IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 5, May 2024

A CNN-Based Framework for Video Analysis and Accident Detection

Arnav Tatewar[1], Sakshi Kothurkar[2], Shreyas Jadhav[3], Venktesh Mahajan[4], Mr. Digambar Jadhav[5], Mrs. Savita Jadhav[6]

Department of Computer Science[1,2,3,4,5]

Department of E&TC Engineering[6]

Dr. D.Y.Patil Institute of Technology, Pimpri, Pune, India

arnav.tatewar49@gmail.com[1], sakshikothurkar19@gmail.com[2], shreyasj.jadhav@gmail.com[3]
mahajanvenkatesh6573@gmail.com[4], digambar.jadhav54@gmail.com[5], savita.jadhav@dypvp.edu.in[6]

Abstract: *This research investigates the development and deployment of a Convolutional Neural Network (CNN) model for automatic accident detection in CCTV footage. The ever-increasing reliance on video surveillance necessitates efficient and accurate methods for accident identification. CNNs, with their inherent ability to learn complex spatial relationships within images, are particularly well-suited for this task. This study proposes a CNN architecture that utilizes a pre-trained MobileNetV2 base for feature extraction, followed by a custom classification head tailored to the specific task of accident vs. no accident classification. The model is trained on a dataset of grayscale video frames, achieving an impressive accuracy of 92% on the testing set. This high level of accuracy suggests that CNNs hold significant promise for real-world accident detection applications. Furthermore, to bridge the gap between research and practical implementation, the model is converted to a TensorFlow Lite (TFLite) format for deployment on resource-constrained devices. Additionally, a user-friendly frontend application is developed, empowering users to interact with the model and analyze both images and videos. This user-centric approach broadens the model's accessibility and paves the way for potential improvements in road safety through real-time accident detection.*

Keywords: Convolutional Neural Network (CNN), Accident Detection, CCTV Footage, MobileNetV2, Computer Vision

I. INTRODUCTION

Road accidents remain a significant global concern, with devastating consequences for individuals, communities, and economies. The growing number of vehicles on the road, coupled with factors like distracted driving and non-compliance with traffic regulations, contribute to a heightened risk of accidents. This is particularly true in densely populated urban areas with congested roadways, where accidents occur with alarming frequency. The impact of these accidents is far-reaching, encompassing fatalities, injuries, and substantial property damage.

Therefore, addressing this complex challenge requires a multi-pronged approach that emphasizes both preventative measures and efficient response systems. While public awareness campaigns and stricter enforcement of traffic laws play a crucial role in reducing accidents, technological advancements offer additional tools to enhance safety.

This research paper explores the potential of Convolutional Neural Networks (CNNs) in accident detection within CCTV footage. CNNs are a powerful type of machine learning algorithm renowned for their remarkable capabilities in image recognition and computer vision tasks. Their ability to learn and extract intricate patterns and features from images makes them ideal for accident detection, even under challenging conditions like low-light settings, adverse weather, or cluttered scenes.

Compared to traditional methods, CNN-based accident detection systems offer several advantages. They can simultaneously identify multiple accidents within a single image, allowing for a comprehensive response. Additionally, they can classify specific accident types, such as vehicle collisions, pedestrian incidents, or road hazards, which facilitates the deployment of targeted emergency response measures. Furthermore, by analyzing object movement within a scene, CNNs can potentially predict accident scenarios, paving the way for preventive interventions.

This research delves into the development of a real-time accident detection model built upon CNN architecture. The model is trained on a comprehensive dataset of real-world accident video footage and rigorously evaluated on a separate testing set. The findings demonstrate the model's exceptional accuracy in identifying accident scenes, even under challenging conditions. This promising performance suggests the potential for deploying such models in real-world applications, including integration within traffic monitoring systems for real-time accident detection, assisting autonomous vehicles in hazard avoidance, and the development of innovative accident prevention and emergency response strategies

