JOURNAL OF
Emerging Paradigms
in Computing Systems
JEPCS

Journal of Emerging Paradigms in Computing Systems (JEPCS) | Volume 03, Issue No. 02, 2026

ARTICLE TYPE: Research Article

Article ID: JEPCS-2026-0302-0067

A Comparative Study of Transfer-Learned VGG-16, Inception-V3, and ResNet-50 for Eye-State Classification in Driver Drowsiness Monitoring

Muhammad Waseem Riaz[1,*]

[1] Head of Department, School of Computing, NCBA&E / Alhamra University, DHA Campus, Lahore, Pakistan

*Corresponding author: hodcs.dha@ncbae.edu.pk

Contributing authors: hodcs.dha@ncbae.edu.pk

Received: October 14, 2026 **Accepted:** November 09, 2026

Abstract- Drowsy driving is a well-documented cause of road crashes, and a system that recognizes eye closure early enough to warn the driver could help prevent a share of them. This paper studies one well-defined component of such a system: the binary classification of a driver's eye state (open versus closed) from static images. It examines whether ImageNet-pretrained convolutional networks, adapted through Transfer Learning (TL), can serve as the perception stage of a driver drowsiness-monitoring process. Three Deep Learning (DL) architectures, VGG-16, Inception-V3 and ResNet-50, were fine-tuned under one common protocol on a public image dataset of open and closed eyes. The 4,040 images were partitioned at the image level into 2,840 training and 1,200 test images. On the test set, VGG-16 reached an Accuracy (ACC) of 99.67%, ResNet-50 96.42%, and Inception-V3 96.08%. With the open-eye class treated as positive, none of the models missed an open-eye image, so all three reach a sensitivity of 1.0. This reflects only that alert (open) eyes were recognized and does not mean that every closed eye was caught. The practically important differences therefore lie in the closed eyes misclassified as open: 4 for VGG-16, compared with 43 for ResNet-50 and 47 for Inception-V3, i.e. a closed-eye detection rate (specificity) of 0.9933, 0.9283 and 0.9217, respectively. The paper is limited to image-level eye-state classification on static images from a single dataset, using an author-defined split that is not guaranteed to be subject-independent, and in-vehicle validation is still required.

Keywords- Eye-state classification; driver drowsiness monitoring; convolutional neural networks; transfer learning; VGG-16

1. Introduction

Fatigue at the wheel is estimated to play a part in 10% to 20% of road accidents every year [1], which corresponds to roughly 100,000 crashes, 50,000 injuries and 800 deaths annually [2]. Set against a global toll that the World Health Organization (WHO) puts at around 1.35 million road deaths per year [4], within which driver fatigue is a recognized and preventable risk factor [1], [3], the need for reliable fatigue detection is clear. Monitoring the eyes is one of the most direct ways of meeting it [3]. Eye closure can be observed without any contact with the driver, and it reveals brief episodes of microsleep. These are particularly dangerous at high speed: at 100 km/h a car travels almost 28 m every second, so even a short lapse leaves the vehicle uncontrolled over a long distance. Progress in Machine Learning (ML) and Deep Learning (DL) over the past decade has made such measurements much more precise, to the point where a DSDS that operates in real time can realistically be built around them.

Convolutional Neural Networks (CNNs) have become the standard tool for image classification, and they are well suited to recognizing sleepiness from facial images. Of all facial cues, the eyes are the ones most often used in ML-based fatigue

detectors [5], since knowing whether they are open or closed is useful for sleep monitoring and eye-strain assessment as well as for driver fatigue detection [6]. Transfer Learning (TL) has made CNNs practical for narrow tasks of this kind. A network pretrained on a large, general dataset already carries reusable features, so a comparatively small set of eye images is enough to fine-tune it [7]. VGG-16 and Inception-V3 are two of the architectures most frequently adapted in this way, largely because of the quality of the features they extract [8]. VGG-16, from the Visual Geometry Group at Oxford, is a plain but deep stack of 16 weight layers [9], whereas Inception-V3, developed at Google, arranges its layers into inception modules so that a deeper network can be computed more economically [10]. The present study compares VGG-16, Inception-V3 and ResNet-50 as eye-state classifiers for driver sleepiness detection, using Accuracy (ACC) and related error metrics as the main criteria and considering their suitability for real-time use.

Sleepiness slows reactions, narrows attention and impairs judgement, and each of these effects raises the risk of a crash. A DSDS that observes the driver continuously and issues a warning as the eyes begin to close could protect passengers and other road users as well as the driver. Transport operators would also benefit, since crashes bring vehicle damage, medical and insurance claims and lost working time, and long-haul drivers, who are the most exposed to fatigue, stand to gain most. Advances in Artificial Intelligence (AI), DL, computer vision and TL have made camera-based driver monitoring far more dependable, yet published systems still vary widely in accuracy, and it is not obvious which pretrained backbone should be chosen. The motivation for this work is therefore practical: to establish how VGG-16, Inception-V3 and ResNet-50 compare when all three are fine-tuned for the same eye-state task under one training protocol.

The contributions of this paper can be summarized as follows:

- A controlled, single-protocol empirical comparison in which three ImageNet-pretrained architectures, VGG-16, Inception-V3 and ResNet-50, are fine-tuned under identical settings to classify eye state (open versus closed) as the perception stage of a driver drowsiness-monitoring pipeline, rather than a new architecture;
- A common, fully specified training protocol (ImageNet-pretrained weights fine-tuned end to end, Adam optimizer, learning rate of 0.00001, batch size of 32, cross-entropy loss, and a fixed author-defined partition of 2,840 training and 1,200 test images), so that differences in performance can be attributed mainly to the architecture rather than to the training settings;
- An evaluation that goes beyond ACC to include the Misclassification Rate (MCR), precision, sensitivity, specificity, F1-score, confusion matrices and training/testing loss and accuracy curves, with particular attention to closed-eye errors, which are the errors that matter most for safety; and
- A safety-oriented error analysis in which the three models are compared primarily by their closed-eye detection rate (specificity) rather than by overall accuracy alone, with VGG-16 reaching a testing ACC of 99.67%, against 96.08% for Inception-V3 and 96.42% for ResNet-50; because the earlier studies used different datasets and protocols, these values indicate where the present models stand rather than a controlled ranking (see Section 5.4).

The rest of the paper is organized as follow: Section 2 reviews earlier DL- and TL-based work on driver sleepiness detection. Section 3 presents the dataset, the overall framework and the three CNN architectures, and Section 4 describes the experimental setup and evaluation metrics. The results are reported and discussed in Section 5, together with a comparison against previous studies, and Section 6 gives the conclusions and directions for future work.

2. Related Work

According to India's transport authority, driver fatigue ranks third among the causes of road accidents. Sri et al. [11] addressed this with a CNN and a VGG-16 TL model that monitor the driver continuously. Facial landmarks are used to locate the eyes and the mouth, sleepiness is inferred from blinking and yawning, and an alert is raised once it is detected. The authors note that earlier models depended on only a few frames and on complicated architectures; their own CNN and VGG-16 reached ACC values of 96.87% and 96.61%, respectively [11].

