Science & Technology Development Journal 2024, 27(2):3401-3415

Open Access Full Text Article

Research article

Real-Time Convolutional Neural Network-Based Method for Detecting and Tracking Human Motion on Quadcopters

Quoc Duy Tran, Duc Thien Tran*

ABSTRACT

This paper proposes a convolutional neural network (CNN) method for human motion detection and tracking on a quadcopter. To address the challenges mentioned above, the proposed methodology is designed on computer vision techniques with an object tracking algorithm and a CNN model. The object tracking algorithm is implemented using a proportional integral differential (PID) controller to calculate the control parameters, including the pitch and yaw angles, in real time. These parameters are determined by calculating the offset between the position of the human and the camera coordinate frame. To achieve accurate object detection, a CNN model is designed based on the single shot multibox detector (SSD) architecture, which is crucial for object detection. The model above is integrated with the MobileNet base network, which is responsible for feature extraction of the object. The use of self-collected person data in model training ensures good performance for this specific application. The object detection results demonstrate that the model achieves a high level of accuracy (98%). The proposed methodology is applied to an NVIDIA Jetson NANO computer. To rigorously assess the control system, the proposed methodology was used to conduct outdoor flight tests on a campus. These tests prioritized minimal pedestrian traffic and stable weather conditions, ensuring a controlled environment for evaluation. Analysis of the flight data and signal graphs provided valuable insights into the effectiveness of the system.

Key words: Human detection, SSD-MobileNet, CNN, Quadcopter, PID, Real-time processing, Embedded systems

Department of Automation Control, Ho Chi Minh City University of Technology and Education, Vietnam

Correspondence

Duc Thien Tran, Department of Automation Control, Ho Chi Minh City University of Technology and Education, Vietnam

Email: thientd@hcmute.edu.vn

History

- Received: 2024-03-02
- Accepted: 2024-05-31
- Published Online: 2024-6-30

DOI :

https://doi.org/10.32508/stdj.v27i2.4262

INTRODUCTION

Recently, quadcopters have garnered significant attention in various applications due to their vertical take-off and landing capabilities, as well as their hovering functionalities[1]. Additionally, quadcopters can handle intricate tasks within crowded environments and have a simpler control system than other types of UAVs[2]. Common applications are focused on surveillance[3], search and rescue[4], mapping[5], autonomous navigation[6], obstacle avoidance[7] and target tracking[8]. However, among the spectrum of vision-based applications, object detection and tracking on quadcopters present significant challenges, particularly in achieving real-time performance. Balancing computational efficiency with detection accuracy is crucial. Real-time operation demands fast processing, while high accuracy ensures reliable object identification. The integration of robust vision-based estimation and control algorithms is essential for addressing these challenges and unlocking the full potential of quadcopters in vision-based applications.

In practical applications, object detection relies on various deep learning-based algorithms, such as the Faster Region-based Convolutional Neural Network[9], Region-based Fully Convolutional Networks[10], You Only Look Once[11] and Single Shot Detector[12]. These algorithms have demonstrated remarkable capabilities in object detection tasks. However, a common challenge associated with these detectors is their high computational complexity, which can hinder their implementation on resource-constrained embedded platforms such as quadcopters. This limitation is particularly relevant for real-time applications that demand fast processing times. To address this challenge, single-shot detectors have emerged as a promising approach for object detection. These detectors, such as YOLO and SSD, process images in a single pass, significantly reducing the computational overhead compared to two-stage detectors such as Faster R-CNN and R-FCN[13]. YOLO, for instance, is renowned for its real-time processing capabilities, making it suitable for applications that require an immediate response[11]. In contrast, SSD strikes a balance between speed and accuracy by predicting multiple bounding boxes for each object, offering a more robust solution for tasks that demand high detection accuracy[12,14,15].

Moreover, numerous control algorithms have been created to address the challenges associated with

Cite this article : Tran Q D, Tran D T. **Real-Time Convolutional Neural Network-Based Method for Detecting and Tracking Human Motion on Quadcopters**. *Sci. Tech. Dev. J.* 2024; 27(2):3401-3415.

tracking humans. In particular, the article presents the identification and tracking of humans employing techniques for visual data manipulation with OpenCV[16]. In[17], a fuzzy logic controller (FLC) was employed as part of a target tracking algorithm. In[18], they proposed a tracking algorithm grounded in Euclidean space equations and image processing through cameras. While prior studies have demonstrated commendable performances, their primary focus lies within the realm of computer vision, neglecting external disturbances such as environmental factors. To achieve high precision in drone control, several controllers have been applied. In[19], a target-tracking control algorithm based on fuzzy PI was devised. This algorithm incorporates a Fuzzy-PI controller to dynamically adjust the parameters of the PI controller, utilizing positional data and changes in position as inputs. In[20], a gain-scheduled PID controller was developed to guide a UAV by continuously adjusting the actuators based on real-time data from the tracking unit and UAV dynamics. In[21], a comprehensive double closed-loop proportion integral differential (PID) controller was meticulously designed, employing estimated states to accurately track and pursue the target. Among them, PID is a promising candidate for drone control because it not only achieves high accuracy but also remains robust to uncertainties from external influences[22]. The strengths of PID include being model-free, requiring no information about the mathematical model of the system, easy implementation on embedded boards, and high precision[23].

This paper presents an approach for detecting and tracking target objects using an SSD object detector on a UAV. To manage the above challenges, the system is separated into two primary components: (1) object motion estimation and (2) object recognition. The object motion estimation algorithm utilizes a proportional integral differential (PID) controller to compute control parameters, which include pitch and yaw angles in real time. These parameters are determined based on the position of the object and are calculated by measuring the offset between the position of the human and the camera coordinate frame. This module achieves robust object tracking across varying relative distances. Object recognition focuses on accurately detecting "person" objects using the SSD architecture. A custom-trained model differentiates between two classes: images containing objects and images without a person present. Self-collected person data training enhances detection performance. Finally, the proposed control is applied to an NVIDIA Jetson NANO embedded computer. A comprehensive outdoor flight experiment is conducted within a campus environment characterized by minimal pedestrian traffic. Additionally, priority is given to selecting days with favorable weather conditions and stable illumination. The analysis includes assessing experimental flight data and signal graphs to evaluate the proposed control system.

The remainder of this paper is structured as follows: The problem statement, the object recognition algorithm and the object motion estimation algorithm are described in Section II. Section III describes the experimental analysis. Finally, Section IV offers conclusions and outlines avenues for future work.

MATERIALS AND METHODS

Preliminary

In Figure 1, the coordinate frames employed for human tracking via a quadcopter are illustrated. The system includes three coordinate frames: $O_E - x_E y_E z_E$ represents the world, $O_B - x_B y_B z_B$ denotes the quadcopter and $O_C - x_C y_C z_C$ signifies the camera coordinates. For computational convenience, we assume that the quadcopter and camera share the same coordinate frame. To address the challenge of the human motion estimation problem, the key challenge is keeping the transformation matrix between the quadcopter and the human being tracked unchanged. To achieve this transformation matrix, which involves both orientation and position, a proposed camera system aims to determine both the orientation and position through the entirety of the captured image. The relative location of the human concerning the quadcopter is calculated using the camera model, which is expressed as (P_B) in the camera coordinates. The target coordinates (P_0) are then determined in quadcopter coordinates. The relationship between these two coordinates is mathematically expressed as follows:

$$\begin{bmatrix} P_B \\ 1 \end{bmatrix} = T_{B0} P_0 = \begin{bmatrix} R_{B0} & t_{B0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ 1 \end{bmatrix} \qquad (1)$$

where R_{B0} and T_{B0} represent the matrix for rotation and the matrix for transformation between the camera framework and quadcopter framework, respectively. t_{B0} denotes the position of camera[24].

Additionally, to identify human subjects from the camera output, a CNN (convolutional neural network) system is utilized for object detection.

