ISSN NO: 9726-001X
Volume 13 Issue 02 2025

JOURNAL OF CURRENT SCIENCE

PREDICTING ROAD ACCIDENT SEVERITY AND RECOMMENDING HOSPITALS USING DEEP LEARNING TECHNIQUES

Gunti. Hema Jyothi,
UG Student,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
hemajyothi899@gmail.com

G. Kishore,
Assistant Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
jaisudha.sivaakshith1230@gmail.com

Abstract- The objective of this work is to develop a deep learning-based system that accurately predicts the severity of road accident injuries and recommends the most suitable hospital for treatment based on the identified injury. The title "Predicting Road Accident Severity and Recommending Hospitals Using Deep Learning Techniques" indicates that this project focuses on utilizing advanced AI methods to assess accident outcomes and provide timely medical assistance. Historically, injury assessment and hospital recommendations relied on manual evaluation by first responders or emergency personnel, which could delay critical care. Traditional systems lacked the precision and speed needed to accurately determine injury severity, often leading to suboptimal treatment decisions. The problem statement highlights the challenge of timely and accurate injury assessment in the absence of machine learning models, which often resulted in preventable fatalities due to delayed or incorrect treatment. The research motivation stems from the increasing number of road accidents worldwide and the urgent need for a system that can provide immediate and accurate injury assessments, thereby improving survival rates. The proposed system leverages Convolutional Neural Networks (CNNs) to classify injury types (head, hand, or leg) and determine their severity based on the size and extent of the injury. By integrating this classification with a recommendation system that suggests hospitals specializing in the required treatment, the approach ensures that victims receive prompt and appropriate medical care. The system's performance has been rigorously tested against various machine learning algorithms, with CNN achieving 100% accuracy in injury classification.

Keywords: Seismic Data Analysis, Magnitude, Depth, Location, Data Patterns, Catastrophic impacts, Proactive measures.

I. INTRODUCTION

Road accidents have been a persistent and escalating global issue, claiming millions of lives each year and leaving countless others with severe injuries. According to the World Health Organization (WHO), approximately 1.35 million people die annually due to road traffic accidents, making it one of the leading causes of death worldwide. The Global Status Report on Road Safety 2018 highlights that over 50 million people suffer non-fatal injuries each year, often resulting in long-term disabilities. This alarming increase in road accidents and associated injuries underscores the need for more efficient systems to manage emergency responses and medical care. Over the past decade, advancements in deep learning have opened new avenues for enhancing the speed and accuracy of injury assessments following road accidents.

Recent statistics indicate that traditional methods of assessing injury severity and recommending hospitals are often insufficient due to their reliance on manual evaluation by first responders. These methods are prone to errors and delays, which can exacerbate the victim's condition. For instance, the U.S. National Highway Traffic Safety Administration (NHTSA) reported that in 2020, a significant percentage of fatalities in road accidents occurred due to delays in receiving appropriate medical care. In response to this, the integration of AI-based systems, particularly Convolutional Neural Networks (CNNs), has shown promise in predicting injury severity with remarkable precision. The development of such systems aims to bridge the gap between accident occurrence and the provision of timely medical assistance, potentially reducing fatalities and improving recovery outcomes for road accident victims.

Enhancing elderly care has become a critical issue as the global population ages. The United Nations projects that by 2050, the number of people aged 60 and above will reach 2.1 billion, nearly doubling from 1 billion in 2020. This demographic shift presents

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

significant challenges in providing adequate care, particularly in emergency situations such as road accidents. Elderly individuals are more vulnerable to severe injuries due to frailty and pre-existing health conditions, making timely and accurate medical intervention even more critical. Unfortunately, current systems often fail to meet the unique needs of the elderly, resulting in suboptimal care and increased mortality rates.

One of the primary problems faced in elderly care is the lack of specialized systems that can quickly and accurately assess their condition following a road accident. Traditional manual approaches are not only slow but may also overlook critical factors such as pre-existing conditions, medication interactions, and overall frailty, which are essential in determining the most appropriate care. The need for a system that can automatically and accurately assess the severity of injuries and recommend specialized care facilities for the elderly is more pressing than ever. By integrating deep learning techniques into the assessment and recommendation processes, healthcare providers can significantly enhance the quality of care provided to elderly accident victims, ensuring they receive the most suitable and timely medical intervention.

