University of Hertfordshire
School of Physics, Engineering and Computer Science

Project Title: Crowd Surveillance Drone

Bachelor of Engineering in Robotics and Artificial Intelligence

Name: Daniel Tan Jian Song

Student ID: 390BEFGR (PSB)

Module: BEng Individual Major Project

Module Code: 6FTC2062

Date: 23th August 2024

School of Physics, Engineering and Computer Science

Table of Contents

List of Figures

University of Hertfordshire

School of Physics, Engineering and Computer Science

Acknowledgements

First and foremost, I am deeply grateful to all the lecturers who have taught and guided our batch of robotics students over the past two years. Their collective effort has created a nurturing and inspiring environment, allowing us to learn and grow as aspiring engineers.

I would also like to extend my sincere thanks to my project supervisor, Mr. Daniel Lim, for his invaluable feedback and suggestions, which helped me improve in areas where I was less confident.

Additionally, I would like to express my appreciation to my classmate, Mr. Zan Xuan, for his guidance and advice on the chassis design. His support was instrumental in the successful completion of this aspect of the project.

University of Hertfordshire

School of Physics, Engineering and Computer Science

Glossary/ Abbreviations/ Nomenclature

SSD: Single Shot Detector (a type of object detection algorithm)
mAh: Milliampere-hour
FOV: Field of View
PID: Proportional Integral Derivative
ESC: Electronic Speed Controller
IMU: Inertial Measurement Unit
CNN: Convolutional Neural Network
UAV: Unmanned Aerial Vehicle
POI: Person of Interest
DIY: Do It Yourself
HOG: Histogram of Oriented Gradients
SVM: Support Vector Machine
LBPH: Local Binary Patterns Histogram
YOLO: You Only Look Once
VOC: Visual Object Classes
LiPo: Lithium Polymer
GPS: Global Positioning System
EMI: Electromagnetic Interference
LiDAR: Light Detection and Ranging
CAAS: Civil Aviation Authority of Singapore
IEEE: Institute of Electrical and Electronics Engineers
IEC: International Electrotechnical Commission

University of Hertfordshire UH

School of Physics, Engineering and Computer Science

***Abstract*: This project presents the development and implementation of a crowd surveillance drone with the primary objective of identifying a person of interest within a crowd. Unlike conventional surveillance systems that monitor entire crowds, this drone is designed to specifically seek out individuals based on facial recognition and object identification. The drone is built on a quadcopter platform powered by a Raspberry Pi 3, equipped with a Raspberry Pi Camera V2, and integrates the MobileNetV2 model for facial recognition and SSD-MobileNet for object detection. Flight control is managed by an Arduino Uno, utilising the Aeroquad flight control algorithm to ensure stable and responsive operation. This project showcases autonomous drone technology potential applications in security and surveillance.**

Introduction and Background

The advancement of UAVs, particularly drones, has revolutionised various fields, including security, surveillance, and search and rescue operations. Among these applications, crowd surveillance is becoming increasingly important in maintaining public safety at large events, protests, and gatherings [1]. Traditional surveillance systems typically focus on monitoring entire crowds, capturing vast amounts of data, often at the expense of missing critical details about specific individuals within the crowd[2]. This project aims to address this limitation by developing a drone-based crowd surveillance system designed to identify and track a POI within a crowd, rather than monitoring the crowd as a whole.

The use of drones in surveillance offers several advantages over traditional methods. Drones provide mobility, allowing for dynamic observation of a crowd from various angles and altitudes. Additionally, they can cover large areas quickly and can be equipped with various sensors and cameras to capture high-quality data. However, the integration of real-time facial recognition and object identification into drones presents a new frontier in surveillance technology, enabling drones not only to observe but also to analyse and respond to specific situations autonomously.[3]

In recent years, machine learning models such as MobileNetV2 and SSD-MobileNet have become increasingly popular for facial recognition and object detection due to their accuracy and efficiency[4][5]. MobileNetV2, a lightweight CNN, is particularly well-suited for deployment on resource-constrained devices like the Raspberry Pi, offering decent accuracy with low computational demands[6].The Aeroquad flight control algorithm, implemented on the Arduino Uno, provides a stable and responsive control system for the quadcopter. This open-source flight control software has been extensively tested and refined, making it a reliable choice for drone projects requiring precise manoeuvrability and stability.[7]

By leveraging the capabilities of a Raspberry Pi 3 and a Raspberry Pi Camera V2, the drone is able to process visual data in real-time, identifying specific individuals based on pre-trained machine learning models. The project also addresses the technical challenges of real-time processing, limited computational resources, and the need for reliable flight control in dynamic environments.

University of Hertfordshire UH

School of Physics, Engineering and Computer Science

Literature Review

UAVs in Surveillance

(UAVs), commonly known as drones, have become a critical tool in surveillance applications due to their versatility, mobility, and ability to cover large areas quickly. The use of drones in surveillance is well-documented in the literature, with various studies highlighting their effectiveness in applications such as border monitoring, disaster management, and crowd surveillance[8].

Crowd surveillance is a complex task that involves monitoring large gatherings to ensure safety, detect unusual activities, and identify specific individuals. Traditional surveillance methods rely heavily on static cameras and human operators, which can be limited in scope and effectiveness, particularly in dynamic environments. UAVs offer a more flexible solution, as they can move freely, providing real-time aerial views of crowds from various angles and altitudes[9]. Drones equipped with cameras and sensors can significantly enhance the efficiency of crowd surveillance by offering a broader perspective and the ability to zoom in on specific individuals or areas of interest[10].

Despite their advantages, UAV-based surveillance systems face several challenges. One significant issue is the limited computational power available on-board UAVs, which constrains the complexity of real-time processing tasks, such as facial recognition and object detection[11]. Moreover, the need for reliable and stable flight control, particularly in crowded or cluttered environments, presents additional challenges.Thus, this emphasises the importance of integrating advanced machine learning algorithms that are optimised for performance on low-power devices, such as the Raspberry Pi, commonly used in UAVs.

Machine Learning for Facial Recognition and Object Detection

Machine learning, particularly deep learning, has revolutionised facial recognition and object detection, enabling significant advancements in accuracy and real-time performance[12]. The use of Convolutional Neural Networks (CNNs) has become the standard approach in these domains, with models such as MobileNetV2 and SSD-MobileNet being widely adopted for their balance of speed and accuracy[13].

MobileNetV2,is a lightweight CNN architecture optimised for mobile and embedded vision applications. It is particularly well-suited for facial recognition tasks in resource-constrained environments, such as those found in UAV systems. MobileNetV2 employs depth wise separable convolutions to reduce the number of parameters and computational complexity, making it an ideal choice for deployment on devices like the Raspberry Pi 3[15][16]. MobileNetV2 in real-time facial recognition applications, even when running on limited hardware, is able to maintain high accuracy while minimising computational requirements, this makes it a key component in this project.

Object detection is another critical aspect of crowd surveillance, enabling the identification of various objects and individuals within a scene. The SSD-MobileNet model, a combination of Single Shot MultiBox Detector (SSD) and MobileNet, offers a fast and efficient solution for real-time object detection[5]. SSD is a method that eliminates the need for a region proposal network, allowing object detection to be performed in a single forward pass of the network, significantly speeding up the process[17]. When combined with MobileNet, SSD provides a lightweight yet powerful object detection framework that is well-suited for use in UAVs.

While models like MobileNetV2 and SSD-MobileNet are effective, their deployment in real-time applications on UAVs requires careful optimization. Issues such as model size, inference time, and power consumption are critical considerations. Techniques such as quantization, model pruning, and hardware acceleration have been explored in the literature to address these barriers[18].

UAV Flight Control Systems

The flight control system is a crucial component of any UAV, responsible for maintaining stability, responding to commands, and ensuring safe operation. The Aeroquad flight control algorithm, used in this project, is an open-source solution that has been extensively tested and refined for use in quadcopter drones.

Aeroquad is one of the most popular open-source flight control systems for quadcopters. It offers a range of features, including PID control, sensor integration, and support for various flight modes. The system is designed to run on microcontrollers like the Arduino Uno, making it accessible for DIY and academic projects[19]. The Aeroquad algorithm provides robust control, even in challenging flight conditions, by integrating data from multiple sensors, such as accelerometers, gyroscopes, and magnetometers.

Maintaining stable flight in dynamic environments, such as crowded areas, presents significant challenges. Wind disturbances, obstacles, and the need for precise manoeuvring require advanced control algorithms.Adaptive control techniques and sensor fusion[20] are critical in enhancing the performance of UAV flight control systems. These techniques enable the drone to adjust its flight parameters in real-time, ensuring stability and accuracy in varying conditions.

