THE UNIVERSITY OF BRITISH COLUMBIA

Topics in AI (CPSC 532S):
Multimodal Learning with Vision, Language and Sound

Lecture 7: Convolutional Neural Networks (part 3)

Logistics:

Assignment 2 was due **yesterday**

Assignment 3 will be posted **tonight** …

Logistics:

Assignment 2 was due **yesterday**

Assignment 3 will be posted **tonight** …

Final Projects … poll results

… 3 more students voted for individual projects than for survey

we will start process of forming groups

Computer **Vision Problems** (no language for now)

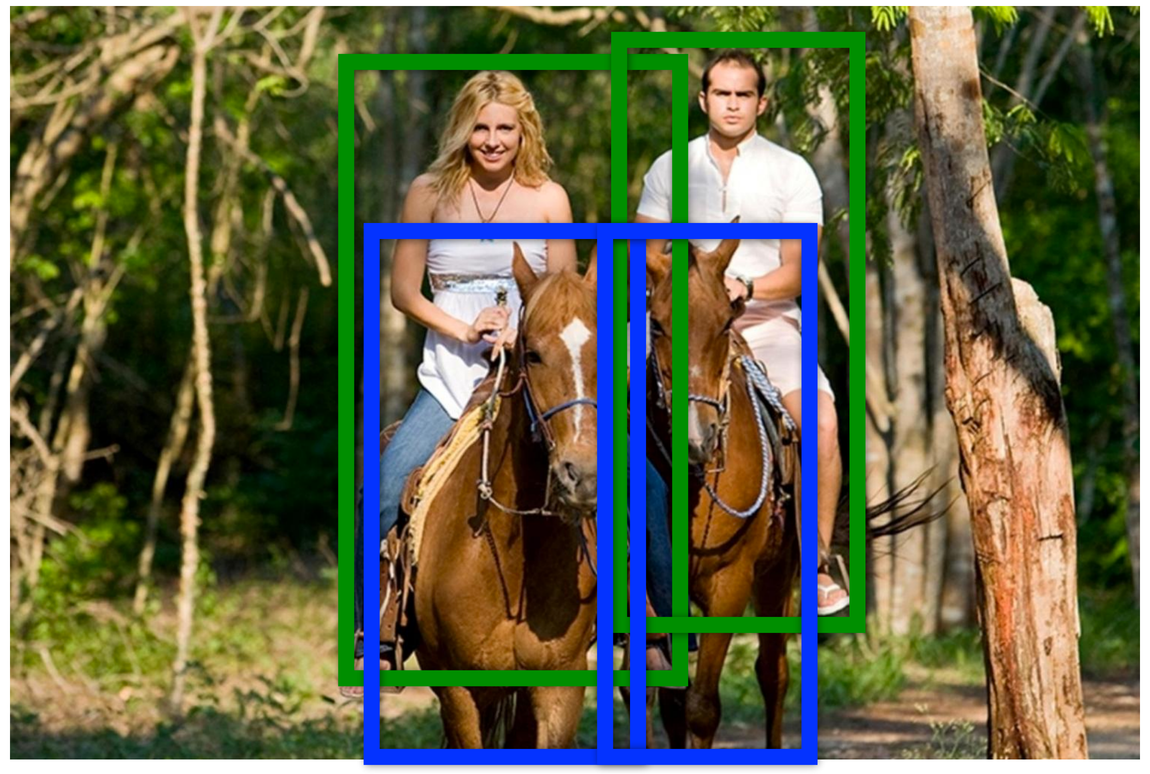
Categorization

Multi-**class**: Horse
Church
Toothbrush
Person

IMAGENET

Multi-**label**: **Horse**
Church
Toothbrush
Person

Detection



Horse (x, y, w, h)
Horse (x, y, w, h)
Person (x, y, w, h)
Person (x, y, w, h)

COCO
Common Objects in Context

Segmentation

Horse
Person

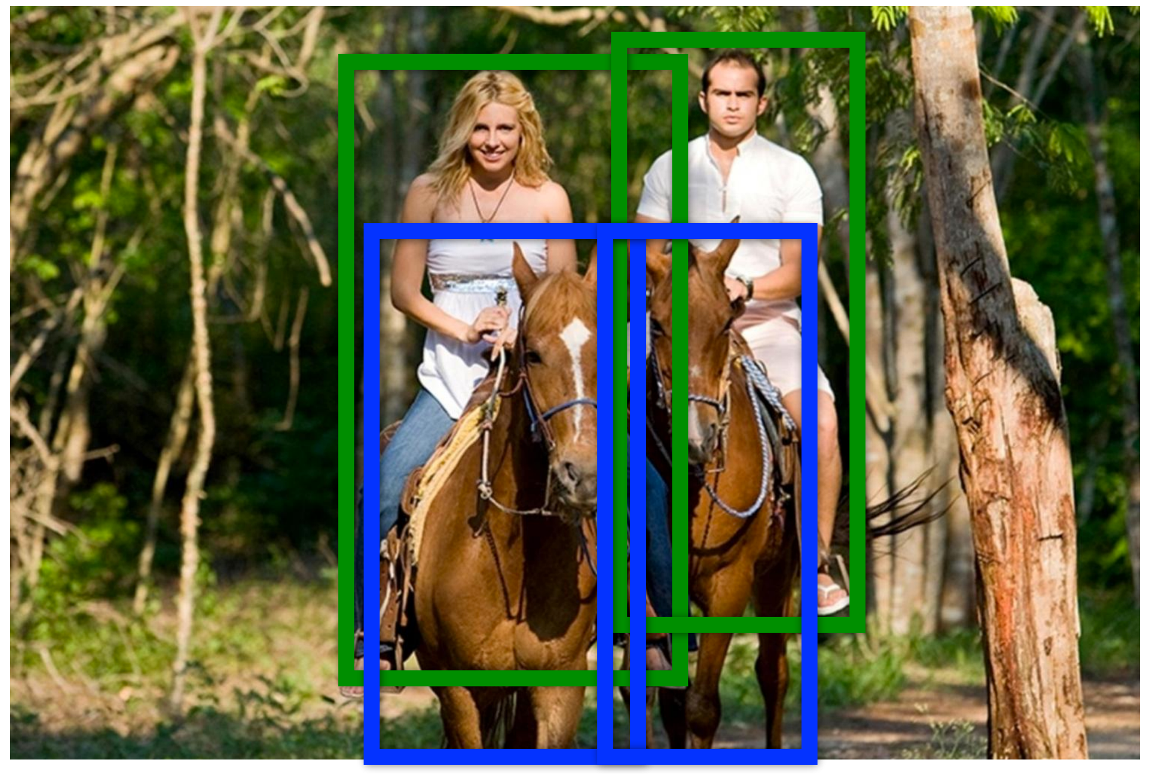
COCO
Common Objects in Context

Instance Segmentation

Horse1
Horse2
Person1
Person2

Computer **Vision Problems** (no language for now)

Detection

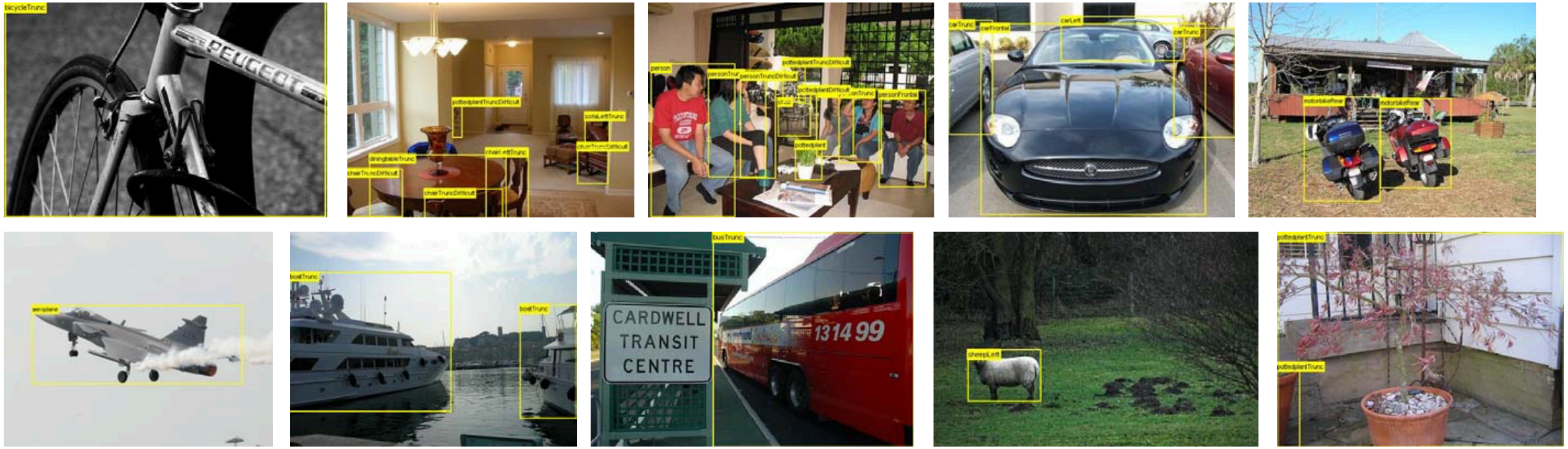


Horse (x, y, w, h)
Horse (x, y, w, h)
Person (x, y, w, h)
Person (x, y, w, h)

COCO
Common Objects in Context

Datasets: Pascal VOC

20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV



Real images downloaded from flickr, not filtered for “quality”

* slide from Andrew Zisserman

Datasets: Pascal VOC

20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

	Training	Testing
Images	10,103	9,637
Objects	23,374	22,992

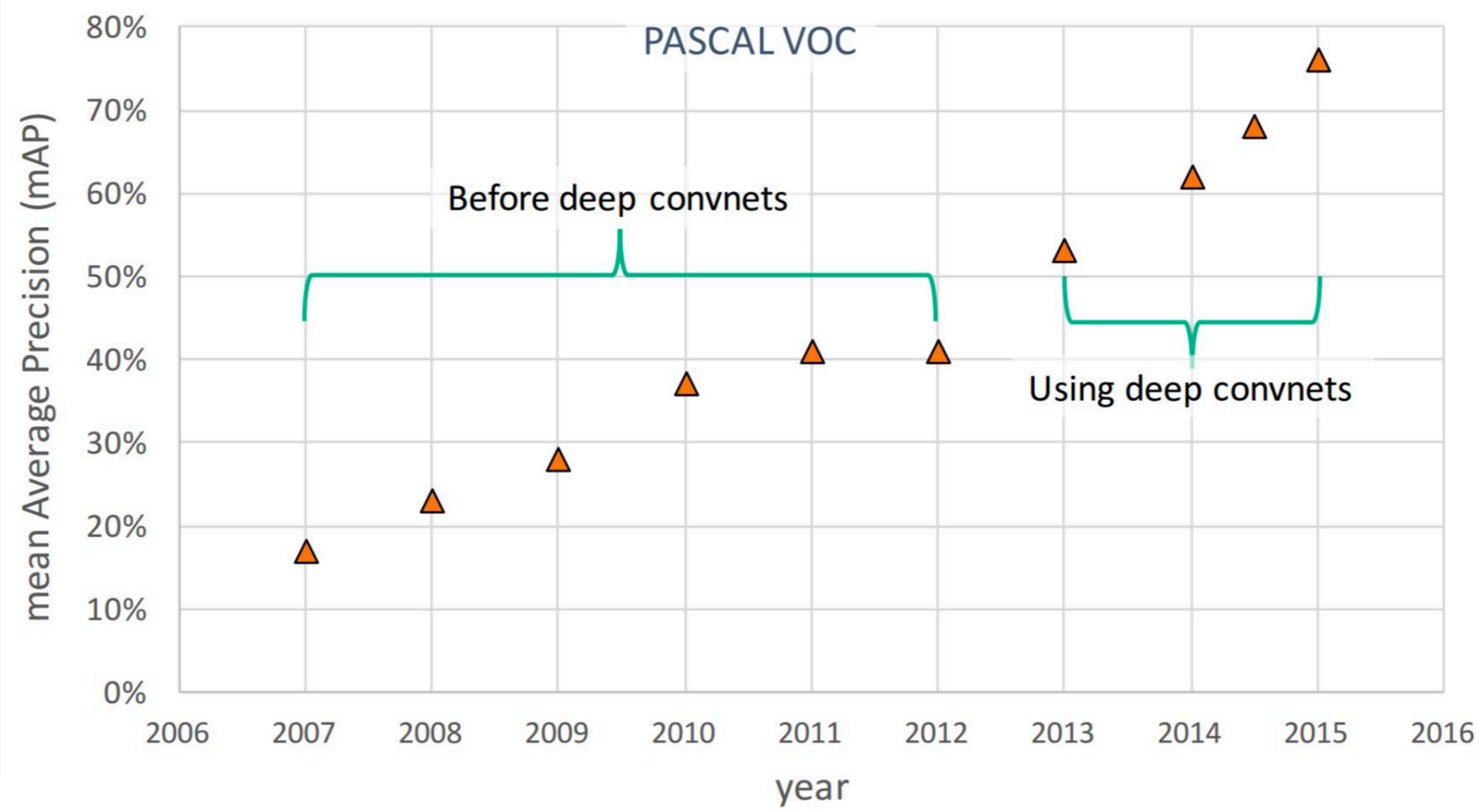
Real images downloaded from flickr, not filtered for “quality”

* slide from Andrew Zisserman

Datasets: COCO

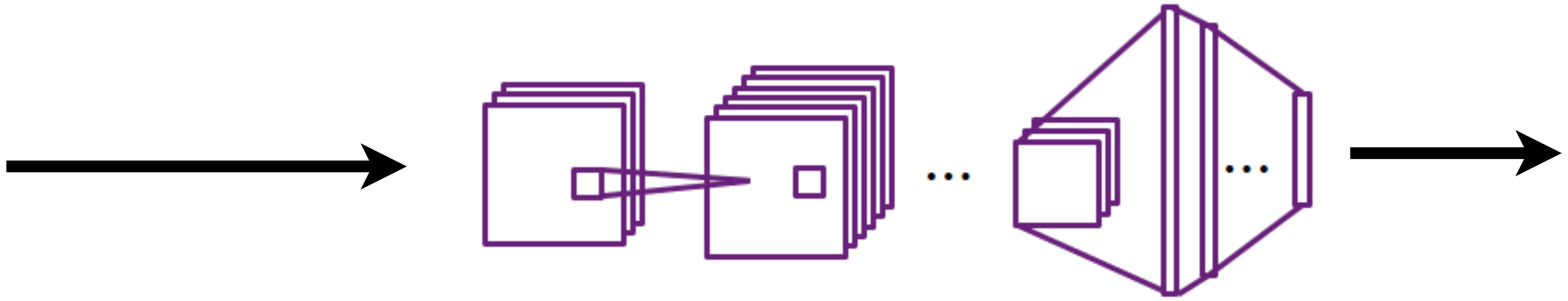
- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

Object **Detection**



* plot from Ross Girshick, 2015

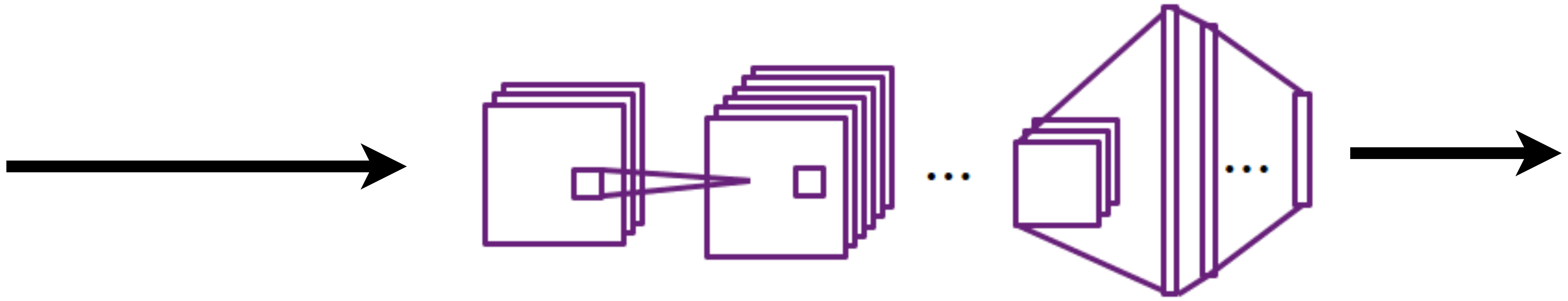
Object **Detection** as Regression Problem



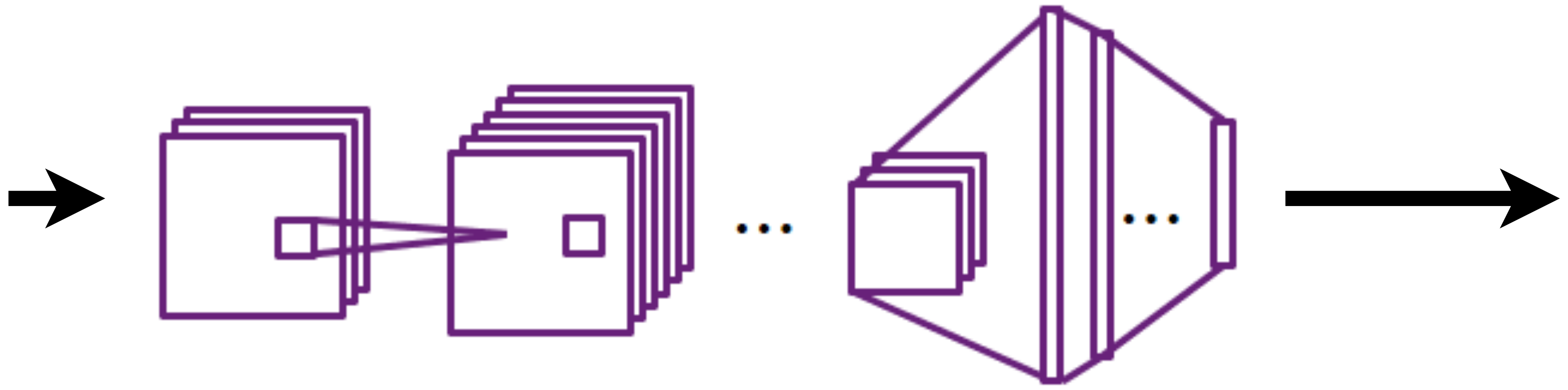
CAT (x, y, w ,h)

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Object **Detection** as Regression Problem



CAT (x, y, w ,h)



DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
…

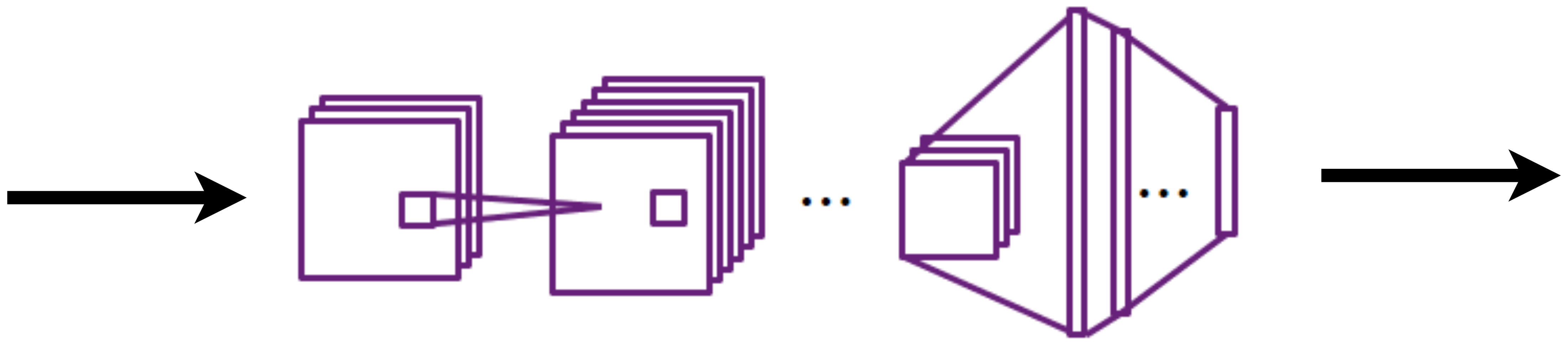
Object **Detection** as Regression Problem

CAT (x, y, w ,h)

Problem: each image needs a different number of outputs

DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
DUCK (x, y, w ,h)
…

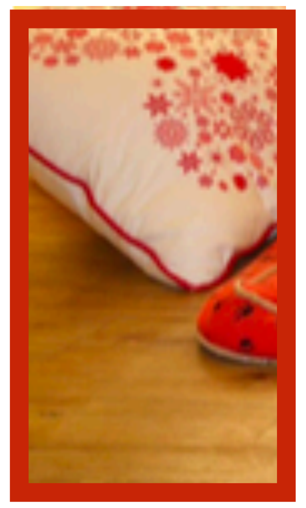
Object **Detection** as Classification Problem

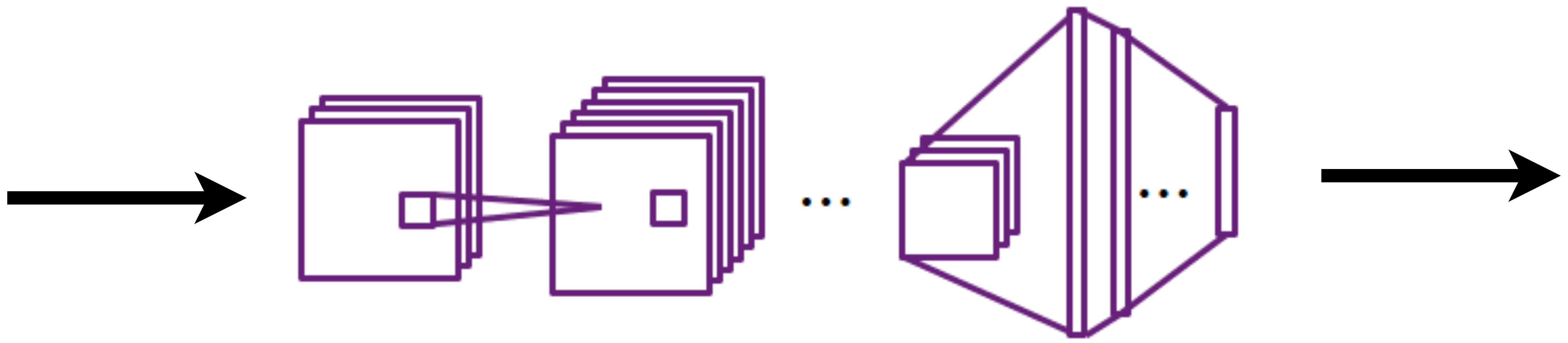


Category	**Prediction**
Dog	No
Cat	No
Couch	No
Flowers	No
Background	**Yes**
…	…

Apply CNN to many different crops in the image and (classification) CNN classifies each patch as object or background

