AUTO COM

"YOUR GUARDIAN ON THE ROAD"

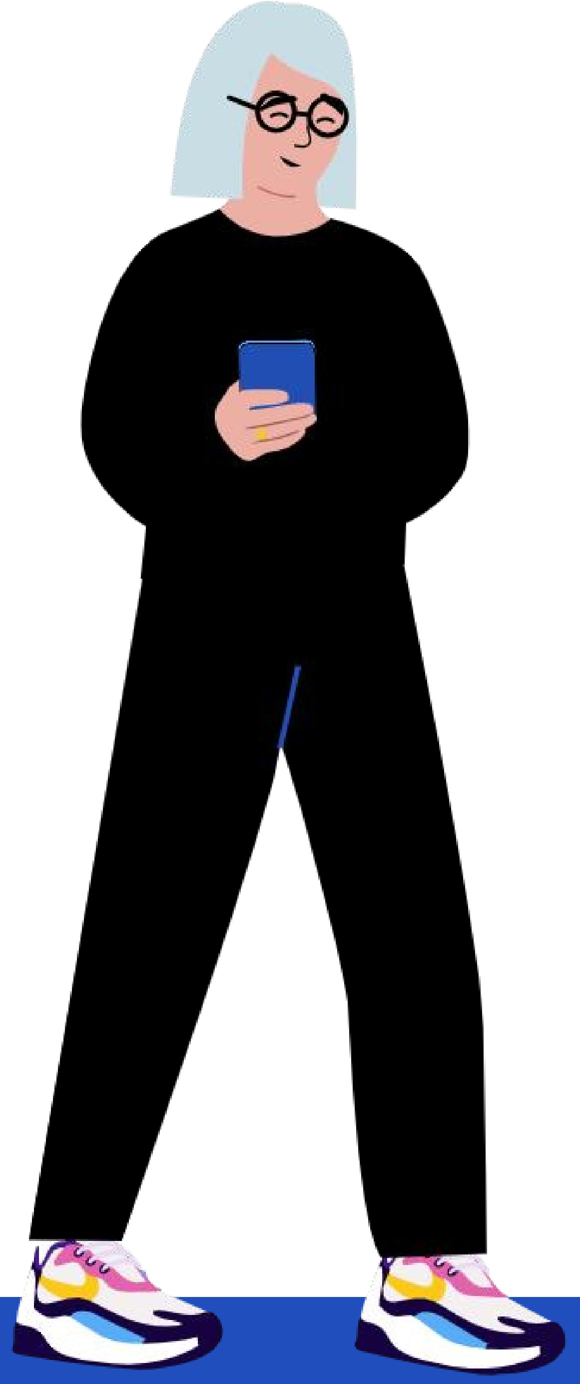
CORPORATE
GURUKUL

PROBLEM STATEMENT

- Distracted Driving Is A Significant Contributor To Road Accidents, Leading To Injuries And Fatalities Worldwide.

- The Challenge Is To Develop An Accurate And Reliable Machine Learning Model That Can Detect And Classify Various Types Of Driver Distractions From Images.

- The Goal Is To Classify Images Into One Of Ten Categories Representing Different Driver Behaviors, Such As Safe Driving, Texting, Talking On The Phone, Eating, And More.



CORPORATE
GURUKUL

MEET OUR TEAM!

Kumaran Hariharan

Frontend And Integration

Roshan Rajendran

Model Design And Training

Tejaswini Swaroop

Model Building

Pradhap Rane

Model Training

Farhad Khan

Data Evaluation And Training

ABSTRACT

- Distracted Driving Poses A Significant Risk To Road Safety, Leading To Numerous Accidents.

- The Aim Of This Project Is To Develop An Advanced Machine Learning Model Capable Of Detecting Various Forms Of Driver Distractions From Images.

- The Project Involves Applying Image Processing Techniques And Deep Learning Algorithms To Identify And Categorize Driver Behaviors Into Distinct Classes Such As Texting, Eating, Talking On The Phone, And Others.

- Providing Real-Time Alerts By Text To Speech Mechanism.

- The Research Integrates
  1.Data Collection
  2.Model Training
  3.Performance Evaluation High Accuracy

IMPORTANT USE CASE

- **Drive Safe, Even When No Ones Watching**
- **Ola, Uber, Safe Driving, Cab**
- **Driving Schools, Post In Person Lectures Driver Monitor System.**
- **Cargo Safety, Ensure Goods Are Safe**
- **TARGET AUDIENCE :-**
  1.Cab Service Providers
  2.Car Rental Service Providers
  3.Driving School.
  4.Insurance Providers.
- **Logistic Service Providers Insurance Service Providers Law Enforcement**