Jain et al. [12] trained Inception-V3 and VGG-16 on the Kaggle "yawn-eye-dataset-new" (YEDN), which holds 2,900 images of open and closed eyes and of open and closed mouths, and assessed them using ACC, recall and F1-score. VGG-16 performed better, with 98.6% ACC and 98.5% recall, and the authors concluded that it is a dependable choice for sleepiness detection [12].

Ahmed et al. [13] also worked with 2,900 images, varying in gender, age, head position and lighting, and used eyeball movement to classify four facial states (eyes open, eyes closed, yawning and not yawning) with a custom CNN and with VGG-16. Their CNN obtained 97% ACC, 99% precision and an F1-score of 99%, whereas VGG-16 reached only 74% ACC [13]. The size of this gap suggests that the way a pretrained network is adapted to the task can matter as much as the choice of architecture.

Among the CNNs used for object detection and image classification, VGG-16 is one of the most familiar; it was the runner-up in the ILSVRC-2014 classification task. Narayanan et al. [14] treated the pretrained network as a fixed feature extractor. Bottleneck features from an ImageNet-trained VGG-16 were fed to a multi-layer perceptron, and multi-threading made it possible to detect driver distraction in live video. The authors report a throughput of 26 frames per second at 96%

Table 1: *Comparison of Earlier Driver Sleepiness Detection Studies.*

Ref.	Technique	Models and ACC	Limitations	Evaluation Metrics
[11]	Facial landmark-based drowsiness detection using eye-blink and yawning analysis	CNN (96.87%); VGG-16 (96.61%)	Moderate detection ACC; limited real-world validation; limited performance evaluation	ACC
[12]	TL-based eye and mouth state classification using the YEDN dataset	VGG-16 (98.6%)	Improvable ACC; limited environmental variability; real-time performance not reported	ACC, recall, F1-score
[13]	DL-based classification of facial sleepiness expressions	CNN (97%); VGG-16 (74%)	Low performance; model performance imbalance	ACC, precision, F1-score
[14]	TL with bottleneck feature extraction and multi-threaded real-time video processing	VGG-16 (96%)	Focused on driver distraction; limited sleepiness-detection scope; hardware-dependent performance	ACC, processing speed (FPS)
[15]	Hybrid real-time DTL with behavioral features and Chebyshev-distance analysis	VGG-16 (97%); VGG-19 (96%); ResNet-50 (98%)	High computational complexity; complex hybrid architecture; limited real-world validation	ACC, precision
[16]	Camera-position-based DL analysis of driver sleepiness	Inception-V3 (85.5%)	Lower detection ACC; camera-position dependency; limited generalization	ACC

ACC, a level they consider comparable to state-of-the-art real-time classifiers [14]. Their target was distraction; even so, the study demonstrates that a VGG-16 backbone is fast enough to operate under real-time constraints.

Turki et al. [15] paired DL with two behavioural indicators, namely the Eye Closure Ratio and the Mouth Opening Ratio. Pretrained VGG-16, VGG-19 and ResNet-50 networks first classified the images offline, and in the real-time stage the Chebyshev distance was applied to refine each decision. After data augmentation, the three networks achieved 97%, 96% and 98% ACC, respectively. Genuine sleepiness was detected with 98% precision, so false alarms were rare [15].

The influence of camera placement was investigated by Rozaimi et al. [16], who compared MobileNet-V2, ResNet-50 and Inception-V3. Inception-V3 gave the best result, an overall ACC of 85.5%, when the camera was mounted in the middle position [16]. The main characteristics of the studies discussed in this section are collected in Table 1.

The studies reviewed above leave little doubt that DL and TL are effective for DSDS. CNN-based methods reported ACC between 96.87% and 97%, and the TL models (VGG-16, VGG-19, Inception-V3 and ResNet-50) performed at a similar or higher level, reaching 98% with ResNet-50 [15] and 98.6% with VGG-16 [12]. Table 1 also makes clear that results vary with the dataset, the camera position, the facial cues selected, the real-time requirements and the use of augmentation. Certain weaknesses appear repeatedly. Datasets tend to be small, some configurations perform poorly, accuracy is sensitive to where the camera is placed, and only a handful of systems have been evaluated in realistic driving conditions. Because each study followed its own dataset and protocol, the reported figures cannot be compared directly, and this is part of the reason the present work evaluates all three architectures under a single protocol.

3. Proposed Methodology

3.1. Dataset and Proposed Framework

The proposed framework is outlined in Figure 1. The eye images come from the MRL Eye Dataset, a public collection of 84,898 infrared eye images acquired from 37 individuals (33 men and 4 women) under varied illumination with several sensors, in which every image is labelled as open or closed [17]. The subset used in this study was obtained from the public Kaggle redistribution of that dataset [17] and consists of 2,040 open-eye and 2,000 closed-eye images (4,040 in total). These images were taken, as provided, from a reduced and class-balanced Kaggle redistribution of the MRL data (dataset identifier prasadvpatil/mrl-dataset; accessed [month and year to be confirmed by the author]); that redistribution already supplies cropped single-eye images sorted into open-eye and closed-eye classes, so no manual selection or cropping from the full 84,898-image collection was performed by the authors; the division into training and test sets, however, was carried out by the authors as described below. During pre-processing, every image is resized to the input dimensions required by the network that receives it (224×224 for VGG-16 and ResNet-50, 299×299 for Inception-V3), handled as a three-channel RGB tensor and normalized with the ImageNet channel means (0.485, 0.456, 0.406) and standard deviations (0.229, 0.224, 0.225); no data augmentation was applied. The authors partitioned these 4,040 images at the image level into a training set of 2,840 images (1,400 closed and 1,440 open) and a test set of 1,200 images (600 closed and 600 open)–an approximately 70/30 split rather than the 80/20 ratio stated in the earlier version of this manuscript. The same partition is used for all three networks, so VGG-16, Inception-V3 and ResNet-50 are trained and tested on exactly the same images. The split is made at the image level; because subject identifiers are not preserved in the Kaggle redistribution, it cannot be guaranteed that images of a given individual appear in only one of the two sets. This limitation, together with the absence of a separate validation set, is discussed in Sections 4 and 6.

