drones

MDPI

Article

Real-Time Survivor Detection System in SaR Missions Using Robots

Kaushlendra Sharma [1], Rajesh Doriya [1], Saroj Kumar Pandey [2], Ankit Kumar [2], G. R. Sinha [3] and Pankaj Dadheech [4,*]

[1] National Institute of Technology Raipur, Raipur 492010, India
[2] Department of Computer Engineering & Applications, GLA University, Mathura 281406, India
[3] International Institute of Information Technology, Bangalore 500032, India
[4] Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur 302017, India
* Correspondence: pankajdadheech777@gmail.com or pankaj@skit.ac.in

Abstract: This paper considers the issue of the search and rescue operation of humans after natural or man-made disasters. This problem arises after several calamities, such as earthquakes, hurricanes, and explosions. It usually takes hours to locate the survivors in the debris. In most cases, it is dangerous for the rescue workers to visit and explore the whole area by themselves. Hence, there is a need for speeding up the whole process of locating survivors accurately and with less damage to human life. To tackle this challenge, we present a scalable solution. We plan to introduce the usage of robots for the initial exploration of the calamity site. The robots will explore the site and identify the location of human survivors by examining the video feed (with audio) captured by them. They will then stream the detected location of the survivor to a centralized cloud server. It will also monitor the associated air quality of the selected area to determine whether it is safe for rescue workers to enter the region or not. The human detection model for images that we have used has a mAP (mean average precision) of 70.2%. The proposed approach uses a speech detection technique which has an F1 score of 0.9186 and the overall accuracy of the architecture is 95.83%. To improve the detection accuracy, we have combined audio detection and image detection techniques.

Keywords: SaR missions; rescue robot; human detection; F1 score; DARP algorithm; mCPP

check for updates

Citation: Sharma, K.; Doriya, R.; Pandey, S.K.; Kumar, A.; Sinha, G.R.; Dadheech, P. Real-Time Survivor Detection System in SaR Missions Using Robots. *Drones* **2022**, *6*, 219. https://doi.org/10.3390/drones6080219

Academic Editor: Javaan Chahl

Received: 28 July 2022
Accepted: 18 August 2022
Published: 22 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Natural catastrophes such as earthquakes, landslides, and other natural disasters have caused substantial damage to people's lives and property in recent years. Victims are frequently trapped in collapsing structures. As per studies, when there is a natural or man-made disaster such as an earthquake, building destruction happens. Then, after the first half-hour, the survival rate percentage is 91, and after one day, the survival rate percentage decreases from 91 to 81. By the next day (2nd day), it decreases to 36.7%. And again, after the fourth day, it decreases to 19% [1]. As a result, there is a great deal of contemporary interest and demand for the creation and comprehension of modern disaster relief approaches.

In such cases, in order to remediate the situation, a search and rescue operation is carried out. The aim of a search and rescue operation is to provide aid and medical attention to the victim as soon as possible. The person is first of all "searched" for, if his whereabouts are unknown to the rescue team. Once the location of the injured person is determined, the next task is to "rescue" them. The term "rescue operation" refers to a circumstance in which it is known that intervening and organizing a person's rescue is necessary [2]. This usually involves taking them to the medical team.

Manually scouting the disaster-affected area is a tricky task. It needs to be conducted fast enough. For rescue team members, it is very risky for their lives to go inside the disaster-affected areas. All these activities are mainly carried out by humans and trained

canines, and they are typically carried out in conditions that are extremely dangerous and perilous for humans. Due to this, we often witness the loss of human lives of both the survivors and rescue workers. To minimize such losses, a robotic system can be effective in detecting humans. The robot's primary goal is to detect living individuals stuck in the debris as soon as possible after the disaster, stream their location, and determine if the area is safe for rescue personnel [1].

Drones are currently employed for a variety of reasons and are now considered standard in all SaR agencies across the world. However, drone-based SaR operations [3] involving manual monitoring by operators present a number of fundamental obstacles. While controlling the aircraft, the operator must examine real-time images on a tiny screen. Because the searching person is so small in comparison to the rest of the environment, they only take up a few pixels on the screen. Even for those who have been trained to do so, maintaining long-term concentration and attention is difficult. People who are being sought are frequently sheltered by plants, hidden behind a stone, or bonded to the ground, complicating the search even in good weather [2]. These challenges can be mitigated by employing an automated system.

This paper aims to provide a solution to this problem by presenting an automated system which can detect the survivors in real time without manual intervention from a live video feed. Unlike other related works, we used a technique which analyzes both the audio and video feed captured for detecting humans in real time with good accuracy. We also attempt to determine whether the area is safe for rescue personnel or not.

We could have used algorithms such as the Faster RCNN for human detection using the video feed, which work by detecting the possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately. However, instead, we used YOLO (You Only Look Once), which performs all of its predictions with the help of a single fully connected layer, reducing the computation time while maintaining good accuracy for practical use cases [4].

The Internet of Things (IoT) has the potential to make the world a more welcoming place for the current and future generations. IoT devices can be used in a variety of ways to promote long-term development. They can be used to measure the physical frameworks of any object and upload them to a cloud server or any online repository for a real-time analysis and evaluation. This allows us to view and analyze measured data from any location across the world. The Internet of Things (IoT) can also facilitate the decentralization of data analysis and processing. This can be achieved by coupling the same with cloud computing. We can also use the transmitted data to control and instruct equipment remotely. In order to do so, we will be required to establish machine-to-machine communication over the Internet. The air-quality detection system is built with open-source hardware such as the Raspberry Pi and WiFi, and it is both practical and energy efficient. The current sensors collect data from many natural affections and send it to the Raspberry Pi, which is controlled by a central server. The inbuilt WiFi module of the Raspberry Pi is used to send the data to the cloud server. The data at the cloud server can then be improvised and displayed to the operators controlling the recipient's server [5].

2. Literature Survey

The literature survey was conducted in two parts. One part of the solution is to enable the robot to navigate the calamity site and explore it for potential human survivors and potential dangers to the rescue workers. The other part is to enable the robot to detect a human survivor in the video feed in real time. Table 1 compares the existing research works for human detection using a video and/or audio analysis of the video feed. Table 2 compares the existing approaches toward multi-robot path planning, coverage, and the exploration of the calamity site. Table 3 compares various cloud service providers based on the number of features provided for different security modules. Table 4 compares different IoT sensors used by different researchers for the purpose of building a rescue bot.

Table 1. Comparison of similar surveys for human recognition and voice recognition.

	Task Undertaken		Learning Type		Training Used		Simulation via
Reference Paper	**Human Recognition**	**Emotion Recognition**	**CNN**	**RNN**	**Unsupervised Learning**	**Supervised Learning**	**Computer Vision**
Chloé Clavel [6]	-	✓	×	×	✓	×	-
Kannan, Haresh [7]	-	✓	✓	✓	×	×	-
Eric Guizzo, Tillman [8]	-	✓	✓	×	×	×	-
Khalil, Ruhul Amin [9]	-	✓	✓	✓	×	×	-
Lygouras, Eleftherios [10]	✓	-	×	×	×	✓	✓
Bejiga, Mesay Belete [11]	✓	-	✓	×	×	×	×
Llasag, Raúl [12]	✓	-	×	×	×	×	✓

Table 2. Comparison of similar surveys for multi-robot path planning and task allocation.

