R6.A.07 - Artificial Intelligence

Deep learning-based object detection

Minh-Tan Pham

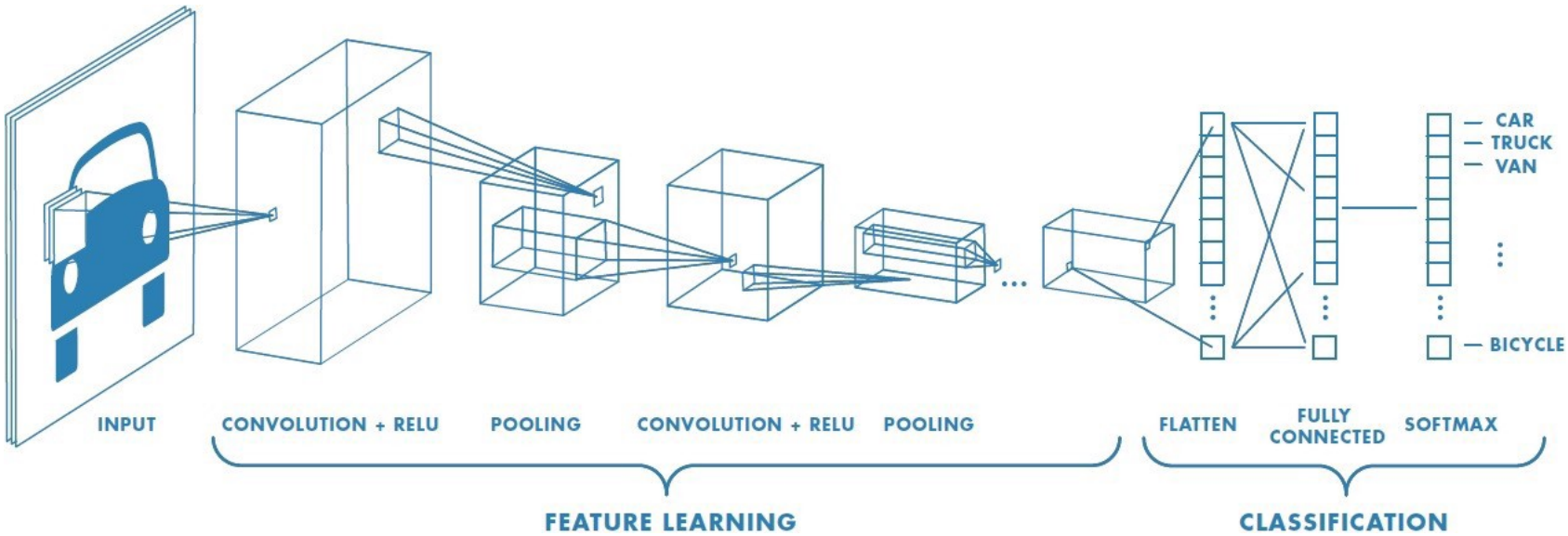
IUT Vannes, Université Bretagne-Sud, Vannes, France

minh-tan.pham@univ-ubs.fr

Table of contents

Table of contents

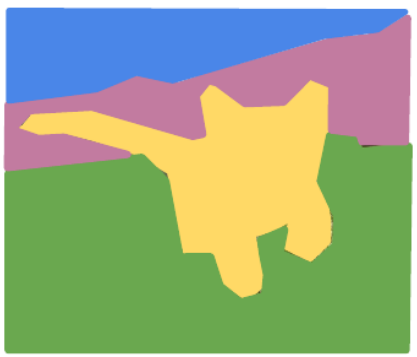
Introduction So far: image classification using CNN



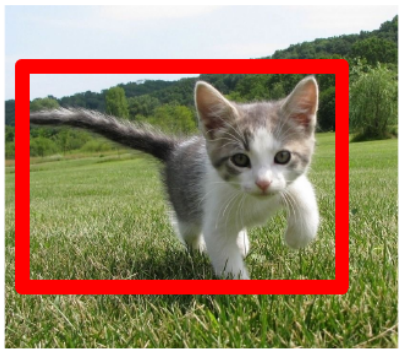
Maybe some remarks? questions?

Other computer vision tasks:

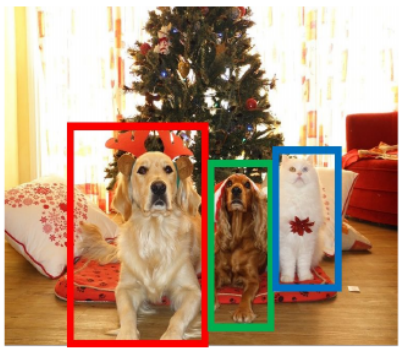
Semantic Segmentation



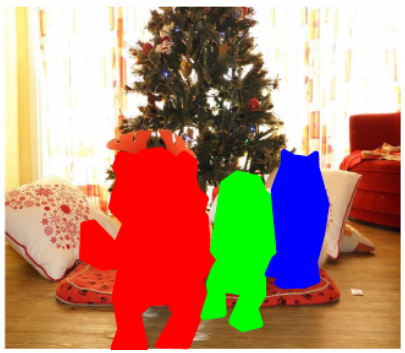
Classification + Localization



Object Detection



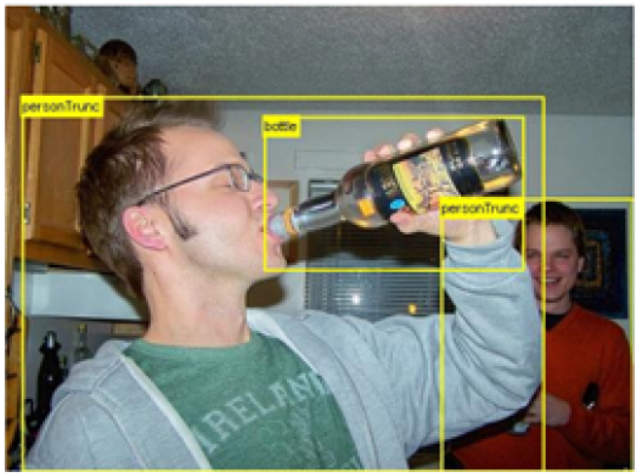
Instance Segmentation

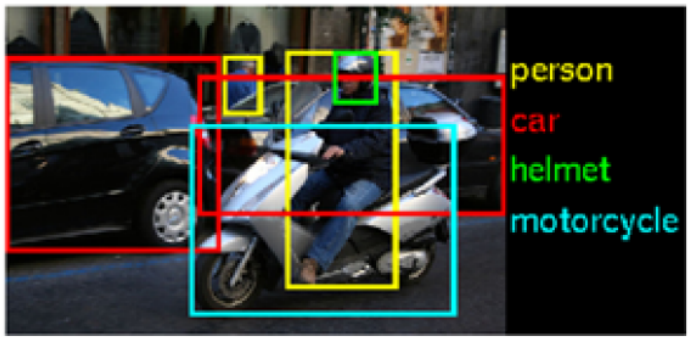


Applications of object detection:

- Computer vision:
  - facial detection and recognition
  - industrial quality check
  - autonomous driving cars
  - people counting and tracking, etc.
- Remote sensing:
  - wildlife animal detection
  - vehicle detection and counting from space
  - ship detection, etc.
- And many others in medical imaging, robotics, astronomy, etc.

In general: bounding box + class for each object within the image[1]





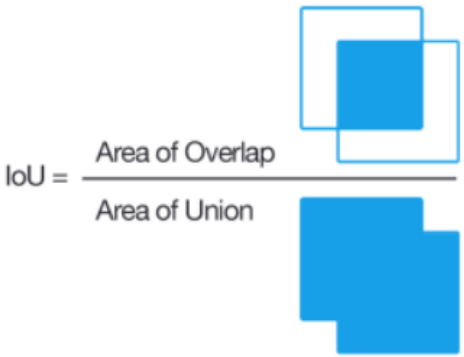
[1] Zhengxia Zou et al. Object Detection in 20 Years: A Survey

In computer vision[2]

Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2018	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
OID-2018	1,743,042	14,610,229	41,620	204,621	1,784,662	14,814,850	125,436	625,282

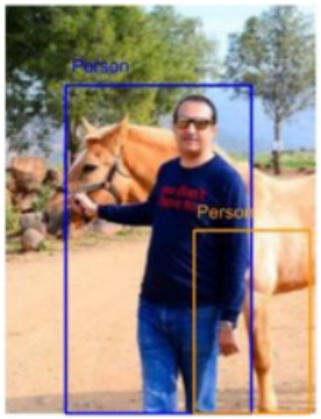
[2] Zhengxia Zou et al. Object Detection in 20 Years: A Survey

How to evaluate the predicted bounding boxes ?



(a) Intersection over Union (IoU)

(b) IoU > 0.5 (accept)



(c) IoU < 0.5 (reject)

What could be other criteria to evaluate the predicted bounding boxes?

Non Maximum Suppression: If predicted bounding boxes overlap, only consider the most confident.

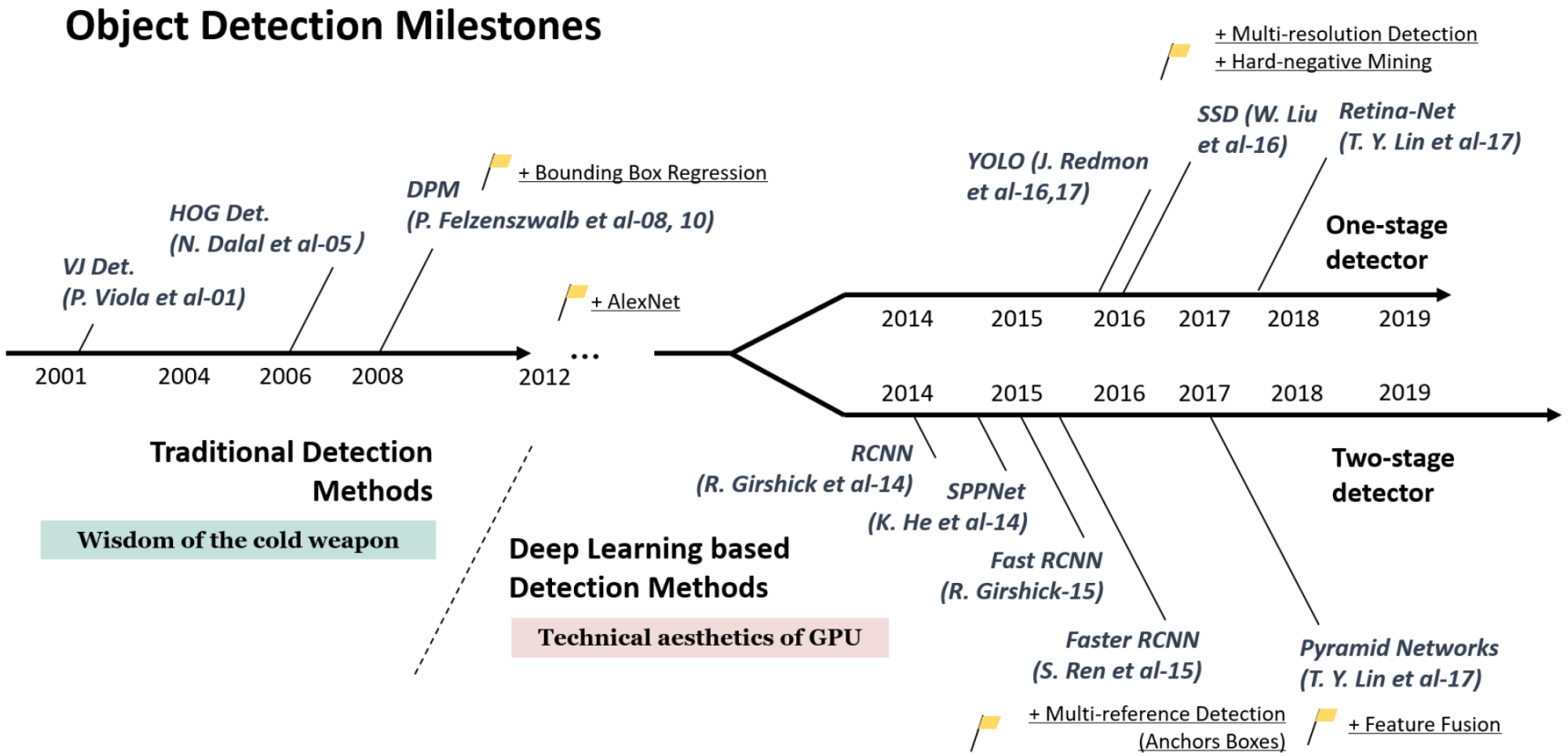
Bounding Box Regression: Regression used to find bounding box parameters (usually the box's center, width and height)

