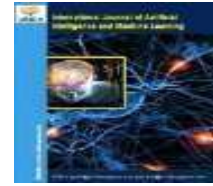




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Driver Drowsiness Detection System based on Ensemble deep learning model

Ali Noor Mahdi^{1*}, Hikmat Zned Neima²

¹ Department of Computer Science, Basra University, Basra, Iraq. E-mail: ali.noor@uobasrah.edu.iq

² Department of Computer Science, Basra University, Basra, Iraq. E-mail: Hikmat.taher@uobasrah.edu.iq

Abstract

Driver fatigue remains one of the most critical causes of road-traffic accidents. In the context, Drowsiness could be shown as a sleepiness state when a driver must have rest, with the emergence of several symptoms that have an impact on the performance of tasks. With the rapid progress of computer vision and deep learning frameworks, intelligent models have emerged as effective solutions for drowsiness detection. Therefore, this paper introduces drowsiness detection techniques using ensemble model based on deep learning algorithms. The ensemble model integrates ResNet50 for eye-state detection, VGG16 for mouth-state detection, and EfficientNetB3 for head position detection network pipelines with weighted average ensemble. The system operates on day and nighttime video frames generated by NITYMED and Yawdd. The ensemble model was assessed with three testing datasets with different conditions. These include daytime, nighttime, and a combined dataset. The model shows superiority in comparison with several existing models. It achieves an accuracy of 98%, 96% and 98% when testing on the combined dataset, daytime, and nighttime frames, respectively.

Keywords: Drowsiness Detection, Eye State Classification, Deep Learning, Convolutional Neural Networks (CNN), Image Classification, Driver Fatigue Monitoring

1. Introduction

Drowsiness can be defined as the transition between alertness and sleep. It is characterized by small reaction times [1]. It refers to a stage of fatigue resulting from mental or physical exhaustion and insufficient sleep. This state may lead to a phenomenon known as microsleep [2]. Therefore, driver drowsiness is a significant factor that causes road accidents [3].

Sleep deprivation, certain medications, and prolonged driving periods cause drowsiness, which usually comes on overtly and not very discreetly. When these conditions occur, the likelihood of an accident increases significantly; thus, fatigue monitoring becomes a key component of intelligent transportation systems.

Over 40% of U.S. adult drivers have admitted to having dozed off while driving at least once, and 20% have experienced drowsiness while driving in the previous year, according to the National Sleep Foundation. Furthermore, half of all drivers reported regularly feeling drowsy while on the road [4]. These statistics underscore the urgency for proactive monitoring and real-time intervention systems to combat fatigue-induced mishaps. Therefore, fatigue-related accidents remain a critical problem, since drowsy driving impairs concentration, reaction time, and decision-making abilities [5].

The driver drowsiness detection (DDD) systems, or commonly referred to as Advanced Driver Assistance Systems (ADAS), are a major contributor to road safety [6]. DDD systems continuously monitor the driver's physiological and behavioral conditions, providing timely alerts to prevent any potential catastrophes. Several methodologies exist for detecting driver fatigue and issuing warnings. physiological-based, vehicle-based, behavioral-based, and hybrid approaches represent the four main categories of DDD techniques.

Deep Learning (DL) and Convolutional Neural Networks (CNNs) variants have since been introduced as DDD systems to identify driver drowsiness through visual, physiological, and behavioral cues. CNN variants can be trained to recognize minute facial expressions, eyelid movements, head orientations, and yawning behaviors that are typically associated with fatigue, as opposed to classical image processing or rule-based systems [7-9]. It can automatically learn and generalize features from a vast amount of labeled data; therefore, it can adapt across different individuals, lighting conditions, and camera angles. This provides a highly versatile solution for real-time driver state monitoring in commercial vehicle platforms [8].

The DDD system includes several modules and stages shown in Fig. 1, starting with data acquisition through a sensing equipment called the input module, thereafter the data should be prepared using the preprocessing module, as in face detection, landmark application, feature selection, extraction, and transformation, dimensionality reduction, and classification or detection based on DL or Machine Learning approaches, and finally, an alarm module triggered in drowsiness cases [5, 7]. For the data acquisition module, the image and video-based module has revolutionized the state of the art, with various datasets having been employed for several DDD systems as follows. These include a module based on eye and face state, Mouth yawning state, Head-based state, and a Hybrid module [10].

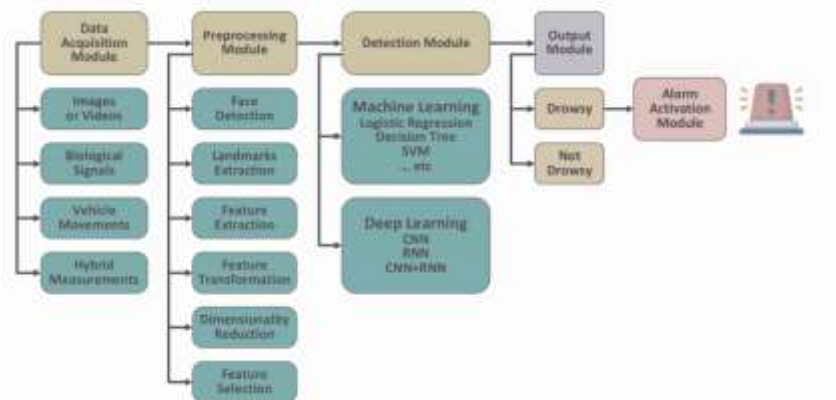


Fig. 1: General diagram flow in DDD systems [7].

These computer vision-based techniques have emerged as a promising non-intrusive alternative, but they face challenges such as:

- Performance degrades under light conditions such as shadows, glare, night driving, or infrared interference, which reduces the model accuracy.
- Many models fail to generalize across individuals with different facial structures, ethnicities, or wearing accessories (i.e., ethnic diversity, age, gender, fatigue tolerance, and glasses).
- DL models trained on limited, imbalanced, nondiverse datasets may achieve high accuracy in controlled settings but perform poorly in real-world environments.
- Problems related to reliability and robustness, when misclassifying alert drivers as drowsy, which reduce trust in the system. Therefore, a highly trusted model that provides a robust decision-making mechanism is needed.

Furthermore, most existing DL-based DDD systems rely on single-modal data, where the model focuses solely on detecting eye status while neglecting other important indicators such as yawning, physiological signals, or head pose, and vice versa.

The contribution of this work is the design of an effective and generalized DDD system based on state-of-the-art DL architectures that aim to improve the classification accuracy across diverse and complex datasets. The proposed framework employs an ensemble learning paradigm that integrates three CNN variants, which are ResNet-50, VGG-16, and EfficientNet-B3. In this configuration, ResNet-50 is fine-tuned to capture eye closure state, VGG-16 is adapted to recognize mouth-opening patterns, and EfficientNet-B3 is utilized to model head pose variations. Furthermore, the ensemble incorporates a staged transfer learning strategy in which the convolutional backbones of all CNN variants are initially frozen for a predefined number of epochs. This approach stabilizes early training, prevents overfitting, and allows the classification layers to adapt to domain-specific features before selectively unfreezing deeper layers for fine-tuning, thereby enhancing the model's generalization capability.

2. Related Work

Several studies adopted DL techniques, such as the work by Choi et al., in [11], which employs a DL technique that aims to build a gaze zone detection scheme based on the conventional CNN. The model learns the driver's features, which are then fed to a support vector machine to identify the driver's drowsiness. Dwivedi et al. (2022) [12] proposed a DL model that detects facial features using a MobileNet model. The eye closure state frames are provided as input to the MobileNet, which attempts to classify the drowsiness status based on the eye state of the driver.

Furthermore, CNN variants-based DDD systems are also being integrated with sensor fusion techniques, cloud computing, and Internet of Things architectures in order to enhance their effectiveness, reliability, and scalability. These systems provide continuous assessment of the driver's alertness, and upon necessity, set off an alarm or activate safety interventions to reduce accident probability [13-15].

In the context, Park et al, [16] introduces a conventional CNN-based model capable of directly predicting the vehicle's longitudinal and lateral speed targets from raw LiDAR and camera inputs. The proposed method leverages an end-to-end learning architecture in which the CNN maps sensor data to control-relevant outputs without intermediate handcrafted stages. The work in [17] introduced a fully connected 4-layer CNN designed to learn hierarchical visual cues, which includes the eye opening and yawning state in the frames. This model captures eye-movement patterns and yawning states precisely. Therefore, it achieves an accurate detection of driver fatigue with 97% accuracy.

Yoheswari in [18] proposed an optimized DDD system utilizing an optimized ML framework using behavioral and physiological signals. By combining models such as SVM, CNN, and Random Forest with optimization methods like GA or PSO, the system significantly improves detection accuracy and reduces false alarms for practical deployment. Deng et al., 2022 [19] proposed an efficient driver fatigue detection system that uses Multi-Task Cascaded Convolutional Networks (MTCNN) for facial feature extraction and physiology analysis through infrared-based remote photo-plethysmography. In this work, a visible-IR image data-based multi-modal feature fusion model has been implemented to improve the accuracy of fatigue detection.

