ICIC Express Letters
Volume **16**, Number **4**, April **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **381–389**

ROBUST PERSON TRACKING BY PAN-TILT CAMERA WITH LOW-COST SINGLE BOARD COMPUTER

Kohei Ogura, Noritaka Shigei* and Hiromi Miyajima

Graduate School of Science and Engineering
Kagoshima University
1-21-40 Korimoto, Kagoshima 890-0065, Japan
{ k0446511; k2356323 }@kadai.jp; *Corresponding author: shigei@eee.kagoshima-u.ac.jp

Received August 2021; accepted October 2021

Abstract. *This study constructs an accurate person tracking camera system by using inexpensive hardware. The system consists of a pan-tilt camera and a single board computer that controls it. YOLOv4 is used for person detection, and PID control is used for camera control. In this study, we propose a learning method for YOLOv4 to improve the identification accuracy of person and camera control methods to improve the tracking accuracy. The effectiveness of the proposed methods is demonstrated in the experiment.*

Keywords: Pan-tilt camera, Person tracking, Face recognition, YOLO, PID control, Single board computer

1. **Introduction.** In recent years, person tracking technology has been attracting attention as it is a very important technology for analyzing human behavior for crime control and marketing applications [1]. Further, surveillance cameras have been introduced in homes and small offices, and are expected to be actively used for various smartification. To realize it, it is desired to construct a high-performance system inexpensively and easily.

The pan-tilt camera is an attractive device that can cover a wide range and can be used for various smart applications. For person tracking, the pan-tilt camera requires to be appropriately controlled by using detection and control technologies. So far, person detection and tracking has been extensively studied. [3, 6] studied person (pedestrian) tracking for images taken with a fixed camera. Nikouei et al. [3] demonstrated that their introduced lightweight CNN (Convolutional Neural Network) is suitable for edge devices such as Single Board Computer (SBC), compared with traditional methods Haar-Cascade and HOG+SVM and deep learning methods SSD-GoogleNet and SSD-Mobilenet [7, 8]. [2, 4, 5] studied person or face tracking using a pan-tilt-zoom or pan-tilt camera. Kumar et al. [2] used a traditional tracking algorithm CAMSHIFT and a coarse 9-ways pan-tilt control. Zhu et al. [4] developed a human following wheeled robot with a pan-tilt camera, used YOLO (You Only Look Once) [9] for person detection, and proposed a tracking algorithm using FlowNet, which is CNN learning optical flow. Mian [5] proposed a face tracking algorithm adjusting the pan, tilt and zoom of a camera to capture a human face at the camera's maximum resolution. However, previous studies above have not considered person tracking with personal identification using a pan-tilt camera and a low-cost SBC.

The purpose of this study is to construct an accurate person tracking camera system with personal identification by utilizing deep learning and inexpensive hardware. The camera system is intended to be used indoors, such as in an small office, to record the behavior of a particular person. The system consists of a pan-tilt camera and an SBC that controls it. YOLOv4 is used for detection of individual faces and person body, and PID (Proportional Integral Differential) control is used for camera control. We propose a learning method for YOLOv4 to improve the identification accuracy of a person and

DOI: 10.24507/icicel.16.04.381

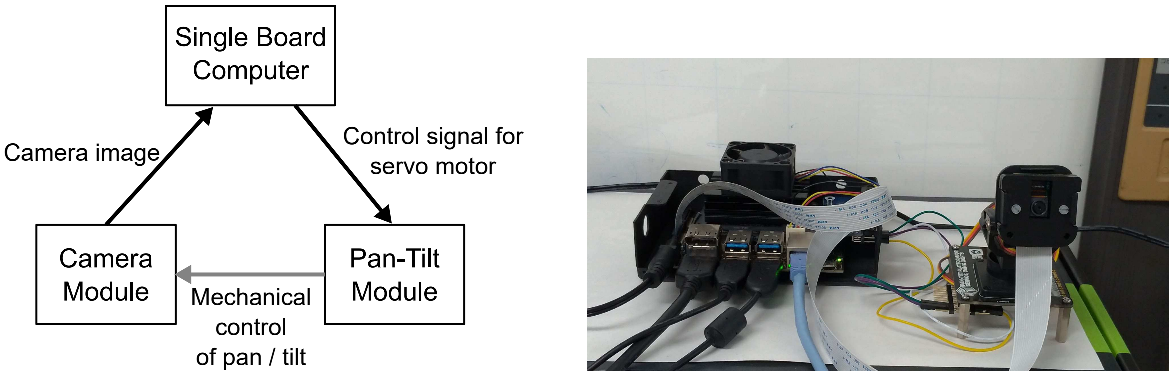
camera control methods such as hybrid control of PID and P to improve the tracking accuracy. The effectiveness of the proposed methods is demonstrated in the experiment. The rest of this paper is organized as follows. Section 2 describes the hardware configuration and element technologies of the tracking camera system used in this paper. Section 3 describes the proposed learning method for YOLOv4 used for person detection. Section 4 describes two types of the proposed person tracking algorithms for controlling a pan-tilt camera. Section 5 demonstrates the effectiveness of the proposed methods. Finally, Section 6 is the conclusion of this paper.

