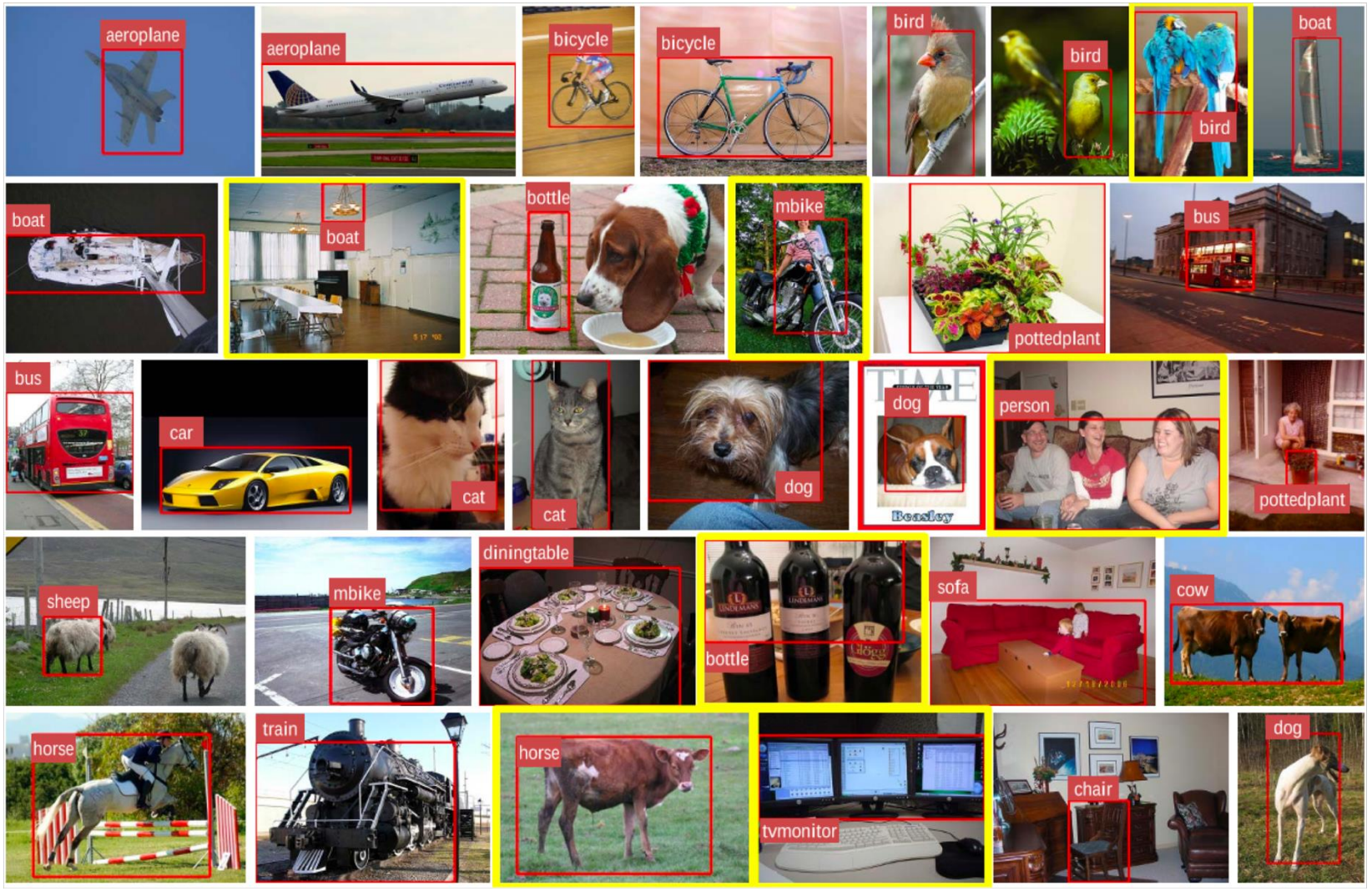
Advances in Deep Learning

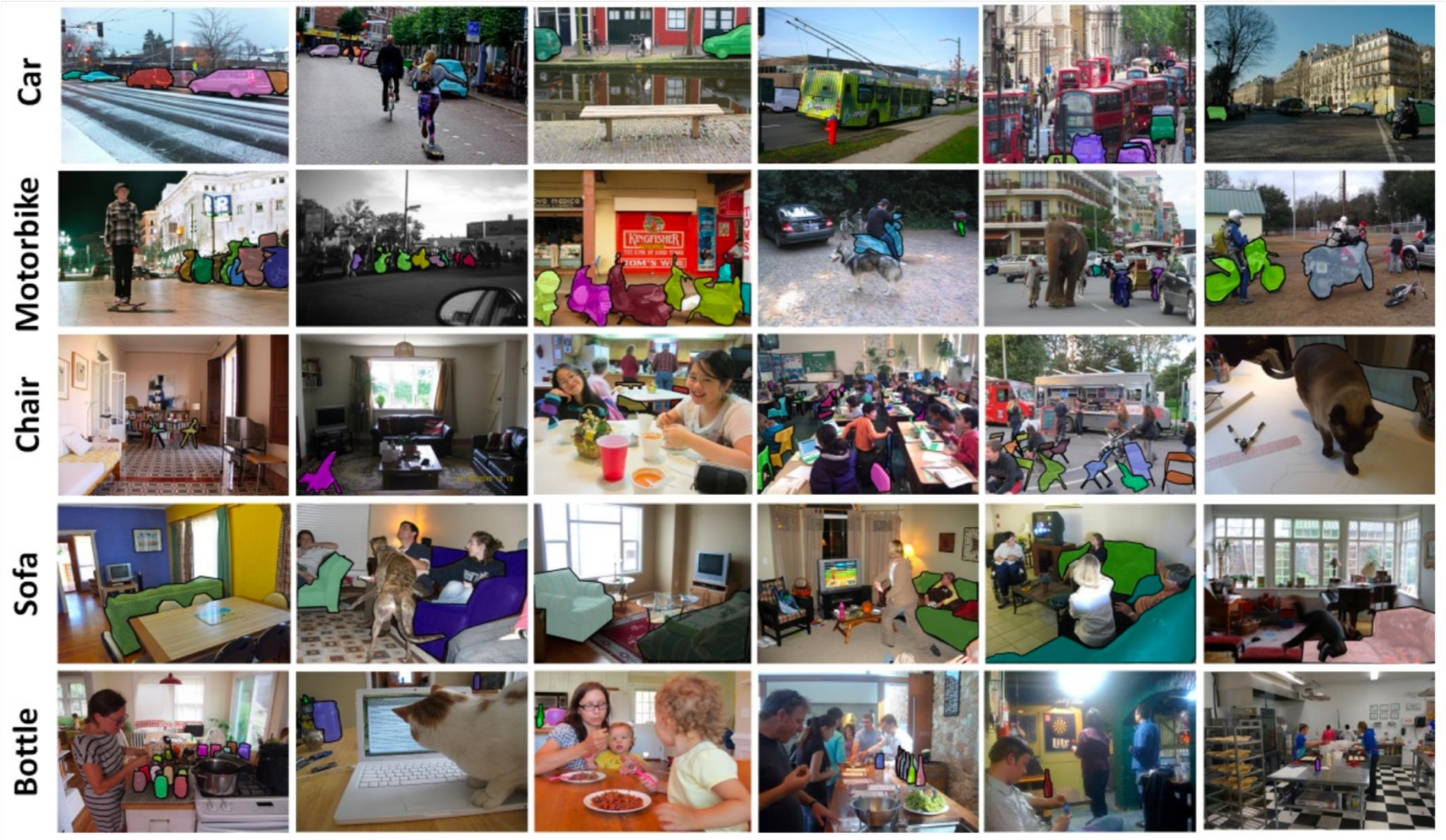
Object Detection

DataSet - Object Detection



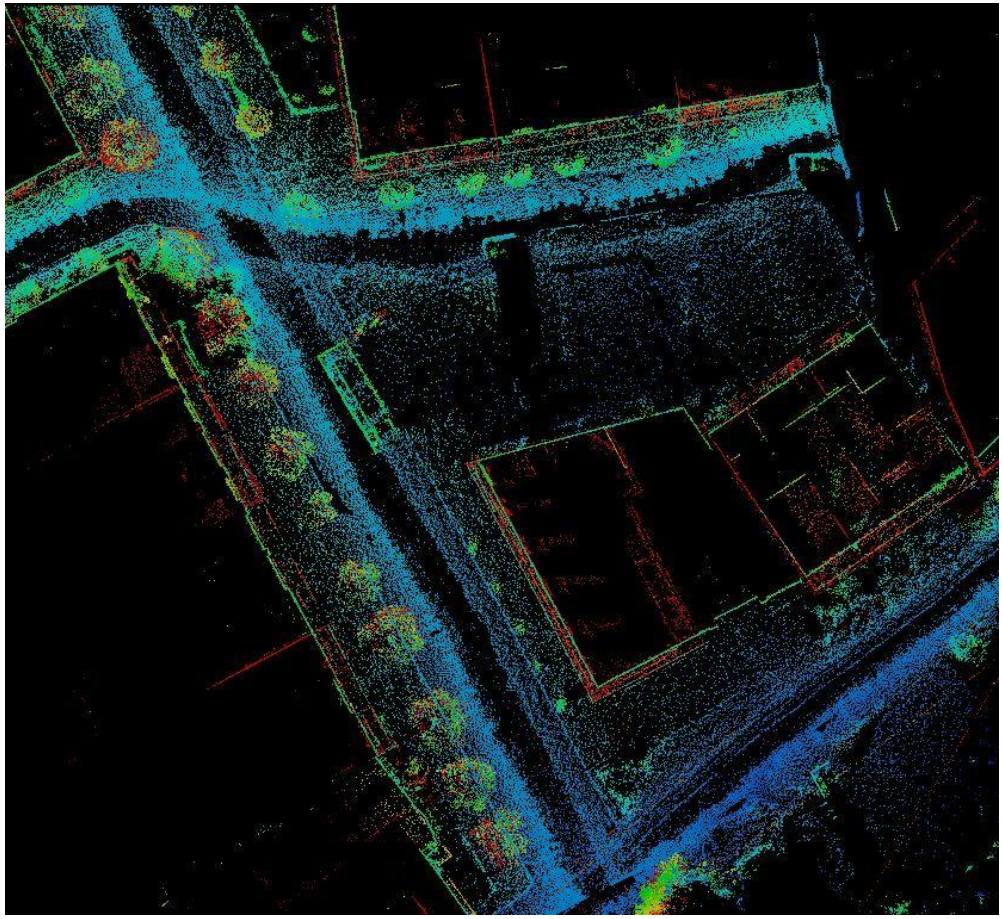
PASCAL VOC 20 classes.

DataSet - Object Detection



Samples of annotated images in the MS COCO dataset.

DataSet – Point Cloud Detection



Samples of KITTI, a point cloud dataset for self-driving cars

http://www.cvlibs.net/datasets/kitti/raw_data.php

DataSet - Pose Estimation

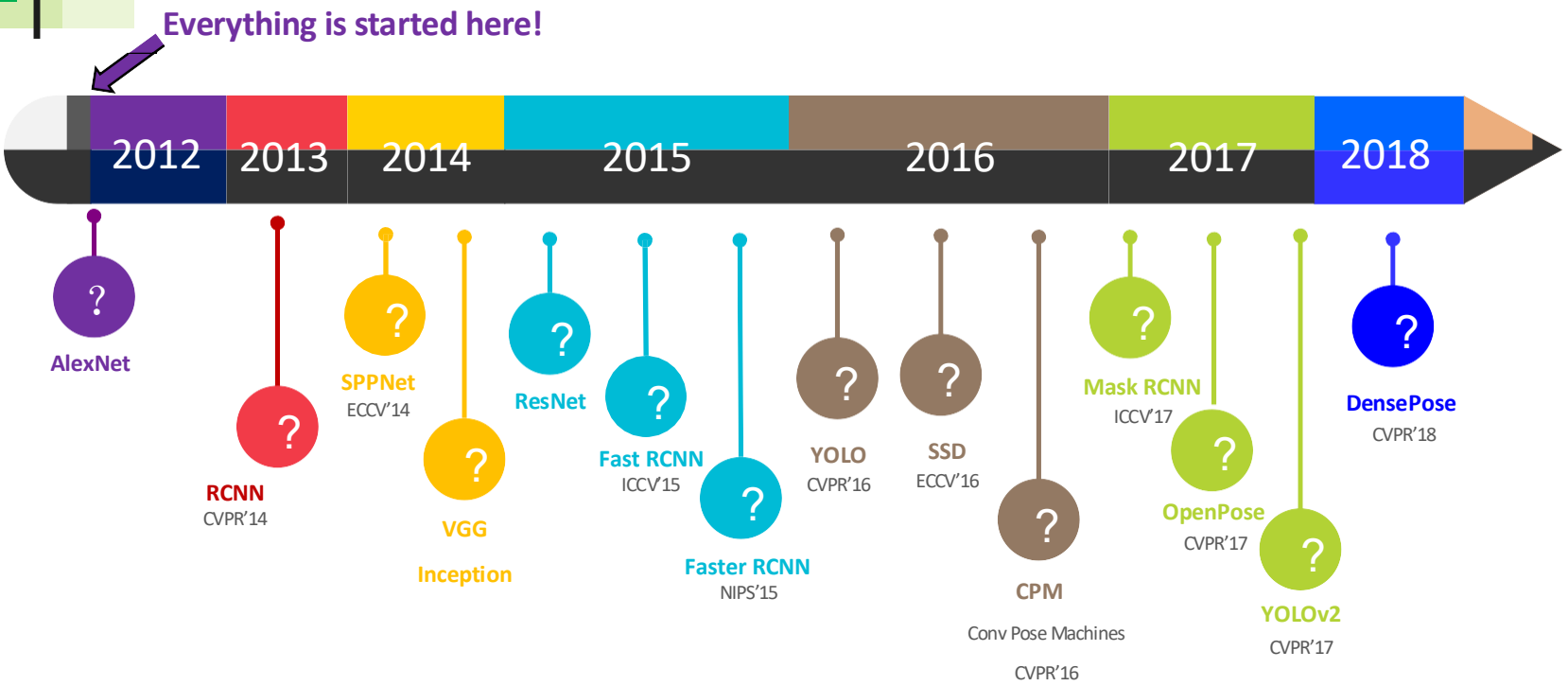
Open pose using CMU Panoptic Studio dataset

https://github.com/CMU-Perceptual-Computing-Lab/openpose

DataSet - Pose Estimation

https://www.youtube.com/watch?time_continue=3&v=Dhkd_bAwwMc

Background



Object Detection: RCNN, SPPNet, Fast RCNN, YOLO, Faster RCNN, SSD, YOLOv2, CenterNet, Reppoints, YOLOV3…

- Face detection; Counting; Visual Search Engine; Geographic Object-Based Image Analysis;

Pose Estimation: To recover the joint positions of articulated limbs of human body given an image or a video. CPM, Mask RCNN, OpenPose, DensePose.

