Chained-Tracker: Chaining Paired Attentive Regression Results for End-to-End Joint Multiple-Object Detection and Tracking

Sajjad Pakdamansavoji (savoji@yorku.ca)

April 12, 2022

Abstract

This report is written for the reproducebility challenge of the deep learning course at York University. The aim of reproducebility challenge is to dive deep into the implementation of a recent paper. The paper I chose is titled Chained-Tracker which was published in the ECCV 2020.

Current Multiple-Object Tracking (MOT) models do not satisfy the criterion to be called an end-to-end pipeline. These models either follow the tracking-by-detection paradigm to conduct object detection, feature extraction and data association separately, or have two of the three subtasks integrated to form a partially end-to-end solution. This work is presented to overcome the previous sub-optimal frameworks which should implicitly integrate all the aforementioned subtasks. It chains paired bounding boxes regression results estimated from overlapping nodes, of which each node covers two adjacent frames. The paired regression is made attentive by object-attention (brought by a detection module) and identity-attention (ensured by an ID verification module). The two major contributions of this works are as follows: chained structure and paired attentive regression, make CTracker simple, fast and effective, setting new MOTA records on MOT16 and MOT17 challenge datasets (67.6 and 66.6, respectively), without relying on any extra training data. Please note that in this source code only main contributions of the authors are implemented while parts which were adopted from previous works were imported from their works. To be more specific the three-branched neural model was implemented in pytorch.My code is publicly available at `https://github.com/SajjadPSavoji/CTracker`.

1 Introduction

Existing MOT solution, regardless of the research made in the past, still suffer from two main obstacles. Firstly, most solutions follow thee tracking-by-detection paradigm [1], meaning that the detection and tracking are performed

separately. While this method might be plausible but is lacks the possibility of global optimization as its main modules are decoupled. It usually contains three sequential subtasks: object detection, feature extraction and data association. Performing this task in isolated subtasks may lead to local optima and more computational cost compared with an end-to-end solution. Furthermore the errors in the first sub module will be propagated into later modules which will consequently affect the performance of the down steam task itself. For instance the performance of data association heavily relies on the quality of object detection.

Secondly, recent methods while trying to gain better overall performance has made this module more and more complex. For instance attention and re-identification are two tricks that have proved to be helpful. Re-identification (or ID verification) is used to extract more robust features for data association. Attention helps the model to be more focused, avoiding the distraction by irrelevant yet confusing information. Although adding these components to the model will improve the performance, their computational over head should not be overlooked.

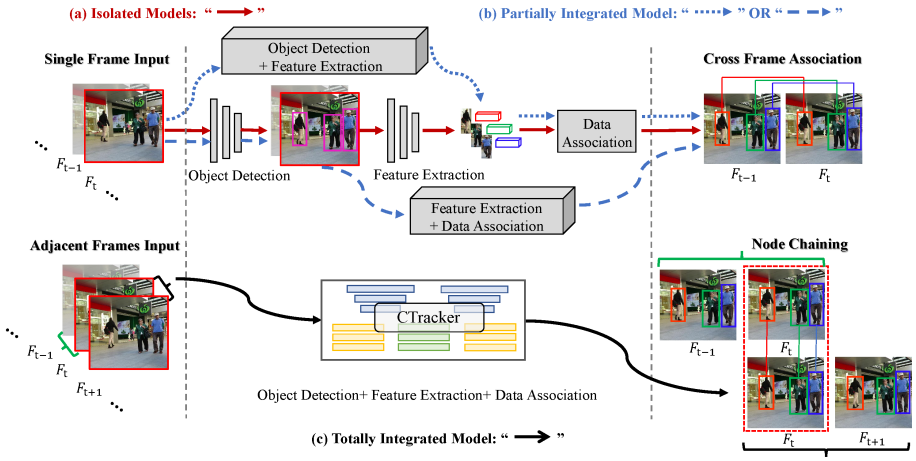

Figure 1: **Comparison of CTracker with other typical MOT methods**

To address the insufficiency of previous works, the authors proposed an online tracking method named Chianed-Tracker which gathers object detection, feature extraction, and data association into a single module. As seen in Fig. 1, it takes in adjacent frames as input and performs a regression to obtain a joint bounding box for the targets that appears in both frames.

