Safe Landing of Drone Using AI-based Obstacle Avoidance

Sedam Lee and Yongjin Kwon
Ajou University, Dept. of Industrial Engineering, Suwon, South Korea
Email: lsd2547@ajou.ac.kr, yk73@ajou.ac.kr

***Abstract*—As the 4th Industrial Revolution being underway, many research works on drones have been actively conducted. One of the most important part of the drone technology is now dwelling on the autonomous identification and avoidance of obstacles during the flight. In usual cases, drones are following the waypoints designated before the flight by relying on the GPS signals. However, when drones are approaching the designated landing site, there might be obstacles and unforeseen objects that may critically jeopardize the safe landing of the drones. Therefore, the safe landing of the drone is becoming a very important issue. In this respect, this study investigates the possibility of applying artificial intelligence (AI) techniques to the drone, in order to enhance the safety. By integrating image sensors, AI-enabled object recognition, and drone flight control computer altogether, the drones can be more safely landed without the fear of being overturned or critically damaged due to unexpected obstacles during the landing phase of the flight.**

***Index Terms*—landing platform tracking, obstacle avoidance, image segmentation, artificial intelligence, two-dimensional coordinates, flight control**

I. Introduction

Since the beginning of the Fourth Industrial Revolution, many research works on drones have been actively carried out [1]. Drones are now being widely used in areas, such as reconnaissance, broadcasting, surveillance, transport and rescue, to just name a few [2] [3]. With the increasing availability of drones, the safety of drone operations during takeoff, landing, and flight is being studied. One of the most important part of the drone technology is now focusing on the autonomous identification and avoidance of obstacles during the flight. In usual cases, drones are following the waypoints designated before the flight by relying on the GPS signals. However, when drones are approaching the designated landing site, there might be obstacles and unforeseen objects that may critically jeopardize the safe landing of the drones. Especially in the urban environment, where landing sites tend to be small (i.e. a rooftop helipad), it becomes really difficult to safely land because there is a very little margin of error during the landing. In part, this is because the GPS signal may not be so accurate. Additionally, there might be unexpected obstacles on and around the helipad. If a drone is blindly trying to land just by relying on the GPS signals, an accident may occur, as shown in Fig. 1. Such situation may critically jeopardize the safe landing of the drones, hence the obstacle avoidance is becoming a very important issue [4]. Yet, it has to be autonomously conducted.

In this respect, this study investigates the possibility of applying artificial intelligence (AI) to the drone, in order to enhance the safety of landing. By integrating image sensors, AI-enabled object recognition, and drone flight control computers altogether, the drone can be safely landed without the fear of being overturned or critically damaged due to unexpected obstacles during the landing phase of the flight.

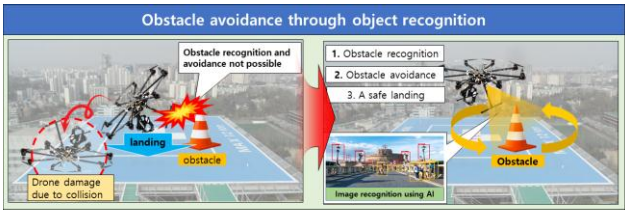

Figure 1. Conceptual diagram of obstacle avoidance in landing

The structure of this paper is as follows. Chapter 2 discusses the related works. Chapter 3 describes the connection between a flight control computer (FCC) and a companion computer. Chapter 4 describes the application of AI techniques on the flight control of the drone, while Chapter 5 describes the AI algorithm verification. The final chapter illustrates the findings and concluding remarks from this study.

II. Related Works

Various studies have been done to automate the landing procedure of UAVs (unmanned aerial vehicles). In order to achieve an automated landing and obstacle avoidance, the drone should be able to find and determine the landing site and obstacles. In most cases, a separate device that assists the drones to land safely is installed and used at the landing site. One paper, entitled "Doppler Effect-Based Automatic Landing Procedure for UAV in Difficult Access Environments," explains this approach. The Doppler effect was used in this paper and a radio beacon was installed to aid the drones for landing [5]. However, it can be cumbersome to use the radio beacon at the landing point, especially when the landing sites can

Manuscript received February 3, 2020; revised September 12, 2020.

doi: 10.18178/ijmerr.9.11.1495-1501

be changing everytime. Another approach is proposed in the paper, entitled “3D Convolutional Neural Networks for Landing Zone Detection.” The LiDAR (laser imaging detection and ranging) was proposed as a method to detect landing space and obstacles [6]. In this case, the LiDAR sensor’s accuracy should be very important. Despite, the use of very accurate LiDAR sensor on drones can be quite expensive.

