International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Accident Detection and Real Time Alert System using CNN

Aayush Kumar Dewangan[1],Thailendra Yadav[2], Akhil Agrawal[3], Ankur Kumar Rai[4]

Under the guidance of **Tinam Agrawal[5]** (Assistant Professor)

Department of Computer Science and Engineering (Data Science)

Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India

---***---

Abstract - Road accidents remain a leading cause of fatalities worldwide, primarily due to delayed emergency response and lack of real-time monitoring systems. This paper presents a deep learning-based Accident Detection and Alert System utilizing Convolutional Neural Networks (CNNs) to identify accidents from video streams in real time. The proposed system processes video frames captured from cameras and classifies them into accident and non-accident scenarios. Upon detection, an alert mechanism is triggered to notify emergency contacts. The system is implemented using Python with libraries such as TensorFlow, OpenCV, NumPy, and Keras. Experimental results demonstrate that the model achieves high accuracy in detecting accident scenarios, making it a reliable and cost-effective solution for enhancing road safety.

Key Words: Accident Detection, Convolutional Neural Network, Deep Learning, Computer Vision, Real-Time Monitoring, Alert System

1.INTRODUCTION

Accidents caused by road traffic are considered a serious problem today because such accidents often result in casualties, deaths, and damage to property. Delays in the detection of accidents and sending notifications about these accidents complicate situations and may lead to tragic consequences. Although there are many cameras in public places, most of them operate under human control and cannot react promptly and correctly during an emergency. Therefore, the development of an automated accident detection system is required to detect the problem and inform the authorities immediately after an incident.

Thanks to recent developments in machine learning algorithms, researchers created computer vision systems that enable analyzing visual information and detecting abnormal activity in it. The Convolutional Neural Networks (CNN) is especially useful for image and video classification **since it** automatically learns important features in frames. The CNN is implemented in the proposed model to detect accident cases in traffic and separate accidents from other regular traffic cases. When an accident is detected, a notification is generated and sent by the system.

This research paper aims to improve road safety by developing an automated accident detection system and making it cheaper and faster to use than manual methods of solving the problem. The model suggested by the authors is based on modern advances in computer vision and deep learning technologies and can detect accidents in real time.

2. Literature Review

There exist a variety of papers discussing the application of deep learning and computer vision to automate road accident detection processes. In particular, the paper "Road Accident Detection Using Deep Learning" from IJNRD utilizes binary image classification and achieves 77.27% accuracy rate in detecting accidents. Additionally, the researchers note that improvements could be attained in future by training the model with a larger set of images and adjusting the interface to make the software usable. Thus, this paper shows that the technology is capable of detecting road accidents provided that enough training data is available.

In their paper "Computer Vision Based Accident Detection and Alert System" from IJRTI, the authors discuss the creation of the system, which includes video or image pre-processing, accident detection using CNN algorithm and alerting function. According to the authors, one of the key benefits of developing such a system is the ability to reduce emergency response time by detecting accidents and sending notifications based on CCTV footage. Thus, this paper may be relevant in the context of traffic monitoring because it involves scene analysis and automated alerting.

One of the important research papers in question, "Real Time Accident Detection System using CNN" by IRJIET, focuses specifically on developing the system, which detects accidents in real time using CNNs. The researchers argue that CNN algorithms are capable of effectively recognizing accident incidents and thus should be applied in practical accident detection systems because of the unique capability to distinguish between accident scenes.

Finally, the paper "Accident Detection System on Roads using Convolutional Neural Network" from IJARSCT utilizes

CNNs to detect and classify different types of road accidents. Namely, CNN algorithms were able to recognize the accident incident and classify the type of the accident, including whether there was a rear-end collision or a side collision. This means that CNNs are capable not only of detecting road accidents but also analyzing their specifics in detail.

In summary, several research papers have shown that CNNs are quite efficient tools for developing accident detection systems for roads. However, many of them lack certain practical aspects because of such factors as the small amount of training data, changing environmental conditions, etc. Thus, this paper aims at developing a practical accident detection and alerting system with future applicability potential to the automotive industry.

3.Proposed Methodology

3.1 System Overview

The Accident Detection and Alert System is meant to detect road accidents from video data and produce alerts automatically. It consists of a series of procedures including video data acquisition, extraction of frames from the videos, image preprocessing, feature extraction, and classification before generating alerts. This tool seeks to minimize the time interval between when the accident occurs and when the response team starts its actions by using computer vision and deep learning methods.