2. Person Tracking Camera System.

In this paper, we consider tracking a specific registered person with a pan-tilt camera. This camera system is supposed to be used indoors such as in an office to automatically record the history of events such as entering and leaving a room.

2.1. Hardware configuration.

Figure 1 shows the hardware configuration of the pan-tilt camera system used in this paper. The system consists of an SBC, a camera module and a pan-tilt module. NVIDIA Jetson Nano (4GB Memory), Raspberry Pi Camera V2 and Pimoroni Pan-Tilt HAT are used as SBC, camera module and pan-tilt module, respectively. Jetson Nano is equipped with 128-core GPU and is utilized for person recognition and person tracking processing. The camera is capable of shooting video at 1080p/30fps, 720p/60fps, or 480p/90fps. The pan-tilt module is capable of 180 degree pan/tilt motion on each axis by controlling two servo motors with PWM signals.


Single Board Computer
Camera image
Control signal for servo motor
Camera Module
Pan-Tilt Module
Mechanical control of pan / tilt

(a) A block diagram (b) A view of the entire system

FIGURE 1. Hardware configuration of pan-tilt camera system

2.2. Person tracking process.

Figure 2 shows the flow of the person tracking process in this paper. The target tracking task is continuously performed by repeating the series of processes (1) to (6).

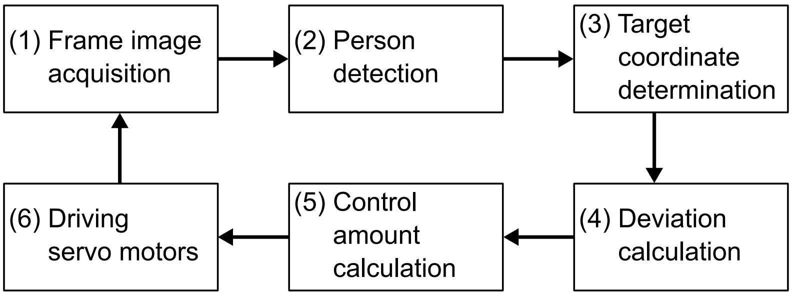

(1) Frame image acquisition
(2) Person detection
(3) Target coordinate determination
(4) Deviation calculation
(5) Control amount calculation
(6) Driving servo motors

FIGURE 2. Person tracking process

Process (1) extracts a frame image from the video image of the camera.

Process (2) detects the faces of particular individuals and the entire bodies of any persons. For each of detected bodies and faces, the system returns the coordinates and size of its bounding box and the probability of certainty of the body or each registered person face. YOLOv4 is used for the person detection.

Process (3) determines the target coordinates $(x_{\mathrm{tg}}, y_{\mathrm{tg}})$ for tracking, where $(x_{\mathrm{tg}}, y_{\mathrm{tg}})$ are coordinates on the image. $(x_{\mathrm{tg}}, y_{\mathrm{tg}})$ are ideally the center coordinates of the detected face or body of the target person. However, when the detection fails or returns multiple faces or bodies, the determination is not trivial. To solve this, several types of algorithms are presented in Section 4.

