SWARM UNMANNED AERIAL VEHICLES (UAVS)-BASED FOG COMPUTING PLATFORM SUPPORTINGINTERNET OF THINGS APPLICATIONS

O. H. El Sayed * S. M. Youssef O. M. Ismail

Department of Computer Engineering, College of Engineering and Technology, Arab Academy for Science, Technology and Maritime Transport, Egypt
osamahesham357@gmail.com, sherin@aast.edu, ossama@aast.edu

KEY WORDS: Unmanned Aerial Vehicles, Drones, Cloud-computing, Object detection, Video stitching, video stabilization.

ABSTRACT:

Swarm robots, particularly drone swarms, are commonly used in search and rescue, military, and detection missions. However, due to their limited computing resources, it can be difficult to handle computation-intensive tasks locally. To address this, cloud-based computation offloading is often used, but it may cause latency issues for time-sensitive tasks like object recognition and path planning. Additionally, in environments with no wireless infrastructure, such as disaster areas or battlefields, cloud computing may not be feasible. To solve this problem, this paper proposes integrating Fog Computing with Swarm of Drones architecture. The paper also formulates the problem as a task allocation problem that minimizes energy consumption while accounting for latency and reliability constraints. To improve swarm autonomy, an integrated framework is proposed, and a testbed development is introduced to support this architecture. The paper reviews existing literature on UAV swarm and proposes a new architecture to enhance swarm autonomy.

1. INTRODUCTION

MAVs, which are compact and maneuverable unmanned aerial vehicles (UAVs), have played a crucial role in advancing the use of unmanned systems for purposes such as surveillance, search and rescue operations, and mapping (Chung A, 2018). Despite their usefulness, MAVs face constraints in terms of their endurance, cargo-carrying capacity, and sensing and computational capabilities. To overcome these limitations, aerial swarms have been developed that work together to accomplish intricate missions. The progress of swarm technology in MAVs hascreated fresh prospects in the realm of robotics and unmanned systems, and ongoing progress is expected to result in novel applications in fields like transportation and logistics (Xi, 2020).

UAV swarms are collections of airborne robots that collaborate either under manual supervision or through onboard processors that operate autonomously. Multi-layered swarms are capable of greater efficiency as a result of having specialized leader drones that manage the actions of multiple drones. Data tasks can be assigned to individual drones, with more computationally intensive tasks being offloaded to servers or processed in the cloud (Tahir, Böling, & Haghbayan, 2019). A model that tackles the problem of redundant data generated by UAV swarms by utilizing feature extraction stitching and geometric relational matrix calculations to generate real-time panoramas, while minimizing the transfer and storage of redundant pixels, as demonstrated in a military assessment. Drones are capable of seamlessly connecting to the internet and gathering as much as 500 gigabytes of data every hour. This data possesses immense potential for big data analysisand could have far-reaching implications across numerous industries, offering a potential solution to some of the most pressing issues confronting humanity (Andrew, 2017).

UAVs offer swift, economical, and secure solutions for various civil and military operations. This article will additionally spotlight the benefits of UAV Fog, a Fog computing platform based on UAVs, for Internet of Things (IoT) applications. This innovation makes use of the advantages and capabilities of both Fog computing and UAVs to provide efficient support for certain IoT applications. UAV Fog enables speedy deployment of Fog capabilities in hard-to-reach or remote locations to effectively serve dynamic IoT applications. Equipped with Fog computing capabilities, a UAV can travel to a specific site to provide support forthe local IoT applications.

2. OVERVIEW OF AERIAL SWARM APPLICATIONS

2.1 Security and Surveillance

Ready-made drones that come equipped with advanced monitoring sensors are highly effective for aerial surveillance insecurity applications, while drone swarms offer even greater surveillance capabilities and are expected to become more prevalent. UAV fleets with advanced intelligence can adjust to environmental changes, making them a formidable tool for surveillance operations, with swarms further enhancing their capabilities. Various sensors, such as RF, RADAR, optical, and acoustic sensors, can be utilized to detect and track objects, and their combination results in improved accuracy. Swarms are particularly useful in anti-UAV and surveillance systems, as they can cover larger areas in shorter periods of time than individual UAVs (Saska M, 2016).

2.2 Collaborative Transportation

Recent studies have showcased the use of small UAVs in tandem to transport larger payloads, achieved through the implementation of decentralized control laws and a centralized motion capture system for state estimation. The system employs a decentralized approach, where each quadrotor independently estimates its pose by utilizing visual-inertial odometry and interacts with neighboring quadrotors to transport rigid rod payloads while evading obstacles and maintaining formation, thereby obviating the requirement for motion capture systems (Ritz R, 2013).

* Corresponding author

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-1041-2023

2.3 Environmental Monitoring

An assemblage of UAVs equipped with sensors was autonomously deployed to fly over flood-prone areas, capturing high-resolution images and videos, and providing precise flood maps and real-time analysis to aid disaster management authorities. The swarms' ability to cover larger areas in a shorterperiod, and its capability to fly over flood-affected zones autonomously, allowed for quicker and more comprehensive data collection, resulting in improved flood management and relief efforts (Abdelkader M, 2014).

