CS 783 Visual Recognition
Assignment 4

Anurag Pandey-160140
Sudhanshu Bansal-160710

April 16, 2019

Introduction:

This assignment aims at implementing **Object Tracking** i.e detecting moving objects such as cars, motorcycles, people etc. and marking bounding boxes around them in a video obtained from surveillance cameras of IIT Kanpur. This video obtained from various surveillance cameras is all the dataset we have for this problem. This makes the task to be an **unsupervised learning** task as the tracking has to be done on a live video without any prior data set available. Other key points which need to be handled during this task is challenges of occlusion, scale, low lighting and also the extra unnecesary contribution from the background

Our Approach:

We tried to explore various options for accomplishing this task. We first tried implementing the ones listed in our proposal particularly implementing light weight models like Single Shot Multibox Detection(SSD) framework with minimal modification for obtaining bounding boxes. But we could not implement it fully. Likewise we proposes some other methods but later we realized that they were semi- supervised and required some sort of labels which we did not had. Hence we have to explore new options and ways around the above task.

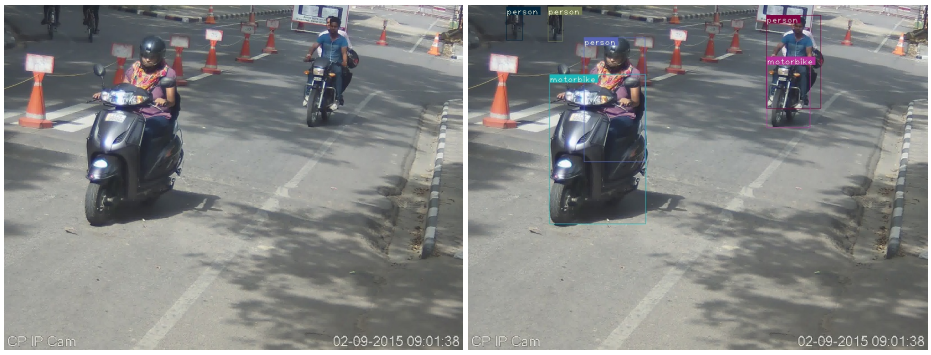
The methods which showed some significant results are listed below :

Approach 1

[2] [1] **YOLO** is an state of the art object detection algorithm trained on COCO dataset. It can be used to detect different objects in an image. Object detection if done with high efficiently, frame by frame can give a realization of real time object tracking.Of several available algorithms to do this, we implemented **SORT - Simple Online Realtime Tracking** - a 2017 paper by Alex

Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft which proposes using a Kalman filter to predict the track of previously identified objects, and match them with new detections.

The results obtained from this method are shown below.



This is one of the best methods we could reach. The other methods are listed below.

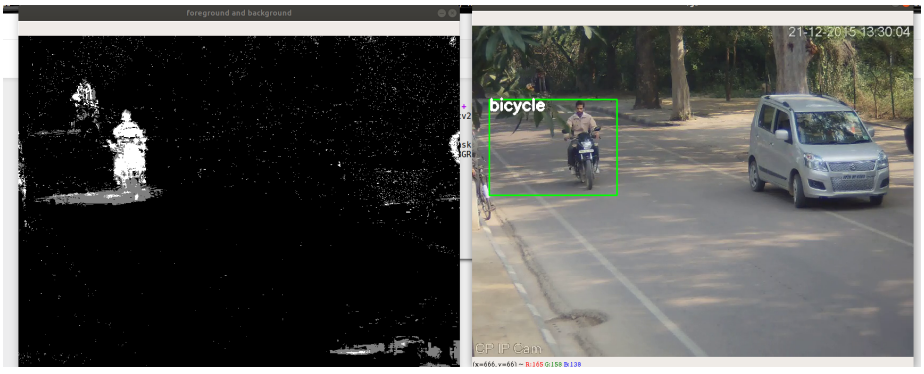
Approach 2

[3] This time we used a feature of Background removal using the Mixture of Gaussians background subtractor function from OpenCV before obtaining the bounding boxes by using **findContours()** from OpenCV.

Next we need a classifier to get the detections on these contours. We picked up pre-trained RESNET-18 model. Obtained a dataset of person, bus , motorbike, bicycle, car truck classes from the given dataset using YOLO. Fine tuned the RESNET-18 model on this dataset. The model thus obtained will serve as our classifier for detection purposes.

This detection method was followed by SORT algorithm to do the final tracking.

The results from this method are :



This method fails miserably when many objects are close to each other in an image. Also the bounding boxes for the cases where are predictions are apparently good are expanding horizontally because of the shadows of the moving objects which are not being removed by the background removal algorithm.

Approach 3

This approach is similar to the above approach. We removed the background from the images as earlier using the Mixture of Gaussians background subtract or function from OpenCV followed by obtaining contour using **findContours()**

by OpenCV.

We used the K Means clustering on the array representation of these contours to obtain classes corresponding to the objects - persons, car , truck , bus, motorcycle , bicycle. We use this KMeans clustering for detection purposes in place of any explicit model. This followed by SORT algorithm for implementing the tracking purposes.

Through out the various methods/approaches tried, we used SORT as a standard tracking algorithm and mainly focused on improving the object detection classifier given the constraint of limited dataset we had.

References

[1] A pytorch implementation of a yolo v3 object detector. https://github.com/ayooshkathuria/pytorch-yolo-v3.

[2] pytorch_objectdetecttrack. https://github.com/cfotache/pytorch_objectdetecttrack?fbclid=IwAR0Xqt0lHb7KIgy_N332BM3sHQmhsAzu1RucvAaR0TBusT2V_IqqSRUQj0I.

[3] Simple object tracking with opencv. https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/.