

# Driver Assistance System Based on Deep Learning for Libyan Road Environment

Ibrahim Alahrash\*<sup>1</sup> , Aeyman Hassan<sup>1</sup> 

<sup>1</sup> Department of Computer Engineering, Faculty of Engineering, University of Zawia, Zawia, Libya,

\*Corresponding author email: a.hassan@zu.edu.ly

Received: 09-11-2024 | Accepted: 01-12-2024 | Available online: 15-12-2024 | DOI:10.26629/uzjest.2024.14

## ABSTRACT

As traffic density continues to increase globally, the frequency of road accidents is expected to rise correspondingly, necessitating the development of driver assistance systems. This paper proposes a dual-camera system to monitor driver drowsiness and detect objects and traffic signs along the driving path. The system integrates two distinct models for drowsiness detection and object recognition. The drowsiness detection model utilises a driver-facing camera, employing a Haar Cascade classifier to identify facial features and monitor eye states. When the system detects that the driver's eyes remain closed for more than one second, an alert is issued to prompt the driver to regain attentiveness. The YOLOv5-based object detection model processes video frames from a road-facing camera to identify relevant objects and traffic signs in real time. The drowsiness detection model was trained locally and deployed on a laptop for real-time testing in Libyan road conditions, whereas the object detection model was trained on a cloud-based server using Google Colab. The findings of this research indicate that the proposed system has significant potential to enhance road safety through real-time drowsiness monitoring and object detection in the Libyan road environment.

**Keywords:** Drowsiness; Object Detection; Driver Assistance; Deep Learning; Libyan Road Environment

## How to cite this article:

Alahrash, I.; Hassan, A. Driver Assistance System Based on Deep Learning for Libyan Road Environment. Univ Zawia J Eng Sci Technol. 2024;2:159-168.

## نظام مساعدة السائق باستخدام التعلم العميق لبيئة الطرق الليبية

أبراهيم الأحرش<sup>1</sup>، أيمن حسين<sup>1</sup>

<sup>1</sup> قسم هندسة الحاسوب، كلية الهندسة، جامعة الزاوية، الزاوية، ليبيا

## ملخص البحث

مع تزايد كثافة حركة المرور عالمياً، يُتوقع أن يتزايد عدد الحوادث المرورية، مما يستدعي الحاجة إلى تطوير أنظمة مساعدة السائق لتوفير قيادة آمنة. يعرض هذا البحث نظاماً مزدوج الكاميرات مصمم لمراقبة نعاس السائق والكشف عن الأجسام والعلامات المرورية على مسار القيادة. يدمج النظام نموذجين مستقلين؛ الأول مخصص للكشف عن نعاس السائق، والثاني

مختص بالتعرف على الأجسام المختلفة في بيئة الطرق اللببية. يعتمد نموذج الكشف عن النعاس على كاميرا موجهة نحو وجه السائق، حيث يستخدم مُصنّف **Haar Cascade** لتحديد ملامح الوجه ومراقبة حالة العينين. في حال اكتشاف أن العينين مغلقتان لمدة تتجاوز الثانية الواحدة، يصدر النظام تنبيهًا لتحفيز السائق على التركيز في القيادة. أما نموذج الكشف عن الأجسام، المعتمد على خوارزمية **YOLOv5**، فيقوم بتحليل إطارات الفيديو المأخوذة من الكاميرا الموجهة للطريق بهدف الكشف عن الأجسام والعلامات المرورية المهمة في الوقت الفعلي. تم تدريب نموذج الكشف عن النعاس على جهاز حاسوب شخصي واختباره بشكل مبدئي في ظروف الطرق اللببية باستخدام جهاز حاسوب محمول، بينما تم تدريب نموذج الكشف عن الأجسام على خادم سحابي عبر منصة **Google Colab**. تُظهر نتائج الدراسة أن النظام المقترح يمتلك قدرة واعدة في تحسين السلامة المرورية من خلال مراقبة نعاس السائق والكشف عن الأجسام في الوقت الفعلي.

