MANTHRA-X:

PIONEERING PRECISION, THE FUTURE OF AUTONOMOUS MOBILITY

Project ID; 24_25J_213

MEMBERS

Main Supervisor
Mr. Samadhi Rathnayake

Co-Supervisor
Dr. Lakmini Abeygunawardhana

Akalanka P.A.A
IT21160448

Ganepola G.A.N.B.
IT21155048

Athukorala W.A.A.D.D.T
IT21162978

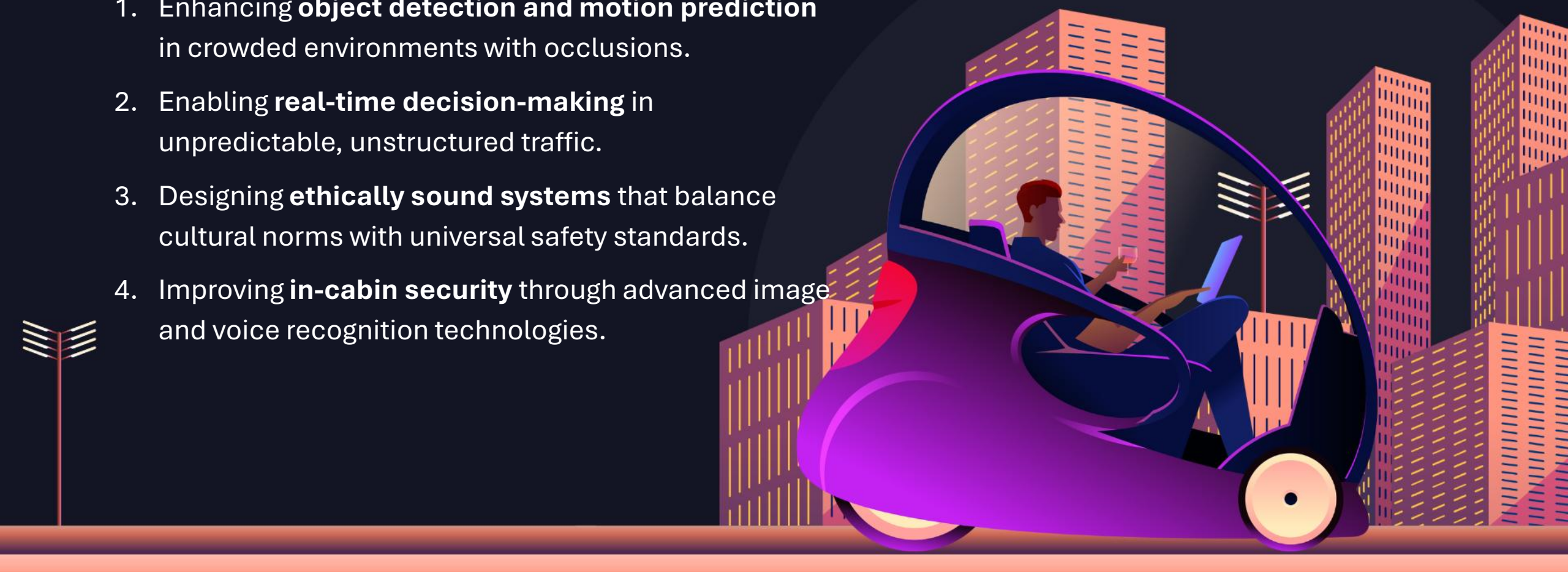
D.M.M.I.T Dissanayaka
IT21174780

INTRODUCTION

BACKGROUND

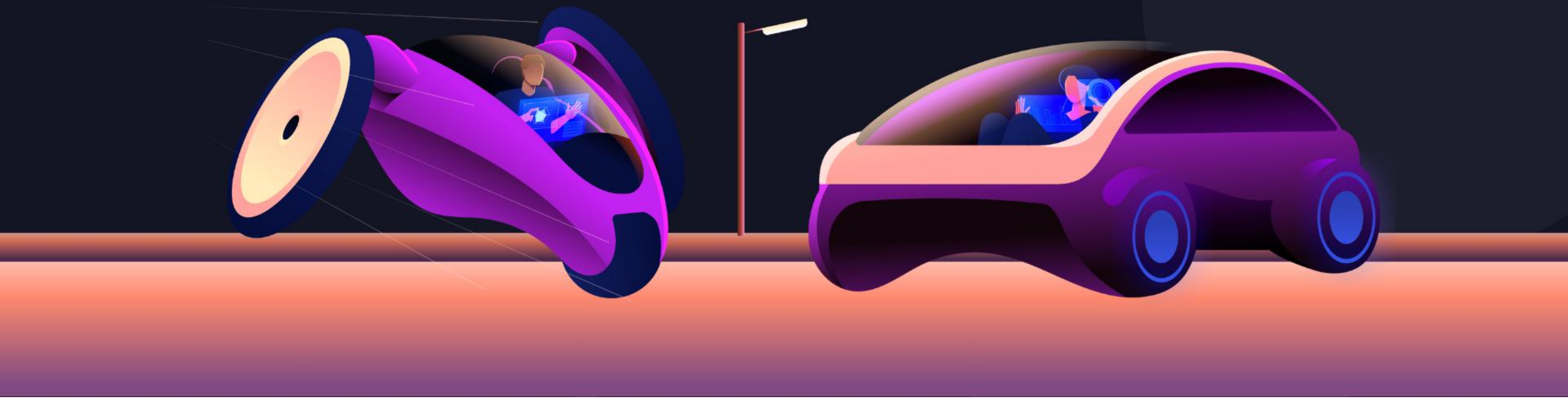
RESEARCH PROBLEM

1. Enhancing **object detection and motion prediction** in crowded environments with occlusions.
2. Enabling **real-time decision-making** in unpredictable, unstructured traffic.
3. Designing **ethically sound systems** that balance cultural norms with universal safety standards.
4. Improving **in-cabin security** through advanced image and voice recognition technologies.

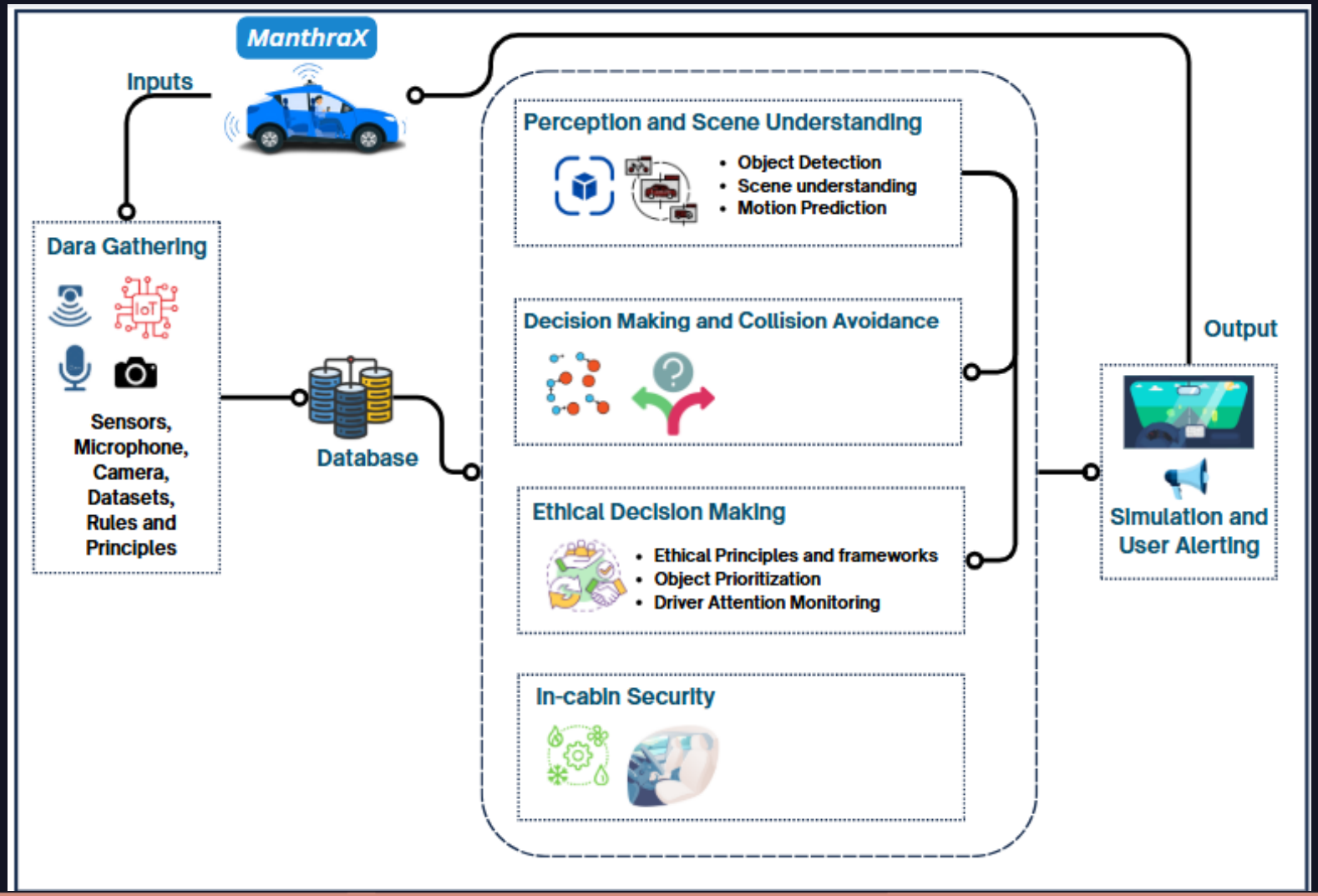


OBJECTIVE

Manthra-X is enhancing perception and scene understanding, decision-making, and in-cabin security. This involves integrating advanced machine learning models and simulations to improve vehicle safety, ethical decision-making, and passenger comfort in autonomous systems.



Overall System Diagram



Component 01

PERCEPION AND SCENE UNDERSTANDING

IT21160448 | AKALANKA P.A.A

INTRODUCTION

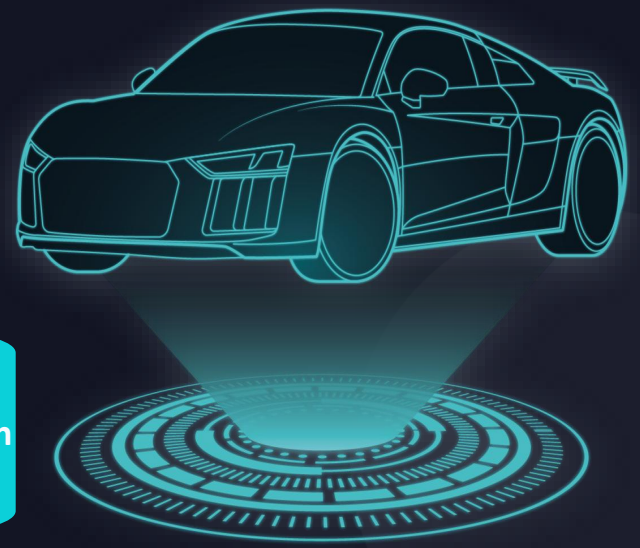
RESEARCH QUESTIONS ?

1. How can we achieve real-time, accurate object detection in complex environments using YOLOv5, handling occlusions and low-light conditions?

2. What role does an attention mechanism play in ensuring the vehicle stays within its lane and adapts to dynamic road conditions?

3. How can a hybrid model combining Graph Neural Networks (GNNs) and Transformers predict the future behavior of surrounding vehicles for collision avoidance?

4. What are the key challenges in integrating object detection, lane keeping, and motion prediction into a unified perception system for real-time autonomous driving?



RESEARCH GAP

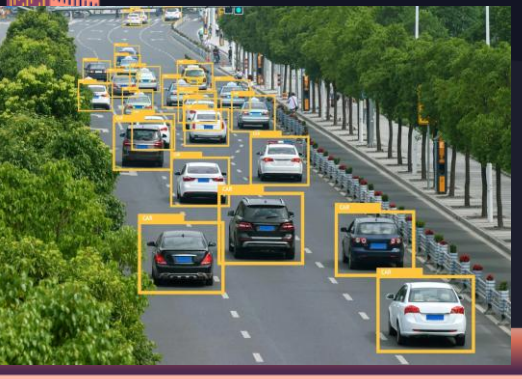
Features	Research 1	Research 2	Research 3	Research 4	Research 5	MANTHRA-X
Modeling of Spatial relationship between objects.	No	No	No	No	No	
Address both spatial relationships and temporal dependencies in object detection.	No	No	No	No	No	
Improved accuracy in occlusions and object deformations.	No	Yes	No	No	No	
Enhanced model's learning capability from unlabeled data.	No	No	No	No	Yes	
Combines different sensor modelities.	Yes	Yes	No	No	No	

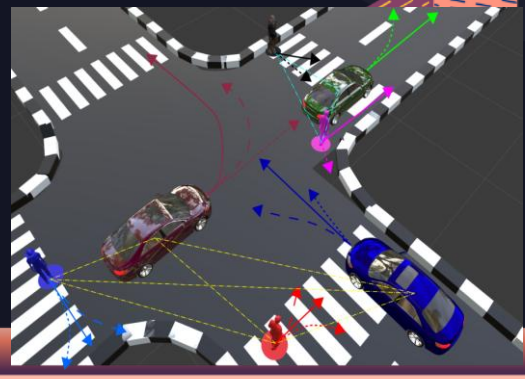
SPECIFIC AND SUB OBJECTIVES

SPECIFIC OBJECTIVE

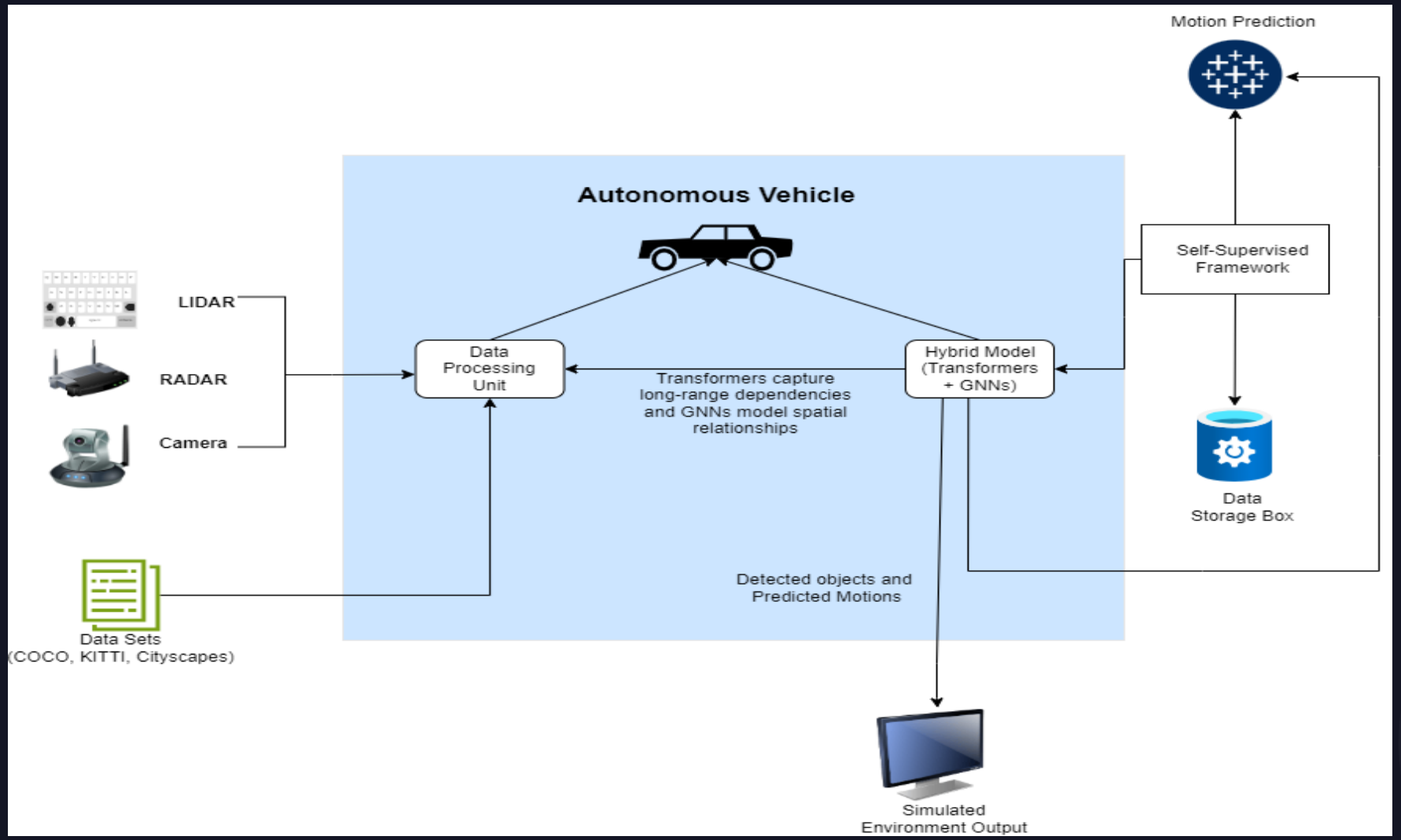
To develop a robust PERCEPTION AND SCENE UNDERSTANDING SYSTEM for Autonomous Vehicles

- REAL TIME OBJECT DETECTION: YOLO V5
- ATTENTION BASED LANE KEEPING: CNN
- MOTION PREDICTION USING HYBRID MODELS: GNN & TRANSFORMER
- SYSTEM INTERGRATION AND REAL TIME PERFORMANCE





SYSTEM DIAGRAM



SUB OBJECTIVE 01 – OBJECT DETECTION – RECAP

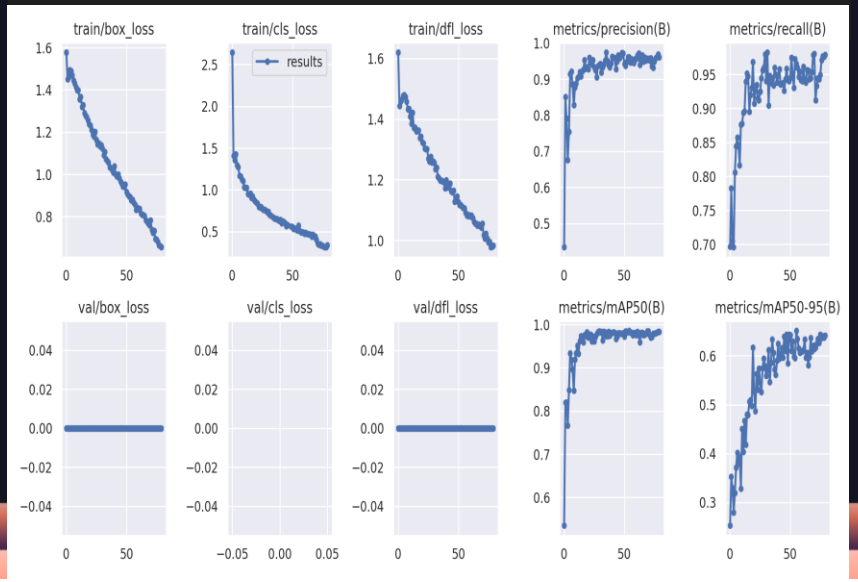
Sensors:
- Lidar Sensor
- Depth Camera

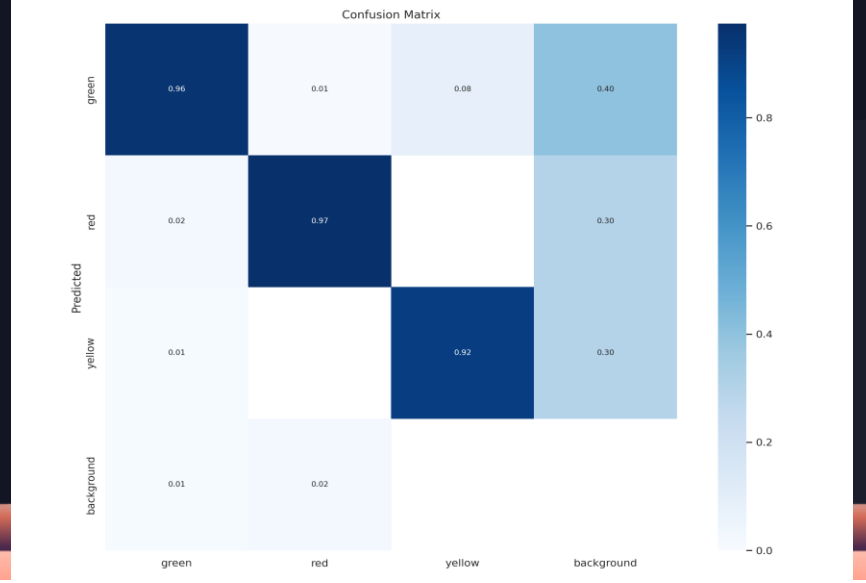
Data Set:
- ROBOFLOW

Object Detection Model:
- YOLO V5

Demonstration:
- CARLA Simulator


```
tensor([[137.12079, 195.06799, 290.26932, 298.42981,   0.94626,   9.00000],
        [455.11258, 171.43365, 467.20508, 214.65485,   0.90649,   5.00000],
        [246.85228, 237.17397, 543.95447, 446.50391,   0.85170,   9.00000]])
```






CARLA
Windows PowerShell

SUB OBJECTIVE 02 – SEMENTIC DATA COLLECTION FEATURE AND LANE KEEPING - RECAP

Sensors:

RGB Camera
Sementic Camera

Data Set:

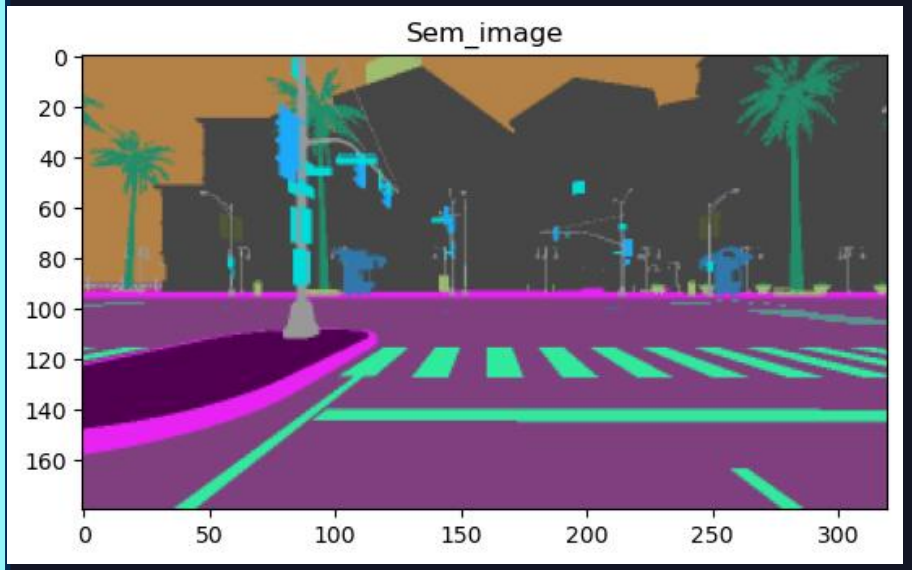
Created by Sementic data collection feature.

