CSL462 - Object Detection (Autorickshaw) (Assignment 4)

Shivam Mittal
www.linkedin.com/in/shivam-mittal

Indian Institute of Technology,
IIT Ropar

1 Abstract

The main task at hand was of object detection (autorickshaws) in images. The dataset [1] consists of 800 labelled images, the first 600 of which are used for training and the latter 200 are used for testing. The method we have used is to first extract the bounding boxes from the training images, then to train a one class classifier to be able to detect parts of an auto (not the whole auto itself). This is done to handle occlusions, and make the approach scale invariant. Hence we extract fixed sized patches from the labelled bounding box and then train the Support Vector Machine. The fixed size patch is then used as our sliding window and is slid across the test image to obtain positive classifications. These positive classified bounding box then undergo non-maximal suppression and heat-map reduction to give us the final results which are pretty convincing considering the small amount of data given for training.
In the following sections, we'll describe more details about the training and testing process and then discuss the results.

2 Training

The basic training procedure implemented in Train_2015CSB1032.m is to extract features from a CNN for the multiple patches of the bounding boxes, and then train a one-class SVM over these patches because we only have foreground data (data for autorickshaw), no background labels (data for non-autorickshaw).

2.1 Data provided

The data [1] provided consists of 800 images with bounding boxes of autorickshaws. We use the first 600 images as training data and the other 200 images as test data.
The bounding boxes for the training data are cropped out from the image. Then to make the detector scale invariant as well as handle occlusions, we selected a fixed patch size of 140*140, then we extract these fixed patches from the bounding boxes at stride of 50 pixels. Features are extracted from these patches as described in the next section.

2.2 Feature extraction

As we know in recent years, deep networks are acing at image classification tasks because of there ability to extract high level features from an image for a given task. The convolutional neural networks have various layers that automatically learn the features upon training. The higher layers of the CNN contain high level features that are very efficient for classification and other tasks.
Hence, using a CNN as a feature extract by extracting its feature layer (layer before the output), and using that as features for some machine learning algorithm is a good idea. We use a pre-trained CNN that is the GoogleNet [2] Model to extract features for our images. The MatConvNet code is used which implements the GoogleNet and has pre-trained weights. We get a 1024 feature vector from this CNN by perform a forward pass and extracted the output of the last fully connected layer.

2.3 Classifier

From the features extracted from the CNN for all the patches from the detection box, we train an one class SVM. One-class classifier are for the use cases when only data for one class is given, and you want to classify an unseen instance belonging to that class or not. We use the fitcsvm function of MATLAB to train a classifier. We use radial basis function as our kernel function as it allows transformation to infininte dimension.

3 Testing

3.1 Sliding window approach

We resize the given test image to a fixed height keeping the aspect ratio same. Then we use the sliding window (with size equal to that of the patch size because that was what was used for training) with fixed stride. Then the features for these window is extracted from the forward pass of the CNN as described in the above section. These features are then fed to the trained SVM to predict whether this is a part of the auto or not. We store all the positive classified windows of autos and then we will use them to predict our final bounding boxes.
We don't need to consider multiple scales or consider different sized windows, because we trained the SVM such as to be able to detect parts of the auto and not necesarily the auto itself. Hence the given patch size window suffices, although we could decrease the stride step to increase accuracy but that would mean increased computation.

3.2 Non maximal supression and heat map consolidation

By using the sliding window approach, we will detect multiple windows, and then we transform these windows to the original scale of the image, and each window may contain either a whole or a part of the auto. We need to construct proper bounding boxes from these many bounding boxes. For example there might be 10 windows detected for a single auto, we need to reduce these windows down to 1.
For this problem, first we use Non-maximal supression (NMS). The basic idea to ignore bounding boxes which have significant overlap with each other. So, boxes which have high intersection over union with each other are ignored.
Still after the step, we might have multiple connected windows, we use the concept of heat map, to generate a single window for all the connected windows. We take a image with intensity value initialized to zeros. Then we increase the heat value for all the positive bounding boxes. So we get a heat map where there are bright values for intersecting bounding boxes, intermediate values for places where only one bounding box was there and zero value for places where there was no bounding box. We use a threshold on the heatmap to get the places on the image where there are chances of true detections.

After the thresholding step, we get an image with some places having bright values and other being dark. The bright values indicate positive detections. We use a function bwboundaries to get connected components from this image. Now, we have a finite set of connected components, and for each component we will take its rectangular cover by choosing the minimum and maximum x and y coordinates, and constructing a rectangle out of that. These will be our resulting bounding boxes. The result of the NMS and consolidation step has been demonstrated in the figures shown.

4 Results and further work

We can see that the detector works good in majority of the cases, and the auto is detected in almost all of the cases. But our detector suffers from 2 problems, one is the localization error and the other is the detection of a single bounding box for multiple autos.
The localization problem is not because of any fault in the approach, the error is because we have taken the patch size to be very large and the stride to be high. If we decrease the patch size and decrease the stride step, we will get good localization. But this was not done, because sliding window operation is already a computationally expensive operation and running the detector with decreased patch size and decreased stride makes it even more expensive computationally, but it can be done given good computation power or time, and the there are good chances of the localization errors reducing.
Also, in some cases where autos are close to each other, the consolidation

Figure 1: Before performing NMS and heat map consolidation - multiple bounding boxes

Figure 2: After performing NMS and heat map consolidation

of bounding boxes make them as one bounding box.
But, overall the detector gives good results.

5 References

[1] Dataset
http://cvit.iiit.ac.in/autorickshaw_detection/files/auto_det_chal_train_7oct.zip

[2] GoogleNet Model
http://www.vlfeat.org/matconvnet/pretrained/