https://doi.org/10.7840/kics.2025.50.5.818

Autonomous Precision Landing for UAVs Using Sequential Object Detection

Seong Won Yoo◆, Soo Young Shin°

ABSTRACT

This paper proposes an autonomous precision landing system for quadrotor UAVs(Unmanned Aerial Vehicles) using sequential object detection. The system utilizes a single one-axis tilt gimbal camera to perform forward view monitoring during flight and precision control during landing, effectively reducing the computational load of the UAV system. The YOLO(You Only Look Once) algorithm is employed for object detection, where the landing pad is detected at higher altitudes, and an auxiliary marker is additionally detected at lower altitudes to enable more precise positional control. Experimental results demonstrate that the proposed sequential detection approach significantly improves landing accuracy compared to conventional methods that rely solely on landing pad detection. Furthermore, the system maintains consistent precision across various altitudes, effectively mitigating bounding box variation issues at close distances. Verification through simulation and real-world experiments confirms the reliability, accuracy, and practicality of the proposed system, even in resource-constrained UAV environments.

Key Words : UAV(Unmanned Aerial Vehicle), Autonomous Precision Landing, Sequential Object Detection, YOLO(You Only Look Once) algorithm

Ⅰ. Introduction

With the increasing utilization of Unmanned Aerial Vehicles (UAVs), research in various fields such as logistics, disaster relief, and surveillance has been actively progressing[1]. Among UAV technologies, autonomous flight capability has garnered significant attention, enabling UAVs to perform missions independently without human intervention. This feature not only reduces operational costs but also enhances efficiency in environments that are difficult or hazardous for human access.

An essential component of autonomous UAV systems is precision autonomous landing technology, which ensures that UAVs can safely land in confined spaces or specific target locations. Precision landing is particularly critical in applications such as cargo delivery, emergency rescue, and urban operations. However, achieving high precision in autonomous landing involves several challenges. Typically, accurate positioning and control require the use of various sensors, which adds complexity to the system and increases the UAV's payload.

Target detection-based landing, which performs autonomous landing using only a single camera without additional sensors or equipment, offers significant ad-

※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2024-00437190, 50%) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2024-RS-2022-00156394, 50%) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation)

◆ First Author : Kumoh National Institute of Technology, yoo124512@gmail.com, 학생회원

° Corresponding Author : Kumoh National Institute of Technology, wdragon@kumoh.ac.kr, 종신회원

논문번호 : 202411-291-D-RU, Received November 18, 2024; Revised December 19, 2024; Accepted December 24, 2024

vantages in terms of efficiently reducing payload and overcoming computational limitations of companion PCs in UAV systems[2,3]. One representative approach to target detection-based landing is the AR(Augmented Reality) marker-based system, which demonstrated high landing precision. However, this approach faced challenges as the distance between the UAV and the landing pad increased, making marker detection difficult[4]. To address this limitation, research has also explored deep learning-based autonomous landing systems, which rely on object detection[5]. Nevertheless, these methods suffer from the issue of decreased alignment accuracy due to the proportional increase in the bounding box size at closer distances.

To address these issues, this study proposes a precision autonomous landing system based on sequential object detection, which overcomes the limitations of existing deep learning-based landing approaches and provides higher landing precision. The system utilizes the YOLO real-time object detection algorithm to detect the landing pad at high altitudes and align the UAV's camera with the center of the pad, ensuring a stable descent. As the altitude decreases, the system detects an auxiliary marker attached to the landing pad to achieve more precise positional control. This sequential object detection approach ensures that the UAV lands precisely at the target location.

YOLO (You Only Look Once) is a deep learning algorithm designed to simultaneously detect and classify objects in images or videos with a single processing pass[6]. In this study, YOLO was chosen for real-time object detection due to its effective balance between speed and accuracy. In UAV landing, where real-time performance is crucial, YOLO's unique architecture provides fast processing speeds while maintaining high detection accuracy. Compared to other deep learning algorithms such as Faster R-CNN and SSD, YOLO offers significantly faster inference times while preserving accuracy, making it a more suitable choice for UAV environments where computational resources are limited, and real-time processing is required[7]. This enables the UAV to quickly detect and precisely align with the landing pad, ensuring the critical factor of landing accuracy is maintained for safe operations.

