Journal of Telecommunication, Electronic and Computer Engineering
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 17 No. 4 (2025) 25-31
jtec.utem.edu.my
DOI: https://doi.org/10.54554/jtec.2025.17.04.004

Design And Implementation of an Object-Following System by Using DJI Tello Drone

Lam Shu Xuan[1], Aminurrashid Noordin[1,*], Suziana Ahmad[1,*], Azhan AB Rahman[1], Mohd Ariffanan Mohd Basri[2], Izzuddin Mat Lazim[3], Mustafa Saad Khalifa[4]

[1]*Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Malaysia.*
[2]*Faculty of Electrical Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, Malaysia.*
[3]*Faculty of Engineering and Built Environment, Universiti Sains Islam Malaysia, Negeri Sembilan, Malaysia.*
[4]*College of Electronic Technology Bani Walid, Tripoli Road, 38645 Bani-Walid, Libya.*

Article Info

Article history:
Received May 5th, 2025
Revised Sep 25th, 2025
Accepted Oct 21st, 2025
Published Dec 24th, 2025

Index Terms:
Drone
Real-time
Object-tracking
Computer-vision
Quadrotor

Abstract

Unmanned aerial vehicles (UAVs) have become increasingly important across a wide range of industries due to their versatility and ease of deployment. This study focuses on the development of a system for real-time object tracking and following, utilizing the DJI Tello drone. The DJI Tello is a lightweight quadcopter equipped with a 5-megapixel camera capable of capturing 720p video at 30 frames per second. The system employs computer vision techniques, particularly Convolutional Neural Networks (CNNs), to detect and track moving objects in real time, while dynamically adjusting the drone's flight path to maintain continuous visual contact. The core functionality involves designing and testing an algorithm that processes the video feed from the drone's camera and transmits flight commands to the drone's controller. The control system is essential for maintaining a safe distance from the target while avoiding collisions with surrounding obstacles. Python was used to communicate with the drone over Wi-Fi, issuing commands for take-off, landing, movement, and flight maneuvers. The system was tested under real-world conditions, such as tracking moving vehicles or pedestrians. By integrating the capabilities of the DJI Tello drone, advanced computer vision algorithms, and a robust control system, the proposed solution demonstrates potential to enhance UAV applications in disaster management, emergency response, and search and rescue operations.

*Corresponding Author: {aminurrashid, suziana}@utem.edu.my

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), commonly referred to as drones, are aircraft operated without an onboard human pilot [1][2] and may include an integrated autopilot system [3]. These vehicles can be controlled remotely or operated autonomously through pre-programmed flight plans and onboard sensors and navigation systems [4]. UAVs have gained significant popularity in various industries due to their adaptability, ease of deployment, and broad range of applications, including disaster management and emergency response [5][6]. In such scenarios, UAVs provide critical advantages, such as the ability to rapidly access remote or hazardous areas, deliver real-time aerial data to support damage assessment, identify survivors, and coordinate relief efforts [7][8].

Object tracking technology from the UAV perspective is widely adopted and has become a fundamental aspect of UAV technology [9]. Visual target tracking technology is extensively applied in both civilian sectors, including security, logistics, and rescue, as well as in military applications. UAVs have proven effective in search and rescue, rapidly surveying large areas and locating missing individuals [10] [11]. In addition, UAVS are capable of achieving precise measurements even at elevated altitudes [12].

Several studies have focused on object tracking using the You-Only-Look-Once (YOLO) algorithm, which is recognized as a powerful technique in vision-based UAV navigation systems [13]. Research in this domain has demonstrated the efficiency and accuracy of the YOLO algorithm in detecting and tracking objects in real-time [14][5]. The versatility of YOLO enables its effective use in various UAV applications, including environmental monitoring and security enhancement.

Several studies have focused on single object tracking, showcasing how UAVs can be programmed to follow a specific target with precision [16] [17]. These studies highlighted the importance of developing robust algorithms that can maintain accurate tracking even in complex environments with dynamic background changes. The development of such algorithms ensured that UAVs can reliably follow moving targets, whether they are vehicles, animals, or human subjects.

Other research has investigated multiple object tracking, where UAVs are tasked with simultaneously following several targets [18]. This capability is particularly useful in scenarios such as crowd monitoring, wildlife tracking, and coordinated search and rescue operations. By leveraging advanced computer vision techniques, UAVs can differentiate between multiple objects, prioritize targets, and adapt their flight paths accordingly to maintain optimal tracking performance [19][20].

In [21][22][23], vision-based object tracking for UAVs using the Robotic Operating System (ROS) was also examined. These studies contributed to the existing body of knowledge by showcasing different approaches and methods to enhance UAV navigation and tracking capabilities.

Convolutional Neural Networks (CNNs) have been widely employed in tracking systems to identify and follow objects in real-time video streams by extracting key features from each frame. CNNs, modeled after the visual processing mechanisms of the human brain, have played a crucial role in driving major advancements in a range of image-related tasks, including image classification, object detection, segmentation, and others[24] [25].

