IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 14, 2020, accepted October 13, 2020, date of publication October 19, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032252

Online Multiple Object Tracking Based on Open-Set Few-Shot Learning

HAN-UL KIM[1], (Member, IEEE), YEONG JUN KOH[2], (Member, IEEE), AND CHANG-SU KIM[1], (Senior Member, IEEE)

[1]School of Electrical Engineering, Korea University, Seoul 02841, South Korea
[2]Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, South Korea

Corresponding author: Chang-Su Kim (changsukim@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) under Grant NRF-2018R1A2B3003896, in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant NRF-2019R1F1A1062907, and in part by the Institute for Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government (MSIT), Training Key Talents in Industrial Convergence Security, under Grant 2019-0-01343.

ABSTRACT How to make an online tracking model effectively adapt to newly appearing objects and object disappearance as well as appearance variations of target objects from few examples is an essential issue in multiple object tracking (MOT). Learning target appearances from few examples is a few-shot classification problem, while identifications of newly appearing objects and object disappearance has the aspect of open-set classification. In this work, we regard online MOT as open-set few-show classification to address both learning from few examples (few-shot classification) and unknown classes such as new objects (open-set classification). Specifically, we develop an embedding neural network, called VOFNet, consisting of convolutional and recurrent parts, to perform open-set few-shot classification. The convolutional part constructs a feature from an example of a target object and the recurrent part determines a representative feature of a target object from few examples. Then VOFNet is trained to provide effective features for open-set few-shot classification. Finally, we develop an online multiple object tracker based on the combination of VOFNet and the bipartite matching. The proposed tracker achieves 49.2 multiple object tracking accuracy (MOTA) with 28.9 frames per second on MOT17 dataset, which shows a significantly better trade-off between the accuracy and the speed than the existing algorithms. For example, the proposed algorithm yields about 3.17 times faster speed with 0.99 times lower accuracy than recent existing MOT algorithm [1].

INDEX TERMS Multiple object tracking, online tracking, open-set classification, few-shot classification.

I. INTRODUCTION

Nowadays, many applications including self-driving vehicles [2], surveillance systems [3], and crowd analysis [4] require various video processing technologies such as person re-identification [5], video segmentation [6], [7] and efficient feature processing [8]. Multiple object tracking (MOT) [9] is one of the important problems for video analysis to estimate the states (or bounding boxes) of as many objects as possible in a video sequence. Many efforts have been made for developing reliable MOT systems, and tracking-by-detection is one of the most successful approaches. The tracking-by-detection approach decomposes MOT into two subproblems: object detection and data association. An object detector finds objects in a video sequence, and then a data association scheme links the detection results to yield object trajectories. With the recent success of deep learning, an effective object detector can be employed, which determines candidate states of objects reliably and independently from the tracking process. Hence, tracking-by-detection has the advantage of being robust against model drifts and yields promising MOT results.

Despite recent achievements, MOT remains a challenging problem. Especially, when an object detector provides inaccurate detection results due to various difficulties such as occlusion, motion blur, and object deformation, it is hard to identify targets during the data association phase.

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

FIGURE 1. An example of open-set few-shot classification to describe MOT problem. In the third figure, a new object and detection error classes are depicted by red and yellow boxes, respectively.

Many offline (or batch) trackers [10]–[14] attempt to overcome these difficulties based on batch data association, which uses full frames in a video to improve the data association accuracy. In general, they design cost functions to formulate data association as optimization problems and then determine optimal trajectories by minimizing the cost functions. They alleviate adverse effects of inaccurate detection results using the information in full frames, but these noncausal methods cannot be applied in real-world applications that require online and real-time processing.

Contrary to offline trackers, online (or causal) trackers [15]–[19] utilize only previous and current frames to link detection results. The lack of future information, however, has an additional problem, as well as the inaccurate detection problem: when an object exists in only a few previous frames, a tracker cannot use a sufficient number of detections for achieving accurate data association. From this perspective, online MOT can be regarded as a *few-shot classification* problem [20], [21], in which a classifier is designed with only a few examples per class. Few-shot learning techniques [22]–[25] hence can be employed to improve MOT performance.

However, it is not straightforward to formulate MOT as few-shot classification, since MOT requires a methodology to identify newly appearing objects, which do not exist in previous frames. In other words, the MOT problem inherently has the aspect of *open-set classification* [26], [27] in that it should be capable of handling unknown classes.

Our work is motivated by two aspects of multiple object tracking covering open-set classification and few-shot classification. First, we introduce the notion of *open-set few-shot classification* to formulate the online MOT problem. In the open-set few-shot classification, we define two unknown classes: 1) newly appearing objects and 2) detection errors. In other words, we formulate MOT as the $(K + 2)$-way classification, where K is the number of target objects in the previous frame and the other two classes represent new objects and detection errors, respectively. Figure 1 illustrates the open-set few-shot classification system, where detected boxes are categorized into the one of $(K + 2)$ classes.

Second, we propose a novel embedding network, called VOFNet (video object embedding network for few-shot learning), which transforms an image space into an embedding space to perform open-set few-shot classification for MOT. VOFNet consists of two sub-networks: convolutional neural network (VOF-CNN) and recurrent neural network (VOF-RNN). VOF-CNN constructs feature vectors of detection boxes for target objects, while VOF-RNN determines the representative feature vector of each target object by exploiting its temporal information. To train the proposed network, we perform an open-set classification based on feature distances between representative vectors and detection results. Finally, we achieve the online MOT based on the combination of VOFNet and the bipartite matching. Experimental results demonstrate that the proposed tracker yields comparable performance to the conventional state-of-the-art trackers but at a fraction of running times on the MOT17 benchmark [28].

To summarize, this work has three main contributions:

- We introduce the concept of open-set few-shot classification
- We propose the embedding scheme to formulate online MOT as an open-set few-shot classification problem.
- We achieve comparable tracking performance to the state-of-the-art trackers in the recent MOT17 benchmark, while demanding much lower computational complexity.

The remainder of this paper is organized as follows: Section II reviews related work. Section III describes the proposed algorithm. Section IV discusses experimental results. Finally, Section V draws conclusions.

II. RELATED WORK

A. MULTIPLE OBJECT TRACKING

Most MOT algorithms adopt the tracking-by-detection approach, which decomposes the MOT task into two subtasks: 1) detecting objects in each frame and 2) associating them temporally to create trajectories. As Table 1 summarizes, MOT algorithms can be categorized into offline and online trackers. Offline trackers use a batch of frames to formulate an optimization problem for accurate data association. For instance, Jiang *et al*. [10] proposed a linear programming method to minimize a data association cost. Zhang *et al*. [11] formulated the data association as a minimal cost flow problem. Berclaz *et al*. [29] also regarded

TABLE 1. Summary of related multiple object trackers.

Category	Method	Strength	Weakness
Offline (Batch)	[10]–[14], [29]–[37]	High performance	Can not be used for online applications
Online (Causal)	[15]–[19], [38]–[45]	Can be used for online applications	Low performance

the data association as a flow optimization problem and solved it using the k-shortest path algorithm. Milan *et al*. [12] defined a continuous energy for finding target locations in a continuous space. Also, Milan *et al*. [13] formulated a discrete-continuous energy to consider both the association of detections and the reconstruction of continuous target states. Rezatofighi *et al*. [30] employed the joint probabilistic data association [46] to link detection results and existing targets with reasonable computation costs. However, these trackers [10]–[13], [29], [30] utilize simple distances or weak appearance models to compute pairwise similarity between detections. Thus, they provide relatively low performances.

To overcome this limitation, robust pairwise similarity costs for data association have been designed. Choi [31] proposed the aggregated local flow descriptor to encode a relative motion pattern between two objects. Fagot-Bouquet *et al*. [32] adopted sparse representation to construct appearance models of detection boxes. Kim *et al*. [14] used CNN features in the multiple hypothesis tracking (MHT) framework [47]. Leal-Taixé *et al*. [33] developed a Siamese network to encode the spatiotemporal structure between two objects. Son *et al*. [34] introduced quadruplet CNNs to determine the similarity between detections using their labels and temporal distances. Sheng *et al*. [35] proposed the heterogeneous association graph, which fuses high-level detections and low-level superpixels for data association. Sheng *et al*. [36] developed the tracklet hypothesis for the MHT framework and proposed the iterative maximum weighted independent set algorithm to track multiple objects in polynomial time. Keuper *et al*. [37] proposed a correlation co-clustering method that associates low-level point trajectories and high-level detected boxes.

