

# Real-Time Accident Detection Leveraging Deep Learning for Enhanced Road Safety

Nikila Annala

Department of Electronics and Communication  
Engineering(JNTUH)  
Institute of Aeronautical Engineering  
Hyderabad, India

T. Naga Nishith

Department of Electronics and communication  
Engineering(JNTUH)  
Institute of Aeronautical Engineering  
Hyderabad, India

Sahithi Chetti

Department of Electronics and communication  
Engineering(JNTUH)  
Institute of Aeronautical Engineering  
Hyderabad, India

V Kishen Ajay Kumar

Department of Electronics and  
Communication Engineering(JNTUH)  
Institute of Aeronautical Engineering  
Hyderabad, India

**Abstract**— Worldwide data show that a significant portion of violent deaths are caused by automobile accidents. The human component has a major impact on the time it takes to dispatch the medical response to the scene and is correlated with the likelihood of survival. For this reason, in addition to the widespread application of intelligent traffic systems and video monitoring, computer vision experts find it desirable to employ an automatic traffic accident detection method. These days, Deep Learning (DL)-based methods have demonstrated excellent results in computer vision challenges involving intricate feature relationships. In light of this, this work creates an automated DL-based technique for video traffic accident detection. The suggested approach makes the assumption that visual characteristics that appear over time describe traffic accident incidents. Consequently, a phase for extracting visual features was followed. Build the model architecture by identifying a transient pattern. Using both public and built-from-scratch datasets, convolution and recurrent layers are used in the training phase to learn the visual and temporal aspects. In public traffic accident datasets, an accuracy of 98% is attained in the detection of accidents, demonstrating a strong capacity in detection independent of the road structure.

**Keywords**— Traffic Accident Detection, Deep Learning, Video Surveillance, Surveillance Computer Vision, Convolutional Neural Networks(CNN), Recurrent Neural Networks.

## I. INTRODUCTION

Different things might lead to road accidents. The most frequent elements that raise the likelihood of them happening include speeding [3,4], the area's climate [2], the geometry of the road [1], and drunk drivers. Although the majority of these incidents merely result in material damage, they can still have a negative impact on the individuals involved and their quality of life in terms of personal safety and traffic mobility. Video cameras are now a tool for managing and regulating traffic in cities because of advancements in technology. They enable the analysis and observation of the movement of traffic inside the city [5]. Control is challenging, though, as the quantity of cameras required to

complete these jobs has been rising steadily over time if automation methods are not put in place since doing so will increase the number of specialists required to meet all the requirements. Numerous methods have been suggested to automate follow-up and control process tasks. A system that uses traffic camera surveillance is one example of this. With the aim of anticipating and regulating the frequency of traffic accidents in the region, these can be used to estimate the speeds and trajectories of the objects of interest [6]. Various methods have been proposed by the scientific community to identify road accidents [7]. These include machine learning, deep learning, social network data analysis, sensor data, and statistics-based techniques. The most recent methods have improved science across a number of domains, including video-based problem solving (video processing). In order to approach a solution for the identification and categorization of traffic accidents based on video, it is crucial to understand these techniques. Better results in the resolution of digital image processing problems have been obtained since the introduction of convolutional layers in the field of neural networks.

## A. Introduction to Deep Learning

In several problems, deep learning algorithms have demonstrated excellent performance, particularly in the areas of picture processing and comprehension. These layers take advantage of the spatial relationship that the input data have, which dense neural networks cannot accomplish because of the volume of information. One of the benefits of using convolutions on input data with lots of characteristics is that the curse of dimensionality can be circumvented. This is an extremely common issue when handling highly complicated data, like pictures. Similarly, it's critical to emphasize that using several convolutional layers aids in the extraction of pertinent visual information from a single dataset, which determines the network's performance. However, in certain cases, the geographical relationship between the data does not play a decisive role. The possible temporal relationship between the data is more significant in particular problems.

