Geo located alert via bottom up IoT model in times of COVID-19 and other disasters

H O Lekshmy¹, T R Krishnaprasad², G Vishwanath Gupta³ and K Namitha⁴

¹-⁴ Dept of Computer Science and Engineering, Amrita School of Engineering,
Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India

* Corresponding author: vishwanathg2202@gmail.com

Abstract: Various disasters either man made or natural have necessitated people around the world to adopt ad-hoc practices in managing or mitigating the aftermath of the disaster. In times of a pandemic or prolonged flood or war it is even more pertinent that people do not detach up on the precautionary/ advisory measures. This paper proposes a mechanism integrating a mobile app with other IoT devices, using which the public can participate in issuing a broadcast alert in their locality.

1. Introduction
Wars, epidemics, etc. brings about a situation where the routine of a day-to-day life is disrupted and certain strategies are recommended by the authorities for enhanced safety of people. In the current scenario of a disaster such as the COVID-19 pandemic continues to result in a high number of health-related effects and increasing mortality rates, thus grinding the world economy into a turmoil. To contain the proliferation of the disease, many countries around the world have enforced various preventive measures such as locking down distinct locales depending upon the concentration of COVID-19 patients, limiting transportation, exhorting citizens to adhere to maintaining social distancing etc. [1]. While many laws are implemented to attenuate the effects of any disaster, information technology and communication applications continue to provide a major role in studying as well as in mitigating the adverse effects of crisis or disasters [2][3]. This paper seeks to provide a mechanism which can forewarn the local citizenry about any disaster and can act as a tool in mitigating the aftermath of a disaster. The proposed work integrates a mobile application with Raspberry-pi, connected with other devices in issuing an effective response in a bottom-up manner so that people can take adequate precaution in times of a disaster.

2. Motivation
If prudent precautionary measures are broadcast to people, several lives can be saved in times of a disaster. Very often, there is a shortfall of precise information collection and dissipation to promulgate an immediate response in managing the disaster and thus saving lives and livelihood. Sometimes, maintaining social distancing is required in times of COVID-19 and technology can improve the maintenance of social distancing. Hence, the proposed model brings forth a participatory framework based on a ground-up rather than top-down approach of information flow in which the local citizenry relays information to the area which would be affected, without waiting for the government to help. Thus, this model strives to reduce the workload of the government as well as to democratize disaster management.
3. Related Works

This paper [4] provides a mobile application, developed on Android using Java, for first responders to a disaster, like fire rescue workers. The back end is provided by EC2 instance of Amazon and with Twilio being the choice for communication related purposes. The application enables the team leader and the front-line rescue workers to communicate by using alerts and notifications. In the paper [5], an automated solution that detects a disaster and displays a prospective route to the disaster site from entities that can provide succor to people affected by the disaster is provided. The model proposes using Arduino, Raspberry Pi to receive sensor data from various clusters via wireless transmission. The data is processed by the server, which in turn communicates with the application whose GUI displays an alert along with route of travel to the user in the event of a disaster.

This paper [6] proposes a crowdsourced geographic information system where common citizens feed in reports to the system about the occurrence of a disaster from a mobile GUI application. The data is then fed to a server along with many other parameters. The server runs a simple additive weighting method for measuring the accuracy of information from a given area as well as to prioritize responses. The data of different regions is then made available on a dashboard.

In the paper [7] the authors propose a crowd sourced disaster relief mobile and web application where the victims of a disaster can issue requests for various requirements like medical, food, evacuation etc. These requests are tagged based on their locations and made available via a map-based interface. The application also provides data integrity, data privacy and data security.

The authors of the paper [8] have proposed a wearable device that helps the wearer in with an alert mechanism in ensuring social distancing in times of a pandemic like COVID-19. The model uses Passive Infrared Motion (PIR) sensor, piezoelectric buzzers, NodeMCU amongst other hardware components. If a user comes within a certain distance then infrared radiation is detected by the sensor and connected NodeMCU triggers the alarm. The module can also be used with Bluetooth and WiFi. In paper [9] the authors propose a model which can connect the medical personnel with the people who have symptoms of COVID-19. The application is developed using Java and Firebase and has GUI with location-based data, with an option to contribute as a volunteer.

