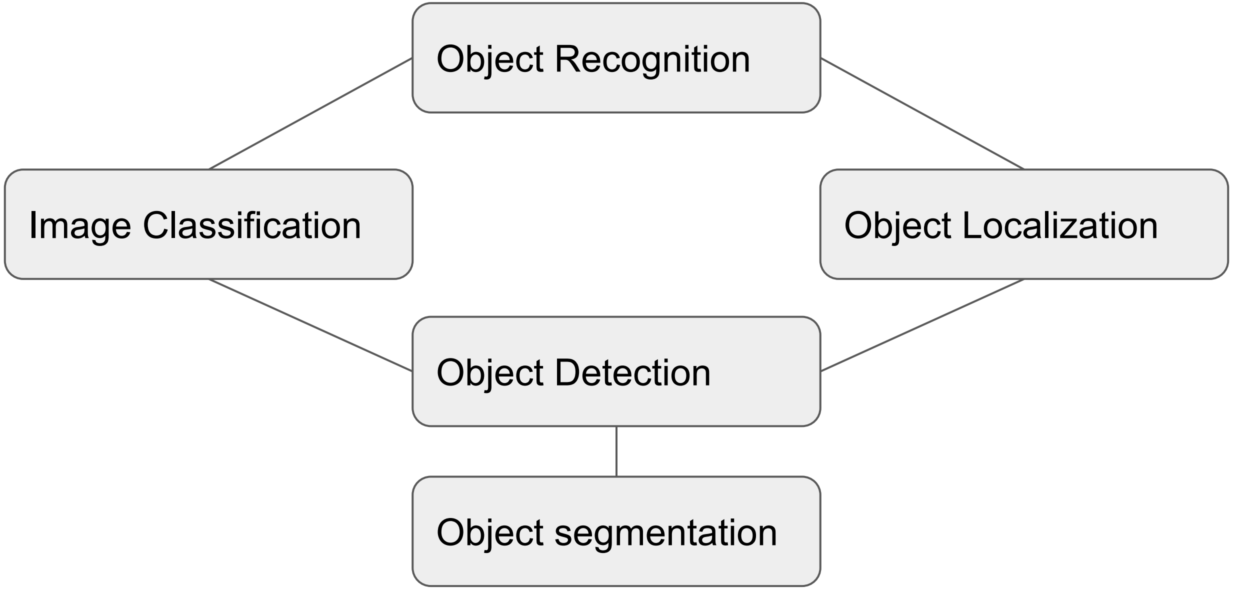
The history of YOLO

Shang-Ching Liu

Contents

What is Object Detection?

We will be using the term object recognition broadly to encompass both image classification (a task requiring an algorithm to determine what object classes are present in the image) as well as object detection (a task requiring an algorithm to localize all objects present in the image — ImageNet Large Scale Visual Recognition Challenge, 2015.



Popular Datasets

COCO

Figure 1. Dataset example,
Capture from https://cocodataset.org/#home

What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

COCO and bounding box

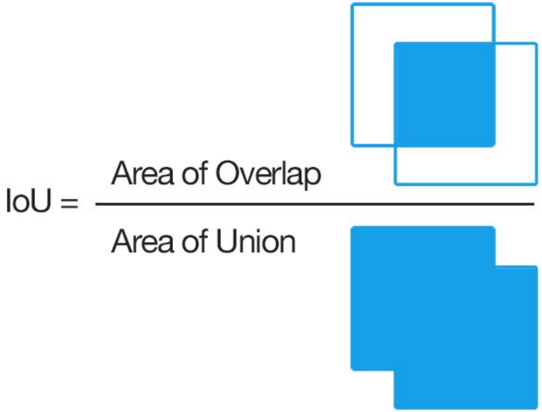
- COCO segmentation format is a list of polygons as RED outline in the figure.
  - each polygon looks like [x0, y0, x1, y1, x2, y2… xn, yn].
- The bounding box get the region out of polygon as [x, y, width, height]

Figure 2.Bounding box from COCO,
https://github.com/cocodataset/cocoapi/issues/102

Average Precision(AP)

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

- Precision = True Positive / (True Positive + False Positive)
- Recall = True Positive / (True Positive + False Negative)
- The average precision can measure different cut-off threshold of IoU.
- AP50? AP75?



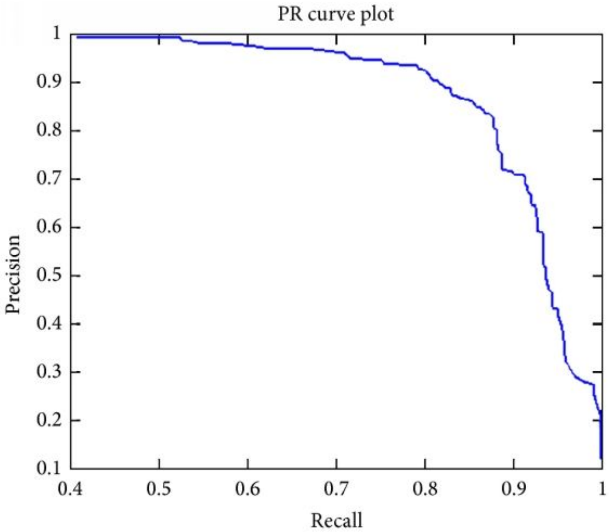


Figure 3. AP Explanation,
Capture from https://yanfengliux.medium.com/the-confusing-metrics-of-ap-and-map-for-object-detection-3113ba0386ef

