IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE VEHICULAR TECHNOLOGY SOCIETY SECTION

Received 9 January 2023, accepted 7 March 2023, date of publication 23 March 2023, date of current version 29 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3260832

RESEARCH ARTICLE

PairingNet: A Multi-Frame Based Vehicle Trajectory Prediction Deep Learning Network

GUAN-WEN CHEN[1], MIN-TE SUN[2], (Member, IEEE), AND TSÌ-UÍ ÍK[1], (Member, IEEE)
[1]Department of Computer Science, College of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan
[2]Department of Computer Science and Information Engineering, National Central University, Taoyuan 320317, Taiwan

Corresponding author: Tsì-Uí Ík (cwyi@nctu.edu.tw)

This work was supported in part by the Center for Open Intelligent Connectivity from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, and in part by the Construction Office of Chiayi County Government of the Republic of China. The work of Tsì-Uí Ík was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 111-3114-H-A49-001, Grant MOST 111-2622-E-A49-009, and Grant MOST 110-2221-E-A49-063-MY3.

ABSTRACT The traffic infrastructure of a city requires evaluation and improvement through a large amount of data analysis. The construction and laborious work of traditional methods make computer vision flourish in traffic analysis. Among different computer vision technologies for intelligent transportation system, one of the most important algorithms is multiple object tracking (MOT). At present, MOT used in traffic analysis has several shortcomings, such as lack of the output of vehicle speed and movement direction, and the consideration of single factor in trajectory tracking. These limitations have affected the results of traffic analysis. This research proposes an end-to-end deep learning network, called PairingNet. In addition to retaining the function and accuracy of the original detection network, PairingNet integrates the calculation of vehicle trajectory into the network through the feature fusion of consecutive images, which is introduced to predict the movement direction and speed of the vehicle. These additional features can be used to better track the trajectories of vehicles. In addition, a pipeline is designed to reduce the loading latency incurred by using the consecutive frames as the input for PairingNet. The experiment results indicate that the vehicle identification of PairingNet reaches 96% accuracy, surpassing the original YOLOv3 as the backbone structure, and reaches a near 100% accuracy rate in the predicted vehicle position. Moreover, with the pipeline process, the inference speed of PairingNet is very close to the original YOLOv3. In MOT results, the MOTA of PairingNet also has a high performance of 91%.

INDEX TERMS Deep learning, multiple object tracking, object detection, traffic analysis, vehicle trajectory.

I. INTRODUCTION

Traffic data collection and analysis are key components in smart cities for efficient and real-time management of transportation networks. This motivates researchers of Intelligent Transportation Systems (ITS) to continuously develop more accurate and more cost-effective methods for traffic data collection. One of the recent methods is the Unmanned Aerial Vehicle (UAV) with video streaming capability which covers wide area, reduces installation costs, and minimizes errors [1], [2], [3], [4]. Also, the UAV supported by image processing techniques allows for collecting microscopic traffic information, such as vehicle type, direction, trajectory, speed, and driving behavior [1], [2], [5]. Thus, intelligent image processing methods are required for detecting and tracking objects captured in the UAV videos.

Several fundamental object detection methods have been used in ITS [1], [2], [3], [4], [6], [7]. Although these studies achieved high accuracy in object detection, the main focus was on detecting a single object to determine road parameters and traffic variables with little information about object type and trajectories. Another popular algorithm for detecting multiple objects and trajectories is Simple Online and Real-time Tracking (SORT) [8]. SORT utilizes the Kalman filter and the Hungarian algorithms to speed up the computation process. Also, SORT provides a method

The associate editor coordinating the review of this manuscript and approving it for publication was Hassan Omar.

to generate trajectories step by step. However, the multiple steps cause more errors since these steps have a knock-on effect, and the huge amount of frame information affects the calculation of the prediction significantly. Further, the Kalman filter based methods suffer from low accuracy particularly when they deal with moving images such as drone videos [1].

This study focuses on detecting vehicles on roads, classifying them, and determining their trajectories. To achieve these goals, the study proposes PairingNet which is an end-to-end network with object detection and prediction algorithm. We introduce the concept of TrackNet [9] which continuously analyzes multiple frames to detect the objects and simultaneously predict their next positions. Therefore, the prediction not only considers the speed and the position, but also the vehicle features. PairingNet is divided into object detection, prediction, and pairing. Object detection is used to determine the positions of vehicles. The position of each vehicle in the next frame is predicted using the information acquired in the previous frames. Finally, the pairing algorithm is used to get the trajectory of each vehicle in the video. To evaluate the performance of PairingNet, aerial videos taken by UAV are used and analyzed. The results show that the mAP of the object detection in PairingNet is 96%, which is similar to YOLOv3, and the precision of the pairing prediction is close to 100%.

In summary, the contributions of this research are as follows.

- We propose PairingNet, an end-to-end object detection and prediction, for vehicle tracking based on the top-view traffic video recorded from a UAV. PairingNet is inspired by YOLOv3 and TrackNet. It takes three consecutive frames as the input to extract more information from vehicles.
- PairingNet detects vehicles with not only the position of the bounding box and vehicle category but also the information including the driving direction and status. The output of PairingNet can be used to improve the performance of the vehicle tracking algorithm.
- To minimize the calculation cost of PairingNet, a pipeline process is introduced to reduce the delay from loading three consecutive frames as the input. The experiment results show that the inference time of PairingNet with pipeline is extremely close to that of YOLOv3.

The rest of this paper is structured as follows. Section II illustrates the related works about object tracking algorithm. Section III discusses the traditional and modern methods of vehicle detection. Section IV-A and Section IV-B explain the proposed PairingNet and the multiple object tracking (MOT). Section V presents the training and testing dataset. Section VI shows the experiment results and performance analysis, and finally Section VII concludes the paper and outlines future work.

