Object Detection

Abdallah Youssef 18015026
Ahmed Bahgat 18010078
Mohamed Metwalli 18011587

December 2022

1 Datasets

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 metric that has been used is mean average precision (mAP) which is the average of all the classes average precisions. The mAP is calculated over multiple IoU thresholds to test the robustness of the model. More information can be found in this documentation.

1.2 Pascal-VOC 2007

Pascal-VOC is another object detection data set with 20 categories. We used 2510 images. We re-used the COCO evaluation metrics on the images inferred only in the bounding boxes where there is an overlap between the classes of the two.

2 Network Models

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

	SSD-ResNet-152	SSD-MobileNet	Faster 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

3 Single-Shot Detector - ResNet

Single-Shot Detectors (SSD) is a family of object detection neural networks which only need one forward pass through the network without the need to have a separate stage for generating detection regions such as sliding-window techniques or region proposal networks lie the R-CNN family. The SSD network usually consists of two parts: the first part is the backbone which is a pre-trained convolutional neural network (CNN) for object classification tasks, the second part is the SSD head which takes the output feature map from the convolutional layers and use them to generate bounding-box positions and class labels.

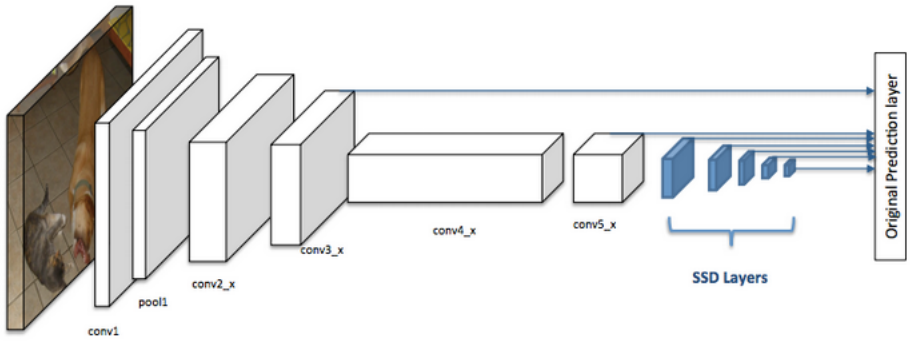


Figure 1: SSD network architecture

3.1 Backbone

The backbone of choice for this network is the ResNet-152 network pretrained on MS-COCO dataset. ResNet-152 is a very deep neural network and is one of the most successful neural networks in the world of object classification. The key contribution of this paper is introducing residual blocks which consists of a number of convolutional layers which do not change size and a skip connection from the input of the block which gets added to the output of the convolutional layer. This helps train very deep neural network by allowing layers that finished training early to converge to the identity transformation.

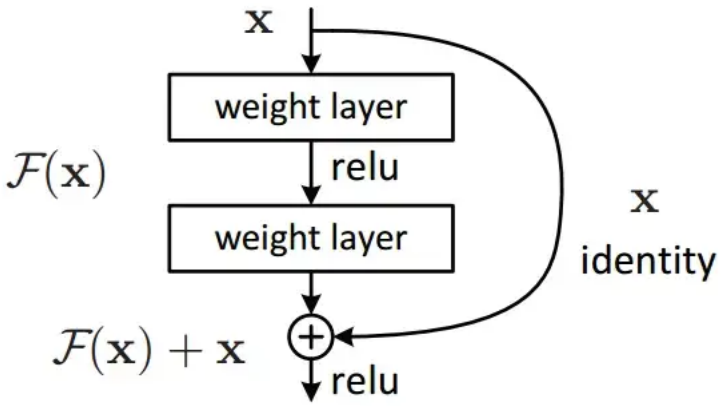


Figure 2: Caption

3.2 SSD Heads

To handle the problem of multi-scale object detection, SSD utilizes the fact that the convolutional layers downsamples the image as you go deeper through the network which acts like an image pyramid. The SSD head is a neural network which takes as an input the feature maps from some of the layers of the backbone and they are used to calculate the output.

3.3 Anchor Boxes

To handle multiple poses of objects, the SSD head uses 4 filters with 4 different aspect ratios. The head calculates 5 values for each anchor box: confidence value, (c_x, c_y) position of the center, and (h, w) for the box.

