Object Tracking

Tomasz Stańczyk
tomasz.stanczyk@inria.fr

With inspiration and some images from the lecture of Prof. Dr. Laura Leal-Taixé: https://www.youtube.com/watch?v=mrMspzKcOAM

3iA Côte d'Azur
Interdisciplinary Institute for Artificial Intelligence

Two types of tracking discussed today

Visual-object tracking - briefly

Multi-object tracking - major part of the lecture

Visual-object tracking (single-object tracking)

Given a video, find out which parts of the image depict the same object in different frames

To give you an idea: https://www.youtube.com/watch?v=IqMgsiU9B5E

It can be applied in surveillance (tracking the target of interest), observation applications (e.g. animals) and so on

Correlation tracker

Implemented in dlib library: http://dlib.net/

Easy to install and run

Previous video - that was the correlation tracker!

Curious? Try it yourself:
http://dlib.net/correlation_tracker.py.html

All instructions provided in the file

Based on the paper: "Accurate Scale Estimation for Robust Visual Tracking"
http://www.bmva.org/bmvc/2014/files/paper038.pdf

Multi-object tracking

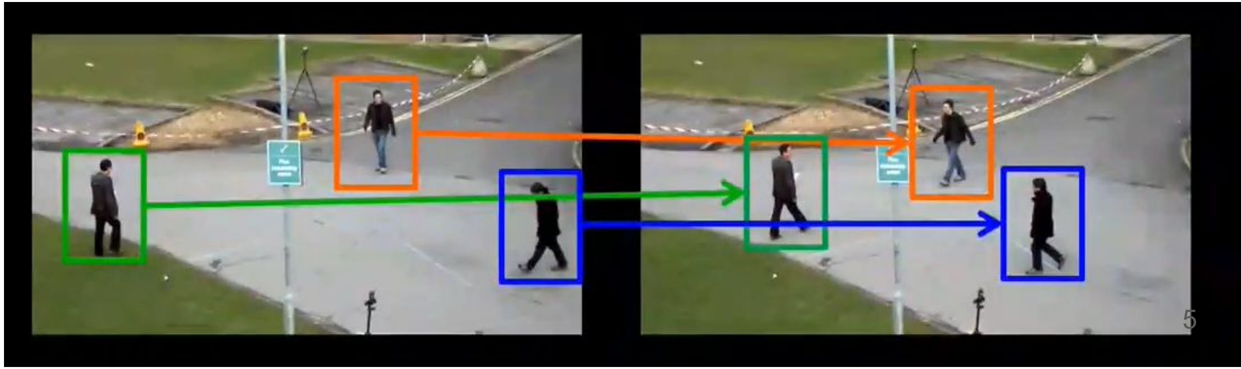
Detect and track all objects in a scene

(Again) Given a video, find out which parts of the image depict the same object in different frames

Detectors are often used as starting points -> Tracking by detection

Creating tracklets

Assigning ID to the objects



The importance of tracking

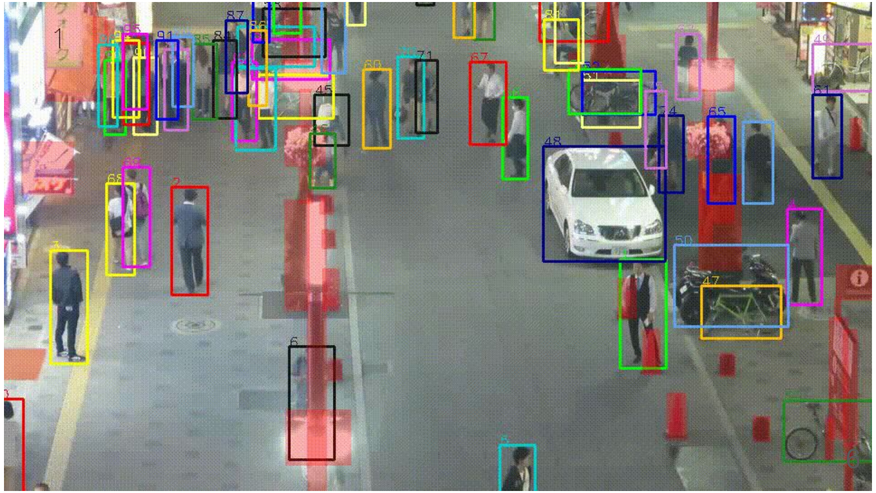
Pointing the objects when detection fails

- occlusions

- variations in viewpoint, pose, blur and illumination between frames of a sequence

- background clutter

Keeping the track of the object(s) of interest (detections can be returned with a random order per each frame, without any ID information)



Reasoning about the dynamic world, e.g. trajectory prediction

What tracking is about

Similarity measurement

Correlation

Correspondence

Matching/retrieval

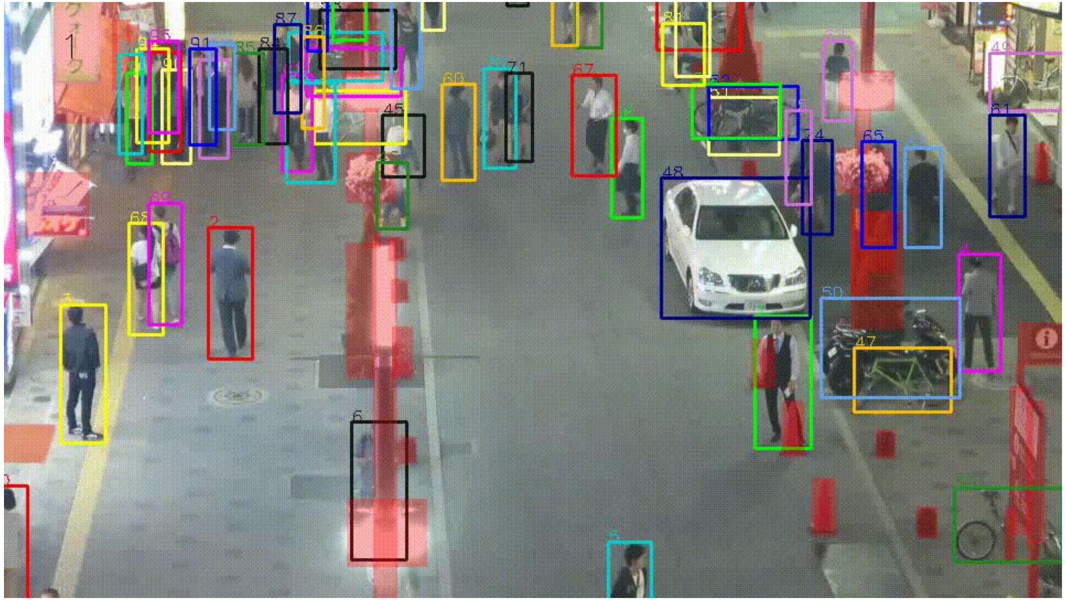
Data association

What tracking is about

Learning to model the target:

- appearance - how the target looks like
    * single-object tracking
    * re-identification
- motion - predicting where the target goes
    * trajectory prediction

Challenges



Multiple object of the same type

Heavy occlusions

Often very similar appearance

Emerging issues

- identity switches,

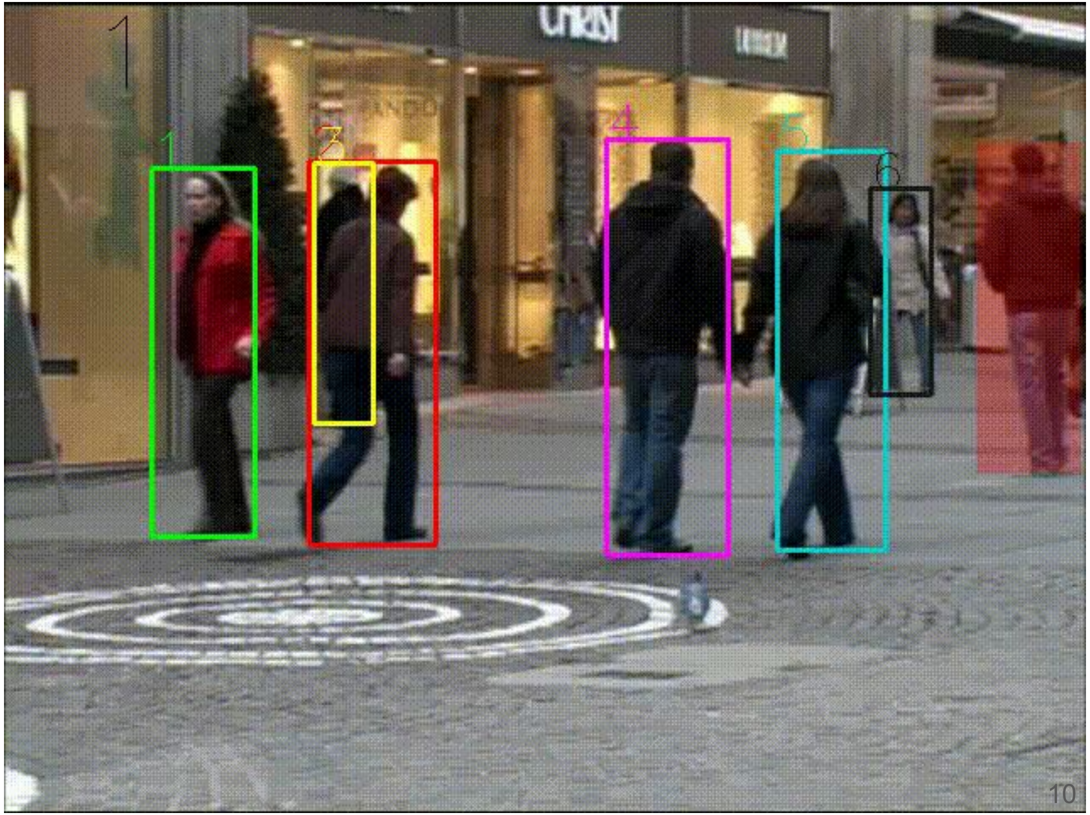
- short tracklets, fragmented tracklets

- targets leaving the scene (and then coming back)