Another more common method is to use the optical flow sensor for controlling UAVs during the landing. One paper entitled, “Efficient Optical Flow and Stereo Vision for velocity Estimation and Obstacle Avoidance,” explains the obstacle avoidance system with the use of optical flow and stereo vision sensors [7]. This combination of sensors is probably the most common at this stage. Nonetheless, in terms of performance, the obstacle avoidance can be accurate, yet difficult to distinguish the type of obstacles. Many different ways of avoiding obstacles are being proposed, such as the paper entitled, “Improved Potential Field Method for Unknown Obstacle Availability Using UAV in Indoor Environment [8].” In this paper, by using on-board visuals and inertial sensing devices, a different method was proposed. However, the same problems still exit, such that it is not easy to find a landing point.

By considering many options and the cost problems, we propose AI-based image recognition approach to find the landing point and to enable the obstacle avoidance. The AI algorithm was based on the CNN-based Yolo v3 algorithm, which is currently widely used in the community and industry. Then, we propose a method of controlling how the drone flies, according to the coordinate region of each recognized object by dividing the image coordinate into nine regions. By doing so, a drone can be equipped with an intelligent function that enables the UAVs a way to find the landing site and furthermore, to avoid the obstacles during the landing. This is very necessary in comparison to the previous approach of using an image-based feature recognition method, which was not effective in finding the obstacles.

III. Connection between Flight Control Computerand Companion Computer

The easiest and most economic way of integrating image sensors with the drone flight control computers is using companion computers. Those are very small and inexpensive, yet many different types are commercially available. Companion computers can control the drones using algorithm codes that process the signals from image sensor. This requires communication with a flight control computer (FCC). In this study, we used the Raspberry Pi as a companion computer. The hardware connection between the FCC and the Raspberry Pi is usually connected directly to the GPIO pins. However, in order to improve the stability of communication, a serial communication method using 'cp2102 UART to TTL serial' module that converts data into serial information has to be used. After the hardware connection is established, Mavlink, the command standard protocol of the drone, is used to facilitate the Raspberry Pi's flight control commands. The Raspberry Pi also enables the AP (access point) feature to control the drones. For this purpose, we used the SIFI dongle equipped with the AP function, a product provided by Netis. Fig.2 illustrates the setup of our work.

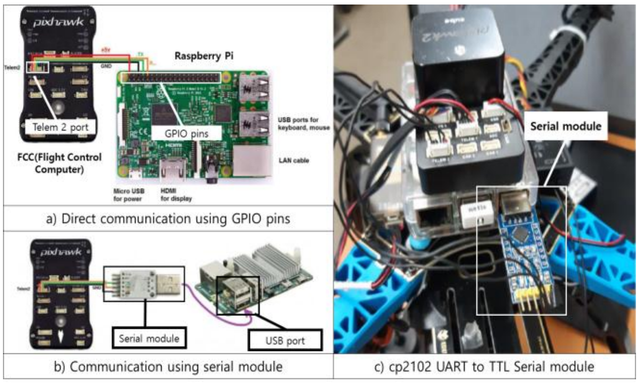

Figure 2. Connection between FCC and companion computer

Mavlink can be applied to various companion computers, while a camera module should be connected as well. Therefore, a relatively light and inexpensive camera module has been connected to the companion computer Raspberry Pi, as shown in Fig. 3 (a). After the connection is successfully conducted and the image control algorithms have been verified, a more expensive image sensor and a high-end companion computer have been tested, namely a ZED camera and a Jetson Tx2 Board for higher computing performance and fast data processing [9]. This is shown in Fig. 3 (b), which is needed to install the obstacle recognition algorithms.

