DESIGN AND BUILD AN EVTOL DRONE

By

Zhengpu Ye

Chenyang Huang

Hongfan Liu

Xuan Chen

Final Report for ECE 445, Senior Design, Spring 2024

Supervisor: Jiahuan Cui

TA: Yanzhao Gong

3 June 2024

Project No. 42

Abstract

We have designed a five-axis vertical takeoff and landing drone, optimizing its mechanical structure and circuit control. This innovative design enhances payload capacity and extends flight distance compared to conventional drones. Additionally, we have integrated visual capabilities, enabling real-time image transmission via online live streaming. We can watch the images transmitted from the drone's perspective and perform car recognition on the images. This feature offers advanced monitoring and analysis functionalities.

Key words: eVTOL, Mechanical structure optimization, Circuit control, Visual capabilities, Car recognition.

Contents

1. Introduction

In recent years, electric vertical take-off and landing aircraft technology has rapidly developed, finding extensive applications in agriculture, medical rescue, logistics, and environmental monitoring. These drones have become a significant force in promoting social progress and technological innovation. For instance, drones are being used for precision agriculture to monitor crops and apply pesticides more efficiently, significantly enhancing agricultural productivity (Zhang & Kovacs, 2012).

Additionally, transport drones hold potential in commercial photography, geological exploration, and rapid building construction. They are being utilized for aerial photography and filming, providing unique perspectives and cost-effective solutions for the media and entertainment industry (Colomina & Molina, 2014). In geological exploration, drones help in mapping and surveying difficult terrains, making the process safer and more efficient (Nex & Remondino, 2014).

As technology advances, the endurance, reliability, and intelligence of drones are improving, enabling them to undertake more complex missions across various fields. Innovations in battery technology and AI-driven navigation systems are enhancing drone performance, allowing them to operate longer and more autonomously (Floreano & Wood, 2015). Heavy-carrying transport drones are becoming the "air workers" of modern society, enhancing work efficiency, responding to public health events, ensuring supply chain stability, and promoting economic development (Garrett & Anderson, 2018).

For instance, during the COVID-19 pandemic, drones were used to deliver medical supplies and vaccines to remote areas, showcasing their potential in addressing public health challenges. With improved regulations, systems, and public acceptance, the prospects for transport drones are expanding, making them an integral part of future logistics and emergency response systems (Clarke, 2014).

2 Design

To better coordinate our work and achieve interdisciplinary collaboration, we divided the work content from hardware to software into three parts: structural design, electronic control, and visual design, and divided their functions into different architectures.

The structural design part focused on the mechanical aspects of the drone, including the airframe, propulsion system, and overall aerodynamics. This ensures the drone's stability, durability, and performance under various environmental conditions.

The electronic control part was responsible for developing and integrating the onboard electronic systems, including flight controllers, sensors, and power modules. I implemented advanced algorithms for autonomous navigation, real-time data processing, and efficient energy management to enhance the drone's operational capabilities.

Lastly, the visual design team concentrated on the integration of camera systems and image processing software. They developed methods for real-time video streaming and object recognition, enabling the drone to perform tasks with high precision.

This interdisciplinary approach allowed us to leverage the expertise of each team, resulting in a robust and versatile drone capable of performing a wide range of applications. Effective collaboration and communication among these teams were essential to integrate their respective contributions seamlessly, ensuring the drone's overall functionality and reliability (Bhardwaj et al., 2016).

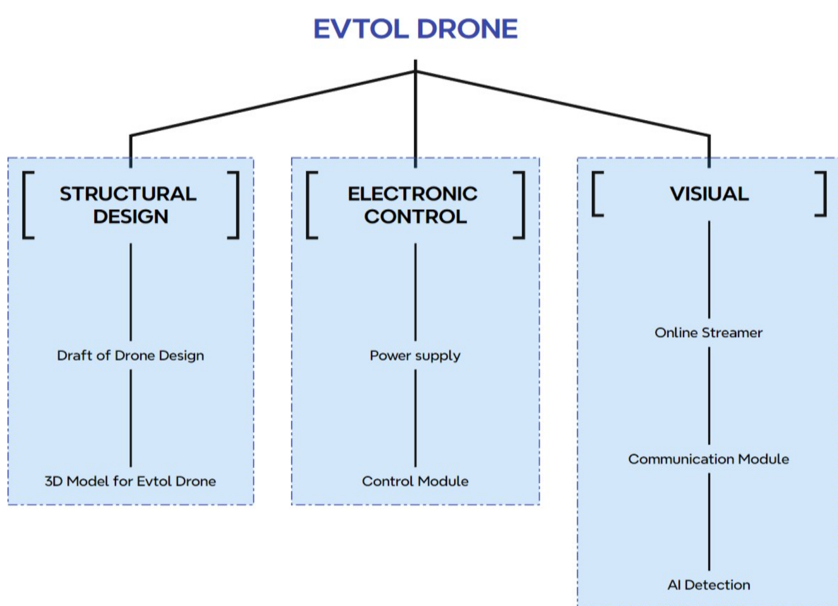

Figure 1: Functions of the Drone

2.1 eVTOL System Architecture and Operation Flow

The following block diagram illustrates the comprehensive system architecture of an electric Vertical Take-Off and Landing (eVTOL) UAV, highlighting the interaction between various hardware and software components.

2.1.1 Sensor Integration

The system begins with a set of sensors, including GPS, IMU (Inertial Measurement Unit), and Barometer/Magnetometer, which provide essential data for navigation and stability.

GPS Data: Provides real-time positioning information.

IMU Data: Offers acceleration and angular velocity data, crucial for orientation and balance.

Barometer/Magnetometer: Supplies altitude and directional information.

2.1.2 Flight Controller

The sensor data is processed through a Pixhawk/PX4 flight controller. This component performs sensor fusion and state estimation, integrating the data to determine the UAV's current state.

Flight Algorithm: The flight controller executes predefined algorithms to maintain stability and control during flight.

2.1.3 Communication and Processing

The Raspberry Pi serves as an intermediary, receiving flight algorithms from the flight controller and converting them into real commands for the UAV.

A SIM card enables data transmission to the cloud for video uploading and remote monitoring.

2.1.4 Cloud Integration

Real-time video data is uploaded to the cloud, enabling remote monitoring and control. Item recognition system is also combined, which enables the drone to recognize roadblocks such as cars and street lamps.

