



Long-Term Tracking

Advanced Computer Vision Methods Project 5

Visual Cognitive Systems Laboratory,
Faculty of Computer and Information Science,
University of Ljubljana



Short-Term vs. Long-Term Tracking

Short-term:
target visible
all time
(partial occlusion)



Long-term:
target can
disappear

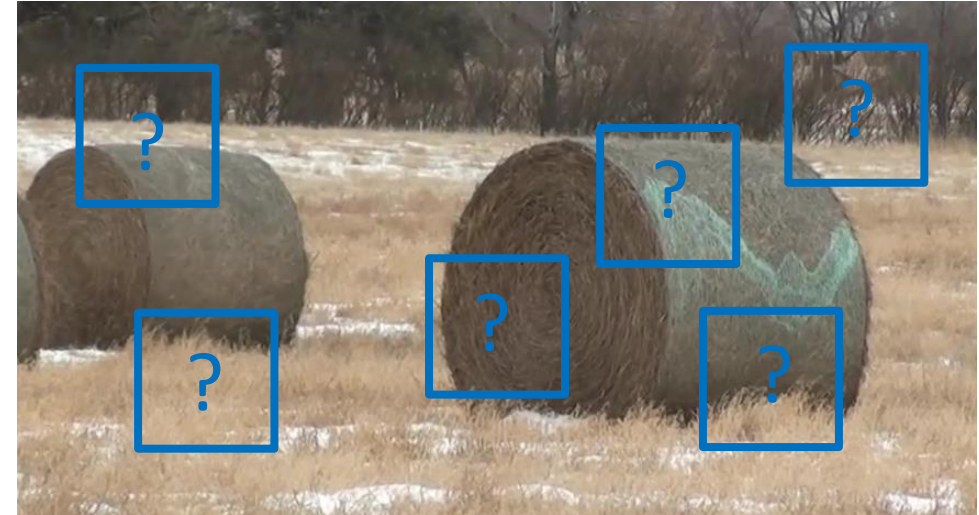


Target not visible



Requirements of Long-Term Trackers

- Know **when** target is not visible
 - Do not update visual model
 - Perform target-redetection
- Target **re-detection**
 - Search for the target on the whole image
(target can re-appear at any position in the image)



