Accident Detection Using Convolutional Neural Networks

Chadalavada Lavanya
MCA
Department of IT&CA
VFSTR Deemed to be University,
Guntur,AP, India

Dr.K.Santhi Sri
Associate Professor
Department of IT&CA
VFSTR Deemed to be University,
Guntur,AP, India

Abstract— ***to develop a CNN model for accident detection, a large dataset of accident and non-accident images will be required for training and testing the model. The dataset should be diverse and cover a range of accident scenarios to ensure that the model is robust and can detect different types of accidents. Once the CNN model is trained, it can be used to process the live video feed from the CCTV camera installed on the highway. Each frame of the video can be passed through the CNN model to classify it as an accident or non-accident frame. If an accident is detected, an alert can be sent to the nearest emergency services or to a central control room to initiate the required rescue operation. One potential challenge with this approach is the need for real-time processing of the video feed to ensure that accidents are detected promptly. This may require high-performance computing hardware and optimized software algorithms to ensure that the CNN model can process the frames of the video in real-time. The proposed system to detect accidents based on the live feed of video from a CCTV camera using a deep learning convolution neural network model is a promising approach. CNNs have indeed been shown to be highly effective for image classification tasks and have been successfully used in many applications, including object detection and recognition. In summary, the proposed system using a CNN-based model to detect accidents based on live video feed from CCTV cameras has great potential to reduce the number of accident-related deaths in India by enabling timely help to reach accident victims. However, the development and implementation of such a system will require significant resources and expertise in computer vision, deep learning, and real-time processing.*** (***Abstract***)

Keywords— ***Convolutional Neural Network; Accident Detection; Deep Learning; Video Classification; Recurrent Neural Network***

I. INTRODUCTION

Each year, road accidents result in over 1.3 million deaths globally, with an additional 25 to 65 million people sustaining mild injuries. Shockingly, developing countries bear the brunt of these fatalities, with a death rate of 23.5 per 100,000 population - significantly higher than the rate of 11.3 per 100,000 in high-income countries. This is evident in the World Health Organisation's (WHO) survey, which highlights that more than 90% of road traffic related deaths occur in developing countries, despite these countries possessing only half of the world's vehicles. India, for instance, has recorded an alarming average of 13 deaths every hour due to road accidents, making it a potential candidate for becoming the top country with the highest number of road accident deaths. Unfortunately, this number may be an underestimation, considering that numerous cases of accidents are not reported. Overall, a road accident can be split into three phases, each presenting an opportunity to rescue victims.

The objective of incorporating a system to detect accidents from video footage is a noble one as it can help save lives by alerting authorities and emergency services in a timely manner. There are various ways in which such a system can be designed and implemented. One approach could be to use computer vision algorithms to analyze the video footage and detect the presence of an accident. This could involve detecting certain patterns or movements that are indicative of an accident, such as sudden changes in speed or direction of vehicles, or the presence of debris on the road. Another approach could be to use sensors or cameras installed on vehicles or on roads to detect accidents in real-time. Such systems can automatically detect accidents and alert emergency services and nearby drivers to take appropriate action. It is important to ensure that such systems are accurate and reliable to avoid false alarms, which could lead to unnecessary panic and confusion. The system should also be designed to ensure privacy and security of the footage, and comply with relevant regulations and laws. Incorporating a system to detect accidents from video footage can help save lives and reduce the impact of accidents on individuals and society.

