Clark Aerospace

Clark College Aerospace Program
AUVSI SUAS 2018 Technical Design Report

Abstract

This paper describes the development and operation of the Clark College Aerospace Program unmanned aerial system for the 2018 AUVSI SUAS competition. The intent was to use experience from last year's competition to build a completely new aerial vehicle capable of completing all the mission objectives as per the AUVSI SUAS 2018 rules. Using resources from Clark College and local supporting businesses the team was able to design and build a completely new modular multirotor UAS for the purposes of autonomous flight, object detection, classification and localization and aerial delivery. Throughout the year the team constructed a robust aerial platform consisting of a mix of commercial components and custom-engineered 3D printed parts to create the UAS described in this technical design report.

Table of Contents

List of Acronyms

AES	Advanced Encryption Standard
AUVSI	Association for Unmanned Vehicle Systems International
ESC	Electronic speed controller
FAA	Federal Aviation Administration
GCS	Ground control station
IMU	Inertial measurement unit
kNN	k-Nearest Neighbor
L2	Euclidean Distance
LiPo	Lithium polymer
ODCL	Object detection, classification, and localization
OSC	Open sound control
R/C	Radio control
ROI	Region of interest
RRT	Rapidly-Exploring Random Trees
SUAS	Student unmanned aerial systems
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UAS	Unmanned aerial system
UBEC	Ultimate battery eliminator circuit
UDP	User Datagram Protocol
WPA2	Wifi Protected Access

Systems Engineering Approach

Mission Requirement Analysis

Clark College Aerospace Program designed an aerial platform to perform all mission objectives in accordance with AUVSI SUAS 2018 rules and mission requirements. This year's design is an octocopter which is a radical departure from the repurposed power glider used in the 2017 competition. The experience from last year's competition exposed some major design flaws in the previous system which were mitigated with the new rotorwing design. These include payload capacity, flight stability and waypoint capture accuracy. Tasks for the mission were determined and prioritized based on such considerations as experience from the previous year and team member capabilities. Mission objectives were categorized as being Low, Medium or High in priority (Table 1).

Table 1: Mission Objective Priorities

Mission Objective	Will Attempt	Priority
Autonomous Flight	Yes	High
Obstacle Avoidance (stationary)	Yes	Med
Obstacle Avoidance (moving)	Yes	Low
Object Detection, Classification, Localization	Yes	High
Emergent Target	Yes	High
Off-Axis Target	Yes	High
Airdrop	Yes	Low
Interoperability	Yes	High
Cyber Security	Yes	Med

Design Rationale

Based on the team's experience at the 2017 competition, it was decided that a rotorcraft was better suited to the mission requirements and team member capabilities. The design proposed for the 2018 competition was a multirotor design capable of lifting a payload of up to 3 kg. The payload estimate was intentionally high in order to allow for further improvements in the following years without requiring a change in airframe or drive components.

A commercial off-the-shelf frame was chosen in order to reduce time and resources devoted to constructing the aerial platform. This allowed the team to devote more time to designing and testing systems such as autonomous flight, object detection and classification, waypoint capture, interoperability, and other mission-critical aspects of the project.

One significant disadvantage that rotorwing platforms have compared to their fixed-wing counterparts is flight time. Last year's Emperor vehicle was capable of sustaining powered flight for over an hour using two 1300 mAh LiPo batteries. The drive system for the octocopter was designed through the use of the eCalc software application. Flight time was maximized at 20 minutes using four 10,000 mAh batteries (Table 2). The limited flight time necessitates several battery swaps during the mission, which must be accounted for when constructing search grids and mission plans.

Table 2: Battery Specifications

Battery		
Load:	6.46	C
Voltage:	21.39	V
Rated Voltage:	22.20	V
Energy:	888	Wh
Total Capacity:	40000	mAh
Used Capacity:	34000	mAh
min. Flight Time:	7.9	min
Mixed Flight Time:	17.2	min
Hover Flight Time:	19.2	min
Weight:	5928	g
	209.1	oz

The autopilot system remains unchanged from last year. It consists of a Pixhawk PX4 flight controller which interacts with a ground station laptop running Mission Planner software. The flight controller utilizes an assortment of integrated and auxiliary sensors such as GPS, barometer, compass and sonar for fully autonomous flight.

System Design

Aircraft

The UAS was designed using a Tarot T18 1250mm octocopter frame which consists of eight spars sandwiched between two carbon fiber plates with two hardpoint mounts located on the underside of the aircraft. Using the hardpoint mounts, a student-designed and manufactured payload platform (Figure 1) which houses the electronic components necessary for object detection, classification and localization, sonar proximity detection, as well as airdrop components and a communication antenna for image relay.

