iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** II **Month of publication:** February 2025

DOI: https://doi.org/10.22214/ijraset.2025.66717

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

Drowsiness Detection Using CNN-LSTM

Vijayalakshmi S [1], Elavendhan K.P [2], Suriya E [3], Harrisasthaa G [4]
Department of Computer Science Engineering, PSG College of Technology

Abstract: ***Driver fatigue is a major reason to road accidents, prompting the need for systems which can detect drowsiness in real time and prevent potential mishaps. It presents a deep learning-based solution utilizing CNN and LSTM networks to accurately find drowsiness based on the video inputs of a driver's face. By analyzing eye and facial expressions, it identifies early signs of fatigue then triggers an alert. The model, trained on a labeled dataset which contains drowsy and non-drowsy drivers, achieves an accuracy of 97.83%. Real-time video processing is powered by OpenCV and advanced facial recognition techniques. The system is implemented on a Raspberry Pi 3, which runs the detection code and displays an alert on an external display board mounted outside the vehicle. When drowsiness is detected, an alert symbol is shown to notify nearby vehicles and reduce potential accidents. This solution demonstrates the potential of combining CNNs for feature extraction part and LSTMs work is to find temporal pattern analysis to enhance road safety and mitigate fatigue related accidents.***

Keywords: Convolutional Neural Networks, Real-time Monitoring, Long Short-Term Memory, Raspberry Pi Implementation, Alert System.

I. INTRODUCTION

Deep learning has recently emerged as the go-to method for intelligent diagnostics of rotating machinery, and it has yielded a number of triumphs. It allows for building more agile model, and precise accuracy at the trade-off of a reduced amount of severe regularization. To present a benchmark study, the author in [2] used seven datasets to evaluate Driver fatigue and studies estimating that driver fatigue leads to a significant percentage of road accidents globally. Several approaches is developed to address this issue, each with its own benefits and limitations. Physiological-based systems, on the other hand, monitor biological signals like heart rate, skin conductivity, and brain activity to detect signs of fatigue. Behavioral-based systems focus on employing features like Eye Aspect Ratio (EAR), yawning detection, and head pose estimation to identify drowsiness. To overcome the limitations of individual methods, hybrid systems combine multiple approaches, such as integrating vehicle dynamics with facial behavior analysis, which improves accuracy and reduces false positives and negatives. However, these systems often require higher computational power, making them less suitable for environments. Popular technologies like have further enhanced drowsiness detection. CNNs excel at extracting spatial features from facial images, while LSTM networks are adept at analyzing temporal patterns, enabling the detection of fatigue-related behaviors over time.

II. OBJECTIVE

The objective of this paper is to design an efficient and reliable real-time Driver Drowsiness Detection and Alert System. Leveraging advanced deep learning techniques, the system aims to detect early signs of fatigue through facial analysis, including prolonged eye closure, yawning, and head movements. The project focuses on ensuring early drowsiness detection and delivering timely alerts to prevent accidents effectively.

III.LITERATURE REVIEW

Early studies on driver drowsiness detection concentrated on visual feature extraction using facial landmarks, eye aspect ratio, and head pose for real-time monitoring. Techniques like random forests, sequential neural networks, and SVM classifiers have achieved high accuracies, with results reaching up to 99% when applied to datasets like the NTHU driver drowsiness detection dataset [1]. The integration of machine learning approaches with physical sensors, such as MQ-3 and Pi cameras, expanded detection capabilities. These systems utilized LSTM-based approaches alongside face and mouth landmark detection, providing robust drowsiness alerts and improved accuracy [2]. Similarly, methodologies employing convolutional neural networks and transfer learning demonstrated effectiveness in extracting relevant features from pre-trained models, achieving accuracies of over 96% through optimized fine-tuning processes [3,17]. Other research focused on capturing real-time driver facial expressions, calculating metrics like Eye Aspect Ratio and Mouth Opening Ratio. These methods employed machine learning algorithms like Support Vector Machines to classify drowsy versus non-drowsy states, reaching sensitivities of 95.58% and specificity of 100% [4,16].