To further boost the performance of this model, authors proposed a joint attention module using confidence scores of classification and re-identification branches. These attention modules guides the regression branch to focus on

informative spatial regions with two other branches. To be more specific the classification branch guides the regression branch to focus on the foreground regions while the ID verification attention helps the regression branch to focus on regions corresponding to the same target. Then, the generated box pairs belonging to the adjacent frame pairs could be associated using simple methods like IoU (Intersection over Union) matching [2].

The proposed models benefits from end-to-end optimization of detection and tracking at the same time. This feature makes this novel model superior compared with the previous methods. Also, performing detection and tracking at the same time will result in better feature extraction and representation thought the network's layers. The contributions of this study can be summarized into three parts:

1. Proposing an end-to-end online Multiple-Object Tracking model
2. Designing a joint attention module to highlight informative regions
3. Achieving the S.O.T.A performance on detection of MOT16 and MOT17.

2 Related Work

2.1 Detection-based MOT Methods

Yu *et. al* [3] proposed the POI algorithm, which conducted a high-performance detector based on Faster R-CNN [4] by adding several extra pedestrian detection datasets. Chen *et. al* [5] incorporated an enhanced detection model by simultaneously modeling the detection-scene relation and detection-detection relation, called EDMT. Furthermore, Henschel *et. al* [6] added a head detection model to support MOT in addition to original pedestrian detection, which also needed extra training data and annotations. Bergmann *et. al*

2.2 Partially End-to-end MOT Methods

Lu *et. al* [7] proposed RetinaTrack, which combined detection and feature extraction in the network and used greedy bipartite matching for data association. Sun *et. al* [8] harnessed the power of deep learning for data association in tracking by jointly modeling object appearances and their affinities between different frames. Similarly, Chu *et. al* [9] designed the FAMNet to jointly optimize the feature extraction, affinity estimation and multi-dimensional assignment. Li *et. al* [10] proposed TrackNet by using frame tubes as input to do joint detection and tracking.

2.3 Attention-assistant MOT Methods

Chu *et. al* [11] introduced a Spatial-Temporal Attention Mechanism (STAM) to handle the tracking drift caused by the occlusion and interaction among targets. Similarly, Zhu *et. al* [12] proposed a Dual Matching Attention Networks

(DMAN) with both spatial and temporal attention mechanisms to perform the tracklet data association. Gao *et. al* [13] also utilized an attention-based appearance model to solve the inter-object occlusion.

3 Methodology

3.1 Problem Settings

Given an image sequence $\{F_t\}_{t=1}^{N}$ with totally N frames, Multiple-Object Tracking task aims to output all the bounding boxes $\{\mathcal{G}_t\}_{t=1}^{N}$ and identity labels $\{\mathcal{Y}_t^{GT}\}_{t=1}^{N}$ for all the objects of interest. $F_t \in \mathbb{R}^{c\times w\times h}$ indicates the t-th frame, $\mathcal{G}_t \subset \mathbb{R}^4$ represents the ground-truth bounding boxes of the K_t number of targets in t-th frame and $\mathcal{Y}_t^{GT} \subset \mathbb{Z}$ denotes their identities. Most of the recent MOT algorithms divide the MOT task into three components.

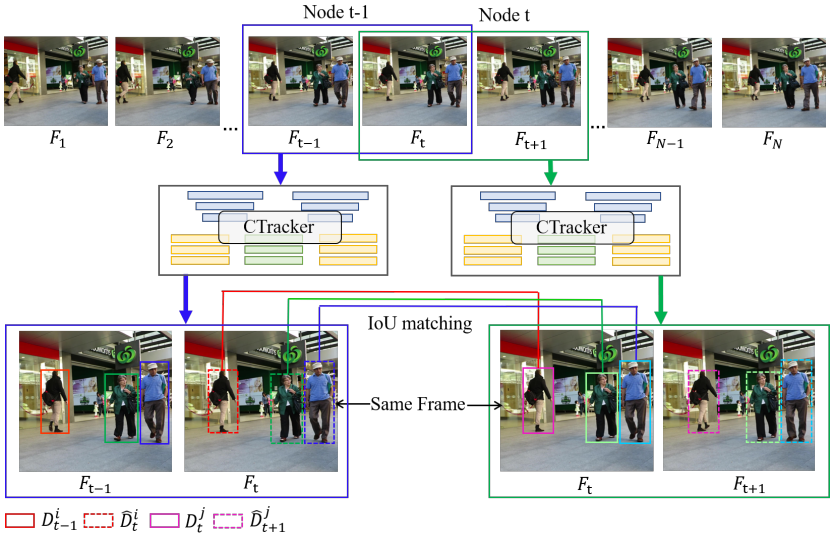

Figure 2: **Illustration of the node chaining**. After generating bounding box pairs by CTracker for two arbitrary adjacent nodes, we chain these two nodes by doing IoU matching on the shared common frame.

