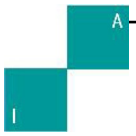
Inteligencia Artificial 28(75) (2025), 15-29
doi: 10.4114/intartif.vol28iss75pp15-29



INTELIGENCIA ARTIFICIAL
http://journal.iberamia.org/

Real Time Distracted Driver Detection Using Xception Architecture And Raspberry Pi

Uma Narayanan[1], Pavan Prajith[2], Rijo Thomas Mathew[3], Royal Alexandar[4] ,Vishnu Vikraman[5]
[1] Assistant Professor at Institute of Science and Technology, CVV, Kerala, India.
[2] Student of Rajagiri School of Engineering and Technology, Kerala, India

[1] uma.narayanan@cvv.ac.in
[2] pavanprajith@gmail.com
[3] rijo0502@gmail.com
[4] royalalexandar01@gmail.com
[5] vishnuvikraman1605@gmail.com

Abstract Researchers are currently focusing on the development of technologies to detect and warn drivers about driving while distracted, as it is a leading cause of traffic accidents. According to the National Highway Traffic Safety Administration's report, distracted driving is responsible for approximately one in every five car accidents. Our objective is to establish a reliable and precise method for identifying distracted drivers and notifying them about their lack of focus. We are drawing inspiration from the success of convolutional neural networks in computer vision for this task. Our approach involves implementing a CNN-based system that can accurately detect when a driver is distracted and identify the specific cause of their distraction. Real-time detection, however, necessitates three apparently mutually exclusive requirements for an optimal network: a small number of parameters, high accuracy, and fast speed.

Keywords: Neural Network, CNN, Xception, VGG16, ResNet50

1 Introduction

The World Health Organization (WHO) reports 1.35 million deaths and 50 million injuries from road traffic accidents per year, making them a serious global problem. A substantial portion of these accidents, approximately 45%, are caused by distracted drivers. In India alone, road traffic fatalities increased by 3.2% from 2015 to 2016, as per a Ministry of Transport Research report. These accidents also have a significant economic impact, with a 3% loss in the GDP of most nations. Distracted driving is defined by the National Highway Traffic Safety Administration as any activity that takes the driver's focus away from the wheel, which is the main reason for traffic accidents.

This paper aims to address the critical issue of distracted driving, which significantly contributes to road accidents and fatalities. Real-time distracted driver detection through deep learning is vital for enhancing road safety and preventing accidents caused by activities like texting and phone use.

1.1 Motivation and Problem statement

Despite the advancement in vehicle safety technologies, distracted driving remains a persistent problem that impairs a driver's ability to react swiftly to unexpected road events. Traditional methods of addressing this issue are often reactive rather than proactive. There is an urgent need for real-time systems that can detect and mitigate distracted driving before accidents occur.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1.2 Proposed Solution

Our research proposes developing a real-time distracted driver detection system using deep learning algorithms. Convolutional Neural Networks (CNNs), a type of deep learning, have shown potential in image identification applications. This study uses the State Farm Distracted Driver Datasets (SFDDD) to identify and categorize driver distractions using the Xception model. A camera mounted inside the car and linked to a Raspberry Pi is used by the system to take an RGB picture of the driver. The Xception model is used to identify objects associated with distracting activities and isolate body part regions of interest (ROI) in images.

1.3 Key Contribution

1. Propose a distracted driving detection model tested using the SFDDD Dataset.
2. Implement various models and choose the best model with optimal results for detecting distracted driving.
3. Identify the body's regions of interest and objects to detect distracted driving.
4. Implement the project for real-time detection.

1.4 Importance of Real-Time Detection

1. **Timely Intervention:** Deep learning algorithms process real-time data from vehicle cameras and sensors to identify signs of distracted driving swiftly. This enables immediate alerts, helping drivers refocus and reducing accident risks.
2. **Accident Prevention:** Distracted driving impairs quick reactions to unexpected road events. Real-time detection prompts drivers to regain focus, lowering collision probabilities and ensuring safer driving conditions.
3. **Data-Driven Insights:** Deep learning models recognize diverse distracting behaviors, aiding authorities and manufacturers in understanding patterns and causes, leading to better safety policies and measures.
4. **Personalized Alerts:** Deep learning adapts to individual driving behaviors, offering personalized alerts for unsafe practices, encouraging better habits over time.
5. **Continuous Monitoring:** Deep learning-based systems operate 24/7, ensuring consistent surveillance of driver behavior, a crucial aspect considering distractions can occur anytime and anywhere.
6. **Evolving Technology:** Improved deep learning algorithms enhance distraction detection accuracy, promising even more reliable real-time systems in the future.
7. **Autonomous Vehicle Precedent:** Real-time distracted driver detection is a crucial step towards autonomous vehicles. Integrating such systems with self-driving technology ensures drivers are ready to take control, maintaining safety during transitions.

The proposed method proactively identifies and addresses distractions, curbing accident risks. This approach has the potential to save lives, reduce injuries, and create safer roads for all. The results of our study concluded that the Xception model, with an accuracy of 85%, provided superior real-time detection results compared to other models like Inception and ResNet, which experienced overfitting problems and gave incorrect predictions.

2 Related Works

This section summarizes the research work and several neural network models that have been developed for the use of distracted driver detection. Distracted driver detection work is getting considerable attention from the research community as the number of accidents are increasing due to driver distraction. It has been found that actions like the use of smartphones and careless driving increase the chance of accidents and such behaviors must be prohibited to prevent accidents. Various models of CNN including VGG16, VGG19, Inception, Xception, EfficientDet etc were implemented for driver distraction detection. Considering the real-time implementation is necessary the majority of existing models fail to satisfy the requirements like accuracy level and time requirement.

Authors in [1] used the state farm dataset with eight different classes of activities calling, texting, everyday driving, operating on radio,In activeness, talking to a passenger, looking behind, and drinking of 26 different

persons for proposing a new model with EfficientDet and EfficientDet. The model provides accurate predictions by picking items that are present in the distracting activities and body part ROI. Five different variants, including EfficientDet(D0-D4) for detection, were constructed in order to evaluate the ideal EfficientDet version with Faster R-CNN and Yolo-V3. Based on the experiment, the recommended approach outperforms all existing cutting-edge models, and EfficientDet-D3, which achieves an average mean Precision of 99.16% when using an epoch fifty, batch size four, and step size 250, which is excellent is the best model for distracted driver detection.

