Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

# Real-time Object Tracking Control of a Quadcopter Using YOLOv5 and Kalman Filter

***Zahra Hassani*** , ***Vahab Nekoukar*** *

*Electrical Engineering School, Shahid Rajaee Teacher Training University, Tehran, Iran.*

## Article Info

**Article History:**
Received 10 July 2025
Reviewed 03 September 2025
Revised 22 September 2025
Accepted 27 September 2025

**Keywords:**
Deep learning
Machine vision
Quadcopter
Target tracking
Unmanned aerial vehicles
Yolo

*Corresponding Author's Email Address: *v.nekoukar@sru.ac.ir*

## Abstract

**Background and Objectives:** Currently, the control engineering community is increasingly focusing on research related to Unmanned Aerial Vehicles (UAVs) due to their versatile capabilities. Among the various applications, target detection and tracking stand out as crucial. Recent advancements in Artificial Intelligence (AI) and Deep Learning (DL) have the potential to enhance the synergy between vision and control in UAV operations. By integrating AI algorithms with control methods, the accuracy of target information can be significantly improved in UAVs. This research introduces an autopilot system for quadcopters to search for and track a predetermined target.

**Methods:** The autopilot system utilizes the YOLO network, a robust convolutional neural network-based system, for real-time target detection. To enhance object tracking robustness, the Kalman filter is integrated into the system. Furthermore, Proportional-Derivative (PD) controllers are utilized to calculate suitable control commands, enabling the quadcopter to effectively track both stationary and moving targets. Additionally, an object retrieval strategy is proposed to locate and recover lost objects during the tracking phase.

**Results:** The effectiveness of the proposed system was evaluated through real-time experimental trials involving diverse scenarios encompassing both stationary and moving targets. The integration of the YOLOv5 network with the Kalman filter substantially improved detection accuracy and stability. Furthermore, the object retrieval mechanism demonstrated high reliability in recovering lost targets, thereby increasing overall system resilience. The PD-based control scheme enabled responsive and precise trajectory adjustments, contributing to consistent target tracking performance across all test cases.

**Conclusion:** Integration of a YOLOv5-based detection module, a Kalman filter for robust tracking, and PD controllers for flight control provides an autonomous quadcopter system capable of detecting and tracking both stationary and moving targets with unknown dynamics. The proposed approach shows promise for real-time autonomous tracking applications and offers a foundation for future development in more complex, outdoor scenarios.

This work is distributed under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

**How to cite this paper:**
Z. Hassani, V. Nekoukar, "Real-time object tracking control of a quadcopter using YOLOv5 and Kalman filter," J. Electr. Comput. Eng. Innovations, 14(1): 217-230, 2026.

**DOI:** 10.22061/jecei.2025.12231.865

**URL:** https://jecei.sru.ac.ir/article_2428.html

2345- 3044/© 2026 The Authors. Published by Shahid Rajaee Teacher Training University.

## Introduction

In recent years, Unmanned Aerial Vehicles (UAVs), particularly quadcopters, have garnered significant attention from researchers due to their diverse capabilities and sizes. Remarkable advancements in Artificial Intelligence (AI) and Computer Vision (CV) have strengthened the link between vision and control for quadcopters. As a result, quadcopters find applications in various domains, including autonomous navigation [1, 2], medical care [3], exploration and mapping [4], path planning [5], search and rescue[6]-[8], object tracking [9], [10], traffic monitoring [11], and obstacle detection [12]. In vision-based quadcopter applications, visual object tracking has become increasingly ubiquitous and represents a critical task [13].

The wide range of applications underscores the critical need to address the fundamental challenges of object detection and tracking. A thorough understanding of the fundamental principles and challenges related to this field is essential for addressing real-world problems and developing practical solutions. Object detection involves localizing and classifying objects within images or video sequences. On the other hand, object tracking focuses on monitoring the movement of a specific object over a time period. In recent years, researchers have increasingly emphasized the significance of object detection and tracking in the context of quadcopter drones [14].

Various methodologies have been developed in the domain of object detection and tracking. Before the arise of Deep Learning (DL) algorithms, traditional object detectors were used for object detection. They typically include three-stage process: 1) proposal generation; 2) feature vector extraction; and 3) region classification. They have limitations, including many redundant proposals during proposal generation, this leads to false positives during classification. Furthermore, each stage of the detection process is designed and optimized independently. So, this approach makes it challenging to achieve a globally optimal solution for the entire system [15].

DL is increasingly gaining attention as a promising area of research for enhancing several applications, such as autonomous vehicles [16] and especially UAVs [17] capabilities. In response to the rapid advancement of DL algorithms, the performance of object detection and tracking has been incredibly improved. DL algorithms have become popular because of two important causes: 1) Availability and abundance of data for processing and 2) Availability of high-end computational resources [18]. In general, object detection algorithms include two approaches: one-staged and two-stage [19]-[21]. For real-time object detection and tracking, the speed of the detection is very important. While two-stage algorithms can provide some assistance with real-time object detection, but it is still impractical for use in some embedded platforms (e.g., vehicle, UAV and etc.) [22]. One-stage algorithms such as Single Shot Multi-Box Detection (SSD) [23], You Only Look Once (YOLO) [24], as well as other versions of the YOLO network [25], [26], are faster than two-stage algorithms like Region Convolutional Neural Networks(R-CNN) [27], Fast Region-Based CNN(Fast R-CNN) [28] and Faster Region-Based CNN (Faster R-CNN) [29]. Among the one-stage algorithms, the YOLO family are particularly famous for capability to detect and classify objects using forward propagation without the need to create proposal regions like two-stage algorithms [30]. Also, due to its fast and accurate detection, the YOLO algorithm has huge potential for being used in real-time applications.

Recent studies have explored the integration of YOLO with the Kalman filter to enhance real-time object detection and tracking performance. Barisic et al [31] developed a system utilizing a YOLO-based convolutional neural network for detecting multirotor UAVs in various environments, enhanced detection by incorporating a Kalman filter, which improves the reliability of position and velocity data. A nonlinear controller based on visual servoing maintains the UAV within the Field of View (FOV) and at the desired distance. Alshaer et al [32] proposed a two-stage system using deep learning and Kalman filter techniques for detecting and tracking consumer-grade UAVs, the study evaluates YOLO models for detection and employs both the Kalman filter and Extended Kalman filter for tracking, resulting in improved real-time tracking accuracy. Likewise, the concentration of many research papers is on object tracking. Zhao *et al*. [33] proposed a framework for moving vehicle detection, tracking, and geolocation. They utilized the YOLOv3 model for vehicle detection. Additionally, a correlation filter was applied to the task of target tracking. Furthermore, they introduced a flight controller based on the results of visual tracking and geolocation to maintain targets within the FOV. The proposed framework was evaluated experimentally. Mercado *et al*. [34] suggested a system for tracking objects with unknown dynamics. A Haar cascade classifier was applied to detect objects. A Kalman filter was employed to estimate the relative position of objects while the drone and target were maintained at a constant distance by using a linear controller. Mokhtari *et al*. [35] developed a system for target tracking with a quadcopter that consisted of four Proportional-Integral-Derivative (PID) controllers and the YOLOv3 model for object detection. Two control goals were achieved: keeping the target in FOV and maintaining the distance between the quadcopter and the target. Dursun *et al*. [36] suggested a distributed system for recognizing and

tracking individuals. They employed the YOLOv2 model to handle object detection. Users set the target; the object information vector was transformed into error signals. Proportional-Derivative (PD) controllers generated control signals to track objects that were embedded within the drone. Kuoh *et al*. [37] proposed a closed-loop end-to-end target tracking system using visual control steering. To address real-time object tracking, they introduced a combination of an adopted CNN and a single-shot multi-box detection. The error was calculated based on the object location and the central coordinate within the image. The closed-loop system was designed by PID controllers.

