Vol 6, Issue 4, April 2026 www.ijesti.com E-ISSN: 2582-9734
International Journal of Engineering, Science, Technology and Innovation (IJESTI)

AI-Powered Real-Time Video Surveillance for Disaster Detection and Response

Dalu Vijay Praneeth

Division of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Tamil Nadu, India.

Email: daluvijaypraneeth@karunya.edu.in

Dr. E. Bijolin Edwin

Division of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Tamil Nadu, India.

Email: bijolin@karunya.edu

ABSTRACT

Loss of life and property due to natural disasters (floods, wildfires, earthquakes, etc.) is an all too common occurrence each year. The speed and effectiveness of any disaster response operation relies heavily on two factors: 1) quality of situational awareness; and 2) time it takes to gather that situational awareness. Many traditional methods of monitoring disasters utilize manual inspections of satellite images, resulting in substantial delays and limited spatial coverage. In this paper, we introduce a real-time video intelligence framework that leverages deep learning object detection, semantic segmentation, and video stream analysis to provide actionable situational awareness during natural disaster events. This framework utilizes live aerial video from aerial drones (UAVs) as well as from fixed surveillance infrastructure to detect and classify key disaster indicators such as extent of floodwaters, extent of fire spread, structural collapses, blocked roads, and presence of survivors. Our multi-modal pipeline includes YOLOv8 for real-time object detection [1], DeepLabV3+ for semantic segmentation [2], and optical flow estimation to analyze motion within dynamic disaster scenes in a continuous, frame-by-frame manner. We validate on publicly available disaster video datasets and demonstrate real-time inference rates >25 frames per second (fps) with an mAP across key disaster detection categories of 87.4%.

Keywords: *Real-Time, Disaster, Detection, Response.*

I. INTRODUCTION

Among the deadliest and most disruptive events throughout history, natural disasters present an enormous threat to humanity today. As reported by the Office of the United Nations Disaster Risk Reduction (UNDRR), natural disasters impacted over two hundred million people and resulted in more than $280 billion in economic losses in just one recent single year [3]. In particular, flooding, wildfires, earthquakes, tropical cyclones, and landslides continue to occur with increasing frequency and intensity due to climate change and increased urbanization, creating mounting pressures on national and local government disaster response services to react faster than ever before and provide resources more efficiently to respond to such disasters.

Disaster response operations are dependent upon situational awareness; therefore, making good decisions when attempting to respond to an emergency requires the decision-making capabilities of individuals at emergency operation centres (EOC). In the chaotic and very dynamic conditions that develop following an incident, decision makers have to quickly assess the amounts of damage sustained by geographic area, the locations where survivors exist, the extent to which the road networks are passable, and efficiently allocate limited rescue and relief resources accordingly. Unfortunately, this decision making process is

negatively affected by the chaotic and fast-paced nature of disaster environments, which are often characterized by fragmented information, potentially destroyed communication infrastructure, and limited access to areas affected by the disaster.

UAVs (Unmanned Aerial Vehicles) have changed the way we view disasters through aerial views of affected areas. When it comes to accessing areas that aren't easily reached by land, UAVs provide a unique advantage because they can fly at low altitudes and collect high-quality images. UAVs generate an incredible amount of data each hour they operate, typically in the range of gigabytes, so examining this data is a sizeable issue. Manually evaluating the data produced by a UAV is not only a time-consuming process for a human operator but also brings with it variability and errors due to operator fatigue. Therefore, there is currently an urgent need for automated systems that can process video from UAVs and still images from fixed cameras in real-time and produce structured datasets for Emergency Operations Centers (EOC) to use during emergency incidents.

The evolution of deep learning, especially over the last decade, has revolutionized visual recognition technologies, providing machines with the ability to accurately perform many visual recognition tasks without the need for humans. A great example of this is the application of convolutional neural networks (CNN), which have displayed an ability to classify, detect, and segment images of objects, with applications to disaster scene analysis. Another example is the recent growth of transformer-based architectures that provide greater levels of context-based visual reasoning compared to previous networks.

