Date – 28th March 2019

CS783 Assignment 3

Object Detection

Team Members

Mohammad Areeb (160403)

Somesh Pandey (160699)

Training Dataset Pre-processing

1.Using given annotations, we cropped and resized chairs, bottles and aeroplanes form given training images and store them in respective folders.

2.Extracted the backgrounds as follows –

a) If no object is detected in image, entire image is treated as background.

b) If 1 object is detected, 8 backgrounds are extracted, 2 from each corner

(1 vertical, one horizontal)

c) If more than one object is detected, 4 more backgrounds are extracted from between the objects, giving total of 12 backgrounds.

For each image, as many backgrounds are extracted as there are objects in that image. (Check ipython notebook for better explanation)

Doing this gives us total of 15692 backgrounds.

We sorted them according to decreasing order of size and chose last 5692 backgrounds which had minimum useful information, as the top images had much information variance and could affect training on non-background classes.

Methods used for Feature Extraction

1. **Single Layer Detection – we used Resnet18, froze all layers and replaced last fully connected layer (having 1000 classes) with new fc having 4 classes.**

2. **Two Layer detection – removed these three layers from last –**

a) Layer 10 - fully connected layer

b) Layer 9 - AdaptiveAveragePool2D layer

c) Layer 8 - Sequential Block containing some Relu, BatchNorm and Conv2D layers

We took output from Layer 7 after applying AdaptiveAveragePool2D and an fc of size 4

Experimentation

1.We experimented with the number of backgrounds to be taken for training – tried 800,1000,1500,2000,5692.

2.Since the 3 classes had significantly different number of images, we tried to balance this by performing augmentation on only bottles and aeroplanes – created 1249 extra planes, and 1300 bottles. However, we did not get any significant improvements using this approach.

3.We tried different values of Aspect Ratio and Stride for Sliding Window approach.

4.We tried different Batch Sizes (16,32,64), different Learning Rates (0.01,0.001 and 0.003), number of EPOCHS and several optimizers (adam, SGD).

Problems Faced

1.Problem statement was VERY ambiguous and it took us a very long time to figure out what was actually being asked.

2.We were getting good accuracy using Validation set from training data, but got poor accuracy on actual test set.

3.We were new to pyTorch, and hence had slight difficulty adjusting to the new library.

4.We faced some problem in creating the architecture for 2-layer detection i.e. removing final layers and taking output from intermediate layers.

References

1.Sliding Window - https://www.pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/

2.Non-Max Suppression - https://www.pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/

3.mAP Documentation - https://github.com/Cartucho/mAP

4.Resnet18 Data Loading and Training – pyTorch Cifar10 Tutorial - https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html