Assignment 2

1 Overview

In this assignment for the first part we have implemented model ROLO [3] for object tracking which uses prediction of YOLO at backend. And for the second part of the assignment we have changed the architecture of ROLO and tested on existing new ones using LSTM and made it compatible for MOT.

2 ROLO

ROLO [3,4] is a spatially supervised recurrent convolutional neural network for visual object tracking. ROLO chooses YOLO in backend for object detection in frames and uses LSTM in the next stage for sequence processing. Network takes a video frames as an input and returns the coordinates of bounding box of an object being tracked. We have taken pre-trained YOLO network on ImageNet, and trained ROLO on OTB-30 subset of 30 videos from benchmark dataset. And performed following experiments:

1. Training on 22 videos and testing on the rest 8 videos.
2. Training LSTM model with 1/3 frames of OTB-30 and test on the rest frames.

3 Part 2 Models

3.1 Bidirectional LSTMs

We replaced LSTMs with Bidirectional LSTMs to track the single object. IOU for Skating1 = 0.43

3.2 Re3

Re3[1,2], an algorithm that uses a recurrent neural network to track generic objects in a variety of nautral scenes and situations. Two image crops fed to the convolutional layers are used with skip connection in between. Output from convolutional layers are fed into single fully connected layer and an LSTM, which finally returns the bounding box.

To use it for MOT, we first initialise detections as is and then depending upon maximum iou matching (using maximum bipartite matching) from previous step, we provide labels to the image.

Re3 is trained on ILSVRC2016 (Imagenent Video) 3862 videos with 1,122,397 images.

4 Algorithm

4.1 Pre-Processing

In the data pre-processing phase firstly we have normalized the data and reshaped the frames to the size of 468 x 468. Also for the ROLO experiments we have split the data into 1/3 parts for training and rest for testing.

5 Results

Results are shown in Figure 3 and 4. Different color bounding have different significance. Red bounding box represent ground truth, blue bounding box represent YOLO prediction and Green represents our model prediction. Figure 3 and 4 shows accuracy of our model. Figure 4 clearly depicts robustness of our model as missing of blue bounding box shows failure of YOLO.

Figure 1 and 2 represent MOT output using Re3 made from our custom test file.

Table 1 and 2 shows difference in IOU results of YOLO and ROLO on different class of videos.

Table 1: Experiment 1 Results (IOU)

TestID	YOLO	ROLO	CLASS
20	0.37	0.14	Girl2
21	0.46	0.05	Skating1
22	0.36	0.46	Skater
23	0.56	0.48	Skater2
24	0.56	0.49	Dancer
25	0.26	0.48	Dancer2
26	0.66	0.08	Carscale
27	0.49	0.53	Gym
28	0.45	0.18	Human8
29	0.29	0.1	Jump

Table 2: Experiment 2 Results (IOU) for step=6

TestID	YOLO	ROLO	CLASS
20	0.37	0.14	Girl2
21	0.46	0.43	Skating1
22	0.36	0.67	Skater
23	0.56	0.47	Skater2
24	0.56	0.49	Dancer
25	0.26	0.58	Dancer2
26	0.66	0.08	Carscale
27	0.49	0.53	Gym
28	0.45	0.19	Human8
29	0.29	0.1	Jump