Example Loss Function

$$\sum_{i \ in \ I} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

I is the set of all matching bounding boxes (Highest IoU with ground truth)

A road map of object detection in computer vision[3]



[3] Zhengxia Zou et al. Object Detection in 20 Years: A Survey

Before deep learning: traditional detection methods

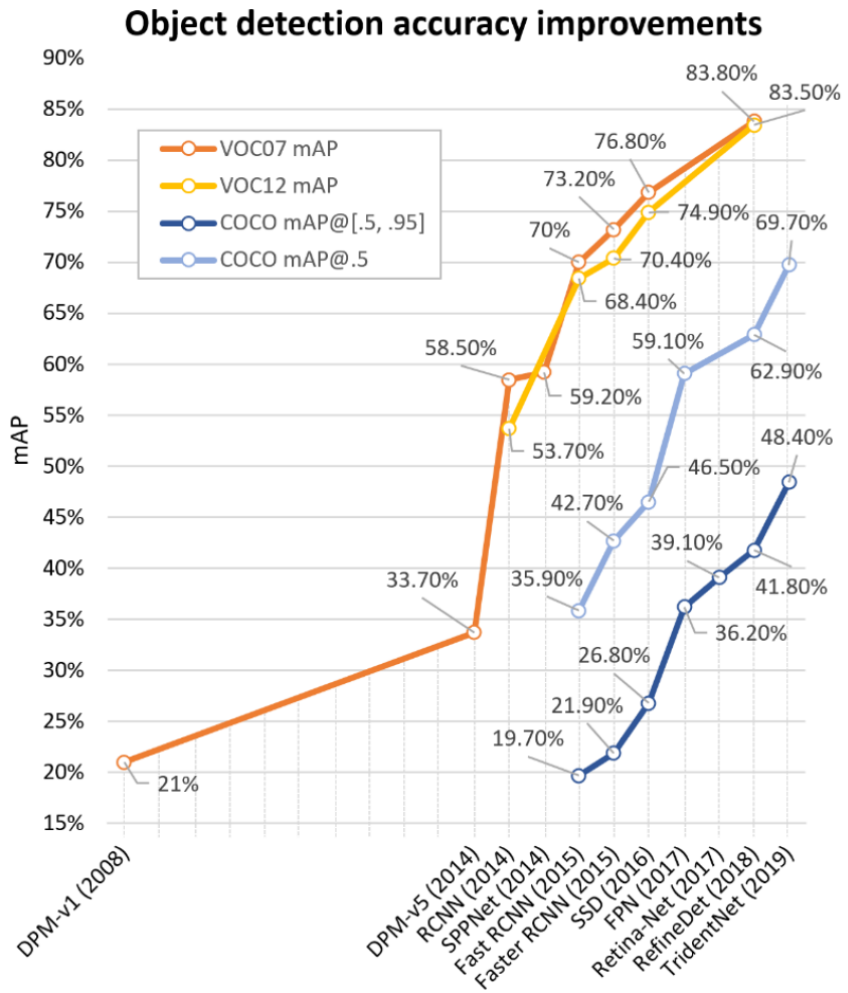
- Hand-crafted feature + classifier
- Features: HOG (Histogram of Gradients), LBP (Local Binary Pattern), textural features, etc.
- Classifier: SVM, Random Forest, etc.

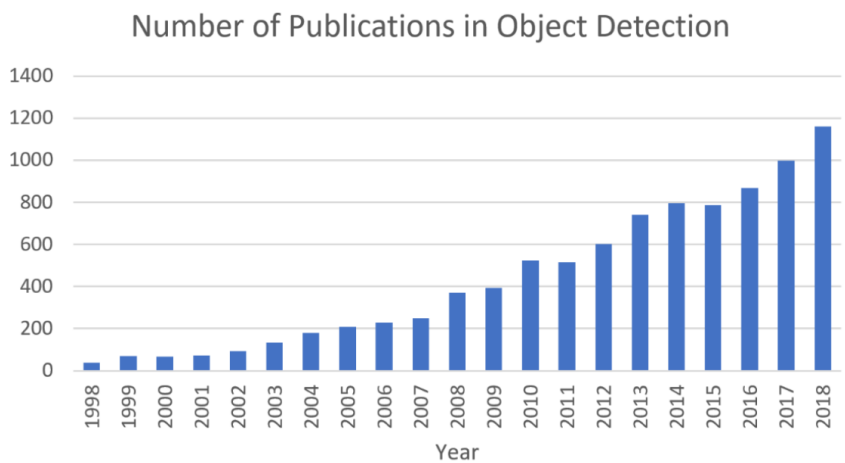
Modern deep-learning object detection[3]

- Two-stage detectors
- One-stage detectors

[3] http://zoey4ai.com/2018/05/12/deep-learning-object-detection/

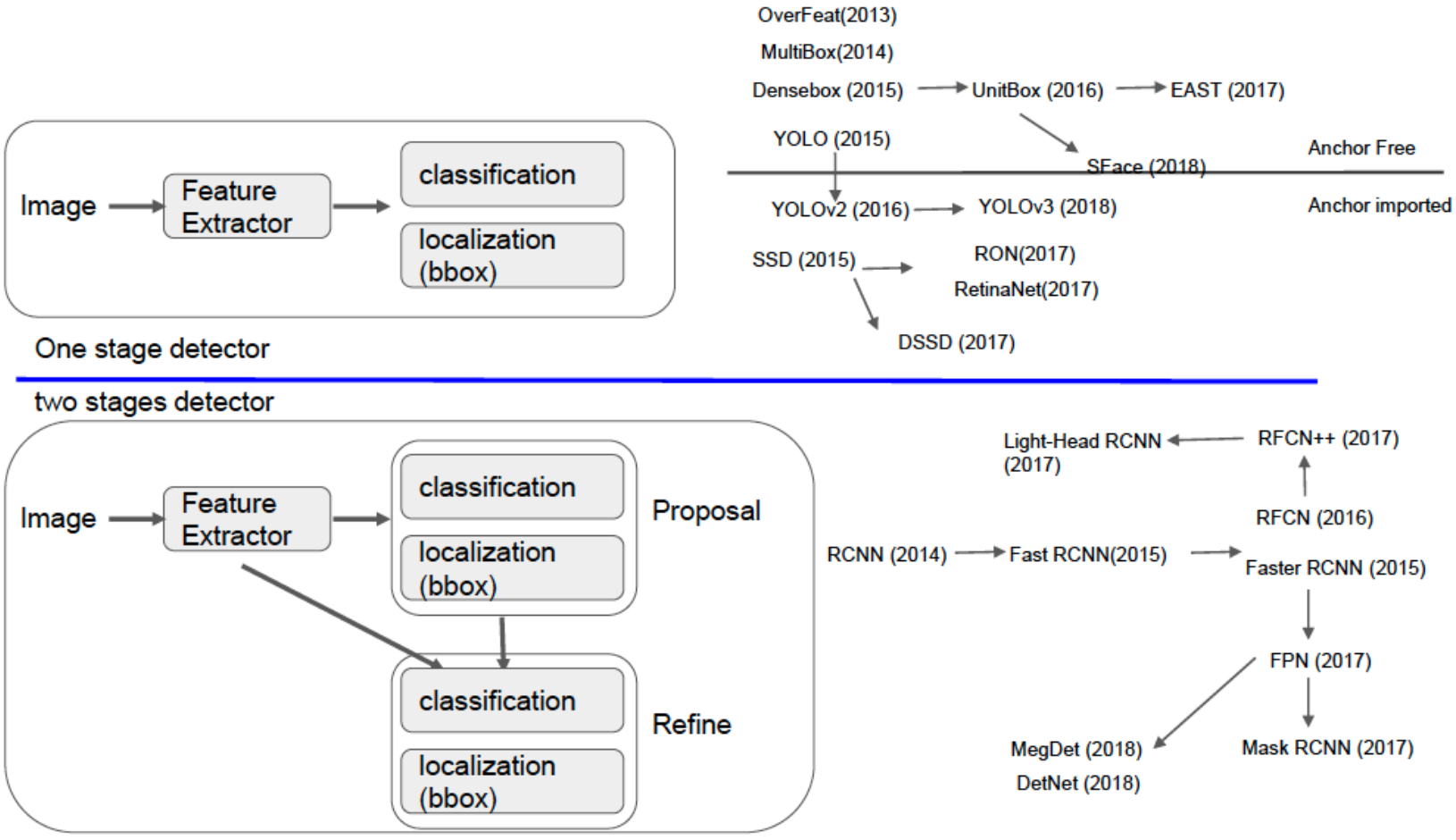
The evolution of deep-learning based object detection[3]





[3] Zhengxia Zou et al. Object Detection in 20 Years: A Survey

One-stage vs Two-stage detectors [3]

