DeTracker: A Joint Detection and Tracking Framework

Juan Diego Gonzales Zuniga, Ujjwal and François Bremond
INRIA Sophia Antipolis, 2004 Route des Lucioles, BP93 Sophia Antipolis Cedex, 06902, France

Keywords: Multiple Object Tracking, Joint Tracking and Detection, Graph Neural Networks.

Abstract: We propose a unified network for simultaneous detection and tracking. Instead of basing the tracking framework on object detections, we focus our work directly on tracklet detection whilst obtaining object detection. We take advantage of the spatio-temporal information and features from 3D CNN networks and output a series of bounding boxes and their corresponding identifiers with the use of Graph Convolution Neural Networks. We put forward our approach in contrast to traditional tracking-by-detection methods, the major advantages of our formulation are the creation of more reliable tracklets, the enforcement of the temporal consistency, and the absence of data association mechanism for a given set of frames. We introduce DeTracker, a truly joint detection and tracking network. We enforce an intra-batch temporal consistency of features by enforcing a triplet loss over our tracklets, guiding the features of tracklets with different identities separately clustered in the feature space. Our approach is demonstrated on two different datasets, including natural images and synthetic images, and we obtain 58.7% on MOT and 56.79% on a subset of the JTA-dataset.

1 INTRODUCTION

Object detection and "*multi-object tracking*" (MOT) are highly coupled tasks. In order to detect a set of targets, one must detect them and follow them through a set of video frames. Conversely, if a set of targets are tracked, it must be possible to localize them across the frames. This complementary nature of the two tasks suggests that it should be possible to train them jointly and obtain bounding boxes and tracklets as part of the same end-to-end pipeline. However in the predominant paradigm of "*tracking-by-detection*" (TBD), detection and tracking are trained separately and thus fail to complement each other's learning. In TBD, post-processed bounding boxes from a pre-trained detector serve as inputs to a tracker. Incorrect detection outputs therefore serve as noisy inputs and lead to incorrect tracklet generation. This is especially pressing in videos with crowded and occluding scenarios which are still significant challenges to existing detectors. Lastly, traditional TBD does not have the in-built mechanism of enforcing temporal consistency across detections, and the manner it has been attempted is by extracting features of detections in order to compute similarity measures between tracklets. This is still done with Siamese/re-id networks (Bergmann et al., 2019) which are trained with large datasets beforehand and handle appearance and geometric similarities but not temporal information, thereby undermining the temporal consistency of tracklets.

In order to improve MOT performance, we investigate both joint detection, tracking and data association components. There are some works (Kim and Kim, 2016; Bergmann et al., 2019; Feichtenhofer et al., 2017; Voigtlaender et al., 2019; Wang et al., 2019; Xu and Wang, 2019; Sun et al., 2020; Wang et al., 2020; Pang et al., 2020) which attempt to address joint-detection-and-tracking (JDT). (Kim and Kim, 2016; Bergmann et al., 2019; Feichtenhofer et al., 2017; Wang et al., 2020) propose to unify object detector with a model-free single-object tracker. The tracker first uses 2D CNNs to extract appearance features from the detection in the previous frame and from the image in the current frame. Based on these appearance features, the tracker regresses the location of the detected object to the current frame. As each object is tracked independently, the data association problem is naturally resolved. (Voigtlaender et al., 2019; Wang et al., 2019; Xu and Wang, 2019) propose to extend the object detector by adding a Re-ID branch, which employs CNNs to extract embeddings for positive and negative samples learned with the triplet loss. At test time, the embedding obtained from the Re-ID branch can serve as a similarity measurement used by the Hungarian algorithm for data association. However, these methods are limited to

Zuniga, J., Ujjwal, . and Bremond, F.
DeTracker: A Joint Detection and Tracking Framework.
DOI: 10.5220/0010875700003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 526-536
ISBN: 978-989-758-555-5; ISSN: 2184-4321

frame-wise detections. Consequently, they only allow frame-by-frame association, requiring manual adjustment of temporal stride. (Sun et al., 2020; Pang et al., 2020) propose a tubelet approach based on multiple input frames based on 3D CNNs as feature extractors but only consider a shallow temporal association proportioned by their respective backbones.

We advocate two important lines of improvements towards better MOT performance –

1. Object interaction within a single frame (spatial) and across multiple frames(spatio-temporal) can be beneficial for object localization and association. We point this argument because existing methods extract features by CNNs for each object at each frame and they are independent both from the same object in other frames and from other objects in any frame.
2. 3D CNN approaches have shown to be an improvement for multiple frame detection and benefit greatly from Graph Convolution Neural Networks as it was shown in with their 2DCNNs counterpart (Wang et al., 2020).

Our technique is different, as it directly computes tracklets over multiple frames by modeling their association, combining the strength of 3D CNNs and GCNNs. We postulate that an end-to-end trainable pipeline where detection and tracking complement each other during the training phase can bolster both detection and MOT. Our proposed method called *DeTracker* conducts joint detection of bounding boxes and tracklets while being end-to-end trainable. DeTracker focuses on 4 principle aspects – a) Detection of object bounding boxes, b) generating tracklets for each detected bounding box, c) ensuring *temporal consistency* i.e. ensuring high intra-class discriminability to discern between tracklets with different identities and simultaneously maintaining low discriminability for tracklets for the same identity computed in different batches of frames and, d) ensuring end-to-end training for the whole pipeline.

Our experiments and analysis confirm that *DeTracker* achieves comparable state-of-art performance on MOT. We achieve of 58.7% on the MOT17 dataset and 56.79% on a subset of the JTA dataset.

