ISSN: 0067-2904

Automatic Object Detection, Labelling, and Localization by Camera's Drone System

Rasool D. Haamied[1], Bushra Q. Al-Abudi, Raaid N. Hassan
Department of Astronomy and Space, University of Baghdad, Baghdad, Iraq

Received: 26/**10**/2021 Accepted: 31/11/2021

Abstract

This work explores the designing a system of an automated unmanned aerial vehicles (UAV) for objects detection, labelling, and localization using deep learning. This system takes pictures with a low-cost camera and uses a GPS unit to specify the positions. The data is sent to the base station via Wi-Fi connection.

The proposed system consists of four main parts. First, the drone, which was assembled and installed, while a Raspberry Pi4 was added and the flight path was controlled. Second, various programs that were installed and downloaded to define the parts of the drone and its preparation for flight. In addition, this part included programs for both Raspberry Pi4 and servo, along with protocols for communication, video transmission, and sending and receiving signals between the drone and the computer. Third, a real-time, modified, one dimensional convolutional neural network (1D-CNN) algorithm, which was applied to detect and determine the type of the discovered objects (labelling). Fourth, GPS devices, which were used to determine the location of the drone starting and ending points . Trigonometric functions were then used for adjusting the camera angle and the drone altitude to calculate the direction of the detected object automatically.

According to the performance evaluation conducted, the implemented system is capable of meeting the targeted requirements.

Keywords: Drone, Object detection, Distance, CNN, Camera, Servo.

الكشف التلقائي للأجسام وتحديد نوعها ومعرفة موقعها بواسطة كاميرة محمولة على الطائرة بدون طيار

رسول دهيم حميد*، بشرى قاسم العبودي، رائد نوفي حسان
كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

الهدف من هذا العمل تصميم نظام آلي للطائرات بدون طيار لكشف الأجسام وتحديد نوعها ومعرفة موقعها وبعدها باستخدام نظام التعلم العمي ، الذي يلتقط الصور بكاميرا منخفضة التكلفة ويستخدم وحدة GPS لتحديد المواقع ترسل البيانات إلى المحطة الأساسية عن طريق اتصال Wi-Fi.

تم وصف اربعة أجزاء رئيسية من النظام المقترح وهي:

*Email: rasoolhaamied@gmail.com

أولا: تجميع اجزاء الطائرة بدون طيار، وإضافة Raspberry Pi4 والتحكم في مسار الطيران.
ثانيا: تنزيل وتثبيت برامج لتعريف أجزاء الطائرة وتجهيزها للطيران وكذلك برامج لكل من Raspberry Pi4 و servo وبروتوكولات الاتصال ونقل الفيديو وإرسال واستقبال الإشارات بين الطائرة و الكمبيوتر.
ثالثاً: معالجة الصور الرقمية المأخوذة من الفيديو الذي تم التقاطه بواسطة الكاميرة في الوقت الحقيقي وباستخدام خوارزمية CNN1D المعدلة والتي يتم تطبيقها لاكتشاف وتحديد نوع الأجسام المكتشفة .
رابعا: الكاميرة المستخدمة موجهة إلى الأسفل مع بزاوية 45^{o} حول المحور الرأسي واستخدام GPS لتحديد موقع نقطة بداية طيران الطائرة والموقع النهائي ، ثم باستخدام المعادلات المثلثية لزاوية الكاميرا وارتفاع الطائرة بدون طيار تم حساب بعد الجسم المكتشف واتجاهه تلقائيا.
ووفقا لتقييم الأداء الذي أجري، فإن النظام المنفذ قادر على الوفاء بالمتطلبات المستهدفة.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have drawn a lot of interest for their image identification capabilities on both vehicles and humans. This interest has been prompted via military applications, such as aerial surveillance, as well as automatic target recognition in commercial and civil applications, including rescue missions and traffic search [1-4]. Various images or videos must be taken and analyzed in such applications, the majority of them must be processed in real-time.

UAV technology is currently employed in various applications, including security, delivery, and remote sensing. Passive optical sensors, like HD digital cameras, are low-power and lightweight sensors which might be utilized to build a more typical "see-and-avoid" system [5]. Furthermore, they create a new type of symbolic information as high-dimensional data which is compatible with the natural environment, allowing for additional decision-making.

The type of UAV employed, along with the basis of flight, its components, and the types of equipment linked to it that enhance the actual speed of the task in real-time, are all covered in this work. These components included Raspberry Pi model [6], MAVLink protocol [7], quadcopter components, DroneKit API [8], ArduPilot open-source autopilot software [9], a web camera, and a ground control station.

2. Related Work

Several research projects have looked into discovering and detecting ground objects of interest from aerial imageries. Since they contain a large amount of data, researchers have challenged the task of understanding these imageries with various ideas and proposals.

Al-Qaraawi *et al*. (2018) used a quadcopter with an attached camera to observe the video transfer process, utilizing several protocols and a monitoring system connected to Raspberry Pi [10]. This work presented an important assessment of video transfer by means of a quadcopter-based Ad-hoc network. The shortcoming is that they employed a quadcopter with manual control as well as a radio transmitter to record videos at the appropriate location.

Kamran *et al*. (2018) investigated the ability to recognize military vehicles and separating them from non-military vehicles on low-altitude aerial imageries, using Faster R-CNN. The experimental results revealed that the use of a customized/prepared dataset for training deep architectures allows them to recognize 7 military and 4 non-military vehicle types [11].

Uus *et al*. (2019), using YOLOv3, looked at identifying various sorts of vehicles from aerial images. To deal with high-resolution, huge images, they developed a method of splitting/recombining and augmenting images for the detection of their components [12].

3. The proposed system

The proposed method involves the design of a drone for the real-time detection and labelling of objects from captured images/videos using a modified 1D-CNN. The block diagram in figure (1) describes the main steps of the proposed system. More details are explained in the following sections.

Figure 1- Block diagram of the proposed system

3.1 Drone Construction and Installation

The prototype of the modified drone that was used in this study was built entirely using materials obtained from the local market. The construction system has two major components, which are the drone itself and a personal computer (PC). Table (1) contains the specifications of the modified drone prototype.

Table 1-The specifications of the drone prototype

Device	Specifications
Frame	Carbon fiber , Model: ZD550 Pro [13]
Motor	5010 360Kv High Efficiency Brushless Motor [14]
Propeller	8×45 Set Carbon Fibre Reinforced
ESC	The Hobbywing XRotor 40A [15]
Battery	The Li-Po , 5400mAh
Radio Controller	Flysky FS-i6S 2.4GHz 10CH AFHDS 2A RC Transmitter With FS-iA10B 10CH Receiver [16]
Flight Controller	Pixhawk [17]
Raspberry	Raspberry Pi4 Model
GPS	Ublox NEO-M8N GPS with compass (ROM) [18]
Camera	1080P Webcam Camera Digital Smart USB
Servo	Model Number: LF-20MG

The drone was built with a higher number of small systems. This type of drones is typically connected with a LAN utilizing WIFI. Other types of remote systems have long ranges, based upon satellite controls. The Raspberry Pi4 represents the control and computation unit that has a wide range of abilities and serves the aim of the present work.

Figure (2) shows the parts that were used and the way to connect them. User's communication can be achieved with Raspberry Pi4 for sending commands to the drone's control unit and receiving the data back from Raspberry Pi4. These data imply detection, object labelling, object direction, and object distance.

