Deep Learning based Pedestrian Detection and Tracking System using Unmanned Aerial Vehicle and Prediction Method

Jae Hee Lee, Chang Jin Seo

Abstract*: *Background/Objectives: Pedestrian detection and tracking system have become an essential field in the object detection and target tracking research area. This study proposes for developing and implementing the fast pedestrian detection and tracking system using Deep learning (YOLOv3), UAV (Unmanned Aerial Vehicle) and prediction method that is the Kalman Filter. Methods/Statistical analysis: The performance of the object detection and tracking system is decided by the performance time and the accuracy of object detection and tracking algorithms. So we applied to the YOLOv3 which is the fast detection method recent at our proposed system and also proposed the Kalman Filter algorithm with a variable detection area as the pedestrian tracking system. Findings: In the experiments, the proposed method successfully detected and tracked pedestrians who move at 53 FPS maximum and 38-43 FPS on average each test videos. The proposed way with variable search ranges made much fewer errors than the traditional object detector with fixed search ranges. Improvements/Applications: This research result can be applied pedestrian moving monitoring system, ITS (intelligent transport system) and security surveillance system.

Index Terms*: *Pedestrian Tracking, Deep Learning, Object Detection, YOLOv3, Kalman Filter

I. INTRODUCTION

In recent years, various intelligent systems employing deep learning techniques have been developed. Until some years ago, the use of deep neural networks in smart applications was hindered by several shortcomings, such as poor recognition rates, slow processing of large datasets, and difficulties in the learning of multilayer networks. However, rapid progress in GPGPU (general-purpose graphics processing units) and the advancement of deep learning architectures have reignited researchers' interest in deep learning applications over the last ten years [1]. Nowadays, deep learning is applied to practical applications in many areas, including intelligent transportation systems, autonomous cars, unmanned aerial vehicles, robots, and artificial intelligence agents [2]. Pedestrian detection and tracking via a low-altitude unmanned aerial vehicle (UAV) is a critical component for building ITS (intelligent transportation system). ITS is a next-generation transportation and traffic management application that exploits advanced technologies in Electricity, Electronics, Information and Communications, and Control Engineering, to improve safety, mobility and efficiency. Such intelligent systems must incorporate the capability to collect traffic data. [1]In current transportation systems, traffic data are mostly managed by fixed sensing nodes, such as a ground loop detector, wireless sensors, and monitoring cameras. These fixed sensors can sustain for a long time once they are installed, but they have high installation and maintenance costs. Alternatively, traffic data can be obtained by analyzing aerial images from satellites, aircraft, helicopters, airships, and UAVs. Satellite and aircraft images are expensive and challenging to reflect time and weather changes on the fly. On the other hand, airships and UAVs can provide high-definition aerial images with relatively low costs. In particular, the use of aerial images taken by UAVs to detect and trace the objects on the ground is an actively researched area [3]. This paper proposes a pedestrian detection and tracking method that uses deep learning networks trained on aerial images from UAVs. The rest of the article is organized as follows. Section 2 describes deep learning, YOLO, and Kalman filter technologies that are adopted in the proposed method. Section 3 states the pedestrian detection and tracking method proposed in this paper. In Section 4, the experiment results of the proposed plan are presented. Finally, conclusion and future work are given in Section 5.

II. RELATED WORK

Various methods for object detection have been proposed to search for an object in an image or a video. The object detection methods relying on spatial domain analysis provide fast and easy object detection, but their performance decreases when the objects rotate, move, or change in size and color. The methods based on frequency domain analysis are more resilient to the rotation, abrupt motion, and changes in the appearance of the object, but their detection speed is slow. It indicates that object detection should be preceded by an image analysis that looks into how image objects change over time. Image analysis is the extraction of meaningful information from images using analysis software that segments pixels in a digital image based on features such as color, texture, density, and frequency. Image analysis algorithms recognize specific shapes and patterns in the images and gather quantitative information that is then used for further data analysis, like identifying a person from

Revised Manuscript Received on May 22, 2019.

JaeHee Lee, Dept. of Information & Communication, DongSeoul University, Sungnam, South Korea.

