Hindawi
Journal of Robotics
Volume 2023, Article ID 5525744, 12 pages
https://doi.org/10.1155/2023/5525744

Research Article

An Improvement of the Camshift Human Tracking Algorithm Based on Deep Learning and the Kalman Filter

Van-Truong Nguyen, Duc-Tuan Chu, Dinh-Hieu Phan, and Ngoc-Tien Tran

Department of Mechatronics Engineering, Hanoi University of Industry, Hanoi 11900, Vietnam

Correspondence should be addressed to Van-Truong Nguyen; nguyenvantruong@haui.edu.vn

Received 21 January 2023; Revised 3 March 2023; Accepted 9 March 2023; Published 21 March 2023

Academic Editor: Keigo Watanabe

Automated human tracking in real time has been applied in many areas such as security, surveillance, traffic control, and robots. In this paper, an improvement of the Camshift human tracking algorithm based on deep learning and the Kalman filter is proposed. To detail an approach by using YOLOv4-tiny to detect a human in real time, Camshift is used to track a particular person and the Kalman filter is applied to enhance the performance of this algorithm in case of occlusion, noise, and different light conditions. The experiments show that the combination of YOLOv4-tiny and the improved Camshift algorithm raises the standard of speed as well as robustness. The proposed algorithm is suitable for running in real time and adapts well to the same color and different light conditions.

1. Introduction

In recent years, object tracking has become a field that has a lot of research and development. Object tracking is applied to track the movement of target objects, such as human tracking, moving target tracking, security, and traffic surveillance. Particularly, human tracking is often used in real life with many applications such as robot vision, traffic monitoring, education, and so on. Any human tracking algorithms need to solve detection and tracking problems across multiple frames [1]. However, the drawbacks of algorithms are neither high speed nor high accuracy. In addition, the ability to work in complex backgrounds, light sources, and many other different conditions is also a challenge that attracts many researchers.

More recently, a lot of deep-learning approaches have been developed to boost accuracy. The merit of this approach is automating feature extraction, high flexibility, and detecting objects with high accuracy. This approach needs a lot of data to train as well as improve accuracy. For example, YOLO [2] has been the best object detection model recently. This algorithm has both high speed and high accuracy. Nevertheless, YOLO needs high-performance devices to ensure computation complexity and training network as well. To reduce parameters and optimizer the network structure, YOLO-tiny is created. This YOLO version is suitable for mobile, embedded devices without changing too much accuracy. R-CNN [3], Fast R-CNN [4], and Faster R-CNN [5] are good detection models. Yet, due to a two-stage network, it is slow on mobile devices. ResNet [6] and VGG [7] are well-known models. The structure of these models is still applied in the later deep learning models. The disadvantage is that the number of parameters is too much, leading to difficulties in the training and real time as well.

To track the target human, the information of the targets is extracted by a detection algorithm that is then handled by tracking algorithms. Tracking algorithms predict target human motion based on color, contour information, and so forth. For instance, Meanshift [8] is the fast-tracking algorithm. This method uses the color feature to determine the target and has many strengths in case of low cost computation or need the robust and effective algorithm, but Meanshift proved ineffective when noise such as background and full occlusion by other objects. SORT [9] is an effective and easily implemented algorithm. However, SORT is left

withsome problems with linear Kalman filters and ID switches. The Camshift [10] algorithm tracks the target object based on the original color. To be more specific, the Camshift algorithm tracks an object by applying the Meanshift method to a region of interest in the image called the "histogram backprojection." Histogram backprojection is based on a histogram model of a feature. This model then is used to search for a similar feature in an image. Camshiftis an efficient algorithm, with high processing, and is easy to implement. Due to the development based on the Meanshift algorithm, the algorithm is easily influenced by background colors or textures.

To improve the disadvantage of similar color or complex context, a lot of methods are proposed to improve the accuracy of target tracking such as the point-based Kanade Lukas Tomasi colour-filter (PKLTF) [11] is a single-target tracker which is constructed based on the Kanade Lukas Tomasi approach and Meanshift algorithm. It is a robust method that can deal with high appearance changes in the target, occlusion, and loss of the target while the tracking process. Many studies presented the solution of combining many features to increase accuracy. Target Tracking Scale Adaptive [12] is based on combining local binary pattern operator and nonlinear Meanshift. In recent years, we witnessed many applications of correlation filters [13–15]. There are lots of advantages such as dealing with high-speedreal-time tracking, scale, rotation, and partial occlusion. High-speed KCF tracker [16] is concerned with the adaptive ability to the problems of scale and obscurity, besides the incredible tracking speed of up to 170 fps. A particle swarm tracking algorithm is based on the increasing inertia weight PSO [17] or the method using color features [18]. The algorithm has the advantage of global tracking, so it can deal with shortcomings as the tracker falls into the local problem. Nevertheless, when the target is fully obscured, the tracker will get lost of the tracking target. YOLOv3-Camshift [19] is an effective and robust tracking algorithm. This ability comes from the high performance of YOLOv3 in object detection and the high speed of Camshift. However, the drawback of Camshift is its high sensitivity with similar colors and light changes. Hence, in complex conditions, the tracking result is often moving around the target. The Kalman filter-Camshift algorithm [20] is an algorithm developed to improve the disadvantage of Camshift. It has good performance and high tracking speed. The Kalman filter provides an efficient method to find the target object and overcome the defect of occlusion in the process of tracking a moving object. Nevertheless, when the target is occluded or changes shape significantly, both Camshift and Kalman filter predict the object's position incorrectly. Moreover, when the target object changes direction or speed, the Kalman filter is easily drifted over time.

In this paper, a human tracking system based on deep learning and improved Camshift is proposed. YOLOv4-tiny is combined with Camshift and Kalman filter [21] to increase the effectiveness in tracking the target human. YOLOv4-tiny is used to detect, and then the Camshift–Kalman filter tracks the target human. The contributions of the paper are shown as follows:

(1) Improving the accuracy of the human tracking system and usingYOLOv4-tiny improve the detection accuracy as well as the ability to work in real time.