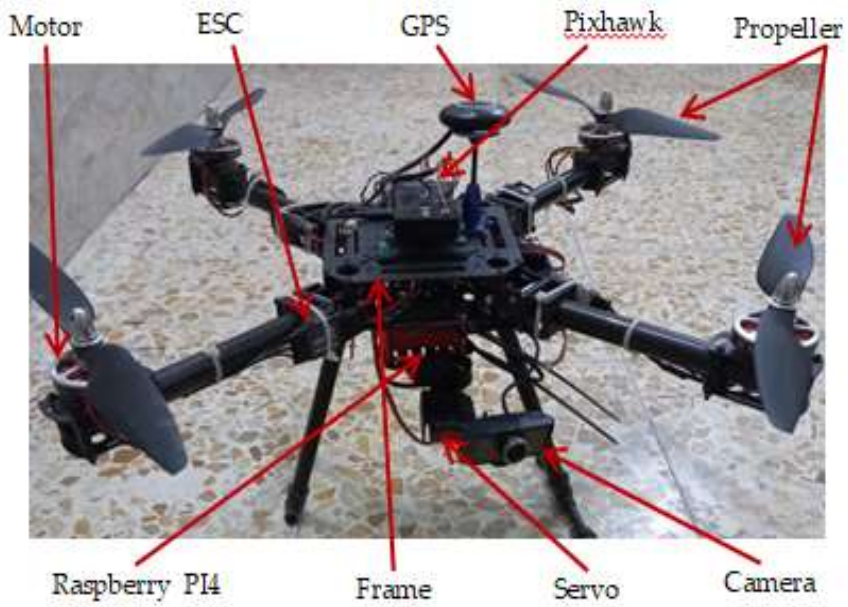

Figure 2-The drone components.

3.2 Drone's Programs

In order for any drone to function, software is a key component. First, the control of the drone, using WiFi over RF (radio frequency), is needed. The commands are executed in real-time. Upon connection, the drone will prepare to accept an instruction file. At this point, on startup, Raspberry Pi 4 sends a signal to ensure that the drone is ready to fly. The Pixhowk in turn scans the motors of all propellers to ensure the absence of operational issues, as well ensuring that the drone can take GPS coordinates. Then the system is ordered to be converted to the guided-flight mode controlled by the PC and, finally, it is ready to fly. Figure (3) illustrates the main components of the operations that involve Raspberry Pi4.

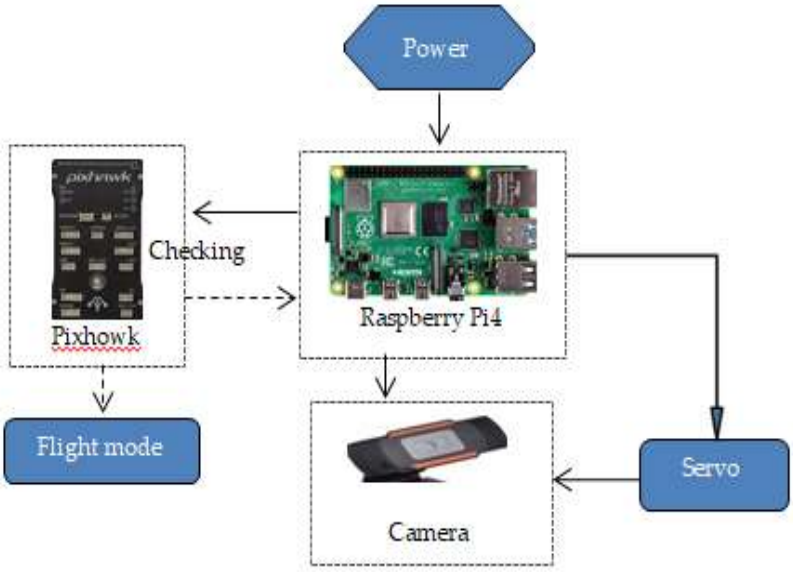

Figure 3-The Raspberry Pi4 operation.

3.2.1 Software Implementation

The use of software is the most common way of coordinating a product application within a work process. Before execution, the product ought to be selected through an evaluation

process that is based on the needs, expected advantages, obstructions, etc. When the arrangement is complete, the execution process can start.

A. Programming the Raspberry Pi4

The drone carries a Raspberry Pi4 which runs a distribution of Debian Linux, known as Raspbian. This distribution of Raspbian comes with Python 3.6.5 software and its development environment. By running a custom Python script, the Raspberry Pi4 is able to handle the operation and its additional subsystems. The operating system (OS) is downloaded and copied to a microSD card. The software selected in the present project is referred to as NOOBS (i.e. New Out Of the Box Software). It can give the user the option of the OS from standard distributions. This software is a simple OS Raspberry Pi4 installation manager.

B. GPS Data

Another goal of the automation process was to implement a system that would allow the drone to find the GPS coordinates for any point that the drone passes. In order to get real-time GPS data, the geographic library was used and the setup was entered into the general program for receiving new data from the GPS module, such as latitude, longitude, and altitude of both the drone and the object detected.

C. Accessing Raspberry Pi4 from a Host PC

There are 2 steps for accessing a system from some remote device. First, through a terminal interface, referred to as the secure shell (SSH), which is a network communication protocol [19]. Second, with the use of software, referred to as the virtual network computing (VNC) server [20]. This software allows a graphical user interface (GUI) to be remotely reflected on Raspberry Pi. The IP address of Raspberry Pi4 is determined initially through running the (ifconfig) command in a terminal window, as can be seen in figure (4).

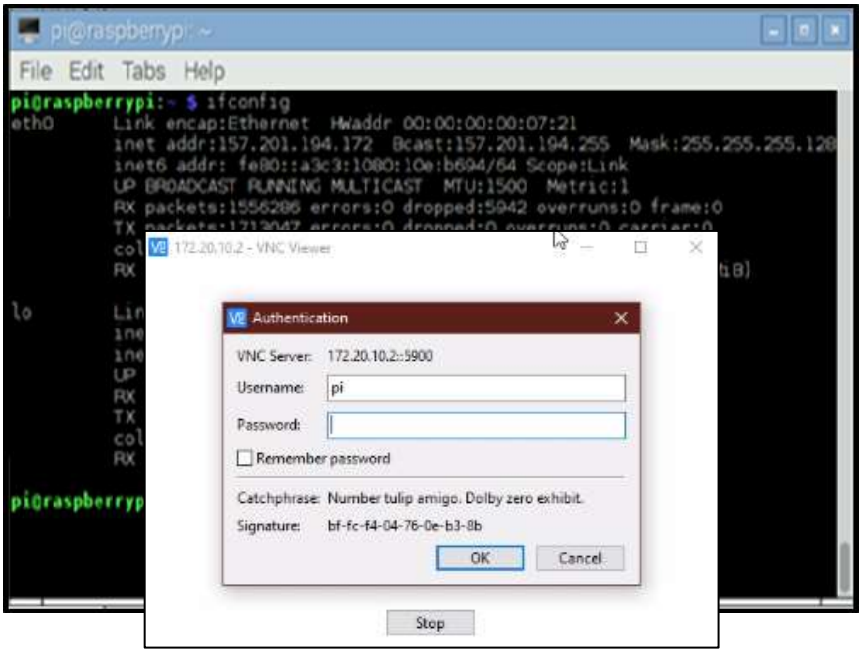

Figure 4-The IP address of the Raspberry Pi4.

3.3 Object Detection Programs

In neural networks, the convolutional neural network (CNN) is one of the main categories to perform image recognition. It takes an input image, processes it, and classifies it under certain categories. The CNNs work similarly, whether they are1-D, 2-D, or 3-D. The difference is represented by the input data structure, as well as the way by which the filter, also referred to as the feature detector or the convolutional kernel, moves over data.