COCO minival

- 5K images
- The lines show the best project of the time

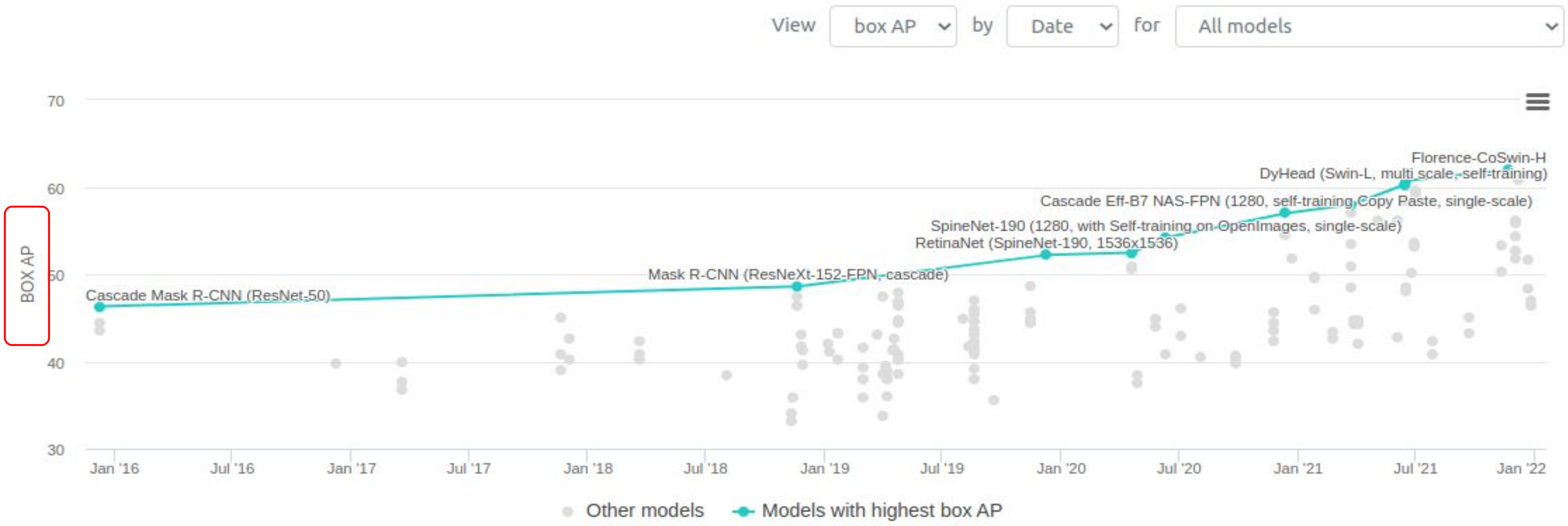


Figure 4.COCO minival benchmark,

Capture from https://paperswithcode.com/sota/object-detection-on-coco-minival

COCO test-dev

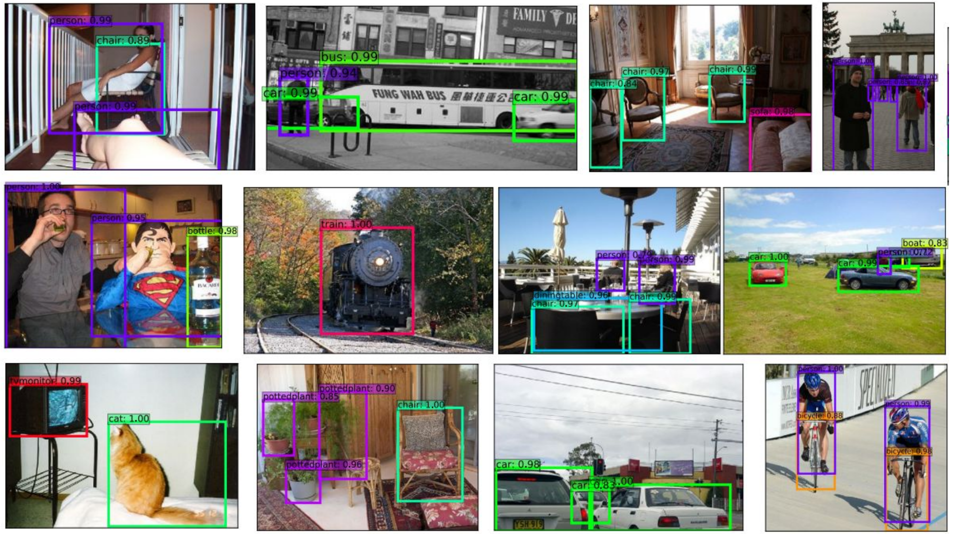
- 20K images

Figure 5.COCO test-dev,

Capture from https://paperswithcode.com/sota/object-detection-on-coco-minival

Pascal VOC 2007

- Pattern Analysis, Statistical Modelling and Computational learning Challenge



Class	#train	#val	#test	Class	#train	#val	#test
1: aeroplane	112	126	204	11: dining table	97	103	190
2: bicycle	116	127	239	12: dog	203	218	418
3: bird	180	150	282	13: horse	139	148	274
4: boat	81	100	172	14: motorbike	120	125	222
5: bottle	139	105	212	15: person	1025	983	2007
6: bus	97	89	174	16: potted plant	133	112	224
7: car	376	337	721	17: sheep	48	48	97
8: cat	163	174	322	18: sofa	111	118	223
9: chair	224	221	417	19: train	127	134	259
10: cow	69	72	127	20: tv/monitor	128	128	229

Figure 6. Dataset example like person, animal, bus, car, chair etc.

Capture from https://arxiv.org/pdf/1708.01241.pdf

Pascal VOC 2007

- Benchmark

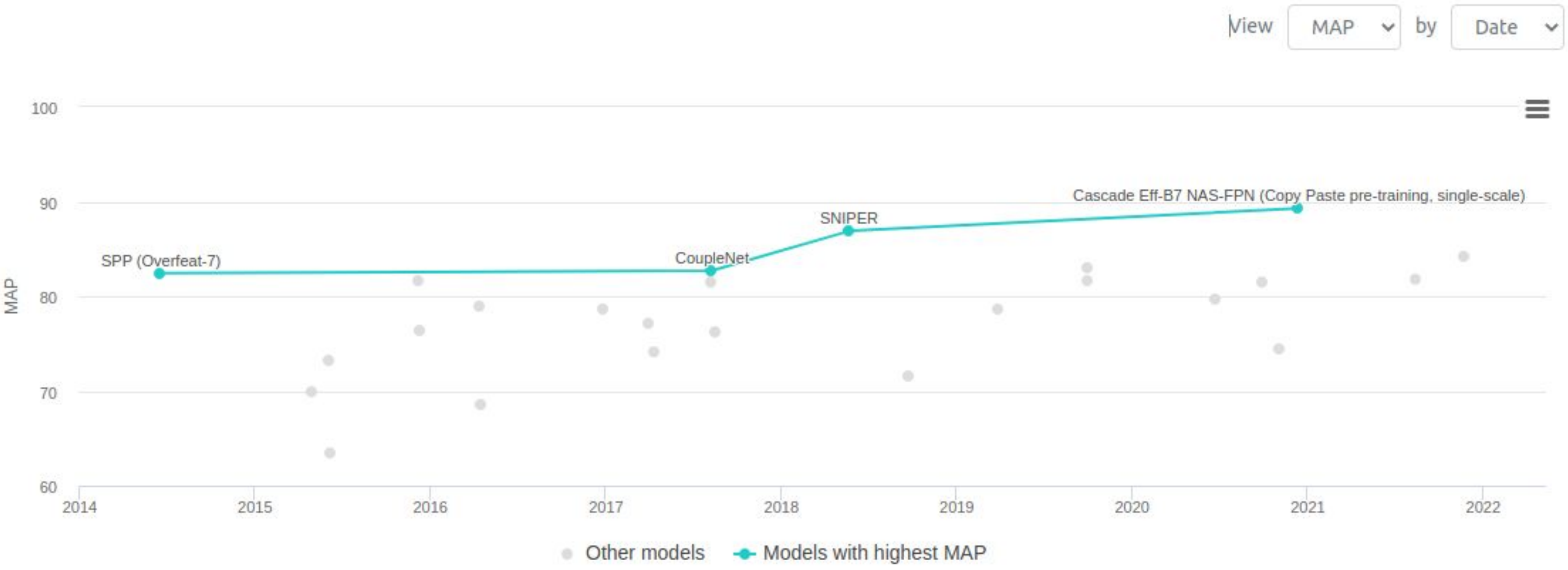


Figure 7.Pascal VOC benchmark,

Capture from https://paperswithcode.com/sota/object-detection-on-pascal-voc-2007

Others

- CrowdHuman

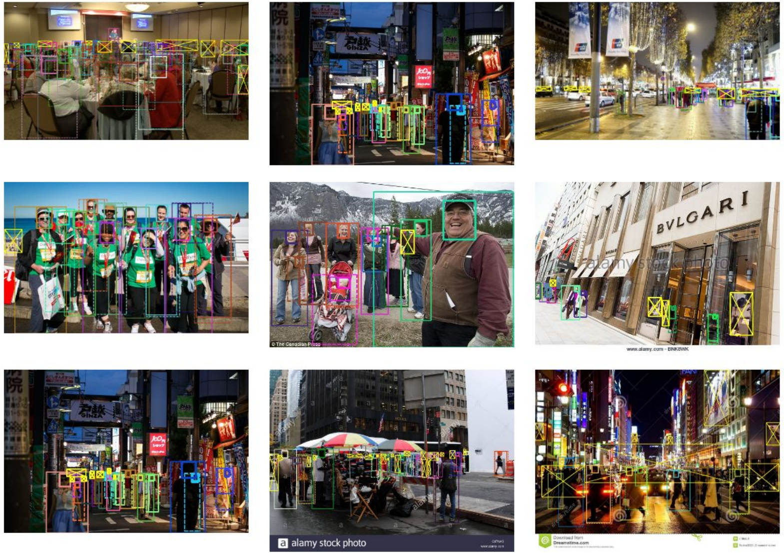


Figure 8.CrowdHuman ,
Capture from https://paperswithcode.com/sota/object-detection-on-crowdhuman-full-body

- KITTI Cars

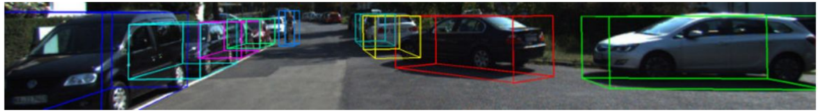


Figure 9.KITTI Cars ,
Capture from https://paperswithcode.com/sota/3d-object-detection-on-kitti-cars-moderate

- ScanNet

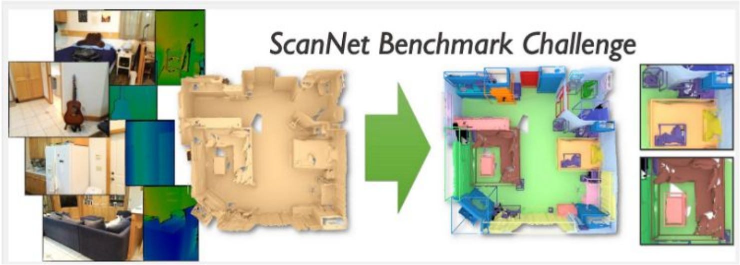
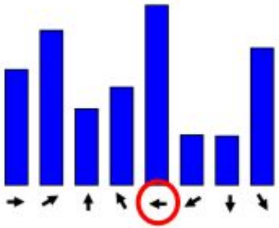


Figure 10.ScanNet , Capture from http://www.scan-net.org/

Object Detection evolution along YOLO

What is before? (Traditional Method)

- Local feature by orientation



(for simplicity only 8 bins displayed)

