Efficient Driver Distraction Detection Using HOG Features and Learnable Pooling Attention Module

1st Advised by: Dr. Shamsad Parvin
University at Buffalo, SUNY

2nd Shweta Ganesh Bankar
University at Buffalo, SUNY

***Abstract*—Road accidents persist as a significant global concern despite ongoing enhancements in vehicular safety technologies, including Advanced Driver Assistance Systems (ADAS). A significant fraction of these accidents arise from driver distraction activities that divert attention away from the task of driving. According to the National Highway Traffic Safety Administration (NHTSA), driver distraction accounts for roughly one-tenth of road accidents in the United States, contributing to more than 3,000 fatalities annually. This paper introduces HOG-LPNet, a novel distracted driver detection model that balances classification accuracy with computational efficiency. Our approach utilizes a lightweight Histogram of Oriented Gradients (HOG) pre-processing technique, which is chosen for its rapid image conversion but at the expense of producing feature maps that are somewhat less distinctive. To mitigate this, we integrate contrast-limited adaptive histogram equalization (CLAHE) for contrast enhancement. We propose a Learnable Pooling Attention Module, adapted from the convolutional block attention module (CBAM) and streamlined to exclude spatial attention. This attention module leverages learnable and global max pooling alongside SiLU-activated fully connected layers to selectively amplify subtle yet discriminative channels in the HOG-transformed feature maps. On the State Farm Dataset, HOG-LPNet achieves 99.52% accuracy, situating it competitively among current pioneering approaches. By coupling fast, edge-oriented pre-processing with a targeted Learnable Pooling Attention strategy, HOG-LPNet offers a robust, near-real-time solution for mitigating driver distraction, ultimately bolstering vehicular safety and reducing accident rates.**

***Index Terms*—Distracted Driver Detection, HOG-LPNet, Convolutional Neural Network (CNN), Histogram of Oriented Gradients (HOG), Contrast Limited Adaptive Histogram Equalization (CLAHE), Learnable Pooling Attention Module , Deep Learning, Road Safety**

I. Introduction

Distracted driving is defined as any activity that diverts attention from the primary task of driving [15]. This includes activities like using a phone for texting or calls, interacting with in-car systems, grooming, eating or drinking, reaching for objects, or talking to passengers. It poses a significant risk to both drivers and non-occupants and has become a critical issue in road safety. Distracted driving has significantly contributed to the rise in traffic accidents in the United States [1][2]. Among the many distractions, texting while driving is particularly concerning, as it takes a driver's eyes off the road for an average of five seconds. This is equivalent to driving the length of a football field at 55 mph with closed eyes [3][16].

Identify applicable funding agency here. If none, delete this.

The impact of distracted driving is reflected in troubling statistics. In 2022 alone, 3,308 people lost their lives in crashes involving distracted drivers, and 289,310 more were injured [3]. Distraction-affected crashes accounted for 8% of fatal crashes, 12% of injury crashes, and 11% of all police-reported traffic accidents [3][17]. Texting while driving increases the risk of a crash by 23 times compared to driving without distractions [2][3][18]. Young drivers aged 15–24 are particularly affected, representing the highest proportion of distraction-related crashes. In the same year, 621 non-occupants, including pedestrians and cyclists, were killed in such incidents [3].

The economic cost of distracted driving is substantial, covering medical expenses, infrastructure repairs, and property damage. Approximately 14% of these damages are linked to mobile phone use while driving [3]. These challenges highlight the need to develop more effective methods to detect distracted driving behaviors.

Recent advancements in computer vision and deep learning have led to significant transformations across industries, including automotive safety. These advancements have been key in developing more efficient models to make vehicles safer and smarter [4]–[8]. Deep learning models, especially Convolutional Neural Networks (CNNs), have become the preferred approach for image classification tasks due to their superior accuracy compared to traditional machine learning methods. Researchers have explored advanced CNN models such as AlexNet [4], VGG [5], GoogleNet [6], ResNet [7], and InceptionResNetV2 [8]. Despite their success, these models often require significant computational resources, which can limit their use in real-time driver monitoring systems.

This paper focuses on lightweight and efficient methods for detecting driver distraction, aiming to address the challenges of real-world applicability and resource constraints.

II. Related Work

III. Dataset

The dataset consists of HOG-transformed grayscale images resized to 400×400 pixels from the original 640×480 images. These pictures, taken inside a vehicle, show different driver activities shown in Fig. 1. Each image falls into one of ten categories: safe driving, texting with the right hand, talking on the phone with the right hand, texting with the left hand, talking on the phone with the left hand, operating the radio, drinking, reaching behind, doing hair and makeup, or talking

