# Object Detection Drone

Kristian Aspi, Jacob Bendele, Hussain Mohammad, and Anthony Pionessa

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This project involves a modular quadcopter drone with object detection capabilities. It involves the selection and construction of hardware, custom CAD modeling and 3D printing, custom PCB designs, and the development of a corresponding phone app for the pilot. All these aspects of the project encompass both CPE and EE knowledge.**

***Index Terms* — Artificial intelligence, CAD, drone, electrical engineering, image classification, image processing, neural networks, quadcopter, research and development**

## I. Executive Summary

The first use of the term drone to describe an unmanned radiocontrolled aircraft dates back to as early as the 1920's. These early drones were not like what one imagines flying through the sky today. In fact, they were full sized planes used primarily for target training by the military. Drones have come a long way since then and have cemented their form in a 4 rotor design called a quadcopter. These quadcopter designs can range from sizes small enough to fit in the palm of your hands to sizes large enough to need both hands to carry. The quadcopter blueprint rapidly developed when electronics were able to keep up with the lightweight, cheap, sensor packed boards used for flight control. These flight control boards emerged around the mid 2000's and are the brains to every modern drone. Following the development of flight controllers, the consumer drone market really started to take off in 2013 with camera mounted drones directed at film makers.

Concurrent to the drone's development in the past decade, computer vision has been making its rise as a forefront topic in research and industry. Computer vision publications have been rising rapidly especially in fields, such as object detection. Just last year the amount of publications containing Object Detection in the title surpassed 2000.[2] Companies like Uber and Tesla are funding and developing advanced computer vision systems for cars in a race to create fully autonomous vehicles. Even in your local retailers you might see security monitors tracking your face as you walk through the doors. So what does this all mean?

To us, drones are a modern technology that have only broken into the consumer market as of very recently. This provided us the ability to explore creating a drone in an era where there is a lot of previous research and information, but still many more new things to discover. Additionally, computer vision has seen its own rise in popularity in today's industry. Computer vision and object detection have been featured in drones before, but there are many problems that can still benefit from this duo. We want to start from the ground up, building a drone, computer vision detection system, and pilot application.

The goal was to utilize our drone system to count objects and relay them back to the pilot via a phone application. Some examples of objects we proposed are cars, people, boats, etc. The detection systems currently on the majority of consumer drones are meant for object avoidance and tracking people for photography. By using the drones aerial view capabilities we can capture video and images of the scene. The proposed detection system will utilize these images and videos to detect and count objects. The pilot application will present the live camera view for navigation. Once the camera is situated over the scene the pilot will see bounding boxes around the detected objects in the live view.

## II. Objective

Overall our Object Detection Drone Project consisted of four subsystems: The Drone Subsystem, Flight Subsystem, Power Subsystem, and Detection Subsystem. In addition to the four subsystems above, there is also a companion smart phone application for the pilot. The objectives for each of the subsystems as well as the pilot companion app will be reviewed here.

The Drone Subsystem encompasses all the physical design of the Object Detection Drone. Our objective for the chassis was to design a lightweight, medium sized aerial drone based on the well-known quadcopter blueprint. The Drone Subsystem is capable of carrying a camera as a payload. Additionally, the physical structure of the drone adheres to a modular design supporting swappable hardware components. By considering modularity as one of our objectives we created an easily serviceable prototype when it comes to the battery, propellers, and motors.

Our Flight Subsystem consists of the brains and flight controls of the Object Detection Drone. Our number one objective with this subsystem was simply to get the drone off the ground under its own power. Getting off of the ground can be broken down into three main tasks: takeoff, flight, and landing. Within these tasks we had smaller, but equally important objectives for the Flight Subsystem. Ultimately during takeoff and landing we wanted to make sure the drone does not suffer any catastrophic damage. While our goal for both takeoff and landing pertain to not damaging the drone, our inflight goals pertain to the flyability of the drone. To us flyability consists of the drone reacting to the commands sent to it, as well as the drone's ability to hover and maintain altitude.

