ONLINE MULTI-OBJECT TRACKING WITH CONVOLUTIONAL NEURAL NETWORKS

Long Chen[1], Haizhou Ai[1], Chong Shang[1], Zijie Zhuang[1], Bo Bai[2]

[1]Tsinghua National Lab for Info. Sci. & Tech. (TNList),
Department of Computer Science and Technology, Tsinghua University, Beijing, China, 100084.
[2]Huawei Technologies, Beijing, China
{l-chen16, shang-c13, zhuangzj15}@mails.tsinghua.edu.cn
ahz@mail.tsinghua.edu.cn, baibo3@huawei.com

ABSTRACT

In this paper, we propose a novel online multi-object tracking (MOT) framework, which exploits features from multiple convolutional layers. In particular, we use the top layer to formulate a category-level classifier and use a lower layer to identify instances from one category under the intuition that lower layers contain much more details. To avoid the computational cost caused by online fine-tuning, we train our appearance model with an offline learning strategy using the historical appearance reserved for each object. We evaluate the proposed tracking framework on a popular MOT benchmark to demonstrate the effectiveness and the state-of-the-art performance of our tracker.

Index Terms— Multi-object Tracking, Convolutional Neural Network, Appearance Model

1. INTRODUCTION

Multi-object tracking (MOT) aims to estimate trajectories of multiple objects in the same category in videos. Driven by advances in object detection, tracking-by-detection approaches are increasingly popular for MOT. These approaches can be categorized into batch methods and online methods, where batch methods utilize detections from future frames and online methods predict object states using only observations from the current as well as past frames. In this work, we focus on online MOT, which is suitable for time-critical applications such as visual surveillance, robot navigation and automated driving.

Recent studies in many computer vision tasks, e.g. image classification [1, 2] and object detection [3, 4, 5], have made great progress with the aid of convolutional neural networks (CNNs). Multi-object tracking also benefits from CNNs with more accurate detectors. However, CNNs have not yet been fully exploited in MOT. Most of the existing tracking-by-detection approaches only consider the final output of the detector but ignore intermediate CNN features. These approaches focus on data association strategies, namely, linking detections to tracklet. They have two shortcomings: the lack of the discriminative ability to deal with occlusion between interacting targets and suffering from detection failures. A typical way to solve these problems is to maintain an instance-specific appearance model for each tracked object and fine-tune the model during the test time using image patches sampled around the target. Considering the unacceptable computational cost for training CNNs online, most existing multi-object trackers use only low-level hand-crafted features.

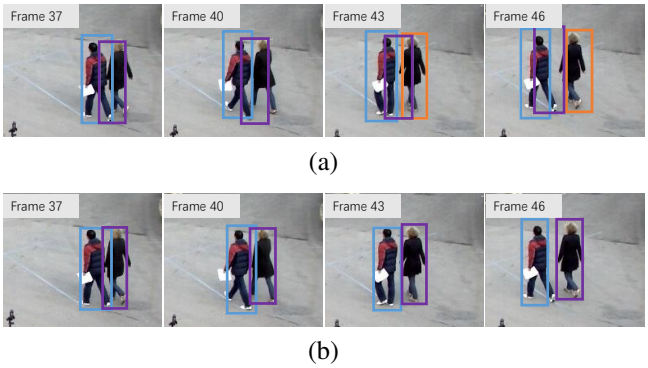

Fig. 1. Tracking results. Different colors represent different identities. (a) Tracking results with only the top convolutional layer. (b) Tracking results with multiple layers.

In this work, we extract features from a CNN-based detector to formulate the appearance model. We exploit multiple convolutional layers from the detector, with the intuition that different layers can extract features at different levels [6, 7]. For instance, the top convolutional layer tends to capture high-level semantic information, which is suitable for distinguishing objects and backgrounds but less discriminative to instances in the same category. In our experiments, the target tends to drift to adjacent objects when only the features from the top layer are used (Fig. 1 (a)). On the other hand, the lower layer can provide more details to tackle occlusion between interacting targets. Fig. 1 (b) shows tracking results using both the top and the lower layers. Moreover, we adopt

an offline learning strategy, which makes use of the historical appearance reserved for each object, to avoid the computational cost caused by online fine-tuning.

