



AI-Based Road Accident Detection and Alert System

Mr. S. A. Zambare¹, Saniya Mainuddin Mulla², Kiran Sanjay Jadhav³,
Shivprasad Suresh Kadam⁴, Reshma Mujofar Pathan⁵

Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Korti, Pandharpur, India¹⁻⁵

Abstract: Road accidents cause heavy loss of life, and the outcome for victims depends strongly on how quickly emergency services are informed. Manual CCTV monitoring is slow and error-prone. This paper presents the design, architecture and implementation of an AI-based Road Accident Detection and Alert System that analyses CCTV or user-uploaded video frame by frame using Open CV and a YOLO deep-learning detector, and forwards confirmed accidents to emergency services. The system is built with Python, Flask, HTML, CSS and JavaScript and offers a public accident-reporting page with automatic GPS location capture, an automatic video-processing pipeline, an accident review dashboard with statistics, and a separate police/ambulance operations console with live alerts and filters. In the test session the dashboard logged 42 detected events, of which 16 were confirmed on review (38.1 %). The work shows that automated detection with human confirmation and location-tagged alerts can shorten the path from an accident to emergency response, and it identifies precision improvement as the main area for further work.

Keywords: Road Accident Detection, Computer Vision, Deep Learning, YOLO, Open CV, CCTV Surveillance, Emergency Alert System, Django, Intelligent Transportation System.

I. INTRODUCTION

Road traffic injuries are a leading cause of death worldwide, and low- and middle-income countries, including India, carry a disproportionate share of the burden [1], [2]. As the number of vehicles in cities grows, so do collisions, injuries, property damage and traffic congestion. The severity of an accident is strongly influenced by how quickly emergency services learn about it: every minute between the crash and the arrival of medical help reduces the chance of saving victims.

Traditional traffic monitoring relies on control-room operators watching CCTV screens or on calls from bystanders. Watching many feeds for long periods causes fatigue, distraction and human error, so an accident may be missed or noticed late, and manual reporting adds further delay. Drivers are often unable to report the exact location, which delays the response even more.

Advances in computer vision and deep learning [3] now make it possible to analyze video automatically. Real-time detectors such as YOLO [4] locate vehicles in every frame, and the positions, motion and interactions of these vehicles can be analyzed to identify collisions, sudden stops and other abnormal behaviour [5], [6], [7]. However, many published systems stop at detection and offer limited automated alerting, location tagging, review workflow and integration with emergency responders.

AccidentAlert, the system proposed in this paper, is an AI-based road accident detection and alert platform that addresses this gap. It processes CCTV or user-uploaded video, detects accidents without any extra sensors or hardware, records the incident with time, location, vehicle count and confidence, and passes confirmed accidents with GPS coordinates to a dedicated police and ambulance console. A public reporting page lets any bystander upload an accident video from a phone, with the location captured automatically. The system thus supports faster, more informed emergency response and contributes to Intelligent Transportation Systems and smart-city safety.

II. LITERATURE REVIEW

A review of previous research on vision-based accident detection and alert systems was carried out. The proposed work is directly related to the following studies.

Ijjina et al. [6] proposed a computer-vision framework that detects vehicles with a deep object detector, tracks them, and analyses speed and trajectory anomalies to recognize accidents in surveillance video. The approach works on ordinary



traffic footage, but the paper concentrates on detection and does not address alerting or emergency-service integration. pawar and Attar [7] applied deep learning to detect and localize road accidents in traffic surveillance videos, and Thakare et al. [8] localized and described accident events by analyzing interactions between objects. Both improve detection but stop short of a complete response workflow.

Zhang and Sung [9] used trajectory tracking with influence maps to identify accidents, while Ghahremannezhad et al. [10] reported real-time accident detection in traffic surveillance based on deep learning. Basheer Ahmed et al. [5] presented a real-time approach that detects and classifies traffic incidents, and Khan et al. [11] studied anomaly detection in traffic surveillance videos. These studies show that vehicle tracking and anomaly analysis are effective cues, but they report detection results rather than a deployable alerting system.