Responsible for powering the entire project is the Power Subsystem. Our objective for this subsystem was to design a reliable powering method for all of the electronic components of the drone. This subsystem can be broken down into two major parts. The first of the two parts that compose the Power Subsystem is the power source. An objective regarding the power source is to make sure it was sufficient for the power draw of the drone electronics and flight times we are looking to achieve. The second part is the power distribution. We made sure the distribution for the power source is accurate for components being driven, as well as its ability to handle power draw for all systems onboard the drone.

The last drone system to discuss is the Detection Subsystem. When taking a look at the consumer market almost every drone has an integrated camera for taking still images as well as video. This is shared with our design, but our main objective for the Detection Subsystem was to pursue utilizing the affixed camera for a more interesting use case; object detection! The computer vision implementation is what sets our drone apart from many of the others found on the market. While other drones have gone fully into the photography and videography realms with hovering for a photo or tracking a face, we wanted to leave our design more open ending. By having the ability to train the detection model on many classes our drone is extensible in the types of objects detection tasks. This system is composed of the camera as well as the transmission hardware necessary to relay the video from the drone to the pilot companion app. Our main objective with this subsystem was creating a stable video transmission. Without a stable video transmission our drone pilot will be flying by sight only and our detection software would have no input to run against.

In addition to the drone our project also has a companion smart phone application for the pilot. This application showcases a live feed view from the camera of the drone. This live feed is processed using the phone's resources to implement the object detection software. The phone application shows bounding boxes around detected objects, object count, and in the case of multiclass object detection, assign a color corresponding to class type.

## III. System Components

Before a system or project starts the design process, requirements specifications were established and detailed enough to make analysis and design possible.

### *A. Frame*

Frame size measured from two opposing motors fits the size of a medium to large drone.

### *B. Flight Controller*

The flight controller acts as the pilot of the drone which handles flight response by controlling the direct RPM of each motor in response to input.

### *C. Electronic Speed Controller*

Four Electronics Speed Controllers (ESC) were needed. The ESC accepts a DC input voltage and produces three out of phase voltages that feed the motor's inputs to control and regulate the speed of an electric motor.

### *D. Motors*

Four brushless motors were used and is the standard for drones our size and capabilities. Brushed motors, though cheaper, are not as efficient and wear out quickly.

### *E. Propellers*

Standard propellers were bought, and size was determined by the estimated frame size, weight, motors, and desired capabilities.

### *F. Remote Controller*

A standard bought controller bought which is ideal for reliability and precision for control. An attachment to mount a phone on the controller was made via 3D printing. The phone displays the aerial feed from the camera systems on the drone. The physical controller allows more precise control of the drone.

### *G. Power Distribution Board*

The PDB breaks out the power source into multiple different connection points for cleaner wiring and build. Additionally, provides DC-DC conversion for components that do not operate at power source voltage.

### *H. Battery*

Our battery pack sustains all the drone functions from motors to sensors. The size of the battery depended on desired flight distance, desired flight speed (dependent on motor power), flight controller and electronics, and weight.

*I. Camera*

Used to get point of view from drone for video processing and flight navigation. It is mounted in a slight angle towards the ground for the best image processing view.

*J. Remote Controller*

Allowed transmission of video feed from the drone to the receiver on the ground. Video is from the camera onboard the drone.

*K. Camera Detection System*

The pilot companion application utilizes the host phone's computational power to implement the detection algorithm. The system has to see the lower surroundings of the drone while hovering and detect obstacles (people or otherwise) via Artificial Intelligence/ Machine Learning. We utilized an open source detection program.

*L. Phone Display Feed and Receiver*

The user's phone displays the camera feed from the drone in a landscape style that occupies the entirety of the user's phone screen. The receiver is connected to the phone via USB. On the camera feed is a visual representation of what the drone is seeing through the use of Artificial Intelligence/Machine Learning.

*M. App*

The app provides a simple interface for object detection using the drone's aerial video on a mobile device. It offloads machine learning object detection tasks to a background process, while providing a clean UI for metrics and video management.

