ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 408
Innovative System Design and Development II

PROJECT ID: *202119*
Autonomous Drone Control

Ahmet Çetin TÜRKYENER
201711066

Songül Meryem ÖZBİLEN
201711048

Elif Yağmur ERATALAY
201811028

Advisor: *Ahmet COŞAR*

Contents

Software Requirements Specification

1.ABSTRACT

In recent years, drone technology shows itself in many areas. For example, it is no longer impossible to see drones flying in the air in the world of journalism and show, cargo distribution, agriculture, and emergency situations. Drones are unmanned aerial vehicles. It consists of components such as propeller, engine, body and flight control board. We will also automate the drone by using artificial intelligence and image processing technologies. Thus, it will be able to detect previously defined obstacles and reach the target. At the same time, we will ensure that when a break occurs, the situation is resolved promptly with the least cost, security issues are kept to a minimum, and the ability to react quickly to obstacles. In this literature review, we mention about the projects that have been done before on this subject, how we will do our own project, the problems that we may encounter and their solutions. Keywords: Drone, Artificial Intelligence and Image Processing

2.INTRODUCTION

Drones are known as Unmanned Aerial Vehicles(UAV) with the name used in the technological field [1]. Drones are robots that can be controlled by remote control or fly automatically under the control of various software added to their embedded systems. Drones, which were used only to provide security in the military field in the past, are now used in many various and useful areas such as transportation, camera shots and fire extinguishing. Drones can perform tasks assigned to them, such as taking off, landing, and flying from one place to another, through a remote controller. These controllers can communicate with the drone using radio waves [2].

The aim of this study is to develop obstacle recognition skills for unmanned aerial vehicles by using Image Processing and Artificial Intelligence methods and in this direction, enabling drones to find their position by detecting the objects around them on autopilot.

In our Literature Review Report, we have Work Part containing Drone Preparation, Image Processing with Deep Neural Network Model and Ensuring that the software created with Image Processing and Deep Artificial Neural Network Model communicates with Pixhawk to automatically fly from desired locations. It also includes some of the problems that we may encounter during the project and their solutions. Finally, there is the conclusion part, where we evaluate the results of all our studies.

3.WORK PART

3.1. Drone Preparation

It is the preparation phase of the test drone. Based on our research, we chose a drone that will be multicopter style [3]. In order to carry out the tests easily, to repair the drone at an affordable cost in the massacres, to make it ready for flight again, and to reduce the security problems, it is considered to build a small, 20-30 cm diameter, quadcopter-style drone with 5-7 inch propellers. We are planning to build a quadcopter-style drone [4], which is the system that gives commands to the motor that provide the balance of the flying device by processing the information coming from the gyro and these sensors through the software running on its processor. It is planned that the engines will be selected strong and the payload of the drone up to 500 grams. It is thought to have a mini camera on the drone, a mini computer for image processing and artificial intelligence. The minicomputer is planned to be Jetson Xavier [5], which will benefit us in artificial intelligence coding and drone automation. According to the results of our research, it is planned to use open source code pixhawk autopilot as a 10 3A regulator autopilot on the drone to feed this computer and camera. This 10 3A regulator is used for all kinds of open source vehicles (multicopter, helicopter airplane, etc.), all kinds of multicopter types (tricopter, quadcopter, hexacopter, etc.) [6]. When a problem is encountered in the next tests, it is thought to have the option of turning off the engines with a remote command for safety reasons. The communication between Jetson and Pixhawk autopilot is planned to be done with UART communication ports, which is a communication protocol that provides communication between the peripheral units of the computers and microcontrollers on the Jetson and Pixhawk. It is considered to use 115,200 baud rate as the UART communication baud rate [7]. It is planned to create the structure of the UART communication system by having 8 bits of data as the UART communication protocol, selecting the parity bit, selecting 1 stop bit and selecting the timeout values and applying these values to both Jetson Xavier and Pixhawk autopilot. It is considered to use Linux Ubuntu operating system on Jetson Xavier as the operating system. The Pixhawk autopilot can communicate with standard 50 hertz motor drivers, but it is planned to increase motor driver communication to 400 hertz to make the drone respond faster. The images taken from the camera will be processed by Jetson using image processing and artificial intelligence and the guidance information obtained from this processing will be quickly transferred to the Pixhawk autopilot via the UART, and the Pixhawk autopilot will be able to fly with automatic responses at 400 hertz. The image processing model with the artificial neural network on our Jetson board is explained in the next phase.

3.2. Image Processing with Deep Neural Network Model

According to our research, first, a data set will be created with the images of the objects to be detected by the drone. This dataset creation process will be done with photographs of objects taken from different angles. The created dataset will not be a complete training set to be tested later, 80% of this dataset will be reserved for training and 20% for testing [8]. According to the results of our research, we planned to use CNN (Convolutional Neural Networks) artificial neural network on datasets for training. As an algorithm, we planned to use the YOLO (You Only Look Once) [9] algorithm. New training datasets will be processed with the YOLO algorithm using CNN. **YOLO (You Only Look Once):** It is an image processing algorithm using CNN (Convolutional Neural Networks). There are several reasons why we chose the YOLO algorithm. These are speed, accuracy and learning capacity for object detection and object recognition. YOLO [10] is generally superior to other algorithms in these aspects, so we plan to use this algorithm. The importance of these issues is outlined below.

Speed: In systems that will work in real time, objects must be detected very quickly. The drone is an agile platform that can move very fast, so speed is one of the most important issues for us. YOLO, which we will use as an algorithm, is 4-5 times faster than other object detection algorithms (such as RetinaNet-101, RetinaNet-50).

Accuracy: Accuracy is important as we are moving quickly with the drone live and YOLO can detect objects with very low error rate. After all, it is an algorithm that can detect objects both quickly and with high accuracy.

Learning capacity: The YOLO algorithm [11] is compatible with datasets expansion. Some other algorithms may suffer from poor prediction performance (especially in terms of accuracy) when there are many datasets. In the YOLO algorithm, a large number of objects can be defined quickly and accurately with extensible data sets.

YOLO's high object detection speed and prediction performance in the fast movements of the drone and its expandable learning capacity made us prefer this algorithm [12]. Our model, which will be formed as a result of our algorithm that will work with CNN on YOLO, will be run with our test data set, and our accuracy rate will be revealed according to the results obtained[13].

We will also include images of different objects in the test data, and in this way, the error rate of the trained model against incorrect images will show how the model will behave against different objects. Our accuracy rate will be determined as a result of the tests we perform with the test data.

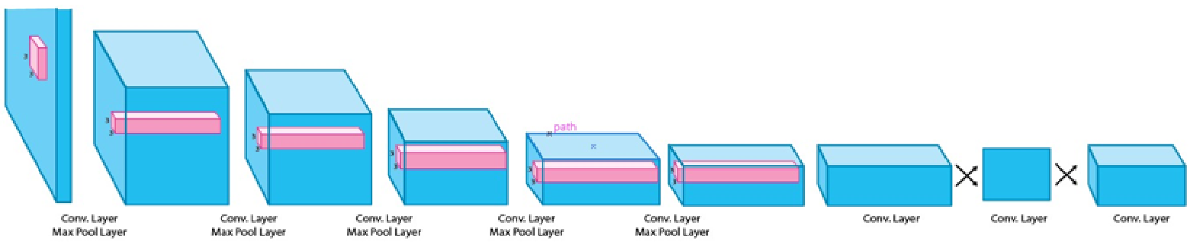

Figure 1: YOLO Architecture

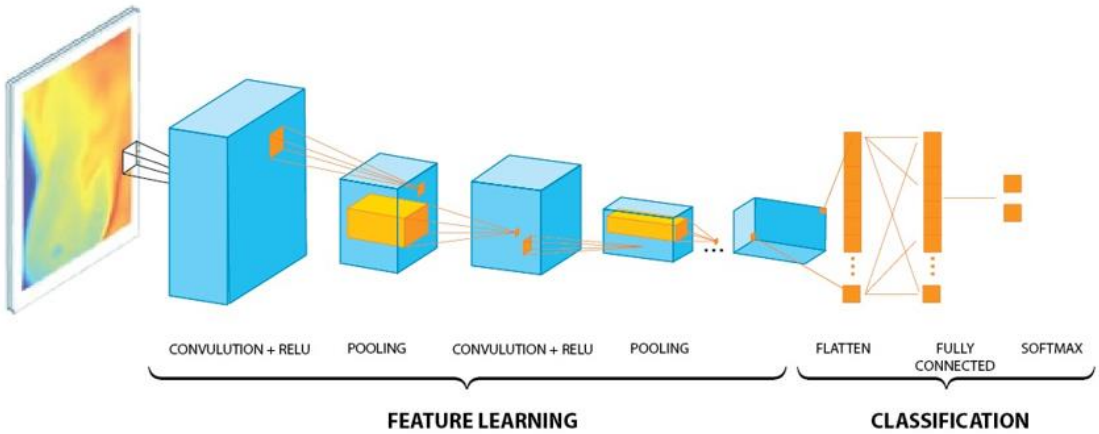

Figure 2: CNN Architecture

CNN (Convolutional Neural Networks)

First used in its modern form in 1990, CNN is a class of artificial neural networks most commonly used for image analysis in deep learning. Because CNN uses ReLU as activation, there can be a large number of data in the nerve and learning can take place without burden. CNN consists of basic foundation. These are Convolutional Layer - Pooling Layer - Fully Connected (FC) Layer.

Convolutional Layer

Our images in the training data have dimensions W x H x 3. W: width H: height 3: RGB (Red Green Blue) (Red Green Blue) value. W x H in the images represents a matrix. Each cell of this matrix represents a pixel. This pixel has an RGB value. This value is between 0-255. An N x N filter is determined according to the matrices of the images. The feature map of the image is created by sliding this filter over the image according to the Stride value (slip variable). This process is repeated many times with different filters. In this way, according to the attribute map, the object's color, corner, protrusion, etc. information is extracted.

ReLU

ReLU (Rectified Linear Unit) is a nonlinear function that operates as f(x) = max(0,x) [15]. So, if the relu function takes – it will output 0, if it takes + it will output that value itself. ReLU, whose main purpose is to get rid of negative values, has a very important position in CNNs. Nonlinear functions such as ReLU, tanh and sigmoid are used to prevent our model from learning negative values or not being able to grasp some features due to these negative values [14].

Max Pooling

Max Pooling is done to reduce the size of images with high dimensions without losing their properties [15]. Max Pooling is done by sliding over our W x H sized image with an N x N matrix. During this process, the highest value within a region of the size of our N x N max pooling matrix is determined as the new pixel value and other values are not used. Using a value directly belonging to the image without deriving a new value also prevents the image from being corrupted. Feature extraction takes place thanks to these processes.

Fully Connected Layer

Our matrix, which reaches up to the Fully Connected layer, is turned into rows by the Flattening process [2]. Values in rows are considered as Input Layer. Our neural network model has 3 stages as Input, Hidden and Output Layers. (Input Layer – Hidden Layer – Output Layer) Input Layer takes the incoming data in rows as input and transfers it to the hidden layers. Input layer node value * hidden layer node value gives the number of connections. The values coming to the Hidden Layer are connected to the Output Layer according to the output state we want. We will have outputs according to the number of objects we have determined in our project, plus the absence of any of these objects. We will have as many connections as hidden layer node value * output layer node value. The values we will end up with will be the object name or the absence of the object. As soon as the object is detected, our drone will react accordingly.

3.3. Ensuring that the software created with Image Processing and Deep Artificial Neural Network Model communicates with Pixhawk to automatically fly from desired locations

We are planning to develop the image processing and artificial neural network application, which we will develop based on the research we have done, in a structure that can process the image coming from the camera at 20 fps and make decisions at 20 hertz accordingly. In this way, it is aimed to process the frames as soon as they come from the camera and transfer the information accordingly. If the processor were on the ground, there would be several hundred milliseconds of delay due to RF (radio frequency) transmit and receive delays. But since the processor is on the drone, communication will take place via Jetson Xavier to UART, and from UART to autopilot at a baud rate of 115,200 via cables without RF (radio frequency) delay, so the drone will be able to react as quickly as possible. Considering that the drone can reach speeds of over 60 km/h, this minimum delay is very important in order to avoid problems. The system will make decisions with an accuracy of over 90 percent in each frame, and when the second frame arrives, it will work with 90 percent plus the remaining ninety percent of the ten percent, that is, 99 percent in total and with 3rd and 4th frames, this accuracy rate will increase much more [16]. We will test the system with many flight trials and we will be able to observe the accuracy rate. We will continue to work by updating the image processing and artificial neural network algorithms in case of potential problems and accelerating them if necessary [17].