ChangJin Seo, Dept. of Information Security Engineering, Sangmyung University, Cheonan, South Korea (Corresponding Author).

their face. Naturally, the accuracy and speed of an object detection algorithm depend on the image analysis algorithm applied to it. Sophisticated object detection algorithms provide high efficiency, but their rate does not make them suitable for real-time detection. Simple object detection algorithms allow fast, real-time object detection in exchange for accuracy. The overall performance of an intelligent system that relies on fast, robust object detection and tracking is heavily influenced by the performance of the object detector it uses [4,5]. In recent years, object detection and tracking methods that use deep learning models have emerged to provide real-time object detection without loss of precision. A considerable amount of research has been conducted to develop fast and accurate object detection and classification systems based on deep neural networks [6,7].

A. Deep Learning

Deep learning has attracted a lot of attention in the machine learning community. The core concepts behind deep learning are based on artificial neural networks, developed in the 1980s. Early neural network algorithms could simulate only a minimal number of neurons at once, so they could not recognize complexity patterns. Due to several shortcomings such as slow learning rates, local minima, overfitting, and initial weights determination, they languished through the 1990s. In the mid-2000s, some researchers helped spark a revival of interest in neural networks with “deep” models that made better use of many layers of software neurons. The emergence of GPGPU with dominant processing power has also significantly decreased the time required for complex matrix operations. Today, deep learning algorithms are used in a variety of speech and image recognition applications, giving a significant performance boost. There exist numerous deep learning models, most of which are derived from artificial neural network models. Typical deep neural networks contain multiple non-linear hidden layers between the input and output layers, and this makes them very expressive models that can learn very complicated relationships between their inputs and outputs. In general, deep learning networks are trained using the backpropagation, a classical method used in artificial neural networks to compute a gradient. Deep neural networks then learn their weights and biases using the gradient descent algorithm. Modern deep learning models also have newly introduced concepts, such as pre-training for initial weights, mini-batches, and dropout, to overcome the problems inherent in traditional artificial neural networks [1].

B. YOLO(You Only Look Once)

YOLO is a network for object detection. As shown in [Figure 1], YOLO has 24 convolutional layers followed by two fully connected layers. The GoogLeNet model for image classification inspires YOLO's network architecture. Instead of the inception modules used by GoogLeNet, YOLO uses 1×1 reduction layers followed by 3×3 convolutional layers. It decreases the amount of computation, making fast object detection possible [8]. YOLO applies a single neural network to the full image. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. This unified architecture makes YOLO extremely fast. As illustrated in Figure 2, YOLO passes the left input image through its network and predicts the bounding boxes and class probabilities, represented in the middle pictures. The right photo in Figure 2 depicts the resulting detections of the YOLO network. YOLO splits the input image into a 7×7 grid (i.e., 49 grid cells). For each grid cell, two bounding boxes of varying sizes are created to locate the object whose center falls inside the grid cell. The YOLO network predicts 98 bounding boxes per image and class probabilities for each of the 49 grid cells. YOLO calculates the confidence score per bounding box so a total of 98 confidence scores can be obtained. The confidence score reflects how likely the box contains an object and how accurate is the bounding box. The higher the confidence score is, the thicker the rendered bounding box is. YOLO applies non-maximal suppression (NMS) to remove bounding boxes with lower confidence. It keeps those with high confidence scores as final predictions (e.g., the three bounding boxes marked in the right image of [Figure 2]).

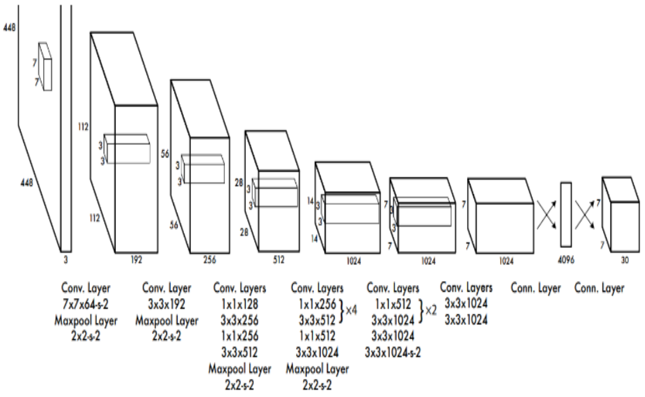


Figure 1. The YOLO Architecture [8].

