# An enhanced approach for Single Shot MultiBox Detector 7 layers (SSD7) using ResNet

Sherif Tohamy

***Abstract*— In this report, We present an enhancement to SSD7 (Single Shot MultiBox Detector) 7 layers version for detecting objects in images using a single deep neural network. SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. 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 network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. We consider the ResNet approach to enhance the performance of SSD7 by combining layers from 2 different architectures in order to get the best performance. As a result we achieved a good detection results with 20%mAP compared with 15%mAP for SSD7 by replacing the the first 3 layers of SSD7 with first 2 stages of ResNet50 architecture although having small dataset with only (22000) samples.**

## I. INTRODUCTION

Object detection is a computer vision (CV) task that involves both main tasks: localizing one or more objects within an image and classifying each object in the image. This is done by drawing a bounding box around the identified object with its predicted class. This means the system does not just predict the class of the image, as in image classification tasks; it also predicts the coordinates of the bounding box that fits the detected object. This is a challenging CV task because it requires both successful object localization, in order to locate and draw a bounding box around each object in an image, and object classification to predict the correct class of object that was localized.

Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run

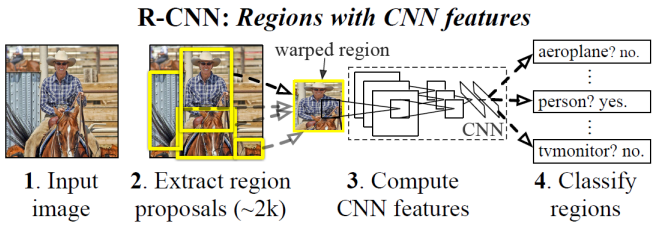



Fig. 1. RCNN object detection system overview. RCNN system: (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs.

at evenly spaced locations over the entire image [1]. More recent approaches like R-CNN [2] use region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene. Figure 1 presents an overview of RCNN method. They propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012—achieving a mAP of 53.3% .However, RCNN is very slow due to the selective search algorithm which proposes about 2,000 RoIs to be examined by the entire pipeline. And also the training is expensive in terms of space and time.

Fast R-CNN was an immediate descendant of R-CNN, developed in 2015 by Ross Girshick [3]. Fast R-CNN resembled the R-CNN technique in many ways but improved on its detection speed while also increasing detection accuracy through two main changes: 1) proposes that we apply the CNN feature extractor first to the entire input image and then propose regions. This way, we run

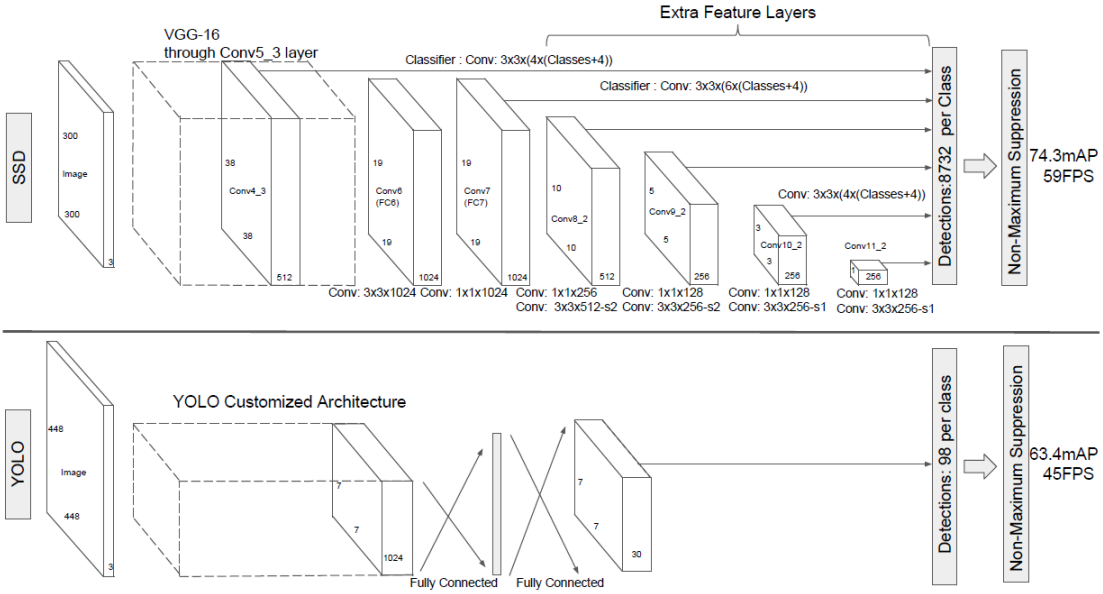



Fig. 2. A comparison between two single shot detection models: SSD and YOLO. SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences. SSD with a 300 x 300 input size significantly outperforms its 448 x 448 YOLO counterpart in accuracy on VOC2007 test while also improving the speed.

