Journal of Information and Organizational Sciences
Volume 49, Number 1 (2025)
Journal homepage: jios.foi.hr

JIOS

DOI: 10.31341/jios.49.1.9
Open Access

UDC 004.8:656.017.2-049.5
Original Scientific Paper

Comparative Study of VGG16, ResNet50, and YOLOv8 Models in Detecting Driver Distraction in Varying Lighting Conditions

Ali Nafaa Jaafar[1*] and Mustafa Nafea Alzubaidi[2]

[1]Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
[2]Computer Techniques Engineering Department, Al-Esraa University College, Baghdad, Iraq
[*]Correspondence: ali_nafaa@mtu.edu.iq

PAPER INFO

Paper history:
Received 03 March 2025
Accepted 09 April 2025

Citation:
Jaafar, A. N. & Alzubaidi M. N. (2025). Comparative Study of VGG16, ResNet50, and YOLOv8 Models in Detecting Driver Distraction in Varying Lighting Conditions. In Journal of Information and Organizational Sciences, vol. 49, no. 1, pp. 139-159

ABSTRACT

Observing driver distractions while driving gives valuable information to prevent accidents, so it is necessary to use effective monitoring methods. Deep learning is showing new capabilities in solving this issue. This study evaluates the results of CNN, YOLOv8, ResNet50 and VGG16 deep learning models as they detect drivers who are practising distracted driving behaviours under real-time and various lighting conditions (day and night). The models were trained on two datasets: the labelled State Farm dataset and the Driver Monitor Dataset (DMD). They successfully identified ten distinct categories of distraction for the State Farm dataset and five categories for the monitoring drivers dataset. Pre-trained models were optimized using transfer learning through fine-tuning to enhance detection accuracy. This paper studies related work on distracted driving and shares ideas for designing advanced systems that use various methods to improve accuracy. YOLOv8 reached an outstanding test accuracy of 98.46% on the State Farm dataset, proving itself superior to other methods and confirming its effectiveness for monitoring. In addition, YOLOv8 reached 96.46% accuracy in the DMD dataset, outperforming VGG16 at 90.58% and ResNet50 at 70.80%. YOLOv8 was able to recognise important driver behaviours in real time with a dataset of 15 subjects and 20 different driving postures. The research proves that the YOLOv8 model is fit for use in intelligent monitoring systems designed to detect distracted driving and promote safer driving through focused actions.

Keywords: VGG16, ResNet50, YOLOv8, distracted driver detection, Transfer learning

1. Introduction

Driver distraction refers to a temporary shift of attention from driving to an unrelated task, object, or event, which reduces awareness and increases the risk of accidents. Distractions can be categorised into four types:

- Visual: Looking away from the road, like checking a GPS.
- Auditory: Hearing unrelated sounds, like a phone ringing.
- Biomechanical: Engaging in physical actions like eating.
- Cognitive: Having mental focus elsewhere, like daydreaming.

The various forms of driver distraction tend to combine when people both text and talk on the phone and radio adjustment at the same time, which demonstrates why reliable monitoring systems should be implemented. Research accuracy and effective prevention strategy development require a standardised definition of driver distraction [1]. Road accidents occur mainly because of driver distraction which endangers both vehicle operators and other people using the roads [2]. Sensor-based along with traditional methods

encounter significant challenges when dealing with the diverse driving situations that occur in actual settings [3]. Machine learning utilises algorithms and statistical models to detect sophisticated patterns in data that elude rules-based systems. These systems learn from data and adjust accordingly to new patterns autonomously without the need for those rules to be defined manually. A major idea is supervised learning, which allows for the development of a link between input variables and output reactions, where one can predict responses for unexplored inputs [4].

New advancements in deep learning together with machine learning have improved our results by enhancing their accuracy and capabilities [5]. Modern computer vision technology working with deep learning enables automatic driver action detection and classification through in-car Infrared imaging cameras operate effectively under both daytime and nighttime conditions [6], [7]. Random neural networks enable driver distraction pattern analysis through camera and computer processing unit integrated systems which monitor driver body positions to collect driving data. Model training mechanisms utilize the gathered data to detect driver behaviors including sleeping and eating and conversing [8], [9].

Transfer learning serves as a method for weight adjustment and model unification to boost classification outcomes. The reliability of CNN-based approaches has improved through recent developments [3], [10]. The research analyses VGG16 ResNet50 and YOLOv8 to establish driver distraction detection methods and design alert systems for accident prevention [8]. VGG16 neural network stores and hosts 16 layers that consist of thirteen convolutional layers with three fully connected layers. A complete training process took place for this network on the wide-ranging State Farm image dataset and the dataset for monitoring drivers. The convolutional layers of this model use 3x3 filters as an established image classification method [11]. The ResNet network allows for better image classification because the residual connections will solve the gradient vanishing problem easily in deep networks training. The system demonstrates exceptional feature extraction ability to accurately identify photos related to driver distractions [12]. Real-time object detection is a strong point of YOLO models which is why YOLOv8 stands out for spotting different objects in images making it an optimal solution for distracted driving hazard monitoring [13].

The field of computer vision uses object detection to identify different objects that exist within an image [14]. Object detection models exist as two primary types: single-stage and two-stage models. The YOLO is a single-stage models that make predictions about bounding boxes and class labels simultaneously through one network computation run without generating region proposals [15]. The two-stage detectors create candidate regions through their initial process before using VGG and ResNet pre-trained convolutional neural networks to extract and classify features [16]. The research demonstrates how object detection methods help enhance distracted driver detection systems [17].

This study introduces an approach to driver distraction detection and offers the following key contributions:

- In our research, we expanded the dataset, which helps the model generalise effectively. We also trained the model on a wide variety of categories related to driver distraction. The method delivers a dual classification system that can be used to predict distracted driving in various vehicles, camera perspectives, and lighting including day and night.
- We present a model that can detect distracted driving in real time. After training our system, we evaluated it under real-world conditions and achieved positive results. This validation demonstrates that our system can generalise effectively across diverse datasets, which is essential given the limited research on real-time distracted driving detection.
- This study compares a CNN-based (VGG16, ResNet50, and YOLOv8) technique for the detection and recognition of distracted drivers to prevent road accidents and compares the results. From the results, we can see the effectiveness of YOLOv8 in the identification of all kinds of distracted behaviours, opening up the path for on-time automated systems that can give warnings for road safety.
- We have effectively tackled the immediate detection of driver distraction at night. Regrettably, the literature on categorising the various distractions using nighttime images is minimal. This study utilises machine learning techniques on a comprehensive and diverse dataset to address this gap.

The organisation of this work is outlined as follows: Section two reviews prior research and provides the scientific background on driver distraction detection. Section three details the methods used in the study, while Section four introduces a dataset. Section five introduces a proposed approach, Section six sets out and discusses experimental results and Section seven concludes with key ideas and results.

2. Related Work

This section offers a discussion of driver distraction research, focusing on main results, methods, and advancements. Vaegae Naveen et al. [18] proposed using VGG16 and ResNet50 to identify distracted drivers, reporting accuracies of 86.1% using the VGG16 framework and 87.92% with ResNet50. They noted that similar poses in images can lead to different misclassifications. Additionally, our research has achieved improved results with VGG16 by using pre-trained weights for initialisation which has resulted in reduced training time. Dhiman A. et al. [19] executed research which contrasted CNN patterns with conventional machine-learning approaches. Their research determined how CNN surpassed logistic regression while beating ResNet50 and VGG16 within the chosen model selection process. The performance of deep learning analysis enables the exploration of potential traffic accidents to create safer traffic systems, while this research focuses on enhancing the real-time detection of distracted drivers using YOLOv8 technology.

Mustafa Aljasim and Rasha Kashef [20] presented an ensemble model that unites VGG16 and ResNet50 in their work. The model reached its maximum accuracy level of 92%. The model faces implementation difficulties on devices with limited computational power because it requires many parameters (138 million for VGG16 and 23 million for ResNet50) in addition to its complex nature. Constructing E2DR models without relying on predecessors is quite challenging. Recognising the importance of computing speed in real-time applications, YOLOv8 has achieved significant performance improvements through our research without significantly increasing computational complexity. Anirudh Muthuswamy et al. [3] presented an ensemble framework combining autoencoders with CNN models VGG19, DenseNet121 and ResNet50. The system adjusts to current conditions through transfer learning and data augmentation while the distraction level changes from one moment to the next. The framework reaches real-time driver distraction detection standards with the help of this system. An ensemble framework performs better than individual classifiers while an autoencoder combination brings equivalent performance improvements for distraction detection when used with the ensemble approach. The limitation of this study is its reliance on a framework that consists only of basic CNN models. There are no real-time object detection methods included, and the training images were all taken in daylight. In our proposed approach, we integrated an object detection model and trained it to enhance the dataset for detecting both daytime and nighttime driving.

A new framework appeared in Abdul Jamsheed V. et al. [21] The proposed framework consists of standard CNN along with deeper and augmented versions where transfer learning is applied to all parts. The authors evaluated their technique using the AUC dataset and obtained a remarkable accuracy rate of 97% through transfer learning. This work restricted the model training data to the State Farm. While there are 10 classes for driver actions, visually similar behaviours (texting vs. phone use) may still be challenging to differentiate. In contrast, our research used several models to accurately identify all categories of driver distraction. Mittal, H and Verma, B [22] Merged innovative methods through which convolutional neural networks received input from attention-based capsule networks (CapsNet). As a result, we obtained very good classification results, which show the merit and great potential of fusion networks. The AUC dataset partial gender balance (22 males, 9 females) may limit the model generalizability. The fixed input size of 128×128 pixels may constrain performance by omitting critical visual details during resizing. The image resize has been set to 256 x 256 for all models used in our work. Chenghao Guo et al. [23] built a temporal information fusion network based on CNN architectures which detects driver distractions. The researchers present beneficial information about implementing fusion neural networks in image processing applications. The self-attention mechanism in transformer methods results in significant success rates in detecting objects and classifying images. The limited resolution of the Brain4cars dataset limits semantic segmentation accuracy, unlike the datasets used in our work.

A modified VGG architecture combined with transfer learning produced results of 96.95% accuracy according to Khalid Alshalfan et al. [24]. The researchers moved the weights from ImageNet into their network structure. The dataset consists of over 33,000 images which cover various classes. The researchers faced challenges while conducting their experiment because they had to distinguish similar yet distinct actions such as conversing with right-hand use from texting with right-hand use. The system struggles to process real images because of a complex parameter list. A large number of parameters may overwhelm the computational resources and could cause system failure. Our research focused on building a comprehensive dataset based on actual images and utilising advanced techniques. Liu Shugang et al. [25] proposed a method for detecting driver distraction using infrared (IR) images. Generally, this method addresses the specific issues of IR imaging by employing preprocessing techniques including image inversion, CLAHE, and histogram equalisation. In response to these issues, the CEAM-YOLOv7 model enhances the YOLOv7 architecture with a Global Attention Mechanism (GAM), uses lightweight techniques, and improves the dataset through Channel Expansion (CE) algorithm. Experimental evaluations demonstrated an improved mean Average Precision (mAP) of 0.736, a

high processing speed (156 FPS), and reduced model complexity, making it suitable for in-vehicle deployment. However, the current model only focuses on four types of distraction behaviours, indicating a need for broader behavioural coverage; while our study analysed fifteen different types of driver distraction behaviours.

3. Theoretical Background

The following section examines CNN-based model architectures which detect driver distraction by using VGG16, ResNet50, and YOLOv8 transfer learning methods.

3.1. CNNs (Convolutional Neural Networks)

Modern image classification together with object detection and processing techniques have advanced considerably due to deep CNNs. This subset of deep learning consists of convolutional, pooling, and fully connected layers, as illustrated in Figure 1 [26].

- The convolutional Networks layer (ConvNets) extracts features from input images to generate the feature map; the convolution process maintains spatial relationships between pixels while learning detailed features from small regions of the image.
- The pooling layer decreases the spatial dimensions of large input images by downsampling while retaining important information. Max pooling is a widely used technique that identifies the maximum value from input features.
- Fully connected layer operates as the multi-layer neural network that predicts the probability of each value belonging to a specific class after feature extraction through convolutional layers and dimensionality reduction via pooling layers [27].

Models trained on larger datasets generally demonstrate better generalisation than those trained on smaller datasets [26]. Researchers have focused on developing convolutional neural networks to accurately detect distracted driver behaviours and ensure compatibility across different types of vehicles. However, this method leads to a complex model that is difficult to utilise in real-time applications such as VGG [28]. The emphasis has moved towards creating lightweight network architectures used in real time, such as Residual Networks, optimised explicitly for low-computation devices [29].

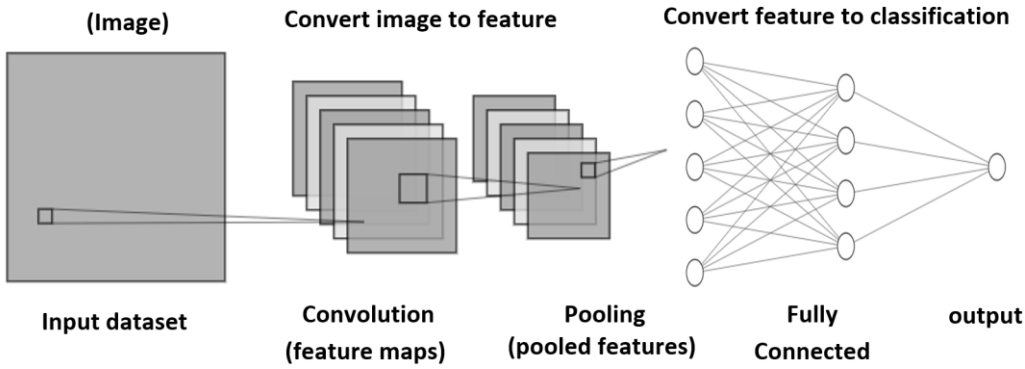

Figure 1. Structure of a CNN.

3.2. Transfer Learning and Model Fine-Tuning

Pre-trained models serve as a foundation for computer vision applications, effectively leveraging their ability to detect generic features in images [30]. Transfer learning entails training a base network on a primary dataset and adapting the acquired features for a different task and dataset within a target network. This process extracts accurate and concise feature sets from the training data [31]. Fine-tuning is a form of transfer learning where the parameters of a pre-trained model are adjusted for a particular task. For example, the final layer of the model trained on the source dataset can be modified and retrained with your data to classify custom categories. This approach helps tailor the model to your requirements and overcome the limitations of the pre-trained version [32]. Numerous pre-trained models, including ResNet50, VGG16, and YOLOv8, have been developed and widely shared. ResNet50 and VGG16 excel in convolutional neural networks for

image object detection, segmentation, and classification tasks, while YOLO is renowned for its speed and accuracy in object detection. All these models can be fine-tuned for specific applications. The details of each model are discussed below [33].

3.2.1. VGG16 model

Figure 2 depicts the VGG16 architecture, a 16-layer version of the VGG model that includes thirteen convolutional layers, Max Pooling layers, and trainable Softmax layers. The network features five convolutional blocks, each succeeded by a Max Pooling layer. Starting with 64 channels, the number of channels doubles after each Max Pooling layer, culminating in 512 [34].

The first two blocks consist of two convolutional layers each, while the following three blocks include three convolutional layer. The architecture concludes with three dense layers, with a feature map size halving after each Max Pooling layer. The ReLU (rectified linear unit) activation function is widely applied throughout the network to introduce non-linearity [35].

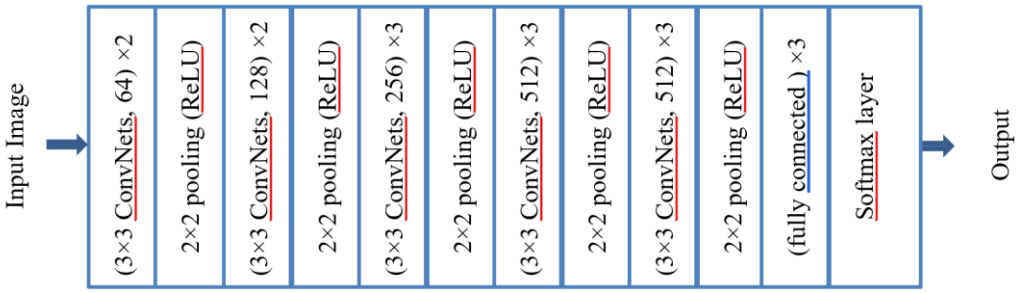

Figure 2. Architecture of the VGG16.

3.2.2. The ResNet50 Model

ResNet50 is part of the Residual Networks family and is a CNN architecture designed to tackle challenges in training deep neural networks, particularly the degradation problem. ResNet50 effectively mitigates this issue by employing residual blocks with skip connections. The architecture utilises bottleneck residual blocks, which consist of 3 convolutional layers: the 1x1 layer for dimensionality reduction, the 3x3 layer for spatial feature extraction, and another 1x1 layer to restore the channel dimensions. Batch normalisation and ReLU activation functions are applied after each layer to enhance learning efficiency. The skip connections help preserve essential information from earlier layers, enabling the training of deeper networks [36].

ResNet50 is composed of 50 stacked bottleneck blocks. The initial layers perform conventional convolution and pooling while the residual blocks extract refined features. The final fully connected layers classify the image, making ResNet50 a highly efficient and accurate model for image classification tasks. The diagram in Figure 3 presents the architecture of the ResNet50 [37].

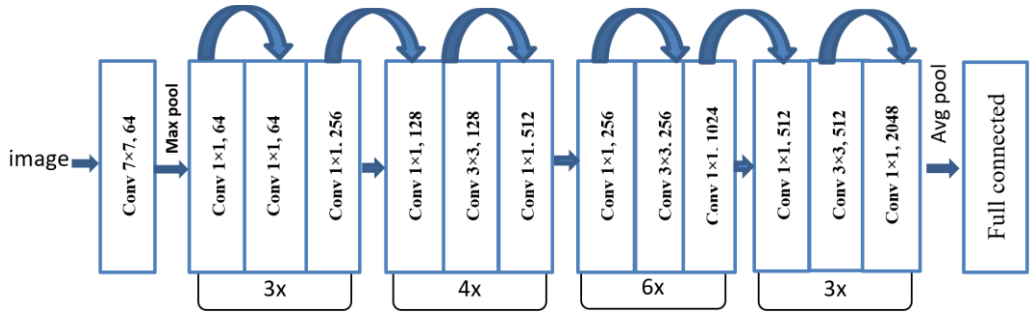

Figure 3. depicts the ResNet-50 model architecture.

3.2.3. YOLOv8 Model Structure

YOLO (You Only Look Once) has revolutionised object detection with its integrated network design that concurrently detects object bounding boxes and classifies labels. Over years, YOLO has evolved through several versions, with the eighth iteration released in January 2023, highlighting significant enhancements [38].

- Backbone: YOLOv8 features an improved Cross Partial Stage (CSP) architecture that divides feature maps for more efficient convolution; this reduces computational complexity while preserving strong learning capabilities. The backbone is built on the C2f module, which is a faster adaptation of the CSP inspired by an ELAN structure used in YOLOv7. Additionally, the incorporation of the SPPF module improves performance in multi-scale detection [39].
- Neck: YOLOv8 includes the PAN-FPN module in its neck, which facilitates effective multi-scale feature fusion. This architecture merges the advantages of the PAN and FPN models, enabling the upper layers to capture detailed contextual information while the lower layers retain accurate spatial details.
- Head: The YOLOv8 features the unique head design that separates classification from detection. Unlike previous anchor-based methods, it uses an anchor-free approach. This means it identifies objects by locating their centres and estimating the distances to the edges of the bounding boxes, eliminating the necessity for predefined anchors.

Figure 4 shows the YOLOv8 model structure, chosen for its lightweight design, which allows for real-time object detection with high performance across various scales [40].

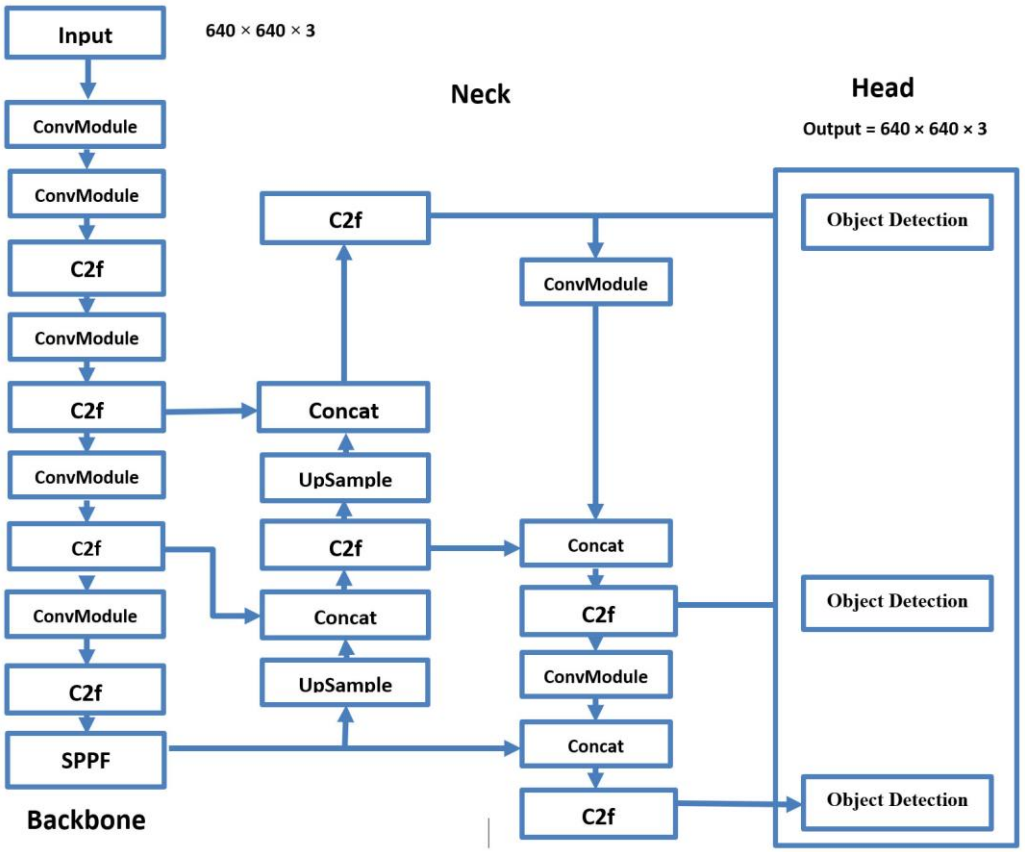

Figure 4. YOLOv8 Model Structure.

4. Datasets

This study classifies distracted driver behaviours using in-vehicle images from the State Farm dataset and the Driver Monitor Dataset, which was introduced in Kaggle. The dataset contains labelled images of ten distinct activities taken during the day for State Farm dataset and five classes taken during the night for the Driver Monitor Dataset, including normal driving, texting, phone use, Cigarette, Closed Eye, drinking, grooming, and conversing with passengers.

The State Farm dataset consists of a labelled training set with 22,424 images and an unlabeled test set of 79,726 images. This study uses the training set, which covers ten categories of driver behaviour: normal driving (c0), messaging with the right hand (c1), speaking on the phone with the right hand (c2), messaging with the left hand (c3), speaking on the phone with the left hand (c4), adjusting the radio (c5), drinking (c6), reaching behind (c7), grooming (c8), and conversing with passengers (c9). Figure 5 summarises the image attributes for each category. The dataset is split into 80% will be allocated for training, 10% will be allocated for testing, and 10% will be allocated for validation to ensure effective model training and evaluation while minimising overfitting. The driver observation dataset includes night-time IR camera images categorised into five main categories for evaluating driver distraction of safety-related: seat belt, phone, cigarette, closed eye, and open eye. The dataset comprises the following number of images: 2039 for Seatbelt, 1495 for Phone, 2423 for Cigarette, 2365 for Closed Eye, and 2359 for Open Eye. Figure 6 summarises the image attributes for each category. Because of this even distribution, detection models can be developed that are both robust and accurate. The dataset is allocated as follows: 80% for training, 10% for validation, and 10% for testing.

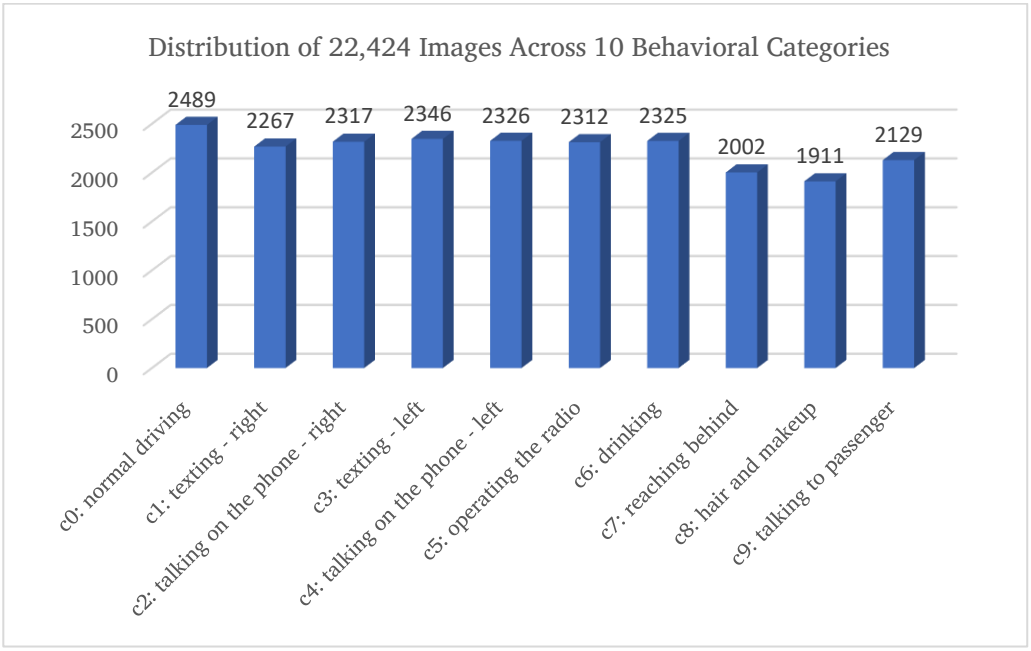

Figure 5. Distribution of Driver Behaviour Classes in the State Farm Dataset.

5. System Architecture Overview

This study investigates deep learning models such as simpler CNN, YOLOv8, ResNet50 and VGG16 for classifying and detecting driver distractions. The research aims to identify the most accurate and efficient model for monitoring driver activity and enhancing road safety by utilising transfer learning and extensive training on labelled datasets from State Farm and Driver Monitor. Algorithm 1 describes the functionality of these techniques.

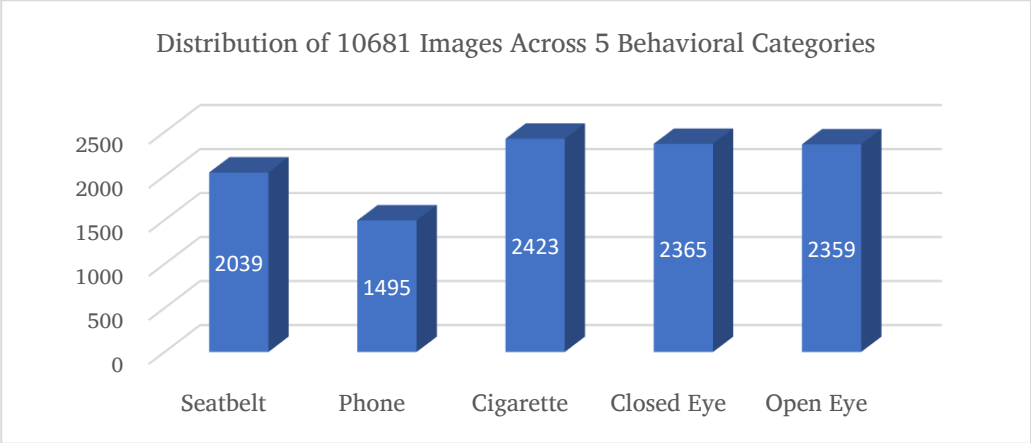

Figure 6. Shows the distribution of the five behaviour classes in the DMD.

Algorithm 1: Driver Distraction Detection by using the simpler CNN, VGG16, ResNet50, and YOLOv8 models:

1. Input: The State Farm and the Driver Monitor Datasets contain labelled images for 15 classes of driver distraction behaviours.
2. Preprocessing: Load and resize all images to 256x256 pixels, normalise pixel values between 0 and 1, and apply data augmentation techniques (rotation, flipping, zooming) to enhance dataset diversity and minimise overfitting.
3. Model Selection:
   a) Model1 (Convolutional Neural Network):
      - Use multiple convolutional layers with ReLU activations for feature extraction.
      - Use max pooling to decrease the dimensionality.
      - Flatten the feature maps and apply dense layers.
      - Use softmax activation to output class probabilities.
   b) Model2 (VGG16 with Transfer Learning):
      - Load the pre-trained VGG16 model with weights, excluding the top classification layers.
      - Freeze all layers except the last five layers.
      - Flatten an output of a last pooling layer.
      - Add custom dense layers and batch normalisation.
      - Apply softmax activation for the multi class classification.
   c) Model3 (ResNet50 with Transfer Learning):
      - ✓ Load the pre-trained ResNet50 model with its weights, omitting the top classification layers.
      - ✓ Freeze all layers except for the last three.
      - ✓ Flatten the output from the residual layers.
      - ✓ Append custom dense layers and apply batch normalisation.
      - ✓ Use a softmax activation function to perform the classification.
   d) Model4 (YOLOv8 for object detection):
      - ✓ Load the pre-trained YOLOv8 model for object detection.
      - ✓ Directly pass the image through the YOLO model to output the predicted class.
4. Use the Adam optimiser with a learning rate of 0.001, applying categorical cross-entropy as the accuracy and loss function in the classification tasks.
5. Training
   - ✓ Divide the dataset into test, training, and validation sets.
   - ✓ Train the models using the following settings:
     - ✓ Learning rate: 0.001
     - ✓ Batch size: 32
     - ✓ Optimiser: Adam

- ✓ Monitor both training and validation accuracy after each epoch.
- ✓ Implement early stopping techniques to validate loss to avoid overfitting while training models for 15 epochs, consistently monitoring their performance at every stage.

6. Evaluate each trained model on an unseen test set, comparing metrics such as validation loss, validation accuracy, training loss, and training accuracy while analysing performance, primarily focusing on YOLO model speed and efficiency for object detection.
7. Comparison and Analysis: The models were compared based on their accuracy in detecting distractions, computational efficiency, and suitability. VGG16 and ResNet50 excelled in classification tasks but required more processing time, while YOLO offered faster detection suitable for applications with a minor accuracy trade-off.
8. Deployment Considerations: Based on the results, the most precise and effective model was selected for potential deployment in driver monitoring systems, with exploration into integrating the model to trigger warnings or interventions when driver distraction is detected.

6. Result and Discussion

Two open-source datasets (the State Farm and the Driver Monitoring) were utilised to classify fifteen distinct types of driver distractions, enabling effective detection across diverse lighting conditions. Deep learning models like simpler CNN, VGG16, ResNet50, and YOLOv8, were utilised to detect distracted driving. Both VGG16 and ResNet50, which were pre-trained on the State Farm dataset and DMD, achieved high accuracy through transfer learning. Meanwhile, YOLOv8 demonstrated superior speed and detection efficiency in real-time. Each model was trained for 15 epochs, and performance was carefully monitored to prevent overfitting. Figure 7 illustrates examples of detected distracted driving behaviours.

Figure 7. illustrates a variety of driver activities.

The CNN architecture starts with multiple Conv2D layers that are responsible for feature extraction. Each Conv2D layer is paired with a MaxPooling2D layer to downsample the spatial dimensions of the data and includes Batch Normalisation to stabilise the activations. The network progressively increases the number of filters from 16 to 64, enabling the network to capture increasingly complex features. After feature extraction, the output is flattened and passed through two Dense layers with 512 and 1024 units, respectively, before reaching the final Dense layer, which has 10 neurons and uses the softmax activation function for classification. Table 1 summarises the model's details, highlighting that it has 29 million parameters and demonstrating its efficiency in image classification through effective pooling and regularisation techniques.

LAYER	TYPE : OUTPUT SHAPE WITH PARAMETERS
conv2D	Conv2D: Output (None, 254, 254, 16) with 448 parameters
max_pooling2D	MaxPooling2D: Output (None, 127, 127, 16) with 0 parameters
conv2D(1)	Conv2D: Output (None, 125, 125, 32) with 4,640 parameters
max_pooling2D(1)	MaxPooling2D: Output (None, 62, 62, 32) with 0 parameters
conv2D(2)	Conv2D: Output (None, 60, 60, 64) with 18,496 parameters
max_pooling2D_(2)	MaxPooling2D: Output (None, 30, 30, 64) with 0 parameters
Flatten(1)	Flatten: Output (None, 57,600) with 0 parameters
Dense(8)	Dense: 512 neurons with 29,491,712 parameters
Dense(9)	Dense: 1,024 neurons with 525,312 parameters
batch_normalization(3)	Batch Normalisation: Output (None, 1,024) with 4,096 parameters
Dense(10)	Dense: 10 neurons with 10,250 parameters

Table 1. The layers and parameters of The ConvNet Model.

Table 2 details the CNN's performance over 15 epochs, highlighting consistent improvement in training and validation metrics. Initially, training accuracy was 21.30% and validation accuracy was 13.29%. By the fifth epoch, these values rose to 61.36% and 63.29%, respectively. Training reached 77.37% and validation at 75.90% between the sixth and tenth epochs, with a significant loss reduction. At the fourteenth epoch, validation accuracy peaked at 88.74% with a loss of 0.3274. With early stopping, the final model achieved 83.39% training accuracy and 85.81% validation accuracy, demonstrating effective generalisation.

Epoch	**Training_Accuracy**	**Training_Loss**	**Validation_Accuracy**	**Validation_Loss**
1	21.30%	2.3138	13.29%	2.2197
2	41.65%	1.7088	38.29%	1.9599
3	50.29%	1.4894	52.25%	1.5354
4	56.84%	1.2452	60.81%	1.4425
5	61.36%	1.1189	63.29%	1.0685
6	67.01%	0.9965	72.07%	0.8172
7	66.91%	0.9483	52.48%	1.4160
8	73.64%	0.7969	81.08%	0.5542
9	75.37%	0.7391	70.72%	0.8689
10	77.37%	0.6708	75.90%	0.7258
11	79.89%	0.6260	79.50%	0.6515
12	80.21%	0.5717	74.55%	0.7923
13	81.04%	0.5612	82.43%	0.5364
14	81.63%	0.5135	**88.74%**	**0.3274**
15	83.39%	0.4946	85.81%	0.4979

Table 2. Training Results of the CNN Model Over Fifteen Epochs.

Figure 8(a) illustrates the model achieving a training accuracy of 90.45% and a validation accuracy of 88.57%, demonstrating effective learning and generalisation. Figure 8(b) shows the loss plot, with training and validation losses steadily decreasing and minimal overfitting observed, attributed to the convergence of the curves, batch normalisation, and efficient feature extraction by the convolutional layers.

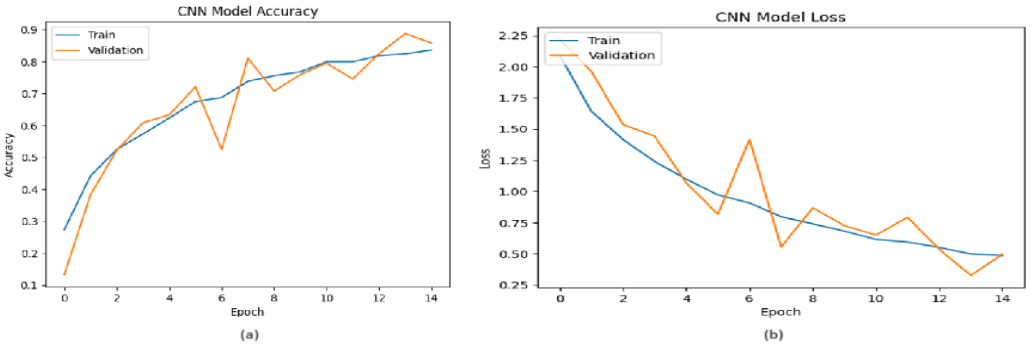

Figure 8. Graphs Illustrating the CNN Model Loss and Accuracy During Validation and Training.

The VGG16 model serves as the feature extractor and is further enhanced with additional dense layers to classify the driver distraction State Farm dataset. This architecture combines the pre-trained VGG16 convolutional layers with fully connected dense layers, batch normalisation, and dropout techniques to accomplish the final classification task, as shown in Table 3.

VGG16 architecture	Layers and parameter counts
Input Layer	Dimensions (256, 256, 3), 0 parameters.
Convolutional Layers (Block 1)	Two Conv2D layers (64 filters each), totalling 38,720 parameters (with a total of 1,792 and 36,928 parameters), followed by MaxPooling2D reducing spatial dimensions to (128, 128, 64).
Convolutional Layers (Block 2)	Two Conv2D layers (128 filters each), totalling 221,440 parameters (with 73,856 and 147,584 parameters), followed by MaxPooling2D reducing dimensions to (64, 64, 128).
Convolutional Layers (Block 3)	Three Conv2D layers (256 filters each), totalling 1,475,328 parameters (with 295,168, 590,080, and 590,080 parameters), followed by MaxPooling2D reducing dimensions to (32, 32, 256).
Convolutional Layers (Block 4)	Three Conv2D layers, each with 512 filters, account for a total of 5,899,776 parameters, broken down as 1,180,160, 2,359,808, and 2,359,808 respectively, followed by a MaxPooling2D layer that reduces the dimensions to (16, 16, 512).
Flatten Layer	Converts feature maps into a single vector (131,072 units).
Dropout Layer	Prevents overfitting by randomly disabling neurons during training.
First Dense Layer	128 neurons, 16,777,344 parameters, followed by Batch Normalization.
Second Dense Layer	256 neurons, 33,024 parameters, followed by Batch Normalization.
Output Layer	10 neurons (final classification), 2,570 parameters.

Table 3. VGG16 Pre-Trained Model with Dense Layers.

The model was compiled using the adam optimiser and categorical cross entropy loss. It achieved optimal performance by leveraging feature extraction and mitigating overfitting with dropout and batch normalisation. The VGG model's performance is improved until it reached a peak training accuracy of 93.03%, a validation accuracy of 98.42%, and a low validation loss of 0.0630 by epoch 10, as shown in Table 4 and Figure 9. The early stopping mechanism ensured the model was restored to this optimal state, highlighting its ability to generalise effectively and accurately classify driver distractions.

Epoch	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss
1	26.58%	2.2992	33.56%	2.2762
2	65.95%	1.0329	69.82%	0.8151
3	79.02%	0.5974	84.46%	0.4737
4	85.28%	0.4547	92.12%	0.2748
5	88.43%	0.3720	88.74%	0.3241
6	89.85%	0.3032	92.79%	0.2049
7	90.92%	0.2964	94.37%	0.1619
8	90.67%	0.2850	96.17%	0.1245
9	92.45%	0.2414	97.30%	0.1161
10	93.03%	0.2107	**98.42%**	**0.0630**
11	93.03%	0.2138	97.07%	0.0910
12	94.41%	0.1654	96.40%	0.0943
13	93.27%	0.2114	97.30%	0.0710
14	94.13%	0.1847	97.97%	0.0570
15	95.42%	0.1632	96.85%	0.1052

Table 4. Training Results of the VGG16 Model Over Fifteen Epochs.

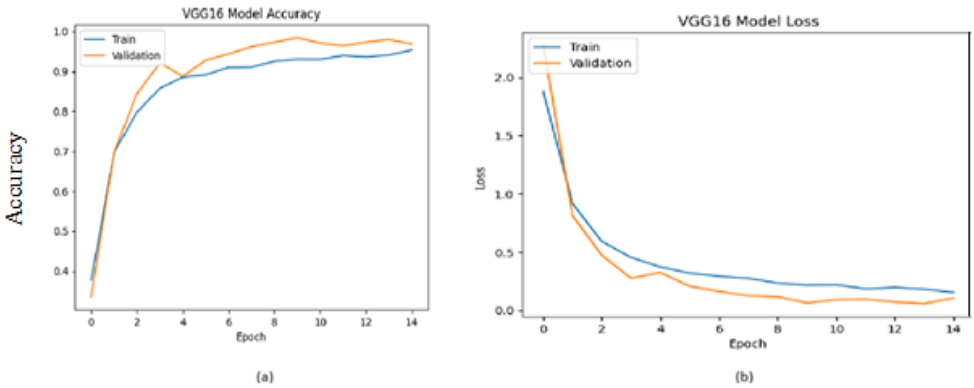

Figure 9. Graphs the Loss and Accuracy During Validation and Training for the VGG16 model.

The ResNet50 model minimises the risk of overfitting and guarantees stable training by freezing the earlier layers. To improve the stability of the ResNet50 model for driver distraction detection, five dense layers with batch normalisation were added, as seen in Table 5. Both the deep residual and the dense layers were kept frozen during the training process. The ResNet50 model effectively utilises pre-trained weights, while the last three layers are made trainable.

ResNet50 architecture	Output shapes and parameter counts
Input Layer	The model starts with an input layer of size ($256 \times 256 \times 3$). A ZeroPadding2D layer is applied to expand the input dimensions to ($262 \times 262 \times 3$).
Initial Convolutional Block	• A Conv2D layer with 64 filters (9,472 parameters) processes the padded input to produce a feature map of ($128 \times 128 \times 64$). • This process includes batch normalisation (256 parameters) and ReLU activation, which contribute to stabilising the network. • MaxPooling2D reduces the spatial dimensions to ($64 \times 64 \times 64$).

Residual Block (conv2_block1)	• Main Path: Two Conv2D layers with 64 filters each, BatchNorm and ReLU are applied, resulting in an output of $(64 \times 64 \times 64)$. • Shortcut Path: A Conv2D layer with 256 filters (16,640 parameters) followed by BatchNorm. • Both paths are combined via an Add layer, producing a $(64 \times 64 \times 256)$ feature map.
Advanced Convolutional Block	• A Conv2D layer with 512 filters (524,800 parameters) generates an $(8 \times 8 \times 512)$ output, followed by BatchNorm (2,048 parameters) and ReLU activation. • The feature map is then passed through another Add layer and output $(8 \times 8 \times 2048)$.
Flattening and Dropout	• The feature map is flattened into a vector of 131,072 units. • A Dropout layer is applied to reduce overfitting.
Fully Connected Layers	• A dense layer with 128 units (16,777,344 parameters) is followed by BatchNorm (512 parameters). • A second dense layer consisting of 256 units (33,024 parameters) is introduced, followed by Batch Normalization (1,024 parameters). • The final output is produced by a dense layer with 10 units (2,570 parameters).

Table 5. The layers and parameters of the ResNet50 model.

The ResNet50 model is trained over 15 epochs and demonstrated significant improvement. The training accuracy rose from 17.73% to 65.24%, while the validation accuracy reached 74.55%. Additionally, the validation loss decreased notably from 11.1845 in the first epoch to 0.7474 by the 10th epoch, highlighting the effectiveness of transfer learning. However, performance plateaued after the 10th epoch, suggesting a potential need for further adjustments, like fine-tuning a learning rate or incorporating regularisation to enhance stability, as shown in Table 6.

Epoch	Train Accuracy	Train Loss	Validate Accuracy	Validate Loss	Time per Epoch (s)
1	0.1773	2.6027	0.0968	11.1845	550
2	0.3429	1.8421	0.1104	4.9851	564
3	0.4342	1.6166	0.1779	5.1718	535
4	0.4462	1.5588	0.3761	1.8280	534
5	0.5051	1.4173	0.2973	5.7755	512
6	0.5176	1.3586	0.3941	2.1522	531
7	0.5462	1.2807	0.4054	1.8823	530
8	0.5844	1.1987	0.5991	1.1240	528
9	0.5766	1.1884	0.6937	0.8567	528
10	0.6211	1.1193	0.7680	0.7474	506
11	0.6125	1.1163	0.6982	0.8633	518
12	0.6296	1.0709	0.7275	0.8295	578
13	0.6458	1.0442	0.7793	0.6865	560
14	0.6400	1.0412	0.6802	0.9194	559
15	0.6524	1.0025	0.7455	0.7572	565

Table 6. Training of the ResNet50 model across 15 epochs.

Figure 10 illustrates the ResNet50 model's consistent improvement in training and validation accuracy over 15 epochs, ultimately reaching 65% and 74%, respectively. Additionally, there is a steady reduction in loss. However, the early epochs showed minor fluctuations in validation performance; the model stabilised by epoch 10, achieving a test accuracy of 77.58%. This demonstrates its strong potential for classifying driver behaviour, with opportunities for further enhancement through fine-tuning.

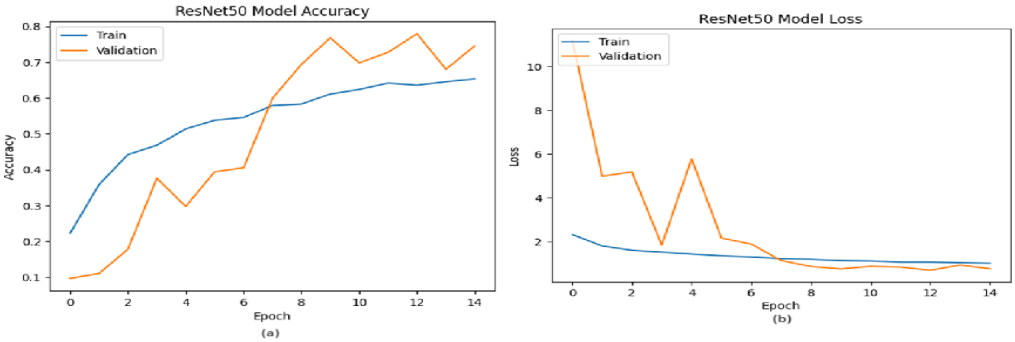

Figure 10. Graphs the Loss and Accuracy During Validation and Training for the ResNet50 model.

The YOLOv8 model was fine-tuned using the State Farm dataset, which includes ten behaviour classes for distracted drivers labelled from c0 to c9 to achieve effective detection. The model utilised pre-trained weights from the yolov8n-cls.pt file. Its architecture incorporates convolutional layers and C2f blocks to enhance feature extraction. The Adam optimiser regulates the learning rate and momentum, ensuring optimal performance.

Table 7 outlines the architecture of the YOLOv8 model, which consists of 99 layers, 1,451,098 trainable parameters, and a computational cost of 3.4 GFLOPs. This streamlined architecture, featuring Conv2D layers and C2f blocks, enables lightweight yet highly accurate detection of driver distractions.

Layer_Number	Layer_Type	Output_Shape	Parameters
0	Conv2D	(None, 112, 112, 16)	464
1	Conv2D	(None, 56, 56, 32)	4,672
2	C2f	(None, 56, 56, 32)	7,360
3	Conv2D	(None, 28, 28, 64)	18,560
4	C2f	(None, 28, 28, 64)	49,664
5	Conv2D	(None, 14, 14, 128)	73,984
6	C2f	(None, 14, 14, 128)	197,632
7	Conv2D	(None, 7, 7, 256)	295,424
8	C2f	(None, 7, 7, 256)	460,288
9	Classify (Head)	(None, 10)	343,050

Table 7. Provides an overview of the YOLOv8 model's architecture.

Figure 11 shows a steady decrease in both training and validation loss over 15 epochs. This trend demonstrates the effectiveness of the YOLOv8 model in learning and its strong ability to generalise. The gradual convergence of these loss metrics towards low values indicates reduced overfitting and indicates strong performance in detecting driver distraction.

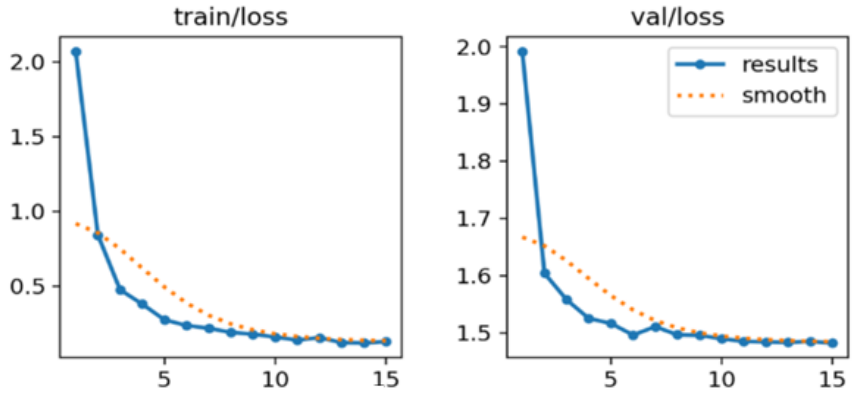

Figure 11. Graphs the Loss During Training and Validation for the YOLOv8 model

Figure 12 demonstrates that the YOLOv8 models top-1 accuracy steadily increases, approaching nearly 99% by the 10th epoch. Meanwhile, the top-5 accuracy reaches 100% within the first few epochs. This robust and consistent performance highlights the model's high reliability for detecting driver distraction.

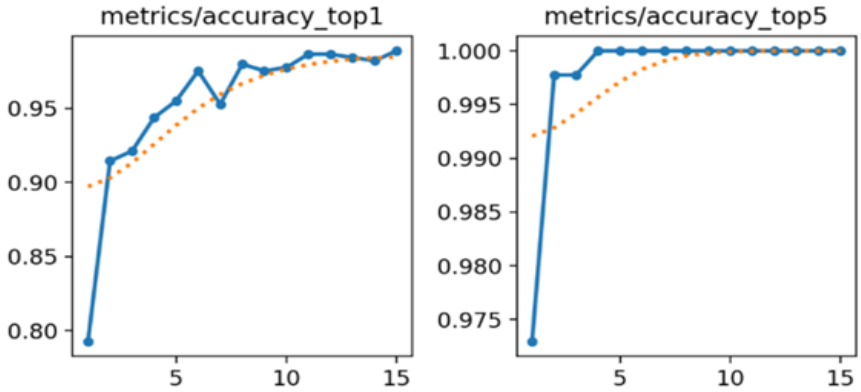

Figure 12. Graphs the Accuracy During Training and Validation of the YOLOv8 Model.

Figure 13 illustrates the normalised confusion matrix indicates for the State Farm dataset that the YOLOv8 model achieved nearly flawless classification, attaining 100% accuracy in most classes (c1, c2, c3, c4, c5, c7, and c9). There were only minor misclassifications in classes c0, c6, and c8, leading to an overall test accuracy of 99.44%. This matrix underscores the model's strong predictive performance, as evidenced by the prominent diagonal values representing accurate detection for each category, thereby validating the reported accuracy.

Figure 14 shows the models YOLOv8, VGG16, and ResNet50 applied to DMD. YOLOv8 outperforms the other models, recording a Top-1 accuracy of 97.7%. Its validation loss stabilises at approximately 1.06, making it highly reliable for real-time detection of driver behaviours such as distraction, drowsiness, or phone usage. Next, the VGG16 model returns a confidence of 96.3% and a slightly higher validation loss of 1.45, which makes for a nice compromise of performance and calculation cost. On the other hand, ResNet50 has the lowest accuracy rate at Top-1 (92.9%) and the highest validation loss of 1.25, indicating less generalisation and overfitting. From these results, YOLOv8 is the ideal selection for DMD applications that need a high level of precision and robustness.

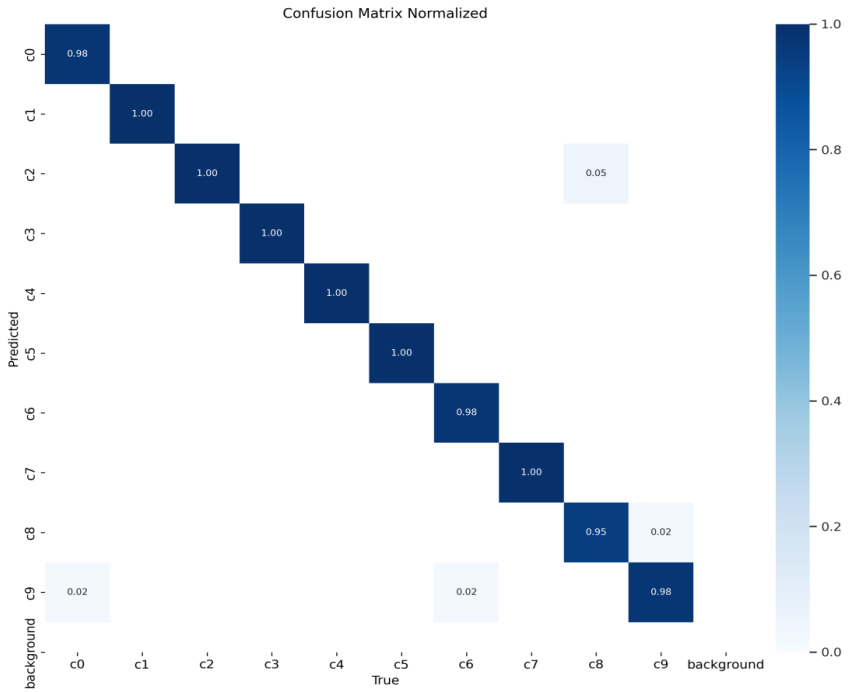

Figure 13. The YOLOv8 model confusion matrix.

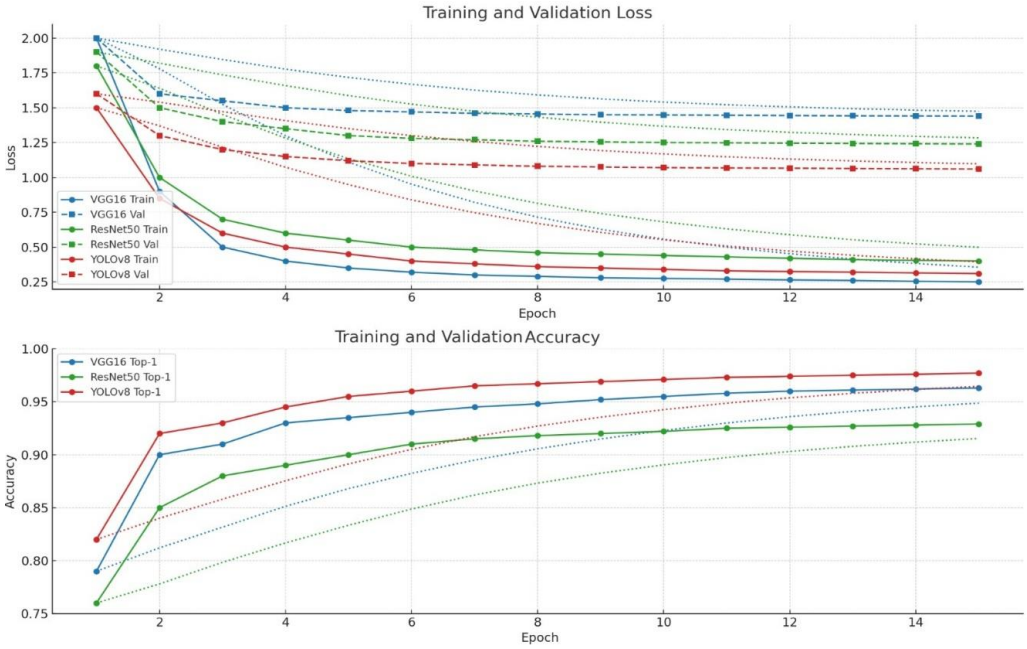

Figure 14. Comparison of VGG16, ResNet50, and YOLOv8 Models in DMD: Training and Validation Losses and Accuracy.

As shown in Figure 15, the proposed YOLOv8's performance on DMD images demonstrates its ability to detect crucial behaviours, including eye status (open/closed), smoking, and seatbelt wearing, in real-time under varying lighting and angles. This validates the effectiveness and reliability of YOLOv8 for practical usage in monitoring drivers.

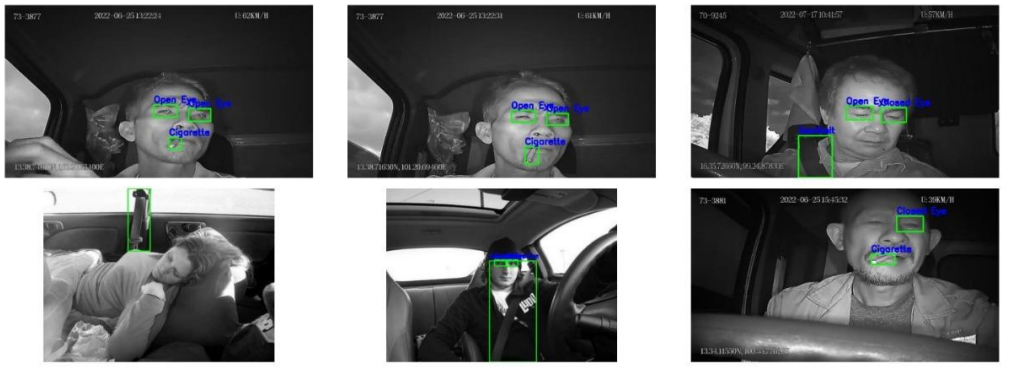

Figure 15. Real-Time Detection of Driver Behaviours Using YOLOv8 in DMD

The Figure 16 displays confusion matrices for VGG16, ResNet50, and YOLOv8, which are evaluated on DMD in five classes: open eye, closed eye, cigarette, phone, and seatbelt. VGG16 exhibits strong overall accuracy, although it occasionally confuses visually similar classes, particularly between the open eye and closed eye. The ResNet50 performs poorly with increased abortion and low precision, especially in phone and seatbelt categories. On the other hand, Yolov8 achieves the best results, displaying the highest true positive rates and minimal confusion between squares. This underscores its robustness and suitability for real-time driver behaviour detection.

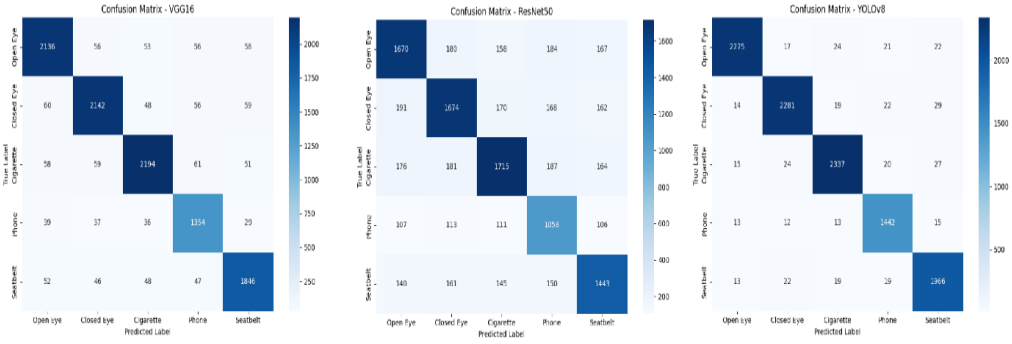

Figure 16. Confusion Matrices of VGG16, ResNet50, and YOLOv8 Models on the DMS Dataset

Figure 17 compares the performance of four models for driver distraction detection across two datasets: the State Farm and the Driver Monitoring. In the State Farm dataset, the accuracy attained the highest percentage in YOLOv8 with 98.46 %, while VGG16 recorded 97.58%, showing that both models are robust in the feature extraction performance. A CNN model was accuracy-wise scored 82.20% which means that increasing its complexity or the tuning process fine-tuning could potentially improve its performance. The ResNet50 model achieved a performance accuracy of 77.80%. However, when evaluated on the DMD, all models experienced a drop in performance reflecting the increased difficulty or variability within the DMD images. YOLOv8 remained the top performer with 96.46%, maintaining high robustness across both datasets. VGG16 also showed strong generalisation with 90.58%, while CNN and ResNet50 saw larger declines, achieving 67.24% and 70.80%, respectively. These results highlight YOLOv8's superior adaptability and confirm its effectiveness for real-world driver behaviour detection tasks.

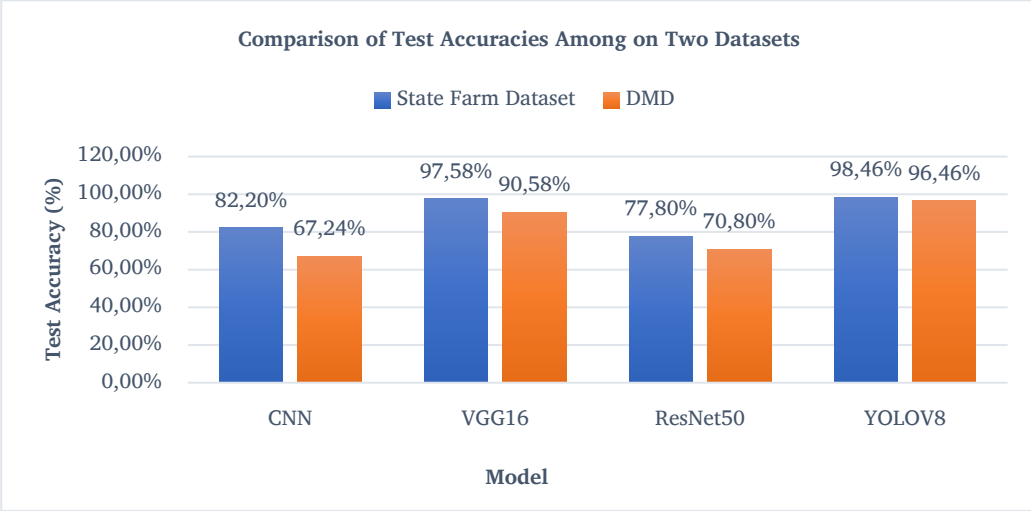

Figure 17. Test Accuracy Results for the proposed work.

The figure 18 illustrates the real-time implementation of the trained YOLOv8-based driver monitoring system. The algorithm was applied using a standard webcam to detect and classify driver behaviours across multiple individuals and scenarios. The system successfully identifies various states such as normal driving, texting right, drinking, and smoking, in addition to critical indicators like open eyes, phone usage, and seatbelt status. By providing real-time detections, the model demonstrates its worth and efficiency for practical situations, successfully establishing behavioural patterns in several scenarios. This confirms the model's readiness for deployment in live driver assistance systems aimed at improving road safety.

Figure 18. illustrates the outcomes of the real-time identification of normal driving, texting, drinking, open eyes, phone usage, and seatbelt status.

7. Conclusion

Distracted driving is now a major concern in recent years. This study examines three neural network architectures: VGG16, YOLOv8, and ResNet50, to detect distracted drivers using two open-source datasets to provide diversity in the number of classes as well as to predict distracted driving in different lighting conditions, day and night and different camera angles. The total data trained in the system exceeds 33,000 images. The dataset was augmented and preprocessed to improve performance and transfer learning was used to train models over fifteen categories of distracted driving. The results show that YOLOv8 performed better than other models once implemented on the State Farm dataset with a resulting test accuracy of 98.46%. VGG16 was followed with great precision with accuracy rate of 97.58%, where effective transfer learning and feature extraction techniques were used. Despite the fact that CNN and ResNet50 models delivered fairly good results, YOLOv8 was faster and more accurate. YOLOv8 maintained its best performance after applying it to the DMD dataset, achieving 96.46%. VGG16 also demonstrated strong generalisation, achieving 90.58%, while CNN and ResNet50 achieved 67.24% and 70.80%, respectively. In totality, speed and precision in YOLOv8 make it the best tool for driver monitoring systems. YOLOv8 facilitates the timely detection of distracted behaviours, applied to real-time images captured from a camera and identifies different driver distraction categories successfully, supporting seamless integration into modern car dashboards that ultimately enhance road safety.

The limitation of this study is that we found the VGG16 model outperforms ResNet50 when applied to two types of datasets. Despite ResNet50 being deeper and more complex, with over 23 million parameters, it requires more training epochs and better learning rate scheduling to achieve optimal performance. In our work, we utilised a relatively short training duration of only 15 to 20 epochs.

Future research could consider different deep learning architectures, specifically their capacity to learn from a limited number of images, in order to add accuracy with fewer samples. This area has received little research attention. Learning based on Sample-Based Learning (SBL) has shown potential in improving accuracy with fewer samples. Additionally, there is a need for careful investigation into the ability to predict human behaviour using visual features, such as mouth movements, or physiological indicators that can identify anomalies in data. A hybrid approach to detecting driver distraction could be developed by combining deep learning techniques with an embedded search strategy.

References

[1] E. Michelaraki, C. Katrakazas, S. Kaiser, T. Brijs, and G. Yannis, “Real-time monitoring of driver distraction: State-of-the-art and future insights,” *Accid Anal Prev*, vol. 192, p. 107241, 2023, doi: https://doi.org/10.1016/j.aap.2023.107241.

[2] K. Khan, S. B. A. Zaidi, and A. Ali, “Evaluating the Nature of Distractive Driving Factors towards Road Traffic Accident,” *Civil Engineering Journal*, vol. 6, pp. 1555–1580, Aug. 2020, doi: 10.28991/cej-2020-03091567.

[3] A. Muthuswamy, M. A. A. Dewan, M. Murshed, and D. Parmar, “Driver Distraction Classification Using Deep Convolutional Autoencoder and Ensemble Learning,” *IEEE Access*, vol. 11, pp. 71435–71448, 2023, doi: 10.1109/ACCESS.2023.3293110.

[4] A. Misra, S. Samuel, S. Cao, and K. Shariatmadari, “Detection of Driver Cognitive Distraction Using Machine Learning Methods,” *IEEE Access*, vol. 11, pp. 18000–18012, 2023, doi: 10.1109/ACCESS.2023.3245122.

[5] A. N. Jaafar, “Enhancement of Breast Cancer Classification Using Bat Feature Selection with Recurrent Deep Learning.,” *J. Comput. Inf. Technol.*, vol. 32, no. 3, pp. 195–215, 2024, [Online]. Available: http://cit.fer.hr/index.php/CIT/article/view/5801

[6] A. Kashevnik, R. Shchedrin, C. Kaiser, and A. Stocker, “Driver Distraction Detection Methods: A Literature Review and Framework,” *IEEE Access*, vol. 9, pp. 60063–60076, 2021, doi: 10.1109/ACCESS.2021.3073599.

[7] M. Al-Bayati, “Deep Learning Model for Predicting Spreading Rates of Pandemics, ‘COVID-19 as Case Study,’” *Journal of information and organizational sciences*, vol. 48, pp. 253–262, Dec. 2024, doi: 10.31341/jios.48.2.1.

[8] M. H. Alkinani, W. Z. Khan, and Q. Arshad, “Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges,” *IEEE Access*, vol. 8, pp. 105008–105030, 2020, doi: 10.1109/ACCESS.2020.2999829.

[9] J. Wang *et al.*, "A Survey on Driver Behavior Analysis From In-Vehicle Cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10186–10209, 2022, doi: 10.1109/TITS.2021.3126231.

[10] V. Yaloveha, D. Hlavcheva, and A. Podorozhniak, "Spectral Indexes Evaluation for Satellite Images Classification using CNN," *Journal of information and organizational sciences*, vol. 45, pp. 435–449, Dec. 2021, doi: 10.31341/jios.45.2.5.

[11] Y. Chen, Y. Chen, S. Fu, W. Yin, K. Liu, and S. Qian, "VGG16-based intelligent image analysis in the pathological diagnosis of IgA nephropathy," *J Radiat Res Appl Sci*, vol. 16, no. 3, p. 100626, 2023, doi: https://doi.org/10.1016/j.jrras.2023.100626.

[12] Y. Chen *et al.*, "Automated Alzheimer's disease classification using deep learning models with Soft-NMS and improved ResNet50 integration," *J Radiat Res Appl Sci*, vol. 17, no. 1, p. 100782, 2024, doi: https://doi.org/10.1016/j.jrras.2023.100782.

[13] S. D, S. J, A. S, A. K, S. P, and P. K, "Proactive Headcount and Suspicious Activity Detection using YOLOv8," *Procedia Comput Sci*, vol. 230, pp. 61–69, 2023, doi: https://doi.org/10.1016/j.procs.2023.12.061.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[16] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[17] A. D. Alvarez, F. S. Garcia, J. E. Naranjo, J. J. Anaya, and F. Jimenez, "Modeling the Driving Behavior of Electric Vehicles Using Smartphones and Neural Networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 3, pp. 44–53, 2014, doi: 10.1109/MITS.2014.2322651.

[18] N. Vaegae, K. Pulluri, K. Bagadi, and O. Oyerinde, "Design of an Efficient Distracted Driver Detection System: Deep Learning Approaches," *IEEE Access*, vol. 10, pp. 116087–116097, Nov. 2022, doi: 10.1109/ACCESS.2022.3218711.

[19] A. Dhiman, A. Varshney, F. Hasani, and B. Verma, "A Comparative Study on Distracted Driver Detection Using CNN and ML Algorithms," in *Proceedings of International Conference on Data Science and Applications*, M. Saraswat, C. Chowdhury, C. Kumar Mandal, and A. H. Gandomi, Eds., Singapore: Springer Nature Singapore, 2023, pp. 663–676.

[20] M. Aljasim and R. Kashef, "E2DR: A Deep Learning Ensemble-Based Driver Distraction Detection with Recommendations Model," *Sensors*, vol. 22, p. 1858, Feb. 2022, doi: 10.3390/s22051858.

[21] A. V Jamsheed, B. Janet, and U. S. Reddy, "Real Time Detection of driver distraction using CNN," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 185–191. doi: 10.1109/ICSSIT48917.2020.9214233.

[22] H. Mittal and B. Verma, "CAT-CapsNet: A Convolutional and Attention Based Capsule Network to Detect the Driver's Distraction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 9561–9570, 2023, doi: 10.1109/TITS.2023.3266113.

[23] C. Guo, H. Liu, J. Chen, and H. Ma, "Temporal Information Fusion Network for Driving Behavior Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 9415–9424, 2023, doi: 10.1109/TITS.2023.3267150.

[24] K. Alshalfan and M. Zakariah, "Detecting Driver Distraction Using Deep-Learning Approach," *Computers, Materials & Continua*, vol. 68, pp. 689–704, Feb. 2021, doi: 10.32604/cmc.2021.015989.

[25] S. Liu, Y. Wang, Q. Yu, H. Liu, and Z. Peng, "CEAM-YOLOv7: Improved YOLOv7 Based on Channel Expansion and Attention Mechanism for Driver Distraction Behavior Detection," *IEEE Access*, vol. 10, pp. 129116–129124, 2022, doi: 10.1109/ACCESS.2022.3228331.

[26] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecol Inform*, vol. 48, pp. 257–268, 2018, doi: https://doi.org/10.1016/j.ecoinf.2018.10.002.

[27] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.

[28] B. Baheti, S. Gajre, and S. Talbar, "Detection of Distracted Driver Using Convolutional Neural Network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1145–11456. doi: 10.1109/CVPRW.2018.00150.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[30] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds., Cham: Springer International Publishing, 2018, pp. 270–279.

[31] Y. Gao and K. Mosalam, "Deep Transfer Learning for Image-Based Structural Damage Recognition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, Apr. 2018, doi: 10.1111/mice.12363.

[32] T. T. Huynh, H. T. Nguyen, and D. T. Phu, "Enhancing Fire Detection Performance Based on Fine-Tuned YOLOv10," *Computers, Materials and Continua*, vol. 81, no. 2, pp. 2281–2298, 2024, doi: 10.32604/cmc.2024.057954.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[34] X. Lu, H. Wang, J. Zhang, Y. Zhang, J. Zhong, and G. Zhuang, "Research on J wave detection based on transfer learning and VGG16," *Biomed Signal Process Control*, vol. 95, p. 106420, 2024, doi: https://doi.org/10.1016/j.bspc.2024.106420.

[35] P. K. Mannepalli, A. Pathre, G. Chhabra, P. A. Ujjainkar, and S. Wanjari, "Diagnosis of bacterial leaf blight, leaf smut, and brown spot in rice leafs using VGG16," *Procedia Comput Sci*, vol. 235, pp. 193–200, 2024, doi: https://doi.org/10.1016/j.procs.2024.04.022.

[36] K. Srinivasan *et al.*, "Performance Comparison of Deep CNN Models for Detecting Driver's Distraction," *Computers, Materials & Continua*, vol. 68, pp. 4109–4124, Apr. 2021, doi: 10.32604/cmc.2021.016736.

[37] L. Ali, F. Alnajjar, H. Jassmi, M. Gochoo, W. Khan, and M. Serhani, "Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures," *Sensors*, vol. 21, p. 1688, Mar. 2021, doi: 10.3390/s21051688.

[38] Y. Gong, Z. Chen, W. Deng, J. Tan, and Y. Li, "Real-Time Long-Distance Ship Detection Architecture Based on YOLOv8," *IEEE Access*, vol. PP, p. 1, Jan. 2024, doi: 10.1109/ACCESS.2024.3445154.

[39] R.-Y. Ju and W. Cai, "Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm," *Sci Rep*, vol. 13, no. 1, p. 20077, 2023, doi: 10.1038/s41598-023-47460-7.

[40] F. Solimani *et al.*, "Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity," *Comput Electron Agric*, vol. 218, p. 108728, 2024, doi: https://doi.org/10.1016/j.compag.2024.108728.