Object Detection

Instructor: Seunghoon Hong
School of Computing, KAIST

Course logistics

- Assignment 2 will be out today
  - **Due date: midnight 10/16**

Today's agenda

- Review: CNN for image classification
- Intro to object detection
    - Problem definition
    - Overall pipeline
    - Evaluation metrics
- Object detection using CNN
    - R-CNN
    - SPPNet, Fast R-CNN, Faster R-CNN

Image classification

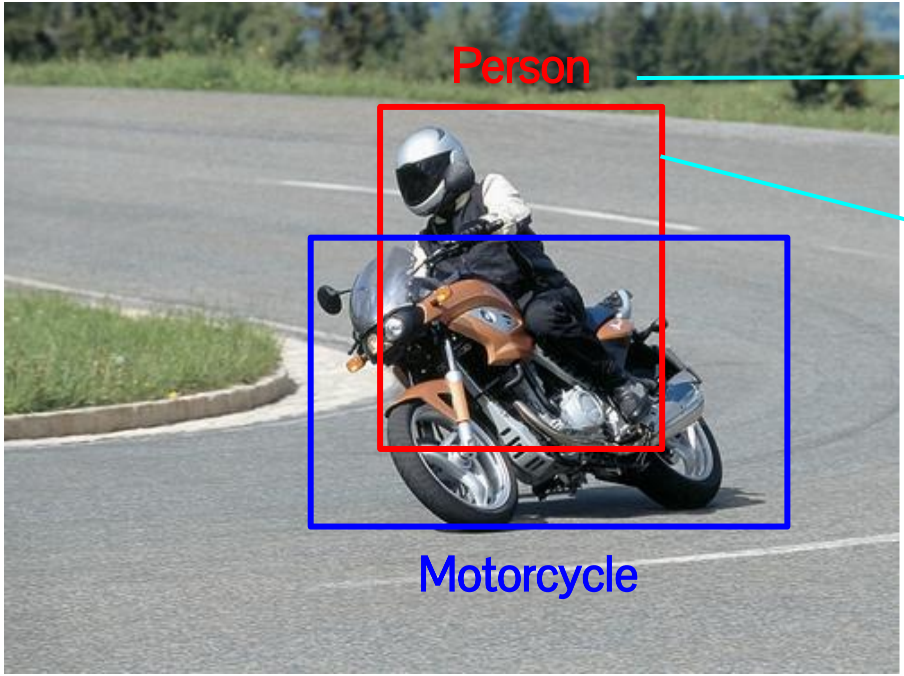
- identifying presence/absence of concepts in an image

Is there motorcycle? Yes
Is there person? Yes
Is there dog? No
...

Outputs:
Motorcycle, person

Object detection

- Classification + **localization** of object instances



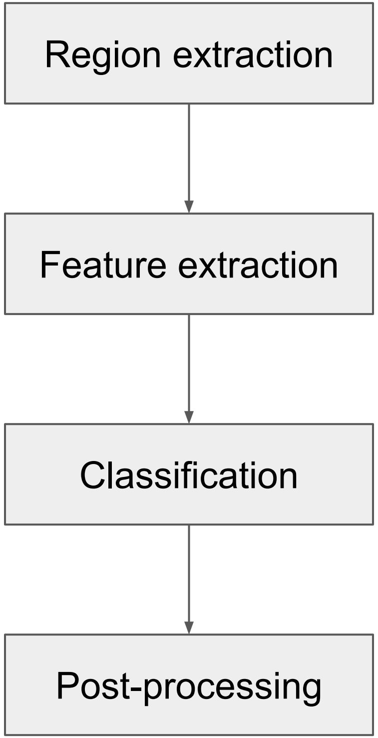
Class label:
A label indicating object category

Bounding box:
A 2d box indicating object location and size

Object detection

- Challenges
  - How do we localize the object?
  - How do we handle variable number of object?
  - How do we handle various size, aspect ratio, etc?

Traditional object detection pipeline



Traditional object detection pipeline

Region extraction

↓

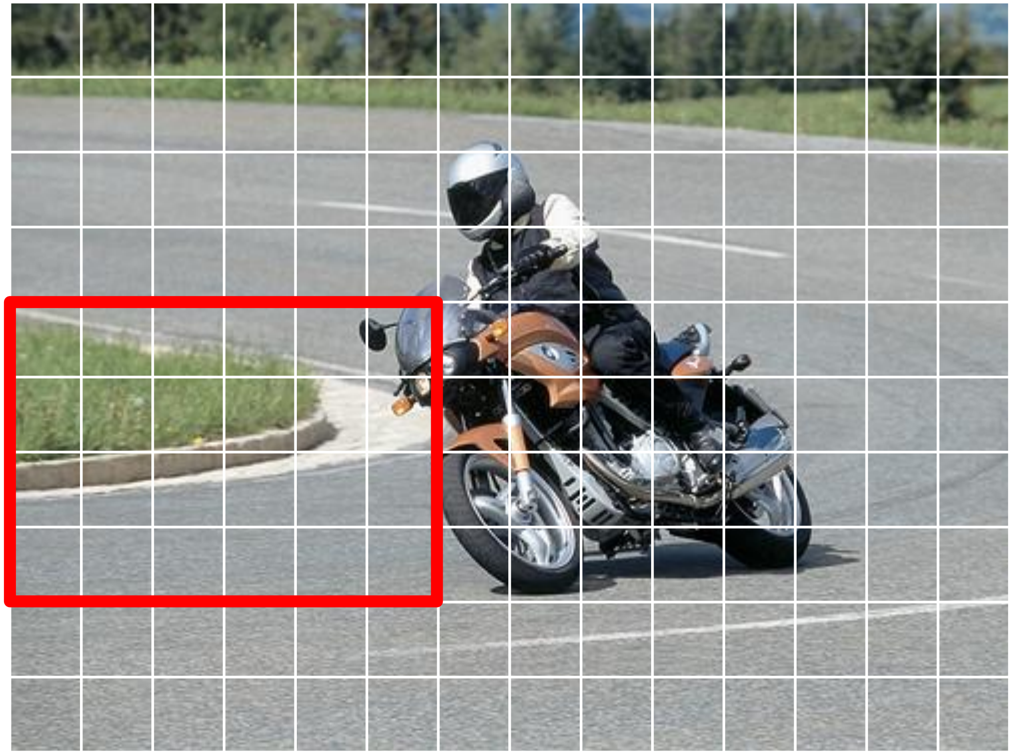
Feature extraction

↓

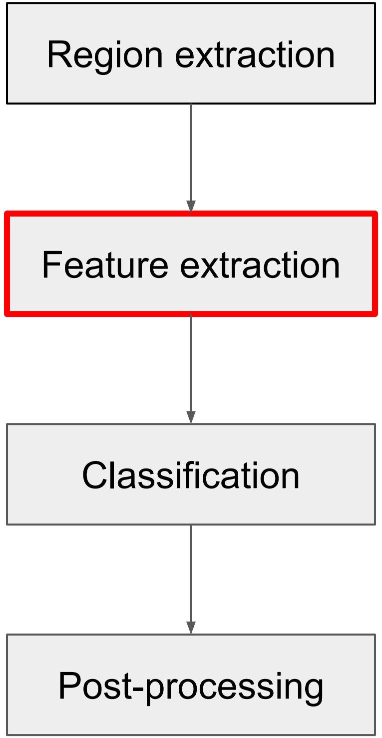
Classification

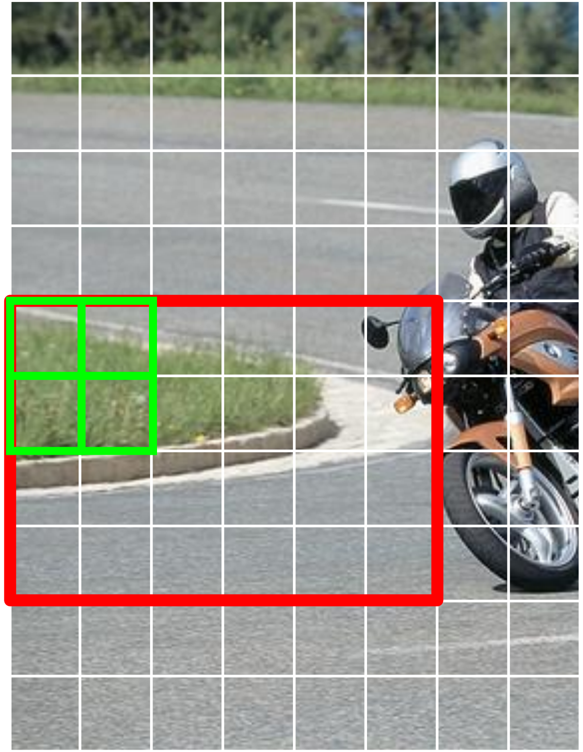
↓

Post-processing

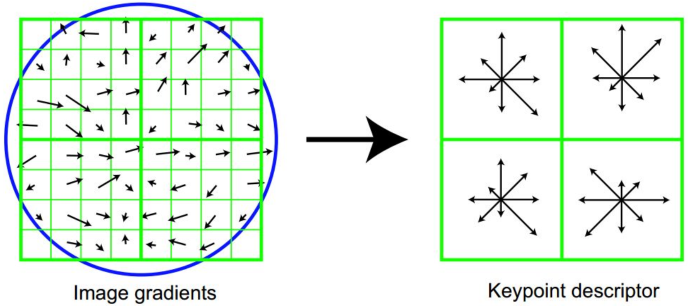


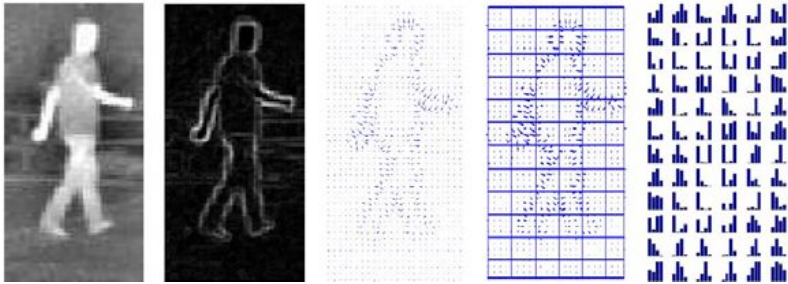
Traditional object detection pipeline



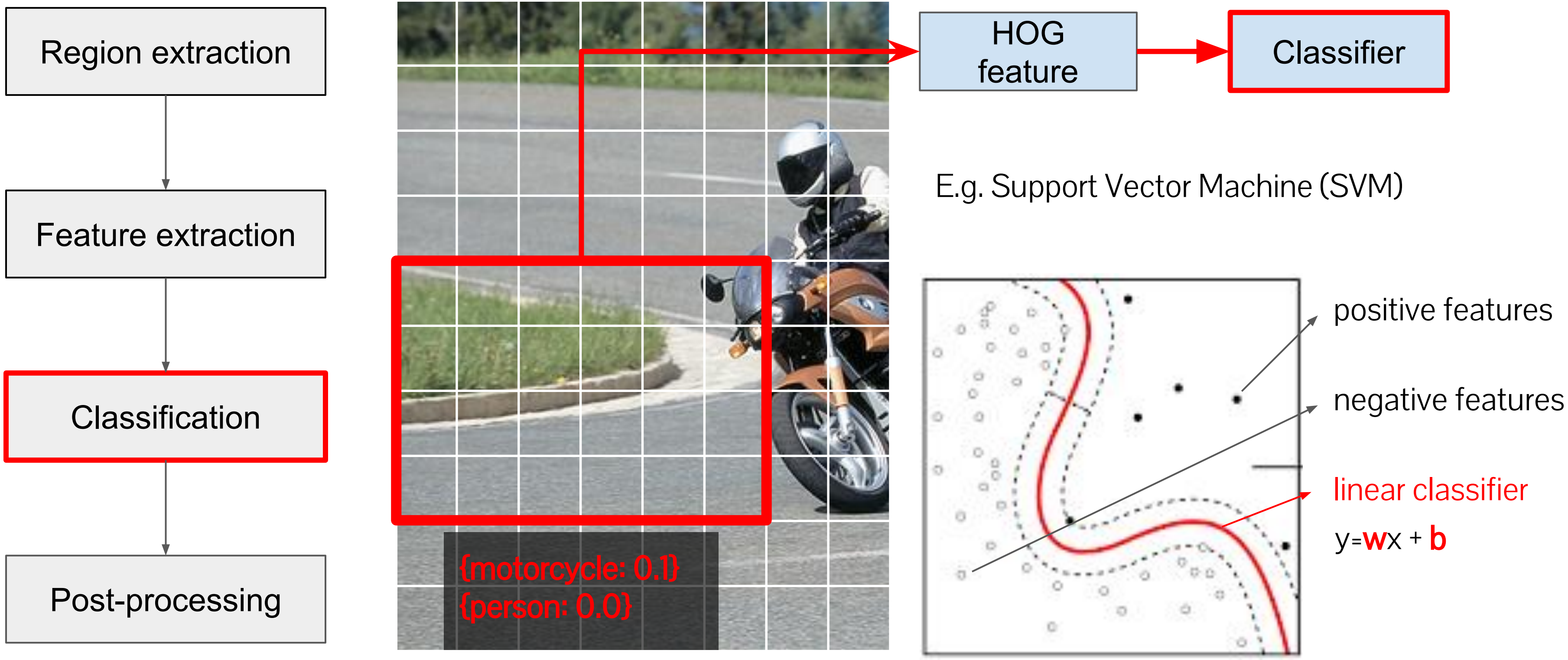


E.g. Histogram of Oriented Gradient (HOG)