(2) By solving the problem of occlusion or noise in the Camshift algorithm, the Kalman filter makes good Camshift'shortcomings. The Kalman filter increases tracking precisely when the object disappears or part occlusion.

(3) The experimental results show that the proposed algorithm adapters help in tracking humans with real-time speed (12 fps). The system also solves full occlusion, complex color, and different light conditions.

In this section, we reviewed the merit and drawbacks of deep learning, tracking algorithms, and the combination of them to overcome that weak point. In Section 2, the structure of YOLOv4-tiny and the concept of tracking human algorithms are provided. The result and discussion of the proposed method will be shown in Section 3. The conclusions are shown in the last section.

2. Materials and Methods

2.1. System Architecture. The proposed algorithm is shown in Figure 1. Firstly, an initial search window is initialized manually from the first frame. The ROI region is created and then converted to HSV color space. A weighted Gaussian mask is applied to reduce the influence of the background noises. H channel is used to establish the histogram of ROI. Subsequently, in the next frame, YOLOv4-tiny is utilized to detect the target human. If humans are detected, the foreground is extracted which shows the best candidate objects. Otherwise, the Kalman filter is applied to predict a new target position based on the information on the target's motion in the previous frame. Next, the Camshift algorithm determines the center of distribution density. The Kalman filter calculates the new predicted position of the target. A position deviation is the distance of the candidate target being determined by the Camshift and Kalman filters. Bhattacharyya distance is calculated to compare similar distribution densities. If the position deviation is smaller than a threshold and the Bhattacharyya distance is also smaller than certain thresholds, the position of the target is determined at the midpoint of the Kalman filter and Camshift. We repeat the process by starting from the human detection step until the last frame.

2.2. YOLOv4-Tiny. YOLOv4-tiny is a small version of YOLOv4. The primary difference between YOLOv4-tiny and YOLOv4 is that the network size is dramatically reduced, and the number of convolutional layers in the CSP Darknet backbone is compressed [2]. Figure 2 shows the network structure of YOLOv4-tiny.

YOLOv4-tiny support input image size is 416×416 and uses the CSPDarknet53-tiny network as a backbone [2]. Three CSP Block is used to extract feature maps with

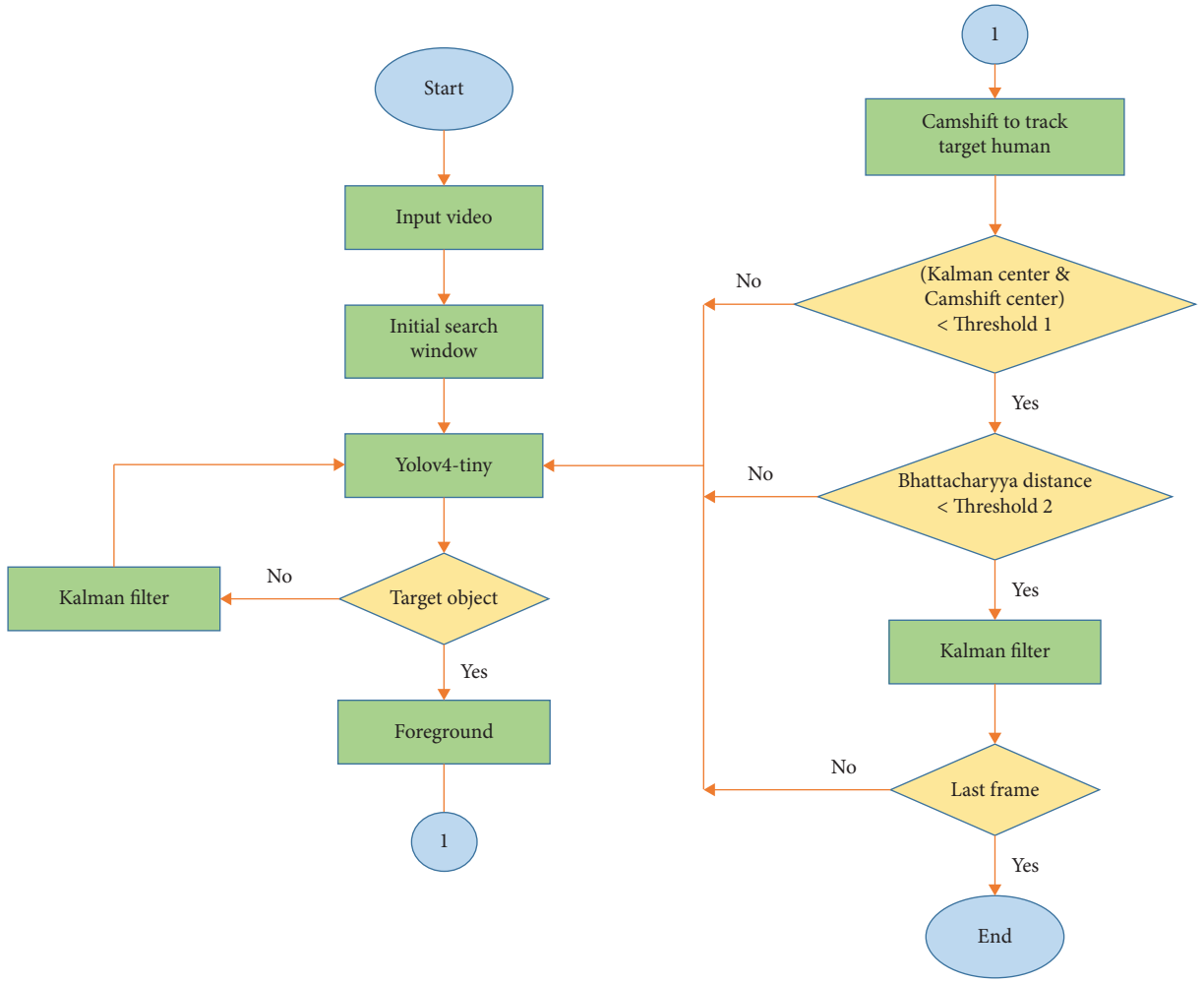


Figure 1: The flow chart of the proposed algorithm.