4.CONCLUSION

As a result, the aim of this study is to develop possible obstacle models for drones by using Image Processing and Artificial Intelligence techniques and to enable the drones to pass these obstacles with autopilot and reach the target location. According to the results of our research, we will use CNN, Image Processing computer programming sub-branches, PyCharm IDE, LabelIMG, Cloud GPU programs and YOLO reference algorithm method. We will use TensorFlow, Keras, OpenCV, NumPy libraries for this project. A low confidence score in object detection is seen as a potential problem. A possible solution for the related problem is to expand the dataset and add different angles to the visuals.

5.REFERENCES

[1] About UAVs [Online]. Available:

https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

[Accessed 02/11/2021]

[2] Study about Drone [Online]. Available:

https://internetofthingsagenda.techtarget.com/definition/drone

[Accessed 02/11/2021]

[3] Study about “Attitude Control of Multicopter” [Online]. Available:

https://core.ac.uk/download/pdf/295548558.pdf

[Accessed 04/11/2021]

[4] Study about “Building a Quadcopter with Arduino” [Online]. Available:

https://abis.bozok.edu.tr/indir.php?file_id=710

[Accessed 06/11/2021]

[5] Jetson Xavier [Online]. Available:

https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit#:~:text=The%20NVIDIA%20Jetson%20AGX%20Xavier%20Developer%20Kit%20is%20the%20latest,drones%20and%20other%20autonomous%20machines

[Accessed 06/11/2021]

[6] About multicopters [Online]. Available:

https://www.dronedoktoru.com/multikopter-nedir.html

[Accessed 06/11/2021]

[7] How UART works [Online]. Available:

https://herenkeskin.com/uart-nedir-ve-nasil-calisir/

[Accessed 07/11/2021]

[8] Shobhit Bhatnagar, "I Classification of Fashion Article Images using Convolutional Neural Networks," 2017, Fourth International Conference on Image Information Processing (ICIIP) [Online]. Available:

https://ieeexplore.ieee.org/document/8313740

[Accessed 08/11/2021]

[9] Xia Zhao, "A novel three-dimensional object detection with the modified You Only Look Once method," 2018, International Journal of Advanced Robotic Systems [Online]. Available:

https://journals.sagepub.com/doi/full/10.1177/1729881418765507

[Accessed 08/11/2021]

[10] Daniel Pestana, "A Full Featured Configurable Accelerator for Object Detection with YOLO, " 2021, IEEE [Online]. Available:

https://ieeexplore.ieee.org/document/9435338

[Accessed 09/11/2021]

[11] About YOLO algorithm [Online]. Available:

https://medium.com/deep-learning-turkiye/yolo-algoritmas%C4%B1n%C4%B1-anlamak-290f2152808f

[Accessed 09/11/2021]

[12] About YOLO algorithm [Online]. Available:

https://www.youtube.com/watch?v=vRqSO6RsptU

[Accessed 10/11/2021]

[13] You Only Look Once: Unified, Real-Time Object Detection [Online]. Available:

https://arxiv.org/pdf/1506.02640.pdf

[Accessed 10/11/2021]

[14] About Residual blocks — Building blocks of ResNet [Online]. Available:

https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec

[Accessed 11/11/2021]

[15] Convolutional Neural Network (CNN) [Online]. Available:

https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad

[Accessed 11/11/2021]

[16] Smart Autopilot Drone System for Surface Surveillance and Anomaly Detection via Customizable Deep Neural Network [Online]. Available:

https://www.researchgate.net/publication/338529296_Smart_Autopilot_Drone_System_for_Surface_Surveillance_and_Anomaly_Detection_via_Customizable_Deep_Neural_Network

[Accessed 12/11/2021]

[17] Detection of a Moving UAV Based on Deep Learning-Based Distance Estimation [Online]. Available:

https://www.mdpi.com/2072-4292/12/18/3035/htm

[Accessed 13/11/2021]

Software Requirements Specification

6.INTRODUCTION

6.1 Goal

The goal of this project is to provide automatic flight-path control according to objects by implementing object recognition algorithms on the drone. Computational sense-making of the objects seen with the unmanned aerial vehicle camera with image processing and artificial intelligence applications, processing this information and the information from the drone autopilot with a processor, creating new movement commands of the drone and transferring these commands to the autopilot are intended, which allows the drone to automatically move according to them without hitting the objects on the path.

6.2 Project Scope

Drones have become popular in recent years and have started to be used in almost every field that will benefit humanity in this regard. People from all over the world, students, academics, private and public sectors continue to launch new projects and new products related to drones. Despite being one of the fastest-growing areas of technology, drone technology has its own shortcomings. One of the most important of these is safe flight with object recognition. Drones cause the most accidents by crashing into objects while moving. They can crash into electricity poles, trees, hilly and mountainous places, buildings, and even more people, cars, etc. In addition, the delay in the use of drone technology near the ground (control patrols in factories, controls between buildings, close-up automatic drone commercials, etc.) is due to the inability of drones to fly safely in these areas due to crashes. Various technologies are being considered as solutions to these issues, and the implementation of object recognition algorithms on the drone that we are planning to do and the project of providing automatic path control according to objects is one of the most important steps that can be taken in this regard.

6.3 Advantages

With this project, drones will start to fly more safely and will get smarter. Accidents and crashes caused by drones hitting objects will begin to decrease, and human and animal injuries and deaths will begin to decrease. In this way, the use of drones in these areas will increase, which is already expensive and therefore has reservations about its use in areas that can experience decimation. In addition, with drones getting smarter in the field of object vision, new technologies will be paved in this regard. For example, it can pave the way for hundreds of future drone technology issues such as automatic tracking of moving objects, dropping cargo on certain objects in a certain place, drones discharged from the automatic charging station by precision landing at the automatic charging station with object vision and continuing to the next patrols as automatic charging. As a result, accident incidents will decrease, new drone technologies will be paved, and its development and use will increase like any technology that has become safer.

6.4 Glossary

CNN: Convolutional Neural Networks

YOLO: You Only Look Once

ReLU: Rectified Linear Unit

GCU: Ground Control Unit [UAV ground control station (UAV – Unmanned aerial vehicles)]

6.5 Documental Overview

This document generally describes the safety of drones on objects on the path, the advantages to be gained if this security is provided, and how the technology will develop in this way. It gives information about autopilot software, flight hardware, communication hardware, mini-computer hardware and informs about image processing and artificial intelligence algorithms to be used on object vision, and the advantages of these algorithms over other algorithms in general.

7. GENERAL DESCRIPTION

7.1 Product Perspective (Product Features)

The project of Implementation of Object Recognition Algorithms on Drone and Providing Automatic Path Control According to Objects consists of hardware and software parts. As hardware, it basically consists of drone, drone autopilot (open-source autopilot), processor (jetson board) and camera. As software, autopilot software (open-source autopilot software), communication software, and image processing and artificial intelligence algorithm software that will work on the jetson card are combined and consist of drone control software, which is the decision-making mechanism.

7.2 Drone Preparation

We will use a small multicopter as a drone. The main reason for this is that drones are dangerous and expensive for testing. A multicopter with 10 inches and six propellers will do. We will build the drone completely ourselves. We will choose the engine, esc, propeller part, which is the thrust block, so that they are compatible with each other. As autopilot, we will choose an open-source autopilot, the autopilot we are considering is Pixhawk. We will have a telemetry for two-way communication. We will build the battery ourselves from lithium-ion cells. There will be a regulator that we can regulate 12V and 5V from the battery. We will design and manufacture the body with 3D programs according to the components we have chosen. On the front will be the camera zone. There will be a minicomputer place for image processing and artificial intelligence. Jetson Xavier was considered as a minicomputer.

7.3 Artificial Neural Network Model and Image Processing Methodology

A data set of objects that the drone can see will be created for image processing and artificial intelligence. This dataset creation process will be done with photographs of these objects taken from different angles. The drone needs to recognize these objects with great precision. For this, a large part (about 80%) of the photographic dataset will be used for training. The rest of the dataset will be used to test the correct functioning of the system. If necessary, we will enlarge this dataset until we reach the accuracy we wish.

Many systems are used for training and recognition algorithms. When we observed our system, we concluded that it should be a structure that will provide high accuracy solutions even when moving fast, especially since the drone moves fast.

As a result of our research, we chose the structure suitable for expanding the speed, accuracy and learning capacity when desired, Convolutional Neural Networks (CNN) artificial neural network on the training data sets and You Only Look Once (YOLO) algorithm as the algorithm. We decided to train our datasets with CNN and set up the algorithm with YOLO. We found that the YOLO algorithm is several times faster than other algorithms we researched, such as RetinaNet. Speed is very important for us in the algorithm, as the drone can accelerate to 60 km/h. The algorithm should not compromise on accuracy while performing such fast processing, YOLO can recognize objects with a very low error rate while performing fast processing. As we mentioned before, we will test our system with training datasets and expand our system with new datasets until we reach the desired performance. It gave us another reason to choose YOLO with extensible datasets and accurate prediction performance when multi-dataset.

CNN is widely used in deep learning to analyse images. CNN uses Rectified Linear Unit (ReLU) as its activation function. It is used for multi-layer deep learning models. Our visuals in the training data in the Convolutional Layer have dimensions W: width x H: height x 3: RGB (Red Green Blue). By performing MaxPooling, we can reduce the size of images with high dimensions without losing their properties. Thus, we will be able to make fast and accurate predictions. Our neural network model will have 3 stages (Input, hidden and output layer). We will have output as the number of objects we have determined from this model plus the number of cases where none of these objects are present.

7.4. Artificial Neural Network & Image Processing Algorithm Result - Drone Steering with Autopilot

Artificial neural network and image processing algorithm are planned to work on Jetson board. The routing information obtained with the outputs obtained because of this algorithm processed with the images taken from the drone camera will be transmitted to the autopilot from the Jetson card. It is planned to communicate with UART ports on Jetson and Pixhawk at 115.200 baud rate over UART. UART communication will be done over 8bit data, by selecting parity bit and 1 stop bit and timeout values. As the operating system, Linux Ubuntu operating system was chosen because it is more reliable and has less losses.

20 fps images coming from the camera will be able to be processed with our artificial neural network and image processing system. The whole system is on the drone and the system is independent from the ground control station. In this way, 2-way communication delays are prevented with telemetry, thus both the decrease in system performance and accidental crashes due to slowdown are prevented, and the task can be completed even in the presence of devices such as signal cutters in the environment. By seeing the system performance, system performance will be increased with extra data sets if necessary.

7.5 Restrictions

The biggest limitation of our project is the processing power, by implementing object recognition algorithms on the drone and providing automatic path control according to objects. Cameras have been minimized with the development of technology and they do not need much energy, but the processors that will take images from these cameras and process it and make artificial neural network analysis are not very small and their energy needs are higher. If these operations are carried out at the ground control station, we will lose as much time as the signals coming and going from the drone, and since this loss of time will cause the drone to process and receive commands compared to the previous state of the drone, it is likely to experience troubles, accidents and incidents. In addition, when there are signal breakers in the environment, the connection between the ground control station and the drone may be cut off, so that the drone will be left without a command and crash. For all these reasons, we decided to keep the processor on the drone, despite the weight and processing power constraints.

7.6 Risks

The drone is an expensive system. In addition, it can accelerate very quickly, go fast and crash due to a technical problem or user error. As the drones get bigger and the carbon-fiber propeller, whose propellers are more efficient, the damage to the environment and people increases. Because of these risks, we decided to make the drone as small as possible for our project tests and use a plastic propeller instead of a carbon-fiber propeller.

8. SYSTEM REQUIREMENTS

8.1. External Interface Requirements

8.1.1. User Interface

The user will operate on the Windows operating system.

8.1.2. Hardware Interface

Required equipment is below:

● A gimbal camera with high resolution, UART communication, and high FPS.

● High performance minicomputer with UART communication, camera input, CUDA and CuDNN features

● Cables for communication

● HDMI cable for image transmission

● Drone body

● Propeller 4 Pcs.

● Engines 4 Pcs.

