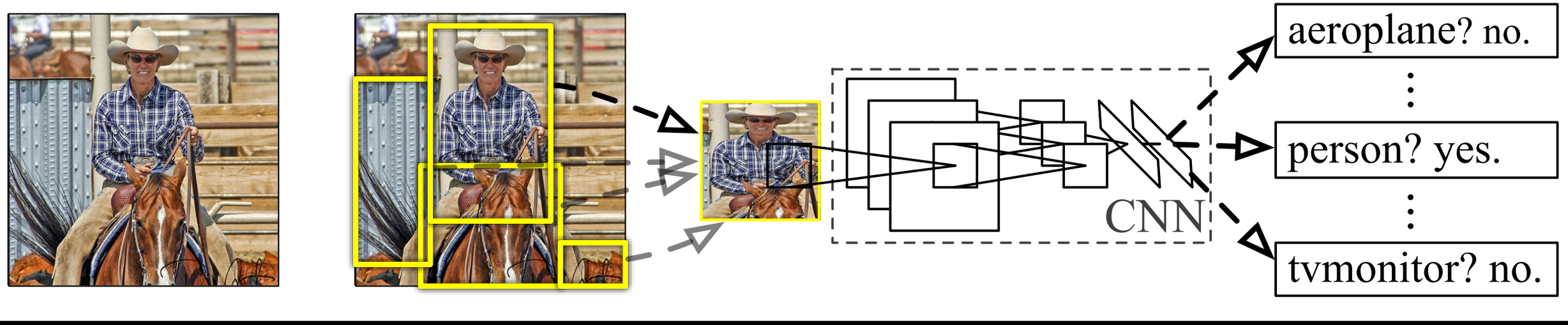


Input image

Extract region proposals (~2k / image)

Compute CNN features

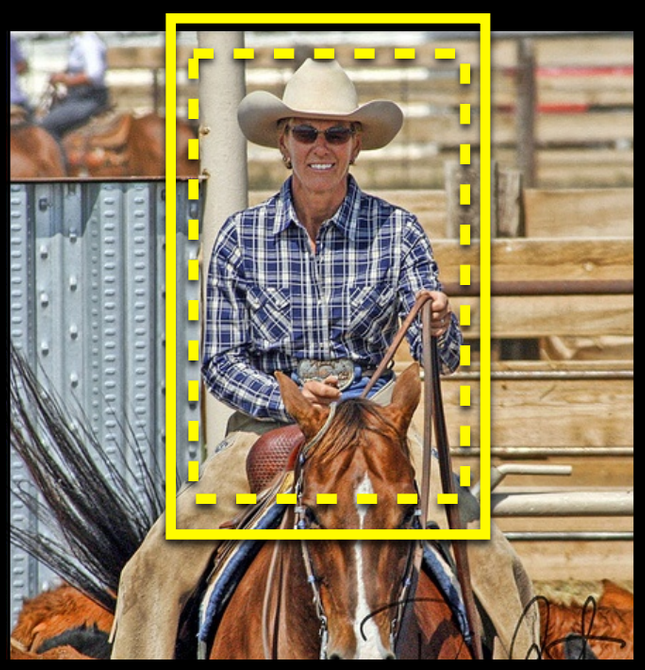
R-CNN at test time: Step 2



Input image

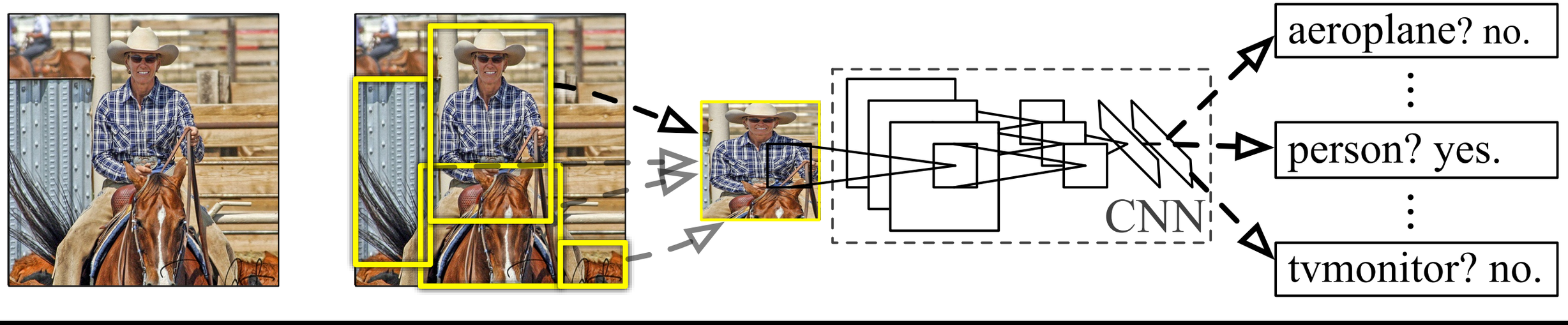
Extract region proposals (~2k / image)

Compute CNN features



Dilate proposal

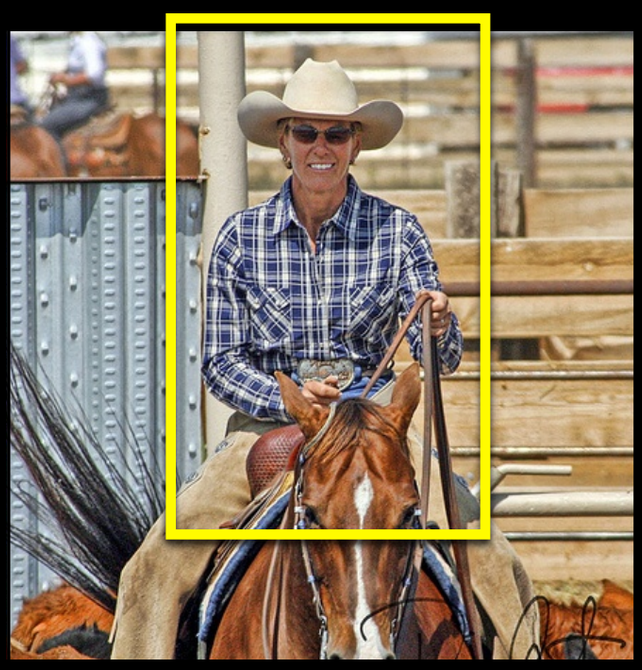
R-CNN at test time: Step 2



Input image

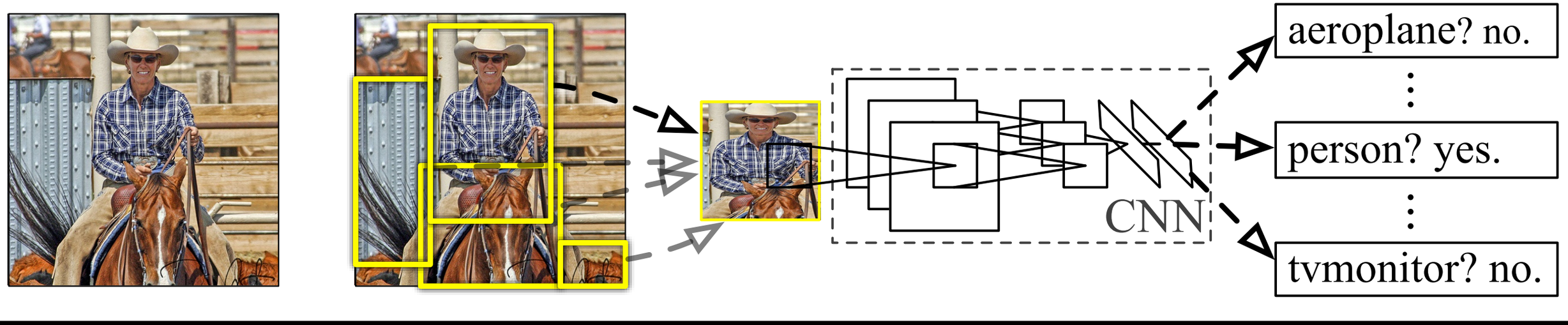
Extract region proposals (~2k / image)

Compute CNN features



a. Crop

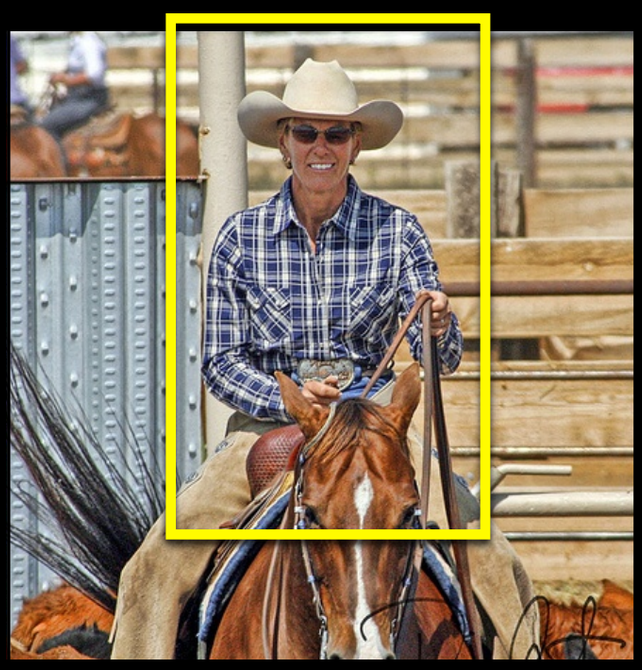
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

Compute CNN features

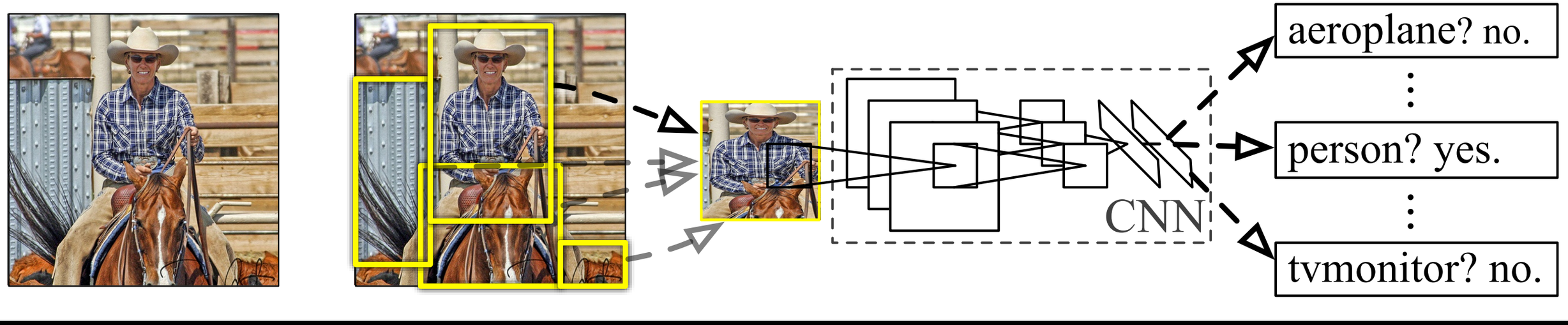


a. Crop

227 x 227

b. Scale (anisotropic)

R-CNN at test time: Step 2



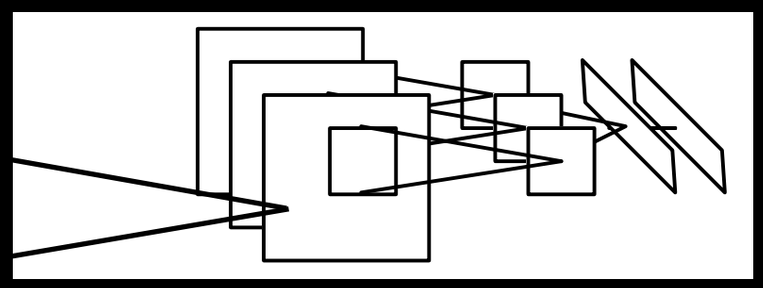
Input image

Extract region proposals (~2k / image)

Compute CNN features

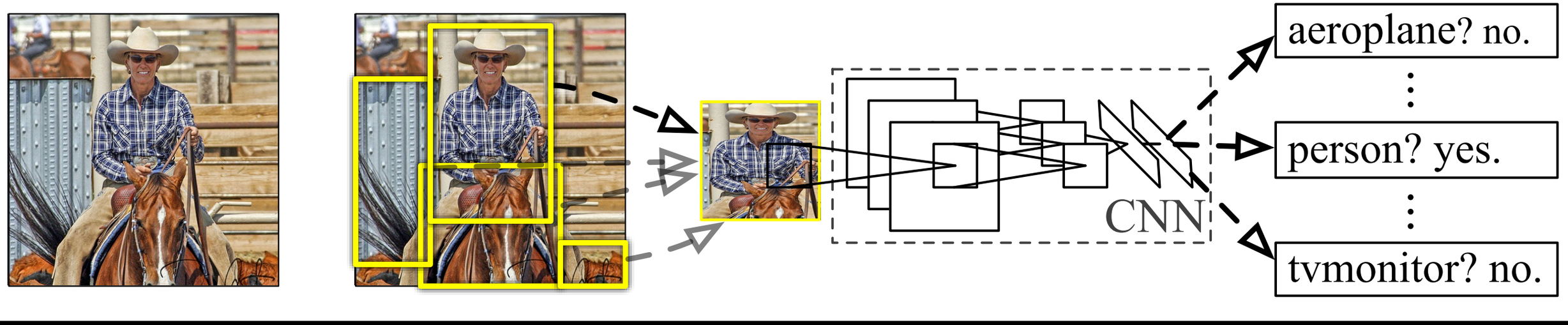
. Crop

b. Scale (anisotropic)



c. Forward propagate
Output: “fc_7” features

R-CNN at test time: Step 3

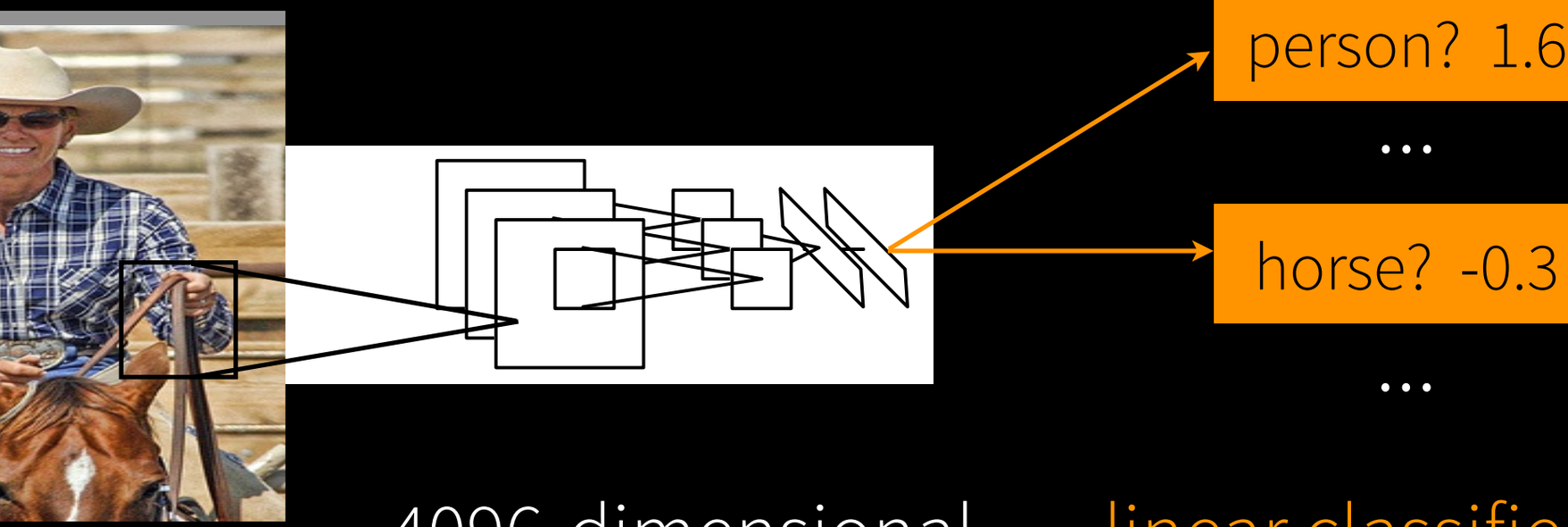


Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions



proposal

4096-dimensional fc_7 feature vector

linear classifiers (SVM or softmax)