The input data can be collected from surveillance cameras, dash cams, and other sources. Video footage is divided into frames that will be used by the model individually. A Convolutional Neural Network is used to recognize the differences between images of accidents and those without incidents. Afterward, the model generates alerts based on the output predictions.

3.2 Data Collection and Dataset Preparation

The success of the proposed system relies on the nature and variation in the choice of data used in training and testing the system. In this case, a dataset of images and/or video frames with accident and non-accidents scenarios will be obtained from relevant sources. Every image is categorized into either of the two categories for ease of analysis. It is desirable that the dataset should have a good balance in order to avoid favoritism of one category over another.

Once the dataset is obtained, it is split into training and testing sets. While the former set is used to develop knowledge of visual accidents, the latter set is used for testing purposes.

3.3 Video Frame Extraction

Since the CNN model is designed to work with images, the video will be converted to a sequence of frames. Sampling is done at equal time intervals such that the scenes of the video are captured in form of frames. This process helps in analyzing the different scenes independently so as to detect anomalies that suggest an accident.

The extraction of frames is an important step, since it allows the conversion of a continuous video stream to discrete inputs. Moreover, it allows real-time processing of the scenes while still maintaining efficiency in computation. In implementation, the frame extraction rate can be modified depending on hardware capabilities.

3.4 Image Preprocessing

Prior to passing the frame to the CNN model, preprocessing is performed on each of them in order to make them more uniform and reduce any noise. The frames are rescaled into the same size in order to keep the consistency of the input sizes that are necessary when training and making inferences with the use of neural networks. In addition to this, pixels within the images are normalized to some standard scale, usually the range of [0, 1]. Other forms of preprocessing may be included to improve image quality in case they are necessary. All these steps help the model concentrate on the important characteristics of the scene.

3.5 CNN-Based Feature Extraction

The preprocessed frames are fed into the Convolutional Neural Network (CNN) for feature extraction. It is worth mentioning that this neural network model can be used for this purpose due to its ability to automatically extract hierarchical features from images. In lower layers, the convolutional process will help in extracting simple features such as edges, curves, and textures; higher layers will **extract** complicated features related to accidents. On the other hand, pooling helps in reducing the number of dimensions while retaining informative features and thus minimizing computational complexity without losing vital visual information.

3.6 Classification

The classification phase decides which of the two classes, namely accident or no accident, the input image belongs to. The fully connected layer processes the features and produces the final output. The output layer utilizes the softmax activation function and provides the probabilities of each class. The class with higher probability becomes the final output. In real-time applications, the decision can either be taken based on one image alone or on multiple images together to increase accuracy, thereby reducing false detection caused by any momentary visual distractions.

3.7 Alert Generation

If there is enough certainty about the occurrence of an accident, the alert generator module is triggered automatically. An alert is sent out to pre-specified recipients through communication methods like emails or instant messaging platforms. Information such as the time at which the accident was detected could be included in the alert along with other relevant details.

The primary purpose of this module is to make sure that assistance arrives as quickly as possible in case of accidents. Automating the process of sending alerts makes the system less dependent on human actions and increases the chances of getting prompt help. Therefore, this system would be ideal for use in monitoring roadways and other smart safety solutions.

3.8 Tools and Technologies Used

This system is designed mainly using the Python language. For the design and training of the CNN model, TensorFlow and Keras frameworks are used. OpenCV is used for capturing video and extracting frames from the video. The library NumPy is used for numerical calculations, whereas Pandas is used for handling and organizing datasets. Visualizing the training process, including accuracy and loss values, is done using the Matplotlib framework. Overall, these technologies provide a convenient and versatile platform for building computer vision-based deep learning applications.

3.9 Workflow Summary

The steps involved in the proposed methodological approach can be described as follows: firstly, the video data is obtained through the camera; secondly, the video is segmented into frames; thirdly, the frames are preprocessed to change their size and normalize; fourthly, the CNN learns useful features from the frames; fifthly, the classification of the scene as an accident or not is carried out; sixthly, if an accident has occurred, the alerting function kicks in.

Thus, such a methodological approach offers a comprehensive solution to the issue of detecting accidents on the roads.