**الكلمات المفتاحية:** النعاس، الكشف عن الكائنات، مساعدة السائق، التعلم العميق، بيئة الطرق اللببية.

## 1. Introduction

The prediction of vehicle behaviours has been extensively studied using deep neural networks. For example, Khoshkangini et al. implemented a predictive model trained on a dataset containing one year of recorded driving data [1]. Although their approach yielded promising results, the model's accuracy was influenced by the characteristics of the driving environment. A comprehensive review compared various machine learning approaches for Advanced Driver-Assistance Systems (ADAS), emphasizing how anomalous road scenarios can be detected using various sensors. The datasets used for evaluation included various weather conditions, allowing for comprehensive testing. Additionally, transfer learning techniques were employed to enhance object detection on the road [2]. Recent research introduced a fusion architecture for object detection that incorporates RGB-Event inputs. This approach leverages asynchronous event sensors to capture rich temporal cues, thereby enhancing the detection of moving objects [3]. Zhang et al. applied point cloud-multiple pixel fusion techniques to LiDAR data for 3D object detection. They achieved high accuracy with low latency, underscoring the approach's effectiveness in real-time applications [4]. To create a multi-model 3D object detection system, the authors integrated LiDAR point clouds with image pixels. They utilised two techniques: InverseAug, which reverses geometry-related augmentations for precise geometric alignment between LiDAR points and image pixels, and LearnableAlign, which uses cross-attention to dynamically identify relationships between image and LiDAR features [5].

An exhaustive review of recent advancements in driver drowsiness detection has been conducted, focusing on three main areas: classification methods, features, and datasets. Various metrics were used to evaluate the performance of different systems. The integration of 5G networks can potentially enhance the effectiveness of these detection systems [6]. Detecting driver drowsiness and fatigue using machine learning (ML) and deep learning (DL) techniques enables notifications to alert drivers when they are at risk of dozing off [7]. Many ML and DL approaches use physiological signals to detect drowsiness and achieve high detection accuracy [8]. EEG (electroencephalogram) and ECG (electrocardiogram) sensors are among the most commonly used signals. However, implementing these methods in real-time applications presents certain complexities [9]. Additionally, face and eye tracking techniques have been employed in driver drowsiness detection. Based on the driver's eye states, three key drowsiness features were extracted: the percentage of eyelid closure, blink frequency, and maximum closure duration of the eyes. These features served as inputs for various ML techniques, including K-nearest neighbours (KNN), support vector machines (SVM), logistic regression, and artificial neural networks (ANN). Among these techniques, KNN demonstrated the highest accuracy [10].

This study addresses two primary objectives. First, it aims to design a system capable of performing both functions on vehicles in near real-time. As a result, this system will enhance the efficiency and effectiveness of driver assistance systems. Second, the study seeks to implement and evaluate this dual-function within the specific context of Libyan road conditions. Through conducting experiments in this unique and challenging environment, the study intends to assess the system's performance and adaptability to local driving conditions. It will ultimately contribute to improving road safety and driver awareness.

## 2. Materials and Methods

The driver assistance system consists of two subsystems that currently work separately. The first subsystem is designed to alert drivers when they are fatigued, helping to prevent accidents. The second subsystem detects various objects on the road, including pedestrians, vehicles, bicycles, and trucks.

### 2.1 Driver Drowsiness Detection

This part of the system is divided into two stages: a training stage and a testing stage.

#### A. Training Stage

The dataset utilised in this model is an open-source dataset called *Drowsiness*, available on the Kaggle platform [11]. It was manually modified by removing two unnecessary classes, leaving only the two key classes: "Closed" (726 images) for closed eyes and "Open" (726 images) for open eyes. The dataset was then split, with 90% allocated for training and 10% for validation. The next step is image preprocessing, during which each image was adjusted before training. In this process, the images were resized to 256x256 pixels and converted to grayscale.