*N. Firmware*

The multitude of hardware components that were needed required parameter tuning and integration together. This was done via programming parameters into the hardware's firmware. Additionally, by utilizing protocols between the hardware devices to communicate data between them.

*O. Stretch Goal: GPS Beacon*

A late project stretch goal to solve the issue of crashing, losing, or having the drone stolen. The GPS Beacon consisted of an ATmega2560 MCU, GT-U7 GPS module, and a GSM SIM800C cellular module. A SMS message would be sent to the cellular module to ping the drone's location. Due to time and commercial constraints, this GPS Beacon remained a stretch goal and never came to fruition. However, a proof of concept utilizing an ATmega2560, GT-U7, and HM-10 BT module was crafted.

## IV. Project Hardware Design

*A. Drone Subsystem*

The Drone Subsystem describes the physical aspects of our drone. This mainly includes the frame and mounts the electronic components are attached to.

We decided on incorporating the H shape design for the frame mainly due to its larger body size and its simple build. The larger size is necessary to mount our selected 3300 mah lipo battery, which is relatively long, along with other components.

The frame is constructed out of 1/2" Schedule 40 PVC with and Tee fittings. PVC pipes are strong, relatively lightweight, and cheap. The Tee fittings also allowed us to incorporate the two main frame designs easily. The pipes were pressed fitted into the Tee fittings and proved to be strong and sturdy, without the need of any additional adhesives.

The body of the drone and the motor mounts are custom designed via SolidWorks and 3D printed with PLA material. The body of the drone consists of two plates where the top plate is attached above the bottom plate by standoffs. The entire assembly is then attached to the PVC frame body by two strips of double sided tape along the vertical pipes of the frame.

The components the bottom plate mounts are: the PDB, flight controller, video transmitter, camera, and controller receiver. The top mounts the Lipo battery and is secured down by cable ties.

The 3D printed body plates and motor mounts were designed to be modular, making it easy to assemble and disassemble.

*B. Flight Subsystem*

The brain of the Object Detention Drone is our Naze32 Full Version Flight Controller. The flight controller receives signals inputted on the remote controller and transmits user inputs to the electronic speed controllers which control the RPM of each motor to perform the desired maneuvers. The Naze32 is based on a 32-bit STM32 Microcontroller that runs at 72MHz. It is an F1 processor meaning it offers 128KB. The Full version of the flight controller is superior to its other version in terms of higher quality sensors. It includes a MS5611 Barometer, HMC5983 Magnetometer, and an MPU6500

IMU which consists of a 3-axis accelerometer, and a 3-axis gyroscope.

*C. Power Subsystem*

We chose the HRB 4S 14.8V 3300mAh lithium polymer battery as the power supply for the Object Detection Drone. Using this battery as our power platform allowed us to achieve our desired flight time as well as sustain the power draw of all of the components. In order to be confident in our power source's capabilities the design choice was modeled by calculations.

To ensure our power source design met our engineering requirement of a 5 minute or greater flight and hover time we need two variables. The first is the capacity of the battery and the second is the total current draw of all of our components. We know the capacity of our battery is 3300 mAh, but to determine our current draw we will need to sum all of the components current draw.

Using the weight determined of 945.20 grams and the fact that we used 3.1 pitched tri-blade propellers we determined the current draw of each motor. The current references are located in the manufacturer's datasheet for our EMAX RS2205-S motors. Determining the reference current we first calculated the thrust that each motor needed to produce for hovering. This was done by simply dividing the gross weight of the drone by four. This came out to roughly 237 grams of thrust per motor. In the manufacturer's reference we were told that the motor draws 5 amps per every 326 grams of thrust. Proportionally, we draw close to 4 amps to reach our 237 grams per motor of thrust. The Mobius action camera is run from a 5V USB cable. The standard for USB 2.0 stated that a compliant port is able to provide 500mA and testing showed that the Mobius while outputting video and not recording draws about 250mA. The Naze32 flight controller has a 5V input, but testing showed on average it drew about 150 mA. The video transmitter at our intended use of 25mW setting draws a maximum of 120mA according to its data sheet. Our last major component, the FlySky receiver drew about 300mA. For our modeling since many of these figures are max draws or tested measurements during intended use we took the ratings at their face values.

$$16.82A = 4A \times 4 + 250mA + 120mA + 150mA + 300mA \quad (1)$$

Our major components at hover draw a total current of 16.82A. Using the lithium polymer battery's rated capacity of 3.3Ah we can determine that flight time as follows:

$$11.77\ min = 3.3Ah / 16.82A \times 60min \quad (2)$$

(1) and (2) show that the power source is sufficient for roughly a 11.8 minute flight hover time which met our engineering requirements.

A quick glance at the manufacturer's reference table showed that the motors at full throttle draw 33.6A. Given our quadcopter design this current had to be multiplied by four resulting in a peak current draw at full throttle of 134.4A for the EMAX RS2205-S motors. The HRB 4S 14.8V 3300mAh lithium polymer battery had a 60C discharge rate. Therefore we verify that our power source is able to handle these bursts without damaging itself or other components.

$$198A = 60C \times 3.3A \quad (3)$$

(3) gives our power source head room when under full throttle conditions of 134.4A load.

One of the most critical components in a drone is the battery as it serves as the power source that supplies power to all components on the drone. We altered the size and weight of the drone in order to conserve the fuel and extend the flight time of the drone for the specific battery. The conservation of fuel through design changes was just as important as the right distribution and management of the battery. For this purpose, a Power Distribution Board was used.

The Power Distribution Board divided the electrical power feed into subsidiary circuits as well as provided a protective fuse and circuit breaker for the safety of the drone. We are using a separate Power Distribution board, with the ESC and Flight Controller connected to it.

A PDB is a simple basic circuit board which connects all of the ground connectors to each other and connects all positive connectors to one another allowing specifically designed power flow to all of the components in the drone in an organized manner.

Voltage Regulators were used on Power Distribution boards which are also known as Battery Eliminator Circuits (BEC). It regulated the voltage from 14.8V (in the case of a 4S Lipo) to 5V as per requirement.

It also regulated and channeled the voltages coming from the battery. As we were using a 4S Lipo battery which gives us ~14.8 Volts, the power Distribution Board converted the 14.8V into 5V or 12V per the requirement of the component.

The Main components which required power in a drone and power is delivered to through the Power distribution board was the ESC for each motor, flight controller, camera and the Receiver/Transmitter. This was very useful as otherwise all the components on the drone should be working on 14.8 V, i.e. the voltage delivered by the battery.

The power distribution board in our Object Detection Drone was our own design including a 5V regulator. The design distributes the power source for easy connections and clean wiring. It was designed to withstand the large current ratings that can occur when the drone is in flight.

The PDB was designed in a way such that the ESC's can be connected to the pads on the PDB in an organized manner. This was done by placing two sets of solder pads for Vin and GND on either side of the board. That way the left side and right side can be connected to two ESC's each.

Moreover, an extra 5V output was added in the design for open-endedness with a set of solder pads (5V and GND) at the front of the boards.

### *D. Detection Subsystem*

Our camera system runs independent from our drone system due to differences in the transmission frequency utilized to transmit the video signal and the transmission frequency utilized to transmit the controller signal to and from the drone. For the camera itself, we went with the Mobius action camera.

The Mobius action camera does not have to be connected to the battery of the drone since the camera itself has an internal battery that can be charged and can stay charged for around eighty minutes.

The Mobius action camera does not have the ability to connect to the video transmitter with the contents that come with the purchase of the camera. The manufacturer of the camera also creates a breakout cable that we can connect to the camera which splits into different cables with the ability to connect them to a transmitter. This breakout cable was modified to fit our needs to connect the camera to the video transmitter.

This cable splits up the mini USB connection into three different connections, one for video output, one for audio output, and one for a voltage source. In the USB connector on the Mobius action camera, the manual states that the connector pin four is used as a sense pin and then the connector pin five is a ground pin.[1] If both of those pins are connected to each other, then the connector pin two will be switched by a builtin USB switch to be the video out pin and the connector pin three will also be switched to be the audio out pin. The breakout cable provides the needed connection to switch the USB connector pins and then also provides access to the cables which we modified as needed to fit the video transmitter we decided on.

We decided that the video transmitter that we used for our project was the Eachine TX805 transmitter. This video transmitter provided an area to connect the wires from the Mobius action camera to the transmitter. Excluding the connections for the transmitters power, a voltage in and a ground connection, the connections that were provided are an audio in connection, a voltage in connection from the camera, a ground connection form the camera, and a video in connection. All of the connections were soldered in place. The Eachine TX805 transmitter also included a connection for an antenna built onto the transmitter where the antenna is mounted.

For transmitting and receiving the controls to the drone, we decided on using the Flysky FS-i6X controller with a matching receiver. The Flysky FS-i6X controller itself provided us with a choice of six to ten channels which we utilized for different commands and other actions we want to transmit to the drone. The controller uses a frequency of 2.4 GHz which should not interfere with our video transmitter signal.

The receiver that was included with the Flysky FS-i6X controller was the Flysky FS-iA6B receiver. The receiver provided the ability to receive six channels which is the lowest amount of channels that the controller was able to utilize. The receiver came equipped with a dual antenna set up which assists with the combating of signal interference and helped provide a more stable signal connection.

The receiver that we used for our project was the Eachine ROTG02. How we connected the video receiver to the phone of the pilot is through a simple USB to mini USB or USB C cable with the USB connection on the receiver.

.

## V. Project Software Design

The software design for this project consisted of two main individual areas of focus and one auxiliary.

### *A. Mobile Application*

The mobile application or as we refer to it Pilot Companion Application or Pilot Application, is designed to ingest the aerial video footage from the drone and perform object detection tasks on it. The application runs on an Android based smart phone. The Pilot Application is designed, such that its main functionality can be used in a "set it and forget it" style. This allows the Pilot to focus on safely piloting the drone instead of how to configure, launch, and set up object detection for their flight. This safety was a key consideration during the software design of the Pilot Application. Therefore, when the application is launched it initializes a detector and file monitoring background service.

The file monitoring service is what allows the app to programmatically check when a video is saved from the drone's flight. This is accomplished via a set folder structure, such that the video receiver will always save a flight video to the same location. Once the file system observer recognizes a valid video, it will begin to parse the video file into frames. These frames are what are ultimately fed into our detector.

Our current detector is based on the popular one-stage architecture. More specifically, our detector model is the COCO SSD MobileNet v1 which is optimized for the processing resources offered on mobile platforms. By utilizing a single-stage detector, the Pilot App was able to benefit from the inference speed advantage common to them. This advantage allows the application to reach real time levels of performance in object detection tasks. Additionally, our model is derived from the Single Shot Multibox Detector (SSD) [3]. SSD has two special properties that make it inherently better for the task of aerial object detection:

1. Default Bounding Boxes: SSD utilizes default bounding boxes with different aspect ratios and scales. Each feature map cell is tiled with these bounding boxes such that they are fixed in relation. The feature map cell position is then used when predicting offsets.
2. Scaled Feature Maps: Utilizing outputs at different levels of the network provide scaled variations of the feature map. These scaled variations can then be used to make predictions for different scaled objects.

What these two features mean, is that our detector handles different aspect ratios, scale, and other warped perceptions of images well. These cases of warped perspective, size, and ratio arise often when taking photos from odd angles; especially angles from an aerial point of view.

The detector is stored as a file of weights that will ultimately be computed against each incoming frame/image. To handle the initialization, optimization, and translation of the model into a usable element for our mobile application we leveraged the TensorFlow Lite library. This is an optimized, mobile focused version of the very popular machine learning library TensorFlow.