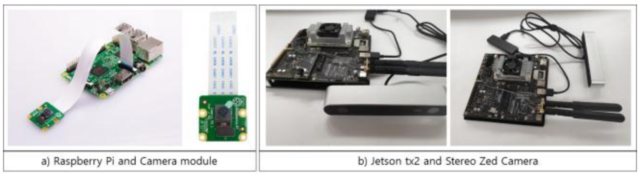

Figure 3. Camera Modules and Companion Computers

IV. Drone Flight Control

A. *Flight Control Algorithm*

When any object is recognized, the drone needs a criterion of judgment as to whether track or avoid the object. The pixels of the image plane are divided and applied to two-dimensional coordinates. After setting the pixel of the camera image to 640 x 480, the center point of the image was set to (0, 0), as shown in Fig. 4, The camera image is divided into 9 sections. If any obstacle is detected in any section, then the drone would move according to the directions set by Table I. The center section and the surrounding ones have a different size configuration, due to the fact that the camera center is the most critical area of landing. The center area is actually where the drone would be touching first during the final phase of the landing.

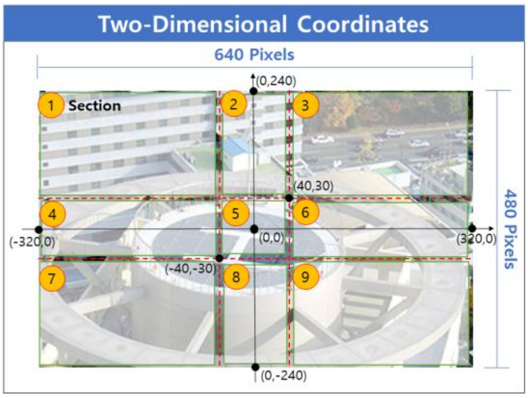

Figure 4. Camera Image Divided into 9 Sections

For example, if the recognized object is a landing platform (i.e., a helipad) and its position is at the section number 9, the drone should fly to <Back-ward-Rightward> to land. If the drone flies to <Backward-Rightward> and the helipad becomes placed inside the section 5, the drone enters the hovering state. During the hovering state, the drone descends down to the preset altitude. The drone then checks again if there is any obstacle. If there is no obstacle, the final landing begins. Fig. 5 and Table I indicate such directional control algorithm during the landing.

TABLE I. FLIGHT INDICATORS BY OBJECT TYPE AND LOCATION

Sort	Flight control by section	
Color	Blue circle =Helipad recognition	Red square=Obstacle recognition
1st Section	Forward - Leftward	Backward - Rightward
2nd Section	Forward	Backward
3rd Section	Forward - Rightward	Backward - Leftward
4th Section	Leftward	Rightward
5th Section	Hovering	Rightward
6th Section	Rightward	Leftward
7th Section	Backward - Leftward	Forward - Rightward
8th Section	Backward	Forward
9th Section	Backward - Rightward	Forward - Leftward

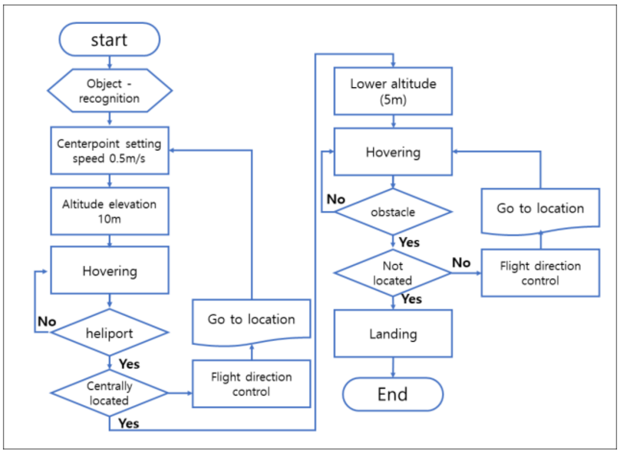

Figure 5. Landing control algorithm

B. *Image Based Object Recognition*

The image recognition using the Raspberry Pi and an inexpensive camera module turned out to be difficult, due to the low image processing speed. This makes it almost impossible to recognize objects using the artificial intelligence algorithm, Yolo v3, which has to handle a lot of computational work. The Raspberry Pi can still be used to recognize a very simple object, and we used the OpenCV library to identify objects [10].