Traditional object detection pipeline



Traditional object detection pipeline

Region extraction

↓

Feature extraction

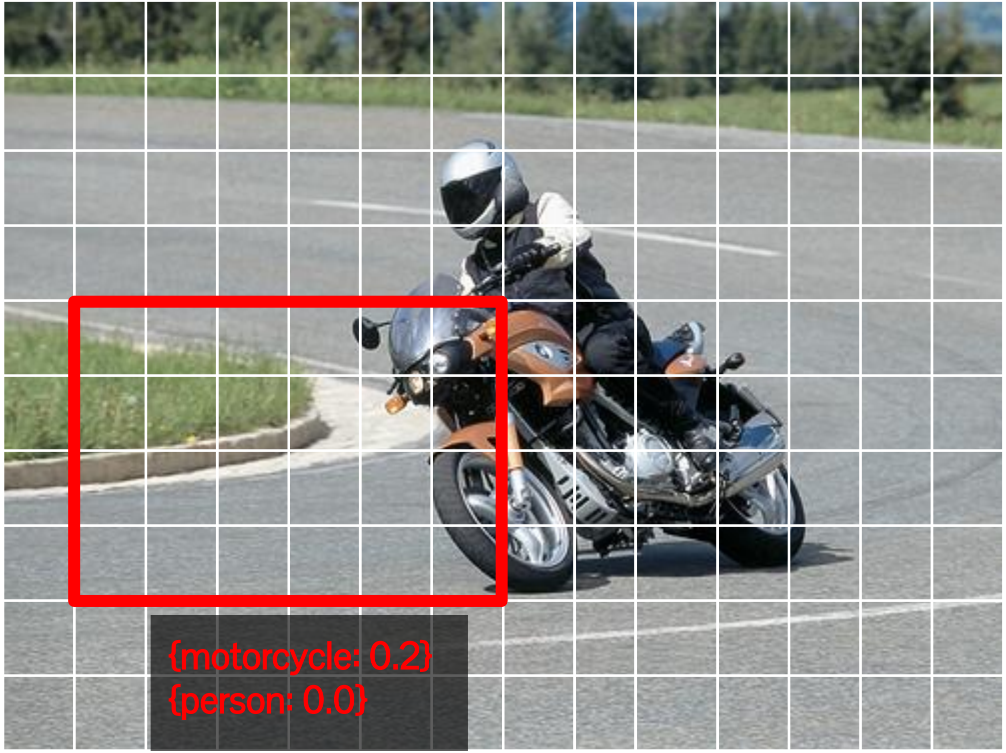
↓

Classification

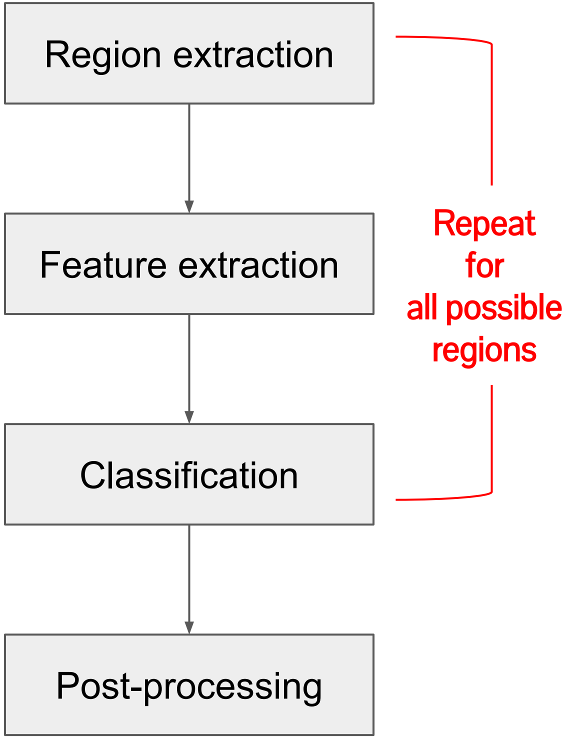
↓

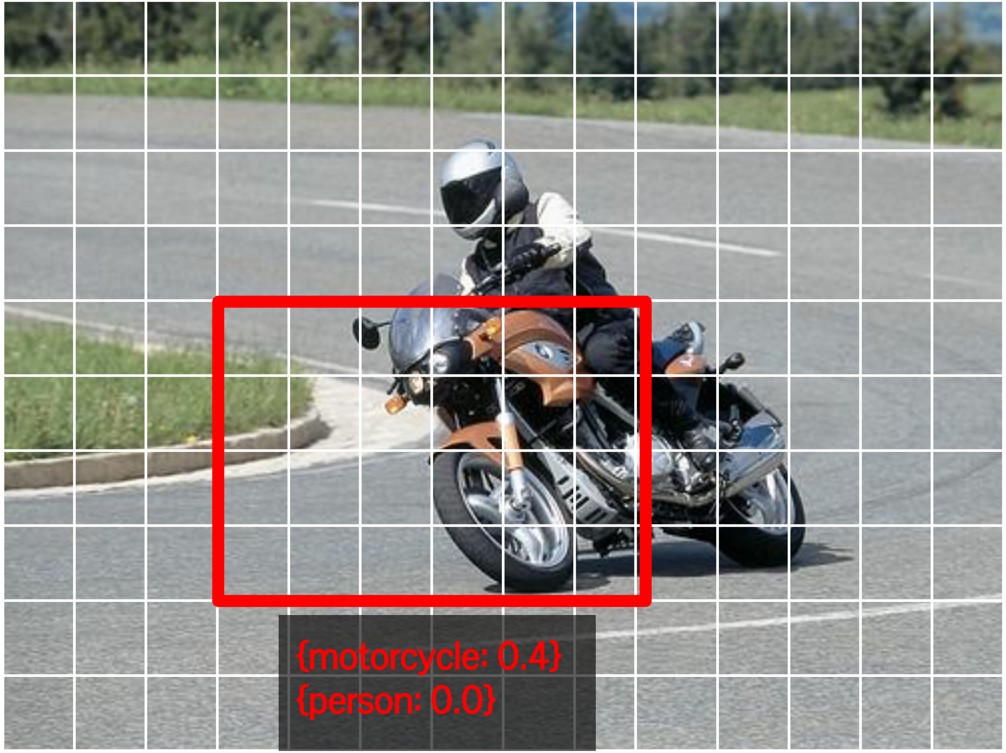
Post-processing

Repeat for all possible regions



Traditional object detection pipeline -- sliding window





Traditional object detection pipeline -- sliding window

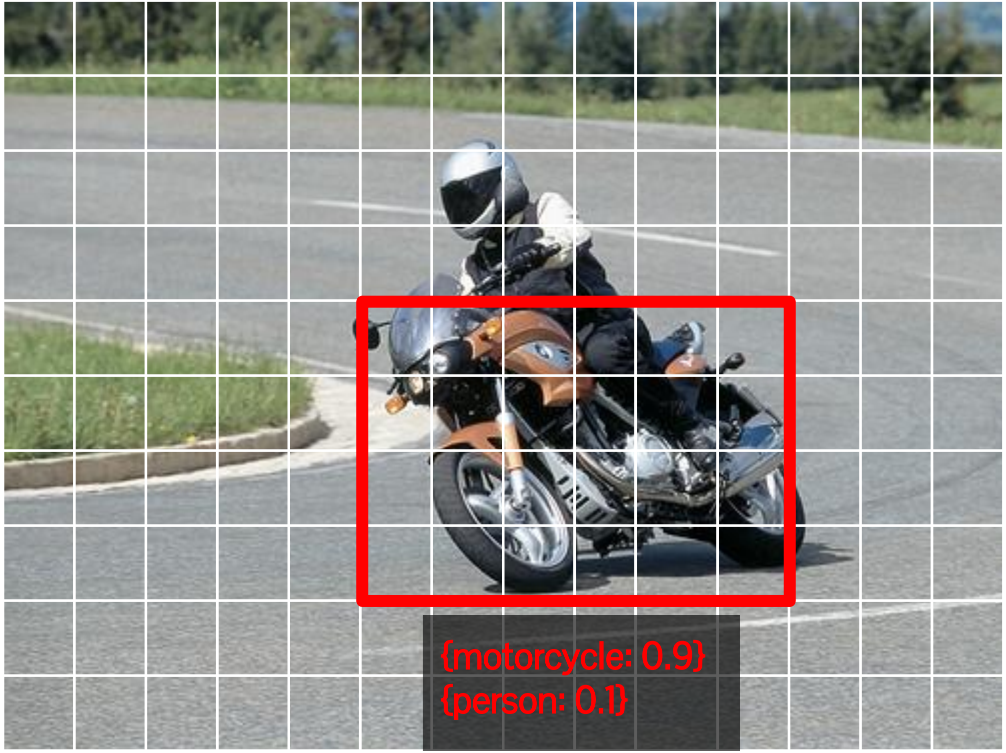
Region extraction

Feature extraction

Classification

Post-processing

Repeat
for
all possible
regions



Traditional object detection pipeline -- sliding window

Region extraction

↓

Feature extraction

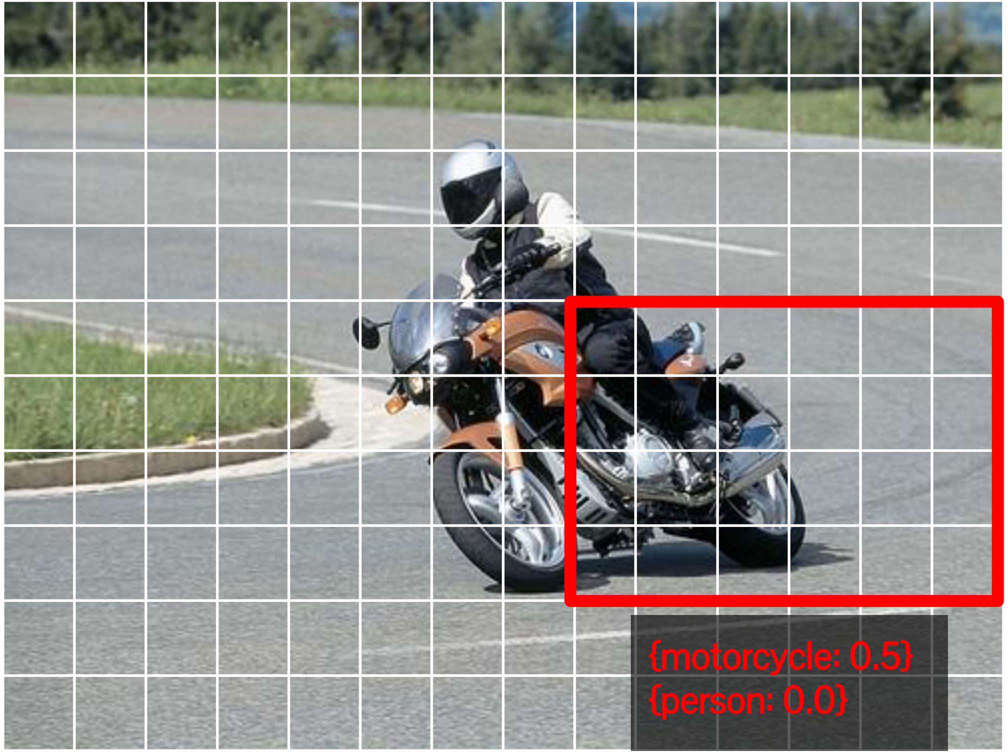
↓

Classification

↓

Post-processing

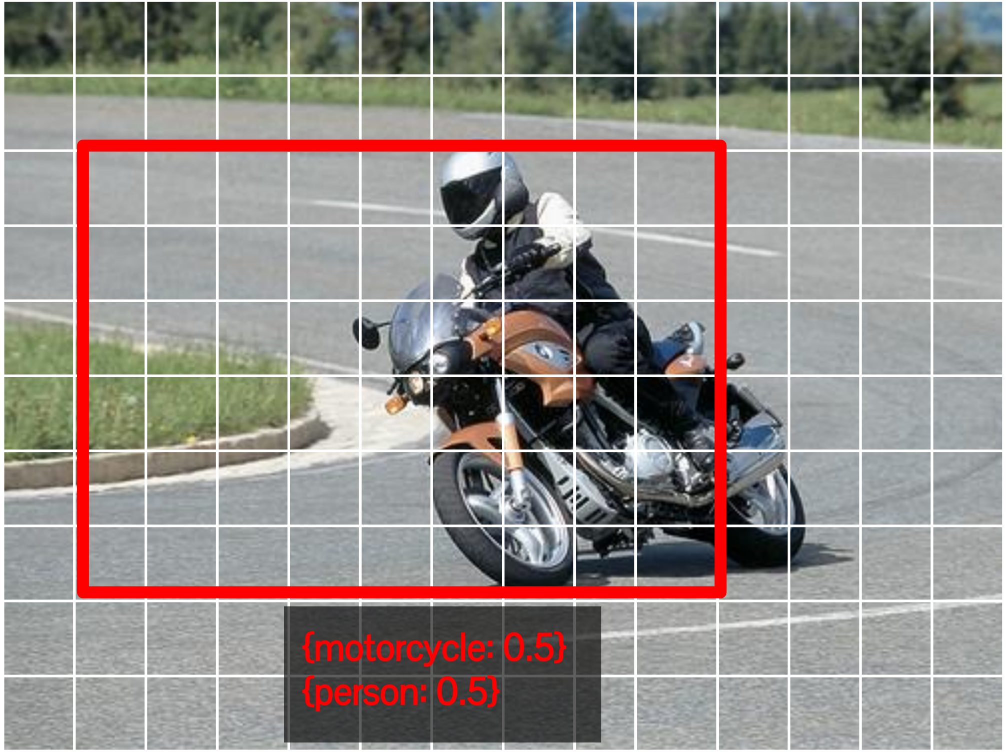
Repeat for all possible regions



Traditional object detection pipeline -- sliding window

Region extraction
↓
Feature extraction
↓
Classification
↓
Post-processing

Repeat for all possible regions, scales



Traditional object detection pipeline -- sliding window

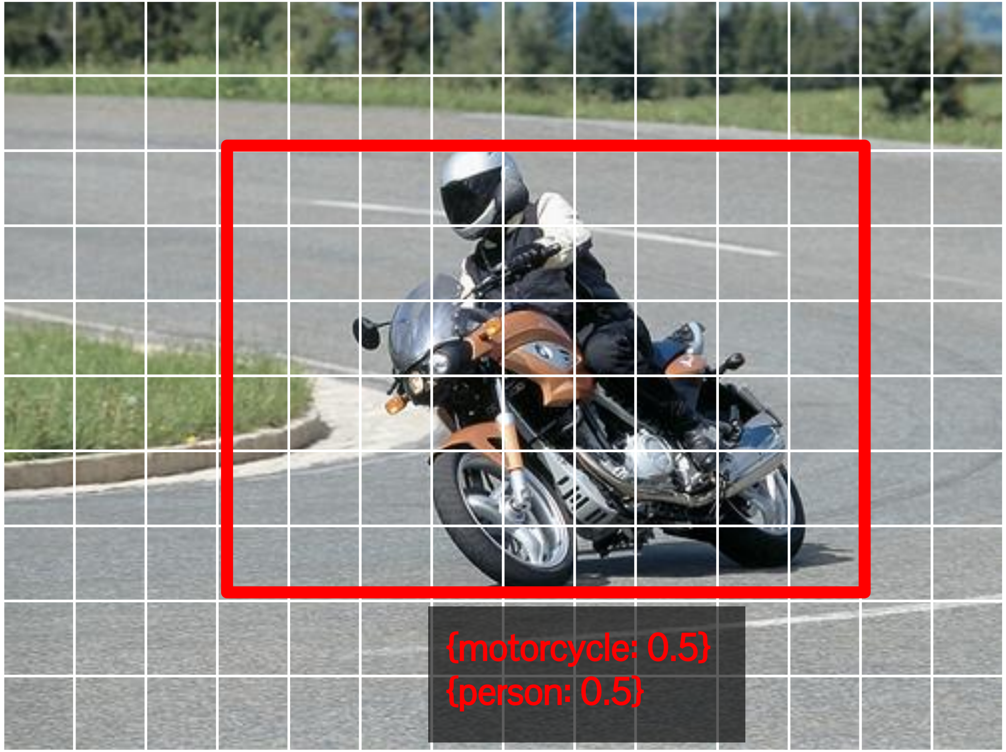
Region extraction

Feature extraction

Classification

Post-processing

Repeat
for
all possible
regions,
scales



Traditional object detection pipeline -- sliding window

Region extraction

↓

Feature extraction

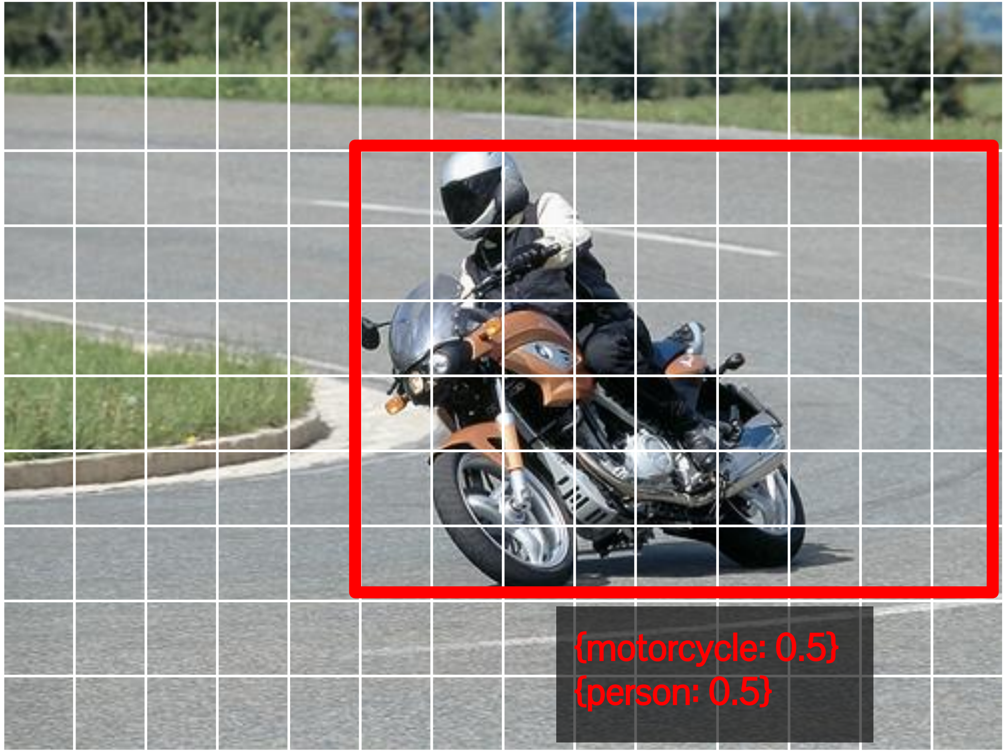
↓

Classification

↓

Post-processing

Repeat for all possible regions, scales



Traditional object detection pipeline -- sliding window

Region extraction

↓

Feature extraction

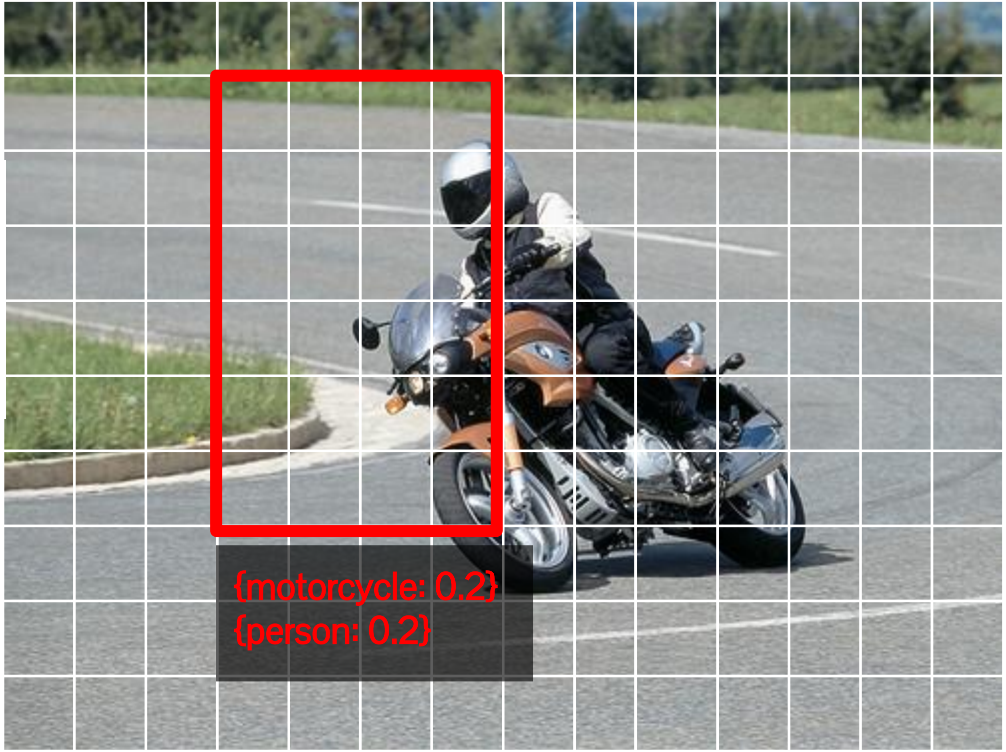
↓

Classification

↓

Post-processing

Repeat for all possible regions, scales, aspect ratio



Traditional object detection pipeline -- sliding window

Region extraction

↓

Feature extraction

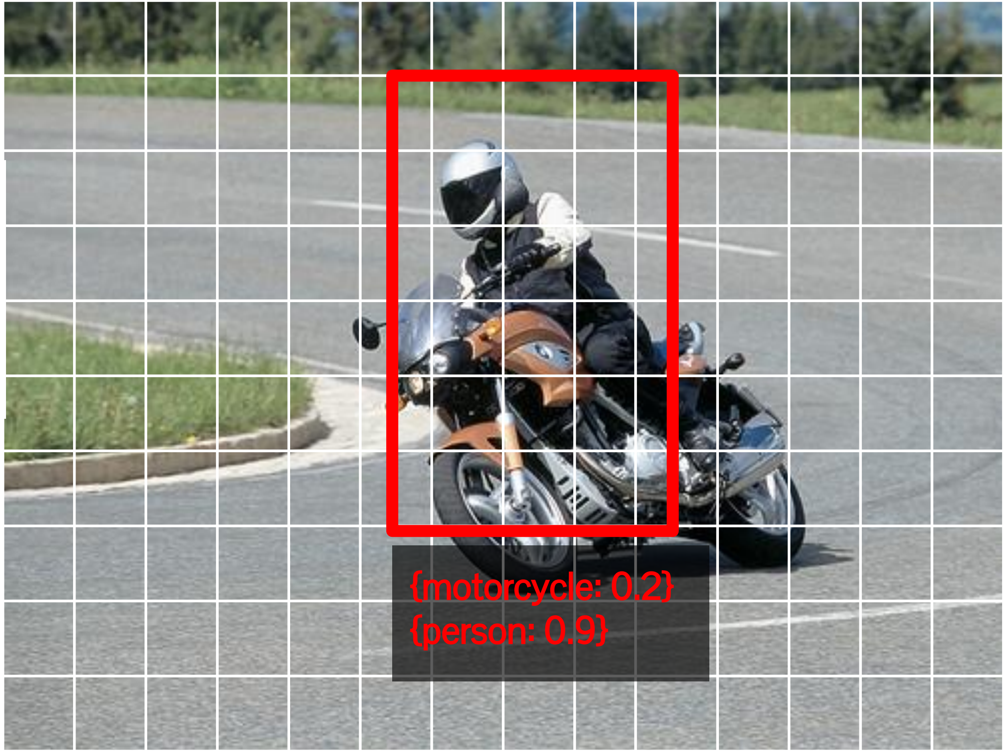
↓

Classification

↓

Post-processing

Repeat
for
all possible
regions,
scales,
aspect ratio



Traditional object detection pipeline -- sliding window

Region extraction

↓

