Applied Intelligence
https://doi.org/10.1007/s10489-022-03473-9

SSL-MOT: self-supervised learning based multi-object tracking

Sangwon Kim[1] · **Jimi Lee**[1] · **Byoung Chul Ko**[1]

Accepted: 4 March 2022

Abstract
Although the use of a Siamese network is the most popular approach in object tracking, it creates an undesirable trivial solution and requires a large amount of training data reflecting changes in the object's shape in every frame. To solve this problem, in this paper, a self-supervised learning method for multi-object tracking (SSL-MOT) based on a contrastive structure is proposed. Unlike the existing SSL, we adopt a generative adversarial network as a preprocessing step to generate various pose changes of tracking objects. A positive pair composed of the augmented image and pose data is applied to the SSL network to learn an encoder that can generate a non-collapsed output vector. To improve the discrimination power of the encoder output features, we propose an affinity correlation distance, which combines invariance and redundancy terms as a loss function for learning. During the test, because only the dot product between two output vectors of the tracker and detection was used for a data association, the computation time was significantly reduced, and thus real-time online tracking about 12 fps was possible. The proposed method is the first attempt to apply SSL to an online MOT. Experimental results on the MOT16, 17, and 20 challenge datasets proved that the proposed method is a fast and reasonable tracking method that occupies less memory and achieves an excellent tracking performance compared to other state-of-the-art methods.

Keywords Multiple object tracking · Self-supervised learning · Pose estimation

1 Introduction

One of the most important issues in multi-object tracking (MOT) is data association in terms of the tracking speed and accuracy. In a data association, online tracking associates trackers and detections using only the past and current frames, and offline-tracking associates trackers and detections using past and future frames. In the tracking-by-detection paradigm, it is important to track objects in real time; therefore, online tracking is more essential than offline matching. However, online tracking has a problem in that the accuracy of the data association is inferior to that of off-line tracking because future frame information is insufficient and must rely solely on object information detected in a few previous frames.

Although conventional data association for online tracking, including a greedy bipartite assignment [1] and optimal Hungarian algorithms [2] have been used, their performance is poor in fast-changing videos because they depend heavily on the appearance of the objects. Various data association methods based on few-shot learning [3] that are robust to changes in appearance have been proposed in the field of tracking and person re-identification. The aim of few-shot learning is to train a tracker based on representation learning and conduct prediction using the nearest-neighbor based on the similarity between objects. Although the nearest-neighbor approach is a non-parametric model and does not require training, its matching performance depends on the distance [3].

Among few-shot-based association techniques, to associate both detections and tracklets, Siamese network have received considerable interest in online tracking [4–9]. Because a Siamese network applies the same network to the detection and tracker and calculates the similarity based on the trainable weighted L1-distance in the output features, it

✉ Byoung Chul Ko
niceko@kmu.ac.kr

Sangwon Kim
eddiesangwonkim@gmail.com

Jimi Lee
jimi88lee@gmail.com

[1] Computer Engineering, Keimyung University, 1095, Dalgubeol-daero, Daegu, 42601, South Korea

Published online: 22 April 2022

does not need to maintain a separate network structure and has the advantage of fast tracking [7]. However, because Siamese network only deals with the optimization aspect of object association [10], its performance may be degraded if the appearance of the object is severely changed.

To improve the tracking performance by reflecting the various changes in objects, triplet [11] and quadruplet [12] network, which improve the Siamese network, have been proposed. In general, Siamese network and their derived networks [11, 12] represent images through supervised metrics and train networks with symmetric and asymmetric structures. After training, one-shot learning can be applied using the network features without retraining.

Problematically, however, with Siamese network, positive and negative pairs used for training must be prepared in advance, and the Siamese network itself causes an undesired trivial solution that collapses all outputs into a constant representation [13]. In addition, to increase the matching accuracy, the structure of the shared network should be deep and complex. In this case, real-time MOT becomes difficult because online tracking is proportional to the number of tracked objects in the scene.

Owing to this limitation of Siamese network, graph neural network (GNN) based MOT approaches have been proposed [10, 14, 15]. GNN-based MOT constructs a graph using detections and tracklet nodes and constructs edges using the geometric information of the nodes. Data association is conducted through an affinity matrix, which is the GNN output between detections and tracklets. GNN-based MOT has the advantage of designing a graph structure based on the interaction between detections and tracklets and measuring the similarity between two objects using node and edge features. However, as the number of tracking objects increases, the graph structure becomes more complex, and the number of computations increases. Therefore, although online learning is possible, real-time tracking becomes difficult.

CenterTrack [8] and TMOH [16] recently refined the public detection used to improve the tracking performance and speed. Although these methods can improve the MOT performance, they have limitations in that they rely heavily on a detection refinement.

Contribution of this study This paper proposes a self-supervised learning (SSL) based Siamese structure for a MOT data association. Some recent SSL models [13, 17–19] have solved the trivial constant representation problem for an input object, which is a limitation of the existing Siamese network structure. The proposed SSL-MOT structure was inspired by SSL models and allows the network to learn by itself with only positive samples during network training. In addition, SSL-MOT recognizes the shape change of given training objects and automatically augments the most similar poses through a pre-trained pose normalized-GAN (PN-GAN) [20].

Through comparative experiments conducted on the benchmark dataset, we show that SSL-MOT outperforms the Siamese network and other state-of-the-art (SOTA) MOT algorithms in terms of performance and speed. Our main contributions can be summarized as follows:

- This study is the first attempt to apply SSL to online MOT.
- To train SSL-MOT when considering the changes in pose of an object, we propose applying a new data augmentation based on both the PN-GAN results and the augmented images.
- Using a PN-GAN, we solved the problem of a lack of training data, and as a result, we designed an MOT model that can achieve better results even with a small number of data.
- The proposed loss function, affinity correlation distance (ACD) learns the encoder network to generate more discriminant feature vectors.
- With the exception of PN-GAN, all SSL-MOT modules can be trained using an end-to-end method, and because they do not require additional fine-tuning for classification, clustering, and regression, the trained modules can be applied.
- The proposed method occupies less memory and provides a fast and high tracking performance compared to SOTA methods.