Reference Paper	**Path Planning**	**Greedy Algorithm**	**Sensor Network**	**Coverage Exploration**	**GNN**	**Robot Task Allocation**
Tolstaya, Ekaterina [13]	✓	✓	×	×	×	✓
Tihanyi, Daniel [14]	✓	✓	×	×	×	✓
Rekleitis, Ioannis [15]	✓	✓	×	✓	×	×
Batalin, Maxim A., and Gaurav S. [16]	×	×	✓	✓	×	×

Table 3. Comparison of cloud service providers based on the amount of features provided for different security modules.

Parameter	**Amazon AWS**	**Google Cloud**	**Microsoft Azure**
IoT Device Security	High	Low	Medium
Security and Compliance Service	Medium	Low	Medium
Backup and Recovery	Medium	None	High
Identity and Access Management	High	High	Medium
Key Management Services	High	High	High
Web Application Firewall	High	High	High

Table 4. Comparison of different IoT sensors used in different approaches.

	Method Used		Sensor Used			
Reference Paper	**Internet of Things**	**Wireless Sensor Networks**	**Gas Sensor**	**Temperature and Humidity Sensor**	**Sound Sensor**	**Rain Sensor**
R. Udaya Bharathi, M. Seshashayee [5]	✓	×	✓	✓	✓	✓
Khaled Bashir Shaban, Abdullah Kadri [17]	×	✓	✓	✓	×	×
Nihal Kularatna, B. H. Sudantha [18]	×	×	✓	×	×	×
Swati Dhingra, Rajasekhara Babu, Amir Gandomi [19]	✓	×	✓	×	×	×
Young Jin Jung, Yang Koo Lee, Dong Gyu Lee [20]	×	✓	✓	✓	×	×

3. Problem Statement

Firestorms, floods, earthquakes, and tsunamis are examples of natural disasters, and biological attacks, nuclear blasts, and terrorist attacks are examples of man-made disasters.

Natural disasters produce emergency circumstances as well as physical and societal chaos. According to the International Federation of Red Cross and Red Crescent Societies, the number of people killed by natural catastrophes has fluctuated considerably over the previous decade, from 14,389 in 2015 to 314,503 in 2010 [1]. Natural disasters affect more than 100 million people each year. The government, businesses, and relief organizations are all attempting to reduce the number of these sad deaths. In this type of emergency condition, for survival, food, water, and medicine is most important for victims. In addition, there must be a source or reliable network to communicate with the victims. For any rescue operation, a disaster recovery network system is most important for victims and rescue team members during the rescue of the victims to communicate with them and obtain the condition of the victims. Searching for and discovering survivors, as well as rescuing them, is sometimes part of a disaster relief operation. Currently, in this approach, rescue team members perform a manual search in a conflict zone which may be very harmful to their life as well. It is also very time-consuming. Most calamities demand a large-scale rescue operation. The majority of such rescue attempts involve human teams that are easily overwhelmed and in desperate need of assistance. Members of rescue teams are sometimes forced to work in hostile and dangerous environments [21].

4. Hypothesis

Over time, robots and artificial intelligence systems have been shown to be extremely reliable in many areas. These one-of-a-kind machines can be used to look for victims as well as gather data that might help improve or optimize search and rescue activities. As a result of the recent advancements in the field of robotics and operations, there has been an increase in the amount of research to explore the possibility of using rescue robots for search and rescue missions [21]. It is desirable for the rescue robots to have the capacity to operate without or with minimum human intervention. Most of the rescue robots that have been designed currently are well-equipped and can operate in adverse situations as well. According to recent research, outfitting robots with sensors improves their effectiveness in search and rescue missions. The robots are designed to explore the affected area, gather the required insights in the scenario, and send the analysis to a base rescue station. The robots will be equipped with WiFi and Bluetooth so that they are able to transmit the data even if they are working in a remote setup. The analysis can further be used by the rescuers to have an initial overview of the potential dangerous spots, the exact location of the survivors, and the shortest risk-free path to reach them [13]. In this paper, we plan to introduce the usage of robots for the initial exploration of the calamity site. The robots will explore the site and identify the location of the human survivors by examining the video feed captured by them and stream the location of the survivors to a centralized cloud server. They will also examine the air quality to determine whether it is safe for rescue personnel to manually visit the calamity site or not.

5. Proposed Solution

The idea is to enable multiple robots to traverse and navigate through the calamity site and examine it for prospective survivors. The robot will also be responsible for identifying the prospective dangers for the rescue workers. It will have an air-quality detector embedded which is responsible for analyzing the quality of the air at the calamity site. Hence, if the surrounding air is not safe for rescue workers, it will signal an alert. The robot will be equipped with a camera and microphone that will record a live stream and the underlying human detection architecture will determine if there is a human survivor or not in the current video frame in real time. In case a survivor is located, the robot will instantly transmit the location coordinate to a centralized cloud server which will be continuously monitored manually for any inputs. The system should work as indicated in Figure 1:

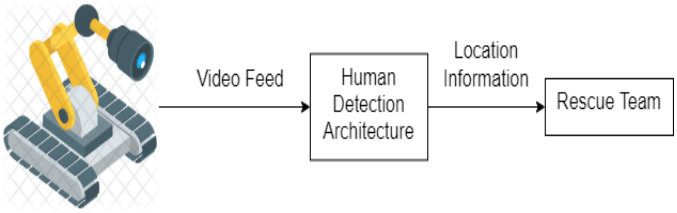


Figure 1. Working of the proposed system.

Figure 2 shows the technical architecture of the proposed solution. It consists of the following parts:

1. Rescue Robo: Used to explore the area and record the live stream.
2. Video Detection Architecture: Used to discover humans in the recorded feed in real time.
3. Audio Detection Architecture: Used to analyze the audio file associated with the live stream and identify the emotion in the human voice.
4. Internet of Things (IoT): Used to stream the location of the detected survivors to a centralized cloud server and to determine the air-quality index of the disaster-affected area.
5. Cloud Server: This is where the service will be deployed. The robot will transmit the location coordinates and the air-quality data to the cloud server.

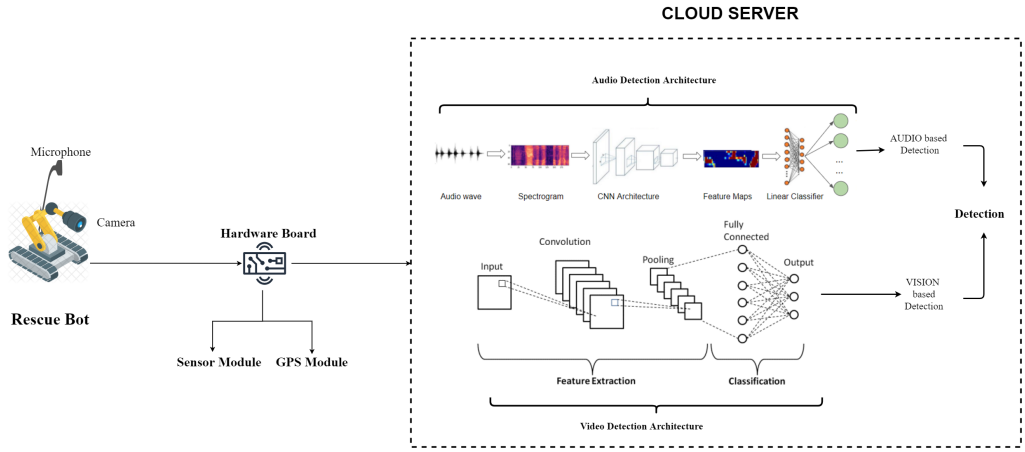


Figure 2. Architecture of the proposed model.

