Automatic Road Accident Detection using Deep Learning

Gopika Rani N[1], Adarsh G[2], Ajay Deepak P M[3], Elanthamil R[4], Jeevan Krishna K V[5], Nirmal M[6]

[1, 2, 3, 4, 5, 6]Department of Computer Science and Engineering, PSG College of Technology, Coimbatore 641004, India

[1]ngr.cse@psgtech.ac.in

[2]adhu1342@gmail.com

[3]ajaydeepak370@gmail.com

[4]elanthamil2003@gmail.com

[5]jeevanvijaylatha@gmail.com

[6]muthukumaresannirmal@gmail.com

Abstract. Accident detection plays an important role in ensuring road safety and providing an immediate response that can significantly reduce the loss of lives. A deep learning model is proposed for efficient accident detection. The initial phase involves the careful curation of a diverse dataset to train models that can autonomously recognize accidents, ensuring precise detection through rigorous optimization. Trained models, adapt at discerning nuances and trigger critical responses, including immediate alerts and coordination with emergency services, thereby enhancing effectiveness in critical situations. A crucial feature includes alerting via the Simple Mail Transfer Protocol (SMTP). In the event of an accident, the system seamlessly interfaces with SMTP, enabling swift communication and prompt notification to the emergency services. The proposed model provides an accuracy of 95%, representing a higher level of accuracy in detecting accidents. Thus, the integrated approach represents a significant leap in road accident detection.

Keywords: Bilinear interpolation, Bounding box, Deep learning, Object localization, Pipeline, Sparse categorical cross entropy.

1 Introduction

In contemporary times, the intricate web of roadways and highways serves as the backbone of global transportation infrastructure. However, the remarkable benefits of this interconnected system come with an inherent set of challenges. Road accidents and their repercussions have cast a shadow on the transportation systems. The advent of deep learning has brought forth a promising solution to address these challenges. These repercussions extend beyond immediate physical harm to encompass profound economic losses, standing as a cause for deep concern. As each accident carries the potential for tragic human consequences, the urgency to implement a robust and efficient system for road accident detection becomes paramount. The development of an efficient system for road accident detection is imperative, not only to identify accidents but also to notify emergency services. In light of these compelling factors, this study emphasizes the pressing need to address the growing imperative for enhanced road safety through the implementation of a sophisticated accident detection system. Deep learning is poised to revolutionize accident detection. Its exceptional capacity to process vast visual data in real time enables accurate and swift accident detection. Through the deep learning models like Convolutional Neural Networks (CNN) and You Only Look Once (YOLO), it surpasses several machine learning models, extracting high-level features from raw input. This empowers the system to identify potential accidents. Trained on extensive datasets, these models excel in recognizing patterns under varying weather, lighting, and traffic conditions. This highlights the potential of deep learning in enhancing accident detection.

Automatic Road Accident Detection using Deep Learning presents a comprehensive approach for better detection of accidents. By seamlessly integrating accident detection with traffic monitoring, the response times can be reduced significantly and provide enhanced traffic flow. By leveraging deep learning algorithms, these systems can analyze real time data to make dynamic decisions, ensuring a swift and effective response to accidents. Several research studies have used computer vision methods in traffic surveillance systems for diverse traffic monitoring applications [1-10]. Additionally, they can provide notifications to emergency services using SMTP, guiding them towards the affected areas. This study represents a significant step forward in creating smarter and more responsive roadways by effective road accident detection.

The following section introduces the related works associated with this study. The third section explains the dataset used for the automatic road accident detection. The fourth section of this study details the implementation of the proposed model. In the fifth section, the results are presented and compared with other deep learning algorithms. Finally, the sixth section concludes this study and briefs the future work.

2 Related Works

Ariba Zahid et al. created fake accident video frames from normal video traffic footage, manually constructing them by simulating accidents. AlexNet outperformed other models providing an 80% true positive rate when detecting accidents in real-world surveillance videos [18]. Tiago Tamagusko et al. delved into the application of Deep Learning techniques for road accident detection, employing Transfer Learning and Synthetic Images. The survey discusses the significance of their methodology, its implications in improving road safety. Model FI_C0166000_MobileNetV2 (based on MobileNetV2) had an mAP of 0.87 and an MCC of 0.68 [11]. Xiaozhou Liu et al. proposes an approach to convert speed time series data into images in which the Gramian Angular Difference Fields (GADF) and Piecewise Aggregate Approximation (PAA) algorithms are employed. CNNs are then used to extract traffic features from these images, framing incident detection as a binary classification problem. This research contributes to the advancement of traffic incident detection methods, emphasizing the potential of CNNs for improving accuracy and efficiency in real-time traffic management systems [17]. G. Rajesh et al. utilize CNNs and recurrent neural networks (RNNs) to process image and sensor data, enhancing real-time accident identification accuracy. Their work achieved a peak accuracy of 85% [12].