- Action Recognition; Human–computer interaction (HCI), game and animation; Clothing parsing

Terminology

mAP (mean average precision)

- AP: e.g. search flower image 1 is flower, 0 is not

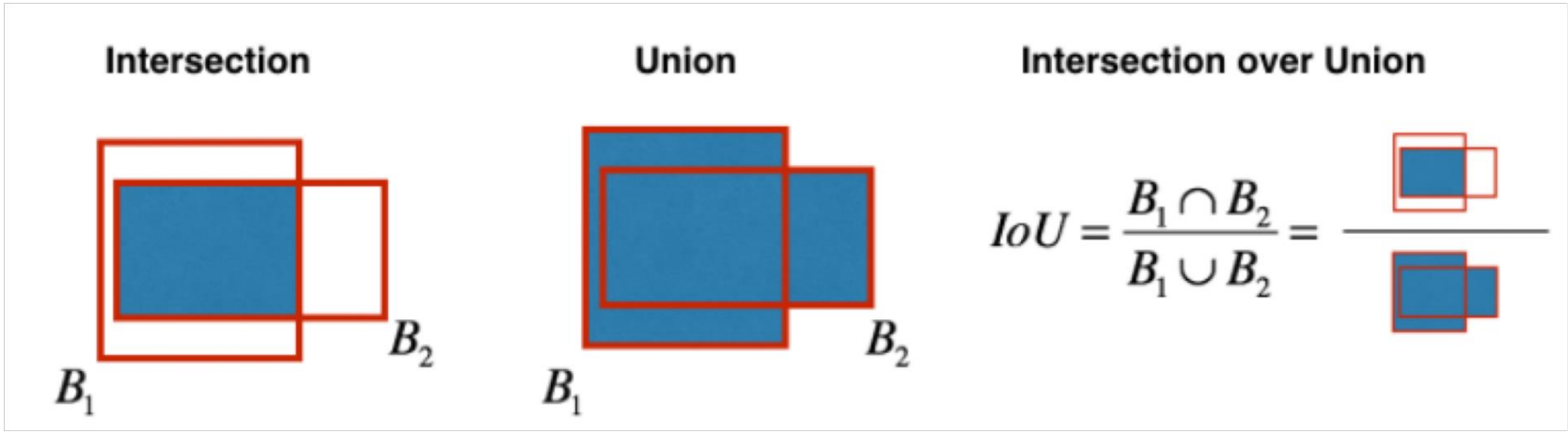
  1,0,0,1,1,1 (query) – 1/1,0,0,2/4,3/5,4/6 (accuracy)

  AP = (1+2/4+3/5+4/6)/6 = 0.461

  - Given AP = 0.5, query can be 0,1,0,1…..
  - (order matters if there is incorrect prediction)
- mAP: average of the AP of each query sequence
  - calculate the average precision for each class in your data based on your model predictions.
  - Average precision is related to the area under the **precision-recall curve** for a class. Then taking the mean of these average individual-class-precision gives you the Mean Average Precision.

Terminology

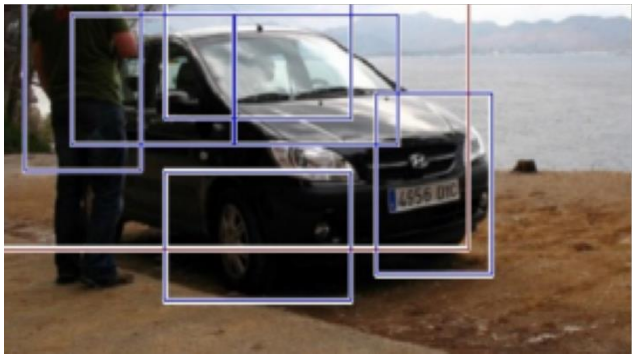
Intersection over Union, or IoU.



Terminology

NMS (Non-Maximum Suppression)

- Selecting the bounding box with the highest confidence.
- Bounding boxes that are close to the selected one are suppressed and merged greedily.
- Next top-scoring one is selected, and the procedure is repeated until no more bounding boxes remain.



First Introduction of CNN to object detection: R-CNN

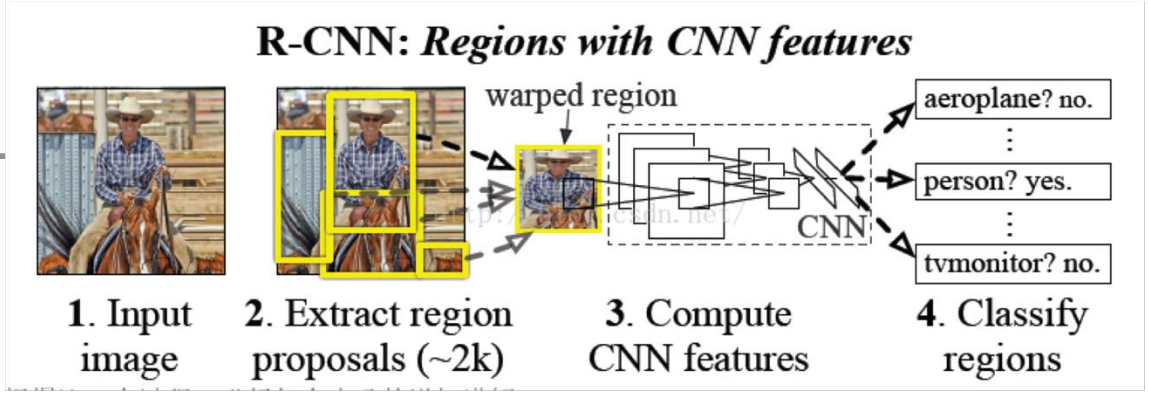
- Main Features:
  - Selective search
  - A natural combination of heuristic region proposal method and ConvNet feature extractor
  - Dataset: Classic object detection extract manually pre-defined features in each region.