II. LITERATURE SURVEY

Year	Authors	Papers	Inference	Outcome/Result	Limitation
2023	Ponnaganti Rama Devi, Dr. K. V. N. Sunitha	Efficient Vehicle Accident Detection System using CNN. Journal of Engineering Sciences, vol. 14, no. 30, pp. 1-12,.	The system utilizes deep learning convolutional neural network models trained to distinguish between accident and non-accident video frames . CNN-based image classifiers have been shown to achieve high accuracy levels of over 95% with less pre-processing compared to other algorithms	The proposed system utilizes deep learning convolutional neural network models to classify video frames into accident and non-accident categories with high accuracy levels of over 95% .	The paper does not provide specific details about the size or composition of the dataset used for training the CNN model, which could impact the generalizability of the results . There is no mention of the specific performance metrics used to evaluate the accuracy and effectiveness of the accident detection system .
2022	Mehta, K.; Jain, S.; Agarwal, A.; Bomnale, A	Road Accident Prediction Using Xgboost. In Proceedings of the 2022 International Conference on Emerging Techniques in Computational Intelligence, ICETCI 2022, Hyderabad, India, 25–27 August 2022.	The paper presents a road accident prediction system that uses a machine learning model, Xgboost, to forecast the severity of accidents based on variables such as day, weather, and road type. The system has achieved over 80	The road accident prediction system developed in the paper achieves more than 80The machine learning model Xgboost is implemented and found to be effective in predicting road accident severity .	The paper does not compare the performance of the Xgboost model with other machine learning models analyzed, such as Random Forest, Support Vector Machine, Stochastic Gradient Descent, and Artificial Neural Network.
2020	T. -N. Le, S. Ono, A. Sugimoto and	Attention R-CNN for Accident	The paper focuses on accident detection by simultaneously	The paper presents extensive experiments conducted on a newly	The paper does not provide specific details about the size

	H. Kawasaki	Detection 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020, pp. 313-320, doi: 10.1109/IV47402.2020.9304730.	detecting object class bounding boxes on roads and recognizing their status, such as safe, dangerous, or crashed. It introduces an attention mechanism called Attention R-CNN, which integrates global contexts from the scene to recognize object characteristic properties.	constructed dataset to evaluate the effectiveness of the proposed network for accident detection. The results demonstrate the effectiveness of the Attention R-CNN network in simultaneously detecting object class bounding boxes on roads and recognizing their status, such as safe, dangerous, or crashed.	or composition of the dataset used for training the CNN model, which could impact the generalizability of the results . There is no mention of the specific performance metrics used to evaluate the accuracy and effectiveness of the accident detection system .
2020	Renu, Durgesh Kumar Yadav, Iftisham Anjum, and Ankita.	Accident Detection using Deep Learning: A Brief Survey. International Journal of Electronics Communication and Computer Engineering, vol. 11, no. 3, pp. 16-25, May 2020..	Various systems have been proposed for automatic road accident detection, including the use of smart phones, vehicular Ad-Hoc networks, GSM and GPS technologies, and mobile applications .The paper highlights the importance of implementing an automatic road accident detection and information communication system in every vehicle .	The performance of the accident detection systems was measured using the accuracy metric, with a minimum accuracy of around 82% and a maximum accuracy of 97.76	The reliability of the hardware components, especially the sensors used in the accident detection system, is a limitation that can affect the accuracy of the overall system . The vanishing gradient problem in the Convolutional Neural Network (CNN) used for feature extraction can affect the accuracy of the system .
2022	H. Ghahremannezhad, H. Shi and C. Liu	Real-Time Accident Detection in Traffic Surveillance Using Deep Learning. 2022 IEEE International Conference on Imaging Systems and	The paper presents a new framework for accident detection in traffic surveillance at intersections using computer vision techniques, including object detection, object tracking, and accident detection based on trajectory conflict analysis. The proposed	The proposed framework achieved a Detection Rate of 93.10% and a False Alarm Rate of 6.89%. The performance of the framework was compared to other representative methods, and it showed superior results in terms of	The proposed framework focuses on accident detection at intersections and may not be applicable to other road segments or scenarios. The evaluation of the framework is primarily based on real traffic video data

		Techniques (IST), Kaohsiung, Taiwan, 2022, pp. 1-6, doi: 10.1109/IST55454. 2022.9827736.	method shows promising results in real-time applications with a low false alarm rate and a high detection rate.	detection rate and false alarm rate. The object detection and object tracking modules were implemented asynchronously to speed up the calculations.	and YouTube video sequences, which may not fully represent all possible real-world scenarios and variations in traffic conditions.
2021	Akshit Diwan, Vandit Gupta and Chaitanya Chadha	Accident Detection Using Mask R-CNN. International Journal for Modern Trends in Science and Technology, Vol. 07, Issue 01, January 2021, pp.- 69-72.	The paper focuses on using the Mask R-CNN approach to detect road accidents. This approach combines image segmentation and semantic segmentation for higher accuracy. Transfer learning is also used to repurpose a pre-trained model for accident detection.	The implementation of the Mask R-CNN approach in the paper successfully detects road accidents. The color of the bounding boxes involved in a collision changes from green to red when a collision is detected The combination of image segmentation and semantic segmentation in the Mask R-CNN approach improves the accuracy of accident detection .	Lack of real-world data: The paper does not mention the specific dataset used for training and testing the model, which raises concerns about the generalizability of the results . Limited evaluation metrics: The paper does not provide detailed information on the evaluation metrics used to assess the performance of the accident detection system.

