Supplementary Materials for Paper: Systematized Event-Aware Learning for Multi-Object Tracking

Hyemin Lee[1] **Daijin Kim**[1]

[1]Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, Korea

This supplementary material contains more detailed illustration of methods, implementation details and the network architecture used in proposed method. In addition, we provide additional experiments based on various affinity function.

1 EXAMPLE OF TARGET STATE TABLE FOR EACH TARGETS

Figure 1 shows the whole possible case of target state and corresponding event categorization. Figure 2 shows example of target state for the ground-truth bounding boxes and the corresponding detection boxes.

2 IMPLEMENTATION DETAILS

For reproducibility, here we provide additional implementation details including network architecture and hyperparameters.

The proposed method was implemented using PyTorch 1.3 and tested on a six-core Intel i7@3.60 GHz CPU and NVIDIA Titan Xp GPU environment. The training requires 4.6GB GPU memory storage and takes about 22-hours for 40 training epochs.

2.1 NETWORK ARCHITECTURE

We used an R-FCN architecture with SqueezeNet as the backbone network for the MOTDT baseline and used the Faster R-CNN detector with ResNet-101 and feature pyramid networks (FPNs) Lin et al. [2017] as the backbone network for the Tracktor baseline. The classification threshold for target initialization was set to 0.3, and the maximum loss time for termination was set to 30 frames. The association network for MOTDT uses GoogLeNet for the association features, and the association network for the Tracktor baseline was implemented based on the Siamese CNN architecture trained on TriNet Hermans et al. [2017] using ResNet-50. We followed the same tracking management strategy baseline tracker excluding the association and training steps. The minimum threshold value for the filtering candidates was set to 0.4. We kill the missing target after 30 frames, without associating with any candidates. The network was trained using stochastic gradient descent over 40 epochs with learning rates of ranging from 10^{-3} to 10^{-5}. The detailed network architectures are illustrated on Figure 3.

2.2 TRACKING MANAGEMENT

We followed the same tracking management strategy baseline tracker excluding the association and training step. The minimum threshold value for filtering candidates was set to 0.4. We limited the possible change in the location to 1/10 of the diagonal length of the frame, and the possible size change as 1/3 of the previous target size. We kill the missing target after 30 frames, without associating with any candidates.