Object **Detection** as Classification Problem

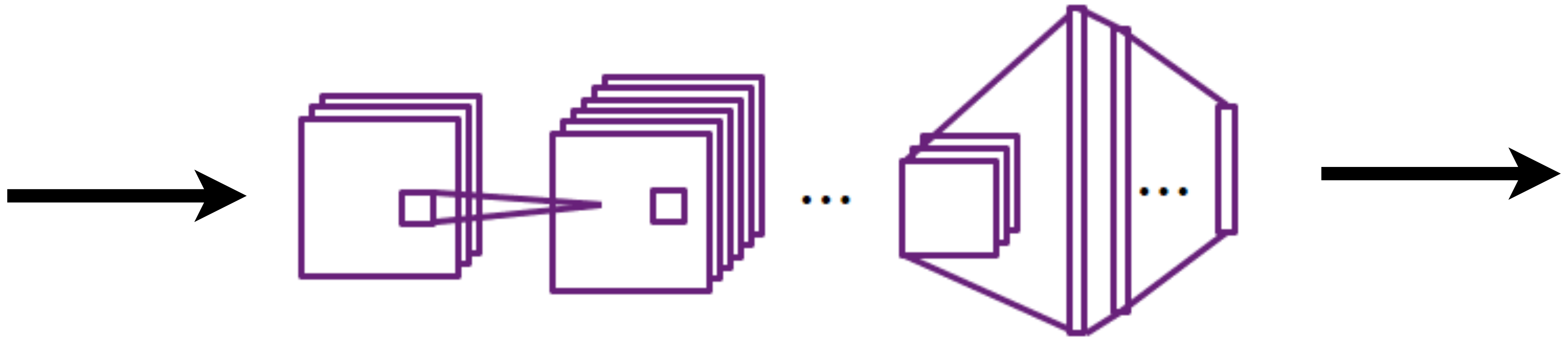




Category	**Prediction**
Dog	No
Cat	No
Couch	No
Flowers	No
Background	**Yes**
…	…

Apply CNN to many different crops in the image and (classification) CNN classifies each patch as object or background

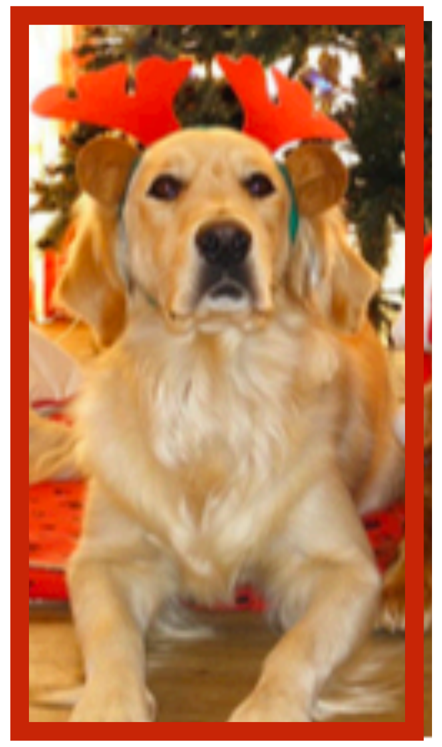
Object **Detection** as Classification Problem

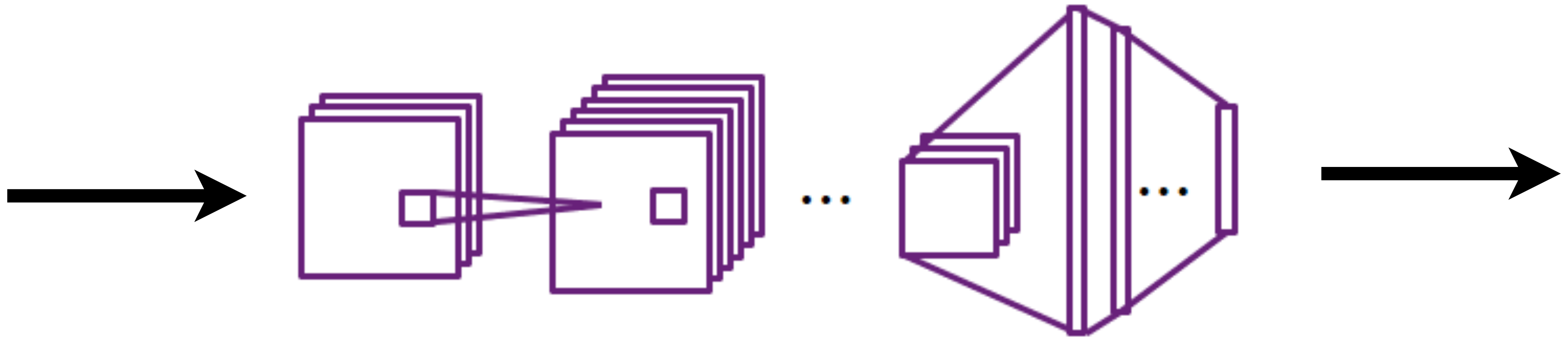


Category	**Prediction**
Dog	**Yes**
Cat	No
Couch	No
Flowers	No
Background	No
…	…

Apply CNN to many different crops in the image and (classification) CNN classifies each patch as object or background

Object **Detection** as Classification Problem

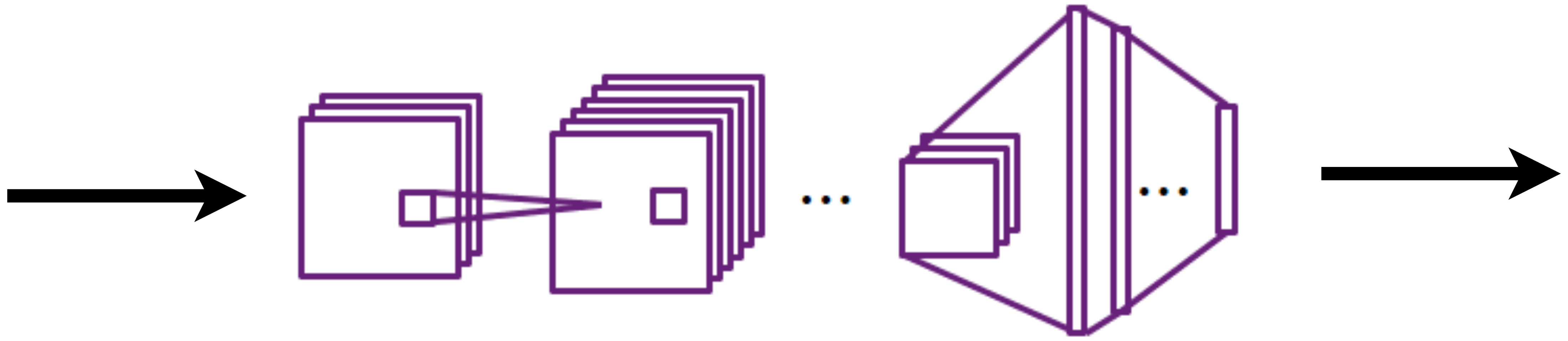




Category	**Prediction**
Dog	**Yes**
Cat	No
Couch	No
Flowers	No
Background	No
…	…

Apply CNN to many different crops in the image and (classification) CNN classifies each patch as object or background

Object **Detection** as Classification Problem



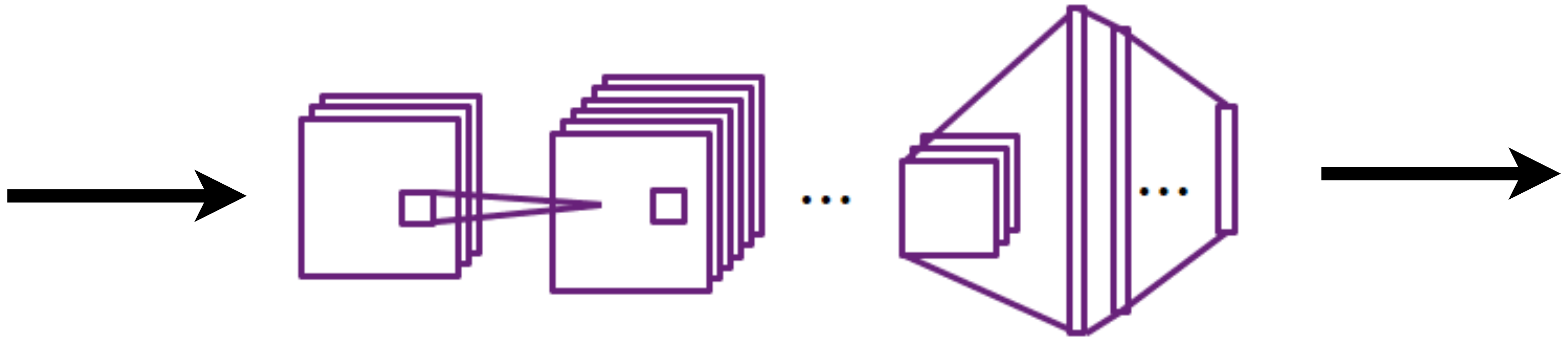
Category	**Prediction**
Dog	No
Cat	**Yes**
Couch	No
Flowers	No
Background	No
…	…

Apply CNN to many different crops in the image and (classification) CNN classifies each patch as object or background

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Object **Detection** as Classification Problem

Problem: Need to apply CNN to **many** patches in each image



Category	**Prediction**
Dog	No
Cat	**Yes**
Couch	No
Flowers	No
Background	No
…	…

Apply CNN to many different crops in the image and (classification) CNN classifies each patch as object or background

Region Proposals (older idea in vision)

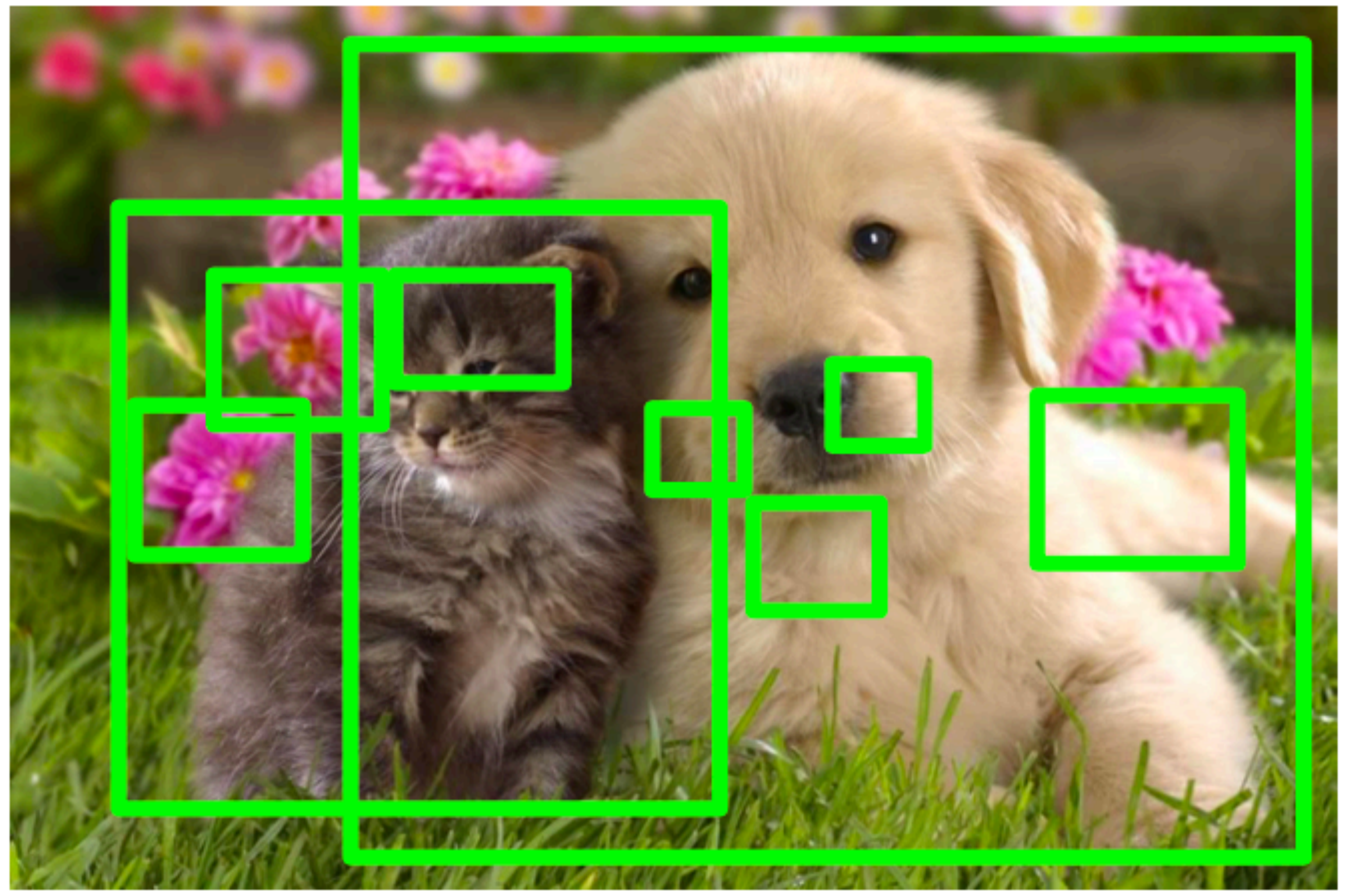
[Alexe et al, TPAMI 2012]
[Uijkings et al, IJCV 2013]
[Cheng et al, CVPR 2014]
[Zitnick and Dollar, ECCV 2014]

Find image **regions that are likely contain objects** (any object at all)

- typically works by looking at histogram distributions, region aspect ratio, closed contours, coherent color

Relatively **fast to run** (Selective Search gives 1000 region proposals in a few seconds on a CPU)

→



Goal: Get "true" object regions to be in as few top K proposals as possible

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

R-CNN

[Girshick et al, CVPR 2014]

Input **Image**

* image from Ross Girshick

R-CNN

[Girshick et al, CVPR 2014]

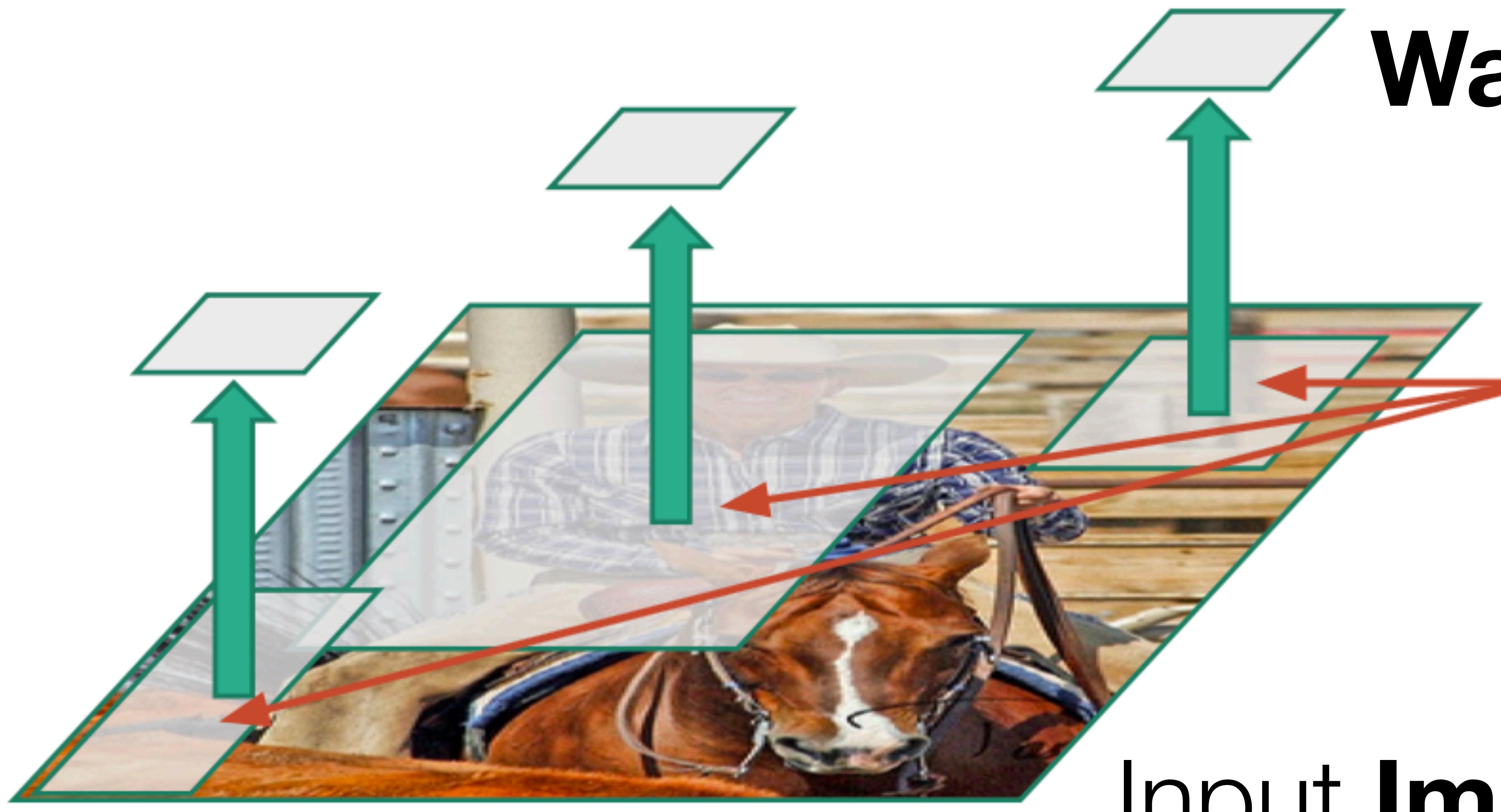
Regions of Interest from
a proposal method (~2k)

Input **Image**

* image from Ross Girshick

R-CNN

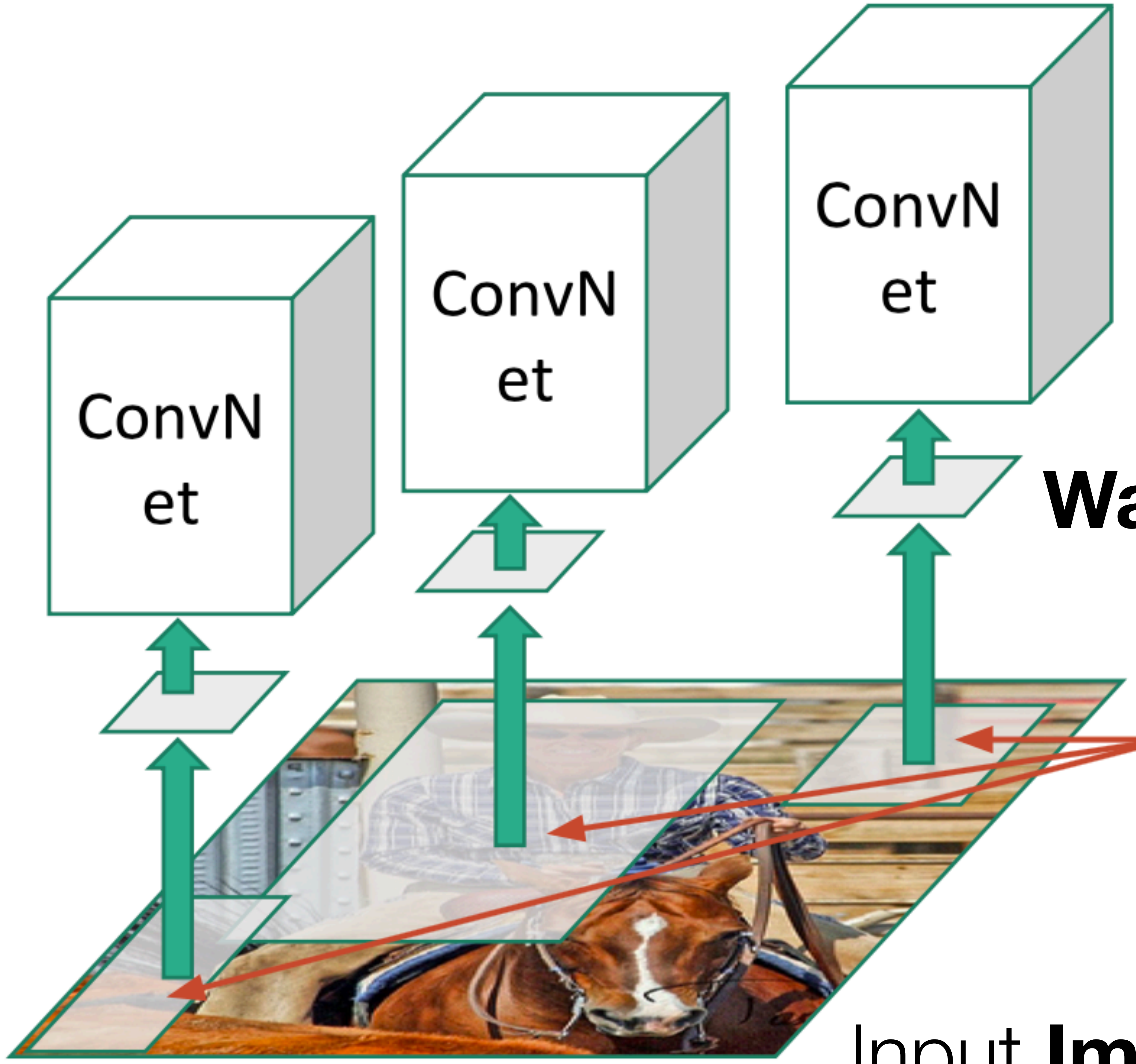
[Girshick et al, CVPR 2014]



* image from Ross Girshick

R-CNN

[Girshick et al, CVPR 2014]



Forward each region through a **CNN**

Warped image regions

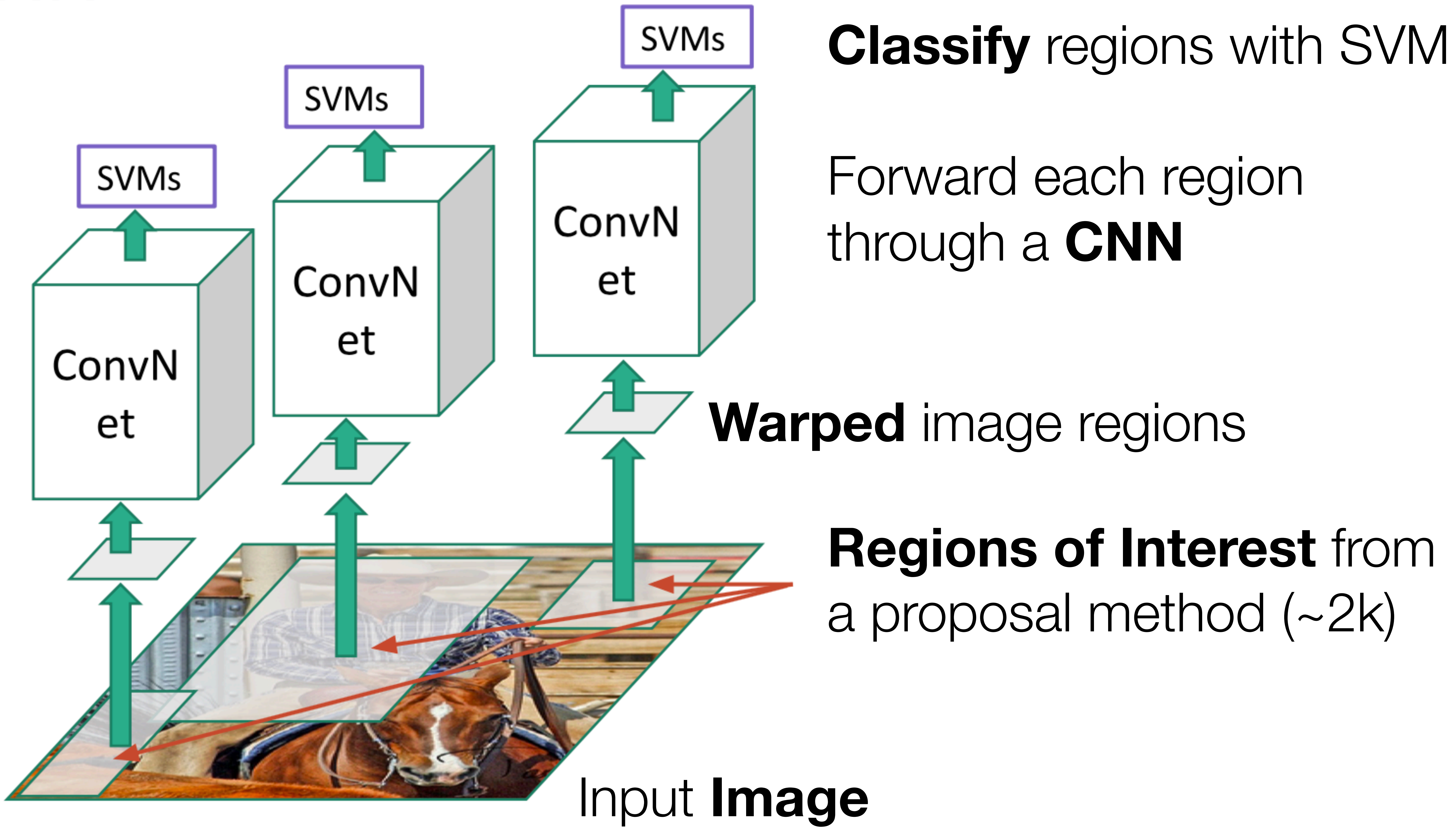
Regions of Interest from a proposal method (~2k)

