applied sciences

MDPI

Article

Multiple Object Tracking Using Re-Identification Model with Attention Module

Woo-Jin Ahn [1], Koung-Suk Ko [1], Myo-Taeg Lim [1,*], Dong-Sung Pae [2] and Tae-Koo Kang [3,*]

[1] School of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea
[2] Department of Software, Sangmyung University, Cheonan 31066, Republic of Korea
[3] Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan 31066, Republic of Korea
* Correspondence: mlim@korea.ac.kr (M.-T.L.); tkkang@smu.ac.kr (T.-K.K.)

Abstract: Multi-object tracking (MOT) has gained significant attention in computer vision due to its wide range of applications. Specifically, detection-based trackers have shown high performance in MOT, but they tend to fail in occlusive scenarios such as the moment when objects overlap or separate. In this paper, we propose a triplet-based MOT network that integrates the tracking information and the visual features of the object. Using a triplet-based image feature, the network can differentiate similar-looking objects, reducing the number of identity switches over a long period. Furthermore, an attention-based re-identification model that focuses on the appearance of objects was introduced to extract the feature vectors from the images to effectively associate the objects. The extensive experimental results demonstrated that the proposed method outperforms existing methods on the ID switch metric and improves the detection performance of the tracking system.

Keywords: computer vision; deep learning; multiple object tracking; re-identification

check for updates

Citation: Ahn, W.-J.; Ko, K.-S.; Lim, M.-T.; Pae, D.-S.; Kang, T.-K. Multiple Object Tracking Using Re-Identification Model with Attention Module. *Appl. Sci.* **2023**, *13*, 4298. https://doi.org/10.3390/app13074298

Academic Editor: Rocco Furferi

Received: 9 February 2023
Revised: 23 March 2023
Accepted: 27 March 2023
Published: 28 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Multiple object tracking (MOT) aims at estimating the trajectories of multiple objects and re-identifying the objects appearing in consecutive frames of videos and real-time cameras. It has been a longstanding objective in the field of computer vision including action recognition [1], sports video analysis [2], closed-circuit television analysis [3], and driving environment recognition [4].

Most of the existing MOT methods [5,6] are based on the two-stage detection-based tracking (DBT) model, which comprises a detection model and an association stage, as in Figure 1. The detection model estimates the bounding box position of the objects in an image, and then, the association stage extracts the features from each detected box to track the objects.

Recent DBT systems have aimed at improving the performance of the detector or the association stage. For the detector, the rise of the convolutional neural network (CNN) [7] has greatly enhanced the accuracy and efficiency of object detection. Meanwhile, for the association stage, the simple online and real-time tracking (SORT) algorithm [8] has been proposed to handle the cross-regions by combining the Kalman filter with the detection system. DeepSORT [9] was found to show significantly improved performance when the SORT algorithm was combined with a re-ID network that considers appearance information. However, it still loses the ID when the appearances of the objects are similar.

To address this issue, we introduced a triplet-based attention association network to fully leverage the visual information of the object. The network was trained using a triplet loss function to estimate whether a given sample is the same as a reference sample or not. To capture the important visual representation of the object, we employed convolutional attention modules in the association network. We further improved the association network by pretraining on the re-ID dataset, which included various perspectives of the object from

the tracking dataset. The results, demonstrated on the KITTI tracking dataset, showed that the proposed method more effectively handles occlusive problems in multi-object tracking than previous studies. The contributions of this research can be summarized as follows:

- We propose a triplet-based association network for multi-object tracking with a novel tracking batch sampler method that can discriminate objects from a similar viewpoint.
- We employed a convolutional attention mechanism in our association network to selectively focus on the features of the object, by assigning different weights to different regions.
- We demonstrate that the proposed method not only outperforms existing methods in terms of switching IDs, but also improves the detection and tracking performance.

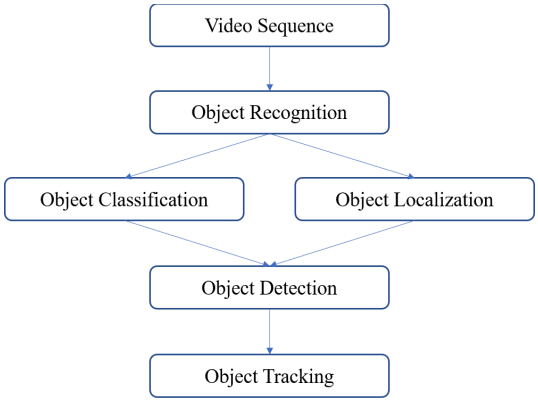


Figure 1. Flowchart of an MOT system.