1) *OpenCV based object recognition.*

First, the image is analyzed to determine the shape of the object. The number of vertices is identified and the outline of the recognized object is derived. The code is implemented to determine the shape of objects. In order to process an image, we used a Mat object, which is the basic image repository of OpenCV. This reads the images in BGR order, with a range of 0 to 255 in the form of 8 bits. The image is stored as a number in the Mat object and displayed using the HighGUI module. In order to recognize and identify the desired color in the image, the HSV conversion method that expresses the brightness of hue, saturation, and luminance is used [11] [12]. In our testing, the landing platform and obstacles were defined as blue circles and red squares, respectively, as shown in Fig. 6.

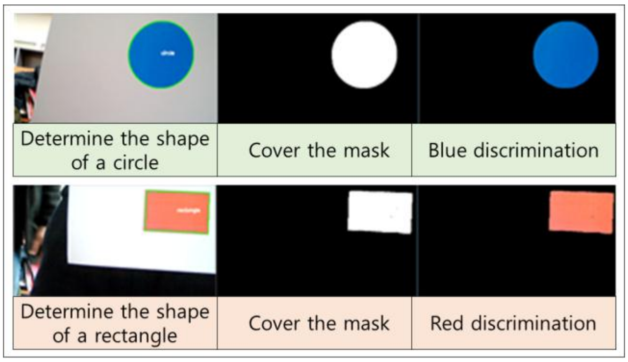

Figure 6. Object recognition using the raspberry pi and camera module

2) *Yolo v3 based object recognition*

After the successful testing of the Raspberry Pi and camera module, a higher processing computer device, the Jetson Tx2 board is integrated. The actual landing control is carried out by using the Jetson Tx2 board and the ZED camera. The objects are recognized using the AI algorithm, Yolo v3, which is divided into N x N Grid cells and computes the objects in each cell. The Jetson board is capable of running the AI algorithm. The boundary boxes are used to determine the position and size of the object. Through this, it is possible to check the center of the detected object (x, y) and the width (w) and height (h) of the object. This coordinate information makes it easy to see exactly where the object is located in camera coordinates [13] [14].

In order to recognize the landing platform and obstacles, more than 1000 helipad images and 1000 obstacle images were collected and trained. In training of AI, two classes are specified, namely the helipad and the obstacles. Landing point and heliport learning process using CCN-based Yolo v3 Algorithm is shown in Figs. 7 and 8.

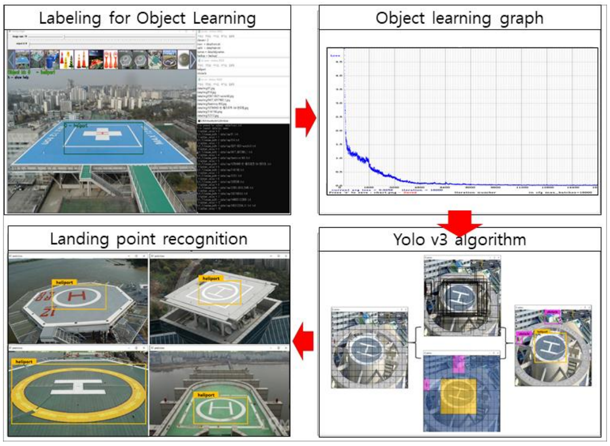

Figure 7. Landing point learning process

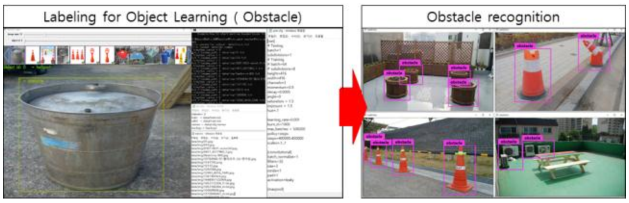

Figure 8. Obstacle learning process

The number of training is carried out with an iteration of 16,000 times, and the filters are 7 (4 + 1 + classes). However, as the number of learning reaches 8,000 iterations, the loss is reduced to 0.0432, as shown in Fig. 9. After that, the amount of recognition error is no longer reduced and becomes over fitted, resulting in an inaccurate object recognition.