Figure 11.Orientation illustration1 , Capture from https://lernen.min.uni-hamburg.de/pluginfile.php/176626/mod_page/content/13/CV1-07-Features1.pdf

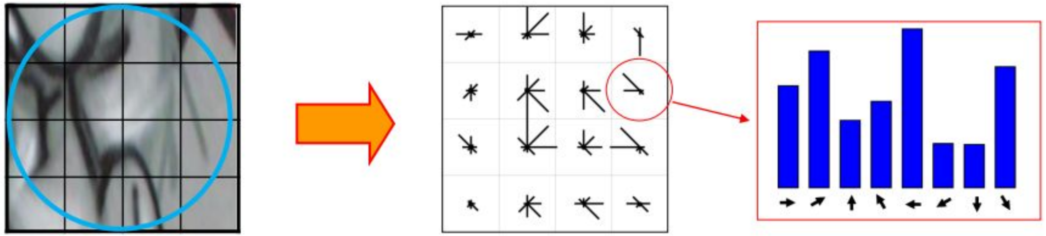


Figure 12.Orientation illustration2 , Capture from https://lernen.min.uni-hamburg.de/pluginfile.php/176626/mod_page/content/13/CV1-07-Features1.pdf

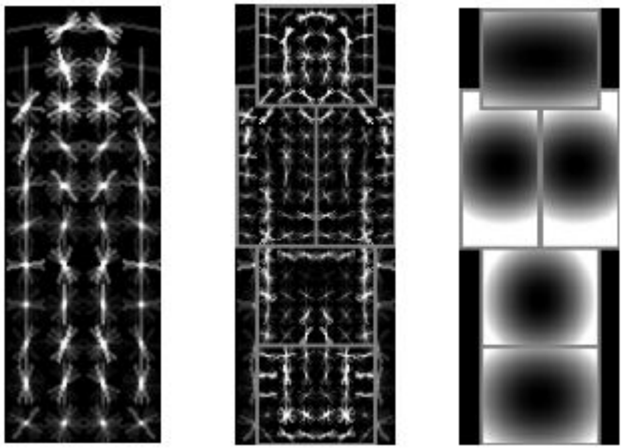
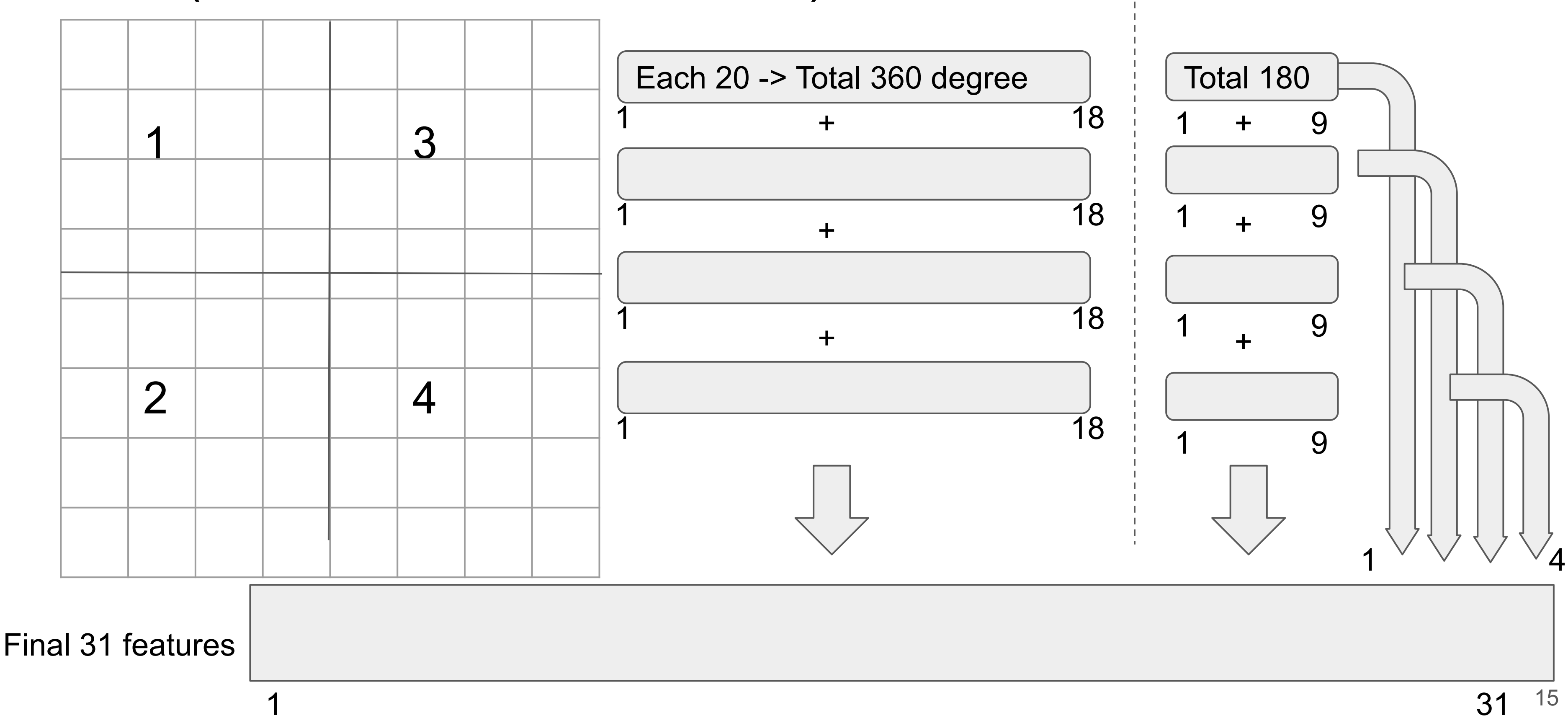


Figure 13. Orientation visualization, Capture from https://lernen.min.uni-hamburg.de/pluginfile.php/176626/mod_page/content/13/CV1-07-Features1.pdf

DPM(Deformable Parts Model)



Results

- Idea toward modern technique
    - Local features
    - Low-level integration of images patches

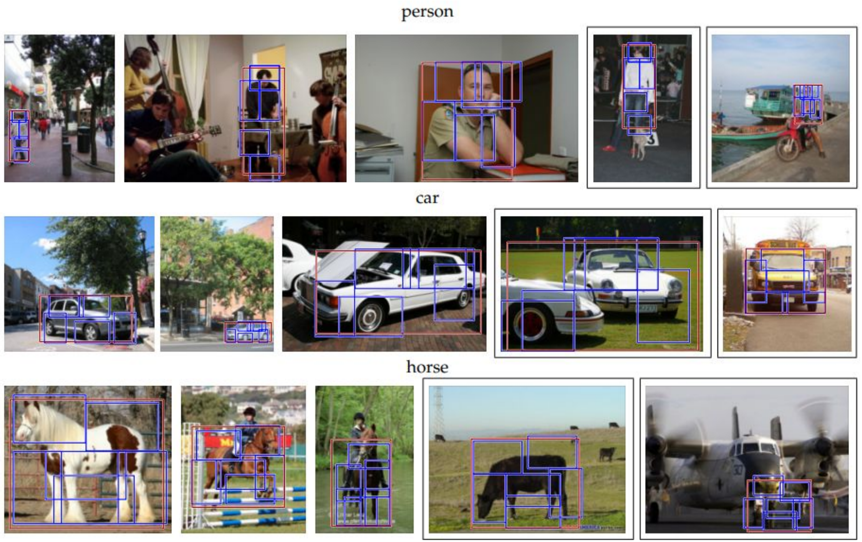


Figure 14.DPM result Image,

Capture from http://cs.brown.edu/people/pfelzens/papers/lsvm-pami.pdf

YOLO

Background Knowledge

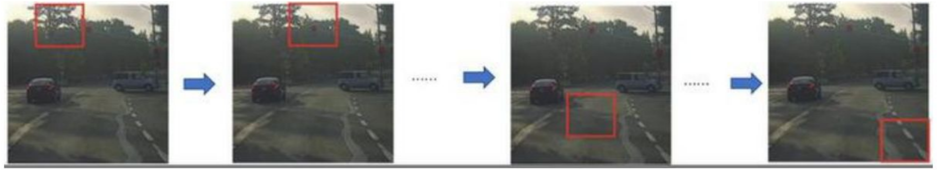


Figure 14.Sliding Windows, Capture from deeplearning.ai

- Sliding windows is the window to slide thought the whole picture.
  - Classify each small region, similar to idea in DPM.
  - Classify each region one by one is not efficient
    - Using Selection Region as R-CNN: Filter the irrelevant parts
    - Using computing process as CNN to parallel computing multi regions

