Project Initial Concept and Design and Presentation

Topic: DroneAI: Humans interacting with Drones

Course Code: FIT3161

Team: MCS14

Submitted by:

Rahul P. Rajendran (32912617)

Lim Zheng Haur (32023952)

Yap Jit Feng (32898339)

Thehara Nikhila Goonewardena (32312512)

Introduction

In the rapidly evolving technological ecosystem, the convergence of drone technology and artificial intelligence (AI) has the potential to revolutionise multiple industries. DroneAI, our project, aims to enhance current techniques in video analysis tailored to drone-captured images and videos. This will enable drones to interact more intuitively with humans, mimicking harmonious interactions like those of dragonflies in shared spaces.

Representation (2)

Representation 1

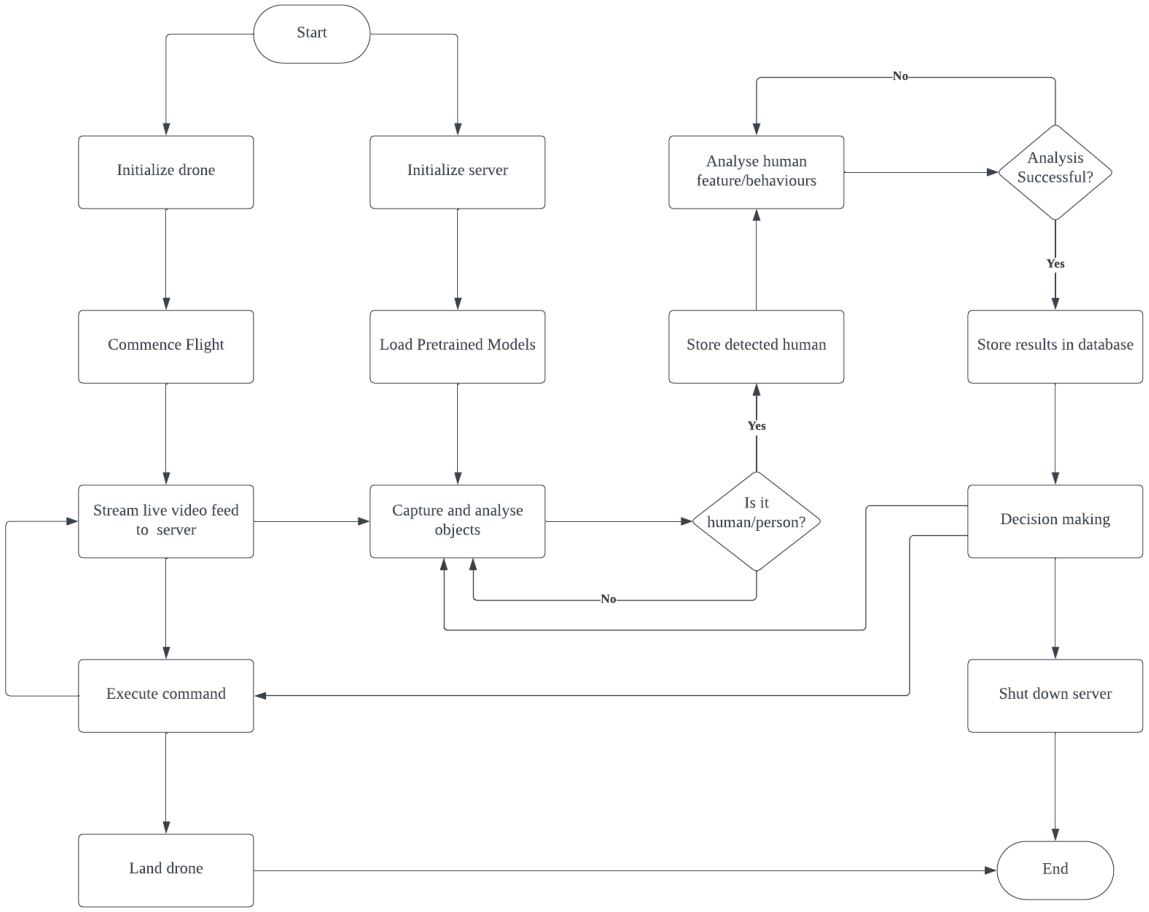

Figure 1: Flow diagram of the drone AI process

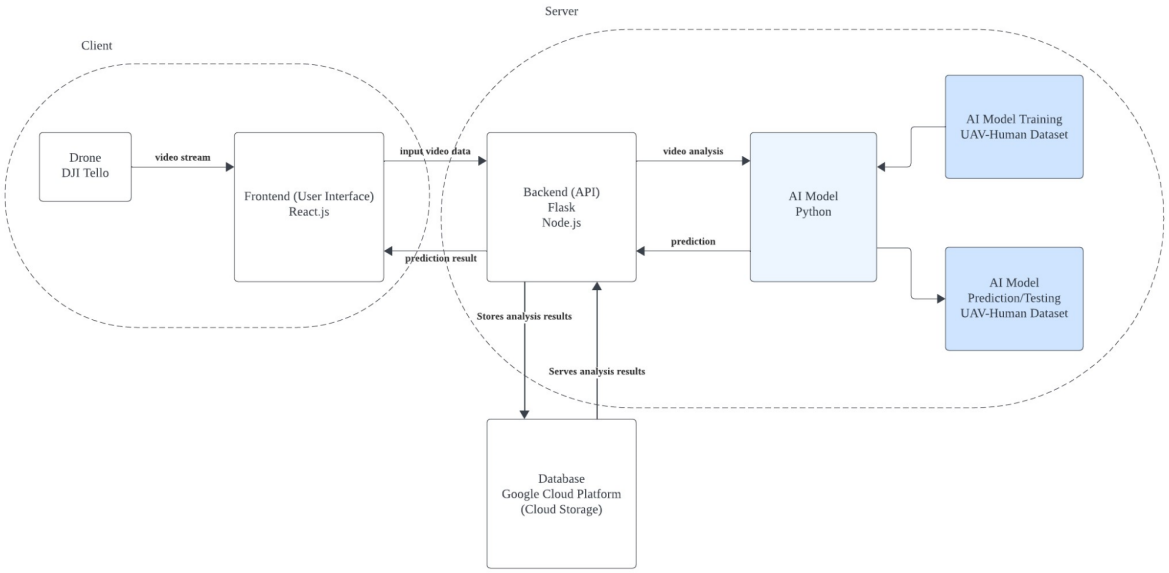

Figure 2: Block diagram of the drone AI process

The block diagram above highlights the architectural software of our project. The client or user side of the web app involves the drone supplying the live video feed to be displayed to the user as well as displaying the action recognition prediction by the AI model back to the user. The web-based frontend framework that will be used is React.js.

On the other hand, the server or backend side of our project involves receiving the input data and analysing the data using our AI model that has been trained and tested on the UAV-Human dataset. The model then provides a prediction of the action and displays that prediction back to the user and stores the prediction in a cloud database. The backend frameworks include Python Flask and Node.js and Google Cloud Storage to store the unstructured data, in the form of video files, and its corresponding analysis results.