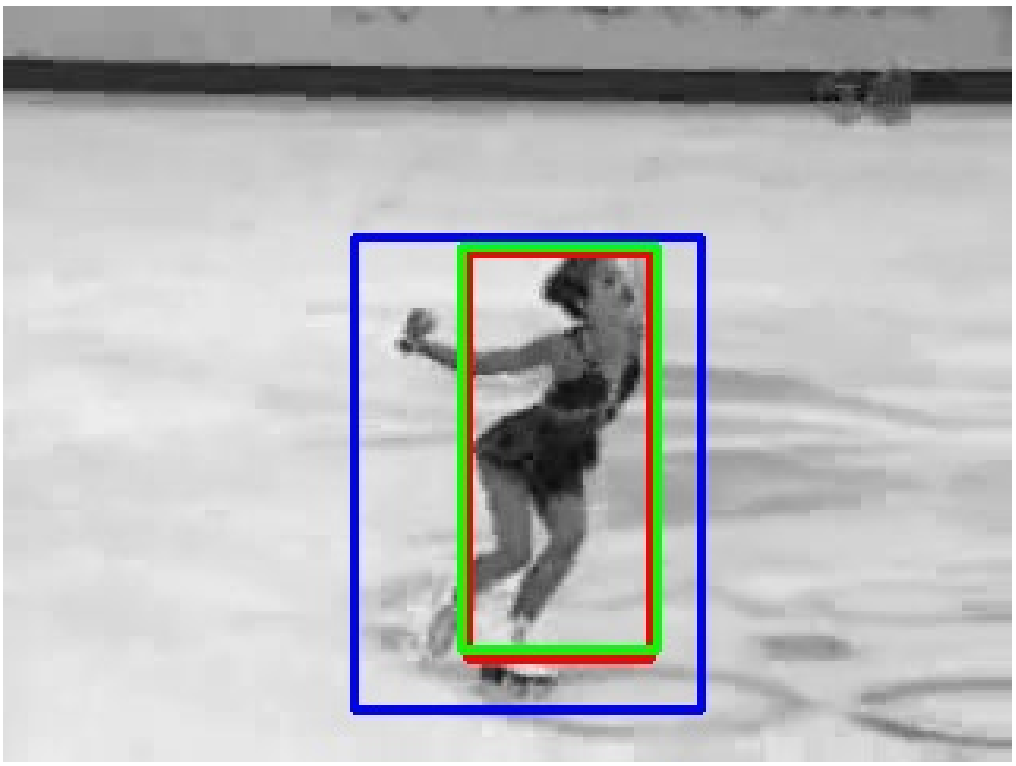

Figure 1: ROLO results on Skater

Figure 2: ROLO results on Skater

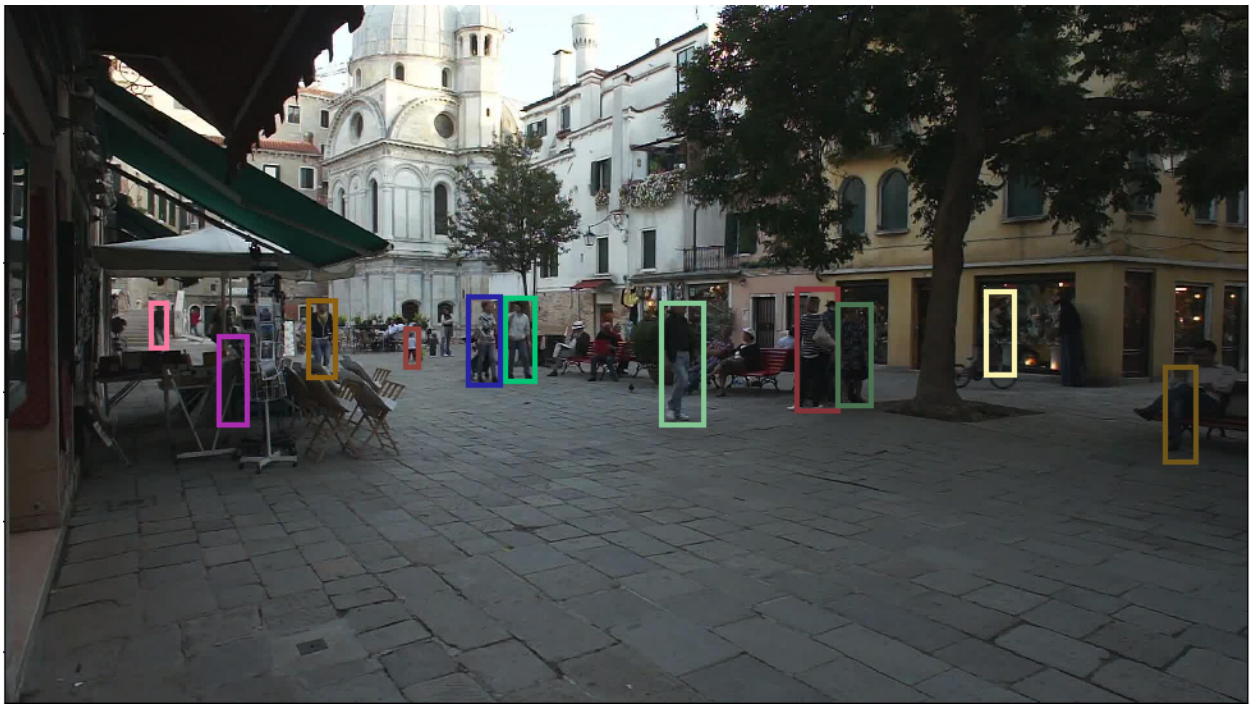

Figure 3: Re3 on MOT

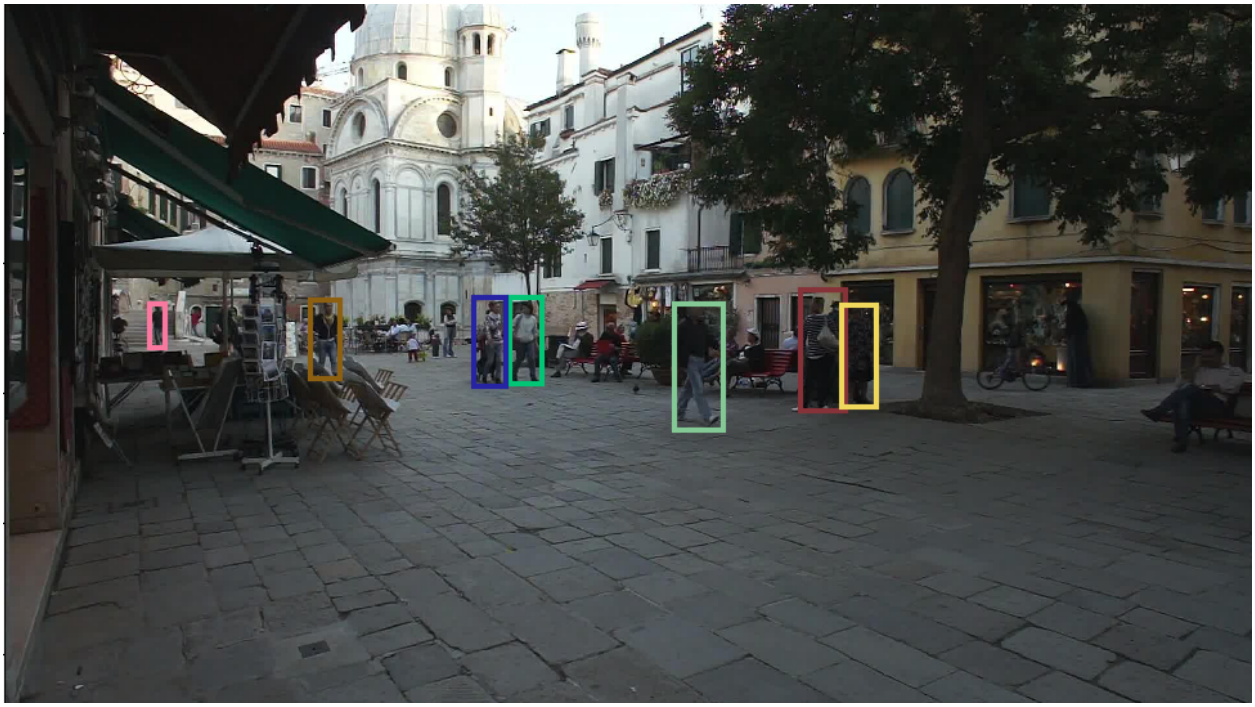

Figure 4: Re3 on MOT

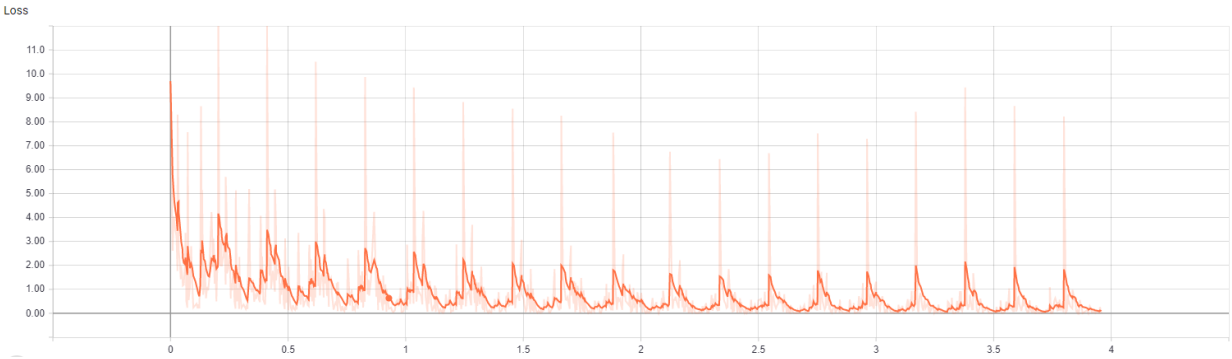

Figure 5: Training Error

6 Reference

1. Gordon, Daniel, Ali Farhadi, and Dieter Fox. "Re3: Real-Time Recurrent Regression Networks for Object Tracking." arXiv preprint arXiv:1705.06368 (2017).

2. https://gitlab.cs.washington.edu/xkcd/re3-tensorflow

3. https://github.com/hizhangp/yolo_tensorflow

4. https://github.com/Guanghan/ROLO

5. http://guanghan.info/projects/ROLO/

6. https://github.com/abewley/sort/blob/master/sort.py