Object Detection

Abdallah Youssef 18015026 | Ahmed Bahgat 18010078 | Mohamed Metwalli 18011587

December 2022

1 Data Sets

1.1 COCO

COCO is an object detection data set with 90 categories and over 200 thousand labeled images.
We used the validation set that contains 5000 images to get the inference results.
The evaluation consists of 12 metrics found in the documentation.

1.2 Pascal VOC

Another object detection data set with 20 categories. We used 2510 images. We re-used the COCO evaluation metrics on the images inferred.

2 Network Models

We used the pretrained models from the TensorFlow 2 Detection Model Zoo.
We chose the three following models: ResNet, MobileNet, and R-CNN.

	ResNet-152	MobileNet	R-CNN
Number of stages	Single	Single	Multi
Number of layers	152	13	101
Precision	0.524	0.481	0.461
Recall	0.488	0.440	0.441
Unique Architecture	Skip connections	Depth-wise separable convolution	Region Proposals

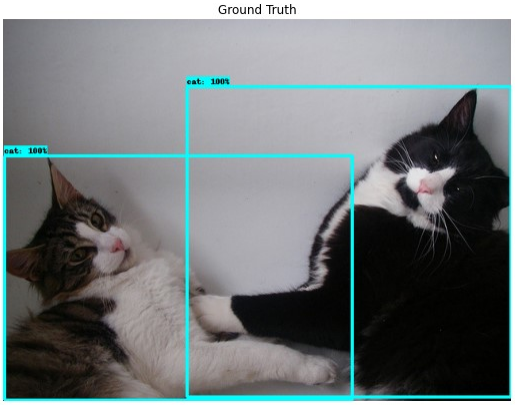
2.1 ResNet

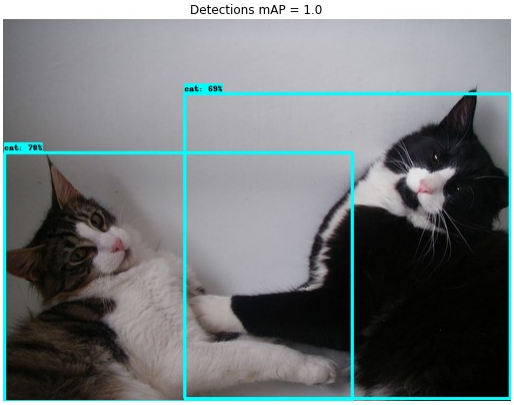
ResNet uses skip connections to create very deep networks overcoming the problem of vanishing gradients.
We used ResNet152 which consists of 152 layers.
It is a single shot detector (SSD).

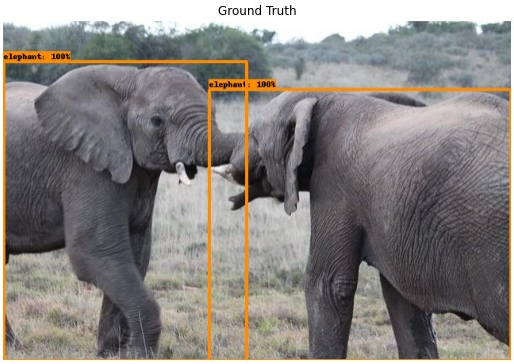
2.1.1 COCO Results

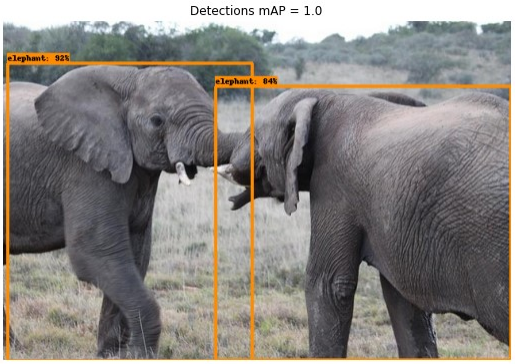
Average Precision (AP) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.350
Average Precision (AP) @[IoU=0.50 — area= all — maxDets=100] = 0.524
Average Precision (AP) @[IoU=0.75 — area= all — maxDets=100] = 0.381
Average Precision (AP) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.350
Average Precision (AP) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 1] = 0.307
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 10] = 0.488
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.524
Average Recall (AR) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.524
Average Recall (AR) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A