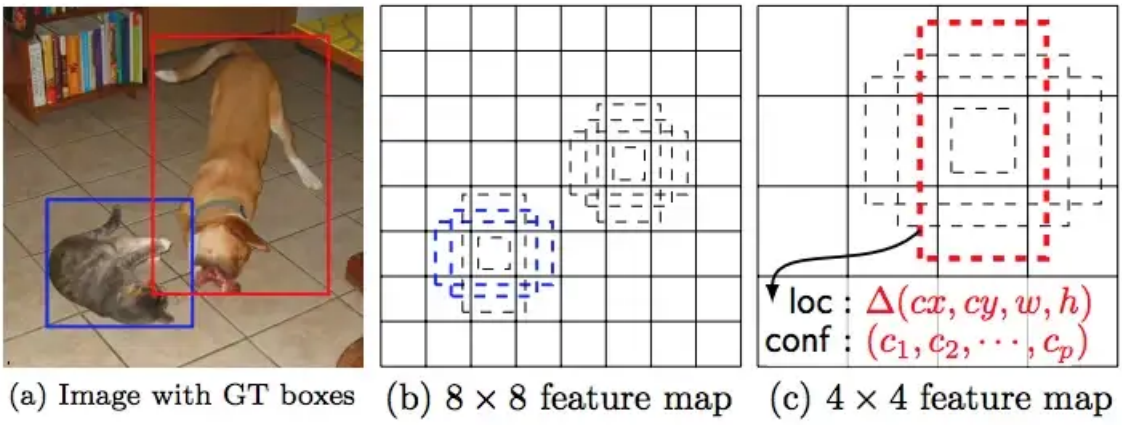


Figure 3: Anchor boxes at different aspect ratios

3.4 Loss Function

Since the object detection task can be thought of as two tasks which are classification and localization, the loss function consists of two parts: L_{conf} which calculates the confidence of our bounding boxes predictions, and L_{loc} which acts as a regressor for the bounding pox positions and dimensions.