D K Yadav et al. proposes a system for accident detection using a dataset containing approximately 1730 videos, which are categorized into accident and non-accident videos. The 3D-RGB values are converted into grayscale values to reduce dimensionality. The study employs Hierarchical RNN which analyzes the sequence of images and detecting the probability of the occurrence of accidents and the Long Short-Term Neural Network which handles the negligent change in weight values of the neural network which achieved an overall accuracy of 92% [19]. Daxin Tian et al. introduces an automatic car accident detection method that leverages Cooperative Vehicle Infrastructure Systems (CVIS) and machine vision. The authors develop a deep neural network model, YOLO-CA, integrating Multi-Scale Feature Fusion (MSFF) and dynamic-weight loss functions to improve small object detection. YOLO-CA has the ability to detect car accidents in just 0.0461 seconds (21.6 FPS) with 90.02% average precision [16]. Deeksha Gour et al. uses the YOLO algorithm to detect moving objects and collisions in live camera feeds, sending emergency alerts to nearby authorities via wireless communication devices [15]. Junliang Li et al. proposes a successful approach called Region-Based Convolutional Network (R-CNN) for effective region-based feature extraction. They also highlight the suitability of part filters in a Deformable Part-Based Model (DPM) for detecting occluded objects.This study achieves highly accurate single-object detection by integrating the two models. Experimental results on the Pascal Visual Object Classes (VOC) dataset confirm that this approach outperforms using R-CNN or DPM individually for multiple object detection [20].

Yang et al. introduces a Deep Convolutional Neural Network (DCNN) that uses vehicle trajectory data to detect and classify six types of traffic incidents. However, the DCNN prioritizes local characteristics inside individual frames and ignores temporal features across video frames [21]. Bongjin Oh et al. combines two CNNs inorder to train, recognize, or extract scene images, and various items in the images can be identified and stored based on scene classes. This hybrid CNN outperforms Places365-ResNet by 3% in terms of top -5 accuracy [22]. J. Redmon et al. discuss the limitations of existing object detection methods and present the key features and advantages of YOLO, including its speed, accuracy, and ability to encode contextual information [14]. A. Tsuge et al. have conducted several studies to develop and evaluate accident detection systems. Under normal traffic flow conditions, the detection rate exceeded 90% during daytime, twilight, or night time. However, during traffic congestion, the reported traffic volume was lower than the actual because of overlapping vehicle images, making it difficult for the system to identify them as individual vehicles [13]. Some of the common system designs consist of a microcontroller, GPS, and a group of sensors to determine different physical parameters related to vehicle motion [23-25].

The main contribution of this study is to train and test a detection model for fast and efficient detection of accidents which will be helpful for faster emergency relief measures. The drawback of the CNN model is that it performs multiple passes which is time consuming that can be avoided. The complexity of R-CNN and DCNN models can be avoided. Moreover, the drawback of dependency on motion sensors can be overcome.

3 Dataset

In this section, the dataset used for this study is discussed. The dataset for the deep learning models has been prepared, and essential data pertaining to both accident and no accident scenarios is provided for the training, validation, and testing phases of the deep learning models.

3.1 Dataset Description

The dataset from Kaggle [26] which encompasses a rich array of photographs (CCTV video frames), providing a diverse and representative sample set for comprehensive training. The dataset revolves around two fundamental categories: "Accident" and "No Accident".

The dataset comprises a total of 990 files, all in the widely supported Joint Photographic Experts Group (JPEG) format. Within the training dataset, a balanced distribution is maintained, featuring 432 no accident files and 379 accident files. The validation dataset includes 63 no accident images and 56 accident images. The test dataset consists of 28 no accident images and 32 accident images. Each object of interest within the dataset must be accurately described and labeled. The bounding boxes define the spatial coordinates of each object within the CCTV video frame, specifying its exact position, size, and orientation. By providing these coordinates, the model can identify the location of objects of interest within an image.

3.2 Dataset Preparation for YOLO model

The dataset [26] used for training the model which was initially in its raw, unprocessed form, necessitating a comprehensive data preparation and annotation process.

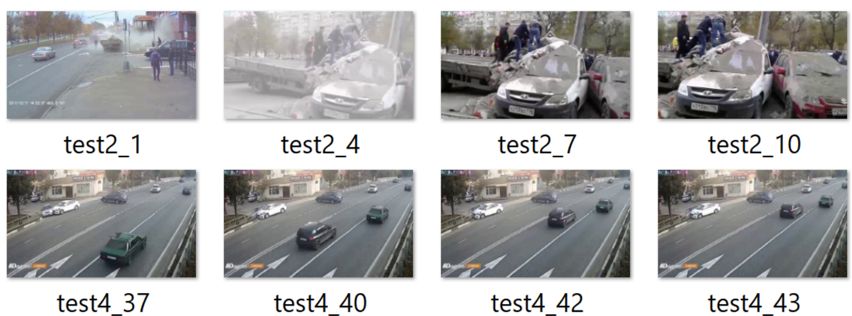

Fig. 1. Unlabelled Images

1. *Unlabeled Data.* Unlabeled data for accident detection refers to raw, not annotated information, lacking specific indications of accidents. The raw data in Fig. 1 is unlabeled. This data has to be converted into bounding box vectors to train the YOLO model.
2. *Drawing Bounding Boxes.* Drawing bounding boxes for accident detection involves visually describing the region of interest, typically an accident or no accident, within a CCTV video frame. This method is widely employed to annotate and train deep learning models for precise accident detection. In this context, the model is tasked with detecting bounding boxes around no accident and accident scenes.