Representation 2

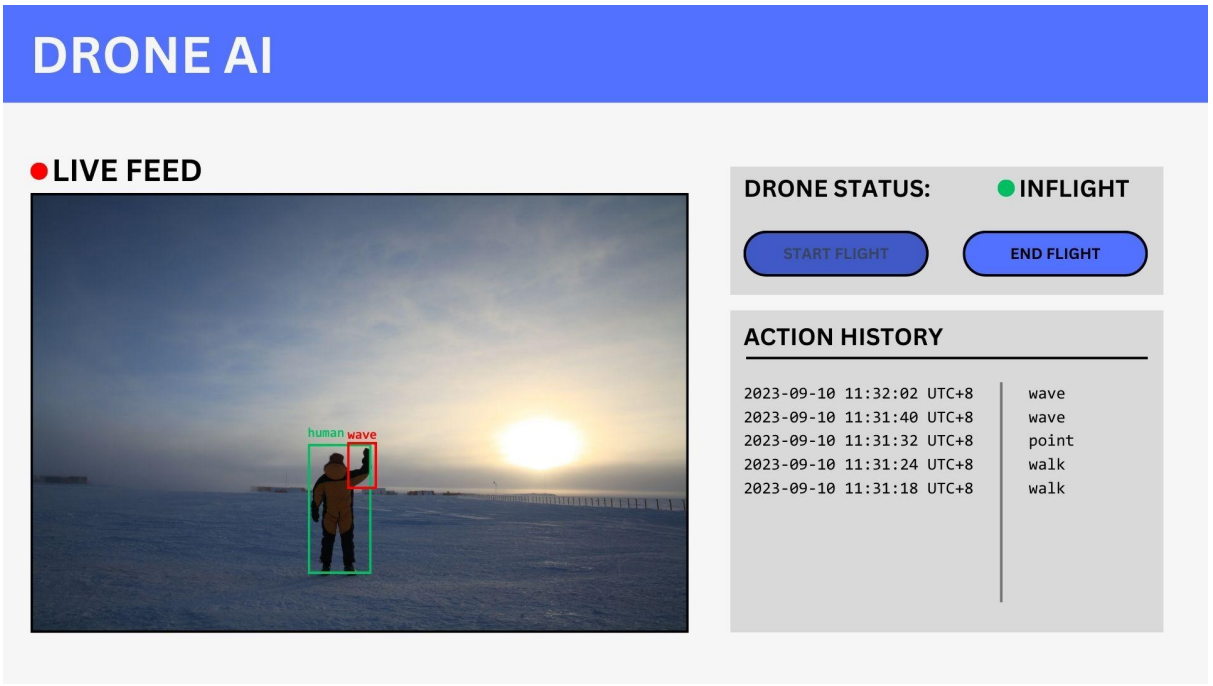

Figure 3: Web user interface mockup

The representation is a web user interface mockup diagram. In the end of our project, we would require a user interface for us to control the video processing of the drone footage. Figure 3 represents our vision of how our end project would look like, for there to be controls to start and end the process, a display for the live drone footage with the drone artificial intelligence results overlay above, and a log of all the actions and behaviours processed by the artificial intelligence. This representation is an initial mockup of our final project, but we expect changes and improvements to arise, and for the final product to be substantially different from our initial mockup. Further features that are in the scope of this project include facial features, emotion recognition which would be included in the video overlay as well as the action logs. It is important to note that the controls of the drone (e.g. flight controls) will be done separately on a Radio Control (RC) Transmitter rather than the web user interface which is solely to control the video processing of the drone footage.

Software Specifications

Software Components or Blocks:

- User Interface (UI): Simple, clean and easy to use for end users. Includes features like real-time video stream, AI analysis results, system controls, etc.
- Backend: Manages AI algorithms, data processing, and communicates with the database and UI.
- Database: Stores video data, AI analysis results and other relevant information.
- Artificial Intelligence (AI) Models: Existing models such as Convolutional Neural Networks (CNN) OR pre-trained models such as Residual Neural Network (ResNet) will be used to accurately categories/predict given video/image data.

Software Relationships and Data Flow:

- UI ↔ Backend: User requests and system commands, AI results display.
- Backend ↔ Database: Data retrieval, storage, and updates.
- Backend: Data processing, where raw video is analysed by AI algorithms.

Software Quality:

- Accuracy: Ensure AI algorithms correctly interpret human behaviours and features.
- Reliability: Continuous and stable operation without unexpected crashes or errors.
- Performance: Algorithms should perform its tasks efficiently and within an acceptable time frame.

Development Platform & Tools:

- Programming Language: Python, a preferred choice given its extensive AI and ML library ecosystem.
- Database System: Google Cloud Storage, designed to store large binary objects such as video files
- IDE: PyCharm or VSCode, offering an optimised environment for Python development.
- AI Library: TensorFlow, an industry frontrunner for AI and ML applications. PyTorch, a deep learning framework known for its flexibility in computing dynamic graphs. OpenCV used for pre-processing images and video data collected by the drone.

- Project Management Tools: Git for collaborative and organised development. Trello to organise project tasks and keep track of progress.
- File Sharing Platforms: Google Drive for sharing relevant files and research findings.
- Communication Platforms: Whatsapp and Discord for basic communication between each group member. Zoom for online face to face meetings with supervisors.

