# Video-based road accident detection on highways: A less complex YOLOv5 approach

K. A. D. Devindu Dharmadasa
*Dept. of ECE, NIT Rourkela*
*Odisha, India*
119EC0186@nitrkl.ac.in

Goutam Kumar Sahoo
*Dept. of ECE, NIT Rourkela*
*Odisha, India*
goutamkr_sahoo@nitrkl.ac.in

Santos Kumar Das
*Dept. of ECE, NIT Rourkela*
*Odisha, India*
dassk@nitrkl.ac.in

Poonam Singh
*Dept. of ECE, NIT Rourkela*
*Odisha, India*
psingh@nitrkl.ac.in

***Abstract*—Advancement of Artificial Intelligence (AI) technologies and the availability of high-end computing devices create scope for the implementation of intelligent transport infrastructure for road safety. This paper proposes an intelligent model for accident detection on highways using more robust, less complex, and more accurate YOLOv5 and StrongSort to locate and track vehicles. It provides a reliable approach for detecting accidents based on vehicle speed, acceleration, trajectory anomalies, and area anomalies. The methodology follows three major steps. The first stage performs vehicle detection, the next stage performs vehicle tracking and feature extraction, and the last stage does crash detection. In this study, only vehicle detection and tracking are addressed using a deep learning-based methodology. However, the accident prediction is done by an algorithm using threshold levels of various parameters such as the speed of vehicles, acceleration anomalies, etc. The model shows good performance when evaluated using the test crash videos under different ambient conditions such as daylight and night.**

***Index Terms*—Accident Detection, Vehicle detection, Object tracking, YOLO v5, StrongSORT.**

## I. Introduction

As the road lengths being increased day by day, the number of motor vehicles has also increased. Since 2000, road lengths have increased by $39\%$ and the number of motor vehicles has increased by $158\%$ [1]. With these advancements taking place road accidents have been increasing tremendously. According to the "Road accident in India" report by the Ministry of road transport and highways transport research wing, New Delhi despite the Covid 19 induced lockdowns and the implementation of the new motor vehicle act in India, traffic accidents resulted in more than $3.4$ lakh injuries and nearly $1.3$ lakh fatalities in the year 2020 [2]. A total of $47,984$ accident-related deaths occurred on national highways in 2020, accounting for around $36.4\%$ of all accident-related deaths in India [2]. Given that India is a party to the Brasilia Declaration, which is a document endorsed by nations with plans to meet sustainable development goal 3.6 is to reduce by half the number of people killed and injured in traffic accidents worldwide by 2020 [1]. From the death statistics on National Highways, it is now one of the prime goals to reduce the number of fatalities through intelligent road infrastructure on highways. The highway scenarios are most challenging to carry out intelligent infrastructure setup where a setup of cameras will be employed for continuous monitoring of the road activity. It will be most important to detect road accidents automatically and alert the highway authorities in time to save the precious lives of many victims [3]–[5]. The causes of road accidents on highways are mostly due to speeding, sometimes dangerous/careless driving or overtaking, and also due to poor weather conditions.

Smart cities are now equipped with smart cameras for monitoring activities. These facilities can be utilized to develop intelligent systems that will take input data from the roadside camera units and optimized algorithms will process efficiently to identify driving anomalies like road accidents. Mostly computer-vision-based techniques use deep learning models for object detection and classification work [6]. In the literature, computer vision-based crash detection work mostly uses deep learning techniques to detect and track objects, while different algorithms are used to determine the occurrence of a crash. For precise object detection in Work by Ijjana *et al.* [7], a mask RCNN is used, and then a productive centroid-based technique is used for surveillance footage. The tracked vehicles are then checked for parameters such as acceleration anomaly, trajectory anomaly and angle anomaly. Despite being tested under conditions such as rain,daylight,snow and hail, framework has shown less effectiveness for high density traffic. Using a geographical feature extractor, a temporal feature extractor, and a binary classifier, Robles-Serrano *et al.* [8] suggested a deep learning technique to automatically predict traffic accidents from movies. Inception V4 architecture was used for spatial feature extraction and temporal feature extraction was performed by convLSTM network [9]. However, a Dense ANN block was used for binary classification. For video traffic accident detection, the proposed model performs well, but is limited to vehicular collisions except for motorcycles, bicycles and pedestrians. The model also displayed problems when identifying crash segments with poor illumination, poor resolution, or occlusion.