MOT Challenge - MOT15

https://motchallenge.net/

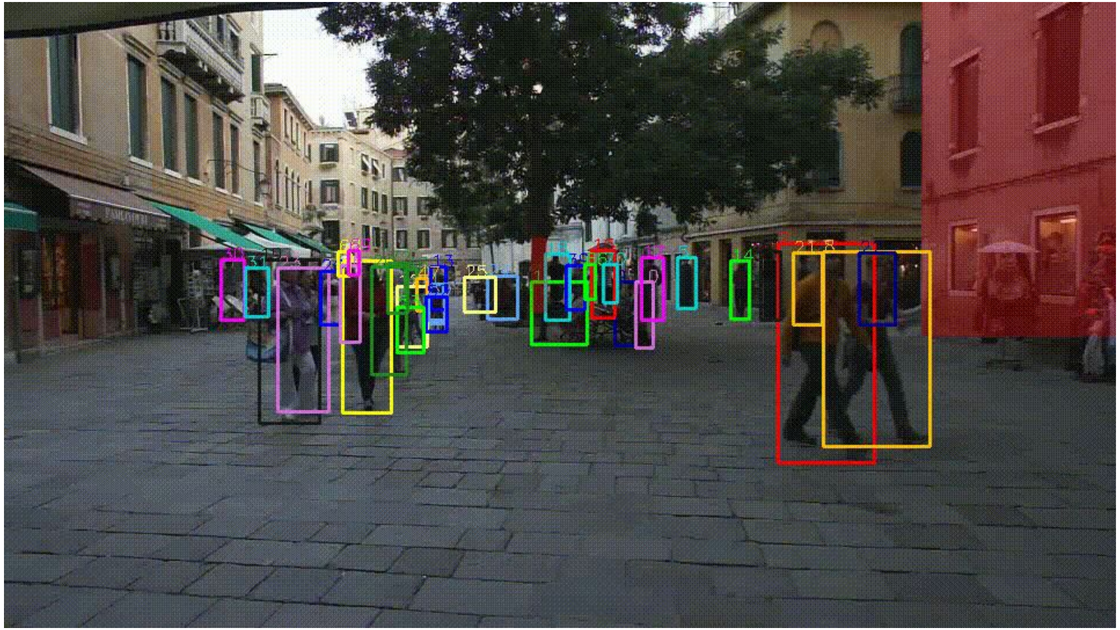
Multi-object tracking challenge



MOT Challenge - MOT17

https://motchallenge.net/

Multi-object tracking challenge



MOT Challenge - MOT20

https://motchallenge.net/

Multi-object tracking challenge

Online tracking and offline tracking

Online tracking: processing frames as they become available

- real-time application, e.g. autonomous driving, AR/VR

- prone to drifting - hard to recover from errors or occlusions

Offline tracking: processing a batch of frames

- good to recover from (short) occlusions

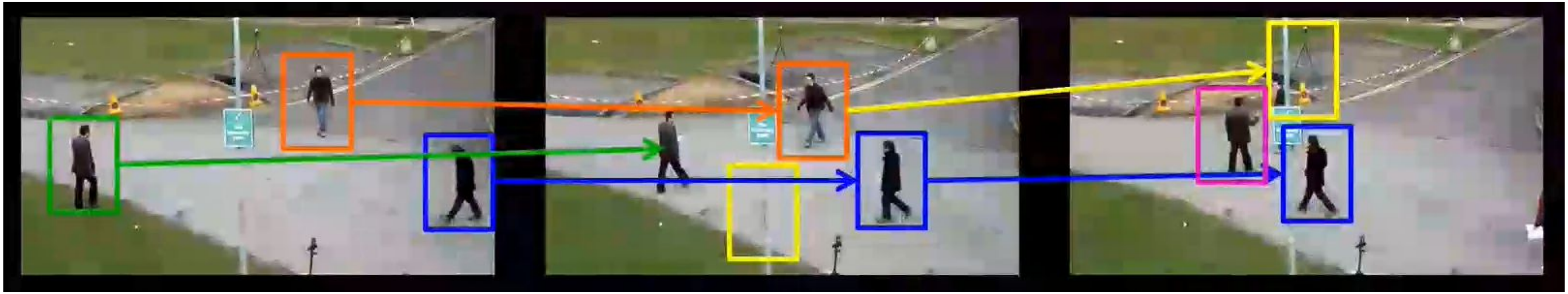
- non suitable for real-time applications

- yet suitable for video analysis, automatic labeling, video editing

Paradigm: Tracking by detection

Detection - detector is run on each frame to obtain a set of proposed locations

Data association - connecting the detections in the temporal domain to create trajectories



A simple online tracker

Track initialization, e.g. using a detector

Prediction of the next position - motion model

Matching predictions with detections - appearance model

A simple online tracker

Prediction of the next position - motion model

- Kalman filter

- Recurrent architectures

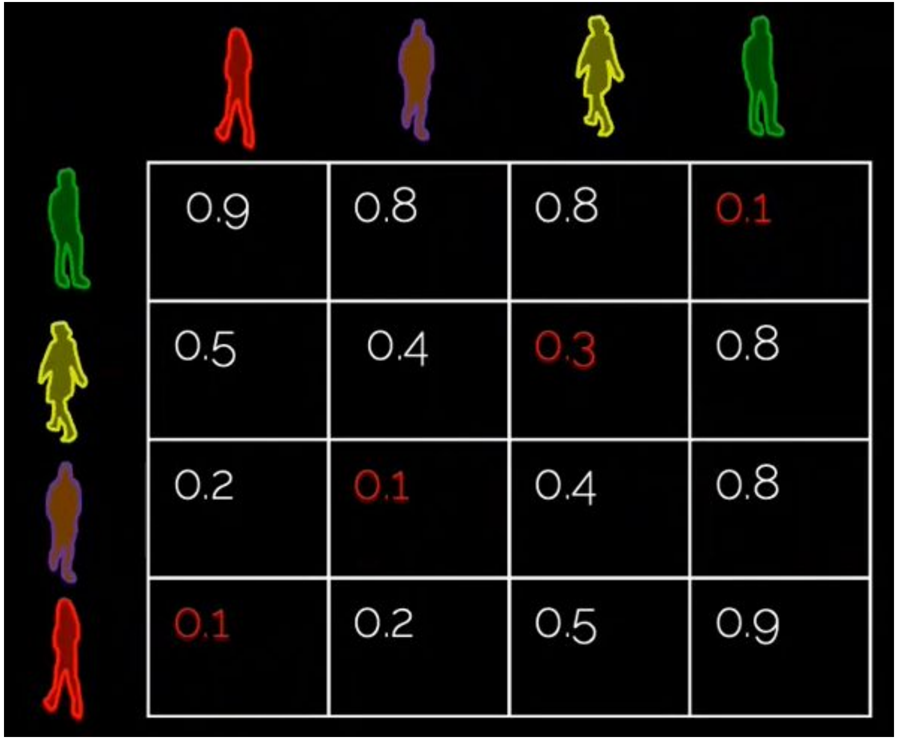
- simple constant velocity model

Bipartite matching

Define distance between boxes, e.g. IoU, pixel distance, reID

Solve the unique matching, e.g. with the Hungarian algorithm

Solutions are the unique assignments that minimize the total cost



The role of learning

Track initialization, e.g. using a detector

- Deep learning based detectors

Prediction of the next position - motion model

- Adding temporal complexity

Matching predictions with detections

- adding feature complexity
- improving appearance models - re-identification

Adding computational complexity

Tracking by detection - DeepSORT

"Simple Online and Realtime Tracking with a Deep Association Metric"

https://arxiv.org/pdf/1703.07402.pdf

Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \ldots, A_{\max}\}$ **do**
  Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
  $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(\boldsymbol{C}, \mathcal{T}_n, \mathcal{U})$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$