Table 2 lists the training settings and the resulting testing ACC of each model. All three networks used the Adam optimizer with a learning rate of 0.00001 and a batch size of 32. VGG-16 was trained for 20 epochs and reached the highest testing ACC, 99.67%. Inception-V3 and ResNet-50 were each trained for 15 epochs and reached 96.08% and

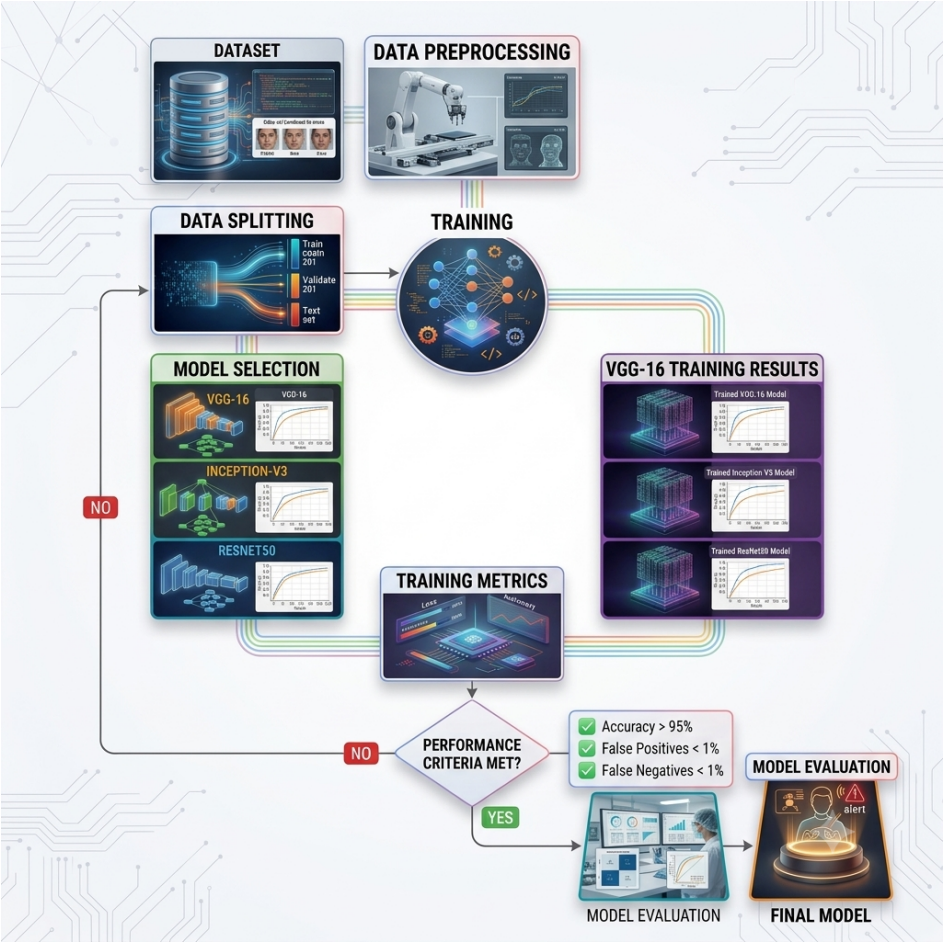


Figure 1: *Overview of the proposed eye-state classification framework.*

96.42%, respectively, which places ResNet-50 marginally ahead of Inception-V3 but clearly behind VGG-16. Figure 1 depicts the overall pipeline conceptually and does not imply any selection on test-set results. In practice each network was trained for a fixed number of epochs, after which the final-epoch model was evaluated once on the test set; no retraining, early stopping or model selection was carried out on the basis of test-set performance (see Section 4). The resulting model was saved to cloud storage for later use.

3.2. Convolutional Neural Networks

The CNN, introduced in [18], takes its inspiration from the way the visual cortex processes images. A typical CNN stacks convolution layers with nonlinear activations and subsampling (pooling) layers. The convolutions learn local patterns in the input image and produce feature maps that become increasingly abstract with depth [19]. Feature extraction is therefore carried out by the convolutional part of the network, while the fully connected (FC) layers at the end perform the classification. In this study, the networks are trained with backpropagation, which adjusts the convolution filters so that they respond to the features that best separate open from closed eyes. Figure 2 shows the generic CNN structure on which the models are based [20].

3.3. VGG-16 Model

VGG-16 is the first of the three eye-state classifiers evaluated in this work. The configuration used is shown in Figure 3.

VGG-16 is built from stacked convolutional layers that all use small 3×3 filters with a stride of 1. The number of filters doubles from one block to the next, from 64 in the first block to 512 in the last, so that the early layers respond to edges and textures and the later ones to larger structures [21]. Each convolution is followed by a Rectified Linear Unit (ReLU), which introduces non-linearity and is defined as

$$f(x) = \max(0, x) \tag{1}$$

Table 2: *Training Hyperparameters and Testing ACC of the Three Models.*

Model	Hyperparameter	Value
VGG-16	Number of epochs	20
	Learning rate	0.00001
	Optimizer	Adam
	Batch size	32
	Testing ACC	99.67%
Inception-V3	Number of epochs	15
	Learning rate	0.00001
	Optimizer	Adam
	Batch size	32
	Testing ACC	96.08%
ResNet-50	Number of epochs	15
	Learning rate	0.00001
	Optimizer	Adam
	Batch size	32
	Testing ACC	96.42%

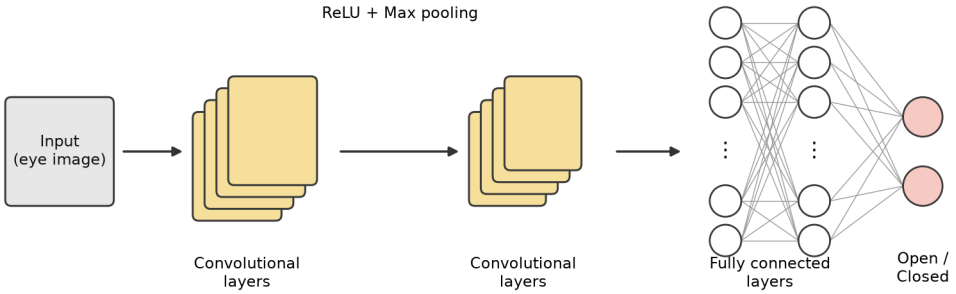


Figure 2: *General structure of a CNN [20].*

where x is the input to the neuron [22]. In this study the standard torchvision VGG-16 configuration was used: no batch normalization was added (the batch-normalized VGG-16 variant was not used), and no dropout layers were introduced beyond the two already present in the original fully connected classifier. Batch normalization after convolutional layers is known to stabilize training, accelerate convergence and reduce sensitivity to shifts in the input distribution [23], and dropout limits overfitting by randomly zeroing a fraction of the activations during training so that the network cannot depend on any single unit [24]; these techniques are described here only as background, to make explicit which components were and were not part of the implemented model. After each group of convolutions, max pooling with a 2×2 window and a stride of 2 halves the spatial size of the feature maps, keeping the strongest responses while reducing computation [25]. The final feature maps are flattened and passed to the FC layers. In the original network these comprise two layers of 4,096 units (each followed by a ReLU and dropout) and a 1,000-unit layer for the ImageNet classes; here only the final 1,000-unit layer is replaced by a two-unit fully connected layer for the open/closed problem [26], while the two 4,096-unit layers and their dropout are retained. The two class scores are converted to probabilities with a softmax and the network is trained with the cross-entropy loss; for the two-class case this softmax reduces to the logistic (sigmoid) function, defined as [27]

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

3.4. Inception-V3 Model

Inception-V3 is a later member of Google's Inception family of CNNs and achieved a top-5 ACC of 0.928 in ILSVRC-2015 [28]. The Inception design seeks to make a network deeper and wider without a matching rise in computational cost. It does this by approximating a sparse network structure with dense building blocks, in which filters of different sizes are grouped into parallel branches [29]. Figure 4 shows the structure of Inception-V3.