Our contribution in this work is three fold:

(1) We propose a CNN-based appearance model for online multi-object tracking, which jointly exploits features at different levels from two convolutional layers.

(2) We use an offline learning strategy to avoid the computational cost for online fine-tuning, making the time consumption manageable.

(3) We develop a unified framework that can effectively perform detection and online tracking with the convolutional features computed once per image.

2. RELATED WORK

Tracking-by-Detection. Recent studies in online MOT focus on the tracking-by-detection framework. These approaches formulate MOT as a data association problem, of which the main task is to link detections to tracked objects [8, 9, 10]. On the other hand, Breitenstein et al. [11] use the continuous confidence of the detector as well as an online trained appearance model for robust multi-person tracking. Online trained appearance models are employed by recent works to tackle intra-category occlusion [12, 11, 13].

CNNs in Tracking. The majority of previous multi-object trackers consider only low-level hand-crafted features while MOT with CNNs is under fully explored. Existing CNN-based trackers are developed primarily for single-object tracking, namely, tracking one target which is specified in the first frame [6, 14, 15, 7]. In [14], rich feature hierarchies of CNNs are transfered for single-object tracking. Furthermore, Wang et al. [6] use multiple convolutional layers to estimate the foreground heat map and locate the target. These two methods update the appearance model by fine-tuning CNNs online, while Held et al. [16] train the network in an entirely offline manner. In this paper, we adopt CNNs to address the online multi-object tracking problem.

3. APPROACH

In this section, we first introduce the CNN-based appearance model, and then our online multi-object tracking framework. We present the offline learning approach at the end.

3.1. CNN-based Appearance Model

Recent successful object detectors, such as Fast R-CNN [3], Faster R-CNN [4] and SDP [5], often utilize a region-based convolutional network (R-CNN) [3] as a feature extractor, and then apply a classifier (fully connected layers, FC layers) on the extracted features.

To exploit CNNs for online MOT, we extract features from multiple convolutional layers of the recent Faster-RCNN detector (with VGG-16 model [2]), to formulate two classifiers, a category classifier and an instance classifier. The architecture of the proposed tracking framework is illustrated in Fig. 2. The top convolutional layer (Conv5-3), followed by a region of interest (RoI) pooling layer [3], is used to extract high-level semantic features for each input RoI in the image. Similar to the Faster R-CNN, we feed the extracted features into the category classifier, which consists of two FC layers and an output layer to estimate $p(obj|\mathbf{I}, \mathbf{x})$. We define $\mathbf{x} = (x, y, u, v)$, i.e. the position and size, as a candidate region, and $p(obj|\mathbf{I}, \mathbf{x})$ as the foreground probability of the candidate $\mathbf{x}$. As for online tracking, we use this probability to prevent the estimate from drifting to the background as detailed in 3.2.

The instance classifier with historical appearance is introduced into the framework, to effectively identify intra-category objects without online fine-tuning steps. The object appearance represents a fixed-length feature vector, which is extracted from a RoI pooling layer attached to Conv3-3 followed with a FC layer. We reserve the appearance of each tracked object at the initial time and update it during the online tracking. When a new frame comes, the historical appearance and the new feature vector extracted from each candidate region are concatenated and fed into the instance classifier to estimate $p(o_t^i|\mathbf{I}, \mathbf{x}, o_{t-1}^i)$, which stands for the probability of the same object. The main benefit of sharing convolutional layers with the detector is that we can effectively perform detection and online tracking with the convolutional features computed once per image.

3.2. Online Multi-Object Tracking

Online tracking can generally be decomposed into two parts, a motion model and an appearance model. Given the previous state of each tracked object, the motion model generates a set of candidate regions in the current frame. The appearance model is subsequently adopted to classify these candidate regions and estimate the new location for each object. We adopt the CNN-based appearance model described above as well as the motion model based on the particle filter [17] to construct our online multi-object tracking framework.