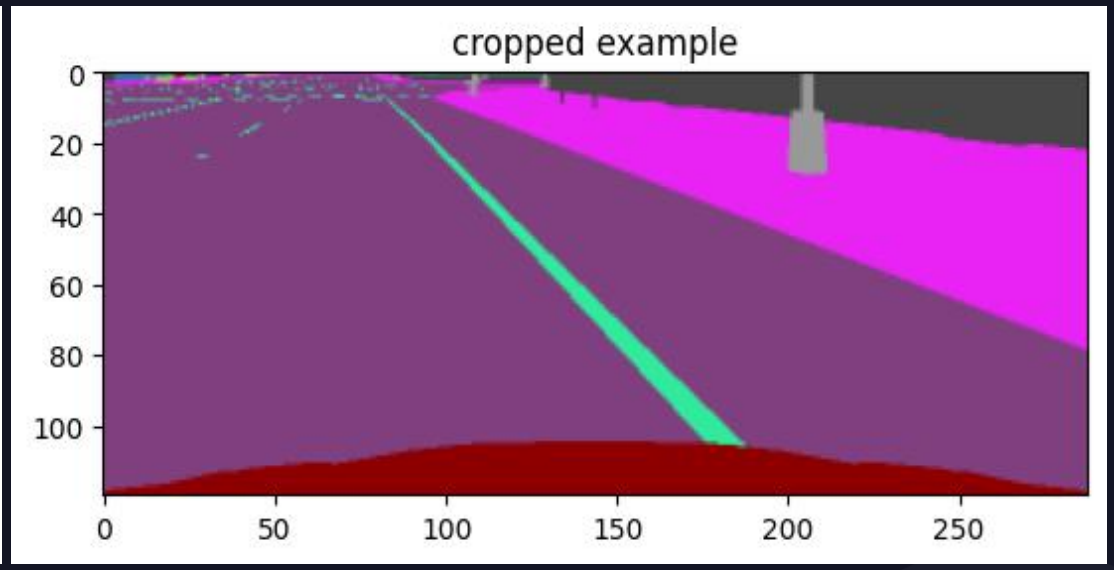
Model:

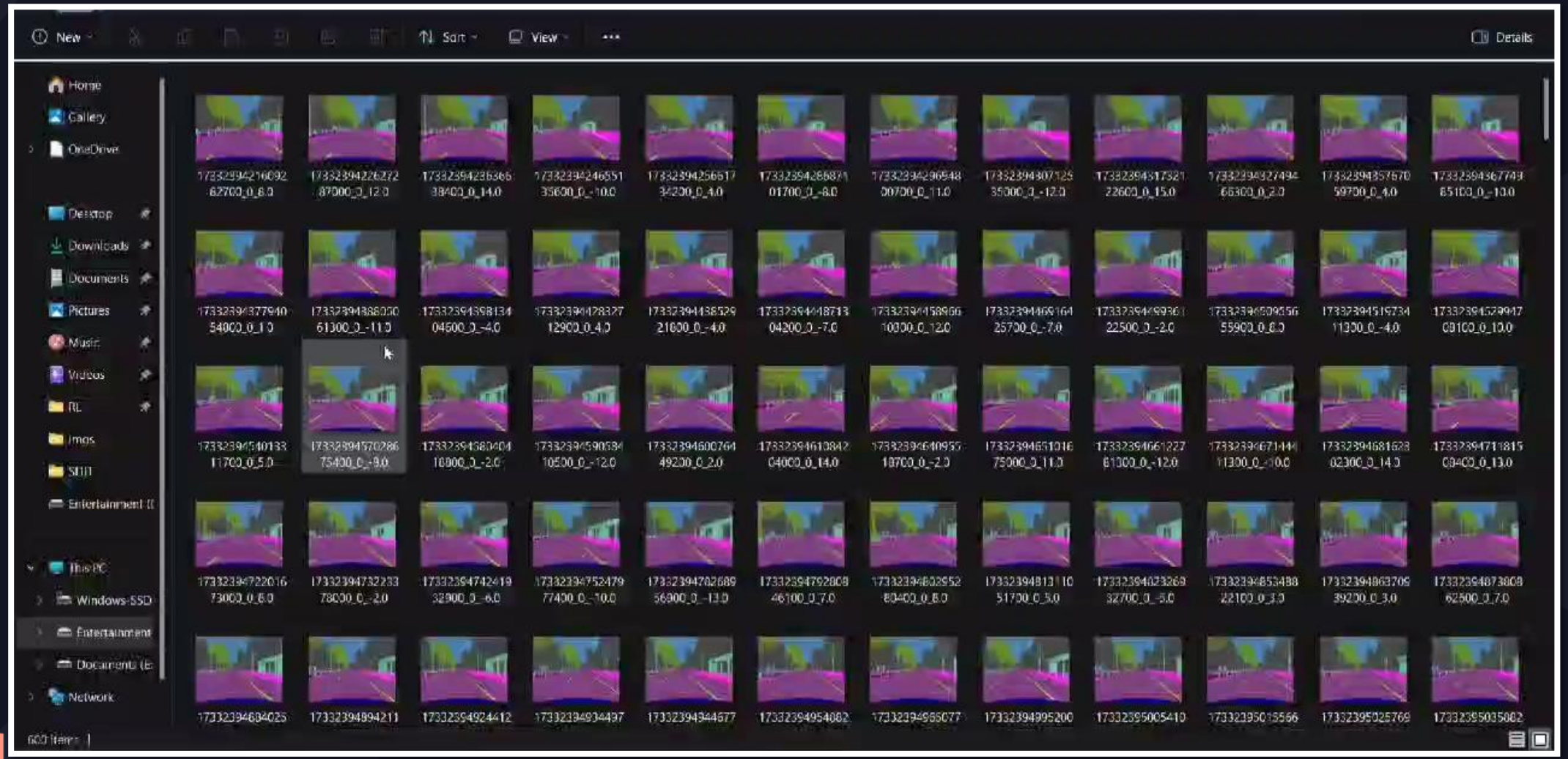
CNN

Demonstration:

CARLA Simulator






New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
RL
Jmos
SLIIT
Entertainment (I
This PC
Windows-SSD
Entertainment
Documents (E:
Network
600 items

SUB OBJECTIVE 03 – MOTION PREDICTION USING HYBRID MODELS [GNN + TRANSFORMER]

GraphSAGE Layer [Type of GNN]
[Spatial Relationships]

+

Transformer Blocks [Type of neural network architecture that uses self-attention mechanism
[Long Range Dependencies]

→

HYBRID MODEL
[GNN + TRANSFORMER]

INPUTS – Data Structure

```
type  - [0,0,0]   : Pdedestrian, Vehicle, Motorcycle
dist  - [0]       : d/range
speed - [0]       : s/max_speed
r_ang - [0]       : a/360
behav - [0,0,0,0] : Stopped, Going (Heading same direction),
Coming (Heading opposite direction), Crossing
```

```
###SAMPLE INPUT###
type = [0, 1, 0] #Vehicle
distance = 10 / 25 = 0.4 #10m away
speed = 30 / 60 = 0.5  #Moving at 30km/h
relative_angle = 90 / 360 = 0.25 #Relative angle 90
Behavior =[0, 1, 0, 0] #Moving along the same direction.
```

DATA_LOADER

Data set Preparation - One Hot Encoding

```
OBJECT_TYPE = {
        0: [1, 0, 0],   # Pdedestrian
        1: [0, 1, 0],   # Vehicle
        2: [0, 0, 1],   # Motorcycle
}
```

```
BEHAVIOR = {
        0: [1, 0, 0, 0],   # Stopped
        1: [0, 1, 0, 0],   # Going
        2: [0, 0, 1, 0],   # Coming
        3: [0, 0, 0, 1]    # Crossing
}
```

Final Feature Vector of one object detected

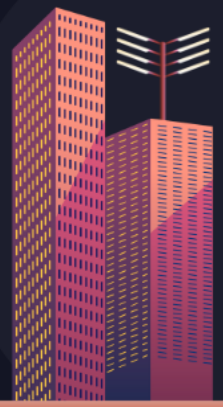
```
Final_feature_vector = [0, 1, 0, 0.4, 0.5, 0.25, 0, 1, 0, 0]
```

```
Dataset size: 20625
Features Shape: (11, 7)
Edge Index Shape: (2, 10)
Labels Shape: (10,)
```

SUB OBJECTIVE 03 – MOTION PREDICTION USING HYBRID MODELS [GNN + TRANSFORMER]

COMPARISON

Model	Strengths	Limitations
CNNs & RNNs	CNNs capture spatial features; RNNs model sequences.	Struggle with object relationships and long-range dependencies.
GNNs	Understands spatial relationships in dense environments.	Weak in capturing temporal patterns.
Transformers	Excellent at modeling long-range dependencies.	Lacks spatial awareness; computationally heavy.
Hybrid Model (GNN + Transformer)	**Best of both worlds** – GNNs for spatial, Transformers for temporal learning. **98.75% accuracy** in motion prediction. **Optimized for real-time scenarios.**	Higher computational cost but vastly superior performance.



SUB OBJECTIVE 03 – MOTION PREDICTION USING HYBRID MODELS [GNN + TRANSFORMER]

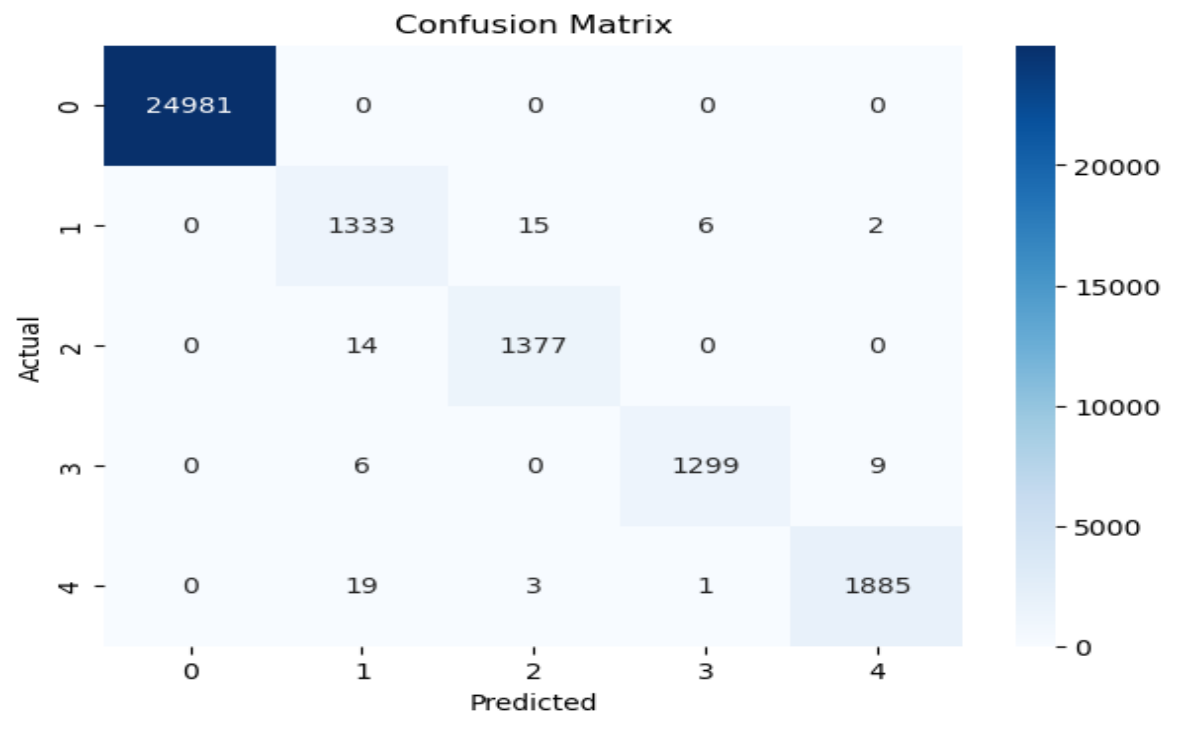
```
test_loss, test_accuracy = model.evaluate(
    test_dataset.map(lambda f, e, l: ((f, e), l))
)
print(f"Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f}")
```

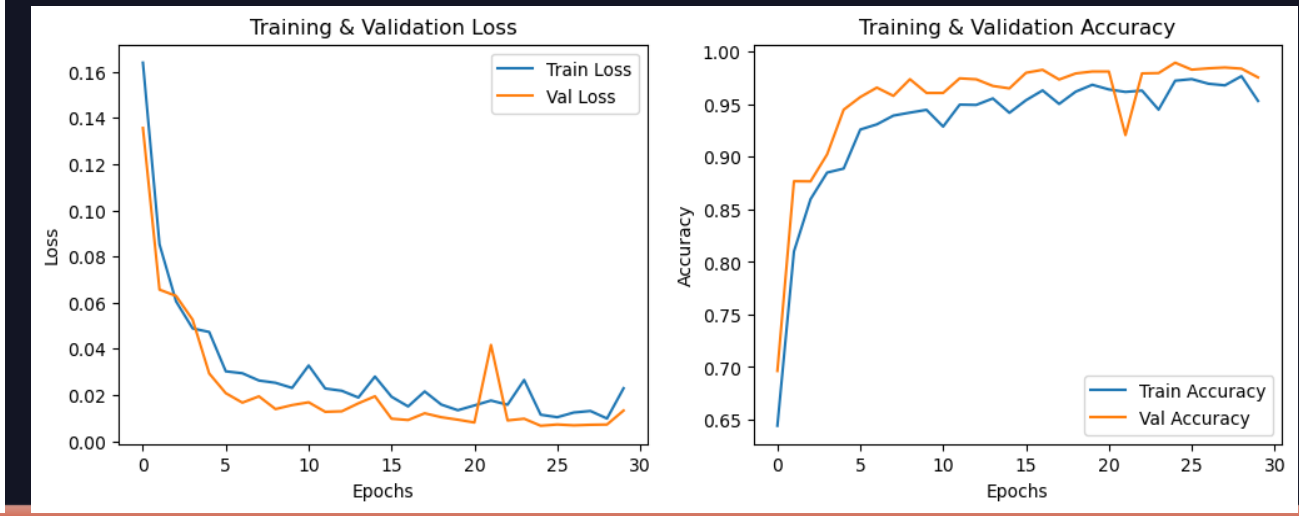
```
Class-wise Accuracy:
Class 2: 0.9899
Class 4: 0.9879
Class 0: 1.0000
Class 3: 0.9886
Class 1: 0.9830
```

```
print(classification_report(y_true, y_pred, digits=4))
```

	precision	recall	f1-score	support
0	1.0000	1.0000	1.0000	24981
1	0.9716	0.9830	0.9773	1356
2	0.9871	0.9899	0.9885	1391
3	0.9946	0.9886	0.9916	1314
4	0.9942	0.9879	0.9911	1908
accuracy			0.9976	30950
macro avg	0.9895	0.9899	0.9897	30950
weighted avg	0.9976	0.9976	0.9976	30950

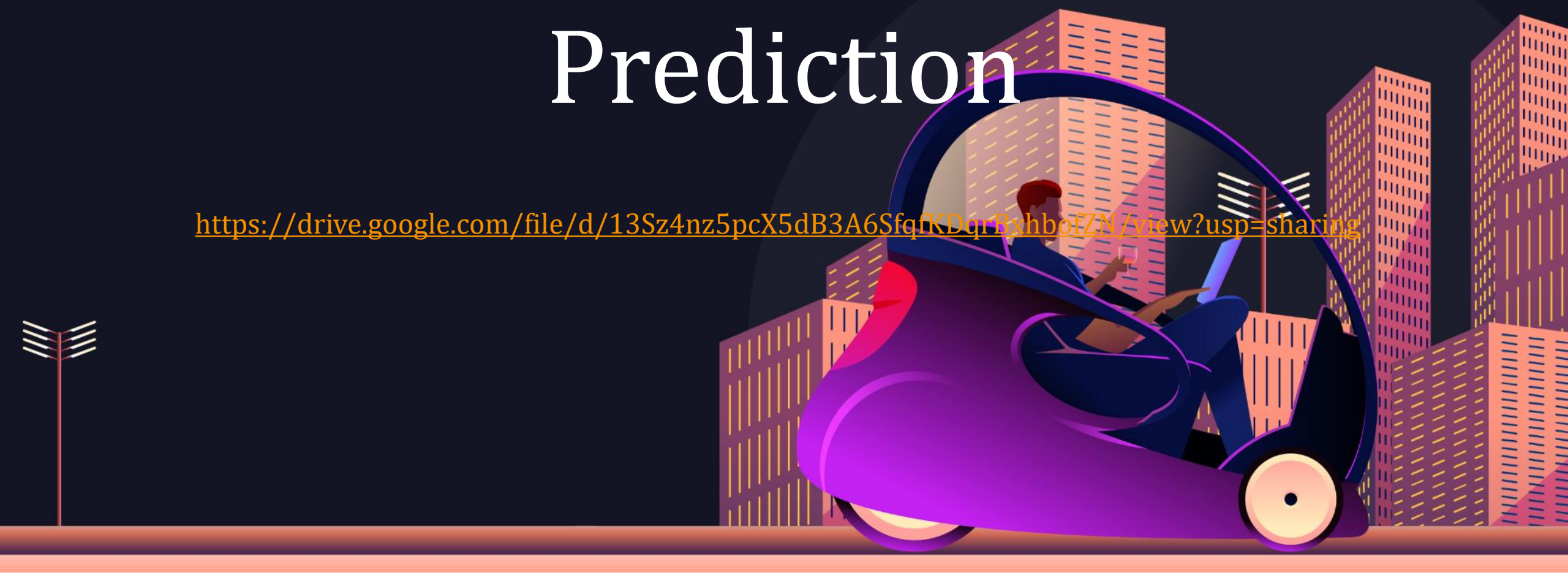
```
97/97 [==============================] - 21s 37ms/step - loss: 0.0065 - custom_accuracy: 0.9875
Test Loss: 0.0065, Test Accuracy: 0.9875
```





Demonstration of Motion Prediction

https://drive.google.com/file/d/13Sz4nz5pcX5dB3A6SfqfKDqrBxhbofZN/view?usp=sharing



Progress

Completed Tasks

- Object Detection by YOLOv5 – PP1
- Semantic Data Collection Feature by CNN –PP1
- Motion Prediction Using Hybrid Model [GNN + TRANSFORMER] – PP2

Upcoming Tasks

- Demonstrating how Hybrid model is better than other models
- Demonstrating how Hybrid model behaves in different scenarios.

REFERENCES

[1]Jyoti Madake;Tejas Lokhande;Atharv Mali;Nachiket Mahale;Shripad Bhatlawande, **"TransVOD: Transformer-Based Visual Object Detection for Self-Driving Cars",** 2024 International Conference on Current Trends in Advanced Computing (ICCTAC)Year: 2024 | Conference Paper | Publisher: IEEE

[2]Tim Meinhardt;Alexander Kirillov;Laura Leal-Taixé;Christoph Feichtenhofer, **"TrackFormer: Multi-Object Tracking with Transformers:",** 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Year: 2022 | Conference Paper | Publisher: IEEE

[3]"Transformers in Autonomous Driving: A Survey"

[4]"Dynamic Graph Neural Networks for Modeling Object Interactions"

[5]"Self-Supervised Learning for Object Detection and Tracking"

TECHNOLOGIES TO BE USED

AI/ML Frameworks:
Tensorflow, Pytorch, Keras

Development Tools:
IDEs: VSCode, PyCharm,
Jupyter Notebook

Cloud Services:
Microsoft Azure

Databases:
NoSQL Databases:
MongoDB, Cassandra

Simulation Frameworks:
CARLA, AirSim, or Unity

Web Frameworks: Node.js,
Flask, Django

IT21155048|GANEPOLA G.A.N.B.

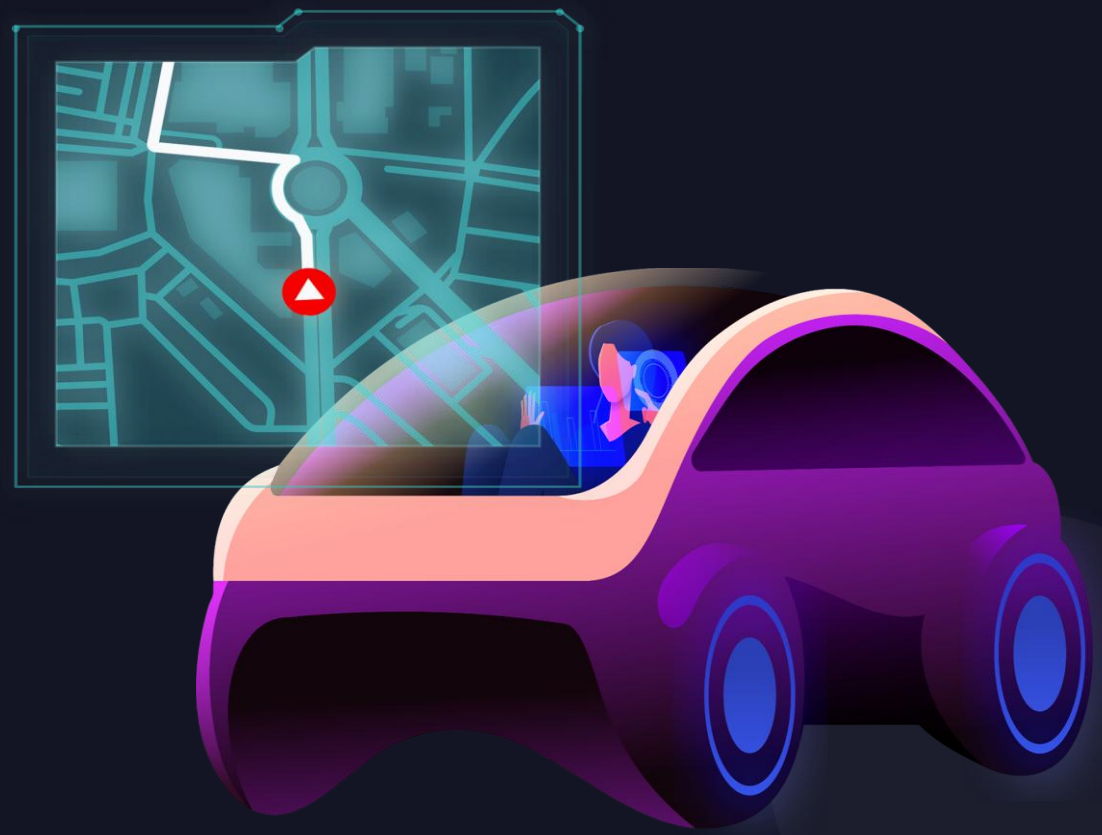
BSc (Hons) in Information Technology (specialization in Data Science)

Component 02

DECISION MAKING AND COLLISION AVOIDANCE

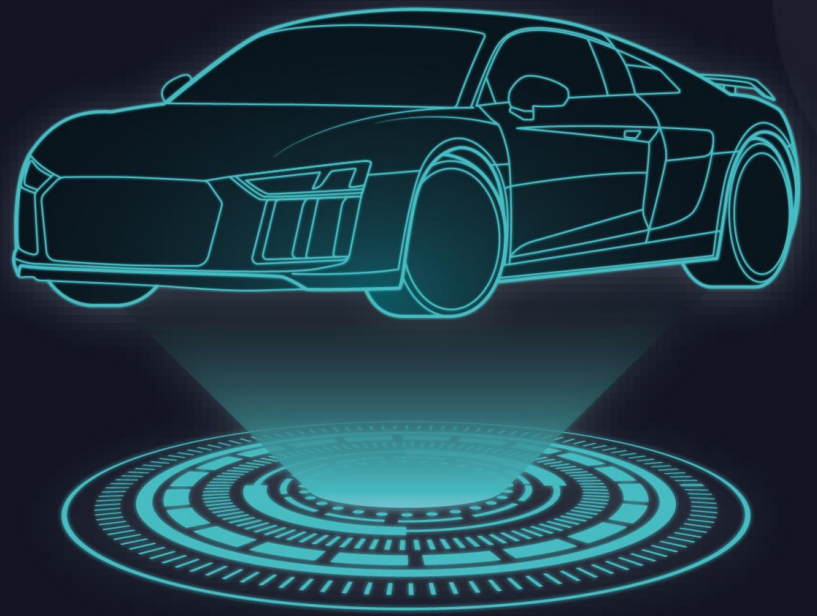
BACKGROUND

Decision Making and Path Planning Approach for Navigation in Complex Traffic Environments and Ensuring Collision Avoidance.



MAIN OBJECTIVE

Develop and integrate advanced decision making and path planning capabilities for vehicles to improve collision avoidance and adaptability in dynamic traffic environments, with a particular focus on addressing the unique driving conditions.



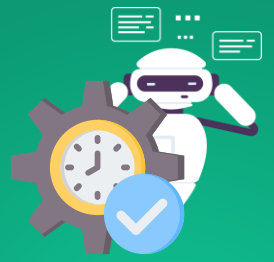
SUB OBJECTIVES

REINFORCEMENT LEARNING MODEL

Develop a reinforcement learning (RL) model tailored for decision-making and path planning in complex traffic environments.

COLLISION AVOIDANCE MECHANISM

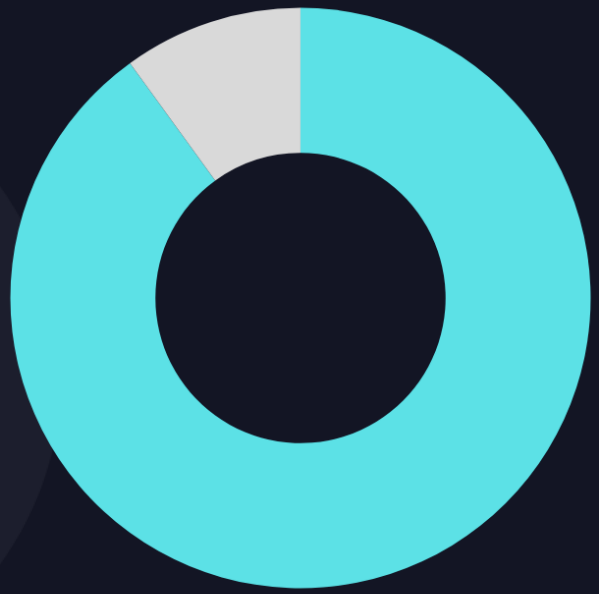
Ensure the mechanism reacts effectively to real-time obstacles, both static and dynamic, while maintaining smooth driving behaviors.



REAL-TIME IMPLEMENTATION AND OPTIMIZATION

Implement the proposed RL-based navigation and collision avoidance system in real-time, addressing challenges like computational efficiency and latency.