2 RELATED WORK

Tracking-by-Detection based Model. Research based on the TBD framework often adopts detection results given by external object detectors and focuses on the tracking part to associate the detection boxes across frames. Many association methods have been utilized on tracking models. In (Berclaz et al., 2011; Li Zhang et al., 2008; Lenz et al., 2014; Pirsiavash et al., 2011; Jiang et al., 2007), every detected bounding-box is treated as a node of graph, the associating task is equivalent to determining the edges where maximum flow (Berclaz et al., 2011; Xiu et al., 2018), or equivalently, minimum cost (Jiang et al., 2007; Pirsiavash et al., 2011; Li Zhang et al., 2008), are usually adopted as the principles. Recently, appearance-based matching algorithms have been proposed (Kim et al., 2015; Fang et al., 2018; Sadeghian et al., 2017). By matching targets with similar appearances such as clothes and body types, models can associate them over long temporal distances. Re-ID techniques (Kuo and Nevatia, 2011; Bergmann et al., 2019; Tang et al., 2017) are usually employed as an auxiliary in this matching framework. However, these methods are limited to frame-wise detections. Consequently, it only allows frame-by-frame association, requiring manual adjustment of temporal stride. Our technique is different, as it directly computes tracklets over multiple frames by linking prediction, enabling realtime solutions while considering all the frames.

3D CNNS for Multi-Object Tracking. Performance of image-based object detectors are limited when facing dense crowds and serious occlusions. Thus, some works utilize extra information as motion (Pang et al., 2020) or temporal features learned by the tracking step to aid detection. Both (Kang et al., 2017) and (Kang et al., 2016) generate multi-frame bounding box tubelet proposals and extract detection scores and features with a CNN and LSTM, respectively. (Pang et al., 2020; Mahadevan et al., 2020) make a case for 3D CNNs for Multi-object tracking and Object Segmentation, both tasks primarily done frame-by-frame. (Pang et al., 2020) is the most similar approach to our method, they predict bounding tubes (Btubes) across 3 different frames. Their prediction model consists of 15 degrees of freedom coordinates across 3 frames and 3 additional values for temporal locations. This allows the target to change its direction once for each prediction. Although (Pang et al., 2020) proposes the concept of Btubes, it does not enforce any temporal constraints and instead bases its tracklet association on different hyper-parameters such as IOU threshold depending on video frame rate and frame overlapping to compensate the aforementioned absence of temporal consistency.

Point-precise Tracking of Multiple Objects. Although some works extend MOT with pixel-precise masks (Milan et al., 2015; Osep et al., 2018), a much larger set of works can be found in the domain of Video Object Segmentation(VOS), which encompasses multiple tasks related to pixel-precise tracking. We tackle the task of MOT in videos by modeling the video clip as a 3D spatio-temporal volume and using a network to track object as points. This network is trained to push points belonging to different object instances towards different, non-overlapping clusters in the embedding space. This differs from most existing approaches, which first generate object detections per-frame, and the associate them over time.

Graph Convolution Neural Networks. GCNNs were first introduced by (Gori et al., 2005) to process data with a graph structure directly with neural networks. The key idea of GNNs is to define a computational graph with nodes and edges relating each other, and to update the node and edge features based on the node-node, node-edge, and edge-edge interactions, i.e., a process that is called feature aggregation. Each with a unique feature aggregation rule, different versions of GCNNs (.e.g, GraphConv, GCN, GAT) were proposed and have shown to be effective. Specifically, in computer vision, we have seen significant improvement using GCNNs in many subfields such as point cloud classification, single object tracking, and semantic segmentation. Despite that advances have been achieved with GCNNs in many fields, then is no published work leveraging GCNNs to model object interactions in object detection and data association for MOT. To the best of our knowledge, our work is the first introducing GCNNs and 3DCNNs to tackle joint detection and tracking. The following sections explain our problem formulation, the network architecture and the loss functions employed, and the inference process.

3 PROBLEM FORMULATION

Let us consider a minibatch $\mathcal{B}$ consisting of N frames of a video $\mathcal{V}$. Let there be $\mathcal{M}$ unique identities in $\mathcal{B}$. We want to detect a set of $K \triangleq \{k_i\}_{i=0}^{\mathcal{M}-1}$ tracklets where $k_i \triangleq \{b_i^j\}_{j=0}^{N-1}$ is described by a set of N bounding boxes. b_i^j is the bounding box of i^{th} identity in the j^{th} frame. A bounding box is given by a vector of tuples of the form $[(x_{tl}, y_{tl}), (x_{br}, y_{br})]$ where tl and br respectively abbreviate the *top-left* and *bottom-right* corners of a bounding box. If an identity i does not exist in frame j then we indicate that $b_i^j = [(0,0),(0,0)]$ without any loss of generality.

Since a video with large number of frames cannot be processed all at once, we need to link tracklets obtained from different minibatches of $\mathcal{V}$. This calls for features corresponding to different tracklets in a minibatch to be far apart. Therefore, we wish to learn a feature transformation $f(.)$ of a tracklet such that,

$$||f(k_i) - f(k_j)||_2 \triangleq \begin{cases} \leq \varepsilon & i = j \\ \geq \frac{1}{\varepsilon} & i \neq j \end{cases} \qquad (1)$$

where ε is an infinitesimally small number. The above formulation of temporal consistency within a minibatch simply denotes that features of same identity should be close while being far apart for different identities. As we will see, the above formulation is well built into DeTracker while supporting end-to-end training.