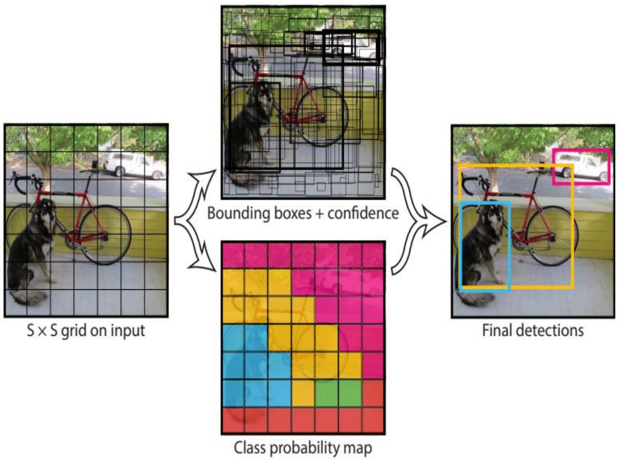


Figure 2. The YOLO Model [8].

C. Kalman Filter

The Kalman filter is a linear minimum variance of error filter, i.e., it is the best linear filter over the class of all linear filters.

The algorithm is recursive, and it can run in real-time processing, using only the current input measurements and the previously calculated state and its uncertainty matrix. Kalman filtering is a particular case of Bayesian filtering with linear, quadratic and Gaussian assumptions. As shown in [Figure 3], the Kalman filter keeps track of the estimated state of the current state vector x_t, along with the uncertainty of the estimate, over time. The estimate is updated using the state transition matrix Φ_t and the associated noise with known covariance Q_t (Gaussian noise) [9].

A tracking model is needed to construct a Kalman filter for object tracking. Our assumptions on pedestrian movement are as follows:

(1) Pedestrians tend to move at a steady pace, so there are no radical motion parameter changes inter-frame.

(2) Pedestrians' movement speed is almost constant over a unit interval of time.

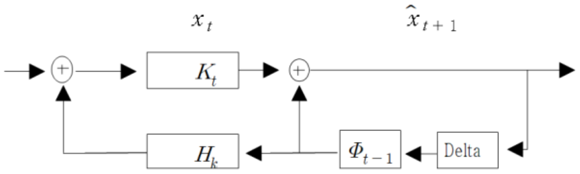


Figure 3. A Block Diagram of the Kalman Filter

III. Proposed Method

A. Implementation of the Proposed Method

The proposed method based on deep learning networks aims to detect and keep track of pedestrians via UAVs in real time. We implemented the proposed pedestrian detection and tracking method using YOLOv3, the latest variant of a famous object detection algorithm YOLO (see https://github.com/AlexeyAB/darknet) [10]. The proposed plan also adopts Kalman filtering with variable search ranges for effective pedestrian tracking. YOLOv3 improves some drawbacks of the previous versions of YOLO and outperforms other object detection algorithms like Faster R-CNN and SSD [11]. In our implementation, SDX-4195, a dedicated server for deep learning operations, was equipped with OpenCV 3.4, CUDA 9.1, Xeon E5-2650 4CPU, and GTX-1080Ti 4GPU, Darknet, an open source neural network framework written in C and CUDA, was installed for programming.

B. Network Training Data

As with any deep learning task, the deep neural network for pedestrian detection and tracking needs to be trained on real-world pedestrian images. The DJI Phantom 3 Professional took the pedestrian images for training. Oblique and orthogonal aerial pictures of the people who walk around the campus were taken at an altitude of 25-30m. The training dataset contained full HD 1920×1080 images with a refresh rate of 60 frames per second (FPS). [Figure 4] shows an example of the aerial pedestrian images in the network training dataset.

C. Training and Detection of Pedestrian

As with any deep learning task, the deep neural network for pedestrian detection and tracking needs to be trained on real-world pedestrian images. The DJI Phantom 3 Professional took the pedestrian images for training. Oblique and orthogonal aerial pictures of the people who walk around the campus were taken at an altitude of 25-30m. The training dataset contained.