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

3.5 Output

3.5.1 Good Examples

Figure 4: COCO good example 1

Figure 5: COCO good example 2

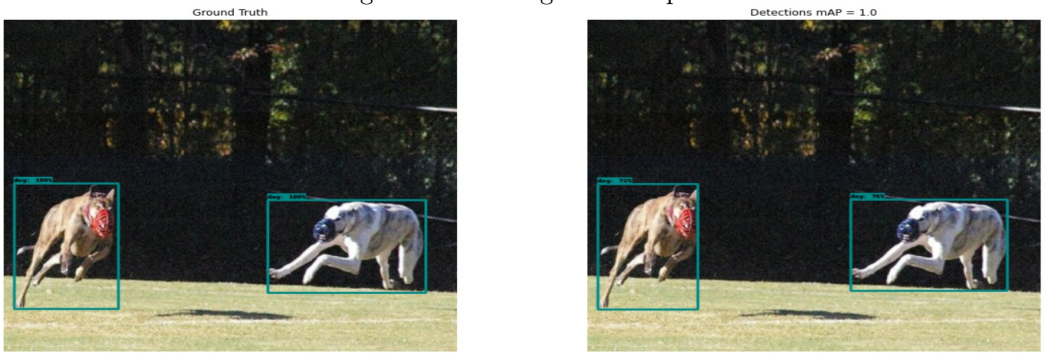


Figure 6: Pascal good example 1

Figure 7: Pascal good example 2

3.5.2 Bad Examples

Figure 8: COCO bad example 1

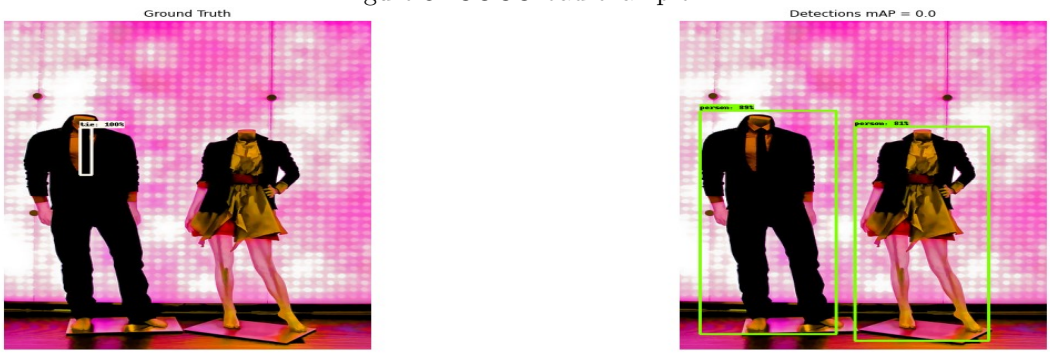


Figure 9: COCO bad example 2

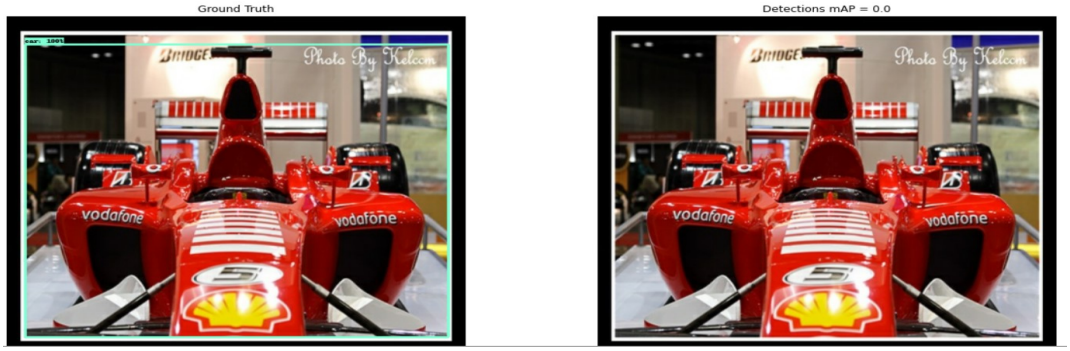


Figure 10: Pascal bad example 1

Figure 11: Pascal bad example 2

4 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.

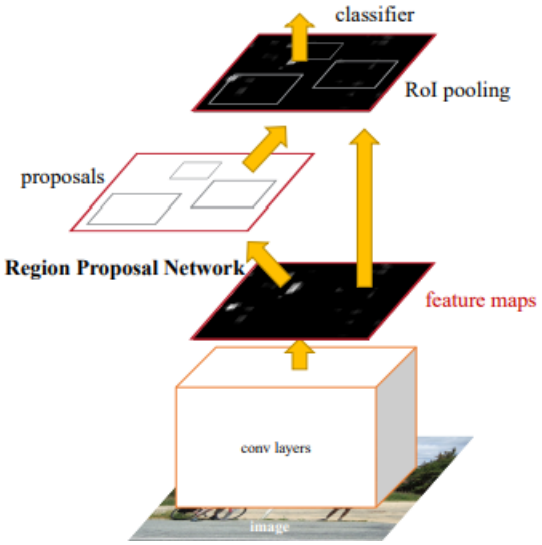


Figure 12: Faster R-CNN architecture overview

4.1 Region Proposal Network

In R-CNN and Fast R-CNN, there was a region proposal stage which was implemented by selective search. The problem was that this step took a lot of time (up to 2 seconds per image) and it is not easily parallelizable. The contribution of Faster R-CNN is to add this selective search to the detection pipeline by introducing region proposal network (RPN). RPN takes as an input the output of the feature map of the backbone and slides a window of size $(n \times n)$ over the image. Each window is mapped to a lower dimensional embedding encoding k region proposals each consists of 5 numbers: (x, y, w, h) bounding box information, and p objectness score. The regions then are fed into the network for doing object classification and bounding box regression.