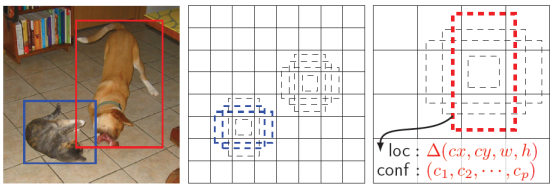
only one ConvNet over the entire image instead of 2,000 ConvNets over 2,000 overlapping regions. 2) It extends the ConvNet's job to do the classification part as well, by replacing the traditional SVM machine learning algorithm with a softmax layer. This way, they have a single model to perform both tasks: feature extraction and object classification. Fast R-CNN trains the very deep VGG16 network 9x faster than R-CNN, 213x faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. However, there is a big bottleneck remaining: the selective search algorithm for generating region proposals is very slow and is generated separately by another model.

Faster R-CNN is the third iteration of the R-CNN family, developed in 2016 by Shaoqing Ren et al [4]. Similar to Fast R-CNN, the image is provided as an input to a convolutional network that provides a convolutional feature map. Instead of using a selective search algorithm on the feature map to identify the region proposals, a region proposal network (RPN) is used to predict the region proposals as part of the training process. The predicted region proposals are then reshaped using an Region of Interest (RoI) pooling layer which is used to classify the image within the proposed region and predict the offset values for the bounding boxes. These improvements reduce the number of region proposals and accelerate the test-time operation of the model to near real-time with then state-of-the-art performance.

In the last few years, new architectures have been created to address the bottlenecks of R-CNN and its successors, enabling real-time object detection. The most famous are the single-shot detector (SSD) [5] and you only look once (YOLO) [6].

Similar to the R-CNN family, The YOLO family is a series of end-to-end DL models designed for fast object detection, and it was among the first attempts to build a fast real-time object detector. It is one of the fastest object detection algorithms out there. Although the accuracy of the models was good but it is still not as good as R-CNNs, they are popular for object detection because of their detection speed, often demonstrated in real-time video or camera feed input.

On the other hand, the SSD network reached new records in terms of performance and precision for object detection tasks, scoring over 74% mAP at 59 FPS on standard datasets such as the




(a) Image with GT boxes (b) $8 \times 8$ feature map (c) $4 \times 4$ feature map

Fig. 3. **SSD framework.** (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8 x 8 and 4 x 4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ((c1, c2, ... , cp)).

PASCAL VOC and Microsoft COCO. A standard pretrained network used for high-quality image classification, which is truncated before any classification layers. In their paper, Liu et al. used a VGG16 [7] network. Other networks like VGG19 and ResNet can be used and should produce good results as we have done in this report by replacing some of the VGG layers with ResNet stages to increase the mAP. We will introduce the SSD approach and model in more details.

### A. SSD Model

SSD framework is based on a feed-forward convolutional network that produces 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 and the SSD framework is illustrated in figure 3. The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers), which is the the base network for SSD. Then an auxiliary structure has been added to the network to produce detections. A comparison between two single shot detection models: SSD and YOLO is descried in figure 2 as stated in their paper.

In this report we consider a ResNet-based approach to enhance the performance of SSD7. Here we will combine layers from 2 different architectures in order to get the best performance although having small dataset with only (22000) samples.

## II. Methods

### A. Dataset

A toy dataset created by Udacity. It has more than 22,000 labeled images and 5 object classes: car, truck, pedestrian, bicyclist, and traffic light. All of the images have been resized to a height of 300 pixels and a width of 480 pixels. And What makes this dataset very interesting is that these are real-time images taken while driving in Mountain View, California, and neighboring cities during daylight conditions.

### B. ResNet50

The Residual Neural Network (ResNet) was developed in 2015 by a group from the Microsoft research team [8]. They introduced a novel residual module architecture with skip connections. The network features heavy batch normalization for the hidden layers. This technique allowed the team to train very deep neural networks with 50, 101, and 152 weight layers while still having lower complexity than smaller networks like VGGNet (19 layers). ResNet was able to achieve a top-5 error rate of 3.57% in the ILSVRC 2015 competition, which beat the performance of all prior ConvNets. To solve the vanishing gradient problem, the authors of ResNet created a shortcut that allows the gradient to be directly back propagated to earlier layers. These shortcuts are called skip connections: they are used to flow information from earlier layers in the network to later layers, creating an alternate shortcut path for the gradient to flow through. Another important benefit of the skip connections is that they allow the model to learn an identity function, which ensures that the layer will perform at least as well as the previous layer. This combination of the skip connection and convolutional layers is called a residual block. ResNet is composed of a series of these residual block building blocks that are stacked on top of each other (figure 4). We have implemented 2 layres of ResNet50: a version of the ResNet architecture that contains 50 weight layers (hence the name). Advanced CNN architectures with 18, 34, 101, and 152 layers by following are illustrated in (figure 5) from the original paper.

