Journal of Sensors, IoT & Health Sciences
ISSN: 2584-2560
Vol.xx(xx), June xxxx, pp.x-xx

Type of Article (Article)

An Integrated Wireless Video Robotic Aerial System for Emergency Real-Time Monitoring

Christopher Ingham[1,2,3], David Reid[1] and Emanuele Lindo Secco[3,*]

[1]AI Lab, School of Computer Science and the Environment, Liverpool Hope University, L16 9JD, UK
[2]CGI, Liverpool, Suite 5, 23 Mann Island, Liverpool, L3 1BP, UK
[3]Robotics Lab, School of Computer Science and the Environment, Liverpool Hope University, L16 9JD, UK
[*]Emanuele Lindo Secco. Email: seccoe@hope.ac.uk
Received: XX April 2025; Accepted: XX Month XXXX

Abstract: Emergency scenarios are becoming more and more demanding for the National Health Services and for emergency operators such as Police and Fire-Fighters. In this context the rapidity and efficiency of the interventions are mandatory skills. We leverage on current technologies to propose an integrated system where conversational interaction with a Machine Vision System can provide a specific behavior to a mobile vehicle combined with a flying drone in order to execute a specific task such as, for example, having the drone following an operator while intervening in the emergency (i.e. earthquake, fire, flooding, etc.). The paper investigates the use of video object detection using a DJI Mavic Air Drone for real-time applications. The object detection system is provided by CGI Machine Vision, detecting objects and people from live footage. The aim of the proposed architecture is to provide an integrated solution for streaming a live video feed from a camera drone to an edge device running CGI Machine Vision to prove the concept of vehicle-based drones to provide aerial situational awareness to emergency operators and law enforcement officers. The paper presents the design, implementation and testing of the system, as well as the successful proof of concept and real-world demonstration. The research gave positive results and demonstrated the capability, along with recommendations for further refinement and next steps.

Keywords: Emergency Interventions, Aerial Monitoring, Human-Machine Interaction, Health Technology Innovation

1 Introduction

The proliferation of *Unmanned Aerial Vehicles* (UAVs) across many industries and sectors is changing the way industries operate, facilitating new and innovative technological advancements to replace existing use cases and carve out new opportunities for industry. One such example explored in this work is the use of UAVs for video object detection specifically for real time applications. UAVs can be combined with many different technologies, including autonomous flight management software, AI models for a variety of outcomes, and integration with other existing technologies and software, providing a huge and varied range of industrial use cases. This paper explores how UAVs can be integrated with AI Vehicle Object Detection models specifically

for real time applications, giving real time outcomes. The specific use case focused on is the use of UAVs to support emergency operators and law enforcement with real time aerial operational support. Despite the increasing use of UAVs in surveillance and operational support, real time applications prove to be challenging due to the complexity of communication, data processing, connectivity in remote areas, integration complexity and data security. Current solutions suffer from high latency, difficulties in detecting objects or people accurately in real world environments, communication, connectivity and security constraints. This work aims to address these challenges by exploring the integration of UAVs with Video Object Detection for real time applications.

The Video Object Detection implementation researched in this work makes use of computer vision framework named CGI Machine Vision. CGI Machine Vision is a versatile and AI model agnostic computer vision framework which can be used across a variety of sensors (i.e. vision camera, FLIR, radar), along with a variety of different open source or custom computer vision models, using either edge processing, cloud processing, or a combination of the two.

The particular research questions explored in this work are as follows: how effective is the use of Video Object Detection with Robotic Aerial Vehicles in an Emergency Scenario? Can such a system provide real time applications with low latency? What operational benefits can be enabled by the use of this technology? How can the solution be kept secure? What does this research tell us about how the solution may need to be adapted for a real-world rollout?

This paper aims to approach these questions through the development of working proof of concept of an integrated system, running video object detection on a live video feed from a robotic aerial vehicle in real time.

2 Foundational Technologies, Context and Related Work

This Paragraph explores the foundational technologies that form part of this research, as well as additional context relating to the specific use case being demonstrated.

2.1 Drones and UAVs

Drones and UAVs cover a broad spectrum from recreational hobby aircraft and drones to full size aircraft that are remotely or autonomously piloted. The emergence of camera drones has further developed the use case for surveillance and aerial situation awareness across industries, ranging from military applications to disaster recovery and law enforcement. Several hardware providers have created enterprise level drones to suit these use cases, with extensive automation [1]. A number of open-source options for software and hardware exist to allow for the creation of custom solutions for secure applications. Examples include PX4 [2,3] and ArduPilot [4], two open-source flight control software tools used in industry to create custom solutions, along with hardware such as the Pixhawk open-source flight controller platform [2,3].

CGI Machine Vision

This research is supported by the company CGI and Liverpool Hope University, and the author is an employee of CGI working in their Emerging Technology Practice focusing on exploring the use of new technology in real applications. One such technology that has been developed by CGI is CGI Machine Vision. This software has been selected as the Video Object Detection framework for this project. CGI Machine Vision is a vision-oriented framework for a wide range of applications and technologies, making use of an integrated novel video generative AI model. The framework allows for a vast range of sensor inputs including visual cameras, infra-

red cameras, radar, LIDAR, sonar and other sources, in combination with an object detection model coupled with a generative AI component which allows users or systems to provide a natural language object detection prompt [5].

CGI Machine Vision has been successfully run on edge devices like the Nvidia Jetson board (Figure 1), and can be deployed as the full image model, or a lighter weight refined model for a particular use case. The object detection model can be packaged on the edge device as a full image model or as a scaled back model for one particular use case reducing consumption. The edge device would connect to cloud infrastructure to provide events and alerts from the software rather than the live video feed, reducing data transfer and communication needs. This is particularly useful in applications that are remote. The edge device can operate with a very low power demand, operating off a small solar panel and battery in the UK climate.

Figure 1: CGI Machine Vision Edge Device

2.2 Related Work

The paper “Hardware and Software Integration of Machine Learning Vision System Based on NVIDIA Jetson Nano” [6] offers an insight into the application of convolutional neural network models on an edge device, the Nvidia Jetson Nano board. The paper demonstrates the deployment of nine pre-trained computer vision models on a Jetson Nano edge device, enabling local processing of image/video data with low latency, and the broadcast of the live video stream to a local device using G-Streamer, posing significant similarities to this work. Nvidia Jetson boards offer a low cost, affordable, modular and portable platform for use in industrial and IoT devices, closely mirroring the industrial applications of CGI Machine Vision. One of the use cases of Machine Vision is for collecting images and data from IoT edge devices for a number of applications, which is initially implemented and explored in this paper. In [6] the authors experienced issues low memory limitations of the Jetson Nano, limiting frame rate on larger models. This may be something to consider in the implementation of this research project. The models in this research will be running on a developer specification laptop with 32GB of ram

rather than a low-cost edge device, mitigating this issue. That work has limited detail on security and data encryption for deployment in sensitive environments which this research aims to explore in detail. On the contrary the present work of this paper aims to build further on the learnings of [6], exploring a real-world use case with ethical and societal impact.