Figure 3 shows the overall architecture of the proposed solution. It gives a basic overview of how the proposed solution is supposed to work.

5.1. Video Detection Architecture

This architecture component breaks up the incoming video feed into small frames and then analyzes them for the presence of a human by treating all the frames as individual frames. This processing comprises the YOLO object detection. You Only Look Once (YOLO) is a technique that is used for object detection in images. It applies a CNN to an image and divides it into several grids. Then, it scans every grid and detects whether the assigned object is present in that grid or not. It calculates the prediction and confidence scores for every grid. The anticipated confidence score examines these bounding boxes. It consists of two fully linked layers and twenty-four convolutional layers in its architecture [4]. The first step for image processing is the size reduction. YOLO reduces the size of any input image to 448 × 448 pixels. It is then fed to a convolutional network. The output of the network is a tensor of dimensions 7 × 7 × 30. This tensor after processing returns the probability distribution and the coordinates of the required rectangle of the bounding box. The image is then passed through a convolutional network, which produces a tensor of 7 × 7 × 30.

The use of a threshold for these confidence ratings (probability) excludes class labels with a score of less than 30% [11]. The existing object detection uses the concept of modifying the classifiers or localizers in order to detect the required objects. The idea is to apply the model at several scales and locations to an image and record the scores obtained. The scores are then analyzed and the regions which score higher than the threshold score are considered to be a region where the object is present.

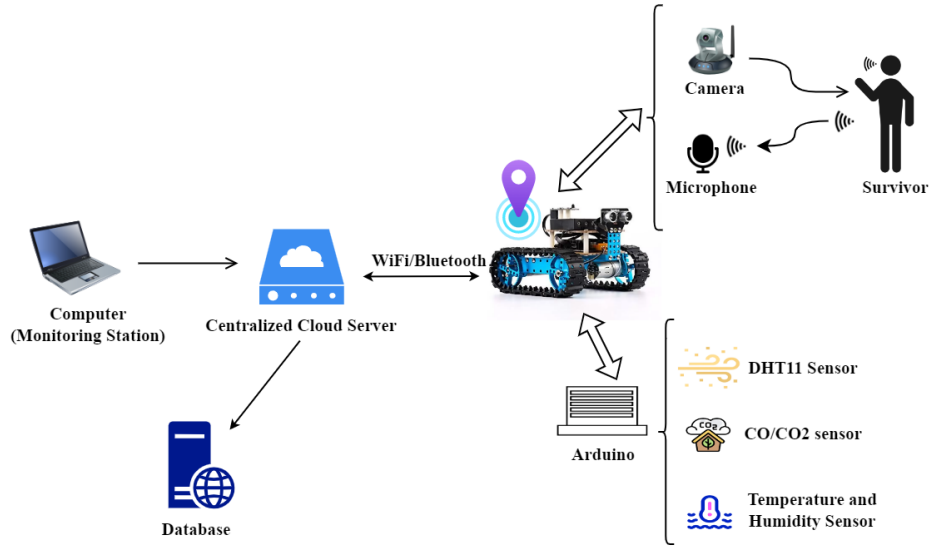


Figure 3. Architecture of the proposed model.

However, here we use a different approach. We run the same neural network to the whole image. It divides the image into several regions and then analyzes them. It then calculates the probabilities for each region and then combines them using a weighted approach. There are a lot of advantages of using the aforementioned approach over the conventional classifier-based approach. The predictions of this model are informed by the global context of the image because it takes the whole picture while classifying [11]. The system is faster than the classifier-based ones. The reason for this is that the system uses a single neural network for the analysis which makes it very false. As per the studies, the model is said to be 100 times faster than the Fast R-CNN and to be 1000 times faster than the R-CNN [11] (Figures 4 and 5).

	Type	Filters	Size	Output
	Convolution	32	3 x 3	256x256
	Convolution	64	3 x 3/2	128x128
1x	Convolution	32	1 x 1	
	Convolution	64	3 x 3	
	Residual			128x128
	Convolution	128	3 x 3/2	64x64
2x	Convolution	64	1 x 1	
	Convolution	128	3 x 3	
	Residual			64x64
	Convolution	256	3 x 3/2	32x32
8x	Convolution	128	1 x 1	
	Convolution	256	3 x 3	
	Residual			32x32
	Convolution	512	3 x 3/2	16x16
8x	Convolution	256	1 x 1	
	Convolution	512	3 x 3	
	Residual			16x16
	Convolution	1024	3 x 3/2	8x8
4x	Convolution	512	1 x 1	
	Convolution	1024	3 x 3	
	Residual			8x8
	Average Pool		Global	
	Connected		1000	
	Softmax			

Figure 4. Layers of the neural network used.

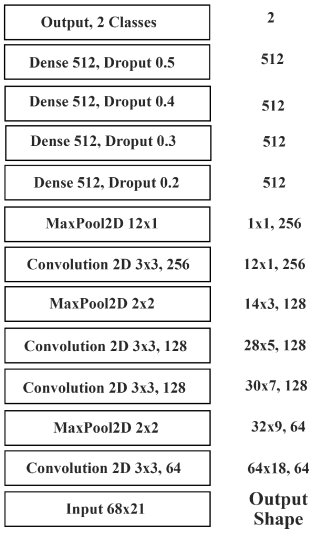


Figure 5. CNN speech music classification architecture.

Advantages of Using YOLO

To detect an item in a picture, most other CNN-based systems rely on classifiers or localizers. These models are applied to an image at various scales and places. In addition, detections are defined as regions with a high confidence score [4]. YOLO has some advantages over traditional methods of object detection.

1. Instead of employing a two-step approach for object classification and localization, YOLO uses a single CNN for both classification and localization.
2. YOLO can process photos at a rate of 40-90 frames per second. This means that streaming video can be handled in real time, with only a few milliseconds of latency.
3. YOLO's architecture makes it exceptionally quick. It is 1000 times faster than the R-CNN and 100 times faster than the fast R-CNN [4].

5.2. Audio Detection Architecture

The CNN input is made up of patches with a size of 68 × 21 (time–feature dimension) and a 695 ms input processing window. There are four convolutional layers and four thick layers in the model. A batch-normalization stage and ReLU activation layers follow all of these levels. Dropout layers follow the dense layers, and as the depth of the network increases, the dropout rates increase. The first convolutional layer has the maximum width to capture the horizontal patterns commonly present in music signals. On the other hand, the last pooling layer works on the temporal dimension, focusing on the pattern with the most activation across this rather long time span. A softmax activation layer is used to implement the output and the probability is predicted.

5.3. Internet of Things (IoT)

The IoT part of the system comes into the picture when we want to transmit some information to the remote cloud server. As discussed in Section 6 of the paper, once the video detection architecture detects the presence of a survivor in the video feed, its location has to be transmitted. Simultaneously, the air quality needs to be monitored continuously for perilous situations. In order to provide these two functionalities, the rescue bot has an IoT component in its architecture which empowers it to perform both in real time.

5.3.1. Transmitting Location Coordinates

As depicted in Figure 6, the IoT setup that transmits the location coordinates includes a GPS tracking device which sends data to a server which is connected with a database. The database is used to persist the location of the survivors. The data are displayed to the personnel device by fetching it from a centralized cloud server. The server receives this

information through the GPS tracking device. It is basically an embedded system that uses GPRS networks to transmit the location coordinates. The information in the database is formatted in a way that Google Earth and Google Maps can search and display [22].