Hardware Specifications

To address the challenges mentioned above, the quadcopter system comprises an executive structure and ground station control, as illustrated in Figure 2. The

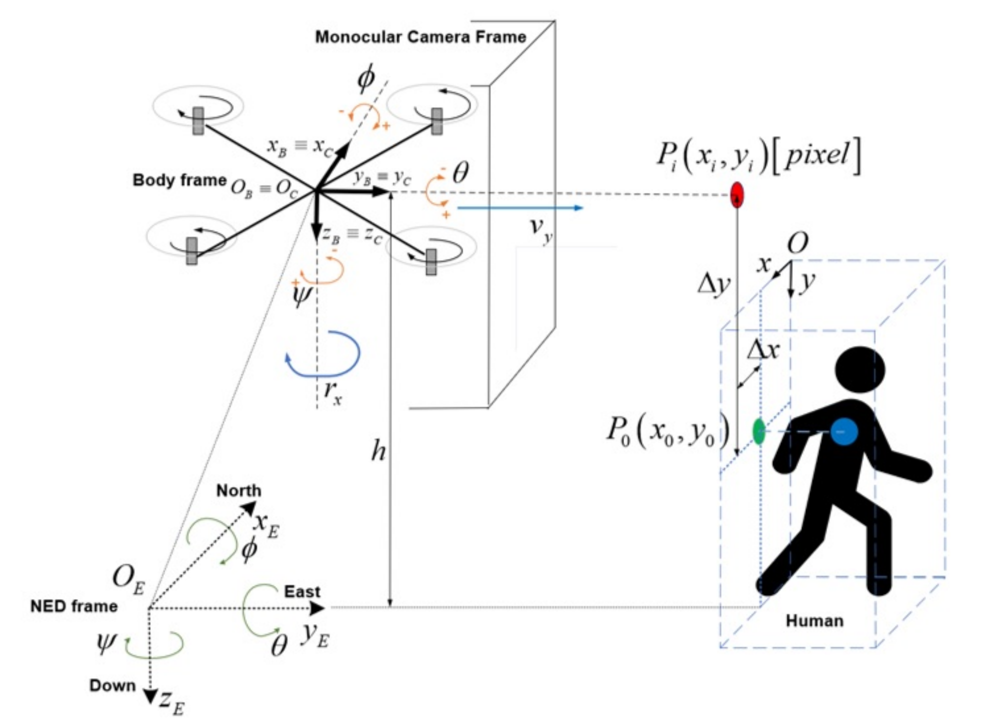


Figure 1: Coordinate frame used for human tracking

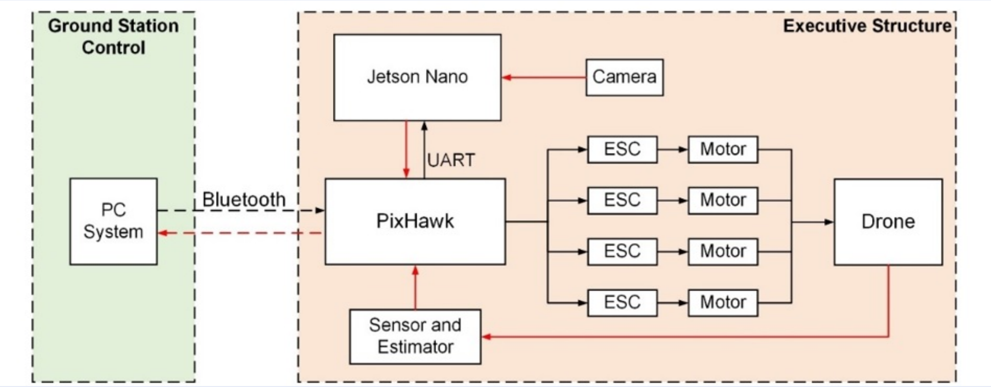


Figure 2: Quadcopter hardware specifications

ground station control is responsible for gathering data from the quadcopter, while the executive structure runs the tracking and detection algorithms.

Control System Overview

The proposed control aims to maintain the transformation matrix between the quadcopter and the tracked human by ensuring consistent output responses. As analyzed in Section II, this transformation matrix involves both the orientation and position of the transformation matrix T_{B0}. This control consists of two main components: vision-based estimation and object tracking control. These parts handle the detection of the targeted human and subsequent human tracking, respectively. The design of this suggested control system overview is outlined in Figure 3.

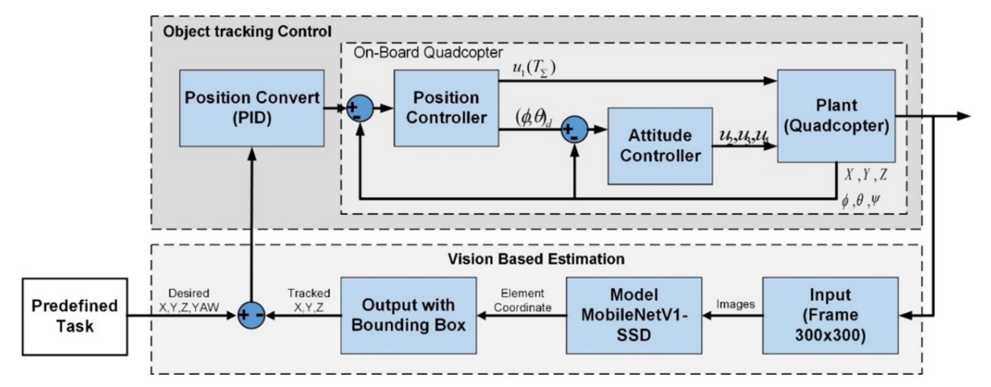


Figure 3: Organization of the Control System

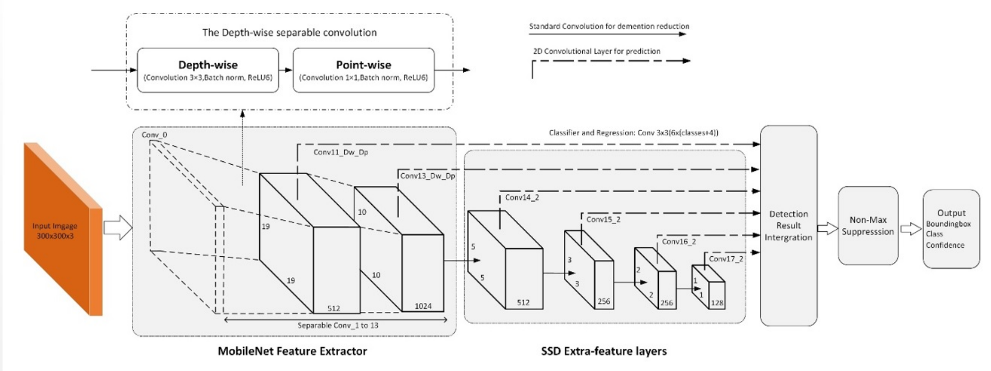

The Depth-wise separable convolution
Depth-wise
(Convolution 3×3,Batch norm, ReLU6)
Point-wise
(Convolution 1×1,Batch norm, ReLU6)
Standard Convolution for demention reduction
2D Convolutional Layer for prediction
Input Imgage 300x300x3
Conv_0
Conv11_Dw_Dp
Conv13_Dw_Dp
19
19
512
10
10
1024
Separable Conv_1 to 13
Classifier and Regression: Conv 3x3(6x(classes+4))
Conv14_2
5
5
512
Conv15_2
3
3
256
Conv16_2
2
2
256
Conv17_2
1
1
128
Detection Result Intergration
Non-Max Suppresssion
Output Boundingbox Class Confidence
MobileNet Feature Extractor
SSD Extra-feature layers

Figure 4: Structure of SSD-MobileNet

Vision-Based Estimation

As illustrated in Figure 4, once an image of an object is received, a CNN algorithm is implemented.