On the other hand, many challenges are presented when deploying these types of models in real-time disaster-mapping context. Disaster video streams are subjected to a number of challenging imaging conditions such as: poor lighting conditions (i.e., smoky, dusty) and/or weather (i.e., heavy rain) and motion blur, as well as having unusual and chaotic layout, thus creating an enormous disparity between the conditions typically found in standard computer vision benchmark tests (i.e., clean and well-lit) compared to the conditions under which disaster videos are recorded. In addition, the requirement for real-time processing imposes another level of challenges because the algorithm must operate continuously as live video streams are being captured, which often requires maintaining a very tight latency capability while still maintaining an adequate level of accuracy in detection/identification of objects in the video.

The focus of this paper is to develop an integrated real-time video intelligence framework for disaster response and management that provides an effective solution to the above challenges. The proposed framework is comprised of three main components: (1) a multiple-stage processing pipeline which includes Object Detection, Semantic Segmentation, and Optical Flow Analyses, that will produce a comprehensive per-frame situational awareness output from live disaster video.

Second, it shows how the pipeline could be deployed on edge computing hardware located on UAV-type platforms, with on-board inference rather than requiring continuous network connectivity. Third, it includes a systematic evaluation of the framework using video data from several disasters, with evaluation metrics based on the model's ability to detect various disaster conditions and types of videos. Lastly, it will document an architecture that connects the video intelligence outputs to a GIS dashboard for real-time visualisation by emergency management personnel.

II. RELATED WORK

Computer vision and deep learning have become integral to disaster response and monitoring through new advances in aerial imagery over the last several years as well as developments within deep learning algorithms. In this section, we evaluate the body of prior research that supports the technical areas comprising the framework presented here.

A. Aerial Imagery Analysis with Respect to Disaster Assessment

While early work focused primarily on analyzing post-disaster aerial imagery via static satellite images as opposed to video feeds, there were many studies using convolutional neural networks (CNN) trained on satellite imagery to assess damages at each building after the disaster occurred. The xBD dataset is not only a large-scale dataset of building damage in satellite imagery but, because of its design, also provides researchers with a relative standard by which to measure their work's effectiveness when utilizing satellite imagery for assessing disaster damage. However, when considering real-time disaster response via satellite imagery as the imaging satellites have a revisit time of 12 – 24 hours long and also most satellite sensors have difficulty penetrating through clouds which are typically present during disasters related to weather (e.g., hurricanes or floods), there is no doubt that there are fundamental limitations with using satellite imagery for real-time disaster response.

Over the last few years, there has been more and more recognition of UAV-based (unmanned aerial vehicle) aerial video as a complementary, and in some cases a superior, form of real-time disaster monitoring. As part of this trend, various benchmark datasets have been created solely for the purpose of performing video analysis using UAV-based aerial video for disaster use cases. The AIDER dataset includes annotated aerial video clips of four types of disasters (fire, flood, collapse of a building, and traffic accidents). The RescueNet dataset contains post-disaster aerial images, and includes detailed semantic segmentation annotations across multiple damage classifications. These datasets have allowed for comprehensive evaluations of detection and segmentation models within disaster-specific visual contexts.

B. Real-Time Object Detection

The real-time object detection paradigm is currently dominated by the YOLO family of object detectors due to their fast inference times and accurate detections through a single regression approach to solve the object detection problem by predicting bounding boxes and class probabilities from the image’s entire pixel set in one forward pass [6]. In subsequent versions of the YOLO architecture, there have been improvements that have improved both accuracy and speed of inference through various architectural changes. YOLOv5 provided notable changes in training consistency and training flexibility for deployment while allowing the use of YOLOv8, the most up-to-date major version of the YOLO model, to outperform its predecessors in accuracy through the use of an anchor-free detection head, updated backbone architecture, and enhanced data augmentation methods [1]. As a result of these attributes, YOLOv8 is a good choice for real-time disaster response video analysis, where the frame rate of the inference has to match the video feed frame rate and due to the model's size needs to be small enough to be deployed at the edge.