In this model, the architecture is sequential and consists of nine layers: three convolutional layers, three max-pooling layers, one flattening layer, and two dense layers, as shown in Figure 1. The training was conducted on a personal computer and it lasted for over an hour. The images were in grayscale. The hyperparameters used in this model are as follows:

- Batch size: 128
- Activation function: ReLU (Rectified Linear Unit)
- Loss function: Categorical Cross Entropy
- Optimizer: Adam (Adaptive Moment Estimation)
- Epochs: 50
- Activation function for the final layer: Sigmoid
- Padding: Same (with zeros added)

After completing all training epochs, the model is saved in H5 file format. This model is subsequently utilised to predict the state of the eye, either open or closed.

#### B. Testing Stage

In order to test the model in the real world, five volunteers were asked to record videos in their vehicles. They used a selfie mobile camera focused on the driver's face. The recording duration is about 30 seconds. The driver closed his eyes three times during the recording. This procedure was performed while the vehicle was parked to avoid any hazards or accidents. The collected data has several variations related to either camera specifications or the captured angle of the camera.

Since the model was trained on grayscale images, input images must be converted to grayscale for accurate predictions. This conversion is accomplished using a function from the OpenCV (Open-Source

Computer Vision) library. Haar feature-based cascade classifiers are used for object detection [12]. Although this approach originally designed for face detection, it can be adapted to detect a wide range of objects, including vehicles, buildings, and even specific items like kitchen utensils or fruit.

The algorithm can be divided into four key stages:

- Calculating Haar features
- Creating integral images
- Applying AdaBoost (Adaptive Boosting)
- Implementing cascading classifiers

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 32)	320
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 128, 128, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 128)	0
flatten (Flatten)	(None, 131072)	0
dense (Dense)	(None, 64)	8388672
dense_1 (Dense)	(None, 2)	130
-----		
Total params: 8,481,474		
Trainable params: 8,481,474		
Non-trainable params: 0		

Figure 1. Structure of the Sequential Model

In this paper, the Haar Cascade classifier is specifically used to detect faces and eyes in images. Once trained, the model is saved in the H5 format and later loaded using functions from the Keras library. When eyes are detected, they are input into the model, which makes a prediction using a sigmoid function. This function transforms the regression output into a decision boundary, making it suitable for binary classification (i.e., determining whether the eyes are open or closed). Based on the prediction, the model either triggers an alert or continues tracking as necessary.

## 2.2 Object Detection

This object detection is employed in this work to detect objects on roads. The dataset used for this model is the Udemy Self-Driving Car dataset, an open-source collection provided by Udemy [13]. For this study, minor adjustments were made to the dataset, including removing four classes and adding one new class. The final categories include:

- Pedestrians
- Cars
- Bikers
- Trucks
- Traffic Lights
- Traffic Light (Green)
- Traffic Light (Red)
- Stop Signs

The dataset consists of 9,570 images, divided into 8,133 for training, 1,245 for validation, and 192 for testing. Data preprocessing was performed using the Roboflow platform, where each image was resized to 416x416 pixels. Object annotations (bounding boxes) were applied to each object, assigning a corresponding class number. The processed images were then uploaded to the Ultralytics AI (artificial intelligence) tool to prepare for model training.

### A. Training Stage

YOLO (You Only Look Once) is a well-known object detection algorithm [14], recognized for its high speed and accuracy. YOLO is a single convolutional network that predicts the bounding boxes and the class probabilities for these boxes. The way YOLO works is by taking an image and splitting it into an  $S \times S$  grid, where many bounding boxes are identified within each grid cell. For each of these bounding boxes, the network outputs a class probability and offset values for the bounding box. The bounding boxes with class probabilities above a threshold value are selected. Therefore, the chosen bounding box will be utilised to locate the object within the image.

