J Inf Process Syst, Vol.15, No.4, pp.1017~1028, August 2019
https://doi.org/10.3745/JIPS.04.0132

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Tracking-by-Detection System for Pedestrian Tracking Using Deep Learning Technique and Color Information

Mai Thanh Nhat Truong* and Sanghoon Kim*

Abstract

Pedestrian tracking is a particular object tracking problem and an important component in various vision-based applications, such as autonomous cars and surveillance systems. Following several years of development, pedestrian tracking in videos remains challenging, owing to the diversity of object appearances and surrounding environments. In this research, we proposed a tracking-by-detection system for pedestrian tracking, which incorporates a convolutional neural network (CNN) and color information. Pedestrians in video frames are localized using a CNN-based algorithm, and then detected pedestrians are assigned to their corresponding tracklets based on similarities between color distributions. The experimental results show that our system is able to overcome various difficulties to produce highly accurate tracking results.

Keywords

Color Distribution, Convolutional Neural Network, Pedestrian Tracking, Tracking-by-Detection

1. Introduction

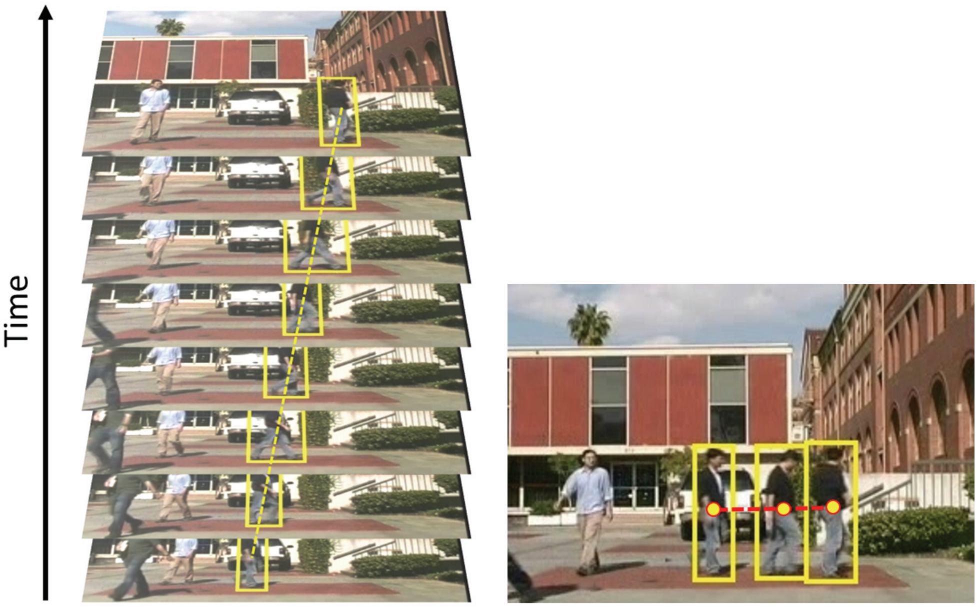
Pedestrian tracking plays a crucial role in various machine vision systems, such as traffic surveillance [1], security systems [2], and, recently, autonomous cars [3]. In general, the goal of a pedestrian tracking algorithm is to locate people walking in video data captured by cameras, and then produce records of the trajectories of the pedestrians, which are called tracklets. So far, pedestrian tracking has remained a highly complex problem. Pedestrian tracking can be more difficult than tracking other objects, because pedestrians are non-rigid objects and their appearances are heterogeneous. Moreover, occlusions frequently occur when pedestrians walk past each other. Other difficulties can include lighting conditions, camera motions, and sudden changes in movements of pedestrians. Designing an algorithm to simultaneously resolve all the aforementioned difficulties is challenging; hence, object tracking algorithms are usually designed to track specific objects under certain conditions [4].

Object tracking approaches can be divided into three categories: point tracking, kernel tracking, and silhouette tracking [5]. Tracking-by-detection is classified as a point-tracking approach. In each video frame, objects are localized by an external mechanism. Features of the detected objects are represented by points. Then, the tracking task is performed using location and motion information for the points in both the current and previous frames.

Manuscript received February 1, 2019; accepted March 8, 2019.

Corresponding Author: Sanghoon Kim (kimsh@hknu.ac.kr)
* Dept. of Electrical, Electronic, and Control Engineering, Hankyong National University, Anseong, Korea ({mtntruong, kimsh}@hknu.ac.kr)

Fig. 1 illustrates how the tracking-by-detection approach works. In each frame, an object detection algorithm localizes the desired objects (Fig. 1(a)). Then, the locations of a target object are associated together into tracklets, or records of trajectories (Fig. 1(b)). Tracking-by-detection is a robust solution for multi-target tracking, because besides tracking it also rectifies two frequently encountered oversights. The first is that several researchers suppose the locations of the targets are given in the first frame of a video. This assumption limits the usability of object tracking algorithms in real-life applications, because most of the time the target locations in the first video frame are not given, or the desired objects do not appear in the first few video frames. The second oversight is that small tracking errors rapidly accumulate, degrading the tracking performance for objects in long videos. Tracking-by-detection can overcome this problem by localizing the target objects in every video frame.