II. RELATED WORK

Traditional MOT algorithms, such as Multiple Hypothesis Tracking [10], Optical flow [11], and Joint Probabilistic Data Association filters [12], usually consume much time, and the time increases exponentially with the number of objects being tracked [13]. Hence, these algorithms are not appropriate for application to traffic surveillance in this study. Instead, the Kalman Filter [14], [15], which has fast calculation, has become widely used in MOT. For example, the SORT [8] method proposed in 2016 a fast and stable algorithm by combining Kalman Filter and Hungarian algorithm [16]. However, SORT has certain limitations, such as using singular consideration factors and the initialization parameter, which greatly affect the performance of the algorithm. Therefore, several studies proposed advanced network based on SORT or Kalman Filter, such as Deep SORT [13] and MOANA [17], which take into account the shape and color of the objects to strengthen the capability of trajectory tracking.

Deep SORT uses the prediction model of Kalman Filter and the Mahalanobis distance to calculate the spatial correlation of objects between two frames in the paring algorithm [13]. In addition, to eliminate a large number of paring errors when the object has high motion uncertainty, Deep SORT considers the appearance of the object as the paring information which greatly improves the object tracking accuracy. The Appearance Metric in Deep SORT is provided in Equation 1.

$$d(i,j) = \min(1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in \mathbb{R}_i) \tag{1}$$

In the equation, $d(i,j)$ measures tracking association with j-th detected object in i-th trajectory. Each bounding box d_j has an appearance feature descriptor r_j with $|r_j| = 1$. For the i-th trajectory, multiple trajectory appearance descriptors $r_k^{(i)}$ are recorded in $R_k = \left\{r_k^{(i)}\right\}_{k=1}^{L_k}$, where k is the number of trajectories. The equation calculates the cosine distance of the i-th trajectory and the latest j-th bounding box, and finds the most similar appearance and spatial paring result to identify a more realistic pairing.

TrackletNet Tracker(TNT) [18] is a tracking algorithm that has made achievements in the MOT field in recent years. It is based on the operation of Tracklet, which integrates the concept of Graphical module combined with cluster analysis to distinguish the tracklet of each different object, and finally reconstructs the complete trajectory. The whole TNT operation is explained as the Tracklet is generated by the result of IoU and appearance feature. Then, the Tracklet represents the nodes in the graph model and pushes two different Tracklets into the network to calculate the similarity P_e of the two Tracklets. Equation 2 is used to get the *EdgeCost* of the two Tracklets (nodes). The P_e is between zero and one, where a higher P_e means a higher similarity, and consequently a closer distance. At the end, the graph model is taken as an input of the cluster analysis. The Tracklets that are put into the same cluster are regarded as the same object, and these Tracklets

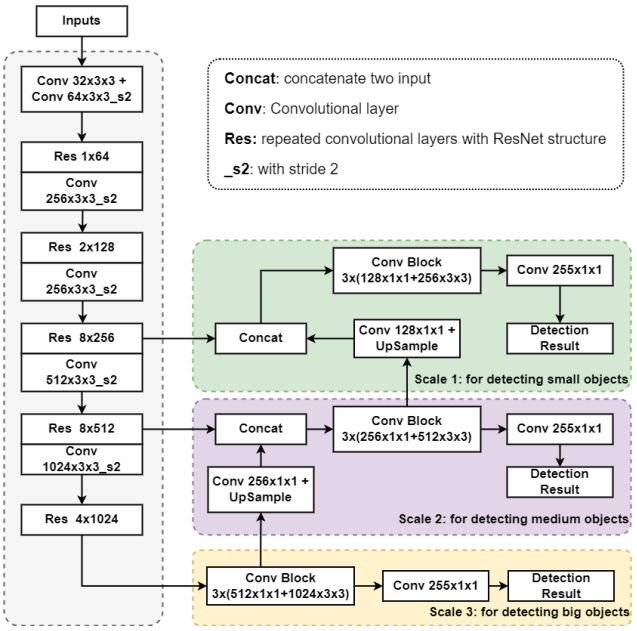

FIGURE 1. Network architecture of YOLOv3.

are then merged into a trajectory.

$$EdgeCost = log(\frac{1 - P_e}{P_e}) \tag{2}$$

III. PRELIMINARY

The collection of traffic data has evolved from simple traditional tools to advanced ones such as Unmanned Aerial Vehicle (UAV). Traditional tools include inductive loop [19] in intrusive way, microwave radar [20] in non-intrusive way, and roadside camera using computer vision. The new data collection method by UAVs reduces the infrastructure cost and maximizes the accuracy. Vehicle detection by UAV uses computer vision methods for video processing and object detection.

The most widely used object detection methods for computer vision are feature extraction by histogram of oriented gradient (HOG) [21], scale-invariant feature transform (SIFT) [22], and speeded up robust features (SURF) [23]. To identify objects in the images, the extracted features are used as the inputs for the training process performed by machine learning methods, such as SVM [24] and Adaboost [25].

Recently, the deep learning network has vigorously been developed due to the advancement of computer hardware. The Convolutional Neural Network (CNN), which is a classic deep learning network, has made a breakthrough in computer vision analysis, and aims to learn features in each layer using different filters [26]. The filters extract the characteristics of objects from the images as the features, then the deeper filters will elicit the more complicated features like patterns or shapes. A well-designed network with a large number of filters in different layers can extract better features to strengthen the performance of object detection. Nowadays, most of the CNN based object detection methods can be classified into two categories, which are region-based methods and region-free methods.

The region-based methods generate a lot of region proposals, then detect the objects and their position with specific regions. R-CNN, Fast R-CNN, and Faster R-CNN are the most popular region-based object detection networks [27]. These networks use CNN to extract features from specific regions, which are selected from the region proposals in different ways, then they use SVM or regressor to classify and position the objects.