In object detection, objects can have different shapes and sizes, prompting the algorithm to generate multiple bounding boxes for each object. To ensure that only one bounding box remains for each object, Non-Maximum Suppression (NMS) is applied. NMS works by suppressing less likely bounding boxes and retaining only the most accurate ones. The process involves the following steps:

1. Select the bounding box with the highest objectiveness score.
2. Compare the overlap (Intersection over Union) of this box with the other bounding boxes.
3. Discard bounding boxes with an overlap (IoU) greater than 50%.
4. Move to the next highest objectiveness score.
5. Repeat steps 2-4 until all bounding boxes have been processed.

In this work, YOLOv5 version is used. Figure 2 illustrates the architecture of this convolution network. It consists of three parts

- Backbone: CSPDarknet:

While Darknet is originally the architecture of YOLO, CSPDarknet uses the Glued Spatial Pyramid method to improve recognition of objects at several scales. It contributes to increasing the speed and accuracy of YOLOv5.

- Neck: PANet:

Path Aggression Network enhances information flow across different levels of the network

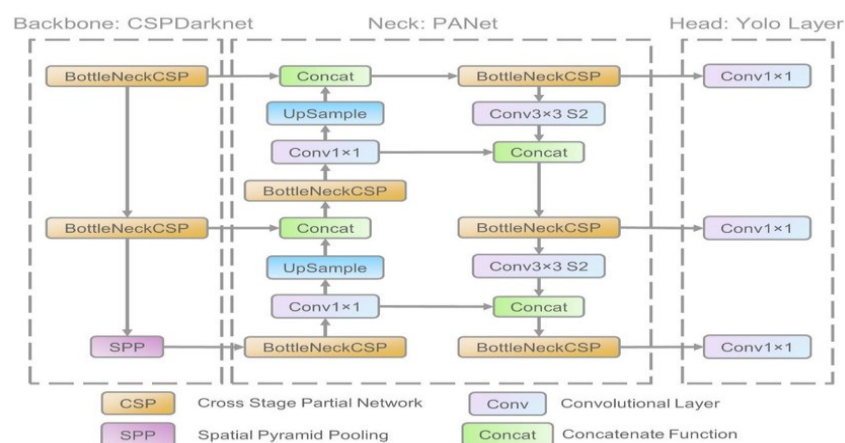


Figure 2. Architecture of YOLOv5 [15]

- Head: YOLO Layer.

The data are first input to CSPDarknet for feature extraction; the features are then fed to PANet for feature fusion. Finally, YOLO Layer outputs detection results (class, score, location, size) [15].

In this study, YOLOv5 was implemented using the PyTorch framework. After the dataset was uploaded to Ultralytics, several parameters were manually configured, including the batch size (determined by the server), the number of epochs (set to 150), and the image size (416 x 416 pixels).

The model architecture consists of 270 layers, with 1,774,741 parameters and 1,774,741 gradients. The model was initialized with pre-trained weights. In object detection, bounding boxes are used to mark the position of objects within an image. Each bounding box is defined by the coordinates of its upper-left and lower-right *corners* and includes five predictions:  $x$ ,  $y$ ,  $w$ ,  $h$ , and a confidence score. The  $x$  and  $y$  coordinates represent the centre of the bounding box relative to the grid cell, while width ( $w$ ) and height ( $h$ ) are relative to the entire image. The confidence score indicates the Intersection over Union (IoU) between the predicted and ground truth boxes. Each grid cell also predicts conditional class probabilities, which are calculated based on whether an object is present within the cell, regardless of the number of existing bounding boxes, only one set of class probabilities is predicted per grid cell.

### B. Testing Stage

The testing data consists of either a single image or a sequence of frames. To collect data for testing in the Libyan environment, a smartphone camera was mounted inside the vehicle. This camera was fixed to look down the street. It recorded various objects, including vehicles, pedestrians, traffic signs, and traffic lights. Ten videos were acquired under various road conditions such as coastal roads, highways, and the city centre.