A suggested solution is implementing a deep learning-based software that utilizes a CNN for the tracking, detection, and classification of the objects from the raw data in real-time. In addition to that, CNN enables the UAVs to convert the information of the object from the immediate environment to the abstract information, which may be comprehended by a machine with no human interferences. According to available data, the machines are capable of executing the tasks of real-time decision-making.

This part includes the implementation of preprocessing programs on each frame of the images selected from the video, extraction of features for each object, and then applying deep learning (CNN) algorithms to classify (label) and detect objects.

3.3.1 Images Preprocessing

The preprocessing of the selected images is the first stage in the proposed detection system. It includes three steps that aim to enhance the images that are going to be sorted in the classification step; First, the conversion of the RGB image to a gray-scale level image; second, applying the histogram equalization method; third, resizing the object image [21].

3.3.2 Experimental Results of the Object Image Detection and Labelling

This section presents the results of our proposed approach for detecting the objects in real-time from video-captured images by using the 1D-CNN model.

A multi-object detection model was constructed with the use of the proposed 1D-CNN model for the detection of objects in frames. The algorithm learns a deep association network from the COCO dataset (The Microsoft Common Objects in Context), which is large-scale object detection, segmentation, and captioning dataset [22].

After performing the preprocessing on each frame of the captured video and extracting the features for each object, we used the 1D-CNN model to discover the objects and classify them according to their types, which resulted in 91 classes. After determining the type of the object, it is surrounded by a box and labelled with a name, as shown in figure (5).

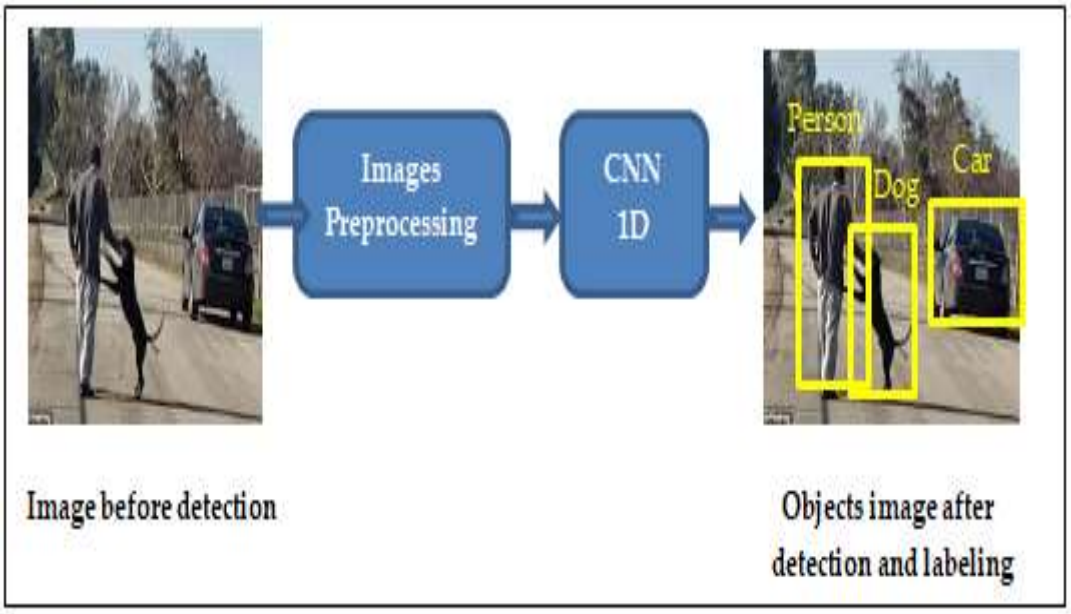

Figure 5- Objects image detection and labelling

3.3.3 Employment of the Developed Embedded System

The present section includes a detailed discussion of the employment of the developed vehicular system (programmable drone) that was built in the present work for finding and detecting the targets during its navigation for the purposes of security, as can be seen in figure(6). The maneuver first takes-off from the ground, as illustrated in figure (6a), and then starts searching for the target, as shown in figure (6b).

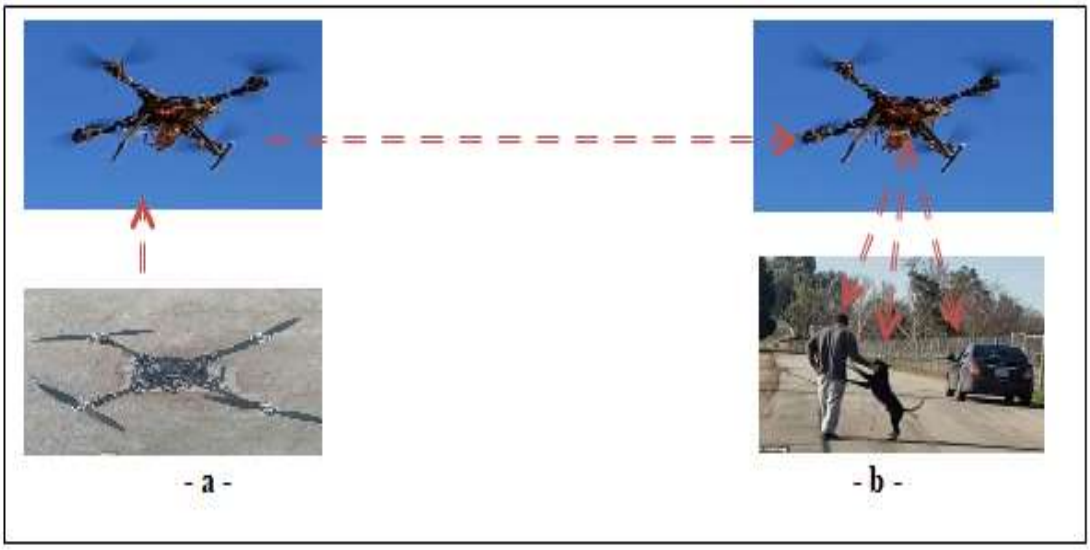

Figure 6-The operation of the aerial vehicle (drone):
a) The drone is taking-off from the ground, following a path waypoint.
b) The drone is looking for object's image.

To construct an on-board system, the Raspberry Pi was combined with the 1D-CNN model, as shown in figure (7).
This on-board Raspberry Pi4 system takes the aerial photos from the camera and performs data processing management.

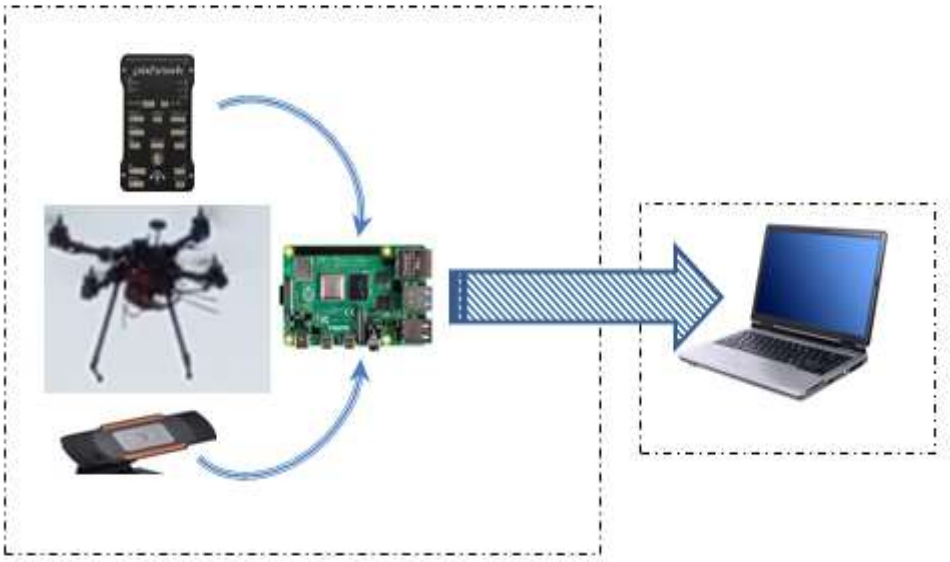

Figure 7- On-board target detection drone system using Raspberry Pi4

In this operation, Raspberry Pi4 is responsible for the processing of all algorithms of the 1D-CNN model and streaming data via the network.
The ground station which is connected to a similar network gains streaming data and shows results. From the output of the monitoring, the type of the detected target is obtained and notified, as shown in figure (8).

