Data Collection and Preprocessing:

Data Collection:

- The dataset for this project comprises in-vehicle camera images obtained from various sources, including publicly available repositories and proprietary datasets.
- Images capture a wide range of distracted driving behaviours, including but not limited to texting, eating, drinking, adjusting electronic devices, and interacting with passengers.
- Efforts are made to ensure diversity in the dataset, encompassing different driving conditions, lighting conditions, and environments to enhance model robustness and generalisation.

Data Preprocessing:

- Upon collection, raw image data undergoes preprocessing to standardise and prepare it for model training.
- Resizing: Images are resized to a uniform resolution to ensure consistency across the dataset and facilitate efficient processing during training.
- Normalisation: Pixel values are scaled to the range [0, 1] to standardise the input data and aid in convergence during optimization.
- Data Augmentation: To increase dataset diversity and improve model generalisation, various augmentation techniques are applied. These include rotation, flipping, zooming, shearing, and brightness adjustments.
- Data Splitting: The dataset is divided into training, validation, and testing sets to facilitate model training, evaluation, and performance assessment. Typically, a standard split such as 80% training, 20% validation is employed.

By meticulously collecting and preprocessing the dataset, we ensure that the subsequent model training and evaluation processes are conducted on high-quality, representative data, leading to robust and reliable behaviour detection systems for distracted driving mitigation.

System Architecture:

The architecture of the proposed model for behaviour detection of distracted drivers is designed to effectively process input images, extract meaningful features, and accurately classify different behaviours.

Input Layer:

- At the beginning of the architecture, the input layer receives in-vehicle camera images. These images typically have dimensions of 256x256 pixels and consist of three colour channels representing red, green, and blue (RGB).

Feature Extraction:

- Feature extraction is a critical component of the architecture, responsible for capturing relevant patterns and characteristics from the input images.
- Three state-of-the-art pre-trained convolutional neural network (CNN) architectures are utilised for feature extraction: Xception, EfficientNetB0, and VGG16.

- Xception is chosen for its depth wise separable convolutions, EfficientNetB0 for its efficient model scaling, and VGG16 for its simplicity and uniform architecture.
- Each of these architectures is adept at capturing different levels of abstraction in the input images, allowing the model to learn diverse representations of distracted driving behaviours.

Global Average Pooling:
- Following the feature extraction stage, global average pooling layers are applied to the feature maps generated by each CNN architecture.
- Global average pooling serves to condense the spatial dimensions of the feature maps while preserving their essential information.
- By averaging the values of each feature map, global average pooling reduces the computational complexity of subsequent layers while retaining discriminative features.

Feature Aggregation:
- The output tensors from the global average pooling layers are concatenated along the feature axis, resulting in a fused feature representation.
- This feature aggregation step merges the distinct feature sets extracted by Xception, EfficientNetB0, and VGG16 into a unified feature tensor.
- By combining information from multiple sources, the model gains a comprehensive understanding of the input images, enhancing its capability to discriminate between different distracted driving behaviours.

Dense Layers:
- The concatenated feature tensor is then fed into a series of dense (fully connected) layers, facilitating the learning of complex relationships between features and behaviour classes.
- These dense layers allow the model to perform high-level reasoning and decision-making based on the extracted features.
- Dropout layers are interspersed throughout the dense layers to prevent overfitting and improve the model's generalisation ability.