Another paper entitled "Dangerous Objects Detection Using Deep Learning and First Responder Drone" [7] provides an in depth exploration of using Deep Learning models running on footage obtained from a drone, to detect dangerous objects using vision and Infra-red cameras. The journal article details the method of running object detection on video obtained from a drone which is passed through a Deep Learning model to identify weapons such as knives, pistols and assault rifles with varying degrees of success. The implementation described in [7] utilised open-source vision models, including YOLOv5, for object detection. Video was recorded on the drone , however this was sent for offline processing, allowing the model to be run on a high specification PC with a high-performance graphics card for optimal results, rather than in real time on an edge device. The YOLOv5 model achieved an mAP50 score of 0.964 and 0.949 for colour and infra-red videos respectively, demonstrating a high confidence in the model's ability to detect dangerous objects. The proposed system of our work aims to further expand on this with an emphasis on real-time results, integrating a drone with edge-based image processing to provide results and situational awareness in real time to officers. One important note in [7] detailed is the issue of the lack of availability of source data which include aerial photographs from drones in a law enforcement scenario to train an image model. "*This suggests that more drone-based datasets are required to make the model more robust, and a customized network, utilizing small-scale networks, can be more suitable for the firearm detection problem*" [7]. It is foreseen that a similar issue may be encountered in the implementation of this work based on the nature of image datasets used to train the data model often being primarily sourced from photographs taken by people rather than from an aerial viewpoint.

Finally, the work of Lo et al. [8] offers an interesting insight into the implementation of an autonomous UAV system to detect and dynamically track objects. The implementation included the use of a small-scale quadcopter UAV with a Intel RealSense D435 stereo camera and Nvidia Jetson TX2 computer onboard the UAV for real time edge processing of frames. The UAV was controlled with a Pixhawk 4 flight controller supported by *Robot Operation System* (ROS). This article primarily focused on the training and deployment of a custom object detection model operating directly on the drone. The implementation within [8] focused primarily on detecting three specific example objects and the tracking of these objects, rather than a wide range of possible objects required for real-time surveillance. The article details in depth the method of training the model, defining the bounding box of the object to be tracked, and the layers of filtering to improve the results, rather than a particular real-world application of the technology. Although the article refers to the use of real time feedback to the drone to provide path planning and autonomous object tracking, the article does not detail this element in depth and suggests that further work was required to implement this. "*In addition, it is also considered that more work on the path planning module could be extended, in which the trajectory should be optimized based upon the following: target motion prediction, dynamic and static obstacles constraints, as well as UAV robot physical limitations*" [8]. This paper aims to further build on the technology discussed in the article, providing a real-time element for supporting emergency and law enforcement with situational awareness as an implementation of operational technology.

3 Materials and Methods

The technical implementation and demonstration of this work faced several limitations and constraints, leading to a continuous process of iterative design and refinement across multiple stages, as outlined in this section. Extensive troubleshooting, problem solving and design revisions were required to successfully implement the proof of concept demonstrating the use of video object detection using a robotic aerial vehicle for a real time application.

3.1 Conceptual Design

The conceptual design illustrates the use the potential use case of the system being demonstrated, and the technology included in the proof of concept. The design addresses the use case of supporting a law enforcement officer with aerial support (Figure 2). The diagram illustrates the items in scope to this proof of concept shown on the right side of the purple curve, and the potential future integration with existing emergency and police technology on the left-hand side of the purple curve.

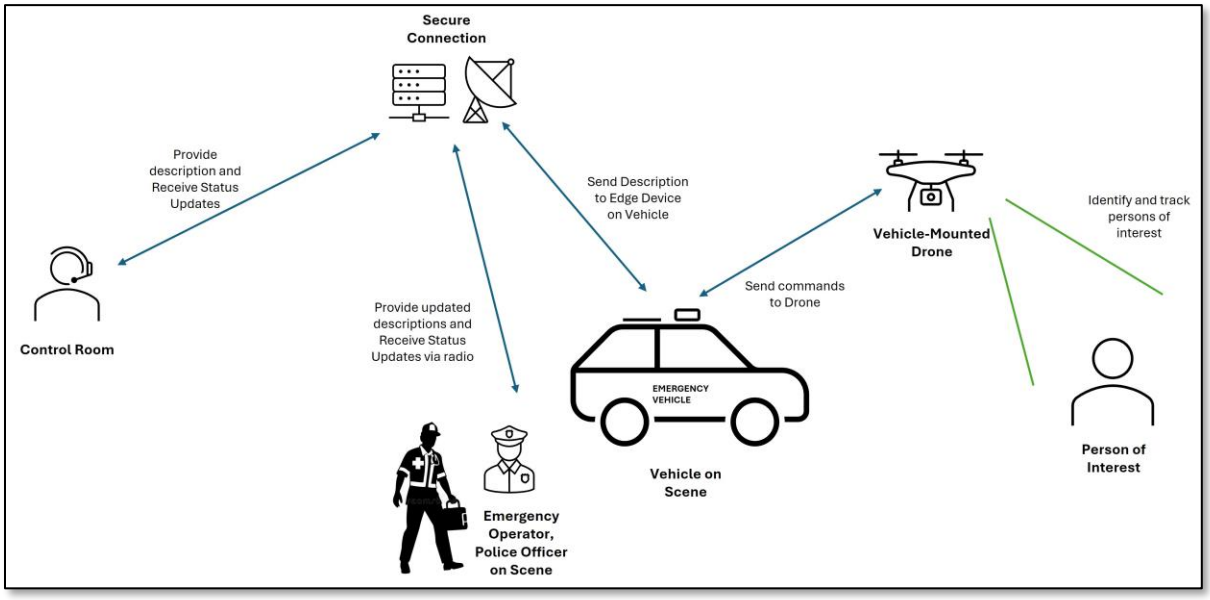


Figure 2: Conceptual Diagram of the Use Case

3.2 Hardware Selection

Due to the connection of this research with the company CGI, CGI Machine Vision was chosen as the vision model to be used in this implementation. This selection dictated a number of hardware choices for the system, with hardware selection primarily focusing on the drone.

Edge Device Hardware - CGI Machine Vision can operate on devices with a Nvidia GPU such as a Nvidia Jetson, or a desktop or laptop with a dedicated Nvidia GPU. For this implementation a laptop was selected to run Machine Vision, namely a Dell Latitude 5540 with a GPU Nvidia GeForce MX550.

Drone Hardware - DJI a popular manufacturer of recreational and professional drones, with a wide range of low-cost drones offering excellent features and intuitive iPhone and Android applications, as well as Windows and Mobile SDKs. To simplify the list of hardware, drones manufactured by DJI were initially selected due to familiarity with the platform. The first drone assessed was the DJI Mavic Mini 2. This drone features a light weight of 246 g, a high-definition gimbal camera, a gyroscope and several sensors for improved stability. The DJI Mavic Mini 2 has limited autonomous flight capabilities and collision avoidance, but it would be used later in the project for recording some cinematic aerial footage for the demonstration. The second drone assessed was the DJI Mavic Air. The Mavic Air is a professional platform, offering 4K video resolution and an increased weight of 430 g. The drone includes front and rear object avoidance and advanced object tracking and navigation modes, allowing the drone to autonomously track an object whilst manoeuvring around objects. The drone includes an "Active Track" feature, allowing a user to provide a bounding box of an object or person on the video feed with the drone autonomously tracking the object for an extended period. The drone also included direct wireless connectivity and was supported by a Windows SDK. This drone was chosen for the implementation of the proof of concept (Figure 3).

Figure 3: The DJI Mavic Air Drone used for the project.

DJI provide several Software Development Kits for Windows, Android and iOS. These kits provide pre-built classes and methods to use common drone components within custom applications. DJI project several versions of the SDKs along with SDKs for each operating system, with varying degrees of content and support for different hardware. Only a small number of the most recently released drones are supported by the latest SDK V5. The SDK API reference documentation can be found on the DJI website [9,10], along with the links to the GitHub site with code [11].

3.3 Design

The following paragraph detail the iterative design process followed during the development of this proof of concept, and the benefits and challenges of each version.

Version 1 - The first version of the implementation focused on the use of the DJI Windows SDK to connect directly to the drone. DJI provide an Android, iPhone and Windows SDK for their drones, along with sample applications allowing developers to test the connectivity of their drone and basic commands. However, when the API key was entered an error was presented. This error was shared with the DJI development support team who advised that the DJI Windows SDK was no longer supported.

