# Object Tracking

Tomasz Stańczyk
tomasz.stanczyk@inria.fr

# 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
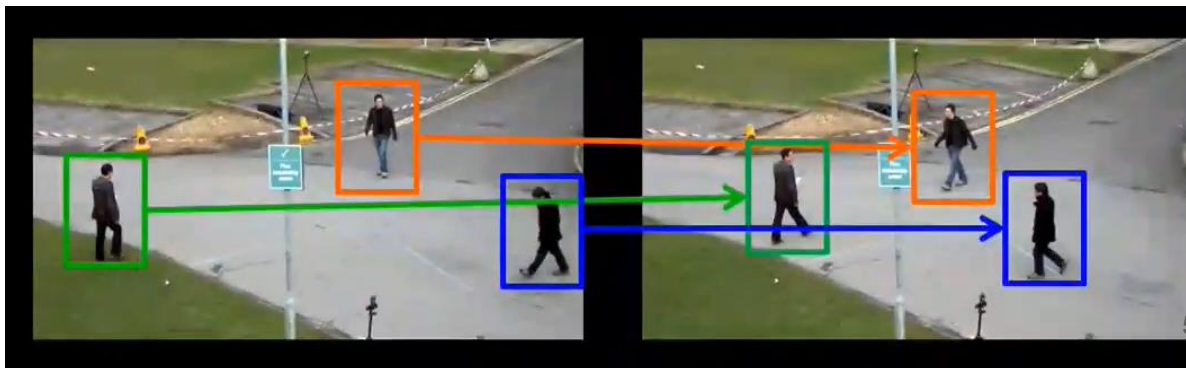
# Multi-object tracking

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
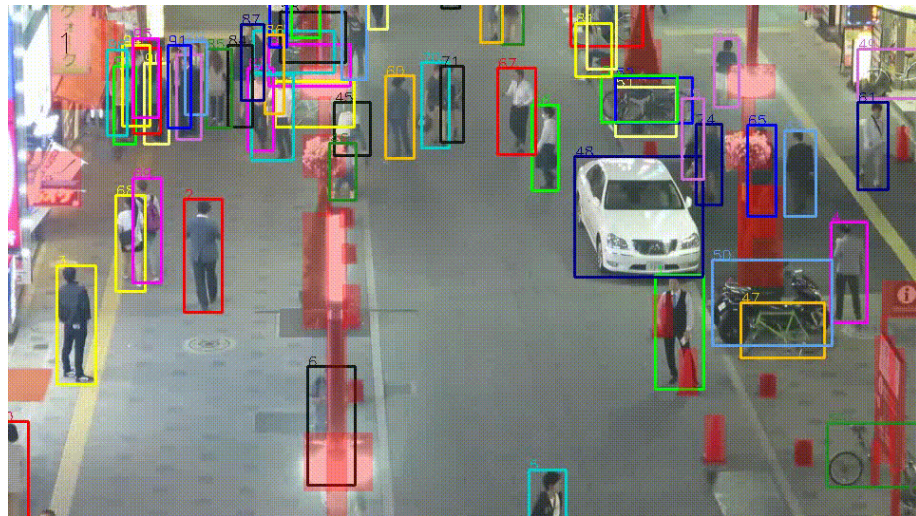
Assigning ID to the objects

# 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

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
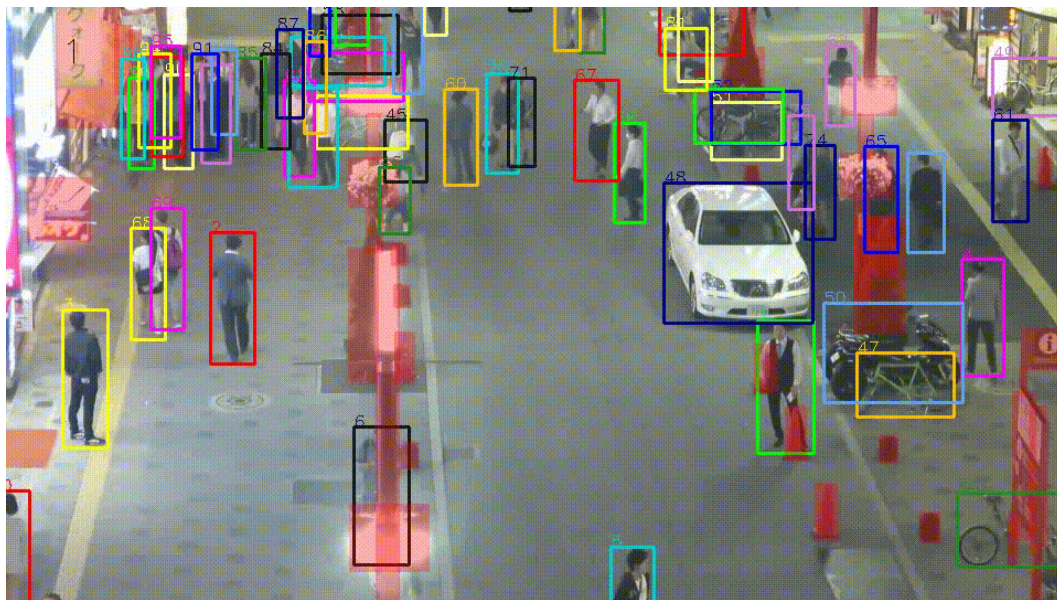
    * trajectory prediction