2.4 UAV-based Fog computing platform for the Internet of Things applications

Drones are limited in terms of their battery life and computing capabilities. If their resources are not sufficient to fulfill local and swarm-level requirements, they can offload tasks to other devices. However, a centralized approach that relies on the cloud for offloading may not always be suitable due to latency, quality of service, and energy consumption constraints. Instead, edge servers can be used for computation-intensive tasks, a concept known as "fog computing". Although edge servers are closer to drones and communication is faster, challenges arise when the swarm moves to locations far away from the edge servers. Placing edge servers in a fixed and trusted position can be difficult, leading to the development of collaborative edge-to-fog computing. This architecture allows drones to act as edge nodes, processing local and offloaded data in real-time. Efficient solutions are needed for orchestrating groups of drones in the field to achieve specific mission objectives (Dadmehr Rahbari, 2021).

Federated Learning (FL) and Deep Reinforcement Learning (DRL) are distributed strategies that have been used in MEC (Mobile Edge Computing) and drone applications, respectively. Federated Deep Reinforcement Learning (FDRL) combines the benefits of both FL and DRL, reducing network overhead, bandwidth consumption, and latency in a swarm of drones (Dadmehr Rahbari, 2021).

To evaluate the learning method in object detection applications, the accuracy of detected objects can be analyzed by rating the drones based on their properties and updating the model accordingly. This method allows drones to update their positions based on received ratings from neighbors (Dadmehr Rahbari, 2021).

3. RELATED WORKS

There are various software packages available, such as Adobe Photoshop, and cameras like the Samsung Gear360, that can stitch together multiple overlapping images to create a wide-angle or 360-degree view. These packages use sophisticated feature detection, matching, and alignment algorithms to create seamless panoramas that can be used for various applications, including virtual reality (VR) and immersive media (Y. Wang, 2015).

For applications like Google Street View, CMOS cameras with fixed geometries and lower sampling rates are used to capture video, which then requires video stitching for processing. This involves aligning and merging multiple frames to create a continuous, panoramic view of the environment (Y. Wang, 2015). The stitching process differs on the cameras motion, there are two types of camera positioning:

3.1 Static Camera

The process of video stitching involves creating a stitching template by selecting frames and using it to stitch together subsequent frames. However, moving objects can result in ghosting. To mitigate this issue, different techniques have been developed (Wei LYU, 2019). One method to address the issue of moving objects in video stitching involves incorporating the moving content into the final images using a conventional algorithm. This is followed by an object detection step, and reliable alignment information is provided by spatially neighboring videos for enhanced accuracy (J. Li, 2015). A model for video stitching was proposed for surveillance applications that uses a coarse-to-fine process. This approach first stitches together background layers when there are no moving objects present. Multiple layersare then generated, each containing different objects, by clustering matched feature pairs with consistent homography. The resulting stitched video provides a comprehensive view of the surveillance scene (Lin, 2016). In order to avoid issues such as missing data, artifacts, or ghosting caused by the presence of moving objects, an analysis of gradient variations in the overlapping regions can be used to adjust the best fit seam (F. Perazzi et al., 2015).The spatial-temporal mesh optimization framework enhances the geometric alignment of input video frames by prioritizing important areas such as spatial and temporal edges while minimizing their impact. This approach optimizes a mesh that considers both spatial and temporal information to achieve accurate alignment and prevent significant distortions in the resulting stitched video (Heng Guo, 2016).

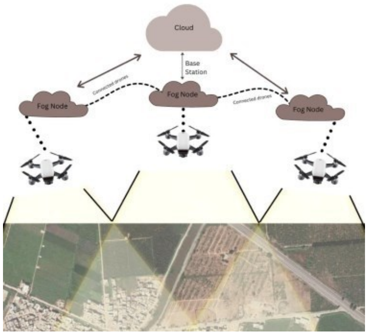
3.2 Moving Camera

Image stitching and stabilization techniques can be combined to address the issue of shaky footage in UAV and smartphone videos. However, when the camera is in motion, it can still be challenging to capture stable footage (F. Perazzi et al., 2015). Applying image stitching algorithms directly to shaky video frames has drawbacks, such as perspective distortions and lack of consideration for temporal smoothness. This can lead to visible jerks in the output (Gu, 2015). The traditional challenges of video stitching have been addressed through the introduction of hardware solutions such as the FC-110 full view camera and theGoPano method. However, these solutions can be expensive andchallenging to implement as they typically require unstructured camera arrays or static cameras (Y. Wang, 2015). The approach to video stitching and stabilization involves a two-step transformation process. The first step is inter-transformation, which aligns the frames spatially, and the second step is intra-transformation, which provides temporal smoothness. This approach employs a mesh-based warping method and a bundled-paths method for the synthesis of virtual camera paths (Guo, 2016).