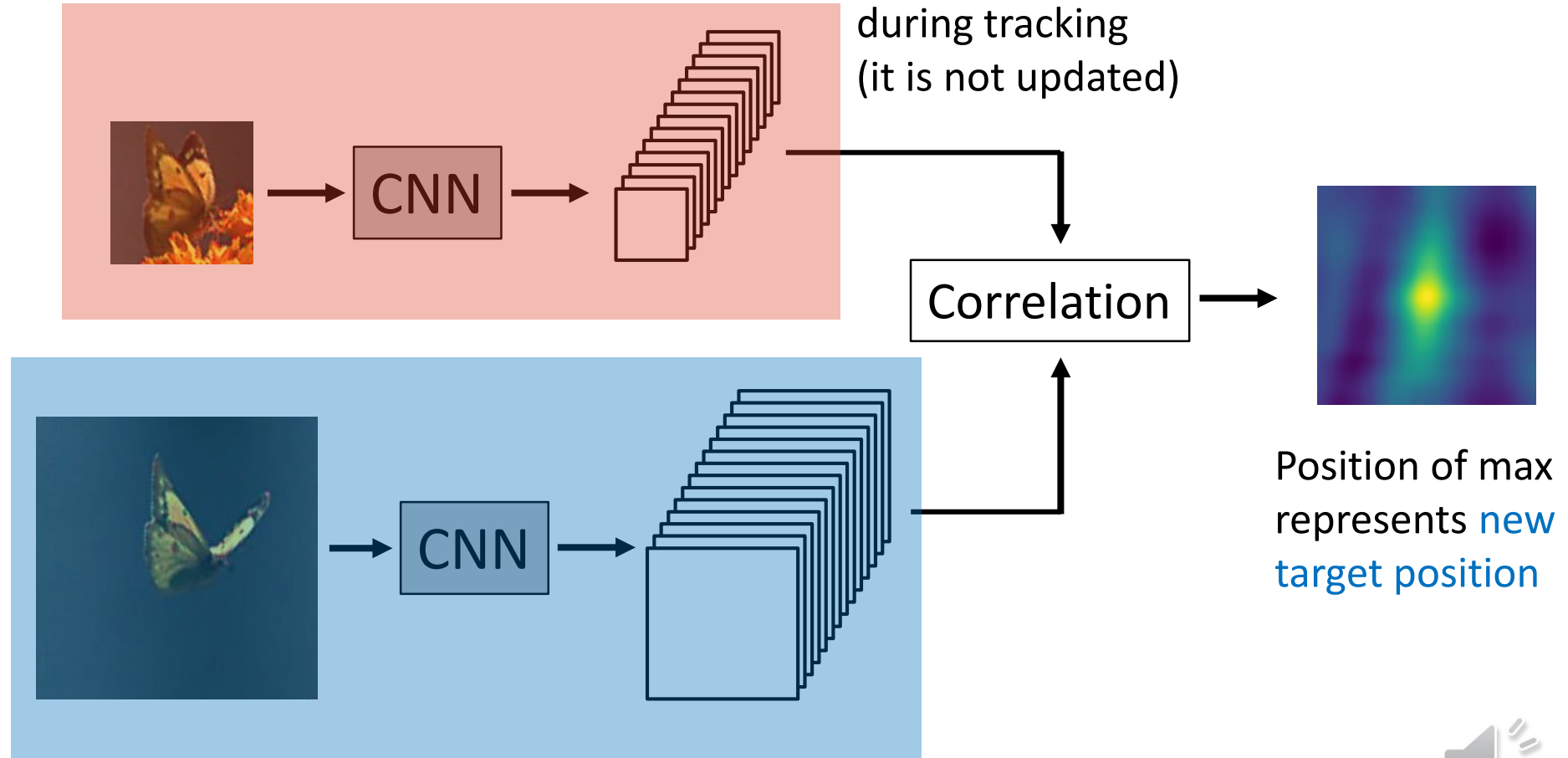
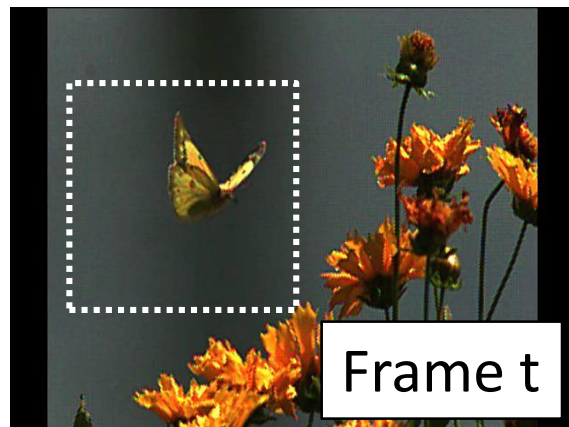
Project 5: Tasks

- Setup the deep CNN-based tracker SiamFC (material on Učilnica)
SiamFC is a short-term tracker
- Run the tracker on the long-term dataset
on at least 1 sequence if you do not have a GPU
- Modify the tracker so that it is long-term
detect when target is not visible and start re-detecting on the whole image
- Report long-term tracking performance of both versions of the tracker



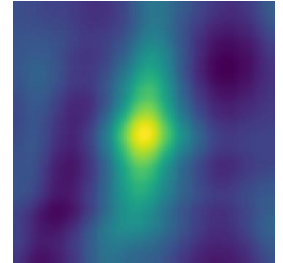
The Short-Term Tracker: SiamFC

- Similar as correlation filter (filter is not trained as DCF, but is just template extracted using deep CNN)



Long-Term Tracking: Failure Detection

- SiamFC localizes the target by correlation
- Position of the maximum represents new position of the target
Value of the maximum can be interpreted as target likelihood (higher the value more likely the region represents target)
- Detect failure by thresholding the value of the maximum in correlation response