For UAVs to operate autonomously in surveillance applications, seamless integration between the flight control system and the onboard processing unit is essential. This integration allows the UAV to respond to real-time data, such as facial recognition results, by adjusting its flight path or hovering over a person of interest.

Applications and Ethical Considerations

UAVs equipped with facial recognition and object detection capabilities have the potential to transform public safety operations. They can be deployed in scenarios such as large public events, protests, or disaster zones to quickly identify and track individuals, assist in crowd control, and provide real-time data to law enforcement.

The use of facial recognition technology in surveillance however, raises significant ethical concerns, particularly regarding privacy and the potential for misuse.The literature emphasises the need for strict regulatory frameworks and transparency in the deployment of these technologies[21]. The balance between security and privacy must be carefully managed to prevent the erosion of civil liberties. Furthermore, the potential for bias in facial recognition algorithms underscores the importance of ensuring that these systems are fair and do not disproportionately impact certain groups.[22]

This literature review has explored the key technologies and challenges associated with the development of a crowd surveillance drone capable of identifying a person of interest within a crowd. The review highlights the significant advancements in UAV technology, machine learning for facial recognition and object detection, and UAV flight control systems. It also underscores the importance of addressing ethical and legal considerations when deploying such systems in real-world scenarios. The integration of these technologies into a cohesive surveillance system, as proposed in this project, represents a significant contribution to the field of security and surveillance, offering a more targeted and efficient approach to crowd monitoring.

Project Work and Methodology

<u>Hardware Design</u>

(Engineering Drawing of 3D printed Components)

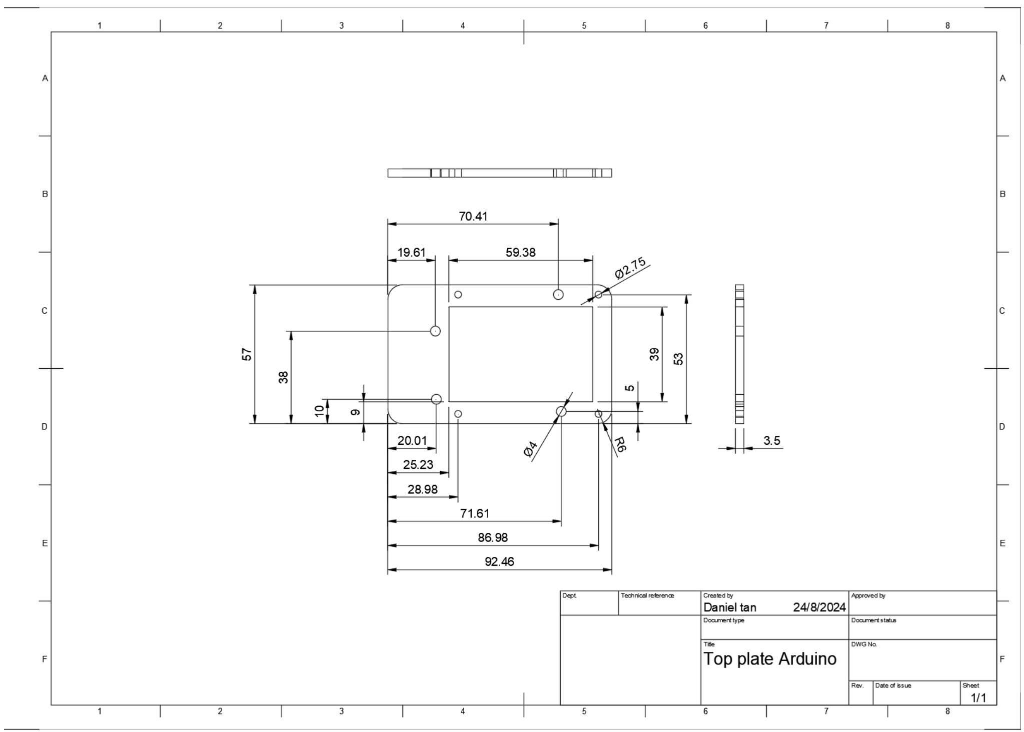

Fig.1 Plate Holder for Arduino Uno

University of Hertfordshire

School of Physics, Engineering and Computer Science

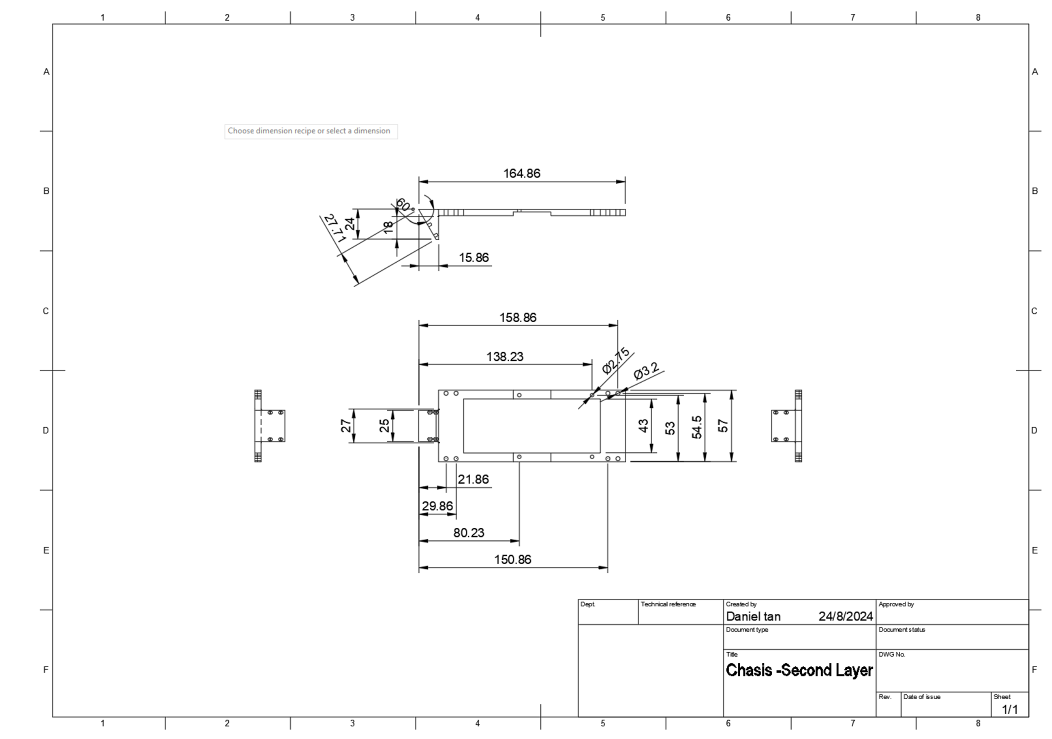

Fig.2 Center Plate with Camera Mount

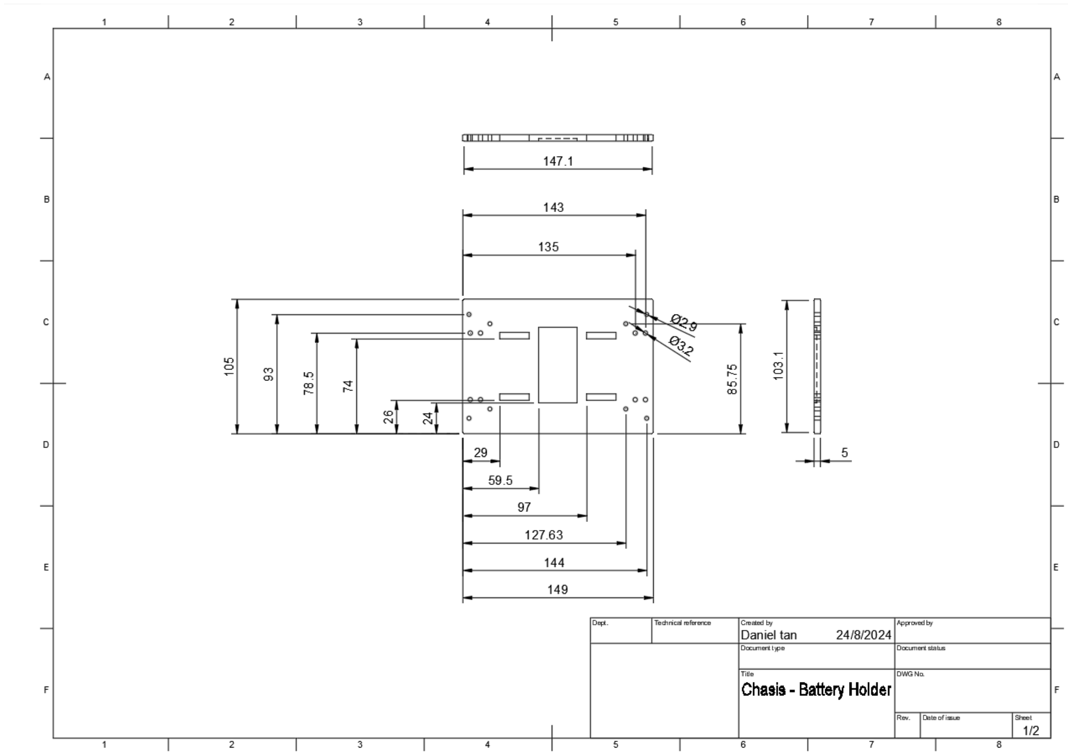

Fig.3 Base Plate of Drone (Holds Battery and Buck Converter)

