Hindusthan International Journal of Multidisciplinary Research (HIJOMR)
Volume 1 Issue 1, September 2024, pp.81-93

DEEP LEARNING AND IOT SYNERGY FOR EFFECTIVE FLOOD MONITORING AND EARLY WARNING USING CROWDSOURCED GEOGRAPHIC INFORMATION

Dr. Gowri. A[1], Mrs. S.Riyaz Fathima[2]

[1]Assistant Professor, Department of Information Technology,
Hindusthan College of Arts & Science, Coimbatore,
Tamil Nadu, India.
gowrihindusthan@gmail.com

[2]Assistant Professor & Head,, Department of Computer Science,
Sree Saraswathi Thyagaraja College, Pollachi,
Tamil Nadu, India.
riyazfathima@stc.ac.in

ABSTRACT

One-eighth of India's land area is prone to flooding, making it one of the world's most flood-prone nations. The primary causes of floods include excessive rainfall, rivers' inability to handle the high flood discharge, and poor drainage systems that fail to swiftly remove precipitation into streams and rivers. But in order to alert people and evacuate them later, knowledge of flooding and rains is vital. This research work employs deep learning, which is performed based on DenseNet& Yolo V3 to more efficiently extract disaster characteristics for automated flood detection using IoT sensors.

Keywords: *Disaster Management, Flood Detection, IoT, DenseNet, YOLO V3.*

1. INTRODUCTION

IoT-based sensors can help identify potentially dangerous situations, such as forest fires and earth movements. They may also be used to monitor river levels and identify instances of flooding. One of the most common and extensive natural disasters in the world is flooding. The main cause of the rise in discharge is persistent rainfall. Rainfall that never stops causes the soil to get saturated and lose its ability to store water, which increases surface drainage. Numerous variables, such as the size and form of a river's drainage basin, the permeability of the rock and soil there, the amount of vegetation there, the number of tributaries that flow into the river, and the gradient of the basin, affect how frequently floods occur in a basin.

Urbanization of the floodplain, deforestation in the flood plains, and inadequate storm drainage in urban areas are all examples of human-induced issues. Furthermore, the intensity and distribution of rainfall are significantly impacted by climate change, which in turn has an effect on floods. In addition to disrupting the transportation and telecommunications networks, floods also harm properties, businesses, and public utilities.

Figure 1: Flood affected Areas

Floods are the most violent, extensive, and frequent natural catastrophes, resulting in significant annual losses in terms of lives lost and property damage. The most recent figures show that the number of people impacted by flooding has been rising quickly as a result of harsh weather, increased urbanization, and insufficient disaster response [3]. Determining the magnitude of the flooding and estimating the damage is crucial in order to offer assistance, compensation, and rescue to those impacted. Since satellites offer synoptic coverage with high periodicity and distinctive spectral signatures, floods may be spotted very instantly. The most effective way to lessen the damage caused by flooding is to install early warning systems.

2. RELATED WORKS

(Sarkar and others, 2017) Examining IoT-based post-disaster management framework approaches is the aim of this paper. Together with some information on IoT-based post-disaster response and recovery, the primary IoT-enabling technologies are discussed. An innovative technique (flowchart) for building an ad hoc network linked to device-to-device interactions is provided as part of a robust post-disaster management framework.

An architecture centered on IOT and intended to create a secure, knowledgeable environment for older adults was proposed by Pandey and Litoriya (2020). In the event of a calamity of any type, including flooding, storms, home fires, volcanic eruptions, and others, the architecture will demand assistance. This emergency program combines aging in-house processes with an assortment of disaster sensors.

The most devastating natural catastrophes are floods, which are also the hardest to model (Mosavi, Ozturk, and Chau, 2018). The chemical study of sapness, accuracy, efficacy, and speed—which is particularly well-researched to offer a comprehensive review of the many ML algorithms utilized in the field—was used to construct machine learning models. A thorough understanding of different approaches in the context of worldwide evaluation is demonstrated by the machine-learning models' performance comparisons using ANFIS, MLP, ANNs, and Deep Learning. After analyzing DL and ML algorithms, ANN—which has an accuracy of 90 to 95—is found to be the best algorithm strategy.

Based on deep learning (DL), (Wu et al., 2020) suggest exploring the distribution patterns of atmospheric and oceanic constituents. 3D-CNN uses technologies such as image processing, data augmentation, and three-dimensional convolutional neural networks to discover the implicit association between the spatial distribution features of three-dimensional environmental variables and changes in the strength of tropical cyclones. In this study, 22 years' worth of western North Pacific analysis datasets (1997–2018) were used, and the model's accuracy was 96%.