3.2 Chained-Tracker Pipeline

Framework. CTracker model requires two adjacent frames for instance the node (F_{t-1}, F_t) and (F_t, F_{t+1}). Given these nodes it can generate bounding box pairs $\{(D_{t-1}^i, \hat{D}_t^i)\}_{i=1}^{n_{t-1}}$ and $\{(D_t^j, \hat{D}_{t+1}^j)\}_{j=1}^{n_t}$. As depicted in Fig. 2, assume that there are only slight shift in the detected boxes $\hat{D}_t^i$ and D_t^j of the common frame, one can use a simple matching strategy to chain the two boxes, instead of using complicated appearance features as in canonical MOT methods.

Node chaining. The node chaining is done in the following steps. Firstly, in the initial node, all detected bounding box $D_1^i \in \mathcal{D}_1$ is initialized as a tracklet with a random identity. Later on, for any node $t > 1$, they chain the adjacent nodes (F_{t-1}, F_t) and (F_t, F_{t+1}) by calculating the IoU (Intersection over Union) between the boxes in $\hat{\mathcal{D}}_t$ and $\mathcal{D}_t$ see Fig. 2. Getting the IoU metric, the matching is done by applying the Kuhn-Munkres (KM) algorithm [14]. For each matched box the previous tracklets are updated accordingly. Any unmatched box D_t^k is initialized as a new tracklet with a new identity.

Robustness enhancement. To further boost the model's robustness to partial occlusions and short-term disappearing, they keep the unmatched tracklets for up to σ frames and keep looking for possible matches in the upcoming frames. To estimate the location of objects in the future frames, they are using a constant speed model[15, 16].

Effectiveness and limitations. Ctracker is effective in cases in which targets appear or disappear; however its effectiveness heavily depends on the hyperparameters used in this model. As for its limitations one should note two major shortcomings. Firstly, the model has a built-in one frame latency. For instance if target is not in frame $t-1$ but appears in frame t, it will not be detected in the node (F_{t-1}, F_t). Secondly, the matching part does not have a gradient, so the pipe line can not be optimized end-to-end.

3.3 Network architecture

The proposed model has three branches; bounding box regressor, classification, and re-identification. It adopts ResNet-50 [17] plus a Feature Pyramid Networks(FPN) for feature extraction. Having the high-level features of each frame, they are concatenated together and fed to the three branches of the network. As can be seen in Fig. 3, the paired boxes regression branch generates a box pair for each target, and the object classification branch predicts a score for each pair indicating the confidence of being foreground. To help the paired boxes regression branch to avoid the distraction by irrelevant yet confusing information, the object classification branch and the extra ID verification branch are used for attention guidance.

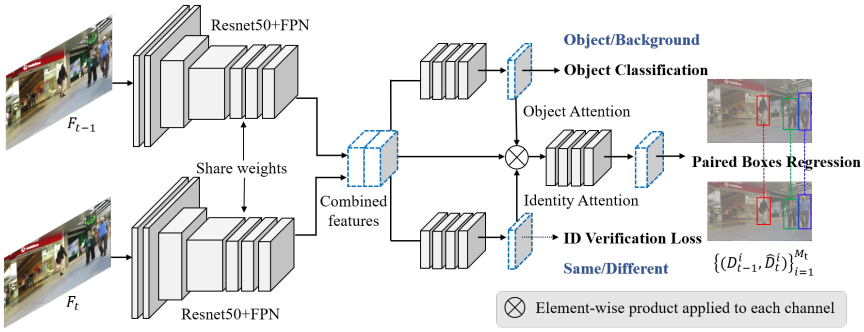

Figure 3: **Network architecture of CTracker**. Given two adjacent frames, we firstly use two backbone branches with tied weights to extract the features. Then we concatenate features of the two frames on channel level and the combined features are used to predict the paired boxes.

