Beyond RetinaNet and Mask R-CNN

Gang Yu
yugang@megvii.com
Outline

• Modern Object detectors
  • One Stage detector vs Two-stage detector
• Challenges
  • Backbone
  • Head
  • Scale
  • Batch Size
  • Crowd
• Conclusion
Modern Object detectors

- Modern object detectors
  - RetinaNet
    - f1-f7 for backbone, f3-f7 with 4 convs for head
  - FPN with ROIAlign
    - f1-f6 for backbone, two fc's for head
  - Recall vs localization
    - One stage detector: Recall is high but compromising the localization ability
    - Two stage detector: Strong localization ability
One Stage detector: RetinaNet

- FPN Structure
- Focal loss

Focal Loss for Dense Object Detection, Lin et al., ICCV 2017 Best student paper
One Stage detector: RetinaNet

- FPN Structure
- Focal loss

Focal Loss for Dense Object Detection, Lin et al, ICCV 2017 Best student paper
Two-Stage detector: FPN/Mask R-CNN

- FPN Structure
- ROIAlign

Figure 1. The Mask R-CNN framework for instance segmentation.

Mask R-CNN, He etc, ICCV 2017 Best paper
What is next for object detection?

- The pipeline seems to be mature
- There still exists a large gap between existing state-of-arts and product requirements
- The devil is in the detail
Challenges Overview

- Backbone
- Head
- Scale
- Batch Size
- Crowd
Challenges - Backbone

• Backbone network is designed for classification task but not for localization task
  • Receptive Field vs Spatial resolution
• Only f1-f5 is pretrained but randomly initializing f6 and f7 (if applicable)
Backbone - DetNet

Backbone - DetNet

A: Dilated bottleNeck

1x1 Conv

3x3, dilate 2, Conv

1x1 Conv

Add

ReLU

B: Dilated bottleNeck with 1x1 conv projection

1x1 Conv

3x3, dilate 2, Conv

1x1 Conv

Add

ReLU

C: Original bottleNeck

1x1 Conv

3x3, Conv

1x1 Conv

Add

ReLU
Backbone - DetNet

D: DetNet Backbone

Stage 4 output 16x stride

Stage 5 output 16x stride

Stage 6 output 16x stride

E: Feature Pyramid Structure

Fully Connect, 1000
## Backbone - DetNet

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Classification Err</th>
<th>FLOPs</th>
<th>FPN on COCO minimal mAP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_{S}$</th>
<th>AP$_{M}$</th>
<th>AP$_{L}$</th>
<th>FPN on COCO test-dev mAP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_{S}$</th>
<th>AP$_{M}$</th>
<th>AP$_{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D-59</strong></td>
<td>23.5</td>
<td>4.8G</td>
<td>40.2</td>
<td>61.7</td>
<td>43.7</td>
<td>23.9</td>
<td>43.2</td>
<td>52.0</td>
<td>40.3</td>
<td>62.1</td>
<td>43.8</td>
<td>23.6</td>
<td>42.6</td>
<td>50.0</td>
</tr>
<tr>
<td>R-62</td>
<td>23.4</td>
<td>4.7G</td>
<td>38.8</td>
<td>60.6</td>
<td>42.4</td>
<td>22.6</td>
<td>41.6</td>
<td>51.6</td>
<td>39.0</td>
<td>61.0</td>
<td>42.3</td>
<td>21.9</td>
<td>41.2</td>
<td>49.7</td>
</tr>
<tr>
<td>R-50</td>
<td>24.1</td>
<td>3.8G</td>
<td>37.9</td>
<td>60.0</td>
<td>41.2</td>
<td>22.9</td>
<td>40.6</td>
<td>49.2</td>
<td>38.4</td>
<td>60.4</td>
<td>41.6</td>
<td>22.5</td>
<td>40.7</td>
<td>47.9</td>
</tr>
<tr>
<td><strong>D-101</strong></td>
<td>23.0</td>
<td>7.9G</td>
<td>41.9</td>
<td>62.8</td>
<td>45.7</td>
<td>25.4</td>
<td>45.2</td>
<td>55.1</td>
<td>42.2</td>
<td>63.2</td>
<td>45.8</td>
<td>24.5</td>
<td>44.8</td>
<td>53.1</td>
</tr>
<tr>
<td>R-101</td>
<td>22.9</td>
<td>7.8G</td>
<td>39.9</td>
<td>62.0</td>
<td>43.7</td>
<td>24.1</td>
<td>43.4</td>
<td>52.0</td>
<td>40.3</td>
<td>62.5</td>
<td>44.0</td>
<td>23.3</td>
<td>43.1</td>
<td>50.6</td>
</tr>
</tbody>
</table>