● Electronic Speed Controller

● Controls

● Autopilot

● Telemetry

● Battery

● GPS

8.1.3. Software Interface

We will develop an automatic flying drone as a prototype. We will use the open source Pixhawk Cube Orange autopilot for this drone and we will use the Mission Planner software interface for this autopilot. We will develop with Python programming language for autopilot software development, artificial intelligence software development and image processing software development, and the integrated software environment we will use for Python will be PyCharm. We will use TensorFlow and OpenCV libraries for Python. We will use the Linux Ubuntu operating system for Jetson Xavier.

8.1.4. Communication Interfaces

An extra communication interface not required.

8.2 Functional Requirements

8.2.1 Profile Management - Use Case:

Use Case:

• Start computer software

• Enter username / password

• Change settings

• Exit

Diagram:

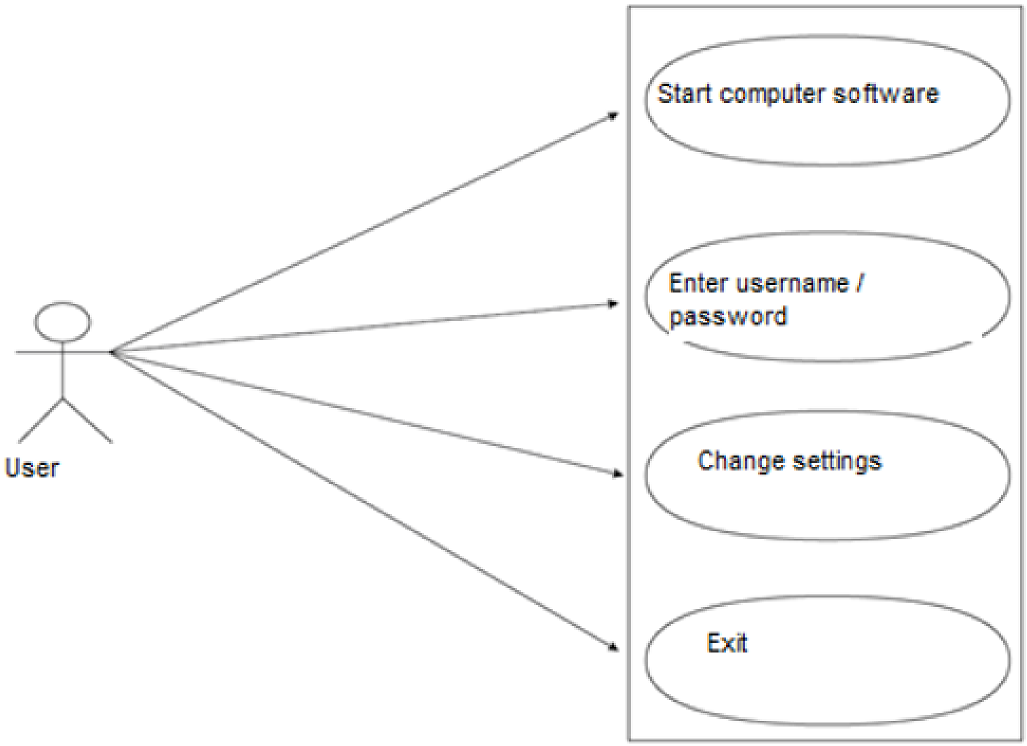

Figure 1: Profile Management - Use Case

Brief Explanation:

In our system, users cannot enter the system directly, users must be authorized. The use of drones is not legally open to everyone, people who have officially obtained the UAV0-UAV1 authorization certificate can use the drone. Users and passwords will be given to those who receive the authorization certificate and know how to use the system, so they will be able to change the settings of the system and use the system and log out whenever they would like.

Step-by-Step Explanation:

1. Anyone can start computer software.
2. Only authorized and qualified officials can log in to the system with a username and password.
3. The authorized user can change the system settings.
4. The user can log out of the system at any time when the usage is over.

8.2.2 Settings Menu - Use Case:

Use Case:

• Setting up a roadmap with waypoints

• Setting waypoints heights

• Adjusting drone speed

• Exit

Diagram:

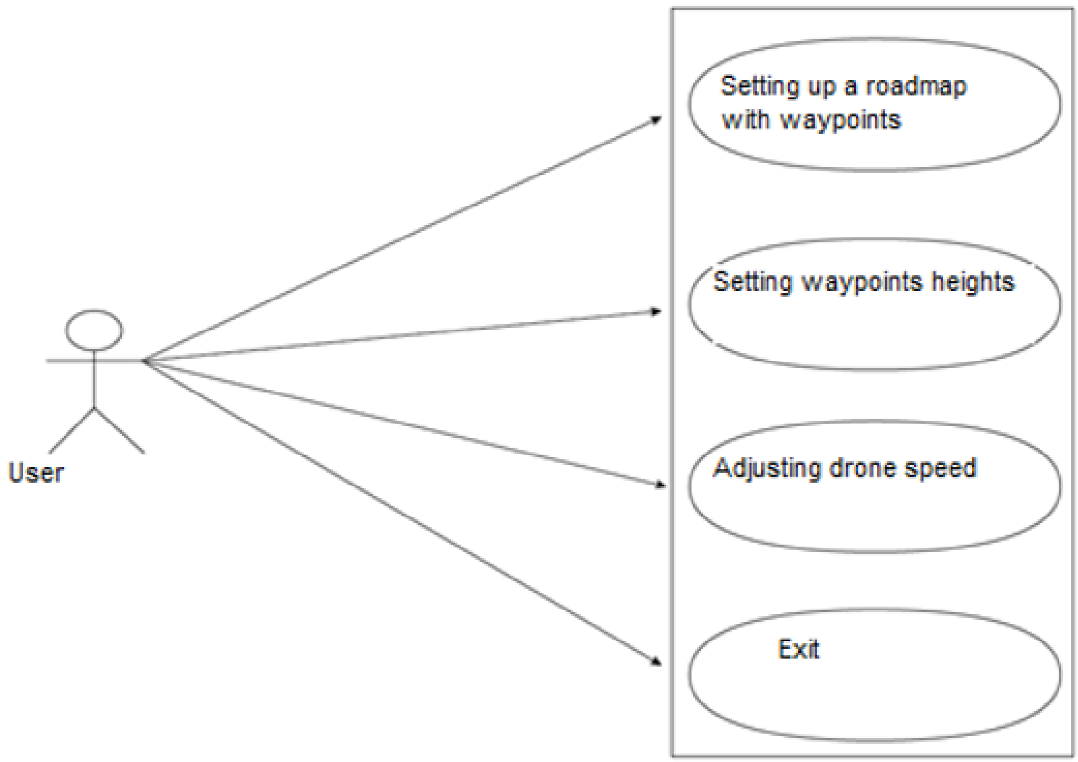

Figure 2: Settings Menu - Use Case

Brief Explanation:

With the settings, the user will be able to set the roadmap from google map. It will be able to determine from where and from which points the drone will pass. The drone will be able to adjust its speed. Height and speed adjustments can be made separately for the desired road parts, and the drone will not be active according to these new settings without sending these settings to the drone via telemetry. The user will be able to log out of the computer software at any time.

Step-by-step explanation:

1. The user sets up the roadmap by creating waypoints on the map.
2. User sets the height of waypoints; each waypoint can be at separate height.
3. User sets the drone speed. Standard drone can set speed or set speed separately for each path section.
4. The user can log out of the computer software at any time.

8.2.3 Detected Object Pass - Use Case

Use Case:

• Setting how many meters before objects change path

• Setting the transition of objects to the right, left, through or over

• Setting how many meters to pass objects

• Setting how many meters after passing objects to enter the defined path again

• Exit

Diagram:

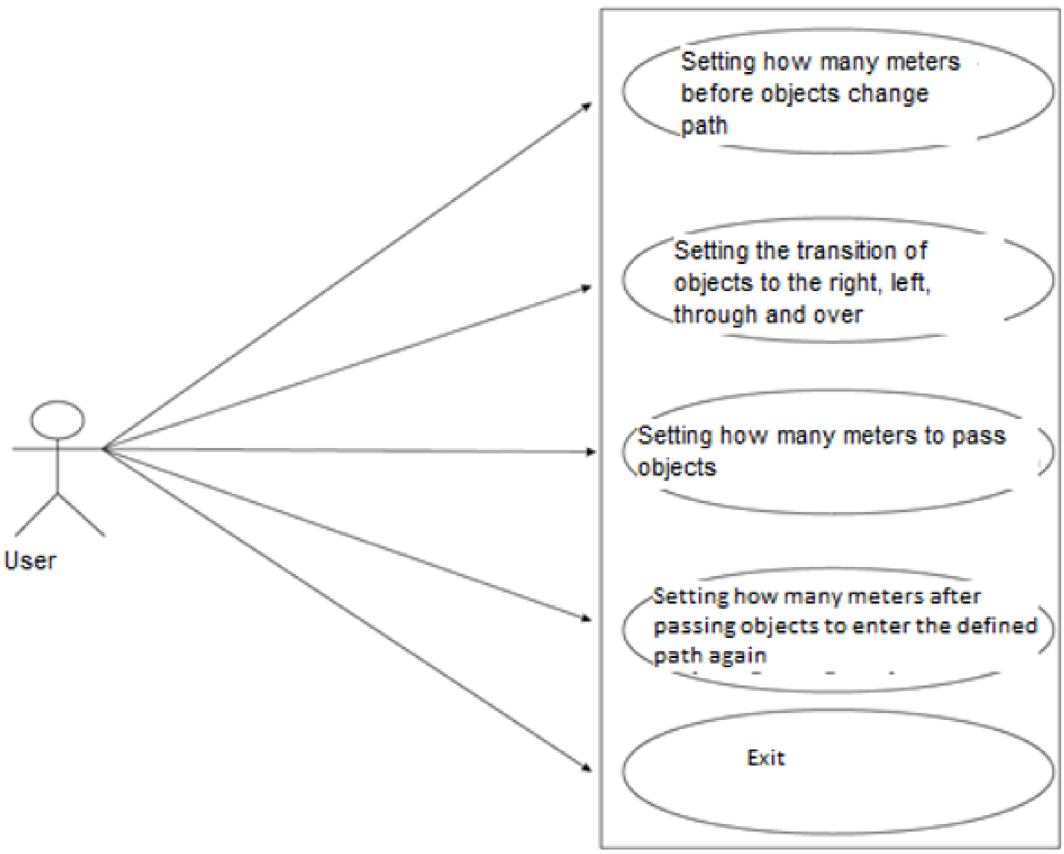

Figure 3: Detected Object Pass - Use Case

Brief Explanation:

Objects will be recognized in the system we train with artificial neural network and image processing algorithms. These settings are made on this menu for testing how close the drone should pass to objects and for drones of different sizes and different speeds.

Step-by-step Explanation:

1. According to the speed and size of the drone, how many meters before the objects should change the path, is adjusted.

2. According to the speed and size of the drone, it is set that it should pass from the right, left, inside or over the objects.

3. According to the speed and size of the drone, how many meters to pass the objects, is adjusted.

4. According to the speed and size of the drone, how many meters it will enter the defined path after passing the objects, is adjusted.

5. Exit from system after all settings.

8.2.4 Usage Menu - Use Case

`Use Case:`

• Send settings to drone

• Launch drone

• Send the new settings to the drone if necessary.