Authors [3] proposed a new D-HCNN model using the concept of decreasing filter size (12x12, 9x9, 6x6, 3x3) with only 0.76M parameters which is very much lesser than any other existing for distracted driver detection. On the AUCD2 dataset, it achieves 1.15% higher accuracy than the native VGG with only 0.5% of the VGG parameters. To enhance the performance metrics, D-HCNN makes use of HOG (Histogram Oriented gradient) pictures, dropout, batch normalization, and L2 regularization. An eight-dimensional histogram was created for each image in the input network model for HOG feature extraction. This was achieved by taking two by two sections from 224 × 224 grayscale photos [3]. By using HOG images, it is possible to drop useless background information and more focus can be given to the driver's movement and body posture. Using the SFDDD and AUC Distracted Driver (AUCD2) datasets, test assessments of the D-HCNN model were conducted. The accuracy on SFD3 is 99.87%, and on AUCD2 it is 95.59%.

In their study [5], the authors aimed to develop a CNN model capable of detecting distracted drivers and identifying the causes of distraction through hand and face localization. They conducted a comparative analysis of various CNN architectures, including VGG-16, ResNet50, and MobileNetV3. Their analysis showed that a number of factors, including dropout values, batch size during simulation, number of epochs, and dataset attributes, affected how accurate these models performed. Notably, the results indicated that MobileNetV2 exhibited superior performance with lower weight compared to the other models. For their experiments, the authors utilized 4,000 and 6,000 images from the State Farm dataset, which They set aside 20% of the photographs for testing and 80% of the images for training. The results showed that ResNet50 and MobileNetV2 had greater accuracy rates 94.50 percent and 98.12 percent, respectively. This model has the potential to be employed in real-time distracted driver detection systems. In 2020, a new Driver Monitoring Dataset (DMD) was introduced [6], and the authors of [7] utilized this dataset for their research. DMD consisted of recordings taken from three inside-the-car viewpoints, each having three cameras positioned to take pictures of the driver's hands, face, and body. Three channels were available for each camera: RGB, infrared, and depth data. 37 volunteers, including 27% women, worked together to collectively generate the dataset, which is a complete resource for studies on driver monitoring.

In their work [8], the authors proposed a model that incorporates three distinct models, synergistically combined into a unified framework. This method comprises three essential modules. The Agreeable CNN module, first and foremost, incorporates three pre-prepared models, specifically ResNet50, Xception, and Origin V3, to extricate highlights from pictures. To line up with the info prerequisites of these models, the first pictures (640 x 480 x 3) go through preprocessing, bringing about components of 224 x 224 x 3 for ResNet50 and 299 x 299 x 3 for both Commencement V3 and Xception. The subsequent module, known as the Component Link module, profoundly intertwines the element vectors created by the Helpful CNN module. The three 1x2048 component vectors are converged into a 3x2048 vector, which is in this manner leveled into a 1x6144 vector. This vector fills in as the contribution for the third module, the Component Characterization module, liable for preparing the loads of the element vectors and arranging the outcomes into ten unmistakable classes. ResNet50 accomplishes the most exactness out of the three pre-prepared models, yet at the most elevated misfortune. ResNet50 explicitly accomplishes a precision of 93.72% on the testing set and 99.29% on the preparation set. Instead of Xception, which accomplishes 99.53% exactness on the preparation set and 91.08% precision on the testing set, Beginning V3 gets 95.31% precision on the preparation set and 90.13% precision on the testing set. Curiously, the Hybrid Cooperative Fusion (HCF) move toward brings down misfortune while likewise further developing precision (99.95% on the preparation set and 96.74% on the validation set).

In another paper, the creator [9] tried to coordinate sensor information into a dream based diverted driver identification model, stressing the system's speculation capacity. This mix brought about a two-stage model. The principal stage included the formation of a dream based Convolutional Neural Network (CNN), while the subsequent stage consolidated sensor information utilizing a Long Short Trem Memory-Recurrent Neural Network (LSTM-RNN). Two datasets, including SFDDD[2] dataset and the second dataset containing OBD sensor information, whirligig and accelerometer information, and driver pictures, were used. Trial results exhibited that the incorporation of sensor information expanded the exactness of ordinary driving recognition from 74% to 85%. Also, both half breed and completely forecast level combination based LSTM models yielded comparable correctnesses of 85%. Moreover, CNN models in view of VGG16 and Beginning V3 accomplished

ISSN: 1137-3601 (print), 1988-3064 (on-line)

correctnesses of 77% and 81%, separately, after fine-tuning. Lang Su et al. [11] employed the DADCNet (Driver Anomaly Detection and Classification Network) to efficiently allocate multimodal and multiview inputs in their research.

In the paper[12], the authors introduced models based on two robust deep learning architectures, namely Visual Geometric Groups (VGG-16) and Residual Networks (ResNet50). The proposed framework comprises several components, utilizing image augmentation techniques, a pre-processing module, and two classification models based on deep learning architectures—ResNet50 and VGG-16. Initially, the data undergoes pre-processing, where the images are resized into a more manageable 64 x 64 x 3 matrix using the Python CV2 library. This resizing step enhances the computational efficiency of the classifier. Subsequently, the data is augmented to achieve optimal results. A DDDS (Distracted Driver Detection System) model is constructed through transfer learning and is deployed to categorize State-Farm images into one of the ten driving classes. Notably, VGG-16 and ResNet50 exhibit strong performance in detecting distracted driving, achieving accuracies of 86.1% and 87.92%, respectively. The proposed model demonstrates particular efficiency in identifying categories c4, c5, and c7.

Authors in [13] used one of the pre-trained models - VGG16 with deep convolutional neural network to classify images. The main idea behind the modeling is using the concept of transfer learning as it is an arduous task to collect related data and rebuild the existing model. After building the model the next process is to fine tune the model using the concept of data augmentation[14]. One of the pre-trained VGG-16 is used to classify images and check accuracy for training data as well as the validation data. The VGG-16 model consists of thirteen convolutional layers and two fully connected layers and 1 SoftMax classifier. The first model developed using neural networks gave an accuracy of 72.40 %. When data augmentation applied for fine tuning the data the accuracy level reached 79.20 %. The pre-trained VGG-16 trained on large dataset of images and then fine-tuned with image augmentation to achieve accuracy of 95.40 %. Benjamin Wagner et al. [15] used four different CNN models trained and were evaluated for driver posture classification. The first two models are given by a modified ResNeXt-34 and ResNeXt-50.

