International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 13 Issue 01 January 2025, Page no. – 4731-4738

Index Copernicus ICV: 57.55, Impact Factor: 8.316

DOI: 10.47191/ijmcr/v13i1.05

Road Accident Emergency Reporting Management System using Neural Network

Nlerum Promise Anebo[1], Igbudu Ezebunwo[2]

[1]Computer Science Department, Federal University Otuoke, Bayelsa State, Nigeria

[2]Computer Science Department, Rivers State University, Nigeria

ARTICLE INFO

Published Online:
11 January 2025

Corresponding Author:
Dr. Promise A. Nlerum

ABSTRACT

Given the high number of road accidents that result in deaths every day, victims may lose their lives if emergency medical personnel or concerned parties fail to arrive promptly to intervene in an accident or if they do so, because they are unaware of the location of a nearby hospital or medical facility, they may not be able to provide the much-needed help. To avert this scenario, this research work seeks to deploy a Deep Learning Road Accident Emergencies Management model, which by examining of video and sensors' data, identifies and reports traffic incidents in real time. A Neural Network (NN) is used by the model to recognize accident scenarios based on anomalous motion patterns inside video frames, including sudden halts and crashes. Put into practice in python, the program incorporates deep learning libraries such as OpenCV and TensorFlow for strong data processing and analysis in real time. To train and evaluate the model, an 80/20 data split, which results in reduced latency and high accuracy in accident detection. The Key performance metrics include a testing accuracy of 96%, an F1-score of 87.5%, and a response time of less than two seconds on average. The study advances the domains of artificial intelligence, transportation safety, and emergency management by offering a practical, real-time solution for automated accident reporting. This quick detection capability allows for timely emergency alerts, which are sent to responders via a web-based interface, giving them precise location details. The model's promise as a life-saving tool in smart city infrastructures is highlighted by its capacity to deliver precise and prompt replies. The Road Accident Emergency Reporting Model has important ramifications for increasing road safety, decreasing emergency response times, and boosting public safety in general due to its strong design and low latency.

KEYWORDS: Deep Learning, Neural Network, OpenCV, TensorFlow, F1-Score,

I. INTRODUCTION

Around 1.3 million people die in traffic accidents worldwide, and between 20 and 50 million are injured (Alomari, 2021). The victims' families, jobs, towns, and states have all been significantly impacted by these injuries and fatalities. Because more people are seeking medical attention as a result of traffic accidents, health security is at risk. Many road traffic accident victims may experience various health issues as a result of this overstretching of resources, particularly if they have recently arrived at the hospital during their sickness cycle Many road traffic accident victims may experience various health issues as a result of this overstretching of resources, particularly if they have recently arrived at the hospital during their sickness cycle (Asor, 2021). Road accidents can be caused by a variety of factors, including: driver distraction, including using electronics while driving, conversing with other passengers, eating while driving, and handling children or pets in the backseat; driver impairment, including fatigue, illness, and driving while under the influence of alcohol or drugs, both legal and illicit; Road conditions, such as materials on the road surface that cause the roads to become slick; road damage, such as potholes; mechanical failure, such as flat tires or tires blowing out; brake failure, axle failure, steering mechanism failure.

An accident involving at least one vehicle on a public road that results in at least one person being hurt or killed is referred to as a road accident. Even if there is occasionally an increase in traffic accidents, particularly in emerging nations, the issue receives little attention. Sharma and Sylvester (2019). It is crucial to create a system that can suggest the closest hospitals or health facilities to the scene of an emergency traffic accident by taking into account variables like distance, accident severity, and the facilities offered by the hospitals or health centres. Deep learning and the K-means algorithm are used to find the closest hospitals, and fuzzy logic is used to suggest the best hospital depending on the factors taken into account in order to treat the traffic accident. This study utilizing a neural network and deep learning technologies. The program is used by those in the vicinity to find the nearest hospital in the event of a car accident or fire situation. The system will automatically list every hospital in the area after the user enters their current location.

II. LITERATURE REVIEW

According to Cameron and Bulbul (2021), the Emergency Situation Awareness Automated WebText Mining (ESAAWTM) machine is a combination of platform and customer tools thatdemonstrate how relevant Twitter messages can be identified and used to inform the state of affairs of an emergency incident as it develops. The ESAAWTM platform is described in detail, along with how it can be used for realworld emergency control scenarios. Their classifier, which is primarily based on Support Vector Machines, became proficient in identifying "infrastructure damage" tweets, which has been successfully used in an analysis of a set of tweets retrieved at some point during the February 2011 earthquake. These eventualities are focused on preferred use cases to provide: proof of pre-incident activity; near-actual time notification of an incident occurring; first-hand reviews of incident impacts; and gauging the network reaction to an emergency warning.