2 Related studies

As described in the Introduction, various data association methods using a CNN [5], a GNN [10, 14, 15], LSTM [21–23], and a Siamese network [4–9] have been proposed for use in MOT. However, in this section, we focus on Siamese-network-based MOT methods that can obtain an optimal performance for online learning. We also introduce SSL methods for visual data clustering and classification.

Siamese network-based MOT Leal-Taixe et al. [4] proposed a Siamese CNN for learning the descriptor encoding of local spatiotemporal structures between two input image patches. To make the Siamese network more discriminative and robust to intra-class variations, a triple network [24] is proposed to build a loss function from three images (anchor, positive, and negative), not only for person re-identification [25]. Son et al. [12] proposed a quadruplet architecture using four images by modifying a Siamese and triplet network to learn the data associations for the MOT. In addition, Zou et al. [26] proposed the F-Siamese tracker to reduce the redundant search space and solve or relieve

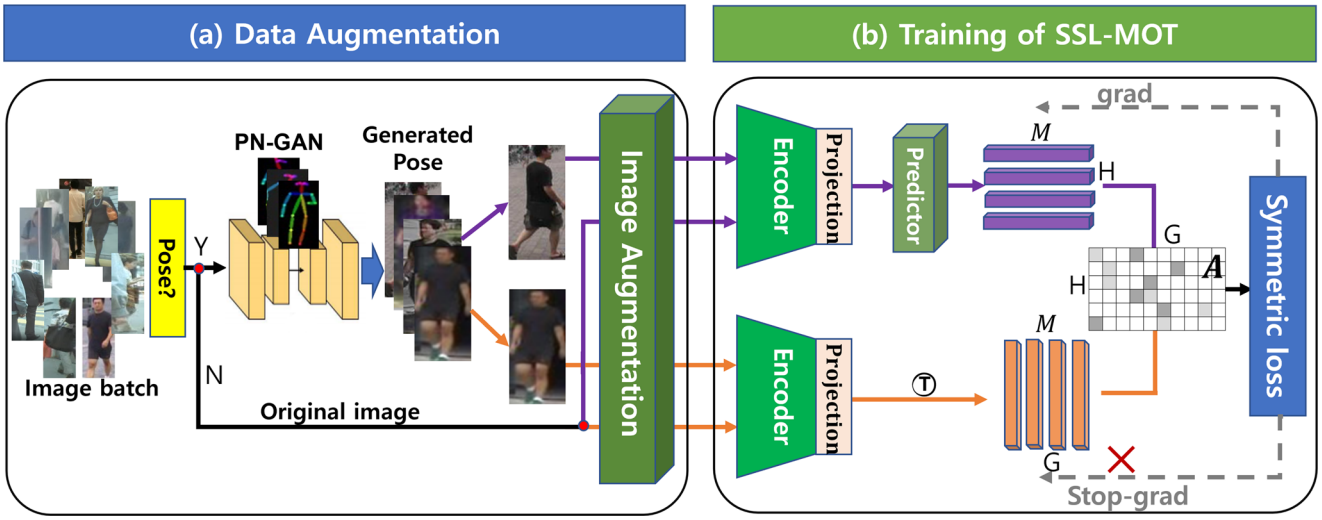

Fig. 1 Data augmentation and SSL-MOT network training process. Image augmentation is applied to one randomly selected pose generated by PN-GAN and the original image without including the full body

the occlusions caused by an object or cluttered background. Moreover, Lee et al. [7] proposed a SiameseRF that shares random forest rules instead of weights to improve the matching performance and solve the existing slow CNN-based tracking problem. Zhang et al. [8] extracted semantic features for MOT using Siamese network and multiple granularity color features. SiamMOT [9] proposed two variants of the Siamese tracker. These two motion models estimate the instance movement between two frames, allowing the detected instances to be connected. Siamese-network-based MOT is an interesting and optimal approach for online tracking; however, intra-class variations owing to a lack of training data and a collapsed representation of the output still remain problems to be resolved.

Self-supervised learning The preferred approach in SSL is learning a representation that is invariant to the distortion of the input sample while constantly avoiding trivial solutions. Typically, SimCLR [18], a type of contrastive learning, has a structure similar to that of a Siamese network, but is used to calculate the loss function in a different manner by defining positive and negative sample pairs. SwAV [27] is a clustering-based SSL that does not directly compare features extracted from images but assigns clusters to different augmented images derived from the same image and exchanges clusters for prediction. BYOL [17] trained an online network by augmenting the same image differently, instead of using the negative pairs used in SimCLR [18] to predict the target network representation of the image. Barlow twins [19] fed two distorted samples from the input image to the two same networks and measured the cross-correlation matrix between the two outputs. This method applies an objective function that naturally avoids this collapse by making the cross-correlation matrix as close to the identity matrix as possible. However, large batches are still required to avoid a collapse representation and are unsuitable for real-time association because the output dimension is as large as 4096. SimSiam [13] directly maximizes the similarity of two views of an image using the symmetric loss, without the use of negative pairs or momentum encoders. It can avoid collapsing by using a stop-gradient operation, operates with normal batch sizes and does not rely on large batch training. Similar to Chen and He's approach [13], SCARF [28] consists of two encoders with shared weights, although both branches have pre-trained heads with shared weights. However, the existing SSL methods do not apply strict end-to-end learning, and after training the encoder, supervised fine-tuned transfer learning must be separately conducted according to the downstream tasks such as clustering, classification, and regression.

In this paper, we propose an online SSL-MOT that is robust to changes in human pose and enables end-to-end learning (Fig. 1) and testing (Fig. 2). Although the basic structure of SSL is referred to the one proposed by Chen and He [13], the internal learning procedure has been reorganized for optimization with MOT. In particular, the newly proposed loss function and encoder is trained to obtain good discriminant power in terms of intra-class and inter-class variability on new data without applying separate transfer learning for downstream tasks, unlike [13]. For online-MOT, the proposed method can quickly compute the affinity scores of the trackers and detections using only the dot product of the output vectors generated by the trained encoder. Among several SSL methods, the proposed SSL-MOT is the first to apply SSL to online MOT.