Step 4: Object proposal refinement

Original
proposal

Linear regression

on CNN features

Predicted
object bounding box

Bounding-box regression

Slow R-CNN

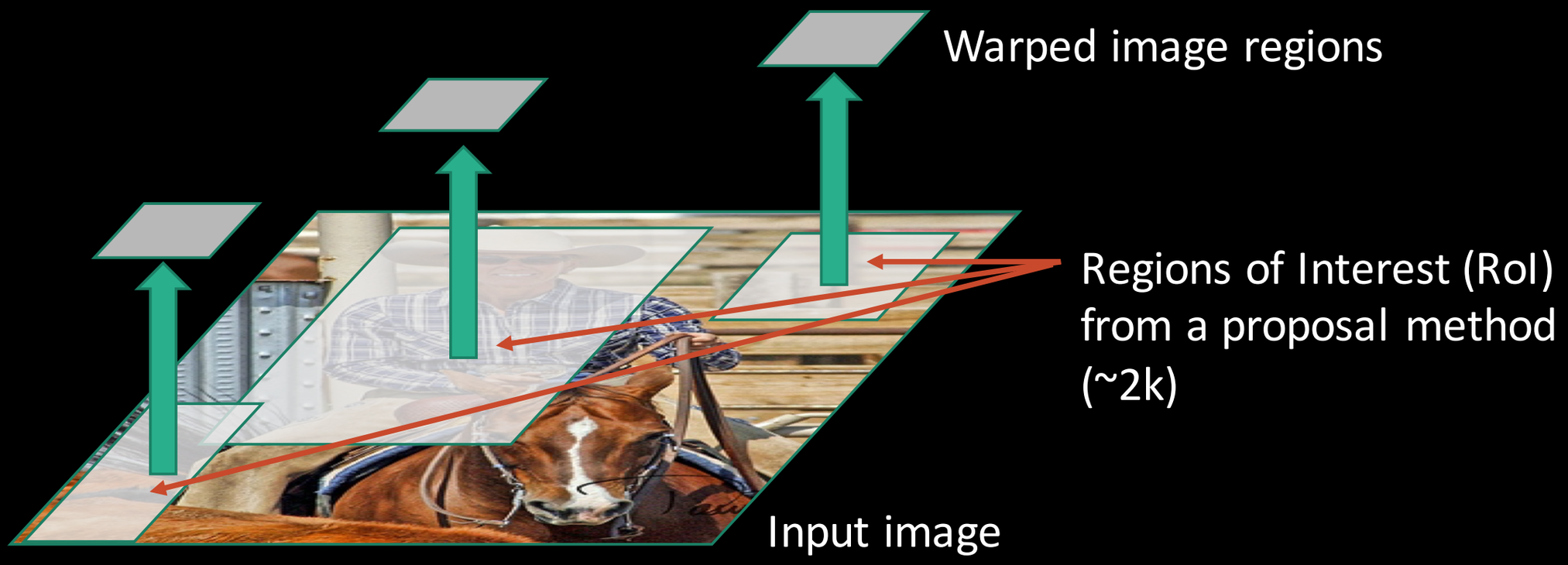
Input image

Girshick et al. CVPR14.

Slow R-CNN

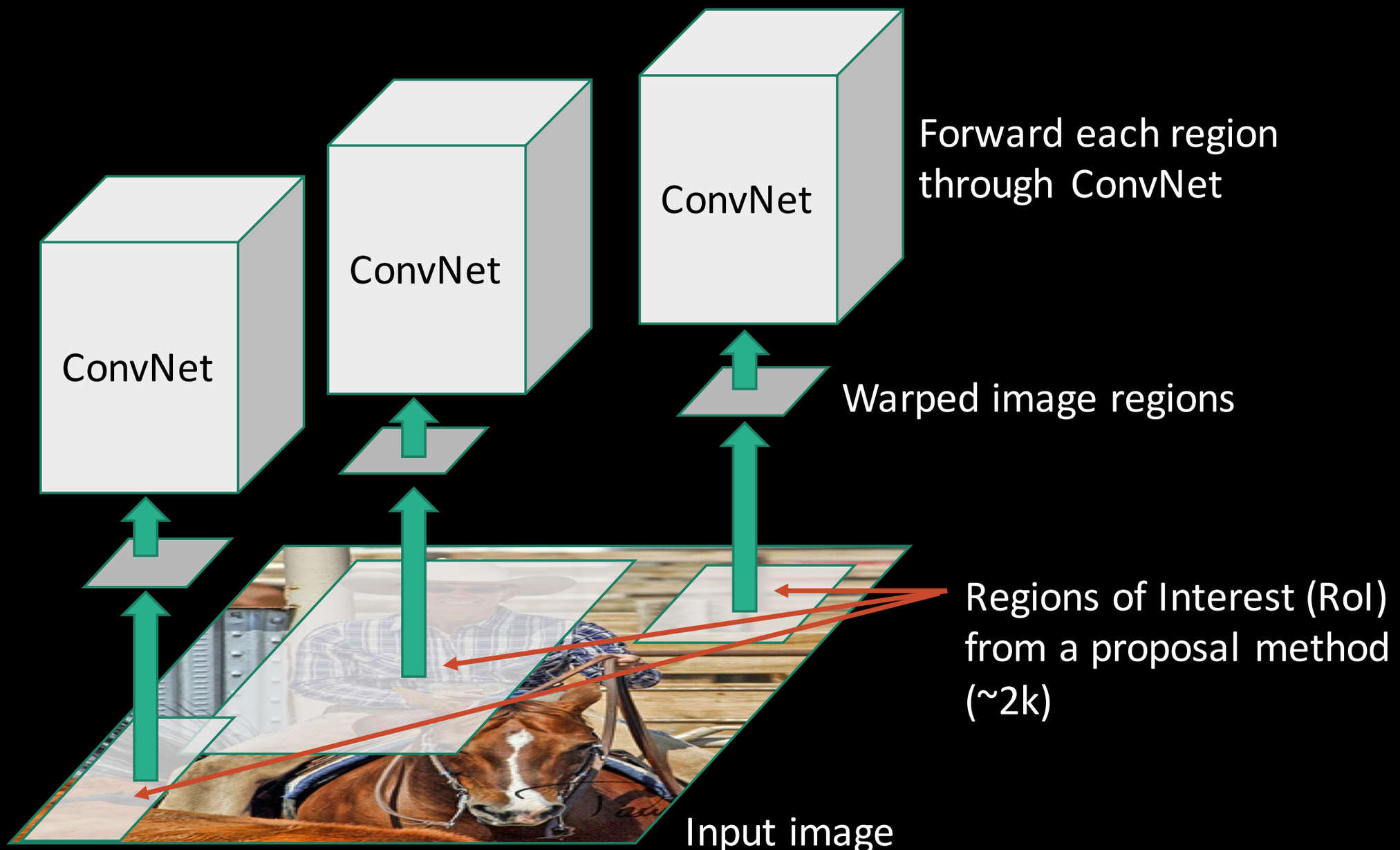
Girshick et al. CVPR14.

Slow R-CNN



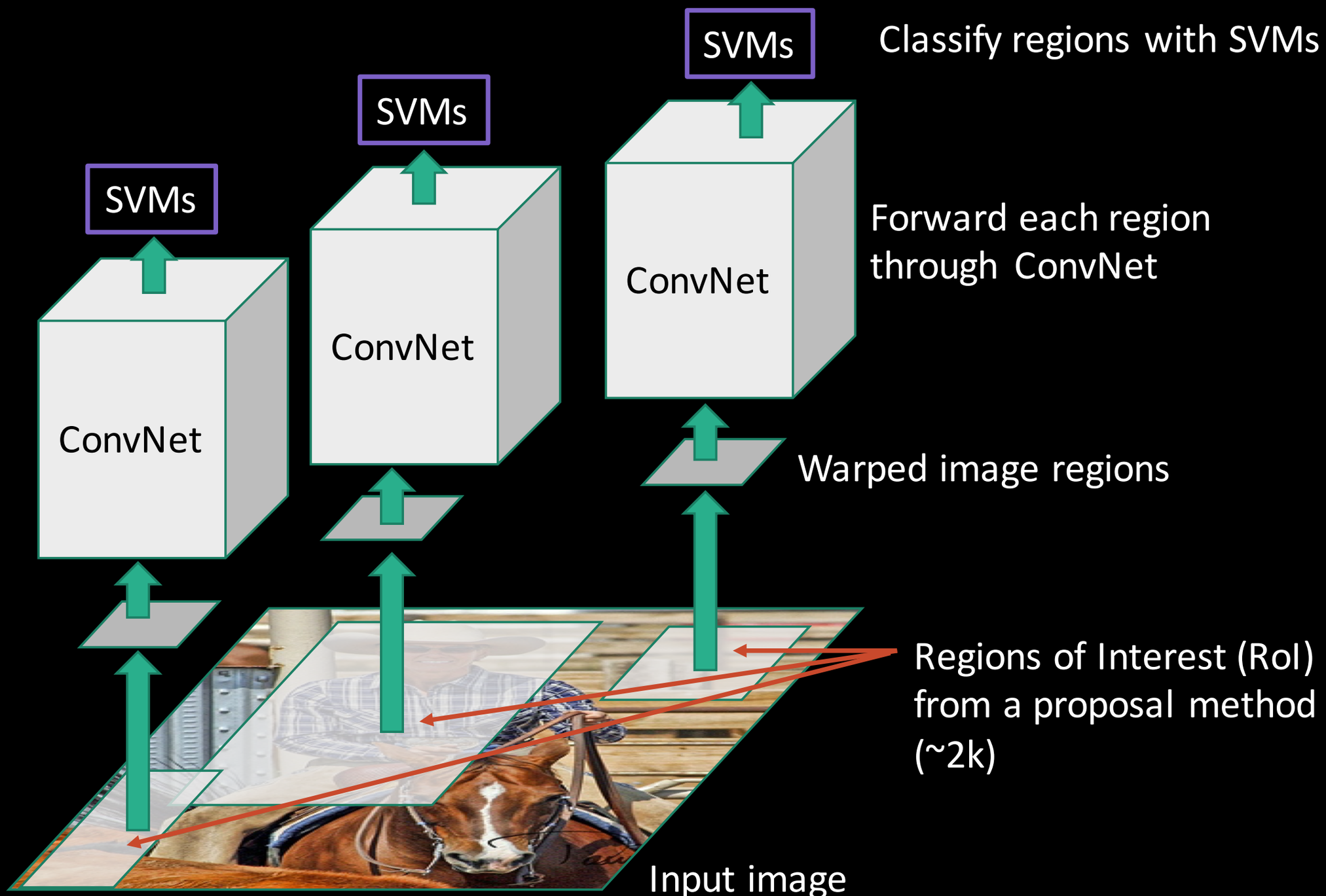
Girshick et al. CVPR14.

Slow R-CNN



Girshick et al. CVPR14.

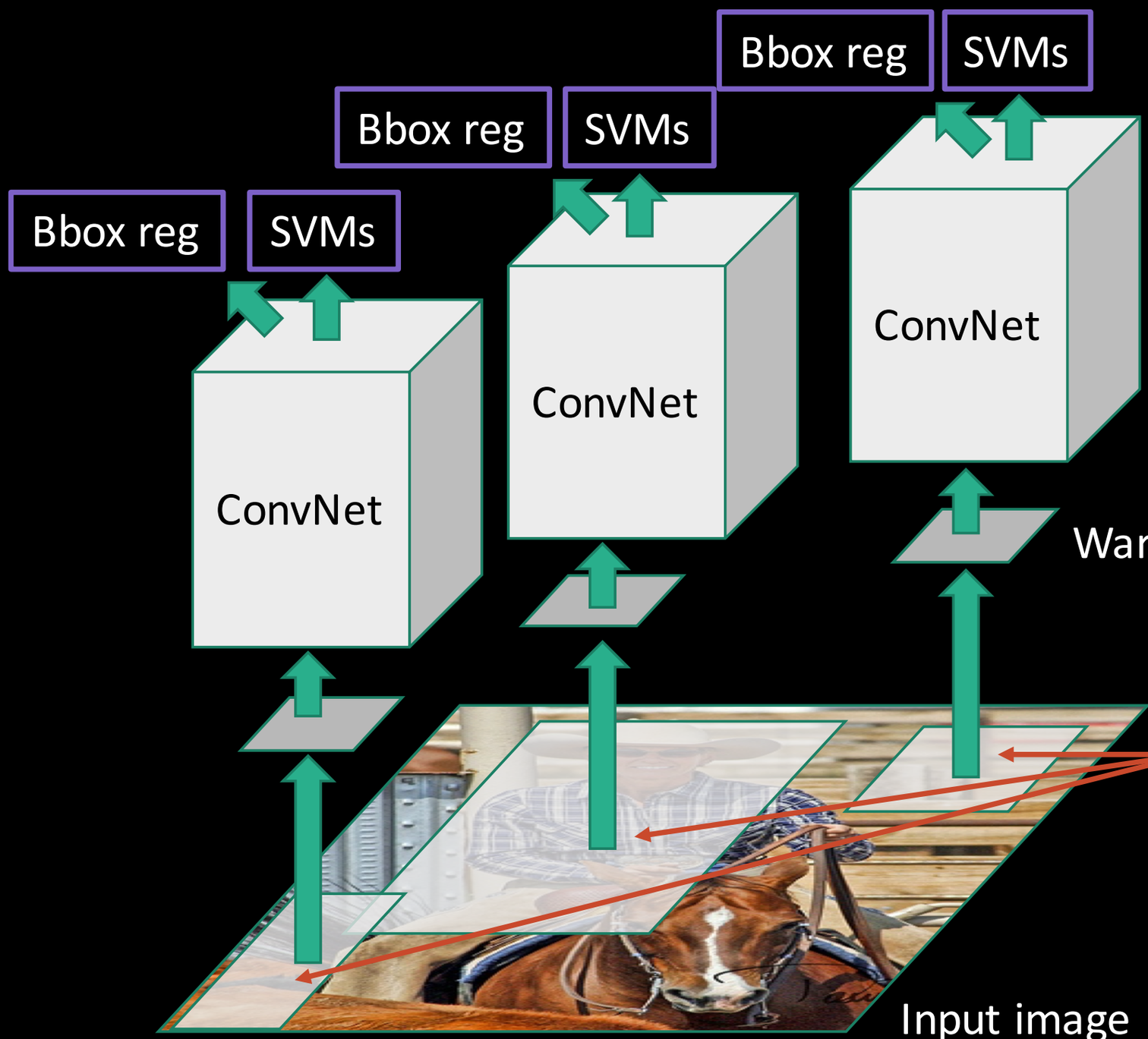
Slow R-CNN



Girshick et al. CVPR14.

