



HAL
open science

Autonomous UAV Edge Architecture for Road Hazards Extended Reality Warnings

Mehdi Salim Benhelal, Xinchun Han, Hossam Afifi, Badii Jouaber, Hassine Moun gla, Adel Mounir Said

► **To cite this version:**

Mehdi Salim Benhelal, Xinchun Han, Hossam Afifi, Badii Jouaber, Hassine Moun gla, et al.. Autonomous UAV Edge Architecture for Road Hazards Extended Reality Warnings. 2022. hal-04894234

HAL Id: hal-04894234

<https://hal.science/hal-04894234v1>

Preprint submitted on 27 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autonomous UAV Edge Architecture for Road Hazards Extended Reality Warnings

M. Benhelal*, X. Han*, H. Afifi*, B. Jouaber*, H. Moun gla*, Adel M. Said†

*Samovar, Télécom SudParis, Institut Polytechnique de Paris, France

†Switching Department, National Telecommunication Institute (NTI), Cairo, Egypt

Abstract—Unmanned Aerial Vehicles (UAV) can be exploited to collect important information about road hazards to be broadcast to people in the vicinity and steer clear of the event area through extended reality. UAVs can also offload large aggregate driving information such as lidar or cameras locally from vehicles without saturating the cellular network. In this work, an edge server-based architecture is introduced to provide fast response to detect potential anomalies. Hence, UAVs are used in tracking local road hazards by collecting information about these hazards and offloading this data to an edge server. Reinforcement learning (RL) is used to automatically deploy the UAV/drone at the spot where possible anomalies are detected. An ideal model was derived for the number of drones required. Drone movement is learned using various deep RL techniques. Experimental results show a very encouraging autonomic deployment of drones.

Index Terms—autonomous driving, UAV offloading, Road Hazard Detection, Machine Learning, extended Reality.

I. INTRODUCTION

IN the world of Internet of Things (IoT), huge information is collected from sensors that can be used in myriad applications [1]. The information collected is in terms of sensor measurements, images, videos, people behaviour, etc. This can be used to detect/predict unplanned events which are called anomalies. An anomaly is a potential event that is considered a black box unless an information is conducted about it.

In one hand, the technology of developing unmanned aerial vehicles (UAV) is increasing with great strides which has attracted its commercial use. The different hardware and software equipping the UAVs makes a difference in their capabilities. Thus, drones with sensors, camera and lidar can be efficiently exploited to capture information about a particular event from all sides.

On the other hand, extended reality has many benefits to get the most out of their virtual reality. It can be used to provide a more realistic view of any event and helps in extracting and accessing remote information without barriers from each issue. This information can be exploited to construct an extended reality for any event in a safe way without physical intervention. Hence, help in classifying the event and know its extent.

Based on the above, we propose an augmented reality system controller hosted in edges to speed up the responses

in case of anomalies which will be described later in the next section. Based on the prediction or the classification of the data collected from the roads sensors, the controller will send a UAV to explore the details and extent of the potential event by sending an autonomous UAV to that location. Detecting/predicting anomalies is a step in the process of dealing with them. The sent UAV will then capture the required data and offload it to the system controller which will be used to built an extended reality to the anomaly to take the correct action towards it and broadcast it to whomever it may concern to minimize its impact.

The contributions of this work are summarized as follows:

- An edge controller detects/predicts anomalies from information collected by IoT devices distributed along roads.
- Use autonomous UAV equipped with (sensors, camera, lidar) to collect and offload detailed information, photos and videos about the event.
- Use detailed UAV's offloaded data to build an extended reality of the event for taking a better action.
- Edge controller announces interested people who will be affected by the event to be away from this area.

There are many previous works that have addressed anomaly detection and suggest different approaches but few considers response action accordingly.

In the work of [2], the authors proposed the use of hyperspectral images in the detection of anomalies. They built two dictionaries, one for background pixels and one for possible anomalies. Each collected image is then analyzed for background, noise, and anomalies layers. Later, they are compared to built dictionaries to discover if there are possible anomalies.

[3] is another work that also uses hyperspectral images to detect anomalies. It is suggested to use Fractional Fourier Transfer (FrFT) to filter noise and improve detection accuracy. In addition, they proposed to assign background pixel weights using spectral and spatial information.

Also, in [4], the authors proposed to detect possible anomalies by extracting the band image from the background which is enhanced by using an image filter to reduce background noise and highlight potential anomalies.

In [5], an anomaly detection algorithm is proposed based on network diffusion that uses transition probability to detect sequences of anomalies from a system monitoring graph that captures the routine behaviours of each entity.