Progress



90%

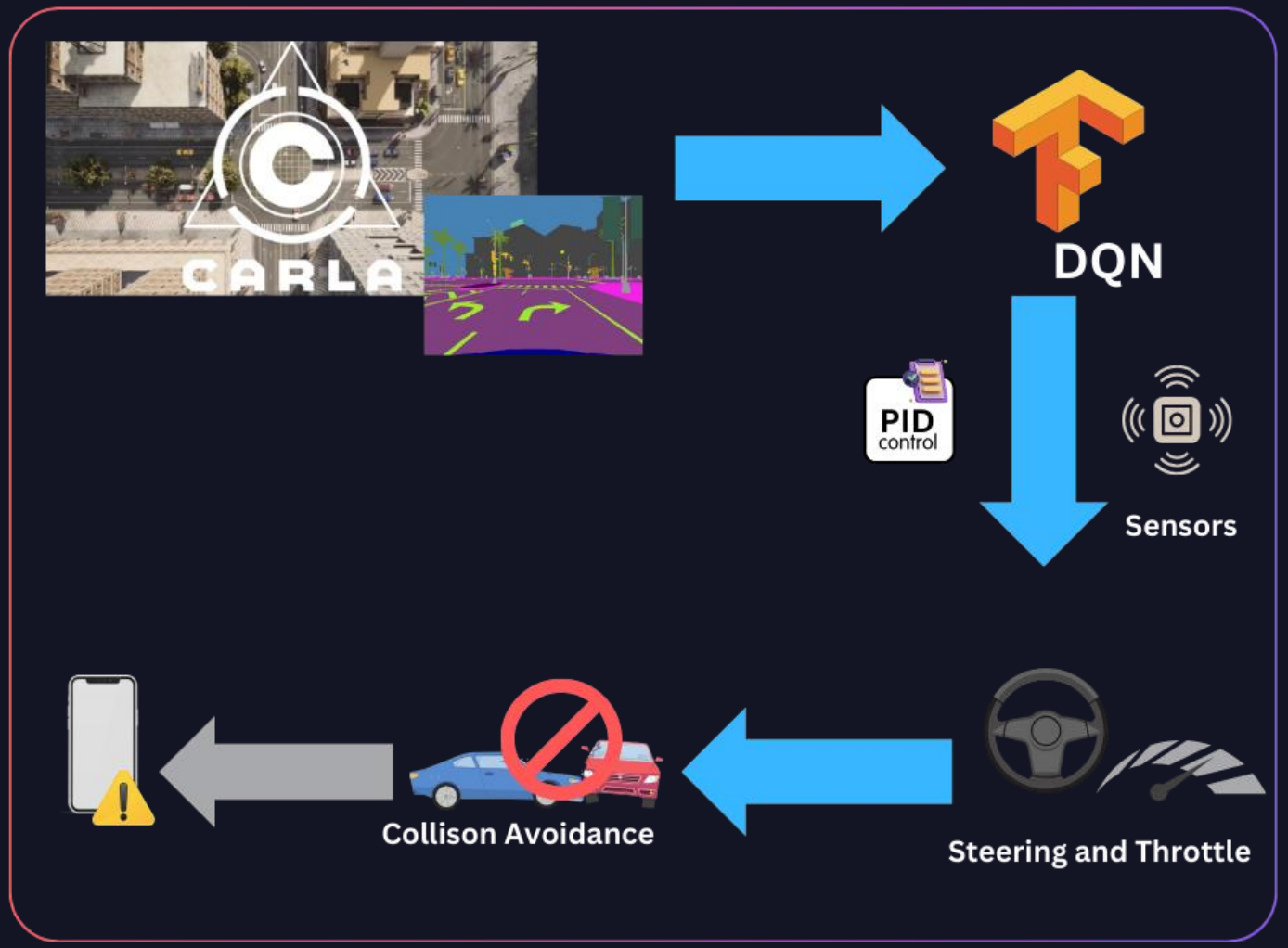
Completed Tasks

- CARLA Environment Setup
- Reinforcement Learning Model
- PID Controller implementation
- Model Integration

Upcoming Tasks

- Alerting System
- Optimize DQN hyperparameters

Progress Diagram



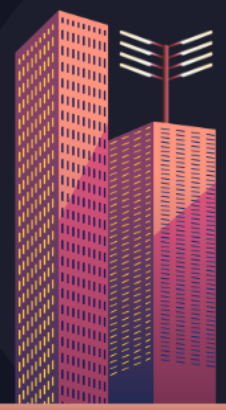
Previous Approach – PPO (Proximal Policy Optimization)

Reinforcement learning algorithm used for continuous action spaces

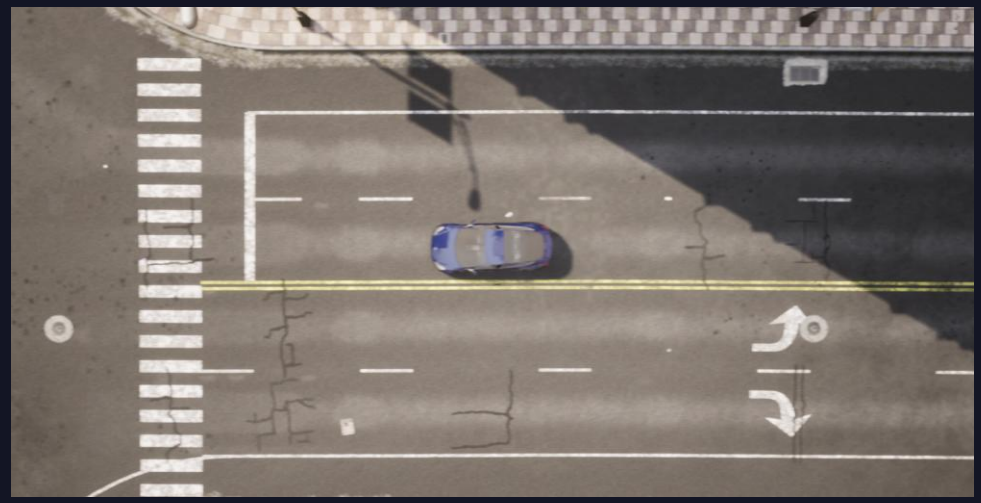
- Computationally expensive.
- Slow convergence in complex environments.
- Difficult to fine-tune for real-time decision-making.

New Approach – DQN (Deep Q-Network)

- More efficient for discrete action spaces.
- Better stability and faster convergence.
- Uses experience replay and target networks for better learning.
- Take less computational power compared to PPO



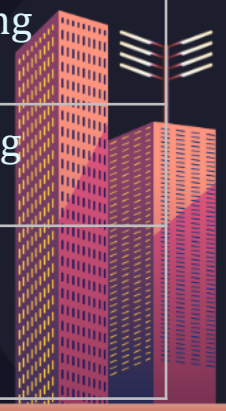
Training Process



```
Episode: 1/40000, Total reward: -3000, Average reward: -3000.0
Weather set to: WeatherParameters(cloudiness=60.000000, precipitation=60.000000, precipitation_deposits=60.000000, wind_intensity=60.000000, sun_azimuth_angle=-1.000
00, sun_altitude_angle=15.000000, fog_density=3.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scattering_
cale=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
Brake - No Preceding Actor
Episode: 2/40000, Total reward: -2820, Average reward: -2910.0
Weather set to: WeatherParameters(cloudiness=60.000000, precipitation=0.000000, precipitation_deposits=50.000000, wind_intensity=10.000000, sun_azimuth_angle=-1.0000
0, sun_altitude_angle=15.000000, fog_density=2.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scattering_s
ale=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
Brake - No Preceding Actor
Episode: 3/40000, Total reward: -3000, Average reward: -2940.0
Weather set to: WeatherParameters(cloudiness=5.000000, precipitation=0.000000, precipitation_deposits=0.000000, wind_intensity=10.000000, sun_azimuth_angle=-1.000000
 sun_altitude_angle=15.000000, fog_density=2.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scattering_sca
e=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
Brake - No Preceding Actor
Episode: 4/40000, Total reward: -2880, Average reward: -2925.0
Weather set to: WeatherParameters(cloudiness=100.000000, precipitation=100.000000, precipitation_deposits=90.000000, wind_intensity=100.000000, sun_azimuth_angle=-1.
00000, sun_altitude_angle=15.000000, fog_density=7.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scatteri
g_scale=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
An error occurred: Spawn failed because of collision at spawn position
Retry in 5s.
```

- The Deep Q-Network (DQN) model is used, which is a value-based reinforcement learning method that learns optimal actions by approximating the Q-value function using a neural network

Reward Function	Condition	Reward Value	Description
Terminal Reward	Collision occurs	-5000	Large penalty for collisions.
	No collisions and episode ends successfully	+1000	Large reward for successful episode completion.
Speeding Reward	No preceding vehicle or safe distance maintained	+60	Reward for maintaining speed without obstacles.
	Preceding vehicle is too close (distance < 12)	-300	Penalty for getting too close to the preceding vehicle.
	Preceding vehicle is at a safe distance (distance ≥ 12)	+30	Reward for maintaining a safe distance.
Brake Reward	No preceding vehicle	-3000	Penalty for braking when no preceding vehicle is present.
	Preceding vehicle is too far (distance > 12)	-1500	Penalty for braking too far from the preceding vehicle.
	Preceding vehicle is too close (distance < 6)	-700	Penalty for braking too close to the preceding vehicle.
	Preceding vehicle is at an optimal distance (6 ≤ distance ≤ 12)	+2000	Reward for braking at an optimal distance.



Testing

```
No Collisons!
Test episode 13
Total reward in test episode 13: 7900
Weather set to: WeatherParameters(cloudiness=5.000000, precipitation=0.000000, precipitation_deposits=50.000000, wind_intensity=10.000000, sun_azimuth_angle=-1.000000
, sun_altitude_angle=45.000000, fog_density=3.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scattering_sca
le=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
Episode time exceeded
No Collisons!
Test episode 14
Total reward in test episode 14: 8020
Weather set to: WeatherParameters(cloudiness=60.000000, precipitation=60.000000, precipitation_deposits=60.000000, wind_intensity=60.000000, sun_azimuth_angle=-1.0000
00, sun_altitude_angle=15.000000, fog_density=3.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scattering_s
cale=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
Episode time exceeded
No Collisons!
Test episode 15
Total reward in test episode 15: 8260
Weather set to: WeatherParameters(cloudiness=60.000000, precipitation=0.000000, precipitation_deposits=50.000000, wind_intensity=10.000000, sun_azimuth_angle=-1.00000
0, sun_altitude_angle=15.000000, fog_density=2.000000, fog_distance=0.750000, fog_falloff=0.100000, wetness=0.000000, scattering_intensity=1.000000, mie_scattering_sc
ale=0.030000, rayleigh_scattering_scale=0.033100, dust_storm=0.000000)
Episode time exceeded
No Collisons!
```

Ln 90, Col 21 Spaces: 4 UTF-8 LF {} Python 3.10.16 ('manthrax')


Maintain Speed
Speed: 0.00
Throttle: 0.80

PID Controller

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\,\mathrm{d}\tau + K_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t},$$

Proportional: Corrects error based on the current deviation.

Integral: Eliminates long-term steady-state error by accounting for past errors.

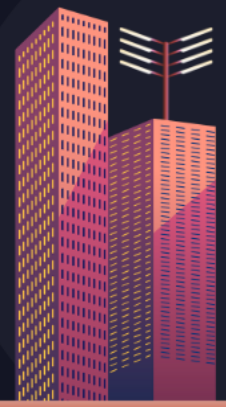
Derivative: Mitigates future error by considering how the error is changing.

- Proportional-Integral-Derivative controller adjusts the steering angle to minimize the deviation from the desired path and control Throttle for fine-tune vehicle behavior, balancing between responsiveness and stability

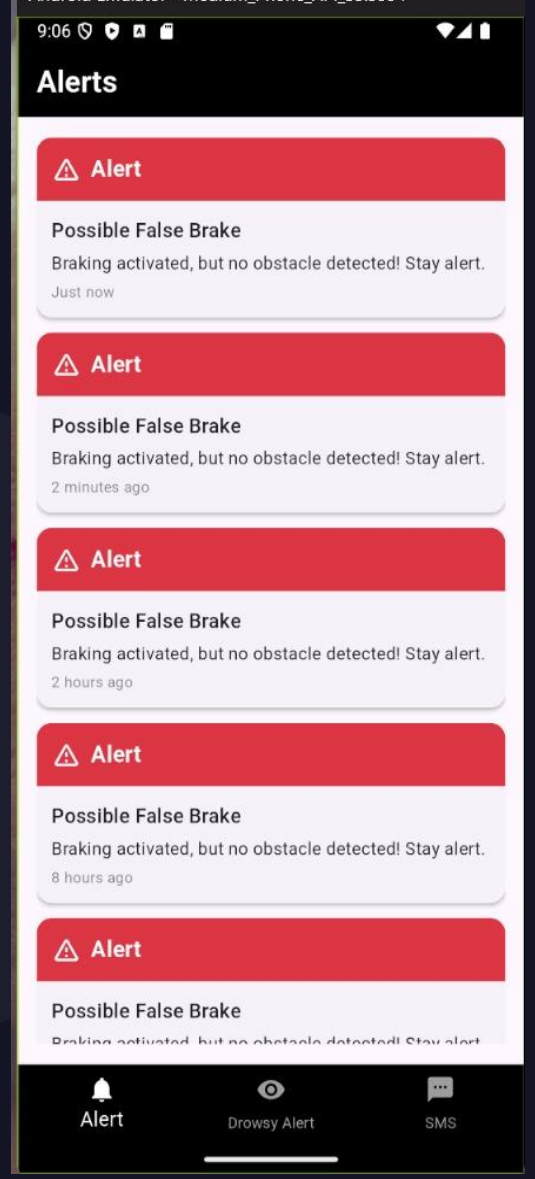
Combining DQN with PID Control

Why Combine with PID?

- PID handles low-level control for smoother acceleration, braking, and steering.
- DQN makes high-level decisions (e.g., avoid obstacles, follow the path).
- How It Works:
  - DQN decides actions based on sensor inputs.
  - PID refines control signals for smooth execution.



Alert System



Purpose:

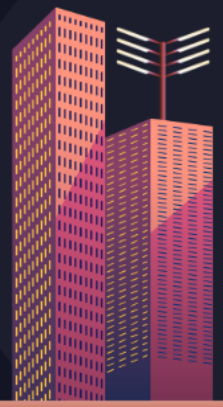
Notify the driver (or monitoring system) about potential collisions.

How It Works:

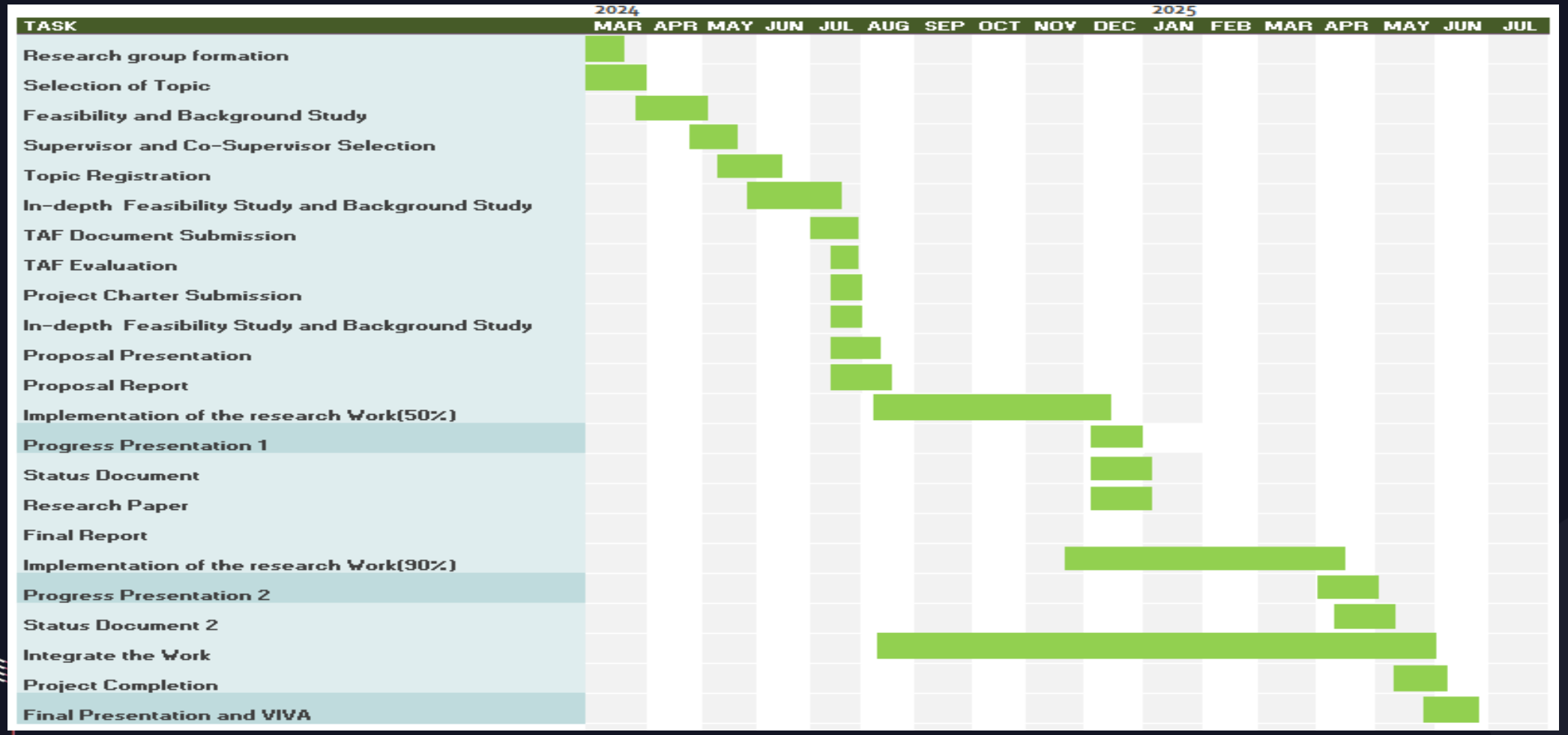
If a risk detected, an alert is sent to the display in vehicle dashboard.

The app displays speed, vehicle behavior, and warnings.

TECHNOLOGIES



GANNT CHART


2024
2025
TASK
MAR APR MAY JUN JUL AUG SEP OCT NOV DEC JAN FEB MAR APR MAY JUN JUL
Research group formation
Selection of Topic
Feasibility and Background Study
Supervisor and Co-Supervisor Selection
Topic Registration
In-depth Feasibility Study and Background Study
TAF Document Submission
TAF Evaluation
Project Charter Submission
In-depth Feasibility Study and Background Study
Proposal Presentation
Proposal Report
Implementation of the research Work(50%)
Progress Presentation 1
Status Document
Research Paper
Final Report
Implementation of the research Work(90%)
Progress Presentation 2
Status Document 2
Integrate the Work
Project Completion
Final Presentation and VIVA

References

[1] C. Liu, S. Lee, S. Varnhagen and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 2017, pp. 174-179, doi: 10.1109/IVS.2017.7995716.

[2[4]R. Emuna, A. Borowsky, and A. Biess, "Deep reinforcement learning for human-like driving policies in collision avoidance tasks of self-driving cars," arXiv preprint arXiv:2006.04218, 2020
[3] Shao, Hao, et al. "Safety-enhanced autonomous driving using interpretable sensor fusion transformer." Conference on Robot Learning. PMLR, 2023..

[4] R. Emuna, A. Borowsky, and A. Biess, "Deep reinforcement learning for human-like driving policies in collision avoidance tasks of self-driving cars," arXiv preprint arXiv:2006.04218, 2020.

[5] Dosovitskiy, Alexey, et al. "CARLA: An open urban driving simulator." Conference on robot learning. PMLR, 2017.

IT21162978|ATHUKORALA W.A.A.D.D.T

BSc (Hons) Degree in Information Technology (specialization in Data Science)

Enhance ethical decision-making by incorporating reinforcement learning-based ethical reasoning alongside driver attention and drowsiness monitoring.

Safety

Fairness

Adaptivity in autonomous driving

Component 03

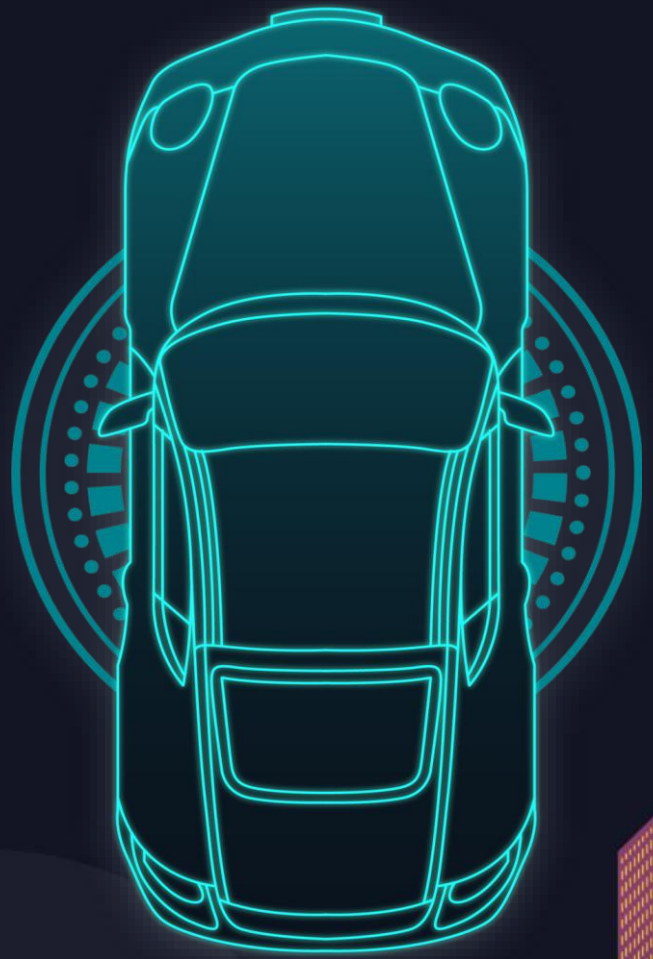
Ethical Decision Making in Autonomous Vehicles

RESEARCH GAP

Features	Research 1	Research 2	Research 3	Research 4	Research 5	MANTHRA-X
Personalized Ethical Decision-Making	No	No	Yes	No	No	
Integration of Real-Time Driver Attention and Drowsiness Monitoring	No	No	Yes	No	Yes	
Sophisticated Object Prioritization	No	No	No	Limited	No	
Ethical Framework Consistency Across Scenarios	Yes	Yes	Yes	No	Yes	
RL based Decision-Making Approaches	No	No	No	Yes	Yes	

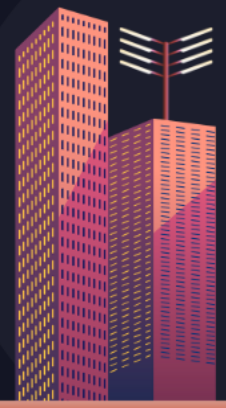
SPECIFIC OBJECTIVE

To develop a comprehensive ethical decision-making system for autonomous vehicles that prioritizes human safety, incorporates driver attention, and adapts to dynamic environments, with a focus on ethical frameworks, object prioritization and ethical considerations.



SUB OBJECTIVES

- Eyeball Tracking System for Attention Monitoring
- Implement ethical decision-making in autonomous vehicles using reinforcement learning.
- Ensure model transparency by balancing safety, driver preferences, and ethical considerations.



SUB OBJECTIVES

- **Implement ethical decision-making in autonomous vehicles using RL.**

Model Used: Deep Q-Network (DQN) forRL

Training Process:

- Perception: Agent collects sensor data from the environment.
- Decision-Making: RL model evaluates scenarios and selects the safest action.
- Reward System: Reinforces ethical and safe driving behavior.

Actions: 9 predefine actions

```python
class CarlaEnv:
    def convert_action_index_to_control(self, action):
        """
        Map the action index to vehicle control commands.

        Parameters:
        action (int): Action index.
        """
        if action == 1:    # Keep going
            self.ego.apply_control(carla.VehicleControl(throttle=0.0, steer=0.0, brake=0.0))
        elif action == 2:  # Brake
            self.ego.apply_control(carla.VehicleControl(throttle=0.0, steer=0.0, brake=1.0))
        elif action == 3:  # Accelerate
            self.ego.apply_control(carla.VehicleControl(throttle=1.0, steer=0.0, brake=0.0))
        elif action == 4:  # Turn left
            self.ego.apply_control(carla.VehicleControl(throttle=0.0, steer=-0.8, brake=0.0))
        elif action == 5:  # Turn left and accelerate
            self.ego.apply_control(carla.VehicleControl(throttle=1.0, steer=-0.8, brake=0.0))
        elif action == 6:  # Turn left and brake
            self.ego.apply_control(carla.VehicleControl(throttle=0.0, steer=-0.8, brake=1.0))
        elif action == 7:  # Turn right
            self.ego.apply_control(carla.VehicleControl(throttle=0.0, steer=0.8, brake=0.0))
        elif action == 8:  # Turn right and accelerate
            self.ego.apply_control(carla.VehicleControl(throttle=1.0, steer=0.8, brake=0.0))
        elif action == 9:  # Turn right and brake
            self.vehicle.apply_control(carla.VehicleControl(throttle=0.0, steer=0.8, brake=1.0))
```

Ethical Decision Scenarios

- Collision Avoidance Dilemma
- Pedestrian vs. Passenger Safety
- Object Prioritization

SUB OBJECTIVES

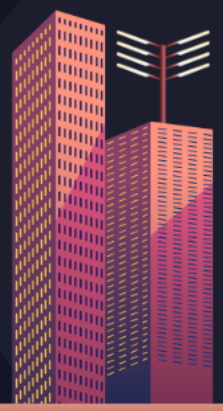
k value (Altruism vs. Egoism)

```
parser.add_argument('--knob-value', type=float, default=0.1, help='k ∈ {0.1, 0.5, 0.9}')
```

- ✓ The k-value in model represents a **trade-off between** altruism (concern for others) and egoism (self-preservation).
- ✓ The k-value (Altruism vs. Egoism trade-off) is used in the reward function to balance ethical decision-making.

- ❑ Lower k-values (0.1): Prioritizes pedestrian safety over vehicle self-preservation.
- ❑ Higher k-values (0.9): Focuses on protecting the vehicle and its passengers.