The study[16] introduces a pragmatic approach by quantifying driver actions and crafting a classification system capable of identifying distractions. The paper presents a diverse ensemble of deep learning models, chosen for their effectiveness in classifying driver distractions and providing in-car recommendations. This initiative aims to curtail distraction levels and enhance in-car awareness, fostering heightened safety. The novel scalable model E2DR, which makes use of stacking ensemble techniques, is presented in this study. E2DR improves accuracy, generalization, and overfitting reduction by combining two or more deep learning models and provides real-time recommendations. Combining the ResNet50 and VGG16 models yields an astounding 92% test accuracy in the best E2DR version. Thanks to innovative data splitting approaches, this performance evaluation is extended to well-known datasets, such as the SDDD. In the study[17], they present an approach centered on a convolutional neural network (CNN) to effectively identify distracted drivers and discern the specific causes behind their distractions, including conversing, sleeping, or eating. This identification leverages face and hand localization. For transfer learning, the research uses four different architectures: VGG-16, ResNet50, MobileNetV2, and CNN. Employing these architectures, we endeavor to develop a robust model.

While existing research emphasizes enhancing distraction detection performance using convolutional neural networks (CNNs) and recurrent neural networks (RNNs), there remains a noticeable gap in the study of distracted driver detection through pose estimation. Addressing this gap, the work introduces the Optimally-weighted Image-Pose Approach (OWIPA) [18], an ensemble of ResNets. The innovative framework is designed to classify distractions using both original images and pose estimation images. To generate pose estimation images, we employ HRNet and ResNet, thereby expanding the scope of our model. ResNet101 is used for original photos in the classification process, while ResNet50 is used for images with pose estimation. Notably, a grid search technique is used to find an ideal weight that is critical to the ensemble's performance. This weight parameter then harmonizes the predictions from both models, effectively amalgamating their outputs. Through this strategic approach, our work strives to bridge the gap in the realm of distracted driver detection and pose estimation, contributing to a more comprehensive understanding of driver behavior.

Vehicle detection under different climatic conditions has been an area of significant research interest. Using the corner point approach, Mallikarjun Anandhalli and Vishwanath P. Baligar [19] presented a method to detect

cars in a variety of environmental circumstances. Corner-based statistical modeling for vehicle detection under diverse conditions was proposed, primarily for traffic surveillance, as a further extension of this work [20]. This research, featured in Multimedia Tools and Applications, provides a comprehensive analysis of vehicle detection performance across different weather conditions, reinforcing the necessity for adaptable detection algorithms in traffic monitoring systems. Additionally, Anandhalli and his team [21] explored the image projection method for vehicle speed estimation in video systems. This study, published in Machine Vision and Applications, contributes to the field by addressing vehicle speed estimation, which is critical for traffic management and safety enforcement, particularly in adverse weather conditions.

3 Proposed Frameworks

Four phases make up our suggested model. The preprocessing of the dataset is done in the first stage. Using the Xception model, we identify the objects participating in the distracting activities in the second and third phases. We sound an alarm based on the forecast in the final stage.

3.1 Pre-processing of Dataset

State Farm organised a competition in April 2016 that was held on Kaggle with the main goal of collecting pictures of distracted driving behaviour. We used the SFDDD, a set of 2D photos taken by cameras mounted on various vehicles' dashboards, to create our suggested algorithm. The main goal of acquiring these photographs was to evaluate their impact and develop measures to lower the number of fatalities attributable to distracted driving, which would improve road safety statistics. The SFDDD originally consisted of two separate folders, each comprising 79,727 test photos and 22,400 training images. These pictures were all taken with 640 x 480 pixel resolution using dashboard-mounted cameras. As shown in Figure 1a, the training dataset was divided into ten separate categories, each of which represented a variety of actions, including using the left or right hand to make a call or send a text, driving as usual, adjusting the radio, moments of inactivity, speaking with a passenger, turning around, and drinking beverages. As seen in figure1 a, there were a variety of photographs in each category. Only the training dataset was used to assess the effectiveness because the test dataset lacked labels. Figure 1b provides a visual representation of select frames from the distracted driver dataset, offering a glimpse into the diverse range of scenarios captured within this valuable dataset.

3.2 Xception

To develop our algorithm, we utilize meticulously preprocessed and annotated data to educate the Xception model. This model serves as the cornerstone for scrutinizing individual picture frames and making determinations regarding the driver's state of distraction. Each image is first converted into a textual label that corresponds to one of our predetermined categories for image classification and detection. Then, using the Xception model, items and regions of interest (ROI) pertinent to the body portions involved in distracting activities are found. This strategy guarantees accurate predictions and produces cutting-edge results. The methodology consists of a series of processes that start with the extraction of convolutional features from the input image, continue with image classification, object detection, ROI identification, and end with predictions generated from a fusion of classification and detection labels.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

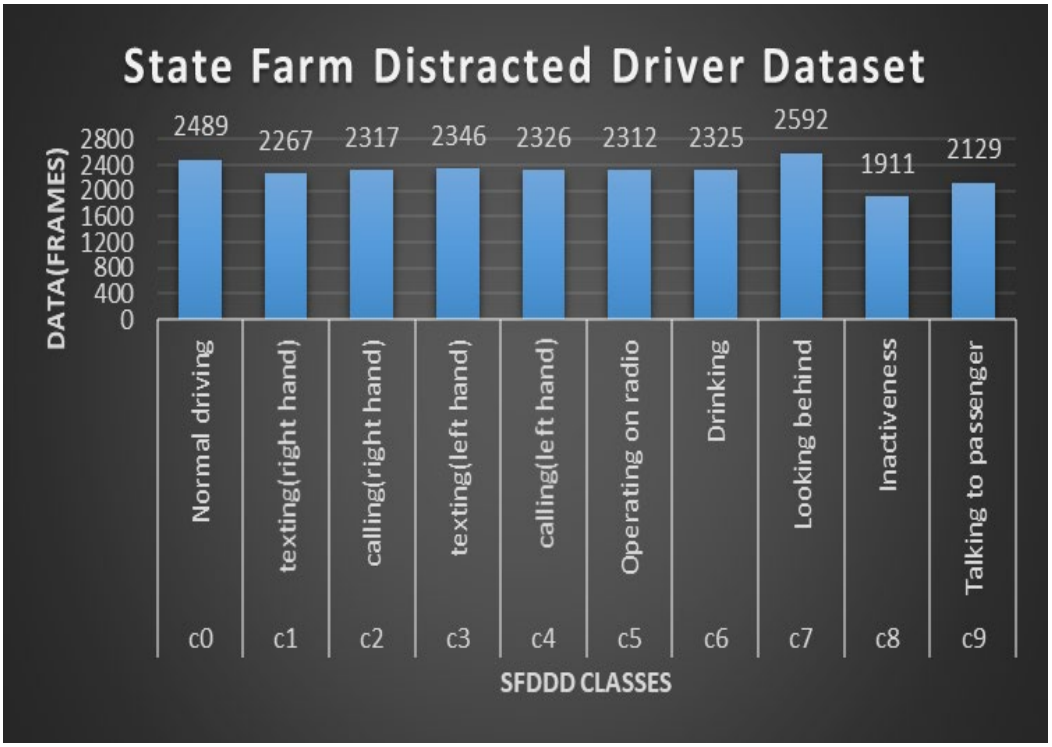


Figure 1a : Summary of SFDDD dataset[2]