Since the model was trained on 416x416 images, all input data used for real-time predictions must be resized to this dimension for consistency and accuracy. During training, the model's performance is continuously monitored, particularly in terms of accuracy, with weights (the model's trainable parameters) being updated throughout the process. As the training progresses through multiple epochs, these weights are refined to achieve optimal values. Consequently, this yields the highest possible accuracy.

## 3. Results and Discussion

The drowsiness detection model performed effectively, achieving a training accuracy of 95% and a validation accuracy of 93%. This indicates that the model generalises well to unseen data, with only a minor decrease in accuracy during validation. Figure 3 illustrates the training process, presenting loss and accuracy curves for both the training and validation datasets. The narrow difference between training and validation accuracies suggests that the model is not overfitting, ensuring reliable predictive

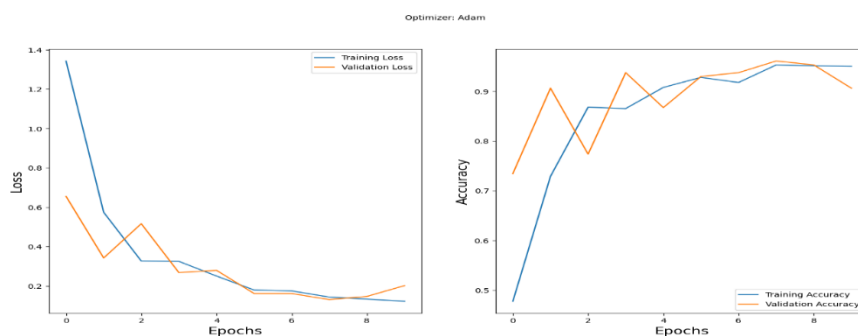


Figure 3. Loss and Accuracy Curves of the Drowsiness Detection Model

performance across various data subsets. The drowsiness detection model demonstrates strong efficiency on test images, even without employing transfer learning after the training phase. When the eyes are open, the model effectively detects the face and then accurately locates the eyes. It determines their states as open, as illustrated in Figure 4(a). Similarly, the model reliably identifies when the eyes are closed, as shown in Figure 4(b). This highlights the model's ability to analyse facial features and detect subtle changes in eye states, which is critical for accurately determining levels of drowsiness. Five video recordings were conducted for different individuals in their vehicles. The algorithm successfully classified the eye states in all the videos, except for three false positives in one video recording. Light variation may affect the performance of drowsiness detection. Adding more data in different lighting conditions to the training dataset, it will contribute for enhancing the accuracy of drowsiness detection.

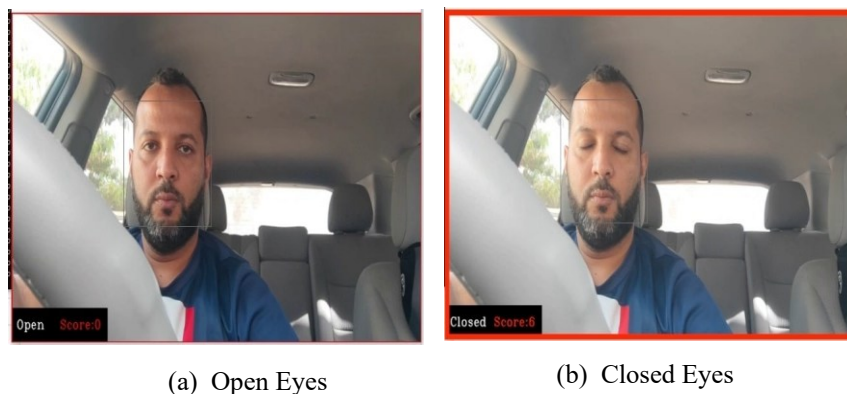


Figure 4. Drowsiness Detection Results

The object detection model achieved a mean average precision (mAP) of 0.69, a precision of 0.79, and a recall of 0.63 during the training phase, as illustrated in Figure 5. These metrics indicate that the model can accurately detect objects with a high degree of precision; however, there is room for improvement in recall, which measures how effectively the model identifies all relevant objects in the dataset.