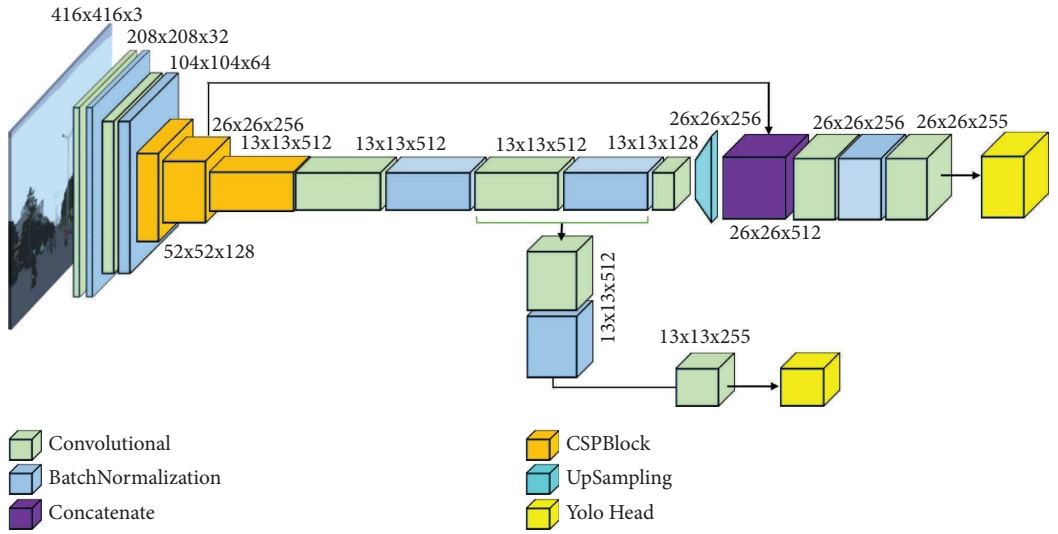


Figure 2: YOLOv4-tiny structure.

different scales [22]. Following each convolutional layer is Batch Normalization and Leaky ReLU activation that reduces the amount of calculation and improves the accuracy. YOLOv4-tiny has two scale feature maps 13×13 and 26×26. YOLO4-tiny has two YOLO heads; YOLO head used the feature map obtained by FPN to make the final prediction.

This structure of YOLOv4-tiny helps reduce computation while maintaining the accuracy of the network [2]. Figure 3 shows the structure of CSPBlock.

When training YOLOv4-tiny, calculating the loss function is very important. Loss functions represent the learning of the model. Given an image, the task of YOLOv4-tiny is to return bounding box coordinates and the name (class) of the object that is needed to detect. The model divides the feature map into $S \times S$ grids, each grid generates B bounding box. Therefore, $S \times S \times B$ bounding boxes are created for the input image. The center of an object lies in some grid; the bounding boxes in this grid will predict the object. YOLO used the feature map obtained by FPN to make the final prediction. To evaluate the algorithm, precision-recall is represented by equations (1) and (2).

$$\text{precision} = \frac{\text{true positive}}{(\text{true positive} + \text{false positive})}, \tag{1}$$

$$\text{recall} = \frac{\text{true positive}}{(\text{true positive} + \text{false negative})}. \tag{2}$$

When training models, the loss function consists of the loss of the predicted center coordinate (loss_1), the loss of the prediction bounding box (loss_2), the loss of the predicted class (loss_3), and the loss of predicted confidence (loss_4). The formula is shown as follows [23]:

$$\begin{aligned}
\text{loss} &= \text{loss}_1 + \text{loss}_2 + \text{loss}_3 + \text{loss}_4,\\
L &= \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \beta_{ij}^{obj} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right]^{ij}\\
&\quad + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \beta_{ij}^{\text{obj}} \left[\left(w_i - \hat{w}_i\right)^2 + \left(h_i - \hat{h}_i\right)^2 \right]\\
&\quad + \sum_{i=0}^{S^2} \beta_{ij}^{\text{obj}} \sum_{c \in \text{class}} \left(p_i(c) - \hat{p}_i(c)\right)^2\\
&\quad + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \beta_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \beta_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2,
\end{aligned} \tag{3}$$

where x_i, y_i are the positions of the prediction bounding box, $\hat{x}_i$, $\hat{y}_i$ are the actual positions reached from training data, w_i, h_i are the width and height of the predicted bounding box, $\hat{w}_i, \hat{h}_i$ are the actual width and height reached from training data, $C_i, \hat{C}_i$ are the confidence score, the intersection part of the predicted bounding box, and the actual box, respectively, β_{ij}^{obj} confirm whether the j prediction bounding box in a cell i has an object, λ_{coord} is a factor to adjust the prediction loss of the prediction box, and λ_{noobj} is a factor to adjust when no target is there in the single grid.

2.3. Camshift Algorithm. Camshift is a nonparametric algorithm and is developed based on Meanshift [24]. Camshift does not need to train such as deep learning models, instead Camshift tracks objects based on the original color of the object. This leading is difficult to track objects if the color or texture changes a lot. The main difference between Camshift and Meanshift is every frame Camshift recomputes automatically the size and rotation of the searching window and measures the distance coefficient by using the Bhattacharyya parameter [25]. This algorithm is an effective method of seeking the local maximum in the density distribution in a stable size and rotation of the window per frame's video. Figure 4 shows the flowchart of Camshift.

Camshift converts input images from RGB color into HSV color, due to Camshift's ease of sensitivity with noises. The result of converting is three components being independent, which comprises H (Hue), S (Saturation), and V (Value) in the HSV space [26]. The H channel makes the color histogram. Then, calculating the color probability distribution map shows the appearance probability of each pixel.