Target might not be visible

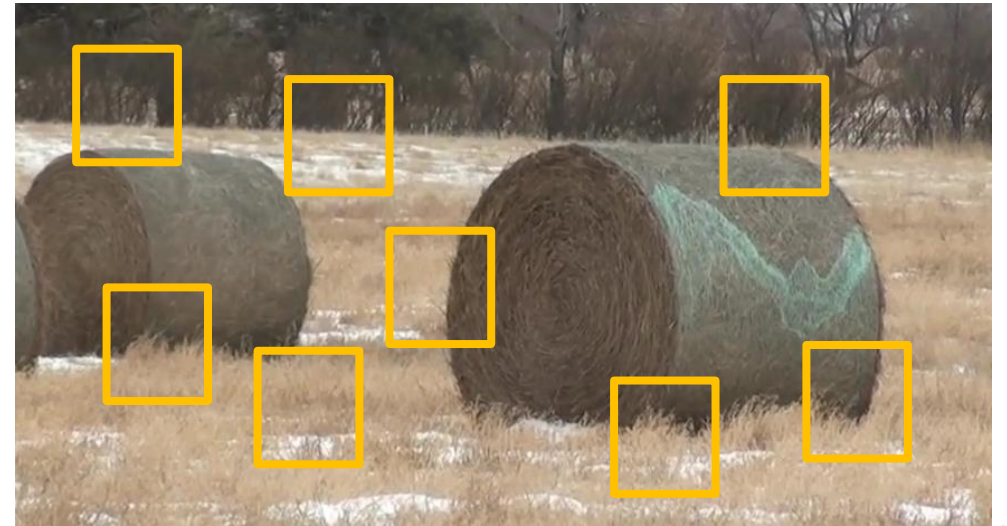


Tracker might fail



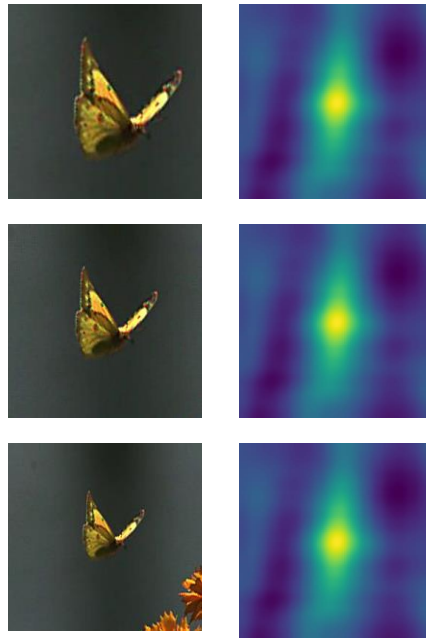
Long-Term Tracking: Target Re-Detection

- When failure is detected
- Search for the target **on the entire image**
- Random sampling (uniform over image, normally distributed around previous location, etc.)
- Report **the most likely position**
- Re-detection is turned off when target is found (using the same mechanism as for failure detection)



Implementation of Target Re-Detection

- Can be efficiently done
- See how localization on the **fixed position** using **multiple scales** is implemented
- Do similar, just with a **fixed scale** and **multiple positions**



Multiple scales on fixed position
(report the most likely)

