

# Camera-Based Automatic Landing of Drones Using Artificial Intelligence Image Recognition

Sedam Lee, Daeil Jo, and Yongjin (James) Kwon\*

Department of Industrial Engineering, Ajou University, Suwon, South Korea

Email: lsd2547@ajou.ac.kr, j11129@naver.com, yk73@ajou.ac.kr

**Abstract**—Today, in the era of the 4th industrial revolution, the Unmanned Aerial Vehicle (UAV) technology is attracting a high attention at home and abroad. As the use of unmanned aerial vehicles becomes more active, safety concerns are also increasing, and countermeasures are also being studied. In fact, the most common reason for UAV's accidents is landing. Therefore, a safer landing method is needed. Existing methods used for safe landing include the use of separate devices for guidance or the utilization of various communication devices within the landing area. Such methods put a great burden on the drone's payload or render them ineffective when the communication becomes unstable. Therefore, in order to enhance the safety of landing, this study proposes the AI-based image recognition approach using the onboard camera. The landing platform images are recognized through the camera mounted on the UAV without adding separate equipment. The camera identifies not only the landing points, but also three-dimensional coordinates of obstacles. This allows a safer landing, while reducing the chance of accidents or fatal damages.

**Index Terms**—landing platform tracking, obstacle avoidance, artificial intelligence, autonomous flight control, drones, UAVs

## I. INTRODUCTION

Since the Unmanned Aerial Vehicles (UAVs) have a great potential in many industrial sectors, it is highly likely to see the use of UAVs in diverse industries as the technology continues to develop. The UAVs, in particular, generally have a higher probability of accidents during the landing, especially when the drones are out of sight and the landing must be performed autonomously. This is because the preplanned, programmed landing site may not have been surveyed by the operators, hence the site may be full of unexpected obstacles or unsuitable for landing due to other reasons. Even the landing site is well prepared, such as the roof-top helipad, the strong wind or other elemental problems from the nature may cause the UAVs not to safely land at the designated points. This leads into the accidents or damages to the UAVs.

Due to these reasons, the control technology for safe landing has been actively developed in recent years [1]. For instance, a separate control aid is placed at the landing site for communications between the UAVs and the auxiliary device. In other cases, a special image

information is used to check the landing location and landing environment for safe landing [1], [2]. However, it is practically difficult to conduct the landing using a separate, auxiliary device, when the landing site is far and unreachable. When the device providing additional information regarding the landing site becomes problematic or unavailable, the landing cannot be conducted safely [3], [4]. In this regard, this study proposes the AI-based image recognition using an onboard camera equipment. Our approach uses an algorithm for safe landing by identifying not only the landing platform, but also three-dimensional coordinates of any obstacles lying within the landing areas through ground projection techniques. Unlike the other methods mentioned, this study focuses on the development of AI-based image recognition technology without using any additional equipment or other auxiliary devices that have to be preinstalled in the landing site.

## II. OVERVIEW OF SELF-LANDING DRONES

As shown in Fig. 1, the algorithm developed in this study recognizes the landing point through AI-based image recognition, and calculates the actual location (i.e., distance and direction) of the landing point. This is different than other methods that automatically guide the drones using GPS signals during the landing. During the approach, the algorithm determines the presence or absence of obstacles in and around the landing site. If there is an obstacle at the landing site, the size and position of obstacles are accurately calculated. Then, the obstacle is avoided autonomously. The brief steps are illustrated in Fig. 1.

The first equipment mounted on our drone is a companion computer that recognizes objects through image recognition. The computer locates them, and determines the flight direction onto the landing site. Unlike the Flight Control Computer (FCC), which aims to control the flight of drones, the companion computer provides a vital functionality for recognizing objects and conducting calculations through the installed AI algorithms in conjunction with the onboard camera [5]. The second equipment is an onboard camera for associated image recognition. This camera has a rotation lever that automatically controls the tilt angles. During the flight for landing, the tilt is adjusted for the shooting angle of the

---

Manuscript received November 16, 2021; revised February 2, 2022.

\*Corresponding author: Yongjin (James) Kwon.

camera in order to continuously capture the landing point [6, 7]. In addition, it is used to provide a live image in real time to the companion computer or to determine the location of an object.

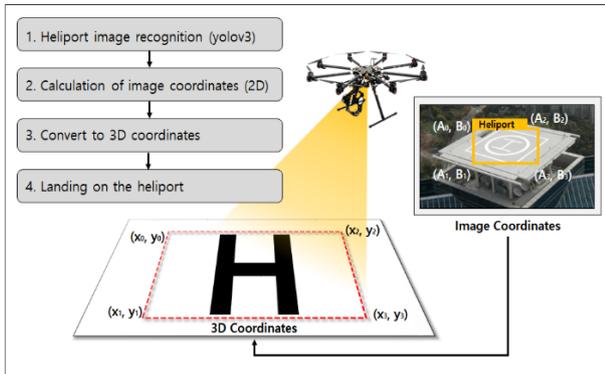


Figure 1. Schematic diagram of an automatic landing drone using image recognition.