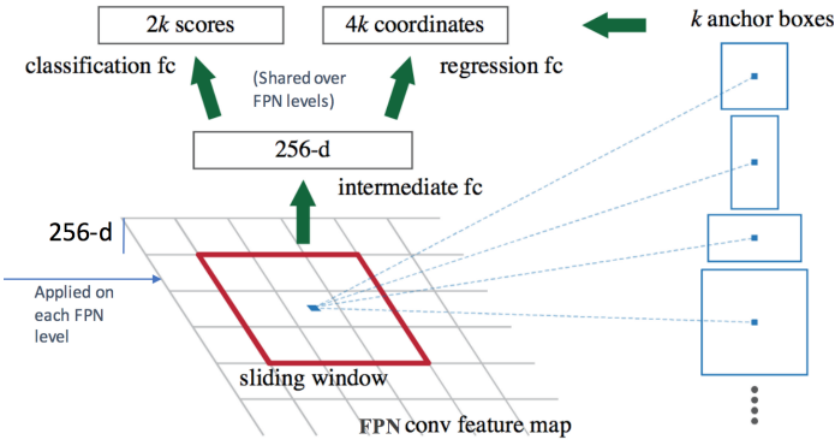


Figure 13: Region proposal network

4.2 Output

4.2.1 Good Examples

Figure 14: COCO good example 1

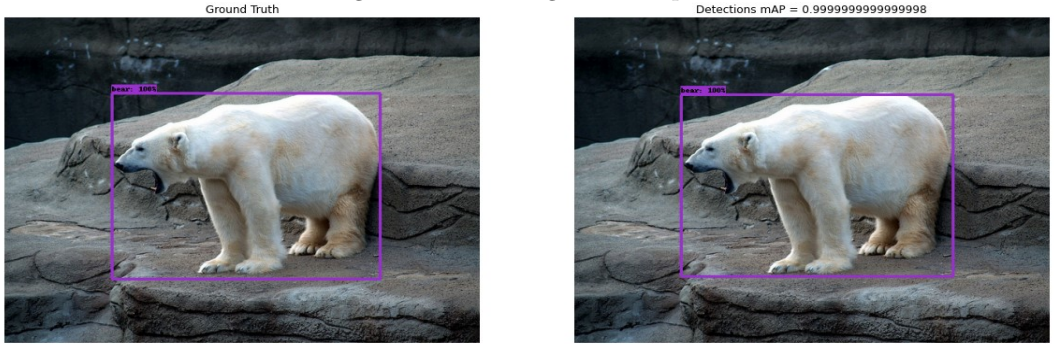


Figure 15: COCO good example 2

Figure 16: Pascal good example 1

Figure 17: Pascal good example 2

4.2.2 Bad Examples

Figure 18: COCO bad example 1

Figure 19: COCO bad example 2

Figure 20: Pascal bad example 1

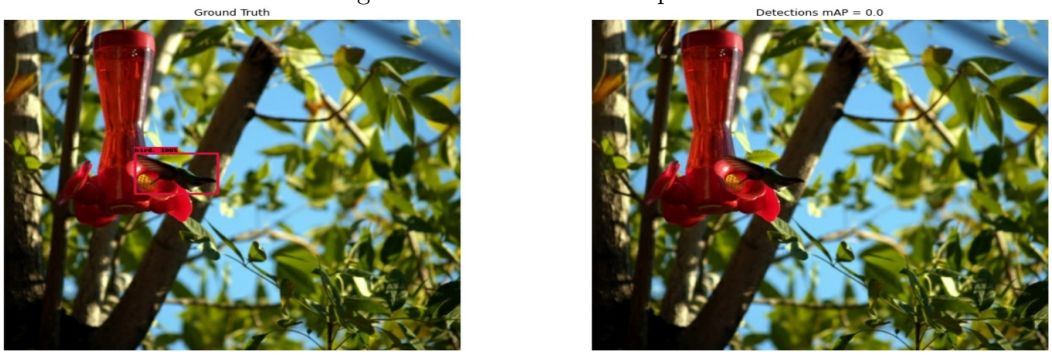


Figure 21: Pascal bad example 2

4.3 SSD-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).

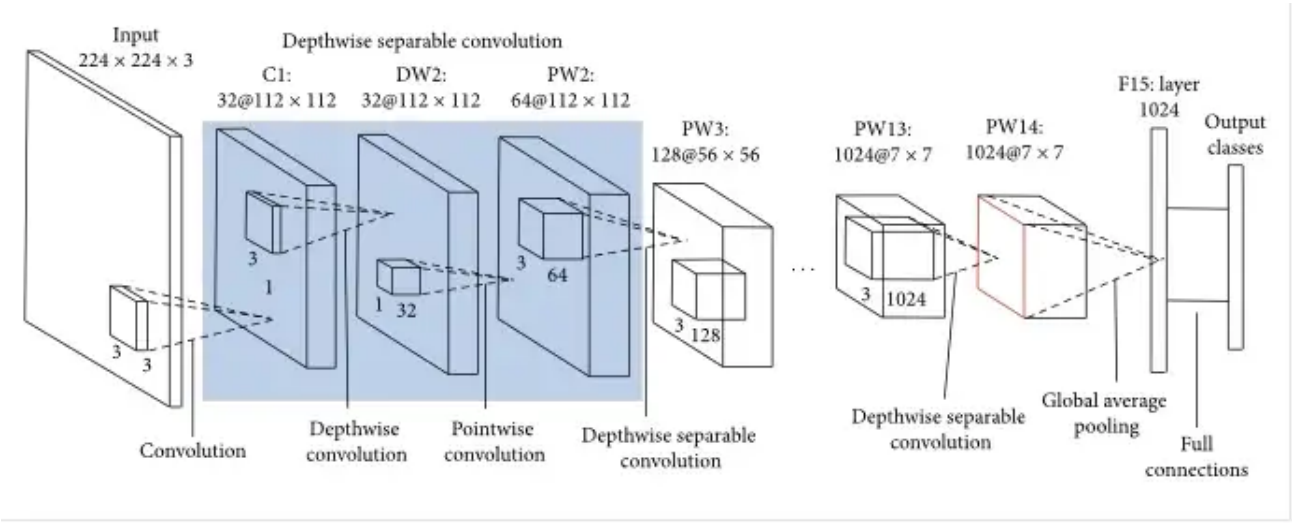


Figure 22: MobileNet architecture

4.4 Depthwise Separable Convolution

The main contribution of this paper is that instead of using normal convolutions which result in a complexity of $\mathcal{O}(n_x \cdot n_y \cdot f_x \cdot f_y)$, separable convolutions are used which reduce drastically the number of parameters for a single filter from $\mathcal{O}(f_x \cdot f_y)$ to $\mathcal{O}(f_x + f_y)$. Also, the time complexity for a single convolution is reduced to $\mathcal{O}((n_x \cdot n_y) \cdot (f_x + f_y))$.