## II. LITERATURE SURVEY

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar: One crucial area of traffic behavior modeling that continues to be relevant is the detection of anomalies with normal aberration in vehicle scene entities [8]. Research in the fields of video analysis and anomaly identification has increased as a result of the availability of traffic video scenes [9]. While most computer vision models focus on analyzing generic traffic scenes and distinguishing between aberrant and normal traffic events, techniques like Sparse Reconstruction, Markov Random Field, and Markov model have had some success. However, the ability to identify traffic irregularities has significantly improved since the development of deep learning. Consequently, deep neural networks are used in the vast majority of studies to identify them. Faster R-CNN, a multi-granularity vehicle tracking approach with modularized elements, is used by Li et al. in their proposal deep learning framework to construct its module for object detection. Similarly, the object detector, background modeler, mask extractor, and tracker comprise its modularized element. Their technique improved anomaly prediction outcomes by utilizing both box and pixel-level tracking strategies. The 2019 AI City Challenge winning solution served as the model for the pixel level tracking. It goes without saying that using both of those tactics in conjunction with the backtracking optimization technique enabled the team to place first in the 2020 NVIDIA AI City Challenge anomaly detection track. With a few exceptions that concentrate on unsupervised approaches, the majority of anomaly detection techniques are typically supervised. An unsupervised anomaly detection framework using data gathered from vehicle trajectories was presented by Zhao et al. Their approach produced better outcomes when implemented a multi-object tracker to lessen the impact of the detector's false positives. Roadside accidents and halted vehicles were among the traffic abnormalities that Mandal et al. in utilized a feature tracker and a pre-trained YOLO network to identify. Using the K-means clustering technique and nearest neighbors, an anomaly detection system uses a YOLO-based object detector in conjunction with post-processing modules to anticipate stationary automobiles. Better performances might have been obtained through training on anomalous traffic video feeds, even if their nearest neighbor and clustering technique required a lot of training. An anomaly detection system containing a background modeller, perspective detection module, and spatial temporal matrix discriminating module was proposed by Bai et al. The study's use of the spatial temporal matrix module changed how the analysis of Integrating trajectory into the investigation of spatial position, providing precise start and stop times as well as an enhanced anomaly detection score that helped the team place first on the 2019 NVIDIA AI City Challenge at the study's use the leader board. The authors of the current study used a cutting-edge YOLO object identification system and concentrated on using post-processing modules as a more heuristic technique to identify abnormalities. Our proposed approach avoids the use of a tracker, unlike some studies that

use vehicle tracking algorithms, especially since tracking individual vehicles would have been challenging and computationally impractical given the obvious majority of vehicles in a traffic scene. It is important to note that the majority of the winning teams in the NVIDIA AI City Challenge for 2018–2020 placed a strong emphasis on enhancing vehicle recognition and background picture segmentation. In addition to a few post-processing modules. Our strategy, which was motivated by these previous solutions, offers a straightforward yet effective framework for road segmentation and background estimation. Using data from detections on foreground and background images, a decision tree method is also used to characterize abnormalities.

## III. EXSISTING SYSTEM

### A. In – vehicle crash detection systems:

These systems are built into cars in order to recognize and react to collisions. Usually, they employ sensors like GPS, gyroscopes, and accelerometers to track the movement of the car and identify any abrupt changes that could be signs of an accident. The system automatically initiates an alarm when it detects an accident, which may include airbag deployment, emergency light activation, or messages to emergency services or pre-designated contacts.

### B. Smartphone Applications:

By leveraging integrated sensors like accelerometers, GPS, and microphones, mobile applications can transform cellphones into accident detection tools. These applications track the mobility and surroundings of the smartphone to identify incidents such as collisions, trips and falls, or medical crises. The app has the ability to automatically send distress signals, share location data, and start emergency calls to emergency services or the pre-identified contacts in the smartphone. targeting the physical infrastructure despite Intelligent Transportation Systems' (ITS) capacity to identify traffic jams, accidents, and other incidents, among many other things. Different scientists and researchers have put out a number of approaches for the automatic identification of accidents. Among the methods they employ are mobile applications, GPS and GSM technologies, mobile applications, VANET (ad hoc networks), and smart phone detection of traffic accidents and their location in smartphone applications as an exsisting system.

Limitations: Particularly the sensors, the gear isn't usually that dependable. It could take some time for the GSM Module to transmit a message. When an accident occurs, the hardware consists of a vibration sensor that detects vibrations. A built-in GSM module then retrieves the GPS module and uses it to communicate the position of the incident as a message to the specified cell number. All of the sensors connected to a central Arduino board, which controls the entire system, are displayed by the GPS and the GSM modules that controls overall system.