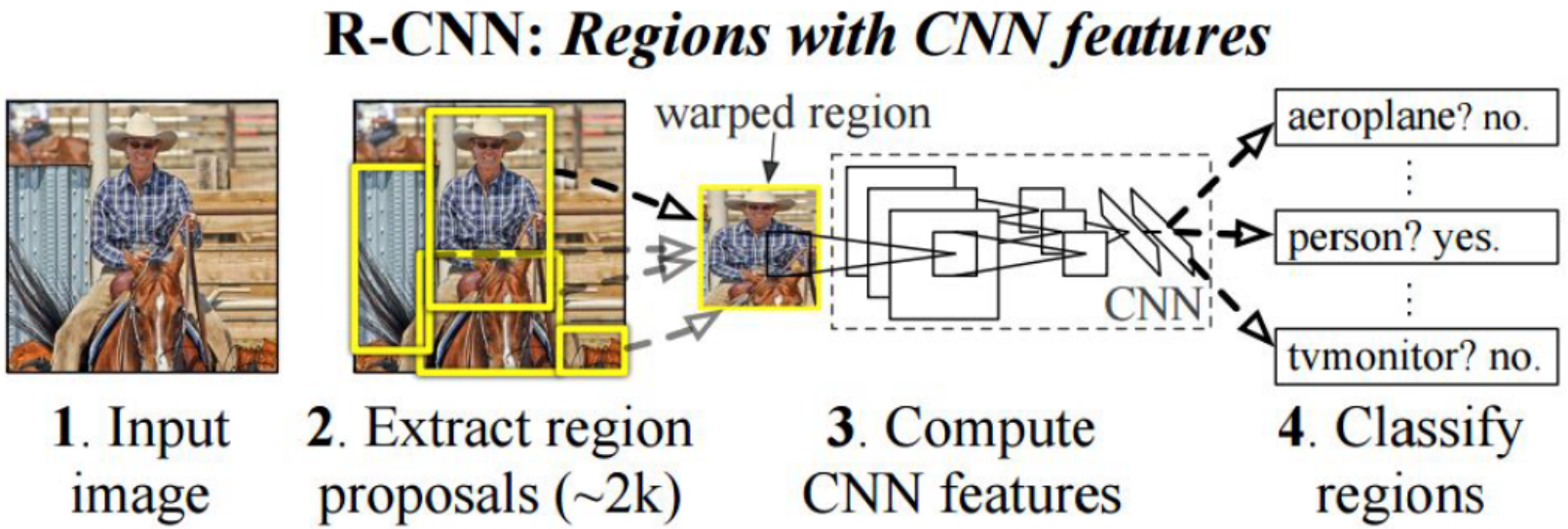


[3] Image source: Gang Yu, An Introduction to Modern Object Detection

Table of contents

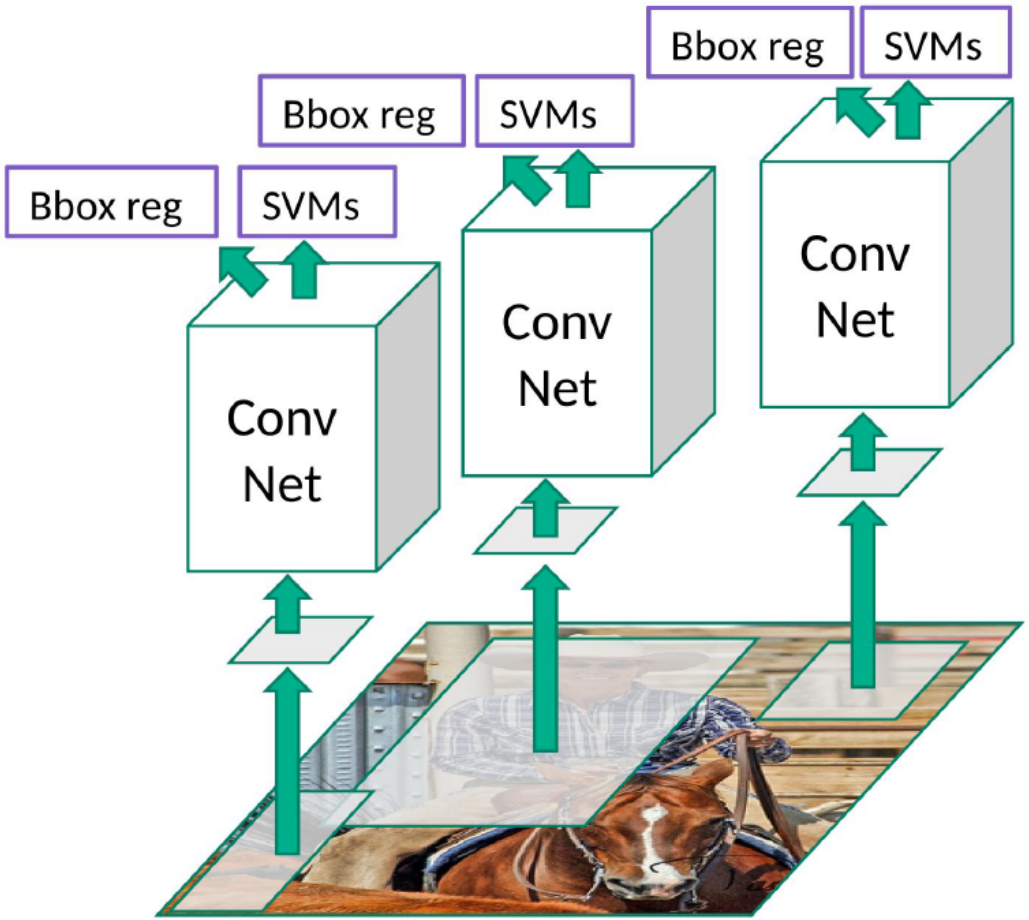
R-CNN: Region proposal + CNN [4]

- Use **Selective Search** to come up with regional proposal
- Then classification + bounding box regression for each proposal
- First object detection method using CNN



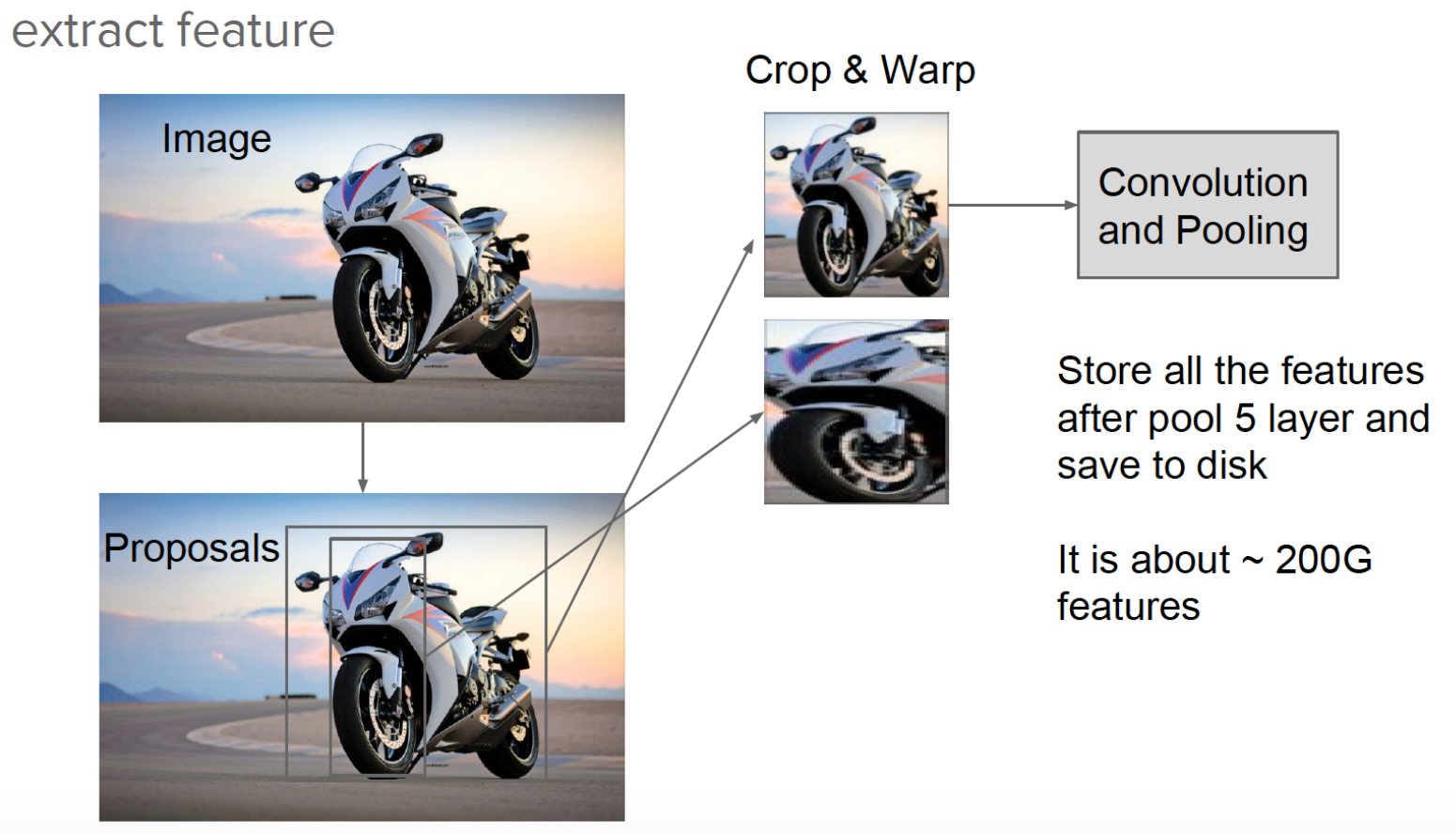
[4] Ross Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014.

R-CNN: Region proposal + CNN [4]



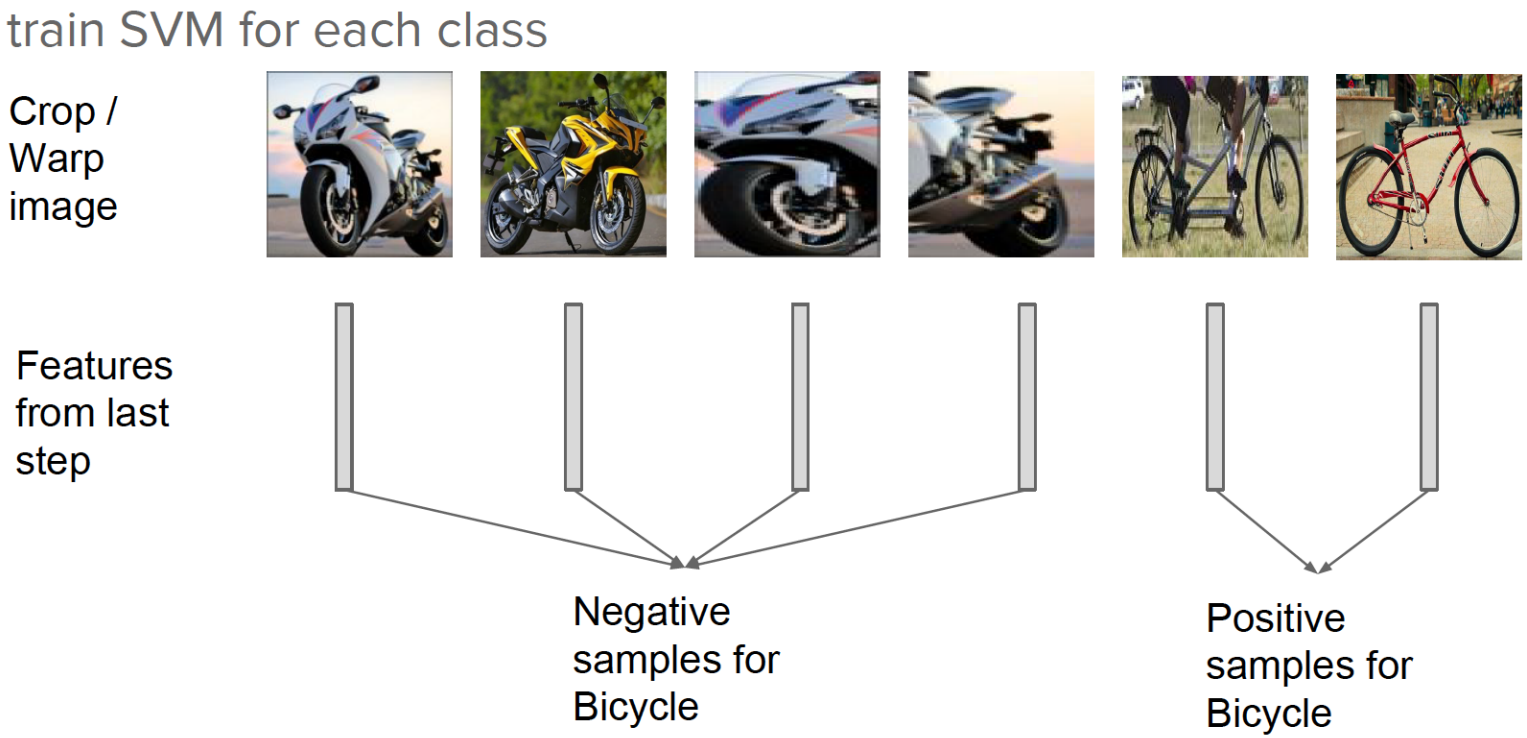
[4] Ross Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014.