• Pause

• Land

• Take off

• Continue

• Finish the mission

Diagram:

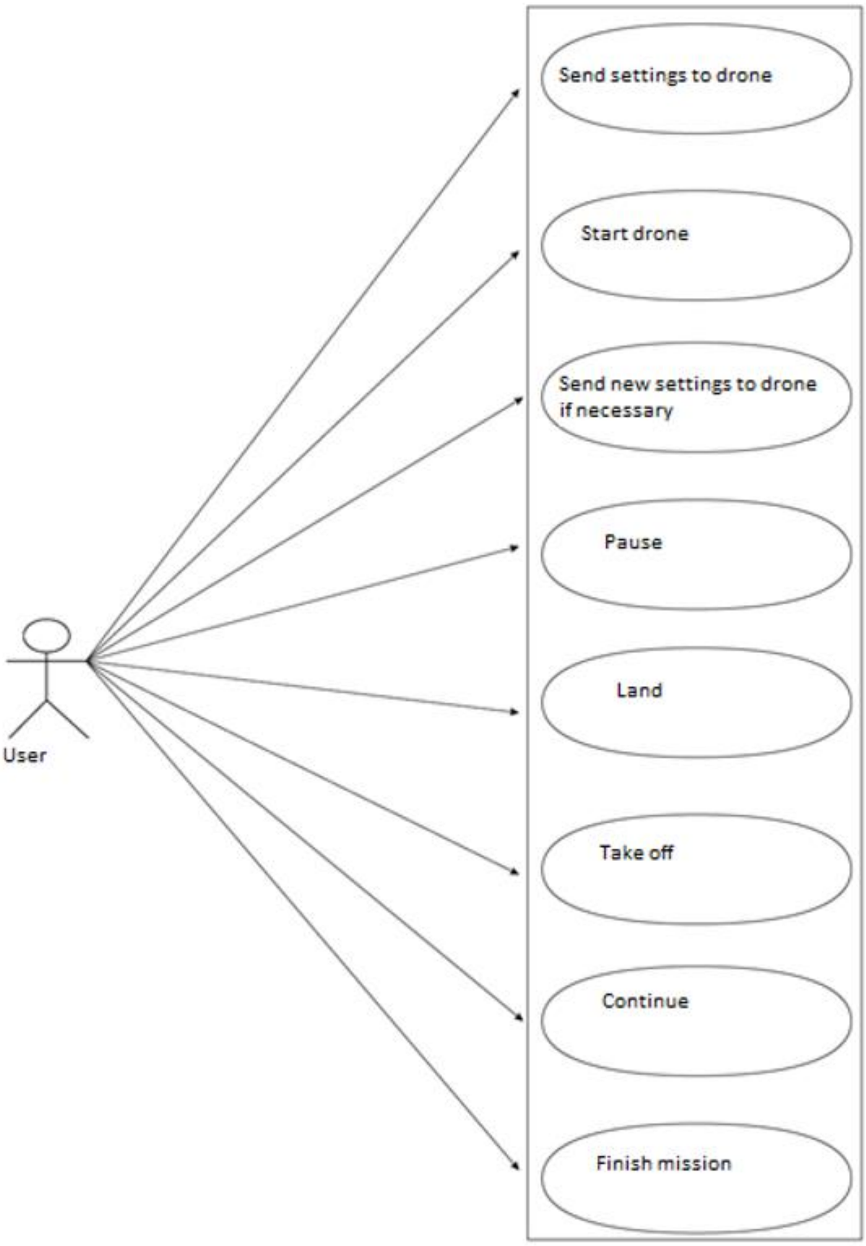

Figure 4: Usage Menu - Use Case

Brief Explanation:

The settings made for the drone are sent to the drone via telemetry and the drone starts, so that the drone automatically takes off and starts to go according to the roadmap. The drone travels completely automatically, so the user cannot make extra directions. When the drone encounters the objects in the data set while on the path, it applies the transition protocol defined for that object from that object (from the right, left, under, inside). When it encounters an object that is not in the dataset, it applies the defined transition procedure for the objects that are not in the dataset. At any time, new settings are sent to the drone via telemetry, and as soon as these settings are sent, the new settings become active and the drone moves according to these settings. The user can pause the mission at any time so that the drone waits in the air, can land the drone at any time, take off and resume the mission at any time. The user can finish the task at any time, or if this command is not given until the drone finishes the task, the task will be completed automatically. When the mission is over, the drone will automatically land. If the user exits the computer software during the mission, the drone will finish the mission and land. After landing, GCU and drone are turned off.

Step-by-step Explanation:

1. The settings are sent to the drone via telemetry.
2. The command to start the drone mission is given from the computer.
3. The drone automatically takes off and starts to go according to the roadmap.
4. When the drone encounters an object in the dataset, it completes the previously defined transit protocol for that object.
5. When the drone encounters an object that is not in the dataset, it completes the previously defined transition protocol for the object that is not in this dataset.
6. The user can send the new settings to the drone via telemetry at any time, so that the new settings are active and the drone continues to fly with these settings.
7. The user can pause, land, take off at any time. Then he can resume the task.
8. The mission ends and the drone automatically lands.
9. GCU and drone are turned off.

8.3 Performance Requirements

The requirements of the minicomputer that we will use for image processing and artificial neural network, are below:

1. 8 GB 128-bit LPDDR4x 59,7 GB/sec RAM or equivalent
2. 384 Core NVIDIA Volta™ GPU Equipped with 48 Tensor Cores or equivalent
3. 6-core NVIDIA Carmel Arm® v8.2 64-bit CPU or equivalent
4. 7-Way VLIW Image Processor or equivalent

8.4 Software System Features

8.4.1. Portability

The project will be able to work with open-source autopilots and autopilots with the appropriate SDK. Therefore, it can be transferred to most different drones on the market. It can be used not only for aircraft, but also for land and submarine vehicles.

8.4.2. Performance

• Each picture (frame) in the image taken from the camera will transmitted through the image processing software in 0.05 seconds.

• The received images will be transmitted through deep learning software within 0.04 seconds.

• It will be transferred to autopilot in 0.05 seconds.

• Image processing, deep learning, communication and autopilot software will run for a frame in a total of 0.14 seconds.

8.4.3. Availability

Our device will have two modes, semi-autonomous and fully autonomous. Thanks to these modes, the flight can be used completely automatically or with control assistance.

8.4.4. Adaptability

The image processing and artificial intelligence software and hardware used on the device are compatible to work with different autopilot hardware.

8.4.5. Scalability

The system has no scalability requirements.

8.5. Security Requirements

Since drones pose a danger to humans and animals by impact and system tests will be carried out, there should be no living being in the environment.

9.REFERENCES

[1] https://arxiv.org/abs/1804.02767

[2] https://medium.com/deep-learning-turkiye/yolo-algoritmas%C4%B1n%C4%B1-anlamak-290f2152808f

[3] https://www.youtube.com/watch?v=vRqSO6RsptU

[4] https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec

[5] https://en.wikipedia.org/wiki/Residual_neural_network

[6] https://stackoverflow.com/questions/58594235/yolo-training-yolo-with-own-dataset

[7] https://ieeexplore.ieee.org/document/9435338

Software Design Document

10. INTRODUCTION

10.1 Purpose

By making use of image processing and artificial intelligence software, it is aimed that the unmanned aerial vehicle detects certain objects and transmits information to the autopilot and completes its movement by avoiding the objects fully / semi-autonomously. In this way, the unmanned aerial vehicles, which are rapidly becoming widespread today, are fully automated according to the task, eliminating the human factor and minimizing the risks.

10.2 Scope

Unmanned aerial vehicles that can perform fully automatic missions, especially in military areas, are a very important need. Meeting this need will reduce the loss of people to almost zero in the military defence of countries. Drones have become popular in recent years and have started to be used in almost every field that will benefit humanity in this regard. People from all over the world, students, academics, private and public sectors continue to launch new projects and new products related to drones. Despite being one of the fastest-growing areas of technology, drone technology has its own shortcomings. One of the most important of these is safe flight with object recognition. Drones cause the most accidents by crashing into objects while moving. They can crash into electricity poles, trees, hilly and mountainous places, buildings, and even more people, cars, etc. In addition, the delay in the use of drone technology near the ground (control patrols in factories, controls between buildings, close-up automatic drone commercials, etc.) is due to the inability of drones to fly safely in these areas due to crashes. Various technologies are being considered as solutions to these issues, and the implementation of object recognition algorithms on the drone that we are planning to do and the project of providing automatic path control according to objects is one of the most important steps that can be taken in this regard.

10.3 Glossary

Autopilot	Autopilot software is designed to limit the device, stabilize it, perform the task autonomously and device safety.
Artificial Intelligence	Learning outcomes revealed by algorithms consisting of linear algebra methods used to calculate the pattern and trend in data sets prepared for a specific job
Image Processing	By performing various matrix and mathematical calculations on the numerical units of visual elements such as photographs and/or videos, the desired information is obtained from the images, or the desired information is added to the images.
CNN	Convolutional Neural Networks
YOLO	You Only Look Once
ReLU	Rectified Linear Unit
GCU	Ground Control Unit [UAV ground control station (UAV – Unmanned aerial vehicles)]

10.4 Overview of Document

In this document, the integration process of artificial intelligence and image processing algorithms into an unmanned aerial vehicle and the process of gaining the ability to perform missions are explained. It is aimed to turn deep learning methods into software, to take matrix calculations on images as inputs in accordance with deep learning methods, and to calculate the outputs to be obtained by means of software. It is our ultimate goal that the device gains the ability to perform autonomous missions by transmitting the outputs to be obtained from image processing or artificial intelligence algorithms, which will work in harmony with each other, to the autopilot software.

10.5 Motivation

Artificial intelligence, which has rapidly increased in popularity in our country and in the world and adapted to people's lives very quickly, unmanned aerial vehicles that revolutionized the history of the world war, and our interest in image processing technology, which is an indispensable element of these two fields, played a key role in our participation in such multi-engineering work. At the same time, we, as senior students of Çankaya University Computer Engineering, who have adopted the ideal of carrying our country further than all the countries of the world in technological developments, are motivated to carry the name of our school and our country to the top.

Artificial intelligence technologies, which are at the base of the change and transformation in the world of computers and electronics, have started to be used in almost every sector. The most important revolution since the industrial revolution, where human power turned into machine power, is the development of artificial intelligence. Machines are developing not only in physical power but also in mental power. There are two most important factors needed for this development: 1. Data 2. Qualified people with high knowledge As a country, it is aimed to extract these low-cost resources from our own essence and to integrate it into the development of artificial intelligence and our own products. If it works integrated with artificial intelligence, the thing that reaches high performances and will be indispensable in the near future is "image processing". The information about an image and the information that will be uploaded to those images by mathematical operations made on pixel basis over images such as photographs or videos are very exciting. And finally, autopilot technology, which is fully task-oriented and fully automatic, is the most basic element of unmanned vehicles. The combination of autopilot, artificial intelligence and image processing will pave the way for the emergence of a product that will minimize human loss. And we will be proud of the emergence of this product.

11. Software Architecture

In order to change the motion directions of the unmanned aerial vehicle, the photograph data of the selected objects will be read with the image processing software. Information about the pixel details of the photos will be taken as input to the artificial intelligence algorithm. As a result of the model formed, learning outcomes will be obtained. Objects will be detected according to the learning outcomes. The distance information of the detected object will be transmitted to the autopilot software via the communication protocol with the image processing algorithm. According to the information he received, the autopilot will provide the movements of the unmanned aerial vehicle.

1. The selected objects will be run on an independent device and will be learned by a deep learning algorithm.
2. Continuous pixel information of video images will be taken in real time.
3. Pixel information will be analysed in real time with an artificial intelligence algorithm that has been learned.
4. With the detection rate of 90% and higher by the artificial intelligence algorithm, the information of the detected object will be transmitted to the image processing algorithm.
5. The image processing algorithm will transmit the distance and direction information of the object to the autopilot as a result of the detection of the object.
6. The autopilot will maintain or change its direction according to the information it receives.

The system will operate with automatic mission-ready power-up. Against the risks that may occur, the user will be connected to the autopilot with a remote control and will enable the unmanned aerial vehicle to be directed.

