

VISION-BASED REAL-TIME COLLISION WARNING: A SYNTHETIC DATA-DRIVEN ADAS FOR ENHANCED ROAD SAFETY

Busra Jahan Tanu^{*1}, Md. Shihab Ahmed² and Md. Shoaib Ahmed³

¹ Student, University of Asia Pacific (UAP), Bangladesh, e-mail: 19201048@uap-bd.edu

² Student, University of Asia Pacific (UAP), Bangladesh, e-mail: 19201051@uap-bd.edu

³ Student, University of Asia Pacific (UAP), Bangladesh, e-mail: 23215018@uap-bd.edu

***Corresponding Author**

ABSTRACT

Road traffic accidents result in over a million deaths annually, with low and middle-income countries experiencing a disproportionate number of these fatalities. In this paper, we present a novel Advanced Driver Assistance System (ADAS) that utilises computer vision techniques to enable real-time vehicle detection, tracking, and collision warning. The system integrates YOLOv5 for efficient vehicle detection, DeepSORT for robust multi-object tracking, and a custom collision-warning evaluator based on bounding-box pixel-area variation to predict accidents. Our system achieves an Accident Detection Rate (ADR) of 80.43% and an Average Time to Collision (ATTC) of 2.6 seconds in simulated environments. To address the lack of diverse real-world data, we generated a synthetic dataset consisting of 50,092 annotated images using the CARLA simulator, which captures a wide range of environmental conditions, including varying lighting, weather, and road scenarios. The dataset enables training and testing of our system under controlled, reproducible conditions while also allowing simulation of rare and dangerous scenarios that would be difficult or unsafe to capture in real life. Experimental results demonstrate that our method satisfies the real-time constraints necessary for deployment in resource-constrained settings, achieving 30 frames per second at 1080p resolution. This approach offers a cost-effective solution for ADAS, especially in regions where expensive sensors such as LiDAR and RADAR are not widely available. The system's performance across a range of environmental conditions, including daylight, fog, rain, and nighttime driving, highlights its robustness and applicability for real-world traffic-safety improvements. Our proposed ADAS can significantly improve road safety by offering an affordable, reliable, and scalable system that operates in real time, providing timely collision warnings to drivers in low-resource environments. Future work will focus on expanding the system to include additional sensor data, conducting real-world validation, and further optimizing it for deployment in actual vehicles.

Keywords: *Advanced Driver Assistance Systems; Traffic Safety; Computer Vision; Vehicle Detection and Tracking.*

1. INTRODUCTION

According to the World Health Organisation (WHO), approximately 1.19 million people die each year due to road accidents. More than 90% of traffic-related fatalities take place in countries with middle and low incomes. These accidents cost most countries 3% of their GDP [("Road Traffic Injuries," n.d.). Road safety is a significant problem in low- and middle-income countries. The primary causes of accidents are typically attributed to variables such as novice drivers, reckless behaviours, substance abuse, distracted driving, mobile phone usage, and inadequate law enforcement measures (Rolison et al., 2018). As per the study published by the World Health Organization (WHO), to ensure the safety of all those using the road, it is necessary to establish a system that can accommodate and account for human mistakes ("Road Traffic Injuries," n.d.). Hence, the imperative for Advanced Driver Assistance Systems (ADAS) has become urgent. The ADAS can warn and influence drivers against risky maneuvers, thus potentially saving thousands of lives (Gouribhatla & Pulugurtha, 2022). However, most of the ADAS rely on expensive sensors (e.g., LiDAR, RADAR) or offline processing pipelines that preclude real-time application. These expensive sensor-fused ADAS are not suitable for vehicles in low-income countries. Consequently, there is a critical need for ADAS solutions that are both cost-effective and capable of operating in real time on widely available hardware.

Additionally, a diverse dataset of vehicle-related data is needed to improve ADAS. However, real-world data collection is often hampered by inconsistent environmental conditions, high annotation costs, privacy concerns, and safety risks when collecting data on accident scenarios. Synthetic data generation, enabled by advanced simulators such as CARLA, offers a compelling alternative by providing a controlled, reproducible environment. This approach enables systematic variation of conditions such as lighting, weather, and traffic, and allows for the safe simulation of rare and dangerous scenarios. Consequently, synthetic datasets can significantly enhance the robustness and adaptability of ADAS, making them indispensable for developing cost-effective, reliable safety systems.

In this study, we present an ADAS that leverages standard CMOS cameras combined with state-of-the-art computer vision algorithms to detect vehicles, track their motion, and anticipate collisions. This system, even though it can be integrated into various vehicles under different conditions, is best suited for highway vehicles that operate at night and are prone to distraction during long drives. Our approach is designed to operate at 30 FPS on 1080p video streams, ensuring timely warnings for the driver. The core contributions of this paper include:

- A novel ADAS pipeline that fuses YOLOv5-based detection, DeepSORT tracking, and a custom collision warning evaluator.
- The creation of a synthetic dataset using the CARLA simulator to capture diverse driving scenarios and varied environmental conditions.
- Extensive experimental evaluation in simulated environments, demonstrating competitive performance with an ADR of 80.43% and an ATTC of 2.6 seconds while operating in real time.