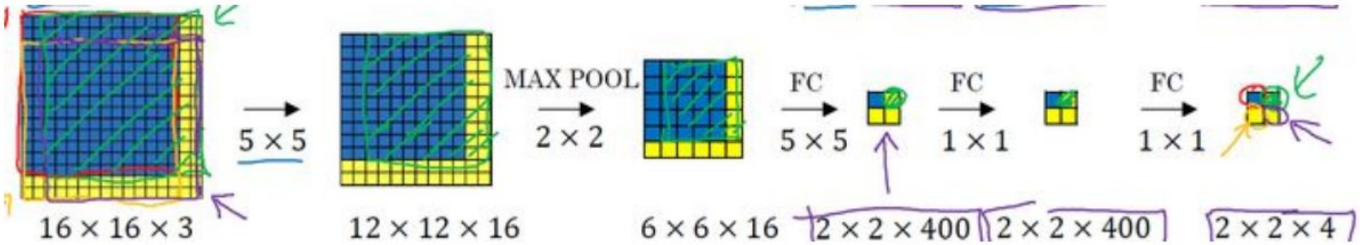


Figure 15.Sliding Windows with CNN implementation, Capture from deeplearning.ai

YOLO

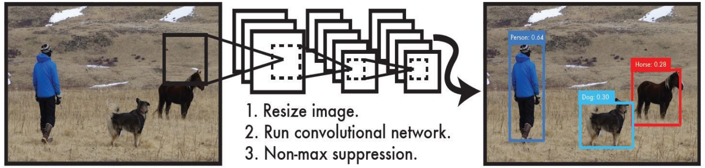


Figure 16.YOLO capture from https://arxiv.org/abs/1506.02640

- Yon Only Look Once
  - Efficient to find objects in the picture
  - Hard to detect the overlap objects because each region apply ONLY ONE OBJECT.

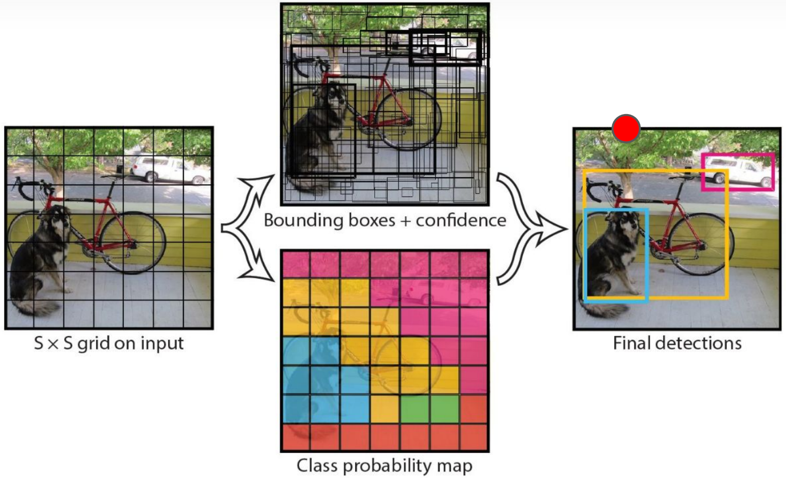


Figure 17.YOLO structure from https://zhuanlan.zhihu.com/p/32525231

Class confidence box(x,y,w,h)

1 20 21 22 25

NMS(Non Maximum Suppression)

- Choose the best bonding box for one object

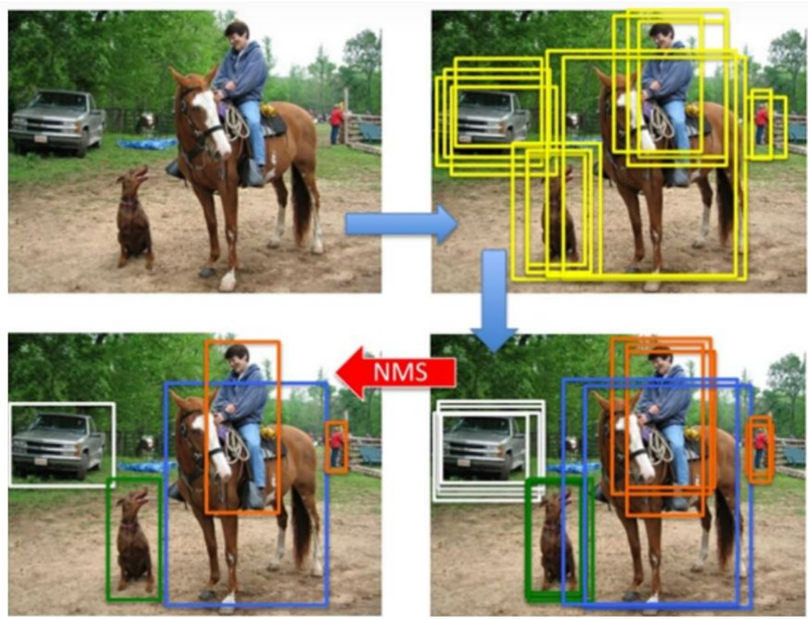
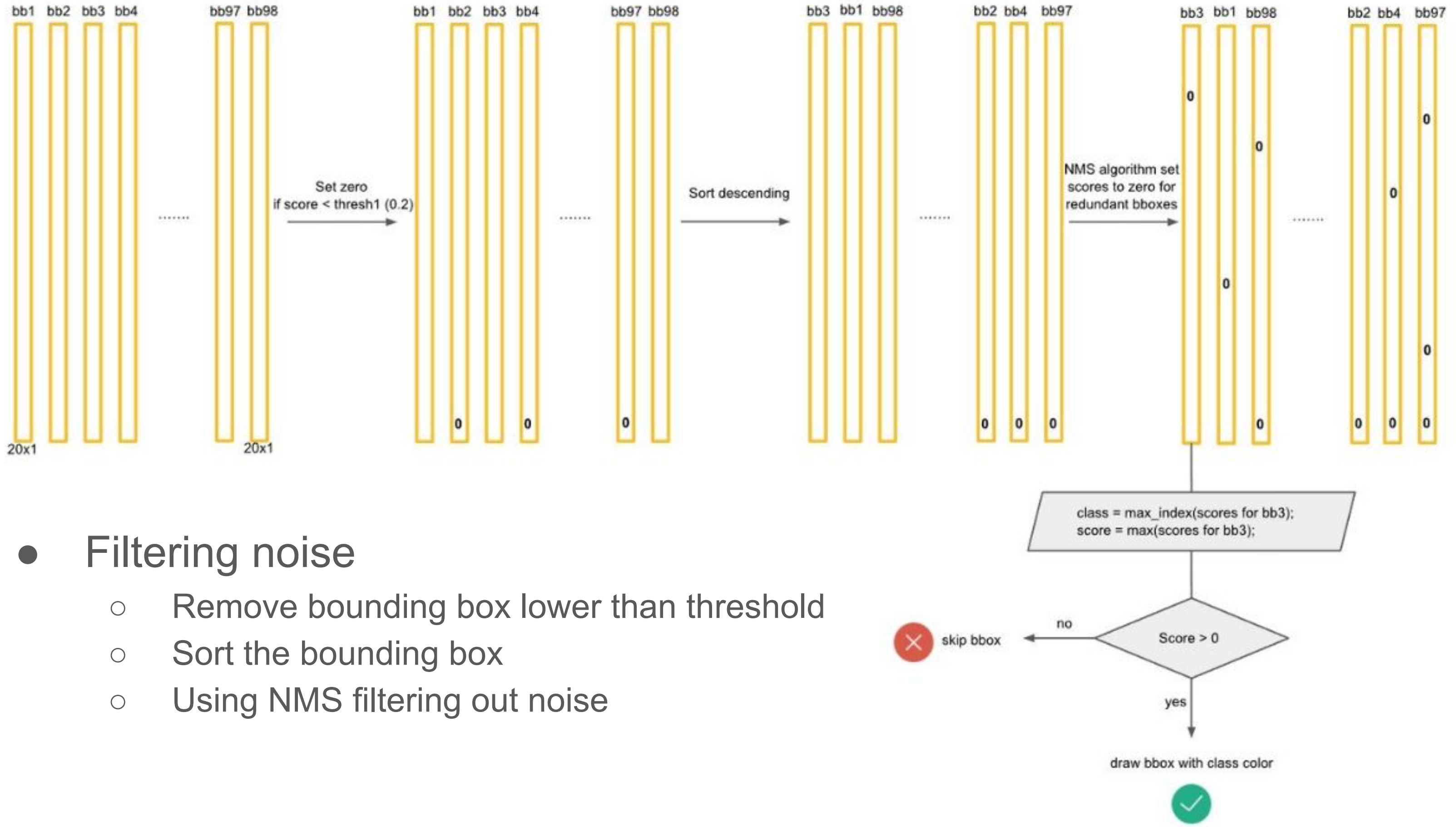