Deep neural network architectures combined with advanced algorithms like squeeze-and-excitation networks revealed the importance of critical facial landmarks such as those around the nose and mouth. These models exhibited cross-database generalizability, enhancing accuracy and usability in diverse conditions [5,17]. Technologies incorporating Raspberry Pi and GPS modules further enhanced real-time detection by analyzing microsleep patterns, even in challenging conditions such as dim lighting or the use of spectacles. Accuracy levels reached 90%, demonstrating practical applications for transportation safety [6]. Additional developments like the DriCare system introduced innovative facial region-tracking algorithms, achieving around 92% accuracy by combining features of eye closure duration and yawning frequency to provide timely fatigue warnings [7]. The proposed system enables effective real-time drowsiness detection, validated, the system offers a practical approach to reducing drowsiness-related accidents with real-time alerts [11]. The optimal EAR limit provided the best balance of detection performance and accuracy. This technique enhances blink detection systems, essential for driver monitoring in real-world conditions [12,18]. The use of PCA and SVM for face detection and recognition has been shown to complement drowsiness detection systems reliant on facial feature analysis [20]. Additional research found that increasing temperatures and varying precipitation patterns are likely to alter fire regimes significantly. Calls for adaptive management strategies in agriculture to cope with changing fire regimes due to climate change [13]. The approach demonstrated reliable detection capabilities and high accuracy in classifying drivers as drowsy or non-drowsy. Offers scalable and effective detection models, potentially deployable for real-world driver safety systems [14,18]. Achieved effective detection of transitions from wakefulness to drowsiness under various conditions in simulated environments. The hybrid approach improves detection accuracy and usability across diverse scenarios, addressing limitations of single-measure methods [15].

IV. METHODOLOGY

A. Long Short-Term Memory

This specialized type of network is designed to process sequential and time-dependent data. Unlike standard RNNs, LSTMs address the vanishing gradient problem, allowing them to effectively capture and retain long-term dependencies. They are composed of memory cells and three primary gates:

Gate 1: finds out what new things to append to it.

Gate 2: Determines what information should be removed in cell's memory.

Gate 3: Manages the cell's output by considering its current state.

In driver drowsiness detection, LSTMs analyze temporal patterns in video sequences, such as prolonged eye closure or repeated yawning. By processing sequences of feature maps extracted by CNNs, LSTMs can predict whether a driver is exhibiting signs of drowsiness based on temporal dependencies.

B. Convolutional Neural Network

CNNs are used for analyzing visual data and are highly effective in tasks such as image recognition, object detection, and facial analysis. Their strength lies in their ability to automatically and adaptively learn spatial hierarchies of features from images. CNN consists of two main types of layers: Layer 1: These layers apply convolutional operations to extract various features. The kernel (or filter) slides over the image, generating feature maps.

Layer 2: The hybrid approach boosts detection accuracy and usability in different situations by overcoming the weaknesses of methods that use only one measure.

Their ability to handle varying facial expressions and environmental conditions makes them ideal for this application. Together, CNNs and LSTMs form a powerful hybrid architecture, leveraging spatial and temporal information for accurate driver drowsiness detection.

V. DESCRIPTION OF THE PROCESS

The driver drowsiness detection system follows a systematic workflow aimed at continuous improvement and process evaluation. Training is conducted on a labeled dataset comprising driver images and videos, annotated with drowsy and non-drowsy states.

A. Data Preprocessing

Before training, the dataset undergoes preprocessing to ensure consistency and reduce noise. Key steps include:

Resizing Images: All video frames are resized to a uniform dimension to satisfy the model's size specifications.

Normalization: Pixel values are scaled to a range between 0 and 1 to speed up model convergence.

Data Augmentation: Techniques such as flipping and rotation adjustments are applied to create variations in the dataset, enhancing the model's ability to generalize and perform better on unseen data.

B. Model

The hybrid model comprises two stages:

CNN Stage: The CNN extracts spatial features, including eye closure and yawning patterns, from each frame. Multiple convolutional and pooling layers ensure that both basic and advanced features are captured.

LSTM Stage: The LSTM processes sequences of feature maps from the CNN to identify temporal dependencies, then prolonged eye closure or repeated yawns, indicative of drowsy states.

C. Training Configuration

Loss Function: The categorical cross-entropy loss function quantifies the difference between the predicted and actual labels, measuring how well the model's predictions align with the true classifications.

Optimizer: Adam optimizer is chosen for its adaptive learning rate and efficient convergence, making it well-suited for training deep learning models.

Learning Rate Scheduling: This scheduler minimizes learning rate during training when the performance plateaus, ensuring fine-tuning of parameters.

D. Training Process

Epochs: It is trained for multiple epochs, where every epoch consists of a forward and a backward pass through the network.

Batching: The dataset is split into mini-batches, enabling iterative weight updates and faster stabilization.

Validation: A portion of the dataset is made for validation, enabling monitoring of the performance on unseen data during training period.

Checkpointing: Model parameters are saved at the end of each epoch to ensure recovery when some problems and lets selecting the best-performing model.

E. Overfitting Prevention

Regularization techniques like dropout layers and data augmentation, are employed to mitigate overfitting. Early stopping is implemented, terminating process of training once the validation performance ceases to improve.

To further improve accuracy, ensemble methods can be applied to combine predictions from multiple models, leveraging their complementary strengths to enhance overall performance.

VI. IMPLEMENTATION

A. Dataset Description

The dataset used in this project was specifically created to meet the unique needs of detecting drowsiness in drivers.The dataset comprises a large number of video frames extracted from driver monitoring systems, categorized into three distinct classes: yawning, eye closure, and normal behavior.