The work of [6] proposed a multistep hyperspectral anomalies detection process. First, an anomaly is detected by dif-

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

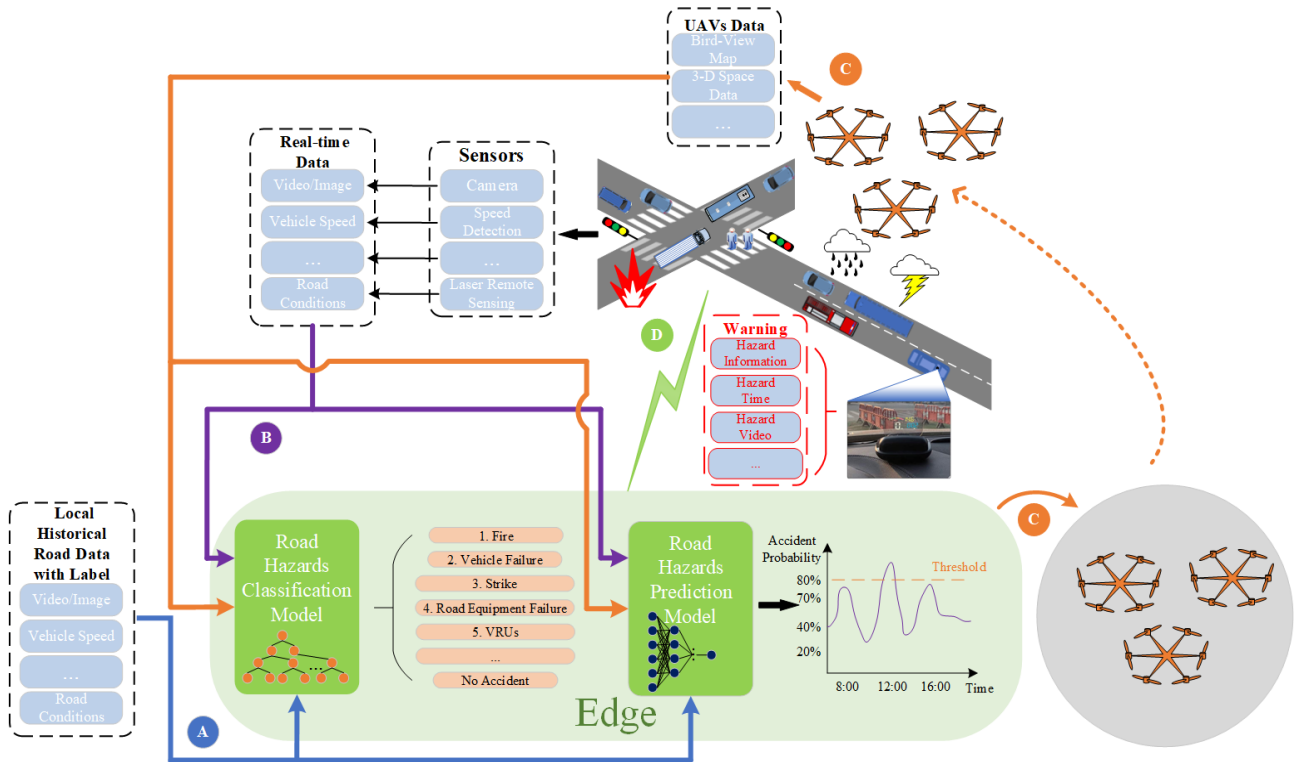


Fig. 1. Augmented reality hazard restitution architecture with UAVs-Assisted Edge.

ferentiating between the event and the background pixels. Second, this result is compared with the results of spatial anomalies to get final detection results with better performance. Also, they proposed using FrFT for better detection results.

Moreover, in [7], the authors proposed an edge- or cloud-centric model to detect and control anomalies in the production lines of CNC and PLC machines. This is to discover possible causes of misbehaviour in production lines. Machine learning (ML) is used as the detector in the collected data.

Previous literature only provides anomalies detection models without proposing a procedure to respond to these detected events. The work of [8] proposes a model for detecting anomalous time series of operational network traffic by collecting tensor streams. These tensors are aggregated and at discrete time and use an unsupervised learning model for detection. In addition, the authors suggested providing an intelligent procedure based on the anomalies detected.

II. SYSTEM ARCHITECTURE

Road hazards, such as fire, vehicle failure, strike, road equipment failure, and Vulnerable Road Users (VRUs), etc, have already posed a significant threat to traffic efficiency, life security and property security. With the development of edge servers and UAVs, they are seen as a promising way to classify and predict road hazards. Therefore, this study aims to present an edge-based architecture and the use of UAVs for collecting and offloading data to assist edge server computing. Figure 7 presents the UAVs-assisted edge-based architecture.

A. Solution Components

Our solution is designed around an edge server. The main architecture components include a group of different sensor types, automated drones and a set of servers installed at the network edge. Databases managing road data characteristics (Openstreet Maps), static sensor information (type, position...) and dynamic sensor data (videos, Lidar, anomalies,...) are regularly updated. The system controller polls the databases and predicts/classifies the possible anomaly. Finally, a broadcast server announces the detected information (from dynamic database) about the road hazard. We describe hereafter each module of the system architecture.

We assume that basic road sensors communicate with the edge through legacy cellular networks. When drones are sent to the road hazard, they communicate with the edge using high speed Wifi or cellular networks (this depends on the amount of data sent from the drone to the edge). The broadcast of potential anomalies to people of interest is done through classical online navigation applications. Some information is announced on electronic road panels.

B. Building and Pre-training Models.

Road hazards consist of several categories as they affect traffic efficiency differently. Moreover, due to the boom in vehicle usage and rapid development of intelligent transportation systems, vast amounts of traffic data are available, which makes road hazard classification and prediction possible. Based on the powerful computing of edge servers, a Road Hazards Classification Model (RHCM) and a Road Hazards Prediction Model (RHPM) are designed. Firstly, the RHCM can classify

road hazards based on input data collected from the road sensors such as collected images/videos, vehicle speed, and road conditions. Secondly, the RHPM can predict road hazards to broadcast warnings to interested people affected by the event. However, both models need to be pre-trained before being deployed, which requires extensive local historical road data and manual labelling of them.

