Long-Term Tracking

Lenart Rupnik, 63220472

I. Introduction

In this homework, we explored the SiamFC (Siamese Fully Convolutional) model for short-term object tracking. SiamFC is a neural network architecture that uses a fully convolutional Siamese network to learn the correlation between the target object and the search region. We evaluated the model's performance on a benchmark dataset and then modified it for long-term tracking, where the target object goes out of frame and reappears after some time. The modified model was evaluated on a dataset containing long-term tracking scenarios, and the results were compared to the original model's performance. The aim of this homework was to gain a deeper understanding of SiamFC and its capabilities for both short and long-term object tracking.

II. Experiments

First we implemented SiamFC tracker with given code. We adjusted it so that it ran on our GPU and then tested it on the all sequences that were provided. Results are shown in Table. I.

Table I
Results of different configuration for long and short tracking.

Tracker	Thr.	Sampl.	Precision	Recall	F-score
SiamFC	/	/	0.57	0.311	0.40
SiamFC-LT	Flex.	Random	0.47	0.28	0.35
SiamFC-LT	4	Random	0.51	0.38	0.44
SaimFC-LT	4	Gauss	0.56	0.39	0.46

Next, we modified SiamFC for long term tracking task. This was done by defining when the target is lost and then performing re-detecting. To decide on when the target is lost, we used certain threshold of max response from SiamFC. To find the best value for threshold, we tested different configuration and values and found out that the tracker performs best when we set threshold to around 3.9. We tried other dynamic calculations for our threshold, but the results were worse, so we decided to stay with our static method. Dynamic threshold was obtained by calculating $threshold = 0.4 * max\_previous\_responses$. During testing, we used 20 random search boxes for re-detecting. The results are shown in Table I.

To improve our results, we also implemented another sampling method. Since most of the time when occlusion happens, the target is somewhere close to the last position, we used Gaussian sampling with the fixed standard deviation. With that, we managed to slightly improve the overall performance. Results can be seen in Table I

We tested different number of searching boxes on first sequence (car) since we know there is only one occlusion, so it was easier to check for number of frames until target is found. Comparison is shown in Table II. Based on our test, we see that an increasing number of samples reduces the number of frames before target is found. In contrast, with increasing number of samples we reduce the speed of finding our target. In sequences that are simple, e.g. car, it seems to be the best to set the number of samples to around 20. That way we get fast re-detecting and decent results.

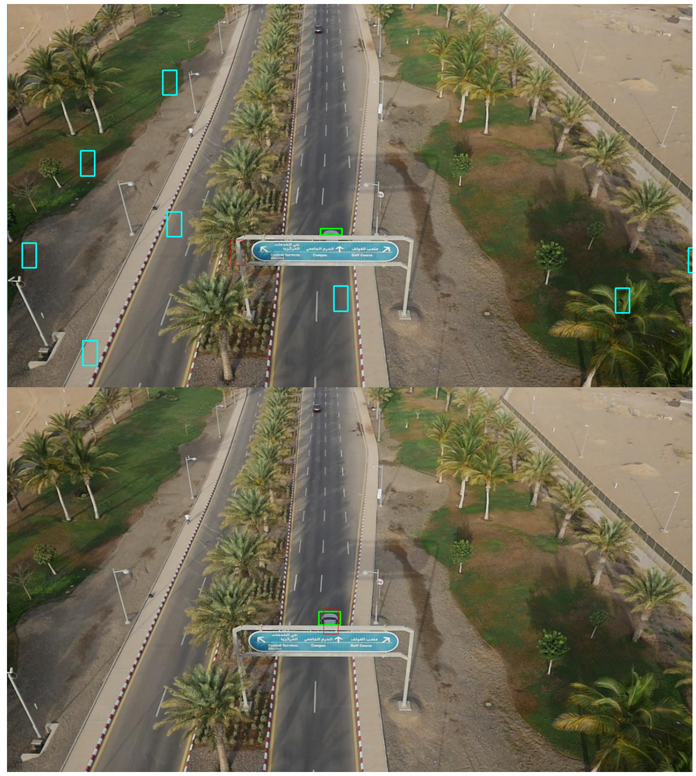

Figure 1. Tracking visualization examples for long-term tracking.

Table II
Comparison of different number of samples to number of frames.

N_Samples	Frames until re-detection	Re-detecting time
5	85	0.63 s
10	69	0.968 s
50	55	2.14 s
100	31	4.47 s

We also visualized tracking results to help ourselves during coding. In Fig. II we show two examples of our visualization. In the first image, we can see how we randomly sampled while re-detecing. Blue squares show randomly selected samples with fixed sizes, while red square shows previous detection position and green square represents ground truth. In the second image, we see re-detected target, corresponding ground truth and removal of random sequences.

One of the reasons for relatively low precision, recall and f-score might be that during evaluation we set overlap to 0 if the target is occluded even if our tracker knows that the target is lost. That way we get slightly worse results for sequences where the target is occluded a lot, e.g. cat1 sequence.

III. Conclusion

During this exercise we educated ourselves about short and long-term trackers which are based on deep CNN methods. We successfully modified short-term tracker SiamFC into long-term tracker. We tested different configurations and provided the best ones based on our tests. In terms of sampling methods, we found out that random sampling performs better than uniform method. There is still room for improvement, for example threshold setting, although our tracker did show decent results.