The zero-order and the first-order distance are defined as follows [26, 27]:

The zero order

$$M_{00} = \sum_x \sum_y I(x, y). \tag{4}$$

The first order

$$\begin{aligned}
M_{10} &= \sum_x \sum_y xI(x, y),\\
M_{01} &= \sum_x \sum_y yI(x, y).
\end{aligned} \tag{5}$$

The centroid of the moving target is defined by

$$\begin{aligned}
x_c &= \frac{M_{10}}{M_{00}},\\
y_c &= \frac{M_{01}}{M_{00}}.
\end{aligned} \tag{6}$$

The size of the search window is

$$s = \sqrt{\frac{M_{00}}{256}}; \text{ the length is 1.2s.} \tag{7}$$

The size of the search window is adjusted by using the value of M_{00} [26]. The center of the search box is moved to the location of the centroid mass. If the window moving distance exceeds a certain threshold, the centroid position and size of the search window are recalculation [28]. This calculation step is repeated until the center position of the search window and center of mass are smaller than the threshold. The search window size and position are used as initialization values in the next frame.

2.4. Kalman Filter. The Kalman filter is a typical tracking algorithm. The Kalman filter uses a series of measurement values, which are affected by noise or error, to estimate the target position in order to increase accuracy compared to the use of only one measurement value. The Kalman filter minimizes the error of the estimation such as position and velocity in the next frame.

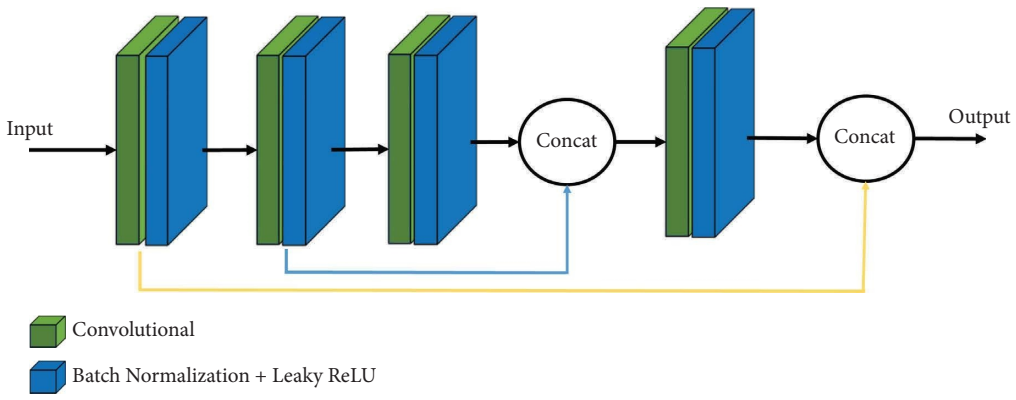


Figure 3: CSPBlock structure.

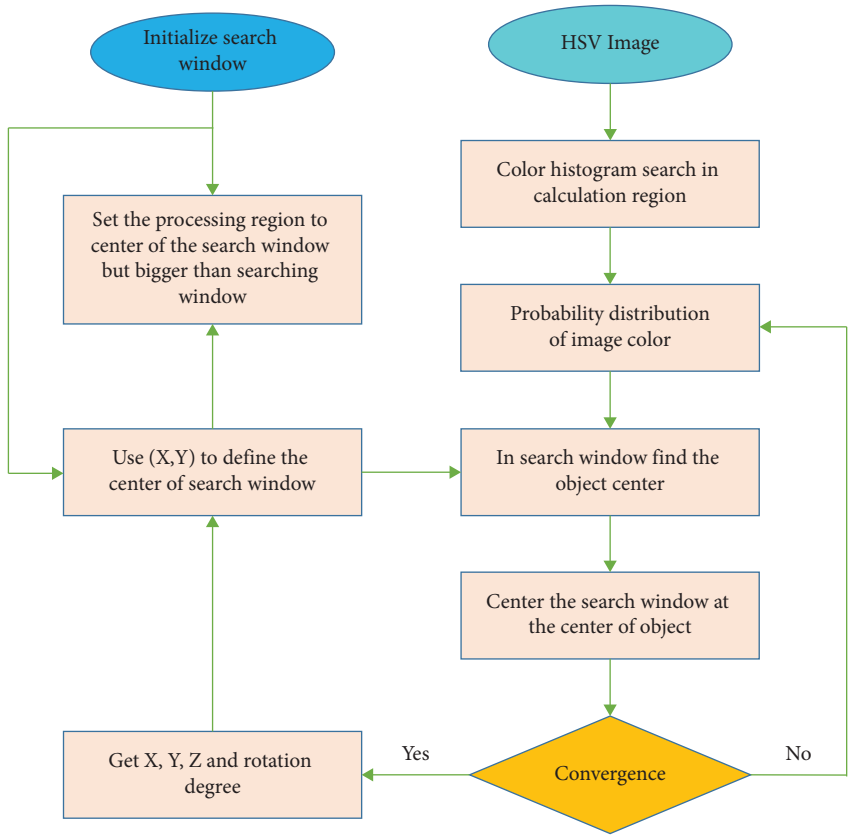


Figure 4: The flowchart of the Camshift algorithm.

In this study, the center of the target object is estimated by the Camshift algorithm, and then it improved the smoothness by using the Kalman filter. The position of the moving target in the next frame can be predicted based on the coordination in the current frame. To detail, the tracking process is a Kalman filter using the tracking parameters from Camshift, then the position of the human is estimated, and subsequently, the output of the Kalman filter is the prediction of object position in the next frame.

The Kalman algorithm can predict accurately the object which is needed for tracking, even when the tracking object is fully occluded. Additionally, the Kalman filter requires small computational power. The Kalman filter uses the information including the position of the object in a frame at time k, the size of the object, width, and length of the search window of the object as input parameters [21]. The variable parameters at time k of the Kalman filter are two vectors including the state vector and the measurement vector [29]. The state vector equation is defined as follows:

$$s_k = (x_k, y_k, W_k, L_k, x_c, y_c). \tag{8}$$

With (x_k, y_k) as the initial positions, (W_k, L_k) are the width and length of the search window and (x_c, y_c) are the centroids of the object.

The measurement vector equation is defined as follows:

$$z_k = (x_k, y_k, W_k, L_k). \tag{9}$$

With (x_k, y_k) as the initial positions, (W_k, L_k) are the width and length of the search window.