Input **Image**

* image from Ross Girshick

R-CNN

[Girshick et al, CVPR 2014]



* image from Ross Girshick

R-CNN

Linear Regression for bounding box offsets

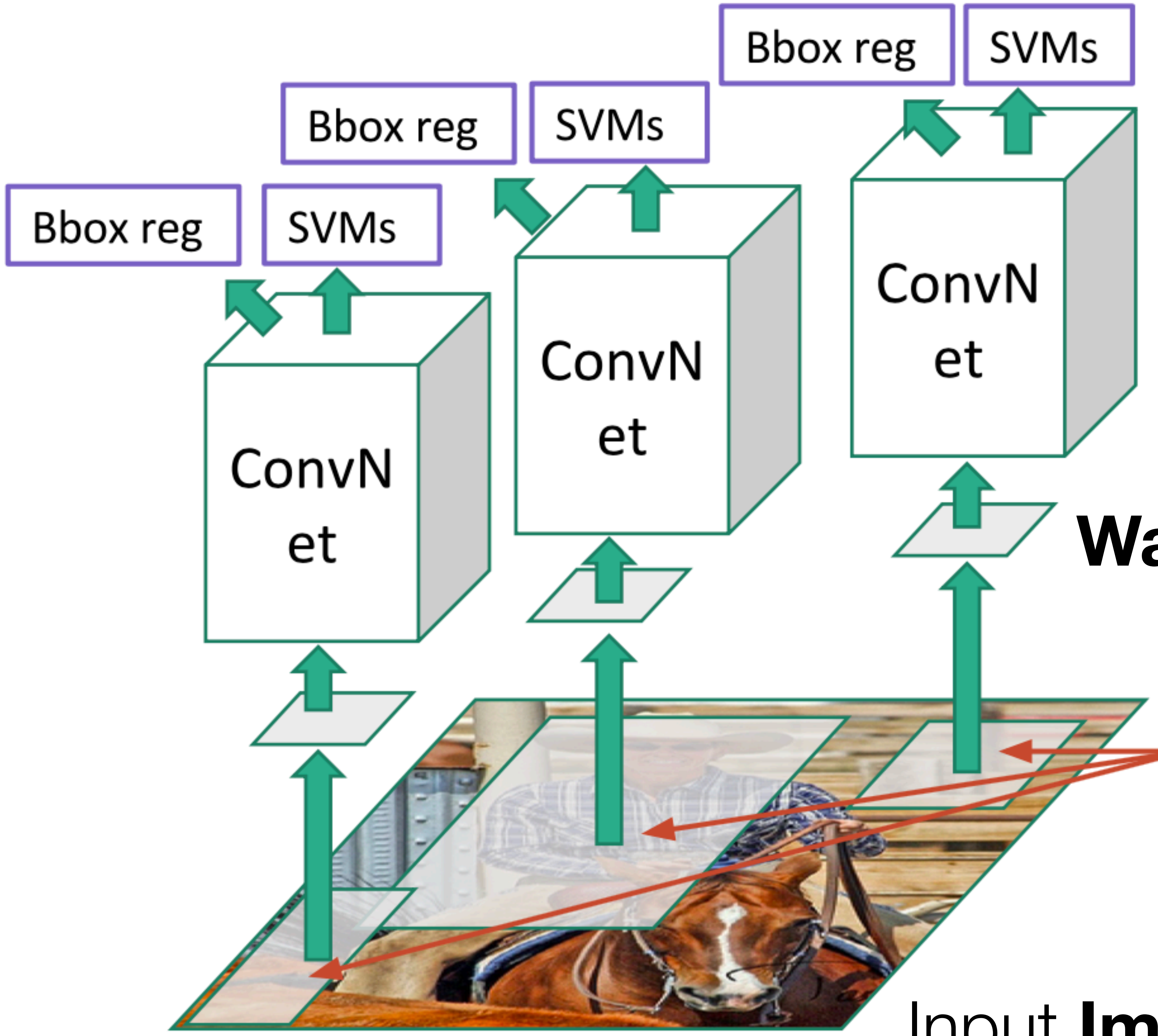
[Girshick et al, CVPR 2014]

Classify regions with SVM

Forward each region through a **CNN**

Warped image regions

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



Input **Image**

* image from Ross Girshick

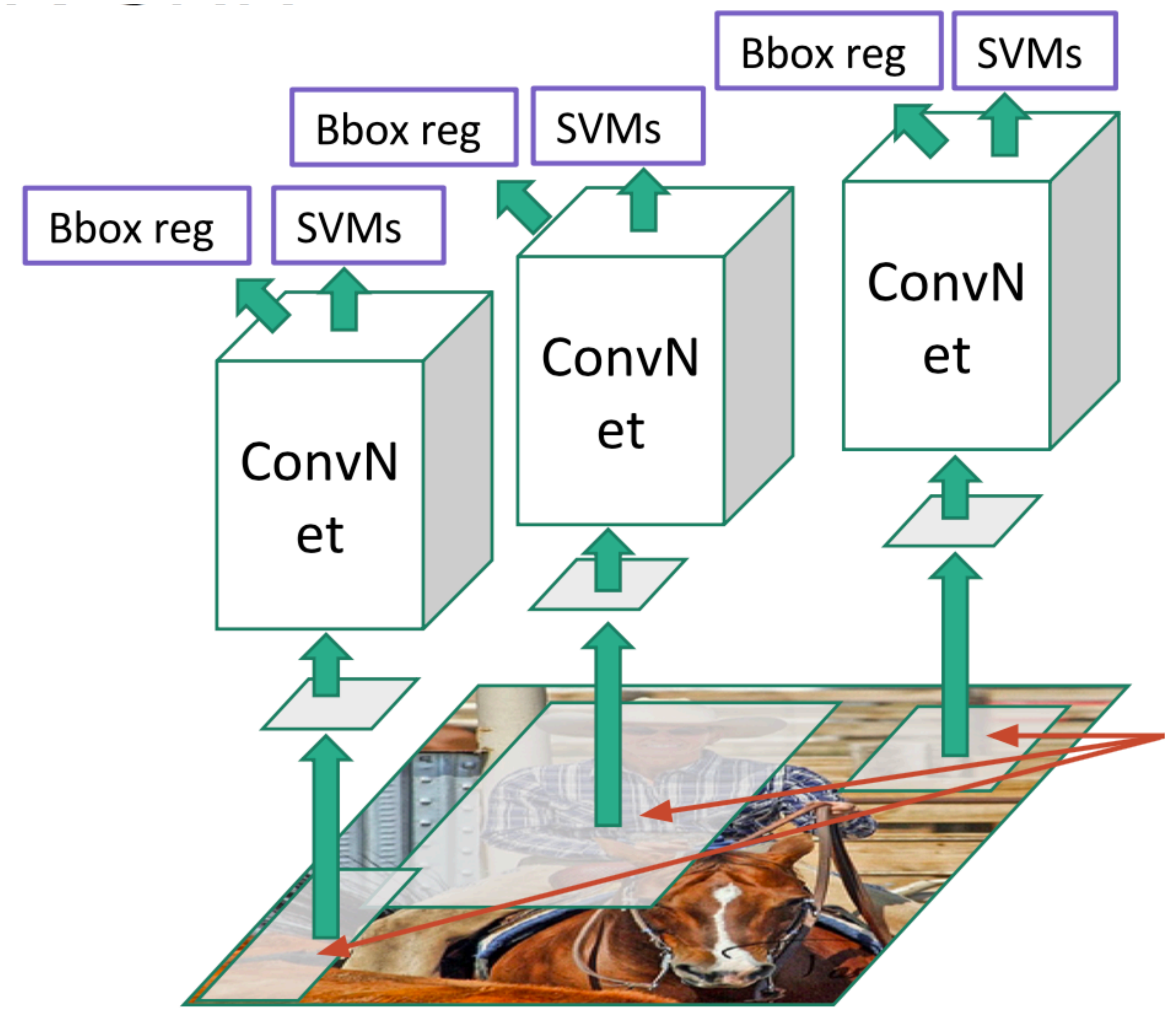
R-CNN: Training

[Girshick et al, CVPR 2014]

Fine-tuning ImageNet CNN on object proposal patches

— > 50% Intersection-over-Union overlap with GT considered "object" others "background"

— batches of 128 (**32 positives, 96 negatives**)



* image from Ross Girshick

R-CNN: Issues

[Girshick et al, CVPR 2014]

Ad-hoc training objectives

— Fine-tune network with softmax objective (**log** loss)

— Train post-hoc linear SVM (**hinge** loss)

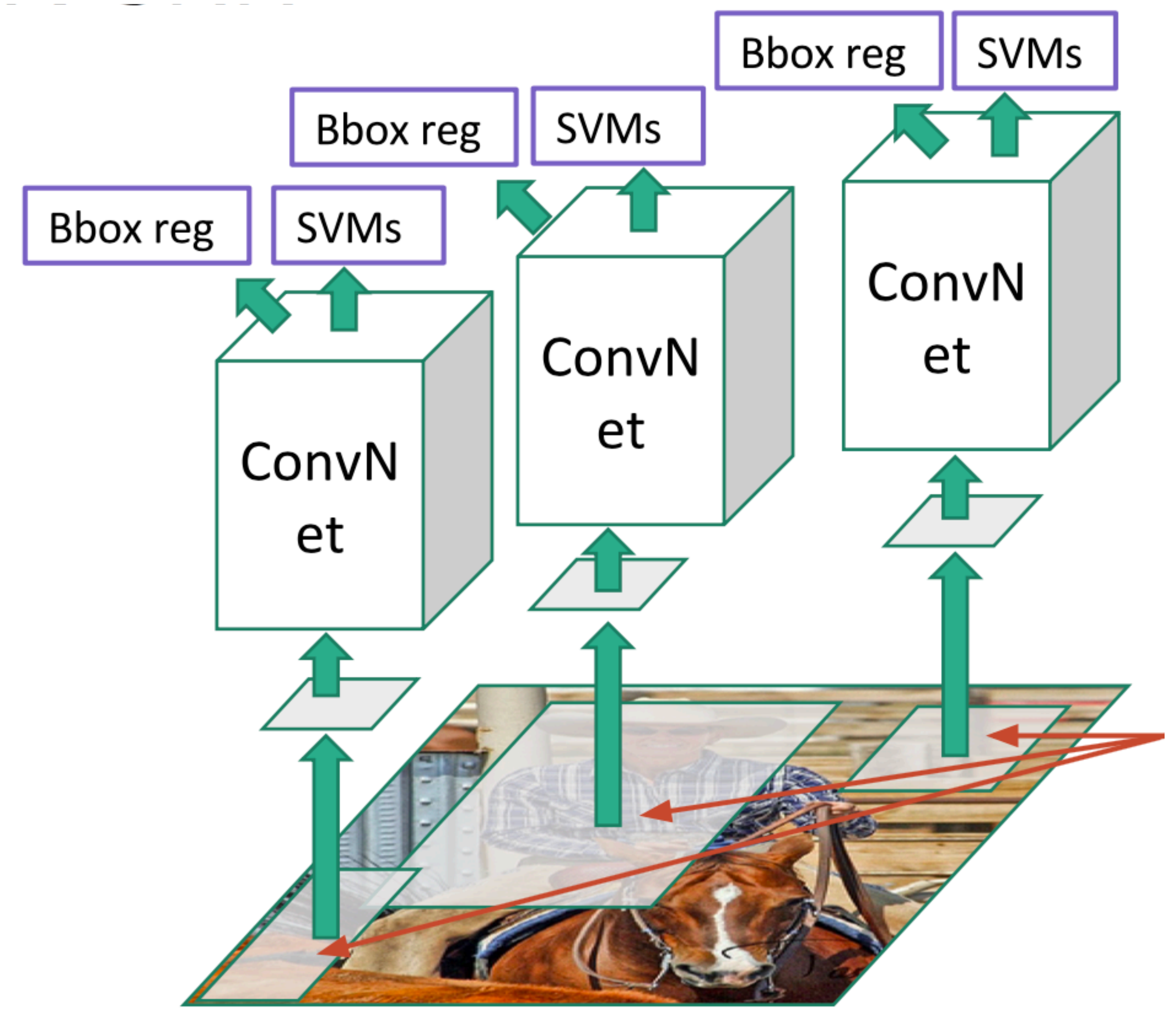
— Train post-hoc bounding-box regression (**least squares**)

Training is slow

— 84 hours and takes a lot of disk space

Inference / **Detection is slow**

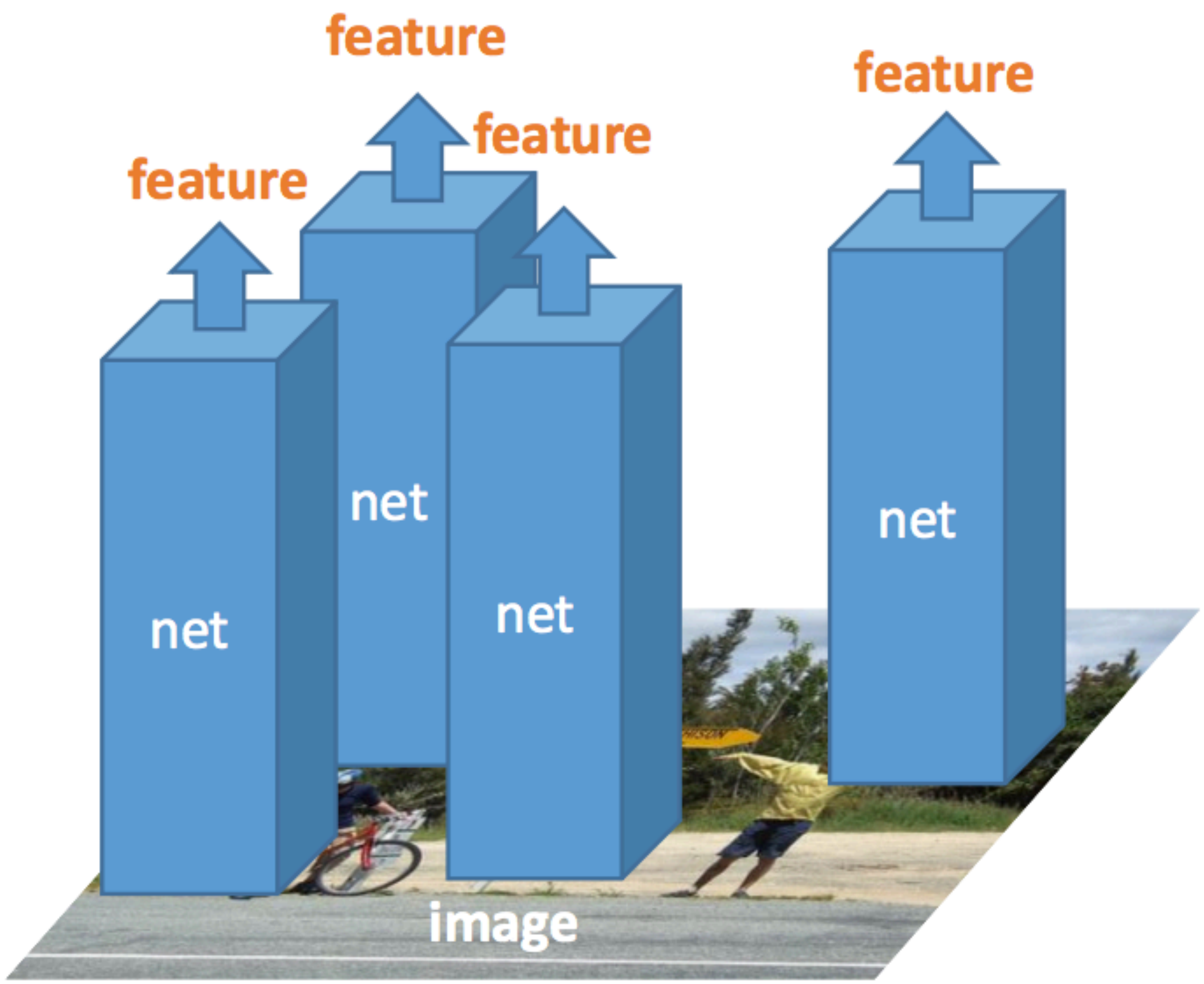
— 47 sec / image with VGG16 [Simonyan et al, ICLR 2015]



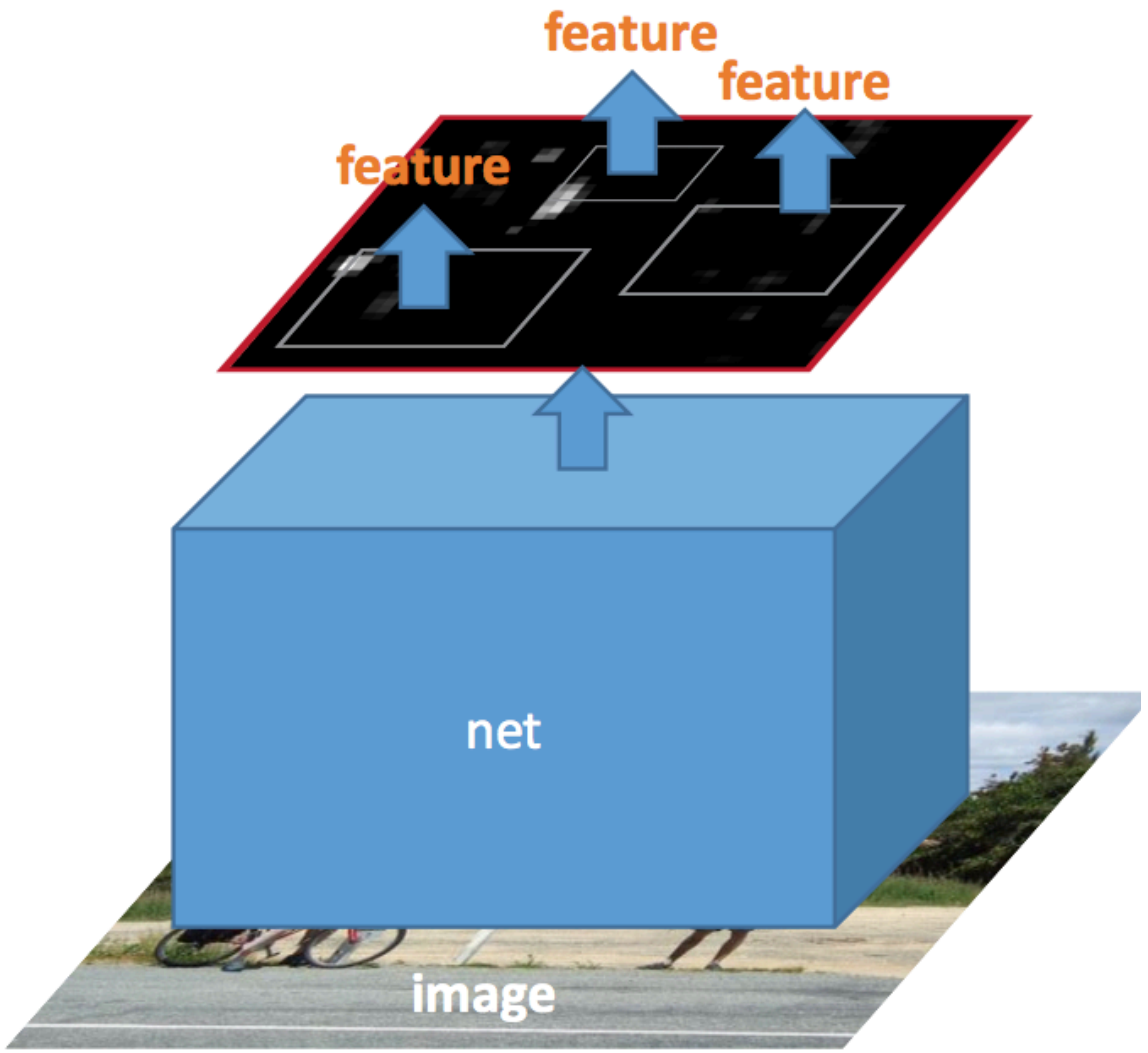
* image from Ross Girshick

R-CNN vs. SPP

[He et al, ECCV 2014]



R-CNN
2000 nets on image regions



SPP-net
1 net on full image

Fast **R-CNN**

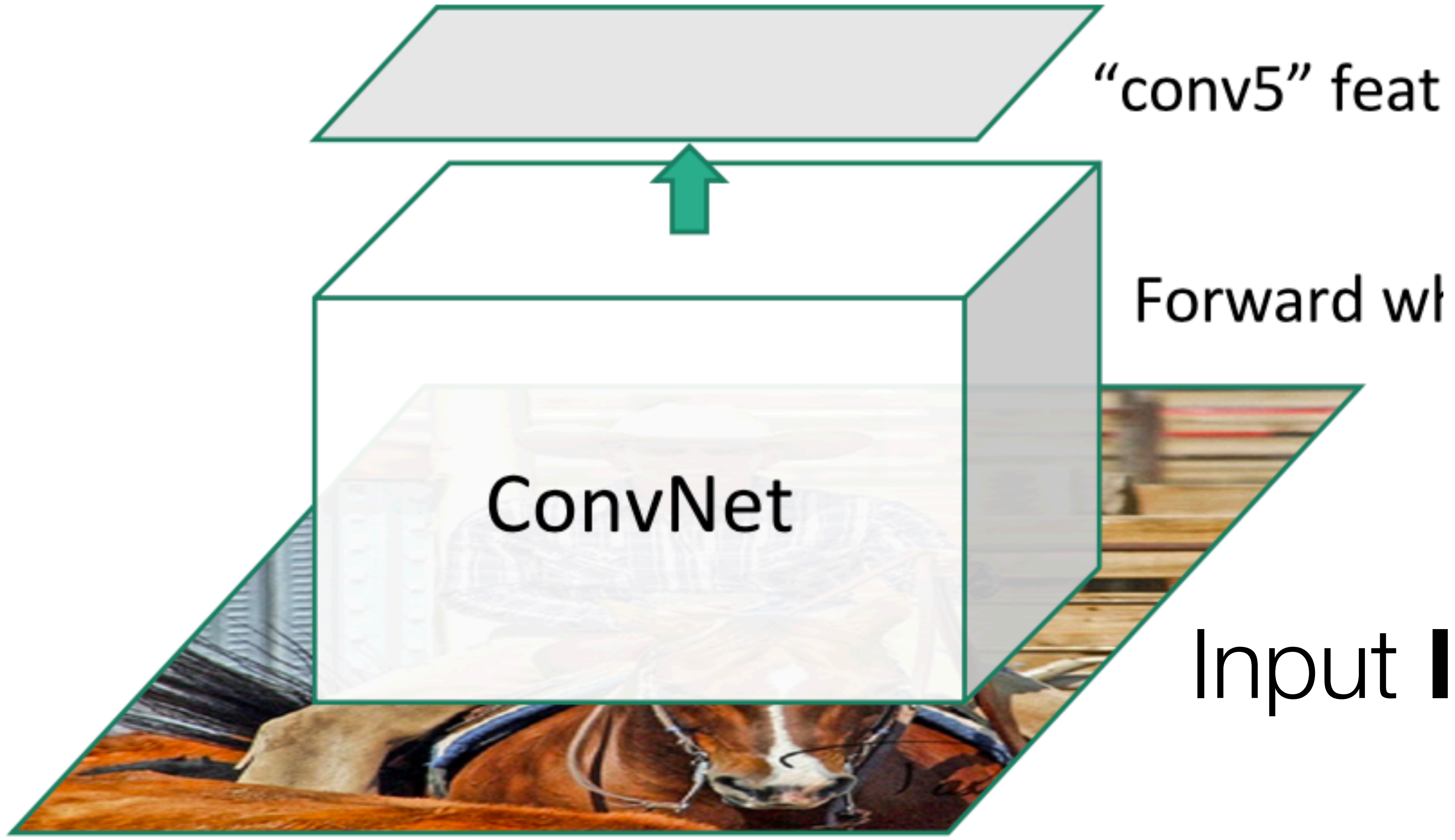
[Girshick et al, ICCV 2015]