2. Related Works

2.1. Object Detection

Object detection refers to the localization and classification of specific objects present in an image. Currently, machine-learning- and deep-learning-based approaches are popular for object detection. Machine learning approaches include Viola–Jones object detection based on Haar features [10] and approaches involving the scale-invariant feature transform and the histogram of oriented gradients [11].

Advances in research on CNNs have significantly influenced the field of object detection [12]. Deep-learning-based methods can be typically separated into two main streams: two-stage detectors and one-stage detectors.

Two-stage detectors such as the region-based CNN (R-CNN) [7], Fast-R-CNN [13], and Faster-R-CNN [14,15] combine the region proposal network and classification network. The R-CNN was the first model to use a CNN for object detection, which has proven that CNNs can exhibit high performance in object detection and classification problems. The R-CNN includes two stages: the region proposal stage and the classification stage. The region proposal stage determines the region of an object independent of the class. Subsequently, each candidate region is assigned to a CNN to predict the class of the object location. However, a disadvantage of this method is that applying CNNs to all candidate regions involves considerable computation. To address this shortcoming, researchers have developed improved algorithms such as Fast R-CNN and Faster R-CNN. However, these algorithms are much more time-consuming than the one-stage detector.

On the other hand, the one-stage detectors include you only look once (YOLO) [16] and the single shot multibox detector (SSD) [17], where the CNN is used with the region proposal in a single shot. A one-stage detector typically includes an algorithm such as YOLO, the SSD, or CenterNet [18]. The YOLO series [19,20] underwent an upgrade of its own algorithm, resulting in significant performance gains. YOLO divides grid cells into input images and uses a CNN to significantly increasethe computational speed. However, methods involving grid cells pose challenges in the detection of objects smaller than the

grid cells. The SSD overcomes this problem by using a one-stage detector and employing a default bounding box, which shows superior performance over the two-stage detector.

In this paper, we adopted a single-stage detector YOLOv3 model to locate the objects in each frame of an image, which is capable of high accuracy and high speed.

2.2. Multiple Object Tracking

Most studies on MOT focus on DBT, which aims to connect the same objects frame to frame according to the detection results of the detector. There are online and offline tracking methods for DBT systems [5]. An online tracking method locates and tracks objects in the previous frame according to the current frame. By contrast, an offline tracking method uses all frames as the input to provide the tracking results. It considers past, as well as future frames for tracking. Therefore, online tracking is used in real-time tracking.

For the association stage, the correlated information of different frames should be used to determine the similarity between objects. One of the basic tracking methods employs the intersection over union (IoU) metric. The detection result is linked by selecting the track with the largest IoU in the previous tracks, as in Figure 2.

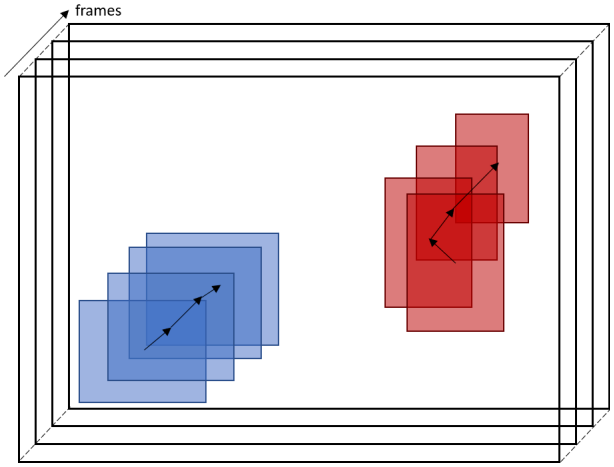


Figure 2. Example of tracking using the IOU.

The well-known SORT predicts the position of a bounding box CNN detection and estimates the trajectory by using a Kalman filter [21], then matches the position by using the Hungarian algorithm [22]. Using IoU-based methods can be effective for tracking a few objects. However, the performance degrades when a large number of objects exist in the images, and also, it incorrectly connects the objects in overlapped or separated situations. Recently, DeepSORT [9] overcame this problem adopting a CNN model for the association stage. The network can consider image information, as well as location information in the bounding boxes. However, the performance degrades when the tracked object has a similar appearance.