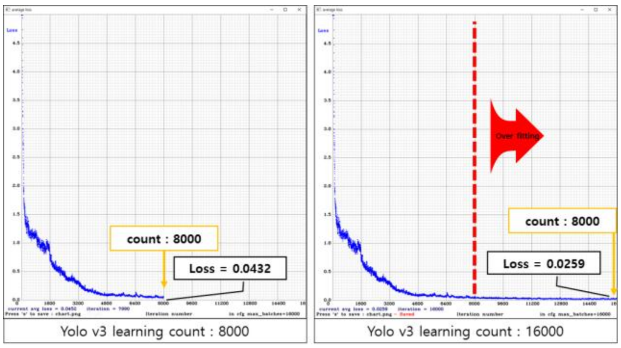

Figure 9. Loss graph and over fitting

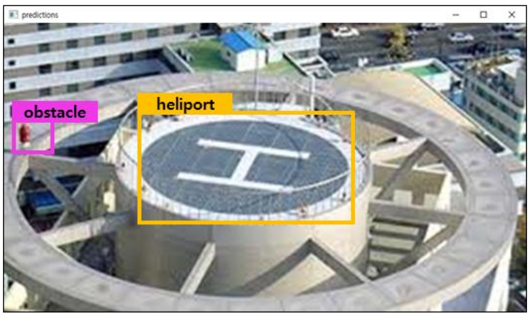

Figure 10. Object recognition using Yolo v3 algorithm

The Yolo v3 algorithm, then, used the weight of 8,000 iterations. When applying the threshold of 0.9 to the recognition of helipad and obstacles, the algorithm can successfully recognize both types, as in Fig. 10.

V. Algorithm Verification

Once the object recognition AI algorithm has been successfully developed, a multicopter drone, as shown in Fig. 11, is assembled. The left-side image shows the drone equipped with the ZED camera and the Jetson Tx2 board, while the right image shows the drone frame with only the Pixhawk FCC and motors installed. The drone has 6 motors and weighs about 3 Kilograms. Fig. 12 shows the actual test flights for verification. The test flight was conducted in our university campus, and successfully demonstrated that the developed algorithm was working fine.

Figure 11. Assembly of drone with the necessary components

Figure 12. Test flight of drone

Fig. 13 shows that the drone flies 'Backward – Rightward' because the landing platform is located in section 9. When the landing platform is in section 5, the altitude is lowered. Once again, if the object recognition algorithm confirms the presence of no obstacles on the landing platform, the drone can safely land. If an obstacle is recognized as shown in Fig. 14, the drone flies ‘Back-Right-Rightward’ so that the obstacle is not visible in the camera image. If no obstacle is further recognized, the landing can be carried out. In Fig. 15, the flight path was visualized by extracting the coordinates of the drone's flight path from the Python-based Dronekit SITL simulation program. The visualization was conducted after several flight tests have been completed. In actual test, one student carried the printed plate that contains the object type (i.e., helipad), and let the drone see the plate and follow the object as the student slowly walking towards the landing site. The drone successfully recognized the object and the FCC automatically guided the drone to follow it. Table II shows the flight path for our actual testing, denoted as (a) to (j). Through this, we can confirm that the implemented algorithm has been normally applied. (10 pixels of image coordinate = 1 meter).