The Inception networks follow the network-in-network idea and have an asymmetric, multi-branch layout. Inception-V1 introduced the inception module, in which convolutions with several kernel sizes run side by side; the module captures

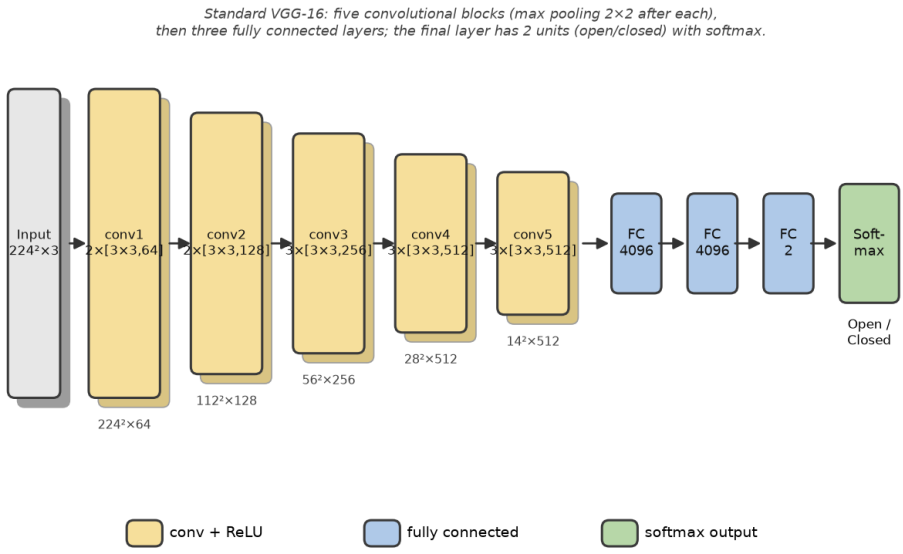


Figure 3: *VGG-16 configuration as used in this study [20].*

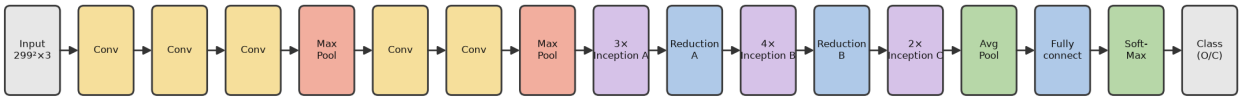


Figure 4: *Inception-V3 architecture [29].*

features at multiple scales and raises ACC while keeping the cost of the convolutions low [30]. Inception-V2 added batch normalization, which speeds up the early stages of training and eases the problems of vanishing gradients and overfitting [31]. Inception-V3 goes a step further and factorizes convolutions, replacing large kernels with smaller and asymmetric ones. Because depth is gained through small kernels, Inception-V3 has 1.38 times fewer parameters than Inception-V2, while the size of the feature maps and the receptive field are preserved and training becomes faster. This factorization strategy has since shaped the design of many later CNNs [32]. Figure 5 shows how an inception module processes its input along parallel branches and concatenates the results.

3.5. ResNet-50 Model

Camera-based sleepiness detection depends on visual cues such as eye closure, yawning and head movement, which an in-vehicle camera can record continuously while the vehicle is being driven [35]. Eye state and other facial features have accordingly served as descriptors in recent deep-learning detectors [34], and the performance of any such classifier depends in part on the quality of the images that the camera delivers. ResNet-50 [38] is a 50-layer CNN organized into residual blocks. Each block learns a residual mapping and adds its input back to its output through a shortcut connection, which lets gradients flow through very deep networks and largely removes the degradation in accuracy that affects plain deep CNNs. Its bottleneck blocks, consisting of 1×1, 3×3 and 1×1 convolutions, keep the computational cost moderate, and ResNet-50 has become one of the most widely used CNN backbones in applied image classification [36]. Attention-augmented ResNet variants have also been proposed in the literature [37]; however, no attention module was implemented in this study. The model used here is the standard ImageNet-pretrained ResNet-50, with its global average pooling retained unchanged and only its final 1,000-unit classification layer replaced by a two-unit fully connected layer for the open/closed task. Figure 6 shows the standard ResNet-50 architecture.

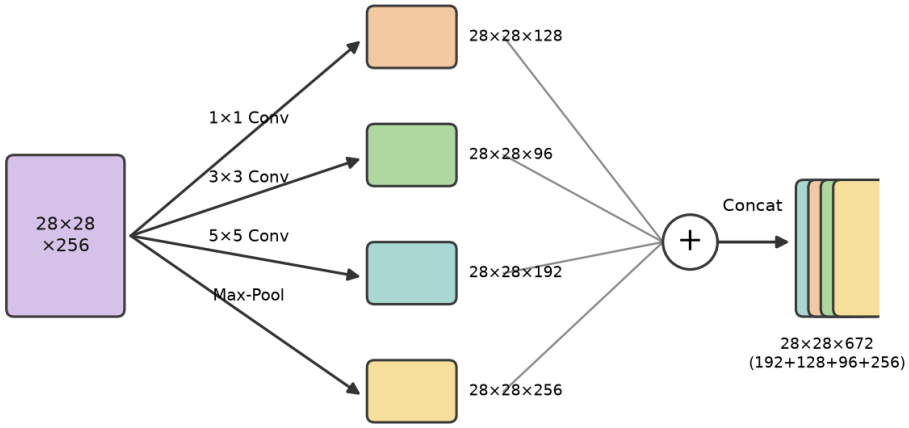


Figure 5: *Parallel convolutions and concatenation in an inception module [33].*

4. Experimental Setup and Evaluation Metrics

The experiments were run in Python on the Google Colab cloud platform. As listed in Table 2, all three models were trained with the Adam optimizer, a learning rate of 0.00001 and a batch size of 32, for 20 epochs (VGG-16) or 15 epochs (Inception-V3 and ResNet-50). Performance is measured with metrics derived from the confusion matrix, as defined in Equations (3)-(8). The Open_Eyes class is taken as the positive class, so a True Positive (TP) is an open eye recognized as open, a True Negative (TN) is a closed eye recognized as closed, a False Positive (FP) is a closed eye mistaken for an open one, and a False Negative (FN) is an open eye mistaken for a closed one. Under this convention, FPs are the errors with safety consequences, because each one is a closed eye that the system fails to notice. Specificity, the proportion of closed eyes that are correctly detected, is therefore the metric most directly linked to missed sleepiness. All three backbones were initialized with weights pretrained on ImageNet-1k (the torchvision default weights) and were then fine-tuned end to end: no layers were frozen, so every convolutional and fully connected parameter was updated during training. Only the classification head of each network was modified for the binary task. For VGG-16, the final 1,000-unit layer of the classifier was replaced by a fully connected layer with two outputs, while the two preceding 4,096-unit layers (with their ReLU activations and dropout) were kept. For ResNet-50, the final 1,000-unit fully connected layer was replaced by a two-output layer after the global average pooling. For Inception-V3, both the main classifier and the auxiliary classifier (AuxLogits) were given new two-output layers, and the auxiliary logits were included in the training loss, following the standard torchvision implementation; at test time only the main output was used. The number of trainable parameters is therefore approximately 134 million for VGG-16, 23.5 million for ResNet-50 and 22 million for Inception-V3, all of them trainable because no layer was frozen.