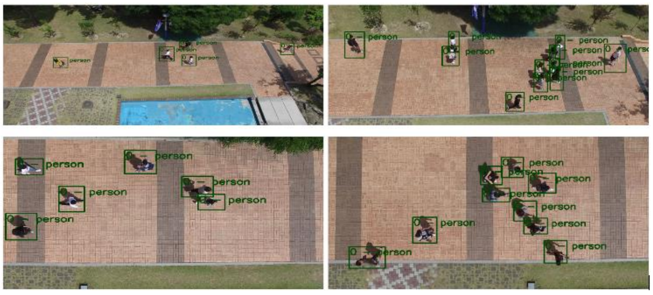
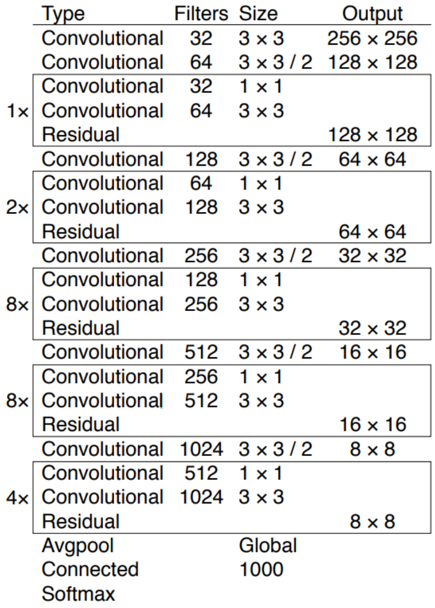


Figure 4. Pedestrian Images for Network Training [12].



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 5. The Structure of the Darknet-53 Model [11].

D. Pedestrian Tracking (Kalman Filter with Variable Search Ranges)

Once pedestrians are detected in the images, their moving speed and direction are needed to keep track of them in consecutive frames. In general, a difference (distance) of the pedestrian positions in the two successive frames is not significant, so the object nearest to the previous location of the target pedestrian within a search range is determined as the target pedestrian in the next frame. However, this approach gives rise to tracking errors when the search range is crowded with pedestrians walking closely or when pedestrians' motion changes dramatically in a short time. The proposed pedestrian detection and tracking method minimize such errors by building a Kalman prediction model with the trajectories of pedestrians extracted in the previous frames. We define a pedestrian motion model to trace a pedestrian in consecutive frames.

Step 1: The state vector of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 1: The Experiment Results of the Proposed Method

test video	total tracking frames	proposed method # of error frames	fixed method # of error frames	mean FPS
#1	4,764	31	52	42.39
#2	8,543	43	79	38.28
#3	28,767	276	485	43.78
#4	48,275	524	937	41.39

the target pedestrian at time k is expressed as follows in Equation 1.

$$\mathrm{x(k)} = [\mathrm{x(k)} \quad \mathrm{y(k)} \quad \Delta\mathrm{x} \quad \Delta\mathrm{y}]^{\mathrm{T}} \quad (1)$$

Step 2: Equation 2 is used to predict the position of the target pedestrian after a discrete time has passed. The estimated state vector of the target pedestrian at time k+1 is represented as follows:

$$\mathrm{x(k+1)} = \Phi(\mathrm{x})\mathrm{x(k)} + \mathrm{w(k)} \quad (2)$$

Step 3: where $\Phi(\mathrm{x})$ is the state transition matrix, and $\mathrm{w(k)}$ is the prediction noise distributed according to a Gaussian distribution with zero mean and covariance matrix $\mathrm{Q(k)}$.

Step 4: Assuming that the pedestrian movement has constant velocity and linear trajectories, the state transition matrix is represented as follows in Equation 3.

$$\Phi(\mathrm{k}) = \begin{bmatrix} 1 & 0 & \Delta\mathrm{t} & 0 \\ 0 & 1 & 0 & \Delta\mathrm{t} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Step 5: Suppose that the state vector and a set of measurement values have a linear relationship. The measurements (or observations) made at time k are defined as follows in Equation 4.

$$\mathrm{z(k)} = \mathrm{H(k)x(k)} + \mathrm{v(k)} \quad (4)$$

Step 6: where $\mathrm{z(k)}$ is the measurement values, $\mathrm{H(k)}$ is the measurement matrix, and $\mathrm{v(k)}$ is the measurement noise (a Gaussian distribution with zero mean and covariance matrix $\mathrm{R(k)}$).