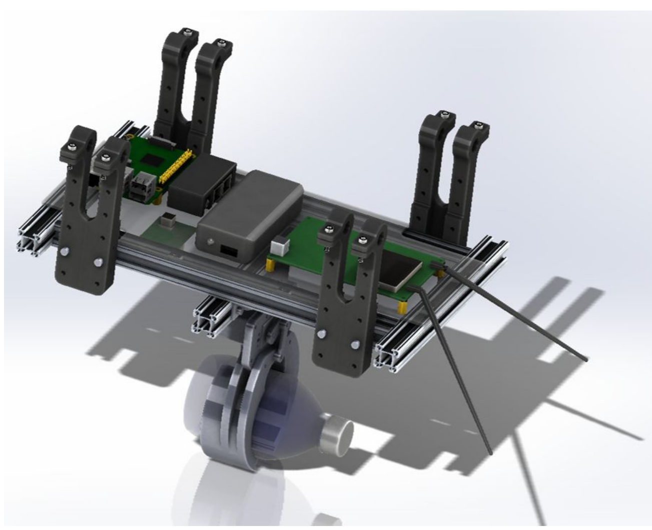

Figure 1: Payload Platform

Another student-designed and manufactured 3D printed platform was created for mounting the Pixhawk flight controller, GPS module, R/C receiver, and telemetry relay antenna on the topside of the aircraft.

The drive system of the aircraft consists of eight Tarot 5008/340kV motors with Tarot 1855 fixed carbon fiber propellers. The motors are controlled by Hobbywing 40A ESCs and power is supplied by four Tattu 10,000 mAh 25 C LiPo batteries. Using eCalc to analyze this prop/motor/ESC combination, the current, voltage, and power requirements of the system were determined (Table 3).

Table 3: Motor Specifications

Motor @ Optimum Efficiency		Motor @ Maximum		Motor @ Hover	
Current:	13.99 A	Current:	32.31 A	Current:	13.26 A
Voltage:	21.79 V	Voltage:	21.24 V	Voltage:	21.81 V
Revolutions*:	6891 rpm	Revolutions*:	6030 rpm	Revolutions*:	4022 rpm
electric Power:	304.7 W	electric Power:	686.4 W	Throttle (log):	53 %
mech. Power:	263.6 W	mech. Power:	556.6 W	Throttle (linear):	64 %
Efficiency:	86.5 %	Power-Weight:	415.1 W/kg	electric Power:	289.1 W
			188.3 W/lb	mech. Power:	233.2 W
		Efficiency:	81.1 %	Power-Weight:	178.0 W/kg
		est. Temperature:	73 °C		80.7 W/lb
			163 °F	Efficiency:	80.7 %
				est. Temperature:	46 °C
		Wattmeter readings			115 °F
		Current:	258.48 A	specific Thrust:	5.72 g/W
		Voltage:	21.39 V		0.2 oz/W
		Power:	5528.9 W		

Even though the maximum power draw exceeds the 680 W manufacturer rating, it falls within the +/- 15% allowable variance and was deemed suitable for our application, particularly since the mission will rarely, if ever, require motors to operate at maximum thrust.

Accounting for a 6.6 lb payload, the vehicle has an all-up weight of 29 lbs with an additional payload capacity of 15 lbs. In this configuration it has a thrust/weight ratio of 1.8, a top speed of 19 mph and a range of 3 miles. It is capable of hovering for 20 minutes which is substantially shorter than the time required to complete the mission. In order to accommodate for this time constraint, custom battery mounts were manufactured in tandem with a wiring harness to allow for quick battery swaps without losing power to the flight controller. This design allows the vehicle to operate in predetermined sections of the search area with a return-to-launch function programmed to switch on at critical battery levels.

Autopilot

Flight Controller

An onboard autopilot flight controller is required to achieve autonomous flight. A major consideration for the choice in a flight controller was prior testing and reliability. The Pixhawk was chosen to be the flight controller because of its

(1) low cost
(2) adequate processing power
(3) use in last year's project and familiarity to the team
(4) compatibility with 8-rotor aircraft
(5) compatibility with the Ardupilot software suite.

ArduPilot was selected to be the autopilot platform because it is free, its open-source nature allows expansion and modification to meet competition requirements, and its popularity provides abundant online resources for expansion and troubleshooting. Notably, ArduPilot includes the X-configuration ArduCopter firmware, which is necessary for the vehicle design chosen for the competition.

The Pixhawk is integrated with a sensor suite that collects information vital to the stable flight of the aircraft and completion of the requirements of the competition (Figure 2). Housed within the flight controller unit is the inertial measurement unit (IMU), which detects and reports the orientation, acceleration, and heading of the aircraft. ArduCopter uses extended Kalman filtering to improve accuracy of the location and heading by removing sensor data estimated to be artifactual. Also contained within the unit is a barometer for measuring altitude. An external GPS sensor with compass records location, redundant heading information, and speed. A rangefinder is mounted externally, oriented downward to guide the aircraft in the final stage of landing by measuring and reporting distances from ground under 23 feet via sonar. A power module sensor is connected to the main power supply and monitors battery voltage and current.