Feature extraction

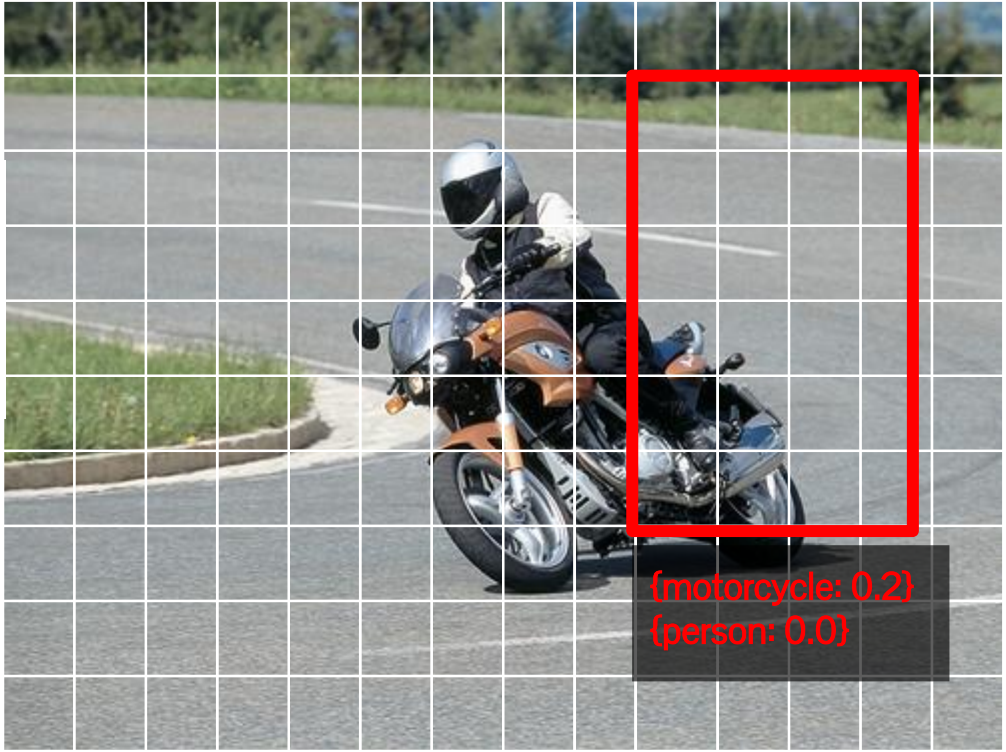
↓

Classification

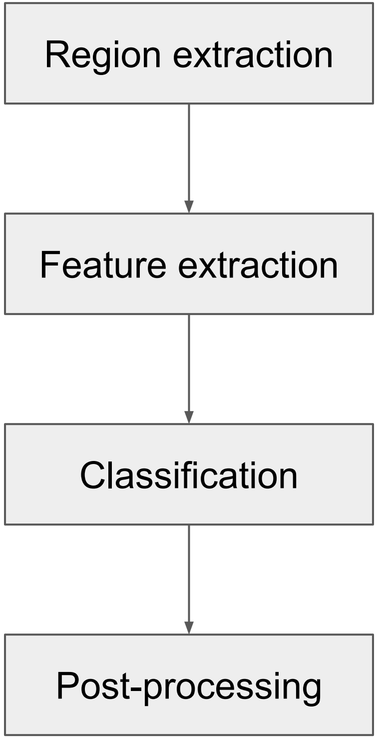
↓

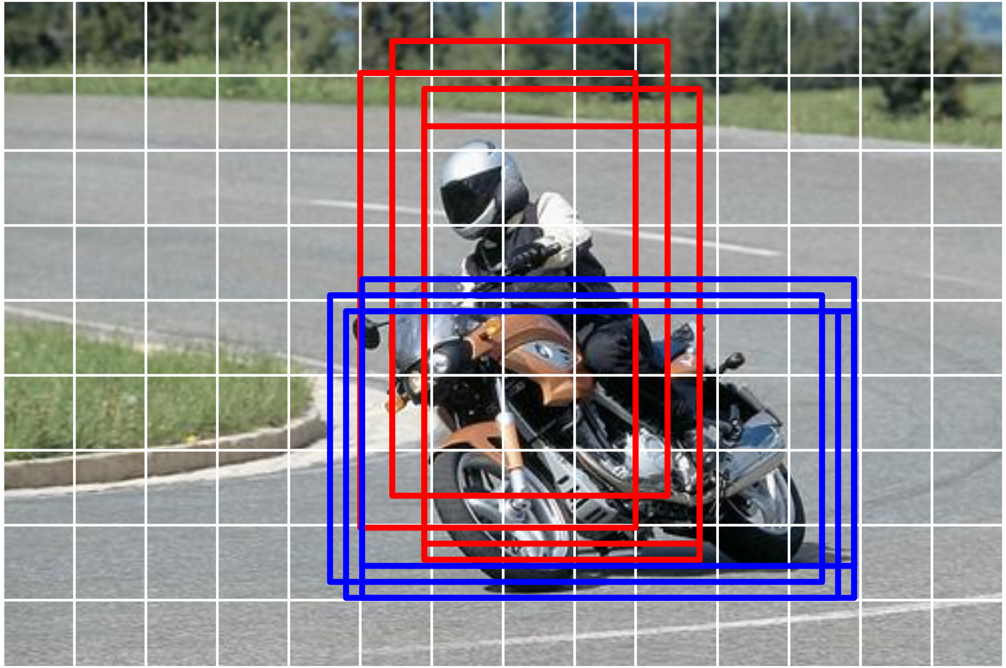
Post-processing

Repeat for all possible regions, scales, aspect ratio



Traditional object detection pipeline





Bounding boxes that have probability > 0.9 for person and motorcycle

Traditional object detection pipeline

Region extraction

↓

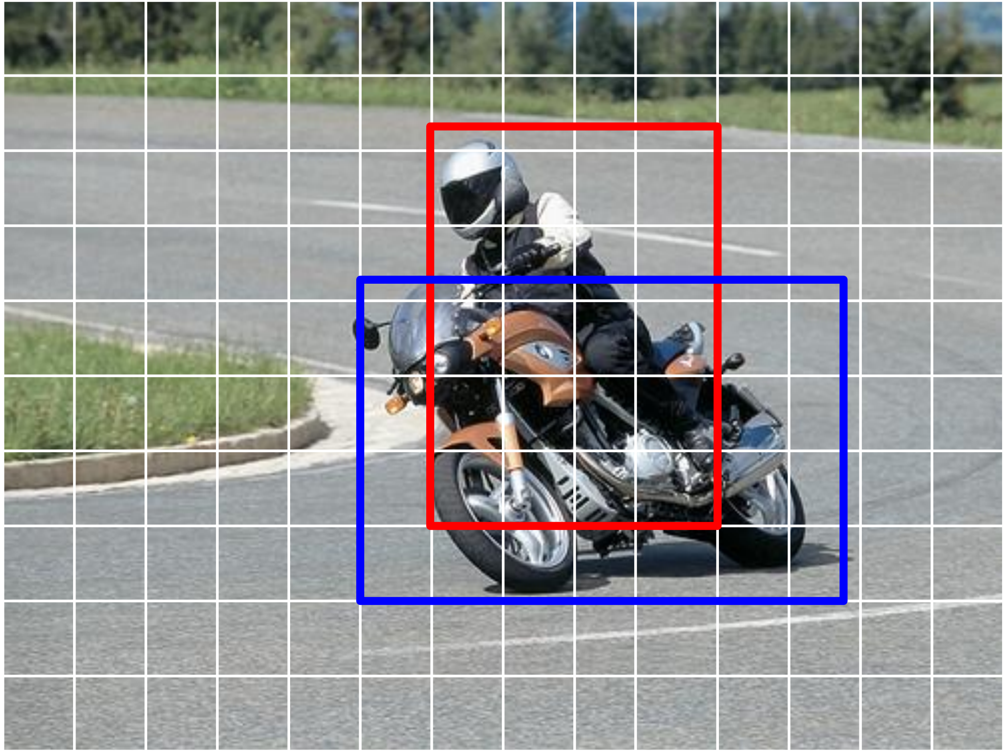
Feature extraction

↓

Classification

↓

Post-processing



Traditional object detection pipeline

Region extraction

↓

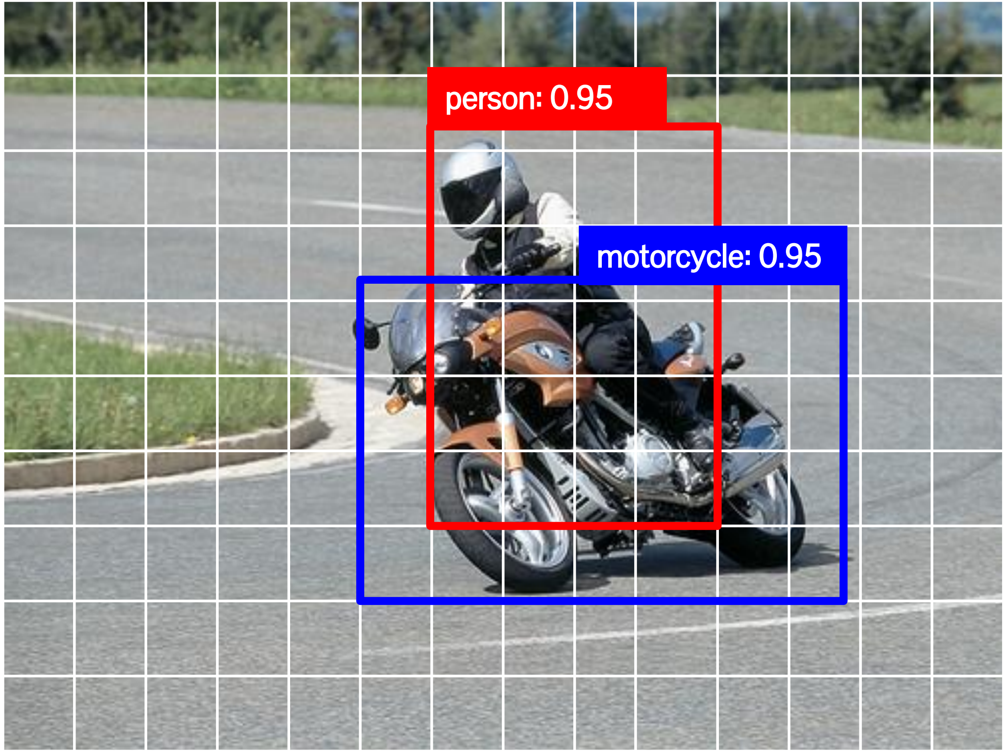
Feature extraction

↓

Classification

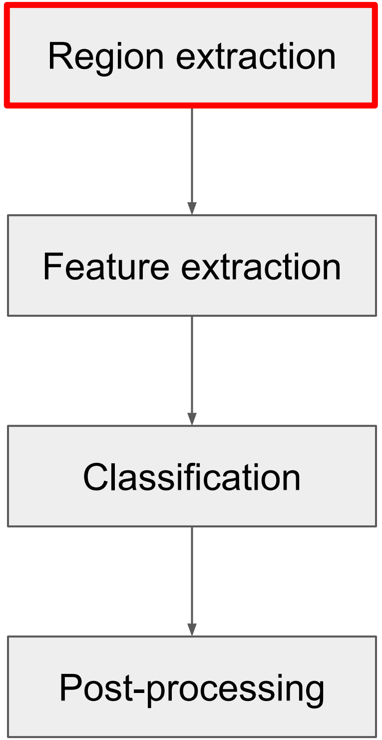
↓

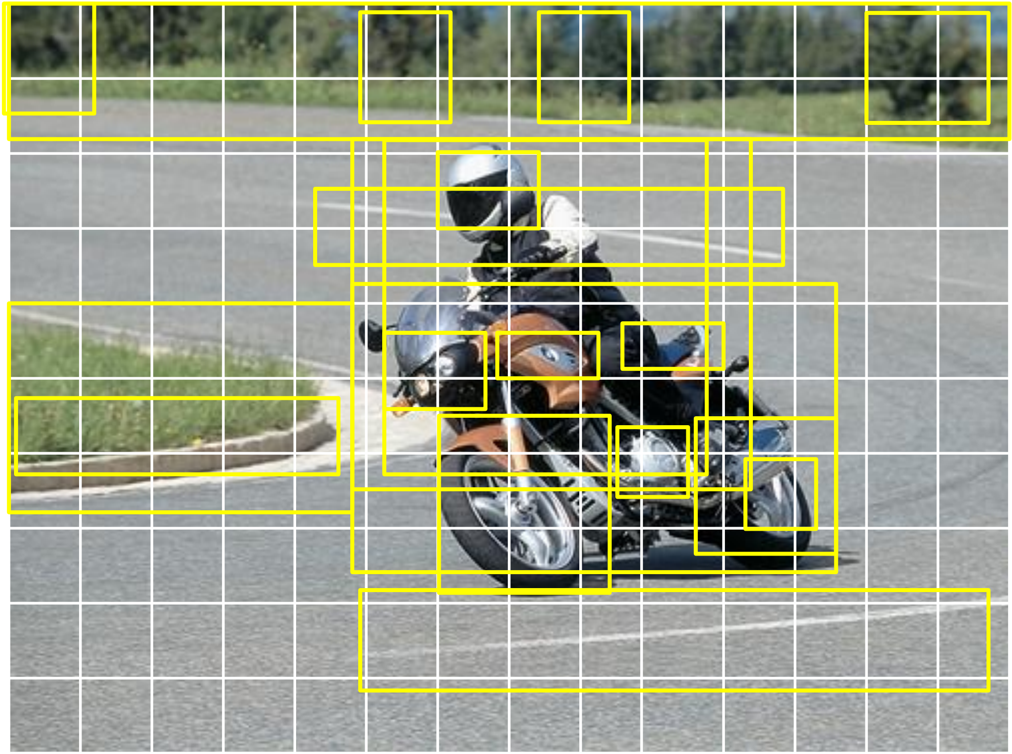
Post-processing



Traditional object detection pipeline -- region proposal

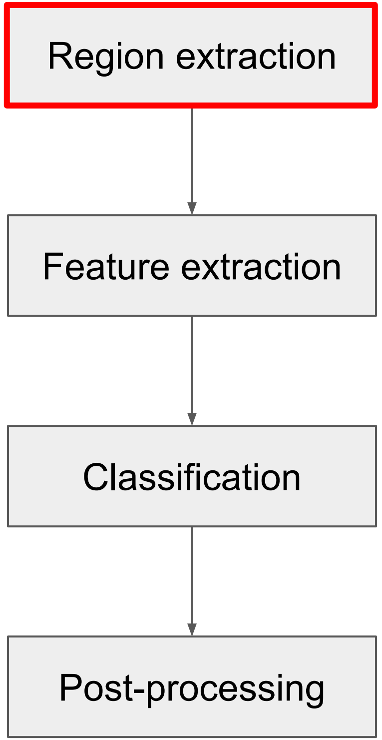
Extract candidate bounding boxes by measuring “objectness” of all possible boxes

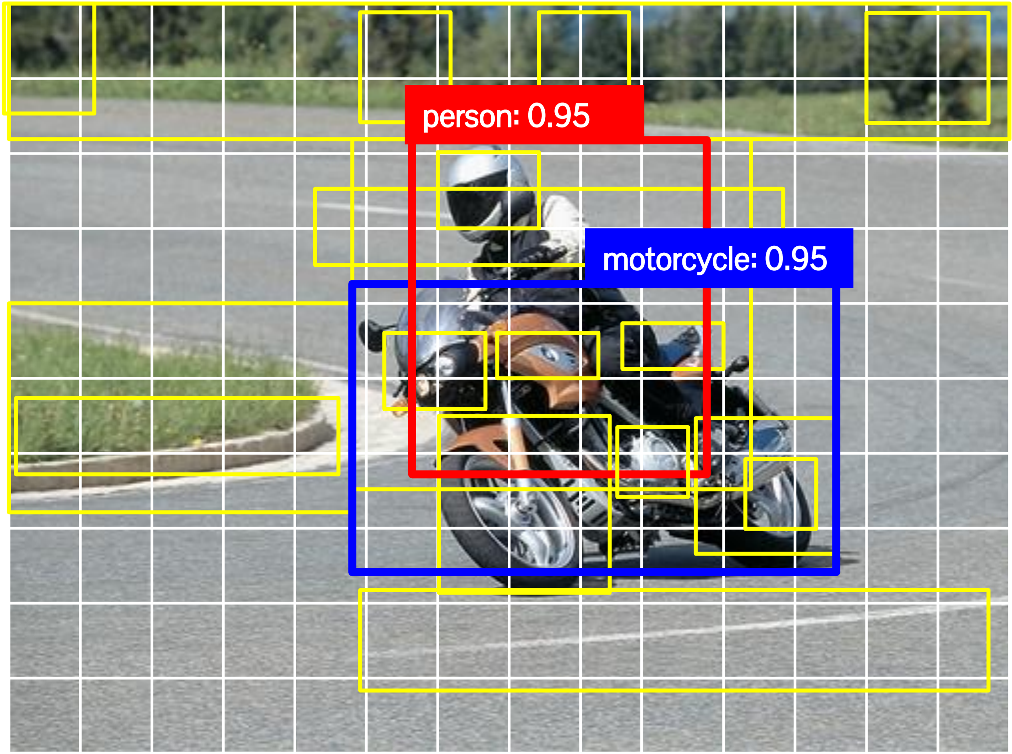




Traditional object detection pipeline -- region proposal

Extract candidate bounding boxes by measuring “objectness” of all possible boxes





Challenge 1. How can we detect these guys?

figure: http://www.deceptology.com/2011-02/participants-in-facebook-game-of-lying.html

Challenge 2. How fast should we detect the object?

- Example: computational cost of sliding window approach
  - Image grid of the size W (width) x H (height)
  - Number of scales: S
  - Number of aspect ratios: A
  - Total number of classification: W x H x S x A
  - If W, H=128, S=5, A=32: **2,420,640** classification per image

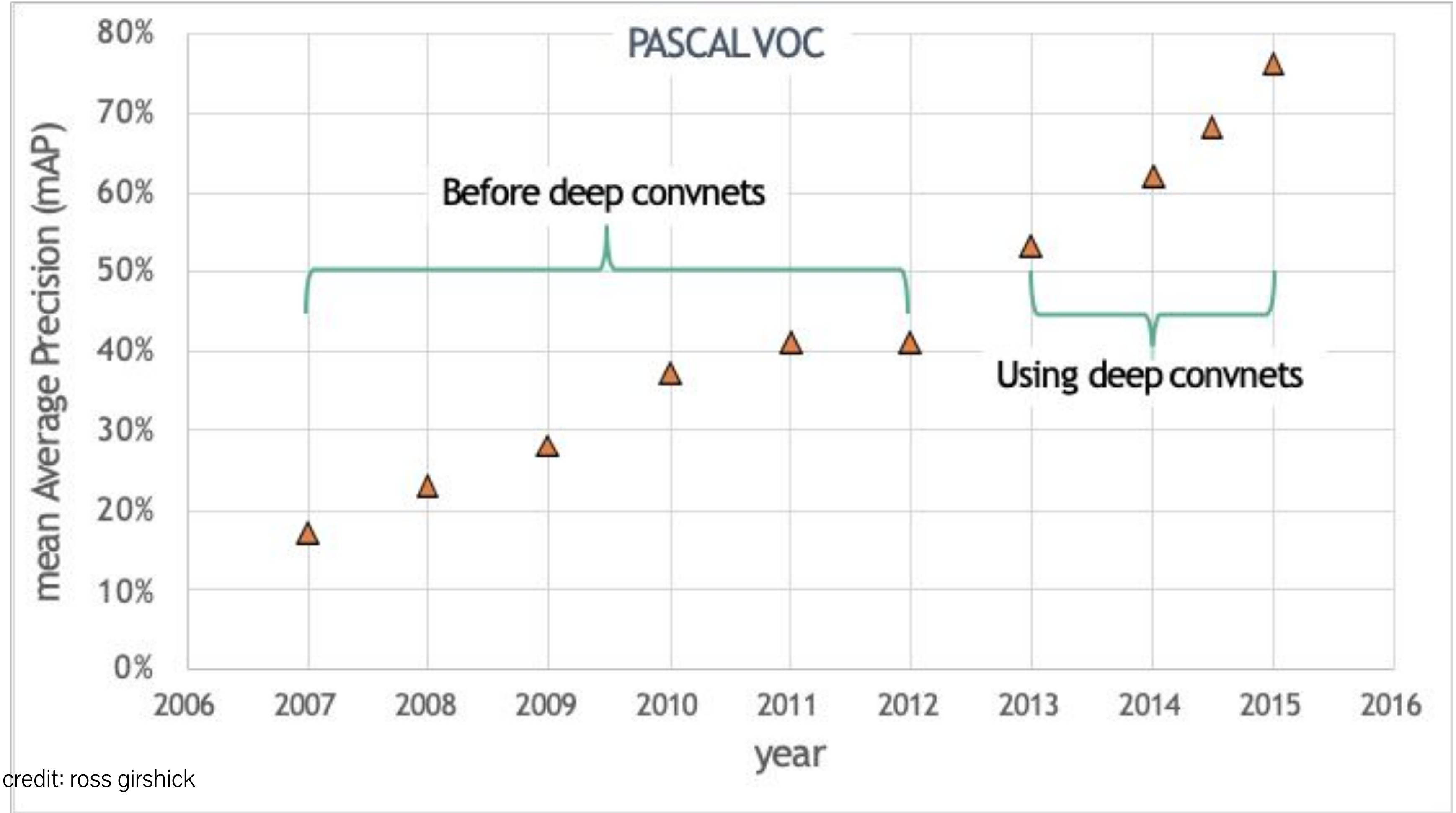
DPM v5[1]: 0.07 FPS = 14s per image

For a car driving at 80 km/h,
it proceeds > 300m to detect objects

Summary: Introduction

- Object detection: region-based classification
- Pipeline of object detection:
  - Region extraction, feature extraction, classification, post-processing
- Challenges:
  - Building a robust representation (i.e. feature)
  - Improving a processing speed