# Challenges

Multiple object of the same type

Heavy occlusions

Often very similar appearance

Emerging issues

- identity switches,
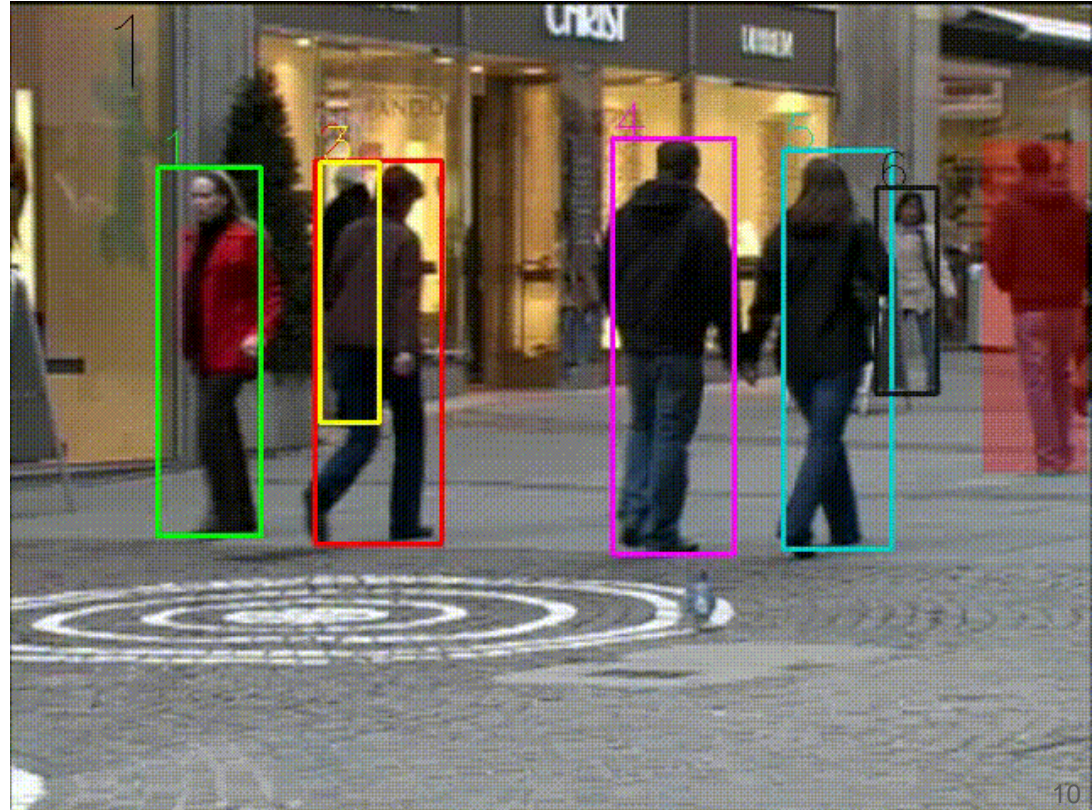
- short tracklets,

- targets leaving the scene (and then coming back)