1.2 Related Works

Vision-based collision warning systems use cameras and computer vision techniques to identify and monitor potential hazards on the road, thereby improving road safety. In early research, these systems were developed by focusing on individual tasks. Many studies have focused solely on object detection using algorithms such as YOLO (Redmon & Farhadi, 2018), SSD (Liu et al., 2015), or Mobile Net (Howard et al., 2017) to detect vehicles and pedestrians. For example, Elkholy et al. (2021) used CNN-based object detectors and inferred collision risk solely from bounding box width. Their method triggers a warning when the bounding box width falls within a predefined percentage of the image width. However, this approach suffers from a critical limitation: it is highly dependent on the absolute size of the detected vehicle. This dependence can lead to misclassifications, failing to trigger warnings for narrow vehicles (e.g., motorcycles or bicycles) and generating false alarms for larger vehicles that are not as close. (Joshi et al., 2024) attempted to address this issue by setting different thresholds for different vehicle classes. Although this class-specific adjustment is a step in the right direction, the experimental evaluation was limited to a small number of images (typically 2–6 samples), casting doubt on the generalizability of the results across diverse driving scenarios. Similarly, (Madhumitha et al.,

2020) estimated relative distance using the centre coordinates of the bounding boxes. While intuitively appealing, this method can be adversely affected by variations in vehicle dimensions and occlusions, resulting in unreliable distance estimates under complex road conditions. (“Forward Collision Warning for Autonomous Driving,” 2022) Implement a forward collision warning system that relies solely on detecting whether an object appears within a predefined region of the image, bypassing complex tracking algorithms. Traditional methods that rely on instantaneous measurements (e.g., bounding box size at a single frame) do not capture the dynamic nature of the vehicle's approach.

Other research efforts separately addressed tasks such as tracking moving objects or estimating the distance between the ego vehicle and potential obstacles (Borrego-Carazo et al., 2020; Kukkala et al., 2018; Nidamanuri et al., 2022; Velez & Otaegui, 2017). Many methods adopt simple algorithms or early deep learning techniques. (Ibrahim et al., 2020) demonstrated that tracking based on minimal spatial overlap across consecutive frames can compute relative speed by comparing the change in bounding box positions. However, such simplistic approaches are prone to failure when objects are occluded or when abrupt manoeuvres occur, limiting their robustness in dynamic real-world environments.

However, real-world driving scenarios are complex, and addressing these tasks in isolation often falls short of providing reliable collision warnings. More recent research has therefore moved toward integrated, end-to-end approaches that combine all these elements into a single framework (Fang et al., 2024). In these modern systems, the camera feed is processed in real time to identify relevant road elements like vehicles and pedestrians, to follow the detected objects over time to understand their motion patterns, and to analyse changes in object size and movement (such as the rate at which a vehicle's bounding box grows) to estimate whether a collision might occur. This integrated approach allows the system to consider both the spatial arrangement (where objects are) and the temporal dynamics (how objects move) together, leading to more accurate and timely collision warnings.

Some researchers have explored end-to-end deep learning models that directly predict collision probabilities from video streams (Jung et al., 2016; Kim et al., 2021; Strickland et al., 2017). Although these methods promise to capture complex spatial and temporal dynamics, they generally require extensive training on large, well-annotated datasets of collision events. Moreover, such methods can be computationally demanding, making real-time implementation challenging in an embedded system (Tanu et al., 2024). Provide a comprehensive study of recent ADAS practices, highlighting not only advancements in sensor fusion, detection, and tracking but also the limitations of using real-world datasets. Such datasets are often restricted by inconsistent scenes, camera distortions, resolution differences, high video compression, and variable lighting, hindering the collection of large-scale, diverse data representative of erratic driving behavior (Janai et al., 2017). Additionally, collecting data in controlled environments becomes exponentially more expensive when diverse scenarios are considered.

Our system integrates YOLOv5, trained on a large, diverse, custom-made synthetic dataset, with DeepSORT and a novel dynamic collision-prediction algorithm to overcome these key challenges effectively.

2. METHODOLOGY

2.1 Proposed System

Our ADAS framework comprises three primary modules: vehicle detection, object tracking, and accident warning evaluation. The overall system architecture is depicted in Fig. 1. The pipeline begins with real-time video acquisition from a dashcam positioned on the ego vehicle. Our proposed system works with a video resolution of 1920x1080 (Width x Height). The video's frame rate is 30 frames per second (FPS). YOLOv5 processes each frame to detect vehicles with bounding boxes. Detected objects are then passed to DeepSORT (Wojke et al., 2017), which maintains consistent identities across frames despite occlusions or rapid movements. Finally, the accident warning evaluator monitors the temporal variation in bounding box areas to predict potential collisions.