Transformer-based detection models such as DETR and its variants have also shown strong benchmarking results in object detection [4]. However, these new detection models consume significantly higher amounts of computational resources and require longer to fully converge during the training process relative to YOLO-based models, limiting their ability to be adopted for real-time latency-sensitive applications and making YOLO-models the preferred choice for video intelligence systems deployed at the edge.

C. Semantic Segmentation for Scene Understanding

Semantic segmentation assigns to every pixel of an input image a semantic class label, providing a very detailed understanding of how a scene is composed of multiple objects instead of just providing bounding boxes for object detectors. For example, in disaster situations, the ability to perform semantic

segmentation makes it possible to define the geographic extent of flooding, create boundaries for the areas covered by fire, and identify which roadways or other travel routes are passable and which routes are not. DeepLab V3+ is a well-known encoder-decoder architecture for the problem of segmentation that incorporates atrous convolution and spatial pyramid pooling to incorporate multi-scale contextual information and to achieve strong performances on benchmarked datasets for scene understanding. One advantage of the use of semantic segmentation in analyzing videos of disasters is that it enables the generation of temporally consistent maps that depict the changing condition of the scene using the per-pixel results from applying semantic segmentation to each frame of the video. Generating these types of maps would not be possible using individual image analysis.

D. Video Analysis and Temporal Modeling

Using temporal processing of video instead of treating the video as a series of independent frames allows for the ability to extract motion based features from disaster scenes that can yield important information regarding the disaster's dynamics. Estimating optical flow provides an estimate of the apparent motion of adjacent regions (pixels) of an image between two consecutive frames (images) by creating a dense representation of motion through the scene which is able to detect the following: survivors who are continuously moving, areas of a fire that are continuously spreading, and/or water levels that are continually rising due to flooding. Optical flow algorithms such as Farneback optical flow estimates motion very quickly; however, Farneback optical flow is affected by noise. Optical flow methods based on deep learning (e.g., FlowNet and RAFT) are more accurate than classical algorithms but require significantly more computations. For real-time applications, an efficient approach would be to use lightweight optical flow to provide motion gating that directs more computationally expensive analysis to those areas that show a high level of change in the region of interest from frame-to-frame, thereby allowing for less computation per frame while still being able to detect dynamic events.

III. PROPOSED METHODOLOGY

A. System Overview

The real time video intelligence framework is designed as a sequence of stages to process video data from either a UAV or fixed camera source producing a situation awareness output in real time. The processing pipeline has four main stages: video ingestion and preprocessing, parallel object detection and semantic segmentation, temporal fusion with optical flow, and generation of output for downstream integration. The entire pipeline is designed to be executed in real time according to the 25 frame per second (fps) processing rate requirement imposed on a single nVIDIA Jetson AGX Xavier edge computing module; therefore making it suitable for on-board deployment on the UAV. The system architecture is depicted in Figure 1.

The videos stream is obtained from either an on-board UAV camera using a direct hardware connection or by means of Real Time Streaming Protocol (RTSP) from a remote IP camera. An Adaptive Frame Buffer (AFB) manages the flow of video data stream through ingestion and discarding of frames from the processing pipeline when the pipeline is under heavy loads in order to maintain real time performance. Once a video frame is ingested it is processed through a standard normalization pipeline consisting of: 1) resolution scaling to 640x640 pixels and 2) optional channel normalization; and 3) optional image enhancement due to negative imaging conditions such as smoke and haze, etc. An example of limitation of achieving good contrast in the presence of haze or smoke would be the use of Contrast-Limited Adaptive Histogram Equalization (CLAHE) techniques.