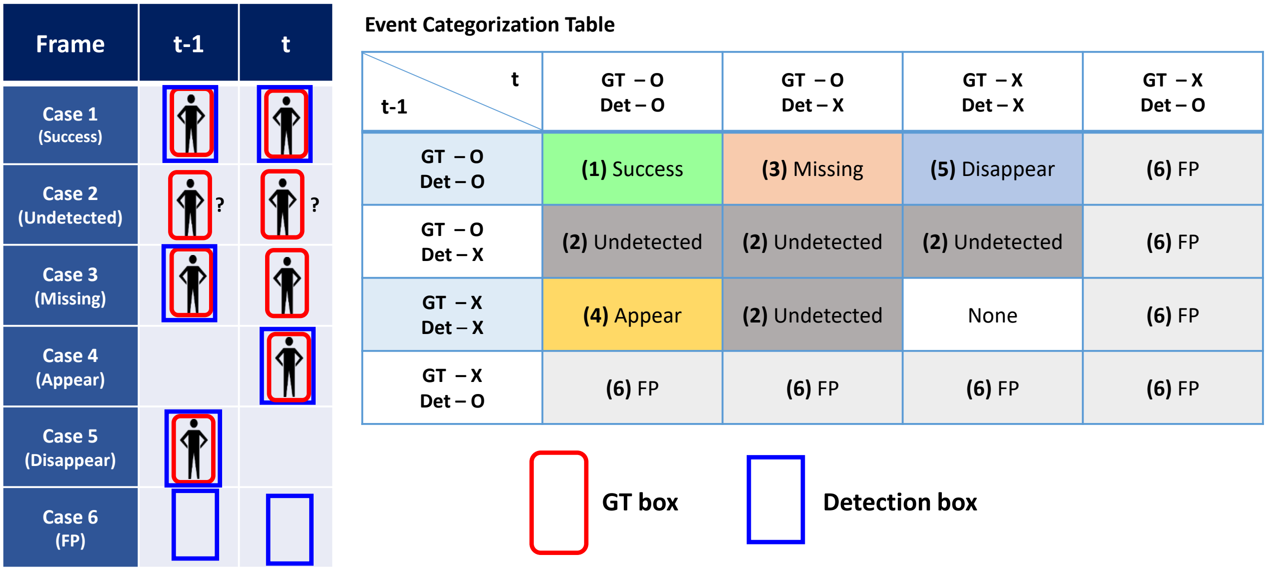
2.3 TRAINING

The network was trained using stochastic gradient descent over 40 epochs with learning rates of ranging from 10^{-3} to 10^{-5}. We generated training samples from the 2D MOT2015 and MOT2017 training sets and split that into 7-fold to train the network. We randomly selected two consecutive frames, and run the simulation trackers to generate samples. We fixed the network weight of DHN and re-identification layer of baseline tracker. We set the α for event-aware loss as 0.5, and β as 2.

3 EXPERIMENTS

In this section, we provide additional experimental results including evaluation on MOT2020 dataset and test results on various affinity function and loss hyperparameters.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).



Event Categorization Table

t-1 \ t	GT – O Det – O	GT – O Det – X	GT – X Det – X	GT – X Det – O
GT – O Det – O	(1) Success	(3) Missing	(5) Disappear	(6) FP
GT – O Det – X	(2) Undetected	(2) Undetected	(2) Undetected	(6) FP
