Embedded Pedestrian Tracking and Detection

MSCV19 Capstone Project, Internal(CMU)

Team Member: Yongxin Wang, Chunhui Liu
Advisor: Professor Kris Kitani

12/06/2019

Introduction

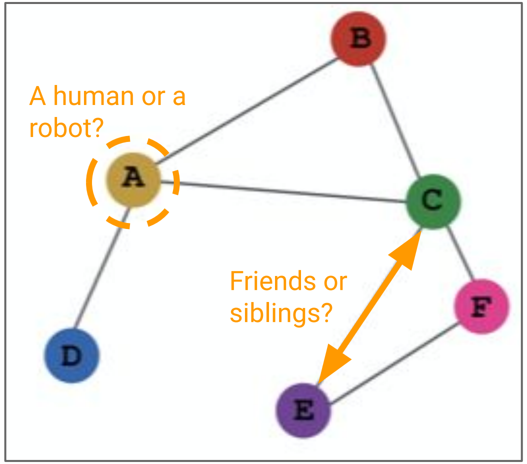
- Problem
  - Multi-target Pedestrian Tracking
- Challenge
  - Accurate detection and association at the same time
- Our take
  - One stage network
  - Graph Neural Network (GNN) for simultaneous detection and association
  - Non-Maximum Suppression specifically tailored for the tracking task

Simultaneous Detection and Tracking with Graph Neural Network

Brief Introduction to GNN

[1] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling, “Modeling Relational Data with Graph Convolutional Networks”, ESWC 2018 (Best Student Research Paper)

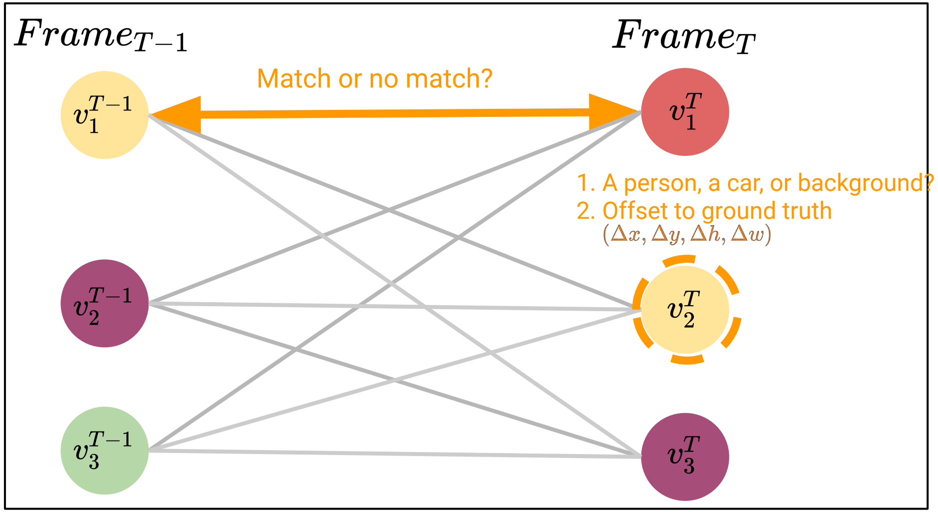
- **Aggregate** node features based on local neighborhood
  - Features can be used for downstream tasks, i.e. node classification [1]
  - Details about how GNN works can be found in Appendix II



An online social network represented by a graph. Each node denotes a feature of an entity within the social network. Each edge denotes the relationship features between entities.