4 DETRACKER

DeTracker consists of 3 main components – a) *3D backbone*, b) *detection head* and c) *tracking head*. The overview of our approach is shown in figure 1. The 3D backbone processes a tensor of shape $\mathcal{B} \times 3 \times N \times H \times W$ of sequential video frames. The resulting feature maps from multiple layers of the backbone are reshaped into a 4D tensor of batch size $\mathcal{B}N$ which are fed to a FCOS (Tian et al., 2019) detector in the FPN (Lin et al., 2017) scheme. The detected bounding boxes are feature pooled using ROI-align (He et al., 2017) and operated upon by a *graph attention network* (GAT) to produce the final tracklets for the minibatch $\mathcal{B}$.

4.1 3D Backbone

We use a 3D backbone for initial feature extraction from a batch $\mathcal{B}$ of N video frames. As against 2D backbones used in many previous works (Zhu et al., 2018), a 3D backbone harnesses spatial as well as temporal characteristics of a video and selectively attends to moving entities (Carreira and Zisserman, 2017). We utilize R(2+1)D (Tran et al., 2018) pretrained on Kinetics-400 (Carreira and Zisserman, 2017) for initial feature extraction. In 3D backbones, temporal pooling leads to reduction in the number of output frames compared to the number of input frames. This is detrimental to our objective which is to detect and track people in each video frame of $\mathcal{B}$. Therefore, we remove all temporal pooling layers to have same number of input and output frames. Spatial pooling in 2D and 3D backbones squeezes the spatial dimensions of a feature map vis-à-vis the input. This leads to lack of feature details from small-scale objects. For a backbone with an output spatial stride of s, any object with spatial dimensions smaller than $s \times s$ reduces to sub-pixel scale in the output feature map and is therefore difficult to detect. To facilitate

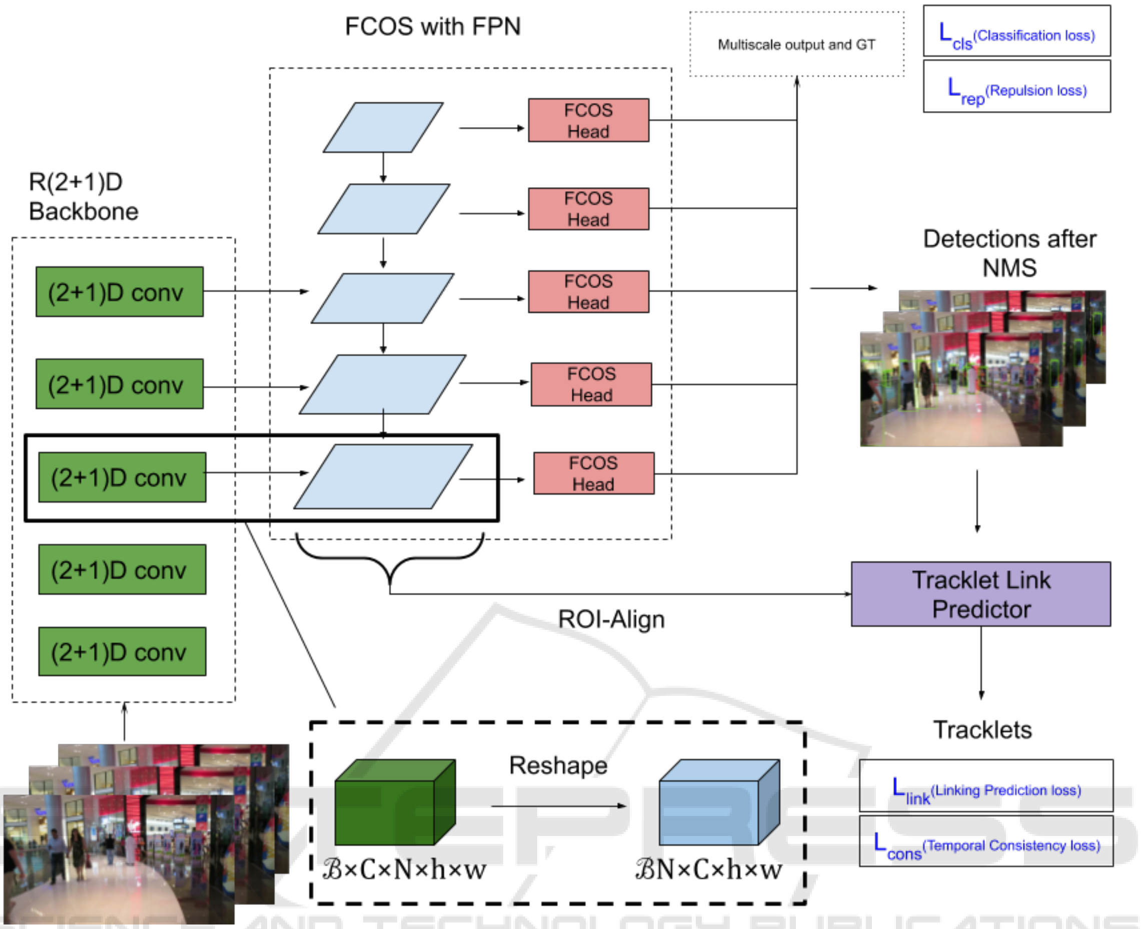

Figure 1: Overview of the proposed approach. A 3D backbone preprocesses a set of videos. The preprocessed feature maps are reshaped into a batch of 3D feature maps which are then fed to a FCOS detector (Tian et al., 2019) implemented in FPN paradigm. The bounding boxes obtained after NMS are pooled using ROI-align (He et al., 2017) and are fed to the Tracklet Link Predictor.

processing under limited computational constraints, s is usually set to a high number in 3D backbones such as 14 in R(2+1)D (Tran et al., 2018) and therefore is unsuitable for detecting any pedestrian less than 14 pixels in height. Existing work on dilated residual networks (Yu et al., 2017) shows that reduction in s leads to higher resolution feature maps capable of producing better classification and segmentation performance. We therefore set the output stride of R(2+1)D to 4, thereby making it theoretically possible to detect even very small-scale pedestrians. The feature maps from the backbone are fed to the detection head which is next described.