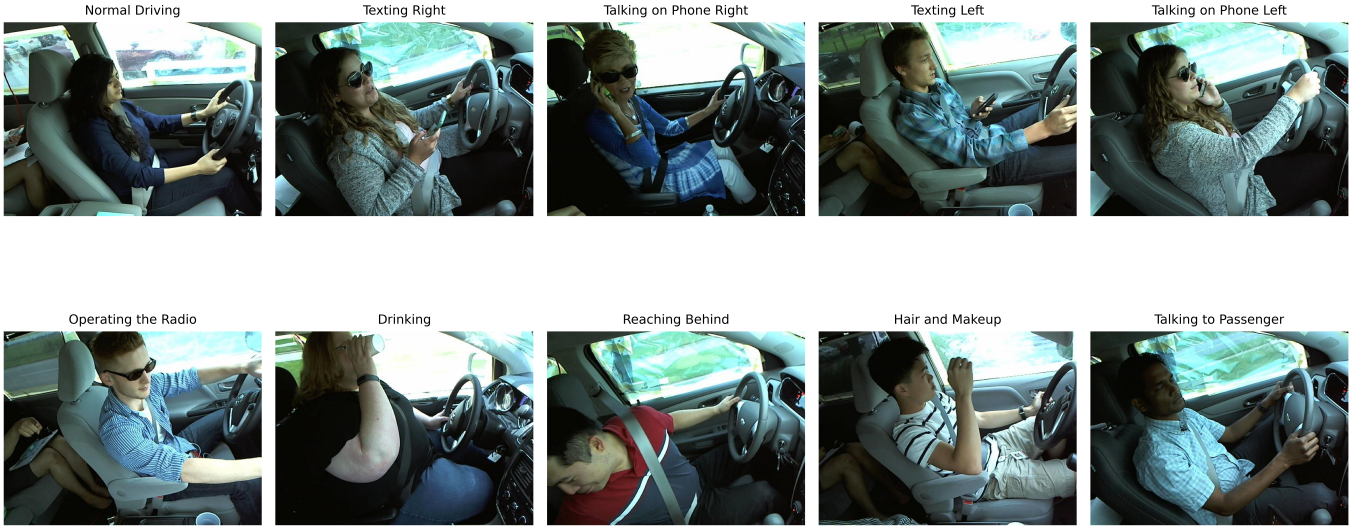


Fig. 1: Examples of images from every class fo the State Farm Dataset.

to a passenger. The distribution of HOG-transformed images across these classes is shown in Table I:

TABLE I: Image Distribution Across Driver Behavior Classes

Class (c1-c10)	Actions	Number of Images
c1	Safe Driving	2,372
c2	Texting with Right Hand	2,209
c3	Talking on the Phone with Right Hand	2,228
c4	Texting with Left Hand	2,255
c5	Talking on the Phone with Left Hand	2,255
c6	Operating the Radio	2,250
c7	Drinking	2,242
c8	Reaching Behind	1,998
c9	Engaging in Hair and Makeup	1,911
c10	Talking to a Passenger	2,108

The dataset shows slight differences in the number of images per class, but it is generally balanced, ensuring a fair representation of driver activities. It also includes 79,728 unlabeled images for testing, which helps evaluate the model fairly in conditions similar to real-world scenarios. All metadata, such as timestamps, was removed to focus solely on visual information, ensuring that the task remains within the scope of computer vision. The dataset was split into training and test sets based on drivers to eliminate the possibility of data leakage, ensuring that no driver appears in both sets. This split helps test the models on new driver patterns, making them more reliable. The images were taken in a controlled environment where participants performed activities in a stationary car being towed. This setup ensured safe and consistent data collection. However, real-world driving may bring additional challenges, such as movement, changing light, and different camera angles. The dataset, with its HOG-transformed images and balanced class distribution, is a valuable resource for developing models to detect distracted driving behaviors. It comes from the Kaggle competition, State Farm Distracted Driver Detection, a well-known benchmark for studying driver behavior.

IV. Technical Approach

A. Idea Behind the Approach

To effectively detect driver distractions while addressing the computational challenges associated with real-time processing due to its high parameter count, the design of HOG-LPNet is based on two key principles:

1) Eliminating Background Noise and Focusing on Driver Posture:
The model concentrates solely on the driver's body outline and posture by removing unnecessary background elements such as clothing colors and lighting variations. Histogram of Oriented Gradients (HOG) is used to preprocess driver images, emphasizing edges and shapes to isolate the driver's posture. This reduction in input complexity allows the Convolutional Neural Network (CNN) to learn more discriminative features, enhancing classification accuracy [28].
2) Enhancing Feature Selection with Channel Attention:
After HOG preprocessing, the extracted features are processed by a CNN integrated with Learnable Pooling Attention Module. The attention module highlights the most critical feature channels, allowing the model to focus on key parts of the driver's posture and actions. This focused approach makes the model more accurate by giving more importance to relevant features and less to unimportant ones while also keeping the computations efficient. The Learnable Pooling Attention compensates for the lightweight nature of HOG preprocessing, ensuring essential features are highlighted without adding significant computational overhead.

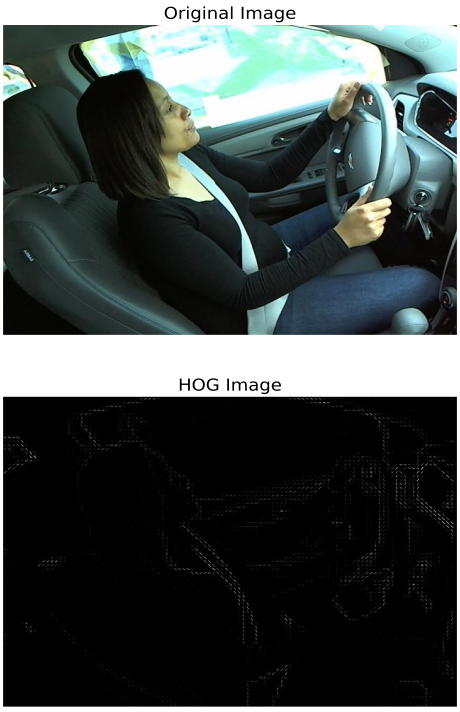


Fig. 2: Original image (above) vs. HOG-preprocessed image (below), highlighting structural edges and driver posture

