Date – 16th April 2019

CS783 Assignment 4

Object Detection and Tracking in Videos

Team Members

Mohammad Areeb (160403)

Somesh Pandey (160699)

Problem Description

1.The problem we have chosen is to take a video file as input, detect all objects in it as soon as they appear, track them throughout the video for as long as they are on the screen, and save this output as a video file showing bounding boxes for the objects with classes attached.

2.We also tried image segmentation on the given camera samples. (This was an optional problem that we chose to work on)

Algorithm for Detection and Tracking

1.We take a video file as input, and loop over the frames of the video.

2.Detection – We have used YOLOv3 for object detection.

The model has been pre-trained on a dataset which consists of 80 classes, including the following – 'person', 'bicycle', 'car', 'bus' and 'truck', and hence is suitable for our problem.

3. Tracking – We have used Correlational Tracker available in dlib library.

For every detected object, we assign it a dlib tracker which tracks it for the next 15 frames.

4. For each frame, after detection, a bounding box is created with class name attached to it, and all frames are saved to a video file in .mp4 format.

Best Algorithm used for Image Segmentation

1. We used a supervised segmentation model – ICNet trained on dataset CityScapes which is very similar to our requirements, and includes the above-mentioned classes.

Experimentation for Detection and Tracking

1.We first tried few trackers available in OpenCV library – 3 of them to be precise - Kernelized Correlation Filters (KCF), CSRT tracker, and MOSSE tracker.

2. We also tried Kalman Filter Tracker available in FilterPy to perform real-time tracking.

3. However, we were not getting satisfactory results with OpenCV or Kalman Filter Tracker even for single object tracking. Hence, we switched to dlib library.

4. We finally used Correlation Tracker available in dlib library for tracking multiple objects.

5.For detection, we first used SSD on each frame.

6. Later, we switched to YOLOv3 as it was giving better accuracy.

7. The YOLOv3 model has been pre-trained on a dataset which consists of 80 classes, including the following – 'person', 'bicycle', 'car', 'bus' and 'truck', and hence is suitable for our problem.

Results for Detection and Tracking

1.KCF tracker easily lost track of the objects as they were moving fast.

2. CSRT tracker was more accurate than KCF tracker, but slightly slower, and hence not suitable for our project.

3. MOSSE tracker was very fast tracker as compared to the above two, however it wasn't as accurate.

4.Kalman Filter was several times faster than dlib correlation filter, but we were not sure about its accuracy, so we preferred dlib

5.YOLOv3 was better than SSD for the purpose of object detection, due to its speed, as well as accuracy, as it has adopted many of the significant features of SSD, like anchor boxes, etc.

Experimentation for Segmentation

1. We first tried unsupervised segmentation through backpropagation, but it failed miserably.
2. We then tried it with colour quantization and also tried it on black and white images, but still did not get good segmentation.
3. We then looked for supervised models which were trained on datasets with similar contents as ours.
4. We tried PSPNet and also fine tuned some hyperparameters but for most of the images, the accuracy was unacceptable.
5. We then tried ENet, but did not get any better accuracy.
6. Finally, we chose ICNet over the others, as it was better results.

Results for Segmentation

1. Colour Quantization does increase the accuracy of segmentation to some extent.
2. ENet, ICNet, and PSPNet are of approximately same accuracy.

Problems Faced

1. As it was an unsupervised task, we tried our best to keep it that way, but unfortunately, but due to the complexity involved, the results were not quite good.
2. Choosing the final tracker was quite involved, as some trackers were giving high accuracy but were turning out to be slow, and others were doing the opposite.
3. Due to unavailability of consistent accuracy metrics, it was difficult to judge different segmentation models, and trackers too.

OUTPUT Files

1. **Link to output video files for detection and tracking - https://drive.google.com/open?id=1CDe4mehfly8QYAzQ-iww6gNGHgxu2EAv**

2. **ICNet Segmentation results on an image –**