Das et al., 2023 [20] introduces DDD system using CNN-LSTM and U-Net-based facial-movement analysis, which achieves higher accuracy than conventional deep-learning models such as VGG-16, GoogLeNet, AlexNet, and ResNet50. The system processes short video clips to recognize eye and facial motion patterns in real time, providing timely alerts. The author in [21] introduces a DDD system based on a modified conventional CNN, in which driver fatigue is evaluated by analyzing eyeball movement and facial expressions. The model classifies the output in multiclass form involving open eyes, closed eyes, yawning, and no yawning. However, the model trained with only a 2900 image dataset achieved 97% accuracy, outperforming VGG16.

Phan et al., 2021 [22] proposed two DDD systems based on detecting blink and yawn using adaptive DL models of MobileNet-V2 and ResNet-50V2. Their experiments showed that the model achieves 97% accuracy. In [23], the authors introduce the use of DLIB library facial landmarks to compute the Mouth Aspect Ratio (MAR) and employ an extreme data augmentation to address class imbalance. The model uses a standard CNN, which achieves 96% accuracy and outperforms several existing state-of-the-art approaches. Dua et al., 2021 [24] propose an ensemble model consisting of four DL model frameworks for DDD. The model integrates AlexNet, VGG-FaceNet, FlowImageNet, and ResNet to analyze environmental features, facial features, head motion, and hand gestures, respectively. By combining the output of the SoftMax classifier for each algorithm using a simple averaging formula, the system achieves an accuracy of 85%.

The study in [25] proposes a modified CNN-based DDD system that works on eye status and MAR. The architecture significantly outperforms InceptionV3, VGG16, and ResNet50 when evaluated on NITYMED dataset. It achieves 99.88% accuracy. However, this model was tested utilizing only 510 frames make it a less robust model. Yang and Okyeon, 2024 [26] proposed a DL-based DDD system that extracts complex features using DLID3-ADAS and ShuffleNet. In addition to that, the Northern Goshawk Optimization (NGO) algorithm is employed to optimize ShuffleNet hyperparameters. The model was evaluated using YawDD dataset. It shows superiority by achieving 97% accuracy.

The author in [27] proposed DDD model using Haar-Cascade detectors and CNN-based facial landmark tracking. The basic idea of the model is to estimate eye-state patterns. Then, it triggers an alarm when it detects a long-term eye closure. The model achieves an accuracy of 96% with low computation complexity.

Author, Year	Dataset	Method	Performance
Choi et al. (2016) [11]	Private Dataset	CNN feature extraction with SVM-based gaze.	95%
Park et al. (2018) [16]	LiDAR + Camera	End-to-end CNN predicting vehicle control from raw sensor data.	93%
T. Ahmed et al. (2019) [17]	YawDD	4-layer CNN for eye-opening detection.	97%
MIB et al. (2020) [21]	YawDD	Modified CNN for eyeball movement and facial expression.	97%
Phan et al. (2021) [22]	YawDD	MobileNet-V2 and ResNet-50V2 for blink and yawn detection.	97%

Dua et al. (2021) [24]	National Tsing Hua University (NTHU) datasets.	Ensemble of AlexNet, VGG FaceNet, FlowImageNet, and ResNet50,	85%
Majeed et al. (2021) [23]	YawDD	DLIB landmarks and MAR with CNN.	96%
Dwivedi et al. (2022) [12]	Private dataset.	MobileNet-based eye-closure classification.	97%
Deng et al. (2022) [19]	Private dataset.	Multiple CNNs with multimodal fusion.	92%
Florez et al. (2022) [25]	NITYMED	Modified CNN using eye region and MAR analysis.	99 %
Das et al. (2023) [20]	Private dataset.	CNN-LSTM and U-Net for facial-motion video analysis	98%
Janarthanan et al. (2023) [27]	Private dataset.	Haar-Cascade eye detection with CNN-based landmark tracking	96%
Yang & Okyeon (2024) [26]	YawDD	ShuffleNet optimized with Northern Goshawk Optimization	97%

3. Theoretical background

In this section, the basics and background of the paradigms that have been implemented are presented. These paradigms are used to formulate the DL-based DDD system ensemble model. Mainly, these frameworks involve three CNN models, including ResNet50, EfficientNetB3, and VGG16.

3.1 ResNet50

ResNet-50 is a 50-layer deep CNN known for the utilization of residual blocks to overcome training issues. ResNet-50, developed by Microsoft Research, was originally trained on the large-scale ImageNet dataset, enabling it to recognize and categorize images across 1000 distinct object classes [28]. ResNet-50 was selected as a primary architecture due to its use of skip connections. These connections enable the input feature maps to bypass intermediate layers and flow directly to deeper layers through identity mappings [29].

Such a design greatly reduces the effects of gradient vanishing during training. By relying on residual learning, ResNet focuses on learning the change between the desired output and the original input. This mechanism ensures that as the depth of the network grows, its performance does not degrade, thereby enhancing the efficiency of both training and feature extraction [30]. The architecture of ResNet-50 is fundamentally built around residual blocks. The network is composed of four major stages of residual blocks and a final classification stage. The following describes the structure [31]:

- Input layer: receives an input image with specific dimensions.
- An initial convolutional layer: it is a 7x7 kernel and 64 filters, which ends with a 3x3 max-pooling layer.
- Core residual blocks: utilize three convolutional layers within their structure. These convolutions are 1x1, 3x3, and 1x1 convolutions.
- Skip connections: add the output of a previous layer to the output of an advanced layer within the residual block.
- Global average pooling layer: This is added after the four main stages, and it is followed by the dense layer.

3.2 EfficientNetB3

EfficientNet-B3 is a CNN that belongs to the EfficientNet family, developed by Google researchers in 2019. The EfficientNet-B3 variant is more advanced than the earlier versions (B0, B1, and B2) and offers higher accuracy at the cost of increased computational demand [32]. EfficientNetB3 uses the Mobile Inverted Bottleneck Convolution (MBConv) block. The MBConv block includes Depth-wise Convolutions (DC), Squeeze-and-Excitation (SE), projection, and Swish Activation (SA).

However, there are two categories of MBConv, which are MBConv1 and MBConv6. In MBConv6, the channel of the input tensor is expanded by applying 1x1 convolution in order to improve the representation of the feature. This is not the case in MBConv1, in which the input is directly learned by DC layer [32-34].

DC technique significantly reduces the number of parameters and computation by subjecting convolutions independently to each channel. SE is applying global average pooling and recalibration. The global average pooling results in 1x1xC, where C is the number of channels in the feature map, passing it to the recalibration module. This module includes two fully connected layers to learn.

The first one reduces the channel, while the second expands it. Then, performing channel-wise multiplication. The output then passes to a 1x1 convolution for reducing the number of channels, in a process called projection, compressing the feature map for the upcoming layer. These features are then fed to batch normalization and a non-linear swish activation function, which helps in providing nonlinearity and training stability [35]. From the figure, the general structure of EfficientNet-B3 starts with a 3x3 convolution layer, and then the structure can be defined by 7 blocks. The first block includes two MBConv1 with 3x3 convolution, while the remaining 6 blocks are all MBConv6 type, with each one repeated 3,3,5,5,6,2 times respectively. The final output is subjected to global pooling and fed to a fully connected layer [36]. Fig. 2-14 and 2-15 demonstrate the general structure of EfficientNet-B3 and the structure of MBConv1 and MBConv6, respectively.

3.3VGG16

VGGNets16 is a member of a family of CNN models introduced by the University of Oxford at the Visual Geometry Group (VGG). These encompass several versions of VGG of 11, 11-LRN, 13, 16, and 19. In the localization task of the 2014 ImageNet Challenge, these architectures achieved first place. The authors demonstrated that, within certain limits, increasing network depth can lead to notable gains in performance [37]. The 16 layers of VGG16 model include 13 convolutional layers and three dense layers. The VGG16 convolution layer can be defined using five blocks. A 3x3 convolutional layer followed by a 2x2 max-pooling layer is the component of each block.

The first two blocks each contain two 3x3, while the rest of the blocks contain three 3x3 convolutions. All layers in the convolutional core use various activation functions [38]. The final output of the last pooling layer is flattened and passed to the block layers of the dense layer. This block has three fully connected layers is such that the first two have 4096 neurons, while the last one has only 1000 neurons. Fig. 2-12 and 2-13 show the convolutional block and the general structure of VGG16, respectively [39].

4. Proposed Work and material

This section outlines the methodology for designing an ensemble DL drowsiness detection system. The system combines three DL models, which are ResNet50, EfficientNetB3, and VGG16, each focused on analyzing a specific visual feature, which are eye state, mouth state, and head tilt.