(Mao et al. 2020) improved the YOLOv3's Convolutional Layer (CL) using the inverted residuals method in order to well-extract the vehicle. To address the multi-scale vehicle OD problem, three spatial pyramid pooling modules were introduced before each YOLO layer to capture multi-scale information. The soft Non-Maximal Suppression (NMS) method was used in place of NMS for managing the overlapping cars shown in traffic videos; it reduced the number of predicted boxes that were missed as a result of vehicle overlaps. The outcome confirmed an outstanding performance, but with so many cars involved, it became challenging to identify whose vehicle's frame was overlapping.

(Galea, Lawrence, and Kunkel 2020) deduce that the model employed fully-connected classifiers to predict if a tropical storm is current in the input and convolutional layers to improvise a pattern to gaze at. With the ERA-Interim dataset, which covered the period from January 1979 to July 2019, the model achieved an accuracy of 99.08%.

Based on deep learning (DL), (Wu et al., 2020) is suggested by exploring the distribution patterns of atmospheric and oceanic constituents. 3D-CNN uses technologies such as image processing, data augmentation, and three-dimensional convolutional neural networks to discover the implicit association between the spatial distribution features of three-dimensional environmental variables and changes in the strength of tropical cyclones. In this study, 22 years' worth of Western North Pacific analysis datasets (1997–2018) were used, and the model's accuracy was 96%.

In 2020, Zhou, Xiang, and Huang introduced a novel deep learning neural network architecture called GC-LSTM, based on information from satellite cloud images. Graph Convolutional Network (GCN): This network uses various algorithms, such as DL neural network and Graph Convolutional–LSTM Network (GC–LSTM), to process the irregular contiguous structure of spacecraft cloud pictures in a beneficial way. Long STM (LSTM) network: This network uses various algorithms to determine the characteristics of satellite cloud pictures over time. With over 1000 typhoon processes in the dataset, the model's accuracy was 95.12%.

(Haut et al., 2019) created an auto encoder, a cloud-based deep neural network implementation for non-linear compression. The master-slave architecture was used by Apache Spark to link the available processing nodes, and the results of the experiment demonstrate that cloud infrastructure systems provide a practical way to handle sizable remotely sensed data sets.

3. SYSTEM METHODOLOGY

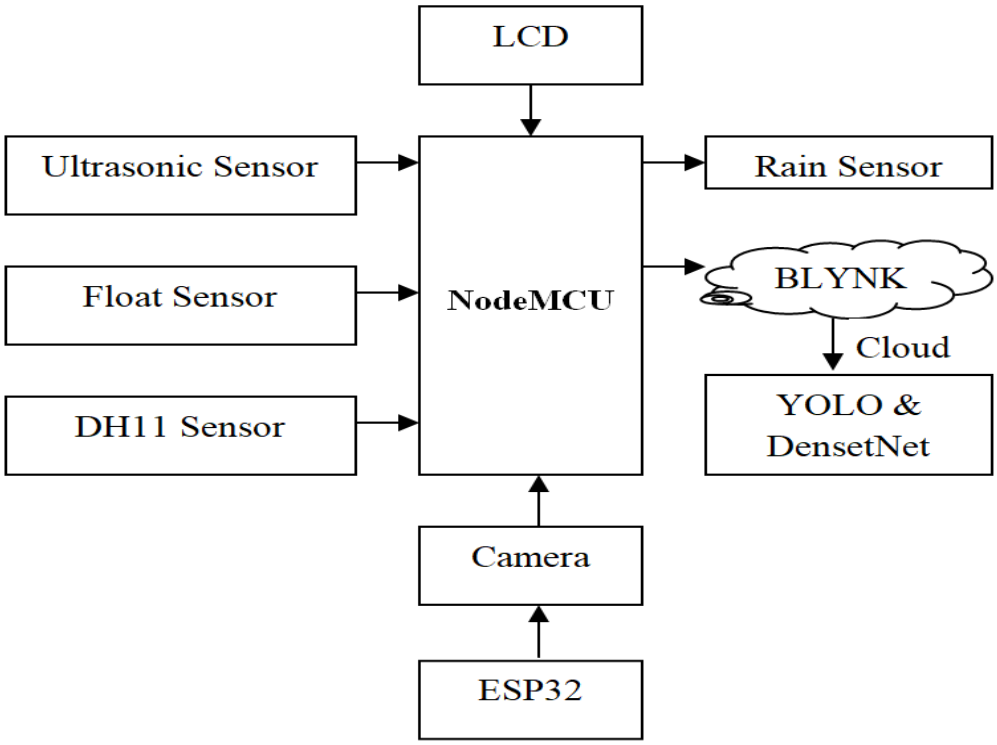

Figure 2: Block Diagram

3.1. Data Collection

This work used IoT sensors to collect the flood-affected image data. These photos were manually classified as either flood or non-flood using distinguishing characteristics that are frequently seen in situations involving both inland and urban flooding. We considered the characteristics of water-logged places, flooded buildings and roads, automobiles, and people to determine the criteria for distinguishing between floods and severe rains. This work used the following sensor to collect the data:

- NodeMCU

NodeMCU is a low-cost open-source IoT platform. Its original hardware components were ESP-12 modules and firmware based on the ESP8266 Wi-Fi SoC. Support for the 32-bit ESP32 MCU was added in subsequent versions. The NodeMCUDev Kit/Board includes the ESP8266 WiFi chip. The TCP/IP protocol is used by Espressif Systems' low-cost ESP8266 Wi-Fi chip. To learn more about the ESP8266, use the WiFi Module.

- DHT11

The DHT11 is an inexpensive digital sensor that monitors humidity and temperature. This sensor can simply communicate with the Arduino so that it can quickly determine the temperature and humidity. For this sensor to have greater resistance values even for the smallest temperature change, semiconductor ceramics or polymers are frequently used in its construction. One reading per second, the sampling rate for this sensor is 1 Hz.

➢ ESP32-CAM

The ESP32-S microcontroller is used by the minuscule ESP32-CAM camera module. Along with several GPIOs for attaching peripherals, the OV2640 camera is also included, along with a microSD card slot. This slot may contain files that will be distributed to clients or images captured by the camera.

➢ Rain Sensor

A switching mechanism known as a rain sensor can detect the presence of precipitation. The idea behind it is that when it rains, the switch is often locked, so it works like a simple switch. This sensor module lets you detect moisture via analog output pins, and when the moisture threshold is crossed, it sends a digital signal. The purpose of this sensor is to install a rain sensor in various headwater sources, such as falls, lakes, rivers, and so on. This sensor detects the intensity of the rain and creates a notification when it starts to rain severely.

➢ Ultrasonic Sensor

An ultrasonic sensor is a technological device that measures an object's distance using ultrasonic sound waves and then transforms the sound into an electrical signal when it gets reflected. The speed at which ultrasonic waves travel beyond that of audible sound. Proximity sensors frequently work with ultrasonic sensors. They are used in vehicle technologies such as self-parking and collision avoidance. Both industrial processes and robot obstacle detection systems include ultrasonic sensors.

➢ Float Sensor

Level sensors include the floating switch kind. To determine the liquid level, a gadget is used. The switch can be used with an alarm, pump, or other equipment. Sensing the water level in the overhead tank is helpful. Float sensors are electrical sensors that sense changes in liquid level relative to a predetermined level and activate automatically.

➢ LCD

LCD module at 40% relative humidity and 40% temperature, respectively. Higher temperatures can cause the display's overall color to alter, while lower temperatures can delay the rate at which the display blinks. When the temperature drops within the designated range, the display will normalize. Polarizer peel-off, bubble formation, and polarization deterioration can all be brought on by heat and humidity.

3.2. Deep Learning Models for Object Recognition

- **DenseNet 121**

DenseNet is an architecture that uses shorter linkages across layers to train DL networks more effectively while also making them deeper. Figure 3 illustrates the design of DenseNet (Densely Connected Convolutional Networks), a deep neural network that presents the idea of dense connections between layers. Some of the issues with gradient vanishing and feature reuse that can arise with conventional convolutional neural networks (CNNs) are addressed by DenseNet designs. Two layers of a CNN called DenseNet are connected to every other layer farther down the network. One variation of the DenseNet architecture is DenseNet121.