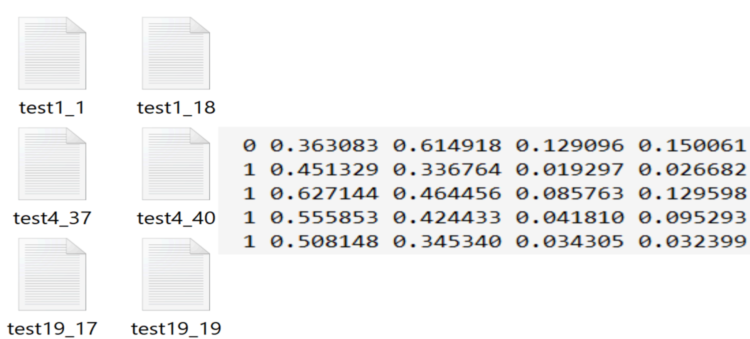

Fig. 2. Labeled Data of Images

3. *Labeled Data.* In Fig. 2, each image in the dataset is associated with a corresponding text file containing vital information. Each label indicates whether the object represents an accident (denoted as "0") or no accident (denoted as "1"). Following this classification, the next two values specify the coordinates of the bounding box's top-left corner, denoting the x and y coordinates. Subsequently, the width and height of the bounding box are provided as the next two values, encapsulating the object's spatial dimensions within the image.

3.3 Dataset Preparation for CNN model

The dataset [26] is preprocessed using bilinear interpolation and resized to 250x250x3. Bilinear interpolation involves considering the values of the surrounding pixels and performing a weighted average to approximate the value at the desired position. The formula used for bilinear interpolation is as follows:

$$I(x, y) = (1 - u) * (1 - v) * I(x_1, y_1) + u * (1 - v) * I(x_2, y_1) + (1 - u) * v * I(x_1, y_2) + u * v * I(x_2, y_2) \quad (1)$$

The variables ‘u’ and ‘v’ are the interpolation coefficients, which are calculated as follows:

$$u = \frac{(x - x_1)}{(x_2 - x_1)} \quad (2)$$

$$v = \frac{(y - y_1)}{(y_2 - y_1)} \quad (3)$$

where $I(x, y)$ is the interpolated value at the point (x, y). The points (x_1, y_1), (x_2, y_1), (x_1, y_2), and (x_2, y_2) are the positions of the surrounding pixels.

4 Implementation

This section shows the implementation of the automatic road accident detection. The accident detection system utilizes the YOLO algorithm as an object detection algorithm, and its results are subsequently pipelined into a CNN model for further refinement of the detection.

4.1 Architecture for Accident Detection

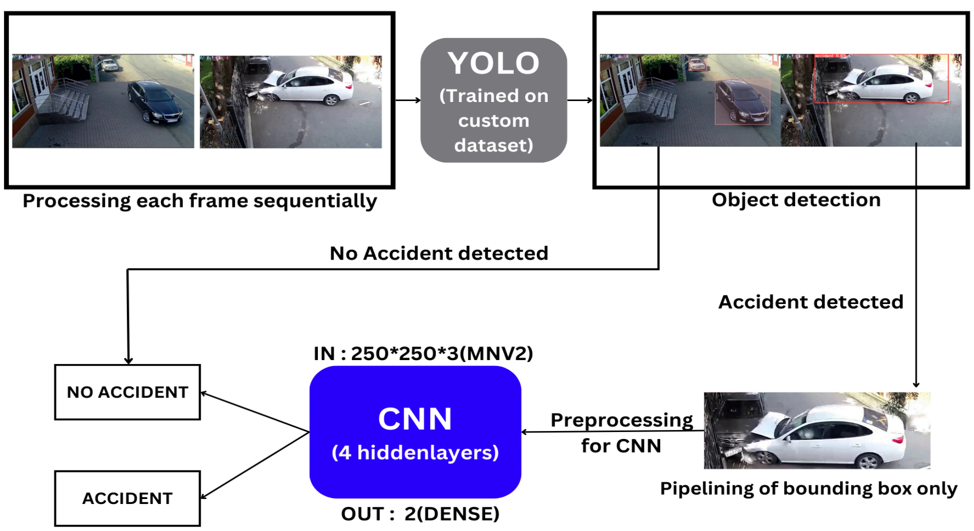

Fig. 3. Architecture of the Proposed Model

The architecture of the proposed accident detection model is shown in Fig. 3, where the initial input to the YOLO object detection algorithm is the CCTV video stream, in which each frame is processed sequentially. This YOLO model provides fast object detection and is trained on a custom dataset. This training helps YOLO model to detect objects in each frame from the CCTV video stream. After successful identification of objects in the frame by the YOLO model, the objects are detected for accidents. If the YOLO model detects no objects as an accident the frame is not processed further and classified as “No Accident”. If the YOLO model detects objects as an accident, the frame is processed further with pipelining to the CNN model. The bounding box of the objects detected as an accident from the frame is extracted and passed to the CNN model. These accident detected objects are then fed into a 6 - layer CNN model. This CNN model has been meticulously trained on a diverse and comprehensive dataset, encompassing a wide array of accident and no accident scenarios. This comprehensive training enables CNN model to make informed decisions when classifying the bounding boxes of the accident detected objects as "Accident" or "No Accident". The YOLO and CNN models are combined to reduce the misclassification of "No Accident" as "Accident" by the YOLO model while maintaining fast detection. Thus by combining the capabilities of the YOLO model for object detection and the CNN model for classification, the proposed model can improve the performance by reducing false positives of the YOLO model.