Because the three architectures require different input sizes, images were resized to 224×224 pixels for VGG-16 and ResNet-50 and to 299×299 pixels for Inception-V3. In every case the image was treated as a three-channel RGB tensor with pixel values scaled to [0, 1] and then normalized with the ImageNet channel means (0.485, 0.456, 0.406) and standard deviations (0.229, 0.224, 0.225). No data augmentation (no flipping, rotation, cropping or colour jitter) was used for any model; the preprocessing was identical across the three networks apart from the resize dimension.

Training used the cross-entropy loss and the Adam optimizer with a learning rate of 0.00001 and the PyTorch defaults for the remaining Adam parameters (β_1 = 0.9, β_2 = 0.999, ϵ = 1e-8 and no weight decay), a batch size of 32, and 20 epochs for VGG-16 and 15 epochs for Inception-V3 and ResNet-50. The newly added classification layers used PyTorch's default initialization, while all other layers started from the ImageNet weights. The experiments were implemented in PyTorch with torchvision (version 0.13 or later) and run on a Google Colab GPU runtime (CUDA). No fixed random seed was set for the reported runs, so exact numerical reproduction of the per-epoch values is not guaranteed; fixing a seed is recommended for future work and is noted among the limitations in Section 6.

In the reported experiments the data were divided into a training set and a held-out test set only; a separate validation set was not used. The per-epoch training and test curves in Figures 7-9 are monitoring curves: after each epoch the model was evaluated on the held-out test set purely to visualize convergence. The test set was never used to update the network weights, and no test-based model selection or early stopping was performed–the weights obtained after the final epoch were the ones reported. Consequently the headline test metrics remain a genuine hold-out evaluation of the final model,

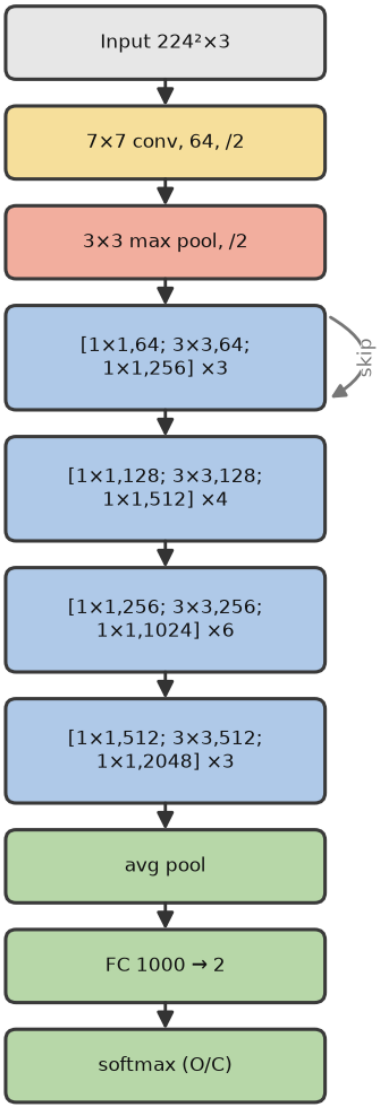


Figure 6: *Standard layer structure of ResNet-50; no attention module is used in this study [38].*

but reporting per-epoch test curves is not ideal practice. For a stricter protocol, future work should carve a dedicated validation subset out of the training data, use it (rather than the test set) for early stopping and model selection, and reserve the test set for a single final evaluation; a subject-independent split is also recommended, as discussed in Section 6.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{3}$$

$$\text{Misclassification Rate} = \frac{FP + FN}{TP + TN + FP + FN} \times 100 \tag{4}$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \tag{5}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100 \tag{6}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100 \tag{7}$$



Figure 7: *Inception-V3 per-epoch training and held-out-test loss and accuracy.*



Figure 8: *VGG-16 per-epoch training and held-out-test loss and accuracy.*

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{8}$$

ACC, MCR, precision, sensitivity (recall), specificity and F1-score are given by Equations (3)-(8), in that order.

5. Results and Discussion

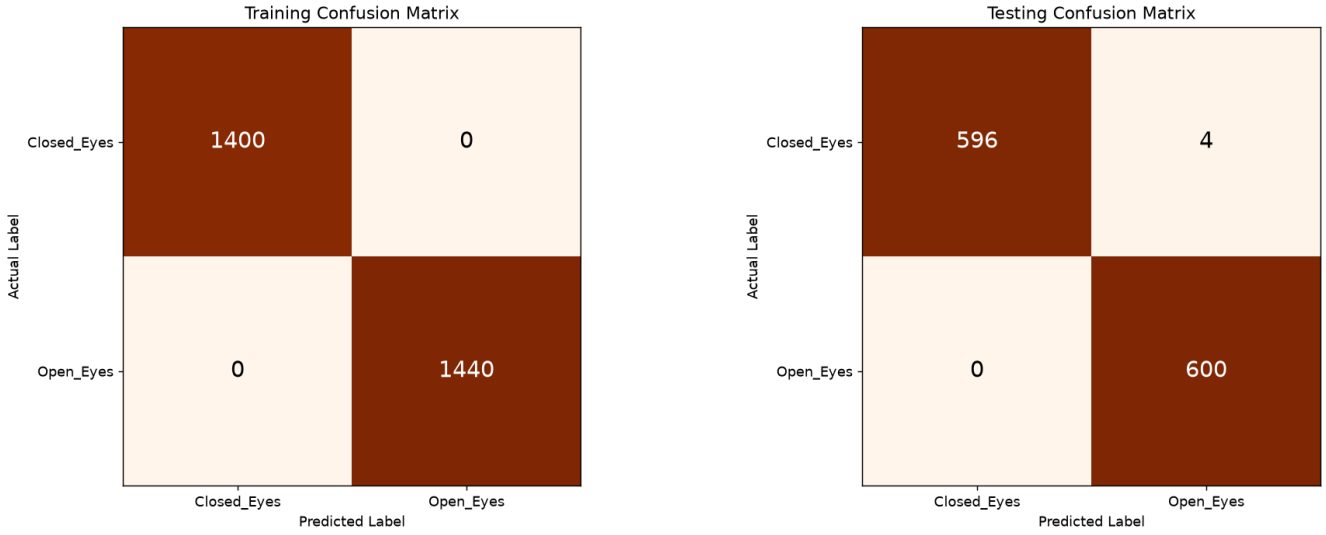
All three fine-tuned networks performed well, and VGG-16 performed best, with a testing ACC of 99.67%. For reference, the best ACC among the studies reviewed in Section 2 was 98.6%, which was likewise obtained with VGG-16 [12]. Inception-V3 reached 96.08%, compared with the 85.5% reported for the same architecture in [16], and ResNet-50 reached 96.42%. These comparisons need some care, because the earlier studies used other datasets, class definitions and protocols. On this basis the present models can be regarded as competitive, although such a comparison cannot by itself show that one method outperforms another under identical conditions. The subsections that follow look more closely at training behaviour, the confusion matrices and the individual metrics.