Version 2 - Following the issue identified with the DJI Windows SDK, further information was gathered on the support of the DJI SDKs. The design was updated to reflect the move to the use of an iPhone connected to the Drone Controller via USB cable. A custom iPhone app was to be created, which incorporated the DJI Mobile SDK for receiving the video feed, and providing coordinates back to the drone to track an object. Additionally, a piece of Python middleware was needed on the windows laptop to take the JSON bounding box coordinates from CGI Machine Vision and send these values to an API server running within the iPhone mobile app, which could in turn be passed to the DJI SDK, instructing the drone to autonomously track the person. Unfortunately, on reading the readme for this piece of code it was found that although the DJI Mavic Air drone supports Active Track via the DJI end user mobile app, the DJI SDK did not include the DJI Mavic Air as a supported device by the Active Track API. Furthermore, the DJI development team also highlighted that the only SDK which was currently supported was the Android SDK.

Version 3 - The third version of the design considered a more simplistic implementation of passing the live video feed to the Windows laptop, making use third party software to simplify the solution, making use of third-party software with no code required. Version 3 made use of the working DJI SDK FPV view that was demonstrated in Version 2. The in-built iPhone Apple screen mirroring component would be used to display the iPhone's screen on the windows laptop using an app called *APowerMirror*. This approach presented a number of issues. The primary issue with this implementation was the requirement for the video to pass through multiple third party applications introducing significant latency.

Version 4 – As identified in *Version 2*, the Android SDK included a pre-built class (*LiveStreamManager*) and set of methods for sending the drone's live video feed to an RTMP server. First the method *setLiveURL* would be called to set the URL or IP address of the RTMP server, followed by the *startStream* method to initiate the video stream. The hardware selection amended from an iPhone to and Android phone to allow the use of the DJI Android MSDK and an Android device was procured. The android device selected was a Samsung Galaxy A20E. This low-cost device offered 4GB RAM which could be used for handling and onward processing of the live video packets, and a sufficiently powered CPU for the tasks required. Initial testing proved successful, with the android phone successfully connecting to the drone, showing the live video feed of the drone on screen. A RTMP server application was installed on the Windows PC to test that the android application could connect to the RTMP server and send the video stream to the device. The android phone was unable to establish a connection with the CGI managed Windows Developer Laptop. Another personal desktop PC was used to test the connection running the same RTMP server application. Due to these findings and its flexibility [12, 13], a final solution was decided which would use a Raspberry Pi 4B with 4GB of memory to sit between the Android

phone and the CGI Developer Laptop to transcode the RTMP stream to RTSP. Figure 4 (left panel) shows the overall final solution adopted within version 4.

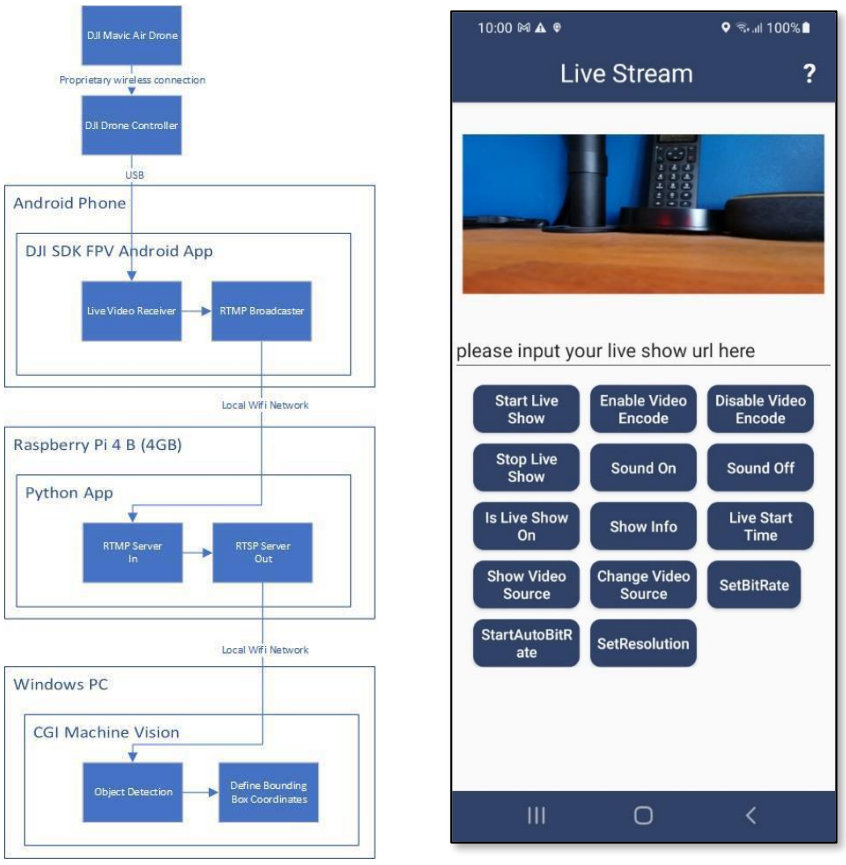


Figure 4: The Version 4 Design Diagram (left panel) and end-user interface (right panel)

This final version of the design made use of the DJI Mobile SDK running on an Android device. This device would connect to an application running on the Raspberry Pi. The application *MediaMTX* would provide an RTMP inbound server on the Raspberry Pi 4B to receive the video stream from the android phone and in turn provide this feed to an RTSP server to create the output video stream across the local network. *MediaMTX* is configured using a YAML file to set the application to receive RTMP in and output RTSP. Figure 4 (right panel) shows the overall aspect of the end-user interface with the embedded window for the video-streaming form the camera of the drone.

3.4 Implementation

Design Version 4 was implemented and tested end to end. The DJI Mobile SDK application was deployed to the Android device. The Python application was deployed to the Raspberry Pi 4B, and CGI Machine Vision was configured to connect to the local RTSP stream from the Pi (Figure 5).

One of the main requirements for this project was to provide a real time demonstration: *The system must be able to operate remotely to perform a demonstration, with minimal need for external internet, and power provided by the vehicle*.

Due to the nature of the demonstration taking place in an area without a local wireless network or mains power, the project was to be implemented using devices which could work off 12v DC power provided from the vehicle, or their own battery power. The drone, the drone controller, the android phone and the CGI developer laptop could operate off their own internal batteries for the purpose of the demonstration. Another restriction applied to the CGI developer laptop was to restrict the ability to create its own local wireless network. A network would be required to allow the devices to connect to each other for the demonstration. An iPhone would also be available for the demonstration which could broadcast its own personal hotspot network; however this would require all devices to be connected via wireless network, and did not offer full management of the network such as a management page to see connected devices and their IP addresses, or to reserve IP addresses for devices within the DHCP server.