A. Landing Point Recognition and Position Calculation

In this study, Yolo v3 is used as AI-based image recognition algorithm, as shown in Fig. 2. By collecting data about objects for recognition in advance, while defining and learning the data, the model is finally extracted. This extracted learning model is loaded on the companion computer. For this computer, the time it takes to recognize and judge an object is very short because the already trained model is used. As described above, since an immediate reaction to an object is possible, the AI algorithm can be applied for real-time purposes [8, 9]. The landing point or obstacle can be determined from the real-time image, and the image coordinates of the recognized object can be estimated through the size of the bounding box. In general, the information about the bounding box is extracted with label, center coordinates, and size (both horizontal and vertical) of the object as pixel values. Through this, the distance measurement method used in image processing and computer vision can be applied. The overall process is outlined in Fig. 2.

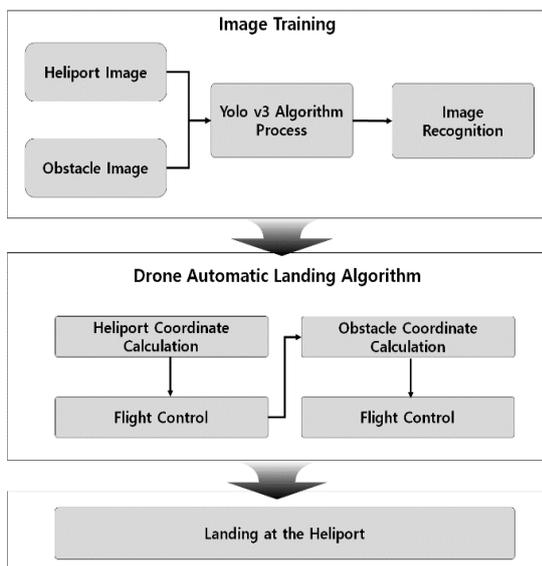


Figure 2. Schematic for automatic landing.

In detail, the implemented method is called a ground projection of image coordinates. The distance, direction, and height of the UAV as to the landing point can be calculated through the internal parameters of the camera, the image coordinates of the object recognized in the image, and the size of the bounding box [10]. Fig. 3 shows the names and coordinate values for each coordinate for ground projection. Here, the horizontal axis is set as the x-coordinate, the vertical axis being the y-coordinate, and the height being set as the z-coordinate.

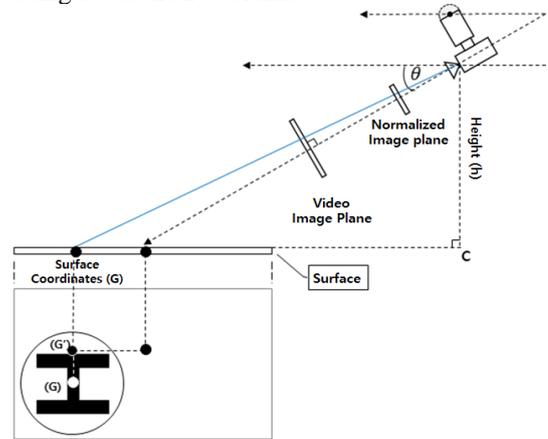


Figure 3. Actual distance measurement method using the images.

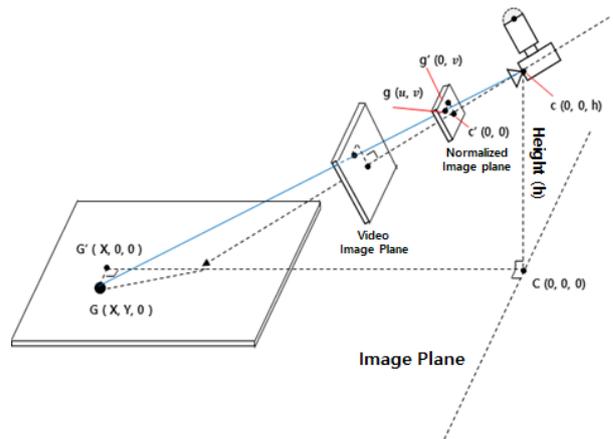


Figure 4. Ground projection (3D) of the image coordinates(2D).

To use this method, the values of camera internal parameters are defined as in Equation (1). In the formula, the height means the surface where the object recognized in the image is located.

- Camera internal parameters (1)
  - Video coordinates:  $a(x, y)$
  - Focal length:  $(fx, fy)$
  - The principal point of the video:  $(cx, cy)$
  - Camera height from the ground:  $(h)$
  - Camera Tilt:  $\theta =$  (level with the ground:  $0^\circ$  ), (below the horizontal: - ), (above horizontal : + )

As in Figs. 3 and 4, the calculation formula for measuring the distance of an object is shown. In the case of Equation (2), some of the well-known equations used in computer vision and image processing are referenced.