In this study, the SSD method relies on a feed forward convolutional network that generates a bounding box. A subsequent nonmaximum suppression step is applied to produce the final detection results 12. Figure 4 illustrates the MobileNetSSD system, which is an extension of MobileNet[25]. However, it eliminates the fully connected layers and softmax components. MobileNet employs depthwise separable convolution for constructing streamlined deep neural networks, leading to enhancements in computational speed and model size[26,27]. Additionally, MobileNet exhibits strong performance in high-quality image classification tasks, contributing to its popularity in scenarios where transfer learning aids in performance improvement.

The aim of the project is to scale the image to a size of 300x300x3 and feed it into the model through 13 depthwise-separable convolution layers to extract the feature maps, as shown in Figure 5[25]. A feature layer with dimensions of 10x10x1024 is selected to detect objects of various sizes. The initial layers (1-5) in this project are utilized for identifying typical characteristics present in the object image. The following layers (from 6 onward) contain more specific information about the object. Next, the output of Conv_13 in the MobileNet base network is sequentially convolved with a 3x3 kernel, Stride = 2, and a 1x1 kernel, Stride = 1, to generate subsequent downsized feature maps. The project requires a total of 6 feature maps to serve as object detection layers. For every cell in the detection feature map, 4 default boxes are set up, each having 5 distinct aspect ratios to encompass size variations. To obtain a single bounding box for a recognized object (person), the prediction box with the greatest level of confidence is selected. Any bounding boxes with an intersection over union (IoU) threshold greater than the set threshold are removed. This process is repeated until only one bounding box remains to be output.

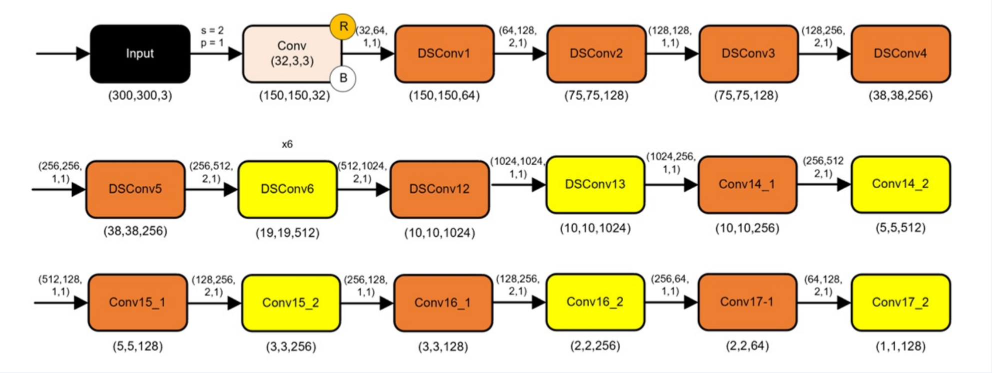


Figure 5: Depthwise-separable convolution layers

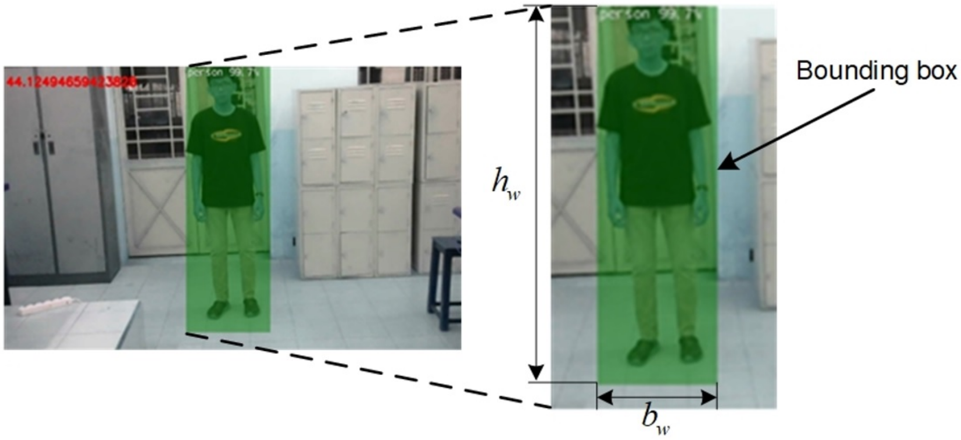


Figure 6: The bounding box after applying SSD-MobileNet

Following the application of SSD-MobileNet, Figure 6 depicts the presentation of a bounding box around the identified person. The positional data of the detected target are then extracted and employed as an input for initiating the object motion estimation algorithm to commence the estimation process.

Control of Object Motion Estimation

Figure 7 indicates the human's position in the camera coordinate system. To track a human using the entire captured image, it is essential to determine the human's position in the coordinate framework fixed to the camera. The $O_C - x_C y_C z_C$ coordinate framework represents the camera coordinates. $P_0\left(x_0, y_0, z_0\right)$ represents the human's position at the center of the camera coordinates, where signifies the width [in pixels] and represents the height [in pixels] of the entire image. Figure 8 illustrates the connection between the camera coordinates and the global coordinates. Calculating the coordinates (y_C, z_C) is feasible because the whole image is two-dimensional. However, it is difficult to calculate the distance in x_C. Consequently, x_C is computed as follows:

$$\theta_1 = \theta_0 \frac{2|y_c| + b_w}{2W} \tag{2}$$

$$x_C = \frac{2|y_C| + b_w}{2\tan\left(\theta_1\right)} \tag{3}$$

where θ_0 is the angle of view of the camera and θ_1 is the angle between the straight line and the z_C-axis. Figure 1 illustrates the coordination frames utilized for human tracking. $O_B - x_B y_B z_B$ represents the quadcopter coordinate system. Within this system, v_x [m/s] denotes the translational velocities of the quadcopter along the x_B-axis in $O_B - x_B y_B z_B$. Additionally, ψ_z [rad/s] signifies the angular velocity of

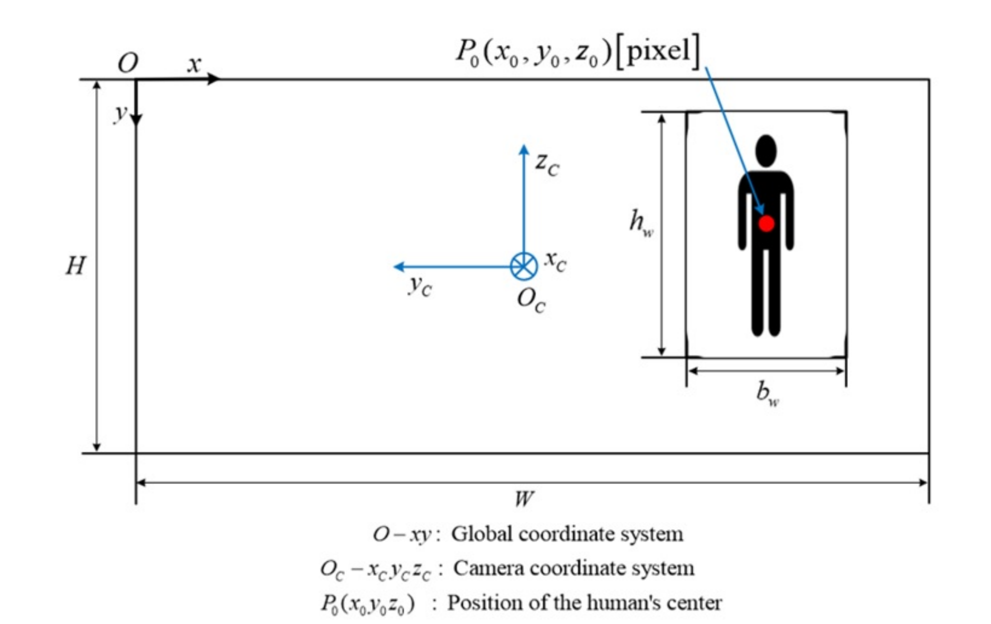

$P_0(x_0, y_0, z_0)$[pixel]
O
x
y
H
W
z_C
x_C
y_C
O_C
h_w
b_w
$O-xy$: Global coordinate system
$O_C-x_Cy_Cz_C$: Camera coordinate system
$P_0(x_0y_0z_0)$: Position of the human's center