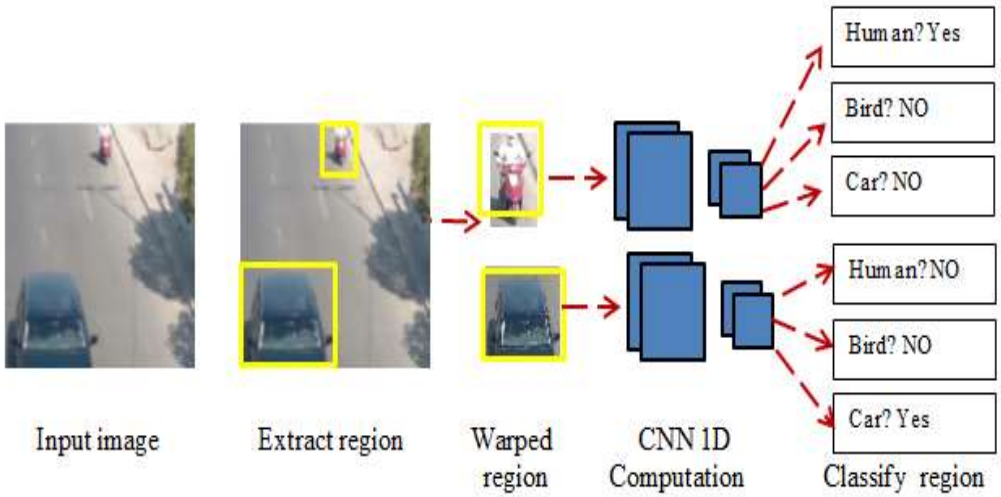

Figure 8- Employment steps of the classification stage based on CNN 1D model.

3.4. Objects localization

In this part, the drone is equipped with a mid-range camera pointed downwards with a 45 degrees positive inclination around the vertical axis. The next section is showing object localization steps.

3.4.1 Results of the Target Localization

Determining the objects image distance to the reference point (home) can be explained as follows:

1- The drone's starting point (home) is determined by finding values (latitude 1, longitude 1), through GPS readings.

2- When the drone flies to another point, both latitude 2 and longitude 2 are determined for the new location.

3- The drone's altitude (h) is calculated from GPS readings.

4- The drone computes the distance (d_1) (between the starting point and new location) using the Euclidean distance formula.

5- When the object's image is discovered and placed in the box, the center of mass will be calculated, as follows:

$$C_x C_y = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right) \text{----------(1)}$$

where:

$C_x C_y$: The coordinates of the center of mass.

(x_1, y_1): Top left point coordinates

(x_2, y_2): Lower right point coordinates

6- The angle of the camera deviation (θ) is calculated through a servo motion, which is the angle between the direction of the camera and the horizon, as shown in figure (9).

7- The distance between the current location of the drone and the location of the object (d_2) is calculated using the trigonometric function, as follows:

$$d_2 = h / \tan(\theta) \text{ ----------(2)}$$

8- The total distance (d) is calculated from the home point, as follows:

$$d = d_1 + d_2 \text{ ----------(3)}$$

Figure (9) illustrates the determination of the distance between the object's image and the home.

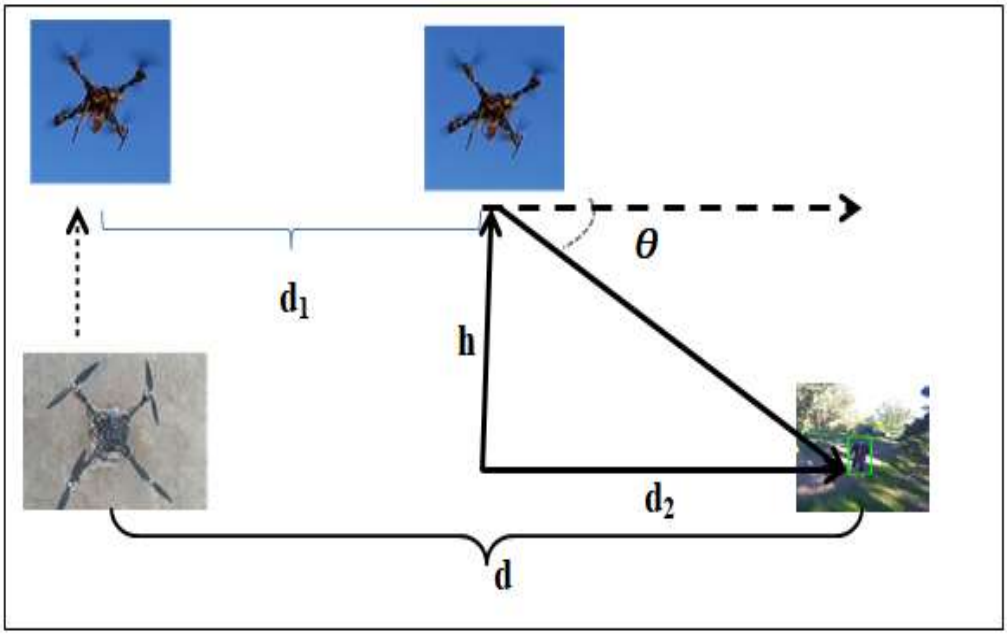

Figure 9-Determining the object's image distance to the home.

At the beginning of the practical experiments of using the drone in detecting moving objects and calculating the distance between the explored object and the drone take-off home site, a significant error was found in the calculations of distance (d_2), as shown in table (2) and can be also seen in figure (10).

Table 2-The object localization result

No.	Servo angle (θ)^o	Altitude (h) m	Distance from home to Drone (d_1)m	Distance from Drone to Object (d_2)m	Total practical distance (d)m	Total True distance (d)m
1	44	5.02	6.2576	5.1983	11.46	20
2	43	5.176	6.2817	5.5505	11.83	20
3	42	5.16	6.2431	5.7307	11.97	20
4	41	5.117	6.2096	5.8864	12.10	20
5	40	5.02	6.1365	5.9826	12.12	20
6	39	5.022	6.1834	6.2016	12.39	20
7	38	4.988	6.2327	6.3843	12.62	20
8	37	4.949	6.1538	6.5675	12.72	20
9	36	4.924	6.1416	6.7773	12.92	20
10	35	4.966	6.2076	7.0921	13.30	20
11	44	4.833	7.2930	5.004	12.30	20
12	43	4.91	7.3644	5.2653	12.63	20
13	42	4.987	7.2282	5.5386	12.77	20
14	40	5.036	7.2261	6.0016	13.23	20
15	38	4.967	7.1429	6.3574	13.50	20