## - Estimation of landing location (2)

- Normal Coordinate Conversion Expression  
:  $g = (u, v)[u = \frac{(x-cx)}{fx}, v = \frac{(y-cy)}{fx}]$
- $\overline{cC} = h$
- $\overline{CG'} = h \times \tan(\frac{\pi}{2} + \theta - \text{atan}(v))$
- $\overline{cG'} = \sqrt{(\overline{cC})^2 + (\overline{CG'})^2}$
- $\overline{cg'} = \sqrt{1 + v^2}$
- $\overline{GG'} = (u \times \frac{cG'}{cg'})$
- $G(X, Y, 0) = (CG', GG', 0)$
- Flight direction the drone  $\theta_{direction}$   
:  $\text{atan}(\frac{cG'}{GG'})$

Equation (3) is necessary to determine whether there is an enough space for safe landing at the recognized landing site by calculating the area. When the vertex of the landing point on the image coordinates is obtained through (3) and the ground is projected through (2), the actual space of the landing point can be calculated. This process is shown in Fig. 5.

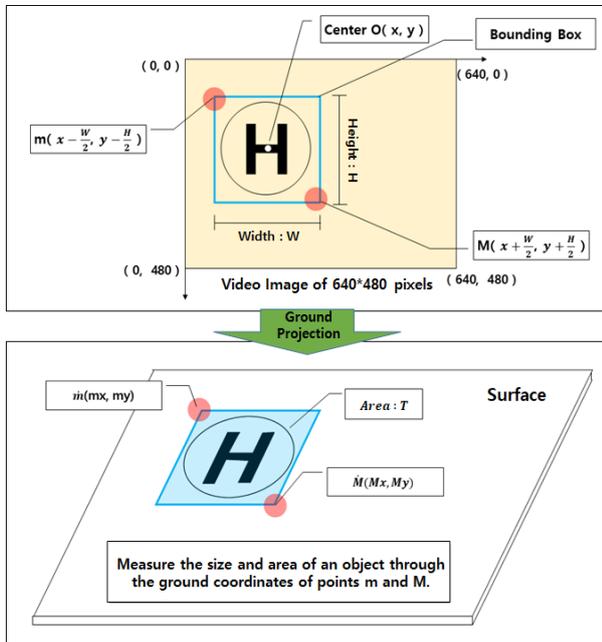


Figure 5. Ground projection for calculating the space (area) of the landing site.

## - Landing space formula (3)

- Heliport's Bounding Box:  
Width(W), Height(H), Center point 0 (x, y)
- The coordinates of the vertex of the Bounding Box closest to the image coordinates (0,0):  
 $m(x - \frac{W}{2}, y - \frac{H}{2})$
- The coordinates of the vertices of the image coordinates and the farthest bounding box (0,0):  
 $M(x + \frac{W}{2}, y + \frac{H}{2})$
- Width of landing site:  
T :  $(Mx - mx) \times (My - my)$

## B. Camera Tilt Angle Control

The UAV can estimate the information on the location of the landing point through Equations (1), (2), and (3). Then, the UAV needs to fly in the direction of landing point. The most important image for estimating the location information of landing site cannot be obtained later, if the camera is fixed during the landing. In order to prevent this and continuously recognize the landing point, the tilt angle of the camera is calculated using the flight distance of the UAV. Equation (4) and Fig. 6 show the method of adjusting the camera tilt angle using the flying distance of the UAV. The details are outlined as follows.

## - Camera tilt angle control (4)

- The drone travel distance:  $l$
- Changed camera tilt:  $\theta' = -\text{atan}(\frac{h}{G'C-l})$

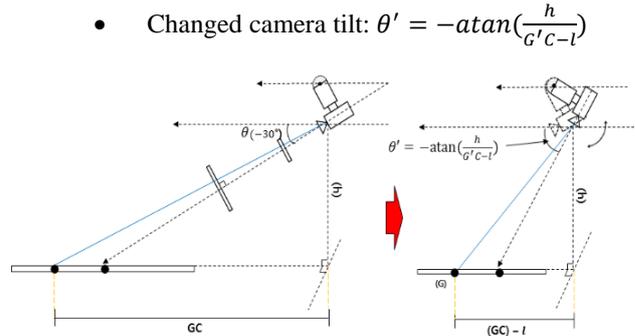


Figure 6. Adjusting the camera tilt angle according to the moving distance of the drone.

## C. Calculating the Size and Position of Obstacles

When the distance between the drone and the landing point becomes  $(GC) - L = 0$ , the tilt angle of the camera also becomes -90 degrees. After that, the drone lowers the altitude by a certain amount by the algorithm. Obstacles may be present when the drone lands on the site. For this purpose, similar to the landing site image recognition, the obstacles are labeled by the Yolo v3 algorithm to determine the presence or absence of obstacles. Unlike the landing site (i.e., heliport), which is a 2D plane with similar shapes, the shape of obstacles differs in various ways. For this, it is necessary to learn and identify objects that may belong to the category of obstacles [11, 12].

The size and location of obstacles are estimated using data such as the center coordinates of the recognized obstacles and the size (width and height) of the bounding box as shown in Figs. 7 and 8. The estimation method is outlined in Equations (5) and (6).