CORPORATE
GURUKUL

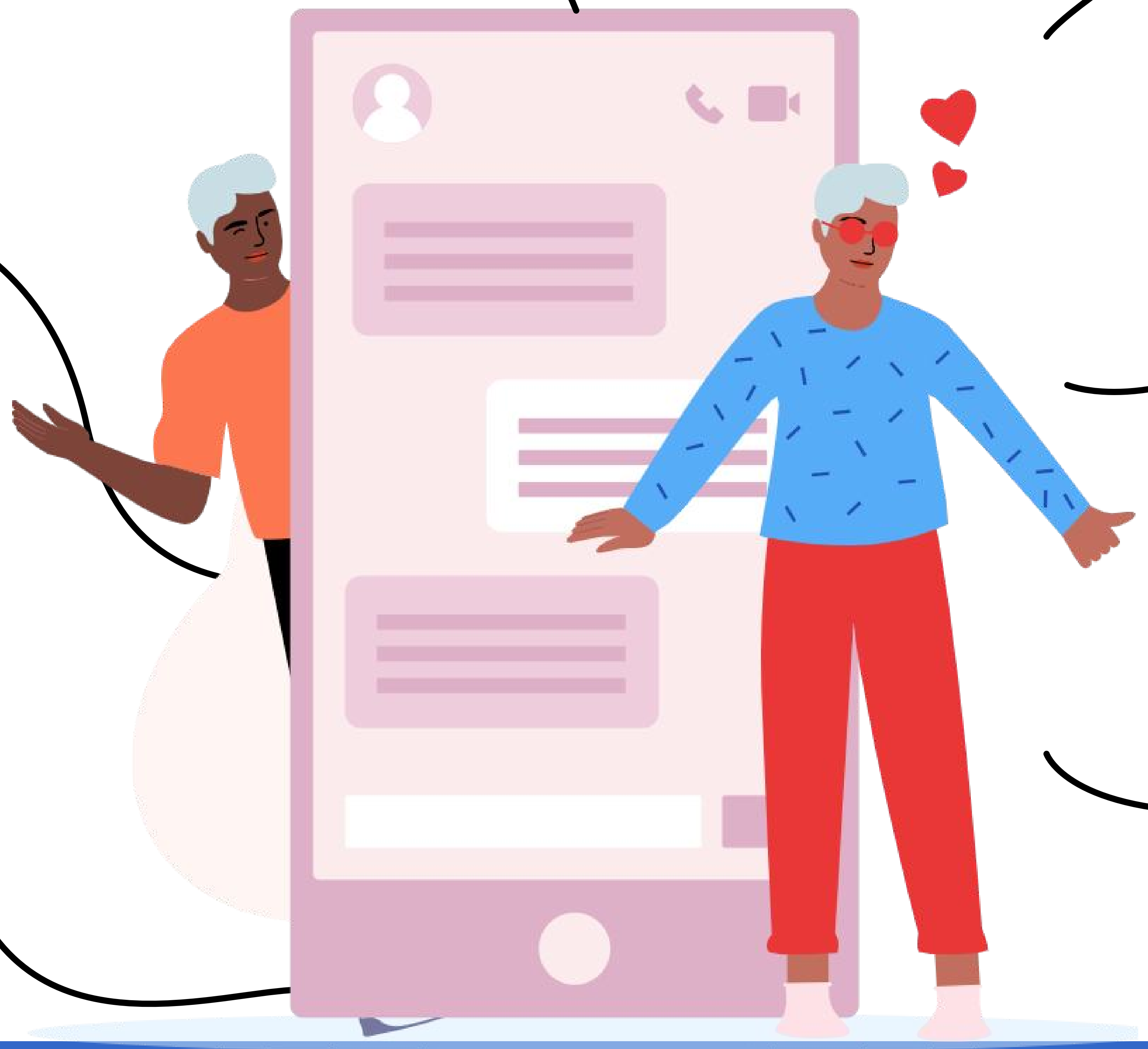
CONCEPTS USED

Streamlit
Frontend: Interactive web application

COMPUTER VISION
Image Classification: Predicting driver action
Preprocessing: Resizing and data augumentation

TENSERFLOW/KERAS
Frameworks: For neural network training
Training: Compiling, evaluating.

DEEP LEARNING
CNNs: For image data processing
Transfer Learning: Fine-tuning MobileNetV2



Data Augmentation:
Enhancements: Improving model performance.

MOBILE NET V2
Pretrained Model: Lightweight CNN
Custom Layers: For classification

Model Evaluation:
Metrics: Accuracy and loss monitoring

CORPORATE
GURUKUL

Data Used!

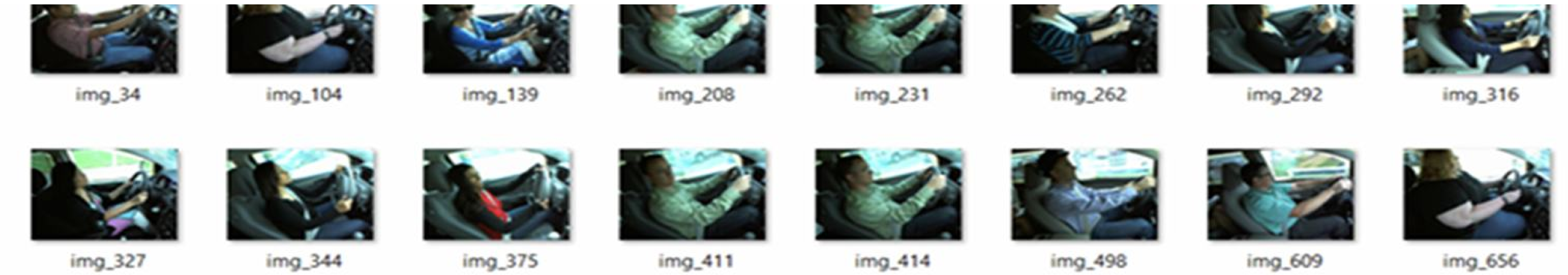
Driver Images, Each Taken In A Car With A Driver Doing Something Unsafe In The Car (Texting, Eating, Talking On The Phone, Makeup, Reaching Behind, Etc) Organized Into 10 Classes.

The 10 Classes To Predict Are:
Co: Safe Driving
C1: Texting-Right
C2: Talking On The Phone – Right
C3: Texting-Left
C4: Talking On The Phone – Left
C5: Operating The Radio
C6: Drinking
C7: Reaching Behind
C8: Hair And Makeup
C9: Talking To Passenger

GLIMPSE OF THE DATASET!