In our proposed application, we are using alarms connected to Arduino to warn the residents of a locality about any threat regarding an approaching disaster. Our application can also warn people about overcrowding in times of a situation like COVID-19 using location-based services and image processing algorithms.

4. Proposed model

![Diagram](image)

**Figure 1. Overview of the proposed model**

Our model has been categorized into three modules, namely, the mobile application module, image processing module in the server and the IoT module. An overview of the proposed model is in the diagram given in figure 1.
4.1 Mobile application
The mobile application provides the user with the option to take a picture and upload it to the server. The user need not provide any other information regarding the place from which the picture is uploaded. The user interface of the application also has an option for the user to enter the type of picture that is sent to the server. The data sent by the user is processed by the server. Server has dual objectives, to determine the type of disaster and to determine whether there are too many people on the road so that an alert can be issued to lessen the spread of diseases like COVID-19. The UI of the mobile application is as shown in figure 2. Due to the travel restriction, we could not take a photo of the street and hence have taken a sample picture from Unsplash.com by Koushik Das [10]. Once the ‘Send’ button is clicked the image is uploaded onto Firebase. We use Pyrebase to download the image from Firebase to send to the server.

4.2 Server-side image processing for detection of k people
Depending upon the text sent by the user, one of the two purposes of the image processing module in the server is to identify whether the image sent by the user has more than K number of persons in close proximity with each other. The server receives the GPS coordinates of the user along with the picture and the subtext.

Once the coordinates and images are obtained, the server uses HaarCascade [12] algorithm to identify the number of people in a picture. The output of our algorithm, in the given context is either 0 or 1, depending on whether there are less than K or more than K number of people in the image, respectively. If more than K people are detected, then that information is passed into locally installed Raspberry Pi in that particular region. Pi creates alert messages and passes it into speakers.
4.3 Server-side image processing for disaster detection:

Once the image of disaster (say flood) is sent by the user through the mobile application, both the image and the GPS coordinates are received by the server. The server runs a machine learning algorithm to determine whether the image is of a disaster like a flood. The classification is made using a Bag of Features (BoF) model. The BoF model creates a multiplicity of features of an image using Speed Up Robust Features (SURF) technique. K-means algorithm is used to reduce this huge multiplicity of features into a much lesser number. Thus, the feature vector, from each category is given as input into a multi class linear Support Vector Machine (SVM) classifier for classification. The server also needs to evaluate the location of the user from which the image has been sent so that the appropriate alarm can be triggered.

Once the image is classified as a ‘flood’ or any other disaster, then the server transmits the GPS location from which the image has been sent to Raspberry Pi. We are using a dataset mentioned in [11]. From the images in the dataset, we are training the model using 328 images which are in the ‘flood’ category and 275 images of the ‘pollution’ category to balance the images in both the categories. The average accuracy of the model is 0.86. The figure 4 represents a sample data set and while figure 5 represents the test data respectively.

4.4 Raspberry Pi connection with Arduino

Raspberry Pi -3 is a credit card sized minicomputer that can perform all functions of a normal computer. It is widely used in IoT projects due its low cost and portability. Raspberry Pi -3 features a Broadcom system on a chip (SoC) with an integrated ARM-compatible CPU and on-chip GPU. Raspberry Pi-3 board contains 40 general purpose input-output pins, two 5-volt pins, two 3-volt pins and several ground pins. Different versions of Raspberry Pi’s are available in the market. In our project we are using Raspberry Pi-3. The Raspberry Pi and each Arduino UNO is connected to nRF24L01. A circuit of the connection between Raspberry Pi and nRF24L01 is given in figure 6.