This custom dataset is essential for accurately training the model to recognize these crucial states that can indicate potential drowsiness or distraction.

Each frame in the dataset is manually labelled based on the driver's state, which is key for supervised learning tasks like this. The frames were classified into the following three categories:

1) *Yawning frames:* These frames capture drivers exhibiting yawning, which is a strong indicator of drowsiness. The model learns to identify key features of yawning, such as the wide opening of the mouth and related facial expressions.
2) *Eye closure frames:* In these frames, the driver's eyes are either fully or partially closed. Extended eye feature is a key for drowsiness, and the model is trained to identify this behavior in order to prevent micro-sleeps or full drowsiness.
3) *Normal frames:*These frames depict a driver who is alert, with open eyes and focused attention. They are essential for the model to establish a baseline of normal, attentive behavior, enabling it to differentiate between drowsy or distracted states

B. Model Training

This section provides the sequence of process used to train the CNN-LSTM model, including data preparation, network architecture, weight initialization, forward and backward propagation, loss calculation, optimization, and model saving.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

C. Train-Test Split

The dataset, which consists of pre-processed videos, is divided into two parts:

Training Dataset (70%): 70% of the data is used for model training. This split guarantees that the model has sufficient data for learning the underlying patterns in the video, thus improving its accuracy.

Test Dataset (30%): For evaluating model performance we use the rest 30% of data. This portion is not utilized during training but plays a critical role in evaluating the new data.

D. Model Initialization

It is employed in this project is a hybrid CNN-LSTM architecture, specifically designed to process video data by capturing both spatial and temporal patterns.

1) CNN Layers

- *First Convolutional Layer:* It contains 32 filters, each with a 3x3 kernel size, and employs a rectified linear unit (ReLU) activation function. This layer captures low-level features, such as edges, from the input.
- *Second Convolutional Layer:* It increases the filter to 64, reuses a kernel with size of 3x3, and then performs the ReLU operation. This layer captures more complex features by imaging larger patterns in video images.
- *Max Pooling Layer:* After convolution operation, this layer with kernel size 2x2 is used. this reduces the spatial dimension of feature map, reduces computational load and extracts the most important features from each region.
- *Dropout Layer:* This is embedded after CNN layer to reduce the probability of overfitting. During the training process, it randomly disables some of the input units as a normalization .

2) LSTM Layers

- *LSTM :* The previous layer , with a feature size of 64, is reshaped to be compatible with the LSTM network.this layer, which has a hidden size of 128, is then applied to model temporal dependencies in the video data. It is particularly effective in capturing sequential patterns over time, which is needed for a video-based applications.
- *Dropout Layer:* An additional dropout layer, with a dropout rate of approximately 0.5, is applied after the LSTM layer to further minimize the risk of overfitting

3) Dense (Fully Connected) Layers

- *Fully Connected Layer (fc1):* The LSTM output is fed to the connected process with 64 neurons and ReLU processing. This process is based on a combination of components, including information extracted from the source and body of the video file.
- *Output Process(fc2):* The final connection process uses the softmax activation function to output useful classes. Each output neuron represents a latent class (drowsy or not), and the softmax function guarantees that the sum of the outputs equals 1, making it appropriate for multi-class classification

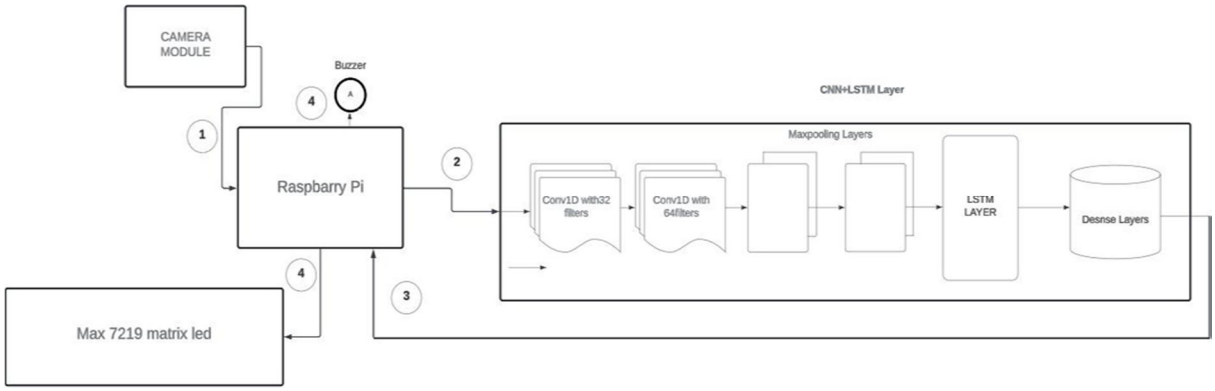


Fig. 1 Proposed Model

The proposed model architecture is shown in Figure 1 includes multiple cnn layers for extracting features, then LSTM layers to analyze time-dependent features. Dropout layers are incorporated to prevent overfitting.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

E. Weight Initialization:

Efficient weight initialization strategies are used to optimize model learning. He Initialization is applied to convolutional layers to maintain activation variance and prevent vanishing gradients with ReLU activations. Xavier Initialization ensures balanced gradients in fully connected layers, aiding faster convergence. LSTM Initialization optimizes learning efficiency by capturing temporal dependencies in sequential data.

F. Forward Pass:

The forward pass outlines the process by which input data travels through the layers of the model:
The input flows through the layers during this stage. Weights in convolutional layers use He Initialization, fully connected layers use Xavier Initialization, and LSTM layers employ customized initialization to optimize performance. The LSTM output feeds into fully connected layers, which use softmax activation to produce class probabilities

G. Training Loop:

The model undergoes training across multiple epochs, with each epoch comprising several iterations:

1) *Loss of cross entropy calculation*: For multiple classses, it is computed between the predicted and the actual class labels. This loss function measures how well the it is align with the true classes.
2) *Focal Loss:* Given the potential imbalance in the dataset, where non-drowsy samples may outnumber drowsy ones, focal loss is employed to place greater emphasis on difficult-to-classify instances, thereby minimizing the impact of easily classified examples on the loss computation.
3) *Backpropagation:* The computed loss is sent back through the network, adjusting the weights to reduce the loss.
4) *Gradient Descent:* During backpropagation, the optimizer (Adam) updates the model's weights by computing gradients for each batch. The gradients are cleared at the start of each batch to ensure proper weight updates.

H. Loss Functions and Optimizers:

Cross-Entropy Loss: It is used to measure the error between the predicted and the actual labels. For multi-class classification tasks, cross-entropy loss provides a way to penalize incorrect predictions and reward correct ones. It penalizes heavily when it assigns high probabilities to the wrong classes.

$$\text{Cross-Entropy} = L(y, t) = -\sum^{n} \ln y_i \quad (1)$$

.

Adam Optimizer: The Adam optimizer integrates momentum for more stable updates and adaptive learning rates for effective weight adjustments, making it well-suited for complex models such as CNN-LSTM.The optimizer ensures faster and more stable training by efficiently navigating the loss landscape.

$$nt = \beta_1 * n_{t-1} + (1 - \beta_1) * \nabla y_t \quad (2)$$

$$ct = \beta_2 * c_{t-1} + (1 - \beta_2) * (\nabla n_t)^2 \quad (3)$$

$$dt+1 = dt - n / ct + e * nt \quad (4)$$

$$bt = bt / 1-nt \quad (5)$$

$$ct = ct / 1-nt \quad (6)$$

I. Facial Landmark Features

The system's Drowsiness Detection Algorithm employs a machine learning model that analyzes critical drowsiness indicators using facial landmark features, as depicted in Figure 2. These indicators include:

1) EAR: Measures the ratio of eye height to width, used to monitor eye closure. A significant drop in EAR indicates prolonged eye closure, which can be a sign of drowsiness.
2) LEAR and REAR (Left Eye and Right Eye Aspect Ratios): These individually track the aspect ratios for each eye to provide a more granular detection of eye closure and asymmetry in eye movement,improving detection accuracy.
3) MAR (Mouth Aspect Ratio): Measures the ratio of the mouth's height to width. Frequent yawning, indicated by a high MAR, is a strong indicator of fatigue.

4) Intersection over Union (IoU): Used to detect head position by evaluating the overlap of bounding boxes of key facial landmarks over time. Significant changes in IoU values can indicate head tilting or nodding, which is often associated with drowsiness. By integrating these indicators, the system can efficiently track and identify driver fatigue in real time, providing precise and prompt alerts.

$$ear = (v1 + v2 + v3) / (3.0 * h) \quad (7)$$

$$mar = (v1 + v2 + v3) / (3.0 * h) \quad (8)$$

```
# Vertical distances
   v1 = euclidean_distance(land_marks[Eyepoints[01]], landmarks[eye_points[11]])
   v2 = euclidean_distance(land_marksEeyepoints[21]], landmarks[eye_points[31]])
   v3 = euclidean_distance(land_marks[Eyepoints[41]], landmarks[eye_points[51]])
# Horizontal distance
   h = euclidean_distance(landmarks[eye_points[6]], landmarks[eye_points[7]])
```