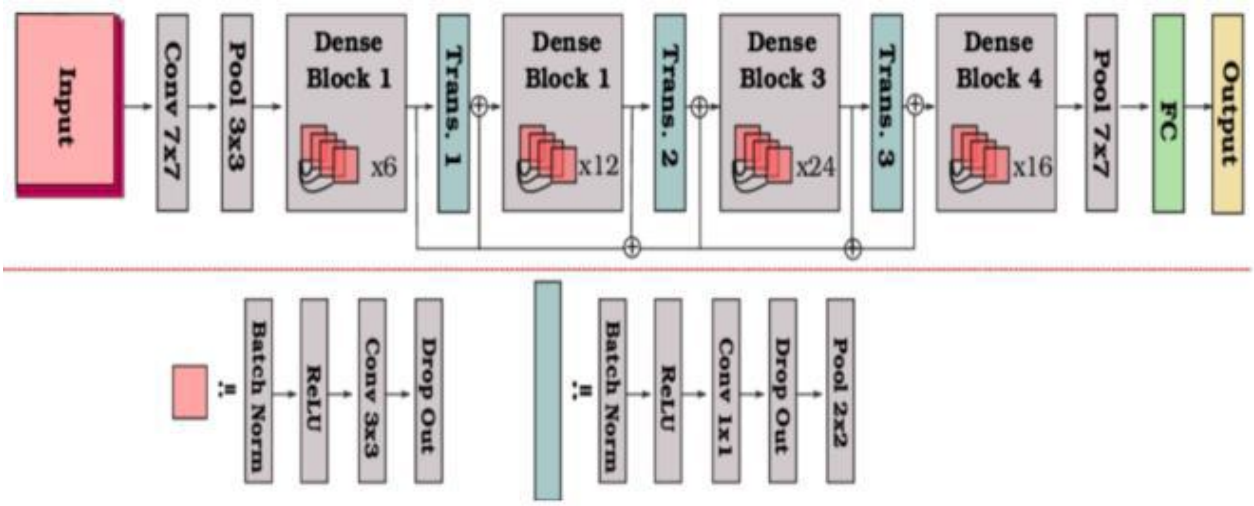

Figure 3: DenseNet Architecture

- **Dense Connectivity:**
  - DenseNet presents dense blocks, in which every layer is feed-forward linked to every other layer.
  - This dense connection makes feature reuse easier and helps solve the vanishing gradient problem.
  - In a dense block, the feature maps from all previous layers are concatenated together and supplied as input to the current layer.
- **Bottleneck Layers:**
  - Before dense connection, DenseNet uses bottleneck layers to minimize the number of channels.
  - A bottleneck layer reduces the number of input channels and calculations by using a 1x1 convolution followed by a 3x3 convolution.
  - This lowers the model's parameters and memory needs.
- **Transition Layers:**
  - To regulate the spatial dimensions and channel diameters, DenseNet has transition layers in between dense blocks.
  - Transition layers smooth the transition from one dense block to the next by reducing the number of channels and spatial resolution.
  - Typically, they are composed of a 1x1 convolution followed by average pooling or stride convolution.
- **Growth Rate:**
  - The number of feature mappings added to each layer in a dense block is determined by the growth rate, a hyperparameter.
  - Although it uses more memory and processing power, a faster growth rate can produce richer feature representations.
- **Fully Connected Layer and Global Average Pooling:**
  - DenseNet designs frequently employ a fully connected layer for the ultimate prediction after utilizing global average pooling.
  - The fully connected layer produces the final class probabilities, and global average pooling reduces the spatial dimensions of feature maps to one value per channel.

3.3. Regularization and Batch Normalization:

- To minimize the internal covariate shift and speed up training, DenseNet uses batch normalization.
- To prevent over fitting, dropout is also used for regularization.

In particular, DenseNet121 designates a DenseNet architectural variation with around 121 layers. It can be adjusted or used for transfer learning on a variety of computer vision problems because it has already been pre-trained on sizable picture datasets like ImageNet. DenseNet architectures are well-liked for a variety of deep learning applications due to their robust performance, economical parameter usage, and capacity to capture rich feature representations.

- **YOLO** - V3

YOLO - V3 is used for object detection. One type of one-stage technique that converts target identification into a regression issue is called YOLO (You Only Look Once). When compared to Faster R-CNN, YOLO obtains location and category predictions directly. At the output end, YOLO-V2 adds convolutional layers in place of fully connected layers (FC). Furthermore, YOLO-V2 utilizes the Batch Normalized New Features Extraction Network (Darknet19).

Darknet53 is used by YOLO-V3 as its feature extraction network. Darknet53 uses a complete.

convolutional network to avoid the information loss brought on by pooling layers (FCN). Essentially, the network is composed of 1 × 1 or 3 × 3 convolutional kernels. It's named Darknet53 since it has 53 convolutional layers in it. YOLO-V3 utilizes the feature pyramid network (FPN) concept. For every input picture, the network performs a five-time down-sampling operation. After the feature extraction process, the resultant feature map is down-sampled by 32×, making it 1/32 of the original image's size. The last three down-sampled layers are then sent by YOLO-V3 to the detection layers to find targets. YOLO-V3 forecasts on three different scales. The three scales, which are 13 x 13, 26 x 26, and 52 x 52 measures. The feature maps are then concatenated. Similarly, this work used the same reasoning for feature maps that have been 8× and 16×down-sampled.