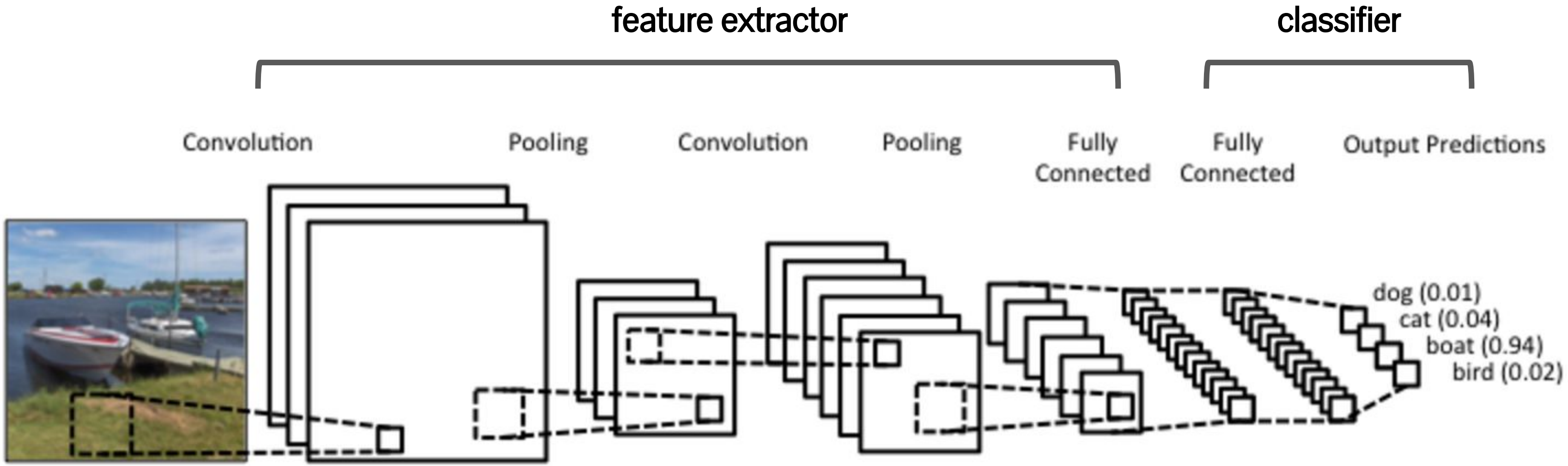
Progress in object detection



Slide credit: ross girshick

Convolutional Neural Network (CNN) for classification

- Operations (low-level)
  - Convolution
  - Non-linear activation (e.g. ReLU)
  - Sub-sampling (e.g. max-pooling)
  - Fully-connected Layer
- Operations (high-level)
  - Feature extractor
  - Classifier (e.g. the last layer)



Convolutional Neural Network (CNN) for Detection

Region extraction

Feature extraction

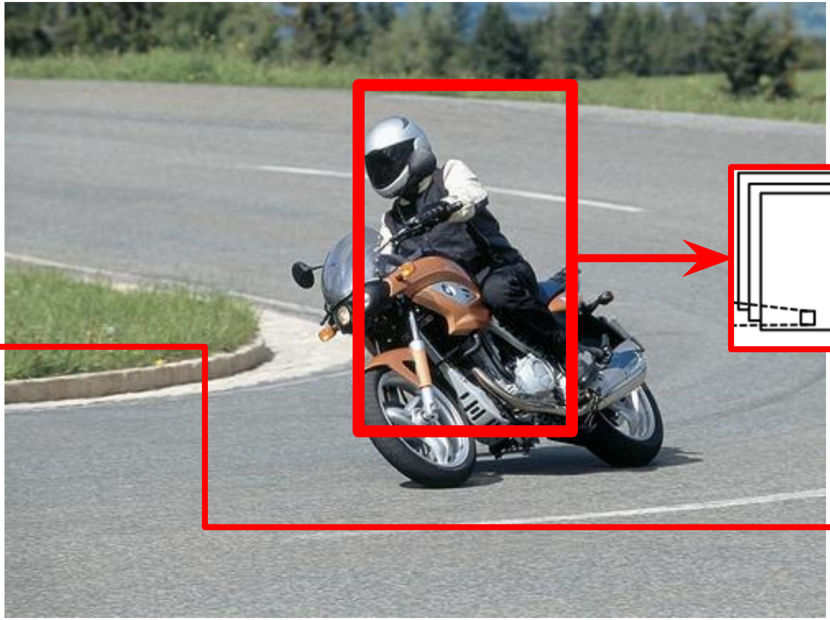
Classification

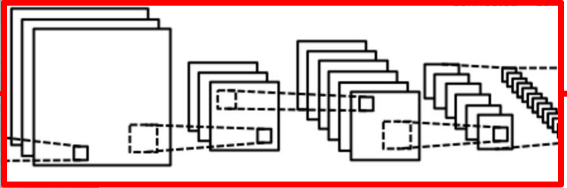
Post-processing

Replacing this with CNN for image classification?

Yeah! Why not!

...but why is it better?





CNN before
the last layer

Convolutional Neural Network (CNN) for Detection

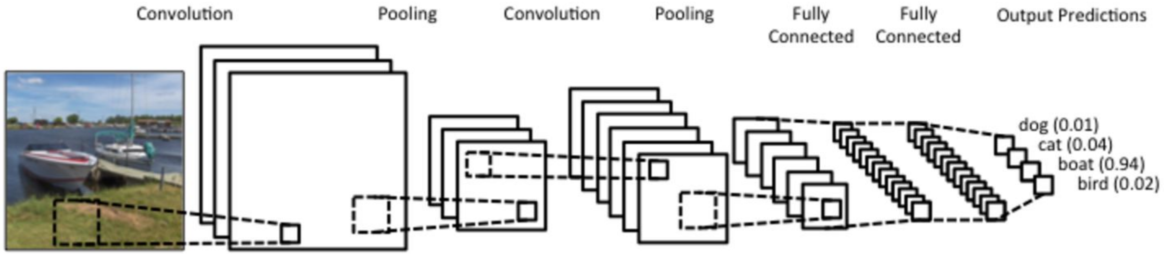
Region extraction

Feature extraction

Classification

Post-processing

The pre-trained network on a large-scale dataset provides
a powerful representation that are robust to various appearance variations!



Region-based CNN (R-CNN)

Girshick et al. CVPR14.

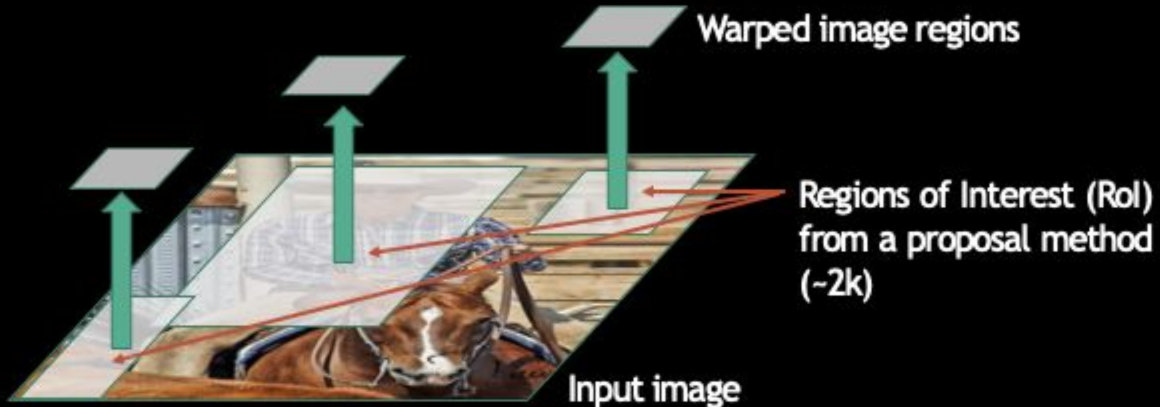
Slide credit: Ross Girshick

Region-based CNN (R-CNN)

Girshick et al. CVPR14.

Slide credit: Ross Girshick

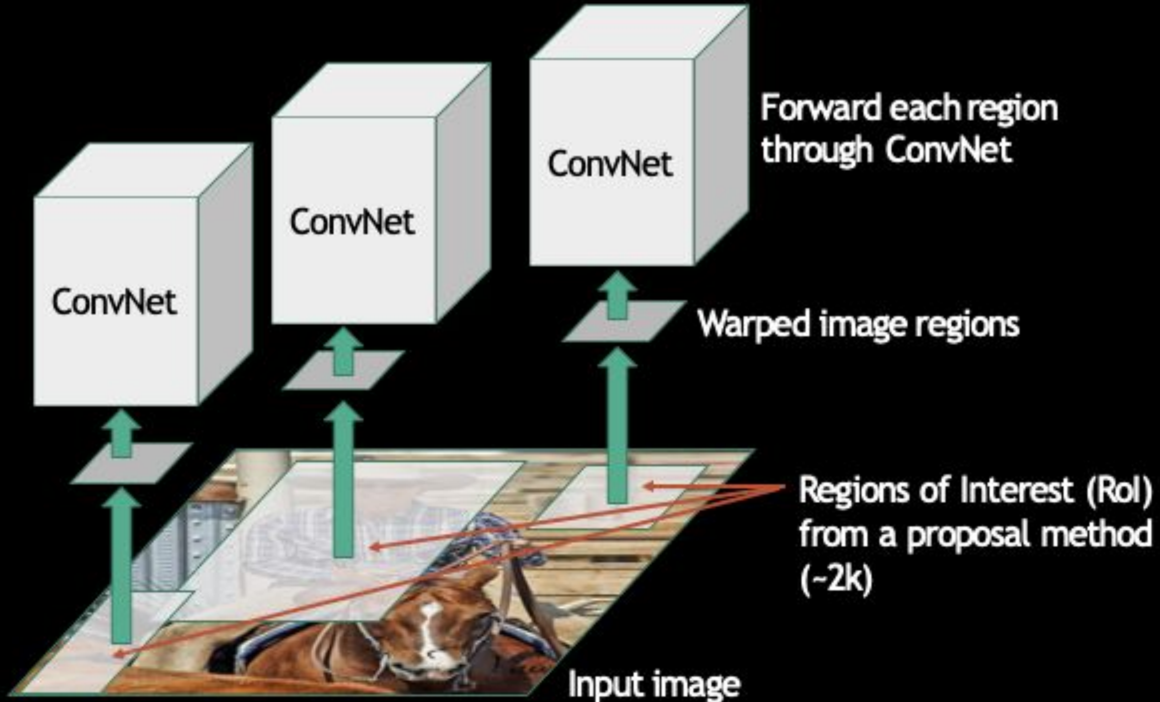
Region-based CNN (R-CNN)



Girshick et al. CVPR14.

Slide credit: Ross Girshick

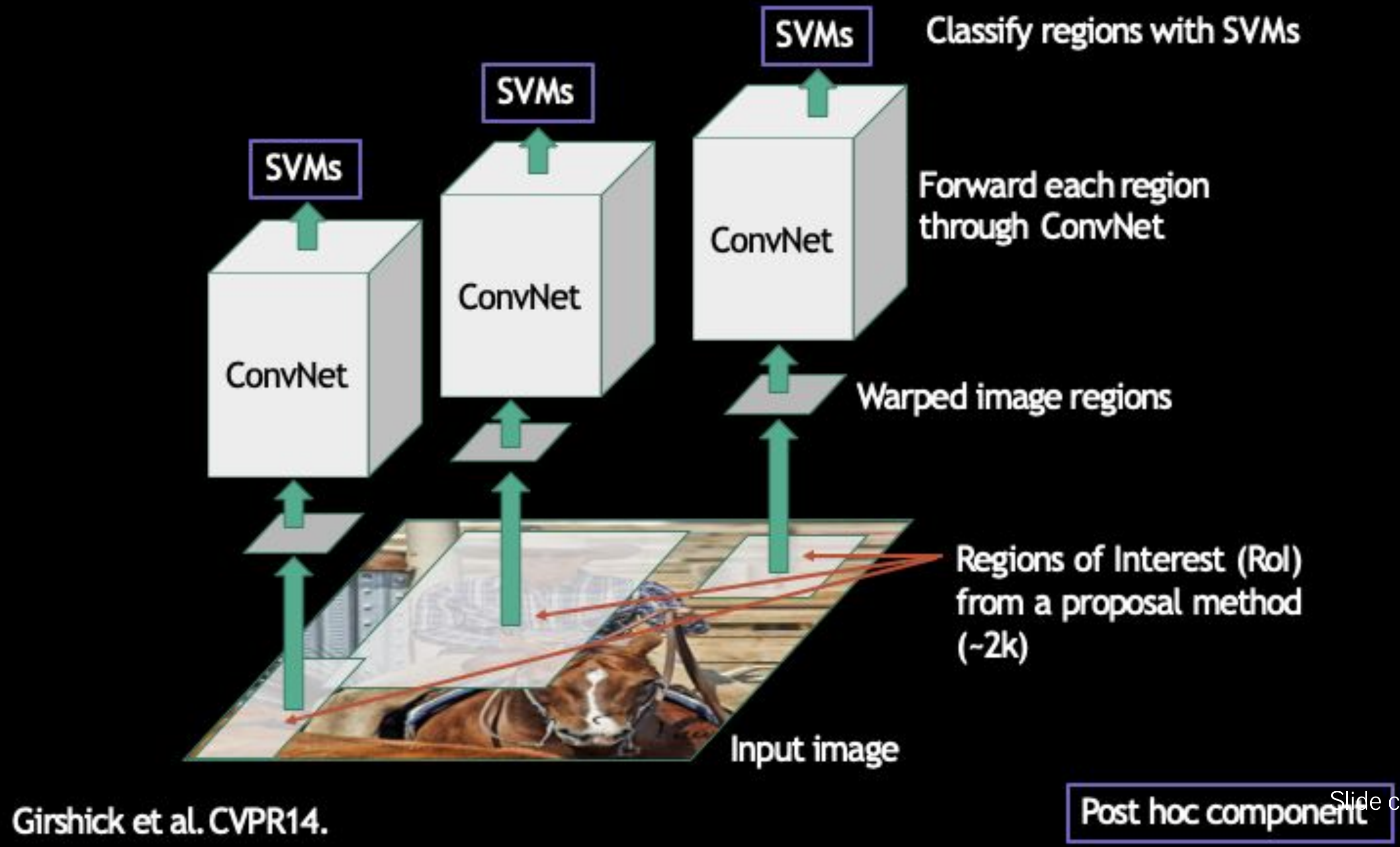
Region-based CNN (R-CNN)



Girshick et al. CVPR14.

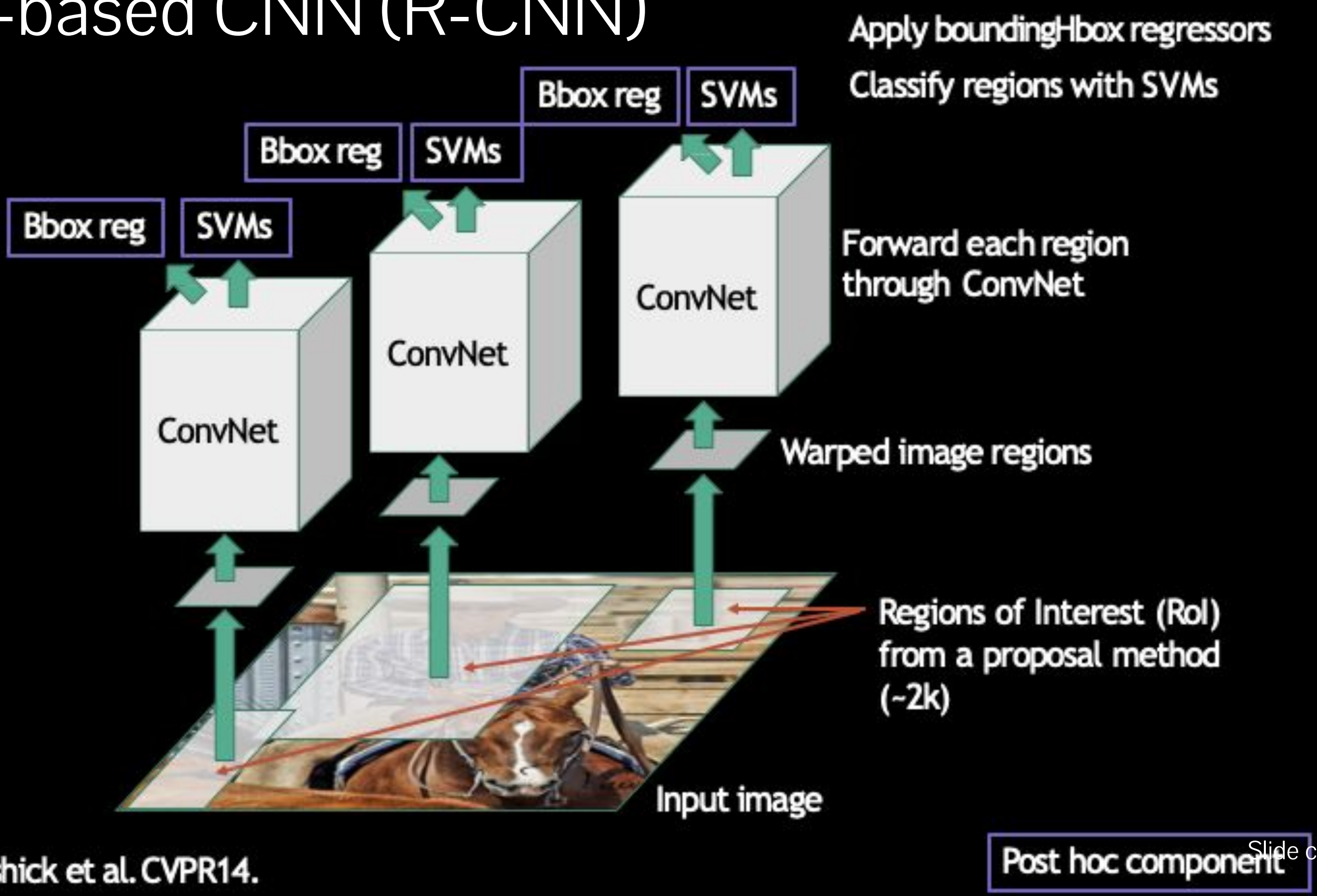
Slide credit: Ross Girshick

Region-based CNN (R-CNN)



Slide credit: Ross Girshick

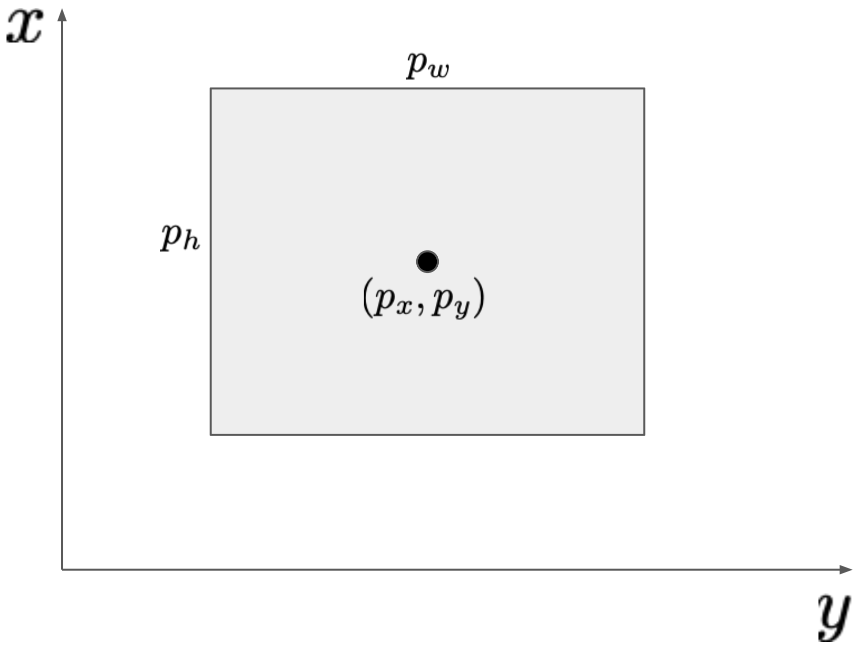
Region-based CNN (R-CNN)



Girshick et al. CVPR14.