4.1Dataset description

To ensure generalizability, this work employs two benchmark video datasets used in driver-drowsiness research in recent times. The first is the Nighttime Yawning and Microsleep Eye-Blink Distraction Dataset (NITYMED) dataset, released in 2022. It consists of 130 high-resolution videos (720p and 1080p) collected from 21 drivers (11 male, 10 female). The videos range in duration from 15 seconds to 2 minutes. Specifically, 107 videos capture yawning instances, each containing at least three yawns, while 21 videos depict microsleep-related behaviors such as prolonged eye closure, head nodding. Additionally, the dataset includes participants with diverse hairstyles, glasses, facial hair, and expressions, enhancing its suitability for robust real-world applications.

The second benchmark is the Yawning Detection Dataset (YawDD) released in 2023. It comprises 351 videos collected from 107 drivers (50 female, 57 male). Each video is 15–40 seconds in duration and recorded at a resolution of 640 × 480 pixels. The dataset is designed primarily for yawning detection, drowsiness recognition, and mouth-activity classification. It also provides multi-angle recordings, including dashboard and mirror perspectives, making it compatible with advanced facial-landmark detection.

When combined, NITYMED and YawDD datasets provide a total of 481 videos across a diverse participant cohort. Together, they offer comprehensive coverage of multiple indicators of driver drowsiness, including eye closure, yawning, and head-pose variations, across a wide range of real-world conditions.

4.2Data preparation and frame analysis

As this study uses a combined dataset, this section describes the data preparation procedures. This stage began with extracting frames from the videos, totaling 15000 frames, by using OpenCV at a rate of 5 frames per second. As raw frames do not contain explicit feature values, they must be analyzed with the following steps.

Step 1: MediaPipe, FaceMesh, and FaceDetection libraries were used to detect 468 facial landmarks, illustrated in Fig. 2. Therefore, using these face frames to compute Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), and head position.

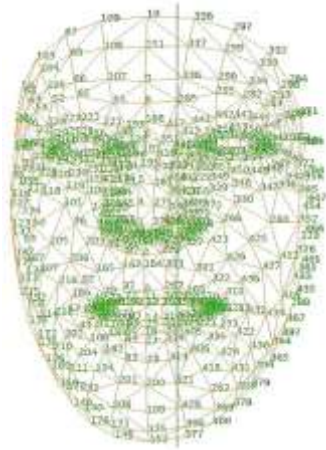


Fig. 2: MediaPipe face mesh solution map [40].

Step 2: Eye landmarks, as illustrated in Fig. 2, were then extracted to calculate the EAR using expression 1, and then classified based on a specific threshold value.

$$EAR = \frac{(|p_2 - p_6| + |p_3 - p_5|)}{(2|p_1 - p_4|)} \quad (1)$$

For instance, an eye is classified as closed (or drowsy) if $EAR < 0.50$ and open (non-drowsy) if $EAR \geq 0.50$. Frames were then saved into separate folders with a tensor size of $224 \times 224 \times 3$. This step resulted in 5121 drowsy frames and 9879 non-drowsy frames.

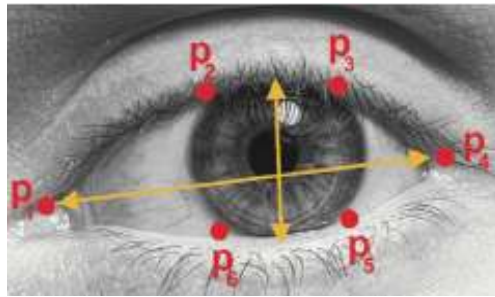


Fig. 2 : MediaPipe eye state landmarks [40].

Step 3: MAR is computed using the formula in (2). The MAR is frequently used to measure the degree of mouth opening in facial expression analysis and emotion detection. Geometric ratio derived from key facial landmarks on the lips.

$$MAR = \frac{||p_{up} - p_{down}||}{2 \cdot ||p_{right} - p_{left}||} \quad (2)$$

Where p_{up} is the center of the upper lip; p_{down} denotes the center of the lower lip; and p_{right} , p_{left} are the right and left corners of the mouth, respectively. The mouth open threshold is set to 0.35; that is, if the $MAR > 0.50$, then the frame is set to drowsy, otherwise it is non-drowsy. After extracting the MAR-based classified frames, the frames were saved into separate folders with a size of 224×224 pixels. This step resulted in 2035 drowsy images and 12965 non-drowsy frames.

Step 4: The head position screening module is computed using the formula in (3) For each frame, a face is localized and 468 facial landmarks are extracted from these, six key landmarks (nose tip, chin, left and right eye corners, and left and right mouth corners) are selected to estimate the 3D head pose using the Perspective-n-Point (PnP) algorithm.

The corresponding 3D reference model points are denoted by M_i and the 2D detected image points are P_i . Given the intrinsic camera matrix K and distortion vector d . The PnP problem estimates the rotation vector r and translation vector t that minimize the reprojection error between the projected 3D points and the observed 2D points. The rotation matrix R obtained from r is then converted into Euler angles $(\phi_{pitch}, \phi_{yaw}, \phi_{roll})$, where

these three denote the vertical tilt (nodding), horizontal rotation (turning left or right), and in-plane rotation (head roll).

$$\begin{matrix} R & t \\ 0 & 1 \end{matrix} = \text{solvePnp}(M, P, K, d) \quad (3)$$

In order to categorize head posture, angular thresholds are applied to the pitch and yaw components. A pitch angle greater than 20° ($\phi_{\text{pitch}} > 20^\circ$) is labeled as downward (forward tilt, often indicative of drowsiness), while a yaw angle exceeding 25° ($\phi_{\text{yaw}} > 25^\circ$) indicates left or right deviation, reflecting normal attention shifts and thus considered non-drowsy. Moreover, frames where the pitch angle is below 20° , and the yaw angle remains within $\pm 25^\circ$ are regarded as forward-facing and also classified as non-drowsy frames. This step resulted in 4781 drowsy frames and 10219 non-drowsy frames.

Step 5: All of the mouth-based, eye-based, and head position-based frames with their labels (i.e., drowsy, non-drowsy) are stored in a separate folder, so that each deep learning algorithm in the ensemble model is trained on the relative feature separately.

Hence, the initial stage of the proposed model, videos were decomposed into individual frames at 5 frames per second using MediaPipe resulting in more than 41000 frames in total.

However, to reduce computation complexity while guaranteeing reliability, these are reduced to 15000 well-diversified frames, with each frame assigned a unique reference identifier. Table 1 shows the information on the entire frame sets, where the drowsy label is represented by the letter “D”, and non-drowsy is denoted by N.

Table 1: The dataset features, classes, and counts.

Feature	Eye state		Mouth state		Head state		Frame label	
Class	D	N	D	N	D	N	D	N
No. of frames	5121 34.1%	9879 65.9%	2035 13.6%	12965 86.4%	4781 31.9%	10219 68.1%	6272 41.8%	8728 58.2%

4.3 Deep learning ensemble models

The classified frames of the three features were split into two sets. The first set is the training set with 80%, while the second is the testing set with 20%. A cross-validation splitter is applied, which creates a shuffled function with a random splitting in which the class proportions are preserved between the two subsets, to ensure balanced class distribution. The training dataset consisted of 12000 frames belonging to each class, while the testing dataset consisted of 3000 frames belonging to two classes.

All trainable models use K-Fold with a five-fold scheme applied to all three datasets, thereby each fold preserves the drowsy and non-drowsy class ratio such that training is performed on four folds and validation is performed on the remaining fold.

In the context, as illustrated in Fig. 3, which shows the general process of the model employed, the 1500 frames are fed to three network models in such a way that the classified frames based on eye state are fed to ResNet50 to train it to be an eye-closure drowsiness detector that learns the eye state of the face frame. In the same way, the classified frames based on head position are fed to EfficientNetB3 to train it to be a head position-based drowsiness detector that learns the head position of the face frame. Finally, the classified frames based on mouth state are fed to VGG16 to train it to be a mouth-opening-based drowsiness detector that learns the mouth state of the face frame.

Starting from models already trained on a huge dataset (i.e., ImageNet), all three algorithms learn the task faster and work well even with fewer labeled frames. Compared with bigger networks, ResNet-50, VGG-16, and EfficientNet-B3 provide strong accuracy with less computational cost.

All algorithms were initialized with transfer learning weights and adapted via a two-stage schedule, which includes a freezing phase and a fine-tuning phase with standard augmentations. In the first phase, the backbone is frozen, and only the classifier head is trained, which stabilizes adaptation and prevents catastrophic forgetting of generic visual features. In the subsequent phase, some high feature extractor layers are unfrozen with a slightly higher learning rate to specialize the representation to ocular cues. This initialization strategy leverages strong priors from large-scale pre-training while maintaining a compact, regularized classifier head, yielding an effective starting point for reliable convergence.

In other words, the training is only performed on the classifier head during the freezing phase; therefore, the learning process is stable and fast. The big pretrained features don't get destroyed, which avoids overfitting and “catastrophic forgetting,” especially with limited data. Once the classifier is reasonable, the earlier layers adapt to the domain. This nudges generic features toward task-specific patterns (eye, head, and mouth cues),

typically giving a clear accuracy boost. Table (2) shows some initial values and selections shared by all three models.