4.2 Detection Head

Video object detection techniques aim at harnessing video-specific characteristics of objects such as optical flow (Zhu et al., 2017), temporal context (Beery et al., 2020) and visual memory over time (Beery et al., 2020; Liu et al., 2019). Harnessing these characteristics aids in improved performance vis-à-vis *per-frame* detection using 2D backbones. Harnessing the aforementioned characteristics often comes at the price of *high parameter count* and *complex architecture* which precludes their efficient usage with a tracking module. For instance, the state-of-art CR-CNN (Beery et al., 2020) model makes use of 2 sets of memory banks along with a *per-frame* FRCNN (Ren et al., 2015) and multiple self-attention modules. This makes the resulting model large thereby making it difficult to use it in a longer pipeline involving object tracking. In our work we rather adopt a middle path of using a static image based object detector utilizing a 3D backbone for *per-frame* detection.

To this end, the output feature map from the R(2+1)D backbone is reshaped to coalesce the batch and frame dimensions to obtain a 3D feature map with a batch size of $N \times \mathcal{B}$ (**figure 1**). Prior use of 3D backbone ensures that the feature maps fed to the detector entail both spatial and temporal characteristics. We utilize FCOS (Tian et al., 2019) as our detector. FCOS is an *anchor-free* detector which predicts bounding boxes at every location in a feature map. The use of a *center loss* ensures a low proportion of *false positives* by assigning bounding boxes far from potential ground truth center, a very low confidence. Our approach is indifferent to the specific choice of a detector and FCOS is chosen for its simplicity, high performance and relative robustness to occlusion (Tian et al., 2019). We utilize FCOS within a FPN (Lin et al., 2017) paradigm to obtain better detection response for pedestrians across scales. To further minimize *false positives* and to avoid multiple duplicate detections we substitute the repulsion loss (Wang et al., 2018) in place of the standard smooth-L1 loss usually used with FCOS.

During training, the detection head is trained using the standard FCOS loss function,

$$L_{det} \triangleq \frac{1}{N_{pos}}(\sum_{a,b} L_{cls}(c_{a,b}, c^{*}_{a,b}) + \mathbb{1}_{c^{*}_{a,b}>0} L_{rep}(t_{a,b}, t^{*}_{a,b})) \tag{2}$$

where (a,b) index feature map locations, L_{cls} is the focal loss, L_{rep} is the repulsion loss, $c_{a,b}$ is the classification probability at a location while $t_{a,b}$ is the regression target prediction at a location. The starred $(*)$ variables denote the corresponding targets. $\mathbb{1}_{c^{*}_{a,b}}$ denotes an indicator function which is 1 for locations where an object of interest i.e pedestrian is present else is 0.

The bounding boxes detected by FCOS after post-processing with NMS are used for discovering tracklets as described in the next section.

4.3 Tracklet Link Prediction

To formalize this linkage operation let us consider that given a bounding box b_i of identity i, $\mathcal{F}(b)$ denotes the frame to which it belongs. Let $\mathbb{B}$ denote the set of all bounding boxes obtained across all N frames. The objective of the linkage operation then is to compute a mapping $\mathcal{R} : \mathbb{B} \times \mathbb{B} \longrightarrow [0,1]$ such that,

$$\mathcal{R}(b_x, b_y) = \begin{cases} 0 & x \neq y \\ 0 & \mathcal{F}(b_x) = \mathcal{F}(b_y) \\ p \in (0,1] & otherwise \end{cases} \tag{3}$$

The third case in equation 3 states that given two bounding boxes which have the same identity and are in two different frames, the mapping $\mathcal{R}(.,.)$ assigns a non-zero probability p indicating that they are likely to represent the same individual. During training target assignment for the computation of the loss function, $p = 1$. During testing, p is predicted by a link prediction approach we describe next.

The tracklet link prediction in our work is done using graph convolutional networks (GCNN). This is motivated by two important observations. Firstly, the collection of bounding boxes from a detector is an unordered set and hence cannot be unambiguously arranged on a regular grid. GCNNs are a natural and popular choice for handling data with an unordered structure (Henaff et al., 2015). Secondly, local context plays an important role in identification tasks such as person re-identification (Li et al., 2019; Farenzena et al., 2010) and tracking.

The nodes V in our graph $G(V,E)$ correspond to each bounding box obtained from FCOS. The node features are obtained by performing ROI alignment (He et al., 2017). The adjacency matrix A for G is defined such that nodes belonging to the same frame are not connected to each other. This reflects the natural assumption that a person can occur only once in each frame. On the other hand, all nodes belonging to different frames are connected as a priori information is not available about linkage. However, during training we perform target assignment for the edge weights of the graph as per equation 3, where we set $p = 1.0$.

Our tracklet link prediction is based on graph attention networks (GAT) (Veličković et al., 2017). GATs are an improvement over traditional GCNNs (Kipf and Welling, 2016) by incorporating *self-attention* mechanism. Self-attention mechanisms allow nodes to attend to features of other nodes. The latent representations for nodes thereby produced are influenced by neighborhood nodes which are highly related to them. This is particularly useful for inductive learning tasks such as link prediction where latent representations based on pairwise relationship are particularly desirable.