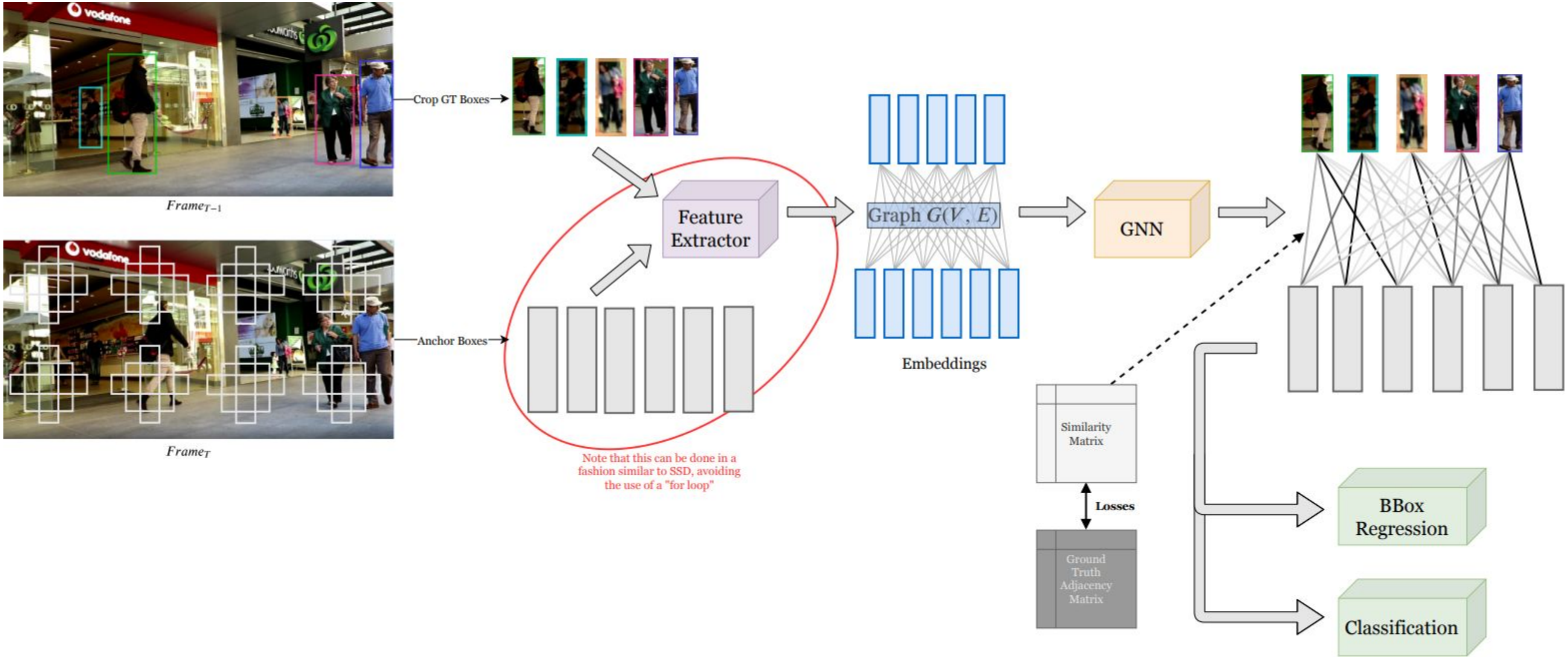
Intuition

- Association can be naturally formulated as a bipartite graph matching
- GNN clusters similar nodes closer together than dissimilar ones
  - Same identities can be clustered closer together across two frames.
- We took a step further
  - GNN for both association and detection