Post hoc component

Slow R-CNN



Apply bounding-box regressors

Classify regions with SVMs

Forward each region through ConvNet

Regions of Interest (RoI) from a proposal method (~2k)

Girshick et al. CVPR14.

Post hoc component

Training R-CNN

Bounding-box labeled detection data is scarce

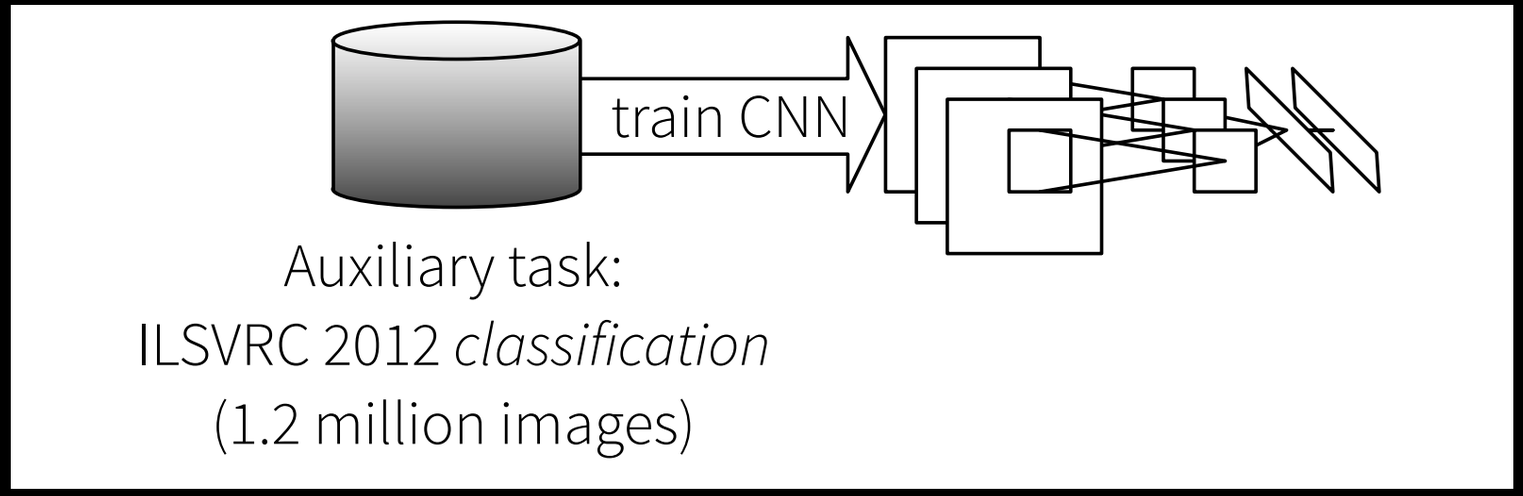
Key insight:

Use *supervised* pre-training on a data-rich auxiliary task and *transfer* to detection

R-CNN training: Step 1

Supervised pre-training

Train a SuperVision CNN* for the 1000-way ILSVRC image classification task

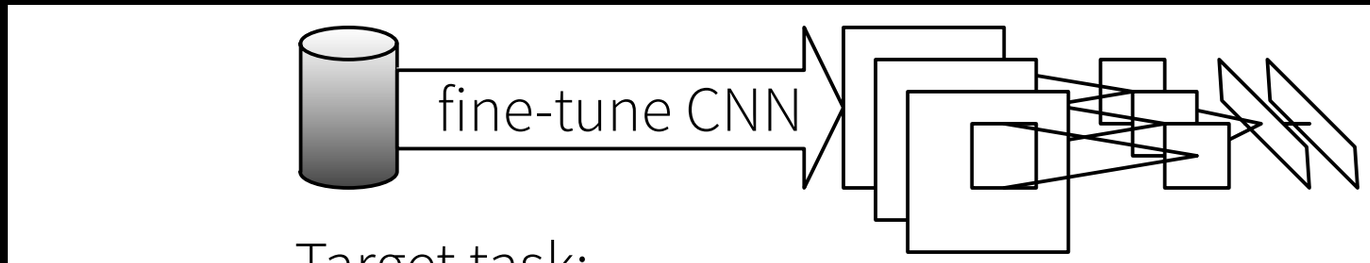


*Network from Krizhevsky, Sutskever & Hinton. NIPS 2012
Also called “AlexNet”

R-CNN training: Step 2

Fine-tune the CNN for detection

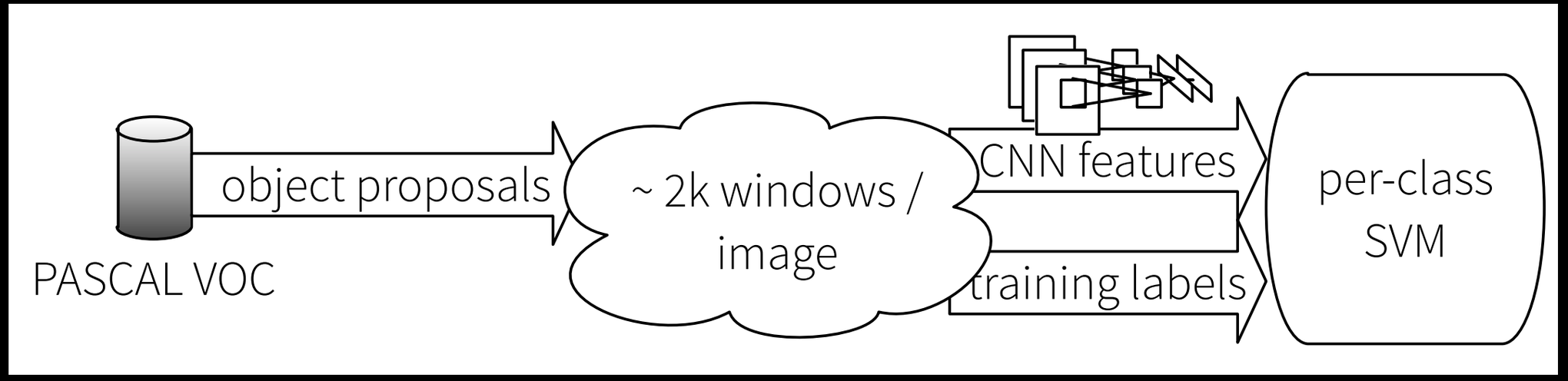
Transfer the representation learned for ILSVRC classification to PASCAL (or ImageNet detection)



Target task:
PASCAL VOC detection
(~25k object labels)

R-CNN training: Step 3

Train detection SVMs
(With the softmax classifier from fine-tuning
mAP decreases from 54% to 51%)



Fast R-CNN

What’s wrong with slow R-CNN?

What's wrong with slow R-CNN?

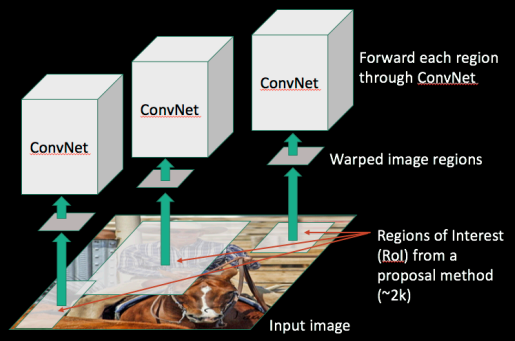
- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressors (squared loss)

What's wrong with slow R-CNN?

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressors (squared loss)
- Training is slow (84h), takes a lot of disk space

What's wrong with slow R-CNN?

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]



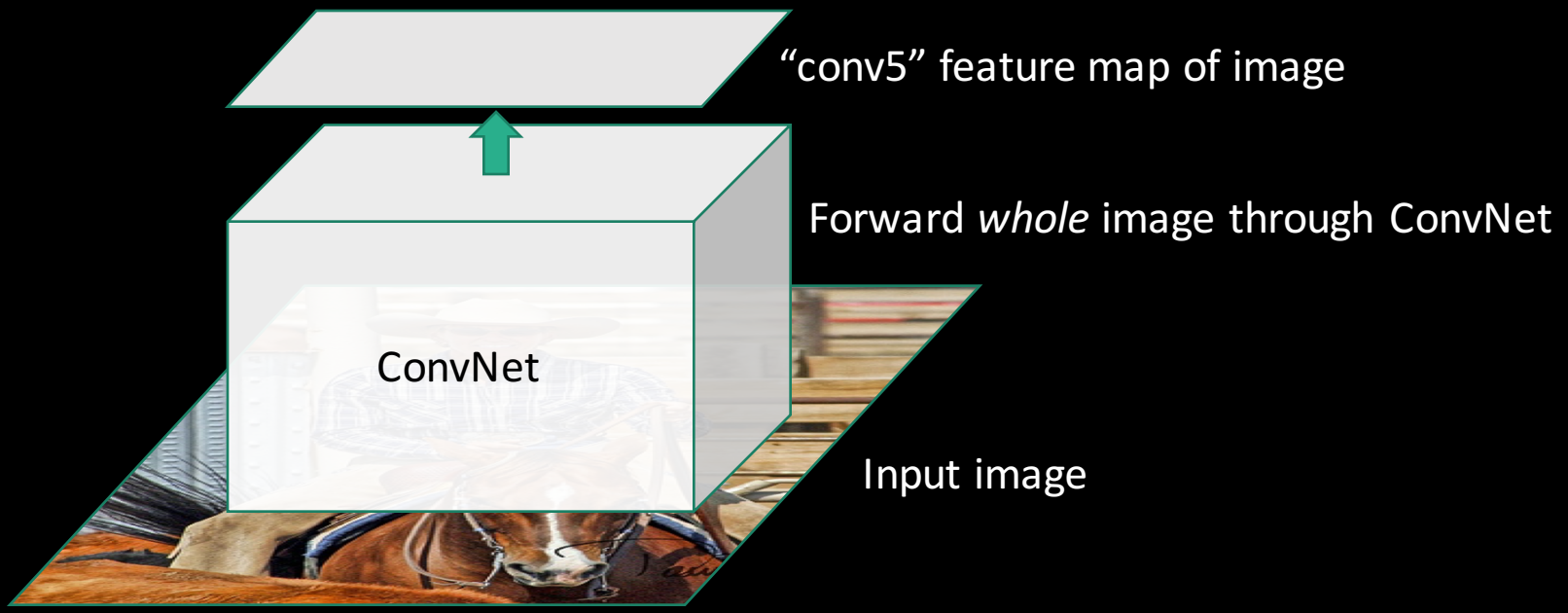
~2000 ConvNet forward passes per image

SPP-net

Input image

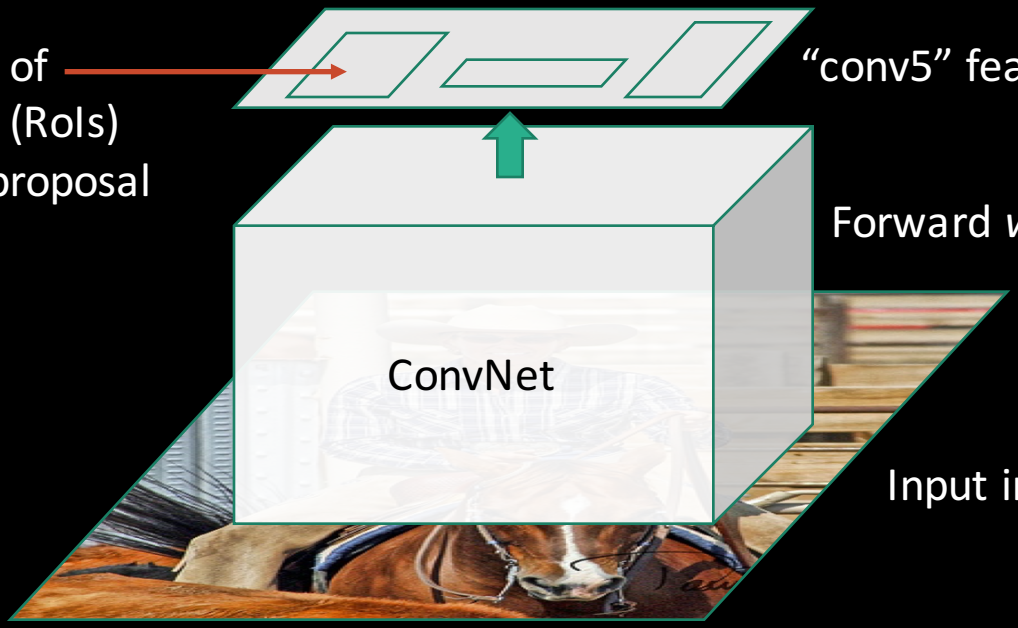
He et al. ECCV14.

SPP-net



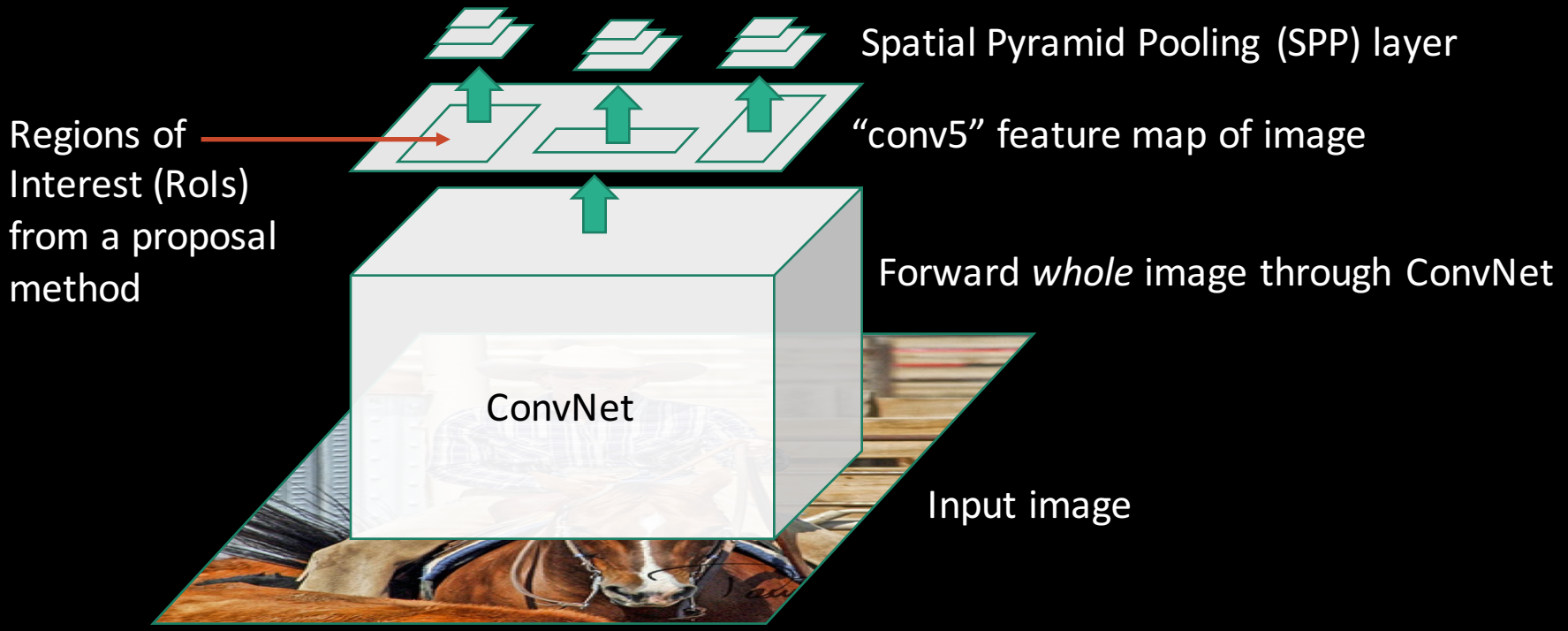
He et al. ECCV14.