To overcome this problem, we considered a triplet loss that can discriminate the details of the visual appearance. Moreover, we improved the association network by employing a convolutional attention module.

3. Proposed Method

This section describes our proposed MOT algorithm. Figure 3 shows the overall flow of the proposed MOT algorithm. Note that the basic structure is similar to the algorithmic flow of SORT [8] and DeepSORT [9]. Given an input image acquired from a video or real-time camera, objects are located by the detector. Here, we employed YOLOv3 [20] as the object detector. Then, objects are separately extracted from the frame and are subjected to the proposed association stage, seeking to associate them with the objects in the previous frame or assigning a new ID for unmatched objects.

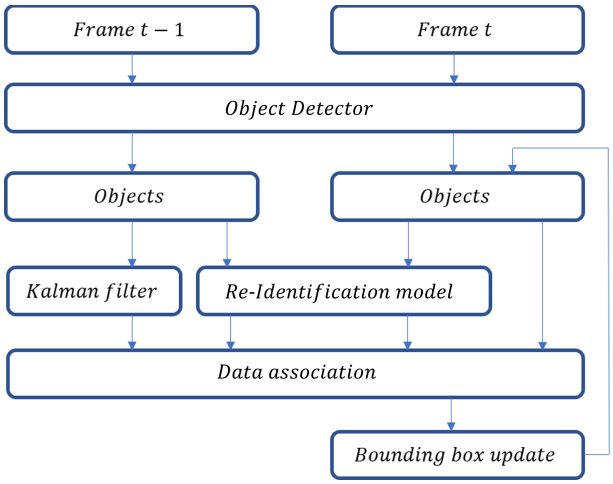


Figure 3. Flowchart of the proposed method.

3.1. Object Detector

We noticed that the performance of detection-based tracking is proportional to the performance of the detector. If the object is not detected, it is excluded from consideration, which is highly likely to result in incorrect ID assignment. Thus, we adopted a single-stage detector YOLOv3 model for the detector of the DBT system, which is capable of high accuracy and high speed, to locate the objects in each frame of an image.

3.2. Object Association

To determine the unique IDs to detect objects consistently across frames, we adopted a strategy inspired by deepSORT, which leverages two types of information to verify the identity of objects detected in consecutive frames. Figure 4 illustrates the purpose of the data association stage.

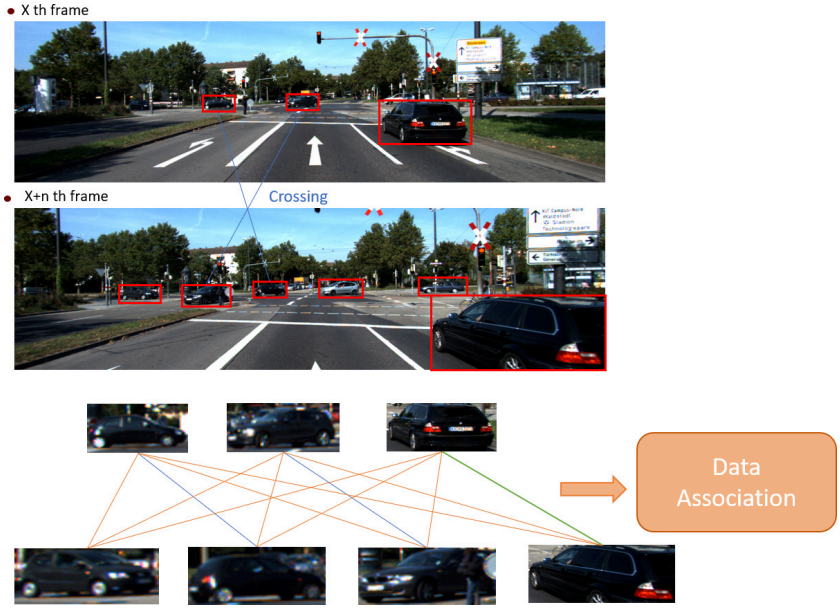


Figure 4. Structure of the data association stage. The red boxes represent the result detected from the object detector.