3.4 Label Assignment and Loss Design

Given a chain node (F_t, F_{t+1}), and the a chained-anchor $A^i_t = (x^{t,i}_a, y^{t,i}_a, w^{t,i}_a, h^{t,i}_a)$, They used ground-truth bounding box matching strategy similar to that of SSD [18]. They use a matrix M to denote the result of such a matching based on which they assign the ground-truth label c^i_{cls} to CTracker's classification branch. Using the same information one can generate the identification labels as well. As for the regression part, they follow the faster RCNN loss to regress the offset of anchors to grand-truth as illustrated in equation 3. The over all loss is given in equation 4 where $\mathcal{F}(.,.)$ is the focal losses

$$c^i_{\text{cls}} = \begin{cases} 1, & \text{if } \Sigma^{K_t}_{j=1} M_{ij} = 1, \\ 0, & \text{if } \Sigma^{K_t}_{j=1} M_{ij} = 0, \end{cases} \tag{1}$$

$$c^i_{\text{id}} = \begin{cases} 1, & \text{if } c^i_{\text{cls}} = 1 \textbf{ and } \mathcal{I}[G^j_t] = \mathcal{I}[G^k_{t+1}], \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

$$\begin{aligned} &L_{reg}(\Delta^{t,i}_d, \Delta^{t+1,i}_{\hat{d}}, \Delta^{t,j}_g, \Delta^{t+1,k}_g) \\ &= \sum_{l \in \{x,y,w,h\}} \left[\text{smooth}_{L_1}(\Delta^{t,i}_{d,l} - \Delta^{t,j}_{g,l}) + \text{smooth}_{L_1}(\Delta^{t+1,i}_{\hat{d},l} - \Delta^{t+1,k}_{g,l}) \right] /8, \end{aligned} \tag{3}$$

$$L_{all} = \sum_{t,i} \left[L_{reg}(\Delta^{t,i}_d, \Delta^{t+1,i}_{\hat{d}}, \Delta^{t,j}_g, \Delta^{t+1,k}_g) + \alpha \mathcal{F}(p^i_{cls}, c^i_{cls}) + \beta \mathcal{F}(p^i_{id}, c^i_{id}) \right], \tag{4}$$

4 Experiment

4.1 Datasets and Evaluation Metrics

This experiment is mainly conducted on two public dataset from the MOT Challenge called MOT16 [19] and MOT17. Both of these data-set include the same images which are 7 training sequences and 7 testing sequences. In public detection MOT17 includes DPM, Faster R-CNN [4] and SDP [20] detectors. MOT Challenge is the standard benchmark for evaluating MOT models; as such, the metrics used in this challenge are also the standard metrics for evaluating the MOT models. These metrics are gathered under CLEAR MOT Metrics [21], including Multiple-Object Tracking Accuracy (MOTA), Multiple-Object Tracking Precision (MOTP), the total number of False Negatives (FN), False Positives (FP), Identity Switches (IDS), and the percentage of Mostly Tracked Trajectories (MT), Mostly Lost Trajectories (ML). Among these metrics, MOTA is the primary metric to measure the overall detection and tracking performance.

4.2 Benchmark Evaluation

To evaluate the model via quantitative metrics, the authors present its performance on the test set of MOT17 in Table **??**. The results of my implementation are also added in the last row of the Table. Comparing the numbers in this table it is obvious that CTracker achieves the best overall performance in private tracking setting a new bar on MOTA, MT, and FP. Its performance on other metrics such as IDF1, MOTP, ML, and IDS, while not being the best, can compete with the previous models.

Table 1: **Tracking results of CTracker on MOT17 train-set**.

Sequence	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓
MOT17-02	0.50	0.44	0.85	14%	14%	3476	12209	189
MOT17-04	0.88	0.78	0.87	66%	2%	7651	11711	216
MOT17-05	0.64	0.66	0.82	41%	21%	909	3010	212
MOT17-09	0.68	0.81	0.86	13%	0%	1412	2965	66
MOT17-10	0.72	0.57	0.81	31%	0%	3700	6100	246
MOT17-11	0.76	0.61	0.87	38%	8%	2345	4183	145
MOT17-13	0.80	0.75	0.81	71%	5%	2030	34333	276

Table 2: **Tracking results of MyTracker on MOT17 train-set**.

Sequence	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓
MOT17-02	0.51	0.43	0.85	12%	13%	3725	12451	179
MOT17-04	0.87	0.78	0.86	63%	3%	8358	11590	241
MOT17-05	0.64	0.66	0.83	42%	20%	896	3055	214
MOT17-09	0.68	0.53	0.87	14%	0%	1308	2918	64
MOT17-10	0.73	0.59	0.81	33%	0%	2510	4580	149
MOT17-11	0.75	0.57	0.87	31%	0%	2510	4580	149
MOT17-13	0.81	0.73	0.81	76%	4%	2296	3593	289

5 Reproduciblity Challenges

While this study offers a wealth of information regarding their model and its training procedure, there are still some shortcomings that makes reproducing its results difficult if not impossible. The following are the key challenges faced:

- Weight initialization technique and random seeds are not given.
- For anchor boxes they used the KMeans clustering with unknown K.
- They did not discuss how they are reducing the learning weight.
- They trained their for 100 epochs which takes a lot of time.

To address these short comings I chose the above mentioned hyper parameters arbitary. The training curve of my implementation is given below.

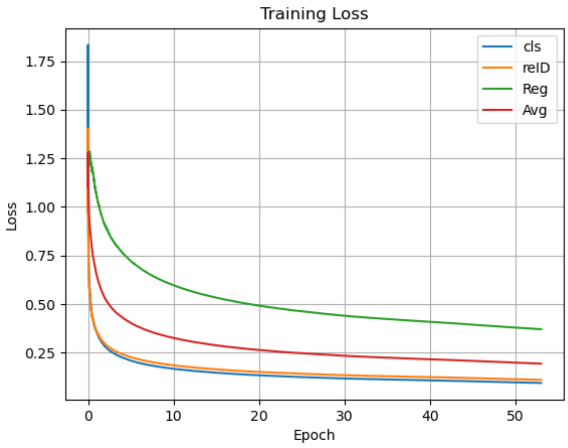

Figure 4: Learning curve of my implementation during 55 epochs

6 Conclusion

The authors suggest an end-to-end object detection and tracking model which takes advantage of attention within its branches. The final reproducability resutl is that due to insufficient information regarding models hyper parameters and its training procedure the results could not be fully replicated; however, choosing arbitrary values for missing information I was able to train this model and get a less impressive result. It is worth mentioning that I was only able to train the model for 55 epochs while the authors indicate training their model for 100 epochs.

References

[1] Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV. (2009)

[2] Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: AVSS. (2017)

[3] Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: Poi: multiple object tracking with high performance detection and appearance feature. In: ECCV. (2016)

[4] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. (2015)

[5] Chen, J., Sheng, H., Zhang, Y., Xiong, Z.: Enhancing detection model for multiple hypothesis tracking. In: CVPRW. (2017)

[6] Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multi-object tracking. In: CVPRW. (2018)

[7] Lu, Z., Rathod, V., Votel, R., Huang, J.: Retinatrack: Online single stage joint detection and tracking. In: CVPR. (2020)

[8] Sun, S., Akhtar, N., Song, H., Mian, A.S., Shah, M.: Deep affinity network for multiple object tracking. TPAMI (2019)

[9] Chu, P., Ling, H.: Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: ICCV. (2019)

[10] Li, C., Dobler, G., Feng, X., Wang, Y.: Tracknet: Simultaneous object detection and tracking and its application in traffic video analysis. arXiv preprint arXiv:1902.01466 (2019)

[11] Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., Yu, N.: Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In: ICCV. (2017)

[12] Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.H.: Online multi-object tracking with dual matching attention networks. In: ECCV. (2018)

[13] Gao, X., Jiang, T.: Osmo: Online specific models for occlusion in multiple object tracking under surveillance scene. In: ACMMM. (2018)

[14] Kuhn, H.W.: The hungarian method for the assignment problem. NRL (1955)

[15] Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP. (2017)

[16] Peng, J., Wang, T., Lin, W., Wang, J., See, J., Wen, S., Ding, E.: Tpm: Multiple object tracking with tracklet-plane matching. PR (2020)

[17] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

[18] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. (2016)

[19] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)

[20] Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: CVPR. (2016)

[21] Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. JIVP (2008)