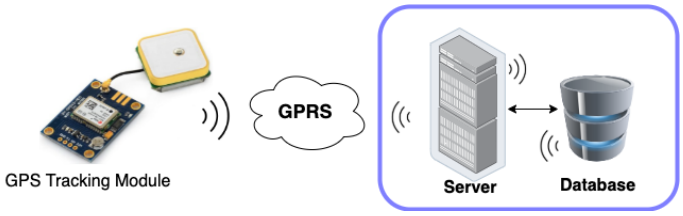


Figure 6. Geo-tracking system.

5.3.2. GPS Tracking Module

As illustrated in Figure 7, the GPS tracking module is based on an AVR RISC microcontroller and several peripheral devices. The RISC microcontroller is an 8-bit low power microcontroller that contains 32k ROM and 2k RAM. The peripheral devices include UART (connects to GPRS/GPS Module), SPI (connects to MMC Module), and I2C (connects to GPIO Control Module). The GPS function is used to detect the device's location. The GPS function then broadcasts the recorded location to the server. The SPI is used to connect to the MMC module in case of connectivity failure or for backup. The MMC module can store location data until the connection is re-established [22].

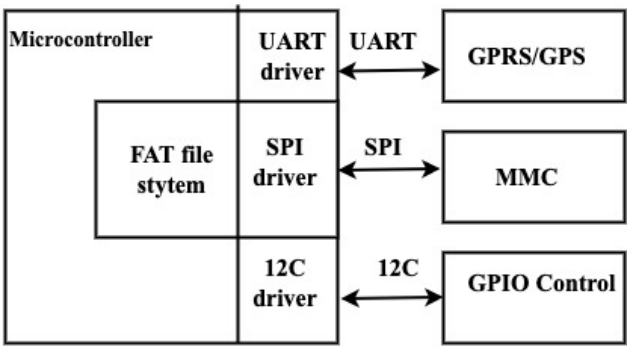


Figure 7. Block diagram of GPS tracking module.

5.3.3. Geo-Tracking Firmware

The geo-tracking firmware is developed and compiled by using the AVR compiler. It works in three phases. The first phase is called initialization. In this phase, the module is initialized and it becomes ready to detect and transmit the location coordinates. This is followed by GPS location reading in which the microcontroller invokes the GPRS/GPS module via a series of AT commands through the UART interface. The third and final phase is data formatting and transmission to the geo-tracking server. The NMEA data are converted to the format specified in Figure 8 and kept as packets. The samples are then grouped together and sent to the geo-server [22].

```
DeviceID,SessionID,UTC_Time,Flag,1311.7836,
N,10056.2358,E,Speed,DDMMYY,Rsv1,Rsv2,Rsv3,
Rsv4,Rsv5,Rsv6^

xxxxxx,yyyyyy,225506,A,1311.7836,N,10056.23
58,E,000.5,191194,aaa,bbb,ccc,ddd,e,f^

xxxxxx,yyyyyy,225526,A,1311.7836,N,10056.23
58,E,000.5,191194,aaa,bbb,ccc,ddd,e,f^\n
```

Figure 8. Format of serialized location data.

5.3.4. Geo-Tracking Server

The GPS tracking module transmits the location coordinate data to the geo-tracking server. In order to do so, it is required to be connected to the GPRS network. The geo-tracking server is a standard PC, preferably Linux based, with support for applications such as PHP, Apache Web Server, and MySQL. [22].

The centralized cloud server is used to receive the location data, add it to the database for persistence, and communicate with the rescue team. A number of sockets of a non-blocking nature are created at the server side. The multi-robot system can then connect to any one of the available sockets and transmit the location coordinates upon detection of the survivors. The received data will then be persisted in the database. The database is designed to support fast CRUD operations [22].

The adopted system should look as depicted in Figure 9.

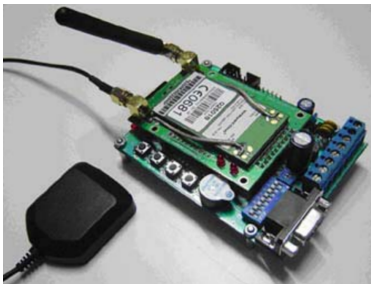


Figure 9. IoT setup.

5.3.5. Monitoring Air Quality

In this system, we use Arduino as a controller module, a sound sensor to detect sound frequencies, a gas sensor to monitor the concentration of CO in the surroundings, a temperature sensor to detect the temperature of the surroundings, and an IoT module [5].

5.3.6. Component Overview

The following are the major modules/components that can be integrated in the rescue robot:

1. Power Module: A power supply, which might be a battery or a controlled power source device, is utilized to give electric power to the boards [5].
2. Controller Module: An Arduino UNO is a controller for a function that already exists. The Arduino board converts analog data from a sensor into cutting-edge data [5].
3. Internet of Things (IoT) Module: The IoT board hoists a variety of web/online application requirements. It is the strongest tool in the arsenal of a system architect. It can be used to add web networks to applications effectively, quickly, and flawlessly. It is an ideal module to use for remote data transmission and remote sensing and controlling, owing to the module's UART refresh feature and site page control [5].
4. Sensor Module: Sensors are used to detect distinct turbulence and frameworks in the atmosphere and in the soil, as well as to gauge the atmospheric conditions. Some notable sensors are DHT11, Temperature Sensor (LM35), and Carbon Monoxide (CO) Sensor [5].

5.3.7. Process Flow

Following are the major modules/components that can be integrated in the rescue robot:

1. Connect the sensors to the microcontroller board.
2. Invoke the sensors to detect the required parameters (temperature, etc.).
3. Process the data to convert it into the required format as stated in Figure 8.
4. Serialize the data so that it can be transmitted over the network.

5. Initialize the WiFi module added to the Arduino board to transmit the data over the network.
6. Transmit the data to the centralized cloud server.

The above process can be depicted by the following flow chart indicated in Figure 10.

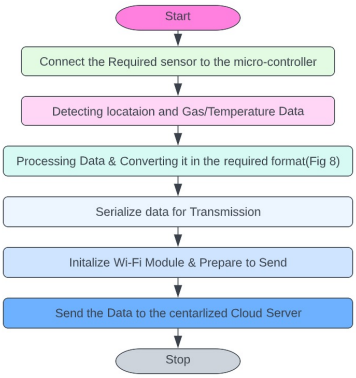


Figure 10. Process flow for monitoring air quality of disaster-affected area.

5.4. Cloud Server

In the domain of cloud robotics, we make use of resources such as the huge processing power, storage, etc., of the virtual machines in the robots. As most of the required computations can be handled in the cloud, our robot's only job remains to relay the data properly to the cloud. Making use of wireless communication is the best choice as we are not limited by any physical constraints [22].

There are several cloud service providers available in the market, such as Amazon AWS, Google Cloud, and Microsoft Azure [23]. Each one of them has its own advantages and disadvantages, as discussed in Table 2. For the scope of our work, we will be using Amazon AWS cloud server services. Figure 11 describes the comparison of security features observed in different domains.

Here, in Figure 12, we have proposed an example of how communication can be achieved between the robots, the base station, and an AWS EC2 (Elastic Compute) instance. The AI model can be deployed on this instance with our web service. The robots can send the audio and video files by leveraging REST APIs.