Simple approach and fast

Important milestone for MOT development

(Used to be) widely used, e.g. by companies

Not suitable for challenging scenarios

Many identity switches

Tracking by detection - ByteTrack

"ByteTrack: Multi-Object Tracking by Associating Every Detection Box"

https://arxiv.org/pdf/2110.06864.pdf

Algorithm 1: Pseudo-code of BYTE.

```
Input: A video sequence V; object detector Det; detection score
       threshold τ
Output: Tracks T of the video
Initialization: T ← ∅
for frame f_k in V do
    /* Figure 2(a) */
    /* predict detection boxes & scores */
    D_k ← Det(f_k)
    D_high ← ∅
    D_low ← ∅
    for d in D_k do
        if d.score > τ then
            D_high ← D_high ∪ {d}
        end
        else
            D_low ← D_low ∪ {d}
        end
    end

    /* predict new locations of tracks */
    for t in T do
        t ← KalmanFilter(t)
    end

    /* Figure 2(b) */
    /* first association */
    Associate T and D_high using Similarity#1
    D_remain ← remaining object boxes from D_high
    T_remain ← remaining tracks from T

    /* Figure 2(c) */
    /* second association */
    Associate T_remain and D_low using similarity#2
    T_re-remain ← remaining tracks from T_remain

    /* delete unmatched tracks */
    T ← T \ T_re-remain

    /* initialize new tracks */
    for d in D_remain do
        T ← T ∪ {d}
    end
end
Return: T
```

Tracking by detection - ByteTrack

Processing on the fly

Fine-tuned YOLOX object detector

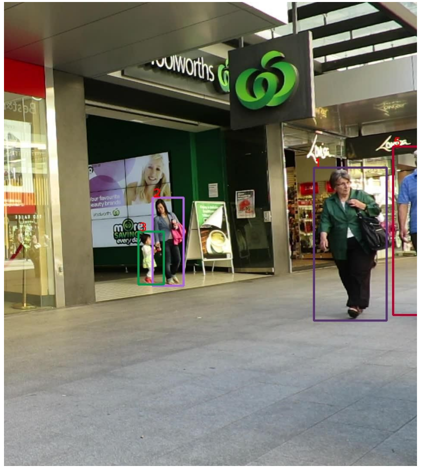
IoU, Kalman filter, well-engineered algorithm

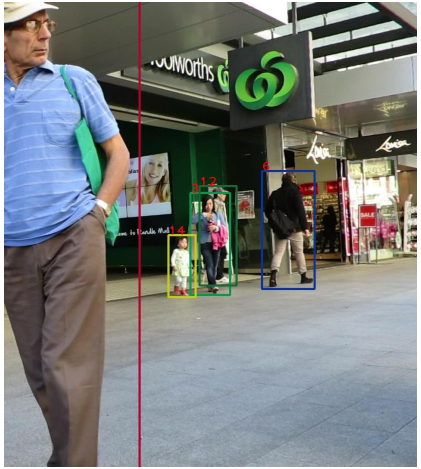
No learning scheme

Very good baseline

Many identity switches

Fragmented tracklets





Tracking by detection - SUSHI

"Unifying Short and Long-Term Tracking with Graph Hierarchies"

https://arxiv.org/pdf/2212.03038.pdf

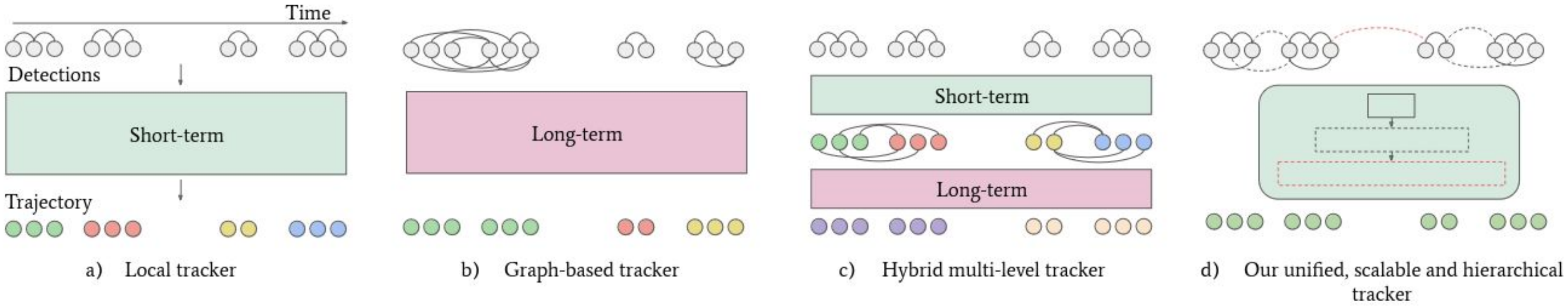


Figure 1. (a) Local tracker focusing on short-term scenarios and lacking robustness at long-term identity preservation (b) Graph-based tracker tackling longer-term association but unable to cover large time gaps due to its limited scalability (c) Hybrid multi-level tracker engineering a combination of techniques but still struggling with scalability (d) Our unified hierarchical tracker with high scalability.

Tracking by detection - SUSHI

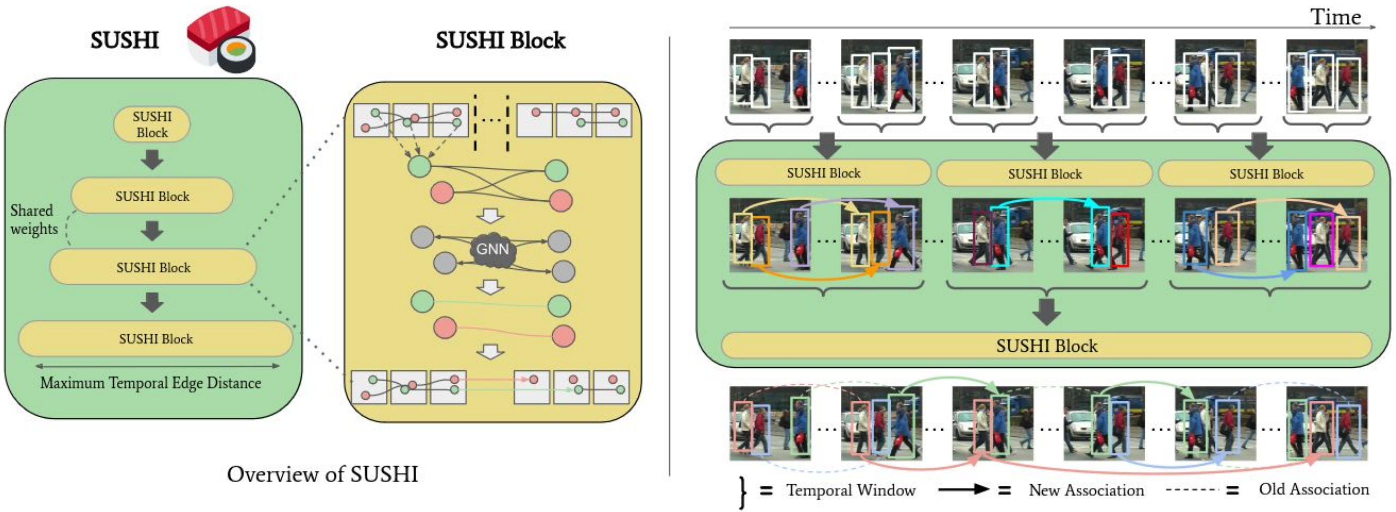


Figure 2. SUSHI consists of a set of SUSHI blocks operating hierarchically over a set of tracklets (with initial length one) in a video clip. Each SUSHI block considers a graph with tracklets from a subclip as nodes, performs neural message passing over it, and merges nodes into longer tracks. Over several hierarchy levels SUSHI blocks are able to progressively merge tracklets into tracks spanning over the entire clip. Notably, *SUSHI blocks share the same GNN architecture and weights*, hence making SUSHI unified across temporal scales.

Tracking by detection - SUSHI

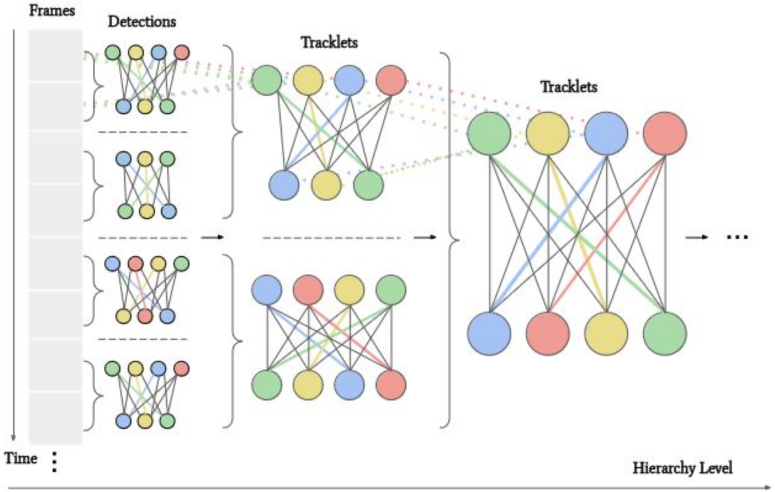


Figure 3. Our hierarchy is based on recursive partitioning of the video clip and we only allow edges within these partitions. After each level, we merge tracklets belonging to the same identity and consider edges spanning across longer timespans.