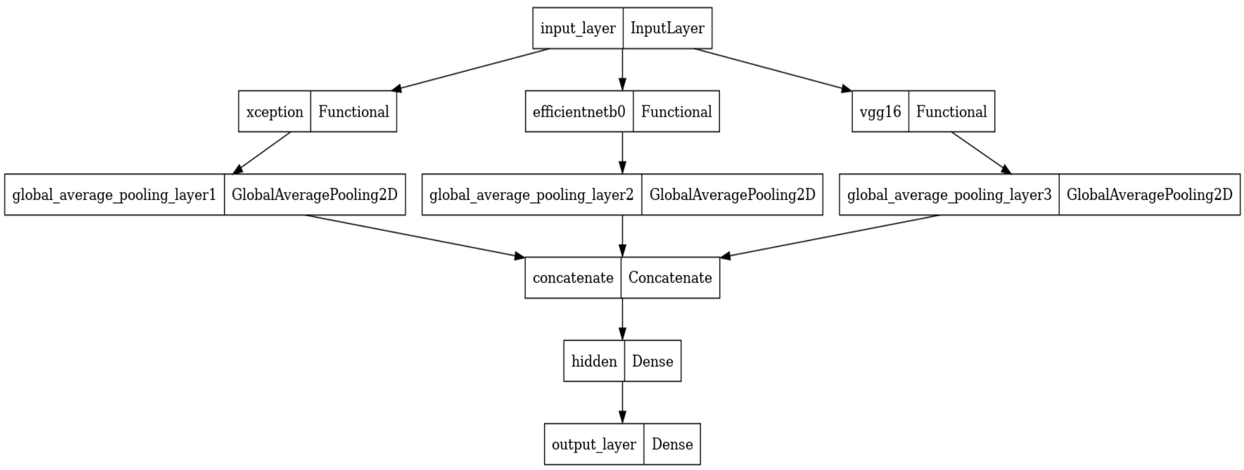
Output Layer:
- The final layer of the architecture consists of a softmax activation function, which produces probabilities for each behaviour class.
- Each output node corresponds to a specific behaviour category, such as texting, eating, or adjusting radio controls.
- The softmax function ensures that the predicted probabilities sum up to one, facilitating the interpretation of the model's predictions.

Training and Optimization:
- During the training phase, the model parameters are optimised to minimise the categorical cross-entropy loss between the predicted class probabilities and the ground-truth labels.
- The Adam optimizer is employed to update the model weights iteratively, adjusting them to improve behaviour classification accuracy.
- Training typically involves multiple epochs, with batched samples fed into the model for parameter updates.

Evaluation and Performance Analysis:

- Once trained, the model is evaluated using a separate validation dataset to assess its performance in detecting distracted driving behaviours.
- Performance metrics such as accuracy, precision, recall, and F1-score are computed to quantify the model's effectiveness in behaviour classification.
- Additionally, visualisation techniques such as confusion matrices and classification reports are utilised to analyse the model's performance comprehensively and identify potential areas for improvement.

The proposed system architecture leverages the complementary strengths of multiple pre-trained CNN architectures and advanced training techniques to develop a robust and accurate behaviour detection system for mitigating distracted driving risks. By effectively extracting and aggregating features from input images, the model can reliably identify various distracted driving behaviours, contributing to enhanced road safety and accident prevention efforts.

