IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

Drivers Drowsiness and Pedestrian Crosswalk Detection

Semina Farook[1], Harikrishnan S R[2], Riyad A[3]
Student, MCA, CHMM College for Advanced Studies, Trivandrum, India[1]
Assistant Professor, MCA, CHMM College for Advanced Studies, Trivandrum, India[2,3]

Abstract: *Road safety is a major concern around the world, especially when it comes to accidents caused by tired drivers and people crossing the road. This project focuses on solving both problems by creating a real-time system that can detect when a driver is feeling drowsy and when pedestrians are near a crosswalk.The first part of the system watches the driver's face using a camera. It uses AI and computer vision tools like CNNs and OpenCV to track features like eye movement and yawning. If the system notices that the driver's eyes are closing for too long or they're yawning a lot—common signs of sleepiness—it will send out a sound or visual alert to wake them up or suggest taking a break.*
The second part of the system helps keep pedestrians safe. It uses a powerful detection model called YOLO to recognize crosswalks and people near the road in real time. This helps the driver stay aware of their surroundings, especially in busy city areas where people often cross the street. If someone is detected near a crosswalk, the system alerts the driver immediately to help avoid accidents.This system is built using Python and deep learning tools like TensorFlow, PyTorch, and Keras. The development process involved collecting and preparing data, training models, and fine-tuning them to work well in different situations.Overall, this project aims to make driving safer by reducing the risk of accidents caused by drowsy driving or pedestrian collisions
.

Keywords: Machine Learning, Deep Learning, Convolutional Neural Network, YOLO V8, PyTorch.

I. INTRODUCTION

Every year, thousands of road accidents happen because drivers fall asleep at the wheel or because pedestrians suddenly appear on the road. These situations are dangerous and often happen without warning. With the rise of technology, we now have the chance to use smart systems to help prevent such accidents and make our roads safer.This project is focused on building a real-time system that can do two important things: detect when a driver is getting drowsy, and spot pedestrians near crosswalks. By combining artificial intelligence (AI) and computer vision, the system can watch the driver and the road at the same time, helping avoid accidents before they happen.The goal is to create a simple but powerful safety tool that can alert drivers when they're too tired to drive safely or when there are pedestrians nearby, especially in busy areas. This kind of system can be a big step toward reducing road accidents and saving lives.

II. LITERATURE REVIEW

Road safety is a paramount concern worldwide, with driver fatigue and pedestrian-related accidents being significant contributors to traffic incidents. This project aims to address these issues through the development of a real-time Driver Drowsiness and Pedestrian Crosswalk Detection system, which integrates advanced deep learning and computer vision technologies.The first component of the system focuses on detecting driver drowsiness, a leading cause of road accidents. Utilizing Convolutional Neural Networks (CNNs) in conjunction with OpenCV, the system captures and analyzes real-time video feeds from a camera positioned to face the driver. It monitors facial landmarks, particularly the eyes and mouth, to identify signs of fatigue such as prolonged eye closure (indicative of microsleeps) and yawning. By calculating metrics like Eye Aspect Ratio (EAR) and mouth openness, the system can effectively recognize drowsiness

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, July 2025 Impact Factor: 7.67

patterns. Upon detecting signs of fatigue, the system promptly triggers an alert—either auditory or visual—to encourage the driver to take corrective action, thereby mitigating the risk of potential accidents.

The second component of the system addresses pedestrian safety by detecting crosswalks and pedestrians using the YOLO (You Only Look Once) framework, known for its speed and accuracy in object detection. YOLO can identify crosswalk patterns on the road and recognize pedestrians in real time, providing essential situational awareness to the driver. This functionality is particularly beneficial in urban settings and areas with high pedestrian traffic. The module can issue timely alerts when pedestrians are detected near crosswalks, especially when the vehicle is in motion, thus aiding drivers in avoiding collisions.

The system is developed using Python and employs deep learning frameworks such as TensorFlow or PyTorch, along with Keras for model training and deployment. Data preprocessing and model selection are critical phases of the project, involving the collection of relevant datasets for both drowsiness and crosswalk detection, as well as the application of augmentation and normalization techniques to enhance model performance and generalization.