4. System Architecture

In order to detect accidents in real-time scenarios, we propose building an accident detection and alert system that involves capturing video footage, identifying accident cases using a convolutional neural network (CNN), and raising alerts if any accident is detected. In terms of system architecture, the model is constructed on a modular basis and consists of five stages: video input, frame extraction, CNN detection, decision-making, and alert creation. Each step performs a unique role within the process flow.

4.1 Input Layer

In the first stage, videos are obtained from the camera source, which may be in the form of a CCTV camera, dashcam, or even any live surveillance system. This video is then streamed to the processing layer. Therefore, this is where all video data will be fed to the model.

Fig 1. Dashcam Input

4.2 Frame Processing Unit

Since deep learning algorithms are built around images rather than videos, this input layer has to be converted into images. The video will be broken down into images and processed. Such tasks will include resizing, normalizing, and removing noise from the images.

4.3 CNN Detection Module

The frames after being processed will be fed into a CNN detection model that learns the significant visual features of the traffic videos. This will include edges, shapes, vehicle shapes, and accident-like features. Afterward, the model will classify each frame into an accident or no accident. This represents the primary component of our system.

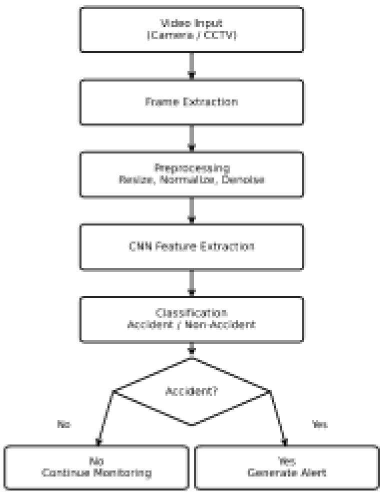

Fig 2. CNN Model Flowchart

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

4.4 Decision Unit

In the next stage, the decision-making process takes place where predictions from the CNN will be analyzed and determined whether the predicted accident is true or not. For increased accuracy, multiple frames may be used before deciding on the output of an accident case.

4.5 Alert Generation Module

The alert generation module will be activated when the prediction from the previous step indicates the presence of an accident. In such cases, a notification will be sent to relevant parties in the form of an email or message. Information such as the time and date of detection can also be added to such notifications.

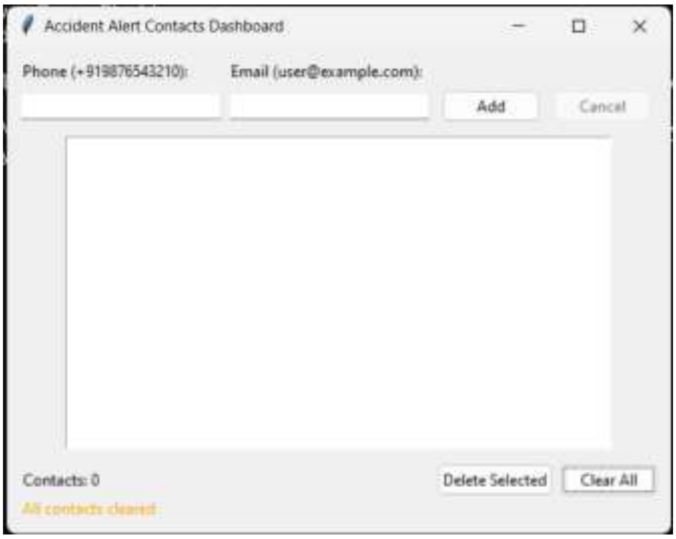

Fig 3. Contact dashboard

4.6 System Flow

As described above, the steps in the proposed accident detection and alerting system include capturing a video, breaking down frames, pre-processing the frames, detecting accidents using CNNs, making decisions about the predictions, and generating alerts whenever there is an accident.

5. Implementation

5.1 Development Environment

The suggested system for accident detection and alerts is developed using Python, an efficient tool for developing computer vision models. In the current project, the TensorFlow and Keras libraries are used for the development of the machine learning model and its training. The OpenCV library is needed for video processing while NumPy, Pandas, and Matplotlib are applied to prepare the dataset and visualize the training results.

5.2 Dataset Preparation

An appropriate dataset is prepared with the use of a video capturing application. The gathered information is divided into training and testing sets for supervised learning purposes. In particular, images are labeled with a class label that corresponds to either an accident or non-accident event. Ideally, the dataset should be balanced to avoid possible class imbalance.