Example: [5]



[5] Sihao Liang, Jiajun Lu and Kevin Perkins, Lecture on Object detection.

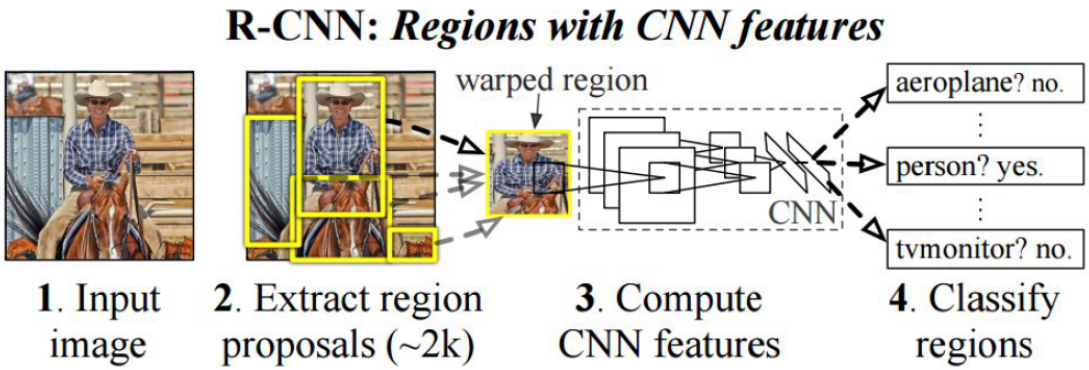
Example: [5]



[5] Sihao Liang, Jiajun Lu and Kevin Perkins, Lecture on Object detection.

R-CNN: Summary [6]

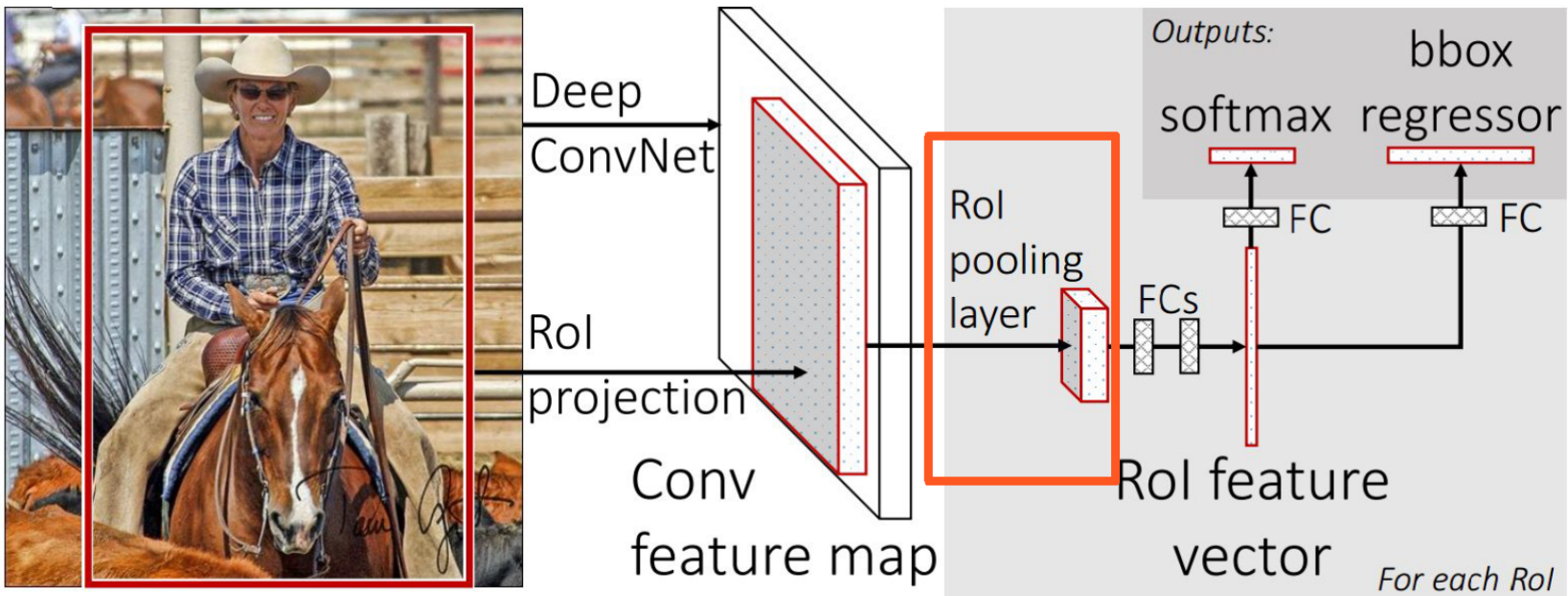
- Simple pipeline!
- Use the CNN as a feature extractor
- Warp cropped regions to make them “square"
- Run an SVM (one per category) on the deep features (stocked on disk ⇒ very heavy!!!)
- Accuracy ☺ but very slow ☹
- Not end-to-end ☹



[6] Ross Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014.

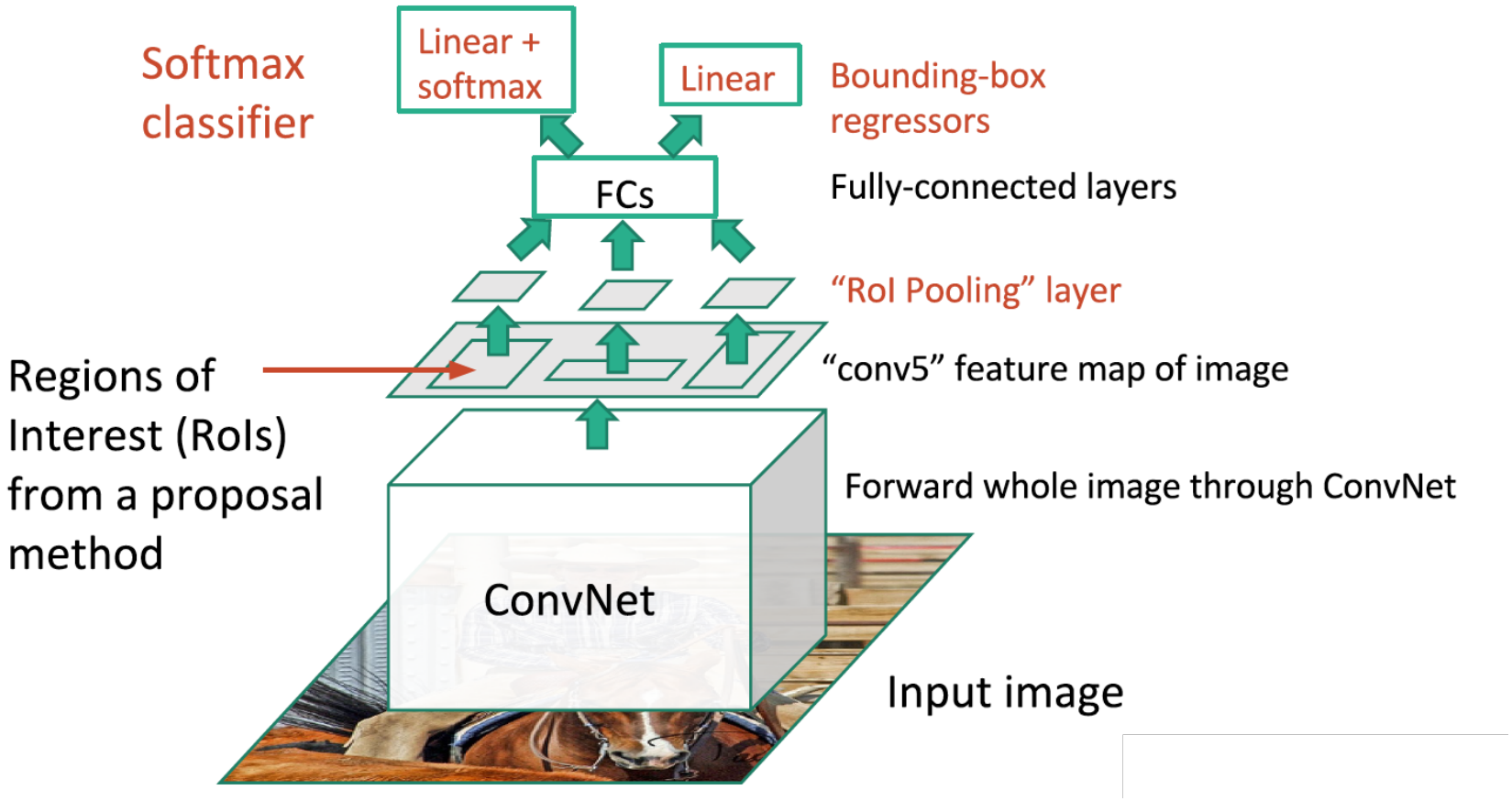
Fast R-CNN: architecture [7]

- **Share convolution layers** for proposals from the same image
- Introduce the **ROI Pooling**
- Faster and more accurate than R-CNN



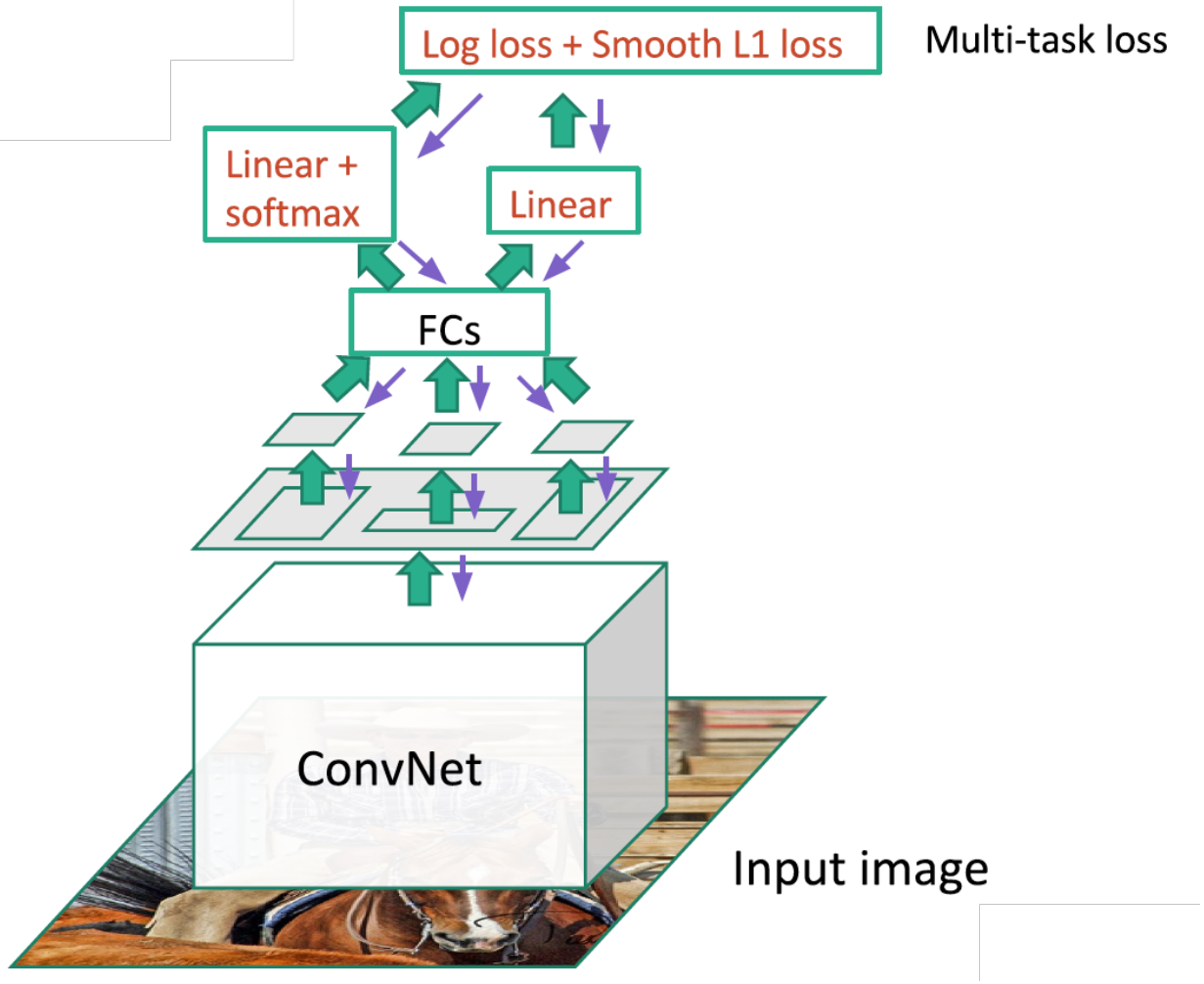
[7] Ross Girshick et al. Fast RCNN, ICCV 2015.