The Kalman filter estimate state s is given by equation (10) as follows:

$$s_k = A_k s_{k-1} + w_{k-1}, \tag{10}$$

where A is the state transition matrix and w_{k-1} is the noise process.

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

Measurement equation

$$z_k = H_k s_k + v_k. \tag{12}$$

With H as the measurement matrix, v_k are the stands for noise.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{13}$$

w_{k-1} and v_k are the white Gaussian noise presented by the equation as follows:

$$p(w) \sim N(0, \mathrm{Q}), \tag{14}$$

$$p(v) \sim N(0, R). \tag{15}$$

Equation (14) is the probability distribution of the process noise $w(t)$ that is assumed a normal distribution with a mean of 0 and a covariance matrix Q. The covariance matrix Q shows the degree of dispersion of random variables. Q is often used to evaluate the correlation between variables. Likewise, equation (15) means that the probability distribution of the measurement noise v is assumed with a mean of 0 and a covariance matrix R. The covariance matrix R represents the measurement noise covariance. This matrix is often used to estimate experimentally or in theoretical models.

Assuming independence from the state and the measurement vectors, the process noise w_{k-1} and measurement noise v_k are considered to be normally distributed and have white distributions. The noise process and the measurement noise are presented as follows:

$$w_{k-1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad v_k = \begin{bmatrix} 0.1 \\ 0.1 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

The updated state of the Kalman filter comprises prediction and correction equations.

Predicting equations

$$\begin{aligned} \widehat{\mathrm{s}}_k^- &= \mathrm{A}\widehat{s}_{k-1} + w_k, \\ \mathrm{P}_k^- &= \mathrm{A}\mathrm{P}_{k-1}\mathrm{A}^{\mathrm{T}} + \mathrm{Q}. \end{aligned} \tag{17}$$

Correction equations

$$\begin{aligned} \mathrm{K}_\mathrm{k} &= \mathrm{P}_\mathrm{k}^- \mathrm{H}^{\mathrm{T}} \left(\mathrm{H}\mathrm{P}_\mathrm{k}^- \mathrm{H}^{\mathrm{T}} + \mathrm{R}\right)^{-1}, \\ \widehat{\mathrm{s}}_\mathrm{k} &= \widehat{\mathrm{s}}_\mathrm{k}^- + \mathrm{K}_\mathrm{k}\left(\mathrm{z}_\mathrm{k} - \mathrm{H}\widehat{\mathrm{s}}_\mathrm{k}^-\right), \\ \mathrm{P}_\mathrm{k} &= \left(1 - \mathrm{K}_\mathrm{k}\mathrm{H}\right)\mathrm{P}_\mathrm{k}^-. \end{aligned} \tag{18}$$

The estimation error is used for all state vectors [29]:

$$E_\mathrm{x} = \mathrm{x}_{\mathrm{Kalman}} - x_{\mathrm{Camshift}}. \tag{19}$$

To verify the estimation error of the Kalman filter and Camshift for human tracking, equation (19) is applied for all parameters of state vectors $(x_k, y_k, W_k, L_k, x_c, y_c)$. Table 1 shows the results of the error calculation of the parameters of state vectors in good lighting room and poor lighting room conditions.

2.5. Similarity Measure. The Bhattacharyya coefficient [30] is used as a similarity function to calculate the degree of similarity between two probability distributions. This is an efficient way to measure the similarity of two probability distributions. In this paper, the Bhattacharyya coefficient is used to compare the similarity between two histograms of the target model and the candidates. This measure locates the object correctly, and the distance between two histograms must be minimized. The Bhattacharyya coefficient [28] expression is given as follows:

$$BC(p, q) = \sqrt{1 - \sum_u \frac{\sqrt{q(u) \bullet p(u)}}{\sum_u q(u) \bullet \sum_u p(u)}}, \tag{20}$$

where $q_u(u = 1 \ldots n)$ is the color histogram of the current tracking target. $p_u(u = 1 \ldots n)$ is the color histogram of the target model.

If the result of the Bhattacharyya coefficient equation (20) is closer to 0, this means that the two color histograms are more similar. By contrast, if the result is closer to 1, the

Table 1: The results of error calculation of the parameters of state vectors.

Condition	Case	Frame	$E(x)$	$E(y)$	$E(W)$	$E(L)$	$E(x_c)$	$E(y_c)$
Good lighting room	Multiple people using the same shirt	2	2	−3	−3	−6	−1	−2
		15	1	0	3	1	0	−3
		30	0	1	1	−2	−1	−1
		35	−27	−18	−16	−59	−2	2
		45	−16	6	5	−5	1	0
	Multiple people using the different shirt	2	0	5	3	1	0	−5
		15	−1	2	0	−2	−1	3
		25	14	−11	7	13	4	0
		30	23	2	13	8	−3	0
		35	2	0	1	−3	0	−1
Poor lighting room	Multiple people using the same shirt	2	−1	−3	−1	−5	1	−1
		30	−2	−3	0	−2	0	1
		120	−1	−4	1	−6	−1	0
		130	5	−6	5	−8	1	−2
		160	60	4	−2	−48	0	1
	Multiple people using the different shirt	2	0	1	−1	−11	0	0
		20	3	1	0	−8	−1	−2
		25	34	−3	−2	−16	0	3
		30	1	2	−3	−10	−3	1
		35	0	3	−1	−9	0	1

difference between the two color histograms is greater [31]. Therefore, setting a detection threshold is needed when tracking failure occurs.

3. Results and Discussion

This section presents the results to demonstrate the accuracy and effectiveness of the proposed algorithm in three different scenarios. The scenarios are tested in good lighting room, poor lighting room, and outdoor conditions. Good lighting room and poor lighting room conditions are evaluated with multiple people using the same shirts and different shirts. For outdoor conditions, only the case of the same shirts is performed. Furthermore, the results are compared with traditional Camshift [10] in three different conditions, as mentioned previously.