The CoSLAM system is a method that integrates 3D reconstruction and camera pose estimation to enable video stitching. It employs the LPVW (Line-preserving video warp) method for mesh-based warpingand stabilization. However, this system may encounter difficulties in areas with uniform textures and in regions with abrupt changes in depth (H. Jiang, 2012). Video stitching is a process that involves creating a single, seamless video by stitching together multiple frames or videos. To do this, a template is constructed by stitching selected frames together using image stitching algorithms. This template is then used to stitch subsequent frames or videos together. However, the presence of moving objects can cause ghosting and blurring in the final stitched video. To address this issue, foreground detection techniques areemployed to identify and isolate moving objects. This helps to ensure accurate stitching and reduces the appearance of ghostingand blurring in the final video (Wei, 2019). Image stitching is a commonly employed technique in computer vision and graphics to overcome the limitations of field of view (FOV). It is an essential method for creating panoramas, wide-angle videos, and enhancing the accuracy and reliability of perception and navigation systems in autonomous vehicles (Gu, 2015).

Fig. 1, The image demonstrates the concept of expanding the field of view (FOV) by using video stitching from separate cameras that acts like fog nodes and communicate with the cloud. While multi-view image stitching methods are commonly used for tasks such as generating panoramas, 360-degree views, and virtual reality, the field of multi-view video stitching has not received as much attention as traditional image stitching, despite its potential applications in areas such as autonomous vehicles and surveillance (Guo, 2016). Multi-view video stitching is a process that involves merging multiple video clips with overlapping fields ofview, resulting in a panoramic video that offers a wider field of view while preserving a consistent relative geometry throughoutthe stitching process (Lin, 2016). When it comes to multi-view video stitching, mobile device camera jitter and other video processing problems can pose additional challenges. Video stabilization techniques can be applied to reduce shaky movements. To tackle hazardous environments or to achieve advanced capabilities beyond human limits, robotic systems equipped with advanced sensors and actuators can be utilized.

Figure 1. Increasing the field of view using cameras' views by video stitching the views and communicate with cloud.



Video stitching involves several challenges that have been extensively studied in literature. These include video stabilization, video synchronization, efficient large-size multi-view video alignment and panoramic video stitching, color correction, and blurred frame detection and repair with the enormous amount of data collected from videos and the computational power needed to support all of that in real time model.

4. PROPOSED CLOUD-BASED INTELLIGENT SWARM DRONES FOR OBJECT DETECTION MODEL

In this section, the proposed Intelligent Swarm Drones for Object Detection Applications will be introduced. The target is to simulate the cooperative process of N drones. Experiments will be carried out with Two DJI Ryze Tello drones applied on distinct scenarios, Webots simulator will be used to test the system using more than two drones. Fig. 2 shows the proposed model includes 4 main phases: Receiving of live streams among Drones, video frame processing, stitching phase and panoramic construction phase, and object detection phase.

4.1 Receiving of live streams phase

The framework methodology for multi-view video stitching using multiple drones includes controlling and receiving live streams from drones with overlapping areas, ensuring synchronization of videos, and minimizing delays. The Tello quadcopter serves as a receiver for commands through its Wi-Fi hotspot, and individual drones are connected through their own Wi-Fi hotspots for swarm control. (Flores, 2019). The Tello drones have the same IP and UDP port for commanding and receiving live streams, making it necessary to use port rerouting methods to receive multiple feeds from multiple drones on the same device.Each drone is connected to a Raspberry Pi and a Wi-Fi adapter to connect to the drone and access point, allowing for both controlling a swarm and retrieving video feed from the drones, see Fig. 3.

The whole system is managed using a cluster of three raspberry PI 4 8Gwhich receive the videos and start the stitching process. Theredirecting of the video streams was done through a raspberry pi zero on top of the Tello as follows:
Raspberry Pi IP Address: **192.168.1.120**
Port Where Video Feed is Received: **11111 Port Where Video Feed is Changed to 11117**
Code for implementing the redirecting:
sudo iptables -t nat -A PREROUTING -s 192.168.1.120 -p udp --dport 11111 -j REDIRECT -- to-port 11117

4.2 Video frame processing phase

The pre-processing stage in real-time video stitching aims to ensure consistent spatial and temporal resolutions.

In video stitching scenarios where the cameras are static, the spatial resolutions of the n video streams are denoted as (W1 x H1), (W2 x H2), ..., (Wn x Hn). Bilinear interpolation employs a linear interpolation approach, taking the weighted average of the four nearest pixels to approximate the value of a new pixel. This method is simple and efficient, providing a reasonable estimation of the pixel value. On the other hand, bicubic interpolation utilizes a 16-pixel neighborhood, forming a 4x4 grid. It employs a cubic polynomial to estimate the value

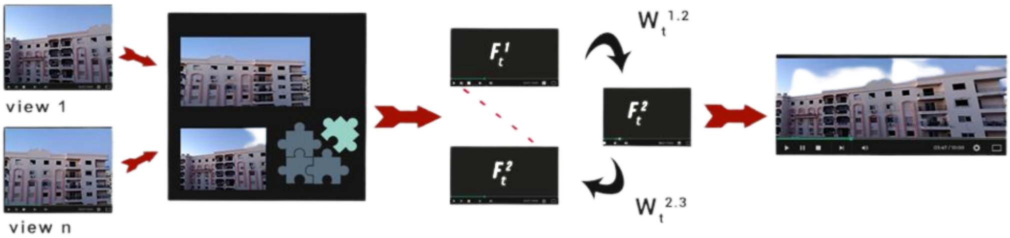


Figure 2. The flow of the framework from getting the video frames till arriving at the final output.