After doing the convolution step, another (1×1) point-wise convolution step is carried out which further reduces the dimensionality of the output feature map of this layer.

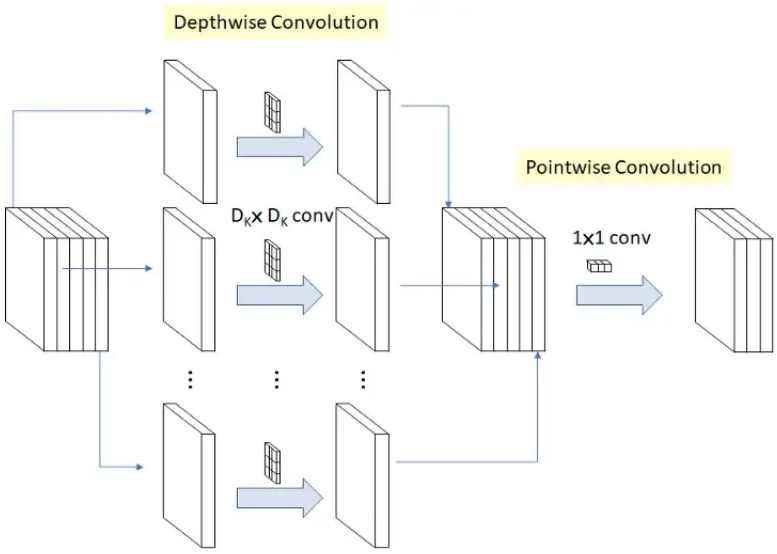


Figure 23: Depthwise separable convolution

4.5 Output

4.5.1 Good Examples

Figure 24: COCO good example 1

Figure 25: COCO good example 2

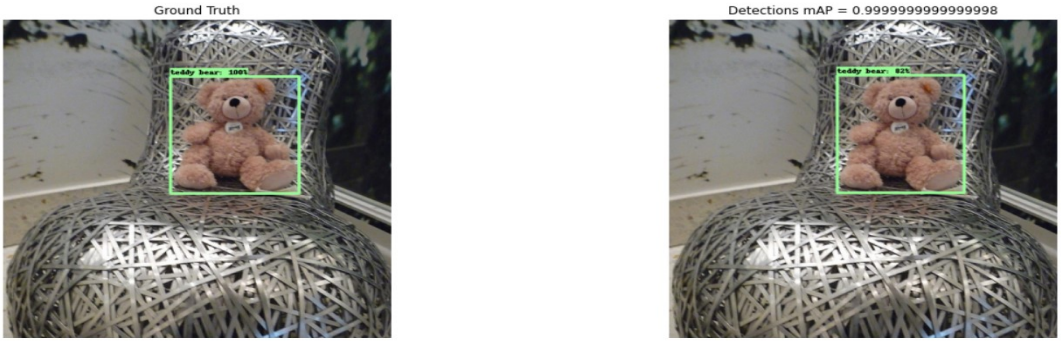


Figure 26: Pascal good example 1

4.5.2 Bad Examples

Figure 27: COCO bad example 1: occlusion

Figure 28: COCO bad example 2: small & far

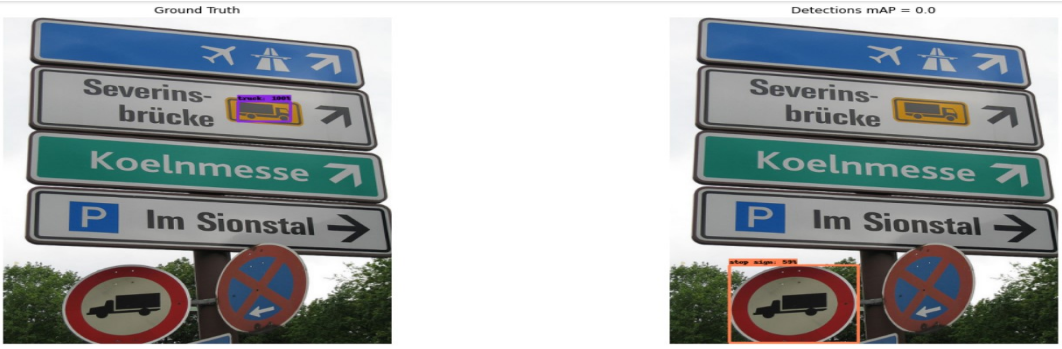


Figure 29: Pascal bad example 1: small & far

5 Comparisons

5.1 General comparison

	SSD-ResNet	Faster R-CNN	SSD-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

5.2 Performance on COCO validation set

	SSD-Resnet	**MobileNet**	**Faster R-CNN**
(AP) @[IoU=0.50:0.95]	**35%**	29%	30.4%
(AP) @[IoU=0.75]	**38.1%**	31%	32.1%
(AP) @[IoU=0.50]	**52.4%**	46.1%	48.1%
mAP	**35%**	29%	30.4%

5.3 Performanve on Pascal-VOC 2007 validation set

	SSD-Resnet	**MobileNet**	**Faster R-CNN**
(AP) @[IoU=0.50:0.95]	**57.7%**	49%	53%
(AP) @[IoU=0.75]	**63.6%**	53.2%	58.4%
(AP) @[IoU=0.50]	**81.9%**	75.5%	77.9%
mAP	**57.7%**	49%	53%

6 Remarks on performance

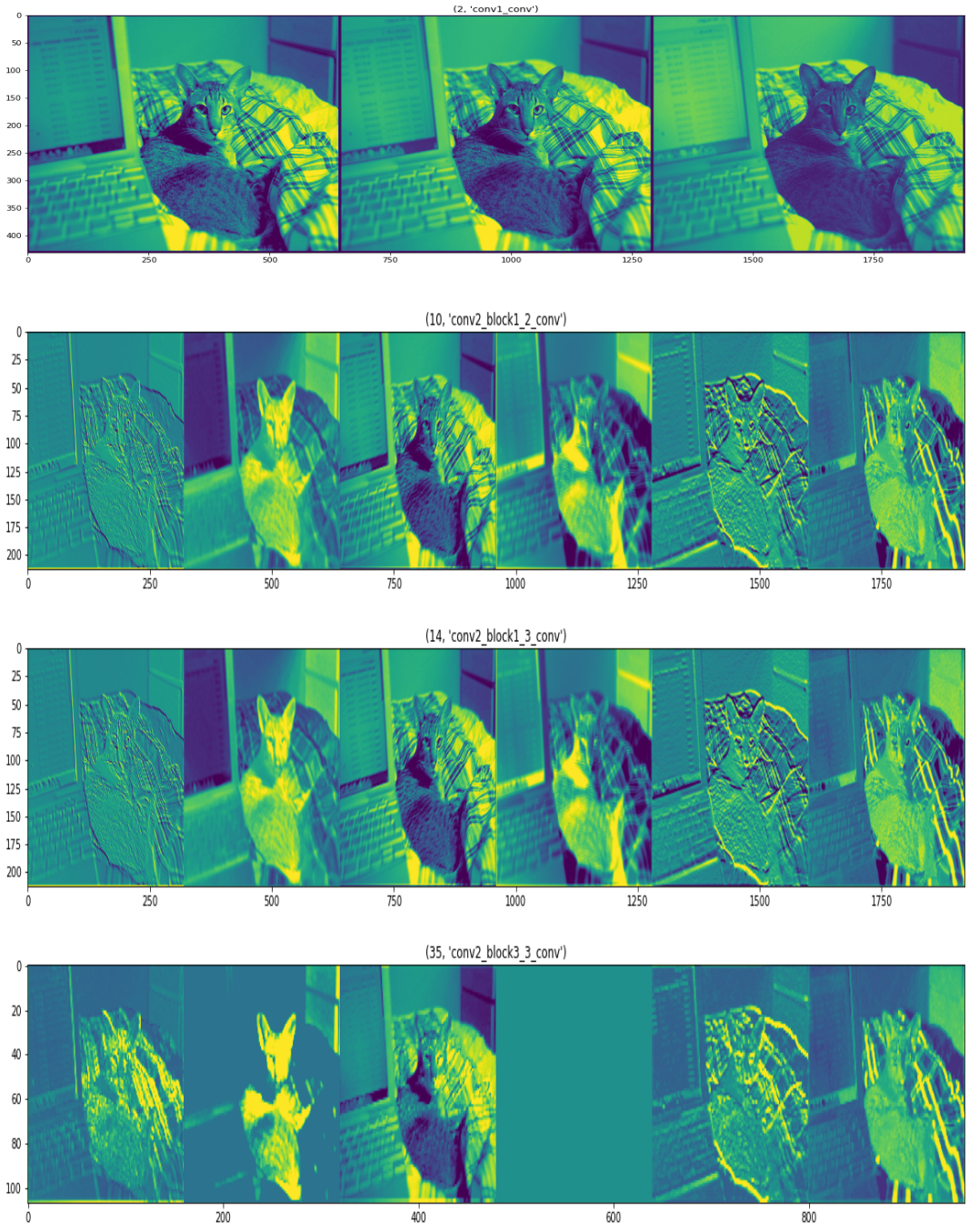
We can see that the SSD-Resnet model is always superior to the other two in terms of the mAP. This is due to the fact that the backbone Resnet-152 is better than the backbone of Faster R-CNN which is Resnet-101 and the backbone of MobileNet. Deeper ResNet model result in better accuracy. However, there is always a tradeoff between speed and accuracy. For example, SSD-ResNet152 gives better results than MobileNet in general. However, it takes only 48 ms for MobileNet to process a single frame while it takes 111 ms for SSD-ResNet152 on the same computing device. This makes MobileNet more suitable for realtime-detection when accuracy can be tolerated.