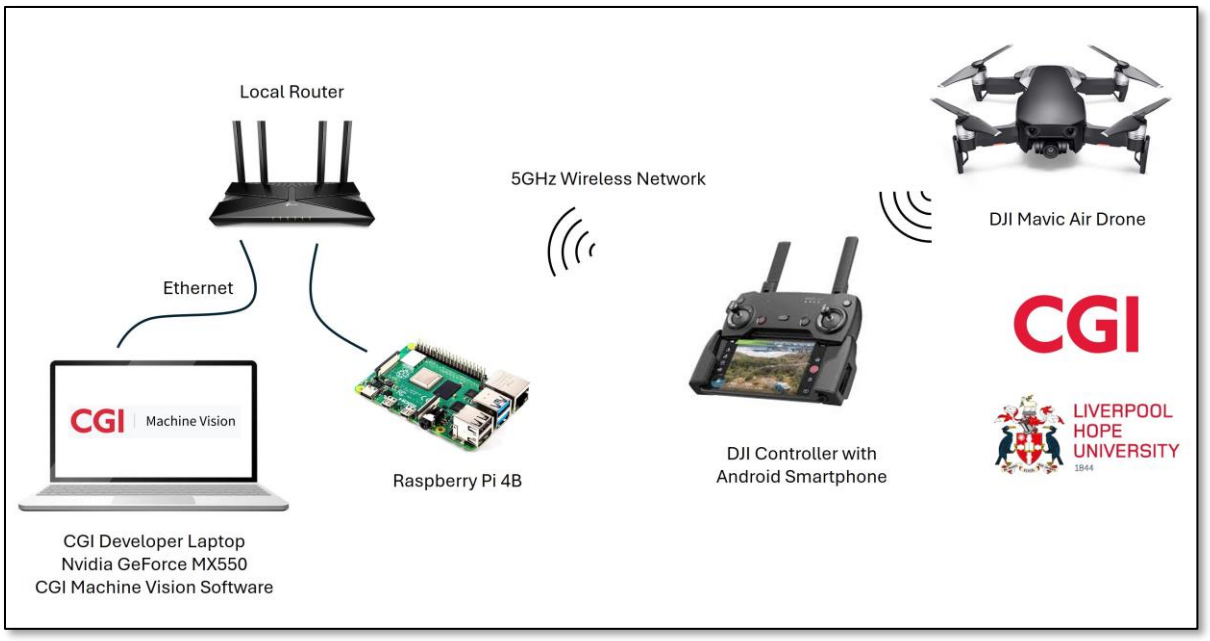
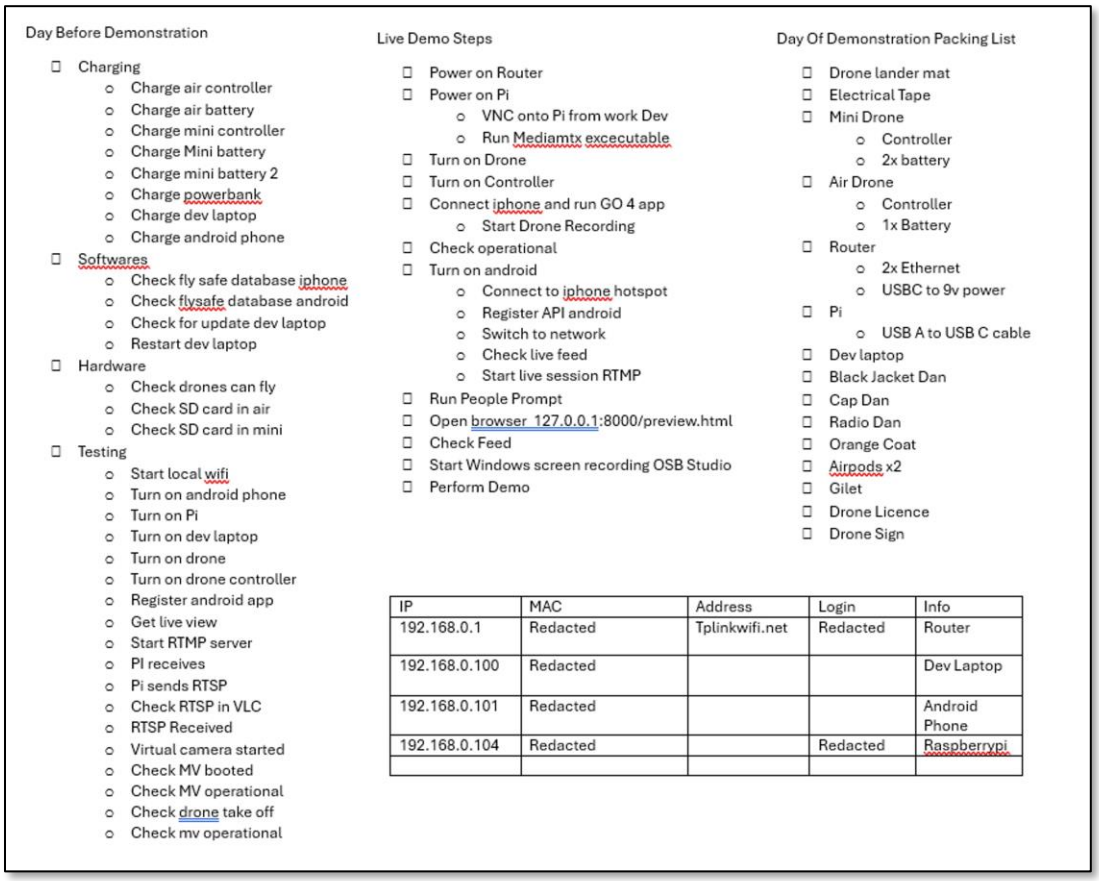


Figure 5: Technology Diagram

It was then decided that the local network for the demonstration would be provided by a wireless router. A TP-Link AC1200 Wireless Dual Band Router was selected to manage the local wireless network, providing 5 GHz wireless network for the Android phone, and ethernet ports allowing low latency wired connection for the Raspberry Pi and CGI Developer Laptop. The demonstration would be tested numerous times prior to the demonstration day, ensuring that each device could be powered on and configured in the shortest possible time. One constraint of the demonstration was the device with the shortest battery time, which in this case was the DJI Mavic Air drone with a battery time of only 30 min.

Multiple checklists were prepared to ensure that all required equipment was taken to site for the demo, all devices were fully charged, and a streamlined process for powering on and configuring each device to give the maximum amount of time for the operation of the demo. The checklist can be found in Figure 6.



Day Before Demonstration

- ☐ Charging
  - Charge air controller
  - Charge air battery
  - Charge mini controller
  - Charge Mini battery
  - Charge mini battery 2
  - Charge powerbank
  - Charge dev laptop
  - Charge android phone
- ☐ Softwares
  - Check fly safe database iphone
  - Check flysafe database android
  - Check for update dev laptop
  - Restart dev laptop
- ☐ Hardware
  - Check drones can fly
  - Check SD card in air
  - Check SD card in mini
- ☐ Testing
  - Start local wifi
  - Turn on android phone
  - Turn on Pi
  - Turn on dev laptop
  - Turn on drone
  - Turn on drone controller
  - Register android app
  - Get live view
  - Start RTMP server
  - PI receives
  - Pi sends RTSP
  - Check RTSP in VLC
  - RTSP Received
  - Virtual camera started
  - Check MV booted
  - Check MV operational
  - Check drone take off
  - Check mv operational

Live Demo Steps

- ☐ Power on Router
- ☐ Power on Pi
  - VNC onto Pi from work Dev
  - Run Mediamtx excecutable
- ☐ Turn on Drone
- ☐ Turn on Controller
- ☐ Connect iphone and run GO 4 app
  - Start Drone Recording
- ☐ Check operational
- ☐ Turn on android
  - Connect to iphone hotspot
  - Register API android
  - Switch to network
  - Check live feed
  - Start live session RTMP
- ☐ Run People Prompt
- ☐ Open browser 127.0.0.1:8000/preview.html
- ☐ Check Feed
- ☐ Start Windows screen recording OSB Studio
- ☐ Perform Demo

Day Of Demonstration Packing List

- ☐ Drone lander mat
- ☐ Electrical Tape
- ☐ Mini Drone
  - Controller
  - 2x battery
- ☐ Air Drone
  - Controller
  - 1x Battery
- ☐ Router
  - 2x Ethernet
  - USBC to 9v power
- ☐ Pi
  - USB A to USB C cable
- ☐ Dev laptop
- ☐ Black Jacket Dan
- ☐ Cap Dan
- ☐ Radio Dan
- ☐ Orange Coat
- ☐ Airpods x2
- ☐ Gilet
- ☐ Drone Licence
- ☐ Drone Sign