3) Optimizing for Real-Time Deployment:
Although HOG-LPNet has approximately 82.64 million parameters, more than some lightweight models, this design choice is intentional to achieve high accuracy. Real-time deployment with such many parameters can be challenging on resource-constrained hardware. However, the model can be optimized by reducing the number of output channels in convolutional layers. This adjustment decreases the parameter count and speeds up inference without significantly lowering accuracy. Additionally, techniques like model pruning or quantization can further reduce the number of parameters and enhance inference speed without substantially compromising performance [29].

B. HOG Pre-processing

Histogram of Oriented Gradients (HOG) is designed to work with grayscale images. It focuses on capturing edge and gradient information effectively represented through intensity variations, as shown in Fig. 2. Each color image is first converted to grayscale to perform HOG feature extraction. This conversion eliminates color distractions and emphasizes the structural outlines of the driver's posture and movements, simplifying the data for more effective feature extraction [30]. Once converted, HOG parameters are set to 12 orientations; each cell-sized at 6×6 pixels and blocks comprising 2×2 cells. This setup balances capturing essential details and keeping the process efficient. The HOG method is applied to the grayscale images to create gradient-based visuals that clearly outline the driver's posture and movements. To make these features stand out, Contrast Limited Adaptive Histogram Equalization (CLAHE) is used to improve contrast and emphasize key details. This pre-processing step reduces input complexity by isolating critical features, allowing the Convolutional Neural Network (CNN) to focus on the most relevant aspects for accurate driver distraction detection.

C. HOG-LPNet Model Architecture

The proposed CNN architecture seen in Fig. 3, comprises four convolutional blocks designed to extract hierarchical features efficiently while maintaining computational simplicity. Each convolutional block uses a consistent 3×3 kernel size with a stride of 1 and padding of 1 to preserve spatial dimensions. The number of filters increases progressively, starting with 32 in the first block and doubling in each subsequent block, reaching 256 filters in the fourth. Batch normalization is applied after the convolutional layers in the first and third blocks to stabilize training and enhance convergence. Additionally, a Learnable Pooling Attention Module follows each convolutional block to dynamically prioritize the most informative feature channels dynamically, improving feature representation while keeping computational costs manageable. After the convolutional layers, the network employs max pooling with a kernel size of 2×2 to downsample the feature maps, reducing their spatial dimensions while retaining critical information. The feature maps output by the fourth convolutional block are flattened into a one-dimensional vector to prepare them for classification. The flattened vector passes through a series of fully connected layers, starting with 512 neurons and progressively reducing dimensionality to 256 neurons. Dropout regularization with a rate of 0.2 is applied between the fully connected layers to prevent overfitting and improve generalization. The final output layer uses a fully connected layer to classify the processed features into the 10 target classes. This design prioritizes accuracy and efficiency by combining convolutional layers for feature extraction with channel attention modules. The increasing number of filters across the layers allows the network to capture simple and complex features. At the same time, the channel attention modules ensure that the most critical parts of the input data are emphasized.

D. Learnable Pooling Attention Module (LPAM) Architecture

The *Learnable Pooling Attention Module (LPAM)* in HOG-LPNet is crafted to amplify the network's focus on crucial features while minimizing less relevant ones, thereby improving the overall feature representation and classification performance. The module operates on the grayscale input feature map $X \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels, and H and W denote spatial dimensions, Fig. 4. It employs three distinct pathways: global max pooling and two learnable 1×1 convolutional pooling layers.

Global max pooling is represented as:

$$M_{\text{max}} = \text{MaxPool}(X), \tag{1}$$

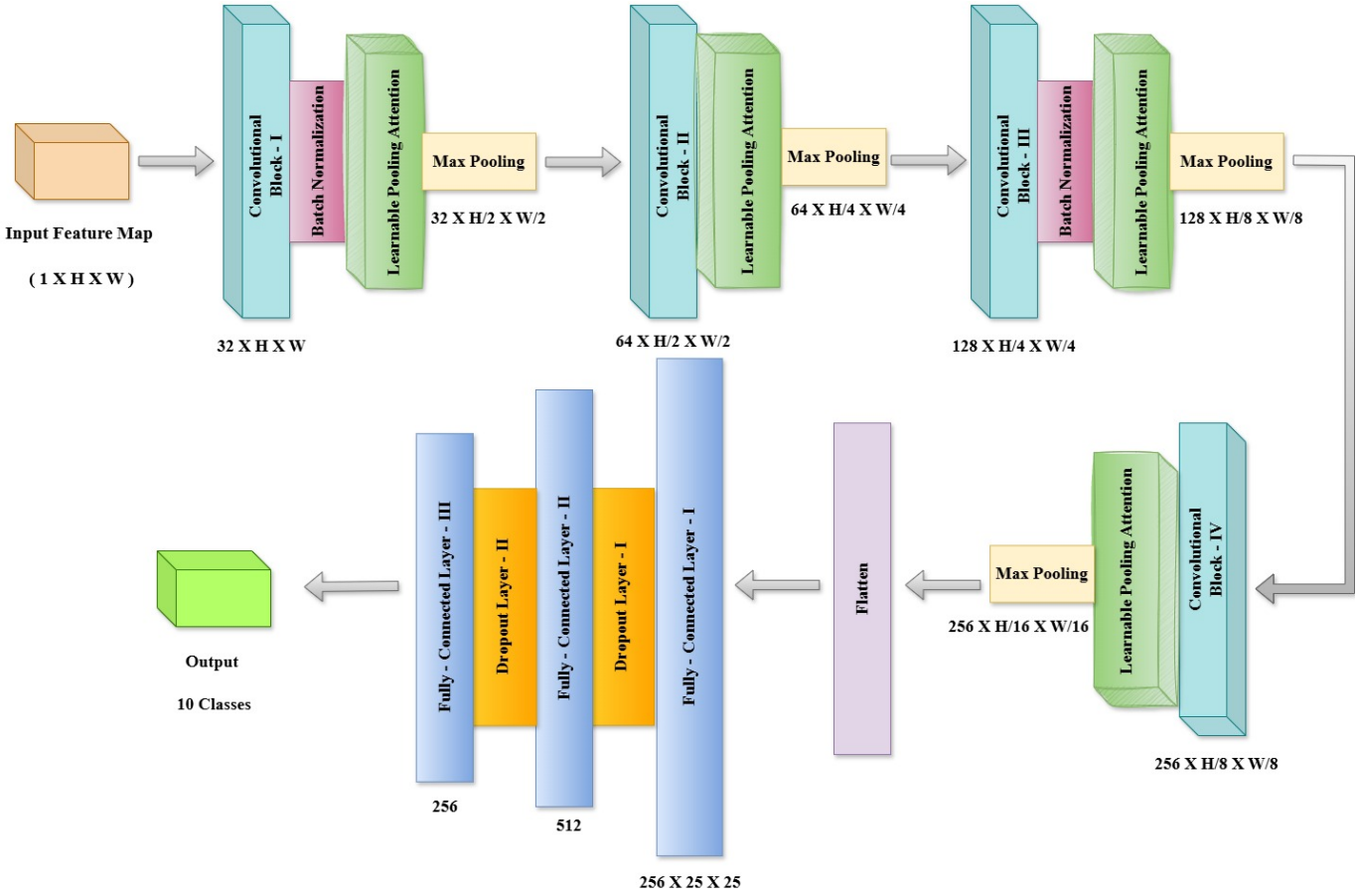


Fig. 3: Architecture of the HOG-LPNet Model

where $M_{\text{max}} \in \mathbb{R}^{C\times 1\times 1}$ summarizes the maximum activation values in each channel. Alongside this, the learnable pooling layers process the feature map using 1×1 convolutions:

$$L_1 = \text{Conv}_{1\times 1}(X), \quad L_2 = \text{Conv}_{1\times 1}(X), \tag{2}$$

producing feature maps $L_1, L_2 \in \mathbb{R}^{C\times H\times W}$. These pathways enable adaptive refinement of the most relevant features.

The outputs from these pathways are transformed through fully connected layers, where the channel dimensionality is reduced by a factor of $r = 8$, leading to a reduced dimension $C' = \frac{C}{r}$. This is expressed as:

$$\begin{aligned} F_1 &= \sigma(\text{Conv}_{1\times 1}(L_1)), \\ F_2 &= \sigma(\text{Conv}_{1\times 1}(L_2)), \\ F_{\text{max}} &= \sigma(\text{Conv}_{1\times 1}(M_{\text{max}})), \end{aligned} \tag{3}$$

where $F_1, F_2, F_{\text{max}} \in \mathbb{R}^{C'\times 1\times 1}$, and σ denotes the SiLU activation function. The SiLU activation introduces non-linearity, improving gradient flow and helping prevent overfitting during training.

The three outputs, F_1, F_2, and F_{max}, are summed elementwise, and a final fully connected layer with a sigmoid activation generates the channel attention map A_{channel}:

$$A_{\text{channel}} = \sigma(\text{Conv}_{1\times 1}(F_1 + F_2 + F_{\text{max}})), \tag{4}$$

where $A_{\text{channel}} \in \mathbb{R}^{C\times 1\times 1}$ dynamically weighs each channel by its relevance.

Finally, the recalibrated feature map X' is computed by applying the attention map to the original input feature map using element-wise multiplication:

$$X' = X \odot A_{\text{channel}}, \tag{5}$$

where $X' \in \mathbb{R}^{C\times H\times W}$ represents the refined feature map with enhanced attention on the most important channels.

By integrating LPAM after each convolutional block, HOG-LPNet ensures a robust and efficient mechanism for feature prioritization at every stage of processing. This balance between computational simplicity and feature enhancement makes HOG-LPNet well-suited for real-time applications, such as detecting driver distractions in resource-constrained environments.