SPP-net



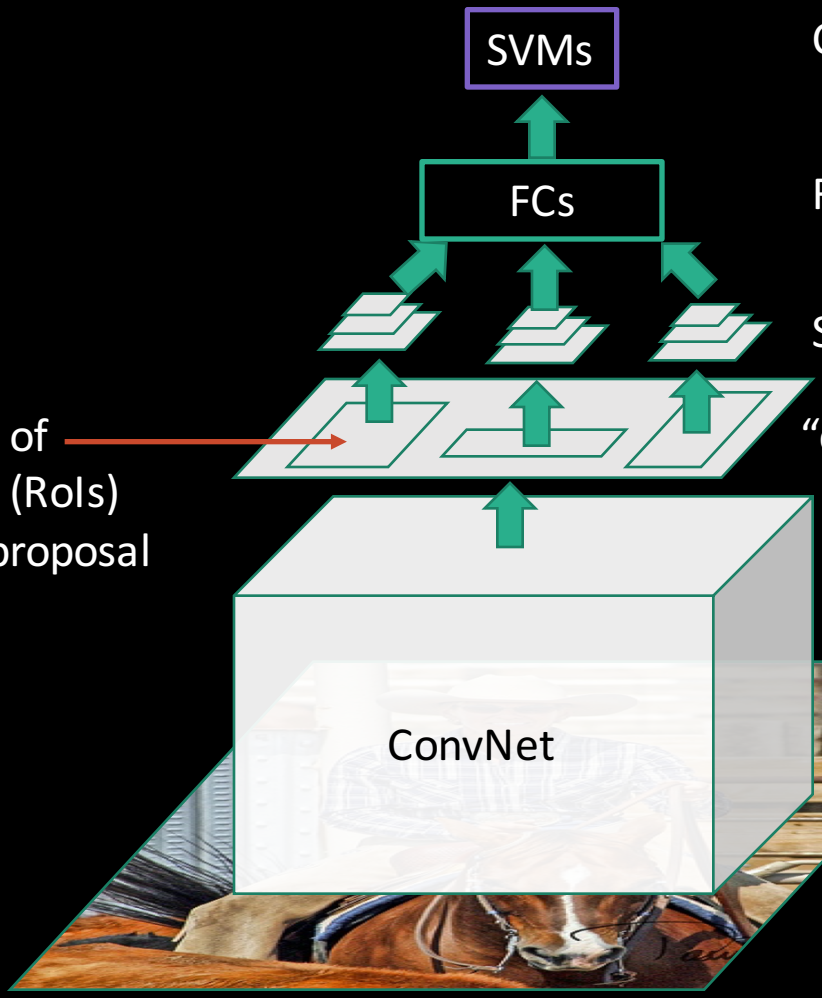
He et al. ECCV14.

SPP-net



He et al. ECCV14.

SPP-net



Classify regions with SVMs

Fully-connected layers

Spatial Pyramid Pooling (SPP) layer

"conv5" feature map of image

Regions of Interest (RoIs) from a proposal method

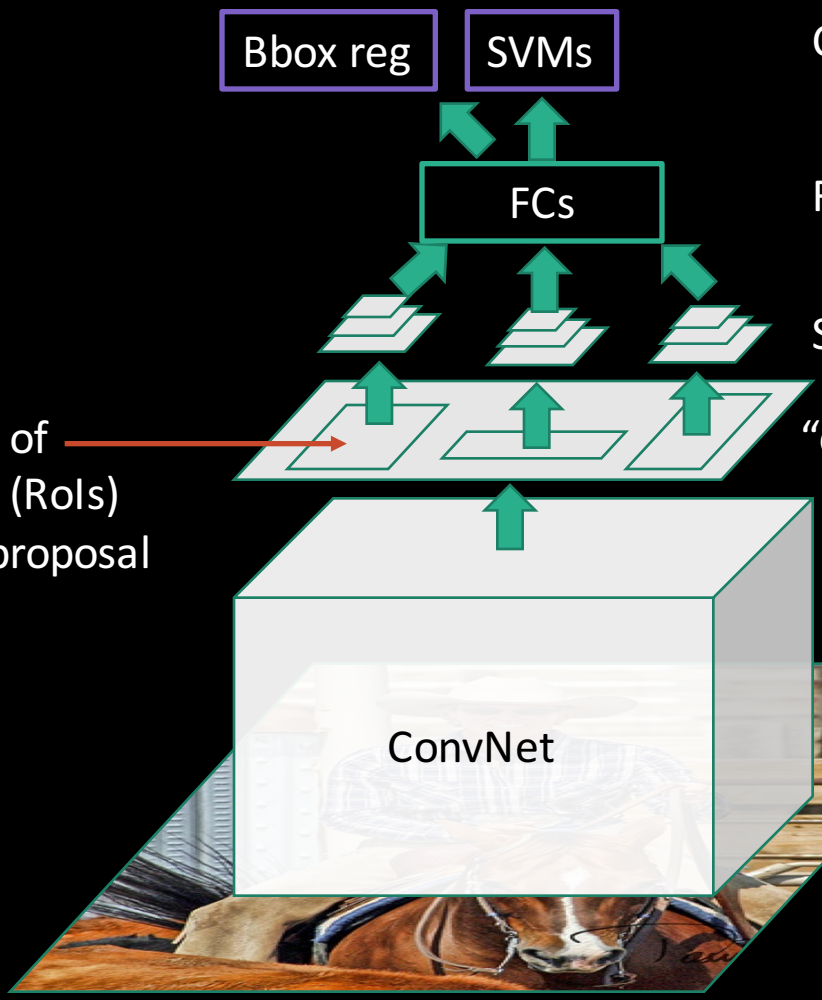
Forward *whole* image through ConvNet

Input image

He et al. ECCV14.

Post hoc component

SPP-net



Apply bounding-box regressors

Classify regions with SVMs

Fully-connected layers

Spatial Pyramid Pooling (SPP) layer

"conv5" feature map of image

Regions of Interest (RoIs) from a proposal method

Forward *whole* image through ConvNet

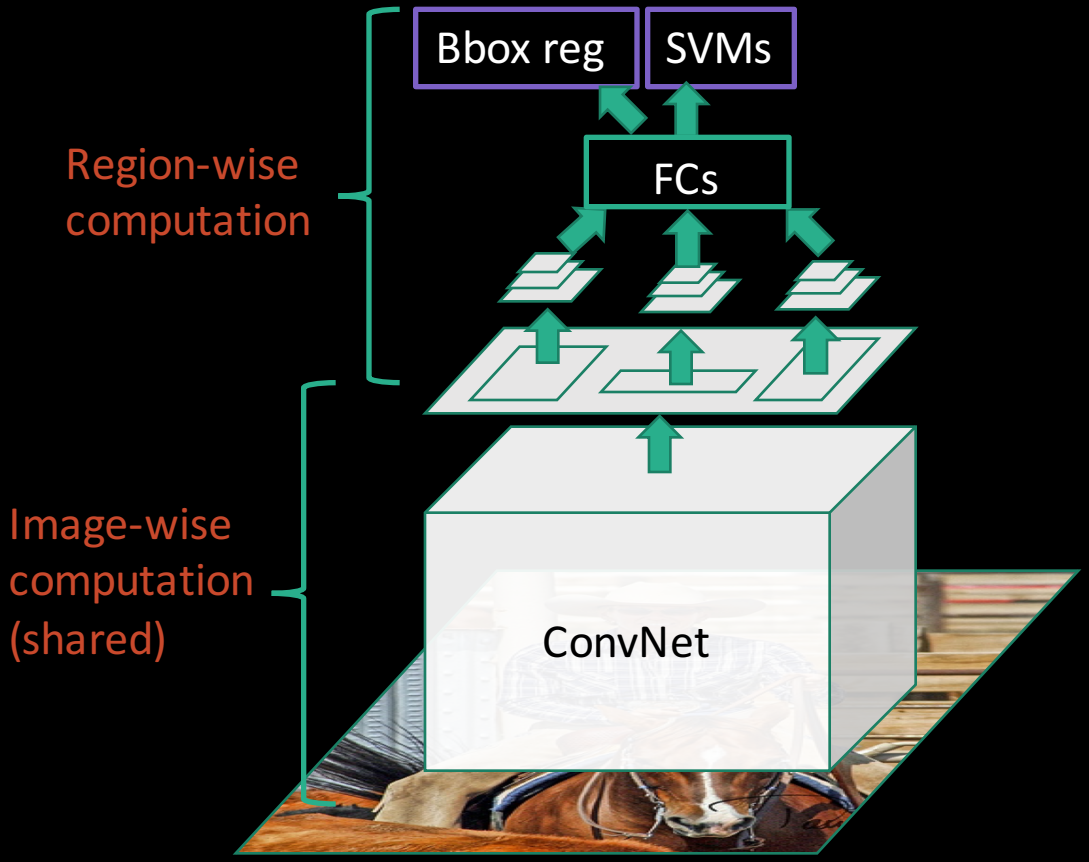
Input image

He et al. ECCV14.

Post hoc component

What's good about SPP-net?

- Fixes one issue with R-CNN: makes testing fast



Post hoc component

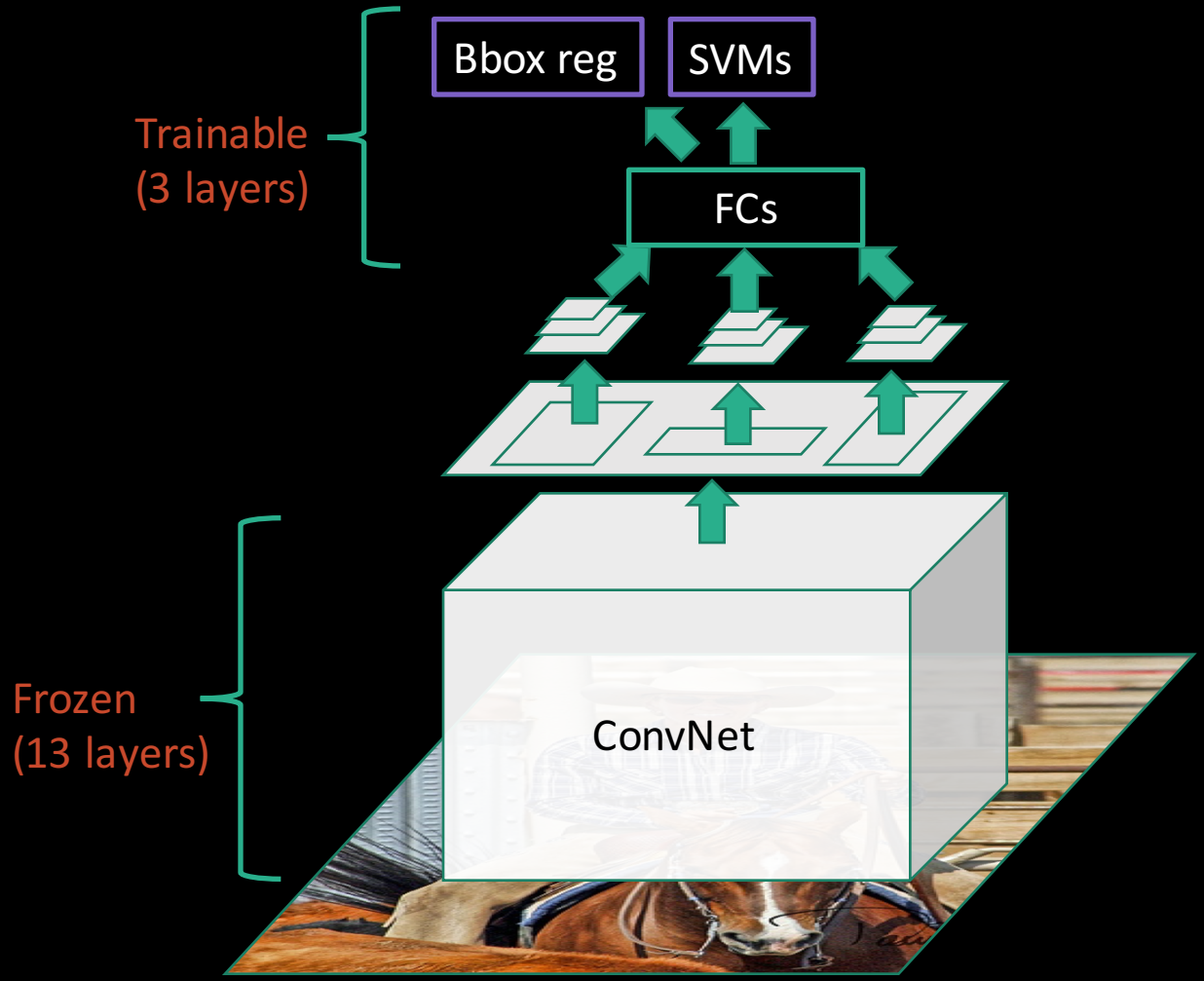
What's wrong with SPP-net?

- Inherits the rest of R-CNN's problems
  - Ad hoc training objectives
  - Training is slow (25h), takes a lot of disk space

What's wrong with SPP-net?

- Inherits the rest of R-CNN's problems
  - Ad hoc training objectives
  - Training is slow (though faster), takes a lot of disk space
- Introduces a new problem: cannot update parameters below SPP layer during training

SPP-net: the main limitation



He et al. ECCV14.