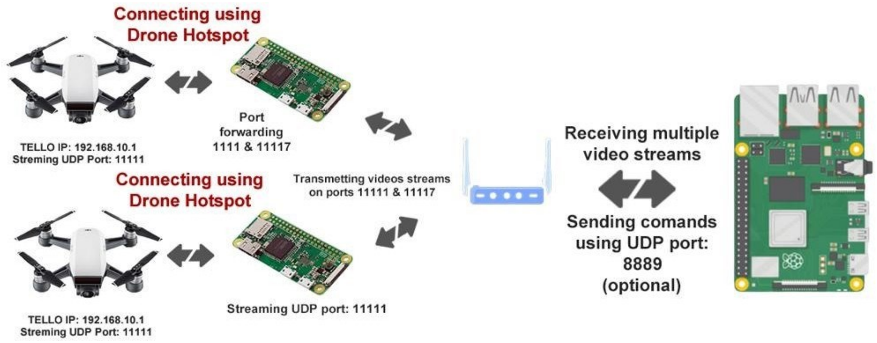

Connecting using Drone Hotspot
Port forwarding 1111 & 11117
TELLO IP: 192.168.10.1
Streming UDP Port: 11111
Connecting using Drone Hotspot
Transmetting videos streams on ports 11111 & 11117
TELLO IP: 192.168.10.1
Streming UDP Port: 11111
Streaming UDP port: 11111
Receiving multiple video streams
Sending comands using UDP port: 8889 (optional)

Figure 3. Port forwarding to receive multiple video streams at same time.

of a new pixel. By considering a larger set of neighboring pixels, bicubic interpolation provides a smoother and more accurate representation of the image, which can be beneficial in certain applications where finer details are important. While bilinear interpolation is generally faster, bicubic interpolation is preferred when higher image quality is desired or when the images contain intricate details. The choice between the two interpolation methods depends on the specific requirements of the application and the available computational resources. It is important to note that neither bilinear interpolation nor bicubic interpolation is specifically tied to camera movement. The selection of the interpolation method is based on the desired image quality and the trade-off between accuracy and computational complexity. In scenarios where the camera is moving, Bilinear Interpolation is used. In both cases, the fundamental concept behind interpolation is to utilize the nearest pixels to a new pixel to estimate its value.

Let FPS1, FPS2, ..., FPSn be the frame rates of the n video streams. For moving camera scenarios, Linear Interpolation is used to estimate missing frames between frames of different video streams. Linear Interpolation uses a straight line to estimate the value of a new frame based on the values of its two nearest frames. For static camera scenarios, Spline Interpolation is used, as it uses a smooth curve to estimate the value of a new frame based on the values of its surrounding frames. The basic idea behind Spline Interpolation is to fit a smooth curve throughthe surrounding frames and use this curve to estimate the value of the new frame.

Pre-processing in real-time video stitching involves mathematical techniques to harmonize resolutions and frame rates, ensuring a seamless and visually appealing final product free from artifacts.

4.3 Stitching and panoramic construction phase

4.3.1. Registration Phase: Brown and Lowe method refers to the scale-invariant feature transform (SIFT) algorithm, which identifies keypoints or interest points that are distinctive and invariant to scale, rotation, and affine transformations. These keypoints are often located in areas of high contrast, which can include blob-like regions, but they are not limited to them. They can also correspond to corners, edges, or other types of distinctive features, making it suitable for use in different scenarios. Other popular feature-based methods for stitching registration include the speeded-up robust features (SURF) algorithm and the oriented FAST and rotated BRIEF (ORB) algorithm (El Shehaby, 2019).

4.3.2. Fusion phase: Blending methods like alpha "feathering" and Gaussian pyramid can be employed in the fusion stage of stitching to combine images smoothly. Alpha blending is more effective when the images are properly aligned, while Gaussian pyramid blends images at various frequency bands, producing a more polished outcome (El Shehaby, 2019). Considering these factors, Gaussian pyramid blending is generally preferred over alpha blending for stitching images captured by moving cameras. Its ability to handle misalignments and produce seamless blends makes it more suitable for maintaining visual consistency and reducing artifacts in the stitched output. While Gaussian pyramid blending may require more computational resources compared to alpha blending, the improved quality and seamless blending results make it a preferred choice in scenarios involving moving cameras.

