International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Vision-Based Driver Drowsiness Detection Using Convolutional Neural Networks: Performance Evaluation and Comparative Analysis

Dr Meenakshi Garg

Received: 02/11/2024 **Revised:** 17/12/2024 **Accepted:** 29/12/2024

Abstract: The global concern over road accidents due to driver fatigue high-lights the need for efficient detection technologies to improve road safety. The practical constraints of traditional methods that rely on physiological signals have prompted researchers to investigate the use of convolutional neural net-works (CNNs) for the purpose of efficient detection of drowsiness. This paper describes a detailed approach to identifying driver fatigue using a convolutional neural network (CNN). The dataset used includes annotated photos of drivers' faces and Haar cascade eyes for accurate model construction and evaluation. Many data preprocessing methods, such as scaling, normalization, and data augmentation, are used to enhance the performance and generalizability of model. The model architecture consists of various layers for effective feature extraction and classification. The process of training entails the adjustment of hyperparameters, such as a learning rate scheduler, Adam optimizer, and early stopping mechanism, to enhance model convergence and mitigate the risk of overfitting. Furthermore, a comprehensive review of previous studies on detect-ing driver drowsiness with convolutional neural networks (CNNs) shows vary-ing levels of accuracy, ranging between 80% and 98%. Expanding upon the findings, our research introduces a resilient convolutional neural network (CNN) methodology that exhibits a remarkable accuracy rate of 99%. This out-come underscores the considerable potential for enhancing road safety.

Keywords: CNN Model, Driver Drowsiness Detection, Driver Fatigue, Haar Cascade Eyes, Comparative Analysis, Road Safety, IEEE Dataport.

Introduction

Recently there has been a growing concern, about the increase in road accidents caused by drivers being drowsy. The World Health Organization estimates that around 1.35 million people die each year in road traffic accidents with drowsy driving being a factor [1]. Driver fatigue, lack of sleep and long hours behind the wheel are contributors to this risk to road safety. Because of this there is a lot of interest from both researchers and the automotive industry in developing systems to detect driver drowsiness.

Traditional methods of detecting drowsy drivers often rely on signals such as EEG (electroencephalography), EOG (electrooculography) and EMG (electromyography).

[1]*Vivekanand Education Society's Institute of Technology (VESIT), Mumbai*
meenakshi.garg@ves.ac.in

Inspired By How Animals' Visual Cortex Works Cnns Have Demonstrated Perfor-Mance, In Tasks Related To Image Classification And Recognition. Using Their Abilities In Feature Extraction And Learning Cnns Are Great, At Recognizing Complex Patterns And Spatial Relationships In Images Making Them Perfect For Examining Driver Behavior Recorded By, In Car Cameras [3]. The Incorporation Of Cnns In Detecting Driver Drows-Iness Holds Promise In Enhancing Road Safety Through Real-Time Monitoring And Alert Systems.

A Key Advantage Of Using Cnn Based Drowsiness Detection Systems Is Their Ability To Automatically Extract Features From Visual Inputs Without The Need For Manual Fea-Ture Engineering. By Employing Convolution, Pooling, And Nonlinear Activation Func-Tions Cnns Effectively Capture Temporal Characteristics From Facial Expressions, Eye Movements And Head Gestures That Indicate Drowsiness [4]. This Inherent Flexibility Allows For The Creation Of Adaptive Drowsiness Detection Systems Of

Accommodating Various Driving Scenarios And Individual Differences.

Furthermore, Cnn Based Drowsiness Detection Systems Can Analyze Data Instanta-Neously Making Them Well Suited For Integration, Into Driver Assistance Systems (Adas). By Observing Driver Actions And Facial Expressions Cnns Can Promptly Detect Signs Of Drowsiness. Issue Timely Alerts Or Interventions To Prevent Potential Accidents. Integrating Cnn Based Detection Systems, Into Vehicles Can Help Reduce The Risks Linked To Driver Fatigue And Significantly Decrease The Number Of Accidents Caused By Drowsy Driving [5].

1 Literature Review

In Their Research Hossain Et Al. [6] Created A Cnn Framework To Identify Driver Dis-Traction Behaviors. The Models They Used Were Simple Cnn, Vgg 16, Resnet50 And Mobilenetv2. After Conducting Experiments Simple Cnn Achieved An Accuracy Of 97.45% Resnet50 Achieved 94.50% Accuracy And Mobilenetv2 Reached 98.12% Accuracy. These Results Highlight The Potential Of Implementing Real Time Driver Dis-Traction Detection Systems. Future Plans Involve Integrating The Model Into An Android Application, For Use In Real Time Scenarios. Although The Study Did Not Include Roc Curve Analysis It Did Feature Accuracy And Loss Curves. Additionally, The Confusion Matrix Demonstrates Promising Outcomes Compared To Studies. This Investigation Represents An Advancement In Driver Safety Technology Paving The Way, For Improved Real Time Monitoring And Prevention Of Driving Incidents.