Work by Yong-Kal Ki *et al.* [10] proposed a vision-based traffic accident detection algorithm that can automatically detect, record, and report accidents at intersections. The model first extracts vehicles from the video image of a charge-coupled device camera, tracks moving vehicles, and extracts features such as velocity variation rate, position area, and direction of moving vehicles. Similarly, to efficiently detect and locate traffic accidents, a region-based strategy for simulating interactions between many moving objects has been developed by Kimin Yun *et al.* [11]. The natural phenomena of water surface movement caused by several moving objects on water, which is modeled into a Motion Interaction Field (MIF) using Gaussian kernels, served as the basis for the proposed method. Some of the significant challenges for this work implementation are as follows.

- Continuous monitoring of roads required uninterrupted power supply to the system.
- Addressing current road problems to upgrade the infrastructure and perform preventive maintenance.
- locating "grey spots", or constantly monitoring dynamic dangers throughout the whole road network by data analysis and mobility analysis, which, left unaddressed, can become blackspots.
- In the field of mobility, data-driven technology solutions includes machine learning, computer vision, and computational sensing techniques.
- One of the most significant difficulties is the India Driving Dataset (IDD), a dataset for comprehending street images in unstructured situations gathered from Indian highways. IDD is characterized by departing from common conceptions of well-defined infrastructure, such as lanes, few traffic participants, little fluctuation in the presence of an item or background, and strict enforcement of traffic laws.

The main contributions of this work are:

- Use of robust YOLO v5 algorithm based on regression for vehicle detection from CCTV footage.
- Using StrongSort, an improved version of the classic separate tracker 'DeepSort' for vehicle tracking purposes.
- Development of an algorithm to predict road accidents using various parameters such as vehicle speed, and acceleration anomalies.
- Development of an algorithm for automatic detection of accidents without human intervention.
- Proposal of a system design to alert authorities such as medical and police assistance at accident sites.

The remaining parts of this paper are organized as follows. Section II provides the motivations and research objectives, Section III presents the proposed methodology, Section IV presents the experimental results and analysis, and Section V gives the concluding remarks with future works.

## II. Motivations and Research Objectives

Highway fatality statistics and literature survey on computer vision technology lead us to work on Intelligent Transportation System (ITS) for safety and immediate health care. Also, the availability of high-end-computing equipment creates scope for vision-based algorithmic implementation in real-time. The purpose of accident detection is to build a system that can automatically identify accidents without human involvement. CCTV-based automatic accident detection systems have been widely researched as it eliminates the human observation factor required for monitoring CCTV $24/7$. Also, it can speed up the process of getting medical and police help to the accident site. The main objectives of this work are (i) design of an accident detection model using deep learning techniques that are robust and can help in the accurate detection of accidents using CCTV footage. (ii) designing an alert system to alert authorities when an accident is detected.

## III. Proposed Methodology

The methodology adopted for the accurate detection of accidents using CCTV footage and alerting the authorities follows three key steps, i.e., (i) Vehicle detection, (ii) Vehicle tracking and feature extraction, and (iii) Accident detection.

### A. Vehicle Detection

In the proposed method, YOLO v5 is used for vehicle detection. YOLO v5 is an object detection algorithm based on regression. YOLO transfers the object detection problem into a regression problem solution by completing the prediction of the classification and location information of the objects in the input images according to the calculation of the loss function [12]. The network architecture of YOLO v5 is shown in Fig. 1. Three components make up the YOLO v5 structure: the model neck, the model head, and the model backbone. Cross stage partial networks (CSP), which extract significant features from a particular input image, make up the model's backbone. Pyramids of features are created by the model neck. Models that use feature pyramids scale objects well in general. It is useful to distinguish between the same thing in various sizes and scales. YOLO v5 uses PAnet as the feature pyramid. Final detection part is done by the model neck. It consumes features from neck and applies anchor boxes on features and generates final output vectors with class probabilities, objecting scores and bounding boxes. The benefit of YOLO v5 is that it is simple to construct and can train the whole set of photos right away because it doesn't use a different network to extract candidate regions. Also, it consumes less processing time than Fast RCNN [13].