We process G with 2 GAT layers, each with 4 attention heads. The feature vectors representing an edge in the output graph are computed using their inner product of the nodes it connects. The edge feature vectors are then processed with 2 fully-connected layers followed by sigmoid activation to obtain the probability value expressing the likelihood of an edge existing between two nodes. Let (q,r) represent the edge connecting nodes q and r in the graph G. During training the link prediction is trained using the standard logistic regression function as follows,

$$L_{link} \triangleq -\frac{1}{|E|} \sum_{(i,j)\in E} y^{*}_{i,j} log y_{i,j} + (1 - y^{*}_{i,j}) log(1 - y_{i,j}) \tag{4}$$

As in equation 2, here the *starred* terms represent the groundtruth values while $y_{i,j}$ represents the predicted probability of the edge between nodes i and j.

During inference, once the links are predicted, a threshold τ is applied and all consecutive temporal

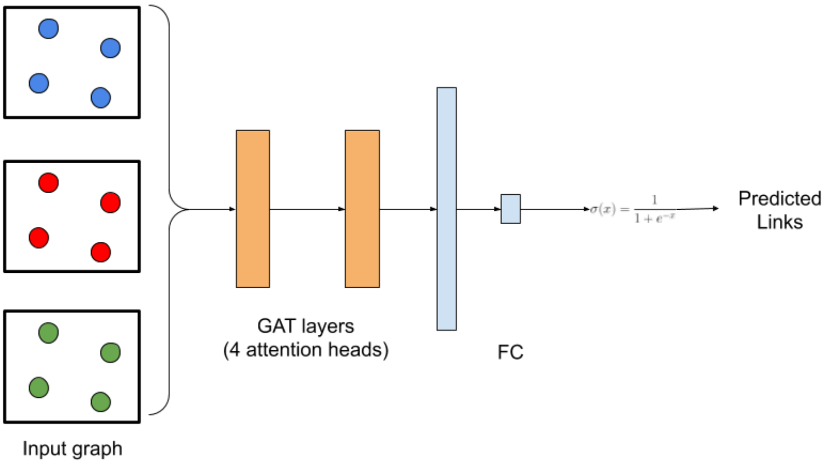

Figure 2: Tracklet link prediction in our work using Graph ATtention networks (GAT) (Veličković et al., 2017). In this figure the case of 3 video frames and $\mathbb{B}$ is shown. Different node colors indicate no connectivity between then while those of different colors are connected according to equation 3.

nodes linked together by edges in G are taken as a tracklet. In this way, the final tracklets along with the corresponding bounding boxes are determined in our approach. Although NMS applied after bounding box detection in FCOS is a non-differentiable operation unlike ROI-align (He et al., 2017). Since, node embeddings in G are obtained using ROI-align, the training is still end-to-end. As was outlined in section 3, ensuring tracklet features corresponding to different identities to be far apart is necessary to ensure good tracklet linkage.

4.4 Temporal Consistency

We enforce temporal consistency described in equation 1 to separate the features learnt for different identities. Following the notations in section 3, let k_i be a tracklet denoting identity i. We define the feature $\mathbb{F}(k_i)$ representing this tracklet as the mean of ROI-align features $\mathcal{E}(.)$ for all bounding boxes within k_i,

$$\mathbb{F}(k_i) \triangleq \mu(\mathcal{E}(b)), b \in k_i \tag{5}$$

We enforce the constraints of equation 1 using triplet loss (Dong and Shen, 2018) terms. Given n tracklets, the triplet loss term can be computed for each of $C(n,2)$ possible pairs of tracklets, such that one is used for extracting the *anchor* and positive sample pair, and the other is used for the negative sample. However, $C(n,2)$ blows up rapidly for large values of n with the triplet loss terms thereby dominating the linkage and detection loss terms. We therefore, sample $T \triangleq min(t, C(n,2))$ tracklet pairs randomly, where t is a hyper-parameter. For MOT17 (Leal-Taixé et al., 2015), we found $t = 20$ as an appropriate value.

Let (k_i, k_j) be one of the T tracklet pairs. We use k_i for extracting the anchor and positive terms of the triplet loss. k_j is used for extracting the negative terms of the triplet loss. The anchor term is taken as $\mathbb{F}(k_i)$ and the negative term is the embedding of one randomly chosen bounding box from k_i. $\mathbb{F}(k_j)$ is the negative term. During training over multiple epochs, many possible combinations of anchor, positive and negative terms are thereby used for triplet calculation thereby mitigating the approximation invoked by the random sampling of T and random choice of samples for triplet calculation. The temporal consistency loss in our work can be written as,

$$L_{consistent} \triangleq \sum_{u=0}^{T-1} L^{u}_{triplet} \tag{6}$$

where $L^{u}_{triplet}$ is the triplet loss term computed for the u^{th} pair.

We process a complete video in batches to obtain tracklets and then use the features representing a tracklet as defined in equation 5 to link the tracklets together using GMMCP (Dehghan et al., 2015).

5 TRAINING

The total loss function for our approach is written as

$$L_{tot} \triangleq L_{det} + L_{link} + L_{consistent} + L_{regularization} \tag{7}$$

where $L_{regularization}$ is the $L2$ regularization losses from the complete architecture. Our implementation is done in PyTorch (Paszke et al., 2017) and our experiments are performed on 2 NVidia V100 GPUs. During training, we perform data augmentation using horizontl flipping, brightness and contrast variations – all done randomly. We utilize stochastic gradient descent (SGD) for training with an initial learning rate (lr) of

Table 1: Tracking results on MOT17: The symbol ↑ indicates higher values are better, and ↓ implies lower values are favored. **Bold** entries indicates best results.