Figure 7: Humans in the camera coordinate system

the quadcopter around the z_B-axis in $O_B - x_By_Bz_B$. The desired human position is designated $\bar{P}_0(\bar{x}_0(= const), \bar{y}_0\,(=0), \bar{z}_0\,(=0))$. In Figure 3, a block diagram of position conversion (PID) concerning the quadcopter velocity for human tracking is depicted. It is necessary to give velocities such that $P_0\,(x_0, y_0, z_0)$ comes to the center ($y_C = z_C = 0$) of images captured by the camera of the quadcopter while maintaining the distance ($x_C = const$) between the quadcopter and the human. Subsequently, the translational velocities v_x [m/s] and the angular velocity ψ_z [rad/s], which enable the quadcopter to track the human, are determined as follows:

$$v_x = k_{px}e_x + k_{ix}\int_0^t e_x\,(\tau)\,d\tau + k_{dx}\frac{de_x}{dt} \tag{4}$$

$$\psi_z = k_{pz}e_z + k_{iz}\int_0^t e_z\,(\tau)\,d\tau + k_{dz}\frac{de_z}{dt} \tag{5}$$

where $e_x = \bar{x}_0 - x_0$ and $e_z = \bar{y}_0 - y_0$ are the errors between the position of the human in the center of the camera coordinate frame and the desired human position. When the human is undetected in the captured images, the values of v_x [m/s] and ψ_z [rad/s] are both set to zero. The quadcopter continues human tracking until a terminal command signal is received. The proposed method effectively enables quadcopters to track humans.

The object tracking algorithm is shown in Algorithm 1. The algorithm takes as input from the image of a person. Following initialization, the quadcopter undergoes a series of checks to ensure safe and reliable operation. This initialization phase might involve calibrating sensors, verifying battery levels, and confirming proper motor function. Once it is given the all-clear, the quadcopter autonomously ascends to a predetermined altitude. This chosen altitude offers a suitable vantage point for the search mission, allowing the camera to capture a wider field of view and potentially increasing the chance of human detection. The quadcopter then starts on a 36-second search mission for a human target. It continuously scans the environment using the SSD-MobileNet model. Upon successful detection, the center offset method is used to track the target by calculating the offset between the person and the center of the image captured by the camera. If the offset exceeds zero and the image center lies outside the bounding box, the quadcopter rotates accordingly; otherwise, it moves forward and backward. In the absence of human detection within a designated timeframe, the system assumes that the target is no longer present. To optimize the search efficiency, the quadcopter performs a preprogrammed 10-degree rotation, expanding the search area and increasing the probability of detection. This iterative process of scanning, tracking (if detected), and rotating continues for a total of 36 seconds. If no human is detected throughout this period, prioritizing safety, the system automatically initiates a landing sequence, returning the quadcopter to the ground.

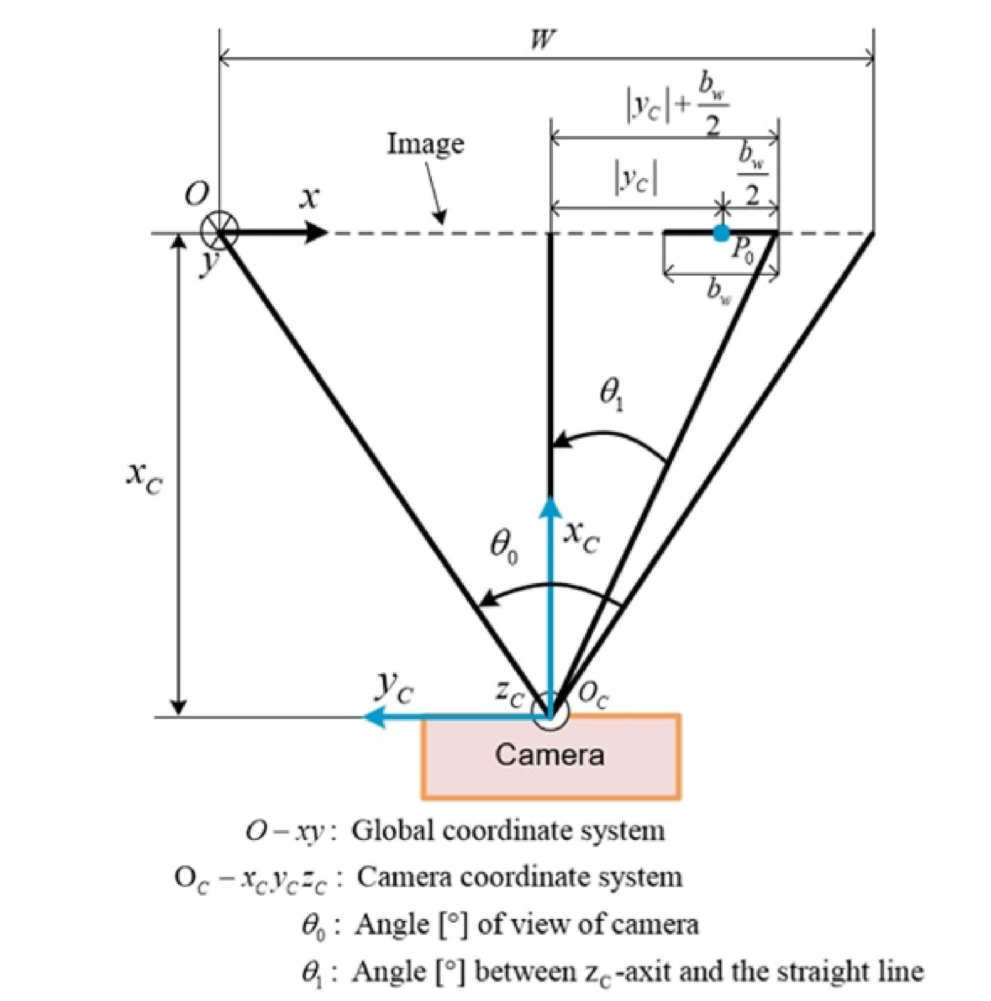

W
$|y_C|+\frac{b_w}{2}$
Image
O
x
y
$|y_C|$
$\frac{b_w}{2}$
P_0
b_w
θ_1
x_C
θ_0
x_C
y_C
z_C
O_C
Camera
$O-xy$: Global coordinate system
$O_C-x_Cy_Cz_C$: Camera coordinate system
θ_0 : Angle [°] of view of camera
θ_1 : Angle [°] between z_C-axit and the straight line

Figure 8: Relationship camera coordinate frame and global coordinate frame

RESULTS AND DISCUSSION

To further assess the benefits of the suggested control, a series of experiments and evaluations on an actual system are carried out.

Experiment description

Figure 9 illustrates the basic movements of the quadcopter during object detection and tracking. We conducted a series of experiments to quantitatively evaluate the algorithm's performance on real hardware. We utilized an NVIDIA Jetson NANO embedded computer for this purpose. The algorithm was implemented in Python within the Ubuntu Linux environment. The experiments were carried out outdoors on the HCMUTE campus. To minimize the presence of multiple objects in the scene, we chose a location with minimal pedestrian traffic. Additionally, favorable weather conditions were ensured to obtain accurate evaluation results. The experiments and results were divided into three parts. First, we evaluated the post-training data to assess the algorithm's ability to detect humans accurately using metrics such as precision, recall, and F1-score. Second, the flight data evaluation focused on system stability and tracking performance. This involved assessing the quadcopter's stability during takeoff, hovering, landing, and directional movements (forward, backward, and rotational). Finally, the data are evaluated when combining object detection and object tracking.

Experimental results

CNN Training

Figure 10 illustrates the process of collecting and preparing data for model training.