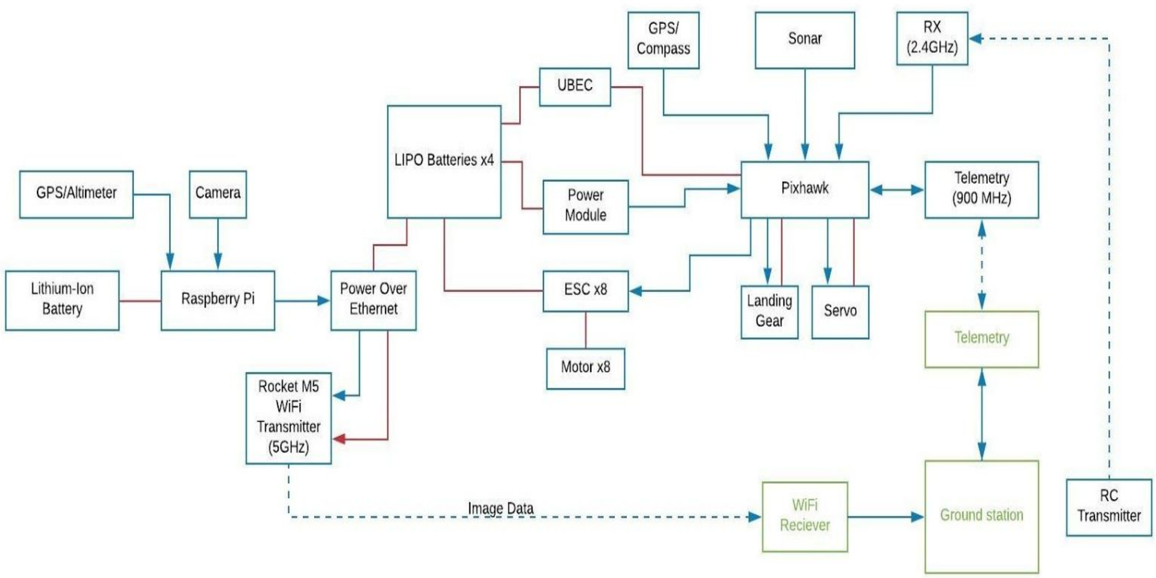

Figure 2: System Configuration

Mission Planner

Data acquired from and reported by the sensor suite is processed onboard to maintain stable flight, navigate to waypoints, and land in a controlled manner. Data from the sensor suite is also processed by the ground control station in-flight to update the flight path for obstacle avoidance, to report required data to the interoperability system, and to report vehicle status. The ground control station (GCS) software application chosen for the project is Mission Planner. Mission planner is an ideal choice for the project because it was designed for ArduPilot and is thus highly compatible with our system. Mission planner also fulfills the requirement that the UAS speed, position, altitude, and mission boundaries are displayed and continually updated (Figure 3).

Figure 3: Mission Planner

Sensor data is relayed at 915 MHz to the ground station via a telemetry radio mounted on the vehicle and on the GCS computer. The communication protocol of Pixhawk is MAVLink which consists of a library of commands. Pixhawk broadcasts these commands through a serial connection to the telemetry radio which is received by the telemetry radio on the ground. This radio is connected via USB to a laptop computer configured as a virtual comm port. Mission Planner reads the MAVLink telemetry data and uses the information to update the navigation map. Autopilot continually updates Mission Planner with all the data points it can perceive such as power use, voltage, current draw, airspeed, altitude, ground speed, and GPS health.

Obstacle Avoidance

The obstacle avoidance system operates on the ground control station by communicating with the Interoperability System and updating Mission Planner. It works along the same principle as modern robotic algorithms. By iteratively updating the waypoints through applying Rapidly-Exploring Random Trees (RRT), it can quickly adapt to static and dynamic obstacles. It also means that the same system can be applied to moving and stationary obstacles. Receiving information from the Python Interoperability library provided, it calculates the new waypoints to avoid the object but remain within its course, and then through a UDP socket, sends it over to the laptop running Mission Planner. This added communication step was required since Mission Planner is on windows while the payload system runs on Ubuntu Linux. An Iron Python script is loaded into Mission Planner that can receive packets from the TCP socket client and updates the mission via Mavlink.

Imaging System

The camera selected for the payload system was a GoPro Hero as it provides affordable high-resolution images that are compatible with Raspberry Pi drivers through using OpenCV 3. The Raspberry Pi Model B was selected not only for its flexibility and low cost, but also for its large amount of documentation, customization ability, and compatibility with secondary hardware.