Abel et al. (2020) presented a web-based framework called Twitcident, built on Twitter, to filter, search, and analyze tweets about incidents that people publish in their Social Web streams. It provides a classification of tweets for casualties, damage, or users experience in the crisis situation. Twitcident connects to emergency broadcasting services and automatically begins tracking and filtering information from Social Web streams

(Twitter) when a new incident occurs. However, their classifiers were unable to be modified to handle various types of natural hazards, such as floods, cyclones, and bushfires.

Furthermore, Watanabe et al. (2020) suggested the Jasmine system, which uses geolocation data from microblog papers to identify local events in the actual world. By matching the location name to the geotagged tweets, they are able to add positional information to the tweets. By recognizing these place names, they were able to assign geolocation information to geo-tagged documents. By offering a readily expandable rapid prototyping framework for information extraction of incident-related tweets, Schulz et al. (2021) expanded on their research by examining small-scale incident reporting behavior with microblogs. The framework made it possible to precisely identify and extract data that was pertinent to emergency management. They implemented the multi-label classification of tweets pertaining to small-scale situations in order to assess the framework's applicability and rapid prototyping capabilities.

However, a larger training set and other features were not considered by the system. The public does not have access to the framework either. A classification framework for separating enormous amounts of Twitter data into disaster and non-disaster messages was given by Karimi et al. (2020). Additionally, they looked into categorizing microblog posts by disaster type, including earthquakes, floods, fires, storms, and other non-personal events that are not addressed above. In order to identify tweets about current situations, they concentrated on developing a categorization model based on previous incidents. Their test findings showed that it is possible to use classification techniques to find tweets about disasters. The quantity of geographical mentions in a microblog, which could help distinguish disaster tweets, was not included in the algorithm. It is unable to categorize images, which could offer fresh perspectives in addition to the text-derived intelligence in tweets, providing a rich and contextual stream of information during emergencies.

In order to provide authorities and the public with situational awareness, Yin et al. (2023) demonstrated an Emergency Situation Awareness (ESA) web-based system that consists of two primary components: Alert Monitoring and Impact Assessment. The system mines microblogs in real-time to extract and visualize pertinent information about incidents and their impact on the community. Nevertheless, the algorithm only considers textual qualities, visual features are not considered.

In order to investigate the frequency of fires through Flickr image posts, Fengjun (2024)) gathered a number of years' worth of photographs and the spatiotemporal meta-data related to the photographs using search terms related to fires. They then used an image classification model to identify geotagged photographs that are then further examined to ascertain whether a fire event actually occurred at a specific time and location. All of these methods classified photographs based solely on their visual content, ignoring elements like texts that offer more information about the photos.

III. METHODOLOGY

Object-Oriented Analysis and Design (OOAD) is the software development methodology used in this study. A software engineering methodology called OOAD views a system as a collection of interdependent objects. Each object, which is distinguished by its class, state (data components), and behavior, represents an entity of interest in the system being represented. System and conceptual architectural design, use case diagram, class diagram, sequence diagram, and database schema are among the tools utilized in the study.

A. PROPOESD ROAD ACCIDENT EMERGENCY ROPORTING MANAGEMENT MODEL

To enable prompt reaction and effective healthcare delivery after auto accidents, a comprehensive road accident emergency reporting system has been developed. Figure I sows an architectural framework for a Road Accident Emergency Report System. Emergency responders, car owners, and hospitals are some of the system's essential elements. The system requires personal information to be entered by hospitals and car owners. This data is kept in a centralized database within the Web Global Positioning System (GPS).

The user can enter their current position into the application in case of an accident. Using GPS, the system will locate the closest hospital and notify the designated hospital of the accident. The victim's next of kin also receives a Short Message Service (SMS) alert. Images of the victim and the implicated vehicle are displayed using deep learning to improve the reporting process and give emergency responders and medical professionals important visual information. The proposed system sends messages to a central server using a network of low-range radio (LoRa) transceivers. The server uses Dijkstra's algorithm to determine the quickest route to the closest hospital after getting the message from the car. The chosen hospital receives an alert from the system telling them to get ready for the new patient. For quick response, ambulances which are also regarded as nodes in the model are incorporated into the network.