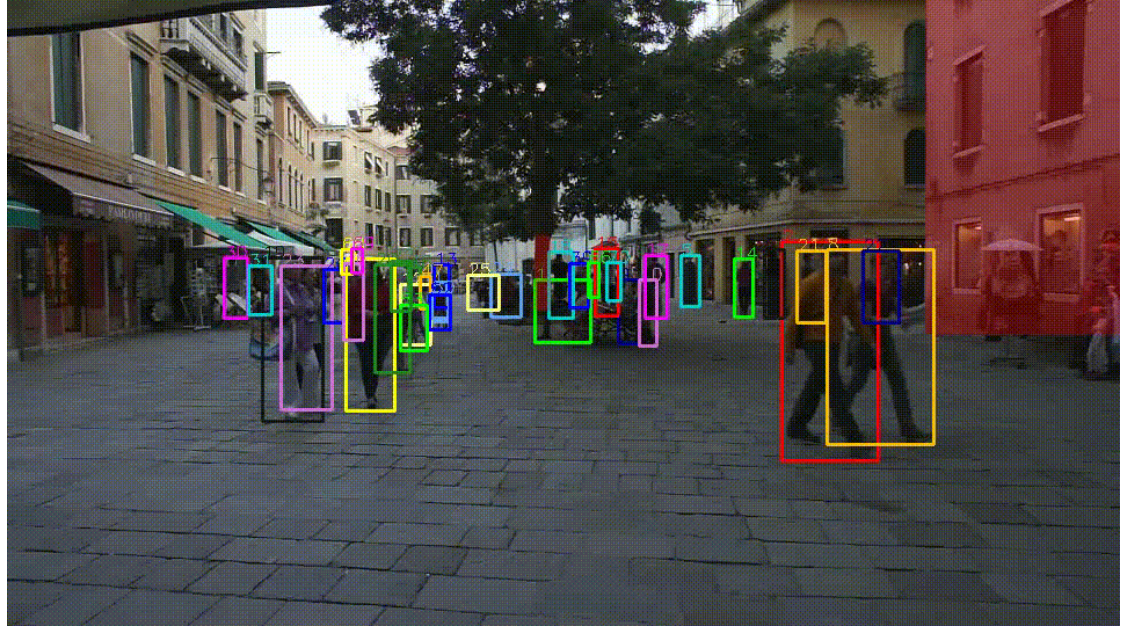# MOT Challenge - MOT15

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
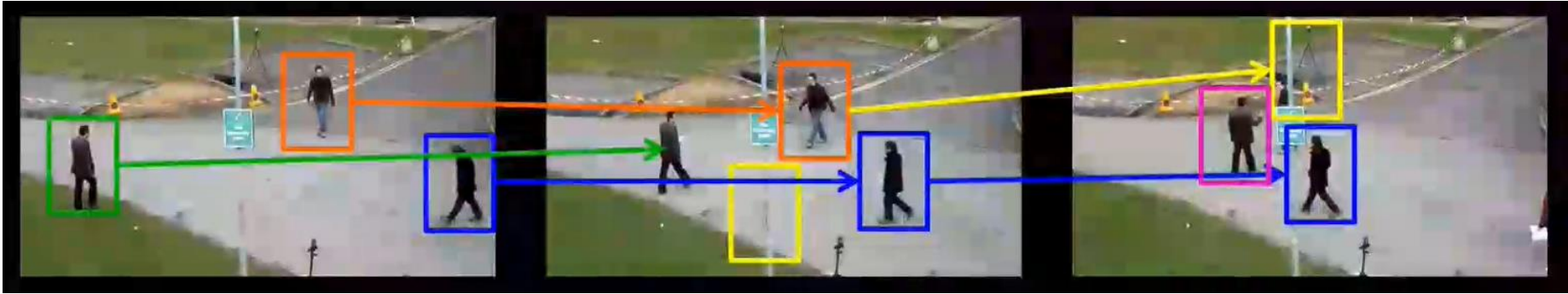
# A simple online tracker

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 $A_{\max}$

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

---

# 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 $\texttt{Det}$; detection score threshold $\tau$

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

1  Initialization: $\mathcal{T} \leftarrow \emptyset$
2  **for** *frame* $f_k$ *in* $V$ **do**
   /* Figure 2(a) */
   /* predict detection boxes & scores */
3     $\mathcal{D}_k \leftarrow \texttt{Det}(f_k)$
4     $\mathcal{D}_{high} \leftarrow \emptyset$
5     $\mathcal{D}_{low} \leftarrow \emptyset$
6     **for** $d$ *in* $\mathcal{D}_k$ **do**
7        **if** $d.score > \tau$ **then**
8           $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$
9        **end**
10       **else**
11          $\mathcal{D}_{low} \leftarrow \mathcal{D}_{low} \cup \{d\}$
12       **end**
13    **end**

   /* predict new locations of tracks */
14    **for** $t$ *in* $\mathcal{T}$ **do**
15       $t \leftarrow \texttt{KalmanFilter}(t)$
16    **end**

   /* Figure 2(b) */
   /* first association */
17    Associate $\mathcal{T}$ and $\mathcal{D}_{high}$ using $\texttt{Similarity\#1}$
18    $\mathcal{D}_{remain} \leftarrow$ remaining object boxes from $\mathcal{D}_{high}$
19    $\mathcal{T}_{remain} \leftarrow$ remaining tracks from $\mathcal{T}$