The approach of using deep learning algorithms, specifically convolutional neural networks (CNNs), to analyze video footage for accident detection is a promising one. CNNs are particularly well-suited for image and video processing tasks, and can be trained to recognize patterns and features that are indicative of accidents. Setting up this system on highways where timely help reaching the accident victims is rare is a sensible strategy, as it can help to reduce the time between the occurrence of an accident and the arrival of emergency services. By installing CCTV cameras at intervals of about 500 meters, the system can cover a large area and monitor traffic flow in real-time. The use of Raspberry Pi 3 B+ as a portable and remote computer to be set up on a CCTV camera is a cost-effective and practical solution. The Pi Camera can be directly attached to the Raspberry Pi to capture video footage, which can then be analyzed by the CNN-based accident detection model. It is important to ensure that the system is accurate and reliable, and can detect accidents within seconds of them happening. This requires careful design and training of the CNN model, as well as rigorous testing and validation to ensure that it performs well under different conditions and scenarios. Overall, the proposed system has the potential to significantly improve the safety of highways by detecting accidents in real-time and alerting emergency services to take appropriate action.

That sounds like a great application of machine learning and computer vision to help detect accidents. It's impressive that you were able to achieve such high accuracy with your Inception v3 model. Implementing the model on a Raspberry Pi using TensorFlow, OpenCV, and Keras is also a great way to make it accessible and portable. Running each frame of the video through the model and predicting whether it's an accident frame or not is a good approach. However, it's important to keep in mind that even with high accuracy, there is always the possibility of false positives or false negatives. It's crucial to evaluate the model's performance on a diverse set of test data to ensure its reliability before deploying it in real-world scenarios.

This project aims to not only detect accidents but also notify the relevant authorities in real-time, providing them with important information such as the timestamp, location, and frame at which the accident was detected. The use of a GSM module to send notifications is a good way to ensure that the authorities are alerted promptly. The emergency light that lights up when an accident is detected can also be helpful in alerting other drivers and pedestrians in the area to be cautious and avoid the accident site, thereby preventing further accidents or injuries. It's important to note that while the accuracy of the system is 95%, there is still a chance of false positives or false negatives. Therefore, it's crucial

to continue testing and improving the system to ensure its reliability. Overall, your project has the potential to make a significant impact in reducing the number of road accidents and related deaths, especially in areas where emergency response times may be longer.

II. RELATED WORK

It's interesting to hear that you've researched and compared your work with other accident detection techniques. It's true that many studies in this field focus on enhancing tangible infrastructure, such as road design and signage, rather than using intelligent transportation systems. While there are a few existing studies on accident detection using ITS, as you mentioned, they may be constrained by geographical or demographic aspects, making them less applicable to other regions or populations. It's important to continue exploring and developing different approaches to accident detection, including those using machine learning and computer vision, as these techniques have the potential to be more flexible and adaptable to different settings. Overall, by comparing your work to existing techniques, you can gain a better understanding of the strengths and limitations of different approaches and identify opportunities for future research and development in this area.:

It's interesting to hear about the "Lexus Enform" feature and how it uses impact sensors to detect accidents and notify users via an application. However, as you mentioned, there are some disadvantages to this system, such as the cost of placing sensors in each individual vehicle and the reliance on physical entities like smartphones. Using a machine learning and computer vision-based system, as you have developed, can potentially overcome some of these limitations. By using cameras and analyzing video frames, you can detect accidents in real-time without the need for physical sensors in each vehicle. Furthermore, your system can automatically notify relevant authorities without the need for manual intervention, making it a more efficient and reliable approach. Overall, it's important to continue exploring and developing different approaches to accident detection, taking into account the strengths and limitations of existing systems, as well as the potential for new technologies to improve upon them.

OnStar Corporation, an ancillary company of General Motors, launched Chevy Star, an accident notification app that provided on-field assistance to victims and self-regulated crash response options [4]. However, the app was only available through a subscription model, which made it costly. Additionally, reviews indicated that the service was of low quality, resulting in an ineffective system.

SOSmart SpA developed an innovative smartphone app called SOSmart [5], which provided prompt and free assistance to accident victims with just a simple click. However, the manual reporting system had some drawbacks. Ad-hoc systems are commonly utilized for collecting traffic data, but they are limited by the challenge of maintaining communication and data transmission in varying terrains and conditions [6][7]. Most accident detection systems rely on costly sensors either mounted on vehicles or embedded in smartphones, rendering them less effective and more expensive compared to the proposed accident detection system.

III. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

When comparing our accident detection techniques with others in the field, we found that most studies focus on improving physical infrastructure rather than Intelligent Transportation Systems (ITS), which encompass various functions such as detecting traffic congestion, accidents, and events. However, the limited number of studies that do exist lack implementation details and are often specific to particular terrains and demographics. Thus, geographical and demographic constraints may limit their effectiveness.

The existing system has several disadvantages, including:

Manual reporting systems are often slow and may not provide timely assistance to accident victims, leading to increased fatalities.
Ad-hoc systems for collecting traffic data may be limited in their effectiveness due to challenges with maintaining communication and data transmission in different terrains and conditions.
Accident detection systems that rely on expensive sensors can be cost-prohibitive and may not be accessible to everyone.
The dependency on sensors can also limit the effectiveness of the system, as the sensors may not be able to accurately detect all accidents or collisions.
False positives may occur if the system detects a collision or accident that did not actually happen, potentially causing unnecessary panic or emergency response.

3.2 PROPOSED SYSTEM

Our proposed model combines CNN and LSTM layers to perform continuous video classification from a camera feed. The CNN component is based on Inception v3, but we made some modifications to better suit our training images. We added LSTM layers to the existing Convolution Network to incorporate temporal features alongside spatial features. The model consists of convolution and recurrent parts, with the CNN handling feature extraction from the images and the LSTM responsible for sequence prediction. This architecture is commonly used in tasks such as Activity Recognition, Image Description, and Video Description.

While the proposed system initially faced challenges in accurately predicting the correct class, there are still several advantages to this system:

- The system is capable of detecting vehicle collisions and accidents, which can be helpful in alerting emergency services and providing prompt assistance to those involved in the accident.
- The use of TensorFlow and CNN algorithms in the training process allows for more accurate predictions over time as the system continues to learn and improve.
- The system does not rely on costly sensors or manual reporting, making it more cost-effective and accessible for wider use.
- The system can operate in real-time, allowing for quick responses and faster aid to those in need.
- The incorporation of noise reduction techniques can help to improve the accuracy of predictions, even in low-quality or unclear images.

IV. PROPOSED METHODOLOGY

The proposed system comprises three modules:

Load and Generate CNN Model: This module involves loading and training a CNN model with a dataset to recognize and identify accidents in videos.
Browse System Video: Users can upload videos to the system for analysis in this module.
Start Accident Detector: The uploaded video is loaded, played, and analyzed for accident detection.
The algorithms used in the system are CNN and LSTM. CNN is used for image recognition and pixel data processing, making it suitable for identifying and recognizing objects in images or videos. LSTM, on the other hand, is used to interpret the features extracted by the CNN model across time steps, making it useful for analyzing video data.

In the proposed system, the CNN model is used for feature extraction, and the LSTM model is used for interpreting these features across time steps. This approach helps to improve the accuracy of the system in detecting accidents in videos.