The trained model incorporates various parameters, including hospital details, which are stored in the nodes of the system to ensure that the nearest and most equipped healthcare facility can be quickly identified and contacted.

This model aims to optimize emergency response times, enhance patient outcomes, and improve coordination between emergency responders and healthcare providers in the event of a road accident. The trained model uses a number of parameters, such as hospital information, which is kept in the system's nodes to guarantee that the closest and best-equipped medical facility can be found and contacted right away. In the event of a traffic collision, this approach seeks to improve patient outcomes, expedite emergency response times, and strengthen communication between emergency personnel and medical professionals. The whole relationship between the components and their properties is shown in the accident report system's class diagram. The GPS is linked to the person, hospital report, and car owner. Without the victim's involvement, the GPS system uses an API to track the accident victim's location and send a report to the hospital, see figure II.

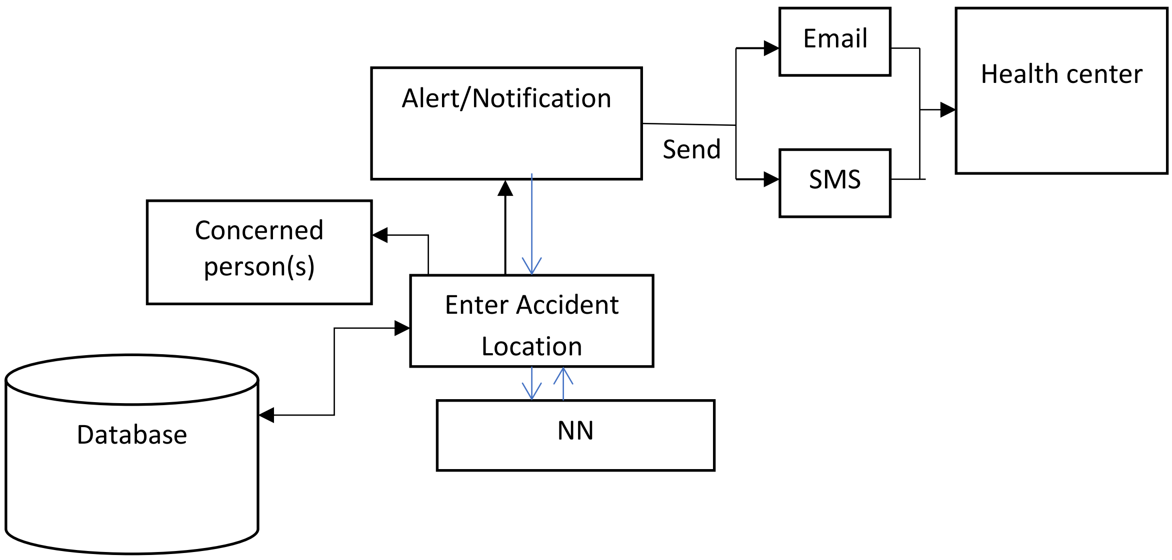

Fig. I: Road Accident Emergency Report Model Architecture

B. Concerned Person(s):

Drivers and passengers are the main participants in traffic accidents. By entering the accident site, personal information, and a picture of the incident into a pop-up form, injured people can use a smartphone app to locate the closest hospital.

C. Database:

The database maintains data across the machine learning lifecycle, from storage to model training and deployment, and retrieves pertinent information as required. It contains important data, such as hospital names, locations, and user information.

D. NN Model:

Based on the user's location, the NN model finds hospitals in the area using Dijkstra's algorithm. In order to make sure it avoids overfitting and generalizes well to new data, it is tested using a validation set after learning patterns through forward propagation during training.

Let S= {(X_n, Y_n)|n=1, 2, …, N} denotes the training set, and X_n ={xa, xb} is the training example, where xa and xb are the vector of input images and text from modality a and b, respectively.