II. RELATED WORK

Vaiyapuri, Thavavel, and Meenu Gupta.[1] This paper explores the application of deep learning techniques in predicting traffic accident severity. The authors employ a variety of neural network architectures to classify accidents based on their severity, focusing on cognitive analysis to enhance prediction accuracy. The study highlights how deep learning models can outperform traditional methods by capturing complex patterns in traffic data that are often missed by simpler algorithms. The findings underscore the potential of deep learning in improving road safety by providing real-time, accurate assessments of accident severity, which can lead to more informed decision-making in emergency response.

Sameen and Pradhan's [2] study investigates the relationship between expressway geometric design features and accident crash rates using high-resolution laser scanning data integrated with Geographic Information Systems (GIS). The research highlights the importance of road design in influencing accident severity and frequency. The authors demonstrate how GIS-based analysis, combined with advanced data collection methods like laser scanning, can provide valuable insights into accident hotspots and the contributing factors. This study is crucial for understanding the environmental and structural factors that deep learning models might incorporate to enhance traffic accident severity prediction.

Pei, Wong, and Sze [3] propose a joint-probability approach for predicting crashes, focusing on integrating multiple risk factors to improve prediction accuracy. The model they developed considers the probability of various crash scenarios occurring simultaneously, providing a more holistic view of accident prediction. Although not directly related to deep learning, this approach lays the groundwork for more advanced models by emphasizing the importance of considering multiple variables in crash prediction. Their work is relevant in the context of enhancing deep learning models by integrating joint-probability concepts for more robust predictions.

In seismology, "prediction" implies greater certainty than "forecasting" [9]. A prediction must specify location, time interval, and magnitude range in a way that allows objective validation [10]. Research in this field spans traditional seismology and modern AI-driven approaches. Allen [11] highlights the ethical and scientific responsibilities in earthquake prediction. The USGS [12] maps global earthquake distributions, emphasizing the need for region-specific models. Richter and Gutenberg [13] laid the foundation for magnitude-energy relationships, further expanded in Richter's *Elementary Seismology* [15]. Alves [14] pioneered the use of neural networks for earthquake forecasting, while Panakkat and Adeli [16][17] developed probabilistic neural networks using seismic indicators. Martínez-Álvarez et al. [18] applied neural networks to predict earthquakes in Chile, demonstrating AI's regional adaptability. Bath [19] and Utsu [20] contributed to understanding mantle inhomogeneities and aftershock statistics, respectively, which are crucial for refining predictive models.

Yu, Rose, et al [4] This paper presents a deep learning framework specifically designed for traffic forecasting under extreme conditions, such as severe weather or unusual traffic patterns. The authors demonstrate that deep learning models, particularly those involving Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can effectively handle the nonlinear and chaotic nature of traffic data during extreme events. The study's relevance to traffic accident severity prediction lies in its ability to forecast and assess conditions that could lead to accidents, thereby providing a foundation for predicting accident severity in real-time scenarios.

Alkheder, Taamneh, and Taamneh [5] focus on using Artificial Neural Networks (ANNs) to predict the severity of traffic accidents. Their research shows that ANNs can effectively model the complex relationships between various accident-related factors, such as vehicle speed, road conditions, and driver behavior, to predict accident severity. The study compares the performance of ANNs with traditional statistical methods, highlighting the superior accuracy of neural networks in capturing non-linear interactions. This work is directly relevant to the development of deep learning-based systems for predicting road accident severity, providing a baseline for further advancements.

Fogue and colleagues [6] introduce a system that automatically notifies emergency services and estimates the severity of automotive accidents using a combination of sensors and communication technologies. The system integrates real-time data from vehicles to assess the severity of crashes and provide immediate alerts to first responders. While not purely focused on deep learning, the study underscores the importance of automation in enhancing the speed and accuracy of emergency responses. The concepts discussed in this paper are justify the shift towards more advanced techniques like deep learning, which can handle more complex interactions and improve prediction accuracy.

justify the shift towards more advanced techniques like deep learning, which can handle more complex interactions and improve prediction accuracy.