Post hoc component

Fast R-CNN

- Fast test-time, like SPP-net

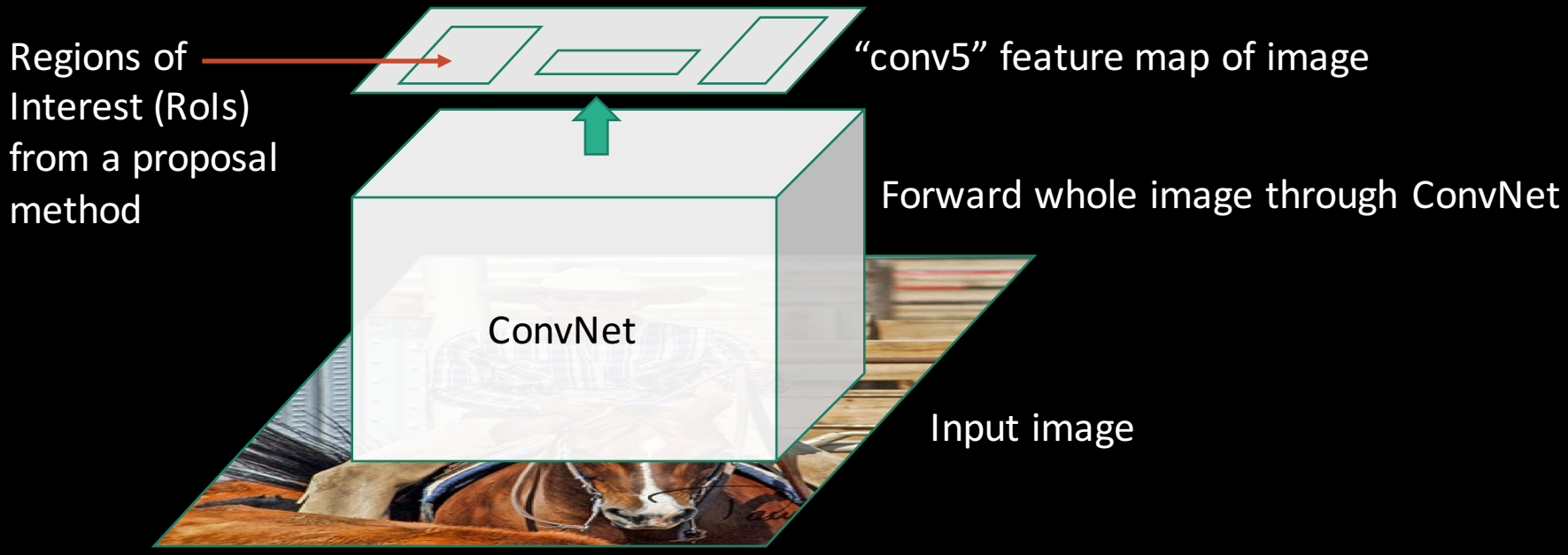
Fast R-CNN

- Fast test-time, like SPP-net
- One network, trained in one stage

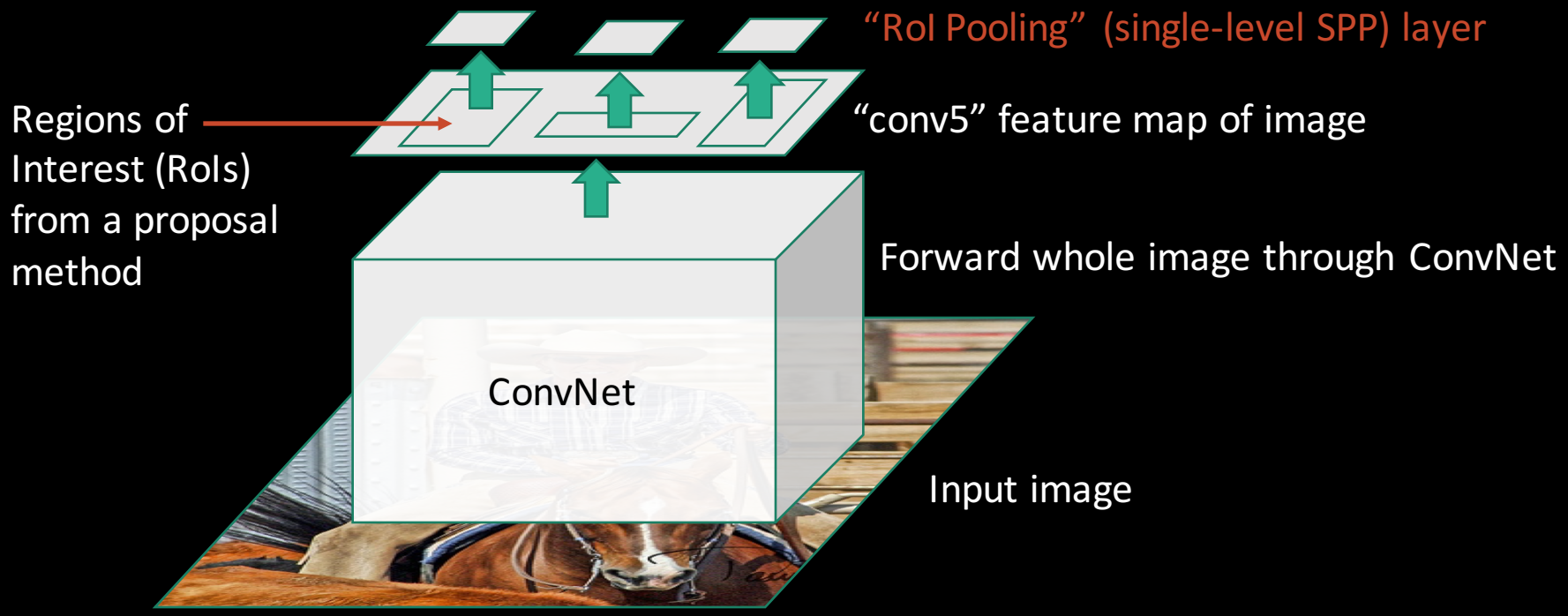
Fast R-CNN

- Fast test-time, like SPP-net
- One network, trained in one stage
- Higher mean average precision than slow R-CNN and SPP-net

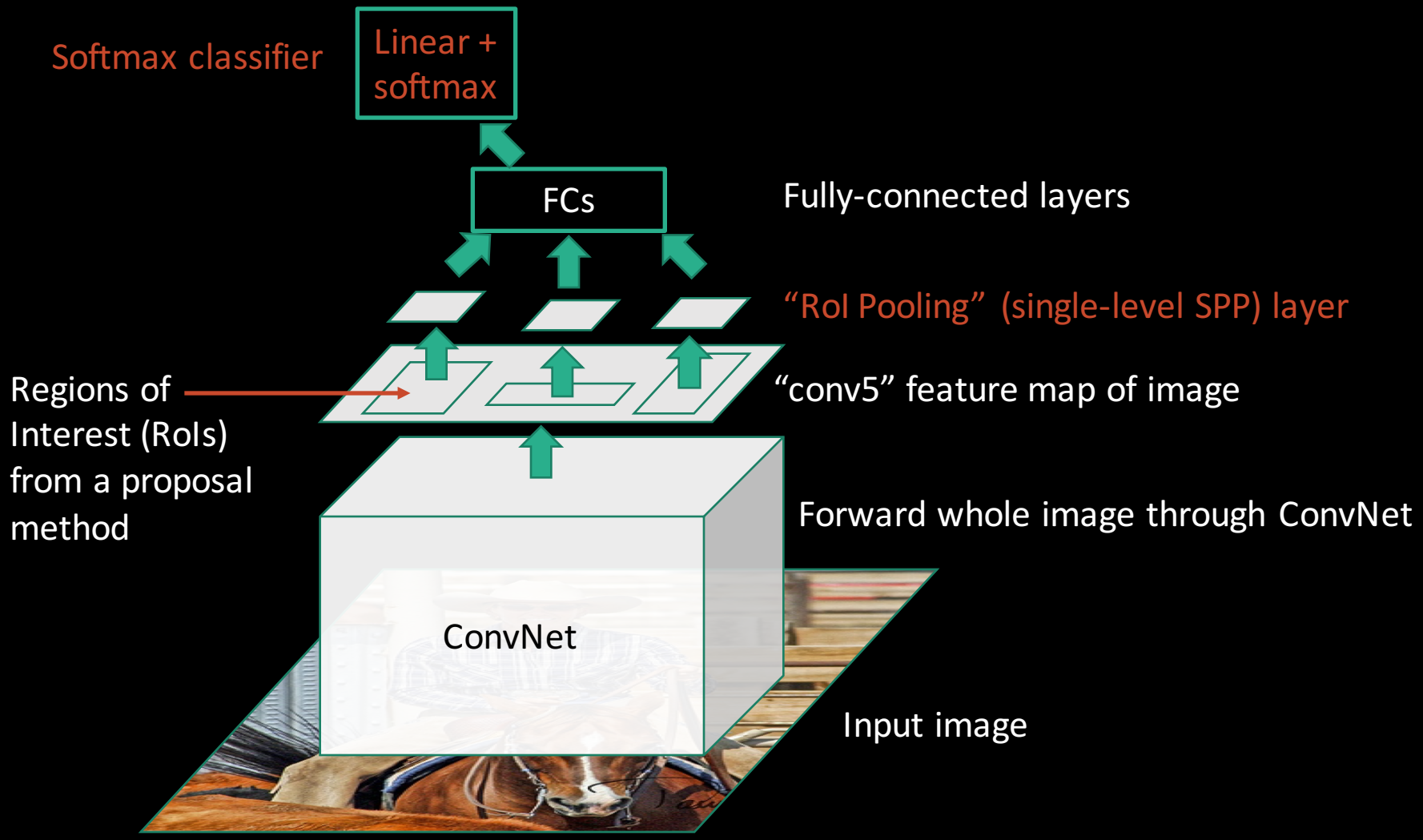
Fast R-CNN (test time)



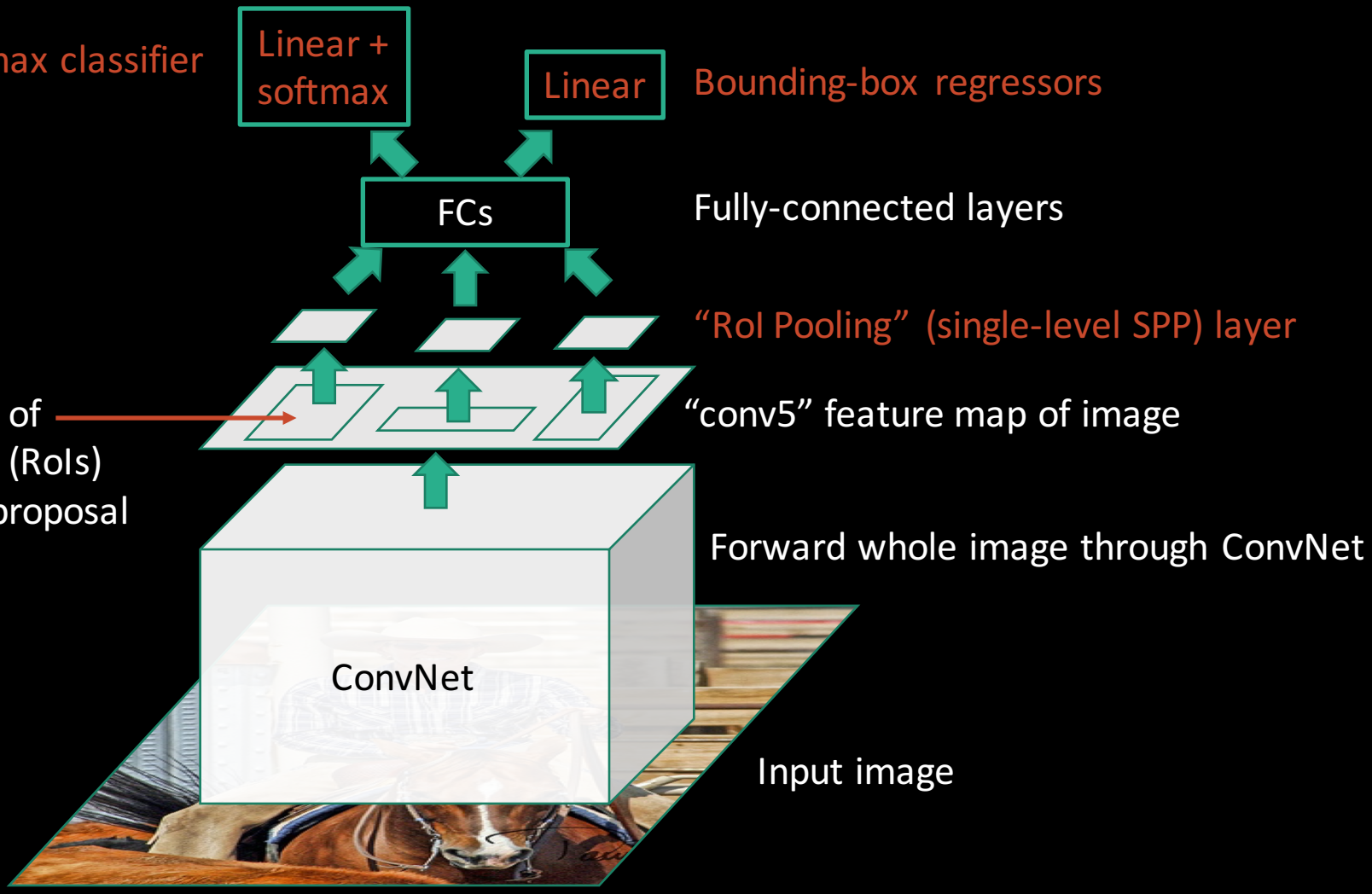
Fast R-CNN (test time)



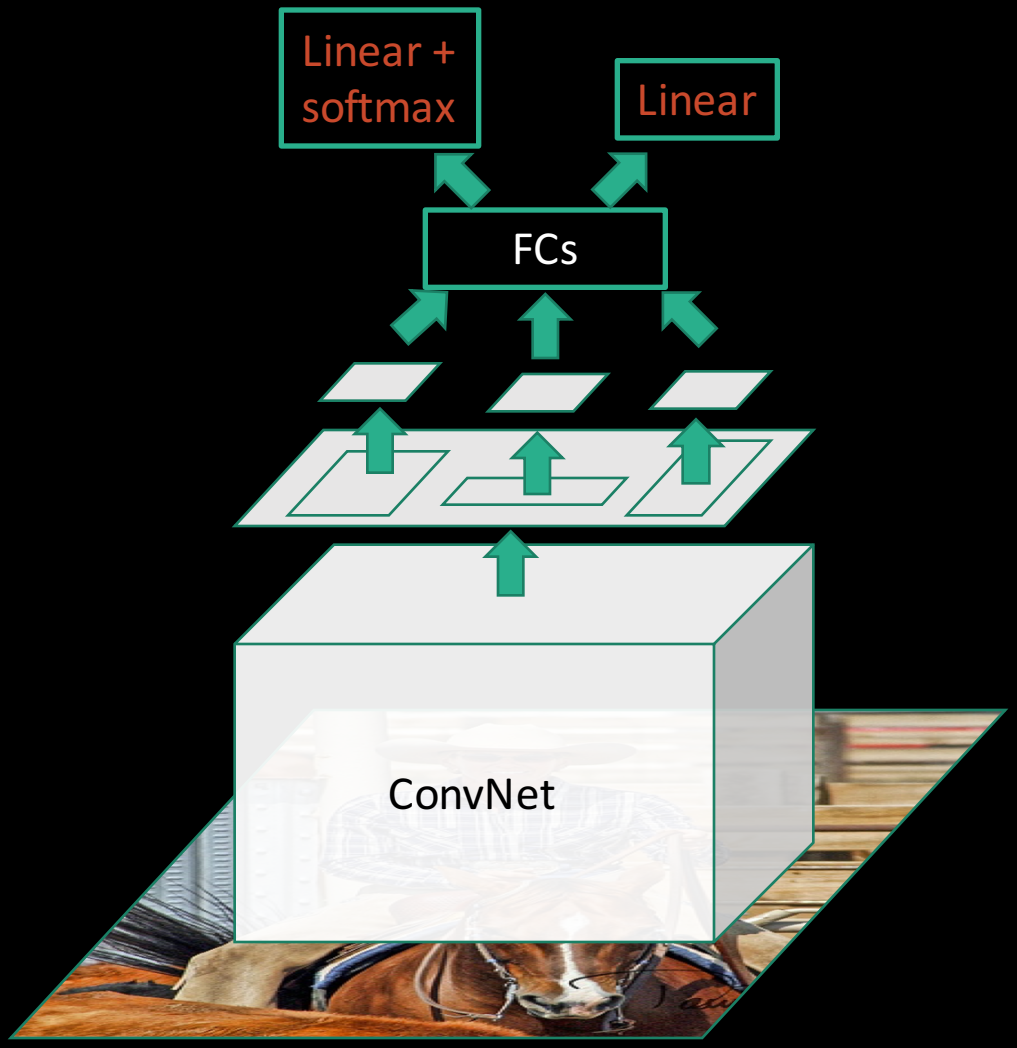
Fast R-CNN (test time)