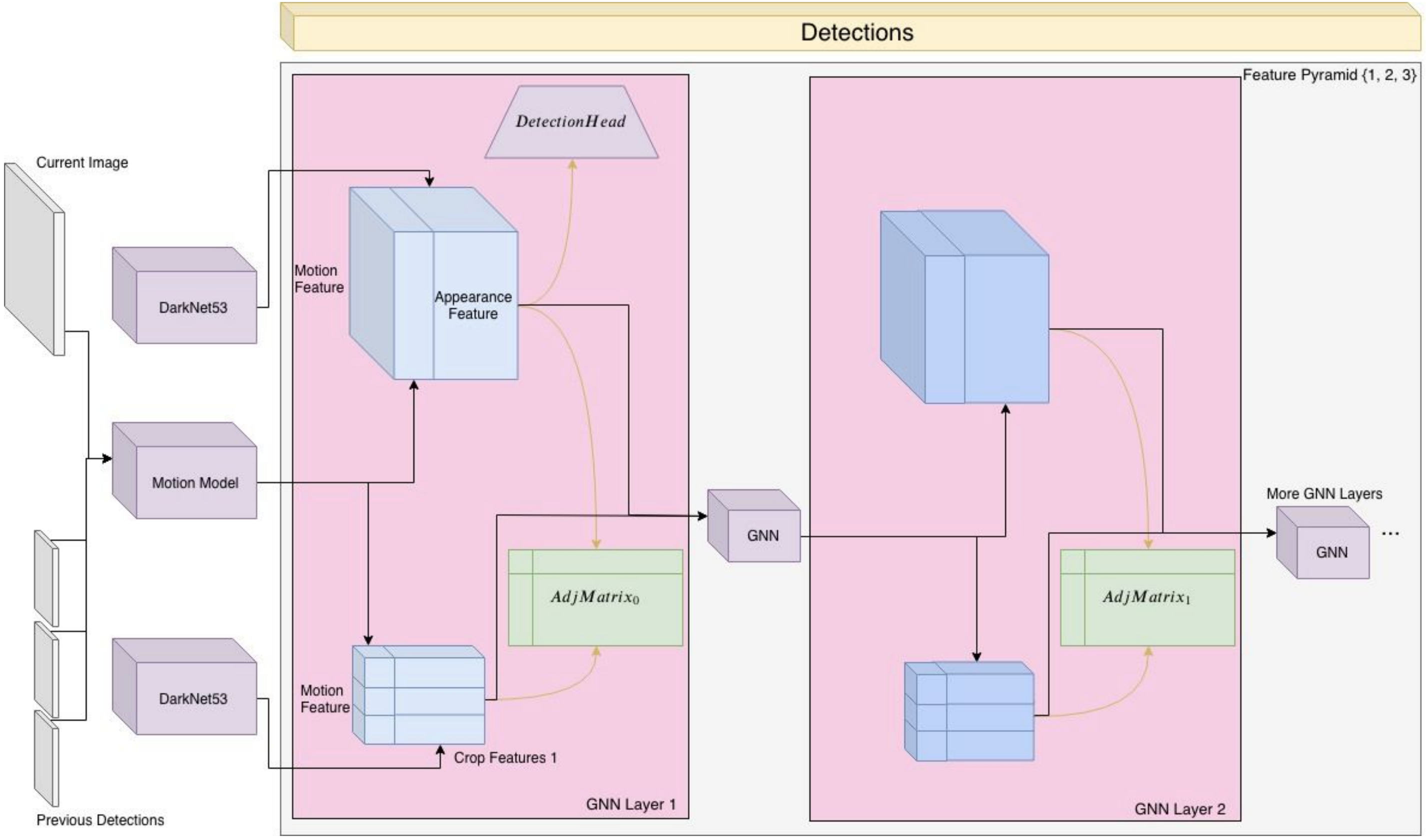


Data association in MOT formulated as graph association. A Node denotes features of a detection (if in $Frame_{T-1}$) or of an anchor box (if in $Frame_T$). An edge denotes relationship features between detections and anchor boxes.

Past: Initial Idea



Present: Our Implementation (YOLOv3 + GCN)



Present: Results on MOT Benchmark

- Validation Performance on MOT15 Benchmark
  - Considering all ground truth pedestrians
  - 0.2 below the SOTA

Tracker	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
Youtu_face [1]	**56.8**	**59.4**	**43.30%**	**21.20%**	**6,452**	**19,642**	**459**
Ours	56.6	50.7	52.13%	18.26%	6,520	7,756	620
H1_SJTUZTE [2]	56.6	57	39.90%	23.90%	7,198	18,926	533
RAR15 [3]	56.5	61.3	45.10%	14.60%	9,386	16,921	428
TRID [4]	55.7	61	40.60%	25.80%	6,273	20,611	351
NOMTwSDP [5]	55.5	59.1	39.00%	25.80%	5,594	21,322	427

[1] A Real-time Deep Graph Matching for Multi-object Tracking. In Tech Report, Tencent, 2018.
[2] W. Lin, J. Peng, S. Deng, M. Liu, X. Jia, H. Xiong. Real-time multi-object tracking with hyper-plane matching (v1). In Tech Report, Shanghai Jiao Tong University & ZTE Corp, 2017.
[3] K. Fang, Y. Xiang, X. Li, S. Savarese. Recurrent Autoregressive Networks for Online Multi-Object Tracking. In The IEEE Winter Conference on Applications of Computer Vision (WACV), 2018.
[4] S. Manen, M. Gygli, D. Dai, L. Van Gool. PathTrack: Fast Trajectory Annotation with Path Supervision. In ArXiv e-prints, 2017.
[5] W. Choi. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor. In ICCV, 2015.

Present: Results on MOT Benchmark

- Validation performance on MOT17 Benchmark
  - Still some space for improvement
  - The major drop in performance lies in the small objects