Tahir et al. [12] built an event-driven road traffic monitoring system based on CCTV video analytics, and Hooda et al. [13] described an accident detection system using surveillance cameras. Joshi and Rao [14] designed a hybrid CNN for real-time traffic monitoring, and Gupta et al. [15] used transfer learning and CNN-based vehicle identification for hit-and-run cases. Choi et al. [16] combined ensemble deep learning with multimodal data from dashboard cameras for crash detection.

Other work relies on sensors and vehicle networks. Prabakar et al. [17] developed a prototype that detects an accident and indicates victim status, Lee [18] proposed a highway accident detection scheme using CCTV, and Zeng et al. [19] studied opportunistic fleets for road event detection in vehicular sensor networks. Such systems require extra hardware in vehicles or along the road, which limits large-scale deployment.

A. Object Detection and Tracking Methods

Two-stage detectors such as Faster R-CNN [20] and Mask R-CNN [21] give high accuracy but are comparatively slow. Single-stage detectors, namely SSD [22] and the YOLO family [4], [23], [24], [25], trade a small amount of accuracy for much higher speed, which suits real-time CCTV monitoring. Detections can be linked across frames with trackers such as SORT [26] and Deep SORT [27], and temporal behaviour can be modelled with LSTM networks [28]. Vision transformers [29] are a recent alternative, but they demand more computation. Detectors are commonly pre-trained on large datasets such as COCO [30] and then fine-tuned on accident footage such as the CCTV accident dataset on Kaggle [31].

B. Research Gap Identification

Three critical research gaps were identified in the body of work above that the proposed system aims to address:

- **Detection without response:** Most studies report accident detection only. Automated alert generation, location tagging and hand-over to police and ambulance services are missing or limited.
- **Dependence on extra hardware:** Sensor and vehicle-network solutions need additional devices in vehicles or on roads, whereas existing CCTV infrastructure is already widely available.
- **No public-reporting and review workflow:** Existing systems rarely combine automatic detection with bystander video reporting, human confirmation, and a responder console with filters and statistics.

III. PROBLEM STATEMENT

Road accidents are increasing rapidly, resulting in loss of life, serious injuries and traffic congestion. Existing monitoring depends on manual CCTV surveillance or public reporting, which causes delays in accident detection and in informing emergency services. Operators cannot watch every camera continuously, callers often cannot give an accurate location, and there is no single platform that connects detection, verification and emergency response.

The following difficulties are therefore faced: (i) accidents are detected late or missed because continuous human monitoring is not possible; (ii) emergency services receive incomplete information, without precise location or video evidence; (iii) detection, reporting and response are handled in separate, disconnected tools; and (iv) authorities lack simple statistics on accident frequency and locations for planning preventive measures.

To combat this, the proposed system automatically detects road accidents from CCTV or uploaded video using AI, records the incident with its confidence and location, allows operators to confirm it, and immediately presents confirmed accidents to emergency services on a dedicated console. The objectives are: (1) to develop an AI-based accident detection system using CCTV footage without additional sensors; (2) to minimize emergency response time through instant alerts; (3) to detect vehicle fire and smoke after an accident and raise high-priority alerts; and (4) to provide an analytics dashboard showing accident trends and high-risk locations.



IV. PROPOSED SYSTEM AND METHODOLOGY

The proposed system is an AI-driven accident detection and alert platform for road safety. With Open CV and YOLO for video analysis, Python Flask as the backend and a browser-based dashboard as the frontend, the overall architecture is shown in Fig. 1.

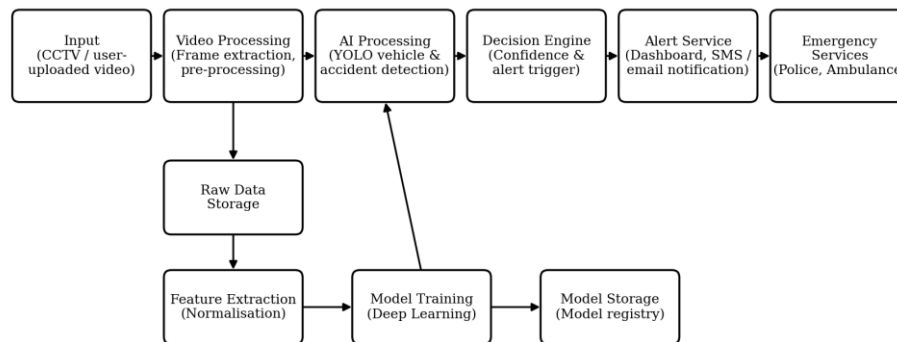


Fig. 1 – System Architecture

A. System Architecture Overview

The end-to-end data flow is as follows. CCTV footage or a user-uploaded video is the input. The video-processing stage splits it into frames and pre-processes them. The AI-processing module applies the trained YOLO model to each frame to detect vehicles and accident indicators. The decision engine compares the result with a confidence threshold and decides whether to trigger an alert. The alert service then notifies the dashboard and emergency services (police and ambulance). Raw data, processed features and trained models are kept in storage, and the model is periodically retrained as shown in the lower part of Fig. 1.

B. Video Processing Module

Video is read with Open CV [32], which supports video files, webcams and CCTV streams. The module extracts frames, resizes them to the detector input size, and applies cleaning and normalization. Frames and metadata are stored in raw data storage; processed features are placed in a feature store for model training.

C. Accident Detection Module

A YOLO-based deep learning detector locates vehicles in each frame. Changes in vehicle position, overlap between bounding boxes, sudden stoppage and abnormal movement across consecutive frames are treated as accident indicators. For each suspicious frame the module records the frame number, the number of vehicles, the number of frames showing indicators, and a confidence score. The detector is trained on traffic and accident video data as described in Section VI.

D. Decision Engine and Alert Service

The decision engine maps the confidence score and indicator count to an alert decision. Detected accidents are stored with the status “Detected”; after review they can be changed to “Confirmed”. Confirmed accidents are passed to the alert service, which publishes them to the emergency console and, where the reporter supplied a phone number or e-mail address, informs the reporter by SMS or e-mail.

E. Backend Infrastructure

The Python Flask framework handles upload requests, starts the processing of each video session, runs business logic, stores accident records in the database and exposes the data to the dashboards. Every uploaded video is given a session ID and a processing status (Pending, Processing, Completed), so that users can continue using the system while analysis runs in the background.

F. User Interface Design

The frontend uses HTML, CSS, JavaScript and Bootstrap. The design principles are: clear status badges (Detected, Confirmed); summary cards for quick reading; one-click actions for uploading videos and viewing accidents; an automatic location form that needs minimal typing for bystanders; and a dark, high-contrast console for emergency operators.



G. Software and Hardware Requirements

Software: Python, Open CV, YOLO, Flask (or Django), HTML, CSS and JavaScript on Windows or Linux. Hardware: a laptop or desktop computer to run the model and dashboard, and a webcam or mobile camera for live testing; CCTV cameras can be connected for real deployment.

V. IMPLEMENTATION DETAILS

The prototype consists of three cooperating parts: a public reporting page, an analysis and review dashboard, and an emergency operations console. Screenshots of the implemented system are given below.

A. Report Accident Module

A bystander reports an accident by uploading a video from a phone or device (Fig. 2). Supported formats are MP4, AVI, MOV, MKV and FLV, up to 500 MB. The current GPS coordinates are captured automatically and converted into a readable location name; e-mail, phone number and incident description are optional. The reporter is informed that the data is processed securely.

Fig. 2 – Report Accident Module

B. Upload and Processing Module

After upload the system creates a video session with a session ID, file name, upload time and status PENDING, and starts the analysis automatically (Fig. 3). The user can open the dashboard or the accident list while the video is still being analyzed, or upload another video.

Fig. 3 – Upload and Processing Module

C. Accident Details Module

Every detected accident has a details page (Fig. 4) showing the captured frame, accident ID, time stamp, location, video ID, confidence, number of vehicles, status and notes. In the example, frame 48 of video session 216 was flagged with five vehicles and a confidence of 46.58 %, and the incident was then marked as confirmed.

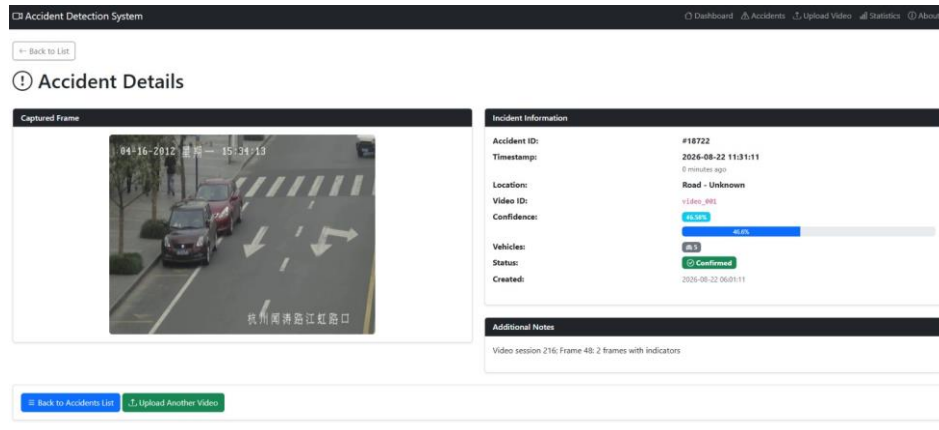


Fig. 4 – Accident Details Module

D. Monitoring Dashboard Module

The dashboard (Fig. 5) shows the total number of accidents, the number of confirmed accidents and the confirmation ratio, together with quick actions and a table of recent detected accidents with time, location, vehicle count and status.

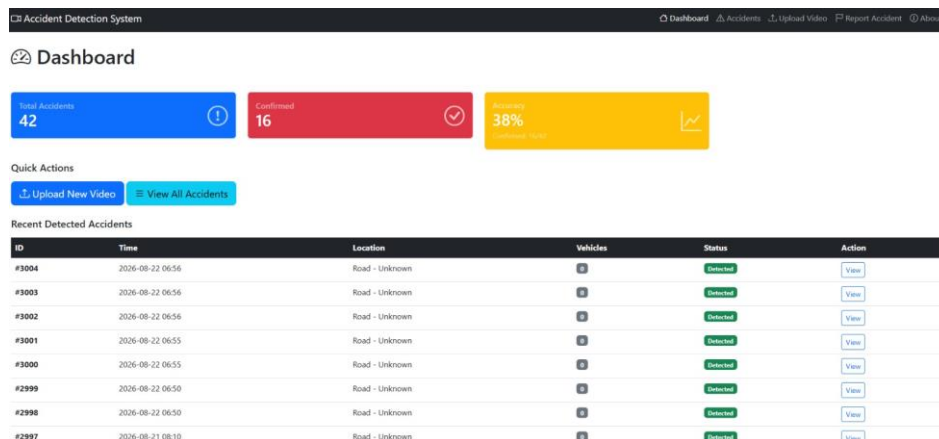


Fig. 5 – Monitoring Dashboard Module

E. Emergency Operations Console

Confirmed accidents are forwarded to a separate console for police and ambulance operators (Fig. 6). It displays active alerts and total reports, filters by city, accident-detected flag and status, and lists each live alert with its video, location, latitude and longitude, reporting status, assigned ambulance and responder contact. The console also provides all-reports and map views and a logout option.

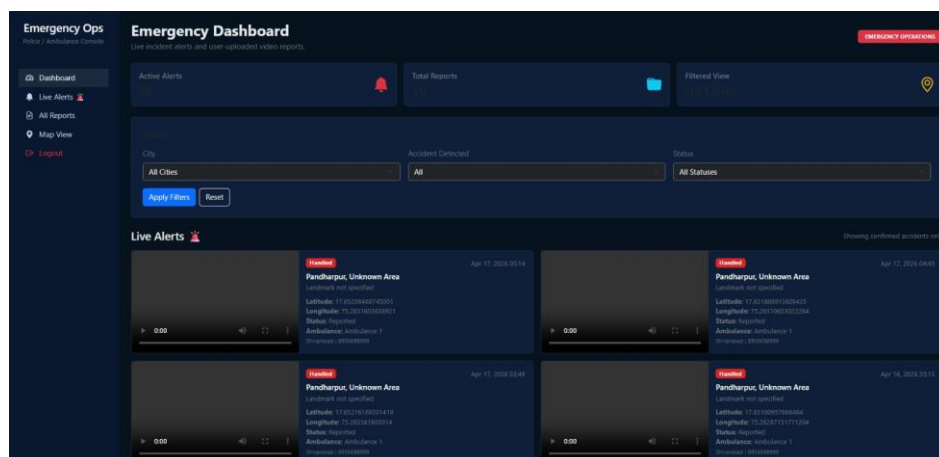


Fig. 6 – Emergency Operations Console



VI. RESULTS AND DISCUSSION

The prototype was tested by uploading traffic video clips through the reporting interface and reviewing the output on the dashboards. The results are discussed below.

• **Detection Performance**

Table II lists the figures recorded by the dashboard. Of 42 events detected, 16 were confirmed on review, giving a confirmation ratio of 38.1 %. This ratio shows how many raised alerts were confirmed; it is not the classification accuracy of the model, which requires a labelled test set and standard metrics. [Insert: training dataset, epochs, image size, precision, recall, F1 score, map, FPS and alert delay from your test videos.]

TABLE II – Detection summary from the monitoring dashboard

Metric	Value
Total accidents detected	42
Accidents confirmed	16
Confirmation ratio (16/42)	38.1 %
Example: confidence	46.58 %
Example: vehicles	5

• **Uniqueness and Innovation**

Table I compares the proposed system with the typical existing approaches found in the literature review. Its main distinguishing feature is the integration of automatic detection, location-tagged public reporting, human confirmation and a dedicated emergency console in one platform that uses only camera video.

TABLE I – Comparative analysis of the proposed system vs. existing systems

Feature	Typical Existing Systems	Proposed System
Monitoring	Manual CCTV observation	Automatic frame-by-frame analysis
Hardware	Extra sensors or in-vehicle units [17], [19]	CCTV / camera video only
Alerting	Manual calls or GSM alert [13]	Automated alert to emergency console
Location	Caller description	GPS captured automatically
Public Reporting	Not available	Video upload from phone
Review Workflow	Limited	Detected → Confirmed
Responder Console	Not integrated	Live alerts, filters, map view
Statistics	Limited	Dashboard totals and recent list

• **Functional Module Performance**

The end-to-end workflow functioned as designed: a video was uploaded with its location, a session was created and processed automatically, the detection was stored with time stamp, vehicle count and confidence, reviewed, and confirmed accidents were listed on the emergency console with GPS coordinates. The dashboard updated its totals and recent-accident table accordingly.

• **Observations and Limitations**

Several entries in the recent-accident table show a vehicle count of zero and remain in the “Detected” state, and the example confirmed accident has a confidence of only 46.58 %. This indicates that the current threshold admits many low-confidence events and that human confirmation is an important safeguard. Performance in poor lighting, rain and heavy occlusion has not yet been evaluated, and fire and smoke detection is planned for future work.



• System Scalability and Future Scope

The modular architecture allows more cameras, detectors and alert channels to be added. Future work includes formal evaluation on a labelled dataset, tracking with SORT or Deep SORT [26], [27] and temporal modelling with LSTM [28] to reduce false alerts, automatic detection of fire and smoke, accident severity estimation, live CCTV stream integration, and analytics of accident-prone locations.

VII. CONCLUSION

This paper presented an AI-based Road Accident Detection and Alert System that uses Open CV, YOLO-based deep learning, a Flask backend and web dashboards to detect accidents from video and pass confirmed incidents, with GPS location, to emergency services. The system combines a public reporting page, automatic video processing, a review dashboard and an emergency operations console in one platform that needs no additional sensors.

By reducing the dependence on continuous human monitoring and by giving responders timely, location-tagged information, the system helps shorten emergency response time and supports safer roads. Its detection precision can be improved further through better training data, tuned thresholds and temporal analysis, and it forms a practical step towards Intelligent Transportation Systems and smart-city road safety.

ACKNOWLEDGMENT

The authors thank **Mr. S. A. Zambare** (Project Guide), **Mr. R. S. Yevale** (Project Coordinator) and **Dr. S. V. Pingale** (Head of Department), Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Pandharpur, for their guidance and support.