The framework utilizes bundled paths and a grid-based detection method with an adaptive local threshold and KLT tracker to estimate inter and intra motions. Bundled-path stabilization helps to mitigate perspective distortions and parallax, while the grid-based detection method detects features within the video, (see Fig. 4.a and 4.b), for feature detection and matching.

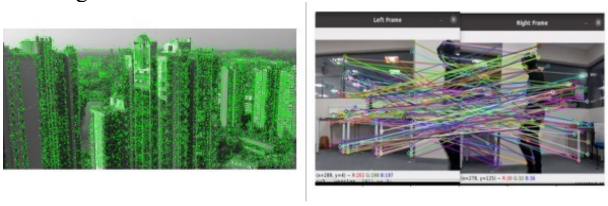


Figure 4.a. Illustrates feature detection, **Figure 4.b.** Illustrates the feature matching.

4.4 Object detection phase

YOLOv4 is an advanced object detection algorithm that provides both speed and accuracy in identifying objects within an input image. It achieves this by dividing the image into a grid and predicting the bounding boxes and objectness scores for each cell. This process is made possible through a highly optimized CNN architecture that includes multiple residual blocks. YOLOv4 also employs logistic and softmax regression to predict objectness scores and class probabilities, respectively, and predicts bounding box coordinates relative to the cell coordinates. Moreover, it is highly configurable, making it suitable for various applications.

4.5 Fog nodes initiation

In the proposed fog computing system, each DJI Tello drone is connected to an onboard Raspberry Pi Zero that serves as a gateway to forward the video streams. Since all Tello drones use the same port to stream video, the Raspberry Pi Zero is configured to port forward the video streams to the Raspberry Pi 4 at the main station.

At the main station, the Raspberry Pi 4 serves as the fog node that receives the video streams from the Raspberry Pi Zeros, processes them using the video stitching algorithm, and publishes the stitched output to the desired location. The Raspberry Pi 4 acts as a gateway between the drones and the cloud, enabling real-time data processing and analysis at the edge.

This distributed architecture provides several benefits over centralized processing. Firstly, it reduces the latency and network traffic by processing the data locally, leading to faster response times. Secondly, it reduces the dependency on cloud services for processing and storage, which can be costly and unreliable in remote areas with limited connectivity. Finally, it enables scalability and flexibility by adding more fog nodes or drones to the system, making it suitable for a wide range of applications.

Overall, the proposed fog computing system using Raspberry Pi Zero and Raspberry Pi 4 provides an efficient platform for real-time video stitching using DJI Tello drones. This system demonstrates the potential of fog computing in edge environments by enabling distributed computing, low latency, and scalability.

4.6 Webots simulation environment

In this research, we propose a fog-based system for real-time video stitching using 4 DJI Mavic Pro drones on Webots, an open-source robot simulator, see Fig. 5. To initiate the system, each drone was programmed to capture and stream video using Python code and the DJI Software Development Kit (SDK). Then the video stitching algorithm which was explained earlier is implemented, after it was optimized and tested using pre- recorded videos, as shown in the experimental results section later on. Then Robot Operating System (ROS) was installed and configured on a local server, which acted as the main station for the fog-based system.

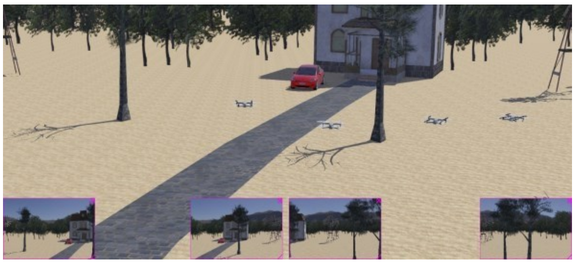


Figure 5. Illustrates the simulation with four drones.

To enable fog computing, each drone was configured to stream video to the local server using the ROS communication protocol, which provided low-latency and efficient data transfer. The server was configured as a fog node, which received the video streams from the drones, processed them using the video stitching algorithm, and published the stitched output to the desired location. A load balancing technique was also implemented to distribute the workload among the fog nodes and optimize the processing time.

The fog-based system using DJI Mavic Pro drones on Webots provides several benefits over traditional centralized processing methods. Firstly, it reduces the dependency on cloud services and enables real-time data processing and analysis at the edge. Secondly, it provides low-latency and efficient data transfer, which is critical for applications such as video streaming and analysis. Finally, it enables scalability and flexibility by adding more fog nodes or drones to the system, making it suitable for a wide range of applications.

Overall, the proposed fog-based system using 4 DJI Mavic Pro drones on Webots provides an efficient and scalable platform for real-time video stitching, demonstrating the potential of fog computing in edge environments. Further work can explore the optimization of the video stitching algorithm, integration of additional sensors, and testing the system in real-world scenarios. The simulation tool videos will be used in future to generate a dataset for video stitching with the ability of using fog computing for processing the outputs.

5. EXPERIMENTAL TESTING

Tello drones were used to capture videos and the processing of these videos was carried out using a cluster of three RPI 4 all over clocked to 2.3 GHz. Additionally, there were three main performance metrics used throughout the process.