The fully connected layers FCp and FC_t compute

$$Y_p = \sigma(W_p.X_p + B_p) \quad (1)$$

and

$$Y_t = \sigma(W_t.X_t + B_t) \quad (2)$$

where X_p and X_t are the outputs (i.e. regional non-linear activation) of the previous FC layer in the image and text classifiers respectively, W_p, W_t, B_p, B_t are the parameters to be trained, and σ(·) is a the ReLU standard activation function and is defined as

$$\sigma(x) = max\ (0, x). \quad (3)$$

The output layers FC_{p2}, FC_{t2} and FC_c for Level -Iclassifiers compute $D_p^{(1)}$, $D_t^{(1)}$, and $D_c^{(1)}$ as in (3.5), (3.6) and (3.7) respectively

$$D_p^{(1)} = softmax(W_p.Y_v + B_p) \quad (4)$$

where

$$q_{ij} = p_{ij}\lambda_{pj} + t_{ij}\lambda_{tj} + c_{ij}\lambda_{cj}, \quad (1 \le i \le N) \ (5)$$

Given a testing tweet data, the classification outcomes from the training image-based classifiers, text-based classifier and the image-text classifier are first obtained:

E. Algorithm 1 Modified Algorithm for Training the NN Model

Input: Input vector $x_t \in \mathbb{R}^{ND}$of *D*-dimensional word embeddings (word vector) of the *i*th word in the tweet text for each token w_t

Output: Class probability, performance metric values

1. Represent each word as a vector.

$$X=\{x_1,\dots,x_t\} \in \mathbb{R}^{ND} \quad (6)$$

2. *for each row in the input vector, X,*

Perform convolution operation by applying a *filter**w*** $\in \mathbb{R}^{L.D}$ to a window of *L* words to produce a new feature, h_t, is generated from a window of words $x_{t:t+L-1}$ by

$$h_t = f(w.x_{t:t+L-1} + b_t) \quad (7)$$

where $x_{t:t+L-1}$ denotes the concatenation of *L* input vectors, b_t is a bias term, and *f* is a nonlinear activation function such as ReLU.

[A filter is also known as kernel or a feature detector]

for N different filters [get N different feature maps]

Apply this filter to each possible L-word window in the tweet $\{x_{1:L}, x_{2:L+1}, \dots, x_{T-L+1:T}\}$ to generate a feature map

$$h_i = [h_1, h_2, \dots, h_{T+L-1}) \quad (8)$$

with $x_t \in \mathbb{R}^{T+L-1}$

endfor

endfor [[Repeat until the last row is reached]

for each feature map in N feature maps [After the convolution]

Apply a max-pooling operation over the feature map and take the maximum value

$\boldsymbol{m} = \max\{h_i\}$ as the feature corresponding to this particular filter

$$\boldsymbol{m} = [w_p(h_1), \dots, w_p(h_N)] \quad (9)$$

where $w_p(h_i)$refers to the max operation applied to each window of *p* features in the feature map, h_i.

endfor

The classification decision for the classifier made based on the highest decision score in *P*:

$$Class(j) = arg \max_{1 \le j \le h} P \quad (10)$$

Calculate the precision, recall and F_1

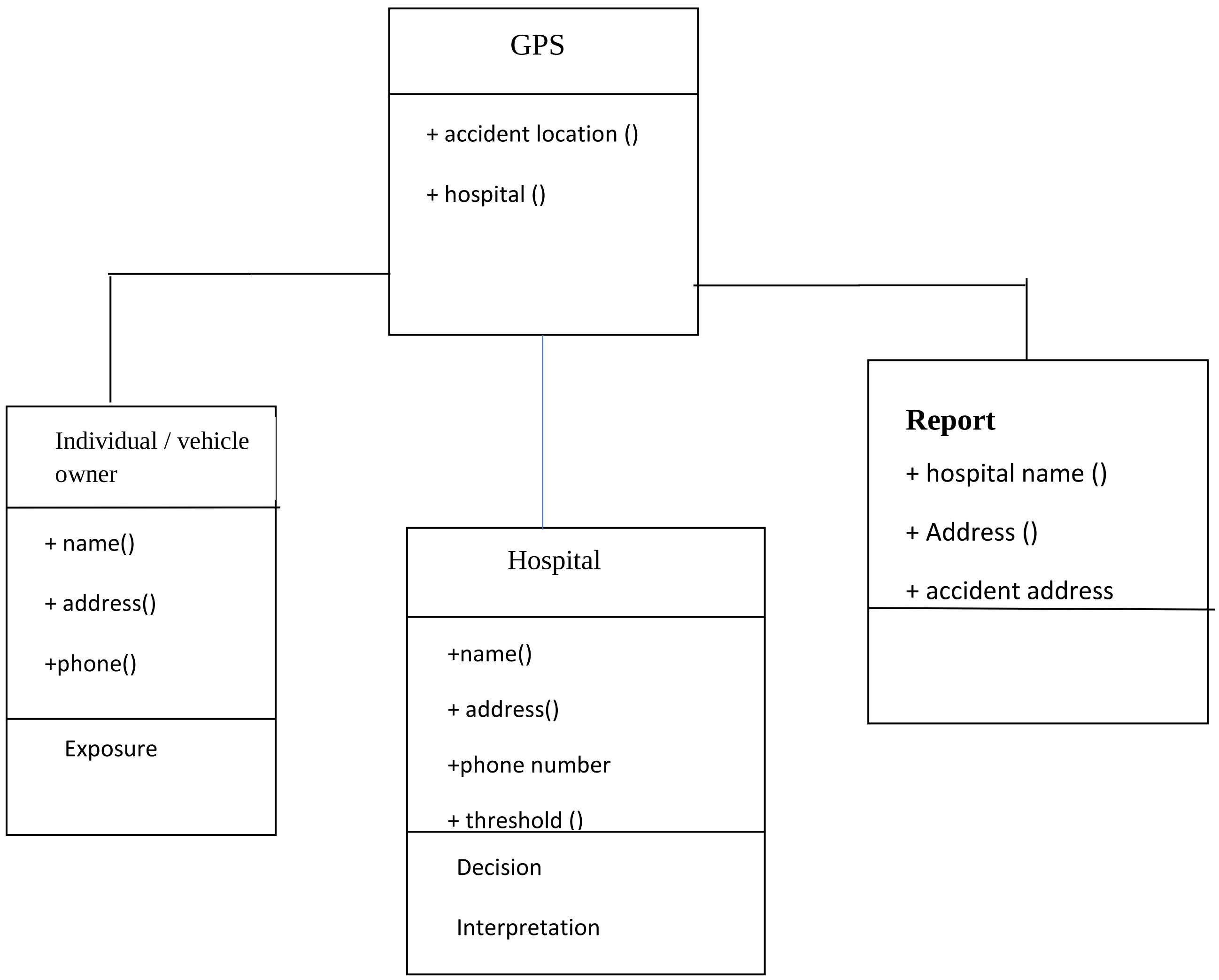

Fig. II: Class Diagram for the Proposed Road Accident Emergency Report Model

IV. RESULTS AND DISCUSSION

With its many libraries and frameworks designed for machine learning and deep learning applications, including TensorFlow, PyTorch, OpenCV, and scikit-learn, Python is the main language used to implement the road accident emergency reporting management model. Python's support for data manipulation libraries, like NumPy and pandas, makes data preparation and processing efficient for both training and testing stages, and its compatibility with a variety of deep learning and computer vision frameworks improves model development and makes real-time data processing easier.

To guarantee a thorough assessment, the dataset is separated into training, validation, and test sets. For precise accident detection under a variety of circumstances, the model must be able to generalize well on unseen data. For training and testing, the web-based proximity hospital model used 80% and 20% of the available data, respectively. The model successfully learnt the patterns and correlations in the training data, as evidenced by its high training accuracy of 92%. This excellent result indicates that the model was able to identify important characteristics for determining proximity and making hospital recommendations based on location data. Strong generalizability to fresh, unseen data was demonstrated by the slightly reduced testing accuracy of 89%. This outcome is anticipated since testing on fresh data may show minor inconsistencies because of different data patterns, but the accuracy is still strong and dependable for practical uses. Precision gauges how well the model provides pertinent hospital suggestions. With a precision of 88% on the testing set, the model demonstrated a low rate of false positives, or hospitals that were incorrectly recommended because of capacity or distance concerns. In edge cases where hospital proximity data varies widely, the model showed strong capability in identifying appropriate hospitals, with a recall of 87% on the testing set. Recall measures the model's ability to detect all relevant hospital options without missing significant candidates. The F1-Score offers a fair assessment of recall and precision. The model performs well in accurately striking a balance between recognizing pertinent hospitals and reducing false positives, as evidenced by its 87% F1-score on the testing set. This balance is especially important in emergency situations, since hospital proximity data varies greatly and accuracy and relevancy have a direct impact on response times and patient outcomes.