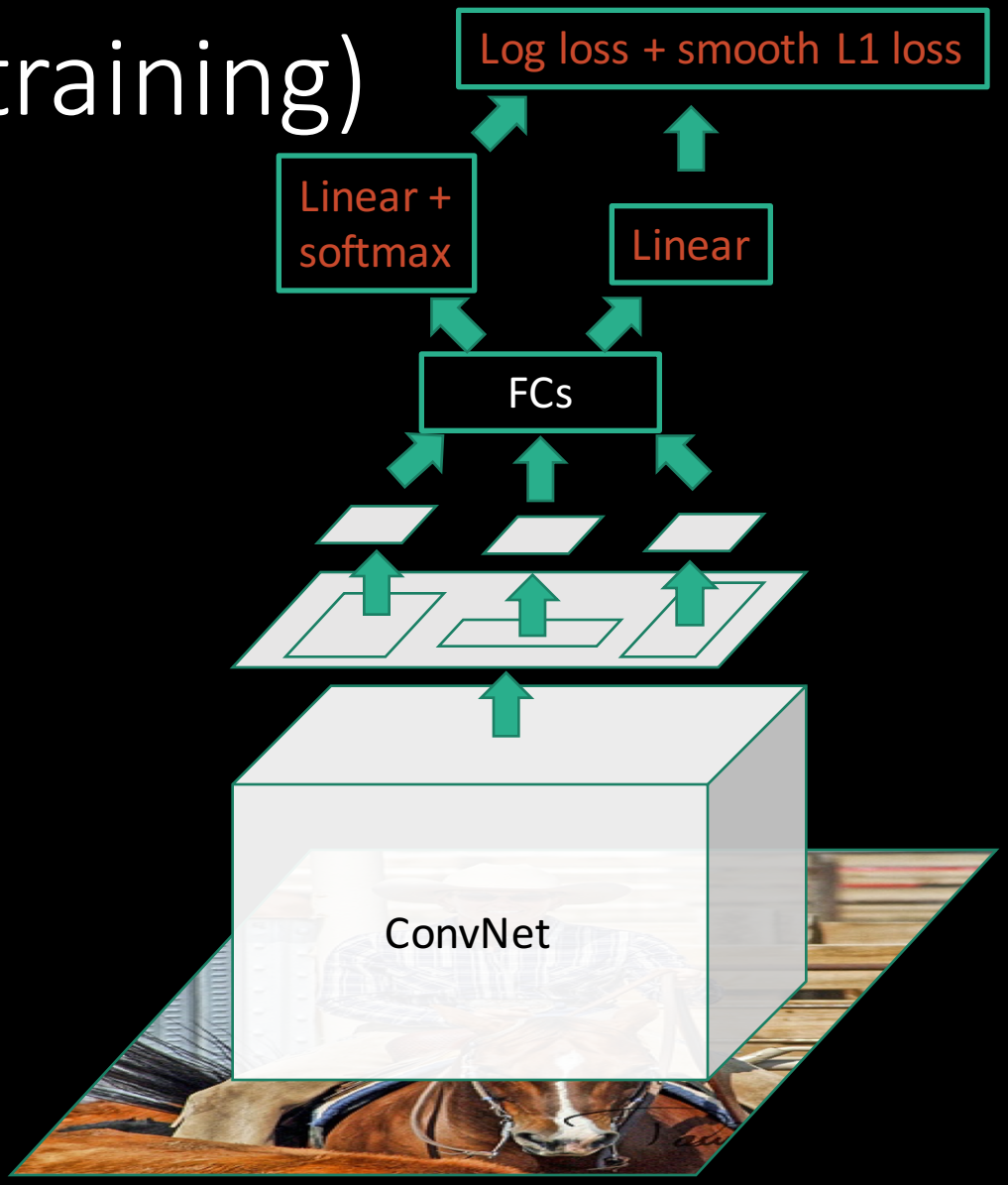
Fast R-CNN (test time)



Fast R-CNN (training)

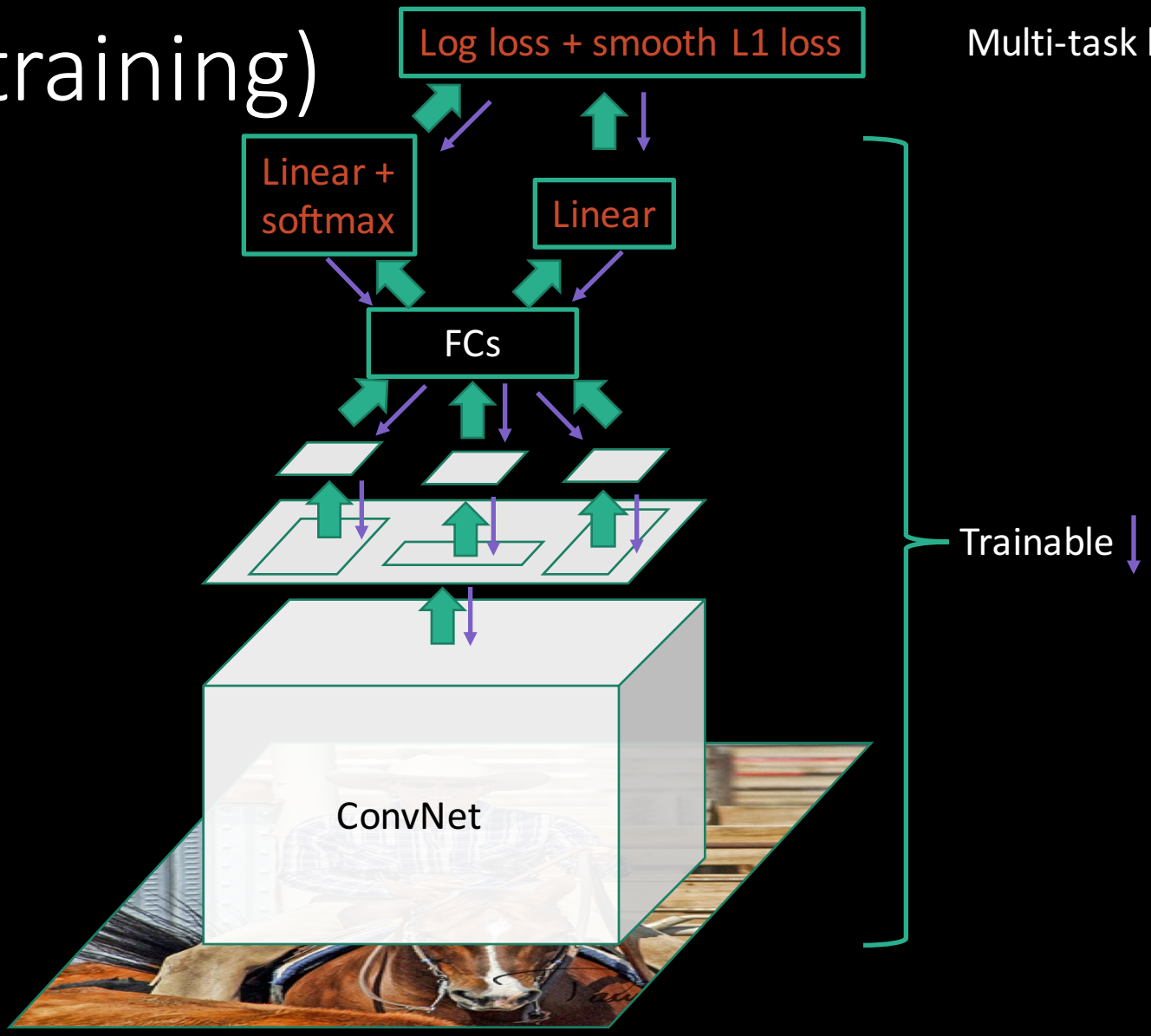


Fast R-CNN (training)

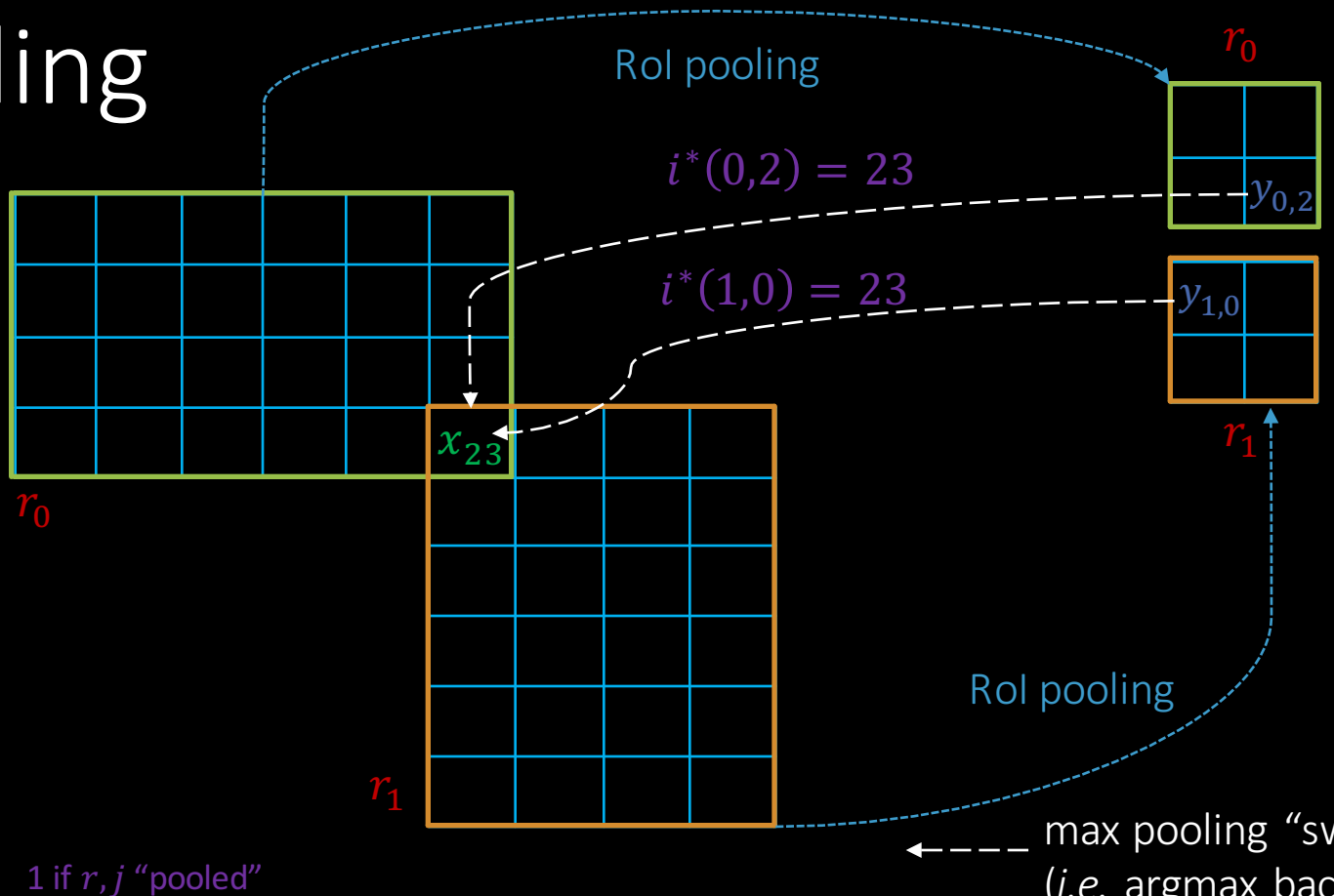


Multi-task loss

Fast R-CNN (training)



Obstacle #1: Differentiable RoI pooling



$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i^*(r,j)] \frac{\partial L}{\partial y_{rj}}$$