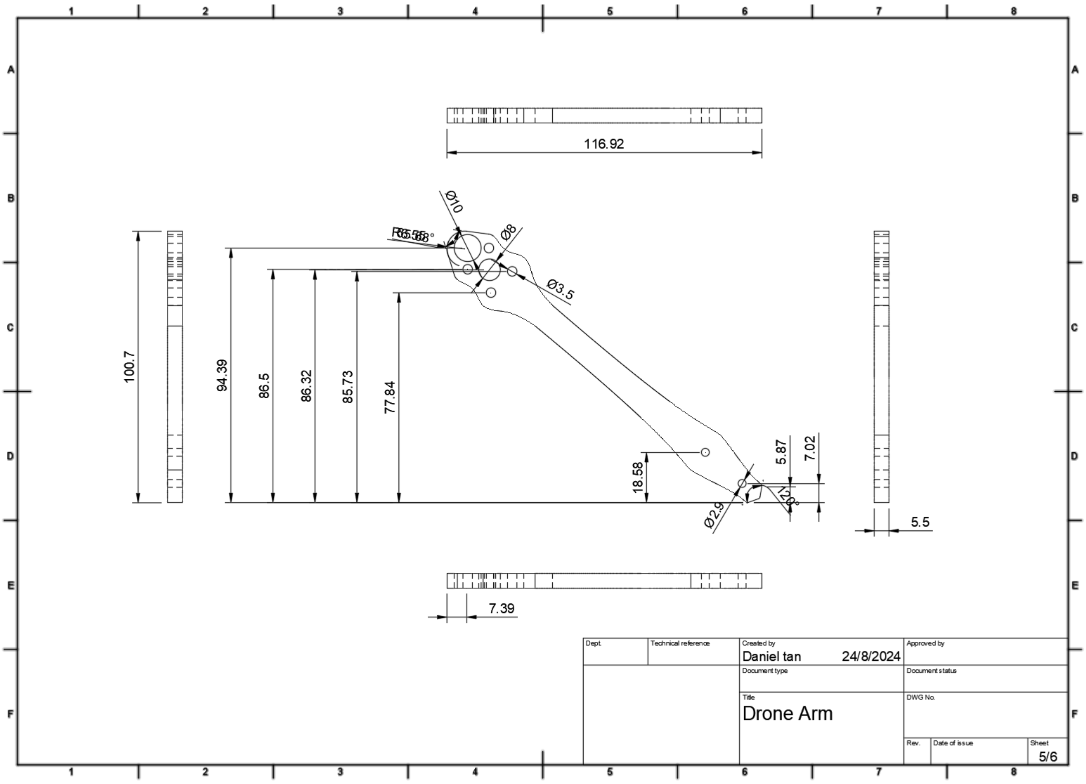

Fig.4 Drone Arm Design

(Hardware Architecture)

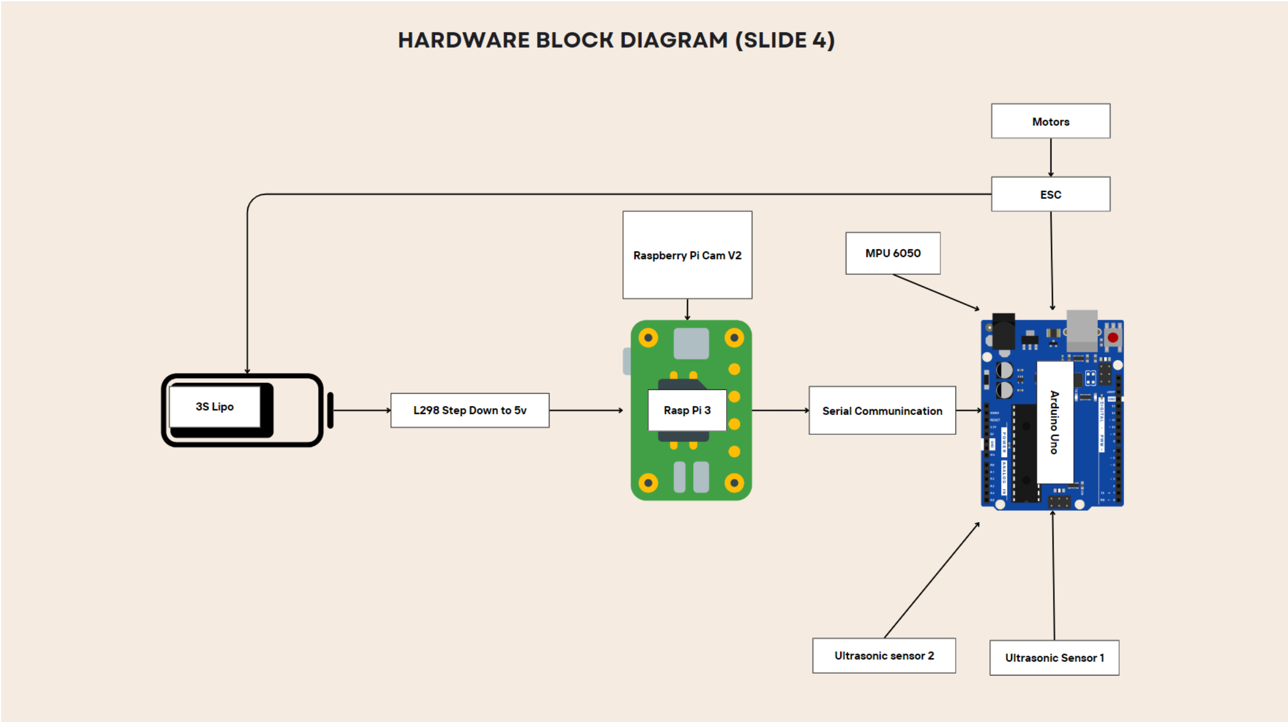

Fig.5 Drone Hardware Block Diagram

- **3S LiPo Battery:** Lipo Battery has an output voltage of 11.1v, it is the primary power source for the drone with components (Raspberry Pi 3, Arduino Uno, ESC, Ultrasonic Sensors, Motors, Gyro and Pi Camera)

- **LM2596 Buck Converter:** Voltage Converter used to step down the 11.1V from the LiPo battery to 5V, which is suitable for powering the Raspberry Pi 3

- **Raspberry Pi 3:** The central processing unit for the drone, responsible for running the facial recognition and object identification algorithms. It is connected to the Raspberry Pi Camera V2 for real-time image capture

- **Raspberry Pi Camera V2:** A high-resolution camera module connected to the Raspberry Pi 3, used for capturing video frames for facial recognition and object detection

- **Arduino Uno:** Microcontroller board that manages the flight control of the drone. It communicates with the Raspberry Pi via serial communication to receive commands and sends PWM signals to the ESC to control the motors. It also reads data from Ultrasonic sensors 1 & 2 to adjust flight controls accordingly

- **MPU 6050:** An IMU sensor connected to the Arduino Uno. It provides orientation data, which is crucial for maintaining stable flight

- **Ultrasonic Sensors (1 & 2):** These sensors are used for obstacle detection and altitude maintenance. Ultrasonic Sensor 1 assists with proximity-based landing, while Ultrasonic Sensor 2 aids in altitude correction during flight

- **ESC:** Critical components that translate the PWM signals from the Arduino Uno into specific motor speeds, enabling precise control over the drone’s movement. Each motor is connected to its respective ESC, which regulates the power delivered to the motors, thus controlling the drone's lift and manoeuvrability

- **Motors:** Responsible for generating the thrust needed for the drone's flight, each motor is connected to its respective ESC

- **Chassis Design:** Took inspiration from the apex drone, used Fusion 360 for stress test as drone chassis design is also done in the same app. In Fusion 360 simulation workspace, ensure proper setting of material properties. Define where the chassis is fixed or supported and apply forces or pressures to the model.