(a) (b)

Fig. 1. Illustration of the tracking-by-detection approach: (a) object detection algorithm localizes desired objects in every frame and (b) locations of an object (yellow dots) in each frame are associated to create a tracklet (red line).

Several pedestrian trackers using this approach have recently been proposed. An algorithm from Dehghan et al. [6] detected each person using a part-based human detector. Then, a global data association method based on generalized graphs was employed to track each individual in the whole video. Zhu et al. [7] combined a single object tracking algorithm and a data association method to create a unified framework. The algorithm is able to deal with noisy detections and interactions between targets. For data association, they proposed a new algorithm called dual matching attention networks. This technique utilized both spatial and temporal attention mechanisms. Jiang et al. [8] proposed a system comprising a two-stage re-identification algorithm, dealing with the cases of track drift and re-entry into the field of view individually. They utilized this mechanism to match the identities of lost and reappearing targets and update the statuses of re-identified targets, through comparing the affinities of their appearances, sizes, and positions.

The general disadvantage of the abovementioned systems is that these pedestrian detectors are not reliable, because they were constructed using hand-crafted features. Moreover, the tracklet association components are complex, increasing the execution times. In this research, we propose a tracking-by-detection system for tracking pedestrians. Aiming to resolve the two abovementioned issues, we designed an algorithm that achieves a high tracking accuracy within a reasonable processing time. For pedestrian detection, we utilized a deep learning technique to achieve the highest accuracy. Recently, deep learning has been widely utilized in several fields, such as medical image analysis [9], image fusion [10], text analysis [11], and sound analysis [12].

Our system incorporates two main components. The first is a pedestrian detector, which combines Faster R-CNN and ResNet-101, two CNN-based algorithms. The locations of pedestrians detected by the first component are then fed into the second component in which spatial overlap properties and color information are used to sort the detected pedestrians into their corresponding tracklets. The details of our tracking system are described in Section 2. The experimental results are presented in Section 3. Finally, our research is concluded in Section 4.

2. Proposed Method

Our tracking system incorporates two main components: pedestrian detection and tracklet association. The pedestrian detector represents a combination of Faster R-CNN and ResNet-101, two CNN-based algorithms. This component processes video frames and returns the locations of pedestrians in the form of bounding boxes. To associate detected pedestrians with their corresponding tracklets, we exploit spatial overlap properties and color information to achieve a high performance while maintaining a low computational cost. The details of each component are presented in the next two subsections.

2.1 Pedestrian Detection

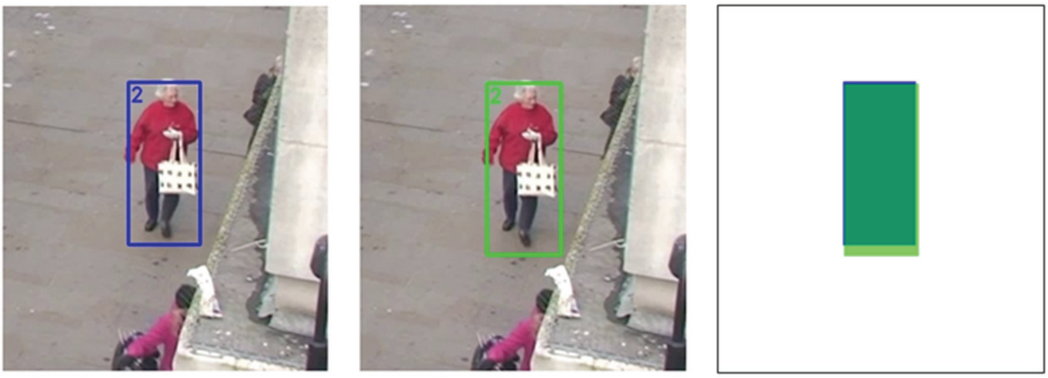
For pedestrian detection in each video frame, we combine Faster R-CNN [13] and ResNet-101 [14]. ResNet-101 is a feature extractor that provides high-level features for Faster R-CNN, an object detector. The combination of Faster R-CNN and ResNet-101 allows the pedestrian detector to produce highly accurate results within a reasonable execution time. Several CNN-based object detectors and feature extractors have been proposed. However, algorithms with highest performance usually take a significant amount of time to process. Several experiments have demonstrated that the combination of Faster R-CNN and ResNet-101 produces the best trade-off between speed and accuracy [15], hence it is appropriate for real time tracking. Although the Faster R-CNN algorithm is able to detect several types of object at once, the processing time is high. In this research we only consider pedestrians as target objects to reduce the execution time.

Faster R-CNN is a two-stage object detector. In the first stage, a feature extractor is utilized to extract visual features from images. Region proposals are then predicted using features at some intermediate levels. In the next stage, these region proposals are used to crop features from the same intermediate feature map. For each proposal, the cropped features are then fed into the remainder of the feature extractor, for predicting their class and class-specific box refinement.