Figure 1b: Sample images in SFDDD[2] Dataset

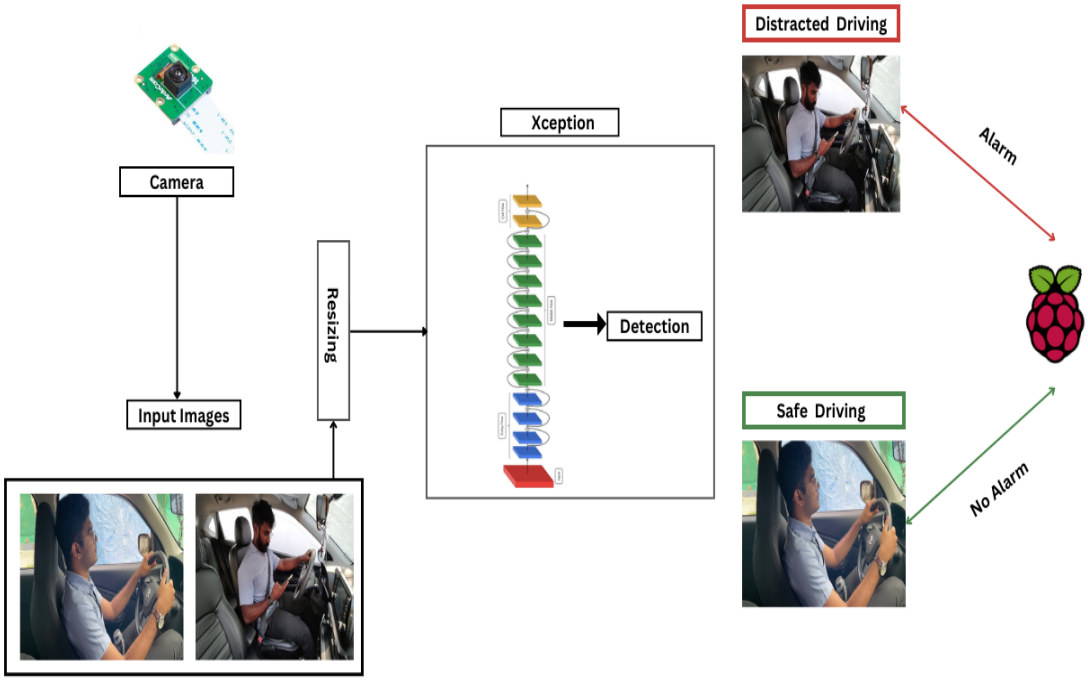


Figure 2: Proposed model overview

Our deep learning model, illustrated in Figure 2, works using image frames taken by a covert camera hidden on the dashboard of the car. These frames go through significant preprocessing, then the Xception model, which was previously pre-trained on a substantial picture classification dataset, is fine-tuned. The next step is to use Xception to locate things and give descriptions of the particular body parts participating in distracting activities. This comprehensive approach empowers us to accurately detect instances of driver distraction. Furthermore, employing Xception, we scrutinize objects within the designated areas of interest associated with distracting behaviors and ascertain the specific body parts involved. This multifaceted analysis culminates in the determination of the driver's distracted behavior.

3.3 Xception Model Architecture

The Xception model was introduced by François Chollet in 2016 as a deep convolutional neural network architecture based on the Inception architecture, which utilizes parallel convolutional layers with variable filter sizes. However, the Xception model differs from the Inception architecture as each Inception module is replaced with a depthwise separable convolution block that consists of a depthwise convolution layer and a pointwise convolution layer. This design results in fewer parameters and computations needed while still maintaining high accuracy. The Xception model incorporates skip connections, batch normalization, and ReLU activation functions in each layer and has 36 convolutional layers. Due to its efficiency and high accuracy, the Xception architecture is widely used for object identification and image classification tasks.

In our paper , the Xception model is pre-trained on the ImageNet dataset and its output is fed into fully connected layers that are added to the model. The fully connected layer consists of 2 hidden layers with 2048 neurons each, ReLU activation, and L2 regularization. Dropout is applied after each fully connected layer to reduce overfitting. The output layer has 10 neurons with softmax activation for multi-class classification. The model is compiled using stochastic gradient descent (SGD) as the optimizer with a learning value of 0.001 and categorical cross-entropy as the loss function. The model is evaluated using accuracy as the metric. We achieved a final accuracy of 85.58%.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

3.4 Raspberry PI and Alarm System

Now, shifting our focus to hardware, the Raspberry Pi Model 3B+ emerges as a cost-effective and compact single-board computer introduced in 2018. It represents a marked improvement over its predecessor, the Raspberry Pi iii Model B, delivering enhanced performance and introducing several novel features. The Model 3B+ is equipped with a robust 1.4 GHz quad-core ARM Cortex-A53 processor, significantly boosting processing speed. Moreover, it boasts dual-band 802.11ac wireless LAN and Bluetooth 4.2 capabilities, significantly enhancing connectivity options. The Ethernet port has also been upgraded to support speeds of up to 300 Mbps, and the card incorporates Gigabit Ethernet via USB 2.0, facilitating faster network connectivity. In terms of memory and storage, the Model 3B+ offers 1 GB of LPDDR2 SDRAM and includes a microSD card slot for storage expansion. Notably, the card is designed with comprehensive power management, featuring a 5V/2.5A DC input. Additionally, it supports Power over Ethernet (PoE) through a separate HAT (Hardware Attached on Top) accessory, adding to its versatility and functionality.

In the proposed method, raspberry pi is used to connect the camera module which is used to take snapshots of the driver and transmit it to the model and the alarm module which sends a signal to the driver once the model checks if the driver is distracted. In order to capture clear footage of the driver, a 16 MP camera is installed along with the Raspberry Pi. The camera is an Arducam 16MP IMX519 (NOIR) camera which is capable of capturing high-quality image and video. NOIR (No Infrared) version of this camera is designed for use in low-light environments. This allows the NOIR version to capture images with good contrast in low-light conditions.
.