University of Hertfordshire UH

School of Physics, Engineering and Computer Science

(Software Architecture)

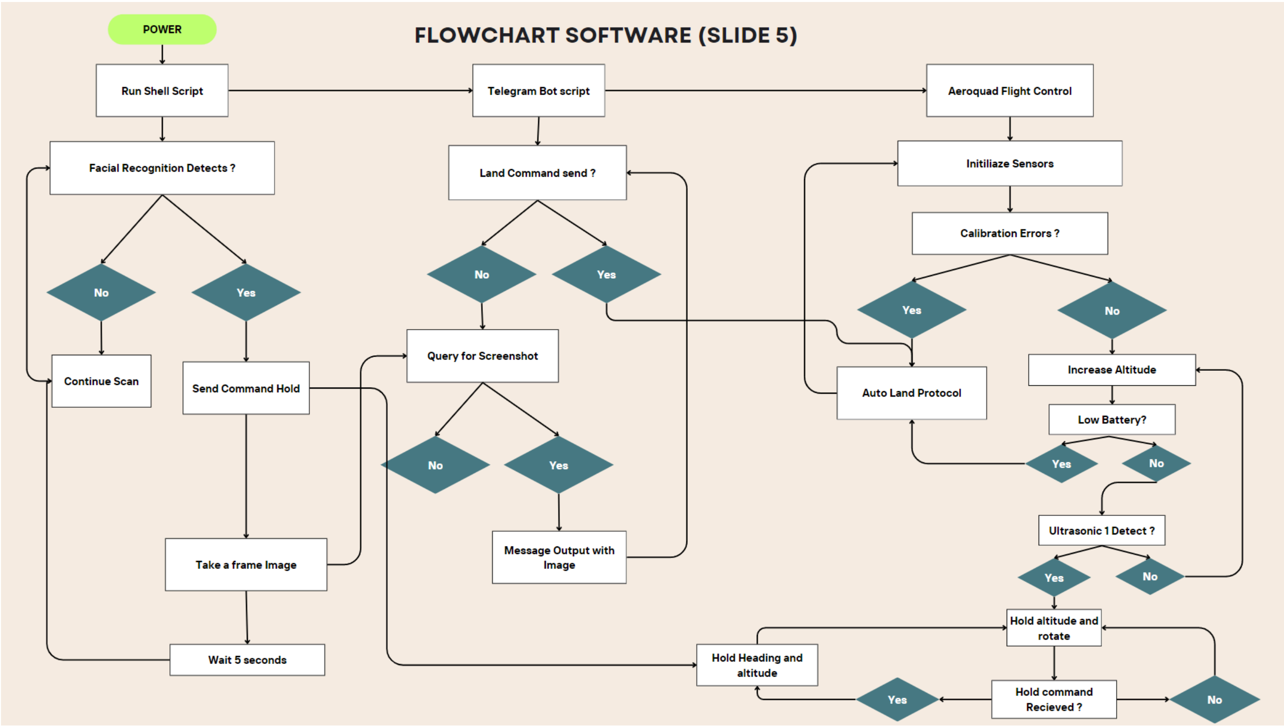

Fig.6 Drone Software Flowchart

- **Shell Script Execution:** Upon powering up, a shell script is executed to initialise the various software components, including the facial recognition, object detection and Telegram Bot Script

- **Facial Recognition and Object Detection:** The Raspberry Pi 3 runs a Python script that utilises the MobileNetV2 model for facial recognition and the SSD-MobileNet model for object detection. If a person of interest is detected, the system sends a "hold" command to the Arduino Uno

- **Telegram Bot Integration:** A Telegram bot script is also run to send notifications and images to the user. If a "land" command is issued, the system queries for a screenshot and sends the image along with a message to the user

- **Aeroquad Flight Control:** Arduino Uno manages the drone's flight using the Aeroquad flight control algorithm. The control loop includes initialising sensors, handling calibration errors, adjusting altitude, and responding to obstacle detection via ultrasonic sensors

- **Command Processing:** The system processes various commands such as "hold," "land," and "rotate," adjusting the drone's behaviour accordingly. For example, if a hold command is received, the drone stabilises and maintains its current position

- **IMU Calibration (Gyroscope):** The calibrateGyro() function in Aeroquad calibrates the gyroscope by calculating zero-point offsets for each axis (X, Y, Z). It captures raw gyroscope data multiple times, stores the values, and calculates the median for each axis to determine the offset. If the difference between readings is within the defined threshold, the median is accepted as the gyro zero. If successful, the function returns true, indicating the gyroscope is calibrated and ready for accurate measurements

- **Transfer Learning with MobileNetV2:** In order to implement a robust facial recognition system on the Raspberry Pi, transfer Learning with MobileNetV2 was utilised. The base MobileNetV2 model, initialised with pre-trained weights from ImageNet, was used with its pre-trained layers frozen to retain general features. A new fully connected layer, specifically tailored for facial recognition, was added and trained on a 160-image dataset using a small learning rate. The Adam optimizer and categorical cross-entropy loss were employed to fine-tune the new layers. The model was then evaluated on a separate validation set to assess its accuracy, precision, recall, and F1-score

(Model Training)

For my project, a dataset which consisted of 160 images of my own face was added. Out of the 160 images, 40 of them were taken under low lighting, 60 of them were taken with a laptop camera and 60 images were captured using the Raspberry Pi Cam V2.

The base MobileNetV2 model, initialised with pre-trained weights from ImageNet, was used with its pre-trained layers frozen to retain general features. A new fully connected layer, specifically tailored for facial recognition, was added and trained on a 160-image dataset using a small learning rate. The Adam optimizer and categorical cross-entropy loss were employed to fine-tune the new layers. The model was then evaluated on a separate validation set to assess its accuracy, precision, recall, and F1-score[23].

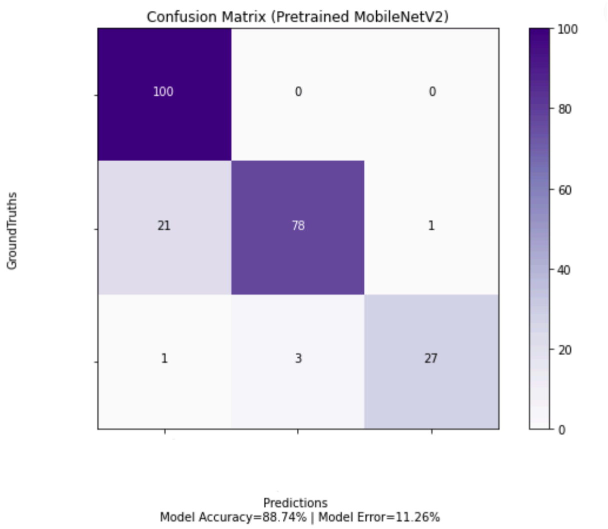

Fig.7 Confusion Matrix of Model Used

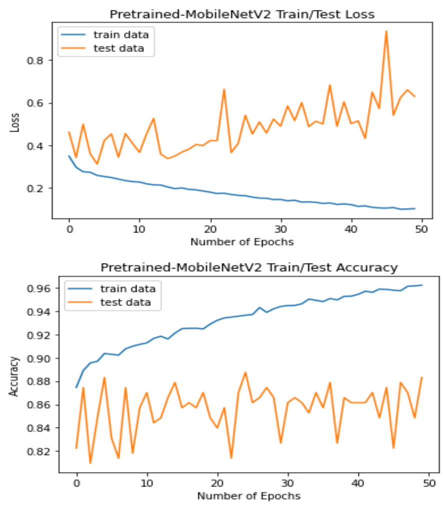

Fig.8 Training Accuracy of MobileNet

(Final Drone Design)