Fast R-CNN: architecture [7]



[7] Ross Girshick et al. Fast RCNN, ICCV 2015.

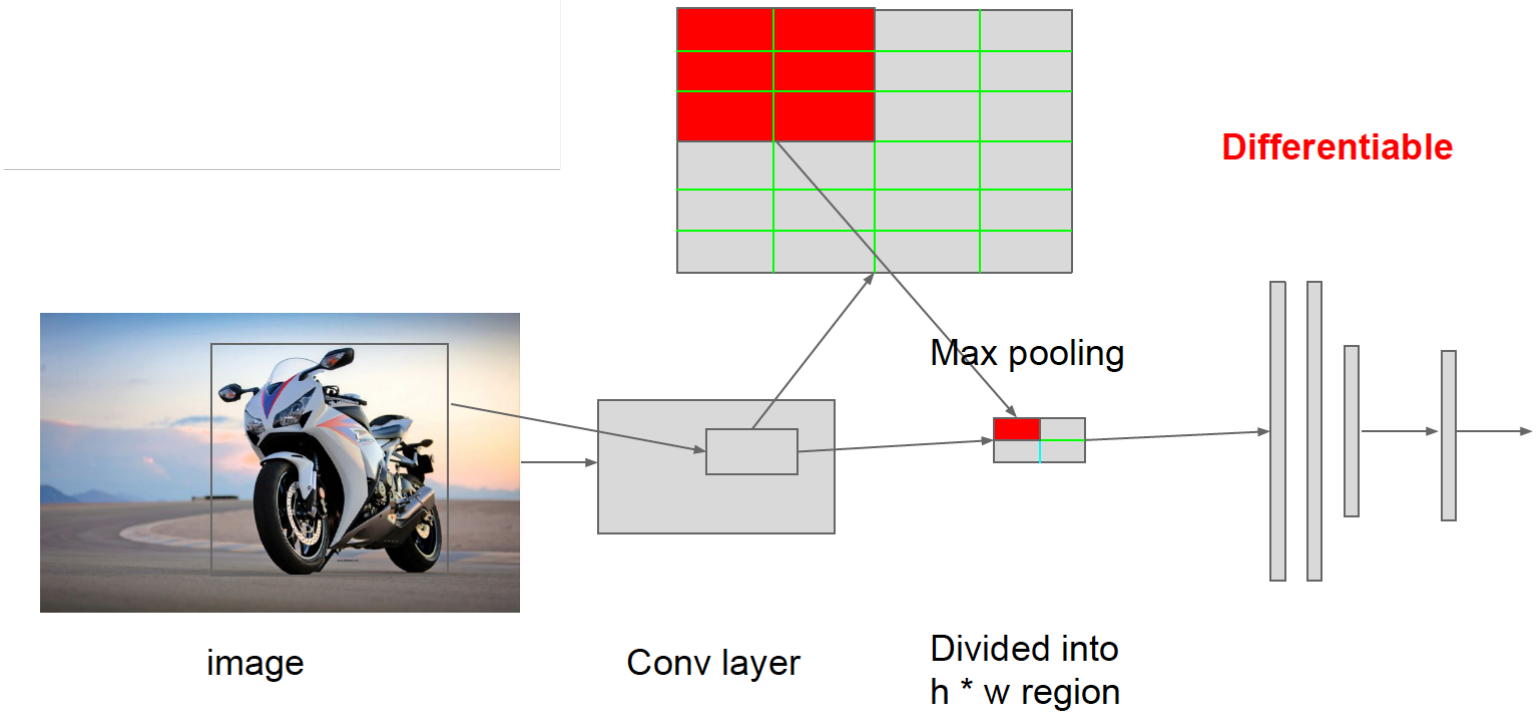
Fast R-CNN: architecture [7]



[7] Ross Girshick et al. Fast RCNN, ICCV 2015.

Explanation: [8]

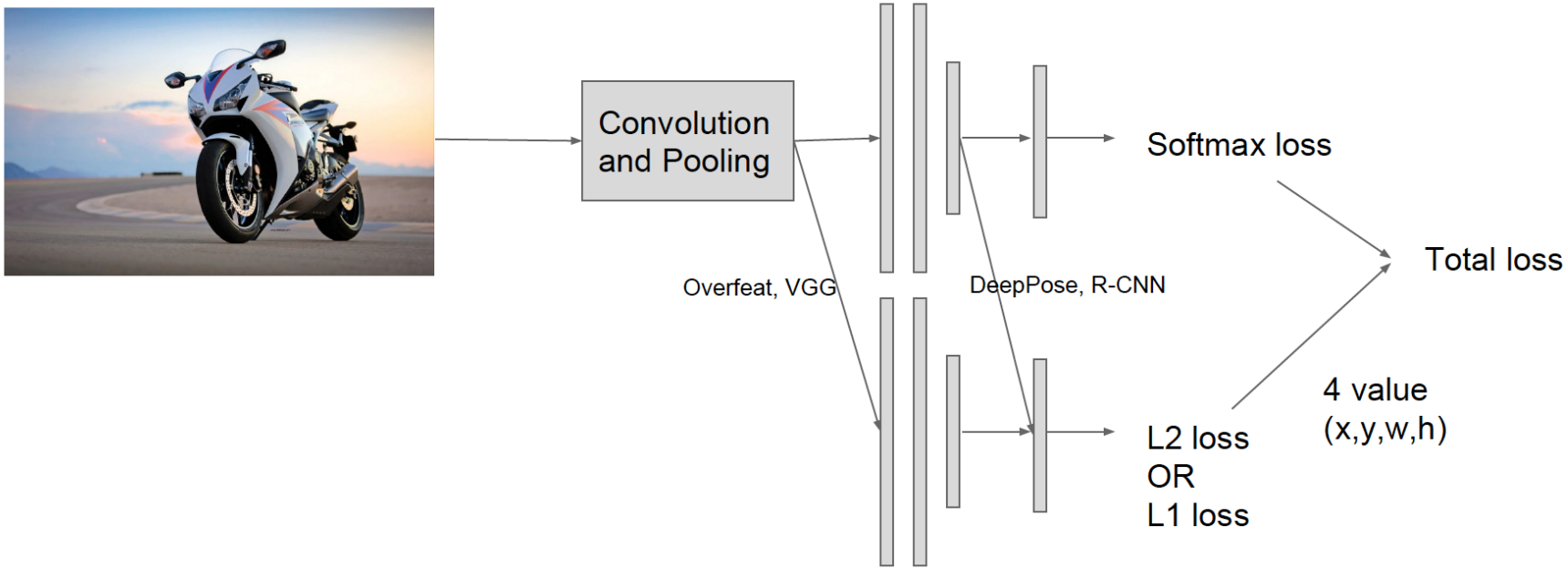
ROI Pooling



[8] Sihao Liang, Jiajun Lu and Kevin Perkins, Lecture on Object detection.

Explanation: [8]

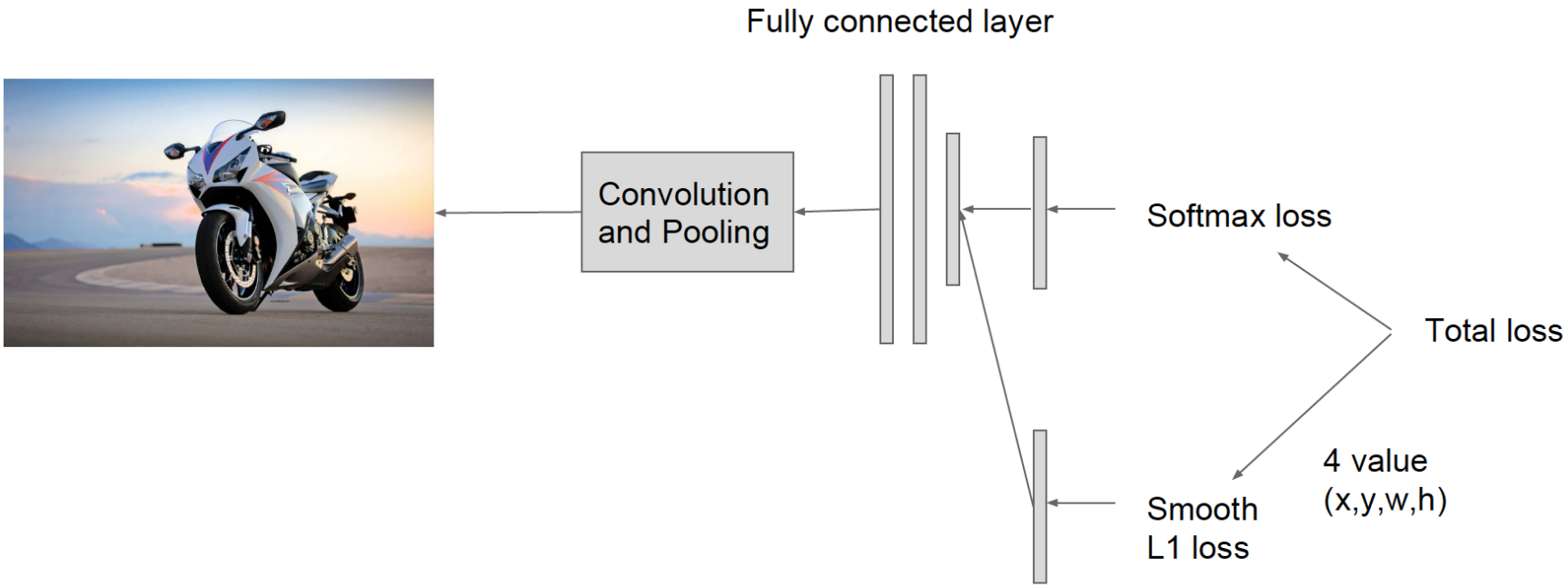
Bbox regressor



[8] Sihao Liang, Jiajun Lu and Kevin Perkins, Lecture on Object detection.