In A Study Mohit Et Al. [7] Proposed A System, For Detecting Driver Drowsiness By Utilizing Four Learning Models; Flowimagenet, Resnet, Vgg Facenet And Alexnet. These Algorithms Analyze Factors Such As Head Movements, Hand Gestures, Facial Ex-Pressions, And Behavioral Cues From Rgb Videos To Detect Signs Of Sleepiness. The Models Are Combined Using A Method After Categorizing The Features Into Non Sleepi-Ness Drowsiness With Eye Blinking, Yawning And Nodding. A Softmax Classifier Is Then Used To Generate The Result. Although Achieving An 85% Accuracy Rate This System Falls Short In Comparison To Studies. Critically The Research Lacks An Evaluation Of The Re-Sults Which Raises Doubts, About The Effectiveness Of This Approach.

In This Research Paper Ahmed Et Al. [8] Introduces A Method, For Identifying Driving Behaviors By Using A Deep Learning-Based Algorithm. It Shows Promising Outcomes For Real Time Detection Of Driver Distractions. Sets A Standard In This Area. The Resulting System For Detecting Driver Drowsiness Achieved An Accuracy Rate Of 97.23%. The Cnn Model Utilizes Three Activation Functions; Rectified Linear Unit (Relu) Sigmoid And Softmax. The Network Structure Was Tested Times To Improve Its Ability To Extract Features And Classify Accurately. Performance Evaluation Of The Model Was Done Using Accuracy And Loss Curves Revealing No Issues, With Overfitting Or Underfitting. Future Developments Will Focus On Exploring The Benefits Of Combining Cnn With Types Of Neural Networks, Which Could Lead To Significant Advancements In This Field.

Based On The Results Of This Study [9] Incorporating Intelligence (Ai) Into Systems Designed To Detect Driver Drowsiness Marks An Advancement, In The Digital Age. Ai, Powered By Machine Learning (Ml) Algorithms Plays A Role In Identifying Tiredness Related Impairments And Assisting In Decision Making Processes To Ensure Drivers Re-Main Alert While Behind The Wheel. To Improve The Efficiency Of Detection Systems Driven By Ai It Is Important To Take An Approach That Addresses Challenges Such As Obtaining Real Time Data And Considering Various Factors. Recognizing The Effective-Ness Of Ai In Reducing Risks Associated With Drowsy Driving Represents A Step, In En-Hancing Road Safety Through Progress.

2 Proposed Model

The Proposed Model Uses A Convolutional Neural Network (Cnn) Architecture To Effec-Tively Detect Driver Drowsiness. It Streamlines Image Classification Tasks Through A Systematic Process. It Consists Of Several Phases. It Starts With Video Frame Acquisition And Preparing The Dataset. Then Loading And Preprocessing Images From The Dataset, Comprising Classes Like 'Closed,' 'Open,"No_Yawn,' And 'Yawn.' After That, The Model Architecture Was Established, Consisting Of Fully Connected Layers For Classification And Convolutional Layers For Feature Extraction. Training Involves Data Augmentation To Enhance Dataset Diversity And A Custom Learning Rate Schedule For

Optimization. Vali-Dation Was Conducted To Monitor And Prevent Overfitting.

2.1 Dataset Overview

The Dataset Used In The Model Includes Images Of Drivers' Faces And Haar Cascade Eyes, Totaling 2900 Pictures Distributed Across Four Categories ('Closed', 'Open', 'No_Yawn', 'Yawn'). Each Image In The Dataset Is Annotated And Sorted Into One Of The Four Classes, Permitting Comprehensive Model Training And Evaluation. This Dataset Has Been Obtained From The Ieee Dataport [10] And Is Publicly Available, Ensuring Transparency And Reproducibility In Research. It Is Made Up Of Clips From An In-Car Camera That Show Drivers Doing Different Expressions And Actions, Like Talking, Sing-Ing, Yawning, And Remaining Silent. The Films Are Shot In A Variety Of Lighting Condi-Tions, Including Natural Ones. Table 1 Offers A Description Of The Dataset.

Table 1. Description Of The Dataset Used.

Class Names	Description	Number Of Samples
Closed	Images Of Eyes Closed	726
Open	Images Of Eyes Open	726
No_Yawn	Images Of Driver In Normal State	725
Yawn	Images Of Driver In Yawning State	723

2.2 Data Loading Process

Initially, 32, 255, And 3 Values Were Assigned To The Constants For Batch Size, Picture Size, And Channels, Respectively. The Code Retrieves The Dataset From The Specified Directory By Utilizing Tensorflow's Image_Dataset_From_Directory Function. It Ensures Repeatability By Rearranging The Dataset And Setting The Seed To 123. For Training, Im-Ages Are Scaled To The Required Size (255×255) And Arranged Into Batches Of 32. This Method Simplifies The Loading Of Picture Datasets, Making Model Training And Prepro-Cessing More Effective. By Offering A Uniform Input Format, This Configuration Makes It Easier To Complete Later Stages Of The Machine Learning Pipeline, Such As Model Con-Struction, Training, And Evaluation [11].

The Set Of Images From The Dataset Is Displayed Using Matplotlib [12], Along With Their Corresponding Labels. The Program Sequentially Processes The Initial Images And Their Corresponding Labels Retrieved From The Dataset. It Then Presents Them In A Grid Arrangement With 3 Rows And 4 Columns. Fig. 1. Shows Samples Of The Images From The Dataset.

Fig. 1. Samples Of Images From The Dataset.

2.3 Dataset Partitioning And Splitting

By Using The 'Get_Split_Data' Function The Dataset Gets Split Into Training, Validation, And Testing Sets. We Can Specify Parameters, Like The Dataset Itself (Ds) The Proportions

For Training, Testing And Validation Sets (Train_Split, Test_Split, Val_Split) And Whether To Shuffle The Data (Shuffle). The Function Also Checks That The Sum Of Proportions Equals 1 To Ensure Division. It Calculates The Size. Shuffles It If Needed. This Makes It Easier To Prep Data For Machine Learning Tasks By Ensuring Distribution, For Training, Validation, And Testing Purposes.

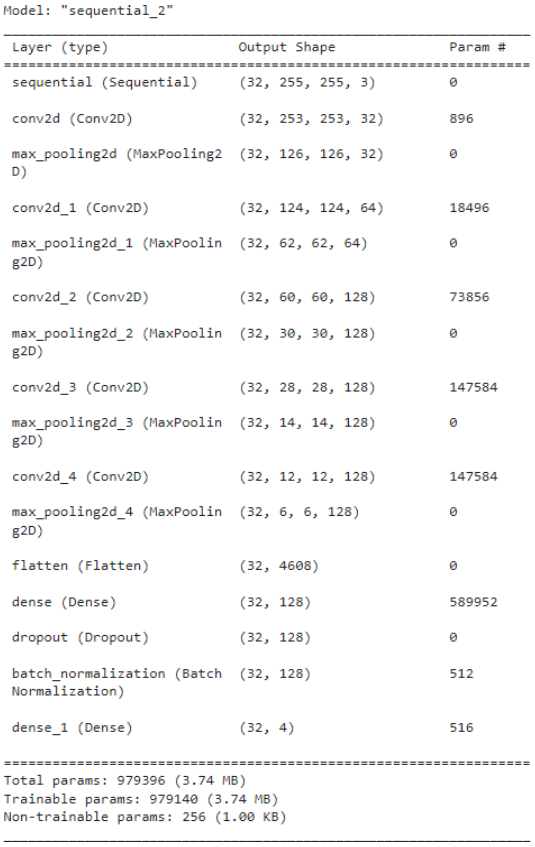
2.4 Data Preprocessing

In Data Preprocessing, Commonly Used Image Preprocessing Approaches In Deep Learn-Ing Pipelines, Specifically For Image Classification Tasks, Were Implemented. Firstly, Scaling And Normalization Are Necessary For Preparing Images To Be Input Into A Ma-Chine Learning Model. The Code Uses Tensorflow's Sequential Api To Construct A Suc-Cession Of Resizing And Rescaling Layers. Resizing Guarantees That All Images Are Of The Same Size (Given By Image_Size), Making Them Consistent For The Model. Rescaling Then Ensures That Pixel Values In The Images Are Scaled Down To A Range Between 0 And 1, Which Assists In Stabilizing And Speeding Up The Training Process. Second, By Mak-Ing Arbitrary Adjustments To The Photos, A Technique, Data Augmentation Is Applied To Boost The Diversity Of The Training Dataset. This Increases The Model's Capacity To Gen-Eralize And Makes It More Tolerant To Changes In The Input Data. The Method Utilizes Data Augmentation Techniques, Including Random Flipping Horizontally And Vertically, As Well As Random Rotation, To The Training Dataset.

These Preprocessing Processes Are Critical For Increasing The Performance And Gen-Eralization Of Machine Learning Models, Especially In Tasks Like Image Classification.

2.5 Model Architecture

The Model Architecture Comprises Various Layers, Including Convolutional, Max-Pooling, Flattening, Dense, Dropout Regularization, And Batch Normalization. These Layers Work Together To Process Input Images (255x255 Pixels With 3 Color Channels) For Making Predictions. Max-Pooling Layers Minimize Spatial Dimensions And Prevent Overfitting, While Convolutional Layers Apply Filters To Extract Features Like Edges And Forms. Dense Layers At The End Perform Classification Based On Learned Features. Drop-Out Regularization And Batch Normalization Improve The Model’s Generalization And Stability. Additionally, An Early Stopping Callback Is Used To Monitor Validation Loss And Restore Optimal Weights. The Architecture Incorporates Relu Activation Functions In Layers With Filter Sizes (32, 64 128). Feature Maps Are Converted Into An Array By A Flattening Layer Followed By A Layer Of 128 Neurons That Uses Relu For Non-Linear Mappings. Overfitting Is Avoided By Dropout Regularization, And Better Convergence Is Achieved By Batch Normalization, Which Standardizes Activations. To Ensure Accurate Predictions Across Classes, The Output Layer Consists Of A Dense Layer With Four Neurons And Softmax Activation For Multi-Class Classification. The Summary Of The Model Can Be Seen In Fig. 2.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 255, 255, 3)         0

 conv2d (Conv2D)             (32, 253, 253, 32)        896

 max_pooling2d (MaxPooling2  (32, 126, 126, 32)        0
 D)

 conv2d_1 (Conv2D)           (32, 124, 124, 64)        18496

 max_pooling2d_1 (MaxPoolin  (32, 62, 62, 64)          0
 g2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 128)         73856

 max_pooling2d_2 (MaxPoolin  (32, 30, 30, 128)         0
 g2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 128)         147584

 max_pooling2d_3 (MaxPoolin  (32, 14, 14, 128)         0
 g2D)

 conv2d_4 (Conv2D)           (32, 12, 12, 128)         147584

 max_pooling2d_4 (MaxPoolin  (32, 6, 6, 128)           0
 g2D)

 flatten (Flatten)           (32, 4608)                0

 dense (Dense)               (32, 128)                 589952

 dropout (Dropout)           (32, 128)                 0

 batch_normalization (Batch  (32, 128)                 512
 Normalization)

 dense_1 (Dense)             (32, 4)                   516

=================================================================
Total params: 979396 (3.74 MB)
Trainable params: 979140 (3.74 MB)
Non-trainable params: 256 (1.00 KB)
_________________________________________________________________
```

Fig. 2. Model Summary

2.6 Role Of Activation Functions In The Cnn Model

In The Presented Cnn Model, Two Main Activation Functions Are Employed: Relu (Rectified Linear Unit) And Softmax [13]. Relu Is Applied Within The Convolutional And Dense Layers, Serving To Introduce Non-Linearity By Outputting The Input If It Is Positive And 0 Otherwise. This Activation Function Aids In Overcoming The Vanishing Gradient Problem And Increases Training Convergence Due To Its Simplicity And Effec-Tiveness [14]. Softmax, On The Other Hand, Is Implemented In The Output Layer To Con-Vert Raw Scores Into Probabilities, Where Each Output Node Reflects The Probability Of The Corresponding Class. This Makes It Suitable For Multi-Class Classification Jobs Be-Cause It Guarantees That The Sum Of Probability For All Classes Equals One [15]. These Activation Functions Are Crucial In The Cnn Model, Facilitating Non-Linear Transfor-Mations Between Input And Output And Generating Meaningful Predictions For Multi-Class Classification Tasks.

2.7 Model Training

A Learning Rate Scheduler, With Decay Was Utilized To Adjust The Learning Rate Over 100 Epochs For Model Optimization. This Method Aids In Achieving A Convergence Towards The Best Solution. The Schedule For Learning Rate Adjustment Is Then Applied To Set Up The Adam Optimizer For Updating The Models' Weights During Training. Furthermore, Data Augmentation Methods Were Applied To The Training Dataset Using Techniques Like Random Flipping, Rotation, Zooming And Adjusting Contrast. These Strategies Help Di-Versify The Training Data And Improve The Model's Ability To Generalize To Data. The Model Is Then Constructed With An Optimizer, A Loss Function, And An Evaluation Metric. It Undergoes Training, On The Training Dataset Using The Method And Its Performance Is Evaluated On The Validation Dataset. To Minimize Loss And Enhance Accuracy Adjust-Ments Are Made To The Models' Parameters Throughout Each Iteration Of Epochs During Training.

2.8 Hyper-Parameters Used

The Hyperparameters Utilized For Training Include The Batch Size, Set At 32 To Find A Balance, Between Speed And Memory Usage. Images Are Resized To 255x255 Pixels To Keep An Input Size. The Model Is Designed To Handle Rgb Images With Three Color Channels. With A Multi-Class Classification Job, Model Is Structured With Four Output Nodes, Each Representing A Class. Training Is Optimized Using The Adam Optimizer With A Learning Rate Scheduler Starting At 0.001 For Adjustments. The Training Process Spans 100 Epochs To Allow Learning From The Dataset. To Prevent Overfitting Early Stopping Is Implemented With A Patience Of 3 Epochs Stopping Training If Validation Loss Does Not Improve Steadily. These Hyperparameters Together Impact How The Model Learns, Adapts, And Performs During Training. Table 2. Provides A Description Of The Hyperpa-Rameters Used.

Table 2. Description Of Hyper-Parameters Used.

Hyper-Parameter	Value
Batch Size	32
Image Size	255 × 255
Number Of Channels	3 (Rgb)
Number Of Classes	4
Optimizer	Adam
Learning Rate	0.001
Number Of Epochs	100
Early Stopping Patience	3

2.9 Visualizing The Training And Validation Performance

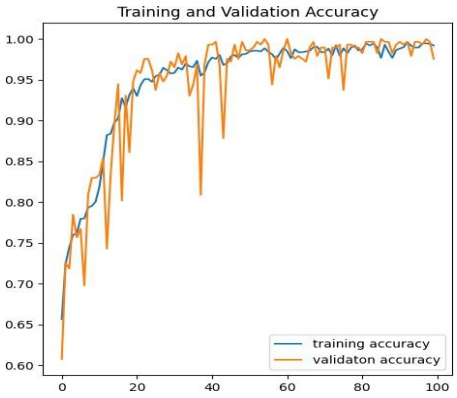


Fig. 3. Accuracy Graph

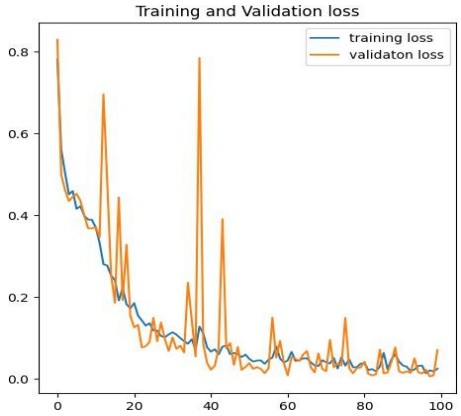


Fig. 4. Loss Graph

The Graph, In Figure 3 Illustrates The Accuracy Of The Model While Figure 4 Displays The Loss Graph. Additionally, There Is A Discussion About The Accuracy And Loss Con-Cerning Both The Training And Validation Data.

Training Loss And Accuracy

Throughout The Epochs There Is A Decrease In Training Loss Indicating That The Model Is Learning Information And Improving Its Predictions Based On The Training Data. Simul-Taneously The Training Accuracy Progressively Increases Over Time Suggesting That The Models' Predictions On The Training Data Are Becoming More Accurate.

Validation Loss And Accuracy

Initially Validation Loss And Accuracy Show Fluctuations But Generally Follow Patterns As Seen In Training Metrics. However, There Are Deviations In Validation Measures, Which May Indicate Issues Like Overfitting Or A Necessity For Optimization.

To Enhance The Model's Performance Various Techniques Such As Dropout Regulari-Zation, Batch Normalization Or Early Stopping Can Be Implemented To Address Overfit-Ting And Improve Adaptability To Data.

Epochs And Training Time

The Duration Of Each Epochs Training Varies But Typically Decreases As Epochs Pro-Gress—A Practice, In Deep Learning Methodologies. Factors Influencing Training Time Include Model Complexity, Dataset Size And Available Hardware Resources.

Although The Model Shows Progress, In Both Training And Validation Accuracy At First There Are Fluctuations In The Validation Metrics Suggesting The Necessity, For Ad-Justments And Potentially Reducing Overfitting.

2.10 Experimental Results

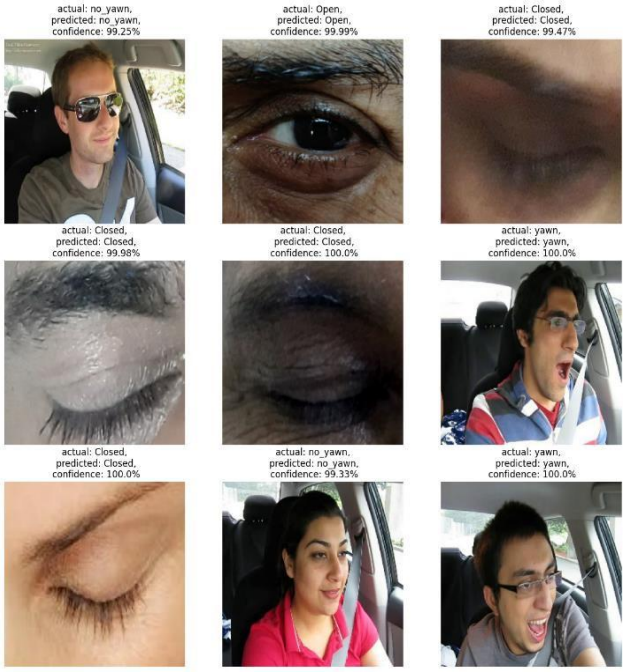


Fig. 5. Predictions Of The Model Made On The Test Dataset.

Fig. 5 Shows The Models Predictions On A Set Of Photos, From The Test Dataset Along With The Predicted Class And Confidence Level For Each Prediction. By Using The Func-Tion For Each Image, The Model Identifies Both The Anticipated Class And Its Confidence Level. The Title Of Each Subplot Indicates The Class Of The Image. The Confidence Level Acts As A Metric To Assess How Well The Model Categorizes Images Into Classes Offering Insights, Into Its Classification Accuracy.

2.11 Confusion Matrix

The Confusion Matrix Shows How Well A Machine-Learning Model Performs In Different Classes [16]. The Anticipated Class Is Represented By Each Column, And The Actual Class Is Represented By Each Row. The Confusion Matrix Displays A Larger Count Of Values Along The Diagonal, Suggesting Accurate Predictions, While The Off-Diagonal Values Signify Misclassifications. By Evaluating The Entries In This Matrix, Many Performance Aspects May Be Determined. The Performance Of The Model Can Be Gauged By Analyz-Ing Metrics Such, As Accuracy, Recall And F1 Score. Furthermore, The Accuracy Rating Can Be Derived By Studying The Data From The Confusion Matrix Along, With Its Visuali-Zation. Figure 6 Depicts The Confusion Matrix.

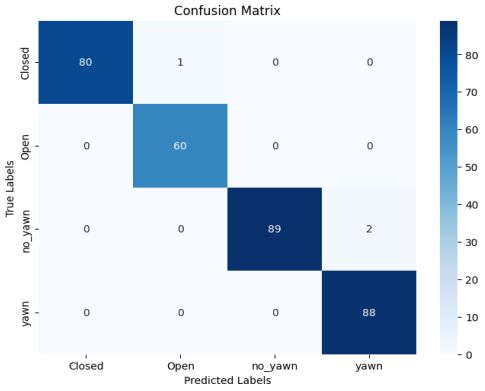


Fig. 6. Heat Map Illustrating The Confusion Matrix, Demonstrating The Performance Of The Model.

2.12 Classification Report

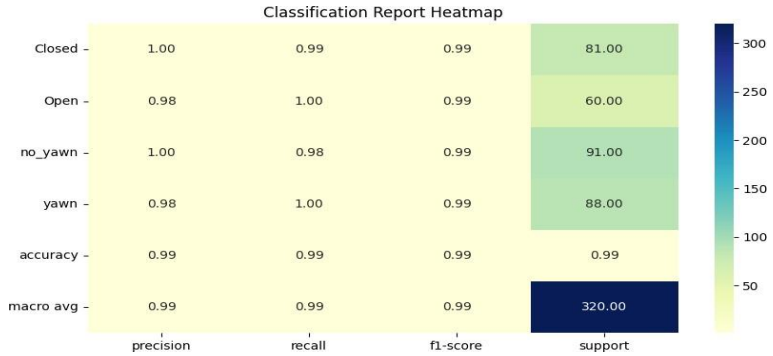


Fig. 7. Heat Map Illustrating The Classification Report Of The Model.

The Fig. 7. Displays The Classification Report For The Model. The Classification Report For The Model Offers A Thorough Assessment Of Its Performance Across Multiple Classes. For The "Closed" Class, The Accuracy Is Outstanding At 1.00 Indicating That All Instances Classified As "Closed" Were Correct With A Recall Rate Of 0.99, Showing That The Model Successfully Identified Most Cases In This Category. The F1 Score, Which Balances Be-Tween Recall And Accuracy Is Also Strong At 0.99. Similar Impressive Accuracy, Recall And F1 Score Values Are Observed For The "Open," "No_Yawn," And "Yawn" Categories, Demonstrating Good Performance Across All Categories. The Overall Accuracy Of The Model Stands At 0.99 Showcasing Its Ability To Correctly Classify Instances From The Dataset. The Average F1 Score Across All Categories Is Also High, At 0.99 Indicating A Good Overall Performance. Additionally, There Is A Support Value Of 320 Denoting The Number Of Cases In The Dataset.

3 Comparison

Table 3. Comparison Of Accuracy

Author	Accuracy Achieved Using Cnn
Suresh Et Al. [17]	93.6%
Li Et Al. [18]	80%
Ahmed Et Al. [19]	85%
Jahan Et Al. [20]	93% To 98%
Ch Et Al. [21]	87%
Ahmed Et Al. [8]	97.23%
Proposed Model	99%

Table 3 Presents A Comparison Of The Accuracies Achieved In Past Studies On Detecting Driver Drowsiness Using Convolutional Neural Networks (Cnns).

Different Studies Have Shown Varying Levels Of Accuracy. For Instance, Suresh Et Al. (2023) Reported That Xception Performed Better Than Deep Convolutional Neural Net-Work (Dcnn) Models Achieving A Validation Accuracy Of 93.6% And A Validation Loss Of 16.9%. Li Et Al. (2019) Achieved Over 80% Accuracy While Ahmed Et Al. (2020) Reached 85% Accuracy. Jahan Et Al. (2018) Demonstrated Accuracy Ranging From 93% To 98%. Ch Et Al. (2023) Obtained An 87% Accuracy Using The Inception V3 Model For Detecting Driver Sleepiness. On The Hand Ahmed Et Al. (2023) Utilized A Cnn Architec-Ture. Significantly Improved Results, With An Average Accuracy Of 97.23%. While Pre-Vious Studies Have Shown Results With Levels Of Accuracy Our Research Aims To Add To This Knowledge Base By Enhancing Detection Effectiveness And Achieving An Outstand-Ing Accuracy Rate Of 99% Surpassing Earlier Findings.

4 Conclusion And Future Scope

This Study Introduces A Model Using A Neural Network (Cnn) Design To Effectively Detect Driver Drowsiness. The Approach Involves Gathering Video Frames, Loading, And Processing The Dataset And Dividing It Systematically. It Incorporates Connected Layers, For Classification And Convolutional Layers For Extracting Features, Utilizing Techniques Such As Data Augmentation And A Unique Learning Rate Schedule. The Model Leverages An Annotated Dataset From The Ieee Dataport, Producing Promising Results With An Astounding Accuracy Of 99%. By Visualizing Training And Validation Data Along With Conducting An Analysis Using A Confusion Matrix And Classification Report We Demon-Strate The Effectiveness Of Our Model. Comparative Analysis With Prior Studies Shows An Enhancement In Accuracy Levels. The Implementation Of This Model Holds Promise For Enhancing Road Safety By Alerting Drivers To Signs Of Fatigue. Future Studies Could Concentrate On Improving The Model's Performance And Exploring Optimization Strate-Gies To Enhance Its Practicality, In Real World Scenarios.

References

1. *Global Status Report On Road Safety 2018*. (2018, June 17). Https://Www.Who.Int/Publications/I/Item/9789241565684

2. Lal Sk, Craig A. A Critical Review Of The Psychophysiology Of Driver Fatigue. Biol Psychol. 2001 Feb;55(3):173-94. Doi: 10.1016/S0301-0511(00)00085-5. Pmid: 11240213.Lecun
Y, Bengio Y, Hinton G. Deep Learning. Nature. 2015;521(7553):436-444.
3. Lecun, Y., Bengio, Y. & Hinton, G. Deep Learning. *Nature* **521**, 436–444 (2015). Https://Doi.Org/10.1038/Nature14539
4. Ch, Venkata & Reddy, U. Srinivasulu & Kishorekolli, Venkata. (2019). Deep Cnn: A Machine Learning Approach For Driver Drowsiness Detection Based On Eye State. Revue D'intelligence Artificielle. 33. 461-466. 10.18280/Ria.330609.
5. Xu J, Jiang Y, Yu L, Et Al. Real-Time Driver Drowsiness Detection Based On Convolutional Neural Networks. Ieee Transactions On Intelligent Transportation Systems. 2017;19(7):2204-2214.
6. Hossain, Md & Rahman, Md & Islam, Manowarul & Akhter, Arnisha & Uddin, Md Ash-Raf & Paul, Bikash Kumar. (2022). Automatic Driver Distraction Detection Using Deep Convolutional Neural Networks. Intelligent Systems With Applications. 14. 200075. 10.1016/J.Iswa.2022.200075.
7. Mohit Dua, Shakshi, Ritu Singla, Saumya Raj, And Arti Jangra. 2021. Deep Cnn Models-Based Ensemble Approach To Driver Drowsiness Detection. Neural Comput. Appl. 33, 8 (Apr 2021), 3155–3168.
8. Ahmed, T., Jyoti, O., & Mou, T. H. (2023, December 13). Drivers' Drowsiness Detection System Enhancement Using Deep Learning: Cnn-Based Approach. *2023 26th Interna-Tional Conference On Computer And Information Technology (Iccit)*.
9. Garg, Meenakshi, And Heramb Shankar Patil. "Comparative Study On Ai In Men-Tal Health And Wellbeing–Current Applications And Trends." *International Research Jour-Nal Of Engineering And Technology* (2021).
10. Shabnam Abtahi, Mona Omidyeganeh, Shervin Shirmohammadi, Behnoosh Hariri, August 1, 2020, "Yawdd: Yawning Detection Dataset", Ieee Dataport, Doi: Https://Dx.Doi.Org/10.21227/E1qm-Hb90.
11. Tensorflow Authors. "Tensorflow: Large-Scale Machine Learning On Heterogeneous Sys-Tems." 2015. Software Available From Tensorflow.Org.
12. Hunter, J.D. "Matplotlib: A 2d Graphics Environment." Computing In Science & Engi-Neering, Vol. 9, No. 3, Pp. 90-95, 2007.
13. Maas, A.L., Hannun, A.Y. And Ng, A.Y. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. Proceedings Of The 30th International Conference On Machine Learning, Vol. 28, 3. Https://Ai.Stanford.Edu/~Amaas/Papers/Relu_Hybrid_Icml2013_Final.Pdf
14. Bridle, J.S. (1990). Probabilistic Interpretation Of Feedforward Classification Network Outputs, With Relationships To Statistical Pattern Recognition. In: Soulié, F.F., Hérault, J. (Eds) Neurocomputing. Nato Asi Series, Vol 68. Springer, Berlin, Heidelberg. Https://Doi.Org/10.1007/978-3-642-76153-9_28
15. Wang, Y.; Li, Y.; Song, Y.; Rong, X. The Influence Of The Activation Function In A Con-Volution Neural Network Model Of Facial Expression Recognition. Appl. Sci. 2020, 10, 1897.
16. Markoulidakis, I.; Rallis, I.; Georgoulas, I.; Kopsiaftis, G.; Doulamis, A.; Doulamis, N. Multiclass Confusion Matrix Reduction Method And Its Application On Net Promoter Score Classification Problem. Technologies 2021, 9, 8
Drowsiness Detection Using Computer Vision. In 2020 International Conference On Computer, Commu-Nication, Chemical, Material And Electronic Engineering (Ic4me2) (Pp. 1-5). Ieee.
20. Jahan, I., Rahman, M. A., & Arafat, M. Y. (2018). Real-Time Drowsiness Detection System For Driver Safety Using Cnn. In 2018 2nd International Conference On Electrical, Comput-Er And Communication Engineering (Ecce) (Pp. 1-6). Ieee.
21. Ch, S. P., Guduru, S., Ronagala, V., Kuresan, H., & Dhanalakshmi, S. (2023, January 24). Automatic System For Driver Drowsiness Detection System Using Deep Learning. 2023 In-Ternational Conference For Advancement In Technology (Iconat).
17. A. Suresh, A. S. Naik, A. Pramod, N. Ashwin Kumar And N. Mayadevi, "Analysis And Im-Plementation Of Deep Convolutional Neural Network Models For Intelligent Driver Drows-Iness Detection System," 2023 7th International Conference On Intelligent Computing And Control Systems (Iciccs), Madurai, India, 2023
18. Li, X., Lu, Z., Zhang, C., & Liu, G. (2019). Driver Drowsiness Detection Based On Convolu-Tional Neural Networks. In 2019 Ieee 4th International Conference On Signal And Image Processing (Icsip) (Pp. 134-139). Ieee.
19. Ahmed, S. M. S., Paul, P., & Biswas, M. (2020). A Novel Approach For Driver