IP	MAC	Address	Login	Info
192.168.0.1	Redacted	Tplinkwifi.net	Redacted	Router
192.168.0.100	Redacted			Dev Laptop
192.168.0.101	Redacted			Android Phone
192.168.0.104	Redacted		Redacted	Raspberrypi

Figure 6: Testing check list and protocol

The car park adjacent to the sports complex at *Liverpool Hope University* was selected as the site for the demonstration. This location provided a good range of landscapes to demonstrate the capabilities of the system including a car park, a lightly wooded area, and a field, each providing different conditions for the demonstration. Permission from Liverpool Hope University was sought to perform the demonstration at the university. A Data Protection Impact Assessment and Health and Safety Risk Assessment were required, ensuring that any potential data protection impact had been assessed and mitigated where possible, along with health and safety risks and mitigation. Due to the DJI Mavic Air drone being greater than 250g, the aircraft was registered with the Civil Aviation Authority. The drone operator undertook a drone theory test by the Civil Aviation Authority and was granted a Flyer ID to fly drones. An Operator ID was also provided for the management and registration of the drone itself.

4 Testing and Results

Testing took place iteratively throughout the design and development process with minimal testing required of the final solution. Prior to the demonstration day, the solution was tested with positive results. The tests focused on the end-to-end implementation of the hardware, from the live feed from the airborne drone to the object detection of the video feed in CGI Machine Vision. The tests demonstrated that the system worked well, with reliability and performance. The testing identified one minor adjustment required for the DJI Camera Drone that would be used to capture cinematic video of the demonstration for the packaging a video.

4.1 Proof of Concept and Demonstration

A demonstration of the proof of concept was planned to test the effectiveness of the solution in real world applications and collect data on reliability and performance of the system, as well as the accuracy of the CGI Machine Vision software at person detection. The demonstration would also be filmed to allow for a video of the use case to be illustrated, along with the ability and limitations of the system. Planning of the demonstration focused on two elements, firstly the testing of the system in a real-world environment, and secondly the capture of a video to illustrate the use case for an emergency or law enforcement officers [14, 15]. The demonstration would make use of the standard CGI Machine Vision model, without any refinement or customisation for the specific application or use case in order to test the ability of the software on a previously unseen application to assess its adaptability. The large language model element of the model used the *bertsquad-10.onnx* open-source model, along with the proprietary in-house trained CGI Machine Vision image model. ***The prompt provided to the model for the purpose of the demonstration was a simple and intuitive command such as "Track the person".*** This prompt was entered into the CGI Machine Vision web GUI.

The demonstration of the proof of concept took place on a Sunday afternoon in March 2025 in a car park at Liverpool Hope University. Checklists were used to ensure that the devices were powered on and configured in the most efficient order to avoid delays and missed steps in the workflow in order to conserve device battery capacity. During the demonstration, the drone itself was manually piloted by the drone operator. Although the drone has the capability to active track persons or objects that are selected on the screen, the DJI Android SDK Application was required to be running in the foreground of the screen at all times to maintain the live video RTMP broadcast. As the DJI Android SDK Application did not include options to enable object tracking, it was not possible to command the drone to track an object autonomously while simultaneously broadcasting the video feed. An actor portraying the person of interest performed different behaviour to test the effectiveness of the vision model, acting out three sequences as below:

- Stationary person of interest
- Person of interest walking across the frame of the drone
- Person of interest walking around the corner of a building
- Person of interest walking into and standing in a lightly wooded area

Additionally, the drone performed a number of sequences:

- Stationary drone at a distance of around 60 m from the person of interest
- Stationary drone at a distance of around 35 m from the person of interest

- Drone following the person of interest
- Drone panning an area which included the person of interest.

For all demonstrations the drone maintained an altitude of around 50 ft or 15 m above ground level. The recording of results of the demonstration and the successfulness of the proof of concept were measured by reviewing a recording of the CGI Machine Vision processed video output showing detected objects, and measurements of the latency of the video feed. OBS Studio software was used on the Windows PC to capture the device's screen to record the output from CGI Machine Vision. The screen recording captured both the command line interface for running CGI Machine Vision, showing successful receipt of the RTSP video stream and Frames per Second, as well as the CGI Machine Vision web GUI, showing the live video feed from the drone, along with real time bounding box tracking of the person. During the demonstration the drone operator measured the latency of the system by commanding the drone to rotate and timing the number of seconds before the same rotation was visible on the CGI Machine Vision display

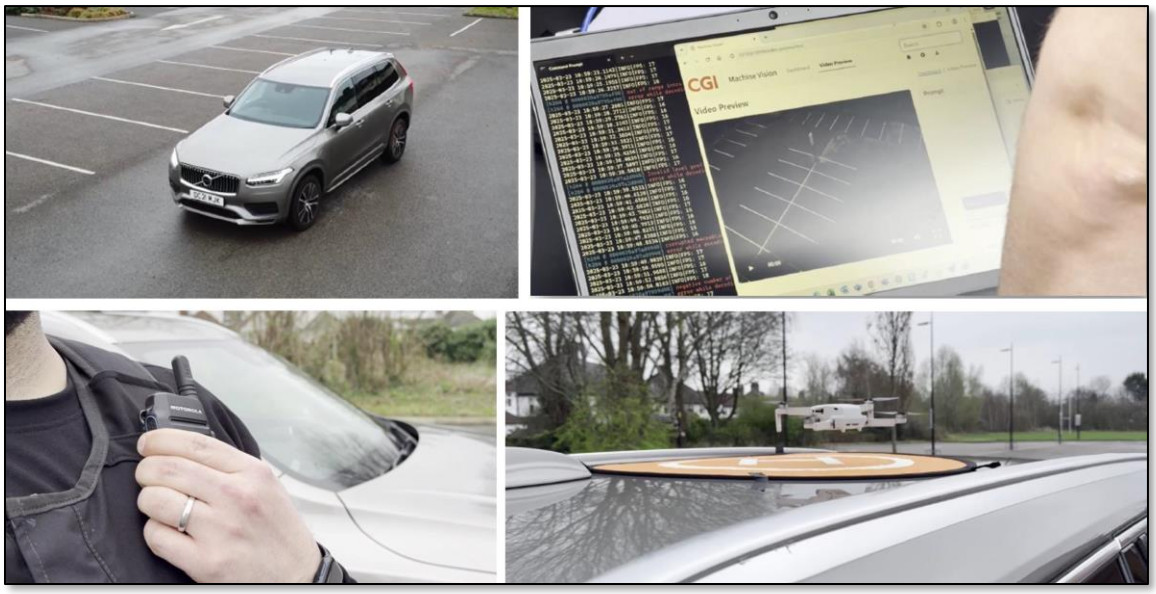


Figure 7: Cinematic Use Case

4.2 Cinematic Use Case Video

In addition to the technical demonstration of the device, cinematic video footage was captured for the creation of a use case demonstration video illustrating the real time use of the drone for law enforcement officers. The cinematic video included:

- Aerial footage of the vehicle arriving at the scene
- An actor portraying a law enforcement officer stepping out of the vehicle and pressing the button on their radio to illustrate an officer giving natural language prompts and commands to the drone
- Aerial footage of the drone taking off and landing on the vehicle

Footage was recorded from the DJI Mavic Air drone used for the demonstration, a DJI Mavic

Mini drone used for cinematic video capture, and an iPhone 16. Figure 7 shows the set-up of the use case

4.3 Results

This section presents the results, namely the Latency of the system along with Frame loss and the accuracy of the object detection, with a focus on real-time applications. During the demonstration, the live output video from the CGI Machine Vision web GUI was recorded using screen recording software on the CGI developer laptop. The drone operator also viewed the real time feed on screen during the demonstration. A video demonstrating the use case of this project, along with the video directly captured from the CGI Machine Vision web GUI is available on request.