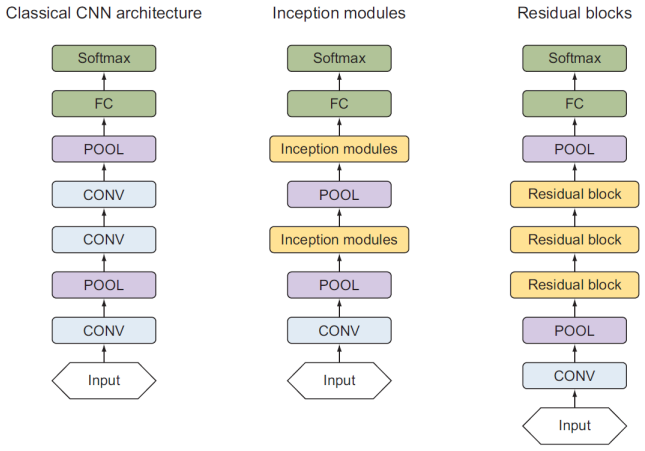



Fig. 4. Classical CNN architecture (left). The Inception network consists of a set of inception modules (middle). The residual network consists of a set of residual blocks (right).

### C. Model

For this project, we are going to build a smaller SSD network called SSD7. SSD7 is a seven-layer version of the SSD300 network. It is important to note that while an SSD7 network would yield some acceptable results, this is not an optimized network architecture. The goal is just to build and enhance a low-complexity network that is fast enough for us to train. and for second version we used a combination between ssd7 and ResNet for the second model's architecture by using the first 2 stages from ResNet50 architecture combined with the last 4 prediction layers of ssd7 benefiting from the fundamental breakthrough with ResNet was that it allowed us to train extremely deep neural networks with hundreds of layers with much less time.

### D. Training

The key difference between training SSD and training a typical detector that uses region proposals, is that ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Some version of this is also required for training in YOLO and for the region proposal stage of Faster R-CNN. 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.

During training we need to determine which default boxes correspond to a ground truth detection and train the network accordingly. For each ground truth box we are selecting from default boxes that vary over location, aspect ratio, and scale. We begin by matching each ground truth box to the default box with the best jaccard overlap (as in MultiBox [9]). 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.

SSD use both the lower and upper feature maps for detection. Figure 3 shows two exemplar feature maps (8x8 and 4x4) which are used in the framework. In practice, we can use many more with small computational overhead. Feature maps from different levels within a network are known to have different.

## III. Experimental Results

Training SSD7 on the aforementioned dataset yields alright results, but to be mentioned, SSD7 and the enhanced version are not carefully optimized network architectures. We used 20 epochs with 1000 steps per each epoch with only 18000 images to train the model. SSD7 validation-loss graph is shown in figure 6 with 2.18 loss and 1.53 validation loss and with 15% mAP. For the second version, it has achieved better results with 1.65 loss and 0.94 validation loss with 20.2% mAP. In addition, although the second version has x3 times more parameters than the ss7 version, it consumed approximately the same training time over the 20 epochs. Results on dataset images for the two versions are represented in figure 8 and figure 9.

Moreover, by using more deep CNNs (ResNet50), we have noticed that the deeper the network, the larger its learning capacity, and the better it extracts features from images. This mainly happens because very deep networks are able to represent very complex functions, which allows the network to learn features at many different levels of abstraction, from edges (at the lower layers) to very complex features (at the deeper layers).

| Layer name | Output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112x112 | 7x7, 64, stride 2 | | | | |
| conv2_x | 56x56 | 3x3, maxpool, stride 2 | | | | |
| | | [3x3, 64; 3x3, 64] x2 | [3x3, 64; 3x3, 64] x3 | [1x1, 64; 3x3, 64; 1x1, 256] x3 | [1x1, 64; 3x3, 64; 1x1, 256] x3 | [1x1, 64; 3x3, 64; 1x1, 256] x3 |
| conv3_x | 28x28 | [3x3, 128; 3x3, 128] x2 | [3x3, 128; 3x3, 128] x4 | [1x1, 128; 3x3, 128; 1x1, 512] x3 | [1x1, 128; 3x3, 128; 1x1, 512] x4 | [1x1, 128; 3x3, 128; 1x1, 512] x8 |
| conv4_x | 14x14 | [3x3, 256; 3x3, 256] x2 | [3x3, 256; 3x3, 256] x6 | [1x1, 256; 3x3, 256; 1x1, 1024] x3 | [1x1, 256; 3x3, 256; 1x1, 1024] x23 | [1x1, 256; 3x3, 256; 1x1, 1024] x36 |
| conv5_x | 7x7 | [3x3, 512; 3x3, 512] x2 | [3x3, 512; 3x3, 512] x3 | [1x1, 512; 3x3, 512; 1x1, 2048] x3 | [1x1, 512; 3x3, 512; 1x1, 2048] x3 | [1x1, 512; 3x3, 512; 1x1, 2048] x3 |
| | 1x1 | Average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8 \times 10^9$ | $3.6 \times 10^9$ | $3.8 \times 10^9$ | $7.6 \times 10^9$ | $11.3 \times 10^9$ |