A review by [26] discussed different object detection models based on Convolutional Neural Networks (CNNs). Among these, MobileNet stood out as a lightweight convolutional neural network architecture optimized for devices with limited computational resources. It employed depth-wise separable convolutions, which significantly reduced the number of parameters and computational demands while maintaining high performance. The Single Shot Detector (SSD) was a type of object detection model that integrated object localization and classification into a single forward pass, using a set of predefined bounding boxes with varying aspect ratios and scales to detect objects at different locations and sizes within an image. In this project, the MobileNet-SSD model was employed, featuring 267 layers and 15 million parameters, to provide real-time inference capabilities on resource-constrained devices like smartphones. Once trained, MobileNet-SSD had a memory footprint of just 63 MB, making it ideal for smaller devices.

This study presents a method for implementing real-time object tracking using a DJI Tello drone. The DJI Tello is a compact quadcopter, weighing only 80 grams, equipped with a 5-megapixel camera capable of capturing 720p video at 30 frames per second [27]. The project applied computer vision techniques, including Convolutional Neural Networks (CNNs), to identify and follow a target in real-time, while dynamically adjusting the drone's flight path to maintain visibility [28].

While models such as YOLO have demonstrated high accuracy, they are often computationally demanding and typically require powerful onboard processors or GPU support, which are not available on low-cost drones. In contrast, this study adopted the MobileNet-SSD model, a lightweight CNN that achieved real-time inference on the resource-constrained Tello EDU drone without additional hardware. This design choice demonstrates the novelty of this work compared to prior YOLO-based implementations by achieving a practical balance between detection accuracy and computational efficiency for autonomous object tracking on a low-cost UAV platform.

II. METHODOLOGY

This section details the hardware and software components used for the object-following system, implemented using the Tello EDU drone (white model). The Tello EDU served as the primary hardware, controlled via a laptop over a Wi-Fi connection. On the software side, Python 3.8 was used for programming, leveraging the OpenCV library [29] [30]. Once the drone successfully took off, the system performed object detection using a lightweight Convolutional Neural Network (CNN), namely the MobileNet-SSD model, which generated bounding-box outputs for the detected object in each video frame.

Figure 1 presents an overview of the object-following system flowchart. As illustrated in the flowchart, when the drone detected an object at the top of its field of view, it ascended. Conversely, if the object appeared near the bottom, the drone descended until it centered the object. When the object was detected to the left or right of the drone's view, the drone adjusted its position accordingly to align itself. If the object was centered, the drone maintained its current altitude. Moreover, the drone moved backward if the object was too close, and moved forward if the object was farther away.

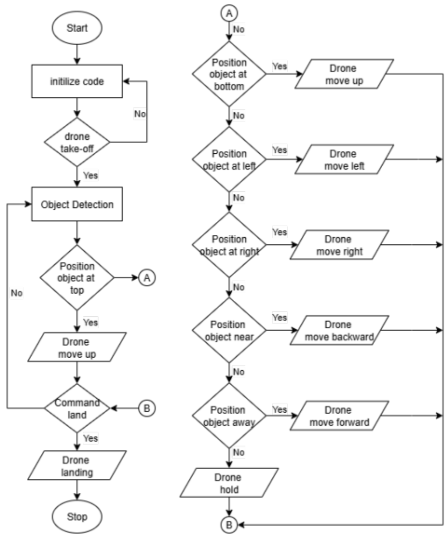


Figure 1. The object-following system flowchart

According to [31], the 3D target coordinates were expressed as a vector $[x_i, y_i, z_i]$, where $x_i = x_{min}$, $y_i = y_{min}$, $z_i = (z_{min} + 150)$. The 150-unit offset was applied to ensure the drone remained above ground, preventing collisions. The index i represented the video frame, where $1 < i < n$, with n representing the final frame in the human motion tracking sequence. To maintain continuous tracking of human movement, a 3D reference point was compared to the target. This reference point, denoted as $[x_0,\ y_0,\ z_0]$, marked the center of the drone's screen, where $x_0 = w/2$, $y_0 = h * w$, and $z_0 = h/2$. Here, w and h represented the width and height of the screen, respectively, with w set to 960 and h to 720, resulting in a 2D resolution of 960 x 720 pixels.

In this work, the target vector represented the center of the detected bounding box, defined as (x_c, y_c), where $x_c = (startX + endX)/2$ and $y_c = (startY + endY)/2$. The benchmark vector corresponded to the reference point at the center of the drone's camera view, denoted as (x_0, y_0, z_0). These vectors were then used to compute displacement values Δx_i, Δy_i, Δz_i , which guided the drone's movements along the horizontal, vertical, and depth axes. The relationship between the target vector and the benchmark vector is illustrated in Figure 2. Additionally, a specific value for y_0 was selected to maintain a safe distance between the target and the DJI Tello EDU drone.