In order to train the YOLOv4-tiny model, 3600 images are collected from a smartphone camera and 2400 images are taken from Google Images. This model is trained on the local computer with the operating system Windows 10 Pro 64-bit, processor Intel Core i5-8500 CPU @ 3 GHz (6 CPUs) and memory 8 GB RAM. The pretrained model is used to save training time on the COCO dataset combined with our dataset.

The tracking system uses a combination of Camshift and Kalman filter algorithms. Hence, the input parameters of the Kalman filter (the position in x, y of the object, the size of the object, as well as the width and length of the search window of the object) are automatically calculated by the Camshift algorithm. The noise covariance matrices (Q, R) are assumed to be constant. The values of the process noise covariance matrix Q and measurement noise covariance matrix R are presented as follows:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{21}$$

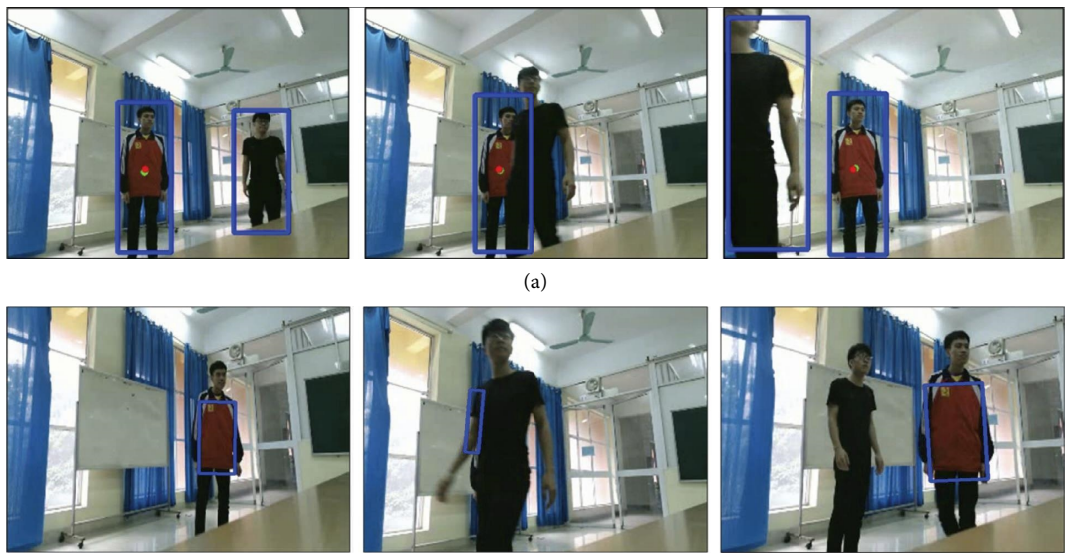
The accuracy of traditional Camshift [10] and the proposed system are evaluated by the total number of frames and the number of false frames in each case. The accuracy is calculated as follows:

$$accuracy\,(\%) = 100 - \frac{false\ frames \times 100}{total\ of\ frames}. \tag{22}$$

Table 2 shows the results of Camshift and the proposed system. Overall, the outcome showed a significantly higher value between the two algorithms. In the traditional Camshift [10], it is very easy to lose the target when the object is the occlusion or in low light conditions. The highest accuracy of traditional Camshift is 90.86% in good light conditions, and the lowest only occupies 47.04% in poor light conditions. The FPS of traditional Camshift is higher than the proposed algorithm, reaching 25. However, the result of the proposed system peaks at 97.77% in good light conditions, and the lowest accuracy is 94.01% in poor light. The performance of the proposed system is higher sharply

Table 2: The experimental results of the proposed system and traditional Camshift.

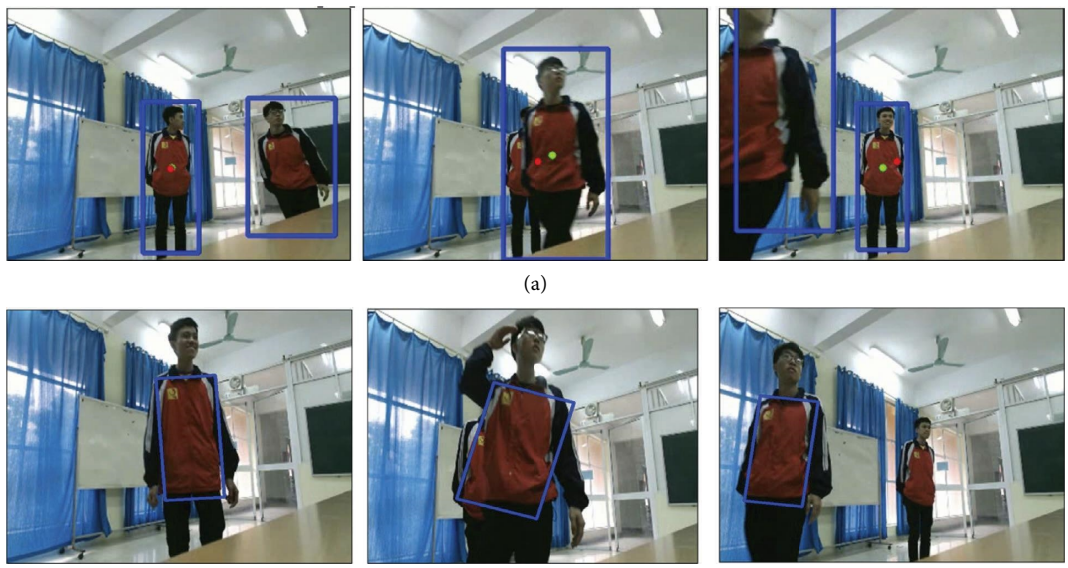
Condition	Test case	Traditional Camshift [10]				The proposed system			
		Number of frames	Number of false frames	Accuracy (%)	FPS	Number of frames	Number of false frames	Accuracy (%)	FPS
Good lighting room conditions (700 lux)	Multiple people using the same shirts	352	136	61.37	25	352	12	96.59	12
	Multiple people using the different shirts	405	37	90.86	25	405	9	97.77	12
Poor lighting room conditions (100 lux)	Multiple people using the same shirts	389	206	47.04	25	389	23	94.08	12
	Multiple people using the different shirts	481	131	72.77	25	481	25	94.80	11
Outdoor conditions (9522 lux)	Multiple people using the same shirts	326	118	63.80	25	326	10	96.93	12



(a)

(b)

Figure 5: The result of the test in good lighting room condition is that people have different color shirts. (a) The proposed method. (b) The traditional Camshift.