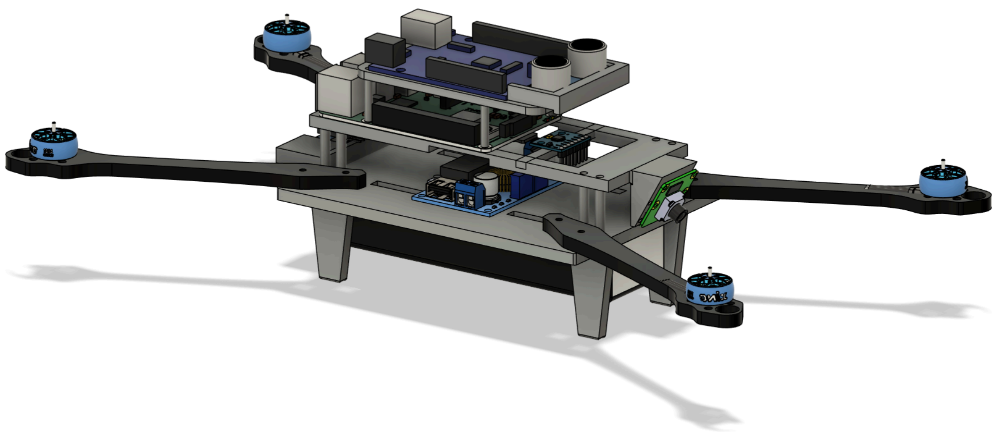

Fig.9 Fusion 360 Drone Impression (Prebuild) - used for stress test

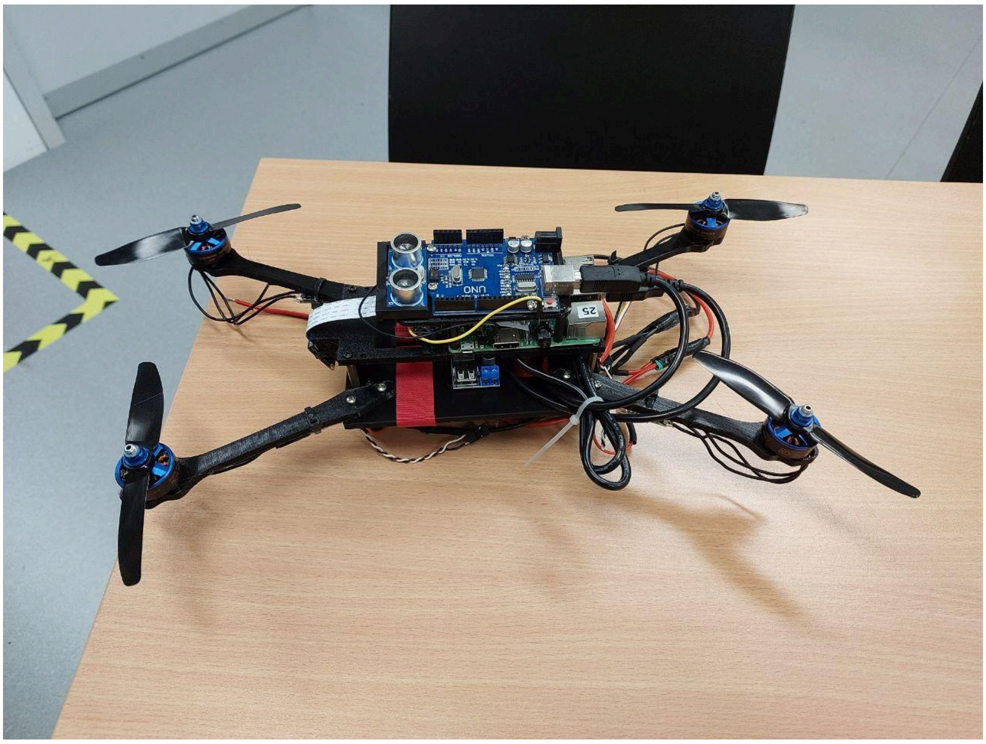

Fig.10 Drone Actual Design

Discussion and Results

System Performance

Drone's overall performance was evaluated in terms of its ability to autonomously identify and track a person of interest within a crowd. The system was designed to function in dynamic environments, where the drone needed to respond to real-time data from its sensors and adjust its flight accordingly.

The Raspberry Pi 3, running the MobileNetV2 and SSD-MobileNet models, demonstrated a satisfactory level of real-time processing capability. The drone was able to detect and recognize faces within 2-3 seconds of scanning, which is an acceptable response time for surveillance applications. However, under heavy computational load or in complex environments with multiple faces, the processing speed slightly decreased, resulting in a 1-2 second delay in recognition. This performance is consistent with the limitations of the Raspberry Pi 3's processing power, suggesting that future iterations of the project could benefit from a more powerful onboard processor or the addition of hardware accelerators.

The accuracy of the facial recognition system was a critical measure of success. The MobileNetV2 model achieved a recognition accuracy of approximately 88.74% in controlled environments with good lighting. This accuracy dropped slightly to around 82.6% in more challenging conditions, such as low light.

Choice of Model (Facial Recognition)

The decision to use MobileNetV2 over other models like Dlib, DeepFace, ResNet, and LBPH was driven by several key factors:

Model	Dataset	F1-score	Recall	Precision	Accuracy
MobileNetV2	VGGFace2	0.93	0.92	0.94	93%
Dlib (HOG + SVM)	LFW	0.87	0.86	0.88	87%
ResNet	CASIA-WebFace	0.96	0.97	0.97	96%
LBPH	AT&T Face Dataset	0.79	0.80	0.80	79%

Note: The above values are representative of general findings from [24][25][26][27]

Efficiency and Speed: MobileNetV2 is optimised for efficiency, utilising depthwise separable convolutions to minimise computational demands, making it ideal for low-powered devices like the Raspberry Pi 3. In contrast, ResNet, although highly powerful, is more computationally intensive, leading to slower processing times, which may not be suitable for real-time applications on a Raspberry Pi. Dlib (HOG + SVM) and LBPH offer quicker processing times but sacrifice accuracy and robustness, particularly in complex scenarios, in favour of being lightweight and less computationally demanding

Balanced Performance: MobileNetV2 offers a strong balance between accuracy and computational efficiency, making it ideal for real-time applications on resource-constrained devices like the Raspberry Pi 3. ResNet, while achieving the highest accuracy, is too computationally demanding for the Raspberry Pi 3, potentially causing delays. Dlib and LBPH are more resource-efficient but lack the accuracy of MobileNetV2, making them less suitable for tasks where precision is critical

Proven Track Record: MobileNetV2 has a strong reputation for delivering reliable performance in mobile and embedded systems, particularly on devices with limited processing power like the Raspberry Pi 3. In contrast, ResNet, known for its accuracy in large-scale applications, is less suited for low-power environments due to its high resource requirements. Dlib (HOG + SVM) and LBPH are effective in simpler applications but fall short in accuracy, making them less suitable for more demanding facial recognition tasks compared to the more advanced MobileNetV2

Choice of Model (Object Identification)

The SSD-MobileNet model used for object detection performed similarly well, accurately identifying objects within the scene. These results demonstrate that the chosen models are effective for the intended application, though there is room for improvement, particularly in handling more complex and cluttered environments.

This comparison highlights why SSD-MobileNetV2 is a suitable choice for object identification on the Raspberry Pi 3:

Model	Dataset	F1-Score	Recall	Precision	Accuracy
SSD-MobileNetV2	COCO	0.81	0.83	0.80	82%
YOLOv3-Tiny	COCO	0.78	0.76	0.79	78%
TinyYOLOv4	VOC	0.85	0.84	0.86	84%
EfficientDet-Lite0	COCO	0.80	0.81	0.79	80%
MobileNetV1 SSD	PASCAL VOC	0.76	0.74	0.77	75%

Note: The above values are representative of general findings from [28][29][30][31][17]

Efficiency and Speed: SSD-MobileNetV2 is highly efficient, making it one of the fastest object detection models for low-powered devices like the Raspberry Pi 3. Its architecture merges the speed of MobileNetV2 with SSD's single-shot detection capabilities, allowing quick frame processing with minimal computational load. In comparison, YOLOv3-Tiny and TinyYOLOv4 are also optimised for speed but may require slightly more processing power, especially in multi-object scenarios. EfficientDet-Lite0, while efficient, is slower than SSD-MobileNetV2 due to its more complex feature pyramid network, making it less ideal for real-time use on resource-constrained devices

Balanced Performance: SSD-MobileNetV2 offers a well-balanced performance on the Raspberry Pi 3, with a good trade-off between speed and accuracy, achieving an F1-score of 0.81 and 82% accuracy. This makes it a reliable option for most object detection tasks in resource-constrained environments. TinyYOLOv4 provides slightly higher accuracy (84%) but may strain the hardware, affecting real-time processing. YOLOv3-Tiny, while a bit less accurate, excels in near real-time performance, making it a strong competitor to SSD-MobileNetV2. EfficientDet-Lite0 also delivers balanced performance, though its slightly lower speed might limit its use in real-time applications

Proven Track Record: SSD-MobileNetV2 is highly regarded in mobile and embedded applications, particularly for object detection on low-powered devices, demonstrating reliability across various real-world tasks. It has become a staple in mobile vision applications due to its balance of speed and accuracy. YOLOv3-Tiny and TinyYOLOv4 are also popular for scenarios requiring fast object detection with decent accuracy, especially where quick responses are essential. EfficientDet-Lite0, while newer and showing promise in scalability and efficiency, hasn't yet reached the same level of adoption as SSD-MobileNetV2 or the YOLO models in low-power settings. The earlier MobileNetV1 SSD has been largely replaced by SSD-MobileNetV2, which offers better performance in both speed and accuracy

Flight Control Performance

Ultrasonic Sensor 1, used primarily for proximity-based landing, consistently detected obstacles within a range of 30 cm. However, the performance varied depending on the surface reflectivity and environmental conditions. Ultrasonic Sensor 2, used for controlled descent during landing protocol had readings which were affected by interference, leading to minor fluctuations in altitude. Despite this, the drone successfully avoided obstacles and maintained a safe distance from nearby objects.