Tracking by detection - SUSHI

Global optimization approach (it needs to see all the frames in advance)

Graph-based association method

Also built on YOLOX (as ByteTrack)

Feature extraction through re-iD and mathematical derivations

Encoding the features further in the architecture

Behaving exceptionally well

Handling many challenging cases, e.g. occlusion,
crowded groups, longer trajectories

Some identity switches still present

Tendency to link newly appearing people to those who have left the scene

Still some Fragmented tracklets

Method	Det Ref.	IDF1 ↑	HOTA ↑	MOTA ↑	ID Sw. ↓
		MOT17 - Public			
Tracktor [3]	Tracktor	55.1	44.8	56.3	1987
LPT [26]	Tracktor	57.7	–	57.3	1424
MPNTrack [6]	Tracktor	61.7	49.0	58.8	1185
Lif_T [17]	Tracktor	65.6	51.3	60.5	1189
ApLift [18]	Tracktor	65.6	51.1	60.5	1709
GMT [16]	Tracktor	65.9	51.2	60.2	1675
LPC_MOT [10]	Tracktor	66.8	51.5	59.0	1122
SUSHI (Ours)	Tracktor	**71.5**	**54.6**	**62.0**	**1041**
		MOT17 - Private			
QDTrack [33]	✗	66.3	53.9	68.7	3378
TrackFormer [30]	✗	68.0	57.3	74.1	2829
MOTR [65]	✗	68.6	57.8	73.4	2439
PermaTrack [47]	✗	68.9	55.5	73.8	3699
MeMOT [8]	✗	69.0	56.9	72.5	2724
GTR [70]	✗	71.5	59.1	75.3	2859
FairMOT [68]	✗	72.3	59.3	73.7	3303
GRTU [49]	✗	75.0	62.0	74.9	1812
CorrTracker [48]	✗	73.6	60.7	76.5	3369
Unicorn [60]	✗	75.5	61.7	77.2	5379
ByteTrack† [67]	✗	77.1	62.8	78.9	2363
ByteTrack [67]	✗	77.3	63.1	80.3	2196
SUSHI (Ours)	✗	**83.1**	**66.5**	**81.1**	**1149**

Table 2. Test set results on MOT17 benchmark. Det. Ref. denotes the public detection refinement strategy. As ByteTrack (gray) uses different thresholds for test set sequences and interpolation, we also report scores by disabling these as ByteTrack† (black).

Tracking by detection concluded

Leverages well the advances in object detection

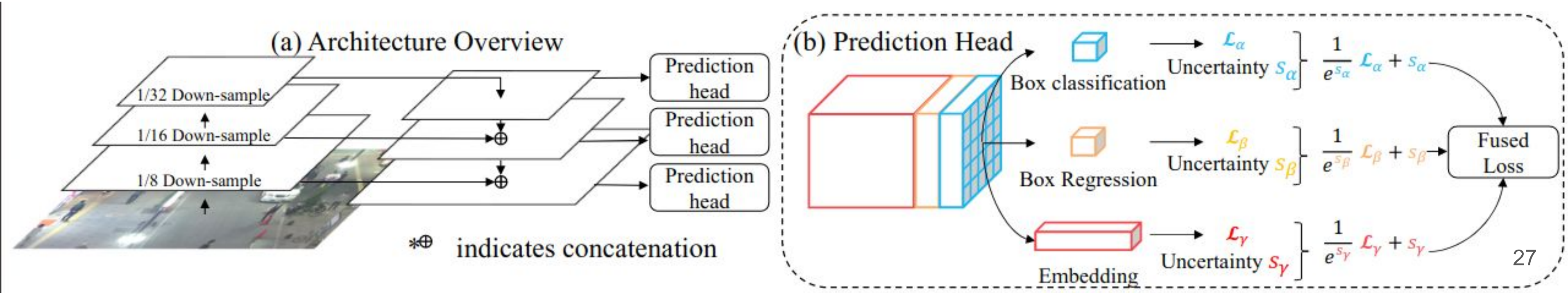
It can be used online (Hungarian) + by batches (adding computational complexity)

Paradigm: Joint detection and tracking

Joint detection and association embedding (JDE) - anchor based

"Towards Real-Time Multi-Object Tracking"

https://arxiv.org/pdf/1909.12605.pdf

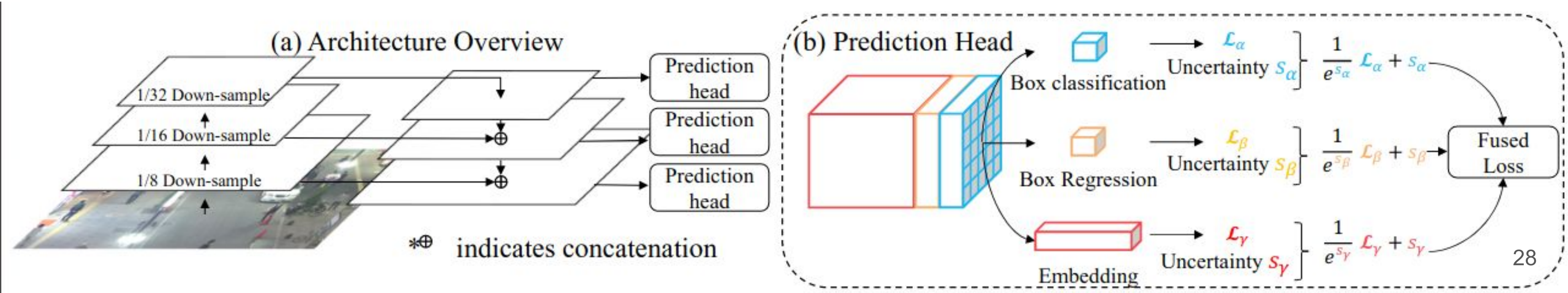


Joint detection and association embedding (JDE) - anchor based

Association via embedding distance

Near-real time (shared backbone)

Jointly training for detection and tracking, but tasks still separated in different heads



Anchor-free JDE - FairMOT

"FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking", https://arxiv.org/pdf/2004.01888.pdf

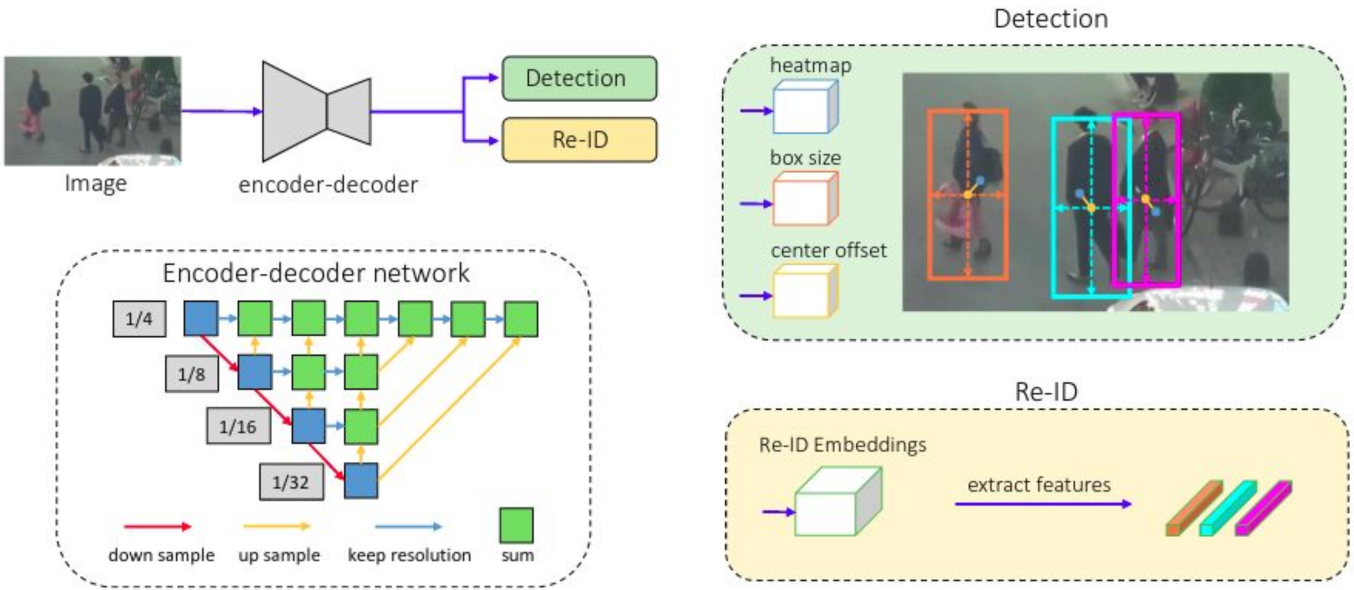
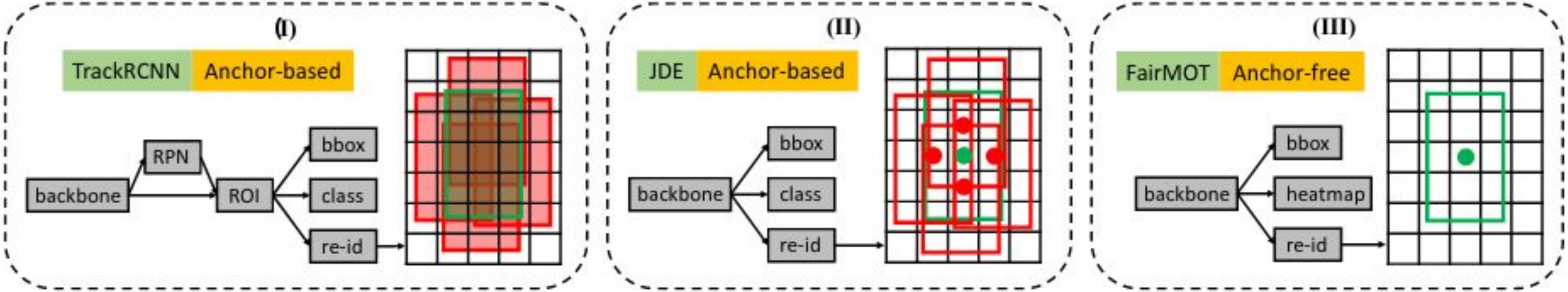