The region-free methods take the whole frame as the input without generating region proposals to detect the objects [28]. The region-free methods also divide a frame into several small regions, and detect objects from these regions. Some of the region-free object detection networks add anchor boxes in every small region before detecting objects to reduce the calculation. The You Only Look Once (YOLO) is a famous region-free object detection network [29]. YOLO is a real-time object detection network proposed in 2015 and extracts features from the entire image. The original concept of YOLO is dividing an image into $S \times S$ grids. For the identification, the grid image is divided into two parts. The first part provides several bounding boxes in different scales to identify the target position and confidence, and the other part forms small areas to distinguish classes. YOLOv1's network architecture has a final output dimension of $7 \times 7 \times 30$ which means $7 \times 7 \times (2 \times 5 + 20)$. Each small area predicts 4 parameters and 1 credibility of 2 identification frames, and 20 class probabilities of each small area.

YOLOv3 introduced residual network (ResNet) and feature pyramid to improve the detection performance in small objects [30], [31]. The architecture of YOLOv3 is shown in Figure 1. The features of an input image are extracted by several ResNet in different scales. Then, the feature pyramid structure is adopted to generate three feature maps. Similar to YOLOv1, input image of YOLOv3 is also divided into several grids for predicting objects. Three prior anchors in different scales are utilized for three feature maps to predict objects. In Figure 1, the size of output from three feature maps is $(5 + class) \times 3$, where 5 represents the 4 parameters and the confidence of the predicting box, *class* is the probability of the object class, and 3 means the predicting boxes from the three different prior anchors. Although the detection accuracy of YOLO is not as high as that of the faster R-CNN, its ultra-fast calculation and medium-to-high accuracy rate are effective for object detection [28].

Object detection algorithms usually take an image as the input of the network to detect and recognize objects. However, the applications in the field of smart transportation, city, and sport often require extraction of trajectories of the objects after the detection to analyze the behavior from them. In Section II, several multiple object tracking algorithms using object bounding boxes are introduced. In these works, objects occluded by obstacles and other objects will often lead to detecting errors. Their trajectories will then

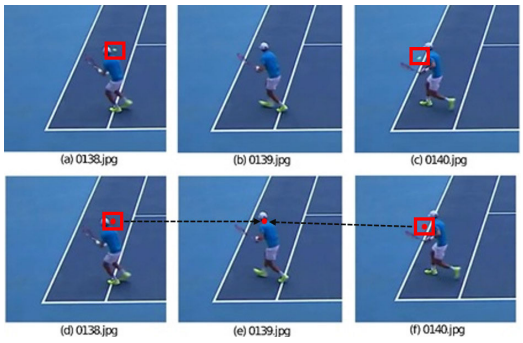

FIGURE 2. The concept of detection by tracking in tennis to reduce the effect of occlusion [9].

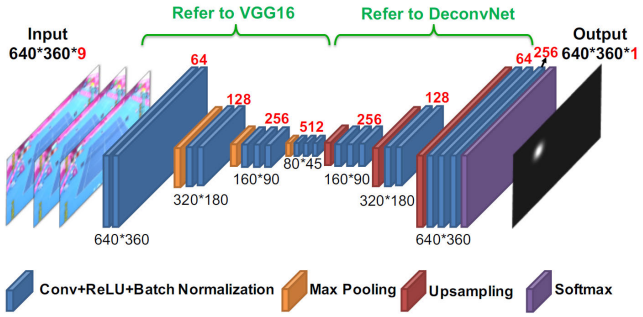

FIGURE 3. The architecture of TrackNet [9].

be interrupted in the tracking algorithm, which will consequently result in tracking errors. Therefore, a concept, called detection by tracking, has been proposed to reduce the errors caused from object occlusions. TrackNet [9] is a deep learning network for tracking a tennis ball at high speed. In Figure 2, (a)-(c) are the three consecutive frames from a video where the tennis ball is occluded by the tennis player in frame (b). The concept of detection by tracking is shown in Figure 2 (d)-(f). The tennis ball can be detected and tracked in the previous and the next frames, then the position of the occluded tennis ball in the middle frame (e) can be predicted.

Figure 3 shows the architecture of TrackNet, and the input is three consecutive frames of the video to realize detection by tracking.

IV. METHOD

A. PairingNet

PairingNet proposed in this research is inspired by YOLOv3 and TrackNet [9]. In TrackNet, the tennis ball is tracked with three input frames to solve the issue caused by occlusions, as shown in Figure 2. YOLOv3 integrates ResNet and feature pyramid to improve the performance of detecting small objects. The proposed PairingNet takes three consecutive images as an input and targets multiple objects in different frames for detection and tracking. For tracking detected objects, PairingNet uses the feature extraction framework ResNet in YOLOv3 to extract similar features from images, and then calculates the information for each object in the consecutive frames. The information consists of the locations of the object in the three frames and the algorithm uses the information to obtain the trajectory of an object. The architecture of PairingNet is shown in Figure 4. From the input, the network extracts features by ResNet, then builds feature pyramid with top-down manner and lateral connections to improve the capability of predicting small objects.

The feature pyramid consists of three feature maps, which are $13 \times 13 \times 1024$, $26 \times 26 \times 512$, and $52 \times 52 \times 256$. PairingNet also predicts three bounding boxes at each scale. The output of the network contains the original five parameters of a bounding box, which are the center coordinates, width, height, confidence, and the additional four new parameters of a predicting pair, which are the coordinates of the predicting position, past confidence, and future confidence, as depicted in Figure 5. In the figure, (f_x, f_y) represents the coordinates of the predicting pair. The parameters *Past* and *Future* are the past and future confidence of the pair, respectively. In addition to the coordinates of the object and the pair, the network also considers the object confidence in previous frame *Past* and next frame *Future* to determine if the object is entering or leaving the section.