Input **Image**

* image from Ross Girshick

Fast **R-CNN**

[Girshick et al, ICCV 2015]

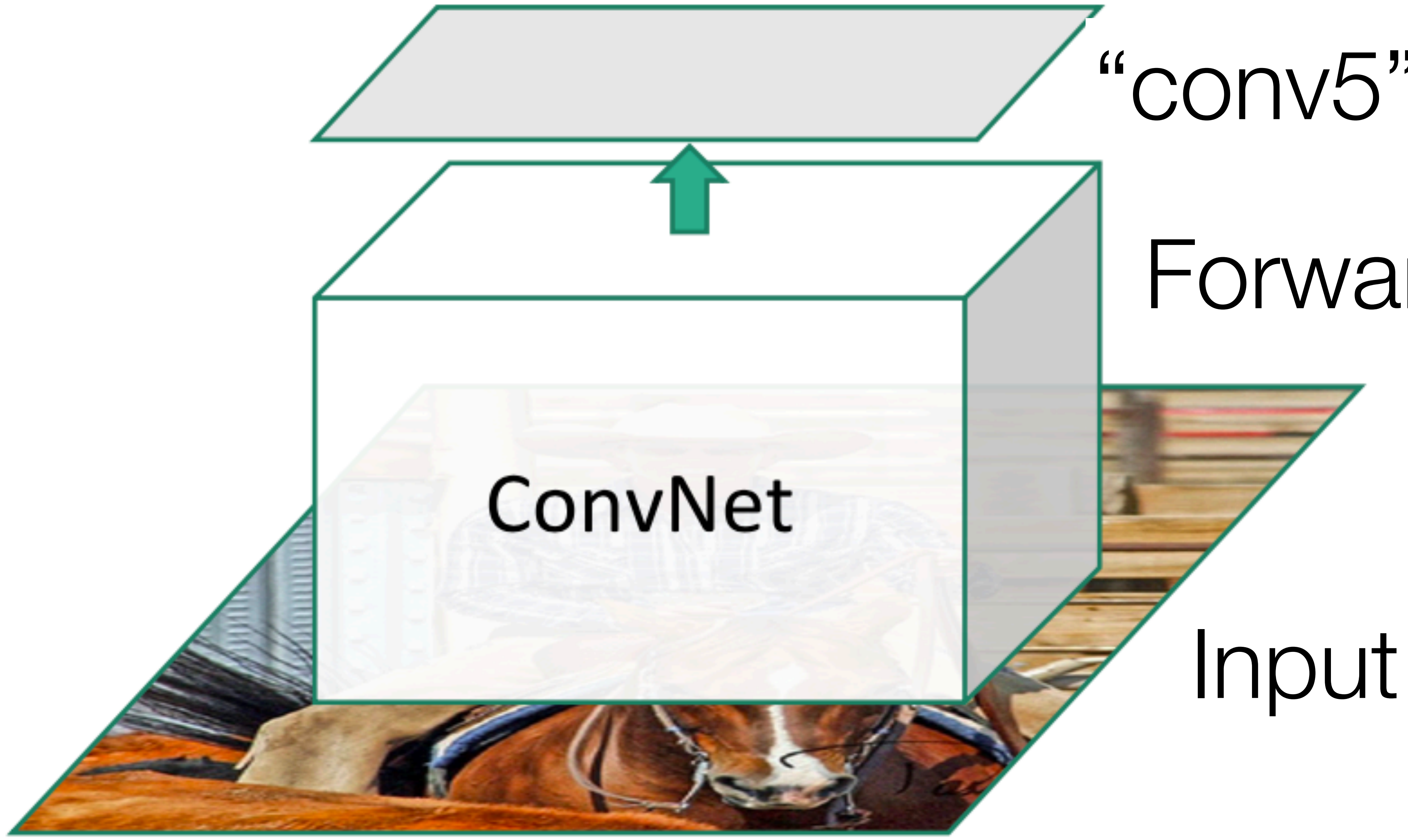


Input **Image**

* image from Ross Girshick

Fast **R-CNN**

[Girshick et al, ICCV 2015]



“conv5” feature map

Forward prop the **whole image** through CNN

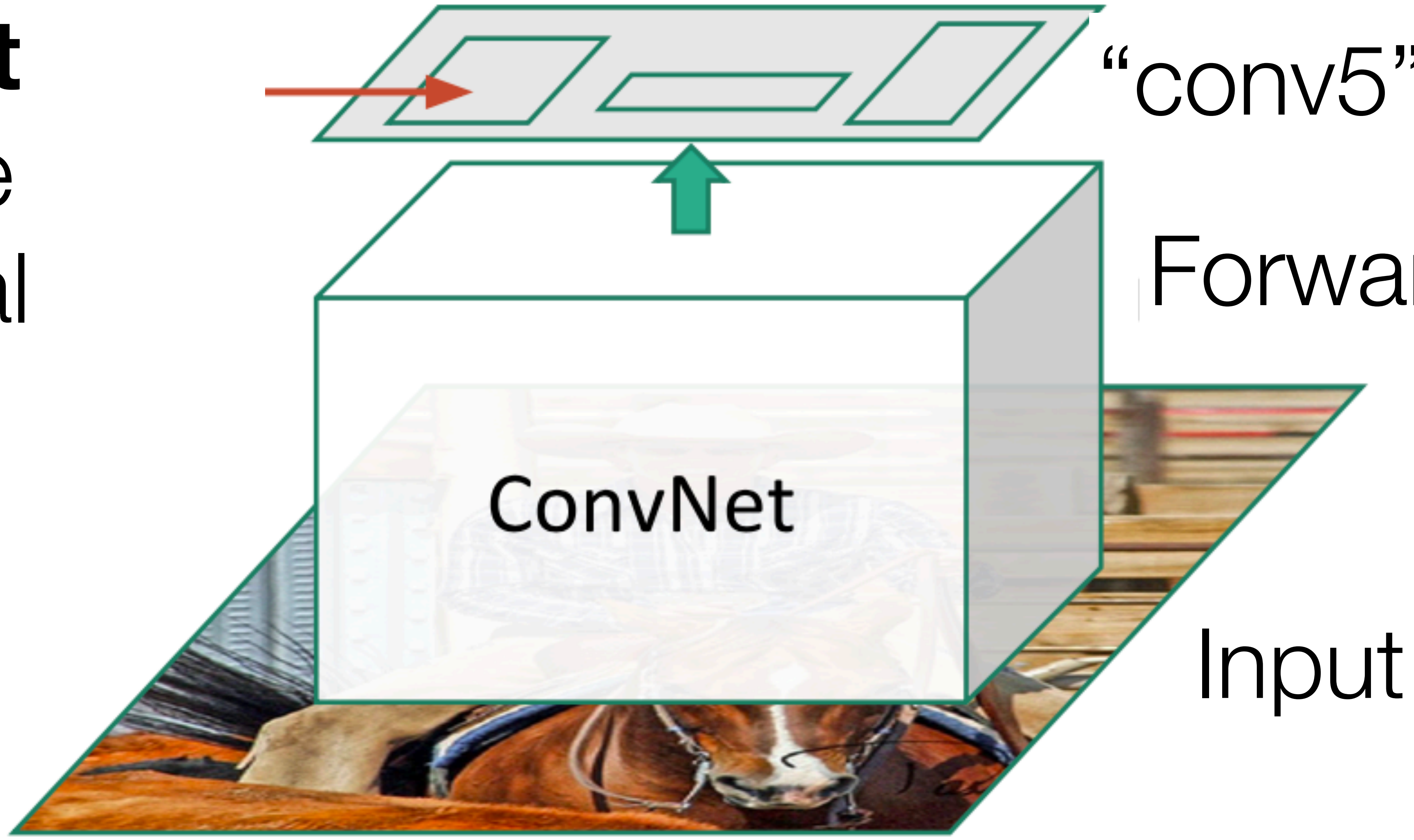
Input **Image**

* image from Ross Girshick

Fast **R-CNN**

[Girshick et al, ICCV 2015]

Regions of Interest from the proposal method



“conv5” feature map

Forward prop the **whole image** through CNN

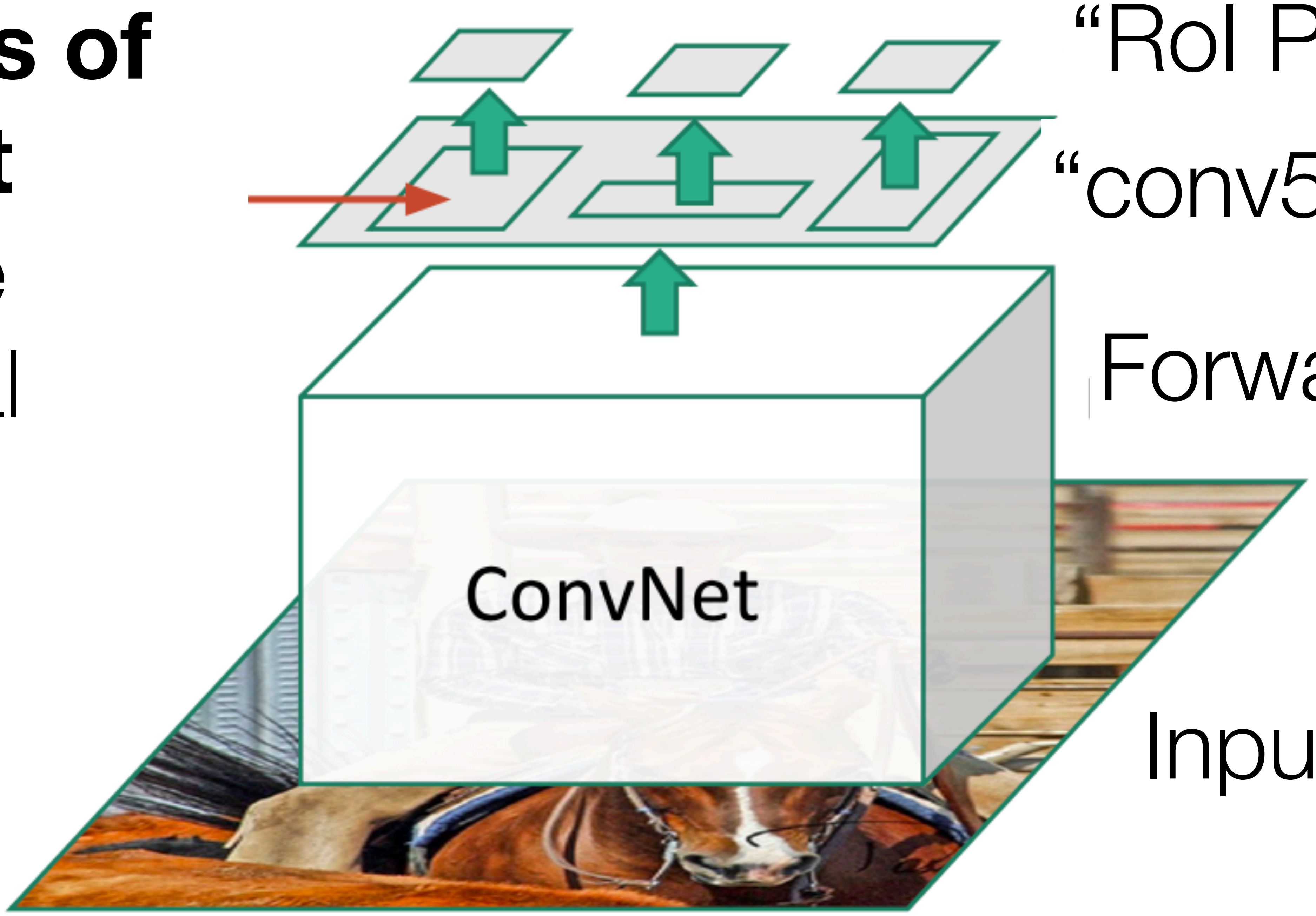
Input **Image**

* image from Ross Girshick

Fast **R-CNN**

[Girshick et al, ICCV 2015]

Regions of Interest from the proposal method



"RoI Pooling" layer

"conv5" feature map

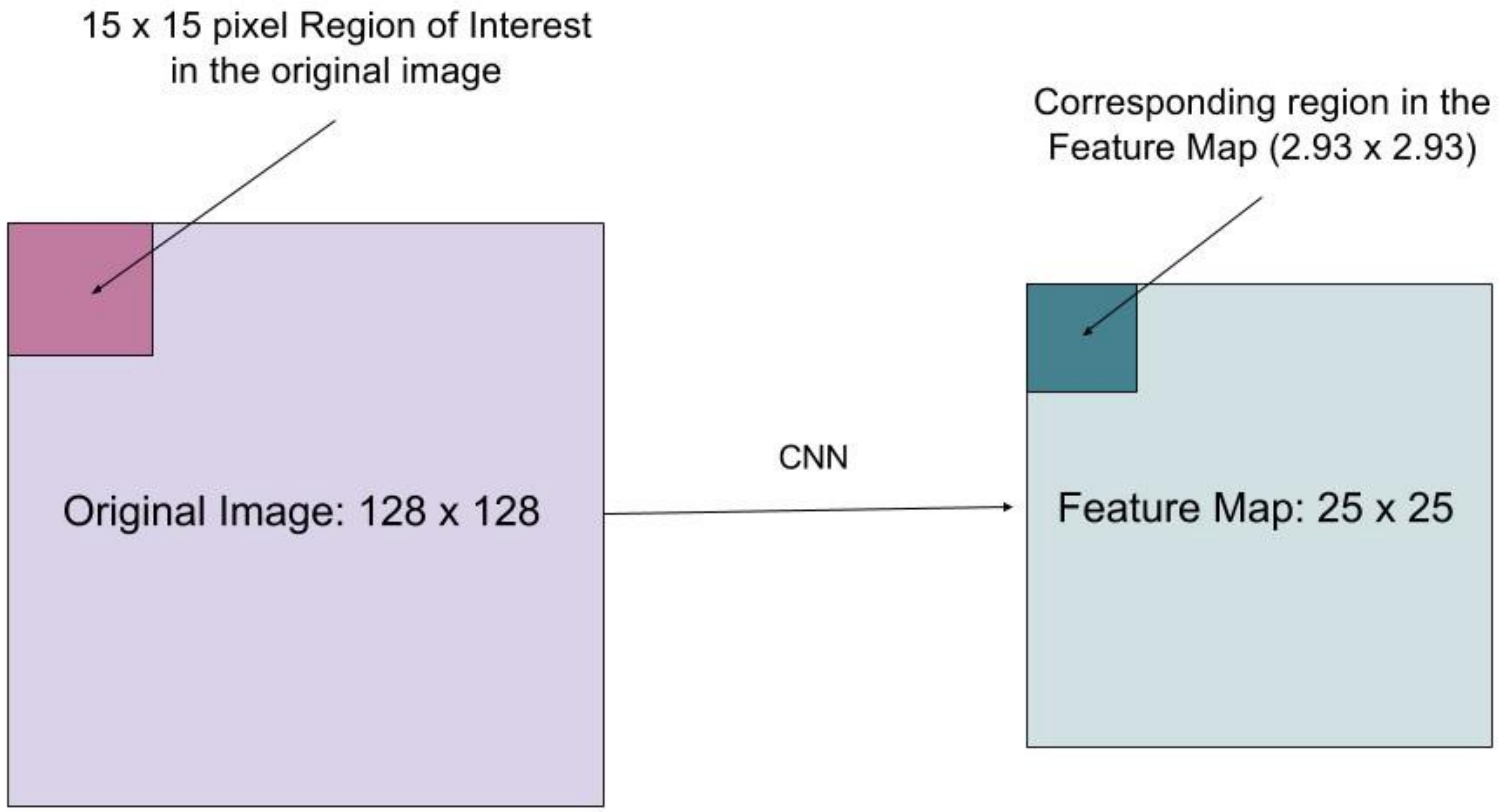
Forward prop the **whole image** through CNN

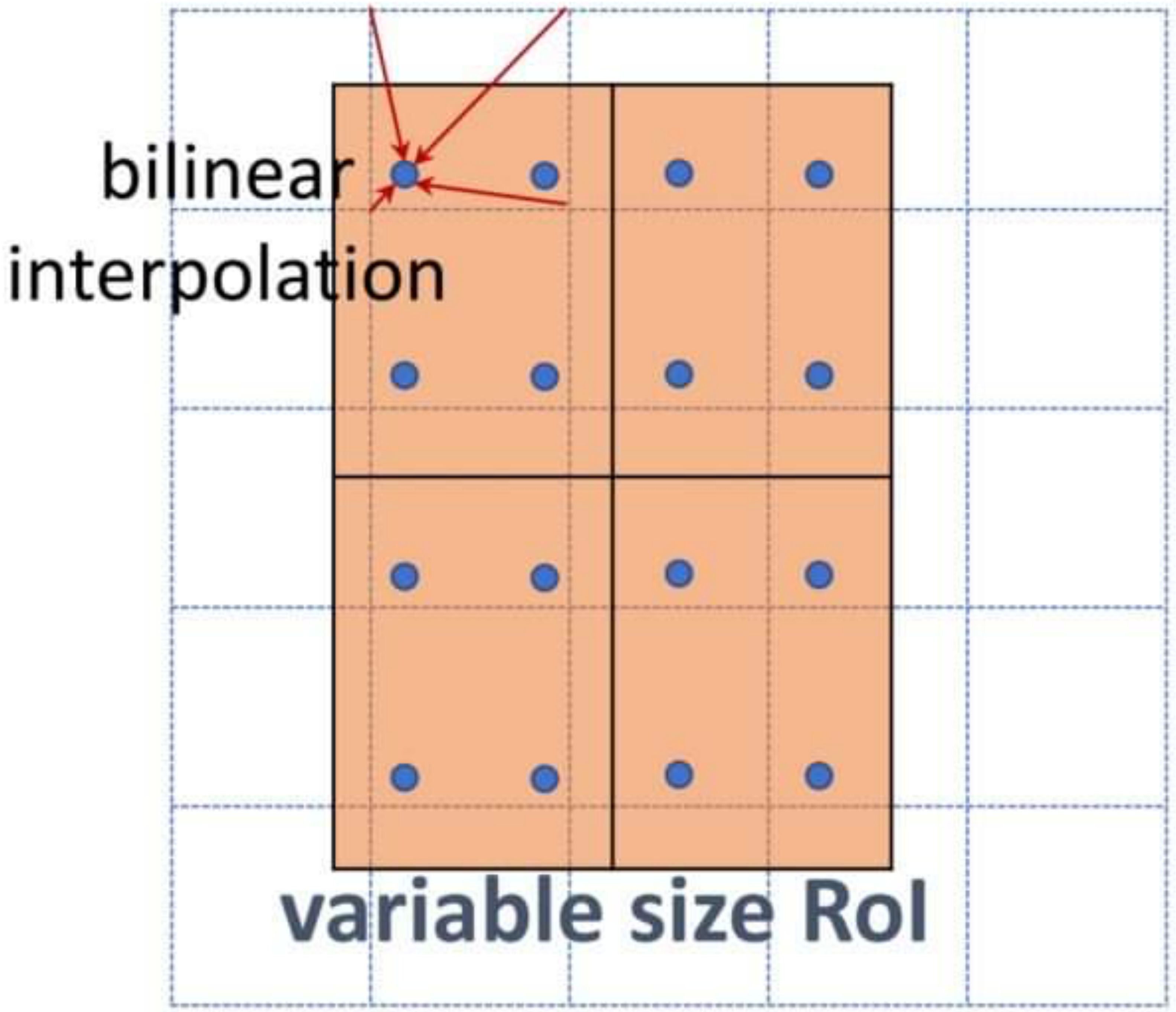
Input **Image**

Girshick, "Fast R-C
Figure copyright R

* image from Ross Girshick

RoI Align


15 x 15 pixel Region of Interest
in the original image
Original Image: 128 x 128
CNN
Corresponding region in the
Feature Map (2.93 x 2.93)
Feature Map: 25 x 25


bilinear
interpolation
variable size RoI

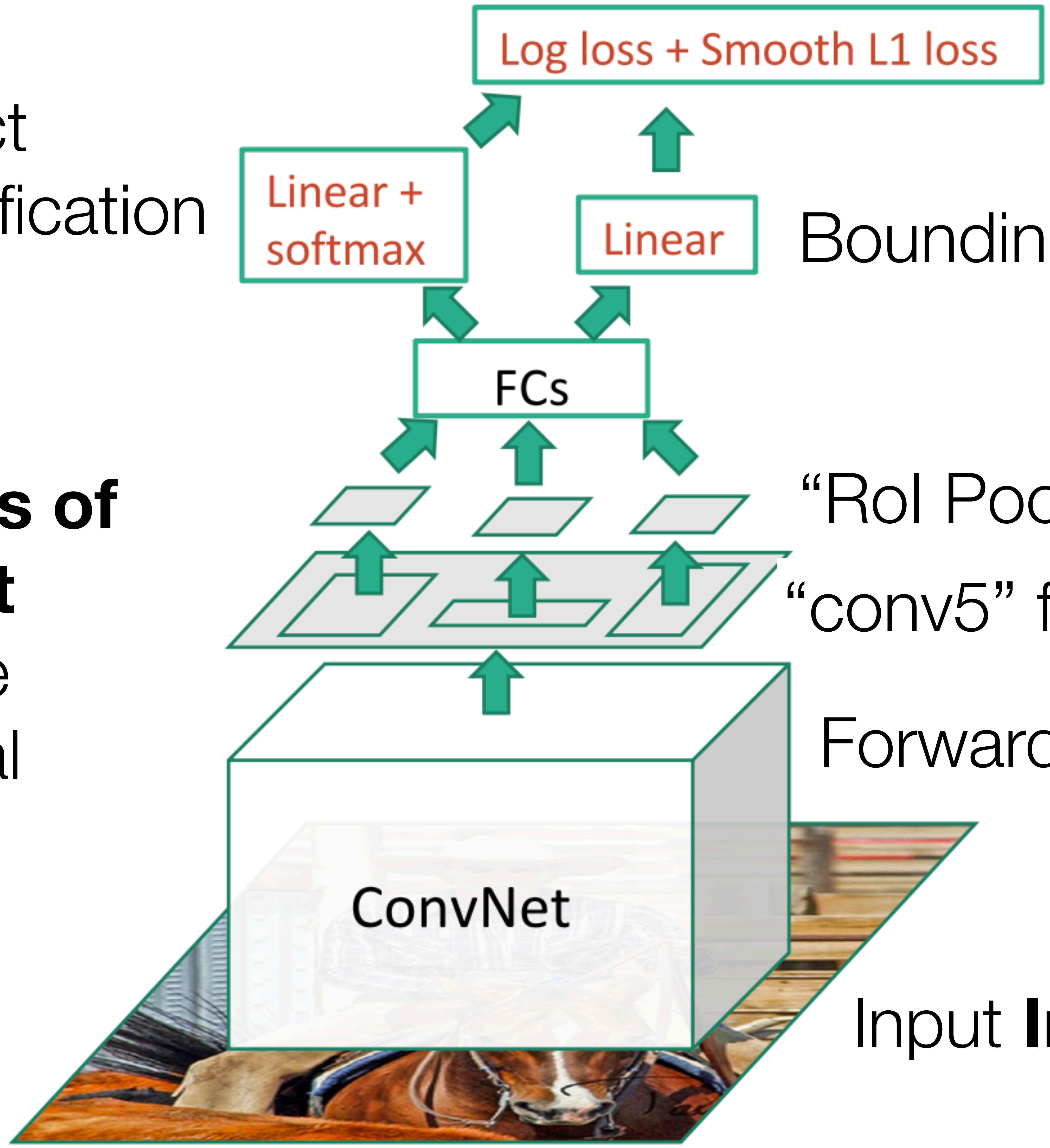
Fast **R-CNN**

[Girshick et al, ICCV 2015]