1 if r, j "pooled" input i; 0 o/w

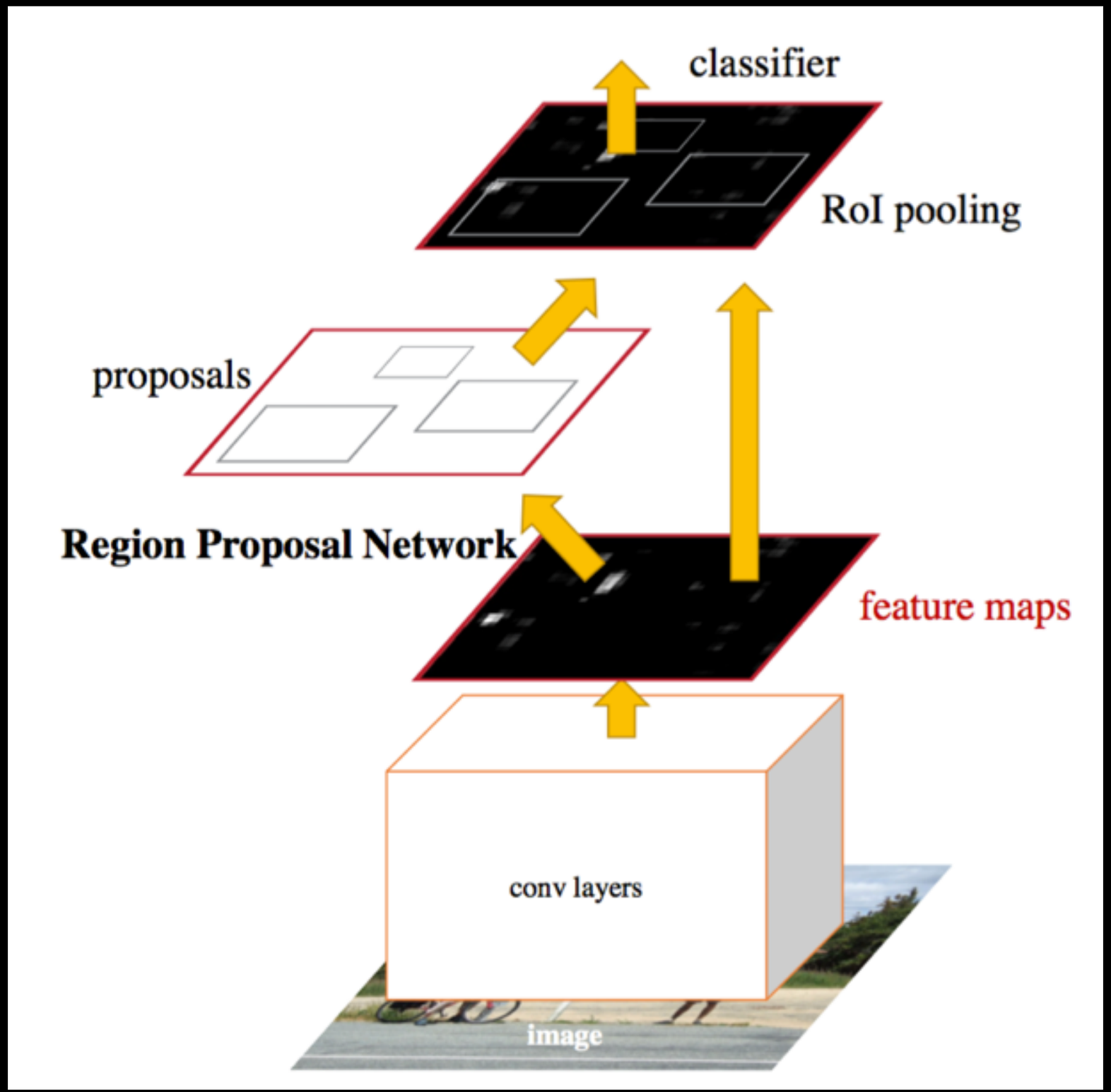
Partial for x_i

Over regions r, locations j

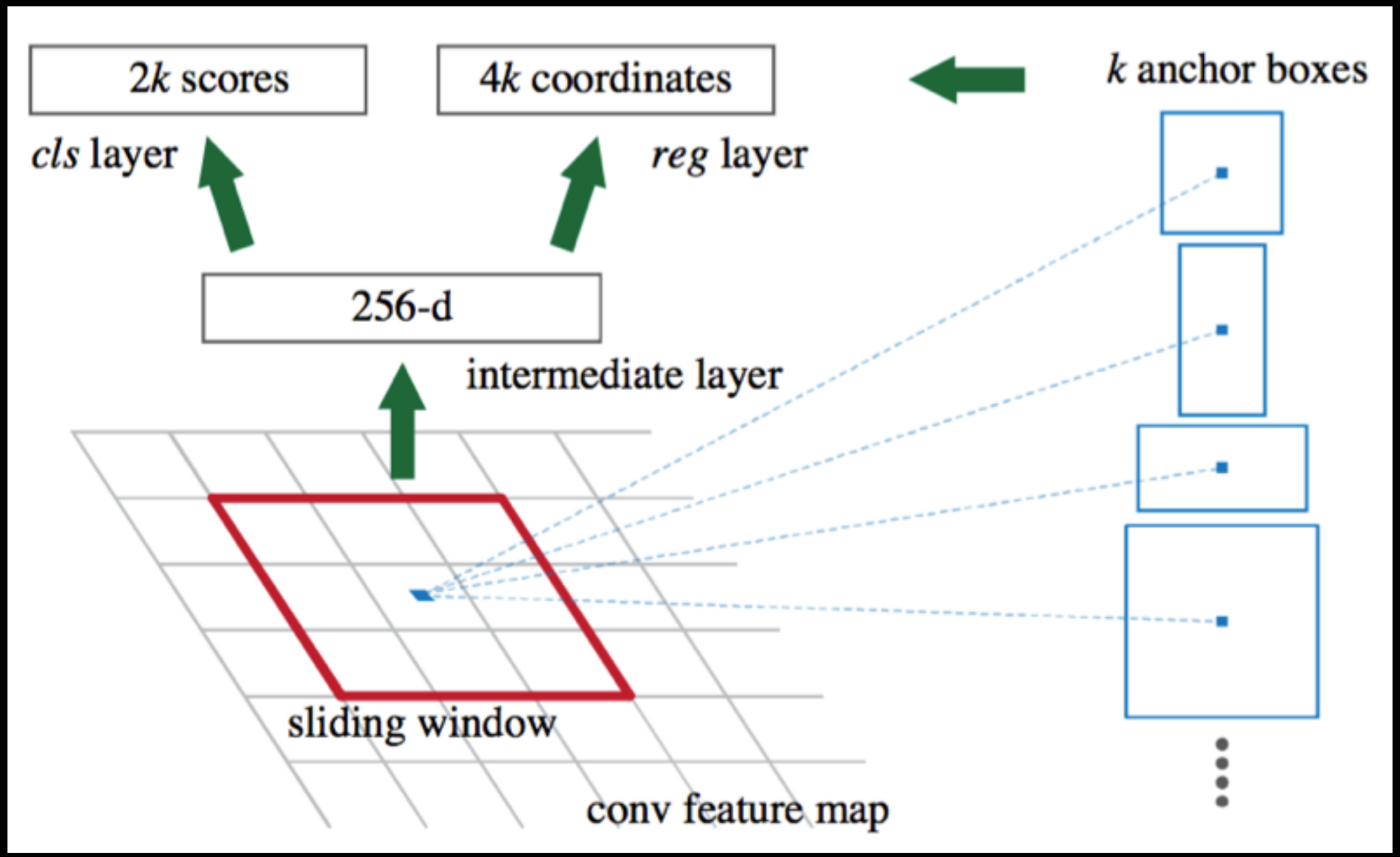
Partial from next layer

Faster R-CNN

Faster R-CNN


classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image

Region Proposal Network



Training Scheme

- Alternating Training:
  1, Train RPN
  2, Fixed Proposals, train Fast R-CNN
  3, Fixed Shared CNN, train RPN with Fast R-CNN fixed
  4, Fixed Shared CNN, fine-tune Fast R-CNN

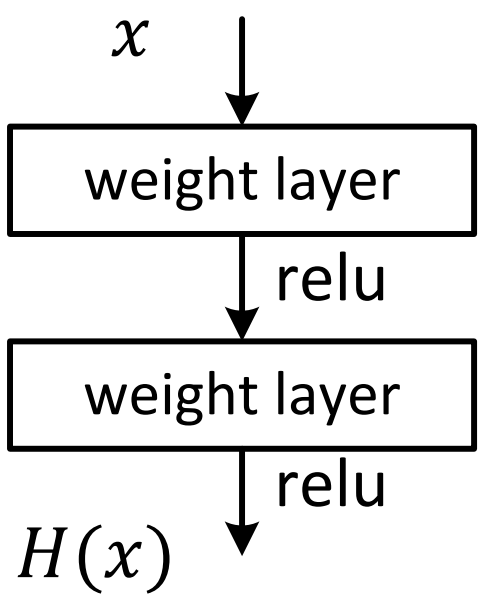
- Approximate Joint Training

- Non-approximate Joint Training

Residual Network

Deep Residual Learning

- Plaint net



any two stacked layers

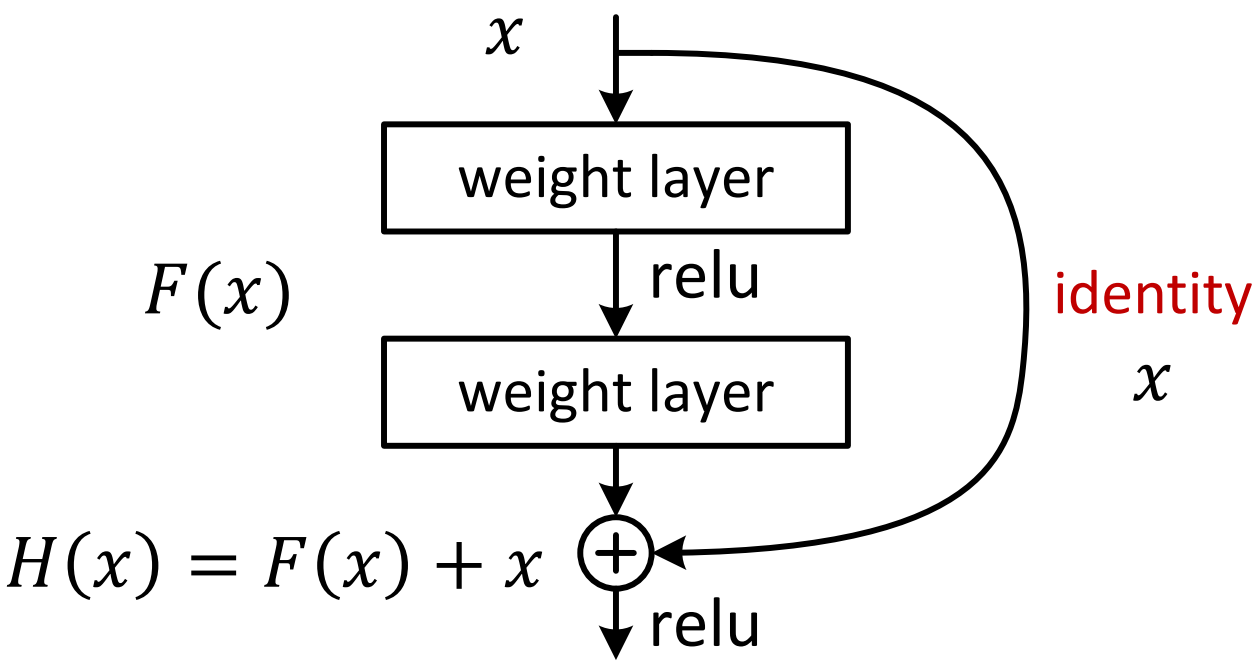
$H(x)$ is any desired mapping,

hope the 2 weight layers fit $H(x)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

~~hope the 2 weight layers fit $H(x)$~~

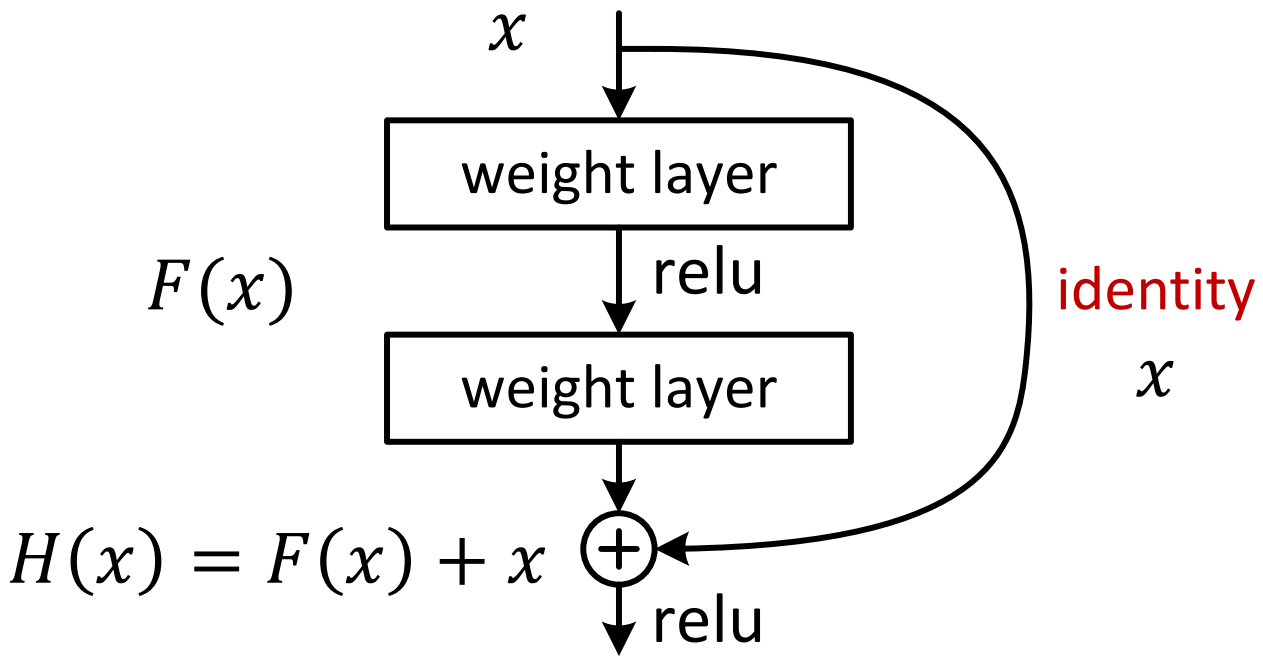
hope the 2 weight layers fit $F(x)$

let $H(x) = F(x) + x$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity



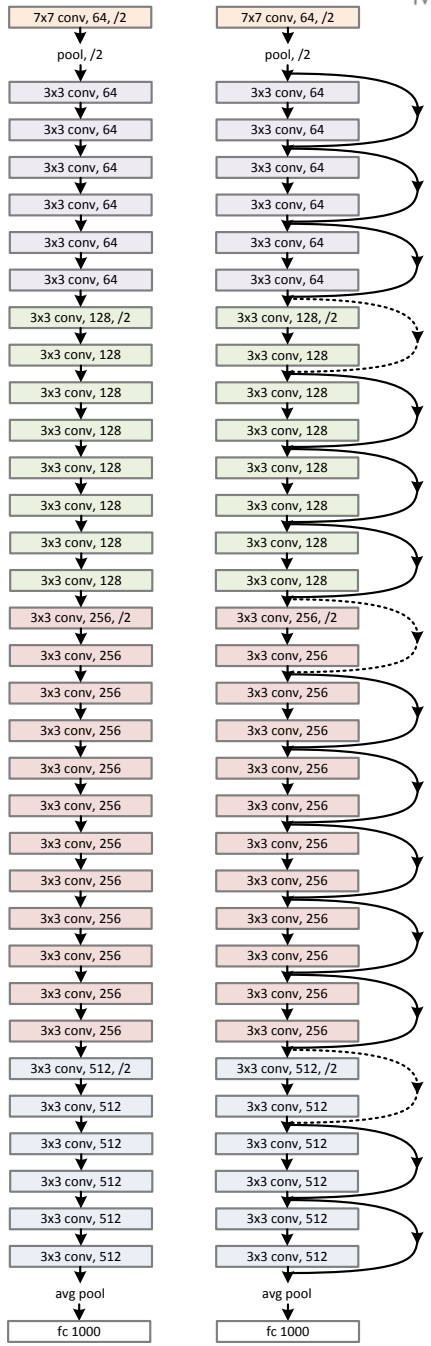
- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Network “Design”

