Scaling Object Detection up to More Categories

Y. Lu, B. Xiao, Z. Gong, X. Zhang, H. Guo, L. Wen
Bytedance AI Lab
From COCO to Object365

- More object categories: 80 -> 365
- More training images: 11W -> 60W
- More data → more gains
- But...
From COCO to Object365

- Object365 dataset has a longer tail
From COCO to Object365

- Class imbalance problem is more severe on Object365

<table>
<thead>
<tr>
<th></th>
<th>COCO</th>
<th>Object365</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max #Instance</td>
<td>262465</td>
<td>2120895</td>
</tr>
<tr>
<td>Min #Instance</td>
<td>198</td>
<td>28</td>
</tr>
<tr>
<td>Max / Min</td>
<td>1326</td>
<td>75746</td>
</tr>
</tbody>
</table>
From COCO to Object365

- More object classes: 80 -> 365
- More training images: 11W -> 60W
- But longer tail and more imbalance data
- What if we simply apply COCO models onto 365 classes?
From COCO to Object365

  - mAP of 44.7 on COCO
- Achieve only mAP of 29.5 on the validation set of Object365

Class AP distribution on Object365

- The AP is worse for the classes with less instances
A detailed look on class 301-365

- **39** out of 65 classes has 0 AP!
A detailed look on class 301-365

- **Zero AP classes:** okra, scallop, pitaya

Most small things with heavy clustering
A detailed look on class 301-365

- **High AP classes:** donkey, polar bear, seal

Most animals, with large scales and simple appearance
Possible solutions

- Expert models
- Data distribution resampling
Expert models

- Fine-tuning the full classes model on class 301-365
- mAP on Class 301-365: 18.4 → 29.5*
  - APs of 46 classes increase

* evaluated on tiny track val set
Expert models

- Introducing expert models improves overall mAP by 1.1
  - Expert 1: 301-365 classes
  - Expert 2: 151-300 classes

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>General model</td>
<td>29.6</td>
</tr>
<tr>
<td>General + Expert 1</td>
<td>29.9</td>
</tr>
<tr>
<td>General + Expert 1 + Expert 2</td>
<td>30.7</td>
</tr>
</tbody>
</table>
Data distribution resampling

- Down-sample classes with huge number of instances
Data distribution resampling

- Down-sample classes with huge number of instances
  - mAP of Class 301-365: 18.4 -> 23.3*
  - overall mAP: 31.3 -> 31.0
- No gain on overall mAP
Further improvement

mAP on validation set

mAP

35

33.5

32

30.5

29.6

29

Cascade RCNN

ResNext101 64x4d

+expert models

+1.1
Further improvement

- A better pretrained backbone improves mAP by 0.6
Further improvement

- Multi-scale training improves mAP by 0.9
Further improvement

- Multi-scale testing and soft NMS improve mAP by 1.4
Further improvement

- Model ensemble improves mAP by 0.9
Tiny track experiments

- Baseline: Cascade R-CNN with ResNext101 64x4d pretrained on COCO
- Pretraining on Full Track dataset improves mAP by 4.2
Tiny track experiments

- Other tricks improve mAP by 5.3

Baseline pretrained on Full Track +4.2
Better backbone +1.3
Multi-scale test & softNMS +1.1
Model ensemble +2.9
## Our final results

<table>
<thead>
<tr>
<th></th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation set (Full track)</td>
<td>34.5</td>
</tr>
<tr>
<td>Test set (Full track)</td>
<td>31.1</td>
</tr>
<tr>
<td>Validation set (Tiny track)</td>
<td>34.8</td>
</tr>
<tr>
<td>Test set (Tiny track)</td>
<td>27.4</td>
</tr>
</tbody>
</table>
Experiment details

- **Basic setting**
  - Cascade R-CNN with 3 stages
  - FPN
  - Deformable convolution

- **Backbones**
  - ResNeXt 101 64x4d / 32x8d
  - SENet154
  - ResNet152

- **Training Pipeline and settings**
  - ImageNet pre-train → COCO pre-train for 12 epochs
  - Full Track: training for 20 epochs (lr 0.1 for 6 epochs, 0.01 for 10 epochs, 0.001 for 4 epochs)
  - Tiny Track: fine-tuning for 10 epochs (lr 0.1 for 4 epochs, 0.01 for 6 epochs)
  - Batch size: 80 (2 imgs/GPU * 40 GPUs)
Conclusion

- **Data distribution matters**
  - Long tail distribution greatly degrades the overall performance

- **Expert helps general model**
  - Expert model can improve APs for long tail classes

- **General model also helps expert**
  - Large data pre-training helps the learning of long tail classes

- **Long tail problem for object detection has not been solved**
We are hiring!

We are hiring research scientists, software engineers, and interns in following areas (@Beijing, Shanghai, Shenzhen):

Machine learning, natural language processing, computer vision, speech recognition and synthesis, and distributed systems.

Email: lab-hr@bytedance.com
THANKS.