2.1.2 Good Examples



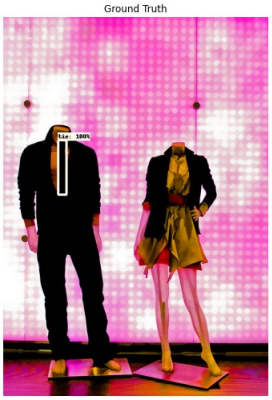


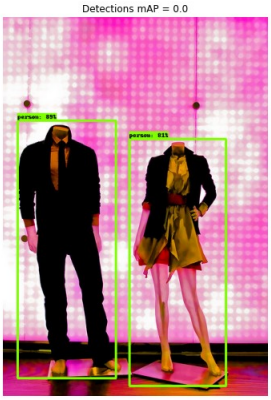




2.1.3 Bad Examples

Occlusion:

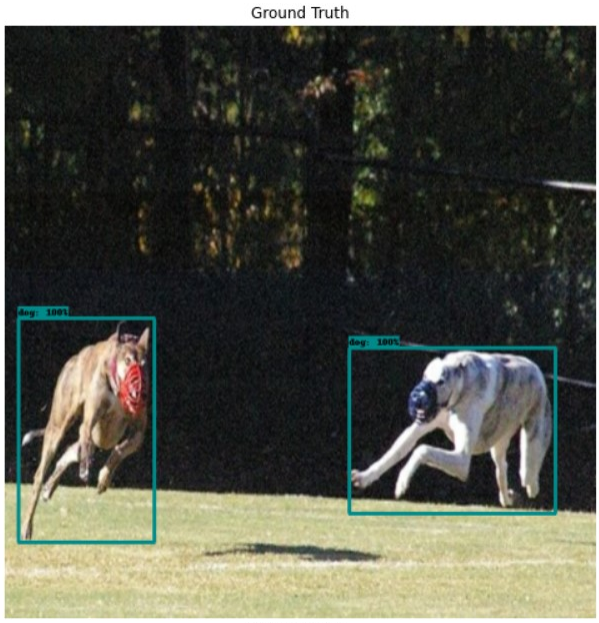


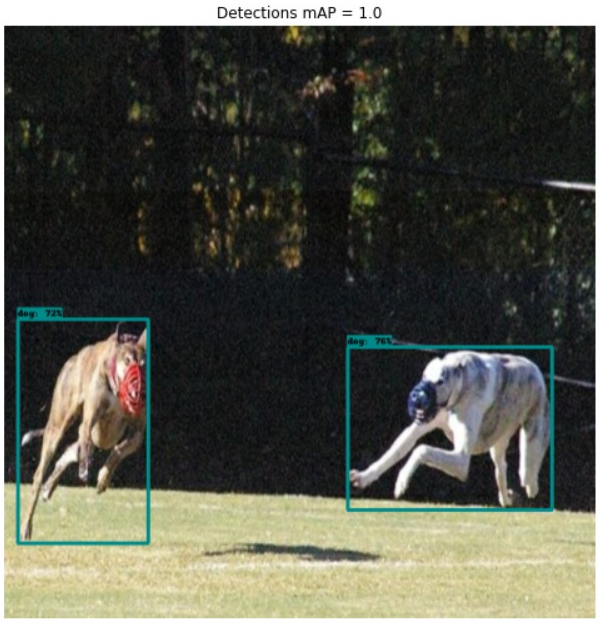


2.1.4 Pascal VOC Results

Average Precision (AP) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.577
Average Precision (AP) @[IoU=0.50 — area= all — maxDets=100] = 0.819
Average Precision (AP) @[IoU=0.75 — area= all — maxDets=100] = 0.636
Average Precision (AP) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.577
Average Precision (AP) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 1] = 0.429
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 10] = 0.669
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.711
Average Recall (AR) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.711
Average Recall (AR) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A

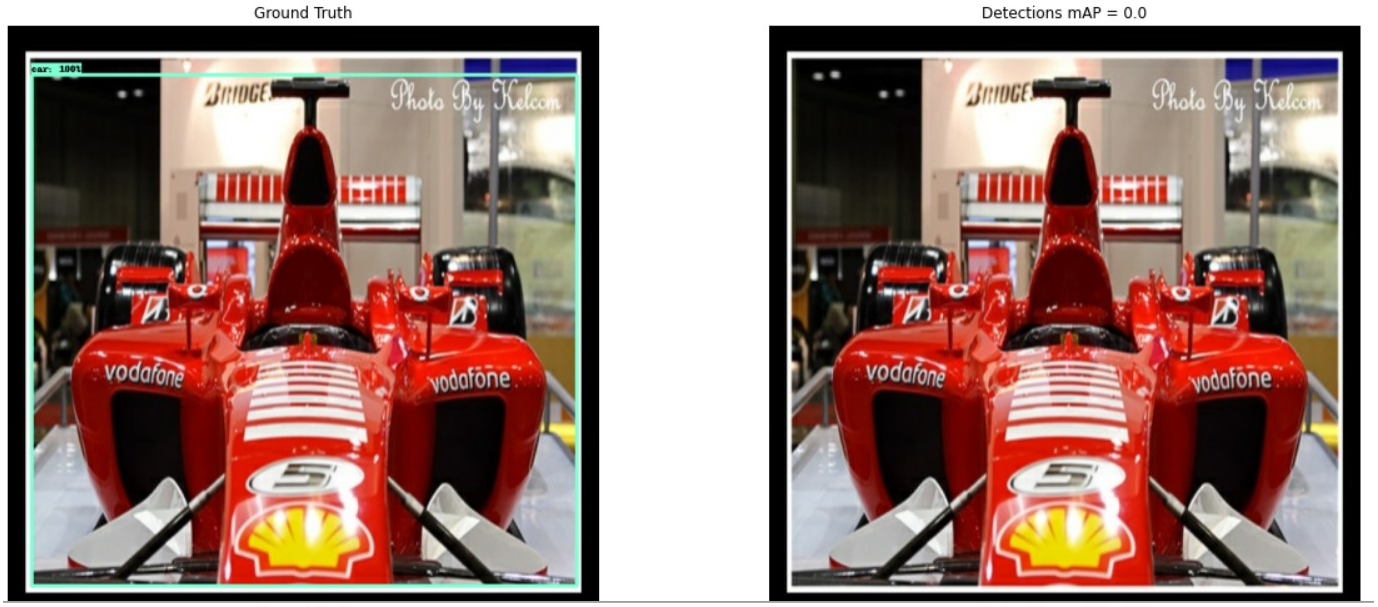
2.1.5 Good Examples





2.1.6 Bad Examples

Occlusion:



2.2 Faster R-CNN

The Faster R-CNN is an improvement on the Fast R-CNN and the R-CNN. It uses the ResNet feature extractor

The R-CNN extracts region proposals from the image instead of trying every sliding window and inputs the warped region proposals into a CNN which outputs a classification.

The Fast R-CNN improves efficiency by eliminating recalculating the features of the overlapping regions. It does so by inputting the entire image into the CNN and then cropping the feature map corresponding to the desired region proposal + ROI pooling.

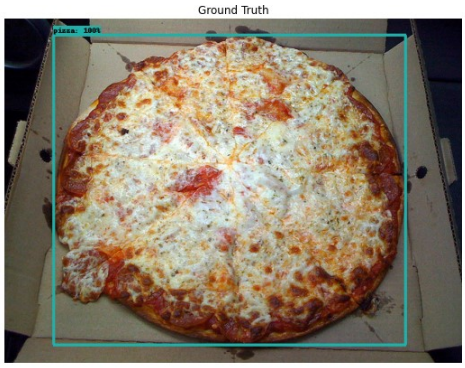
The Faster R-CNN further improves the speed by doing the region proposal extraction as part of the network architecture.

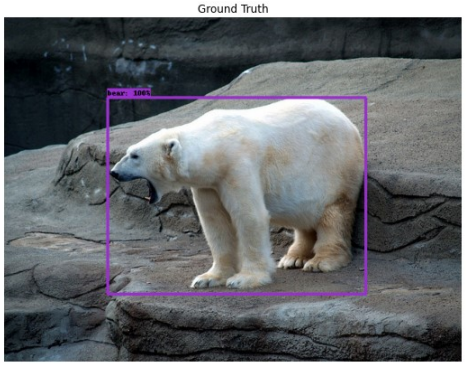
```
feature_extractor {
  type: "faster_rcnn_resnet101_keras"
  batch_norm_trainable: true
}
```

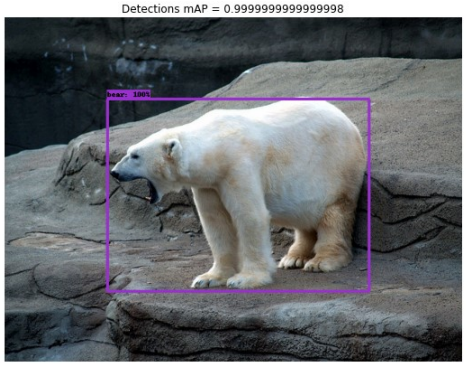
2.2.1 COCO Results

Average Precision (AP) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.304
Average Precision (AP) @[IoU=0.50 — area= all — maxDets=100] = 0.481
Average Precision (AP) @[IoU=0.75 — area= all — maxDets=100] = 0.321
Average Precision (AP) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.304
Average Precision (AP) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 1] = 0.282
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 10] = 0.440
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.469
Average Recall (AR) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.469
Average Recall (AR) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A

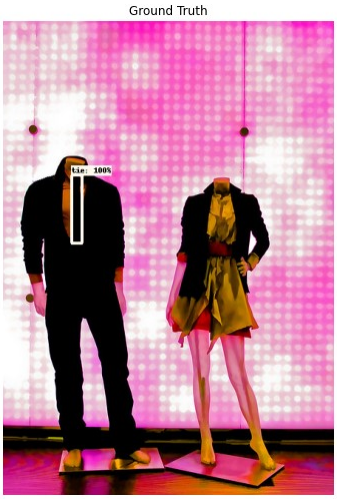
2.2.2 Good Examples

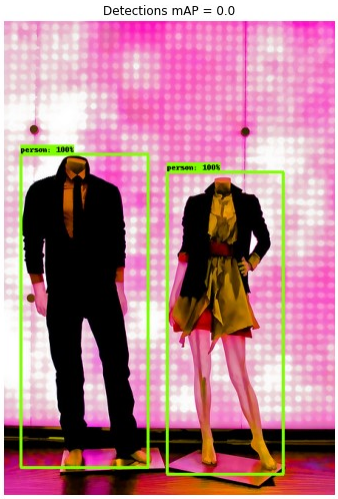


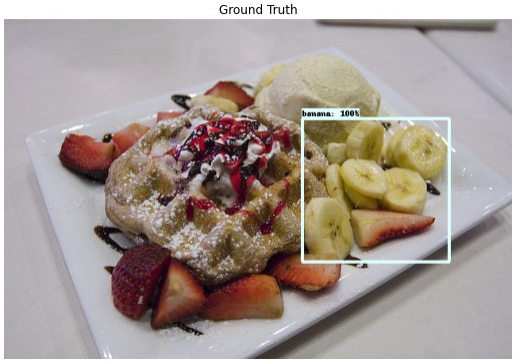


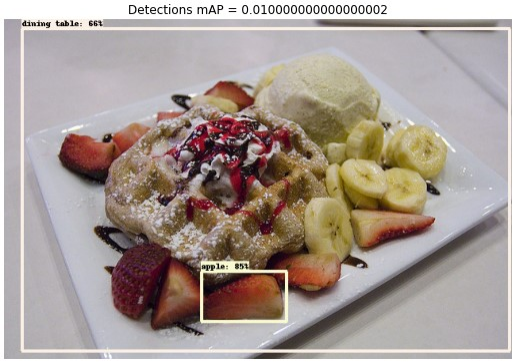


2.2.3 Bad Examples







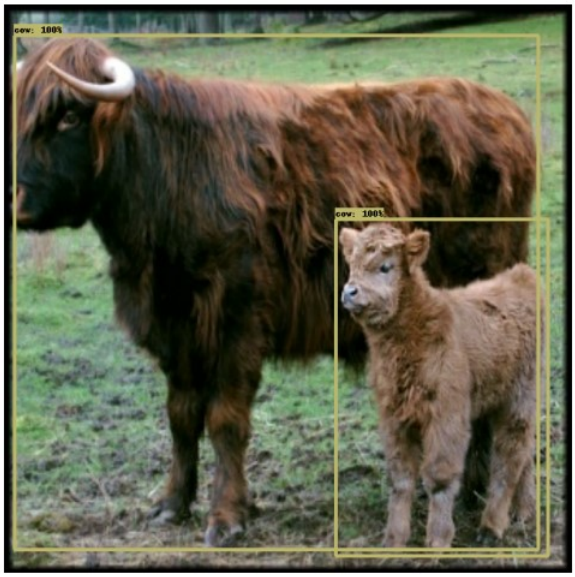


2.2.4 Pascal VOC Results

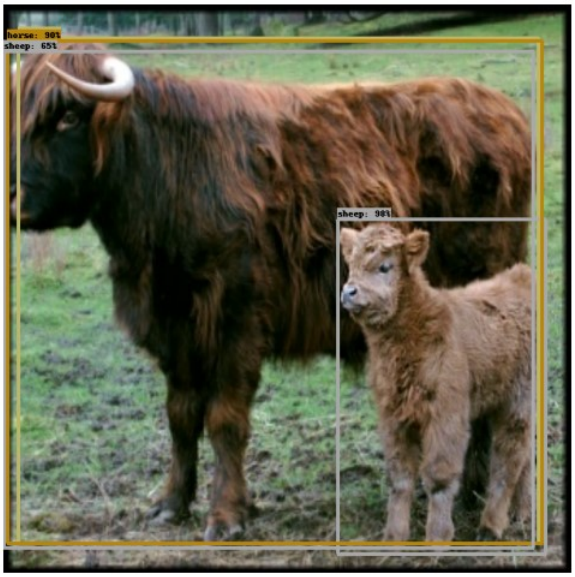
Average Precision (AP) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.530
Average Precision (AP) @[IoU=0.50 — area= all — maxDets=100] = 0.779
Average Precision (AP) @[IoU=0.75 — area= all — maxDets=100] = 0.584
Average Precision (AP) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.530
Average Precision (AP) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 1] = 0.407
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 10] = 0.632
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.663
Average Recall (AR) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.663
Average Recall (AR) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A

2.2.5 Good Examples

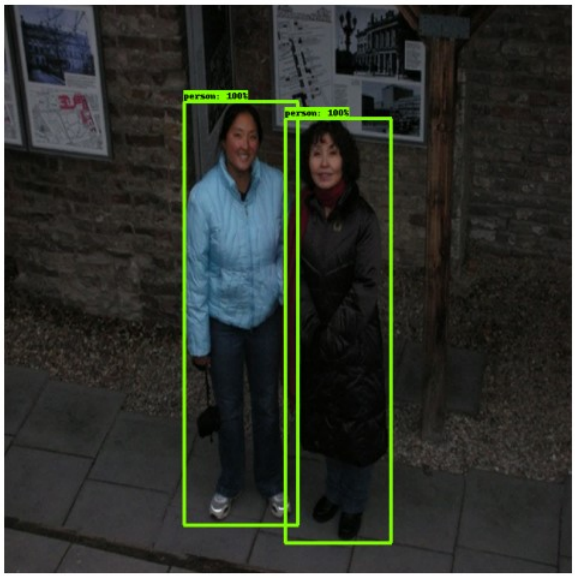
Ground Truth



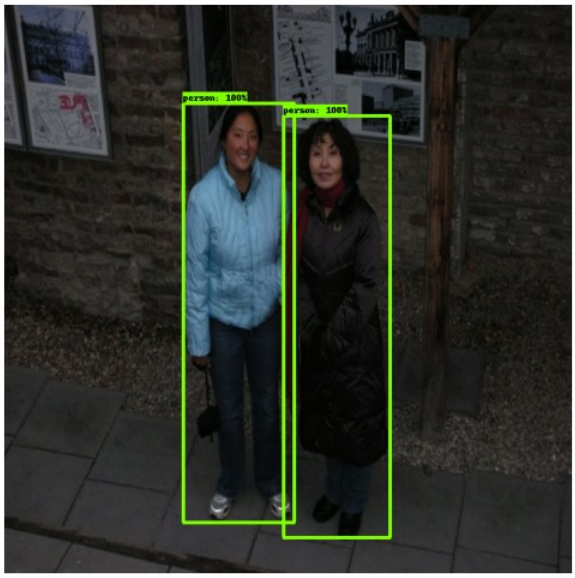
Detections mAP = 1.0



Ground Truth



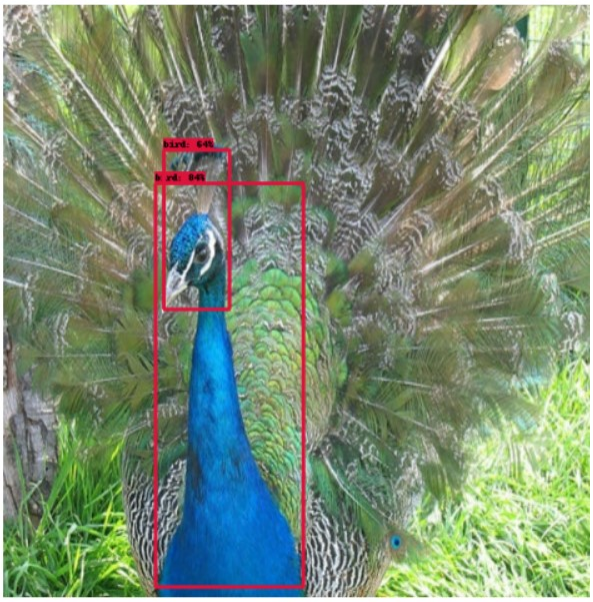
Detections mAP = 1.0



2.2.6 Bad Examples

Ground Truth

Detections mAP = 0.0



Ground Truth

Detections mAP = 0.0

2.3 MobileNet

MobileNet is a lightweight network used for low-compute environments such mobile and embedded vision applications. It uses 13 layers of depth-wise and point convolutions instead of regular convolutions which are much more expensive. It is a single-shot detector (SSD).

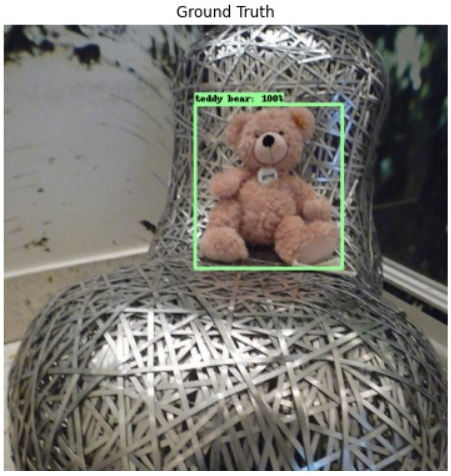
2.3.1 COCO Results

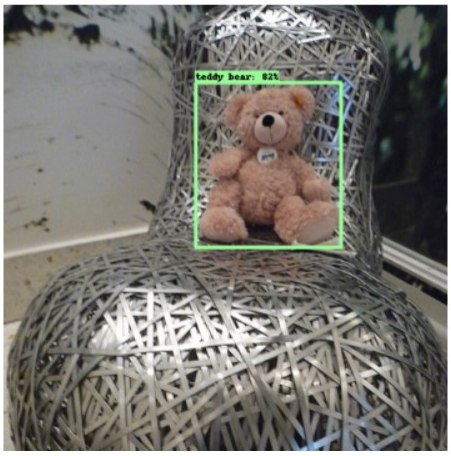
Average Precision (AP) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.290
Average Precision (AP) @[IoU=0.50 — area= all — maxDets=100] = 0.461
Average Precision (AP) @[IoU=0.75 — area= all — maxDets=100] = 0.310
Average Precision (AP) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.290
Average Precision (AP) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 1] = 0.274
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 10] = 0.441
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.476
Average Recall (AR) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.476
Average Recall (AR) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A

2.3.2 Pascal VOC Results

Average Precision (AP) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.490
Average Precision (AP) @[IoU=0.50 — area= all — maxDets=100] = 0.755
Average Precision (AP) @[IoU=0.75 — area= all — maxDets=100] = 0.532
Average Precision (AP) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.490
Average Precision (AP) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 1] = 0.386
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets= 10] = 0.612
Average Recall (AR) @[IoU=0.50:0.95 — area= all — maxDets=100] = 0.655
Average Recall (AR) @[IoU=0.50:0.95 — area= small — maxDets=100] = 0.655
Average Recall (AR) @[IoU=0.50:0.95 — area=medium — maxDets=100] = N/A
Average Recall (AR) @[IoU=0.50:0.95 — area= large — maxDets=100] = N/A

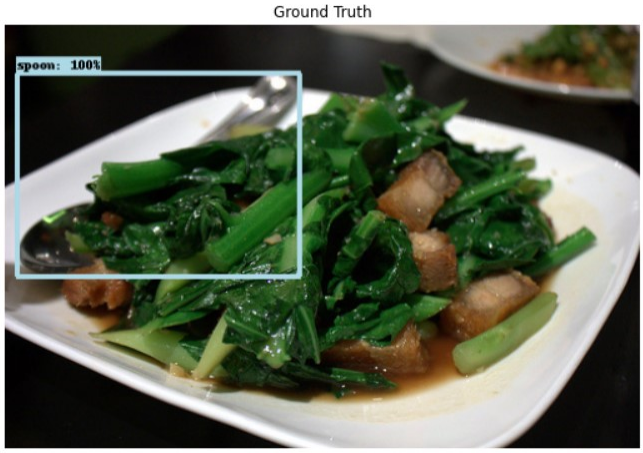
2.3.3 Good Examples



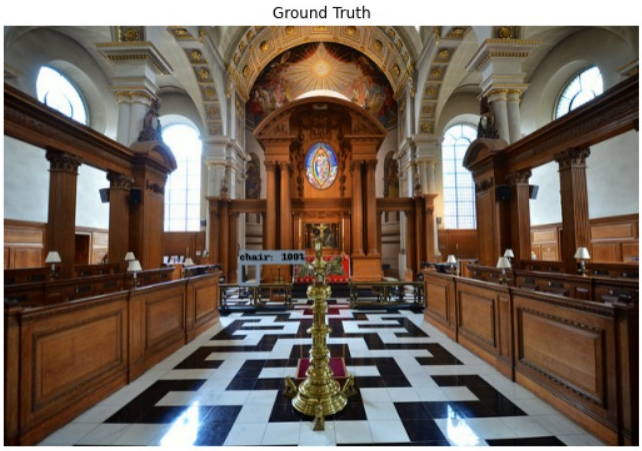


2.3.4 Bad Examples

Occlusion:



Small and far:



3 Comparisons

	ResNet	Faster R-CNN	MobileNet
Number of stages	single	multi	single
Speed(ms)	111	55	48
Image Size	640*640	640*640	640*640
Suitable Use Cases	the most prominent objects	the most prominent objects	mobile vision apps
Unsuitable Use Cases	standalone use in mission-critical applications	autonomous driving	small objects
Supporting Batching	doesn't support	doesn't support	supports