#### IV.METHODOLOGY

**Data Collection:** Compile a sizable collection of accident photos. These pictures could include and exclude accident shots. **Preprocessing of the data:** resize images to a standard size appropriate for CNN input. Adjust pixel to a standard scale. Expand the dataset to boost its quantity and variety. Techniques like rotation, flipping, and cropping may be used for this. **Data Labelling:** each picture has a label based on the class it belongs to (accident no accident). **Splitting the data into training, validation, and testing sets** is known as data splitting. An 80-25 split could be typical. **Model Architecture Selection:** For image classification tasks, the CNN sequential model architecture is employed. **Model training:** set random weights at the beginning of the selected CNN architecture. Utilizing the training dataset, train the model. Make use of optimization strategies such as backpropagation and mini-batch gradient descent. To avoid overfitting, keep an eye on the model's performance on the validation. **User Module:** By using this module, authorities or users can log in to the program and access the details of an accident, including the location, photo, and map. Ambulances, hospitals, and police stations use this module to get to a destination quickly. **System Module:** This module loads a trained module, captures live video from a camera, converts it to frames, sends it to a model to determine whether an accident will occur, beeps, and takes a picture to transmit to a user module for display on a webpage.

#### CNN ALGORITHM

One kind of deep neural network that is very useful for processing organised, grid-like input, like photographs, is a Convolutional Neural Network (CNN) sequential model. TensorFlow/Keras' sequential model is a linear stack of layers, where each layer processes the incoming data according to a predetermined method before passing the result to the layer above it.

**Input Layer:** The raw input data is received by the input layer, usually as multidimensional arrays of pixel values representing images.

Convolutional layers are in charge of figuring out the spatial hierarchies of the patterns found in the input data. In order to extract features from the input image, each convolutional layer applies a set of learnable filters, commonly referred to as kernels. Backpropagation is used during training to modify the filters in order to acquire features pertinent to the task (e.g., detecting edges, textures or higher-level features). Convolutional layers output is subjected to activation functions (such as RELU) to create non-linearity and help the network recognize intricate patterns.

**Layers for pooling:** Using down sampling, layer pooling reduces the spatial dimensions of feature maps produced by convolutional layers while preserving crucial information by layers for pooling. Max pooling and average pooling are popular pooling operations that choose the maximum or average value within a given feature map region, respectively. by pooling the input data, the network's computational cost can be decreased and the learnt features become more resilient to slight distortions and translations.

**Flattening:** The feature maps are compressed into a one-dimensional vector after going through a number of convolutional and pooling layers. The process of flattening transforms the feature maps' spatial data into a format that can be sent to the fully connected layers.

Dense or fully connected layers process the flattened feature vectors and use the learnt features to make high-level decisions and reasoning. The model can learn intricate correlations between features because every neurone in a fully connected layer is connected to every other neurone in the layer above it. To add non-linearity, activation functions such as ReLU are added to the fully linked layers' output. To stop overfitting, dropout layers can be introduced, which randomly remove a portion.

**Output layer:** The CNN model's output layer, which comes in last, is in charge of generating the model's predictions. The output layer may have one or more neurones, each of which represents a class or a probability distribution across classes, depending on the job. Softmax and sigmoid are often used activation functions for the output layer in multi-class and binary classification, respectively.

**Loss Function and Optimisation:** Using a loss function (binary cross-entropy, for binary classification, for example), the model's predictions are compared to the ground truth labels during training. In order to minimise the loss function, an optimiser (such as Adam, SGD) modifies the model's weights and biases in accordance with the loss gradient that is calculated by backpropagation.

**Training:** Using a labelled dataset, the model is trained iteratively by modifying its parameters (weights and biases) in order to update the parameters, the complete dataset is run through the model both forward and backward over the course of several epochs of training. To keep an eye on the model's capacity for generalizing and to avoid overfitting, its performance is accessed during training on a different validation dataset.

**Evaluation:** To determine the model's accuracy and capacity for generalisation on fresh data, its performance is assessed on a test dataset that has not yet been observed.