In addition to the original loss function of YOLOv3, PairingNet loss L_{pair} includes four new loss terms: L_{fx}, L_{fy}, L_{Past}, andL_{Future}. L_{fx} and L_{fy} are the Mean Square Error (MSE) of the x and y coordinates, respectively. L_{Past} and the L_{Future} are the Binary Cross-Entropy (BCE) of the past and future confidence, respectively, as the past and future confidence are binary. The used BCE is different from the original BCE Loss calculation method since a weight α is added to make the calculation of the loss function more consistent with the results of the experiment. The modified BCE is given by Equation 3.

$$WeightedBCE = -\frac{1}{n}\sum_{i=1}^{n}[\hat{x}_i log(x_i) + \alpha \times (1-\hat{x}_i) log(1-x_i)] \tag{3}$$

When analyzing the dataset and the two parameters, future and past, we can see that the probability of these parameters to be zero is low. For example, a car takes about 5 seconds from appearing on one side of the screen to driving away from the other side when there is no red light. If the experiment used eight FPS to train the network, as in the benchmark, the probability of the past or future parameter to be 0 is only about $\frac{1}{40}$. The past parameter is zero at the moment when the car has just entered the screen, and the future parameter is zero at the moment when the car is about to leave the screen. For all remaining frames, these parameters are all ones. To prevent the network from biasing towards predicting one, the error when the standard is zero is given a relatively large weight.

The newly added four loss terms all rely on the effectiveness of the original YOLOv3 object detection. If the object detection performance of PairingNet is not as expected, the subsequent calculation of each pair is severely limited by detection errors, which will make the network meaningless. Therefore, in the calculation of PairingNet loss function L_{pair},

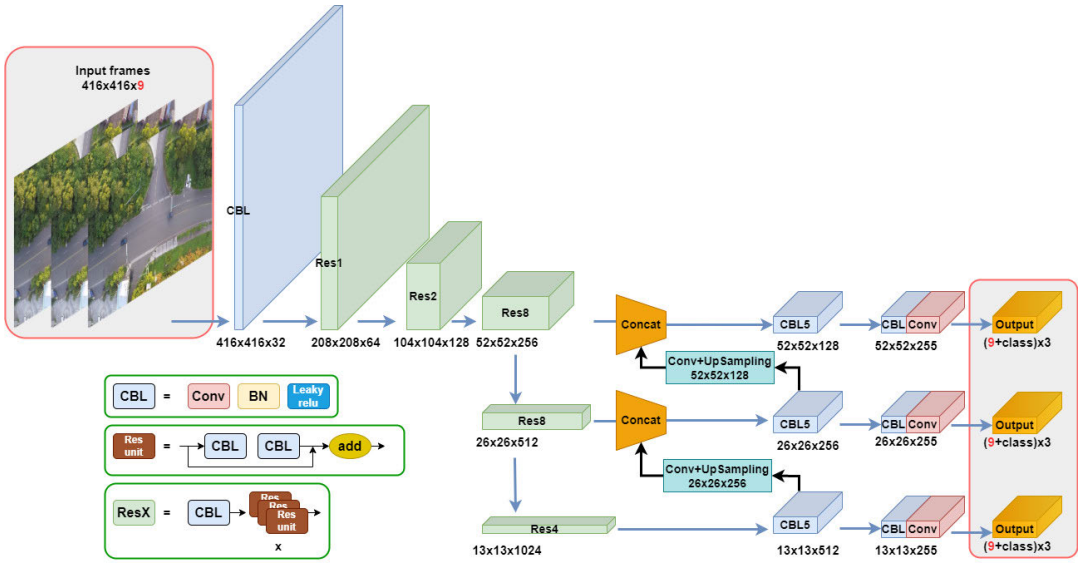

FIGURE 4. The architecture of PairingNet, which is based on YOLOv3 and Tracknet.

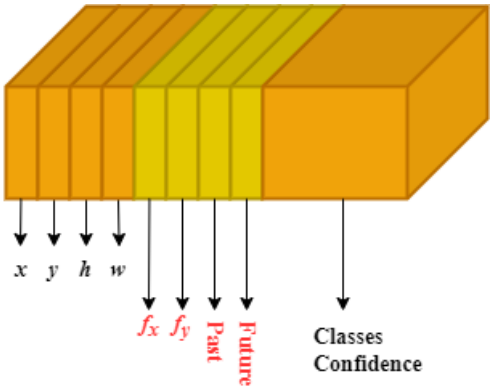

FIGURE 5. Pairing output, including original YOLOv3 output (green) and our new output (yellow).

a weight value β is added to distinguish the original L_{yolov3} from other newly added Loss, as shown in Equation 4.

$$L_{pair} = \beta \times L_{yolov3} + [L_{fx} + L_{fy} + L_{past} + L_{future}]. \quad (4)$$

It is expected that the coordinates of each pair can be found without reducing the detection performance, as illuminated in Equation 4.

Three consecutive images from the video are used as the inputs of PairingNet to extract additional features from the detected vehicle. Compared to the single image as the input of YOLOv3 [29], PairingNet consumes more time to load multiple images during inference. To reduce the data loading latency, a pipeline is introduced to load the images from the video. At first, PairingNet loads the first three frames of the video, then the first of the three frames will be deleted after inference. The next frame right after the first three frames will be loaded for the next inference. This process is repeated so that at each inference, only one frame needs to be loaded. As will be shown in Section VI-B, this pipeline trick significantly improves the inference speed of PairingNet to be comparable to YOLOv3's.