Figure 18.NMS structure from https://zhuanlan.zhihu.com/p/32525231

Figure 19.YOLO predict process with NMS from https://zhuanlan.zhihu.com/p/32525231



- Filtering noise
  - Remove bounding box lower than threshold
  - Sort the bounding box
  - Using NMS filtering out noise

YOLO Comparison

- 30FPS is the standard for web-based videos
- YOLO get a perfect mAP.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 20.YOLO benchmark from https://zhuanlan.zhihu.com/p/32525231

YOLO V2

Background knowledge

- Using multi anchor box to capture overlap objections

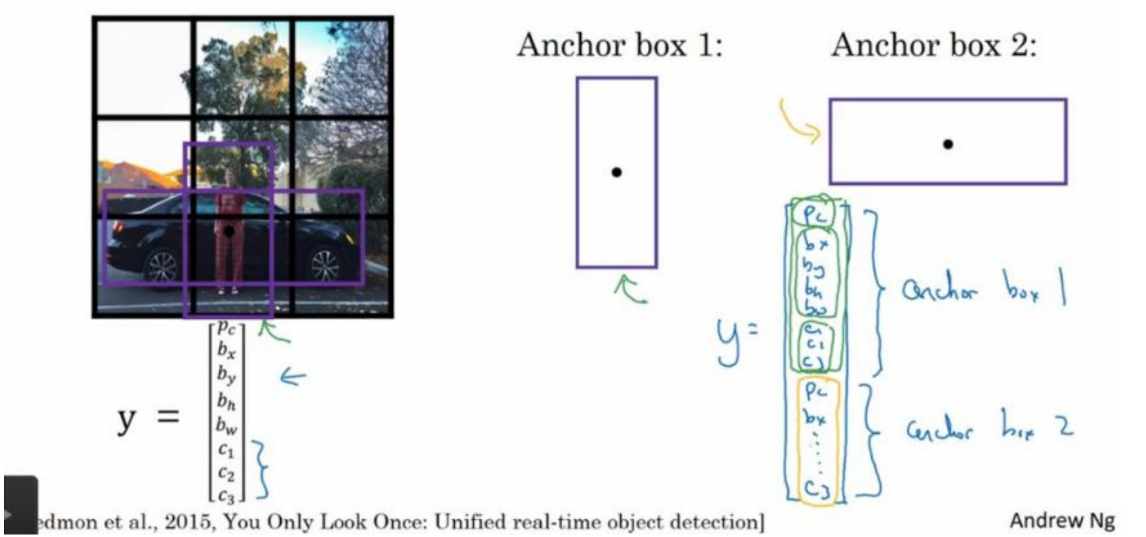


Figure 21.Multiple Anchor box Illustration Capture from https://zhuanlan.zhihu.com/p/31292482

Backbone

- Darknet-19 Compare to GooleNet (YOLO v1) have less convolutional layer
  - Darknet-19: convolutional layer + 5 max pooling layer
  - GooleNet: 24 convolutional layer + 2 full connected layer

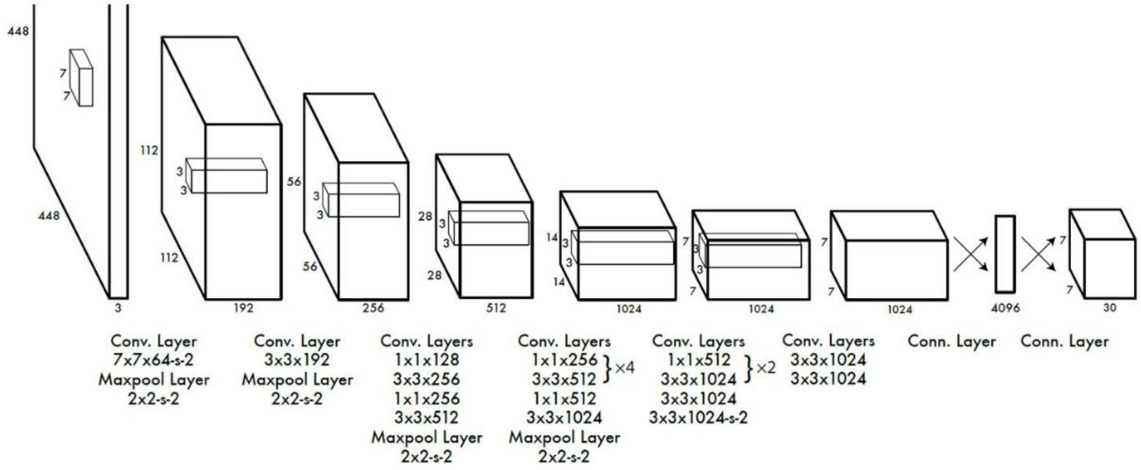


Figure 22. YOLO Network structure,

Capture from https://zhuanlan.zhihu.com/p/31292482

Improvement

- Higher resolution + Batch Normalization
  - 224*224 → 448*448(ImageNet)