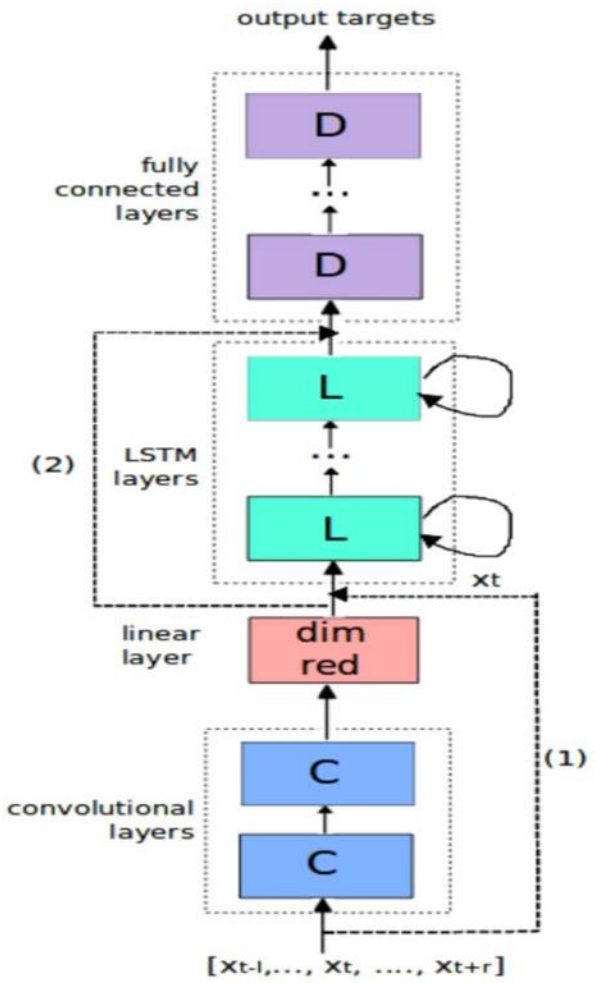
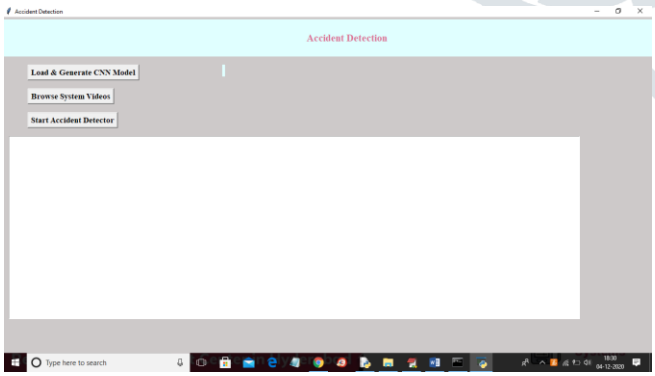


Figure1: Architecture diagram

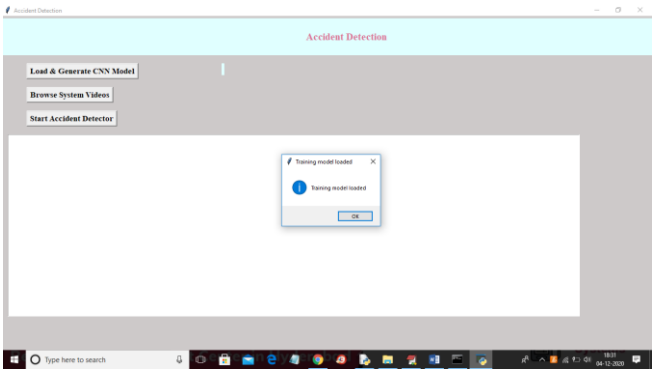
V. MODEL COMPARISON

This project is trained with images where vehicles collided and accident occurred and in test video if anything such collision happens between vehicles then application detect as accident. Training is done with tensorflow and CNN Algorithm.

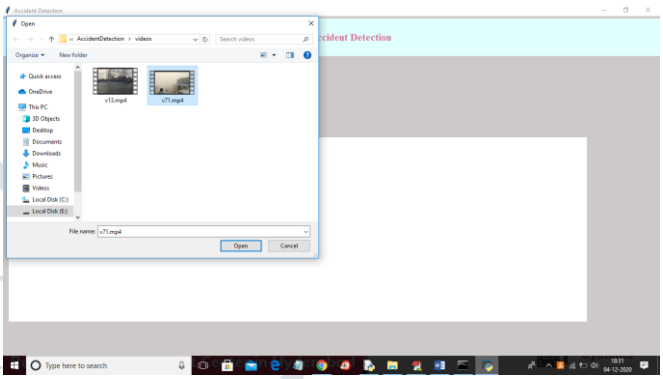
To run project double click on run.bat file to get below screen