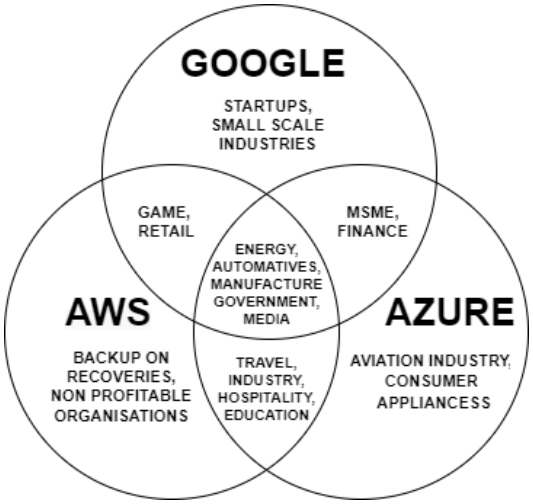


Figure 11. Comparison of security features used in different domains.

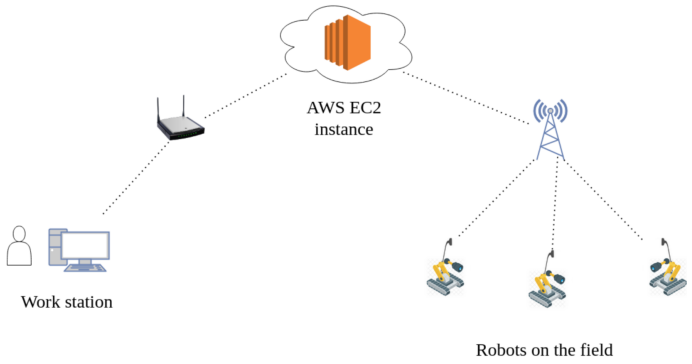


Figure 12. Communication between robots and base stations via Amazon EC2 instance.

6. Multi Robot Coverage and Path Planning

6.1. Introduction

One of the most difficult challenges in robotics is determining the best path for a given region of interest that includes all points while avoiding sub-areas with special features (e.g., obstacles, no-fly zones, etc.). This problem is commonly referred to as coverage path planning (CPP) in the literature, although it can also be referred to as sweeping, thorough geographical search, area patrolling, and so on. In this work, we are concerned with the CPP problem for a multi-robot system.

Of course, after a (presumably) lengthy period of time, a single robot moving randomly in space can achieve this. However, due to the limited battery capacity of autonomous vehicles, as well as the continually expanding areas that must be covered/monitored, various autonomous devices with advanced path planning algorithms have been deployed.

Representing the area is a major challenge for the mCPP problem. Separating the field into identical cells (e.g., in the size of a robot) is one of the most frequent area representation strategies, as it allows for easy coverage of each cell. Because the union of the cells only approximates the target region for any arbitrarily shaped area, this technique, which is also used in our approach, is called approximate cellular decomposition [24].

The approach we are using can be divided into two phases.

Phase 1: Using a constraint satisfaction approach, the available cells are partitioned into as many classes as the robots. The goal of this clustering is to keep the following characteristics:

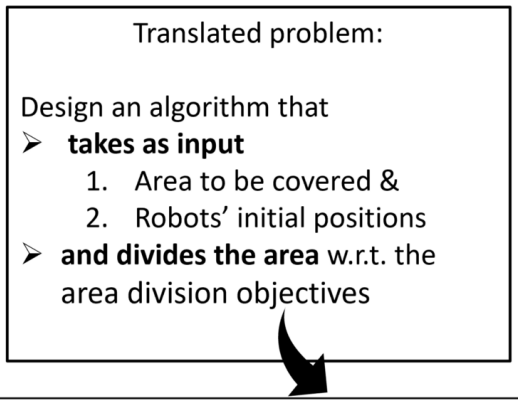
1. Comprehensive coverage.
2. The operation without any prior preparation.
3. The full use of multi-robot dynamics.

The Divide Areas based on Robot's Beginning Locations (DARP) method, which is capable of producing the ideal cell assignment with respect to the robots' initial positions, is at the heart of the suggested algorithm. The latter can be accomplished by using a cyclic coordinate descent strategy with known convergence qualities that is precisely customized to the task at hand.

Phase 2: The STC method develops the ideal path for each robot's cluster in a dispersed manner during the second phase.

6.2. Problem Formulation

The problem is to traverse the entire area in the minimum time possible. The system should take the area to be covered as input and the initial position of the robots as input and should yield an optimum way for the robots to cover the area. This would require the algorithm to divide the area efficiently in accordance with the optimality conditions of the mCPP problem. This is illustrated in Figure 13.



mCPP optimality conditions	Area division objectives
Eliminate backtracking	Exclusive, spatially-connected robot regions
Leave no area uncovered	The union of all sub-areas should reconstruct the original area
Equal robots' trajectories	Equal sub-areas for each robot
Avoid mismatch between current position and starting location	Each robot's initial position should be inside its exclusive region

Figure 13. Communication between robots and base stations via Amazon EC2 instance.

6.3. Voronoi Space Partitioning

Now that we have converted the original mCPP configuration to an identical area-division structure, we can focus on the techniques and methodologies that could be used to deal with it. The Voronoi space partitioning is perhaps the first method that springs to mind.

Several prerequisites are already met by using the Voronoi space partition, but the failure to construct equal sections, regardless of the robots' initial placements (3rd aim), is a serious concern. In this situation, the overall quality of the solution would be heavily influenced by the robots' initial placements.

However, Voronoi partitioning combined with another metric function can satisfy previously established area-division goals and hence solve the original mCPP challenge.

6.4. Divide Areas Algorithm for Multi-Robot Coverage and Path Planning Problem

The main idea is to keep the assignment process from the Voronoi partitioning and tune each robot's metric function to fulfill the 3rd (equal division) target without violating the other objectives. The algorithm uses a cyclic coordinate descent optimization scheme to update each robot's territory independently while still fulfilling the overall mCPP goals [24].

The advantages of using the DARP algorithm:

- Coordinate descent algorithms guarantee convergence.
- It has a fast optimization procedure.
- It is very simple to implement.

The disadvantages of using the DARP algorithm:

- There is no guarantee of spatial connectivity inside each robot's domain.

This algorithm, as stated by Algorithm 1, has been simulated for a system for three robots and the results have been discussed in Section 8.

Algorithm 1 DARP Algorithm

while All area objectives are not met **do**
  Every (x,y) cell is assigned to a robot as per A(x,y) = argmin E_i(x,y) where i $\in 1, \ldots, N$
  for ith robot **do**
    Calculate k_i.
    $m_i \leftarrow m_i + \eta(k_i - f_i)$
    Calculate connectivity matrix C_i
    $E_i \leftarrow C_i \cdot m_i E_i$
  end for
end while

7. Experiment

7.1. Setup

The experiments were performed on "Google Collab" with Intel(R) Xeon(R) CPU @ 2.30 GHz, cache size 46,080 KB, and single 12 GB NVIDIA Tesla K80 GPU. Datasets used for speech classification were GTZAN [25], Scheirer-Slaney [26], and MUSAN [27]. Dataset used for human detection was COCO [28].

7.2. Working

The video and audio feeds are first separated from the given file. Then, audio is sent to ImaSegmenter to obtain intervals having speech, and parallelly, we detect humans in the video feed using YOLO v3. We have used multiprocessing to save time, and we allocate two different processes for these tasks as they are independent of each other.

Once we have detected speech (with audio detection) and humans (using image detection) along with the accuracies, we find the zones where both the audio and humans are detected. We call these zones "intersecting intervals", for finding the accuracy for these intervals we use Equation (1):

$$\text{Accuracy} = \frac{(\alpha \times \text{SA} + \beta \times \text{VA})}{2} \tag{1}$$

where 'SA' is the accuracy of speech detection and VA is the accuracy of human detection by video and α and β denote the confidence ratio selected for speech and human detection. For the areas having high background noises, value of α should be lesser, whereas for situations where there are a lot of obstructions in the video feed due to dust and smoke or the lighting is very less, then β should be less. For finding out the intersection of the intervals, Algorithm 2 has been used.