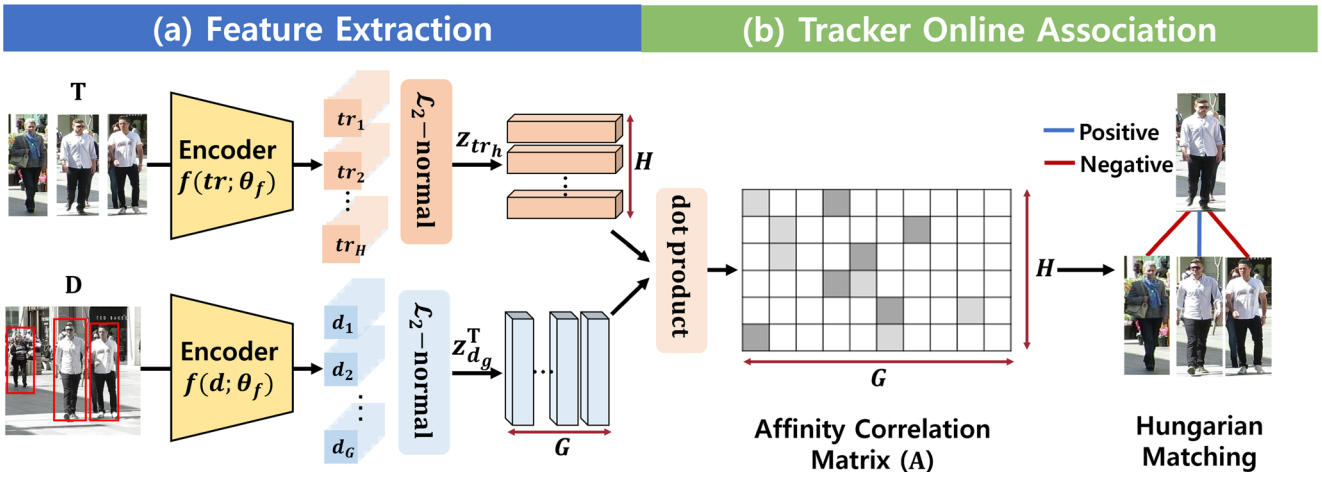

Fig. 2 Online association pipeline using SSL-MOT for test data

3 Method

In this section, we describe how to create a training dataset for SSL training using PN-GAN [20], and introduce SSL-MOT learning for data association using an augmented training set.

Data augmentation With SSL, positive or negative pairs are used for network training after taking several different augmentations of the image. The image augmentation techniques used for SSL consist of random cropping, resizing, flipping, color jittering, and Gaussian blur. However, in human tracking, pose changes frequently occur, and it is difficult to accurately match images through a simple image augmentation. Therefore, we adopt PN-GAN [20] as the preprocessing for image augmentation, as shown in Fig. 1a. In PN-GAN, when a single pedestrian image is fed, a synthesized image with the same identity and different poses is generated according to the multiple poses desired. By feeding the input image x to the PN-GAN, the generated pose images are obtained, and the image augmentation is randomly applied to the poses. However, because PN-GAN can generate new poses only when a full body is included in the image, PN-GAN is applied only when a pose is detected using OpenPose [29] as a preprocessing. Otherwise, only image augmentation is applied. The augmented images are used to train the complementary SSL-MOT, as shown in Fig. 1b.

Training of the SSL-MOT network SSL-MOT network takes two augmented images for the training encoder. The SSL-MOT network consists of two branches. The first network of branch-1 outputs a representation vector from the augmented input images with the encoder network $f(\cdot)$. The encoder network $f(\cdot)$ then combines ResNet-50 [30] as a backbone network with a projection MLP. Another network is responsible for transforming one view with a prediction head $h(\cdot)$ and matching it with the other. Branch-2 consists of only one encoder network $f(\cdot)$ and shares weights with the branch-1 encoder. The predictor prevents output collapsing during training and is not used during testing. At test time, the predictor is replaced with downstream tasks according to the data. To avoid requiring additional training for transfer learning after SSL-MOT training, the SSL-MOT needs to be trained end-to-end to output an affinity matrix that can check the similarity of two input images.

Let us suppose that the batch size is n and minibatch $\mathcal{B}$ is loaded with $\{x_i\}_{i=1}^{n}$. Assume that image augmentation is applied to image x_i, as shown in Fig. 1a, and one of the images is tracker tr_i and the other is detection d_i, respectively. Each augmented image is then fed to an encoder network $f(\cdot)$ that shares the weights, the output of which is denoted as $z_{tr_i} = f(tr_i)$ and $z_{d_i} = f(d_i)$. The z_{tr_i} of branch-1 is fed to the prediction head again to generate a new M-dimensional output vector p_{tr_i}, and is expressed as $p_{tr_i} = h(f(tr_i))$. In branch-2, the encoder outputs the same M-dimensional vector z_{d_i}. However, it is used without feeding the prediction head.

Previous SSL-based methods [13, 17–19, 27] learned representations from millions of images, and can perform transfer learning based on objectives (classification, clustering and regression) and fine-tune in some supervised tasks. These approaches are not strictly end-to-end learning, and after the encoder is trained, transfer learning must be separately performed again according to the purpose. Loss functions mainly used in SSL include the mean squared error (MSE) [17], L1-loss, cosine similarity [13], cross-entropy, and cross-correlation matrix [19]. The goal of these losses is to ensure that SSLs are trained to generate a learned representation such that the output vectors are clustered

well. Therefore, we need a loss function that can calculate the similarity between two images through one-stage end-to-end learning without additional transfer learning.

Instead of loss functions commonly used in SSL, we propose a new affinity correlation distance (ACD) that combines the cross-correlation calculated from the output batch matrix of image pairs with an invariance term and a redundancy reduction term inspired by Zbontar et al. [19]. However, unlike Zbontar et al. [19], we did not observe a representation correlation between feature dimensions, but rather a correlation between batch samples. In other words, the correlation value between similar sample pairs in the batch is high, and the correlation value is low for the neighboring samples.

First, we created a batch matrix of branch-1 $p_{tr} \in \mathbb{R}^{(n\times M)}$ from n mini-batches. In branch-2, we also created a batch matrix $z_d \in \mathbb{R}^{(n\times M)}$ from n mini-batch samples. The affinity correlation matrix A is then obtained through the dot product from the two batch matrices.

$$A = \mathcal{N}(p_{tr} \cdot z_d{}^{\intercal}) \in \mathbb{R}^{n\times n}, \tag{1}$$

where $\mathcal{N}$ denotes the L-2 normalization. The affinity correlation matrix A is used to calculate the proposed ACD.

$$ACD(p_{tr}, z_d) = \sum_i (1 - A_{ii})^2 + \lambda \sum_i \sum_{i\neq j} A_{ij}^2, \tag{2}$$

where λ is a weight that offsets the importance of the first and second terms of loss, and i and j are the indexes of the batch sample.

In the preceding invariance term of (2), the same sample pairs try to yield a value of zero, and in the rear redundancy reduction term, try to have a value close to zero for different samples.

By applying ACD, it is possible to solve the problem of applying extra transfer learning separately from representation learning, depending on the purpose. After training, the affinity score of the tracker and detection can be easily calculated using only the dot product of the output vectors generated by the learned encoder. Therefore, an encoder based on the ACD is trained to achieve a good discrimination power in terms of intra-class and inter-class variability for two inputs when the it is applied to new data without separately applying transfer learning for the downstream task as shown in Fig. 2

The weight-sharing SSL-MOT must guarantee the modeling invariance in which two observations produce the same output. Therefore, similar to Chen and He [13] and Grill et al. [17], the augmented images are fed to different branches, and $ACD(p_d, z_{tr})$ is calculated. In addition, the symmetry loss is calculated using these two ACDs:

$$\mathcal{L} = \frac{1}{2}\{ACD(p_{tr}, z_d) + ACD(p_d, z_{tr})\} \tag{3}$$

SSL-MOT applies a stochastic gradient optimizer (SGD) to backpropagate the average loss at every training step. It is important to note that we use a stop-gradient as in [13], which updates only the encoder and projection network parameters θ of branch-1 and does not update the encoder parameters of branch-2 as follows:

$$\theta_f, \theta_h \longleftarrow SGD(\theta, \nabla\mathcal{L}, \eta), \tag{4}$$

where η is the learning rate. By using a stop gradient, the output vector cannot be generated. The augmented tracker tr and detection d are fed alternately to branch-1 and branch-2 (with a stop gradient applied), and thus (3) is changed as follows:

$$\mathcal{L} = \frac{1}{2}\{ACD(p_{tr}, stopgrad(z_d)) + ACD(p_d, stopgrad(z_{tr}))\} \tag{5}$$

The process of training SSL-MOT using N unlabeled images is described in Algorithm 1.

Algorithm 1 Training of SSL-MOT.

Input: Optimizer: SGD, No. of images: N, Total number of optimization steps: K
Parameter: θ_f for f and θ_h for h
Output: Trained Siamese encoder (f_θ)

for $k = 1, ..., K$ **do**
  $\mathcal{B} \leftarrow \{x_i\ D\}_{i=1}^{n}$
  for $x_i \in \mathcal{B}$ **do**
    $tr_i \Leftarrow aug(x_i)$ ▷ image augmentation
    $d_i \Leftarrow aug(x_i)$
    $z_{tr_i} \leftarrow stopgrad(f(tr_i))$ ▷ stop gradient
    $z_{d_i} \leftarrow stopgrad(f(d_i))$
    $p_{tr_i} \leftarrow h(z_{tr_i})$ ▷ compute prediction of tr_i and d_i
    $p_{d_i} \leftarrow h(z_{d_i})$
  end for
  $ACD(p_{tr}, z_d)^k$, $ACD(p_d, z_{tr})^k$ ▷ compute affinity correlation distance
  $\mathcal{L}_k = \frac{1}{2}\{ACD(p_{tr}, z_d)^k + ACD(p_d, z_{tr})^k\}$ ▷ compute the symmetric loss
  $\delta\theta_k \leftarrow \partial_\theta \mathcal{L}_k$ ▷ compute a gradient
  $\theta_f, \theta_h \leftarrow SGD(\theta_k, \partial\theta_k, \eta)$ ▷ update parameters
end for

Tracker online association After training, SSL-MOT is used to generate an affinity matrix between the tracker and detection during actual tracking. In the t-th frame ($t > 1$), let a tracker set consisting of H trackers be $\mathbf{T}^{t-1} = \{tr_1^{t-1}, tr_2^{t-1}, ..., tr_H^{t-1}\}$, and let a detection set consisting of G detections be $\mathbf{D}^t = \{d_1^t, d_2^t, ...d_G^t\}$. An individual tracker and detection are represented as a vector consisting of the center position (x, y), width (wi), and height (he) elements such as $tr_h^t = [x_h^t, y_h^t, wi_h^t, he_h^t]$ and $d_g^t = [x_g^t, y_g^t, wi_g^t, he_g^t]$. In every frame, each element d_g of the detection set $\mathbf{D}$ is input to the trained encoder f, and L2-normalization is applied to the output to generate a new d-dimensional feature vector z_{d_g}. A set of transformed detections $\mathbb{Z}_d$ is obtained through the same operation for all elements of set $\mathbf{D}$. Similarly, the same encoder is applied to each element tr_i of the tracker to obtain a new feature vector z_{tr_h}. A set of transformed trackers $\mathbb{Z}_{tr}$ is also obtained through the same operation for all elements of set $\mathbf{T}$.

The goal of SSL-MOT is to estimate an $H \times G$ affinity correlation matrix $A = \{z_{hg}\}^{|\mathbf{T}^{t-1}| \times |\mathbf{D}^t|}$ based on the pairwise similarity of features extracted from $\mathbf{T}$ trackers and $\mathbf{D}$ detections in online MOT. To create an affinity matrix $\mathbf{A}$ between the trackers and detections, we transform the set $\mathbb{Z}_{tr}$ into an $(H \times d)$ matrix and transpose the set $\mathbb{Z}_d$ into $(G \times d)^\intercal$. We then take the dot product between two matrices, as shown in (6).

$$\mathbf{A} = \exp\{\frac{1}{\tau} \cdot (\mathbb{Z}_{tr} \cdot \mathbb{Z}_d^\intercal)\} \tag{6}$$

where τ is a constant value used to widen the difference between the output values.

As an element of $\mathbf{A}$, z_{hg} has a large value if d_g^t is associated with tracker $tr_h^{(t-1)}$ and z_{hg} has a small value if the two objects are not associated. In the affinity correlation matrix, if the affinity score is less than the threshold ρ_1, we filter the affinity score of the pair to zero to improve the matching speed. For the cost function c, we combine an affinity score with a relative L1-center distance $Dis(tr_h^{t-1}, d_g^t)$ in the appearance space using a weighted sum.

$$c(tr_h^{t-1}, d_a^t) = \alpha \cdot z_{hg} + \beta \cdot Dis(tr_h^{t-1}, d_g^t), \tag{7}$$

where α and β denote the weights, which are 0.5 and 0.5, respectively. Cost c is used for Hungarian matching.

Figure 2 shows the overall tracking process of the proposed SSL-MOT at test time.

Tracker state updating

In every frame except for the first frame, the trackers and detections generate trajectories by updating new connections through the proposed SSL-MOT-based data association and Hungarian matching. When a tracker is matched through a data association, the state of the tracker is updated for the next frame by combining the previous tracker with the current detection state as $tr_i^t = \delta \cdot tr_i^{t-1} + (1 - \delta) \cdot d_j^t$. Here, δ is an adjustable parameter that controls the state of the tracker and is normally set to 0.5.

We define the online MOT matching rules as follows: 1) If a detection does not match any tracker, this detection is assigned as a new potential tracker, and if the potential tracker matches more than ρ_2 times, it is newly assigned as a true tracker $\mathbf{T}^t \leftarrow \mathbf{T}^{t-1} \cup (d_a^t, h + 1)$; otherwise, it is recognized as a false tracker and removed from the tracker pool. 2) If the tracker does not match any detection, it may disappear temporarily through an occlusion, and thus it is not removed immediately and is observed while remaining in the tracker state for ρ_2 frames. However, if there is no match for more than ρ_2 frames, it is considered a missing tracker and removed from the tracker set $\mathbf{T}^t \leftarrow \mathbf{T}^{t-1} \not\ni tr_h^{t-1}$. The value of ρ_2 can also be adjusted depending on the nature of the dataset.

4 Experiment

In this section, we describe the detailed implementation of SSL-MOT, the dataset used for a performance comparison, and the evaluation metrics. We also verified the effectiveness of each component of the proposed algorithm.

Implementation details The proposed SSL-MOT was implemented using Pytorch and Pytorch-Geometric, and the model was trained and tested on an Intel Core i9-9900k CPU cluster using an Nvidia GeForce RTX 2080ti GPU. The network configuration and learning settings of the proposed method are as follows.

- OpenPose consists of a multistage CNN. The first stage predicts the part affinity fields, and the second stage predicts the confidence map. The PN-GAN generator is designed based on the ResNet-50 and is an encoder–decoder network, which generates four consecutive canonical poses for a given input image.
- The image augmentation pipeline consists of five image transformations (Gaussian blurring, horizontal flipping, perspective, cut-out, and brightness jittering), applied randomly with a certain probability.
- Encoder f is composed of ResNet-50 as the backbone and an MLP consisting of three additional layers. The batch size was 512, and batch normalization was used for each fully connected (fc) layer. SGD was used for the optimizer and learning rate of $\eta \times BachSize/256$, with a base $\eta = 0.05$, and the weight decay was 1×10^{-4}.
- Prediction network h consists of two fc layers, the output of which is 2048-d.

Table 1 Comparison of MOT performance according to the data augmentation configuration of the proposed SSL-MOT

Augmentation	MOTA↑	IDF1↑	FP↓
(1) w/ Image Aug.	59.6%	55.1%	922
(2) PN-GAN+ w/o Image Aug.	65.9%	**68.3%**	**749**
(3) PN-GAN+ w/ Image Aug.	68.1%	62.3%	758
(4) PN-GAN+OpenPose+w/ Image Aug.	**68.9%**	65.5%	763

Datasets For SSL-MOT training, we combined Market1501 [31] with MOT16 [32], and MOT17 [33] training datasets. The MOT20 [34] training dataset was not included in the training because of its severe occlusion. Market1501 contains 12,936 training images from six cameras. We conducted experiments on MOT challenge datasets, MOT16, MOT17 and MOT20. For a fair comparison with other methods, the MOT performance was measured using the public detections provided by MOT Challenge.

Metrics To evaluation the MOT performance of the proposed method, we used the following evaluation metrics: multi-object tracking accuracy (MOTA), identity F1 score (IDF1), number of false positives (FP), number of false negatives (FN), number of ID switches (IDsw), and Hz (frames per second).

5 Ablation studies

In this section, we describe a series of ablation studies conducted to understand the importance of the individual components of the proposed SSL-MOT. We measured the ablation performance using the MOT17 training set because the MOT evaluation can only be applied through the online evaluation server, and the performance evaluation is limited to four trials. As standard practice in the MOT literature [10, 35], we used five video sequences of the MOT17 training set (MOT17-02, 05, 10, and 11) and test set (MOT17-04, 09, and 13) filmed from a static and moving camera.

