MIL-UT at ILSVRC2015

Masataka Yamaguchi, Qishen Ha, Katsunori Ohnishi, Masatoshi Hidaka, Yusuke Mukuta, Tatsuya Harada
The University of Tokyo
MIL: Machine Intelligence Laboratory

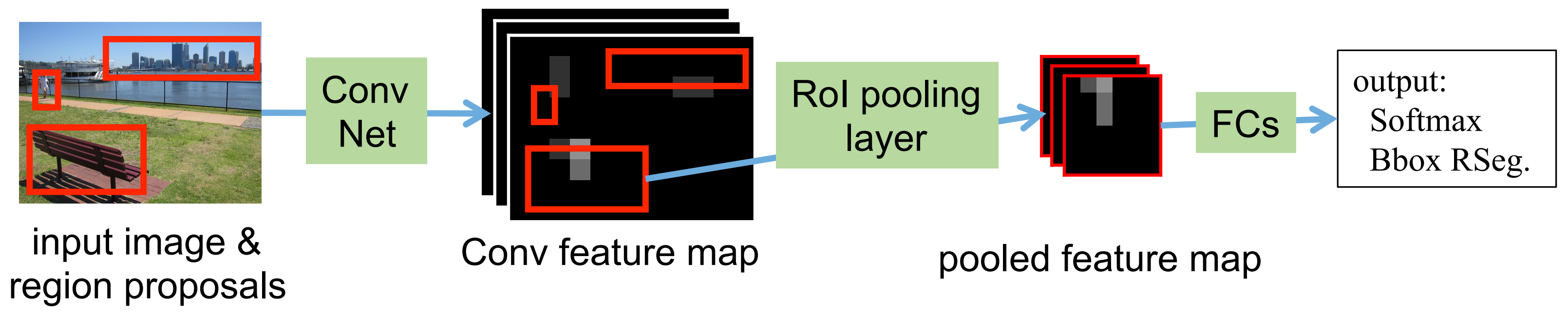
Overview

We use **Fast-RCNN** [Girshick, 2015] as the base detection system and **VGG-16** [Simonyan and Zisserman, 2014] as the base model. 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 demonstrate that **replacing pool4 layer** rather than pool5 layer **with RoI pooling layer** improves mAP.

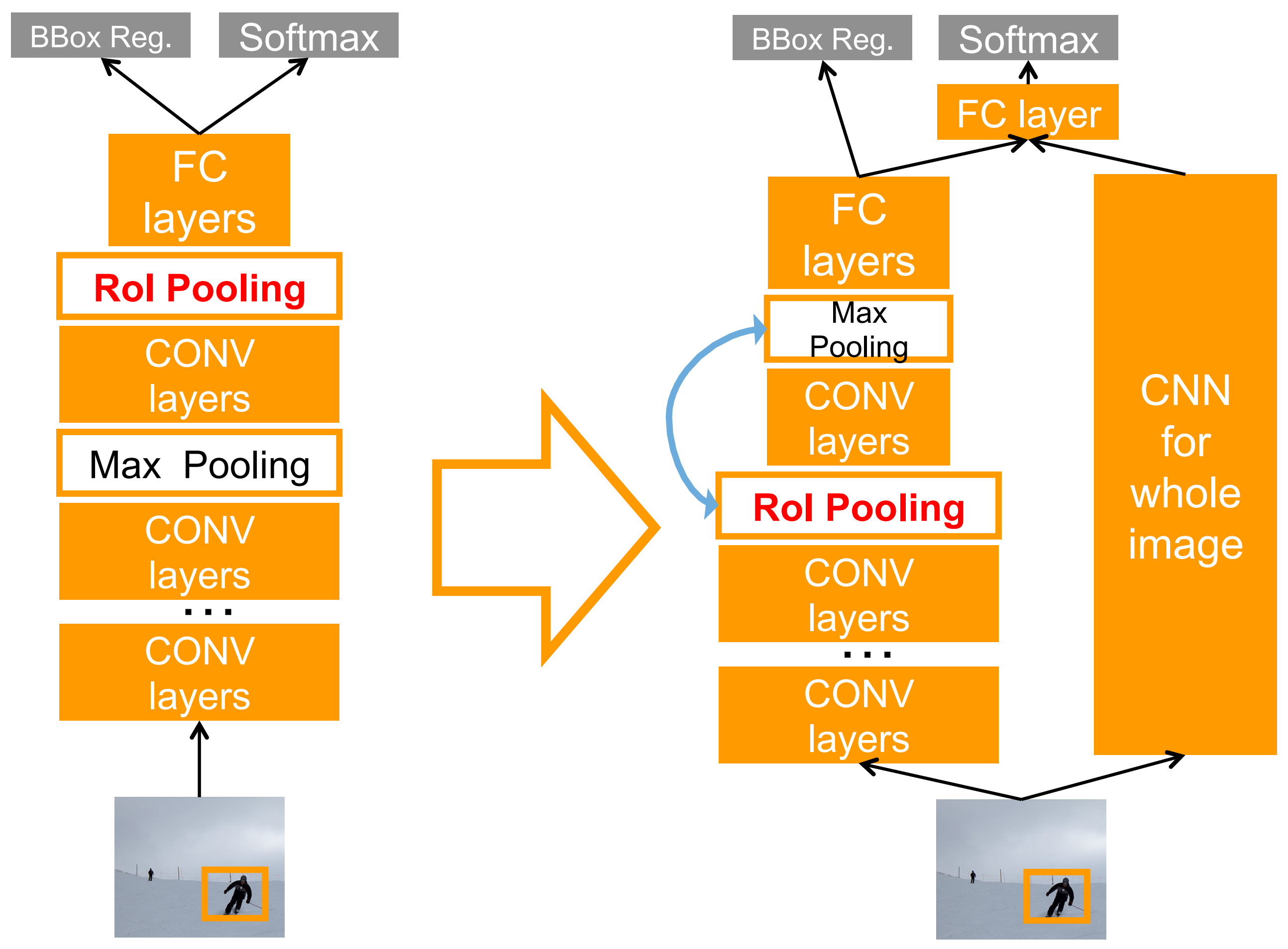
We submitted two results. One is obtained by model fusion using the same weights for all models and the other is obtained by **model fusion using weights learned by Bayesian optimization** on the val2 dataset.