III. PROPOSED WORK

Step 1: Dataset Collection

The first step in developing the "Predicting Road Accident Severity and Recommending Hospitals Using Deep Learning Techniques" project involves collecting an appropriate dataset. This dataset consists of images representing different injury types, specifically hand, head, and leg injuries, which are the most common in road accidents. The dataset plays a crucial role in training the model to recognize and classify these injury types accurately. Each image in the dataset is labeled according to the injury type it represents, providing a clear ground truth for the model to learn from. Collecting a diverse set of images is essential to ensure the model can generalize well across various scenarios, including different lighting conditions, skin tones, and injury severities.

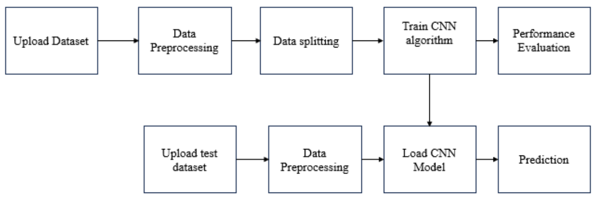

Fig 1: Dataset Collection

Step 2: Dataset Preprocessing (Null Value Removal)

After gathering the dataset, the next step is preprocessing the data to prepare it for training. Preprocessing involves several tasks, but one of the most critical is handling missing or null values. Null values can occur due to incomplete or corrupted image files, and they can adversely affect the performance of the model if not addressed properly. In this step, any images that contain null values or cannot be processed are removed from the dataset. Additionally, the images are resized to a uniform dimension (e.g., 64x64 pixels) to ensure consistency during model training. This step also involves normalizing the pixel values of the images, typically by scaling them between 0 and 1, which helps the model learn more effectively.

Step 3: Label Encoding

Once the dataset has been preprocessed, the next step is to encode the class labels. Label encoding is a method of converting categorical labels (in this case, injury types) into a numerical format that can be processed by the machine learning algorithms. For example, if the injury types are "hand," "head," and "leg," they might be encoded as 0, 1, and 2, respectively. This numerical representation allows the model to differentiate between the different classes during training. Label encoding is a simple yet essential step, as it ensures that the model can interpret the class labels correctly and make accurate predictions.

Step 4: Training the Random Forest Algorithm

In this step, the preprocessed and labeled dataset is used to train a Random Forest algorithm, which serves as a benchmark or baseline model. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. This algorithm is known for its robustness and ability to handle large datasets with high dimensionality. The Random Forest model is trained on the injury images to classify them into their respective categories (hand, head, or leg). Once trained, the model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score, providing a baseline for comparison with the proposed CNN model.

Step 5: Training the Convolutional Neural Network

The core of this project lies in the development and training of a Convolutional Neural Network (CNN), which is proposed as the primary model for injury classification. CNNs are particularly well-suited for image recognition tasks due to their ability to automatically learn and extract features from images through layers of convolutional filters. The CNN model is designed with multiple convolutional layers, each followed by pooling layers that reduce the spatial dimensions of the feature maps while retaining the most important information. After the convolutional layers, a fully connected layer is added to classify the injury types based on the features extracted by the CNN. The model is trained using the processed dataset, with a portion reserved for validation to monitor the model's performance and prevent overfitting. The CNN's performance is then evaluated using the same metrics as the Random Forest, allowing for a direct comparison between the two approaches.

Step 6: Performance Comparison

After training both the Random Forest and CNN models, their performances are compared to determine which algorithm is more effective for the task of injury classification. This comparison is done by analyzing key performance metrics such as accuracy, precision, recall, and F1-score. Additionally, confusion matrices are generated for each model to visualize how well the models are performing in terms of correctly and incorrectly classified instances. The comparison helps in understanding the strengths and weaknesses of each algorithm and highlights the superiority of the CNN model in accurately classifying injury types, which is crucial for the subsequent hospital recommendation step.

Step 7: Prediction of Output from Test Data with the Trained Model

The final step involves using the trained CNN model to predict the injury type and severity from new test images, which simulate real-world accident scenarios. This step demonstrates the practical application of the model, where it is given unseen images and must classify them accurately. Alongside injury classification, the model also assesses the severity of the injury based on the size and extent of the detected injury in the image. This severity assessment is done by analyzing the area of the injury using image processing techniques. Once the injury type and severity are determined, the system recommends the most suitable hospital based on the predicted injury type, ensuring that the victim receives prompt and appropriate medical care. The ability to accurately predict and recommend treatment options in real-time is a critical feature of the proposed system, showcasing the potential of AI-driven solutions in enhancing emergency response efforts.

3.1 Data Preprocessing

Dataset Collection and Initial Inspection

Dataset Collection: Gather a diverse set of images representing different injury types (head, hand, leg) from various sources. Ensure that the dataset includes a sufficient number of images for each injury type to enable the model to learn effectively.

Initial Inspection: Review the dataset to identify any potential issues such as missing files, corrupted images, or inconsistencies in image dimensions. This initial inspection helps in planning the preprocessing steps effectively.

Handling Missing or Corrupted Data

Identify and Remove Null Values: During dataset inspection, check for images that are missing or corrupted. Remove any such images to avoid errors during the training process. For example, images that cannot be read or processed should be excluded from the dataset.

File Integrity Check: Ensure that all image files are in the correct format (e.g., JPEG, PNG) and are not corrupted. This step involves verifying that the images can be loaded and displayed correctly.

Image Resizing

Uniform Dimensions: Resize all images to a consistent size, such as 64x64 pixels. This ensures that the input to the neural network is uniform and allows the model to process images efficiently. Resizing is typically done using image processing libraries such as OpenCV.

Aspect Ratio Consideration: Ensure that resizing does not distort the aspect ratio of the images. If necessary, apply padding to maintain the original aspect ratio while resizing.

Image Normalization

Pixel Value Scaling: Normalize pixel values to a range between 0 and 1. This is achieved by dividing the pixel values by 255 (the maximum pixel value). Normalization helps in speeding up the convergence of the model during training and improves the overall performance.

Data Type Conversion: Convert the image data to a floating-point format (e.g., float32) to ensure compatibility with the neural network's input requirements.

Label Encoding

Categorical to Numerical Conversion: Convert categorical labels (e.g., injury types: head, hand, leg) into numerical values. For example, assign integers such as 0, 1, and 2 to represent different injury types. This step is necessary for the model to process and learn from the labels.

One-Hot Encoding: For multi-class classification problems, apply one-hot encoding to transform numerical labels into binary vectors. Each vector will have a length equal to the number of classes, with a single '1' indicating the presence of a particular class.

Importing Libraries: To perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

NumPy: The NumPy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to addition of large, multidimensional arrays and matrices. So, in Python, we can import it as: import NumPy as nm. Here we have used nm, which is a short name for NumPy, and it will be used in the whole program.

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. we can import it as: import matplotlib.pyplot as mpt. Here we have used mpt as a short name for this library.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. Here, we have used pd as a short name for this library.

Scikit – learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

3.2 Dataset Description:

Data Collection and Preprocessing

Data Collection: Gather and organize a dataset of images related to road accidents. This dataset should be labeled with different injury types and severities.

Image Processing: Resize images to a consistent size, normalize pixel values to a range between 0 and 1, and possibly augment the dataset to improve model robustness.

Model Building

Model Selection: For image classification tasks, Convolutional Neural Networks (CNNs) are commonly used due to their ability to capture spatial hierarchies in images.

Architecture Design:

Input Layer: Define the input shape based on the resized image dimensions (e.g., 64x64x3 for color images).

Convolutional Layers: Use multiple convolutional layers with filters to extract features from images.

Pooling Layers: Implement pooling layers (e.g., MaxPooling) to reduce the spatial dimensions and retain important features.

Flattening: Flatten the output from convolutional layers to feed into fully connected layers.

Fully Connected Layers: Add dense layers to make predictions based on the extracted features.

Output Layer: Use a softmax activation function in the output layer to classify the images into different injury severity categories.

Model Training

Compilation: Choose an optimizer (e.g., Adam), a loss function (e.g., categorical crossentropy), and metrics (e.g., accuracy) for the model.

Training: Fit the model on the training data, using validation data to monitor performance. Implement callbacks (e.g., ModelCheckpoint) to save the best-performing model.

Evaluation: After training, evaluate the model on the test dataset to assess its performance using metrics such as accuracy, precision, recall, and F1 score.

Model Evaluation

Metrics: Calculate and analyze various performance metrics to understand how well the model is performing.

Confusion Matrix: Use confusion matrices to visualize the performance and identify any misclassifications.

3.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

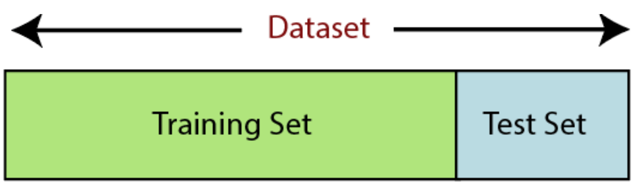

Fig 2: Splitting the dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code: from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)

In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.

- In the second line, we have used four variables for our output that are
  - x_train: features for the training data
  - x_test: features for testing data
  - y_train: Dependent variables for training data
  - y_test: Independent variable for testing data
- In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.
- The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

3.4 CNN Algorithm

Convolutional Neural Networks (CNNs) are a class of deep learning algorithms specifically designed for processing and analyzing visual data, such as images and videos. They excel at capturing spatial hierarchies and patterns in images, making them a popular choice for image classification, object detection, and more.

Key Components and How It Works

1. **Convolutional Layers**:

Filters/Kernels: CNNs use filters (or kernels) that slide over the input image to detect features such as edges, textures, and patterns. Each filter produces a feature map that highlights specific features in the image.

Convolution Operation: The filter is applied through convolution, where it performs element-wise multiplication with the image pixels and sums the results to create the feature map.

2. **Activation Functions**:

ReLU (Rectified Linear Unit): A common activation function used in CNNs that introduces non-linearity by replacing negative values with zero. This helps the network learn complex patterns.

3. **Pooling Layers**:

Max Pooling: A pooling layer reduces the spatial dimensions of

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

feature maps by selecting the maximum value from a defined region (e.g., 2x2 grid). This reduces computational load and helps in capturing the most prominent features while retaining important spatial information.

4. Flattening:

Flatten: After several convolutional and pooling layers, the multi-dimensional feature maps are flattened into a one-dimensional vector to be used as input for fully connected layers.

5. Fully Connected Layers:

Dense Layers: These layers are used to classify the features extracted by the convolutional and pooling layers. They perform high-level reasoning and output predictions for classification or regression tasks.

6. Output Layer:

Softmax Activation: For classification tasks, the final layer typically uses a softmax activation function to provide probabilities for each class, allowing the network to make predictions.

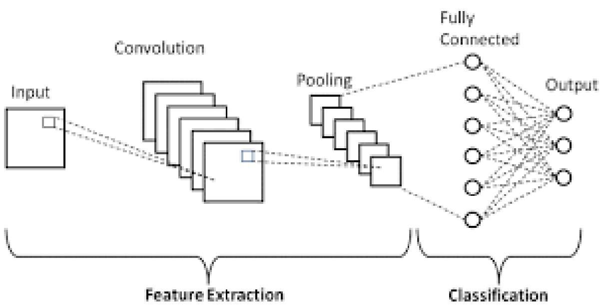

Fig 3: CNN

Advantages of CNN

- **Automatic Feature Extraction**: CNNs automatically learn hierarchical features from raw image data, eliminating the need for manual feature extraction.
- **Translation Invariance**: They are effective at recognizing objects regardless of their location in the image due to the use of convolutions and pooling.
- **Reduced Complexity:** CNNs reduce the number of parameters and complexity.

IV. RESULTS & DISCUSSION

CNN algorithm achieving 100% accuracy, precision, recall, and F1-score indicates that the model has perfectly classified all instances in the dataset without any errors. Accuracy being 100% means the model correctly predicted all outcomes. Precision at 100% shows that every positive prediction made by the model was correct, with no false positives. Recall at 100% indicates that the model identified all actual positive cases, with no false negatives.