III. PROPOSED SYSTEM

The system we're proposing is a smart, all-in-one solution designed to make driving safer by focusing on two major risks: driver drowsiness and pedestrian detection. Instead of relying on older sensor-based systems or only addressing one problem at a time, this system uses modern technologies like artificial intelligence and computer vision to watch both the driver and the road in real time. Inside the vehicle, a webcam monitors the driver's face to spot signs of fatigue—like closed eyes, frequent blinking, yawning, or a drooping head. Using tools like CNNs (Convolutional Neural Networks) and OpenCV, the system can understand these small but important signals. If the driver appears drowsy, it immediately sends out an alert—either a sound, a flashing light, or both—urging them to refocus or take a break. At the same time, a second camera looks ahead to detect pedestrians and crosswalks on the road using a powerful object detection model called YOLO (You Only Look Once). This helps the system quickly identify people walking nearby or crossing the street—even in places without clearly marked crosswalks or in low-light conditions. If someone is detected, the driver gets a real-time warning, giving them enough time to slow down or stop safely. What makes this system stand out is that it combines internal monitoring of the driver with external awareness of the road, creating a complete safety net. It's designed to work in all types of weather and lighting, and it's lightweight and affordable enough to be added to almost any vehicle, including older models. Overall, this system aims to reduce accidents, protect both drivers and pedestrians, and move us toward a smarter, safer future on the roads.

IV. ALGORITHM

Convolutional Neural Networks

A Convolutional Neural Network, or CNN for short, is a special type of artificial intelligence that's really good at understanding images. Just like how our eyes and brain work together to recognize faces, objects, or scenes, a CNN helps computers do the same. It looks at an image in small parts, finds important patterns like edges, shapes, or colors, and then uses that information to figure out what the image shows. For example, in a driver drowsiness system, a CNN can learn to recognize when a person's eyes are closing or when they're yawning by studying lots of pictures of both sleepy and alert faces. Over time, it gets better and better at making these decisions on its own. This makes CNNs a powerful and essential tool for tasks like face detection, medical imaging, self-driving cars, and more.

YOLOv8 (You Only Look Once – Version 8)

YOLOv8 (You Only Look Once version 8) is an advanced real-time object detection model that helps computers quickly and accurately identify objects in images or videos—like people, cars, or traffic signs. It works in three main steps: first, it looks at the image and pulls out important visual features (like shapes and edges), then it combines these features in a smart way to understand both the big picture and small details, and finally, it predicts what objects are in the image, where they are, and how confident it is. YOLOv8 is fast, accurate, and works even on smaller devices like phones or car cameras, making it perfect for tasks like pedestrian detection or monitoring driver drowsiness in real time.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

V. PACKAGES

OpenCV

OpenCV (short for Open Source Computer Vision) is a powerful and widely used tool that helps computers understand and work with images and videos. Just like our eyes and brain help us see and recognize things, OpenCV gives similar "vision" to machines. It can do things like capture video from a webcam, detect faces and eyes, track movements, and even recognize gestures or objects. In this project, OpenCV is used to watch the driver's face in real time—checking if their eyes are closing, if they're yawning, or if their head is dropping—so it can spot signs of drowsiness early. It's fast, flexible, and works well with cameras, making it perfect for building real-time safety systems like this one.

PYTorch

PyTorch is a popular and easy-to-use deep learning library that helps computers learn from data—just like how humans learn from experience. Think of it as a toolkit for building smart systems that can recognize images, understand speech, or make predictions. In this project, PyTorch can be used to train a model that learns to detect whether a driver looks sleepy or alert, based on thousands of example images. It's known for being flexible and beginner-friendly, making it easier for developers and researchers to experiment, test ideas, and build powerful AI models quickly. Plus, PyTorch works really well with GPUs (graphic cards), so it can train models faster and handle real-time tasks like monitoring drivers or detecting pedestrians.

Keras

Keras is a simple and easy-to-use tool that helps you build deep learning models without needing to write a lot of complicated code. You can think of it like a user-friendly interface for creating smart systems that can learn from data—such as detecting if a driver is sleepy by analyzing their facial expressions. Keras runs on top of more powerful libraries like TensorFlow and lets you quickly design and train models using just a few lines of code. It's great for beginners and professionals alike because it makes building AI models faster, easier, and more understandable. In this project, Keras is especially helpful for creating and training the Convolutional Neural Network (CNN) used to detect signs of drowsiness in real time

VI. EXPERIMENTAL RESULTS & PERFORMANCE EVALUATION

After building the system, we tested it to see how well it actually works in real-world situations. We ran experiments using video footage and image datasets that include both alert and drowsy drivers, as well as scenes with pedestrians and crosswalks in different environments (like daytime, nighttime, and crowded streets). The goal was to check how accurately and quickly the system could detect signs of fatigue in drivers and recognize pedestrians on the road.