The first is the object's predicted position, obtained by applying a Kalman filter to the detection results. The second is the object's appearance features, extracted using a deep learning model. By combining these two sources of information, we generated a score matrix that provides a more reliable estimate of object identities across frames as follows:

$$c(i,j) = \lambda IoU(B_i, B_j) + (1-\lambda)Triplet(i,j), \quad (1)$$

where λ denotes a weighting factor. The details of each information function $IoU(B_i, B_j)$ and $Triplet(i,j)$ are described in the sections below. Once the cost matrix $c(i,j)$ between the current track and the detection results is calculated, an ID is assigned to the object with the maximum value via the Hungarian algorithm [22]. Unmatched objects among the detection results are provided a new ID. Objects that do not appear for a certain period during the tracking are deleted to ensure a high computational speed.

3.2.1. Kalman-Filter-Based Object Localization

The Kalman filter plays an important role in the DBT system by predicting the state of an object in future frames and linking it with existing tracks. Specifically, it uses information about the object's bounding box detected in the current frame to estimate its position in the next frame. In our implementation, we employed a linear constant velocity model for computational efficiency. Although extended Kalman filters can handle nonlinearity, they may diverge if the initial state estimation is incorrect. To address this, unscented Kalman filters can be used with sampling techniques, but these are computationally expensive. Since Kalman filters are suited for continuous systems, they are particularly useful in memory-constrained environments. Furthermore, since they utilize a recursive function, they do not require a history beyond the previous state, which enables efficient memory management.

The association between the ith object in the previous track (time t) and the jth object detected in the current frame (time $t+1$) is calculated. The IoU between the Kalman-filter=predicted bounding box of i, B_i and the detected bounding box of j, B_j, to be displayed in the current frame is computed as follows:

$$IoU(B_i, B_j) = \frac{area(B_i \cap B_j)}{area(B_i \cup B_j)}. \quad (2)$$

For overlapping images, the IoU is calculated as shown in Figure 5. Through the above equation, the object shown in the current frame is calculated to be deeply related to the tracks in the coordinates.

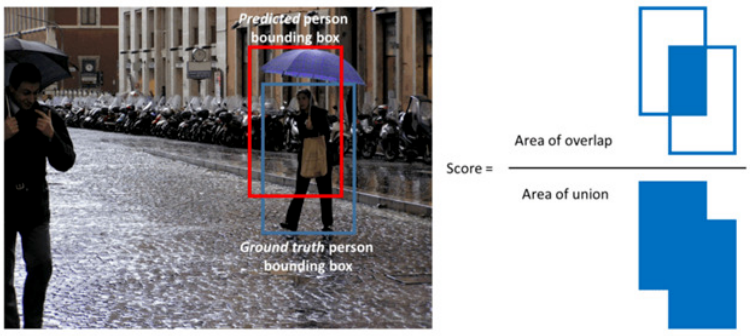


Figure 5. Intersection over union (IoU).

3.2.2. Triplet-Based Association Network with Attention

The Kalman filter, which solely relies on boundary box coordinates, is prone to frequent ID switches when objects overlap or are separate. To overcome this limitation, we propose a solution by constructing an embedding network to incorporate image information and compute a similarity matrix for the IDs across images. The embedding network was built using a triplet loss and an attention module.

Network Architecture

An attention block was added to the embedding network configuration used in [9] by inserting it after the residual block [23]. The attention module utilizes the method proposed in CBAM [24], which has been demonstrated to enhance performance by adding attention blocks to bottleneck locations. Attention blocks have a negligible impact on the existing parameters since they do not have a large number of parameters. Consequently, they can enhance performance while maintaining the computational speed. The detail of the attention association network architecture is shown in Figure 6.

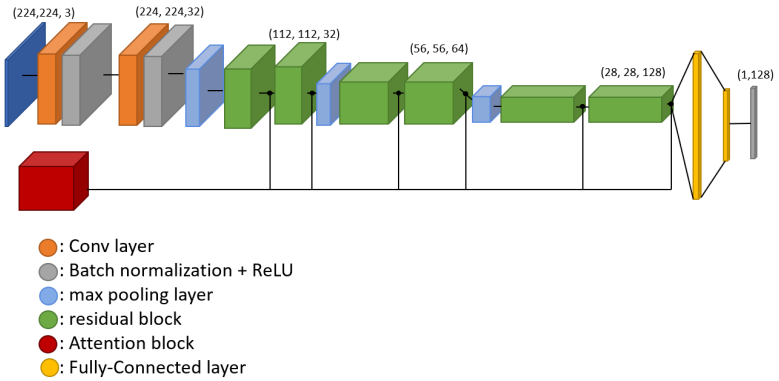


Figure 6. Network architecture of proposed embedding network.