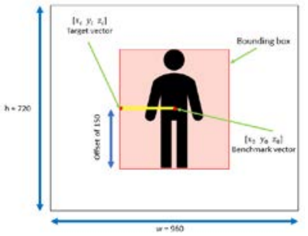


Figure 2. Target vector and benchmark vector

Table 1 presents the pixel-based distance parameters and corresponding drone movements used in the algorithm. On the horizontal axis, when Δx_i was less than -100, greater than 100, or between -100 and 100, the drone executed a left turn, right turn, or maintained its previous position, respectively. Along the depth axis, if Δy_i was below 170000, above 250000, or within the range of 170000 to 250000, the drone moved forward, backward, or held its current position, respectively. On the vertical axis, when Δz_i was less than -55, the drone ascended; if greater than 55, it descended; and if between -55 and 55, it remained at the same height.

Table 1
The distance conditions and corresponding drone movements

No.	Distance Conditions (Pixels)	Drone movement
1	Δx_i < -100	Turn left
2	Δx_i > 100	Turn right
3	-100 < Δx_i < 100	Hold
4	Δy_i < 170000	Move forward
5	Δy_i > 250000	Move backward
6	170000 < Δy_i < 250000	Hold
7	Δz_i < -55	Move up
8	Δz_i > 55	Move down
9	-55 < Δz_i < 55	Hold

Algorithm: Human motion tracking algorithm

Input: Bounding box coordinates $(startX, startY, endX, endY)$
Output: Distance vector = $[x_i\ , y_i\ , z_i]$
Initialize: Δx_0 =0, Δy_0 =0, Δz_0 =0,
x_i = 0, y_i =0, z_i =0, i = 1
$x_i = \frac{w}{2}, y_i = h * w, z_i = \frac{h}{2}$

while bounding box exists **do**
 # Compute bounding box width and height
update a target vector from incoming frames by
$w = int(endX - startX)$ ◊
startX,startY=person detection of starting x, y-axis
$h = int(endY - startY)$ ◊
endX,endY=person detection of ending x, y-axis
$area = h * w$

 # Compute center of bounding box
$x_c = int\big((startX + endX)/2\big)$
$yc = int\big((startY + endY)/2\big)$

 # Define target vector based on center point
$x_i \leftarrow x_c$;
$y_i \leftarrow y_c$;
$z_i \leftarrow (z_{min} + 150)$;

 # Compute displacement relative to reference point (x0, y0, z0)
$\Delta x_i \leftarrow (x_0 - x_i)$;
$\Delta y_i \leftarrow (y_0 - y_i)$;
$\Delta z_i \leftarrow (z_0 - z_i)$;

 # Drone movement rules (same as Table 1)
for each target vector frame i **do**
if Δx_i < -100 **then** the drone turns left
else if Δx_i > 100 **then** the drone turns right
else initialize action of Δx_i = 0
end if

if Δy_i > 250000 **then** the drone moves backward
else if Δy_i < 170000 **then** the drone moves forward
else initialize action of Δy_i = 0
end if

if Δz_i > 55 **then** the drone moves down
else if Δz_i < -55 **then** the drone moves up
else initialize action of Δz_i = 0
end if

end for
end while

A. Horizontal Human Motion Tracking

In the proposed algorithm, the reference vector was initialized at $[0, y_0\ , z_0]$. Suppose the target vector was set to $[-150, y_i, z_i]$, with the target positioned on the left side. This resulted in a distance vector of $[-150, y_i, z_i]$ at a specific time $(T = t)$. Since Δx_i was equal to -150, and according to Condition 1 in Table 1 $(\Delta x_i < -100)$, the drone turned left $(T > t)$. If the target vector was updated to $[-90, y_i, z_i]$, the distance vector became $[-90, y_i, z_i]$. Following Condition 3 in Table 1 $(-100 < \Delta x_i < 100)$, the drone maintained its previous turning-left state. This process is depicted in Figure 3.

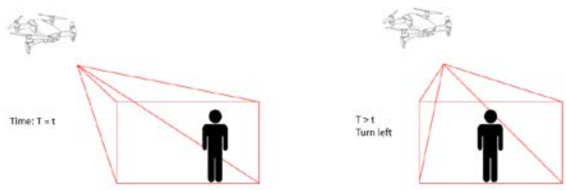


Figure 3. Human motion tracking when the drone turns left

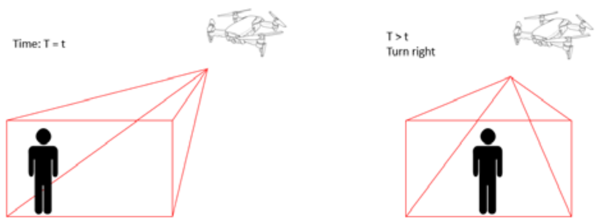


Figure 4. Human motion tracking when the drone turns right

In this algorithm, the benchmark vector was set at $[0, y_0, z_0]$. Considering a target vector of $[150, y_i, z_i]$, where the target remained on the right side, the resulting distance vector was $[150, y_i, z_i]$ at a specific time $(T = t)$. Since Δx_i was 150, and based on Condition 2 in Table 1 $(\Delta x_i > 100)$, the drone turned right at a later time $(T > t)$. Now, if the target vector was updated to $[90, y_i, z_i]$, the new distance vector became $[90, y_i, z_i]$. According to Condition 3 in Table 1 $(-100 < \Delta x_i < 100)$, the drone maintained its previous trajectory of turning right. This situation is depicted in Figure 4.