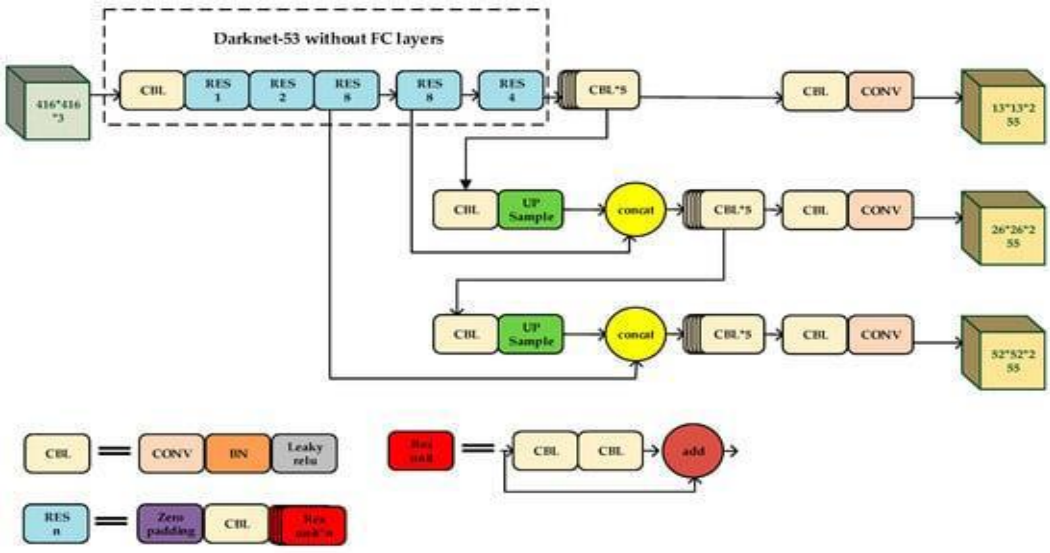

Figure 4: The network of YOLO - V3

4. EXPERIMENTAL RESULTS

In this work the flood affected image, area image are collected through IoT sensors,

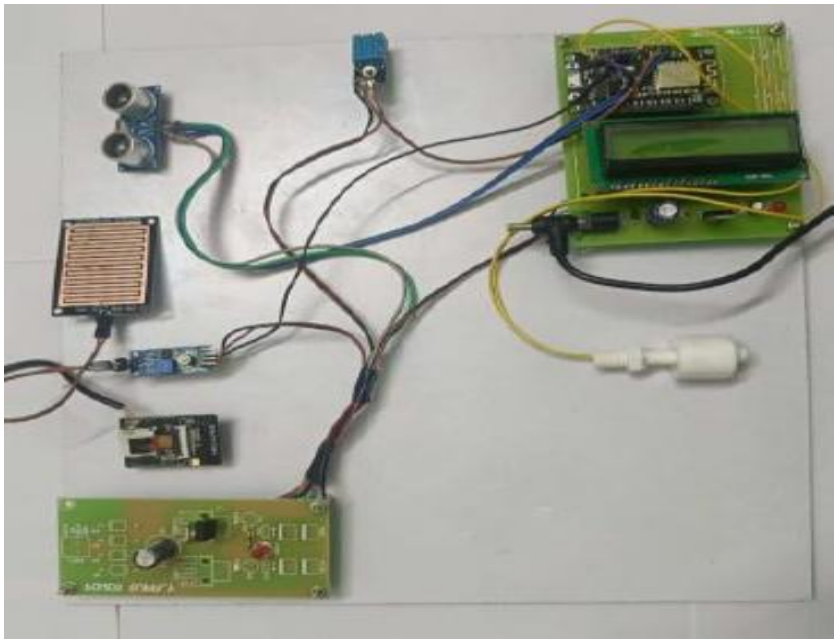

Figure 5: IoT Hardware Setup

Figure 5 shows the IoT hardware setup to collect the input images.