Different from offline trackers, many trackers [15]–[19], [38]–[43] perform tracking online in a causal manner, by exploiting the information in previous and current frames only. Bae and Yoon [15] introduced the tracklet confidence to represent the detectability and continuity of a target object. They designed an online data association algorithm based on the confidence. Chen *et al*. [38] utilized R-FCN detector scores [48] to select reliable candidate bounding boxes. Yoon *et al*. [18] introduced an one-shot learning method for data association and integrated it into the MHT framework.

In [16], [17], [39], RNNs have been utilized for MOT. Milan *et al*. [16] solved the combinatorial problem of the data association by employing long short-term memory (LSTM) networks. However, their tracker exploits no appearance information and thus does not provide competitive results. Kim *et al*. [39] introduced the bilinear LSTM, which constrains its memory and new input to have a linear relationship, to learn a sequential appearance model. Sadeghian *et al*. [17] trained Siamese CNNs to construct appearance, motion, and interaction features and employed LSTMs to encode long-term temporal dependencies from these features. Our tracker is closely related to [17] in that we also adopt both CNN and RNN to consider spatiotemporal information. However, ours is different from their approach in that we develop an attention technique for RNN to alleviate the negative effects of occlusions.

Single object trackers are employed for online MOT in [40]–[42]. Kim and Kim [40] developed a cooperative tracking algorithm, which uses an object detector and a single object tracker jointly. Their algorithm traces each detected object using the single object tracker. Similarly, Chu *et al*. [41] employed a single object tracker for each object and updated the trackers to adapt to target appearance variations. Zhu *et al*. [42] separated target objects into tracked ones and lost ones and used a single object tracker and a data association scheme for the tracked targets and the lost targets. Using single object tracker, these algorithms [40]–[42] can find target objects that object detectors fails to locate, but they require heavy computational loads due to the use of many single object trackers.

Recently, the probability hypothesis density (PHD) filter [49] has drawn much attention in the online MOT problem. Fu *et al*. [19] proposed an adaptive gating scheme and an online group-structured dictionary learning to improve the PHD filter. Fu *et al*. [43] utilized different types of human detector (full-body and body-part) for the PHD filter.

Most of the tracking-by-detection methods focus on the problem of the data association. However, their performances are strongly affected by the quality of detection results. Thus, Zhou *et al*. [44], [45] proposed the deep neural networks to revise misaligned detection results and showed that their alignment methods are useful in the tracking-by-detection framework.

B. FEW-SHOT LEARNING

The objective of few-shot learning is to design a classifier using limited training data, *i.e*. only a few examples (in the extreme case, one example) per class [20], [21]. Many few-shot techniques are based on the nearest neighbor classification, which is a non-parametric model and does not need training. However, the performance of the nearest neighbor classification depends on the distance. Therefore, many algorithms transform an input space into a feature space (or embedding space), in which distances can be computed effectively. Goldberger *et al*. [50] proposed the neighborhood component analysis (NCA) to learn the linear

transform to maximize the performance of the k nearest neighbor classification. Salakhutdinov and Hinton [51] extended NCA to nonlinear transforms using neural networks. Weinberger *et al*. [52] proposed the large margin nearest neighbor classifier, which constrains examples in different classes to be separated by a large margin, and Min *et al*. [53] adopted neural networks to improve this large margin classifier.

Vinyals *et al*. [22] proposed the matching network, which performs nearest neighbor classification based on cosine distances in embedding space, and mimics few-shot classification tasks during their training phase as well as the test phase. Snell *et al*. [23] proposed the prototypical network, which transforms data points in a class so that they cluster tightly around a prototype in an embedding space. The prototype is the mean of examples in the class. And the classification is performed by comparing the distances of a query point to the prototypes. Sung *et al*. [24] proposed the relation network to learn distances for nearest neighbor classification, instead of using conventional metrics such as cosine distances. Li *et al*. [25] proposed the local image-to-class descriptor for few-shot classification based on pixel-wise cosine similarities between a query image and k nearest neighbor example images in the class. These few-shot learning algorithms [22]–[25] are related to the proposed method in that they learn an embedding space, where a class is represented faithfully with a few examples. However, their algorithms may not proper to video objects, since they are not designed to exploit the characteristics of sequential data. In contrast, the proposed method trains RNN for this purpose and effectively handles video objects in tracking applications.

C. OPEN-SET CLASSIFICATION

Open-set classification assumes that a training set cannot contain all possible classes in a test set. In this scenario, classes are categorized into either known or unknown: known classes are included in both training and test sets, wheres unknown classes are only in the test set. Therefore, open-set classification extends the standard classification with the requirement of recognizing test data in unknown classes. For this purpose, Phillips *et al*. [26] proposed the operating threshold to discriminate known classes from unknown ones. Scheirer *et al*. [27] formalized open-set classification by introducing the concept of open space risk, and then integrated it into the empirical risk minimization. In [54], Scheirer *et al*. presented the compact abating probability model to extend [27] to a multiclass setting. Bendale and Boult [55] proposed a method to adopt deep learning for open-set classification. They introduced the OpenMax layer to predict the unknown class probability. In [56], Oza and Patel trained an open-set classifier based on class conditioned auto-encoders. Recently, Liu *et al*. [57] defined the open long-tailed recognition problem that integrates difficulties in a real world scenario such as imbalanced examples, few examples, and open classes.

III. PROPOSED ALGORITHM

This section proposes a novel online MOT algorithm based on open-set few-shot learning. First, we introduce the notion of open-set few-shot classification and formulate MOT in the framework. Second, we propose an embedding scheme to transform an image space into an embedding space, where the nearest neighbor classification is performed. Third, we develop an online multiple object tracker based on the trained embedding space.

Online MOT can be regarded as few-shot classification [20], [21]. Queries and classes correspond to detection results in a current frame t and identified (tracked) objects in the previous frame $t-1$, respectively. As queries are classified using only a few examples in few-shot classification, detection results are assigned object labels by a data association scheme in online MOT. From this perspective, an effective few-shot learning technique can be used for the data association, thereby improving the tracking performance. However, MOT differs from the conventional few-shot classification in that MOT should handle unknown classes, *i.e*. detection results, which are unmatched with previously identified objects. Thus, MOT is also an open-set classification problem [26], [27], since it should recognize queries in unknown classes as well. Therefore, we introduce the notion of open-set few-shot classification to formulate online MOT, which has the properties of both open-set classification and few-shot classification.

Let us consider the case that there are K classes in a training dataset. The training dataset, $\{(x_1, y_1), \cdots, (x_N, y_N)\}$, contains labeled examples, where x_i is the ith example and $y_i \in \{1, \cdots, K\}$ is its class label. Then, the kth class $\mathcal{C}_k$ is given by the set of examples with label k,

$$\mathcal{C}_k = \{x_i : y_i = k\}_{i=1}^{N}. \tag{1}$$

There are only a few examples in each class (few-shot classification). Let q be a query and y be its class label. It is possible that q does not belong to any class $\mathcal{C}_k$, $1 \leq k \leq K$ (open-set classification). In such a case, it should be declared to be in the unknown class. Thus, in the open-set few-shot classification, q should be classified into one of the previously identified classes $\mathcal{C}_1, \cdots, \mathcal{C}_K$ or the unknown class.

In MOT, we consider two cases of the unknown class: newly appearing objects or detection errors. Note that it is important to distinguish between the two cases. When a detection result corresponds to a new object at a current frame, a tracker should create a new class for the object and construct its appearance model to estimate its states in subsequent frames. On the other hand, when a detection result is wrong, a tracker should discard it to prevent tracking errors in future frames. Hence, we divide the unknown class into two subclasses C_0 and C_{-1}, representing the new object class and the detection error class, respectively. Let us consider a tracking scenario in which there are K objects in the previous frame $t-1$ and N detection results in the current frame t. By introducing the unknown classes C_0 and C_{-1}, we formulate this scenario as the open-set few-shot classification,

where the tracker classifies each of the N detection results into one of the following $K+2$ classes:

- $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K$: previously identified object classes
- $\mathcal{C}_0$: new object class
- $\mathcal{C}_{-1}$: detection error class

A. EMBEDDING SPACE LEARNING

As pointed out in few-shot learning studies [22]–[25], a well-designed embedding space enables a non-parametric method based on a simple metric to model a class with only a few examples. Also, an embedding space should provide reliable metrics to discriminate unknown classes from known classes. Therefore, learning an effective embedding space $\phi(\cdot)$ is essential for accurate open-set few-shot classification. To this end, we develop the embedding network, called VOFNet, which transforms query q and known classes $\mathcal{C}_1, \cdots, \mathcal{C}_K$ into the embedding space, *i.e.* $\phi(q)$ and $\phi(\mathcal{C}_1), \cdots, \phi(\mathcal{C}_K)$, respectively. Figure 2 illustrates VOFNet containing a CNN and an RNN. We refer to them as VOF-CNN and VOF-RNN, respectively.