C. Deploying RHCM and RHPM.

Many sensors existed in modern urban roads, for example, video streams or image data can be obtained from camera sensors, speed data can be obtained from speed detection sensors, road information data can be obtained from laser remote sensing sensors, etc. The RHCM can classify whether a hazard has occurred and what kind of hazard has occurred, such as fire, vehicle failure, etc. The RHPM can predict the probability of a road hazard occurring in the future time. Unlike pre-training, the data input to both models after deployment is in real-time. In addition, both models can be optimised while working. To observe the performance of the RHCM, metrics such as accuracy, precision and recall^{[9][10]} can be adopted. Regarding the RHPM it can be estimated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and goodness of fit (R^2).^[11]

D. UAVs Assistance

Although many metrics exist to assess the performance of both models, a UAVs-assisted approach is still required to increase the models' credibility and robustness. There are two conditions for triggering UAVs to track road hazards: 1) the RHCM classifies the current scenario as a hazard scenario; 2) the RHDM predicts that the probability of an accident is greater than the Threshold(a custom parameter), i.e. $P_{acc} > Threshold$. Either of the two conditions will trigger the edge server to command the UAVs to identify road hazards.

Reinforcement Learning (RL) as a feedback learning method has been widely used in robotics and UAVs, therefore this study takes RL to control UAVs to track local road hazards. Once the UAVs arrive at the destination, they will collect e.g. bird-view map, 3-D spatial data, etc. and upload them to the edge server. After the edge server has received the data, RHCM and RHPM will optimise their output and finally verify whether there are road hazards or future road hazards.

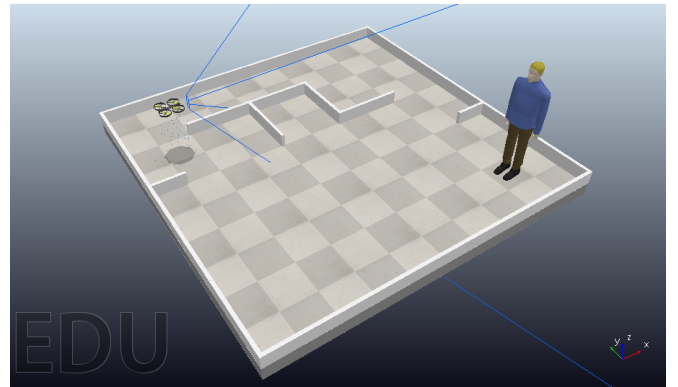
E. Broadcasting to Local Users

When the models have identified local road hazards, the edge server will broadcast road hazard warnings to the local users, including hazard information, future collision time and so on, to advise the users to choose another road, thus optimising traffic efficiency, avoid secondary road hazards and safeguard personal safety.

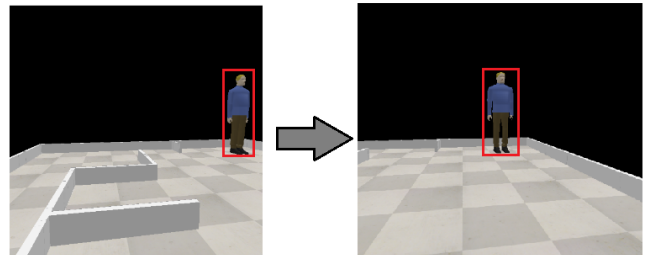
Inevitably, data security and privacy issues are encountered during the uploading and broadcasting process.

III. AUTONOMOUS UAV POSITIONING

In this section we propose to design our custom RL environment as illustrated in Fig. 8a with the help of CoppeliaSim simulator [12] that will involve flying a drone equipped with a visual sensor to track person in the same way road hazard objects can be tracked by UAVs to gather more data and information through the different sensors equipped on them, in this scenario, the UAV's reward is built in such a way that it increases when the tracked object is in the middle of the camera as shown in Fig. 8b. To do this, we utilize the euclidean distance to be able to minimize the distance between the center of the camera and the the object, model free RL that takes the desired actions to achieve our objective with low latency and a light weight, real time object detection algorithm, the agent gets a good reward when able to track the object and center it, we ran our simulation on a Windows 10 machine with a 10th Generation Intel Core i7 CPU, 32 GB of RAM and an NVIDIA GeForce GTX 1650 Ti.