Figure 6: Number of Image Trained

Figure 6 illustrates the number of images used to train for object detection and Figure 7 shows the training results.

```
Epoch 1/15
45/45 [==============================] - 392s 8s/step - loss: 0.7717 - accuracy: 0.5632 - val_loss: 0.7284 - val_accuracy: 0.66
48
Epoch 2/15
45/45 [==============================] - 332s 7s/step - loss: 0.6193 - accuracy: 0.6681 - val_loss: 0.5523 - val_accuracy: 0.74
52
Epoch 3/15
45/45 [==============================] - 312s 7s/step - loss: 0.5289 - accuracy: 0.7292 - val_loss: 0.4840 - val_accuracy: 0.77
84
Epoch 4/15
45/45 [==============================] - 343s 8s/step - loss: 0.4325 - accuracy: 0.7958 - val_loss: 0.3037 - val_accuracy: 0.86
15
Epoch 5/15
45/45 [==============================] - 312s 7s/step - loss: 0.4006 - accuracy: 0.8292 - val_loss: 0.2575 - val_accuracy: 0.90
03
Epoch 6/15
45/45 [==============================] - 318s 7s/step - loss: 0.3378 - accuracy: 0.8708 - val_loss: 0.2138 - val_accuracy: 0.92
24
Epoch 7/15
45/45 [==============================] - 319s 7s/step - loss: 0.2981 - accuracy: 0.9014 - val_loss: 0.1980 - val_accuracy: 0.94
46
Epoch 8/15
45/45 [==============================] - 516s 12s/step - loss: 0.2828 - accuracy: 0.9132 - val_loss: 0.1891 - val_accuracy: 0.9
529
Epoch 9/15
45/45 [==============================] - 286s 6s/step - loss: 0.2583 - accuracy: 0.9187 - val_loss: 0.1853 - val_accuracy: 0.95
57
Epoch 10/15
45/45 [==============================] - 243s 5s/step - loss: 0.2340 - accuracy: 0.9382 - val_loss: 0.1594 - val_accuracy: 0.97
51
Epoch 11/15
45/45 [==============================] - 254s 6s/step - loss: 0.2077 - accuracy: 0.9549 - val_loss: 0.2036 - val_accuracy: 0.95
84
Epoch 12/15
45/45 [==============================] - 244s 5s/step - loss: 0.1861 - accuracy: 0.9708 - val_loss: 0.1951 - val_accuracy: 0.95
84
Epoch 13/15
45/45 [==============================] - 238s 5s/step - loss: 0.1798 - accuracy: 0.9708 - val_loss: 0.1732 - val_accuracy: 0.98
34
Epoch 14/15
45/45 [==============================] - 254s 6s/step - loss: 0.1699 - accuracy: 0.9736 - val_loss: 0.1818 - val_accuracy: 0.96
12
Epoch 15/15
45/45 [==============================] - 247s 5s/step - loss: 0.1575 - accuracy: 0.9736 - val_loss: 0.1738 - val_accuracy: 0.96
40
```

Figure 7: Training Output

The model starts with a pre-trained DenseNet121 backbone, applies temporal aggregation using time-distributed layers, adds some fully connected layers for classification, and then fine-tunes the model for improved performance on your specific task.

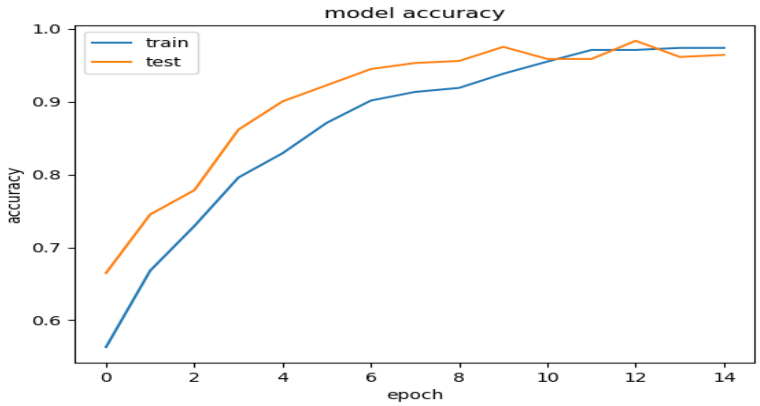

Figure 8: Accuracy curve for DenseNet121

When the testing dataset with I3D model was used to evaluate the testing accuracy, the result was 96.19% with pre-trained DenseNet121. Accuracy- Loss graphs for I3D model with DenseNet121 are shown in figure 8 and figure 9 respectively

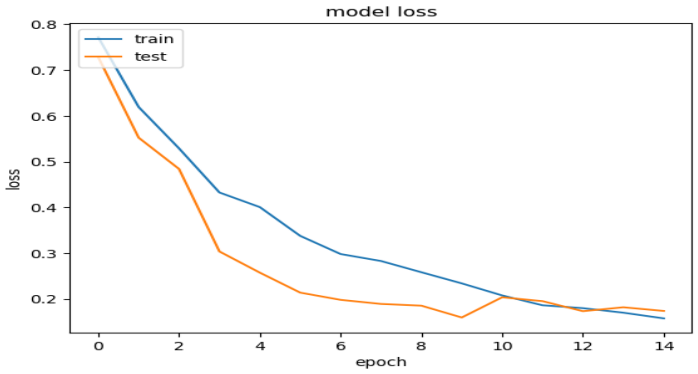

Figure 9: Loss curve for with DenseNet121

```
The image is of a Flood
CLASS: person, Location: 11.7338,78.9592
water level: 72, Temperature: 25, Humidity:

CLASS: person, Location: 11.7338,78.9592
water level: 80, Temperature: 25, Humidity:
```

Figure 10: Flood Detected Image using DenseNet

Figure 10 depicts Flood Detected Image using DenseNet. IoT is used to find water level, temperature and humidity level using IoT sensors.

```
The image is of a No flood
```

Figure 11: Non-flood Image using DenseNet

Figure 11 demonstrates the non-flood image, which is detected using DenseNet.

