Re-Identification for Multi-Object Tracking Using Triplet Loss

Koung-Suk Ko[1], Woo-Jin Ahn[1], Geon-Hee Kim[1], Myo-Taeg Lim[1*], Tae-Koo Kang[2*], Dong-Sung Pae[3*]
[1]*Department of Electrical Engineering, Korea University, Seoul, Republic of Korea*
[2]*Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan, Republic of Korea*
[3]*Department of Software, Sangmyung University, Cheonan, Republic of Korea*
[*] *Corresponding author*
gaengko,mlim@korea.ac.kr

***Abstract*—Assigning a consistent identification(ID) number is a chronic problem in the tracking model. However, recent tracking models lose the ID because it focuses only on the previous frame. This paper constructed a tracking deep learning model using triplet loss to give consistent ID to objects detected while tracking. We also show the best way for pre-processing the input for the triplet-tracking model, which inputs various image sizes. The experimental result of 97.76% accuracy on KITTI shows the effectiveness of our result.**

***Index Terms*—Metric Learning, Re-Identification, Triplet Loss, Multi-Object Tracking**

I. Introduction

Autonomous driving system has been a hot potato recently. Especially, assigning consistent IDs in Multi-Object Tracking has been an important issue in crowded situations to track the same object. With the rise of the deep learning [1] method, tracking [2], [3] performance on road showed a novel improvement. However, recent tracking models [4] assigns a new ID when the same object is briefly obscured or disappeared out of the camera. To solve these problems, we proposed the Re-Identification method for tracking. The recent object tracking model showed an outstanding performance on tracking multi-objects. However, this method has a limitation on losing the ID while tracking. When the object is occluded on the road situation, it losses its ID because the tracking model only connects the tracking objects with the previous frame, which cannot be seen due to the occlusion. The ID loss problem can be solved with the objective of the re-identification research, which is mainly to detect the same person or object in different view cameras such as CCTV [5], [6]. However, most of the re-identification problem has a similar image size for the network because most of the tasks are base on the top-view image. In this paper, we propose a metric learning technique using re-identification triplet loss [7] on the tracking task. We find an object image with the same ID in the current frame from an object image in several previous frames. We also show the stabilized metric learning method [8] and effective pre-processing method for triplet on various sizes of images. In Section 2, we introduce the overall framework of our method. Section 3 introduces the KITTI(Karlsruhe Institute of Technology and Toyota Technological Institute) dataset, experimental settings, and provide experiment outcomes. Section 4 summarizes and concludes the paper.

II. Proposed Method

A. Network architecture

The proposed network for object detection using the camera consists of feature extraction and classification parts as fig. 1. The CNN(Convolution Neural Network) is capable of these feature representations using large amounts of data. A CNN learns hierarchical features through stacked convolution layers. A metric learning trains CNN to have similar embedding spaces for data in the same class. We used pre-trained GoogleNet [9] on ImageNet [10] dataset an embedding network to extract the feature vectors of the inputs. In contrast to the typical CNN, GoogleNet uses the Inception module on each layer to form a more diverse set of feature maps. It also uses the GAP(Global Average Pooling) instead of using the FC(Fully Connected) layer at the classification part. While FC uses a massive number of parameters, GAP has the advantage of not using weights.

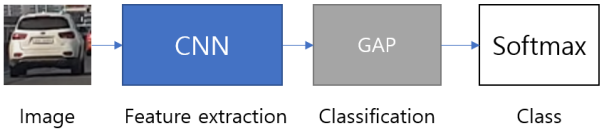

Fig. 1. Structure of object detection using CNN.

B. Feature extraction

The architecture of the network using triplet loss is shown in fig. 2. The network has three inputs. One is an anchor image, a reference image, another is a positive image belonging to the same ID as the anchor, and the other is a negative image belonging to different IDs as the anchor. The network extracts the feature vector from the inputs with the share weights to determine whether the between images are the same object or not.

To train the network, we introduce metric learning. This method calculates the relationships between data as distances with the deep learning-based network. In other words, if the Euclidean distance between the feature vector of the two data

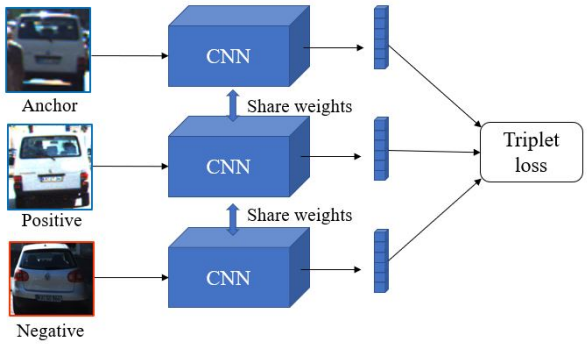

Fig. 2. Schematic diagram of triplet network structure.

is close, it is likely to be the same class (or ID), and if it is far, it is likely to be another class. As shown in fig. 3, the distance between data of the same ID is minimized as learning progresses. Also, data from different IDs be separated from each other. The triplet loss is designed to minimize the distances between an anchor and positive while to maximize the distance between an anchor and negative using the feature vector obtained from the CNN. The distance between feature vectors is calculated as L2-norm, as in eq 1.

$$D_{x_1,x_2} = \| f(x_1) - f(x_2) \|_2 \tag{1}$$

where x_1, x_2 are the input images which are an anchor and positive or negative. f implies the embedding network. Triplet loss is defined as follows:

$$l_{triplet}(a,p,n) = [D_{a,p} - D_{a,n} + \alpha]_+ \tag{2}$$

where a, p, n are anchor, positive, negative images and α means the margin between distances. The network is trained to minimize the triplet loss as shown in fig. 3

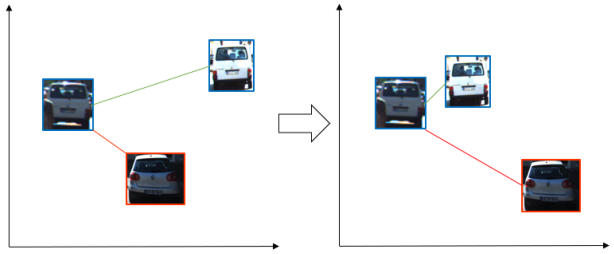

Fig. 3. Result of metric learning.

C. Dataset sampling

The method of selecting the input samples of a triplet is critical and significantly influences the learning result. Triple loss has no labels other than anchor, positive, and negative relationships. Therefore, the triplet loss does not converge stably. So when selecting positive and negative, it is necessary to select a problematic condition to judge [8]. A well-known method is to select hard negative, but we have constructed the method of selecting positive and negative appropriately in the tracking dataset. Most tracking models find objects with the same ID as the currently detected objects in the previous frame. So when we selected the negative, we selected from the same video as the anchor image. Also, we selected the negative priority from the nearby frame of the anchor image. In the case of positive, the probability of being in the same video is high, so it was selected randomly.

The method of making the input image also influences the learning outcome. The object images captured from the video have different sizes. The embedding network needs fixed size inputs, so the size of the images must be unified. So we tried three methods to check the influences of each (fig. 4). The first method is to resize the image with bilinear (fig. 4 (a)). The second method is to fill up the insufficient space in the captured image with black (fig. 4 (b)). Since continuously detected images may have similar sizes, we applied a unified background. For the same reason, the third method filled the background of the image with RGB's mean value, as attempted in [11] (fig. 4 (c)).

(a) Resize (b) Black Background (c) RGB Mean Background

Fig. 4. Results of making the input image using three methods. (a) Resized the image(resize) with bilinear method (b) Filled the background of the image with black(BB). (c) Filled the background of the image with each mean of RGB(MB).

III. Experiments

A. Dataset configuration

We used KITTI MOT(Multi-Object Tracking)-dataset [12] for the experiment. The dataset consists of 20 videos and has the object's bounding box and ID information in ground truth for each frame. Based on this information, all of the object images were cropped from the video.

B. Experimental setting

We train the model for 100 epoch applying the Adam optimizer [13] with default settings ($\epsilon = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$).The learning rate is decreased by a factor of 0.8 after each 10^{th} epoch. The margin α is set to 1 and each batch size is set to 128.

C. Experimental result

The experiment was conducted in three videos. We assumed that the ground-truth label image in each frame was detected as a result of the detection. The distances between the object image of the current frame and all the object images in the

previous five frames were calculated, respectively. The accuracy was measured by checking that the minimum distance image's ID matches the ID of the current image. According

TABLE I
ACCURACY OF EACH SAMPLING METHODS

Video Num	Accuracy		
	Resize	***BB***	***MB***
Video 1	96.828	77.729	87.824
Video 2	96.56	74.351	88.36
Video 3	97.759	82.549	89.076

to the results of the experiment, resizing the image shows the best performance. In shown fig. 5, the BB(Black Background) method, it does not work properly if the input image is dark or if there are many black objects in comparison. The MB(Mean Background) method performs better than the BB method but does not perform best.

Fig. 5. Incorrect test result sample.

IV. CONCLUSION

Our experiments show a relatively high level of accuracy in the MOT dataset without considering the data's coordinates. The experimental result of 97.76% accuracy on KITTI shows the effectiveness of our result. Also, in making the input image, we found that the resize method had the best performance. It shows enough to make up for the deficiencies of the tracking model.

ACKNOWLEDGMENT

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea NRF) funded by the Ministry of Education (No. 2019R1A2C108974211)

REFERENCES

[1] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Proc. Neural Information and Processing Systems, 2012.
[2] Y. Fan et al., "ReMOTS: Self-Supervised Refining Multi-Object Tracking and Segmentation ",eprint arXiv:2007.03200, 2020
[3] J. Pang, L. Qiu, H. Chen, Q. Li, T. Darrell, F. Yu, "Quasi-Dense Instance Similarity Learning", arXiv:2006.06664, 2020
[4] X. Zhou, V. Koltun, P. Krähenbühl,"Tracking Objects as Points", arXiv:2004.01177v2, 2020
[5] Ratnesh Kumar, Edwin Weill, Farzin Aghdasi,and Parthsarathy Sriram, "Vehicle re-identification: an efficient baseline using triplet embedding", arXiv preprint arXiv:1901.01015, 2019.
[6] Neeti Narayan, Nishant Sankaran, Srirangaraj Setlur and Venu Govindaraju,"Re-identification for online person tracking by modeling space-time continuum", Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018), pp. 1438-1447.
[7] H. Wang, J. Hou, and N. Chen, "A survey of vehicle re-identification based on deep learning", IEEE Access, vol. 7, pp. 172443-172469, 2019.
[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering", in CVPR, pp. 815-823, 2015.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 248-255, 2009.
[10] C. Szegedy et al., "Going deeper with convolutions", 2014
[11] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking", 2016, [online] Available: https://arxiv.org/abs/1606.09549.
[12] A. Geiger, P. Lenz and R. Urtasun,"Are we ready for autonomous driving? the KITTI vision benchmark suite", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3354-3361, 2012.
[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", 2014.