**Prediction:** Following training and assessment, the model can be applied to forecast previously unobserved data.

## V. PROPOSED SYSTEM

A mechanism was implemented to identify mishaps in the recorded footage. This method makes use of several deep learning ideas. To train a model that could detect accidents in movies as needed, the Convolutional Neural Network with LSTM units was included. Convolutional Neural Network (CNN): Characterized by translation invariance and a shared-weights design. An example of a deep learning system specifically made for processing structured grid data, like photographs, is a convolutional neural network (CNN). In addition to being used for image identification, they have transformed the field of computer vision and are also useful for video analysis, natural language processing, and other purposes.

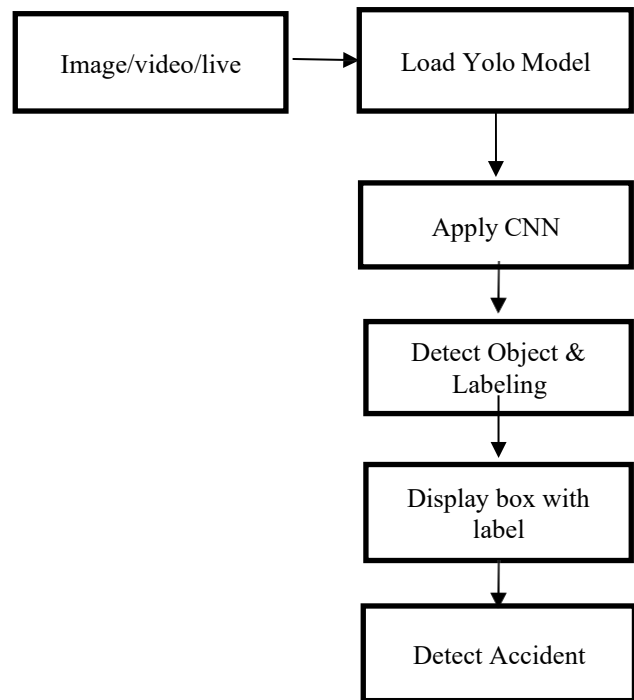


Figure 5.1: Flow chart of CNN

From fig 5.1 We can observe the flow chart of CNN algorithm the ResNet-50 model architecture is primarily implemented using the imageai open source library. A 50-layer deep convolutional neural network is called ResNet-50. It is possible to load a pre-trained version of the network that was trained on more than a million photos from the ImageNet database. The pretrained network is capable of classifying pictures into a thousand distinct item categories. The network has so acquired comprehensive feature representations for a wide range of images. Images having a resolution of 224 by 224 are supported by the network. Figure 2.1 above depicts the proposed design of ResNet. The datasets that were gathered are used to train this model. Example data sets and their corresponding class values are supplied into the model during training, which is carried out in a supervised learning manner. Model image frames from gathered data sets are utilized for learning; these frames are preprocessed to fit the model's size and pixel resolution.

**Advantages:**

Object Detection and Transfer System, or ODTs, is used in the suggested system. It tracks the live feeds by connecting to the inside

tunnel's live CCTV feeds. The live streams' segments are chopped before each object in the frame is identified. The system for tracking and detecting throughout time. It forecasts the further position and the grants ID based on the algorithm class and B-Box being reached the objective detection system. Accident detection and alarm systems improve road safety and save lives by drastically cutting down on emergency response times. They reduce false positives and negatives and increase accident detection accuracy. Congestion and subsequent accidents are less likely when traffic management systems are integrated. These devices also offer useful data for infrastructure development and urban planning. Moreover, they endorse the insurance industry by providing trustworthy accident records and supporting the investigation of fraud.

## VI. SOFTWARE REQUIREMENTS

Advanced accident detection and alerting systems can be developed with Python and deep learning algorithms. Because of its adaptability in managing activities like data preprocessing, model training, and system integration, Python is used as the main programming language. Deep learning frameworks like TensorFlow and PyTorch enable programmers to create intricate neural networks that can accurately analyze and interpret visual data from security cameras. These frameworks make it easier to apply recurrent neural networks (RNNs) to comprehend temporal patterns suggestive of accidents and convolutional neural networks (CNNs) to extract pertinent features from video feeds. Libraries for real-time image processing, such as OpenCV, are also part of the Python ecosystem and improve the system's ability to quickly identify and address incidents. Making use of deep learning and Python guarantees that accident detection. In practical applications, systems can achieve reliable performance, scalability, and efficiency.