Step 7: The filter's inputs are a four-dimensional vector, representing the coordinate x and y and variances in x and y directions. The measurement matrix $\mathrm{H_k}$ is expressed in Equation 5.

$$\mathrm{H(k)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

Step 8: $\mathrm{w(k)}$ is the prediction noise, and $\mathrm{v(k)}$ is the measurement noise. The difference $\mathrm{v(k)}$ between the predicted values $\hat{\mathrm{z}}(\mathrm{k|k-1})$ and the measurement values $\mathrm{z(k)}$ at time k is computed using Equation 6 and 7.

$$\hat{\mathrm{z}}(\mathrm{k|k-1}) = \mathrm{H(k)}\hat{\mathrm{x}}(\mathrm{k|k-1}) \quad (6)$$

$$\mathrm{v(k)} = \mathrm{z(k)} - \hat{\mathrm{z}}(\mathrm{k|k-1}) \quad (7)$$

The measurement noise $\mathrm{v(k)}$ is called the "innovation." The proposed method uses the innovation to set a search range for pedestrian tracking in the following frames. When the movement speed of the target pedestrian is fast, a more extensive search range is set. If the target pedestrian moves slowly, a narrower search range is given, enabling more fine-grained searches.

IV. Experimental Results

The performance of the proposed method that employs deep neural networking and Kalman filtering with variable search ranges was evaluated in the experiments. Its deep neural network was trained on aerial pedestrian images that were taken from the campus sky, at different angles. The dataset consisted of 4,157 training images and 1,386 test images. After the training was completed, the proposed method yielded an average intersection over union (IoU) of 79.68% and a mean average precision (mAP) of 0.9081 on the test set, with 3,453 true positives, 263 false positives, and 16 false negatives at the threshold of 0.25.

In the case of a single pedestrian, the proposed method successfully traced the pedestrian once he or she was detected. However, the following problems arose when there were multiple pedestrians.

1) Combination: More than two pedestrians are recognized as a single pedestrian.

2) Dissociation: A single pedestrian is known as several pedestrians.

