Deep Learning Assignment 3 Report(2018)

Domain Shift in Semantic Segmentation

Rajeev Nagarwal(2014084), Harshit Verma(2014042), Himanshu Sagar(2014046)

May 5, 2018

1 Dataset Preprocessing

We used SYNTHIA RAND CVPR'018 dataset for source domain. In the source domain there are 13,410 annotated images. For the target domain we used CAMVid dataset.The dataset consisted of a video, we sampled frames from it at fps = 4. The synthia dataset had 12 classes.

2 Approach and Architecture

We used the training technique and architecture give in the paper Unsupervised PixelLevel Domain Adaptation with Generative Adversarial Networks, 2017. The architecture consists of three blocks: Discriminator, Generator, Classifier. Generator takes the source domain image and tries to generate fake image from the source domain. Discriminator takes the target domain image and the fake generate image from the generate image. It tries to discriminate between the fake and real image. Classifier is our semantic segmentation classifier. We have used SegNet 2015 classifier for this purpose. The SegNet classifier we have used consists of 3 convolution layers followed by 3 deconvolution layers. Along with convolution and deconvolution, we have used batch normalization. For optimization we are using two types of losses. First loss is Domain loss which involves discriminator discriminating between real and fake images. Second loss is the classification loss which minimizes the semantic segmentation classification loss. We are using cross domain entropy loss. We used leaky relu instead of relu activation with slope 0.2. After every convolutional layer in discriminator we introduced dropout layer with keep probability of 0.9. The approach and architecture that we have used are given in the following figures.

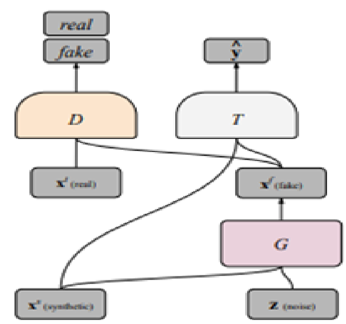

Figure 1: D stands for Discriminator, G stands for Generator and T stands for Classifier

3 Challenges

We faced many challenges. Some are given below:

1) Image size we used was 720x960. With this image size we were running into memory exhaustion problem. Also with this image size we were able to use only batch size of 1, which was hardly of any use. To deal with this problem, we resized the image to 180x240. With this configuration we were able to use batch size of 20.

2) We faced the problem of NaN loss for adversary. We solved this problem by using sigmoid activation in the loss function instead of using at the architec-

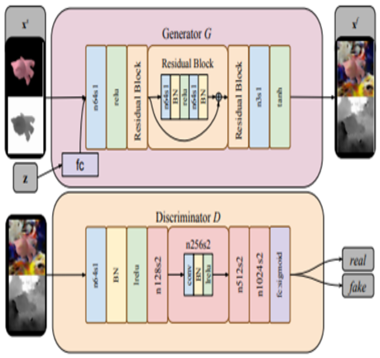

Figure 2: Architecture for Generator and Discriminator. Generator along with an image takes a noise. Discriminator takes input as an image.

$$\mathcal{L}_d(D,G) = \mathbb{E}_{\mathbf{x}^t}[\log D(\mathbf{x}^t;\boldsymbol{\theta}_D)] + \mathbb{E}_{\mathbf{x}^s,\mathbf{z}}[\log(1 - D(G(\mathbf{x}^s,\mathbf{z};\boldsymbol{\theta}_G);\boldsymbol{\theta}_D))]$$

Figure 3: Domain Loss

$$\mathcal{L}_t(G,T) = \mathbb{E}_{\mathbf{x}^s,\mathbf{y}^s,\mathbf{z}}\big[-\mathbf{y}^{s\top}\log T\,(G(\mathbf{x}^s,\mathbf{z};\boldsymbol{\theta}_G);\boldsymbol{\theta}_T) - \mathbf{y}^{s\top}\log T(\mathbf{x}^s);\boldsymbol{\theta}_T\big] \qquad (3)$$

Figure 4: Classification Loss

Figure 5: Segmentation output vs. Groundtruth. Test data taken from youtube

ture level.

4 Results

To analyse the performance of our model we use the trained classifier on the target dataset and we also run our model on the Indian dashcam road videos from the youtube. Below are some of the results.