TABLE I. SOFTWARE REQUIREMENTS

The following software specifications must be met in order to construct the application:

Operating system	Windows 7/8/10
Programming Language	Python
IDE	Python, Anaconda Navigator
Dataset	Trained Model

Real-Time Processing: In order to quickly identify incidents, the system must process sensor data and video feeds in real-time. High Accuracy: To reduce false positives and negatives and provide dependable alerts, a high detection accuracy must be attained. Scalability: The capacity to manage fluctuating data loads and traffic patterns in various geographical areas without experiencing a decline in performance of the system. Low

Latency: Quick alarm production and notification times to emergency services and pertinent parties of the system. Reliability: Reducing downtime by ensuring that the system performs consistently under a variety of environmental circumstances and traffic volumes of the system. The connection: Smooth connection with the current infrastructure for public safety, emergency services, and traffic control systems. Data Security and Privacy: Ensuring secure data transmission and storage while adhering to privacy standards in data handling. In order to quickly identify incidents, the system must process sensor data and video feeds in real-time. To reduce false positives and negatives and provide dependable alerts, a high detection accuracy must be attained. The capacity to manage fluctuating data loads and traffic patterns in various geographical areas without experiencing a decline in performance of the system.

## VII. RESULTS

### A. SYSTEM TESTING

One of the most important parts of computer programming triggers is testing and debugging; without functional programming, the system would never generate the intended output. The ideal way to conduct testing is to ask user development to help find all faults and bugs. Tests are conducted using the sample data. The quality of the data used in the testing process is more important than its quantity. The purpose of testing is to make sure the system was functioning correctly and effectively before to directives for live operation.

### B. CODE TESTING

This looks at the program's logic. For instance, tests and verifications were conducted on the logic for updating different sample data and with the sample file and directories.

Testing Specifications: Putting this specification into practice by first determining what the software must accomplish and how it must operate in different scenarios. Every module has test cases covering a range of scenarios and combinations of criteria.

Testing units: We test each module separately and integrate it with the system as a whole during the unit testing process. Verification efforts are concentrated on the smallest software design unit within the module through unit testing. Another name for this is module testing. Each system module is tested independently. Another name for this is module testing. Testing is done right during the programming phase.

Every module is found to function satisfactorily in terms of the expected output from the module during the testing phase. Additionally, there are some validation tests for fields. For instance, a validation check is carried out to see if the user's input varies and the data entered is genuine. Errors in the system are quite straight forward to locate in the testing units of the system.

### C. Route Mapping Testing

Testing an AI-enabled accident detection and alert system on a route map involves several critical steps. First, define test scenarios to assess detection accuracy, alert timeliness, map integration, and false positive/negative rates. Prepare a test environment using simulated and real-world accident data across various route types. During testing, ensure the system accurately detects accidents and measures alert latency, maintaining performance under different load conditions. Verify that accident locations are correctly marked on the map and that notifications are sent promptly through all

intended channels. Collect user feedback to refine notification content and format. Analyze data to identify and resolve issues, continuously improving the system through iterative testing. Tools like simulation software, sensor data integration, mapping APIs, and communication APIs are essential in this process. Testing an AI-enabled accident detection and alert system on a route map involves several critical steps. The ideal way to conduct testing is to ask user development to help find all faults and bugs. Tests are conducted using the sample data. The quality of the data used in the testing process is more important than its quantity of testing an AI-enabled accident detection and alert system on the route map involves several critical steps.