4.2 YOLO model for Accident Detection

The YOLO model provides exceptional speed and efficient object localization of CCTV video frames. Unlike traditional methods that require multiple passes through an image, the YOLO model uses a single-pass approach that significantly reduces processing time for accident detection. The YOLO model may produce false positives in accident detection. These false positives can arise from challenging lighting conditions, occlusions, or other confounding elements present in the CCTV video frame. To address this, a complementary approach is implemented by combining the YOLO model with the CNN model, in a pipelining framework.

1. *Training of YOLO Model*. The YOLO model is trained on a labeled custom dataset. The pre-trained model used is YOLOv5s. The specifications for the training environment are as follows:
   - Epochs: 100
   - Data: customdata.yaml
   - Weights: yolov5s.pt
   - Cache: True

   The precision, recall and mAP almost becomes constant at 100 epochs. The file "customdata.yaml" contains information about the custom classes and storage paths of the images used. The file "yolov5s.pt" contains the weights of the pretrained YOLOv5s model. The cache is set to "True" to increase the speed of training the model. Fig. 4 shows the metrics during the training phase of the YOLO model. The precision, recall and mAP is observed to increase as the training progresses, while the box loss, object loss and class loss decreases. The final box loss, object loss and class loss values are 0.028207, 0.025233 and 0.0020997 respectively while the final precision, recall, mAP0.5 and mAP0.5/0.95 are 0.77676, 085656, 0.84299 and 0.49212 respectively.

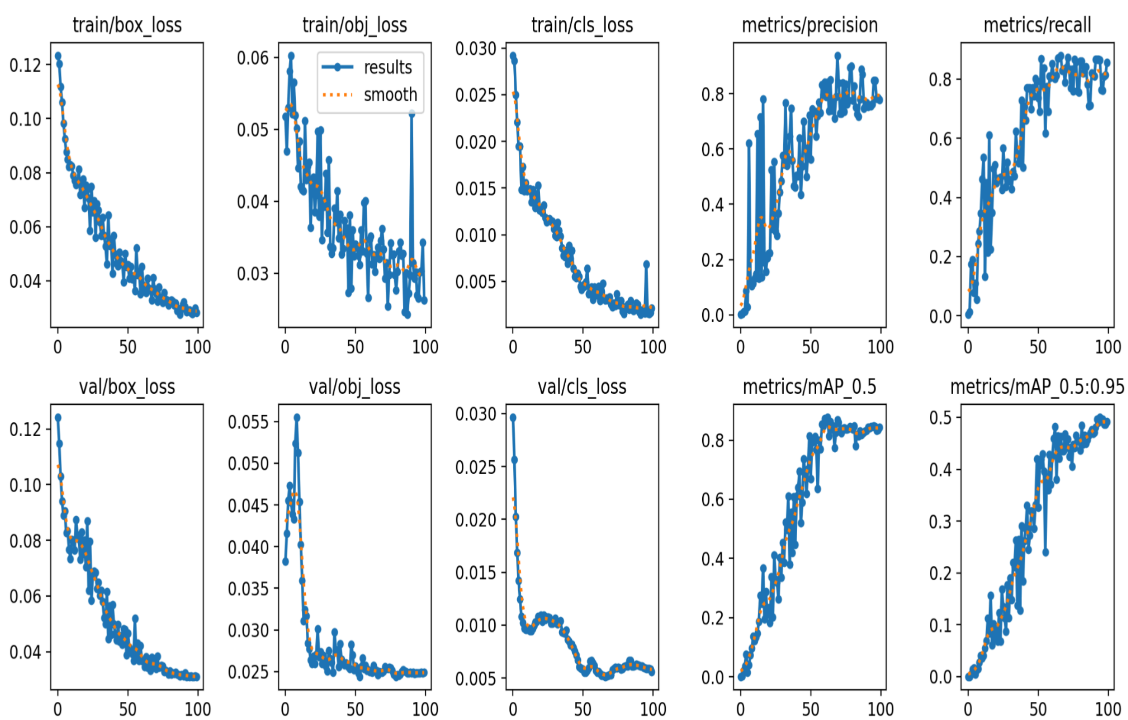

Fig. 4. Results of the YOLO model during the training phase

4.3 CNN model for Accident Detection

The CNN model operates on a finer scale, scrutinizing the regions of accidents in a frame detected by the YOLO model with a higher level of granularity. This fine-grained analysis enables the CNN model to provide a more nuanced assessment of potential accident scenarios, significantly reducing the occurrence of false alarms. By harnessing the effective processing capabilities of the YOLO model and the detailed feature analysis of the CNN model, this pipeline not only significantly reduces false positives but also enhances the overall accuracy and reliability of accident detection.