III. PROPOSED WORK

A. Dataset Exploration and Limitations:

The "Accident Detection From CCTV Footage" dataset on Kaggle offered a promising starting point for our CNN-based accident detection project. Initially, the dataset's organization resembled the classic MNIST, with training, validation, and test folders containing image files. However, upon closer examination, we discovered that the images weren't single frames but likely captured accidents or non-accidents unfolding across multiple frames.

Despite not being a collection of isolated event frames, the dataset provides valuable benefits like:

Real-world Footage: The dataset uses real-world CCTV footage, which is more representative of actual accident scenarios compared to staged or controlled environments. This can improve the model's generalizability to real-world situations.

Labeled Data: The images are pre-labeled as "accident" or "non-accident," which streamlines the training process for the CNN model.

While the dataset offers advantages, it also presents limitations:Sequence Information Missing: The dataset likely doesn't explicitly capture the temporal sequence of frames within an accident or non-accident event.

To address these limitations, we considered:Frame Extraction where we analyse the video input and extract individual frames either continuously or with a lag, these frames represent key moments of accidents and non-accidents which the model will detect to classify the whole video as an 'accident' or a 'non-accident'.

B. Data Preprocessing:

The pre-processing utilizes TensorFlow's ImageDataGenerator class to preprocess the accident detection dataset. The primary functionalities are:

1. Normalization: The rescale=1/255 argument within ImageDataGenerator performs image normalization. Pixel intensities within each image are scaled from a range of 0-255 to a range of 0-1. This normalization facilitates the training process by establishing a consistent data format for the CNN.
2. Grayscale Conversion: The color_mode='grayscale' argument specifies the conversion of all images to grayscale format. This can be beneficial if color variations within the dataset are irrelevant to accident detection. Converting to grayscale reduces the dimensionality of the data, potentially improving training efficiency.
3. Resizing: The target_size=(img_height, img_width) argument resizes all images to a uniform size of img_height x img_width pixels, 48x48 pixels in ourcase. This ensures consistency within the dataset and allows the CNN to learn features more effectively across different images with the same dimensions.
4. Batching: The batch_size argument defines the batch size for the training data. Here, the data is grouped into batches of batch_size images 32 images in this case. Batching improves training efficiency by allowing the CNN to process data in manageable chunks, optimizing memory usage and computational resources.
5. By applying these preprocessing steps, the ImageDataGenerator prepares the accident detection dataset for training a CNN model. The normalized, potentially grayscale, and consistently sized images delivered in batches provide a suitable data format for the CNN to learn and extract relevant features for accident detection.

C. Model Building:

Convolutional Neural Network Architecture for Accident Detection: Our model leverages a pre-trained MobileNetV2 base model followed by a custom classification head designed for the specific task.

1. Pre-trained Model (MobileNetV2):The model utilizes a pre-trained MobileNetV2 model as its foundation. MobileNetV2 is a lightweight and efficient CNN architecture pre-trained on the ImageNet dataset for image classification. By leveraging the pre-trained weights, the model benefits from the feature extraction capabilities learned on a large dataset, even though the original classification task (ImageNet) differs from accident detection. The include_top=False argument during loading ensures that the final classification layers of MobileNetV2 are excluded, allowing us to build upon its feature maps for our custom classification head.The custom classification head, stacked on top of the pre-trained MobileNetV2, is responsible for learning features specific to accident detection.
2. Input Layer:input_ = tf.keras.layers.Input(shape=(48, 48, 1)): This defines the input layer, specifying that the model expects grayscale images with a size of 48x48 pixels and a single channel. This aligns with the image preprocessing step where images were converted to grayscale.
3. Convolutional Layers:Three convolutional layers (Conv2D) are employed to extract hierarchical features from the input images. Each layer uses a 3x3 kernel size, ReLU activation for non-linearity, and a stride of 2 for downsampling. The number of filters progressively increases across these layers (32, 64, and 128 filters respectively). These layers aim to learn increasingly complex spatial features relevant to accident detection within the images.
4. Dropout Layers:Dropout layers (Dropout) are inserted after each convolutional layer. These layers randomly drop a specific percentage of activations(outputs) from the previous layer during training. This helps prevent overfitting by forcing the model to not rely too heavily on any particular features. The dropout rates used in this architecture are 0.2, 0.2, and 0.5 for the respective convolutional layers.
5. Flatten Layer:x = tf.keras.layers.Flatten()(x): This layer transforms the output from the final convolutional layer (typically a 3D tensor) into a 1D vector. This is necessary before feeding the data into fully-connected layers for further processing.
6. Fully-Connected *Layers:*Two fully-connected (Dense) layers follow the flatten layer. These layers perform matrix multiplications to learn more intricate relationships between the flattened features. The first layer utilizes 64 neurons with ELU activation, while the final layer has 2 neurons with softmax activation.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