A. Latency

One main component of this research is the applicability of the solution and technology to real-time applications, for the use case explored here being to provide aerial situational awareness to a law enforcement officer. Latency forms a key part of the applicability and suitability of the technology from the perspective that lower latency allows near real time feedback on a system to enabling a system to receive video, detect objects, and autonomously fly with high accuracy and reliability. Increased latency reduces the accuracy and reliability of such a system. Latency was measured between the time the drone's onboard gimbal camera receives a video image to the time that same image appeared on the CGI Machine Vision web GUI screen on the windows laptop. Latency was measured in multiple scenarios to understand latency throughout the system, namely:

- Latency between the drone action and the live video display in the Android App.
- Latency between the drone action and the live video display of the RTSP stream on the CGI Laptop using VLC Media Player.
- Latency between the drone action and the live video display of CGI Machine Vision within the CGI web GUI on the CGI Laptop.

Latency was measured by timing the seconds between the drone operator commanding movement or rotation of the drone and the time that the movement is visible on the relevant display.

- *Latency between the drone action and the live video display in the Android App.*

The latency between drone action and live video display in the android app was measured to be less than 0.5 s in all scenarios. The live video feed in the Android app mirrored the drone's movement with very low latency allowing the drone to be remotely piloted for the purpose of the demo using this video feed.

- *Latency between the drone action and the live video display of the RTSP stream on the CGI Laptop using VLC Media Player.*

VLC media player is a lightweight video playback software which allows for a RTSP stream to be received and displayed on screen. The latency between the drone action and the live video display in VLC media player running on the CGI laptop was measured during both a pre-demonstration test and the demonstration itself. The latency was reliably measured to be between the range of 3 seconds to 5 seconds, with the average latency of 4 seconds during the tests. This

latency related to the video feed passing through the DJI Android RTMP broadcaster component, the wireless network to the router, the ethernet cable to the Raspberry PI, processed by the python app from RTMP to RTSP and back to the CGI laptop using ethernet. This level of latency would likely allow real time applications with a high level of reliability and accuracy.

- *Latency between the drone action and the live video display of CGI Machine Vision within the CGI web GUI on the CGI Laptop.*

The latency from the drone action to the live video display of CGI Machine Vision was measured to be significantly longer than the other measurements. The latency was measured both during the pre-demonstration test and the demonstration itself and was measured to be between the range of 5 s and 12 s during the tests. The average latency from the readings was 10 seconds, with latency initially lowest on the startup of the system usually increasing to around the maximum latency after 60-90 s and then reducing to the average latency for the remainder of the operation. This latency demonstrates a significant increase compared to the video stream being received in VLC media player on the same CGI laptop. Most of this latency relates to the RTSP stream being read by CGI Machine Vision, the video being ingested into the software, the processing of the image through the neural network, the bounding box of the detected object being added to the video feed, and the feed being rendered in the browser. When running the CGI Machine Vision model using the laptop's internal camera the latency is relatively low, usually under 1 second, suggesting that the additional latency may be due to the way CGI Machine Vision receives the RTSP stream and passes it to the neural network for processing. This may be something that can be further refined and optimised.

B. Frame Loss

The video recording linked above shows the output of the CGI Machine Vision object detection captured from the web GUI. It was noted during the outdoor live video demonstration that examples of frame loss and compression artifacts were visible on the video screen. Frame loss can be seen in some parts of the video where the frame becomes significantly corrupted causing a black or dark screen to be shown briefly instead of the video feed. Additionally, compression artifacts are shown far more frequently on the output video, showing ghosting of the person being tracked, causing the CGI Machine Vision software to detect the person twice when only one person was present in the frame (Figure 8).

The DJI Drone uses a proprietary H.264 video encoding and compression to send the live camera view to the DJI Mobile SDK operating on the Android phone. The Android phone sends this video to an RTMP server on the raspberry pi using TCP. After conversion, the Raspberry Pi sends the video feed to the CGI laptop over RTSP. The H.264 video encoding and compression makes use of periodic I-Frames which are full resolution image frames of the video for reference. Between these I-Frames are P or B Frames which only provide information on pixels that have changed since the last I-Frame. This form of compression allows for high-definition video to be sent over lower bandwidth networks, however if packets of data from the P or B frames are lost along the way this is shown as frame freezing, smearing, tearing or ghosting within the video displayed. When a new full I-frame is periodically received, the video refreshes removing any previous artifacts. Packet loss through the wireless network, or at other points in the system is the cause of the compression artifacts shown on the video.

Despite these issues, CGI Machine Vision does continue to track the person of interest with

a high level of accuracy and reliability despite the poor quality of video feed provided.

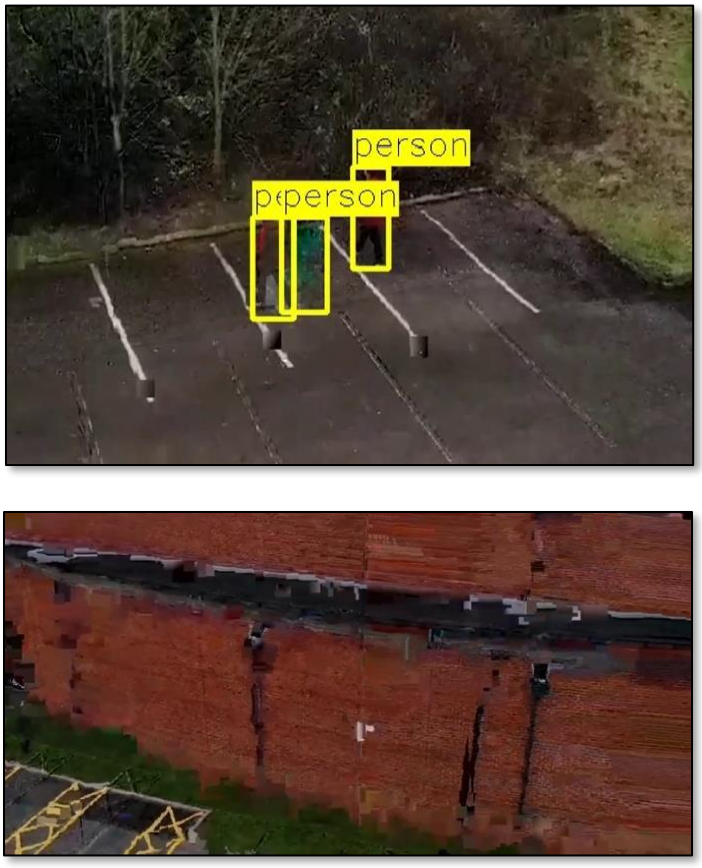


Figure 8: Example of Ghosting and Tearing (top and bottom panels, respectively)

C. Object Detection

The CGI Machine Vision model used for this demonstration was an image detection model trained in house by CGI for a wide range of applications. The model was not configured or refined for this particular use case to assess the ability of the standard model across previously unencountered applications and settings. The data and structure within the pre-trained model itself and design of the of the object detection framework are the intellectual property of CGI and are not explored here, however the reliability and accuracy of the object detection has been reviewed for its suitability for the case of providing real-time applications.

Stationary Drone, subject walking across the frame

Sections of the demonstration were recorded with the drone at distances of 60 m and 35 m from the person of interest, all at an altitude of 15 m. The drone is stationary in the air while the person of interest walks across the frame from one side to another. During these segments, CGI Machine Visio is able to reliably and accurately track the person of interest for more than 90% of the time

the person is on screen. The times when the person is not correctly identified by CGI Machine Vision are when there are significant compression artifacts, distorting the image.