## - Obstacle position estimation formula (5)

- $\dot{c}\dot{C} = \dot{h}$
- $\dot{C}O' = h \times \tan(\frac{\pi}{2} - \text{atan}(v))$
- $\dot{c}O' = \sqrt{(\dot{c}\dot{C})^2 + (\dot{C}O')^2}$
- $\dot{c}o' = \sqrt{1 + v^2}$
- $OO' = \dot{v} \times \frac{\dot{c}O'}{\dot{c}o'}$
- $O(x', y', 0) = (\dot{C}O', OO', 0)$

## - Estimation of the size of obstacles (6)

- Obstacle Bounding Box:  
Width(W), Height(H), Center point 0 (x, y)

- Bounding Box vertex of obstacle closest to (0,0):

$$Om(x' - \frac{W}{2}, y' - \frac{H}{2})$$

- Bounding Box vertex of the obstacle farthest to (0,0):

$$Om(x' + \frac{W}{2}, y' + \frac{H}{2})$$

- Ground projection coordinates of points Om and OM:

$$Om(mx, my), OM(Mx, My)$$

- Width of obstacle

$$T': (Mx - mx) \times (My - my)$$

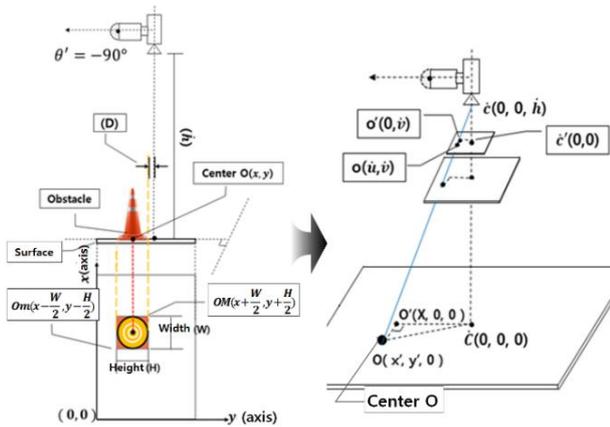


Figure 7. Method of estimating obstacle location (distance, direction) using computer vision (image processing) formulas.

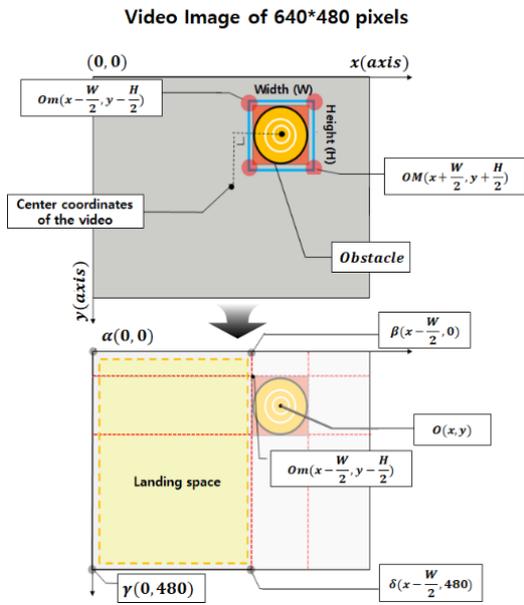


Figure 8. Calculate the area of the landing space.

The position and size of the obstacle can be calculated through Equation (6), and it is shown in Fig. 8.

As in Fig. 8, when an obstacle is located at the landing site, it is possible to predict the area where the drone can land by tapping the position of the obstacle. The landing area can be classified into 4 types as shown in Fig. 9. It is

difficult to figure out which point of area 1 or area 2 has the larger area if visually inspected.

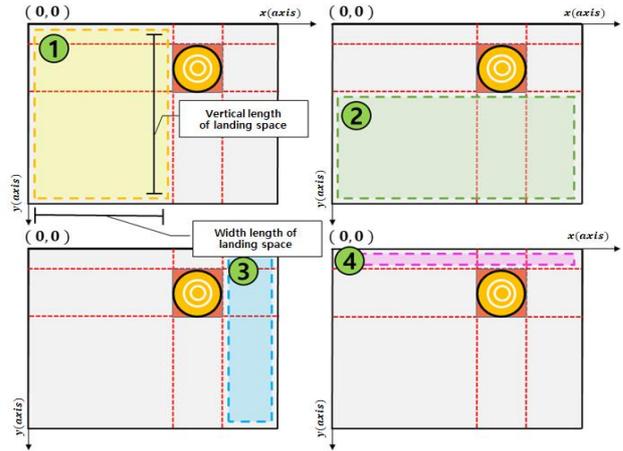


Figure 9. Estimated area of the landing space.

Additionally, in order for the drone to safely land, a safe distance must be guaranteed. The method of determining the safe landing area is shown in Table I. It can be referred to Fig. 10.