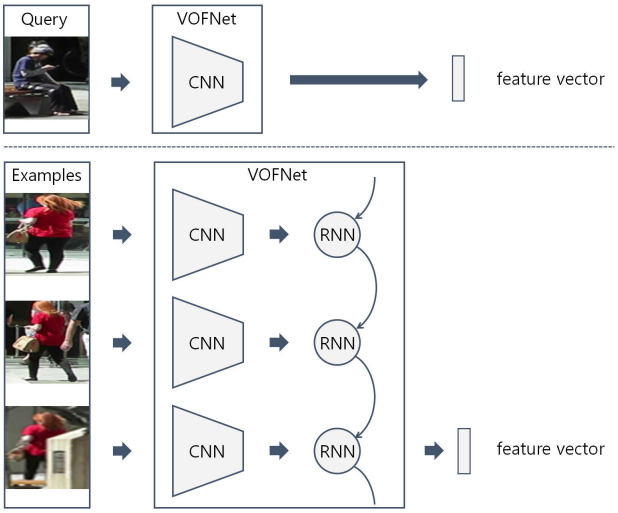

FIGURE 2. Illustration of VOFNet, composed of a CNN and an RNN. Given an image patch, the CNN extracts the feature vector. Then, the RNN takes a series of feature vectors from the CNN and integrates them to yield the representative feature vector for the open-set few-shot classification.

VOF-CNN transforms queries and examples of identified objects into the embedding space. More specifically, it takes an RGB image patch of size 96×192 and produces a 512-dimensional feature vector. VOF-CNN consists of a backbone network and a normalization layer. For the backbone, we employ the EfficientNet-B0 model [58] pretrained with ImageNet [59] and replace its classification layer with a fully-connected layer to yield a 512-dimensional output. The normalization layer constrains an embedding space to be a unit sphere. We set the height of patch to be larger than the width to consider pedestrians, which is the important object in MOT.

When a known class $\mathcal{C}_k$, $1 \leq k \leq K$, has multiple examples, we aggregate them in the embedding space to obtain

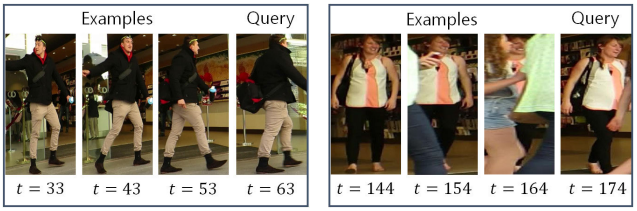

FIGURE 3. Examples and queries. In general, as in (a), the latest example is the most similar to a query. However, in some cases, such as the occlusion in (b), the latest example fails to represent the class properly.

the representative feature $\phi(\mathcal{C}_k)$. A straightforward approach is to average features of examples, as in the prototypical networks [23], but this cannot exploit the characteristics of a video object effectively. For instance, in Figure 3(a), the query is the most similar to the example in the latest frame. In general, the recent example is closer to a query in the embedding space than the mean vector of all examples. However, it is not robust either to consider the recent example only. In Figure 3(b), an object is occluded in the latest frame. Thus, it may cause misclassification if the class is represented by the recent example only.

For effective aggregation of examples in a class, we use an RNN [60] that is capable of encoding temporal histories of video objects. As illustrated in Figure 2, the proposed VOF-RNN takes a series of feature vectors from VOF-CNN and integrates them into the representative feature vector. Suppose that there are N examples, $\{x_i : y_i = k\}_{i=1}^N$, in the kth object $\mathcal{C}_k$. Then, we extract feature vectors of the examples $\{\phi(x_i)\}_{i=1}^N$ by applying VOF-CNN to the examples. Given the features $\{\phi(x_i)\}_{i=1}^N$, VOF-RNN updates the states $\{h_i\}_{i=1}^N$ and produces the outputs $\{o_i\}_{i=1}^N$ recursively:

$$h_i = (1-\alpha_i)h_{i-1} + \alpha_i f(W_{ss}h_{i-1} + W_{sz}\phi(x_i)) \quad (2)$$

$$o_i = g(W_{so}h_i) \quad (3)$$

Here, W_{ss}, W_{sz}, W_{so} are trainable weight matrices of size 512×512. f and g are the ReLU activation function and the l2-normalization function, respectively. Different from the standard RNN, Eq. (2) includes an attention weight α_u to encourage a conservative RNN state update. Given the feature vector of the current frame example, we obtain this attention weight using two fully-connected layers and a sigmoid activation layer. Note that MOT sequences often consider objects in crowded scene whose appearance are distorted due to occlusion. And this distorted example may cause inadequate RNN update that decreases classification accuracy. In this case, the attention weight can alleviate the negative effect of distorted example.

Then, given a sequence of examples $\{x_i : y_i = k\}_{i=1}^N$ for the kth object, we set the representative feature $\phi(\mathcal{C}_k)$ as,

$$\phi(\mathcal{C}_k) = o_N. \quad (4)$$

Specifically, the VOF-RNN updates its hidden state h_i and output vector o_i via (2) and (3), respectively, until $i = N$,

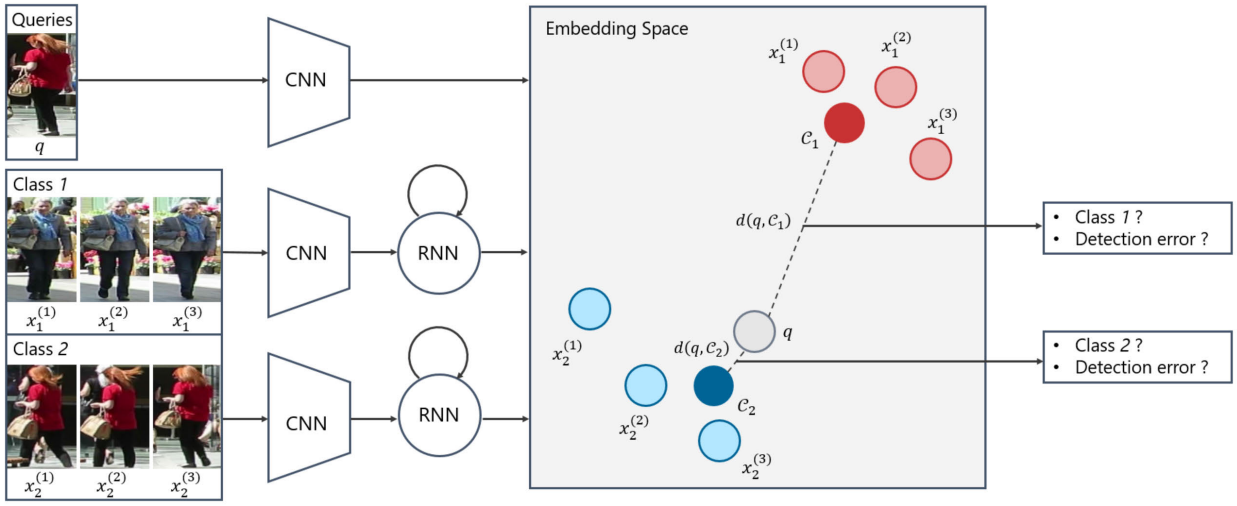

FIGURE 4. An example of the proposed embedding space learning.

where N is the number of examples. In addition to the capability of encoding temporal histories, VOF-RNN has another advantage that it requires only the previous state h_{i-1} and the input $\phi(x_i)$ to compute the state h_i and the output o_i at the current state. Therefore, it reduces the memory resource requirement during the tracking, by storing only the recent state of a class, instead of its all examples.