For object detection, we chose YOLOv5 for its outstanding performance in real-time object detection. The YOLOv'X models are preferred over R-CNN (Girshick et al., 2013) and Faster R-CNN (Ren et al., 2015) for real-time pedestrian identification. While R-CNN achieves a detection rate of 0.5 Hz, Faster R-CNN achieves around 17 Hz. In comparison, YOLO9000 achieves a detection rate of 90 Hz at a resolution of 288x288 pixels. The YOLOv5 surpasses the YOLO9000 (Redmon & Farhadi, 2017) regarding mean average precision (mAP) and inference speed.

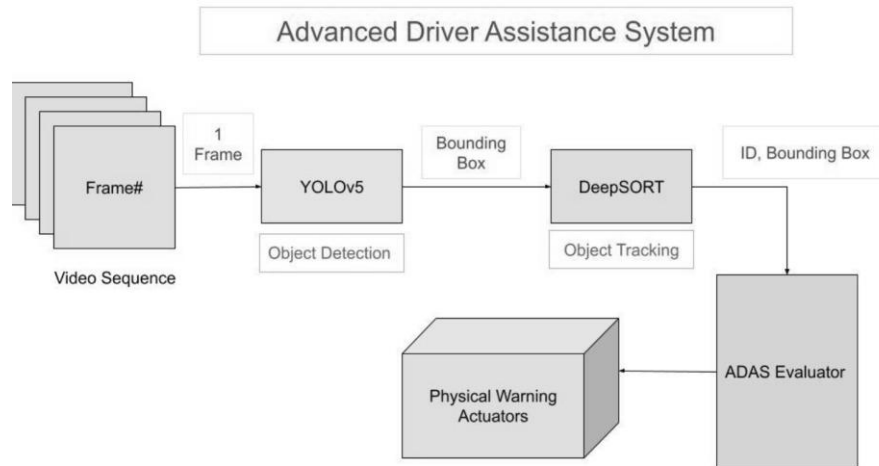


Figure 1: The block diagram of the proposed ADAS pipeline

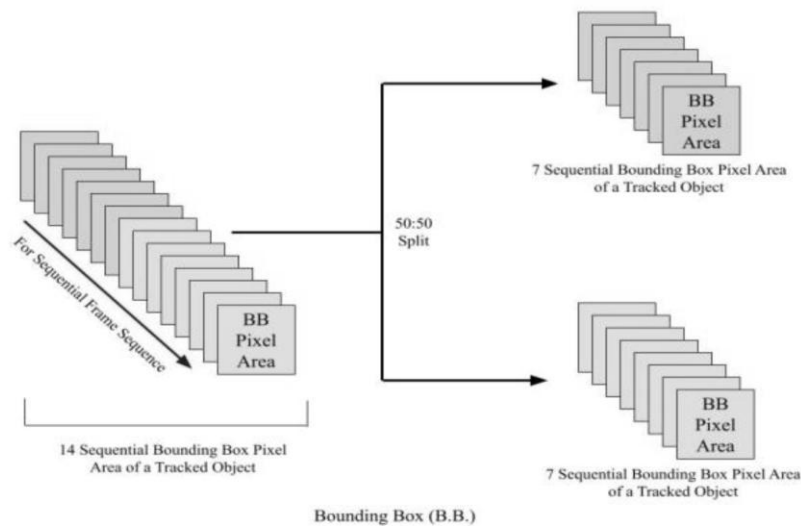


Figure 2: Illustration of a 50:50 split of the 14 sequential bounding

Additionally, for applications that demand fast inference, simple integration, and flexible deployment on resource-constrained devices, YOLOv5 has emerged as a favoured choice over other YOLO versions due to its lightweight design and superior real-time performance. YOLOv5 performs better than Mask R-CNN and Faster R-CNN in terms of operating frequency and average precision (AP). The vehicle detection system will utilise YOLOv5m weights trained on the MSCOCO2017 dataset and a custom-made dataset containing 50,092 samples. This combination of public and custom synthetic data will ensure fast, precise vehicle detection.

The tracking method DeepSORT is employed in conjunction with YOLOv5. It is an improved version of the Simple Online Real-Time Tracking (SORT) (Bewley et al., 2016), a precondition for this system. DeepSORT has demonstrated strong multi-object tracking accuracy (MOTA) in the MOT16 tracking

benchmark (Milan et al., 2016). It has surpassed other tracking methods in terms of MOTA, ID Switching, and speed in the MOT16 benchmark.

We combine YOLOv5 as the multi-object detector and DeepSORT as the multi-object tracker to gather crucial information about tracked objects, including their IDs and bounding box coordinates (X_{min} , Y_{min} , X_{max} , Y_{max}). This data is then propagated to our ADAS Evaluator module.

We developed the 'Evaluator module' to provide essential information, including the ID of each tracked object, its bounding box coordinates. (X_{min} , Y_{min} , X_{max} , Y_{max}) The computed pixel area of each bounding box using the $P_{area}(X_{min}, Y_{min}, X_{max}, Y_{max})$ action, and the serial number of each video frame.