B. MULTIPLE OBJECT TRACKING ALGORITHM

The vehicle information extracted from every three consecutive frames can be paired via the MOT algorithm. Hungarian algorithm, one of the most popular algorithms of MOT, is a minimum loss optimization algorithm to solve assignment problems. For example, Equation 5 shows a loss matrix of vehicle trajectories, each row in the matrix is a detected vehicle in the current frame, and each column is a trajectory extracted from previous frames. End each element of the matrix denotes the loss of the detected vehicle referring to the trajectory. The multiple vehicle tracking task is to find out the optimal solution to assign the detected vehicle to a trajectory, and each trajectory can only be paired to a single vehicle of the current frame. In this case, the optimal solution of Equation 5 using Hungarian algorithm is L_{11}, L_{23}, and L_{32}. The total loss of the solution is $2 + 1 + 0 = 3$, which is the lowest loss.

$$L = \begin{bmatrix} 2 & 3 & 0 \\ 4 & 5 & 1 \\ 1 & 0 & 5 \end{bmatrix} \quad (5)$$

In this paper, Hungarian algorithm is used for tracking vehicles, and the results are optimized using additional parameters, which are the threshold of distance and vehicle status. Vehicles detected from object detection algorithm contain true positive results, false positive results, and false negative results. The higher performance of the algorithm causes smaller quantities of false positive and false negative results. To reduce the incorrect tracking results affected by the object detection, the distance between detected vehicles in consecutive frames is utilized for optimization. Note that the

driving speed of the same vehicle should be in a reasonable range and the driving speed of different vehicles in the same direction are close to each other. As a consequence, a distance threshold is used as a filter, which will remove the tracking results if the vehicle distance between the last frame in the trajectory and the current frame is more than the threshold. On the other hand, parked vehicles on the roadside also may cause mismatches of the tracking results when a detected vehicle is driving through. In this paper, each detected vehicle in the video will be marked with a tag after PairingNet to determine its status, whether the vehicle is moving or not. There are two cases when a vehicle is tagged to be not moving. The first case is the vehicle is parked, and the second case is that it is waiting for a traffic light. In the case when a detected vehicle in the current frame is marked as moving and is assigned to a trajectory, but the vehicle in the latest frame is marked as stop, the assignment will be removed if the distance between the vehicle in those two frames is higher than a different threshold.

After most of the detected vehicles are assigned to existing trajectories using Hungarian algorithm with optimization, some of the vehicles may not belong to any trajectory because they are just entering the road section in the video. Therefore, the values of *Past* and *Future*, part of the output of PairingNet, can be used to identify the unassigned vehicles. When *Past* is false and *Future* is true, it means the last frame of the input of PairingNet does not contain the unassigned vehicle. So the vehicle can be treated as an entering vehicle to the road section in the video. In this case, a new trajectory will be created to record the unassigned vehicle. After all detected vehicles are assigned to a trajectory, some of the existing trajectories will be removed if no updated vehicle is assigned to them for a duration longer than a threshold.

V. DATASET

The dataset used in this study is recorded from aerial videos with the versatile action camera at an altitude of 25 and 50 meters over an intersection in Hsinchu, Taiwan, as shown in Figure 6. The vehicles and pedestrians in the videos are labeled for object detection and tracking, and the labels are classified into eight categories: pedestrian, bike, scooter, sedan, bus, s_Truck for small truck, l_Truck for large truck, and trailer, as shown in Figure 7. In the s_Truck category, the length of the truck is less than 3 meters.

GoPro Hero 7 [32], a versatile action camera, is used for recording the video in this paper. The camera has several operational modes with different points of view (PoV) in vertical, horizontal, and diagonal view, as shown in Table 1. Each video in the dataset has a resolution of 1920 × 1080 in wide mode, and its horizontal PoV is 118.2°. In most videos, the altitude of UAV is 50 meters, so the frame width in the video is about 173 meters. Two image sizes, 416 and 608 pixels as frame width, are used as the input of PairingNet, so the scales are 0.416 and 0.285 meter per pixel in the two frame sizes. However, small objects like pedestrians, bikes, and scooters are more difficult to be detected than large objects. To improve small object detection, additional videos with the altitude of UAV being 25 meters are collected. The configurations used to collect all videos are listed in Table 2.

FIGURE 6. Aerial scenes in dataset.

FIGURE 7. Category of object detection dataset, including pedestrian and vehicle.

TABLE 1. Degree of camera view.

Mode	Vertical PoV	Horizontal PoV	Diagonal PoV
4 × 3 Wide	94.4°	122.6°	149.2°
4 × 3 Linear	71.0°	86.7°	100.0°
16 × 9 Wide	69.5°	118.2°	133.6°
16 × 9 Linear	56.7°	87.6°	95.5°

TABLE 2. Small object scale.

	Pedestrian	Bike	Scooter
real (meter)	0.4 × 0.4	1.6 × 0.4	1.7 × 0.6
50m, 416p, 0.416m	1 pixel	4 pixel	4-8 pixel
50m, 608p, 0.285m	2-4 pixel	6-12 pixel	12 pixel
25m, 416p, 0.208m	4 pixel	16 pixel	24 pixel
25m, 608p, 0.143m	4-9 pixel	36 pixel	48 pixel

An open-source online labeling tool, called Video Annotation Tool from Irvine, California (VATIC) [33], is used in this research. The design of the user interface is suitable for labeling consecutive frames with multiple objects. In our dataset, all the objects of interest are labeled by VATIC for PairingNet. There are 9 parameters to be marked, which are *TrackID*, *xmin*, *ymin*, *xmax*, *ymax*, *FrameID*, *Lost*, *Occluded*, and *Label*. *TrackID* denotes the object in the same track, so the objects with the same *TrackID* in different frames can be identified as the same object. The *xmin*, *ymin*, *xmax*, and

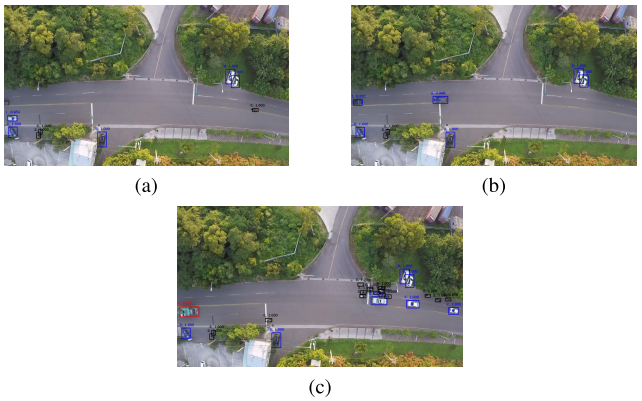

FIGURE 8. Object detection and pairing results in the test data. (a) and (b) show the appearing status and leaving status separately, and (c) is a situation of many vehicles that are closed.

ymax are the upper-left and the lower-right corners of the bounding box that locates an object in the frame. *FrameID* marks the identity of current frame to the objects. The label *Lost* is 1 if the object is outside of the screen. *Occluded* will be marked 1 if the object is occluded. *Label* denotes the category that the object belongs to. The label data of the objects in each frame are exported to a different annotation file separately.