1. *Structure of CNN model.* The structure of the CNN model has 6 layers. Pretrained MobileNetV2 is used as the input layer. The Output layer is a dense layer that outputs the classification result as “Accident” or “No accident”. The six layers are:
   - Input Layer
     - Input dimension : 250x250x3
     - Output dimension : 8x8x1280
   - Convolutional Layer 1
     - Input dimension : 8x8x1280
     - Output dimension : 6x6x32
   - Convolutional Layer 2
     - Input dimension : 6x6x32
     - Output dimension : 4x4x64
   - Convolutional Layer 3
     - Input dimension : 4x4x64
     - Output dimension : 2x2x128
   - Flatten Layer
     - Input dimension : 2x2x128
     - Output dimension : 512
   - Dense Layer
     - Input dimension : 512
     - Output dimension : 2
2. *Training of CNN model.* To train a CNN model, a custom labeled dataset is used. Subsequently, the images are processed and partitioned into training, validation, and test sets. A CNN model structure is formulated and paired with a loss function and optimizer. The model undergoes training, refinement, assessment, and is eventually deployed for the accident detection. The CNN model undergoes training for 50 epochs using the dataset. The accuracy of the model becomes constant at 50 epochs. The chosen loss function is sparse categorical cross entropy, and accuracy is employed as the training metric.

4.4 Pipelining of models

The YOLO model is utilized to detect objects in the provided CCTV video frames. Upon the YOLO model's detection of objects, they undergo assessment for accidents. If the YOLO model discerns no objects indicative of an accident, the frame is not subject to further processing and is labeled as "No Accident". Conversely, if the YOLO model detects objects as part of an accident, the frame undergoes additional processing through pipelining to the CNN model. It generates coordinates for bounding boxes encapsulating the detected accident. It is crucial to acknowledge that bounding boxes may have varying dimensions, but the CNN model can only process inputs in a specific dimension, which is 250x250x3 in this case. Hence, bounding box image snippets must be preprocessed into an RGB (Red, Green and Blue) array of the specified dimensions. The bounding box, labeled with a class 0 (indicating an accident), is then fed into the CNN model after conversion into an image format suitable for processing. If the CNN model predicts a value of 0, it confirms the presence of an accident in the image. Conversely, if the prediction is 1, it signifies the absence of an accident.

4.5 Alert System

The alert system serves the purpose of notifying the emergency services about the accidents, by providing essential details such as location, co-ordinates, date and time. In this system, SMTP is employed to alert the emergency services. To mitigate the occurrence of false alarms stemming from brief overlapping of vehicles, which could be erroneously classified as accidents by the proposed model, a threshold is implemented. This threshold denotes the number of consecutive frames that must be classified as “Accident” to trigger the alert system.

Algorithm 1 Implementation of the Proposed Model

Input:
- CCTV video stream
- Custom labeled image dataset
- Pretrained models : MobileNetv2 , YOLOv5s

Output:
- Proposed Model Predictions: “Accident” or “No Accident”

Steps:

```
1:  Train the YOLOv5s model on the training dataset and store the learned weights.
2:  Train the CNN model on the training dataset and store the learned weights.
3:  Load the custom trained YOLO model weights and custom trained CNN model weights which were learned
during the training phase
4:  Initialize frames, detections, count
5:  frames = capture_frame(CCTV video stream)
6:  for i ←1 to length(frames) do
        /*Compute object detections class and bounding boxes*/
7:     detections[i] ← pass each frame into the YOLO model
8:       for j ←1 to length(detections[i]) do
              /*Only the Accident bounding boxes are pipelined*/
9:            if detections[i][j]=="Accident" then
10:              Resize bounding box into 250x250 and perform bilinear interpolation
                   /*Compute the CNN model prediction*/
11:               detections[i][j] ← pass the bounding box to the CNN model
12:         end if
13:      end for
        /*Check if Accident class is present in a frame*/
14:      if  “Accident” in detections[i] then
15:           count++
16:      else
17:              count = 0
18:      end if
      /*Accident detection in subsequent frames within one second indicates the presence of accident*/
19:      if count == number of frames in a second then
20:           return “Accident”
21:      end if
22:  end for
      /*If No Accident is detected in subsequent frames within one second*/
23:  return “No Accident”
```

5 Results

In this section, the results of the proposed model for detecting accidents is presented and compared with the other deep learning algorithms.

5.1 Results of the Proposed Model

The performance of the proposed model for accident detection is presented. Table 1 shows the time per frame for the detection process of “No Accident” and “Accident”. For “No Accident”, YOLO model detects directly without the need for pipelining, so time per frame is less when compared to time per frame with “Accident”. The increase in time per frame for detecting “Accident” when compared to that of “No Accident” is because of the pipelining of YOLO and CNN models. The CNN model takes a higher time per frame thereby increasing the overall time per frame of the system. However, accidents are an infrequent occurrence hence the occasional jump in time per frame. The tradeoff of increase in the time per frame in a real time application is small when compared to the accuracy achieved by combining the models. The YOLO model has a convincing mAP of 0.49. The accuracy achieved by the CNN model is 0.95. The accuracy achieved by the proposed model is also 0.95, even though the accuracy achieved by the CNN model and the proposed model is the same, but the time taken by the CNN model is 1.4 times greater than the proposed model. Therefore, an increase in time per frame in the proposed model due to the pipelining process is worth the tradeoff. Fig. 5 shows the detection of false positives by the YOLO model. The class of the bounding box is "No Accident". But the YOLO model classifies the bounding box as an "Accident". False positives can trigger false alerts to emergency services. In order to overcome this issue, pipelining with CNN model is performed when YOLO model detects an "Accident".