Table 1: Algorithm 1: Object Tracking Algorithm

Algorithm 1: Object Tracking Algorithm

```
input: Image person
outputs: $v_x$ and $\psi_z$
begin
    /* Initialize */
    Sensor calibration, battery level verification, motor confirmation
    Take off quadcopter
    while (within 36 seconds)
        Detect human using SSD-MobileNet
        if (objects) then
            Calculate the center of the frame, the person $P_0$
            Calculate the offset between the person and the frame ( $e_x$, $e_y$)
            if (offset > 0) & (not centered) then
                Calculate PID control for rotation $\psi_z$
                Send the rotation control command
            end
            else
                Calculate PID control for forward, backward $v_x$
                Send the forward, backward control command
            end
        end
        Rotation by an angle of 10 degrees
    end while
    Landing
end
```

This study employed a single shot detector (SSD) implemented on a powerful processing unit for human detection on a quadcopter. The SSD model was specifically trained to recognize a single class: individuals (persons). To train and evaluate this model effectively, we constructed a comprehensive image dataset containing two distinct categories: images with objects (primarily featuring individuals) and images devoid of objects. The images were carefully curated to ensure their suitability for real-world applications involving human detection in a quadcopter environment. The image acquisition process involved capturing video footage from the quadcopter's camera. The footpad showcased a diverse range of human subjects, including group members and other individuals within the research laboratory. This footpad was then painstakingly segmented into individual frames, resulting in a raw dataset of approximately 1000 images. To augment the dataset and enhance its learning potential, we employed data augmentation techniques. Redundant images were removed, and a subset of images was transformed using basic manipulations (rotation, scaling, flipping, and brightness) to introduce variations, enrich the dataset and promote model generalizability. Figure 11 shows the process of labeling the data from the dataset.

The dataset comprised a total of 1000 images, maintaining a 3:1 ratio between images with and without objects. Each image featuring a person was meticulously labeled for accurate object identification during training. Subsequently, these images were divided into three distinct sets—training (70%), validation (20%), and testing (10%)—for network training and evaluation. The network configuration included a dropout ratio of 0.7, a kernel size of 3x3, a box code size of 4, and a learning rate of 0.001. The training process was conducted through 200 iterations using Google Colab. Figure 12 illustrates the model's outcomes after completion of the training process.

To assess how well the proposed object detection method performs on an embedded computer, experiments were conducted using the confusion matrix method. The experiments were conducted 50 times and included both positive and negative person instances. These experiments yielded the following metrics: precision = 0.96078, recall = 0.98, and F1 = 0.9703. Figure 13 illustrates the results of the training model.

Additionally, the object detection process analyzed a frame and generated an output for the detected object within a time span of 5 ms. During this 5 ms interval, frames captured by the quadcopter's camera underwent processing, and the CNN provided the output

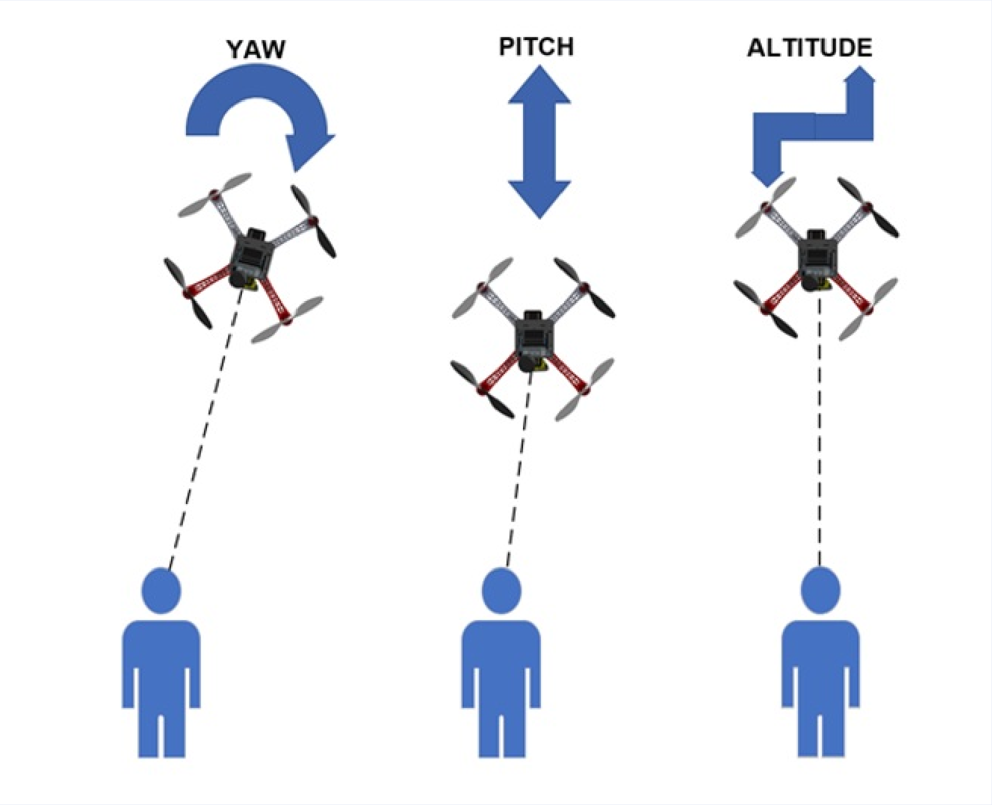


Figure 9: Basic movements of the quadcopter when detecting and tracking objects

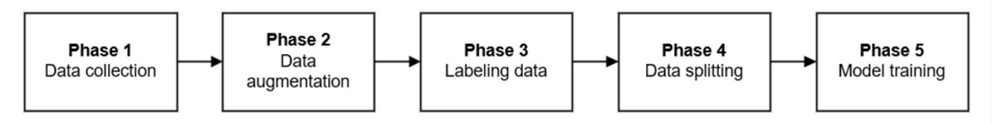


Figure 10: The process of collecting and preparing data.

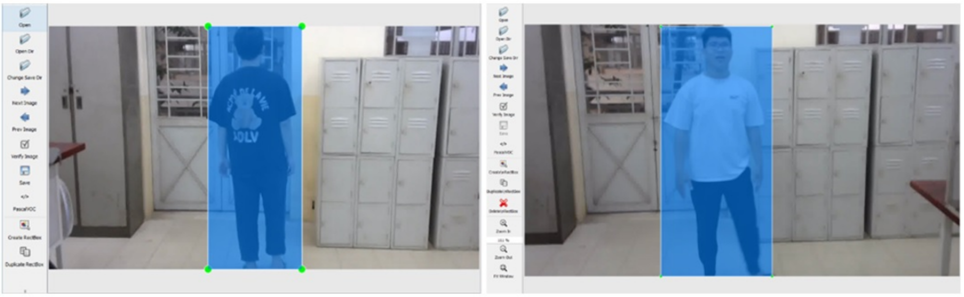


Figure 11: Images from the dataset are labeled.