    Classification: ImageNet ILSVC 2012, 1000 classes;

    Detection: PASCAL VOC 2007, 20 object classes with locations, fine-tuning

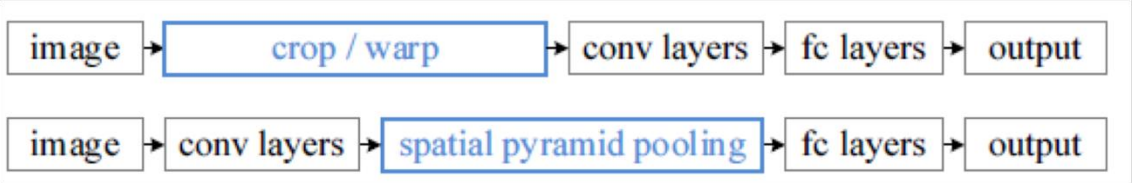
Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

R-CNN

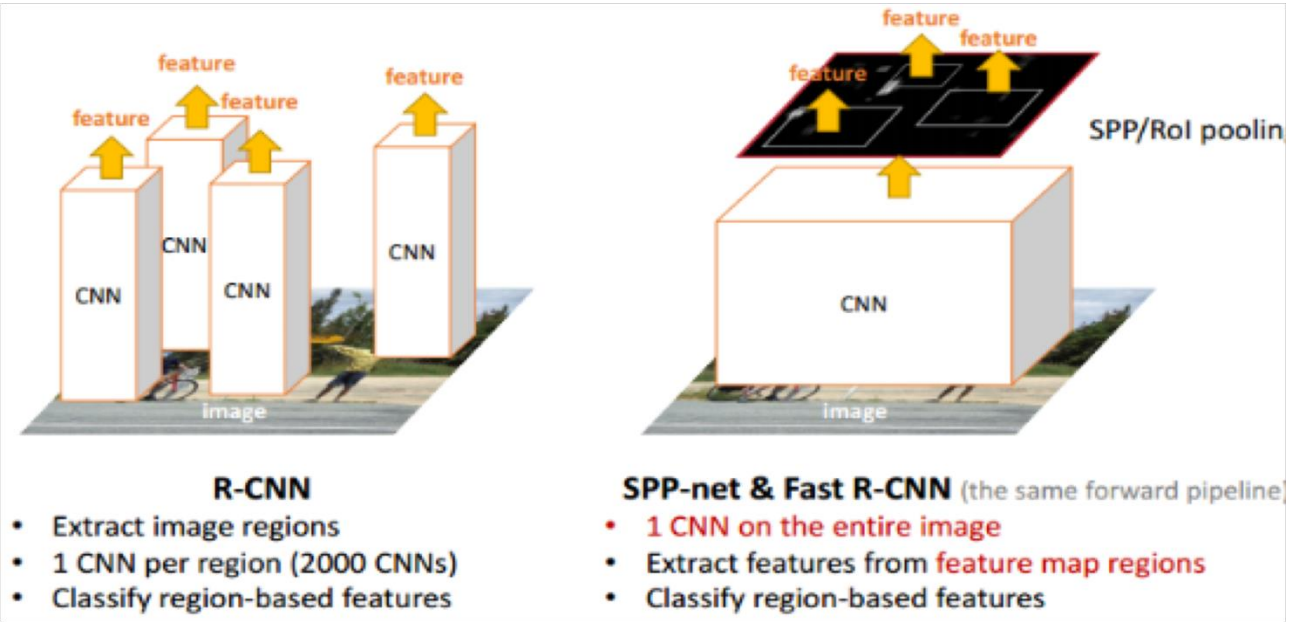


- Candidate Region Generation
  - ~2k candidate regions in one picture (selective search)
  - Proposed regions are cropped and warped to a fixed-size 227x227 image
- Feature Extraction
  - Apply alexnet on each region: 4096 features (fc7)
- Classification
  - SVM classifier given extracted features
- Bounding Box Regression
  - Using regression to amend the bounding box location

Spatial Pyramid Pooling (SppNet)



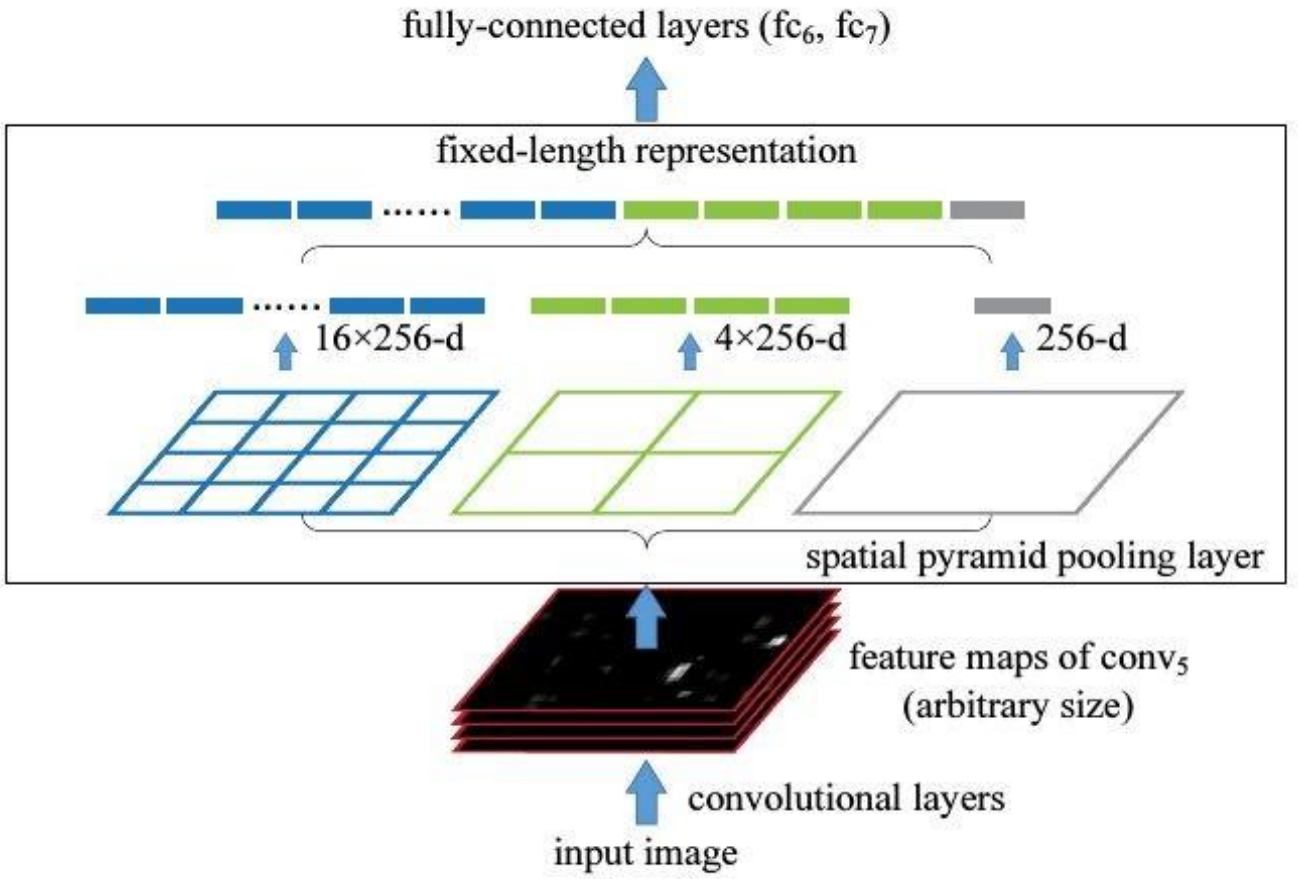
- SPPnet calculates feature once and use SPP to obtain a fix-length feature vector



- two orders of magnitude speed up.