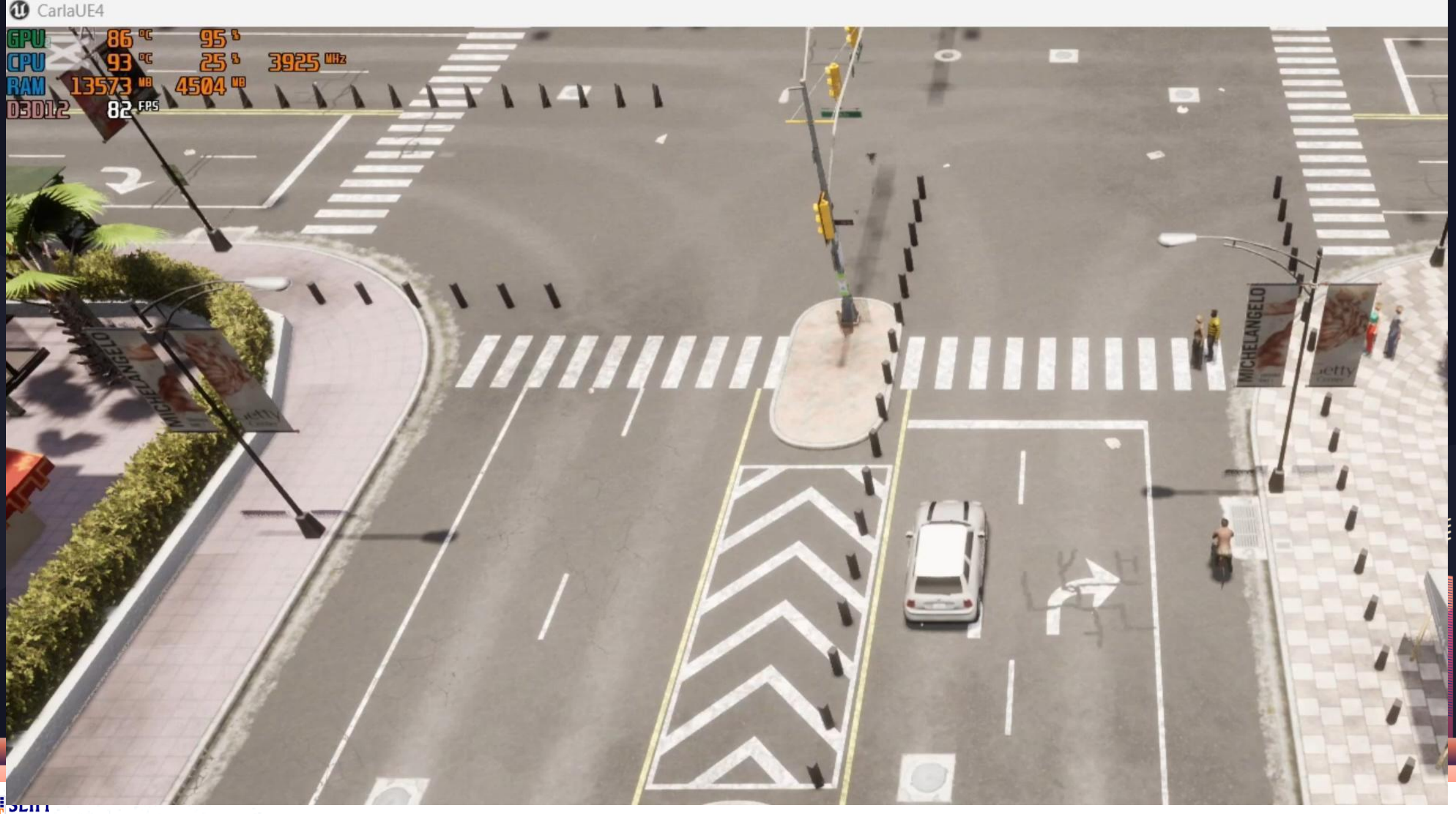
Maintaining Model Transparency & Balancing Decisions



Reward and Penalty

- Rewards List:
  - Step Reward: Reward based on the probability of injury, vehicle dynamics, and braking/acceleration conditions.
  - Safe Episode Completion Reward: +200 for successfully crossing the intersection without any collisions.
- Penalties List:
  - Collision Penalty: -400 * (Sum of injury probabilities of ego vehicle and others). Applied when a collision occurs.
  - Braking/Acceleration Penalty: Reduction in step reward when extreme braking or acceleration is applied.
  - Lateral Acceleration Penalty: Penalty applied if lateral acceleration exceeds the allowed threshold.
  - Excessive Speed Penalty: Higher injury probability leading to greater negative rewards if the vehicle speed increases.
  - Time Limit Penalty: Episode termination if the maximum time is exceeded.

```
Testing Scenarios

NLL: 0.3467050338848517, RMSE: 0.022930994629859924
NLL: 0.34658475360728974, RMSE: 0.006686224602162838
NLL: 0.34673997102341514, RMSE: 0.0257975235581398
Test episode 1 - Collision with: 5
Total reward in test episode 1: -19.33472135345529
NLL: 0.34720099329215864, RMSE: 0.05009659007191658
NLL: 0.34678002320228607, RMSE: 0.028737016022205353
NLL: 0.346619673935614, RMSE: 0.013576589524745941
something wierd!
Test episode 2 - Collision with: 0
Total reward in test episode 2: -78.29360167916937
Spawn failed, Retrying in 5s. Please check server connection and whether simulator is running.
Spawn failed, Retrying in 5s. Please check server connection and whether simulator is running.
NLL: 0.3468437681458888, RMSE: 0.03287503123283386
NLL: 0.34675004989080294, RMSE: 0.026567554101347923
NLL: 0.347010367977668, RMSE: 0.04179912433028221
Test episode 3 - Collision with: 2
Total reward in test episode 3: 0.44525385413140484
NLL: 0.346648080434760 9, RMSE: 0.01726336032152176
NLL: 0.3471109934190 3195, RMSE: 0.04636380076408386
```


CarlaUE4
GPU 86 °C 95 %
CPU 93 °C 25 % 3925 MHz
RAM 13573 MB 4504 MB
D3D12 82 FPS
MICHELANGELO

SUB OBJECTIVES

Quick Recap

➢ Eyeball Tracking System for Attention Monitoring

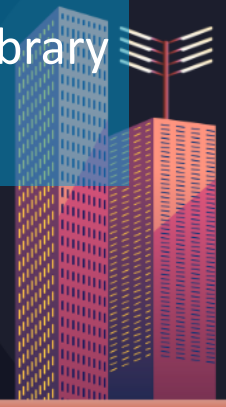
- To monitor the driver's gaze direction to assess if they are paying attention to the road.
- To detect eye closures and drowsiness in real time.
- To provide actionable feedback (e.g., warnings) to ensure driver focus and safety.

Key Metrics:

- Gaze Ratios: Determine the driver's focus area.
- EAR (Eye Aspect Ratio): Measure eye openness to detect closure and drowsiness.

Methods and Technologies

- MediaPipe Face Mesh - library
- OpenCV



SUB OBJECTIVES

Driver Gaze and
Head Position Detection

```python
# Initialize MediaPipe Face Mesh
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(
    max_num_faces=1,
    refine_landmarks=True,
    static_image_mode=False,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
)

# Indices for key landmarks on the iris and eye corners
left_iris_index = 468
left_eye_inner = 133
left_eye_outer = 33
left_eye_top = 159
left_eye_bottom = 145

right_iris_index = 473
right_eye_inner = 362
right_eye_outer = 263
right_eye_top = 386
right_eye_bottom = 374
```

```python
# Function to calculate gaze direction
def is_concentrated(face_landmarks, frame_width, frame_height):
    # Get positions of left and right iris centers

    left_rye_z = face_landmarks.landmark[left_iris_index].z
    right_iris_z = face_landmarks.landmark[right_iris_index].z

    if abs(left_rye_z - right_iris_z) > 0.4: # head pose threshold
        return False

    left_eye_y = face_landmarks.landmark[left_eye_outer].y * frame_height
    right_eye_y = face_landmarks.landmark[right_eye_outer].y * frame_height
    eye_y_difference = abs(left_eye_y - right_eye_y)

    if eye_y_difference > 40: # head tilt threshold
        return False

    left_iris_x = face_landmarks.landmark[left_iris_index].x * frame_width
    right_iris_x = face_landmarks.landmark[right_iris_index].x * frame_width

    # Get eye corner positions
    left_eye_inner_x = face_landmarks.landmark[left_eye_inner].x * frame_width
    left_eye_outer_x = face_landmarks.landmark[left_eye_outer].x * frame_width
    right_eye_inner_x = face_landmarks.landmark[right_eye_inner].x * frame_width
    right_eye_outer_x = face_landmarks.landmark[right_eye_outer].x * frame_width

    # Calculate relative positions for gaze detection
    left_gaze_ratio = (left_iris_x - left_eye_inner_x) / (left_eye_outer_x - left_eye_inner_x)
    right_gaze_ratio = (right_iris_x - right_eye_inner_x) / (right_eye_outer_x - right_eye_inner_x)

    # Check if both irises are centered within each eye (approx. 0.5 indicates centered)

    if abs(left_gaze_ratio - right_gaze_ratio) < 0.14: # eyeball gaze threshold
        return True
```

SUB OBJECTIVES

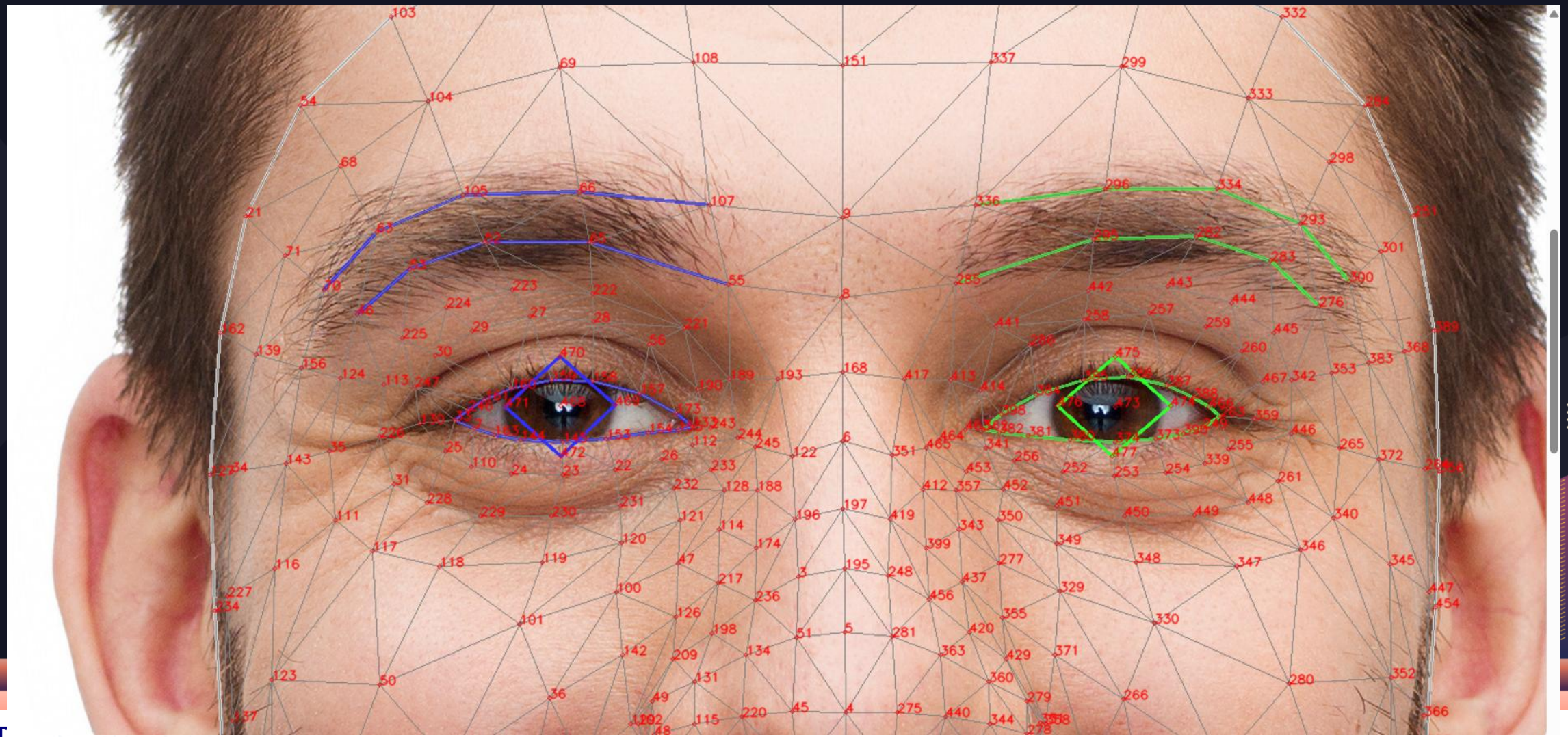
Eye Closure Detection

```python
# Initialize MediaPipe Face Mesh
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(
    max_num_faces=1,
    refine_landmarks=True,
    static_image_mode=False,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
)

# Indices for key eye landmarks
left_eye_top = 159
left_eye_bottom = 145
left_eye_left = 133
left_eye_right = 33

right_eye_top = 386
right_eye_bottom = 374
right_eye_left = 362
right_eye_right = 263
```

```python
# If landmarks are detected, calculate EAR for each eye
if results.multi_face_landmarks:
    for face_landmarks in results.multi_face_landmarks:
        # Calculate EAR for each eye
        left_ear = calculate_ear(left_eye_top, left_eye_bottom, left_eye_left, left_eye_right, face_landmarks, frame_width, frame_height)
        right_ear = calculate_ear(right_eye_top, right_eye_bottom, right_eye_left, right_eye_right, face_landmarks, frame_width, frame_height)

        # Check if both eyes are closed
        if left_ear < EAR_THRESHOLD and right_ear < EAR_THRESHOLD:
            if eyes_closed_start_time is None:  # Start timing if eyes just closed
                eyes_closed_start_time = time.time()
            else:
                closed_duration = time.time() - eyes_closed_start_time
                cv2.putText(frame, f"Eyes Closed for: {closed_duration:.1f}s", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)

                # Check if closed time exceeds threshold
                if closed_duration > 2.0 and not warning_displayed:
                    color =  (0, 0, 255)
                    warning_displayed = True

                if warning_displayed:
                    cv2.putText(frame, "WARNING", (10, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
        else:
            # Reset timer and warning flag when eyes are open
            eyes_closed_start_time = None
            warning_displayed = False
            color =  (0, 255, 0)
```

SUB OBJECTIVES

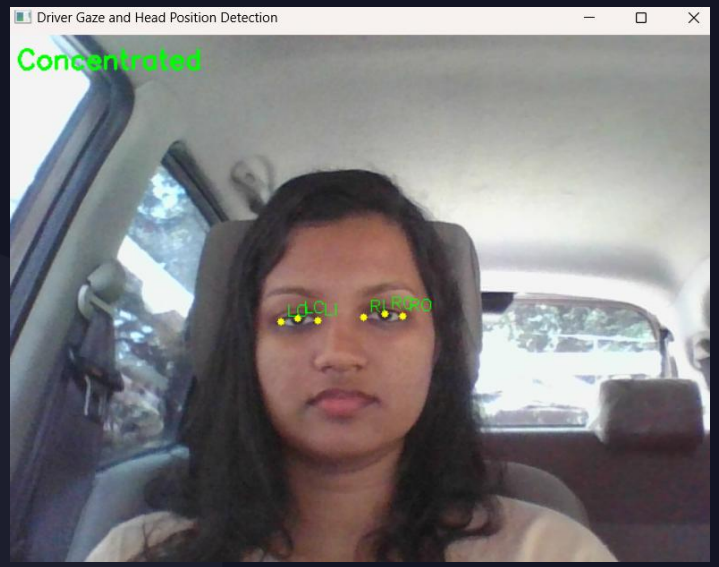
Face landmark detection guide | Google AI Edge | Google AI for Developers