Algorithm 2 Algorithm to find out the intersection between intervals

int i = j = 0
INTERSECTIONS = [] //Initialize the intersections as an empty array
while (i < N and j < M) **do**
  l = max(ARR1[i][0], ARR2[j][0]) //Lower boundary of intersection
  r = min(ARR1[i][1], ARR2[j][1]) //Higher boundary of intersection
  if l <= r **then**
    INTERSECTIONS.append([l,r]) //valid intersection
  end if
  if ARR1[i][1] <ARR2[j][1] **then**
    i += 1 //If i-th interval's higher boundary is smaller
  else
    j += 1
  end if
end while

8. Results

For human detection with video feed, we used YOLO v3 which provided a mAP score of 57.9%. We improved this mAP score by re-training the model for the purpose of human detection specifically, and with this, we increased the mAP to 70.2 %. With this, we were able to detect humans in real-time speed with outstanding accuracy (Figure 14).

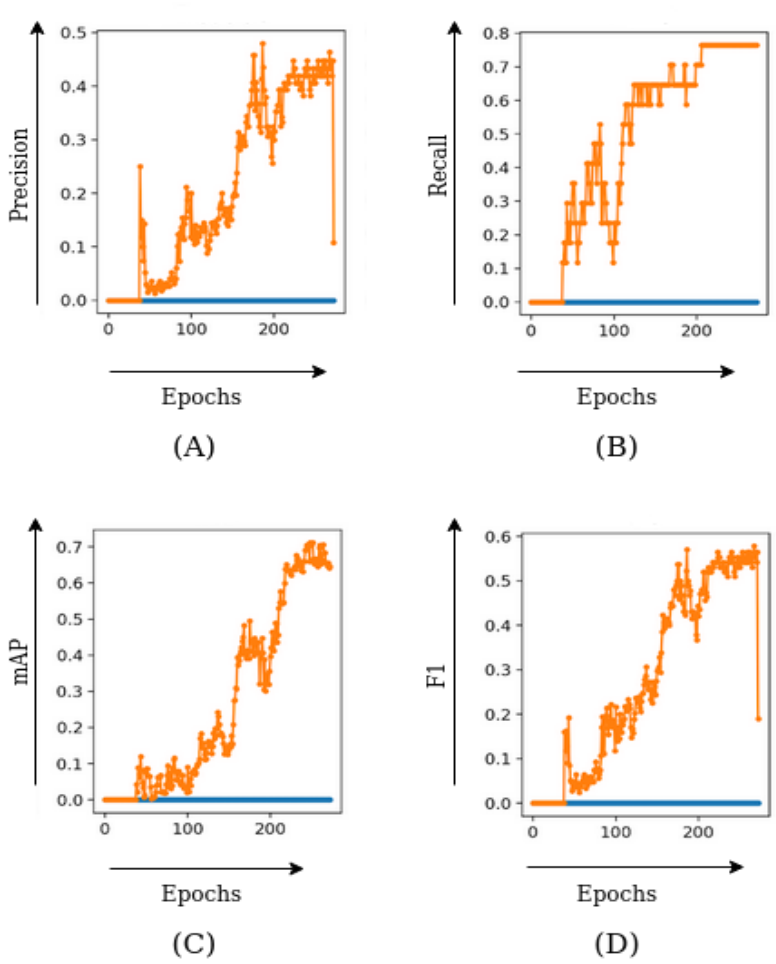


Figure 14. Performance of the video detection model is depicted in this figure. (**A**) Precision vs. Epoch. (**B**) Recall vs. Epoch. (**C**) mAP vs. Epoch. (**D**) F1 score vs. Epoch.

In order to conduct this section of the experiment, we took 10 video files of different time durations. Some had a clear video stream, whereas others did not. This kind of dataset was taken to ensure that the robot can detect humans even if its vision is obscured due to the presence of smoke, etc. We fed the video file as input and logged the output as a CSV file for further analysis. The actual time for which a human was present in the file manually and was compared with the obtained values to calculate the accuracy of the model. We found that the average accuracy of the model came out to be 95.83%. From Table 5, it can be concluded that despite having obscured vision, the robot's performance remains unaffected.

Table 5. Accuracy of the model with respect to detected human time.

	Total Video	Human Detection Duration (s)				Accuracy
Sample Number	**Duration (s)**	**Using Audio Only**	**Using Video Only**	**Overall Time Detected**	**Actual Time Detected**	**(%)**
1	22.83	18	3.6	19.8	20	99
2	20.71	16	15.6	17.4	19	91.58
3	26.07	26.1	3.9	25.8	26	99.23
4	20.01	15.8	0	15.8	16	98.75
5	15.38	0	12.3	12.3	13	94.62
6	13.01	8	12.4	12.4	13	95.39
7	8.17	6.5	1.8	6.5	7	92.86
8	24.9	9.1	0	9.1	10	91
9	16	16	6	16	16	100
10	25	14.2	19.6	23	24	95.83

Performance Analysis

In order to quantitatively analyze our robot's ability to detect humans, we compare its results with the results obtained by other well-established and time-tested human classification models. The mean average precision (mAP) is the most common evaluation metric that is used for evaluating the performance of any object detection model. Table 6 and Figure 15 compare the mAP values obtained in the case of our model and the values obtained by other pre-trained models on the COCO dataset.

Table 6. Comparison of mAP of the used classification model with other pre-trained models on COCO dataset [29].

S. No.	Method	Mean Average Precision (mAP)	Size (MB)
1	SSD321	45.4	410.21
2	DSSD321	46.1	510.34
3	R-FCN	51.9	454.98
4	RetinaNet-50-500	50.9	584.66
5	Our Model	70.2	366.10

The REPERE challenge corpus is an annual French challenge that aims to support the ongoing research works in the field of object detection. We compared the F-measure obtained in the case of our CNN with the other standard models that participated in the challenge. Table 7 and Figure 16 compare the performances obtained in the REPERE challenge corpus dataset in terms of the F-measure.

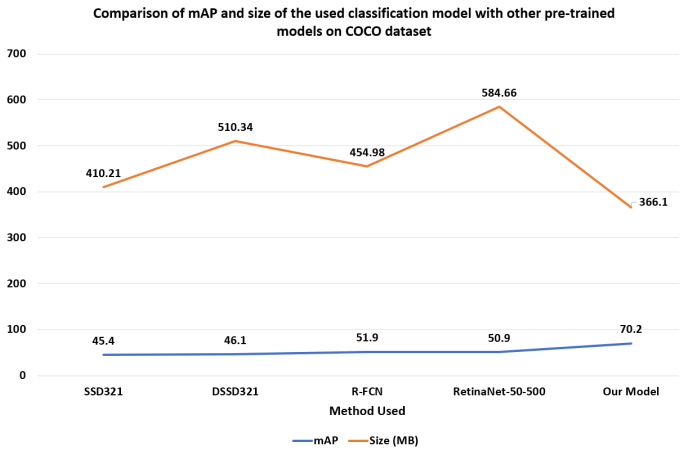


Figure 15. Comparison of mAP of the used classification model with other pre-trained models on COCO dataset.

Table 7. Comparison of performances obtained on REPERE challenge corpus [30].

