Paper Review: SSD - Single Shot MultiBox Detector

A Revolutionary Approach to Real-Time Object Detection

Abstract

This paper review examines the SSD (Single Shot MultiBox Detector) architecture, a groundbreaking approach to real-time object detection proposed by Liu et al. SSD represents a paradigm shift in object detection by eliminating the need for region proposals and feature resampling, achieving both high accuracy and real-time performance. The method discretizes the output space using default boxes across multiple feature maps at different scales and aspect ratios. SSD300 achieves 74.3% mAP on VOC2007 at 59 FPS, significantly outperforming previous single-shot detectors like YOLO while matching the accuracy of slower proposal-based methods like Faster R-CNN. This review discusses the abstract, architecture, and key discriminating features that set SSD apart from older object detection models.

1 Introduction

Prior to SSD, state-of-the-art object detection systems followed a two-stage approach: hypothesize bounding boxes through region proposals (e.g., Selective Search, RPN), then resample pixels or features for classification. While methods like Faster R-CNN achieved high accuracy, they were computationally intensive, operating at only 7 FPS. SSD addresses this limitation by introducing a single-shot detection framework that achieves comparable or better accuracy while operating at real-time speeds.

2 Abstract Overview

The SSD paper presents a unified framework for detecting objects using a single deep neural network. The key contributions highlighted in the abstract include:

- **Single-Shot Architecture**: Complete elimination of proposal generation and subsequent resampling stages
- **Multi-Scale Detection**: Predictions from multiple feature maps with different resolutions to handle various object sizes
- **Default Boxes**: Discretization of bounding box space using default boxes with different aspect ratios and scales
- **Performance**: SSD300 achieves 74.3% mAP at 59 FPS on VOC2007, and SSD512 achieves 76.9% mAP, outperforming Faster R-CNN

The abstract emphasizes that SSD is *simple relative to methods that require object proposals* and provides a *unified framework for both training and inference*, making it easy to train and integrate into larger systems.

3 SSD Architecture

3.1 Overall Framework

The SSD architecture builds upon a base network (VGG-16) truncated before classification layers and adds auxiliary convolutional layers that progressively decrease in size. The key architectural components are illustrated in Figure 1.

SSD: Single Shot MultiBox Detector

Wei Liu[1], Dragomir Anguelov[2], Dumitru Erhan[3], Christian Szegedy[3],
Scott Reed[4], Cheng-Yang Fu[1], Alexander C. Berg[1]

[1]UNC Chapel Hill [2]Zoox Inc. [3]Google Inc. [4]University of Michigan, Ann-Arbor
[1]wliu@cs.unc.edu, [2]drago@zoox.com, [3]{dumitru,szegedy}@google.com,
[4]reedscot@umich.edu, [1]{cyfu,aberg}@cs.unc.edu

Abstract. We present a method for detecting objects in images using a single deep neural network. 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 network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, COCO, and ILSVRC datasets confirm that SSD has competitive accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. For 300 × 300 input, SSD achieves 74.3% mAP[1] on VOC2007 `test` at 59 FPS on a Nvidia Titan X and for 512 × 512 input, SSD achieves 76.9% mAP, outperforming a comparable state-of-the-art Faster R-CNN model. Compared to other single stage methods, SSD has much better accuracy even with a smaller input image size. Code is available at: `https://github.com/weiliu89/caffe/tree/ssd` .

Keywords: Real-time Object Detection; Convolutional Neural Network

1 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. This pipeline has prevailed on detection benchmarks since the Selective Search work [1] through the current leading results on PASCAL VOC, COCO, and ILSVRC detection all based on Faster R-CNN[2] albeit with deeper features such as [3]. While accurate, these approaches have been too computationally intensive for embedded systems and, even with high-end hardware, too slow for real-time applications.

[1] We achieved even better results using an improved data augmentation scheme in follow-on experiments: 77.2% mAP for 300×300 input and 79.8% mAP for 512×512 input on VOC2007. Please see Sec. 3.6 for details.