Figure 4 illustrates the proposed training strategy for VOFNet. Let us consider an embedding space, where feature vectors $\phi(q)$ and $\phi(\mathcal{C}_k)$ represent a query q and the kth object class $\mathcal{C}_k$, respectively. The distance $d(q, \mathcal{C}_k)$ between q and $\mathcal{C}_k$ is defined as

$$d(q, \mathcal{C}_k) = \|\phi(q) - \phi(\mathcal{C}_k)\|_2. \tag{5}$$

where $\|\cdot\|_2$ is the l_2-norm of a feature vector. For learning the embedding space, we train two binary classifiers for each known class k, which determine whether the query belongs to 1) kth class or not (known class classifier) and 2) detection error or not (detection error classifier), based on the distance $d(q, \mathcal{C}_k)$. The smaller the distance $d(q, \mathcal{C}_k)$, the more likely that the query belongs to the kth class. In contrast, the larger the distance $d(q, \mathcal{C}_k)$, the query can be regarded as detection error.

Specifically, given the query q, we estimate the probabilities $\hat{y}_{k,1}$ and $\hat{y}_{k,2}$ for the known class classifier and the detection error classifiers, which are given by

$$\hat{y}_{k,1} = \sigma(m_l - d(q, \mathcal{C}_k)), \quad \hat{y}_{k,2} = 1 - \sigma(m_h - d(q, \mathcal{C}_k)) \tag{6}$$

where $\sigma(\cdot)$ is a sigmoid function to yield probability. To compute the probabilities, we use two thresholds m_l and m_h. The threshold m_l is the maximum distance to identify the same object, whereas the threshold m_h is used to recognize the detection error. When the distance $d(q, \mathcal{C}_k)$ is smaller than m_l, the query q has the high probability to belong to the class k according to the sigmoid function. In contrast, when the distance $d(q, \mathcal{C}_k)$ is lager than m_h, the probability for the detection error $\hat{y}_{k,2}$ has a large value. In this work, the thresholds m_l and m_h in (6) are fixed to 0.5 and 1.0, respectively. We then compute the binary cross-entropy loss for both classifiers in the kth class as

$$\begin{aligned} L_k = &-y_{k,1} \log \hat{y}_{k,1} - (1 - y_{k,1}) \log(1 - \hat{y}_{k,1}) \\ &- y_{k,2} \log \hat{y}_{k,2} - (1 - y_{k,2}) \log(1 - \hat{y}_{k,2}) \end{aligned} \tag{7}$$

where the ground-truth label $y_{k,1}$ is 1 if the query q belongs the kth class, and 0 otherwise. Similarly, the ground-truth label $y_{k,2}$ is 1 if the query q is detection error, and 0 otherwise. Finally, we minimize total losses for all known classes, *i.e.* $\sum_{k=1}^{K} L_k$, to train VOFNet via the stochastic gradient decent.

B. ONLINE DATA ASSOCIATION

Let $\mathcal{D}^{(t)} = \{q_1^{(t)}, \cdots, q_N^{(t)}\}$ be the set of detections (or detected bounding boxes) and $\mathcal{A}^{(t)} = \{\mathcal{C}_1^{(t)}, \mathcal{C}_2^{(t)}, \cdots, \mathcal{C}_{K^{(t)}}^{(t)}\}$ be the active set of identified object classes at frame t. In the open-set few-shot formulation, the objective of data association is to predict class labels $\{y_1^{(t)}, \cdots, y_N^{(t)}\}$ of the detections in $\mathcal{D}^{(t)}$, where $y_i^{(t)} \in \{-1, 0, 1, \cdots, K^{(t)}\}$, based on the learned embedding space in Section III-A. Note that the labels $y_i^{(t)} = -1$ and $y_i^{(t)} = 0$ denote the detection error class $\mathcal{C}_{-1}$ and the new object class $\mathcal{C}_0$, respectively.

To initialize the active set $\mathcal{A}^{(1)}$ at the first frame, we simply regard each detection as the identified object, *i.e.* $\mathcal{C}_k^{(1)} = \{q_k^{(1)}\}$. From the second frame, given the active set $\mathcal{A}^{(t-1)}$, we perform data association to assign labels to each detection in $\mathcal{D}^{(t)}$. First, VOFNet transforms each detection $q_i^{(t)}$ and each identified object $\mathcal{C}_k^{(t-1)}$ into the embedding space to obtain features $\phi(q_i^{(t)})$ and $\phi(\mathcal{C}_k^{(t-1)})$, and distances between the detections and the identified objects are computed via Eq. (5). Then, we determine whether each detection $q_i^{(t)}$ is the

detection error or not by averaging distances from the object classes in the active set $\mathcal{A}^{(t-1)}$:

$$q_i^{(t)} \in \mathcal{C}_{-1}^{(t)} \quad \text{if } \frac{1}{K}\sum_{k=1}^{K} d(q_i^{(t)}, \mathcal{C}_k^{(t-1)}) > m_h \tag{8}$$

where m_h is the threshold that is used in VOFNet learning in Section III-A. We discard detections in $\mathcal{C}_{-1}^{(t)}$ from $\mathcal{D}^{(t)}$.

Next, we collect candidate detections for data association from $\mathcal{D}^{(t)}$ by exploiting distances between the detections and the identified objects. When a detection $q_i^{(t)}$ in $\mathcal{D}^{(t)}$ has the most similar feature to the object $\mathcal{C}_k^{(t-1)}$ and the distance $d(q_i^{(t)}, \mathcal{C}_k^{(t-1)})$ is sufficiently small, it is likely that $q_i^{(t)}$ belongs to $\mathcal{C}_k^{(t-1)}$. Therefore, we compose the set of candidate detections $\tilde{\mathcal{D}}^{(t)}$ as follow,

$$q_i^{(t)} \in \tilde{\mathcal{D}}^{(t)} \quad \text{if } \min_{\mathcal{C}_k^{(t-1)} \in \mathcal{A}^{(t-1)}} d(q_i^{(t)}, \mathcal{C}_k^{(t-1)}) < m_l. \tag{9}$$

Otherwise, when a detection does not satisfy the candidate condition in Eq. (9), that detection can be regarded as newly appearing object at frame t. This is because detection errors are excluded in the detection set $\mathcal{D}^{(t)}$ via Eq. (8). We then define the new object class as $\mathcal{C}_0^{(t)} = \mathcal{D}^{(t)} - \tilde{\mathcal{D}}^{(t)}$. Notice that each detection in the new object class $\mathcal{C}_0^{(t)}$ generates new identified object class $\mathcal{C}_{\tilde{K}^{(t)}}$, where $\tilde{K} > K^{(t-1)}$, in the active set $\mathcal{A}^{(t)}$ at frame t.

We associate detections in $\tilde{\mathcal{D}}^{(t)}$ with the identified objects in the active set $\mathcal{A}^{(t-1)}$. We formulate data association as a bipartite matching problem to enforce one-to-one matching constraint. We construct a bipartite graph $\mathcal{G} = (U, V, E)$, where $U = \{u\}$ and $V = \{v\}$ are two independent node sets, and $E = \{e_{u,v}\}$ is an edge set. The edge $e_{u,v}$ connects nodes u and v with a nonnegative cost $c_{u,v}$. In this work, detection candidates in $\tilde{\mathcal{D}}^{(t)}$ and the identified objects in $\mathcal{A}^{(t-1)}$ become the node sets U and V, respectively. The nonnegative cost $c_{u,v}$ is assigned the distance in (5).

Given the bipartite graph $\mathcal{G}$, we determine the matching between the node sets U and V, which minimizes the sum of nonnegative costs, subject to the one-to-one constraint. To this end, we formulate the objective function for the bipartite matching problem, which is given by

$$\begin{aligned} \underset{\mu_{u,v}}{\text{minimize}} \quad & \sum_{u\in U}\sum_{v\in V} \mu_{u,v} c_{u,v} \\ \text{subject to} \quad & \sum_{u\in U} \mu_{u,v} \le 1 \ \text{ for each } \ v \in V \\ & \sum_{v\in V} \mu_{u,v} \le 1 \ \text{ for each } \ u \in U \\ & \mu_{u,v} \in \{0, 1\}. \end{aligned} \tag{10}$$

where $\mu_{u,v}$ is a matching variable that equals 1 if u is matched to v, and 0 otherwise. We employ the Hungarian algorithm [61] to minimize this objective function. Then, we add each detection candidate in $\tilde{\mathcal{D}}^{(t)}$ to the matched object class to update the active set from $\mathcal{A}^{(t-1)}$ to $\mathcal{A}^{(t)}$. For instance, when a detection $q_i^{(t)}$ is matched with the object class $\mathcal{C}_k^{(t-1)}$, the object class for frame t is updated by $\mathcal{C}_k^{(t)} = \mathcal{C}_k^{(t-1)} \cup q_i^{(t)}$.

Figure 5 shows examples when the number of detection candidates and identified objects are different. Specifically, Figure 5(a) illustrates an example of a tracking scenario, when the number of object classes is less than that of detection candidates. On the other hand, a tracking scenario in Figure 5(b) includes more object classes than detection candidates. If one-to-one matching constraint is enforced as in (10), an object class $\mathcal{C}_3^{(t-1)}$ in Figure 5(a) and a detection candidate q_3 in Figure 5(b) are unmatched. Therefore, these examples bring up the additional issues how to handle unmatched detections and unmatched object classes. First, we regard unmatched detection candidates as newly appearing objects and include them to the new object class $\mathcal{C}_0^{(t)}$.

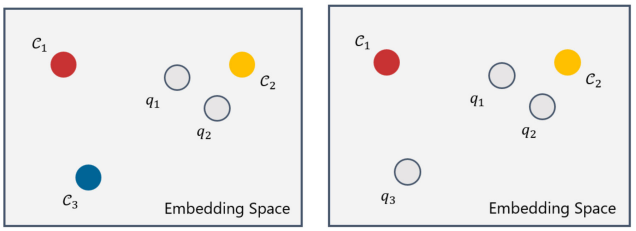

FIGURE 5. An example of tracking scenario. $\mathcal{C}_1$ and $\mathcal{C}_2$ denote target objects, while q_1, q_2 and q_3 are detection results.

For the unmatched objects, there are two possible cases: 1) failures of the object detector to find objects and 2) permanent object disappearance. In the first case, undetected objects may reappear in future frames. Therefore, the tracker should maintain those object classes to resume tracking, when the object detector re-identify them. In the second case, the tracker should terminate tracking of permanently disappeared objects. To consider both temporary and permanent disappearance, we record the number of successive unmatched frames τ for each disappeared object. Then, we regard that object permanently disappears when τ is larger than a threshold θ frames and exclude the permanently disappeared object from the active set $\mathcal{A}^{(t)}$. In this work, we experimentally set θ to 30.

Figure 6 illustrates the proposed MOT process. In this example, the active set $\mathcal{A}^{(t-1)}$ contains three objects that exist in frame $t-1$. First, VOFNet embeds the objects and the detection results in $\mathcal{D}^{(t)}$ into the embedding space. We then perform the nearest neighbor classification, and declare the detection $q_3^{(t)}$ as a new object. Using the other detections and the objects in the active set $\mathcal{A}^{(t-1)}$, we construct the bipartite graph $\mathcal{G}$ and obtain the optimal matching using the Hungarian algorithm. As a result, the bounding boxes of the objects $\mathcal{C}_1^{(t-1)}$ and $\mathcal{C}_2^{(t-1)}$ are determined to be $q_1^{(t)}$ and $q_2^{(t)}$, respectively. Next, we verify that $q_3^{(t)}$ is a new object and include it in the active set $\mathcal{A}^{(t)}$ at frame t. Finally, we check disappearing objects. In this example, we cannot find the bounding box of $\mathcal{C}_3^{(t)}$ for a long duration. Therefore, $\mathcal{C}_3^{(t)}$ is

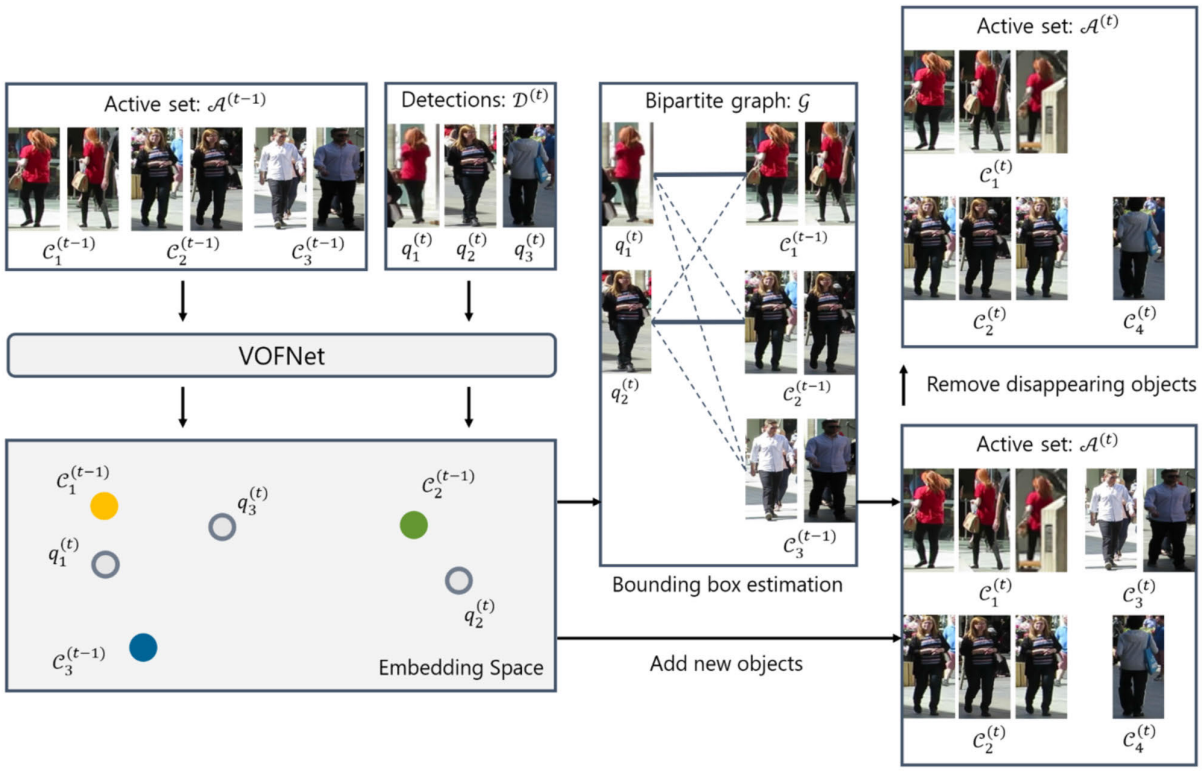

FIGURE 6. An example of the proposed MOT process.

not added to $\mathcal{A}^{(t)}$. Algorithm 1 summarizes the online data association method for MOT.

IV. EXPERIMENTS

In this section, we evaluate the proposed tracker on MOT benchmark datasets [28]. We briefly introduce the datasets and provide implementation details for our experiments. Then, we investigate the effectiveness of the proposed tracker by performing ablation studies. Also, we show qualitative MOT results. Finally, we compare the performance of proposed tracker with recent state-of-the-arts for quantitative analysis.

A. DATASET

MOT17 dataset [28] consist of 14 video sequences, which are divided into 7 training and 7 test sequences. For each sequence, the dataset provides bounding boxes obtained from three detectors: DPM [62], FRCNN [63], and SDP [64] detectors. In MOT17, ground-truth annotations are available for the training sequences only, to avoid fitting of the methods to test sequences.

The benchmarks adopt various evaluation metrics to quantify the MOT performance, which are defined in [9], [65]. Multiple object tracking accuracy (MOTA) is a metric to consider several failure cases, which is composed of the number of false positives (FP), false negatives (FN), and identity switches (IDS), where

- FP: an estimated state does not include objects
- FN: a tracker misses objects
- IDS: an object is assigned a different class label from the label in the previous frame.

More precisely, MOTA is defined as

$$MOTA = 100 \times \left(1 - \frac{\#FP + \#FN + \#IDS}{\#GT}\right) \quad (11)$$

where GT denotes ground-truth states and # denotes the number. Multiple object tracking precision (MOTP) measures the average overlap ratio between estimated states and corresponding annotations. Identification F_1 (IDF1) is the ratio of correctly estimated states over the average number of ground-truth and estimated states. Mostly tracked targets (MT) is the number of objects whose trajectories are estimated accurately by tracking results more than 80% of frames. Similarly, mostly lost targets (ML) is defined to the number of objects whose trajectories are covered by tracking results less than 20% of frames. For MT and ML, the benchmarks consider that a target state is accurately estimated when the overlap ratio between it and predicted state is greater than 0.5. Also, the runtime speed (Hz) is included as another benchmark evaluation metric.

