SSD: Single Shot MultiBox Detector

Abstract

Our approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location.

At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape.

Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes.
SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single
This makes SSD easy to train and straightforward to integrate into systems that require a detection component.

Compared to other single stage methods, SSD has much better accuracy even with a smaller input image size.

Introduction

Current state-of-the-art object detection systems are variants of the following approach: hypothesize bounding boxes, resample pixels or features for each box, and apply a high-quality classifier.
While accurate, these approaches have been too computationally intensive for embedded systems and, even with high-end hardware, too slow for real-time applications.

the first deep network based object detector that does not resample pixels or features for bounding box hypotheses and is as accurate as approaches that do.
small convolutional filter to predict object categories and offsets in bounding box locations, using separate predictors (filters) for different aspect ratio detections, and applying these filters to multiple feature maps.

these modifications can achieve high-accuracy using relatively low resolution input, further increasing detection speed.

The Single Shot Detector (SSD)

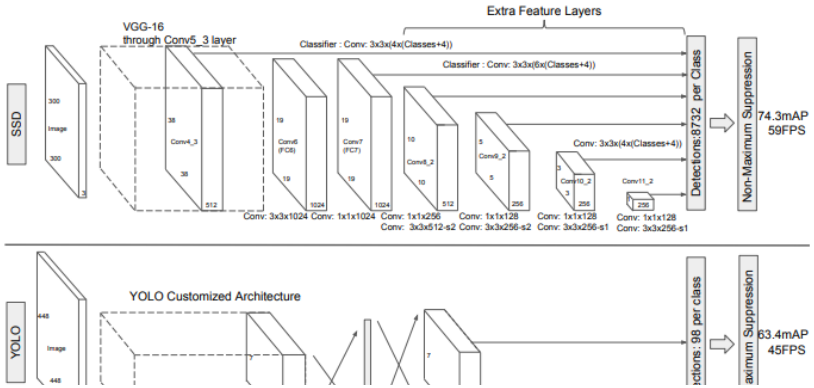
Model

The SSD approach is based on
a feed-forward convolutional network -> a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes
followed by a non-maximum suppression step to produce the final detections.

Multi-scale feature maps for detection

We add convolutional feature layers to the end of the truncated base network.

Convolutional predictors for detection



Each added feature layer can produce a fixed set of detection predictions using a set of convolutional filters.
The bounding box offset output values are measured relative to a default box position relative to each feature map location

Default boxes and aspect ratios
We associate a set of default bounding boxes with each feature map cell, for multiple feature maps at the top of the network.
The default boxes tile the feature map in a convolutional manner, so that the position of each box relative to its corresponding cell is fixed.
At each feature map cell, we predict the offsets relative to the default box shapes in the cell, as well as the per-class scores that indicate the presence of a class instance in each of those boxes.

Our default boxes are similar to the anchor boxes used in Faster R-CNN [2],
however we apply them to several feature maps of different resolutions.

Training
ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs.

Once this assignment is determined, the loss function and back propagation are applied end-to-end.
Training also involves choosing the set of default boxes and scales for detection as well as the hard negative mining and data augmentation strategies.

Matching strategy
During training we need to determine which default boxes correspond to a ground truth detection and train the network accordingly.

We begin by matching each ground truth box to the default box with the best jaccard overlap (as in MultiBox [7]).
Unlike MultiBox, we then match default boxes to any ground truth with jaccard overlap higher than a threshold (0.5).
This simplifies the learning problem, allowing the network to predict high scores for multiple overlapping default boxes rather than requiring it to pick only the one with maximum overlap.

Training objective
The SSD training objective is derived from the MultiBox objective [7,8] but is extended to handle multiple object categories.

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

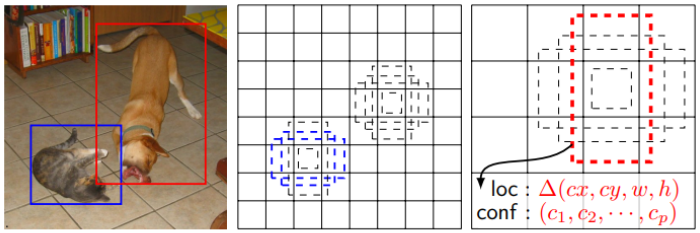
The localization loss is a Smooth L1 loss [6] between the predicted box (l) and the ground truth box (g) parameters.
Similar to Faster R-CNN [2], we regress to offsets for the center (cx, cy) of the default bounding box (d) and for its width (w) and height (h).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (3$$

The confidence loss is the softmax loss over multiple classes confidences (c).
and the weight term α is set to 1 by cross validation.