**Table 1.** Comparison of ‘D’ DetNet and ‘R’ ResNet. We report both results on ImageNet classification (Top1 Error) and FPN COCO detection. Results validate that DetNet is more suitable for object detection. Keeping same model size, DetNet consistently outperform ResNet.
## Backbone - DetNet

<table>
<thead>
<tr>
<th>Models</th>
<th>scales</th>
<th>mAP</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;60&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;70&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;80&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;85&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>over all scales</td>
<td>37.9</td>
<td>60.0</td>
<td>55.1</td>
<td>47.2</td>
<td>33.1</td>
<td>22.1</td>
</tr>
<tr>
<td></td>
<td>small</td>
<td>22.9</td>
<td>40.1</td>
<td>35.5</td>
<td>28.0</td>
<td>17.5</td>
<td>10.4</td>
</tr>
<tr>
<td></td>
<td>middle</td>
<td>40.6</td>
<td>63.9</td>
<td>59.0</td>
<td>51.2</td>
<td>35.7</td>
<td>23.3</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>49.2</td>
<td>72.2</td>
<td>68.2</td>
<td>60.8</td>
<td>46.6</td>
<td>34.5</td>
</tr>
<tr>
<td>DetNet-59</td>
<td>over all scales</td>
<td>40.2</td>
<td>61.7</td>
<td>57.0</td>
<td>49.6</td>
<td>36.2</td>
<td>25.8</td>
</tr>
<tr>
<td></td>
<td>small</td>
<td>23.9</td>
<td>41.8</td>
<td>36.8</td>
<td>29.8</td>
<td>17.7</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>middle</td>
<td>43.2</td>
<td>65.8</td>
<td>61.2</td>
<td>53.6</td>
<td>39.9</td>
<td>27.3</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>52.0</td>
<td>73.1</td>
<td>69.5</td>
<td>63</td>
<td>51.4</td>
<td>40.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>scales</th>
<th>mAR</th>
<th>AR&lt;sub&gt;50&lt;/sub&gt;</th>
<th>AR&lt;sub&gt;60&lt;/sub&gt;</th>
<th>AR&lt;sub&gt;70&lt;/sub&gt;</th>
<th>AR&lt;sub&gt;80&lt;/sub&gt;</th>
<th>AR&lt;sub&gt;85&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>over all scales</td>
<td>52.8</td>
<td>80.5</td>
<td>74.7</td>
<td>64.3</td>
<td>46.8</td>
<td>34.2</td>
</tr>
<tr>
<td></td>
<td>small</td>
<td>35.5</td>
<td>60.0</td>
<td>53.8</td>
<td>43.3</td>
<td>28.7</td>
<td>18.7</td>
</tr>
<tr>
<td></td>
<td>middle</td>
<td>56.0</td>
<td>84.9</td>
<td>79.2</td>
<td>68.7</td>
<td>50.5</td>
<td>36.2</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>67.0</td>
<td>95.0</td>
<td>90.9</td>
<td>80.3</td>
<td>63.1</td>
<td>50.2</td>
</tr>
<tr>
<td>DetNet-59</td>
<td>over all scales</td>
<td>56.1</td>
<td>83.1</td>
<td>77.8</td>
<td>67.6</td>
<td>51.0</td>
<td>38.9</td>
</tr>
<tr>
<td></td>
<td>small</td>
<td>39.2</td>
<td>66.4</td>
<td>59.4</td>
<td>47.3</td>
<td>29.5</td>
<td>19.6</td>
</tr>
<tr>
<td></td>
<td>middle</td>
<td>59.5</td>
<td>87.4</td>
<td>82.5</td>
<td>72.6</td>
<td>55.6</td>
<td>41.2</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>70.1</td>
<td>95.4</td>
<td>91.8</td>
<td>82.9</td>
<td>69.1</td>
<td>56.3</td>
</tr>
</tbody>
</table>
### Backbone - DetNet