The "Accident" class bounding box in Fig. 5 detected by the YOLO model is passed to the CNN model. The CNN model classifies the bounding box as "No Accident" and the result is shown in Fig. 6. Thereby preventing the possibility of false positives. The confidence value of each bounding box denotes the probability of the class within the bounding box. It is mentioned beside the class of each bounding box in Fig. 5 and Fig. 6.

Table 1. Time taken for each frame by the Proposed model for no accident and accident detection

Sequence of Frames	Class	Time taken for detection
1	No Accident	224.4 ms
2	Accident	923 ms
3	No Accident	233.6 ms
4	Accident	924 ms

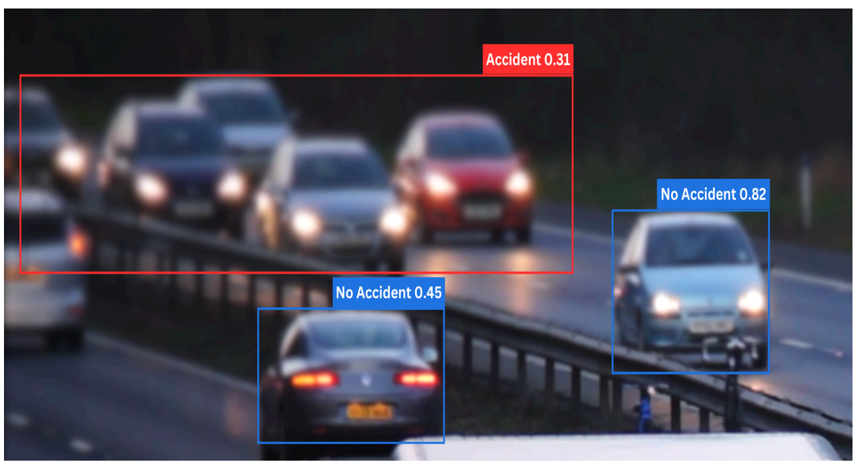

Fig. 5. False positive detected by the YOLO model

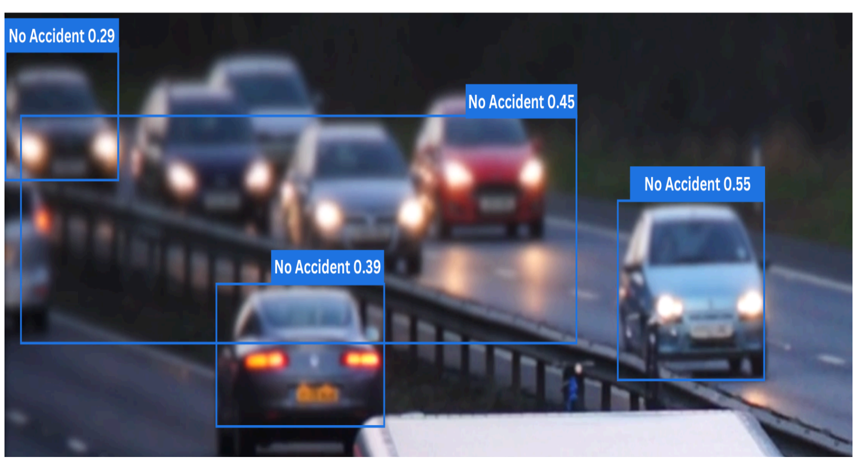

Fig. 6. Correction of Classification by the CNN model

The model is tested on 60 images. The confusion matrix of the results predicted by the proposed model is shown in Fig. 7. Analysis of Fig. 7 reveals that the model predicted 31 "Accident" cases and 2 "No Accident" cases as "Accident" cases. Furthermore, there were 1 false negative case and 26 true negative cases. The classification report for the proposed model was constructed, as shown in Table 2. The accuracy, precision, recall and F1 score of the proposed model are 0.95, 0.93939, 0.965384 and 0.95384. Fig. 8 shows the notification received when an Accident is detected with the help of the SMTP. The message includes location, co-ordinates, date and time of the accident.

