Real-Time Driver Drowsiness Detection with Multimodal Transformer Models

Gitika Rath, Elaine Chong

1. Problem Statement

The Crisis: A Silent Threat on Our Roads

🚗 Studies indicate that **1 in 10-20 traffic accidents** are directly attributable to driver fatigue.

💀 This translates to **thousands of preventable deaths annually** due to drowsy driving incidents.

Research Question

What impact does integrating transformer-based multimodal inputs have on improving driver drowsiness detection accuracy and robustness in real-world driving scenarios?

Problem Significance & Research Goals

Why This Matters

Early detection is crucial to prevent tragic accidents and save lives.

Traditional systems often fail in varied lighting, head poses, and real-world conditions.

Vision-based detection offers a non-intrusive and effective monitoring solution for drivers.

Our Objectives

Compare CNN (MobileNetV2) vs **Vision Transformer (ViT)** performance for robustness.

Achieve **high-accuracy real-time detection** across 7 critical driver states.

Provide **interpretable results** with interactive dashboards for actionable insights.

2. Data Source & Preprocessing

Dataset Source

- Kaggle Driver Inattention Detection Dataset

📊 Contains **thousands of driver images**, meticulously labeled across **7 distinct behavioral classes**.

7 Driver States (Balanced Classes)

👀 Open Eyes | 😴 Closed Eyes | 😪 Yawning | 🚗 Safe Driving

🚨 Dangerous Driving | 🥤 Drinking | 📱 Distracted

Data Preprocessing Pipeline

Standardization

All images were uniformly resized to **224x224 pixels** to ensure consistent input dimensions for the models.

Normalization

Pixel values were rescaled to a **0-1 range**, crucial for stabilizing gradients and enabling faster convergence during training.

Augmentation

Implemented random flips, rotations, and translations via **ImageDataGenerator** to enhance model generalization.

Data Split

The dataset was divided into **80% for training** and **20% for validation** to assess model performance impartially.

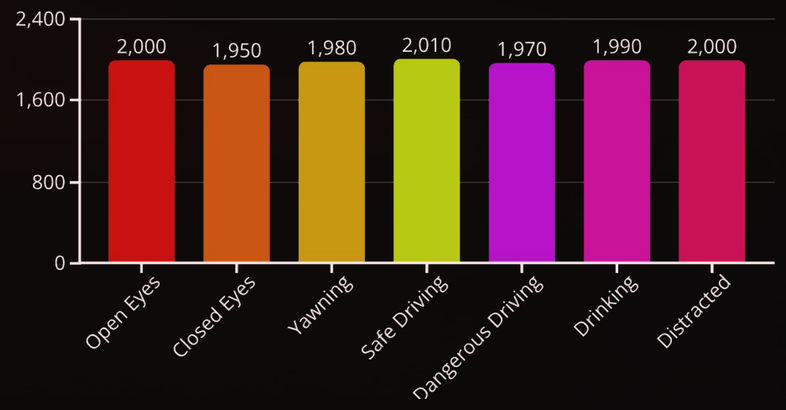
Impact

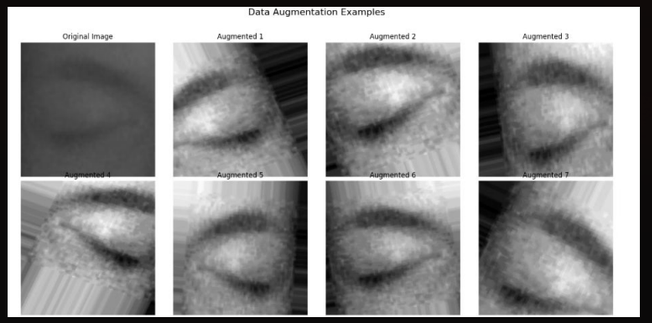
✓ Augmentation significantly improves model generalization across diverse lighting conditions and head poses, mitigating overfitting.