Object Detection, Classification, Localization

To autonomously locate and classify the emergent and standard targets, we have improved commonly used computer vision methods. Locating regions of interest through selective search, a method often used in regional convolutional neural networks, and blob detection, it is possible to locate targets at a close range as well as far away (Figure 4). The benefit of selective search is that it can easily locate object regions but can have an issue with smaller targets unless its parameters are tuned to pick up small targets but would then create too many ROIs. Blob detection is a frequently used method for small ROIs, often used for aerial imagery analysis. But as our vehicle is designed to high and low flights, its vision system must accommodate both. Therefore, the combination of both methods may increase latency, it increases the accuracy of the system. Future versions of the system will incorporate altitude to adjust the hyperparameters of both methods to further increase the performance with targeting static and known targets.

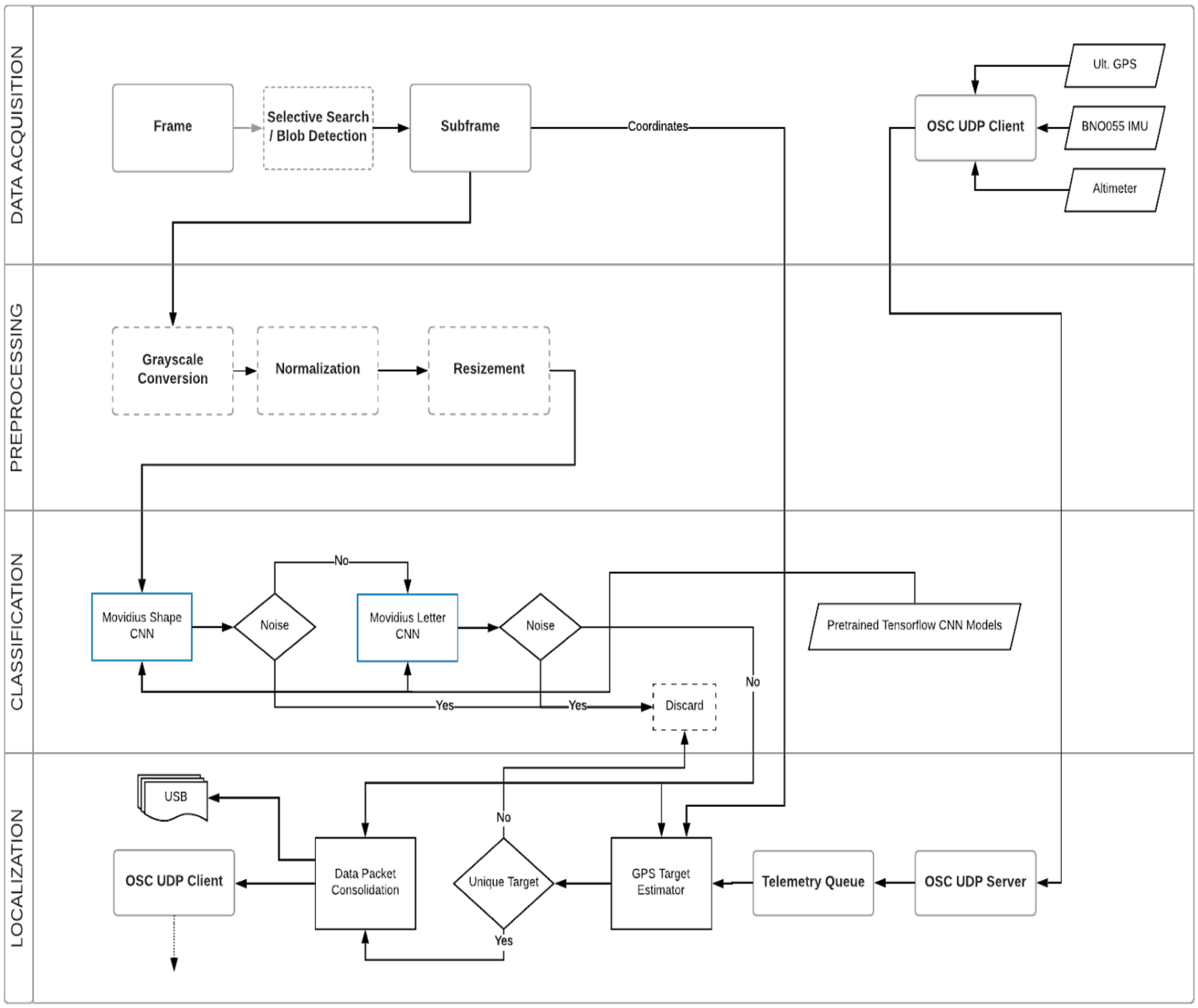

Figure 4: Algorithm Diagram

To optimize the performance of classification, there are several methods used. To make sure that the lighting of the field does not create misclassification from differences between the generated training and testing data, the frame is converted to grayscale and then normalized. When creating the testing data, Gaussian noise was applied to compensate for uniform generated colors. The placements of all shapes and letters were randomized in placement and rotation to avoid overfitting to ideal targets. Using OpenCV3, over a million training examples were generated (Figure 5).