The system's ability to respond to user commands, such as "hold," "land," and "rotate," was tested extensively. The drone's response time to these commands was generally within 1 second, with the "hold" command being the most critical for surveillance operations. The drone's ability to maintain its heading and altitude after receiving the "hold" command was consistent, with no significant drift observed. The "land" command, triggered by low battery levels or user input, initiated the auto-landing protocol, which was executed smoothly in all test scenarios.

Choice of Flight Control Algorithm

AeroQuad is particularly well-suited for projects involving low-powered devices like the Arduino Uno due to its simplicity, low hardware requirements, and ease of use. AeroQuad's ability to operate efficiently on the Arduino platform, which has limited processing power compared to more advanced flight controllers like those used with PX4 or ArduPilot.

The comparison table below shows the difference between the flight control algorithms which are commonly used:

Feature	PX4	ArduPilot (ArduCopter)	AeroQuad
Supported Platforms	STM32, NuttX, Linux, ROS	STM32, Linux, Pixhawk, BeagleBone	Arduino Uno, Mega
Flight Modes	Manual, Stabilised, Autonomous, Acro, etc.	Manual, Stabilised, Autonomous, Loiter, RTL, etc.	Manual, Stabilised
Hardware Requirements	High (requires powerful flight controllers)	High (supports advanced sensors and controllers)	Low (runs on Arduino, limited processing power)
Flexibility and Customization	High (modular, ROS integration, advanced features)	High (extensive mission planning, customization)	Medium (customizable within Arduino limitations)
Learning Curve	Steep (complex, feature-rich)	Steep (many features and configurations)	Moderate (simpler, Arduino-friendly)
Supported Vehicles	Multirotors, Fixed-wing, VTOL, Rovers, Boats	Multirotors, Fixed-wing, VTOL, Rovers, Boats	Multirotors (primarily quadcopters)
Open Source	Yes, active community	Yes, large and active community	Yes, community-driven, Arduino-focused

Comparison with PX4 and ArduPilot

Hardware Compatibility: Unlike PX4 and ArduPilot, which require more powerful flight controllers and often more advanced hardware setups, AeroQuad is designed specifically for the Arduino platform. This makes it a perfect choice for projects that use the Arduino Uno, where minimal hardware overhead is crucial[32]

Ease of Implementation: The learning curve for AeroQuad is moderate and more approachable compared to the steep learning curve associated with PX4 and ArduPilot. This simplicity makes AeroQuad more accessible for users who are focused on getting their project up and running quickly without needing to delve into the complexities of advanced autonomous flight modes or extensive configuration options that PX4 and ArduPilot offer[33]

Customizability: While AeroQuad offers a medium level of customization within the constraints of the Arduino platform, it is more than sufficient for most basic to intermediate drone projects. PX4 and ArduPilot, on the other hand, offer extensive customization and flexibility, but this often comes at the cost of complexity and the need for more robust hardware[34]

Conclusions

The crowd surveillance drone project successfully demonstrated the integration of advanced computer vision techniques and UAV technology to create a targeted and efficient surveillance system capable of identifying and tracking a person of interest within a crowd. The project leveraged the Raspberry Pi 3 as the primary processing unit, utilising the MobileNetV2 model for facial recognition and the SSD-MobileNetV2 model for object identification, both of which proved to be well-suited for real-time applications on resource-constrained hardware.

The selection of MobileNetV2 for facial recognition was pivotal, as it provided a balance between accuracy and computational efficiency, enabling the system to perform reliably on the Raspberry Pi 3. Similarly, SSD-MobileNetV2 was chosen for object identification due to its streamlined architecture, which facilitated quick processing times without overwhelming the limited resources of the Raspberry Pi. These models were instrumental in achieving the project's goal of real-time surveillance, ensuring that the drone could effectively scan, identify, and follow the person of interest within a crowded environment.

The decision to implement the AeroQuad flight control algorithm on the Arduino Uno further underscored the project's focus on efficiency and simplicity. AeroQuad, designed specifically for low-powered platforms like the Arduino, provided stable and responsive control of the drone, ensuring consistent flight performance even with the limited processing capabilities of the Arduino Uno. The drone's ability to maintain a steady hover, respond to user commands, and avoid obstacles in real-time was a testament to the effectiveness of AeroQuad in this context.

Throughout the project, several key challenges were addressed, including the limitations of hardware processing power, the need for reliable real-time data processing, and the integration of various software and hardware components into a cohesive system. The project successfully navigated these challenges, resulting in a robust and functional surveillance drone that met its primary objectives.

The drone was tested in various scenarios, demonstrating its capability to identify and track individuals within a crowd while maintaining flight stability and responding to user inputs. The integration of a Telegram bot for real-time notifications and image transmission added a valuable layer of user interaction, further enhancing the system's practicality for real-world applications.

In summary, this project represents a significant achievement in the development of a cost-effective and efficient crowd surveillance system. By combining the strengths of advanced machine learning models, low-powered computing platforms, and a streamlined flight control system, the project successfully met its objectives, paving the way for further advancements in the field of UAV-based surveillance. The results highlight the potential of such systems to enhance public safety and security in various contexts, offering a targeted and efficient approach to monitoring large crowds and identifying individuals of interest.

Further Development

As the crowd surveillance drone project has successfully demonstrated the integration of basic machine learning models and flight control algorithms on a low-powered platform, several avenues for future development and enhancement are evident. These improvements aim to increase the drone's autonomy, enhance its processing capabilities, and provide a more user-friendly experience. The following key areas outline potential directions for future work:

Autonomous Navigation and Tracking:

Implementing the PX4 flight control system would be a significant step in this direction, as it supports a wide range of advanced sensors, including barometers, GPS, and magnetometers, which are essential for autonomous operations. By integrating these sensors, the drone could autonomously navigate complex environments, follow a predefined path, and track a person of interest without human intervention. This would not only enhance the drone's functionality but also expand its application potential in areas like search and rescue, wildlife monitoring, and large-scale event surveillance.

Better Energy Efficiency and Power Management:

Energy efficiency is critical for extending the operational time of the drone, particularly in surveillance scenarios where the drone may need to hover and monitor an area for extended periods. Future work could focus on optimising power consumption through better power management strategies. This might include the integration of more efficient power regulation circuits, the use of higher capacity or smart LiPo batteries, and the implementation of low-power modes during periods of inactivity. Additionally, exploring solar power or other renewable energy sources could further extend the drone's flight time, making it more suitable for long-duration missions.

Enhanced Image Processing and AI Capabilities:

The current system's image processing and AI capabilities, while adequate, are limited by the processing power of the Raspberry Pi 3. Future upgrades could involve transitioning to more powerful platforms such as the Raspberry Pi 4 or NVIDIA Jetson Nano. These platforms offer significantly more processing power, enabling the implementation of more advanced AI models and real-time processing of higher resolution video streams. This could lead to improved accuracy in facial recognition, faster object detection, and the ability to handle more complex scenarios, such as multi-object tracking and predictive analytics. Moreover, integrating advanced AI techniques like deep reinforcement learning could further enhance the drone's decision-making capabilities.

Upgrade to a Dedicated Flight Controller:

The AeroQuad flight control algorithm, while effective for the current setup, has limitations in terms of advanced flight capabilities and sensor integration. Upgrading to a dedicated flight controller like PX4 would unlock the potential for more sophisticated flight operations, including precise altitude control with barometers, advanced obstacle avoidance with LiDAR, and waypoint navigation using GPS. The addition of LiDAR sensors, in particular, would greatly improve the drone's ability to detect and avoid obstacles in real-time, enabling safer and more reliable autonomous flights in cluttered or dynamic environments. This upgrade would transform the drone from a basic surveillance tool into a more versatile and capable UAV platform.

Enhanced User Interface and Interaction:

The current user interaction is primarily facilitated through a Telegram bot, which, while functional, could be significantly enhanced by developing a dedicated mobile application. This application could provide users with a more intuitive and comprehensive control interface, including real-time video feeds, drone status monitoring, and the ability to set specific search parameters or areas of interest. The mobile app could also offer advanced features such as flight path planning, dynamic rerouting based on real-time data, and the ability to control multiple drones simultaneously. This would make the system more accessible to users, expanding its utility in various applications, from security and surveillance to research and environmental monitoring.