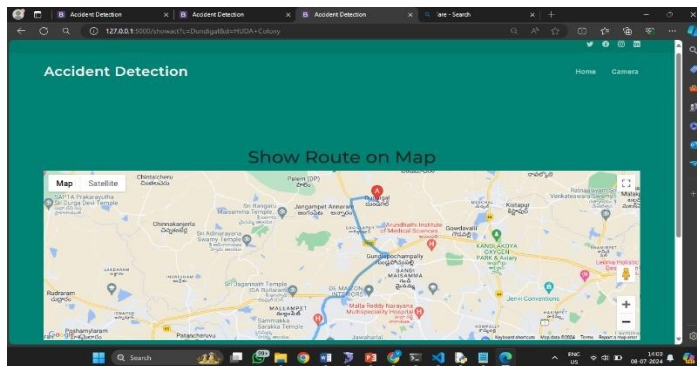


Figure 7.1: Route Map Testing

Figure 7.1, "Route Map Testing," illustrates the route where an accident is detected and an emergency is notified. To verify the dependability and efficacy of an AI-enabled accident detection and alarm system for emergency scenarios on a route map, there are a few crucial actions that must be taken during testing. Establish test scenarios first, emphasizing timely alerts, accurate accident detection, and a reduction in false positives and negatives. Create a test environment that includes real-world and simulated accident data from a variety of routes. Make sure the system reliably identifies incidents and promptly notifies users during testing, even with different load scenarios. Make sure that any incidents are appropriately noted on the route map and that emergency alerts are given out via email, SMS, and app notifications, among other channels. Assemble the user input to enhance the relevancy and clarity of notifications. Examine test data to find any problems, and keep improving the system by testing and adjusting iteratively. Make use of resources including communication services, mapping APIs, simulation software, and sensor integration to enable thorough testing and guarantee the system functions properly in emergency situations. There are various phases involved in testing an AI-enabled accident detection and alert system on a route map to make sure the system correctly identifies incidents and promptly notifies users of them.

#### D. Accident Detection Destination

Creating test scenarios to gauge timely alerts, detection accuracy, and reducing false positives and negatives are all part of the process. Utilizing accident data from real and simulated scenarios on a variety of routes, a test environment is set up. The System Continuous examination of test findings aids in problem identification, system enhancement, and the emergency situation reliability.

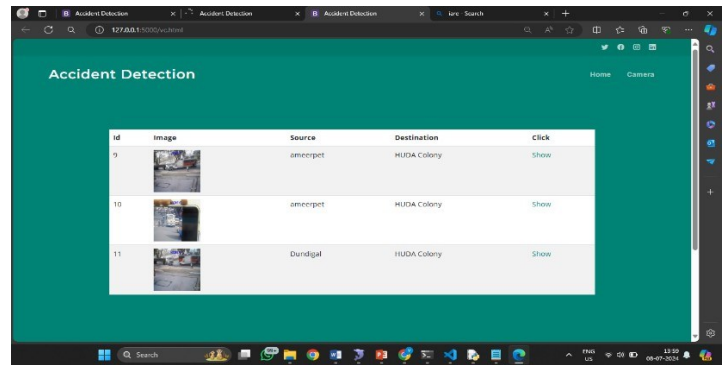


Figure 7.2: Accident Detection Destination

From Accident Detection Destination (Fig. 7.2) Ensuring that an accident detection and alert system reliably detects accidents, issues timely alerts, and efficiently manages updates relevant to the destination are all part of testing the system for emergency scenarios. Creating test scenarios to gauge timely alerts, detection accuracy, and reducing false positives and negatives are all part of the process. Utilizing accident data from real and simulated scenarios on a variety of routes, a test environment is set up. It is critical to confirm throughout testing that incidents are appropriately classified, alerts are sent out on time, and users are appropriately informed of route updates. Furthermore, the system's functionality under various load scenarios is evaluated, and the accuracy and lucidity of user alerts are tested. To improve the system, continuous changes and ongoing data analysis are crucial. Instruments such as Comprehensive testing is made possible by simulation software, sensor integration, mapping APIs, and communication services, which guarantee the system operates dependably in actual emergency scenarios. To verify an AI- powered accident detection and alarm system's efficacy in actual emergency scenarios, a thorough testing process comprising multiple stages must be followed. To evaluate detection accuracy, alert timeliness, and the system's capacity to change the route to the target following an accident, test scenarios are first defined. This include assessing the system's performance in handling route updates, measuring the interval between accident detection and alert transmission, and identifying accidents using sensor data. In order to conduct testing, a variety of routes and accident scenarios are created using both simulated and real-world data. Additionally, it entails monitoring system performance at different loads, improving detecting algorithms to reduce the number of false positives and negatives while making sure that notifications are the much understandable and useful. The testing process

is further assisted by resources like communication services, sensor integration, mapping APIs, and simulation software. The System Continuous examination of test findings aids in problem identification, system enhancement, and the emergency situation reliability.