Multi-task loss

Object classification

Bounding box regression



“RoI Pooling” layer

“conv5” feature map

Regions of Interest from the proposal method

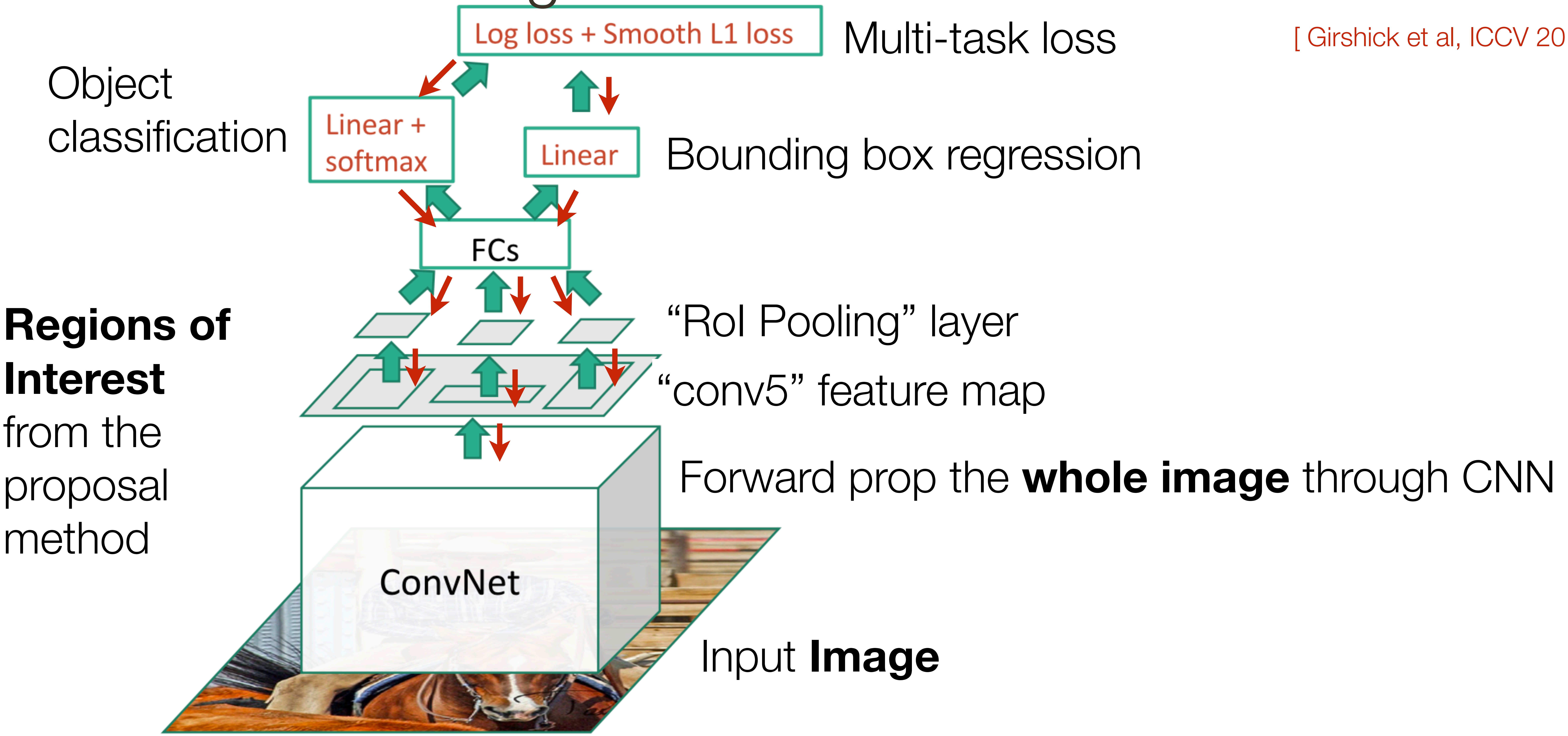
Forward prop the **whole image** through CNN

Input **Image**

* image from Ross Girshick

Fast **R-CNN:** Training

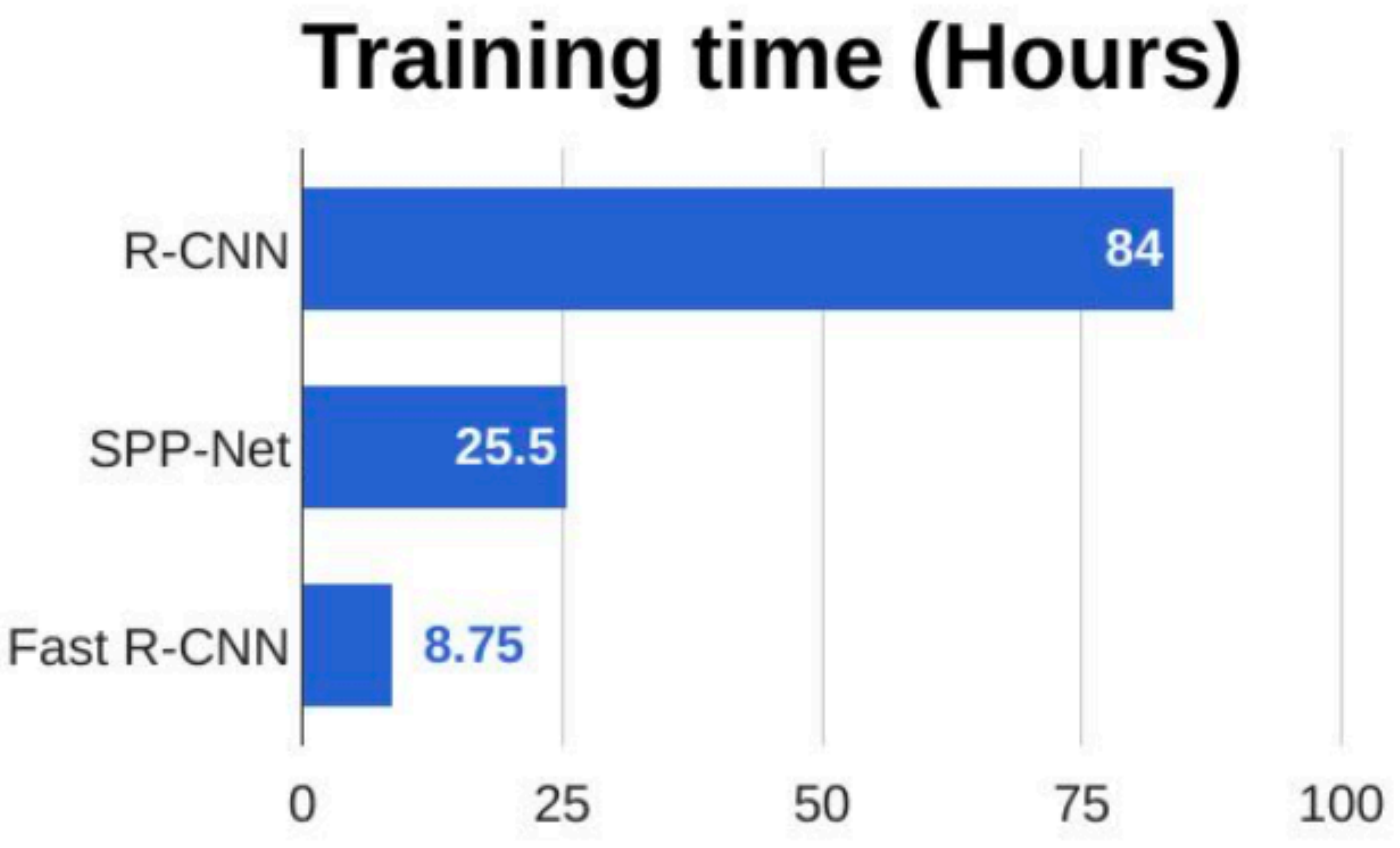
[Girshick et al, ICCV 2015]

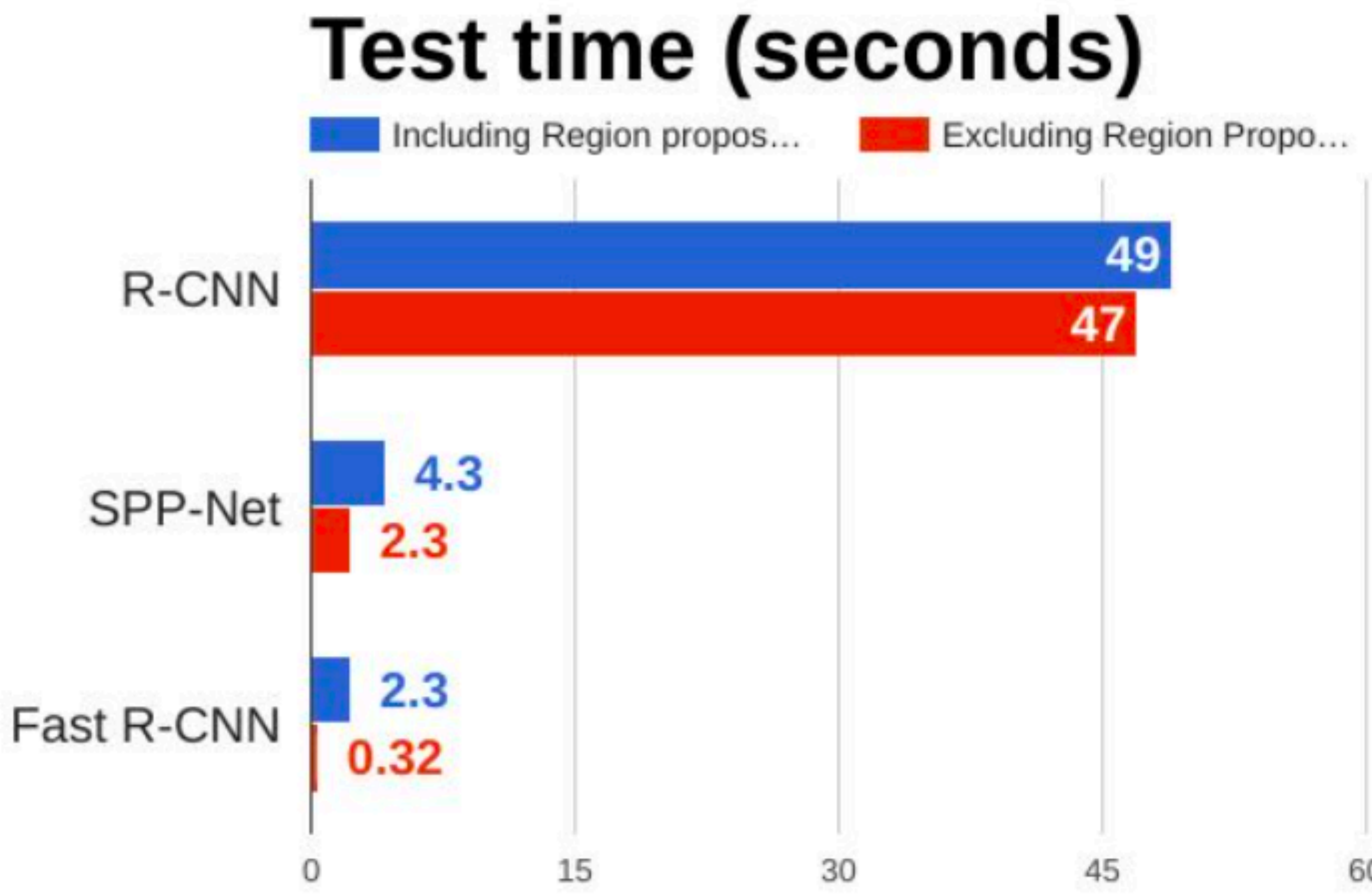


* image from Ross Girshick

R-CNN vs. **SPP** vs. **Fast R-CNN**

[Girshick et al, CVPR 2014]
[Girshick et al, ICCV 2015]
[He et al, ECCV 2014]

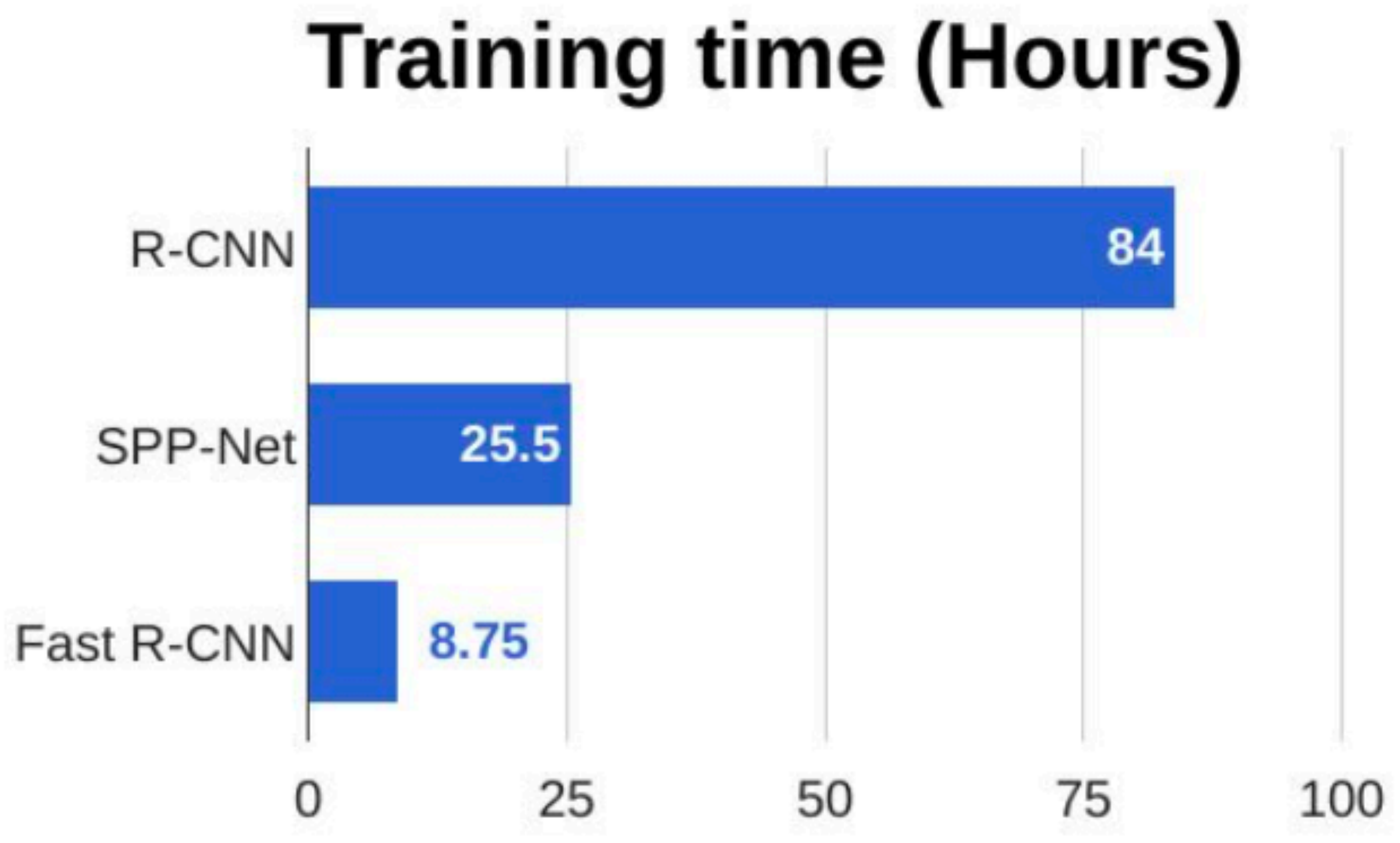


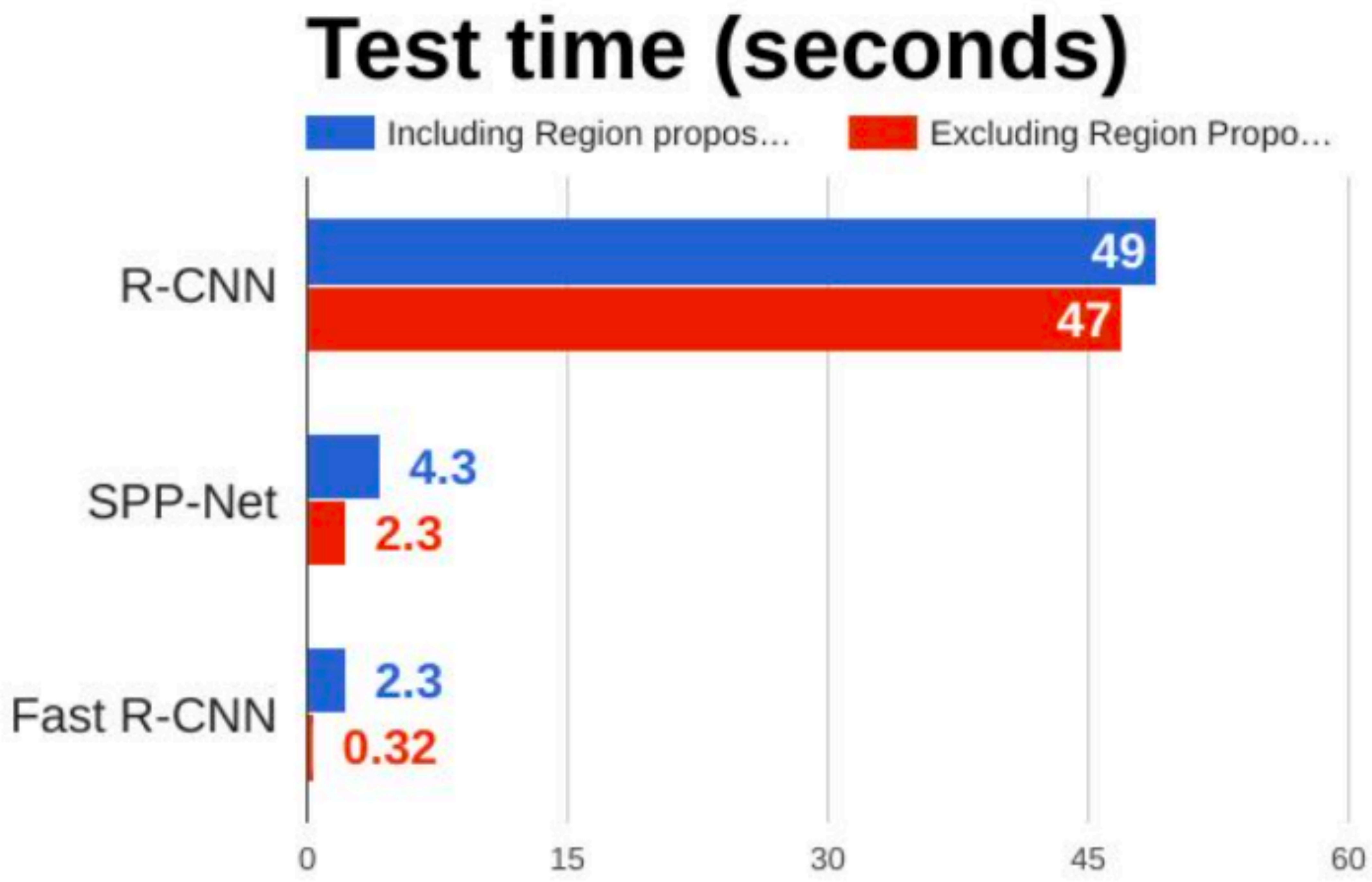


* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

R-CNN vs. SPP vs. Fast R-CNN

[Girshick et al, CVPR 2014]
[Girshick et al, ICCV 2015]
[He et al, ECCV 2014]





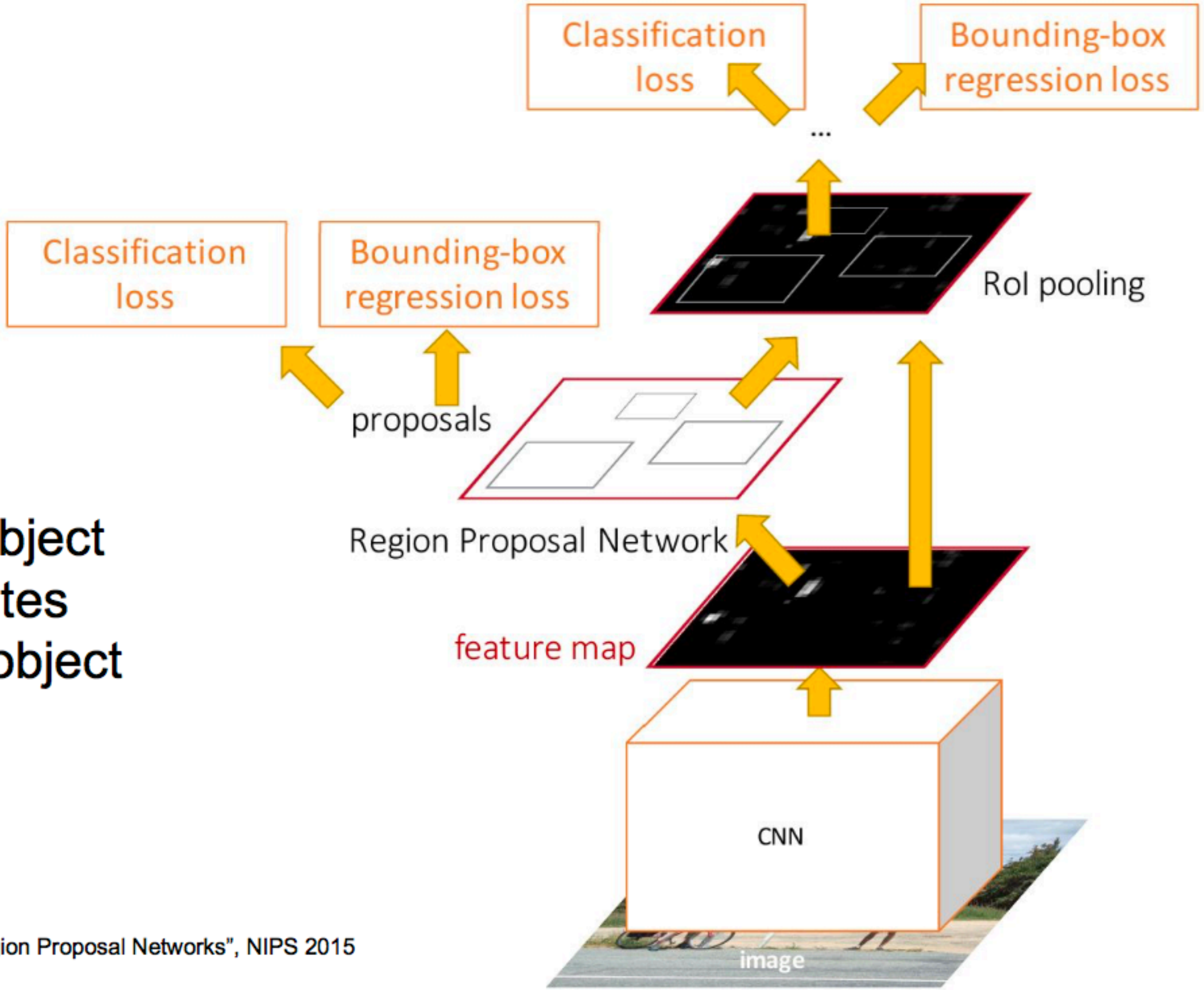
Observation: Performance dominated by the region proposals at this point!