After associating a 14-pixel area of a tracked object (with its assigned ID), those 14-pixel areas are evenly divided into two sets, S_1 and S_2 , based on their frame numbers (Figure 2). A higher frame number indicates that the frame comes later than the lower one. Each frame number is distinct. The pixel-area delta of the associated tracked object will be calculated for each set. The set S_1 contains the latest half of the 14-pixel area of a given tracked object; the set S_2 contains the prior half. Let $A = [a_1, a_2, \dots, a_{14}]$ be the list of 14 frames for a tracked object (with a_1 corresponding to the earliest frame and a_{14} to the latest). This 14-frame window (0.46s at 30 FPS) was chosen to balance noise filtering with low latency; a larger 30-frame window would introduce a 1-second delay, which is unsuitable for real-time safety. Now splitting the list into two sets based on frame order: $S_1 = \{a_i \mid 8 \leq i \leq 14\}$, $S_2 = \{a_i \mid 1 \leq i \leq 7\}$

$$P_{area}(a) = |(Y_{max} - Y_{min}) \cdot (X_{max} - X_{min})| \quad (1)$$

Equation 1 computes the pixel area of the tracked object for a single frame. Here, (X_{min} , Y_{min} , X_{max} , Y_{max}) corresponding to the bounding box coordinates of the tracked object for a single frame.

Now, the Pixel Area Delta is:

$$\Delta P_{area} = \sum_{i=8}^{14} P_{area}(a_i) - \sum_{i=1}^7 P_{area}(a_i) \quad (2)$$

Tracked Object Pixel Area Delta per Second

$$(OPADS) = \frac{\Delta P_{area}}{(Serial_{LastFrame\ of\ S_1} - Serial_{LastFrame\ of\ S_2})} \times \frac{1}{30} \quad (3)$$

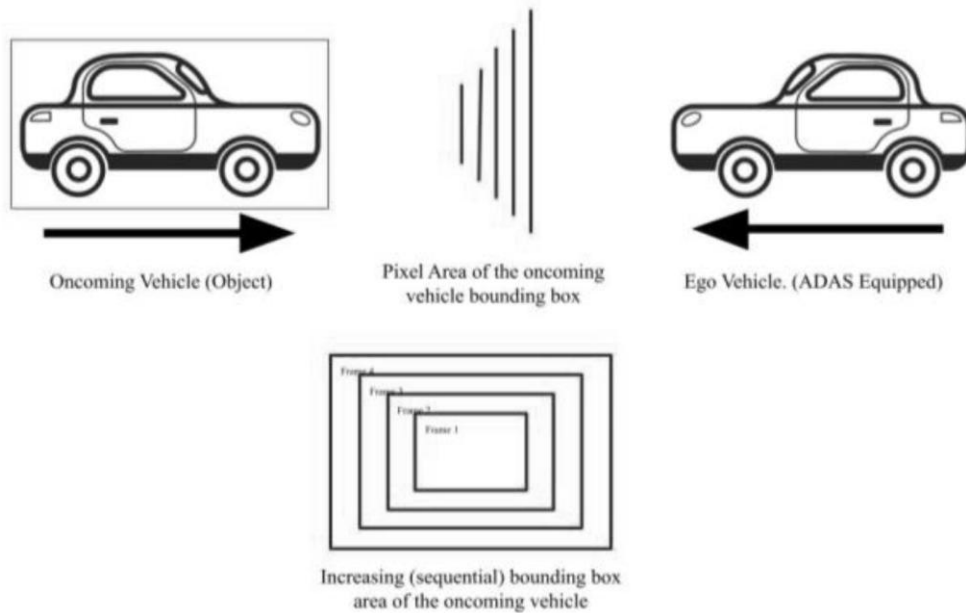


Figure 3 : Illustration of the increasing bounding box area size of an oncoming vehicle.

An object that is moving away from the ego vehicle will have a negative Tracked Object Pixel Area Delta Per Second, and the object that is moving towards the ego vehicle will have a positive (higher) Tracked Object Pixel Area Delta Per Second, as shown in Figure 3.

Within our ADAS Evaluator, the term "Object Pixel Area Delta Per Second" (OPADS) denotes the delta in the pixel area of a bounding box for an object during a defined period. The factor of (1/30) in Equation 3 is a normalisation constant based on the 30 FPS capture rate. It converts the pixel area change from a 'per-frame' value into a 'per-second' rate (OPADS). We formulated and simulated a scenario using the CARLA simulator to calculate this threshold. A stationary vehicle was placed in a CARLA Simulator environment, while an ego vehicle was driven towards it to replicate an imminent crash. Many time-to-collision (TTC) situations were stored when OPADS values were recorded at a TTC of 2 seconds. The ego vehicle was simulated at 40, 50, 60, 70, and 80 km/h to cover a range of Time-to-Collision (TTC) scenarios. The ego vehicle was started 30 meters from the stationary target vehicle, leading to varying relative velocities between the ego and the target. The average OPADS was computed and set as the threshold OPADS for the warning evaluator.