B. Tracking Human Motion in Forward and Backward Directions

In this proposed algorithm, the reference vector was set to $[x_0, 0, z_0]$. Assuming the target vector was defined as $[x_i, 150000, z_i]$, the resulting distance vector became $[\Delta x_i, 150000, \Delta z_i]$ at a given time $(T = t)$. Since Δy_i was equal to 150000 and fell under Condition 4 in Table 1

($\Delta y_i < 170000$), the drone moved forward at a subsequent time ($T > t$). When the target vector was updated to $[x_i, 180000, z_i]$, it fell within the range of Condition 6 ($170000 < \Delta y_i < 250000$) in Table 1. As a result, the drone continued its forward movement without altering its state. This process is depicted in Figure 5.

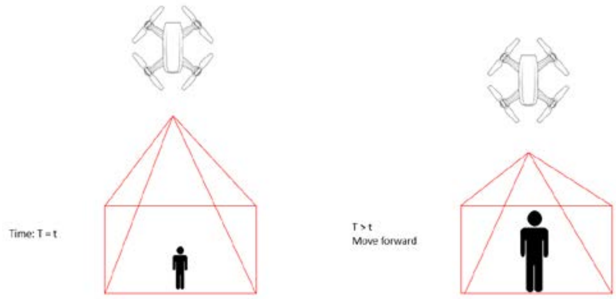


Figure 5. Human motion tracking when the drone moves forward

In this algorithm, the benchmark vector was set at $[x_0, 0, z_0]$, while the target vector was initially defined as $[x_i, 280000, z_i]$. This produced a distance vector of $[\Delta x_i, 280000, \Delta z_i]$ at a specific time ($T = t$). Since Δy_i was equal to 280000 and satisfied Condition No. 5 in Table 1 ($\Delta y_i > 250000$), the drone moved backward at a later time ($T > t$). The target vector was then updated to $[x_i, 240000, z_i]$, and as it fell within the range of Condition 6 ($170000 < \Delta y_i < 250000$) in Table 1, the drone continued moving backward without any further change. This process is depicted in Figure 6.

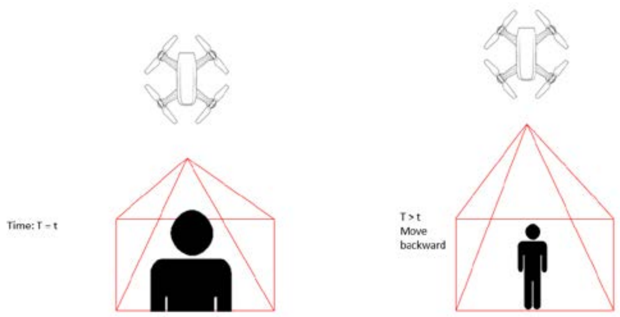


Figure 6. Human motion tracking when the drone moves backward

C. *Vertical tracking of human motion*

In the algorithm, where the reference vector was initialized at $[x_0, y_0, 200]$. Let the target vector be defined as $[x_i, y_i, 100]$. The value of 100 was adjusted by an offset of 150, resulting in a displacement of $[x_0, y_0, -100]$ at time ($T = t$). Since Δz_0 was equal to -100, according to Condition 7 ($\Delta z_0 < -55$) in Table 1, the drone descended at a later time ($T > t$). Subsequently, the target vector was updated to $[x_i, y_i, 150]$, and the displacement became $[x_i, y_i, -50]$. As there was no significant change under Condition 9 ($-55 < \Delta z_i < 55$), the situation remained stable. This scenario is illustrated in Figure 7.

Figure 7. Human motion tracking when the drone moves down

In this algorithm, the benchmark vector was set at $[x_0, y_0, 200]$, while the target vector was defined as $[x_i, y_i, 300]$. The value of 300 included an offset of 150, resulting in a distance of $[x_0, y_0, 100]$ at a given time ($T = t$). Since Δz_0 was equal to 100, and based on Condition 8 ($\Delta z_0 > 55$) in Table 1, the drone ascended at a later time ($T > t$). Subsequently, the target vector was adjusted to $[x_i, y_i, 150]$, and the distance vector became $[x_i, y_i, 50]$. According to Condition 9 ($-55 < \Delta z_i < 55$), no further adjustment was required. This process is depicted in Figure 8.

Figure 8. Human motion tracking when the drone moves up

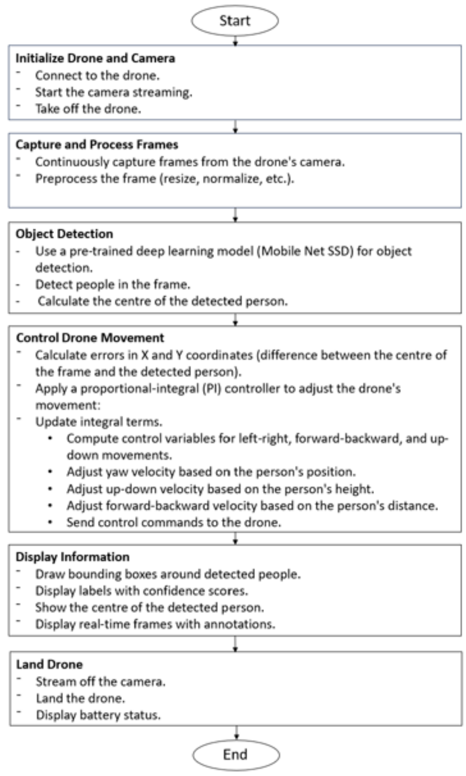


Figure 9. Programming code flowchart

III. RESULT AND ANALYSIS

A. *Optimal Detection Range Between a Person and DJI Tello Drone*

This experiment was conducted to establish the optimal detection range between a person and the DJI Tello drone, as referenced in [32]. The trials were carried out in an outdoor environment. Initially, both the drone and the individual were positioned at the starting point, with the individual subsequently walking towards the designated end point, while the drone remained stationary at the starting location. The total distance between the starting and ending points was set to 10 meters. Figure 10 illustrates the experimental setup. Figure 11 demonstrates that the person remained detectable within a 10-meter range. However, once the individual moved beyond this limit, detection was no longer possible.

Figure 10. Experimental area

Figure 11. Person in the range of 10 m

The distance measurements in this experiment were taken at 1m, 5m, 10m, and greater than 10 m, with detection outcomes represented as (✓) for detected and (X) for not detected. The experiment was repeated over five rounds to ensure the accuracy of the collected data. Table 2 presents the results of person detection at various distances.

Table 2.
Results of the detection of persons in different distances

Round	Distance(m)			
	1	5	10	<10
1	/	/	/	X
2	/	/	/	X
3	/	/	/	X
4	/	/	/	X
5	/	/	/	X

B. Investigation of Light Intensity for Human Motion Detection

This study investigated the effect of light intensity in detecting target motion. The experiment was structured in two phases: one examining indoor light intensity and the other assessing outdoor light intensity [33]. A Lux Light Meter Pro mobile application, installed on a smartphone, was used to measure light intensity [34]. The indoor experiments were conducted during daytime and divided into two scenarios: one with the lamp switched on and the other with the lamp switched off.

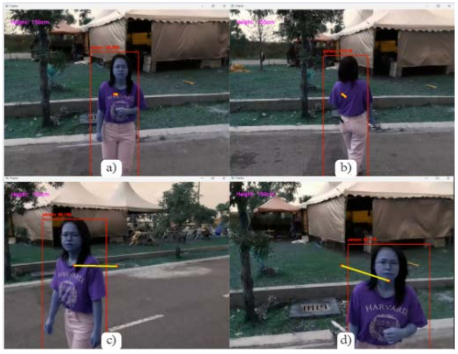
In the lamp-off condition, the maximum light intensity recorded was 26 lux, while in the lamp-on condition, the maximum light intensity reached 34 lux. Figure 12 presents the experimental results for the lamp-off and lamp-on scenarios during forward, backward, left, and right movements, demonstrating clear visibility of the target and facilitating the detection of its motion.

Light off

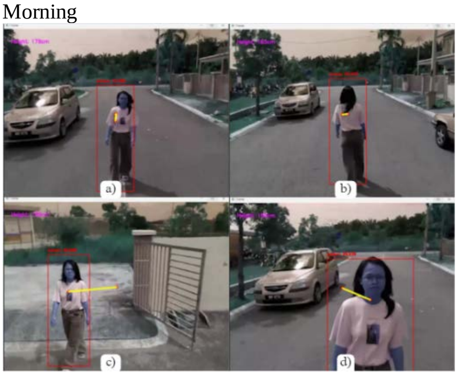
Light on

Figure 12. Person a) moving forward; b) moving backward; c) move left; d) move right

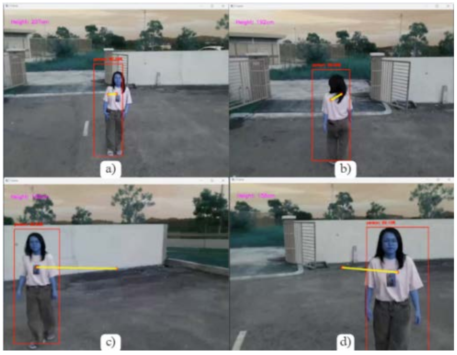
The outdoor experiment was conducted across three distinct time periods: morning (10:00–11:00), afternoon (14:00–15:00), and evening (18:00–19:00). During these intervals, the recorded light intensities were 3085 lux in the morning, 3442 lux in the afternoon, and a maximum of 300 lux in the evening. The experimental results are presented in Figure 13 for the morning, afternoon, and evening sessions, respectively. The visibility of the target was clearly maintained, allowing for effective motion detection as it moved forward, backward, left, and right.