### B. Vehicle tracking and feature extraction

Once all vehicles in a given frame are correctly detected by YOLO v5 model, the vehicles are assigned a unique ID and tracked until the vehicle is out of the field of view of the CCTV camera. For tracking purposes, the StrongSort model is used, which is an improved version of the classic separate tracker DeepSort and is the successor to Simple Online and Realtime Tracking (SORT) [14]. The SORT framework is simple and handles object tracking in real-time in image space using Kalman filtering and the Hungarian method for frame-by-frame data associated utilising a bounding box overlap association metric. Work by Bewley *et al.* in [14] suggested DeepSORT which includes a deep learning component that incorporates the visual features of an object. StrongSort in the work proposed by Yunhao Du *et al.* [15] includes improvements over DeepSort as it has a stronger feature extractor that transforms the original simple CNN into DeepSORT, and uses the NSA Kalman algorithm which is adapted to low-quality detection. Accident detection needs to extract several features such as speed, acceleration, position, area (size) and direction of the vehicles.

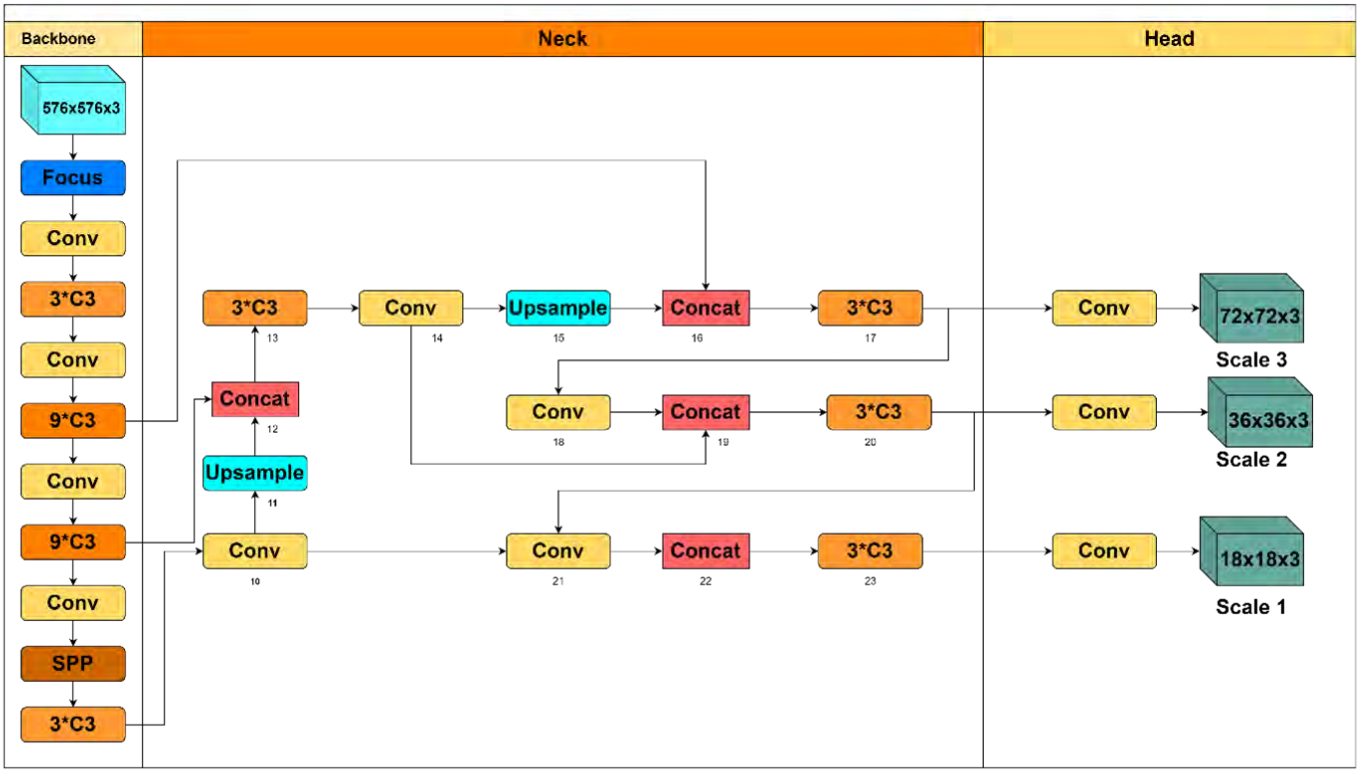


Fig. 1: Network Architecture of YOLO v5

### *C. Accident Detection*

YOLO v5 captures CCTV video frame by frame. First YOLO takes a frame and detects all the vehicles that appear in the frame. In the context of highways, the vehicle classes of interest are car, bus, and truck. However, YOLO v5 can detect a total of six vehicle classes. Then the frame with the detected vehicles is processed by DeepSORT. If five vehicles are detected in a frame, then DeepSORT will loop through the frames five times and create bounding boxes for each vehicle and assign vehicle IDs to them. The centroid of the bounding box $(x, y)$ is calculated from the coordinates given by YOLO using equation 1.

$$(x, y) = (\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}) \tag{1}$$