   /* Figure 2(c) */
   /* second association */
20    Associate $\mathcal{T}_{remain}$ and $\mathcal{D}_{low}$ using $\texttt{similarity\#2}$
21    $\mathcal{T}_{re-remain} \leftarrow$ remaining tracks from $\mathcal{T}_{remain}$

   /* delete unmatched tracks */
22    $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_{re-remain}$

   /* initialize new tracks */
23    **for** $d$ *in* $\mathcal{D}_{remain}$ **do**
24       $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
25    **end**
26 **end**
27 Return: $\mathcal{T}$

20

# 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



(a) Architecture Overview

1/32 Down-sample
1/16 Down-sample
1/8 Down-sample

Prediction head
Prediction head
Prediction head

$*\oplus$ indicates concatenation

(b) Prediction Head

Box classification $\quad \mathcal{L}_\alpha$ Uncertainty $s_\alpha$ $\quad \dfrac{1}{e^{s_\alpha}}\mathcal{L}_\alpha + s_\alpha$

Box Regression $\quad \mathcal{L}_\beta$ Uncertainty $s_\beta$ $\quad \dfrac{1}{e^{s_\beta}}\mathcal{L}_\beta + s_\beta$

Embedding $\quad \mathcal{L}_\gamma$ Uncertainty $s_\gamma$ $\quad \dfrac{1}{e^{s_\gamma}}\mathcal{L}_\gamma + s_\gamma$

Fused Loss
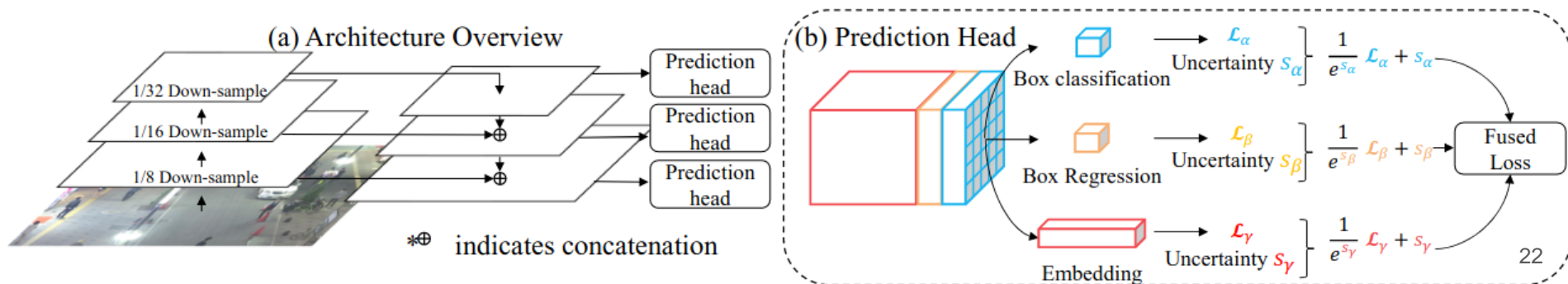
# Joint detection and association embedding (JDE) - anchor based

Association via embedding distance

Near-real time (shared backbone)

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



(a) Architecture Overview

1/32 Down-sample
1/16 Down-sample
1/8 Down-sample

Prediction head

*⊕ indicates concatenation

(b) Prediction Head

Box classification → $\mathcal{L}_\alpha$, Uncertainty $s_\alpha$ → $\frac{1}{e^{s_\alpha}} \mathcal{L}_\alpha + s_\alpha$

Box Regression → $\mathcal{L}_\beta$, Uncertainty $s_\beta$ → $\frac{1}{e^{s_\beta}} \mathcal{L}_\beta + s_\beta$

Embedding → $\mathcal{L}_\gamma$, Uncertainty $s_\gamma$ → $\frac{1}{e^{s_\gamma}} \mathcal{L}_\gamma + s_\gamma$