S. No.	Model	F-Measure
1	GMM	95.74
2	I-vector	95.51
3	Our Model	95.83

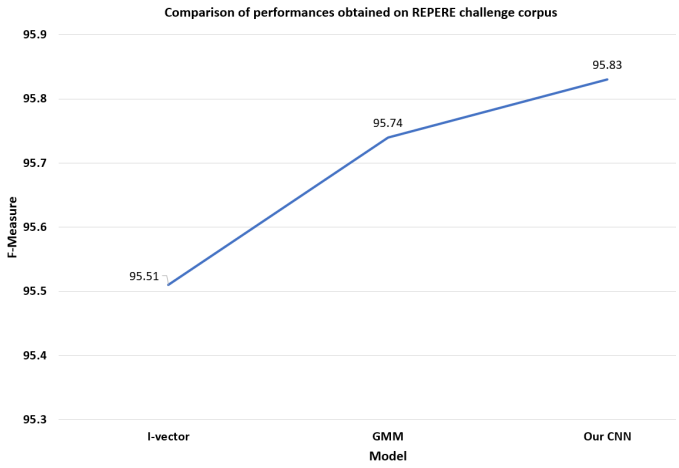


Figure 16. Comparison of performances obtained on REPERE challenge corpus.

Using the proposed approach, we were able to detect humans in videos with background noises and other obstructions. It will be helpful in SaR operations where smoke, dust, noise, low light conditions, etc., make the task of human detection very difficult [31].

To analyze the performance of DARP, we ran a simulation with three robots in four different environments of sizes 8×8, 16×16, 32×32, and 64×64 and some obstacles in between. The metrics we analyzed were the average percentage of the cells that will be covered by a particular robot, the average no. of turns that robots need to make to cover the entire grid, and the total execution time for running the simulation. The simulations were performed on a computer with 8 GiB of RAM and an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz processor. Table 8 and Figure 17 compare the performance of DARP in different environments.

Table 8. Comparison of performance of DARP in different environments.

S. No.	Dimension	Average Percentage of Cells for a Robot	Average Number of Turns	Execution Time (seconds)
1	8×8	27.07	6.33	9.43
2	16×16	27.6	17	9.58
3	32×32	31.9	41.66	38.19
4	64×64	32.51	83.33	72.94

Figure 17 shows that the average percentage of the cells covered by each of the robots is around 33% which shows a uniform distribution of the available area among the available robots. Moreover, we note that the average no. of turns and time taken for execution increases as the grid size increases (Figure 18).

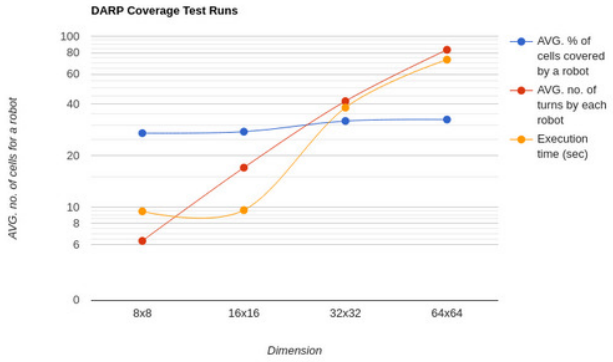


Figure 17. DARP test run analysis on various environment sizes.

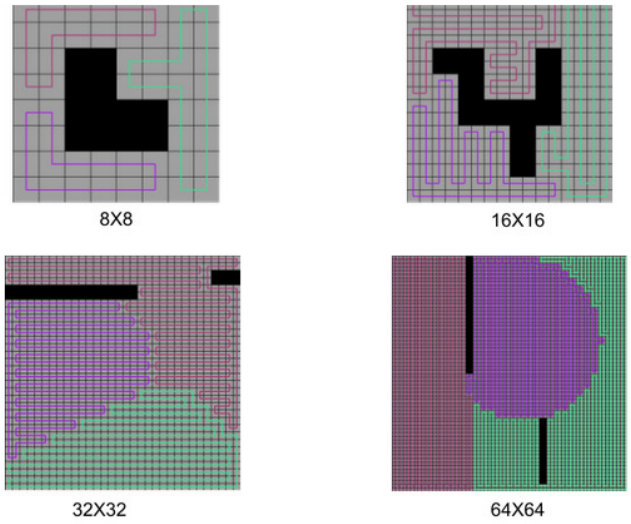


Figure 18. DARP test run analysis on various environment sizes.

9. Concerns

In the proposed solution, the robot will be moving in areas which are completely new to the rescue personnel. As robots and humans both will be exploring the sites, there is always a chance of injury due to collision; hence, some measures should be taken so as to avoid these scenarios. By taking the advantage of motion planning techniques, we can avoid collisions, but for this, the robot should be familiar with the territory and using such techniques will require huge computational resources [32,33].

Another way of avoiding unnecessary collisions between humans and robots is to use sensors, such as force load sensors, sensors on skin, etc., but there is also a downside to this approach, and that is, by adding these sensors, we are incorporating the risk of damage because these sensors usually cannot withstand high external forces which might be present in the operation site. Moreover, the sensors increase the overall weight of the robot and thus the maximum payload of the robot is reduced [34].

Force sensors are currently used in various domains to access external forces, but there are many issues associated with them, such as the bad assessment bandwidth, and there is a lot of noise in the readings also. These problems may lead to the poor performance of the overall architecture [35].

10. Conclusions and Future Scope

Several studies and research works have clearly established the capabilities and usefulness of using robots over UAVs in rescue and search missions. They have proved to be a valuable asset to determine and extract useful and non-redundant data from the surroundings which aids the process of a search. The rescue robots are required to operate in

dangerous disaster-prone areas and often in surroundings that are fatal for humans. Hence, it is absolutely necessary for them to be able to do so with minimal human intervention, preferably autonomously. This is now being achieved with the help of the increased usage of sensors. However, sensors have certain drawbacks in terms of their brittle nature and the associated cost of installing them which in turn increases the price of a bot. Therefore, there is an overwhelming need for developing sensorless rescue robots [36]. There is a growing demand for more research on optimizing the performance of rescue robots without employing the usage of sensors or minimizing their use. Making the robot fully autonomous and/or semi-autonomous and extending its operation range via wireless networks is a promising field of research these days [11]. Moreover, to enable robots such that they move and convey patients out of disaster zones, advanced motion control systems are required. Sometimes it may be very harmful to rescue team workers if they go into disaster areas, so in these types of situations, with the help of IoT technologies, we can use air-quality systems that can help the team. It monitors the air quality of that area by using an Arduino microcontroller. The gas sensor is utilized for checking the different types of dangerous gas [37]. The temperature and dampness sensor continuously provides the actual temperature and humidity conditions within that area [5]. Thus, with the help of IoT technology, it provides all the reports of that area to the monitoring system.

Future Scope

The scope of the current project was limited to enabling the robot to identify the human and transmit its location to a centralized cloud server. However, there is a huge scope for improvement in the future. The robot can be enabled to provide the shortest risk-free path that can be taken by the rescue personnel to reach the survivor using the SOTA path planning algorithm [14]. We can also extend the scope to provide an optimal method for exploring the disaster-affected area using multiple robots [13]. The accuracy of the human detection model can be much improved by incorporating the heat-vision feed [10]. The robot discussed in the paper has a lot of sensors attached to it. The same can be resolved by replacing the rescue bot with a sensorless rescue bot [36]. We can add a method so the robot can detect the survivor's direction with the help of voice frequency. We can add a GPS tractor with a robot so the rescue team can navigate the location of the survivor.