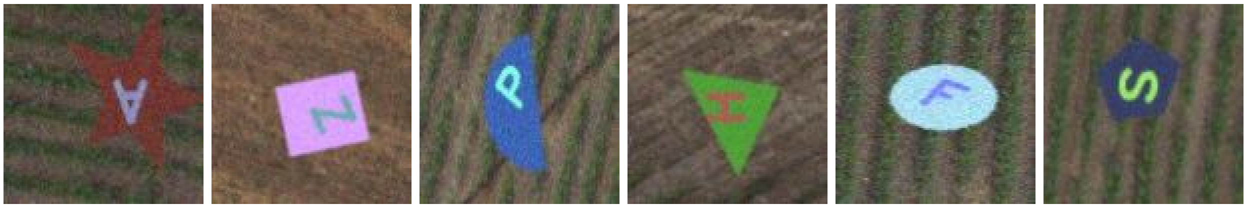

Figure 5: Sample of Generated Training Data

While K-Means clustering and Support Vector Machines have proven successful in binarized subframes, Deep Convolutional Neural Networks were selected for classification due to its versatility to be further applied to future targeting systems. Of the frameworks available such as Caffe, Keras, or Torch, Tensorflow was selected for its established usability, increasing popularity, documentation, and support. Shallow Feed forward networks have been shown to behave fast in similar cases such as with MNIST but do not involve convolutional layers so the image would have to undergo an initial resizement to center the image. And although an implementation of fRCNN of RCNN would implement the selective search already as well as including a SVM, it would have been unnecessary computation to run a convolutional network over the entire HD frame. The framework selected was AlexNet as it could handle harder classification problems later since it was created for the ILSVRC or ImageNet dataset. And while further version may incorporate Inception v4, latency is the greatest issue for deep learning UAS applications and Inception architecture would over-complex for the required task. To reduce overfitting, optimizations such as dropout and weight regularization were employed. Utilizing Intel's AI DevCloud, the network was built with Google Tensorflow, visualized with Tensorboard, trained on Xeon Scalable Processors and deployed in real-time using the Movidius Neural Compute Stick (Figures 6 and 7).