Process (4) calculates the deviations (e_x, e_y) between the target coordinates and the center coordinates $(x_{\mathrm{cnt}}, y_{\mathrm{cnt}})$ of the image.

$$\begin{cases} e_x = x_{\mathrm{cnt}} - x_{\mathrm{tg}} \\ e_y = y_{\mathrm{cnt}} - y_{\mathrm{tg}} \end{cases} \tag{1}$$

Process (5) calculates the control amounts (u_x, u_y), where u_x and u_y correspond to the pan angle and tilt angle, respectively. PID control or its variant is used for the calculation of u_x and u_y.

Process (6) controls servo motors for pan and tilt angles according to u_x and u_y.

2.3. **YOLOv4.** YOLOv4 [11] is used for person detection in process (2). YOLO [9] is a family of object detection algorithms using CNN and it can simultaneously detect and identify objects at high speed and with high accuracy. In the algorithm, an input image is divided into an $S \times S$ grid, each grid cell estimates the coordinates and confidence scores of B bounding boxes, and each grid cell predicts C conditional class probabilities, each of which is corresponding to one of C classes to be detected. Then, for each bounding box, the confidence score R_c of class c among C classes to be detected is given as the product of the confidence score of the box and the conditional class probability of class c at the box. YOLOv4 was developed after YOLOv3 and is 10%∼12% better than the accuracy of YOLOv3.

2.4. **PID control.** Let $u(t)$, $y(t)$, $r(t)$ and $e(t)$ be control, output, target and error (deviation) values at time t, respectively. Let K_{P}, K_{I} and K_{D} be proportional, integral and derivative gains, respectively. The control value $u(t)$ is determined by the following equation.

$$u(t) = K_{\mathrm{P}} e(t) + K_{\mathrm{I}} \int_0^t e(\tau) d\tau + K_{\mathrm{D}} \frac{de(t)}{dt} \tag{2}$$

The proportional control by term $K_{\mathrm{P}} e(t)$ makes the control value $u(t)$ proportional to the error $e(t)$ and it brings the response $y(t)$ closer to the target value $r(t)$. The integral control by term $K_{\mathrm{I}} \int_0^t e(\tau) d\tau$ makes $u(t)$ proportional to the integral of $e(t)$ over time and it reduces the residual error. The derivative control by term $K_{\mathrm{D}} \frac{de(t)}{dt}$ cancels out the influence of sudden disturbances by making $u(t)$ proportional to the magnitude of the change.

3. **Training of YOLOv4.** In this paper, the person detection process detects the following $C = C_{\mathrm{i}} + 1$ object classes: C_{i} individual classes $c_1, c_2, \ldots, c_{C_{\mathrm{i}}}$ and person class c_{p}. The target class for tracking is selected from C_{i} individual classes.

3.1. **Training data.** A training data is an image with bounding boxes of objects and their class labels. An example of training data is shown in Figure 3. For individual classes, the face part is specified by a bounding box labeled with the corresponding individual class. For the person class, a person's entire body is specified by a bounding box labeled as the person class.

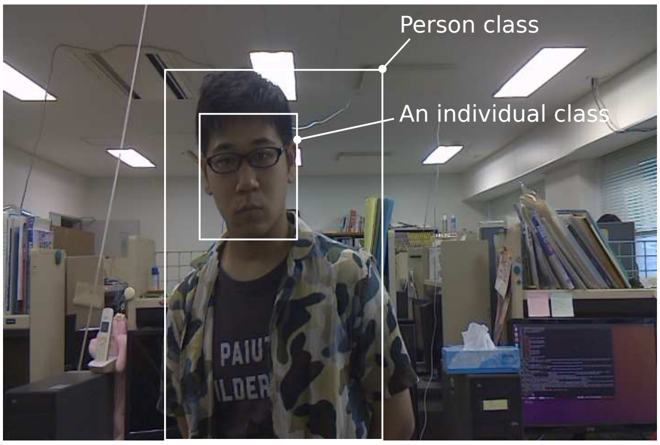


Figure 3. An example of training data

Two types of datasets A and B are used for training of YOLOv4. The images in dataset A are taken with different location or camera from where the person tracking is performed. The images in dataset B are taken at a location where the person tracking is performed. In general, tracking systems are desired to be easily deployed at various installation locations. The purpose of using two types of datasets is to reduce the number of collected images after installation by performing pre-training with dataset A before installation so that the system can be deployed quickly.

3.2. **Training algorithm.** The detection model of YOLOv4 pre-trained on the COCO dataset (referred to as pre-trained model) is fine-tuned by using datasets A and B. The learning algorithm is shown below.