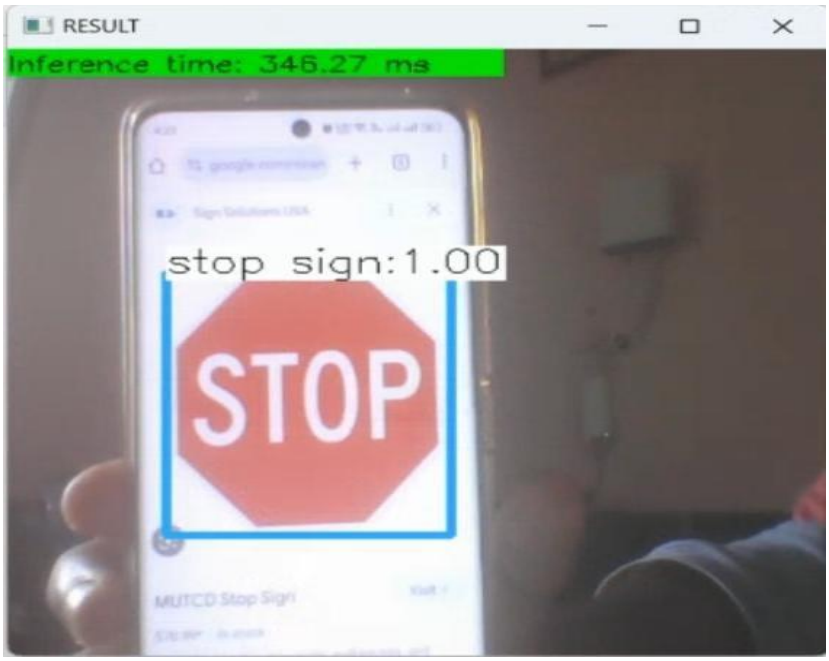

Figure 12: Street Sign Object Detected Result

Figure 12 illustrates Stop Sign Object Detected Result using YOLO – V3.

5. CONCLUSION

This can expedite the dissemination of information to the authorities and rescue personnel and facilitate communication amongst the many impacted groups. The app will provide all of the information gathered and share it with the relevant authorities, and the app will establish the network needed for communication between organizations and authorities. It will provide the relief management team with an advantage since it will make it possible for them to get in touch with the impacted population.

REFERENCES

[1] Aher, S. B. and Tiwari, D. R. (2018) 'Trends in causes and impacts of accidents in Indian railway', Journal of Social Sciences, 55(1–3), pp. 34–44. doi: 10.30901/24566756.2018/55.1-3.2226.

[2] Choubin, B. et al. (2019) 'Snow avalanche hazard prediction using machine learning methods', Journal of Hydrology, 577(July), p. 123929. doi: 10.1016/j.jhydrol.2019.123929.

[3] Fernandez, D. et al. (2019) 'FPGA implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images', Journal of Real-Time Image Processing, 16(5), pp. 1395–1406. doi: 10.1007/s11554-016-0650-7.

[4] Jung, D. et al. (2020) 'Conceptual framework of an intelligent decision support system for smart city disaster management', Applied Sciences (Switzerland), 10(2). doi: 10.3390/app10020666.

[5] Kamruzzaman M, Sarkar NI, Gutierrez J, Ray SK. A study of IoT-based post-disaster management. In: 2017 International Conference on Information Networking (ICOIN) [Internet]. Da Nang, Vietnam: IEEE; 2017 [cited 2018 Jun 25]. p. 406 10. Available from: http://ieeexplore.ieee.org/document/7899468/

[6] Khan, A., Gupta, S. and Gupta, S. K. (2020) 'Multi-hazard disaster studies: Monitoring, detection, recovery, and management, based on emerging technologies and optimal techniques', International Journal of Disaster Risk Reduction, 47, p. 101642. doi: 10.1016/j.ijdrr.2020.101642.

[7] Mois, G., Folea, S. and Sanislav, T. (2017) 'Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring', IEEE Transactions on Instrumentation and Measurement, 66(8), pp. 2056–2064. doi: 10.1109/TIM.2017.2677619.