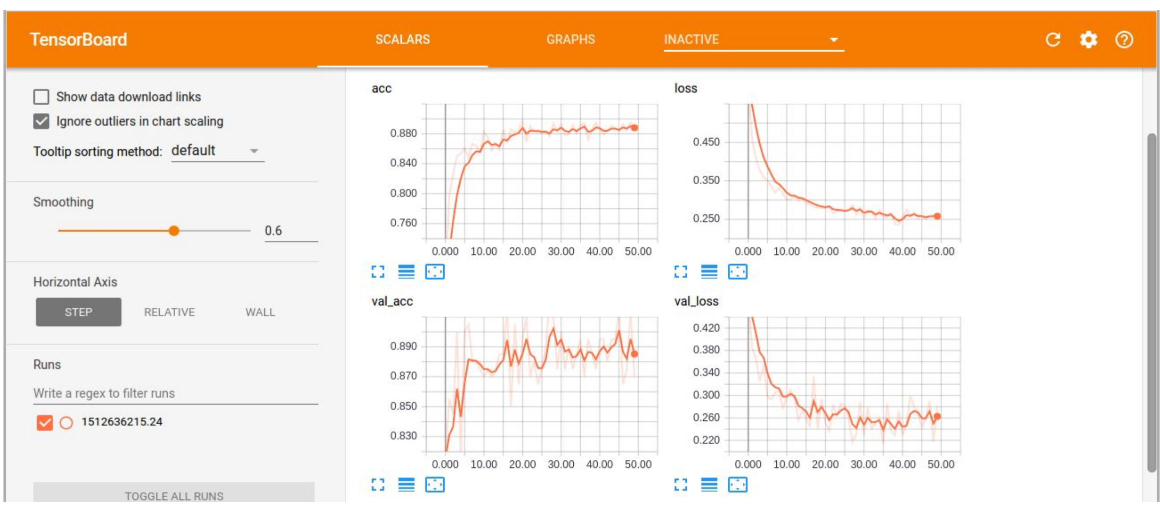

Figure 6: Tensorboard Neural Network Architecture Visualization

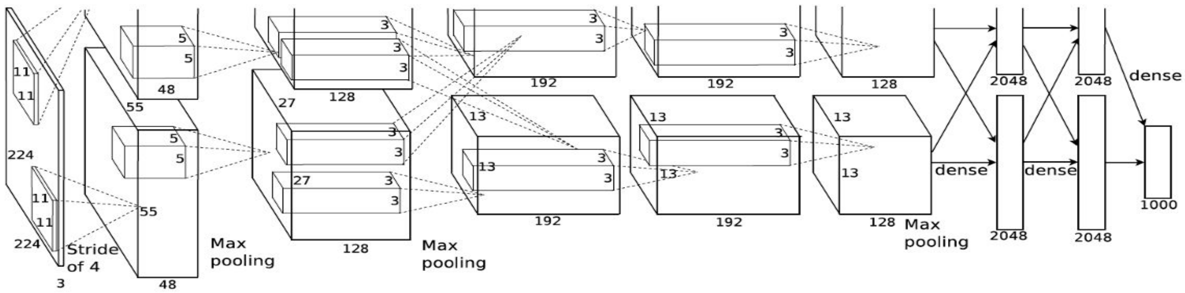

Figure 7: AlexNet Architecture Diagram

Using the secondary sensors onboard the payload bay and coordinates of the confirmed target on the ground, the localization system can estimate the GPS coordinates of the target. Compensating for the rotating reference frame through a matrix correction function and the absolute orientation given by the BNO055 IMU, the polar vector from the vehicle to the target is calculated using the ratio of the coordinates and total pixel length and height and the angle ranges of the camera. Using an altimeter and onboard GPS, our system calculates the distance and direction of the target and extracts its location. To solve the issue of repeating targets as we cannot just assume that there will be no repeating combinations of shapes and letters, the estimated location is compared with the estimated locations of any of the same given combination. It is eliminated from the final output if it does not pass a L2 distance threshold.

In order to extract the color of the shapes and letters, the average outer frame pixel color is taken for background and a kNN clustering function separates out the colors. These main colors are then compared against a dictionary of colors and the closest L2 distance is selected. To optimize this process, the frame is initially blurred using a Gaussian blurring function and decreased in size to increase speed.

To switch to manual image analysis mode, it can be set to different modes based upon the required mission and tested latency. To transmit images, it sends out the pre-processed subframe over a UDP Multithreading OSC client until it receives a confirmation packet from a secondary TCP server. This method ensures that the packet can be sent securely without fear of system failure from a broken socket connection. The second automatic mode is to have the data stored on a USB with the required telemetry included for review after landing.

The main emphasis of the system design of the computer vision system is to reduce the total number of computations required for any given frame. The GPS estimation function runs only if the two convolutional neural networks both classify the frame as non-noise (not background). Instead of sending an HD frame to the ground station for processing, all processing is done onboard. The Movidius Neural Compute Stick runs the most computationally expensive operations of the process at high speed and low power while the Raspberry Pi Model B itself was optimized to no longer run as a miniature PC but had its graphical functions disabled and situated with openings such that the movement of the vehicle would cool down the electronics. Through these methods, we have obtained a powerfully flexible platform for running computer vision experiments while using a fraction of the usual cost associated with similar performance.

Communications

Communications fall into three categories: radio control link, autopilot telemetry, and image capture (Figure 8). Our radio control link operates on the 2.4 GHz frequency spectrum using a FrSky receiver/transmitter combo. Frequency hopping is employed to reduce interference in the 2.4 GHz range and all communication over this frequency falls under part 15 FCC compliance. This is a standard hobby radio controlled communication system that is employed by most out of the box R/C radio systems.