All vehicles detected in a particular frame are stored in a dictionary which will be addressed as a dictionary '$A$' in the upcoming explanations. This dictionary contains vehicle IDs and their bounding box centroids. This dictionary is also used to check whether the bounding boxes of the two vehicles are overlapping. Another dictionary will be maintained with all the vehicles detected so far from all the frames. This dictionary will be called dictionary '$B$' in the following explanations. Dictionary '$B$' is used to calculate the speed and acceleration of the vehicle. Another dictionary is maintained with all the vehicles with their IDs and the fields of their bounding boxes, which will be referred to as dictionary '$C$'.

The proposed approach follows a criterion that includes some key points to predict the accident occurrence are: (i) Overlapping of bounding boxes of vehicles, (ii) Determining the area anomaly of bounding boxes, (iii) Determining speed and their change in acceleration, (iv) Determining trajectory and their angle of intersection. Area, acceleration and angle anomalies are determined by the above criteria as shown in Fig. 2. These anomalies are used to decide whether an accident has occurred or not.

*1) Checking if bounding boxes are overlapping:* Overlapping of bounding boxes is checked by comparing centroids stored in a dictionary '$A$'. Here it is checked whether the centroids of the two bounding boxes are the same or close enough to the given range for both axes. It is observed that when two bounding boxes collide, the YOLO model recognizes both vehicles as one vehicle (one bounding box) during the collision. Again a couple of frames after the collision, the vehicles will be separately identified by the YOLO model. So, if the bounding boxes are overlapping, then it is checked whether the same vehicle is available in the next frame.

*2) Determining the area anomaly of bounding boxes:* As two vehicles are overlapping at the exact moment of the collision YOLO model identifies the collided vehicles as one vehicle, and this large bounding box is assigned the same ID as the vehicle. Therefore one of the bounding boxes will experience its largest area after a collision with another vehicle. This scenario is shown in Fig. 3 and 4. After the