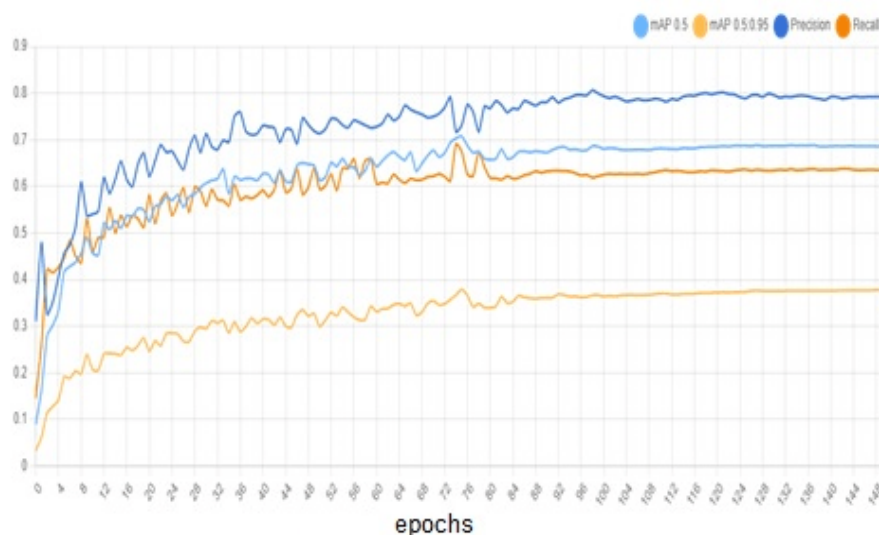


Figure 5. YOLO Model Training Results

A framework for drowsiness detection was proposed by Krishna et al. [16]. They employed YOLOv5 for face extraction, and the ViT framework (transformer model) was utilised for eye state classification. They obtained a training accuracy of 96.2% and a validation accuracy of 97.4% on the UTA-RLDD dataset. Their testing accuracy was 95.5%. Although they achieved slightly higher accuracy during the training stage. Their work was performed on a different dataset. The previous study involved 36 participants in the test dataset compared to only five participants in this paper.

Following the completion of the training phase, the model was evaluated in a real-world Libyan environment. A camera mounted on the vehicle captured live-streamed footage of the road during the experiments. The model successfully identified and drew bounding boxes around objects, using different colours for each box to distinguish between categories. Additionally, it provided confidence scores for the classified objects. As demonstrated in Figure 6, the object detection model accurately detected all objects on the scene, including vehicles, pedestrians, and traffic lights. To further assess its performance, eleven videos were recorded on different types of Libyan roads, including highways, main streets, secondary roads, and the city centre. Each video was approximately one minute in duration. The model demonstrated a high level of accuracy in predicting most objects within the videos, even in the presence of occlusions in certain frames.

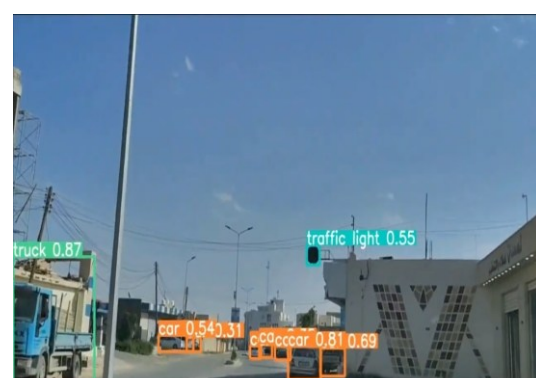


Figure 6. Object Detection Results

Figure 7 presents several unsuccessful results. In Figure 7(a), the classification model misclassified large vehicles as trucks due to their visual similarities. This issue can be addressed by incorporating more example images of such vehicles into the training dataset. Another error is depicted in Figure 7(b), where the model incorrectly classified a pharmacy light as a traffic light.