The different software components can be broken down into the user interface (UI), backend, database and AI/ML model to be used. The primary interface of the DroneAI system that end users would interact with will be relatively simple as end users would primarily be focused on the results of the data analysis. As such, ease of use and understanding takes priority for the UI. The backend involves the algorithms that preprocess data, carry out analysis and provide the results to the UI. The database on the other hand stores the video files that are used to train and test the AI model as well as any video files used in real-life testing.

As for our development platform and tools, Python will be our main language used as it hosts a vast ecosystem of libraries and frameworks which are crucial components for our DroneAI project. Moreover, most drone manufacturers do offer Python APIs or SDKs that allow ease of control over the drone and more importantly retrieving video streams from the drone resulting in simpler communication with the drone hardware.

Regarding the AI libraries to be used, OpenCV (Open Source Computer Vision Library) excels in video processing tasks such as frame manipulation and video stream handling and is compatible with various file formats and camera sources. On the other hand, TensorFlow is used because it not only provides a powerful deep learning framework used to train deep neural networks but also contains a vast collection of pre-trained deep learning models for action recognition. Finally, PyTorch is also another relatively new deep learning framework that can act as a suitable replacement for TensorFlow with similar functionalities but limited visualisation and debugging capabilities when compared to TensorFlow.

The database that we plan to use is Google Cloud Storage which is a cloud-based storage that is able to store and serve unstructured data making it the perfect choice for video file storage. For instance, if a relational database, such as PostgreSQL is used, we would not be able to store video files directly, instead only the metadata of the file can be stored, thus with cloud storage we are able to store and serve with minimal metadata-related queries making it more scalable and cost-effective. Furthermore, Google Cloud Storage is fully compatible with Python and even provides a Python client library which ensures ease of communication between the backend and database.

As for our other tools, Git will be used to handle the collaborative development of our project between team members as it allows multiple members to work on the project at the same time and able to revert any changes to the project if mistakes were made. Trello is used to organise our tasks in the form of boards which are intuitive and track progress to ensure we do not exceed the timeframe of our project.

Hardware Specifications

Drone Hardware:

- Camera: High-resolution RGB-IR camera.
- Sensors: Infrared, Lidar, and ultrasonic sensors for comprehensive environment perception.
- Battery: sufficient power needed for the hardwares and AI computation power.

Data Storage Requirement:

- Local Storage: High-speed SSDs with a capacity of 1TB for temporary storage and faster read/write operations.
- Cloud Storage: AWS S3 Buckets for long-term storage and archiving.

Networking Requirement:

- Onboard Communication: 5G modules for real-time data transmission.
- Ground Station: Robust Wi-Fi routers with extended range capabilities for uninterrupted data reception and command transmission.

Computing Requirement:

- On-Drone
    - Powerful on-board GPU for real-time AI computations.
    - Sufficient random access memory (RAM) for data processing and storage
- Ground Station: High-performance multicore CPU coupled with a dedicated GPU for data processing and AI model training sessions.

The main hardware component for DroneAI is the drone itself as the entire project revolves around it. The drone would need components such as a camera, sensors and large batteries. The camera should be a high-resolution camera that can record its surroundings up to 1080p. High resolution footage captured from the camera allows more accurate analysis for the learning model. Sensors with ultrasonic sensors need to be included to record the drone's environmental surroundings. Large battery is needed for the drone so that we can work with the drone for a longer period of time.

The data collected from the camera will be stored in a solid state drive (SSD) with memory capacity of 1TB. This is because the SSD is a lot faster than the hard disk drive. Since the

camera is recording video footage in 1080p, the video footage will take a larger amount of space in the SSD. Thus we will have at least a 1TB worth of SSD. The data collected in the drone will then be transferred into a cloud storage so that there will be backup for the data and the storage space in the cloud is limitless.

A Low-Latency network is needed for data transmission from the drone to the ground control stations (GCS). This allows fast communication between the drone and ground control station to analyse the video footage and perform appropriate decision making for the drone. GCS would need a wifi router with extended range capabilities to allow the drone to go further away to collect more data.

