CS783 : Visual Recognition

Aaditya Singh
160002

Rohin Garg
160583

Assignment 3: Object Detection

March 28, 2019

Sliding Windows

9 Windows are chosen for every image based on the image's size. 3 scales: 0.25, 0.5, 0.75 of the original image's shape (converted to square) were chosen and for each scale 3 aspect ratios: (1:1), (2:1) and (1:2). At each step the part of the image inside the window acts as input for the clssifier and each output is used to create a list consisting of the object which the classifier predicts, the object confidence and the box coordinates. All the boxes for the different scales and aspect ratios are stored.

Non Max Suppression

We use the boxes obtained by the sliding windows and apply non-max suppression (NMS) to predict only the best boxes from all the boxes. First of all, we remove all of the boxes whose object confidence is below a certain threshold. Then for each type of object (aeroplane, bottle and chair), we first find out the box with the largest object confidence, store it and remove all those boxes with which its intersection over union is greater than the threshold including itself. This process is repeated until no boxes remain in the original list and the stored boxes are our final predictions.

One Layer Detection

Background class

We slide windows of size and aspect ratios over an image with a stride of and choose only those part of images for background which have lesser intersection over union with all of the bounding boxes present in that image.

Classification task

We have used pre-trained ResNet-18 network for the purpose of classification. Instead of training just the fully connected layers, we also trained the ResNet-18 layers, which resulted in an accuracy of on the validation set.

Object detection

We slide windows (different for each image) over an image with a stride of 30 to obtain the list of all bounding boxes along with the object class and confidence for that image, and then iteratively remove those boxes which have high intersection over union with the boxes having the highest object confidences for all types of objects.

Two Layer Detction

We got inspired from the idea of using outputs from multiple internal convolution layers applied in Single Shot Detector (SSD) although it uses additional convolution layers, which we were not supposed to.
The central idea for using the predictions of an inner layer as in Single Shot Detector (SSD) is the fact that CNN reduces the spatial dimension gradually, which also decreases the resolution of the feature maps. Thus, we use lower resolution layers along with the higher resolution layers to detect the larger scale objects.

Classification task

We concatenated the ouptut from second last block of the ResNet-18 network after average pooling with the output of the last block after average pooling. We then added a fully connected layer and trained the entire network, which resulted in an accuracy of on the validation set.

Object detection

We again slide the same windows for the trained modified model (Concat ResNet-18). The NMS function which we used previously was used here as well.

Challenges faced

Repetitive background images

Our code for extracting background from the sliding windows resulted in a lot of repititve images. We overcame this issue by skipping steps whenever a background image was encountered.

Misclassification errors

We observed that our model was predicting a lot of backgroung patches as 'chair' instead of background quite frequently inspite of having a high validation accuracy. We estimated that this was due to a lesser number of background images in the training set, initially close to 1200. So we increased the background images to around 26000, by taking backgrounds from all 20 classes of the PascalVOC dataset. This solved the issue quite significantly.

Additional steps: Clipped ResNet-18

We also tried to clip the ResNet-18 network to obtain the output from an inner block of the network, and use it along with the outputs from the standard ResNet-18 network. We achieved this by removing the fully connected layer, the average pooling layer and the last block of the ResNet-18 network. We then added an average pooling layer and a fully connected layer. We then trained this entire network, and obtained the bounding boxes in the same way as we did before. Finally we used the bounding boxes from both the original network and applied NMS. But this approach lead to poorer results, and therefore we did not used it for the two layer detection.

Results

For Average Precision, the a bounding box was chosen as true positive is it's IOU with actual bounding box of the same class is greater then 0.5

Training Classification Accuracy:

- One Layer Detection = 99%
- Concat ResNet-18 = 99%
- Clipped ResNet-18 = 99%

Validation Classification Accuracy:

- ResNet-18 = 98%
- Concat ResNet-18 = 99%
- Clipped ResNet-18 = 94%

Mean Average Precision:

- One Layer Detection = 0.245
- Two Layer Detection or Concat ResNet-18 = 0.34

AP for the three classes:

- Aeroplane:
  - One Layer Detection = 0.31
  - Two Layer Detection or Concat ResNet-18 = 0.40
- Bottle:
  - One Layer Detection = 0.24
  - Two Layer Detection or Concat ResNet-18 = 0.32
- Chair:
  - One Layer Detection = 0.15
  - Two Layer Detection or Concat ResNet-18 = 0.26

References

Transfer Learning tutorial in Pytorch

`https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html`

Sliding window implementation for Python

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

C4W3L07 Nonmax Suppression

`https://www.youtube.com/watch?v=VAo84c1hQX8&t=365s`

Single Shot Detector for real-time processing

`https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-mult`