2.1.5 Controller Hardware

Commands from the Raspberry Pi are sent to the controller hardware, which manages the actuators.

2.1.6 Actuators

The actuators execute the commands, adjusting the UAV's orientation and propulsion.

2.1.7 Subsystems

There are two subsystems contained in our eVTOL:

1. Flight Control System: Comprises direction control and power control, ensuring precise maneuverability.
2. Power System: Includes motors (the drone has five motors in reality, the figure only shows Motor1 and Motor2) and propellers (the drone also has five motors in reality, the figure only shows Prop1 and Prop2), providing the necessary thrust for take-off, flight, and landing.

2.1.8 Conclusion

The drone system depicted in the diagram is designed for seamless integration of its various components, enabling efficient and reliable flight operations. The use of advanced sensors and robust control algorithms ensures high precision and stability, making the eVTOL UAV suitable for a wide range of applications.

2.2 Structural Design

The structural design encompasses several key elements, including the airplane's shape, motor positioning, overall stability, and assembly processes.

Our design phase is segmented into three primary stages: initial concept, detailed design, and subsequent adjustments and improvements. Each of these stages will be elaborated upon in the following sections.

2.2.1 initial concept

Initially, considering the vertical take-off and landing capability, it was essential to incorporate devices that provide direct upward thrust. Simultaneously, to achieve prolonged horizontal flight, fixed wings and forward propulsion were necessary. Drawing inspiration from existing UAV structures, Figure 2 illustrates our initial design.

The fuselage is constructed from lightweight EPO (expanded polyolefin) material. Four rotors are utilized to generate vertical lift, while a propulsion motor and fixed wing ensure efficient horizontal flight. This configuration represents our initial conceptualization.

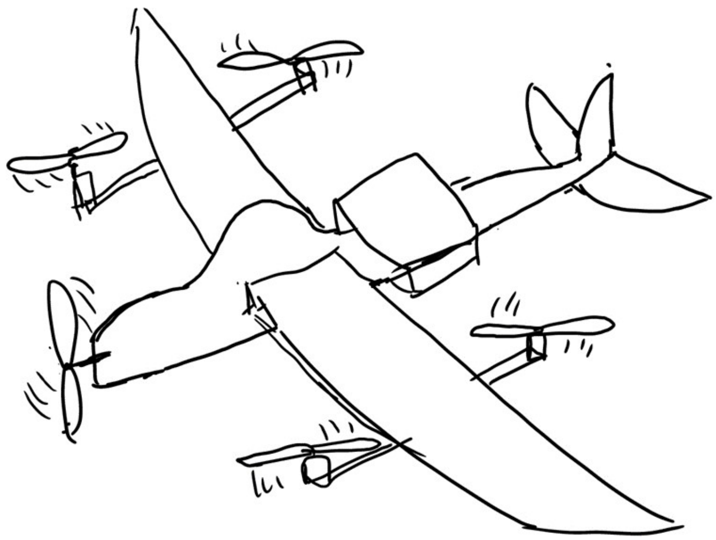

Figure 2: Draft of Drone Design

2.2.2 Detailed Design

After conceptualizing the initial idea, the specific structure was designed using 3D modeling software. At this stage, the connections between the four-axis motors and the carbon tubes, as well as between the carbon tubes and the fixed wing, were addressed. Bolts and nuts were primarily used for fastening, complemented by hexagonal aluminum posts and some custom-printed parts.

During this period, the appropriate motors and batteries were selected based on the aircraft's weight assessment. The carbon tubes were custom-fabricated, and the necessary fixings were produced using the school's 3D printer.

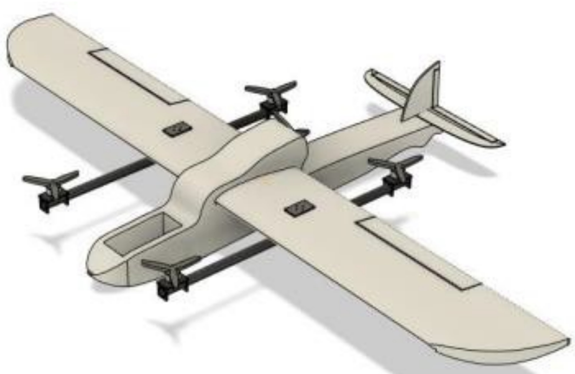

Figure 3: 3D Model of the eVTOL Drone

2.2.3 Assembly, adjustment and refinement

After constructing and initially completing the airplane, several issues were identified.

The motors were not securely fastened, resulting in instability and wobbling. This was attributed to the softer wings. To address this, we added two additional carbon tubes to enhance the structural stability, as shown in Figure 4.

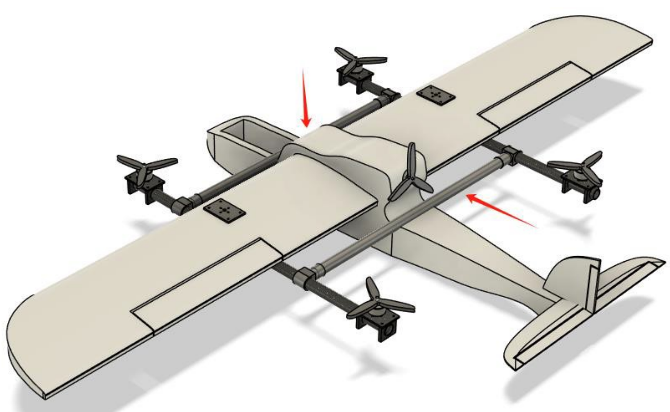

Figure 4: New Carbon Tubes

We replaced the original motors with more powerful models after determining that the initial motors were underpowered. The new motors feature XT60 connectors, which mitigates issues related to soldering.