The particle filter is a common approach to estimate hidden states of targets with a sequential Bayesian inference. Each particle consists of a tuple $(\mathbf{x}, w)$, where $\mathbf{x}$ is a candidate region and w stands for the importance weight. At each time step t, we re-sample a fixed number of particles for each tracked object with a constant velocity motion model. Then the weight of each particle is estimated by our appearance model:

$$w_t \propto p(o_t^i|\mathbf{I}, \mathbf{x}, o_{t-1}^i) \cdot I(p(obj|\mathbf{I}, \mathbf{x}) > \tau_w), \qquad (1)$$

where $I(v)$ denotes an indicator function that returns 1 if v is true, otherwise returns 0, $\mathbf{x}$ is a candidate region and τ_w is a specified threshold.

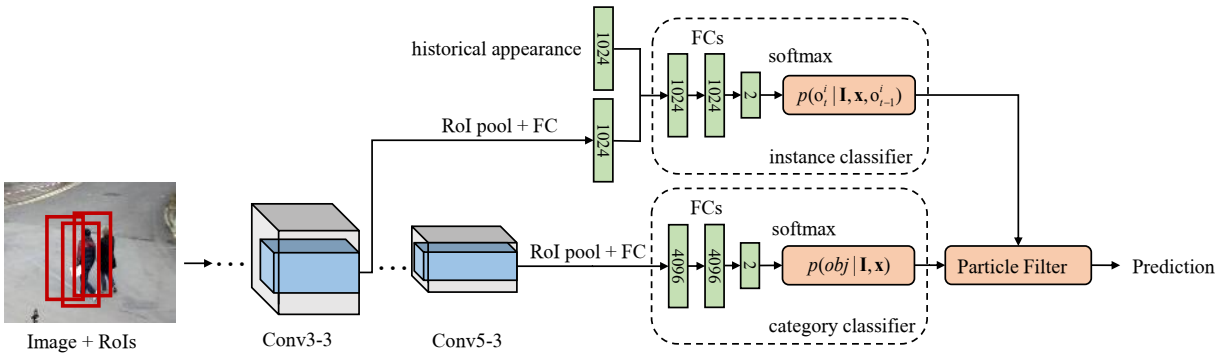

Fig. 2. The architecture of the proposed online MOT framework.

The multi-object tracking procedure in our framework contains three steps. For an incoming frame, we first estimate the new location of each object by the particle filter and the appearance model. Then detections that do not overlap with tracked objects are used to initialize new objects and retrieve missing objects. We update the historical appearance for each object at the end. To avoid introducing the background noise, we replace the reserved appearance only if $p(obj|\mathbf{I}, \mathbf{x}) > \tau_u$, where $\tau_u > \tau_w$ is the update threshold.

3.3. Learning Appearance Model

In order formulate the detection and tracking in a unified framework, we adopt the Faster R-CNN detector that is trained on the VOC2007 dataset [18], and keep the parameters fixed. That is, we only train the network following Conv3-3, which consists of a RoI pooling layer, three fully connected layers, and an output layer. The first FC layer is adopted to encode CNN features as the object appearance, while the following layers constitute the instance classifier.

A training example comprises a pair of appearances, the historical and the current. During the tracking process on training videos, feature vectors from candidate regions and corresponding historical appearances are collected as training examples. In each frame, we randomly select 30% of RoIs from all particles that have intersection over union (IoU) with the ground-truth bounding box of at least 0.6 as positive examples, and take the same number of RoIs, of which IoUs are less than 0.5, as negative examples. After collecting 10k training examples, we use the Adam optimizer, with a mini-batch size of 128 and a learning rate of 1e-5, to minimize the cross-entropy loss for 10 epochs. Then we repeat the training procedure 10 times by re-sampling training examples with the network just learned.

4. EXPERIMENTS

Dataset. In this section, we evaluate our online tracking framework on the MOT15 benchmark [19]. This pedestrian tracking dataset contains a training set and a test set each with 11 video sequences. To confirm the effectiveness of the proposed appearance model, we perform experiments on a validation set of 6 sequences that separated from the training set. We further evaluate our tracker on the test set and compare it with state-of-the-art online MOT methods.

Table 1. Comparisons of tracking performances on validation set when one of the classifiers is disabled.

Tracker	MOTA↑	MT(%)↑	FAF(%)↓	IDS↓
Full model	**39.0**	20.0	**1.04**	**114**
w/o Instance	32.8	15.4	1.17	190
w/o Category	34.8	**21.8**	1.42	119
SORT [9]	34.0	20.5	1.24	274

Evaluation metrics. We adopt widely used metrics to evaluate the proposed tracker, which include multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP) [20], false alarm per frame (FAF), the number of mostly tracked targets (MT, $> 80\%$ recovered), the number of mostly lost targets (ML, $< 20\%$ recovered) [21], fragmented trajectories (Frag), false positives (FP), false negatives (FN), and identity switches (IDS).

4.1. Performance Validation

To demonstrate the effectiveness of the proposed appearance model with two classifiers formulated from different convolutional layers, we examine the tracking performance on the validation set when one of these classifiers is disabled. Results are reported in Table 1, where the arrow after each metric indicates that higher (↑) or lower (↓) value is better. SORT [9] is a baseline tracker where the data association is carried out without any appearance model. In this experiment, we utilize detections from the Faster R-CNN for all trackers for a fair comparison. As shown in Table 1, our tracker with the full model improves by about 5% in MOTA and suffers

Table 2. Online tracking performances on MOT15 test set.

Tracker	Det	MOTA↑	MOTP↑	MT(%)↑	ML(%)↓	FP↓	FN↓	FAF↓	IDS↓	Frag↓
EAMTT [10]	Priv	**53.0**	75.3	**35.9**	19.6	7538	**20590**	1.3	776	1269
MDP_SubCNN [8]	Priv	47.5	74.2	30.0	**18.6**	8631	22969	1.5	628	1370
SORT [9]	Priv	33.4	72.1	11.7	30.9	7318	32615	1.3	1001	1764
TDAM [22]	Pub	33.0	72.8	13.3	39.1	10064	30617	1.7	464	1506
MDP [8]	Pub	30.3	71.3	13.0	38.4	9717	32422	1.7	680	1500
SCEA [23]	Pub	29.1	71.1	8.9	47.3	6060	36912	1.0	604	**1182**
oICF [24]	Pub	27.1	70.0	6.4	48.7	7594	36757	1.3	**454**	1660
AP_RCNN (Ours)	Priv	**53.0**	**75.5**	29.1	20.2	5159	22984	0.9	708	1476
AP_RCNN_pub (Ours)	Pub	38.5	72.6	8.7	37.4	**4005**	33204	**0.7**	586	1263

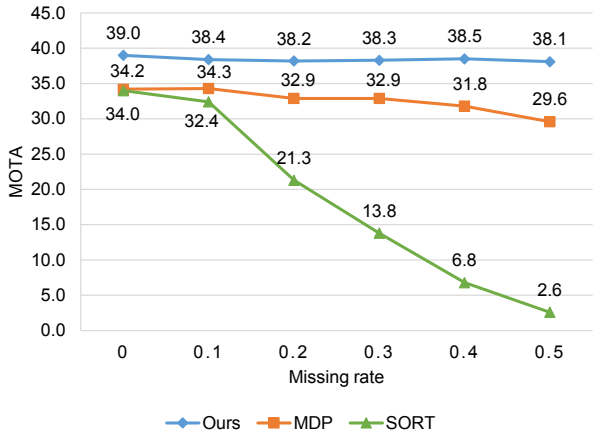

Fig. 3. MOTA at different detection missing rates. Our tracker suffers less from detection failures.

less from identity switch (IDS) than in SORT. As for different classifiers, IDS is nearly twice that of the full model when the instance classifier is disabled. On the other hand, when the category classifier is disabled, the increasing of FAF demonstrates the usage of this classifier: preventing the target from drift to backgrounds. These two classifiers in different levels both are crucial components in our framework.

Our tracker suffers less from detection failures since detections in the framework are only used to initialize new objects and retrieve missing objects. We confirm this by applying different detection missing rates as shown in Fig 3. Detections from the Faster R-CNN are randomly discarded with the missing rate of 0, 0.1, 0.2, 0.3, 0.4, and 0.5, respectively. As the missing rate increases, there is a significant performance drop on SORT since it associates detections without any appearance information. MDP [8] formulates the online MOT problem as decision making in Markov Decision Processes and performs template tracking with optical flows. When the missing rate is 0.5, MOTA of MDP is reduced by about 5%, while it of our tracker decreases by less than 1%.

4.2. Evaluation on Testing Set

We name the proposed tracker as AP_RCNN (appearance model with region-based convolutional neural network) and evaluate it on the MOT15 benchmark. Experimental results are shown in Table 2. Note that the tracking performance is heavily dependent on the detector, the direct comparison of tracking methods with different detectors is meaningless. For a fair comparison, we test our tracking methods based on detections provided by the benchmark (AP_RCNN_pub) and detections collected by EAMTT [10] (AP_RCNN), respectively. *Pub* and *Priv* in the table indicate which type of detections is used for tracking, public detections provided by the benchmark or the private detector. As for tracking with public detections, our tracker achieves the best score and improves about 5% in MOTA compared with state-of-the-art methods.

5. CONCLUSION

We address online multi-object tracking with a CNN-based appearance model, where CNN features from multiple layers are utilized to learn a category-level classifier and an instance-level classifier. To prevent the target from drifting to the background, high-level semantic features extracted from the top layer are used for the category classifier. We further tackle intra-category occlusion with the instance classifier, which is formulated by the lower convolutional layer. In the test time, we reserve and update object appearances instead of fine-tuning the network online, to trade off between the instance-specific discriminative ability and the computational complexity. Moreover, our model allows the feature sharing between detectors and trackers, making it an efficient unified framework. Experimental results on the MOT15 benchmark demonstrate the effectiveness of the proposed framework.

6. ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (Project Number 61521002).

7. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[3] Ross Girshick, "Fast r-cnn," in *ICCV*, 2015.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[5] Fan Yang, Wongun Choi, and Yuanqing Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *CVPR*. IEEE, 2016.

[6] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," in *ICCV*, 2015.

[7] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015.

[8] Yu Xiang, Alexandre Alahi, and Silvio Savarese, "Learning to track: Online multi-object tracking by decision making," in *ICCV*, 2015.

[9] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016.

[10] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *ECCV Workshop on Benchmarking Multi-Target Tracking*, Amsterdam, Netherlands, 2016.

[11] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool, "Online multi-person tracking-by-detection from a single, uncalibrated camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820–1833, 2011.

[12] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang, "Tracklet association with online target-specific metric learning," in *CVPR*. IEEE, 2014.

[13] Seung-Hwan Bae and Kuk-Jin Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *CVPR*. IEEE, 2014.

[14] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[15] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*. IEEE, 2016.

[16] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," in *ECCV*, 2016.

[17] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," .

[19] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, 2015.

[20] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[21] Yuan Li, Chang Huang, and Ram Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *CVPR*. IEEE, 2009.

[22] Min Yang and Yunde Jia, "Temporal dynamic appearance modeling for online multi-person tracking," *Computer Vision and Image Understanding*, 2016.

[23] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon, "Online multi-object tracking via structural constraint event aggregation," in *CVPR*, 2016.

[24] Hilke Kieritz, Stefan Becker, Wolfgang Hübner, and Michael Arens, "Online multi-person tracking using integral channel features," in *Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016.