TABLE I. HOW TO SELECT A LANDING POINT AFTER AVOIDING OBSTACLES

| Choose Landing Area  |              |               |                |               |
|--|--------------|---------------|----------------|---------------|
| If the section is larger than the drone size (+safety distance), it is '(O)' and if it is smaller, it is '(X)'           |              |               |                | Landed or Not |
| Landing Area Selection Criteria  | Section Area | Section Width | Section Height |               |
|  | O            | O             | O              | Possible      |
|  | O            | O             | X              | Impossible    |
|  | O            | X             | O              | Impossible    |
|  | X            | X             | X              | Impossible    |
| If there are several sections suitable for landing, landing on the section when the following conditions are all "true". |              |               |                |               |
| 1. 'Area' is the biggest section   |              |               |                |               |
| 2. If both "Width" and "Height" are larger than "drone size (+safety distance)"  |              |               |                |               |
| 3. If 'Width' or 'Height' is the biggest section   |              |               |                |               |

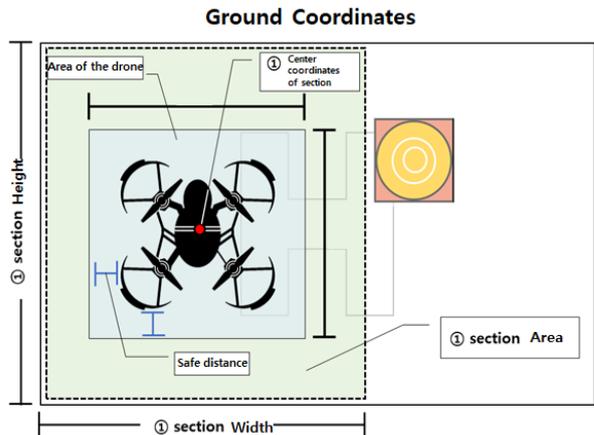


Figure 10. Comparison of landing area width and safety distance.

The sequence of algorithm developed in this study is shown in Fig. 11. After recognizing the landing point and calculating the distance between the drone and the landing

point through a position estimation formula, the camera tilt angle is controlled according to the moving distance of the drone. When the camera tilt angle is  $-90$  degrees, the drone lowers its altitude and recognizes obstacles through the Yolo v3 algorithm.

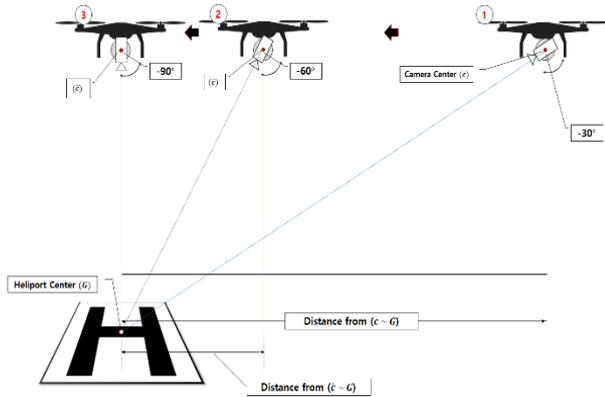


Figure 11. Flight control process of automatic landing algorithm-(1).

After lowering the altitude, the drone calculates the area of the obstacle's bounding box as shown in Fig. 12 to obtain the area of the actual ground coordinates. Then, it compares the area against the size of the drone, in order to check whether the area is big enough to accommodate.

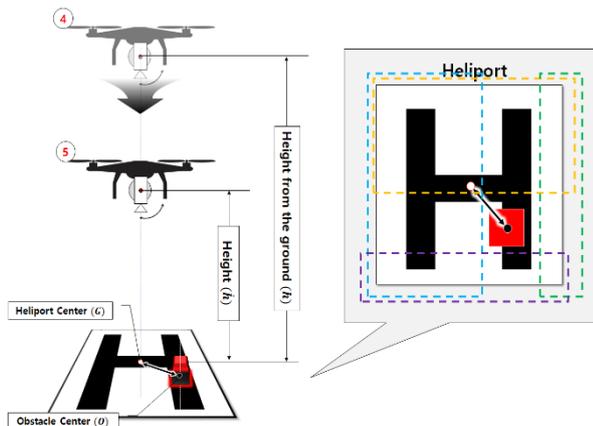


Figure 12. Flight control process of automatic landing algorithm-(2).

### III. DRONE AUTOMATIC LANDING ALGORITHM

The automatic landing algorithm, proposed in this study, is outlined as in Fig. 13. This is to enable a safe landing through 3D coordinates of the ground projection values. Besides, the recognition of landing points and obstacles from real-time images and calculated coordinates must be done simultaneously.

The companion computer we used is Jetson Tx2. This is mounted as shown in Fig. 14. This computer is capable of processing the Yolo v3 algorithm. The stepper motor (an open-loop system) adjusts the tilt angle between  $-30^\circ$  and  $-90^\circ$  to track the landing point. For the camera, the information regarding the internal parameters are known.