VI. RESULT AND EVALUATION

The results of object detection, pairing, and tracking are presented in this section. Section VI-A shows the experiment results using test data from the dataset. Then Section VI-B compares the results of object detection and pairing between YOLOv3 and ours, and Section VI-C exhibits the performance of the multiple object tracking algorithm.

A. DETECTION AND TRACKING RESULT

The output of PairingNet includes the vehicle location, predicted moving vector, vehicle state, and object confidence. The confidence threshold is preset to 0.8, and the Non-Maximum Suppression (NMS) threshold is 0.25. As shown in Figure 8, the first number on top of the bounding box of the vehicle in the picture represents its condition. Condition 1, 2, and 3 indicate the three instantaneous situations, i.e., just enter the screen, move in the screen, and about to leave the screen. The second number is the confidence of prediction. The vector in the bounding box represents the vehicle direction and speed. For example, the scooter in Figure 8c has condition 2, and its vector points from the right to the left in the screen. It means that this scooter is clearly visible in the screen in last frame and also moves in the same direction as the predicted vector. In Figure 8a, the vehicle is leaving the section, so its condition is 3. In addition, because we cannot see this vehicle in the next frame, the vector will not be shown in the bounding box. Figure 9 shows tracking result in two consecutive frames. The number on top of the bounding box of the vehicle is the tracking ID, so each vehicle will have a unique tracking ID in the consecutive frames.

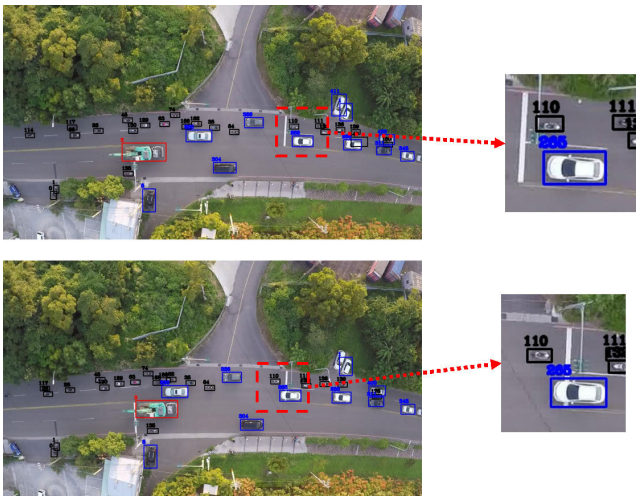

FIGURE 9. Visualization of the tracking results.

B. EVALUATION OF PairingNet

To evaluate the proposed PairingNet, a total of 13312 images are used as training data, and 3328 images are used as testing data. We start with the detection evaluation, and then show the pairing evaluation. The performance of object detection result is evaluated by mean average precision (mAP), which is determined by the value of *Precision* and *Recall*. *Precision* and *Recall* are calculated by the number of true positive (TP), false positive (FP), and false negative (FN). TP means the detected vehicle is the same as the ground truth. FP means the detected vehicle does not exist in the ground truth. FN indicates the object in the ground truth is not detected by the network. *Precision* is defined as $TP/(TP+FP)$, and $Recall = TP/(TP + FN)$. Intersection over union (IoU) is used to determine the correctness of detected object with ground truth, which is calculated by the ratio of the area of overlap and the area of union between two bounding boxes. In our experiment, a detected vehicle will be a TP instance while the IoU between the bounding boxes of detected vehicle and the ground truth is higher than 0.5, which is the same as in YOLOv3 [29].

1) DETECTION EVALUATION

Table 3 shows the detection results of the test data by PairingNet. The mAP in the table reaches 96.4% for the five categories. The accuracy and recognition rate of each category exceed 90%; the precision and recall reach 96.1% and 98.6% respectively. Comparing these results with those of YOLOv3 in Table 4, PairingNet results have higher recognition accuracy through the feature fusion of consecutive images. Also, the recognition of small objects, including pedestrians and scooters, has improved significantly. In particular, the pedestrian recognition increases by 27.7%. However, in multiple consecutive images, s_Truck has obviously more FN because the only recognition difference between s_Truck and l_Truck is the size. The probability of s_Truck to be misjudged as l_Truck increases after the feature fusion of multiple images.

TABLE 3. The detection result of PairingNet.

	TP	FP	FN	Precision	Recall	AP
Scooter	15999	1214	355	92.9%	97.8%	91.9%
Sedan	20766	284	153	98.7%	99.3%	98.7%
s_Truck	383	19	11	95.3%	97.2%	96.0%
l_Truck	578	15	6	97.5%	99.0%	97.0%
Pedestrian	30	2	0	93.8%	100.0%	98.3%
Total / Mean	37756	1534	525	96.1%	98.6%	96.4%

TABLE 4. The detection result of YOLOv3.