Tracker	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
PT17	**66.9**	**66.6**	**36.80%**	**21.30%**	**32,502**	**150,750**	**3,567**
MTGCN	63.9	55.4	33.40%	19.60%	30,423	169,755	3,747
SST	52.4	49.5	21.40%	30.70%	25,423	234,592	8,431
Deep_Track	52.3	47.3	19.70%	36.10%	16,981	246,393	5,573
YOSEMITE	50.9	56	18.90%	33.80%	25,295	249,365	2,397
Ours	**48.7**	**36.5**	**22.39%**	**30.60%**	**2,361**	**17,050**	**176**
AEb_Exp_6	48.1	45.9	18.10%	39.50%	17,371	273,117	2,352
AEb_Exp_4	38.6	39.3	14.80%	46.40%	16,841	327,217	2,206

Present: Results on MOT Benchmark

- Importance of GCN

Backbone	Model	Test set	Matching	MOTA ↑	MOTP ↑	IDF1 ↑	ID Sw. ↓	MT ↑	ML ↓	FP ↓	FN ↓
DarkNet53	YoloV3 + GCN	MOT15	Adjacency matrix **before** GCN	15	79.3	17.3	16	5.03%	70.85%	397	12476
DarkNet53	YoloV3 + GCN	MOT15	Adjacency matrix **after** GCN	33.4	79.6	29.9	228	17.59%	36.68%	1464	8468

- Importance of other components
  - Loss reweighting + Motion module

Backbone	Model	Dataset	MOTA ↑	MOTP ↑	IDF1 ↑	ID Sw. ↓	MT ↑	ML ↓	FP ↓	FN ↓
DarkNet53	YoloV3 + GCN	MOT17	19.6	82.4	18.2	354	1.87%	64.18%	692	30513
DarkNet53	YoloV3 + GCN + ReWeight	MOT17	22.3	81.9	18.6	428	2.99%	59.33%	762	29283
DarkNet53	YoloV3 + GCN + Motion	MOT17	38.1	82.4	29.6	573	13.43%	47.39%	1561	22139
DarkNet53	**YoloV3 + GCN + Motion + ReWeight**	MOT17	**48.7**	**82**	**36.5**	**176**	**22.39%**	**30.60%**	**2361**	**17050**

Present: Results on MOT Benchmark

Future: Improvements and directions

- Data augmentation to compensate small objects
- Consider a temporal GNN that takes in multiple frames (>2)
- Preparing Arxiv paper

Tracking NMS

Non-Maximum Suppression algorithm designed for tracking task, instead of detection task.