Moving forward from the architecture, the detector is configured for object detection of people. Other objects can be configured either via training or setting available labeling data. As previously mentioned, the detector is launched from the background service after a new drone video has been created. Upon being fed the parsed frames from the drone video, our detector will bound each area of the frame/image it infers a person is in with a red rectangle referred to as a bounding box. This bounding box is only placed when the detector has a confidence threshold of 50% or greater. After the detection of all of the frames they then need to be converted back into a video for convenient playback. This conversion is accomplished via the FFmpeg library. The Pilot Application utilizes this library by organizing the stream of detected images for encoding back into a video file.

To interact with all of the processing that is happening in the background it was key that we implemented an easy to use front end for the Pilot.

Our background services in Android are launched on a separate thread allowing for the pilot's phone to continue using other applications. By using background services the Pilot Application still is provided access to the applications shared resources. This means we can display convenient notifications called Toasts for our Pilot Applications' current state regardless of the state of the phone. These notifications are the chosen method for relaying the current state of processing: Started, In Progress, and Finished.

When launching the Pilot Application the user will be able to see a list of previously detected videos. If a video finishes processing it will be seamlessly added to the list. If a video was made in error or is no longer needed a convenient UI for deleting videos is available.

Upon selecting a video from the list the user gets to watch the playback of the detected video. Stats about the detector such as the max count of objects detected, filename, threads used, and inference time are displayed.

*B. Firmware Tuning*

Many of our hardware components ran some form of custom or open source firmware. These firmwares required configuration, tuning, and tweaking. Unlike the application development, this process was solely testing driven. Small iterations and changes were made for multiple purposes. Noteworthy of which are:

1. Trimming the controller's inputs: without modifying the cutoff points of the signal the flyability of the drone was almost non-existent. Channels for throttle, roll, pitch, and yaw were jittery and inconsistent causing the drone to drift around and behave erratically.
2. PID Tuning: Flight controllers use a PID loop to control the motor output based on external factors, input, and error. Originally our PID loop was considering our controller input very heavily. This made the drone sensitive to even the slightest input from the controller. This oversensitivity resulted in twitchy flight and many small crashes. By programming these parameters we were able to tune out these unwanted behaviors.

*C. Stretch: GPS Beacon*

The proof of concept software for the GPS Beacon was written in Arduino's subset of C/C++. Design required initializing serial connections between the GT-U7, HM-10, and ATmega2560. Once established we initialize the GT-U7 GPS module and wait for it to get a satellite fix. The programming loop then reads incoming commands sent to the HM-10 Bluetooth module. If the command matches our required “location” command then the drone is pinged and we receive the latitude and longitude. This process would have been almost identical in a finalized version of the GPS Beacon with a cellular enabled communication interface.

## VI. Testing And Results

### *A. Drone Subsystem*

The drone subsystem is designed to be strong and study as well as modular.

To test the modularity of our motor mount system, we timed the assembly and disassembly process. The aim was to have the process take under 5 minutes, and it did.

We also did a crash test twice, once on purpose and the other time by accident. The first scenario was a purposeful collision towards a fence. The second was an accidental shutdown of the drone in midair. Besides some breakage of the 3D printed plates, all the components and entire frame was completely fine. The ESC’s and most of the wires were protected inside the PVC pipes.

### *B. Flight Subsystem*

To test the flight subsystem, we first tested the connection between our flight controller and motors. We connected the flight controller to an open source flight controller software, CleanFlight, on a laptop via microUSB. This also supplies power to the flight controller. We then connected the ESC wires to the flight controller and powered a brushless motor by an external power supply. Through CleanFlight, we were able to run the motor and confirm the desired spin orientation. If the motor were spinning in the wrong direction, switching the orientation of two of the soldered wires of the ESC was the only thing needed to be done. This was done on all four motors.

### *C. Power Subsystem*

The testing of our power subsystem was simple after getting all the SMD soldering done onto the PDB. We connected the external power supply to one of the pads on the PDB and confirmed that it outputs the same on the other motor pads by using a digital multimeter. We tested using a 16.8 voltage output from the power supply as it is about the voltage output that comes from our lipo battery. We then tested the outputs of the 5V voltage regulator pads and confirmed it was outputting correctly.