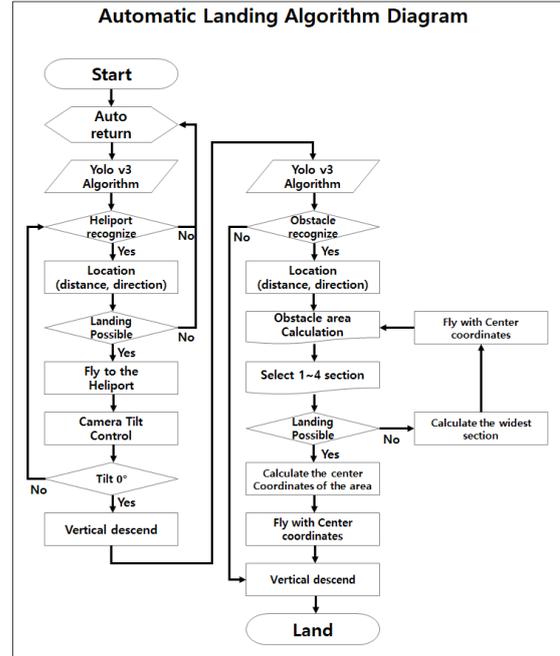


Figure 13. Automatic landing drone using AI-based image recognition technology.

### <Automatic Landing Drone using Video Recognition> Drawing

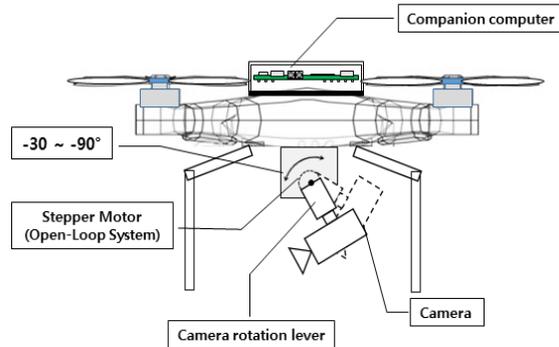


Figure 14. Drone and camera connection drawing.

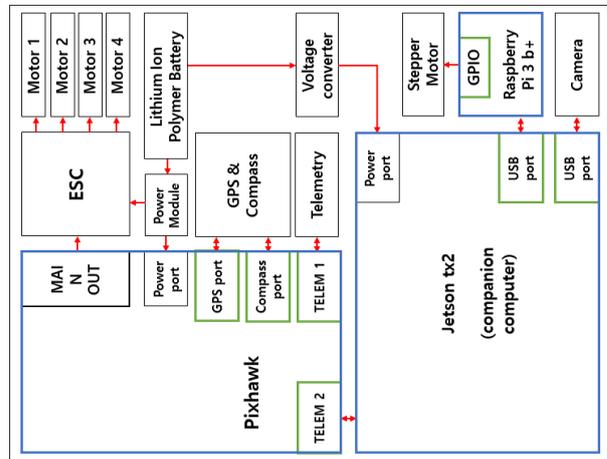


Figure 15. Drawings of drone-mounted devices.

Since the stepper motor cannot be directly linked with the Jetson Tx2, another companion computer, a Python-based module, Raspberry Pi, was used together. The

overall schematic drawing is composed as in Fig. 15. The Algorithm below is the Pseudo code for the automatic landing process of self-landing drone.

| Algorithm: Automatic landing |   |
|------------------------------|---|
| 1                            | <b>for</b> interaction = 0 (Start Yolo_v3 Recognition)  |
| 2                            | <b>if</b> Recognition(Heliport)<br>Measure $Tilt_{control}$ (Eq.2, Eq.3)  |
| 3                            | <b>if</b> (Heliport <sub>size</sub> > UAV <sub>size</sub> )<br>Flying by $\theta_{direction}$ and $Tilt_{control}$                |
| 4                            | <b>if</b> ( $\theta_{direction} > -90^\circ$ )<br>Start Yolo_v3   |
| 6                            | <b>if</b> Recognition(Obstacle)<br>Measure Obstacle <sub>size</sub> (Eq.5, Eq.6)  |
| 7                            | <b>if</b> (Section <sub>1~4</sub> is greater than UAV <sub>size</sub> by at least one<br>Choose the widest section<br><b>Land</b> |
| 8                            | <b>else if</b> (Recalculate)  |
| 9                            | <b>else</b> (Return to path)  |
| 10                           | <b>else if</b> (Recognize again Obstacle)   |
| 11                           | <b>else</b> (Return to path)  |
| 12                           | <b>else if</b> ( $Tilt_{control}$ )   |
| 13                           | <b>else</b> (Return to path)  |
| 14                           | <b>else if</b> (Recalculate)  |
| 15                           | <b>else</b> (Return to path)  |
| 16                           | <b>else</b> (return to path)  |

#### IV. ALGORITHM VERIFICATION

In order to verify the algorithm, it is necessary to check with the actual drone by loading the algorithm onto the companion computer [13, 14]. However, our area is urban and drone flight is strictly restricted. In the event of unwanted drone crash, there is a great risk of casualties as well as property damages. Therefore, the algorithm was verified by designing an environment similar to the actual flight environment using the Unity 3D program. Fig. 16 shows the constructed environment for simulated drone testing.