	TP	FP	FN	Precision	Recall	AP
Scooter	15717	1333	638	92.2%	96.1%	89.4%
Sedan	20620	685	307	96.8%	98.5%	95.7%
s_Truck	392	16	2	96.1%	99.5%	97.7%
l_Truck	578	18	6	97.0%	99.0%	96.3%
Pedestrian	26	8	4	76.5%	86.7%	70.6%
Total	37333	2060	957	94.8%	97.5%	89.9%

TABLE 5. The detection result of PairingNet when training on 3 FPS.

	TP	FP	FN	Precision	Recall	AP
Scooter	16186	1405	161	92.0%	99.0%	92.9%
Sedan	20713	763	190	96.4%	99.0%	96.7%
s_ruck	384	76	10	83.81%	97.2%	92.1%
l_ruck	584	28	1	95.4%	99.8%	97.6%
Pedestrian	25	85	5	22.7%	83.3%	20.0%
Total	37902	2357	367	94.0%	99.0%	79.9%

One of the advantages of PairingNet is that it achieves tracking at different sampling rates. Table 5 shows the training results in 3 FPS. As can be seen in the table, each category exhibits a high recognition rate at both high and low sampling rates. In the data collection of this experiment, the amount of data for pedestrians is low and the frequency for pedestrians to appear in the images is low. As a consequence, the result of the low sampling rate for pedestrian detection is worse than that of the high sampling rate. If the amount of pedestrian information is increased, this issue should be solved.

In order to test the ability of generalization of the network on the training data set, the model is also trained by half of the training dataset, which is 6700 images, and the test data remained the same. Table 6 shows that most of the results are slightly worse than the training results of the original data. Particularly, the performance results of pedestrians and trucks have significantly reduced. The reason for this phenomenon is that the frequency that pedestrians and trucks appear in the original training data is very low. As a result, the reduction of the training data leads to insufficient training for these two categories. Therefore, a certain amount of training materials must be collected. Although the size of training data can be appropriately reduced for faster network training, it is still recommended to use more training data to achieve a better tracking performance.

PairingNet increases the input to three consecutive images and adds four additional output parameters. This may increase the inference time because of the additional network calculations. To evaluate the efficiency of the network under different sizes of input and output, the input of PairingNet is modified to take either one image, two consecutive images, and three consecutive images. In addition, the output of PairingNet is modified to be either with or without the four additional parameters.

TABLE 6. The detection result of PairingNet when training on 6700 images.

	TP	FP	FN	Precision	Recall	AP
Scooter	15329	2660	1018	85.2%	93.8%	85.6%
Sedan	20607	555	296	97.4%	98.6%	97.7%
s_ruck	383	257	11	59.8%	97.2%	70.6%
l_ruck	583	41	1	93.4%	99.8%	96.1%
Pedestrian	0	0	30	0.0%	0.0%	0.0%
Total	36902	3513	1356	91.3%	96.5%	70.0%

TABLE 7. Comparing inference time between different input and output in milliseconds.

	Full Time	Networks	Data Loading
YOLOv3	71.8	33.1	38.0
PairingNet 2 + 0	148.5	33.0	115.5
PairingNet 2 + 4	142.6	35.2	106.4
PairingNet 3 + 0	201.5	34.6	166.0
PairingNet 3 + 4	202.1	35.8	165.3
PairingNet 3 + 4 (with loading speedup)	79.2	35.5	44.0

Table 7 shows the inference time of YOLOv3 and PairingNet with different input and output settings. The first number following PairingNet is the number of consecutive input images, and the second number denotes the output with or without four additional parameters. The second column of Table 7 shows the total inference consuming time, and can be split into two processes, which are the network and the data loading. It can be seen that there is not much difference in the time required for the calculation of the *Networks*, but the time difference in *DataLoading* is large. Each additional image increases the loading time by approximately 70 milliseconds, which dramatically affects the overall computation speed. This explains that the network itself has little effect on the computation time. In order to speed up data loading, the input images need to be loaded in pipeline. That is, at each timestamp, the latest two input images of the previous timestamp are kept and only one additional image is loaded. In this way, except for the first two timestamps, the data loading time can be greatly reduced, which will bring our ParingNet close to the time consumption of the original YOLOv3. The results of PairingNet after the input pipeline implementation are shown in the last row.

2) PAIRING EVALUATION

In addition to the detection results, the performance of vehicle pairing prediction is shown in Table 8. We notice that the number of TP and Pair TP are the same, while Pair FP is zero. It means that if a vehicle is detected by PairingNet, then the pair of this vehicle will also be detected. The accuracy of predicting the past and future parameters is 99.9%, and the relatively important negative situation in the evaluation of the access status has also reached 100% prediction. In addition, the average error of the motion vector prediction is about 1.1 pixels, a very small gap, which shows that PairingNet

TABLE 8. PairingNet pairing result.

	TP	FN	Pair TP	Pair FP	Future Error	Acc
PairingNet	37555	525	37555	0	35	98.6%

TABLE 9. MOT result.

GT	FP	FN	MME	MOTA	MOTP
63985	1795	3320	362	91.6%	82.7%

has high vehicle motion estimation accuracy. PairingNet also greatly improves the accuracy of the vehicle pairing as it reaches 98.1%.

C. EXPERIMENT OF TRACKING ALGORITHM

Table 9 shows the result of the trajectory tracking algorithm using CLEAR MOT as the evaluation method. The Ground Truth (GT) shows the number of objects in the test data. The Mismatch Error (MME) is the number of the vehicles in different tracking id. In the table, PairingNet has as high as 82.7% MOTP accuracy on IoU and 91.6% MOTA accuracy. The results show that PairingNet method improves the performance of MOT.