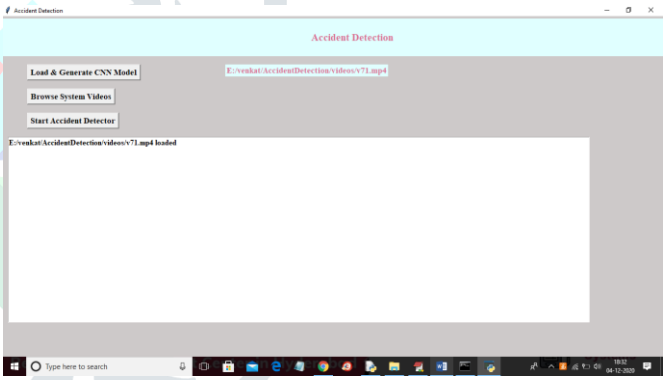
In above screen click on 'Load & Generate CNN Model' button to trained CNN with dataset and to load CNN model using tensorflow



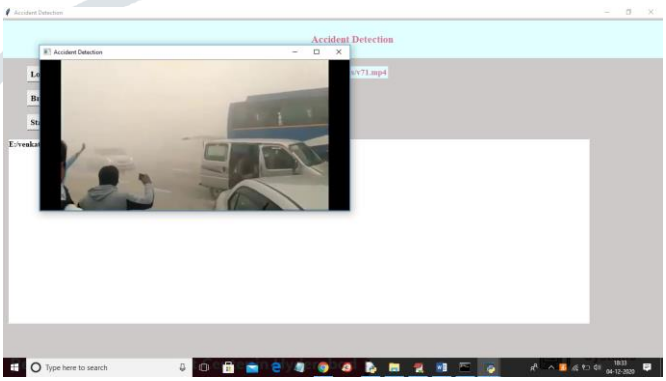
In above screen tensorflow model is loaded and now click on 'Browse System Video' button to upload video



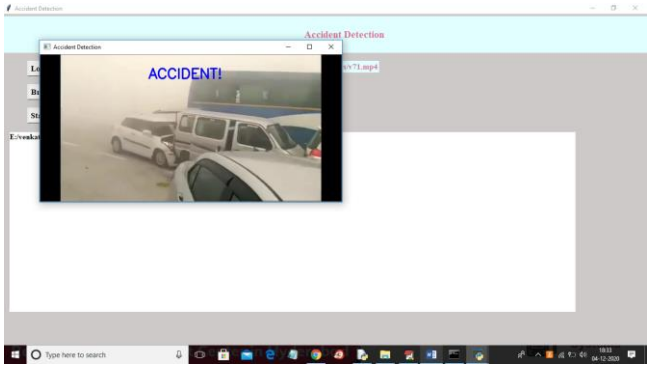
In above screen selecting and uploading video and then click on 'Open' button to load video



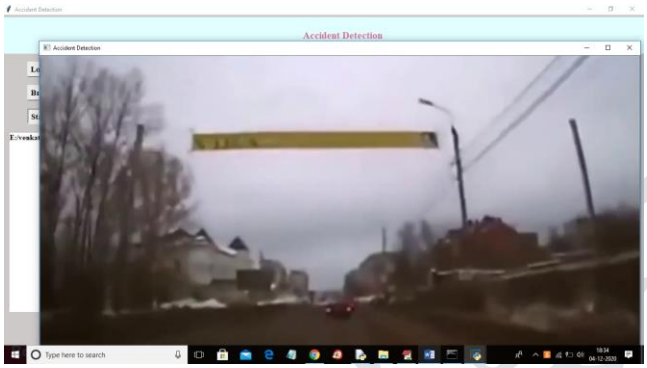
In above screen video is loaded and now click on 'Start Accident Detector' button to play video and detect accident