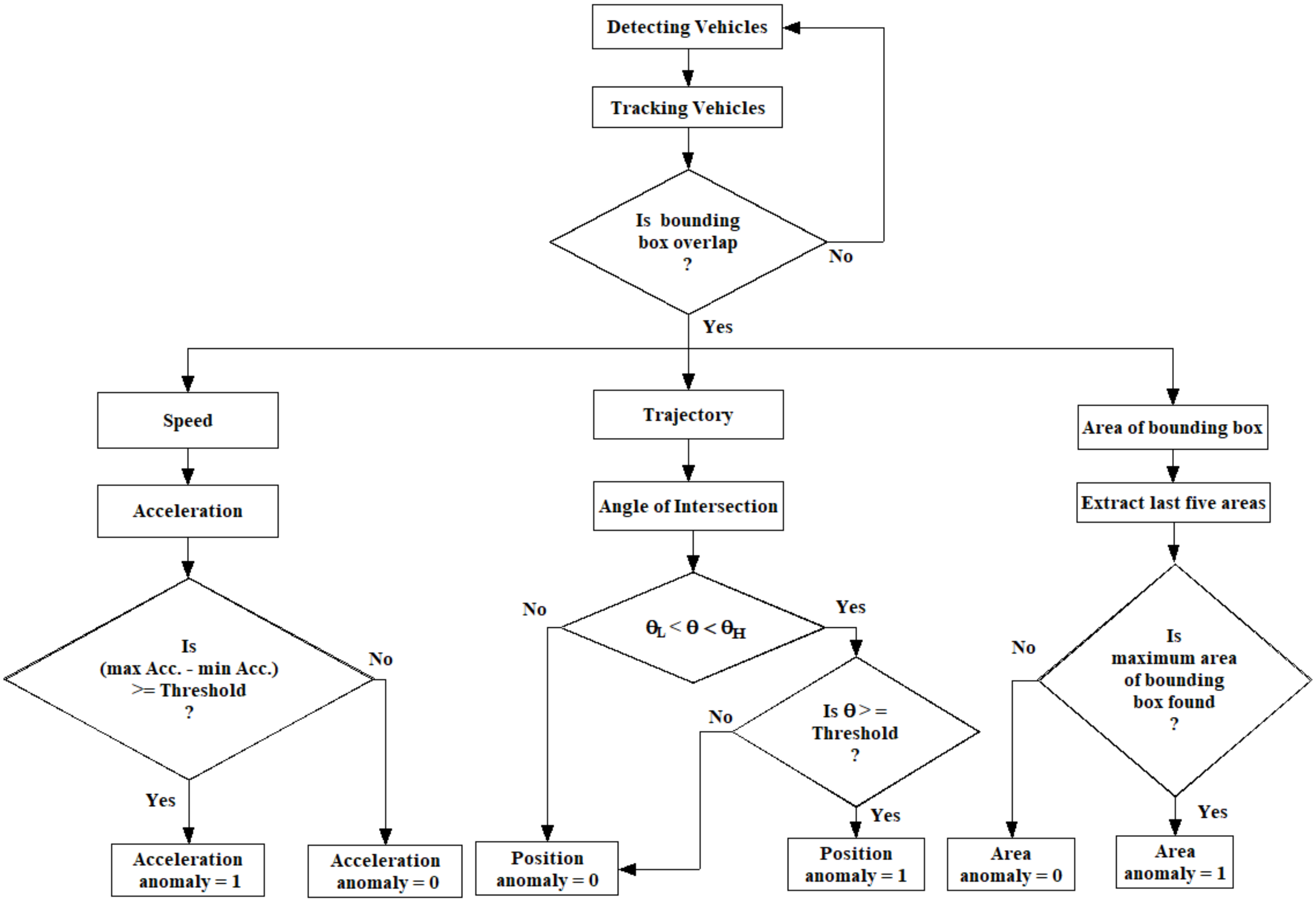


Fig. 2: Workflow diagram of Accident detection algorithm

overlapping bounding boxes are identified, the regions of the bounding boxes involved in the collision are extracted from the dictionary '$C$'. Then it is checked whether the recorded maximum area is available in the last five area values of those particular bounding boxes. The area anomaly is set as '1' if the maximum area is found among the last five area values and is set to '0' otherwise.


Fig. 3: Before collision.