Method	**Detr**	**MOTA↑**	**IDF↑**	**MT↑**	**ML↓**	**FP↓**	**FN↓**	**IDS↓**
SCNet	Priv	60.0	54.4	**34.4**	**16.2**	72230	**145851**	7611
LSST(Feng et al., 2019)	Pub	54.7	**62.3**	20.4	40.1	26091	228434	**1243**
Tracktor(Bergmann et al., 2019)	Pub	53.5	52.3	19.5	36.3	12201	248047	2072
Tracktor++V2(Bergmann et al., 2019)	Pub	56.3	55.1	21.1	36.3	**8866**	235,449	1987
JBNOT(Henschel et al., 2019)	Pub	52.6	50.8	19.7	25.8	31572	232659	3050
FAMNet(Chu and Ling, 2019)	Pub	52.0	48.7	19.1	33.4	14138	253616	3072
TubeTK(Pang et al., 2020)	w/o	**63.0**	58.6	31.2	19.9	27060	177483	4137
JDMOT(Wang et al., 2020)	w/o	56.4	42.0	16.7	40.8	17421	223974	4572
Ours	w/o	58.7	56.9	28.7	20.2	38556	189612	4830

0.005 for first 10K iterations following which lr is reduced to 0.0005. The lr is then decayed by 5 times after every 50K iterations until the loss value is stablized. For NMS we use a threshold of 0.6 while for thresholding the tracklets (sec. 4.3), we set $\tau = 0.5$.

6 EXPERIMENTS

We demonstrate the tracking performance of our proposed DeTracker on 2 datasets focusing on pedestrian tracking. We use a subset of the JTA (Fabbri et al., 2018) dataset, which we call as the *small-JTA* in addition to the MOT17(Milan et al., 2016) dataset as described in the following.

Table 2: Comparison of detection results on MOT17Det challenge.

Method	**AP**
JDMOT (Wang et al., 2020)	0.81
Tracktor (Bergmann et al., 2019)	0.72
Ours	72

6.1 Datasets

Multi Object Tracking Challenge-17 (MOT17). The multi-object tracking challenge (MOT17) benchmark (Leal-Taixé et al., 2015; Milan et al., 2016) contains a set of challenging videos with varying points of view, different object sizes, motion and frame rate. 14 videos in total, 7 for training and 7 for testing. For tracking, MOT Challenges have two separate tracks in the leader board: public and private. Methods in the public track use public detections provided by the challenges while methods in the private track can use their own detections. As our method performs joint detection and data association, we compare against state-of-art published methods from both public and private tracks.

Joint-Track-Auto (JTA) Dataset. While good for comparing the performance of a model against the literature, the MOT17 dataset is not very appropriate for an ablation study: it does not contain any validation videos and has only 7 training videos, which are too few to further split into train and val. Furthermore, the number of submissions to the MOT17 leaderboard is limited to 4. Hence we carry out our ablation studies on a synthetic dataset: Joint Track Auto (JTA) (Fabbri et al., 2018). JTA is a dataset annotated for human pose estimation and tracking and its videos are auto generated using the Grand Theft Auto (GTA) game engine. The dataset contains 512 videos of 500k frames and with almost 10M accurately annotated. Since the dataset is not annotated with person bounding boxes, we adapt the human pose estimation annotations as follows. We take the 22 body joints of each person and fit a tight bounding box around them, a similar process to (Amato et al., 2019). For ablation studies we randomly select 25 videos which we denominate *small-JTA*.

Evaluation Metrics. We use the CLEAR (Kasturi et al., 2009) and IDF1 (Ristani et al., 2016) metrics for MOT evaluation. For object detection, we report the Average Precision using the official MOT17Det evaluation protocol. To compare with state-of-art methods, we evaluate on the test set by submitting our results to the official MOT test server. Also, we divide the provided train set into two subsets: one for training and one for validation, and use the validation set for ablation study.

6.2 Results and Comparative Analysis

Evaluation Multi-Object Tracking. We show MOT performance of our proposed method in MOT17 test set and compare with published state-of-the-art methods in table 1. Our approach has achieved comparable results compared to other methods outlined in table 1. Most methods featured on the MOT17 challenge use public detections provided with the training data or utilize external detectors for bounding box extraction. On that basis, TubeTK (Pang et al., 2020) and JDMOT (Wang et al., 2020) are most suitable joint detection and tracking approaches comparable to ours. Our method outperforms JDMOT in all metrics except FP. Compared to TubeTK (Pang et al., 2020), our method performs worse though without using any post-processing approach such as TubeNMS. Unlike TubeTK, our approach does not use 3D heads for detection and bounding boxes are detected *per-frame*. The use of 3D heads causes TubeTK to output bounding box tubes directly at different scales. While this relegates the task of linking bounding boxes to the detector, it brings in additional complexity. For instance, suppression of duplicate tubes requires two hyperparameters, which is difficult to tune. On the other hand our method has a single 2D NMS which has one single parameter.

6.3 Ablation Studies

We perform ablation studies on the *small-JTA* dataset due to its smaller size and also due to a limitation of 4 maximum submissions to MOT challenge (Leal-Taixé et al., 2015) website. We perform a total of 5 ablation studies; each reflecting the relevance of one important component of our framework as described below

Impact of Triplet Loss:

Table 3: Impact of incorporating triplet loss (Dong and Shen, 2018) in equation 7 on MOTA of small-JTA dataset. Improvement (in % is computed as (with - without)/(without).

Triplet Loss	MOTA
Without	0.5437
With	**0.5679**
Improvement	4.45%

Table 3 shows that a 4.45% improvement in MOTA is observed on incorporation of the triplet loss in equation 7. As explained in section 4.4, the anchor and positive samples of triplet loss are taken as the tracklet feature and a randomly chosen bounding box (*of the same identity* feature, all bounding box features corresponding to the same identity are driven towards their mean which is the feature of the entire tracklet. As a result, triplet loss incorporation in equation 7, assists in creating distinct clusters of features; each cluster corresponding to a unique identity. This helps in minimizing identity switches as tracklets are linked across different minibatches of a video.