Morning



Afternoon



Evening

Figure 13. Person a) moving forward; b) moving backward; c) moving left; d) moving right

C. Automatic Target Motion Tracking

This section presents an automatic object-tracking system using a DJI Tello drone to follow a moving target [35]. The experiment covers two scenarios. In the first scenario, the drone tracked a person walking along a curved path. In the second, the drone tracked a person within a multi-person environment.

In the first scenario, the curved path is illustrated in Figure 14. The person moved from the designated starting point to the endpoint. Once the drone detected the individual, it tracked their movement along the curved path. The movement of an individual walking along a designated path, as detected by the drone, was captured and recorded. Sixteen sample frames were extracted from the recorded footage. The starting point was labelled as A1, while the endpoint was marked as A16, as shown in Figure 15. Figure 15 illustrates the selected sampling frames A1 through A16 from the recorded video. The drone successfully tracked the person form starting point to endpoint without loss.

In the second scenario, where multiple individuals appeared within the tracking frame, the object-following system prioritized detecting the person nearest to the drone. Initially, with only one person in view, the drone tracked them. However, when a second individual moved closer, the system shifted its focus to the newcomer, as demonstrated in frames B1 to B16 of Figure 16. Frames B3-B4 highlight the target detection process when the second person approached closer than the initial one. If the person being tracked moved out of the frame, the system initiated a search by rotating the drone from the last known position where the target was lost, continuing until the person was detected again.

Although the system demonstrated reliable performance in controlled environments, several limitations remained. First, the MobileNet-SSD occasionally generated multiple bounding boxes in multi-object scenes, which led to ambiguity in target selection and occasional tracking errors. Second, environmental constraints beyond light intensity such as background clutter, partial occlusion of the target, drone vibration, and outdoor wind disturbances could reduce detection stability and flight control accuracy. These challenges highlighted the scope for future improvements, for instance, by integrating multi-object disambiguation strategies and sensor fusion approaches.

Figure 14. Curve motion path

Figure 15. Sampling frames A1-A16 captured from recording video

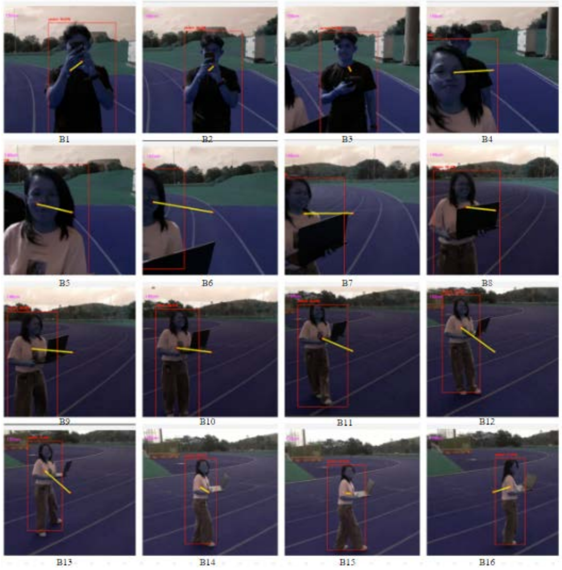


Figure 16. Target detection when a new person comes closer than first person

IV. CONCLUSION

This paper presented the development and implementation of an object-following system using the DJI Tello EDU drone. The project leveraged Convolutional Neural Networks (CNNs), a robust computer vision technique, to perform real-time object detection. The DJI Tello drone was controlled wirelessly through Wi-Fi using Python, with a custom program developed to send flight commands to the drone's flight controller for object tracking. A preliminary review of existing research on object-tracking algorithms provided foundational insights for the project. The key goal was to create an efficient algorithm that enabled the drone to accurately track objects. Multiple versions of the algorithm were designed, tested, and refined to achieve optimal performance. A Proportional-Integral-Derivative (PID) controller was incorporated to ensure the drone maintained a safe distance from the target and avoided obstacles. Finally, a series of experiments were conducted to validate the system's performance. The algorithm was successfully developed and optimized, enabling the DJI Tello drone to autonomously follow objects while maintaining a consistent safe distance. Future work will focus on extending the system to handle multi-object tracking scenarios, improving robustness under outdoor conditions such as wind and occlusion, and exploring sensor fusion or more advanced CNN architectures to enhance detection accuracy and adaptability. These improvements would broaden the applicability of the system and strengthen its performance in real-world deployments.

ACKNOWLEDGMENT

The authors would like to thank, Fakulti Teknologi dan Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka (UTeM), Ministry of Higher Education and the Advanced Academia-Industry Collaboration Laboratory (AiCL) UTeM for supporting this research.

CONFLICT OF INTEREST

Authors declare that there is no conflict of interest regarding the publication of the paper.

AUTHOR CONTRIBUTION

The authors confirm contribution to the paper as follows: study conception and design: Lam Shu Xuan, Aminurrashid Noordin; data collection: Lam Shu Xuan, Aminurrashid Noordin, Izzuddin Mat Lazim; analysis and interpretation of findings: Aminurrashid Noordin, Suziana Ahmad, Azhan AB Rahman, Mohd Ariffanan Mohd Basri; draft manuscript preparation: Lam Shu Xuan, Suziana Ahmad, Mustafa Saad Khalifa. All authors had reviewed the findings and approved the final manuscript.

REFERENCES

[1] Ghazbi, S. Norouzi, Y. Aghli, M. Alimohammadi, and A. A. Akbari., "Quadrotors Unmanned Aerial Vehicles: A Review." International Journal on Smart Sensing and Intelligent Systems 9 (1): 309–33, 2016. https://doi.org/10.21307/ijssis-2017-872.

[2] Tahir, None Muhammad Naveed, None Yubin Lan, None Yali Zhang, None Huang Wenjiang, None Yingkuan Wang, and None Syed Muhammad Zaigham Abbas Naqvi, "Application of Unmanned Aerial Vehicles in Precision Agriculture." In Precision Agriculture, pp. 55–70, 2023. https://doi.org/10.1016/b978-0-443-18953-1.00001-5.

[3] Sujatha, None K., None Npg. Bhavani, VictoSudha George, None T.Kalpatha Reddy, None AKannan, None A. Ganesan, and None Naresh, "AI-Based Controller Design for Autopilot System of UAV." Evergreen 11 (3), 2024. 2004–10. https://doi.org/10.5109/7236847.

[4] Sivakumar, Mithra, and Naga Malleswari Tyj. "A Literature Survey of Unmanned Aerial Vehicle Usage for Civil Applications." Journal of Aerospace Technology and Management 13(1), 2021. https://doi.org/10.1590/jatm.v13.1233.

[5] Telli, Khaled, Okba Kraa, Yassine Himeur, Abdelmalik Ouamane, Mohamed Boumehraz, Shadi Atalla, and Wathiq Mansoor, "A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs)." Systems 11 (8): 400, 2023. https://doi.org/10.3390/systems11080400.

[6] Leira, Frederik S., Håkon Hagen Helgesen, Tor Arne Johansen, and Thor I. Fossen, "Object Detection, Recognition, and Tracking From UAVs Using a Thermal Camera." Journal of Field Robotics 38 (2), pp. 242–67, 2020. https://doi.org/10.1002/rob.21985.

[7] Diaz, Osvaldo Escobar, Abraham Sanchez Lopez, Maria Beatriz Bernabe Loranca, and Rogelio Gonzalez Velazquez, "Real Time Drone Object Tracking Using Histograms of Oriented Gradients and Particle Filters", 2016 Fifteenth Mexican International Conference on Artificial Intelligence (MICAI), 2016. https://doi.org/10.1109/micai-2016.2016.00014.

[8] Zaidi, Ahmad Bilal, and Sadaf Zahera, "Real-time object detection and video monitoring in Drone System." International Research Journal of Engineering and Technology (IRJET) 10 (8): 432, 2023. https://www.irjet.net/archives/V10/i8/IRJET-V10I873.pdf.

[9] Sun, Lifan, Xinxiang Li, Zhe Yang, and Dan Gao, "Visual Object Tracking Based on the Motion Prediction and Block Search in UAV Videos." Drones 8 (6): 252, 2024. https://doi.org/10.3390/drones8060252.

[10] Torrico, Raul Alvarez., "Write an Object Tracking Drone Application - Circuit Cellar." Circuit Cellar. January 29, 2021. https://circuitcellar.com/research-design-hub/write-an-object-tracking-drone-application/.

[11] M. Inti, "Object detection and person tracking using UAV," B. Eng., Institute of Science and Technology, India, 2022.

[12] Zhao, Qiang, and Limei Peng, "High-altitude Multi-object Detection and Tracking Based on Drone Videos." Journal of Networking and Network Applications 2 (1): 36–42, 2022. https://doi.org/10.33969/j-nana.2022.020103.

[13] Lo, Li-Yu, Chi Hao Yiu, Yu Tang, An-Shik Yang, Boyang Li, and Chih-Yung Wen, "Dynamic Object Tracking on Autonomous UAV System for Surveillance Applications." Sensors 21 (23), pp. 7888, 2021. https://doi.org/10.3390/s21237888.

[14] Nawawi, Sophan Wahyudi, Ahmed Ashraf Mohamed Ahmed Abdou, and Nur Azlina Abd Aziz, "View of CNN-based off-board computation for real-time object detection and tracking using a drone." Journal of Tomography System & Sensors Application 4 (2), 2021. https://tssa.my/index.php/jtssa/article/view/165/70.

[15] J. Wu, "Robust Object Recognition and Tracking with Drones," M. Eng, Uppsala University, Sweden, 2024.

[16] Zhao, Xin, Shiyu Hu, Yipei Wang, Jing Zhang, Yimin Hu, Rongshuai Liu, Haibin Ling, et al.. "BioDrone: A Bionic Drone-Based Single Object Tracking Benchmark for Robust Vision." International Journal of Computer Vision 132 (5), pp.1659–84, 2023. https://doi.org/10.1007/s11263-023-01937-0.

