Overview

We use Fast R-CNN [Girshick, ICCV 2015] as the base detection system and VGG-16 [Simonyan and Zisserman, arXiv 2014] as the base network within Fast R-CNN.

We improve the detection accuracy by concatenating the whole image features with the fc7-layer output and using it as the input to the inner product layer before Softmax. In addition, we show that replacing fourth max pooling layer rather than fifth max pooling layer with RoI pooling layer improves mAP.

We submitted two results. One is obtained by model fusion using the same weights for all models and the other is obtained by model fusion using weights learned by Bayesian optimization on the Val2 dataset.

Base Detection System

Fast R-CNN [Girshick, ICCV 2015]

- Use VGG-16 [Simonyan and Zisserman, arXiv 2014] as the base network within Fast R-CNN
- Use single-scale mode when training and testing

Techniques to Improve mAP

- Retrain the network with object-level annotations in LOC and DET dataset and initialize the network within Fast R-CNN with retrained one
- Inspired by DeepID-net [Ouyang et al., CVPR 2015]
- Use multiple region proposal methods when testing
  - Selective Search [Uijlings et al., IJCV 2013]
  - also used when training
  - Multibox [Erhan et al., CVPR 2014]
- Test not only the original images but also horizontally-flipped ones and combine them

Network Architecture Improvements

- Use single-scale mode when training and testing
- Pre-training the network with object-level annotations
- Retrain the network with object-level annotations
- Use multiple region proposal methods when testing
- Selective Search [Uijlings et al., IJCV 2013]
- also used when training
- Multibox [Erhan et al., CVPR 2014]
- Test not only the original images but also horizontally-flipped ones and combine them

Contextual Modeling

- “Scalable, High-Quality Object Detection” [Szegedy et al., arXiv 2014]
- Concatenate the whole image features with the fc7-layer output and use it as the input to the inner product layer before Softmax
- Fix the weights of CNN for whole image

RoI Pooling at An Earlier Stage

- Replace “Pool4” rather than “Pool5” with RoI pooling layer
- When training, 1.5 times slower than replacing “Pool5” with it

Model Ensemble

We submitted two results obtained by two different model ensemble strategies.

1. Ensemble using same weights for all models
2. Ensemble using weights learned by Bayesian optimization
   - Learn weights separately for each class on Val2 (see the R-CNN paper)
   - At step $t+1$, choose weights $\omega_{t+1}$ as follows:
     
     $\omega_{t+1} = \arg\max_{\omega \in D} \max_{i=1}^N AP(\omega_i) \cdot P(y \mid \omega, U_t) dy$

     where
     
     $D = \{ w \mid \sum_{i=1}^N w_i = 1, 0 \leq w_i \leq 1 \}, U_t = \{(w_i, AP(\omega_i)) \mid i = 1, 2, \ldots, N\}$
     
     $N$ : the number of models

Results of DET Task

mAP on Val2 (Using Single Model)

- Pre-training the network with object-level annotation
- RoI Pooling layer
- mAP on Val2
- mAP on Val2 and Test (Using Four Models)
- model ensemble strategy
  - by averaging
  - using weights learned by BO on Val2

Acknowledgement

We acknowledge the support of NVIDIA and IBM Cloud with the GPU resources.