(a)

(b)

Figure 6: The result of the test in good lighting room condition is that people have the same shirts. (a) The proposed method. (b) The traditional Camshift.

when compared with traditional Camshift [10]. The proposed algorithm has an FPS range from 11 to 12 enough for the system to still work in real time.

The proposed algorithm is tested with the likeness and different shirts in different light conditions. The green point is Camshift, the red point is the Kalman filter, and the blue bounding boxes are human detection. Figures 5 and 6 show the result of the test in good light conditions. The results show that the proposed algorithm works well in this condition. The algorithm detects all people in the frame, and the tracking algorithm still tracks the target humans even when it is occlusion by another human. In good light conditions, the results using traditional Camshift [10] reach an accuracy of 90.86% in different shirt conditions. This is also the highest accuracy of Camshift in three test conditions. Regarding the case of the same shirts, due to sensitivity with similar colors shirts, the accuracy of Camshift is down to 61.37%. Besides, Camshift switches to tracking other humans when the target human is occlusion.

(a)

(b)

Figure 7: The result of the test in poor lighting room condition is that people have the same color shirts. (a) The proposed method. (b) The traditional Camshift.

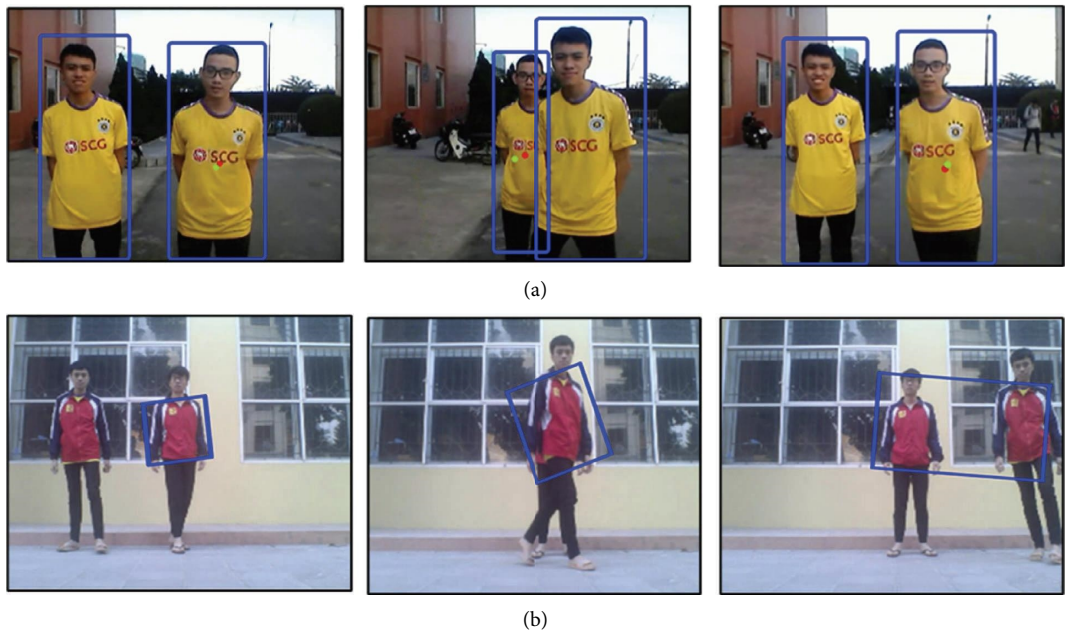
(a)

(b)

Figure 8: The result of the test in poor lighting room conditions is that people have different color shirts. (a) The proposed method. (b) The traditional Camshift.

Figures 7 and 8 show the test experiments in poor lighting conditions with the same and different color shirts. Sometimes, YOLOv4-tiny cannot detect all humans in the frame. Camshift centroid has the deviation from the Kalman centroid when the target human is occlusion by the other human. However, the proposed system still tracks the target human. The accuracy of the proposed method is 94.80% and 94.08% in the different and same color shirts, respectively. In traditional Camshift [10], such as the test in good lighting room conditions, Camshift faces difficulty in tracking the same color target. The accuracy of this case is lowest at 47.04%. In the case of different shirts, when the target human is obscured. The tracking bounding box of Camshift is mounted on the wall because its color is somewhat similar to the shirts. The accuracy reaches 72.77% in this case.

To evaluate more about the ability of the proposed algorithm in complex conditions, the algorithm is tested in outdoor conditions with the same shirts. Figure 9 shows the test results with traditional Camshift [10] and the proposed



(a)

(b)

Figure 9: The result of the test in outdoor conditions is that people have the same color shirts. (a) The proposed method. (b) The traditional Camshift.

system in outdoor conditions. The Camshift algorithm still tracks the target human until the human is covered by other people. Due to the same color shirts, Camshift fails to follow the target. While the proposed algorithm tracks the target human even in case it is obscured by the same color shirts. The accuracy of Camshift and the proposed method are 63.80% and 96.93%, respectively.

4. Conclusions

In this study, an improvement of Camshift based on deep learning and the Kalman filter algorithm is proposed for tracking a target human. To be more detailed, the proposed system is designed based on the combination of YOLOv4-tiny, Camshift, and Kalman filter. This algorithm can be used in real time because of its fastness and robustness. The experiments pointed out that the proposed method can track humans in difficult conditions such as the same color, disappearing problem, or light change conditions. The proposed system solves the problem of occlusion or noise in the traditional Camshift algorithm. By testing and comparing our method with the traditional Camshift, our method achieves better results with a higher accuracy dramatically. The tracking speed is 12 FPS and can meet the requirement in real time. However, the lowest accuracy of the proposed system in poor light conditions is 94.08%. In the future, deep reinforcement learning will be applied to improve object detection in complex light conditions as well as increase the accuracy of the model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Consent

The participants gave their consent for this experiment, and informed consent was obtained from all 375 individual participants included in the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Noor, M. Waqas, M. I. Saleem, and H. N. Minhas, "Automatic object tracking and segmentation using unsupervised SiamMask," *IEEE Access*, vol. 9, pp. 106550–106559, 2021.