5.1. Training and Testing Behavior

The training and held-out-test loss and ACC of Inception-V3, VGG-16 and ResNet-50 are plotted against the number of epochs in Figures 7-9. As explained in Section 4, these test curves are monitoring curves only and were not used for weight updates, early stopping or model selection. On the training data the three networks were almost indistinguishable: training ACC was about 99.9%-100%, and training loss approached zero. On the test set, however, they diverged. Inception-V3 began at roughly 84% testing ACC, climbed rapidly and stabilized near 96%, while its testing loss stayed between 0.25 and 0.45. ResNet-50 oscillated between 95% and 97% testing ACC, and its testing loss ranged from 0.07 to 0.16. VGG-16 was the most stable, holding a testing ACC of 99.6%-99.8% with a testing loss of only 0.006–0.014. For Inception-V3 and ResNet-50, the gap of roughly four percentage points between training and testing ACC indicates a degree of overfitting



Figure 9: *ResNet-50 per-epoch training and held-out-test loss and accuracy.*



(a) *Training* ***(b)*** *Testing*

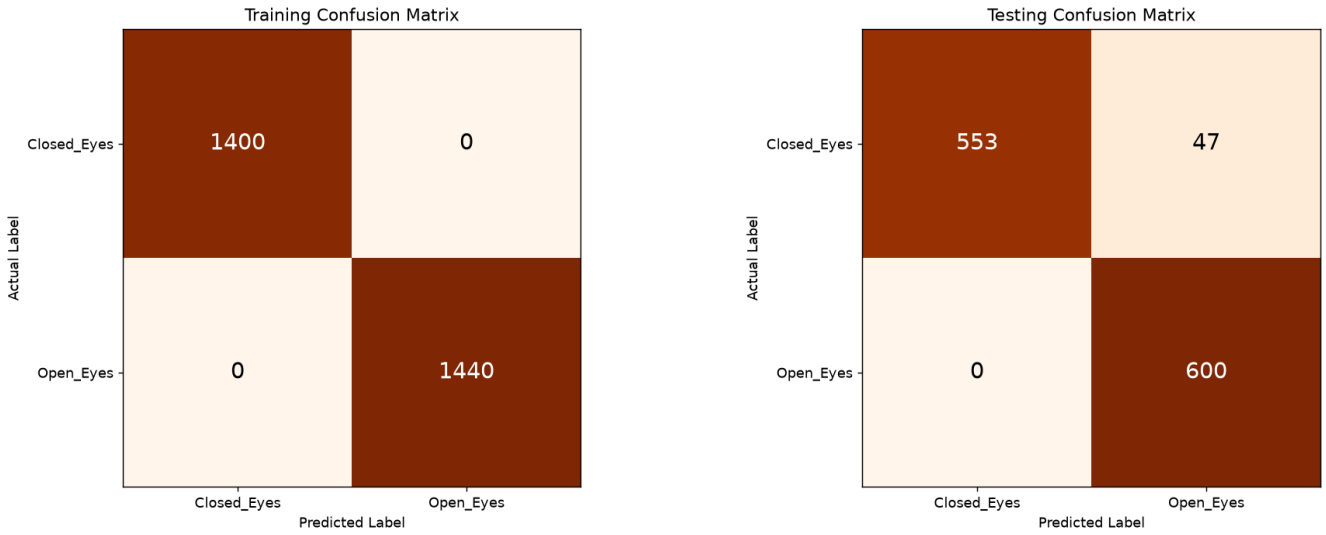
Figure 10: *Confusion matrices of the VGG-16 model: (a) training and (b) testing.*

that is much smaller for VGG-16. Because the test curves of these two networks had largely levelled off by epoch 15, the five additional epochs given to VGG-16 are unlikely to explain most of the difference, although they cannot be ruled out as a contributing factor.

5.2. Confusion Matrix Analysis

Figure 10 give the training and testing confusion matrices of VGG-16. On the 2,840 training images (1,400 Closed_Eyes and 1,440 Open_Eyes), the model made no errors, which corresponds to 100% ACC and 0% MCR. On the unseen test images, it recognized all 600 Open_Eyes images and 596 of the 600 Closed_Eyes images. Its only four errors were therefore closed eyes labelled as open, a very low FP rate that makes the model a strong candidate for the eye-state classification stage of a real-time drowsiness-monitoring system.

For Inception-V3 (Figures 11), training was again free of errors, with all 1,400 Closed_Eyes and 1,440 Open_Eyes images classified correctly. On the test set, all 600 Open_Eyes images were recognized, but only 553 of the 600 Closed_Eyes images were, which gives 47 FPs and no FNs. The overall ACC of the model remains high, yet about one closed eye in thirteen was missed, so its specificity, that is, its detection rate for closed eyes, is clearly lower than that of VGG-16. ResNet-50 (Figures 12) likewise classified all of its training images correctly (1,400 Closed_Eyes and 1,440 Open_Eyes, i.e. the same 2,840 training images used for the other two models), with 0% MCR. On the test set, it recognized all 600 Open_Eyes images and thus produced no FNs, and it labelled 557 of the 600 Closed_Eyes images



(a) *Training* ***(b)*** *Testing*

Figure 11: *Confusion matrices of the Inception-V3 model: (a) training and (b) testing.*

Table 3: *Test-Set Performance of the Three Models.*

Model	TP	TN	FP	FN	ACC	MCR	Precision	Sensitivity	Specificity	F1-score
VGG-16	600	596	4	0	0.9967	0.0033	0.9934	1.0000	0.9933	0.9967
Inception-V3	600	553	47	0	0.9608	0.0392	0.9274	1.0000	0.9217	0.9623
ResNet-50	600	557	43	0	0.9642	0.0358	0.9331	1.0000	0.9283	0.9654

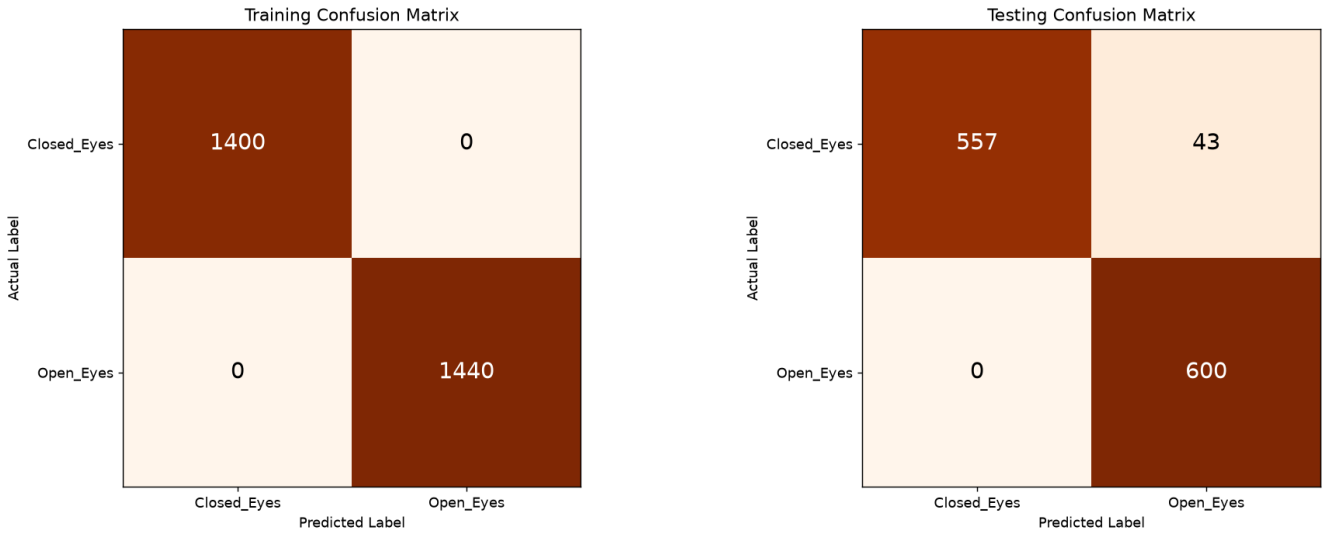
correctly while misclassifying the remaining 43 as open. With 43 closed-eye errors, ResNet-50 lies between VGG-16 and Inception-V3.

5.3. Comparative Metric Analysis

Table 3 compares the test metrics of the three models. None of them produced an FN on the 600 Open_Eyes test images, so all three have a sensitivity of 1.0000, and the differences between them lie entirely in the FPs, that is, in closed eyes classified as open. VGG-16 made only four such errors, which gives an ACC of 99.67%, an MCR of 0.0033, a precision of 0.9934, a specificity of 0.9933 and an F1-score of 0.9967. ResNet-50 and Inception-V3 reached lower ACC values of 96.42% and 96.08%, respectively, because of their larger FP counts (43 and 47). The perfect sensitivity of all three models should not be over-interpreted. With open eyes as the positive class, sensitivity measures how reliably an alert driver is recognized as alert, whereas specificity measures how reliably a closed eye is caught. On this safety-relevant measure, VGG-16 (0.9933) is well ahead of ResNet-50 (0.9283) and Inception-V3 (0.9217), and this is the main reason for preferring it as the classifier in the proposed system.

5.4. Comparison with Previous Literature

Table 4 places the present results beside those of earlier studies. Most of that work used CNN, VGG-16, VGG-19, ResNet-50 or Inception-V3 models and reported only ACC and MCR. The best earlier results were an ACC of 0.9860 for VGG-16 [12] and of 0.9800 for ResNet-50 [15]. The VGG-16 model trained here achieved an ACC of 0.9967 and an MCR of 0.0033, together with a precision of 0.9934, a sensitivity of 1.0000 and an F1-score of 0.9967, so precision and sensitivity are well balanced. Inception-V3 and ResNet-50 achieved respectable ACC values of 0.9608 and 0.9642. They too had a sensitivity of 1.0000; their precision, however, was lower (0.9274 and 0.9331, respectively), because a larger number of closed eyes were labelled as open. As a result, their F1-scores fell to 0.9623 and 0.9654. As noted earlier, the published figures come from different datasets and settings, so they show where the present models stand rather than providing a controlled ranking. Within this study, where the protocol was shared, VGG-16 was the strongest model on every metric, and its low MCR and high specificity make it the most suitable of the three for the proposed system.



(a) *Training* ***(b)*** *Testing*

Figure 12: *Confusion matrices of the ResNet-50 model: (a) training and (b) testing.*

Table 4: *Results of the Present Study alongside Those Reported in Earlier Work.*

Ref.	Model	ACC	MCR	Precision	Sensitivity	F1-score
[11]	CNN	0.9687	0.0313	–	–	–
	VGG-16	0.9661	0.0339	–	–	–
[12]	VGG-16	0.9860	0.0140	–	–	–
[13]	CNN	0.9700	0.0300	–	–	–
	VGG-16	0.7400	0.2600	–	–	–
[14]	VGG-16	0.9600	0.0400	–	–	–
[15]	VGG-16	0.9700	0.0300	–	–	–
	VGG-19	0.9600	0.0400	–	–	–
	ResNet-50	0.9800	0.0200	–	–	–
[16]	Inception-V3	0.8550	0.1450	–	–	–
Proposed	VGG-16	0.9967	0.0033	0.9934	1.0000	0.9967
	Inception-V3	0.9608	0.0392	0.9274	1.0000	0.9623
	ResNet-50	0.9642	0.0358	0.9331	1.0000	0.9654

6. Conclusion and Future Work

This study fine-tuned three well-established DL architectures, VGG-16, Inception-V3 and ResNet-50, to classify eye state for driver sleepiness detection. VGG-16 performed best, with a testing ACC of 0.9967 and only four of 600 closed-eye images misclassified, while Inception-V3 and ResNet-50 reached 0.9608 and 0.9642, respectively. Because all three models recognized every open eye, the comparison was decided by how well they detected closed eyes, which is also the error that matters most for safety. One caveat concerns model size. VGG-16 has far more parameters than either of the other two networks, and on constrained in-vehicle hardware a lighter model may still be the better choice; because inference cost was not measured in this study, that trade-off has yet to be quantified. On balance, the findings suggest that a TL-based eye-state classifier can serve as the core of a practical DSDS, as long as its present limitations are dealt with. Only one public dataset of static eye images was used, and the system has not yet been exposed to real driving conditions, in which illumination changes, drivers may wear eyeglasses and the head moves. The reported accuracy may also be optimistic if images of the same person appear in both the training and the test sets. Future work will accordingly focus on subject-independent evaluation, real-time implementation and in-vehicle testing, robustness to real-world variation, measurement of inference latency on embedded hardware, and the integration of multimodal data to raise the ACC and reliability of the system further.

Author Declarations

Author Contributions (CRediT): The author contributed to Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Visualization, Writing - Original Draft, and Writing - Review & Editing. The author read and approved the manuscript in its submitted form.

Funding: No external funding was received for this work.

Data Availability Statement: The open and closed-eye images used in this paper are publicly available on Kaggle [17]. Other materials can be obtained from the corresponding author on reasonable request.

Acknowledgments: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Ethics Statement: Not applicable.

Declaration of Generative AI: During the preparation and revision of this manuscript, the author used Claude (Anthropic) to format the manuscript to the journal template and for language editing, including the rephrasing of sentences. The author reviewed and verified all resulting content, including every reported figure, and takes full responsibility for the accuracy, integrity and content of the manuscript.

References

[1] National Academies of Sciences, Engineering, and Medicine, *Commercial Motor Vehicle Driver Fatigue, Long-Term Health, and Highway Safety: Research Needs*. Washington, DC, USA: The National Academies Press, 2016.