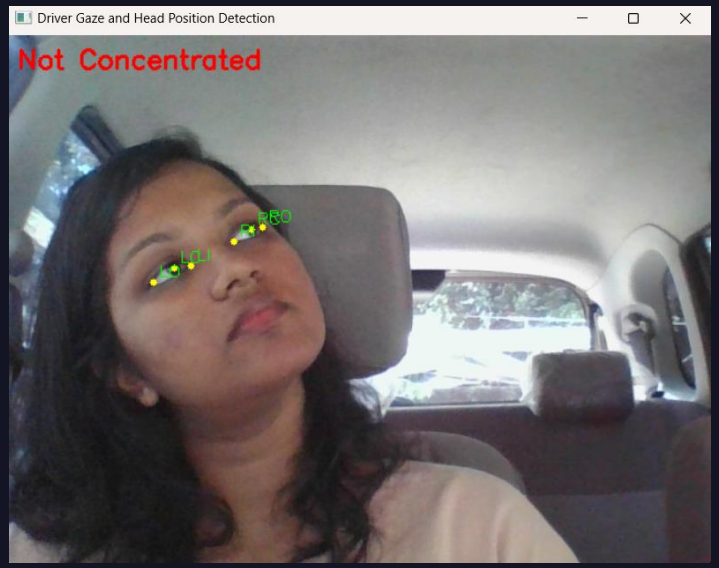


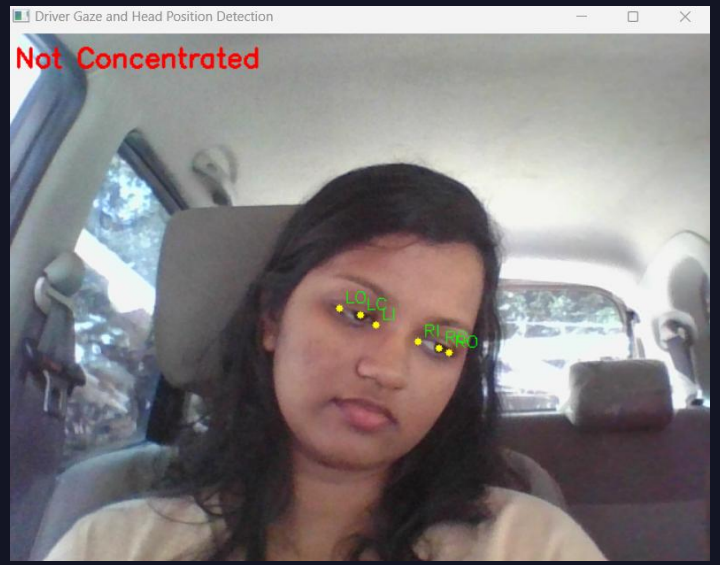
Testing

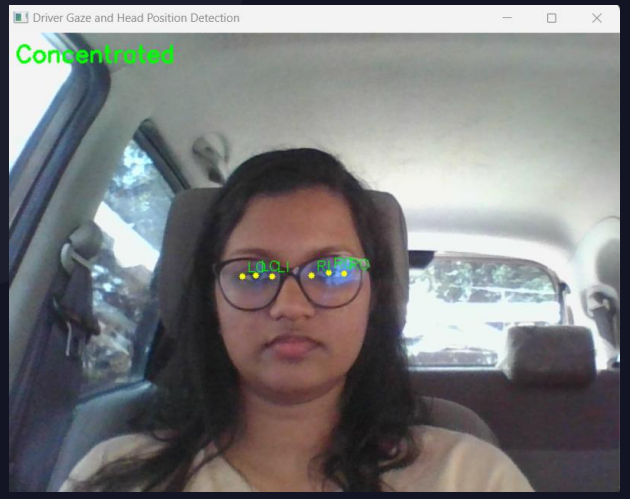
SUB OBJECTIVES

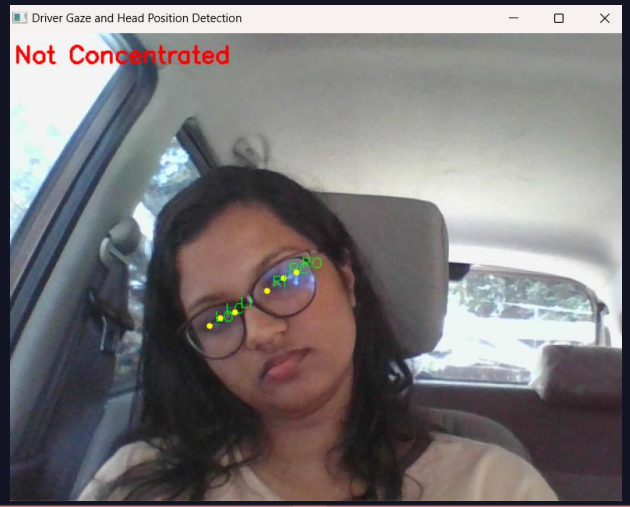
Driver Gaze and Head Position Detection

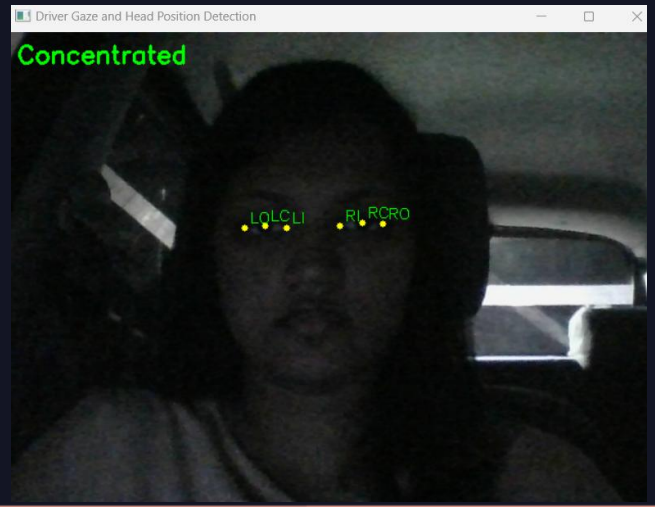


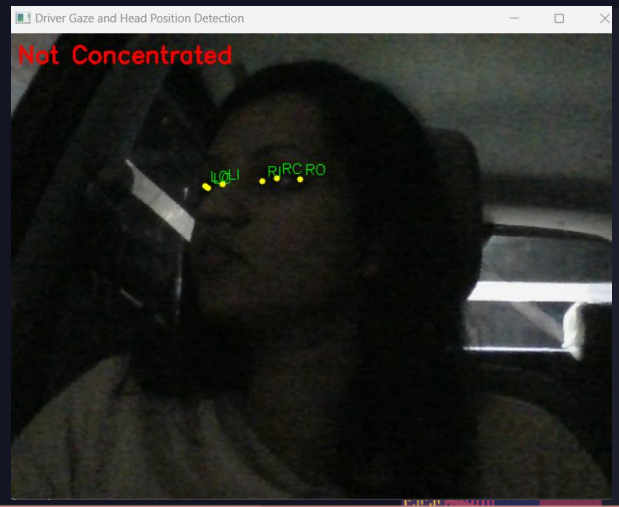








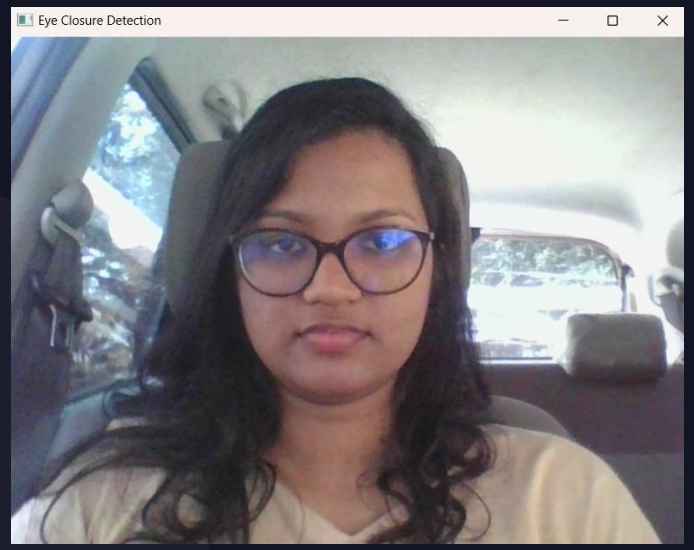


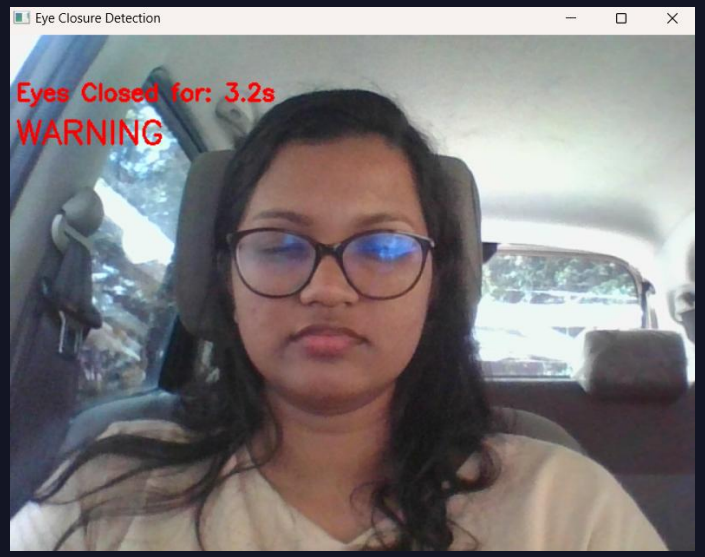


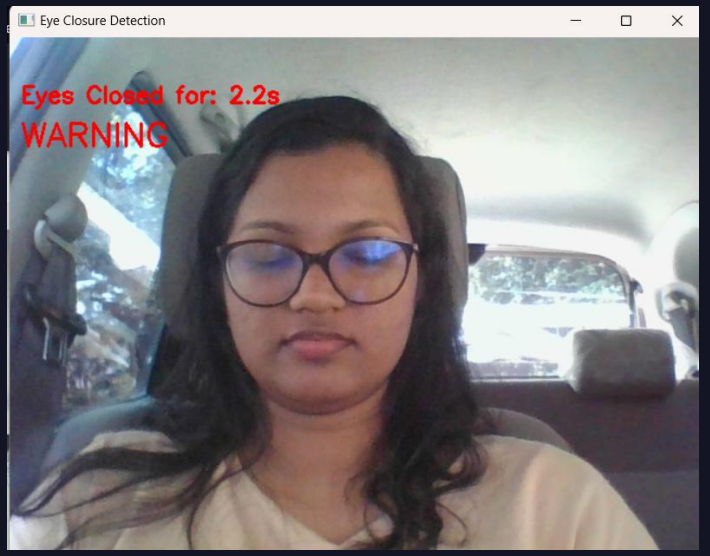
Testing

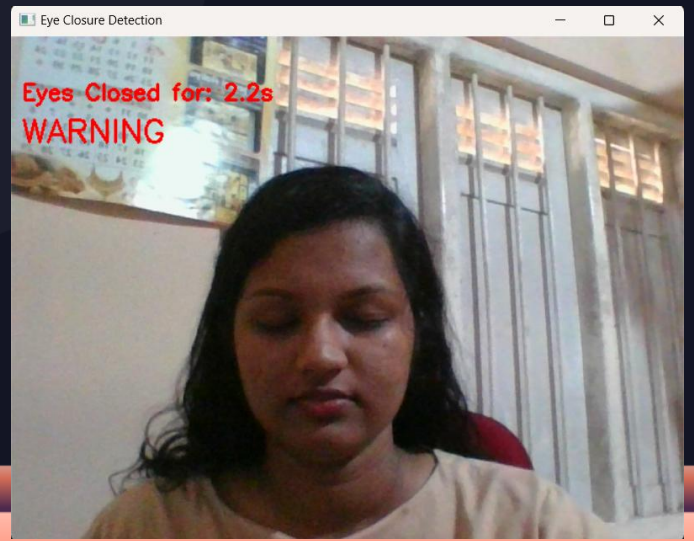
SUB OBJECTIVES

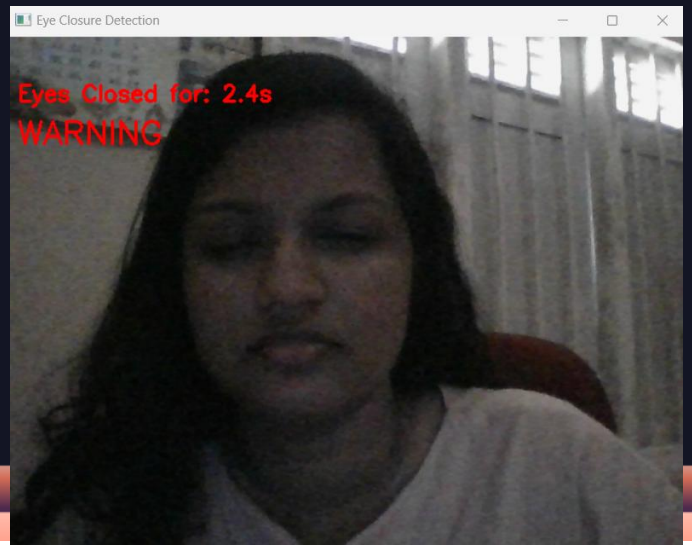
Eye Closure Detection







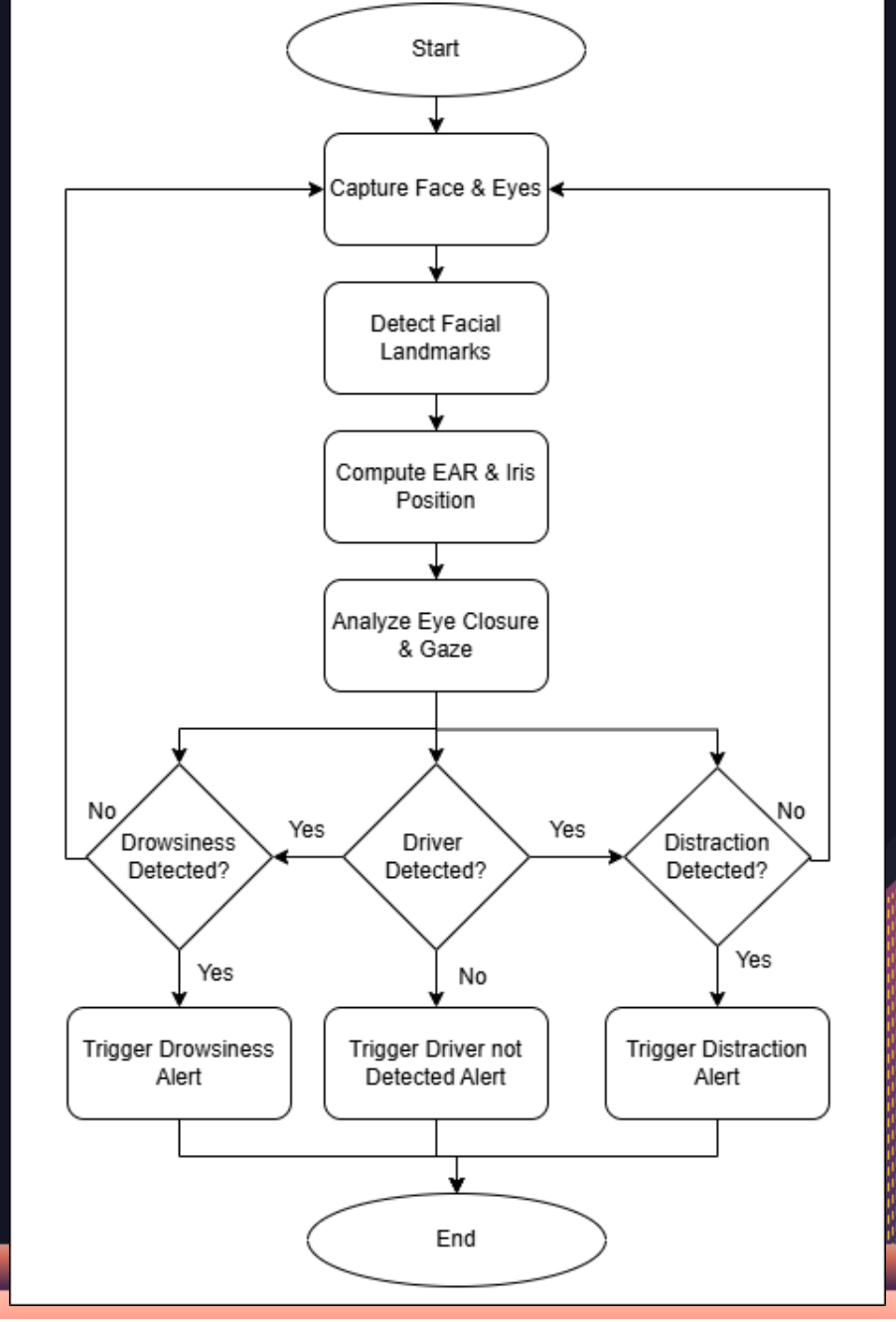




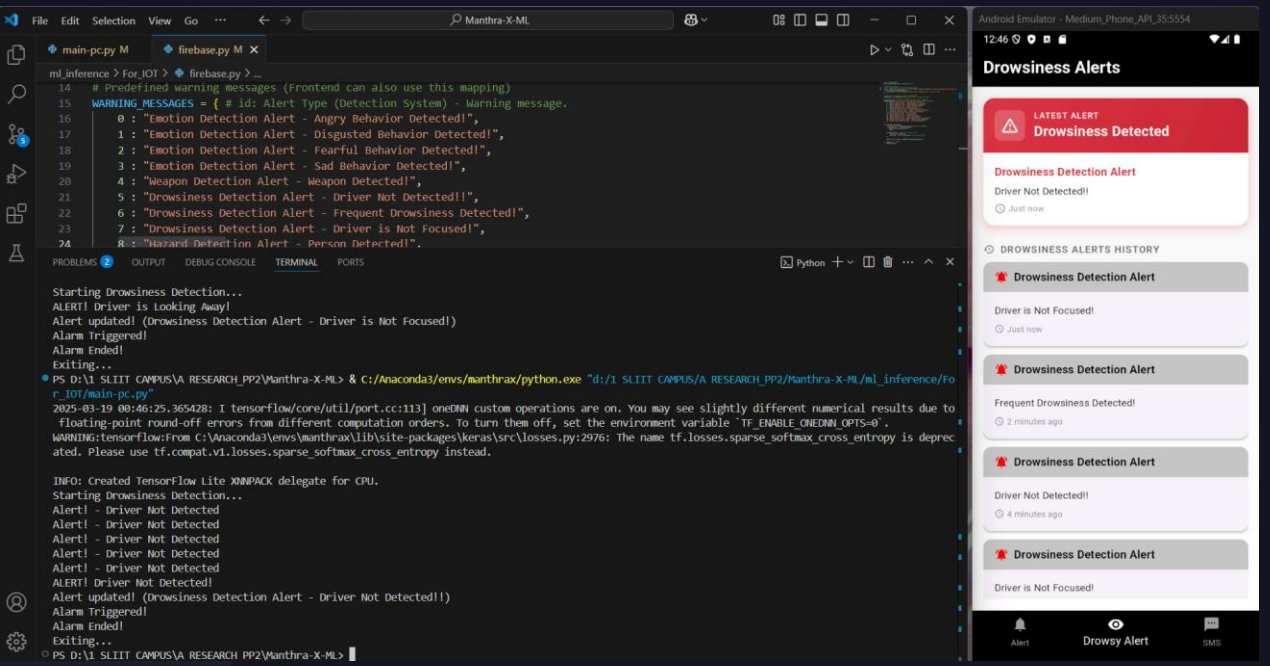
SUB OBJECTIVES

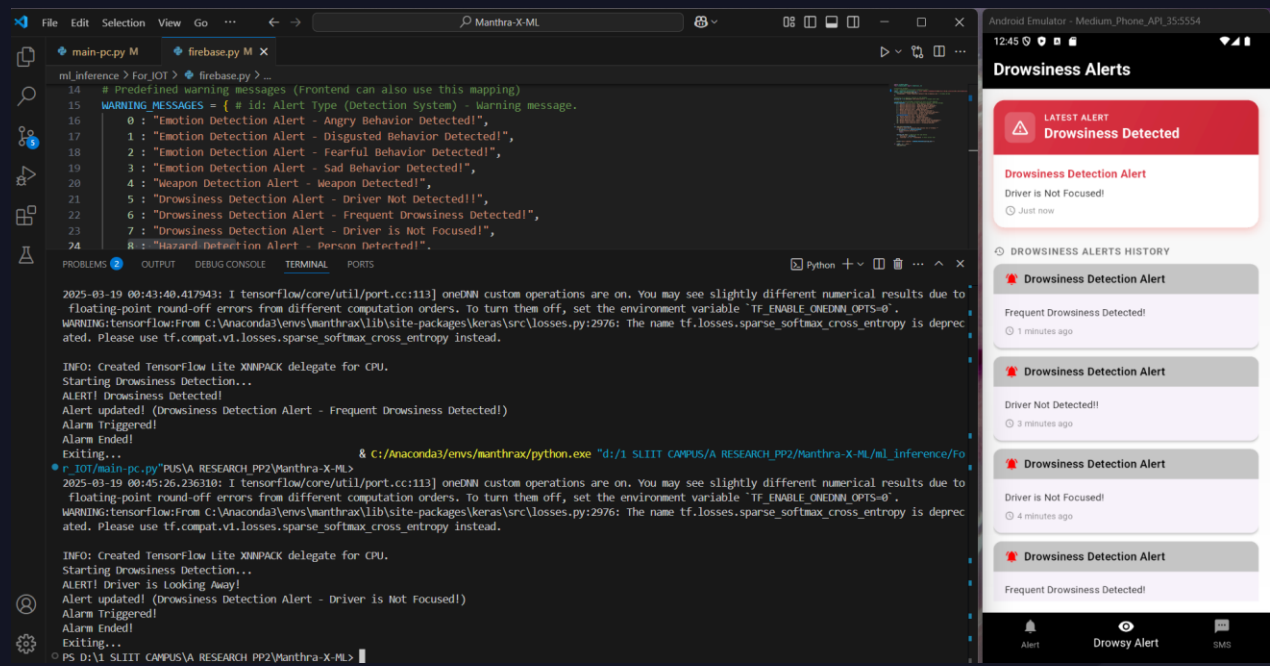
```python
def detect_drowsiness(frame):
    """
    Output:
        0: Driver Not Detected
        1: Drowsiness
        2: Driver is looking away
    """
```

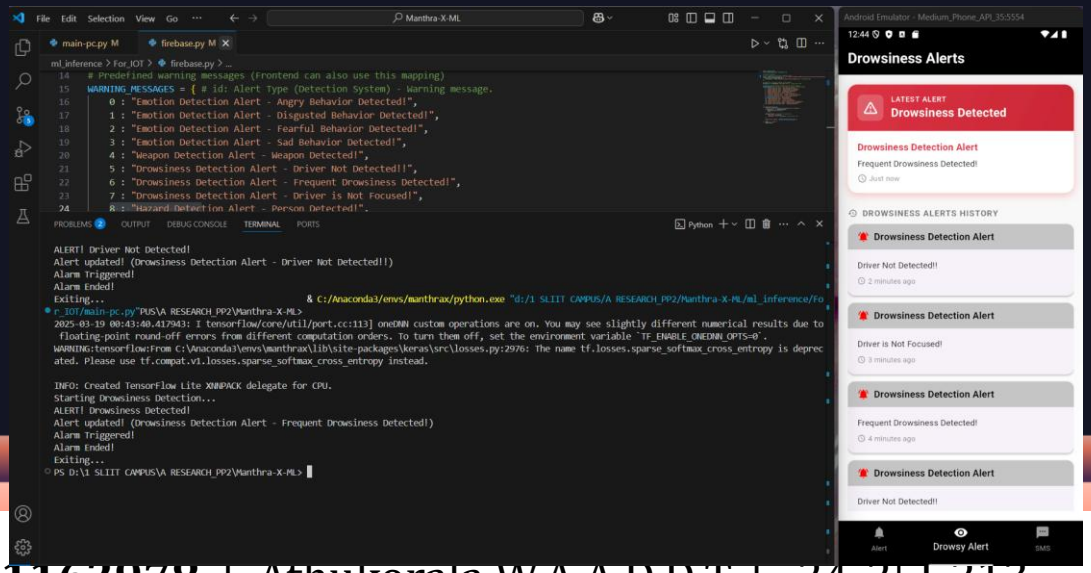
```python
# Dictionary to track consecutive occurrences of each status
status_counter = {0: 0, 1: 0, 2: 0}
threshold_counts = {0:5, 1:5, 2:10}  # Trigger alert if status happens 10 times in a row
```



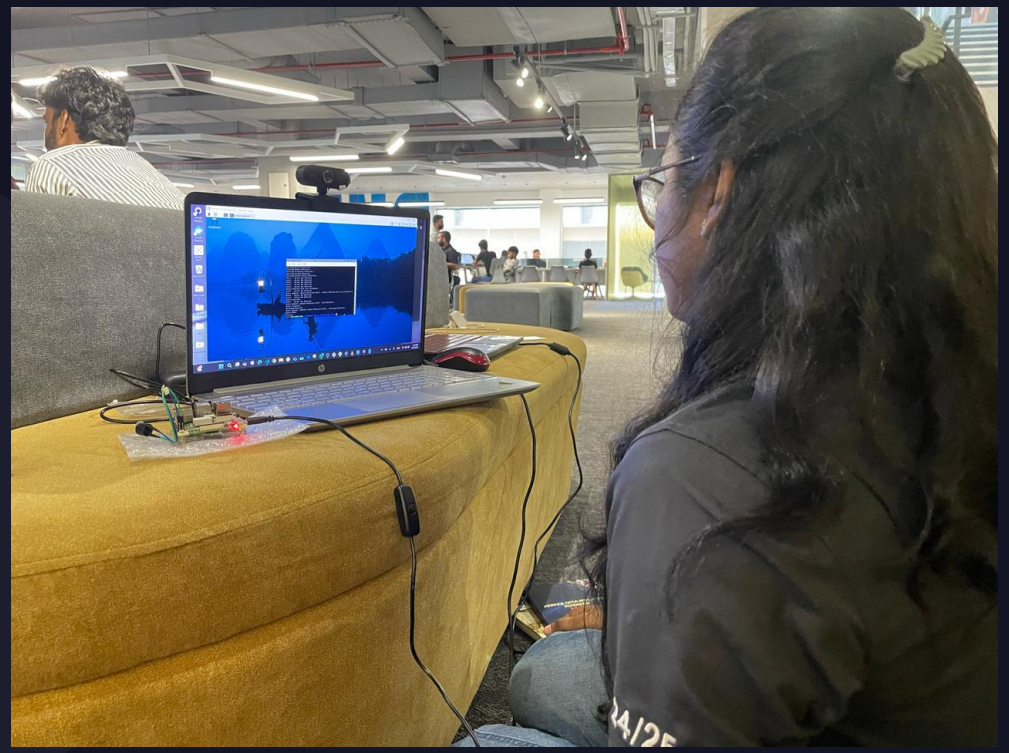
Testing SUB OBJECTIVES With Mobile APP

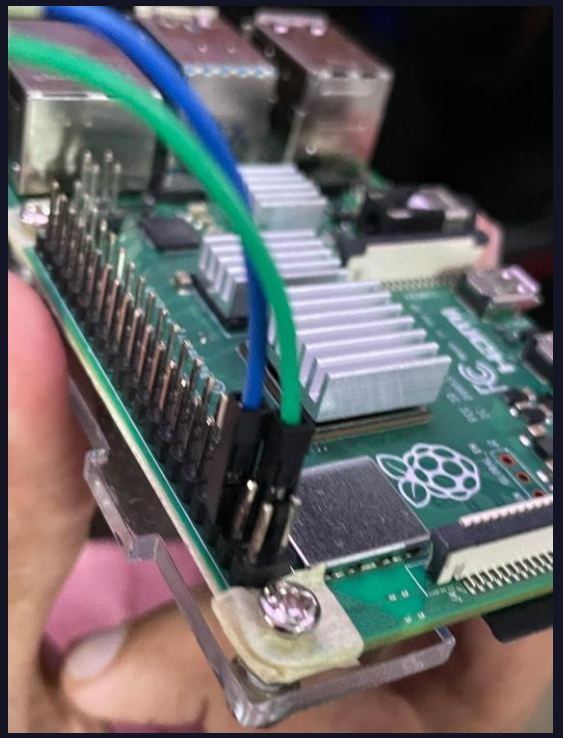






SUB OBJECTIVES





IOT

Why Choose ?

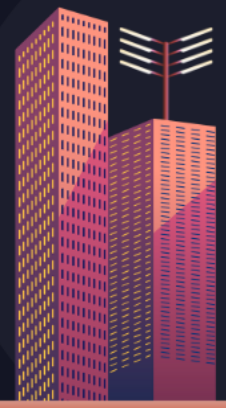
MediaPipe Face Mesh

- Pre-Trained Model
- Accuracy
- Efficiency
- Adaptability

OpenCV

- Real-Time Video Processing
- Flexibility
- Works seamlessly with MediaPipe

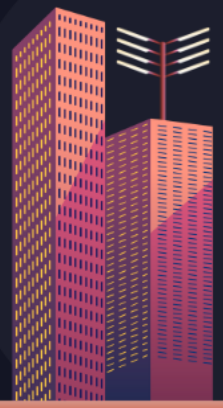
Scalability
Customizable
Cost-Effective Solution



Remaining Tasks

- Mobile Application-Based Alerting System - Drowsy Alert Vibration.
- SMS system.
- Add separate camera for Driver Monitoring
- Testing.

90%
Completed



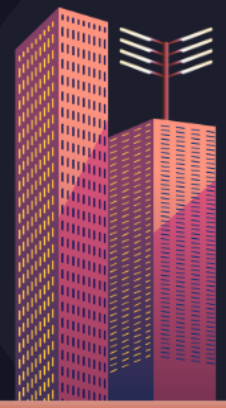
METHODOLOGY

SYSTEM DIAGRAM

REQUIREMENT SPECIFICATION

- System Requirements
- Software Requirements
- Personal Requirements

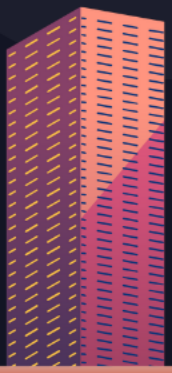
TECHNOLOGIES TO BE USED



REQUIREMENT SPECIFICATION

System Requirements

- **High-performance Computing (HPC) Cluster or Workstations:** For processing complex algorithms related to ethical decision-making and real-time data analysis.
- **High-definition Cameras**: For capturing the driver's attention and external environment.
- **High-speed Internet Connection:** For real-time data transfer, communication with external systems, and updates.



REQUIREMENT SPECIFICATION

Software Requirements

- **Development Tools:**
  IDEs: VSCode, PyCharm, Jupyter Notebook
  Version Control: Git(GitHub, GitLab)

- **Frameworks and Libraries:**
  AI/ML: Tensorflow, Pytorch, Keras, Media pipe Face mesh

  Other : OpenCV

- **Cloud Services:**
  Microsoft Azure for Computing, storage, and machine Learning

- **Databases:**
  NoSQL Databases: MongoDB, Cassandra

- **Simulation Frameworks:**
  CARLA

REQUIREMENT SPECIFICATION

Personal Requirements

- Technical Proficiency
- Analytical Thinking
- Time Management Skills
- Commitment to Learning
- Teamwork and Communication
- Project Management
- Adaptability and Flexibility

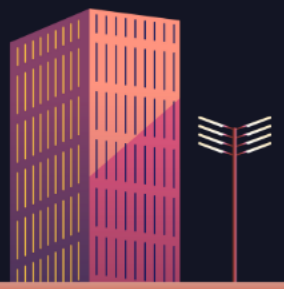
TECHNOLOGIES TO BE USED

AI/ML Freameworks:
Tensorflow, Pytorch, Keras

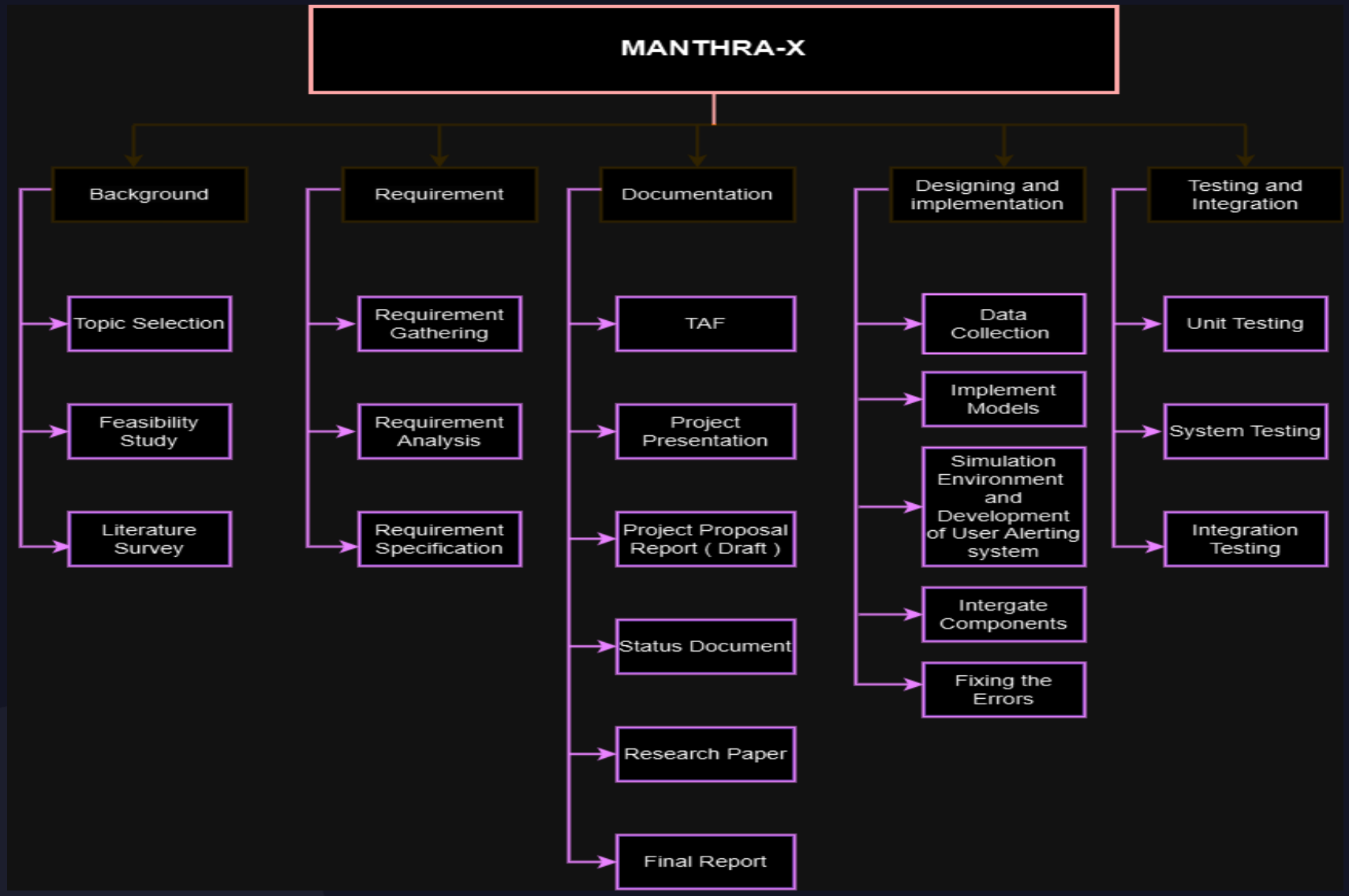
Development Tools:
IDEs: VSCode, PyCharm,
Jupyter Notebook