Additionally, this paper adopts the YOLOv5s[8] model, selected for its superior real-time object detection performance and efficiency on embedded systems, such as the lightweight companion PC mounted on the UAV. Compared to other versions, YOLOv5s demonstrates a better balance of detection accuracy and computational efficiency, making it highly suitable for resource-constrained environments[9,10]. High recall is critical in UAV landing control to ensure the landing pad is consistently detected, reducing the risk of landing failures and enhancing operational safety.

The structure of this paper is organized as follows: Section 2 describes the landing scenario and system model, the application of YOLO, and the proposed landing system algorithm. Section 3 presents the experimental methods, results, and a comparative analysis with previous research. Finally, Section 4 concludes with the findings from the implementation and experiments.

Ⅱ. Autonomous Precision Landing for UAVs using Sequential Object Detection

2.1 Landing Scenario

Fig. 1 illustrates the flowchart of the autonomous precision landing system proposed in this paper. Initially, the UAV utilizes GPS(Global Positioning System) to receive the coordinates of the landing loca-

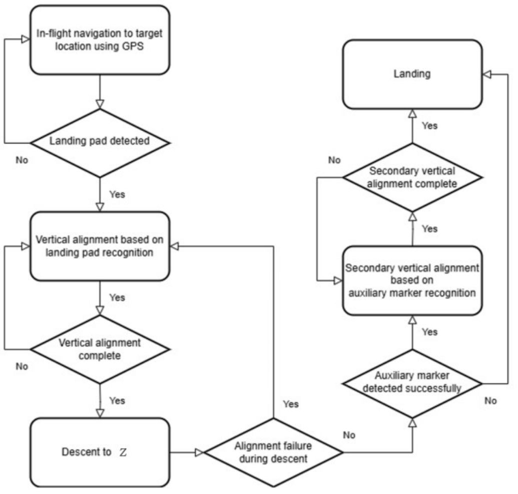

In-flight navigation to target location using GPS
Landing pad detected
No
Yes
Vertical alignment based on landing pad recognition
Yes
Vertical alignment complete
No
Yes
Descent to Z
Alignment failure during descent
Yes
No
Auxiliary marker detected successfully
No
Yes
Secondary vertical alignment based on auxiliary marker recognition
Yes
Secondary vertical alignment complete
No
Yes
Landing

Fig. 1. flowchart of the System

tion and move to the target area[11]. Subsequently, using a 1-axis tilt gimbal camera oriented downward, the UAV performs real-time object detection to detect the pre-trained landing pad. Once the landing pad is detected, the UAV adjusts its position by moving forward, backward, left, or right to align vertically with the detected object. When the system determines that the vertical alignment is complete, the UAV descends to an altitude below 2 meters and hovers at that position. Subsequently, the UAV performs a more precise landing based on the detection of the auxiliary marker.

2.2 System Model

The system model proposed in this paper consists mainly of the UAV, the GCS (Ground Control System), and a VPN(Virtual Private Network) server for data communication[12,13] between the UAV and the GCS, as shown in Fig. 2. Through this architecture, the GCS can monitor the UAV's flight, issue mission commands, and observe the video feed from the camera mounted on the UAV in real time.

In this paper, The UAV is a multirotor platform, and its key components are as follows. The UAV is built on a Holybro X500 quadrotor frame and is equipped with a GPS and Pixhawk 6C Flight Control Unit (FCU) for providing approximate location information during flight. To facilitate forward vision and real-time object detection for landing, a SIYI A2 mini 1-axis tilt gimbal camera is mounted on the UAV. Additionally, a 1D LiDAR distance sensor is used for more accurate altitude measurements. An Nvidia Jetson Orin NX companion computer is used to receive commands from the GCS and to process computations needed to identify the landing pad based on the camera footage.