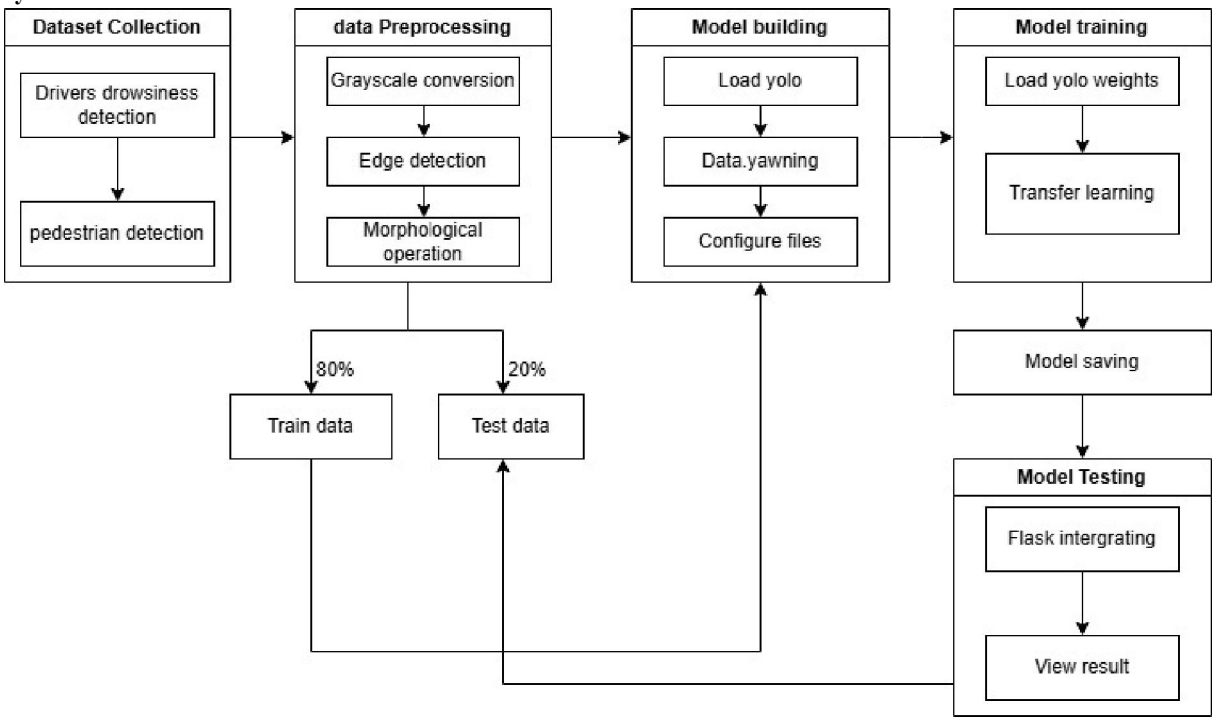
For driver drowsiness detection, the system successfully identified key signs like eye closure, yawning, and head movement with high accuracy. It was able to send out alerts within a second of spotting these signs, making it fast enough for real-time use. The model showed strong performance even under different lighting conditions or with drivers wearing glasses.

For pedestrian and crosswalk detection, the YOLOv8 model did an excellent job of spotting people and road markings in live video. It could detect pedestrians from various angles and distances, even when the lighting wasn't ideal. The system gave timely alerts when someone was near or crossing the road, allowing enough time for the driver to react.

Overall, the system performed efficiently and reliably, with high accuracy and quick response time. These results show that it can be a practical and effective solution for improving road safety. It's also lightweight and can be run on standard hardware, which means it can be used in many types of vehicles without needing expensive equipment.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