Effectiveness of PN-GAN image augmentation In SSL-based methods [13, 17–19], data augmentation is an extremely important factor influencing the performance of SSL and may require significant effort and expertise [19]. Unlike the existing SSL methods, the proposed method learns a generalized SSL-MOT network optimized for tracking by using PN-GAN with image augmentation. It recognizes the pose of a human and generates new poses that can be changed in the next frame instead of general image augmentation.

To prove the effectiveness of PN-GAN-based data augmentation, we conducted comparative experiments on four cases of augmentation: (1) common image augmentation of SSL, (2) PN-GAN without image augmentation, (3) PN-GAN with image augmentation, and (4) PN-GAN and OpenPose with image augmentation. With the proposed method, image augmentation was applied by selecting only a random patch, resizing, and horizontal flip, considering the frequent occlusions of pedestrians and the camera perspective.

As shown in Table 1, the overall performance of the PN-GAN-based methods was improved compared to the basic image augmentation, method (1), because SSL-MOT was trained by predicting the pose of an object. In the case of method (4), MOTA increased by 0.8% and IDF1 increased by 3.2% compared to method (3) because method (4) was trained by considering not only the pose change but also the image augmentation. Although method (2) showed a slightly better performance than method (4) in IDF1 and FP, method (4) showed a higher performance by more than 3% in MOTA which is an important indicator in MOT. Therefore, in subsequent experiments, we used the SSL-MOT model trained by combining PN-GAN and OpenPose using image augmentation.

Effectiveness of similarity metric for loss To measure the performance of the proposed similarity metric ACD, the image matching performance was measured while changing the similarity metric used for symmetry loss in the same network structure. During the experiment, we compared the performance of the three main methods used in the loss

Table 2 Performance evaluation according to similarity metric for loss function

Method	MOTA↑	IDF1↑	FP↓
MSE	60.3%	41.0%	1,247
Cosine similarity	65.7%	49.5%	964
Proposed ACD	**68.9%**	**65.5%**	**763**

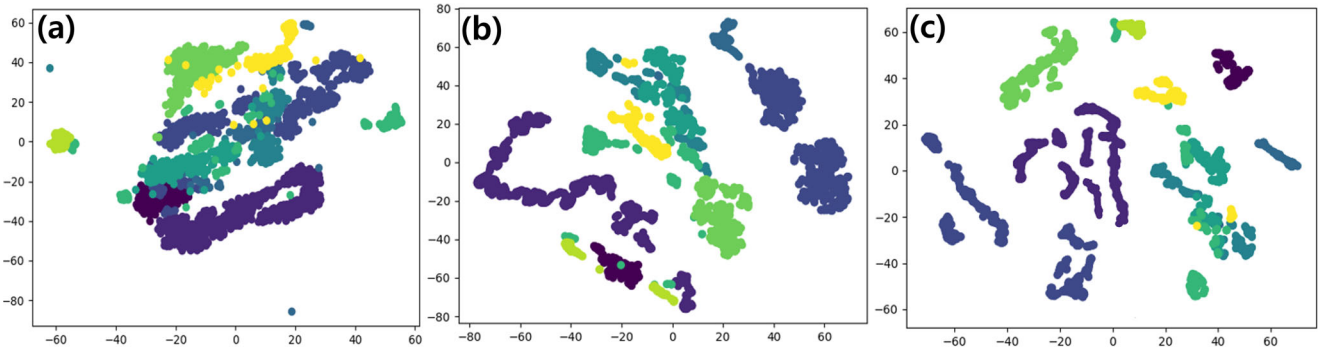

Fig. 3 Visualizations of 2,732 images for 10 people from the MOT17 data set, **a** MSE, **b** Cosine similarity, **c ACD**

Table 3 Comparison of MOT performance according to SSL approaches

Method	MOTA↑	IDF1↑	FP↓	Hz
BYOL	51.3%	36.3%	3,040	2.18
SimSiam	40.4%	21.1%	6,448	10.01
Barlow Twins	62.1%	51.3%	1,378	0.6
SSL-MOT	**68.9%**	**65.5%**	**763**	**13.2**

Table 4 Comparison of our proposed method with SOTA methods on the MOT16 dataset

Trackers	MOTA↑	IDF1↑	FP↓	FN↓	IDsw↓	Hz
BLSTM_MTP_O	48.3%	53.5%	9,792	83,707	**435**	**21.0**
MFI_TST	59.9%	**58.7%**	3,660	68,923	616	0.7
ArTIST-T	56.6%	57.8%	**3,532**	75,031	-	4.5
SiamesRF	57.9%	41.7%	8,196	66,538	2,051	11.1
SSL-MOT	**62.1%**	45.2%	4,760	**63,238**	1,022	13.7

Table 5 Comparison of our proposed method with SOTA methods on the MOT17 dataset

Trackers	MOTA↑	IDF1↑	FP↓	FN↓	IDsw↓	Hz
CenterTrack	**61.5%**	**59.6%**	14,076	200,672	2,583	17
BLSTM_MTP_O	51.5%	54.9%	29,616	241,619	2,566	**20.1**
MFI_TST	60.1%	58.8%	13,503	209,475	**2,065**	2.2
ArTIST-T	56.7%	57.5%	**12,353**	230,437	-	4.5
SiamesRF	55.8%	46.7%	12,822	233,573	3,174	12.4
SSL-MOT	**61.5%**	55.0%	16,841	**197,485**	2,957	12.9

Table 6 Comparison of our proposed method with SOTA methods on the MOT20 dataset

Trackers	MOTA↑	IDF1↑	FP↓	FN↓	IDsw↓	Hz
TMOH	**60.1%**	**61.2%**	38,043	**165,899**	2,342	0.6
MFI_TST	59.2%	59.1%	36,402	172,785	1,914	0.5
ArTIST-C	53.6%	51.0%	**7,765**	230,576	**1,531**	1.0
SSL-MOT	48.1%	37.5%	70,716	186,779	11,247	**1.6**

functions of SOTA SSL: MSE [19], cosine similarity [13], and proposed ACD.

As shown in Table 2, MSE showed the lowest performance in three metrics. Unlike cosine similarity, which considers only the angular distance without considering the magnitude, ACD showed the best performance using the dot product result that can keep the angular and magnitude properties between the output vectors.

The encoder is trained to reduce intra-class variability and increase the inter-class variability, and the encoder output vector has the property of increasing inter-class discrimination using the invariance and redundancy reduction term of ACD. To test the discrimination power of the output vector of the encoder network trained using the ACD, the output vector was visualized using t-SNE. After encoder training, 10 people images were input and expressed as t-SNE. As shown in Fig. 3, the proposed ACD achieved a good discrimination power in terms of intra-class and inter-class variability compared to MSE and cosine similarity, and maximized the similarity between identical persons within the embedded feature space.

MOT comparisons of SSL approaches To find the most useful SSL structure for MOT, the proposed SSL-MOT was tested using the MOT17 dataset by measuring four performance metrics with BYOL [17], SimSiam [13], Barlow Twins [19], and SSL-MOT, which were trained using the same data augmentation.

As shown in Table 3, the proposed SSL-MOT exhibited the best performance in all metrics. SimSiam and SSL-MOT have similar basic network structures and can generate non-collapsed output vectors using a stop-gradient and symmetric loss. However, SSL-MOT was able to improve the performance of tracker matching compared to SimSiam through an ACD-based loss function. During the test, SSL-MOT can estimate the exact affinity score by calculating only the dot product on the output vectors without fine-tuning, and thus even when there are many matching targets, a shorter data association time and faster speed (Hz) can be obtained in comparison to other methods.

Additionally, because SSL-MOT may have different performance depending on the encoder backbone network, we compared the MOTA and speed while changing the backbone network into three types: ResNet-18, ResNet-50, and ResNet-101. When ResNet-50 was used, like the result in Table 3, MOTA was 68.9% and the speed was 13.2Hz. However, when ResNet-18 was used, the MOTA dropped significantly to 1.2%(67.7%), but the speed slightly increased to 0.3Hz (13.5Hz). When ResNet-101 was used, MOTA slightly increased by 0.1%(69.0%), but speed decreased by 2.1Hz (11.2 Hz). From the experimental results, we can see that when ResNet-50 is used as the backbone of the SSL-MOT encoder, it is most reasonable in terms of accuracy and speed.

6 Evaluation on MOT challenge benchmark

To validate the MOT performance of the proposed method, we conducted comparative experiments using the SOTA MOT methods on the MOT16, 17, and 20 test datasets. Comparative methods were selected from among the SOTA online tracking methods.

The MOT performance on the MOT16 dataset was measured using four SOTA online MOT approaches as follows: (1) BLSTM_MTP_O [23], (2) MFI_TST [36], (3) ArTIST-T [37], (4) SiamesRF [7], and (5) our proposed SSL-MOT. As shown in Table 4, the proposed method achieved the highest MOTA and FN scores while maintaining fast tracking speed in all methods. MFI_TST showed the best performance in IDF1, but the tracking speed is 0.7, which is the slowest and most important factor, MOTA, is 2.2% lower than the proposed method. BLSTM_MTP_O had a high tracking speed, but showed the lowest MOTA as a trade-off for speed.

In the MOT17 dataset, CenterTrack [38] was added to the same online-based MOT method. As shown in Table 5, CenterTrack showed a higher performance for MOT17 compared to the comparative methods. However, because CenterTrack uses the refinement of public detections, it is not directly comparable to other methods that use only MOT public detections [23]. Except for CenterTrack, the proposed method achieved the best MOTA and FN as similar to MOT16 results. Although BLSTM_MTP_O showed the fastest speed, this method showed 10% lower MOTA than the proposed method.

In the MOT20 dataset, TMOH [16] was added instead of CenterTrack. As shown in Table 6, although TMOH showed the best performance among the online MOT methods, this method also uses a refinement of public detection such as CenterTrack and has a disadvantage in that the processing speed is extremely low at 0.6 Hz. The proposed method showed a lower performance because we did not use a detection refinement and excluded the training data of MOT20 from training. However, the proposed method showed the fastest tracking speed of 1.6 Hz online.

From the experiments, we can confirm that the proposed method achieves an excellent performance in terms of tracking accuracy and speed among the SOTA online MOT approaches. However, the ID switch and FP caused by the long and frequent occlusion in MOT20 cannot be solved with the proposed SSL alone. Therefore, if motion prediction such as a Kalman filter is added, performance can be improved. Since the proposed SSL network can generate non-collapsed output vectors in various applications, we can expand and apply it to person re-identification [39] or social representation [40] as well as 3D mesh simulation [41] in future studies.

7 Conclusion

In this study, we introduced a new tracking approach that combines the SSL and MOT. Through the proposed SSL, we generated a non-collapsed output vector, and through ACD loss, we were able to learn an SSL network that improved the feature discrimination. The proposed SSL-MOT can measure the similarity between objects using only the dot product without an additional network in the output vector; thus, object tracking is possible within a short period of time. In addition, it has the advantage of being robust to pose changes in objects through learning using PN-GAN.

Acknowledgements This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), Ministry of Education, under Grant 2019R1I1A3A01042506.

References

1. Shu G, Dehghan A, Oreifej O, Hand E, Shah M (2012) Part-based multiple-person tracking with partial occlusion handling. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 1815–1821
2. Kuhn HW (1955) The hungarian method for the assignment problem. Naval Res Logist Quart 2(1-2):83–97
3. Kim H-U, Koh YJ, Kim C-S (2020) Online multiple object tracking based on open-set few-shot learning. IEEE Access 8:190312–190326
4. Leal-Taixé L, Canton-Ferrer C, Schindler K (2016) Learning by tracking: Siamese cnn for robust target association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 33–40
5. Chu P, Ling H (2019) Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6172–6181
6. Lee S, Kim E (2018) Multiple object tracking via feature pyramid siamese networks. IEEE Access 7:8181–8194
7. Lee J, Kim S, Ko BC (2020) Online multiple object tracking using rule distillated siamese random forest. IEEE Access 8:182828–182841
8. Zhang Z, Zhang Y, Cheng X, Lu G (2021) Siamese network for object tracking with multi-granularity appearance representations. Pattern Recogn 118:108003
9. Shuai B, Berneshawi A, Li X, Modolo D, Tighe J (2021) Siammot: Siamese multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 12372–12382
10. Papakis I, Sarkar A, Karpatne A (2020) Gcnnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. arXiv:2010.00067
11. Ristani E, Tomasi C (2018) Features for multi-target multi-camera tracking and re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6036–6046
12. Son J, Baek M, Cho M, Han B (2017) Multi-object tracking with quadruplet convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5620–5629
13. Chen X, He K (2021) Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 15750–15758
14. Dai P, Weng R, Choi W, Zhang C, He Z, Ding W (2021) Learning a proposal classifier for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2443–2452
15. He J, Huang Z, Wang N, Zhang Z (2021) Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5299–5309
16. Stadler D, Beyerer J (2021) Improving multiple pedestrian tracking by track management and occlusion handling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 10958–10967
17. Grill J-B, Strub F, Altché F, Tallec C, Richemond PH, Buchatskaya E, Doersch C, Pires BA, Guo ZD, Azar MG et al (2020) Bootstrap your own latent: A new approach to self-supervised learning. arXiv:2006.07733
18. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR, pp 1597–1607
19. Zbontar J, Jing L, Misra I, LeCun Y, Deny S (2021) Barlow twins: Self-supervised learning via redundancy reduction. arXiv:2103.03230
20. Qian X, Fu Y, Xiang T, Wang W, Qiu J, Wu Y, Jiang Y-G, Xue X (2018) Pose-normalized image generation for person re-identification. In: Proceedings of the European conference on computer vision (ECCV), pp 650–667
21. Lu Y, Lu C, Tang C-K (2017) Online video object detection using association lstm. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2344–2352
22. Liu H, Zhang H, Mertz C (2019) Deepda: Lstm-based deep data association network for multi-targets tracking in clutter. In: 2019

22th International Conference on Information Fusion (FUSION). IEEE, pp 1–8
23. Kim C, Fuxin L, Alotaibi M, Rehg JM (2021) Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9553–9562
24. Ge W (2018) Deep metric learning with hierarchical triplet loss. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 269–285
25. Hermans A, Beyer L, Leibe B (2017) In defense of the triplet loss for person re-identification. arXiv:1703.07737
26. Zou H, Cui J, Kong X, Zhang C, Liu Y, Wen F, Li W (2020) F-siamese tracker: A frustum-based double siamese network for 3d single object tracking. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 8133–8139
27. Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A (2020) Unsupervised learning of visual features by contrasting cluster assignments. arXiv:2006.09882
28. Bahri D, Jiang H, Tay Y, Metzler D (2021) Scarf: Self-supervised contrastive learning using random feature corruption. arXiv:2106.15147
29. Cao Z, Simon T, Wei S-E, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7291–7299
30. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
31. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: A benchmark. In: Proceedings of the IEEE international conference on computer vision, pp 1116–1124
32. Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K (2016) Mot16: A benchmark for multi-object tracking. arXiv:1603.00831
33. MOT Benchmarks https://motchallenge.net/data/MOT17/
34. Dendorfer P, Rezatofighi H, Milan A, Shi J, Cremers D, Reid I, Roth S, Schindler K, Leal-Taixé L (2020) Mot20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003
35. Li J, Gao X, Jiang T (2020) Graph networks for multiple object tracking. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 719–728
36. Yang J, Ge H, Yang J, Tong Y, Su S (2021) Online multi-object tracking using multi-function integration and tracking simulation training. Appl Intell:1–21
37. Saleh F, Aliakbarian S, Rezatofighi H, Salzmann M, Gould S (2021) Probabilistic tracklet scoring and inpainting for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 14329–14339
38. Zhou X, Koltun V, Krähenbühl P (2020) Tracking objects as points. In: European Conference on Computer Vision. Springer, pp 474–490
39. Si T, He F, Wu H, Duan Y (2022) Spatial-driven features based on image dependencies for person re-identification. Pattern Recogn 124:108462
40. Pan Y, He F, Yu H (2020) Learning social representations with deep autoencoder for recommender system. World Wide Web 23(4):2259–2279
41. Liang Y, He F, Zeng X (2020) 3d mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. Integr Comput-Aided Eng (Preprint):1–19

Sangwon Kim received a BS and MS degree in Computer Engineering from Keimyung University, Daegu, Korea, in 2018 and 2020. He is currently a Ph.D student in the computer vision and pattern recognition laboratory of Keimyung University, Korea. His current research interests include depth estimation from monocular images and deep learning model compression.

Jimi Lee received a BS and MS degree in Computer Engineering from Keimyung University, Daegu, Korea, in 2019 and 2021. She is currently a researcher in Hanjin Information Systems & Telecommunication Co. She received a best paper award from IEIE conference in 2020. Her current research interest includes multiple object tracking, deep learning and depth estimation from monocular image.

Byoung Chul Ko received his B.S. degree from Kyonggi University, Suwon, Korea, in 1998 and M.S. and Ph.D. degrees in Computer Science from Yonsei University, Seoul, Korea, in 2000 and 2004, respectively. From 2004 to 2005, he was a senior researcher at Samsung Electronics, Suwon, Korea, where he worked on the Ubiquitous Robot Companion project. He is currently a Professor in the Department of Computer Engineering and Vice Dean of College of Engineering, Keimyung University, Daegu, Korea. He is also a Chief of the KMU AI Fusion Research Center. He was selected as the Best Researcher and a Lecturer at Keimyung University, in 2013, 2014, 2015, 2018-2021. His current research interests include interpretable machine learning, deep model compression, advance driver assistance systems, and biomedical image processing. He has been a Committee Member of ACPR. He has received excellent paper awards from the several conferences. He is a Senior member of IEEE and has been severed in various international journals and conferences, including a Special Issue Editor for the Sensors.