(a) Custom environment



(b) Objective

Fig. 2. UAV positioning environment

A. Software architecture

In a RL scenario there is an agent as well as an environment. The agent selects an action and is rewarded by the environment in addition to transitioning into a new state. This learning process will continue until the objective or another condition are fulfilled, In our case the agent is a UAV with the main goal of tracking the person in our custom environment, the episode ends if the target is either too far away from the camera or if there is a collision with another object or obstacle, in the other hand the agent's reward increases when the object is within the camera's central point, in order to accomplish this

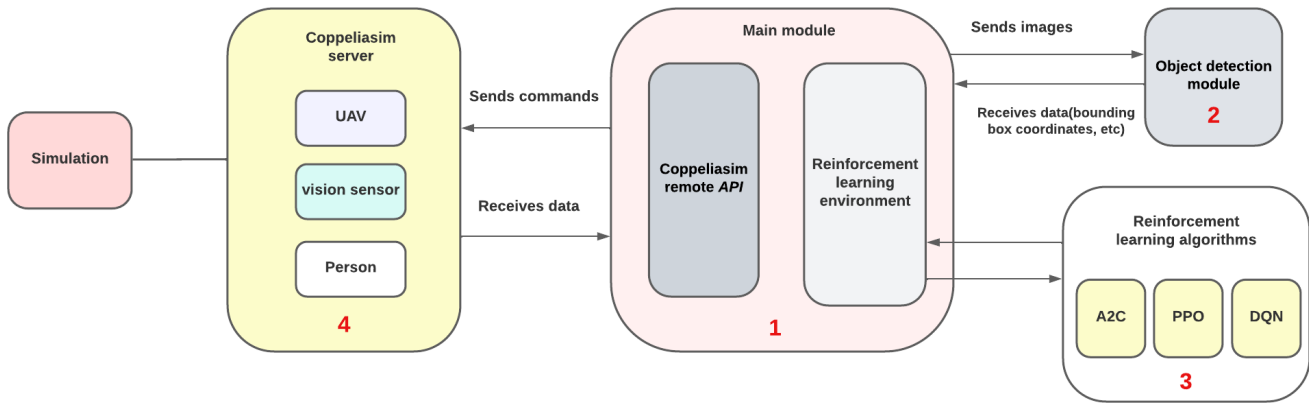


Fig. 3. Software architecture.

goal we designed the software architecture of Fig. 9 which is composed of the following main parts:

1) *Main module*: This module is considered as the core of the architecture because it contains the necessary functions for training and comparing the results of our RL algorithms, it is also responsible of sending commands to the server and retrieving images and other valuable information about the objects while in the same time transmitting the raw images to the object detection module to extract the bounding boxes coordinates of the selected object and other data for the training.

2) *Object detection module*: which involves detecting instances of objects of a specific class in an image and returning the bounding boxes coordinates that will be used for the training of our RL algorithms. In order for the drone to track the person, getting the bounding boxes of the object position is crucial, so using a light weight real-time Object detection algorithm is necessary, YOLOv7 is the state of the art of the object detection models and has proven that it surpasses all known object detectors in terms of accuracy and time needed for the detection of objects as shown in [13], therefore it is the algorithm used in our work.

3) *Reinforcement Learning Algorithms*: are divided into two big families Model-based methods that attempts to plan by utilizing some environmental information, Planning using a model can boost the algorithm's sample efficiency and Model-free methods that focuses primarily on learning from the environment and require no prior knowledge they can be applied to any RL problem, our application requires low latency, so we selected RL algorithms that do not require a model of the environment to make predictions or decisions, due to their fast selection of an action and ability to process the current state of the environment, RL algorithms are able to make decisions in real-time. Model-free algorithms that were used in our work are: value based methods that picks the best action to do in the present state, policy based methods where a stochastic policy function that links state to action is directly learned and finally the actor critic methods a combination of the two cited before, the algorithms used in our work are presented as follow:

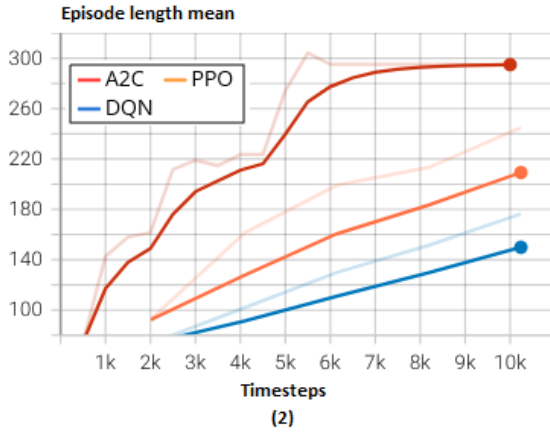
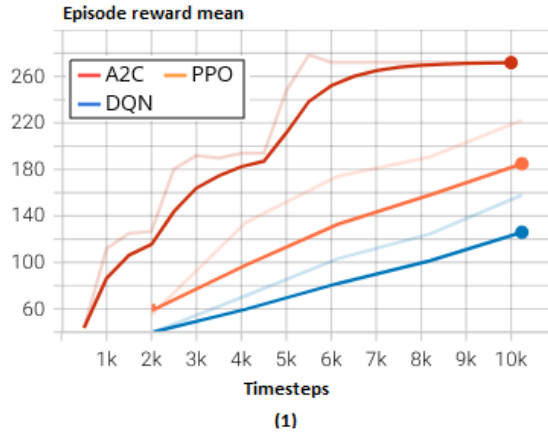
- Proximal Policy Optimization (PPO) is a policy gradient method that can be used for environments with either discrete or continuous action spaces, it uses an advantage function $A(s,a)$ to determine its policy The advantage function basically tells the algorithm how much better was the action that he took based on the expectation of what would happen then what really happened.
- Deep Q-Learning (DQN) where state-value function is estimated in a Q-Learning structure with a neural network that utilizes a Loss function instead of an equation, in addition to the Predicted Q Value and a Target Q Value.
- Advantage Actor Critic (A2C) combines two types of RL algorithms, Policy Based and Value Based, together. Policy Based agents directly learn a policy (a probability distribution of actions) mapping input states to output actions, Value Based algorithms learn to select actions based on the predicted value of the input state or action, A2C only uses a single neural network to estimate both the value function for the current policy and the optimal policy this also enables the algorithm to have a low latency.