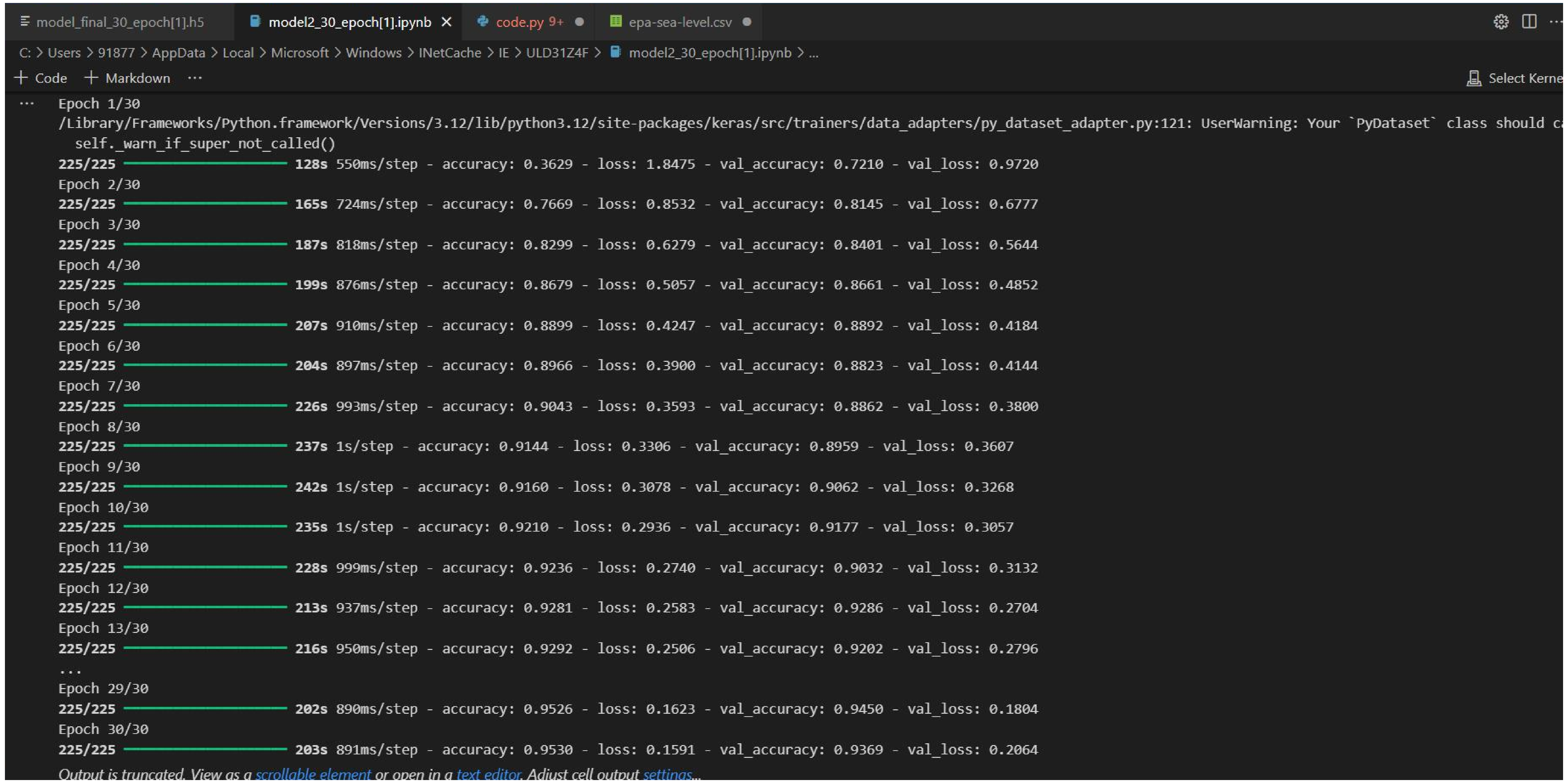
C0:Safe State

C8:Unsafe State(Hair And Makeup)



CORPORATE
GURUKUL

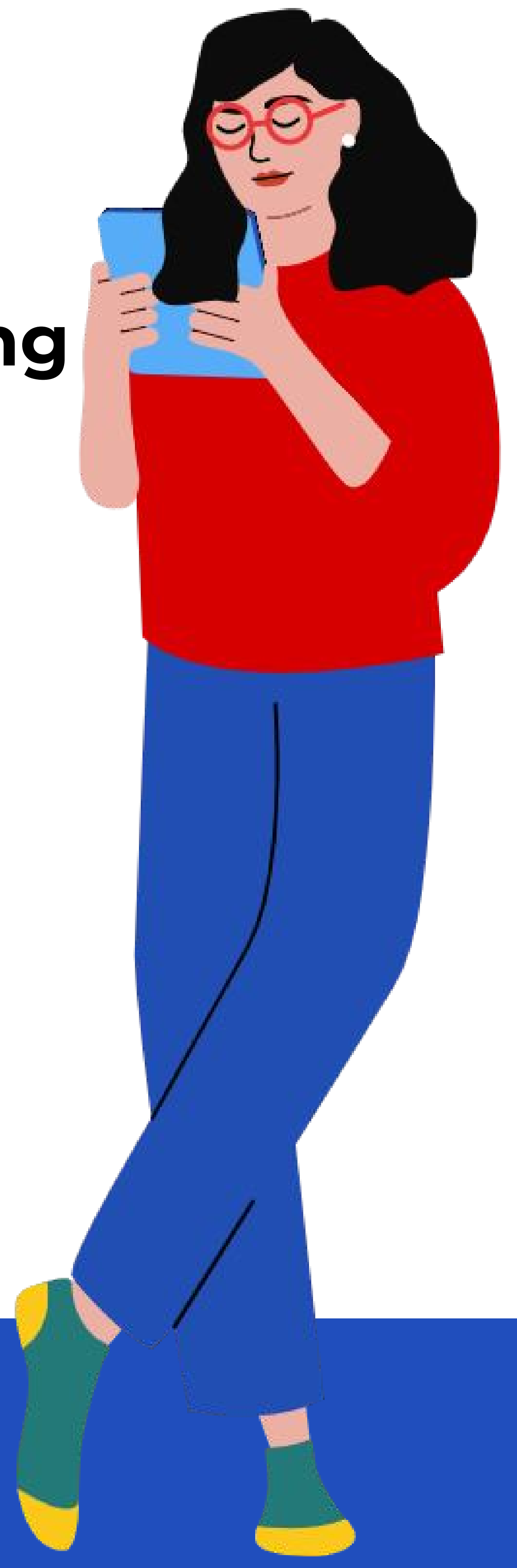
MODEL TESTING AND CELL OUTPUT




```
Epoch 1/30
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should c
  self._warn_if_super_not_called()
225/225 ━━━━━━━━━━━━━━━━━━━━ 128s 550ms/step - accuracy: 0.3629 - loss: 1.8475 - val_accuracy: 0.7210 - val_loss: 0.9720
Epoch 2/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 165s 724ms/step - accuracy: 0.7669 - loss: 0.8532 - val_accuracy: 0.8145 - val_loss: 0.6777
Epoch 3/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 187s 818ms/step - accuracy: 0.8299 - loss: 0.6279 - val_accuracy: 0.8401 - val_loss: 0.5644
Epoch 4/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 199s 876ms/step - accuracy: 0.8679 - loss: 0.5057 - val_accuracy: 0.8661 - val_loss: 0.4852
Epoch 5/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 207s 910ms/step - accuracy: 0.8899 - loss: 0.4247 - val_accuracy: 0.8892 - val_loss: 0.4184
Epoch 6/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 204s 897ms/step - accuracy: 0.8966 - loss: 0.3900 - val_accuracy: 0.8823 - val_loss: 0.4144
Epoch 7/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 226s 993ms/step - accuracy: 0.9043 - loss: 0.3593 - val_accuracy: 0.8862 - val_loss: 0.3800
Epoch 8/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 237s 1s/step - accuracy: 0.9144 - loss: 0.3306 - val_accuracy: 0.8959 - val_loss: 0.3607
Epoch 9/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 242s 1s/step - accuracy: 0.9160 - loss: 0.3078 - val_accuracy: 0.9062 - val_loss: 0.3268
Epoch 10/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 235s 1s/step - accuracy: 0.9210 - loss: 0.2936 - val_accuracy: 0.9177 - val_loss: 0.3057
Epoch 11/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 228s 999ms/step - accuracy: 0.9236 - loss: 0.2740 - val_accuracy: 0.9032 - val_loss: 0.3132
Epoch 12/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 213s 937ms/step - accuracy: 0.9281 - loss: 0.2583 - val_accuracy: 0.9286 - val_loss: 0.2704
Epoch 13/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 216s 950ms/step - accuracy: 0.9292 - loss: 0.2506 - val_accuracy: 0.9202 - val_loss: 0.2796
...
Epoch 29/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 202s 890ms/step - accuracy: 0.9526 - loss: 0.1623 - val_accuracy: 0.9450 - val_loss: 0.1804
Epoch 30/30
225/225 ━━━━━━━━━━━━━━━━━━━━ 203s 891ms/step - accuracy: 0.9530 - loss: 0.1591 - val_accuracy: 0.9369 - val_loss: 0.2064
```