- Keep it simple
- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2
  - Simple design; just deep!
- Other remarks:
  - no max pooling (almost)
  - no hidden fc
  - no dropout

plain net



ResNet

Microsoft Research

ICCV15 International Conference on Computer Vision

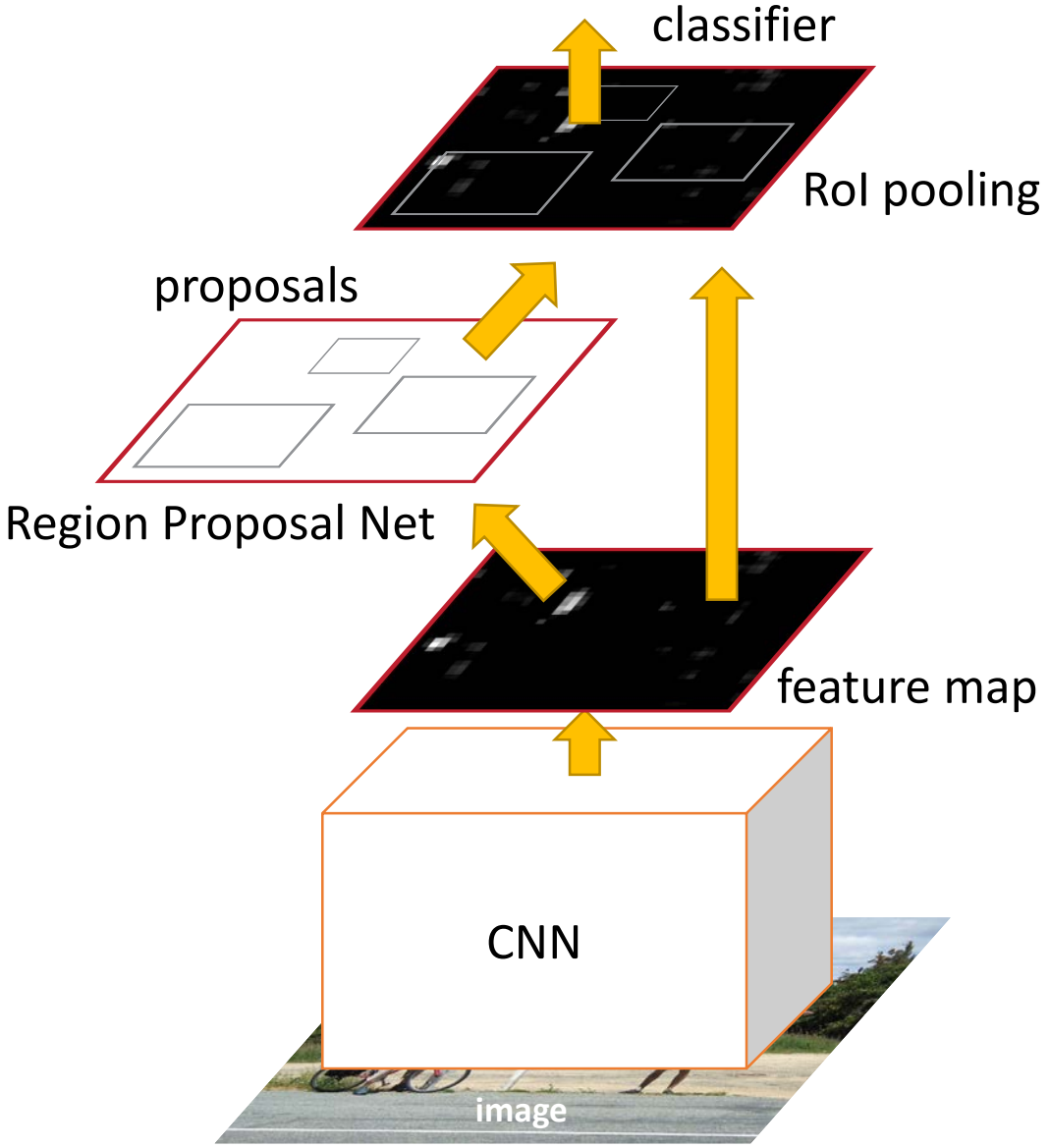
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Object Detection (brief)

- Simply "Faster R-CNN + ResNet"

Faster R-CNN baseline	mAP@.5	mAP@.5:.95
VGG-16	41.5	21.5
ResNet-101	**48.4**	**27.2**

COCO detection results
(ResNet has 28% relative gain)

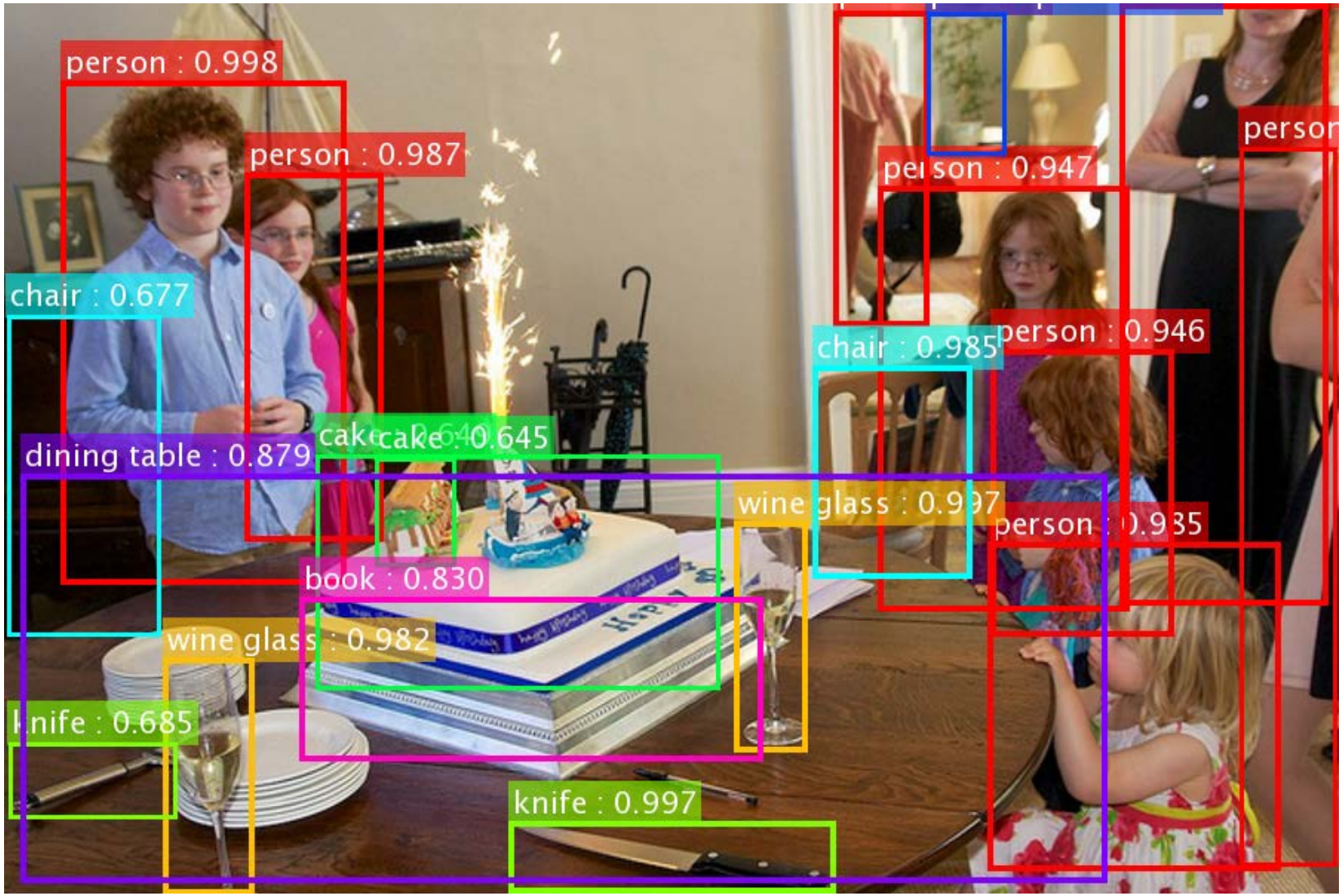


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

Object Detection (brief)

- RPN learns proposals by extremely deep nets
  - We use only 300 proposals (no SS/EB/MCG!)
- Add what is just missing in Faster R-CNN…
  - Iterative localization
  - Context modeling
  - Multi-scale testing
- All are based on CNN features; all are end-to-end (train and/or inference)
- All benefit more from deeper features – cumulative gains!

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

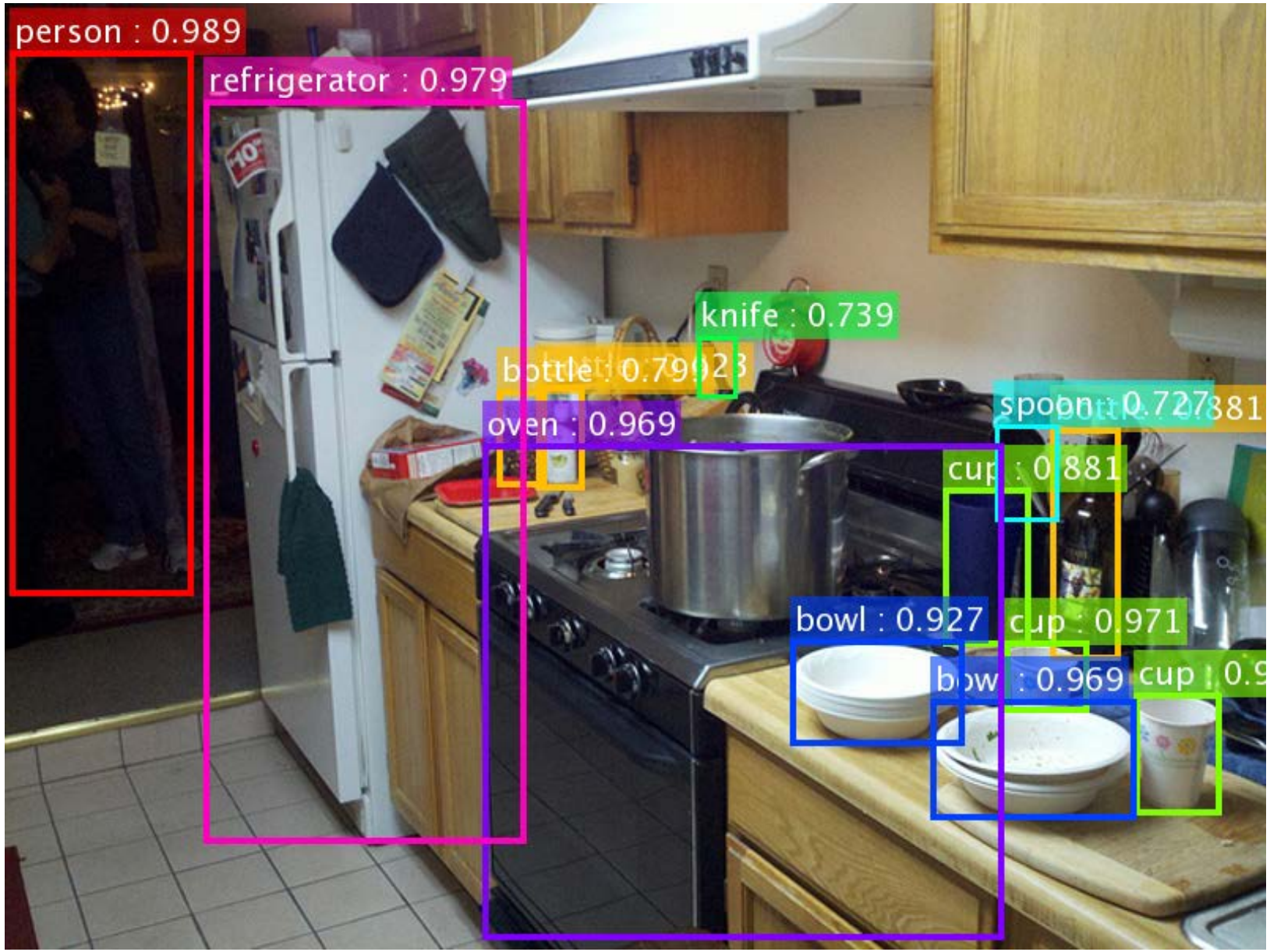


Our results on COCO – too many objects, let's check carefully!

*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

Microsoft Research

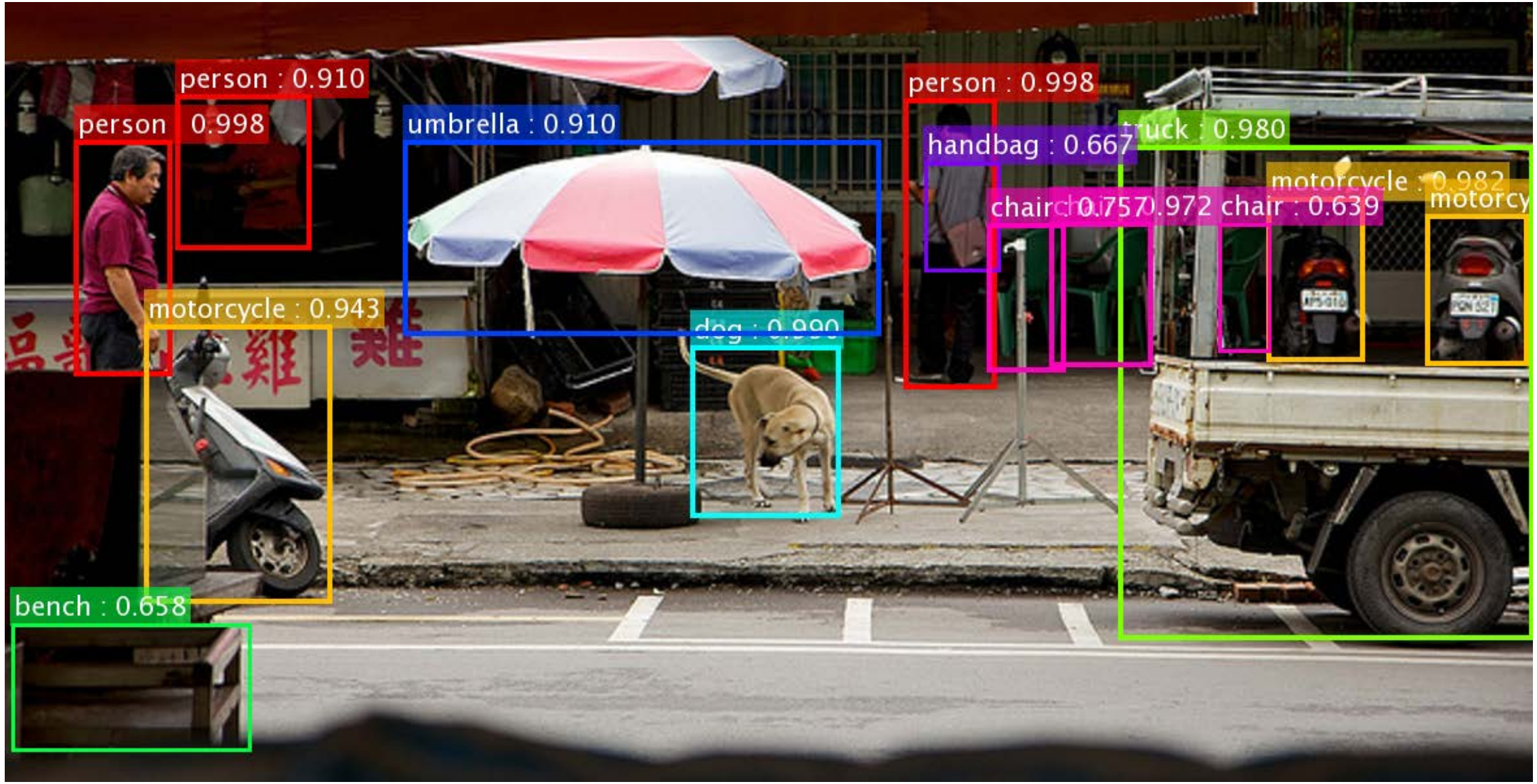


*the original image is from the COCO dataset

ICCV15 International Conference on Computer Vision

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.



*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Thanks!