B. IMPLEMENTATION DETAILS

We implement the proposed tracker in the Python language using the TensorFlow 2.0 library. Experiments are performed

Algorithm 1 Online Data Association

Require: Detection results $\mathcal{D}^{(1)}, \cdots, \mathcal{D}^{(T)}$
Ensure: Active sets $\mathcal{A}^{(1)}, \cdots, \mathcal{A}^{(T)}$
Initialize $\mathcal{A}^{(1)}$
for $t = 2$ **to** T **do**
  $\mathcal{A}^{(t)} \leftarrow \phi$
  Transform $\mathcal{D}^{(t)}$ into the embedding space
  Divide $\mathcal{D}^{(t)}$ into $\tilde{\mathcal{D}}^{(t)}, \mathcal{C}_l^{(\sqcup)}, \mathcal{C}_{-\infty}^{(\sqcup)}$
  Add new objects in $\mathcal{C}_l^{(\sqcup)}$ to $\mathcal{A}^{(t)}$
  Associate $\tilde{\mathcal{D}}^{(t)}$ with $\mathcal{A}^{(t-1)}$
  if Unmatched detection candidate case **then**
    Add this detection candidate to $\mathcal{A}^{(t)}$
  else if Unmatched object case **then**
    if $\tau < \theta$ **then**
      $\tau \leftarrow \tau + 1$
      Add this object to $\mathcal{A}^{(t)}$
    else
      Terminate tracking of this object
    end if
  else
    Update matched objects
    Add matched objects to $\mathcal{A}^{(t)}$
  end if
end for

on a personal computer with an Intel I7-7700K CPU and a NVIDIA 2080 Ti GPU.

Training is done in two steps: First, we train VOF-CNN to find the effective embedding space for the open-set few-shot classification. Second, we train VOF-RNN to learn the encoding scheme for sequential data. For the VOF-CNN training, we randomly sample two successive frames from the MOT17 training set. From the previous frame, we construct object classes, $\mathcal{C}_1, \ldots, \mathcal{C}_K$, using detection boxes, whose intersection over union (IOU) scores with corresponding annotations are larger than 0.5. Notice that each object class contains only one example in the VOF-CNN training. In other words, VOF-CNN is trained to consider a one-shot scenario. In the current frame t, we collect detected bounding boxes and regard each detection box as a query. Then, we assign two types of labels for each query: one indicates whether the query belongs to the object or not, and the other denotes whether the query is a detection error or not. Thus, when K objects exist in the previous frame, each query has total $2K$ labels.

In the VOF-RNN learning, we randomly choose a frame from MOT17 training sequences and construct object classes, $\mathcal{C}_1, \ldots, \mathcal{C}_K$, from 30 consecutive previous frames. Specifically, for each object class, we randomly extract 10 examples from the 30 previous frames. Therefore, VOF-RNN is trained to construct a representative vectors with only a few example. As done in VOF-CNN, we extract queries from the current frame and assign class labels for training VOF-RNN. Notice that the trained VOF-CNN is used to extract feature of each query and each example for the VOF-RNN learning.

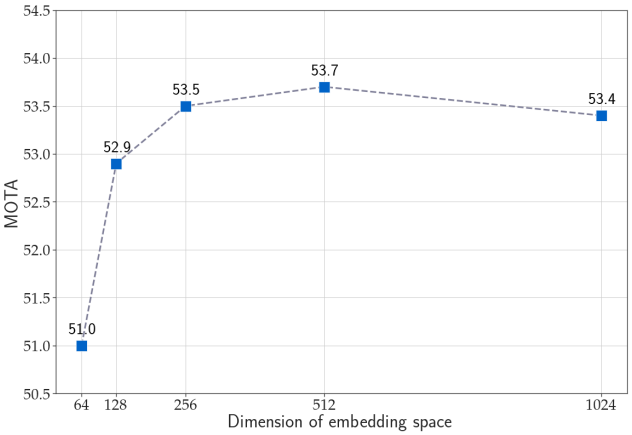

FIGURE 7. MOTA scores according to dimensions of the embedding space.

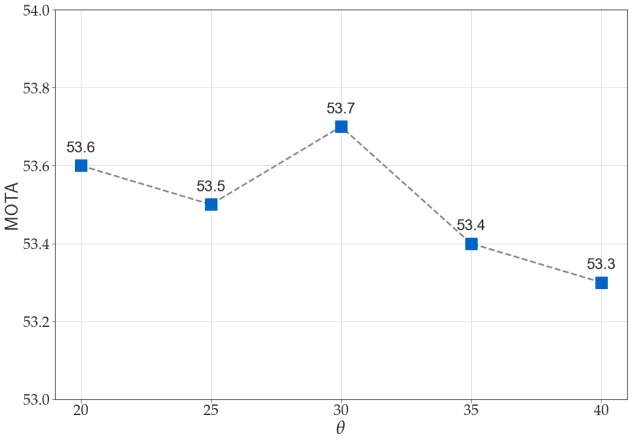

FIGURE 8. MOTA scores according to the threshold θ.

TABLE 2. MOTA scores according to embedding spaces and representative vectors of object classes on the MOT17 training sequences.

Embedding Space	Representative Vector	MOTA
Baseline	Latest Example	48.3
	Average	47.9
	Moving average	48.5
	VOF-RNN w/o attention	48.5
	VOF-RNN	48.6
VOF-CNN	Latest Example	52.2
	Average	51.6
	Moving average	52.7
	VOF-RNN w/o attention	53.3
	VOF-RNN	53.7

For training both VOF-CNN and VOF-RNN, we perform the data augmentation by applying the horizontal flipping to training data with probability 0.5. Also, we employ the Adam optimizer [66] with a learning rate of 0.0001. The training is iterated for 40, 000 episodes. We decrease the learning rate by a factor of 0.1 at the 20, 000th episode.

TABLE 3. Performance comparison of the proposed tracker with the state-of-the art trackers on the MOT17 test sequences. The best results are boldfaced.

Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Hz ↑
A. Offline (batch) trackers								
TC [67]	**51.9**	**58.1**	544	836	36,164	**232,783**	2,288	0.7
HAF [35]	51.8	54.7	**551**	893	33,212	236,772	1,834	0.7
NOTA [68]	51.3	54.5	403	833	**20,148**	252,531	2,285	**17.8**
jCC [37]	51.2	54.5	493	872	25,937	247,822	1,802	1.8
STRN [69]	50.9	56.0	446	**797**	25,295	249,365	2,397	13.8
MHT-DAM [14]	50.7	47.2	491	869	22,875	252,889	2,314	0.9
TLMHT [36]	50.6	56.5	415	1,022	22,213	255,030	**1,407**	2.6
MHT-bLSTM [39]	47.5	51.9	429	981	25,981	268,042	2,069	1.9
SAS [70]	44.2	57.2	379	1,044	29,473	283,611	1,529	4.8
B. Online trackers								
FAMNet [71]	**52.0**	48.7	450	787	14,138	253,616	2,689	0.0
OneShotDa [18]	51.4	**54.0**	**500**	878	29,051	243,202	2,593	1.8
MOTDT [38]	50.9	52.7	413	841	24,069	250,768	2,474	18.3
MTDF [43]	49.6	45.2	444	**779**	37,124	**241,768**	5,567	1.2
DASOT [1]	49.5	51.8	481	814	33,640	247,370	4,142	9.1
OTCD [72]	48.6	47.9	382	970	18,499	268,204	3,502	15.5
DMAN [42]	48.2	55.7	454	901	26,218	263,608	**2,194**	0.3
AM-ADM [73]	48.1	52.1	316	934	25,061	265,495	2,214	5.7
PHD-GSDL [19]	48.0	49.6	402	838	23,199	265,954	3.998	6.7
MASS [74]	46.9	46.0	399	856	25,733	269,116	4.478	17.1
LM-NN [75]	45.1	43.2	348	1,088	**10,834**	296,451	2.463	0.9
FPSN [73]	44.9	48.4	388	844	33,757	269,952	7,136	10.1
Proposed	49.2	51.1	433	869	23,108	260,562	3,168	**28.9**

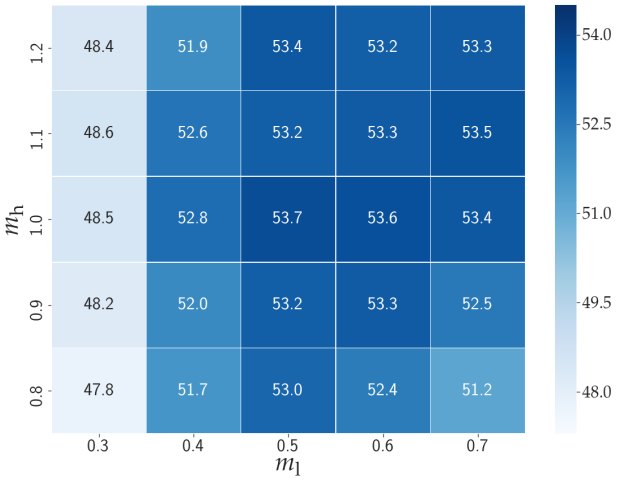

FIGURE 9. MOTA scores according to the thresholds m_l and m_h.