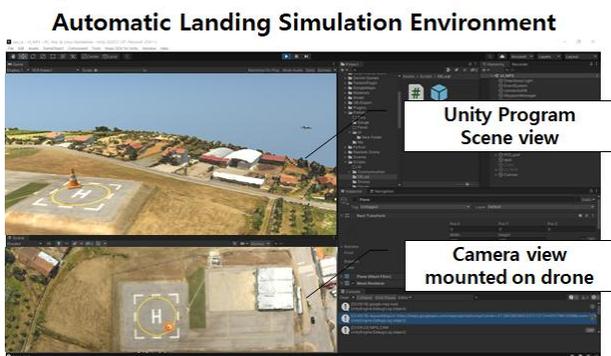


Figure 16. Unity 3D program simulation environment.

For our algorithm verifications, a drone model equipped with actual aerodynamics was used. Since our developed algorithm is to land drones at the correct location, when a

heliport is found during automatic landing, the experiment is conducted by setting the drone to discover the heliport from an arbitrary location. The drone model also carries the camera model that can see and send information to the algorithm. The overall setting and the workings of drone is very similar to the actual flight, as shown in Fig. 17.

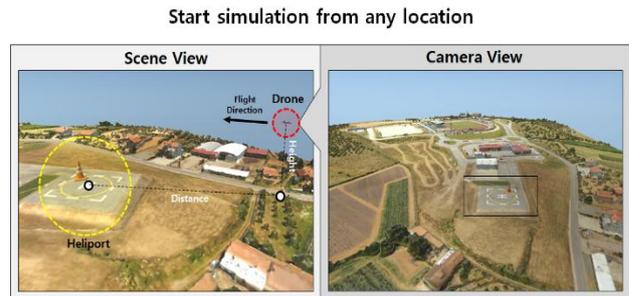


Figure 17. (Left)Unity Program simulation scene view / (Right)camera view mounted on the drone.

When configuring the simulation environment using the Unity 3D, obstacles were randomly placed on the heliport. As shown in Fig. 18, after flying to the landing position, it goes through the process of determining whether an obstacle is recognized through the Yolo v3 algorithm. In the Unity environment configured for the verification, obstacles on the heliport were recognized as tested. Then, it was confirmed that the drone successfully lands by following the process that is commanded by the companion computer. This is shown in Fig. 19.

#### Obstacle recognition after arriving on the heliport

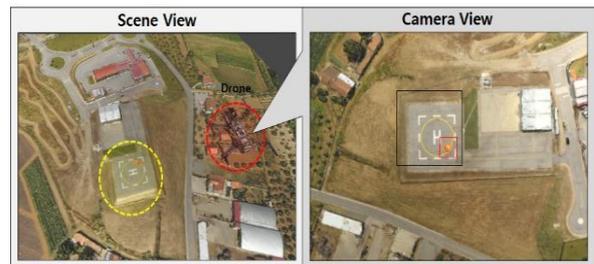


Figure 18. Obstacle recognition on the heliport.

#### Landing after avoiding obstacles

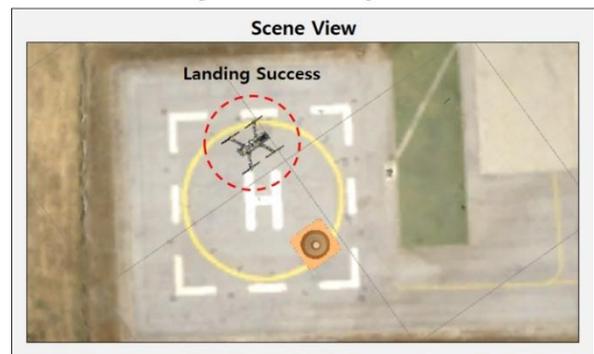


Figure 19. Lands safely with a distance after obstacle avoidance by algorithm.

The flight path of the simulated drone in the Unity environment was plotted by extracting x, y, and z values

every 0.1 second. Fig. 20 shows the path of the drone flying in the simulation environment. Since the actual dynamic model is applied to the drone flight, it is possible to confirm that noises such as vibration and wind. Despite, the drone can fly to the helipad and recognize obstacles, then land itself safely by avoiding them.

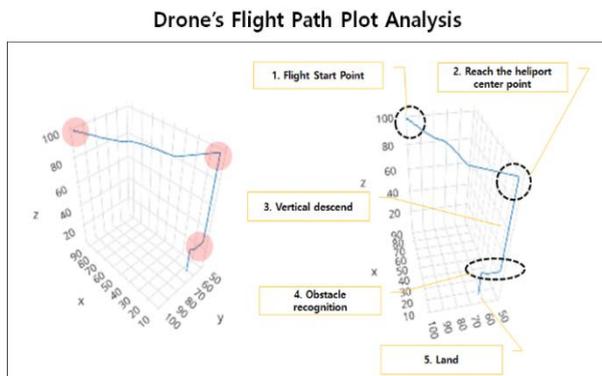


Figure 20. Flight path plot on unity environment.