Fig. 1 Overview of our one-shot tracker *FairMOT*. The input image is first fed to an encoder-decoder network to extract high resolution feature maps (stride=4). Then we add two homogeneous branches for detecting objects and extracting re-ID features, respectively. The features at the predicted object centers are used for tracking.

Anchor-free JDE - FairMOT



(a) *Comparison of the existing one-shot trackers and FairMOT*

(b) *One anchor contains multiple identities*

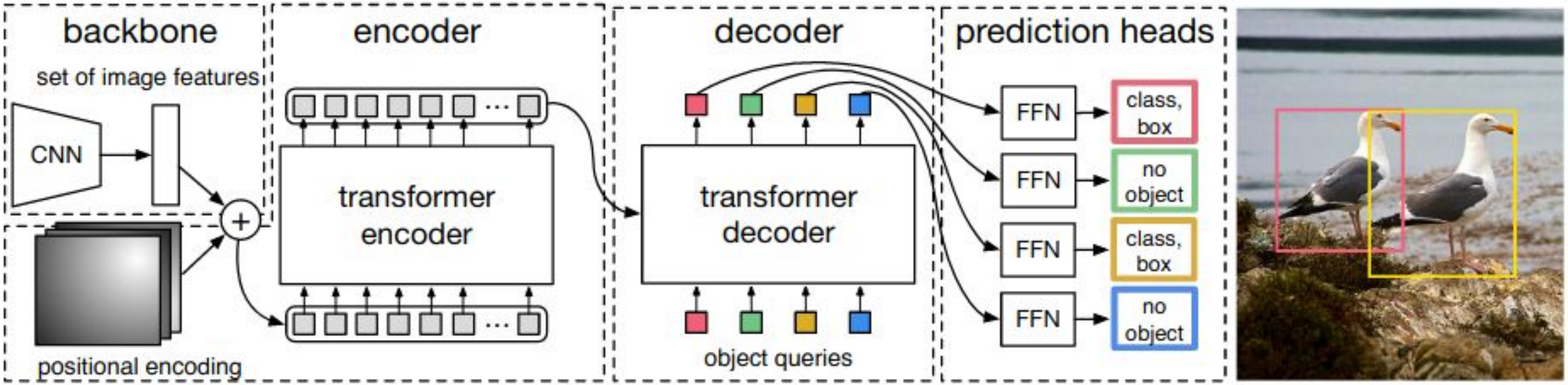
(c) *Multiple anchors response for one identity*

(d) *One point for one identity*

Detection with transformers - DETR

"End-to-End Object Detection with Transformers"

https://arxiv.org/pdf/2005.12872.pdf



Tracking with transformers - TrackFormer

"TrackFormer: Multi-Object Tracking with Transformers"

https://arxiv.org/pdf/2101.02702.pdf

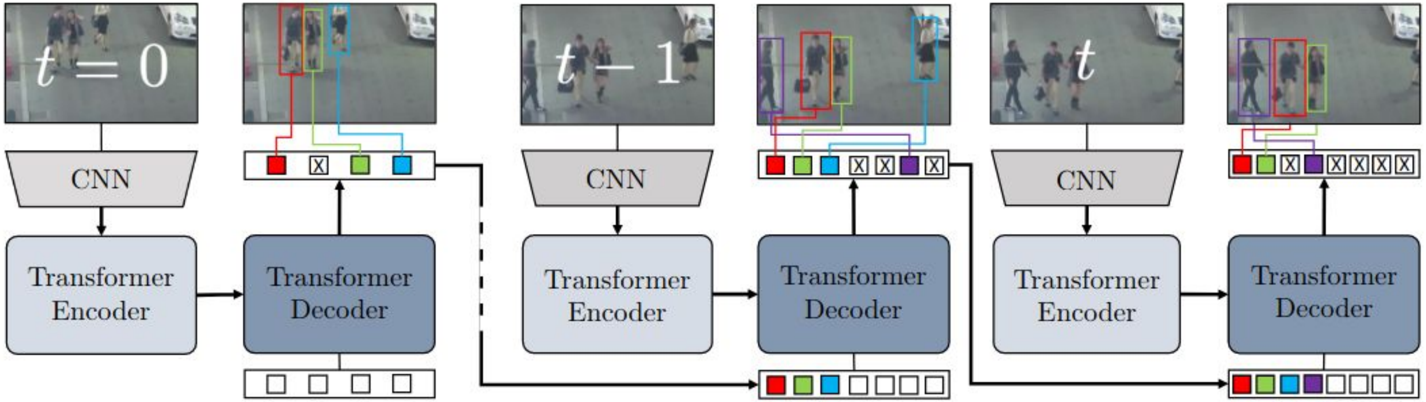


Figure 2. **TrackFormer** casts multi-object tracking as a set prediction problem performing joint detection and **tracking-by-attention**. The architecture consists of a CNN for image feature extraction, a Transformer [51] encoder for image feature encoding and a Transformer decoder which applies self- and encoder-decoder attention to produce output embeddings with bounding box and class information. At frame $t = 0$, the decoder transforms N_{object} object queries (white) to output embeddings either initializing new autoregressive **track queries** or predicting the background class (crossed). On subsequent frames, the decoder processes the joint set of $N_{\text{object}} + N_{\text{track}}$ queries to follow or remove (blue) existing tracks as well as initialize new tracks (purple).

Tracking with transformers - TrackFormer

Nice solution naturally merging detection and data association

Generally very good performance

Yet difficult to train, a lot of data required (MOT datasets are not sufficient)

Current trends

https://paperswithcode.com/sota/multi-object-tracking-on-mot17

https://paperswithcode.com/sota/multi-object-tracking-on-mot20-1

https://paperswithcode.com/sota/multi-object-tracking-on-dancetrack

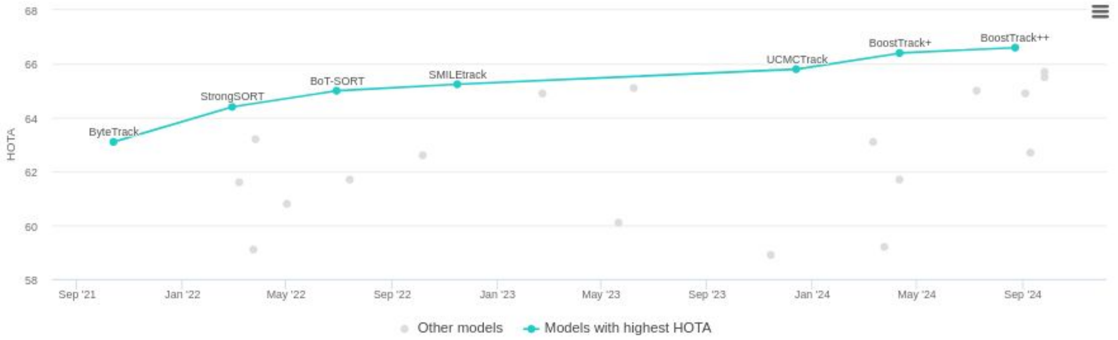
https://paperswithcode.com/sota/multiple-object-tracking-on-kitti-test-online

Let us see what are the approaches currently leading on the known benchmarks…

Multi-Object Tracking on MOT17

Leaderboard Dataset

View HOTA by Date for All models



Filter: appearaence untagged

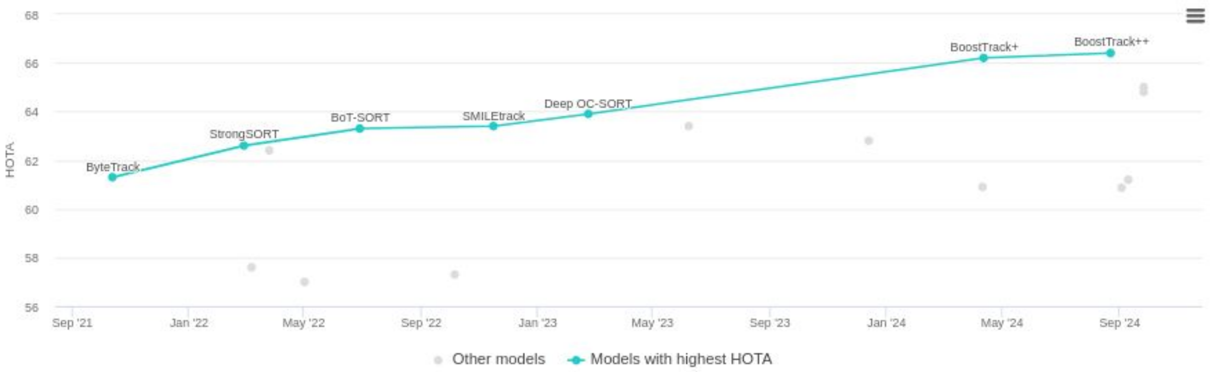
Edit Leaderboard

Rank	Model	HOTA↑	MOTA	IDF1	AssA	DetA	e2e-MOT	Speed (FPS)	Extra Training Data	Paper	Code	Result	Year	Tags
1	**BoostTrack++**	66.6	80.7	82.2					✓	BoostTrack++: using tracklet information to detect more objects in multiple object tracking			2024	
2	**BoostTrack+**	66.4	80.6	81.8					✓	BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking			2024	
3	**UCMCTrack**	65.8	80.5	81.1					×	UCMCTrack: Multi-Object Tracking with Uniform Camera Motion Compensation			2023	

Multi-Object Tracking on MOT20

Leaderboard Dataset

View HOTA by Date for All models



Filter: untagged

Edit Leaderboard

Rank	Model	HOTA	MOTA	IDF1	AssA	Speed (FPS)	Extra Training Data	Paper	Code	Result	Year	Tags
1	**BoostTrack++**	66.4	77.7	82			×	BoostTrack++: using tracklet information to detect more objects in multiple object tracking			2024	
2	**BoostTrack+**	66.2	77.2	81.5	68.6		✓	BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking			2024	
3	**AdapTrack**	65.0	75.0	80.7	67.8		✓	AdapTrack: Adaptive Thresholding-Based Matching For Multi-object Tracking			2024	

Thus tracking by detection!

Very good detections needed

Bipartite matching:

- define distance between boxes, e.g. IoU, pixel distance, reID -> **good features extracted and distances defined**

- solve the unique matching, e.g. with the Hungarian algorithm -> cannot change much here...

Exemplary approach

"Hard to Track Objects with Irregular Motions and Similar Appearances? Make It Easier by Buffering the Matching Space"

https://arxiv.org/pdf/2211.14317.pdf

"Hard to Track Objects with Irregular Motions and Similar Appearances? Make It Easier by Buffering the Matching Space"

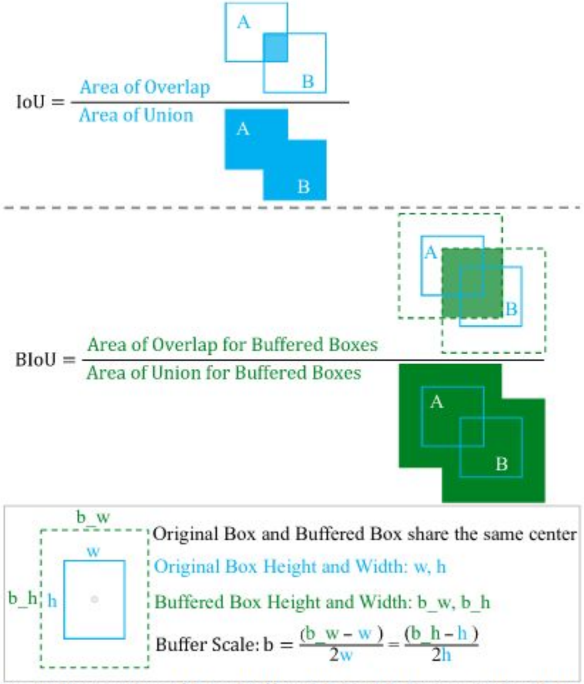


Fig. 2: **Illustration of how Buffered IoU (BIoU) is calculated**. Our BIoU adds a buffer that is proportional to the original bounding box. It does not change the location center, scale ratio, and shape of the original bounding boxes but expands the original matching space.

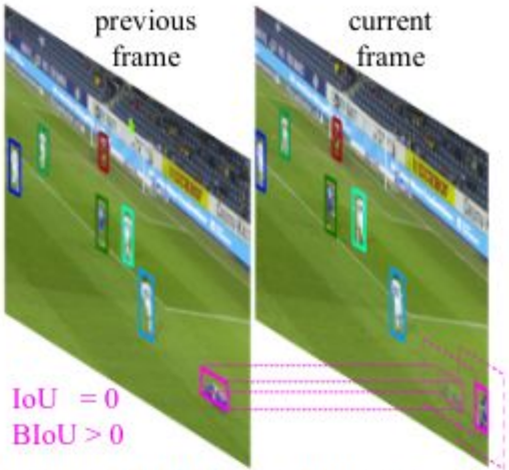


Fig. 3: **An illustration of BIoU forms better cross-frame geometric consistency than IoU**. The bounding box of an identical object shares the same color. The magenta object has no overlapping detections between adjacent frames. Whether this is caused by the fast movement or incorrect motion estimation, our BIoU expands the matching space to reduce the miss matching.

Multi-Object Tracking on DanceTrack

Leaderboard Dataset

View HOTA by Date for All models

Filter: untagged

Edit Leaderboard

Rank	Model	HOTA	MOTA	IDF1	AssA	DetA	Extra Training Data	Paper	Code	Result	Year	Tags
1	**MOTIP** (Deformable DETR, with DanceTrack val and CrowdHuman)	73.7	92.7	78.4	65.9	82.6	✓	Multiple Object Tracking as ID Prediction			2024	
2	**MOTRv2**	73.4	92.1	76.0	64.4	83.7	✓	MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors			2022	
3	**MOTIP** (Deformable DETR, with CrowdHuman)	71.4	91.6	76.3	62.8	81.3	✓	Multiple Object Tracking as ID Prediction			2024	

Multi-Object Tracking on DanceTrack

Leaderboard Dataset

View HOTA by Date for All models

Filter: untagged

Edit Leaderboard

Rank	Model	HOTA↑	MOTA	IDF1	AssA	DetA	Extra Training Data	Paper	Code	Result	Year	Tags
1	**MOTIP** (Deformable DETR, with DanceTrack val and CrowdHuman)	73.7	92.7	78.4	65.9	82.6	✓	Multiple Object Tracking as ID Prediction			2024	
2	**MOTRv2**	73.4	92.1	76.0	64.4	83.7	✓	MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors			2022	
3	**MOTIP** (Deformable DETR, with CrowdHuman)	71.4	91.6	76.3	62.8	81.3	✓	Multiple Object Tracking as ID Prediction			2024	

Thus transformers!

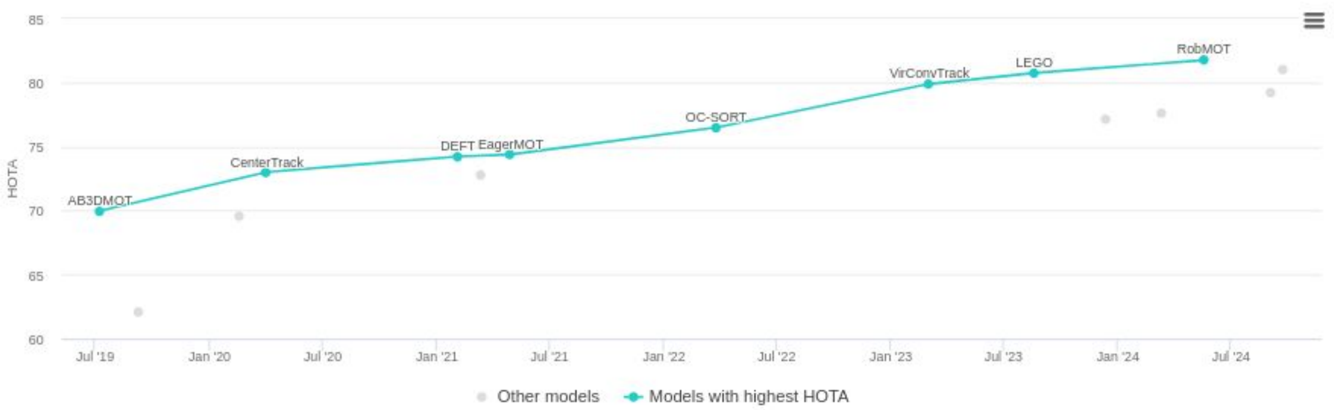
Performing well when the subjects remain mostly and the scene (DanceTrack)

Struggling when subjects often enter and leave the scene (e.g. MOT17)

Multiple Object Tracking on KITTI Test (Online Methods)

Leaderboard Dataset

View HOTA by Date



Filter: untagged

Edit Leaderboard

Rank	Model	HOTA↑	MOTA	IDSW	Paper	Code	Result	Year	Tags
1	**RobMOT**	81.76	91.02	7	RobMOT: Robust 3D Multi-Object Tracking by Observational Noise and State Estimation Drift Mitigation on LiDAR PointCloud			2024	
2	**MCTrack**	81.07	89.82	64	MCTrack: A Unified 3D Multi-Object Tracking Framework for Autonomous Driving			2024	
3	**KFDL**	81.06	90.29	22					

Thus current trends:

Depending on the ~~dataset~~ *environment* characteristics

Tracking by detection, with simple yet powerful ideas and improvements

Let’s try to use both, tracking by detection *and* transformers!

https://arxiv.org/abs/2409.14220

Temporally Propagated Masks and Bounding Boxes: Combining the Best of Both Worlds for Multi-Object Tracking

Tomasz Stanczyk
Inria centre at Université Côte d’Azur
2004 Rte des Lucioles, 06902 Valbonne, France
tomasz.stanczyk@inria.fr

Francois Bremond
Inria centre at Université Côte d’Azur
2004 Rte des Lucioles, 06902 Valbonne, France
francois.bremond@inria.fr

McByte

We propose to use

a temporally propagated mask

as an association cue for MOT

McByte

We propose to use

a temporally propagated **m**ask

as an association **c**ue for MOT

We call it McByte

McByte

Using a mask temporal propagator

In this case: Cutie https://arxiv.org/pdf/2310.12982

Figure 1. Temporally propagated mask can be helpful in cases of high occlusion. The person with the red mask is tracked only by its limited visible parts (pointed by yellow arrows for the clarity). Input image data from [28]. Best seen in color.

Segmentation mask potential

Temporally propagated segmentation mask can be powerful

…but on its own, the mask is not sufficient for MOT

In fact, it harms performance if not used properly!
(tables later in the slides)

Built on the top of ByteTrack as the baseline

(Reminder)

"ByteTrack: Multi-Object Tracking by Associating Every Detection Box"

https://arxiv.org/pdf/2110.06864.pdf

Now let's incorporate a temporally propagated mask!

(Cutie is a transformer-based solution)

Algorithm 1: Pseudo-code of BYTE.

Input: A video sequence V; object detector Det; detection score threshold τ

Output: Tracks $\mathcal{T}$ of the video

```
Initialization: T ← ∅
for frame f_k in V do
    /* Figure 2(a) */
    /* predict detection boxes & scores */
    D_k ← Det(f_k)
    D_high ← ∅
    D_low ← ∅
    for d in D_k do
        if d.score > τ then
            D_high ← D_high ∪ {d}
        end
        else
            D_low ← D_low ∪ {d}
        end
    end

    /* predict new locations of tracks */
    for t in T do
        t ← KalmanFilter(t)
    end

    /* Figure 2(b) */
    /* first association */
    Associate T and D_high using Similarity#1
    D_remain ← remaining object boxes from D_high
    T_remain ← remaining tracks from T

    /* Figure 2(c) */
    /* second association */
    Associate T_remain and D_low using similarity#2
    T_re-remain ← remaining tracks from T_remain

    /* delete unmatched tracks */
    T ← T \ T_re-remain

    /* initialize new tracks */
    for d in D_remain do
        T ← T ∪ {d}
    end
end
Return: T
```

Applying the mask to MOT

We assign to each tracklet (tracked object) an initial segmentation mask (using SAM https://arxiv.org/abs/2304.02643)

We temporally propagate each mask along the next frames of the video sequence (Cutie https://arxiv.org/pdf/2310.12982)

We update the tracklets based on the temporally propagated mask signal

We manage and update the masks accordingly with the tracklet management and system

Applying the mask to MOT

During each frame we compute two ratios:

- the bounding box coverage of the mask, referred to as mask match no. 1, mm_1:

$$mm_1^{i,j} = \frac{|pix(mask(tracklet_i)) \cap pix(bbox_j)|}{|pix(mask(tracklet_i))|} \tag{1}$$

- the mask fill ratio of the bounding box, referred to as mask match no. 2, mm_2:

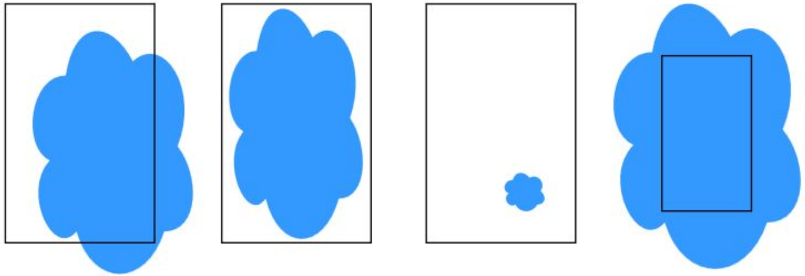
$$mm_2^{i,j} = \frac{|pix(mask(tracklet_i)) \cap pix(bbox_j)|}{|pix(bbox_j)|} \tag{2}$$

where $pix(\cdot)$ denotes pixels of the mask or within the bounding box, and $mask(\cdot)$ denotes the TP mask assigned to the tracklet. $|\cdot|$ denotes the cardinality of the set. Note that all $mm_1, mm_2 \in [0, 1]$.

Applying the mask to MOT

$$mm_1^{i,j} = \frac{|pix(mask(tracklet_i)) \cap pix(bbox_j)|}{|pix(mask(tracklet_i))|}$$

$$mm_2^{i,j} = \frac{|pix(mask(tracklet_i)) \cap pix(bbox_j)|}{|pix(bbox_j)|}$$



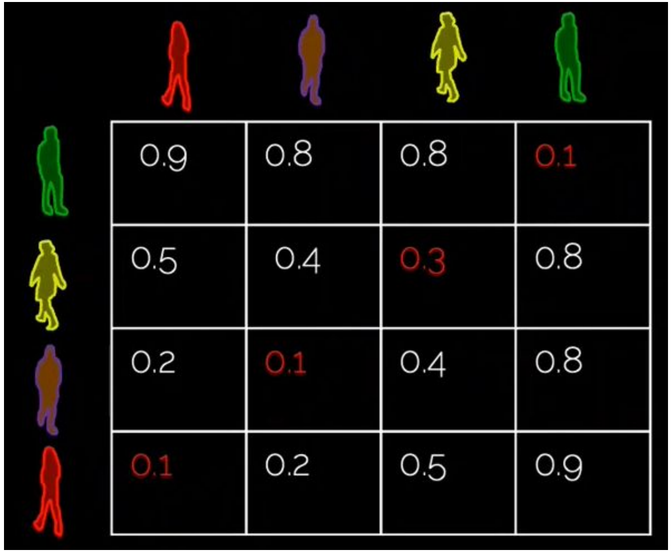
$mm_1 < 1.0$	$mm_1 = 1.0$	$mm_1 = 1.0$	$mm_1 < 1.0$
$mm_2 < 1.0$	$mm_2 \sim 1.0$	$mm_2 \ll 1.0$	$mm_2 = 1.0$

Figure 4. Cases showing the differences in mm_1 and mm_2 (Sec. 3.3) values of a temporally propagated mask (in blue) within a bounding box. The most optimal case for the mask to provide a good guidance is the second one from the left, where both mm_1 and mm_2 are as close to 1 as possible.

Applying the mask to MOT

$$costs^{i,j} = costs^{i,j} - mm_2^{i,j} \qquad (3)$$

where $costs^{i,j}$ denotes the cost between tracklet i and detection j.



0.9	0.8	0.8	0.1
0.5	0.4	0.3	0.8
0.2	0.1	0.4	0.8
0.1	0.2	0.5	0.9

The table image: courtesy of Prof. Dr. Laura Leal-Taixé: https://www.youtube.com/watch?v=mrMspzKcOAM

Conditional use

We need to see when the mask is actually reliable!

For each tracklet-detection pair, which could be ambiguous (a few or more detection boxes close to each other), we consider some conditions:

- Check if the mask is actually on the scene
- Check the confidence of the mask prediction
- Check if mm_2 is high enough (if it's not a noise)
- Check if mm_1 is high enough (if the mask indeed belongs to the considered tracklet)

Only in case of ambiguity *and* if all conditions are met, perform:

$$costs^{i,j} = costs^{i,j} - mm_2^{i,j} \quad (3)$$

where $costs^{i,j}$ denotes the cost between tracklet i and detection j.

Our tracking pipeline

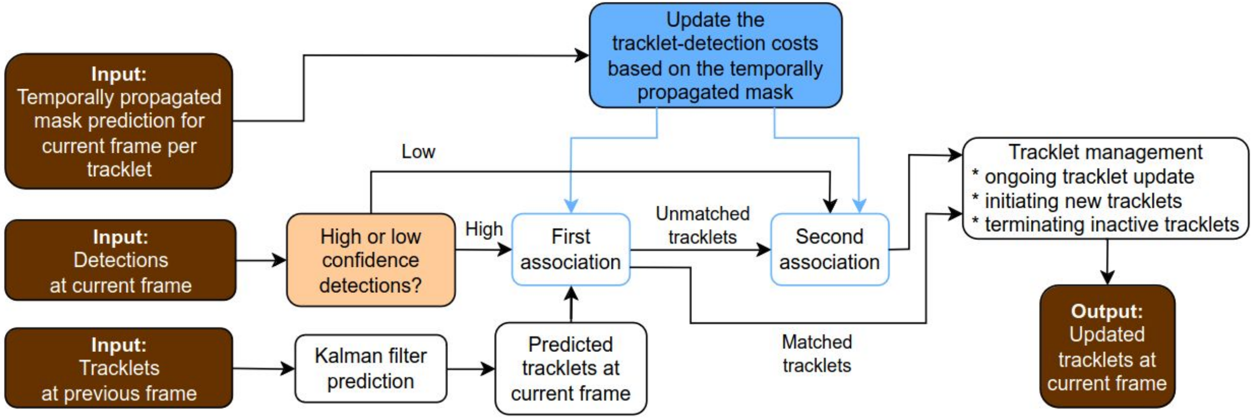


Figure 2. McByte tracking pipeline with the mask cue guidance. Temporally propagated mask signal is incorporated as an association cue in the tracklet-detection association steps.

State-of-the-art comparison

Method	HOTA	IDF1	MOTA
ByteTrack [40]	47.7	53.9	89.6
OC-SORT [4]	55.1	54.9	92.2
Deep OC-SORT [27]	61.3	61.5	92.3
C-BIoU [35] *	45.8	52.0	88.4
StrongSORT++ [12]	55.6	55.2	91.1
Hybrid-SORT [36]	65.7	67.4	91.8
McByte (ours)	**67.1**	**68.1**	**92.9**

Table 3. Comparing McByte with state-of-the-art tracking-by-detection algorithms on DanceTrack test set [33].

Method	HOTA	IDF1	MOTA
With parameter tuning per sequence			
ByteTrack [40]	63.1	77.3	80.3
StrongSORT++ [12]	64.4	79.5	79.6
OC-SORT [4]	63.2	77.5	78.0
Deep OC-SORT [27]	64.9	80.6	79.4
Hybrid-SORT [36]	64.0	78.7	79.9
Without parameter tuning per sequence			
ByteTrack [5]	62.8	77.1	78.9
C-BIoU [35] *	62.4	77.1	79.5
McByte (ours)	**64.2**	**79.4**	**80.2**

Table 4. Comparing McByte with state-of-the-art tracking-by-detection algorithms on MOT17 test set [28].

** C-BIoU: no code provided, thus we implement it ourselves*

Method	HOTA	IDF1	MOTA
Transformer-based			
MOTR [38]	57.8	68.6	73.4
MeMOTR [13]	58.8	71.5	72.8
MOTRv2 [41]	62.0	75.0	78.6
MOTIP [14]	59.2	71.2	75.5
Global optimization			
SUSHI [5]	66.5	83.1	81.1
Joint detection and tracking			
FairMOT [39]	59.3	72.3	73.7
RelationTrack [37]	61.0	75.8	75.6
CenterTrack [42]	52.2	64.7	67.8
Tracking-by-detection with parameter tuning per sequence			
ByteTrack [40]	63.1	77.3	80.3
StrongSORT++ [12]	64.4	79.5	79.6
OC-SORT [4]	63.2	77.5	78.0
Deep OC-SORT [27]	64.9	80.6	79.4
Hybrid-SORT [36]	64.0	78.7	79.9
Tracking-by-detection without parameter tuning per sequence			
ByteTrack [5]	62.8	77.1	78.9
C-BIoU [35] *	62.4	77.1	79.5
McByte (ours)	64.2	79.4	80.2

Table 10. Extended state-of-the-art method comparison on MOT17 [28] test set.

Method	HOTA	IDF1	MOTA
Transformer-based			
MOTR [38]	54.2	51.5	79.7
MeMOTR [13]	63.4	65.5	85.4
MOTRv2 [41]	73.4	76.0	92.1
MOTIP [14]	67.5	72.2	90.3
Global optimization			
SUSHI [5]	63.3	63.4	88.7
Joint detection and tracking			
FairMOT [39]	39.7	40.8	82.2
CenterTrack [42]	41.8	35.7	86.8
Tracking-by-detection			
ByteTrack [40]	47.7	53.9	89.6
OC-SORT [4]	55.1	54.9	92.2
Deep OC-SORT [27]	61.3	61.5	92.3
C-BIoU [35] *	45.8	52.0	88.4
StrongSORT++ [12]	55.6	55.2	91.1
Hybrid-SORT [36]	65.7	67.4	91.8
McByte (ours)	67.1	68.1	92.9

Table 11. Extended state-of-the-art method comparison on DanceTrack [33] test set.

State-of-the-art comparison

Method	HOTA	IDF1	MOTA
ByteTrack [40]	72.1	75.3	94.5
OC-SORT [4]	82.0	76.3	**98.3**
C-BIoU [35] *	72.7	76.4	95.4
McByte (ours)	**85.0**	**79.9**	96.8

Table 5. Comparing McByte with state-of-the-art tracking-by-detection algorithms on SoccerNet-tracking 2022 test set [9].

Method	HOTA	MOTA	HOTA	MOTA
	Pedestrian		Car	
ByteTrack [40]	54.3	63.7	47.3	34.9
PermaTr [34]	47.4	65.1	78.0	91.3
OC-SORT [4]	54.7	65.1	76.5	90.3
StrongSORT++ [12]	54.5	67.4	77.8	90.4
McByte (ours)	**57.0**	**68.9**	**80.8**	**92.5**

Table 6. Comparing McByte with state-of-the-art tracking-by-detection algorithms on KITTI-tracking test set [16]. KITTI evaluation server does not provide IDF1 scores.

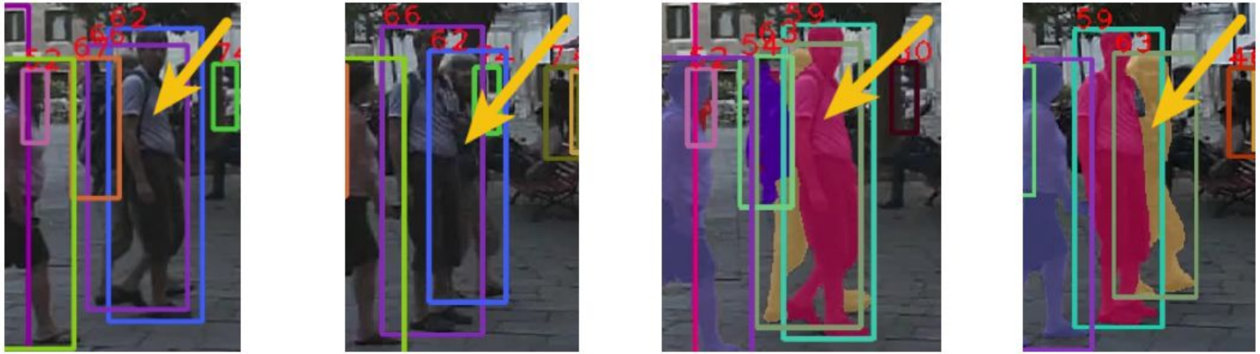
** C-BIoU: no code provided, thus we implement it ourselves*

Visual results

Frame 319 (baseline) Frame 401 (baseline) Frame 319 (McByte) Frame 401 (McByte)

Figure 3. Visual output comparison between the baseline and McByte. With the temporally propagated mask guidance, McByte can handle longer occlusion in the crowd - see the subject with ID 54 on the output of McByte. Input image data from [28]. Best seen in color.

Visual results



Frame 459 (baseline) Frame 475 (baseline) Frame 459 (McByte) Frame 475 (McByte)

Figure 8. Visual output comparison between the baseline and McByte. With the temporally propagated mask guidance, McByte can handle the association of an ambiguous set of bounding boxes - see the subjects with IDs 59 and 63 on the output of McByte. Input image data from [28]. Best seen in color.


ПЕРВЕНСТВО РОССИИ
ПО ХУДОЖЕСТВЕННОЙ ГИМНАСТИКЕ
Carlsberg
Thuner Amtsanzeiger
OCHSNER SPORT
frame: 0 fps: 0.00 num: 9

Questions and answers