[17] Mercado-Ravell, Diego A., Pedro Castillo, and Rogelio Lozano, "Visual Detection and Tracking With UAVs, Following a Mobile Object." Advanced Robotics 33 (7–8), pp. 388–402, 2019. https://doi.org/10.1080/01691864.2019.1596834.

[18] Huang, Wei, Xiaoshu Zhou, Mingchao Dong, and Huaiyu Xu, "Multiple Objects Tracking in the UAV System Based on Hierarchical Deep High-resolution Network." Multimedia Tools and Applications 80 (9), pp. 13911–29, 2021. https://doi.org/10.1007/s11042-020-10427-1.

[19] Liu, Shuai, Xin Li, Huchuan Lu, and You He. 2022. "Multi-Object Tracking Meets Moving UAV", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June, pp. 8866, 2022. https://doi.org/10.1109/cvpr52688.2022.00867.

[20] Yuan, Yubin, Yiquan Wu, Langyue Zhao, Huixian Chen, and Yao Zhang., "Multiple Object Detection and Tracking From Drone Videos Based on GM-YOLO and Multi-tracker." Image and Vision Computing 143 (February), pp. 104951, 2024. https://doi.org/10.1016/j.imavis.2024.104951.

[21] Bartak, Roman, and Adam Vykovsky, "Any Object Tracking and Following by a Flying Drone." 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI), October, pp. 35–41, 2015. https://doi.org/10.1109/micai.2015.12.

[22] Fadhillah, Ahmad Haris Indra, Ahsanu Taqwim Safrudin, and Surya Darma, "Unmanned Aerial Vehicle Object Tracking and Following." Journal of Physics Conference Series 1528 (1): 012018, 2020. https://doi.org/10.1088/1742-6596/1528/1/012018.

[23] Fadhillah, Ahmad Haris Indra, Ahsanu Taqwim Safrudin, and Surya Darma, "Unmanned Aerial Vehicle Object Tracking and Following." Journal of Physics Conference Series 1528 (1): 012018, 2020. https://doi.org/10.1088/1742-6596/1528/1/012018.

[24] Gk, Dayananda, None Sreerama Samartha Jg, and Vayusutha M, "AI-Enabled Automatic Attendance Monitoring Systems." Evergreen 11 (2), pp. 913–26, 2024. https://doi.org/10.5109/7183374.

[25] Taye, Mohammad Mustafa, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions." Computation 11 (3(, pp. 52, 2023. https://doi.org/10.3390/computation11030052.

[26] Sultana, F., A. Sufian, and P. Dutta, "A Review of Object Detection Models Based on Convolutional Neural Network." Advances in Intelligent Systems and Computing, January. pp.1–16, 2020. https://doi.org/10.1007/978-981-15-4288-6_1.

[27] Cummings, Myles. "DJI Ryze Tello Review." Space.Com. January 25, 2022. Accessed March 12, 2025. https://www.space.com/DJI-ryze-tello-review.

[28] Ghosh, Anirudha, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, and Debashis De., "Fundamental Concepts of Convolutional Neural Network." In Intelligent Systems Reference Library, pp. 519–67, 2019. https://doi.org/10.1007/978-3-030-32644-9_36.

[29] PythonGeeks Team, "What is OpenCV? – An Introduction Guide." https://pythongeeks.org/what-is-opencv (Accessed Dec. 16, 2024).

[30] Boonsongsrikul, Anuparp, and Jirapon Eamsaard. "Real-Time Human Motion Tracking by Tello EDU Drone." Sensors 23 (2), pp. 897, 2023. https://doi.org/10.3390/s23020897.

[31] Diez-Tomillo, Julio, Jose Maria Alcaraz-Calero, and Qi Wang, "Dynamic-Distance-Based Thresholding for UAV-Based Face Verification Algorithms." Sensors 23 (24(, pp. 9909, 2023. https://doi.org/10.3390/s23249909.

[32] Li, Xi, Noam Levin, Jinlong Xie, and Deren Li, "Monitoring Hourly Night-time Light by an Unmanned Aerial Vehicle and Its Implications to Satellite Remote Sensing." Remote Sensing of Environment 247 (June), pp.111942, 2023. https://doi.org/10.1016/j.rse.2020.111942.

[33] S. Buddies and S. Buddies, "Science with a Smartphone: Lux Meter | STEM Activity," Science Buddies, Mar. 02, 2021. https://www.sciencebuddies.org/stem-activities/science-with-a-smartphone-lux-meter.

[34] Alansari, Mohamad, Oussama Abdul Hay, Sara Alansari, Sajid Javed, Abdulhadi Shoufan, Yahya Zweiri, and Naoufel Werghi, "Drone-Person Tracking in Uniform Appearance Crowd: A New Dataset." Scientific Data 11 (1), 2024. https://doi.org/10.1038/s41597-023-02810-y.