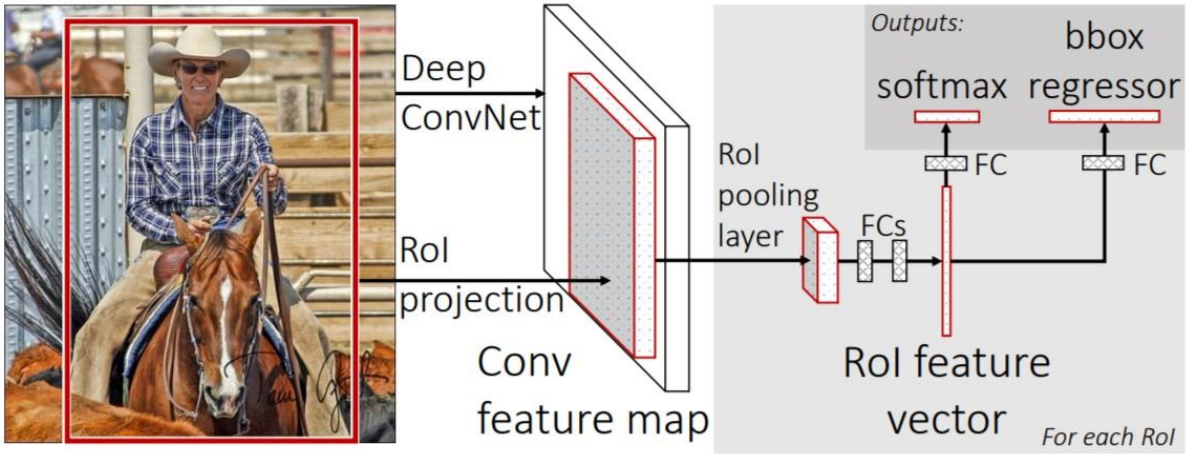
"Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition" Kaiming He et. al

Spatial Pyramid Pooling (SppNet)



- Image with 224x224, after CONV layers, output: 13*13*256. Output: (16+4+1)x256;
- Different Images size, same output size.

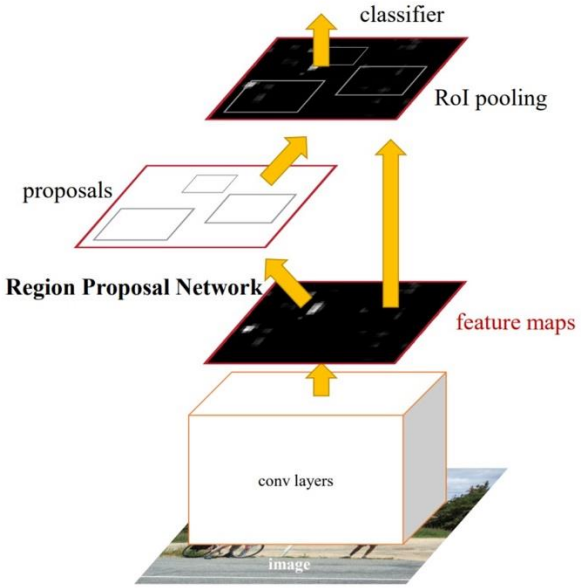
Fast R-CNN



- Contribution:
  - Essentially SPPNet with trainable feature extraction network.
  - Region-of-Interest (**RoI: given boundaries on an image of an object or on a drawing**) from same sampled image share computation
  - Joint Training: Train bbox offset and classifier (use softmax) in one stage using multi-task loss

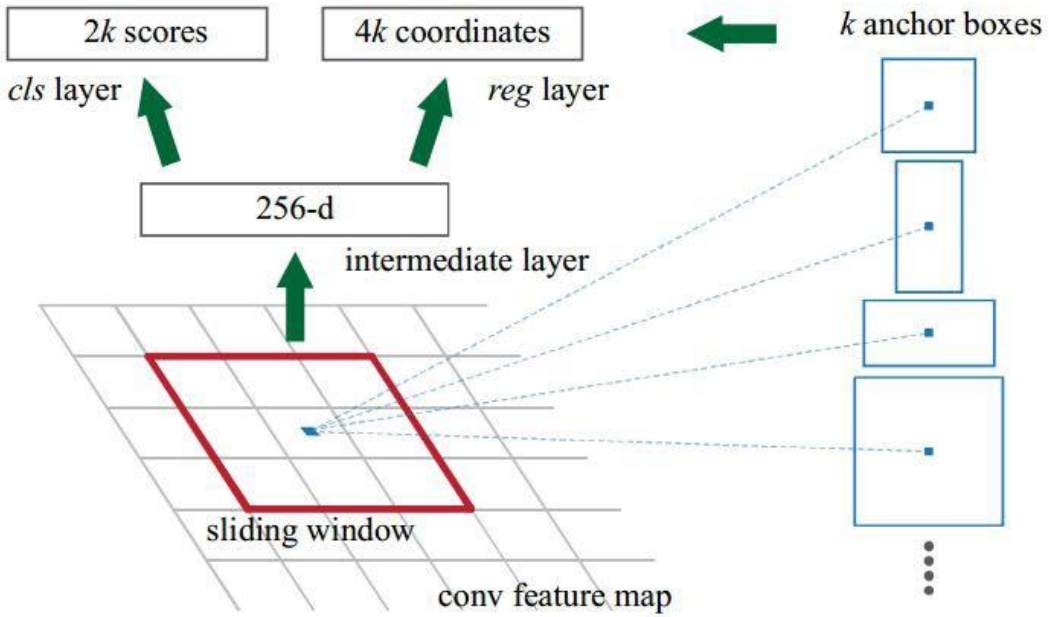
Faster R-CNN

- Faster R-CNN = fast R-CNN + RPN
- Implement region proposal using CNN on GPU, complete end-to-end model
- Propose Region-Proposal-Network (RPN).
  - A small ConvNet looking at global feature volume in the sliding window
  - Each sliding window has 9 anchor boxes;
  - Bounding box regression and box confidence scoring.



"Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks " Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Faster R-CNN



Propose Region-Proposal-Network (RPN).

1. A small ConvNet looking at global feature volume in the sliding window
2. Each sliding window has 9 anchor boxes;
3. Bounding box regression and box confidence scoring.

Where are we?

- R-CNN 47s/image 66% mAP in ECCV 2014

  Fast R-CNN 3s/image 70% Map in ICCV 2015

  Faster R-CNN 5fps 73.2% Map in NIPS 2015

- Lets get faster!!

YOLO (You only look once)

- Main feature:
  - Fast, 45 fps on NVIDIA Titan X 63.4% mAP (fast YOLO, 9 conv 155 fps, 52.7%)
  - Use complete image as context, rarely mis-predict background as object
  - Good generalization when applied to new domains or unexpected inputs

	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
YOLO	63.4	45 FPS	22 ms/img

"You Only Look Once: Unified, Real-Time Object Detection" Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

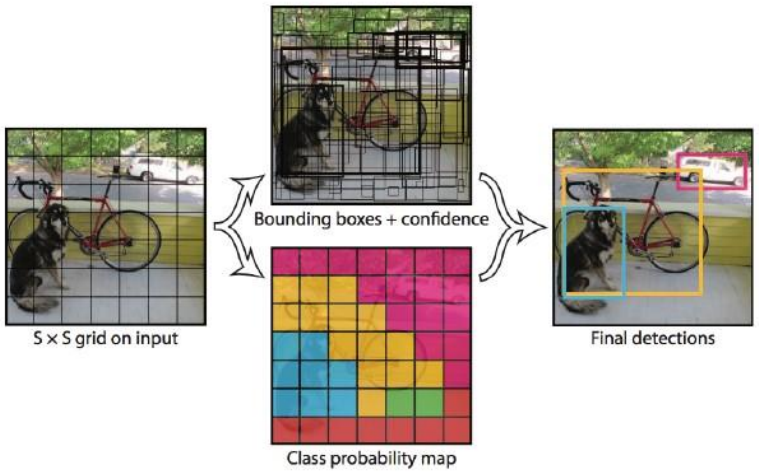
YOLO (You only look once)