4) *Coppeliasim server module*: The Coppeliasim server contains the main functionalities of the different objects(flying, walking, etc.) and also all the functions allowing it to receive remote commands and sending valuable information about the environment like images or positions.

B. RL Results

We created our custom RL environment with the objective of tracking an object (a person in this case) thus the agent gets a good reward when this object is in the center of the camera, to compare the different model-free RL algorithms the main metric used is the mean episode reward which is the average reward that the RL model receives over the course of an episode, a higher mean episode reward indicates that the RL model is performing well, earning a high reward and has been able to successfully track the person and keep it in the center of the camera for a significant portion of the episode. The second metric is the Episode length mean, a bigger Episode length mean is a sign that the model is learning and making

progress because an episode ends if either the person is too far from the camera or a collision occurs. A2C outperformed the other algorithms in terms of episode reward mean and Episode length mean for 10000 time steps as shown in Fig. 10 and took *1hour11minutes20seconds* to do it, PPO also gives good results but in our case comes with the withdraw of taking *1hour33minutes14seconds* to train which is the longest time compared to others, DQN got the worst episode reward mean but took only *1hour1minutes23seconds* which is the least time to train and it could be a considerable solution in case of technical limitations, PPO outperformed DQN because an optimal policy might be easier to represent than a value function, as it only needs to learn what action to take, while a value function has to further express how good an action is, A2C combines both value and policy based approaches for better results.



Algorithm	Training time for 10k timesteps
DQN	<i>1hour1minutes23seconds</i>
PPO	<i>1hour33minutes14seconds</i>
A2C	<i>1hour11minutes20seconds</i>

Fig. 4. RL algorithms results

C. UAV Results

UAV are controlled with an optimized integer linear programming algorithm. The algorithm is responsible for calcu-

lating the needed drones, the actions that each drone should do (send to location, return, backhaul information, charge drone). In this subsection, we present the results of the UAV number needed for each type of the collected data for the potential anomaly. We consider the UAV has the capability to be equipped by different types of sensors as lidar, camera for video streaming and capturing photos, and IoT sensors.

As explained, in the RHCM architecture, the sent drones will send the collected information from the remote location to the edge server. The data is analyzed, re-organized and prepared for consumption in the neighborhood.

We also assume that the UAVs are not equipped with the same sensors types. So we choose suitable UAVs according to the type of anomaly detected during the classification/prediction of the collected road data. Therefore, each UAV will offload a different data type based on its equipped sensors and thus will need a different bandwidth (BW). Figure 11 shows the range of data volume collected for each sensor type used. X-axis is the sensor type, Y-axis is the bitrate (Mb/S) and the color of each line corresponds to a drone with a different sensor type. For the collected Lidar data, its accuracy changes according to the number of collected points, frames, and distance resolution, so the size of the captured data changes as shown in the figure. Also, the size of the captured video data changes according to the used camera resolution and codec. The same concept applies to the volume of data collected for the images and Iot sensors indicated in the figure. Given that the drone is equipped with this set of sensors, so the dump data size is the sum of each sensor's data and its accuracy.

According to the proposed work, the anomaly information will be announced to people in the vicinity of the event area to warn them to mitigate the anomaly area using the electronic road panels around the anomaly location. Moreover, it can be published with online navigation software such as many applications advertise real-time traffic status. This information could be the lidar images collected by the drone that will be displayed on a head-up display as shown in Fig. 12. In addition, with the proposed virtual reality, the transmitted data stream will be updated accordingly.

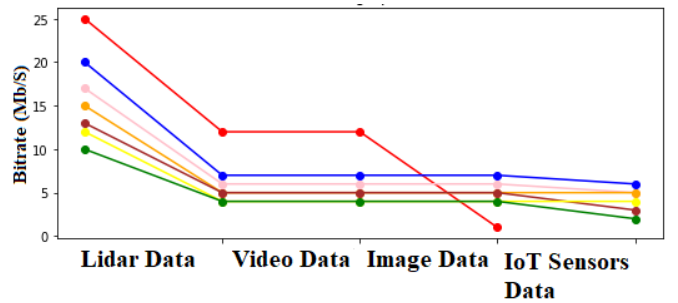


Fig. 5. Drones Traffic Types.

IV. CONCLUSION

An architecture to enable extended reality dealing with road hazards is presented. The architecture is based on early detection of road anomalies using machine learning. UAVs

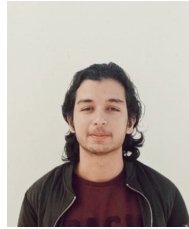


Fig. 6. Head up Display receiving the edge processed information in the hazard location

are trained with different reinforcement learning algorithms to locate the anomaly and offloading the gathered information to the edge server. A simple prototype using commercial drones is trained and used to validate the concept. Then, an optimal integer linear programming algorithm calculates the necessary number of drones, and manages the drone through their journey to and back from the hazard location. Information is processed in the edge server and broadcast to vehicles in the hazard position.