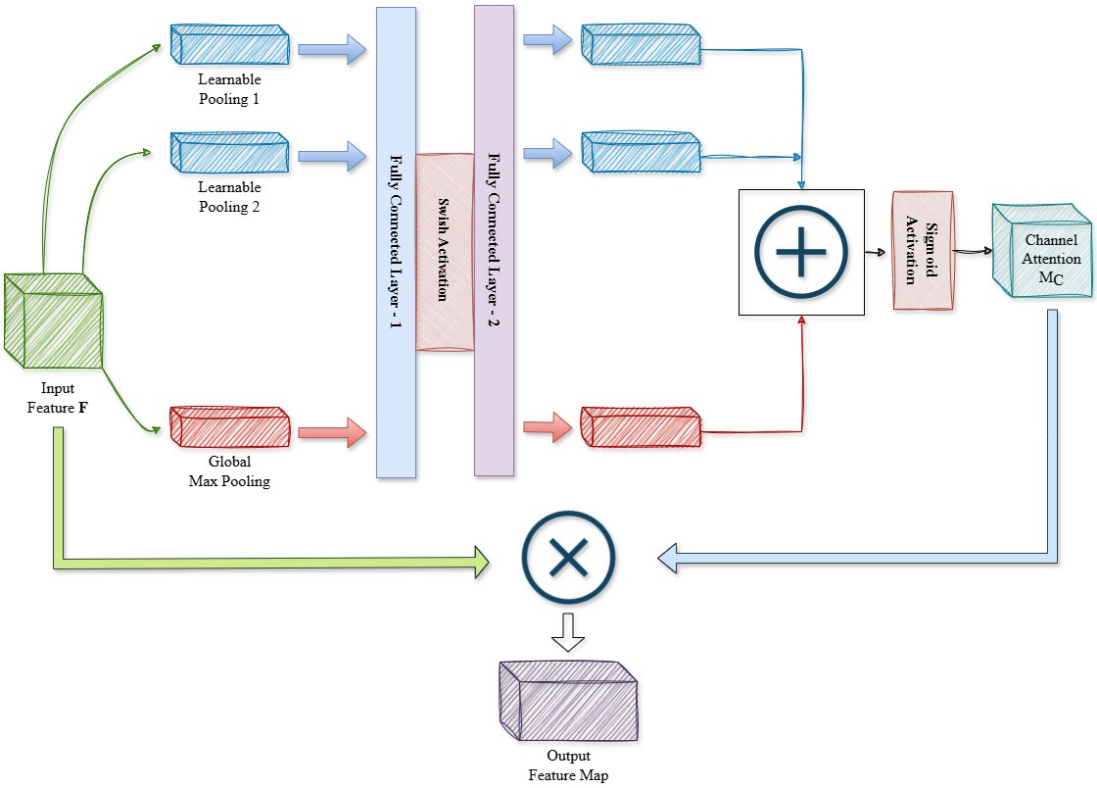


Fig. 4: Learnable Pooling Attention Module Architecture

V. Experimentation and Results

In this section, we describe the experiments conducted to evaluate the performance of HOG-LPNet in detecting distracted driving behaviors. The dataset, comprising ten distinct driver activity classes, was split into training and testing sets, with 25% of the data reserved for testing. Notably, the architecture of HOG-LPNet does not employ global average pooling in the fully connected layers. Instead, a flattened feature map of size 256×25×25 is directly used as the input to the fully connected layer.

The choice not to use global average pooling was made after initial experiments showed that it caused the loss of critical spatial details needed for accurate classification. This loss of information resulted in much lower accuracy and caused the model to overfit the training data. Therefore, the feature maps were flattened directly, keeping their spatial dimensions intact. This approach allowed the network to retain critical information about the structure and position of features, which helped the model generalize better and achieve higher accuracy.

During experimentation, we encountered problems with the overfitting of training data. Shallow architectures with a limited number of convolutional layers failed to capture sufficient hierarchical features, leading to reduced accuracy in training and validation data. Deeper architectures significantly increased the number of parameters without improving performance, making them less efficient.

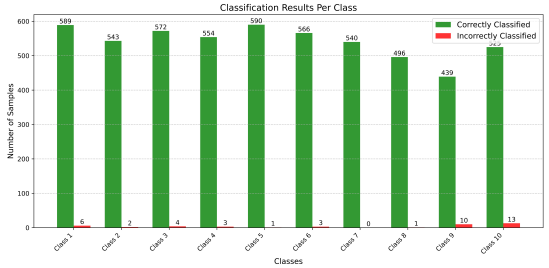
The chosen medium-depth architecture provided a good balance by effectively extracting key features while keeping the computation manageable. To address overfitting, dropout layers were added to the fully connected layers, reducing reliance on specific neurons. Batch Normalization was applied after alternate convolutional layers to stabilize training and improve generalization. These changes made the model more reliable and less prone to overfitting.

The Learnable Pooling Attention Module (LPAM) played a vital role in refining the feature selection. It highlighted the most important details in the input, enabling the network to focus on essential elements at every processing stage. Placing LPAM after each convolutional block helped the model produce a more accurate and efficient representation of the input features, enhancing its ability to detect driver distractions effectively. The experiments conducted to evaluate the model's performance are detailed below.

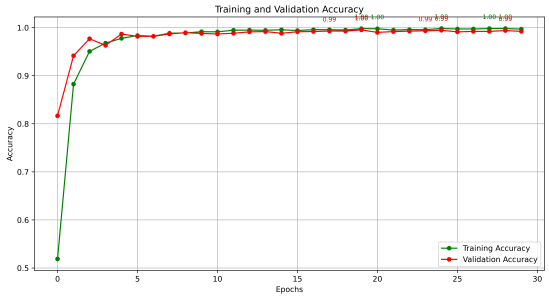
A. *Experiment 1*

In the first experiment, the CNN baseline model was trained with minimal augmentation, applying only random rotations to add slight variability to the image orientation. Training was carried out for 30 epochs with a batch size of 1 and a learning rate of 0.001, taking 7 hours and 9 minutes to complete. The model achieved an average accuracy of 98.16% and a maximum accuracy of 99.52%. The classification results per class, as shown in Fig. 5. (a), demonstrated good performance across all categories. However, classes like "Engaging in

Hair and Makeup" and "Talking to a Passenger" showed slightly higher misclassification rates than others. This could be due to similarities in posture or hand movements with other activities. The training and validation graphs in Fig. 5. (b), (c) demonstrate effective model performance, with a steady decrease in loss indicating successful convergence and minimal overfitting, and a rapid rise in accuracy during early epochs stabilizing near 100%, showcasing robust learning and generalization.



(a)



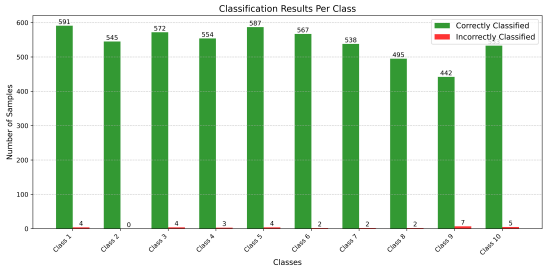
(b)

(c)

Fig. 5: Experiment 1: a) Classification results, b)training and validation accuracy, and c) training and validation loss plots.

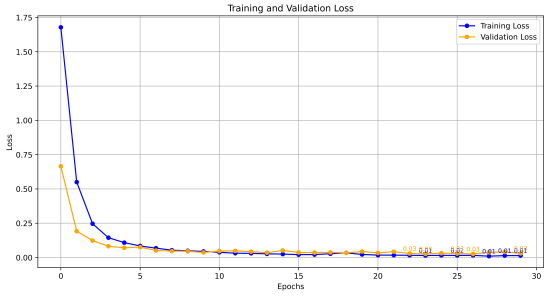
B. Experiment 2

The results of Experiment 2 show improved model performance across most classes, with high accuracy and only a few misclassifications. Some classes, like "Texting with Right Hand," were classified ideally, showing the benefits of the improved preprocessing and augmentation methods. However, there were still a few errors in classes like "Talking to a Passenger" and "Engaging in Hair and Makeup" as shown in Fig.6. (a). Normalization, with a mean of 0.5 and a standard deviation of 0.5, helped keep pixel values consistent and made the training process more stable. The model performed well overall; however, subtle activities remained more challenging, indicating areas that could still be improved. The model achieved an average accuracy of 97.93% and a maximum accuracy of 99.49%, with a batch size of 10, a learning rate of 0.01, and training conducted for 30 epochs. This experiment also highlighted the balance between training speed and accuracy. Training took 3 hours and 34 minutes, faster than the baseline experiment, but it slightly dropped average accuracy. The training and validation loss graph in Fig. 6.(b), (c) shows a more consistent decrease compared to the previous graphs, indicating improved stability in model training. Similarly, the training and validation accuracy graph reflects faster convergence and reduced fluctuations, highlighting better generalization and overall performance compared to earlier experiments.



(a)

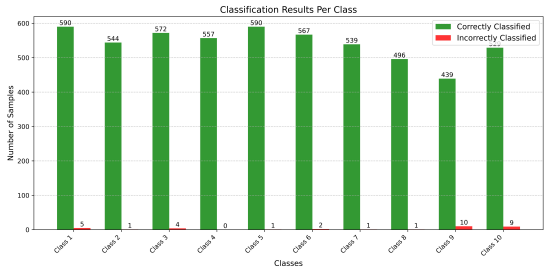
(b)



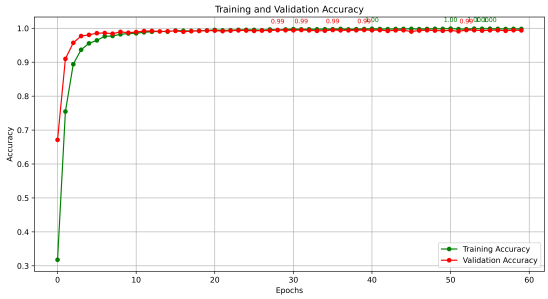
(c)

Fig. 6: Experiment 2: a) Classification results, b) training and validation accuracy, and c) training and validation loss plots.

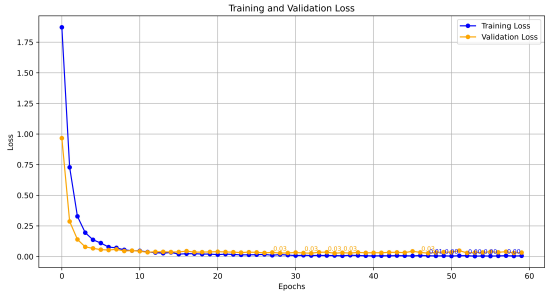
C. Experiment 3



(a)



(b)



(c)

Fig. 7: Experiment 3: a) Classification results, b) training and validation accuracy, and c) training and validation loss plots.

In this experiment, the preprocessing and augmentation steps from Experiment 2 were kept the same, including random rotations and normalization of pixel values with a mean of 0.5 and a standard deviation of 0.5. The number of epochs was increased to 60 to study the impact of more extended training. The model was trained with a batch size of 10 and a reduced learning rate of 0.001 to ensure stable weight updates. The training took 7 hours and 45 minutes. The model achieved an average accuracy of 98.47% and a maximum accuracy of 99.49%. Extending the training period helped the model learn better, improving its overall performance compared to Experiments 1 and 2. Specifically, the average accuracy achieved in this experiment was the highest among the three, demonstrating the advantages of longer and more stable training. The reduced learning rate ensured steady progress during training and improved the model's performance on unseen data. This setup effectively balanced accuracy and training duration for extended sessions. The classification results in Fig. 7. (a), showed high accuracy across most classes, with only a few mistakes in challenging categories such as "Talking to a Passenger" and "Engaging in Hair and Makeup." Similar to the earlier experiments, these errors were likely caused by similarities in movements and features between some activities.

D. Comparison with other models

TABLE II: Comparison of Models for Driver Distraction Detection

Model Name	Parameters (in Millions)	Accuracy (%)
AlexNet + TripletLoss	63.2	98.6
VGG-GAP	140	98.7
Vanilla CNN with data augmentation	26.05	97.05
InceptionV3 + Xception-50 + Xception + VGG-19	214.3	97.0
HOG-LPNet	82.64	99.52

The comparison of models in Table II, highlights a balance between accuracy, size, and complexity. AlexNet + TripletLoss is a straightforward model with 63.2 million parameters and achieves 98.6% accuracy. It is efficient and works well for systems with moderate processing power. VGG-GAP, on the other hand, improves accuracy slightly to 98.7%, but it comes with a much larger size of 140 million parameters, which makes it less practical for lightweight systems. The Vanilla CNN with Data Augmentation is the most miniature model, with only 26.05 million parameters, and achieves 97.05% accuracy. This makes it a good choice for environments where resources are limited, though it doesn't perform as well as the more advanced models. The ensemble model, which combines InceptionV3, Xception-50, Xception, and VGG-19, uses a vast 214.3 million parameters to achieve 97.0% accuracy. Despite its complexity, its accuracy is slightly better than the Vanilla CNN, which shows that adding more layers or models doesn't always guarantee significantly better results. The most effective model is the proposed HOG-LPNet, which achieves the highest accuracy of 99.52% while keeping the parameter count at 82.64 million. It achieves a good balance between performance and efficiency by using HOG preprocessing and a Learnable Pooling Attention Module. This makes it an excellent choice for real-time systems like detecting driver distractions, where both speed and accuracy are critical. In summary, simpler models like Vanilla CNN are useful for low-resource situations, but HOG-LPNet stands out as the best option for high-accuracy tasks without overwhelming resource requirements. It strikes the right balance between being fast, efficient, and accurate.

CONCLUSION

This article introduced the HOG-LPNet model, a balanced neural network designed to classify driver activities and detect distracted driving. The model uses techniques like the

Learnable Pooling Attention Module (LPAM), dropout layers, and batch normalization to strengthen feature selection, reduce overfitting, and maintain stable training. Experiments were conducted on a dataset with ten different driver activity classes to study the effects of preprocessing, augmentation, and hyperparameter tuning on the model's performance. The model achieved a maximum accuracy of 99.52%, the best average accuracy of 98.47%, and the lowest average loss of 0.0586 with extended training. These results show that HOG-LPNet provides both high accuracy and efficient performance. The findings confirm the design of HOG-LPNet, especially its ability to handle subtle and complex activities. This work makes roads safer by offering a practical, real-time driver monitoring solution. Future studies can focus on expanding the dataset, testing the model in diverse driving conditions, and exploring improved architectures to further enhance its capabilities.

REFERENCES

[1] National Highway Traffic Safety Administration, "Distracted Driving," [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813559

[2] National Highway Traffic Safety Administration, "The Issue & Consequences of Distracted Driving," [Online]. Available: https://www.nhtsa.gov/risky-driving/distracted-driving

[3] U.S. Department of Transportation, National Highway Traffic Safety Administration, National Center for Statistics and Analysis, Distracted Driving in 2022: Traffic Safety Facts - Research Note, DOT HS 813 559, Apr. 2024. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/78043

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "AlexNet," in Proc. 25th Conf. Neural Information Processing Systems (NIPS), 2012. Available: https: https://dl.acm.org/doi/10.1145/3065386

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014. [Online]. Available: https://arxiv.org/pdf/1409.1556

[6] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594. keywords: Computer architecture;Convolutional codes;Sparse matrices;Neural networks;Visualization;Object detection;Computer vision,

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016. [Online]. Available: https://arxiv.org/pdf/1512.03385

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," arXiv preprint arXiv:1602.07261, 2016. [Online]. Available: https://arxiv.org/pdf/1602.07261

[9] Köhncke, Martin & Jaelani, Yogi & Mendler, Alexander & Neumann, Lizzie & Wittenberg, Philipp & Rode-Klemm, Alina & Keßler, Sylvia. (2024). Static and Dynamic Load Tests on the Bridge Vahrendorfer Stadtweg. 10.48550/arXiv.2412.15713.