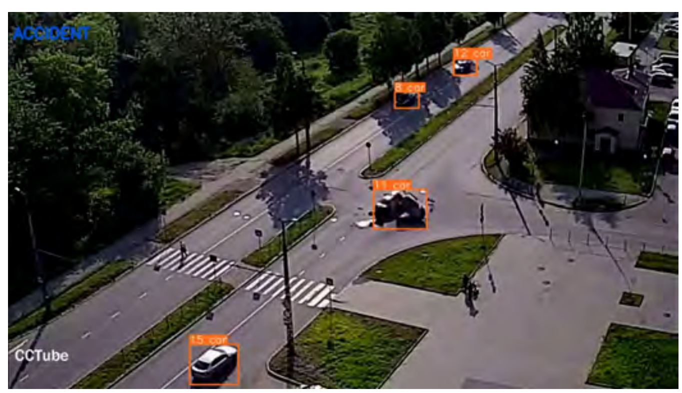


Fig. 4: After collision.

*3) Determining speed and change in acceleration:* After the overlapping of bounding boxes is detected, the algorithm checks for an acceleration anomaly. To calculate acceleration, velocity must be calculated. To calculate the speed, the recorded centroids of the overlapped bounding boxes are extracted from the dictionary '$B$'. From these centroids, the Euclidean distance is calculated for each successive five frames, and the speed is calculated using the equations 2 and 4. The interval between video frames, denoted as $\tau$, is represented in the equation 3. This is found by taking the reciprocal of the Frames Per Second (FPS) value of the video. $\tau$ is used in the calculations of equations 4 and 6. The speed should be scaled using the frame height and bounding box height as shown in equation 5. Then from equation 6, the acceleration can be found from the difference of speed in the time period. A decision is then made by comparing the difference between minimum and maximum acceleration with a given threshold acceleration anomaly.

*4) Determining angle of intersection:* Trajectories of the vehicles are derived by computing the difference between the centroids of a tracked vehicle which are obtained from the dictionary '$B$' for every five successive frames. The magnitude of this 2D vector is then found. Then scalar division of the obtained vector is performed by its magnitude for normalization of the vector. Then the vector's magnitude is compared with a threshold and if it exceeds the threshold, vector is then stored in a dictionary which contains normalized direction vectors for each tracked object. If not it's discarded. Then this trajectory is displayed by extrapolating the vector. The angle of intersection for two vehicles is determined by equation 7. If $\theta$ is greater than a given threshold angle anomaly is set to '1' or else '0'.

$$Euclidean\_distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

$$\tau = \frac{1}{FPS} \tag{3}$$

$$Speed = \frac{Euclidean\_distance}{(\tau) \times interval} \tag{4}$$

$$Speed_s = (\frac{H - h}{H} + 1) \times Speed \tag{5}$$

$$A = \frac{Speed_s^2 - Speed_s^1}{\tau \times interval} \tag{6}$$

$$\theta = \frac{\mu_1 \times \mu_2}{|\mu_1||\mu_2|} \tag{7}$$

By the combination of all computed anomalies, each individual threshold is assigned a weight based on their values, which results in a score between 2 and 3. A score higher than 2 is considered an accident.

## IV. Results and Analysis

All the experiments were conducted on a computer with Intel(R) Core(TM) i5-12500H CPU @ 3.10 GHz, and 16GB main memory (RAM). All programs were written in Python, and video processing was done using OpenCV 4.0.

### *A. Dataset used*

YOLO v5 is pre-trained using the MS COCO dataset [16], which can detect objects belonging to 80 classes, three of which belong to the proposed task, car, truck, and bus. StrongSORT is trained using MOT 17 dataset [17], which is a popular dataset for MOT (Multiple Object tracking), which consists of 7 sequences, $5,316$ frames for training and 7 sequences, 5919 frames for testing. Vehicle crash video clips are taken from the CADP dataset that are recorded from CCTV cameras and used to test crash detection algorithms [18].