Choosing scales and aspect ratios for default boxes
Motivated by these methods, we use both the lower and upper feature maps for detection.

We design the tiling of default boxes so that specific feature maps learn to be responsive to particular scales of the objects.
By combining predictions for all default boxes with different scales and aspect ratios from all locations of many feature maps, we have a diverse set of predictions, covering various input object sizes and shapes.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Hard negative mining
After the matching step, most of the default boxes are negatives, especially when the number of possible default boxes is large.
Instead of using all the negative examples, we sort them using the highest confidence loss for each default box and pick the top ones so that the ratio between the negatives and positives is at most 3:1.

Data augmentation
– Use the entire original input image.
– Sample a patch so that the minimum jaccard overlap with the objects is 0.1, 0.3, 0.5, 0.7, or 0.9.
– Randomly sample a patch.

We keep the overlapped part of the ground truth box if the center of it is in the sampled patch.

Experimental Results

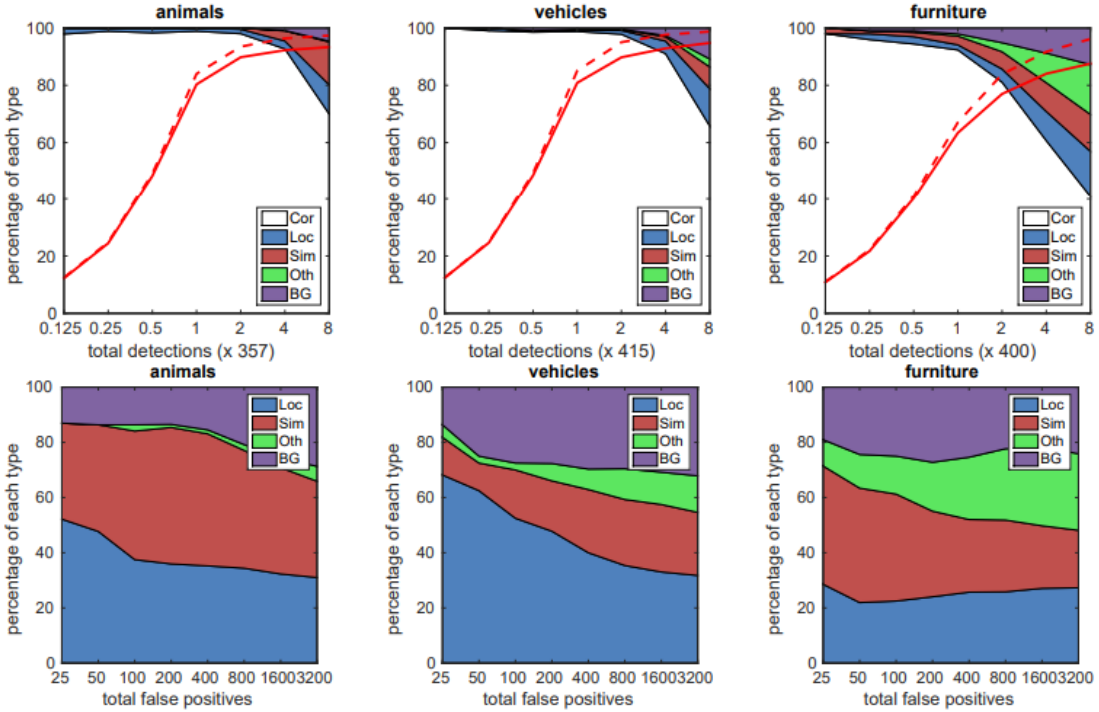
Base network
Similar to DeepLab-LargeFOV [17], we convert fc6 and fc7 to convolutional layers, subsample parameters from fc6 and fc7, change pool5 from 2 × 2 − s2 to 3 × 3 − s1, and use the a trous ` algorithm [18] to fill the ”
We remove all the dropout layers and the fc8 layer.

PASCAL VOC2007

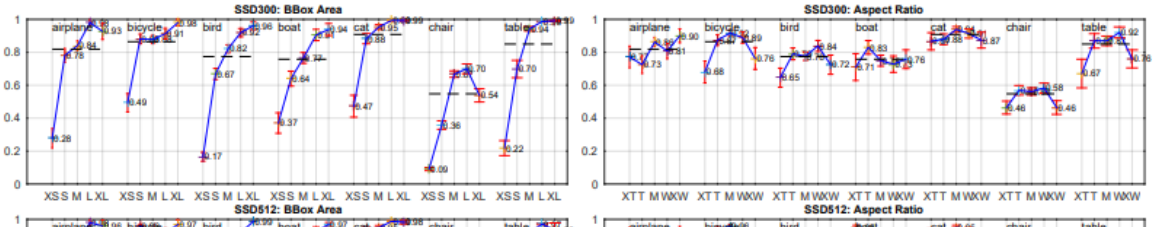
All methods fine-tune on the same pre-trained VGG16 network.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9

SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**



Compared to R-CNN [22], SSD has less localization error, indicating that SSD can localize objects better because it directly learns to regress the object shape and classify object categories instead of using two decc
However, SSD has more confusions with similar object categories (especially for animals), partly because we share locations for multiple categories.