Fast R-CNN



- Ross Girshick, “Fast R-CNN”, ICCV 2015
- CNN model
  - VGG-16 (Simonyan and Zisserman, "Very deep convolutional networks for large-scale image recognition.” arXiv 2014)
- Train and test on the single-scale mode

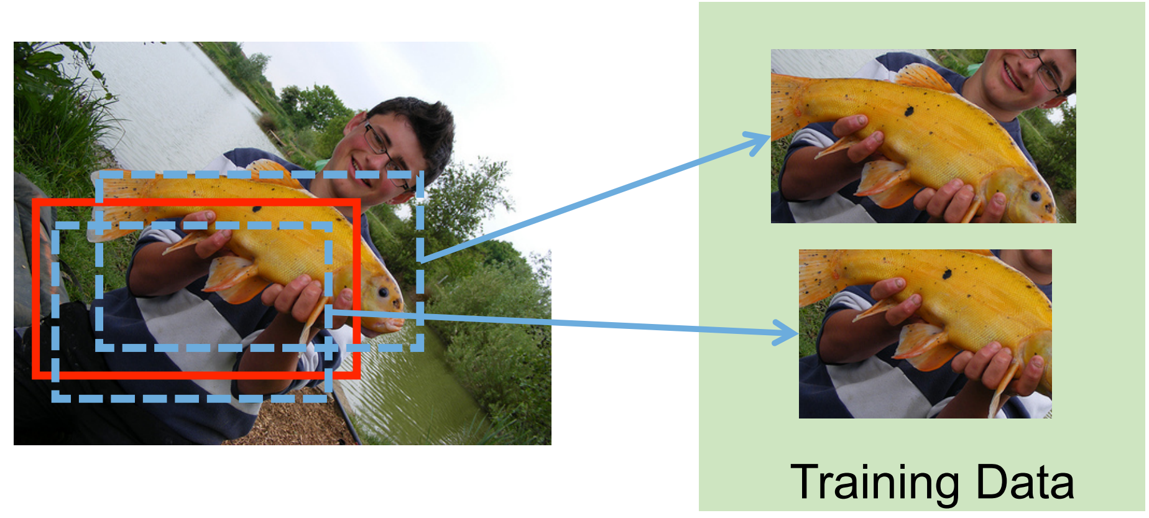
Improvement on the Network Architecture



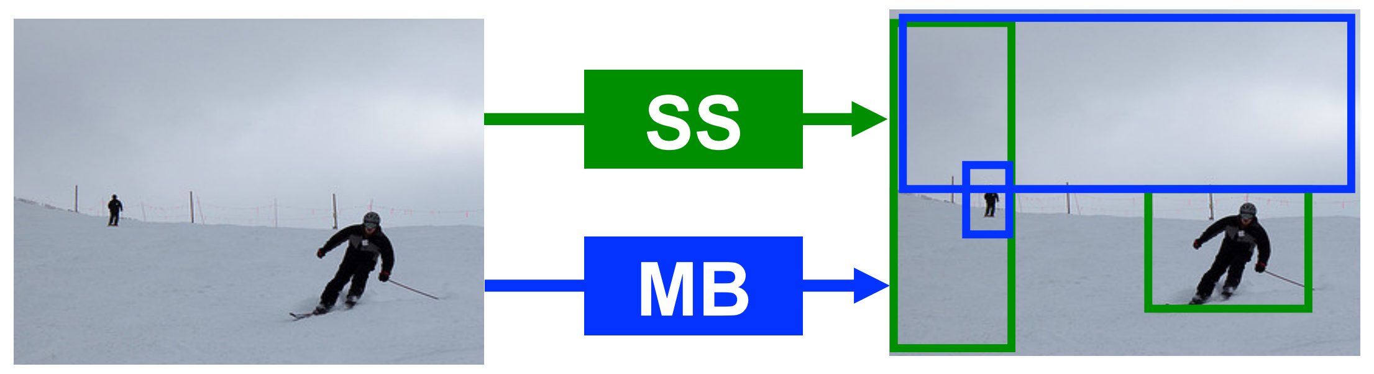
- Context Modeling
  - Concatenate the whole image features with the fc7-layer output and use it as the input to the inner product layer before Softmax
  - Fix the weights of CNN for whole image except its last FC layer for simplicity
- Replacing “pool4” layer with RoI pooling layer
  - Computational time per iteration gets 1.5 times slower