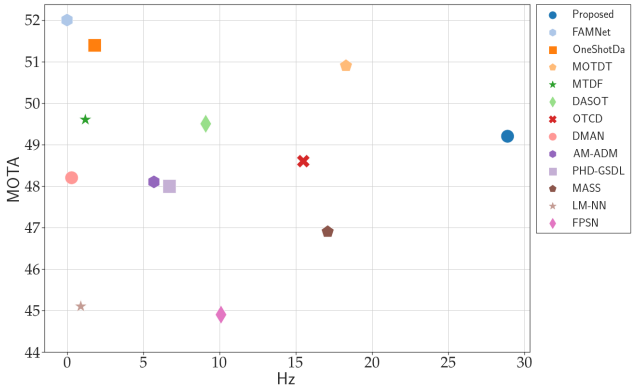

FIGURE 10. MOTA scores versus speed (Hz) on the MOT17 test sequences.

C. PERFORMANCE ANALYSIS

We analyze the impacts of various components in the proposed algorithm. For this purpose, we use the MOT17 training sequences, since the annotations for the test sequences are not released. We perform the cross-fold validation on the seven training sequences. Specifically, we use six sequences for the training and one sequence for the evaluation and repeat this process for each sequence.

Table 2 provides MOTA scores of the proposed tracker according to embedding spaces and feature extraction methods for encoding representative vectors of object classes. Specifically, "Baseline" denotes the EfficientNet-B0 model pretrained with ImageNet, while "VOF-CNN" is the trained embedding space via section III-A. Also, as the methods for encoding representative vectors, "Latest Example" utilizes the latest example in the object class, "Average" computes the mean vector of all examples in the object class as done in [23], and "Moving average" computes the exponential moving mean vector with the decay rate 0.1. For "VOF-RNN w/o attention", we fix the attention weight α_u in Eq. (2) to 1.0. Therefore, "VOF-RNN w/o attention" becomes the standard RNN that does not include the attention weight.

From Table 2, we can make some observations: First, learning a useful embedding space is essential to yield promising tracking performance. Notably, "VOF-CNN" provides better results than "Baseline", for all cases of the representative vectors. Second, we observe that it is important to exploit

FIGURE 11. Tracking results of the proposed tracker on the MOT17 training sequences: (a) "MOT17-02," (b) "MOT17-04," (c) "MOT17-05," (d) "MOT17-09," (e) "MOT17-10," (f) "MOT17-11," and (g) "MOT17-13" sequences.

FIGURE 12. Tracking results of the proposed tracker on the MOT17 test sequences: (a) "MOT17-01," (b) "MOT17-03," (c) "MOT17-06," (d) "MOT17-07," (e) "MOT17-08," (f) "MOT17-12," and (g) "MOT17-14" sequences.

the characteristics of video objects. "Latest Example" and "Moving average" outperform "Averagee" since a object in a video tends to have similar appearances in adjacent frames. In this regards, "VOF-RNN" supports accurate tracking by learning an effective encoding scheme for video objects. So, it provides higher MOTA scores than simple encoding schemes. Moreover, note that "VOF-RNN" requires only the recent state vector to yield the object's representative feature. Therefore, it reduces memory and processing loads during tracking. Finally, the attention weight further improves the proposed tracker's performance by reducing the adverse effect of distorted examples.

Next, we compare the performance of the proposed tracker with different hyper-parameter settings. Figure 7 evaluates MOTA scores according to the dimension of the proposed embedding space. Specifically, we train five VOF-CNNs that produce feature vectors with size of 64, 128, 256, 512, and 1024, respectively. Thenm we train VOF-RNN for each VOF-CNN. In Figure 7, the best performance is achieved when we set the dimension of embedding space to 512. Therefore, we use this setting for the proposed tracker. Figure 8 investigates the impact of the threshold frame θ that determines disappearing objects from the active set. In Figure 8, we observe that the proposed tracker is not sensitive to the threshold θ in the range from 20 to 40.

In addition, we compare MOTA scores according to thresholds m_l and m_h. Specifically, we set the range of m_l from 0.3 to 0.7 and the range of m_h from 0.8 to 1.2. Figure 9 shows the results of these experiments. In Figure 9, too low threshold $m_l = 0.3$ decreases MOTA scores since it is too strict condition to assign an object class to detected bounding boxes. As a result, the number of false negatives increases. The proposed algorithm works reliably when using thresholds $m_l \in \{0.5, 0.6, 0.7\}$ and $m_h \in \{1.0, 1.1, 1.2\}$. In this wor, we use $m_l = 0.5$ and $m_h = 1.0$, which yield the best performance.

D. COMPARISON WITH CONVENTIONAL TRACKERS

Table 3 compares the proposed VOFNet tracker with recent state-of-the-art offline and oneline trackers on the MOT17 test sequences. As compared with the offline trackers, the proposed algorithm provides the competitive MOTA performance, event though the offline trackers require all frames to achieve multiple object tracking. Also, the proposed tracker achieves the almost real-time processing (28.9 Hz) and yields the best speed in Table 3. As compared with the online trackers, the proposed algorithm ranks 6th in terms of MOTA. Notice that the proposed tracker achieves the comparable performances to the conventional online trackers, even though the proposed one surpasses other algorithms for speed. To analyze the trade off between accuracy and speed, Figure 10 plots the MOTA score with respect to the Hz. In Figure 10, we observe that the proposed tracker shows the best trade-off between accuracy and speed.

E. QUALITATIVE MOT RESULTS

Figure 11 shows qualitative tracking results of the proposed algorithm on the MOT17 training sequences. In this test, DPM is used as the detector. Each sequence contains different difficulties: "MOT17-02" was recorded in cloudy weather. Thus, it is difficult to identify objects due to low contrast. "MOT17-04" has the highest density of objects, in which objects experience occlusion frequently. "MOT17-05" contains lots of motion blur caused by camera movements. "MOT17-09," "MOT17-10," "MOT17-11," and "MOT17-13' suffer from large variation in object scale and too small objects. Despite these difficulties, we see that the proposed VOFNet tracker yields promising tracking results. Finally, Figure 12 shows tracking results of the proposed algorithm on the MOT17 test sequences with SDP detector. Similar to training sequences, test sequences also include various difficulties such as occlusion, scale variation, and motion blur. We observe that the proposed algorithm tracks multiple targets accurately.

The proposed tracker does not use additional object detectors to refine object detection results. As a result, the proposed algorithm is weak to tracking target objects whose detection boxes are not provided by object detectors in some frames. Figure 13 shows failure examples of the proposed algorithm on a challenging scenario that the DPM object detector misses many objects. As in Figure 13(a), the proposed tracker fails to track a person with index 1 at the frame 313, since a detection box for the person is not provided.

FIGURE 13. Failure examples of the proposed tracker on the MOT17 test sequences: (a) "MOT17-01" and (b) "MOT17-14 sequences.

V. CONCLUSION

In this paper, we introduced the notion of open-set few-shot classification to formulate the online MOT problem. Then, we proposed a novel embedding network, named VOF-Net, to perform the open-set few-shot classification. VOFNet includes VOF-CNN and VOF-RNN. VOF-CNN finds a non-linear mapping from an image space into an embedding space, where the open-set few-shot classification is performed effectively. VOF-RNN learns an encoding scheme to construct the representation feature of sequential data in the embedding space. Finally, we developed the online tracker, based on VOFNet. Experimental results demonstrate

that, despite of the computational simplicity, the proposed VOFNet tracker yields comparable or better performance than the conventional state-of-the-art trackers in the MOT17 benchmark.