Triplet Loss

The loss function was designed to not only bring semantically similar inputs closer together in the embedding space, but also to ensure that dissimilar inputs are separated by a significant distance. This results in a discriminative embedding space where each input is mapped to a unique and distinguishable feature vector.

The triplet loss is computed using three inputs: an anchor image (used as a reference), a positive image with the same ID as the anchor, and a negative image with a different ID, as shown in Figure 7. The embedding network shares the weights to compute the feature vectors of these images and learn to distinguish whether the images represent the same object, which results in Figure 8.

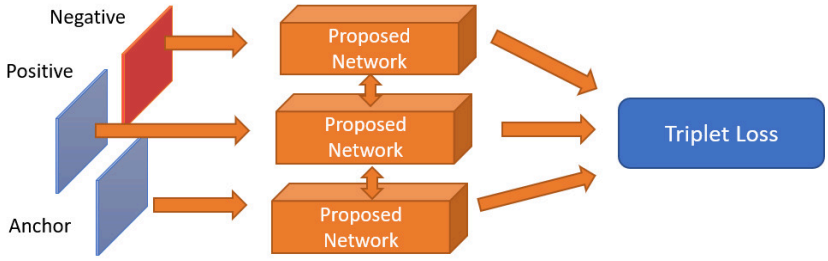


Figure 7. Schematic of the triplet structure of the proposed network.

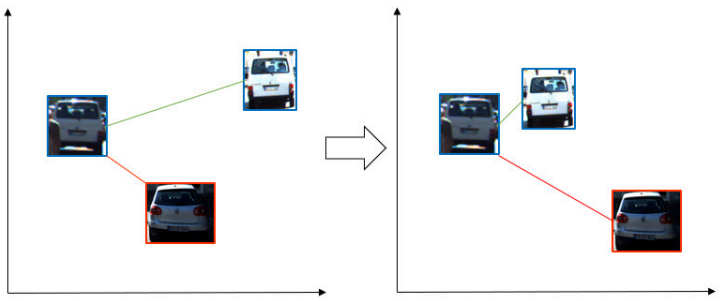


Figure 8. Example of metric learning.

The triplet loss was designed to minimize the distance between an anchor and the positive image and to maximize the distance between an anchor and the negative image by using the feature vector obtained from the CNN. The distance between feature vectors is calculated as the L2-norm as follows:

$$Triplet(i,j) = \| f(i) - f(j) \|_2, \tag{3}$$

where i and j are the input images, with one of them being an anchor image and the other a positive or negative image. $f(\cdot)$ represents the proposed embedding network. The triplet loss is defined as:

$$l_{triplet}(a,p,n) = [D_{a,p} - D_{a,n} + \alpha]_+, \tag{4}$$

where a, p, and n are the anchor, positive image, and negative image and α denotes the margin between the distances. The network was trained to minimize the triplet loss.

Negative Hard Sampler

The metric learning process can be unstable and may oscillate without converging, as the labels are not quantified and the model expects clustered data in the embedding space. To address this issue, Reference [25] proposed a negative hard sampler method that selects negative samples greater than the distance of the positive samples relative to an anchor during triplet construction. This method can facilitate fast convergence and high performance. However, for tracking datasets, a different sampler construction method may be more appropriate. Tracking datasets consist of several videos, and objects are typically tracked within the same video. Therefore, it is more appropriate to select hard negative samples from the same video rather than from other videos, as this can facilitate discrimination learning for overlapping objects and similar backgrounds within the same video. Positive samples of the same class are also typically present in the same video, so they can be randomly selected as existing samplers. This method of selecting negative hard samples from the same video has been shown to be more appropriate for tracking datasets, as verified through experiments.