Fused Loss

23
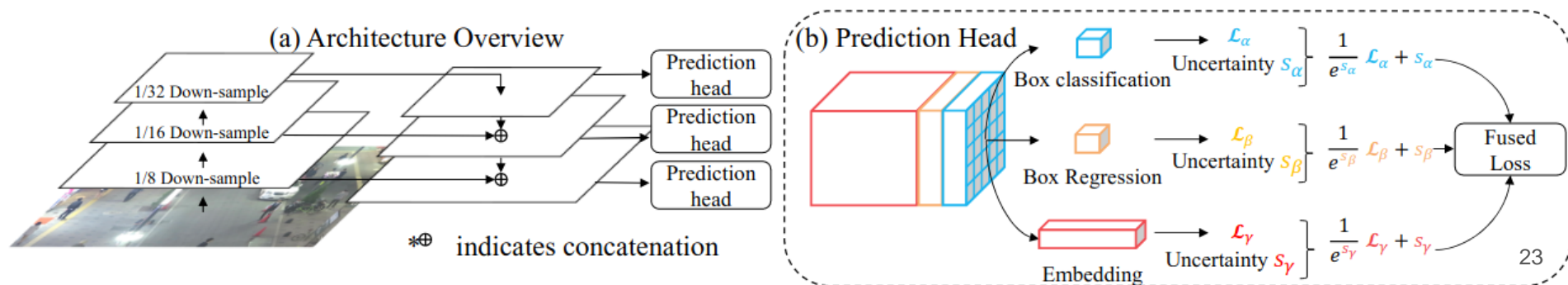
# Anchor-free JDE - FairMOT

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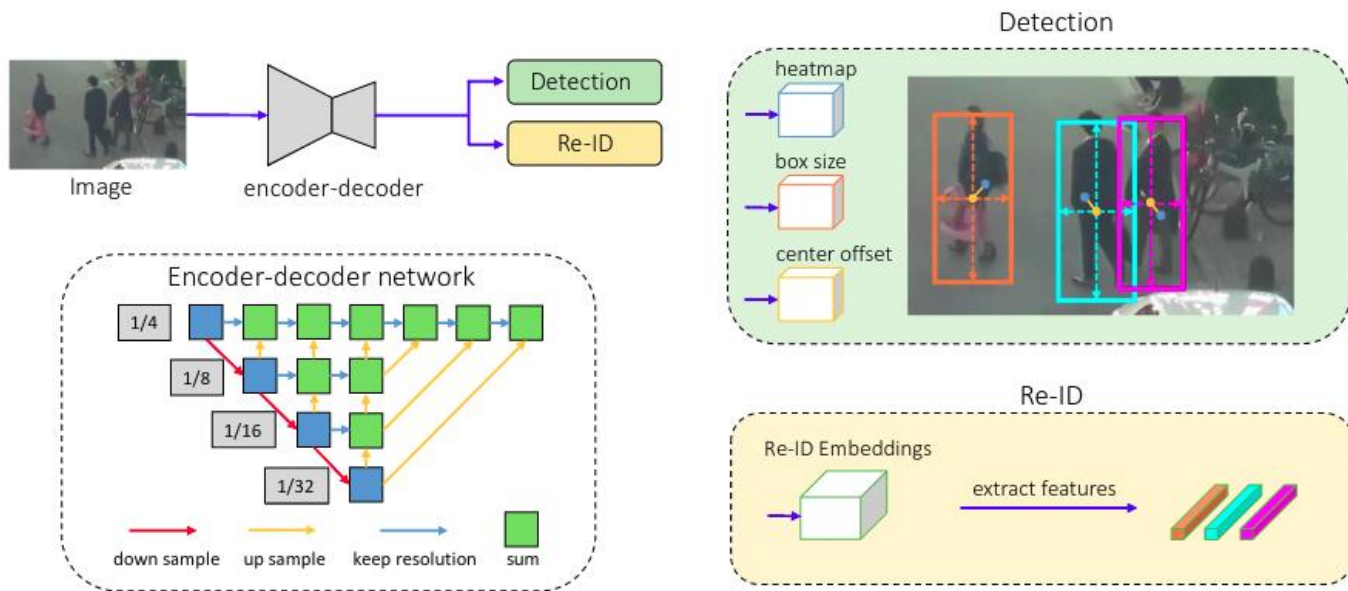


**Fig. 1** Overview of our one-shot tracker *FairMOT*. The input image is first fed to an encoder-decoder network to extract high resolution feature maps (stride=4). Then we add two homogeneous branches for detecting objects and extracting re-ID features, respectively. The features at the predicted object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

# Anchor-free JDE - FairMOT
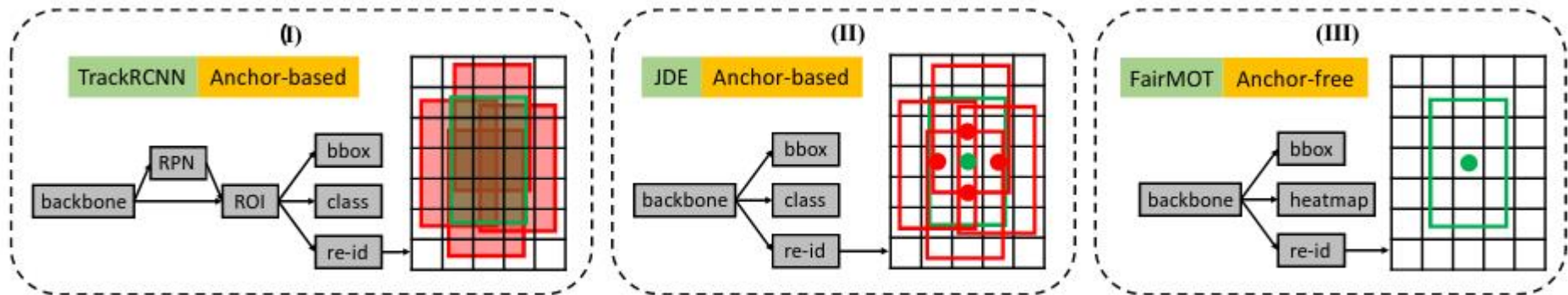


(a) Comparison of the existing one-shot trackers and FairMOT

(b) One anchor contains multiple identities

(c) Multiple anchors response for one identity

(d) One point for one identity

# Detection with transformers - DETR
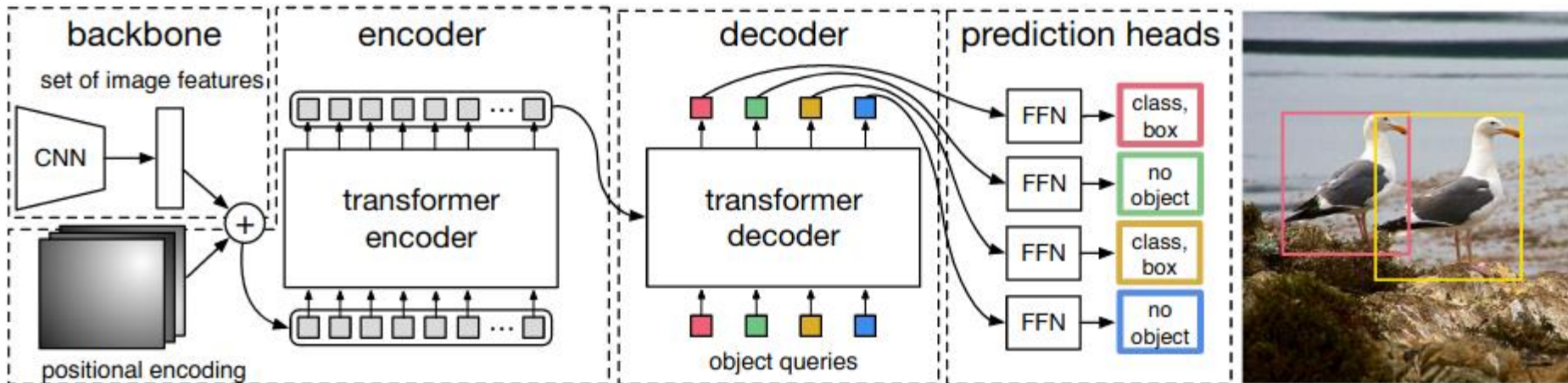
"End-to-End Object Detection with Transformers"

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

# Tracking with transformers - TrackFormer

"TrackFormer: Multi-Object Tracking with Transformers"
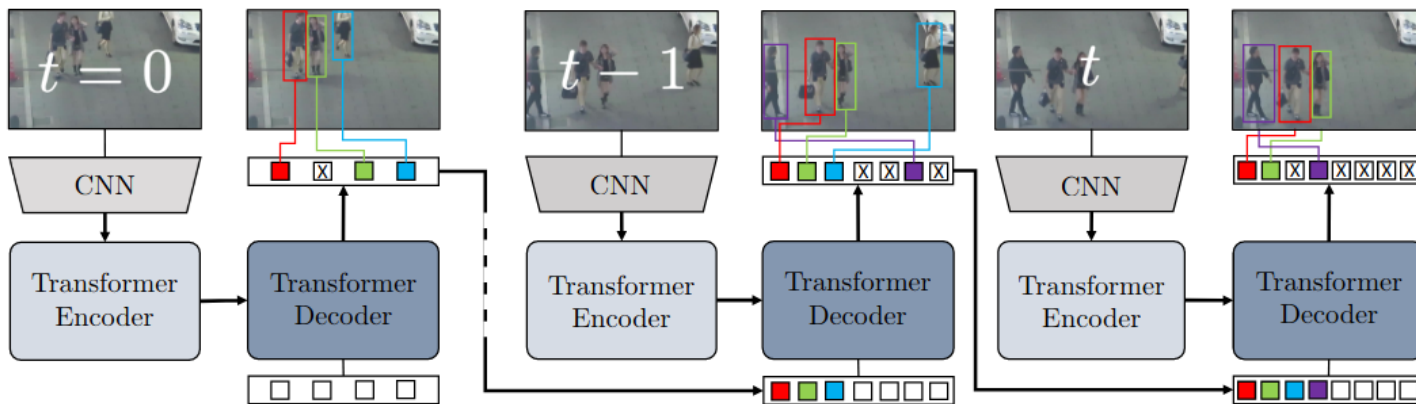
https://arxiv.org/pdf/2101.02702.pdf



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 $N_{object} + N_{track}$ 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)
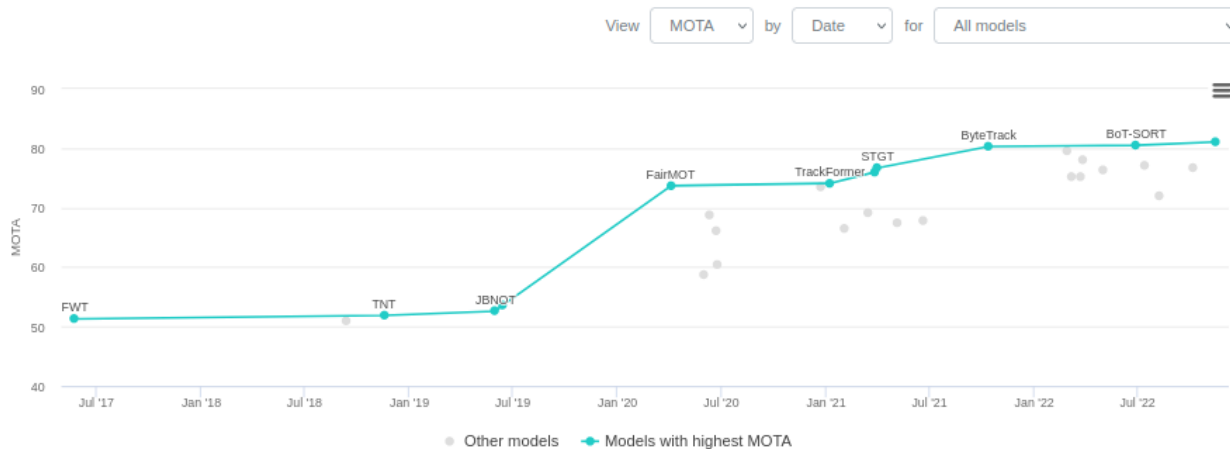
# Current trends

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

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

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

# Multi-Object Tracking on MOT17

Leaderboard    Dataset

View [ MOTA ▾ ] by [ Date ▾ ] for [ All models ▾ ]



● Other models  ●—● Models with highest MOTA

Filter: [ untagged ]

[ Edit Leaderboard ]

| Rank | Model | HOTA | MOTA⬆ | IDF1 | Extra Training Data | Paper | Code | Result | Year | Tags ✎ |
|------|-------|------|-------|------|---------------------|-------|------|--------|------|------|
| 1 | **SMILEtrack** | 65.24 | 81.06 | 80.5 | ✓ | SMILEtrack: SiMIlarity LEarning for Multiple Object Tracking | ⬡ | ⤓ | 2022 | |
| 2 | **BoT-SORT** | 65.0 | 80.5 | 80.2 | ✓ | BoT-SORT: Robust Associations Multi-Pedestrian Tracking | ⬡ | ⤓ | 2022 | |
| 3 | **ByteTrack** | | 80.3 | 77.3 | ✓ | ByteTrack: Multi-Object Tracking by Associating Every Detection Box | ⬡ | ⤓ | 2021 | |
| 4 | **StrongSORT** | 64.4 | 79.6 | 79.5 | ✓ | StrongSORT: Make DeepSORT Great Again | ⬡ | ⤓ | 2022 | |

30

# Multi-Object Tracking on MOT20

Leaderboard    Dataset

View [ MOTA ▾ ] by [ Date ▾ ] for [ All models ▾ ]



○ Other models  ●— Models with highest MOTA
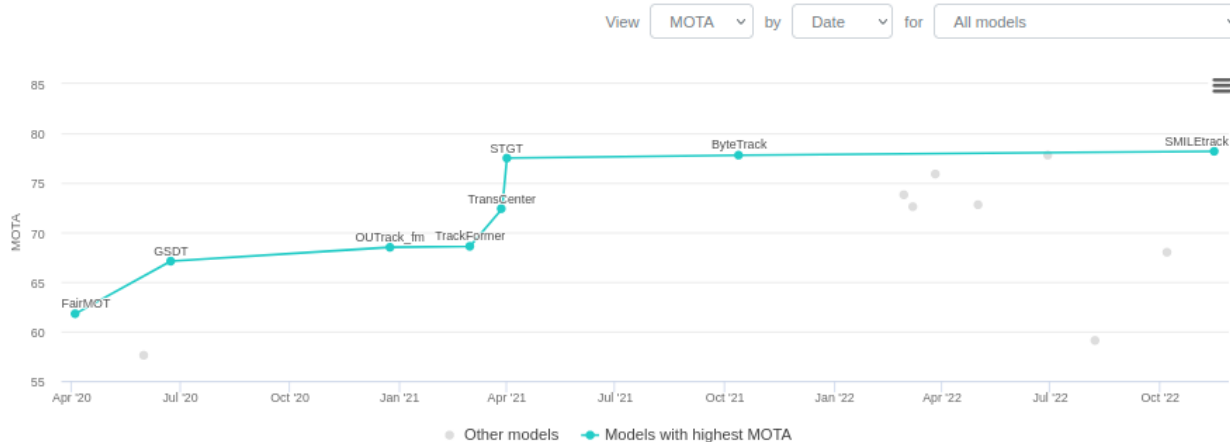
Filter: [ untagged ]                                      [ Edit Leaderboard ]

| Rank | Model | MOTA↑ | IDF1 | HOTA | Extra Training Data | Paper | Code | Result | Year | Tags ✎ |
|------|-------|-------|------|------|---------------------|-------|------|--------|------|--------|
| 1 | SMILEtrack | 78.19 | 77.53 | 63.43 | ✓ | SMILEtrack: SiMIlarity LEarning for Multiple Object Tracking | ○ | → | 2022 | |
| 2 | BoT-SORT | 77.8 | 77.5 | 63.3 | ✓ | BoT-SORT: Robust Associations Multi-Pedestrian Tracking | ○ | → | 2022 | |
| 3 | ByteTrack | 77.8 | 75.2 | 61.3 | ✓ | ByteTrack: Multi-Object Tracking by Associating Every Detection Box | ○ | → | 2021 | |
| 4 | STGT | 77.5 | 75.2 | | ✓ | TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking | | → | 2021 | |

# 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 with state-of-the-art results

"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"



$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

$$BIoU = \frac{\text{Area of Overlap for Buffered Boxes}}{\text{Area of Union for Buffered Boxes}}$$

Original Box and Buffered Box share the same center
Original Box Height and Width: w, h
Buffered Box Height and Width: b_w, b_h
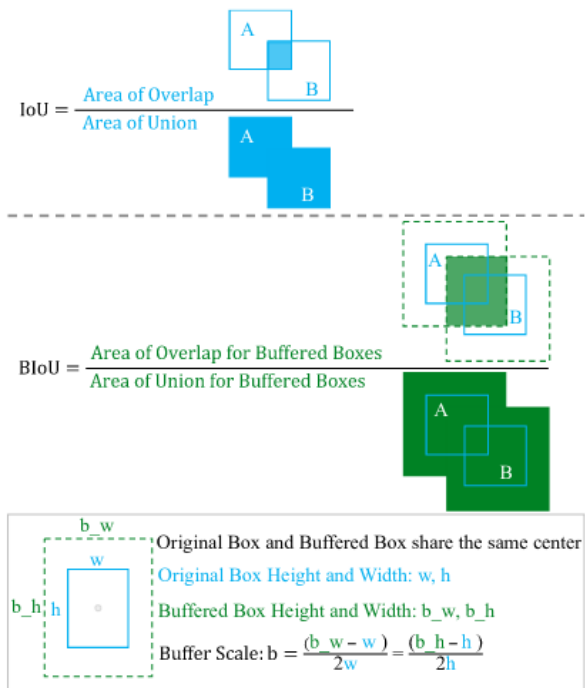Buffer Scale: $b = \frac{(b\_w - w)}{2w} = \frac{(b\_h - h)}{2h}$

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.
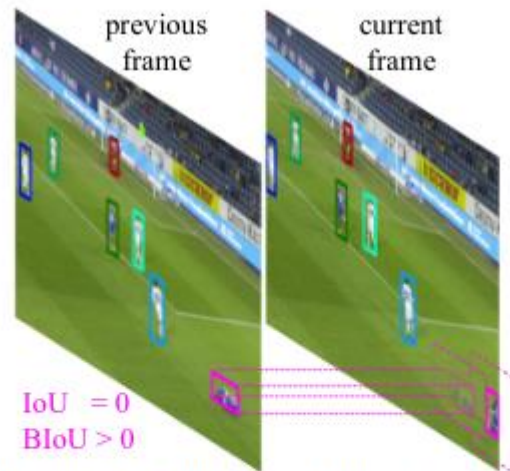


$IoU = 0$
$BIoU > 0$

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.

# Thus current trends:

Tracking by detection, with simple yet powerful ideas and improvements

# Questions and answers