[2] H. A. Khalil, S. A. Hammad, H. E. Abd El Munim, and S. A. Maged, "Low-cost driver monitoring system using deep learning," *IEEE Access*, vol. 13, pp. 14151–14164, 2025.

[3] M. S. Al-Quraishi, S. S. A. Ali, A. Q. Muhammad, T. B. Tang, and S. Elferik, "Technologies for detecting and monitoring drivers' states: A systematic review," *Heliyon*, vol. 10, no. 20, 2024.

[4] World Health Organization, *Global Status Report on Road Safety 2018*. Geneva, Switzerland: World Health Organization, 2018. [Online]. Available: https://www.who.int/publications/i/item/9789241565684

[5] L. Kong, K. Xie, K. Niu, J. He, and W. Zhang, "Remote photoplethysmography and motion tracking convolutional neural network with bidirectional long short-term memory: Non-invasive fatigue detection method based on multi-modal fusion," *Sensors*, vol. 24, no. 2, 2024.

[6] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 596–614, 2010.

[7] S. Saurav, P. Gidde, R. Saini, and S. Singh, "Real-time eye state recognition using dual convolutional neural network ensemble," *Journal of Real-Time Image Processing*, vol. 19, no. 3, pp. 607–622, 2022.

[8] B. Sowmiya, K. Saminathan, and M. C. Devi, "An ensemble of transfer learning based InceptionV3 and VGG16 models for paddy leaf disease classification," *ECTI Transactions on Computer and Information Technology*, vol. 18, pp. 89–100, 2024.

[9] D. Hong et al., "Genetic syndromes screening by facial recognition technology: VGG-16 screening model construction and evaluation," *Orphanet Journal of Rare Diseases*, vol. 16, no. 1, Art. no. 344, 2021.

[10] E. Bazgir, E. Haque, M. Maniruzzaman, and R. Hoque, "Skin cancer classification using Inception network," *World Journal of Advanced Research and Reviews*, vol. 21, no. 2, pp. 839–849, 2024.

[11] B. R. Sri, Y. Akanksha, R. Puthali, and T. Anuradha, "Early driver drowsiness detection using convolution neural networks," in *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2021, pp. 1779–1784.

[12] B. Jain, A. Yadav, and R. Gill, "Comparative study of CNN architectures for driver drowsiness," in *Proceedings of the 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Jul. 2023, pp. 1–7.

[13] M. I. B. Ahmed et al., "A deep-learning approach to driver drowsiness detection," *Safety*, vol. 9, no. 3, Art. no. 65, 2023.

[14] A. Narayanan, V. Aiswaryaa, A. T. Anand, and N. Kadiresan, "Real-time detection of distracted drivers using a deep neural network and multi-threading," in *Proceedings of the International Conference on Artificial Intelligence and Data Engineering*. Singapore: Springer Nature Singapore, May 2019, pp. 89–100.

[15] A. Turki, O. Kahouli, S. Albadran, M. Ksantini, A. Aloui, and M. B. Amara, "A sophisticated drowsiness detection system via deep transfer learning for real time scenarios," *AIMS Mathematics*, vol. 9, no. 2, pp. 3211–3234, 2024.

[16] I. D. B. Rozaimi, F. H. K. Zaman, S. A. C. Abdullah, H. Z. Abidin, and L. Mazalan, "Driver drowsiness detection using deep learning models based on different camera positions," in *Proceedings of the 11th International Conference on Information and Communication Technology (ICoICT)*, Aug. 2023, pp. 611–616.

[17] P. Patil, "Drowsiness detection dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[19] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[20] I. Jahan, K. M. A. Uddin, S. A. Murad, M. S. U. Miah, T. Z. Khan, M. Masud, S. Aljahdali, and A. K. Bairagi, "4D: A real-time driver drowsiness detector using deep learning," *Electronics*, vol. 12, no. 1, Art. no. 235, 2023, doi: 10.3390/electronics12010235.

[21] S. Tammina, "Transfer learning using VGG-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications*, vol. 9, no. 10, 2019, doi: 10.29322/IJSRP.9.10.2019.p9420.

[22] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2015, doi: 10.1109/IJCNN.2015.7280459.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, vol. 37, 2015, pp. 448–456.

[24] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Computer Vision–ACCV 2016 Workshops*. Cham, Switzerland: Springer, 2017, pp. 189–204, doi: 10.1007/978-3-319-54184-6_12.

[25] L. R. Al-Khazraji, A. R. Abbas, and A. S. Jamil, "The effect of changing targeted layers of the deep dream technique using VGG-16 model," *International Journal of Online and Biomedical Engineering*, vol. 19, no. 3, 2023.

[26] Z. Liu, J. Peng, X. Guo, S. Chen, and L. Liu, "Breast cancer classification method based on improved VGG16 using mammography images," *Journal of Radiation Research and Applied Sciences*, vol. 17, no. 2, Art. no. 100885, 2024.

[27] T. N. Abu-Jamie and S. S. Abu-Naser, "Classification of sign-language using VGG16," 2022.

[28] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[29] M. A. S. Al Husaini, M. H. Habaebi, T. S. Gunawan, M. R. Islam, E. A. A. Elsheikh, and F. M. Suliman, "Thermal-based early breast cancer detection using Inception V3, Inception V4 and modified Inception MV4," *Neural Computing and Applications*, vol. 34, no. 1, pp. 333–348, 2022, doi: 10.1007/s00521-021-06372-1.

[30] C. Szegedy et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[31] L. Emma, "Building and training neural networks: Architectures, optimization, and challenges," Mar. 2025.

[32] R. Khanam, M. Hussain, R. Hill, and P. Allen, "A comprehensive review of convolutional neural networks for defect detection in industrial applications," *IEEE Access*, vol. 12, pp. 94250–94295, 2024.

[33] X. Wang et al., "A recognition method of ancient architectures based on the improved Inception V3 model," *Symmetry*, vol. 14, no. 12, Art. no. 2679, 2022.

[34] O. F. Hassan, A. F. Ibrahim, A. Gomaa, M. A. Makhlouf, and B. Hafiz, "Real-time driver drowsiness detection using transformer architectures: A novel deep learning approach," *Scientific Reports*, vol. 15, no. 1, Art. no. 17493, 2025.

[35] L. Sharara et al., "A real-time automotive safety system based on advanced AI facial detection algorithms," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 6, pp. 5080–5100, 2023.

[36] S. E. Abdallah, W. M. Elmessery, M. Y. Shams, N. S. A. Al-Sattary, A. A. Abohany, and M. Thabet, "Deep learning model based on ResNet-50 for beef quality classification," *Optimization*, vol. 30, p. 31, 2023.

[37] V. Y. Cambay, P. D. Barua, A. Hafeez Baig, S. Dogan, M. Baygin, T. Tuncer, and U. R. Acharya, "Automated detection of gastrointestinal diseases using ResNet50*-based explainable deep feature engineering model with endoscopy images," *Sensors*, vol. 24, no. 23, Art. no. 7710, 2024, doi: 10.3390/s24237710.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.