Motivation

Baseline and Proposed Methods

Experiments

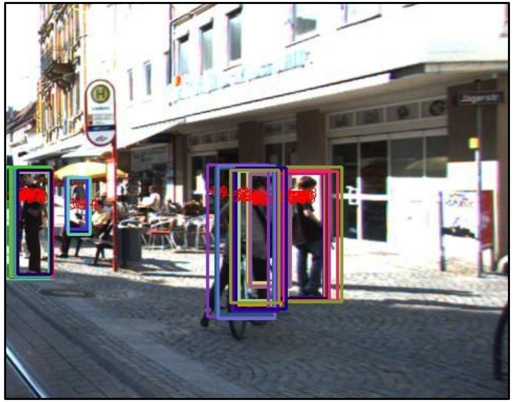
Motivation

Tracking task and Detection task hold different assumptions.

- For detection, we assume that bounding boxes are not overlapped with each other.
- For tracking, two person can be overlapped with each other.
- For tracking, we also have previous tracking results as the prior knowledge.

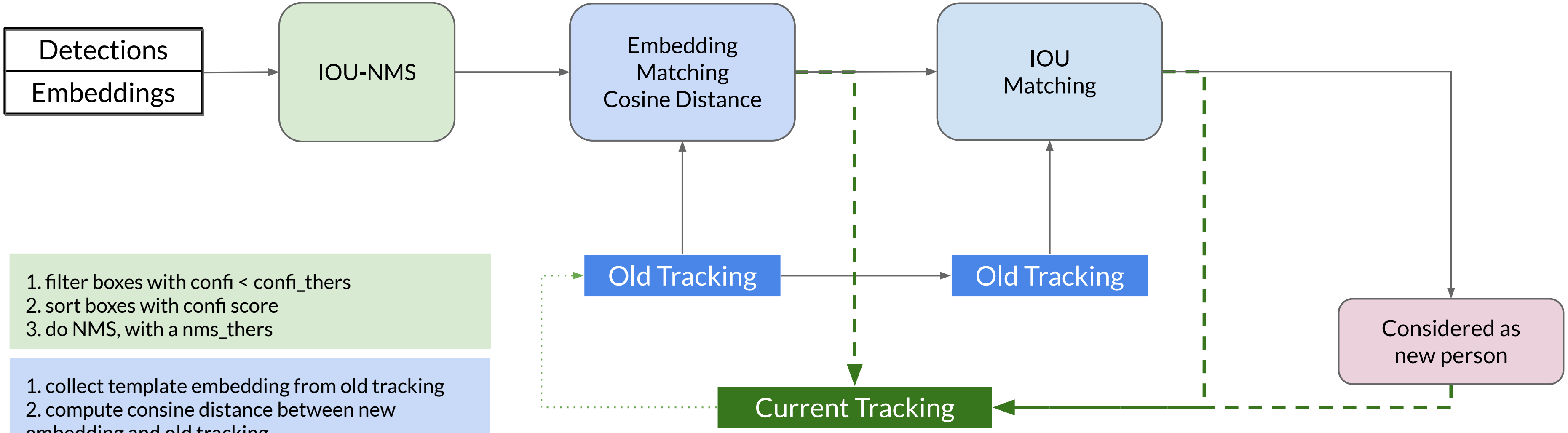
Tracking-NMS: Refind some good candidates and make exceptions.

NMS (0.5+0.5)



All Candidates

Baseline Pipeline: JDE Tracker[1]