B. Object Detection Module

The object detection module utilizes a YOLOv8-Medium model that has been optimized to identify disaster-specific objects that correspond to seven distinct categories of disasters, including (1) a person (survivor), (2) a vehicle (with four or more wheels), (3) debris, (4) a flooded area, (5) an area of fire, (6) structural collapse, and (7) rescue equipment [1]. The Optimization of this model was done through three different data sets: (1) The AIDER Data Set; (2) The HERIDAL Data Set that contains images of humans in disaster situations; (3) and other image and video data from disasters available to the general public. In addition to the data from these data sets, data augmentation was completed during optimization by adding random horizontal flips of images, adding random scales and colors to the images (i.e., adding mosaic images), and adding mosaic images with a variety of colors to improve Rigor across a variety of different imaging conditions that would be found in Actual Disaster Areas when using a camera to bring to their end users in the Data From Disaster Areas.

The Fine-tuned YOLOv8-Medium Object Detection model can perform inference of each frame to receive object detections in about 18 milliseconds on the Jetson AGX Xavier hardware platform. While allowing 40 milliseconds to provide a total of 25 frames of evidence, the headroom of 22 milliseconds can be used processing of each of the remaining elements of the entire processing pipeline can acceptably “handle” record of object detections, including filtering and selecting the best objects to determine whether or not they will be passed to the next step of processing. Object detection will be separated based on a 0.45 object confidence threshold and processed using Non-Maximum Suppression using a 0.5 IoU threshold. The output of the object detection module will be in the form of a list of bounding boxes, class labels, and object confidence scores for each frame, and subsequently passed to the futures temporal fusion module and future output generation module**.**

C. Semantic Segmentation Module

The semantic segmentation process of the classifier is completed in parallel with the object detection process using a lightweight DeepLabV3+ variant based on a MobileNetV3 network due to the MobileNetV3's superior trade-off between segmentation performance and processing speed compared to other networks [2]. The classifier will categorize each pixel into an appropriate semantic category (e.g., clear ground, flooded ground, fire or smoke, collapsed structure, intact structure, vegetation, water body, and sky), as well as an occupancy count of the pixels with vegetation, water, and sky for that pixel.

The MobileNetV3-DetailedLabV3+ Combination-based Edge Device classifier can complete object identification and classify pixels in your frames within an average 22 milliseconds per frame (i.e., at the edge device). Additionally, the pixel-level flood and fire class labels can be referenced, respectively, to add complementary data to the object detection output for object identification and tracking through the use of detailed flood and fire class masks; as well as an occupancy count for each of the flood and fire class labels for all pixels.

The georeferencing of the occupancy count is completed by using the UAV telemetry data stream (which provides GPS position, altitude, and orientation) at each frame timestamp when processing and forwarding frames to the GIS dashboard for occupancy (i.e., the update of each occupancy map). The occupancy map and occupancy count will continue to update as new frames are processed and sent to the GIS dashboard.

D. Temporal Fusion with Optical Flow

A lightweight Farneback optical flow module calculates dense motion fields between consecutive pairs of frames using a sensor that has been downsampled to a resolution of 320x320 pixels [7]. The motion field is used in two ways: first, the magnitude of the flow vectors is used to identify areas of dynamic

activity (i.e., where there is a lot of movement), allowing high-priority processing and alerts for those areas in future frames; second, the flow vectors between frames are compared against the identified objects in the area in order to determine if they are individuals who are actively moving or not (e.g., potentially unconscious or trapped).

In addition to this, the temporal fusion stage also includes a tracklet-based tracking module that uses a Simple Online and Real-time Tracking (SORT) algorithm to ensure that consistently identifiable objects within frames can be identified over time [8]. The tracklet continuity assists the estimate of directional trajectories of moving survivors, vehicles and some other objects. As a result, emergency coordinators receive information about both where survivors currently exist as well as the direction and speed of their travel, which is useful for coordinating potential rescues.

E. GIS Integration and Alert Generation

The results from the detection, segmentation and tracking modules, which include timestamps, UAV telemetry geospatial coordinates, detection class labels and confidence values, segmentation class area statistics, and active tracklet states, are compiled into a frame event record and sent using a lightweight MQTT messaging protocol to a central processing server that maintains an updated live GIS layer with updates occurring every second. At the central command center, emergency management personnel can use a web-based dashboard to visualize the current state of a disaster scene as seen from one vantage point with detection and segmentation outputs overlaid on an image of a geographically referenced base map; additional visual cues provide users with information about the level of severity coded by color and any high-priority events coded with alert triggers (for example, survivor detected and/or rapid fire spread).