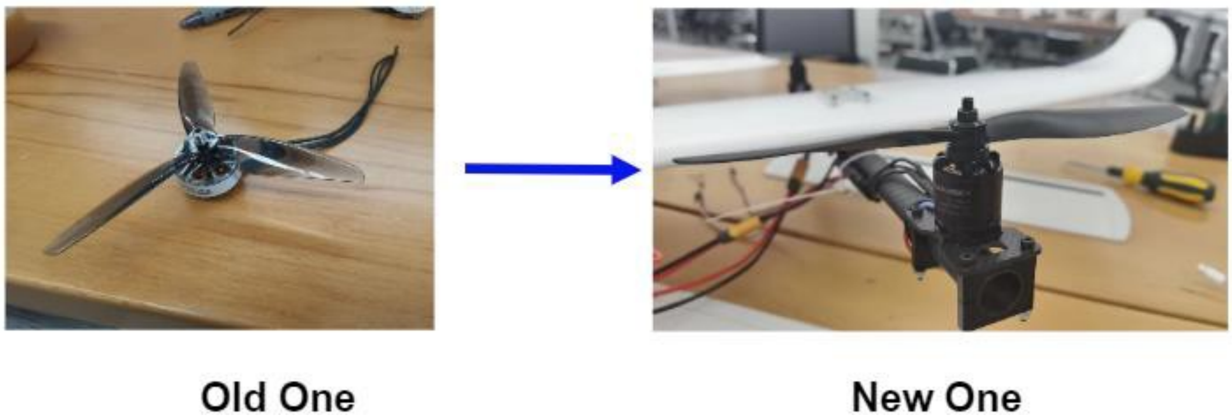

Figure 5: Change of Motors

Additionally, we observed that the plane's center of gravity was too far forward. Consequently, we repositioned the battery from the nose to the center of the fuselage. Another challenge was the placement of the circuit components. Initially, we intended to house all components in the nose cavity, but this resulted in an unstable center of gravity, limited space, and inconvenient debugging. To resolve this, we mounted the ESCs (Electronic Speed Controllers) on the new horizontal carbon tubes and placed the splitter plate on the fuselage. The Pixhawk flight controller was affixed to the outside of the headstock, facilitating easier adjustments.

After these optimizations, the final version of our design is illustrated in Figure 6 below.

Figure 6: Final Version of the Drone

2.3 Power Supply

The circuit is powered by a BOSLI-PO battery with a capacity of 1300 mAh. Our power budget is 195 A (maximum) at 22.2 V, primarily consumed by five propeller motors. Each motor draws 36.6 A of current. This battery was selected for its affordability and ability to provide sufficient power for our needs.

Figure 7: BOSLI-PO Battery

TABLE 1: Requirement and Verification of Power supply

Requirement	Verification
Outputs larger than 183 A (36.6 A*5=183 A) at 22.2 V	A. Measure the open-circuit voltage with a voltmeter, ensuring that it is at 22.2 V B. Ensure that the currents through the motors are about 36.6 A using an ammeter in series

2.4 Control Module

The control module receives signals from the drone controller and manages the flight operations. It comprises a Pixhawk flight controller, a remote-control receiver, a remote control, and five electronic speed controllers.

2.4.1 Assembly, adjustment and refinement

The Pixhawk 2.4.8 Drone Flight Controller functions as the brain of our drone. It interprets user commands and transmits electrical signals to the electronic speed controllers. We selected this version due to its robust features and affordable price.

Figure 8: Pixhawk Flight Controller

2.4.2 Micro zone MC6C remote control and MC7RB remote-control receiver

The remote control and receiver ensure that the drone can receive our commands accurately. This system performed well in our computer software simulations and flight tests.

Figure 9: Microzone MC6C Remote Control and MC7RB Remote-control Receiver

2.4.3 BLHe li-S electronic speed controllers

The electronic speed controllers (ESCs) regulate the speed of the motors. We selected BLHe li-S ESCs for their compatibility with various communication protocols.

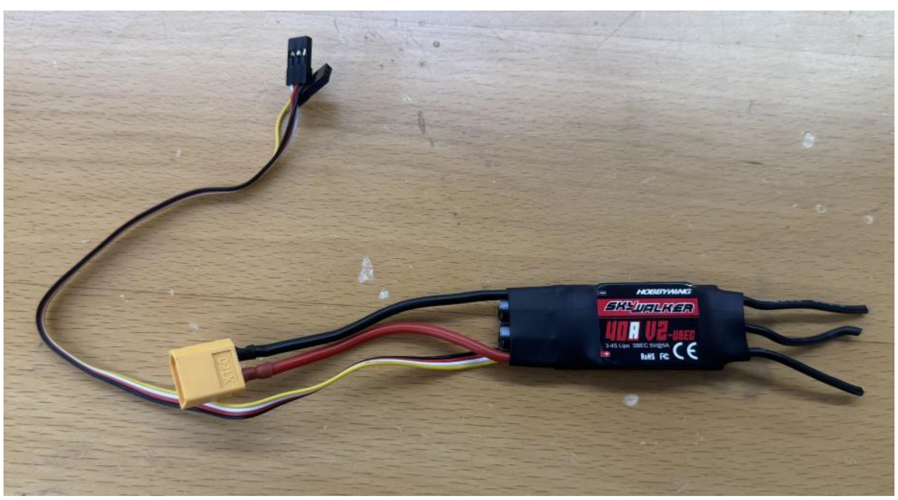

Figure 10: BLHe li-S Electronic Speed Controller

2.4.4 Ground Control Software

We selected Mission Planner VTOL Survey as our ground control station. Mission Planner is a versatile ground control software for planes, copters, and rovers. It can be used both as a configuration utility and as a dynamic control supplement for autonomous vehicles (ArduPilot Documentation, 2024, para. 2).

Using Mission Planner, we can calibrate the drone's accelerometer, GPS, remote controller and its receiver, and electronic speed controllers. Additionally, we can adjust the drone's parameters to enhance flight stability and efficiency.