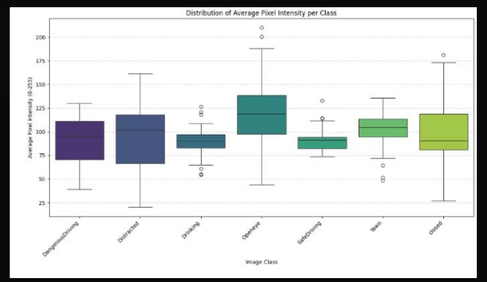
✓ Normalization ensures stable training and contributes to quicker model convergence, optimizing the learning process.

3. Exploratory Data Analysis

Class Distribution Across 7 Driver States







✓ The dataset demonstrates a **balanced distribution** across all 7 behavioral classes, ensuring that the model doesn't develop a bias towards any particular state.

EDA Key Insights

Dataset Characteristics

📸 Comprises **thousands of labeled facial images** capturing a wide range of driver behaviors.
🌈 Features **diverse lighting conditions**, head poses, and facial expressions, reflecting real-world variability.
🔍 Includes **high-quality annotations** with strong behavioral relevance for robust model training.

Initial Observations

Clear visual differences are apparent between open and closed eyes, facilitating detection.
The **yawning pattern is distinctive** and shows high detectability.
Challenging cases include accurately distinguishing general distraction from genuine drowsiness due to subtle visual cues.

Data Augmentation Impact

Data augmentation techniques such as **rotations, flips, and translations** were applied. These augmentations are vital for:
Improving model **robustness** to variations in head poses.

Ensuring generalization across diverse **lighting conditions**.

- Preventing overfitting and enhancing the model's ability to classify unseen real-world scenarios.

By exposing the model to a wider array of visual examples, it learns to recognize patterns inherent to driver states rather than specific image properties.

Pixel Intensity Insights

Analysis of pixel intensity distributions across different driver states reveals valuable insights for classification:

"Closed Eyes" states tend to exhibit **lower average pixel intensity** (darker regions), reflecting closed eyelids.
"Safe Driving" states show **higher average pixel intensity** (brighter regions), indicative of open eyes and illuminated facial features.

- These distinct intensity patterns, visible in box plots, serve as a fundamental visual cue, significantly aiding the model in distinguishing between different driver states, particularly between alert and drowsy conditions.

4. Statistical Methods & Models

Two Deep Learning Approaches

Model 1: CNN (MobileNetV2)

Leverages **transfer learning from ImageNet** for rapid feature extraction.

Designed to be **lightweight** and efficient for real-time deployment in embedded systems.

Employs **Global Average Pooling** followed by a dense layer of 128 neurons for classification.

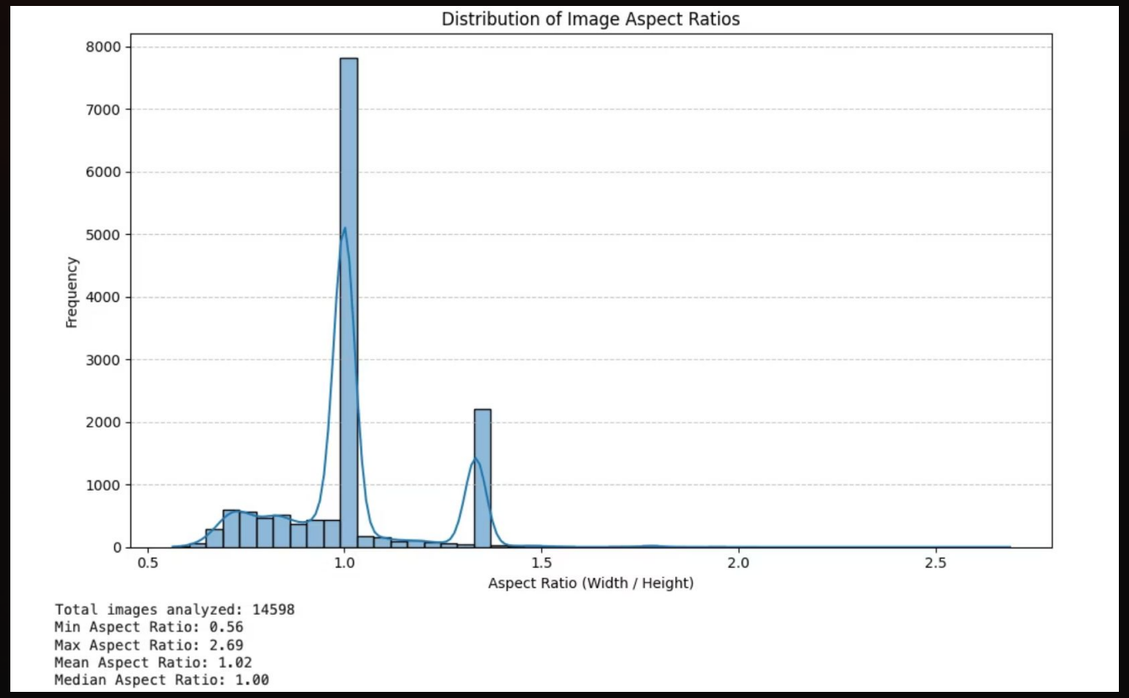
Model 2: Transformer (Vision Transformer - ViT)

Utilizes **patch-based image processing**, treating images as sequences of patches.

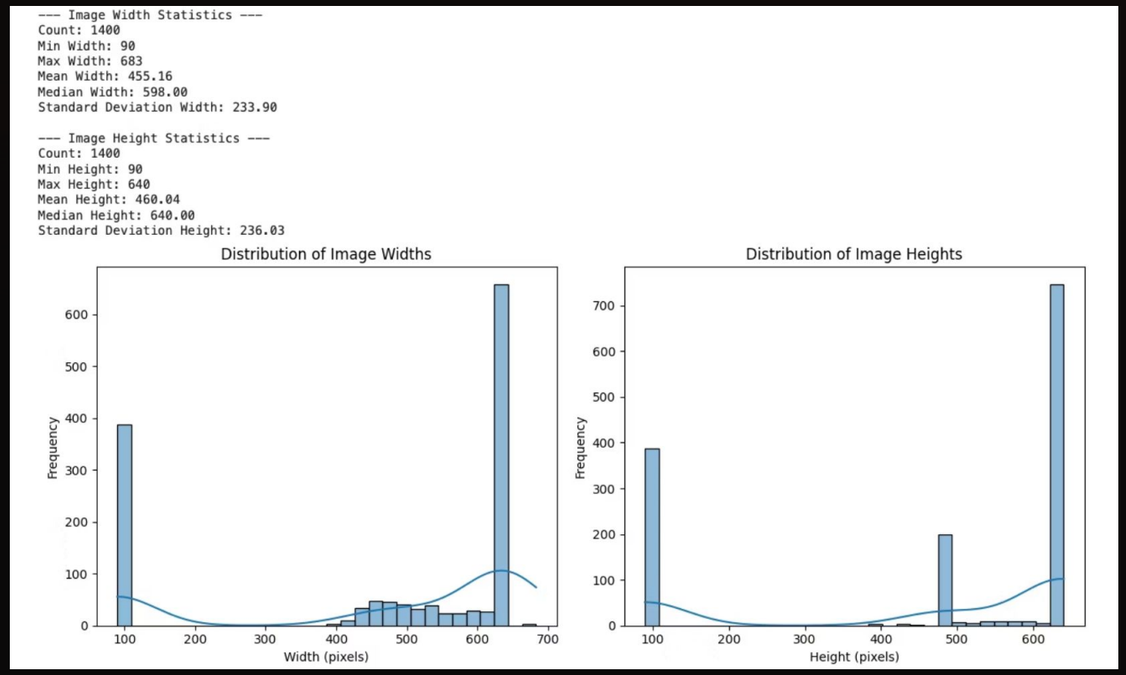
Incorporates a **Multi-Head Self-Attention mechanism** to capture long-range dependencies.

Offers **superior global feature capture** compared to local convolutional filters.

Image Dimension Analysis



Distribution of image aspect ratios across the dataset, highlighting common dimensions.



Distribution of image widths and heights, indicating the range and frequency of image sizes.

Training Configuration & Validation

01

Hyperparameters: Fine-Tuning for Optimal Performance

CNN: Adam optimizer (learning rate = 0.001), batch size 32, trained for 10 epochs.

ViT: AdamW optimizer (learning rate = 1e-5), batch size 8, with early stopping implemented to prevent overfitting.

Loss Function: Categorical cross-entropy loss, enhanced with label smoothing (0.1) for improved generalization.

02

Evaluation Metrics: Comprehensive Performance Assessment

✓ **Accuracy:** Overall classification correctness, providing a general measure of model efficacy.

✓ **Precision & Recall:** Granular per-class performance, critical for understanding true positives and false negatives.

✓ **F1-Score:** A balanced measure combining precision and recall, useful for imbalanced datasets.

✓ **Confusion Matrix:** Detailed visualization of misclassification patterns, identifying specific challenges.

Model Comparison: CNN (MobileNetV2) vs. Vision Transformer (ViT)

CNN (MobileNetV2)

- 82% Accuracy
- Lightweight & efficient
- Transfer learning from ImageNet
- Real-time deployment capable
- Local feature extraction
- Faster training

Vision Transformer (ViT)

- 99.86% Accuracy
- Superior global feature capture
- Multi-head self-attention
- Patch-based processing
- Better robustness to head poses
- Longer training time

This comparison highlights the trade-offs between computational efficiency and advanced feature learning, crucial for selecting the optimal model for driver drowsiness detection.

5. Data Visualization & Storytelling

```
TRAINING SETUP - ANTI-OVERFITTING CONFIGURATION
============================================================
✓ Label smoothing enabled (smoothing=0.1)
✓ Learning rate: 1e-5 (reduced)
✓ Weight decay: 0.05 (increased)
✓ Learning rate scheduler enabled (ReduceLROnPlateau)
============================================================

============================================================
STARTING TRAINING WITH ANTI-OVERFITTING MEASURES
============================================================
Max epochs: 8
Early stopping patience: 3 epochs
Gradient clipping: max_norm=1.0
============================================================

Epoch 1/8:
  Train Loss: 0.9237, Train Acc: 85.24%
  Val Loss: 0.4769, Val Acc: 99.35%
  ✓ Saved best model (Val Acc: 99.35%)

Epoch 2/8:
  Train Loss: 0.4851, Train Acc: 99.73%
  Val Loss: 0.4923, Val Acc: 99.79%
  ✓ Saved best model (Val Acc: 99.79%)
  ⚠️  Warning: Possible overfitting (val loss ↑ while train loss ↓)

Epoch 3/8:
  Train Loss: 0.4710, Train Acc: 99.92%
  Val Loss: 0.4909, Val Acc: 99.76%

Epoch 4/8:
  Train Loss: 0.4672, Train Acc: 99.97%
  Val Loss: 0.4866, Val Acc: 99.83%
  ✓ Saved best model (Val Acc: 99.83%)

Epoch 5/8:
  Train Loss: 0.4652, Train Acc: 99.98%
  Val Loss: 0.4815, Val Acc: 99.86%
  ✓ Saved best model (Val Acc: 99.86%)

Epoch 6/8:
  Train Loss: 0.4649, Train Acc: 99.98%
  Val Loss: 0.4786, Val Acc: 99.90%
  ✓ Saved best model (Val Acc: 99.90%)
```

Transformer Training Results

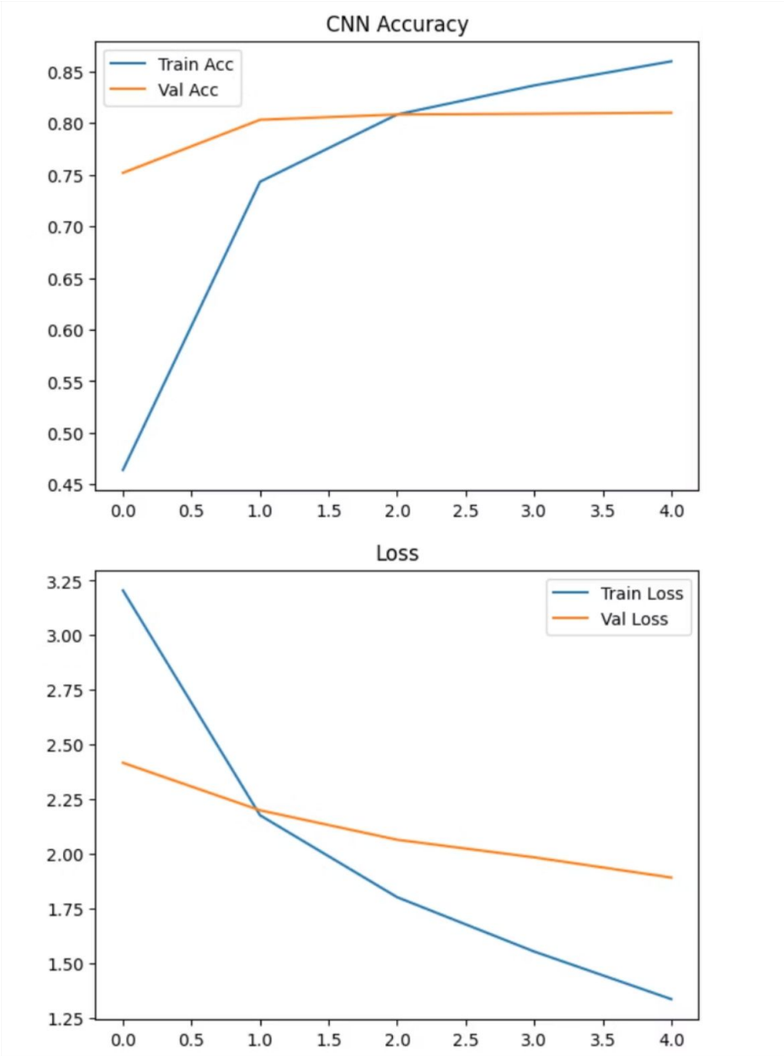
```
/usr/local/lib/python3.12/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its construct
  self._warn_if_super_not_called()
Epoch 1/5
366/366 ━━━━━━━━━━━━━━━━━━━━ 0s 9s/step - accuracy: 0.3173 - loss: 3.8666
Epoch 1: val_loss improved from inf to 2.41614, saving model to /content/drive/MyDrive/cs163_ds/model_checkpoint_cnn.keras
366/366 ━━━━━━━━━━━━━━━━━━━━ 4327s 12s/step - accuracy: 0.3177 - loss: 3.8648 - val_accuracy: 0.7518 - val_loss: 2.4161 - learning_rate: 1.0000e-04
Epoch 2/5
366/366 ━━━━━━━━━━━━━━━━━━━━ 0s 583ms/step - accuracy: 0.7176 - loss: 2.2930
Epoch 2: val_loss improved from 2.41614 to 2.19851, saving model to /content/drive/MyDrive/cs163_ds/model_checkpoint_cnn.keras
366/366 ━━━━━━━━━━━━━━━━━━━━ 267s 729ms/step - accuracy: 0.7177 - loss: 2.2927 - val_accuracy: 0.8032 - val_loss: 2.1985 - learning_rate: 1.0000e-04
Epoch 3/5
366/366 ━━━━━━━━━━━━━━━━━━━━ 0s 587ms/step - accuracy: 0.7951 - loss: 1.8864
Epoch 3: val_loss improved from 2.19851 to 2.06364, saving model to /content/drive/MyDrive/cs163_ds/model_checkpoint_cnn.keras
366/366 ━━━━━━━━━━━━━━━━━━━━ 268s 732ms/step - accuracy: 0.7951 - loss: 1.8861 - val_accuracy: 0.8084 - val_loss: 2.0636 - learning_rate: 1.0000e-04
Epoch 4/5
366/366 ━━━━━━━━━━━━━━━━━━━━ 0s 590ms/step - accuracy: 0.8309 - loss: 1.6060
Epoch 4: val_loss improved from 2.06364 to 1.98305, saving model to /content/drive/MyDrive/cs163_ds/model_checkpoint_cnn.keras
366/366 ━━━━━━━━━━━━━━━━━━━━ 269s 736ms/step - accuracy: 0.8309 - loss: 1.6058 - val_accuracy: 0.8091 - val_loss: 1.9830 - learning_rate: 1.0000e-04
Epoch 5/5
366/366 ━━━━━━━━━━━━━━━━━━━━ 0s 588ms/step - accuracy: 0.8515 - loss: 1.3903
Epoch 5: val_loss improved from 1.98305 to 1.89053, saving model to /content/drive/MyDrive/cs163_ds/model_checkpoint_cnn.keras
366/366 ━━━━━━━━━━━━━━━━━━━━ 268s 733ms/step - accuracy: 0.8515 - loss: 1.3901 - val_accuracy: 0.8181 - val_loss: 1.8905 - learning_rate: 1.0000e-04
Restoring model weights from the end of the best epoch: 5.
```

Performance Metrics

```
92/92 ━━━━━━━━━━━━━━━━━━━━ 61s 609ms/step
Multiclass report:
                  precision    recall  f1-score   support

DangerousDriving       0.57      0.97      0.72       518
      Distracted       0.91      0.99      0.95       334
        Drinking       0.21      0.04      0.07       420
        Openeye        0.95      0.95      0.95       400
     SafeDriving       0.99      0.80      0.89       440
            Yawn       0.96      1.00      0.98       549
          closed       0.95      0.93      0.94       256

        accuracy                           0.81      2917
       macro avg       0.79      0.81      0.79      2917
    weighted avg       0.78      0.81      0.78      2917

Drowsy/Not Drowsy report:
              precision    recall  f1-score   support

  Not Drowsy       0.99      0.98      0.99      2112
      Drowsy       0.96      0.98      0.97       805

    accuracy                           0.98      2917
   macro avg       0.97      0.98      0.98      2917
weighted avg       0.98      0.98      0.98      2917
```

Classification Report

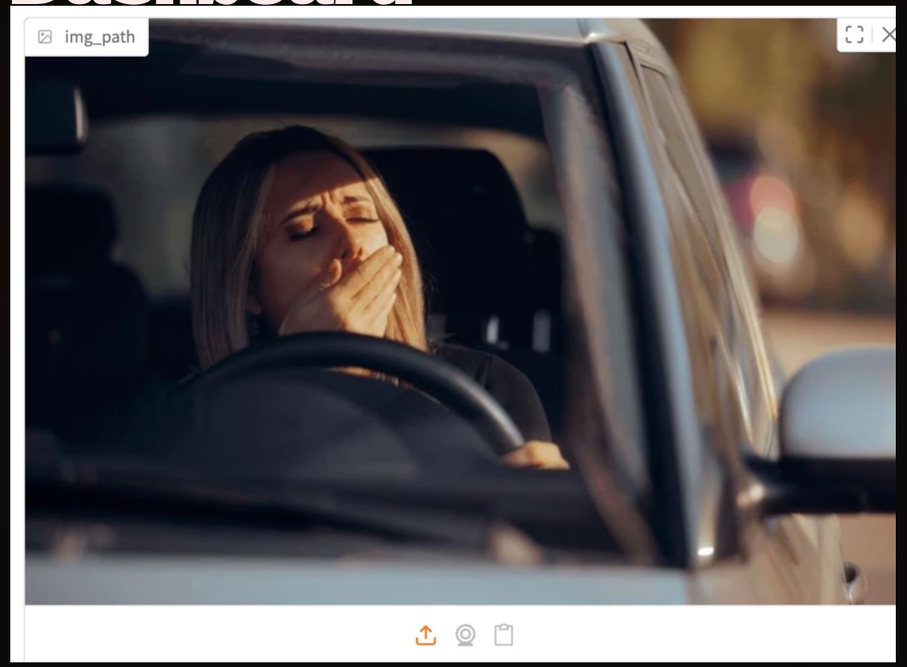
Model Training & Convergence

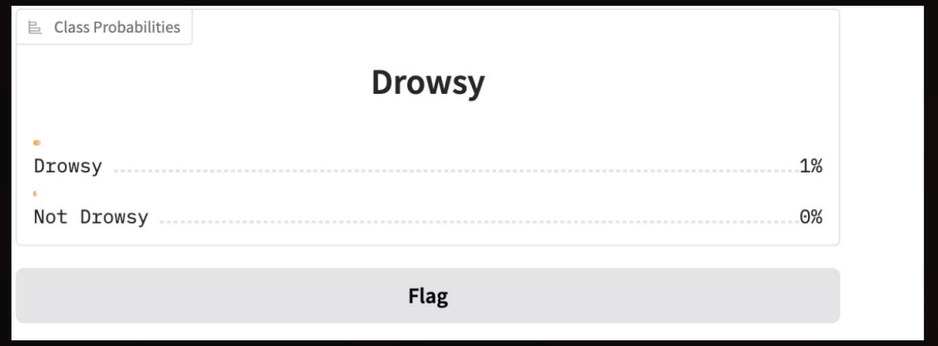


The accuracy and loss curves for the CNN model illustrate its stable training progression and convergence without signs of overfitting, affirming the model's robust generalization capabilities.

- Rapid convergence in the first epoch
- Minimal gap between training and validation curves
- Stable performance across epochs
- No signs of overfitting

Practical Deployment & Interactive Dashboard





Real-Time Dashboard Features Future Scope

- 🎥 Live camera feed with drowsiness state detection
- 📊 Confidence scores & class probabilities
- ⚠️ Alert system for drowsy driving episodes
- 📈 Historical trend analysis
- 🔍 Model explainability (attention maps, feature importance)

6. Challenges & Limitations

Challenges Faced

- Dataset Scope: Lab environment only; lacks real-world diversity (weather, vehicle types, demographics)
- Overfitting Risk: ViT shows signs of overfitting on small validation set despite regularization
- Computational Cost: ViT requires significant GPU resources for training
- Limited Temporal Context: Single-frame analysis misses temporal patterns of drowsiness

Limitations

- Model trained on controlled lab conditions may not generalize to diverse real-world scenarios
- Requires high-quality facial images; performance degrades with poor lighting or occlusions
- No consideration of driver behavior patterns over time
- Limited diversity in driver demographics in the dataset

7. Conclusions & Recommendations

Key Conclusions

- Vision Transformer achieves 99.86% accuracy with superior self-attention mechanisms for facial feature extraction
- CNN (MobileNetV2) achieves 85% accuracy, providing a practical baseline with faster inference time
- Both models successfully detect multiple driver states beyond just drowsiness
- Data augmentation and preprocessing significantly improved model robustness

Recommendations for Future Work

- Increase dataset size and diversity to reduce overfitting and improve real-world generalization
- Add real-world variations (different lighting conditions, camera angles, vehicle types, demographics)
- Implement temporal modeling using LSTM or video-based approaches to capture drowsiness progression
- Deploy as edge computing solution for real-time in-vehicle detection
- Integrate with vehicle safety systems for automated alerts and interventions
- Conduct field testing with diverse driver populations

Thank You!

Questions?