University of Hertfordshire UH

School of Physics, Engineering and Computer Science

Project Management Review

1. Ghantt Chart Comparison

The comparison highlights the critical activities necessary for the project's success, such as drone chassis design, model sourcing, and training. A significant difference between the two charts arose when it was discovered that the originally planned model, TinyYOLOv4, was too computationally expensive for the Raspberry Pi 3. This realisation led to a necessary timeline adjustment and a switch to MobileNetV2, which better suited the hardware’s limitations. This change, however, set the project back by two weeks.The experience underscored the importance of incorporating extra time into the planning phase to account for potential setbacks and failures, allowing for more realistic timelines in future projects.

2. Key Changes and Justifications

One of the most significant changes in the project was the shift from TinyYOLOv4 to MobileNetV2 for facial recognition. TinyYOLOv4, initially chosen for its accuracy and real-time capabilities, proved too demanding for the Raspberry Pi 3, causing delays and processing issues. The switch to MobileNetV2, which offers a better balance of efficiency and accuracy, was necessary to meet the project's performance requirements, despite setting the project back by approximately two weeks. This adjustment was crucial to ensure that the final system operated effectively within the constraints of the hardware.

3. Strengths and Weaknesses in Management

As a project manager, I realised that I underestimated the complexity of building a quadcopter, especially the integration of software with flight mechanics and power calculations. However, I was particularly satisfied with my ability to recover and get the project back on track after the failure of implementing TinyYOLOv4. A key area for improvement is allowing more time to fully explore and verify potential solutions before committing, as well as planning for possible setbacks that could impact the project timeline.

4. Areas for Improvement and Mitigation Strategies

Several areas for improvement were identified, particularly in quality management. The initial sensor calibration did not account for environmental variations, such as temperature changes. Future iterations should incorporate more robust calibration procedures, possibly with real-time adjustments. Additionally, more rigorous testing protocols are necessary to identify software bugs and hardware failures early on. Implementing automated testing frameworks and conducting extensive field tests under varying conditions would enhance the quality and reliability of the drone, ensuring compliance with higher safety and performance standards.

School of Physics, Engineering and Computer Science

Quality Management Review

1. Role of Relevant Standards in Relation to the Project

In engineering, standards play a crucial role in ensuring the safety, reliability, and quality of products and services. These standards are established by professional organisations and regulatory bodies to provide a framework for consistent practices and to prevent engineers from cutting corners during both the design and implementation stages of a project. The role of these standards in my project is to ensure that every aspect of the drone, from its design to its operation, adheres to established safety and quality guidelines, thereby minimising the risk of failure and ensuring that the product meets or exceeds customer expectations[35].

Standards vs. Regulations: It's important to distinguish between standards and regulations. Standards are typically voluntary guidelines that establish best practices within an industry. For instance, IEEE 802.11 standards guide wireless communication protocols to ensure compatibility and performance across different devices. Regulations, on the other hand, are mandatory rules enforced by government bodies or industry regulators, such as the CAAS rules for drone operation, which ensure that drones are operated safely and within legal boundaries.

In the context of my project, relevant standards could include those related to **electrical safety (e.g., IEC 60950-1)**, which ensures that electronic equipment is safe to operate, and **EMI shielding standards (e.g., IEEE 299)**, which protect my drone's sensitive electronics, such as the gyroscope and GPS modules, from interference that could disrupt operations. Following these standards not only ensures compliance with industry norms but also enhances the reliability and safety of the drone, which is crucial for maintaining trust with users and stakeholders.

2. Advantages and Disadvantages of Adopting Quality Standards

Adopting quality standards in my project offers a range of advantages and disadvantages that must be carefully weighed, particularly in a business environment.

Advantages: The primary advantage of adhering to quality standards is the **assurance of consistent product quality**. By following established standards, my project ensures that each unit of drone is built to the same specifications, reducing variability and enhancing the product's reliability. This consistency builds **customer trust and satisfaction**, as clients can be confident that the drones will perform as expected, leading to higher customer retention and positive brand reputation. Additionally, quality standards often lead to **process improvements**, such as streamlined production methods that reduce waste and errors, ultimately saving costs and improving efficiency. For example, integrating EMI shielding standards into the drone's design could prevent signal interference, leading to more reliable operations and fewer malfunctions.

Disadvantages: However, adopting these standards also presents certain challenges. One of the most significant disadvantages is the **increased cost and time associated with compliance**. Implementing quality assurance protocols requires additional resources, including specialised materials (like EMI shielding), more rigorous testing procedures, and possibly slower production times. This can increase the cost of production, which may need to be passed on to the consumer, potentially affecting competitiveness in price-sensitive markets. Moreover, adhering strictly to standards can sometimes introduce **rigidity into the development process**, stifling innovation by focusing too much on compliance rather than exploring creative solutions.

Do the Advantages Outweigh the Disadvantages? In my opinion, the advantages of adopting quality standards far outweigh the disadvantages. The initial costs and potential delays associated with quality assurance are investments that pay off in the long run through improved product reliability, customer satisfaction, and a stronger market position. As a consumer, I value products that I can trust to work safely and effectively, even if they come at a higher price. This perspective underscores the importance of maintaining high standards in engineering practices, ensuring that the end product is both safe and of high quality.

3. Proposed Improvements

In the context of my project, specific areas for improvement could further enhance the quality and safety of the drone, ensuring it operates more reliably and efficiently.

Replacement of Arduino Uno with PX4 Flight Control Board: One of the most significant improvements would be replacing the Arduino Uno with a more advanced flight control board like the **PX4**. The PX4 flight controller comes equipped with integrated sensors, including a barometer, gyroscope, accelerometer, and magnetometer, all of which are essential for stable and accurate flight. Unlike the Arduino Uno, which requires manual coding to integrate and manage these sensors, the PX4's onboard systems automatically handle sensor fusion, significantly reducing the risk of coding errors that could lead to mid-flight issues. This would not only simplify the development process but also improve the drone's reliability and performance, particularly in maintaining stable flight and accurate altitude control.

Use of Fuses: Another critical improvement is the addition of more fuses in the drone’s electrical connections. Fuses provide essential protection for the drone's circuits by breaking the connection if the current exceeds safe levels, preventing overheating or fire hazards. This is especially important given the increased power demands that come with integrating more advanced components like the PX4 flight controller.

EMI Shielding: EMI shielding is another crucial area for enhancement. EMI can disrupt the performance of the drone's sensitive components, such as the PX4's sensors and GPS modules, leading to inaccurate readings or loss of control. Implementing effective EMI shielding, either through EMI filters or by strategically placing components to minimise interference, would significantly improve the drone's operational reliability, ensuring that it performs well even in environments with high levels of electromagnetic noise.

Integrated Sensor Advantages: The PX4 flight controller's integrated sensors eliminate the need for external components like separate barometers or gyroscopes, which were necessary with the Arduino Uno. This not only reduces the complexity of the drone's design but also minimises the risk of errors that could arise from manually coding sensor interactions. The PX4's built-in sensors provide more accurate and reliable data, which is crucial for precise altitude measurement and stable flight. This improvement would result in a more robust and reliable drone, capable of handling more complex flight manoeuvres and maintaining stability in a variety of conditions.

By replacing the Arduino Uno with a PX4 flight controller and implementing these additional improvements, it would significantly enhance the drone's performance, safety, and reliability. These upgrades ensure that the drone meets higher standards of quality, making it more suitable for real-world applications where precision and dependability are paramount.

University of Hertfordshire UH

School of Physics, Engineering and Computer Science

Additional Pointers

(Awareness of Time Management)

Effective time management was a critical component of this project. From the outset, establishing a realistic timeline was essential, especially given the complexity of integrating various hardware and software components. One key lesson learned was the importance of building in buffers to accommodate unforeseen setbacks, such as the switch from TinyYOLOv4 to MobileNetV2. This adjustment caused a two-week delay, underscoring the need for flexibility in project planning. Moving forward, more detailed planning and regular progress reviews will be implemented to ensure that the project stays on track. Additionally, time management tools like Gantt charts and project management software were instrumental in tracking milestones and adjusting the timeline as necessary, ensuring that all tasks were completed efficiently.

(Awareness of Costs)

Cost management was a significant factor throughout the project, guiding decisions at every stage. The choice to use the Arduino Uno and Raspberry Pi 3 was driven by the need to balance performance with affordability. Both of these components are cost-effective yet capable of handling the essential tasks required for the drone, such as flight control and real-time facial recognition. This decision allowed the project to stay within budget while still achieving the necessary functionality.