Simulation Frameworks:
CARLA, AirSim, or Unity

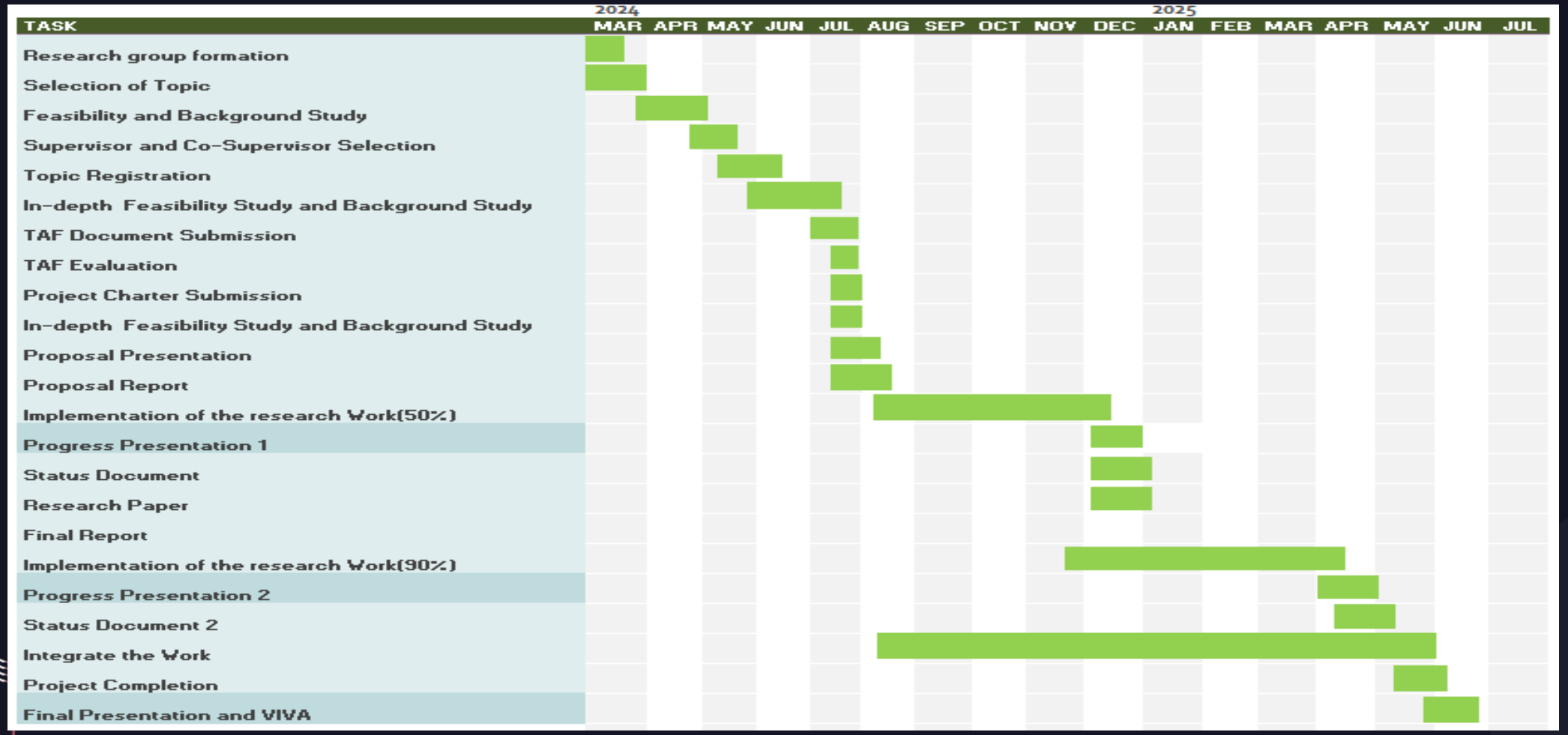
Web Frameworks: Node.js,
Flask, Django



WORK BREAKDOWN CHART



GANNT CHART


2024
2025
TASK
MAR APR MAY JUN JUL AUG SEP OCT NOV DEC JAN FEB MAR APR MAY JUN JUL
Research group formation
Selection of Topic
Feasibility and Background Study
Supervisor and Co-Supervisor Selection
Topic Registration
In-depth Feasibility Study and Background Study
TAF Document Submission
TAF Evaluation
Project Charter Submission
In-depth Feasibility Study and Background Study
Proposal Presentation
Proposal Report
Implementation of the research Work(50%)
Progress Presentation 1
Status Document
Research Paper
Final Report
Implementation of the research Work(90%)
Progress Presentation 2
Status Document 2
Integrate the Work
Project Completion
Final Presentation and VIVA

REFERENCES

- G. Goodall, "Incorporating Ethical Considerations Into Automated Vehicle Control," IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 3, pp. 63-71, Fall 2017, doi: 10.1109/MITS.2017.2717543.

Incorporating Ethical Considerations Into Automated Vehicle Control | IEEE Journals & Magazine | IEEE Xplore

- Y. Shi, X. Ma, and B. Xu, "Decision-Making Technology for Autonomous Vehicles: Learning-Based Methods, Applications and Future Outlook," IEEE Transactions on Vehicular Technology, vol. 69, no. 9, pp. 9245-9261, Sept. 2020, doi: 10.1109/TVT.2020.3014587.

Decision-Making Technology for Autonomous Vehicles: Learning-Based Methods, Applications and Future Outlook | IEEE Conference Publication | IEEE Xplore

- A. Pan, J. Li, and Y. Wang, "Large Language Models for Autonomous Driving: Real-World Experiments," in Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 12345-12354, doi: 10.1109/CVPR52688.2022.01234.

Large Language Models for Autonomous Driving: Real-World Experiments (arxiv.org)

- P. Lin, K. Abney, and G. Bekey, "Ethical Decision Making in Autonomous Vehicles: The AV Ethics Project," IEEE Intelligent Systems, vol. 30, no. 5, pp. 20-26, Sept.-Oct. 2015, doi: 10.1109/MIS.2015.60.

Ethical Decision Making in Autonomous Vehicles: The AV Ethics Project | Science and Engineering Ethics (springer.com)

- J. Wang, S. Liu, and Y. Zhang, "Emotion Detection and Face Recognition of Drivers in Autonomous Vehicles in IoT Platform," IEEE Internet of Things Journal, vol. 7, no. 3, pp. 1858-1869, Mar. 2020, doi: 10.1109/JIOT.2020.2963784.

Emotion detection and face recognition of drivers in autonomous vehicles in IoT platform - ScienceDirect

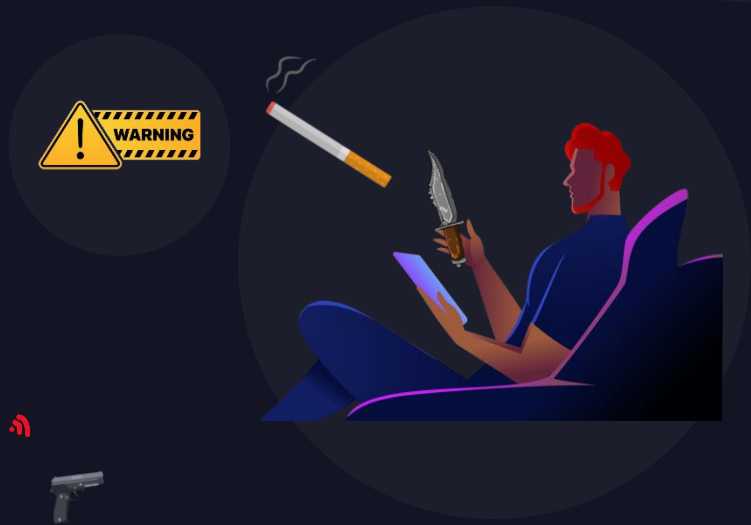
IT21174780|D.M.M.I.T.DISSANAYAKA

BSc (Hons) Degree in Information Technology (specialization in Data Science)

Enhancing Autonomous Vehicle Safety in cabin

Background

Identifying and monitor security risks such as weapons and unauthorized items in the car cabin.



- Image Recognition: Use deep learning models to process real-time camera feeds and detect unauthorized items.
- Voice Analysis: Analyze audio inputs to identify suspicious or unauthorized behaviors through voice patterns.
- **Facial Emotion Detection**: Implement facial emotion detection to identify emotional cues from both attackers and passengers, helping to assess threat levels based on expressions of aggression, stress, or fear.

Research Gap

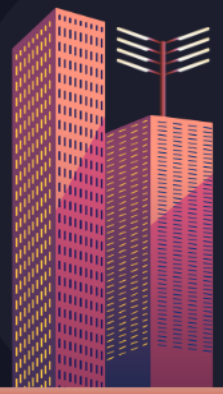
Enhancing Low-Light Resolution for In-Cabin Security in Autonomous Vehicles

Addressing Voice Overlap Challenges in Multi-Occupant Autonomous Vehicle Cabins

Enhancing the speed and accuracy of real-time data processing

False Positives and False Negatives

There is a need for systems that ensure privacy while still providing effective monitoring.



Competition comparison

Features	In-Cabin Monitoring for Avs[6]	sensor fusion in autonomous vehicles[4]	Autonomous Vehicles [5]	Security strategy for autonomous vehicle[7]	MANTHRA-X (proposed System)
Integration of Image , Facial emotion detection and Voice Recognition	✗	✗	✗	✗	✓
False Positive/Negative Reduction in Security Systems	✗	✓	✗	✗	✓
Advanced Convolutional Neural Networks for Vehicle Security	✓	✓	✗	✓	✓
Voice Tone and sounds in Security Systems	✗	✗	✗	✗	✓
Ethical and Privacy Considerations	✓	✗	✓	✗	✓

Research Problem

How to ensure accurate and reliable detection in autonomous vehicles to prevent security breaches ?

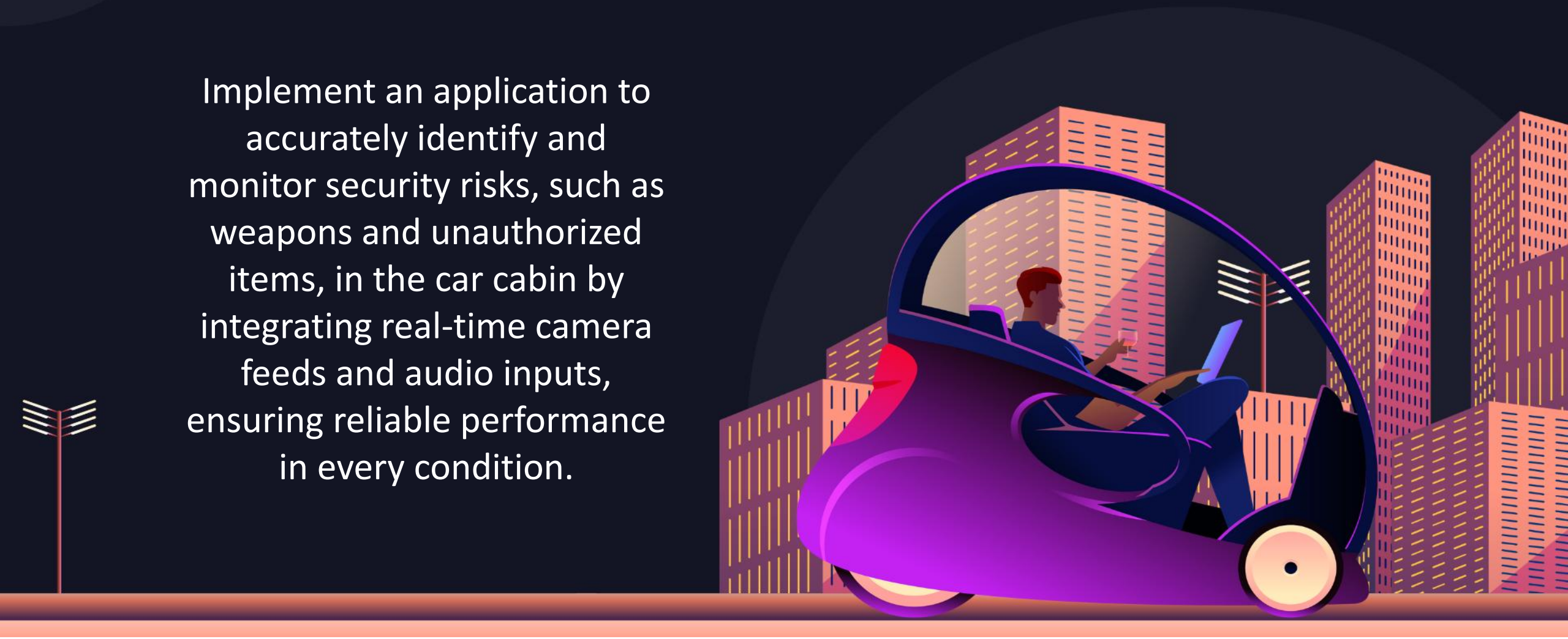
Synchronizing object and voice detection to accurately identify unauthorized activities.

How challenging is it to identify unauthorized items, such as weapons, in low-light scenarios?

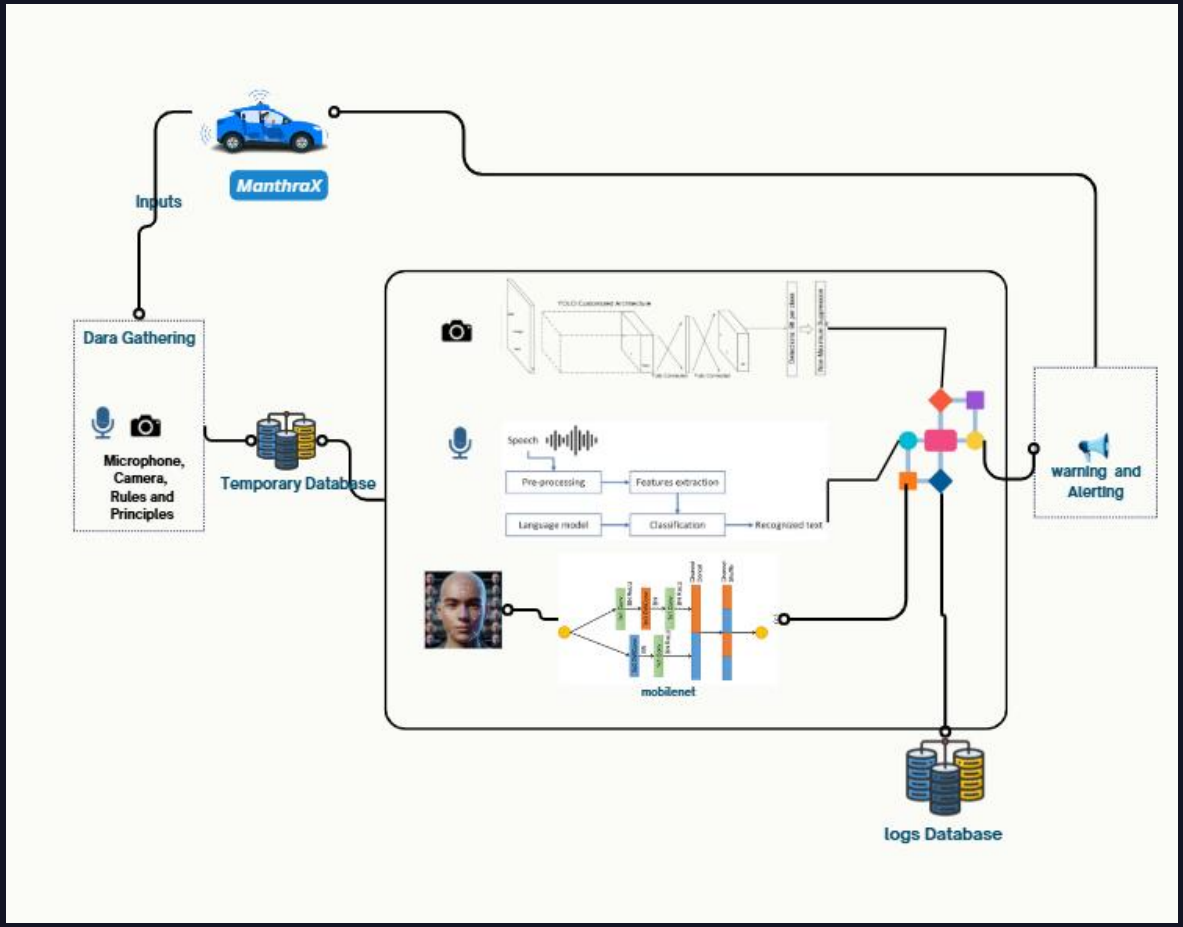
Identifying unauthorized voices or keywords in an environment with multiple overlapping voices.

SPECIFIC OBJECTIVE

Implement an application to accurately identify and monitor security risks, such as weapons and unauthorized items, in the car cabin by integrating real-time camera feeds and audio inputs, ensuring reliable performance in every condition.



Methodology



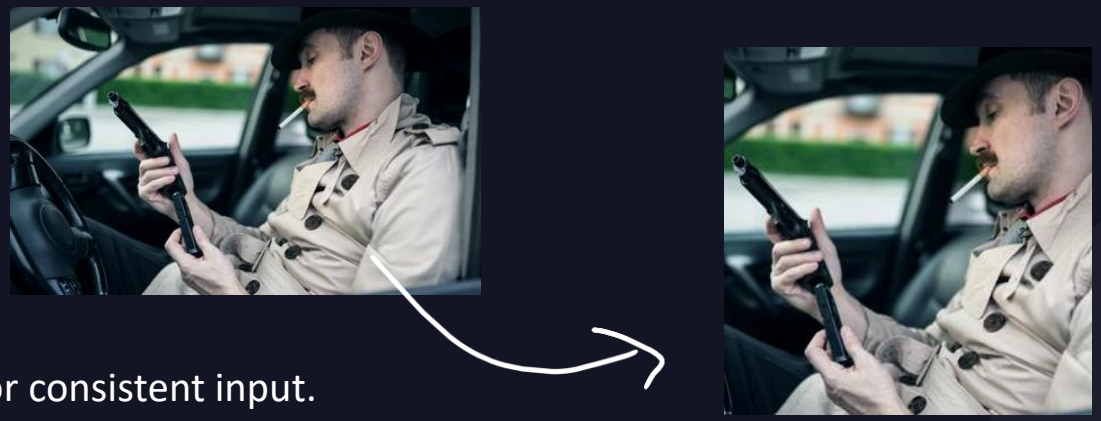
SUB OBJECTIVE 01 - Weapon and harmful objects detection

Identifying and monitor security risks such as weapons and unauthorized items in the car cabin – images




SUB OBJECTIVE 01 - Weapon and harmful objects detection

DATA PREPROCESSING



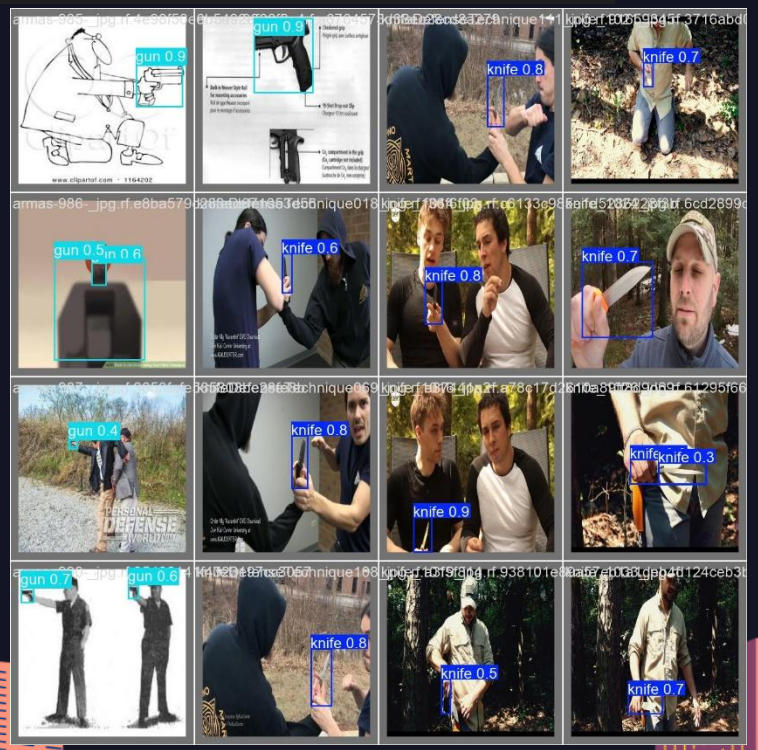
- **Resizing**: Uniform image resizing to 640x640 for consistent input.
- **Augmentation**: Enabled with transformations like flips, scaling, rotations, and cropping.
- **Normalization**: Scales pixel values to a 0–1 range for model compatibility.
- **Batch Loading**: Processes images in batches of 16 with applied transformations.

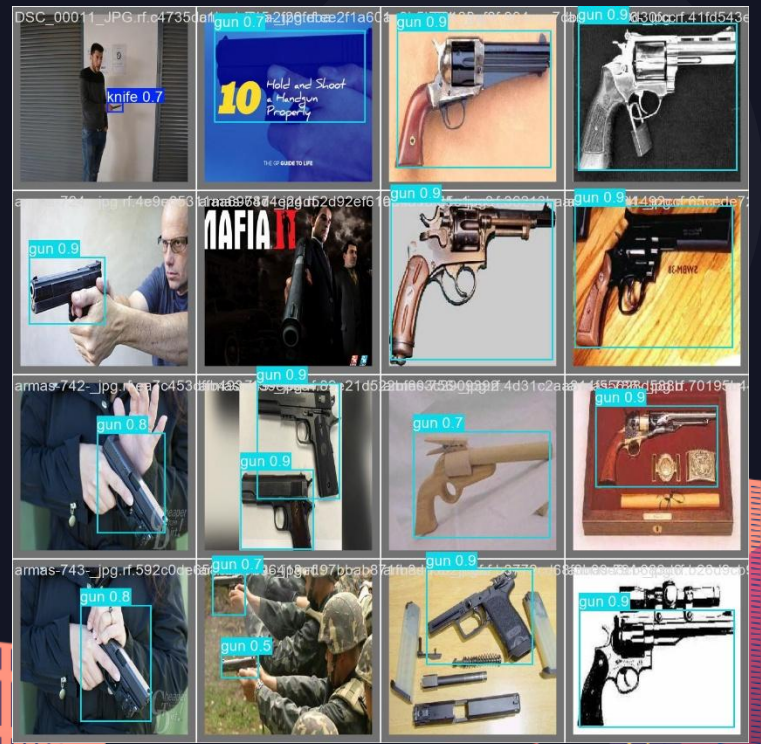
EVIDENCE OF COMPLETION

```
Validating runs\detect\train4\weights\best.pt...
Ultralytics 8.3.21  Python-3.8.20 torch-2.4.1 CUDA:0 (NVIDIA GeForce RTX 3070 Laptop GPU, 8192MiB)
YOLOv5s summary (fused): 193 layers, 9,112,310 parameters, 0 gradients, 23.8 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:08<00:00,  1.86it/s]
                   all       1011       1132      0.917      0.867      0.926      0.652
                 knife        424        439      0.934      0.913      0.949      0.625
                   gun        587        693        0.9      0.821      0.903       0.68
Speed: 0.1ms preprocess, 0.9ms inference, 0.0ms loss, 1.8ms postprocess per image
Results saved to runs\detect\train4
```

- **overall accuracy is approximately 92.6%**.

The model demonstrates high accuracy in detecting classes with strong overall performance.



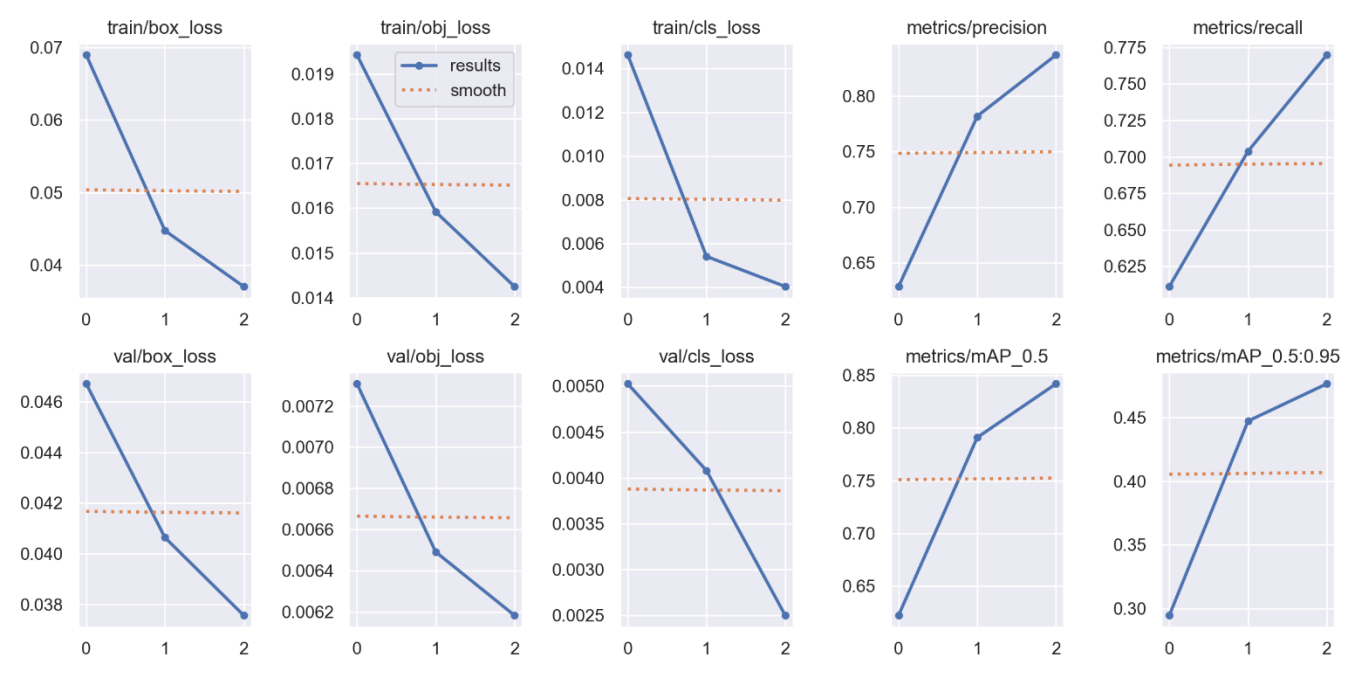


EVIDENCE OF COMPLETION

```
Results for img21.jpg:
image 1/1: 280x390 1 gun
Speed: 3.5ms pre-process, 19.1ms inference, 35.7ms NMS per image at shape (1, 3, 480, 640)
C:\Users\User/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `tor
  with amp.autocast(autocast):
```

EVIDENCE OF COMPLETION

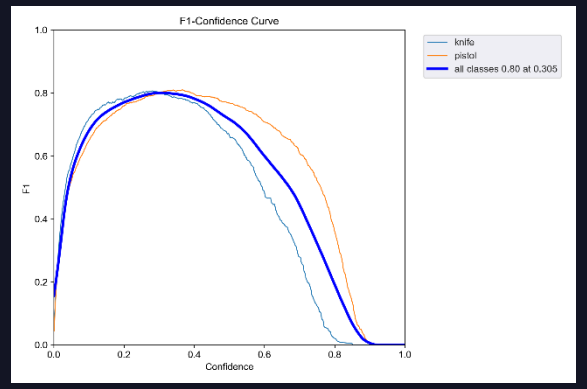
- **Training Losses**: Decreasing box, objectness, and classification losses indicate effective learning.
- **Validation Metrics**: Reduced losses and improved precision, recall, and mAP show strong generalization.
- **Performance Trend**: Continuous improvement across epochs highlights successful optimization.