VII. CONCLUSION AND FUTURE WORKS

This research proposes PairingNet which is a new multi-object tracking method. Different from the previous tracking algorithms that are based on object identification, PairingNet uses the features of multiple consecutive images to integrate the feature extraction and regression calculations of the original object detection network. PairingNet successfully adds the object tracking function into the YOLOv3 end-to-end object detection network and reduces the accumulated difference between object detection and trajectory prediction. In addition, because of the feature fusion and the extraction application, the calculation of trajectory prediction is no longer just the result of a single feature of speed and position but refers to a large number of related multiple features. The feature fusion of continuous images slightly improves the effect of object detection. The object detection mAP of PairingNet reached 96.4%, and the prediction of the pair position reached a success rate of nearly 100%. The overall computing time has also been improved to achieve the same real-time performance as YOLOv3. In terms of trajectory prediction, using Hungarian algorithm and the prediction of the pair position, CLEAR MOT's MOTP achieves an accuracy of 82.7% of the IoU standard, and MOTA also has an accuracy rate of 91.6%. PairingNet has demonstrated significant research results in real-time object tracking projects.

At present, the application environment of PairingNet uses drones to collect and analyze information on flat roads. In the future, we expect PairingNet to be applied to low angle cameras at intersections and to be transplanted to the analysis of highway monitoring cameras. In addition, although PairingNet has integrated the functions of trajectory prediction and object detection, multi-object tracking has additional operations for object pairing in addition to the aforementioned calculations. In the future, we will merge the object pairing operation with PairingNet to achieve the end-to-end operation of multi-object tracking.

ACKNOWLEDGMENT

This work is based on the master's thesis of Ming-Hong Lin, who received the M.S. degree from the Department of Computer Science, Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan.

REFERENCES

[1] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang, "Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 54–64, Jan. 2019.

[2] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 297–309, Feb. 2015.

[3] X. Li, J. Tan, A. Liu, P. Vijayakumar, N. Kumar, and M. Alazab, "A novel UAV-enabled data collection scheme for intelligent transportation system through UAV speed control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2100–2110, Apr. 2020.

[4] B. Coifman, M. McCord, R. Mishalani, and K. Redmill, "Surface transportation surveillance from unmanned aerial vehicles," in *Proc. 83rd Annu. Meeting Transp. Res. Board*, Washington, DC, USA, 2004, pp. 1–9.

[5] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, Mar. 2017.

[6] S. Wang, F. Jiang, B. Zhang, R. Ma, and Q. Hao, "Development of UAV-based target tracking and recognition systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3409–3422, Aug. 2020.

[7] S. Zhang, L. Chai, and L. Jin, "Vehicle detection in UAV aerial images based on improved YOLOv3," in *Proc. IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, Oct. 2020, pp. 1–6.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[9] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. Ik, and W.-C. Peng, "TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–8.

[10] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–8, Jan. 2004.

[11] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 5, nos. 1–10, p. 4, 2001.

[12] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3047–3055.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[14] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in *Proc. Workshop Motion Video Comput.*, 2002, pp. 169–174.

[15] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image Vis. Comput.*, vol. 22, no. 2, pp. 143–155, Feb. 2004.

[16] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[17] Z. Tang and J.-N. Hwang, "MOANA: An online learned adaptive appearance model for robust multiple object tracking in 3D," *IEEE Access*, vol. 7, pp. 31934–31945, 2019.

[18] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with TrackletNet," 2018, *arXiv:1811.07258*.

[19] W. Balid, H. Tafish, and H. H. Refai, "Intelligent vehicle counting and classification sensor for real-time traffic surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1784–1794, Jun. 2018.

[20] S.-L. Jeng, W.-H. Chieng, and H.-P. Lu, "Estimating speed using a side-looking single-radar vehicle detector," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 607–614, Apr. 2013.
[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.
[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[23] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
[24] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[25] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1995.
[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 84–90.
[27] Y. Liu, L. Zhang, Z. Chen, Y. Yan, and H. Wang, "Multi-stream Siamese and faster region-based neural network for real-time object tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 1–14, Jun. 2020.
[28] D. Tian, C. Lin, J. Zhou, X. Duan, Y. Cao, D. Zhao, and D. Cao, "SA-YOLOv3: An efficient and accurate object detector using self-attention mechanism for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 1–12, Dec. 2020.
[29] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[32] *Hero7 BLACK|GoPro*. Accessed: Jan. 26, 2023. [Online]. Available: https://gopro.com/en/us/update/hero7-black
[33] Irvine. *Video Annotation Tool From Irvine, California*. Accessed: Jan. 26, 2023. [Online]. Available: http://www.cs.columbia.edu/vondrick/vatic/

GUAN-WEN CHEN is currently pursuing the Ph.D. degree with the Department of Computer Science, Institute of Computer Science and Engineering, National Yang Ming Chiao Tung University. His research interests include intelligent transportation systems and computer vision.

MIN-TE SUN (Member, IEEE) received the B.Sc. degree from National Taiwan University, the M.Sc. degree from Indiana University, Bloomington, and the Ph.D. degree in computer and information science from The Ohio State University. He is currently a Professor with the Department of Computer Science and Information Engineering, National Central University, Taiwan. His research interests include distributed computing and the IoT. He is a member of the ACM.

TSÌ-UÍ İK (Member, IEEE) received the B.S. degree in mathematics and the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1991 and 1993, respectively, and the Ph.D. degree in computer science from the Illinois Institute of Technology, Chicago, IL, USA, in 2005.

He had been a Senior Research Fellow with the Department of Computer Science, City University of Hong Kong. He is currently a Professor with the Department of Computer Science and the Director of the Institute of Computer Science and Engineering, National Yang Ming Chiao Tung University. His research interests include intelligent applications, such as intelligent sports learning and intelligent transportation systems, mobile sensing, machine learning, deep learning, wireless sensor, and ad hoc networks.

Dr. İk was bestowed the Outstanding Young Engineer Award by the Chinese Institute of Engineers, in 2009, and the Young Scholar Best Paper Award by the IEEE IT/COMSOC Taipei/Tainan Chapter, in 2010. He received the Best Paper Award at ITST 2012. He received the three-year Outstanding Young Researcher Grant from the National Science Council, Taiwan, in 2012.