### *B. Experimental Results and Analysis*

The proposed method correctly detects and tracks all vehicles using YOLO v5 and StrongSort techniques, respectively. The results of vehicle detection and tracking are shown in Fig. 5 and Fig. 6. From the results, it can be observed that vehicle detection and tracking are based on the bounding box concept. Also, on the top left of the video frame "ACCIDENT" is displayed whenever the algorithm detects an accident. The testing video pattern is considered as a $10s$ moving picture used for accident detection.

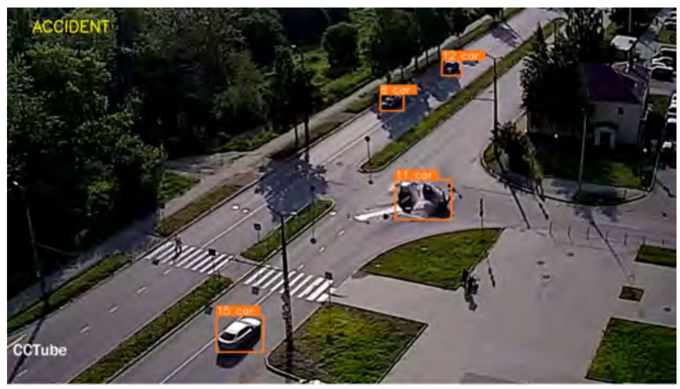


Fig. 5: Accident Detected.


Fig. 6: Accident Detected.

Table I represents the calculated performance parameters for the two videos as shown in Fig. 5 and Fig. 6. Row 1 and row 2 show average speed and acceleration. From the last five accelerations recorded the difference between the maximum and

TABLE I: Performance Results of the proposed model.

| Parameter | Video 1 (Fig. 5) | Video 2 (Fig. 6) |
|---|---|---|
| Average Speed (pixels/s) | 4.72 | 2.19 |
| Average Acceleration (pixels/s$^2$) | 5.72 | 2.15 |
| Minimum Acceleration (pixels/s$^2$) | 0 | 0.29 |
| Maximum Acceleration (pixels/s$^2$) | 93.8 | 8.92 |
| Average Areas (pixels$^2$) | 1237.8 | 3553.0 |
| Maximum Area (pixels$^2$) | 1802 | 4158 |
| Intersection Angle (rad) | 1.89 | 1.97 |

minimum acceleration as shown in rows 3 and 4 are taken and if its higher than a certain threshold an acceleration anomaly is recorded. Bounding box areas of vehicles are recorded in a dictionary along with their vehicle ids until a vehicle leaves the frame. Average of recorded bounding box areas of a detected vehicle for five frames before the collision is shown in row 5 of Table 2. When a collision happens maximum area, which is shown in row 6 is obtained from the aforementioned dictionary and is checked if its available in the last five area values. If its available area anomaly is set as 1 and else 0. Row 7 shows the intersection angle which is calculated from equation 7 and if the angle is higher than a certain threshold position anomaly is set to 1 else 0. From the result analysis, most of the time it was possible to detect the accident using only acceleration anomaly and position anomaly. Therefore, acceleration anomaly and position anomaly are given more importance in deciding whether an accident has occurred or not. The field anomaly is also introduced for more robustness of the algorithm.

With the change of light and ambiance of the atmosphere, YOLO tends to identify the same vehicle with two different IDs in concurrent frames. This leads the algorithm to identify it as a collision which ends up as a false alarm in most cases. As aforementioned after the collision both of the vehicles are identified as one unit (if the vehicles stay close after the collision); this makes it impossible to calculate the speed and acceleration anomalies of the vehicles after the collision.

## V. Conclusions

This study proposes a comprehensive method for detecting vehicle accidents on highways. The main focus of the proposed study is to use a deep learning-based approach for vehicle detection and tracking, with the intention of improving accident detection. This work uses several parameters such as vehicle speed, acceleration, and intersection angle as well as supporting parameters such as the area of the bounding box. YOLOv5 and StrongSort models are used for efficient vehicle identification and tracking. The advantages of using these state-of-the-art techniques are improved robustness and reduced complexity as well as increased accuracy. In the future, with multi-camera integration, the proposed model can provide a comprehensive view of highway accidents, and it can alert authorities using notification services in real-time. The latest version YOLOv7 can also be used for more accurate real-time object detection models in various computer vision applications [19].