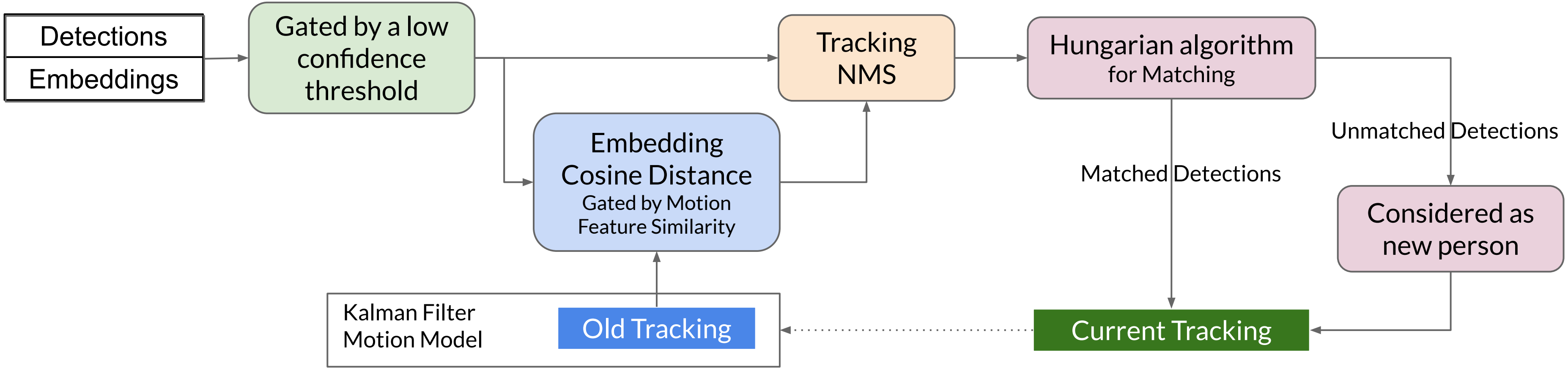
1. filter boxes with confi < confi_thers
2. sort boxes with confi score
3. do NMS, with a nms_thers

1. collect template embedding from old tracking
2. compute consine distance between new embedding and old tracking
3. match using Hungarian method

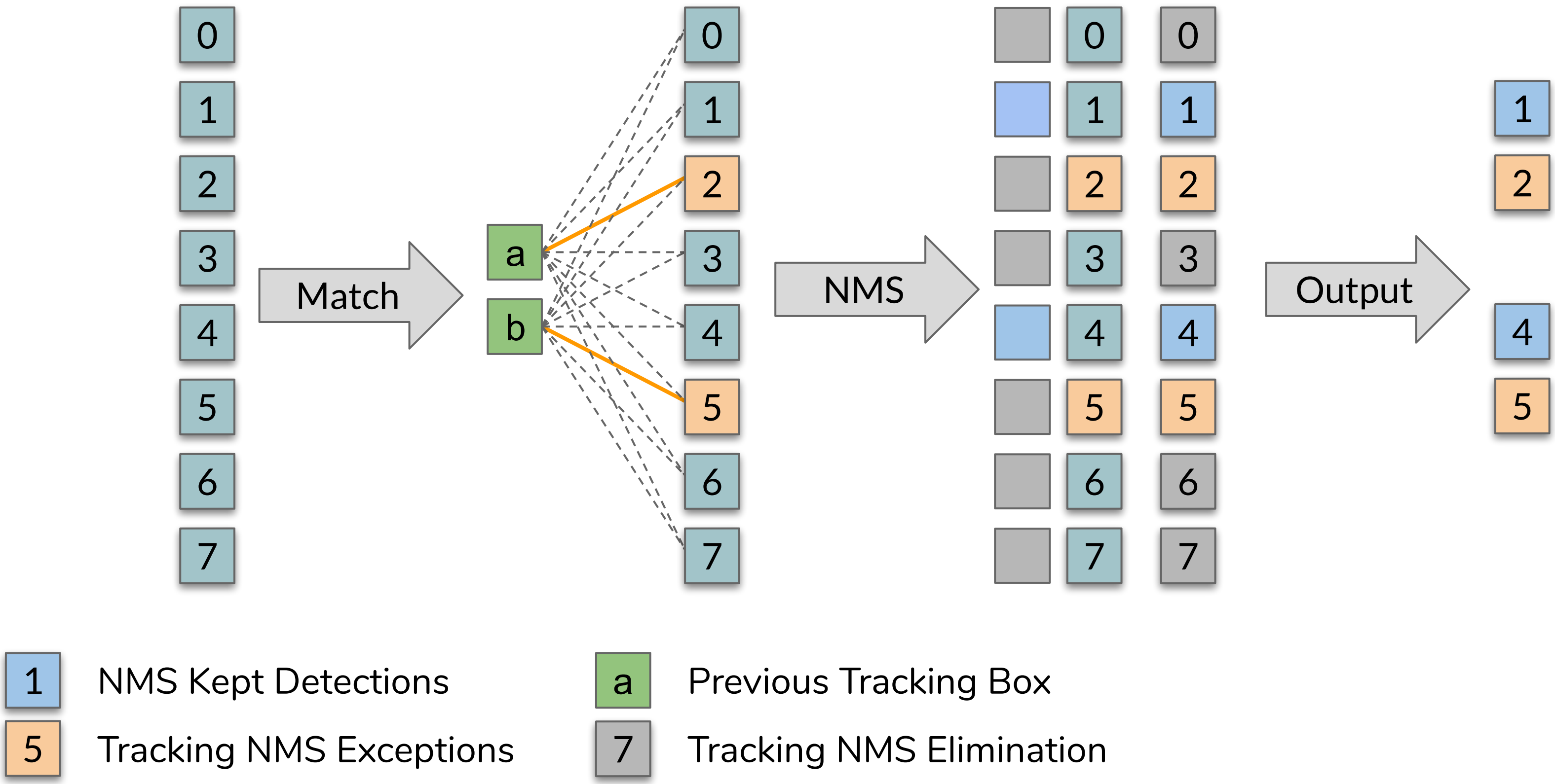
1. compute IOU distance between unmatched detections and unmatched tracking
3. match using Hungarian method

[1] Wang, Zhongdao, et al. "Towards Real-Time Multi-Object Tracking." *arXiv preprint arXiv:1909.12605* (2019).

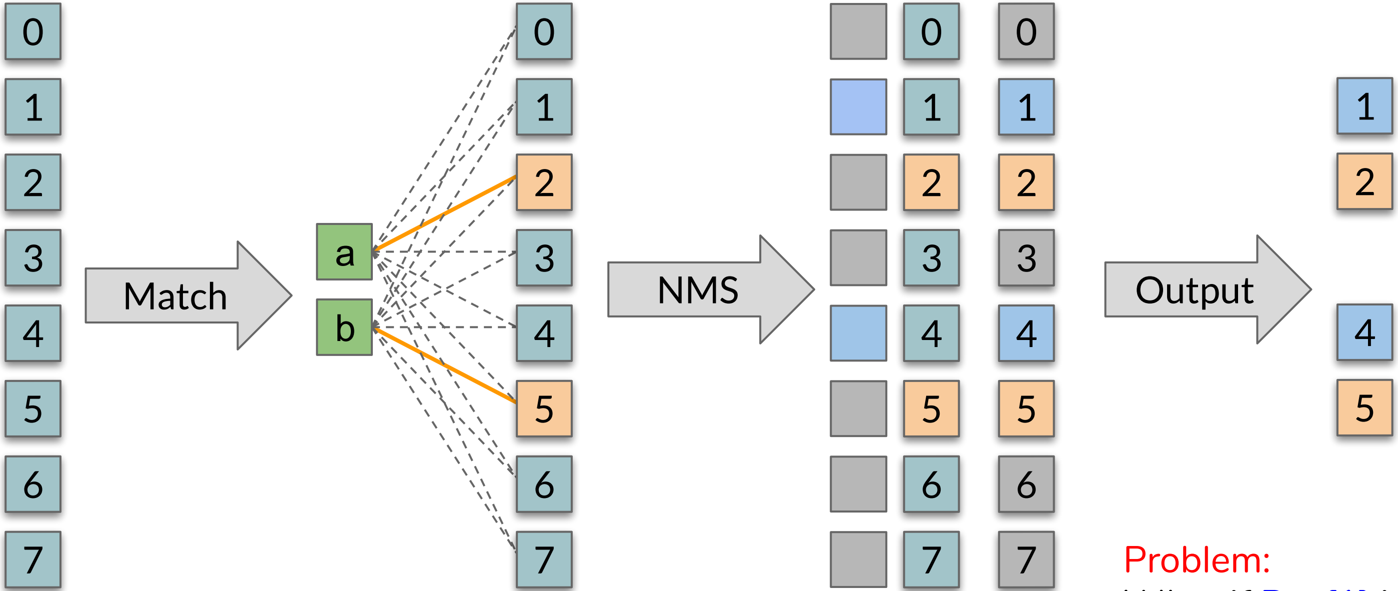
Proposed Pipeline with Tracking NMS



Tracking NMS: Version 1



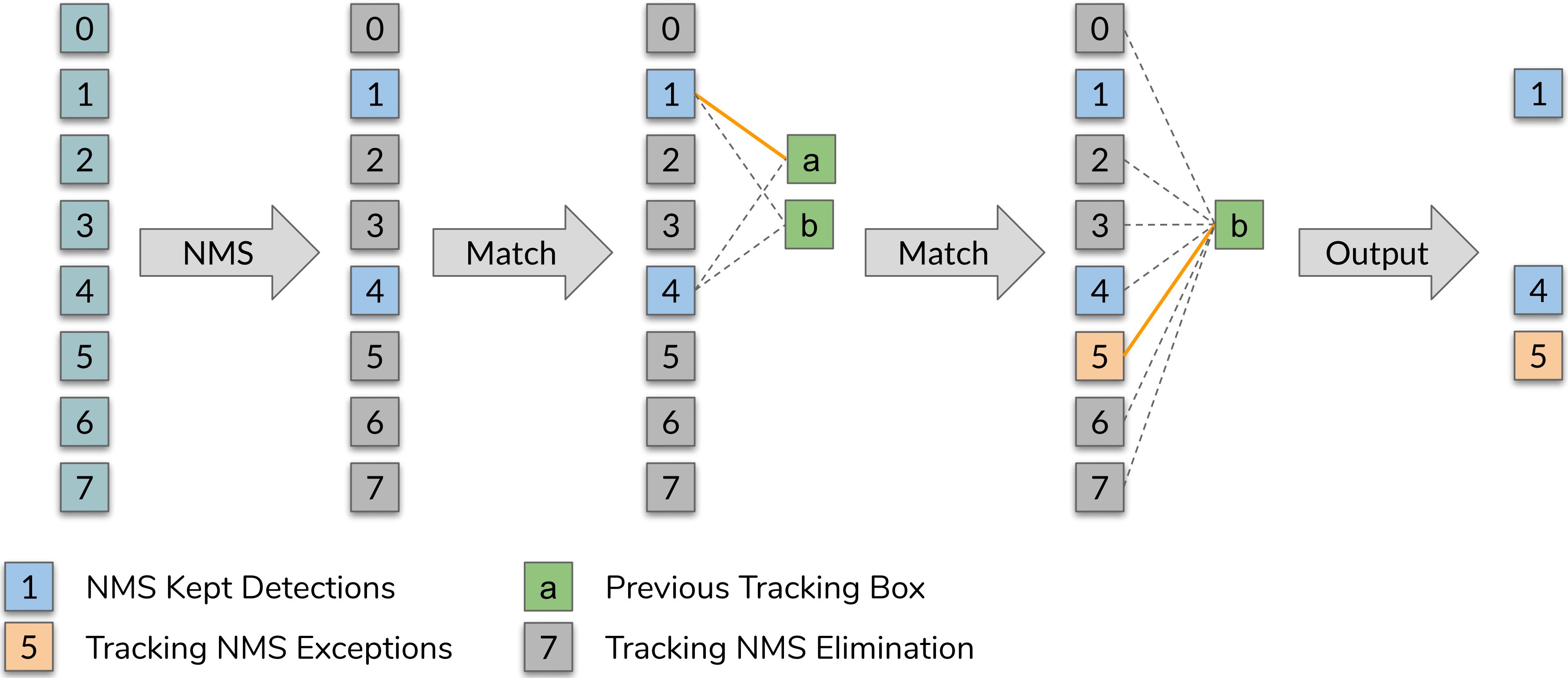
Tracking NMS: Version 1



Problem:
What if Box[1] is also a good matching for tracking[a]

1	NMS Kept Detections	a	Previous Tracking Box
5	Tracking NMS Exceptions	7	Tracking NMS Elimination

Tracking NMS: Final Version

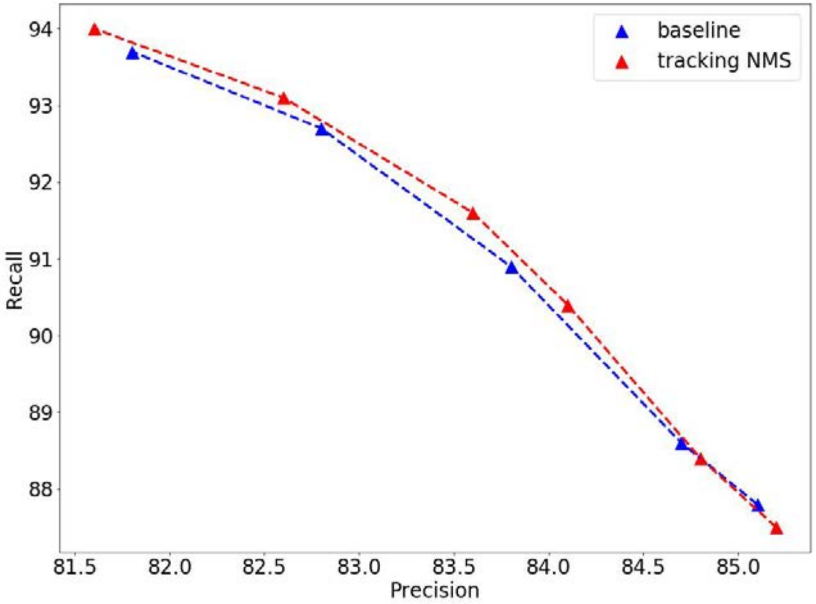


Experiments 1/3: Ablation Study

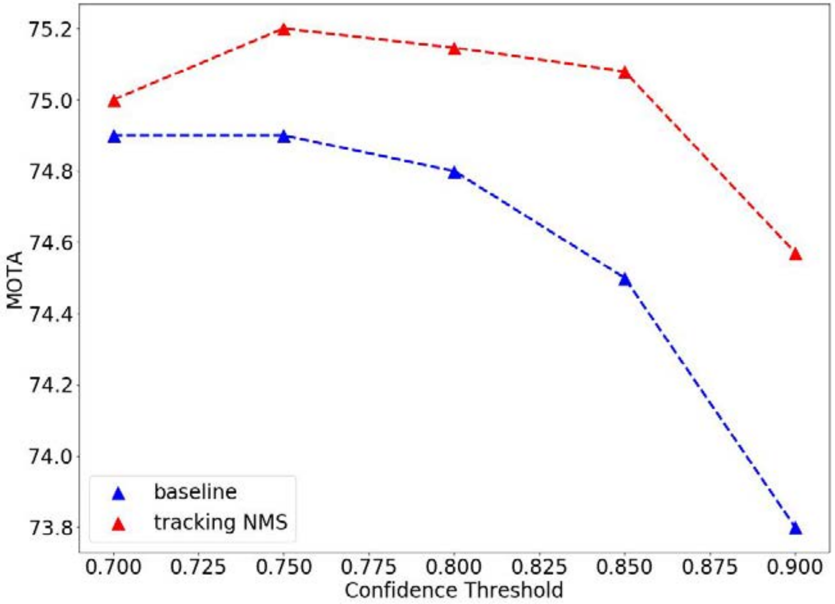
Methods	MOTA ↑	IDF1 ↑	IDP ↑	IDR ↑	Box Recall ↑	Box Precision ↑	FP↓	FN↓	ID Switches↓	Fragmentations ↓
JDE Tracker Official Code	74.33%	67.00%	72.73%	62.04%	80.39%	94.25%	5504	22016	1303	2385
Tracking NMS w/o Motion Model	74.950%	66.641%	71.948%	62.063%	81.249%	94.189%	5629	21057	1445	2600
Tracking NMS Version 1	74.945%	66.648%	71.906%	62.107%	81.289%	94.114%	5709	21012	1415	2586
Tracking NMS with Multi Candidates	74.655%	66.608%	71.075%	62.670%	82.077%	93.086%	6846	20127	1489	2295
Tracking NMS Final	**75.166%**	66.736%	71.371%	62.666%	82.126%	93.535%	6375	20072	1441	2294

MOTA: Multiple Object Tracking Accuracy. This measure combines three error sources: false positives, missed targets and identity switches.

Experiments 2/3: Curves



Bounding Box Precision Recall Curve



Final MOTA Comparison

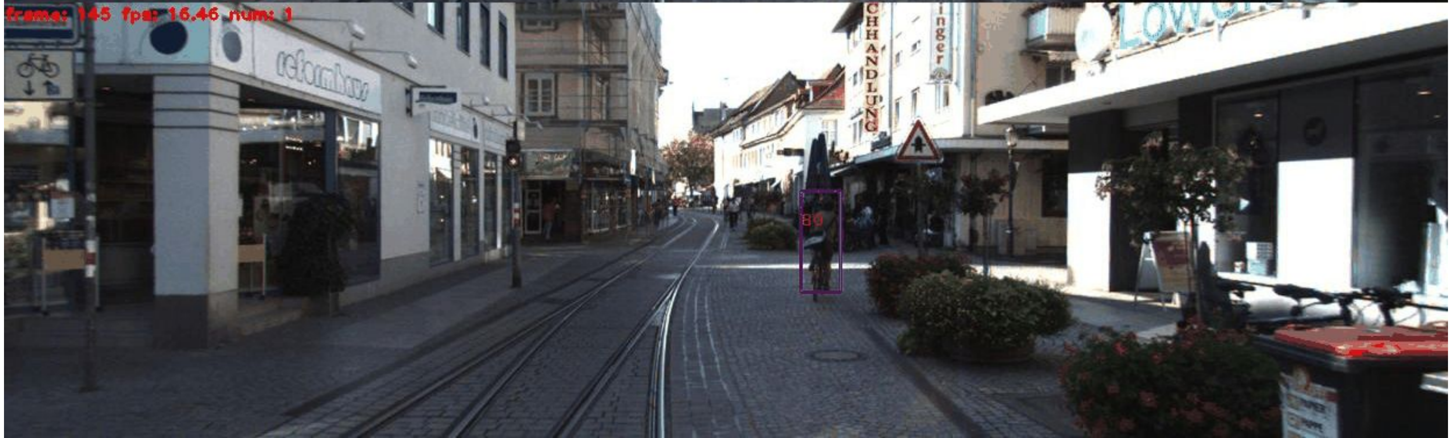
Experiments 3/3: Final Comparison

Methods	Dataset	MOTA ↑	IDF1 ↑	IDP ↑	IDR ↑	Box Recall ↑	Box Precision ↑	FP↓	FN↓	ID Switches↓	Fragmentations↓
JDE Tracker Official Code	MOT15 Train	67.98%	76.75%	73.84%	79.90%	88.56%	81.84%	1548	901	73	160
Tracking NMS Final	MOT15 Train	**72.64%**	79.35%	78.35%	80.37%	88.12%	85.90%	1139	936	80	160
JDE Tracker Official Code	MOT17 Train	74.33%	67.00%	72.73%	62.04%	80.39%	94.25%	5504	22016	1303	2385
Tracking NMS Final	MOT17 Train	**75.17%**	66.74%	71.37%	62.67%	82.13%	93.53%	6375	20072	1441	2294

More Examples

Baseline

Tracking NMS



Thank you
Q&A