Training Algorithm

Step 1: Perform fine-tuning of the pre-trained model by using dataset A. The fine-tuned model is referred to as initial trained model.

Step 2: Let the initial trained model perform the detection of images of dataset B.

Step 3: Let a subset of dataset B whose members (data) satisfy any of the following conditions be dataset B'.

- The detection of any individual class contained in the data is failed in Step 2.
- An individual class c not contained in the data is detected in Step 2, and its confidence score R_c exceeds threshold θ.

Step 4: Let a subset of dataset A whose members (data) are annotated with any individual class be dataset A'.

Step 5: Perform fine-tuning of the initial trained model by using the union data of datasets A' and B'. The resulting model is the final trained model.

4. Person Tracking Algorithms.

4.1. **Algorithm 1.** This algorithm has two options. One is whether to track a face or a body, and the other is whether to use PID control or hybrid control of PID and P.

In this section, two types of tracking algorithms, Algorithms 1 and 2, are presented. Algorithm 1 considers two cases: one is to track a face, and the other is to track a body. It is specified in advance whether to track a face or a body. Face tracking is suitable for tracking specific individuals, but the face is often undetected, and tracking becomes unstable. On the other hand, body tracking is stable when only the tracked person is visible, but it is necessary to find the target person when multiple persons are visible. However, Algorithm 1 does not consider body tracking when multiple persons are detected. Algorithm 2 tracks a body and uses the detected face to identify the body to track.

The tracking stabilization measures introduced in the two algorithms are as follows: 1) how to determine the target coordinates when any face or body is not detected, and 2) hybrid control of PID and P.

In the description of the algorithms, the following is assumed. Let the size of the frame image be $W \times H$. Let $\left(x_{\rm tg}^{-}, y_{\rm tg}^{-}\right)$ be the target coordinates at the previous time step. If the target person is detected, let $(x_{\rm tp}, y_{\rm tp})$ be the center coordinate of the target person's face. For $i = 1, 2, \ldots, N_{\rm pc}$, let $\left(x_{\rm pc}^{(i)}, y_{\rm pc}^{(i)}\right)$ be the center coordinate of the ith entire body detected as the person class, where $N_{\rm pc}$ is the number of bodies detected as the person class.

Condition. Specify the target class $c_{\rm tgt}$ among individual classes $c_1, c_2, \ldots, c_{C_{\rm i}}$ and $c_{\rm p}$. If $c_{\rm tg}$ is an individual class, then the face detected as class $c_{\rm tg}$ is tracked. Otherwise, one of the detected bodies is tracked. Specify the control strategy from PID control and hybrid control of PID and P.

Step 1 (Target coordinates determination). If no object with class $c_{\rm tg}$ is detected, $(x_{\rm tg}, y_{\rm tg}) \leftarrow \left(x_{\rm tg}^{-}, y_{\rm tg}^{-}\right)$. Otherwise, the center coordinates of the detected object of class $c_{\rm tg}$ are selected as the target coordinates $(x_{\rm tg}, y_{\rm tg})$, where if there are multiple objects of class $c_{\rm tg}$ then the object with the maximum confidence score is selected.

Step 2 (Control value determination). The deviations e_x and e_y are calculated by using Equation (1). If the control strategy is PID, then the control values u_x and u_y are calculated by PID control of Equation (2) with $K_{\rm P}, K_{\rm I}, K_{\rm D} \neq 0$. Otherwise, the hybrid of PID and P is performed as follows: If the target coordinates $(x_{\rm tg}, y_{\rm tg})$ are located in the area of size $W/4 \times H/4$ in the center of the image as shown in Figure 4. The control values u_x and u_y are calculated by PID control of Equation (2) with $K_{\rm P}, K_{\rm I}, K_{\rm D} \neq 0$. Otherwise, the control values u_x and u_y are calculated by P control of Equation (2) with $K_{\rm P} \neq 0$ and $K_{\rm I}, K_{\rm D} = 0$.