In our system, the feature extractor utilized by Faster R-CNN is ResNet-101. By using a special component called “residual block”, ResNet allows the construction of very deep networks, for high-performance feature extraction. In ResNet-101 and other very deep networks (ResNet-50/101/152), a special type of residual block called “bottleneck block” is employed. The purpose of this bottleneck design is to increase the computational efficiency. After performing object detection with Faster R-CNN, the locations of detected pedestrians are fed into the next component.

2.2 Tracklet Association

After acquiring the detection result from the first component, the second component assigns each detected pedestrian to its corresponding tracklet. Assuming that the video recording devices have performed without any interruption and there are missing frames, we employ spatial overlap properties [16] to determine the identity of a given pedestrian. The location of the bounding box of a pedestrian is compared with the locations of the other bounding boxes in the previous frame. Because there are no missing frames and the walking speed is slow, the bounding box of a pedestrian in the current video frame always overlaps with the bounding box of that pedestrian in the previous frame (Fig. 2). Hence, a given pedestrian will be assigned to the tracklet that overlaps their bounding box in the current frame.



(a) (b) (c)

Fig. 2. The bounding boxes of a pedestrian in two consecutive frames overlap each other: (a) bounding box in frame *t* – 1, (b) Bounding box in frame *t*, and (c) overlapping region.

If there are two or more bounding boxes that overlap with the region of the given pedestrian, i.e., there were collisions between pedestrians or several pedestrians walking together, then we employ color distributions to find the correct tracklets. The color distribution has been proven as an effective similarity measure in object tracking [17]. For each bounding box, we calculate three-dimensional (3D) histograms in an RGB space using 8×8×8 bins. The 3D histograms have the form of $P = (p_1, p_2, \dots, p_M)$ where p_i is the probability of occurrence of the color in i^{th} bin, and M is the total number of bins.

After calculating 3D histograms, we calculate the statistical distance between two bounding boxes using the Hellinger distance. Given the color distributions $P = (p_1, p_2, \dots, p_M)$ and $Q = (q_1, q_2, \dots, q_M)$ of two bounding boxes, first we calculate the Bhattacharya coefficient

$$B_C(P, Q) = \sum_{m=1}^{M} \sqrt{p_m q_m}. \tag{1}$$

Then the Hellinger distance is calculated as

$$H_D(P,Q) = \sqrt{1 - B_C(P,Q)}. \tag{2}$$

A pedestrian will be assigned to the tracklet with the lowest distance. Fig. 3 illustrates the construction of the 3D color histogram. The probability of occurrence is represented by size of the spheres, bigger sphere means higher probability.

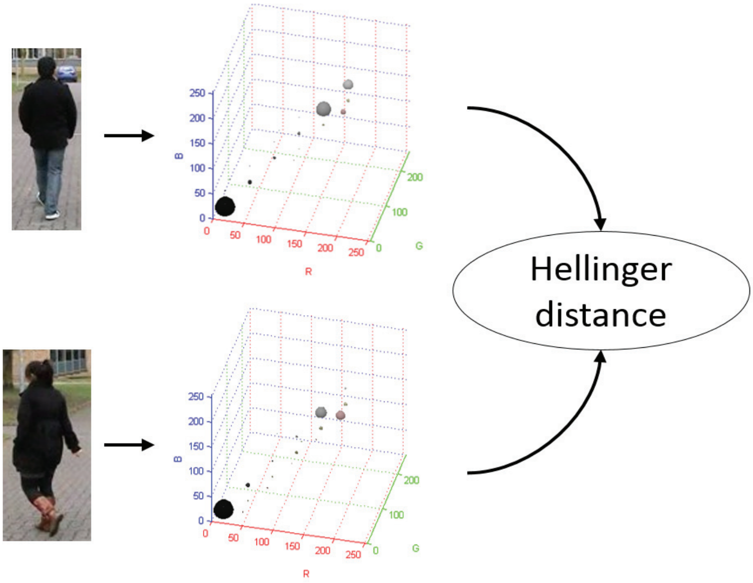


Fig. 3. Calculate the similarity between two color distributions using the Hellinger distance.

The flowchart in Fig. 4 summarizes our pedestrian tracking system. Given a video frame at time *t*, pedestrians are detected in this frame. Then, the locations of the pedestrians are compared with those in the video frame at time *t* – 1. Overlapping regions are utilized to determine the correct tracklets for the detected pedestrians. If more than two regions overlap with each other, then the distances between color histograms will be used.

3. Experimental Results

In this section, we demonstrate the performance of the proposed pedestrian tracking system. The algorithm was implemented in Python running on a Linux operating system. The implementation was deployed on a desktop computer, which was equipped with an Intel Core i7-8700K, 32 GB memory, and an NVIDIA Titan XP GPU. The OpenCV library was employed to process the video frames, and the TensorFlow library was used for the implementation of the CNN-based pedestrian detector. The testing videos were acquired from [18-20]. A preview of the testing videos is presented in Fig. 5.