In the tracking process, feature vectors are extracted from the image information of the objects using the embedding network trained with the triplet loss. To compare the feature vectors of existing tracks with those of the objects in the current frame, the feature vectors of existing tracks were used as arithmetic means because the existing tracks consist of images extracted from multiple frames, some of which may have occlusions or unclear appearances. To take comprehensive information into account, the arithmetic mean of the feature vectors extracted from the images on the track was adopted. This approach can also be interpreted in terms of the characteristics of metric learning because the arithmetic mean of a particular class vector represents the information that best represents that class. The parameter $\bar{r}_i$ is computed as follows:

$$\bar{r}_i = \frac{1}{L} \sum_{k=1}^{L} r_i^k, \tag{5}$$

where L is the number of times a track has been detected and r_i^k is a feature vector extracted from the detected image.

Subsequently, we computed the detected results and L2-norm distance in the current frame. Figure 9 shows the process of computing an object's appearance distance as follows:

$$Triplet(i,j) = max(0, 1 - \| \bar{r}_i - r_j \|)_2. \tag{6}$$

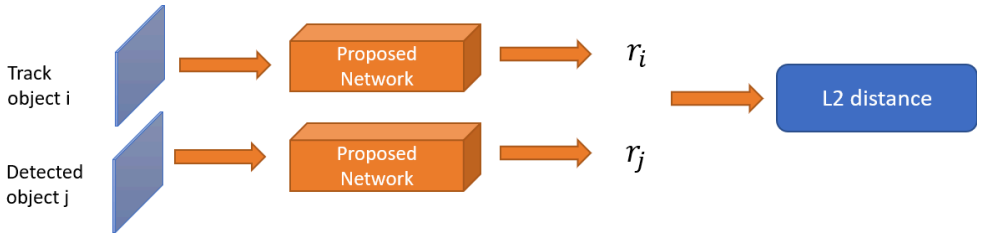


Figure 9. Appearance distance between objects.

4. Experimental Results

4.1. Datasets

To conduct an experiment on a road vehicle tracking dataset, we used the KITTI (OD and MOT) [26] and VeRi-776 [27] datasets for tracking objects on a road. The KITTI OD dataset was used for pretraining the detector (YOLOv3), which comprises 7481 stereo images. The VeRi-776 dataset was configured for vehicle re-identification, and it comprises images of vehicles taken from 20 videos recorded in a city. Each vehicle image has an ID, model, license plate, color, etc., as labels, and images of each vehicle were taken from various positions, namely the front, rear, and side. The dataset was used to configure the pretrained model of the embedding network.

KITTI Multi-Object Tracking Dataset

The KITTI MOT dataset was used for fine-tuning and testing the proposed tracking model and other existing methods. However, since the dataset comprises video frame images, it should be created with a new structure suitable for ID information. As shown in Figure 10, object images corresponding to each ID were cropped and stored by using the ground truth information in the frames. Object images were extracted from multiple frames on the basis of their IDs and configured to be one class. To evaluate the performance quantitatively, we employed the recall, precision, Frag, MOTP, ID switch, and MOTA from [26,28].

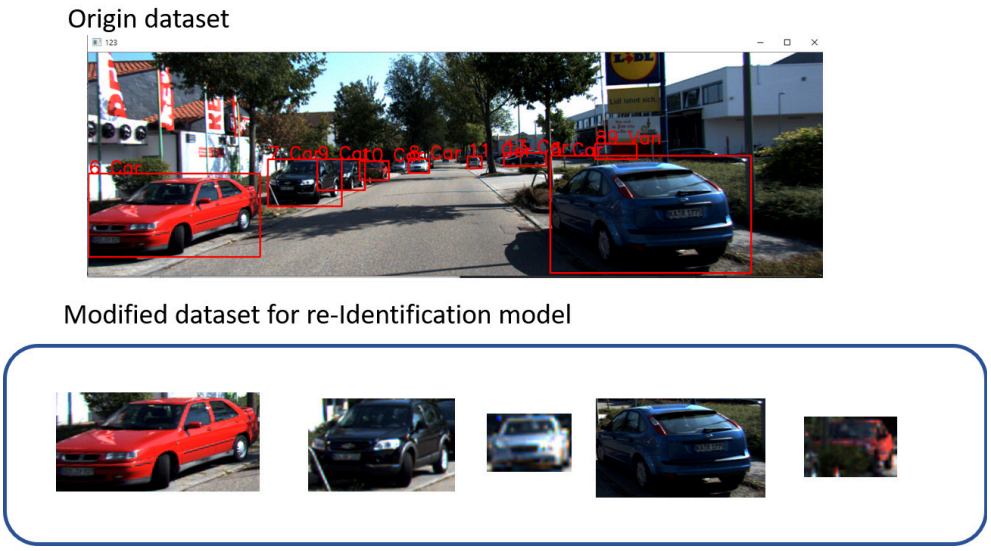


Figure 10. Generated ID dataset for KITTI multi-object tracking dataset.