Drone following the subject.

Segments of the demonstration were recorded with the drone actively following the person of interest, at an altitude of 15 m. During these segments, CGI Machine Vision is able to reliably and accurately track the person of interest for in excess of 80% of the time the person is on screen. During these segments there are more significant compression artifacts from packet loss, due to the movement of both the person and the drone itself. As before, the times when the person is not correctly identified by CGI machine vision are when there are significant compression artifacts on screen.

Figure 9: Person of Interest in a lightly wooded area, Altitude of 15 m

Subject in a lightly wooded area

Segments of the demonstration were recorded with the drone stationary or panning around a lightly wooded area with the person of interest in the wooded area, partially obscured by trees. The drone was flown at an altitude of 15 m (Figure 9). During these segments, CGI Machine Vision is not able to reliably and accurately track the person of interest. The duration that CGI Machine Vision can track the subject accurately is less than 50% of the time the person is on screen. Some compression artifacts are visible on screen during this segment, however even when there are little to no artifacts, CGI Machine Vision struggles to identify the person accurately and reliably.

———

These results show that the standard, un-refined CGI Machine Vision model has a high level of accuracy and reliability in tracking persons of interest at a range of distances but is less accurate and reliable when in more complex scenarios or where the person is obscured by trees. Table 1

reports the overall results vs the drone positioning with respect to the subject, according to the different tested scenarios.

Table 1: Results summary

Scenario	Drone Altitude [m]	Horizontal Distance [m]	Percentage of time person successfully tracked
Car Park Stationary Drone	15	35 and 60	90%
Car Park Moving Drone	15	variable	80%
Wooded Area Moving Drone	15	variable	50%

5 Discussion

The results demonstrate that the system, proof of concept and demonstration was successful in implementing a video object detection system using a robotic aerial vehicle with a view to be used for real-time applications.

A. Latency and Frame Loss

The results show that latency in the system can be variable and potentially has room for further refinement. In the vein of the real-time use case of tracking a person of interest for law enforcement personnel, the latency measured in this demonstration of an average of 10 seconds poses a significant challenge to the use of this solution for real-time applications. In this real-time use case, the drone would be required to scan an area for a person of interest, identify the person and track the person for a period of time. The person may be stationary or walking as shown in this demonstration but may also be running. Due to the technical limitations mentioned earlier, this proof of concept was not able to provide a bounding box to track back to the drone, however given the latency it can be assumed that the drone would not have been able to successfully autonomously track the person. The demonstration in the article Dynamic Object Tracking on Autonomous UAV System for Surveillance Applications [8, 16] was able to successfully track objects in real-time. The demonstration detailed in the article brought the computer vision model and edge computing onboard the drone, rather than near to the drone, significantly reducing latency in providing detection and returning instructions back to the drone. CGI Machine Vision is able to successfully run low latency object detection on a Nvidia Jetson board, as used in two of the publications referenced earlier. The recommendation for progressing this proof of concept forward would be to move the processing onboard the drone.

B. Object Detection Accuracy

The results from the tracking of the person of interest by CGI Machine Vision are very successful. The model was able to detect a person of interest with a high degree of accuracy. The image model used in the demonstration was a standard model trained on a wide range of images. The accuracy and reliability of the model could be improved with further refinement for this particular use case. Training the model on domain-specific data such as publicly available drone footage would allow for an improved person detection model from an aerial viewpoint. Additionally, working in collaboration with emergency operators and police forces, aerial drone footage could be used to train the data from emergency interventions, helicopter camera recordings, large event drone surveillance footage and other domain-specific scenarios, storing the sensitive recordings in a secure environment. Further training could allow for the model to differentiate between likely law enforcement officers that are likely to be in a scene to reduce false positive results.

C. Addressing the Research Questions

The research questions to be explored as part of this paper were as follows: how effective is the use of Video Object Detection with Robotic Aerial Vehicles? Can such a system provide real time applications with low latency? What operational benefits can be enabled by the use of this technology? How can the solution be kept secure? What does this research tell us about how the solution may need to be adapted for a real-world rollout?

- *How effective is the use of Video Object Detection with Robotic Aerial Vehicles?*

The results of the object detection from CGI Machine Vision show very positive results, with a high degree of accuracy and reliability of the object detection. It was anticipated that the novel scenario presented to the model may pose a challenge due to lack of domain-specific dataset training, however the model provided positive results despite this. The video object detection using a robotic aerial vehicle in this proof of concept and demonstration was very effective.

- *Can such a system provide real time applications with low latency?*

The term low latency can be subjective depending on the application and use case of the system. In this example, the use case of the system was to support real-time applications in the form of aerial situational awareness for law enforcement officers.

In the scenario of providing aerial situational awareness, an average latency of 10 seconds is positive, for example when searching a large area for a missing person or identifying a person in a crowd, however an average latency of 10 seconds would not be suitable for enabling feedback to autonomous flight model or tracking component. This limitation would require the drone in this proof of concept to be manually operated at all times. To evaluate the research question, the system can provide low latency results for real time applications, but not in all scenarios.

- *What operational benefits can be enabled by the use of this technology?*

This technology has a number of key advantages comparted to the current aerial situational awareness provision from services such as NPAS. The two main advantages demonstrated in this proof of concept are the ability for a robotic aerial vehicle-based solution to be deployed

instantaneously, and for the solution to be delivered at a significantly lower cost. This would allow for exponentially improved value and outcomes for officers and citizens through the deployment of many low cost autonomous drones for a significantly higher number of reported incidents.

- *How can the solution be kept secure?*

Applying this solution and technology for the use by law enforcement officers would require a significant evaluation of security considerations. If this proof of concept were to be taken forward, a refined solution would be developed and submitted for review and approval by security officers with a law enforcement agency. CGI Machine Vision allows for the model to be trained using secure domain-specific images, storing the images and resulting model within a secure environment, for deployment to edge devices on the drone itself. The security requirements would vary depending on the specific agency and would form part of any future solution design.

- *What does this research tell us about how the solution may need to be adapted for a real-world rollout?*

The research positively demonstrates that a proof of concept of operating video object detection using robotic aerial vehicles can be successfully deployed as a proof of concept and demonstrated. The demonstration in this project required significant manual configuration, setup and deployment on the day of the demonstration. A real-world rollout would require a fully autonomous system to maintain the hardware, launch and retrieve the hardware, and keep the hardware secure from tampering. These elements would be explored further as part of a minimum viable product stage following the proof of concept.

D. Hardware Suitability

The DJI Mavic Air drone was selected due to a number of features such as the ability to control the drone over a wireless network, the ability to actively track a person, and the availability of the Software Developers Kit, however a number of these features were unavailable or unsupported, resulting in some features of the proof of concept being dropped.

E. Potential for Further Refinement

This research provided significant learnings through the iterative design approach and review of different technologies to create the final solution. A number of recommendations for further development are documented below.

Drone - The DJI Mavic Air provided a good test bed for this proof of concept, however it should not be taken forward beyond this proof of concept. DJI provide a number of enterprise level drones and real-world management software. For example, the DJI Dock 3 drone compromises an autonomous remotely operated drone with an accompanying base station allowing the drone to be autonomously launched, retrieved, charged and protected [1]. In the specific use cases of applications with a national security and data sovereignty element, further analysis of any Software Developers Kits, libraries and software would need to be undertaken to ensure data sovereignty of

sensitive information.