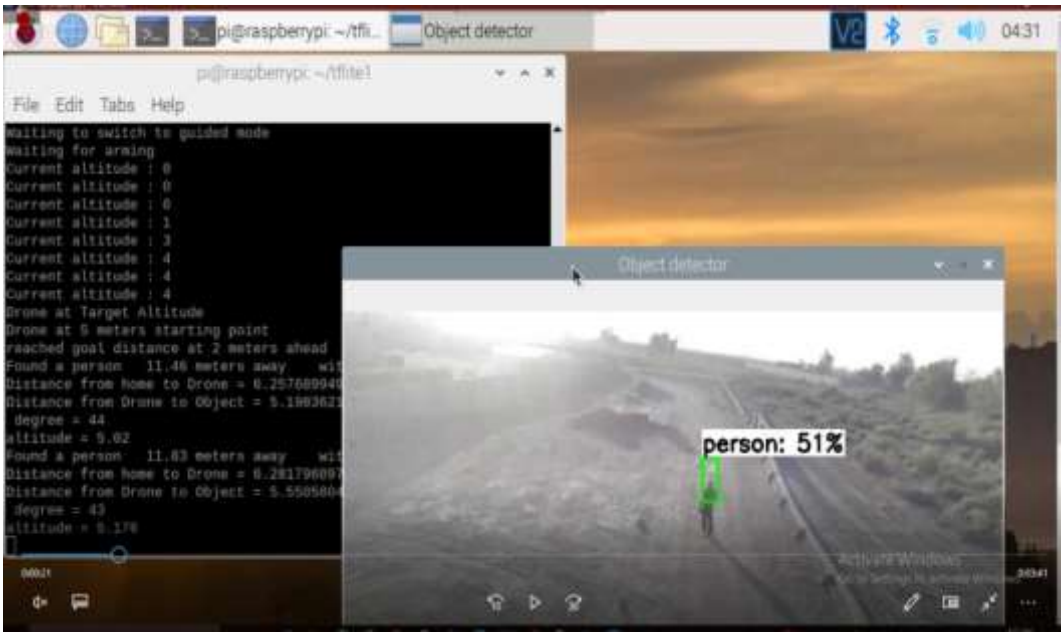

Figure 10- Detecting, labeling, and localization of the object.

Figure-11 shows the difference between true values and the practical values of the distance.

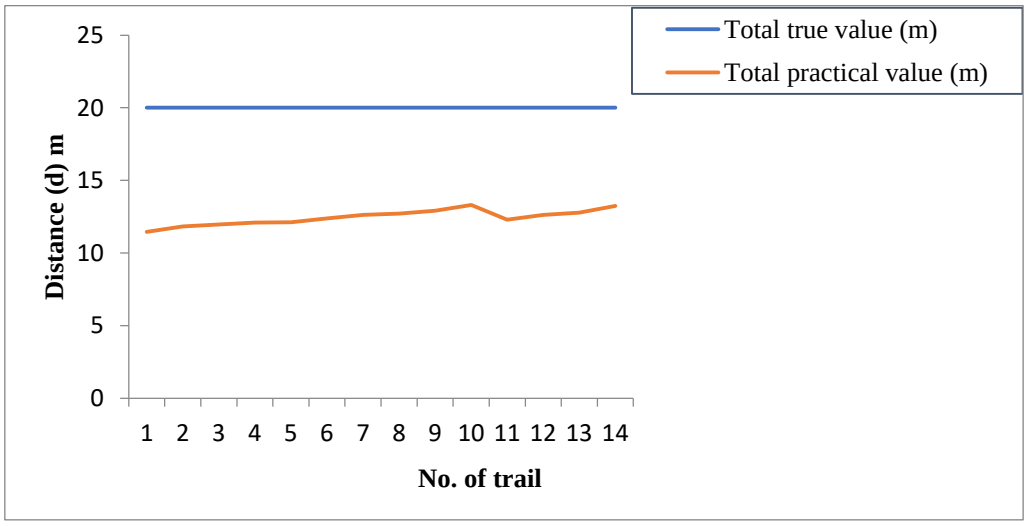

Figure 11-Distance as a function of no. of trails.

These errors were addressed by processing the servo angle and the direction of its motion, so that it follows the moving object and moves the camera in the right direction.
On the other hand, a correction was added where it was followed. As stated earlier, we took the center of the box around the body, which caused an amount of error in the measurements. Thus, it was corrected by taking the middle of the horizontal axis with the lowest point of the vertical axis, as follows:

$$C_x C_y = \left(\frac{x_1 + x_2}{2}, y_1\right) \text{----------(4)}$$

Then, the localization calculation can be divided into three cases:
First case: the movement of the drone towards the target while the target is stationary
Table (3) contains the corrected calculations of the object localization for the first case, as can be seen in figure (12).

Table 3-Corrected localization measurements

No.	Servo angle (θ)^o	Altitude (h) m	Distance from home to Drone (d_1)m	Distance from Drone to Object (d_2)m	Total practical distance (d)m	Total True distance (d)m
1	44	4.849	13.0655	5.0212	18.09	20
2	43	4.935	13.2951	5.2921	18.59	20
3	42	4.981	13.3358	5.5319	18.87	20
4	40	4.922	13.2781	5.8658	19.14	20
5	39	4.879	13.2712	6.0250	19.30	20
6	38	4.837	13.2854	6.1910	19.48	20
7	36	4.738	13.4530	6.5983	20.05	20
8	34	4.785	16.9853	7.0940	24.08	20
9	26	4.777	11.6031	9.7943	21.40	20
10	25	4.743	9.72623	10.1713	19.90	20
11	24	4.803	9.7176	10.7877	20.51	20
12	22	4.817	10.849	10.819	21.66	20

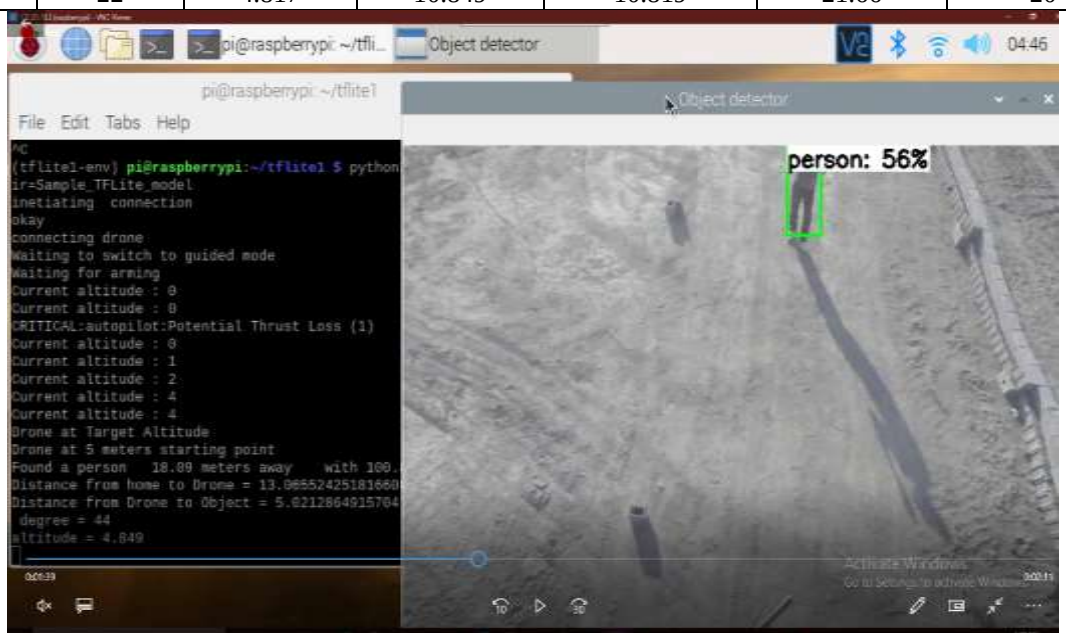

Figure 12- Detecting, labeling, and correcteing the localization measurements

While figure (13) illustrates the results of the analysis of these corrections, which show a quantitative enhancement