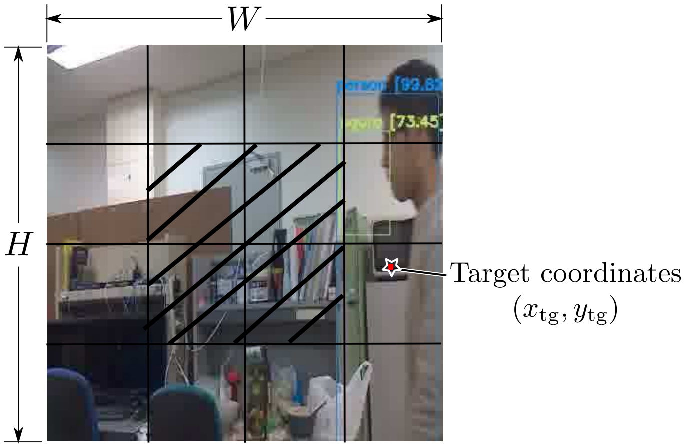


Figure 4. Target coordinates

Step 3 (Servo motors' angles determination). Let θ_x^{-} and θ_y^{-} be the servo motors' angles at the previous time step. Let θ_x^{+} and θ_y^{+} be the servo motors' angles at the next time step. The servo motors' angles θ_x and θ_y are calculated based on u_x and u_y. If θ_x and θ_y exceed the movable range or the target person has not been detected for more than $T_{\rm to}$ seconds, $\left(\theta_x^{+}, \theta_y^{+}\right) \leftarrow \left(\theta_x^{-}, \theta_y^{-}\right)$. Otherwise, $\left(\theta_x^{+}, \theta_y^{+}\right) \leftarrow (\theta_x, \theta_y)$.

4.2. **Algorithm 2.** Algorithm 2 always tracks a body and determines the tracked body by using the detected face of the target person. As methods for determining the body to track when the target face is not detected, the following two strategies are considered: coordinates-based and control-value-based. Which strategy to use is specified in advance. The hybrid control of PID and P is used for calculating the control values.

Condition. Specify the Target Coordinates Determination (TCD) strategy from coordinates-based and control-value-based.

Step 1 (Target coordinates determination). If no body with the person class is detected, $(x_{\mathrm{tg}}, y_{\mathrm{tg}}) \leftarrow \left(x_{\mathrm{tg}}^{-}, y_{\mathrm{tg}}^{-}\right)$. Otherwise, the center coordinates $\left(x_{\mathrm{pc}}^{(i_{\mathrm{tp}})}, y_{\mathrm{pc}}^{(i_{\mathrm{tp}})}\right)$ of the i_{tp}th detected body are selected as the target coordinates $(x_{\mathrm{tg}}, y_{\mathrm{tg}})$, where for coordinates-based TCD strategy,

$$i_{\mathrm{tp}} = \begin{cases} \underset{i \in \{1,2,\ldots,N_{\mathrm{pc}}\}}{\arg\min} \left(\left(x_{\mathrm{tp}} - x_{\mathrm{pc}}^{(i)}\right)^2 + \left(y_{\mathrm{tp}} - y_{\mathrm{pc}}^{(i)}\right)^2 \right) & \text{if target person is detected,} \\ \underset{i \in \{1,2,\ldots,N_{\mathrm{pc}}\}}{\arg\min} \left(\left(x_{\mathrm{tg}}^{-} - x_{\mathrm{pc}}^{(i)}\right)^2 + \left(y_{\mathrm{tg}}^{-} - y_{\mathrm{pc}}^{(i)}\right)^2 \right) & \text{otherwise} \end{cases} \tag{3}$$

and for control-value-based strategy,

$$i_{\mathrm{tp}} = \begin{cases} \underset{i \in \{1,2,\ldots,N_{\mathrm{pc}}\}}{\arg\min} \left(\left(x_{\mathrm{tp}} - x_{\mathrm{pc}}^{(i)}\right)^2 + \left(y_{\mathrm{tp}} - y_{\mathrm{pc}}^{(i)}\right)^2 \right) & \text{if target person is detected,} \\ \underset{i \in \{1,2,\ldots,N_{\mathrm{pc}}\}}{\arg\min} \left(\left(u_x^{-} - u_x^{(i)}\right)^2 + \left(u_y^{-} - u_y^{(i)}\right)^2 \right) & \text{otherwise} \end{cases} \tag{4}$$

and $u_x^{(i)}$ and $u_y^{(i)}$ are the control values calculated by using the method of Step 2 and ith detected body.