AI models will need a sufficient amount of power to analyse human behaviour efficiently. A graphic processor unit (GPU) will be required to enhance the performance and efficiency of the machine learning algorithms. This is because GPUs are built to perform parallel tasks, meaning that it is able to split large amounts of tasks into different processors simultaneously. Random access memory (RAM) will be needed as it acts as a temporary storage for collected dataset. Thus, high capacity of RAM will be needed to compute large-scale datasets.

Justification of Choices

Different amounts of Machine Learning Models were studied to determine the best algorithm to use in our problem. These machine learning models are object identification models that assist us in differentiating different objects in a given video frame. We have analysed three different classification models: You Only Look Once, Faster R-CNN and Single Shot Multi Detector.

Model	You Only Look Once (YOLO)	Faster R-CNN	Single Shot Multi Detector (SSD)
Description	YOLO is a real time detection machine learning algorithm that treats object detection as a regression task. This means that it uses multiple data to create a single prediction.	Faster R-CNN is an improved version of the R-CNN model. It combines regional proposal networks and the original CNN model together to develop an efficient model.	SSD is a real time detection model that uses cnn model to determine the layout of the image. It then uses anchor boxes which are predefined sizes of boxes to create a prediction.
Architecture	Divides images into grids and predicts class probability for each grid (Jiang et al, 2022).	Regional proposal extraction. Uses Selective search to determine the object region. Uses cnn model to classify the region object.(Jiang & Learned-Miller, 2017).	Similar to CNN, but it uses the CNN model to determine the layout of the box with different resolutions (Liu et al, 2022). Then it uses an anchor box to generate different predictions.
Speed	Fastest	Slowest	Middle of the pact
Data Type	Image Data, Bounding Box Coordinates, Class Labels	Image Data and Class Labels	Image Data, Bounding Box Coordinates, Class Labels
Accuracy	Accuracy lower than Faster R-CNN and SSD	Best accuracy. Needed large amount of data to compute	Good Accuracy and increasing dataset size continues to increase accuracy
Object Size	Does not work well with different sizes object but recent update made it better	Not effective in dealing different sizes objects	Effective in classifying different sizes objects
Detection Type	Real Time Detection	Detect objects in an image	Real Time Detection

Below, we have analysed three other pre-trained machine learning models that we are considering as our action recognition model.

Model	**Inflated 3D CNN (I3D)**	**ResNet (2+1)D (R(2+1)D)**	**Temporal Shift Module (TSM)**
Description	I3D is a spatial temporal architecture that extends the capabilities of 2D deep neural networks.	R(2+1)D is a 3D convolutional neural network for action recognition.	TSM is a module that models temporal information efficiently.
Architecture	Combines the output of two 3D CNNs, one processing a group of RGB frames and the other processing a group of optical flow predictions among consecutive RGB frames (Carreira and Zisserman, 2017).	Employs R(2+1)D convolutions in a ResNet inspired architecture. (Tran et al., 2017). Two stream architecture, processing spatial and temporal information separately.	TSM performs efficient temporal modelling by moving the feature map along the temporal dimension. It is computationally free on top of a 2D convolution, but achieves strong temporal modelling ability (Ji Lin et al., 2019)
Computational Complexity	Very intensive for real time recognition as 3D convolutions are employed	Efficient due to reduced computational complexity.	Efficient as it achieves the performance of 3D CNN but maintains 2D CNN's complexity
Data Type	RGB video frames and optical flow predictions.	RGB video frames	RGB video frames
Accuracy	High accuracy on datasets like Kinetics.	High accuracy on datasets like Sports-1M, Kinetics, UCF101, and HMDB51	High accuracy on datasets like COCO, Kinetics, UCF 101 and Something-Something.
Detection Type	Real Time or Near Real Time Detection	Real Time or Near Real Time Detection	Real Time or Near Real Time Detection

References

Carreira, J., & Zisserman, A. (2017). Quo Vadis, action recognition? A new model and the kinetics dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.502

Jiang, H., & Learned-Miller, E. (2017). Face detection with the faster R-CNN. *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. https://doi.org/10.1109/fg.2017.82

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of Yolo algorithm developments. *Procedia Computer Science*, *199*, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

Lin, J., Gan, C., & Han, S. (2019). TSM: Temporal shift module for efficient video understanding. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2019.00718

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Computer Vision – ECCV 2016*, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2018.00675