ELU (Exponential Linear Unit) activation is used in the dense layers. It offers similar benefits to ReLU (prevents vanishing gradients) but avoids the "dying ReLU" problem where some neurons might never activate.

The final layer with 2 neurons and softmax activation outputs a probability distribution over two classes, likely representing "accident" and "no accident." The softmax function ensures the probabilities for these two classes sum to 1. Fig. 1. shows the structure of the CNN model along with the input and output size for each layer in it.

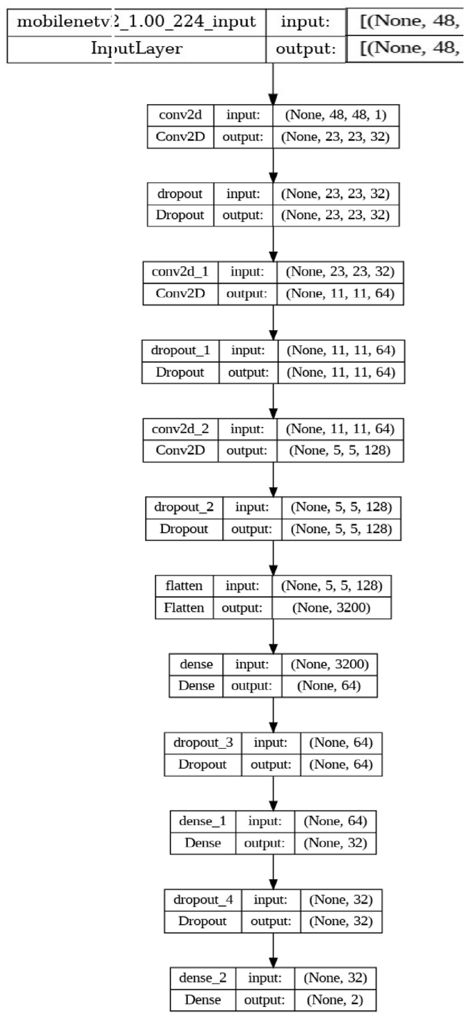

Fig.1. CNN architecture

In essence, this custom classification head refines the pre-extracted features from MobileNetV2 and learns higher-level abstractions that are discriminative for accident detection within the grayscale video frames. The combination of convolutional layers for feature extraction, Dropout for regularization, and fully-connected layers with ELU activation allows the model to learn a robust mapping from the input images to the probability of an accident scenario being present.

D. Model Compilation and Training Parameters:

The compiled model utilizes the Adam optimizer with a learning rate of 0.001 for gradient descent during training. Binary cross-entropy serves as the loss function for the binary classification task (accident vs. no accident). Accuracy is tracked as the primary metric to monitor classification performance. For training we employed the prepared training

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 5, May 2024

data and validation data for 50 epochs. Batch size is set to 64 images to balance efficiency with memory usage.Early stopping with a patience of 4 epochs is implemented to halt training if validation loss fails to improve. Additionally, it restores the model with the lowest validation loss, preventing overfitting.A learning rate reduction strategy using ReduceLROnPlateau is incorporated. If validation loss stagnates for 3 epochs, the learning rate is reduced by a factor of 0.1. This technique helps the model escape local minima and potentially improve convergence.

These parameters collectively govern the model's training process, aiming to optimize its performance for accident detection.

E. Results and Discussion:

The performance of the developed CNN model for accident detection in CCTV footage was assessed using a combination of metrics. These include monitoring the training and validation loss/accuracy curves during the training process.