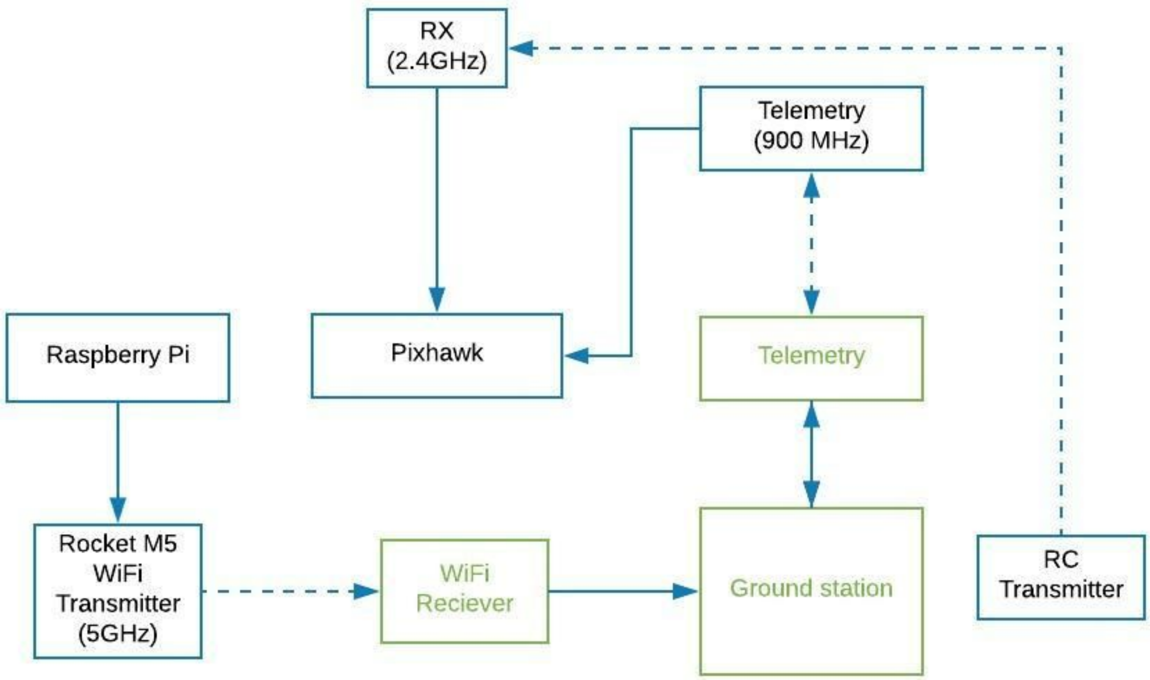

Figure 8: Communication Network

Autopilot telemetry will be transmitted via MAVLink protocol using 900 MHz 3DR radio telemetry system. The radio relays the MAVLink protocol telemetry data between the Pixhawk and the ground control station computer which runs the Mission Planner Software. In addition to receiving telemetry, this link allows for commands to be transmitted to the aircraft during flight.

The image data telemetry system operates using a long-range directional wifi network. By using a Ubiquiti NanoStation on the ground station and a Ubiquiti Rocket M5 onboard, it achieves a range of over 9 miles. Of the two main options available for Python socket transmissions, UDP is superior to TCP as it provides reliability over long distance although it does not ensure transmission of packages. But as the main mission commands can be sent reliably through a TCP socket, the actual data transmission can be broadcast over UDP. To prepackage the data, the easiest and proven method is Open Sound Control or OSC, a protocol often used for transmission of moderately large data packets such as video frames.

In order to maintain connection, a student-designed two-axis gimbal tracks the UAS and points the Ubiquiti Nanostation towards the Ubiquiti Rocket M5. The gimbal mechanism shown in (Figure 9) is entirely student designed and 3D printed. It consists of a platform attached to a camera tripod which rotates using a NEMA 17 stepper motor and a servo which tilts the antenna to the appropriate angle as directed by the onboard Raspberry Pi 3 computer. Using a Raspberry Pi to communicate with Mission Planner to obtain vehicular telemetry, the position of the UAS relative to the ground station can be triangulated and the gimbal correctly positioned. The data received from the Nanostation is sent from the Raspberry Pi to the Linux laptop to be processed and sent to the Interoperability System.

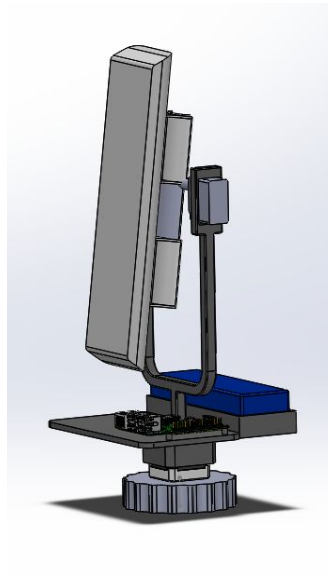

Figure 9: Ground Station Gimbal

Air Delivery

The 2018 SUAS competition requires the UAS to drop an 8 oz Nestle water bottle onto a location defined by GPS coordinates. The water bottle is to disperse the water contained within upon reaching the target. Success of the air delivery objective was determined to hinge upon several important components:

(1) A mechanism capable of releasing the water bottle from the vehicle
(2) Activation of the release mechanism at the correct time and location
(3) A process by which water is reliably released from the water bottle at the target zone, which may range from hard concrete to soft grassy field
(4) A strategy to secure the water bottle to the UAS in a manner conducive to stable flight

A servo-driven gripper was chosen to be the air delivery mechanism. The servo will be actuated by the Pixhawk flight controller to which it is directly wired. The arms of a basic, commercially-available gripper unit were replaced with 3D-printed arms that conform to the diameter of the water bottle. To increase friction between the water bottle and gripper arms, rubber bands will be wrapped around the water bottle where the surfaces meet (Figure 10).

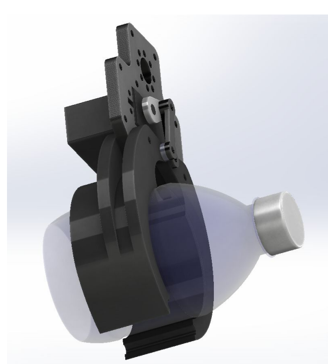

Figure 10: Air Delivery Mechanism

A GPS unit connected to the Pixhawk provides the necessary location data for the drop zone. Releasing the water bottle at the correct time is greatly simplified by the octocopter design chosen for the competition compared to fixed-wing aircraft, for which correct timing of the release must be calculated based on speed, heading, and altitude. The octocopter will simply hover over the drop zone at a predetermined altitude and release the water bottle directly above the target. Adjustments to compensate for drift due to wind currents could be made, but they will be considered to have negligible impact in part because no attachments that could significantly increase drag will be added to the water bottle.