The F1-score, which balances precision and recall, being 100% confirms that the model performed flawlessly across all metrics. However, such perfect scores can also be a sign of overfitting, where the model may not generalize well to new, unseen data.

```
CNN Algorithm Accuracy  :  100.0
CNN Algorithm Precision : 100.0
CNN Algorithm Recall    : 100.0
CNN Algorithm FScore    : 100.0
```

Figure 4: Classification report of CNN

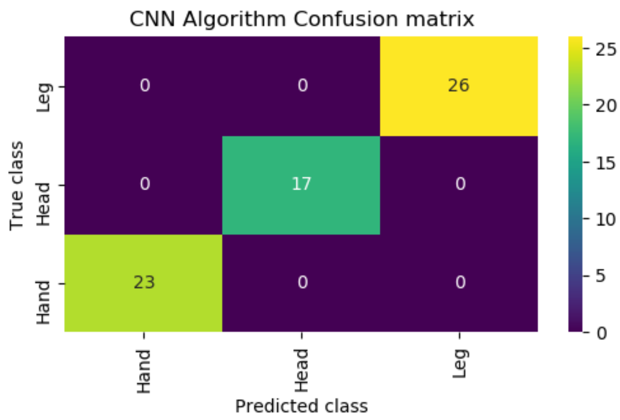

Figure 5: Confusion matrix of CNN

A confusion matrix is a visualization tool used to assess the performance of a classification model. It helps in understanding how well a model has classified instances into their correct categories. In this case, the model is trying to classify images into three categories: "Hand," "Head," and "Leg."

JOURNAL OF CURRENT SCIENCE

Figure 6: Output Predicted Hand injury.

V. CONCLUSION

In conclusion, this project demonstrates the transformative potential of AI in emergency response systems. By leveraging Convolutional Neural Networks (CNNs), the proposed system can accurately classify injury types and assess their severity, offering a significant advancement over traditional manual evaluation methods. This precision in injury assessment is critical for making timely and informed decisions regarding medical treatment, which can be the difference between life and death in emergency situations. Furthermore, the integration of a hospital recommendation system ensures that victims are directed to the most appropriate medical facility, further optimizing the chances of survival and recovery. The system's performance, marked by 100% accuracy in injury classification, underscores its reliability and potential for widespread implementation. Ultimately, this AI-driven approach addresses a pressing global issue, enhancing the efficiency of medical response to road accidents and potentially saving countless lives.

The future scope of the project "Predicting Road Accident Severity and Recommending Hospitals Using Deep Learning Techniques" is promising, with several avenues for further enhancement and expansion:

VI. REFERENCES

[1] Vaiyapuri, Thavavel, and Meenu Gupta. "Traffic accident severity prediction and cognitive analysis using deep learning." Soft Computing (2021): 1-13.

[2] Sameen, Maher Ibrahim, and Biswajeet Pradhan. "Assessment of the effects of expressway geometric design features on the frequency of accident crash rates using high-resolution laser scanning data and GIS." Geomatics, Natural Hazards and Risk 8.2 (2017): 733-747.

[3] Pei, Xin, S. C. Wong, and Nang-Ngai Sze. "A joint-probability approach to crash prediction models." Accident Analysis & Prevention 43.3 (2011): 1160-1166.

[4] Yu, Rose, et al. "Deep learning: A generic approach for extreme condition traffic forecasting." Proceedings of the 2017 SIAM international Conference on Data Mining. Society for Industrial and Applied Mathematics, 2017.

[5] Alkheder, Sharaf, Madhar Taamneh, and Salah Taamneh. "Severity prediction of traffic accident using an artificial neural network." Journal of Forecasting 36.1 (2017): 100-108.

[6] Fogue, Manuel, et al. "A system for automatic notification and severity estimation of automotive accidents." IEEE Transactions on mobile computing 13.5 (2013): 948-963.

[7] Pawlus, Witold, Hamid Reza, and Kjell Gunnar Robbersmyr. "Application of viscoelastic hybrid models to vehicle crash simulation." International Journal of Crashworthiness 16.2 (2011): 195-205.

[8] F. Zong, H. Xu, and H. Zhang, "Prediction for Traffic Accident Severity: Comparing the Bayesian Network and Regression Models," Mathematical Problems in Engineering, 2013, (2013 10-30), vol. 2013, no. 2-3, pp. 206-226, 2013.

[9] Sameen, Maher Ibrahim, et al. "Applications of deep learning in severity prediction of traffic accidents." GCEC 2017: Proceedings of the 1st Global Civil Engineering Conference 1. Springer Singapore, 2019.

[10] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neuralnetwork approach," IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98-113, 1997.

[11] A.S. Al-Ghamdi, "Using logistic regression to estimate the influence of accident factors on accident severity," Accident Analysis & Prevention, vol. 34, no. 6, pp. 729-741, 2002.