REFERENCES

- [1] A. Jarwan, A. Sabbah, and M. Ibnkahla, "Information-oriented traffic management for energy-efficient and loss-resilient iot systems," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7388–7403, 2022.
- [2] N. Huyan, X. Zhang, H. Zhou, and L. Jiao, "Hyperspectral anomaly detection via background and potential anomaly dictionaries construction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2263–2276, 2019.
- [3] C. Zhao, C. Li, S. Feng, N. Su, and W. Li, "A spectral-spatial anomaly target detection method based on fractional fourier transform and saliency weighted collaborative representation for hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5982–5997, 2020.
- [4] W. Wang, W. Song, Z. Li, B. Zhao, and B. Zhao, "A novel filter-based anomaly detection framework for hyperspectral imagery," *IEEE Access*, vol. 9, pp. 124 033–124 043, 2021.
- [5] B. Dong, Z. Chen, L.-A. Tang, H. Chen, H. Wang, K. Zhang, Y. Lin, and Z. Li, "Anomalous event sequence detection," *IEEE Intelligent Systems*, vol. 36, no. 3, pp. 5–13, 2021.
- [6] Z. Li, Y. Zhang, and J. Zhang, "Hyperspectral anomaly detection for spectral anomaly targets via spatial and spectral constraints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.
- [7] L. Bacchiani, G. De Palma, L. Sciuillo, M. Bravetti, M. Di Felice, M. Gabbriellini, G. Zavattaro, and R. Della Penna, "Low-latency anomaly detection on the edge-cloud continuum for industry 4.0 applications: the seawall case study," *IEEE Internet of Things Magazine*, vol. 5, no. 3, pp. 32–37, 2022.
- [8] M. Shajari, H. Geng, K. Hu, and A. Leon-Garcia, "Tensor-based online network anomaly detection and diagnosis," *IEEE Access*, vol. 10, pp. 85 792–85 817, 2022.
- [9] C. Gutierrez-Osorio and C. Pedraza, "Modern data sources and techniques for analysis and forecast of road accidents: A review," *Journal of traffic and transportation engineering (English edition)*, vol. 7, no. 4, pp. 432–446, 2020.
- [10] T. Bokaba, W. Doorsamy, and B. S. Paul, "Comparative study of machine learning classifiers for modelling road traffic accidents," *Applied Sciences*, vol. 12, no. 2, p. 828, 2022.
- [11] V. Najafi Moghaddam Gilani, S. M. Hosseinian, M. Ghasedi, and M. Nikookar, "Data-driven urban traffic accident analysis and prediction using logit and machine learning-based pattern recognition models," *Mathematical problems in engineering*, vol. 2021, 2021.
- [12] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.
- [13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: <https://arxiv.org/abs/2207.02696>



Benhelal Mehdi Salim Received his master degree in Networking and Computer Systems from Djilali Liabes University, Sidi Bel Abbas, Algeria in 2022 He is currently pursuing his phd degree with Télécom SudParis, Polytechnic Institute of Paris, France. His reseach interest include Autonomous vehicles, UAV positioning.



Hossam Affi received the Ph.D. degree in computer science from Inria Sophia Antipolis in 1992. He is a Professor with Telecom SudParis, SAMOVAR Laboratory, where he works on machine learning for mobile and security protocols. He visited Washington University, St. Louis, as a Postdoctoral Researcher, where he worked on IP switching techniques. He was appointed as an Assistant Professor with ENST Bretagne (Mines Atlantic today), in the field of high speed networking. After his tenure and a sabbatical with Nokia Research Labs, Mountain View, USA, he took the current position with Telecom SudParis since 2000. His current interests cover vehicular and user centric behaviors.



Jouaber, Badii is Full Professor at Telecom SudParis and at the Institut Polytechnique de Paris. Since 2021, he is leading the R3S research team within the SAMOVAR research Lab. Badii is graduated from Sorbonne University in 1998 with a doctoral Degree in Computer Science, He also holds a Master's degree from the Université de Versailles (1994) and an Engineering degree in Computer Science from ENSI (Ecole Nationale Des Sciences de l'Informatique (ENSI, Tunisia, 1993). He obtained the HDR "Habilitation à Diriger les Recherche" from the Université of Saclay (in 2019). After his PhD, Badii worked for Alcatel Research and Innovation (now Nokia Bell Labs) from 1998 to 1999 and for Alcatel Mobile Division (now Nokia) from 1999 to 2000 in optimization and resource management for cellular networks. In 2001, he joined Télécom SudParis, within the Wireless and Multimedia Services department. His research interests include: Wireless Networks, Resource Allocation, Optimization, Wireless Sensor Networks, Mobility Management, Vehicular Networking and Green communications.



Xinchen Han received his M.Eng. degree in control science and engineering from The State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China, in 2022. He is currently pursuing his Ph.D. degree at Institut Polytechnique de Paris Samovar Lab. His main research interests include autonomous driving and reinforcement learning.