Impact of Self-attention GAT Layers:

Table 4: Impact of using GAT layers for tracklet link prediction vis-à-vis directly predicting links by computing inner product of ROI-aligned features of bounding boxes.

Self-attention	MOTA
Without	0.5229
With	**0.5679**
Improvement	8.6%

To study the quantitative impact of using GAT layers for tracklet link prediction, we directly used the graph $G(V,E)$ in section 4.3 and created edge features by inner product. Table 4 shows that about 8.6% improvement is obtained in MOTA metric on usage of GAT layers. Self-attention mechanism in GAT (Veličković et al., 2017) ensures semantically more meaningful graph transformation thereby improving edge features in case of occlusion.

Impact of Repulsion Loss:

Table 5: Repulsion loss boosts up to 3.99%.

Regression Loss	AP
Smooth-L1	0.6713
Repulsion	**0.6981**
Improvement	3.99%

Repulsion loss (Wang et al., 2018) plays an important role in our approach as shown in table 5 where the AP on the samll-JTA dataset improves by nearly 4%. Repulsion loss decreases the number of false positive detections and therefore reduces the number of irrelevant nodes in our graph.

Impact of Feature Pyramids:

Table 6: FPN provides a significant nearly 5.6% improvement in AP over small-JTA dataset.

FPN	AP
Without	0.6610
With	**0.6981**
Improvement	5.6%

Feature pyramids add some memory overhead as feature maps from multiple layers need to be stored for top-down and lateral processing. Table 6 shows that on the small-JTA dataset, we notice (table 6) a significant improvement of 5.6% in terms of AP. Therefore though the added memory overhead prevents us from processing large number of video frames simultaneously, FPN enables us to capture more true-positives thereby improving the overall performance of our approach.

Number of Simultaneous Frames Processed:

Table 7: Effect of changing number of frames simultaneously processed by our approach.

# Frames	MOTA
4	**0.5679**
6	0.5658
Change from 4 to 6 frames	-0.3%

Due to added memory overhead by FPN, we are unable to process large number of frames simultaneously. It is of interest to process large number of frames to capture long-term spatio-temporal nature of tracklets. As shown in table 7, we observe the performance dip slightly on moving from 4 to 6 frames. We believe this to be due to a more complex graph as well as due to training perturbations. We aim for a more exhaustive analysis of this ablation to a future work.

7 DISCUSSION

In order to understand the benefits and caveats of DeTracker, it is necessary to analyze how the different aspects of our architecture compare with state-of-start approaches.

7.1 Feature Extraction

In DeTracker, we utilize ROI-Alignment (He et al., 2017), this method allows us to perform feature extraction directly from our networks' embedding tensors without needing extra layers. In contrast, existing tracking methods extract features by feeding the cropped bounding-boxes to multiple layers in a reid fashion scheme, these networks output similarity scores between patches of ids that are needed for data association. This approach increases the complexity of the framework, adds cumbersome time, and in most cases needs multiple training stages. It is true that revisiting the bounding-box for feature extraction improves considerably the similarity measures but at what cost? It is possible for DeTracker to adopt these re-id implementation as many trackers (Bergmann et al., 2019; Ristani and Tomasi, 2018) do, but in the spirit of simplifying the models DeTracker uses a straightforward solution that does not need multiple training stages for feature extraction.

7.2 Post-processing Heuristics

In order to have online tracking algorithms we need to only look at past and present frames, DeTracker is truly online. Methods such (Bergmann et al., 2019; Pang et al., 2020) claim to be online but due to post-processing and additional steps to create final trajectories are only near-online. In (Pang et al., 2020) the overlapping of frames is a crucial factor that gives the method a leeway in which multiple btubes need to be processed in order to create final trajectories. The online status of (Bergmann et al., 2019) is incumbent upon the size (i.e *number of layers*) of the ReID networks incorporated and moreover requires external data for effective training. Therefore a change in the ReID network to incorporate real life runtime constraints can potentially disrupt the online nature of (Bergmann et al., 2019), while this is not the case with our approach.

8 FUTURE WORK

To conclude our analysis, we bring forward two approaches how to utilize DeTracker as a starting point for future research.

DeTracker with Extensions. Apply DeTracker to a given set of tracklets and extend it with tracking specific methods, given that we already proportion embeddings for each tracklet identity. Scenarios with large and highly visible objects will be covered by the frame to frame bounding box regression. For the remaining, it seems most promising to implement a motion model, taking into account the individual movements of objects. In addition, such a motion predictor reduce the necessity for an advanced killing policy.

Tracklet Generation. Analogous to tracking-by-detection, we propose a tracking-by-tracklet approach. Indeed, many algorithms already use tracklets as input, as they are richer in information for computing motion or appearance models. However, usually a specific tracking method is used to create these tracklets. We advocate the exploitation of our tracklet detector itself, not only to create sparse detections, but clip to clip tracklets. The remaining complex tracking cases ought to be tackled by a subsequent tracking method.

In this work, we have formally defined those hard cases, analyzing the situations in which not only our method but other dedicated tracking solutions fail. And by doing so, we question the current focus of research in multi-object tracking, in particular, the missing confrontation with challenging tracking scenarios.

9 CONCLUSION

In this paper we propose an approach to jointly detect and track multiple targets in a video stream. Our work capitalises on existing developments in backbone architectures, object detection and attention mechanisms to facilitate end-to-end training of both detection and multi-target tracking. This reinforces the idea that detection and tracking are complementary operations and thus indeed can be performed together. We view this work as a solid baseline to which refinements which form our future work can give way to better and faster frameworks which can be integrated with other high-level computer vision tasks.