Thresholds for automated alerts are set by the operations center during the mission start, however these thresholds are also adjusted dynamically within the parameters set by the operations center as necessary during the course of a mission. Alerts will be created when a survivor is identified with confidence greater than 0.7 for three or more consecutive frames, the area of flood spread increases greater than 10% within a 30-second interval, or when a structural collapse event is detected. Alerts will be delivered via SMS and push notifications to first responders' mobile devices in the field to achieve rapid deployment without requiring responders to leave their stage.

IV. DATASET AND EXPERIMENTAL SETUP

A. Datasets

The object detection system is tested against the AIDER dataset, which has 8588 annotated aerial images, with 4 classes of disasters. This dataset is subdivided into training, validation and test datasets, with sizes of 70%/15% and 15%, respectively. Additional images are sampled from the HERIDAL dataset (1068 annotated images of humans that have survived a disaster or need rescue from mountains) to add to the variety of training data, especially for the survivor and vehicle detection categories.

The RescueNet dataset is used to evaluate the semantic segmentation of objects, consisting of 4494 high-resolution images taken after disasters and contain pixel-level annotations across 10 categories, but only the 8 most relevant categories to the system are evaluated.

A video benchmark of 12 publicly available disaster video clips that are collected from news and emergency services archives is constructed for evaluating the system latency and throughput. There are disaster video clips for floods, wildfires, building collapses, and earthquakes. Each clip has a length of between 2 and 8 minutes. A team of human annotators creates ground truth annotations for a subset of the video clips using a semi-automated annotation process, resulting in a total of 3,420 annotated test frames available for quantitative evaluation.

B. Implementation Details

Implementing the models in Python with PyTorch 2.0. For fine-tuning the YOLOv8-m model is done using 100 epochs with an initial learning rate of 0.01, with cosine decay on learning rate (and a batch size of 16, on a workstation that has an NVIDIA(RTX3090) GPU). For the deep lab v3+ segmentation model, the MobileNetV3-Large backbone has been pre-trained with ImageNet weights, and the decoder/classification head has been trained for 80 epochs on a rescueNet train split utilizing the Adam optimizer at a learning rate of 3e-4 with a polynomial learning rate schedule. Edge inference is performed on NVIDIA (RTX Joker) using Jetpack 5.1 (and TensorRT8.5 has been optimized for both models to maximize throughput on edge hardware).

The sort tracking module utilizes the Kalman filter for state prediction and the Hungarian algorithm for assignment, with unmatched tracklets having a max age of 10 frames, and track confirmation utilizing a min hit count of 3. The Farneback optical flow computation has a pyramid scale of 0.5, 3 pyramid levels, a window size of 15 and 3 iterations. GIS integration is done through the QGIS Python API for server-side layer management, and through a React based web front-end for the OPS center dashboard.

C. Evaluation Metrics

The performance of object detection will be assessed using the traditional measure of mean average precision @ IoU 0.5 (mAP@0.5) and with IoU from 0.5-0.95 (mAP@0.5:0.95), while average precision will be calculated for each detection class (there are 7 detection classes).

The mean Intersection over Union (mIoU) value will be calculated using the 8 different semantic categories in order to determine the performance of semantic segmentation. The system performance will be assessed as frames-per-second throughput and end-to-end latency from frame capture to the GIS dashboard update. Performance of alert generation for survivor detection and flood extent change will be assessed in terms of recall and precision from the annotation video benchmark.

V. RESULTS AND DISCUSSION

A. Object Detection Performance

The refined YOLOv8-m model achieves an overall mean Average Precision (mAP) of 87.4% based on testing using the AIDER benchmark. Compared to the original YOLOv8-m (trained only) on the COCO dataset, this reflects an improvement of 6.2% and a 3.8% improvement over previously reported results using the AIDER benchmark. In terms of average precision for each category: people 84.1%, vehicles 91.3%, debris piles 82.6%, flooded areas 89.7%, fire regions 93.2%, structural collapse 78.4%, and rescue equipment 88.9%. The low performance of the collapsed structural class corresponds to the considerable variance among types (materials) and causes of disasters causing buildings to collapse, as noted in the disaster damage literature [5].