1) Delay of the output video: This was measured by recording the time before and after the processing and computing the difference between these times. This metric identify any delays in the output video and optimize the processing pipeline for faster video output.

2) Stability score: This metric measured the smoothness of a stitched video, as in equation 1 below. A high stability score, close to 1, indicated a smooth and stable stitched video. This metric evaluates the quality of the output video and identifyareas for improvement (S. Liu, 2013).

$$S = \frac{1}{N}\sum_{i=1}^{N} E_i \quad (1)$$

where S = stability score
N = the total number of retained feature tracks with a length greater than 20 frames.
E_i = the energy percentage over the selected frequency components for the i-th feature track.

3) Stitching score: This metric measured the quality of the stitching in a stitched video, as in equation 2 below. A low stitching score indicated good alignment and high-quality stitching, while a high stitching score suggested potential issues with the stitching. This metric evaluates the overall qualityof the output video and identify any potential issues that may need to be addressed (C. Buehler, 2001).

$$S = f(Q, A) \quad (2)$$

where S = Stitching score
Q = set of quality parameters that measure the alignment and quality of the stitching, including sharpness, color consistency, geometric accuracy, and visual artifacts.
A = set of quality metrics or parameters that measure the alignment and quality of the stitching which were mentioned in Q.
f() = combination of these quality metrics and their respective weights to compute the stitching score.

5.1 Real-life scenarios

The first scenario pertains to the capture of multiple images of a static scene using cameras that remain fixed in position and do not move during the image capture process. The objects in the scene are also static and do not undergo any movement or change in position between the different images that are captured (see Fig. 6).

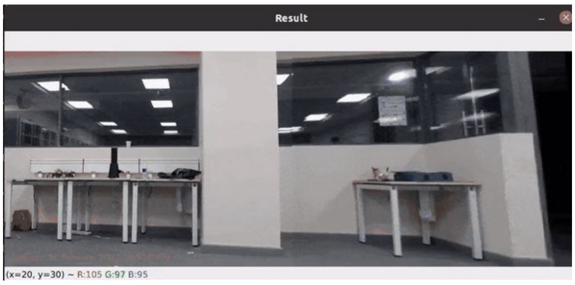


Figure 6. Illustrates the panorama view.

The second scenario involves capturing images of a scene using stationary cameras that do not move during the image capture process, but where the objects in the scene are in motion, (see Fig. 7). The primary challenge in this scenario is to detect and track the moving objects in the scene accurately while accounting for changes in lighting, shadows, or occlusions. This requires the application of advanced computer vision techniques, such as object detection, tracking, and recognition, along with the ability to handle multiple object trajectories and occlusions.

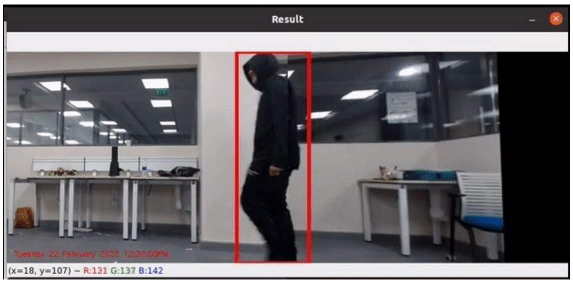

Result
Tuesday 22 February 2022 12:30:06PM
(x=18, y=107) ~ R:131 G:137 B:142

Figure 7. Illustrates the panorama view.

The third scenario pertains to the capture of images of a stationary scene using cameras that are in motion, (see Fig. 8). The primary challenge in this scenario is to achieve video stabilization, which involves removing camera shake and enhancing the quality of the final video. This can be accomplished through advanced computer vision algorithms that analyze the camera's motion and compensate for any movements or vibrations, resulting in a smoother and more engaging viewingexperience.

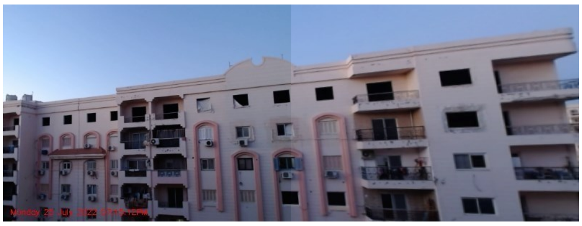


Figure 8. Illustrates the panorama view.

5.2 Edge to fog computing in drones' swarms

Drone swarm edge computing and fog computing are both important technologies that enable efficient processing of data and computation in distributed environments. Edge computing refers to the practice of processing data locally at the edge of a network, rather than sending it back to a centralized data center for processing. In the context of drone swarms, (see Fig. 9), edge computing can help reduce latency and bandwidth requirements by processing data directly on the drones themselves. This allows for faster decision-making and more efficient use of network resources.

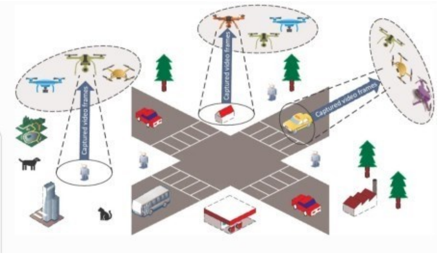


Figure 9. Illustrates a swarm of drones collecting data.

Fog computing, on the other hand, refers to the practice of extending cloud computing capabilities to the edge of a network. Fog computing provides a layer of intermediate computing resources between the drones and the centralized cloud, enabling more efficient processing of data and better management of network resources. In the context of drone swarms, fog computing can help optimize resource usage and reduce latency by providing computing resources closer to the drones themselves, (see Fig. 10).

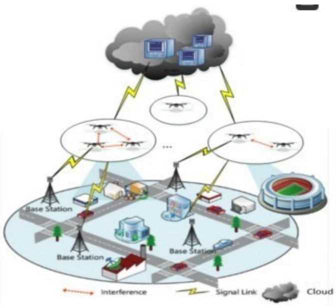


Figure 10. Illustrates a swarm of drones communicate to the cloud.

Both edge and fog computing are important in the context of drone swarm technology, as they enable more efficient use of resources and faster decision-making. By leveraging these technologies, drone swarms can become more effective and scalable, enabling a wide range of applications across many different industries. The objective of developing a cloud system is to provide autonomous mission planning, monitoring, and control as well as to enable a global data space where data can be easily transferred and managed within the system. The data collected throughout the mission can then be used for the analysis of the autonomous mission.

5.3 Results

The delay in the output video is estimated to be 580 milliseconds which identifies the output video rate by approximately 2 frames per second. Table 1 presents the results of the three trials, which show a consistent high stability score, indicating that the stitched videos were relatively smooth and stable throughout all experiments. However, the stitching score varied significantly across the three trials, suggesting lower quality stitching with less alignment compared to the first trial. Notably, the stability score remained high despite the independent and varied environments of each experiment, while the stitching score appeared to be more affected by specific environmental factors and conditions. Therefore, a high stability score is desirable, indicating a smoother and more stable stitched video, while a lower stitching score is also desirable, indicating better alignment and higher quality stitching.

	Scenario I	Scenario II	Scenario III
Stability Score	1.00	0.93	0.90
Stitching Score	0.67	1.01	1.02

Table 1. The comparison of the "Stability Score" and "Stitching Score" for different experimental scenarios.

6. EXPERIMENTAL RESULTS AND DISCUSSION

6.1 Video Stitching Dataset

The model utilized videos from a dataset that had been usedin previous papers to compare stitching quality results with otherresearch, as noted in paper (Heng Guo, 2016). The dataset included six videos,of which two were captured while in motion, causing shaky frames and temporal jitters that could affect the final stitching quality. The dataset is used to compare the results with a stable model and identify the possibility of deploying the model on fog node (RPI 4) instead of a computer.

6.2 Object Detection Dataset

The COCO dataset is an extensively utilized benchmark forobject detection tasks, comprising over 330,000 images and 2.5 million object instances spanning across 80 categories where the following categories was the main focal point Person, Bicycle, Car, Motorcycle, Airplane, Bus, Train, Truck and Boat. The dataset offers highly precise annotations that enable effective training and testing of computer vision models. The training process of YOLOv4 runs 6000 epochs and took serval hours.

6.3 Performance Measures

6.3.1. Video Stitching metrics: Table 2 presents the stability and stitching scores of the tested examples. The results indicate a noteworthy enhancement in stability for all examples based on the scores obtained with average improvement in the stitching score values.

Video	Length	Stitching score		Stability score	
Video 1	0:00:13	0.99	0.9	0.83	0.88
Video 2	0:00:12	0.54	0.51	0.78	0.82
Video 3	0:00:10	1.07	1.01	0.82	0.86
Video 4	0:00:13	1.01	0.89	0.88	0.89
Video 5	0:00:15	0.43	0.37	0.84	0.88
Video 6	0:00:12	1.04	0.94	0.91	0.94

Table 2. Experimental results for provided videodataset.

6.3.2. Object detection metrics: The object detection model's performance was evaluated using three metrics: mean average precision (mAP), accuracy, and prediction time. The model achieved a mAP of 89.5% and an accuracy of 93.28%. In terms of prediction time, it takes an average of 4.9 milliseconds for the model to process an image and generate predictions on a core i7-1165g7 with 16gb ram with an RTX2070 gpu. Fig. 11 shows a graphical representation of this measurement. When the model was deployed on an RPI 4 cluster the model achieved a prediction time average of 129.6 milliseconds.

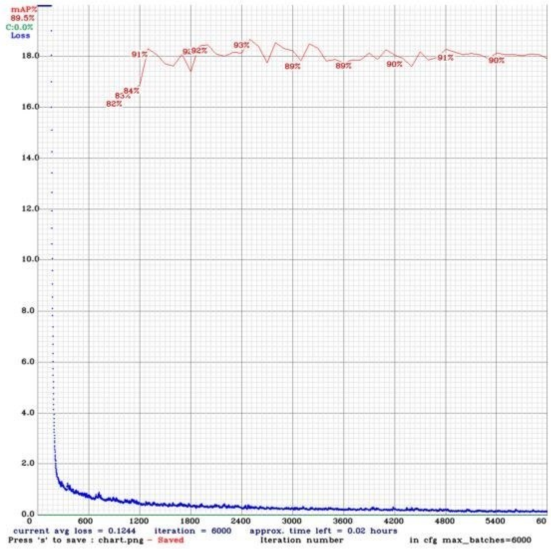


Figure 11. mAP and loss graph for object detection model.

7. CONCLUSION

The proposed model works on jointly stitching and stabilizing the live stream from two or more quadcopters. The estimation of inter motion between the live feeds from the cameras and intra motion between the frames of the same video. The entire process is turned into an optimization problem to get the best fit stitching and stabilized video, so the intra motion method assures the temporal smoothness sustainability between the different frames of the same video, and the inter motion method assures the forcing of the spatial alignment between the multiple videos provided by the drones. Handle scenes with parallax, each video frame is divided into smaller cells so that it is easier to use the bundled-path methodology. Additionally, the model can be extended to a greater number of quadcopters by utilizing Webots simulator for algorithm testing purposes.

The suggested model displayed exceptional object detection performance, achieving high average precision measures across multiple recall levels. Moreover, it exhibited consistent and robust performance in diverse experimental settings, with relatively high stability and stitching scores which indicates how the model is effective.

The model addresses the limitations of video streaming a single drone. While the drone swarm system has been rapidly developing, the current approach significantly hinders the interaction, resource sharing, and ability of the drone swarm to perform complex tasks such as real time video stitching and storing much more data. To overcome these limitations, a new cloud-based drone swarm platform architecture is introduced in this paper. The platform enables real time processing of drone swarms' data andeliminates restrictions on the state of the man-machine interaction by allowing connection to the cloud platform at any time. Additionally, the architecture provides cloud storage and cloud computing support for drones, improving their ability to perform complex tasks.

The discussion covers the implementation of a fog-based system for real-time video stitching using DJI Mavic Pro drones on Webots. The system utilized a combination of programming languages and software tools for capturing and streaming video, video stitching, and communication between the drones and the fog node. The drones were configured to stream video to the local server acting as a fog node, which processed the video using the video stitching algorithm and published the stitched output. The system provides several advantages such as real-time processing, low-latency and efficient data transfer, and scalability. Future research can focus on optimizing the video stitching algorithm, integration of additional sensors, and testing the system in real-world scenarios.

REFERENCES

Abdelkader M, S. M. (2014). Optimal multi-agent path planning for fast inverse modeling in UAV-based flood sensing applications. . international conference on unmanned aircraft systems (ICUAS).

Andrew, M. S. (2017). Drone swarms—A monograph by school of advanced military studies.

C. Buehler, M. B. (2001). Non-metric image-based rendering for video stabilization. IEEE CVPR, 2.

Chung A, P. A. (2018). A survey on aerial swarm robotics. IEEE Trans Robot.

Dadmehr Rahbari, M. M. (2021). Edge-to-Fog Collaborative Computing in a Swarm of Drones.

El Shehaby, M. &. (2019). An Efficient Multi-View Panoramic Imaging and Extra Compression of Surveillance Cameras' Footage Using Stitching.

F. Perazzi et al. (2015). "Panoramic video from unstructured camera arrays,".

Flores, J. J. (2019). Let's Democratize Drones! Using the Ryze Tello Drone as a Tool for Ecological Farm Design & Landscape Ecology Research.

Gu, W. J. (2015). "Video stitching with spatial-temporal content preserving warping," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW).

Guo, H. &. (2016). Joint Video Stitching and Stabilization From Moving Cameras.

H. Jiang, H. L. (2012). 3D Reconstruction of Dynamic Scenes with Multiple Handheld Cameras.

Heng Guo, S. L. (2016). Joint Video Stitching and Stabilization From Moving Cameras. IEEE Trans Image Process. .

J. Li, W. X. (2015). "Efficient Video.

Lin, K. &.-F. (2016). Seamless Video Stitching from Hand-held Camera Inputs.

Ritz R, D. R. (2013). Carrying a flexible payload with multiple flying vehicles. IEEE/RSJ international conference on intelligent robots and systems.

S. Liu, L. Y. (2013). Bundled camera paths for video stabilization. ACM Trans, 32, 78.

Saska M, V. V. (2016). Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. J Intell Robot Syst.

Tahir, A., Böling, J., & Haghbayan, M.-H. e. (2019). Swarms of unmanned aerial vehicles — A survey. In: Journal of Industrial Information Integration.

Wei LYU, Z. Z. (2019). A survey on image and video stitching.

Wei, L. &. (2019). A survey on image and video stitching.

Xi, T. J. (2020). Review of unmanned aerial vehicle swarm communication architectures and routing protocols. Appl Sci.

Y. Wang, J. W.-F. (2015). "Camswarm: Instantaneous smartphone camera arrays for collaborative photography.".