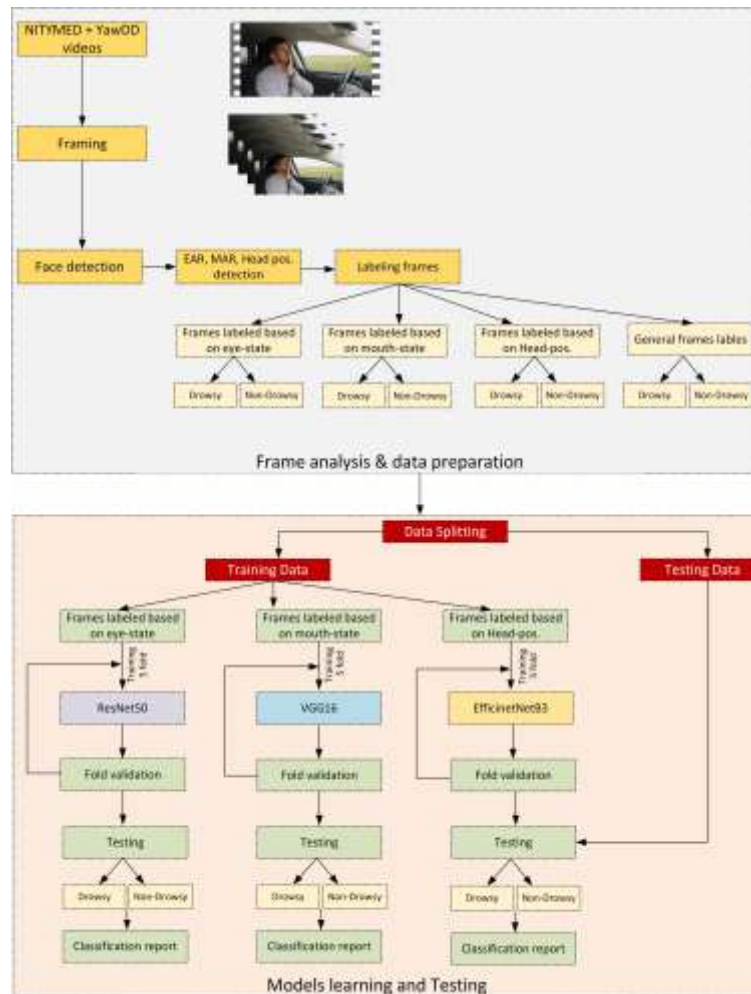


Fig. 3: Diagram of the proposed model.

Table 2: initial values and selections for all three models.

Component	Value or Choice
Backbone	ResNet-50, VGG-16, EfficientNet-B3
Input shape	224 x 224 x 3 (RGB)
classifier head	Global average pooling with Dropout of 0.5, and a fully connected classifier with the sigmoid function.
Frozen phase	All backbone layers are non-trainable; train the head only.
Fine-tuning phase	Some high-feature extractor layers.
Optimizer (frozen)	Adam
Optimizer (fine-tune)	Adam
Loss / Metric	Sparse categorical cross-entropy/accuracy
Dropout	0.5
No. of folds	5
Frozen epochs	5
Unfrozen epochs	20~25

4.3.1 ResNet50 for classifying eye-based frames

As illustrated in Fig. 4, which shows the structure of ResNet50 model utilized to classify the eye-based frames, practically, RGB eye-based frames of size $(224 \times 224 \times 3)$, preprocessed with the preprocess method, which is a routine used strictly as a feature extractor during the first stage.

ResNet50 employs the canonical feature extractor, which is a 7×7 convolution with a stride equal to 2 and a max-pool stem followed by the four bottleneck residual stages starting from conv2 to conv5 with block counts [3,4,6,3]. and stagewise down sampling to $56 \times 56 \times 256$, $28 \times 28 \times 512$, $14 \times 14 \times 1024$, and $7 \times 7 \times 2048$ feature tensors. The classifier head performs global average pooling on the $7 \times 7 \times 2048$ activation to obtain a 2048-dimensional embedding, applies dropout for regularization, and maps it via a Sigmoid function to probabilities over the two classes. In the freezing phase, the full backbone is frozen, and only the classifier head is trained, while in the fine-tune phase, conv4 to conv5 are unfrozen and trained.

The training process monitors the validation loss values and applies early stopping criteria with six patience epochs and a reduced learning rate. The mini-batches are of size 64, and the training selection is estimated via a 5-fold cross-validation on the training set.

4.3.2 VGG-16 for classifying mouth-based frames

VGG-16 model has five convolutional blocks with (3×3) filters, two conv layers in blocks [1,2] and three conv layers in blocks [3,4,5], each block followed by a (2×2) max-pool with a stride equal to 2, yielding stagewise spatial reductions down to a terminal feature tensor of $(7 \times 7 \times 512)$.

The classifier head applies global average pooling over this $(7 \times 7 \times 512)$ to produce a 512-dimensional embedding, then uses dropout at a rate of 0.5 for regularization and a final SoftMax layer to map to posterior probabilities over the two classes. Training adopts the same two-phase protocol used with ResNet-50, where in the freezing phase, the full backbone is frozen, and only the classifier head is trained, while in the fine-tuning phase, blocks 4 and 5 are unfrozen and trained.

Validation loss is monitored with early stopping at a patience of six epochs and a reduced learning rate with values. The mini-batches comprise 64 frames, and model selection is estimated via 5-fold stratified cross-validation on the training set. Fig. 5, and illustrate the structure of the model utilized to classify the mouth-based frames into a decision of drowsy driver or non-drowsy.

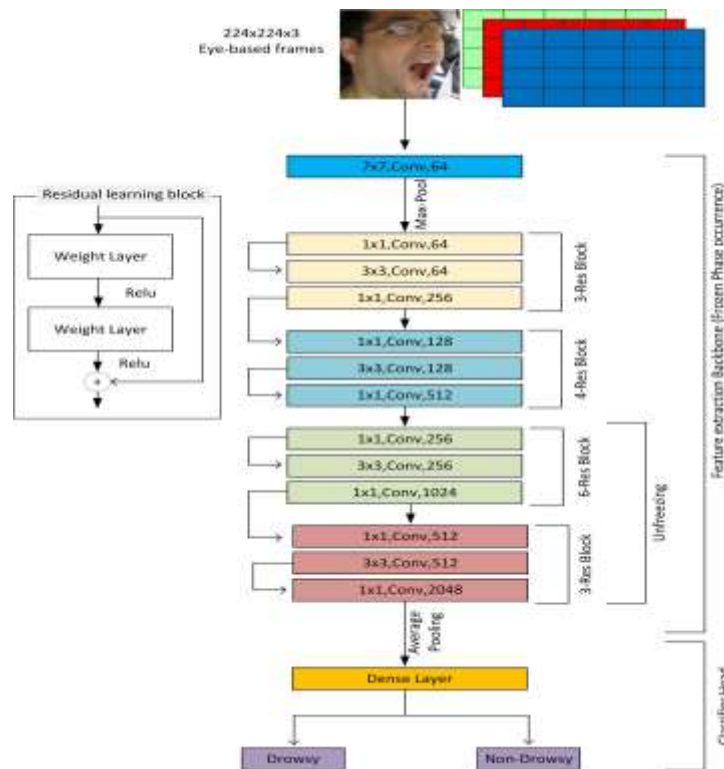


Fig. 4: ResNet-50 structure for classifying eye-based frames.

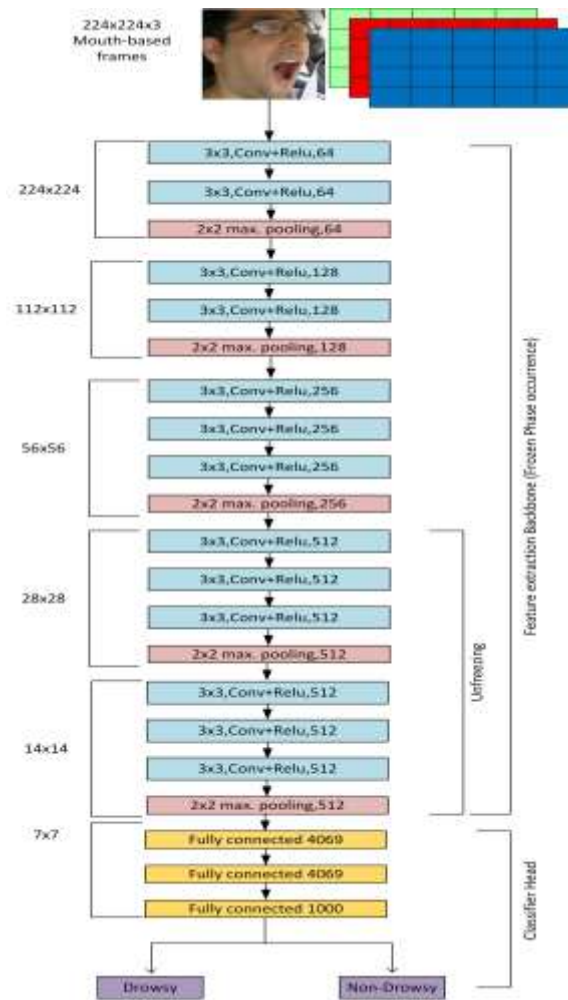


Fig. 5: VGG-16: Structure for classifying yawning-based frames.

