Hacettepe University Department of Computer Engineering
BBM 418 - Computer Vision Laboratory

Assingment 3
Single Object Tracking with Regression Networks

21328064 Kübra Hanköylü

May 30, 2019

1 Introduction

In this assignment, we were expected to build a basic single object tracker by training the network on the given videos. For that purpose, the network will be trained with two frames features. Namely, we will keep frames as pairs like $(t_0,\ t_1)$, $(t_1,\ t_2)$ etc. and combined features of the tuples will be used the train process.

From those combined features, the network will learn to predict a bounding box which will contain the object to be tracked. For this aim, the network will benefit from previous frame's bounding box to predict the possible position of the object in the current frame.

2 Implementation Details

2.1 Data Preparation

First of all, I read the annotations from file. From those annotations, I have extracted the directory of frames and then, I read the frames from the video dataset file. To make process easier, I kept the frame and its annotation in a dictionary for each frame.

For the aim of reading dataset, I have written two custom dataset classes from Pytorch's Dataset Class whose name are "CustomDataset" and "CropDataset". CustomDataset class is used for reading data with no change on frame. CropDataset class is executing the following processes:

- Cropping the frame in the size of 2 times enlarged search region which has obtained from the given target bounding box in annotation.
- Rescale the cropped image's size to (224, 224) for feeding process of the network.

CropDataset class executes the above processes on the annotations as well. Train and validation datasets have read separately and the network trained with both of them sequantially. So with that, I achieved a combine training data which includes both. Finally, test dataset have read with CustomDataset class.

2.2 Model Preparation

Firstly, I have loaded pre-trained Vgg-16 model. Then, I have extracted the pool-5 layer features for each image in the pairs. After that, AvgPool2d has applied to those features so that I have obtained 1x512 vectors. For each pair, the vectors of the features concatenated to get 1x1024 vector.

Lastly, I have changed the fully-connected layers of the model's classifier. So, I had fully connected layers such as FC(1024,1024)-ReLU-FC(1024, 1024)-ReLU-FC(1024, 4) that maps input features to ground-truth bounding box of the second frame relative to the search region, being an output vector in the size of 1x4.

2.3 Training of The Network

The network have trained with both train and validation datasets. In each epoch, data called by random indices to achieve the shuffled dataset goal. For each pair in the dataset, I fed the models classifier with features of the first element in the pair. Then, I got the predicted annotations for rescaled version of second element in the pair as output.

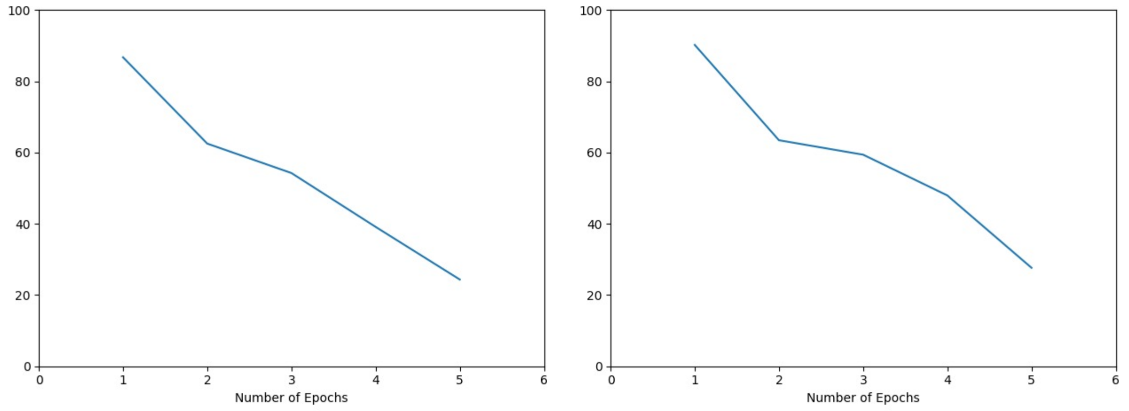
After that, I calculated the loss with MSE function and called backward process with that loss to train the network. In each epoch in the training process, I kept the loss values for the aim of visualization.

2.4 Testing of The Network

In the test part of the network, I have extracted the features of frames in each pair of test dataset. I used trained model to predict annotations by feeding those features. The predicted annotations are for a cropped and rescaled image. For that reason, I have reversed the operations crop and rescale on annotations. By this way, I get predicted annotations for original sized frames.

Finally, I have calculated and saved the loss values for each predicted annotations. And, I kept the frame and predicted annotations together in a dictionary.

3 Experimental Results



(a) Loss Graphic for Learning rate : 0.001

(b) Loss Graphic for Learning rate : 0.001

Figure 3.1: Train and Validation Loss Graphs

As you can see above graphs, the training and validation loss for different learning rates (Figure 3.1 (a) and (b)) are decreasing in each epoch. Unfortunately, test part of the network could not be finished. It executes until 289th sample then my computer starts to freeze. For that reason, I do not have any test results to show and explain. I believe, this issue is about RAM usage. But, I think the code will work on a much stronger machine just fine.