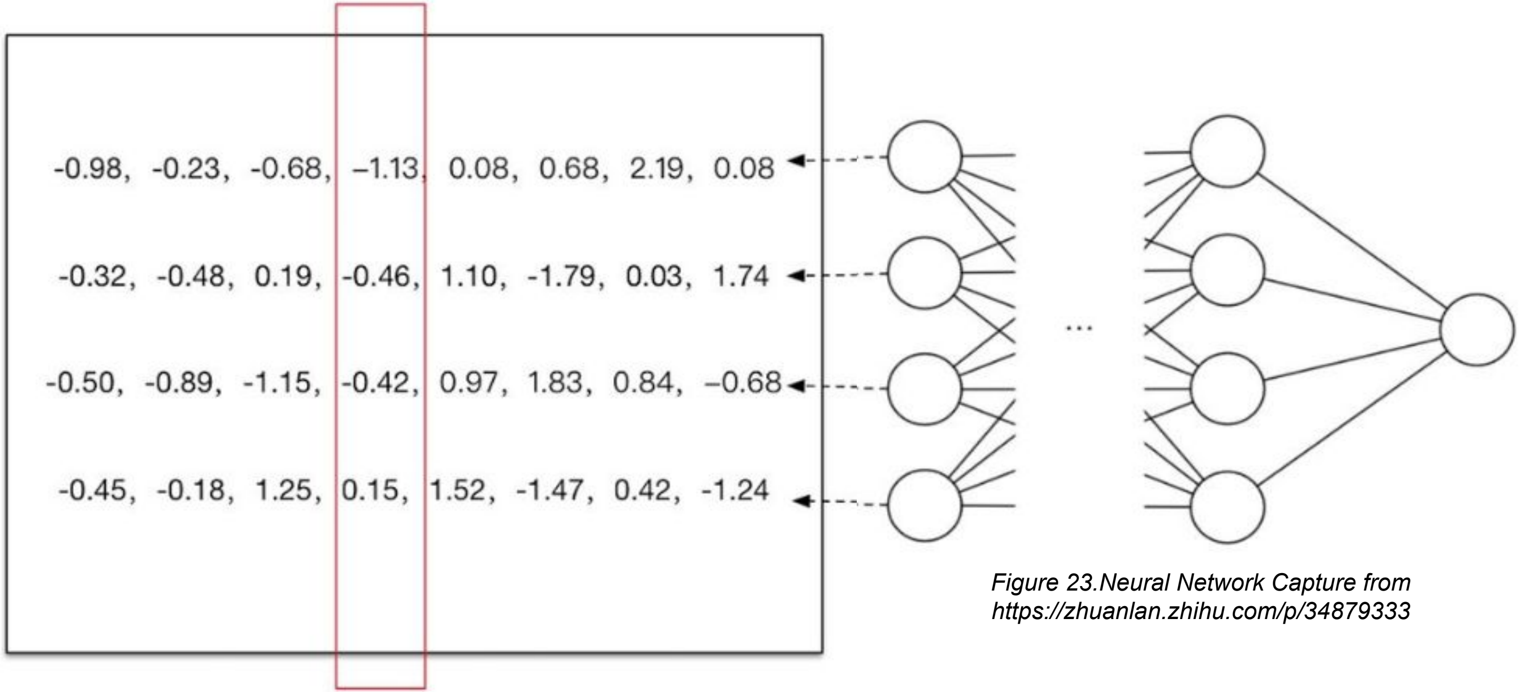


Figure 23.Neural Network Capture from https://zhuanlan.zhihu.com/p/34879333

YOLO V3

Background Knowledge

- FPN(Feature Pyramid Networks)
  - Combine different levels of features

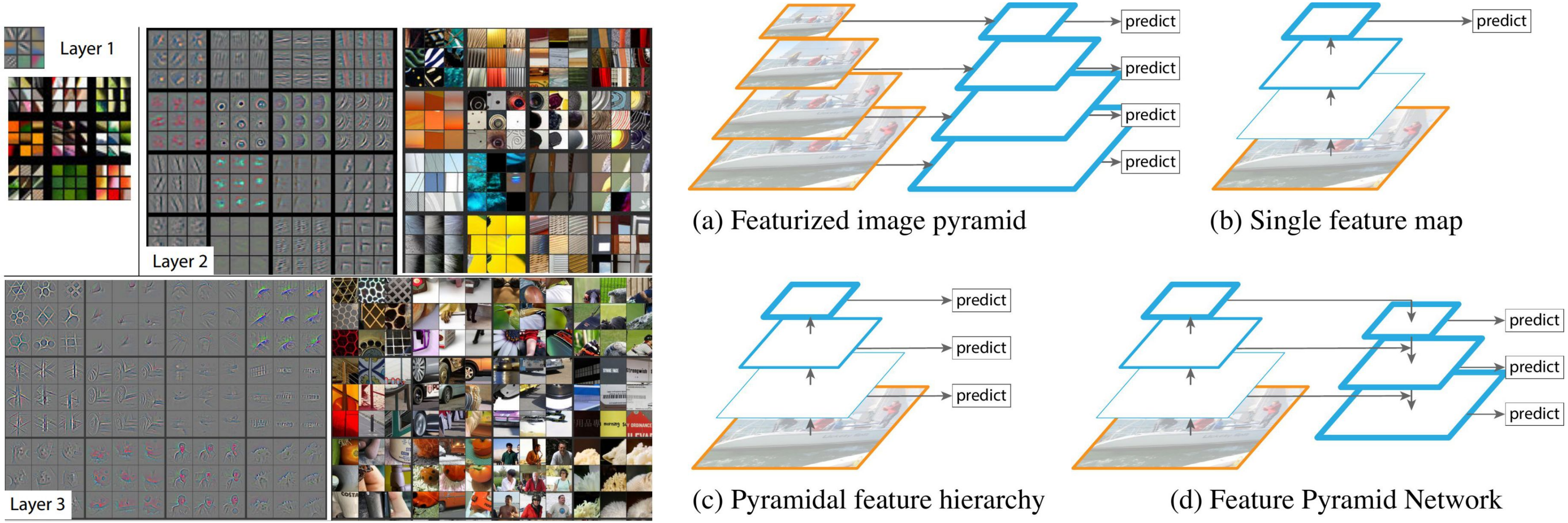


Figure 24. FPN Illustration Capture from https://paperswithcode.com/method/fpn

Residual learning

- Residual Learning → Learning Target change from F(x) to F(x) + x = O(x)
- The whole framework learned O(x) - x (Output - Input)

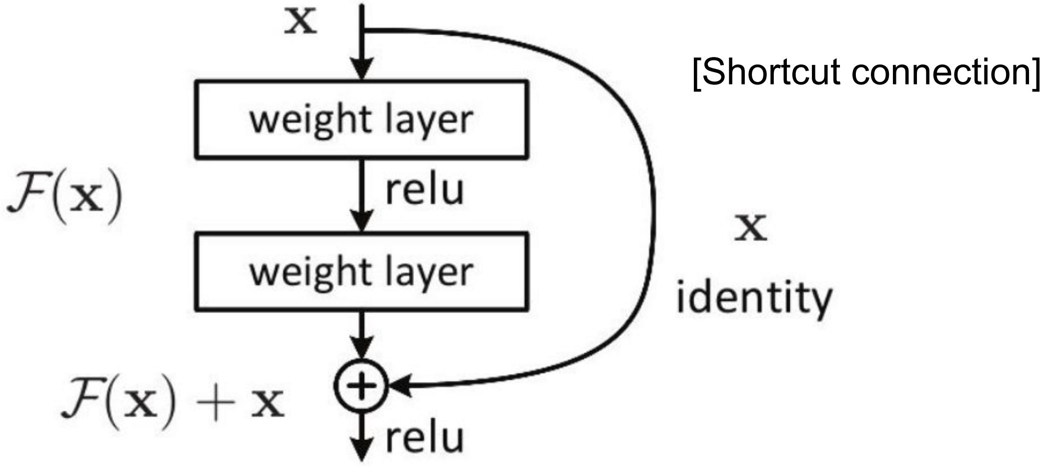


Figure 25. Residual Learning Illustration Capture from https://www.zhihu.com/question/64494691

DenseNet

- DenseNet
  - The extreme type of using ResNet
  - Use Much more memory

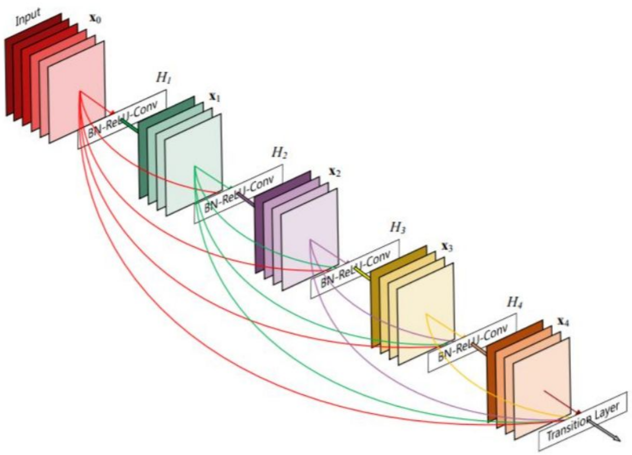


Figure 26. DenseNet core idea illustration Capture from https://www.zhihu.com/question/64494691

Dense Exper	Params	Cifar 10	Cifar 10+	Cifar 100	Cifar 100+
Res-164	1.7M	11.26	5.46	35.58	24.33
Res-1001	10.2M	10.56	4.62	33.47	22.71
Dense-40-12	1.0M	7.00	5.24	27.55	24.42
Dense-100-12	7.0M	5.77	4.10	23.79	20.20
Dense-100-24	27.2M	**5.83**	**3.74**	**23.42**	**19.25**

Figure 27. DenseNet memory Usage Capture from https://www.zhihu.com/question/60109389

Improvement combination

- FPN(Pass different scale) + ResNet(Add) + DenseNet(concat)

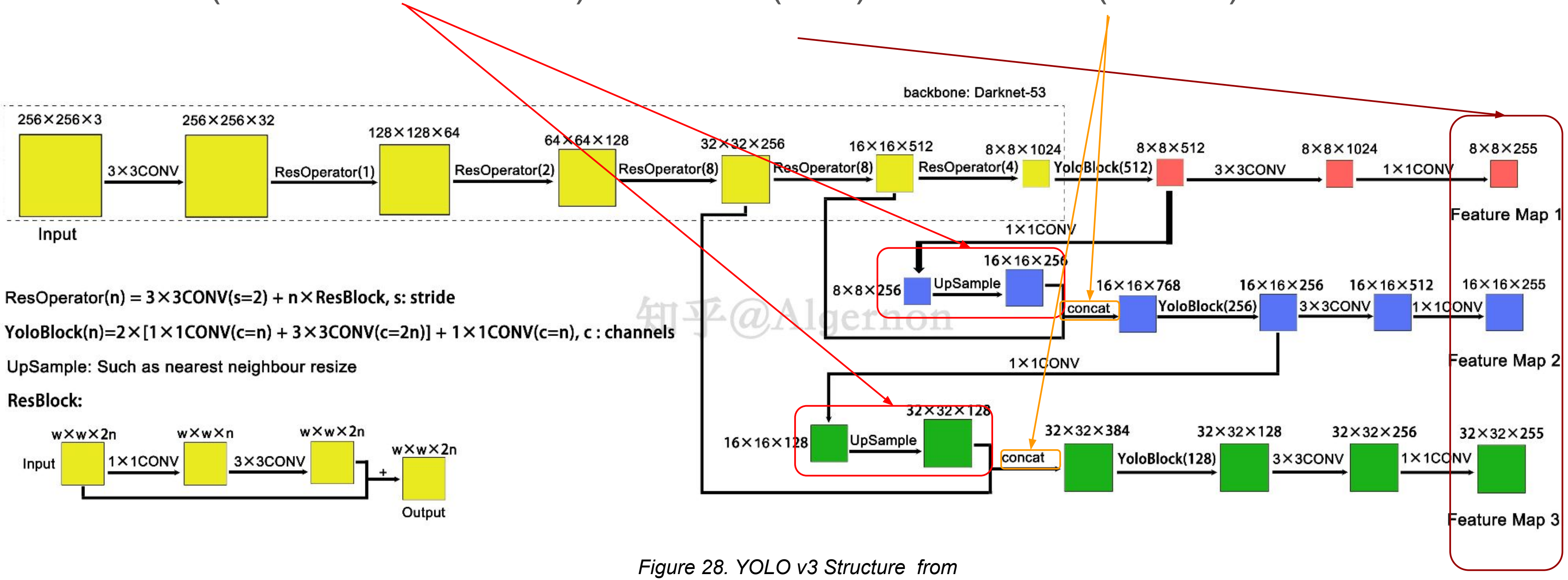


Figure 28. YOLO v3 Structure from https://github.com/thisiszhou/SexyYolo/blob/master/data/yolov3.jpg