However, working within a limited budget also required careful planning to avoid unexpected expenses. For instance, sourcing affordable components without compromising quality was critical to ensuring the overall reliability of the drone. Additionally, the project had to consider the cost of peripherals and sensors, like the gyroscope and camera, which were necessary for the drone's operation. Throughout the project, cost-effective alternatives were evaluated, and only those that met the performance criteria were selected, ensuring that the project remained financially viable.

In future projects, a more detailed cost analysis could be performed at the planning stage, potentially identifying areas where higher initial investments might lead to long-term savings, such as by reducing maintenance costs or improving efficiency. Furthermore, the decision to use easily accessible and well-supported platforms like Arduino and Raspberry Pi also minimised costs related to troubleshooting and development, as extensive community support reduced the need for expensive proprietary solutions

(Awareness of Market Needs in Singapore)

Understanding market needs in Singapore was crucial in shaping the project's objectives and design. Singapore's push towards becoming a Smart Nation means there is a growing demand for advanced surveillance and monitoring technologies. The drone developed in this project aligns with these market needs by offering a cost-effective solution for crowd surveillance, which could be applied in public safety, event management, and urban planning. Furthermore, Singapore's compact urban landscape necessitates the development of drones that are efficient, reliable, and capable of operating in densely populated areas. The project's focus on real-time facial recognition and autonomous navigation directly addresses these needs, positioning the drone as a potential solution for enhancing public security and efficient city management in Singapore.

University of Hertfordshire

School of Physics, Engineering and Computer Science

<u>(Awareness of Sustainability)</u>

Sustainability was a key consideration in the development of this project, particularly in the context of Singapore's drive towards becoming a green and smart city. The choice of components and materials was guided by the principles of sustainability, opting for energy-efficient hardware like the Raspberry Pi 3 and components that have a lower environmental impact. Additionally, the drone’s design focused on maximising energy efficiency, which not only reduces operational costs but also aligns with Singapore’s broader environmental goals. The inclusion of potential future upgrades, such as solar-powered systems or more efficient batteries, was considered to further enhance the drone's sustainability. Furthermore, the modularity of the drone’s design allows for easy upgrades and repairs, reducing the need for complete replacements and minimising electronic waste. By aligning with Singapore’s sustainability initiatives, this project not only meets current market demands but also contributes to the nation’s long-term environmental goals.

References

[1] Xu, B., Li, B. and Wang, L., 2020. Iot System Based on Drones for Public Safety and Search and Rescue Missions. The Frontiers of Society, Science and Technology, 2(12)
[2] Elharrouss, O., Almaadeed, N. & Al-Maadeed, S., 2021. A review of video surveillance systems. *Department of Computer Science and Engineering, Qatar University, Doha, Qatar*. Received 13 May 2018, Revised 7 February 2021, Accepted 2 April 2021, Available online 16 April 2021, Version of Record 21 April 2021
[3] Adorni, E., Rozhok, A., Revetria, R. and Ivanov, M., 2021, October. Literature review on drones used in the surveillance field. In Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists (pp. 20-22)
[4] Suma, M., Premananda, R. and Harakannanavar, S.S., 2023. Development of Object Recognition Model Using Machine Learning Algorithms on MobileNet V2. Int. J. Advanced Networking and Applications, 15(02), pp.5908-5914.
[5] Chiu, Y.C., Tsai, C.Y., Ruan, M.D., Shen, G.Y. and Lee, T.T., 2020, August. Mobilenet-SSDv2: An improved object detection model for embedded systems. In 2020 International conference on system science and engineering (ICSSE) (pp. 1-5). IEEE.
[6] Rios, A.C., dos Reis, D.H., Da Silva, R.M., Cuadros, M.A.D.S.L. and Gamarra, D.F.T., 2021, August. Comparison of the yolov3 and ssd mobilenet v2 algorithms for identifying objects in images from an indoor robotics dataset. In 2021 14th IEEE International Conference on Industry Applications (INDUSCON) (pp. 96-101). IEEE.
[7] Raj, T. and Hashim, F.H., 2016, November. Development of an autonomous flight controller circuit with real time data transmission. In 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES) (pp. 115-119). IEEE.
[8] Semsch, E., Jakob, M., Pavlicek, D. and Pechoucek, M., 2009, September. Autonomous UAV surveillance in complex urban environments. In 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (Vol. 2, pp. 82-85). IEEE.
[9] hakur, N., Nagrath, P., Jain, R., Saini, D., Sharma, N. and Hemanth, D.J., 2021. Artificial intelligence techniques in smart cities surveillance using UAVs: A survey. Machine Intelligence and Data Analytics for Sustainable Future Smart Cities, pp.329-353.
[10] Huang, H. and Savkin, A.V., 2021. Navigating UAVs for optimal monitoring of groups of moving pedestrians or vehicles. IEEE Transactions on Vehicular Technology, 70(4), pp.3891-3896
[11] Alhafnawi, M., Salameh, H.A.B., Masadeh, A.E., Al-Obiedollah, H., Ayyash, M., El-Khazali, R. and Elgala, H., 2023. A survey of indoor and outdoor uav-based target tracking systems: Current status, challenges, technologies, and future directions. IEEE Access, 11, pp.68324-68339.
[12] Wang, X., 2016. Deep learning in object recognition, detection, and segmentation. Foundations and Trends in Signal Processing, 8(4), pp.217-382.
[13] Galvez, R.L., Bandala, A.A., Dadios, E.P., Vicerra, R.R.P. and Maningo, J.M.Z., 2018, October. Object detection using convolutional neural networks. In TENCON 2018-2018 IEEE region 10 conference (pp. 2023-2027). IEEE.
[14] Hu, L. and Ge, Q., 2020, June. Automatic facial expression recognition based on MobileNetV2 in Real-time. In Journal of Physics: Conference Series (Vol. 1549, No. 2, p. 022136). IOP Publishing.
[15] Foley, D. and Oreilly, R., 2018. An Evaluation of Convolutional Neural Network Models for Object Detection in Images on Low-End Devices. AICS, 2259, pp.1-12.
[16] Sukkar, V. and Ercelebi, E., 2023. A Real-time Face Recognition Based on MobileNetV2 Model. Balkan Journal of Electrical and Computer Engineering (BAJECE), 1(1).

[17] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.
[18] Kulkarni, U., Meena, S.M., Gurlahosur, S.V. and Bhogar, G., 2021. Quantization friendly mobilenet (qf-mobilenet) architecture for vision based applications on embedded platforms. Neural Networks, 136, pp.28-39.
[19] Ucgun, H., Okten, I., Yuzgec, U. and Kesler, M., 2022. Test platform and graphical user interface design for vertical take-off and landing drones. Romanian Journal of Information Science and Technology, 25(3), pp.350-367.
[20] Noordin, A., Basri, M.A.M. and Mohamed, Z., 2018. Sensor fusion for attitude estimation and PID control of quadrotor UAV. International Journal of Electrical and Electronic Engineering and Telecommunications, 7(4), pp.183-189.
[21] Finn, R.L. and Wright, D., 2012. Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications. Computer Law & Security Review, 28(2), pp.184-194.
[22] Wang, N., Christen, M. and Hunt, M., 2021. Ethical considerations associated with “humanitarian drones”: a scoping literature review. Science and engineering ethics, 27(4), p.51.
[23] Gulzar, Y., 2023. Fruit image classification model based on MobileNetV2 with deep transfer learning technique. Sustainability, 15(3), p.1906.
[24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
[25] King, D.E., 2009. Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10, pp.1755-1758.
[26] He, K., Zhang, X., Ren, S., and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)
[27] Ahonen, T., Hadid, A. and Pietikainen, M., 2006. Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence, 28(12), pp.2037-2041.
[28] Howard, A.G., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.
[29] Redmon, J., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
[30] Bochkovskiy, A., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[31] Tan, M., Pang, R. and Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).
[32] Chen, M.Y., Edwards, D.H., Boehmer, E.L., Eller, N.M., Slack, J.T., Speck, C.R., Brown, S.M., Williams, H.G., Wilson, S.H., Gillum, C.S. and Lewin, G.C., 2013, April. Designing a spatially aware and autonomous quadcopter. In 2013 IEEE Systems and Information Engineering Design Symposium (pp. 213-218). IEEE.
[33] Kishor, V. and Singh, M.S., 2017. Design and Development of Arduino Uno based Quadcopter. International Journal of Engineering and Manufacturing Science, 7, pp.1-9.
[34]Ebeid, E., Skriver, M. and Jin, J., 2017, August. A survey on open-source flight control platforms of unmanned aerial vehicle. In 2017 euromicro conference on digital system design (dsd) (pp. 396-402). IEEE.
[35] Winfield, A., 2019. Ethical standards in robotics and AI. Nature Electronics, 2(2), pp.46-48. (Quality Standards)