4.3.3 EfficientNet-B3 for classifying head position-based frames

For the head-position classification task, input RGB frames are standardized to a resolution of (300×300×3). These frames are subsequently normalized using the preprocess input function, which aligns the data distribution with ImageNet statistics. During the initial training stage, EfficientNet-B3 backbone, pre-trained on ImageNet, is employed as a static feature extractor with its parameters frozen.

EfficientNetB3 architecture utilizes a compound-scaled topology, commencing with a stride equal to 2 stem convolutions, followed by a sequence of MBConv blocks.

For the specified (300×300) input, the B3 backbone has a total stride of 32, yielding a terminal feature tensor with dimensions of (10 ×10×1536). The classifier head applies global average pooling to produce a 1536 flattened array, then uses dropout for regularization and a final Sigmoid function.

Training follows the same two-phase protocol used with ResNet-50 and VGG-16, in which the freezing phase freezes EfficientNet-B3 backbone and only the classifier head is trained, while the fine-tuning phase unfreezes block6 and block7.

Validation loss is monitored with early stopping at a patience of six epochs and a reduced learning rate. The mini-batches comprise 64 frames, and model selection is estimated via 5-fold stratified cross-validation on the training set. Fig. 6 illustrates the structure of the model used to classify head-position-based frames as either drowsy or non-drowsy.

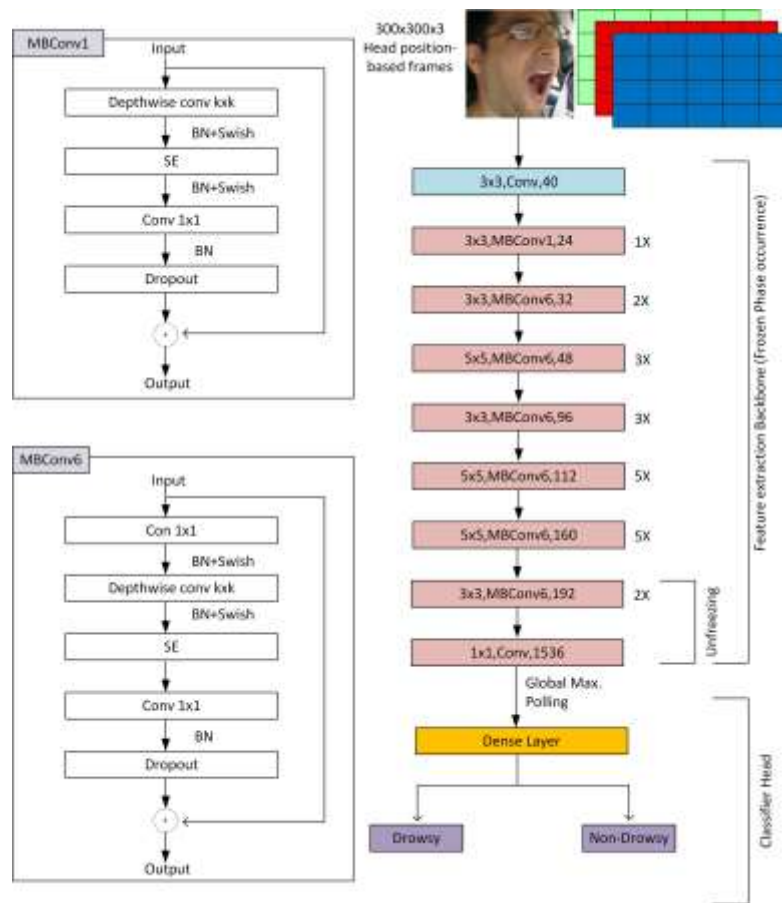


Fig. 6: EfficientNet-B3 architecture for classifying head-position-based frames.

5. System setup and experimental results

This section presents the results with a comprehensive analysis obtained from training three deep learning models for driver drowsiness detection.

5.1 System setup

All three network models, along with the meta-learner PSOF, have been trained and evaluated using Colab Pro+, Python 3, A100 GPU, and system RAM, GPU RAM, and disk with 167GB, 80GB, and 235GB, respectively. Furthermore, the proposed model was implemented on the Google Colab Pro+ environment as the integrated development environment (IDE). The implementation was carried out in Python 3.11, supported by several external libraries, including Keras, OpenCV, NumPy, Matplotlib, MediaPipe, PyTorch, and TensorFlow. These are known for their comprehensive documentation, active developer community, and ease of installation and use.

5.2 The Experimental results of the three models

In this section, the results obtained from applying the three models and the ensemble model were presented. Each model is assessed utilizing the performance metrics, which are precision, recall, accuracy, and F1-score, which collectively measure both predictive effectiveness and generalization capability.

ResNet50 was trained and tested to obtain an accurate assessment of its predictive performance. To evaluate the model's robustness under different data conditions, during the training phase, the validation reports and plotting are demonstrated for all folds. The performance of the ResNet50 in the eye-state-based drowsiness detection classifier was evaluated using the metrics of accuracy, precision, recall, F1-score, Macro average, and weighted average. The quantitative results obtained from testing the data on the best fold out of 5-fold are summarized in Table 3.

Table 3: Results of evaluating the testing set on ResNet50 best fold.

	Precision	Recall	F1-score	Support

Non_Drowsy	0.91	0.88	0.90	1024
Drowsy	0.94	0.96	0.95	1976
Accuracy				0.93
Macro avg	0.93	0.92	0.92	3000
Weighted avg	0.93	0.93	0.93	3000

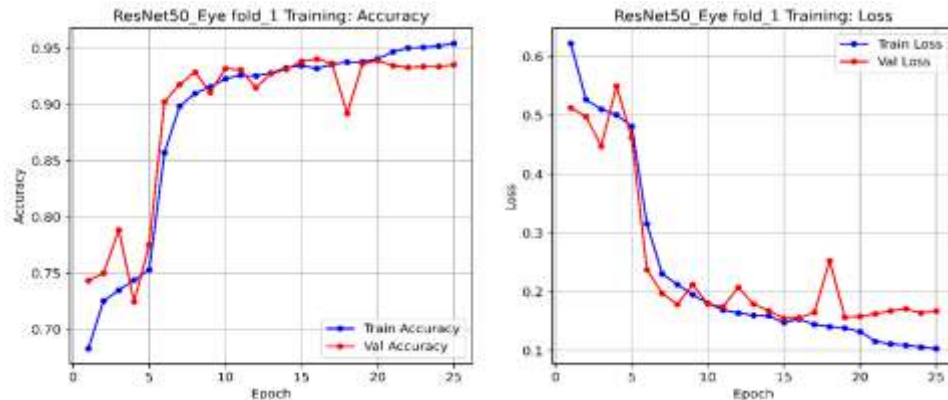


Fig. 7: ResNet50 best fold accuracy and loss vs epochs.

Fig. 7 illustrates the best fold training stream by plotting the accuracy of training and validation versus epochs and the loss values of training and validation versus epochs. From these curves, a consistent convergence trend is observed, with both the accuracy of training and validation increasing steadily throughout the early epochs and stabilizing after the 20th epoch. The validation accuracy remained highly consistent with the training accuracy throughout all fold stages, indicating minimal overfitting.

Overall, the model's learning ability was efficient, achieving an average accuracy of 93% to 95% across all fold stages. The close alignment between the training and validation curves suggests the model's effectiveness in extracting the eye region.

Fig. 8 presents CM of the final test set of the ResNet50 model. The results demonstrate a consistently high classification performance with the majority of samples accurately identified as either Drowsy or Non-Drowsy. This confirms model's robustness, with 1,891 true positives, 899 true negatives, 85 false negatives, and 125 false positives. This translates to a high recall for drowsiness detection, indicating the model's strong ability to correctly identify fatigued drivers.

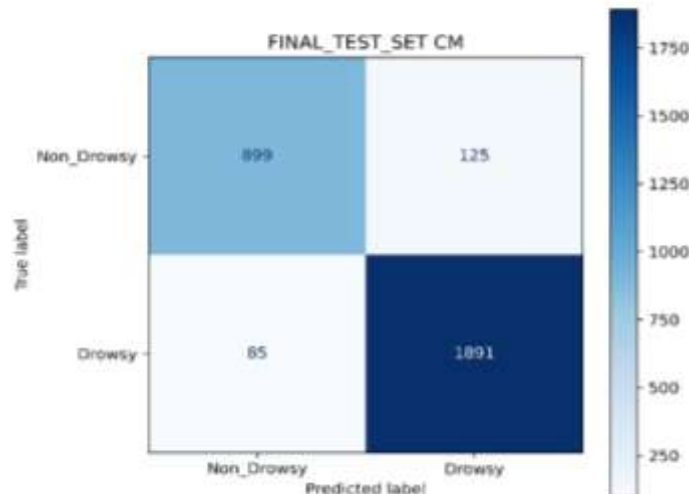


Fig. 8: ResNet50 confusion matrices for test set.