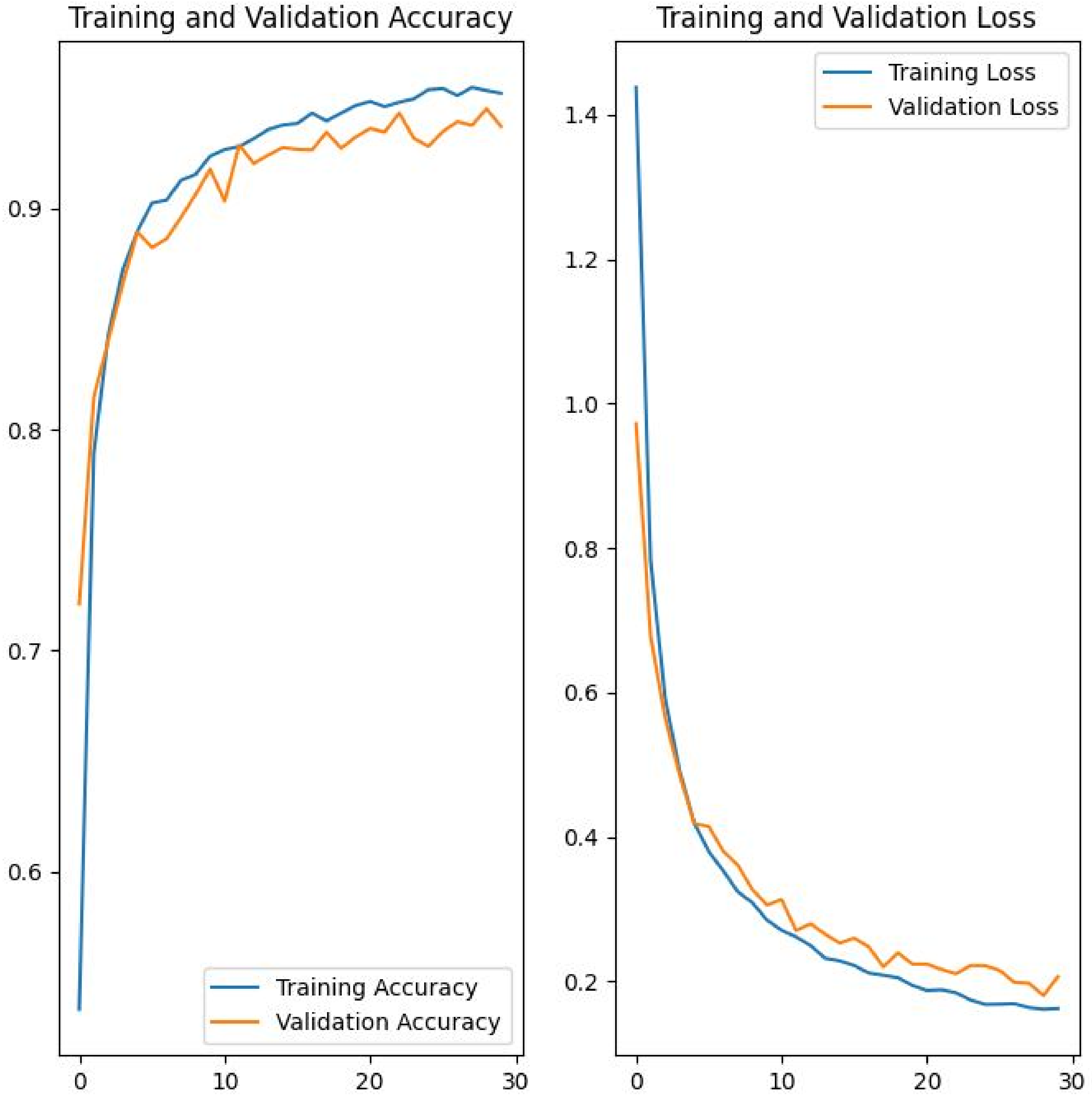
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

- **Model Testing**

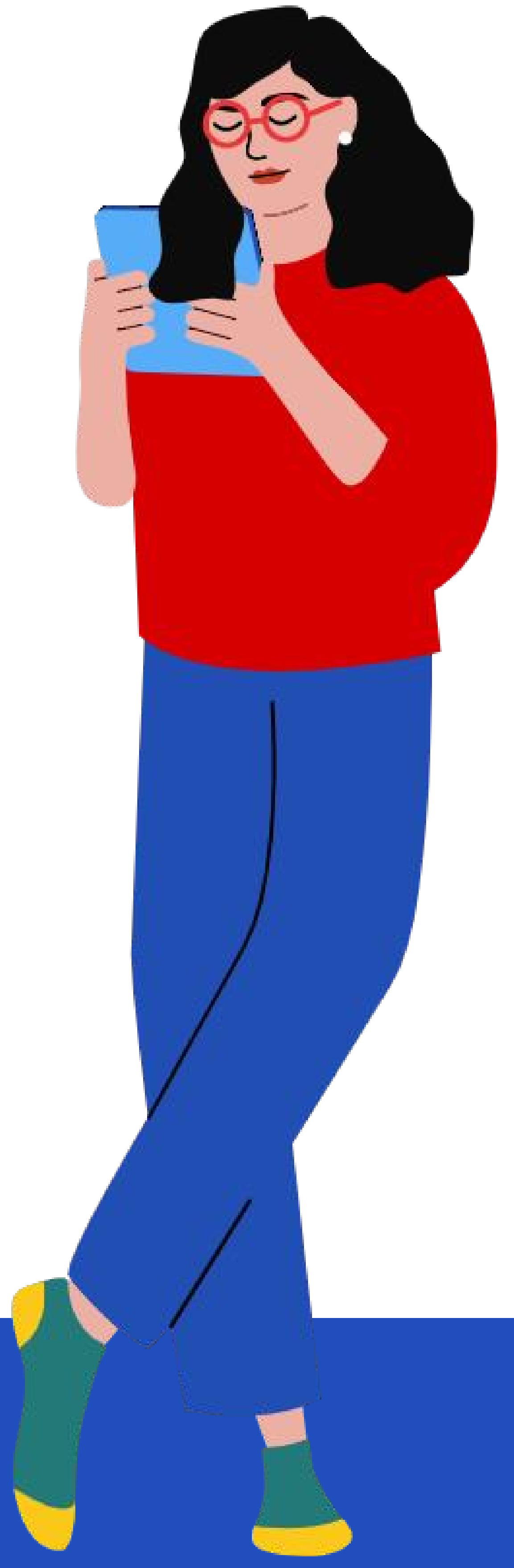


CORPORATE GURUKUL

Model Testing And Accuracy

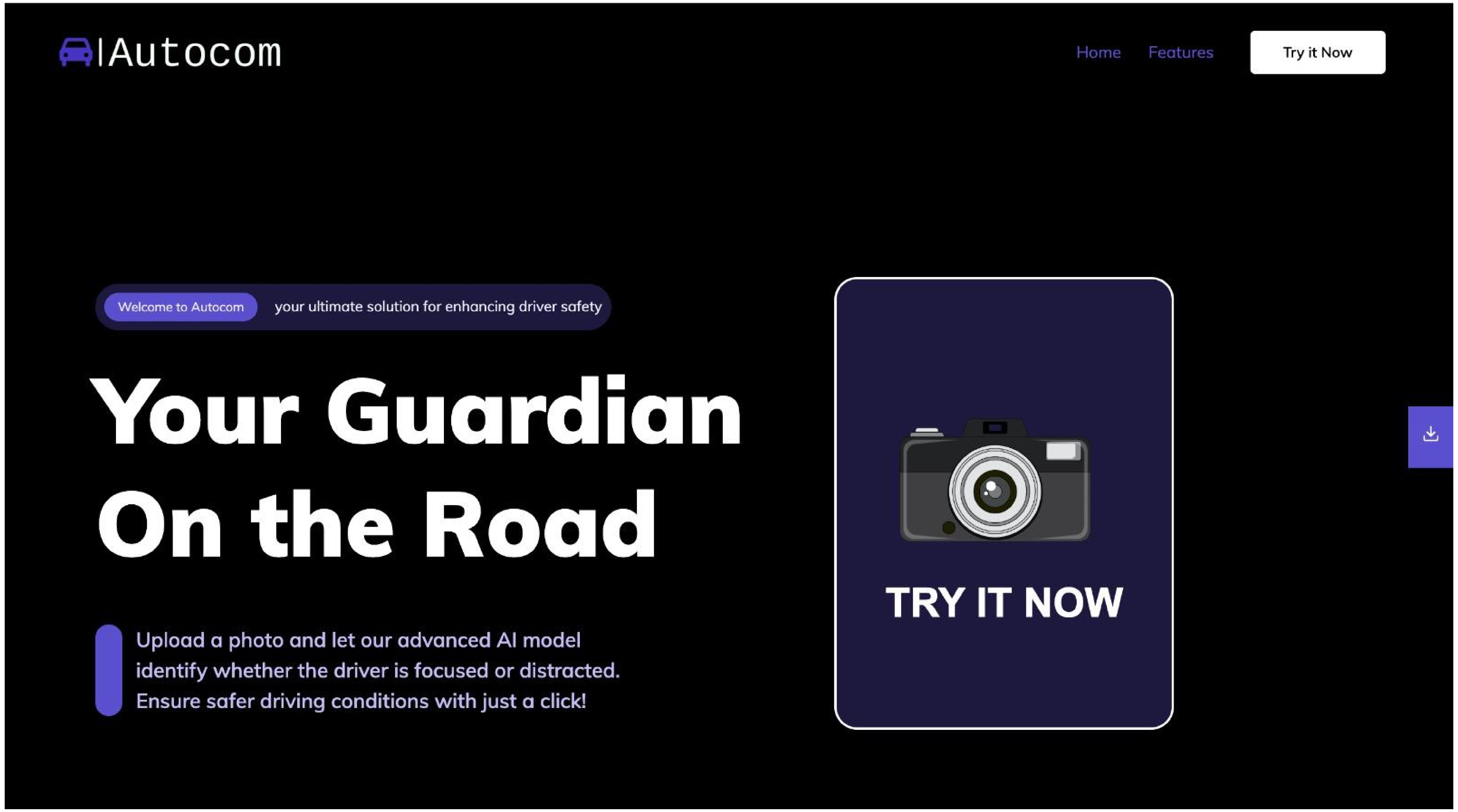


- Accuracy

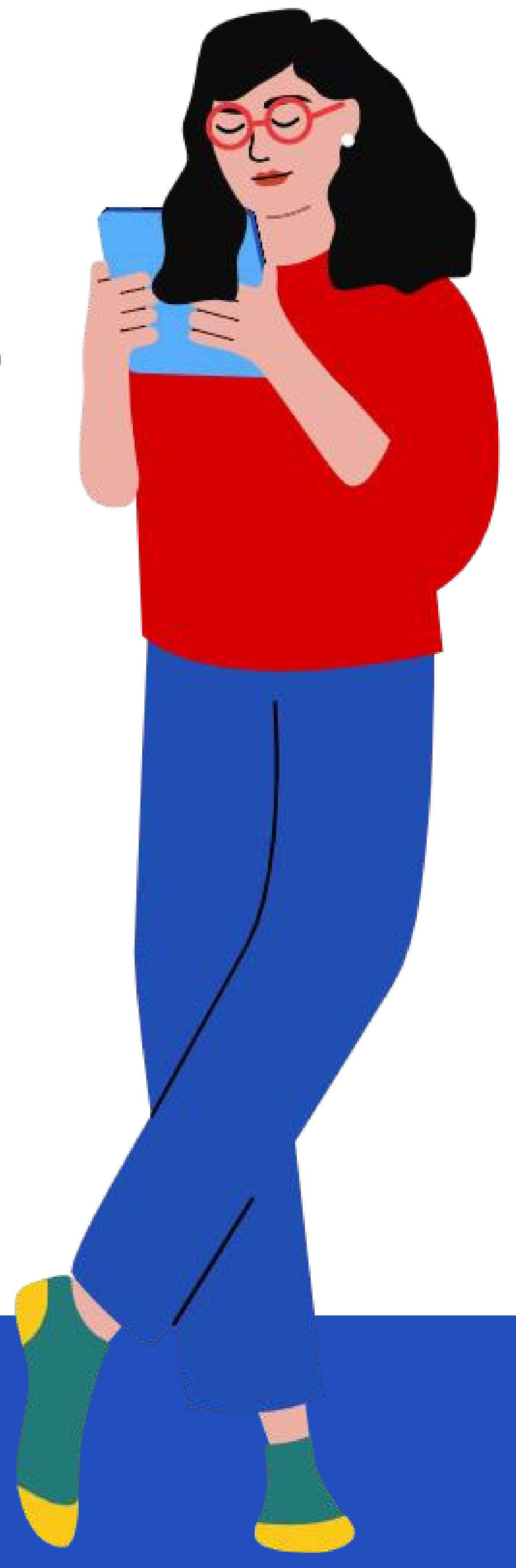


CORPORATE
GURUKUL

Glimpse Of The Project!

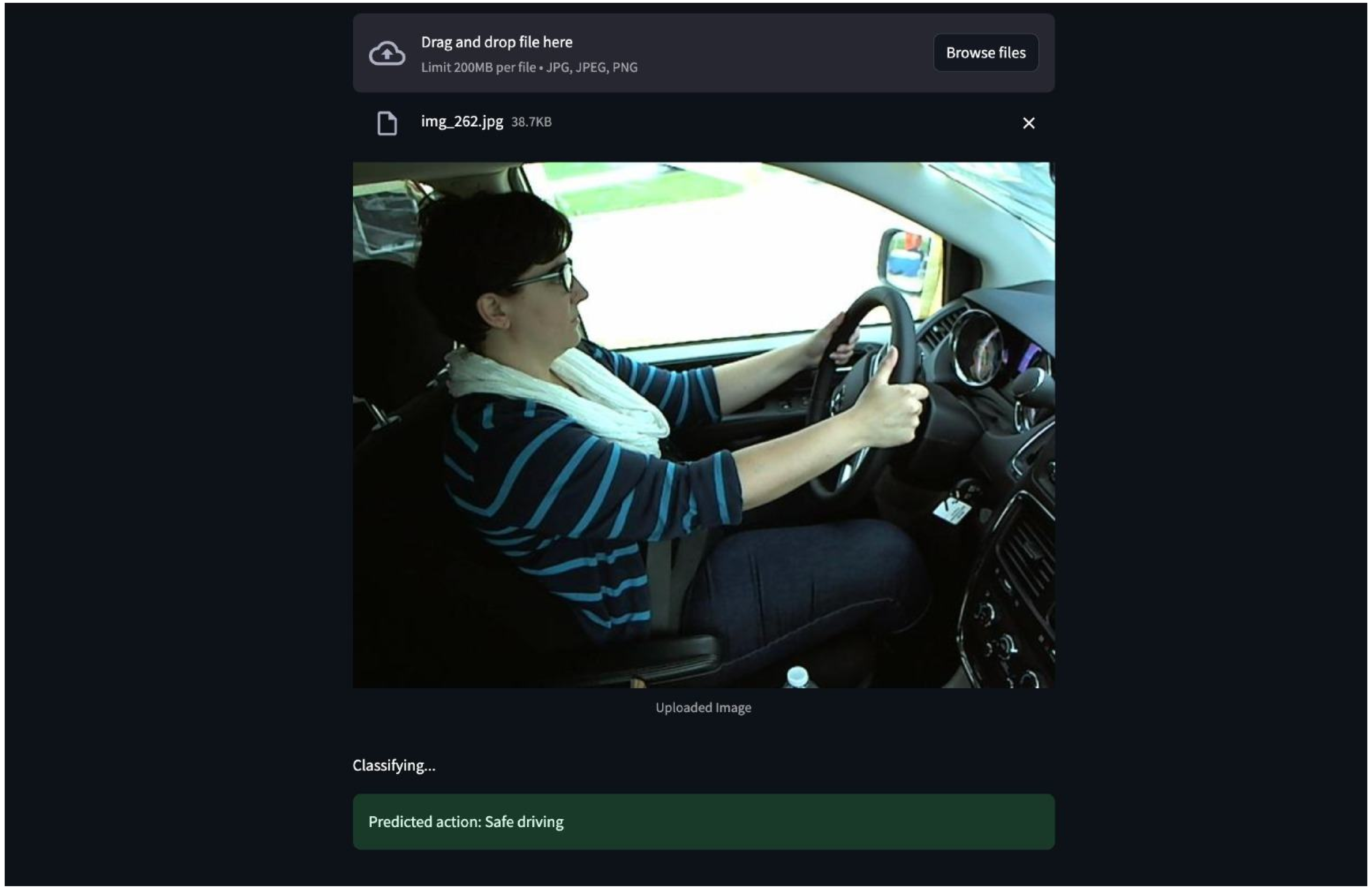


■ HOME PAGE

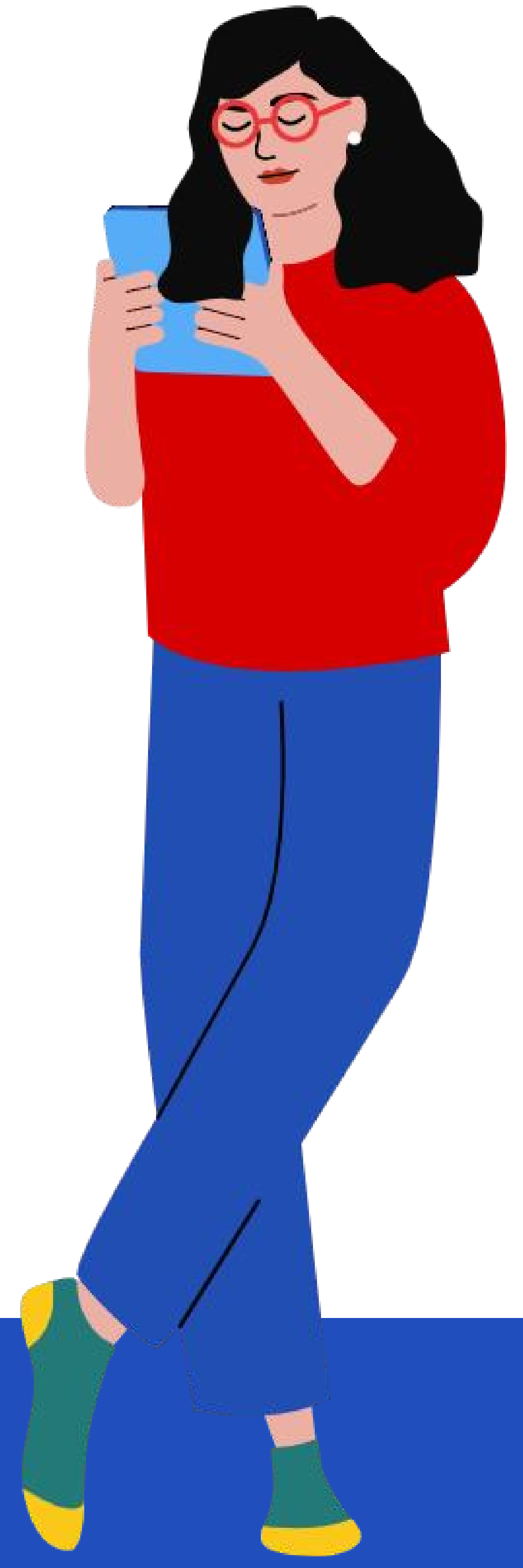


CORPORATE
GURUKUL

Glimpse Of The Project!

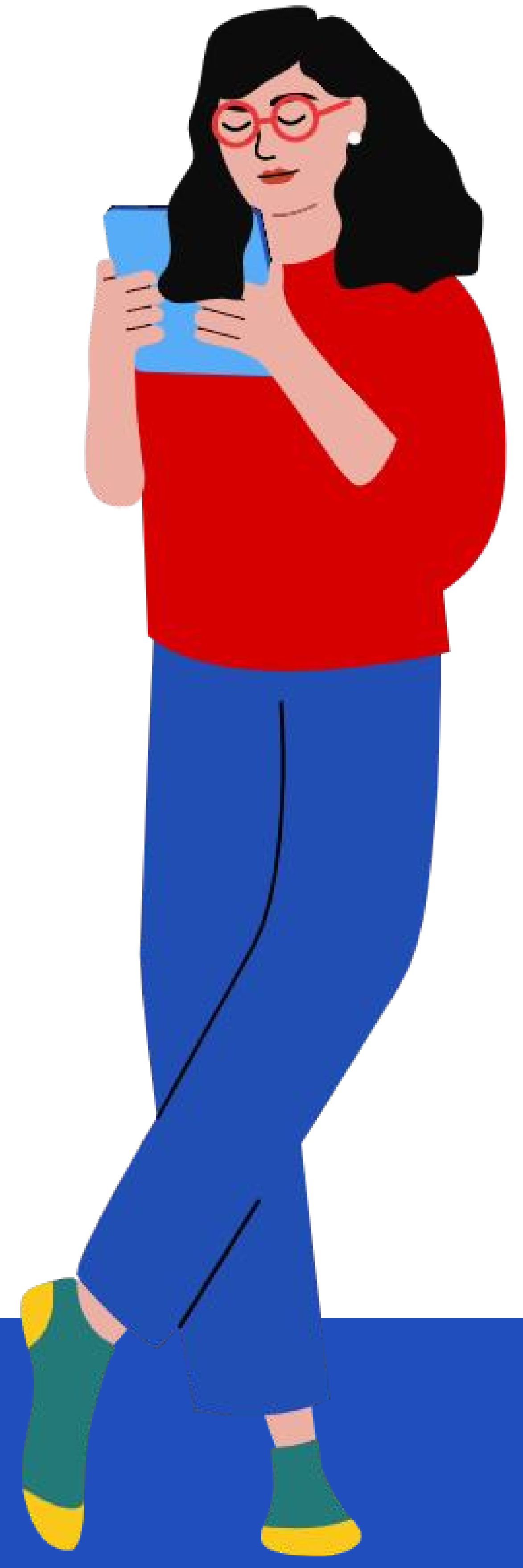


- **OUR MODEL**



CORPORATE
GURUKUL

LET'S CHECK OUT THE PROTOTYPE!!!



MOBILE NET V2 ARCHITECTURE

Type: Convolutional Neural Network (CNN)

Key Features:

- **Depthwise Separable Convolutions: Reduces Computational Cost By Factorizing A Standardconvolution Into A Depthwise Convolution Followed By A Pointwise Convolution.**
- **Inverted Residuals And Linear Bottlenecks: Improves Model Performance By Optimizing The Flow Of Information Through The Network.**

WE LEARNT!

Deep Learning Implementation

Data Augmentation

Model Training and Evaluation

Image processing

Deployment with streamlit

TensorFlow lite

CORPORATE
GURUKUL

Challenges faced

- Integration with Webcam
- Data Imbalance
- Model Optimization
- Real-Time Processing

Future Enhancement

1.Real-Time Alerts:
•Integrate Audio And Visual Alerts For Detected Risky Behaviors.

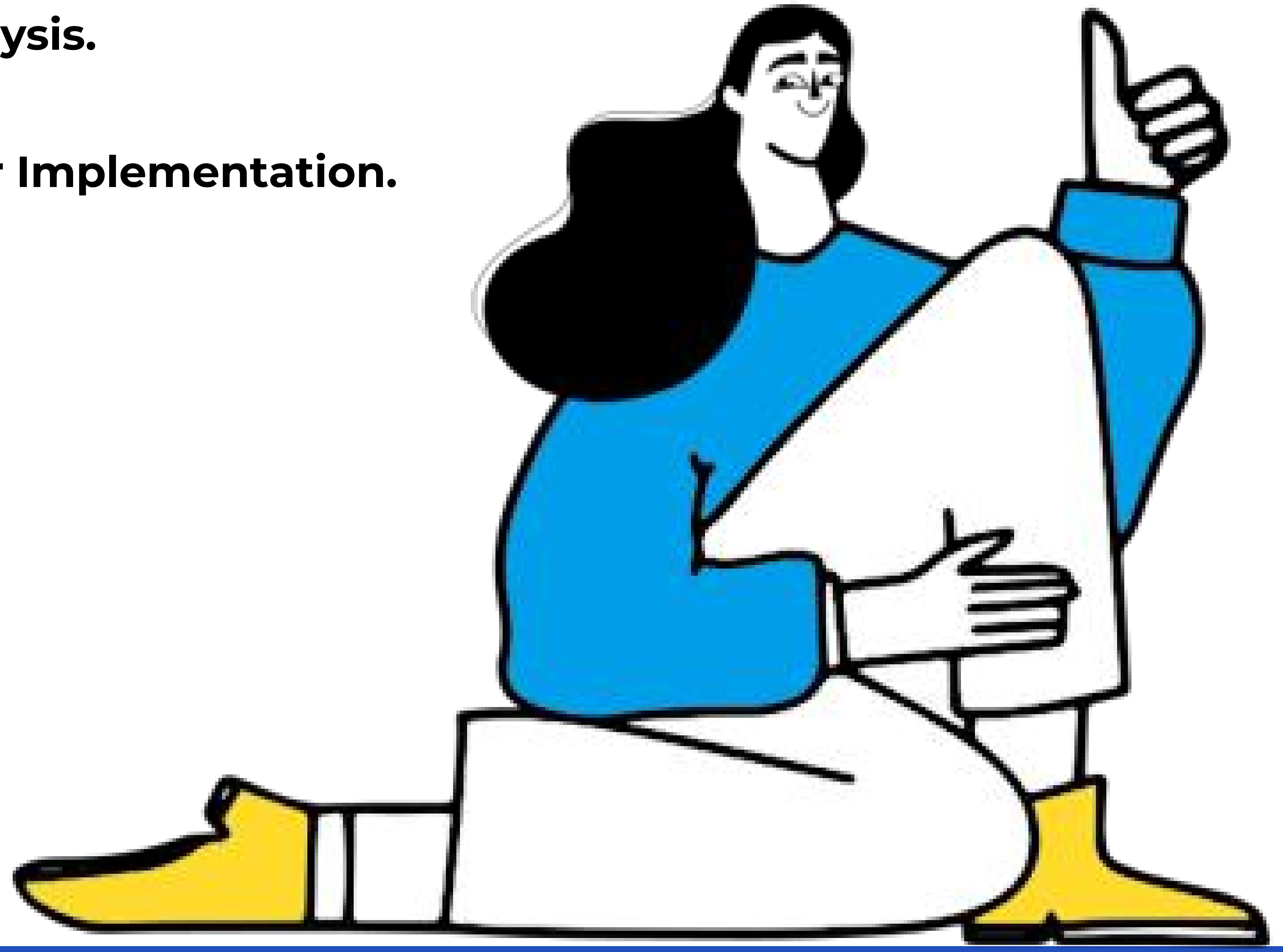
2.Extended Dataset:
•Incorporate Additional Driver Behaviors And Larger Datasets For Improved Accuracy.

3.Multimodal Inputs:
•Combine Image Data With Sensor Data (E.G., Speed, GPS) For Comprehensive Analysis.

4.Edge Deployment:
•Optimize The Model For Deployment On Edge Devices Like Raspberry Pi For In-Car Implementation.

5.Behavior Trends:
•Analyze And Visualize Driver Behavior Trends Over Time For Individual Drivers.

6.Custom Alerts:
•Allow Users To Set Custom Alerts For Specific Behaviors.



CONCLUSION

- This project demonstrates the potential of machine learning and computer vision techniques in enhancing road safety by detecting and classifying distracted driving behaviors.

- By accurately identifying various types of distractions from driver images, the developed model can be integrated into advanced driver assistance systems to provide real-time alerts, helping to mitigate the risks associated with distracted driving.