REFERENCES

Amato, G., Ciampi, L., Falchi, F., Gennaro, C., and Messina, N. (2019). Learning pedestrian detection from virtual worlds. In *International Conference of Image Analysis and Processing (ICIAP)*, pages 302–312.

Beery, S., Wu, G., Rathod, V., Votel, R., and Huang, J. (2020). Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13075–13085.

Berclaz, J., Fleuret, F., Turetken, E., and Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33.

Bergmann, P., Meinhardt, T., and Leal-Taixé, L. (2019). Tracking without bells and whistles. In *The IEEE International Conference on Computer Vision (ICCV)*.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Chu, P. and Ling, H. (2019). Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6171–6180.

Dehghan, A., Modiri Assari, S., and Shah, M. (2015). Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4099.

Dong, X. and Shen, J. (2018). Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 459–474.

Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., and Cucchiara, R. (2018). Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*.

Fang, K., Xiang, Y., Li, X., and Savarese, S. (2018). Recurrent autoregressive networks for online multi-object tracking. pages 466–475.

Farenzena, M., Bazzani, L., Perina, A., Murino, V., and Cristani, M. (2010). Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2360–2367. IEEE.

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2017). Detect to track and track to detect. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3057–3065.

Feng, W., Hu, Z., Wu, W., Yan, J., and Ouyang, W. (2019). Multi-object tracking with multiple cues and switcher-aware classification. *CoRR*, abs/1901.06129.

Gori, M., Monfardini, G., and Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Henaff, M., Bruna, J., and LeCun, Y. (2015). Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

Henschel, R., Zou, Y., and Rosenhahn, B. (2019). Multiple people tracking using body and joint detections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Jiang, H., Fels, S., and Little, J. J. (2007). A linear programming approach for multiple object tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., and Wang, X. (2017). Object detection in videos with tubelet proposal networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–897.

Kang, K., Ouyang, W., Li, H., and Wang, X. (2016). Object detection from video tubelets with convolutional neural networks.

Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., and Zhang, J. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336.

Kim, C., Li, F., Ciptadi, A., and Rehg, J. M. (2015). Multiple hypothesis tracking revisited. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4696–4704.

Kim, H.-U. and Kim, C.-S. (2016). Cdt: Cooperative detection and tracking for tracing multiple objects in video sequences. volume 9910, pages 851–867.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kuo, C. and Nevatia, R. (2011). How does person identity recognition help multi-person tracking? In *CVPR 2011*, pages 1217–1224.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. (2015). MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*. arXiv: 1504.01942.

Lenz, P., Geiger, A., and Urtasun, R. (2014). Followme: Efficient online min-cost flow tracking with bounded memory and computation.

Li, J., Wang, J., Tian, Q., Gao, W., and Zhang, S. (2019). Global-local temporal representations for video person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3958–3967.

Li Zhang, Yuan Li, and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Liu, M., Zhu, M., White, M., Li, Y., and Kalenichenko, D. (2019). Looking fast and slow: Memory-guided mobile video object detection. *arXiv preprint arXiv:1903.10172*.

Mahadevan, S., Athar, A., Ošep, A., Hennen, S., Leal-Taixé, L., and Leibe, B. (2020). Making a case for 3d convolutions for object segmentation in videos. In *BMVC*.

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*. arXiv: 1603.00831.

Milan, A., Leal-Taixé, L., Schindler, K., and Reid, I. (2015). Joint tracking and segmentation of multiple targets. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5397–5406.

Osep, A., Mehner, W., Voigtlaender, P., and Leibe, B. (2018). Track, then decide: Category-agnostic vision-based multi-object tracking. pages 1–8.

Pang, B., Li, Y., Zhang, Y., Li, M., and Lu, C. (2020). Tubetk: Adopting tubes to track multi-object in a one-step training model. In *CVPR*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Pirsiavash, H., Ramanan, D., and Fowlkes, C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. pages 1201 – 1208.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking.

Ristani, E. and Tomasi, C. (2018). Features for multi-target multi-camera tracking and re-identification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6036–6046.

Sadeghian, A., Alahi, A., and Savarese, S. (2017). Tracking the untrackable: Learning to track multiple cues with long-term dependencies. pages 300–311.

Sun, S., Akhtar, N., Song, X., Song, H., Mian, A., and Shah, M. (2020). Simultaneous detection and tracking with motion modelling for multiple object tracking. *ECCV*.

Tang, S., Andriluka, M., Andres, B., and Schiele, B. (2017). Multiple people tracking by lifted multicut and person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3701–3710.

Tian, Z., Shen, C., Chen, H., and He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., and Leibe, B. (2019). Mots: Multi-object tracking and segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7934–7943.

Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., and Shen, C. (2018). Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7774–7783.

Wang, Y., Weng, X., and Kitani, K. (2020). Joint detection and multi-object tracking with graph neural networks. *arXiv preprint arXiv:2006.13164*.

Wang, Z., Zheng, L., Liu, Y., and Wang, S. (2019). Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*.

Xiu, Y., Li, J., Wang, H., Fang, Y., and Lu, C. (2018). Pose Flow: Efficient online pose tracking. In *BMVC*.

Xu, Y. and Wang, J. (2019). A unified neural network for object detection, multiple object tracking and vehicle re-identification. *ArXiv*, abs/1907.03465.

Yu, F., Koltun, V., and Funkhouser, T. (2017). Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480.

Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., and Yang, M.-H. (2018). Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382.

Zhu, X., Wang, Y., Dai, J., Yuan, L., and Wei, Y. (2017). Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417.