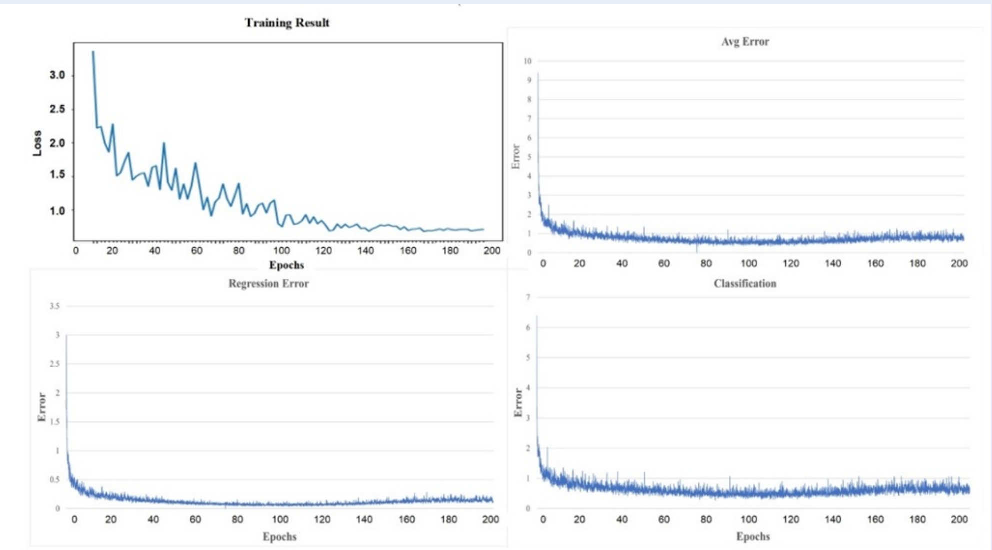


Figure 12: Outcomes of the model after training

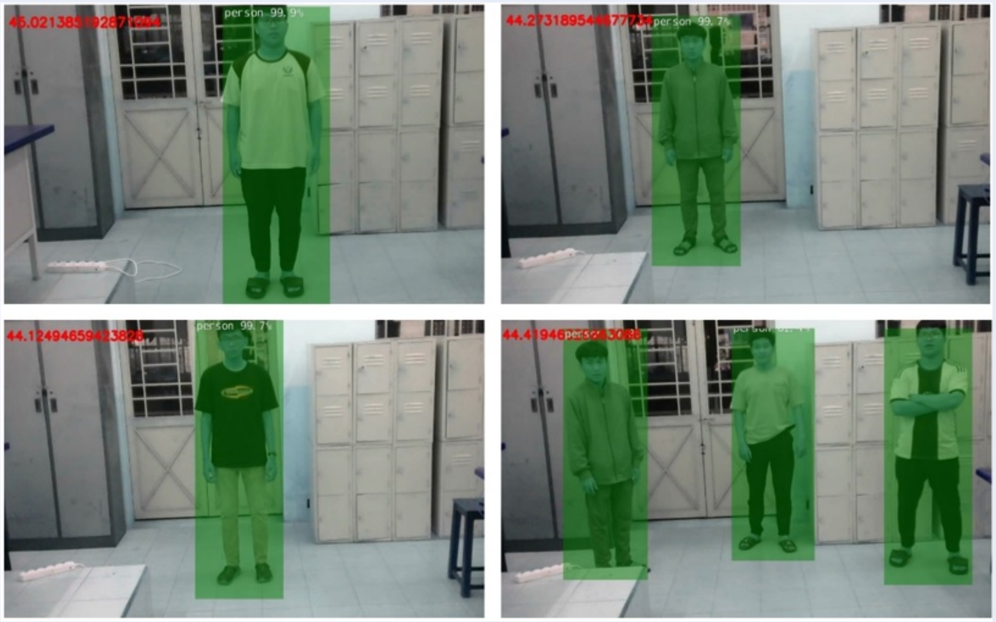


Figure 13: Output of the CNN model

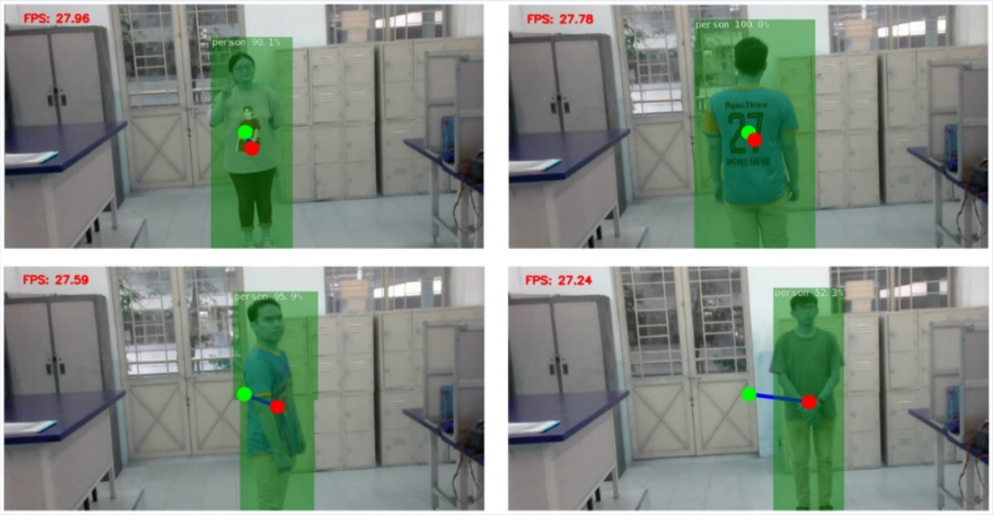


Figure 14: Evaluate the processing speed of the model

in the form of an image featuring the bounding box around the identified object, achieving a frame rate of 27 frames per second (FPS), as shown in Figure 14.

Altitude Control

To test the performance of the proposed controller, an altitude experiment is first carried out. The objective of this altitude test is to control the quadcopter to take off vertically to a desired altitude of approximately 2.5 meters and maintain that altitude for approximately 100 seconds before landing. In Figure 15, the initial altitude (Z) is set to zero because the altitude of the quadcopter remains fixed at 2.5 meters upon takeoff. After removal, the quadcopter hovers at this fixed altitude (Z).

Forward, reverse motion control

Following the altitude experiment, rotation and forward-backward experiments are conducted. The aim of this experiment is to control the aircraft to move at predetermined speeds and angles. The preset speed is 1 m/s, and the rotation angle is set to 90 degrees; this process is repeated three times within a 60-second flight time. The resulting data are represented as squares in Figure 16.

Combined Control

After conducting two flight experiments involving tracking in the forward, backward, and object rotation directions, the goal is for the quadcopter to detect objects within the frame and simultaneously perform forward-backward movement and object tracking. Figure 17 shows the real-world object tracking experiment. The detected object will move freely to verify the accuracy of the system. The validation flight process took place over approximately 300 seconds.

Based on the signals from the graphs, we can observe the aircraft's status during the tracking process in the forward, backward, and rotation directions. The roll angle is approximately equal to 0. In the yaw angle response graph, the aircraft rotates from approximately 180 degrees to 0 degrees within 80 seconds, from the 50th to the 130th second, after which it moves northward. During this time, the yaw angle experiences only slight rotation in the north direction. This indicates that the quadcopter tracks the detected object relatively well. From the velocity response graph in the x-direction, it is evident that the aircraft's velocity in the x-direction is very low, indicating slow forward movement. However, it still responds effectively to track the object. Furthermore, the engine pulse output graph shows that the engines pulse continuously when the aircraft is in a combined state. Pulse generation during takeoff and landing is very fast, demonstrating stable takeoff and landing. The resulting data are represented as squares in Figure 18.

While our object detection and tracking algorithm demonstrated promising results, its real-world implementation presented unforeseen hurdles. A significant challenge arose from the delayed response data received from the quadcopter. This latency, attributed to the limitations of Bluetooth data transmission, created a disadvantage in the real-time processing pipeline. Furthermore, the hardware of onboard cameras occasionally hinders the ability of SSD object detectors to consistently identify target objects.