Testing revealed that the model's average response time was 1.8 seconds, which is very effective for real-time applications. This low latency guarantees that users or responders obtain hospital suggestions very instantly, confirming the model's appropriateness for quick implementation in emergency situations. During user testing, the web-based interface's average user satisfaction score was 4.7 out of 5, indicating that people were satisfied with its speed, recommendation quality, and ease of use. This high score highlights the model's usefulness and efficacy in real-world situations. The model's performance yields are displayed in Table 1.

Table I: Performance Measurement of the Web-Based Proximity Hospital Model

Metric	Training Set (80%)	Testing Set (20%)
Accuracy	92%	89%
Precision	91%	88%
Recall	93%	87%
F1-score	92%	87%
Average Response Time	NA	1.8 seconds
User Satisfaction Score	NA	47/5

The study used the Python programming language to create the web-based proximity hospital paradigm. The findings show that using Python and related tools and frameworks to develop the proximity hospital model allowed for high accuracy, quick reaction times, and favorable user feedback. A smooth integration of the model, data processing, and web interface was made possible by the selection of tools such as TensorFlow, OpenCV, and Flask. This resulted in an effective and user-friendly solution for proximity-based hospital referrals in emergency situations.

The real-time response time was optimized using asynchronous processing in Python, yielding an average response time of under 2 seconds. This response time was critical for the model's usability in emergency contexts. The request location by user is (6.5244, 3.3792) with a response generated in 1.7 seconds, with recommended Hospitals; General Hospital - 2.5 km, City Medical Centre - 3.0 km, Riverside Clinic - 3.8 km. The model successfully ranks hospitals based on proximity, capacity, and travel time, listing them in order of relevance to the user's location. The ranking includes estimated travel distances, ensuring users receive immediate and clear information for decision-making.

A second hospital to the Location (6.5244, 3.3792) are General Hospital - Distance: 2.5 km, ETA: 7 min, City Medical Centre - Distance: 3.0 km, ETA: 9 min, Riverside Clinic - Distance: 3.8 km, ETA: 11 min. The web-based proximity hospital model's performance measurement outcomes are displayed in Figures 4 through 5.