To find the optimal coordinates of the Critical Section Area, we placed a stationary vehicle 30 meters from the ego vehicle in the CARLA Simulator. Next, we used the ego car's observed lanes to determine the slope of the Critical Section Area (Figure 4). Then we utilised this specific configuration to adjust the coordinates (X_1, Y_1) precisely, (X_2, Y_2) , (X_3, Y_3) , and (X_4, Y_4) of the Critical Section Area for vehicles approximately 50 meters away from the ego vehicle. Figure 4 visually represents the calibration of the Critical Section Area and the placements of the coordinates. The provided coordinates of the Critical Section Area enable precise measurement and analysis of OPADS. As a result, we can determine the optimal threshold for our custom warning evaluator.

In our final evaluation step, the (X_1, Y_1) , (X_2, Y_2) , (X_3, Y_3) , (X_4, Y_4) are the pixel coordinates as shown in Figure 4, which identifies the critical section of the frame. $(X_{min}, Y_{min}, X_{max}, Y_{max})$ are the bounding box coordinates of a tracked object.



Figure 4 : Illustration of coordinates (specific pixel point in frame) identifying the critical section $(X_1 = 150, Y_1 = 1050)$, $(X_2 = 1525, Y_2 = 1050)$, $(X_3 = 930, Y_3 = 550)$, $(X_4 = 1060, Y_4 = 550)$

The Centroid of the bounding box is calculated by,

$$Mid_x = \frac{X_{max} - X_{min}}{2} + X_{min} \quad (4)$$

$$Mid_y = \frac{Y_{max} - Y_{min}}{2} + Y_{min} \quad (5)$$

The final evaluation, which will trigger a warning for a tracked object if all of the following conditions are true,

1. Object Pixel Area Delta Per Second > Object Pixel Area Delta Per Second Threshold.
2. $\frac{Mid_y - Y_1}{Mid_x - X_1} \geq \frac{Y_3 - Y_1}{X_3 - X_1}$
3. $\frac{Mid_y - Y_2}{Mid_x - X_2} \geq \frac{Y_4 - Y_2}{X_4 - X_2}$

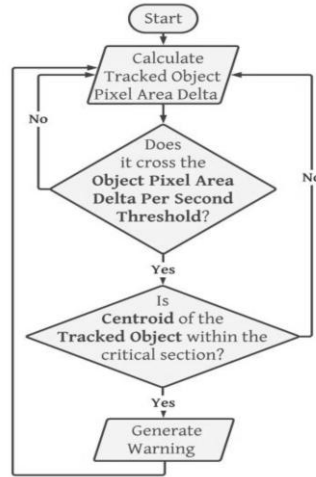


Figure 5 : The flow diagram of our proposed ADAS Evaluator
The entire evaluation process flow graph is shown in Figure 5.

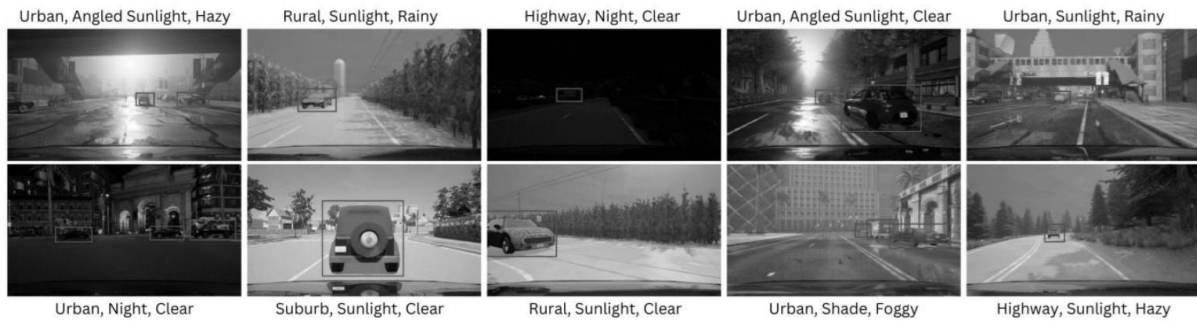


Figure 6: A careful curation of sample images from the dataset, with precise annotations to accurately emphasise the cars with precision and clarity.



Figure 7: The left one illustrates the annotation of vehicles within 50 meters from the ego vehicle. The middle one illustrates ignoring vehicle annotations beyond 50 meters from the ego vehicle. The right one is an example of a zero-vehicle instance in the training dataset.

2.2 Dataset Description

In this study, the *MS-COCO 2017* dataset (Lin et al., 2014) is used to pre-train the YOLOv5 multi-object detector. We created a new, custom-vehicle-annotated dataset to train the YOLOv5 multi-object detector. 330,000 images are used from the MS-COCO 2017 dataset, which covers a wide range of

topics, including many vehicles such as automobiles, trucks, buses, and motorcycles. In addition, 50,092 of our custom-made images are specifically annotated for vehicles to train the final YOLOv5 multi-object detector.

To overcome the challenge of collecting diverse and secure data, including different road conditions, weather scenarios, and lighting variations, we employed synthetic data generation. We utilise the CARLA simulator to generate a synthetic, custom-made dataset. (Dosovitskiy et al., 2017) Created the Car Learning to Act (CARLA) simulator, a framework that allows researchers to simulate extremely lifelike virtual environments. CARLA’s advanced simulation capabilities allow us to emulate a variety of driving scenarios with precise control over environmental factors.

The dataset comprises 50,092 images (equivalent to 38.3 Gigabytes) captured from an ego vehicle’s perspective. Each image is rendered at a resolution of 960×540 pixels with a 90° field of view, replicating a typical dashcam setup. Various conditions are simulated by varying lighting, weather, and road conditions. Specifically, the lighting conditions include daylight, shade, night-time, and angled sunlight; the weather scenarios range from clear to foggy, hazy, and rainy; and the road types span urban, suburban, rural, and highway settings. Every environment has distinct features, including changing levels of traffic density, object arrangement, and object shade. The wide range of variations allows us to evaluate and improve our ability to recognise things in various driving situations.

The dataset obtained, as shown in Figure 6, provides a comprehensive representation of real driving situations, including photos that capture diverse lighting, weather, and road conditions. The variety of situations this range offers establishes a solid basis for training and assessing models, guaranteeing their efficacy in demanding autonomous and computer vision tasks.

To accurately align 3D bounding boxes from the CARLA simulator with collected images and generate precise 2D bounding boxes, we implemented a mapping process that extracts the maximum and minimum X, Y, and Z coordinates from each 3D box. This process then leverages the X and Y values to produce a streamlined 2D representation, reducing the original 8 coordinates to 4. This method enabled us to annotate 50,092 vehicle images captured from the ego vehicle's perspective with exceptional accuracy. Additionally, by applying a 50-meter distance threshold, we ensured that only vehicles within a clearly defined range were annotated, thereby optimizing visual clarity and minimizing issues such as pixelation.

Table 1 : Comparison between pre-trained YOLOv5 (m6) and our custom trained YOLOv5 model

<i>Lighting</i>	<i>Weather</i>	<i>Environment</i>	<i>Mean Average</i>	<i>Mean Average Precision</i>
			<i>Precision</i> <i>(mAP:0.5) %</i> <i>Pretrained Model</i>	<i>(mAP:0.5) %</i> <i>Our Trained Model</i>
Daylight	Clear	Urban	74.31	81.45
Daylight	Foggy	Rural	70.23	78.33
Daylight	Rainy	Highway	65.36	71.06
Shade	Clear	Urban	71.72	78.84
Shade	Rainy	Highway	59.65	66.92
Shade	Foggy	Rural	68.75	74.03
Angled Sun Light	Clear	Urban	66.23	75.52
Angled Sun Light	Foggy	Urban	63.79	72.12
Angled Sun	Rainy	Urban	55.32	63.28

Light				
Night	Clear	Urban	28.24	45.23
Average			62.36	70.67

Table 2 Time-to-collision result of our proposed ADAS model.

<i>Lighting</i>	<i>Weather</i>	<i>Environment</i>	<i>Average time-to-collision (ATTC)</i> <i>(seconds)</i>	
Daylight	Clear	Urban	3.02	
Daylight	Foggy	Rural	2.9	
Daylight	Rainy	Highway	2.6	
Shade	Clear	Urban	2.9	
Shade	Rainy	Highway	2.5	
Shade	Foggy	Rural	2.7	
Angled Sun Light	Clear	Urban	2.8	
Angled Sun Light	Foggy	Urban	2.7	
Angled Sun Light	Rainy	Urban	2.3	
Night	Clear	Urban	1.7	
Average			2.64	

The dataset annotation process was executed in two essential steps to ensure compatibility with the YOLOv5 training framework. Initially, we annotated the dataset using the widely accepted PASCAL VOC format, which provides necessary information such as bounding boxes and labels, ensuring seamless interoperability with various computer vision tools. We then converted these annotations to the YOLOv5 format, complete with the required metadata, ensuring the dataset was perfectly aligned with YOLOv5's specifications for model training and inference. This sequential conversion facilitated efficient utilisation of the annotations within the framework and maintained the benefits of the PASCAL VOC format's interoperability and precise structure.

Our custom dataset includes images with zero annotations that unnecessarily increase the dataset size but do not adversely affect YOLOv5 training. Additionally, the CARLA Simulator occasionally produces incorrect bounding box sizes for motorbikes and bicycles.

However, our synthetic dataset offers significant advantages over real-world data collection. It is highly cost-effective and scalable, and it provides complete control over environmental variables such as lighting, weather, and traffic conditions. Moreover, it eliminates safety and privacy concerns, enables efficient iterative modifications, and allows for the ready simulation of accident scenarios, making it more practical and resource-efficient.

2.3 Setup and Training

The training was executed on a system with a Ryzen 7 5800X CPU, 32 GB of DDR4 RAM, an RTX 3070 GPU (8 GB VRAM), and a 256 GB SSD. The process began with meticulously annotating over

50,000 images, assigning precise bounding-box coordinates and class labels to each vehicle to boost training efficiency. Transfer learning was applied by initializing the YOLOv5 model with pre-trained YOLOv5 m6 weights, using the "yolov5m6.yaml" configuration file. Training parameters included an input size of 640×640 pixels, a batch size of 16, and 100 epochs. For vehicle detection during inference, a confidence threshold of 0.25 was selected as the score value to consider a detection valid. The entire training process took approximately 23 hours.

We assess our system using standard metrics. The evaluation metrics include mean average precision (mAP) at IoU thresholds of 0.5 and from 0.5 to 0.95, precision and recall for detection accuracy, accident detection rate (ADR) to measure the ratio of correctly identified collision events to total collisions, and average time-to-collision (ATTC), which quantifies the temporal gap between warning issuance and the actual collision.

3. RESULTS AND DISCUSSION

After training the YOLOv5 model for 100 epochs on 50,092 images, the model achieved a mAP of 0.8461 at an IoU of 0.5 and a mAP of 0.7029 averaged over IoU thresholds from 0.5 to 0.95. With a single class ('vehicle'), the reported precision is 0.927, indicating a low rate of false positives, and the recall is 0.7934, reflecting a minimised rate of false negatives.

We evaluated vehicle detection by designing 10 distinct scenarios in the CARLA simulator, capturing 30-second video clips per scenario, applying a 50-meter distance threshold to filter out distant vehicles, and manually annotating all test footage with bounding boxes. The performance of our custom-trained YOLOv5 model was then compared to the pre-trained YOLOv5 m6 model using ground truth data and mean average precision (mAP) as the primary metric. Table 1 shows that, on average, our trained model performed (8.31%) better than the pre-trained YOLOv5 (YOLOv5m6) model. We saw the most significant difference in 'Night' light conditions, where our model performed ~17% better than the YOLOv5 pre-trained model.

Using DeepSORT for tracking, our system maintained stable vehicle identities across frames, and the accident warning evaluator successfully predicted collisions. We evaluated the collision warning feature of our ADAS module by designing 10 scenarios in the CARLA simulator, each with 3 sub-variations, resulting in 30 collision situations. The ego vehicle deliberately engaged in head-on collisions at speeds of 30-45 km/h, capturing 12-second video clips to analyse pre-collision moments. The system's effectiveness was assessed using the average time-to-collision (ATTC) metric, which measures the time interval between the issuance of a warning and the collision. ATTC was calculated by,

$$TTC = \frac{frame_{collision} - frame_{warning}}{30} \quad (6)$$

Where, $frame_{collision}$ is the collision occurrence frame number, and $frame_{warning}$ is the initial collision warning frame number. Our system achieved an ATTC of 2.64 seconds across various conditions and an accident detection rate (ADR) of 80.43%. Our method achieves real-time performance while addressing prior limitations, making it more viable for real-world driving scenarios by ensuring timely and accurate collision alerts.

4. CONCLUSIONS

In this study, we propose a cost-effective Advanced Driver Assistance System (ADAS) that utilises computer vision techniques to enhance road safety, particularly in resource-limited settings. The system incorporates CMOS RGB cameras, YOLOv5 for real-time vehicle detection, DeepSORT for robust multi-object tracking, and a custom collision-warning module based on changes in bounding-box pixel area. Systematic validation in the CARLA simulator demonstrated the system's capability to simulate real-world driving hazards, accurately detect vehicles, and deliver timely collision alerts. Results indicate that our ADAS framework achieved an accident detection rate (ADR) of 80.43% and an average time-to-collision (ATTC) of 2.6 seconds in simulated settings, showing it can identify potential

accidents early enough for driver intervention. The ADR can be improved by integrating sensor fusion (combining camera data with radar/ultrasonic sensors) to improve depth accuracy in poor weather. The system's ability to process video data in real time at 30 FPS, coupled with its reliance on affordable hardware, offers a notable improvement over offline ADAS methods and provides a scalable, low-cost alternative to traditional sensor-based systems such as LiDAR and RADAR. Although initial simulation results are promising, all tests were conducted in controlled environments using synthetic data generated by the CARLA simulator. Therefore, further validation and fine-tuning in real-world scenarios will be necessary. Integrating data from real-world datasets, such as the Waymo Open Dataset, will help improve performance under unpredictable conditions, including variations in driver behaviour, road anomalies, and environmental factors not fully represented in simulation. The next phase involves deploying the system in actual vehicles to evaluate its real-world applicability and enhance its robustness across different traffic situations. Future efforts will also focus on optimising the system's deployment architecture, ensuring seamless integration with vehicle hardware, and tackling challenges such as real-time processing on low-power devices. A limitation of this study is the testing speed range of 30–45 km/h; future work will focus on validating the system at high-speed highway levels. Ultimately, the aim is to transfer this research into a practical, life-saving platform that improves road safety and can be adopted across a broad range of vehicles, especially in areas with limited access to advanced ADAS technologies.

DECLARATION OF USE OF AI

The authors declare that they used artificial intelligence (AI) tools to assist in preparing this manuscript. All content was reviewed and approved by the authors, who assume full responsibility for the final version. The AI tool did not produce any original scientific results or conclusions.