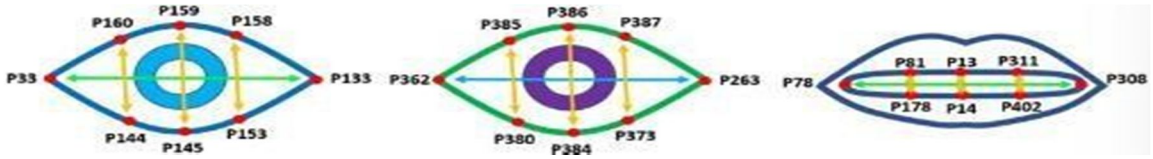


Fig. 2 EAR and MAR Identification Points

J. Hardware

The system uses a Raspberry Pi 3 microcontroller to manage external alerts via a buzzer and a MAX7219 LED Matrix Display, with the components summarized as follows:

Raspberry Pi 3: Acts as the central microcontroller, receiving serial signals from the CNN-LSTM software to control the buzzer and LED matrix.

Buzzer: It emits audio signals to alert the driver and capture their attention when drowsiness is identified

MAX7219 LED Matrix Display: Provides visual alerts with messages like “Drowsy” or “Wake Up” to reinforce the audio signals.

Circuit Configuration:

The LED matrix uses SPI protocol for efficient data transmission, while the buzzer is controlled via digital pin 8 for simple on/off signaling.

Hardware-Software Integration:

The CNN-LSTM model processes video input to detect drowsiness. Upon detection, it sends a serial signal to the Raspberry Pi, which activates the audio and visual alert systems which is shown in Figure 3.

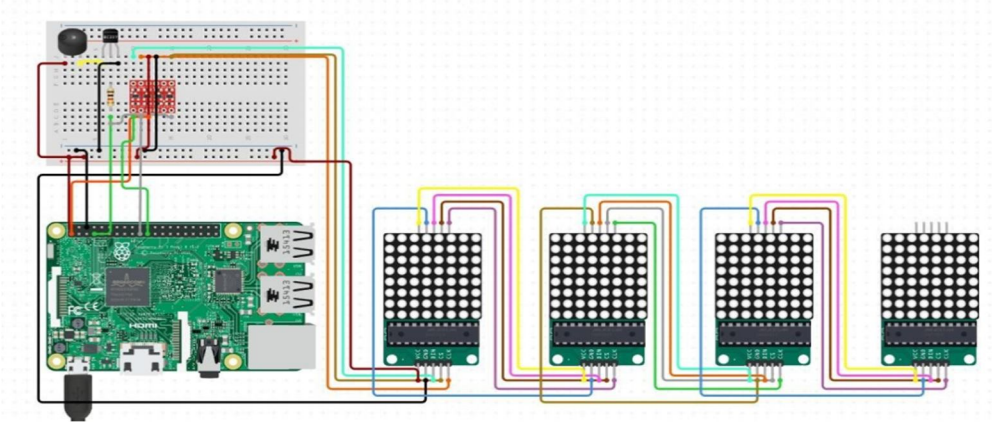


Fig. 3 Hardware Configuration

VII. EXPERIMENTAL RESULTS AND DISCUSSIONS

This experiment aims to assess the effectiveness of our techniques. It focuses on identifying early signs of driver drowsiness by finding facial features, including eye balls and head position, from live video feeds. The ultimate goal is to minimize traffic accidents by delivering timely warnings upon detecting fatigue.

Accuracy Graph: Defines the accuracy of our model during training and validation. It indicates how well the model is learning over time in the Figure 4.

Loss Graph: The loss graph monitors the loss, which measures the disparity between predicted and actual values. A lower loss indicates improved model performance, as illustrated in Figure 5.

Confusion Matrix: Displays correct and incorrect classifications. It helps evaluate the distinguishes between sleepy and non-sleepy states in Figure 6.

Learning Rate: Figure 7 illustrates the variation in learning rate during training. An optimal learning rate facilitates faster and more efficient model convergence.

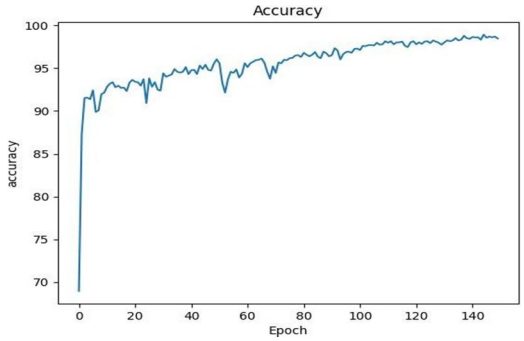


Fig. 4 Accuracy

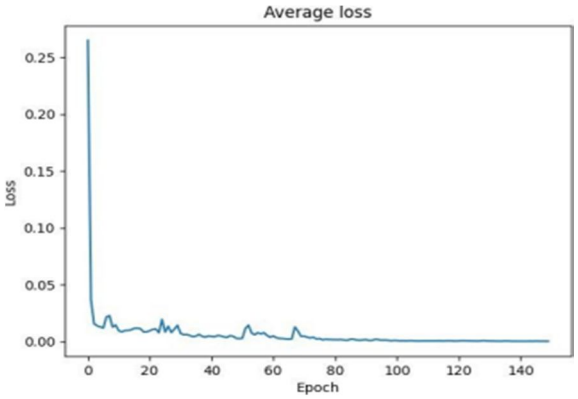


Fig. 5 Average Loss

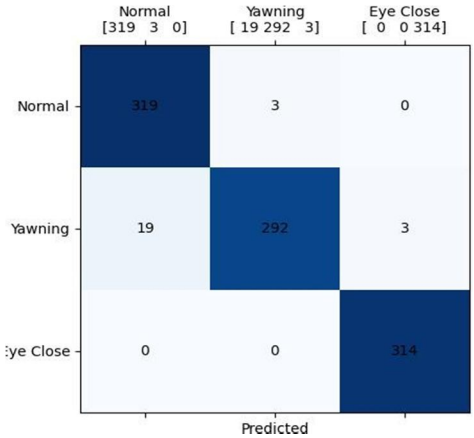


Fig. 6 Confusion Matrix

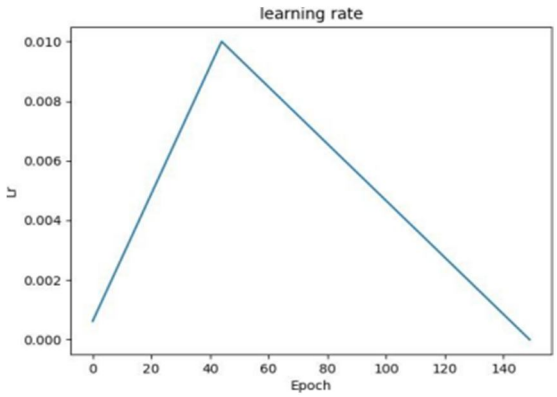


Fig. 7 Learning Rate

VIII. MODEL PERFORMANCE EVALUATION

A. Epochs

Initially, it was trained with 20 epochs, achieving stable performance with an accuracy of 95.65%. It effectively differentiated between drowsy and non-drowsy states. When training was extended to 150 epochs, the model's accuracy improved to 97.83%, with a better balance between precision and recall, indicating enhanced learning and more accurate predictions over the additional epochs. The analysis evaluates the performance metrics of models trained for 20 and 150 epochs, offering insights into their effectiveness based on accuracy, F1 score, precision, and support are represented in table 1.

Table 1. Comparison of metrics side by side for epoch 20 vs epoch 150.

	precision		recall		f1-score		support	
	Epoch (20)	Epoch (150)	Epoch (20)	Epoch (150)	Epoch (20)	Epoch (150)	Epoch (20)	Epoch (150)
Normal	0.91	0.94	0.95	0.99	0.93	0.97	319	322
yawning	0.94	0.99	0.89	0.93	0.92	0.96	302	314
Eye Close	0.98	0.99	0.98	1.00	0.98	1.00	329	314
Accuracy	NA	NA	NA	NA	0.95	0.97	950	950
Macro avg	0.95	0.97	0.94	0.97	0.94	0.97	950	950
Weight avg	0.95	0.97	0.95	0.97	0.95	0.97	950	950