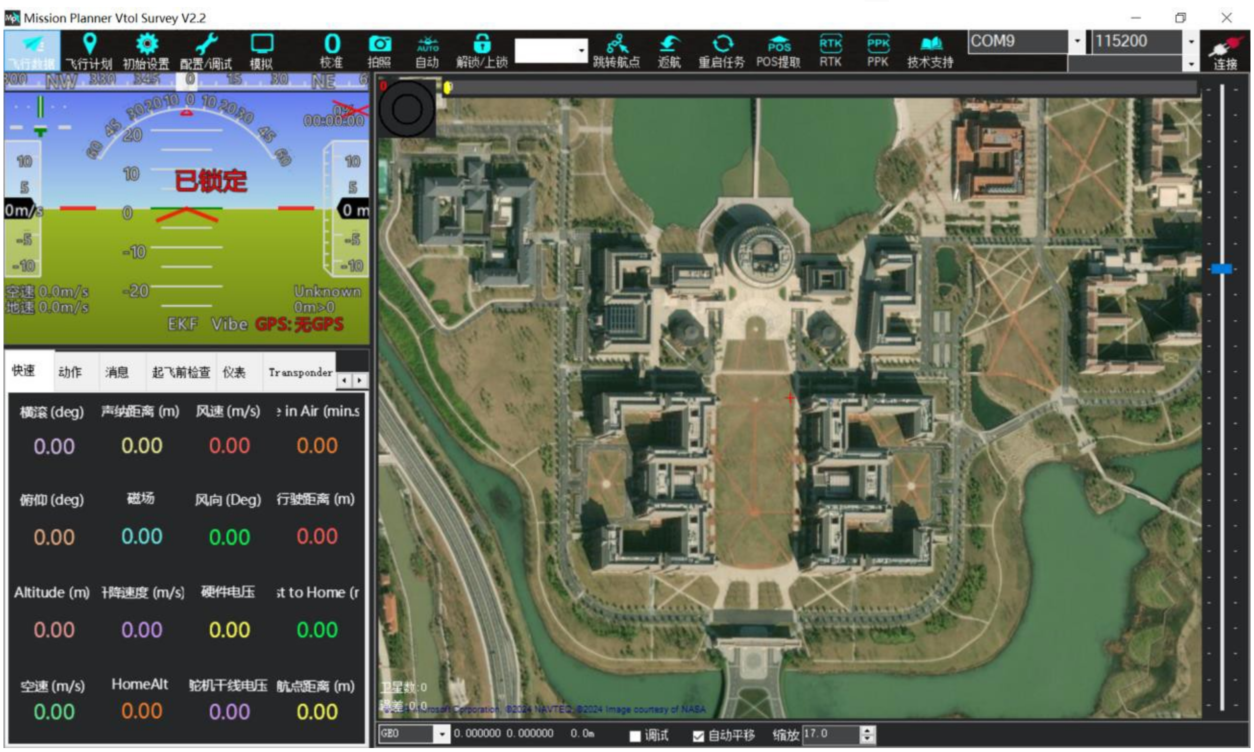

Figure 11: Mission Planner VTOL Survey

(Introduction and figure of Mission Planner VTOL survey: https://ardupilot.org/planner/docs/mission- planner-overview.html)

2.5 visual functions

During the construction of our drone, we also integrated visual functions. This functionality primarily comprises a Raspberry Pi PCB board, a USB camera, and a communication module based on a SIM card. Below, we will elaborate on the following points.

2.5.1 Raspberry Pi(3B) in eVTOL drone

The Raspberry Pi serves several key functions in our drone:

1. **Controller**: Acts as the main controller, processing flight control logic and responding to remote commands, and works with flight control boards like Pixhawk for complex flight modes and navigation.

2. **Real-Time Video Transmission**: Connects to a USB camera to capture and transmit real-time video back to the ground station, providing FPV or video monitoring.

3. **Data Collection and Processing**: Connects to multiple sensors (GPS, barometer, temperature sensor) to collect flight data for logging, environmental monitoring, and GIS data collection.

4. **AI and Image Processing**: Processes images and runs lightweight AI models for real-time target recognition, tracking, and obstacle avoidance, enhancing autonomous flight capabilities.

5. **Communication and Networking**: Supports Wi-Fi and Bluetooth for remote control and data transmission and enables network communication with other drones or ground stations for collaborative tasks and group flights.

2.5.2 USB connected micro camera.

Combining a Raspberry Pi with a USB camera creates a cost-effective and powerful video surveillance system. Using software like Motion, it can perform motion detection, video recording, and real-time streaming. In our project, the USB camera acts as the "eyes" of the drone, aiding in navigation, obstacle avoidance, and object recognition. This is crucial for developing autonomous vehicles and drones. Our drone will primarily use the mjpg-streamer module for video communication and OpenCV Python for deep learning, enabling object recognition.

2.5.3 SIM card communication module

With the hardware in place, our visual system can broadcast videos on the same LAN. However, we need a custom communication module to simulate real-world communication.

We chose to build a SIM card-based communication module for the Raspberry Pi, providing mobile network access. This setup allows for data transmission and remote control without relying on Ethernet or Wi-Fi. The key functions include data transmission, remote monitoring, and management. The Raspberry Pi can send and receive data over mobile networks, enabling remote login, command execution, and configuration updates from anywhere.

Additionally, the SIM card module supports SMS and, in some cases, voice calls, expanding the communication capabilities of the Raspberry Pi.

2.5.4 Implementation of video live streaming

Setting up a camera for live streaming on a Raspberry Pi can be done using mjpg-streamer. Since mjpg-streamer is not available in the standard Raspberry Pi repository, we will compile it from source.

After compiling and installing mjpg-streamer, it can be run to start the video stream. The video stream can then be accessed via the Raspberry Pi's IP address on the local area network.