Step 2 (Control value determination). This step is the same as for the hybrid control of PID and P in Algorithm 1.

Step 3 (Servo motors' angles determination). This step is the same as for Algorithm 1.

5. **Experimental Evaluation.**

5.1. **Experimental conditions.**

YOLOv4-tiny [13], a simplified model of YOLOv4, is used for the detector. Its pre-trained model is trained by using MS COCO 2017 dataset [14], having 80 classes, including person class. The datasets A and B consist of 2,815 and 990 images. The images show 12 persons and are labeled with three individual classes (for face) c_1, c_2, c_3 and one person class (for body) c_{p}. The size of input images for YOLOv4 is $W = H = 416$. 20% of the data was used for validation and 80% of the data was used for training. In training, the batch size is 64 and the total number of batches is 8,000. $\theta = 0.8$, $T_{\mathrm{to}} = 2.0$ sec., for pan $K_{\mathrm{P}} = 0.05$, $K_{\mathrm{I}} = 0.0015$ and $K_{\mathrm{D}} = 0.001$, and for tilt $K_{\mathrm{P}} = 0.03$, $K_{\mathrm{I}} = 0.0015$ and $K_{\mathrm{D}} = 0.001$.

5.2. **Experiment 1.**

Before evaluating our proposed training and tracking algorithms, in this experiment, the pre-trained model of YOLOv4-tiny used in this paper is compared with other pre-trained detector models of YOLOv3-tiny, SSD-Mobilenet-v2, and SSD-Inception-v2. YOLOv3-tiny is a simplified model of YOLOv3, which is the previous version of YOLOv4, and the training data of its pre-trained model and its input image size are the same as for YOLOv4-tiny. SSD (Single Shot MultiBox Detector) [7] is another object detection model, MobileNet v2 is lightweight architecture for implementing CNN [8], and Inception is an architecture implementing convolutions with low computational cost while keeping high quality [15]. For SSD-Mobilenet-v2 and SSD-Inception-v2, their input image size is 300×300 and their pre-trained models are trained by using MS COCO dataset [14], having 91 classes, including person class.

The test dataset consists of 50 images taken at the same location as dataset B, and all the images contain only person class labels. In the dataset, 37, 8, 3 and 2 images in the test dataset contain 1, 2, 3 and 4 person class labels, respectively.

Table 1 shows the experiment result of the recall, F1-score, and computation time, where the computation time is the processing time per image. According to the result, YOLOv4-tiny is the most accurate, and the calculation time is almost the same as YOLOv3-tiny, which is the fastest. Therefore, among the four models, YOLOv4-tiny is considered the best model for the system of this paper.

Table 1. Accuracy and computation time for detection models

Model	Recall	F1	Time [s]
YOLOv4-tiny	0.871	0.931	0.143
YOLOv3-tiny	0.686	0.814	0.133
SSD-Mobilenet-v2	0.643	0.783	0.973
SSD-Inception-v2	0.743	0.853	1.175

5.3. **Experiment 2.** In this experiment, the accuracy of fine-tuned detection models is evaluated. The model fine-tuned by the training algorithm presented in Subsection 3.2 is compared with the model fine-tuned by using only dataset A. The test dataset contains 65, 14, 23 and 810 objects for classes c_1, c_2, c_3 and c_{p}. The detection threshold is set to 0.5 for all classes. Table 2 shows the number of data for each class in datasets A, B, A' and B'. Precision, recall and F1 score for the test dataset are shown in Table 3. According to the result, the proposed training method improves F1 scores for all classes and recall rates for all classes except for c_2. The low improvement of c_2 is considered due to a small number of training data, as shown in Table 2.