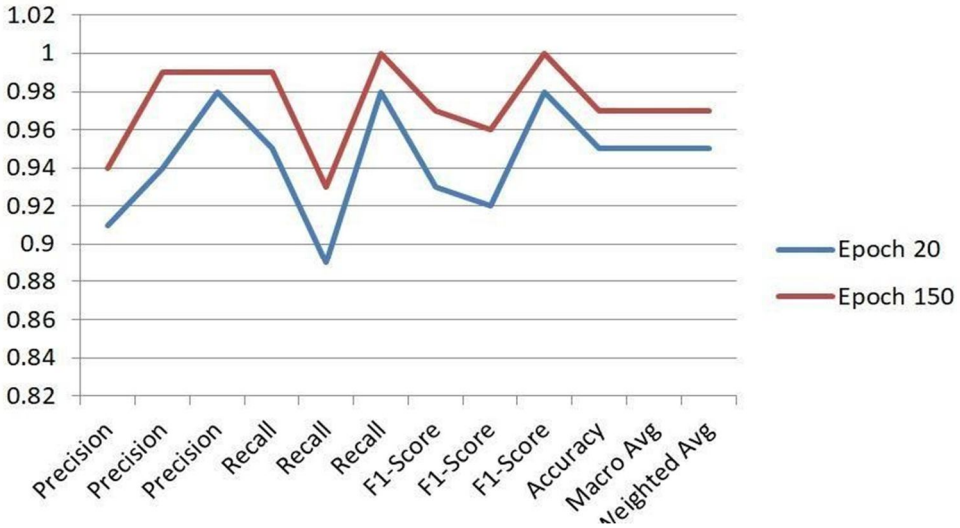


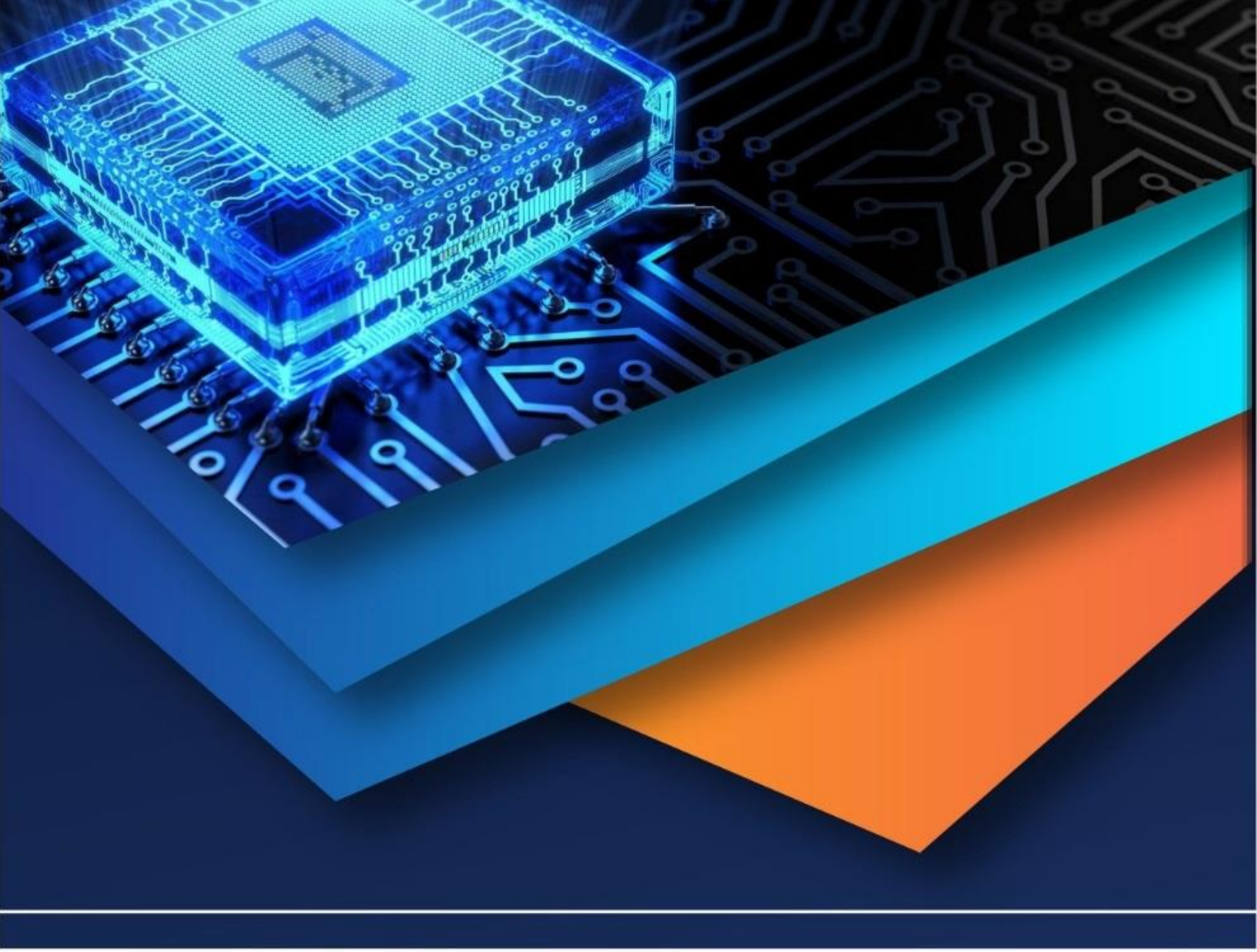
Fig. 8 Accuracy Comparison

IX. CONCLUSION

The proposed drowsiness detection and alert system integrates advanced machine learning techniques with hardware components to enhance driver safety. By leveraging a CNN-LSTM model, the system effectively analyzes real-time video feeds to detect signs of fatigue and distraction. The incorporation of hardware components like the Raspberry Pi 3, MAX7219 LED Matrix Display, and buzzer enables prompt and efficient audio-visual alerts, creating a dependable warning mechanism for drivers. This combination of advanced software processing and optimized hardware integration delivers a reliable and adaptable solution for avoiding any issues.