4 Experimental Analysis and Result

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.85.12    Driver Version: 525.85.12    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P0    25W / 300W |      0MiB / 16384MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Figure 3: Collab GPU specifications

We divided our dataset into two parts for training purposes: 80% training data and 20% test data. We trained our model in google colab using a GPU hardware accelerator as shown in figure 3, premium GPU class and high

RAM run time shape. Before finalizing the Xception model , we trained and tested a few models. Some of them are :

4.1 ResNet50

A CNN model was created based on the pre-trained ResNet50 architecture. ResNet50 is a deep residual network that was trained on the ImageNet dataset for image classification tasks. We loaded the pre-trained ResNet50 model and set the input to be the shape of the images we want to classify. The output was then fed into fully connected layers that were added to the model. The activation of each feature map in the output is averaged in the first layer, which is a global average pooling layer. This lowers the model's parameter count and aids in avoiding overfitting. There are three more dense layers added, the latest one having 512 neurons and the first two having 1024 neurons each. Relu activation functions, which are used in these layers, give the model nonlinearity and enable us to comprehend more intricate representations of the input data. Dropout regularization is applied after the first and third dense layers to prevent overfitting. Dropout randomly turns off a percentage of the neurons in the layer during training, forcing the model to be more robust and generalizable features. Batch normalization is then applied after the second dense layer to improve the stability of the network.

Ten neurons with softmax activation for multi-class classification are present in the output layer at the end. Using a learning rate of 0.001, the optimizer SGD (stochastic gradient descent) is utilized to construct the model. We employ categorical cross-entropy as our loss function. Another option is Adam Optimizer. The metric used to assess models is accuracy. Figure 4 illustrates the 88% accuracy of our model; nevertheless, using our own test cases, the model produced inaccurate predictions.

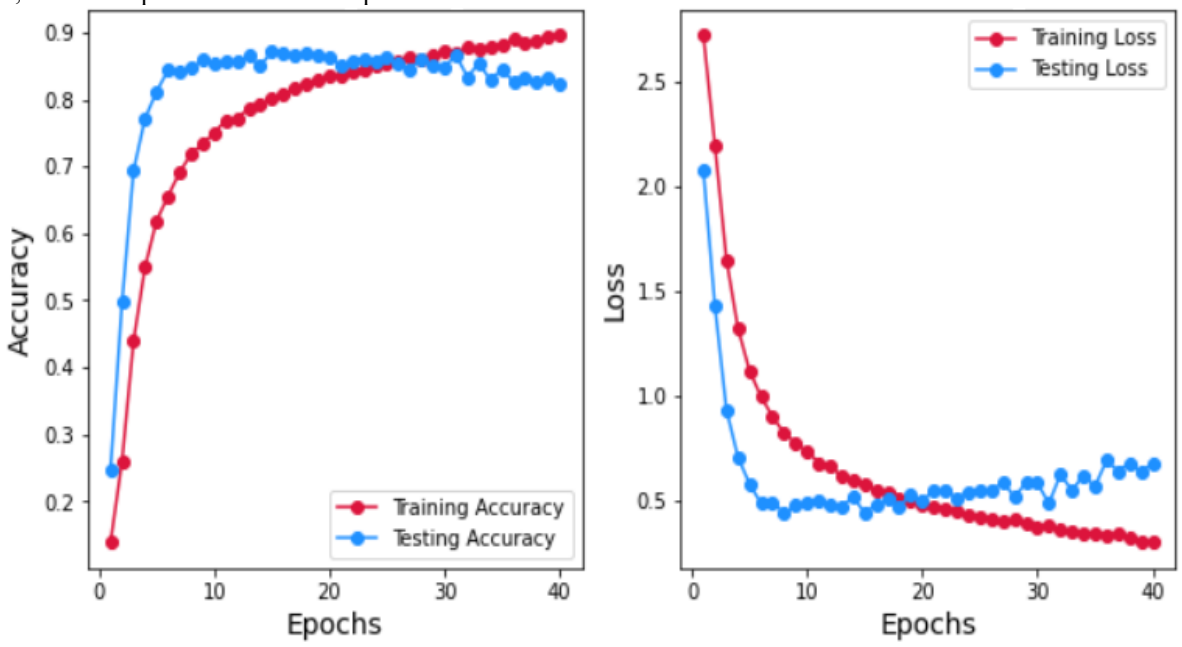


Figure 4: Accuracy and Loss graphs of ResNet50

4.2 VGG16:

With the help of the VGG-16 architecture, another pre-trained CNN model was developed.Although VGG16 contains twenty-one layers total—sixteen convolutional layers, five Max Pooling layers, and three Dense layers—it only has sixteen weight layers, or learnable parameters layers. The convolution and max pool layers are consistently arranged throughout the whole architecture. Utilizing the VGG16 convolutional neural network architecture as a foundational feature extractor, our approach incorporates multiple fully connected layers for classification. The introduction of a GlobalAveragePooling2D layer effectively reduces the output tensor's dimensionality by computing the average value of each feature map. To enhance the model's capacity to generalize, a dense layer housing 1024 units alongside a ReLU activation function is implemented, while a dropout regularization layer with a 0.1 rate is introduced to mitigate overfitting concerns.

An additional dense layer with 1024 units and ReLU activation is added to the model to further enhance its representational power. This layer is followed by a batch normalization layer for output tensor normalization.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Overfitting still needs to be addressed, and this is done by integrating a dense layer with 512 units and ReLU activation with a dropout regularization layer set at a 0.5 rate. As part of our optimization strategy, we employ the stochastic gradient descent (SGD) optimizer with a learning rate of 0.001. The architecture is completed with a final dense layer that consists of 10 units and a softmax activation function. This layer generates a probability distribution across all 10 classes. The model uses accuracy as the evaluation metric, SGD as the optimizer, and categorical cross-entropy as the loss function. Our VGG16-based model attains compelling performance, culminating in a final accuracy of 86.89%, a loss value of 0.3812, and a validation loss of 0.5298. This intricate model design, bolstered by judiciously chosen layers and regularization techniques, collectively contributes to its commendable accuracy and robust performance.

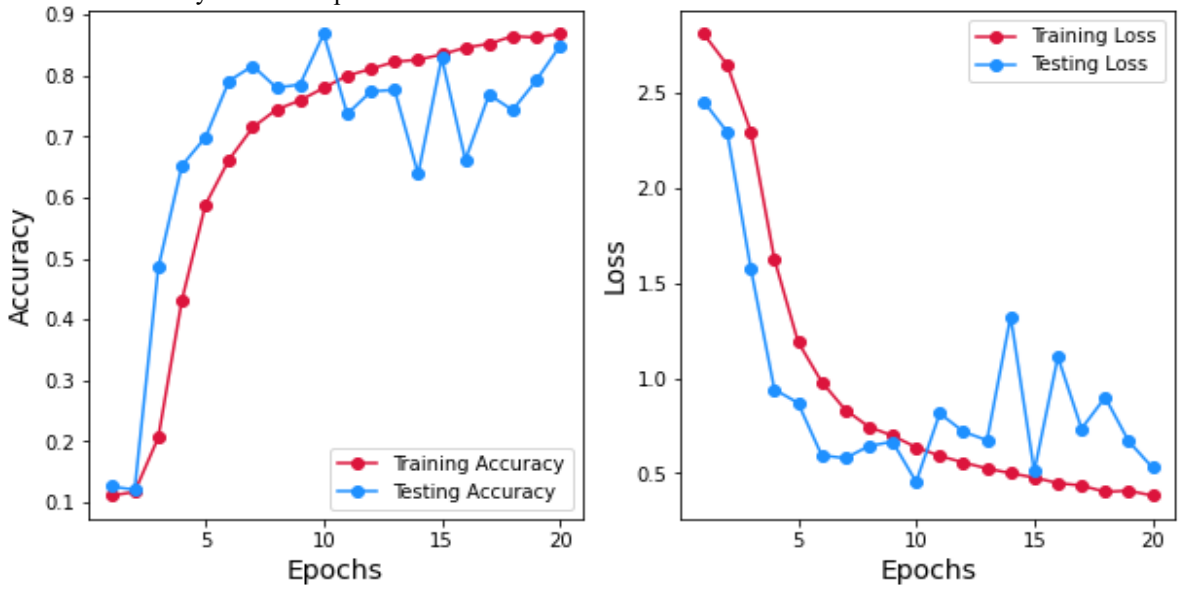


Figure 5: Accuracy and Loss graphs of VGG16

4.3 Xception

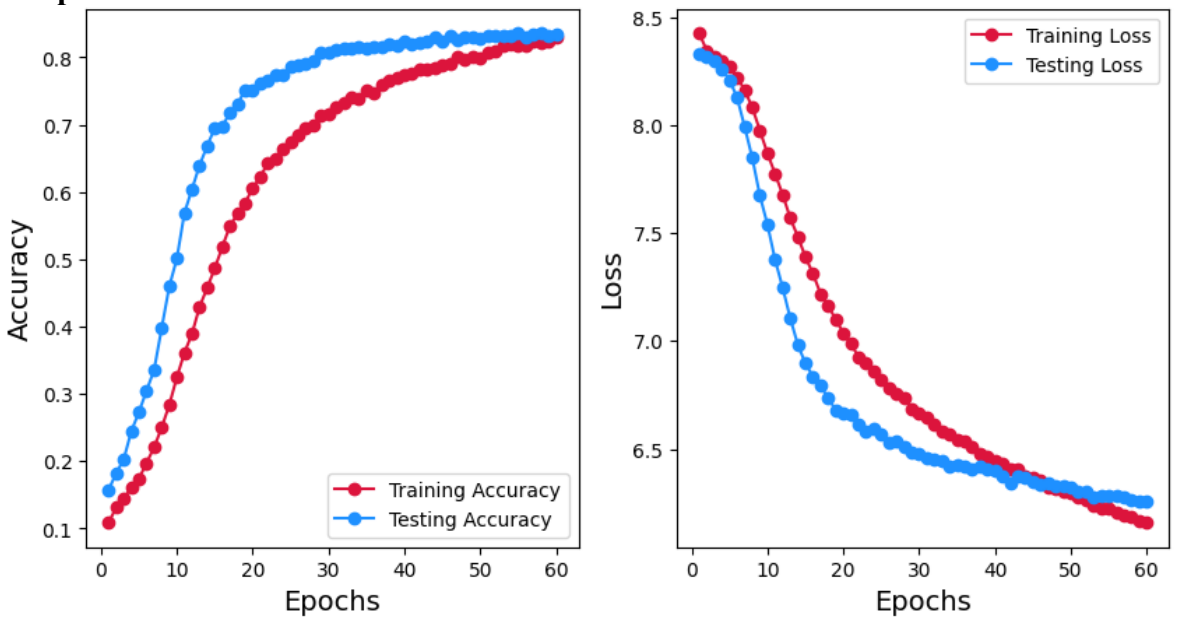


Figure 6: Accuracy and Loss graphs of Xception model

In our work, fully connected layers are added to the Xception model after it has been pre-trained on the ImageNet dataset. ReLU activation, L2 regularization, and two hidden layers with 2048 neurons each make up the fully linked layer. To lessen overfitting, dropout is performed following each completely linked layer. In the

output layer, there are ten neurons with softmax activation for multi-class categorization. Stochastic gradient descent (SGD) is used as the optimizer and categorical cross-entropy as the loss function to assemble the model with a learning value of 0.001. The model is evaluated using accuracy as the metric. We achieved a final accuracy of 85.58% as shown in figure 6. Even though its accuracy is comparatively less compared to other models, it predicted the test cases most correctly. By evaluating all the three models and their accuracy with predictions, we come to the conclusion that Xception would be the best model. Table 1 displays a summary of the literature. Figure 7 displays the various neural network frequencies of the various models employed in the comparison analysis, and Figure 8 displays the paper according to the year of publication.

Table 1 : Summary of Comparative study

Model/Module	**Purpose**	**Key Features**	**Usage in Distracted Driver Detection**	**Pros**	**Cons**
EfficientNet[1]	Scalable convolutional Neural network architecture.	Compound Scaling, Efficient architecture	Detecting driver distraction based on images from a dashboard camera	State-of-the-art accuracy with efficient model architecture	Limited research and applicability in distracted driver detection tasks
EfficientDet[1]	Object detection network based on EfficientNet	Efficient architecture, compound scaling	Detecting driver distraction based on images from a dashboard camera	State-of-the-art accuracy with efficient model architecture	Limited research and applicability in distracted driver detection tasks
Decreasing Filter size v/s Increasing Filter size[2]	Optimal model architecture for feature extraction	Varying filter size, comparison of performance	Determine optimal architecture for distracted driver detection models	Can enable the optimization of model size and computational efficiency while maintaining accuracy	Can lead to reduced model capacity and limited accuracy
Parameter Quantity Reduction and Accuracy Promotion[2]	Reduce model complexity and improve performance Improve Computation speed and efficiency	Pruning, quantization	Improve performance of distracted drive detection models on mobile devices	Enables the optimization of model size and computational efficiency while maintain accuracy	Can lead to reduced model capacity and limited accuracy
Parallelism of a Convolution Network[2]	Improve Computation speed and efficiency	Parallel computing, distributed learning	Enable real time distracted driver detection on edge devices	Enable faster inference on parallel computing	Limited applicability in single-GPU systems or when batch size is small
VGG16[4] [12] [13]	Deep convolutional neural network architecture	Sequential layers, max pooling	Detecting driver distraction based on image from a dashboard camera	Good balance between accuracy and model complexity	Limited applicability in real-time application due to computational

ISSN: 1137-3601 (print), 1988-3064 (on-line)

					demands
MobileNetV2 [4]	Light-weight convolution neural network architecture	Depthwise separable convolutions, efficient architecture	Detecting drive distraction based on images from a dashboard camera	Efficient model architecture with good accuracy	Limited capacity for capturing fine-grained details in images.
ResNet50 Model [4][7][12]	Improve classification accuracy	Residual connections, deep architecture	Detecting drive distraction based on images from a dashboard camera	Strong performance in image classification	Possibility of overfitting due to a large number of parameters.
Inception V3 Model[7]	Improve classification accuracy	Multiple parallel convolutions, reduce computation cost	Detecting drive distraction based on images from a dashboard camera	Good balance between accuracy and model complexity	Limited applicability in real-time applications due to computational demands.
Exception Model [7]	Improve classification accuracy	Attention mechanism, sparsity regulation	Detecting drive distraction based on images from a smartphone or smartwatch	State-of-the-art accuracy in image classification tasks	Very large model size and computational demands
Cooperative CNN Module [7]	Improve classification accuracy	Multi-stage processing, cooperative learning	Detecting drive distraction based on facial expression analysis	Good accuracy for image classification	Limited research and applicability in distracted driver detection tasks.
Feature Concatenation Model[7]	Combine features from multiple sources	Concatenate features, flexible architecture	Detecting drive distraction based on facial expression analysis and sensor data	Enable the combination of multiple sensor data for improved accuracy	Limited applicability in image-based driver distracted detection
Fusion of Sensor and Image-based Driver Detection Model[8]	Combine features from both sensor and image data	Feature fusion, multimodal learning	Detecting driver distraction based on both facial expression analysis and sensor data	Enables the combination of complementary information for improved accuracy	Can be computationally demanding and difficult to optimize jointly
LSTM RNN for Sequential Learning [9]	Model sequential dependencies in driver behaviour	Recurrent neural network long short-term memeory	Detecting driver distraction based on time series data from sensors or cameras	Can can caputure temporal dependiencies in driver behavior	Can be prone to overfitting and difficult to train with limited data

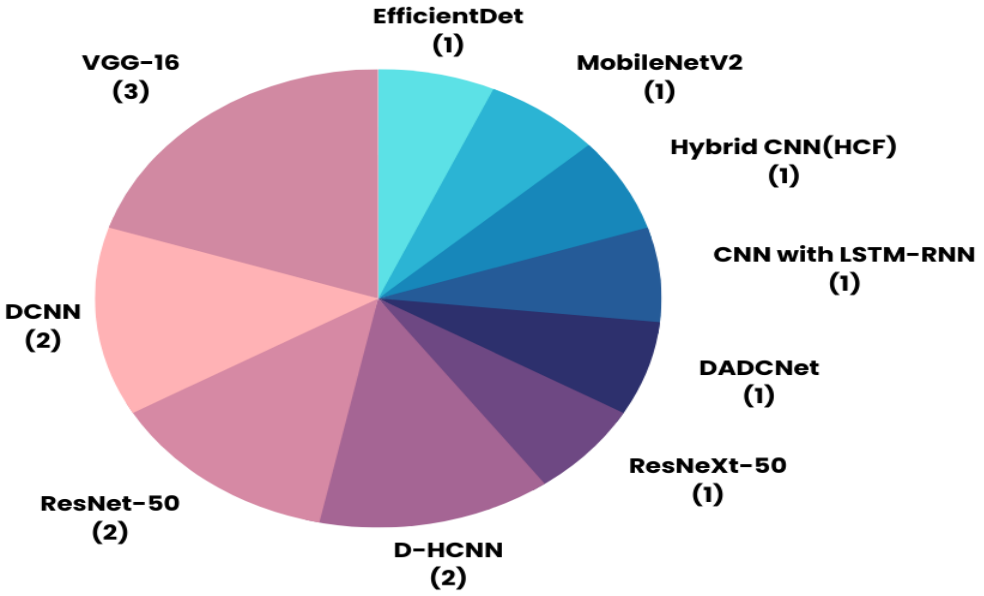


Figure 7: Frequency of different models used in a comparative study

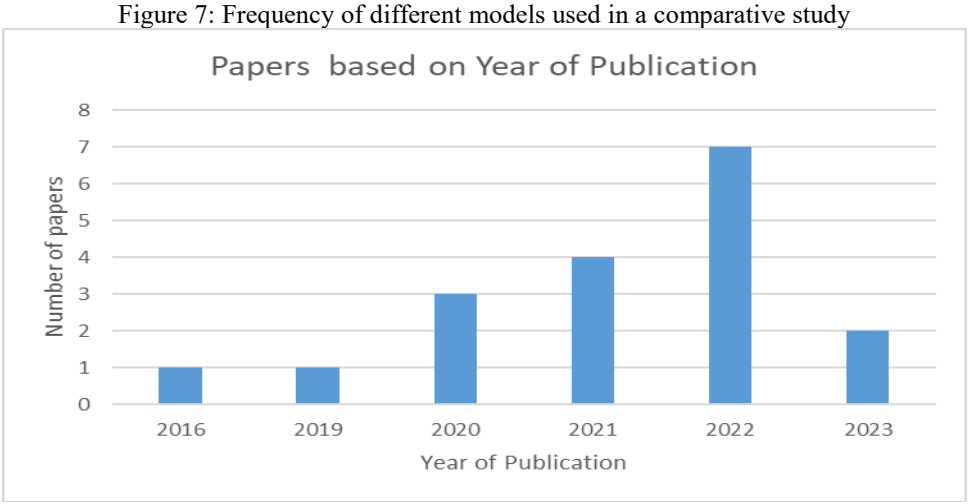


Figure 8: Details of paper used in a comparative study

5 Conclusion

This research employs the Xception model to efficiently predict a range of distracted driving activities, thereby mitigating the occurrence of severe accidents. The Xception model exhibited notable performance with an accuracy rate of 85.58%, surpassing counterparts like ResNet and VGG16 that achieved accuracy levels exceeding 90%. However, the true strength of the Xception model surface during real-time detection scenarios. Despite its slightly lower accuracy than ResNet and VGG16, the Xception model demonstrated superior performance in practical applications. This suggests that ResNet and VGG16 encountered challenges, potentially due to their susceptibility to overfitting. This shows a divergence in performance could potentially be attributed to Xception Model's capability to capture intricate features that are crucial for accurate predictions in dynamic, real-time environments.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Building upon this research, our current focus is on enhancing accuracy while minimizing false predictions. Introducing a more comprehensive dataset holds promise for reducing misclassifications, a critical aspect given the model's sensitivity to varying lighting conditions; it tends to struggle under low light circumstances. Unfortunately, the challenge lies in limited resources and data availability, impeding seamless implementation. Despite these hurdles, we remain optimistic about crafting and deploying an advanced, fully functional prediction model, delivering heightened accuracy and reliability.

References

[1] Sajid, Faiqa, Abdul Rehman Javed, Asma Basharat, Natalia Kryvinska, Adil Afzal, and Muhammad Rizwan. "An efficient deep learning framework for distracted driver detection." IEEE Access 9 (2021). doi**:** 10.1109/ACCESS.2021.3138137

[2] Anna Montoya, Dan Holman, SF data science, Taylor Smith, Wendy Kan. (2016). State Farm Distracted Driver Detection. Kaggle. https://kaggle.com/competitions/state-farm-distracted-driver-detection.

[3] Qin, Binbin, Jiangbo Qian, Yu Xin, Baisong Liu, and Yihong Dong. "Distracted driver detection based on a CNN with decreasing filter size." IEEE Transactions on Intelligent Transportation Systems 23, no. 7 (2021). doi: 10.1109/TITS.2021.3063521

[4] Liu, Dichao, Toshihiko Yamasaki, Yu Wang, Kenji Mase, and Jien Kato. "Toward Extremely Lightweight Distracted Driver Recognition With Distillation-Based Neural Architecture Search and Knowledge Transfer." IEEE Transactions on Intelligent Transportation Systems (2022). doi**:** 10.1109/TITS.2022.3217342

[5] Md. Uzzol Hossain, Md. Ataur Rahman, Md. Manowarul Islam, Arnisha Akhter, Md. Ashraf Uddin, Bikash Kumar Paul, "Automatic driver distraction detection using deep convolutional neural networks", Intelligent Systems with Applications, Volume 14, 2022, doi: https://doi.org/10.1016/j.iswa.2022.200075.

[6] J. D. Ortega, N. Kose, P. Cañas, M. A. Chao, A. Unnervik, M. Nieto, and L. Salgado, ''DMD: A large-scale multi-modal driver monitoring dataset for attention and alertness analysis,'' in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, Aug. 2020. doi: https://doi.org/10.48550/arXiv.2008.12085

[7] Ezzouhri, Amal, Zakaria Charouh, Mounir Ghogho, and Zouhair Guennoun. "Robust deep learning-based driver distraction detection and classification." IEEE Access 9 (2021). doi**:** 10.1109/ACCESS.2021.3133797

[8] Huang, Chen, Xiaochen Wang, Jiannong Cao, Shihui Wang, and Yan Zhang. "HCF: A hybrid CNN framework for behavior detection of distracted drivers." IEEE access 8 (2020). doi**:** 10.1109/ACCESS.2020.3001159

[9] Omerustaoglu, Furkan, C. Okan Sakar, and Gorkem Kar. "Distracted driver detection by combining in-vehicle and image data using deep learning." Applied Soft Computing 96 (2020). https://doi.org/10.1016/j.asoc.2020.106657

[10] P. -W. Lin and C. -M. Hsu, "Innovative Framework for Distracted-Driving Alert System Based on Deep Learning," in *IEEE Access*, vol. 10, pp. 77523-77536, 2022, doi: 10.1109/ACCESS.2022.3186674.

[11] L. Su, C. Sun, D. Cao and A. Khajepour, "Efficient Driver Anomaly Detection via Conditional Temporal Proposal and Classification Network," in *IEEE Transactions on Computational Social Systems*, vol. 10, no. 2, pp. 736-745, April 2023, doi: 10.1109/TCSS.2022.3158480.

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

[13]Tammina, Srikanth. "Transfer learning using vgg-16 with deep convolutional neural network for classifying images." International Journal of Scientific and Research Publications (IJSRP) 9, no. 10 (2019). doi: 10.29322/IJSRP.9.10.2019.p9420

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

[15] B. Wagner, F. Taffner, S. Karaca and L. Karge, "Vision Based Detection of Driver Cell Phone Usage and Food Consumption," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4257-4266, May 2022, doi: 10.1109/TITS.2020.3043145.

[16] Aljasim, M.; Kashef, R. E2DR: A Deep Learning Ensemble-Based Driver Distraction Detection with Recommendations Model. *Sensors* 2022, *22*, 1858. https://doi.org/10.3390/s22051858

[17]Md. Uzzol Hossain, Md. Ataur Rahman, Md. Manowarul Islam, Arnisha Akhter, Md. Ashraf Uddin, Bikash Kumar Paul, Automatic driver distraction detection using deep convolutional neural networks, Intelligent Systems with Applications, Volume 14, 2022, 200075, ISSN 2667-3053, https://doi.org/10.1016/j.iswa.2022.200075
[18] Koay, H.V.; Chuah, J.H.; Chow, C.-O.; Chang, Y.-L.; Rudrusamy, B. Optimally-Weighted Image-Pose Approach (OWIPA) for Distracted Driver Detection and Classification. *Sensors* 2021, *21*, 4837. https://doi.org/10.3390/s21144837
[19] Mallikarjun Anandhalli and Vishwanath P. Baligar (2017)"An Approach to Detect Vehicles in Multiple Climatic Conditions Using the Corner Point Approach. Journal of Intelligent Systems, 27(3), pp. 363-376.
[20] Mallikarjun Anandhalli,Pavana Baligar, Vishwanath P. Baligar and Tanuja A "Corner Based Statistical Modelling in Vehicle Detection Under Various Condition for Traffic Surveillance", in Multimedia Tools and Applications,81, 28849–28874 (2022), https://doi.org/10.1007/s11042-022-12422-0.
[21] Mallikarjun Anandhalli,Pavana Baligar, Santosh S. Saraf and Pooja Deepsir" Image projection method for vehicle speed estimation model in video system", Machine Vision and Applications, Vol. 33, Iss. 7. DOI: https://doi.org/10.1007/s00138-021-01255-w

ISSN: 1137-3601 (print), 1988-3064 (on-line)