Previous studies have predominantly relied on simulation studies, lacking validation through actual flight tests and overlooking solutions for target loss during tracking and subsequent recovery. This paper presents an intelligent system designed for real-time search, detection, and tracking of moving objects using a quadcopter, along with a method for target retrieval in case of loss during the tracking phase. The closed-loop system integrates YOLOv5, a renowned model for target detection recognized for its speed and accuracy balance, with two PD controllers serving as the autopilot for the quadcopter. Initially, the system explores the environment following a predefined flight path and captures images through a camera mounted on the quadcopter. These images are then transmitted to the ground station for processing by the YOLOv5 model. Upon target detection, the search phase is terminated, and the tracking phase begins. Throughout the tracking phase, the autopilot system computes Flight Commands (FCs) based on tracking errors using the two PD controllers and wirelessly transmits FC to the quadcopter. Autopilot endeavors to keep the target within the FOV of the quadcopter. In cases of target loss, the system directs the quadcopter to recover the lost target. Two scenarios are considered. In the first scenario, either the target is lost within a few frames, or multiple targets are identified in a single frame. In such instances, a Kalman filter is proposed for estimating the target's position. However, if the count of frames where the target is not detected surpasses a set threshold, the quadcopter reverts to the search phase to locate the target. The Kalman filter design operates under the assumption that the target moves with constant acceleration. The key contributions of this work are as follows:

1) Improving the robustness of the object detection algorithm by integrating YOLOv5 with the Kalman filter.

2) Implementing real-time position control for a quadcopter to track a moving object with unknown dynamics, with no need for target prior information and the quadcopter's attitude and position.

## Quadcopter Dynamics

Understanding the concept of a quadcopter's dynamics is crucial for studying its applications. The dynamics of a quadcopter in the body frame involves the interaction of forces, torques, and motion states, which play a pivotal role in the quadcopter's stability, control, and overall flight performance. The axes of this frame are defined as $X_B$, $Y_B$, and $Z_B$, which are shown in Fig. 1. By comprehending these dynamics relative to the body frame, engineers and researchers can effectively develop control strategies, motion planning algorithms, and state estimation techniques necessary for reliable quadcopter operation in diverse environments and applications.

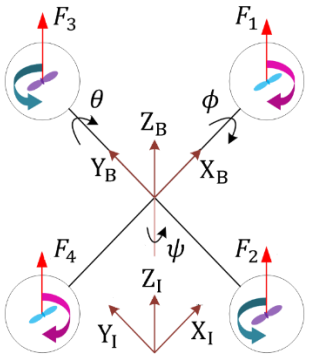


Fig. 1: Configuration of a typical quadcopter.

The quadcopter has four rotors ($i$ = 1 to 4), and each rotor generates a thrust vector $F_i$ exerted on the quadcopter's body frame. Every thrust vector generates a torque vector $T_i$ around the mass center of the quadcopter [38]. By adjusting the speed of the rotors, the quadcopter can move along all three axes. The rotation of the quadcopter around the $X_I$, $Y_I$ and $Z_I$ axes are defined by the roll angle $\phi$, pitch angle $\theta$ , and yaw angle $\psi$, respectively [38]. Also, $T_\phi$, $T_\theta$, and $T_\psi$ denote the torques generated by the four motors around $X_B$, $Y_B$, and $Z_B$, respectively. The thrust vector $F_i$ generated by the *i*th rotor is given by (1).

$$F_i = b_i\omega_i^2 \quad i = 1, 2, 3, 4 \tag{1}$$

where $b_i$ is an unknown positive constant that can be determined by static thrust tests [39] and $\omega_i$ is the angular speed of the $i$th rotor. The total thrust vector $T$ and the resultant torque vectors in the body frame can be written as (2)-(5).

$$T = F_1 + F_2 + F_3 + F_4 \tag{2}$$

$$T_\phi = l(F_2 - F_4) \tag{3}$$

$$T_\theta = l(F_3 - F_1) \tag{4}$$

$$T_\psi = c_p(F_2 + F_4 - F_1 - F_3) \tag{5}$$

The variable $l$ represents the distance from the center of mass to the center of the rotors, and $c_p$ denotes an unknown parameter. By employing the Newton-Euler

method, the dynamic equations of the quadcopter's body can be expressed. (6) and (7) show the dynamic equations of the quadcopter.

$$ma = F_p + F_g + F_a + F_r \tag{6}$$

$$J\dot{\Omega} = T_g + T_p + T_a + T_r - \Omega \times \mathrm{J}\Omega \tag{7}$$

that $m$ is the quadcopter's mass, $a$ denotes the linear acceleration of the mass center in the inertial frame, and $J$ represents the moment of inertia matrix. Also, $F_a = [F_{ax}, F_{ay}, F_{az}]^{\mathrm{T}}$ denotes unknown aerodynamic, and $F_r = [F_{rx}, F_{ry}, F_{rz}]^T$ refers to residual forces. $F_g$ and $F_p$ are the gravitational and motor produced forces, respectively. $T_a = [T_{a\phi}, T_{a\theta}, T_{a\psi}]^T$ shows unknown aerodynamic acting on the canter of mass, and $T_r = [T_{\mathrm{r}\phi}, T_{\mathrm{r}\theta}, T_{\mathrm{r}\psi}]^T$ defines residual torques acting on the canter of mass. $T_g$ and $T_p$ are the gravitational and motor produced torques, respectively.

The angular velocity of the quadcopter in the body frame ($\Omega$) and the inertia matrix ($J$) are determined by (8) and (9).

$$\Omega = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{8}$$

$$J = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{bmatrix} \tag{9}$$

and

$$F_g = [0 \quad 0 \quad -mg]^{\mathrm{T}} \tag{10}$$

$$F_P = \begin{bmatrix} \cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi \\ \cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi \\ \cos\phi\cos\theta \end{bmatrix}^{\mathrm{T}} \tag{11}$$

$$T_g = [-qI_r\Omega_r \quad pI_r\Omega_r \quad 0]^{\mathrm{T}} \tag{12}$$

$$T_p = [\mathrm{T}_\phi \quad T_\theta \quad T_\psi]^{\mathrm{T}} \tag{13}$$

that $g$ is the gravitational constant, $I_r$ is the inertial moment of rotors and $\Omega_r = \sum_{i=1}^{4} \omega_i$. Using (2), (6), (10) and (11), the linear motion equation of the quadcopter can be obtained by

$$\ddot{x} = \frac{1}{m}[(\cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi)T + F_{ax} + F_{rx}] \tag{14}$$

$$\ddot{y} = \frac{1}{m}[(\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi)T + F_{ay} + F_{ry}] \tag{15}$$

$$\ddot{z} = \frac{1}{m}[(\cos\phi\cos\theta)T - mg + F_{az} + F_{rz}] \tag{16}$$

The equation governing angular motion can be derived in a manner like (14)-(16), but it tends to be quite intricate. To simplify matters, the angular motion equation is demonstrated as (17).

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = f_r(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) + g_r(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})[T_g + T_p + T_a + T_r] \tag{17}$$

where $f_r$ and $g_r$ are two nonlinear matrix functions of Euler angles and their first derivatives. If it is assumed that rotor's speed controllers exhibit gain-like behavior and the square of the rotors' angular velocity are taken as the control input ($u_i = \omega_i^2$), then (17) can be reformulated in a compact form as (18).

$$\ddot{x}_r = f_r(x_r, \dot{x}_r) + g_r(x_r, \dot{x}_r)\, u \tag{18}$$

where $x_r = [\phi, \theta, \psi]^{\mathrm{T}}$ is state vector, $u = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$ is the control input, and $f_r$ and $g_r$ are unknown matrix functions.

The flight controller of the quadcopter adjusts the rotors' speed based on commands received from either the pilot or the autopilot system. In fact, the flight controller determines the duty cycle of the Pulse-Width Modulation (PWM) signals that control the rotors' speed. The pilot sends four different commands: Throttle, which controls the rotation speed of the motors; Yaw, which manages rotation around the vertical axis ($Z_B$); Pitch, which handles tilting or moving forward and backward; and Roll, which is responsible for tilting or moving left and right. Consequently, the autopilot system must be capable of generating these four types of commands.

## Methodology

This section presents the proposed autopilot system, and the methodologies employed. The quadcopter's flight operation is divided into two distinct phases: search and track, with object detection serving as a critical component in both. To facilitate real-time object detection, the system first establishes a reliable communication link between the quadcopter and the processing unit. In the proposed system, images captured by the quadcopter's onboard camera are wirelessly transmitted to a personal computer (PC) via a Wi-Fi connection. Subsequently, the target is detected, search or tracking commands are generated, and control signals are computed and transmitted wirelessly back to the quadcopter. These control signals are essential for the accurate execution of the quadcopter's autonomous operations. The communication architecture between the quadcopter and PC, which functions as the ground station, is illustrated in Fig. 2. This wireless link enables real-time image processing and the transmission of control commands necessary for object tracking.

Following the communication architecture, the operational flow of the proposed autopilot system is illustrated in Fig. 3.

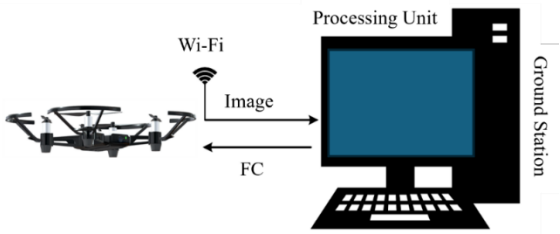


Fig. 2: Communication structure between the quadcopter and PC.

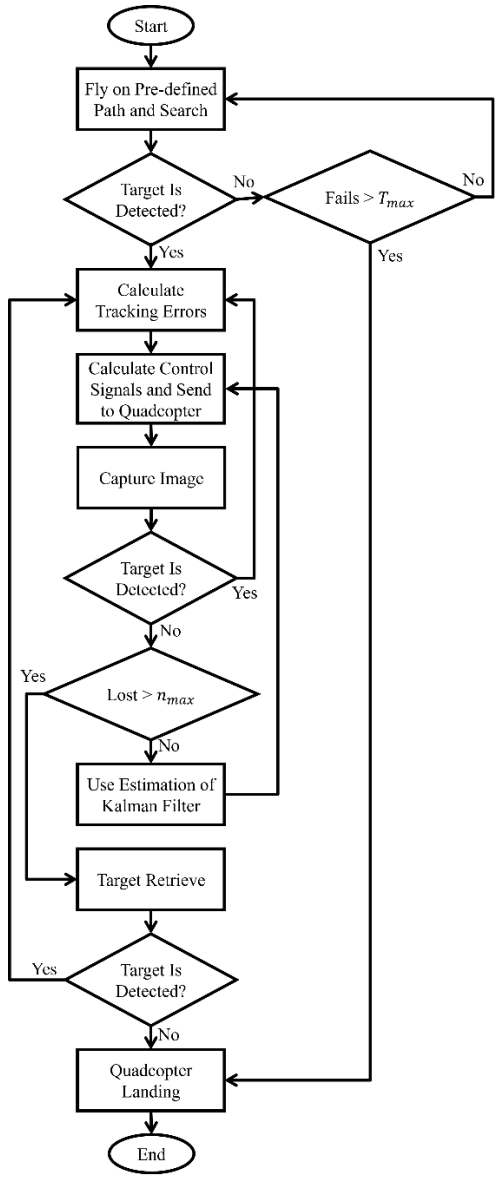


Fig. 3: Flowchart of the proposed autopilot system.

The system consists of three interconnected subsystems: target detection, target search, and target tracking. After take-off, the quadcopter enters the search phase and transitions to the tracking phase once the target is successfully detected. During tracking, a Kalman filter and a target-retrieving algorithm are employed to maintain accuracy, particularly in cases of temporary target loss. The pseudocode of the proposed method is shown in pseudocode 1. A detailed description of each subsystem is provided below.

```
Initialize system parameters
Initialize PD controller gains
Initialize Kalman filter states
While system is running:
  Acquire image frame from onboard camera
  Detect target using YOLO network
  If target is detected:
    Extract target position (bounding box center)
    Update Kalman filter with measurement
    Compute control error
    Generate control commands using PD controller
    Send commands to quadcopter
    Reset loss counter
  Else: Target not detected
    Increment loss counter
  If L <= n_max:
    Predict target position using Kalman filter
    Compute control error
    Generate control commands using PD controller
    Send commands to quadcopter
  Else if time_since_loss <= T_max:
    Predict target position using Kalman filter
    Compute control error
    Generate control commands using PD controller
    Send commands to quadcopter
  Else:
    initiate search behavior
End While
```

Pseudocode 1: The pseudocode of the proposed method.

*Target Detection*

Target detection plays a pivotal role in the proposed system, as it is integral in both the search and tracking phases. In the context of object tracking, the process begins with localizing the target by identifying a two-dimensional Bounding box (B-box) within the image. This process is achieved using a detection model, which generates the B-box and enables estimating the target state [10]. The increasing deployment of UAVs has underscored the demand for detection models that provide both high speed and accuracy in real-time applications [40]. However, limitations in image resolution and computational resources pose challenges, particularly in detecting small objects. To address these constraints, this paper employs the YOLO network, a state-of-the-art object detection model. Among its variations, YOLOv5 is recognized for its optimal trade-off between speed and accuracy in real-time scenarios [41].

YOLOv5 and its successors offer a compelling balance between speed and computational efficiency compared to earlier YOLO versions and other object detection models. While older models like YOLOv3 and YOLOv4 provide strong accuracy, they are heavier and slower, making them less suitable for real-time or edge

applications. YOLOv5, particularly the smaller variants, significantly reduces model size and increases inference speed, enabling high-performance real-time detection. Newer versions such as YOLOv6–v8 further optimize this trade-off, delivering ultra-fast processing without substantial loss of accuracy. In contrast, two-stage detectors like Faster R-CNN and transformer-based models like DETR achieve higher precision but at the cost of much greater computational complexity and slower inference, highlighting YOLO's strength for applications where speed and efficiency are critical.

YOLOv5 is offered in multiple configurations, such as small, medium, large, and extra-large models, allowing customization based on performance requirements. Additionally, it supports various input image scales, enhancing detection performance a range of object sizes and aspect ratios [42]. The architecture of YOLOv5, illustrated in Fig. 4, comprises three main components: 1) Backbone-CSPDarknet; 2) Neck-PANet; and 3) Head-YOLO Layer.

The backbone architecture typically incorporates well-established CNN models such as VGG, ResNet, DenseNet, MobileNet, EfficientNet, and CSPDarknet53. In the neck component, PANet performs feature aggregation through both bottom-up and top-down pathways, facilitating multi-scale feature fusion via up-sampling and down-sampling operations [41]. The head of the model comprises three convolution layers responsible for predicting the B-box coordinates, object confidence scores, and class probabilities. The input image undergoes a two-stage process: initial feature extraction by CSPDarknet53, followed by further feature fusion through PANet. Ultimately, the YOLO layer utilizes the fuzzed features to generate object detection outputs [43].

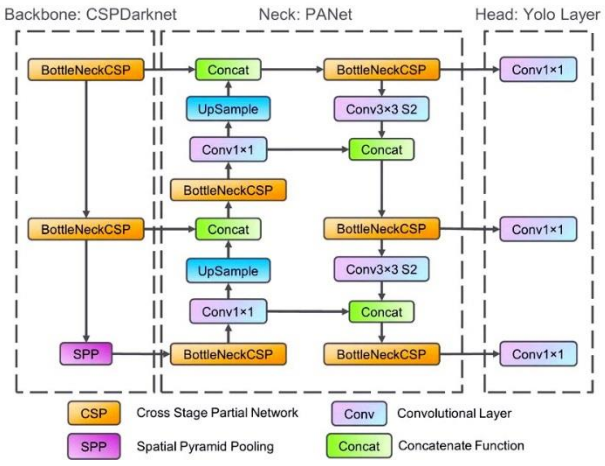


Fig. 4: Network architecture of the Yolov5 [44].

YOLOv5 demonstrates improved performance over its predecessor, YOLOv3. In YOLOv3, feature extraction was performed using Darknet19, which exhibited limitations in detecting small objects [43]. YOLOv5 addresses this issue by employing CSPDarknet for enhance feature extraction. it divides the input image into an $s * s$ grid, with each grid cell responsible for detecting objects whose center lies within that cell. The model then generates a B-box containing key information about the object's location in the captured image. the output of YOLOv5 and example of the B-box information are illustrated in Fig. 5.

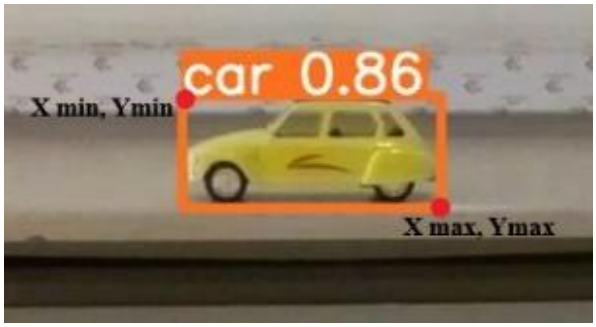


Fig. 5: Information about the YOLOv5 B-box.

Following target detection, the central coordinates of the B-box (critical for enabling precise tracking) are calculated using the positional data provided by the detection algorithm, as formulated in (19) and (20).

$$x_{bbox\_cent} = \frac{x_{min} + x_{max}}{2} \tag{19}$$

$$y_{bbox\_cent} = \frac{y_{min} + y_{max}}{2} \tag{20}$$

The parameters $\boldsymbol{x_{min}}$, $\boldsymbol{y_{min}}$, $\boldsymbol{x_{max}}$, and $\boldsymbol{y_{max}}$ denote the top-left and bottom-right corner coordinates of the B-box, respectively as illustrated in Fig. 3.

In the initial phase, a custom YOLOv5 model was trained using a dataset sourced from Kaggle, supplemented with extensive data augmentation techniques to increase the dataset's diversity and improve model generalization. However, despite these efforts, the performance of the custom-trained model was inconsistent and failed to match the robustness and accuracy of the pre-trained YOLOv5 model. The pre-trained model, developed on the COCO dataset (a large-scale, diverse benchmark widely regarded in the computer vision community) demonstrated superior generalization across varied object categories and environmental conditions. Moreover, training from scratch on a limited or domain-specific dataset reduced adaptability in real-world scenarios. To ensure high detection accuracy and computational efficiency the pre-trained YOLOv5 model was adopted as the final configuration for object detection in this paper. Furthermore, the integration of the YOLOv5 detection

model with a Kalman filter significantly enhances the object tracking capability. This fusion approach improves tracking accuracy and stability, particularly in scenarios involving occlusions or temporary target loss.

*Kalman Filter*

The main limitations of the object detection algorithm are false positive detections and missed detections. A false positive detection happens when the system mistakenly identifies a condition or attribute as present when it isn't. On the other hand, missed detections, also referred to as false negatives, occur when the system fails to recognize a condition or attribute that is indeed present. In this paper, to deal with these limitations, the Kalman filter are employed.

The kinematic of the target can be written as (21).

$$\frac{d}{dt}\begin{bmatrix} x_{bbox\_cent} \\ vx_{bbox\_cent} \\ y_{bbox\_cent} \\ vy_{bbox\_cent} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{bbox\_cent} \\ vx_{bbox\_cent} \\ y_{bbox\_cent} \\ vy_{bbox\_cent} \end{bmatrix} + \begin{bmatrix} 0 \\ ax_{xbbox\_cent} \\ 0 \\ ay_{bbox\_cent} \end{bmatrix} \quad (21)$$

that $vx_{bbox\_cent}$, $vy_{bbox\_cent}$, $ax_{bbox\_cent}$ and $ay_{bbox\_cent}$ are the speed and acceleration of the B-box's center point, respectively. The acceleration is considered as process white noise.

At each step of target tracking, one of four states may occur:

1) Single Target Detected: The position of the B-box's center is calculated using (19) and (20).

2) Multiple Target Detected: The position of the B-box's center is estimated using the Kalman filter, and the target whose center is closest to the estimated position is selected.

3) No Target Detected (Short Duration): If no target is detected and the target has been missing for fewer than $n_{max}$ consecutive frames, the estimated position of the B-box's center is used.

4) No Target Detected (Long Duration): If no target is detected and the target has been missing for $n_{max}$ consecutive frames, target tracking is stopped, and the quadcopter begins searching for the target using the method described in section IV.

## Target Search

In the proposed system, target search is performed while the quadcopter follows a predefined yet arbitrary flight path, which introduces no constraints on trajectory designed. Prior to initiating the search phase, the desired target such as human, ball, or vehicle must be specified at the ground station. The quadcopter begins in a stationary position on the ground, and the autopilot system issues throttle commands to initiate takeoff.

Upon reaching an altitude of one meter, the quadcopter stabilizes and commences the search operation. During this phase, the quadcopter executes a continuous yaw rotation at a rate of $n$ revolutions per minute, controlled by yaw commands generated by the autopilot system. At each sampling time, the onboard camera captures an image frame and transmits it wirelessly to the ground station.

The detection algorithm as described in section III.A is applied to each frame to determine the presence of the target. Once the target is successfully identified, the quadcopter halts its rotation, marking the end of the search phase. Conversely, If the quadcopter fails to detect the target after $T_{max}$ rotations, the system terminates the search and initiates a controlled landing procedure at the original take-off location. Alternatively, this feature can be leveraged during the search phase in scenarios where the target is not predetermined, enabling the user to manually select the target from the images transmitted by the quadcopter to the ground station.

*Target Tracking*

This section presents the proposed autopilot framework for object tracking. As illustrated in Fig. 6, the closed-loop system integrates YOLOv5 for target detection, a Kalman filter for estimation of the target's position and PD controllers for tracking the target. Once the desired target is identified in an image, the search phase is terminated, and the target tracking phase commences. The principal aim of the tracking operation is to accurately position and continuously maintain the target within the FOV of the quadcopter.

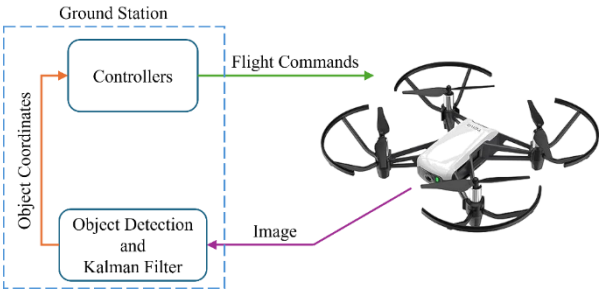


Fig. 6: Proposed closed-loop system of target tracking.

To facilitate the tracking process, it is essential to quantify the positional error between the target and the center of the image frame. Fig. 7 illustrates the image coordinates with the detected target. The tracking error is defined as the Euclidean distance between the center of the FOV and the centroid of the bounding box (B-box) encompassing the target, as shown in Fig. 7. This error metric serves as the input to the PD controllers, which generate corrective control signals to adjust the quadcopter's position and maintain the target centrally aligned within the FOV.

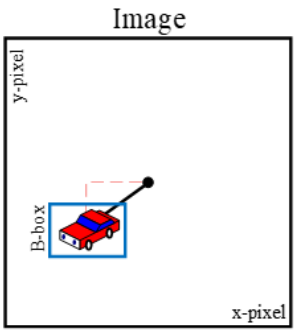


Fig. 7: System coordinates.

If $(Img_{xcent}, Img_{ycent}) = (0,0)$ denote the coordinates of the FOV's center, then the tracking error signals are computed using (22), (23)

$$e_x(t) = x_{Bbox\_cent} - Img_{xcent} = x_{Bbox\_cent} \quad (22)$$

$$e_y(t) = y_{Bbox\_cent} - Img_{ycent} = y_{Bbox\_cent} \quad (23)$$

where $(x_{Bbox\_cent}, y_{Bbox\_cent})$ represent the coordinates of the B-box center. These error signals are utilized as inputs to the autopilot system, which modulates the FCs through two PD controllers in order to maintain the target within the FOV. The PD controllers are responsible for generating control actions that guide the quadcopter's motion along the vertical and horizontal axes, corresponding to upward, downward, leftward, and rightward movements. At each sampling time, upon computation of the tracking error, the quadcopter updates its control actions based on the error magnitude and direction. Consequently, the FCs are derived according to (24) and (25).

$$FC_{up-down}(t) = K_p e_y(t) + K_D \dot{e}_y(t) \quad (24)$$

$$FC_{left-right}(t) = K_p e_x(t) + K_D \dot{e}_x(t) \quad (25)$$

here, $FC_{up-down}$ and $FC_{left-right}$ correspond to the the throttle and roll FCs, respectively.

*Target Retrieving*

During object tracking operations, there may be instances where the target becomes undetected or is lost in some frames. While most existing studies overlook this issue, the absence of a retrieval mechanism can significantly hinder the continuity and reliability of the tracking process. This limitation becomes especially critical in mission-critical scenarios such as search and rescue operations or high-value industrial and commercial applications. Failure to retrieve a lost target in such cases could pose severe safety risks or result in substantial financial losses [45]. Therefore, implementing a robust target retrieval mechanism is imperative for enhancing tracking reliability. Target retrieval methods typically fall into two categories: utilizing new positional data or leveraging historical information. In this paper, a retrieval strategy based on previously recorded error signals is proposed. When the target is lost for a duration exceeding a specified threshold, the last known error signal values are used to guide the quadcopter's motion in an attempt to reacquire the target.

Specifically, if the target remains undetected for $n_{max}$ consecutive frames, the ground station computes the retrieval command based on the latest error values ($e_x$, $e_y$). If $e_x$ or $e_y$ is positive, it implies the target has exited the FOV through the right or upper boundaries, respectively. In this case, the quadcopter is instructed to rotate to the right or ascend. Conversely, negative error values indicate that the target has exited through the left or bottom boundaries, prompting the quadcopter to rotate to the left or descend. Altitude adjustments are made cautiously, ensuring that the quadcopter does not descend below the minimum safe altitude or ascend beyond permitted altitude.

If the target is still not detected after one full rotation following the initial retrieval attempt, the quadcopter terminates the operation and lands at its current location. Alternatively, it can be instructed to return to its hangar based on mission requirements.

Table 1 summarizes the logic used to determine the retrieval commands based on the latest error signal values.

Table 1: Logic of the retrieval commands

| Condition | Error Signal ($e_x$, $e_y$) | Quadcopter Action |
|---|---|---|
| Target exits FOV to the right | ($e_x > 0$) | Rotate right |
| Target exits FOV to the left | ($e_x < 0$) | Rotate left |
| Target exits FOV upwards | ($e_y > 0$) | Ascend |
| Target exits FOV downwards | ($e_y < 0$) | Descend |
| Target undetected after full rotation | N/A | Land or return to the hangar |

## Experimental Results

To assess the performance of the proposed autopilot system, real-time experiments were carried out using the DJI Tello programmable quadcopter in a controlled indoor environment. The objective was to evaluate the system's ability to autonomously search for and track a designated target, specifically a yellow toy car.

The experimental setup consisted of a ground station implemented on a personal computer equipped with an

Intel Core i7 processor, an NVIDIA RTX 2060 graphics card, and 32 GB of RAM. This system was responsible for executing the target detection algorithm based on the YOLOv5 network and computing the necessary flight control signals. Communication between the quadcopter and the ground station was established via a Wi-Fi connection, enabling real-time transmission of image frames and control signals.

The DJI Tello quadcopter, chosen for its affordability and ease of programmability, weighs approximately 80 grams and is equipped with a 5-megapixel camera capable of recording 720p high-definition video. It also features electronic image stabilization to reduce motion blur and ensure clearer image capture during flight. The drone has a control range of up to 100 meters, making it well-suited for small-scale indoor experiments in academic environments.

Two experimental scenarios were defined to evaluate the tracking capabilities of the system. The first scenario involved tracking a stationary target, while the second focused on tracking a moving target. In both cases, the quadcopter followed a consistent initialization procedure. At the start of each experiment, the drone ascended vertically to a height of one meter and initiated the search phase by rotating about its yaw axis at a rate of approximately 36 degrees per second. Image frames captured by the onboard camera were continuously transmitted to the ground station for processing. Upon successful detection of the target, the system transitioned from the search phase to the tracking phase, wherein the autopilot controller adjusted the FCs to maintain the target within the FOV.

All experiments were conducted in an indoor setting with a uniform white background to minimize visual noise and ensure consistent detection performance. This controlled environment allowed for a clear assessment of the tracking behavior and robustness of the proposed autopilot system.

The specific parameters used for the autopilot system during these experiments are summarized in Table 2.

Table 2: Autopilot system's parameters

| **Parameter** | $T_{max}$ | $n_{max}$ | $K_p$ | $K_d$ | Sampling Time (sec) |
|---|---|---|---|---|---|
| **Value** | 1 | 3 | 0.1 | 0.01 | 0.1 |

*Scenario One*

In the first scenario, the performance of the proposed system was evaluated by tracking a stationary object at three different distances between the quadcopter and the target: one meter, two meters, and three meters. The target, a yellow toy car, was placed on a table with a height of 100 cm. The objective was to assess the system's ability to maintain accurate tracking and positioning of a fixed object within the quadcopter's FOV under varying spatial conditions.

Fig. 8 illustrates the quadcopter's view during the flight test of fixed target at distance of one, two and three meters. In this figure the red dot represents the center of the FOV. The results indicate that the system consistently achieved its primary objective of precisely detecting, locating and consistently keeping the object within the quadcopter's FOV across all tested distances.

The experimental observations confirmed that the autopilot system could guide the quadcopter with sufficient precision to compensate for minor deviations and ensure continuous target visibility. Specifically, in the two-meter trial, the quadcopter executed corrective movements in the upward and leftward directions to maintain the target's alignment with the center of the image frame. The sequential images in Fig. 8 further validate the accuracy and reliability of the proposed method, demonstrating its effectiveness in scenarios involving static target tracking.


Fig. 8: A view of the quadcopter in the flight test of fixed target tracking at distance of two meters.

To further evaluate the system's performance, Fig. 9 presents the tracking error signals and corresponding control commands for each fixed-distance tracking case. The data illustrates that the proposed method effectively minimizes the tracking errors while maintaining control signal values within the expected operational range of ±100.

Across all trials, the quadcopter consistently maneuvered to align the target with the center of the FOV, demonstrating the robustness of the control strategy. An important observation is the smoothness of the quadcopter's flight path. The autopilot system generated gradual, stable control commands that avoided sudden or erratic actions. This resulted in a controlled and progressive movement toward the target, which is particularly advantageous in scenarios requiring precise and safe positioning. These results underscore the reliability and stability of the proposed system in maintaining accurate tracking of stationary targets.

*Scenario Two*

To further evaluate the performance of the proposed autopilot system in dynamic conditions, an experiment was designed in which the target moved horizontally from right to left at an approximate speed of 6 cm/s on a table. The quadcopter was initially positioned three meters from the target.

Fig. 10 illustrates the quadcopter's FOV during this scenario. demonstrating the system's capability to achieve control objectives and consistently keep the target within the quadcopter's FOV. Approximately three seconds after initiating the tracking phase, the red dot is observed to lie within the bounding box of the detected object. This positioning confirms the system's ability to adjust the quadcopter's orientation effectively, ensuring that the moving target remains centered within the FOV.

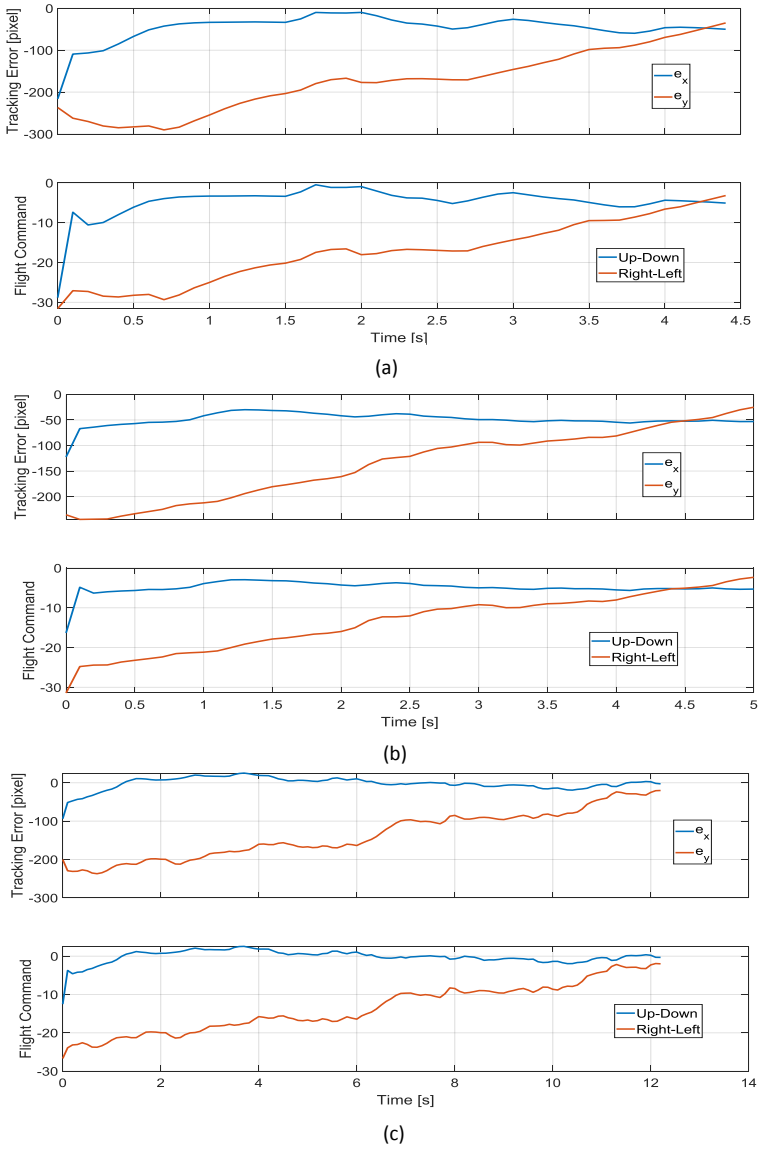


Fig .9: Tracking errors and control signals for a fixed target tracking at distances of: (a) one meter, (b) two meters, and (c) three meters.


Fig. 10: A view of the quadcopter in the flight test of moving target tracking.

The quantitative results, including tracking error and control signal data, are presented in Fig. 11. These results further demonstrate the proposed system's capacity to generate appropriate control commands, resulting in stable and responsive flight behavior.

While complete elimination of the tracking error over prolonged periods is inherently challenging (due to both the nature of the Tello drone's limited flight dynamics and the continuous motion of the target) the tracking errors remain consistently bounded around zero.

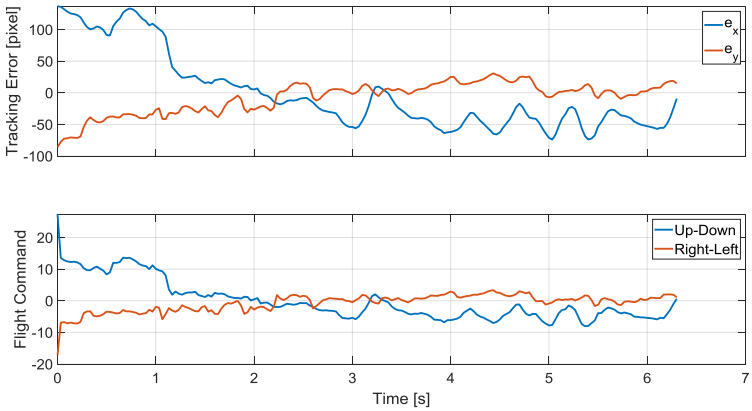


Fig. 11: Tracking errors and control signals for a moving target tracking.

This behavior indicates that the target is maintained near the center of the FOV, validating the accuracy and responsiveness of the autopilot in handling moving targets.

The normalized root mean square Error (NRMSE) of the tracking is reported in Table 3 for both scenarios. The NRMSE metric is calculated using (26) and (27) for X and Y axes, where n denotes the number of measurement data.

$$NRMSE_x = \left( \frac{1}{1280} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i)^2} \right) \quad (26)$$

$$NRMSE_y = \left( \frac{1}{720} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i)^2} \right) \quad (27)$$

The results indicate that the tracking error for stationary targets decreases as the target exists farther from the quadcopter. Additionally, the tracking error for moving targets is greater than for stationary targets, which is an obvious outcome.

Table 3: The normalized root mean square of the tracking errors for two experimental scenarios

| scenarios | Distance | $NRMSE_x$ | $NRMSE_y$ |
|---|---|---|---|
| **One** | 1 m | 0.0452 | 0.2629 |
| | 2 m | 0.0398 | 0.2104 |
| | 3 m | 0.0132 | 0.2070 |
| **Two** | --- | 0.0471 | 0.0356 |

## Discussion

The proposed framework demonstrated robust target tracking in controlled indoor environments through a simple yet effective combination of a PD controller and Kalman filter, supported by empirically tuned retrieval parameters. While computationally efficient, the PD controller is limited in managing nonlinearities, time-varying dynamics, and external disturbances. Additionally, the retrieval strategy, which relies on the last known error, assumes moderately smooth target motion, reducing effectiveness against highly erratic trajectories.

Our integrated system, combining the YOLO network, Kalman filter, and PD controllers, offers distinct advantages over these approaches. Our system provides holistic integration of perception, control, and retrieval strategies for a complete target tracking mission, including the crucial phase of object recovery. Unlike Alshaer's work [32], SMART-TRACK [46] and VTD3 [47], which primarily focus on detection and tracking, our framework extends to active control and a dedicated object retrieval strategy. Furthermore, while other works also address drone target tracking, our use of a simpler PD controller offers advantages in terms of computational efficiency and ease of implementation, particularly for resource-constrained UAV platforms, while still achieving robust tracking performance. The unique inclusion of an object retrieval strategy in our system provides a critical capability for re-acquiring lost targets, a feature often absents in many conventional tracking frameworks.

A critical practical consideration is the communication latency between the ground station, where image processing and control computations occur, and the quadrotor. This latency introduces a delay between target detection and control command execution, potentially degrading tracking performance, especially for targets moving faster than those tested in our experiments. In such cases, the quadrotor may react to outdated position information, increasing tracking errors and destabilizing the control loop. While the current setup manages latency adequately for moderately paced targets, future work could address this limitation by deploying larger quadcopters equipped with onboard processing capabilities to enable fully autonomous operation and facilitate reliable outdoor deployment, thereby extending the applicability of the proposed framework beyond controlled indoor environments.

Future improvements could also include advanced control strategies such as adaptive control or model predictive control for proactive disturbance rejection. Retrieval enhancements may involve stochastic motion models combined with dynamically adjusting detection thresholds and feature scales to maintain accuracy under variable lighting and target distances, complemented by adaptive Kalman filtering to ensure robust performance in challenging outdoor conditions.

## Conclusion

This paper presented the development and evaluation of an autonomous quadcopter-based system for tracking both stationary and moving targets with unknown dynamics. The proposed system integrates three main components: target detection, search, and tracking. Target detection was implemented using the YOLOv5 neural network combined with the Kalman filter to implement robustness against detection failures, including false positives and false negatives. Additionally, an object retrieval mechanism was incorporated to enable re-identification of targets that momentarily exit the field of view. For flight control, two PD controllers were employed to generate the flight control signal for the quadcopter.

The system's performance was validated through a series of real-time experiments involving fixed and moving target tracking scenarios. The results demonstrated that the system successfully maintained the target within the quadcopter's field of view across varying conditions. Notably, in fewer than 10% of frames captured during tracking, the target was not detected; however, the Kalman filter provided effective estimations to compensate for these instances. Overall, tracking performance remained stable, and the system achieved its objectives with minimal degradation, even in the absence of target identification.

The integrated target tracking with a quadcopter framework, combining the YOLO network, Kalman filter, and PD controllers, offers significant advantages over conventional alternatives. Unlike two-stage detectors such as Faster R-CNN, the YOLO-based design is optimized for real-time operation in dynamic missions while maintaining high detection accuracy. The Kalman filter provides robust state estimation by fusing detection outputs over time, and the PD controllers, in synergy with advanced vision modules, ensure responsive and precise target following. A dedicated object retrieval mechanism further enables recovery of lost targets; a feature often absent in similar systems.

As such, future work will aim to enhance the system's robustness in unstructured environments. Potential improvements include integrating stereo vision to estimate the depth (longitudinal distance) between the quadcopter and the target, thereby enhancing control accuracy. Additionally, the use of a larger quadcopter equipped with onboard processing capabilities could enable full autonomy and facilitate deployment in outdoor environments, thereby extending the applicability of the proposed method.

## Author Contributions

Dr. Nekoukar outlined the overall roadmap, while Asadi conducted the literature search on key research in this field. The proposed method was jointly developed by both authors and implemented and tested by Asadi. The initial draft of the manuscript was prepared by Asadi and subsequently revised by Dr. Nekoukar.

## Acknowledgment

The authors would like to thank the editor and anonymous reviewers.

## Conflict of Interest

The authors declare that they have no potential conflicts of interest related to the publication of this

work. Furthermore, all ethical considerations—including plagiarism, informed consent, research misconduct, data fabrication or falsification, duplicate publication or submission, and redundancy—have been fully observed by the authors.

## Abbreviations

| | |
|---|---|
| *AI* | Artificial Intelligence |
| *CV* | Computer Vision |
| *DL* | Deep Learning |
| *FC* | Flight Control |
| *FOV* | Field of View |
| *R-CNN* | Region Convolutional Neural Networks |
| *PD* | Proportional-Derivative |
| *SSD* | Single Shot Multi-Box Detection |
| *UAV* | Unmanned Aerial Vehicles |
| *YOLO* | You Only Look Once |

## References


[1] H. Y. Lin, X. Z. Peng, "Autonomous quadrotor navigation with vision based obstacle avoidance and path planning," IEEE Access, 9: 102450-102459, 2021.

[2] R. P. Padhy, "Monocular Vision Aided Autonomous UAV Navigation in Indoor Environments," 2020.

[3] S. I. Khan et al., "UAVs path planning architecture for effective medical emergency response in future networks," Phys. Commun., 47: 101337, 2021.

[4] N. Batsoyol, Y. Jin, H. Lee, "Constructing full-coverage 3D UAV ad-hoc networks through collaborative exploration in unknown urban environments," in Proc. 2018 IEEE International Conference on Communications (ICC): 1-7, 2018.

[5] C. Wang, P. Liu, T. Zhang, J. Sun, "The adaptive vortex search algorithm of optimal path planning for forest fire rescue UAV," in Proc. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC): 400-403, 2018.

[6] G. Bevacqua, J. Cacace, A. Finzi, V. Lippiello, "Mixed-initiative planning and execution for multiple drones in search and rescue missions," in Proc the International Conference on Automated Planning and Scheduling, 25: 315-323, 2015.

[7] N. Dilshad, J. Hwang, J. Song, N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in Proc. 2020 International Conference on Information and Communication Technology Convergence (ICTC): 728-732, 2020.

[8] T. Tomic et al., "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," IEEE Rob. Autom. Mag., 19(3): 46-56, 2012.

[9] Z. Kalal, K. Mikolajczyk, J. Matas, "Tracking-learning-detection," IEEE Trans. Pattern Anal. Mach. Intell., 34(7): 1409-1422, 2011.

[10] S. Wu, R. Li, Y. Shi, Q. Liu, "Vision-based target detection and tracking system for a quadcopter," IEEE Access, 9: 62043-62054, 2021.

[11] Y. Qu, L. Jiang, X. Guo, "Moving vehicle detection with convolutional networks in UAV videos," in Proc. 2016 2nd International Conference on Control, Automation and Robotics (ICCAR): 225-229, 2016.

[12] P. C. Chen, Y. C. Chiang, P. Y. Weng, "Imaging using unmanned aerial vehicles for agriculture land use classification," Agriculture, 10(9): 416, 2020.

[13] S. Kapania, D. Saini, S. Goyal, N. Thakur, R. Jain, P. Nagrath, "Multi object tracking with UAVs using deep SORT and YOLOv3 RetinaNet detection framework," in Proc. the 1st ACM Workshop on Autonomous and Intelligent Mobile Systems: 1-6, 2020.

[14] Y. Liu, Z. Meng, Y. Zou, M. Cao, "Visual object tracking and servoing control of a nano-scale quadrotor: System, Algorithms, And Experiments," IEEE CAA J. Autom. Sin., 8(2): 344-360, 2021.

[15] X. Wu, D. Sahoo, S. C. Hoi, "Recent advances in deep learning for object detection," Neurocomputing, 396: 39-64, 2020.

[16] S. Kuutti, R. Bowden, Y. Jin, P. Barber, S. Fallah, "A survey of deep learning applications to autonomous vehicle control," IEEE Trans. Intell. Transp. Syst., 22(2): 712-733, 2020.

[17] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," J. Sensors, 2017.

[18] T. Diwan, G. Anirudh, J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," Multimedia Tools Appl., 82(6): 9243-9275, 2023.

[19] V. K. Sharma, R. N. Mir, "A comprehensive and systematic look up into deep learning based object detection techniques: A review," Comput. Sci. Rev., 38: 100301, 2020.

[20] Q. Song et al., "Object detection method for grasping robot based on improved YOLOv5," Micromachines, 12(11): 1273, 2021.

[21] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, S. Tang, "Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges," Materials, 13(24): 5755, 2020.

[22] Q. Qiu, D. Lau, "Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images," Autom. Constr., 147: 104745, 2023.

[23] W. Liu et al., "Ssd: Single shot multibox detector," in Proc. Computer Vision–ECCV 2016: 14th European Conference: 21-37, 2016.

[24] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection," in Proc.the IEEE Conference on Computer Vision and Pattern Recognition: 779-788, 2016.

[25] J. Redmon, A. Farhadi, "YOLO9000: better, faster, stronger," in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 7263-7271, 2017.

[26] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[27] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 580-587, 2014.

[28] R. Girshick, "Fast r-cnn," in Proc. the IEEE International Conference on Computer Vision: 1440-1448, 2015.

[29] S. Ren, K. He, R. Girshick, J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," arXiv:1506.01497v3, 2016.

[30] S. V. H. Pham, K. V. T. Nguyen, "Productivity assessment of the yolo v5 model in detecting road surface damages," Appl. Sci., 13(22): 12445, 2023.

[31] A. Barisic, M. Car, S. Bogdan, "Vision-based system for a real-time detection and following of UAV," in PROC. 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS): 156-159, 2019.

[32] N. Alshaer, R. Abdelfatah, T. Ismail, H. Mahmoud, "Vision-Based UAV detection and tracking using deep learning and kalman filter," Comput. Intell., 41(1): e70026, 2025.

[33] X. Zhao, F. Pu, Z. Wang, H. Chen, Z. Xu, "Detection, tracking, and geolocation of moving vehicle from uav using monocular camera," IEEE Access, 7: 101160-101170, 2019.

[34] D. A. Mercado-Ravell, P. Castillo, R. Lozano, "Visual detection and tracking with UAVs, following a mobile object," Adv. Rob., 33(7-8): 388-402, 2019.

[35] M. A. Mokhtari, M. Taheri, "Real-time object detection and tracking using YOLOv3 network by quadcopter," Mech. Based Des. Struct. Mach: 1-19, 2022.

[36] H. E. Dursun, E. C. Güven, B. Avci, T. Kumbasar, "Recognizing and tracking person of interest: A real-time efficient deep learning based method for quadcopters," in Proc. 2023 10th International Conference on Recent Advances in Air and Space Technologies (RAST): 1-7, 2023.

[37] K. P. Gama, C. Hu, W. Hao, "Visual tracking based deep learning and control design onboard small-sized quadrotor UAV," in Proc. 2022 41st Chinese Control Conference (CCC): 5865-5870, 2022.

[38] V. Nekoukar, N. M. Dehkordi, "Robust path tracking of a quadrotor using adaptive fuzzy terminal sliding mode control," Control Eng. Pract., 110: 104763, 2021.

[39] J. Ajmera, V. Sankaranarayanan, "Point-to-point control of a quadrotor: Theory and experiment," IFAC-PapersOnLine, 49(1): 401-406, 2016.

[40] A. Benjumea, I. Teeti, F. Cuzzolin, A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," arXiv preprint arXiv:2112.11798, 2021.

[41] T. Ahmad, M. Cavazza, Y. Matsuo, H. Prendinger, "Detecting human actions in drone images using YOLOv5 and stochastic gradient boosting," Sensors, 22(18): 7020, 2022.

[42] N. K. Pham SVH, "Productivity assessment of the Yolo V5 Model in detecting road surface damages.," Appl. Sci., 13(22), 2023.

[43] U. Nepal, H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," Sensors, 22(2): 464, 2022.

[44] R. Xu, H. Lin, K. Lu, L. Cao, Y. Liu, "A forest fire detection system based on ensemble learning," Forests, 12(2): 217, 2021.

[45] V. S. Nguyen, J. Jung, S. Jung, S. Joe, B. Kim, "Deployable hook retrieval system for UAV rescue and delivery," IEEE Access, 9: 74632-74645, 2021.

[46] K. Gabr, M. Abdelkader, I. Jarraya, A. AlMusalami, A. Koubaa, "Smart-track: A novel kalman filter-guided sensor fusion for robust uav object tracking in dynamic environments," IEEE Sensors J., 25(2): 3086-3097, 2024.

[47] X. Zhao, X. Huang, J. Cheng, Z. Xia, Z. Tu, "A vision-based end-to-end reinforcement learning framework for drone target tracking," Drones, 8(11): 628, 2024.

## Biographies

**Zahra Hassani** received her B.Sc. in Electrical Engineering from Dr. Shariati University in 2022 and her M.Sc. in Electrical Engineering from Shahid Rajaee Teacher Training University in 2024. Her research interests include UAV control, sensor fusion, autonomous systems, AI-based object tracking**,** decision making algorithms, and safety and reliability in autonomous driving.

• Email: zahrahasani@sru.ac.ir
• ORCID: 0009-0009-8737-7823
• Web of Science Researcher ID: NA
• Scopus Author ID: NA
• Homepage: NA

**Vahab Nekoukar** received the B.Sc. degree in Electrical Engineering from the Khaje Nasir Toosi University of Technology, Tehran, Iran, in 2005, the M.Sc. degree in Electrical Engineering from Tarbiat Modares University, Tehran, in 2007, and the Ph.D. degree in Electrical Engineering from Iran University of Science and Technology, Tehran, in 2012. In 2014, he joined the School of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran, as an Assistant Professor. His current research interests include control of biological systems, robotics, navigation, and guidance systems.

• Email: v.nekoukar@sru.ac.ir
• ORCID: 0000-0001-5196-1264
• Web of Science Researcher ID: NA
• Scopus Author ID: 26030036600
• Homepage: www.sru.ac.ir/nekoukar