SSD: Single Shot MultiBox Detector

Author: Wei Liu et al.
Presenter: Siyu Jiang
Outline

1. Motivations
2. Contributions
3. Methodology
4. Experiments
5. Conclusions
6. Extensions
Motivation
Motivation

Faster R-CNN: Box Classification and Regression are being done 2 times.
Motivation

1. Two stage object detection is time-consuming. Faster R-CNN is faster but not fast enough.
Motivation

YOLO: Fast but not accurate enough.
Motivation

1. Two stage object detection is time-consuming. Faster R-CNN is faster but not fast enough.
2. YOLO: Fast but not accurate enough.
3. Object detection needs a good tradeoff between accuracy and speed.
Contribution

SSD: Single Shot MultiBox Detection (ECCV 2016)
Contribution

1. Achieves competitive mAP (72.1) as faster R-CNN (73.2).
2. Much faster (58 fps) than faster R-CNN (7fps) and YOLO (45fps), making accurate real-time detection possible.
3. Makes predictions on multiple feature maps with different resolutions to handle objects of different sizes.
Network Architecture

VGG network - extract features

Detectors on multiple feature maps
Default (Anchor) Boxes

- Similar to faster R-CNN, SSD also uses anchor boxes.
- At each feature map position, anchor boxes have different aspect ratios.

Feature Map

Each position has anchor boxes of different aspect ratio

$$a_r : \text{Aspect ratio}$$

$$a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$$
Default (Anchor) Boxes

- Anchor boxes of different feature maps have unique scales. So different feature maps are responsible for objects of different sizes.

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

- \( S_k \) denotes the scale of the \( k \)-th feature map. \( m \) is the number of feature maps for prediction. \( S_{\min} = 0.2, S_{\max} = 0.9 \).
Convolutional Filters for Prediction

- On each feature map, two types of convolutional filters will be applied:
  - C filters for category prediction, where C is the number of object categories.
  - 4 filters for bounding box regression, 4 for the coordinates x, y, w, h.

- Together there will be \((c + 4)k\) filters for each feature map, \(k\) is the number of anchor box types.

- The output for a \(m \times n\) feature map will be a map of \(m \times n \times (c + 4)k\), indicating the category and coordinates for each bounding box.
Training objective

- The loss used in SSD is a combination of confidence loss and localization loss.
  \[
  L(x, c, l, g) = \frac{1}{N} \left( L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right)
  \]

- Confidence loss is the softmax loss over multiple classes confidences.
  \[
  L_{conf}(x, c) = -\sum_{i \in Pos} x_i^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0)
  \]

- Localization loss is the same smooth L1 loss as faster R-CNN.
  \[
  L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_L(l_i^m - \hat{g}_j^m)
  \]
  \[
  \hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h
  \]
  \[
  \hat{g}_j^w = \log \left( \frac{g_j^w}{d_i^w} \right) \quad \hat{g}_j^h = \log \left( \frac{g_j^h}{d_i^h} \right)
  \]
Matching Strategy

- During training, we need to determine which anchor boxes are corresponding to ground truth boxes.
  1. Match each ground truth box to the anchor box with the best jaccard overlap.
  2. Match default box to any ground truth with jaccard overlap higher than 0.5.

- Question: Why not using predicted box (anchor box after regression) for matching?

- Question: Negative anchor boxes are much more than positive anchor boxes, how to deal with this imbalance?
Hard Negative Mining

- Number of negative anchor boxes $\gg$ number of positive anchor boxes
- Calculate the confidence loss of each negative anchor boxes.
- Select anchor boxes that have highest loss as negative training samples.
- Keep the ratio between negatives and positives 3:1.
- This leads to faster convergence and stable training.

- Question: Is this approach sufficient for training negative samples?
  - No. A recent paper show that negative samples need more elegant loss calculation.
  - (ICCV 2017 best student paper award: *Focal Loss for Dense Object Detection*, Lin et al.)
Experiments

- Base network: VGG16, pretrained on ILSVRC dataset.
- Add layers on top of VGG Conv5 layer.
- Use dilated convolution in Conv6 layer.
- Dataset: Pascal VOC and MS COCO.
Results – using multiple layers for prediction

- Predicting on multiple layers is better.
- Removing boundary boxes hurts the accuracy for high level feature maps.

<table>
<thead>
<tr>
<th>Prediction source layers from:</th>
<th>mAP use boundary boxes?</th>
<th># Boxes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>conv4_3</td>
<td>74.3</td>
<td>63.4</td>
</tr>
<tr>
<td>conv7</td>
<td>74.6</td>
<td>63.1</td>
</tr>
<tr>
<td>conv8_2</td>
<td>73.8</td>
<td>68.4</td>
</tr>
<tr>
<td>conv9_2</td>
<td>70.7</td>
<td>69.2</td>
</tr>
<tr>
<td>conv10_2</td>
<td>64.2</td>
<td>64.4</td>
</tr>
<tr>
<td>conv11_2</td>
<td>62.4</td>
<td>64.0</td>
</tr>
</tbody>
</table>

Table 3: Effects of using multiple output layers.
Results – scale & data augmentation

- Using more scales of anchor boxes is better.
- Data Augmentation is crucial to SSD. Increases mAP by 8.8%.