## References


[1] P. L. research, "Overview of road accidents in india," in *Overview of Road Accidents in India, PRS Legislative research*. PRS Legislative research, 2017, pp. 1–3.

[2] M. o. R. T. Transport Research Wing and Highways, "Road accidents in india," in *Transport Research Wing, Ministry of Road Transport and Highways*, 2020, pp. 1–267.

[3] G. K. Sahoo, S. A. Patro, P. K. Pradhan, S. K. Das, and P. Singh, "An IoT-based intimation and path tracing of a vehicle involved in road traffic crashes," in *IEEE-HYDCON*, 2020, pp. 1–5.

[4] K. Parasana, G. K. Sahoo, S. K. Das, and P. Singh, "A health perspective smartphone application for the safety of road accident victims," in *Proc. 2nd IEEE Advanced Commun. Technol. Signal Process.(ACTS)*, 2021, pp. 1–6.

[5] G. K. Sahoo, H. Srivastava, U. N. C. Unagalle, S. A. P. Mihiran, K. D. Jayarukshi, S. K. Das, and P. Singh, "Accident rescuing system for vehicles in road traffic: A smart phone application," in *Proc. 4th IEEE International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, 2022, pp. 419–424.

[6] G. K. Sahoo, S. K. Das, and P. Singh, "A deep learning-based distracted driving detection solution implemented on embedded system," *Multimedia Tools and Applications*, pp. 1–24, 2022.

[7] E. P. Ijjina, D. Chand, S. Gupta, and K. Goutham, "Computer vision-based accident detection in traffic surveillance," in *Proc. IEEE 10th Int. Conf. Computing Commun. Netw. Technol. (ICCCNT)*, 2019, pp. 1–6.

[8] S. Robles-Serrano, G. Sanchez-Torres, and J. Branch-Bedoya, "Automatic detection of traffic accidents from video using deep learning techniques," *Computers*, vol. 10, no. 11, p. 148, 2021.

[9] R. Nayak, M. M. Behera, V. Girish, U. C. Pati, and S. K. Das, "Deep learning based loitering detection system using multi-camera video surveillance network," in *Proc. IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, 2019, pp. 215–220.

[10] Y.-K. Ki and D.-Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 188–194, 2007.

[11] K. Yun, H. Jeong, K. M. Yi, S. W. Kim, and J. Y. Choi, "Motion interaction field for accident detection in traffic surveillance video," in *Procc. 22nd IEEE Int. Conf. Pattern Recogn.*, 2014, pp. 3062–3067.

[12] R. Nayak, M. M. Behera, U. C. Pati, and S. K. Das, "Video-based real-time intrusion detection system using deep-learning for smart city applications," in *Proc. IEEE Int. Conf. Advanced Netw. Telecommunications Syst. (ANTS)*, 2019, pp. 1–6.

[13] H.-K. Jung and G.-S. Choi, "Improved yolov5: Efficient object detection using drone images under various conditions," *Applied Sciences*, vol. 12, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/14/7255

[14] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.

[15] Y. Du, Y. Song, B. Yang, and Y. Zhao, "Strongsort: Make deepsort great again," *arXiv preprint arXiv:2202.13514*, 2022.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[17] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 104–119, 2021.

[18] A. P. Shah, J.-B. Lamare, T. Nguyen-Anh, and A. Hauptmann, "Cadp: A novel dataset for cctv traffic camera based accident analysis," in *Proc. 15th IEEE Int. Conf. Advanced Video Signal Based Surveillance (AVSS)*, 2018, pp. 1–9.

[19] S. K. Mistry, S. Chatterjee, A. K. Verma, V. Jakhetiya, B. N. Subudhi, and S. Jaiswal, "Drone-vs-bird: Drone detection using yolov7 with csrt tracker," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–2.