The quantitative results obtained from testing the data on the best validation fold is summarized in Table 4.

Table 4: Results of evaluating the testing set on VGG16 best fold.

	Precision	Recall	F1-score	Support
Non Drowsy	0.95	0.94	0.94	407
Drowsy	0.99	0.99	0.99	2593
Accuracy				0.98
Macro avg	0.97	0.96	0.97	3000
Weighted avg	0.98	0.98	0.98	3000

Fig. 9 illustrates the best fold obtained training stream. In the figure, the model exhibits a rapid and stable convergence, achieving a high validation accuracy of nearly 98% after approximately 7–10 epochs. The training accuracy increases gradually, while the validation accuracy rises sharply during the epochs. The validation curves slightly surpass the training curves in several folds, suggesting that the model generalizes well without significant overfitting

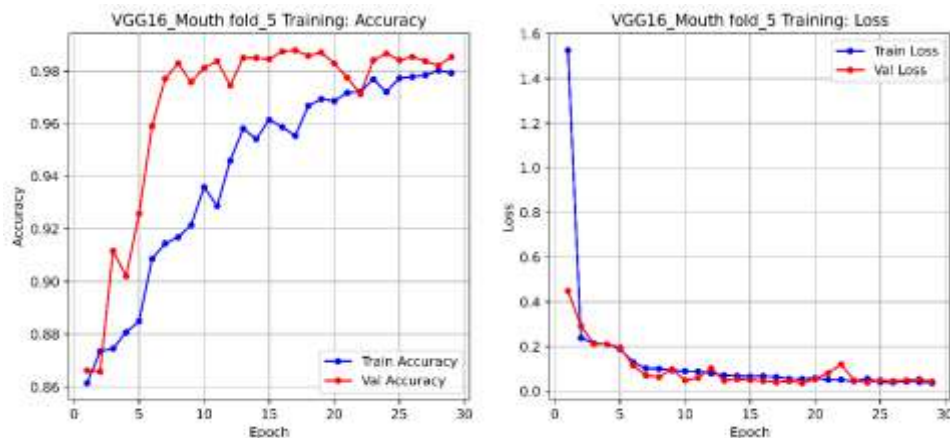


Fig. 9: VGG16 best accuracy and loss Vs epochs.

Similarly, the loss curves show a steep decline during the initial epochs, converging to near-zero values for both training and validation sets. The minimal fluctuations in validation loss confirm stable learning and effective optimization throughout the training process. Overall, the VGG16 shows consistent performance across the five folds, indicating robustness and reliability in capturing mouth-related features for drowsiness detection.

Fig. 10 shows the confusion matrices of the final test set of the VGG16 model. On the final test set, the model correctly classified 2,574 drowsy and 381 non-drowsy samples, with only 45 total misclassifications, achieving near-perfect precision and recall. This indicates that the model maintains remarkable stability and generalization across unseen data.

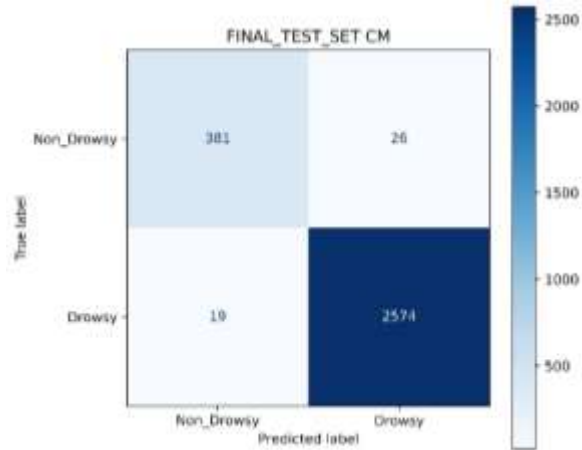


Fig. 10: VGG16 confusion matrices for test set.

The quantitative results obtained from testing the data on the best validation fold is summarized in Table 5.

Table 5: Results of evaluating the testing set on EfficientNetB3.

	Precision	Recall	F1-score	Support
Non_Drowsy	0.92	0.84	0.88	956
Drowsy	0.93	0.97	0.95	2044
Accuracy			0.93	3000
Macro avg	0.93	0.90	0.91	3000
Weighted avg	0.93	0.93	0.92	3000

Fig. 11 illustrates the entire best fold training stream by plotting the training and validation accuracy versus the number of epochs and the training and validation loss values versus the number of epochs. In the figure, the model exhibits good convergence with stable and high validation accuracy, demonstrating strong learning efficiency and generalization. This highlights the robustness of the EfficientNetB3 architecture when applied to head-position features for driver drowsiness detection, achieving fast learning, stable, and exceptional generalization capability.

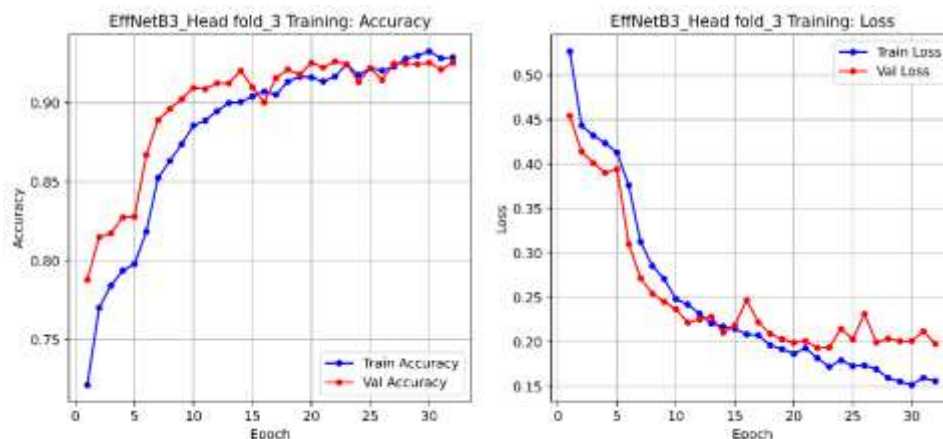


Fig. 11: EfficientNetB3 best fold accuracy and loss Vs epochs.

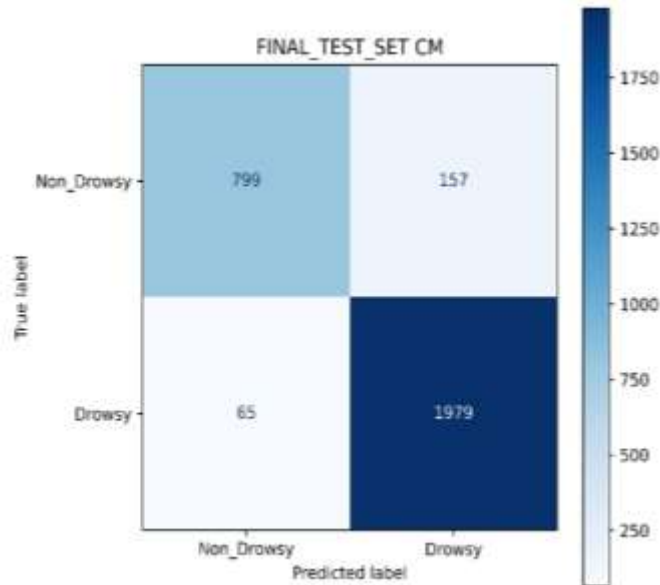


Fig. 12: EfficientNetB3 confusion matrices for the test set.

5.3 The Experimental results of the ensemble model

A weighted average method was employed to compute the final decision out of the three model schemes to classify each frame according to the driver's drowsiness state. 3000 testing frames were used to assess the performance of the weighted average-based ensemble model in correctly classifying the predicted label.

The classification report obtained is shown in Table 6. The results show that the ensemble model performed well in classifying both labels. The obtained recall and precision are 0.98 and 0.99, respectively. The Drowsy class demonstrated the high sensitivity of the model to identifying drowsiness with few FN. With a recall of 0.98 and a precision of 0.91, the non-drowsy class also performs well, demonstrating a trustworthy differentiation. Despite the small data imbalance, the macro and weighted averages show constant performance across classes, and an overall accuracy of 0.98 validates solid generalization on the test set.

Table 6: Results of evaluating the testing set on the ensemble model.

	Precision	Recall	F1-score	Support
Non-Drowsy	0.91	0.98	0.94	456
Drowsy	0.99	0.98	0.99	2544
Accuracy			0.98	3000
Macro avg	0.96	0.98	0.97	3000
Weighted avg	0.97	0.98	0.98	3000

Fig. 13 shows the confusion matrix obtained from evaluating the model with the testing dataset. The confusion matrix shows that weighted average ensemble model achieved strong classification consistency. Out of all samples, 449 non-drowsy and 2497 drowsy frames were correctly identified, while only 7 non-drowsy and 47 drowsy frames were misclassified. This distribution confirms high precision and recall for both classes, demonstrating the model's ability to minimize false detections and maintain reliable performance across alert and fatigued states.