<table>
<thead>
<tr>
<th>more data augmentation?</th>
<th>SSD300</th>
</tr>
</thead>
<tbody>
<tr>
<td>include ${\frac{1}{2}, 2}$ box?</td>
<td>✓</td>
</tr>
<tr>
<td>include ${\frac{1}{3}, 3}$ box?</td>
<td>✓</td>
</tr>
<tr>
<td>use atrous?</td>
<td>✓</td>
</tr>
</tbody>
</table>

| VOC2007 test mAP         | 65.5   | 71.6   | 73.7   | 74.2   | 74.3   |
Data Augmentation

- Photo-metric distortions.
- Random crop:
  - Use the entire original input image.
  - Crop a patch so that the minimum jaccard overlap with the objects is 0.1, 0.3, 0.5, 0.7 or 0.9.
  - Randomly crop a patch.
- Question: Will data augmentation also benefit this much to Faster R-CNN?
  - No, because faster R-CNN uses a feature pooling step which is relatively robust to object translation.
Results – small objects

- SSD does not perform well on small objects.
- No feature resampling step in SSD.
- Relatively low-level feature maps are responsible for detecting small objects. These feature maps do not have sufficient high-level semantic information.
More data augmentation for small objects

- Keep small objects small.

  - Random expand 4x width
  - Random expand 4x height
  - Fill with grey pixels

  Random sample

  Crop & resize
Results – after data augmentation for small objects

- An increase of 2% - 3% in mAP.
Results - comparison with other methods

- It is claimed in the paper that SSD outperforms Faster R-CNN and YOLO in both speed and accuracy.
- SSD300 is the first real-time method to achieve above 70% mAP.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>FPS</th>
<th>batch size</th>
<th># Boxes</th>
<th>Input resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN (VGG16)</td>
<td>73.2</td>
<td>7</td>
<td>1</td>
<td>~ 6000</td>
<td>~ 1000 × 600</td>
</tr>
<tr>
<td>Fast YOLO</td>
<td>52.7</td>
<td>155</td>
<td>1</td>
<td>98</td>
<td>448 × 448</td>
</tr>
<tr>
<td>YOLO (VGG16)</td>
<td>66.4</td>
<td>21</td>
<td>1</td>
<td>98</td>
<td>448 × 448</td>
</tr>
<tr>
<td>SSD300</td>
<td>74.3</td>
<td>46</td>
<td>1</td>
<td>8732</td>
<td>300 × 300</td>
</tr>
<tr>
<td>SSD512</td>
<td>76.8</td>
<td>19</td>
<td>1</td>
<td>24564</td>
<td>512 × 512</td>
</tr>
<tr>
<td>SSD300</td>
<td>74.3</td>
<td>59</td>
<td>8</td>
<td>8732</td>
<td>300 × 300</td>
</tr>
<tr>
<td>SSD512</td>
<td>76.8</td>
<td>22</td>
<td>8</td>
<td>24564</td>
<td>512 × 512</td>
</tr>
</tbody>
</table>
Results - comparison with other methods

- Outperforms Faster R-CNN in every category. VOC 2007, VOC 2012, MS COCO.
- Notice that the input image size of Faster R-CNN is at least 600 x 600.

| 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| 76.6  | 82.1| 77.0  | 72.5  | 71.2   | 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 |
Results - comparison with other methods

- Outperforms Faster R-CNN in every category. VOC 2007, VOC 2012, MS COCO.
- Notice that the input image size of Faster R-CNN is at least 600 x 600.

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

Table 5: COCO test-dev2015 detection results.
Are the results really this good?

- *Speed/accuracy trade-offs for modern convolutional object detectors (CVPR 2017)*
- A detailed comparison of SSD, Faster R-CNN, and R-FCN.

- SSD is not as good as faster R-CNN or R-FCN in most cases.
- Better in large objects, worse in small ones.

*Figure 4: Accuracy stratified by object size, meta-architecture and feature extractor. We fix the image resolution to 300.*
Results - comparison with other methods

- SSD leads in detection speed; it is good at speed / accuracy tradeoff.
Conclusion

- Introduces a single-stage detector for object detection.
- Uses convolutional predictor on multiple feature maps, each responsible for a unique scale of objects.
- Achieves competitive accuracy and faster speed on various datasets.
Extensions?

- Use skip connections and deconvolution to integrate low-level location information and high-level semantic information.

Feature Pyramid Networks for Object Detection (CVPR 2017)

DSSD: Deconvolutional Single Shot Detector (2017)
References

- https://www.youtube.com/watch?v=P8e-G-Mhx4k