#### VIII. CONCLUSION AND FUTURE SCOPE

For particularly specific issues, pre-trained neural networks are unable to generate a vector with relevant information. As a result, utilizing instances of the problem that has to be solved, the weights of these models must be adjusted. While the computational cost, processing time, and accuracy in accident detection present better results by not conditioning the selection of frames to a metric, the technique that best represents a temporal segment of a traffic accident does not eliminate any data because the similarity values between the segments of the techniques with frame selection present negligible differences between them. The comprehension of video scenes in artificial vision has advanced significantly. Artificial neural networks are among the most effective methods. The creation of an alert system for accidents system is a notable development in car safety technology. These devices can quickly identify collisions and notify emergency personnel, thereby saving lives and lessening the severity of injuries, by combining sensors, GPS, and communication technology. Prompt accident detection with many sensors, automatic alarm production via GPS and communication modules, improved safety for drivers and passengers, useful data collecting for research, and raised user awareness through real-time notifications are some of the key accomplishments of the present systems. Thanks to technology improvements and a growing focus on road safety, accident detection and alert systems have a bright future ahead of them. Integration with self-driving cars, the use of AI and machine learning to improve accident prediction and prevention, and enhanced 5G communication networks are possible future improvements technology, as well as interaction with the infrastructure of smart cities to enable coordinated emergency responses. Furthermore, enhanced user interfaces could guarantee timely and appropriate driver reactions, global standardization could guarantee system consistency and reliability, and wearable technology could improve safety for pedestrians and bikers. Widespread adoption can be further encouraged by lowering costs through increasing production and technological breakthroughs, partnering with insurance companies to offer incentives for automobiles equipped with cutting-edge technology, and more. Even if the safety of vehicles has been greatly increased by the systems in place now, further study and technology developments have the potential to make roads even safer for everyone.

#### REFERENCES

- [1] Li, M.Z. The Road Traffic Analysis Based on an Urban Traffic Model of the Circular Working Field. *journal "Acta Math. Appl. Sin"*. 2004, 20, 77–84.
- [2] Chu, W.; Wu, C.; Atombi, C.; Zhang, H.; Özkan, T. Traffic Climate, Driver Behaviour, and Accidents Involvement in China. *journal "Accid. Anal. Prev"*. 2019, 122, 119–126.
- [3] Guimarães, A.G.; da Silva, A.R. Impact of Regulations to Control Alcohol Consumption by Drivers: An Assessment of Reduction in Fatal Traffic Accident Numbers in the Federal District, Brazil. *Journal "Accid. Anal. Prev"*. 2019, 127, 110–117.
- [4] Nishitani, Y. Alcohol and Traffic Accidents in Japan. *Journal "IATSS Res"*. 2019, 43, 79–83.
- [5] Mahata, D.; Narzary, P.K.; Govil, D. Spatio-Temporal Analysis of Road Traffic Accidents in Indian Large Cities. *Journal "Clin. Epidemiol. Glob. Health"* 2019, 7, 586–591.
- [6] Sheng, H.; Zhao, H.; Huang, J.; Li, N. A Spatio-Velocity Model Based Semantic Event Detection Algorithm for Traffic Surveillance Video. *Journal "Sci. China Technol. Sci"*. 2010, 53, 120–125.
- [7] Parsa, A.B.; Chauhan, R.S.; Taghipour, H.; Derrible, S.; Mohammadian, A. Applying Deep Learning to Detect Traffic Accidents in Real Time Using Spatiotemporal Sequential Data., *journal "arXiv:1912.06991"*. 2019.
- [8] Joshua, S.C.; Garber, N.J. Estimating Truck Accident Rate and Involvements Using Linear and Poisson Regression Models. *Journal "Transp. Plan. Technol."* 1990, 15, 41–58.
- [9] Arvin, R.; Kamrani, M.; Khattak, A.J. How Instantaneous Driving Behavior Contributes to Crashes at Intersections: Extracting Useful Information from Connected Vehicle Message Data. *Journal "Accid. Anal. Prev"*. 2019, 127, 118–133.