Successful dispersal of the contents of the water bottle is complicated by the variability of the possible surfaces that the target zone may be comprised of (i.e. concrete, soil, asphalt, grassy field). The forces from impact of the water onto each of these surfaces may be sufficient to rupture the water bottle, fulfilling the dispersal portion of the mission objective. Repetitive drops from different heights onto the softest of the possible surfaces is necessary to test reliable breakage of the 8 oz Nestle water bottle. At the time of writing this document, this testing has not yet been completed. In the event that the water bottle does not reliably rupture even at the maximum allowed height (750 feet - ground elevation at the base), an internally-bladed housing that insures dispersal of the water upon impact with the ground will be affixed to the water bottle.

The water bottle must be secured to the UAS in a secure manner. Cargo that is free to move around during flight may damage components on the aircraft, shake parts loose, produce wasteful strain on the motors of the octocopter that must continually stabilize itself, etc. When closed, the gripper adequately restricts movement of the water bottle.

Cyber Security

To protect the network security between the ground station and vehicle, the wireless connection has a WPA2 AES protocol built in. This method is the highest rated industry level wifi security protocol, the same method used to protect government communications. Another level of AES security is placed on data coming in from the TCP socket which is verified with a shared key file between the ground station and vehicle for the change of mission signals. Additional security is placed on the ground station itself, not only is the user interface password protected, but with every login, it records an image profile of the user which is automatically sent to the cloud when connected to internet and read-only locally. Future versions will incorporate Dlib Facial Landmark Detection, Haar Cascades, and Support Vector Machines to identify users (Figure 11).

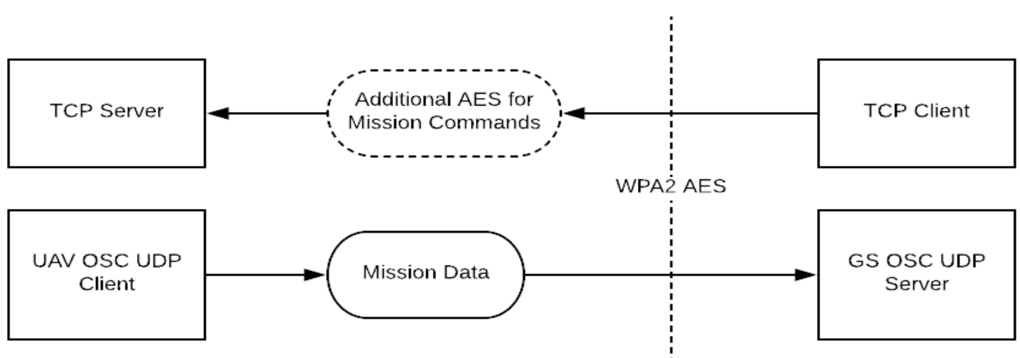

Figure 11: Security Diagram

Safety, Risks and Mitigations

Developmental Risks and Mitigations

In order to minimize the possibility of human error, multiple checklists were employed for both the aerial system and ground control station. Detailed inspections are carried out before and after each flight and any issues present have to be cleared by the team lead and safety officer before operations can continue. Any issues with the airframe, autopilot and any related systems are fixed before next flight. Arming of the UAS is done in stages specifically designed to ensure that no one is near the propellers when the motors are fully armed.

Since mishandled LiPo batteries present a significant risk of injury or property damage, measures were taken to mitigate this risk. Batteries are charged using a ISDT SC-620 charger with a parallel charging board which allows four batteries to be safely charged at once, and power is supplied by a Chargery S600 power supply. Batteries are checked prior to each flight for correct and even charge across all cells and charge is actively monitored during flight. Special LiPo bags are used for transportation and batteries are only shipped via ground transport.

At least one fire extinguisher and a first aid kit is required to be present at all times. Due to the risk of electrical shock and arcing, the design employed arc-free AS150 connectors.

Mission Risks and Mitigations

Redundancy is built into many of the systems on the UAS. Barometric and compass data are measured in both the 3DR GPS unit and Pixhawk flight controller in order to avoid autopilot errors during autonomous flight. GPS data is gathered by the 3DR GPS module attached to the Pixhawk as well as a tandem GPS unit located in the payload bay which is attached to the onboard Raspberry Pi 3B computer. Latitude and longitude data is transmitted to the ground station via wireless data link and allows the mission controllers to verify that the systems are in agreement throughout the extent of the mission.

In order to minimize the risk of any damage to property, all flight testing was performed away from private residences and areas with high vehicular traffic. Flight plans are designed to keep the UAS away from hazards both to itself and others in order to avoid destruction of the vehicle as well as potential injury to bystanders. Testing is conducted away from controlled airspace as per FAA regulations.

Autopilot malfunction presents a large risk since an out-of-control UAS can be quite hazardous to both people and property. In the event that the autopilot does not perform as intended, a manual override is installed and a safety pilot is always on hand and ready to take over the flight controls.