5.3 Video Processing

In the current process, the system receives video data from the user, including the footage obtained using a live camera feed, CCTV camera streams, dashboard cameras, and recorded videos. The video stream is separated into frames. Each frame represents an image and is sent to the preprocessing step for further analysis.

5.4 Preprocessing

Before feeding into the CNN model for classification, each frame is preprocessed to ensure optimal results. Specifically, the following actions are taken. Firstly, the frame size is fixed by resizing the image to a predetermined value. Further, pixel values are normalized. Additionally, any noise removal or enhancement algorithms may be applied.

5.5 CNN Model Implementation

CNN is implemented to classify traffic accidents as such. CNN includes several convolutional layers for feature extraction, followed by a number of pooling layers. Fully connected layers are used for classification. The learned convolutional layers are able to detect key patterns such as edges and textures. The output layer outputs a prediction of whether an input image belongs to one of the two classes.

5.6 Training Process

During the training procedure, the convolutional neural network learns the relationships between features of accident/non-accident images through adjusting parameters inside itself. Iterations over a set number of epochs are performed until a model capable of correctly classifying the input data is obtained. Monitoring changes in accuracy and loss is performed to evaluate the quality of training; furthermore, validation data could be used to detect any signs of overfitting.

5.7 Accident Detection and Alert Generation

When the model has been trained to satisfactory standards, it can be put into use for detecting accidents. After the image is analyzed, its probability of being associated with an accident event is computed; in case this probability reaches or exceeds a predetermined threshold, the system sends out an alert notification. Notifications can be sent either by email or via instant messaging apps to the appropriate parties.

Fig 4. Alert Generation

5.8 Result Visualization

The results obtained from the CNN classifier are presented graphically by visualizing metrics such as accuracy and loss; they provide valuable insights regarding the current state of training and indicate whether improvements are possible. In addition to the aforementioned metrics, the output consists of real-time classification results and alerts whenever accidents are detected.

5.9 Conclusion

As was demonstrated, the described accident-detection framework combines elements of video preprocessing and analysis using deep learning techniques; notifications about the detected incidents are generated automatically in the form of emails or messages. Overall, the approach is rather promising in terms of its applicability to real-world scenarios.

6. Results and Discussion

6.1 Experimental Results

In order to test the performance of the accident detection algorithm, the experiment used a dataset containing images depicting both accident and non-accident scenarios. The results of the CNN training process indicated that the accuracy of the proposed model is approximately 98.2% on the training data and approximately 87.8% on the validation set. It shows that the algorithm has learned the patterns of accident detection successfully.

According to the obtained results, one can state that the proposed algorithm is capable of distinguishing between accident scenes and normal traffic scenes with great accuracy. Although the validation accuracy score is significantly lower than the training one, it still indicates satisfactory performance. As can be seen from the discrepancy between the two accuracy scores, there could be some patterns specific to the dataset learned by the model.