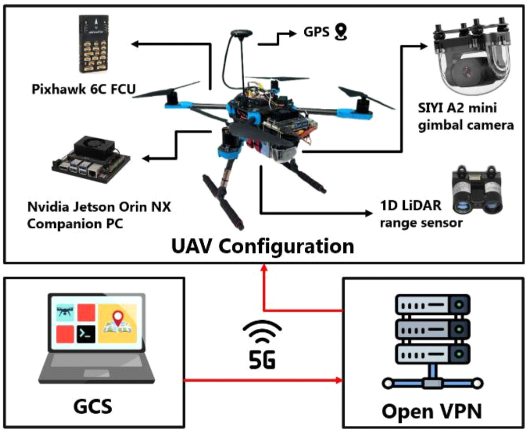


Fig. 2. System Model

2.3 Application of YOLO(You Only Look Once)

Fig. 3 shows the design of the landing pad used in this paper. To collect image data, the landing pad with an H marker was captured using a UAV from various altitudes and positions, resulting in 120 images. To augment the dataset, all images were rotated, expanding the dataset to a total of 1,080 images. Separate labeling was applied to create distinct datasets for the landing pad model and the H auxiliary marker model. Both datasets were trained using the YOLOv5s model, generating two independent weight files. These are used for landing control at high and low altitudes, respectively. Fig. 4 shows the change in object detection classes based on altitude.

The decision not to train both classes on a single weight file is due to a spatial constraint inherent in YOLO. Two classes overlap, each grid cell can only

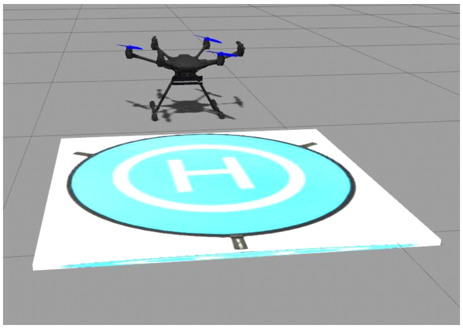


Fig. 3. Design of the Landing pad

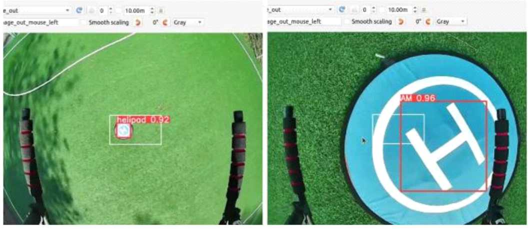
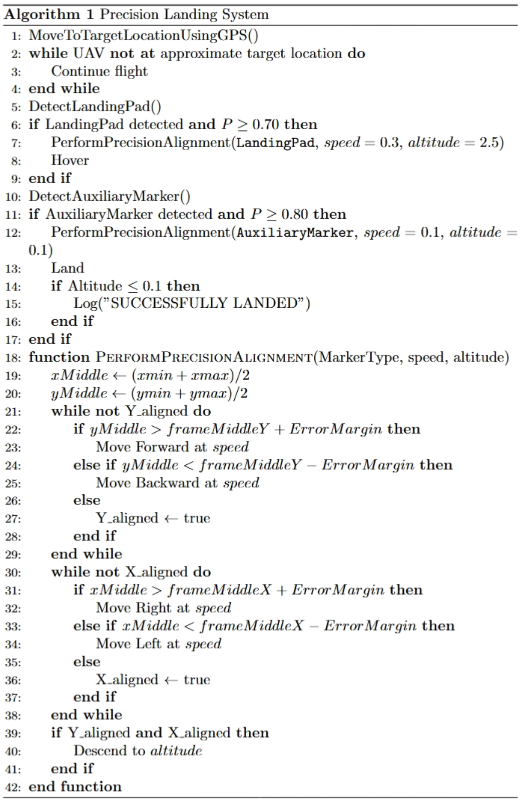


Fig. 4. Change in Object Detection Classes Based on Altitude

represent one class, increasing the likelihood that only one of the two overlapping objects will be detected. This can negatively affect landing stability. While there are multiclass classification techniques to address this issue, the computational performance limitations of the companion PC mounted on the UAV were considered. Thus, instead of employing more complex computations, the system was implemented with a focus on lightweight design, adopting a sequential object detection approach based on altitude.