Table 2. Classification report of the proposed model

	Precision	**Recall**	**F1 Score**	**Support**
Accident	0.94	0.97	0.95	32
No Accident	0.96	0.93	0.95	28
accuracy			0.95	60
macro avg	0.93939	0.96875	0.95384	60
weighted avg	0.95	0.95	0.95	60

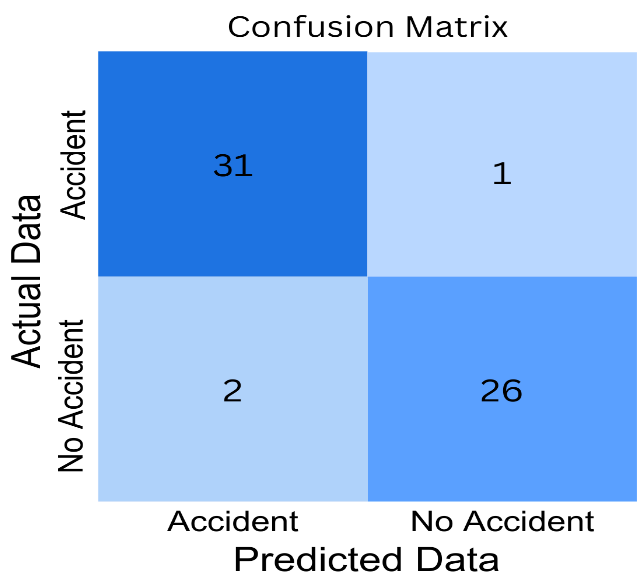

Fig. 7. Confusion matrix of results predicted by the proposed model

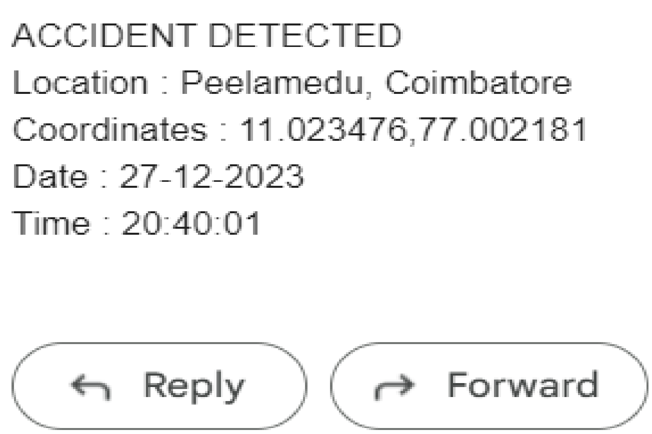

Fig. 8. Notification when Accident is detected

5.2 Comparison with other Deep Learning Algorithms

The comparison between the proposed model and other deep learning models that includes FI_C0166000_EfficientNetB1, CNN (9 layers), and CNN with LSTM is performed and shown in Table 3. Among other deep learning models the CNN with LSTM model provided a maximum accuracy of 91.38%, while the proposed model achieved a better accuracy of 95%. It can be inferred that the proposed model achieves the highest accuracy when compared to other deep learning models. Overall, the proposed model shows a significant improvement in accuracy when compared to other deep learning models. Thus, from Table 3, it can be inferred that the proposed model provides a better performance and better prediction compared to the other deep learning models.

Table 3. Comparison between accuracy of various Deep Learning models with the proposed model

Models	Accuracy
FI_C0166000_EfficientNetB1[11]	0.88
CNN (9 layers) [12]	0.85
CNN with LSTM [19]	0.9138
Proposed model	0.95

6 Conclusion

The integration of YOLO and CNN models in a pipelining framework will help improve the efficiency in detecting road accidents. This approach leverages the capabilities of the YOLO model and the detailed feature analysis of CNN model to significantly reduce false positives and enhance the overall accuracy of accident detection. The performance of the pipelining approach is rigorously evaluated across diverse scenarios and environmental conditions. This will help identify areas for improvement and fine-tuning, ensuring the system's effectiveness in a wide range of contexts. The proposed model provides better detection compared to other deep learning algorithms. The future work can be performed by the integration of other deep learning algorithms that could further enhance the system's adaptability and decision making capabilities. These techniques could increase the model's ability to handle more complex scenarios. The scalability of the system can be explored, particularly in the context of handling large volumes of video data from multiple cameras. As part of the future work, integrating with other communication protocols can also be considered to improve the alert system.

Acknowledgement

This work was supported by PSG College of Technology, India. The authors would like to sincerely thank all the reviewers for their valuable feedback regarding this work.

References

1. H. Shi, C. Liu, (2018) A New Foreground Segmentation Method for Video Analysis in Different Color Spaces, 24th International Conference on Pattern Recognition (ICPR), Beijing, China, pp. 2899-2904, doi: 10.1109/ICPR.2018.8545500
2. G. Liu et al, (2022) Smart Traffic Monitoring System Using Computer Vision and Edge Computing, IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 12027-12038, doi: 10.1109/TITS.2021.3109481
3. H. Ghahremannezhad, H. Shi, C. Liu, (2020) Automatic Road Detection in Traffic Videos, IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Exeter, United Kingdom, pp. 777-784, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00123
4. H. Ghahremannezhad, H. Shi, C. Liu, (2020) A New Adaptive Bidirectional Region-of-Interest Detection Method for Intelligent Traffic Video Analysis, IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Laguna Hills, CA, USA, pp. 17-24, doi: 10.1109/AIKE48582.2020.00012
5. H. Ghahremannezhad, H. Shi, C. Liu, (2020) Robust Road Region Extraction in Video Under Various Illumination and Weather Conditions, IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), Genova, Italy, pp. 186-191, doi: 10.1109/IPAS50080.2020.9334959
6. H. Shi, H. Ghahremannezhadand, C. Liu, (2020) A Statistical Modeling Method for Road Recognition in Traffic Video Analytics, 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Mariehamn, Finland, pp. 000097-000102, doi: 10.1109/CogInfoCom50765.2020.9237905