nRF24L01 is a single chip wireless radio transceiver module available in the market. This module can send and receive data. nRF24L01 operates on 2.4 GHz-2.5GHz ISM band. This module works with the help of SPI communications. Normally nRF24L01 covers a range of 100 m. We can increase its range by using antennas and powerful batteries. Nowadays more advanced boards are available in the market. In this proposed model we are using 1100-meter-long distance nRF24L01+PA+LNA wireless module with antenna. This module can provide a range over 1100 m.
4.5 Arduino connection with nRFVLI01
Arduino UNO is an open-source microcontroller. This board incorporates fourteen input output pins that can connect to various external elements. Apart from this, Arduino UNO incorporates six analog pins. Arduino UNO is inexpensive when compared with Raspberry Pi. Each Arduino is connected to a speaker. These speakers having a dimension of 70 * 30 * 16 can produce loud stereophonic sound.

![Arduino Connection Diagram]

Figure 6. Raspberry-Pi connection with nRF24L01

The Arduino-nRF24L01 connection circuit diagram is given in figure 7. Power cable of nRF24L01 is connected to VCC in Arduino UNO to provide power. Similarly, the ground pin of nRF24L01 is connected with ground pin in Arduino UNO. MISO from nRF24L01 is connected to pin no twelve of Arduino UNO. Similarly, MOSI in nRF24L01 is connected to pin number eleven. Chip Set (CS) from nRF24L01 is connected to pin number ten of Arduino UNO and SCK is connected to pin number thirteen of Arduino UNO. This is the connection between our Arduino UNO and nRF24L01. After that we connect the stereo speaker with the circuit. Stereo speakers usually have two terminals - A POS terminal and ground. The POS terminal produces sound in a specified direction. POS terminal of the stereo speaker is connected to pin number three of Arduino UNO. Ground of the speaker is connected to the ground pin in Arduino. We can use a breadboard for proper connection.

Each Raspberry Pi is connected with multiple Arduino UNO. This connection can be formed by using the principle of master-slave circuit. Here nRF24L01 connected in Raspberry Pi is considered as Master. While nRF24L01 connected with an Arduino UNO can be considered as a slave. By this method we can connect up to 5 slaves /nRF24L01 in the same wireless network. For example: Let us consider our central server is placed in the district capital, then our local server Raspberry Pi (local server) is placed in each Panchayat, while our Arduino modules are placed in each ward of
the Panchayat. This way we can divide our entire region into small clusters. By this, communication becomes easy in every corner of the district.

4.6 Wireless communication between Raspberry-Pi-nRFVL01 and Arduino-nRFVL01
Here we are creating a wireless network (as is visualized in figure 1) using the nRF24L01 module. RF24 network library helps to form an Arduino wireless network. Now we define a byte array that contains the pipe address, through which nRF24L01 modules communicate, we will assign these addresses in octal format. Here we are using the concept of a master-slave circuit to form the wireless network. nRF24L01 connected with Raspberry is considered as master, while those connected with Arduino are treated as slaves. The GPS location of each Arduino module is mapped with the address that is already stored in Raspberry. When the server sends GPS location to Raspberry, Raspberry finds Arduino located near to the corresponding location, then it retrieves the receiver side address of nRF24L01 connected with that Arduino. After that Raspberry starts transmitting messages through the master nRF24L01 transceiver module. Slave nRF24L01 transceiver module in Arduino sides listens to these transmitted messages. Arduino connected with this module analyses the incoming signals and takes necessary actions like issuing alarm, sending messages to local authorities, etc. To avoid the local populace from panicking upon hearing an alarm, the alarm will be followed by a customized message. The message to be relayed via the alarm will be decided by the Arduino from a set of preexisting messages depending upon the incoming data.