Fig. 5(a) shows the testing video from [18], which is titled as “woman”. There is a single pedestrian in this video, and rather than being stationary, the camera follows the movement of the pedestrian. Several partial occlusions occur during the video as the pedestrian walks behind cars. We used this video to

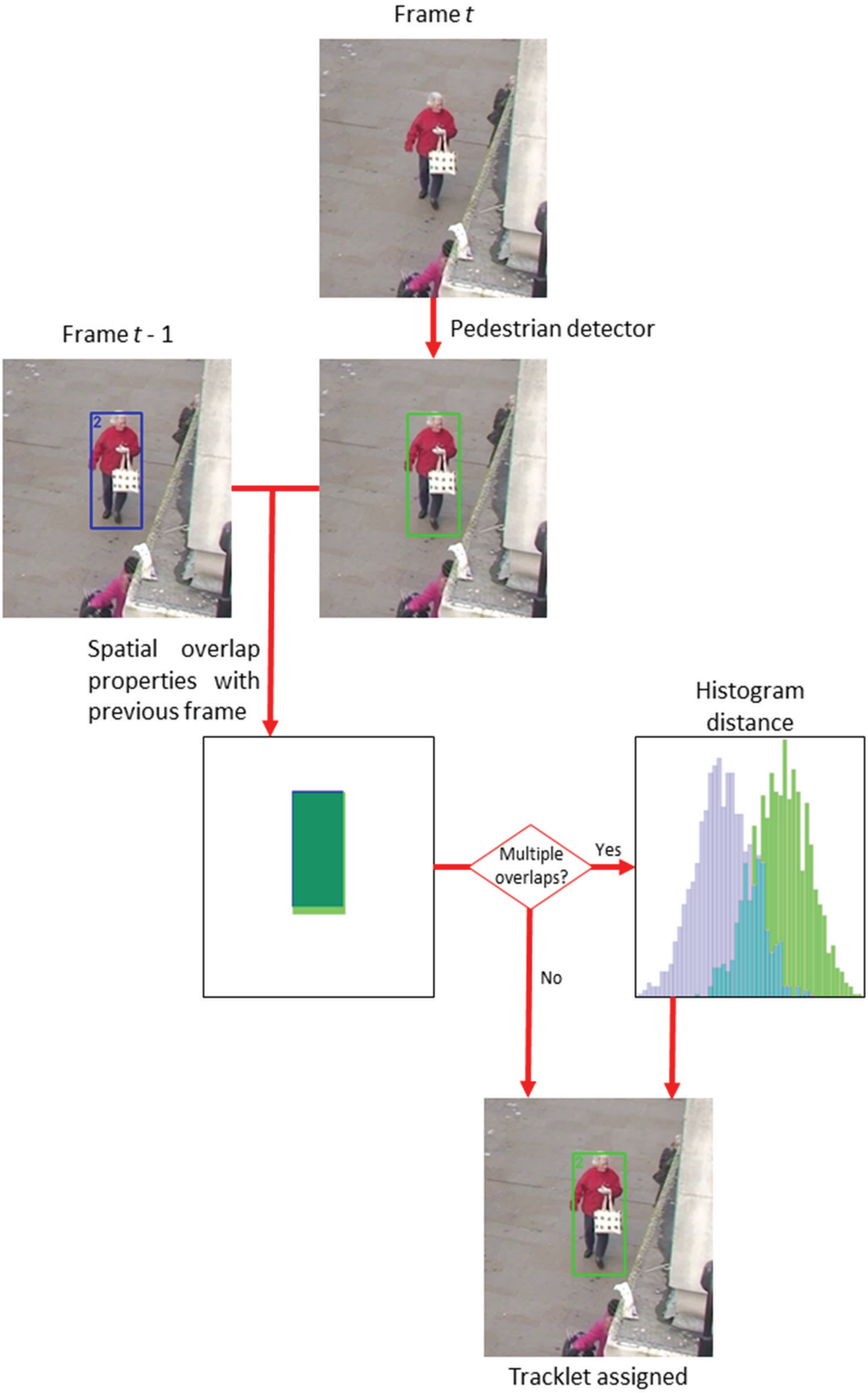
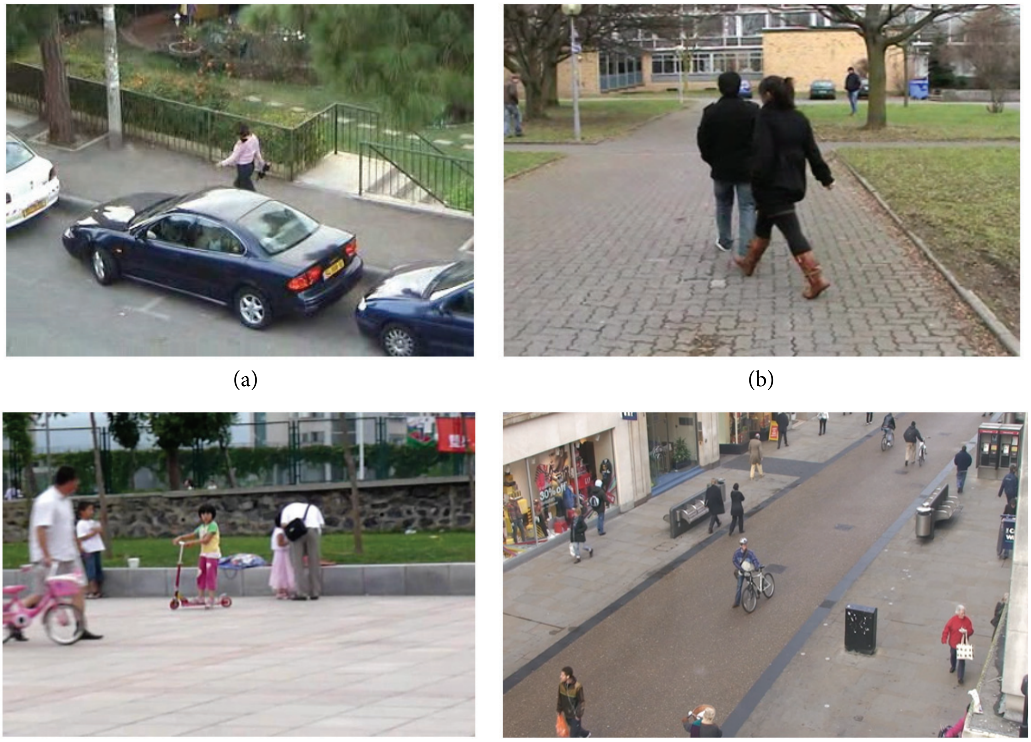