GT – X Det – X	(4) Appear	(2) Undetected	None	(6) FP
GT – X Det – O	(6) FP	(6) FP	(6) FP	(6) FP

Figure 1: Process of assigning event for each detection box by associating with ground-truth boxes. If the ground-truth box is undetected in the previous frame, we do not care about that box. If the box is detected in the previous frame and undetected in the current frame, it is considered a missing target. If the target appears in the current frame, it is considered a new target appearance. The disappeared target in ground-truth at current frame is considered a disappeared target. The remaining detections that are not associated whether in previous and current frame are treated as false positives (FPs).

3.1 EVALUATION ON MOT2020 DATASET

he MOT2020 test dataset contains four test sequences, including densely crowded scene. The results obtained for the MOT2020 test dataset are reported in Table 1.

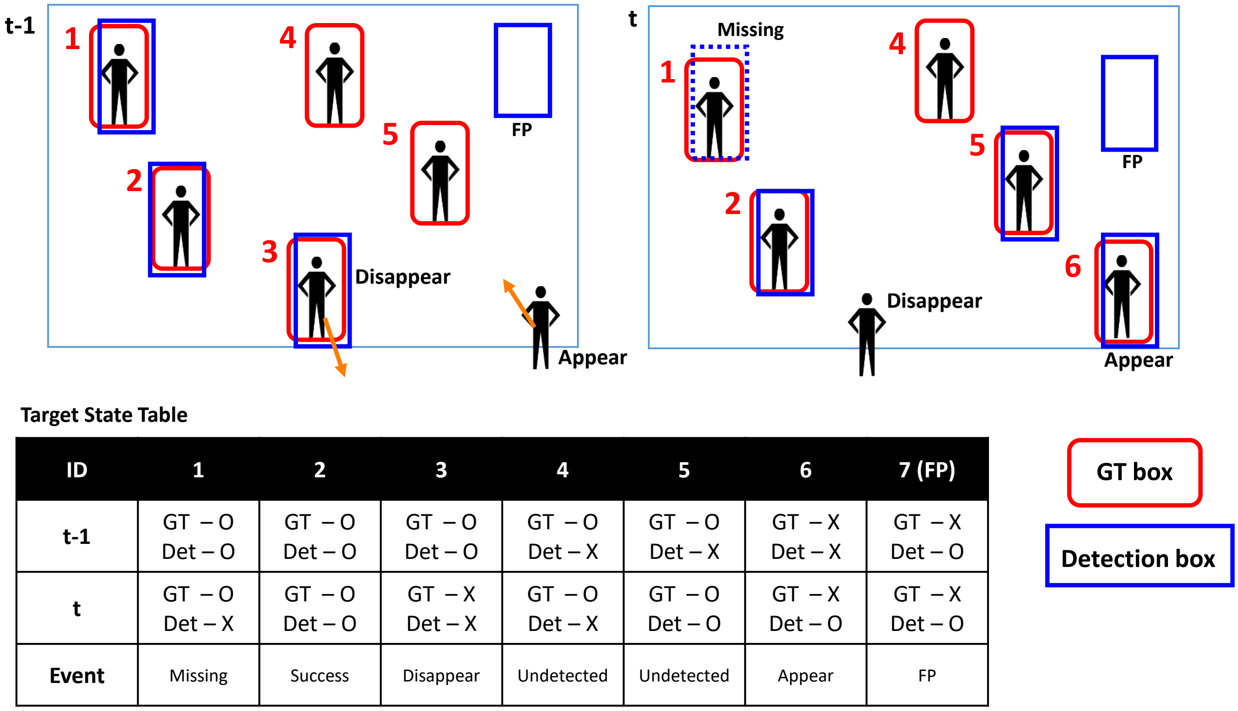
3.2 TEST RESULTS OF VARIOUS LOSS WEIGHTING FACTOR