Hassine Mouncla (Member, IEEE) received the Engineering degree in computer science from Computer Sciences Institute, the M.S. degree in artificial intelligence and combinatorial optimization for telecommunications from the Sup' Galilee Institute, in 2002, and the Ph.D. degree in computer networks from the University of Sorbonne Paris Nord, France, in 2006. He is an Associate Professor with HDR (Accreditation to Supervise Research), University of Paris. He is Member of Security and Optimization of Communication Systems. He is a Researcher and

currently an associated researcher with Institut Polytechnique de Paris. He was a Researcher with INRIA until 2008, and a Research Fellow with CNRS-LIPN Laboratory, University of Sorbonne Paris Nord until 2007. He is involved in quite a number of challenging research activities revolving around cutting edge technologies in the areas of network softwarization, network slicing, softwarization of service, and content delivery platforms and unmanned aerial vehicles. His current research interests lie in the field of wireless networking, IoT, ICN, edge computing for 5/6G networks.



Adel Mounir Said received his PhD in Electronics and Communication Engineering from Ain Shams University (Egypt) jointly supervised by Pierre and Marie Curie University (UPMC) and Telecom Sud-Paris (France) in 2014. He received the M.Sc. from the Arab Academy for Science, Technology and Maritime Transport in 2007 and B.S. from the Higher Technological Institute in 2001. He received the Egyptian Syndicate of Engineering Award in 2001 for his achievements. Adel Mounir Said is currently an assistant professor and supervisor of

the switching department at the National Telecommunication Institute in Egypt. His research interests include fixed-mobile convergence, machine learning, artificial intelligence, wireless sensor networks, vehicular networks, the Internet of Things, and software-defined networks.

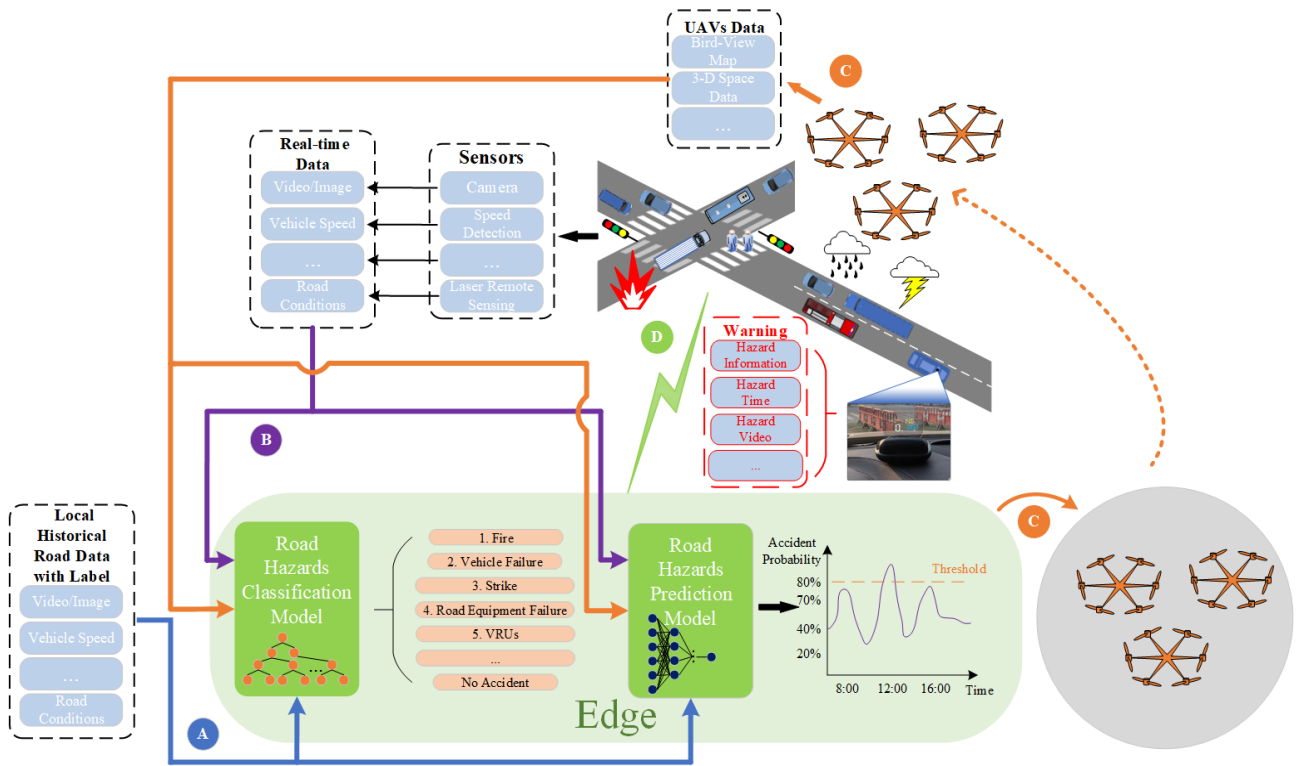
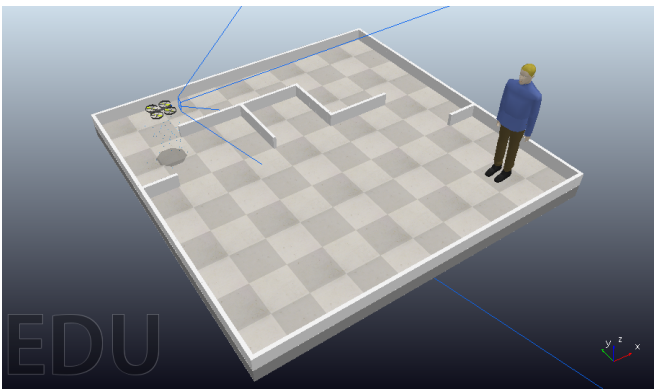
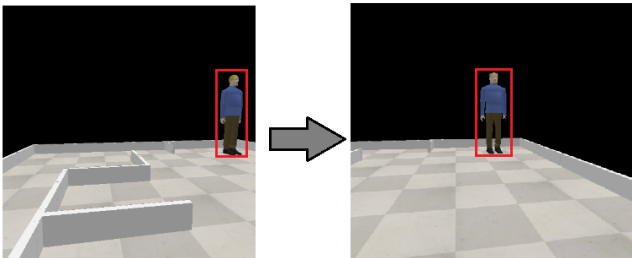