[10] Neumann, T. (2024). Analysis of Advanced Driver-Assistance Systems for Safe and Comfortable Driving of Motor Vehicles. Sensors, 24(19), 6223. Available: https://doi.org/10.3390/s24196223

[11] Ashlesha Kumar, Kuldip Singh Sangwan, Dhiraj. Computer Vision Approaches for Driver Distraction Detection. arXiv. [Online]. Available: https://arxiv.org/pdf/2401.10213v1

[12] N. Darapaneni, J. Arora, M. Hazra, N. Vig, S. S. Gandhi, S. Gupta, and A. R. Paduri, "Detection of Distracted Driver using Convolution Neural Network," arXiv, 2022. [Online]. Available: https://arxiv.org/abs/2204.03371.

[13] Ou, C., Ouali, C., Karray, F. (2018). Transfer Learning Based Strategy for Improving Driver Distraction Recognition. In: Campilho, A., Karray, F., ter Haar Romeny, B. (eds) Image Analysis and Recognition. ICIAR 2018. Lecture Notes in Computer Science(), vol 10882. Springer, Cham. Available: https://doi.org/10.1007/978-3-319-93000-8_50

[14] E. Akdag, Z. Zhu, E. Bondarev, and P. H. N. de With, "Transformer-based Fusion of 2D-pose and Spatio-temporal Embeddings for Distracted Driver Action Recognition," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 5453–5462, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1109/CVPRW59228.2023.00576.

[15] Sun, X. (2018). Investigating Problem of Distracted Drivers on Louisiana Roadways. Retrieved from https://repository.lsu.edu/transet_pubs/25

[16] Empowering Road Safety — The Ward Law Group. https://www.855dolor55.com/blog/the-ward-law-group-response-to-florida-new-texting-while-driving-law/

[17] Distracted Driving Prevention New PSA campaign from Ad Council, NHTSA — Ad Council. https://www.adcouncil.org/learn-with-us/press-releases/for-goodness-sakes-nhtsa-and-the-ad-council-release-new-creative-for-distracted-driving-prevention-month

[18] The Dangers of Texting While Driving: Why It's Not Worth the Risk - Trial Lawyer View. https://triallawyerview.com/the-dangers-of-texting-while-driving-why-its-not-worth-the-risk-2/

[19] Y. Liang, M. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 2, pp. 340–350, 2007.

[20] Chihang, Zhao & Zhang, Bailing & He, Jie & Lian, Jie. (2012). Recognition of driving postures by contourlet transform and random forests. Intelligent Transport Systems, IET. 6. 161-168. 10.1049/iet-its.2011.0116.

[21] E. Ohn-Bar and M. M. Trivedi, "Head, Eye, and Hand Patterns for Driver Activity Recognition," IEEE Intelligent Vehicles Symposium, 2013.

[22] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang and D. Levi, "Driver Gaze Tracking and Eyes Off the Road Detection System," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 2014-2027, Aug. 2015, doi: 10.1109/TITS.2015.2396031.

[23] State Farm Distracted Driver Detection. Available at: https://www.kaggle.com/c/state-farm-distracted-driver-detection

[24] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time Distracted Driver Posture Classification," CoRR, vol. abs/1706.09498, 2017. [Online]. Available: http://arxiv.org/abs/1706.09498.

[25] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver Distraction Identification with an Ensemble of Convolutional Neural Networks," CoRR, vol. abs/1901.09097, 2019. [Online]. Available: http://arxiv.org/abs/1901.09097.

[26] N. Moslemi, R. Azmi and M. Soryani, "Driver Distraction Recognition using 3D Convolutional Neural Networks," 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 2019, pp. 145-151, doi: 10.1109/PRIA.2019.8786012.

[27] A. A. Q. Mohammed, X. Geng, J. Wang, and Z. Ali, "Driver distraction detection using semi-supervised lightweight vision transformer," Engineering Applications of Artificial Intelligence, vol. 129, p. 107618, 2024. [Online]. Available: https://doi.org/10.1016/j.engappai.2023.107618.

[28] Ye, L. et al. (2020). Using CNN and Channel Attention Mechanism to Identify Driver's Distracted Behavior. In: Pan, Z., Cheok, A., Müller, W., Zhang, M. (eds) Transactions on Edutainment XVI. Lecture Notes in Computer Science(), vol 11782. Springer, Berlin, Heidelberg. Available: https://link.springer.com/chapter/10.1007/978-3-662-61510-2_17

[29] Li, Y., Xu, P., Zhu, Z., Huang, X., Qi, G. (2022). Real-Time Driver Distraction Detection Using Lightweight Convolution Neural Network with Cheap Multi-scale Features Fusion Block. In: Jia, Y., Zhang, W., Fu, Y., Yu, Z., Zheng, S. (eds) Proceedings of 2021 Chinese Intelligent Systems Conference. Lecture Notes in Electrical Engineering, vol 804. Springer, Singapore. Available: https://link.springer.com/chapter/10.1007/978-981-16-6324-6_24

[30] Mefteh, S., Kaâniche, MB., Ksantini, R., Bouhoula, A. (2023). Learning Human Postures Using Lab-Depth HOG Descriptors. In: Nguyen, N.T., et al. Computational Collective Intelligence. ICCCI 2023. Lecture Notes in Computer Science(), vol 14162. Springer, Cham. Available: https://dl.acm.org/doi/10.1007/978-3-031-41456-5_42