We trained proposed network using various loss weighting factors. The results show that how each weighting factor affect to each evaluation term, especially on FP and FN term. In the results, we can see the balance of weight makes good results and the FN factor have higher impact compared with FP factor. The results are shown on Table 2.

References

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.



Target State Table

ID	1	2	3	4	5	6	7 (FP)
t-1	GT – O Det – O	GT – O Det – O	GT – O Det – O	GT – O Det – X	GT – O Det – X	GT – X Det – X	GT – X Det – O
t	GT – O Det – X	GT – O Det – O	GT – X Det – X	GT – O Det – X	GT – O Det – O	GT – X Det – O	GT – X Det – O
Event	Missing	Success	Disappear	Undetected	Undetected	Appear	FP

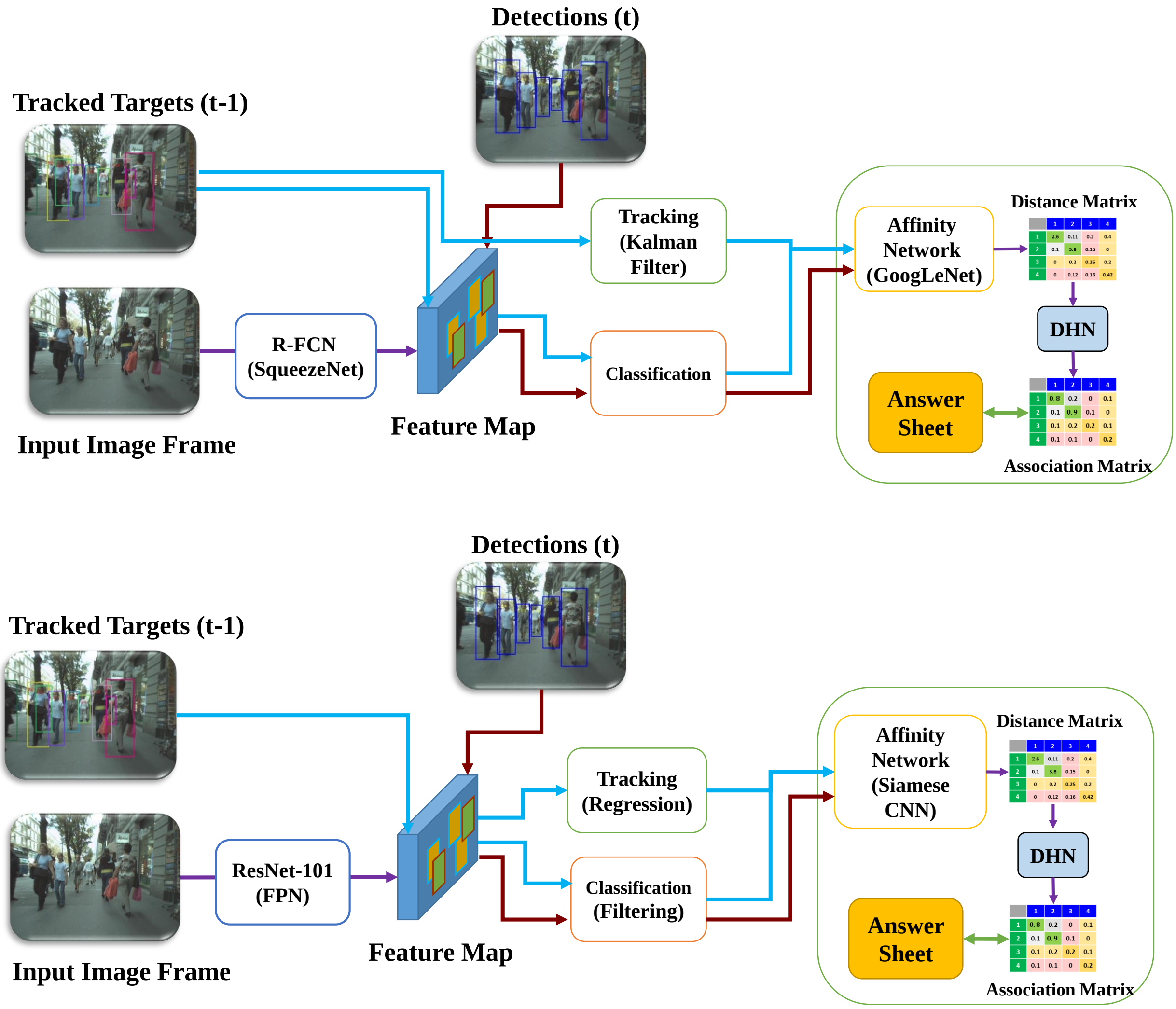
Figure 2: Example of target state table for each targets.

Table 1: Tracking Performance on the MOT2020 benchmark test set. Best in bold.

Method	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
GNNMatch	54.5	49.0	**32.8**	25.5	9522	223611	2038
Tracktor++v2	52.6	52.7	29.4	26.7	**6930**	236680	**1648**
SEAT (Tracktor++v2)	**54.9**	**51.3**	32.2	**24.1**	8509	**223105**	1877

Table 2: Training results of various loss weighting factor.

Loss weight	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
$\alpha = 0, \beta = 0$	67.8	66.0	43.7	**16.1**	2771	32955	425
$\alpha = 0.5, \beta = 0$	68.1	68.1	42.3	17.4	1007	34432	371
$\alpha = 1, \beta = 0$	67.6	69.0	39.9	17.0	**715**	35295	337
$\alpha = 0, \beta = 0.5$	68.7	69.5	42.8	17.5	2037	32818	425
$\alpha = 0, \beta = 1$	68.9	70.2	45.6	16.8	1818	32707	379
$\alpha = 0, \beta = 2$	69.0	70.1	45.0	16.7	1904	32522	305
$\alpha = 0.5, \beta = 0.5$	69.2	69.9	45.2	16.3	1757	32563	388
$\alpha = 0.5, \beta = 2$	**69.3**	**71.8**	**45.6**	16.5	2108	**32078**	**297**



Distance Matrix

	1	2	3	4
1	2.6	0.11	0.2	0.4
2	0.1	3.8	0.15	0
3	0	0.2	0.25	0.2
4	0	0.12	0.16	0.42

Association Matrix

	1	2	3	4
1	0.8	0.2	0	0.1
2	0.1	0.9	0.1	0
3	0.1	0.2	0.2	0.1
4	0.1	0.1	0	0.2

Figure 3: Network architectures of proposed methods.