YOLO V4

Background Knowledge (1)

- SPP(spatial pyramid pooling layer)
  - Merge multi-scale sense in the model
- PAN(Path Aggregation Network)
  - Predict pare of mask at the same time
  - Using global feature augmentation, enhance feature accuracy

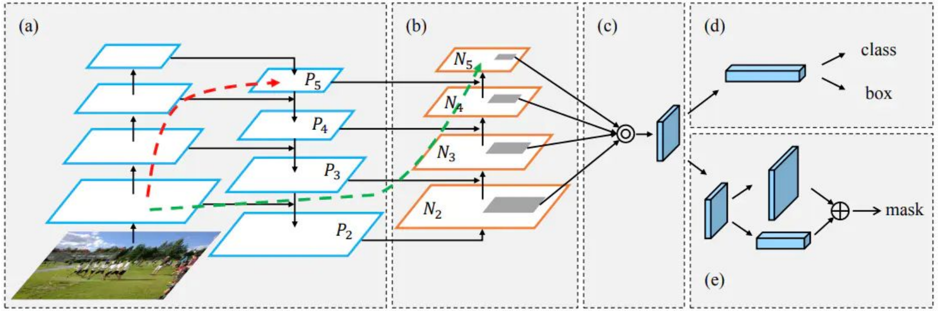


Figure 30. PAN illustration (a) FPN (b) Global feature augmentation (c) Fusion global feature (d) Classification Network for bounding box(e) Mask prediction by fully connected network,
from https://www.jianshu.com/p/34e033961acf

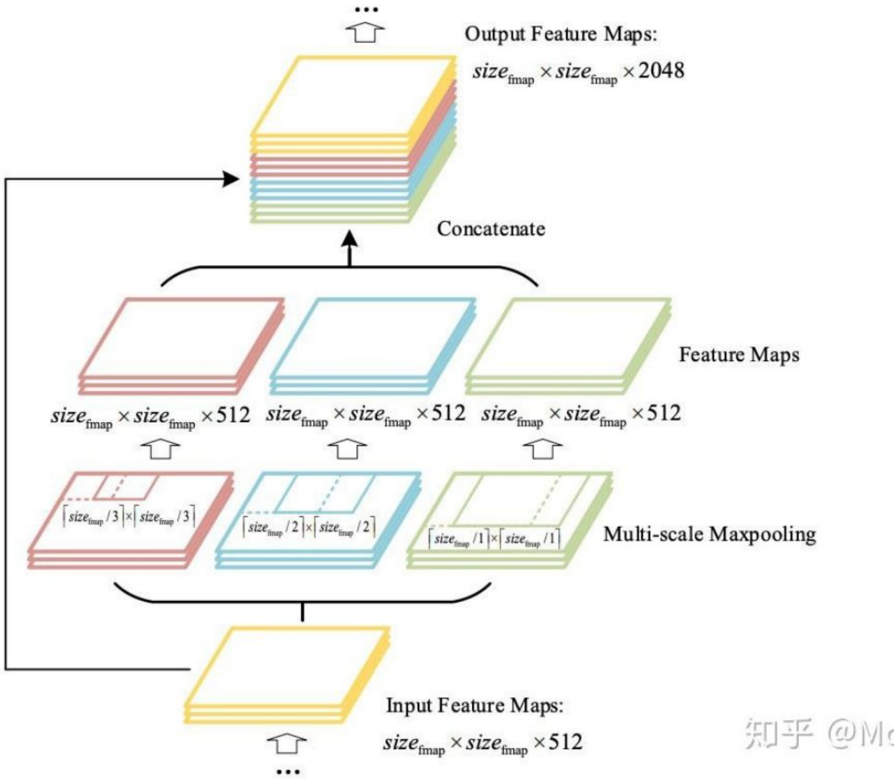


Figure 29. SPP illustration capture from https://zhuanlan.zhihu.com/p/135980432

Background Knowledge (2)

- Mask Network
  - Helping to feature selection region, idea from RCNN
  - Add fully connected layer to detect the location features.
  - ROI = Region of Interest
    - ROI pooling,
    - ROI Align

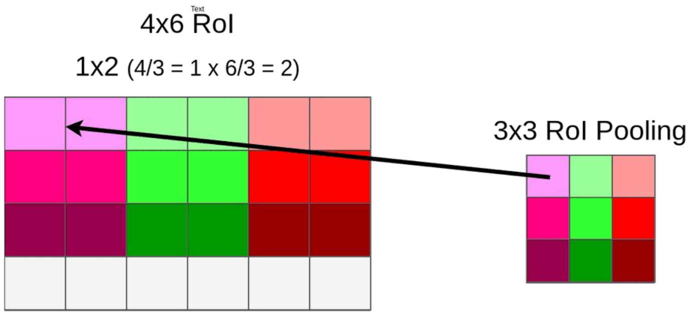


Figure 32. ROI Pooling,
from https://www.jianshu.com/p/34e033961acf

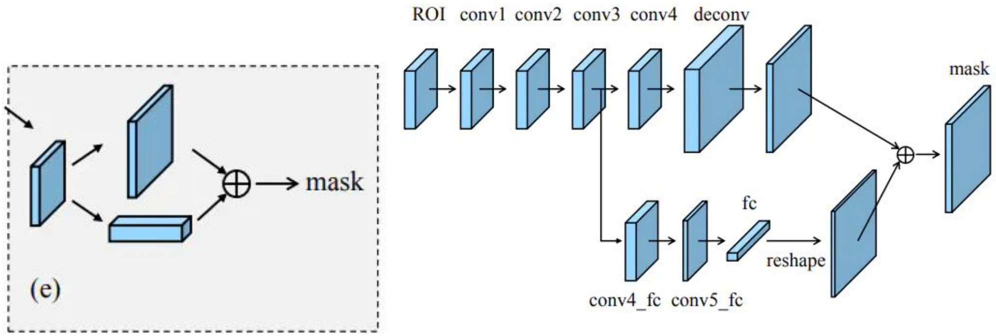


Figure 31. PAN illustration from https://www.jianshu.com/p/34e033961acf

Figure 33.ROI Align,
from https://www.jianshu.com/p/34e033961acf

Background Knowledge (3)

- Mosaic: Merge 4 picture to 1, reduce the number of input

Figure 34. YOLO v4 data augmentation from https://blog.roboflow.com/yolov4-data-augmentation/

YOLO V4

- Overview
  - Structure: DarkNet53 + ResNet (CSPResBlock) + SPP + PANet
  - Bag of specials: SPP, PANet …
  - Bag of freebies: Mosaic, etc…

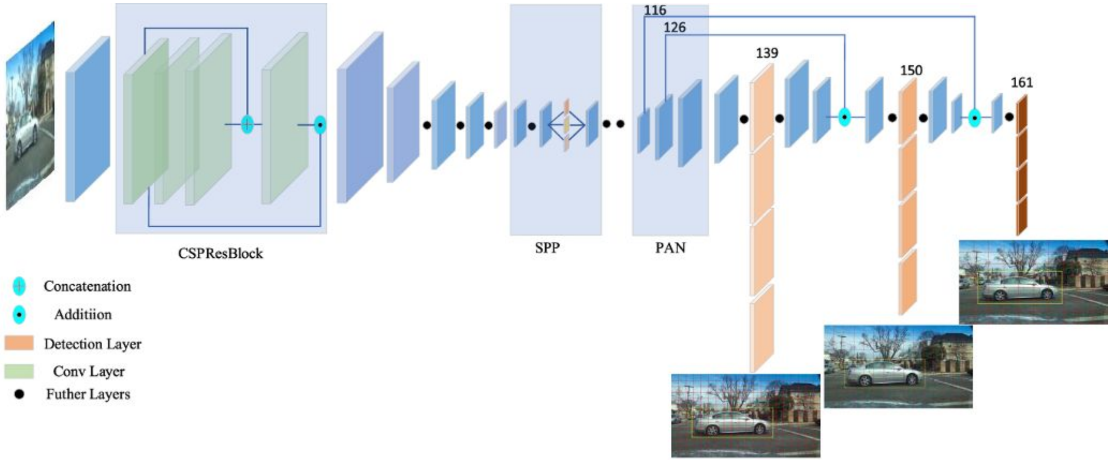


Figure 35. YOLO v4 Network Structure,
From https://sh-tsang.medium.com/review-yolov4-optimal-speed-and-accuracy-of-object-detection-8198e5b37883