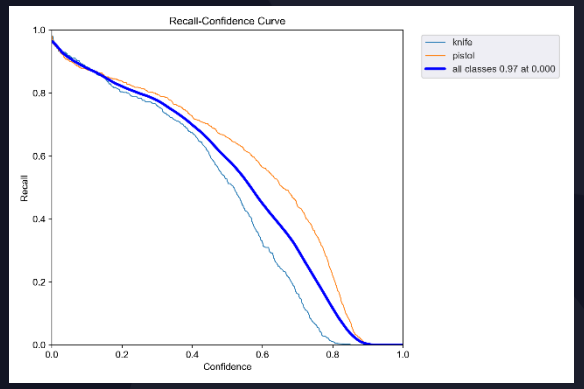


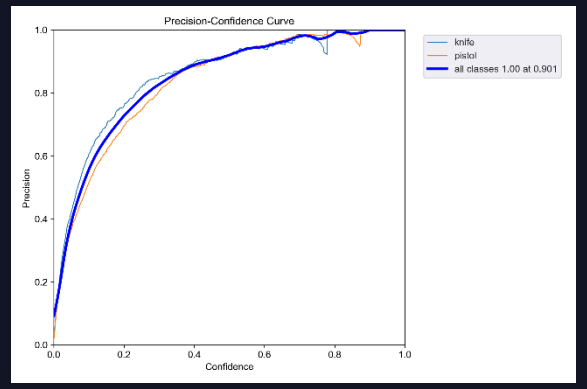
EVIDENCE OF COMPLETION

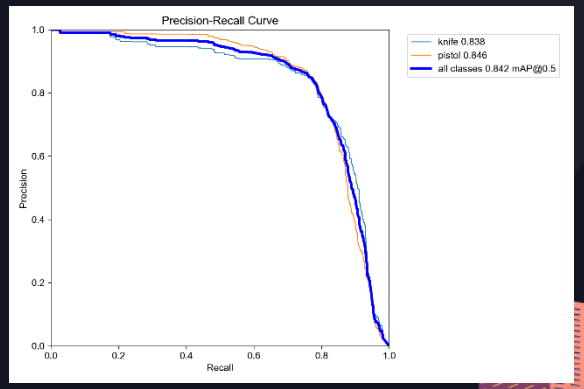
- Precision improves as confidence increases, while recall decreases.

- For all classes, the mAP@0.5 score of **0.842** demonstrates solid performance in detecting and classifying objects.









SUB OBJECTIVE 02 - Harmful status detection by audio

- Analyze audio inputs to identify suspicious or unauthorized behaviors through voice patterns - AUDIO

```
neutral
screaming
glass breaking
gunshot
```

SUB OBJECTIVE 02 - Harmful status detection by audio

DATA PREPROCESSING

- **Silence Removal**: The audio is processed to remove silences based on a defined threshold, ensuring the important parts of the sound remain.

- **Resampling**: Audio is resampled to a consistent sample rate of **22,050 Hz**.

- **Chunking & Padding**: The audio is split into **2.5-second chunks** for uniformity. If the audio is too short, it is padded with silence.

- **Output**: The final audio chunks are saved as **.wav files** for further use or analysis.

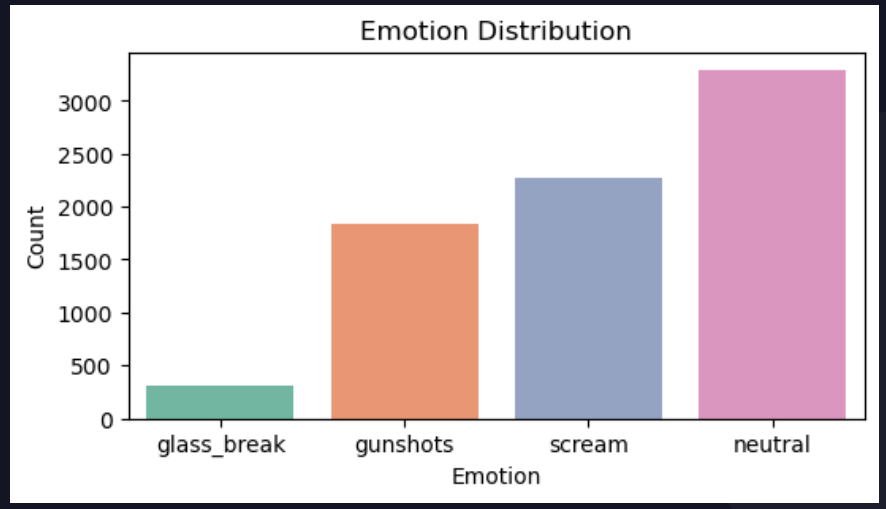
EVIDENCE OF COMPLETION - Data Augmentation

1.Noise Injection

2.Time Stretching

3.Time Shifting

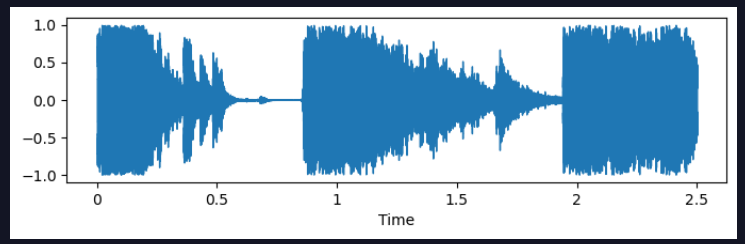
4.Pitch Shifting



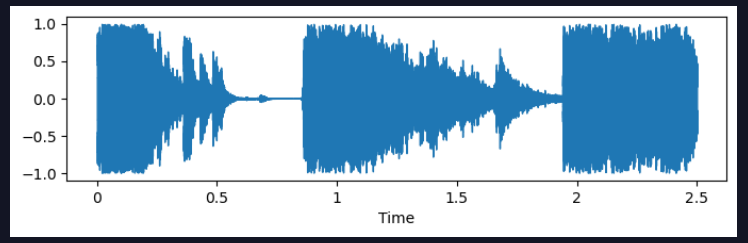
These techniques introduce variability into the training data, helping the model generalize better by mimicking real-world variations in audio data.

EVIDENCE OF COMPLETION

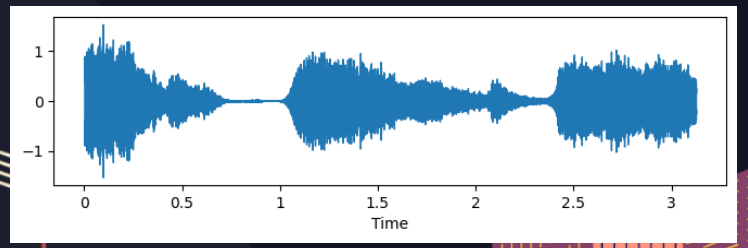
1. Original audio



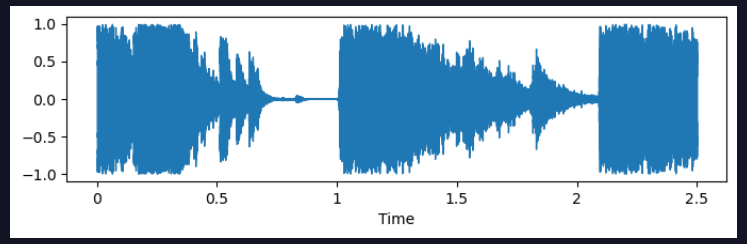
2. Noise injection



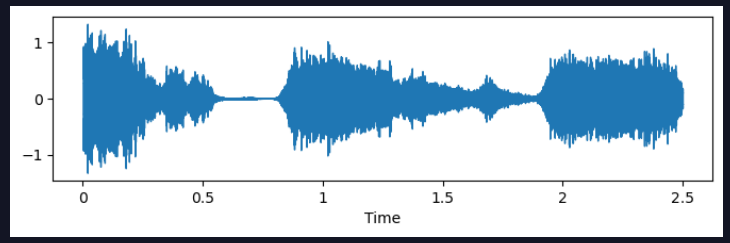
3. stretching



4. shifting



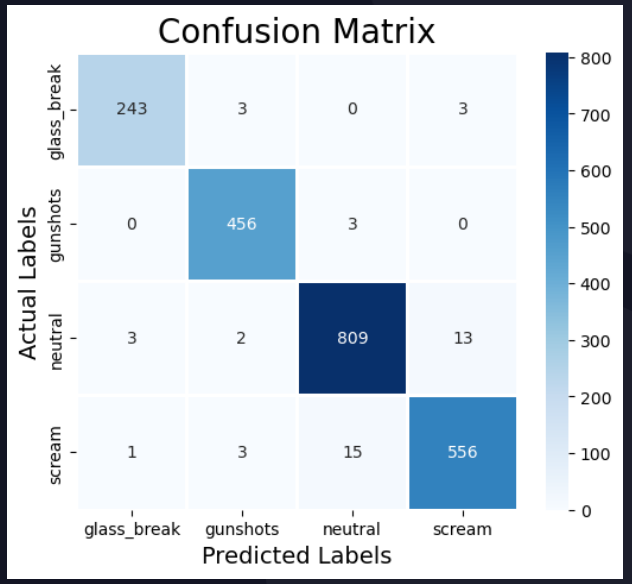
5. Pitch

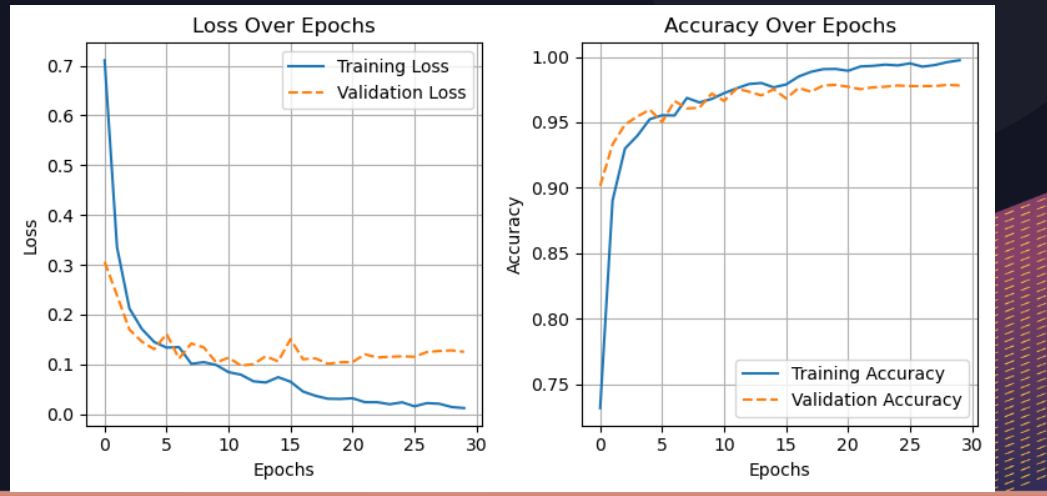


EVIDENCE OF COMPLETION

- The model generalizes well with a strong performance on validation data.
- The model effectively distinguishes between different sound categories, with minor misclassifications. Performance for high-risk sounds like **gunshots** and **glass_break** is particularly reliable.

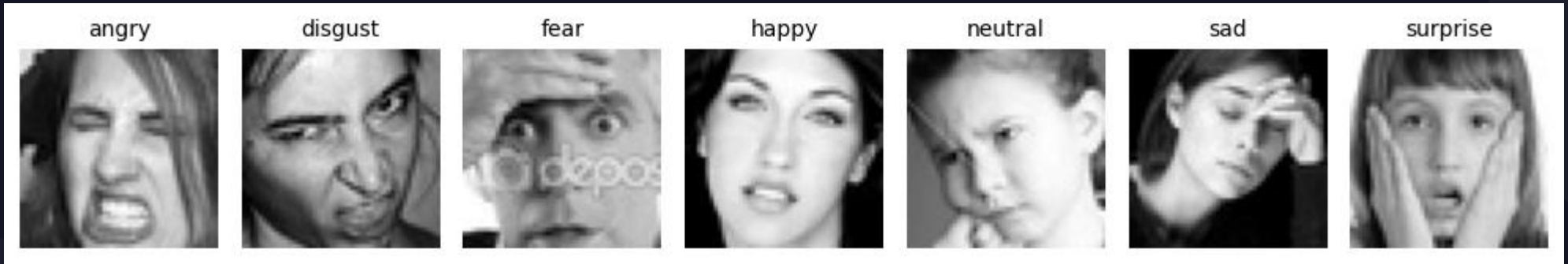
```
Overall Accuracy: 97.82%
```





SUB OBJECTIVE 03 - Facial emotion detection

Implement facial emotion detection to identify emotional cues from both attackers and passengers, helping to assess threat levels based on expressions



SUB OBJECTIVE 03 - Facial emotion detection

DATA PREPROCESSING

- **Batch Size**: 64
- **Input Size**: Images resized to **224x224** pixels
- **Data Augmentation**:
- Horizontal flipping applied to training data to improve generalization.
- Labels are one-hot encoded for multi-class classification.

- **Custom Preprocessing**:

- Grayscale images (e.g., FER2013 dataset) are converted to RGB by duplicating the single channel, ensuring compatibility with RGB-based models.

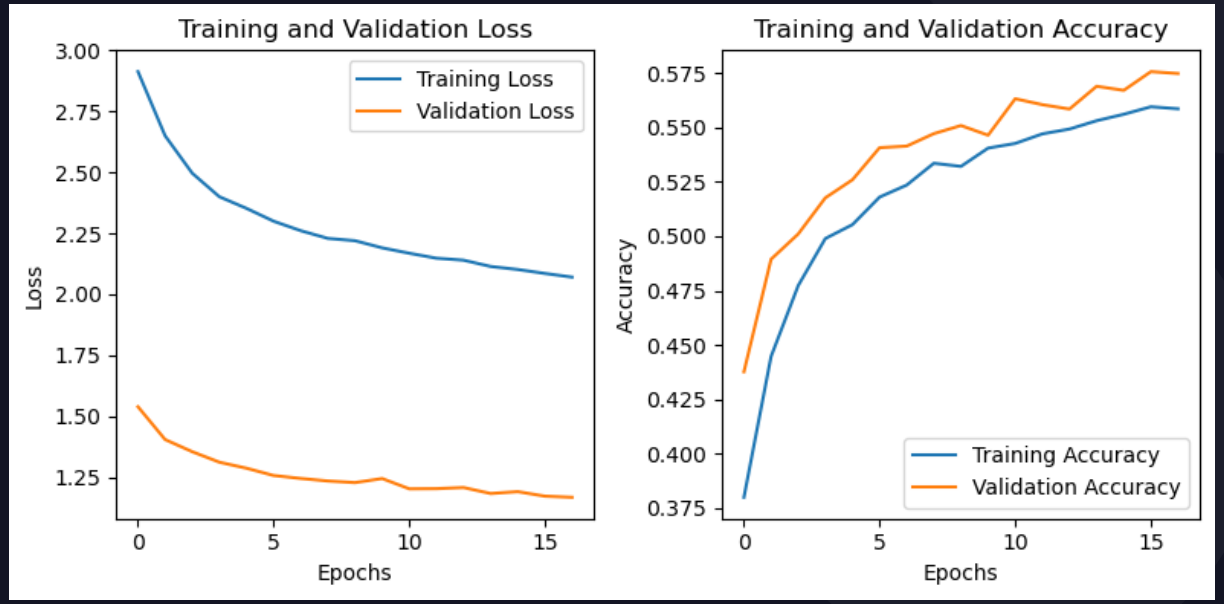
EVIDENCE OF COMPLETION

```
443/444 [============================>.] - ETA: 0s - loss: 3.3456 - accuracy: 0.2696
Epoch 1: val_accuracy improved from -inf to 0.32268, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 29s 66ms/step - loss: 3.3450 - accuracy: 0.2696 - val_loss: 1.7808 - val_accuracy: 0.3227
Epoch 2/3
443/444 [============================>.] - ETA: 0s - loss: 3.1651 - accuracy: 0.3172
Epoch 2: val_accuracy improved from 0.32268 to 0.33959, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 28s 64ms/step - loss: 3.1645 - accuracy: 0.3172 - val_loss: 1.7517 - val_accuracy: 0.3396
Epoch 3/3
443/444 [============================>.] - ETA: 0s - loss: 3.0941 - accuracy: 0.3318
Epoch 3: val_accuracy did not improve from 0.33959
444/444 [==============================] - 28s 62ms/step - loss: 3.0943 - accuracy: 0.3316 - val_loss: 1.7413 - val_accuracy: 0.3320
```

```
444/444 [==============================] - 31s 68ms/step - loss: 2.9128 - accuracy: 0.3799 - val_loss: 1.5388 - val_accuracy: 0.4376
Epoch 5/20
444/444 [==============================] - ETA: 0s - loss: 2.6499 - accuracy: 0.4447
Epoch 5: val_accuracy improved from 0.43764 to 0.48939, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 29s 66ms/step - loss: 2.6499 - accuracy: 0.4447 - val_loss: 1.4044 - val_accuracy: 0.4894
Epoch 6/20
443/444 [============================>.] - ETA: 0s - loss: 2.4960 - accuracy: 0.4772
Epoch 6: val_accuracy improved from 0.48939 to 0.50100, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 30s 68ms/step - loss: 2.4964 - accuracy: 0.4773 - val_loss: 1.3546 - val_accuracy: 0.5010
Epoch 7/20
443/444 [============================>.] - ETA: 0s - loss: 2.4002 - accuracy: 0.4989
Epoch 7: val_accuracy improved from 0.50100 to 0.51763, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 29s 65ms/step - loss: 2.3998 - accuracy: 0.4989 - val_loss: 1.3115 - val_accuracy: 0.5176
Epoch 8/20
443/444 [============================>.] - ETA: 0s - loss: 2.3527 - accuracy: 0.5053
Epoch 8: val_accuracy improved from 0.51763 to 0.52595, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 29s 64ms/step - loss: 2.3522 - accuracy: 0.5053 - val_loss: 1.2868 - val_accuracy: 0.5259
Epoch 9/20
444/444 [==============================] - ETA: 0s - loss: 2.2995 - accuracy: 0.5180
Epoch 9: val_accuracy improved from 0.52595 to 0.54071, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 29s 64ms/step - loss: 2.2995 - accuracy: 0.5180 - val_loss: 1.2570 - val_accuracy: 0.5407
Epoch 10/20
444/444 [==============================] - ETA: 0s - loss: 2.2606 - accuracy: 0.5235
...
Epoch 20/20
444/444 [==============================] - ETA: 0s - loss: 2.0703 - accuracy: 0.5586
Epoch 20: val_accuracy did not improve from 0.57569
444/444 [==============================] - 29s 65ms/step - loss: 2.0703 - accuracy: 0.5586 - val_loss: 1.1676 - val_accuracy: 0.5748
```

EVIDENCE OF COMPLETION

```
Epoch 18: val_accuracy improved from -inf to 0.63876, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 79s 172ms/step - loss: 1.8238 - accuracy: 0.6208 - val_loss: 0.9844 - val_accuracy: 0.6388
Epoch 19/27
444/444 [==============================] - ETA: 0s - loss: 1.5385 - accuracy: 0.6685
Epoch 19: val_accuracy improved from 0.63876 to 0.65783, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 76s 170ms/step - loss: 1.5385 - accuracy: 0.6685 - val_loss: 0.9507 - val_accuracy: 0.6578
Epoch 20/27
444/444 [==============================] - ETA: 0s - loss: 1.3974 - accuracy: 0.6931
Epoch 20: val_accuracy improved from 0.65783 to 0.66829, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 75s 170ms/step - loss: 1.3974 - accuracy: 0.6931 - val_loss: 0.9238 - val_accuracy: 0.6683
Epoch 21/27
444/444 [==============================] - ETA: 0s - loss: 1.2915 - accuracy: 0.7115
Epoch 21: val_accuracy improved from 0.66829 to 0.68750, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 76s 170ms/step - loss: 1.2915 - accuracy: 0.7115 - val_loss: 0.8827 - val_accuracy: 0.6875
Epoch 22/27
444/444 [==============================] - ETA: 0s - loss: 1.1872 - accuracy: 0.7329
Epoch 22: val_accuracy improved from 0.68750 to 0.69108, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 76s 171ms/step - loss: 1.1872 - accuracy: 0.7329 - val_loss: 0.8975 - val_accuracy: 0.6911
Epoch 23/27
444/444 [==============================] - ETA: 0s - loss: 1.0890 - accuracy: 0.7551
Epoch 23: val_accuracy improved from 0.69108 to 0.69538, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 77s 173ms/step - loss: 1.0890 - accuracy: 0.7551 - val_loss: 0.8968 - val_accuracy: 0.6954
Epoch 24/27
...
Epoch 27/27
444/444 [==============================] - ETA: 0s - loss: 0.7487 - accuracy: 0.8323
Epoch 27: val_accuracy improved from 0.69796 to 0.70212, saving model to mobilenet_face_ft.h5
444/444 [==============================] - 77s 174ms/step - loss: 0.7487 - accuracy: 0.8323 - val_loss: 0.9770 - val_accuracy: 0.7021
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```



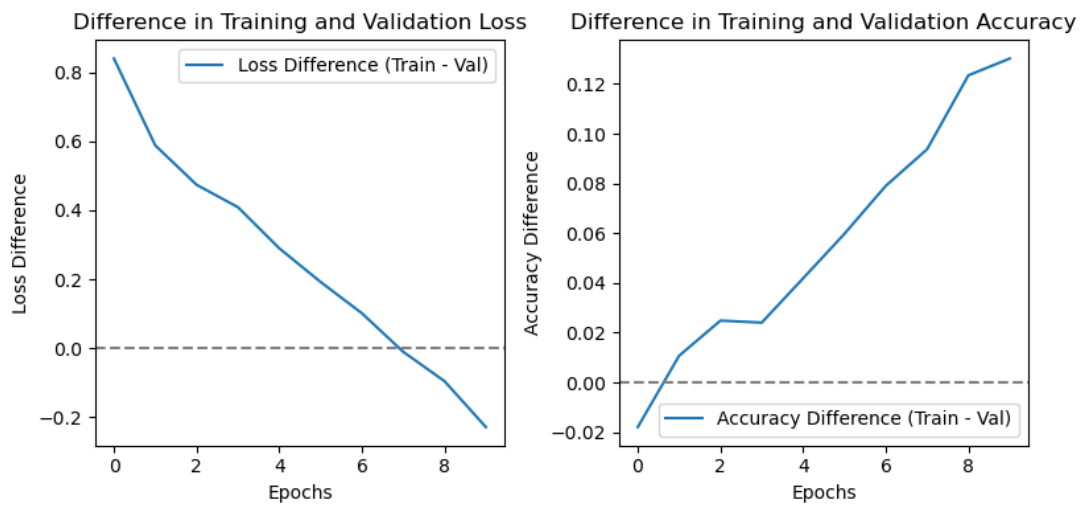
- **Training Loss:** Decreasing steadily (final: 0.7487)
- **Training Accuracy:** Improved to 83.23%
- **Validation Loss:** Fluctuating but manageable (final: 0.9770)
- **Validation Accuracy:** Improved from 63.88% to 70.21%

EVIDENCE OF COMPLETION

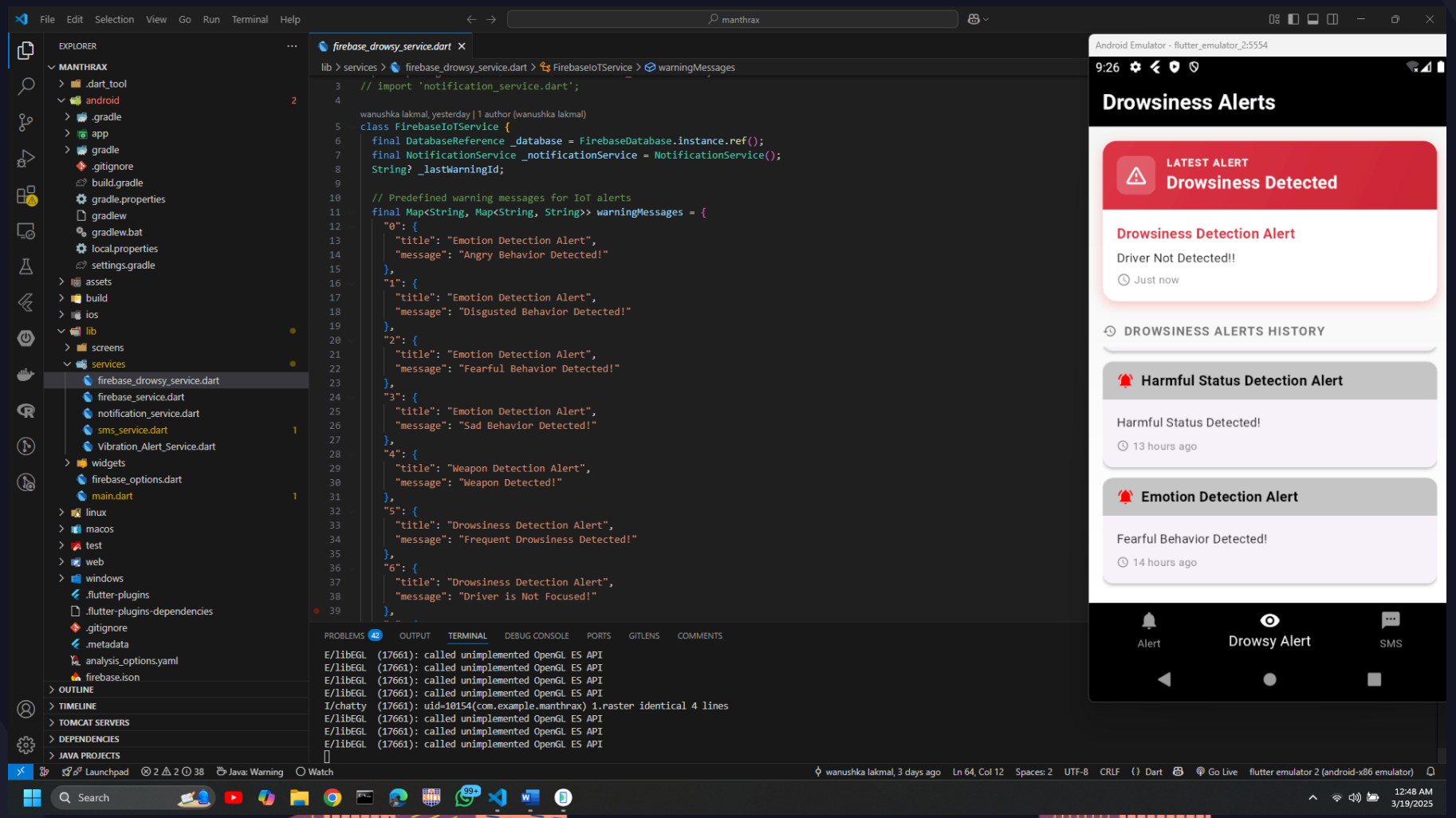
•**Loss Difference:** Training and validation losses converge, indicating a balanced learning process.

•**Accuracy Difference:** Training accuracy slightly exceeds validation accuracy, suggesting a good fit without major overfitting.

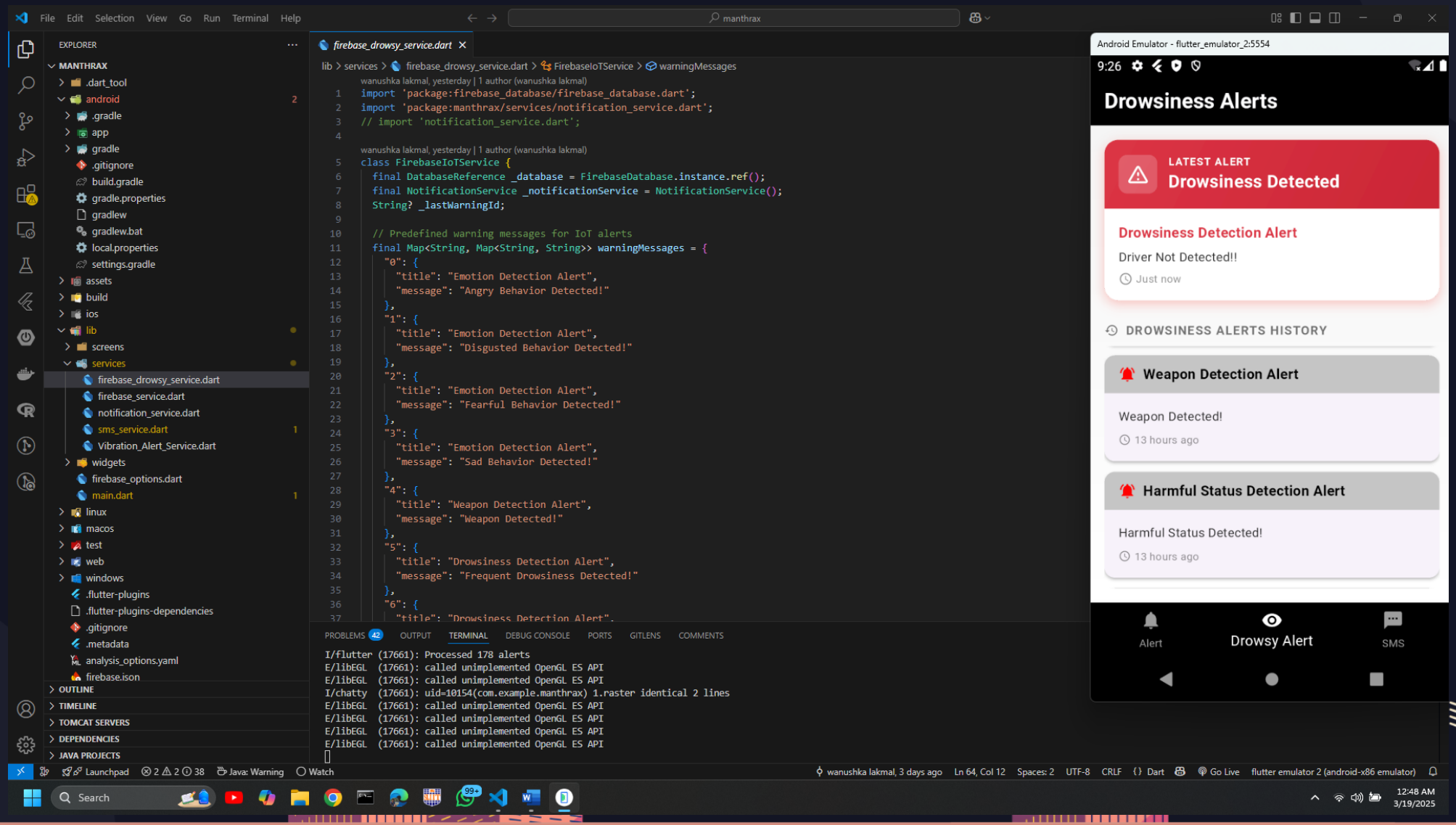
•**Key Insight:** The model demonstrates consistent improvement with minimal divergence between training and validation performance.



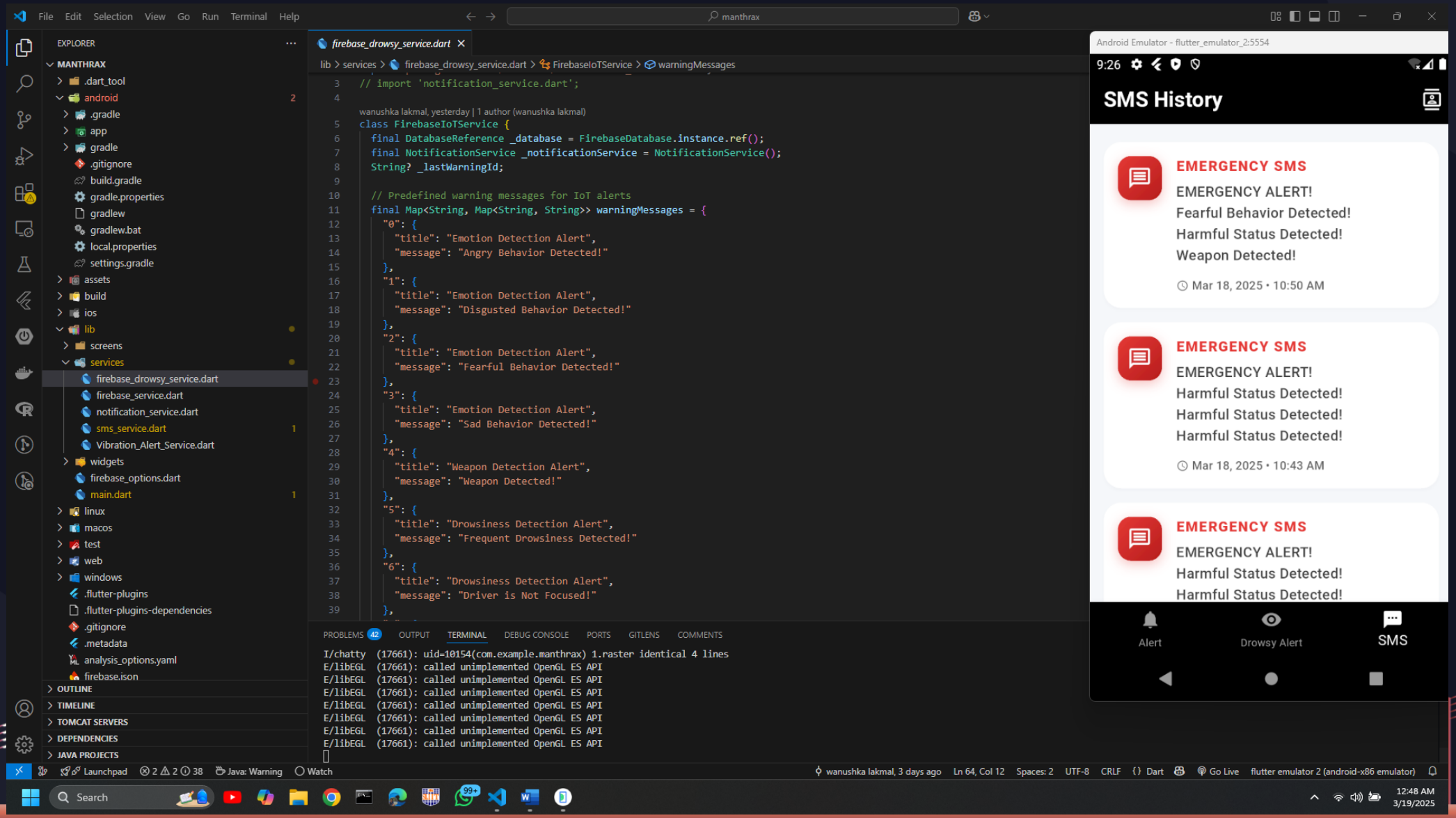
alert and notification system



alert and notification system



alert and notification system



Progress

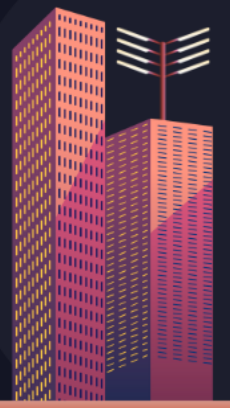
Completed Tasks

- Weapon and harmful objects detection
- Facial emotion detection
- Harmful status detection by audio
- alert and notification system
- System Integration

Upcoming Tasks

. Alert Handling Update

. Model Integrate Algorithm For Update



Technologies to be Used

AI/ML Frameworks:
TensorFlow, PyTorch, Keras.

Development Tools:
IDEs: VSCode, PyCharm, Jupyter Notebook

Simulation Frameworks:
CARLA, AirSim, or Unity

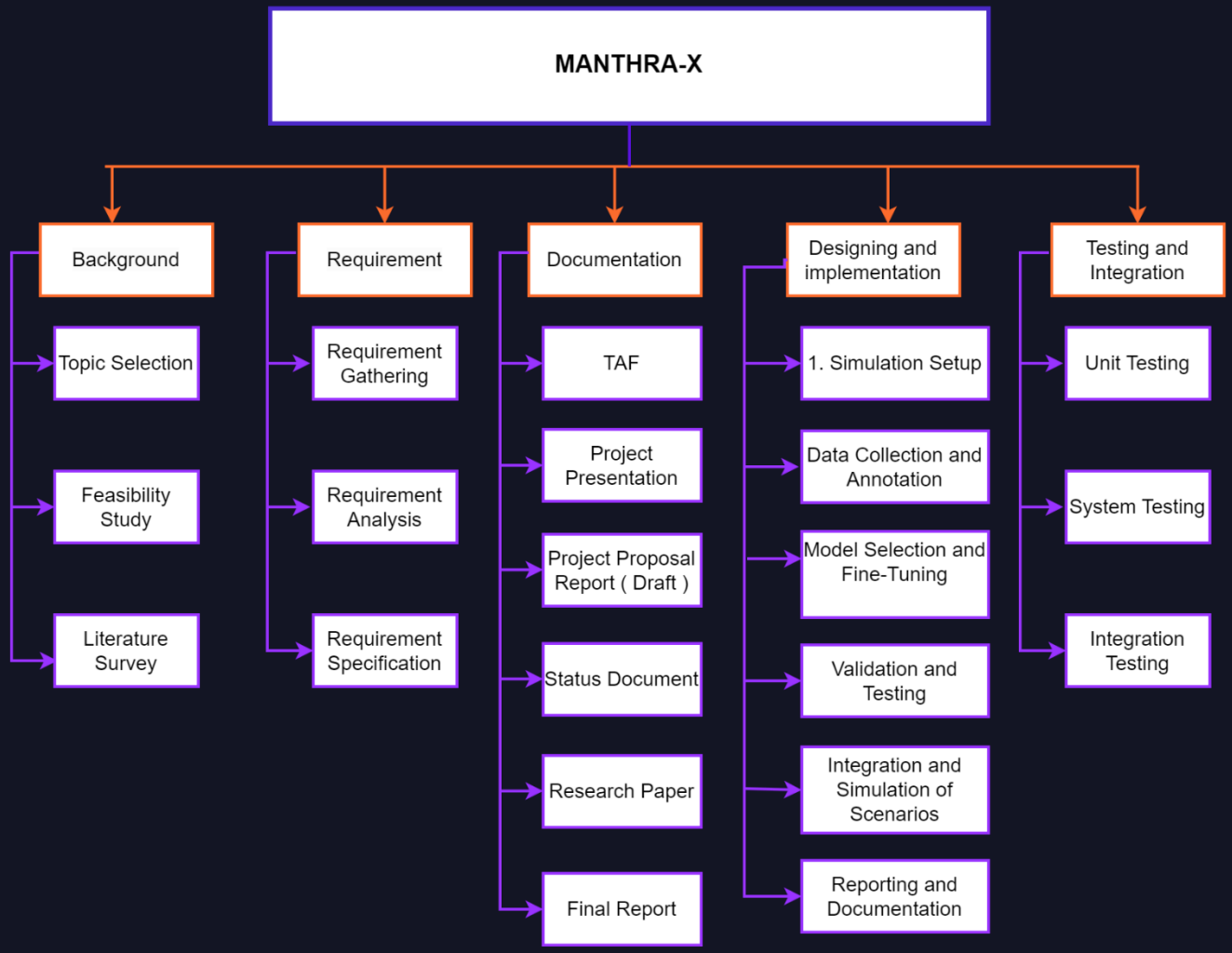
Security: Encryption.

REQUIREMENT SPECIFICATION

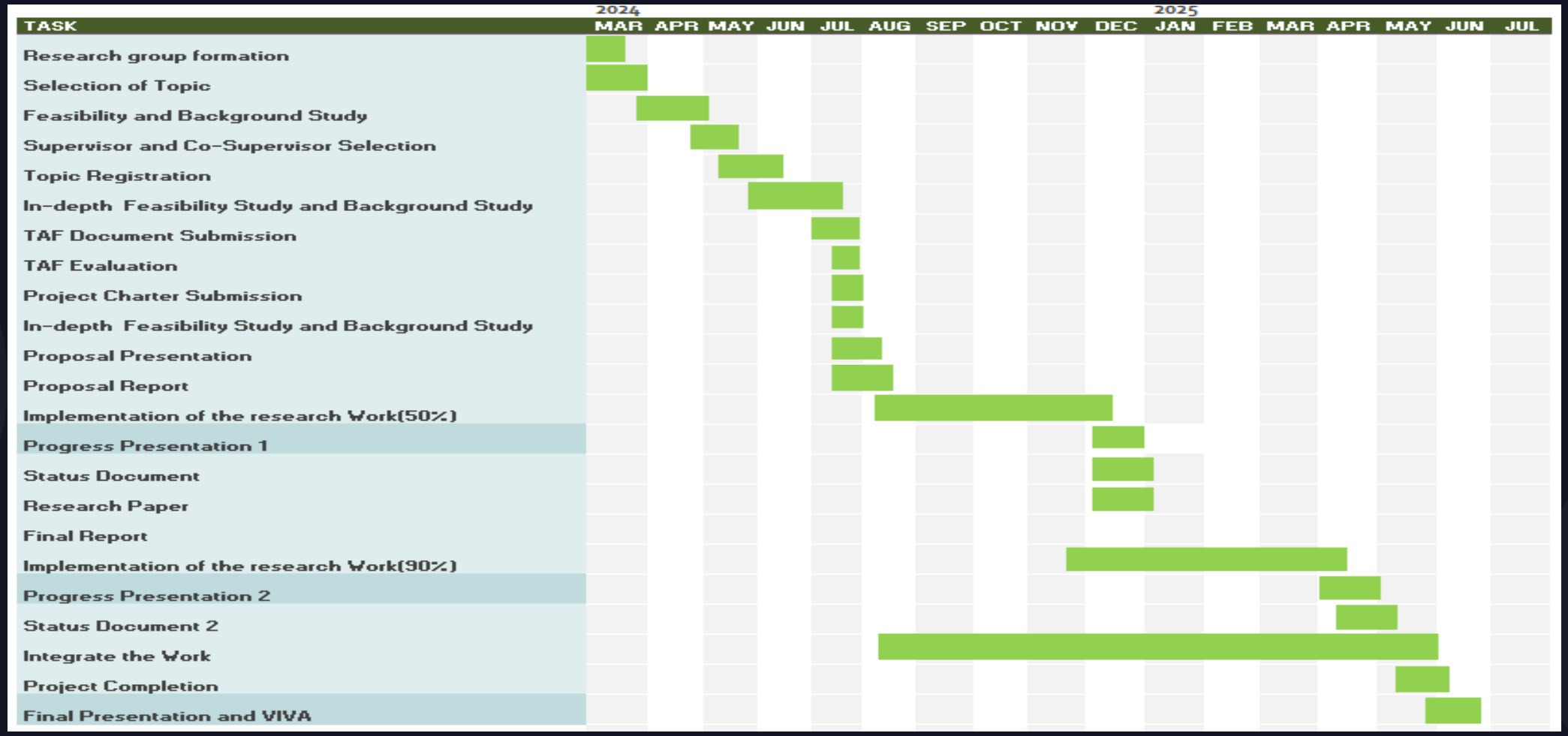
System Requirements

- **High-performance computing (HPC) cluster or workstations:** For model training, simulation, and real-time processing.
- **LiDAR sensors:** For accurate distance and object detection.
- **High-resolution cameras:** For capturing detailed visual information.
- **Microphones: High-quality microphones are essential for capturing clear audio input.**

Work Breakdown Structure (WBS)

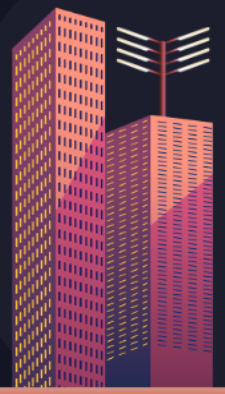


GANNT CHART


2024
2025
TASK
MAR APR MAY JUN JUL AUG SEP OCT NOV DEC JAN FEB MAR APR MAY JUN JUL
Research group formation
Selection of Topic
Feasibility and Background Study
Supervisor and Co-Supervisor Selection
Topic Registration
In-depth Feasibility Study and Background Study
TAF Document Submission
TAF Evaluation
Project Charter Submission
In-depth Feasibility Study and Background Study
Proposal Presentation
Proposal Report
Implementation of the research Work(50%)
Progress Presentation 1
Status Document
Research Paper
Final Report
Implementation of the research Work(90%)
Progress Presentation 2
Status Document 2
Integrate the Work
Project Completion
Final Presentation and VIVA

References

- [1] Gun Detection in Real-Time Video Streams Using Convolutional Neural Networks. (n.d.). Gun Detection in Real-Time Video Streams Using Convolutional Neural Networks | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8681234
- [2] Real-Time Object Detection with Deep Learning for Autonomous Vehicles. (n.d.). Real-Time Object Detection with Deep Learning for Autonomous Vehicles | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9073802
- [3] Voice Recognition Systems for Security Applications: A Survey. (n.d.). Voice Recognition Systems for Security Applications: A Survey | IEEE Journal Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8847891
- [4] Multimodal Sensor Fusion for Enhanced Security and Comfort in Autonomous Vehicles. (n.d.). Multimodal Sensor Fusion for Enhanced Security and Comfort in Autonomous Vehicles | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9354376
- [5] D. Garikapati and S. S. Shetiya, "Autonomous Vehicles: Evolution of Artificial Intelligence and Learning Algorithms," in IEEE International Conference on Autonomous Systems, Tokyo, Japan, Feb. 2024, pp. 1-10.
- [6] A. Mishra, S. Lee, D. Kim, and S. Kim, "In-Cabin Monitoring System for Autonomous Vehicles," Sensors, vol. 22, no. 12, pp. 4360
- [7] A. A. Alsulami, Q. Abu Al-Haija, B. Alturki, A. Alqahtani, and R. Alsini, "Security strategy for autonomous vehicle cyber-physical systems using transfer learning," Journal of Cloud Computing: Advances, Systems and Applications, vol. 12, no. 1, pp. 1-18, 2023.



SUB OBJECTIVES

- Incorporate and Operationalize Ethical Principles and Frameworks-

Identify and integrate relevant ethical principles (e.g., utilitarianism, deontological ethics) into the system's decision-making processes to ensure alignment with established ethical standards.

- Develop and Integrate Driver Emotion Recognition-

Create and implement a system using voice and facial recognition technologies to accurately monitor and interpret the driver's emotional state, ensuring this information is effectively used in ethical decision-making.

- Establish Object and Road User Prioritization Techniques-

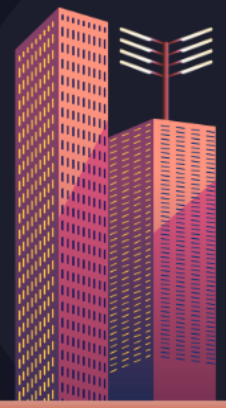
Develop methods for prioritizing objects and vulnerable road users based on ethical considerations and contextual information, ensuring decisions are aligned with safety and ethical standards.

- Ensure System Adaptability to Dynamic and Diverse Environments –

Design mechanisms that allow the system to adapt to varying driving conditions, societal values, and situational demands, maintaining ethical and safety standards.

- Enhance Transparency and User Trust-

Create features that provide clear explanations of the system's decision-making processes, improving transparency and fostering trust among users regarding the ethical decisions made by the vehicle.

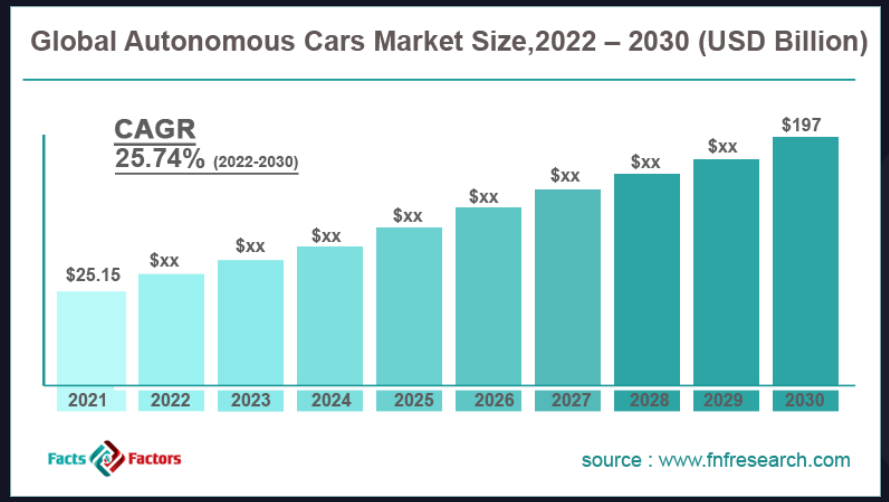


Market Analysis

1. **Autonomous Vehicle Market Overview:**

- The global autonomous vehicle market is projected to grow significantly, with a CAGR of 25.74% from 2022 to 2030. The demand is driven by technological advancements, increased safety concerns, and the potential for reducing traffic congestion.

- Key players in the market include Tesla, Waymo, Cruise, and Uber, with significant investments in R&D and collaborations to improve autonomous driving technologies.



Commercialization

Market Potential

- **Demand Drivers**: Increased safety and efficiency in traffic management.
- **Urban Mobility**: Rising demand for advanced transportation solutions.
- **Emerging Economies**: High growth potential in developing countries.

Commercialization

Partnerships

- **Local Governments**: Collaborate on pilot programs and regulatory approvals.
- **Automotive Manufacturers**: Partner for technology integration and scaling production.
- **Tech Companies**: Work with AI and sensor technology firms for innovation and development.
- **Academic Institutions**: Engage with universities for research support and testing in real world scenarios.

Thank you