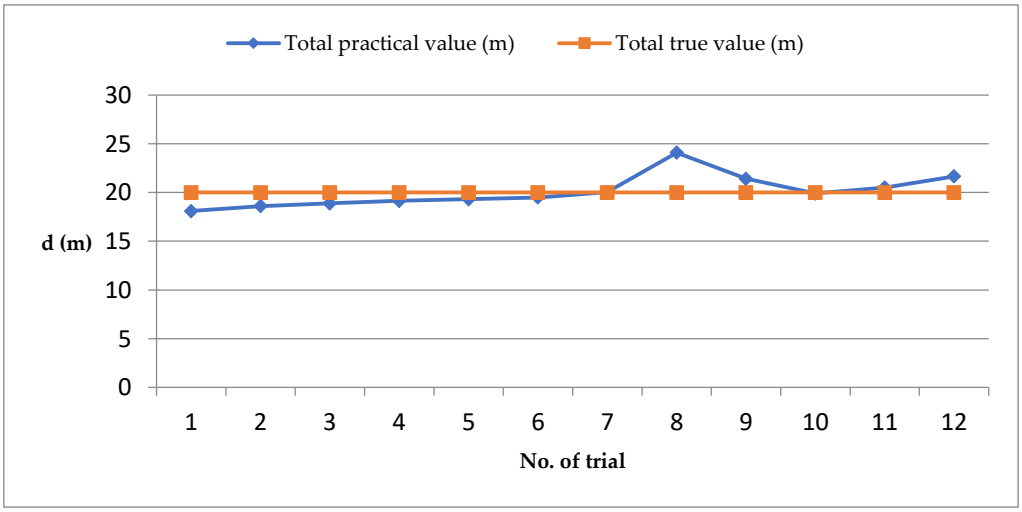

Figure 13- Correction results of the total distance d (m) as a function of no. of trails.

Figure (14) presents the relationship between the distance (d_2) as a function of the tilt angle (θ) of the camera for the table (3).

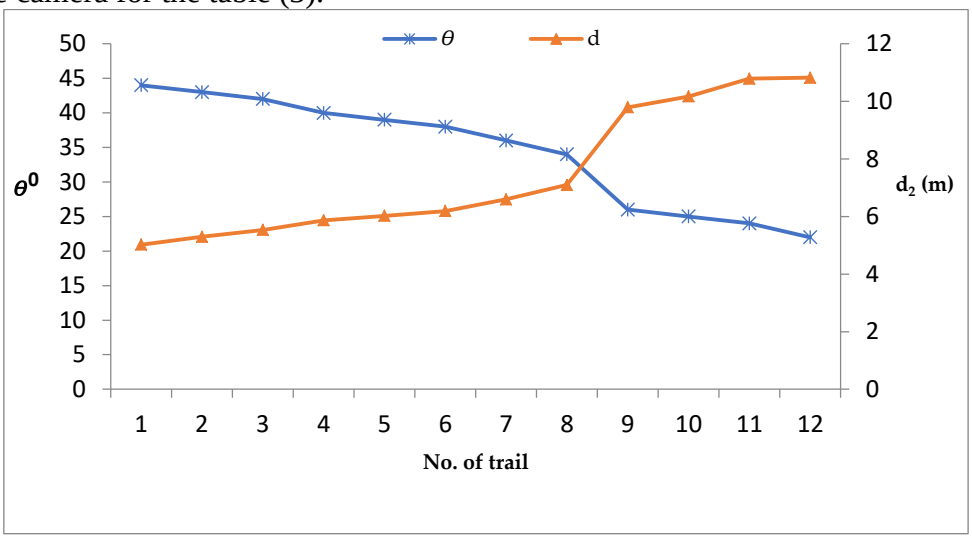

Figure 14- Relationship between tilt angle (θ) and distance (d_2) as a function of no. of trails.

The relations and the behavior of the curves in figure (14) conform with eq.(2).

Second case: Measurements were made when the drone was stationary and the detected object was moved. The results are shown in table .

Table 4-Localization measurements for the second case

No.	Servo angle (θ)^o	Altitude (h) m	Distance from home to Drone (d_1)m	Distance from Drone to Object (d_2)m	Total practical distance (d)m	Total True distance (d)m
1	44	4.806	4..9775	5.6600	10.64	11
2	43	4.822	5.1709	5.6583	10.83	11
3	42	4.771	5.56854	5.2987	10.87	10
4	41	4.806	5.5082	5.5286	11.04	10
5	39	4.806	5.5951	5.9349	11.53	11
6	37	4.755	5.6437	6.3100	11.95	11
7	37	4.776	5.6196	6.3379	11.96	10
8	37	4.807	5.6538	6.3791	12.03	11
9	37	4.754	5.6629	6.3087	11.97	12
10	36	4.755	5.6211	6.5446	12.17	13
11	31	4.732	5.6029	7.8753	13.48	15
12	30	4.794	5.6226	8.3034	13.93	15
13	29	4.84	5.6938	8.7315	14.43	15
14	28	4.778	5.6938	8.9861	14.68	16
15	24	4.814	5.7370	10.8124	16.55	17
16	26	4.859	5.6875	9.9624	15.65	15
17	26	4.808	5.6552	9.8578	15.51	14
18	25	4.803	5.6706	10.3000	15.97	15.5
19	24	4.814	5.7370	10.8124	16.55	16
20	27	4.854	5.7927	9.5265	15.32	14

Figure (15) illustrates the results of the analysis for the second case.

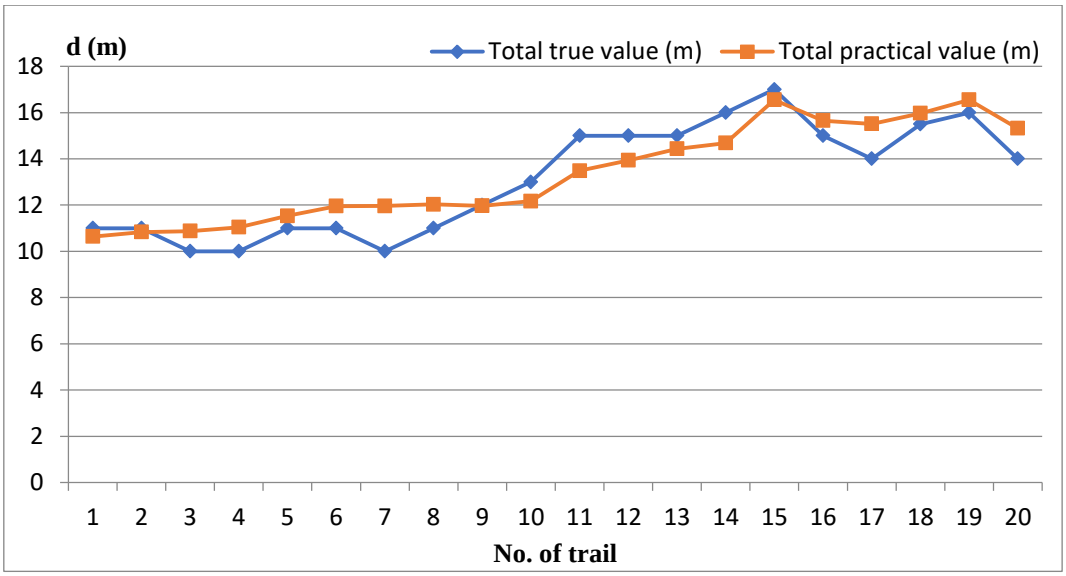

Figure 15-Correction results of the total distance d (m) as a function of no. of trails for the second case.

Based on the calculations made by using equation (2), the results showed that the servo angle decreases, while the distance between the drone and the body increases, which matches the behavior shown in figure (16).

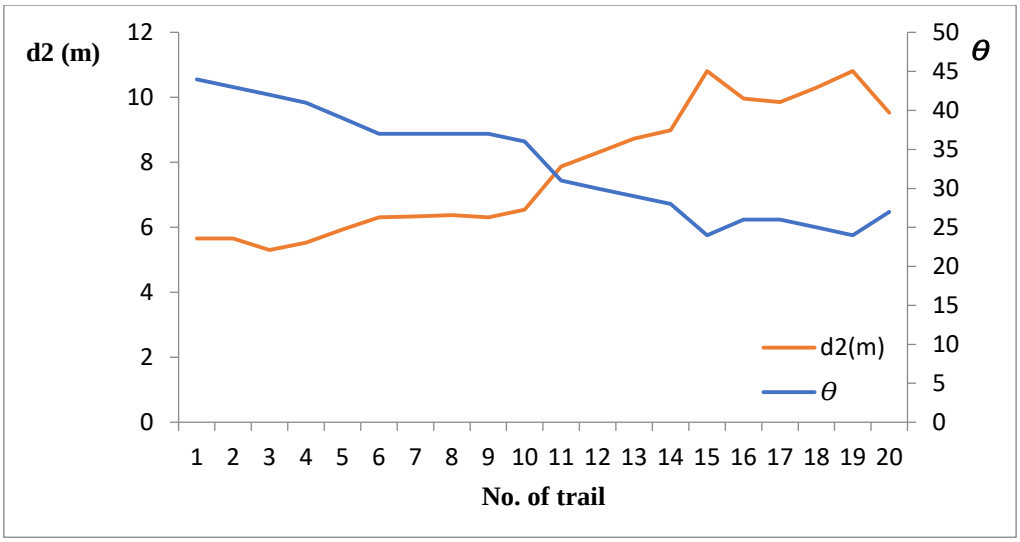

Figure 16-Relationship between distance (d_2) and tilt angle (θ) as a function of no. of trails for the second case

Third case: The results were calculated when the detected object was stationary and the drone moved up, as shown in table (5).

Table 5-Localization measurements for the third case

No.	Servo angle (θ)^o	Altitude (h) m	Distance from home to drone (d_1)m	Distance from drone to object (d_2)m	Total practical distance (d)m	Total true distance (d)m
1	19	3.04	5.51	8.8288	14.3388	15
2	20	3.11	5.47	8.5446	14.0146	15
3	23	4.21	5.687	9.9181	15.6051	15

4	21	3.988	5.12	10.389	15.509	15
5	28	5.03	5.021	9.460	14.481	15
6	28	5.117	4.959	9.6236	14.5826	15
7	30	5.924	5.507	10.2606	15.7676	15
8	33	5.988	5.11	9.2207	14.3307	15
9	35	7.05	5.37	10.0684	14.4384	15
10	39	7.12	5.27	8.7924	14.624	15
11	42	8.03	5.01	8.9182	13.9282	15
12	44	8.10	5.47	8.3877	13.8577	15

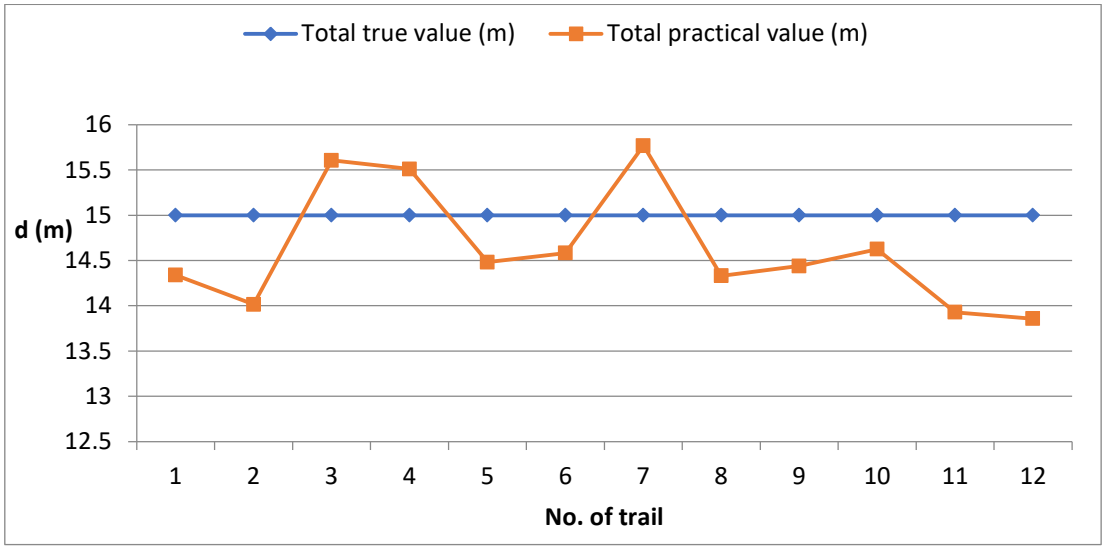

Figure 17-The distance (d) as a function of no. of trails for the third case.

From figure (18), we can notice that when the altitude incresses, the tilt angle of the camera increases signficantly**.**

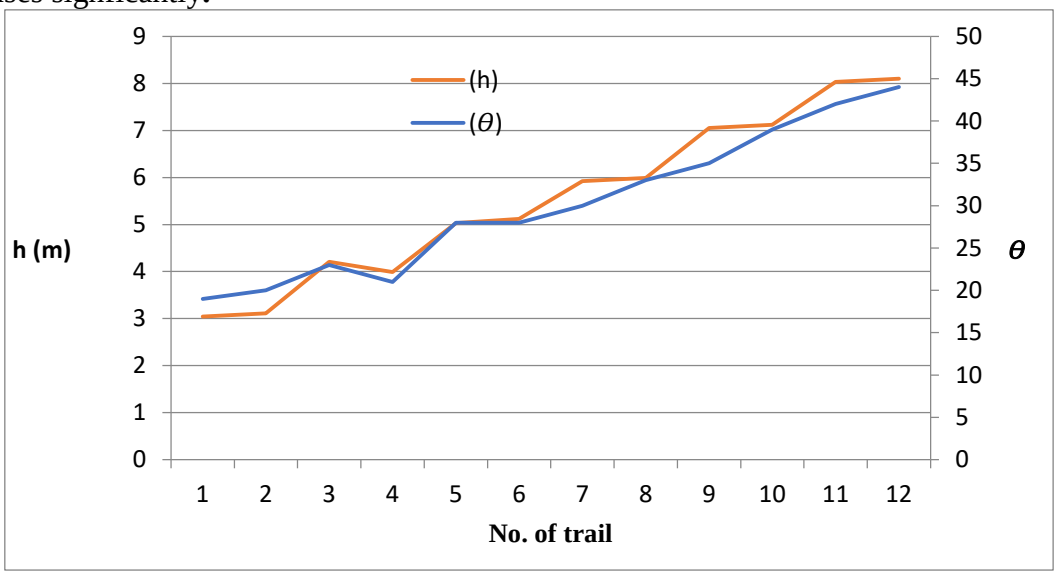

Figure-18-Relationship between the tilt angle (θ) as a function of no. of trails for the third case.

The determination of the direction of the object was achieved by obtaining the position and angle of the drone by using a compass (angle value for north and south geographic) and subtracting the value from the original value of the drone's position before taking-off, as shown in figure (19). When the object is detected, it is located at the same angle with the

drone. By knowing the first direction that represents the location of the drone (home), it is possible to determine the direction of the object with high accuracy, using the compass within a gyroscopic element.

Table 6-The object direction

	Direction of the object to left (Degree)	Direction of the home (Degree)	Direction of the object to right (Degree)
1	95.53	100.91	103.32
2	90.55	100.91	105.04
3	87.31	100.91	110.90
4	80.55	100.91	126.04
5	74.80	100.91	130.53
6	71.88	100.91	146.04

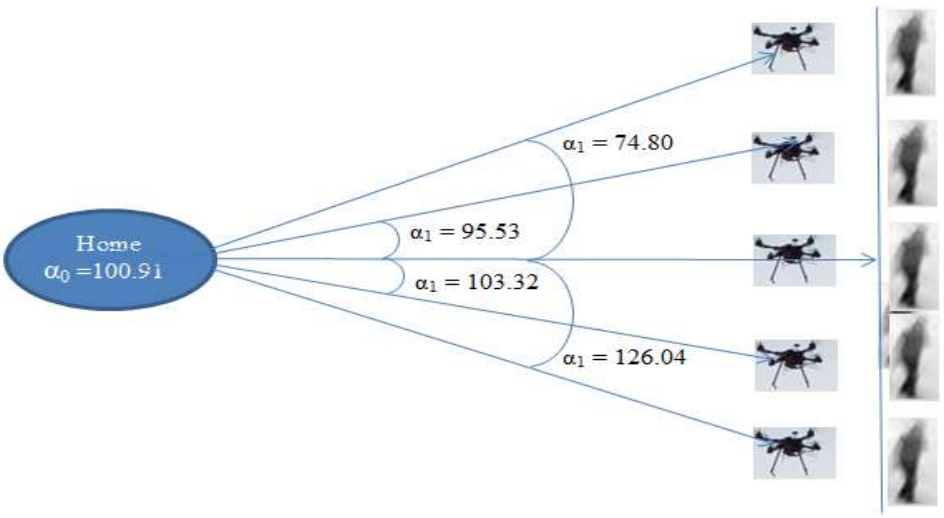

Figure 19-The object direction

$$\alpha = \alpha_0 - \alpha_1 \quad \text{----------(5)}$$

where:

α : Angle of the discovered object

α_0 : Angle of the home

α_1 : Drone angle when the object is detected.

From equation (5), the positive angle value means that the object moves to the left of the home, but if it is negative, it moves to the right of the home.

4. Conclusions

1- The aim of this work was to develop the ability of drones to detect, label, and localize (in real-time) objects from video images captured. Several conclusions can be summarized from this work. The drone was successfully designed and suitable components were selected, which are available on the local market. The drone was programmed and connected to the computer through Raspberry Pi4 and using communication protocols. Pixhawk controller was used directly on Raspberry Pi4. Good results were obtained after improvement to determine the distance and location of the object by knowing the distance and direction and by correcting the calculations of the distance.

5. References

[1] D. Sincha, M. Chervonenkis, and P. Skribtsov, "Vehicle detection in aerial traffic monitoring," *Am. J. Appl. Sci.*, vol. 13, no. 1, pp. 46–54, 2016, doi: 10.3844/ajassp.2016.46.54.

[2] P. Ramon, B. C. Arrue, J. J. Acevedo, and A. Ollero, "Visual Surveillance System with Multi-UAVs Under Communication Constrains," *Adv. Intell. Syst. Comput.*, vol. 417, pp. 705–713, 2016, doi: 10.1007/978-3-319-27146-0_54.

[3] D. Bein, W. Bein, A. Karki, and B. B. Madan, "Optimizing Border Patrol Operations Using Unmanned Aerial Vehicles," *Proc. - 12th Int. Conf. Inf. Technol. New Gener. ITNG 2015*, no. June, pp. 479–484, 2015, doi: 10.1109/ITNG.2015.83.

[4] H. A. Kurdi and J. P. How, "Bio-inspired algorithm for task allocation in multi-UAV search and rescue missions," *2016 AIAA Guid. Navig. Control Conf.*, 2016, doi: 10.2514/6.2016-1377.

[5] S. Yadav, M. Sharma, and A. Borad, "Thrust efficiency of drones (quadcopter) with different propellers and and their payload capacities," *Int. J. Aerosp. Mech. Eng.*, vol. 4, no. 2, pp. 18–23, 2017, [Online]. Available: http://www.ijamejournals.com/pdf/rpm11174.pdf.

[6] R. Pi, "Raspberry Pi 4 Computer Model B," *Raspberrypi.Org*, no. May, pp. 1–6, 2020, [Online]. Available: www.raspberrypi.org.

[7] "Messages (common) · MAVLink Developer Guide." https://mavlink.io/en/messages/common.html (accessed Oct. 29, 2021).

[8] "DroneKit." https://dronekit.io/ (accessed Oct. 29, 2021).

[9] "ArduPilot Documentation — ArduPilot documentation." https://ardupilot.org/ardupilot/ (accessed Oct. 29, 2021).

[10] S. M. Al-Qaraawi, A. D. Salman, and M. S. Ibrahim, "Performance evaluation of ad-hoc based aerial monitoring system," vol. 7, no. 4, pp. 1794–1800, 2019.

[11] F. Kamran, M. Shahzad, and F. Shafait, "Automated Military Vehicle Detection From Low-Altitude Aerial Images."

[12] J. Uus and T. Krilavičius, "Detection of different types of vehicles from aerial imagery," *CEUR Workshop Proc.*, vol. 2470, pp. 80–85, 2019.

[13] "ZD550 Pro 550mm 4 Axis Carbon Fiber Quadcopter Frame Umbrella Folding Drone | eBay." https://www.ebay.com/itm/284216755351 (accessed Oct. 29, 2021).

[14] "5010 360Kv High Efficiency Brushless Motor – Unmanned Tech UK FPV Shop." https://www.unmannedtechshop.co.uk/product/5010-360kv-high-efficiency-brushless-motor/ (accessed Oct. 29, 2021).

[15] "Hobbywing XRotor 40A ESC(no BEC)." https://www.foxtechfpv.com/hobbywing-xrotor-40a-escno-bec-p-1706.html (accessed Oct. 29, 2021).

[16] "FlySky FS-i6S 10CH Transmitter & FS-X6B 2.4GHz PPM Receiver | Flying Tech." https://www.flyingtech.co.uk/electronics/flysky-fs-i6s-10ch-transmitter-fs-x6b-24ghz-ppm-receiver (accessed Oct. 29, 2021).

[17] "3DR Pixhawk 1 Flight Controller (Discontinued) | PX4 User Guide." https://docs.px4.io/master/en/flight_controller/pixhawk.html (accessed Oct. 29, 2021).

[18] R. R. Muhammed and A. S. Mahdi, "Accurate three dimensional coordinates measurements using differential GPS real time kinematic mode," *Iraqi J. Sci.*, vol. 59, no. 2, pp. 1146–1151, 2018, doi: 10.24996/IJS.2018.59.2C.20.

[19] J. Malliss, V. Patel, S. Yadav, H. Varun, and S. Patnaik, "Real Time Communication (RTC) Device using Raspberry Pi," vol. 9, no. 2, pp. 95–99, 2018.

[20] "Download VNC Server | VNC® Connect." https://www.realvnc.com/en/connect/download/vnc/ (accessed Nov. 13, 2021).

[21] A. A. A. Karim and R. A. Sameer, "Image feature extraction and selection," *Iraqi J. Sci.*, vol. 59, no. 3, pp. 1501–1508, 2018, doi: 10.24996/IJS.2018.59.3B.16.

[22] "COCO - Common Objects in Context." https://cocodataset.org/#download (accessed Oct. 29, 2021).