Fig. 4. Flowchart of proposed pedestrian tracking system.

evaluate the performance of our system for single-target tracking and dealing with partial occlusions. Fig. 5(b) and (c) show testing videos titled “person crossing” (from [18]) and “girl” (from [19]), respectively. In both videos, the camera follows the movement of a selected pedestrian. In “person crossing”, the camera follows the only person appearing in the first frame, while in “girl” the camera follows the girl riding a scooter. Several total and partial occlusions occur throughout the two videos, and in some cases pedestrians with similar appearances cross each other. Fig. 5(d) shows the video from [20], called “AVG-TownCentre”. The camera is stationary in this video and is placed in a crowded location. We employed these videos to evaluate the performance of the proposed algorithm for multi-target tracking.



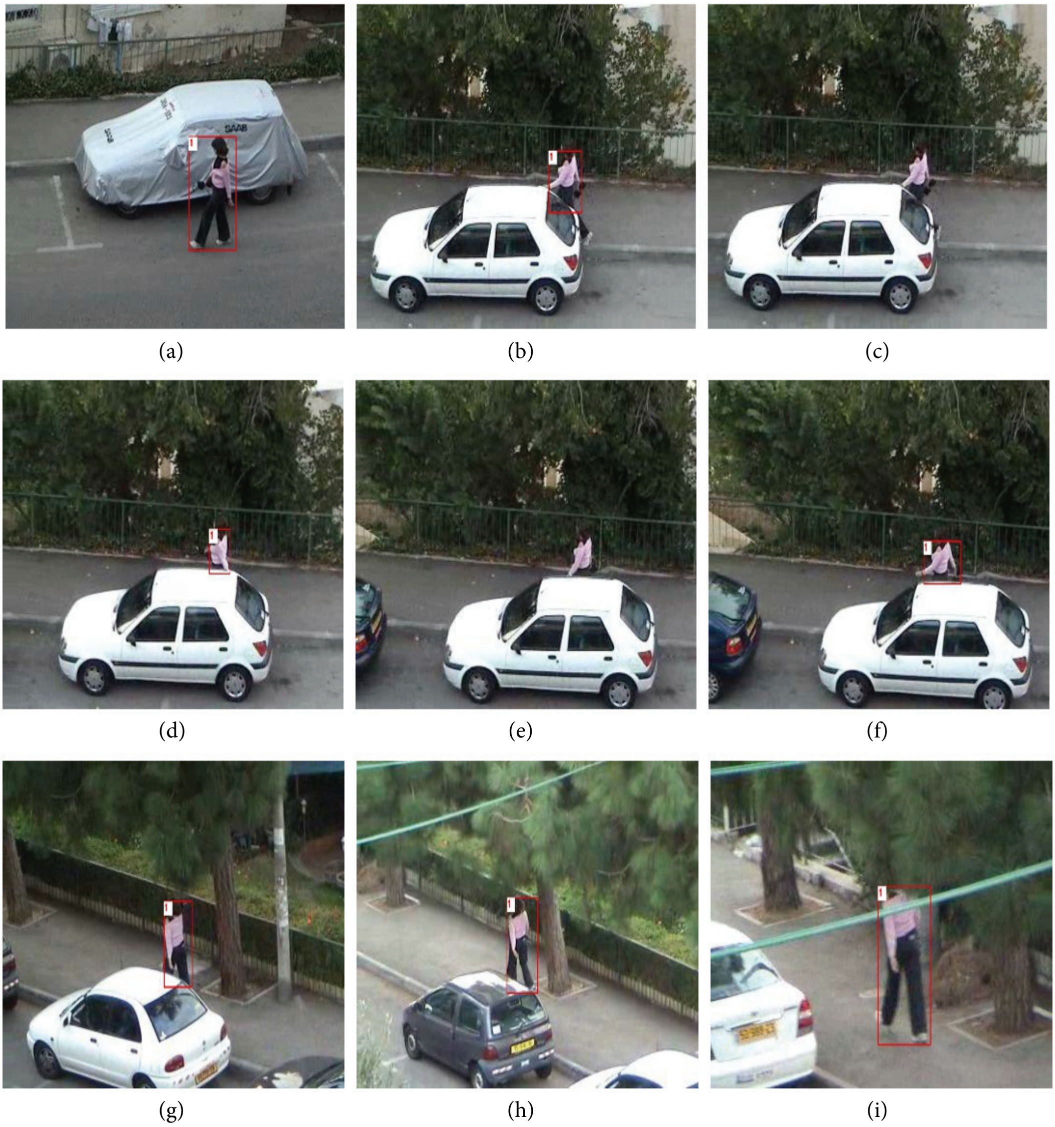
(a) (b)

