Advanced Methods in Computer Vision
Exercise 5: Long-term Tracking

Dimitar Stefanov, 64170394

I. Introduction

In this work, we analyze the performance of the short-term tracker SiamFC [1] for the purpose of long-term tracking. The performance of the tracker is presented through 3 measures: precision, recall and F-score. As the tracker was not designed for long-term tracking, our main goal was to implement the re-detection stage and consequently boost the results.

II. Experiments

A. Non-modified SiamFC used for long-term tracking

The SiamFC tracker, in its original version, was created to perform short-term tracking. So, first, we wanted to see how it does on a longer sequence where target might also get lost, without any prior improvements to the tracker. The sequence used for this test was **car9**. The results obtained from the tracking were the following:

- *precision* = 0.65
- *recall* = 0.28
- *F-score* = 0.39

It can be immediately noticed that our recall score is rather low. The explanation for this is that once the car went under the sign and disappeared for a short period of time, we were unable to locate it again and were predicting bounding boxes in wrong areas of the image in the continuation. As this target disappearance occured at frame 769 and our sequence is of length 1879, this kind of result for *precision* should not come as a surprise. *F-score* represents a combination of *precision* and *recall*, so the low *recall* value leads also to a lower *F-score*. Overall, it can be said the tracker managed to satisfactory track the object until it vanished for a few frames. Then, it lost the target and never recovered.

B. SiamFC tracker enhanced with re-detection mechanism

In the second stage of testing, we improved our tracker and added a re-detection mechanism to it. Firstly, we needed to search the entire image once the target disappeared. However, the image was quite large compared to our bounding box, so we decided to utilize random uniform sampling and generate 20 possible object positions. Then, we extracted patches at these positions and compared them to the image template. By observing the changes in the confidence score throughout the first test when the tracker didn't include re-detection, we noticed the value of 6 could be an adequate threshold. To further elaborate, the confidence score in that test was initialized to a high value (10000 in our case). Understandably, the score dropped once we started tracking, to a value around 9. This was followed by a gradual decrease to approximately 6, when the score stabilized. However, the moment the target disappeared, we saw a plunge in the confidence measure to roughly 2, and because we didn't recover in the continuation, the score stayed at 2-3. Therefore, we concluded 6 would be a suitable threshold option. So, the combination of random uniform sampling and threshold set at 6 yielded these scores:

- *precision* = 0.63
- *recall* = 0.50
- *F-score* = 0.56

The *precision* score didn't change significantly, but there was a drastic improvement to the *recall* (79 % higher *recall* value). The reason is rather simple - we are able to re-detect the target after it disappears, and continue to track it successfully until the end of the sequence. This also improves the *F-score* and even further supports our decision to have 6 as a threshold value.

C. Analysis of confidence score threshold

The SiamFC tracker calculates a cross-correlation score between the extracted region and the template to determine if the region matches the target. The position of the peak in this cross-correlation map is then set as the new object center. So, we opted to take advantage of this peak value as our confidence score (some additional processing such as upsampling, interpolation, etc. is done, however the essence still remain the cross-correlation values). And, as already explained in the previous section, at first we set the threshold for the confidence score to 6. Then, in search of the optimal value for the threshold, we started lowering it. The performance of the tracker when using these new thresholds is presented in Table I.

Table I
Different threshold values were tested in order to observe their influence on tracker's performance.

Threshold	precision	recall	F-score
5	0.609	0.595	0.602
5.25	0.613	0.565	0.588
5.50	0.615	0.560	0.586
5.75	0.621	0.534	0.574

Higher thresholds led to better *precision*, because in order to say that we have located the target, we need to be more confident in our response, meaning lower chance of misslocating the target and improvement in this metric. On the other hand, by waiting to be certain in our decision, we sometimes don't say we have found the target. So, we actually report a true positive as a false negative, and worsen the *recall* score. Overall, if we were to chose the best threshold value based on the F-score (as done in the VOT challenges), then setting the threshold to 5 would be the correct choice.

D. Influence of number of randomly sampled regions on re-detection capability

In this section, we investigate the relation between the number of randomly sampled regions and the speed of the re-detection process. For this purpose, we checked what happens when we reduce the number of sampled regions (5 or 10 regions) or increase it (30 or 50 regions). A summary of these tests is provided in Table II.

Table II
Total number of frames needed for re-detection throughout the whole sequence for different number of randomly sampled regions.

# of samples	5	10	20	30	50
# of frames till re-detection	42	44	37	49	40

The trend we thought we would discover during these tests was that a higher number of sampled regions helps perform the re-detection faster. Nevertheless, our results show that for any of the chosen numbers of sampled regions we spend around 40 frames for re-detection on the entire sequence. This kind of result can be attributed to the random sampling. In terms of the scores for the metrics we have been considering so far, as expected, in all cases the performance is similar:

- *precision* $= 0.61$
- *recall* $= 0.60$
- *F-score* $= 0.60$

E. Visualization of the re-detection process

In the sequence we chose for testing - **car9**, the target disappears at only point. In Figures 1 and 2, we show the moment we lose the object of interest and the moment we again start tracking it (the situation depicted is when confidence score threshold is set to 5 and we sample 5 random patches).

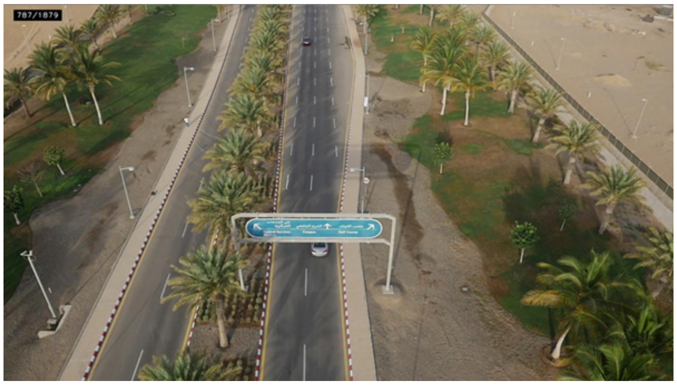

Figure 1. Once the target goes under the sign, the confidence score falls below the threshold (in the particular image it's set to 5), and we lose the object.

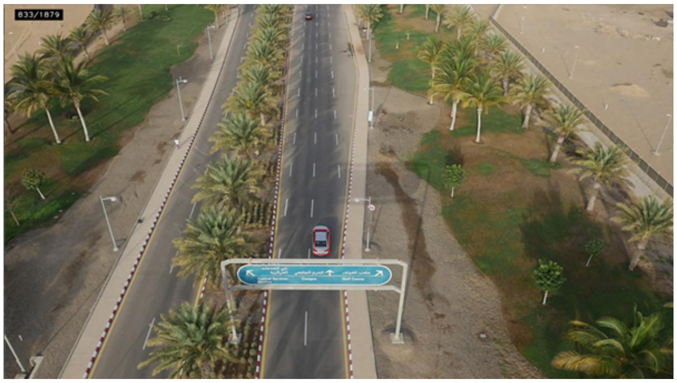

Figure 2. After some period, we managed to again locate the target, and continue to track it successfully from there on.

III. Conclusion

To conclude, we showed that the short-term tracker SiamFC, if enhanced by a re-detection mechanism, can produce satisfactory results in long-term tracking as well. When constructing this mechanism, attention needs to be paid to the confidence score threshold we set, the number of randomly sampled regions, but also to the way we sample. We considered random uniform sampling on the whole image. Another option, which could be quite adequate too, is Gaussian sampling around the last position where we were still tracking the target.

References

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. arXiv preprint:1606.09549, 2016.