4.2. Experimental Environments

The simulation results were conducted with a Core i5-9400F (up to 4.1 GHz), 32 GB DDR4, and GTX 1080Ti, and PyTorch was used as the framework. From the KITTI OD dataset, 80% of the samples were used for training and 20% of the samples for testing. The embedding network for re-identification was pretrained on the 37,930 images in the VeRi-776 training dataset. To train the embedding network, we used the Adam optimizer

with default settings ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). The learning rate was set to 0.01, and the scheduler was set to 0.9 after every two epochs. The model was trained for 100 epochs in the pretraining process. Fine-tuning was performed with the reconstructed KITTI MOT dataset. As mentioned, 17 images were used as the training set and 3 as the test set because the dataset does not provide labels for the test set. In Equation (5), λ was set to 0.1, and in Equation (4), α was set to 1.

4.3. Experimental Results and Analysis

4.3.1. Comparison with Other Models

This section presents a comparison between the proposed model and other models of DBT systems. The detector consists of those presented in each paper. Our proposed model showed the best performance on all metrics compared with the other algorithms as shown in Table 1. Furthermore, compared with the configuration of DeepSORT, it has a similar shape and speed while still showing better performance. The addition of the attention module and proven methods such as fine-tuning improved the performance. We also achieved positive results for our novel attempts, such as adopting the average value of the feature vectors and constructing a sampler.

Table 1. Comparison of the proposed method with other models. The arrows indicate better performance for each metric. (i.e., down arrow—lower is better).

Detector	Algorithm	Recall↑	Precision↑	Frag↓	MOTP↑	IDs↓	MOTA↓
FcRCNN	SORT [8]	0.67	0.58	81	0.53	151	0.58
YOLOv3	GoogLeNet + FC [20]	0.88	0.83	33	0.65	46	0.72
FcRCNN	DeepSORT [9]	0.75	0.67	42	0.58	51	0.66
YOLOv3	Triplet (ours)	0.9	0.85	25	0.68	28	0.73

4.3.2. Effectiveness of the Attention Module

This section compares the model size and number of parameters when the attention module was added. We added the attention module to the rear end of the residual blocks. As mentioned, the number of parameters in the attention module was not large. Table 2 shows that there was no significant difference compared to the existing model.

Table 2. Comparison of additional computational cost of attention blocks.

	Proposed Network	Without Attention Module
Total No. of parameters	13,562,160	13,549,856
Input size (MB)	0.57	0.57
Forward/backward size (MB)	179.66	146.24
Param size (MB)	51.74	51.69
Estimated total size (MB)	231.97	198.50

The table shows a slight difference in the forward/backward size and estimated total size, but the total number of parameters was almost the same.

4.3.3. Effectiveness of Backbone Network with the Proposed Method

This section details the performance comparison of structural changes in the embedding network. The detector was YOLOv3. The experimental results are presented in Table 3.

Table 3. Experimental results for embedding networks. The arrows indicate better performance for each metric. (i.e., down arrow—lower is better).

Detector	Embedding Network	Recall↑	Precision↑	Frag↓	MOTP↑	IDs↓	MOTA↓
YOLOv3	SORT	0.87	0.84	36	0.66	71	0.69
	GoogLeNet + FC	0.9	0.83	33	0.65	46	0.72
	Triplet w/o att (ours)	0.89	0.85	27	0.66	31	0.71
	Triplet (ours)	0.9	0.85	25	0.68	28	0.73

We experimented with a network consisting of 128 features as the outputs by adding FCL to GoogLeNet. Table 3 shows the performance of the proposed network with and without the attention module. The experimental results showed that the proposed model with the attention module was the most-effective model.

4.3.4. Effectiveness of Fine-Tuning

This section discusses the performance improvements achieved when pretrained models were configured using the VeRi-776 dataset and then fine-tuned. Table 4 shows the differences between models that conducted training only with the KITTI MOT dataset and models for which pretrained models were constructed and fine-tuning was performed. The experimental results also indicated that fine-tuned models have better performance.

Table 4. Results of fine tuning. The arrows indicate better performance for each metric. (i.e., down arrow—lower is better).

Model	Option	Recall↑	Precision↑	Frag↓	MOTP↑	IDs↓	MOTA↓
Proposed model	-	0.89	0.85	27	0.66	31	0.71
Proposed model	Fine-tuning	0.9	0.85	25	0.68	28	0.73

5. Conclusions

MOT is one of the most-popular topics in computer vision. With the development of deep learning, research on computer vision has intensively evolved. We aimed to improve the performance by focusing on representative solutions to the MOT problem.

In this paper, we proposed an effective utilization of image information in an online tracking method for real-time tracking. A vehicle re-ID dataset was used to track vehicles on roads. Validated methods such as fine-tuning were applied to distinguish objects with similar appearances. We also proposed the use of an appropriate sampler and mean vectors to address the occlusion problem arising from DBT and showed that they can help with reliably distinguishing between similar backgrounds. Furthermore, in the construction of the embedding network for re-identification, we proposed the use of a network with attention modules to enhance the performance while preserving the existing computational speed. By using an attention module, we achieved high performance while maintaining the size of the network model. We also demonstrated that our tracking system can show significantly improved tracking metrics compared with representative tracking systems.

One of the factors that contributed to the significance of our results was the use of proven learning performance improvement methods. The use of fine-tuning, online batch samplers, and attention modules improved the performance of the embedding network. This showed that the performance improvement measures devised by various CNN-related studies can help improve the tracking performance.

The results of DBT studies are proportional to the performance improvement achieved for detectors and embedding networks. Now that computer vision and CNNs are actively evolving with hardware performance, there is a clear possibility that multi-object tracking research can evolve. Furthermore, even if new deep learning techniques are developed, they can be easily applied to tracking systems. The scaling of deep learning algorithms is

limited by hardware limitations. Current research trends focus on optimization methods for limited environments. Obviously, simply scaling the size of deep learning networks and datasets has the potential to produce good results. However, in this paper, we showed that performance can be improved even in constrained environments with well-known optimization methods and approaches. Advances in computer vision will continue to improve the performance of tracking systems, and tracking systems will evolve through various attempts.

Author Contributions: Conceptualization, W.-J.A., K.-S.K., M.-T.L. and T.-K.K.; methodology, W.-J.A., K.-S.K., M.-T.L. and T.-K.K.; software, W.-J.A., K.-S.K. and T.-K.K.; validation, M.-T.L. and T.-K.K.; formal analysis, W.-J.A., K.-S.K. and T.-K.K.; investigation, M.-T.L. and T.-K.K.; resources, D.-S.P. and T.-K.K.; data curation, W.-J.A., K.-S.K. and T.-K.K.; writing—original draft preparation, K.-S.K. and T.-K.K.; writing—review and editing, W.-J.A., M.-T.L. and T.-K.K.; visualization, D.-S.P. and T.-K.K.; supervision, M.-T.L. and T.-K.K.; project administration, M.-T.L. and T.-K.K.; funding acquisition, M.-T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) (Grant No. 2022R1F1A1073543).

Data Availability Statement: The KITTI and VeRi-776 datasets presented in this study are openly available in the KITTI paper [26] and VeRi-776 paper [27], respectively.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9.
2. Rangasamy, K.; As'ari, M.A.; Rahmad, N.A.; Ghazali, N.F.; Ismail, S. Deep learning in sport video analysis: A review. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2020**, *18*, 1926–1933. [CrossRef]
3. Sreenu, G.; Durai, S. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. *J. Big Data* **2019**, *6*, 1–27. [CrossRef]
4. Chen, Y.; Xie, X.; Yu, B.; Li, Y.; Lin, K. Multitarget vehicle tracking and motion state estimation using a novel driving environment perception system of intelligent vehicles. *J. Adv. Transp.* **2021**, *2021*, 1–16. [CrossRef]
5. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
6. Ciaparrone, G.; Sánchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [CrossRef]
7. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
8. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
9. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
10. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, pp. 1–11.
11. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
12. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A survey of deep learning-based object detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
13. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37, Part I 14.
18. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
19. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
20. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
21. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
22. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
25. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
26. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
27. Lou, Y.; Bai, Y.; Liu, J.; Wang, S.; Duan, L. Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3235–3243.
28. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 1–10. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.