Deep Learning Assignment Group 6

Dilara Saritas: 2145080, Evangelia Santorinaiou: 2134847, Giada Kruidenier:, Lisa Hamers: 2050279, Niya Neykova: 2147122, and Ruben Laout: 211198

Tilburg University

Deep Learning Spring 2025

Dr. Görkem Saygili and Rastislav Hronský

March 2nd, 2025

Deep Learning Assignment Group 6

Explicit Contributions

Dilara Saritas: Report

Evangelia Santorinaiou: Baseline model and plots, hyperparameter optimization, part of the discussion of improved models

Giada Kruidenier: Hyperparameter optimization, Methodology section, Hyperparameter tuning, Transfer learning, preprocessing of images

Lisa Hamers: Hyperparameter optimization, Methodology section, Introduction, Discussion

Niya Neykova: Hyperparameter optimization, Transfer Modeling, report APA

Ruben Laout: Report, ethical considerations, conclusion

Method

Baseline Model

The baseline model, consisting of 3 convolution layers with 8 filters each, followed by a simple dense structure and trained using the Adam optimizer, demonstrated moderate performance but showed signs of overfitting. With a training accuracy of 83.64% and a validation accuracy of 78.97%, it was evident that the model was struggling to generalize. Its shallow architecture limited feature extraction, while the lack of regularization made it overly sensitive to training data. Given that this project involves multiclass classification, we selected the *Softmax* activation function for the model's final layer. For the hidden layers the Rectified Linear Unit (*ReLu*) has been used. Participant information goes here.

Preprocessing experiments and their impact

To enhance the baseline model, we tested different preprocessing techniques, including class imbalance handling and data augmentation. For class imbalance, we applied class weight adjustments by modifying the loss function to assign higher weights to the minority class, ensuring better representation. However, this approach led to a decline in performance compared

to the baseline model, achieving a best validation accuracy of 0.4563. Similarly, data augmentation was implemented to artificially increase dataset size and diversity by applying transformations to existing images. Despite its potential benefits, the model performed worse, with a best validation accuracy of 0.5791. Given the negative impact of both techniques, neither was incorporated into the final model improvement process.

Improved model experiments

To address these issues, the *imporved_model_v*3 increased the number of filters (32 to 256) and added a fourth convolutional layer, allowing better hierarchical feature learning. The kernel size in the first convolution layer was increased to 5*x*5 to enhance feature extraction, while batch normalization was introduced after each layer to stabilize training. Additionally, dropout was applied to reduce overfitting and L2 regularization to prevent excessively large weight updates. Early stopping was also implemented, allowing training to stop automatically when validation loss ceased to improve. The number of epochs was increased from 10 in the baseline model to 30 in *imporved_model_v*3. Further experimentation with 40 epochs was conducted; however, the performance did not improve, suggesting that additional training did not provide further benefits. Finally, SGD with momentum (0.9) replaced *Adam*, as previous studies have shown that *SGD* often generalizes better in CNN-based image classification Gupta et al., 2021. These changes resulted in a significant performance boost, achieving a training accuracy of 92.3%, a validation accuracy of 90,9%,and a test accuracy of 91.37%. Finally, fewer misclassifications were made in critical categories, such as “Distracted” and “Safe Driving”, and the ROC curves demonstrated better class separability.

Best model experiments

Hyperparameter tuning was performed using *Optuna*. The tested hyperparameters included the number of filters (ranging from 8 to 32, with a step size of 8), kernel size ((3,3), (5,5)), dense units (ranging from 10 to 100, with a step size of 10), learning rate (1e-4 to 1e-2), optimizer choice (*Adam, RMSprop*), and activation function (*ReLU, Leaky ReLU, Tanh, Sigmoid*). The best hyperparameters obtained during tuning were: number of filters: 8, kernel size: (5,5),

dense units: 100, learning rate: 0.0012401486269841 86, optimizer: *RMSprop*, and activation function: *Leaky ReLU*. The obtained accuracy was 0.9044 on the test set and 0.8616 on the validation set. To refine the model further, a second hyperparameter search was conducted with an adjusted search space based on the previously obtained best parameters. This led to a configuration with 64 filters, a kernel size of (5,5), 30 dense units, a learning rate of 0.000319, *Adam optimizer,* and *ReLU activation*. Additionally, the number of convolutional layers was increased to four, batch size was set to 64, and dropout was adjusted to 0.219 to mitigate overfitting. Training this model for six epochs yielded a validation accuracy of 0.8845, which was promising. Extending the training to 30 epochs further improved validation accuracy to 0.9350. To address overfitting, *L2 regularization* was introduced, but this negatively impacted validation accuracy, even at low dropout and L2 values. As an alternative, dropout was increased to 0.25 while L2 regularization was removed. Finally, it was decided to increase the number of channels over the depth of the network. The initial layer was set to 32 filters, with the number of filters doubling at each subsequent layer. This was done because the width and height of the latent space decreases with every maxpooling layer, and this compensates by increasing the depth (i.e., the number of channels). This architectural change, combined with the tuned hyperparameters, resulted in the best validation accuracy of 0.9511 at epoch 27.

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 68, 124, 32)	832
batch_normalization_12 (BatchNormalization)	(None, 68, 124, 32)	128
max_pooling2d_6 (MaxPooling2D)	(None, 34, 62, 32)	0
dropout_15 (Dropout)	(None, 34, 62, 32)	0
conv2d_13 (Conv2D)	(None, 30, 58, 64)	51,264
batch_normalization_13 (BatchNormalization)	(None, 30, 58, 64)	256
dropout_16 (Dropout)	(None, 30, 58, 64)	0
conv2d_14 (Conv2D)	(None, 26, 54, 128)	204,928
batch_normalization_14 (BatchNormalization)	(None, 26, 54, 128)	512
max_pooling2d_7 (MaxPooling2D)	(None, 13, 27, 128)	0
dropout_17 (Dropout)	(None, 13, 27, 128)	0
conv2d_15 (Conv2D)	(None, 9, 23, 256)	819,456
batch_normalization_15 (BatchNormalization)	(None, 9, 23, 256)	1,024
flatten_3 (Flatten)	(None, 52992)	0
dropout_18 (Dropout)	(None, 52992)	0
dense_9 (Dense)	(None, 60)	3,179,580
dense_10 (Dense)	(None, 30)	1,830
dropout_19 (Dropout)	(None, 30)	0
dense_11 (Dense)	(None, 6)	186

Figure 1

Best Model Parameters

Transfer Learning

For the transfer learning model, the color mode was set to RGB, as this is the expected input format for most pretrained base models. To fine-tune DenseNet, an initial model was implemented with two dense layers and batch normalization to stabilize the training process and enhance generalization. This model achieved a validation accuracy of 0.7929, which was not optimal. To improve performance, an additional dense layer was introduced, and the size of the dense layers was increased. Despite these modifications, the validation accuracy remained inconsistent, indicating potential instability in the learning process. To address this, a learning rate scheduler was incorporated to gradually reduce the learning rate during training. The rationale behind this change was to minimize the risk of overshooting optimal parameter values, allowing the model to converge more effectively. This adjustment led to an improvement in validation accuracy, reaching 0.8309.

Results

To gain a better understanding of the dataset, we randomly selected 15 samples and displayed their corresponding labels directly on the images. Additionally, a bar plot was created to illustrate the distribution of class labels within the dataset. The results reveal a significant imbalance in class representation. This imbalance could impact model training, making it more difficult for the network to learn minority class features effectively. However, addressing this imbalance through data augmentation or class weighting did not improve overall model performance.

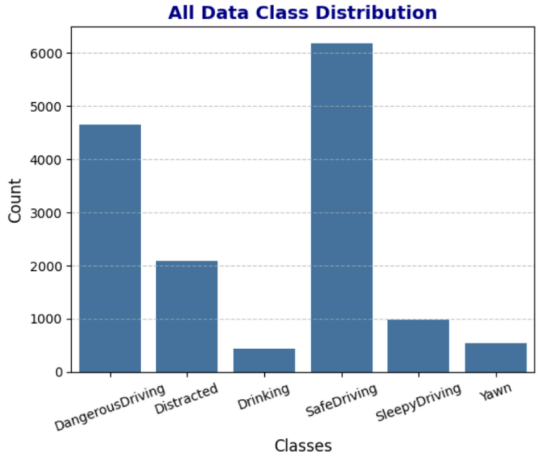

Figure 2

Class Distribution

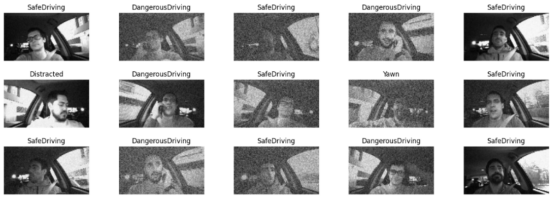

Figure 3

15 data samples

Baseline

The baseline model achieved an accuracy of 79% on the test set, with a precision of 0.71, recall of 0.65, and F1-score of 0.66. The ROC curves indicate that while the model performed reasonably well on some classes, it struggled particularly with 'Distracted' and 'Safe Driving,' where AUC scores were lower. The confusion matrix further highlights this, showing high misclassification rates for these categories, with 'Distracted' often being confused with 'Safe Driving' and 'Yawning' frequently misclassified.

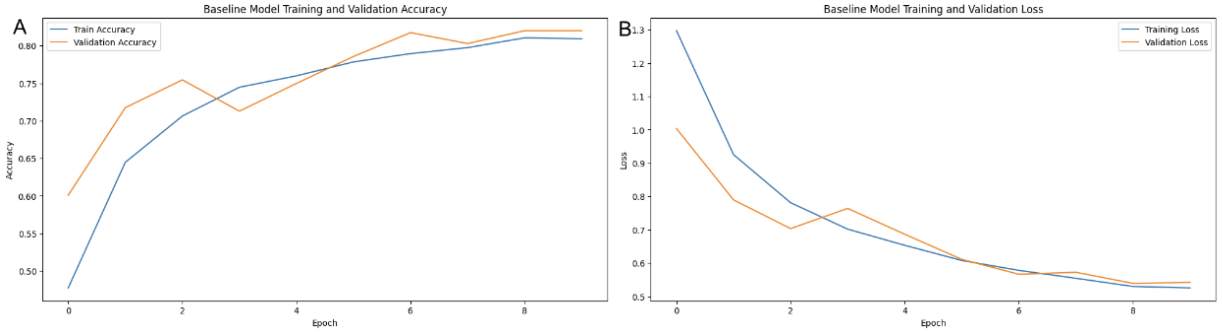

Figure 4

a): Baseline model training and validation accuracy b) Baseline model training and validation loss

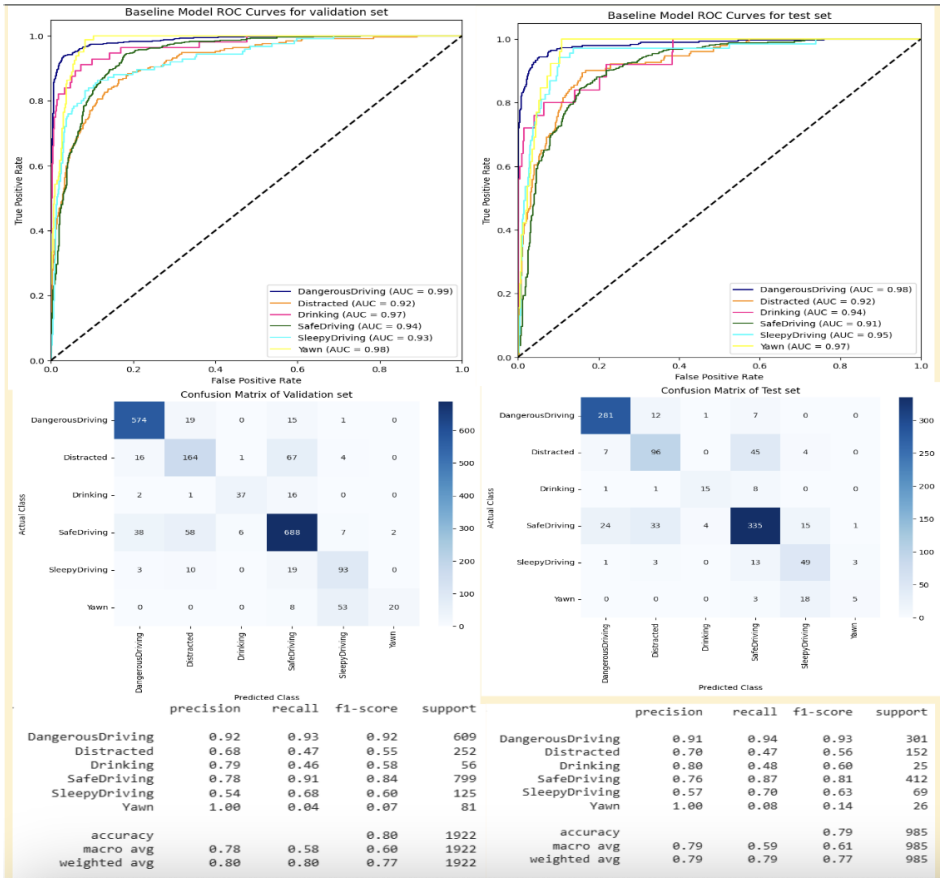

Figure 5

Baseline model classification results

Best model

Our best model demonstrated a significant improvement over the baseline in all performance metrics. The ROC curve comparison shows that the best model consistently achieved higher AUC scores across all classes. The confusion matrix further illustrates the reduction in misclassifications. The baseline model had frequent misclassifications between the 'Distracted' and 'Safe Driving' categories. While these categories saw the most improvement in the best model, they remain the most commonly confused pair. The recall score for the baseline model was relatively low (0.65), indicating poor performance in minority classes . In contrast, the best model achieved a recall of 0.92, indicating a much stronger capability to correctly identify instances across all classes. The architectural enhancements, including additional convolutional layers, increased filter sizes in deeper layers, optimized dropout rates, and batch normalization, significantly contributed to improved feature extraction and reduced overfitting.

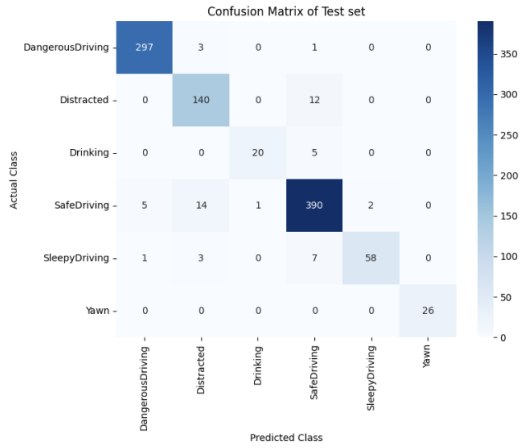

Figure 6

Best Model Confusion Matrix

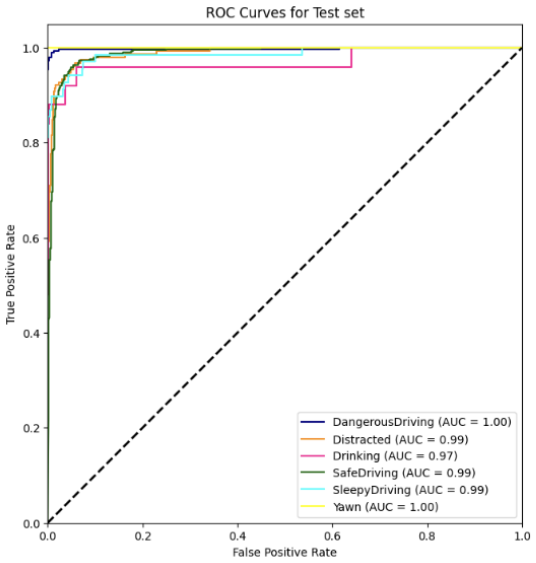

Figure 7

Test ROC

Test Set Classification Report:

	precision	recall	f1-score	support
DangerousDriving	0.98	0.99	0.98	301
Distracted	0.88	0.92	0.90	152
Drinking	0.95	0.80	0.87	25
SafeDriving	0.94	0.95	0.94	412
SleepyDriving	0.97	0.84	0.90	69
Yawn	1.00	1.00	1.00	26
accuracy			0.95	985
macro avg	0.95	0.92	0.93	985
weighted avg	0.95	0.95	0.94	985

Figure 8

Test set classification report

Transfer Model

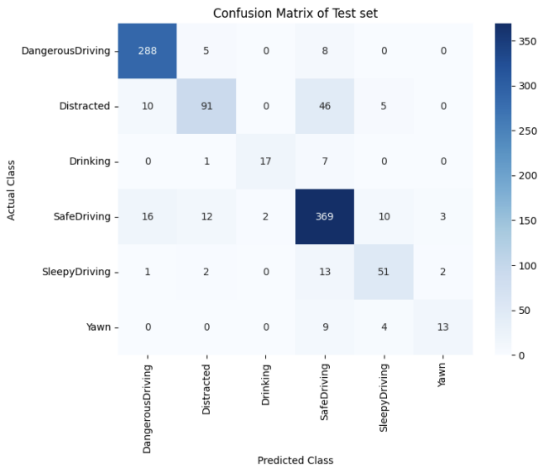

Figure 9

Transfer model confusion matrix

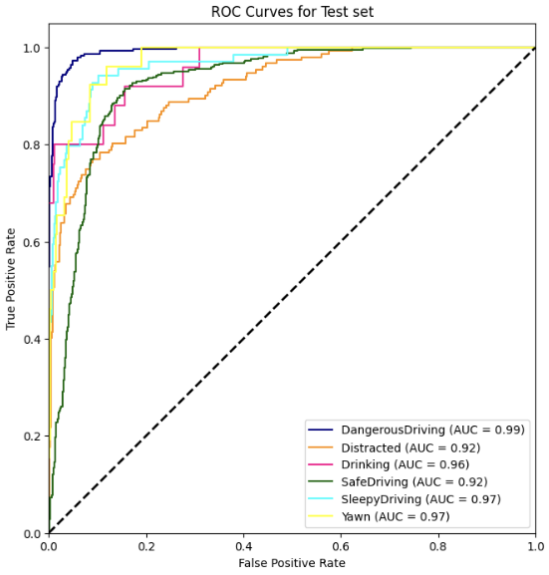

Figure 10

transfer model test ROC

Test Set Classification Report:

	precision	recall	f1-score	support
DangerousDriving	0.91	0.96	0.94	301
Distracted	0.82	0.60	0.69	152
Drinking	0.89	0.68	0.77	25
SafeDriving	0.82	0.90	0.85	412
SleepyDriving	0.73	0.74	0.73	69
Yawn	0.72	0.50	0.59	26
accuracy			0.84	985
macro avg	0.82	0.73	0.76	985
weighted avg	0.84	0.84	0.84	985

Figure 11

Transfer model test classification report

Results comparison

Discussion

Ethical Considerations

Bias in Driver Inattention Detection

Bias in deep learning models can arise from imbalanced training data, leading to inaccurate classifications for underrepresented groups. Research shows that biases can affect the accuracy of these systems across various demographics, such as age or physical conditions Barry, 2024. To mitigate these biases, it is essential to use diverse datasets, implement fairness-aware algorithms, and continuously evaluate models to ensure accurate and equitable detection for all drivers.

Privacy Concerns

Real-time monitoring via cameras and sensors raises privacy concerns, as it involves collecting sensitive driver data. Unauthorized access or misuse could lead to surveillance risks. Strict data protection measure such as anonymization, encryption, and compliance with regulations like GDPR are essential. Transparency and user consent mechanisms should also be prioritized.

Security Vulnerabilities

AI-driven monitoring systems may be vulnerable to cyber threats, including adversarial attacks that manipulate the system's classifications. Research by Ibrahum et al. (2025) emphasizes the effectiveness of adversarial training and edge computing in enhancing the robustness and security of AI systems.

Model Improvements

One possible improvement to the model's performance could be using a dataset with color (RGB) images instead of grayscale for driver inattention classification. With three channels (red, green, and blue) instead of just one, color images allow for richer feature extraction. Research shows that classification accuracy improves with higher-quality features Bansal et al., 2023. In addition to the dataset characteristics, previous research found that MobileNetV2 achieved the highest accuracy at 98.12%, outperforming simple CNNs, VGG16, and ResNet50 Hossain et al., 2022. Given these results, it could be worth trying to fit this algorithm to our dataset in future implementations. Additionally, using RGB images instead of grayscale could further reduce bias by preserving more visual details, such as skin tones and environmental lighting conditions. This can help improve accuracy, ensuring a more equal performance across different demographics.

For this project, we were not allowed to modify the pre-processing code, including image down-sampling. However, this could serve as a potential avenue for future improvements by adjusting image resolution. The key trade-off here is that larger input images provide more detailed features but require the network to process exponentially more pixels, significantly increasing the computational load Saponara and Elhanashi, 2021. We also attempted to address class imbalance by applying class weight adjustments during training, but this did not yield significant improvements. A potential future improvement could involve applying smoother class weight adjustments to fine-tune the model's learning process more effectively and improve the classification of underrepresented classes.

Another way to improve the transfer learning model's performance would be modifying the last layers of the convolutional network instead of only adjusting the fully connected layers.

Fine-tuning deeper layers would allow the model to adapt pre-trained features more effectively to the dataset's specific characteristics, improving classification accuracy.

Conclusion

In this project, we explored deep learning techniques for driver inattention classification, leveraging convolutional neural networks (CNNs) and transfer learning. Starting with a baseline model with a validation accuracy of 78.97%. We systematically improved performance through hyperparameter tuning, architectural modifications, and the application of advanced techniques such as batch normalization, dropout, and data augmentation. The best-performing model achieved a validation accuracy of 95.11%, demonstrating the effectiveness of these refinements.

Transfer learning experiments with DenseNet121 improved compared to the baseline model, but performed worse than our custom CNN model. This could be that the pre-trained features were not well-suited for the dataset, and freezing most convolutional layers limited the model's ability to adapt to the specific task.

Despite these advancements, challenges remain, including dataset biases, potential overfitting, and computational constraints. Future work could explore the use of RGB images for richer feature extraction, higher-resolution inputs, and alternative architectures like MobileNetV2, which has shown promising results in similar tasks.

References

Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2023). Transfer learning for image classification using vgg19: Caltech-101 image data set. *Journal of ambient intelligence and humanized computing*, 1–12.

Barry, K. (2024). How driver monitoring systems can protect drivers and their privacy. https://www.consumerreports.org/electronics/privacy/driver-monitoring-systems-can-protect-drivers-and-privacy-a7714760430/

Gupta, A., Ramanath, R., Shi, J., & Keerthi, S. S. (2021). Adam vs. sgd: Closing the generalization gap on image classification. *OPT2021: 13th Annual Workshop on Optimization for Machine Learning*, 1–7.

Hossain, M. U., Rahman, M. A., Islam, M. M., Akhter, A., Uddin, M. A., & Paul, B. K. (2022). Automatic driver distraction detection using deep convolutional neural networks. *Intelligent Systems with Applications*, *14*, 200075.

Ibrahum, A. D. M., Hussain, M., & Hong, J.-E. (2025). Deep learning adversarial attacks and defenses in autonomous vehicles: A systematic literature review from a safety perspective. *Artificial Intelligence Review*, *58*(1), 1–53.

Saponara, S., & Elhanashi, A. (2021). Impact of image resizing on deep learning detectors for training time and model performance. *International Conference on Applications in Electronics Pervading Industry, Environment and Society*, 10–17.