(a) Misclassification of Truck



(b) Misclassification of Traffic Light

Figure 7. Unsuccessful Results

Light variation also affects the object detection system due to its reliance on normal cameras. The system is currently designed to work during the daytime. To enable the object detection system to function at



night, depth information is needed. This information can be obtained through the use of a LiDAR (Light Detection and Ranging) sensor. Data fusion of depth and colour information is useful for improving the performance of object detection. Because the training dataset is not specific to Libyan road conditions. Object detection might face many challenges arising from road quality. For instance, the system might find it difficult to recognise some traffic signs. These traffic signs may not be clear either due to dust or because they have not been renewed for a long time. Additionally, there is variation in traffic signs according to different countries. Collecting more data specific to the Libyan environment will decrease the number of false detections.

YOLOv5 was used with a thermal camera for object detection by Farooq et al. [17]. After experimented with YOLOv5 using stochastic gradient descent (SGD) and Adam optimisation for training, they obtained a mean average precision (mAP) of 91.31% with the SGD optimiser. Precision was 75% with SGD, and recall was 93.76% with Adam. Vo and Quach [18] focused on object detection at nighttime. They used YOLOv5 with thermal camera as well. However, in addition to using SGD and Adam optimisation, different activation functions were applied. The mean average precision at IoU threshold of 0.5 (mAP 0.5) was 0.79. Precision and recall were 0.85 and 0.73, respectively. Although this study has lower results for these metrics compared to previously mentioned studies, its results are promising because it relies only on a normal smartphone camera. The data fusion between information from normal and thermal cameras will enhance the results of the object detection system.

Table 1 illustrates the system specifications used to train both the drowsiness detection model and the object detection model. These specifications include details about the hardware and software components, which are crucial in influencing the efficiency and performance of the training process.

Even though the drowsiness detection and object detection systems work independently, the outputs of both systems can be integrated to increase the safety of the driver. For instance, if the driver is drowsy, the drowsiness detection system will alert the driver. However, the driver may take some time to perceive the whole scene. As a result, this could lead to dangerous situations, especially if pedestrians are crossing the road. The object detection system can assist in such scenarios by detecting all objects, including vehicles and pedestrians until the driver regains full attention.

Table 1: System Specifications

Drivers	Desktop
Motherboard	Asus TUF B-450
Processor	AMD Ryzen 7 2700 3.5 GHz up to 4.1 GHz
RAM	16 GB 3000 MHz
GPU	Gigabyte RX-580 4GB
Windows	10

#### 4. Conclusions

This paper presents the development of two systems utilizing deep learning techniques for driver drowsiness detection and object identification, specifically tailored to Libyan road conditions within the domain of computer vision. The driver drowsiness detection system processes video input to evaluate the state of the driver's eyes, classifying them as either open or closed. This model estimates the driver's drowsiness level by applying convolutional neural networks (CNNs). In addition, road object detection

was performed using YOLOv5 (You Only Look Once version 5), a deep learning algorithm proficient at detecting multiple classes, including vehicles, pedestrians, traffic signs, and traffic lights. The YOLOv5 algorithm demonstrated high accuracy in object detection, achieving confidence levels above 90%. Transfer learning may further enhance the system's accuracy. Lidar sensors and thermal cameras can overcome issues related to lighting conditions. Due to the limited size of the dataset specific to Libyan roads, collecting more data and using a deeper convolutional network may improve the driver assistance system as a whole. Future research could incorporate features such as lane tracking while focusing on optimising both the speed and accuracy of the implemented models.