Explanation: [8]

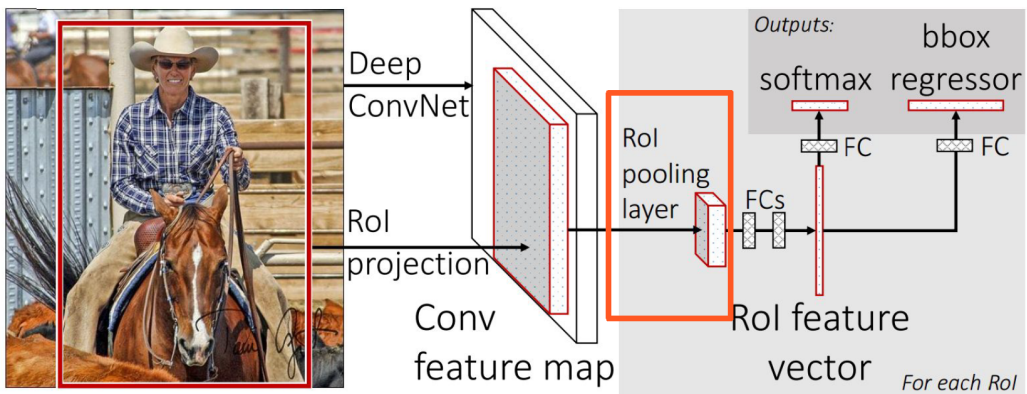
Bbox regressor



[8] Sihao Liang, Jiajun Lu and Kevin Perkins, Lecture on Object detection.

Fast R-CNN: Summary [9]

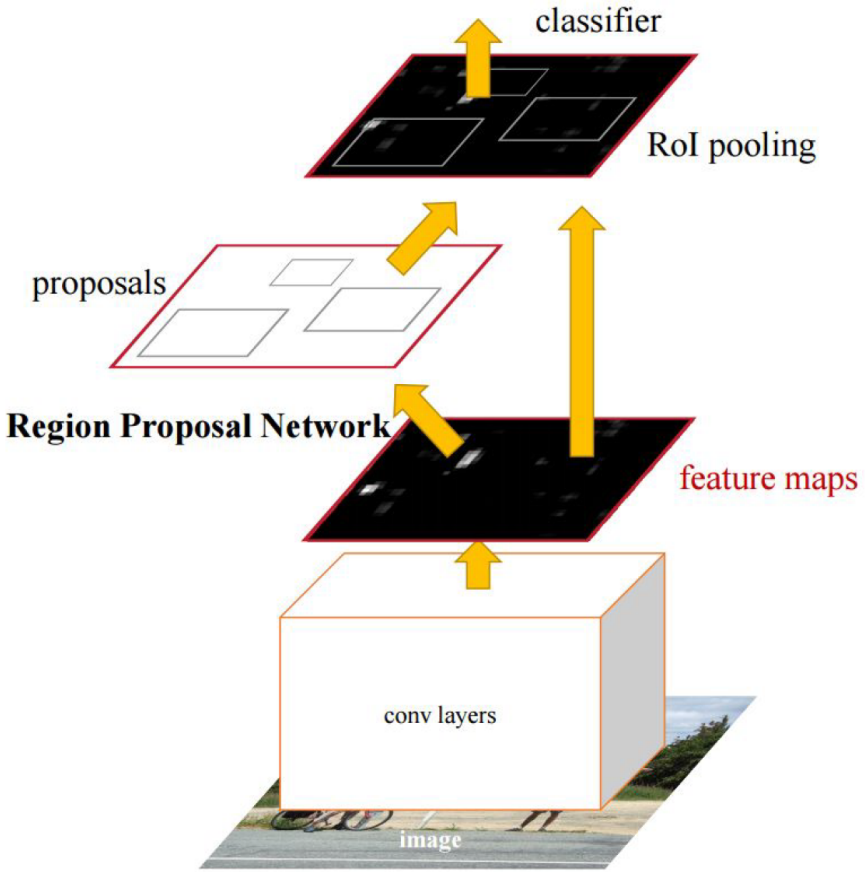
- Do not compute forward propagate from scratch for each box
- Compute feature maps once and reuse the convolutions for different boxes
- New: ROI pooling
- End-to-end training ☺
- More accuracy and must faster (25x) than R-CNN ☺
- External box proposals needed ☹



[9] Ross Girshick et al. Fast RCNN, ICCV 2015.

Faster R-CNN: architecture [10]

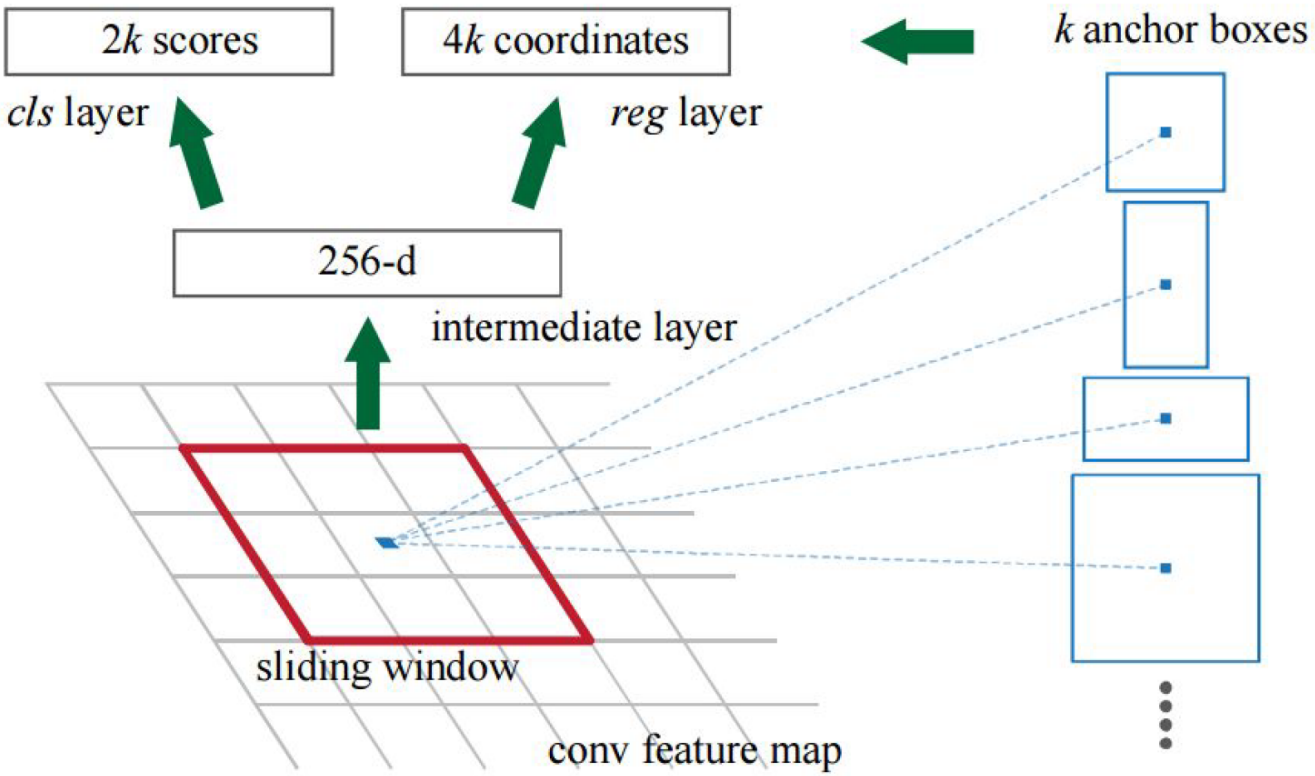
- Don’t need to have external regional proposals (which is the bottleneck)
- RPN - Regional Proposal Network



[10] Shaoqing Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NeurIPS’15.

Faster R-CNN: architecture [10]

- Don't need to have external regional proposals (which is the bottleneck)
- RPN - Regional Proposal Network



[10] Shaoqing Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NeurIPS'15.

Faster R-CNN: architecture [10]

- Don’t need to have external regional proposals (which is the bottleneck)
- RPN - Regional Proposal Network

	Faster R-CNN	Fast R-CNN	R-CNN
Test time/image With proposal	0.2S	2s	50s
-test speedup	250x	25x	1x
mAP	66.9	66.9	66

(Training on PASCAL VOC 2007 dataset)

[10] Shaoqing Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NeurIPS’15.

Other SOTA two-stage detectors

- R-FCN: Object Detection via Region-based fully convolutional networks (NeurIPS 2016)
- FPN: Feature pyramid networks for object detection (CVPR 2017)
  $\Rightarrow$ great performance for detecting objects with a wide variety of scales

Remarks

- Faster R-CNN $\Rightarrow$ most exploited in the literature
- FPN $\Rightarrow$ a basic building block of many latest detectors
- **Faster R-CNN + FPN** $\Rightarrow$ an excellent choice to work on two-stage object detector for your applications !

Table of contents

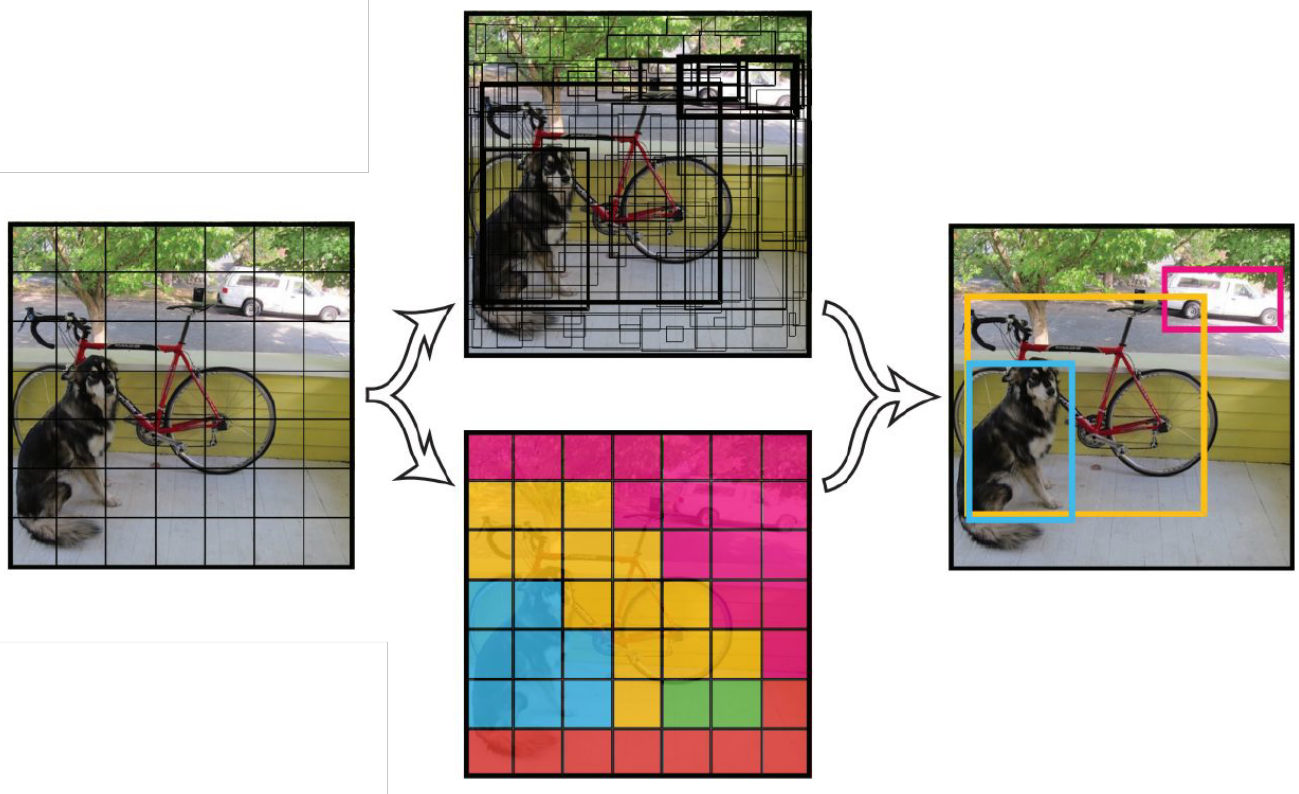
One-stage detectors

Problems with two-stage detectors

- Complex Pipeline
- Hard to optimize each component
- Main: **Slow** (Cannot run in real time)

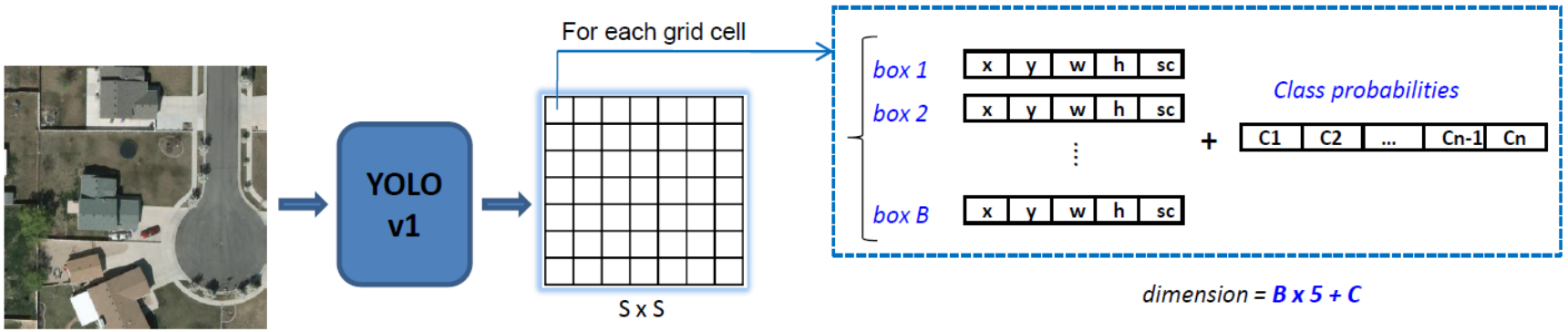
YOLO: You Only Look Once [11]

- Consider detection a regression problem
- Use a single CNN
- Runs once on entire image ⇒ Very Fast!



[11] Joseph Redmon et al., You only look once: Unified, real-time object detection, CVPR 2016.

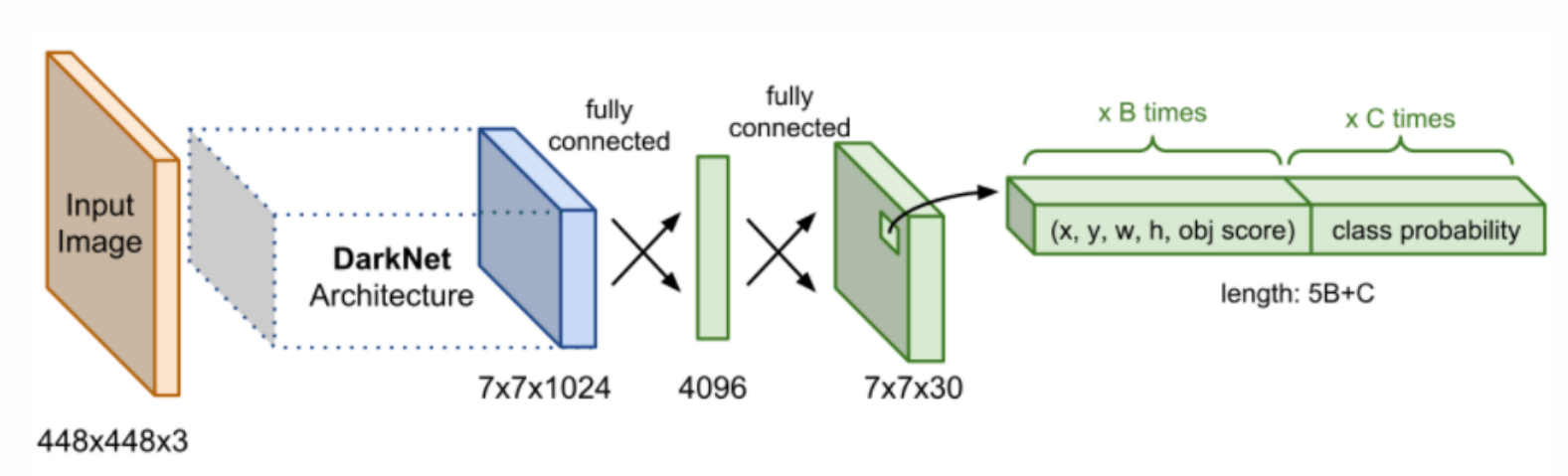
YOLO: You Only Look Once [11]

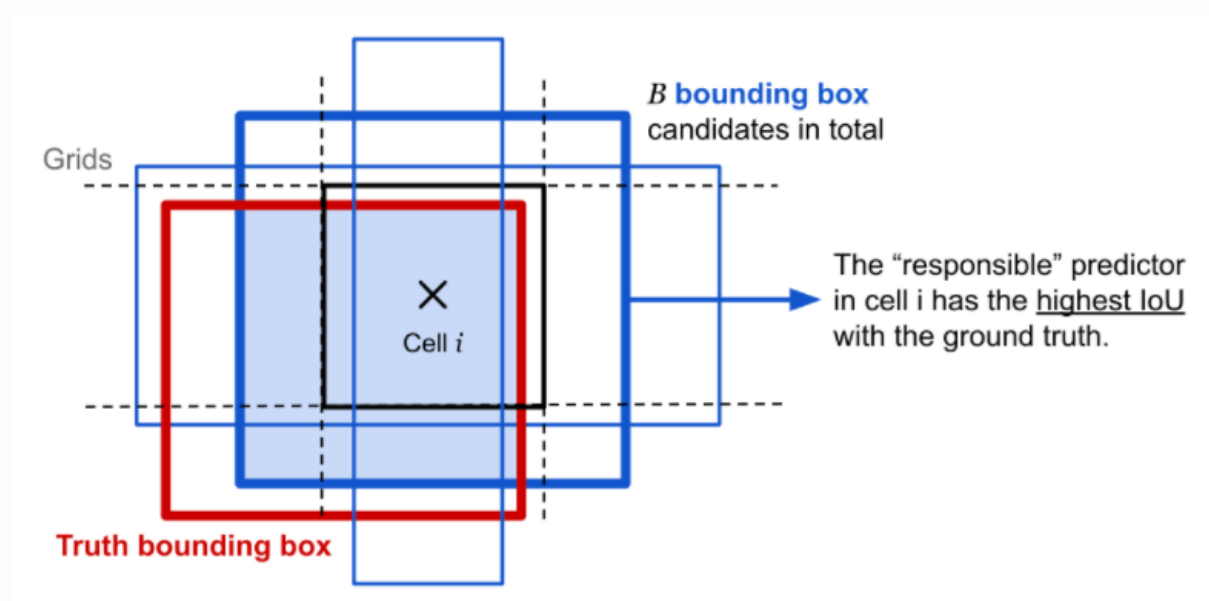


In YOLOv1, the output is a tensor of dimension $(S, S, B \times 5 + C)$ with (S, S) the size of the grid, B the number of predicted boxes for each cell and C the number of classes. By default, $S = 7, B = 2$ and $C = 20$ for the PASCAL VOC dataset. For an input image of size 448×448 pixels, the output is a tensor of size $7 \times 7 \times 30$.

[11] Joseph Redmon et al., You only look once: Unified, real-time object detection, CVPR 2016.

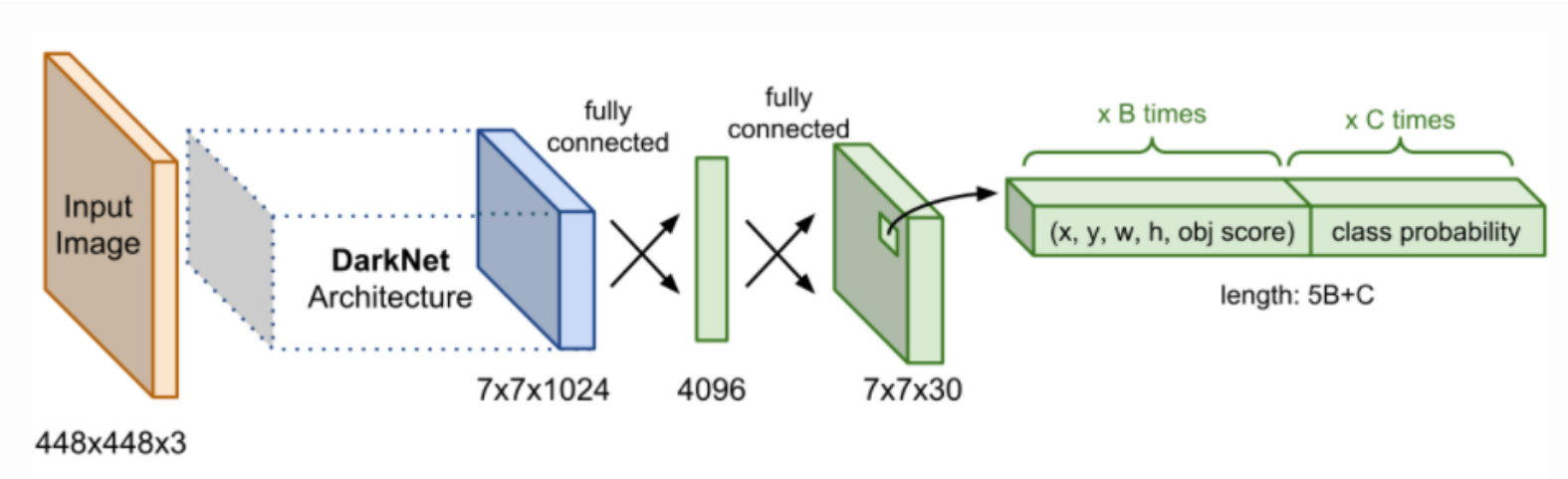
YOLO: You Only Look Once [11] Let's take a closer look!





[11] Joseph Redmon et al., You only look once: Unified, real-time object detection, CVPR 2016.

YOLO: You Only Look Once [11] And the loss function!



$$\mathcal{L}_{\text{loc}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

$$\mathcal{L}_{\text{cls}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (1_{ij}^{\text{obj}} + \lambda_{\text{noobj}}(1 - 1_{ij}^{\text{obj}}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \in \mathcal{C}} 1_i^{\text{obj}} (p_i(c) - \hat{p}_i(c))^2$$

$$\mathcal{L} = \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{cls}}$$

[11] Joseph Redmon et al., You only look once: Unified, real-time object detection, CVPR 2016.

