Few-shot adaptive Faster R-CNN — Supplementary Material

Tao Wang[1] Xiaopeng Zhang[1,2] Li Yuan[1] Jiashi Feng[1]
[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2]Huawei Noah's Ark Lab, Shanghai, China
twangnh@gmail.com zhangxiaopeng12@huawei.com ylustcnus@gmail.com elefjia@nus.edu.sg

Implementation details

We run the experiments with Nvidia GeForce GTX TITAN X GPUs. We use PyTorch to implement the proposed model. All the network models, including the domain discriminators, are optimized with common SGD optimizer with momentum. The learning rate is set as 0.0001 for both the feature generator and domain discriminators. Each mini-batch contains 1 target domain image and 3 source domain images. The 4 images are sampled randomly. Features from the target domain image and one source domain image are paired to generate the source-target feature samples. Features from the other two source domain images are paired to form the source-source feature samples. The image-level domain discriminators have input batch size of 400, 200, 100 respectively for small, medium and large scale image-level adaptation module. A half of the input batch is composed of source-target pairs and the other half contains source-source pairs. The instance-level domain discriminator's input batch size is not fixed and is based on the object instance presented in the batch sample. For the Cityscapes to Foggy Cityscapes scenario, to implement the instance-level adaptation module, we split the dataset according to object class. We feed image batch containing objects of one class for each update, the class of each update is randomly selected.

When source model feature regularization is applied, one of the the three source images discussed above is fed through both source trained model and feature generator to calculate the regularization term.

Discriminator architecture

For the image-level adaptation module, the discriminator is composed of Conv-3x3-1024-1024, Relu, Avg_pool, fc-1024-1 sequentially. Here Conv-3x3-1024-1024 indicates convolution layer with 3x3 kernel and channel number of 1024 for both input and output feature, where channel number 1024 is the result of concatenation of pooled VGG16 relu_5_3 feature for the paired samples; fc-1024-1 denotes fully connected layer with input dimension of 1024 and output dimension of 1, corresponding to source and target domain prediction (*i.e.*, 0 for source domain and 1 for target domain).

For instance-level adaptation module, the discriminator is composed of fc-8192-1024, dropout-0.4, fc-1024-2C sequentially. Where fc-8192-1024 denotes fully connected layer with 8192 input dimension and 1024 output dimension, here dimension number of 8192 is the result of concatenation of VGG16 fully connected layer output for the paired samples; dropout-0.4 denotes dropout layer with zero probability of 0.4; fc-1024-2C denotes fully connected layer with input dimension of 1024 and output dimension of $2 \times C$, where C is the number of object classes.