Figure 1: SSD Framework: (a) Input image with ground truth boxes, (b) 8×8 feature map, (c) 4×4 feature map showing default boxes and predictions for location offsets and class confidences.

3.2 Base Network

SSD uses VGG-16 pre-trained on ImageNet as the base network with the following modifications:

- Converts fc6 and fc7 to convolutional layers
- Changes pool5 from $2 \times 2 - s2$ to $3 \times 3 - s1$
- Uses atrous (à trous) algorithm to maintain resolution
- Removes dropout layers and fc8

3.3 Multi-Scale Feature Maps

SSD adds several convolutional feature layers after the base network:

- **Conv4_3**: 38×38 (with L2 normalization)
- **Conv7 (FC7)**: 19×19
- **Conv8_2**: 10×10
- **Conv9_2**: 5×5
- **Conv10_2**: 3×3
- **Conv11_2**: 1×1

Each layer produces predictions at different scales, enabling the network to detect objects of various sizes naturally.

3.4 Default Boxes and Predictions

At each location in each feature map, SSD predicts:

1. **Class scores**: Confidence for each object category
2. **Bounding box offsets**: Adjustments to default box coordinates

For a feature map of size $m \times n$ with p channels and k default boxes per location, SSD uses $3 \times 3 \times p$ kernels to produce $(c+4)k$ outputs per location, where c is the number of classes and 4 represents the box offset parameters.

3.5 Scale and Aspect Ratio Calculation

Default box scales are computed linearly across layers:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \quad k \in [1, m] \tag{1}$$

where $s_{\min} = 0.2$, $s_{\max} = 0.9$, and m is the number of feature maps.

Aspect ratios: $a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$

Box dimensions:

$$w_k^a = s_k \sqrt{a_r} \tag{2}$$
$$h_k^a = s_k / \sqrt{a_r} \tag{3}$$

For aspect ratio 1, an additional box with scale $s'_k = \sqrt{s_k s_{k+1}}$ is added, resulting in 6 default boxes per location.

4 Training Methodology

4.1 Matching Strategy

SSD uses a sophisticated matching strategy:

1. Match each ground truth box to the default box with best Jaccard overlap
2. Match default boxes to any ground truth with Jaccard overlap > 0.5

This allows multiple overlapping default boxes to predict the same object, simplifying learning.

4.2 Loss Function

The multi-task loss combines localization and confidence losses:

$$L(x, c, l, g) = \frac{1}{N}(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \tag{4}$$

where N is the number of matched default boxes.

Localization Loss (Smooth L1):

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smoothL1}(l_i^m - \hat{g}_j^m) \tag{5}$$

with offsets:

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \tag{6}$$
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \tag{7}$$

Confidence Loss (Softmax):

$$L_{\text{conf}}(x, c) = -\sum_{i \in \text{Pos}} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in \text{Neg}} \log(\hat{c}_i^0) \tag{8}$$

where $\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$

The weight α is set to 1 by cross-validation.

4.3 Hard Negative Mining

To address class imbalance, SSD:

- Sorts negative examples by confidence loss
- Selects top negatives maintaining a 3:1 negative-to-positive ratio
- This leads to faster optimization and more stable training

4.4 Data Augmentation

Critical for performance, SSD employs extensive data augmentation:

- Use entire original image
- Sample patches with minimum Jaccard overlap: 0.1, 0.3, 0.5, 0.7, 0.9
- Random patches
- Patch sizes: [0.1, 1] of original
- Aspect ratios: $[\frac{1}{2}, 2]$
- Horizontal flip with probability 0.5
- Photo-metric distortions
- **Zoom out**: Place image on 16× canvas (improves small object detection by 2-3% mAP)

5 Key Discriminating Features vs. Older Models

5.1 Single-Shot Detection (vs. Two-Stage Methods)

Older Approach (Faster R-CNN):

- Stage 1: Region Proposal Network generates proposals
- Stage 2: Fast R-CNN classifies each proposal
- Requires feature resampling (ROI pooling)
- Speed: 7 FPS

SSD Innovation:

- Single forward pass for detection
- No region proposals, no feature resampling
- All computation in one network
- Speed: 59 FPS (SSD300), 22 FPS (SSD512)
- Accuracy: Comparable or better than Faster R-CNN

5.2 Multi-Scale Feature Maps (vs. Single-Scale Detection)

Older Approach (YOLO):

- Uses only the topmost feature map (7×7)
- 98 predictions per image
- Poor performance on small objects
- mAP: 63.4%

SSD Innovation:

- Uses 6 feature maps at different scales (38×38 to 1×1)
- 8,732 predictions for SSD300, 24,564 for SSD512
- Naturally handles objects of various sizes
- mAP: 74.3% (SSD300), 76.8% (SSD512)

Table 1: Performance Comparison on VOC2007 Test

Method	**mAP (%)**	**FPS**	**Input Size**	**# Boxes**
Fast R-CNN	70.0	-	∼1000×600	∼2000
Faster R-CNN	73.2	7	∼1000×600	∼6000
YOLO	63.4	45	448×448	98
SSD300	**74.3**	**59**	300×300	8,732
SSD512	**76.8**	**22**	512×512	24,564

5.3 Convolutional Predictors (vs. Fully Connected)

Older Approach (YOLO):

- Uses fully connected layers for prediction
- Fixed spatial relationships
- Less efficient parameter usage

SSD Innovation:

- Uses 3×3 convolutional filters for prediction
- Convolutional manner preserves spatial information
- More efficient and flexible
- Different predictors for different feature layers

5.4 Default Boxes with Multiple Aspect Ratios

Similar to Faster R-CNN Anchors but Enhanced:

- Faster R-CNN: Anchors only at RPN stage, single feature map
- SSD: Default boxes at *multiple* feature maps with different scales
- Varying aspect ratios: {1, 2, 3, 1/2, 1/3}
- Efficiently discretizes the space of possible box shapes

5.5 End-to-End Training

Older Approach (Faster R-CNN):

- Alternating training between RPN and Fast R-CNN
- Complex training procedure
- Two dependent networks

SSD Innovation:

- Simple end-to-end training
- Single network optimization
- Unified loss function
- Easier to train and deploy

6 Experimental Results

6.1 PASCAL VOC Performance

Table 2: PASCAL VOC2007 Test Results (07+12+COCO Training)

Method	**Data**	**mAP**	**aero**	**bike**	**bird**	**boat**
Faster R-CNN	07+12+COCO	78.8	84.3	82.0	77.7	68.9
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	76.2
SSD512	07+12+COCO	**81.6**	**86.6**	88.3	**82.4**	76.0

6.2 COCO Dataset Results

Table 3: COCO test-dev2015 Detection Results

Method	**mAP@0.5:0.95**	**mAP@0.5**	**mAP@0.75**	**AP**$_S$	**AP**$_M$	**AP**$_L$
Fast R-CNN	19.7	35.9	-	-	-	-
Faster R-CNN	24.2	45.3	23.5	7.7	26.4	37.1
SSD300	23.2	41.2	23.4	5.3	23.2	39.6
SSD512	**26.8**	**46.5**	**27.8**	9.0	28.9	**41.9**

SSD512 shows 5.3% improvement in mAP@0.75 and 4.8% improvement in AP for large objects compared to Faster R-CNN.

6.3 Model Analysis

Table 4: Ablation Study: Effects of Design Choices on SSD300

Configuration	**VOC2007 test mAP**
Base (no augmentation, no $\{1/2, 2, 1/3, 3\}$ boxes)	65.5
+ Data augmentation	71.6
+ $\{1/2, 2\}$ aspect ratio boxes	73.7
+ $\{1/3, 3\}$ aspect ratio boxes	74.2
Full SSD300 (+ atrous)	**74.3**

Key findings:

- **Data augmentation**: +8.8% mAP improvement
- **Multiple aspect ratios**: +2.6% mAP improvement
- **Atrous convolution**: Maintains accuracy with 20% speed increase

6.4 Sensitivity Analysis

SSD shows different performance characteristics across object sizes:

- **Small objects**: Weaker performance (improved with larger input size)
- **Large objects**: Excellent performance, very robust
- **Various aspect ratios**: Robust handling due to multiple default box shapes
- **Localization**: Better than R-CNN (less localization error)
- **Similar categories**: More confusion (shares locations for multiple categories)

7 Architectural Innovations Summary

1. **Elimination of Region Proposals**
   - Direct prediction eliminates proposal generation overhead
   - No ROI pooling or feature resampling required
   - Fundamental speed improvement
2. **Multi-Scale Detection Framework**
   - Predictions from 6 different feature map resolutions
   - Lower layers: fine details for small objects

- Upper layers: semantic information for large objects
- Shares parameters across scales

3. **Default Box Mechanism**
   - Fixed set of default boxes per location
   - Multiple scales and aspect ratios
   - Efficiently discretizes output space
   - Similar to anchors but applied at multiple scales
4. **Convolutional Prediction Architecture**
   - 3×3 kernels for class and location prediction
   - Different predictors for each feature layer
   - Preserves spatial information
   - Efficient parameter usage
5. **Unified End-to-End Training**
   - Single network, single loss function
   - No alternating training procedures
   - Straightforward optimization
   - Easy integration into larger systems

8 Comparison with Contemporary Methods

8.1 vs. Faster R-CNN

Table 5: SSD vs. Faster R-CNN Comparison

Characteristic	**Faster R-CNN**	**SSD**
Architecture	Two-stage	Single-stage
Region Proposals	Yes (RPN)	No
Feature Resampling	Yes (ROI pooling)	No
Feature Maps Used	Single (for detection)	Multiple (6 layers)
Training	Alternating	End-to-end
Speed (FPS)	7	59 (SSD300), 22 (SSD512)
Accuracy (VOC2007)	73.2%	74.3% (SSD300), 76.8% (SSD512)

8.2 vs. YOLO

Table 6: SSD vs. YOLO Comparison

Characteristic	**YOLO**	**SSD**
Feature Maps Used	1 (7×7)	6 (38×38 to 1×1)
Predictions	98	8,732 (SSD300)
Prediction Method	FC layers	Convolutional filters
Default Boxes	Grid cells	Multi-scale, multi-aspect
Accuracy (VOC2007)	63.4%	74.3%
Speed (FPS)	45	59
Small Object Detection	Poor	Better

9 Limitations and Future Directions

Despite its innovations, SSD has some limitations:

1. **Small Object Performance**: While better than YOLO, still lags behind Faster R-CNN on very small objects
2. **Similar Category Confusion**: Shares locations for multiple categories, leading to confusion
3. **Default Box Design**: Optimal tiling strategy remains an open question

Proposed improvements:

- Better default box alignment with receptive fields
- Enhanced data augmentation (zoom out improves small object detection by 2-3%)
- Use of faster base networks (ResNet, MobileNet)
- Integration with recurrent networks for video detection

10 Conclusion

SSD represents a significant advancement in object detection by successfully combining high accuracy with real-time performance. The key innovations that distinguish it from older models are:

1. **Single-shot architecture** eliminating proposal generation
2. **Multi-scale feature map predictions** for handling various object sizes
3. **Default boxes with multiple aspect ratios** at each feature map location
4. **Convolutional predictors** for efficient parameter usage
5. **End-to-end training** with unified loss function

These innovations enable SSD to achieve 74.3% mAP at 59 FPS (SSD300) and 76.8% mAP at 22 FPS (SSD512) on PASCAL VOC2007, outperforming both Faster R-CNN in speed and YOLO in accuracy. SSD demonstrates that carefully designed single-shot detectors can match or exceed the accuracy of slower two-stage methods while maintaining real-time performance, making it a foundational work for modern object detection systems.

The paper's extensive ablation studies and analysis provide valuable insights into what makes object detection systems effective, particularly highlighting the importance of multi-scale predictions, diverse default boxes, and aggressive data augmentation. SSD has influenced numerous subsequent architectures and remains a benchmark for evaluating new object detection methods.