REFERENCES

- [1]. World Health Organization, "Road traffic injuries – Fact sheet," WHO, Geneva, 2024. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>
- [2]. World Health Organization, Global Status Report on Road Safety 2015. Geneva: WHO, 2015.
- [3]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [4]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [5]. M. I. B. Basheer Ahmed, R. Zaghoud, M. S. Ahmed, R. Sendi, S. Alsharif, J. Alabdulkarim, and G. Krishnasamy, "A real-time computer vision-based approach to detection and classification of traffic incidents," *Big Data Cogn. Comput.* vol. 7, no. 1, p. 22, 2023.
- [6]. E. P. Ijjina, D. Chand, S. Gupta, and K. Goutham, "Computer vision-based accident detection in traffic surveillance," in *Proc. 10th Int. Conf. Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1–6.
- [7]. K. Pawar and V. Attar, "Deep learning-based detection and localization of road accidents from traffic surveillance videos," *ICT Express*, vol. 8, no. 3, pp. 379–387, 2022.
- [8]. K. V. Thakare, D. P. Dogra, H. Choi, H. Kim, and I. J. Kim, "Object interaction-based localization and description of road accident events using deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20601–20613, 2022.
- [9]. Y. Zhang and Y. Sung, "Traffic accident detection method using trajectory tracking and influence maps," *Mathematics*, vol. 11, no. 7, p. 1743, 2023.
- [10]. H. Ghahremannezhad, H. Shi, and C. Liu, "Real-time accident detection in traffic surveillance using deep learning," in *Proc. IEEE Int. Conf. Imaging Systems and Techniques (IST)*, 2022, pp. 1–6.
- [11]. S. W. Khan, Q. Hafeez, M. I. Khalid, R. Alroobaea, S. Hussain, J. Iqbal, and S. S. Ullah, "Anomaly detection in traffic surveillance videos using deep learning," *Sensors*, vol. 22, no. 17, p. 6563, 2022.
- [12]. M. Tahir, Y. Qiao, N. Kanwal, B. Lee, and M. N. Asghar, "Real-time event-driven road traffic monitoring system using CCTV video analytics," *IEEE Access*, 2023.
- [13]. R. Hooda, R. Kawediya, K. Sonigara, and S. Andhare, "Accident detection system using surveillance cameras," *Int. Res. J. Eng. Technol.*, vol. 7, no. 4, 2020.
- [14]. R. M. Joshi and D. S. Rao, "AlexDarkNet: Hybrid CNN architecture for real-time traffic monitoring with unprecedented reliability," *Neural Comput. Appl.*, pp. 1–9, 2024.
- [15]. R. Gupta, R. Kakkar, S. Tanwar, and D. Garg, "Transfer learning and CNN-based vehicle identification approach for hit-and-run cases," in *Proc. 3rd Int. Conf. Advancement in Electronics & Communication Engineering (AECE)*, 2023, pp. 1–13.



- [16]. J. G. Choi, C. W. Kong, G. Kim, and S. Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras," *Expert Syst. Appl.*, vol. 183, p. 115400, 2021.
- [17]. S. Prabakar et al., "An enhanced accident detection and victim status indicating system: Prototype," in *Proc. Annual IEEE India Conf. (INDICON)*, 2012.
- [18]. I. J. Lee, "An accident detection system on highways through CCTV with Calogero-Moser system," in *Proc. 18th Asia-Pacific Conf. Communications (APCC)*, 2012, pp. 522–525.
- [19]. Y. Zeng, D. Li, and A. V. Vasilakos, "Opportunistic fleets for road event detection in vehicular sensor networks," *Wireless Netw.*, vol. 22, no. 2, pp. 503–521, 2016.
- [20]. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [21]. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2961–2969.
- [22]. W. Liu et al., "SSD: Single shot multibox detector," in *Proc. European Conf. Computer Vision (ECCV)*, 2016, pp. 21–37.
- [23]. J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv: 1804.02767*, 2018.
- [24]. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv: 2004.10934*, 2020.
- [25]. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF CVPR*, 2023, pp. 7464–7475.
- [26]. A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2016, pp. 3464–3468.
- [27]. N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE ICIP*, 2017, pp. 3645–3649.
- [28]. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [29]. A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," *arXiv: 2010.11929*, 2020.
- [30]. T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
- [31]. Ckay16, "Accident detection from CCTV footage," *Kaggle dataset*, 2024. [Online]. Available: <https://www.kaggle.com/datasets/ckay16/accident-detection-from-cctv-footage>
- [32]. G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.