[8] Mosavi, A., Ozturk, P. and Chau, K. W. (2018) 'Flood prediction using machine learning models: Literature review', Water (Switzerland), 10(11), pp. 1–40. doi: 10.3390/w10111536.

[9] Pandey, P. and Litoriya, R. (2020) 'Elderly care through unusual behavior detection: A disaster management approach using IoT and intelligence', IBM Journal of Research and Development, 64(1–2), pp. 1–11. doi: 10.1147/JRD.2019.2947018.

[10] Saha, H. N. et al. (2017) 'Disaster management using Internet of Things', 2017 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON 2017, (August), pp. 81–85. doi: 10.1109/IEMECON.2017.8079566.

[11] Shakya, S., Kumar, S. and Goswami, M. (2020) 'Deep Learning Algorithm for Satellite Imaging Based Cyclone Detection', IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, pp. 827–839. doi: 10.1109/JSTARS.2020.2970253.

[12] Sun, W., Bocchini, P. and Davison, B. D. (2020) Applications of artificial intelligence for disaster management, Natural Hazards. Springer Netherlands. doi: 10.1007/s11069-020-04124-3.

[13] Ullah, I. et al. (2019) 'Cloud Based IoT Network Virtualization for Supporting Dynamic Connectivity among Connected Devices', Electronics, 8(7), p. 742. doi: 10.3390/electronics8070742.4.

[14] Wu, C. et al. (2020) '3D CNN-Enabled Positioning in 3D Massive MIMOOFDM Systems', IEEE International Conference on Communications, 2020- June, pp. 0–5. doi: 10.1109/ICC40277.2020.9149427.

[15] Zafar, U. et al. (2019) 'Exploring IoT Applications for Disaster Management: Identifying Key Factors and Proposing Future Directions', pp. 291–309. doi: 10.1007/978-3-319-99966-1_27.

[16] Zhou, J., Xiang, J. and Huang, S. (2020) 'Classification and prediction of typhoon levels by satellite cloud pictures through GC–LSTM deep learning model', Sensors (Switzerland), 20(18), pp. 1–17. doi: 10.3390/s20185132.

[17] Wu, C. et al. (2020) '3D CNN-Enabled Positioning in 3D Massive MIMOOFDM Systems', IEEE International Conference on Communications, 2020- June, pp. 0–5. doi: 10.1109/ICC40277.2020.9149427.

[18] Zhou, J., Xiang, J. and Huang, S. (2020) 'Classification and prediction of typhoon levels by satellite cloud pictures through GC–LSTM deep learning model', Sensors (Switzerland), 20(18), pp. 1–17. doi: 10.3390/s20185132.

[19] Haut, J. M. et al. (2019) 'Cloud deep networks for hyperspectral image analysis', IEEE Transactions on Geoscience and Remote Sensing, 57(12), pp. 9832–9848. doi: 10.1109/TGRS.2019.2929731.
[20] Galea, D., Lawrence, B., and Kunkel, J.: Detecting Tropical Cyclones using Deep Learning Techniques , EGU General Assembly 2020, Online, 4–8 May 2020, EGU2020-9870, https://doi.org/10.5194/egusphere-egu2020-9870, 2020.
[21] NushratHumaira, Vidya S. Samadi, And Nina C. Hubig, "DX-FloodLine: End-To-End Deep Explainable Pipeline for Real Time Flood Scene Object Detection From Multimedia Images", IEEE ACCESS, Volume 11, 2023.

AUTHOR PROFILE:

Author – 1

Dr.Gowri.A is an Assistant Professor in the Department of Information Technology at Hindusthan College of Arts & Science, with research interests in artificial intelligence, machine learning, and Digital Image Processing. She holds a Ph.D. from Bharathiar University and with the teaching experience of 18 years. She has published numerous papers in top-tier conferences and journals, has written 3 books on Computer Science, receiving awards such as the Best Senior Faculty Award and Best Educator Award. A member of the Computer Society of India, Dr. Gowri's work focuses on developing innovative solutions for real-world problems.

Author – 2

Ms.S.Riyaz Fathima had obtained M.Sc(CS)., M.Phil.(CS) from Bharathiar University, Coimbatore, India. She is pursuing Ph.D. (Computer Science) from Bharathiar University, Coimbatore, India. She has more than eight years of teaching Computer Science in Colleges Connected to Bharathiar University, Coimbatore, India. She is currently employed in SreeSaraswati Thiyagaraja College (Autonomous), Pollachi. Her specializations include Computer Networks, Software Engineering, Database Management Systems, and current area of research interest is Information Security. She has published many chapters and research papers, presented many contemporary analysis researches and has written beneficial text books on Computer Science.