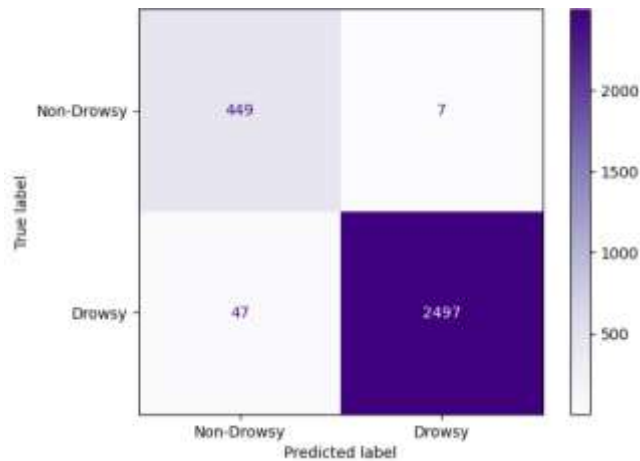


Fig. 13: The ensemble model confusion matrix.

5.3.1 Classification model result on nighttime frames.

To assess the ensemble model performance at night, 4000 frames have been captured from nighttime videos, which were typically observed during nocturnal driving. The resulting probabilities of eye, mouth, and head position-based values outputs from the three models were averaged. This nighttime subset serves as an independent evaluation set to quantify the robustness and generalization capability of the proposed framework under reduced illumination and increased visual noise.

Table 7 outlines the classification report obtained from testing the mode by utilizing the nighttime frames.

Table 7: Results of evaluating the nighttime testing set on ensemble model.

	Precision	Recall	F1-score	Support
Non-Drowsy	0.90	0.97	0.94	1261
Drowsy	0.99	0.95	0.97	2739
Accuracy			0.96	4000
Macro avg	0.94	0.96	0.95	4000
Weighted avg	0.96	0.96	0.96	4000

The model demonstrates strong discriminative capability across both classes, achieving an overall accuracy of 96% on the 4000-nighttime evaluation set.

The non-drowsy class achieves a precision of 0.90, indicating that most frames predicted as awake are indeed correct, and a recall of 0.97, showing that almost all true alert frames are successfully detected. For the Drowsy class, the system attains a precision of 0.99, reflecting an exceptionally low false-positive rate, and a recall of 0.95, confirming reliable detection of drowsy behavior. The corresponding F1-scores of 0.94 and 0.97 for the two classes further highlight the balance between precision and recall. Overall, the model demonstrates effectiveness in distinguishing between alert and drowsy states under nighttime conditions.

Fig. 14 shows the confusion matrix obtained from evaluating the model utilizing the 4000-nighttime evaluation set. In the figure, the model performs reliably across both classes. For the non-drowsy class, the system correctly identifies 1228 frames, misclassifying only 33 frames as drowsy. This reflects the ability of the model to avoid false alarms for alert drivers. For the Drowsy class, 2606 frames are correctly detected, while 133 are incorrectly labeled. Although this denotes a small number of missed detections, the overall trend shows consistent recognition of drowsiness even under nighttime conditions.

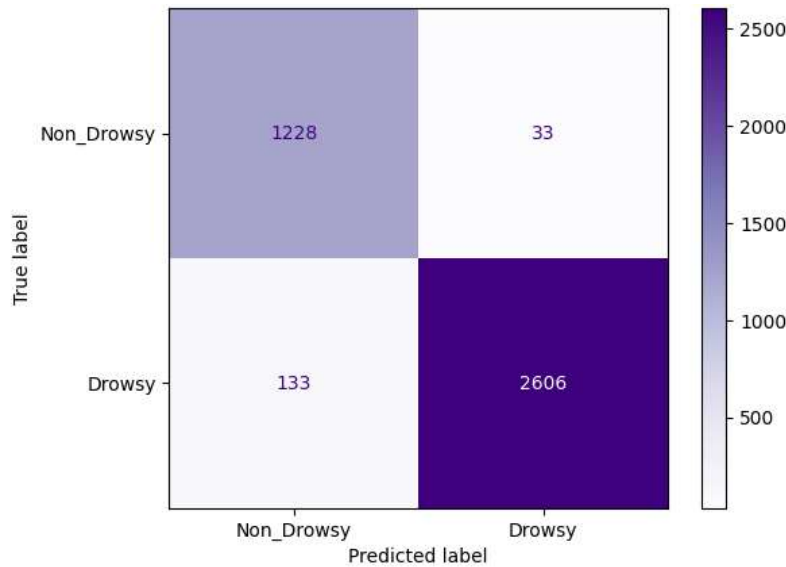


Fig. 14: The ensemble model confusion matrix on nighttime frames.

5.3.2 Classification model result on day frames.

To evaluate the ensemble model under daytime conditions, a total of 4000 frames were extracted from daytime driving videos. Table 8 outlines the classification report obtained from testing the mode by utilizing the daytime frames. The model shows an effective performance as it gains an accuracy of 98% on 4000 daytime frames.

The model achieves a precision of 0.92 for the non-drowsy frames. This confirms that most frames predicted as awake are correctly classified. The model achieves a recall of 0.99 which suggests that nearly all non-drowsy frames are successfully detected.

Table 8: Results of evaluating the daytime testing set on ensemble model.

	Precision	Recall	F1-score	Support
Non-Drowsy	0.92	0.99	0.95	741
Drowsy	0.99	0.98	0.99	3259
Accuracy				0.98
Macro avg	0.96	0.98	0.97	4000
Weighted avg	0.98	0.98	0.98	4000

The model achieves a precision of 0.99 for the drowsy frames. This confirms a low FP rate. The model achieves a recall of 0.98 which suggests reliability in detecting drowsy behaviors. The model achieves an F1 score of 0.95 and 0.99 for the non-drowsy and drowsy frames, respectively. This indicates a balance between precision and recall, and further shows effectiveness in distinguishing between both classes under daytime conditions.

Fig. 15 shows CM resulting from evaluating utilizing 4000-daytime frame set. In the figure, the model performs reliably across both classes. The model correctly identifies 733 frames for the non-drowsy class while misclassifying only 8 frames. For the drowsy class, 3195 frames are correctly identified, whereas only 64 frames are misclassified. Although this represents a small number of missed detections, the results demonstrate that the model consistently recognizes drowsiness even under nighttime conditions.

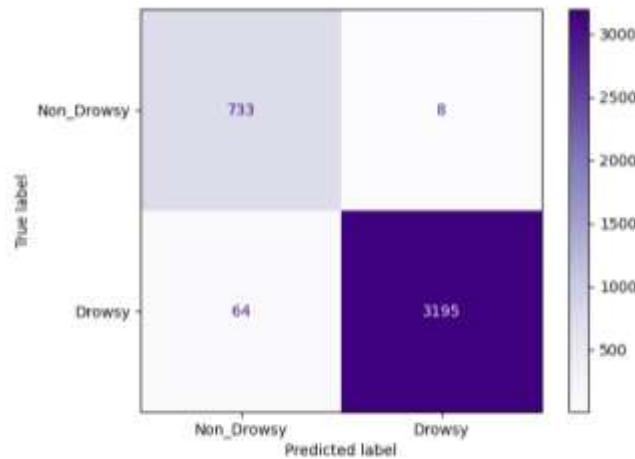


Fig.15: The ensemble model confusion matrix on daytime frames.

5.3.3 Comparison with State-of-the-art Results

The models and the ensemble model have been compared to three state-of-the-art models, which were previously discussed in chapter one, seeking the highest accuracy in DDD task. Table 9 outlines the results of the proposed model in comparison with previous studies in driver drowsiness detection.

Table 9: Comparison results against the competitor schemes.

Author	year	Dataset	Accuracy
Yang & Yi [26]	2024	YawDD	95%
Majeed et al. [23]	2023	YawDD	96%
Janarthanan et al. [27]	2025	NITYMED	96%
Proposed	2025	NITYMED & YawDD	98%

This table summarizes the various studies that has been developed to identify drowsiness using the different DL approaches. From the table, it can be seen that the proposed ensemble model outperformed the competitors method in terms of accuracy in detecting the drowsy drivers. This is due to the appropriate consideration of the core DL methods, including the ResNet50, VGG16, and EfficientNetB3, and further to the freezing-based training.

6. Conclusion

The driver drowsiness problem is considered a new topic that has received increased attention in recent years. Therefore, this thesis has proposed a novel DDD system that employs a deep-learning ensemble model to detect driver drowsiness states. The designed ensemble framework integrates three CNNs that use two combined datasets of NITYMED and YAWDD in order to train is capturing the eye state, yawing behavior, and head pose estimation, and thus fuses their sigmoid outputs by weighted average to achieve robust and reliable detection across diverse conditions. The experimental results demonstrated that the proposed DDD system achieves high accuracy under daytime, nighttime, and combined testing scenarios, outperforming several existing state-of-the-art approaches. This confirms the importance of using ensemble decision-making to capture the facial behavioral features that single-modal systems often can't extract or precisely classify.