[2] Z. Jiang, L. Zhao, S. Li, and Y. Jia, "Real-time object detection method based on improved YOLOv4-tiny," 2020, https://arxiv.org/abs/2011.04244.

[3] P. Bharati and A. Pramanik, "Deep learning techniques—R-CNN to mask R-CNN: a survey," *Computational Intelligence in Pattern Recognition*, pp. 657–668, 2020.

[4] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Computer Networks*, vol. 168, Article ID 107036, 2020.

[5] C. Cao, B. Wang, W. Zhang et al., "An improved faster R-CNN for small object detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019.

[6] K. He, X. Zhang, S. Renand, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[7] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications*, vol. 9, no. 10, Article ID p9420, 2019.

[8] J. Yang, S. Rahardja, and P. Franti, "Mean-shift outlier detection and filtering," *Pattern Recognition*, vol. 115, Article ID 107874, 2021.

[9] V. T. Nguyen and D. T. Chu, "Implementation of real-time human tracking system based on deep learning using kinect camera," in *Intelligent Systems and Networks,* Springer, Berlin, Germany, 2022.

[10] Y. Zhang, "Detection and tracking of human motion targets in video images based on camshift algorithms," *IEEE Sensors Journal*, vol. 20, no. 20, pp. 11887–11893, 2020.

[11] V. T. Nguyen, A. T. Nguyen, V. T. Nguyen, H. A. Bui, and X. T. Nguyen, "Real-time target human tracking using camshift and LucasKanade optical flow algorithm," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 2, pp. 907–914, 2021.

[12] Z. Shu, G. Liuand, and Z. Xie, "Real time target tracking scale adaptive based on LBP operator and nonlinear meanshift," in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 130–133, Nanjing, October 2017.

[13] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, June 2010.

[14] A. Lukezic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4847–4856, Honolulu, HI, USA, July 2017.

[15] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European conference on computer vision*, pp. 254–265, Zurich, Switzerland, September 2014.

[16] V. T. Nguyen, A. T. Nguyen, V. T. Nguyen, and H. A. Bui, "A real-time human tracking system using convolutional neural network and particle filter," in *Proceedings of the Intelligent Systems and Networks*, pp. 411–417, Hanoi, Vietnam, March 2021.

[17] S. Guo, T. Zhang, Y. Song, and F. Qian, "Color feature based object tracking through particle swarm optimization with improved inertia weight," *Sensors*, vol. 18, no. 4, 2018.

[18] W. Ahmed, O. M. Jamil, M. F. Shirazi, and M. H. Abbasi, "PSO with Gompertz increasing inertia weight," in *Proceedings of the IEEE Conference on Industrial Electronics and Applications*, Melbourne, VIC, Australia, June 2013.

[19] J. Zhao, C. Li, Z. Xu, L. Jiao, Z. Zhao, and Z. Wang, "Detection of passenger flow on and off buses based on video images and YOLO algorithm," *Multimedia Tools and Applications*, vol. 81, no. 4, pp. 4669–4692, 2022.

[20] S. Huang and J. Hong, "Moving object tracking system based on Camshift and Kalman filter," in *Proceedings of the 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 1423–1426, Xianning, China, April 2011.

[21] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," in *Proceedings of the IEEE International Conference on Information and Automation*, pp. 1862–1866, Harbin, China, June 2010.

[22] S. Ali, A. Siddique, H. F. Ateş, and B. K. Güntürk, "Improved YOLOv4 for aerial object detection," in *Proceedings of the 29th Signal Processing and Communications Applications Conference*, pp. 1–4, Istanbul, Turkey, June 2021.

[23] Q. Xu, R. Lin, H. Yue, H. Huang, Y. Yang, and Z. Yao, "Research on small target detection in driving scenarios based on improved yolo network," *IEEE Access*, vol. 8, pp. 27574–27583, 2020.

[24] D. Sijie, X. Hongxin, and L. Tianping, "Implementation of camshift target tracking algorithm based on hybrid filtering and multifeature fusion," *Journal of Sensors*, vol. 2020, Article ID 8846977, 13 pages, 2020.

[25] B. Yousefi, C. I. Castanedo, X. P. Maldague, and G. Beaudoin, "Assessing the reliability of an automated system for mineral identification using LWIR Hyperspectral Infrared imagery," *Minerals Engineering*, vol. 155, Article ID 106409, 2020.

[26] Y. Lv and H. Zhu, "An improved camshift tracking algorithm based on LiDAR sensor," *Journal of Sensors*, vol. 2021, Article ID 3353032, 10 pages, 2021.

[27] X. Hu and B. Huang, "Face detection based on SSD and CamShift," in *Proceedings of the IEEE 9th Joint International Information Technology and Artificial Intelligence Conference*, vol. 9, pp. 2324–2328, Chongqing, China, December 2020.

[28] X. Xiao, J. Wang, Q. Shen, and Y. Wang, "An improved CamShift algorithm based on FAST-SIFT feature detection matching," in *Proceedings of the IEEE International Conference on Information Communication and Signal Processing*, pp. 64–68, Singapore, September 2018.

[29] A. Salhi and Y. J. Ameni, "Object tracking system using camshift, meanshift and kalman filter," *world academy of science, engineering and technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 6, pp. 421–426, 2012.

[30] J. H. Lee, H. J. Jungand, and J. Yoo, "A real-time face tracking algorithm using improved camshift with depth information," *Journal of Electrical Engineering and Technology*, vol. 12, no. 5, pp. 2067–2078, 2017.

[31] H. Yu, A. Sharmaand, P. Sharma, and P. Sharma, "Adaptive strategy for sports video moving target detection and tracking technology based on mean shift algorithm," *International Journal of System Assurance Engineering and Management*, pp. 1–11, 2021.