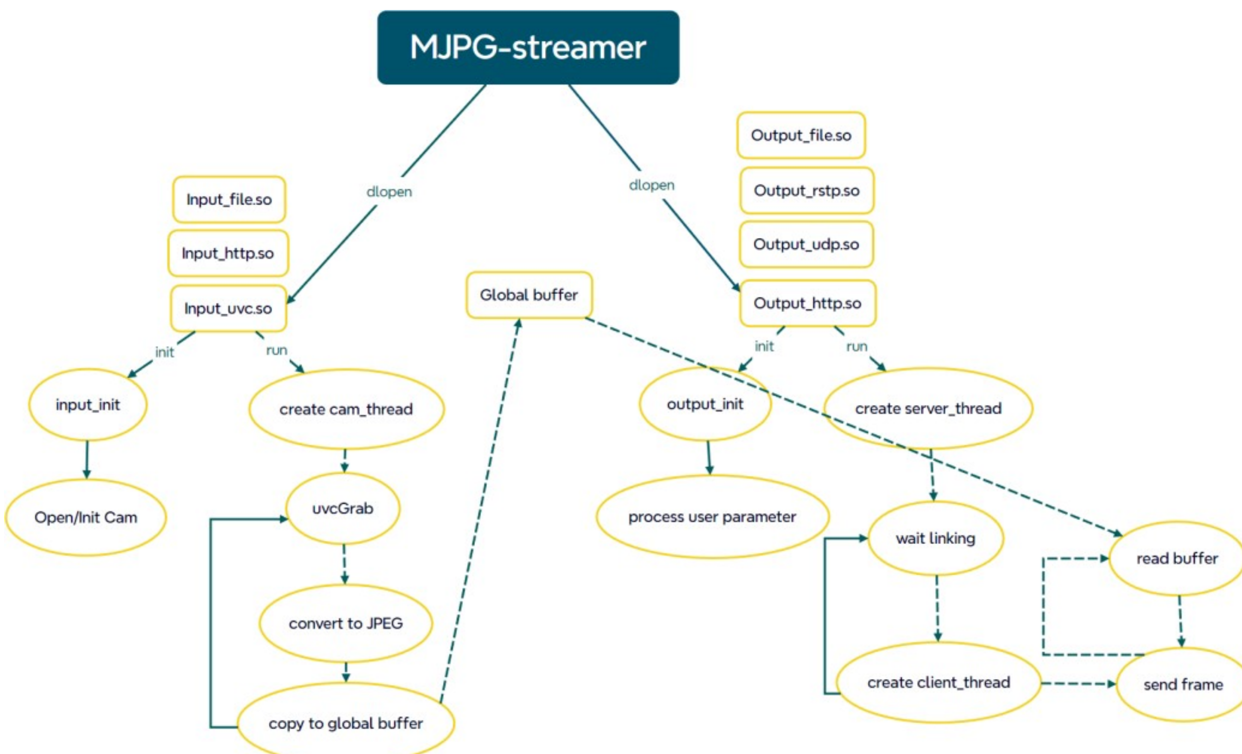

Figure 12: Mjpg-streamer Function

As a lightweight video streaming software, mjpg-streamer offers several advantages:

1. **Low Latency**: Provides lower latency during streaming, crucial for fast-response applications like remote control or monitoring.
2. **Lightweight**: Written in C, it is resource-efficient and suitable for devices with limited resources, such as the Raspberry Pi.
3. **High Compatibility**: Supports various USB cameras and operating systems, making it versatile for different hardware and software configurations.

2.5.5 Rental and video uploading of cloud platforms.

In this step, we will first rent a cloud platform, such as Alibaba Cloud or Huawei Cloud. Next, we will configure video capture and encoding settings using FFmpeg to encode the video stream for Internet transmission. We will set specific protocols (such as RTMP or HLS) and the stream destination URL according to the cloud platform's requirements.

Once configured, we will start the video stream using FFmpeg or similar tools to capture and transfer the video to the cloud platform. The final step involves processing and distribution on the cloud platform, including video analysis, storage, and transcoding. Additionally, we will configure access control and security measures to ensure the secure transmission and access of video streams.

2.5.6 Use Yolov5 to complete image recognition

By integrating the deep learning algorithm based on YOLOv5 (You Only Look Once version 5) and the UA-DETRAC vehicle detection dataset, we achieved real-time monitoring and identification of highway vehicles. The system continuously captures highway video streams during flight and analyzes road conditions to provide critical information for traffic management, safety supervision, and accident response.

YOLOv5 is a leading object detection framework known for its excellent detection speed and accuracy. In our project, the YOLOv5 model has been specially trained to adapt to various types of vehicles in the highway environment. Additionally, the UA-DETRAC public dataset, which contains diverse vehicle images and annotation information, enhances the model's ability to detect vehicles of different types and sizes in complex traffic scenarios.

The trained model is deployed on the drone. The onboard camera transmits images back to the server in real-time, where the vehicle recognition model quickly processes the video data to detect and mark the location and trajectory of the vehicles.

```python
def parse_opt(known=False):
    data = "engine/configs/voc_local.yaml"
    workers = 8
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', type=str, default='engine/pretrained/yolov5s.pt', help='initial weights path')
    parser.add_argument('--cfg', type=str, default="yolov5s.yaml", help='model.yaml path')
    parser.add_argument('--data', type=str, default=data, help='dataset.yaml path')
    parser.add_argument('--hyp', type=str, default='data/hyps/hyp.scratch-v1.yaml', help='hyperparameters path')
    parser.add_argument('--epochs', type=int, default=300)
    parser.add_argument('--batch-size', type=int, default=8, help='total batch size for all GPUs')
    parser.add_argument('--imgsz', '--img', '--img-size', type=int, default=640, help='train, val image size (pixels)')
    parser.add_argument('--rect', action='store_true', help='rectangular training')
    parser.add_argument('--resume', nargs='?', const=True, default=False, help='resume most recent training')
    parser.add_argument('--nosave', action='store_true', help='only save final checkpoint')
    parser.add_argument('--noval', action='store_true', help='only validate final epoch')
    parser.add_argument('--noautoanchor', action='store_true', help='disable autoanchor check')
    parser.add_argument('--evolve', type=int, nargs='?', const=300, help='evolve hyperparameters for x generations')
    parser.add_argument('--bucket', type=str, default='', help='gsutil bucket')
    parser.add_argument('--cache', type=str, nargs='?', const='ram', help='--cache images in "ram" (default) or "disk"')
    parser.add_argument('--image-weights', action='store_true', help='use weighted image selection for training')
    parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--multi-scale', action='store_true', help='vary img-size +/- 50%%')
    parser.add_argument('--single-cls', action='store_true', help='train multi-class data as single-class')
    parser.add_argument('--adam', action='store_true', help='use torch.optim.Adam() optimizer')
    parser.add_argument('--sync-bn', action='store_true', help='use SyncBatchNorm, only available in DDP mode')
    parser.add_argument('--workers', type=int, default=workers, help='maximum number of dataloader workers')
```

Figure 13: Yolo train.py

3. Design Verification

The primary objective of the Design Verification process for our eVTOL drone was to ensure that all design specifications were met. This included verifying lift capacity, stability during flight, energy efficiency, and adherence to safety standards. Ensuring these criteria were met was crucial for guaranteeing functionality and compliance with aviation regulations.

3.1 Structural Function

3.1.1 Ground Control Software

Carbon tubes were inserted into the wings to enhance structural strength, while four additional carbon tubes were used to secure the motors in position, ensuring they remained level. After thorough testing, the overall structure demonstrated stability during flight.