(c) (d)

Fig. 5. Test video sequences: (a) “woman”, (b) “person crossing”, (c) “girl”, and (d) “AVG-TownCentre”.

3.1 Tracking Single Pedestrian

In this test, we utilized the video “woman” from [18] to evaluate the performance of our system for single-target tracking and dealing with partial occlusions. Fig. 6 depicts tracking results obtained by our system. The object detector successfully detected the pedestrian, and the tracklet was correctly assigned (with “1” as the identifying number) until frame 111 (Fig. 6(b)). In frame 112 (Fig. 6(c)), the object detector could not localize the pedestrian owing to a partial occlusion. However, a few frames later the object detector was able to again recognize and detect the pedestrian. Although failed detection occurred several times, owing to the tracklet assignment algorithm the pedestrian was still assigned to the correct tracklet (Fig. 6(d)–(f)). Our algorithm tracked the pedestrian accurately until the final frame of the video (Fig. 6(i)).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

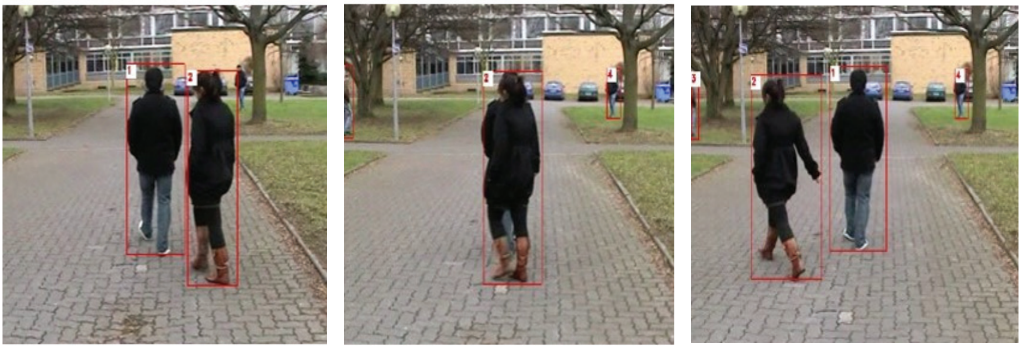
Fig. 6. Results of tracking single target: (a) frame 1, (b) frame 111, (c) frame 112, (d) frame 123, (e) frame 130, (f) frame 135, (g) frame 379, (h) frame 480, and (i) frame 596.

3.2 Tracking Multiple Pedestrians

To evaluate the performance of our system for tracking multiple pedestrians, we employed the videos shown in Fig. 5(b)–(d). First, we applied our algorithm to the video "person crossing", and the results are presented in Fig. 7. In this video, several total occlusions occurred. Fig. 7(a) shows two people who are about to cross each other. In Fig. 7(b), person number 2 completely covers person number 1, making the tracklet of person number 1 disappear. When the occlusion ended, person number 1 appeared again, and the tracklet was assigned correctly (Fig. 7(c)). It is worth noting that there is very little difference in appearance between person number 1 and 2. The use of color distributions showed its effectiveness in this case.

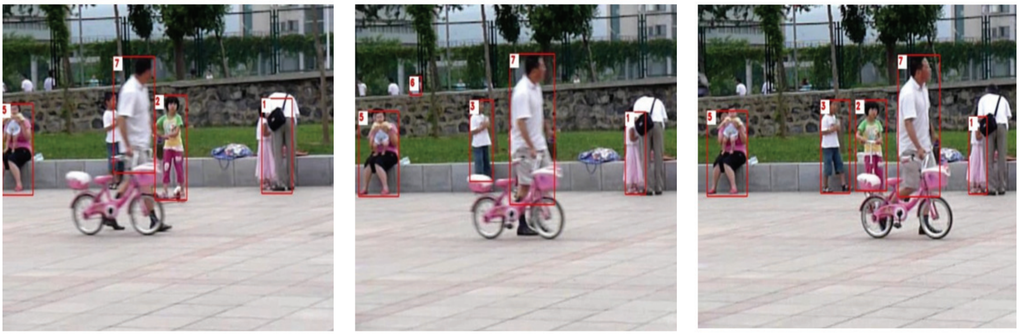
Similarly, several total occlusions occurred in the video “girl”. Fig. 8(a) shows that person number 7 is about to occlude girl number 2. In Fig. 8(b), person number 7 has fully occluded girl number 2, and the tracklet of the girl has disappeared. A few frames later in Fig. 8(c), girl number 2 appears again, and our algorithm successfully assigns the tracklets for all detected pedestrians.