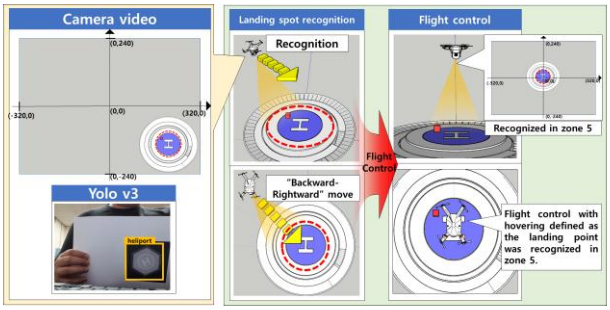

Figure 13. Landing platform recognition and direction control

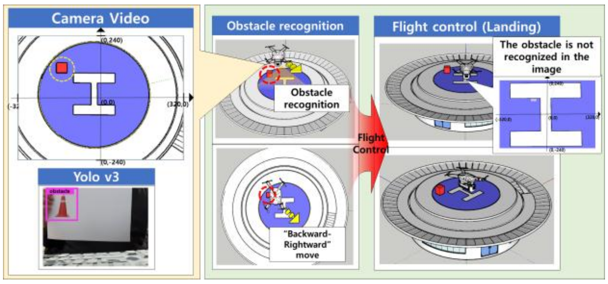

Figure 14. Obstacle recognition and direction control

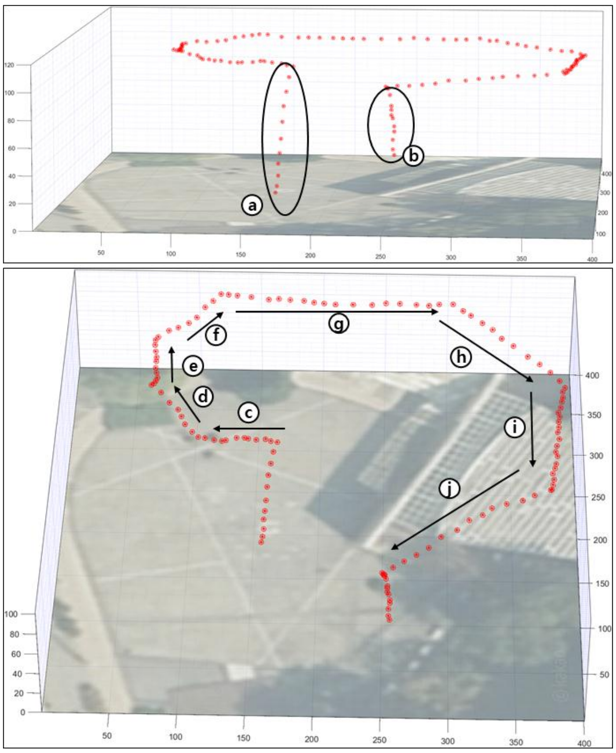

Figure 15. SITL drone flight trajectory

By showing the working algorithms in Table III and confirming that the developed algorithm is actually controlling the drone away from the detected obstacle, one can assume that the drone can be a lot safer during the landing. This can actually minimize the potential loss of expensive cargo carried by the drone, and also reduce the accident rate of the drone.

TABLE II. DRONE TRAJECTORY TABLE

Point	Flight Direction	Camera Image Section
a	(10 meter) Altitude elevation	-
b	(5 meter) Altitude descent	Section 5
c	Leftward	Section 4
d	Forward - leftward	Section 1
e	Forward	Section 2
f	Forward – rightward	Section 3
g	Rightward	Section 6
h	Backward – rightward	Section 9
i	Backward	Section 8
j	Backward - leftward	Section 7

TABLE III. EXCERPT FROM FLIGHT CONTROL CODE

```
//Set center coordinate of camera image of 640*480 Pixel

def locationX(cX):
    return cX - 320
def locationY(cY):
    return 240 - cY
//Calculate the center point of an object

for c in cnts_blue:
M = cv2.moments(c)
if int(M["m00"]) != 0:
cX = int((M["m10"] / M["m00"]))
cY = int((M["m01"] / M["m00"]))
LoX = locationX(cX)
LoY = locationY(cY)
print ("X: ", LoX,"Y: ", LoY)

//Flight control so that the landing platform is in the center of the camera pixel

    if LoX > 40:
        if LoY > 30 :
            print ("Go Foward and Rightward")
            set_velocity_body(vehicle, gnd_speed, gnd_speed, 0)
        elif LoY < -30 :
            print ("Go backward and Rightward")
            set_velocity_body(vehicle, -gnd_speed, gnd_speed, 0)
        else:
            print ("Go rightward")
            set_velocity_body(vehicle, 0, gnd_speed, 0)
    elif LoX < -40:
        if LoY > 30 :
            print ("Go Forward and Leftward")
            set_velocity_body(vehicle, gnd_speed, -gnd_speed, 0)
```

```
        elif LoY < -30:
            print ("Go Backward and Leftward")
            set_velocity_body(vehicle, -gnd_speed, -gnd_speed, 0)
        else:
            print ("Go Leftward")
            set_velocity_body(vehicle, 0, -gnd_speed, 0)
    else:
        if LoY > 30:
            print ("Go Forward")
            set_velocity_body(vehicle, gnd_speed, 0, 0)
        elif LoY < -30:
            print ("Go backward")
            set_velocity_body(vehicle, -gnd_speed, 0, 0)
        else:
            print ("Target Reached")
            set_velocity_body(vehicle, 0, 0, 0)
            if detect_landing platform == 10:
                break
            detect_landing platform += 1
        if cv2.waitKey(1) == 27:
break
```