3.1.2 Lift Capacity Test

We measured the maximum lift capacity of the drone to ensure it could ascend while carrying the intended payload. The test was conducted in an outdoor environment, similar to the real working conditions. The lift capacity was within the expected thresholds, validating both the aerodynamic design and motor configuration.

3.2 Flight Function

3.2.1 Flight Stability Test

Flight tests were conducted to evaluate the drone's stability during takeoff, hovering, maneuvering, and landing. Data was collected using onboard sensors. During these tests, the motors maintained a steady power output with minimal deviation and responded well to control inputs during maneuvers. However, oscillations were observed during rapid directional changes, which will require further investigation. These oscillations indicate a need for adjustments in the drone's flight control algorithms to enhance stability.

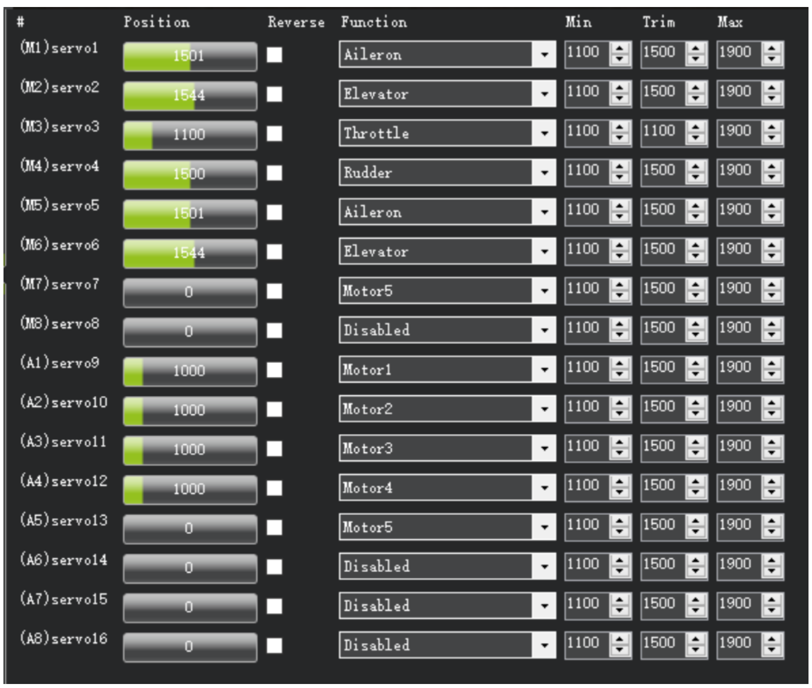

Figure 14: Interface of Data from Sensors

3.2.2 Battery Efficiency Test

We conducted multiple flight cycles to compare the actual energy consumption with the estimated consumption, verifying battery efficiency under various operating conditions. The drone achieved a maximum flight time of over 20 minutes. While the efficiency was within the acceptable range, there is still room for optimization.

3.3 Visual Functions

The ultimate goal of the vision component is to install a vision module based on the Raspberry Pi on the aircraft. To verify the integrity of the entire system, we need to ensure the stability of the Raspberry Pi system, the reliability of the camera's image capture, the consistency of the live stream upload, the stability of the 4G signal, and the accuracy of YOLO recognition.

3.3.1 External camera and mjpg live streaming module

In our design, we used a Raspberry Pi as the core of the visual system. We installed the Raspberry Pi operating system.

Using the "lsusb" command, we can verify the USB connection between the Raspberry Pi and the camera.

For the mjpg-streamer module, as previously mentioned, its basic function is to synchronize and live stream the camera's images to the local area network. First, we check the Raspberry Pi's IP address on the network.

Once connected to the same local area network, we can access the live video by opening the corresponding IP address.

Figure 15: The Ip Address

3.3.2 Rental and video uploading of cloud platform.

We rented the OSS platform for this stage. We received Raspberry Pi streams through local computers and then pushed them through the OSS platform to achieve full-platform video live streaming.

Ultimately, we enabled live streaming accessible across the entire network on any platform, not limited to the local area network. Additionally, we implemented live video recording and cloud storage. Upon ending the live broadcast, the recorded footage is automatically saved and uploaded to the OSS cloud platform, similar to a civil aviation "black box."

3.3.3 4G sim card communication module

We implemented a 4G module using a communication card, allowing us to create a small local area network or directly transmit information to other terminals using 4G signals.

3.3.4 YOLO car recognition part

We adopted YOLOv5 and trained a model autonomously. The training process is illustrated in the following figures.