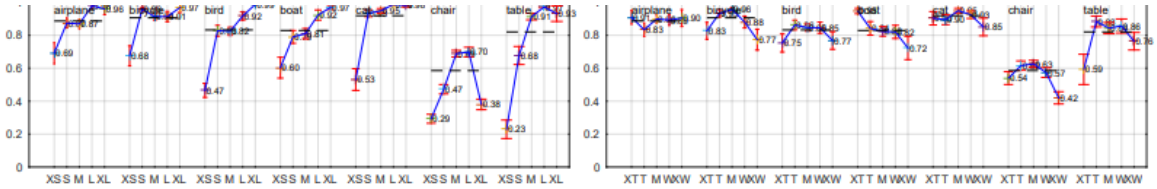


Figure 4 shows that SSD is very sensitive to the bounding box size.
Increasing the input size (e.g. from 300×300 to 512×512) can help improve detecting small objects, but there is still a lot of room to improve.

On the positive side, we can clearly see that SSD performs really well on large objects.
And it is very robust to different object aspect ratios because we use default boxes of various aspect ratios per feature map location.

Model analysis

Data augmentation is crucial

	SSD300				
more data augmentation?		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔			✔	✔
use atrous?	✔	✔	✔		✔
VOC2007 `test` mAP	65.5	71.6	73.7	74.2	**74.3**

More default box shapes is better

Atrous is faster.

Prediction source layers from:						mAP use boundary boxes?		# Boxes
conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	Yes	No	
✔	✔	✔	✔	✔	✔	74.3	63.4	8732
✔	✔	✔	✔	✔		**74.6**	63.1	8764
✔	✔	✔	✔			73.8	68.4	8942
✔	✔	✔				70.7	69.2	9864
✔	✔					64.2	64.4	9025
	✔					62.4	64.0	8664

Table 3: **Effects of using multiple output layers.**

Multiple output layers at different resolutions is better

A major contribution of SSD is using default boxes of different scales on different output layers.

To measure the advantage gained, we progressively remove layers and compare results.

For a fair comparison, every time we remove a layer, we adjust the default box tiling to keep the total number of boxes similar to the original (8732).

When we stack boxes of multiple scales on a layer, many are on the image boundary and need to be handled carefully.

We observe some interesting trends. For example, it hurts the performance by a large margin if we use very coarse feature maps (e.g. conv11 2 (1 × 1) or conv10 2 (3 × 3)).

If we only use conv7 for prediction, the performance is the worst, reinforcing the message that it is critical to spread boxes of different scales over different layers.

Besides, since our predictions do not rely on ROI pooling as in [6], we do not have the collapsing bins problem in low-resolution feature maps [23].

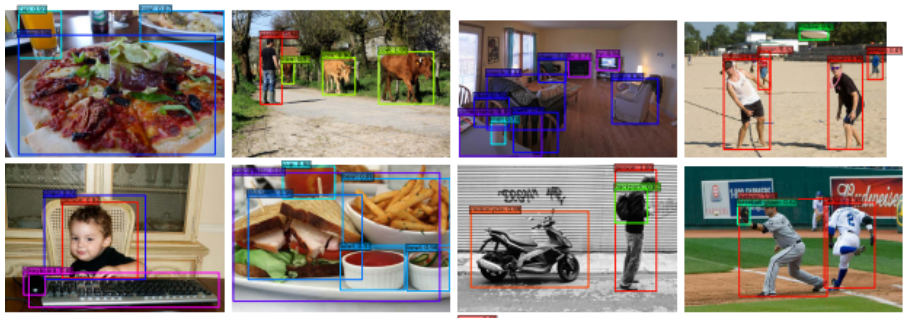
The SSD architecture combines predictions from feature maps of various resolutions to achieve comparable accuracy to Faster R-CNN, while using lower resolution input images.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast[6]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster[2]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
Faster[2]	07++12+COCO	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2
YOLO[5]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD300	07++12+COCO	77.5	90.2	83.3	76.3	63.0	53.6	83.8	82.8	92.0	59.7	82.7	63.5	89.3	87.6	85.9	84.3	52.6	82.5	**74.1**	**88.4**	74.2
SSD512	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
SSD512	07++12+COCO	**80.0**	**90.7**	**86.8**	**80.5**	**67.8**	**60.8**	**86.3**	**85.5**	**93.5**	**63.2**	**85.7**	**64.4**	**90.9**	**89.0**	**88.9**	**86.8**	**57.2**	**85.1**	72.8	**88.4**	**75.9**

COCO

Method	data	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
		0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast [6]	train	19.7	35.9	-	-	-	-	-	-	-	-	-	-
Fast [24]	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster [2]	trainval	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [24]	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster [25]	trainval	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300	trainval35k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512	trainval35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**

Table 5: **COCO `test-dev2015` detection results.**

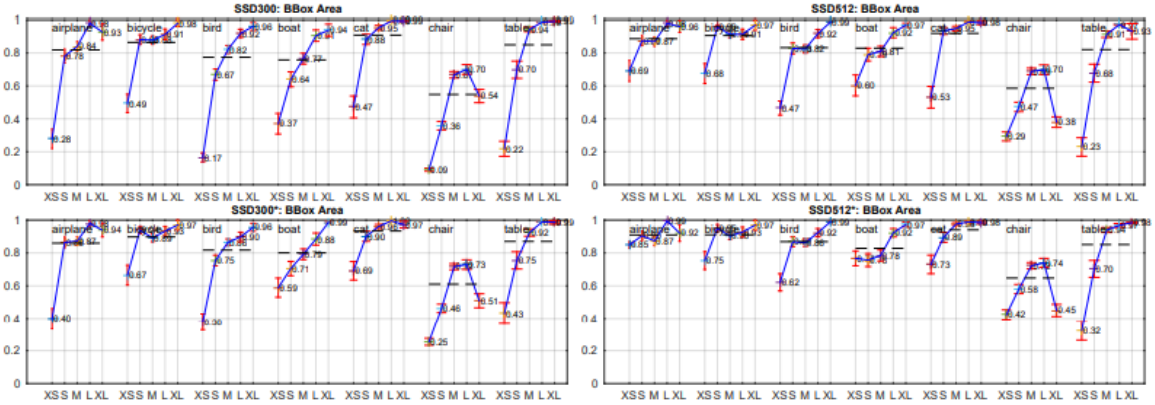


Preliminary ILSVRC results

We applied the same network architecture we used for COCO to the ILSVRC DET dataset [16].

Data Augmentation for Small Object Accuracy

Method	VOC2007 test		VOC2012 test		COCO test-dev2015 trainval35k		
	07+12	07+12+COCO	07++12	07++12+COCO			
	0.5	0.5	0.5	0.5	0.5:0.95	0.5	0.75
SSD300	74.3	79.6	72.4	77.5	23.2	41.2	23.4
SSD512	76.8	81.6	74.9	80.0	26.8	46.5	27.8
SSD300*	77.2	81.2	75.8	79.3	25.1	43.1	25.8
SSD512*	**79.8**	**83.2**	**78.5**	**82.2**	**28.8**	**48.5**	**30.3**



Inference time

Considering the large number of boxes generated from our method, it is essential to perform non-maximum suppression (nms) efficiently during inference.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

SSD512	76.8	22	8	24564	512 x 512

Related Work

Our SSD is very similar to the region proposal network (RPN) in Faster R-CNN in that we also use a fixed set of (default) boxes for prediction, similar to the anchor boxes in the RPN.
Thus, our approach avoids the complication of merging RPN with Fast R-CNN and is easier to train, faster, and straightforward to integrate in other tasks.

Another set of methods, which are directly related to our approach, skip the proposal step altogether and predict bounding boxes and confidences for multiple categories directly.
Our SSD method falls in this category because we do not have the proposal step but use the default boxes.

If we only use one default box per location from the topmost feature map, our SSD would have a similar architecture to OverFeat [4];
if we use the whole topmost feature map and add a fully connected layer for predictions instead of our convolutional predictors, and do not explicitly consider multiple aspect ratios, we can approximately reproduce Y

Conclusions

This paper introduces SSD, a fast single-shot object detector for multiple categories.

A key feature of our model is the use of multi-scale convolutional bounding box outputs attached to multiple feature maps at the top of the network.
We experimentally validate that given appropriate training strategies, a larger number of carefully chosen default bounding boxes results in improved performance.

Apart from its standalone utility, we believe that our monolithic and relatively simple SSD model provides a useful building block for larger systems that employ an object detection component.
A promising future direction is to explore its use as part of a system using recurrent neural networks to detect and track objects in video simultaneously.