General Notes

Region Proposal based methods (RCNN varients) generally have higher recall than YOLO methods which are more focused on frame rate. Speed comparisons between models are suspect given hardware and feature extractor considerations. Google has a good recent comparison of modern object detectors excluding YOLO varieties.

It appears the greatest challenge lies in region proposal as very good classifiers already exist, and are somewhat interchangeable at this level. Segmentation is now increasingly linked to object detection as it is a “simple” addition to classify a single image within a known box.

State of the art detection:

Faster-RCNN, SSD, YOLOv2, R-FCN

State of the art segmentation:

Mask RCNN, FCIS, TA-FCN

SSD

<https://arxiv.org/abs/1512.02325>

RCNN

<https://arxiv.org/abs/1311.2524>

Fast RCNN

<https://arxiv.org/abs/1504.08083>

Faster RCNN

<https://arxiv.org/abs/1506.01497>

Mask RCNN

<https://arxiv.org/abs/1703.06870>

Mask RCNN is built on top of Faster RCNN and includes a concurrent object pixel mask segmentation system. Current state of the art for instance segmentation. Contributes (1) a spatially aware feature map for each RoI using bilinear interpolation instead of quantization (RoIAlign), (2) a semantic segmentation addon module to an existing object detection system.

Say 20% longer than Faster RCNN but cheats by using only top 100 detection boxes instead of 300. Compare directly to Faster RCNN?

YOLO

<https://arxiv.org/abs/1506.02640v5>

<https://pjreddie.com/darknet/yolo/>

More Localization errors than Fast RCNN, uses fixed grid to distribute objects across image. Uses GoogleNet base. Fixed grid is a huge limitation fixed in v2, cant predict small objects, but helps prevent duplicate detections. YOLO claims a simpler pipeline than RCNN as it is a single net. Finds combining with Fast-RCNN to be beneficial to performance.

YOLO 9000/ YOLOv2

<https://arxiv.org/abs/1612.08242v1>

Outperforms Faster RCNN with a higher frame rate on VOC2012test

FCIS (Microsoft)

<https://arxiv.org/abs/1611.07709>

<https://github.com/msracver/FCIS>

Won 2016 COCO segmentation Mask-RCNN outperforms

R-FCN

<https://arxiv.org/abs/1605.06409>

TA-FCN 2016 COCO instance segmentation challenge, based on R-FCN

Speed/Accuracy trade-offs for modern convolutional object detectors (Google)

<https://arxiv.org/abs/1611.10012>

Best comparison of modern object detectors speed/accuracy as well as feature extractors.

Doesn’t compare to YOLO

Chronological Order

RCNN, Fast RCNN, Faster RCNN, YOLO, SSD, FCIS, YOLOv2, Mask RCNN

Datasets

COCO – Common Objects in Context

<http://mscoco.org>

Microsoft Sponsored

Multiple Objects per image

80 object categories

5 captions per image

Object segmentation ground truth

Cittyscapes Dataset – Semantic Understanding of Urban Street Scenes

<https://www.cityscapes-dataset.com/dataset-overview/>

Dense Pixel level annotations 30 classes

Pascal VOC (finished as of 2012) – Visual Object Classes

<http://host.robots.ox.ac.uk/pascal/VOC/>

2012 – 20 classes 11530 images, 27450 ROI 6929 segmentations

ILSVRC – Imagenet Large Scale Visual recognition Challenge

200 Categories

Includes object detection from video

Other

<https://github.com/tensorflow/models/tree/master/object_detection>

Google just released the Tensorflow Object Detection AP (June 2017)

Includes a few trainable detection models including SSD with Moblie Net. This would make a good benchmark.

MobileNet

<https://arxiv.org/abs/1704.04861>

SqueezeNet

<https://arxiv.org/abs/1602.07360>