Author Contributions: Conceptualization and Data curation, K.S.; Formal analysis, R.D.; Software, S.K.P.; Methodology and Writing—original draft, A.K.; Supervision and Writing—original draft, G.R.S.; Investigation and Project administration and Writing—original draft, P.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *World Disasters Report 2020: Come Heat or High Water—Tackling the Humanitarian Impacts of the Climate Crisis Together [EN/AR]*; Disaster Report; International Federation of Red Cross and Red Crescent Societies: Geneve, Stwitzerland, 2020.
2. Sambolek, S.; Ivasic-Kos, M. Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors. *IEEE Access* **2021**, *9*, 37905–37922. [CrossRef]
3. Alsamhi, S.H.; Almalki, F.; Ma, O.; Ansari, M.S.; Lee, B. Predictive estimation of optimal signal strength from drones over IoT frameworks in smart cities. *IEEE Trans. Mob. Comput.* **2021**. [CrossRef]
4. Shinde, S.; Kothari, A.; Gupta, V. YOLO based Human Action Recognition and Localization. *Procedia Comput. Sci.* **2018**, *133*, 831–838. [CrossRef]

5. Bharathi, R.U.; Seshashayee, M. Weather and Air Pollution real-time Monitoring System using Internet of Things. *Int. J. Innov. Technol. Explor. Eng. (IJITEE)* **2019**, *8*, 348–354.
6. Clavel, C.; Vasilescu, I.; Devillers, L.; Richard, G.; Ehrette, T. Fear-type emotion recognition for future audio-based surveillance systems. *Speech Commun.* **2008**, *50*, 487–503. [CrossRef]
7. Venkataramanan, K.; Rajamohan, H.R. Emotion recognition from speech. *arXiv* **2019**, arXiv:1912.10458.
8. Guizzo, E.; Weyde, T.; Leveson, J.B. Multi-time-scale convolution for emotion recognition from speech audio signals. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020.
9. Khalil, R.A.; Jones, E.; Babar, M.I.; Jan, T.; Zafar, M.H.; Alhussain, T. Speech emotion recognition using deep learning techniques: A review. *IEEE Access* **2019**, *7*, 117327–117345. [CrossRef]
10. Lygouras, E.; Santavas, N.; Taitzoglou, A.; Tarchanidis, K.; Mitropoulos, A.; Gasteratos, A. Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations. *Sensors* **2019**, *19*, 3542. [CrossRef]
11. Bejiga, M.B.; Zeggada, A.; Nouffidj, A.; Melgani, F. A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sens.* **2017**, *9*, 100. [CrossRef]
12. Llasag, R.; Marcillo, D.; Grilo, C.; Silva, C. Human detection for search and rescue applications with uavs and mixed reality interfaces. In Proceedings of the 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 19–22 June 2019.
13. Tolstaya, E.; Paulos, J.; Kumar, V.; Ribeiro, A. Multi-robot coverage and exploration using spatial graph neural networks. *arXiv* **2020**, arXiv:2011.01119.
14. Tihanyi, D.; Lu, Y.; Karaca, O.; Kamgarpour, M. Multi-robot task allocation for safe planning under dynamic uncertainties. *arXiv* **2021**, arXiv:2103.01840.
15. Rekleitis, I.; New, A.P.; Rankin, E.S.; Choset, H. Efficient boustrophedon multi-robot coverage: An algorithmic approach. *Ann. Math. Artif. Intell.* **2008**, *52*, 109–142. [CrossRef]
16. Batalin, M.A.; Sukhatme, G.S. The analysis of an efficient algorithm for robot coverage and exploration based on sensor network deployment. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005.
17. Shaban, K.B.; Kadri, A.; Rezk, E. Urban Air Pollution Monitoring System with Forecasting Models. *IEEE Sens. J.* **2016**, *16*, 2598–2606. [CrossRef]
18. Kularatna, N.; Sudantha, B.H. An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements. *IEEE Sens. J.* **2008**, *8*, 415–422. [CrossRef]
19. Dhingra, S.; Madda, R.B.; Gandomi, A.H.; Patan, R.; Daneshmand, M. Internet of Things Mobile–Air Pollution Monitoring System (IoT-Mobair). *IEEE Internet Things J.* **2019**, *6*, 5577–5584. [CrossRef]
20. Jung, Y.; Lee, Y.K.; Lee, D.; Ryu, K.; Nittel, S. Air Pollution Monitoring System based on Geosensor Network. In Proceedings of the IEEE International Geoscience & Remote Sensing Symposium, IGARSS 2008, Boston, MA, USA, 8–11 July 2008; pp. 1370–1373. [CrossRef]
21. Walker, J. *Search and Rescue Robots—Current Applications on Land, Sea, and Air*; EMERJ: Boston, MA, USA, 2019.
22. Chadil, N.; Russameesawang, A.; Keeratiwintakorn, P. Real-time tracking management system using GPS, GPRS and Google earth. In Proceedings of the 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Krabi, Thailand, 14–17 May 2008; Volume 1, pp. 393–396. [CrossRef]
23. Dutta, P.; Dutta, P. Comparative Study of Cloud Services Offered by Amazon, Microsoft and Google. *Int. J. Trend Sci. Res. Dev.* **2019**, *3*, 981–985. [CrossRef]
24. Kapoutsis, A.C.; Chatzichristofis, S.A.; Kosmatopoulos, E.B. DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning. *J. Intell. Robot. Syst.* **2017**, *86*, 663–680. [CrossRef]
25. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
26. Scheirer, E.; Slaney, M. Construction and evaluation of a robust multifeature speech/music discriminator. In Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 21–24 April 1997; Volume 2, pp. 1331–1334. [CrossRef]
27. Snyder, D.; Chen, G.; Povey, D. Musan: A music, speech, and noise corpus. *arXiv* **2015**, arXiv:1510.08484.
28. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
29. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Doukhan, D.; Carrive, J.; Vallet, F.; Larcher, A.; Meignier, S. An Open-Source Speaker Gender Detection Framework for Monitoring Gender Equality. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5214–5218. [CrossRef]
31. Alsamhi, S.H.; Almalki, F.A.; Al-Dois, H.; Shvetsov, A.V.; Ansari, M.S.; Hawbani, A.; Gupta, S.K.; Lee, B. Multi-drone edge intelligence and SAR smart wearable devices for emergency communication. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6710074. [CrossRef]

32. Heinzmann, J.; Zelinsky, A. Quantitative safety guarantees for physical human-robot interaction. *Int. J. Robot. Res.* **2016**, *22*, 479–504. [CrossRef]
33. Jimenez, P.; Thomas, F.; Torras, C. Collision detection algorithms for motion planning. In *Robot Motion Planning and Control*; Lecture Notes in Control and Information Sciences; Springer: London, UK, 1998; Volume 229, pp. 305–343.
34. Bicchi, A.; Tonietti, G. Dealing with the safety-performance tradeoff in robot arms design and control. *IEEE Robot. Autom. Mag.* **2004**, *11*, 22–33. [CrossRef]
35. Mitsantisuk, C.; Ohishi, K.; Katsura, S. Estimation of action/reaction forces for the bilateral control using Kalman filter. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4383–4393. [CrossRef]
36. Pillai, B.M.; Suthakorn, J. Challenges for Novice Developers in Rough Terrain Rescue Robots: A Survey on Motion Control Systems. *J. Control Sci. Eng.* **2019**, *2019*, 2135914. [CrossRef]
37. Laijawala, V.; Masurkar, M.; Khandekar, R. Air Quality Monitoring System. 2019. Available online: https://ssrn.com/abstract=3454389 (accessed on 14 April 2022).