- When you get new position check if the target is still lost



Long-Term Tracking Performance Evaluation

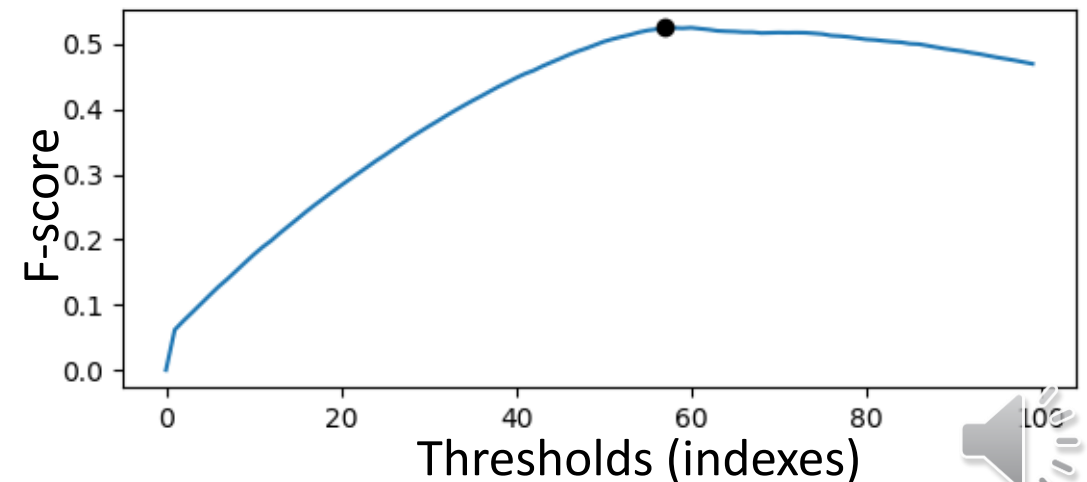
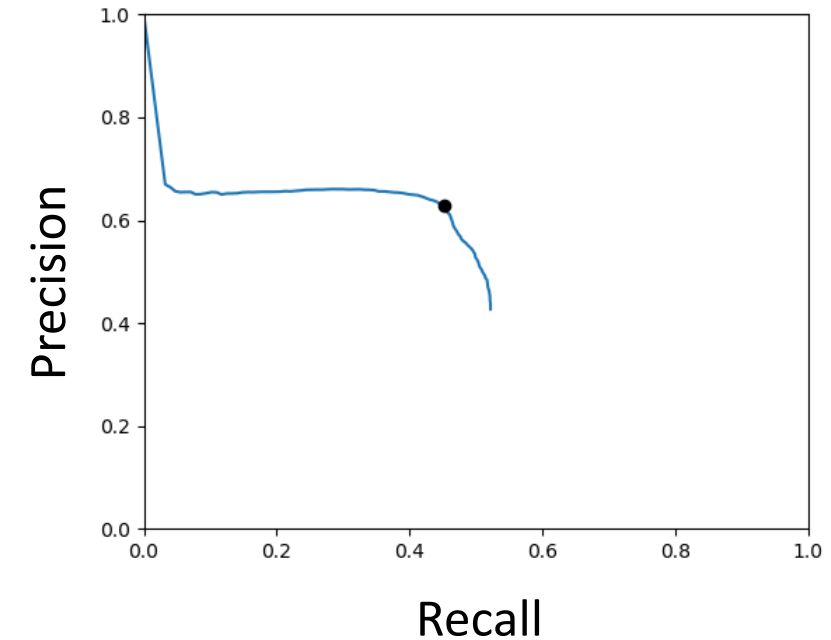
- Tracker reports target **position** (bounding box) and its **confidence** for a given position
- Confidence for SiamFC: maximum of a correlation response
- Performance measures: tracking **precision** and **recall**, tracking **F-measure**
- Calculate measures for different confidence score thresholds
report the optimal one (where F-score is the highest)
- Experiment methodology: tracker is initialized at the beginning of the sequence
and let to track to the end
(tracking failures are not considered and tracker is not re-initialized)



Tracking Precision and Recall

- Given a confidence score threshold:
 - Predictions with lower score are considered as predicted target not visible
- Tracking precision:
average overlap on frames where **predictions are made**
- Tracking recall:
average overlap on frames where **target is visible**
- Tracking F-score:

$$F = \frac{2 \cdot Pr \cdot Re}{Pr + Re}$$



Tracking Precision and Recall: SiamFC

- If you do not have a GPU: evaluation on 9 sequences might take a lot of time – [select one sequence](#)
- Example: sequence car9

	Pr	Re	F-score
SiamFC	0.65	0.27	0.38
SiamFC-LT	0.60	0.59	0.60

On the whole dataset:

	Pr	Re	F-score
SiamFC	0.62	0.32	0.42
SiamFC-LT	0.58	0.42	0.49



Tracking Precision and Recall: SiamFC

- Download the dataset: <http://box.vicos.si/alanl/dataset-lt.zip>
- You can select only one sequence by modifying the file *list.txt* in dataset folder
- Download the pre-trained network: http://box.vicos.si/alanl/siamfc_net.pth
- Source code for tracker and evaluation scripts are on Ucilnica
- Run tracker on a dataset:

```
python run_tracker.py --dataset <path/to/dataset> --net <path/to/network> --results_dir <path/to/results/directory>
```
- Run evaluation:

```
python performance_evaluation.py --dataset <path/to/dataset> --results_dir <path/to/results/directory>
```