Faster **R-CNN**

Make CNN do proposals!

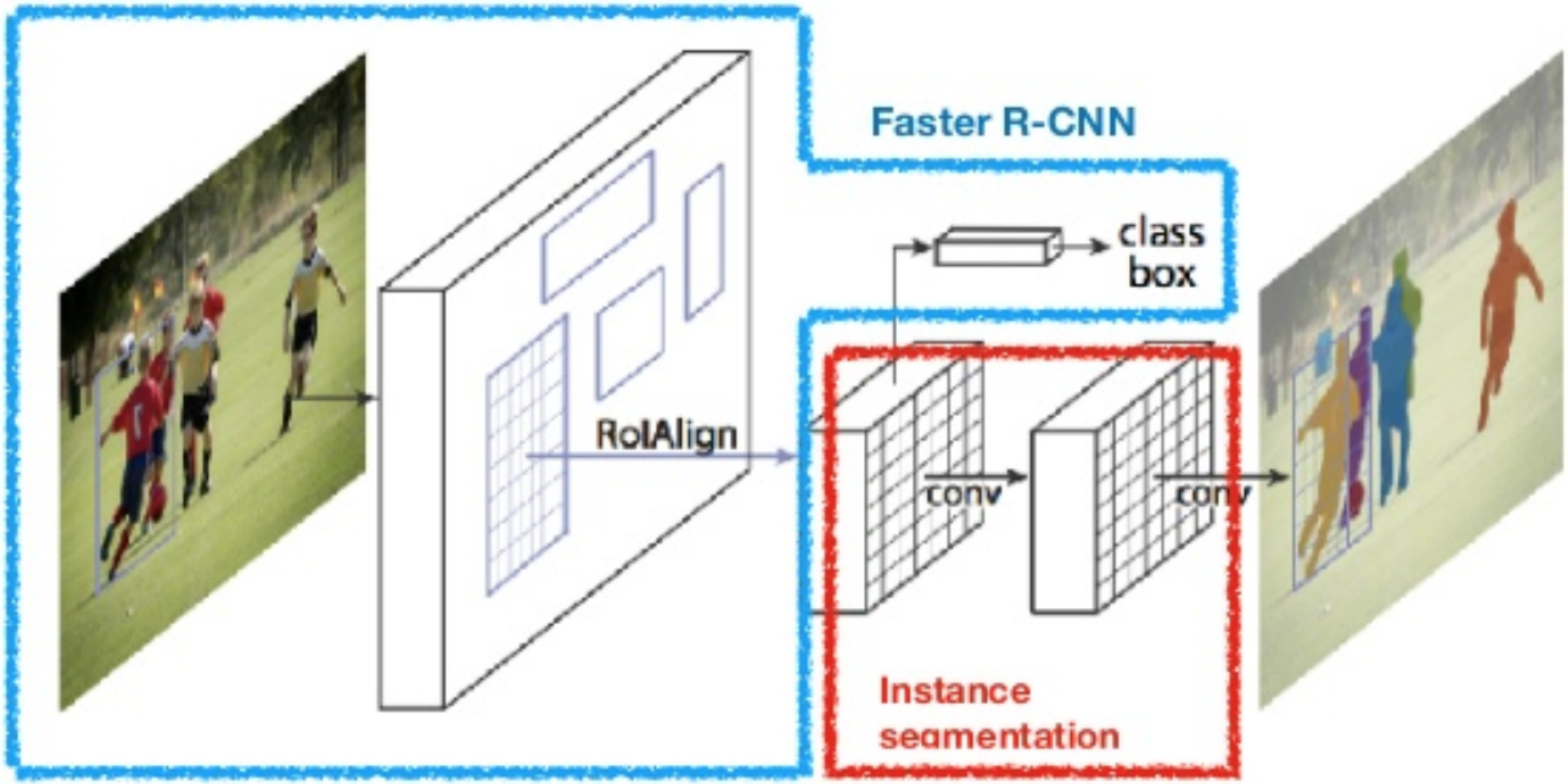
Insert **Region Proposal Network (RPN)** to predict proposals from features

Jointly train with 4 losses:
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Mask R-CNN

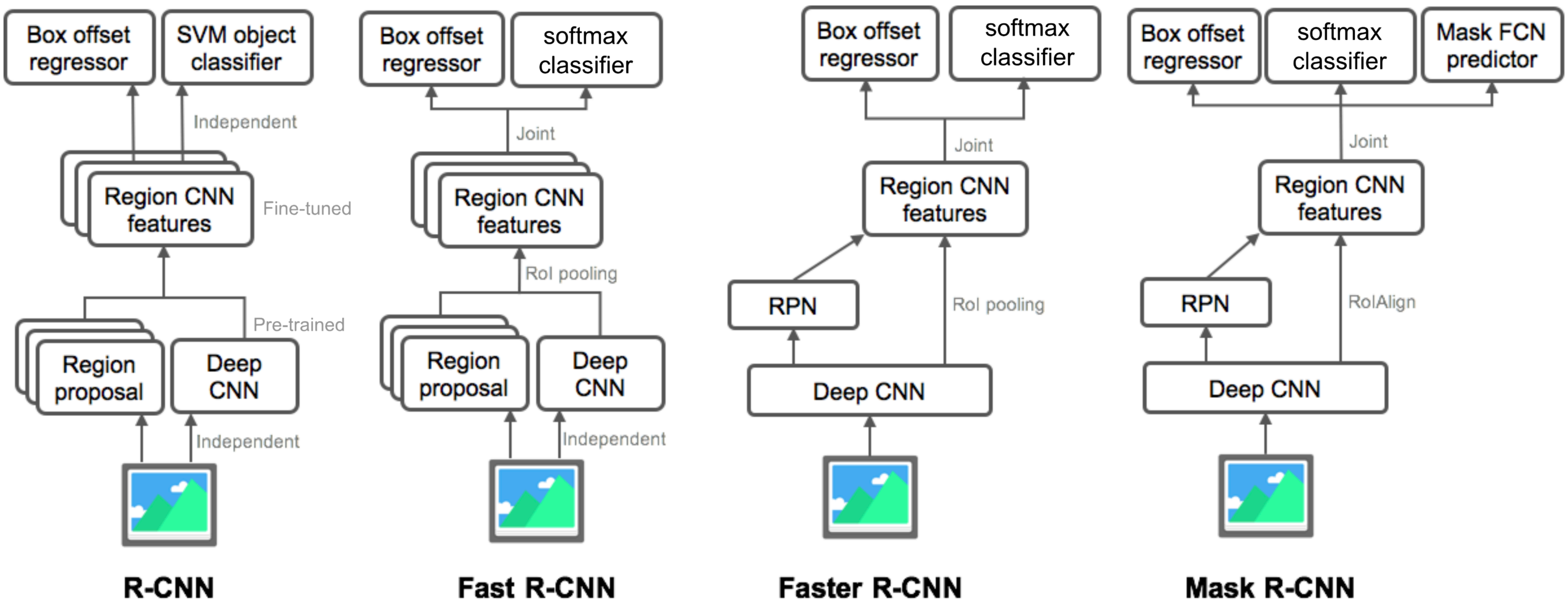


[He et al, 2017]

Mask R-CNN

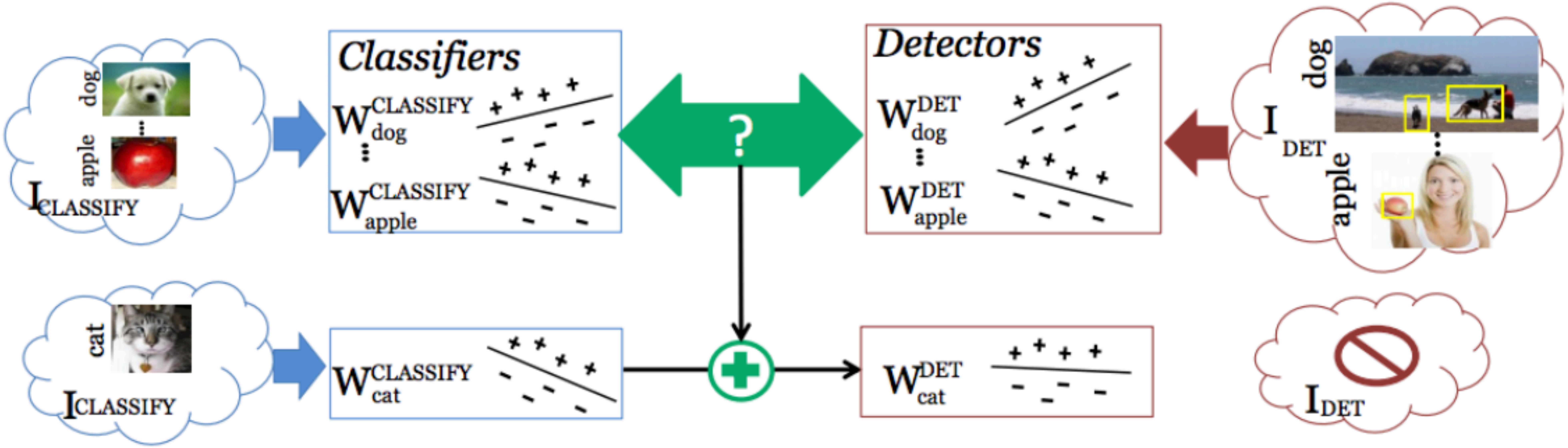
[He et al, 2017]

Summary of R-CNN Family of Models



https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html

LSDA: Large Scale Detection through Adaptation



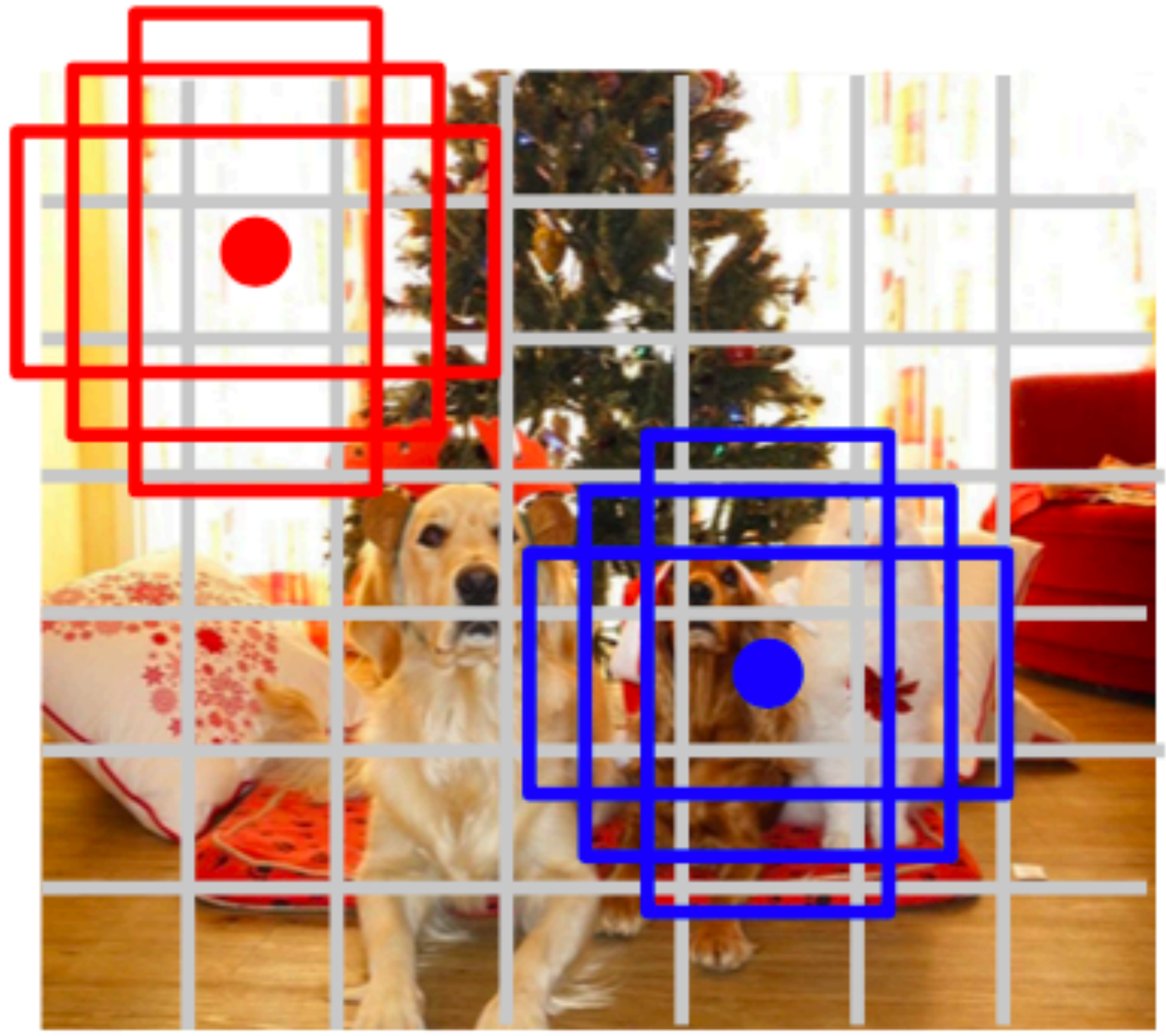
$$\mathbf{W}_{cat}^{DETECT} = \mathbf{W}_{cat}^{CLASIFY} + \delta \mathbf{W}_{cat}$$

[Hoffman et al, NIPS 2014]

YOLO: You Only Look Once

[Redmon et al, CVPR 2016]

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

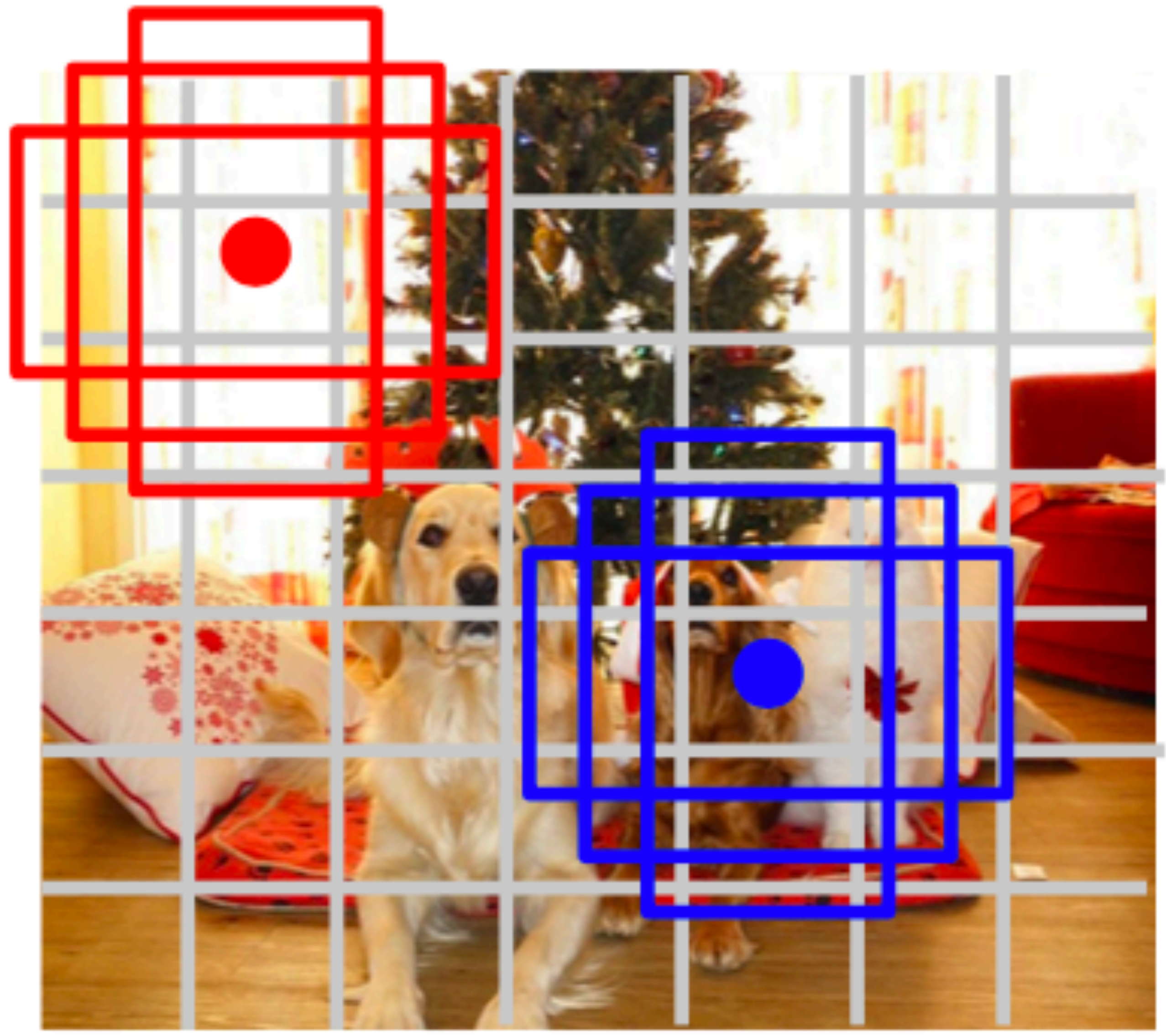
Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers:
  (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

YOLO: You Only Look Once

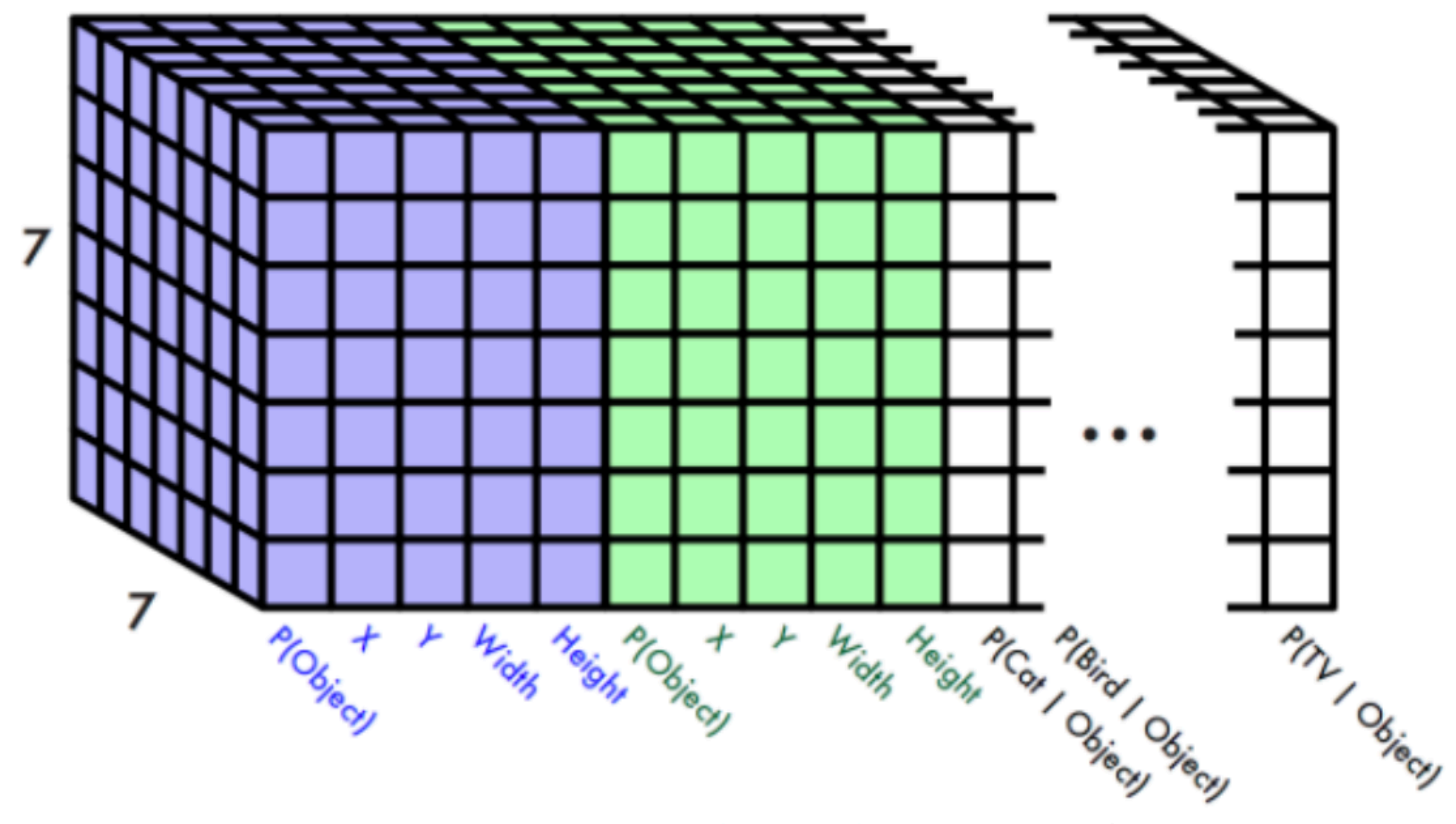
[Redmon et al, CVPR 2016]

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3



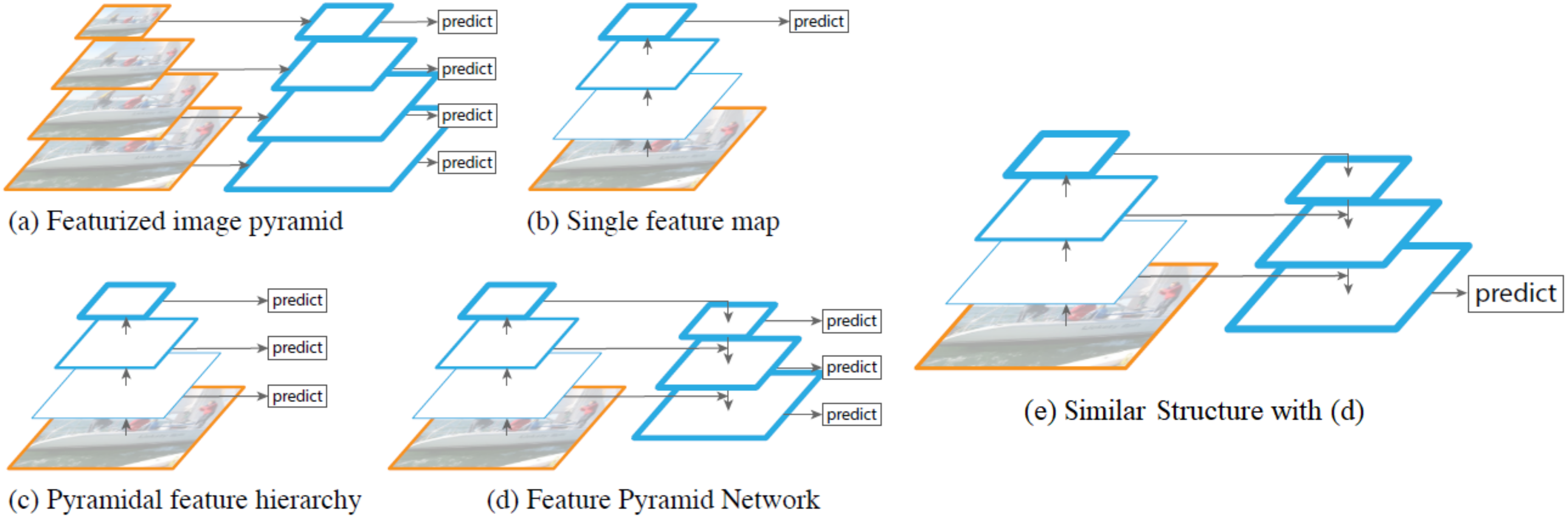
YOLO v2

http://pjreddie.com/yolo

YOLO v2

http://pjreddie.com/yolo

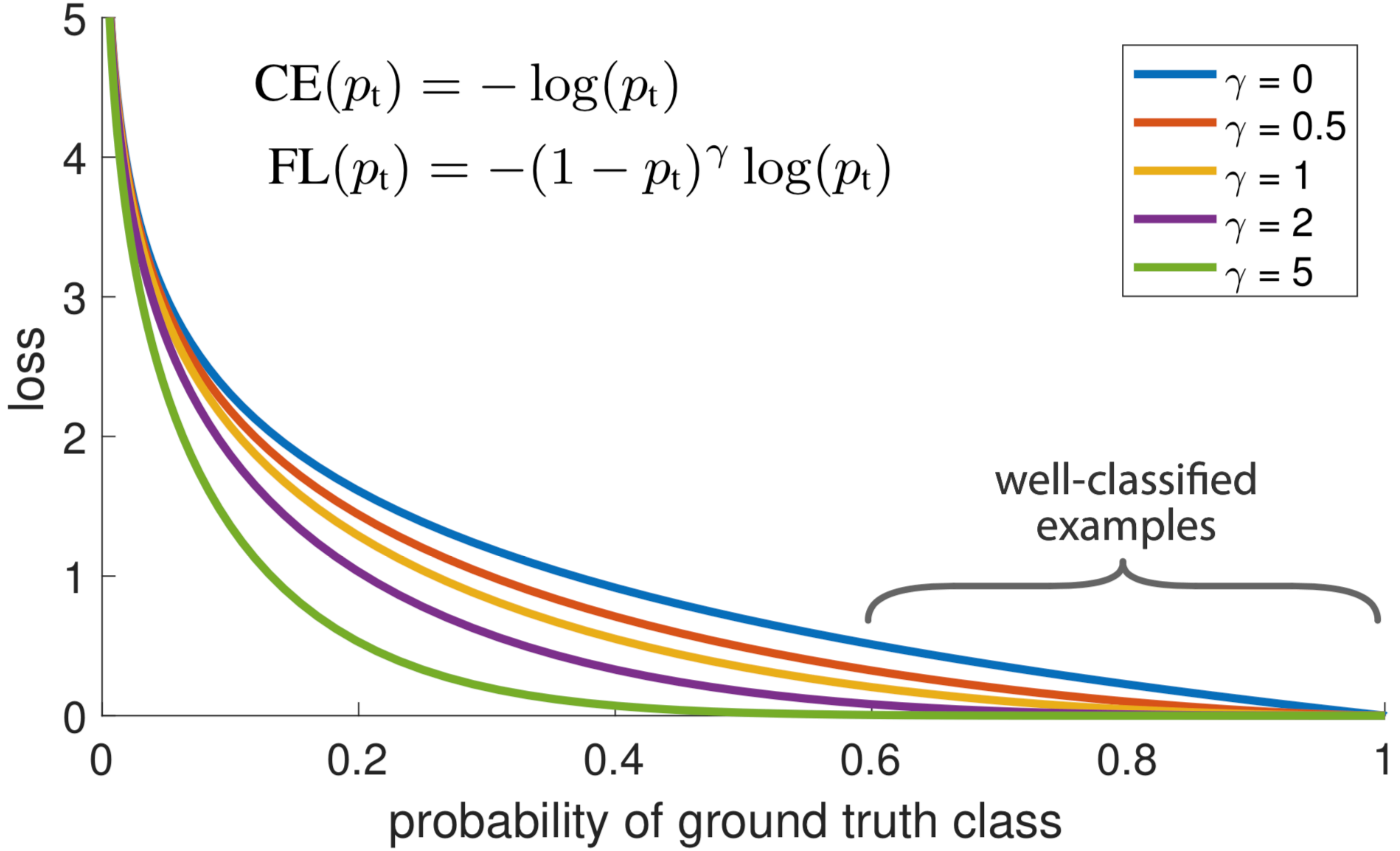
Feature **Pyramid** Networks



[Lin et al, CVPR 2017]

Focal Loss

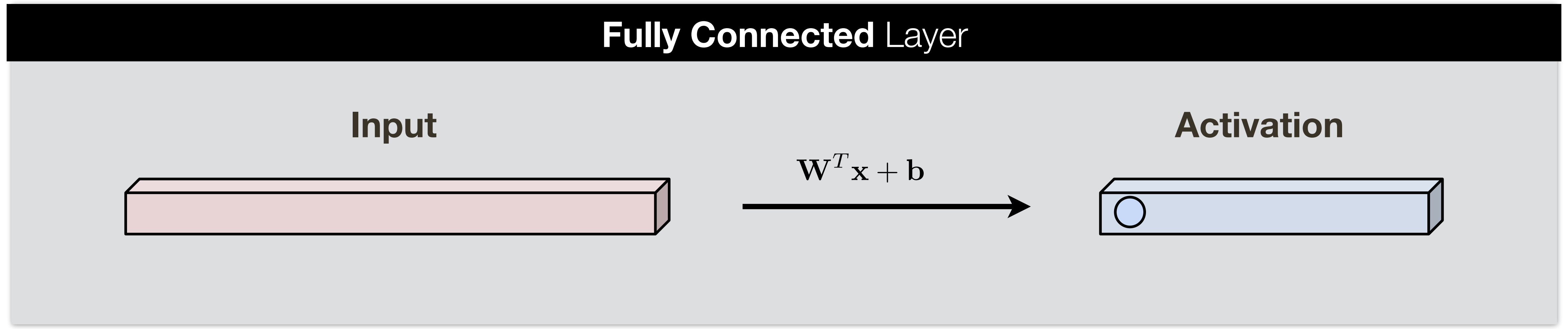
$$p_t = \begin{cases} p & \text{if } \; y = 1 \\ 1 - p & \text{otherwise} \end{cases}$$



[Lin et al, ICCV 2017]

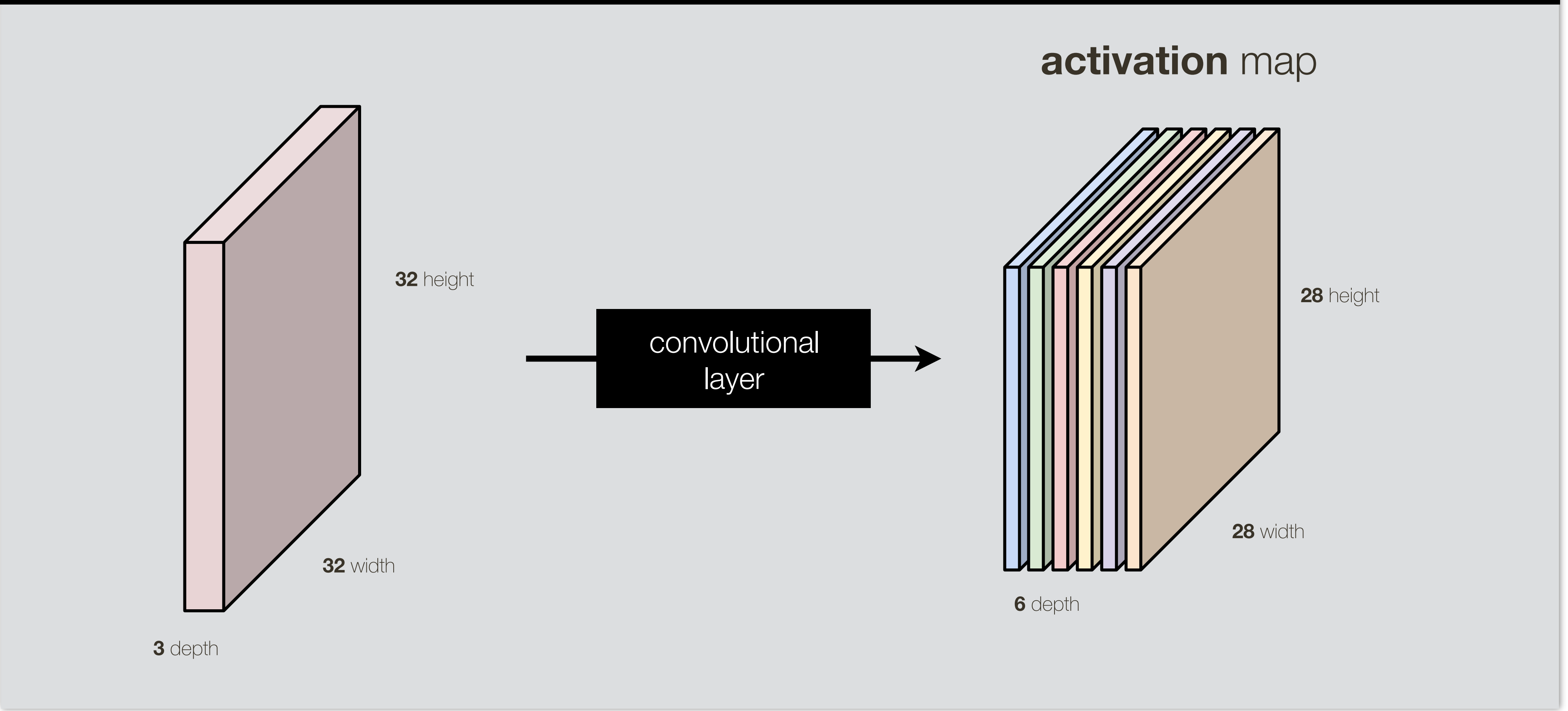
Review of **CNNs**

Review of **CNNs**

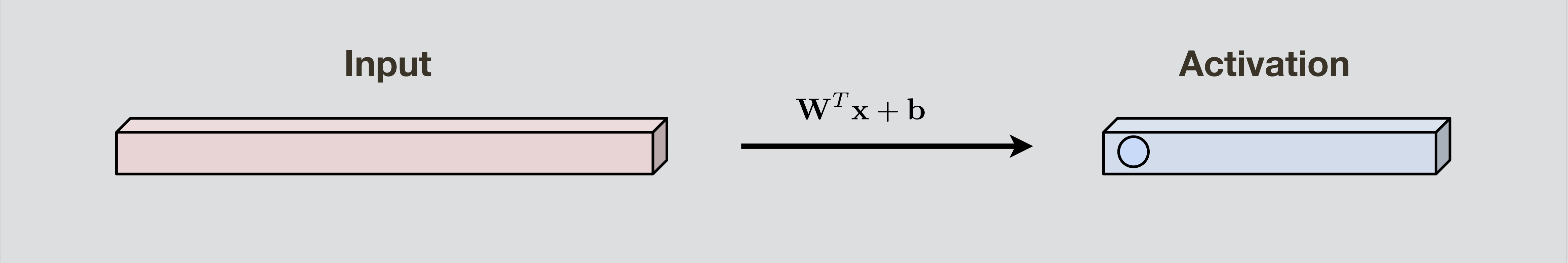


Review of **CNNs**

Convolutional Layer

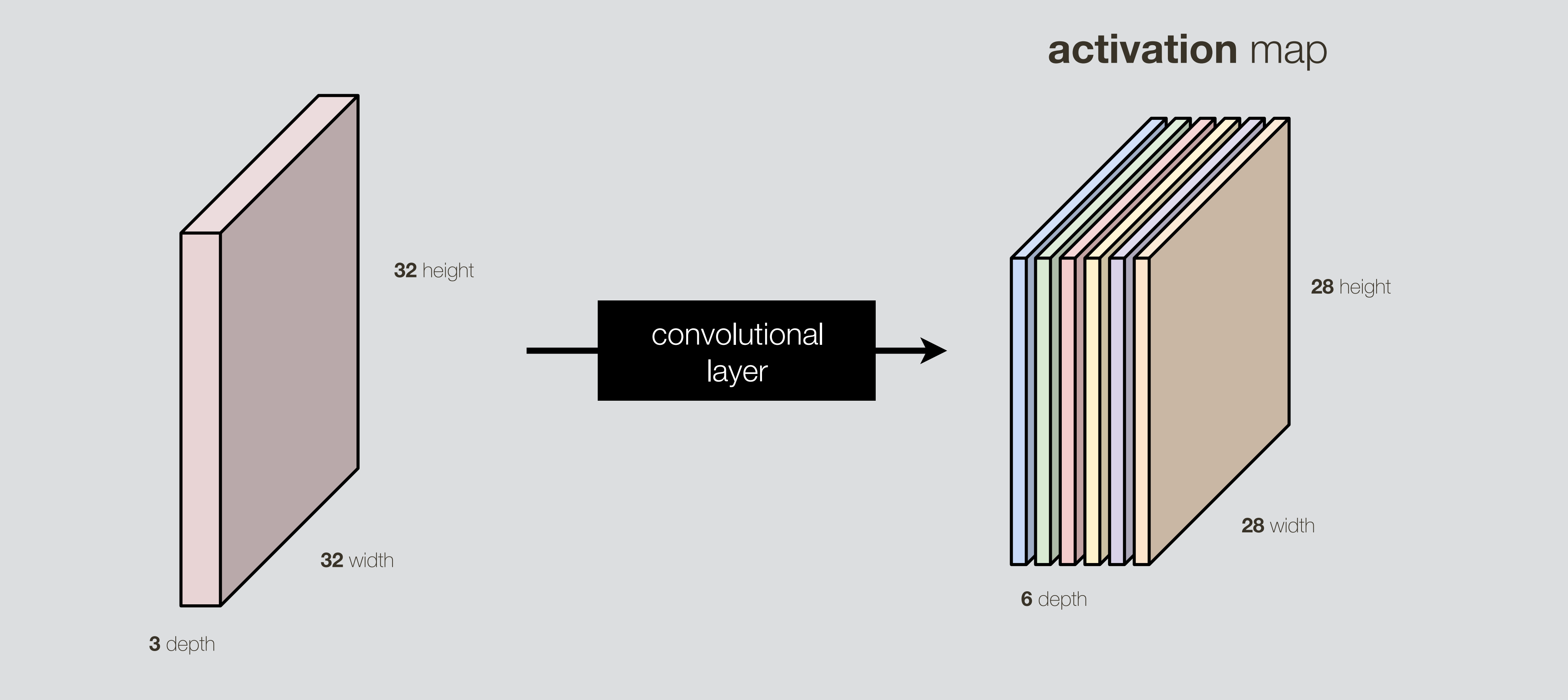


Fully Connected Layer

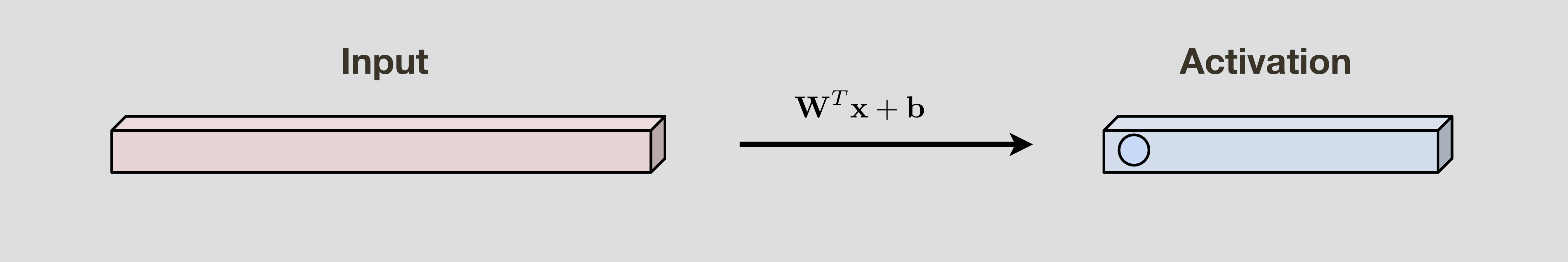


Review of **CNNs**

Convolutional Layer

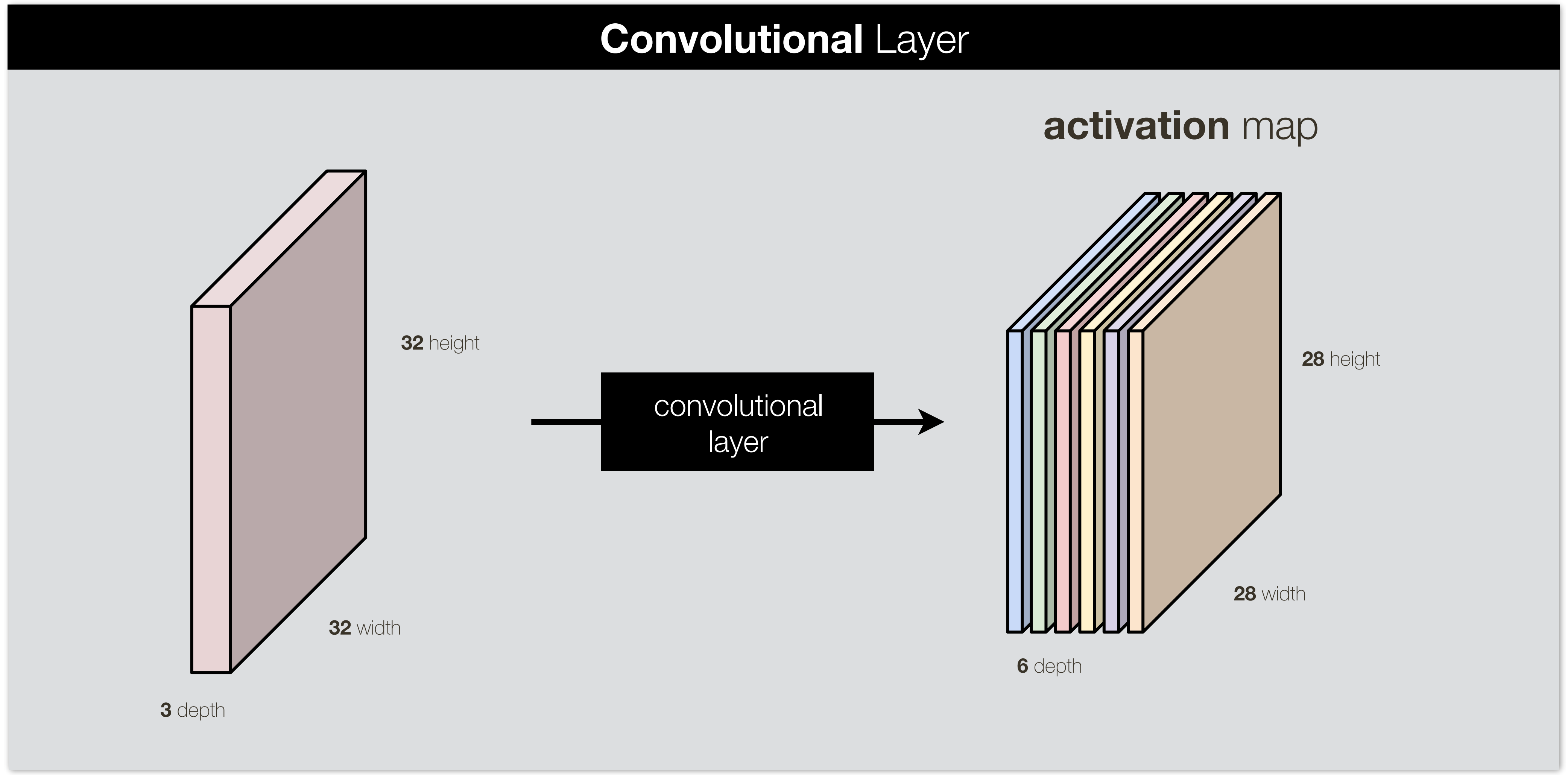


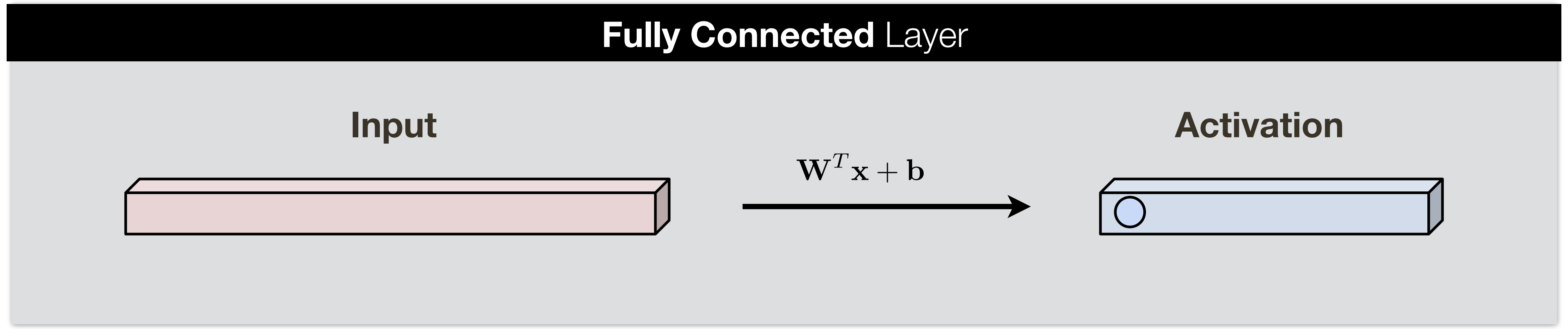
Fully Connected Layer

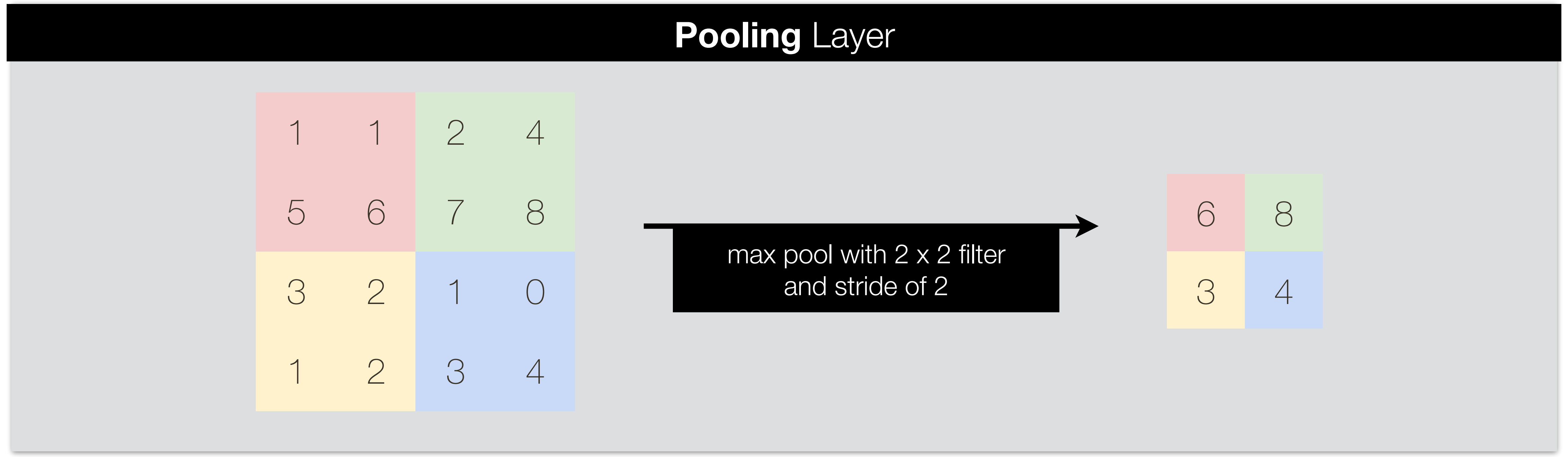


Pooling Layer

Review of **CNNs**





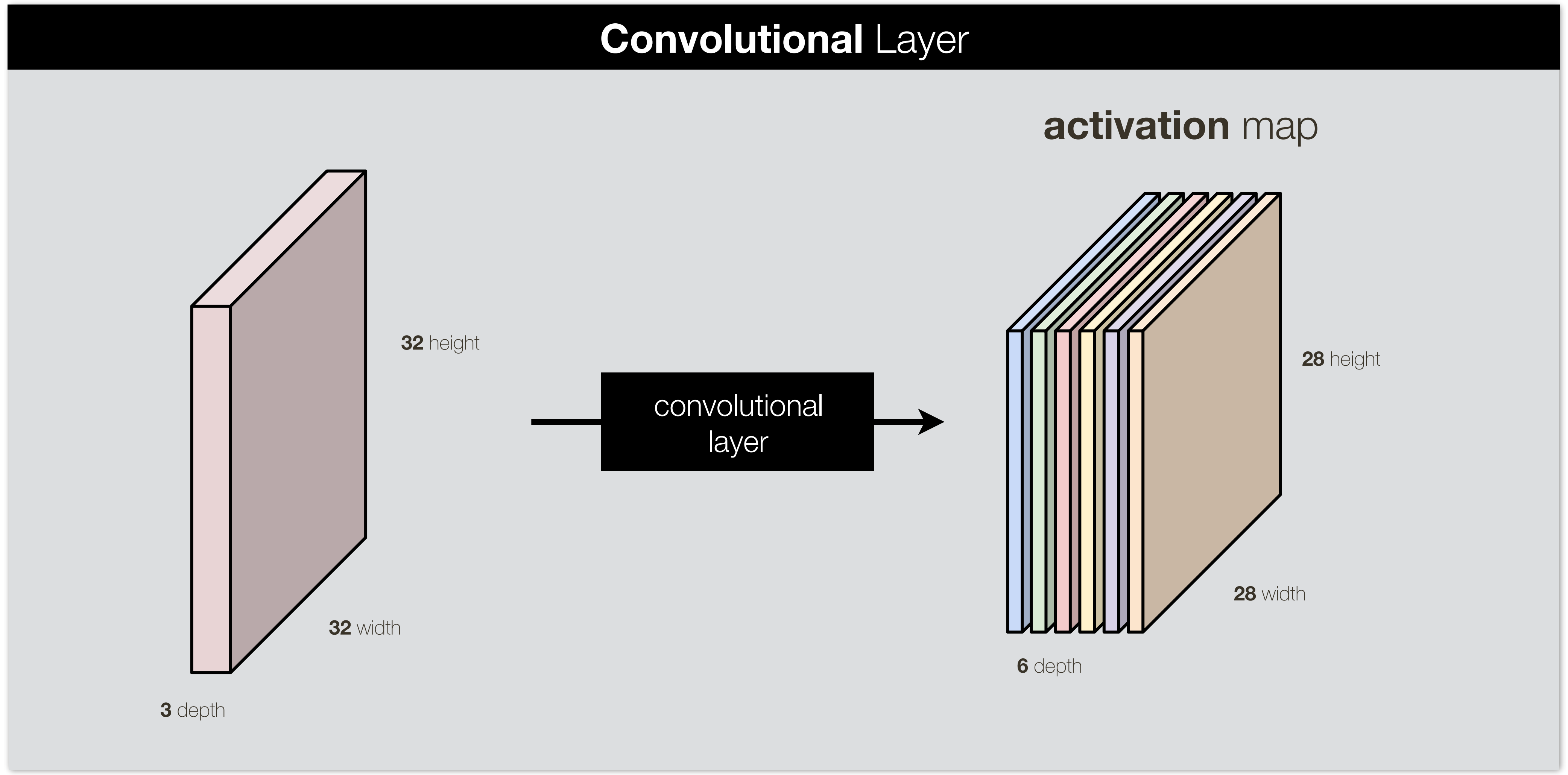


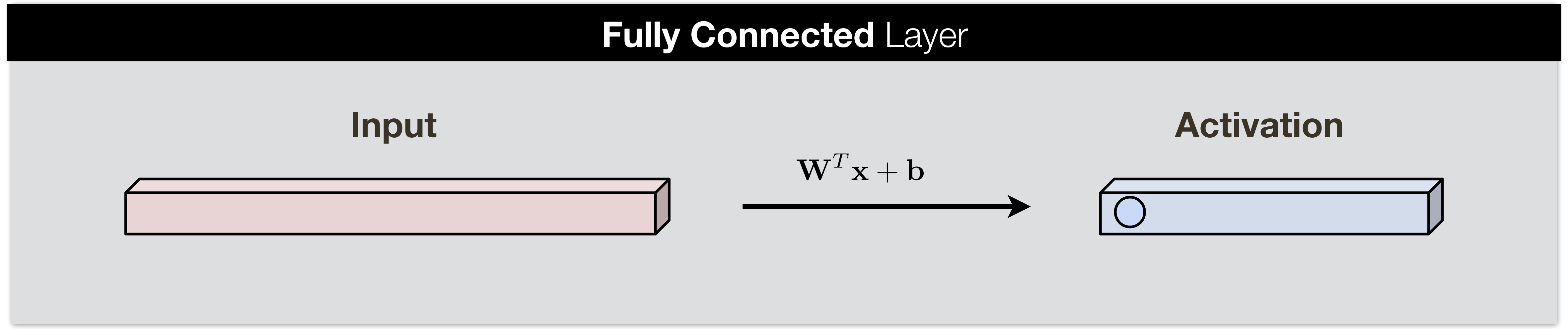
Effective Techniques for **Training**

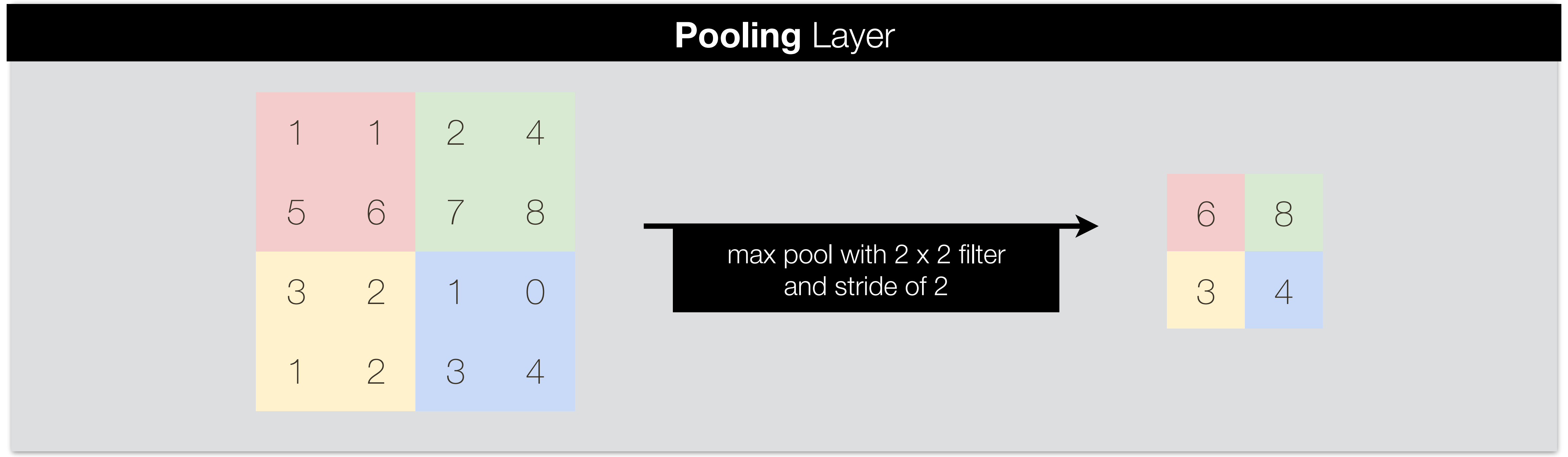
— **Regularization:** L1, L2, data augmentation

— **Transfer Learning:** fine-tuning networks

Review of **CNNs**







Effective Techniques for **Training**

— **Regularization:** L1, L2, data augmentation

— **Transfer Learning:** fine-tuning networks

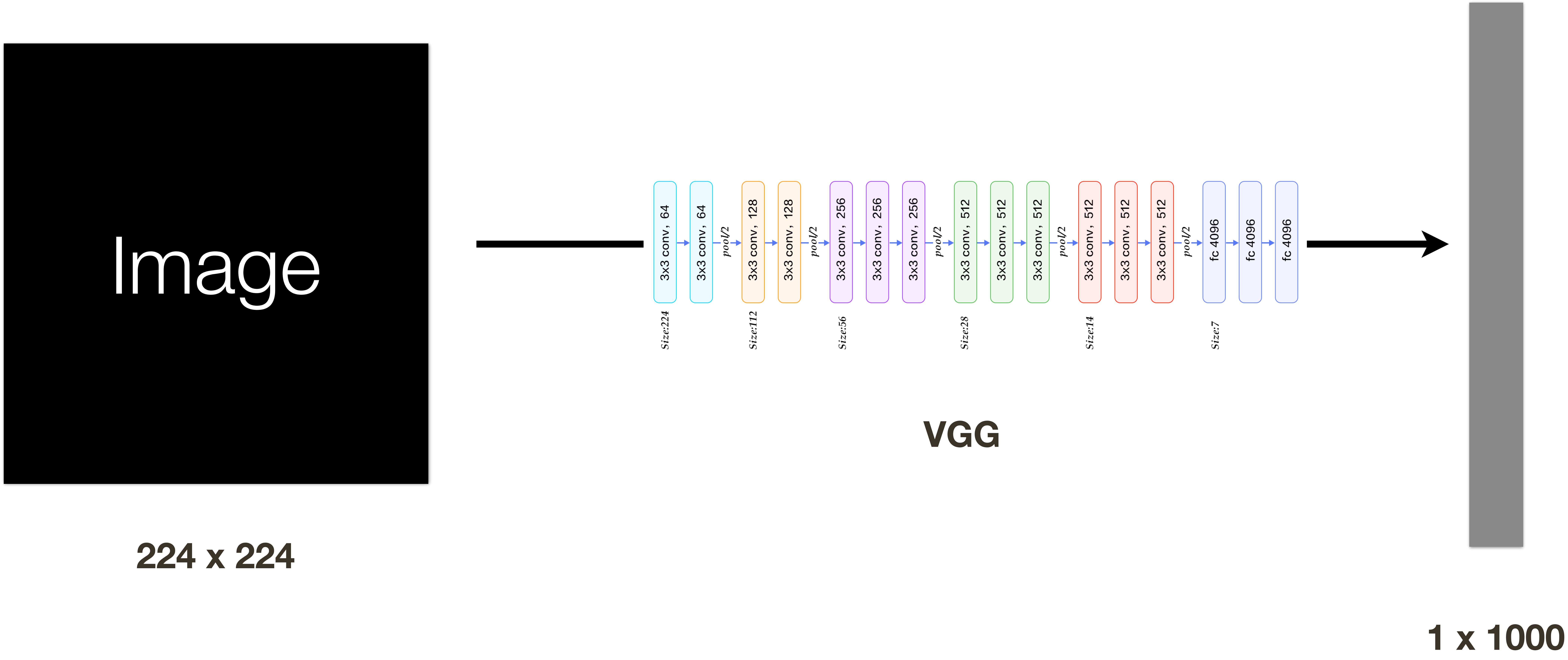
Vision **Applications** of CNNs

— **Classification:** AlexNet, VGG, GoogleLeNet, ResNet

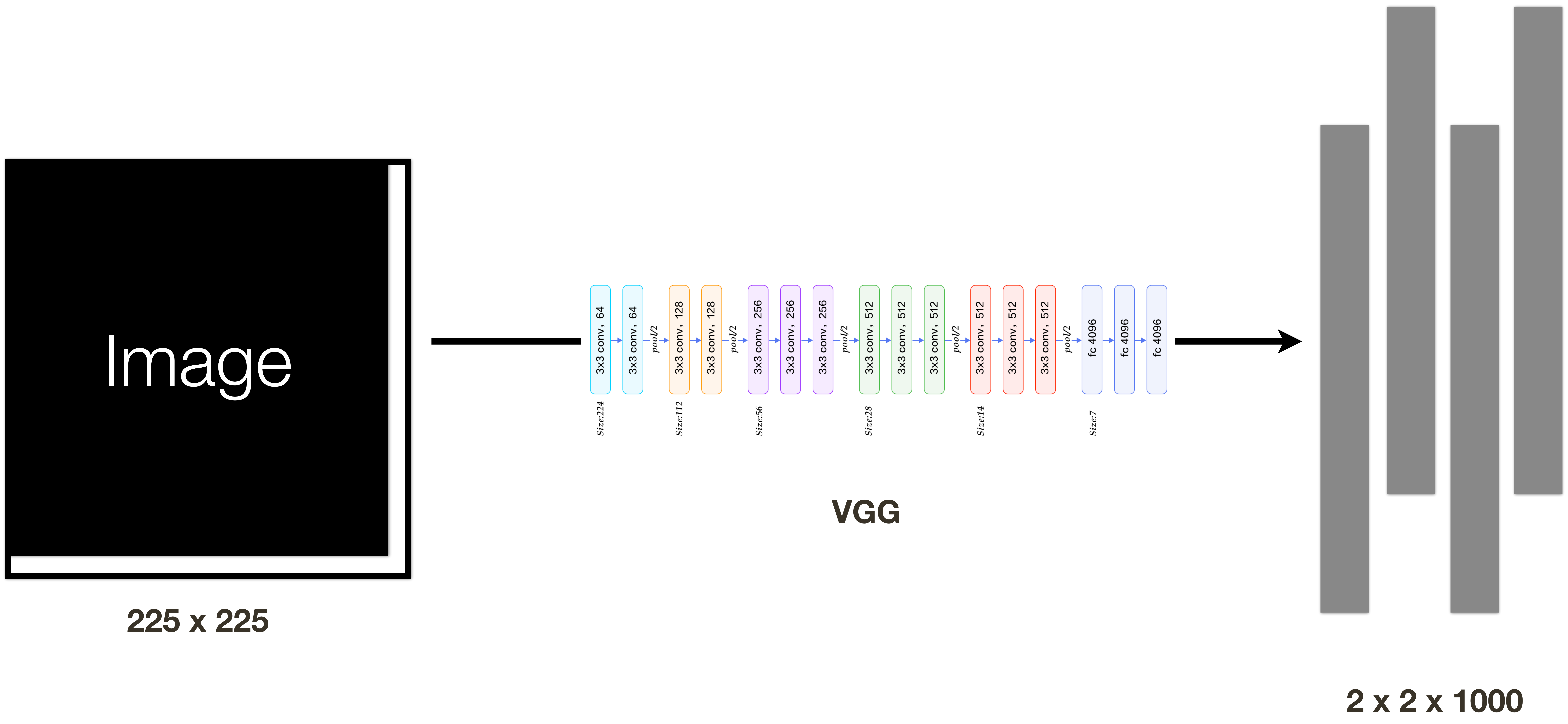
— **Segmentation:** Fully convolutional CNNs

— **Detection:** R-CNN, Fast R-CNN, Faster R-CNN, YOLO

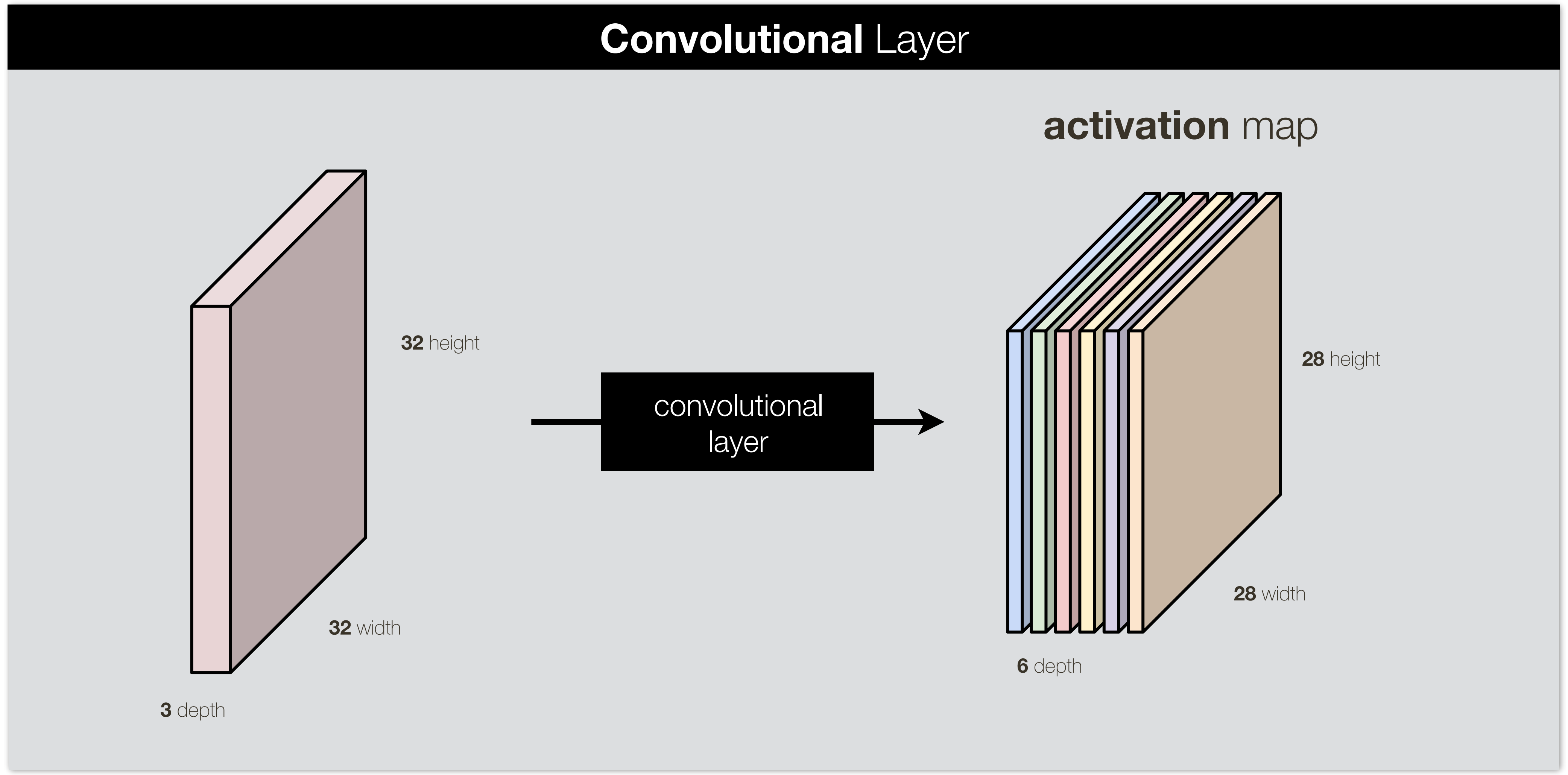
Any **CNN** Could be **Fully Convolutional**

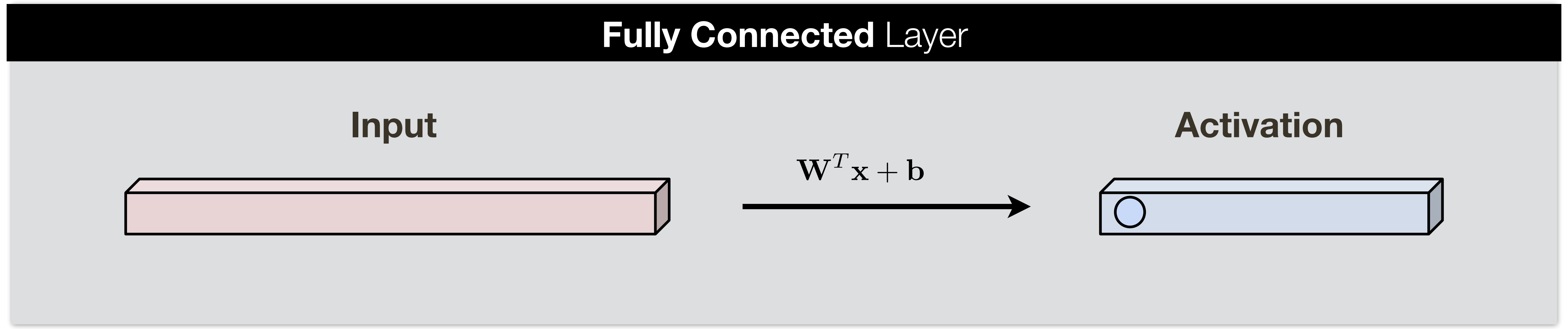


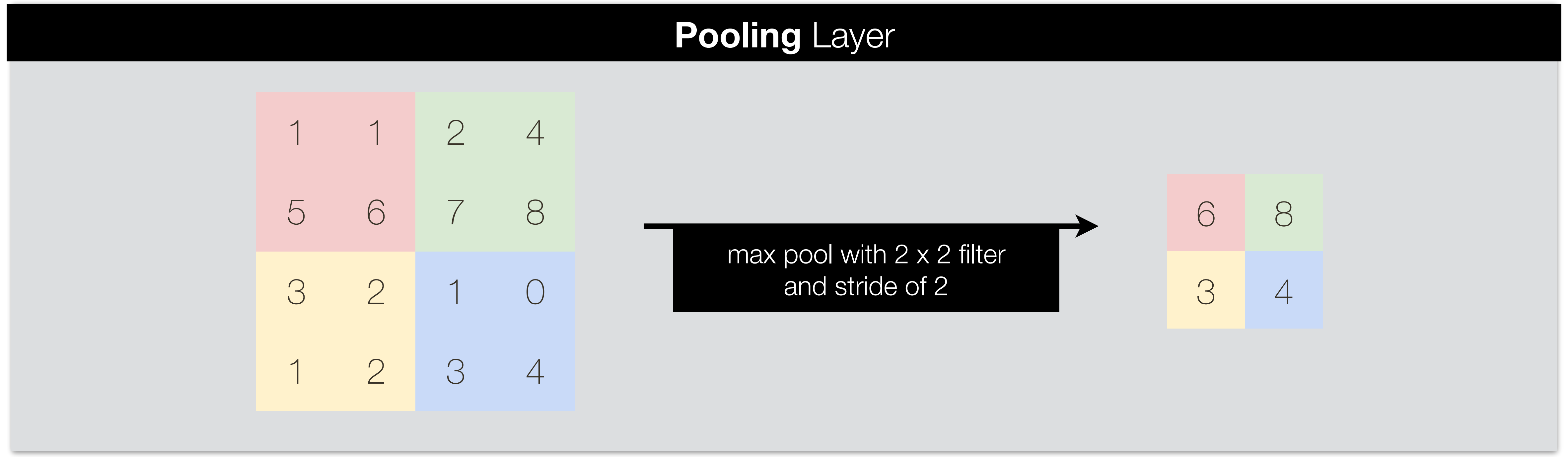
Any **CNN** Could be **Fully Convolutional**



Review of **CNNs**







Effective Techniques for **Training**

— **Regularization:** L1, L2, data augmentation

— **Transfer Learning:** fine-tuning networks

Vision **Applications** of CNNs

— **Classification:** AlexNet, VGG, GoogleLeNet, ResNet

— **Segmentation:** Fully convolutional CNNs

— **Detection:** R-CNN, Fast R-CNN, Faster R-CNN, YOLO