7. H. Ghahremannezhad, H. Shi, C. Liu, (2020) A Real Time Accident Detection Framework for Traffic Video Analysis, 16th International Conference on Machine Learning and Data Mining, Amsterdam, Netherlands, pp. 77–92
8. M. O. Faruque, H. Ghahremannezhad, C. Liu, (2019) Vehicle classification in video using deep learning, 15th International Conference on Machine Learning and Data Mining, New York, USA, pp. 117–131
9. H. Ghahremannezhad, H. Shi, C. Liu, (2021) A New Online Approach for Moving Cast Shadow Suppression in Traffic Videos, IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, pp. 3034-3039, doi: 10.1109/ITSC48978.2021.9565049
10. H. Shi, H. Ghahremannezhad, C. Liu, (2021) Anomalous Driving Detection for Traffic Surveillance Video Analysis, IEEE International Conference on Imaging Systems and Techniques (IST), Kaohsiung, Taiwan, pp. 1-6, doi: 10.1109/IST50367.2021.9651372
11. Tiago Tamagusko, Matheus Gomes Correia, Minh Anh Huynh, Adelino Ferreira, (2022) Deep Learning applied to Road Accident Detection with Transfer Learning and Synthetic Images, Transportation Research Procedia, vol. 64, pp. 90-97, doi: 10.1016/j.trpro.2022.09.012
12. G. Rajesh, A. R. Benny, A. Harikrishnan, J. Jacob Abraham, N. P. John, (2020) A Deep Learning based Accident Detection System, International Conference on Communication and Signal Processing (ICCSP), Chennai, India, pp. 1322-1325, doi: 10.1109/ICCSP48568.2020.9182224
13. A. Tsuge, H. Takigawa, H. Osuga, H. Soma, K. Morisaki, (1994) Accident vehicle automatic detection system by image processing technology, Vehicle Navigation and Information Systems Conference, Yokohama, Japan, pp. 45-50, doi: 10.1109/VNIS.1994.396868
14. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, (2016) You Only Look Once: Unified, Real-Time Object Detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788, doi: 10.1109/CVPR.2016.91
15. Deeksha Gour, Amit Kanskar, (2019) Optimized-YOLO: Algorithm for CPU to Detect Road Traffic Accident and Alert System, International Journal of Engineering Research & Technology (IJERT), vol. 8, doi: 10.17577/IJERTV8IS090056
16. D. Tian, C. Zhang, X. Duan, X. Wang, (2019) An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems, IEEE Access, vol. 7, pp. 127453-127463, doi: 10.1109/ACCESS.2019.2939532
17. X. Liu, H. Cai, R. Zhong, W. Sun, J. Chen, (2020) Learning Traffic as Images for Incident Detection Using Convolutional Neural Networks, IEEE Access, vol. 8, pp. 7916-7924, doi: 10.1109/ACCESS.2020.2964644
18. A. Zahid, T. Qasim, N. Bhatti, M. Zia, (2023) A data-driven approach for road accident detection in surveillance videos, Multimedia Tools and Applications, doi: 10.1007/s11042-023-16193-
19. D. K. Yadav, Renu, Ankita, I. Anjum, (2020) Accident Detection Using Deep Learning, 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, pp. 232-235, doi: 10.1109/ICACCCN51052.2020.9362808
20. J. Li, H. -C. Wong, S. -L. Lo, Y. Xin, (2018) Multiple Object Detection by a Deformable Part-Based Model and an R-CNN, IEEE Signal Processing Letters, vol. 25, no. 2, pp. 288-292, doi: 10.1109/LSP.2017.2789325
21. Da Yang, Yuezhu Wu, Feng Sun, Jing Chen, Donghai Zhai, Chuanyun Fu, (2021) Freeway accident detection and classification based on the multi-vehicle trajectory data and deep learning model, Transportation Research Part C: Emerging Technologies, vol. 130, doi: 10.1016/j.trc.2021.103303
22. B. Oh, J. Lee, (2018) A case study on scene recognition using an ensemble convolution neural network, 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, Korea (South), pp. 1-1, doi: 10.23919/ICACT.2018.8323751
23. B. Sumathy, L. Sundari, S. J. Priyadharshini, G. Jayavarshini, (2021) Vehicle Accident Emergency Alert System, IOP Conf. Series: Materials Science and Engineering, Publishing doi:10.1088/1757-899X/1012/1/012042.
24. R. Raffik, M. M. Jones, T. Murugajothi, B. Kannadasan, (2021) Intelligent Accident Detection and Smart Alert System for Vehicles, International Journal of Mechanical Engineering, Vol. 6, No.3, pp.2391-2396
25. A. G. Francis, C. Gottursamy, S. R. Kumar, M. Vignesh, T.P. Kavin, (2020) Accident Detection and Alerting System Using GPS & GSM, International Journal of Advanced Science and Technology Vol. 29, No. 3, pp. 3598 – 3601
26. Charan Kumar C, (2020) Accident Detection From CCTV Footage, Kaggle, https://www.kaggle.com/datasets/ckay16/accident-detection-from-cctv-footage