5. Conclusion
Image captured using a mobile phone and its location is sent to the server using a mobile app. Server runs face recognition algorithm and determines whether the image contains more than k people or not. If the output of the image classification algorithm indeed states that there are more than k people, then the server labels the picture as crowded and passes information to Raspberry Pi-3 in corresponding GPS
coordinates. If the image is detected as that of an approaching disaster like a flood, then that information is sent to Raspberry Pi-3 along with the GPS coordinates. Here Raspberry Pi acts as a local server. This is usually placed in places that have an internet connection. Information received in Raspberry Pi is passed to Arduino UNO through nRF24L01 transceiver connected on both sides. Arduino generates auto generated voice messages or alarms to warn people. In case of emergency messages are sent to Raspberry Pi. Raspberry pi uses Google text to speech API to produce high quality audio. This audio file is saved in Raspberry pi. Raspberry locates Arduino UNO located near corresponding GPS coordinates. And passes the audio file to Arduino Uno through nRF24L01. This sound is reflected through 3W Stereo Speaker.

References

[1] Lakshmi Priyadarsini S, Suresh M. Factors influencing the epidemiological characteristics of pandemic CoVld-19: A TISM approach. Int J Healthc Manag 2020;13:89–98. https://doi.org/10.1080/20479700.2020.1755804.

[2] Geetha M, Manoj M, Sarika AS, Mohan M, Rao SN. Detection and estimation of the extent of flood from crowd sourced images. Proc 2017 IEEE Int Conf Commun Signal Process ICCSP 2017 2018;2018-Janua:603–8. https://doi.org/10.1109/ICCSP.2017.8286429.

[3] McLain ML, Babu SK, Bijlani K, Jayakrishnan R, Bhavani RR. Collaborative Game Based Learning of Post-Disaster Management. 2016 IEEE 8th Int Conf Technol Educ Collab 2016:80–7. https://doi.org/10.1109/T4E.2016.23.

[4] Shaghaghi N, Patel S, Pabari B, Francis M. Implementing Communications and Information Dissemination Technologies for First Responders. 2019 IEEE Glob. Humanit. Technol. Conf. GHTC 2019, 2019. https://doi.org/10.1109/GHTC46095.2019.9033107.

[5] Goyal A, Meena K, Kini K, Parmar P, Zaveri M. Real Time Collaborative Processing for Event Detection and Monitoring for Disaster Management in IoT Environment. 2019 10th Int Conf Comput Commun Netw Technol ICCCNT 2019 2019:4–10. https://doi.org/10.1109/ICCCNT45670.2019.8944590.

[6] Puspita IA, Soesanto RP, Muhammad F. Designing Mobile Geographic Information System for Disaster Management by Utilizing Wisdom of the Crowd. 2019 IEEE 6th Int. Conf. Ind. Eng. Appl. ICIEA 2019, 2019. https://doi.org/10.1109/IEA.2019.8715031.

[7] Guntha R, Rao SN, Ramesh M V. Architectural Considerations for Building a Robust Crowdsourced Disaster Relief Application. 2020 Int. Conf. Commun. Syst. NETworkS, COMSNETS 2020, 2020. https://doi.org/10.1109/COMSNETS48256.2020.9027394.

[8] Raghav S, Vijay G, Harika PS, Rao AV, Gopinath A, Sai Shibu NB, et al. Suraksha: Low Cost Device to Maintain Social Distancing during CoVld-19 2020:1476–80. https://doi.org/10.1109/iccc49313.2020.9297503.

[9] Saxena T, Anuragi P, Shinde G, Yadav N, Digalwar M. COWAR: An Android Based Mobile Application to Help Citizens and CoVld-19 Warriors, 2021. https://doi.org/10.1109/cict51604.2020.9312073.

[10] Unsplash n.d.

[11] Barz B, Schröter K, Münch M, Yang B, Unger A, Dransch D, et al. Enhancing Flood Impact Analysis using Interactive Retrieval of Social Media Images 2019.

[12] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2001;1:11–8. https://doi.org/10.1109/cvpr.2001.990517.