Slide credit: Ross Girshick

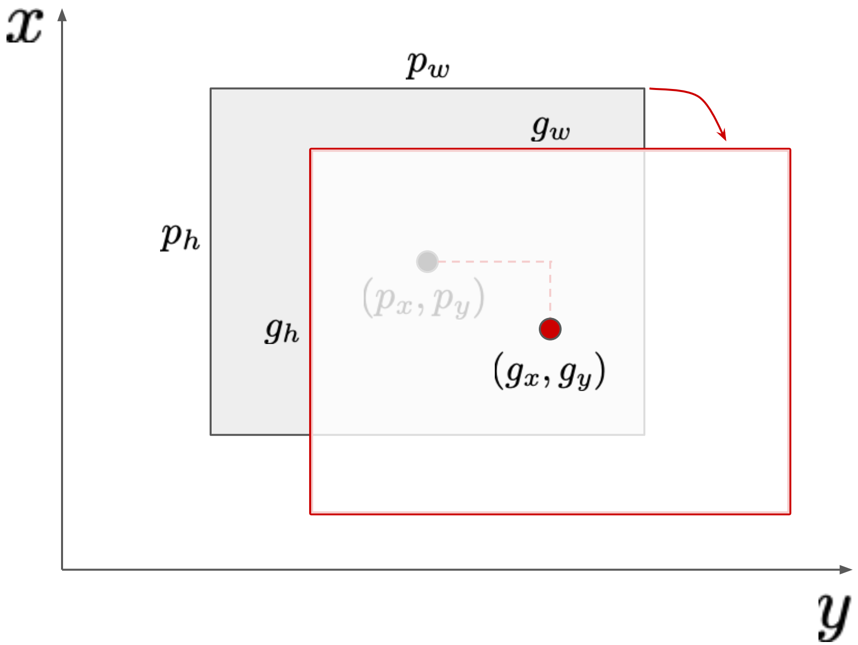
Bounding box Regression



Predicted bounding box

$$\mathbf{p} = (p_x, p_y, p_w, p_h)$$

Bounding box Regression



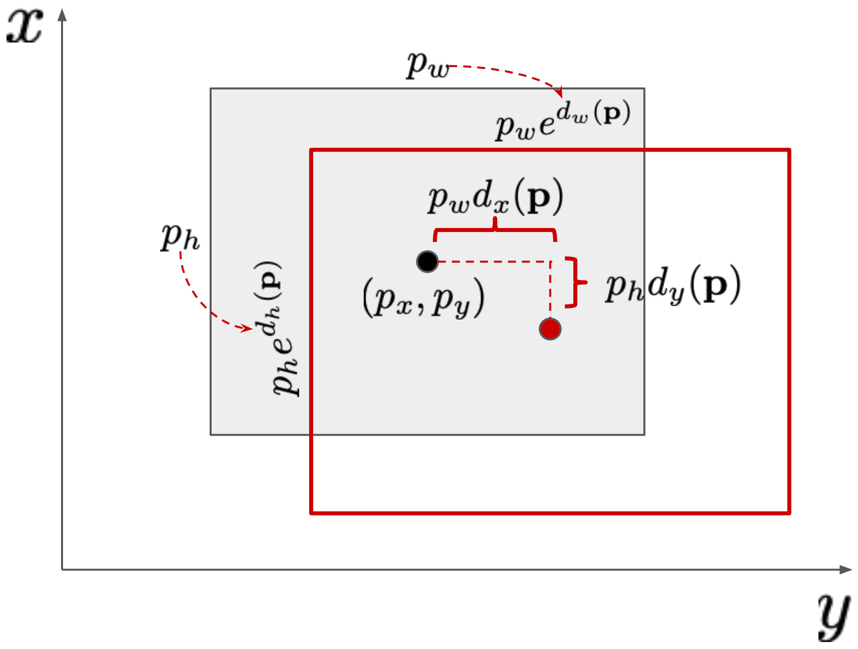
Predicted bounding box

$$\mathbf{p} = (p_x, p_y, p_w, p_h)$$

Target bounding box

$$\mathbf{g} = (g_x, g_y, g_w, g_h)$$

Bounding box Regression



Predicted bounding box

$$\mathbf{p} = (p_x, p_y, p_w, p_h)$$

Target bounding box

$$\mathbf{g} = (g_x, g_y, g_w, g_h)$$

Compute the displacement that modifies the predicted box to the target box

$$\mathbf{d}(\mathbf{p}) = (d_x(\mathbf{p}), d_y(\mathbf{p}), d_w(\mathbf{p}), d_h(\mathbf{p}))$$

- **Predicted box**

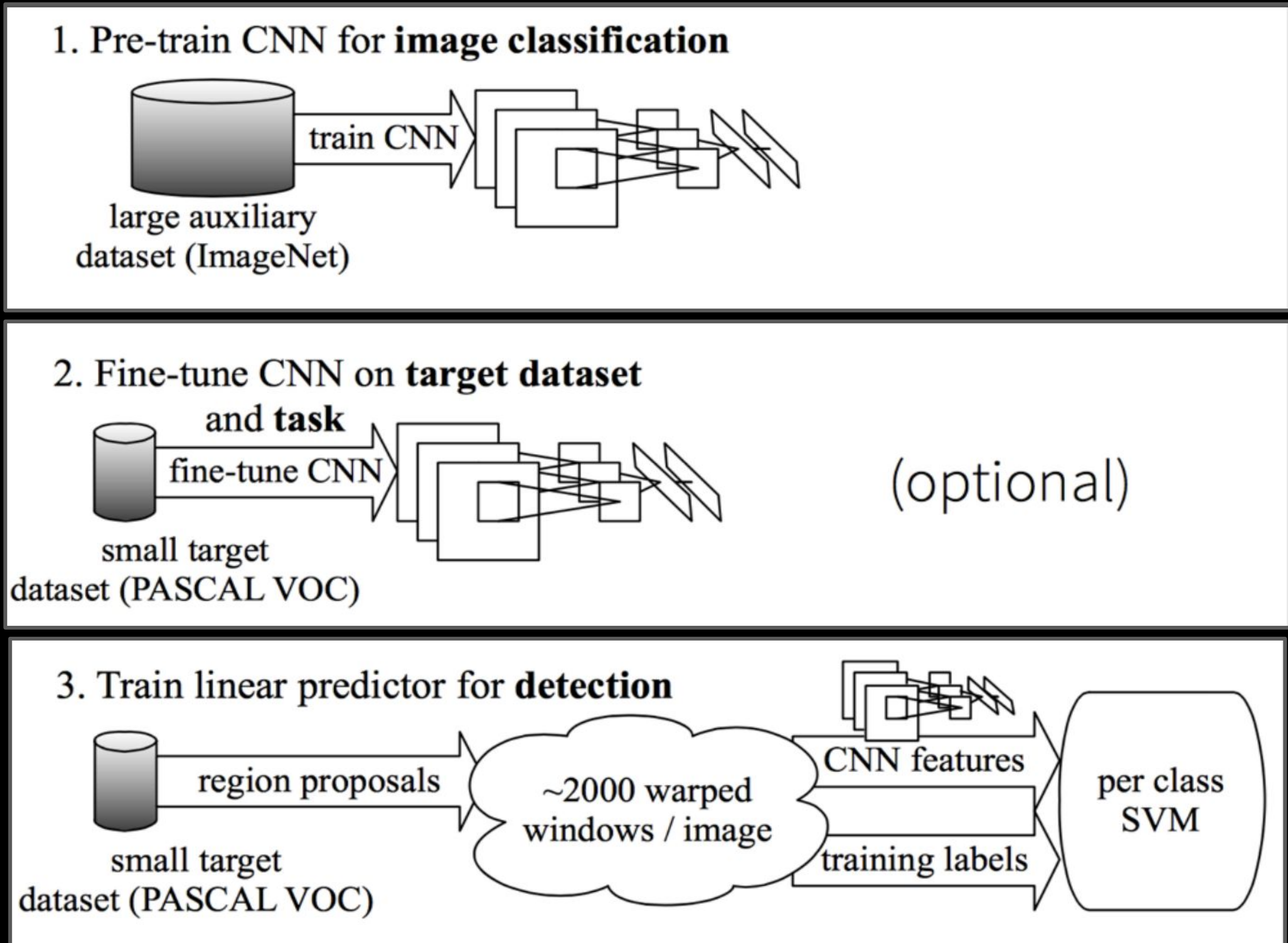
$$\begin{aligned} \hat{g}_x &= p_w d_x(\mathbf{p}) + p_x \\ \hat{g}_y &= p_h d_y(\mathbf{p}) + p_y \\ \hat{g}_w &= p_w \exp(d_w(\mathbf{p})) \\ \hat{g}_h &= p_h \exp(d_h(\mathbf{p})) \end{aligned}$$

- **Target displacement**

$$\begin{aligned} t_x &= (g_x - p_x)/p_w \\ t_y &= (g_y - p_y)/p_h \\ t_w &= \log(g_w/p_w) \\ t_h &= \log(g_h/p_h) \end{aligned}$$

The loss: $$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2$$

Training

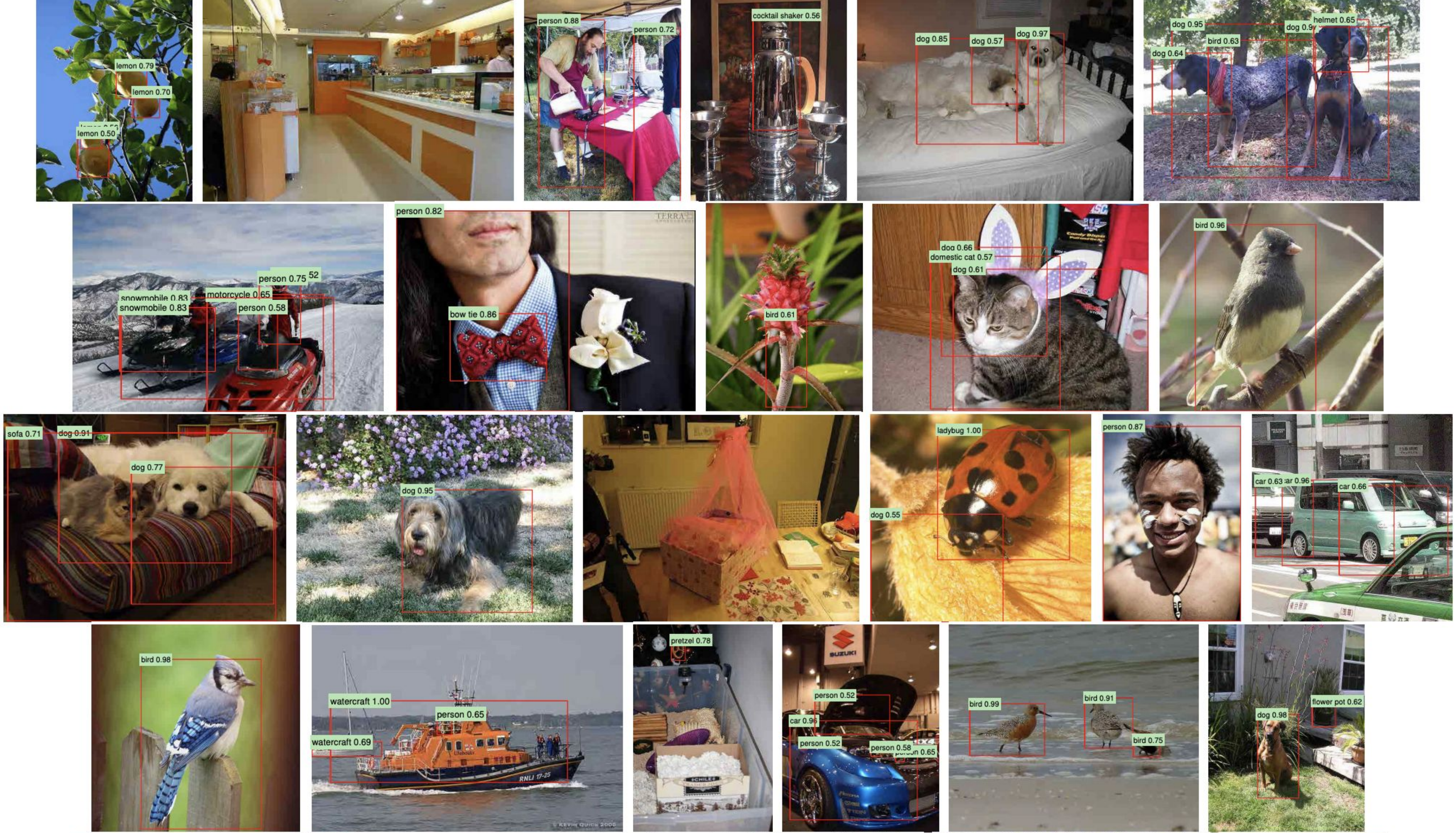


(optional)

Results

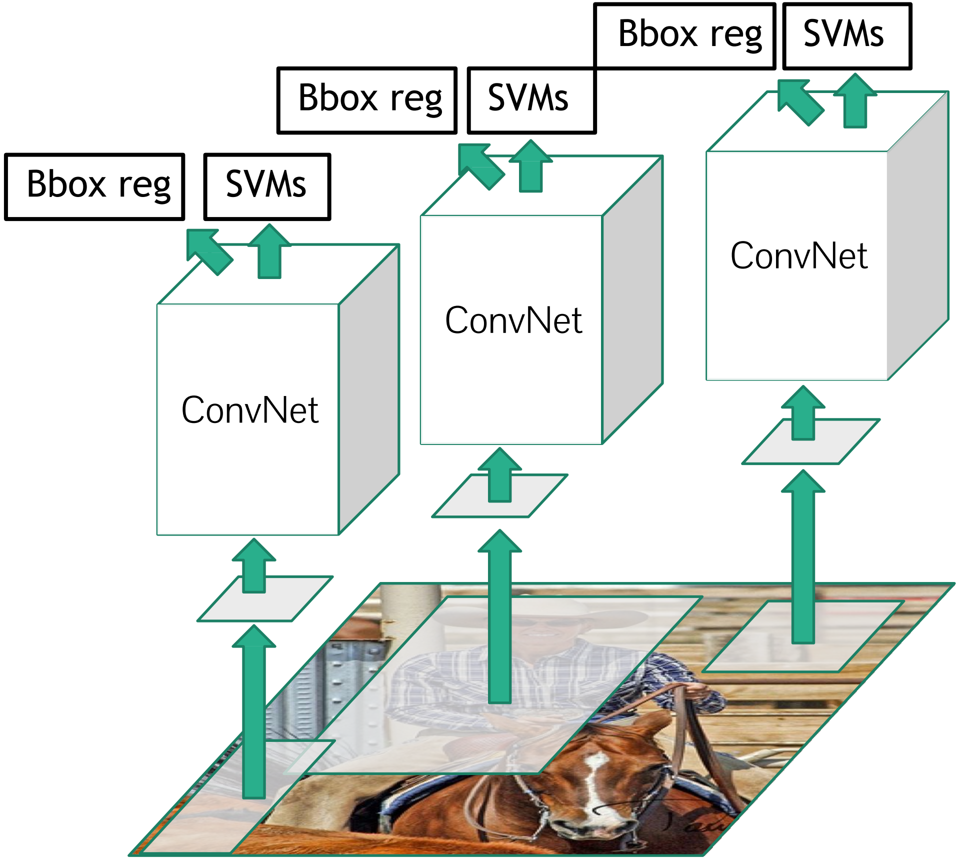
VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [23]	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [21]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [54]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [57]	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN T-Net	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN T-Net BB	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	53.7
R-CNN O-Net	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2	59.8
R-CNN O-Net BB	**79.3**	**72.4**	**63.1**	**44.0**	**44.4**	**64.6**	**66.3**	**84.9**	**38.8**	**67.3**	**48.4**	**82.3**	**75.0**	**76.7**	**65.7**	**35.8**	**66.2**	**54.8**	**69.1**	**58.8**	**62.9**

Slide credit: Ross Girshick



lemon 0.79
lemon 0.70
lemon 0.50
person 0.88
person 0.72
cocktail shaker 0.56
dog 0.85
dog 0.57
dog 0.97
dog 0.95
helmet 0.65
bird 0.63
dog 0.64
person 0.75
snowmobile 0.83
motorcycle 0.65
snowmobile 0.83
person 0.58
person 0.82
bow tie 0.86
bird 0.61
dog 0.66
domestic cat 0.57
dog 0.61
bird 0.96
sofa 0.71
dog 0.91
dog 0.77
dog 0.95
ladybug 1.00
dog 0.55
person 0.87
car 0.63
car 0.66
bird 0.98
watercraft 1.00
person 0.65
watercraft 0.69
pretzel 0.78
person 0.52
car 0.96
person 0.52
person 0.58
bird 0.99
bird 0.91
bird 0.75
dog 0.98
flower pot 0.62

Quiz: limitation of R-CNN?



Feature extraction
for every regions!

Summary: R-CNN

- Integrate CNN to detection pipeline
- Pipeline
  - Region proposal, CNN feature extraction, classification (SVM), post-processing
- Pros
  - Powerful representation by CNN
  - Robust to various appearance variations
- Cons
  - Slow processing time: CNN forward for every region proposal
  - Separate training of feature extractor (CNN) and classifier (SVM)

Improving R-CNN

- Improving the speed
  - How can we extract features more efficiently?
  - How can we potentially remove region proposal operation?
- Improving the training
  - How can we train all model components end-to-end?
- In the remaining class..
  - We will learn following=up works that improve R-CNN!
  - SPPNet, Fast & Faster R-CNN

SPP-Net (Spatial Pyramid Pooling)

He et al. ECCV14.

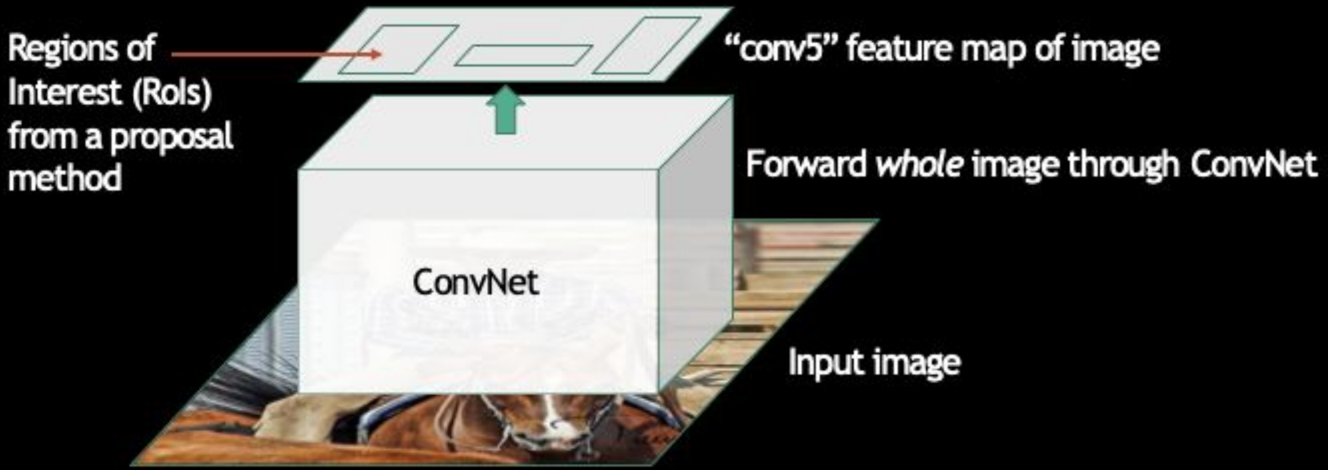
Slide credit: Ross Girshick

SPP-Net (Spatial Pyramid Pooling)

He et al. ECCV14.