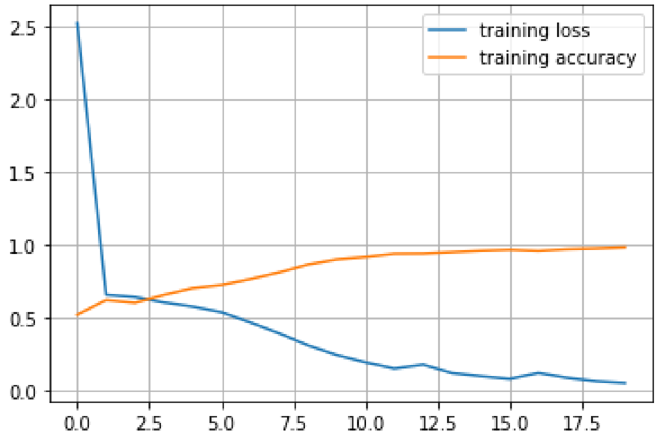

Fig 5. Training loss & accuracy

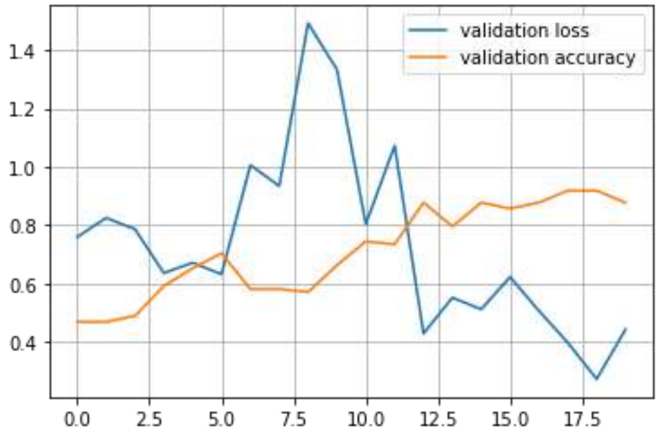

Fig 6. Validation loss & accuracy

6.2 Discussion

The results obtained in the experiment prove the suitability of the chosen approach for accident detection tasks. The CNN is able to learn salient features in video frames such as abrupt deformation of objects, unusual vehicle position, and patterns caused by collisions. Therefore, it can distinguish between accident and non-accident images rather accurately.

The difference between training and validation accuracy scores proves the fact of overfitting; however, it is quite usual for deep neural networks. To avoid such a problem, it is possible to provide more training data, apply data augmentation techniques, use better regularization, etc. Regardless of this drawback, the current algorithm can perform sufficiently well to detect accidents in real time.

Another advantage of the suggested method is the ability to generate alerts. Once the accident is detected, the alert can be generated instantly, allowing quick reactions from traffic service officers. Thus, this solution appears superior to traditional methods requiring manual monitoring.

6.3 Interpretation of Results

The trained network shows a high accuracy level of 98.2% and indicates the proper learning of the features associated with the accidents' data set. An accuracy of 87.8% for the validation process confirms a successful performance on unseen data, but there are possibilities of improvement. Regarding a practical implementation of this model in an actual safety application, the obtained performance is quite encouraging since it can provide immediate warning in real-time situations.

Overall, the results obtained during this study show that the designed CNN-based system can be considered a successful accident detection tool. In addition to accurate classification ability, the automatic generation of alerts makes the system useful for road safety applications.

6.4 Conclusion

The experiment carried out for evaluating the accident detection system shows the success in obtaining a highly accurate model. This system is able to detect accident scenes in real time based on video feed and automatically send an alert to the appropriate authorities.

7. Conclusion and Future Scope

7.1 Conclusion

The current study presents a CNN-based system for accident detection and alert generation, aimed at identifying road accidents from a video feed and sending out automatic alerts. The presented framework provides a unified solution that comprises video capturing, frame preprocessing, deep learning-based classification, and generating alerts. Based on the conducted experiments, the training accuracy is close to 98.2%, while the validation accuracy is around 87.8%. This suggests that the proposed framework can distinguish between accidents and regular traffic conditions and provide accurate results in most cases. It becomes evident that there is considerable potential in employing machine learning in road safety applications.

In particular, the presented method reduces the need for constant human monitoring of surveillance video feeds and enables more efficient identification of accidents and generation of alerts, improving the chances for timely assistance to victims. All in all, the designed framework provides an efficient, cost-effective, and intelligent solution that can be utilized in existing traffic monitoring frameworks.

7.2 Future Scope

Despite satisfactory results achieved in the current study, there are various improvements that can be done to enhance the system. One of such improvements is related to the development of a mobile application, which should be capable of performing the tasks of accident detection using CNN on Android platforms. In particular, such an application can perform accident detection based on the footage received from the camera located inside the car's dashboard and send alerts in case an accident happens.

Moreover, future studies may involve increasing the dataset size to accommodate additional accident situations, weather and lighting conditions, and camera angles. The exploration of alternative deep learning architectures, optimization methods, and implementation of edge computing technologies may improve the accuracy of classification and speed up the process significantly. Moreover, integration of additional features like GPS, cloud computing, and connectivity with smart city infrastructure may allow the application of accident information to nearby hospitals and traffic authorities automatically.

Reference

1.Road Accident Detection Using Deep Learning, International Journal of Novel Research and Development (IJNRD), 2024.

2.Computer Vision Based Accident Detection and Alert System, International Journal for Research Trends and Innovation (IJRTI), 2023.

3.Real Time Accident Detection System using CNN, International Research Journal of Innovations in Engineering and Technology (IRJIET), 2025.

4.Accident Detection System on Roads using Convolutional Neural Network, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)