## V. CONCLUSION

In this study, we developed the artificial intelligence-based algorithm that can identify objects, avoid obstacles, and safely guide the drone to the designated landing site. It was initiated by the fact that the actual landing cannot always become assisted by the ancillary devices, or the landing site cannot be always clear of obstacles or unexpected debris. The landing is the final, yet still unfinished part of the drone flight. Therefore, in order to conclude the drone flight, the landing must be performed without any accidents. The landing can be very hazardous, especially when the operator cannot see the landing site. In this situation, the proposed method can greatly enhance the level of safety by lowering the chance of drone crashing into the unwanted obstacles. In addition, since most drones are equipped with camera devices, our approach can be easily adapted to any drones. Therefore, the proposed method is very practical and can be safely landed in various environments without any equipment and auxiliary devices preinstalled in the landing site.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

Authors conducted the research; analyzed the data; and wrote the paper. All authors had approved the final version.

## ACKNOWLEDGMENT

This research was supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) and Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, the Republic of Korea (Grant Number: 2020M3C1C1A01084900).

## REFERENCES

- [1] D. Jo and Y. Kwon, "Development of autonomous VTOL UAV for Wide Area Surveillance," *World Journal of Engineering and Technology*, vol. 7, no. 1, pp. 227-239, 2019.
- [2] J. He, Y. S. Li, and K. K. Zhang, "Research of UAV flight planning parameters," *World Journal of Engineering and Technology*, vol. 3, no. 4, pp. 43-45, 2012.
- [3] L. Bin, "UAV Communications for 5G and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241-2263, 2018.
- [4] Truong, N. Quang, "Deep learning-based super-resolution reconstruction and marker detection for drone landing," *IEEE ACCESS*, vol. 7, pp. 61639-61655, 2019.
- [5] J. M. Kelner and C. Ziolkowski, "Doppler effect-based automatic landing procedure for UAV in difficult access environments," *Journal of Advanced Transportation*, vol. 2017, p. 9, 2017.
- [6] D. Maturana and S. Scherer, "3d convolutional neural networks for landing zone detection from lidar," in *Proc. 2015 ICRA*, pp. 3471-3478, May 2015.
- [7] M. G. Kimberly, D. C. Gudio, D. W. Christophe, T. Karl, K. Hilbert, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics Automation Letters*, vol. 2, no. 2, pp. 1070-1076, April 2017.
- [8] T. M. Thi, C. Cosmin, H. Andres, and D. K. Robin, "Obstacle avoidance using UAV in indoor environment," in *Proc. IEEE 14th International Symposium on Applied Machine Intelligence and Informatics*, pp. 21-23, 2016.
- [9] A. Gunawan, A. Chowanda, and J. S. Suroso, "Fast object detection for quadcopter drone using deep learning," in *Proc. 3rd International Conference on Computer and Communication System (ICCCS)*, 2018.
- [10] Z. L. Hao, X. H. Chen, J. Q. Hu, X. Wen, and X. D. Wang, "OpenCV-based automatic detection system for automobile meter," *Applied Mechanics and Materials*, vol. 615, pp. 149-152, 2014.
- [11] N. J. Uke and R. Thool, "Moving vehicle detection for measurements traffic count using OpenCV," in *Proc. 4th International Conference on Conference on Electronics Computer Technology (ICECT 2012)*, 2012.
- [12] A. Kam, Ng. T, N. Kingsbury, and W. Fitzgerald, "Content based Image retrieval through object extraction and querying," in *Proc. IEEE Workshop on Content Based Access of Image and Video Libraries*, 2000, pp. 91-95.
- [13] M. Yue, W. Wei, H. Hao, and B. Jingxuan, "A visual/inertial integrated landing guidance method for UAV landing on a ship," *Aerospace Science and Technology*, vol. 85, pp. 474-480, 2019.
- [14] G. S. L. K. Chand, M. Lee, and S. Y. Shin, "Drone based wireless mesh network for disaster/military environment," *Journal of Computer and Communications*, vol. 6, no. 4, pp. 44-52, 2018.

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License (), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.



**Sedam Lee** has many years of engineering experience in industrial engineering. He has conducted research applied to industrial engineering concepts in military and simulation. He has a major interest in the fields of simulation, unmanned systems, and UAV. He has studied his master's degree at Ajou University.



**Daeil Jo** has many years of engineering experience in industrial engineering and mechanical engineering. He has experience concerning R&D and manufacturing in Field. He carried out research applied to industrial engineering concepts of unmanned aerial systems. He has great interest in CAD & CAE and UAV. He received his master's degree from Ajou University.



**Yongjin (James) Kwon** is currently in the Department of Industrial Engineering, Ajou University. Before joining Ajou, he was on the faculty of Drexel University, located in Philadelphia, USA. While he was in Drexel, he obtained research funding from government agencies (NSF, DoED) and private companies. He published many international journal articles and presented in many conferences over the years. His research areas include AI-enabled automation systems, military AI, defense simulation, and drones.