References

- [1] M. E. Shaik, "A systematic review on detection and prediction of driver drowsiness," *Transportation research interdisciplinary perspectives*, vol. 21, p. 100864, 2023.
- [2] M. Ahmed, S. Masood, M. Ahmad, and A. A. Abd El-Latif, "Intelligent driver drowsiness detection for traffic safety based on multi CNN deep model and facial subsampling," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 10, pp. 19743-19752, 2021.

- [3] E. Civik and U. Yuzgec, "Real-time driver fatigue detection system with deep learning on a low-cost embedded system," *Microprocessors and Microsystems*, vol. 99, p. 104851, 2023.
- [4] C. Anderson et al., "Feeling sleepy? stop driving—awareness of fall asleep crashes," *Sleep*, vol. 46, no. 11, p. zsad136, 2023.
- [5] Y. Albadawi, M. Takruri, and M. Awad, "A review of recent developments in driver drowsiness detection systems," *Sensors*, vol. 22, no. 5, p. 2069, 2022.
- [6] L. Masello, G. Castignani, B. Sheehan, F. Murphy, and K. McDonnell, "On the road safety benefits of advanced driver assistance systems in different driving contexts," *Transportation research interdisciplinary perspectives*, vol. 15, p. 100670, 2022.
- [7] S. A. El-Nabi, W. El-Shafai, E.-S. M. El-Rabaie, K. F. Ramadan, F. E. Abd El-Samie, and S. Mohsen, "Machine learning and deep learning techniques for driver fatigue and drowsiness detection: a review," *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 9441-9477, 2024.
- [8] A. Rajkar, N. Kulkarni, and A. Raut, "Driver drowsiness detection using deep learning," in *Applied Information Processing Systems: Proceedings of ICCEET 2021*: Springer, 2021, pp. 73-82.
- [9] Y. Suresh, R. Khandelwal, M. Nikitha, M. Fayaz, and V. Soudhri, "Driver drowsiness detection using deep learning," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021: IEEE, pp. 1526-1531.
- [10] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, "A survey on state-of-the-art drowsiness detection techniques," *IEEE Access*, vol. 7, pp. 61904-61919, 2019.
- [11] I.-H. Choi, S. K. Hong, and Y.-G. Kim, "Real-time categorization of driver's gaze zone using the deep learning techniques," in *2016 International conference on big data and smart computing (BigComp)*, 2016: IEEE, pp. 143-148.
- [12] S. A. Dwivedi, A. Attry, and K. Singla, "Leveraging transfer learning for driver drowsiness detection," in *Advances in Data and Information Sciences: Proceedings of ICDIS 2021*: Springer, 2022, pp. 603-611.
- [13] X. Cui, X. Li, X. Zheng, and Y. Ren, "Driving behavior primitive classification using CNN-based fusion models," *IEEE Access*, vol. 12, pp. 56344-56355, 2024.
- [14] M. Muzammel, M. Z. Yusoff, M. N. M. Saad, F. Sheikh, and M. A. Awais, "Blind-Spot Collision Detection System for Commercial Vehicles Using Multi Deep CNN Architecture," *Sensors*, vol. 22, no. 16, p. 6088, 2022.
- [15] V. U. Maheswari, R. Aluvalu, M. P. Kantipudi, K. K. Chennam, K. Kotecha, and J. R. Saini, "Driver drowsiness prediction based on multiple aspects using image processing techniques," *IEEE Access*, vol. 10, pp. 54980-54990, 2022.
- [16] M. Park, H. Kim, and S. Park, "A convolutional neural network-based end-to-end self-driving using LiDAR and camera fusion: Analysis perspectives in a real-world environment," *Electronics*, vol. 10, no. 21, p. 2608, 2021.
- [17] T. Ahmed, O. Jyoti, and T. H. Mou, "Drivers' drowsiness detection system enhancement using deep learning: CNN-based approach," in *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 2023: IEEE, pp. 1-6.
- [18] S. Yoheswari, "OPTIMIZED DRIVER DROWSINESS DETECTION USING MACHINE LEARNING TECHNIQUES," 2024.
- [19] W. Deng and R. Wu, "Real-time driver-drowsiness detection system using facial features," *Ieee Access*, vol. 7, pp. 118727-118738, 2019.
- [20] S. Das, S. Pratihari, B. Pradhan, R. H. Jhaveri, and F. Benedetto, "IoT-assisted automatic driver drowsiness detection through facial movement analysis using deep learning and a U-Net-based architecture," *Information*, vol. 15, no. 1, p. 30, 2024.
- [21] M. I. B. Ahmed et al., "A deep-learning approach to driver drowsiness detection," *Safety*, vol. 9, no. 3, p. 65, 2023.
- [22] A.-C. Phan, N.-H.-Q. Nguyen, T.-N. Trieu, and T.-C. Phan, "An efficient approach for detecting driver drowsiness based on deep learning," *Applied Sciences*, vol. 11, no. 18, p. 8441, 2021.
- [23] F. Majeed, U. Shafique, M. Safran, S. Alfarhood, and I. Ashraf, "Detection of drowsiness among drivers using novel deep convolutional neural network model," *Sensors*, vol. 23, no. 21, p. 8741, 2023.
- [24] M. Dua, Shakshi, R. Singla, S. Raj, and A. Jangra, "Deep CNN models-based ensemble approach to driver drowsiness detection," *Neural Computing and Applications*, vol. 33, no. 8, pp. 3155-3168, 2021.
- [25] R. Florez, F. Palomino-Quispe, A. B. Alvarez, R. J. Coaquira-Castillo, and J. C. Herrera-Levano, "A Real-Time Embedded System for Driver Drowsiness Detection Based on Visual Analysis of the Eyes and Mouth Using Convolutional Neural Network and Mouth Aspect Ratio," *Sensors*, vol. 24, no. 19, p. 6261, 2024.
- [26] E. Yang and O. Yi, "Enhancing road safety: Deep learning-based intelligent driver drowsiness detection for advanced driver-assistance systems," *Electronics*, vol. 13, no. 4, p. 708, 2024.

- [27] A. Janarthanan, A. Paramarthalingam, S. Sundaresan, and S. Yokeshvaran, "Intelligent Drowsiness Detection using Haar Cascade Classifier and Convolutional Neural Network," *Journal of Ubiquitous Computing and Communication Technologies*, vol. 7, no. 2, pp. 176-194, 2025.
- [28] L. Wen, Z. Xiao, X. Xu, and B. Liu, "Disaster Recognition and Classification Based on Improved ResNet-50 Neural Network," *Applied Sciences*, vol. 15, no. 9, p. 5143, 2025.
- [29] C.-Y. Lee and T.-A. Le, "Identifying faults of rolling element based on persistence spectrum and convolutional neural network with ResNet structure," *Ieee Access*, vol. 9, pp. 78241-78252, 2021.
- [30] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999-7019, 2021.
- [31] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello, "Comparing inception V3, VGG 16, VGG 19, CNN, and ResNet 50: a case study on early detection of a rice disease," *Agronomy*, vol. 13, no. 6, p. 1633, 2023.
- [32] A. Batool and Y.-C. Byun, "Lightweight EfficientNetB3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images," *IEEE access*, vol. 11, pp. 37203-37215, 2023.
- [33] S. Abd El-Ghany, M. Elmogy, and A. A. El-Aziz, "Computer-aided diagnosis system for blood diseases using efficientnet-b3 based on a dynamic learning algorithm," *Diagnostics*, vol. 13, no. 3, p. 404, 2023.
- [34] G. Kaur, N. Sharma, and R. Gupta, "Wheat leaf disease classification using EfficientNet B3 pre-trained architecture," in *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, 2023: IEEE, pp. 1-5.
- [35] G. George and S. Anusuya, "Leveraging EfficientNet-B3 with Advanced Fine-Tuning for Precise Breast Cancer Classification," in *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2024: IEEE, pp. 938-944.
- [36] N. Alshdaifat et al., "Automated blood cancer detection models based on efficientnet-b3 architecture and transfer learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 36, no. 3, pp. 1731-1738, 2024.
- [37] C. Szegedy et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
- [38] D. Yang, C. Martinez, L. Visuña, H. Khandhar, C. Bhatt, and J. Carretero, "Detection and analysis of COVID-19 in medical images using deep learning techniques," *Scientific Reports*, vol. 11, no. 1, p. 19638, 2021.
- [39] W. Nash, T. Drummond, and N. Birbilis, "A review of deep learning in the study of materials degradation," *npj Materials Degradation*, vol. 2, no. 1, p. 37, 2018.
- [40] A. H. Khaleel, T. H. Abbas, and A.-W. Sami Ibrahim, "Best low-cost methods for real-time detection of the eye and gaze tracking," *i-com*, vol. 23, no. 1, pp. 79-94, 2024.