Slide credit: Ross Girshick

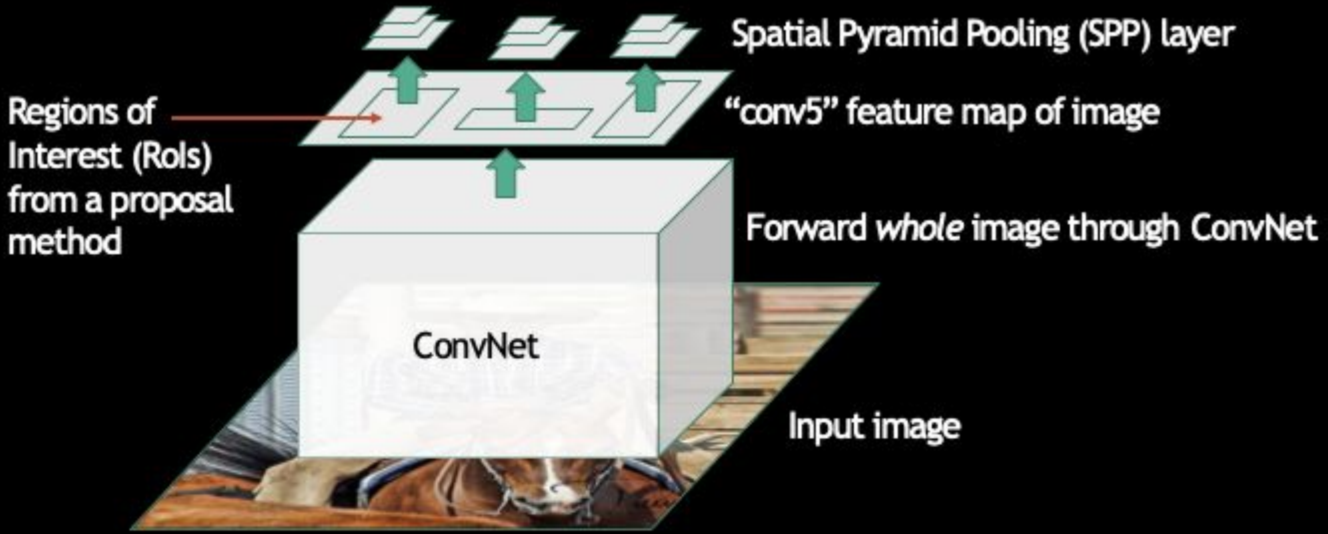
SPP-Net (Spatial Pyramid Pooling)



He et al. ECCV14.

Slide credit: Ross Girshick

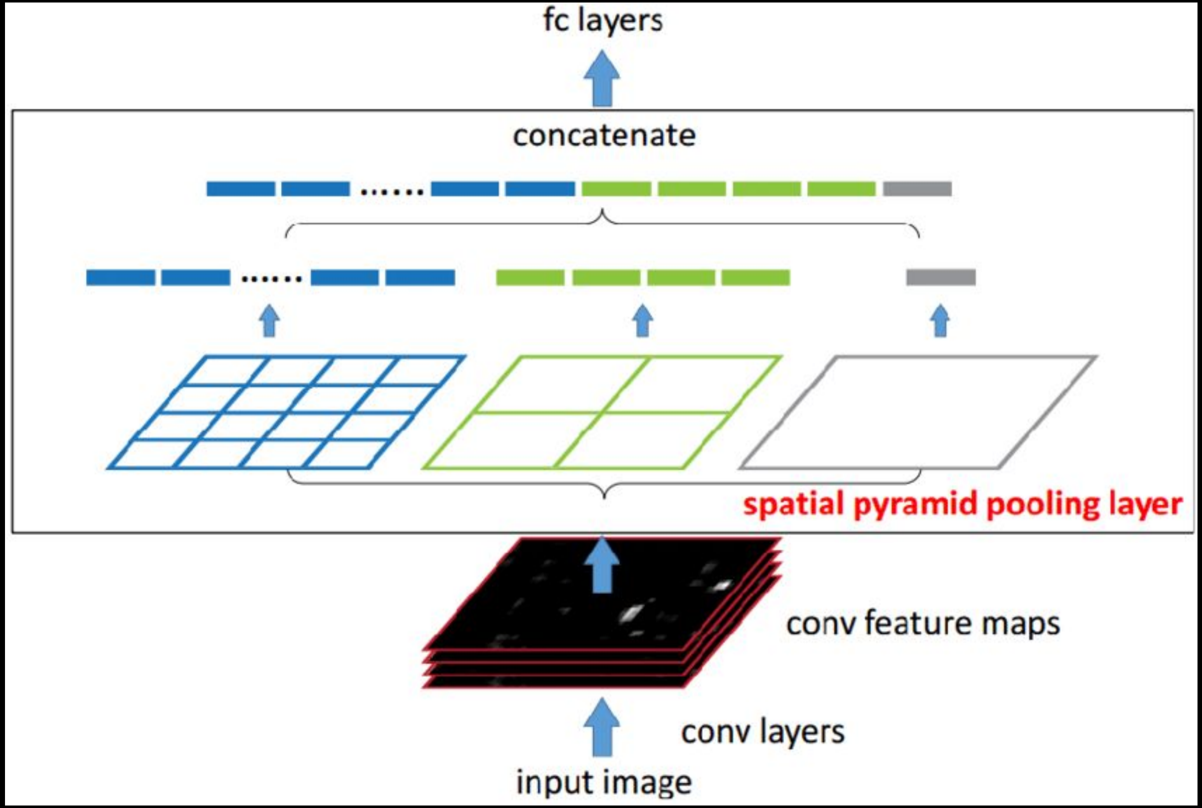
SPP-Net (Spatial Pyramid Pooling)



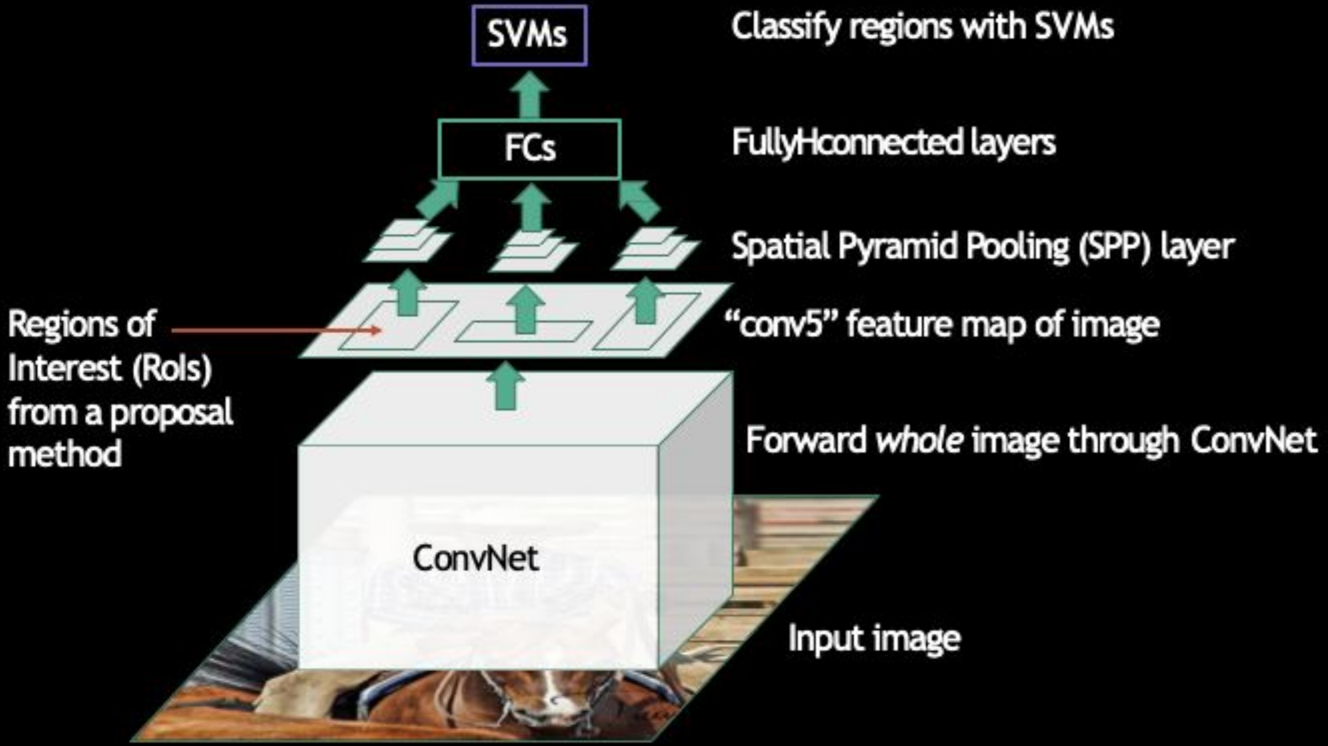
He et al. ECCV14.

Slide credit: Ross Girshick

Spatial Pyramid Pooling



SPP-Net (Spatial Pyramid Pooling)

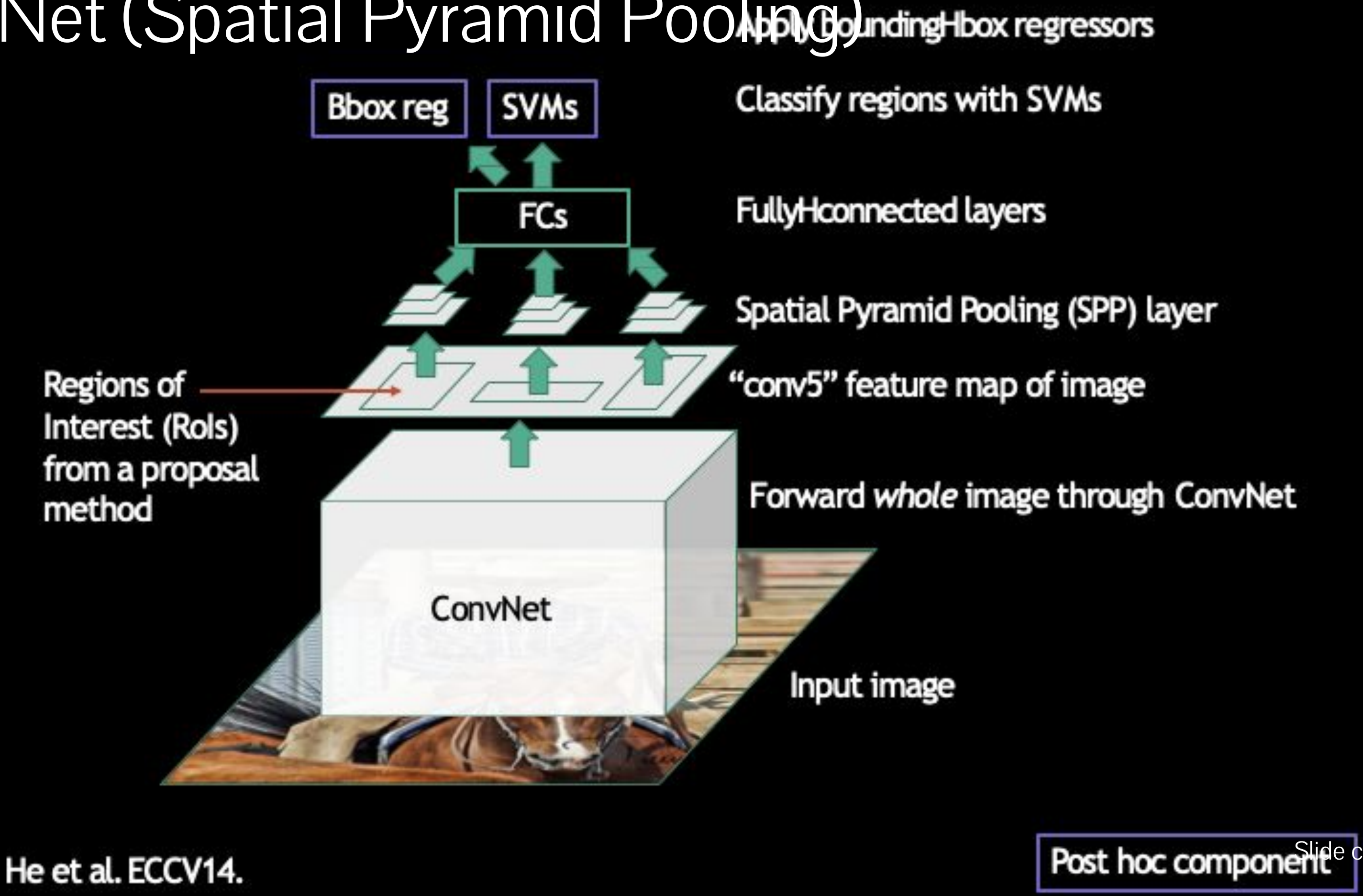


He et al. ECCV14.

Post hoc component

Slide credit: Ross Girshick

SPP-Net (Spatial Pyramid Pooling)



Slide credit: Ross Girshick

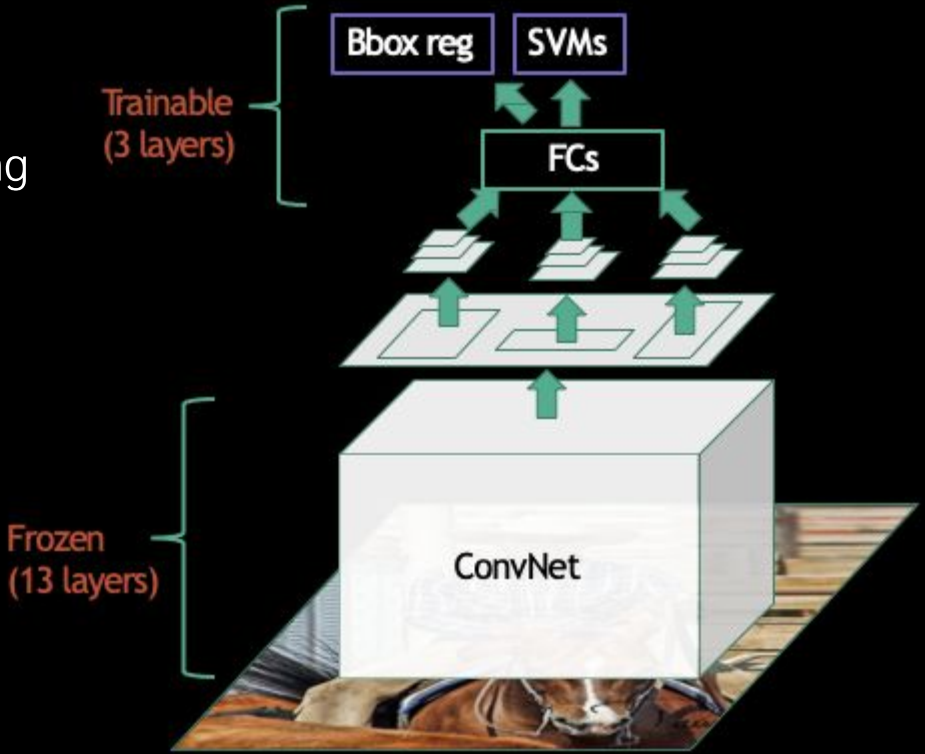
Limitation of SPP-Net

Inherits many problems in R-CNN

- Ad-hoc training object (SVM)
- Faster than R-CNN, but still slow training

Introduce another problem

- Convolutional layers are fixed!



Slide credit: Ross Girshick

Fast R'CNN

- Fast testHtime, like SPPHnet

Slide credit: Ross Girshick

Fast R'CNN

- Fast testHtime, like SPPHnet
- One network, trained in one stage

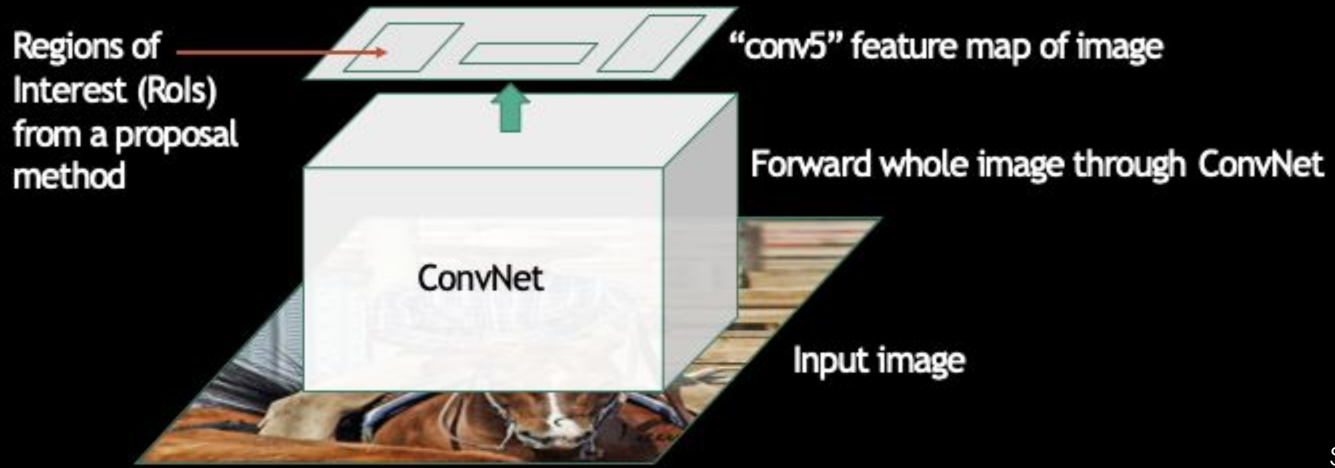
Slide credit: Ross Girshick

Fast R'CNN

- Fast testHtime, like SPPHnet
- One network, trained in one stage
- Higher mean average precision than slow RHCNN and SPPHnet

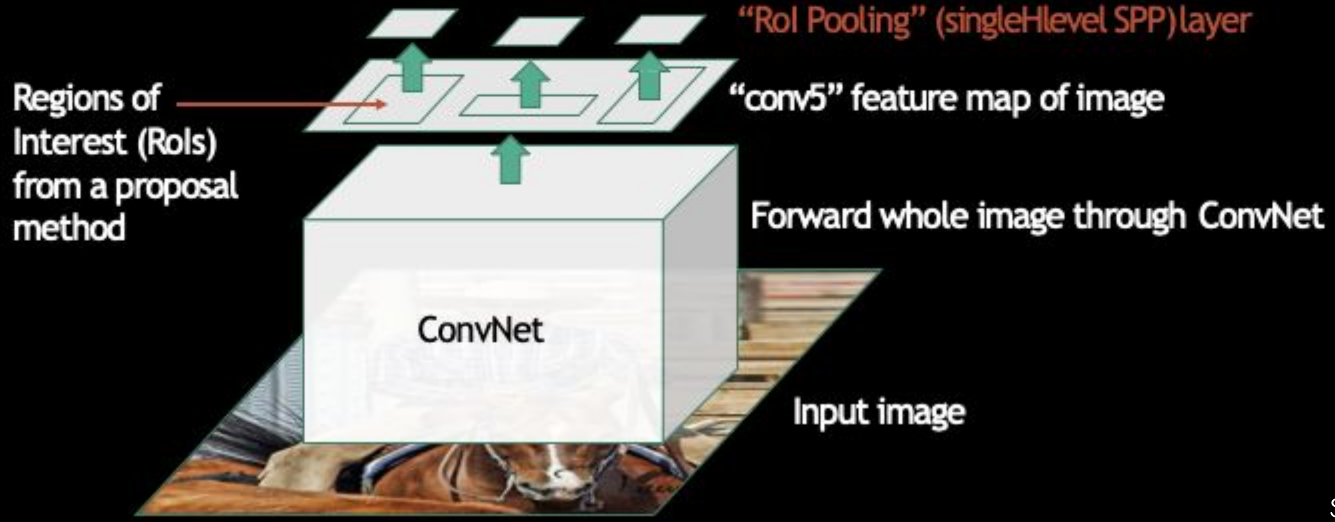
Slide credit: Ross Girshick

Fast R'CNN (test time)