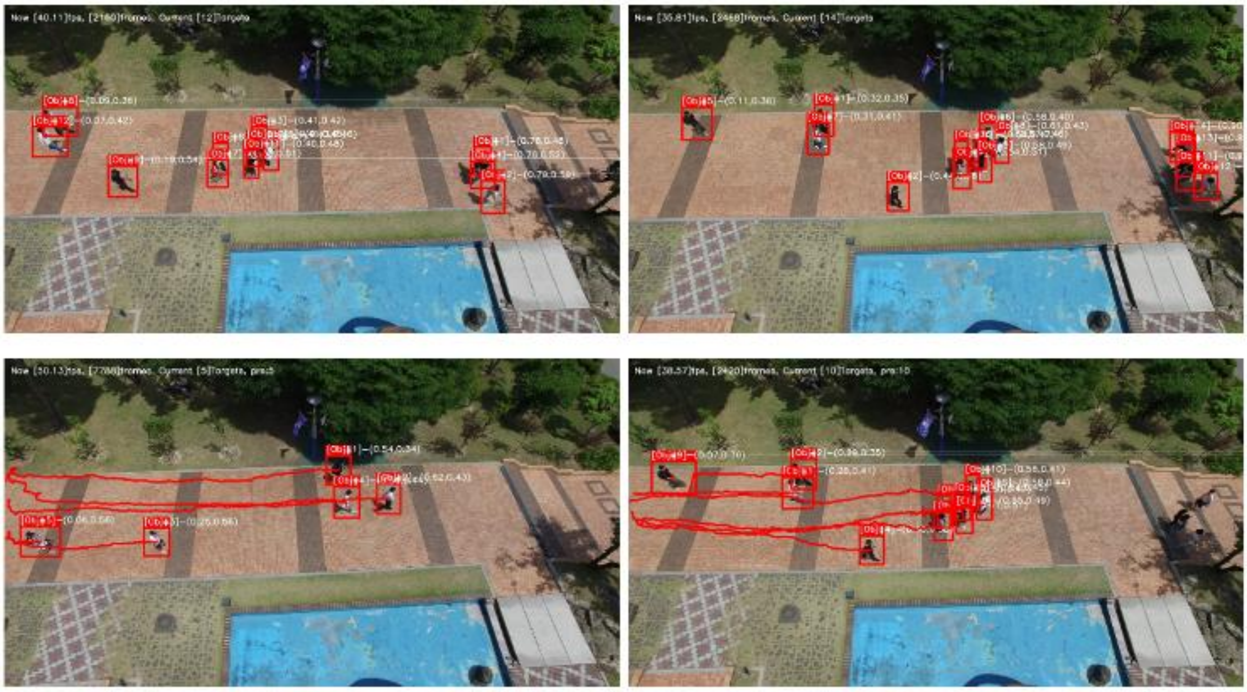


Figure 6: Pedestrian Detection and Tracking in Oblique Aerial Images

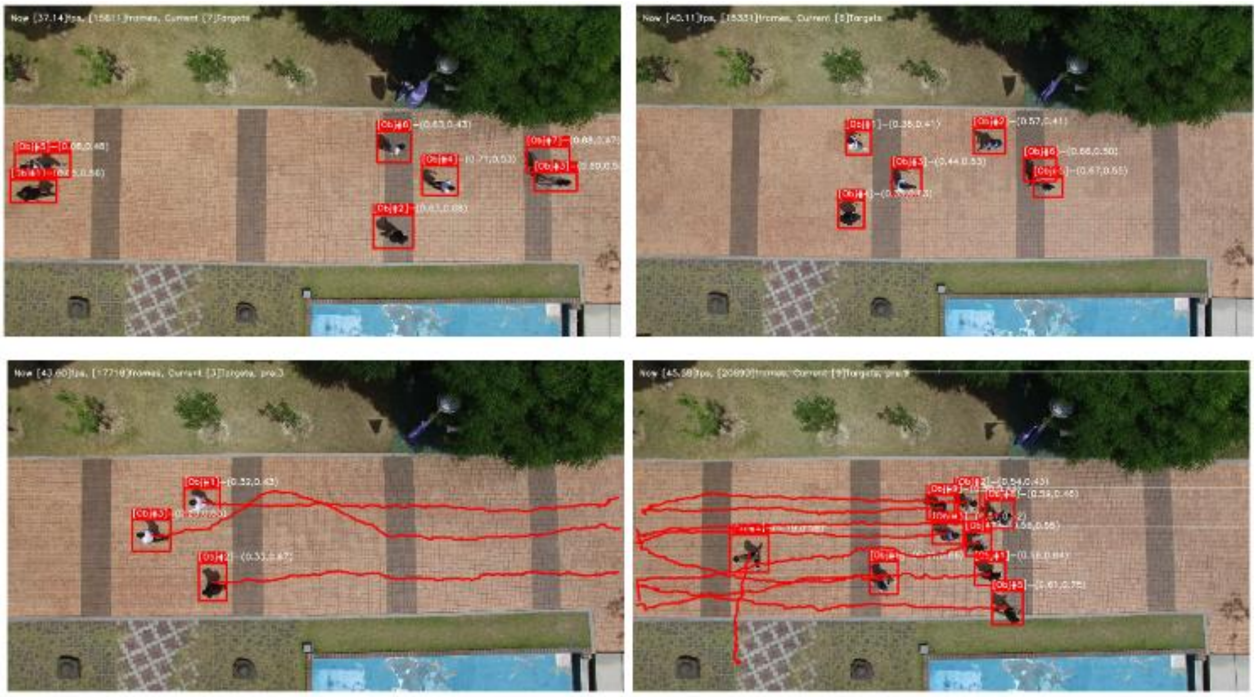


Figure 7: Pedestrian Detection and Tracking in Orthogonal Aerial Images

The problem of combination occurred when pedestrians, which were spatially separated at the beginning, converged. In the oblique aerial images, such pedestrians were overlapped, and those in the back were concealed from view, thus failing to be detected. The orthogonal aerial images that have a vertical picture on the ground alleviated this problem to some degree, but not entirely. When pedestrians walk hand in hand or arm in armor when they put their arms around each other's shoulders, the combination problem still arose. The proposed method flags those who are combined with the 'combined pedestrian' label and keeps track of the combined pedestrian. The problem of dissociation occurred when the combined pedestrian, a group of pedestrians who were being tracked together, diverged. This problem also occurred purely due to detection errors. The proposed method flags those who get separated from the combined pedestrian with the 'dissociated pedestrian' label and keeps track of them individually. [Table 1] presents the performance of the object tracking performed in our experiments. Here, the proposed method of having variable search ranges is compared to the traditional manner with fixed search ranges.