2.4 Autonomous Precision Landing

This section provides a detailed explanation of the algorithm for the proposed UAV precision landing system using the YOLOv5s model to detect objects in real time and to perform precision landing based on the detected bounding box information. Fig. 5 illustrates the pseudocode describing the operation of the entire system, outlining the steps necessary for the UAV to reach the target landing point safely and accurately. The algorithm consists of two main stages, with precise control at each stage to ensure successful landing at the target point. The landing algorithm was first thoroughly tested in the Gazebo simulation environment[14].



```
Algorithm 1 Precision Landing System
MoveToTargetLocationUsingGPS()
while UAV not at approximate target location do
    Continue flight
end while
DetectLandingPad()
if LandingPad detected and $P \geq 0.70$ then
    PerformPrecisionAlignment(LandingPad, $speed = 0.3$, $altitude = 2.5$)
    Hover
end if
DetectAuxiliaryMarker()
if AuxiliaryMarker detected and $P \geq 0.80$ then
    PerformPrecisionAlignment(AuxiliaryMarker, $speed = 0.1$, $altitude = 0.1$)
    Land
    if Altitude $\leq 0.1$ then
        Log("SUCCESSFULLY LANDED")
    end if
end if
function PERFORMPRECISIONALIGNMENT(MarkerType, speed, altitude)
    $xMiddle \leftarrow (xmin + xmax)/2$
    $yMiddle \leftarrow (ymin + ymax)/2$
    while not Y_aligned do
        if $yMiddle > frameMiddleY + ErrorMargin$ then
            Move Forward at $speed$
        else if $yMiddle < frameMiddleY - ErrorMargin$ then
            Move Backward at $speed$
        else
            Y_aligned ← true
        end if
    end while
    while not X_aligned do
        if $xMiddle > frameMiddleX + ErrorMargin$ then
            Move Right at $speed$
        else if $xMiddle < frameMiddleX - ErrorMargin$ then
            Move Left at $speed$
        else
            X_aligned ← true
        end if
    end while
    if Y_aligned and X_aligned then
        Descend to $altitude$
    end if
end function
```

Fig. 5. Precision Landing Algorithm

Firstly, pixel coordinates within the camera frame are represented as (*x*, *y*), where the *x* value increases from left to right across the frame and the *y* value increases from top to bottom. This allows for precise specification of a location within the frame.

The size of the camera frame is defined by the width in pixels *cam_width* and height in pixels *cam_height*. The center of the frame is calculated as half of each axis.

$$frameMiddleX = cam\_width / 2 \tag{1}$$

$$frameMiddleX = cam\_width / 2 \tag{2}$$

(*frame MiddleX*, *frame MiddleY*) represent the center position of the camera frame. During landing control, the UAV's position is adjusted based on the relationship between the frame center and the detected bounding box center.

The coordinates of the detected bounding box include the minimum and maximum pixel values for both the *x* and *y* axes. Specifically, *xmin* represents the left boundary pixel value of the bounding box along the *x*-axis, while *xmax* represents the right boundary. Similarly, *ymin* and *ymax* represent the top and bottom boundary pixel values of the bounding box along the *y*-axis, respectively.

The coordinates of the bounding box center pixel (*xMiddle*, *yMiddle*) can be calculated as shown in Equations (3) and (4). This allows the position of the object within the camera frame to be determined.

$$xMiddle = (xmin + xmax) / 2 \tag{3}$$

$$yMiddle = (ymin + ymax) / 2 \tag{4}$$

For precision landing, it is necessary for the UAV and landing pad to be vertically aligned. To achieve this, the UAV moves forward, backward, left, or right until the bounding box center coordinates (*xMiddle*, *yMiddle*) align with the camera frame center coordinates (*frame MiddleX*, *frame MiddleY*).

However, perfectly aligning these two sets of coordinates is challenging due to the nature of quadrotor UAVs, necessitating an acceptable error margin. Here, *ErrorMargin* represents a variable that indicates the number of pixels for the margin of error, which varies with altitude, as shown in Table 1.

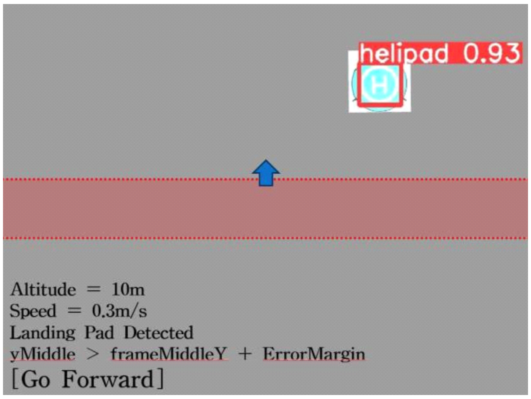
Table 1. Error Margin Based on Height for detected Classes

Detected Class	Altitude Range (z)	Error Margin from frameMiddle X,Y
Landing Pad	10m ≤ z	± 10%
Landing Pad	8 m ≤ z < 10m	± 12%
Landing Pad	6 m ≤ z < 8m	± 14%
Landing Pad	4 m ≤ z < 6m	± 16%
Landing Pad	z < 4m	± 20%
Auxiliary marker	0m ≤ z	± 5%

2.4.1 Landing Pad-Based Alignment

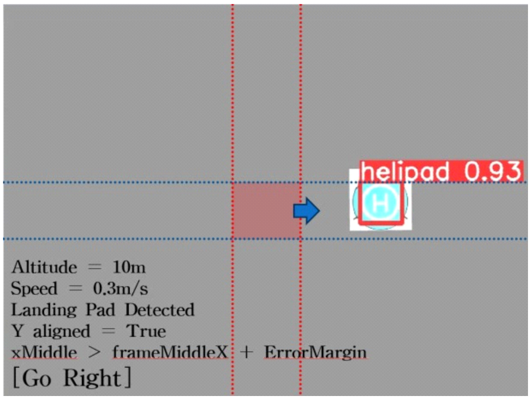
In the first stage of the algorithm, UAV control begins when the landing pad is detected, and the object detection confidence *P* is at least 70. During this process, the UAV is controlled at a speed of 0.3 m/s. The UAV repeats forward or backward movements to achieve Y-axis alignment until the center pixel of the bounding box is within the acceptable error range along the Y-axis of the camera frame. For example, as shown in Fig. 6, if the *yMiddle* value is above the y-pixel boundary defined by *frameMiddleY* plus the ErrorMargin, the UAV will move forward to align along the Y-axis within the acceptable error range. Then, the UAV continues to align along the X-axis until the bounding box center is within the X-axis central error range. Similarly, as shown in Fig. 7, if *xMiddle* is to the right of the x-pixel boundary defined by *frameMiddleX* plus the *ErrorMargin*, the UAV will move to the right to perform X-axis alignment within the error range.

Once both X and Y axis alignments are completed, the UAV descends to an altitude of approximately 2 meters while searching for the auxiliary marker. During the descent, as the distance between the UAV and the landing pad decreases, the size of the bounding box increases. Consequently, the possibility of the bounding box center deviating from the central error range also increases. To prevent this and to reduce frequent horizontal adjustments during descent, the central error range was progressively expanded as the altitude decreased, as shown in Table 1. This approach allows the UAV to descend quickly while maintaining the landing pad relatively centered in the camera frame, ensuring stability during descent until the UAV reaches a height that requires precise control for landing.



Altitude = 10m
Speed = 0.3m/s
Landing Pad Detected
yMiddle > frameMiddleY + ErrorMargin
[Go Forward]

Fig. 6. Example of Landing Pad-Based Y-axis alignment



Altitude = 10m
Speed = 0.3m/s
Landing Pad Detected
Y aligned = True
xMiddle > frameMiddleX + ErrorMargin
[Go Right]

Fig. 7. Example of Landing Pad-Based X-axis alignment

2.4.2 Auxiliary Marker-Based Alignment

The auxiliary marker is a small marker additionally attached to the landing pad and is used to achieve more precise positional control at low altitudes. When the UAV begins to detect the auxiliary marker, the existing real-time landing pad detection is terminated properly to reduce computational load and increase detection stability. After descending to an altitude of 2 meters, the UAV starts detecting the auxiliary marker, and if the detection accuracy exceeds 80%, the second alignment phase begins. Forward, backward, left, and right directional control is performed in the same manner as the first alignment; however, to align with higher precision for the auxiliary marker, the speed is reduced to 0.1 m/s. In addition, the central error margin for the X and Y axes is significantly reduced compared to the first alignment, as shown in Table 1, allowing for more precise control. Finally, once both X and Y axis alignments are complete, the UAV descends to the ground to land safely. This approach allows the UAV to detect the smaller auxiliary marker at low altitudes where the landing pad cannot be fully recognized, enabling more precise alignment and accurate landing. Fig. 8 shows this example.

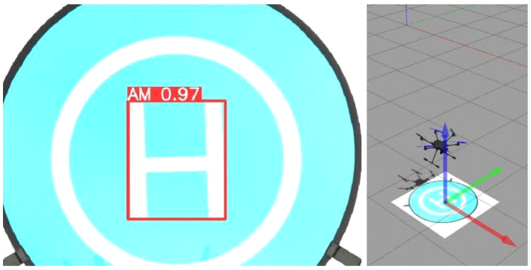


Fig. 8. Example of Auxiliary Marker Detection at Low Altitude

Ⅲ. Experiment

3.1 Experimental Method

In this paper, two experiments were conducted to evaluate the performance of the Autonomous Precision Landing using Sequential Object Detection system. The experimental environment is described in Table 2.

1) Performance Comparison with Previous Method
2) Evaluation of the Proposed Method at Various Altitudes

First, to compare the performance with previous methods, the conventional method of detecting only the landing pad and the proposed method of sequentially detecting the landing pad and auxiliary marker were tested. Both methods initiated landing from a height of 10 meters, and a total of 10 landing trials were performed. After landing, the distance between the UAV and the center of the landing pad was measured as the landing error, and the landing time was also recorded.

Table 2. Experimental Environment Specifications

Spec	Configuration
GCS Laptop	I7 - 13700H
UAV	X500 v2
Companion PC	Nvidia Jetson Orin NX
FC	Pixhawk 6C
Medel	YOLOv5 small
Camera	SIYI A2 mini
Location	KIT Soccer field
Time	15:00 KST
Wind Speed	3.8 m/s

Next, the proposed method was evaluated at various altitudes to measure landing accuracy and landing time, thereby verifying the consistency and robustness of the proposed system.

3.2 Performance Comparison with Previous Method

3.2.1 Previous Method: Landing Using Only the Landing Pad Detection

Table 3 presents the experimental results when landing was performed ten times using only the landing pad detection. In this experiment, after the UAV detected the landing pad, the time taken for the UAV to descend to the ground and the vertical distance error between the center of the landing pad and the point directly below the UAV-mounted camera were measured. As a result, the minimum error distance was 9 cm, while the maximum error distance reached

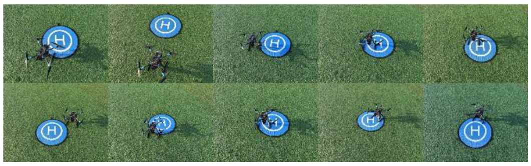


Fig. 9. Results of Landing with Only Landing Pad Detection

Table 3. Landing Performance with Only Landing Pad Detection

Test Count	Error (cm)	Time (sec)
1	22	51
2	91	77
3	43	37
4	9	108
5	23	78
6	45	131
7	21	104
8	14	42
9	25	98
10	47	131

91 cm. The landing time ranged from a minimum of 37 seconds to a maximum of 131 seconds.

3.2.2 Proposed Method: Landing with Sequential Detection of Landing Pad and Auxiliary Marker

Table 4 shows the experimental results when landing was performed using sequential detection of both the landing pad and an auxiliary marker. The measurement method was identical to that used in Table 3. The results showed that the minimum error distance was 5cm and the maximum error distance was 31cm, with relatively consistent error values. The time required for landing ranged from a minimum of 55 seconds to a maximum of 219 seconds, which indicates that additional control based on sequential object detection led to a relatively longer landing time.

Table 4. Landing Performance with Sequential Detection

Test Count	Error (cm)	Time (sec)
1	6	98
2	16	85
3	28	133
4	31	120
5	11	186
6	5	56
7	8	151
8	5	74
9	5	110
10	24	78

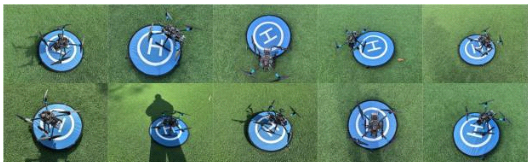


Fig. 10. Results of Landing with Sequential Detection

3.2.3 Comparative Results

Table 5 summarizes the mean error distance and mean landing time between the approach of detecting only the landing pad and the approach of sequentially detecting the landing pad and auxiliary marker for landing. The proposed method demonstrated an improvement in mean error distance, reducing it by 20.8 cm compared to the conventional method, resulting in landings closer to the center of the landing pad. Additionally, the standard deviation (STD) was reduced by 15 cm, indicating more consistent performance with the proposed method. Although the mean landing time was 23.4 seconds longer for the proposed method, it showed that more precise landings could

Table 5. Comparison of Landing Performance Metrics

Metric	Only Landing Pad Detection	Sequential Detection
Mean Error (cm)	34.8	14.0
STD (cm)	24.9	9.9
Mean Time (sec)	85.7	109.1

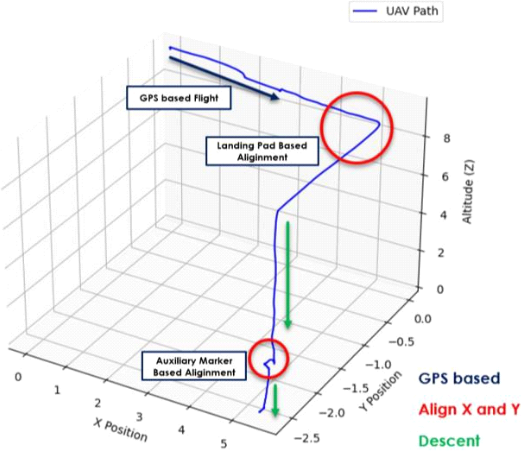


Fig. 11. UAV Landing Trajectory of Proposed Method

be achieved through additional control mechanisms. This highlights the suitability of the proposed method for drone landing applications where accuracy takes precedence over speed.

From Fig. 11 shows the landing Trajectory, it can be confirmed that the proposed method allows the UAV to follow the intended flight path during the landing process.

3.3 Evaluation of the Proposed Method at Various Altitudes

The proposed landing method performs precision control based on the auxiliary marker at a low altitude of 2 meters, regardless of the initial landing height. As a result, the landing accuracy is not influenced by the starting altitude. To verify this, the UAV conducted 10 landings each from initial altitudes of 5m, 10m, and 15m. The mean landing error and standard deviation were calculated, and the average landing time was measured.

As shown in Table 6, the landing time increases proportionally with the starting altitude. This is because additional alignment control is performed during the descent whenever the UAV deviates from its intended alignment. However, despite differences in the initial landing heights, the mean landing error and standard deviation remain consistent across all cases, demonstrating the reliability and precision of the proposed method.

Table 6. Comparison of Proposed Method Performance at Various Altitudes

Altitude	5m	10m	15m
Mean Error (cm)	15.3	14.0	13.4
STD (cm)	8.6	9.9	9.4
Mean Time (sec)	90.6	109.1	116.8

Ⅳ. Conclusion

This paper proposed a precision landing system for UAVs using sequential object detection, which detects the landing pad at high altitudes and an auxiliary marker at low altitudes to enhance landing accuracy. By adopting the YOLO object detection algorithm, the system effectively balances speed and accuracy, making it suitable for UAVs with limited computational resources.

Experimental results demonstrated the superiority of the proposed method over conventional approaches. Sequential detection significantly improved landing accuracy, particularly by addressing alignment issues caused by bounding box variations at close distances. Furthermore, tests conducted at various altitudes showed consistent landing precision, confirming the system’s reliability, despite landing time increasing with altitude.

The proposed system overcomes the limitations of previous methods while maintaining a lightweight design and ensuring stable and precise landings. Future work will focus on improving detection performance in diverse environments and optimizing control algorithms to further reduce landing time, enhancing the system’s overall practicality and efficiency.

References

[1] S. A. H. Mohsan, et al., “Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends,” *Intell. Serv. Robotics*, vol. 16, pp. 109-137, Jan. 2023.

[2] L. Xin, et al., “Vision-based autonomous landing for the UAV: A review,” *Aerospace*, vol. 9, no. 11, p. 634, 2022.

[3] M. Torabbeigi, G. J. Lim, and S. J. Kim, “Drone delivery scheduling optimization considering payload-induced battery consumption rates,” *J. Intell. & Robotic Syst.*, vol. 97, pp. 471-487, 2020.

[4] H. H. Kang and S. Y. Shin, “Precise drone landing system using aruco maker,” *J. KICS*, vol. 47, no. 1, pp. 145-150, 2022. (https://doi.org/10.7840/kics.2022.47.1.145)

[5] H. H. Kang and S. Y. Shin, “UAV automatic landing system using gimbal camera angle control and object detection,” *J. KICS*, vol. 48, no. 2, pp. 241-248, 2023. (https://doi.org/10.7840/kics.2023.48.2.241)

[6] J. Redmon, et al., “You only look once: Unified, real-time object detection,” *IEEE*

Conf. CVPR, pp. 27-30, 2016. (https://doi.org/10.1109/CVPR.2016.91)

[7] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of yolo v3, faster r-CNN, and SSD for real-time pill identification," *Research Square,* 2021. (https://doi.org/10.21203/rs.3.rs-668895/v1)

[8] G. Jocher, *YOLOv5 by Ultralytics* (Version 7.0), Computer software, 2020. (https://doi.org/10.5281/zenodo.3908559)

[9] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance comparison of YOLOv5 and YOLOv8 architectures in human detection using aerial images," *Ultima Computing: J. Sistem Komputer*, vol. 15, no. 1, pp. 8-13, 2023. (https://doi.org/10.31937/sk.v15i1.3204)

[10] F. Dang, et al., "YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems," *Computers and Electr. Agric.*, vol. 205, no. 107655, 2023.

[11] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-grade global positioning system (GPS) accuracy and reliability," *J. Forestry*, vol. 103, no. 4, pp. 169-173, 2005. (https://doi.org/10.1093/jof/103.4.169)

[12] P. Likhar, R. S. Yadav, and M. K. Rao, "Performance evaluation of transport layer VPN on IEEE 802.11 g WLAN," *Trends in Netw. and Commun.: Int. Conf., NeCOM, WeST, WiMoN*, vol. 197, pp. 407-415, Chennai, India, Jul. 2011.

[13] M. Feilner, "*OpenVPN: Building and integrating virtual private networks*," Packt Publishing Ltd., 2006. (dl.acm.org/doi/abs/10.5555/1202604)

[14] Gazebo, accessed on: Jan. 24, 2020 [Online] (http://www.gazebosim.org/)

Seong Won Yoo

Feb. 2023 : B.S. degree, Gyeongsang National University

Mar. 2023~Current : M.S. student, Kumoh National Institute of Technology

<Research Interests> Autonomous driving, Deep learing

[ORCID:0009-0008-7684-170X]

Soo Young Shin

Feb. 1999 : B.S. degree, Seoul University

Feb. 2001 : M.S. degree, Seoul University

Mar. 2010~Current : Professor Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, South Korea

<Research Interests> Wireless communications, Deep learning, Machine learning, Autonomous driving

[ORCID:0000-0002-2526-2395]