VOC2007 mAP	Pool5→RoI Pool	Pool4→RoI Pool
w/o whole image feature	66.7	68.9
with whole image feature	67.8	70.1

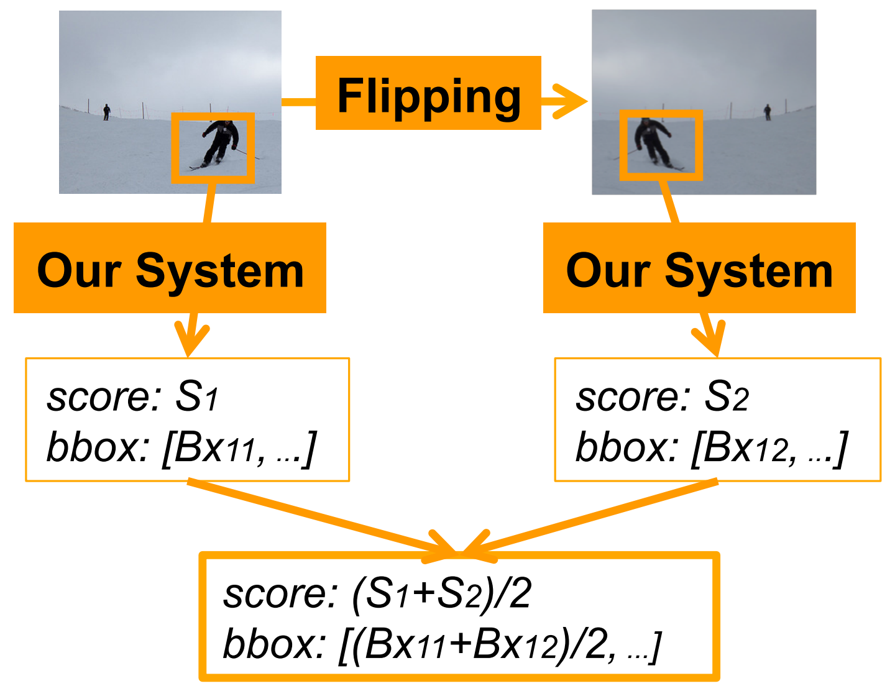
Techniques to Improve mAP



- Retrain VGG-16 on annotated bounding boxes in CLS-LOC and DET dataset
  - DeepID-net [OuyangTODO]
  - We follow the RCNN framework[??]



- Use multiple region proposal methods when testing
  - Selective Search [TODO refer]
    - when training and testing
  - Multibox [TODO refer]
    - when testing



- Test not only the original images but also horizontally-flipped ones and combine them

Model Ensemble

- 1. Ensemble using same weights for all models
- 2. Ensemble using weights learned by Bayesian Optimization
  - At step t+1, choose weights $\boldsymbol{w}_{t+1}$ as follows:

$$\boldsymbol{w}_{\boldsymbol{t+1}} = \underset{\boldsymbol{w}\in D}{\operatorname{argmax}} \int \max\left(0, y - \max_{i=1,\cdots,t} AP\left(\boldsymbol{w_i}\right)\right) P\left(y \mid \boldsymbol{w}, U_t\right) dy$$

where

$$D = \left\{\boldsymbol{w} \mid \sum_{i}^{N} w_i = 1, 0 \le w_i \le 1\right\}, U_t = \left\{\left(\boldsymbol{w_i}, AP\left(\boldsymbol{w_i}\right)\right) \mid i = 1, 2, \cdots, t\right\}$$

N : *the number of models*

  - Learn weights separately for each class on Val2 (see R-CNN paper)

Results of DET Task

- mAP Using Single Model (on Val2)

	Pool5→RoI	Pool4→RoI
original VGG-16	42.9	44.2
pre-trained on annotated boxes	45.6	45.6

- mAP Using Single Model (on Val2 and Test)

	mAP on Val2	mAP on Test
by averaging	48.1	todo
using weights learned by BO	50.6	todo

Acknowledgement

IBM
NVIDIA