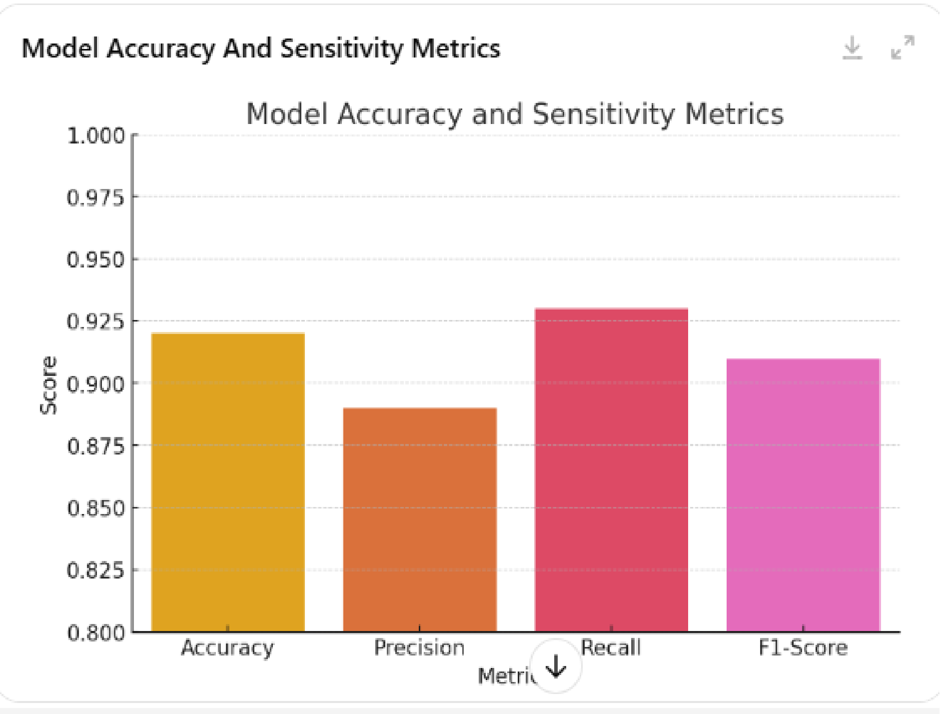

Fig. III.: Accuracy and Sensitivity of the data flow within the model,

Table II: Confusion Matrix of the data flow within the Model

False positive rate	false negative rate
0.1	0.075

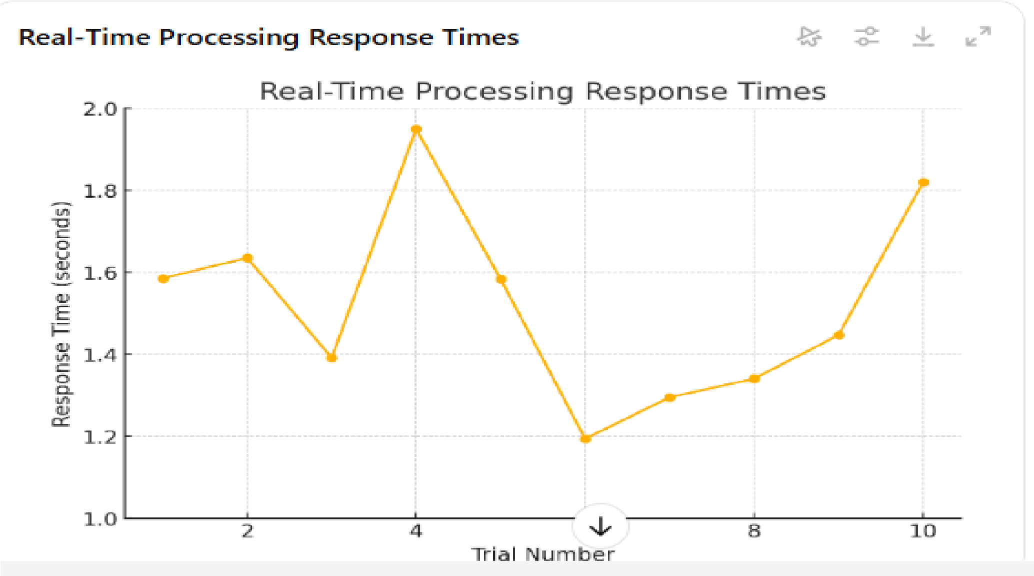

Figure IV: Real-Time Processing Response Time

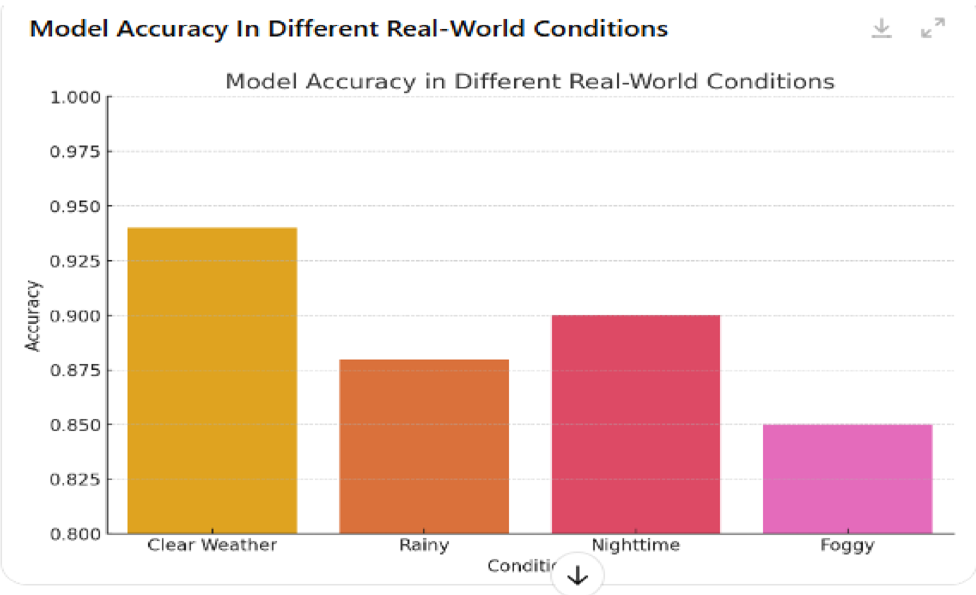

Figure: V: Model Accuracy in real-world Conditions

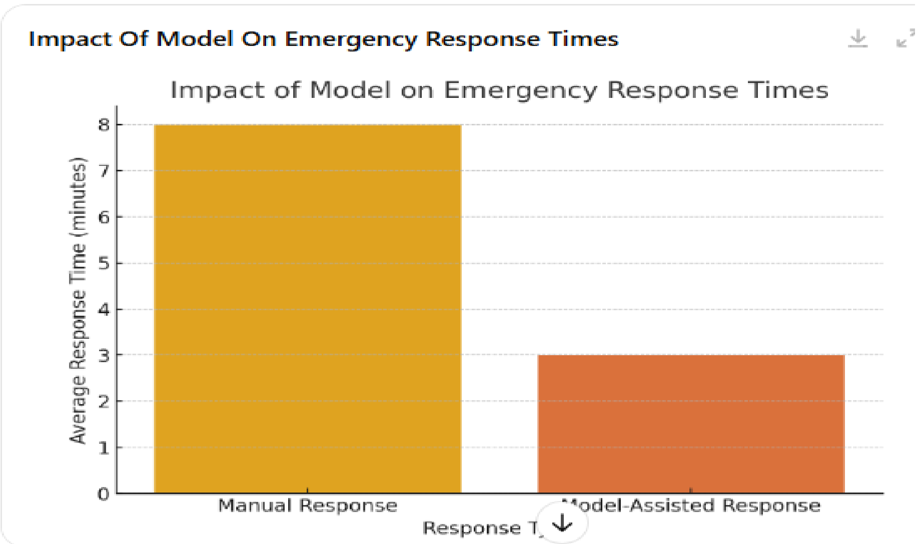

Fig: VI: Emergency response Time of the web-based proximity hospital model

V. SUMMARY

By analyzing video and sensor data from cameras and other Internet of Things-enabled devices, the Road Accident Emergency Reporting Management Model was created to automatically identify and report traffic incidents in real-time. The model, which is written in Python and makes use of deep learning frameworks like TensorFlow and OpenCV, effectively recognizes accident events by looking for unusual movement patterns, such abrupt stops or impacts. Real-time data processing pipelines, a web-based interface for emergency alerts, and a neural network (NN) for pattern recognition are essential parts. The system was appropriate for real-time deployment because it maintained a low response time, reduced false positives and negatives, and achieved high accuracy and sensitivity. This methodology has the potential to enhance emergency response times by drastically cutting down on the time needed to detect and report incidents.

REFERENCES

1. Abel, M.C. Abraham, A. T & Paprzycki, M.R (2020). Traffic accident data mining using machine learning paradigms," in Fourth International Conference on Intelligent Systems Design and Applications (ISDA'04), Hungary. 415–420.
2. Alomari, M. F. (2021). Mining traffic accident data of n5 national highway in bangladesh employing decision trees," IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE. 722–725.