When YOLOv8-m is compared with other architectures tested on the same evaluation dataset, the YOLOv8-m model is appropriate for this application. For instance, the Faster R-CNN ResNet-50 baseline achieved an mAP of 48.3% while the YOLOv8-m model achieved an mAP of 54.7%, demonstrating greater accuracy with an anchor-free detection head; however, the Faster R-CNN model only produced eight frames per second for inference time, while YOLOv8-m produced an effective 55 frames per second on the edge processing hardware, making YOLOv8-m more suitable for real-time implementations than Faster R-CNN models. The smaller version of YOLOv8 which was YOLOv8-nano achieves 42 frames per second, however it is giving you a much lower mAP@0.5 score of 79.1% thus showing that YOLOv8-m is ideally positioned between the accuracy and speed for this use-case.

B. Semantic Segmentation Performance

The DeepLabV3+ model that uses MobileNetV3 as a backbone achieved a total mIoU of 72.8% on the test split from the RescueNet dataset. The sky class provided the highest IoU score at 97.3% and the water body class followed closely at 88.4% since they have high visual dissimilarities. Flooded ground was the most important class from the perspective of disaster response activities with 79.6% IoU. The lowest performance occurred for the fire and smoke class where the IoU score was 61.2%. The lower level can be attributed to the inability to determine accurate boundaries for smoky scenes due to their high visual variation and also large overlap with other classes such as hazy or cloudy regions in some aerial disaster scenes. The class of collapsed structures had an IoU of 68.9% which correlates with the difficulty that was illustrated in the detection results for this class.

Looking at the qualitative comparison of video benchmarks from each of the segmentation output also illustrates how the accumulation of each frame (e.g., segmentation masks) can create a consistent and spatially-accurate persistent flood inundation map over multi-minute sequences. The georeferenced maps of extents of flooding generated from segmentation outputs exhibit a good visual agreement to spatially-derived maps of ground truth obtained from satellite images captured within six hours of when the UAV mission occurred (for three of the flood simulation videos), which illustrates that segmentation outputs can be useful for producing operational flood mapping results.

C. System Throughput and Latency

The complete pipeline, including TensorRT optimization, achieves an average of 28.3 frames per second (fps), far exceeding the target frame rate of 25 fps, when used with 1080p input video on the Jetson AGX Xavier Edge Hardware. The total end-to-end latency for the complete processing from frame capture to GIS dashboard update is 1.34 seconds, 36 milliseconds of which are spent in processing on the Jetson. The remaining time consists of latency due to wireless transmission of the results and server-side processing of the GIS Layer. The onboard processing component is well within acceptable limits for real-time processing and has excess capacity for additional processing tasks.

Based on the memory consumption estimations, the onboard TensorRT optimized detection and segmentation model uses 4.2 GB of GPU memory out of the 32 GB available on the Jetson AGX Xavier. The CPU utilization for the preprocessing, optical flow, tracking, and telemetry fusion components averages only 38% across all 8 ARM CPU cores; therefore, there is significant CPU headroom to accommodate additional processing tasks on the Jetson AGX Xavier Edge Hardware.

D. Alert Generation Performance

The survivor detection alarm system attained a recall of 91.2% and a precision of 87.6% at the default confidence and time series consistency threshold for the annotated video benchmark. The false alarm rate of 12.4% is mainly caused by the detection of mannequins and disaster response training dummies found in some of the test clips. There are also instances in which highly motion-blurred frames of rapidly maneuvering UAVs have been incorrectly detected because they do not last for three consecutive frames as required. Flood extent change alerts achieved a recall of 94.7% and a precision of 89.3%. The majority of the false alarms for flood extent change alerts come from noticeable large camera pan movements, resulting in apparent changes to the segmented flood area due to new areas in the scene entering the camera's field of view.