Table 2. Number of data for each class in datasets

Class	DS A	DS B	DS A'	DS B'
c_1	701	78	701	50
c_2	701	21	701	12
c_3	701	46	701	33
c_{p}	2815	1372	2103	550

Table 3. Accuracy of fine-tuned detection model

(a) For using only dataset A

	c_1	c_2	c_3	c_{p}	Mean
Precision	0.11	0.21	0.01	1.00	0.33
Recall	0.43	0.50	0.04	0.78	0.44
F1	0.17	0.29	0.01	0.88	0.34

(b) For training algorithm in 3.2

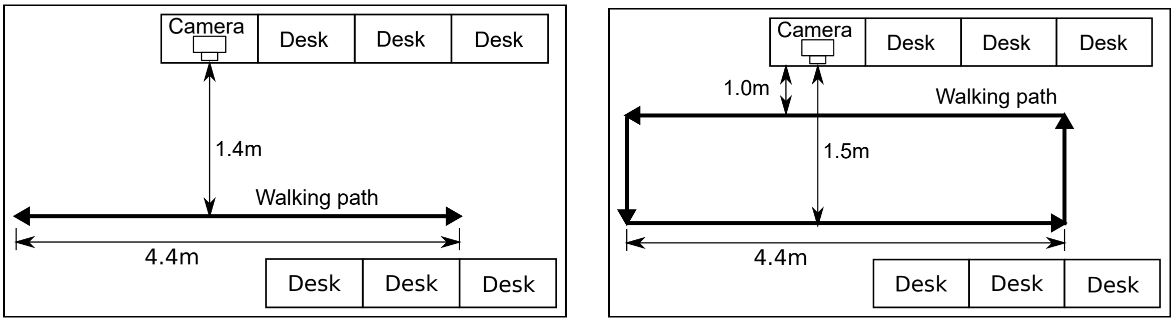
	c_1	c_2	c_3	c_{p}	Mean
Precision	0.78	0.86	0.46	1.00	0.77
Recall	0.85	0.43	0.87	0.92	0.77
F1	0.81	0.57	0.60	0.96	0.73

5.4. **Experiment 3.** This experiment evaluates Algorithm 1 from the following perspectives: 1) which is better to track, a face or a body, and 2) which is better, PID control or hybrid control of PID and P.

In the experiment, there is only one person in the shooting range of the camera, and one trial is to track a person who walks around the walking route shown in Figure 5(a) for 3 minutes. For body and face to be tracked, person class c_{p} and individual class c_1 are used, respectively. The accuracy of tracking is evaluated in terms of the following Target Tracking Rate (TTR):

$$TTR = \frac{N_{\mathrm{suc}}}{N} \tag{5}$$

where N is the number of total frames and N_{suc} is the number of frames showing the target person's face.



(a) For one-person walking

(b) For two-person walking

Figure 5. Walking paths

Table 4 shows the TTR of three trials for four cases where the tracking target (face or body) and the control method (only PID or hybrid of PID and P) are different. Body tracking is very stable and face tracking often fails. This is because the detection rate of the individual class is lower than that of the person class. Regarding control methods, the hybrid of PID and P controls is better than using only PID control. In PID control, the control amount related to D control becomes very large when the state changes from undetected to detected. It is considered that the hybrid control makes this phenomenon less likely to occur.

Table 4. Target tracking rate by Algorithm 1 for one-person walking case

Target	Control	TTR			
		Trial 1	Trial 2	Trial 3	Mean
Body	PID	1.000	1.000	1.000	1.000
	PID&P	1.000	1.000	1.000	1.000
Face	PID	0.397	0.233	0.262	0.297
	PID&P	0.324	0.373	0.552	0.416

5.5. **Experiment 4.** Algorithm 2 is evaluated for two-person walking cases, where the target person and a non-target person are walking in front of the camera. Unlike Algorithm 1, Algorithm 2 tracks only the body of specified person. Further, when the face of the target person is not detected, Algorithm infers the body to track by either of coordinates-based or control-value-based. It is examined which strategy is better.

In the experiment, the two persons walk around the walking route shown in Figure 5(b), and the one trial is to track the target person for 3 minutes. The target person is of individual class c_1 and the non-target person is one of the six persons: two persons of individual classes c_2 and c_3, two persons of non-individual class in datasets A and B and two persons not in datasets A and B. The accuracy is evaluated in terms of TTR of Equation (5).

Table 5 shows the TTR of three trials for six cases where the non-target persons are different. Trials with $TTR > 0.95$ are mostly success and track the target throughout the trial. As TTR decreases, the non-target is often tracked. When two persons pass each other the tracking targets are switched, and for trials with low TTR it takes a long time to track the wrong person. According to the result, the control-value-based strategy has four trials with a particularly low TTR of $TTR < 0.8$ and on the other hand the coordinates-based strategy has only one trial. Therefore, the coordinates-based strategy is more resistant to passing each other and is superior to the control-value-based strategy. Regarding the types of non-target, the control-value-based strategy is particularly inaccurate for individual classes, and the coordinates-based strategy is slightly less accurate for non-individual class's persons in training datasets.