Fig. 7. Augmented reality hazard restitution architecture with UAVs-Assisted Edge.



(a) Custom environment



(b) Objective

Fig. 8. UAV positioning environment

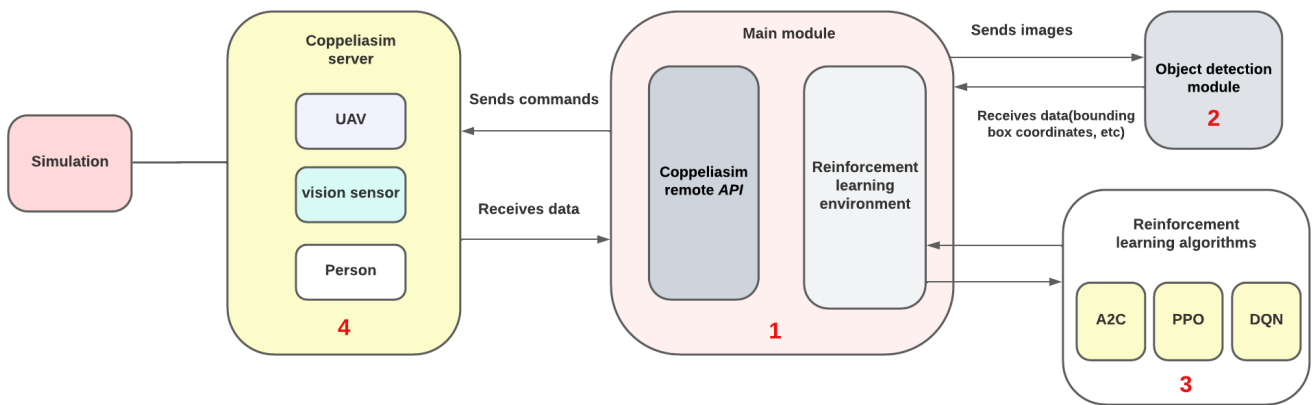
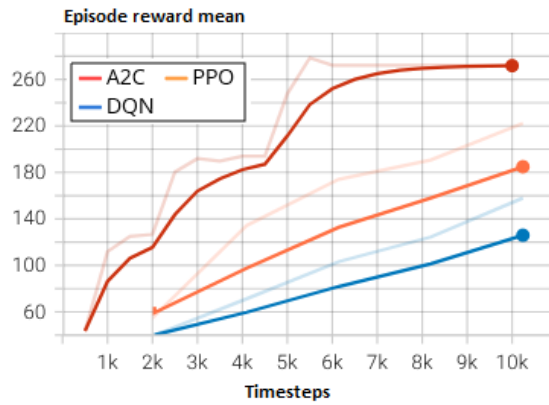
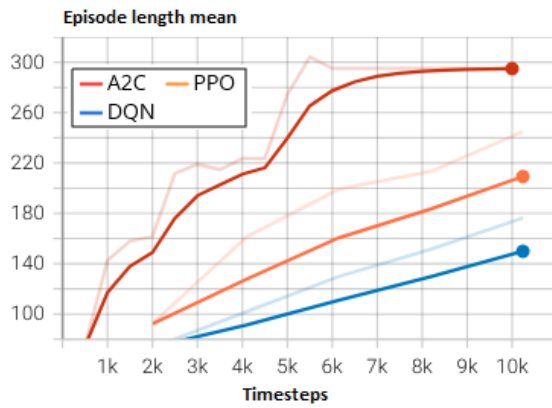


Fig. 9. Software architecture.



(1)



(2)

Algorithm	Training time for 10k timesteps
DQN	1hour1minutes23seconds
PPO	1hour33minutes14seconds
A2C	1hour11minutes20seconds

Fig. 10. RL algorithms results

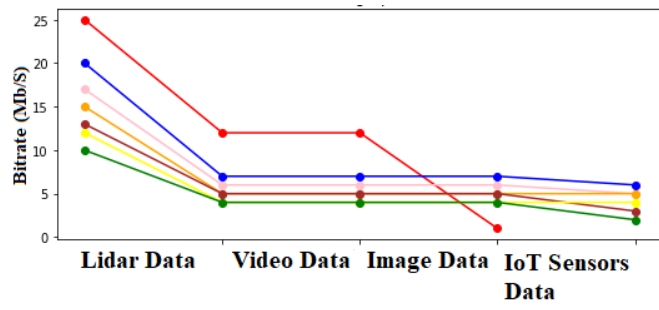


Fig. 11. Drones Traffic Types.



Fig. 12. Head up Display receiving the edge processed information in the hazard location