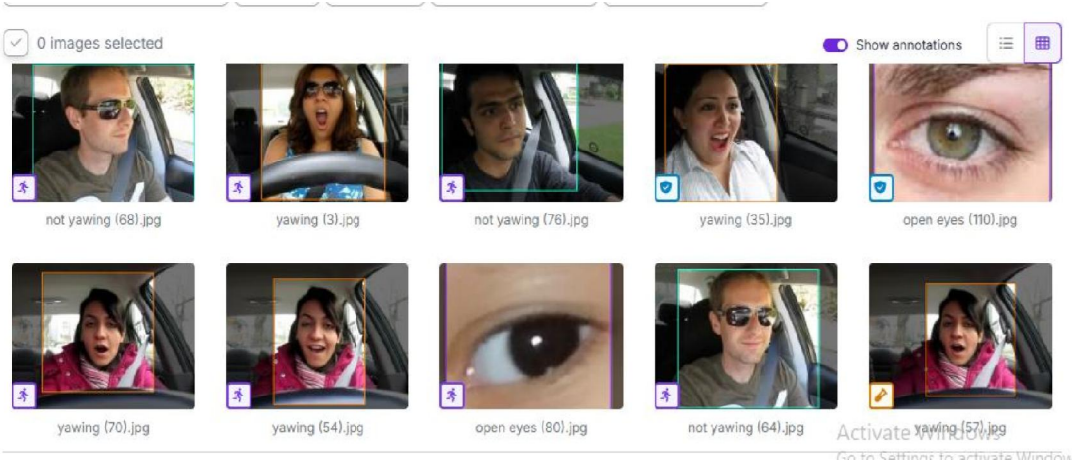
System Architecture



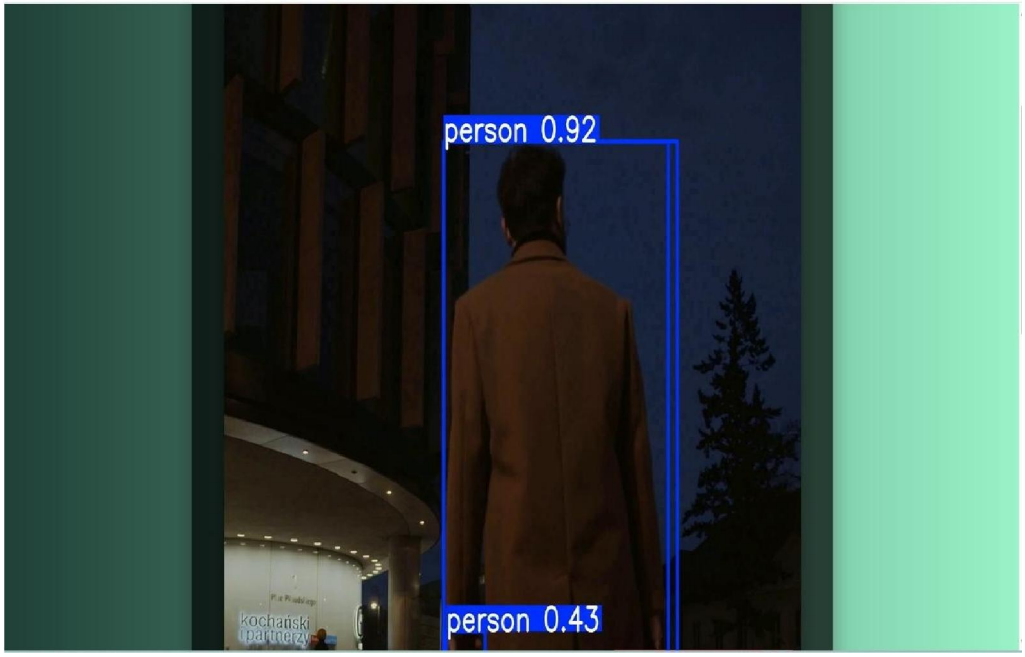
Preprocessed pedestrian crosswalk images

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

Preprocessed drivers drowsiness images



Detect pedestrian crosswalk result



VII. ACCURACY GRAPH

Evaluating the model accuracy is an essential part of the process in creating machine learning models to describe how well the model is performing in its predictions. Evaluation metrics change according to the problem type. In this post, we'll briefly learn how to check the accuracy of the regression model in R. The linear model (regression) can be a typical example of this type of problem, and the main characteristic of the regression problem is that the targets of a dataset contain the real numbers only. The errors represent how much the model is making mistakes in its prediction. The basic concept of accuracy evaluation is to compare the original target with the predicted one according to certain metrics. Accuracy of the model changes during the training process. The x-axis represents the number of training epochs (or iterations), and the yaxis shows the corresponding accuracy achieved by the model on either the training set or the validation set.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