Fig. 5. Architecture of several ResNet variations from the original paper.

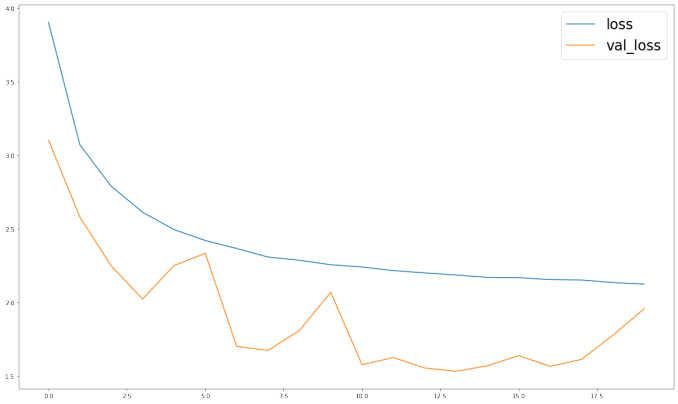



Fig. 6. Validation-loss graph for SSD7

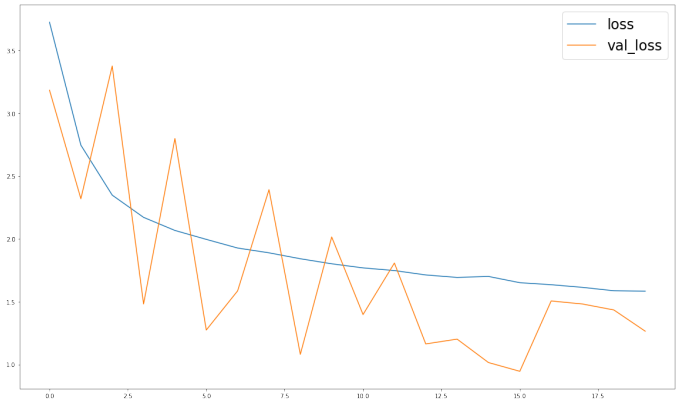



Fig. 7. Validation-loss graph for second version

## IV. Discussion and Conclusion

In this report we proposed a ResNet-based deep learning approach to enhance the perfomance of the proposed method SSD. we combined two different convolutional neural networks (CNNs) architectures in order to get the best performance although having small dataset with only (22000) samples. As a result we achieved a good classification accuracy by replacing the the first 4 layers with the first 2 stages of ResNet50 architecture. We have proved that the deeper the network, the larger its learning capacity, and the better it extracts features from images by showing how our second version with ResNet50 outperformed the first version with VGG-based layers.

## References

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010. 1, 4

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition


Fig. 8. SSD7: Object detection results

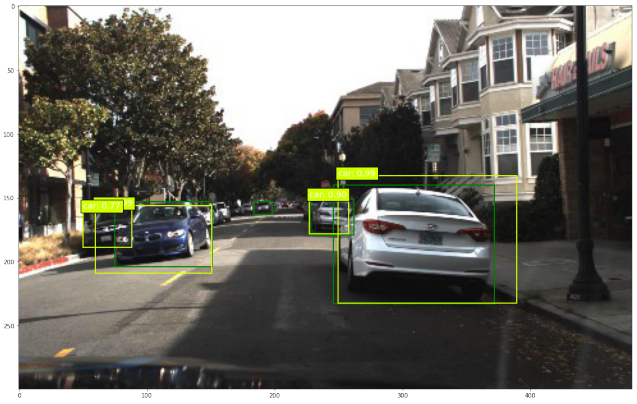



Fig. 9. Enhanced Version: Object detection results

(CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014. 1, 4, 7

[3] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015. 2, 5, 6, 7

[4] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 5, 6, 7

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: Single Shot MultiBox Detector," 2016, http://arxiv.org/abs/1512.02325.

[6] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in International Conference on Learning Representations (ICLR), 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," 2015, http://arxiv.org/abs/1512.03385.

[9] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)