REFERENCES

[1] Florez, R., Palomino-Quispe, F., Coaquira-Castillo, R.J., Herrera-Levano, J.C., Paixão, T., & Alvarez, A.B. (2023). "A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification." Applied Sciences, Vol. 13, No. 4, pp. 1-18.
[2] Salem, D., & Waleed, M. (2024). "Drowsiness Detection in Real-Time via Convolutional Neural Networks and Transfer Learning." Journal of Engineering and Applied Science, Vol. 71, No. 5, pp. 1-15.
[3] Liu, M.Z., Xu, X., Hu, J., & Jiang, Q.N. (2021). "Real-time detection of driver fatigue based on CNN-LSTM." IET Image Processing, Vol. 16, No. 4, pp. 1-8.
[4] Gomaa, M.W., Mahmoud, R.O., & Sarhan, A.M. (2022). "A CNN-LSTM-Based Deep Learning Approach for Driver Drowsiness Prediction." Journal of Engineering Research (ERJ), Vol. 6, No. 3, pp. 59-71.
[5] Quddus, A., Zandi, A.S., Prest, L., & Comeau, F.J.E. (2022). "Using long short-term memory and convolutional neural networks for driver drowsiness detection." International Journal of Automotive Technology and Management, Vol. 61, No. 5, pp. 1-14.
[6] Komal, P., Sharma, A., Lamba, B., Nagpal, S., & Chauhan, S. (2020). "Driver Drowsiness Detection System using Convolutional Neural Network." International Journal of Advanced Research in Computer and Communication Engineering, Vol. 9, No. 5, pp. 90-97.
[7] Agarwal, S. & Nachappa, M.N. (2022). "Driver Drowsiness Detection System Using CNN." International Journal of Science, Engineering and Technology, Vol. 10, No. 2, pp. 1-6.
[8] Walizad, M.E., Hurroo, M., & Sethia, D. (2022). "Driver drowsiness detection system using convolutional neural network." International Conference on Trends in Electronics and Informatics (ICOEI), pp. 14-20.
[9] Jackson, M.L., Raj, S., Croft, R.J., Hayley, A.C., Downey, L.A., Kennedy, G.A., & Howard, M.E. (2015). "Slow eyelid closure as a measure of driver drowsiness and its relationship to performance." Attention, Perception, & Psychophysics, Vol. 41, No. 7, pp. 1-15.
[10] Safarov, F., Akhmedov, F., Abdusalomov, A.B., Nasimov, R., & Cho, Y.I. (2023). "Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer Vision and Eye-Blink Analyses for Enhanced Road Safety." Sensors, Vol. 23, No. 11, pp. 18-27.
[11] Albadawi, Y., AlRedhaei, A., & Takruri, M. (2023). "Real-Time Machine Learning-Based Driver Drowsiness Detection Using Visual Features." Journal of Imaging, Vol. 9, No. 5,pp. 45-53.
[12] Dewi, C., Chen, R.-C., Chang, C.-W., Wu, S.-H., Jiang, X., & Yu, H. (2022). "Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety." Electronics, Vol. 11, No. 19, pp. 83-92.
[13] Singh, J., Kanojia, R., Singh, R., Bansal, R., & Bansal, S. (2022). "Driver Drowsiness Detection System – An Approach By Machine Learning Application." PeerNet Research, Vol. 62, No. 5, pp. 1-9.
[14] Rahman, A., Sirshar, M., & Khan, A. (2015). "Real time drowsiness detection using eye blink monitoring." In National Software Engineering Conference (NSEC), pp. 1-5.
[15] Bajaj, J.S., Kumar, N., Kaushal, R.K., Gururaj, H.L., Flammini, F., & Natarajan, R. (2023). "System and Method for Driver Drowsiness Detection Using Behavioral and Sensor-Based Physiological Measures." Sensors, Vol. 23, No. 3, pp. 292-300.
[16] Pandey, P., Sharma, M., Saxena, P., & Dwivedi, R.K. (2023). "Driver Drowsiness Monitoring and Detection using Machine Learning." International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp. 1-6.
[17] Ayub, M., Al-Zahrani, H.S.A., & Alzahrani, F.M. (2023). "Smart Drowsiness Detection System Using a Hybrid Approach." IEEE Access, Vol. 11, No. 7 pp. 854-864.
[18] Bhatia, A.K., & Awasthi, G.L. (2013). "Driver Drowsiness Detection: A Review." Transportation Research Part F: Traffic Psychology and Behaviour, Vol. 20, No. 3, pp. 160-170.
[19] Li, Y., Zhang, S., Zhu, G., Huang, Z., Wang, R., Duan, X., & Wang, Z. (2023). "ACNN-Based Wearable System for Driver Drowsiness Detection." Sensors, Vol. 31, No. 5, pp. 1-15.
[20] S Vijayalakshmi, JU Maheswari, K Jananiyie (2023). "Face Detection and Recognition using Machine Learning Techniques." Journal of Innovative Image Processing, vol.4, issue no 4, page 316-327



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY