https://doi.org/10.31436/iiumej.v26i2.3488

COMPARATIVE ANALYSIS OF VISION TRANSFORMERS AND CNN MODELS FOR DRIVER FATIGUE CLASSIFICATION

FADHLAN HAFIZHELMI KAMARU ZAMAN*, NG KOK MUN, SYAHRUL AFZAL CHE ABDULLAH

Vehicle Intelligence and Telematics Lab, Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor

**Corresponding author: fadhlan@uitm.edu.my*

(Received: 30 October 2024; Accepted: 28 January 2025; Published online: 15 May 2025)

ABSTRACT: This study provides a comprehensive evaluation of Convolutional Neural Network (CNN) and Vision Transformer (ViT) models for driver fatigue classification, a critical issue in road safety. Using a custom driving behavior dataset, state-of-the-art CNN and ViT architectures, including VGG16, EfficientNet, MobileNet, Inception, DenseNet, ResNet, ViT, and Swin Transformer, were analyzed in this study to determine the best model for practical driver fatigue monitoring systems. Performance metrics such as accuracy, F1-score, training time, inference time, and frames per second (fps) were assessed across different hardware platforms, including a high-performance workstation, Raspberry Pi 5, and a desktop with a Graphic Processing Unit (GPU). Results demonstrate that CNN models, particularly VGG16, achieve the best balance between accuracy and efficiency, with an F1-score of 0.97 and 77.00 fps on a desktop. On the other hand, Swin V2S outperforms all models in terms of accuracy, achieving an F1-score of 0.99 and 61.18 fps on a GPU, although it exhibits limited efficiency on embedded systems. This study significantly contributes by providing practical recommendations for selecting models based on performance needs and hardware constraints, highlighting the suitability of ViTs for high-computation environments. The findings support the development of more efficient driver fatigue monitoring systems, offering practical implications for enhancing road safety and reducing traffic accidents.

ABSTRAK: Kajian ini merupakan penilaian komprehensif terhadap model Konvolusi Rangkaian Neural (CNN) dan Transformer Penglihatan (ViT) bagi pengelasan keletihan pemandu, iaitu satu isu kritikal dalam keselamatan jalan raya. Menggunakan set data tingkah laku pemanduan tersuai, seni bina terkini CNN dan ViT, termasuk VGG16, EfficientNet, MobileNet, Inception, DenseNet, ResNet, ViT dan Transformer Swin dianalisa dalam kajian ini bagi menentukan model terbaik bagi sistem pemantauan keletihan pemandu yang praktikal. Metrik prestasi seperti ketepatan, skor F1, masa latihan, masa inferens, dan bingkai sesaat (fps) telah dinilai merentasi pelbagai platfom perkakasan, termasuk stesen kerja berprestasi tinggi, Raspberry Pi 5, dan komputer meja dengan Unit Pemprosesan Grafik (GPU). Dapatan kajian menunjukkan bahawa model CNN, khususnya VGG16, mencapai keseimbangan terbaik antara ketepatan dan kecekapan, dengan skor F1 sebanyak 0.97 dan 77.00 fps pada komputer meja. Sebaliknya, Swin V2S mengatasi semua model dari segi ketepatan, mencapai skor F1 sebanyak 0.99 dan 61.18 fps pada GPU, walaupun menunjukkan kecekapan yang terhad pada sistem terbenam. Kajian ini memberikan sumbangan yang signifikan dengan menyediakan cadangan praktikal bagi pemilihan model berdasarkan keperluan prestasi dan kekangan perkakasan, serta menonjolkan kesesuaian ViT bagi persekitaran berkomputasi tinggi. Penemuan ini menyokong pembangunan sistem pemantauan keletihan pemandu yang lebih cekap, dengan implikasi praktikal bagi meningkatkan keselamatan jalan raya dan mengurangkan kemalangan.

KEYWORDS: *Deep learning, CNN, vision transformer, driving behavior, embedded system, Raspberry Pi*

1. INTRODUCTION

Traffic accidents have claimed the lives of an estimated 1.35 million people annually, or 3,700 people per day. Road accident victims, their families, and entire nations suffer significant financial losses in the event of traffic accidents. In Malaysia, the government has lost at least RM3.12 million for every life, according to MIROS's 2018 Value of Statistical Life (VSOL). On average, Malaysia has 18 fatal road accidents every day, making it a serious public health issue for the nation [1].

Improving vehicle driving safety has become a priority in academia and the automotive industry to lower the risk of vehicle accidents. Numerous elements affect vehicle driving safety, but one of the most important is the driver's health and condition. The driver's emotions, psychology, and physiology all impact this condition. The two most important negative indicators of the driver's state are fatigue and distraction. According to studies, 36% of highway fatalities are caused by exhaustion and distraction [2]. There is a negative impact of fatigue and sleep-related safety risks on driving performance [3].

As a result of the rapid advancement of technologies like machine vision and deep learning [4], it is now a popular research area to use images and videos to identify driver fatigue and attention states to reduce the risk of vehicle accidents. One of the deep learning algorithms that is commonly used for image and video recognition is the Convolutional Neural Network (CNN) [5], [6]. It uses convolutional layers to automatically and adaptively learn the spatial hierarchies of features in grid-like data, such as images and photographs. Convolutional, pooling, and fully connected layers are some of the layers that CNNs use to identify patterns and structures in visual input. Another popular deep learning architecture intended for image classification is called a Vision Transformer (ViT) [7], [8]. ViTs employ transformer models, which were first created for natural language processing, as opposed to Convolutional Neural Networks (CNNs), which employ convolutional layers. To capture global context and dependencies, they split an image into fixed-size patches, insert them into sequences, and process them using self-attention techniques. Because of this, Vision Transformer can perform image classification tasks with excellent accuracy.

There are various state-of-the-art architectures for CNN, such as ResNet, VGGNet, DenseNet, InceptionNet, MobileNet, and EfficientNet. At the same time, Vision Transformer has popular architectures such as ViT and Swin Transformer. In numerous previous studies, CNN and ViT have been used recently to classify driving behaviors. Poon et al. developed a non-contact driving behaviour detection system for the improvement of driving safety using a YOLO-based CNN architecture [9] whereas ResNet50 has been shown to produce a very good performance in driving behavior classification for night driving [10]. On the other hand, Lian et al. proposed Stargazer, a straightforward yet powerful action temporal localization framework that utilises rich temporal aspects about human behavioural data. Stargazer is an efficient ViT-based system that has produced good performance and shown the efficacy of our model on the Naturalistic Driver Action Recognition of the AI City Challenge 2022 [11].

Several studies have benchmarked CNNs over ViT to evaluate their performance in many classification tasks, including the recognition of Japanese Sign Language [12], detection of bacterial strains [13], detection of distracted drivers [14], wheat disease classification [15], and COVID-19 classification [16]. These studies found CNNs outperforming ViTs only in COVID-19 classification, while ViTs consistently excelled in other classification tasks. However, the

https://doi.org/10.31436/iiumej.v26i2.3488

computational complexity of ViTs in real-world applications remains unexamined, and their comparative performance with CNNs in classifying driver fatigue behaviors is still unclear. It is also not yet clear whether ViT or CNN will perform better against the other in classifying fatigue driving behaviours. Thus, this paper provides a comprehensive overview and performance analysis of several state-of-the-art CNN architectures and Vision Transformer models for classifying driver fatigue behaviours, such as nodding and yawning. We utilize our driving behavior dataset for benchmarking. The architectures are compared using standard evaluation metrics, including classification accuracy, precision, recall, F1 scores, and model training and inference time on a desktop with a GPU and a Raspberry Pi 5, followed by a detailed analysis and discussion.

The paper is structured as follows: Section II provides an overview of CNNs and Vision Transformers and the models this study considers. Section III reviews previous approaches for detecting distracted driving behaviours while Section IV details the experimental procedures, evaluation metrics, and dataset used. Section V presents our experimental results and analysis. Finally, Section VI offers conclusions.

2. OVERVIEW OF CNN AND ViT ARCHITECTURES

CNNs and ViTs, two well-known deep learning architectures utilised in image classification problems, are examined in this section. Effective feature extraction and visual data classification are made possible by these models' unique structures and processing methods. CNNs are excellent at identifying spatial relationships in images because of their layered approach of convolutional and pooling operations. In contrast, ViTs use transformer-based self-attention techniques, which were first created for natural language processing, to capture global context and long-range dependencies in an image.

2.1. Convolutional Neural Network Architectures

CNNs are deep learning models designed to analyze images and videos. Convolutional layers, activation functions, pooling layers, fully connected layers, and an output layer are among the basic layers that make up its architecture. The convolutional layers apply filters to the input image to extract features like edges, textures, and patterns. The model can learn intricate patterns thanks to the non-linearity introduced by activation functions like ReLU. By downsampling the feature maps, pooling layers minimise computational cost and spatial dimensions while avoiding overfitting. The output layer usually employs a softmax function to calculate probabilities for each class. In contrast, fully connected layers, positioned at the network's end, convert the high-level characteristics into classification results.

A CNN uses several convolutional and pooling layers to extract hierarchical features for image classification, starting with low-level edges and textures and working up to high-level shapes and objects. By modifying the weights of its filters and fully connected layers, the model is trained on a labeled dataset and uses backpropagation to minimize the error between its predictions and actual labels. Once trained, the CNN recognises learnt patterns to classify new images accurately. CNNs are an invaluable tool in computer vision applications because of their ability to effectively analyse visual data and recognise spatial dependencies, which makes them highly successful for image classification. In this work, ResNet [17], EfficientNet [18], Inception [19], VGGNet[20], DenseNet [21], and MobileNet [22] are selected as the CNN architectures to benchmark against Vision Transformer.

- **ResNet (Residual Network)**: ResNet tackles the issue of vanishing gradients in deep networks by introducing the idea of residual learning. To enable the network to learn

https://doi.org/10.31436/iiumej.v26i2.3488

identity mappings, it uses shortcut connections that bypass one or more levels. By promoting gradient flow during backpropagation, this architecture makes it possible to build incredibly deep networks like ResNet-50 and ResNet-101, which achieve great performance.

- **EfficientNet**: EfficientNet is intended to provide excellent accuracy using fewer parameters and lower processing expenses. It uses compound scaling to balance network depth, width, and resolution to optimize speed. Neural Architecture Search (NAS) was used to create the baseline model, EfficientNet-B0, to identify an efficient structure. Due to its Mobile Inverted Bottleneck Convolution (MBConv) blocks and Swish activation function, EfficientNet achieves state-of-the-art accuracy in image classification tasks while being computationally efficient.
- **Inception**: Using the Inception architecture, several convolutional filters of varying widths are applied within a single layer using Inception modules. This lessens the computational load and enables the network to capture various spatial features. These modules are stacked in the design to provide reliable performance and effective learning. Through significant optimisations, Inception-v3 and subsequent versions further increase accuracy and efficiency.
- **VGGNet**: The Visual Geometry Group (VGG) created VGGNet, renowned for its depth and simplicity. Small (3×3) convolutional filters are used throughout the network, and additional layers are stacked to boost depth, as in VGG-16 and VGG-19. This method increases classification performance, but because there are more parameters, it requires more memory and processing power.
- **MobileNet**: MobileNet prioritises efficiency and low computing costs in its design of mobile and embedded vision applications. It uses depth-wise separable convolutions, drastically reducing the number of parameters and calculations by dividing normal convolution into depth-wise and pointwise convolutions. MobileNet is appropriate for resource-constrained environments, with variations like MobileNetV2 and MobileNetV3, which further optimise performance and efficiency.

2.2. Vision Transformer Architectures

Vision Transformer is a new deep learning model designed for image classification that utilizes the transformer architecture originally developed for language processing. To maintain spatial information, it divides an image into fixed-size patches, embeds them into vectors, and subsequently incorporates positional encodings. The output is then utilized by the classification head to predict image classifications. ViTs excel in picture classification tasks, particularly with large datasets, due to their ability to capture global context and long-range dependencies. To benchmark CNNs against Vision Transformer, ViT [8] and Swin Transformer [23] architectures and their variants are selected.

- **Vision Transformer (ViT)**: The Vision Transformer (ViT) is a groundbreaking architecture that employs the transformer model for image classification. It flattens and linearly embeds the fixed-size patches it creates from an input image. To preserve spatial information, these patch embeddings are augmented with positional encodings. A conventional transformer encoder receives the embedded patches and employs multi-head self-attention to capture relationships throughout the image. ViT has demonstrated competitive performance on extensive image classification tasks and is excellent at capturing global context.

- **Swin Transformer**: The Swin Transformer (Shifted Window Transformer) is an improved vision transformer architecture that aims to increase efficiency and scalability. It presents a hierarchical structure that uses a shifted window method to handle images. This approach splits the image into non-overlapping windows to reduce computational complexity and calculates self-attention within each window. The windows are moved between layers to facilitate cross-window connections and allow the model to gather local and global information efficiently. Swin Transformer is particularly effective for various vision applications, including segmentation, object detection, and image classification.

3. RELATED WORKS

3.1. CNN-Based Behavior Classification Approaches

Li et al. introduced the Deep Multichannel Network Model (DMNM) framework [24], which processes real driving data divided into external, internal, and habit dimensions using fully connected layers and CNNs. The model achieves high classification accuracy (95%) by filtering correlations between factors and integrating multidimensional data. DMNM includes a comprehensive analysis and effective correlation elimination. However, it also presents challenges like data processing and model training complexity, potential overfitting due to high feature dimensionality, and significant computational resource requirements. Multiple CNN models have also been used to detect driving behaviors under different lighting conditions. The model uses MobileNetV2, GoogleNet, InceptionV3, and ResNet-50 to analyze various driving patterns, distinguishing between day and night driving behaviors. The data for training the models includes video and sensor inputs collected from actual driving scenarios [25].

A 3D CNN called DriftNet has also been used in detecting aggressive driving behaviors, particularly car drifting, from video data [26]. The implementation employs DenseNet architecture to effectively learn spatial and temporal features from traffic videos. The model was trained on a custom dataset of car drifting clips collected from YouTube, achieving a validation accuracy of 77.5%. Among its advantages, DriftNet demonstrates high accuracy and effectiveness in learning complex features, utilizes transfer learning to enhance performance on limited data, and achieves superior validation accuracy compared to other models tested.

Qu et al. developed HAR-Net, a deep learning model to detect dangerous driving behaviors such as eating, drinking, smoking, and phone use [27]. The model processes optical flow, RGB, and RGBD data separately, integrating them through spatial-temporal fusion. It combines ResNet-50 with an hourglass network and incorporates an attention mechanism, achieving a mean average precision (mAP) of 98.84% on the constructed dataset [27]. Octave-Like Convolutional Neural Network (OLCMNet) [28] has been proposed for detecting driver distraction, where its main implementation focuses on incorporating a high-frequency branch and two low-frequency branches in the OLCM block, enhancing the network's ability to capture diverse feature information. The results demonstrate that the OLCMNet achieves better overall accuracy than other networks, particularly on the StateFarm and LDDB datasets. The advantages of the OLCMNet include its superior accuracy and effective feature capture. At the same time, its disadvantages are the increased computational cost and the need for careful tuning of hyperparameters to balance accuracy and speed.

3.2. Vision Transformer-Based Behavior Classification Approaches

Liang et al. introduced a system that uses a transformer-based approach to detect driver actions from in-vehicle video footage [11]. Stargazer leverages an improved multi-scale vision

https://doi.org/10.31436/iiumej.v26i2.3488

transformer (MViT) network to learn hierarchical representations of driver actions. It employs a sliding-window classification strategy for accurate temporal localization. The work shows that re-training on large-scale video action datasets and multi-crop data augmentation enhances training efficiency and model robustness. The advantages of this method include high accuracy in detecting and localizing driver actions and effective temporal feature capture. However, the system has high computational requirements and faces challenges in real-time deployment on vehicle onboard devices.

Similarly, a hierarchical vision transformer with shifted windows has been proposed to detect distracted driving behaviors [29]. This model leverages the Swin Transformer architecture, fine-tuned with pre-trained weights on the ImageNet dataset, to classify distracted drivers accurately. The implementation includes a detailed analysis of the model's performance on the AUC Distracted Driver Dataset, achieving a classification accuracy of 95.72%. This approach achieved high accuracy in distracted driver detection and has efficient computation due to limited self-attention calculations within local windows.

Besides, ViT and CNN have been used in a hybrid approach that combines the strengths of both methods to detect driver distraction [30]. The proposed model, FPT, integrates the Twins Transformer framework with redesigned residual embedding and lightweight group convolution modules, reducing computational complexity and improving feature extraction capabilities. This hybrid approach enhances the model's ability to capture local and global features, which is crucial for accurately identifying distracted driving behaviors. The implementation includes a cross-entropy loss function with label smoothing to enhance model learning and prevent overfitting. The model's effectiveness is demonstrated on two large-scale driver distraction datasets, achieving high accuracy and stability. However, the complexity of the transformer architecture and the need for substantial computational resources are noted as disadvantages, requiring further optimization for practical deployment.

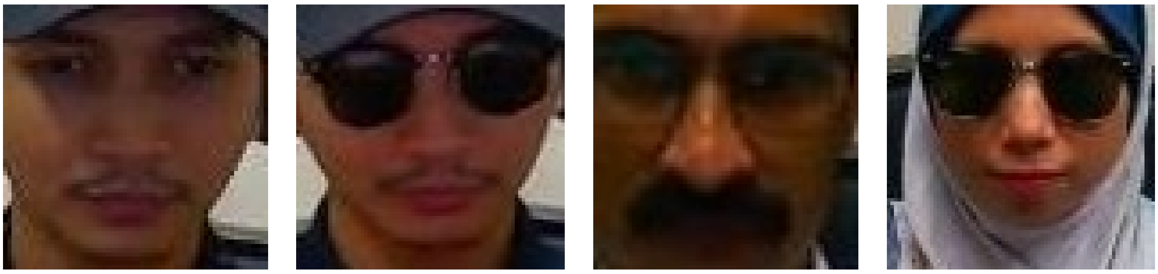
4. EXPERIMENTS

This section provides details of the data preparation and experimental setup used to evaluate CNNs and ViTs. Evaluation is conducted during model development and deployment to assess the models' performance.

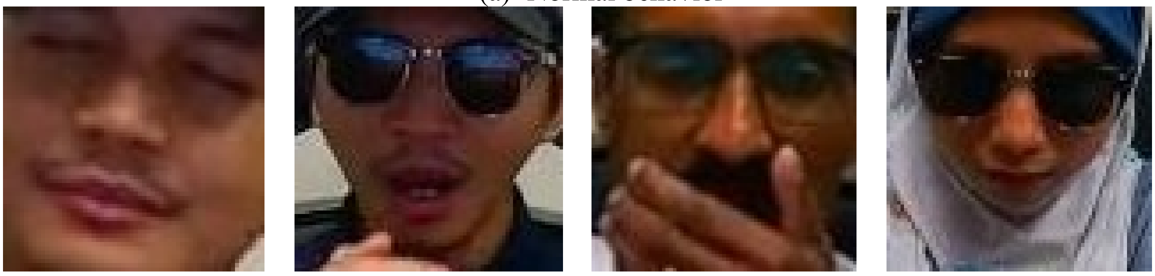
4.1. Data Preparation

To enable the system to classify fatigue behaviors, the model was trained using two classes of driving behaviors: normal and fatigue. Figure 1 presents examples from each behavior class. This targeted classification approach allows the model to focus on specific behaviors, enhancing its learning efficiency. The datasets, encompassing normal, yawning, and drowsiness behaviors, were collected in the VITAL Laboratory, Universiti Teknologi MARA, where 44 participants engaged in a driving simulation designed to mimic real-life conditions. Each participant simulated scenarios aligned with the predefined categories of normal driving, yawning, and drowsiness. An essential consideration during data collection was ensuring data diversity. Given that the models are intended for real-time applications, diverse training data is critical to enable the model to adapt to various real-world scenarios. The diversity was achieved by including different individuals as drivers and varying the accessories worn, such as hats and spectacles.

https://doi.org/10.31436/iiumej.v26i2.3488



(a) Normal behavior



(b) Fatigue behavior

Figure 1. Example of images used in the dataset, showing the behaviors (a) normal behavior and (b) fatigue behavior.

Before training and validation, the data underwent pre-processing steps, including face cropping, normalization, and augmentation techniques to ensure a more robust model could be obtained. The image augmentations employed here are image scaling, random zoom, width and height shift, horizontal flip, and random brightness. CNN models are trained using images of 100x100 dimensions, whereas Vision Transformer models are trained using 224×224 images as required by the models. A manual annotation method accurately labeled the processed images into normal, yawning, and drowsiness classes. This approach ensured that the model was provided with high-quality training data for recognizing fatigue-related driving behaviors. In total, 43,034 images were used for training and validation, whereas 3,599 images were used for testing, with distinct individuals (non-overlapping) for each dataset to ensure unbiased evaluation. During training, all layers in all pre-trained models are unfrozen and optimized.

4.2. Experimental Setup

This work uses three separate computing platforms to evaluate the models under varying hardware configurations. The primary platform for training is a high-performance workstation equipped with an Intel® Core™ i7-14700KF 3.40 GHz processor, 32 GB of memory, and an NVIDIA RTX 4070 12GB GPU. This setup ensures efficient and accelerated training of deep learning models, capable of handling large datasets and complex computations essential for developing CNN and ViT models. For testing, a Raspberry Pi 5 features an ARM Cortex-A76 (ARM v8) 2.4 GHz processor, 8 GB of memory, and a VideoCore VII (800 MHz) GPU. This platform represents low-cost, embedded computing environments, allowing us to assess the models' performance on resource-constrained devices. Evaluating the Raspberry Pi 5 is critical for real-world applications in automotive systems, where such embedded systems are prevalent. Additionally, a desktop setup is used with an Intel® Core™ i7-13700 2.10 GHz processor, 16 GB of memory, and an NVIDIA RTX 4060 8GB GPU for further testing. This desktop configuration provides a middle ground in computing power between the workstation and the Raspberry Pi. It offers a more accessible and typical consumer-grade environment to assess the models' performance, which is relevant for broader applications. The specifications for each computing platform are given in Table 1.

https://doi.org/10.31436/iiumej.v26i2.3488

Table 1. Computing platforms used for training and testing

Specs.	**Computing Platforms**		
	Workstation (for model training)	***Raspberry Pi 5 (for model testing)***	***Desktop with GPU (for model testing)***
Processor	Intel® Core™ i7-14700KF 3.40 GHz	ARM Cortex-A76 (ARM v8) 2.4GHz	Intel® Core™ i7-13700 2.10 GHz
Memory	32 GB	8 GB	16GB
GPU	NVIDIA RTX 4070 12GB	VideoCore VII (800 MHz)	NVIDIA RTX 4060 8GB

During the training session, the time required to train the models is collected to provide valuable insights into their efficiency, practicality, and suitability for different applications. After that, the accuracy, precision, recall, and F1-score are evaluated along with the average inference time for all models. The equations for the accuracy, precision, recall, and F1-score are given in Eq. (1), Eq. (2), Eq. (3), and Eq. (4), respectively. Subsequently, the models are deployed on the Raspberry Pi 5 and a desktop computer with a GPU to evaluate their performance in classifying distracted driving behaviors. The models are run as part of the intelligent behavior detection program shown in Figure 2, which consists of face detection and eye detection, whereby the performance is evaluated in terms of frames per second (fps). The pseudocode of the intelligent behavior detection program, encompassing the functions and procedures required to assess the CNN and ViT models, is provided in Algorithm 1.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{4}$$

where True positive (TP) is a correct classification of the fatigue behavior, True negative (TN) is a correct classification of the normal behavior, False positive (FP) is an incorrect classification of the normal behavior as fatigue behavior, and False negative (FN) is an incorrect classification of the fatigue behavior as normal behavior.

Figure 2. Visualization of the outputs from the intelligent behavior detection program used to evaluate the model performance.

https://doi.org/10.31436/iiumej.v26i2.3488

Algorithm 1. The pseudocode for the intelligent behavior detection program

```
Start program

1: Import necessary AI, computer vision, and sensor interfacing libraries.
2: Define auxiliary functions:
        - ‘eye_close_detector’: Detects if the driver’s eyes are closed based on eye aspect ratio (EAR).
        - ‘post_process_behavior’: Processes behavior status (e.g., distracted, drowsy) based on thresholds.
        - ‘behavior_detector’: Identifies driver behavior using a pre-trained model.
        - ‘detect_face’: Detects face and extracts facial region for further analysis.
        - ‘get_frames’: Extracts RGB and grayscale frames for processing.
        - ‘load_model_and_labels’: Loads AI model and label data for driver behavior classification.
3: Initialize video capture and other counters.
        -  Load AI model using ‘load_model_and_labels’
4: Start main loop to process video frames:
        - Capture a video frame and apply overlay text.
        - Get full, cropped, and grayscale frames using ‘get_frames’.
5: Every ‘face_detector_interval’ frames:
        - Detect face using `detect_face`.
        - Detect eye status using ‘eye_close_detector’.
6: Every ‘behavior_detector_interval’ seconds:
        - If a face is detected, classify driver behavior using ‘behavior_detector’.
        - Process behavior status with ‘post_process_behavior’.
7: Display information on the video frame, including fps, behavior, eye status, and behavior history.

End program
```

The overall experiment workflow adopted in this work to benchmark the CNNs and ViTs in classifying driver fatigue behaviors is given in Figure 3. The pre-trained models that are compared in the experiments are Vision Transformer models: ViT B16, ViT B32, ViT L16, ViT L32, Swin B, Swin S, Swin T, Swin V2B, Swin V2S, Swin V2T, and CNN models: VGG 16, VGG 19, ResNet-50, MobileNetV2, InceptionV3, MobileNetV3, DenseNet-121, EfficientNetB0, EfficientNetV2 B0, and ResNet-152. According to Figure 3, CNN and ViT pre-trained models are trained using only the training data from the dataset. After model training is completed, each model is tested using the test data, and the performance is evaluated using accuracy, precision, recall, F1-score, training time, and inference time metrics. Subsequently, each model is deployed into the intelligent behavior detection program on a desktop with a GPU and a Raspberry Pi 5. Here, the performance evaluated is the frames per second (fps).

5. RESULTS AND DISCUSSIONS

This section presents the results of the experiment, in which the performance of the CNN and Vision Transformer models is analyzed in terms of accuracy, precision, recall, and F1-score, as well as training time, inference time, and computation requirements on the desktop with GPU and Raspberry Pi 5, measured in frames per second.

5.1. Overall Accuracy and Performance

The results tabulated in Table 2 show the test performance of CNN and Vision Transformer models evaluated in this work. Vision Transformer models are trained using different numbers of embeddings according to the architecture requirements defined in Table

https://doi.org/10.31436/iiumej.v26i2.3488

2. The batch size and epochs are also set to match the models' memory requirements and provide the best test accuracy.

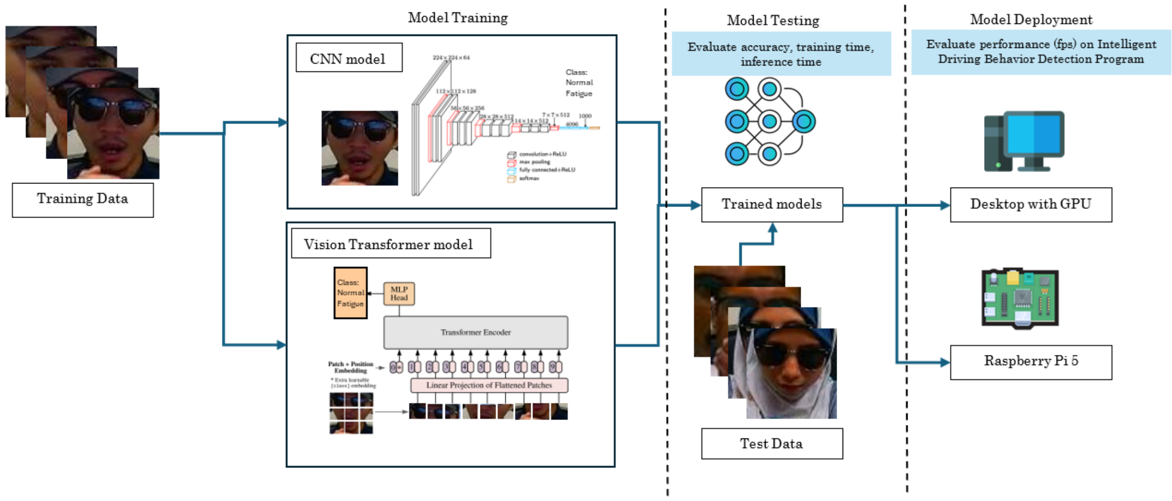


Figure 3. The overall experiment workflow adopted in this work is to benchmark the performance of CNN and ViT models in driver fatigue classification

Table 2. Overall performance comparisons

Models	Test Performance				# params	# emb.	batch size	epo chs	RPi 5 (fps)	Desktop (fps)
	Acc.	Prec.	Recall	F1						
ViT B16	0.9686	0.9459	0.9674	0.9561	85,801,730	768	32	10	7.73	64.86
ViT B32	0.8413	0.7843	0.8700	0.8062	87,458,306	768	16	10	18.67	73.22
ViT L16	0.8032	0.7518	0.8398	0.7669	303,305,730	1024	16	10	0.73	51.27
ViT L32	0.9177	0.8664	0.9434	0.8942	306,537,450	1024	16	10	0.89	66.6
Swin B	0.9705	0.9451	0.9757	0.9592	86,745,274	1024	16	10	1.32	58.35
Swin S	0.9716	0.9490	0.9734	0.9604	48,838,796	768	16	10	9.64	58.17
Swin T	0.9472	0.9073	0.9584	0.9290	28,291,428	768	32	10	18.1	69.06
Swin V2B	0.9619	0.9305	0.9693	0.9478	86,907,898	1024	16	10	1.27	58.82
Swin V2S	0.9897	0.9915	0.9789	0.9850	48,969,980	768	16	10	4.05	61.18
Swin V2T	0.9505	0.9123	0.9610	0.9332	27,584,108	768	32	10	14.29	64.38
VGG16	0.9808	0.9793	0.9653	0.9721	14,715,714	-	256	20	37.81	77.52
VGG19	0.9763	0.9816	0.9506	0.9650	20,025,410	-	256	20	36.75	73.71
ResNet-50	0.9566	0.9321	0.9462	0.9389	23,591,810	-	256	20	36.5	65.03
MobileNet V2	0.9519	0.9553	0.9051	0.9272	2,260,546	-	256	20	40.19	80.13
Inception V3	0.9691	0.9696	0.9411	0.9544	21,806,882	-	256	20	40.68	69.76
MobileNet V3	0.9605	0.9717	0.9155	0.9400	2,998,274	-	256	20	40.11	79.69
DenseNet-121	0.9713	0.9779	0.9399	0.9573	7,039,554	-	256	20	38.81	67.72
EfficientNet B0	0.9686	0.9669	0.9421	0.9538	4,052,133	-	256	20	42.91	69.61
EfficientNet V2B0	0.9605	0.9722	0.9150	0.9399	5,921,874	-	256	20	42.25	68.59
ResNet-152	0.9416	0.9046	0.9382	0.9198	58,375,042	-	32	20	27.82	61.31

According to Table 2, among the CNN models, MobileNetV2 and MobileNetV3 demonstrate strong performance across all metrics, with particularly high accuracy and F1-scores. MobileNet V3 slightly outperforms MobileNet V2 in precision and recall. EfficientNet B0 and EfficientNet V2B0 also perform well, maintaining high accuracy and F1-scores. DenseNet121 exhibits robust performance similar to EfficientNet models, with balanced metrics. VGG16 and VGG19 show excellent performance, especially in precision and recall, resulting in high F1-scores. VGG16 delivers the best F1-score among all tested CNN models at 0.9721. ResNet-50 exhibits strong performance with balanced metrics, slightly lower than

https://doi.org/10.31436/iiumej.v26i2.3488

InceptionV3 and VGG models, but still robust. For the ViT models, Swin T, Swin S, Swin VSS, and Swin V2B perform well, particularly Swin V2S, which shows high precision and recall, translating to a high F1-score. ViT B16 shows competitive performance with high accuracy and F1-score, but slightly lower precision and recall than Swin models. ViT B32, ViT L16, and ViT L32 show varying performance, with ViT L32 lagging in recall and F1-score but maintaining decent accuracy and precision. Swin V2S gives the best F1-score among Vision Transformer models at 0.9850, and surpasses VGG16, which is the best-performing CNN model.

Based on Table 2, comparing the two types of models, CNN models like VGG16, VGG19, and Inception V3 exhibit high accuracy. ViT models such as Swin S, Swin V2S, and ViT B16 also show high accuracy but with some variability among variants. Regarding precision, ViT models, particularly Swin V2S, show higher precision than many CNN models. CNN models generally exhibit strong precision, with VGG and Inception models leading. Regarding recall, ViT models such as Swin V2S, Swin S, and ViT B16 exhibit higher recall, indicating better performance in identifying all relevant instances. CNN models like VGG19 and InceptionV3 also demonstrate high recall.

5.2. Model Training and Model Inference Performance

As shown in Table 2, ViT models also tend to have a much higher number of parameters. ViT L32 has over 303 million parameters, significantly higher than the 2.9 million parameters in MobileNetV2. This higher parameter count can contribute to the increased training time and potentially higher model complexity, impacting the inference time. Thus, every model's training and inference time is measured and shown in Figure 4 and Figure 5, respectively.

According to Figure 4, ViT models generally require significantly longer training times than CNN models. For instance, ViT L16 has the longest training time at 17087.90 seconds, whereas MobileNetV2 has the shortest at 1682.62 seconds. Even CNN ResNet-152's longest model training time, at 2698.13 seconds, is almost two times shorter than the fastest model training of Vision Transformer ViT B32, at 4688.91 seconds. This indicates that Vision Transformers are more computationally intensive and require more computing resources for training.

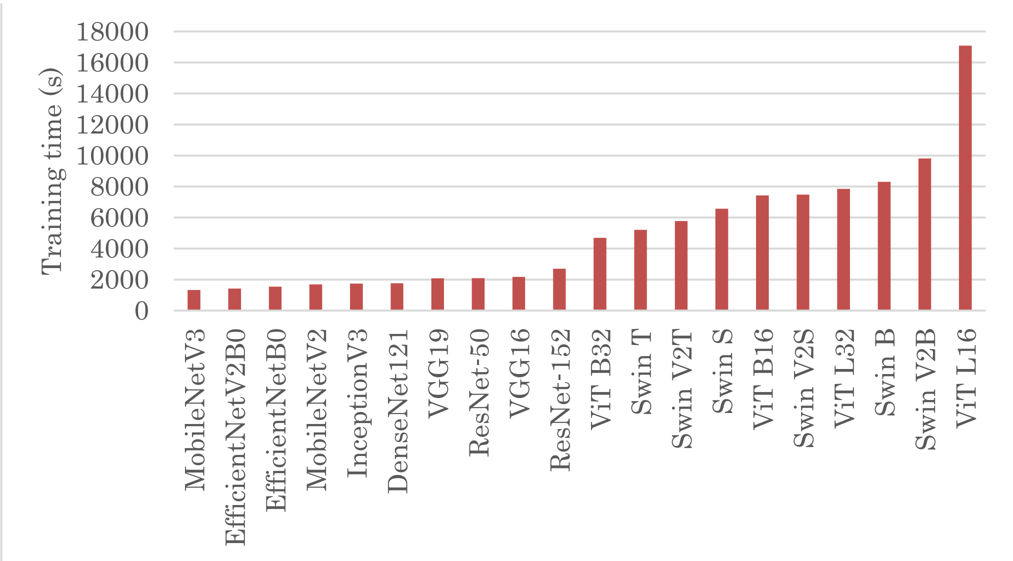


Figure 4. The training time required for each model, run on the workstation

https://doi.org/10.31436/iiumej.v26i2.3488

Furthermore, the inference times for these models are measured. The time taken to process the image and apply the models is considered in the measurement of inference time. Surprisingly, based on Figure 5, Vision Transformers' inference times are shorter than CNNs. For example, ViT L32 has an average inference time of 0.0055 seconds, whereas MobileNet V2 has an inference time of 0.0731 seconds, even though the ViT L32 model has higher complexity. To ensure that the inference performance on the powerful workstation used for training is translated into the actual application implementation, each model is tested in an intelligent driving behavior detection program, as mentioned earlier. This comparison is crucial for understanding how these models perform in different hardware environments, particularly for real-time applications like driving behavior monitoring systems.

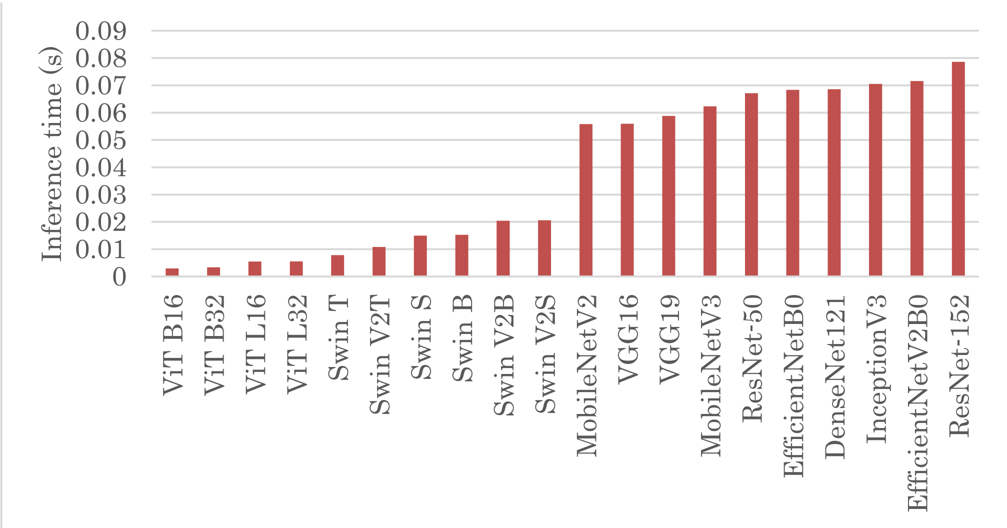
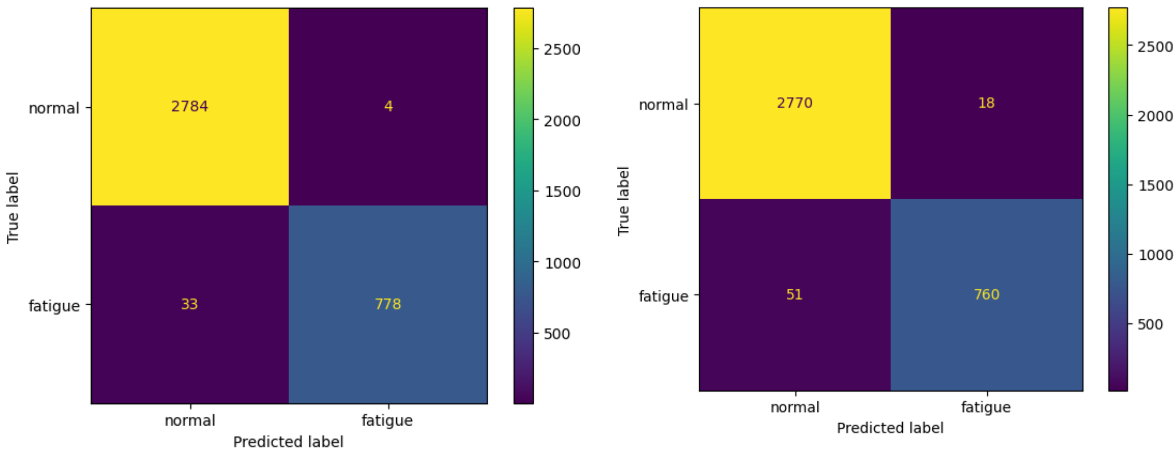


Figure 5. The inference time required for each model, run on the workstation

5.3. Swin V2S vs. VGG-16 Classification Results Analysis

Based on the results tabulated in Table 2, the top two best CNN and ViT models according to the F1 Score are the VGG-16 and Swin V2S models, with 0.9721 and 0.9850 F1 Scores, respectively. In this section, the results of the two models are compared through an analysis of the confusion matrix, as presented in Figure 6.



(a) Swin V2S (b) VGG-16

Figure 6. The confusion matrix from the test dataset classification using the Top-2 classifiers: (a) Swin V2S and (b) VGG-16

https://doi.org/10.31436/iiumej.v26i2.3488

From Figure 6, Swin V2S demonstrates high accuracy, particularly for the "normal" class, with a low number of false positives (4) compared to VGG-16 (18). Its ability to correctly identify both classes, especially "fatigue," is evident, with only a small number of fatigue cases misclassified as normal cases (33) compared to (51) by the VGG-16 model. The VGG-16 model shows a slightly higher rate of errors in both false positives and false negatives compared to Swin V2S. Specifically, it has slightly more difficulty differentiating between "normal" and "fatigue" states, leading to greater misclassification in both cases. It clearly shows that the Swin V2S model consistently outperforms VGG-16 in distinguishing between normal and fatigue driving conditions, showing fewer errors across both classes. Figure 7 shows several examples of misclassifications made by the models.

Swin V2S performs superiorly over VGG16 in driver fatigue classification due to its hierarchical architecture with shifted windows, which enables efficient capture of both local details and global context. This ability to integrate fine-grained features with broader patterns is enhanced by its self-attention mechanism within non-overlapping windows, making distinguishing between fatigue and normal states robust. While VGG16 offers a simpler sequential architecture with small filters, it cannot handle long-range dependencies of the relationship between spatially distant regions of an image, a strength of Swin V2S.

According to Figure 7, most misclassifications are due to the apparent similarity of some fatigue behavior images with normal behavior images. This similarity challenges the model by having shared visual characteristics and low inter-class variability. Additionally, since the data collection requires the participants to imitate fatigue behavior, the imitations may not be as realistic as the actual behaviors, i.e., when the drivers are in a fatigue or drowsy condition while driving. This has contributed to several misclassifications by the models.

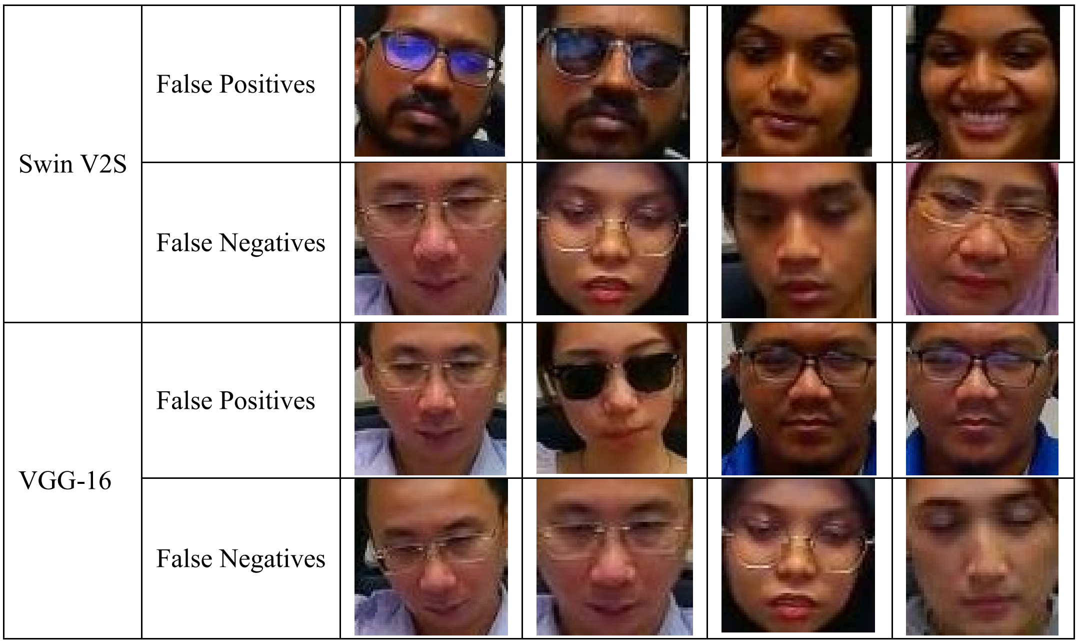


Figure 7. Examples of several misclassifications by the Swin V2S and VGG-16 models. False Positives refer to the incorrect classification of normal behavior as fatigue behavior, whereas False Negatives refer to the incorrect classification of fatigue behavior as normal behavior

5.4. Model Performance in an Intelligent Driving Behavior Detection Program

Each of the CNN and ViT models trained earlier is loaded and deployed in the intelligent driving behavior detection program to evaluate the inference speed of each model on different computing platforms. The platforms used in the experiment are a Raspberry Pi 5 and a desktop computer with a GPU. The inference performance is shown in Figure 8.

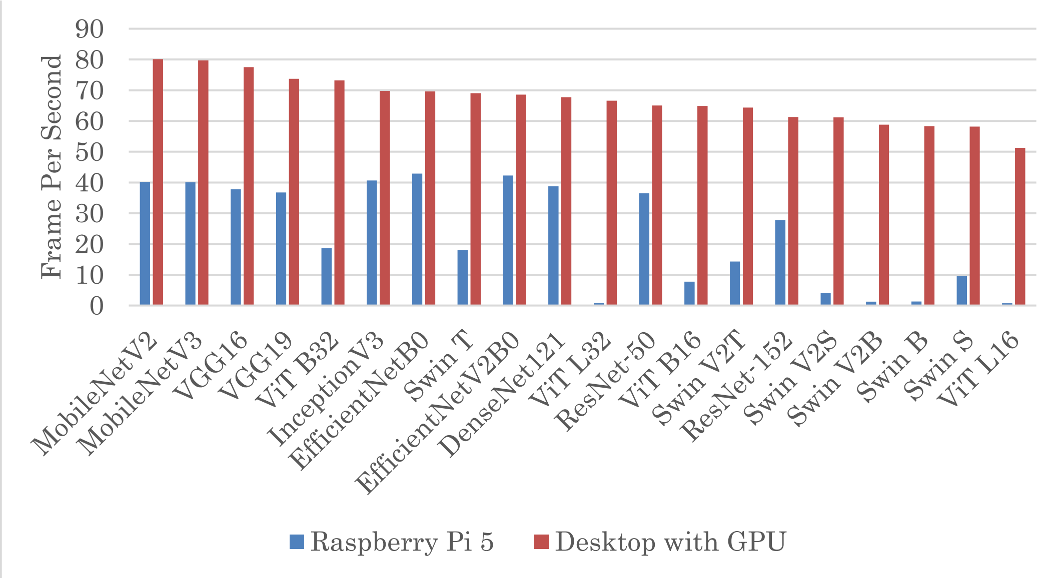


Figure 8. The comparison of processing time measured in fps for each model implemented in the intelligent driving behavior detection program.

According to Figure 8, the inference performance on the desktop with a GPU shows a clear advantage across all models, achieving significantly higher fps than the Raspberry Pi 5. MobileNet models demonstrate exceptional performance on the desktop, reaching close to 80 fps, indicating their efficiency and suitability for high-speed inference tasks. Other CNN models, such as VGG16, VGG19, and InceptionV3, also perform faster on the desktop, maintaining fps values around 70 and above. ResNet models achieve moderate fps values, around 60, indicating good performance but slightly less efficiency than the more streamlined models.

The inference performance on the Raspberry Pi 5 is notably lower, reflecting the constraints of running complex models on a less powerful, embedded platform. EfficientNetB0 and V2B0 still perform well, achieving more than 30 fps, which is respectable for such a device. Other CNN models perform relatively well, maintaining fps values of more than 25 fps. However, Vision Transformer models like ViT B16, ViT B32, and larger ViT variants such as ViT L16 and L32 exhibit very low fps on the Raspberry Pi 5, often below 10 fps, highlighting the significant computational demands of these models that are not well-suited for low-power embedded devices.

On the other hand, Vision Transformer models achieved almost equivalent fps performance to CNN models on a desktop with a GPU. With a GPU for computation, the fastest Vision Transformer model (ViT B32) can achieve higher fps than CNN InceptionV3. Even the slowest Vision Transformer model, ViT L16, achieved more than 50 fps, indicating that Vision Transformer models, while offering higher accuracy and better handling of complex patterns, demonstrate excellent fps in environments with high computational capabilities, such as desktops with powerful GPUs. To further illustrate the performance comparisons of CNN and Vision Transformer models in actual applications, the comparisons are given in Figs. 9 and 10.

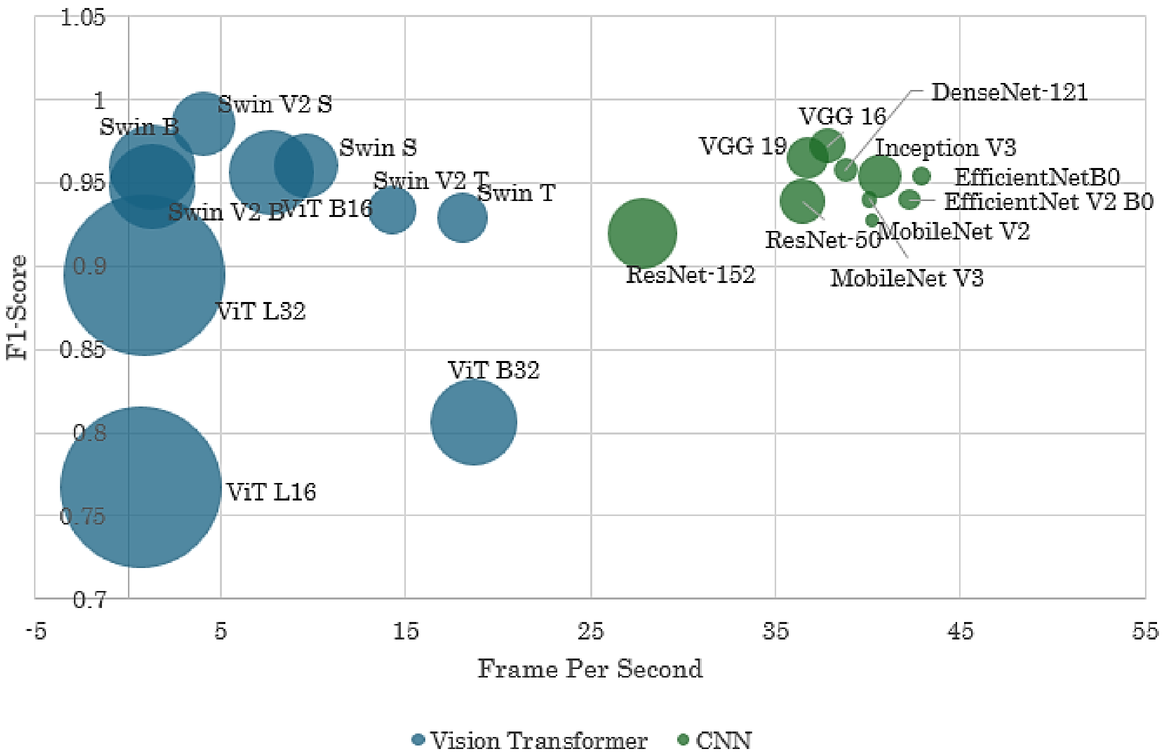


Figure 9. Performance comparison in terms of F1-score, fps, and number of parameters (defined by the bubble size) for each model run in the intelligent digital video recorder program running on Raspberry Pi 5.

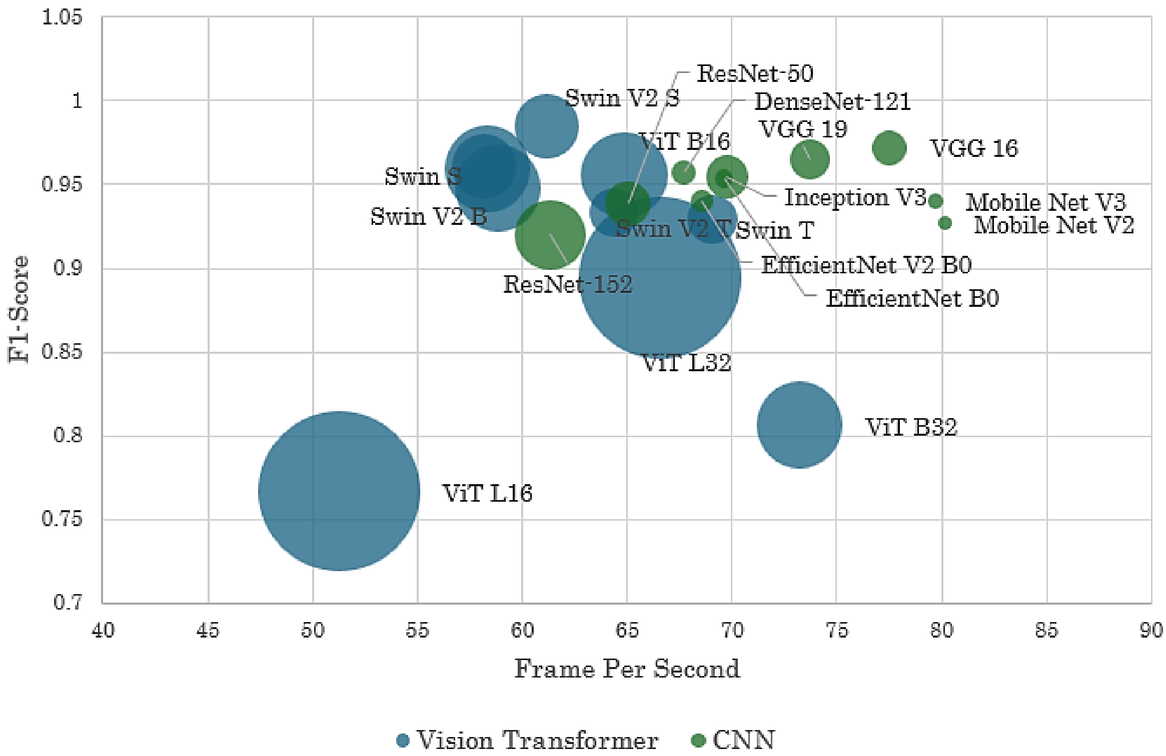


Figure 10. Performance comparison in terms of F1-score, fps, and number of parameters (defined by the bubble size) for each model run in the intelligent digital video recorder program running on the desktop with a GPU.

Based on Figure 9, all CNN models can deliver more than 25 fps performance on the Raspberry Pi 5. However, only ViT B32 and Swin V2T can provide acceptable fps on Raspberry Pi 5 at around 14-18 fps. The Vision Transformer model with the best F1-Score, Swin V2S, can only deliver 4 fps on the Raspberry Pi 5. Meanwhile, according to Figure 10, all models benefit from the enhanced computational power on the desktop with a GPU, but CNNs still slightly outperform Vision Transformer in inference speed.

VGG16 and VGG19 achieve more than 70 fps with high F1-scores of 0.97 and 0.95, respectively, indicating their efficiency and suitability for high-speed, real-time applications. MobileNetV2 and MobileNetV3 perform exceptionally well, maintaining fps around 80 with good F1-scores. In contrast, Vision Transformer models like ViT B16, ViT B32, and Swin variants achieve lower fps on GPU but still more than 45 fps, though Swin V2S stands out among Vision Transformers with relatively better performance, with 61 fps and 0.99 F1-score.

To summarize the findings, VGG16 is recommended as the best CNN model due to its superior balance of speed and accuracy across different hardware environments, making it ideal for both embedded and high-performance applications. Swin V2S is recommended as the best Vision Transformer model for its high accuracy and reasonable inference speed, especially in high-performance environments, despite its lower fps than CNN models. For deployment on computation machines with a GPU, Swin V2S will deliver the best performance and excellent computation speed, while VGG16 should be used if deployment is targeted at embedded platforms.

6. CONCLUSION

This paper investigates and compares CNN models against Vision Transformer (ViT) models in classifying fatigue driving behaviors. Both model types were evaluated on their accuracy and performance during training and inference on desktop and embedded platforms. CNN models consistently deliver high accuracy with balanced precision and recall, with VGGs, InceptionV3, and MobileNets showing exceptional performance across all metrics. ViT models, particularly the Swin variants, excel in precision and recall but require longer training and computational resources. CNN models like VGGs, InceptionV3, and MobileNets are reliable for balanced performance with high accuracy and efficient processing. In contrast, ViT models, especially Swin variants, are suitable for applications demanding higher precision and recall, provided robust computational resources are available. VGG16, with a 0.97 F1-score and 77 fps on GPU and 37.81 fps on Raspberry Pi 5, is the top CNN model for its balance of speed and accuracy. Swin V2S, achieving 0.99 F1-score and 61.18 fps on GPU, is the best ViT model, recommended for high-performance settings despite lower fps than CNNs. ViT models, though slower on embedded platforms, can surpass CNNs in accuracy in environments with powerful GPUs. Future work should explore hybrid models combining CNN and ViT strengths to enhance driving behavior classification further. Real-time testing on larger, more diverse datasets and advanced data augmentation and temporal context integration through RNNs or LSTMs could improve model robustness and accuracy. Optimizing models for deployment on low-power embedded devices while maintaining high accuracy is crucial for practical, real-world automotive applications.

ACKNOWLEDGEMENT

The authors extend their gratitude to the Ministry of Higher Education for their support through the Fundamental Research Grant Scheme FRGS/1/2023/TK07/UITM/02/23 and to the College of Engineering at Universiti Teknologi MARA for their assistance in this project.

REFERENCES

[1] Ministry of Transport Malaysia, “Malaysia Road Fatalities Index,” Ministry of Transport Malaysia Official Portal. Accessed: Jun. 13, 2024. [Online]. Available: https://www.mot.gov.my/en/land/safety/malaysia-road-fatalities-index

[2] A. Fernández, “Facial attributes recognition using computer vision to detect drowsiness and distraction in drivers,” Electronic Letters on Computer Vision and Image Analysis, vol. 16, no. 2, pp. 25–28, 2017, doi: 10.5565/rev/elcvia.1134.

[3] B. Shandhana Rashmi and S. Marisamynathan, “Factors affecting truck driver behavior on a road safety context: A critical systematic review of the evidence,” Oct. 01, 2023, KeAi Communications Co. doi: 10.1016/j.jtte.2023.04.006.

[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[5] L. Cun et al., “Handwritten Digit Recognition with a Back-Propagation Network.”

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale Video Classification with Convolutional Neural Networks.” [Online]. Available: http://cs.stanford.edu/people/karpathy/deepvideo

[7] A. Vaswani et al., “Attention Is All You Need,” Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[8] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.11929

[9] Y. S. Poon et al., “Driver Distracted Behavior Detection Technology with YOLO-Based Deep Learning Networks,” in ISPCE-ASIA 2021 - IEEE International Symposium on Product Compliance Engineering-Asia, Proceeding, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ISPCE-ASIA53453.2021.9652435.

[10] M. F. Ishak, F. H. K. Zaman, N. K. Mun, S. A. C. Abdullah, and A. K. Makhtar, “Improving night driving behavior recognition with ResNet50,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 33, no. 3, pp. 1974–1988, 2024, doi: 10.11591/ijeecs.v33.i3.pp1974-1988.

[11] J. Liang, H. Zhu, E. Zhang, and J. Zhang, “Stargazer: A Transformer-based Driver Action Detection System for Intelligent Transportation,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE Computer Society, 2022, pp. 3159–3166. doi: 10.1109/CVPRW56347.2022.00356.

[12] T. Kondo, S. Narumi, Z. He, D. Shin, and Y. Kang, “A Performance Comparison of Japanese Sign Language Recognition with ViT and CNN Using Angular Features,” Applied Sciences (Switzerland), vol. 14, no. 8, Apr. 2024, doi: 10.3390/app14083228.

[13] H. E. Kim, M. E. Maros, T. Miethke, M. Kittel, F. Siegel, and T. Ganslandt, “Lightweight Visual Transformers Outperform Convolutional Neural Networks for Gram-Stained Image Classification: An Empirical Study,” Biomedicines, vol. 11, no. 5, May 2023, doi: 10.3390/biomedicines11051333.

[14] H. V. Koay, J. H. Chuah, and C. O. Chow, “Convolutional Neural Network or Vision Transformer? Benchmarking Various Machine Learning Models for Distracted Driver Detection,” in IEEE Region 10 Annual International Conference, Proceedings/TENCON, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 417–422. doi: 10.1109/TENCON54134.2021.9707341.

[15] S. Z. Ul Abidin, H. M. Lashari, and R. F. Ahmad, “ViT vs CNN: A Comparative Study of Wheat Disease Classification for Custom Data,” in Proceedings - 2023 International Conference on Frontiers of Information Technology, FIT 2023, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 274–279. doi: 10.1109/FIT60620.2023.00057.

[16] M. M. Sufian, E. G. Moung, J. A. Dargham, F. Yahya, and S. Omatu, “Pre-trained Deep Learning Models for COVID19 Classification: CNNs vs. Vision Transformer,” in 4th IEEE International

Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IICAIET55139.2022.9936852.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2015arXiv151203385H

[18] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

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

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2014arXiv1409.1556S

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.06993

[22] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2017arXiv170404861H

[23] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," arXiv e-prints, p. arXiv:2103.14030, Mar. 2021, doi: 10.48550/arXiv.2103.14030.

[24] D. Li, Y. Wang, and W. Xu, "A Deep Multichannel Network Model for Driving Behavior Risk Classification," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 1, pp. 1204–1219, Jan. 2023, doi: 10.1109/TITS.2022.3201378.

[25] M. F. Bin Ishak, F. H. K. Zaman, N. K. Mun, and A. K. Makhtar, "Day Driving and Night Driving Behavior Detection Using Deep Learning Models," in International Conference on ICT Convergence, IEEE Computer Society, 2023, pp. 463–468. doi: 10.1109/ICoICT58202.2023.10262745.

[26] A. Noor, B. Benjdira, A. Ammar, and A. Koubaa, "DriftNet: Aggressive Driving Behaviour Detection using 3D Convolutional Neural Networks," in Proceedings - 2020 1st International Conference of Smart Systems and Emerging Technologies, SMART-TECH 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 214–219. doi: 10.1109/SMART-TECH49988.2020.00056.

[27] Z. Qu, L. Cui, and X. Yang, "HAR-Net: An Hourglass Attention ResNet Network for Dangerous Driving Behavior Detection," Electronics (Switzerland), vol. 13, no. 6, Mar. 2024, doi: 10.3390/electronics13061019.

[28] P. Li et al., "Driver Distraction Detection Using Octave-Like Convolutional Neural Network," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 8823–8833, Jul. 2022, doi: 10.1109/TITS.2021.3086411.

[29] H. Vin Koay, J. Huang Chuah, and C. O. Chow, "Shifted-Window Hierarchical Vision Transformer for Distracted Driver Detection," in TENSYMP 2021 - 2021 IEEE Region 10 Symposium, Institute of Electrical and Electronics Engineers Inc., Aug. 2021. doi: 10.1109/TENSYMP52854.2021.9550995.

[30] H. Wang et al., "FPT: Fine-Grained Detection of Driver Distraction Based on the Feature Pyramid Vision Transformer," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 2, pp. 1594–1608, Feb. 2023, doi: 10.1109/TITS.2022.3219676.