### *D. Detection Subsystem*

To test the detection subsystem, we first tested the connection between the camera and the video transmitter. To test the connection we temporarily connected the video transmitter to a 14 V power supply to ensure that it was capable of handling the amount of power that the battery would supply. We then temporarily connected the video transmitter to the camera using the appropriate cable connections that we established in our detection subsystem design to ensure that the camera receives power through the transmitter. When both received the appropriate power, a LED illuminated on both devices.

After we established that the connection between the camera and the video transmitter was working as intended, we then tested the connection between the video receiver and the phone using a USB to USB C cable. After connecting the receiver to the phone, we were able to check the phone for an external device connection. We were also able to open the Eachine mobile application which displayed the current frequency that the receiver is set to receive.

After testing the receiver we were able to keep the receiver connected to the phone and also turn on the transmitter and camera combination which starts transmitting the video feed from the camera automatically. To test the connection between the transmitter and receiver, we had to have the receiver search a preset range of frequencies for the frequency that the transmitter was using. After they were paired, the video feed from the camera was shown in the Eachine mobile application and that proved that the camera, video transmitter, and video receiver were able to be connected together and operated as intended.

To test the drone controller and receiver, we had to connect the flight controller to the computer and also to a 5 V power supply since the USB connection is only able to power the flight controller and not the receiver. We then connected the receiver to the flight controller via the appropriate cable connections we established in our detection subsystem design. After all of the connections were made, we opened the drone software on the laptop to establish the connection to the flight controller. To pair the drone controller and receiver, we had to perform a particular series of controller inputs that then pairs the two devices. When everything was successful, the drone model in the computer software responded to the controller inputs by shifting depending on the different inputs.

## *E. Detection Performance*

Initially testing the detector started with single images taken from the Mobius action camera. Accuracy was measured as:

$$TP / (TP + FP - Duplicates) \quad (4)$$

Where TP are true positives, FP are false positives, and duplicates are duplicate TPs. Performance accuracy was above or equal to our specified 80%. However, the detector will infer duplicates and our code does not account for these, therefore we subtract them from the total detections. Since videos are simply streams of images our accuracy metrics continued over into vide detection..

# VII. Administrative Content

For the budget of our project, we decided to set the budget at $500 or less due to the project being self-funded. We did reach out to some potential sponsors, but we did not receive any responses. In total, all of the components for the drone cost us $411.04 which was below the budget.

Throughout the summer and fall semester, we were able to adhere to our self-assigned timeline with little deviation. There were only a couple of instances where we did not meet our initial due date, but we never had to push the date back more than a week.

# Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Samuel Richie and Dr. Lei Wei over the course of Senior Design I/II as well as the professors in UCF that teach for a passion and guide us students to understand and learn concepts from their respective fields.

# References

[1] V. LaManna, "Mobius Video/Audio-Out Cable," Mobius-actioncam.com. [Online]. Available: https://www.mobius-actioncam.com/store/products/videoaudio-out-cable/. [Accessed: 28-Jul-2020].

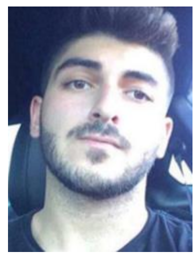
[2] Z. Zhou, Z. Shi, Y. Guo, et al. 2019, Object Detection in 20 Years: A Survey, arXiv:1905.05055 [cs.CV], (May, 2019). This work has been submitted to the IEEE TPAMI for possible publication.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European conference on computer vision. Springer, 2016, pp. 21–37.

# Biographies

**Kristian Aspi** is graduating Computer Engineering major from the University of Central Florida. He plans to pursue a career in computer engineering.

**Jacob Bendele** is a graduating Computer Engineering major from the University of Central Florida. He is currently pursuing software engineering opportunities and plans to relocate to California.



**Hussain Mohammad** is a graduating Electrical Engineering major from the University of Central Florida and plans to pursue a career in electrical engineering.

**Anthony Pionessa** is a graduating Computer Engineering major from the University of Central Florida. He has accepted an Associate Cyber Information Assurance Analyst position at Northrop Grumman to start at the beginning of 2021.