Nevertheless of its effectiveness, the proposed tracker has a limitation on inaccurate detection results as in Figure 13. This problem can be addressed by motion models such as Kalman filter or Particle filter, which estimate object positions, even when detectors fail to find target objects during tracking. Therefore, it remains future works to integrate open-set few-shot learning and motion modeling techniques.

REFERENCES

[1] Q. Chu, W. Ouyang, B. Liu, F. Zhu, and N. Yu, "DASOT: A unified framework integrating data association and single object tracking for online multi-object tracking," in *Proc. AAAI*, Feb. 2020, pp. 10672–10679.

[2] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Auto. Robots*, vol. 26, nos. 2–3, pp. 123–139, Apr. 2009.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man Cybern., C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[4] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 367–386, Mar. 2015.

[5] Q. Leng, M. Ye, and Q. Tian, "A survey of open-world person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1092–1108, Apr. 2020.

[6] A. Sasithradevi and S. Mohamed Mansoor Roomi, "A new pyramidal opponent color-shape model based video shot boundary detection," *J. Vis. Commun. Image Represent.*, vol. 67, Feb. 2020, Art. no. 102754.

[7] S. H. Abdulhussain, S. A. R. Al-Haddad, M. I. Saripan, B. M. Mahmmod, and A. Hussien, "Fast temporal video segmentation based on krawtchouk-tchebichef moments," *IEEE Access*, vol. 8, pp. 72347–72359, 2020.

[8] S. H. Abdulhussain, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, T. Baker, W. N. Flayyih, and W. A. Jassim, "A fast feature extraction algorithm for image and video processing," in *Proc. IJCNN*, 2019, pp. 1–8.

[9] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008.

[10] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[11] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[12] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[13] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2054–2068, Oct. 2016.

[14] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4696–4704.

[15] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018.

[16] G. P. Mauroy and E. W. Kamen, "Multiple target tracking using recurrent neural networks," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Feb. 2017, pp. 4225–4232.

[17] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 300–311.

[18] K. Yoon, J. Gwak, Y.-M. Song, Y.-C. Yoon, and M.-G. Jeon, "OneShotDA: Online multi-object tracker with One-Shot-Learning-Based data association," *IEEE Access*, vol. 8, pp. 38060–38072, 2020.

[19] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi, "Particle PHD filter based multiple human tracking using online group-structured dictionary learning," *IEEE Access*, vol. 6, pp. 14764–14778, 2018.

[20] M. G. Miller, N. E. Matsakis, and P. A. Viola, "Learning from one example through shared densities on transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2000, pp. 464–471.

[21] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[22] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, Dec. 2016, pp. 3630–3638.

[23] J. Wang and Y. Zhai, "Prototypical siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 4077–4087.

[24] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[25] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7253–7260.

[26] P. J. Phillips, P. Grother, and R. Micheals, "Evaluation methods in face recognition," in *Handbook of Face Recognition*. London, U.K.: Springer, 2011, pp. 551–574.

[27] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[28] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*. [Online]. Available: http://arxiv.org/abs/1603.00831

[29] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[30] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3047–3055.

[31] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3029–3037.

[32] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle, "Improving multi-frame data association with sparse representations for robust near-online multi-object tracking," in *Proc. ECCV*, Oct. 2016, pp. 774–790.

[33] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 418–425.

[34] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3786–3795.

[35] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous association graph fusion for target association in multiple object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3269–3280, Nov. 2019.

[36] H. Sheng, J. Chen, Y. Zhang, W. Ke, Z. Xiong, and J. Yu, "Iterative multiple hypothesis tracking with tracklet-level association," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3660–3672, Dec. 2019.

[37] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, "Motion segmentation & multiple object tracking by correlation co-clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, Jan. 2020.

[38] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[39] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear LSTM," in *Proc. ECCV*, Sep. 2018, pp. 208–224.

[40] H.-U. Kim and C.-S. Kim, "CDT: Cooperative detection and tracking for tracing multiple objects in video sequences," in *Proc. ECCV*, Oct. 2016, pp. 851–867.

[41] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4846–4855.

[42] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proc. ECCV*, Sep. 2018, pp. 379–396.

[43] Z. Fu, F. Angelini, J. Chambers, and S. M. Naqvi, "Multi-level cooperative fusion of GM-PHD filters for online multiple human tracking," *IEEE Trans. Multimedia*, vol. 21, no. 9, pp. 2277–2291, Sep. 2019.
[44] Q. Zhou, B. Zhong, Y. Zhang, J. Li, and Y. Fu, "Deep alignment network based multi-person tracking with occlusion and motion reasoning," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1183–1194, May 2019.
[45] Q. Zhou, B. Zhong, X. Lan, G. Sun, Y. Zhang, B. Zhang, and R. Ji, "Fine-grained spatial alignment model for person re-identification with focal triplet loss," *IEEE Trans. Image Process.*, vol. 29, pp. 7578–7589, 2020.
[46] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, Jul. 1983.
[47] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
[48] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. NIPS*, Dec. 2016, pp. 379–387.
[49] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.
[50] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. NIPS*, Dec. 2005, pp. 513–520.
[51] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, Mar. 2007, pp. 412–419.
[52] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. NIPS*, Dec. 2006, pp. 1473–1480.
[53] R. Min, D. A. Stanley, Z. Yuan, A. Bonner, and Z. Zhang, "A deep non-linear feature mapping for large-margin kNN classification," in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 357–366.
[54] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.
[55] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1563–1572.
[56] P. Oza and V. M. Patel, "C2AE: Class conditioned auto-encoder for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2302–2311.
[57] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2532–2541.
[58] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, Jun. 2019, pp. 6105–6114.
[59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[60] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[61] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.
[62] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[63] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[64] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.
[65] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2953–2960.
[66] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, May 2015, pp. 1–11.
[67] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with TrackletNet," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 482–490.
[68] L. Chen, H. Ai, R. Chen, and Z. Zhuang, "Aggregate tracklet appearance features for multi-object tracking," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1613–1617, Nov. 2019.
[69] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3987–3997.
[70] A. Maksai and P. Fua, "Eliminating exposure bias and metric mismatch in multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4634–4643.
[71] P. Chu and H. Ling, "FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6171–6180.
[72] Q. Liu, B. Liu, Y. Wu, W. Li, and N. Yu, "Real-time online multi-object tracking in compressed domain," *IEEE Access*, vol. 7, pp. 76489–76499, 2019.
[73] S.-H. Lee, M.-Y. Kim, and S.-H. Bae, "Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures," *IEEE Access*, vol. 6, pp. 67316–67328, 2018.
[74] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104423–104434, 2019.
[75] M. Babaee, Z. Li, and G. Rigoll, "A dual CNN–RNN for multiple people tracking," *Neurocomputing*, vol. 368, pp. 69–83, Nov. 2019.

HAN-UL KIM (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2014 and 2020, respectively. His research interests include computer vision and machine learning, especially in the problems of object tracking.

YEONG JUN KOH (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2011 and 2018, respectively. In March 2019, he joined as an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, where he is currently a Professor. His research interests include computer vision and machine learning, especially in the problems of video object discovery and segmentation.

CHANG-SU KIM (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Seoul National University, in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, USA. From 2001 to 2003, he coordinated the 3D Data Compression Group, National Research Laboratory for 3D Visual Information Processing, SNU. From 2003 and 2005, he was an Assistant Professor with the Department of Information Engineering, Chinese University of Hong Kong. In September 2005, he joined the School of Electrical Engineering, Korea University, where he is currently a Professor. He has published more than 250 technical papers in international journals and conferences. His research topics include image processing and computer vision. He received the Distinguished Dissertation Award for his Ph.D. degree, in 2000. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award from *Journal of Visual Communication and Image Representation* (JVCI). He is a member of the Multimedia Systems and Application Technical Committee (MSATC) of the IEEE Circuits and Systems Society. He is also an APSIPA Distinguished Lecturer for term 2017–2018. He served as an Editorial Board Member of JVCI and an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a Senior Area Editor of JVCI and an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA.