CDSSD: Refreshing Single Shot Object Detection Using A Conv-Deconv Network

Vijay Gabale[1] and Uma Sawant[2]

[1] Huew, India. vijay@huew.co
[2] IIT Bombay, India. uma.sawant@gmail.com

Abstract. Single shot multi-box object detectors [13] have been recently shown to achieve state-of-the-art performance on object detection tasks. We extend the single shot detection (SSD) framework in [13] and propose a generic architecture using a deep convolution-deconvolution network. Our architecture does not rely on any pretrained network, and can be pretrained in an unsupervised manner for a given image dataset. Furthermore, we propose a novel approach to combine feature maps from both convolution and deconvolution layers to predict bounding boxes and labels with improved accuracy. Our framework, Conv-Deconv SSD (CDSSD), with its two key contributions – unsupervised pretraining and multi-layer confluence of convolution-deconvolution feature maps – results in state-of-the-art performance while utilizing significantly less number of bounding boxes and improved identification of small objects. On 300×300 image inputs, we achieve 80.7% mAP on VOC07 and 78.1% mAP on VOC07+12 (1.7% to 2.8% improvement over StairNet [21], DSSD [5], SSD [13]). CDSSD achieves 30.2% mAP on COCO performing at-par with R-FCN [3] and faster-R-FCN [18], while working on smaller size input images. Furthermore, CDSSD matches SSD performance while utilizing 82% of data, and reduces the prediction time per image by 10%.

Keywords: Single Shot Detection, Unsupervised Learning, Feature Map Confluence

1 Introduction

Image object detection involves identifying bounding boxes encapsulating objects and classifying each bounding box to recognize the underlying object category. Recently there has been mounting interest in the research community to detect multiple objects in an image using Single Shot Detection techniques [13, 16]. These techniques effectively combine region proposal and classification into a single step by foregoing the candidate box proposal (or region proposal) module employed by several two-step detection techniques [6, 7, 18, 1, 11]. Not only this results in much faster object detection but it also improves accuracy [13, 16, 5, 21]. One of the two prominent works, You Only Look Once (YOLO) [16], considers the global feature map of an image and utilizes a fully-connected layer to output object detections with a fixed set of regions. The other prominent work,

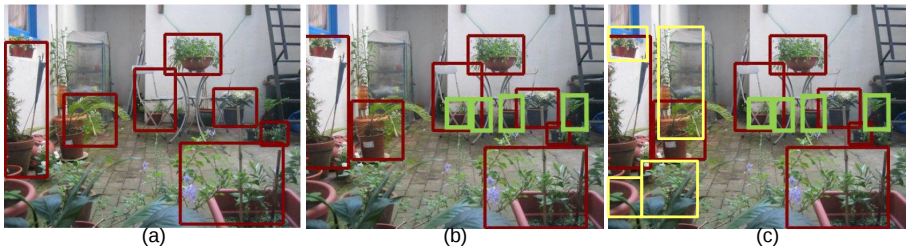

Fig. 1. Detection output comparison of (a) SSD [13], (b) Stairnet [21], and (c) CDSSD. CDSSD results in superior performance in detecting small as well as large objects

Single Shot MultiBox Detector (SSD, henceforth) [13], considers a set of layers (or feature maps) and a set of boxes at various scales, and employs convolutional filters to predict objects inside each box. Owing to its design choice to consider multiple feature maps from different layers in a deep network (multi-scale representation), SSD performs significantly better than YOLO.

While SSD [13] has achieved state-of-the-art results, it has three fundamental drawbacks. (a) When applying default bounding boxes, SSD considers each feature map in isolation (see Fig. 2). Thus it can not exploit the semantic information of later layers for better object detection on initial layers. Consequently, SSD does not perform well on smaller size object detection which is attempted by initial layers. (b) SSD architecture relies on features maps pretrained on the classical Imagenet dataset [20, 9] without attempting to learn robust feature maps from the vast collection of unlabeled datasets. (c) SSD needs to evaluate several thousands of bounding boxes to detect only a few objects in an image.

Several follow-up works attempt to eliminate limitation (a) by combining feature maps at different layers of convolution networks, or inserting additional context by extending the base convolution block with a deconvolution block [16, 13, 5, 21, 17, 10, 1, 11, 2]. However, none of the prior approaches explore unsupervised pretraining to learn robust features; but use either VGG-16 [20] or ResNet-101 [9] to bootstrap the object detection training. [16, 13, 5, 21, 17, 11] partially exhibit some scope to improve the performance on objects of different sizes and scales by combining information from different feature maps. However, they rely on features computed only from convolution networks, or result in consideriably slower speed detection [5], or are not end-to-end trainable. In contrast to this prior work, we draw inspirations from convolution-deconvolution techniques used in semantic segmentation tasks [15, 22], and base our design on convolution auto-encoders. Specifically, our contributions are as follows:

- We design an end-to-end trainable convolution-deconvolution based single shot detection framework to detect multiple objects in an image. This framework enables unsupervised pretraining of the underlying network.
- We design a refined SSD technique that carefully combines feature maps from both convolution and deconvolution blocks. Fusing of generic features from initial layers close to the input with semantically rich features of later

layers close to the output detection from *both convolution and deconvolution blocks* helps us significantly reduce the required number of default bounding boxes.
- On input image size of 300×300, we achieve state-of-the-art accuracy on several object detection tasks with 80.7% mAP on VOC07, 78.1% mAP on VOC07+12 (1.7% improvement over StairNet [21, 5], 2.8% improvement over [13]), and 30.2% mAP on COCO. We improve detection performance of both small as well as large objects, as well as visually impoverished objects while reducing the prediction time per image by 10%.

2 Limitations of Related Work

As compared to SSD, some recent approaches [6, 7, 18] first learn a separate bounding box (or region) proposal network, followed by learning a separate classification network on top of the proposal network. However, such two-stage object detectors suffer from high memory usage and poor inference time. In comparison, SSD networks [19, 13, 16] have been shown to perform better and faster. Furthermore, most of the object detection techniques, including Overfeat [19], SPPnet [8], Fast R-CNN [6], Faster R-CNN [18], and YOLO [16], utilize only a single layer (typically the top-most layer) of a convolution network to detect objects. This approach does not exploit different feature sets learned by different feature maps at different scales [13, 5, 21], and therefore is severely limited in identifying objects of different sizes and scales. In comparision, the state-of-the-art SSD networks [13, 17] utilize feature maps from different layers in order to focus on objects that appear in certain sizes. However, they operate on each feature map independently without combining them in a meaningful manner. Hence, these SSD networks [13, 17] do not particularly perform well towards identification of smaller size objects [1, 11, 5, 21].

In order to consider feature maps from different layers in a combined fashion, [1] concatenates features of different layers before applying box proposals to detect objects. Taking a step further, [2] applies deconvolution on multiple layers of the underlying convolution network to increase feature map resolution. However, it results in significant memory and prediction time requirement. [11] too leverages the pyramidal shape of the convolution network and attempt to utilize semantics at different scales of feature maps by inserting nearest neighbor upsampling. In another work, instead of focusing only on the convolution block, [5] adds a deconvolution context layer to address the problem of shrinking resolution of feature maps in the convolution block. [21] further exploit the deconvolution context and design a top-down feature combining module that progressively encodes semantic information with low level features.

Our approach is partially inspired by [5, 21] in terms of adding deconvolution context and utilizing feature maps at different layers in a network. However, as shown in Fig. 2 neither [5, 21] nor any of the prior approaches explore unsupervised learning to improve SSD [13] performance. Moreover, none of the prior work exploits the difference in features learned by different layers in both convolution and deconvolution blocks. By refreshing SSD with unsupervised learning

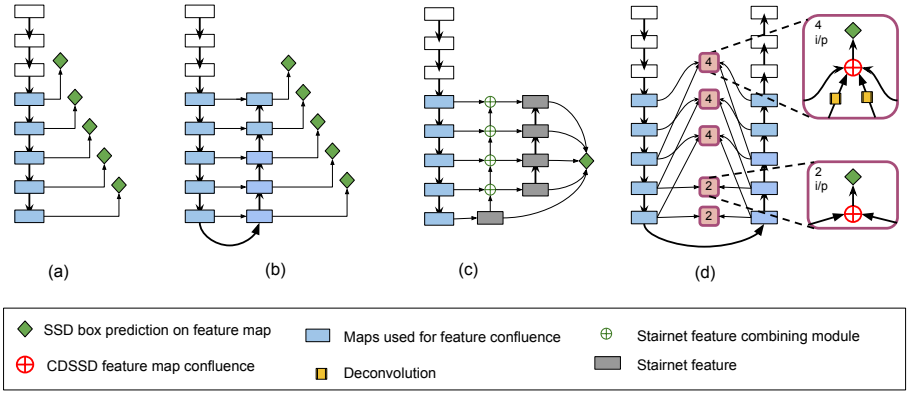

Fig. 2. Difference in SSD architectures in using deconvolution and feature map confluence (a) SSD [13], (b) DSSD [5], (c) Stairnet [21], (d) CDSSD (this work)

and confluence of feature maps from convolution and deconvolution blocks, we show that our approach results in state-of-the-art performance on benchmark datasets [12, 4].

3 CDSSD Architecture

In this section, we first give a primer on SSD architecture. We then progressively introduce unsupervised learning and feature map confluence in the SSD architecture. Finally we showcase a method to reduce the requirement of default bounding boxes, and then explain our methodology of training and testing.

3.1 SSD

The SSD network is a convolutional architecture that utilizes different layers to predict presence of multiple objects in an image. To recognize objects at different scales, SSD utilizes predictions on different feature maps, each from a different layer, of a single network. These feature maps are processed by a fixed-size collection of bounding boxes customized for each layer. For feature map f of size $m \times n$ with p channels, K default-sized bounding boxes are applied on each of $m \times n$ cells. Subsequently, C filters of size $3 \times 3 \times p$ are applied for each cell and for a given bounding box to produce individual scores to predict each of C classes, and 4 additional filters are applied to produce offsets (center co-ordinate, height, width) to position the box on the underlying cell in order to encapsulate the object (as shown in Fig. 3(c)). Note that, for a given feature map f, the default boxes are scaled with a scaling factor f^{scale} with respect m and n and thus, they are customized to have different aspect ratios. Hence, bounding boxes on initial stage feature maps cover a smaller receptive field to identify objects at a smaller scale, whereas bounding boxes on later stage feature maps cover larger receptive fields to identify objects with larger scale. By utilizing predictions for

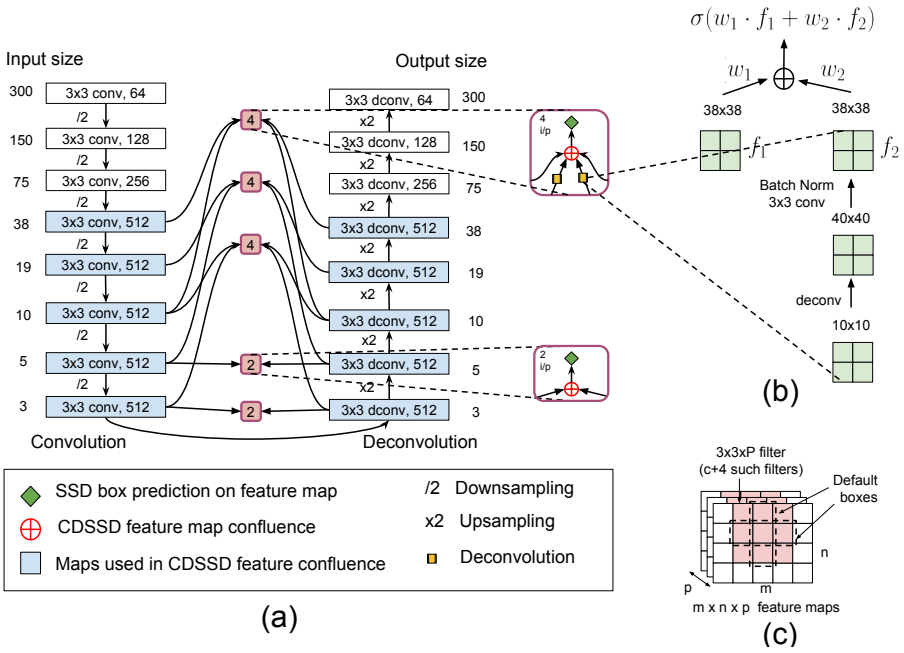

Fig. 3. CDSSD combines information from convolution and deconvolution feature maps

all the default boxes with different scales and aspect ratios from all locations of many feature maps, SSD attempts a diverse set of predictions, covering various input object sizes and shapes.

3.2 Unsupervised pretraining

Our first fundamental improvement to SSD is to facilitate unsupervised training of the underlying network architecture. As we show in Section 4, this results in significant performance improvement. We use ResNet 101 architecture [9] and construct a convolution-deconvolution based auto-encoder (shown in Fig. 3(a)). Previously [5] have shown that ResNet 101 architecture results in more than 1.4% mAP gain in SSD as compared to VGG16 [20]. For the deconvolution block, we use learned upsampling and learned deconvolution, instead of bi-linear upsampling. The deconvolution block produces an image of the same dimension as input. We use an input image of $300 \times 300 \times 3$, with 7 meta-layers of convolution and pooling and 7 meta-layers of deconvolution with learned upsampling. Given an image dataset, we first pretrain the architecture before applying supervised object detection. [3] Since our architecture is based on fully convolution networks, CDSSD can in fact process any arbitrary sized images.

[3] Our network is not symmetric. During deconvolution, we simply apply learned upsampling and learned deconvolution without residual blocks.

3.3 Combining feature maps

[23, 14] observe that the initial layers of a deep network lack strong semantic information and respond to only high-level features of an image. Furthermore, the improvement in acquiring semantic information across consecutive feature maps is only marginal, especially in initial layers of a network. Based on these observations, our second fundamental improvement to SSD is to fuse generic and semantic features to enrich feature maps. Unlike prior work, we combine features from different layers of both convolution and deconvolution network (Fig. 3(a)).

To augment feature maps from layers at different levels, firstly, we combine layer l with layer $l + level\_stride$. Based on observations in [23, 14], we do not fuse consecutive layers, but set $level\_stride$ as 2 to receive sufficient semantic information gain. However, since different layers have different sizes as well as different scales of bounding box, we apply a learnable upscaling operation on layer $l + level\_stride$ (Fig. 3 (b)) to combine them effectively. The scaling operation ensures that the resulting feature map has the same dimension as layer l while it also accounts for semantic information contained in layer $l+level\_stride$. For example, as shown in Fig. 3 (b), to scale 10×10 feature map, we first apply $4 \times 4 \times 512$ deconvolution operation and then apply a $3 \times 3 \times 512$ convolution operation to reduce the feature map size to 38×38. This is followed by a batch normalization layer to receive the final 38×38 feature map. Note that, we apply similar operation on both convolution and deconvolution blocks to process different layers. Addition of context from deconvolution block only improves the performance as we show in Section 4, without affecting the detection speed.

Secondly, for a given level of a meta-layer, we combine all the four feature maps; two from the convolution block and two from deconvolution block, as shown in Fig. 3(a), into a final feature map by taking element-wise learnable ReLU operation. Based on observations in [5], we further apply 3×3 filter on this feature map to extract another layer of features. Similar to SSD, we then apply a set of K default-boxes and $(C + 4) \times m \times n \times K$ filters on the resulting feature map to predict detection of objects. We apply this set of operations on meta-layer 3 to meta-layer 5 as shown in Fig. 3 (a). Since there are no feature maps to pair with the last $level\_stride$ of convolution and initial $level\_strides$ of deconvolution feature maps (6th and 7th meta-layer), we combine them in element-wise learnable ReLU and process the resulting feature map. Since 6th and 7th meta-layers have higher reception field and contain richer semantic information, they are quite capable of detecting bigger size and scale objects [5].

3.4 Box pooling: reducing the number of default boxes

In the original SSD implementation, the authors apply default bounding boxes to every cell of $m \times n$ feature map with p channels. We consider a *box-pooling approach* where we pick the dominant cell in $l \times l$ window, with a stride of l on $m \times n$ feature map, and apply a set of default boxes on the dominant cell. This reduces the number default boxes by l^2 per feature map. This design choice is governed by two phenomena observed during our ablation study: (1)

Unsupervised pre-training helps in learning significantly better feature maps (2) Given that we combine feature maps from different layers of both convolution and deconvolution blocks, there is no need to exhaustively search for objects for every cell of of every feature map. We show in Section 4 that box-pooling does not affect precision and recall of object detection.

Similar to SSD [13], we tile the default boxes of different scales on different features maps so that specific feature maps learn to be responsive to particular scales of the objects. To compute different aspect ratios for each cell, we take a statistical approach and compute a cumulative distribution of aspect ratios of the ground truth boxes in a given dataset. We then divide the distribution into B bins and pick the average value of a bin as one of the aspect ratio, thus resulting in B aspect ratios. For each $b_i \in B$, for a feature map with size $m \times n$ and scale of f^{scale}, we then set height to be $m \times b_i \times f^{scale}$ and width to be $n \times b_i \times f^{scale}$. With optimized aspect ratios that fit the underlying dataset and different scales for different layers, we apply appropriate default boxes at box-pooled locations in each feature map, covering different object sizes and shapes.

4 Results

Our experiments are governed to answer the following key question: *can we achieve state-of-the-art results on object detection benchmarks by employing unsupervised learning and confluence of feature maps from convolution and deconvolution blocks?* Towards answering this question, we compare our approach with prior work on two benchmark datasets: PASCAL VOC and MS COCO. We compare our approach with the original SSD [13] that employs only convolution block, DSSD [5] that uses deconvolution blocks as additional context for convolution blocks, and Stairnet [21] that progressively merges feature maps close to traditional classification layers with feature maps close to input layers. SSD, DSSD and Stairnet do not employ unsupervised learning and do not consider confluence contextual and semantic features from convolution and deconvolution blocks. We also do an extensive ablation study to quantify improvement by each of the modules that we have contributed to extend SSD framework. We develop CDSSD as a Tensorflow module.

4.1 Training

The configuration of our network architecture is shown in Fig. 3. We keep the dropout layers during unsupervised training and remove them while training for object detection. We train our models on Azure GPU instances that have NVIDIA K80 GPUs with 12GB of memory. We use batch size of 16, momentum as 0.9 and weight decay 0.0005. Similar to SSD [13], we match a default box to target ground truth boxes, if Jaccard overlap is larger than a threshold (e.g. 0.5). We compute the target ground truth box for each layer of the network by scaling it with respect to the feature map and original image sizes. We minimize the joint localization loss (i.e., smooth L1) and confidence loss (i.e., softmax-cross-entropy). To avoid the imbalance between the positive and negative training

examples, we sort the negative boxes using the joint loss for each default box and then pick the top ones to maintain a 2:1 negative to positive ratio. We found 2:1 ratio leads to faster optimization as compared to the ratio of 3:1 as mentioned in the original SSD paper.

We further make the model robust to different input object sizes and shapes by invoking extensive augmentation. Specifically, we sample a patch from a ground truth box so that the minimum Jaccard overlap with the objects is 0.5, 0.7, or 0.9. Furthermore, we randomly sample a patch between [0.5, 1] of the original image size, and the aspect ratio is between [1, 2]. Also, we randomly flip each patch horizontally with probability of 0.5, apply different transformations such as gaussian blur, emboss, edge prominence, random black-out of 20% of pixels, and color (hue, saturation, contrast) distortions. We apply 3×3 box pooling for layer 3 and 4, 2×2 box pooling for layer 5, and no box pooling for layer 6 and 7. We apply non-maximum suppression (NMS) to post-process the predictions to get final detection results.

4.2 PASCAL VOC

method	network	mAP	boxes	fps	lib
YOLOv2_352 [16]	DarkNet-19	73.7	98	81	DarkNet
SSD300 [13]	VGGNet	77.5	8732	62	Caffe
DSSD321 [5]	ResNet-101	78.6	43688	9.5	Caffe
Stairnet [21]	VGGNet	78.8	8732	30	PyTorch
CDSSD300	ResNet-101	**80.7**	1182	51	TF
CDSSD300 (82% data)	ResNet-101	77.9	1182	51	TF

Table 1. Comparison of single-shot detection techniques trained on VOC07+12 trainval and evaluated on VOC2007 test dataset. CDSSD outperforms other state-of-the-art methods while maintaining high speed of detection.

When training on VOC07+12 trainval, we train the entire network with learning rate at 10^{-3} for 45K batches, and then with learning rate of 10^{-4} for 60K batches to execute unsupervised pretraining on the underlying train dataset [4]. During object detection training, we again fine-tune the entire network with learning rate of 2×10^{-3} for 40K iterations, and 60K iterations with learning rate of 10^{-4}. Results over VOC07 test dataset are shown in Tab. 1. To evaluate on VOC12 test dataset, as shown in Tab. 2, we use VOC07 trainvaltest, VOC12 trainval for training. We train CDSSD model for 65K iterations with 10^{-3} learning rate and 2×10^{-4} learning rate for 80k iterations for unsupervised pretraining, and 10^{-3} and 10^{-4} learning rate for supervised training for 40K and 65K iterations respectively.

We see that by adding unsupervised pretraining and confluence of feature maps, CDSSD consistently outperforms SSD, DSSD, Stairnet by 1% to 5% points for several object categories. CDSSD especially shows significant improvement for small objects such as bird, tv and bottle. Furthermore, CDSSD also shows

[4] Due to reduced batch size, the number of batches or iterations are increased as compared to the original SSD work.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
SSD300 [13]	**88.1**	82.9	74.4	61.9	47.6	82.7	**78.8**	91.5	58.1	80.0
DSSD321 [5]	87.3	83.3	75.4	64.6	46.8	82.7	76.5	**92.9**	59.5	78.3
StairNet [21]	87.7	83.1	74.6	64.2	51.3	**83.6**	78.0	92.0	58.9	**81.8**
CDSSD224	85.2	79.5	71.4	60.1	44.5	79.1	74.8	84.3	57.9	79.2
CDSSD300	87.4	**83.9**	**78.3**	**69.5**	**54.7**	80.2	76.3	88.7	**63.4**	79.9
CDSSD300 (82%)	85.8	82.7	75.3	64.5	50.5	80.1	75.2	85.8	60.0	78.4

method	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SSD300 [13]	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
DSSD321 [5]	64.3	**91.5**	86.6	**86.6**	82.1	53.3	79.6	75.7	85.2	73.9
StairNet [21]	66.2	89.6	86.0	84.0	**82.6**	50.9	**80.5**	71.8	**86.2**	73.5
CDSSD224	63.8	85.1	84.3	84.3	82.9	52.4	77.2	72.8	83.6	72.8
CDSSD300	**69.2**	89.3	**87.8**	85.6	82.3	**56.8**	76.9	**76.2**	84.3	**77.4**
CDSSD300 (82%)	66.4	83.4	82.1	84.7	80.3	53.7	75.8	71.9	80.6	74.5

Table 2. mAP comparison of single-shot detection techniques trained on VOC07 trainvaltest, VOC12 trainval and evaluated on VOC12 test dataset. CDSSD results in state-of-the-art performance for several object categories.

significant improvement for objects such as boat and horse that have definite backgrounds. CDSSD detects majority objects with high confidence with less localization error and less confusion for similar object categories [5]. Recall of CDSSD is 93.5% for "strong" criteria of jaccard of overlap of 0.5, about 10% better than SSD. Finally, CDSSD achieves high-precision at high-recall range and outperforms SSD and Stairnet (Tab. 3).

method	data	recall			
		0.5	0.7	0.9	mAP@70%
SSD300	07+12	91.9	79.7	34.4	44.9
Stairnet	07+12	94.3	83.5	38.8	48.1
CDSSD	07+12	**96.1**	**87.0**	**44.2**	**52.6**

Table 3. VOC 2012 test dataset to observe mAP at recall greater than 0.7

4.3 Ablation study

To further quantify the benefits of CDSSD, we do an ablation study to progressively add its features and measure mAP on VOC12 test dataset. To quantify the performance over different sized objects, we consider objects of three different sizes. Following the methodology in [21], we order the ground truth bounding boxes on test set for each class by area. We further divide the boxes into three part: small: less than 25%, medium: between 25% to 75%, and large: above 75% of image size. Furthermore, when evaluating objects of each size, we ignore the the ground truth labels for other sizes. As shown in Tab. 4, CDSSD shows significant improvement using confluence of feature maps, on individual convolution and deconvolution blocks as well as combination of convolution and deconvolution feature maps. CDSSD especially shows considerable improvement on small size objects; it performs about 9% to 14% mAP better than prior work.

[5] Details omitted due to lack of space

conv-feat confluence	deconv-feat confluence	box pooling	unsup pretraining	total boxes	overall mAP	small-O mAP	medium-O mAP	large-O mAP
no	no	no	no	17464	74.5	42.6	76.9	80.6
no	no	yes	no	1182	70.4	35.1	71.5	75.3
yes	no	no	no	17464	74.9	46.5	77.1	80.9
no	yes	no	no	17464	75.4	47.9	77.8	81.8
no	yes	yes	no	1182	74.5	45.2	76.6	78.9
yes	yes	no	no	8752	76.2	56.5	80.2	83.7
yes	yes	no	yes	8752	**78.3**	**59.0**	**81.6**	**85.0**
yes	yes	yes	yes	1182	78.1	57.4	81.2	84.7

Table 4. Effects of progressively adding confluence of feature maps on convolution block, deconvolution block, unsupervised learning, and box pooling. Box pooling does not hamper the performance while drastically reducing the box requirement.

To quantify the performance of unsupervised pretraining when not pretrained on the underlying dataset, we train our convolution and deconvolution network on imagenet dataset to initialize the weights of the network (similar to SSD [13], DSSD [5], Stairnet [21]). From the table, we also observe that unsupervised learning gives a 2.1% jump in overall mAP. Furthermore, after applying box pooling, i.e, after reducing the number of boxes from 8732 to 1182, we observe that CDSSD sees only marginal reduction in mAP. Note that, box pooling is not effective without unsupervised learning and confluence of feature maps as shown in Tab. 4. Thus, combining unsupervised learning with feature map confluence and box pooling, CDSSD results in state-of-the-art results on object detection datasets while reducing the number of default bounding boxes.

The original version of SSD [13] uses 8732 boxes, DSSD uses substantially more (17080 to 43688 boxes), whereas CDSSD uses only 1183 boxes. As a result, SSD takes 46 FPS and DSSD takes 9.5 FPS where CDSSD clocks 51 FPS on Titan X GPU with a batch size of 1. While Residual-101 network is slower than VGGNet used in SSD, the reduction in default boxes not only decreases the prediction time but also time spent in non maximal suppression. Furthermore, the extra deconvolution layers do not incur an overhead since the confluence operation is light weight, and CDSSD operates on the same number of feature maps as the original SSD. Thus, CDSSD achieves improved accuracy while maintaining one of the fastest detection performance.

4.4 MSCOCO

To evaluate CDSSD on MSCOCO dataset, we first optimize the sizes of default bounding boxes as per the dataset (as explained in Section 3.4) to train and test prediction of classes and offsets. We train the network in an unsupervised manner for 260K iterations with learning rate of 10^{-3}. We use the trainval35k dataset and train the network in a supervised fashion for 210K iterations with learning rate of 10^{-3} and 120K iterations with learning rate of 2×10^{-4}. We show the results on test-dev2015. As shown in Tab. 5, CDSSD performs consistently better than SSD and DSSD even at higher Jaccard overlap threshold (0.75), and

method	avg.precision, IoU 0.5:0.95/0.5/0.75	avg.precision, area S/M/L	avg.recall, #Dets 1/10/100	avg.recall, area S/M/L
SSD300	25.1/43.1/25.8	6.6/25.9/41.4	23.7/35.1/37.2	11.2/40.4/58.4
DSSD321	28.0/46.1/29.2	7.4/28.1/47.6	25.5/37.1/39.4	12.7/42.0/62.6
CDSSD300	29.2/48.2/29.9	8.8/31.2/49.3	26.1/39.2/42.3	13.6/44.3/63.7

Table 5. Evaluation of CDSSD on MSCOCO dataset

for different sized objects. Improvement in detection of large objects shows that CDSSD is able to learn better and robust features. These results corroborate the benefits of CDSSD on generic object detection datasets towards a better single-shot detection framework. Fig. 4 shows object detections on COCO test set images. Our model shows improvements on several fronts such as small objects like donuts; dense objects e.g. airplanes; objects with distinct context such as clocks; and objects that have specific relationships with the background.

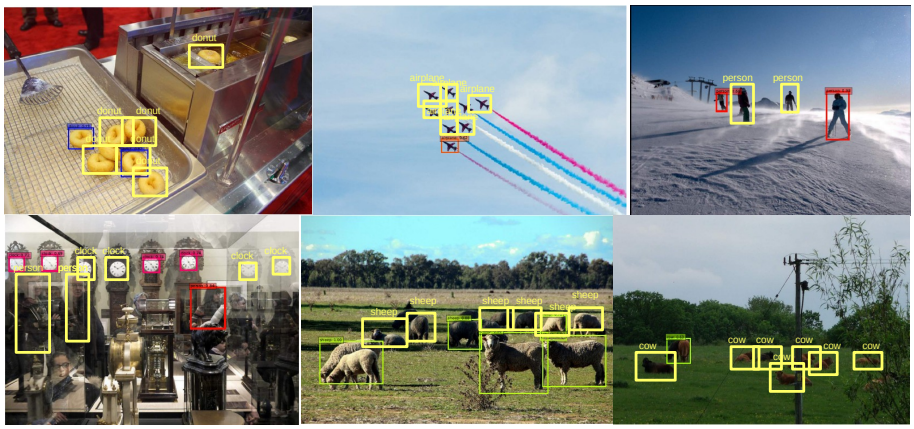

Fig. 4. CDSSD out-performs in capturing objects of different size and scale in comparison to SSD [13]

5 Conclusion

We design an end-to-end framework using convolution-deconvolution deep networks to improve the state-of-the-art of single shot object detection techniques. Using a combination of unsupervised learning and confluence of feature maps with different receptive fields, we demonstrate substantial improvement in mAP for different objects in PASCAL VOC and MS COCO datasets while reducing the bounding box requirement by 8 times, thus improving inference time by 10%. As a future work, our approach can be used to improve region proposal based detection techniques as well. We also believe that our work can inspire several extensions to find more effective and efficient ways to combine feature maps of convolution and deconvolution blocks to improve image classification, object detection and semantic segmentation approaches.

References

1. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. CVPR (2016)
2. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: ECCV (2016)
3. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. CoRR abs/1605.06409 (2016)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (Jun 2010)
5. Fu, C., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD : Deconvolutional single shot detector. CoRR abs/1701.06659 (2017)
6. Girshick, R.: Fast r-cnn. In: ICCV (2015)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR abs/1406.4729 (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015)
10. Jeong, J., Park, H., Kwak, N.: Enhancement of SSD by concatenating feature maps for object detection. CoRR abs/1705.09587 (2017)
11. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
12. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR abs/1405.0312 (2014)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV (1). LNCS, vol. 9905 (2016)
14. Luo, W., Li, Y., Urtasun, R., Zemel, R.S.: Understanding the effective receptive field in deep convolutional neural networks. CoRR abs/1701.04128 (2017)
15. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. CoRR abs/1505.04366 (2015)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
17. Ren, J.S.J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.W., Xu, L.: Accurate single stage detector using recurrent rolling convolution. In: CVPR (2017)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS
19. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Lecun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR (2014)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
21. Woo, S., Hwang, S., Kweon, I.S.: Stairnet: Top-down semantic aggregation for accurate one shot detection. CoRR abs/1709.05788 (2017)
22. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. CoRR abs/1511.07122 (2015)
23. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. CoRR abs/1412.6856 (2014)