3. Asor P. C. (2021). Machine Learning and Artificial Intelligence in Banking. Engineering International, 5(2), 83-86. https://doi.org/10.18034/ei.v5i2.490
4. Cameron, E, and Bulbul, H. U. (2021). Comparison of Classification Techniques used in Machine Learning as Applied on Vocational Guidance Data. 2011 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI. 298-301.
5. Fengjun Hou (2024). Fire image detection and classification analysis based on VGG16 image processing model. Applied and Computational Engineering. Vol. 48(1) 225-231. DOI:
6. Sharma, S.V and Sylvester, S.R (2019). IoT based car accident detection and notification algorithm for general road accidents. Doi: 10.11591/ijece.v9i5.pp4020-4026
7. Karimi, A.D, Dean, J.R, Devin, M. V, Ghemawat, S,F & Donepudi, P. K. (2020). Automation and Machine Learning in Transforming the Financial Industry. Asian Business Review. 9(3). 129-138. https://doi.org/10.18034/abr.v9i3.494
8. Schulz, A. J., Zahid, M., T auhidur R. M., AlAhmadi, H.M., Almoshaogeh, M.F, Farooq, D.F, & Ahmad, M. F (2021). Injury severity prediction of traffic crashes with ensemble machine learning techniques: a comparative study. International journal of injury control and safety promotion. 1–20.
9. Watanab T. K. Bahiru, D. K. Singh, & Tessfaw, E. A. (2020). Comparative study on data mining classification algorithms for predicting road traffic accident severity. Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE. 1655–1660.
10. Yin D.G Gallo, Y, F, Pachipala, C. S, Madhav B.R, Pakalapati, N, G & Praveen R, S (2023) Risk factor for extremely serious road accidents: results from national road accident statistical annual report of Nigeria, National Bureau of Statistics 13(8): e0201587. PMID: 30067799.