State of art based COCO

- YOLO v4 did well on low latency which is why people using it for real-time processing.
- State-of-Art
  - SwingV2, AP = 63.3
  - Dynamic HEAD, AP = 60.6

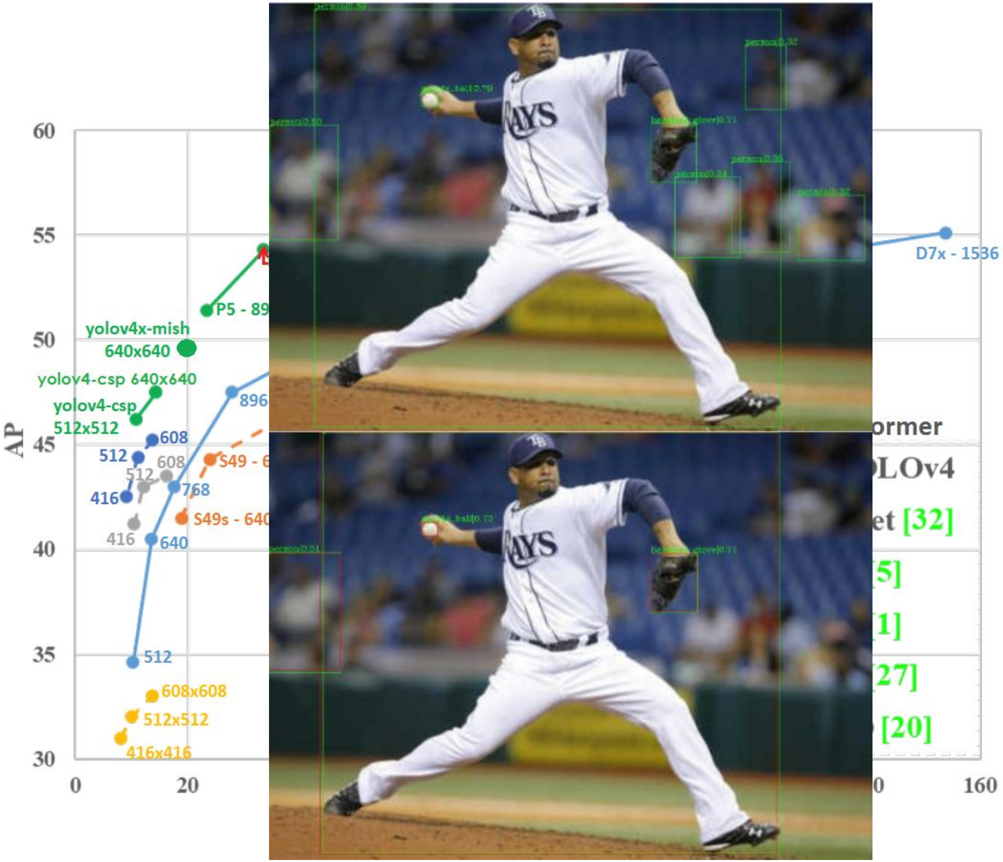


Figure 36: YOLO \& others vs benchmark 2020, Figure 37: Swing V2 vs Schmak

LIVE DEMO

- YOLO v5
  - Idetector APP
  - Using in book detection tasks in librarian robots project

Figure 38. YOLO v5 in App Store,
from https://github.com/ultralytics/yolov5

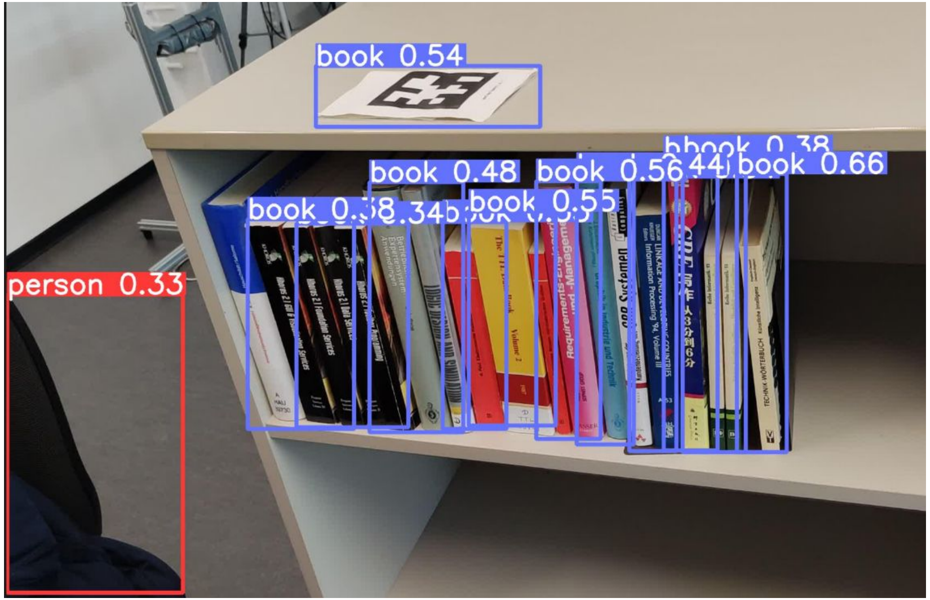


Figure 39. YOLO v5s result for books,

Reference

- DataSets
  - COCO minival https://paperswithcode.com/sota/object-detection-on-coco-minival
  - COCO test-dev https://paperswithcode.com/sota/object-detection-on-coco
  - Pascal VOC https://paperswithcode.com/sota/object-detection-on-pascal-voc-2007
  - CrowdHuman (full body) https://paperswithcode.com/sota/object-detection-on-crowdhuman-full-body
  - KITTI Cars Moderate https://paperswithcode.com/sota/3d-object-detection-on-kitti-cars-moderate
  - 3D Object Detection on KITTI Cars Easy https://paperswithcode.com/sota/3d-object-detection-on-kitti-cars-easy
  - 3D Object Detection on KITTI Cars Hard https://paperswithcode.com/sota/3d-object-detection-on-kitti-cars-hard
  - 3D Object Detection on ScanNetV2 https://paperswithcode.com/sota/3d-object-detection-on-scannetv2
  - 3D Object Detection on SUN-RGBD val https://paperswithcode.com/sota/3d-object-detection-on-sun-rgbd-val

Reference

- Others
  - YOLO https://arxiv.org/abs/1506.02640, https://zhuanlan.zhihu.com/p/32525231
  - YOLO V2 https://arxiv.org/abs/1612.08242, https://zhuanlan.zhihu.com/p/31292482
  - YOLO V3 https://arxiv.org/abs/1804.02767, https://zhuanlan.zhihu.com/p/76802514
  - YOLO V4 https://arxiv.org/abs/2004.10934, https://zhuanlan.zhihu.com/p/135980432
  - YOLO V5 https://github.com/ultralytics/yolov5
  - EfficientDet https://www.zhihu.com/question/357037757
  - Dynamic Head https://arxiv.org/pdf/2106.08322v1.pdf#page=10&zoom=100,412,316
  - Swing V2 https://arxiv.org/pdf/2106.08322v1.pdf#page=10&zoom=100,412,316
  - YOLO Comparison https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109
  - Hog https://zhuanlan.zhihu.com/p/85829145
  - Sobel https://zhuanlan.zhihu.com/p/67197912
  - DPM https://blog.csdn.net/ttransposition/article/details/41806601, https://www.zhihu.com/question/48445722
  - Bach normalization (BN) https://zhuanlan.zhihu.com/p/34879333
  - FLOPS https://en.wikipedia.org/wiki/FLOPS#:~:text=In%20computing%2C%20floating%20point%20operations,than%20measuring%20instructions%20per%20second.
  - DarkNet https://github.com/pjreddie/darknet
  - darknet https://github.com/pjreddie/darknet
  - anchor box https://zhuanlan.zhihu.com/p/31292482

Thank you for your attention