## REFERENCES

- [1] R. Khoshkangini, P. Mashhadi, D. Tegnered, J. Lundström, and T. Rögnavaldsson, "Predicting vehicle behavior using multi-task ensemble learning," *Expert Systems with Applications*, vol. 212, p. 118716, 2023..
- [2] B. J. Tanu, M. S. Ahmed, M. A. Islam, and B. J. Ferdosi, "A comparative study of recent practices and technologies in advanced driver assistance systems," *International Journal of Computer Applications*, vol. 975, pp. 8887, 2024.
- [3] Z. Zhou, Z. Wu, R. Boutteau, F. Yang, C. Demonceaux, and D. Ginjac, "RGB-event fusion for moving object detection in autonomous driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7808-7815, May 2023.
- [4] Y. Zhang, K. Liu, H. Bao, Y. Zheng, and Y. Yang, "PMPF: Point-cloud multiple-pixel fusion-based 3D object detection for autonomous driving," *Remote Sensing*, vol. 15, no. 6, p. 1580, 2023.
- [5] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, and M. Tan, "DeepFusion: Lidar-camera deep fusion for multi-modal 3D object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17182-17191, 2022.
- [6] Y. Albadawi, M. Takruri, and M. Awad, "A review of recent developments in driver drowsiness detection systems," *Sensors*, vol. 22, no. 5, p. 2069, 2022.
- [7] S. A. El-Nabi, W. El-Shafai, E. S. M. El-Rabaie, K. F. Ramadan, F. E. Abd El-Samie, and S. Mohsen, "Machine learning and deep learning techniques for driver fatigue and drowsiness detection: A review," *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 9441-9477, 2024.
- [8] M. Doudou, A. Bouabdallah, and V. Berge-Cherfaoui, "Driver drowsiness measurement technologies: Current research, market solutions, and challenges," *International Journal of Intelligent Transportation Systems Research*, vol. 18, pp. 297-319, 2020.
- [9] A. A. Saleem, H. U. R. Siddiqui, M. A. Raza, F. Rustam, S. Dudley, and I. Ashraf, "A systematic review of physiological signals based driver drowsiness detection systems," *Cognitive Neurodynamics*, vol. 17, no. 5, pp. 1229-1259, 2023.
- [10] A. A. Bamidele, K. Kamardin, N. S. N. Abd Aziz, S. M. Sam, I. S. Ahmed, A. Azizan, and H. M. Kaidi, "Non-intrusive driver drowsiness detection based on face and eye tracking," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, 2019.
- [11] R. Fusek, "Pupil localization using geodesic distance," in *Advances in Visual Computing: 13th International Symposium, ISVC 2018, Las Vegas, NV, USA, November 19–21, 2018, Proceedings 13*, Springer International Publishing, pp. 433-444, 2018.
- [12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. I-I, Dec. 2001.
- [13] R. Slim, A. Sharaf, J. Slim, and S. Tanveer, *The Complete Self-Driving Car Course - Applied Deep Learning*. Udemy, Sep. 2020. [Online]. Available: <https://www.udemy.com/course/applied-deep-learningtm-the-complete-self-driving-car-course/> (accessed on 5 09 2023).
- [14] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [15] G. Jocher, "YOLOv5 by Ultralytics," [Online]. Available: <https://github.com/ultralytics/yolov5>. Accessed: Feb. 30, 2023.
- [16] Krishna, Ghanta Sai, Kundrapu Supriya, and Jai Vardhan. "Vision transformers and YoloV5 based driver drowsiness detection framework." *arXiv preprint arXiv:2209.01401*. (2022).
- [17] M. A. Farooq, P. Corcoran, C. Rotariu, and W. Shariff, "Object detection in thermal spectrum for advanced driver-assistance systems (ADAS)," *\*IEEE Access\**, vol. 9, pp. 156465-15648, 2021.
- [18] H. T. Vo and L. D. Quach, "Advanced Night time Object Detection in Driver-Assistance Systems using Thermal Vision and YOLOv5," *\*International Journal of Advanced Computer Science and Applications\**, vol. 14, no. 6, 2023.