As can be seen in the table, the proposed plan is about 1.5 times more precise than the conventional method with fixed search ranges. [Figure 6] shows pedestrian detection and tracking in oblique aerial images, and [Figure 7] shows the same task carried out with orthogonal aerial images.

V. Conclusion

This paper has proposed the real-time pedestrian detection and tracking method that uses the deep neural network trained on aerial pedestrian images. In the experiments, the proposed method successfully detected and tracked pedestrians who move at 53 FPS maximum and 35-43 FPS on average. The proposed plan with variable search ranges made much fewer errors than the traditional object detector with fixed search ranges. Even though the dataset used in the experiments was not sufficiently big and diverse to make a definite statement, the proposed method achieved better speed and accuracy than other object detection algorithms. In the future, the proposed approach will be studied further with an extended set of aerial images taken at different times of day, and its performance in detecting and tracing different kinds of objects (e.g., persons, cars, and traffic lights) will be examined.

REFERENCES

1. *Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature, 2015 (521):436-444. DOI: https://doi.org/10.1038/nature14539*
2. *Dollar P., Wojek C., Schiele B., Perona P., Pedestrian detection: A benchmark. Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on, June 2009: 304-311. DOI: 10.1109/CVPR.2009.5206631*
3. *Rodrigo Benenson, Mohamed Omran, Jan Hosang, Bernt Schiele, Ten Years of Pedestrian Detection, What Have We Learned? Computer Vision - ECCV 2014 Workshops LNCS, 2015 8926:613-627. https://doi.org/10.1007/978-3-319-16181-5_47*
4. *Yao Wentao, Deng Zhidong, A Robust Pedestrian Detection Approach based on Shapelet Feature and Haar Detector Ensembles. Tsinghua Science and Technology, Feb 2012 17(1):40-50. DOI: 10.1109/TST.2012.6151906*
5. *Zhiqian Chen, Kai Chen, Chen J., Vehicle and Pedestrian Detection Using Support Vector Machine and Histogram of Oriented Gradients Features. Computer Sciences and Applications (CSA), 2013 International Conference, Dec. 2013:365-368. DOI: 10.1109/CSA.2013.92*
6. *Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Jun2 2017 39(6):1137–1149. arXiv:1506.01497v3*
7. *Mahyar Najibi, Mohammad Rastegari, Larry S. Davis, G-CNN: An Iterative Grid Based Object Detector. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 2369-2377. arXiv:1512.07729v2*
8. *Redmon Joseph, Divvala Santosh, Girshick Ross, Farhadi Ali, You Only Look Once: Unified, Real-Time Object Detection. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:779-788. arXiv:1506.02640v5*
9. *S. Y. Chen, Kalman Filter for Robot Vision: A Survey. IEEE Transactions on Industrial Electronics, Nov. 2012 59(11):4409 – 4420. DOI: 10.1109/TIE.2011.2162714*
10. *Joseph Redmon, Ali Farhadi, YOLO9000: Better, Faster, Stronger. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017:7263-7271. arXiv:1612.08242v1*
11. *J. Redmon and A. Farhadi, YOLOv3: An incremental improvement. Computer Vision and Pattern Recognition, 2018, arXiv:1804.02767v1*
12. *Alexey, Yolo-v3 Windows and Linux version. [Internet]. [cited 2019 April 08], Available from: https://github.com/AlexeyAB/darknet (website)*

AUTHORS PROFILE

JaeHee Lee has received his Ph. D. in Electronic Engineering from Kwangwoon University, South Korea in 2001. He started his career as ADD (Agency for Defense Development) in 1987. From 1999 to currently, he is working as a Professor at DongSeoul University, Dept. of Information Telecommunication, South Korea and involved actively in teaching and research.

ChangJin Seo has received his Ph. D. in Multimedia Engineering from Pusan National University, South Korea in 2003. He started his career as Sensor Technology Research Center (STRC) in 1999 and thereafter he moved to Sungduk University in 2000 as a Professor. From 2013 to currently, he is working as a Professor at Sangmyung University, Dept. of Information Security Engineering, South Korea and involved actively in teaching and research mainly in the area of Deep Learning, Object Detection and Target Tracking.