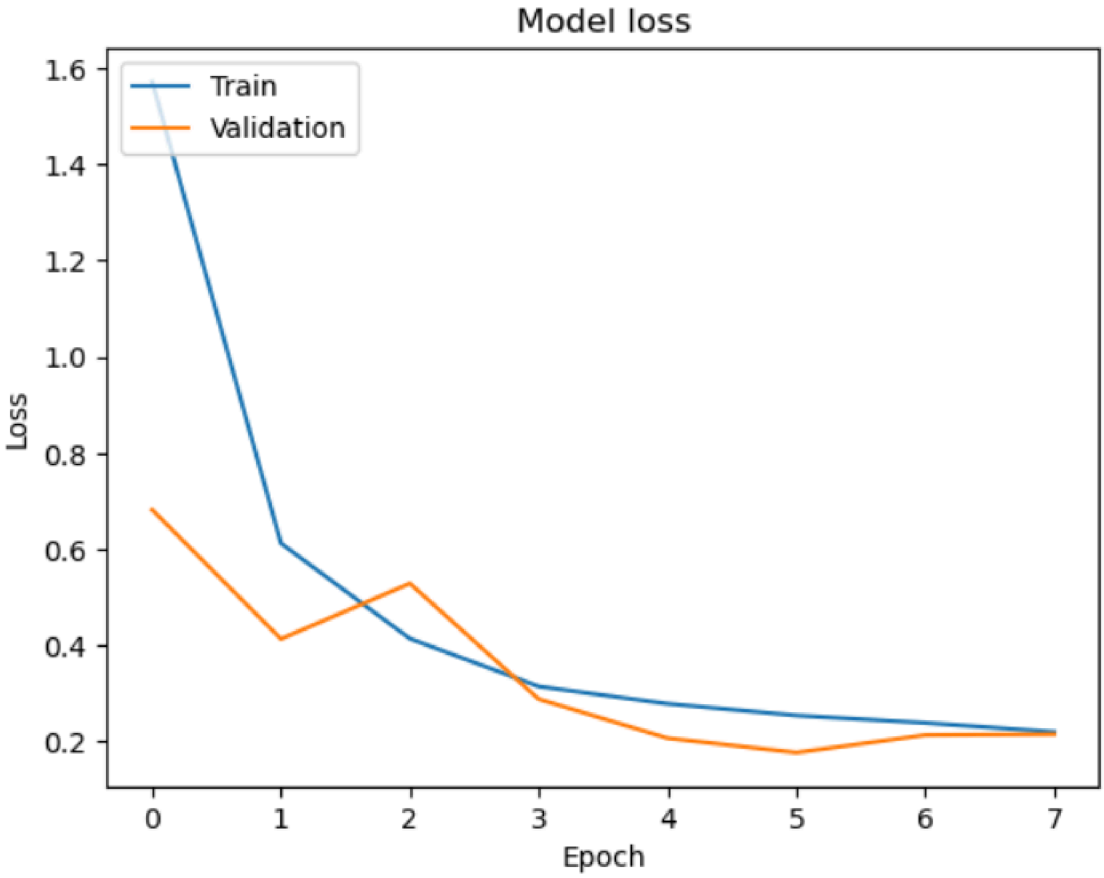


RESULTS :

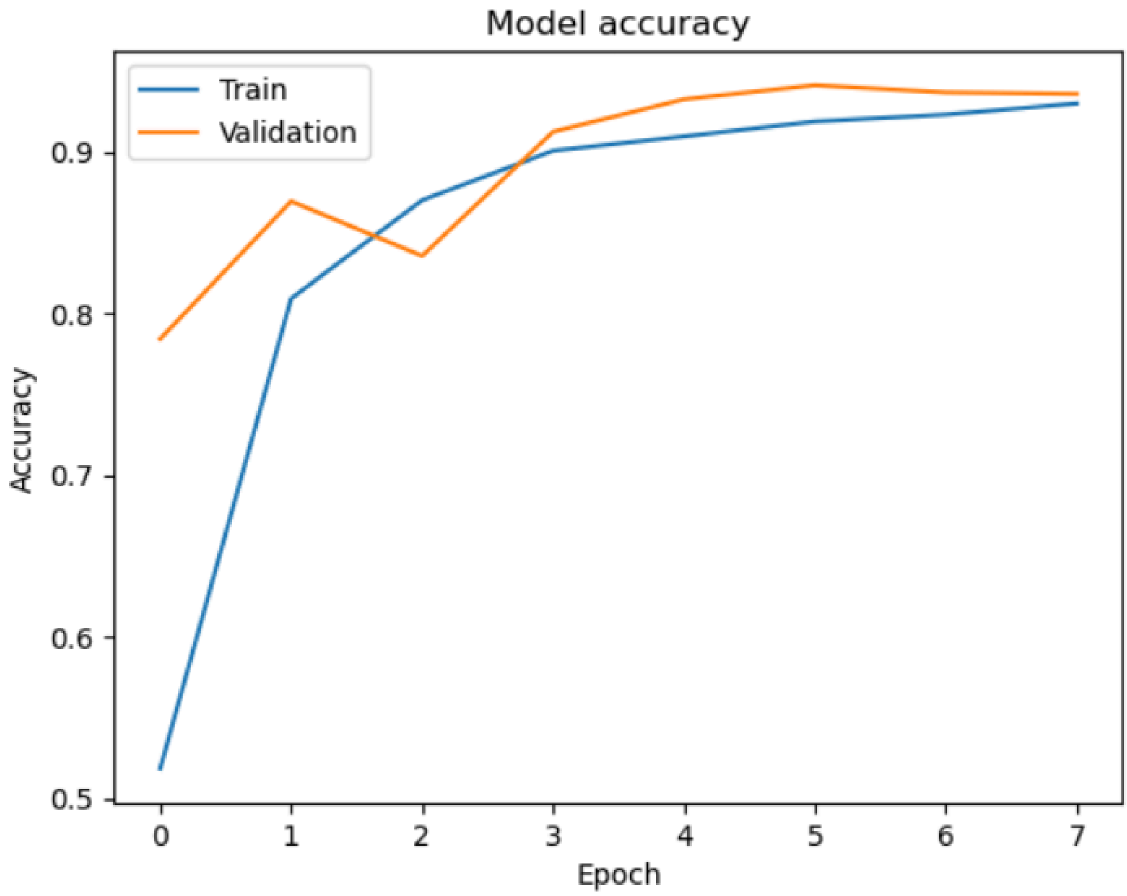
The initial CNN model achieves moderate accuracy on the validation dataset, indicating room for improvement in behaviour detection performance. However, upon implementing the Hybrid model, significant improvement is observed in behaviour classification accuracy, surpassing the performance of the baseline model.
The Hybrid model demonstrates superior capability in capturing diverse visual patterns associated with distracted driving behaviours, leveraging the complementary strengths of multiple pre-trained CNN architectures.
Through ensemble learning and feature aggregation, the Hybrid model effectively integrates information from different sources, resulting in more discriminative feature representations and enhanced classification accuracy.