The loss function graph in Fig.2 shows how the gap between the lines narrows as epochs increase. This suggests the model is learning well while maintaining some ability to generalize to unseen data (validation set).

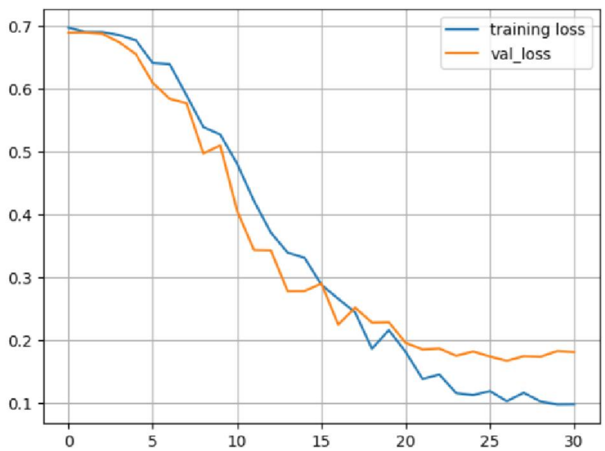

Fig.2. Loss function graph

And in the Fig. 3. Accuracy Graph, both the training accuracy and validation accuracy increase as the number of epochs progresses. This indicates that the model's performance on both the training data and the validation data is improving as it trains.

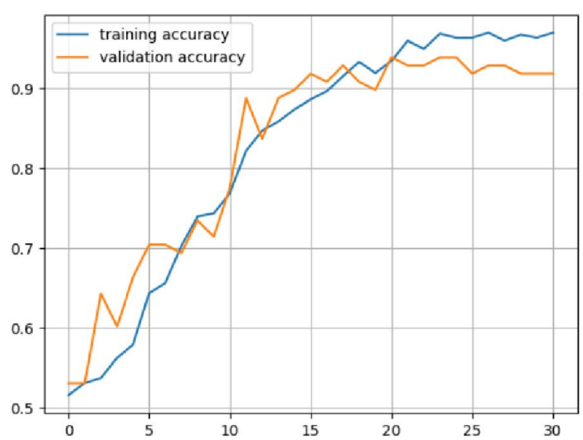

Fig.3. Accuracy Graph

Additionally, the final evaluation on the testing dataset yielded a loss value of 0.1907 and an accuracy of 0.9200, further indicating the model's effectiveness in classifying accident scenarios.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

Also, the model's output format aligns with the binary classification task. It assigns a probability value between 0 and 1 for each input frame. A probability closer to 1 signifies a predicted accident scenario, while a value closer to 0 indicates no accident.

Evaluation on the testing dataset yielded an accuracy of 92%. This suggests the model can effectively distinguish between accident and non-accident frames.

IV. USER INTERFACE FOR MODEL INTERACTION

To facilitate seamless integration of the developed convolutional neural network (CNN) model into real-world applications, particularly those involving resource-constrained devices, the model is exported into a TensorFlow Lite (TFLite) format. Following conversion to the TFLite format CNN model is integrated with a frontend application. This frontend facilitates user interaction. Streamlit, a Python library used to create web applications from code, is employed to construct this user interface.This frontend empowers users to analyze both image and video data for potential accident detection:

1. *Image-Based Analysis*: Users can directly upload image files depicting accident scenes. The frontend seamlessly transfers the uploaded image to the TFLite model for prediction.
2. *Video-Based Analysis:* For video analysis, the frontend uses OpenCV, a computer vision library. It extracts frames from the uploaded video at an adjustable interval. Each extracted frame undergoes appropriate scaling to match the dimensions that the TFLite model was trained on. This ensures compatibility and accurate predictions. The scaled frames are then fed to the TFLite model for real-time inference, enabling the frontend to provide analysis of an accident at various scenarios within the video.

Prior to interaction with the TFLite model, a critical step is taken for both images and video frames – image scaling. The frontend resizes the images to precisely match the input format expected by the model. This ensures compatibility and facilitates accurate predictions from the TFLite model.