Building based Drones - The example explored in this research was to develop a vehicle-based drone solution, however during the research an alternative building-based solution was considered. Vehicle-based drones would require a small footprint on the vehicle due to the size of the vehicle and other equipment located on top of the vehicle. Drones also make use of sensitive magnetic compasses and sensors that can be affected by other communications devices on a vehicle presenting further challenges. Additionally, vehicle-based drones would be readily accessible by the public presenting a risk of vandalism. Vehicle based drones would rely on a vehicle arriving on a scene before being launched. Alternatively, in larger built-up areas, drones based on top of buildings, or on secure land could provide a number of benefits. Drones could be distributed across areas of higher crime, with a larger physical footprint for additional hardware to autonomously launch and manage the drone. An inter-connected building-based drone network would be able to take flight as soon as an incident is reported instead of when an ambulance or a police vehicle arrives on scene. With airspeeds of up to 50mph a drone could arrive on scene significantly quicker than a vehicle and could provide a real-time video feed and situational awareness to an emergency coordination centre, a police command centre and operators or officers en-route to the scene. This approach would provide a number of significant benefits to further enhance the operational outcomes from such a solution.

Model and Latency optimisation - A new custom CGI Machine Vision model should be developed, including training on domain-specific image types in a variety of weather and time of day conditions. Any new model should be extensively tested, which may identify a need for further refinement of colour filtering to improve the reliability and accuracy of detecting persons of interest wearing specific coloured clothing or items. Latency must be optimised further if any proposed solution is to provide real-time autonomous aerial navigation and tracking of a person of interest, most likely using edge computing on board the chosen drone. This approach would also reduce the requirement for continual connectivity which may be important in areas with reduced connectivity infrastructure.

6 Conclusion

This research project demonstrated a proof of concept for real-time video object detection using a DJI Mavic Air drone and CGI Machine Vision software. The integration solution demonstrated potential for improving current aerial situational awareness provision. Other technologies and types of human/operator – drone interactions could be considered and integrated in the system in the future, considering that human gesture and physiological or behavioural signals could also provide proper information of the operator intentions especially in emergency scenarios where the rapidity and efficiency are compulsory [17-21].

While the system operated reliably in the demonstration, further work is required to improve the robustness and reduce latency for deployment in a real-world scenario. Further development should explore a refined technological solution and further software development for autonomous operation, enabling smarter autonomous decision making in line with other solutions that are currently proposed in the market [22].

Acknowledgement: This work was presented in dissertation form in fulfilment of the requirements for the BEng Robotics for the student Christoper Ingham under the supervision of EL Secco and D Reid from the AI and Robotics Labs, School of Computer Science and the Environment, Liverpool Hope University.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: This research has been supported by the company CGI and by the School of Computer Science and the Environment, Liverpool Hope University. The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] DJI. (n.d.). DJI Enterprise - Drone Solutions for Your Business. [online] Available at: https://enterprise.dji.com/.

[2] Pixhawk. (n.d.). Pixhawk Homepage. [online] Available at: https://pixhawk.org/ [Accessed 3 Apr. 2025].

[3] PX4 Open-Source Autopilot. (2019). PX4 Open-Source Autopilot. [online] Available at: https://px4.io/.

[4] ArduPilot (n.d.). Open Source Drone Software. Versatile, Trusted, Open. ArduPilot. [online] ardupilot.org. Available at: https://ardupilot.org/.

[5] Richards, N. (no date) CGI Machine Vision, CGI. Available at: https://www.cgi.com/au/en-au/solutions/cgi-machine-vision (Accessed: 13 March 2025).

[6] Manolescu, D., Reid, D. and Emanuele Lindo Secco (2024). Hardware and Software Integration of Machine Learning Vision System Based on NVIDIA Jetson Nano. Lecture notes in networks and systems, pp.129–137. doi:https://doi.org/10.1007/978-3-031-54053-0_10.

[7] AlJundi, Z., Alsubaie, S., Faheem, M.H., Almarashi, R.M., Qazi, E.-H. and Kim, J.H. (2025). Dangerous Objects Detection Using Deep Learning and First Responder Drone. International Journal of Digital Crime and Forensics, [online] 16(1), pp.1–18. doi:https://doi.org/10.4018/ijdcf.367034.

[8] Lo, L.-Y., Yiu, C.H., Tang, Y., Yang, A.-S., Li, B. and Wen, C.-Y. (2021). Dynamic Object Tracking on Autonomous UAV System for Surveillance Applications. Sensors, 21(23), p.7888. doi:https://doi.org/10.3390/s21237888.

[9] DJI Mobile SDK Reference. (n.d.). Available at: https://developer.dji.com/api-reference/android-api/Components/LiveStreamManager/DJILiveStreamManager.html.

[10] Dji.com. (2020). DJI Developer. [online] Available at: https://developer.dji.com/mobile-sdk-v4/ [Accessed 10 Apr. 2025].

[11] GitHub. (2025). DJI-SDK. [online] Available at: https://github.com/dji-sdk [Accessed 10 Apr. 2025].

[12] Z Isherwood, EL Secco, A Raspberry Pi computer vision system for self-driving cars, Computing Conference, 2, 910-924, 2022, DOI: 10.1007/978-3-031-10464-0

[13] J Lyons, O Anicho, EL Secco, Raspberry-PI based design of an interactive Smart Mirror for daily life, Digital Technologies Research and Applications, Scientific Published Limited, 3(2), 89-103, 2024, DOI: 10.54963/dtra.v3i2.259

[14] www.npas.police.uk. (n.d.). About Us | National Police Air Service. [online] Available at: https://www.npas.police.uk/about-us.

[15] March 2021 NPAS costs FOI (2021). https://www.westyorkshire.police.uk/sites/default/files/foi/2021-05/march_2021_foi_1124-21_npas_costs.pdf (Accessed: March 26, 2025).

[16] Tharmalingam K, Secco EL, A Surveillance Mobile Robot based on Low-Cost Embedded Computers, 3rd International Conference on Artificial Intelligence: Advances and Applications, 25, 323-334, (ICAIAA 2022) DOI : 10.1007/978-981-19-7041-2

[17] Bilawal Latif, N Buckley, EL Secco, Hand Gesture & Human-Drone Interaction, Intelligent Systems Conference (IntelliSys), 3, 299-308, 2022, DOI: 10.1007/978-3-031-16075-2

[18] T Scott Chu, A Chua, JA Jose, E Sybingco, C Espulgar, NJ Romblon, EL Secco, Development and Performance Analysis of Orthogonal Sonar Array for Autonomous Mobile Robot SLAM Implementation, ASEAN Engineering Journal (AEJ), 14(4) DOI: https://doi.org/10.11113/aej.v14.20687

[19] TS Chu, AY Chua, EL Secco, A Study on Neuro Fuzzy Algorithm Implementation on BCI-UAV Control Systems, ASEAN Engineering Journal (AEJ), 12, 4, 75-81, 2022, DOI: 10.11113/aej.v12.16900

[20] TS Chu, AY Chua, EL Secco, Performance Analysis of a Neuro Fuzzy Algorithm in Human Centered & Non-Invasive BCI, Sixth International Congress on Information and Communication Technology (ICICT), 2021 - Lecture Notes in Networks and Systems, 2, 241-252, Springer, ISBN 978-981-16-2380-6 – DOI: https://doi.org/10.1007/978-981-16-2380-6_22

[21] D Vasile, Hamzah AlZu'bi, EL Secco, Interactive Conversational AI with IoT Devices for Enhanced Human-Robot Interaction, Journal of Intelligent Communication (E-ISSN: 2754-5792), 2025, DOI: https://doi.org/10.54963/jic.v3i1.317

[22] BYD cars now have an on-vehicle DJI drone launch platform, The Verge 2025. [online] Available at: https://www.theverge.com/news/622963/byd-dji-vehicle-mounted-drone-launcher [Accessed 17 Apr. 2025].