11.1 Class Diagram

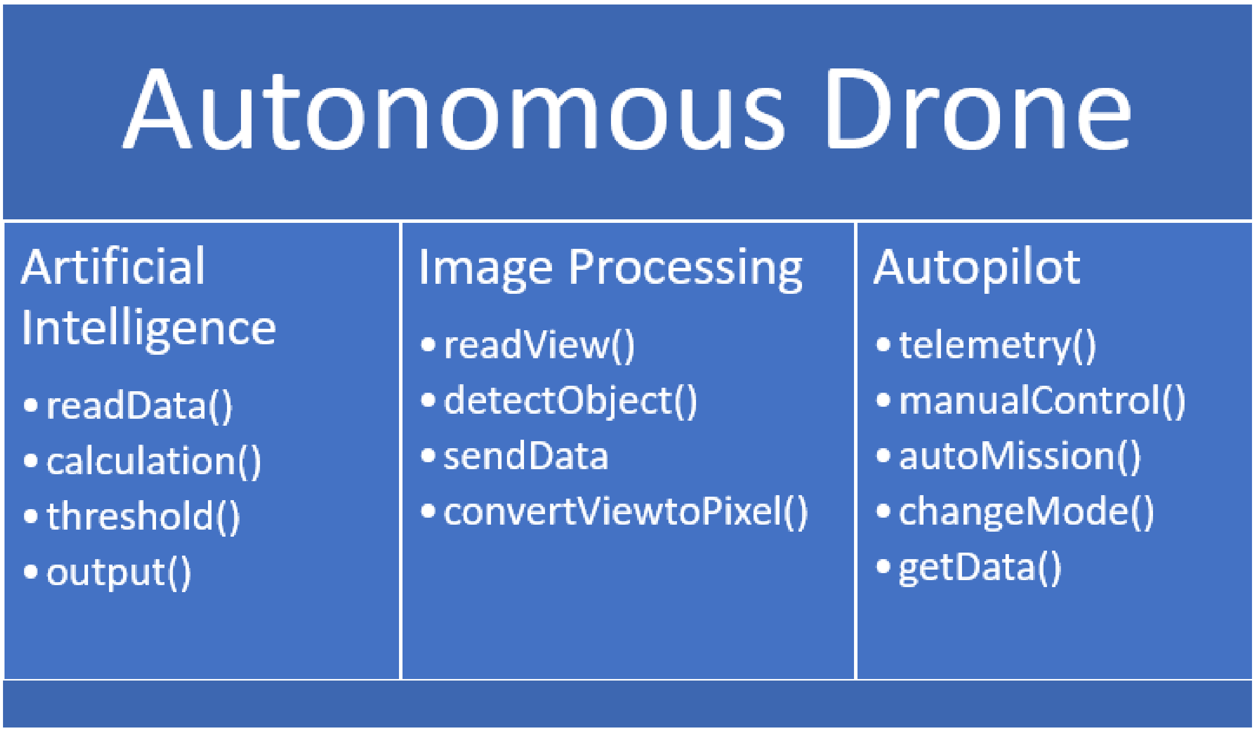

Figure 1: Class Diagram

11.2 Decomposite Diagram

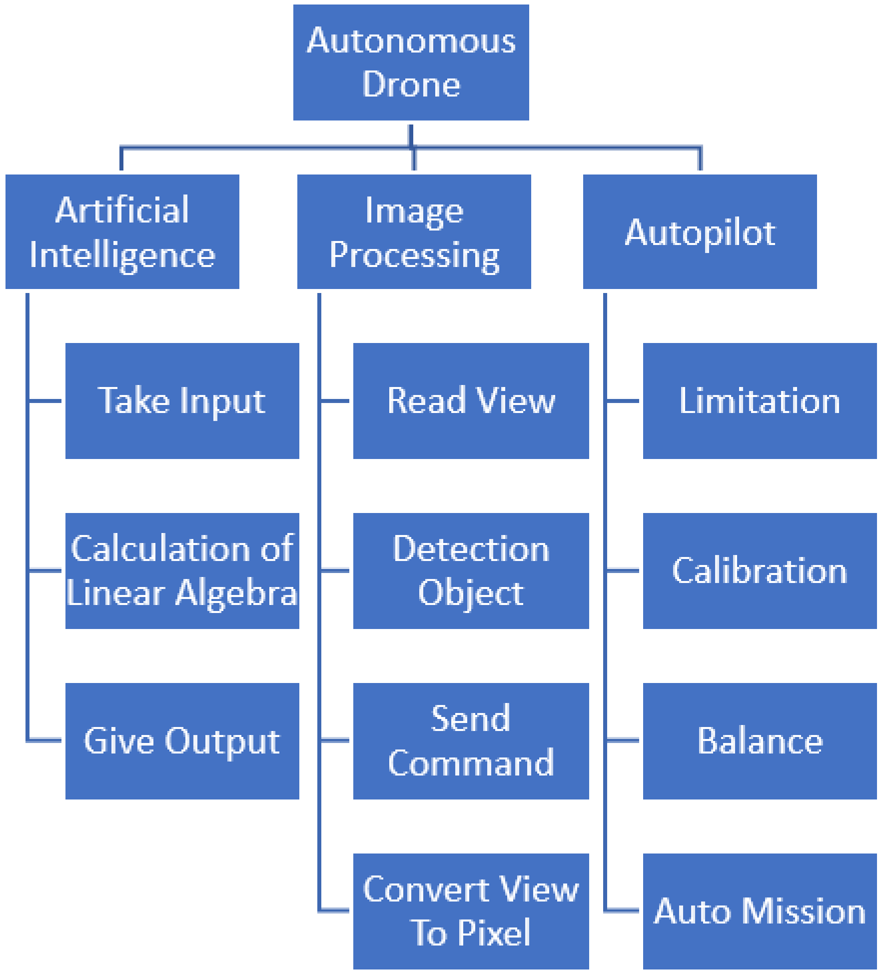

Figure 2: Decomposite Diagram

11.3 Data Flow Diagram

11.3.1 Level 0 of DFD

Mission: Mission is the process by which the drone detects certain objects and performs evasive maneuvers (with changes in speed and direction) and these occur fully autonomously.

