Journal of Science and Technology
ISSN: 2362-1443 (Print) 3059-9490 (Online)
2024, Vol. 4, No. 2 (Special Issue), 58-64

Real-Time Driver Drowsiness Detection Using CNN: A Web-Based Application Approach

Prashuma Lama
Nepal College of Information Technology
Pokhara Universiry
prasuma25lama@gmail.com

Epsa Pokhrel
Nepal College of Information Technology
Pokhara Universiry
epsapokhrel14@gmail.com

Sampada Subedi
Nepal College of Information Technology
Pokhara University
smsubedii@gmail.com

Prakash Paudel
Nepal College of Information Technology
Pokhara Universiry
visitppdl@gmail.com

Article History:
Received: 31 July 2024
Revised: 14August 2024
Accepted: 18 December 2024

Keywords— Convolutional Neural Networks, Haar cascade classifier,ReLU, softmax, computer vision, kaggle, precision, recall, f1 score

***Abstract*—Driver drowsiness is a significant contributor to traffic accident. Lack of sleep or long driving hours can impair a driver's attention and mental performance, which raises the possibility of harm or death in accidents. To solve this issue we have introduced Driver Drowsiness Detection System. The fundamental goal of our method is to use computer vision to identify the driver's level of tiredness based on their eye condition and to trigger an alarm to notify the user when a sleepy state is noted. This study presents a web application based on Convolutional Neural Networks for real-time drowsiness identification with an accuracy of 97%. The methodology involves analyzing facial features, particularly eye closure, using computer vision techniques, ReLU and Softmax activation functions to alert drivers when their eyes remain closed beyond a set threshold. Haar Cascade classifiers are employed for face and eye detection to preprocess image data effectively. The dataset used consists of images of faces and eyes of people sourced from Kaggle. The results shows the efficiency of this approach in real-time detection of drowsiness,achieving 97% weighted average score for precision,recall , and F1 scores across all classes. This system highlights the potential of CNNs and computer vision in addressing the critical issue of driver fatigue. The importance of this research lies in its ability to enhance road safety by developing a robust,real-time drowsiness detection mechanism that can handle safety of drivers.**

I. Introduction

The increasing number of traffic accidents worldwide has raised serious concerns about public health, with driver fatigue and drowsiness, playing a crucial role [1]. According to data from the National Highway Traffic Safety Administration (NHTSA), sleepy driving is a factor in 91,000 police-reported crashes that result in 50,000 injuries and 800 fatalities annually, or between 1% and 2% of all crashes, injuries, and deaths. [2]. Effective ways to identify and stop driver drowsiness are desperately needed, as demonstrated by the seriousness and frequency of these occurrences. So, one of the main fields of study in the effort to increase road safety is the development of trustworthy and efficient technologies to identify and reduce driver drowsiness. Road safety is seriously threatened by driver sleepiness, which is defined as a reduction in attention and cognitive function brought on by sleep deprivation or monotonous driving conditions. Drivers who are sleep deprived are more prone to be in accidents because they react more slowly, make poorer decisions, and are less alert [3].

Driver drowsiness detection systems aim to identify signs of driver fatigue and provide immediate warnings to pre-vent fatalities. [4]. Computer vision and machine learning just a few of the technologies that these systems make use of. Despite the significant improvements in the subject of drowsiness detection mechanisms, a number of challenges and limitations persist. The fact that people react differently to fatigue is one of the main challenges [5]. The symptoms of drowsiness in drivers can vary, making it challenging to come up with a general treatment. Environmental elements that impact detecting system accuracy include road types and lighting conditions. One important area of continuing research is ensuring the robustness and reliability of these systems under various circumstances. Based on the results of the National Sleep Foundation's Drowsy Driving Survey, the percentage of people who report driving when sleepy is shown in the bar chart "Fig.1" below [6].

With the introduction of new technologies and the advance-ment of diverse research, the field of driver drowsiness detec-tion has an optimistic future. The

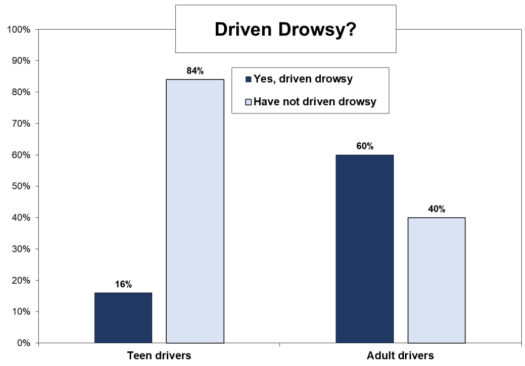

Fig. 1. Frequency of drowsy driving according to National Sleep Foundation's Drowsy Driving Survey [6]

capabilities of drowsiness detection systems can be further improved by integrating wearables, vehicle-to-everything (V2X) connectivity, and Internet of Things (IoT) devices [7]. Keeping in mind the importance of safe driving, our team has developed a web-based application that allows users to log in and monitor their drowsiness levels in real-time. After the detection of drowsiness, system sounds an alarm, which helps keep drivers awake and focused on the road. Our project's primary goals are:

- To use computer vision to detect the driver's drowsiness according to their eye state.
- To use alarm sound to alert the user whenever sleepy state is detected.

II. Related Works

In previous years, numerous techniques have been applied to enhance the accuracy of driver drowsiness detection systems. These methods utilize a variety of signals and techniques to identify signs of drowsiness in drivers.

Mardi et al. [8] This approach highlights the effectiveness of EEG signals in capturing the physiological indicators of drowsiness. They developed a model utilizing Electroencephalography (EEG) signals to detect drowsiness. They extracted chaotic features and the logarithm of energy from the EEG signals to differentiate among drowsy and alert states. An Artificial Neural Network (ANN) was then employed for classification, resulting in an accuracy of 83.3%.

Danisman et al. [9] recommended a method based on changes in eye blink rate, which is a well-known indicator of drowsiness. After identifying the facial region in photos using the Viola-Jones detection algorithm, they used a neural network-based detector to detect the pupils. The blink rate was determined by the system, and a higher blink rate was used to indicate tiredness. With an impressive 94% accuracy, the approach showed to be a reliable means of detecting indicators of drowsiness based on indicators of vision. Because of its great accuracy, this system has the potential to be a reliable real-time driver fatigue monitoring tool that improves road safety by immediately alerting driver when they are at risk of falling asleep.

Cui et al. [10] used a condensed and comprehensible Convolutional Neural Network (CNN) to identify common EEG characteristics among many people to detect driver drowsiness. By integrating the Global Average Pooling (GAP) layer into the model architecture, the Class Activation Map (CAM) technique was utilized to identify the input signal regions having the highest influence on classification. The suggested model can classify 2-class cross-subject EEG signals with an average accuracy of 73.22% on 11 participants, according to the results.

Dwivedi et al. [11] developed a model using Representation Learning, a part of Convolutional Neural Networks(CNNs) to detect driver drowsiness. The video frames that were retrieved were fed to face detectors based on Jones Harr-like features and Voila, and then they were sent again to a CNN with many layers. The outputs of the hidden layer, which were considered the extracted features, were used to train the softmax layer classifier. With a 78% accuracy rate, the model demonstrated the value of deep learning methods in this field.

Picot et al. [12] employed a combination of eyes and brain activity to identify driver fatigue. An electrode gel with a single channel was utilized to track brain activity. Characterization and blinking are employed to track visual activity. EOG is used to extract blinking features whereas fuzzy logic was used to merge these two features into an EOG-based detector. This study obtained an accuracy of 80.6% when evaluated on a dataset of twenty distinct drivers.

Said et al. [13] presented an eye-tracking-based driver drowsiness system. The system alerts the driver by sounding an alarm when it senses fatigue.This study used the Viola-Jones model to determine the areas of the face and eyes. It produced an accuracy of 72.8% outside and 82% inside in tests that were carried out indoors.

Park et al. [14] proposed a deep learning-based approach for detecting driver drowsiness using input videos. Three unique deep networks—AlexNet, VGG-FaceNet, and FlowImageNet—are used in their technique to extract and learn features from the video data. These networks are used to identify complex patterns and characteristics that point to fatigue. An accuracy of about 73% was obtained in the studies done on the NTHU driver drowsiness video collection. This study highlights the potential for better accuracy and robustness in real-world applications by demonstrating how successful it is to use various deep-learning frameworks to improve sleepiness detection.

Jabbara et al. [15] developed a deep learning-based algorithm to identify driver fatigue in Android applications. In this case, a model based on facial landmark point detection

Fig. 2. closed eyes [18]

was created. Here, pictures are initially taken from video frames, and landmark coordinate points are retrieved using the Dlib library. These landmark locations are provided as the multilayer perceptron classifier's input. Based on these points, the classifier classifies the data as sleepy or not sleepy. Applying the NTHU Drowsy Driver Detection Dataset [16], this approach was tested and shown to be more than 80% accurate.

Reddy et al. [17]addressed the challenge of deploying a real-time drowsiness detection system on low-cost embedded boards. Their method improves practical deployment by reducing a huge baseline model into a lightweight form appropriate for embedded systems. The suggested method runs at 14.9 frames per second (FPS) on a Jetson TK1 and achieves 89.5% accuracy in a 3-class categorization. The compression of this type guarantees effective operation with low power consumption and makes integration with vehicle Electronic Control Units (ECUs) easier. The method points out a compromise between maintaining a high level of detection accuracy and ensuring system feasibility for in-car real-time applications.

III. Methodology

A. Datasets

The dataset shown in "Fig2" and "Fig3" utilized for this project has been sourced from Kaggle. It comprises images categorized into four distinct classes: "yawn," "no yawn," "closed," and "open." Additionally, the folder includes Haar Cascade classifier files for frontal face detection, as well as right eye and left eye detection [18].

B. Algorithm

1) Haar Cascade Algorithm
The Haar Cascade algorithm was chosen for its proven effectiveness in real-time object detection, particularly in face detection, which is crucial for preprocessing in our drowsiness detection model. Developed by Viola and Jones [19] , this algorithm excels in detecting objects like faces in images using simple rectangular features called "Haar-like features." These features

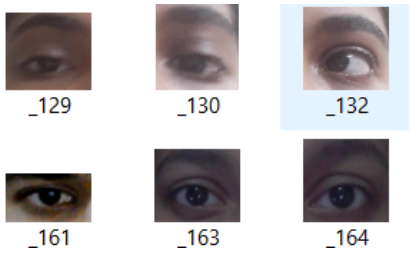

Fig. 3. open eyes [18]

help identify variations in pixel intensities, which allows the algorithm to focus on regions in the image that are most likely to include the desired object. We selected the Haar Cascade algorithm over other alternatives, such as HOG (Histogram of Oriented Gradients) or LBP (Local Binary Patterns), because of its efficient use of computational resources, making it highly suitable for real-time applications [20]. The sequential nature of Haar Cascade allows it to quickly discard irrelevant image sections, reducing the need for intensive computation in areas with little significance. This approach enhances detection speed, which is critical for our use case of real-time drowsiness detection [21].

A series of classifiers, each intended to identify ever more complex features, are used by the algorithm. The sequential method employed by the Haar Cascade enables it to efficiently eliminate irrelevant sections of the image, concentrating its computing efforts on places that show possibilities.

- Use of Haar Cascade in Our Code:

Face Detection: To detect faces in images, the Haar Cascade algorithm is applied. The Haar Cascade classifier for face detection is loaded and applied to images in the dataset by the function 'face_for_yawn'. The identified faces are then removed and reshaped in preparation for further development.
Training Setup: The model is trained using the scaled and identified facial images. By performing this preprocessing, it is ensured that the model is trained on the face and not the full image.
Integration using Convolutional Neural Networks (CNNs): The processed images (faces) are sent into a CNN (Convolutional Neural Network) following face detection. These faces are directed to be classified by CNN into many categories, including "yawn," "no_yawn," "Closed," and "Open."

2) CNN Algorithm:
For the image classification task, we employed a Convolutional Neural Network (CNN) due to its higher capability to learn hierarchical representations from image data [22]. CNNs are particularly well-suited for

image-related tasks because they can automatically extract features at various levels of abstraction, starting from basic edges to more complex patterns such as facial expressions or eye closure [23].

We opted for CNNs instead of traditional machine learning algorithms like SVM (Support Vector Machine) or KNN (K-Nearest Neighbors) because CNNs can directly process raw image pixels without requiring extensive manual feature engineering [24]. CNNs are also robust to changes in scale, orientation, and translation, making them ideal for real-world applications where faces or expressions might vary in appearance.Convolutional layers enable CNNs to effectively learn spatial feature hierarchies, while techniques like pooling aid in dimensionality reduction of the data, mitigating the risk of overfitting [25]. Additionally, CNNs benefit from weight sharing, reducing the number of trainable parameters and improving generalization to new data.

Multilayer perceptrons are regularized variants of CNNs. Generally speaking, multilayer perceptrons refer to completely connected networks, in which every neuron in one layer is coupled to every other layer's neuron. These networks' "fully-connectedness" renders them vulnerable to overfitting of data. Regularization techniques often include including a magnitude measurement of weights into the loss function [26]. CNNs, on the other hand, approach regularization differently. Using the hierarchical structure in the data, they piece together more complicated patterns from smaller, simpler patterns [27].

CNNs can automatically learn hierarchical representations of features from the input images, which makes them especially useful for image classification problems [28]. CNNs can learn patterns at various abstraction levels thanks to this hierarchical method, which starts with basic characteristics like edges and textures and progressively progresses to more complicated features that are important for categorization. The capacity of CNNs to share weights across several input components reduces the number of parameters and improves the network's ability to generalize to new, unknown data, which is one of its main advantages. When features appear in multiple areas of an image, weight sharing becomes especially helpful.

CNNs are also robust to translation, rotation, and scaling of the input images,making them appropriate for a variety of real-world applications where images may vary in size and orientation [29] . Additionally, CNNs can be trained efficiently on large datasets using techniques like stochastic gradient descent, backpropagation, and modern optimization algorithms, enabling them to learn complex patterns from massive amounts of data.

C. Data Preprocessing

• Image Loading and Preprocessing: Images are resized to a fixed size (145x145 pixels) using cv2.resize to ensure uniformity across the dataset.

• Face Detection: Using a machine learning object detection approach, to detect faces in the images the Haar Cascade Classifier is utilized. Thousands of both positive and negative photos have been used to pre-train this algorithm to recognize faces.

• Eye Detection: For the eye state detection ("Closed" or " Open"), the haarcascade eye.xml file is used in conjunction with face detection. This ensures that only the eye region is processed further.

• Data Augmentation and Preparation Data Augmentation: " ImageDataGenerator" is used to apply various transformations (rescaling, zooming, horizontal flipping, rotation) to the training data, enhancing the model's robustness.

• Train-Test Split: train test split divides the dataset into training and testing sets.

• Building CNN Model Architecture:

Convolutional Layers: Four convolutional layers with ReLU activation functions are used for feature extraction.

Pooling Layers: Max Pooling layers are used after each convolution layer to reduce the spatial dimensions.

Dropout: To stop the over-fitting process, a dropout layer is applied.

Fully Connected Layers: Dense layers are employed after flattening to get the ultimate categorization.

• Compilation: Categorical Cross-entropy is the loss function that is employed. Adam is the optimizer that is employed.

• Performance metrics: The accuracy metric is employed.

• Training: Using the supplemented training data (train generator), the model is trained. The model's performance is tracked throughout training using validation data (test generator).

• Evaluation and Prediction: Model Evaluation: The test dataset is used to assess the trained model, and the accuracy, confusion matrix, and classification report are used to gauge the model's performance.

IV. Result and Discussion

A. Classification Report

From the classification report shown in "Table I" it is clear that the model achieved high precision (0.95), recall (0.95), and F1-score (0.95) across all classes (yawn, no_yawn, Closed, Open), with an overall accuracy of 0.97. Each class's precision and recall are above 85%, reflecting the model's ability to correctly identify and distinguish between the different states with minimal false positives and false negatives, suggesting its robustness and reliability for real-time driver drowsiness detection. These values are calculated

TABLE I
CLASSIFICATION REPORT

	precision	recall	f1-score	support
yawn	0.92	0.87	0.89	63
no_yawn	0.91	0.93	0.92	74
Closed	0.99	0.99	0.99	215
Open	0.99	0.99	0.99	226
accuracy			0.97	578
macro avg	0.95	0.95	0.95	578
weighted avg	0.97	0.97	0.97	578

using below defined methods and equations:

1) Accuracy:
By dividing the number of accurately predicted cases by the total number of instances, accuracy determines how accurate a model is overall. A high accuracy score means that a large percentage of the model's predictions are accurate. It is computed as follows:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total number of instances}} \quad (1)$$

2) Precision:
Accuracy of optimistic forecasts is measured by precision. Its main goal is to reduce the number of false positives, meaning that when the model predicts a positive result, it is probably right. Precision is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

3) Recall:
Recall evaluates the model's capacity to spot true positives while reducing false negatives. A high recall rate indicates that the model successfully catches every positive occurrence in the dataset. The formula for recall is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

4) F1 Score:
The harmonic mean of recall and precision is the F1 score. It is especially helpful in striking a balance between recall and precision by offering a single metric that takes both into account. When recall and precision are both high, the F1 score is also high. It is computed as follows:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

B. Confusion Matrix

The confusion matrix in the figure "Fig.4 illustrates the performance of a classification model on four classes: yawn, no yawn, closed, and open. It displays the quantity of accurate and inaccurate forecasts. The model correctly predicts 55 out of 63 yawn instances, 69 out of 74 no yawn instances, 212 out of 215 closed instances, and 224 out of 226 open instances. Misclassifications include 7 yawns as no yawn, 5 no yawns as yawn, 3 closed as open, and 2 open as closed. The matrix indicates strong performance overall, with particularly high accuracy in distinguishing closed and open states.

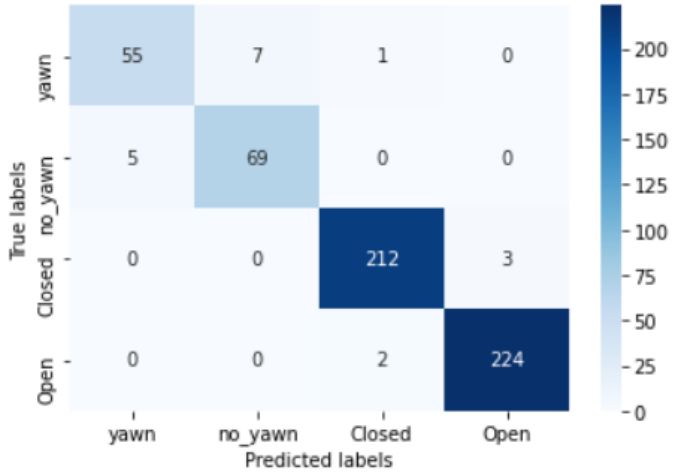

Fig. 4. Confusion matrix

C. Training and Validation Loss Curve

The value of the loss function—typically cross-entropy loss—on the training and validation datasets during a 50-epoch period is displayed as a loss curve in "Fig.5". When the model gets better at fitting the training data, the training loss (blue line) gets smaller over the course of epochs.

Epochs = 50
Validation Loss = 0.10
Training Loss = 0.12

D. Training and Validation Accuracy Curve

The accuracy curve displayed in "Fig.6" illustrates how well the model performed throughout 50 epochs on the training and validation datasets. How well the model predicts the training data is shown by the blue line, which is the training accuracy. The validation accuracy, or how effectively the model generalizes to new data, is represented by the red line.

Epochs = 50
Validation Accuracy = 0.97
Training Accuracy = 0.96

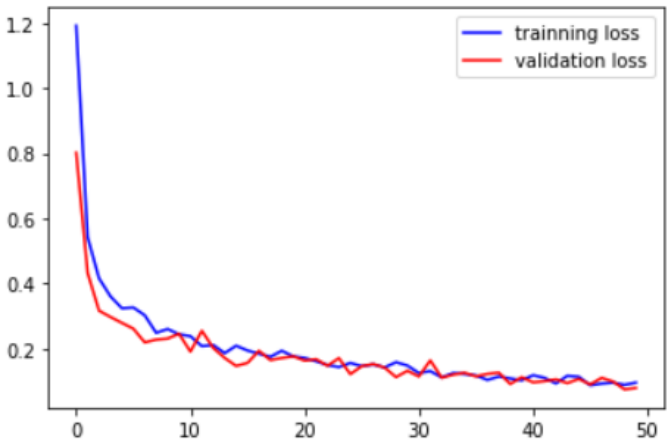

Fig. 5. Training and Validation Loss Curve

Fig. 6. Training and Validation Accuracy Curve

E. Precision-Recall Curve

The precision-recall curve shown in "Fig.7", indicates high performance of all model classes. For the yawn and no_yawn classes, the model achieves near-perfect precision and recall, with Average Precision (AP) values of 0.97 and 0.98, respectively. It achieves perfect recall and precision for both closed and open classes (AP = 1.00). This demonstrates that the model has low error rates and can accurately differentiate between these classes.

Average Precision of Class yawn = 0.97
Average Precision of Class no_yawn = 0.98
Average Precision of Class Closed = 1.00
Average Precision of Class Open = 1.00

F. Performance Analysis

"Table.II" demonstrates that after 50 epochs, the proposed model achieved training accuracy of 96.36%, validation accuracy of 97.75%, and testing accuracy of 97%.

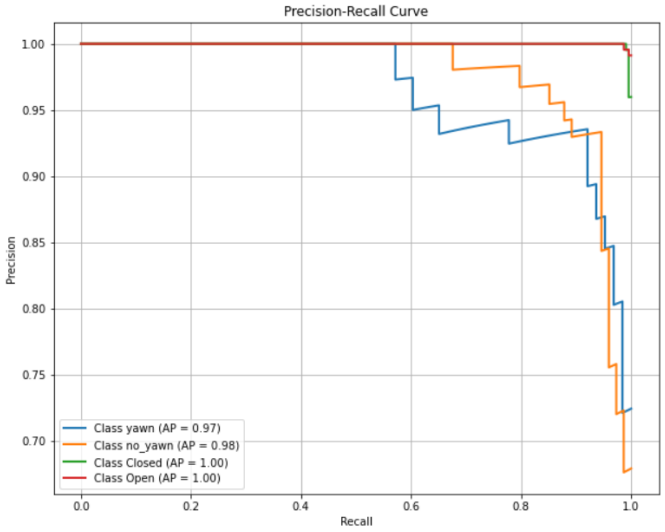

Fig. 7. Precision-Recall Curve

Training Accuracy = 0.96
Validation Accuracy = 0.97
Testing Accuracy = 0.97

TABLE II
ACCURACY OF PROPOSED MODEL

Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)
96.36 %	97.75 %	97 %

CONCLUSION

To sum up, driver drowsiness is a serious problem that has a big influence on road safety and causes a lot of accidents every year. To reduce these concerns, sleepiness detection technologies must be developed with accuracy and efficiency. By implementing Convolutional Neural Networks (CNNs) when combined with real-time video processing, these systems are able to evaluate facial characteristics and eye closure in order to promptly send alerts that improve driver awareness and lower the risk of accidents.

REFERENCES

[1] A. Bener, E. Yildirim, T. Özkan, and T. Lajunen, "Driver sleepiness, fatigue, careless behavior and risk of motor vehicle crash and injury: Population based case and control study," *Journal of Traffic and Transportation engineering (English edition)*, vol. 4, no. 5, pp. 496–502, 2017.

[2] National Highway Traffic Safety Administration (NHTSA), "Drowsy driving in fatal crashes, united states, 2017–2021," 2021. Accessed: 2024-07-28.

[3] M. H. Smolensky, L. Di Milia, M. M. Ohayon, and P. Philip, "Sleep disorders, medical conditions, and road accident risk," *Accident Analysis & Prevention*, vol. 43, no. 2, pp. 533–548, 2011.

[4] I. Nasri, M. Karrouchi, K. Kassmi, and A. Messaoudi, "A review of driver drowsiness detection systems: Techniques, advantages and limitations," 2022.

[5] 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.

[6] N. S. Foundation, "National sleep foundation's drowsy driving survey," tech. rep., Institution Name, 2023. Accessed: 2024-06-26.

[7] S. K. Datta, R. P. F. Da Costa, J. Härri, and C. Bonnet, "Integrating connected vehicles in internet of things ecosystems: Challenges and solutions," in *2016 IEEE 17th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*, pp. 1–6, IEEE, 2016.

[8] A. S. N. M. M. M. Mardi, Zahra, "Detection of driver drowsiness using eeg signals," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 1, pp. 12–16, 2011.

[9] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," in *2010 International Conference on Machine and Web Intelligence*, pp. 230–233, 2010.

[10] J. Cui, Z. Lan, Y. Liu, R. Li, F. Li, O. Sourina, and W. Müller-Wittig, "A compact and interpretable convolutional neural network for cross-subject driver drowsiness detection from single-channel eeg," *Methods*, vol. 202, pp. 173–184, 2022.

[11] K. Dwivedi, K. Biswaranjan, and A. Sethi, "Drowsy driver detection using representation learning," in *2014 IEEE International Advance Computing Conference (IACC)*, pp. 995–999, 2014.

[12] A. Picot, S. Charbonnier, and A. Caplier, "On-line detection of drowsiness using brain and visual information," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 764–775, 2012.

[13] S. Said, S. Alkork, T. Beyrouthy, M. Hassan, O. E. Abdellatif, and M. F. Abdraboo, "Real Time Eye Tracking and Detection- A Driving Assistance System," *Advances in Science, Technology and Engineering Systems Journal*, vol. 3, no. 6, pp. 446–454, 2018.

[14] S. Park, F. Pan, S. Kang, and C. Yoo, "Driver drowsiness detection system based on feature representation learning using various deep networks," in *Computer Vision – ACCV 2016 Workshops* (C. Chen, J. Lu, and K. Ma, eds.), vol. 10118 of *Lecture Notes in Computer Science*, pp. 147–160, Springer, Cham, 2017.

[15] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Real-time driver drowsiness detection for android application using deep neural networks techniques," *Procedia Computer Science*, vol. 130, pp. 400–407, 2018. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.

[16] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Real-time driver drowsiness detection for android application using deep neural networks techniques," *Procedia Computer Science*, vol. 130, pp. 400–407, 01 2018.

[17] B. Reddy, Y.-H. Kim, S. Yun, C. Seo, and J. Jang, "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[18] kaggle, "Eyes and mouth dataset." https://www.kaggle.com/datasets/serenaraju/yawn-eye-dataset-new.

[19] A. Tam, "Using haar cascade for object detection." https://machinelearningmastery.com/using-haar-cascade-for-object-detection/, Year. Accessed: July 12, 2024.

[20] J. Rajevenceltha, V. H. Gaidhane, and V. Anjana, "A novel approach for drowsiness detection using local binary patterns and histogram of gradients," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp. 1–6, IEEE, 2019.

[21] W. Berger, "Deep learning haar cascade explained." https://www.willberger.org/cascade-haar-explained/, Year. Accessed: June 23, 2024.

[22] Z. Yan, V. Jagadeesh, D. DeCoste, W. Di, and R. Piramuthu, "Hd-cnn: Hierarchical deep convolutional neural network for image classification," in *International Conference on Computer Vision (ICCV)*, vol. 2, pp. 435–443, Citeseer, 2015.

[23] A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–5, 2020.

[24] S. Boumaraf, X. Liu, Y. Wan, Z. Zheng, C. Ferkous, X. Ma, Z. Li, and D. Bardou, "Conventional machine learning versus deep learning for magnification dependent histopathological breast cancer image classification: A comparative study with visual explanation," *Diagnostics*, vol. 11, no. 3, p. 528, 2021.

[25] Y. Shu, *Deep convolutional neural networks for object extraction from high spatial resolution remotely sensed imagery*. PhD thesis, University of Waterloo, 2014.

[26] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, 2023.

[27] M. Mishra, "Convolutional neural networks." https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939, 2019.

[28] Y. Guo, Y. Liu, E. M. Bakker, Y. Guo, and M. S. Lew, "Cnn-rnn: a large-scale hierarchical image classification framework," *Multimedia tools and applications*, vol. 77, no. 8, pp. 10251–10271, 2018.

[29] M. Browne and S. S. Ghidary, "Convolutional neural networks for image processing: an application in robot vision," in *Australasian Joint Conference on Artificial Intelligence*, pp. 641–652, Springer, 2003.