Unified Detection

- Divide the image into S x S grid.
- Each grid cell predict
  - B bounding boxes (x, y, width, height)
  - B confidence scores as $C = \Pr(Object) * IOU_{pred}^{truth}$
  - C conditional class probabilities as $P = \Pr(Class_i|Object)$
  - Total Prediction as S x S x (B x 5 + C)
- At test time we can get class-specific confidence scores for each box as

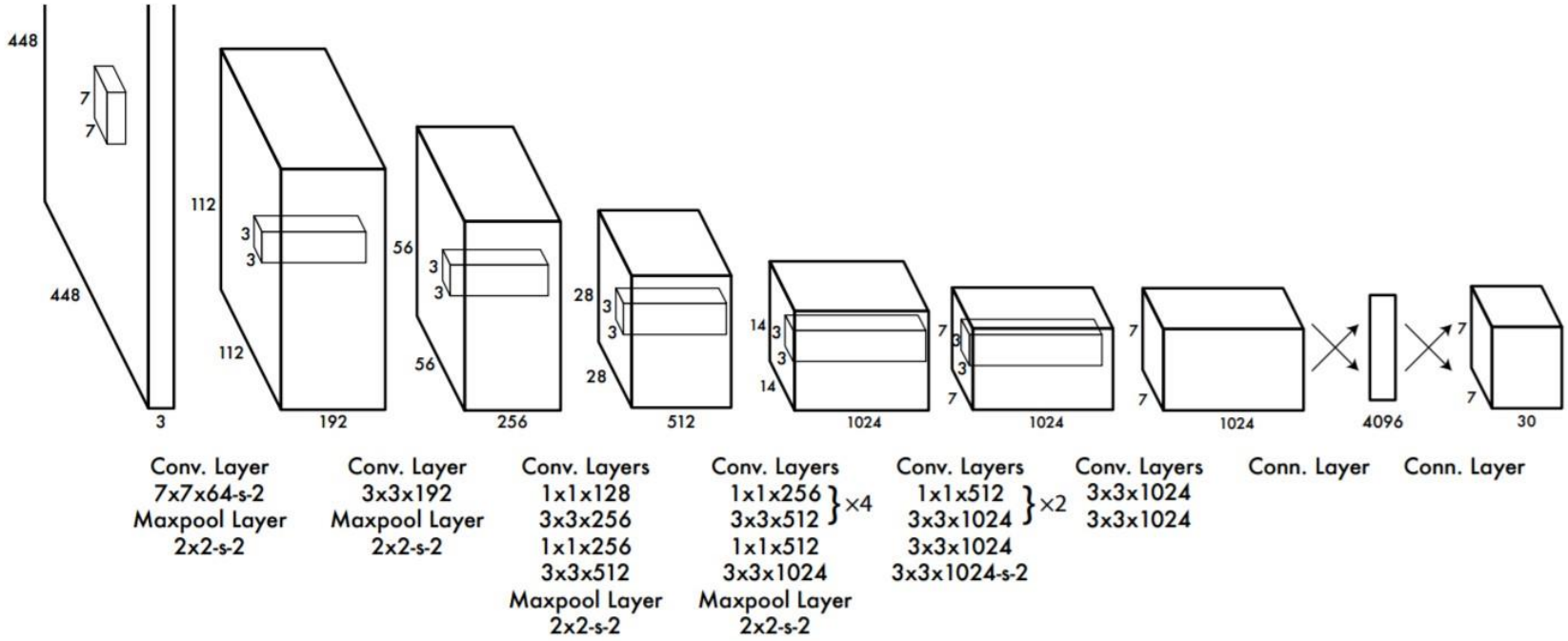
$$\Pr(Class_i|Object) * \Pr(Object) * IOU_{pred}^{truth} = \Pr(Class_i) * IOU_{pred}^{truth}$$

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



YOLO Network Design

24 conv layers and 2 fc layers



YOLO Inference & Limitation

Grid classification * bounding box confidence = class-specific bounding box confidence score $\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$

Set threshold to filter out the bbox with low scores

NMS for remaining bbox to get final result

Limitation:

- Bad for objects in one grid or close;
- Bad generalization for objects with unusual aspect ratio;

YOLO v2

- No FC layers;
- Add batch normalization;
- Using anchor boxes;
  - ➢ More recall; No accuracy degradation
- Shrink input images from 448 to 416;

	PASCAL 2007 mAP (%)	Speed
YOLO	63.4	45 FPS, 22ms/img
YOLOv2	76.8	67 FPS, 15ms/img

YOLOv2

Faster

- 19 Convolutional layers
- 5 max pooling layers
- 5.58 billion operations

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

YOLOv3

YOLOv3: An Incremental Improvement

Joseph Redmon Ali Farhadi
University of Washington

Abstract

We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more accurate. It's still fast though, don't worry. At 320 × 320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves 57.9 AP_{50} in 51 ms on a Titan X, compared to 57.5 AP_{50} in 198 ms by RetinaNet, similar performance but 3.8× faster. As always, all the code is online at https://pjreddie.com/yolo/.

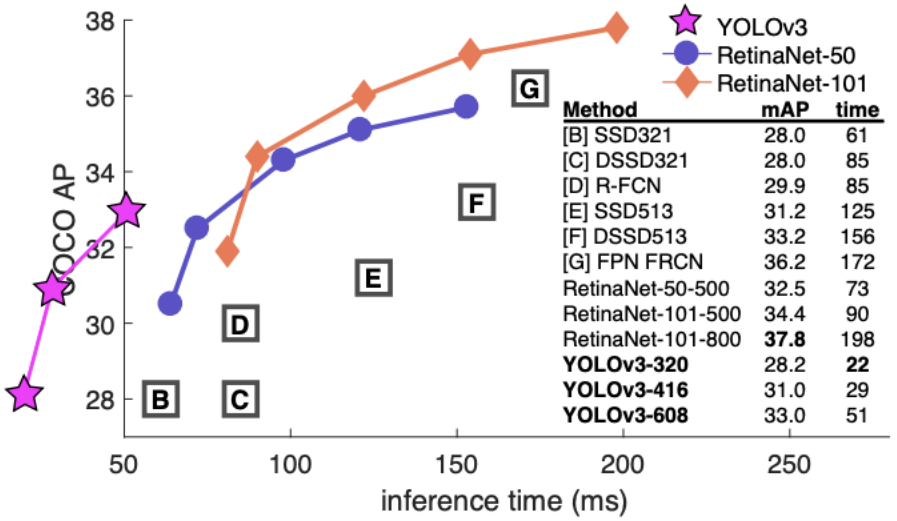


Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

1. Introduction

YOLOv3

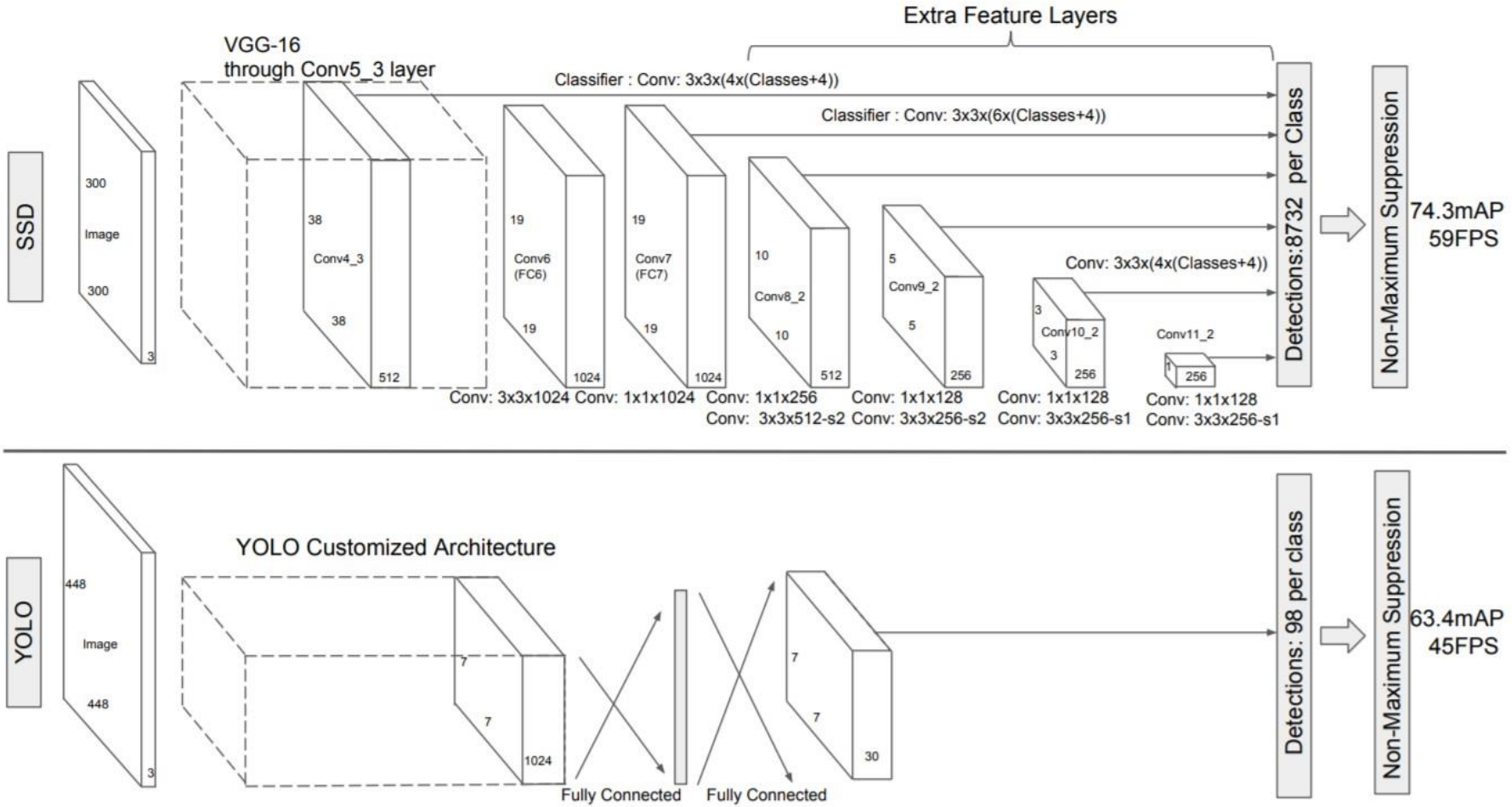
1. Introduction

Sometimes you just kinda phone it in for a year, you know? I didn't do a whole lot of research this year. Spent a lot of time on Twitter. Played around with GANs a little. I had a little momentum left over from last year [12] [1]; I managed to make some improvements to YOLO. But, honestly, nothing like super interesting, just a bunch of small changes that make it better. I also helped out with other people's research a little.

Actually, that's what brings us here today. We have a camera-ready deadline [4] and we need to cite some of the random updates I made to YOLO but we don't have a source. So get ready for a TECH REPORT!

The great thing about tech reports is that they don't need intros, y'all know why we're here. So the end of this introduction will signpost for the rest of the paper. First we'll tell you what the deal is with YOLOv3. Then we'll tell you how we do. We'll also tell you about some things we tried that didn't work. Finally we'll contemplate what this all means.

SSD: Single Shot MultiBox Detector



SSD: Single Shot MultiBox Detector

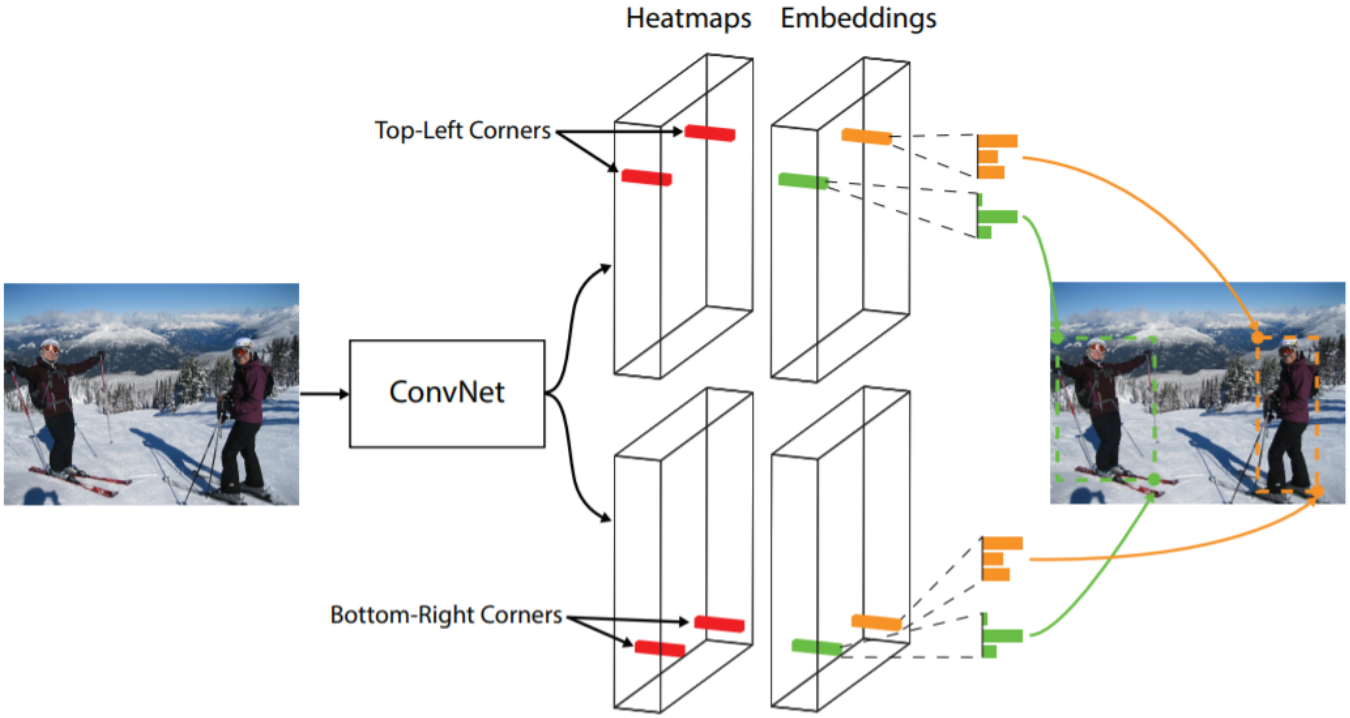
Better than YOLO in:

- Higher accuracy;
- Good for objects in one grid or close;
- Good generalization for objects with unusual aspect ratio;

SSD leverages the Faster RCNN's RPN, using it to directly classify object inside each prior box instead of just scoring the object confidence (similar to YOLO).

It improves the diversity of prior boxes' resolutions by running the RPN on multiple conv layers at different depth levels.

Recent Progress in Object Detection Anchor-Free Methods

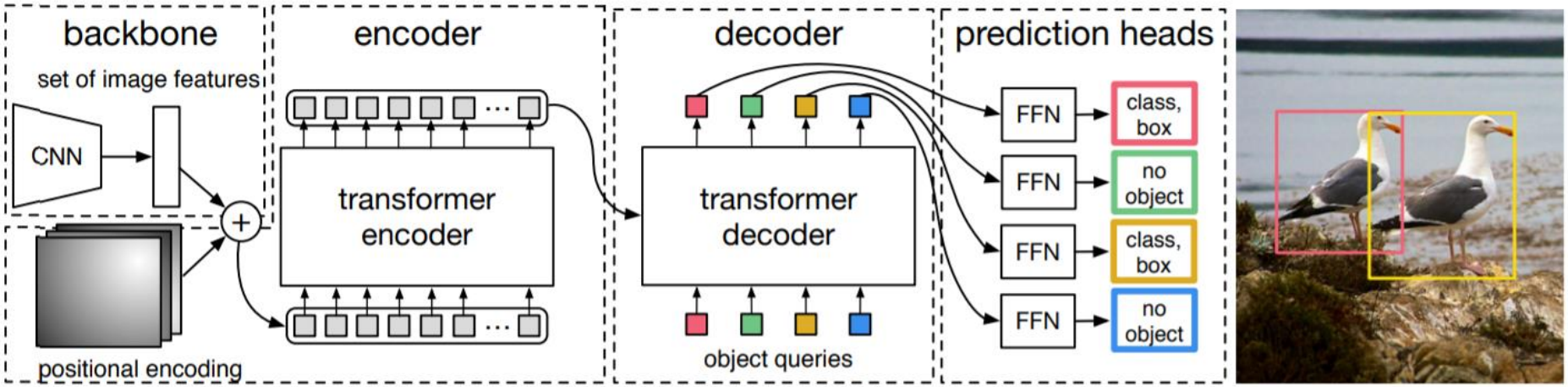


CornerNet: Detecting Objects as Paired Keypoints

- In Faster RCNN and YOLO, each object is recognized with an anchor. Anchor-based methods are sensitive to datasets, and consume a large amount of computation.
- CornerNet replace anchors with corners (Top-left and Bottom-right).

https://arxiv.org/pdf/1808.01244.pdf

Recent Progress in Object Detection
DETR: Object Detection with Transformer

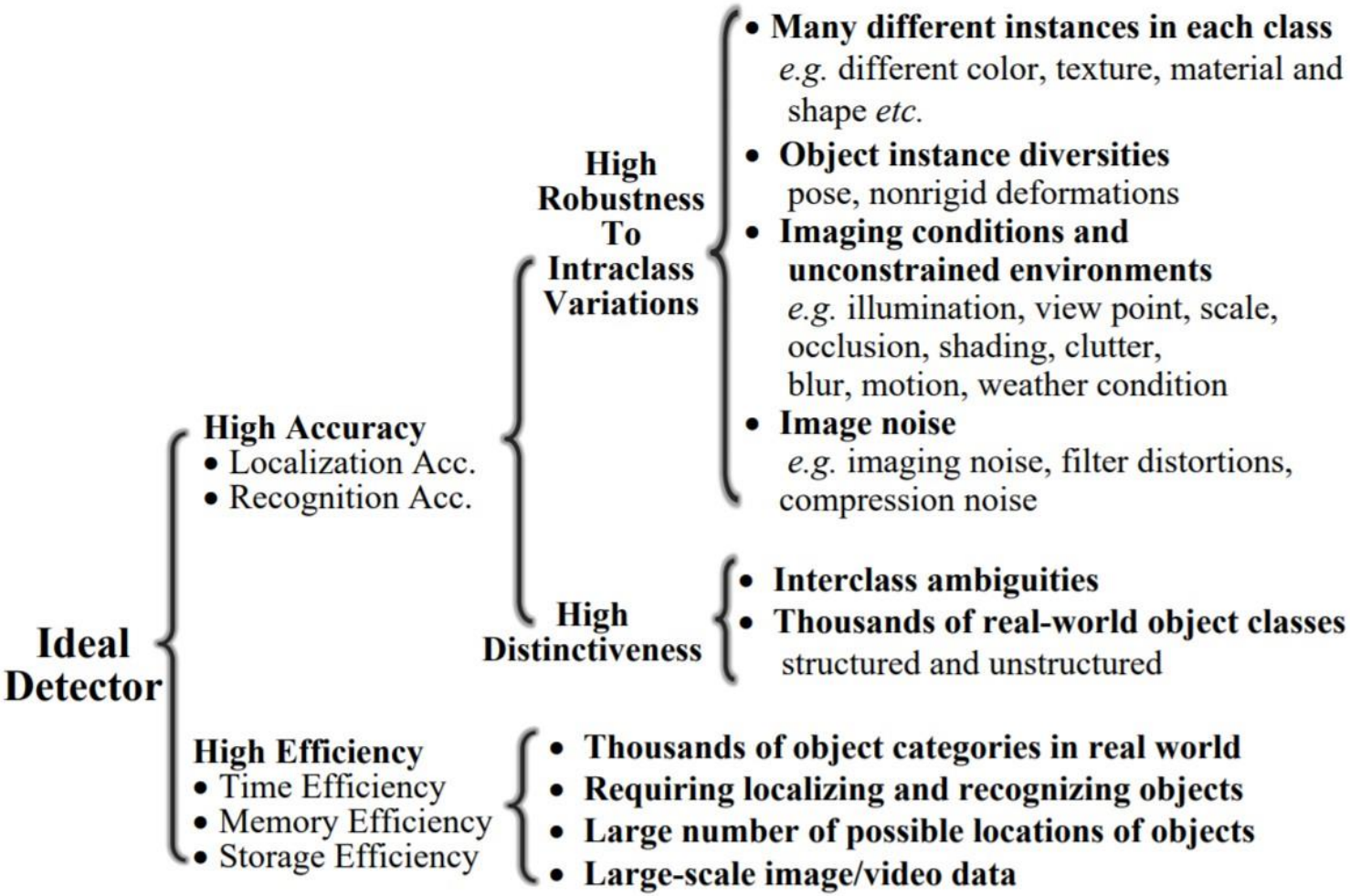


End-to-End Object Detection with Transformers

- Extract features with CNN. Then, features and position encoding are inputted into the encoder of a transformer.
- The encoded features and object queries are inputted into the decoder. The decoder compute whether there is an object in the query.
- DETR is the first network which regard object detection as a sequential task, and solves it with Transformer.

https://arxiv.org/pdf/2005.12872.pdf

Conclusion



Summary of challenges in generic object detection

https://arxiv.org/pdf/1809.02165.pdf