Fast Multiple Object Tracking Using Siamese Random Forest without Online Tracker Updating

Jimi Lee, Sangwon Kim, Byoung Chul Ko
Dept. of Computer Engineering
Keimyung University, Daegu, Korea 42601
{jimi88lee,eddiesangwonkim}@gmail.com,{niceko}@kmu.ac.kr
http://cvpr.kmu.ac.kr

Abstract

A Siamese convolution neural network (CNN) is the most popular data association method in the field of object tracking owing to its good matching performance and network sharing support. However, it is unsuitable for real-time tracking in low-end systems because numerous parameters are still required. Unlike with traditional data association methods, to build an efficient co-learning framework in terms of the multiple object tracking (MOT) performance and tracking speed, , we do not apply a CNN-based Siamese structure. Instead, we propose a Siamese random forest (RF) framework that enables high-speed learning and classification by combining RF with Siamese structures. During the learning process, a feature for SiameseRF uses a densely transformed feature map of a CNN network for object detection, and the shared RF is learned in the directions of increasing similarity to the first pair (anchor, positive) and increasing difference between the second pair (anchor, negative). Unlike a CNN, the two SiameseRFs do not share weights, but do share a rule structure that composes a tree. The proposed method was successfully applied to various MOT benchmark datasets while maintaining a robust tracking performance despite the camera movement or crowded pedestrians.

1. Introduction

Multiple object tracking (MOT) is essential for various tracking technologies such as video surveillance, autonomous driving, a human-computer-interface (HCI), and augmented reality (AR). Many online and offline MOT tracking techniques have recently applied a deep neural network (DNN) instead of a conventional tracking technique. However, offline tracking is unsuitable for real-time object monitoring or other applications because all frames must be considered to verify the tracking path. For online MOT, Kalman- or particle-filter based methods have mainly been used [1], although studies on DNN-based MOT have recently produced remarkable results [2–6]. As a common point, both offline and online MOT commonly use the tracking-by-detection (TBD) paradigm. However, regardless of how good the detection method is, if an object is missed or an inaccurate object is detected owing to an occlusion of the object or camera shaking, the tracking performance can be significantly deteriorated. Therefore, various data association methods have been proposed to compensate for the inaccuracy of MOT detection. Real-time tracking in MOT is closely related to the efficiency of the data association. Siamese convolutional neural network (CNN) [2][5][6][7] based trackers have received significant interest in real-time tracking. A Siamese CNN applies the same network to the detection and tracker and calculates the similarity based on the difference in the output feature value. Therefore, a Siamese CNN does not need to maintain a separate network structure and has an advantage of fast tracking. Although a Siamese structure shows a good matching performance between objects, the shared network for similarity matching still has a large number of hyper parameters and a slow tracking speed owing to the complex network structure when combined with a CNN.

2. Related Studies

In studies on MOT tracking, long-term appearance models using features from a DNN [8], DeepMatching [9], and a quadruplet convolutional neural network [10] have /demonstrated a better tracking performance. However, such methods are unsuitable for online tracking because the network structure is complicated and the object tracking path of multiple frames must be analyzed.

Tracking using a Siamese CNN for person re-identification in MOT has recently been studied [2][5][6][7]. A Siamese CNN applies the same network to the detection and tracker and calculates the similarity in the difference between output function values. Therefore, a Siamese CNN does not need to maintain a separate network structure and has the advantage of fast tracking. Although a Siamese CNN shows a good matching performance between objects, a shared network for similarity matching still includes numerous hyper parameters and a slow tracking speed owing to a complex network structure. Therefore, Siamese CNN-based MOT methods may be infeasible for real-time tracking in a real-world environment.

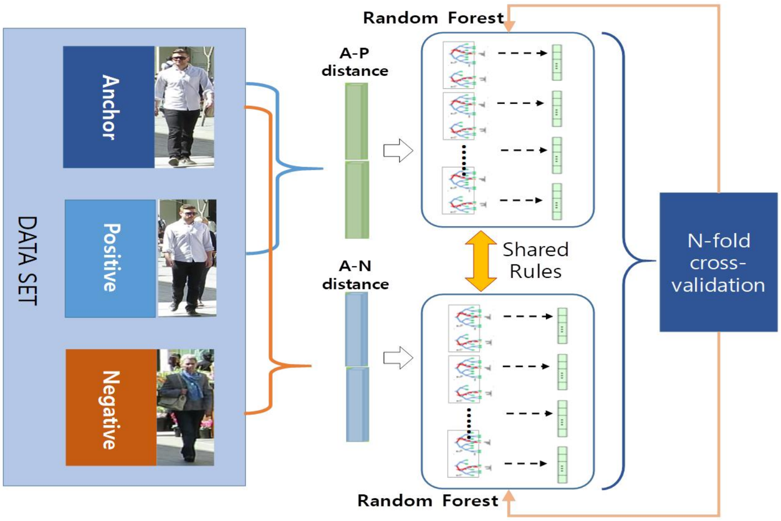

Fig. 1. The procedure of SiameseRF learning. Two RFs receive A-P and A-N distance vectors of {*anchor, positive*} and {*anchor, negative*} pairs as input. The shared RF is learned in the direction of increasing similarity to the first pair and increasing difference to the second pair.

3. Siamese Random Forest

To build an efficient joint learning framework in terms of the MOT performance and tracking speed, unlike existing methods, we do not use a CNN-based Siamese structure. Instead, we propose a Siamese Random Forest (RF) framework that combines an accurate RF with a Siamese structure with a high-speed learning and classification. When an object is detected, it must be input into a Siamese RF to measure its similarity to existing trackers. The most basic but important step for measuring the similarity is currently the feature extraction step. RF shows an extremely good performance for tabular data but has a disadvantage in terms of a poor performance when applying unconditioned data such as images and video. Therefore, an optimal feature extraction step that can effectively distinguish objects should be applied as a preprocess of a Siamese RF. In this study, we apply global averaging pooling (GAP) to feature maps obtained from different layers of darknet53, the backbone network of YOLOv3 [11], to reduce the computation time for feature extraction. The GAP method is used to reduce the spatial dimensions of a 3D tensor and has the advantage of minimizing an overfitting by reducing the total number of parameters in the model. We extract partial feature maps of the first and second layers corresponding to the bounding box (bbox) of darknet. Next, two 1 × 1 × C feature vectors are generated by applying GAP to each partial feature map. The created features are called condensed features, and two condensed features are concatenated to become one final condensed appearance feature (CAF).

During the learning process of SiameseRF, an initial RF consisting of L ensemble trees is created. Although two RFs receive {*anchor, positive*} pairs and {*anchor, negative*} pairs as input, both RFs share the same structure. Therefore, during the learning process, the shared RF is learned in the direction of increasing similarity to the first pair and in the direction of increasing difference to the second pair, as shown in Fig. 1. Unlike a Siamese CNN, the two RFs do not share weights, but they do share rules composing a tree. As the input of each RF, the CAF difference, which are the appearance features of each image, is input as a feature. A vector $AP \in \mathbb{R}^{1\times m}$ is a distance vector *if* and *only if* the following holds:

$$\mathrm{AP}_i = d(a_i,\, p_i),\ \mathrm{AN}_i = d(a_i, n_i), for\ 1 \le i \le \mathrm{m}, \tag{1}$$

where d is an L2 distance function, $a_i \in anchor$, $p_i \in positive, n_i \in negative$, and m is the number of samples in each label of *anchor, positive,* and *negative.*

To repeat the learning phase of a shared RF, in this study, a K-fold cross validation for the training process is adopted to improve the accuracy of the model. The K-fold cross-validation method automatically determines the optimal rule number and parameters while reducing the risk of an over-fitting. The learning process of Siamese RF is as follows:

- Step 1: K-1 folds are selected from the entire training dataset S, and the other folds are used for the validation set.
- Step 2: The AP-distance vector for a sample pair {*anchor*, *positive*} and the AN-distance vector for a sample pair {*anchor*, *negative*} are estimate using the CAF.
- Step 3: The training sample pairs are input into each RF sharing the rules. The decision to update rules consisting of a shared RF depends on whether the K-fold cross-validation converges.
- Step 4: RF composed of L trees in step *p* is trained using the A-P and A-N distance vectors.
- Step 5: After the training of K-1 folds, A-P and A-N distance vectors are obtained from the pairs of {*anchor*, *positive*} and {*anchor*, *negative*} samples in the

validation set. The triplet loss using Eq. (2) is calculated by applying the AP and AN distance vectors to the learned RF. This process is performed for all n pairs in the validation set.

$$L(a,p,n)_{\mathrm{k}} = max(\|1-\mathrm{RF}_k(AP)\|^2 - \|\mathrm{RF}_k(AN)\|^2 + \alpha, 0), \quad (2)$$

- Step 6: Store the learned RF structure and total loss J. Steps 1–6 are repeated until each K fold has been used as the testing fold.

$$J_k = \sum_{i=1}^{n} L(a_i, p_i, n_i)_k^i \quad (3)$$

- Step 7: When the learning is completed for all K-folds, the RF with the smallest total loss J is determined as the final Siamese RF.

$$k = arg \min_{k\in K} J_k \quad (4)$$

After training a SiameseRF through K-fold verification, during the actual tracking, the CAF extracted from a detection and a tracker are input into the learned SiameseRF, and the similarity probability of the two objects becomes the appearance score for an association.

In every frame, detections are assigned to the tracker based on the Hungarian method and three measures, namely, the inverse probability value of SiameseRF ($\hat{P}_{Siam}$), aspect ratio (A_{ratio}), and L1-center distance (Dis). Finally, for a cost function of the association matching, we combine three distance measures using a weighted sum:

$$\mathrm{c}(\mathrm{tr}^{\mathrm{i}}, \mathrm{d}^{\mathrm{j}}) = \alpha \cdot \hat{\mathrm{P}}_{\mathrm{Siam}}(\mathrm{tr}^{\mathrm{i}}|\mathrm{d}^{\mathrm{j}}) + \beta \cdot \mathrm{A}_{\mathrm{ratio}}(\mathrm{tr}^{\mathrm{i}}, \mathrm{d}^{\mathrm{j}}) + \gamma \mathrm{Dis}(\mathrm{tr}^{\mathrm{i}}, \mathrm{d}^{\mathrm{j}}), \quad (5)$$

where α, β, and γ denote the weights, which are 0.6, 0.2, and 0.2, respectively, and these weights were found based on several experiments.

If the detected object and tracker are matched, the state of the tracker is updated by combining the states of the current tracker and detection. If the tracker does not find a match during τ frames, the object is considered to have disappeared and is deleted. Using the same method, if the detected object does not match any tracker, the object is assigned as a potential tracker, and if a match occurs between the tracker and detected object over τ frames, it is assigned as a new tracker; otherwise, a false detection is recognized and the object is removed.

4. Experiment

4.1. MOTSChallenge 3: Tracking Only

The proposed algorithm was successfully applied to MOTSChallenge workshop 2020 benchmark video sequences captured from a stereo camera, which include multiple objects in various environments. Specifically, in the high accuracy detection set provided in advance, the proposed algorithm yields a considerably accurate tracking performance in terms of sMOTSA [12], i.e., 1) 60% for MOTS20 Pedestrians, 2) 71.4% for KITTI Cars, and 3) 60.9% for KITTI Pedestrians, respectively. In terms of the computation time, it took an average of 8.2 fps for the MOTS20 dataset and 12.4 fps for the KITTI dataset.

4.2. MOT16

We also measured the tracking results for MOT16 data. For the experiment, the same image sequences of MOTS Challenge 2020 were used, where Yolov3 was used as a detector, and the given MOT16 training data were used for learning. In Table 1, the results of comparative experiments of the MOT16 test dataset show that SiameseRF is relatively faster than other MOT algorithms with similar results. In addition, compared to state-of-art online-based MOT methods, the proposed SiameseRF shows excellent results in terms of the overall performance.

TABLE I. RESULTS ON MOT16 TEST SET. BEST IN BOLD.

	Method	MOTA(%)↑	MOTP(%)↑	FAF↓	MT(%)↑	ML(%)↓	FP↓	FN↓	IDsw↓	Frag↓	Hz↑
Offline	LM_CNN[3]	67.4	79.1	1.7	38.2	19.2	10,109	48,435	931	1,034	1.7
	KDNT [14]	68.2	**79.4**	1.9	41.0	**19.0**	11,479	45,605	933	1,093	0.7
	LMP_p[15]	**71.0**	80.2	1.3	**46.9**	21.9	6,213	**44,564**	**434**	**587**	0.5
	HDTR[17]	53.6	80.8	**0.8**	21.2	37.0	**4,714**	79,353	618	833	**3.6**
Online	JCSTD [13]	47.4	74.4	1.4	14.4	36.4	8,076	86,638	1,266	2,697	**8.8**
	Tracktor[16]	56.2	79.2	**0.4**	20.7	35.8	**2,394**	76,844	617	1,411	1.6
	MLT[18]	52.8	76.1	0.9	21.1	42.4	5,362	80,444	**299**	**702**	5.9
	DMAN[6]	46.1	73.8	1.3	17.4	42.7	7,909	89,874	532	1,616	0.3
	RAR16[4]	**63.0**	78.8	2.3	39.9	22.1	13,663	**53,248**	482	1,251	1.6
	Ours	57.9	**79.3**	1.4	28.5	22.1	8,196	66,538	2,051	2,549	7.8

5. Conclusion

In this paper, we proposed SiameseRF, which can be quickly learned and tested with a small number of parameters instead of a Siamese CNN, which is frequently used for a data association in MOT. According to the nature of RF, the proposed SiameseRF uses K-fold validation instead of a back propagation, and thus the learning is fast and optimal tree rules can be generated. In addition, because the rules of the tree constituting the RF are shared with each RF, the memory requirement can be reduced during testing. It was confirmed experimentally that the proposed method can be used for online tracking in embedded systems with limited system resources. In a future study, we will consider how to reduce the system resources while maintaining the performance by distilling the tree rules constituting SiameseRF.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2019R1I1A3A01042506).

References

[1] S. J. Kim, J. Y. Nam, and B.C. Ko. Online Tracker Optimization for Multi-Pedestrian Tracking Using a Moving Vehicle Camera. IEEE Access, 6: pp.48675 - 48687, 2018

[2] S. Lee, and E. Kim. Multiple Object Tracking via Feature Pyramid Siamese Networks. IEEE ACCESS, 7: pp.8181-8194, 2019

[3] M. Babaee, Z. Li, and G. Rigoll. A Dual CNN-RNN for Multiple People Tracking. Neurocomputing, 2019

[4] K. Fang, Y. Xiang, X. Li, and S. Savarese. Recurrent Autoregressive Networks for Online Multi-Object Tracking. IEEE Winter Conference on Applications of Computer Vision (WACV), 2018

[5] B. Cuan, K. Idrissi, and C. Garcia. Deep Siamese Network for Multiple Object Tracking. IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), 2018

[6] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. H. Yang. Online Multi-Object Tracking with Dual Matching Attention Networks. ECCV, 2018

[7] P. Chu and H. Ling. FAMNet: Joint Learning of Feature, Affinity and Multi-dimensional Assignment for Online Multiple Object Tracking. ICCV, 2019

[8] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited, ICCV, 2015.

[9] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Multi-person tracking by multicut and deep matching, ECCV, 2016

[10] J. Son, M. Baek, M. Cho, and B. Han. Multi-object tracking with quadruplet convolutional neural networks, CVPR, 2017

[11] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. arXiv:1804.02767, 2018

[12] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. MOTS: Multi-Object Tracking and Segmentation. CVPR, 2019

[13] W. Tian, M. Lauer, L. Chen. Online Multi-Object Tracking Using Joint Domain Information in Traffic Scenarios. IEEE Transactions on Intelligent Transportation Systems, 21: pp.374-384, 2019

[14] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, J. Yan. POI: Multiple Object Tracking with High Performance Detection and Appearance Feature. BMTT, SenseTime Group Limited, 2016

[15] S. Tang, M. Andriluka, B. Andres, B. Schiele. Multiple People Tracking with Lifted Multicut and Person Re-identification. CVPR, 2017

[16] P. Bergmann, T. Meinhardt, L. Leal-Taixé. Tracking without bells and whistles. ICCV, 2019

[17] M. Babaee, A. Athar, G. Rigoll. Multiple People Tracking Using Hierarchical Deep Tracklet Re-identification. In arXiv preprint arXiv:1811.04091, 2018

[18] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke and Z. Xiong. Multiplex Labeling Graph for Near Online Tracking in Crowded Scenes. IEEE Internet of Things Journal, 2020