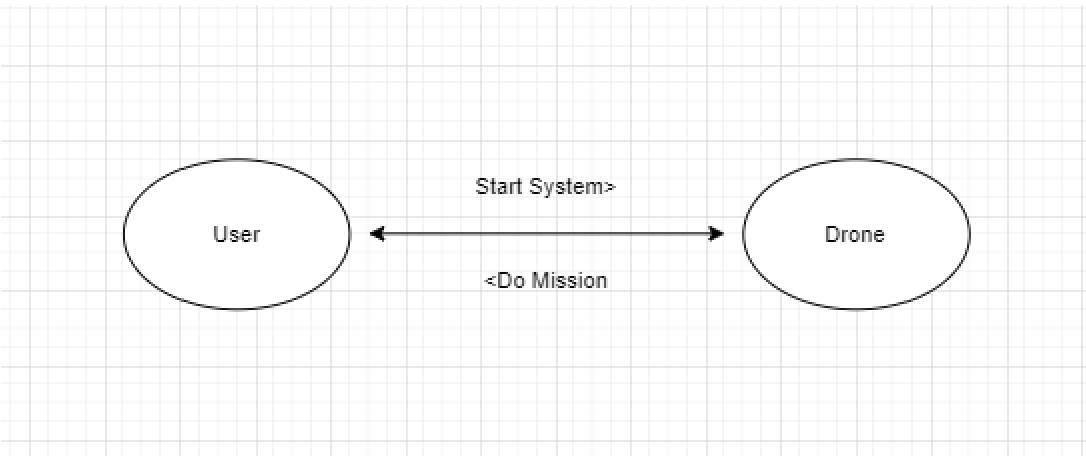

Figure 3: Level 0 of DFD

11.3.1 Level 1 of DFD

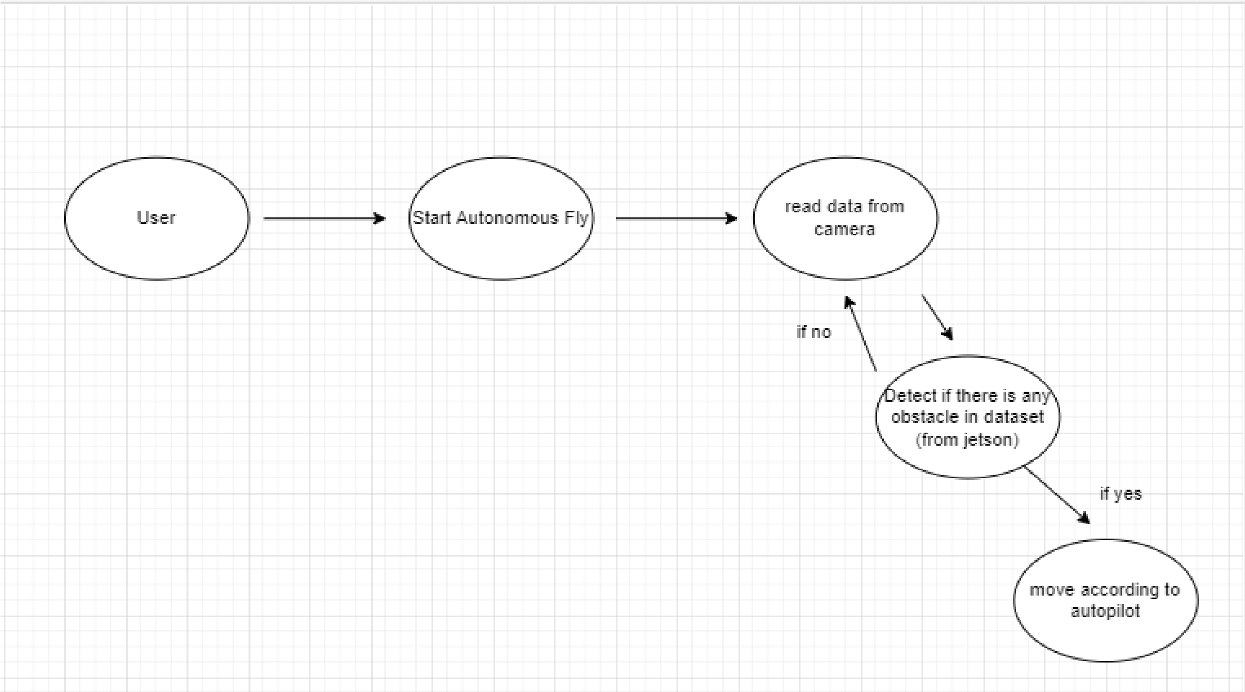

Figure 4: Level 1 of DFD

11.4 Activity Diagram

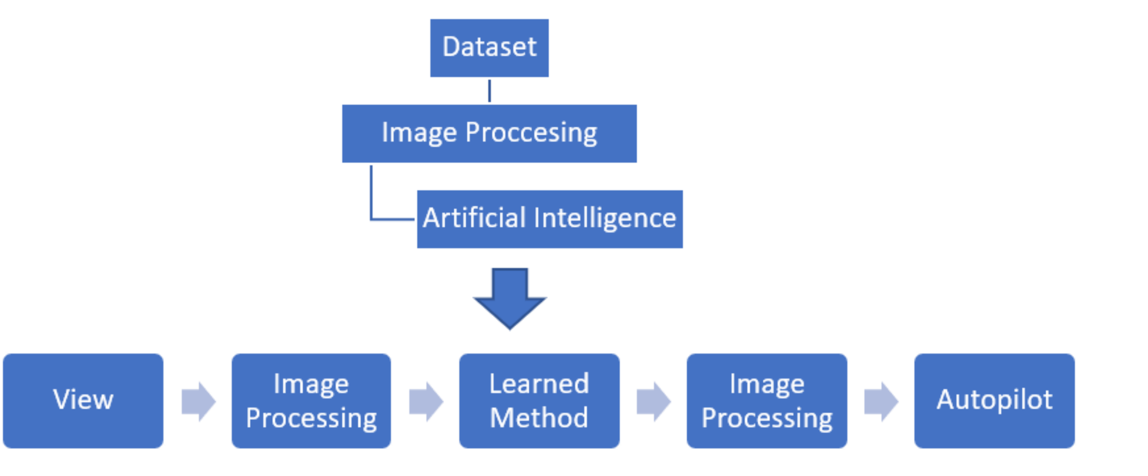

Figure 5: Activity Diagram

11.5 Sequence Diagram

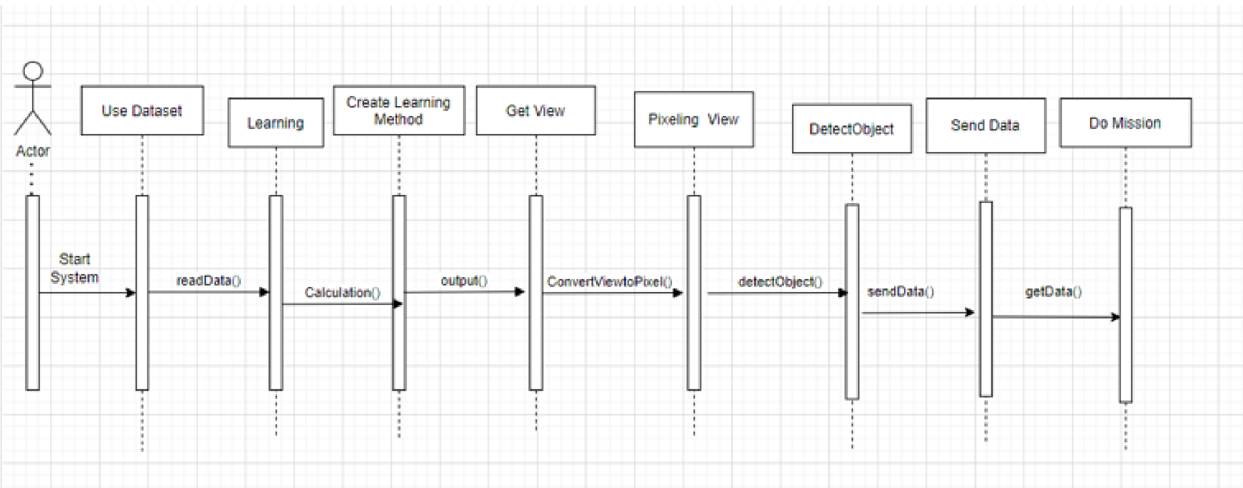

Figure 6: Sequence Diagram

12.USE CASE REALIZATIONS

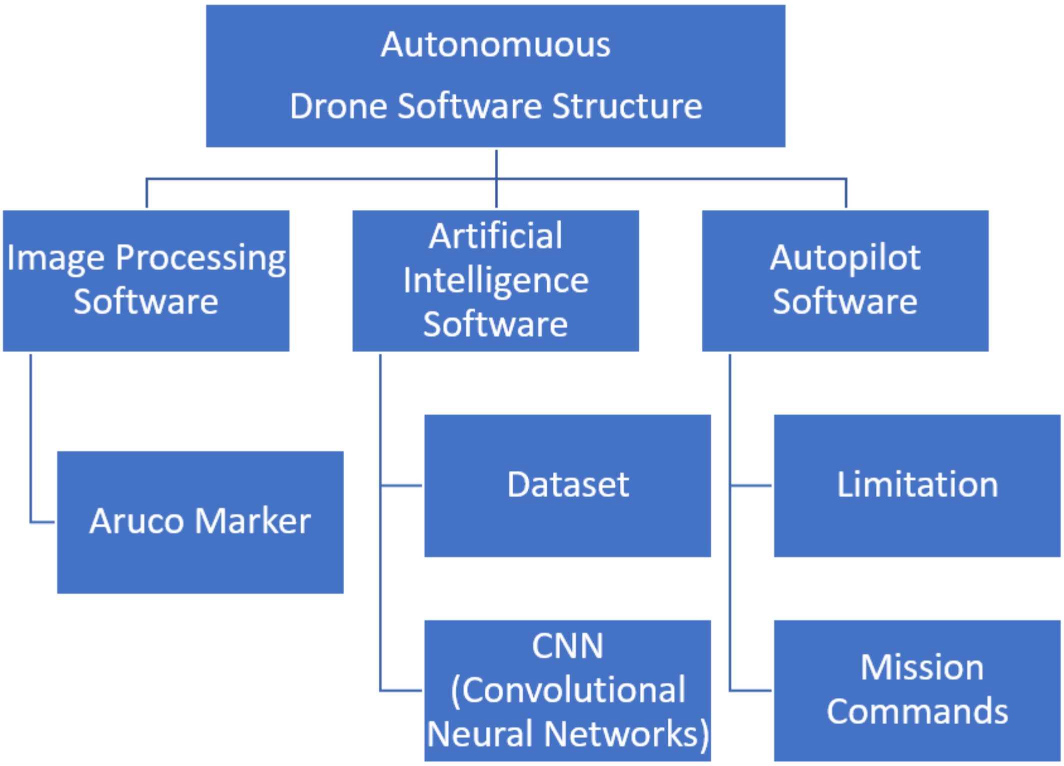

Figure 7: Software Structure of Autonomous Drone

12.1 Image Processing software

It will be used to determine the location, distance and distinguishing features of the object. It will be ensured that the features of the objects are transmitted as input to the artificial intelligence algorithm at the pixel level. With some filters to be used, the distinguishing features on the image will be determined.

12.1.1 Aruco marker

A standard size object that will be used in pixel cm conversion to determine the distance information of the object and to be calibrated for the first time.

12.2 Artificial Intelligence Software

It is the software that will be used in the formation of the learning algorithm and the learned model. This software uses the CNN structure, which is one of the deep neural network models, and puts the inputs into various linear algebraic equations and gives learning output according to the result obtained.

12.2.1 Data set

Data sets are at the heart of the Artificial Intelligence Algorithm's ability to provide learning. The more diverse the datasets contain, the higher the accuracy of the AI algorithm in learning outcomes. Having as many images as possible in the objects we selected while creating the data set will increase our learning percentage. If the learning rate is low, the data set can be expanded.

12.2.2 CNN (Convolutional Neural Networks)

Convolutional Neural Networks (CNN) is a class of artificial neural networks most commonly used for image analysis in deep learning. Because CNN uses ReLU as activation, there can be a large number of data in the nerve and learning can take place without burden.

12.3 Autopilot software

It is software that enables unmanned vehicles to perform autonomous or semi-autonomous tasks.

12.3.1 Limitation

Since device security is more important than anything else, developers make limits to secure the device with the help of autopilot. These limitations can be speed, incline, ascent, descent, etc.

12.3.2 Task Commands

According to the outputs from image processing and artificial intelligence software, information is transferred to the autopilot. The transmitted information is made meaningful on autopilot. Acceleration, deceleration, maneuvering, stopping and all other movements of our unmanned aerial vehicle according to the position of the object and the object information are transferred to the hardware with the autopilot software.

Figure 8: Mission Planner Moving Drone on Simulation Pixhawk

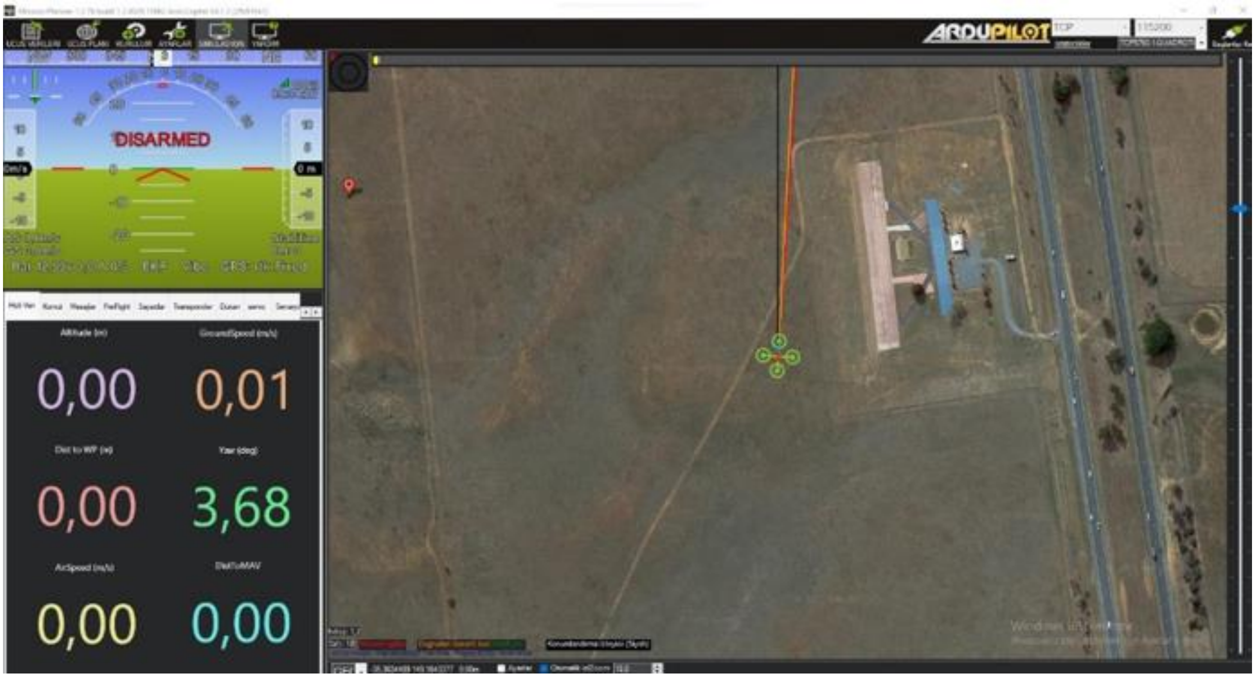

Figure 9: Mission Planner Do Mission

13. REFERENCES

[1] About UAVs [Online]. Available: https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

[2] Study about Drone [Online].
Available: https://internetofthingsagenda.techtarget.com/definition/drone

[3] Study about “Attitude Control of Multicopter” [Online].
Available: https://core.ac.uk/download/pdf/295548558.pdf

[4] Study about “Building a Quadcopter with Arduino” [Online].
Available: https://abis.bozok.edu.tr/indir.php?file_id=710

[5] Pixhawk Overview [Online]. Available: https://ardupilot.org/copter/docs/common-pixhawk-overview.html

[6] About multicopters [Online]. Available: https://www.dronedoktoru.com/multikopter-nedir.html

[7] How UART works [Online]. Available: https://herenkeskin.com/uart-nedir-ve-nasil-calisir/

[8] Shobhit Bhatnagar, “I Classification of Fashion Article Images using Convolutional Neural Networks,” 2017, Fourth International Conference on Image Information Processing (ICIIP) [Online]. Available: https://ieeexplore.ieee.org/document/8313740

[9] Xia Zhao, "A novel three-dimensional object detection with the modified You Only Look Once method," 2018, International Journal of Advanced Robotic Systems [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1729881418765507

[10] Daniel Pestana, "A Full Featured Configurable Accelerator for Object Detection with YOLO, " 2021, IEEE [Online]. Available

https://ieeexplore.ieee.org/document/9435338

[11] About YOLO algorithm [Online]. Available: https://medium.com/deep-learning-turkiye/yolo-algoritmas%C4%B1n%C4%B1-anlamak-290f2152808f

[12] About YOLO algorithm [Online]. Available: https://www.youtube.com/watch?v=vRqSO6RsptU

[13] You Only Look Once: Unified, Real-Time Object Detection [Online]. Available: https://arxiv.org/pdf/1506.02640.pdf

[14] About Residual blocks — Building blocks of ResNet [Online]. Available: https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec

[15] Convolutional Neural Network (CNN) [Online]. Available: https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad

[16] Smart Autopilot Drone System for Surface Surveillance and Anomaly Detection via Customizable Deep Neural Network [Online]. Available: https://www.researchgate.net/publication/338529296_Smart_Autopilot_Drone_System_for_Surface_Surveillance_and_Anomaly_Detection_via_Customizable_Deep_Neural_Network

[17] Detection of a Moving UAV Based on Deep Learning-Based Distance Estimation [Online]. Available: https://www.mdpi.com/2072-4292/12/18/3035/htm

14. TEST PLAN

14.1 INTRODUCTION

Version Control

Version No	Description of Changes	Date
1.0	First Version	April 01, 2022

14.2 Overview

The Autonomous Drone Control project has been designed with artificial intelligence and image processing algorithms to enable the drone to fly in a controlled way on a track. Functions, use cases and requirements specified previously in our SRS and SDD documents.

14.3 Scope

This document contains information about the test plan of the Autonomous Drone Control project. The following sections briefly explain what the test criteria will be and how we will do the testing part.

14.4 Terminology

Acronym	Definition
SRS	Software Requirements Specification
SDD	Software Design Document
SDK	Software Development Kit
PID	Proportional–Integral–Derivative

14.5 FEATURES TO BE TESTED

This section lists and gives a brief description of all the major features to be tested. For each major feature there will be a Test Design Specification added at the end of this document.

14.5.1 Computer Vision Software Test (CV_ST)

14.5.1.1 Object Detection Test (CV_ST.ODT)

Correct detection of certain objects with the image taken from the camera is tested based on the following.

14.5.1.1.1 How Long Does It Take to Detect (FPS) (CV_ST.ODT.FPS)

In this test, any of the previously determined objects must be detected within a maximum of 2 seconds after entering the camera frame.

14.5.1.1.2 Accuracy Rate (CV_ST.ODT.AC)

In this test, the accuracy or inaccuracy of the detected object is tested (0-1).

14.5.1.1.3 Detection Distance (CV_ST.ODT.DS)

In this test, the detection of the detected object at a specified distance is tested.