Table 5. Target tracking rate of Algorithm 2 for two-person walking case with target person of class c_1 and non-target person

Non-target	Control-value-based					Coordinates-based				
	Trial 1	Trial 2	Trial 3	Mean		Trial 1	Trial 2	Trial 3	Mean	
Person of class c_2	0.969	0.655	0.775	0.800	0.778	0.997	0.989	0.989	0.992	0.983
Person of class c_3	0.978	0.617	0.675	0.757		0.988	0.951	0.984	0.983	
Person A in DSs A&B	0.952	0.963	0.828	0.914	0.906	0.952	0.993	0.951	0.965	0.914
Person B in DSs A&B	0.842	0.933	0.919	0.898		0.678	0.920	0.991	0.863	
Person C not in DSs A&B	0.936	0.921	0.922	0.926	0.942	0.981	0.984	0.969	0.978	0.986
Person D not in DSs A&B	0.935	0.971	0.964	0.957		0.993	0.990	0.997	0.993	

6. **Conclusions.** This paper develops an accurate person tracking camera system by utilizing YOLO and Jetson Nano. The fine-tuning algorithm for YOLO and camera control methods are proposed. The effectiveness of the proposed methods is demonstrated in one-person and two-person walking cases. The developed system can be used to record employee behavior in a small office. Future works include improving accuracy when wearing a mask or when three or more persons are photographed simultaneously.

Acknowledgment. This work is supported by JSPS KAKENHI Grant Number 20K119 94.

REFERENCES

[1] R. Oami et al., A recent trend of image analysis technologies for crime prevention and marketing applications, *The Journal of the Institute of Image Information and Television Engineers*, vol.70, no.1, pp.63-68, 2016 (in Japanese).

[2] P. Kumar, A. Dick and T.-S. Sheng, Real time target tracking with pan tilt zoom camera, *2009 Digital Image Computing: Techniques and Applications*, pp.492-497, 2009.

[3] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B. Choi and T. Faughnan, Smart surveillance as an edge network service: From Harr-Cascade, SVM to a lightweight CNN, *Int. Conf. on Collaboration and Internet Computing*, pp.256-265, 2018.

[4] Z. Zhu, H. Ma and W. Zou, Human following for wheeled robot with monocular pan-tilt camera, *arXiv.org*, arXiv: 1909.06087, 2019.

[5] A. Mian, Realtime face detection and tracking using a single Pan, Tilt, Zoom camera, *Int. Conf. Image and Vision Computing New Zealand*, pp.1-6, 2008.

[6] L. Wang, J. Gui, Z.-M. Lu and C. Liu, Fast pedestrian detection and tracking based on ViBe combined HOG-SVM scheme, *International Journal of Innovative Computing, Information and Control*, vol.15, no.6, pp.2305-2320, 2019.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, SSD: Single shot multibox detector, *arXiv.org*, arXiv: 1512.02325, 2015.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, *arXiv.org*, arXiv: 1704.04861, 2017.

[9] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You only look once: Unified, real-time object detection, *arXiv.org*, arXiv: 1506.02640, 2015.

[10] NVIDIA, DATA SHEET *NVIDIA Jetson Nano System-on-Module*, v 1.0, 2020.

[11] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, *arXiv.org*, arXiv: 2004.10934, 2020.

[12] A. Rosebrock, Pan/tilt face tracking with a Raspberry Pi and OpenCV, *Pyimagesearch*, https://www.pyimagesearch.com/2019/04/01/pan-tilt-face-tracking-with-a-raspberry-pi-and-opencv/, Accessed on Nov. 10, 2021.

[13] C.-T. Wang, A. Bochkovskiy and H.-Y. M. Liao, Scaled-YOLOv4: Scaling cross stage partial network, *arXiv.org*, arXiv: 2011.08036, 2020.

[14] *COCO – Common Objects in Context*, https://cocodataset.org/, Accessed on Nov. 10, 2021.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe and J. Shlens, Rethinking the inception architecture for computer vision, *arXiv.org*, arXiv: 1512.00567, 2015.