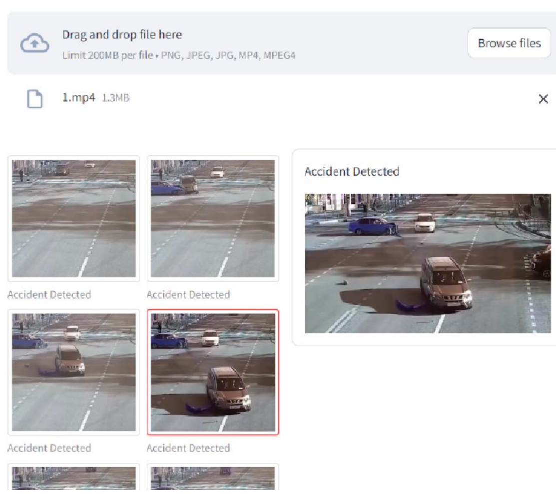

Fig.4. User Interface

The Fig. 4. Shows the frontend/interactive interface for this model, where an mp4 file has been passed and frames are chosen from the video with an interval to then pass onto the model where it classifies the image as “Accident Detected” or “No Accident Detected.”

In summary, the Streamlit based frontend offers a user-centric platform for accident detection. Users can upload images or videos, and the frontend handles the communication with the TFLite model. The frontend delivers real-time predictions for accident scenarios within the provided content, making way for further advancements in real-time accident detection systems.

V. CONCLUSION

This work investigated the development of a Convolutional Neural Network (CNN) model for accident detection in CCTV footage. The model leveraged a pre-trained MobileNetV2 base for feature extraction, followed by a custom classification head designed for the specific task. The grayscale video frames were preprocessed for efficient training and fed into the model.

The compiled model employed the Adam optimizer and binary cross-entropy loss function suitable for the binary classification of accident scenarios (accident vs. no accident). Training was conducted for 50 epochs with early stopping and learning rate reduction techniques to prevent overfitting and enhance generalization.

Evaluation on the testing dataset yielded an accuracy of 92%. This indicates that the model can effectively classify video frames, assigning a binary output probability – 1 for a predicted accident and 0 for no accident – with a high degree of accuracy. This promising performance suggests the potential of the CNN architecture for real-world accident detection applications.

To facilitate real-world integration, the model was converted to a TensorFlow Lite (TFLite) format. TFLite's reduced size and optimization for mobile and embedded devices enable deployment on resource-constrained devices, expanding potential applications. Furthermore, a user-friendly frontend application was developed using Streamlit. This frontend empowers users to interact with the TFLite model for both image and video analysis, delivering real-time predictions for accident scenarios. This user-centric approach broadens model accessibility and paves the way for real-world impact.

Future work may involve exploring different pre-trained models or network architectures to potentially improve accuracy and robustness. Additionally, incorporating techniques for handling imbalanced datasets, if present, could further enhance model performance.

REFERENCES

[1]. Ponnaganti Rama Devi, Dr. K. V. N. Sunitha. "Efficient Vehicle Accident Detection System using CNN." Journal of Engineering Sciences, vol. 14, no. 30, pp. 1-12, 2023.

[2]. Mehta, K.; Jain, S.; Agarwal, A.; Bomnale, A. Road Accident Prediction Using Xgboost. In Proceedings of the 2022 International Conference on Emerging Techniques in Computational Intelligence, ICETCI 2022, Hyderabad, India, 25–27 August 2022.

[3]. T. -N. Le, S. Ono, A. Sugimoto and H. Kawasaki, "Attention RCNN for Accident Detection," 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020, pp. 313-320, doi: 10.1109/IV47402.2020.9304730.

[4]. Renu, Durgesh Kumar Yadav, Iftisham Anjum, and Ankita. "Accident Detection using Deep Learning: A Brief Survey." International Journal of Electronics Communication and Computer Engineering, vol. 11, no. 3, pp. 16-25, May 2020.

[5]. H. Ghahremannezhad, H. Shi and C. Liu, "Real-Time Accident Detection in Traffic Surveillance Using Deep Learning," 2022 IEEE International Conference on Imaging Systems and Techniques (IST), Kaohsiung, Taiwan, 2022, pp. 1-6, doi:10.1109/IST55454.2022.9827736.

[6]. Akshit Diwan, Vandit Gupta and Chaitanya Chadha, "Accident Detection Using Mask R-CNN", International Journal for Modern Trends in Science and Technology, Vol. 07, Issue 01, January 2021, pp.- 69-72.

[7]. Kabir, M. F., & Roy, S. (2022). Real-time vehicular accident prevention system using deep learning architecture. Expert Systems with Applications, 206. 10.1016/j.eswa.2022.117837

[8]. Kattukkaran N. George A. Haridas T. P. M. (2017). Intelligent accident detection and alert system for emergency medical assistance. 2017 International Conference on Computer Communication and Informatics (ICCCI),1- 6.10.1109/ICCCI.2017.8117791