14.5.1.1.4 Color Detection Test (CV_ST.ODT.CD)

With the software test prepared in this test, the correct color is determined by comparing the colors that we will use in the objects with the camera and the human eye.

14.5.1.2 Commanding Accuracy Test (CV_ST.CAT)

In this test, the position of the detected object on the image is determined, the motion command is sent to the SDK software by adhering to the predetermined motion directions according to the determined position.

14.5.2 Hardware Tests (HT)

14.5.2.1 Signal Performance Test (HT.SPT)

There is a certain distance between the device and the emergency control, if there is command transmission despite this distance, the test is successful.

14.5.2.2 Flight Speed Test (HT.FST)

Testing the movements of the drone after detecting the obstacle in terms of speed and time.

14.5.2.3 Battery Health Test (HT.BHT)

The batteries used are tested with battery testers to see if there is an evenly distributed voltage to each cell.

14.5.2.4 Vibration Test (HT.VT)

Carrying the drone in a deactivated state in the vehicle on a determined bumpy road, then checking whether the device is working or not.

14.5.2.5 Flight Time Test (HT.FTT)

-The time tester of how long the drone is suspended in the air.
-Drone completion time test with remote control (Slow, Medium, Fast use)
-Drone's autonomous track completion time test

14.5.2.6 Camera Test (HT.CT)

Connects the camera to any computer. This is tested with any application that opens a camera view.

14.5.2.7 GPS Test (HT.GT)

Location accuracy is determined by entering the latitude and longitude data obtained from GPS into the map application of a mobile device that has a map application and can access location information.

14.5.3 Autopilot Software Tests

14.5.3.1 SDK Test for Simulation (Connection-Contact) (AST.STS)

This test is done for the simulation side before flying the drone. It is done via Mission Planner. Its purpose is to send commands to the drone.

14.5.3.1.1 Command Receive Accuracy Test (AST.STS_CRAT)

The purpose of this test is to test whether the commands we send to the drone are correctly received. We can also do this test while the drone is on the ground.

14.5.3.1.2 Axiom Truth Test (AST.STS_ATT)

In this test, we test whether the drone receives the commands we send correctly and applies them accordingly.

14.5.3.1.3 Command Receiving Test from Different Devices (AST.STS_CRTDD)

In this test, it is tested whether the drone receives commands from different devices.

14.5.3.2 Autopilot Flight Test (AST.AFT)

Its purpose is to fly the drone according to the command in the real environment.

14.5.3.2.1 PID Test (AST.AFT_PID)

It is the test that checks whether the drone vibrates and shakes in the air. If it vibrates and oscillates, it means that the PID settings are not correct and must be corrected.

14.5.3.2.2 SDK (AST.AFT_SDK)

After the tests in the simulation are over, the commands we apply on the SDK are tested again in the flight test.

14.5.3.3 Direction Test of Motors (AST.DTM)

It is for testing the correctness of the direction of the motors on the drone. Hexacopter-style drones must have 3 engines in the ccw direction and 2 in the cw direction.

14.5.3.4 Calibration Test of Gyroscope (AST.CTG)

These are the tests made so that the drone can stand without swaying in a windless area.

14.5.3.5 Compass Calibration Test (AST.COMCT)

The compass uses the earth's magnetic field to determine which direction the drone is heading. The magnetic fields in our flight area (for example, high voltage wires) or the magnetic field created by the power cables and batteries on the multicopter affect the operation of this compass sensor.

14.5.3.6 Controller Calibration Test (AST.CCT)

It is a feature that is tested when using different controllers or introducing a new controller to the drone. It allows us to see the min and max (controller signal) values of the controller. The autopilot is calibrated according to its values.

FEATURES NOT TO BE TESTED

We will not do these tests:
-Mini Computer (Jetson)
-Autopilot Hardware
-Drone Body Strength
-Internal Hardware (Motor-Esc)
because they are done by those who sell them.

ITEM PASS/FAIL CRITERIA

Describe the general rule to use to decide when a test case passes and when it fails.

Exit Criteria

Describe under what conditions the testing of the product is considered successful. Some examples are:

- 100% of the test cases are executed
- 93% of the test cases passed
- All High and Medium Priority test cases passed

REFERENCES

[1] SRS link:
https://github.com/CankayaUniversity/ceng-407-408-2021-2022-Autonomous-Drone-Control/wiki/SRS-(Software-Requirements-Specification)

[2] SDD link:
https://github.com/CankayaUniversity/ceng-407-408-2021-2022-Autonomous-Drone-Control/wiki/SDD-(Software-Design-Document)

14.6 TEST DESIGN SPECIFICATIONS

14.6.1 Computer Vision Software Test (CV_ST)

14.6.1.1 Object Detection Test (CV_ST.ODT)

TC ID	Requirements	Priority	Scenario Description
CV_ST.ODT.FPS.01	3.2.3	H	Calculate how long it's take to detect object
CV_ST.ODT.AC.01	3.2.3	H	Measuring the accuracy of the detected object
CV_ST.ODT.DS.01	3.2.3	H	Calculation of the detection distance of the object
CV_ST.ODT.CD.01	3.2.3	H	Making accurate color determinations of objects

14.6.1.2 Commanding Accuracy Test (CV_ST.CAT)

TC ID	Requirements	Priority	Scenario Description
CV_ST.CAT.01	3.2.3	H	The command given according to the position of the object is calculated

14.6.2 Hardware Tests (HT)

14.6.2.1 Signal Performance Test (HT.SPT)

TC ID	Requirements	Priority	Scenario Description
HT.SPT.01	3.2.4	H	The signal performance between the device and the controller is measured.

14.6.2.2 Flight Speed Test (HT.FST)

TC ID	Requirements	Priority	Scenario Description
HT.FST.01	3.2.4	H	Calculation of the movements that the drone will make after detecting the obstacle as speed and time.

14.6.2.3 Battery Health Test (HT.BHT)

TC ID	Requirements	Priority	Scenario Description
HT.BHT.01	3.2.4	H	It is calculated whether the batteries used are sufficient or not.

14.6.2.4 Vibration Test (HT.VT)

TC ID	Requirements	Priority	Scenario Description
HT.VT.01	3.2.4	H	It is determined how durable the drone is.

14.6.2.5 Flight Time Test (HT.FTT)

TC ID	Requirements	Priority	Scenario Description
HT.FTT.01	3.2.4	H	-The time tester of how long the drone is suspended in the air. -Drone completion time test with remote control (Slow, Medium, Fast use) -Drone's autonomous track completion time test

14.6.2.6 Camera Test (HT.CT)

TC ID	Requirements	Priority	Scenario Description
HT.CT.01	3.2.4	H	Opening of the camera is checked.

14.6.2.7 GPS Test (HT.GT)

TC ID	Requirements	Priority	Scenario Description
HT.GT.01	3.2.4	H	The accuracy of the location is calculated

14.6.3 Autopilot Software Tests (AST)

14.6.3.1 SDK Test for Simulation (AST.STS)

TC ID	Requirements	Priority	Scenario Description
AST.STS_ CRAT.01	3.2.4	H	Checks commands we send to the drone are correctly received
AST.STS_ ATT.01	3.2.4	H	Checks drone receives the commands we send correctly and applies them accordingly.
AST.STS_ CRTDD.0 1	3.2.4	H	Tests drone receives commands from different devices.

14.6.3.2 Autopilot Flight Test (AST.AFT)

TC ID	Requirements	Priority	Scenario Description
AST.AFT_ PID.01	3.2.4	H	Tests that checks whether the drone vibrates and shakes in the air.
AST.AFT_ SDK.01	3.2.4	H	After the tests in the simulation are over, SDK commands are tested again in the flight test.

14.6.3.3 SDK Test for Simulation (AST.STS)

TC ID	Requirements	Priority	Scenario Description
AST.STS.0 1	3.2.4	H	SDK tests are retested.

14.6.3.4 Direction Test of Motors (AST.DTM)

TC ID	Requirements	Priority	Scenario Description
AST.DTM.01	3.2.2	H	It is for testing the correctness of the direction of the motors on the drone.

14.6.3.5 Calibration Test of Gyroscope (AST.CTG)

TC ID	Requirements	Priority	Scenario Description
AST.CTG.01	3.2.2	H	Tests the drone can stand without swaying in a windless area.

14.6.3.6 Compass Calibration Test (AST.COMCT)

TC ID	Requirements	Priority	Scenario Description
AST.COMCT.01	3.2.4	H	Determine which direction the drone is heading.

14.6.3.7 Controller Calibration Test (AST.CCT)

TC ID	Requirements	Priority	Scenario Description
AST.CCT.01	3.2.4	H	Tests using different controllers or introducing a new controller to the drone.

14.7 Detailed Test Cases

CV_ST.ODT.FPS.01

TC_ID	**CV_ST.ODT.FPS.01**
Purpose	Object Detection Time
Requirements	3.2.3
Priority	High
Estimated Time Needed	2 Second
Dependency	Object, Camera, Software program open and usable
Setup	Trained model running
Procedure	[A01] Turning on the camera
	[A02] Running the program
	[A03] Displaying the object

	[V01] Successful recognition of the object in the required time
	-
Cleanup	Stop the program

CV_ST.ODT.AC.02

TC_ID	**CV_ST.ODT.AC.02**
Purpose	Object Detection
Requirements	3.2.3
Priority	High
Estimated Time Needed	2 Second
Dependency	Object, Camera, Software program open and usable
Setup	Trained model running
Procedure	[A01] Turning on the camera
	[A02] Running the program
	[A03] Displaying different, unrelated objects
	[V01] Successful recognition of the object
	-
Cleanup	Stop the program

CV_ST.ODT.DS.03

TC_ID	**CV_ST.ODT.DS.03**
Purpose	Object Detection
Requirements	3.2.3
Priority	High
Estimated Time Needed	2 Second
Dependency	Object, Camera, Software program open and usable
Setup	Trained model running
Procedure	[A01] Turning on the camera
	[A02] Running the program
	[A03] The object is displayed from different distance meters.
	[V01] Successful recognition of the object
	-
Cleanup	Stop the program

CV_ST.ODT.CD.04

TC_ID	**CV_ST.ODT.CD.04**
Purpose	Object Detection
Requirements	3.2.3
Priority	High
Estimated Time Needed	2 Second
Dependency	Object, Camera, Software program open and usable
Setup	Trained model running
Procedure	[A01] Turning on the camera
	[A02] Running the program
	[A03] Objects of different colors are displayed.
	[V01] Successful recognition of the object
	-
Cleanup	Stop the program

CV_ST.CAT.01

TC_ID	**CV_ST.CAT.01**
Purpose	Testing the issuance of the redirect command
Requirements	3.2.3
Priority	High
Estimated Time Needed	2 Second
Dependency	Object, Camera, Software program open and usable
Setup	Trained model running
Procedure	[A01] Turning on the camera
	[A02] Running the program
	[A03] Displaying the object
	[A04] Correct entry of function parameters according to the position of the object on the screen
	[V01] Successful piloting of the drone
	-
Cleanup	Stop the program

HT.SPT.01

TC_ID	**HT.SPT.01**
Purpose	Testing Signal Performance
Requirements	3.2.4

Priority	High
Estimated Time Needed	5 Minutes
Dependency	The Controller, Receiver module on the drone
Setup	Telemetry settings have been made
Procedure	[A01] Turning on the remote controller
	[A02] Powering the drone
	[A03] Physically moving away up to the distance determined gradually with the remote control.
	[V01] Successful reception of the signal when the maximum distance is reached
	-
Cleanup	Turning off the power and turning off the remote control

HT.FST.01

TC_ID	**HT.FST.01**
Purpose	Testing the axiom speed of the drone
Requirements	3.2.4
Priority	High
Estimated Time Needed	Flight Time
Dependency	Camera, Entire system working
Setup	All autopilot settings are made and computer vision software is running
Procedure	[A01] Powering the drone
	[A02] Bringing the drone to a certain height with the remote controller
	[A03] Making flight mode autonomous
	[A04] The drone detects the obstacle
	[V01] Successfully ensuring the avoidance movement of the drone according to the detected obstacle in the desired conditions
	-
Cleanup	Turning off the power

HT.BHT.01

TC_ID	**HT.BHT.01**
Purpose	Testing the Health of the Battery
Requirements	3.2.4
Priority	High

Estimated Time Needed	30 Second
Dependency	Battery, Battery Testing tool
Setup	-
Procedure	[A01] Connecting the battery testing tool to the battery connector
	[V01] Determining whether the battery is healthy or not by observing that the voltage values of the battery cells are close to each other
	-
Cleanup	Removing the battery testing tool from the connector

HT.VT.01

TC_ID	**HT.VT.01**
Purpose	Performing the vibration test
Requirements	3.2.4
Priority	High
Estimated Time Needed	15 Minutes
Dependency	Drone, vehicle, bumpy road
Setup	-
Procedure	[A01] Putting the drone in the car
	[A02] Driving the car over rough roads
	[V01] Powering the drone after the trip and the system continuing to work in the same way
	-
Cleanup	Removing the drone from the vehicle

HT.FTT.01

TC_ID	**HT.FTT.01**
Purpose	Testing the drone's flight time
Requirements	3.2.4
Priority	High
Estimated Time Needed	2 Hours
Dependency	Battery,Drone
Setup	-
Procedure	[A01] Powering the drone
	[A02] Bringing the drone to a certain height with the remote controller and holding the air until the battery level reaches the critical point and measuring the time.