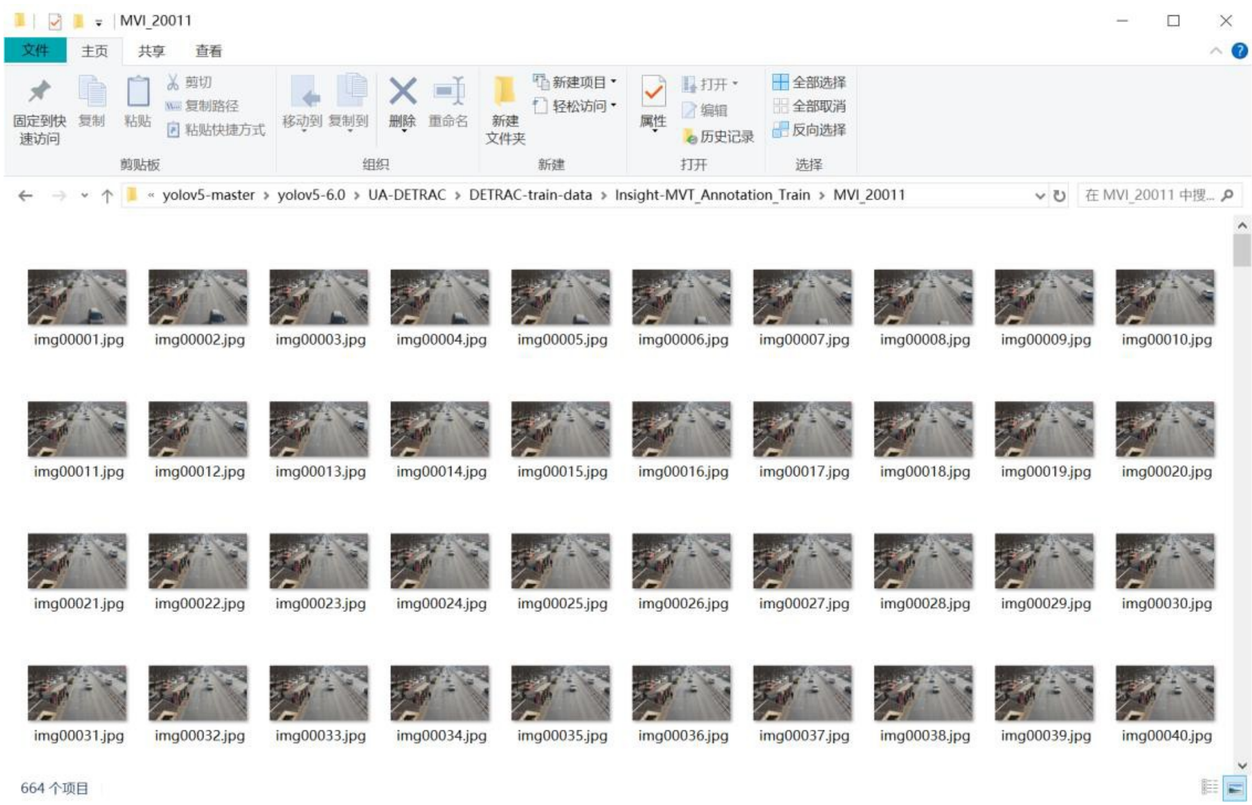

Figure 16: Picture Data

Figure 17: Annotation File

e	train/b	train/o	train/cl	metrics/p	metrics	metrics.	metrics/m	val/bo	val/ob	val/cls	x	x	x/
0	0.087937	0.049659	0.020328	0.85182	0.29447	0.36731	0.13417	0.071884	0.054159	0.01546	0.003332	0.003332	0.070014
1	0.062574	0.039983	0.013392	0.81748	0.60996	0.70846	0.37697	0.059641	0.045212	0.009071	0.006665	0.006665	0.040014
2	0.055072	0.035946	0.009561	0.89193	0.7701	0.87069	0.5641	0.053481	0.040436	0.006636	0.009998	0.009998	0.010013
3	0.050751	0.033327	0.007856	0.93203	0.84739	0.92611	0.6542	0.049007	0.036502	0.005237	0.009998	0.009998	0.009998
4	0.048366	0.031573	0.006952	0.94191	0.87019	0.94136	0.68805	0.04717	0.034599	0.004735	0.009998	0.009998	0.009998
5	0.047063	0.030694	0.00653	0.94863	0.88843	0.9508	0.7076	0.046183	0.033632	0.004473	0.009997	0.009997	0.009997
6	0.0463	0.030024	0.006239	0.95259	0.89614	0.95574	0.71952	0.045556	0.032984	0.004315	0.009995	0.009995	0.009995
7	0.045799	0.029654	0.006066	0.9572	0.89803	0.95805	0.72698	0.045139	0.032612	0.004218	0.009992	0.009992	0.009992
8	0.045342	0.029289	0.005886	0.95617	0.90337	0.95948	0.73111	0.044882	0.032339	0.004154	0.009989	0.009989	0.009989
9	0.045018	0.02905	0.005768	0.95787	0.90551	0.96081	0.73504	0.044697	0.032166	0.004106	0.009986	0.009986	0.009986
10	0.044657	0.028778	0.005663	0.95992	0.90654	0.96187	0.73833	0.044547	0.032015	0.004066	0.009982	0.009982	0.009982
11	0.044425	0.028626	0.005572	0.96019	0.90864	0.96306	0.74133	0.044409	0.031869	0.004034	0.009978	0.009978	0.009978
12	0.044206	0.02848	0.005504	0.96444	0.90751	0.96415	0.74367	0.044306	0.031749	0.004002	0.009974	0.009974	0.009974
13	0.044013	0.028283	0.005441	0.96631	0.90835	0.96518	0.74582	0.044213	0.031644	0.003973	0.009969	0.009969	0.009969
14	0.043903	0.028235	0.005375	0.96602	0.91076	0.96614	0.74897	0.044102	0.031524	0.003942	0.009963	0.009963	0.009963
15	0.043809	0.028179	0.005361	0.96559	0.91312	0.96712	0.75091	0.043985	0.031408	0.003915	0.009957	0.009957	0.009957
16	0.043616	0.027904	0.005319	0.96507	0.91615	0.96807	0.75348	0.043868	0.031281	0.003883	0.009951	0.009951	0.009951

Figure 18: Identification Results

Due to the limitations of the YOLO version and our PC capabilities, real-time image recognition was not feasible, leading to significant transmission and processing delays. Therefore, we chose to perform image recognition on the recorded footage after the live video ends.

Our model, trained with extensive car image annotations, has the basic ability to recognize cars in general environments.

4. Costs

4.1 Parts

Following is a starter table for parts costs.

Table 2: Parts Costs

Part	Manufacturer	Retail Cost (RMB)	Bulk Purchase Cost (RMB)	Actual Cost (RMB)
EPO fuselage and wings	Making the fuselage	50	50	50
Motors and paddles	Driving an airplane	350	350	350
Battery	energy supply	160	160	160
Fixings (carbon tubes, screws, etc.)	stationary part	200	200	200
mc6c remote control	remote control	130	130	130
BLHe li-S ESCs	electronic speed controller	45*5=225	225	225
Pixhawk2.4.8	flight controller	400	400	400
Raspberry 3B	The main part of the entire visual system	350	350	350
4G module	Data transmission and communication	95	95	95
Ali Cloud server	Data storage and parallel computing	180	158.5	317
Website domain name*2	Push flow and pull flow	17	13.5	27
CSI Camera	Vedio capture	65	65	65
Total				

4.2 Labor

Labor cost: The labor cost is an important part for the senior design and the cost are estimated as below. The estimated salary for person is 100 ￥/ h (standard salary for Zhejiang University undergraduates). The normal work time per week is estimated for 40 hours (10 hours per person) according to our estimation for the senior design. We have nine weeks to complete our senior design project.

Thus, the total is torty hours per week multiplied by one hundred dollars per hour multiplied by nine weeks equals thirty-six thousand dollars.