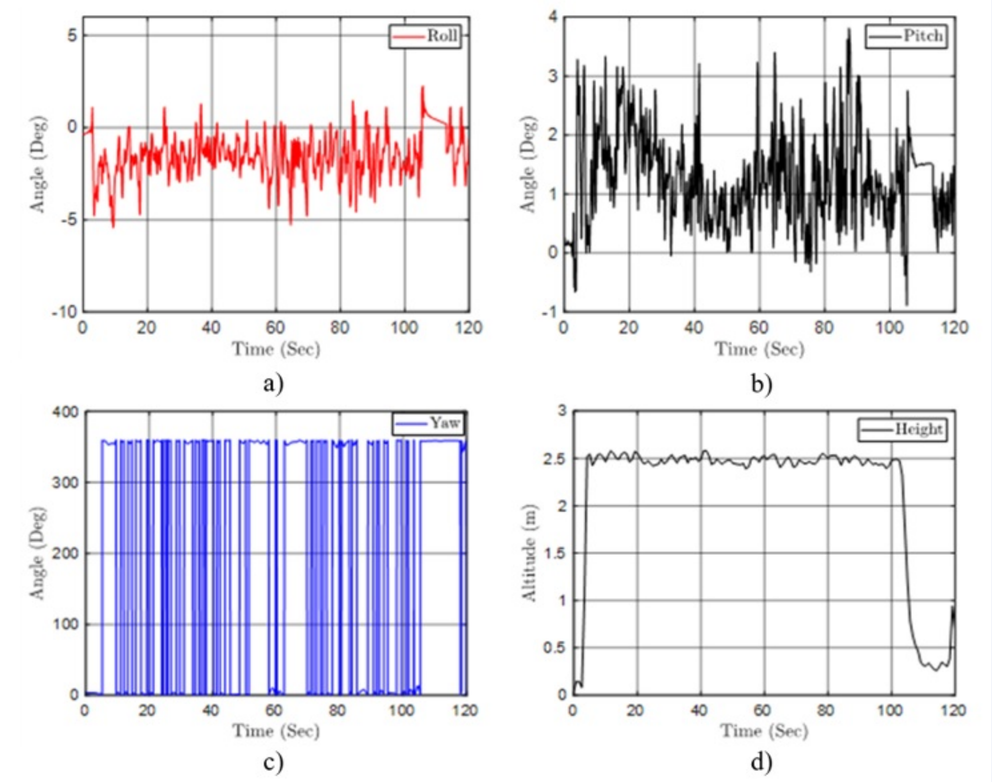


Figure 15: Output responses of the altitude control experiment

This limitation was particularly evident under varying lighting conditions, where real-time object detection proved challenging.

CONCLUSION

This study presents a novel approach for human motion detection and tracking on a quadcopter, leveraging the power of convolutional neural networks (CNNs). The proposed system, implemented on an embedded computer, comprises two key components: object recognition and object motion estimation. The object recognition module employs a CNN-based SSD model to identify moving objects within the camera's field of view. This model effectively generates bounding boxes around detected objects, extracting their center positions for precise tracking. Simultaneously, the object motion estimation module, powered by a PID controller, dynamically adjusts the quadcopter's flight path to pursue the target object even under varying speeds. The experimental results demonstrate the impressive capabilities of the system. The object recognition algorithm boasts high accuracy in object detection and categorization while maintaining low power consumption and achieving a high frame rate (fps). However, real-time implementation has revealed limitations associated with communication latency due to Bluetooth data transmission and onboard camera hardware constraints. These limitations manifested as occasional delays in receiving data and hindered object detection accuracy under varying lighting conditions.

In the future, this work paves the way for further advancements. Integrating vision-based techniques with a stereo camera to estimate the distance between the quadcopter and the target object has emerged as a crucial area for future research and development. This advancement would enable more precise object tracking and navigation, particularly in complex environments. Additionally, the focus will shift toward developing more sophisticated algorithms for handling multiple objects. By incorporating techniques for multiobject tracking, the system could effectively track and differentiate between multiple people in high-density environments. This advancement would be invaluable for applications such as search and rescue operations in crowded areas or autonomous surveillance tasks involving multiple targets.

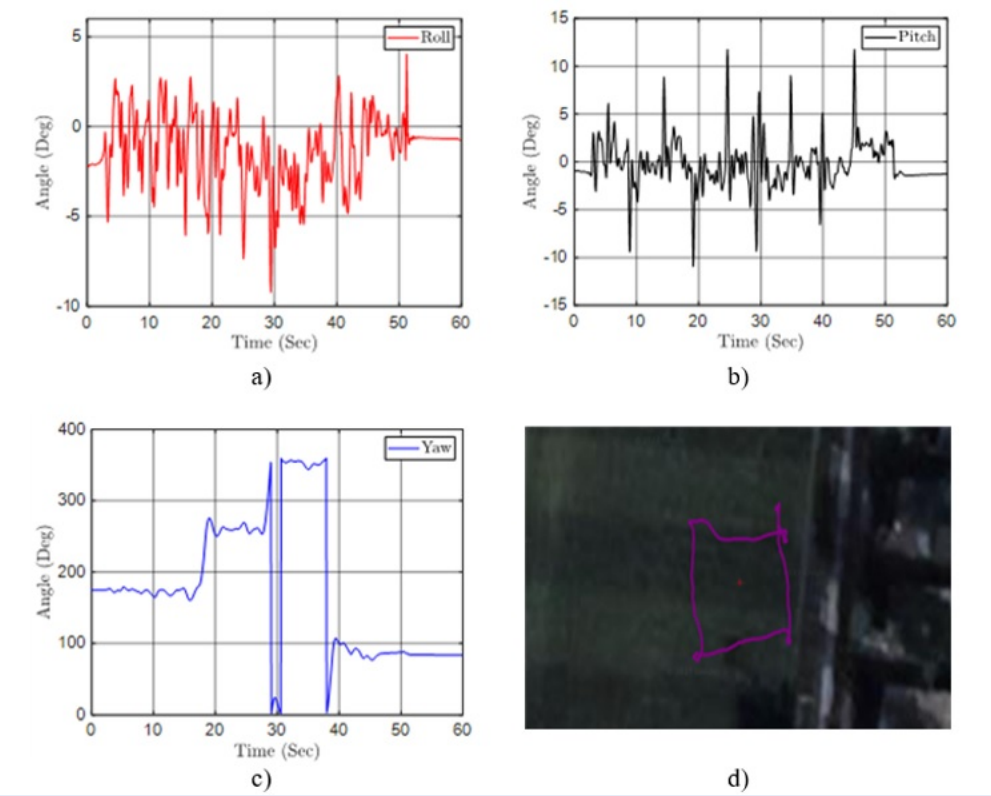


Figure 16: Output responses of the forward and reverse motion control experiments

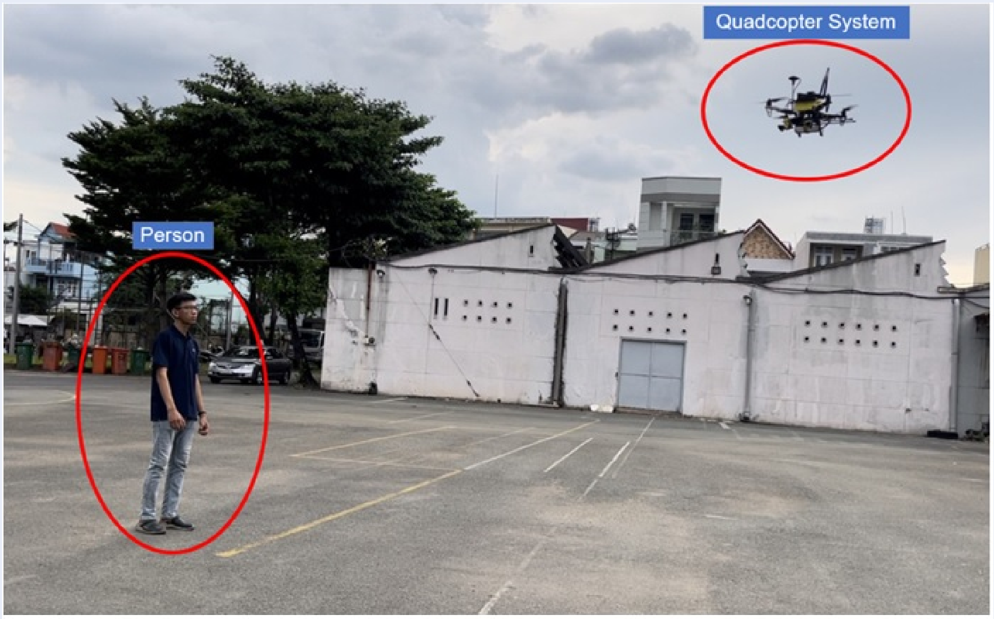
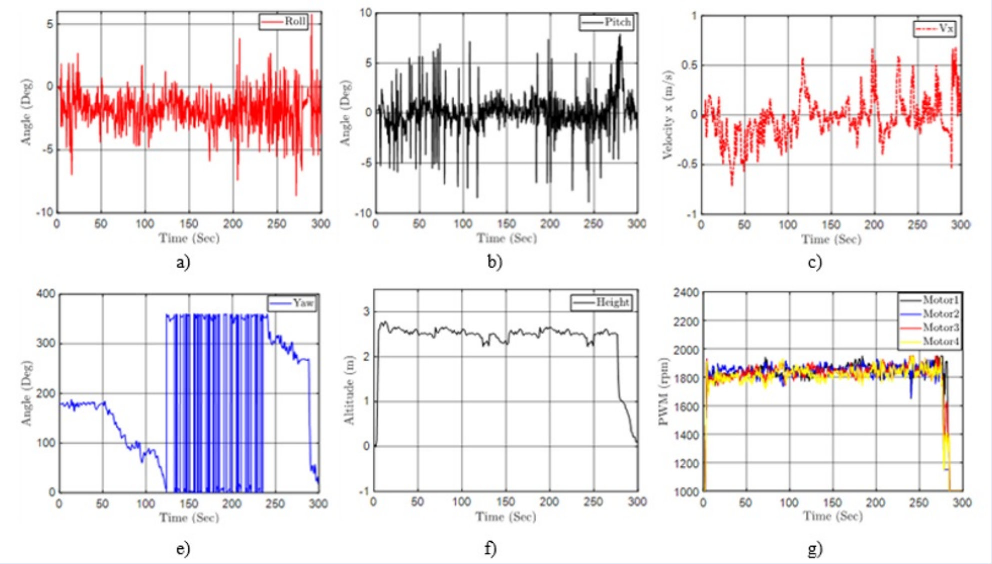