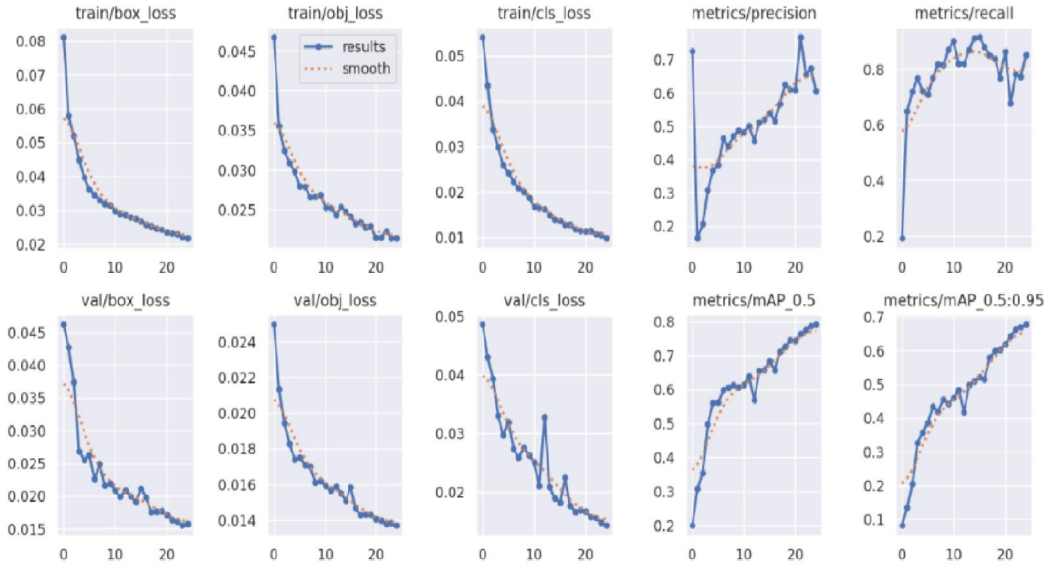


Fig: Accuracy Graph

VIII. LIMITATION

While the system works well overall, it does have a few limitations to keep in mind. For example, it might not perform accurately in very dark or overly bright conditions—like driving at night without enough light inside the car, or when strong sunlight hits the camera. Also, if the camera isn't positioned correctly or is of low quality, it might have trouble clearly seeing the driver's face or spotting pedestrians on the road. Things like sunglasses, masks, or even a tilted head can block the system's view and make it harder to detect signs of tiredness. Sometimes, it might give a false alert or miss warning signs altogether, since people show fatigue differently. On the road, heavy rain, fog, or poor visibility can affect how well the system detects pedestrians or crosswalks. And lastly, while it can warn the driver in real time, it doesn't actually control the vehicle—so it still relies on the driver to react and take action.

IX. FUTURE SCOPE

This system has a lot of potential to grow and improve. In the future, it could be made even smarter by using more advanced AI models that better understand different driver behaviors and road situations. For example, it could learn to recognize more subtle signs of drowsiness or adapt to different lighting and weather conditions without losing accuracy. The system could also be connected with the vehicle's controls—so instead of just giving alerts, it might slow the car down or safely pull it over if the driver doesn't respond. Integration with GPS and traffic data could help predict high-risk zones and give even earlier warnings. It could also be expanded to detect other distractions, like phone usage or looking away from the road. With further development, this kind of system could become a standard safety feature in all kinds of vehicles—from private cars to buses and trucks—helping reduce accidents and making roads safer for everyone.

X. CONCLUSION

This system offers a practical and intelligent solution to two major causes of road accidents: driver drowsiness and pedestrian collisions. By combining facial monitoring with real-time object detection, it helps drivers stay alert and aware of their surroundings. The use of deep learning technologies like CNN and YOLO makes the system fast, accurate, and reliable under various driving conditions. While it does have some limitations, such as sensitivity to lighting or camera placement, its ability to provide real-time alerts adds an important layer of safety to any vehicle.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

Overall, this project takes a meaningful step toward making driving safer—for both drivers and pedestrians—and lays the groundwork for smarter, AI-powered transportation systems in the future

REFERENCES

[1]. Paul Viola and Michael J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, 2001.

[2]. Soukupová, Tereza, and Jan Čech. "Real-time eye blink detection using facial landmarks." *21st Computer Vision Winter Workshop*, 2016.

[3]. Redmon, J., & Farhadi, A. "YOLOv3: An Incremental Improvement." arXiv preprint arXiv:1804.02767, 2018.

[4]. Jocher, G. et al., "YOLOv5 by Ultralytics." [Online]. Available: https://github.com/ultralytics/yolov5

[5]. Ultralytics, "YOLOv8 - State-of-the-Art Object Detection," [Online]. Available: https://docs.ultralytics.com/

[6]. Bradski, G. "The OpenCV Library." *Dr. Dobb's Journal of Software Tools*, 2000.

[7]. Paszke, A. et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library." *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8]. Chollet, F. "Keras: The Python Deep Learning library." [Online]. Available: https://keras.io/

[9]. King, D. E. "Dlib-ml: A Machine Learning Toolkit." *Journal of Machine Learning Research*, 2009.

[10]. Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., & Hariri, B. (2011). "Yawning detection using embedded smart cameras." *IEEE Transactions on Instrumentation and Measurement*, 2011