5. Conclusion and Future work

In designing and building our drones, we successfully constructed the basic structure, controlled the circuits, and achieved visual effects. Our wing and fuselage design reduced drag and increased load capacity, while the five-axis circuit control system improved energy efficiency. Additionally, our visual system allows real-time access to drone-captured video via any terminal and network, with accurate vehicle recognition. We aim to transmit recognized results in real-time for path planning in future iterations.

5.1 Accomplishments

One of the standout achievements of this project was the successful integration and testing of an electric propulsion and control system. This system meets our initial performance metrics, providing enhanced thrust-to-weight ratios essential for the eVTOL's vertical takeoff and landing capabilities. The propulsion and control system demonstrated reliability across multiple test flights, establishing a solid foundation for future scalability.

In visual development, we resolved Raspberry Pi startup, implemented a push module and camera for image capture and streaming, and utilized the Ari cloud server for web broadcasting and AI vehicle recognition, as well as large-capacity flight record storage.

Upon startup, the Raspberry Pi connects to the power supply, initiates all programs sequentially via a self-start script, and completes the process of transmitting images from the drone to the server for real-time broadcast and background recognition.

.

5.2 Uncertainties

During visual development, we encountered challenges such as the need for video transcoding to make broadcasts from the Ali Cloud server viewable on live platforms. Additionally, despite the powerful parallel computing capabilities of the server, real-time video recognition was not feasible. Consequently, we had to handle live broadcasting and video recognition separately, storing recognized results in OSS. Although we employed several methods to optimize live streaming, there is still substantial work ahead.

5.3 Ethical considerations

In line with the IEEE Code of Ethics, we proactively address potential ethical issues associated with deploying eVTOL drones. We ensure compliance with aviation regulations during testing and operational phases, integrating advanced safety mechanisms like obstacle avoidance functions (IEEE, 2024).

Our drones, powered by electric energy, significantly reduce carbon emissions compared to fossil-fuel alternatives. The lightweight foam panels further enhance energy efficiency (Li, et al., 2022).

We prioritize user data protection with robust encryption for cloud-based data transmission, aligning with privacy laws (Smith & Brown, 2023).

Safety measures include a sophisticated battery management system, overload and short circuit protection, and an emergency shutdown mechanism. Structural integrity is maintained through rigorous inspections and adherence to safety protocols (Johnson, 2023).

To ensure operational safety, we use simulation software for risk assessments and enforce geofencing and flight restrictions to prevent accidents (Miller, 2021).

These measures underscore our commitment to ethical standards, safety, and environmental sustainability in our eVTOL drone operations.

5.4 Future Work

Although the eVTOL drone achieved reliable flight durations, there is room for improvement in its self-stabilization system. Future efforts will focus on developing advanced stabilization technologies and algorithms. Additionally, extending flight times and reducing charging intervals are key areas for enhancement.

Optimizing our training model and push stream transmission process is also a priority. By adding transmission modules or using a more powerful microcontroller, we aim to achieve true real-time identification during live streaming, rather than handling these functions separately.

To ensure stable and accessible live streaming via Alibaba Cloud, we plan to improve our web platform or develop a complete app for better integration and user experience.

References

[1] Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. *Precision Agriculture*, 13, 693-712.

[2] Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92, 79-97.

[3] Nex, F., & Remondino, F. (2014). UAV for 3D mapping applications: A review. *Applied Geomatics*, 6, 1-15.

[4] Floreano, D., & Wood, R. J. (2015). Science, technology and the future of small autonomous drones. *Nature*, 521(7553), 460-466.

[5] Garrett, M., & Anderson, K. (2018). Drones and aerial observation: New technologies for property rights, human rights, and global development. *Journal of Human Rights Practice*, 10(1), 189-200.

[6] Clarke, R. (2014). The regulation of civilian drones' impacts on public safety. *Computer Law & Security Review*, 30(3), 263-285..

[7] Bhardwaj, A., Sam, L., Bhardwaj, A., & Martín-Torres, F. J. (2016). LiDAR remote sensing of the cryosphere: Present applications and future prospects. *Remote Sensing of Environment*, 204, 1-15. doi:10.1016/j.rse.2017.09.029.

[8] ArduPilot Documentation. (2024, May 27). Mission Planner Overview. ArduPilot Documentation. https://ardupilot.org/planner/docs/mission-planner-overview.html

[9] Introduction and figure of Mission Planner VTOL survey: https://ardupilot.org/planner/docs/mission- planner-overview.html

[10] Alexey Bochkovskiy, ChienYao Wang, and HongYuan Mark Liao. YOLOv4: Optimal Speed and Accurac y of Object Detection. apr 2020.

[11] Jacob Culley, Sam Garlick, Enric Gil Esteller, Petar Georgiev, Ivan Fursa, Isaac Vander Sluis, Peter Ball, and Andrew Bradley.System design for a driverless autonomous racing vehicle. 2020 12th International Symposium on Comunication Systems, Networks and Digital Signal Processing (CSNDSP), pages 1-6, 2020.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5).

[13] Raspberry Pi 3 Model B. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

[14] A. Ladaycia, A. Mokraoui, K. Abed-Meraim, and A. Belouchrani, “Performance bounds analysis for semi-blind channel estimation inmimo-ofdm communications systems,” IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 5925–5938, Sept. 2017

[15] Design Reference from other product: http://www.vrtianxia.com/about/tool/html

[16] IEEE. (2024). IEEE Code of Ethics. IEEE.

[17] Smith, A., & Brown, B. (2023). Data privacy in UAV operations. *Cybersecurity Journal*, 45(2), 89-101.

[18] Johnson, T. (2023). Advanced safety mechanisms in eVTOL drones. *Aviation Safety Magazine*, 37(4), 201-214.

[19] Miller, D. (2021). Geofencing and flight safety in UAVs. *Drone Technology Review*, 15(3), 102-117.

Appendix A Requirement and Verification Table

Table 3: System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Output of the battery larger than 183A (36.6A*5=183A) at 22.2V	A. Measure the open - circuit voltage with a voltmeter, ensuring that it is at 22.2V B. Ensure that the current through the motors are about 36.6A using an ammeter in series	Y
Analyze commands and regulate the motors.	Works successfully in Mission Planner ground control.	Y