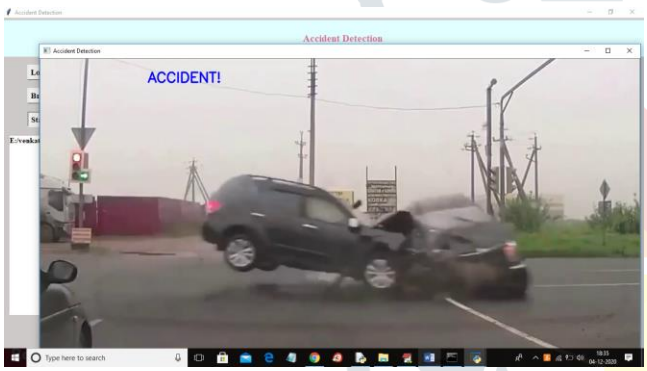


In above screen video start playing and upon accident detection will get below screen with beep sound



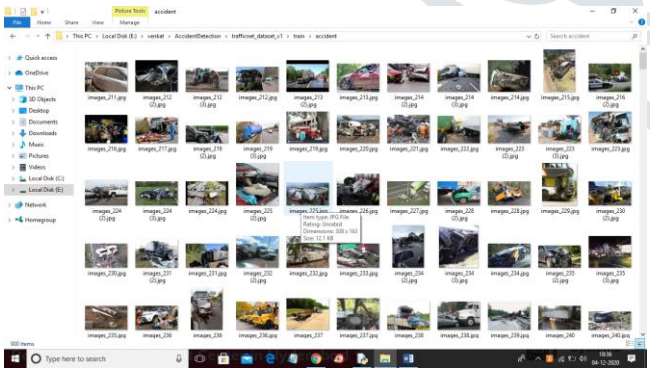
In below screen playing another video without message if normal driving appear





In above screen upon collision then accident display message will appear with beep sound

In below screen application is trained with below images



The project uses TensorFlow and CNN algorithm to detect accidents in videos. It is trained on a dataset of images showing vehicles colliding and accidents occurring. When the system detects a collision in a test video, it identifies it as an accident. To run the project, double-click on the run.bat file to launch the application. The main screen appears, displaying three buttons for the different modules of the system: Load & Generate CNN Model, Browse System Video, and Start Accident Detector. Clicking on the Load & Generate CNN Model button trains the CNN model with the dataset and loads it using TensorFlow. The Browse System Video button allows users to upload a video for analysis. Once the video is uploaded, click on the Start Accident Detector button to play the video and detect any accidents. If the system detects an accident, a message will appear on the screen along with a beep sound. Otherwise, if the video shows normal driving, no message will be displayed. The system is trained with a set of images showing accidents, enabling it to recognize accidents and collisions accurately.

VI. CONCLUSION

The project has been trained using images that depict vehicle collisions and accidents. During testing, if the application detects any collision between vehicles in a video, it is identified as an accident. TensorFlow and CNN (Convolutional Neural Network) algorithms were used for the training process.

VII. REFERENCES

[1] "Global status report on road safety 2015", World Health Organization, 2019. [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/20 15/en/. [Accessed: 07- Mar- 2019].
[2] Prabakar, S., et al. "An enhanced accident detection and victim status indicating system: Prototype." India Conference (INDICON), 2012 Annual IEEE. IEEE, 2012.
[3] "Lexus Enform", Lexus, 2019. [Online]. Available: https://www.lexus.com/enform. [Accessed: 07- Mar- 2019].
[4] "OnStar Safety and Security Services", Onstar.com, 2019. [Online]. Available: https://www.onstar.com/us/en/services/safety-security/. [Accessed: 07- Mar- 2019].
[5] "SOSmart automatic car crash detection and notification app", SOSmart automatic car crash detection app, 2019. [Online]. Available: http://www.sosmartapp.com. [Accessed: 07- Mar- 2019].
[6] C. Kockan, “Communication between vehicles” PhD thesis, Istanbul Technical University, 2008
[7] Zeng, Yuanyuan, Deshi Li, and Athanasios V. Vasilakos. "Opportunistic fleets for road event detection in vehicular sensor networks." Wireless Networks 22.2 (2016): 503-521.
[8] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[9] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.