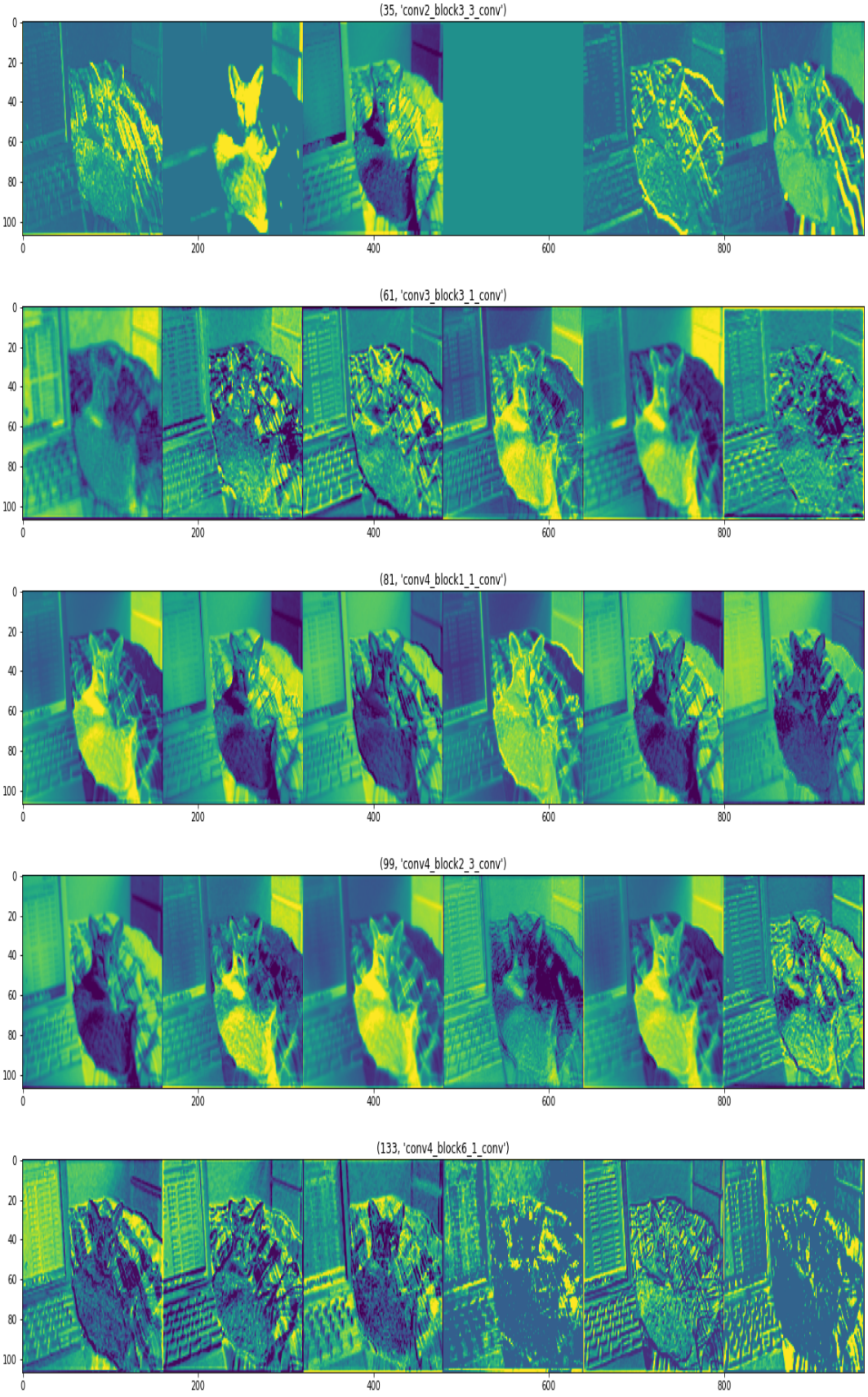
7 Other networks

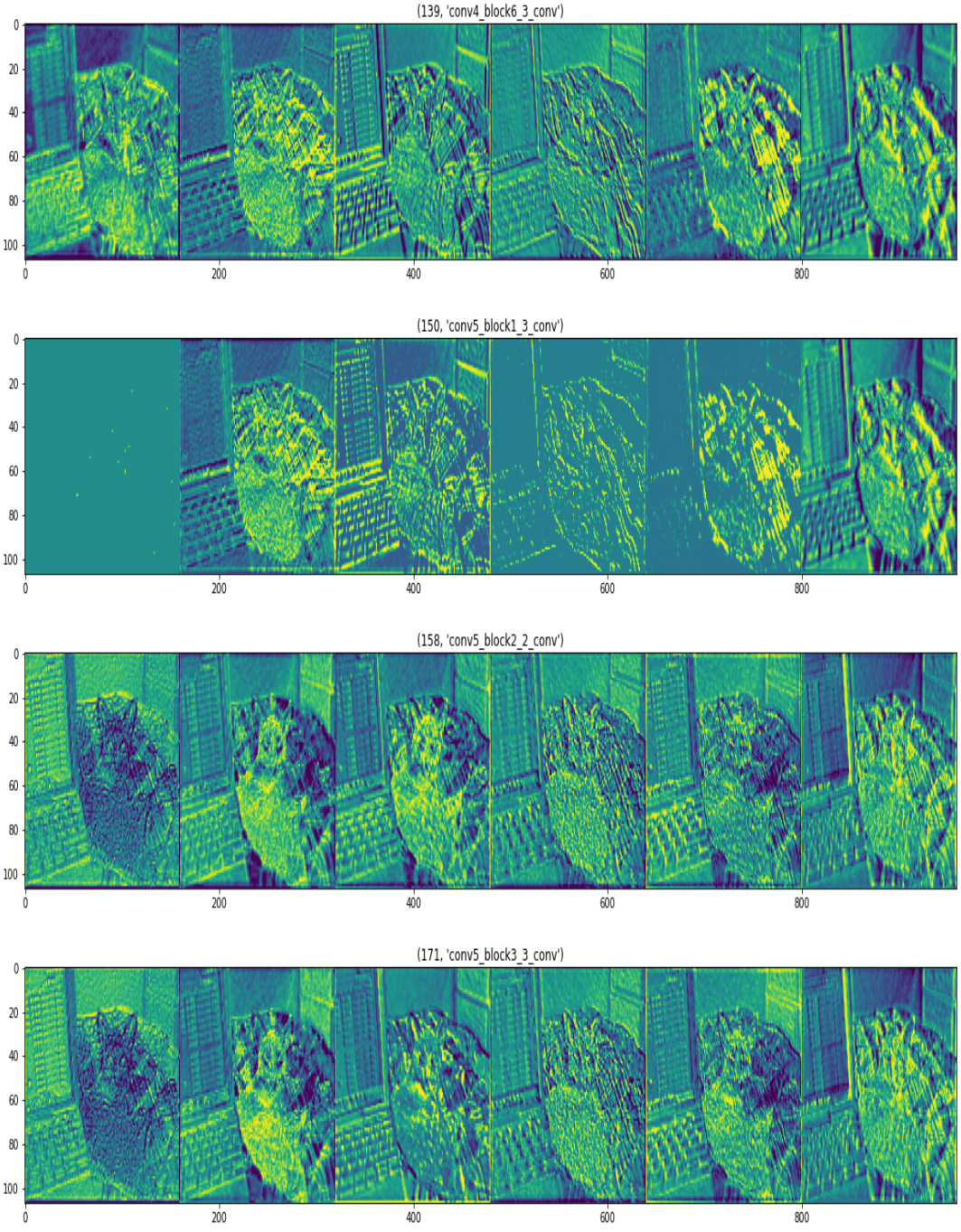
We also tried CenterNet network which resulted in a $mAP = 26.2\%$ over COCO dataset and $mAP = 42.2\%$ over Pascal-VOC dataset. However, we did not have enough time to study the network architecture.

8 Feature Maps

Feature maps are extracted from the ResNet backbone which has been used in two examples here. We can observe that in the first few layers, the model only detects general features such as edges and corners. As we go deep, the model extracts more features regarding the semantic features of the objects ecisting in the image.




(35, 'conv2_block3_3_conv')
(61, 'conv3_block3_1_conv')
(81, 'conv4_block1_1_conv')
(99, 'conv4_block2_3_conv')
(133, 'conv4_block6_1_conv')
0
20
40
60
80
100
0
200
400
600
800


(139, 'conv4_block6_3_conv')
(150, 'conv5_block1_3_conv')
(158, 'conv5_block2_2_conv')
(171, 'conv5_block3_3_conv')