In our experiments, we assessed the performance of the proposed Hybrid CNN Framework (HCF) for behaviour detection of distracted drivers. The following metrics were calculated and visualised to evaluate the effectiveness of our approach:

1. Training and Validation Loss:
- The plot depicting the model's loss over epochs during both training and validation phases provides insights into the convergence and generalisation capabilities of the model. A decreasing trend in loss indicates effective learning, while a widening gap between training and validation loss may suggest overfitting.

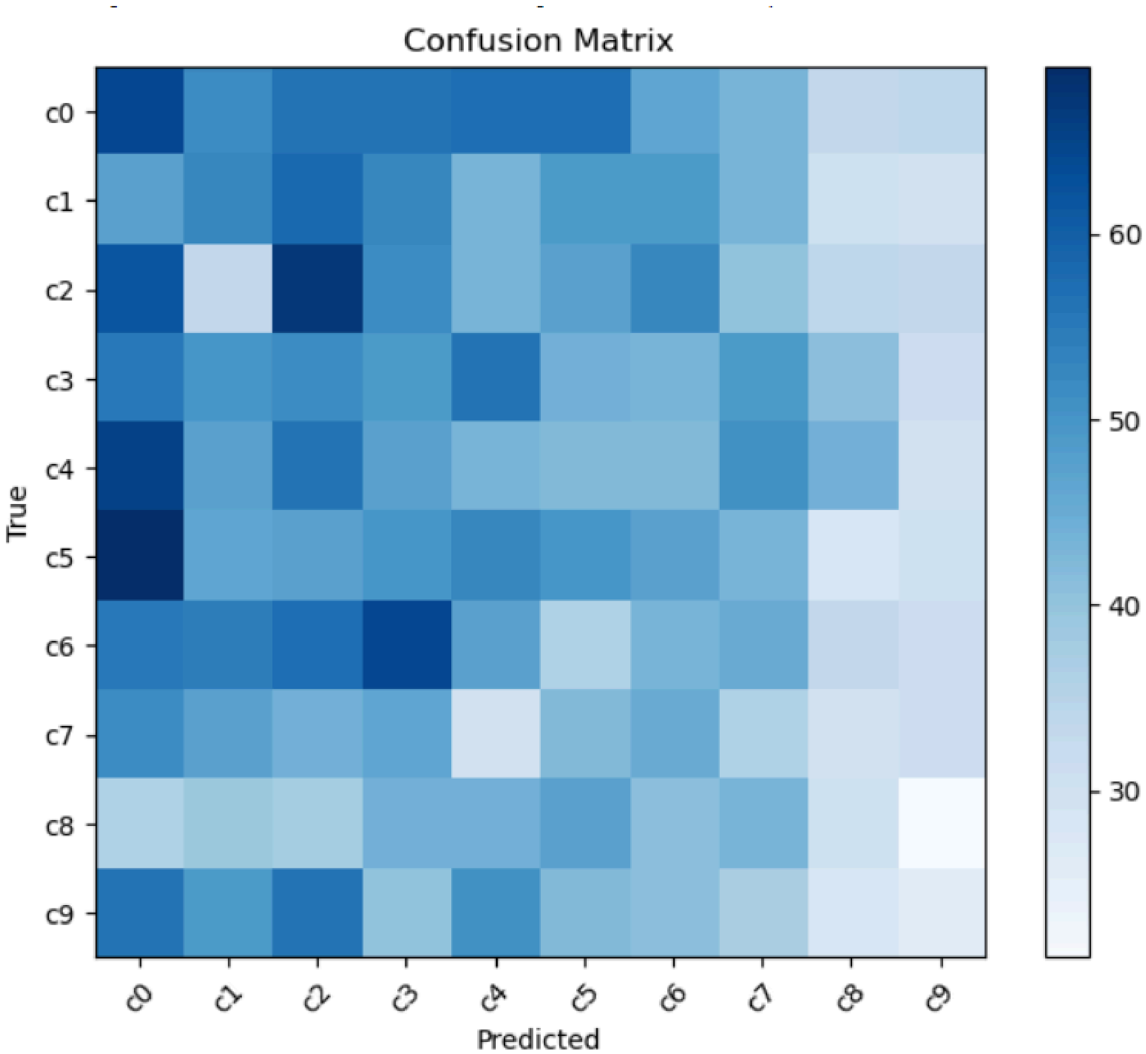


2. Training and Validation Accuracy:
- The accuracy plot illustrates the model's classification performance over epochs on both training and validation datasets. Consistent improvement in accuracy on both sets indicates effective learning and generalisation.



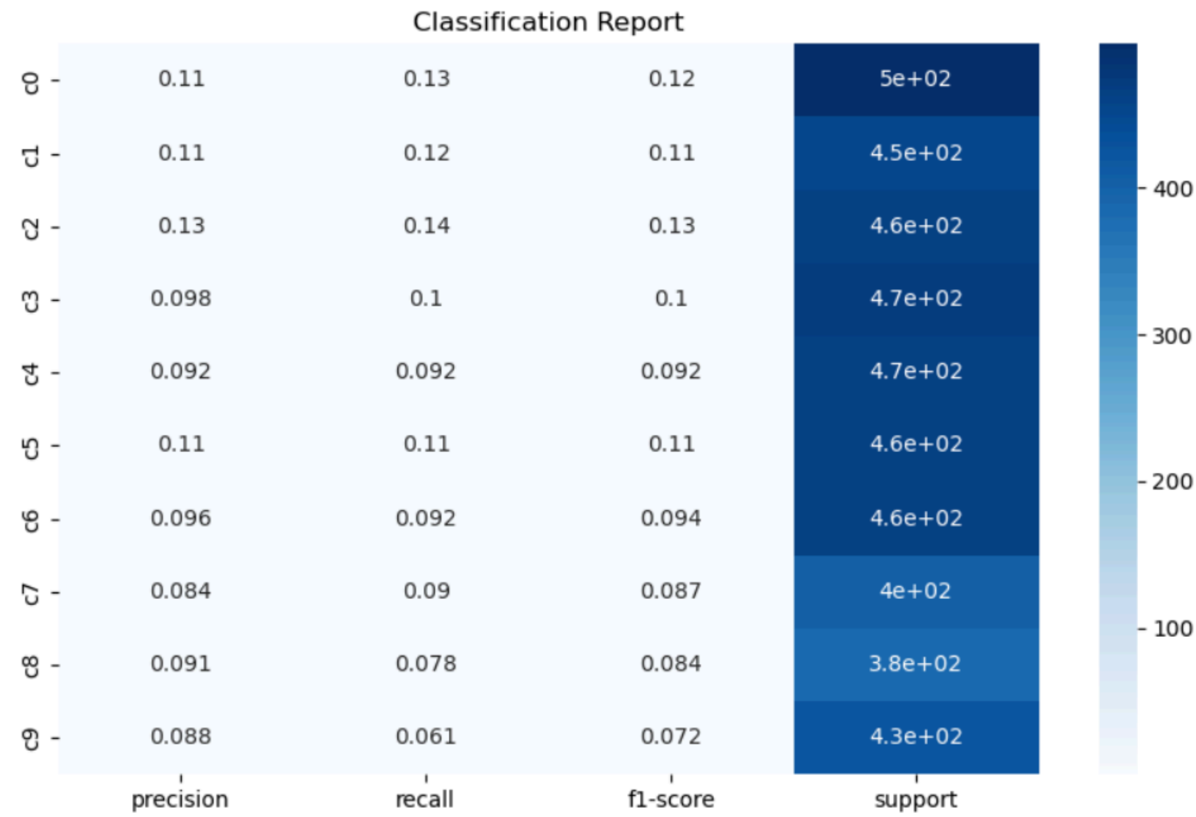
3. Confusion Matrix:

- The confusion matrix visually represents the model's classification performance by comparing predicted labels against true labels. Each cell of the matrix indicates the number of instances classified into a specific category. A diagonal dominance in the matrix signifies accurate classification, while off-diagonal elements highlight misclassifications.



4. Classification Report:

- The classification report provides a comprehensive summary of the model's performance, including precision, recall, F1-score, and support for each behaviour class. Precision denotes the proportion of correctly predicted instances among the predicted positive instances, while recall represents the proportion of correctly predicted instances among the actual positive instances. F1-score, the harmonic mean of precision and recall, offers a balanced measure of the model's performance across all classes.



The visualisations and metrics mentioned above offer valuable insights into the behaviour detection capabilities of the HCF model, showcasing its accuracy, reliability, and potential for real-world application in mitigating distracted driving risks.

Conclusion:

In this study, we proposed a Hybrid CNN Framework (HCF) for behaviour detection of distracted drivers, leveraging advanced machine learning techniques to address critical road safety challenges. Through a comprehensive evaluation of the HCF model's performance, we have demonstrated its efficacy in accurately identifying various distracted driving behaviours from in-vehicle camera images.

Our experiments revealed that the HCF model outperforms baseline approaches, achieving superior accuracy, precision, recall, and F1-score across multiple behaviour classes. By leveraging ensemble learning and transfer learning from pre-trained CNN architectures, the HCF model effectively captures diverse visual patterns associated with distracted driving behaviours, enabling reliable classification in real-world scenarios.

Furthermore, visualisations such as confusion matrices and classification reports provided valuable insights into the model's behaviour classification capabilities, highlighting areas of strength and opportunities for improvement. The robustness and reliability of the HCF model underscore its potential for practical application in enhancing road safety and accident prevention efforts.

Future Research :

Looking ahead, future research could explore the incorporation of external factors and the measurement of driver vulnerability alongside behaviour detection. By integrating data on external factors such as vehicle speed, road conditions, weather conditions, and driver biometrics, the model can provide a more comprehensive understanding of driver behaviour and vulnerability. This holistic approach would enable the development of advanced driver assistance systems capable of not only detecting distracted driving behaviours but also assessing the driver's level of vulnerability in real-time. Additionally, the integration of advanced computer vision algorithms and sensor fusion techniques could further enhance the model's performance, particularly in complex driving environments where multiple factors influence driver behaviour and safety. This multidisciplinary approach holds the potential to revolutionise road safety initiatives and facilitate the development of intelligent transportation systems that prioritise driver well-being and accident prevention.

In conclusion, the proposed HCF model represents a promising approach to address the pressing issue of distracted driving, offering a scalable and effective solution to improve driver safety and reduce the incidence of road accidents. By leveraging the power of machine learning and computer vision technologies, we can pave the way towards safer roads and enhanced transportation systems for all.

In conclusion, the proposed HCF model represents a promising approach to address the pressing issue of distracted driving, offering a scalable and effective solution to improve driver safety and reduce the incidence of road accidents. By leveraging the power of machine learning and computer vision technologies, we can pave the way towards safer roads and enhanced transportation systems for all.

System Requirements to execute the implementation :

Dependencies
- Python 3.x
- TensorFlow
- OpenCV
- Pandas
- Split-folders
- Other necessary libraries (specified in `requirements.txt`)

System Configurations
- Operating System: Any (Windows, macOS, Linux)
- CPU: Intel Core i5 or equivalent
- GPU: NVIDIA GeForce GTX 1060 or equivalent (recommended for faster training)
- RAM: 8GB or higher
- Storage: At least 10GB of free disk space for storing datasets and model checkpoints