	[A03] Attaching a full battery to the drone
	[A04] To measure the flight and time until the battery level reaches the critical point in sport mode with the drone's remote control.
	[A05] Attaching a full battery to the drone
	[A06] Autonomous completion of the track and time measurement of the drone.
	[V01] Detection of drone flight times before the battery runs out
	-
Cleanup	Turning off the power

HT.CT.01

TC_ID	**HT.CT.01**
Purpose	Camera Test
Requirements	3.2.4
Priority	High
Estimated Time Needed	20 Second
Dependency	Camera, Computer
Setup	An application that can open a camera view
Procedure	[A01] Camera connects to computer
	[A02]The program that can open the camera image is run
	[V01] Accurate capture of the camera image
	-
Cleanup	Removing the camera from the computer

HT.GT.01

TC_ID	**HT.GT.01**
Purpose	GPS Test
Requirements	3.2.4
Priority	High
Estimated Time Needed	5 Minutes
Dependency	GPS module, computer, mobile device
Setup	Program to read GPS data

Procedure	[A01] GPS module connects to computer
	[A02] Running the program to read the GPS data
	[A03] Reading the GPS data
	[A04] Opening the map application on the mobile device and obtaining the location information
	[V01] Determining the accuracy by comparing the data received from the GPS with the current information of the mobile device
	-
Cleanup	Removing GPS module from the computer

AST.STS_CRAT.01

TC_ID	**AST.STS_CRAT.01**
Purpose	Sending commands to the drone
Requirements	3.2.4
Priority	High
Estimated Time Needed	2 Hours
Dependency	Drone, Mission Planner
Setup	Mission Planner setup, completed drone
Procedure	[A01] Open the SDK code
	[A02] Come to the change mode function
	[A03] We put the drone in guided mode
	[A04] Check if it switch to sent mode using this function
	[A05] We enter arm-disarm command via SDK, then we observe on simulation
	[V01] Commands were able to be sent to the drone correctly.
Cleanup	Stop the code from running

AST.STS_ATT

TC_ID	**AST.STS_ATT.01**
Purpose	Checking if the drone is receiving the sent commands correctly
Requirements	3.2.4
Priority	High

Estimated Time Needed	30 Minutes
Dependency	Drone, Mission Planner
Setup	Mission Planner setup, completed drone
Procedure	[A01] Open the SDK code
	[A02] Altitude command is entered to the drone
	[A03] Take off function is called
	[A04] Observe whether it reaches the desired meter.
	[A05] Then we call the manual control function and check how fast it goes with the values we give.
	[V01] The drone has correctly executed the entered commands.
Cleanup	Stop the code from running

AST.STS_CRTDD

TC_ID	**AST.STS_CRTDD.01**
Purpose	Tests drone receives commands from different devices.
Requirements	3.2.4
Priority	High
Estimated Time Needed	5 Minutes
Dependency	Drone, Mission Planner, Jetson
Setup	Mission Planner setup, completed drone
Procedure	[A01] RF antennas are attached to the drone
	[A02] Simulation tests and RF antenna tests are done on PC.
	[A03] Jetson is mounted on the drone.
	[A04] Different devices are plugged in and tried
	-
	[V01] The drone successfully receives commands from different devices.
Cleanup	Stop the code from running

AST.AFT_PID

TC_ID	**AST.AFT_PID.01**
Purpose	Tests that checks whether the drone vibrates and shakes in the air.

Requirements	3.2.4
Priority	High
Estimated Time Needed	20 Minutes
Dependency	Drone, Mission Planner
Setup	Mission Planner setup, completed drone
Procedure	[A01] The drone is lifted by remote control in windless weather .
	[A02] It is put into Autotune mode.
	[A03] The drone makes its own automatic pid.
	[A04] When finished, the drone is lowered. Drone memorizes what has been done.
	[A05] Again, the drone is lifted and the flight is made to check the settings.
	[A06] If it moves aggressively and its braking is hard, we increase the hardness value of the autotune and repeat the test. If it is soft, we lower the value and repeat the test.
	[V01] The test of the drone to go without vibrating and shaking in the air has been successfully completed.
Cleanup	Stop the code from running

AST.AFT_SDK

TC_ID	**AST.AFT_SDK.01**
Purpose	After the tests in the simulation are over, SDK commands are tested again in the flight test.
Requirements	3.2.4
Priority	High
Estimated Time Needed	30 Minutes
Dependency	Drone, Mission Planner, Jetson
Setup	Mission Planner setup, completed drone
Procedure	[A01] After the tests in the simulation are over, the commands we apply on the SDK are tested again in the flight test.
	[A02] Extra RF antenna is attached.
	[A03] RF antenna works as if there is a cable in between
	[A04] After the simulation tests are finished, the functions there are tested on the real drone.(We can test change mode, arm-disarm command, rtl, etc.)
	-

	[V01] The commands we applied on the SDK were successfully retested in the flight test.
Cleanup	Stop the code from running

AST.DTM

TC_ID	**AST.DTM.01**
Purpose	It is for testing the correctness of the direction of the motors on the drone.
Requirements	3.2.2
Priority	High
Estimated Time Needed	10 Minutes
Dependency	Drone, Mission Planner, Jetson
Setup	Mission Planner setup, completed drone
Procedure	[A01] Open Mission Planner
	[A02] Go to the engine test tab
	[A03] We turn each motor one by one in that tab and look at the motor direction (without the propeller attached).
	[A04] If the motor direction is wrong, we swap two of the motor's 3 wires.
	-
	[V01] If the motor direction is correct, the test is complete
Cleanup	Close Mission Planner , stop the engines

AST.CTG

TC_ID	**AST.CTG.01**
Purpose	Tests the drone can stand without swaying in a windless area.
Requirements	3.2.2
Priority	High
Estimated Time Needed	10 Minutes
Dependency	Drone, Mission Planner, Jetson
Setup	Mission Planner setup, completed drone
Procedure	[A01] It’s done with a water gage.

	[A02] Go to Mission Planner.
	[A03] In order for the autopilot to stay in balance, tests are performed by turning the spirit level straight, right, left, nose down, nose up, completely upside down, respectively, in the calibration tab.
	-
	[V01] Drone can stand successfully without swaying in a windless area.
Cleanup	Close Mission Planner , stop the code

AST.COMCT

TC_ID	**AST.COMCT.01**
Purpose	Determine which direction the drone is heading.
Requirements	3.2.4
Priority	High
Estimated Time Needed	10 Minutes
Dependency	Drone, Mission Planner, Jetson
Setup	Mission Planner setup, completed drone
Procedure	[A01] Go to Mission Planner.
	[A02] Calibration is started via the compass calibration tab
	[A03] The drone is rotated to move in 3 axes
	[A04] After the calibration level bar is filled, the autopilot compass calibration is completed.
	[V01] We can successfully determine which direction the drone is heading.
Cleanup	Close Mission Planner

AST.CCT

TC_ID	**AST.CCT.01**
Purpose	Tests using different controllers or introducing a new controller to the drone.
Requirements	3.2.4
Priority	High
Estimated Time Needed	5 Minutes
Dependency	Drone, Mission Planner, Jetson
Setup	Mission Planner setup, completed drone
Procedure	[A01] Go to Mission Planner.

	[A02] Navigate to the control calibration tab on the Mission Planner.
	[A03] Press the radio calibration button
	[A04] All functions and keys of the remote are set to min and max values.
	[A05] Radio calibrate complete button is pressed
	[V01] Controller calibration is done successfully
Cleanup	Close Mission Planner

14.8 TEST RESULTS

Individual Test Results

TC ID	Priority	Date Run	Run By	Result	Definition
CV_ST.ODT.FPS.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	FPS is low for initial tests, but improvement continues.
CV_ST.ODT.AC.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	Successfully recognizing objects.
CV_ST.ODT.DS.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	Detects how far the object is and does not approach the object after a certain distance.
CV_ST.ODT.CD.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	It can detect the color of the object successfully.
CV_ST.CAT.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	The redirect command has been issued successfully in 100 milliseconds.
HT.SPT.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	The signal was successfully received when the toleranced distance of 500 meters was reached.
HT.FST.01	H	26.05.2022	Ahmet Çetin Türkyener	Pass	Under the desired conditions, the avoidance movement of the drone according to the detected obstacle was successfully achieved. It flies at a constant speed of 4 m/s.

HT.BHT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	Since the voltage values of the battery cells are close to each other, the batteries are healthy.
HT.VT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	The system was able to continue to operate in the same way when power was supplied after the drone had passed through rough roads.
HT.FTT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	It flew for 10 minutes on a single battery.
HT.CT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	Camera is working.
HT.GT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	Position accuracy can be achieved with tolerances of +3 or - 3 m.
AST.STS_CRAT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	The drone can detect commands. Tried with Mission Planner as well.
AST.STS_ATT.01	H	26.05.2022	Elif Yağmur Eratalay	Pass	Verified with test drone, the drone can go towards given directions.
AST.STS_CRTDD.01	H	26.05.2022	Songül Meryem Özbilen	Pass	It also worked successfully with the test drone as a different vehicle.
AST.AFT_PID.01	H	26.05.2022	Songül Meryem Özbilen	Pass	Different pid values were entered in trials, and the optimum was determined. P value and i value were determined as 0.135 and d value as 0.301.
AST.AFT_SDK.01	H	26.05.2022	Songül Meryem Özbilen	Pass	The commands we applied on the SDK were successfully retested in the flight test.
AST.DTM.01	H	26.05.2022	Songül Meryem Özbilen	Pass	Motor direction has been tested correctly.

AST.CTG.01	H	26.05.2022	Songül Meryem Özbilen	Pass	Drone can stand successfully without swaying in a windless area.
AST.COMCT .01	H	26.05.2022	Songül Meryem Özbilen	Pass	We can successfully determine which direction the drone is heading.
AST.CCT.01	H	26.05.2022	Songül Meryem Özbilen	Pass	Controller calibration is done successfully.

15.CONCLUSION

This document includes wide information about our project that titled as "Autonomous Drone Control". In this project, we developed possible obstacle models for drones using Image Processing and Artificial Intelligence techniques, and enabled the drones to overcome these obstacles with autopilot and reach the target location.

For this study, we used the previously researched PyCharm IDE, LabelIMG, Cloud GPU programs and the YOLO reference algorithm method. We tried to achieve a high confidence score in object detection. As a possible solution to the related problem, we expanded the dataset and added different angles to the visuals. When we chose objects such as triangle, circle, square for our datasets, we had difficulty in training because they had few features. That's why we changed our objects to laptop, human, car. We collected real-life data and labeled it using labelImg. Then we embedded this train data into our yolo algorithm for training. Thus, we have introduced our objects to our algorithm.

After completing our training, we completed our tests on Computer Vision Software, Hardware and Autopilot Software. As object detection test, we tested features such as how long does it take to detect, accuracy rate and the detection of the detected object at a specified distance. As hardware tests; we tested signal performance, flight speed, battery health, vibration, flight time, camera and GPS. As Autopilot Software Tests; we tested SDK test for simulation, command receive accuracy, axiom truth and command receiving test from different devices. We also successfully performed PID, Flight, SDK, Calibration tests. Also, as a result of the tests, we preferred to use a battery in which cells are used side by side, since the presence of many cells on top of each other upsets the balance of the drone.