REFERENCES

- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and real-time tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, 3464–3468. <https://doi.org/10.1109/ICIP.2016.7533003>
- Borrego-Carazo, J., Castells-Rufas, D., Biempica, E., & Carrabina, J. (2020). Resource-Constrained Machine Learning for ADAS: A Systematic Review. *IEEE Access*, 8, 40573–40598. <https://doi.org/10.1109/ACCESS.2020.2976513>
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). *CARLA: An Open Urban Driving Simulator* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1711.03938>
- Fang, J., Qiao, J., Xue, J., & Li, Z. (2024). Vision-Based Traffic Accident Detection and Anticipation: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(4), 1983–1999. <https://doi.org/10.1109/TCSVT.2023.3307655>
- Forward Collision Warning for Autonomous Driving. (2022). *Journal of Logistics, Informatics and Service Science*. <https://doi.org/10.33168/LISS.2022.0315>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). *Rich feature hierarchies for accurate object detection and semantic segmentation* (Version 5). arXiv. <https://doi.org/10.48550/ARXIV.1311.2524>
- Gouribhatla, R., & Pulugurtha, S. S. (2022). Drivers' behavior when driving vehicles with or without advanced driver assistance systems: A driver simulator-based study. *Transportation Research Interdisciplinary Perspectives*, 13, 100545. <https://doi.org/10.1016/j.trip.2022.100545>
- Ibrahim, A. M., Hassan, R. M., Tawfiles, A. E., Ismail, T., & Darweesh, M. S. (2020). Real-Time Collision Warning System Based on Computer Vision Using Mono Camera. *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 60–64. <https://doi.org/10.1109/NILES50944.2020.9257941>
- Janai, J., Güney, F., Behl, A., & Geiger, A. (2017). *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art* (Version 3). arXiv. <https://doi.org/10.48550/ARXIV.1704.05519>

- Joshi, V., Gote, R., Kakad, D., Dode, S., & Jaiswal, R. C. (2024). *Collision warning system for vehicles using deep learning*. 030003. <https://doi.org/10.1063/5.0227567>
- Jung, H., Choi, M.-K., Soon, K., & Jung, W. Y. (2016). *End-to-End Pedestrian Collision Warning System based on a Convolutional Neural Network with Semantic Segmentation* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1612.06558>
- Kim, H., Lee, K., Hwang, G., & Suh, C. (2021). Predicting vehicle collisions using data collected from video games. *Machine Vision and Applications*, 32(4), 93. <https://doi.org/10.1007/s00138-021-01217-2>
- Kukkala, V. K., Tunnell, J., Pasricha, S., & Bradley, T. (2018). Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles. *IEEE Consumer Electronics Magazine*, 7(5), 18–25. <https://doi.org/10.1109/MCE.2018.2828440>
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014). *Microsoft COCO: Common Objects in Context* (Version 3). arXiv. <https://doi.org/10.48550/ARXIV.1405.0312>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). *SSD: Single Shot MultiBox Detector*. <https://doi.org/10.48550/ARXIV.1512.02325>
- Madhumitha, G., Senthilnathan, R., Ayaz, K. M., Vignesh, J., & Madhu, K. (2020). Estimation of Collision Priority on Traffic Videos using Deep Learning. *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, 1–6. <https://doi.org/10.1109/ICMLANT50963.2020.9355992>
- Milan, A., Leal-Taixe, L., Reid, I., Roth, S., & Schindler, K. (2016). *MOT16: A Benchmark for Multi-Object Tracking* (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.1603.00831>
- Nidamanuri, J., Nibhanupudi, C., Assfalg, R., & Venkataraman, H. (2022). A Progressive Review: Emerging Technologies for ADAS Driven Solutions. *IEEE Transactions on Intelligent Vehicles*, 7(2), 326–341. <https://doi.org/10.1109/TIV.2021.3122898>
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525. <https://doi.org/10.1109/CVPR.2017.690>
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1804.02767>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* (Version 3). arXiv. <https://doi.org/10.48550/ARXIV.1506.01497>
- Road traffic injuries. (n.d.). *World Health Organization*.
- Rolison, J. J., Regev, S., Moutari, S., & Feeney, A. (2018). What are the factors that contribute to road accidents? An assessment of law enforcement views, ordinary drivers' opinions, and road accident records. *Accident Analysis & Prevention*, 115, 11–24. <https://doi.org/10.1016/j.aap.2018.02.025>
- Strickland, M., Fainekos, G., & Amor, H. B. (2017). *Deep Predictive Models for Collision Risk Assessment in Autonomous Driving* (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.1711.10453>
- Tanu, B. J., Ahmed, Md. S., Islam, Md. A., & Ferdosi, B. J. (2024). A Comparative Study of Recent Practices and Technologies in Advanced Driver Assistance System. *International Journal of Computer Applications*, 186(28), 54–66. <https://doi.org/10.5120/ijca2024923796>
- Velez, G., & Otaegui, O. (2017). Embedding vision-based advanced driver assistance systems: A survey. *IET Intelligent Transport Systems*, 11(3), 103–112. <https://doi.org/10.1049/iet-its.2016.0026>
- Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. *2017 IEEE International Conference on Image Processing (ICIP)*, 3645–3649. <https://doi.org/10.1109/ICIP.2017.8296962>