YOLO: Very fast but many limitations

- Struggles with small objects
- Struggles with unusual aspect ratios
- Poor localization

	Yolo	Faster R-CNN (VGG-16)
mAP	63.4	73.2
FPS	45	7

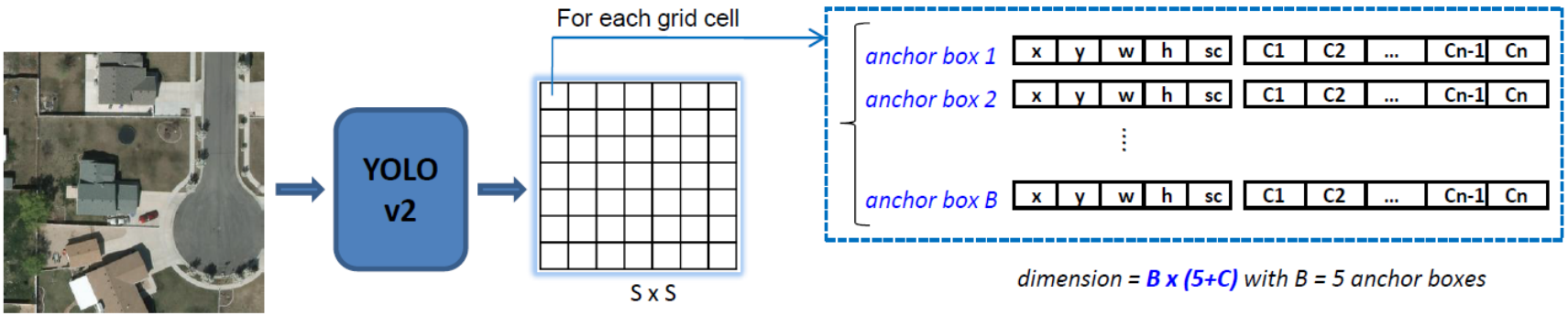
Trained on Pascal VOC 2007 + 2012 dataset

YOLO9000: better, faster, stronger [12]

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

[12] Joseph Redmon et al., YOLO9000: better, faster, stronger, CVPR 2017.

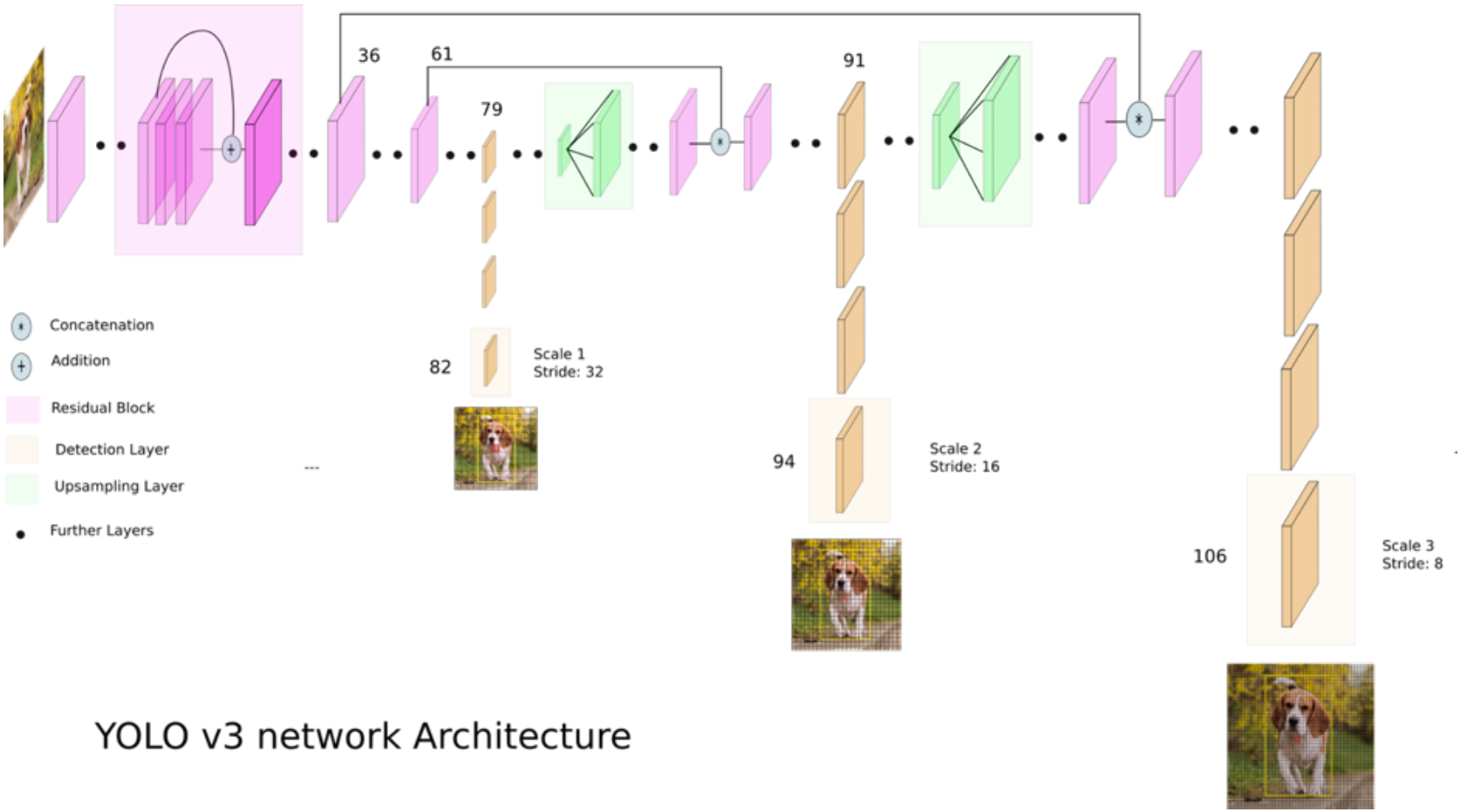
YOLO9000: better, faster, stronger [12]



In YOLOv2, the output is a tensor of dimension $(S, S, B \times (5 + C))$. The difference is that the class probabilities are calculated for each anchor box. By default, $S = 13, B = 5$ anchor boxes and $C = 20$ for the PASCAL VOC dataset. For an input image of size 416×416 pixels, the output is a tensor of size $13 \times 13 \times 125$.

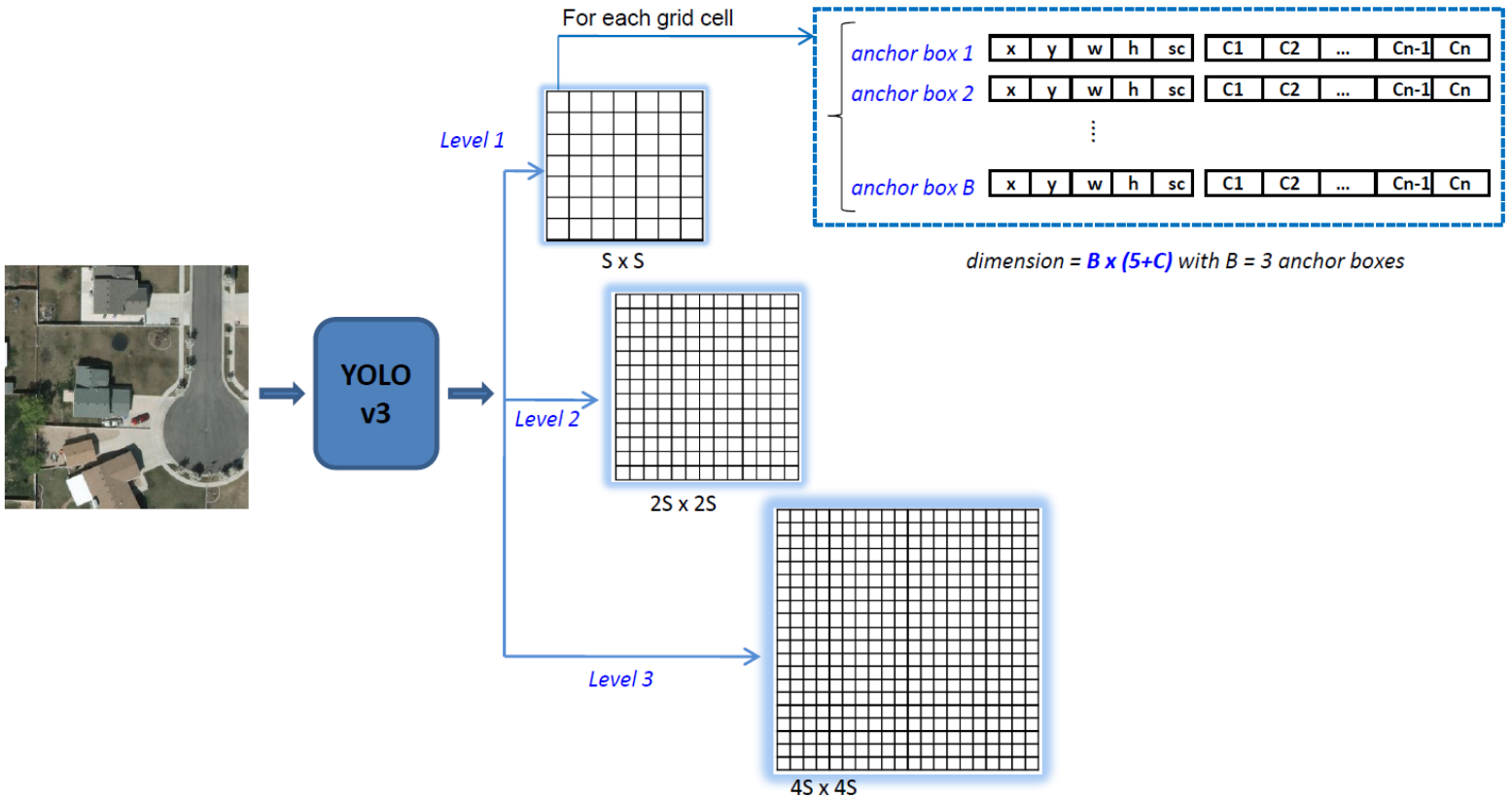
[12] Joseph Redmon et al., YOLO9000: better, faster, stronger, CVPR 2017.

YOLOv3: multi-level detection [13]



[13] Joseph Redmon et al., Yolov3: An incremental improvement, Arxiv 2018.

YOLOv3: multi-level detection [13]



In YOLOv3, the output consists of 3 tensors of dimension $(S, S, B \times (5 + C))$, $(2S, 2S, B \times (5 + C))$ and $(4S, 4S, B \times (5 + C))$ which correspond to the 3 detection levels (scales). By default, $S = 13, B = 3$ anchor boxes and $C = 80$ for the COCO dataset. For an input image of size 416×416 pixels, the outputs are three tensors of size $13 \times 13 \times 255$, $26 \times 26 \times 255$ and $52 \times 52 \times 255$.

[13] Joseph Redmon et al., Yolov3: An incremental improvement, Arxiv 2018.

One-stage detectors Others

Other SOTA one-stage detectors

- SSD: Single shot multibox detector (ECCV 2016) and its variants!
- RetinaNet (ICCV 2017): focal loss
- EfficientDet (CVPR 2020)
- YOLOv4: Optimal Speed and Accuracy of Object Detection, Arxiv 2020
- Scaled-YOLOv4: Scaling Cross Stage Partial Network, CVPR 2021
- YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, Arxiv 2022
- YOLOv8 Ultralytics

Your remarks ???

References I

Books & articles:

1. Zhengxia Zou et al. **Object Detection in 20 Years: A Survey.** Arxiv, 2020.
2. Liu, Li, et al. **Deep learning for generic object detection: A survey.** International journal of computer vision 128.2 (2020): 261-318.

Courses:

1. Lecture 11: Detection and Segmentation, Fei-Fei Li, Justin Johnson & Serena Yeung
   `http://cs231n.stanford.edu/slides/2017/`
2. Lecture 07: Object Detection, Sihao Liang, Jiajun Lu & Kevin Perkins
   `http://slazebni.cs.illinois.edu/spring17/`
3. Lecture 6: Convnets for object detection and segmentation, Deep Learning at UvA
   `https://uvadlc.github.io/lectures/feb2016/`