Figure 17: Object tracking experiment in the real world



a) b) c)

e) f) g)

Figure 18: Output responses of the combined experiment

COMPETING INTERESTS

None

NONFINANCIAL COMPETING INTERESTS

None

ACKNOWLEDGMENTS

This research was implemented at the Robotics and Intelligent Control Laboratory (RIC Lab), Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam.

AUTHOR CONTRIBUTION

The authors acknowledge the support of time and facilities from the Robotics and Intelligent Control Laboratory for this study.

REFERENCES

1. Shakhatreh H, Sawalmeh AH, Al-Fuqaha A, Dou Z, Almaita E, Khalil I, et al. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. Ieee Access. 2019;7:48572-634;Available from: https://doi.org/10.1109/ACCESS.2019.2909530.
2. Deepak B, Singh P. A survey on design and development of an unmanned aerial vehicle (quadcopter). International Journal of Intelligent Unmanned Systems. 2016;4(2):70-106;Available from: https://doi.org/10.1108/IJIUS-10-2015-0012.
3. Zaheer Z, Usmani A, Khan E, Qadeer MA, editors. Aerial surveillance system using UAV. 2016 thirteenth international conference on wireless and optical communications networks (WOCN); 2016: IEEE;Available from: https://doi.org/10.1109/WOCN.2016.7759885.
4. Bevacqua G, Cacace J, Finzi A, Lippiello V, editors. Mixed-initiative planning and execution for multiple drones in search and rescue missions. Proceedings of the International Conference on Automated Planning and Scheduling; 2015;Available from: https://doi.org/10.1609/icaps.v25i1.13700.
5. Qin H, Meng Z, Meng W, Chen X, Sun H, Lin F, et al. Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments. IEEE Transactions on Vehicular Technology. 2019;68(2):1339-50;Available from: https://doi.org/10.1109/TVT.2018.2890416.
6. Padhy RP, Xia F, Choudhury SK, Sa PK, Bakshi S. Monocular vision aided autonomous UAV navigation in indoor corridor environments. IEEE Transactions on Sustainable Computing. 2018;4(1):96-108;Available from: https://doi.org/10.1109/TSUSC.2018.2810952.
7. McGuire K, De Croon G, De Wagter C, Tuyls K, Kappen H. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. IEEE Robotics and Automation Letters. 2017;2(2):1070-6;Available from: https://doi.org/10.1109/LRA.2017.2658940.
8. Le T-L, Quynh NV, Long NK, Hong SK. Multilayer interval type-2 fuzzy controller design for quadcopter unmanned aerial vehicles using Jaya algorithm. IEEE Access. 2020;8:181246-57;Available from: https://doi.org/10.1109/ACCESS.2020.3028617.
9. Faster R. Toward real-time object detection with region proposal networks. Advances in neural information processing systems. 2015;9199(10.5555):2969239-50;.
10. Dai J, Li Y, He K, Sun J. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems. 2016;29;.
11. Redmon J, Divvala S, Girshick R, Farhadi A, editors. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition; 2016;Available from: https://doi.org/10.1109/CVPR.2016.91.
12. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al., editors. Ssd: Single shot multibox detector. Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14; 2016: Springer;Available from: https://doi.org/10.1007/978-3-319-46448-0_2.

13. Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B. A survey of modern deep learning based object detection models. Digital Signal Processing. 2022;126:103514;Available from: https://doi.org/10.1016/j.dsp.2022.103514.
14. Phadtare M, Choudhari V, Pedram R, Vartak S. Comparison between yolo and ssd mobile net for object detection in a surveillance drone. Int J Sci Res Eng Man. 2021;5:1-5;.
15. Alkentar SM, Alsahwa B, Assalem A, Karakolla D. Practical comparation of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection. Journal of Engineering. 2021;27(8):19-31;Available from: https://doi.org/10.31026/j.eng.2021.08.02.
16. George RP, Prakash V, editors. Real-time human detection and tracking using quadcopter. Intelligent Embedded Systems: Select Proceedings of ICNETS2, Volume II; 2018: Springer;Available from: https://doi.org/10.1007/978-981-10-8575-8_29.
17. Pei C, Zhang J, Wang X, Zhang Q. Research of a nonlinearity control algorithm for UAV target tracking based on fuzzy logic systems. Microsystem Technologies. 2018;24:2237-52;Available from: https://doi.org/10.1007/s00542-017-3641-0.
18. Chen P, Dang Y, Liang R, Zhu W, He X. Real-time object tracking on a drone with multi-inertial sensing data. IEEE Transactions on Intelligent Transportation Systems. 2017;19(1):131-9;Available from: https://doi.org/10.1109/TITS.2017.2750091.
19. Rabah M, Rohan A, Mohamed SA, Kim S-H. Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-PI. IEEE access. 2019;7:38407-19;Available from: https://doi.org/10.1109/ACCESS.2019.2906345.
20. Rabah M, Rohan A, Haghbayan M-H, Plosila J, Kim S-H. Heterogeneous parallelization for object detection and tracking in UAVs. IEEE access. 2020;8:42784-93;Available from: https://doi.org/10.1109/ACCESS.2020.2977120.
21. Wu S, Li R, Shi Y, Liu Q. Vision-based target detection and tracking system for a quadcopter. IEEE Access. 2021;9:62043-54;Available from: https://doi.org/10.1109/ACCESS.2021.3074413.
22. Praveen V, Pillai S. Modeling and simulation of quadcopter using PID controller. International Journal of Control Theory and Applications. 2016;9(15):7151-8;.
23. Joseph SB, Dada EG, Abidemi A, Oyewola DO, Khammas BM. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. Heliyon. 2022;8(5);PMID: 35600459. Available from: https://doi.org/10.1016/j.heliyon.2022.e09399.
24. Luukkonen T. Modeling and control of quadcopter. Independent research project in applied mathematics, Espoo. 2011;22(22);.
25. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C, editors. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition; 2018;Available from: https://doi.org/10.1109/CVPR.2018.00474.
26. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861. 2017;.
27. Szegedy C, Reed S, Erhan D, Anguelov D, Ioffe S. Scalable, high-quality object detection. arXiv preprint arXiv:14121441. 2014;.