The results demonstrate that the proposed alarm system provides a high degree of sensitivity to the most critical life-safety events while maintaining an acceptable false alarm rate that does not overwhelm the emergency operations center personnel with false notifications. The comparison between the alarm system and the baseline with frame-by-frame detection without temporal consistency filtering shows that by

adding the three-frame consistency filter to the alarms, there is a 34% reduction in the false alarm rate and only 4% reduction in recall; evidence supporting the value of temporal reasoning for real-time disaster videos.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

In this paper, we describe a framework for the use of video to provide intelligence in real-time for disaster management and response. The framework consists of an integrated, unified processing pipeline designed to work on edge hardware mounted to unmanned aerial vehicles (UAVs). This pipeline combines four types of artificial intelligence (AI)-based analysis: object recognition (YOLOv8), semantic segmentation (DeepLabV3+), optical flow analysis (Farneback), and multi-object tracking (SORT). The framework was able to process video in real-time with a frame rate of 28.3 frames per second, had an average latency for GIS updates of 1.34 seconds, had a survivor detection recall of 91.2%, and had a flood mapping mean intersection over union (mIoU) of 79.6% for a disaster specific event. In addition, GIS-based operations center dashboards can be updated continuously with the outputs from the processing pipeline providing emergency management personnel with up to date situational awareness of ongoing disasters without the need for manual review of video footage.

The findings of this work support the assertion that automated video intelligence is ready for practical application in disaster response. This framework provides decision-makers with an efficient means to obtain structured information and respond quickly when making decisions, especially during the initial critical hours of a disaster. This framework is applicable to a large variety of disaster types and can be expanded to included new categories of detected objects and integrated with other types of sensors.

B. Future Work

Multiple avenues exist for future work. Enhancing the system to incorporate multi-spectral, thermal infrared video streams from UAVs using a multi-sensor payload would greatly enhance detection performance for locating survivors in smoke-filled environments where visible spectrum cameras inherently struggle. By providing the ability to identify the heat signature left behind by survivors in the smoke-filled and/or low-light environments using a thermal infrared camera, the current system's capability to locate survivors using visible spectrum may be significantly increased.

The second option is to replace the current processing structure of one UAV with a cooperative framework consisting of three or more UAVs. This will help coordinate the coverage of large disaster areas. An architecture where each UAV does its processing on-board and passes compressed detection and segmentation information to ground stations will drastically reduce the need for bandwidth when operating multiple UAVs and will allow for significantly wider coverage of disaster areas where the potential for casualties will be high.

The third benefit would be integrating natural language generation capabilities to automatically generate a standard generated report for each active situation report generated from processed video intelligence data to further ease the cognitive burden of EOC (emergency operations center) staff members. The ability of large language models to summarize structured data has proven effective; therefore, when connected to geospatial event records generated from the video intelligence pipeline, these same models could be used to automate the generation of standardized incident reports at pre-determined frequencies during a disaster response operation. Finally, conducting a prospective field evaluation of the overall framework in cooperation with a regional emergency management organization would provide important, real-world, operational validation of the usefulness of the system and assist in identifying operational performance issues not addressed through laboratory experiments or analysis of available datasets.

REFERENCES

1. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub repository, 2023. Available: https://github.com/ultralytics/ultralytics.
2. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proc. European Conference on Computer Vision (ECCV), pp. 801–818, 2018.
3. United Nations Office for Disaster Risk Reduction (UNDRR), "Global Assessment Report on Disaster Risk Reduction," Geneva, Switzerland, 2022.
4. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in Proc. European Conference on Computer Vision (ECCV), pp. 213–229, 2020.
5. R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xBD: A dataset for assessing building damage from satellite imagery," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.
7. G. Farneback, "Two-frame motion estimation based on polynomial expansion," in Proc. Scandinavian Conference on Image Analysis (SCIA), pp. 363–370, 2003.
8. A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in Proc. IEEE International Conference on Image Processing (ICIP), pp. 3464–3468, 2016.