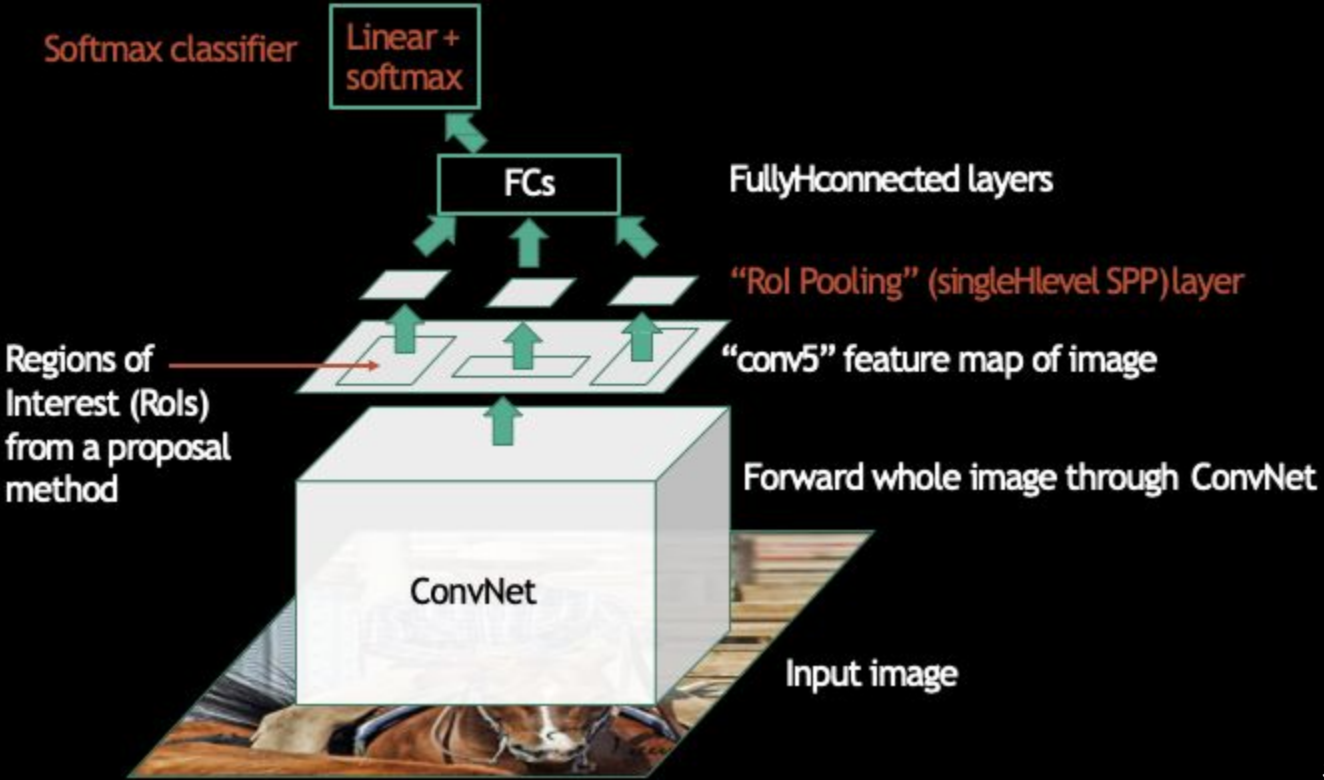
Slide credit: Ross Girshick

Fast R'CNN (test time)



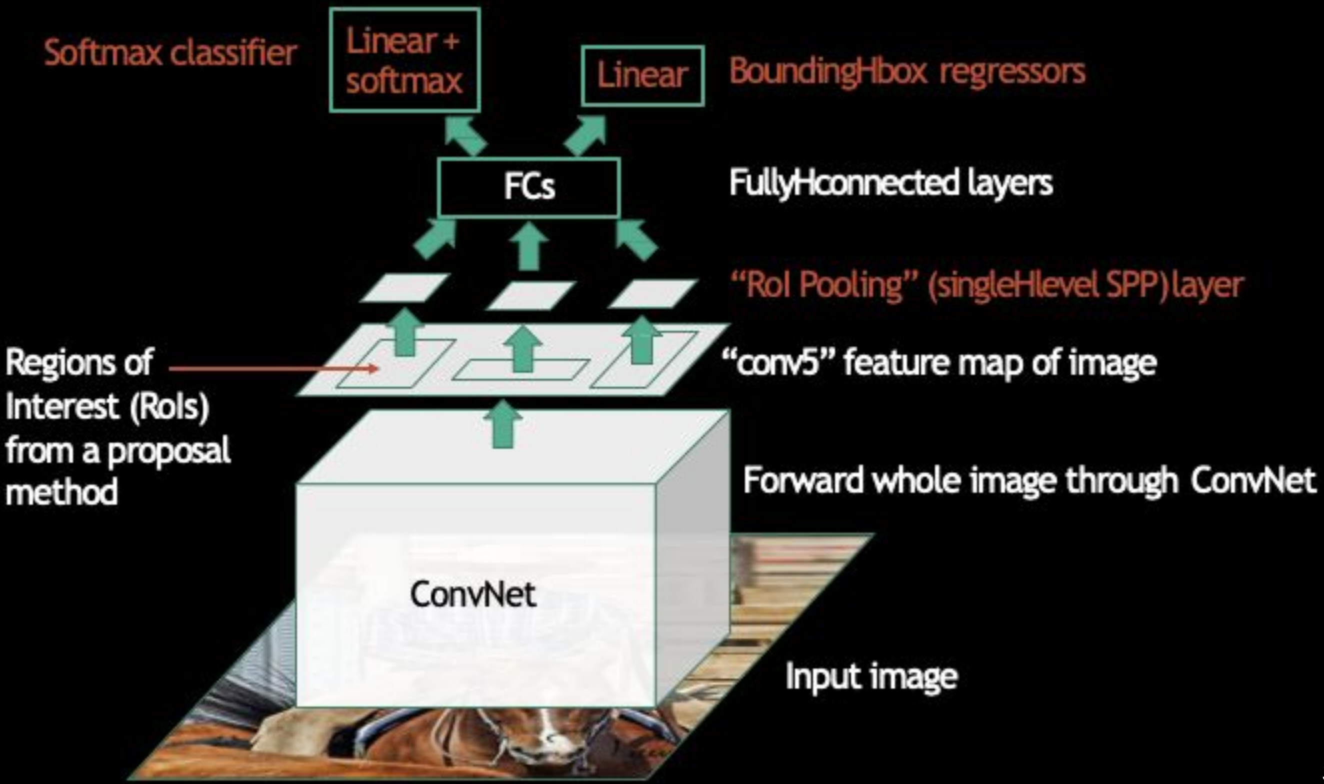
Slide credit: Ross Girshick

Fast R'CNN (test time)



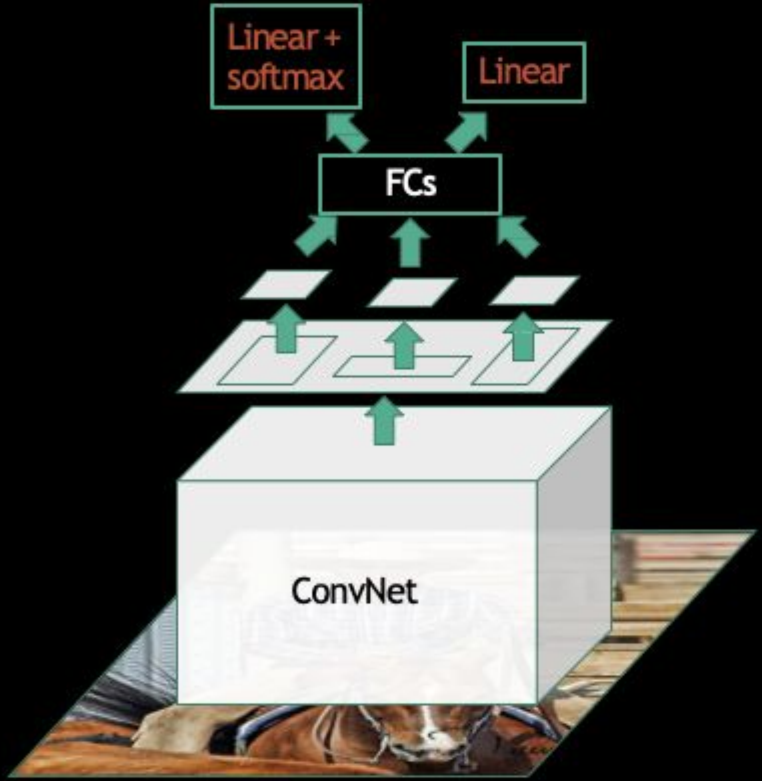
Slide credit: Ross Girshick

Fast R'CNN (test time)



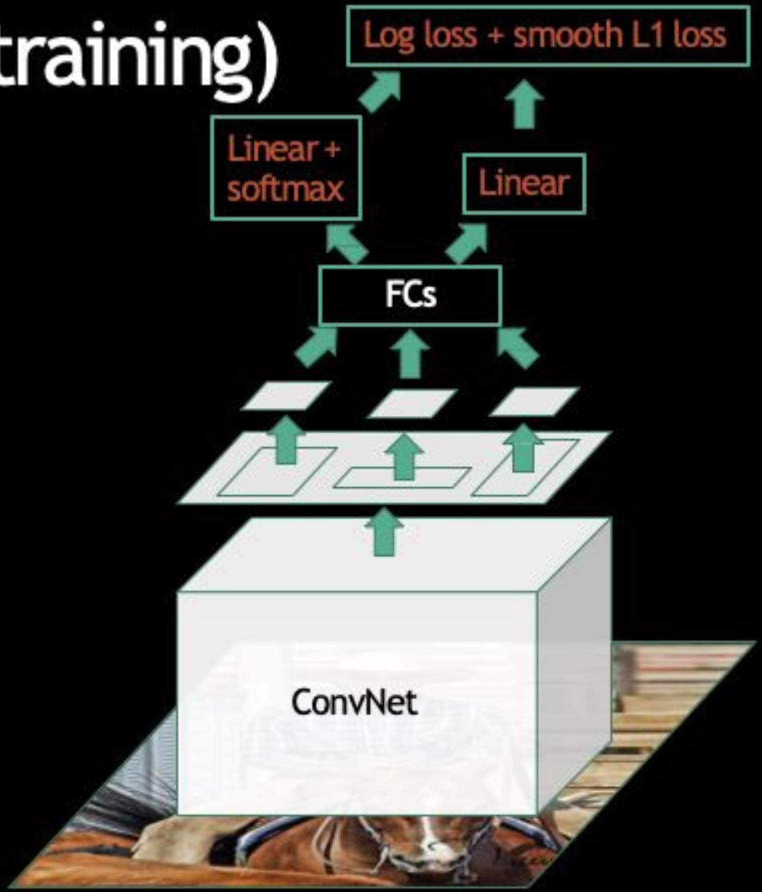
Slide credit: Ross Girshick

Fast R'CNN (training)



Slide credit: Ross Girshick

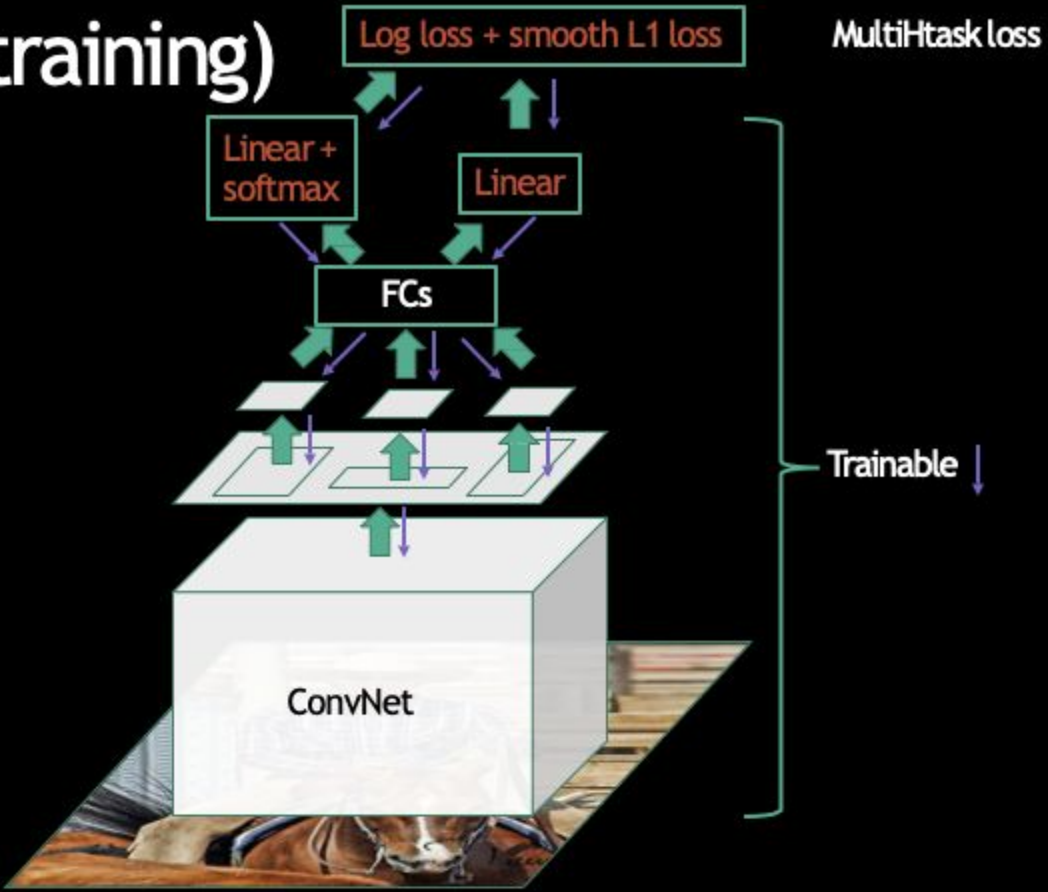
Fast R'CNN (training)



MultiHtask loss

Slide credit: Ross Girshick

Fast R'CNN (training)



Slide credit: Ross Girshick

Main results

	Fast R'CNN	R'CNN [1]	SPP'net [2]
Train time (h)	**9.5**	84	25
HSpeedup	**8.8x**	1x	3.4x
Test time / image	**0.32s**	47.0s	2.3s
Test speedup	**146x**	1x	20x
mAP	**66.9%**	66.0%	63.1%

Timings exclude object proposal time, which is equal for all methods.
All methods use VGG16 from Simonyan and Zisserman.

[1] Girshick et al. CVPR14.
[2] He et al. ECCV14.

Slide credit: Ross Girshick

Main results

	Fast R'CNN	R'CNN [1]	SPP'net [2]
Train time (h)	**9.5**	84	25
HSpeedup	**8.8x**	1x	3.4x
Test time / image	**0.32s**	47.0s	2.3s
Test speedup	**146x**	1x	20x
mAP	**66.9%**	66.0%	63.1%

Timings exclude object proposal time, which is equal for all methods.
All methods use VGG16 from Simonyan and Zisserman.

[1] Girshick et al. CVPR14.
[2] He et al. ECCV14.

Slide credit: Ross Girshick

End'to'end training matters

	Fast R'CNN (VGG16)		
FineHtune layers	≥fc6	≥conv3_1	≥conv2_1
VOC07 mAP	61.4%	66.9%	67.2%
Test time per image	0.32s	0.32s	0.32s
			1.4x slower training

Slide credit: Ross Girshick

Multi'task training helps

	Fast R'CNN (VGG16)			
MultiHtask training?		Y		Y
StageHwise training?			Y	
TestHtime bbox reg.			Y	Y
VOC07 mAP	62.6%	63.4%	64.0%	66.9%

Slide credit: Ross Girshick

Multi'task training helps

	Fast R'CNN (VGG16)			
MultiHtask training?		Y		Y
StageHwise training?			Y	
TestHtime bbox reg.			Y	Y
VOC07 mAP	62.6%	63.4%	64.0%	66.9%

Trained without
a bbox regressor

Slide credit: Ross Girshick

Multi'task training helps

	Fast R'CNN (VGG16)			
MultiHtask training?		Y		Y
StageHwise training?			Y	
TestHtime bbox reg.			Y	Y
VOC07 mAP	62.6%	63.4%	64.0%	66.9%

Trained with
a bbox regressor,
but it's disabled at
test time

Slide credit: Ross Girshick

Multi'task training helps

	Fast R'CNN (VGG16)			
MultiHtask training?		Y		Y
StageHwise training?			Y	
TestHtime bbox reg.			Y	Y
VOC07 mAP	62.6%	63.4%	64.0%	66.9%

Post hoc bbox
regressor, used
at test time

Slide credit: Ross Girshick

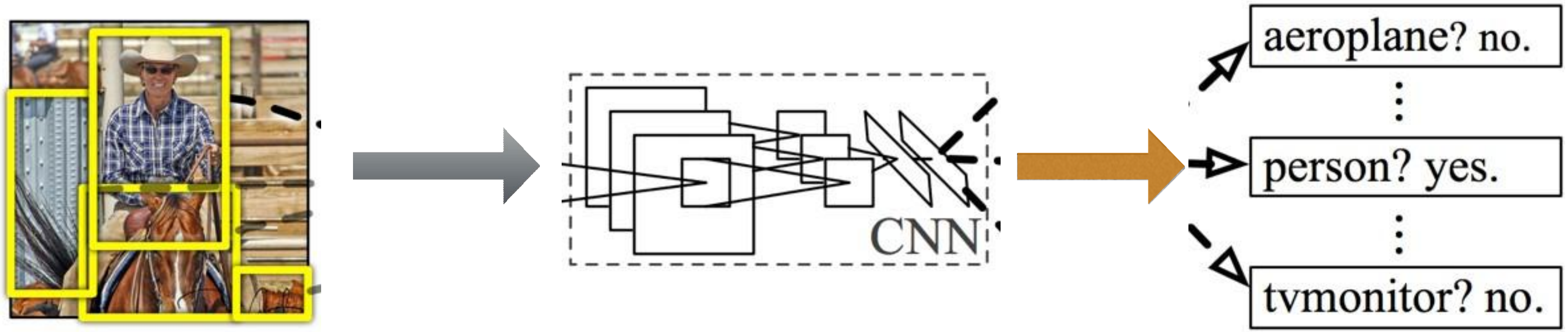
Multi'task training helps

	Fast R'CNN (VGG16)			
MultiHtask training?		Y		Y
StageHwise training?			Y	
TestHtime bbox reg.			Y	Y
VOC07 mAP	62.6%	63.4%	64.0%	66.9%

MultiHtask objective,
using bbox regressors
at test time

Slide credit: Ross Girshick

Summary: SPPNet, Fast-RCNN

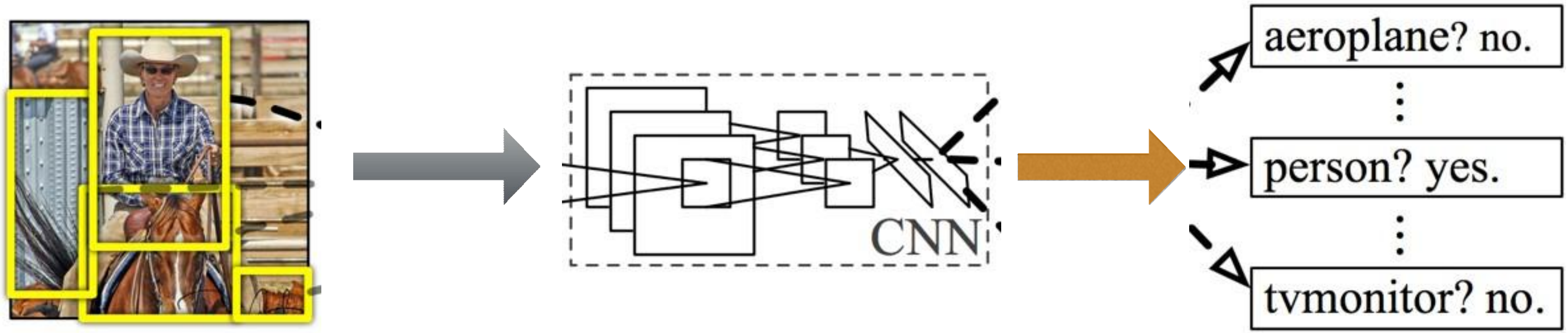


Region extraction

Feature extraction

Classification

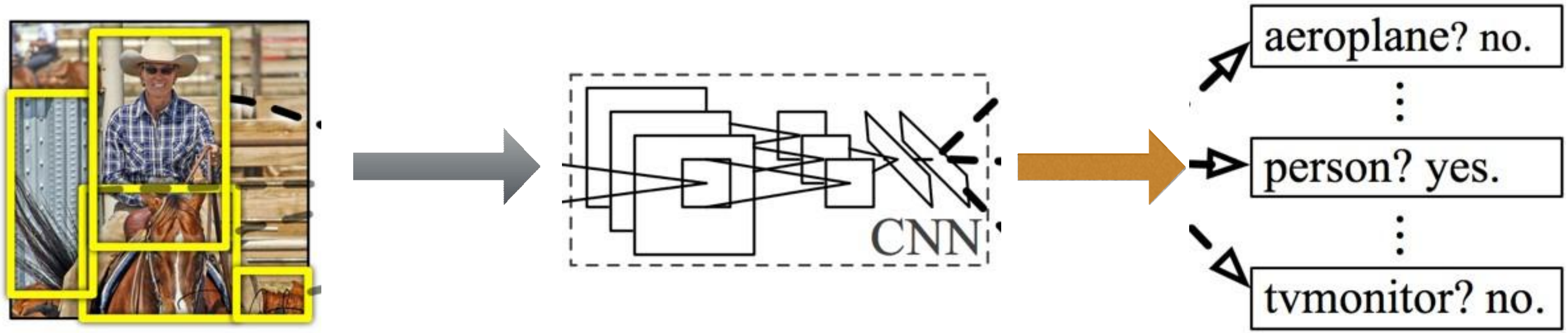
Summary: SPPNet, Fast-RCNN



Region extraction **Feature extraction** **Classification**

SPPNet: reduce the computation
by sharing the feature extraction

Summary: SPPNet, Fast-RCNN



Region extraction

Feature extraction

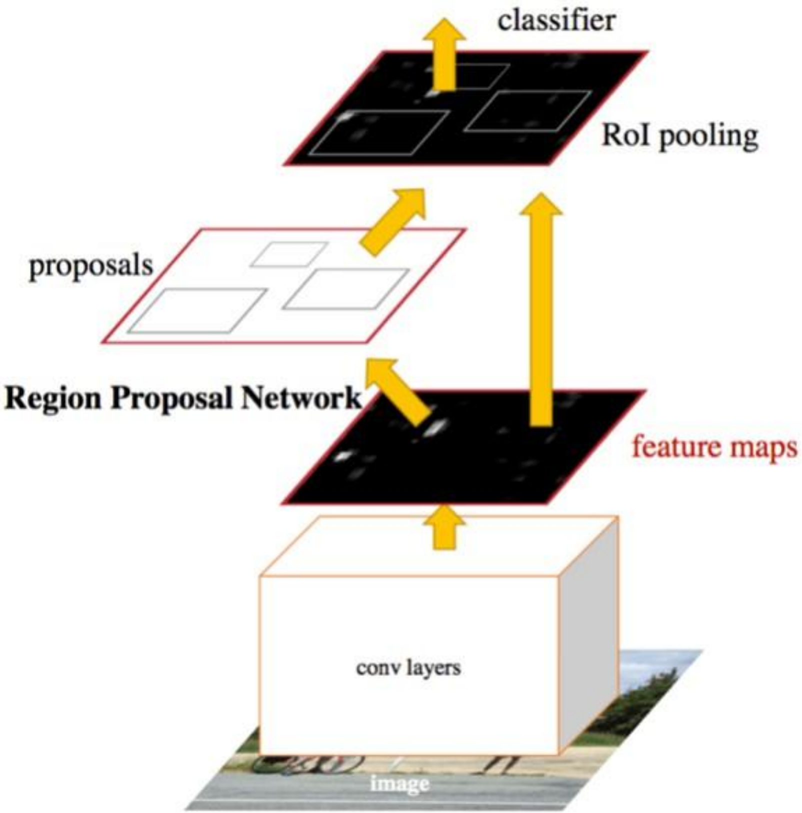
Classification

SPPNet: reduce the computation by sharing the feature extraction

Fast RCNN: train classifier and Bounding box regression with CNN in an end=to-end manner

Faster R-CNN

Fast R-CNN +
Region Proposal
Network (RPN)

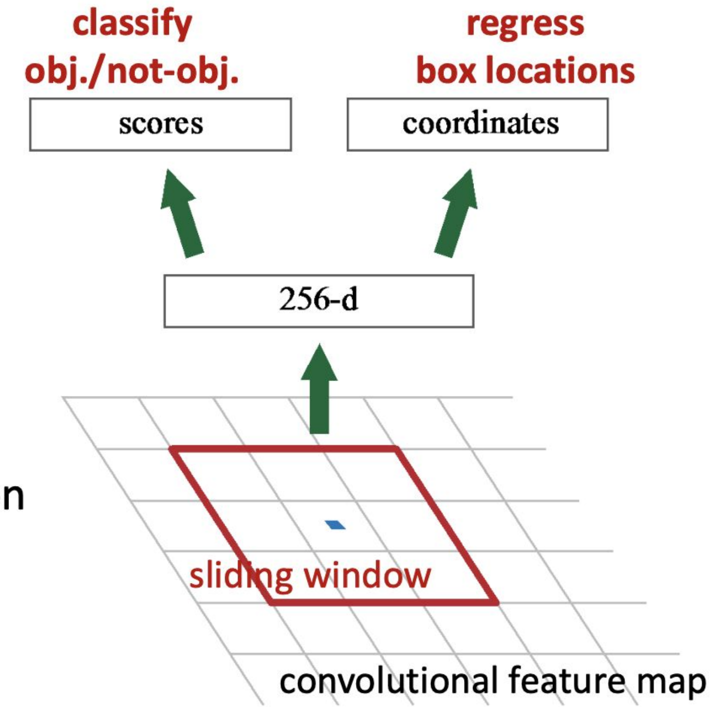


Slide credit: Kaiming He

Region Proposal Network

- Slide a small window on the feature map
- Build a small network for:
  - classifying object or not-object, and
  - regressing bbox locations

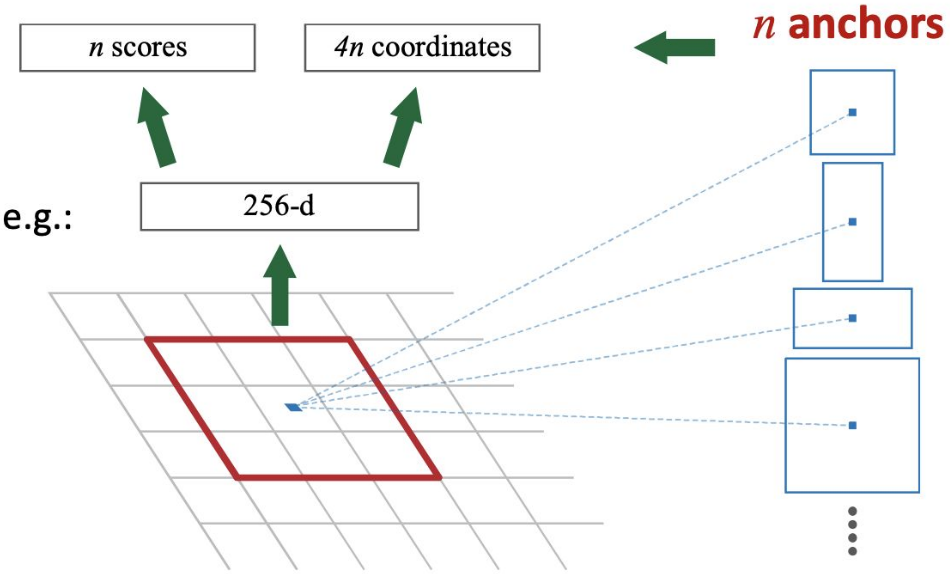
- Position of the sliding window provides localization information with reference to the image
- Box regression provides finer localization information with reference to this sliding window



Slide credit: Kaiming He

Anchors as references

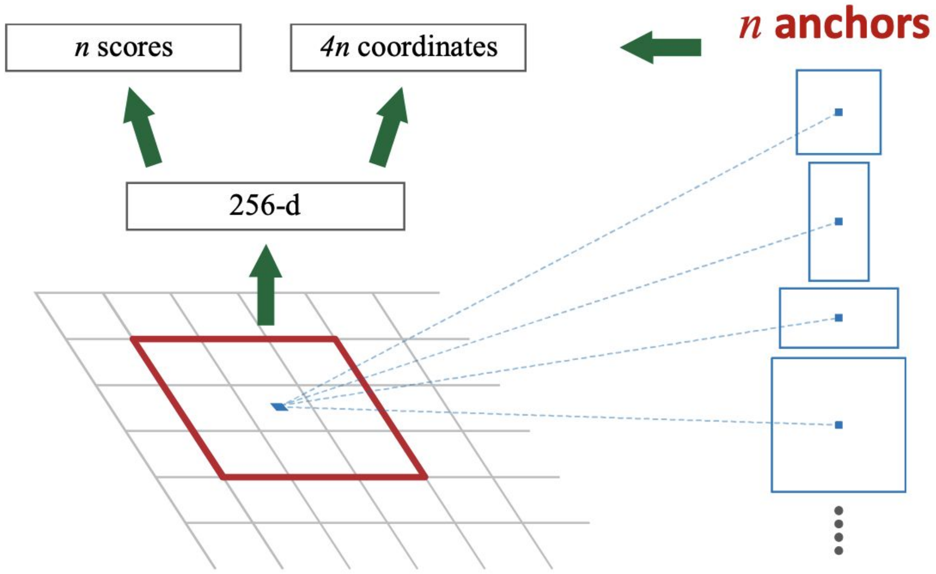
- **Anchors**: pre-defined reference boxes
  - Box regression is with reference to anchors:
    regressing an anchor box to a ground-truth box
  - Object probability is with reference to anchors, e.g.:
    - anchors as positive samples: if IoU > 0.7 or IoU is max
    - anchors as negative samples: if IoU < 0.3



Slide credit: Kaiming He

Anchors as references

- **Anchors**: pre-defined reference boxes
- Translation-invariant anchors:
  - the same set of anchors are used at each sliding position
  - the same prediction functions (with reference to the sliding window) are used
  - a translated object will have a translated prediction

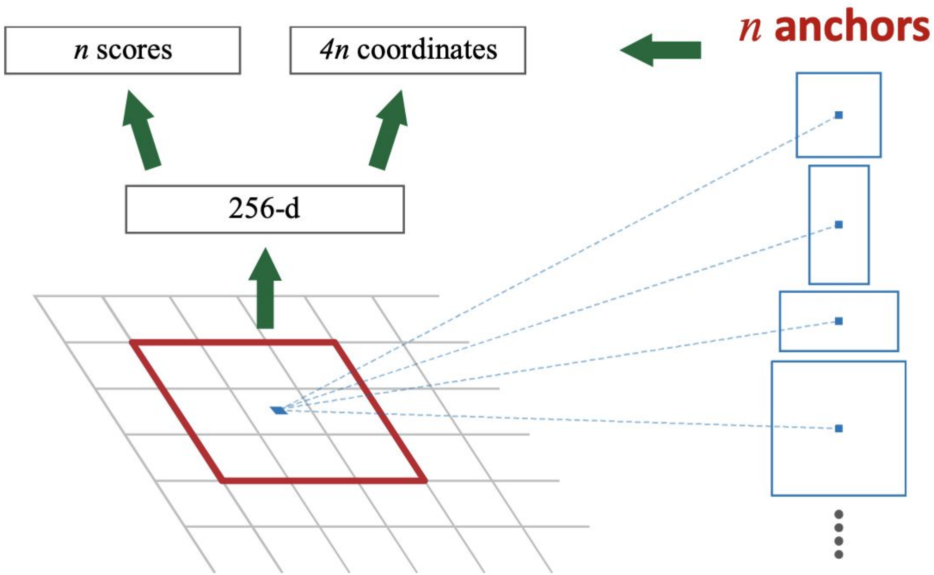


Slide credit: Kaiming He

Anchors as references

- **Anchors**: pre-defined reference boxes

- Multi-scale/size anchors:
  - multiple anchors are used at each position:
    e.g., 3 scales (128^2, 256^2, 512^2) and 3 aspect ratios (2:1, 1:1, 1:2) yield 9 anchors
  - each anchor has its own prediction function
  - single-scale features, multi-scale predictions



Slide credit: Kaiming He

Faster R-CNN: training

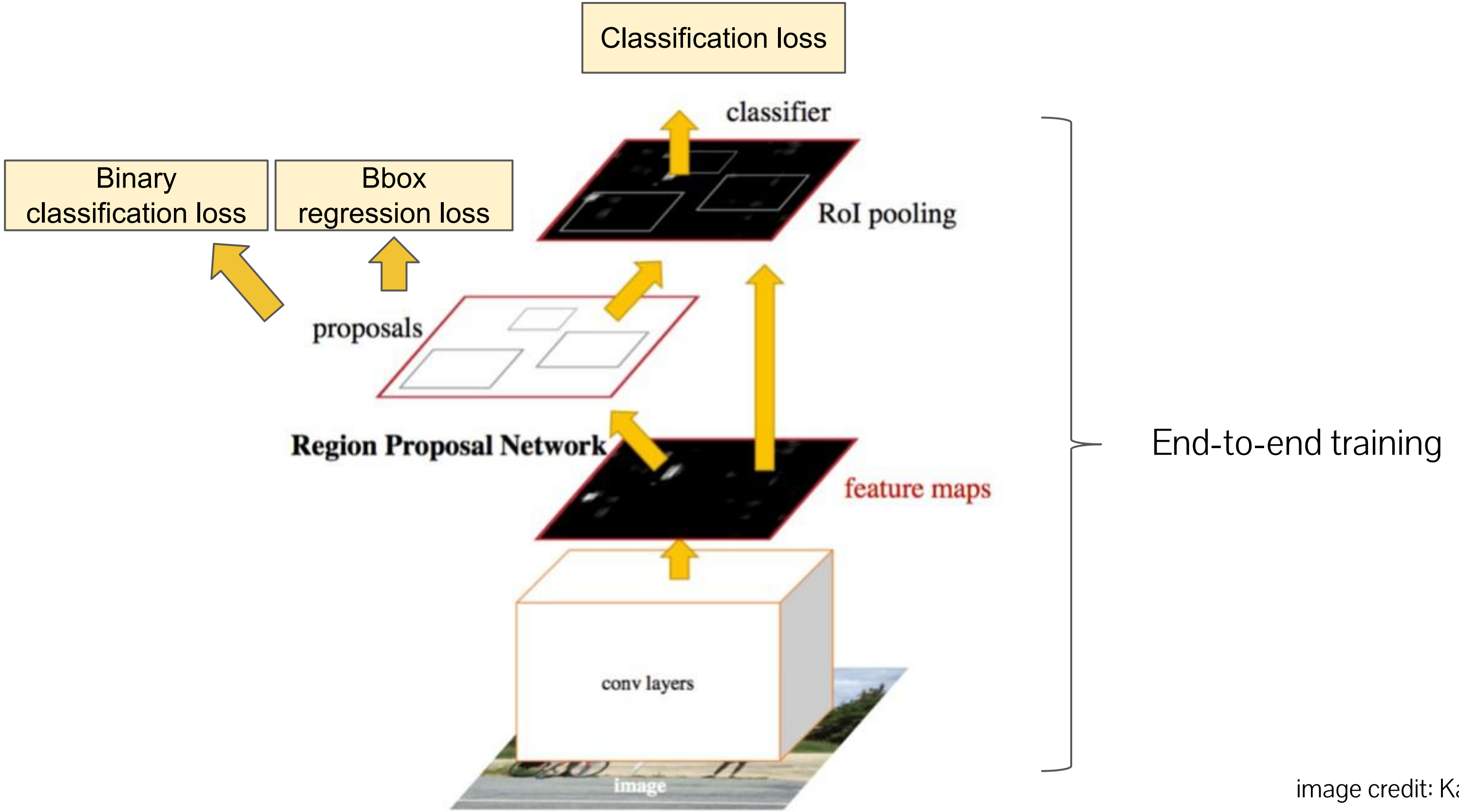
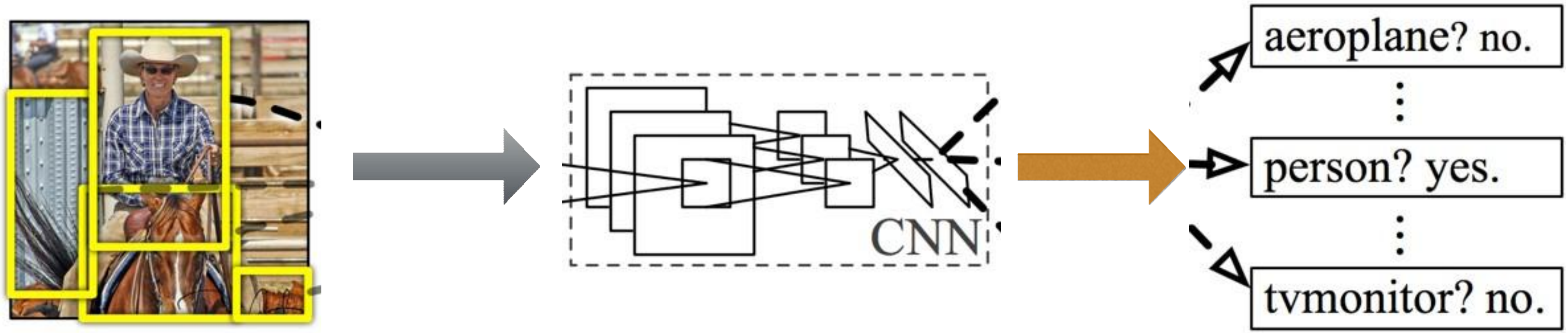


image credit: Kaiming He

Comparisons of accuracy vs. speed

	Pascal 2007 mAP	**Speed**	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img

Summary



Region extraction

Feature extraction

Classification

Faster RCNN: learn to extract region proposals

SPPNet: reduce the computation by sharing the feature extraction

Fast RCNN: train classifier and Bounding box regression with CNN in an end=to-end manner