<table>
<thead>
<tr>
<th>Models</th>
<th>Backbone</th>
<th>mAP</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;</th>
<th>AP_s</th>
<th>AP_m</th>
<th>AP_l</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD513 [3]</td>
<td>ResNet-101</td>
<td>31.2</td>
<td>50.4</td>
<td>33.3</td>
<td>10.2</td>
<td>34.5</td>
<td>49.8</td>
</tr>
<tr>
<td>DSSD513 [3,37]</td>
<td>ResNet-101</td>
<td>33.2</td>
<td>53.3</td>
<td>35.2</td>
<td>13.0</td>
<td>35.4</td>
<td>51.1</td>
</tr>
<tr>
<td>Faster R-CNN +++ [11]</td>
<td>ResNet-101</td>
<td>34.9</td>
<td>55.7</td>
<td>37.4</td>
<td>15.6</td>
<td>38.7</td>
<td>50.9</td>
</tr>
<tr>
<td>Faster R-CNN G-RMI&lt;sup&gt;2&lt;/sup&gt; [38]</td>
<td>Inception-ResNet-v2</td>
<td>34.7</td>
<td>55.5</td>
<td>36.7</td>
<td>13.5</td>
<td>38.1</td>
<td>52.0</td>
</tr>
<tr>
<td>RetinaNet [4]</td>
<td>ResNet-101</td>
<td>39.1</td>
<td>59.1</td>
<td>42.3</td>
<td>21.8</td>
<td>42.7</td>
<td>50.2</td>
</tr>
<tr>
<td>FPN [33]</td>
<td>ResNet-101</td>
<td>37.3</td>
<td>59.6</td>
<td>40.3</td>
<td>19.8</td>
<td>40.2</td>
<td>48.8</td>
</tr>
<tr>
<td>FPN</td>
<td>DetNet-59</td>
<td><strong>40.3</strong></td>
<td><strong>62.1</strong></td>
<td><strong>43.8</strong></td>
<td><strong>23.6</strong></td>
<td><strong>42.6</strong></td>
<td><strong>50.0</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>Backbone</th>
<th>mAP</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;</th>
<th>AP_s</th>
<th>AP_m</th>
<th>AP_l</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNC [39]</td>
<td>ResNet-101</td>
<td>24.6</td>
<td>44.3</td>
<td>24.8</td>
<td>4.7</td>
<td>25.9</td>
<td>43.6</td>
</tr>
<tr>
<td>FCIS [40] + OHEM [41]</td>
<td>ResNet-101-C5-dilated</td>
<td>29.2</td>
<td>49.5</td>
<td>-</td>
<td>7.1</td>
<td>31.3</td>
<td>50.0</td>
</tr>
<tr>
<td>FCIS+++ [40] + OHEM</td>
<td>ResNet-101-C5-dilated</td>
<td>33.6</td>
<td>54.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mask R-CNN [33]</td>
<td>ResNet-101</td>
<td>35.7</td>
<td>58.0</td>
<td>37.8</td>
<td>15.5</td>
<td>38.1</td>
<td>52.4</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>DetNet-59</td>
<td><strong>37.1</strong></td>
<td><strong>60.0</strong></td>
<td><strong>39.6</strong></td>
<td><strong>18.6</strong></td>
<td><strong>39.0</strong></td>
<td><strong>51.3</strong></td>
</tr>
</tbody>
</table>
Challenges - Head

- Speed is significantly improved for the two-stage detector
  - RCNN -> Fast RCNN -> Faster RCNN -> RFCN
- How to obtain efficient speed as one stage detector like YOLO, SSD?
  - Small Backbone
  - Light Head
Head – Light head RCNN

Challenges - Scale

- Scale variations is extremely large for object detection
Challenges - Scale

- Scale variations is extremely large for object detection
- Previous works
  - Divide and Conquer: SSD, DSSD, RON, FPN, ...
    - Limited Scale variation
  - Scale Normalization for Image Pyramids, Singh etc, CVPR2018
    - Slow inference speed
- How to address extremely large scale variation without compromising inference speed?
Scale - SFace

Challenges - Batchsize

• Small mini-batchsize for general object detection
  • 2 for R-CNN, Faster RCNN
  • 16 for RetinaNet, Mask RCNN

• Problem with small mini-batchsize
  • Long training time
  • Insufficient BN statistics
  • Inbalanced pos/neg ratio
Batchsize – MegDet

Challenges - Crowd

• NMS is a post-processing step to eliminate multiple responses on one object instance
  • Reasonable for mild crowdness like COCO and VOC
  • Will Fail in the case when the objects are in a crowd
Crowd - CrowdHuman

Introduction to Face++ Detection Team

- Category-level Recognition
- Detection
  - Face Detection:
  - Human Detection:
  - General Object Detection:
      https://github.com/zengarden/light_head_rcnn
- Segmentation
- Skeleton
    https://github.com/chenyilun95/tf-cpn
Thanks