VI. Conclusion

This paper addresses the development of flight control algorithm for safe landing of drone, instead of solely relying on the GPS signals. Raspberry Pi, one of the companion computers, was first used for object recognition and flight control. The drone was later equipped with a more powerful companion computer, in order to implement the AI image process algorithm. The image-based object recognition allows the drones to safely land on a preprogrammed site, despite the unexpected presence of the obstacles. Such capability gives the drones to actually see and detect the obstacles during the final decent phase of the flight. During the actual test flights, due to the high data throughput, there was a delay in each image frame. In addition, it was difficult to check the drone's camera image and data processing in real time. Despite, it was confirmed that the drone can recognize the landing platform according to the trained algorithm and safely lands by avoiding obstacles. Even though our testing was limited, as a result of this study, one can expect that the drone can fly more safely without the fear of losing expensive cargo and drone itself.

As the technology continues to advance, the proposed method can also progress further in terms of recognizing a wide array of obstacles under any conditions. One critical bottleneck in implementing our proposed method was the processing speed of AI algorithm inside the drone. The drone cannot fly with the high performance computer, and the current version of the processing computer that was installed onto our drone was slow to do the request work. One way of circumvent this problem is to use the 5G network technology. Drone can just send the captured images on to the ground station, and the computer situated in the ground station processes the AI algorithm and directly resend the output (i.e., the drone control commands) to the drone. By doing so, the drones can be free of installing heavy computing devices, and the obstacle avoidance performance can be also enhanced.

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 work was supported in part by a grant from LigNex1 Company.

References

[1] D. Jo, 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, 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, pp. 9, 2017

[6] D. Maturana and S. Scherer, "3d convolutional neural networks for landing zone detection from lidar," in 2015 ICRA, pp. 3471–3478, May 2015.

[7] M. G. Kimberly, D. C. Gudio, Christophe D.W, Karl T, Hilbert K, "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, D. K. Robin, "Obstacle avoidance using UAV in indoor environment," in *Proc. IEEE 14th International Symposium on Applied Machine Inteligence and Informatics*, pp. 21~23, 2016.

[9] A. Gunawan, A. Chowanda, 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, X. Dong Wang "OpenCV-Based automatic detection system for automobile meter," *Applied Mechanics and Materials*, vol. 615, pp. 149-152, 2014.

[11] N. J Uke, R. Thool, "Moving vehicle detection for measurings traffic count using OpenCV," in *Proc. 4th International Conference on Conference on Electronics Computer Technology (ICECT 2012)*, 2012.

[12] A. Kam, T. Ng, N. Kingsbury, W. Fitzqerald, "Content based Image Retrieval through Object Extraction and Querying," *IEEE Workshop on Content Based Access of Image and Video Libraries*, pp. 91-95, 2000.

[13] M. Yue, W. Wei, H. Hao, 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. Siva L. Krishna Chand, M. Lee, 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.

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.

Yongjin Kwon has many years of engineering experience in industrial and academic settings. He has extensive experience & practical knowledge in current design, manufacturing and quality control. His work has been cited a number of times in high profile journals. He is currently a professor in the Department of Industrial Engineering at Ajou University. Prior to joining Ajou, he was on the faculty of Drexel University, Philadelphia, USA.