The pedestrian tracking results for the video “AVG-TownCentre” are depicted in Fig. 9. Fig. 9(a) shows the first frame of the video. In this frame, person number 7 (top-left corner of the video frame) collides with another people. A few seconds later (Fig. 9(b)), person number 7 is still correctly tracked, owing to color information, and the other person in the collision has been assigned a new number. The other people have been also been tracked correctly.



(a) (b) (c)

Fig. 7. Results of tracking in the video “person crossing”: (a) before occlusion, (b) during occlusion, and (c) after occlusion.

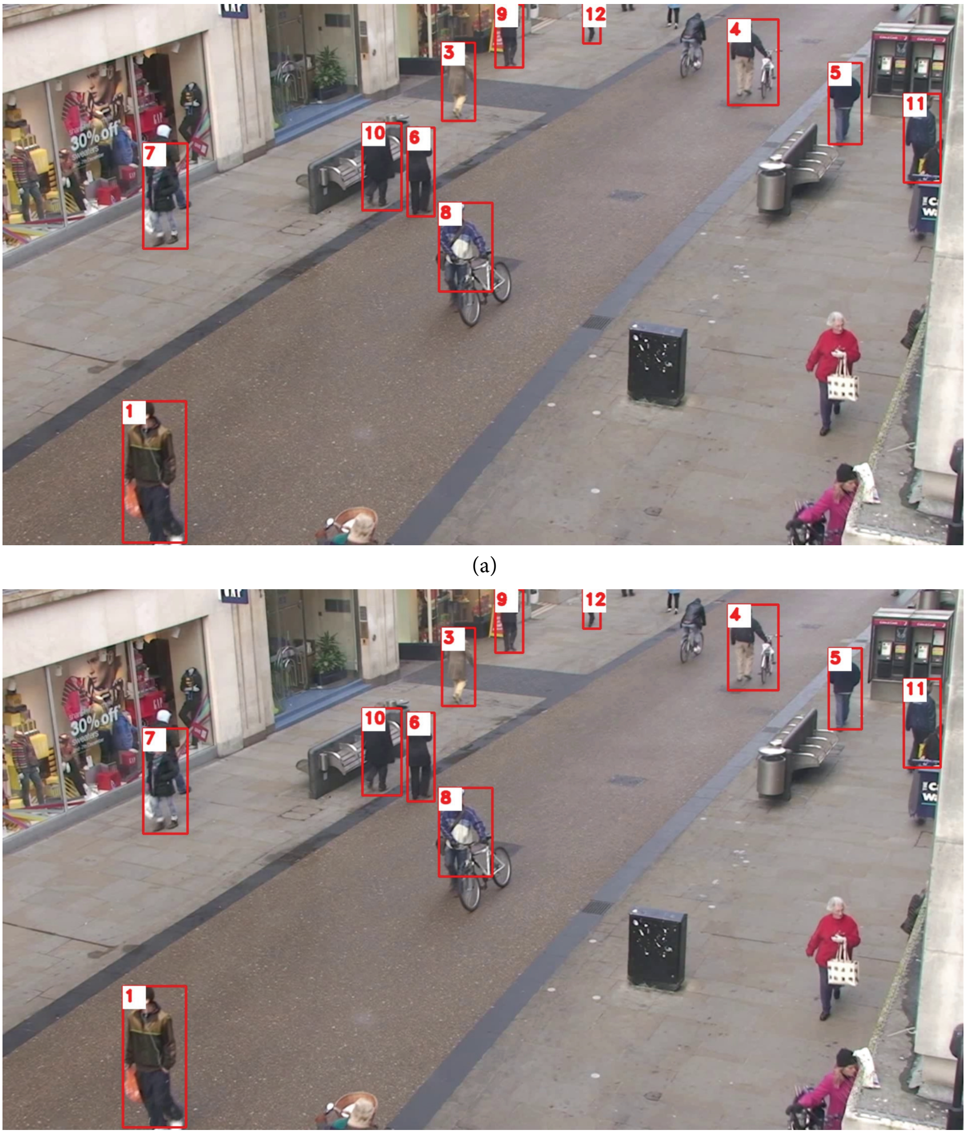


(a) (b) (c)

Fig. 8. Results of tracking in the video “girl”: (a) before occlusion, (b) during occlusion, and (c) after occlusion.

4. Conclusion

Pedestrian tracking is an important component in various vision-based applications, such as autonomous cars and surveillance systems. In this research, we proposed a tracking-by-detection system for tracking pedestrians. Our system incorporates two main components. The first is a pedestrian detector, which combines Faster R-CNN and ResNet-101, two CNN-based algorithms. The locations of



(a)

(b)

Fig. 9. Tracking multiple pedestrians on a crowded road: (a) tracking all pedestrians and (b) all tracklets were assigned correctly

pedestrians detected by the first component are then fed into the second component, where spatial overlap properties and color information are utilized to associate detected pedestrians with their corresponding tracklets. ResNet-101 is a feature extractor that provides high-level features for Faster R-CNN, an object detector. The combination of Faster R-CNN and ResNet-101 allows the pedestrian detector to produce accurate results with a short execution time. The spatial overlap properties and color information yield accurate tracklet assignments while maintaining a low computational cost. The experimental results proved the effectiveness of our proposed algorithm for pedestrian tracking. For further development, we will enhance the capability of our system by improving the accuracy of the object detector, along with proposing an improved model for tracklet assignment.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2015R1D1A1A01057518). We also gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] C. Sutheerakul, N. Kronprasert, M. Kaewmoracharoen, and P. Pichayapan, “Application of unmanned aerial vehicles to pedestrian traffic monitoring and management for shopping streets,” *Transportation Research Procedia*, vol. 25, pp. 1717-1734, 2017.

[2] K. V. Sriram and R. H. Havaldar, “Human detection and tracking in video surveillance system,” in *Proceedings of 2016 IEEE International Conference on Computational Intelligence and Computing Research*, Chennai, India, 2016, pp. 1-3.

[3] F. Li, R. Zhang, and F. You, “Fast pedestrian detection and dynamic tracking for intelligent vehicles within V2V cooperative environment,” *IET Image Processing*, vol. 11, no. 10, pp. 833-840, 2017.

[4] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: an experimental survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442-1468, 2014.

[5] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: a survey,” *ACM Computing Survey*, vol. 38, no. 4, article no. 13, 2006.

[6] A. Dehghan, H. Idrees, A. R. Zamir, and M. Shah, “Automatic detection and tracking of pedestrians in videos with various crowd densities,” in *Pedestrian and Evacuation Dynamics 2012*. Cham: Springer, 2014, pp. 3-19.

[7] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. H. Yang, “Online multi-object tracking with dual matching attention networks,” in *Computer Vision – ECCV 2018*. Cham: Springer, 2018, pp. 366-382.

[8] Y. Jiang, H. Shin, J. Ju, and H. Ko, “Online pedestrian tracking with multi-stage re-identification,” in *Proceedings of 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Lecce, Italy, 2017, pp. 1-6.

[9] N. Yu, Z. Yu, F. Gu, T. Li, X. Tian, and Y. Pan, “Deep learning in genomic and medical image data analysis: challenges and approaches,” *Journal of Information Processing Systems*, vol. 13, no. 2, pp. 204-214, 2017.

[10] X. Feng, “Infrared and visible image fusion based on NSCT and deep learning,” *Journal of Information Processing Systems*, vol. 14, no. 6, pp. 1405-1419, 2018.

[11] Z. Jiang, S. Gao, and W. Dai, “A CTR prediction approach for text advertising based on the SAE-LR deep neural network,” *Journal of Information Processing Systems*, vol. 13, no. 5, pp. 1052-1070, 2017.

[12] S. D. You, C. Liu, and W. Chen, “Comparative study of singing voice detection based on deep neural networks and ensemble learning,” *Human-centric Computing and Information Sciences*, vol. 8, no. 1, article no. 34, 2018.

[13] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 770-778.

[15] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, pp. 3296-3297.

[16] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proceedings of 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Lecce, Italy, 2017, pp. 1-6.

[17] M. T. N. Truong and S. Kim, "Parallel implementation of color-based particle filter for object tracking in embedded systems," *Human-centric Computing and Information Sciences*, vol. 7, article no. 2, 2017.

[18] T. Vojir, "Tracking Dataset," 2017 [Online]. Available: http://cmp.felk.cvut.cz/~vojirtom/dataset/.

[19] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, et al., "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 3-53.

[20] L. Leal-Taixe, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: towards a benchmark for multi-target tracking," 2015 [Online]. Available: https://arxiv.org/abs/1504.01942.

Mai Thanh Nhat Truong https://orcid.org/0000-0002-6448-7837

He received his B.Sc. degree in Mathematics and Computer Science from Ho Chi Minh City University of Science, Vietnam in 2014 and M.Sc. degree in Electrical, Electronic, and Control Engineering from Hankyong National University, Korea in 2017. Since September 2017, he has been with the Department of Electrical, Electronic, and Control Engineering in Hankyong National University, Korea as a PhD candidate. His research interests are machine vision and image analysis.

Sanghoon Kim https://orcid.org/0000-0001-5351-8215

He received B.Sc., M.Sc., and Ph.D. degrees in Electronic Engineering from Korea University, Seoul, in 1987, 1989, and 1999, respectively. From 1989 to 1994, he was a Research Engineer with LG Semiconductor Company, where he was engaged in research and development of PC chipset design. From January 2004 to January 2005, he was a Visiting Scholar with the University of Maryland, College Park, MD, USA. Since September 1999, he has been with Hankyong National University, Anseong, Korea, where he is currently a Professor. His current research interests are image processing, object detection, and robot vision. Prof. Kim is a member of the IEEE and Korean Information Processing Society.