MANTHRA-X:

PIONEERING PRECISION, THE FUTURE O F 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

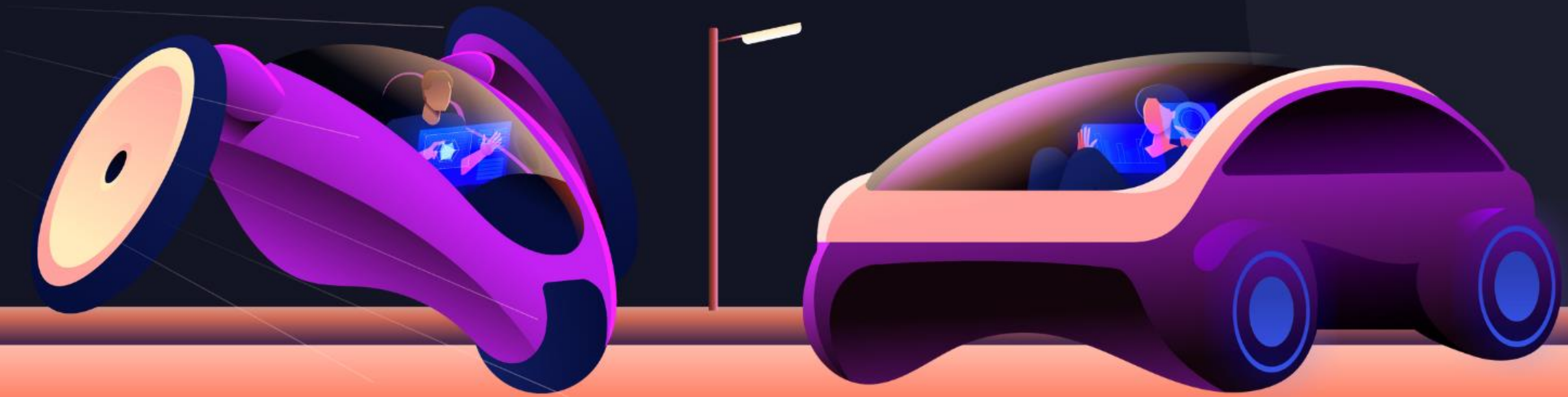
BACKGROUND

RESEARCH PROBLEM

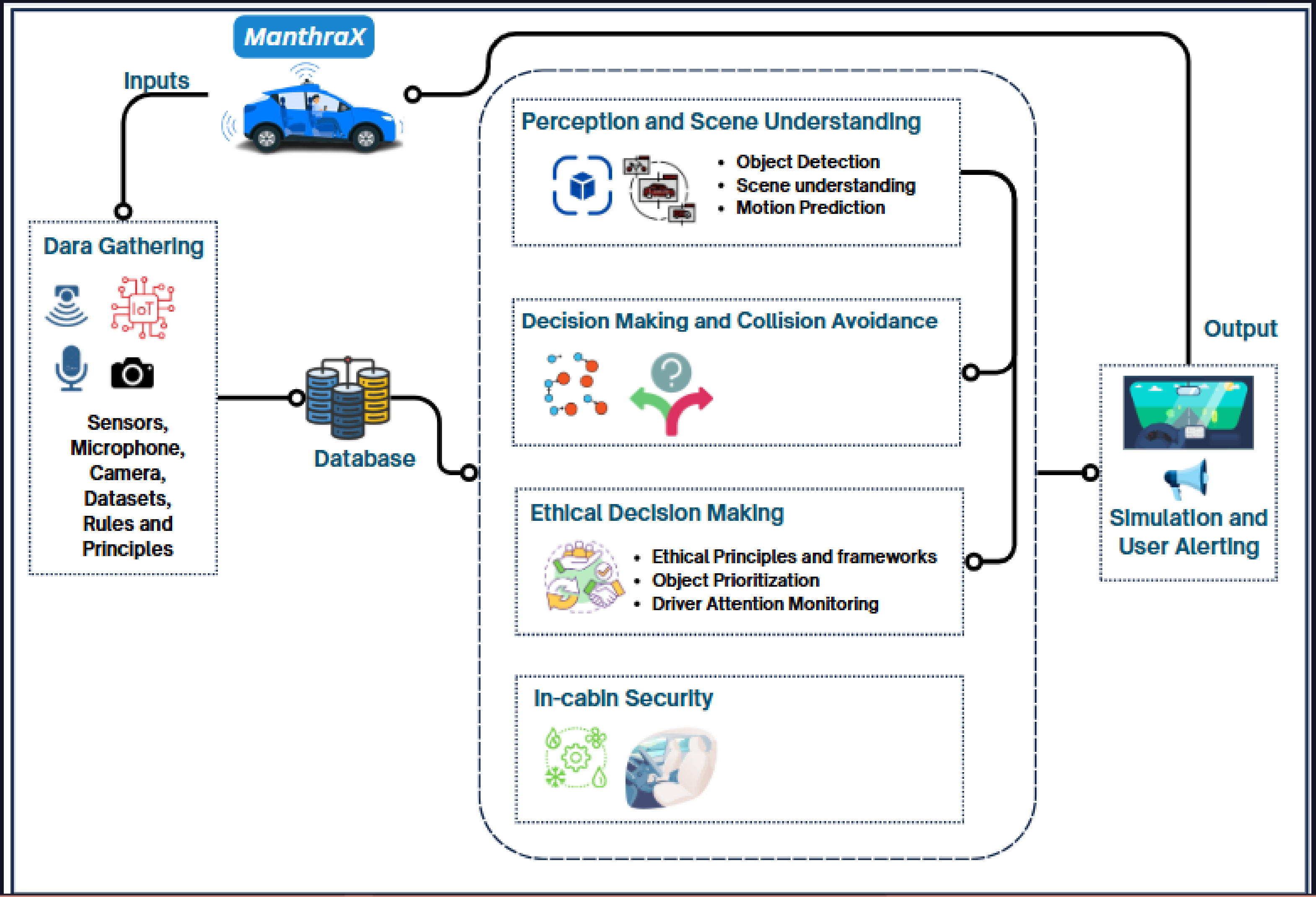
1. Enhancing **object detection and motion prediction** in crowded environments with occlusions.
2. Enabling **real-time decision-making** in unpredictable, unstructured traffic.
3. Designing **ethically sound systems** that balance cultural norms with universal safety standards.
4. Improving **in-cabin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



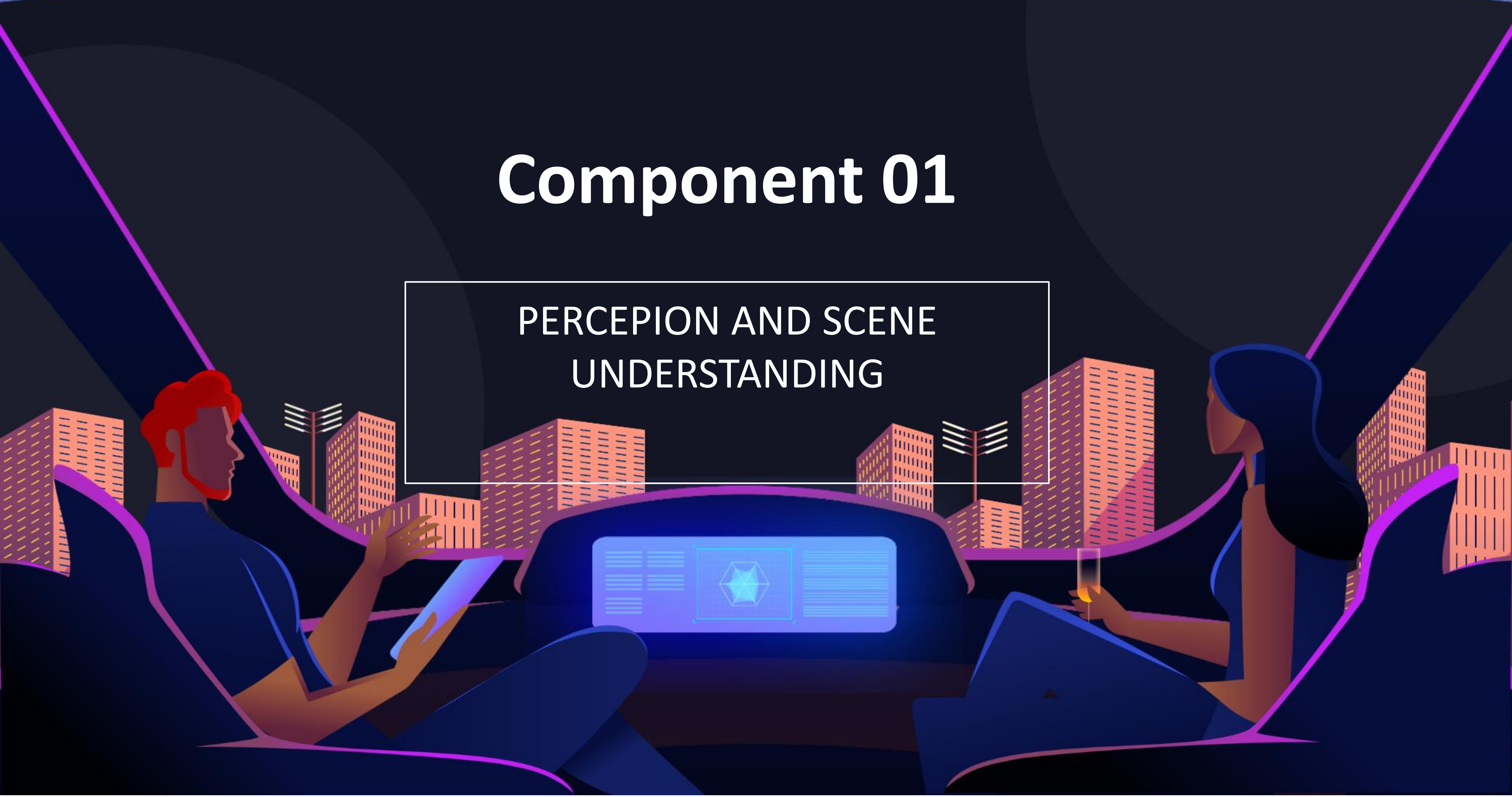
IT21160448 | AKALANKA P.A.A

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

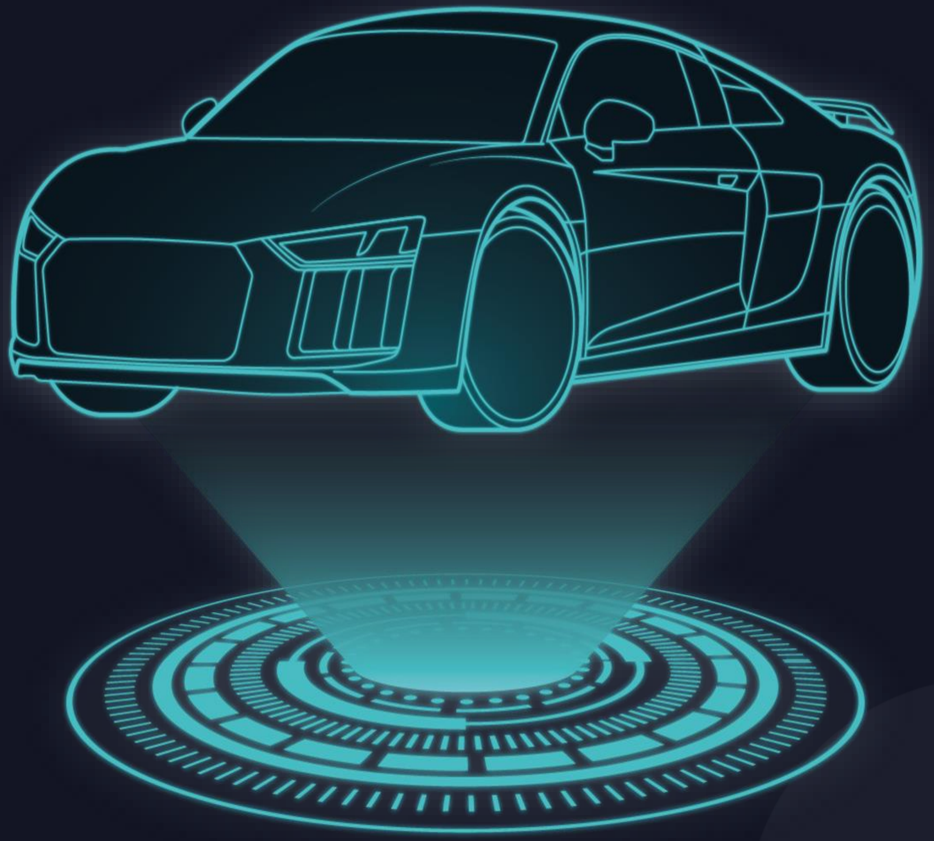
DATA SCIENCE

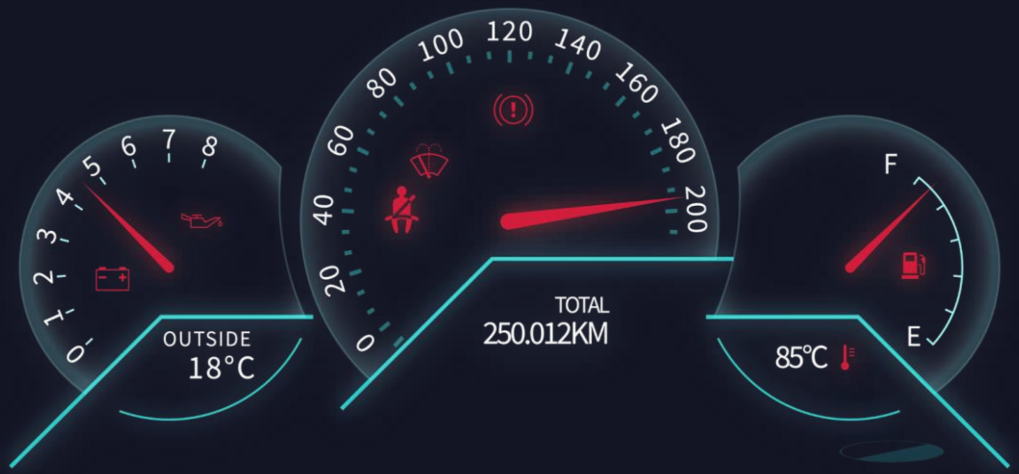
Component 01

PERCEPION AND SCENE UNDERSTANDING



Enhancing Perception and Motion Prediction in Autonomous Vehicles Through Hybrid Modeling and Self-Supervised Learning.





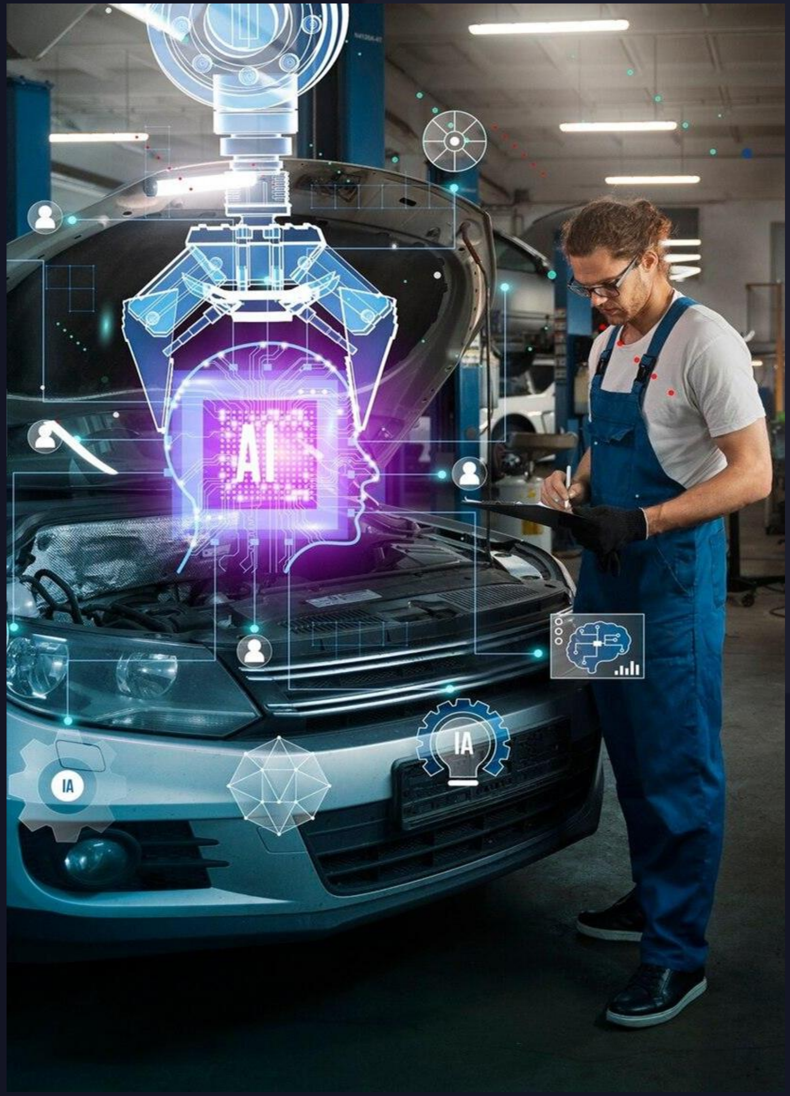
BACKGROUND

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	✓

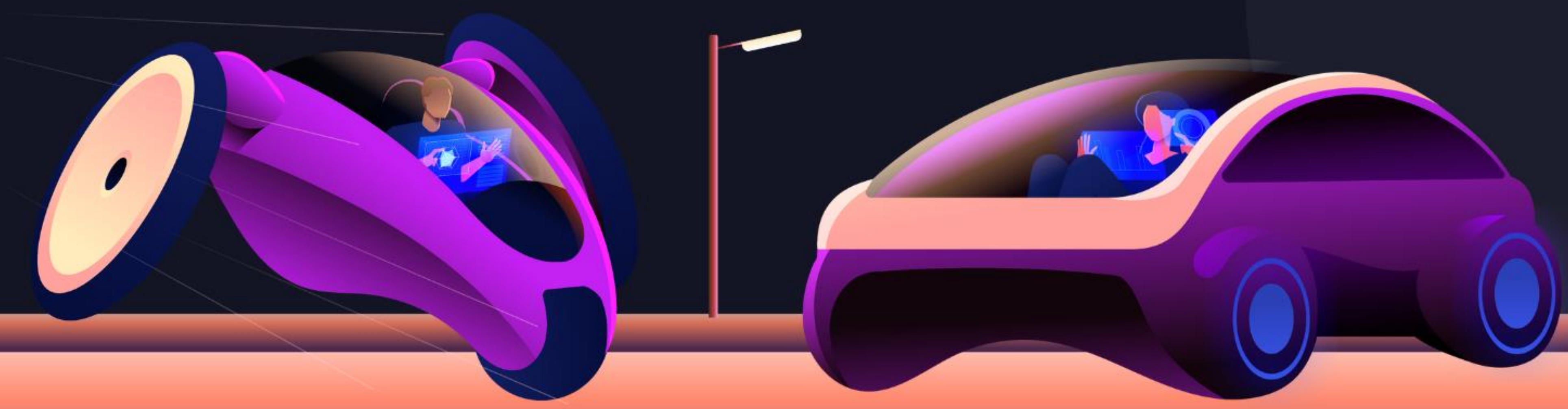
RESEARCH PROBLEM?

- How can graph neural networks be effectively integrated with transformer architectures for multi-modal sensor fusion in autonomous vehicles?
- What self-supervised tasks are most effective for improving object detection and motion prediction in challenging driving conditions?
- Can self-supervised pre-training enhance the performance of hybrid models for autonomous vehicle perception?

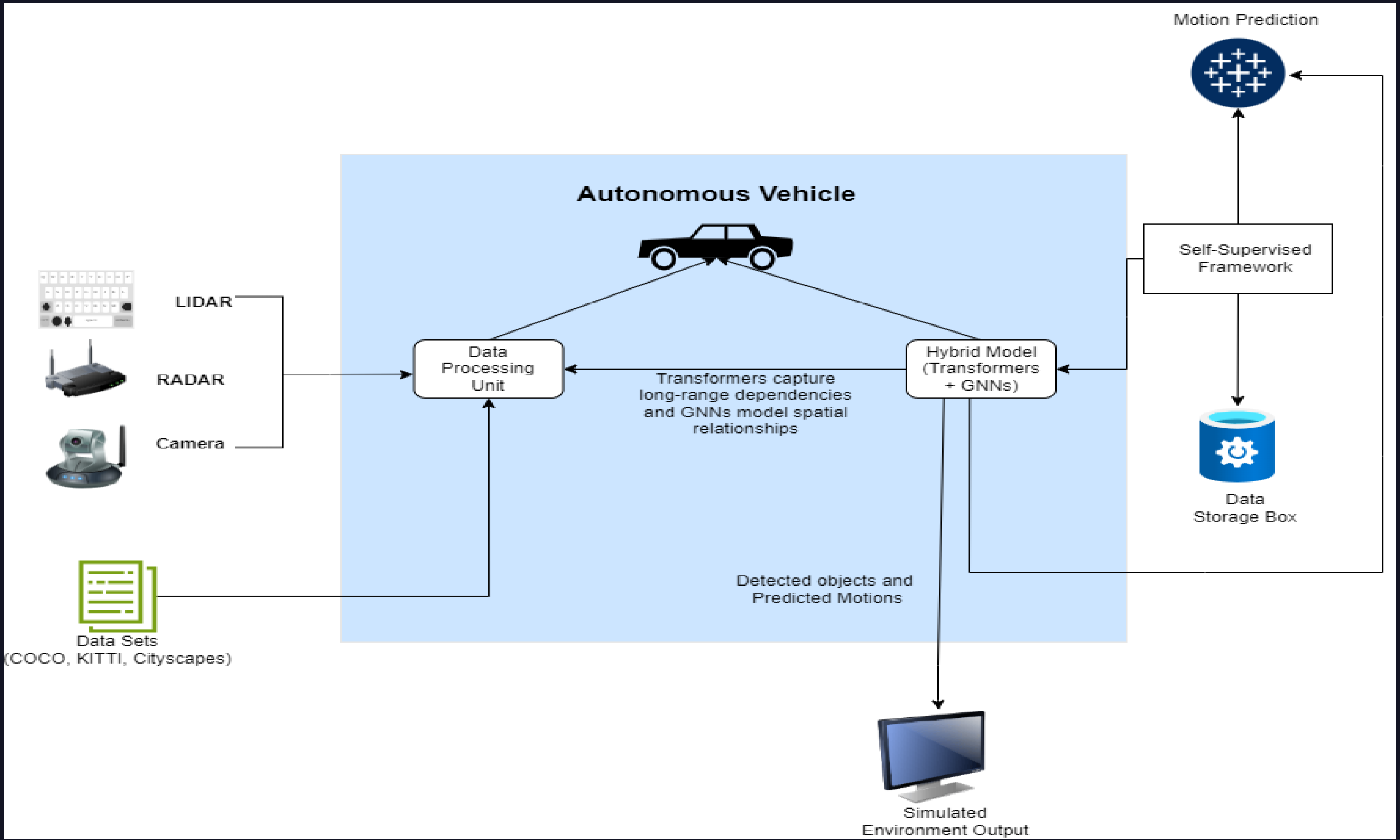


SPECIFIC OBJECTIVE

- To develop a robust and efficient perception system for autonomous vehicles capable of accurately detecting, tracking, and predicting the behavior of objects in complex urban environments by integrating hybrid model architectures and self-supervised learning techniques.



SYSTEM DIAGRAM



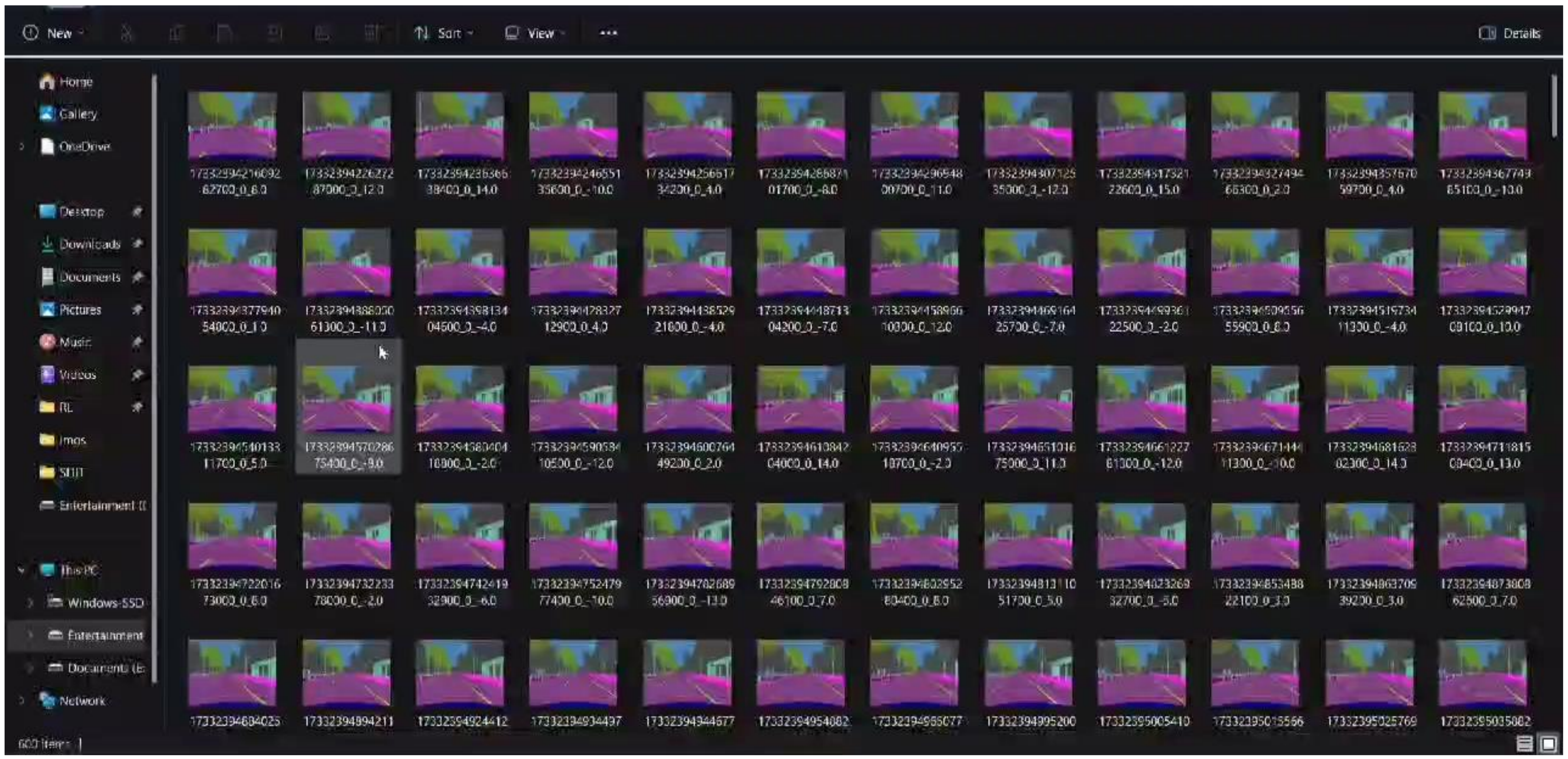
Demonstration of Object Detection

Training & Testing


```
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

Demonstration of Sementic Data Extraction


New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
This PC
Network

Future Directions

1.Motion Prediction:

- Train models for object trajectory prediction.
- Use sensor fusion with LIDAR, RADAR, and cameras.

2.Hybrid Model Integration:

- Combine Transformer Networks and GNNs.
- Test on crowded, multi-object scenarios.

3.Self-Supervised Learning:

- Implement frameworks to learn from unlabeled data.
- Enhance accuracy with pseudo-label generation.

4.Attention Mechanisms:

- Improve detection with attention-based focus.
- Integrate into motion prediction models.

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.
- **Radar sensors:** For object detection and velocity estimation in adverse weather conditions.
- **High-resolution cameras:** For capturing detailed visual information.
- **IMU (Inertial Measurement Unit):** For vehicle motion sensing and stabilization.

REQUIREMENT SPECIFICATION

Software Requirements

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

Frameworks and Libraries:
AI/ML: Tensorflow, Pytorch, Keras
Libraries: OpenCV, PCL, RADAR Data processing libraries

Simulation Frameworks:
CARLA, AirSim, or Unity

Databases:
NoSQL Databases: MongoDB, Cassandra

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

Security: Cryptography Libraries

Web Frameworks: Node.js, Flask, Django

REQUIREMENT SPECIFICATION

Personal Requirements

- Skills and Expertise.
- Time Management.
- Continuous Learning.
- Collaboration and Communication.
- Project Management

TECHNOLOGIES TO BE USED

AI/ML Freameworks:
Tensorflow, Pytorch, Keras

Development Tools:
IDEs: VSCode, PyCharm,
Jupyter Notebook

Cloud Services:
Microsoft Azure

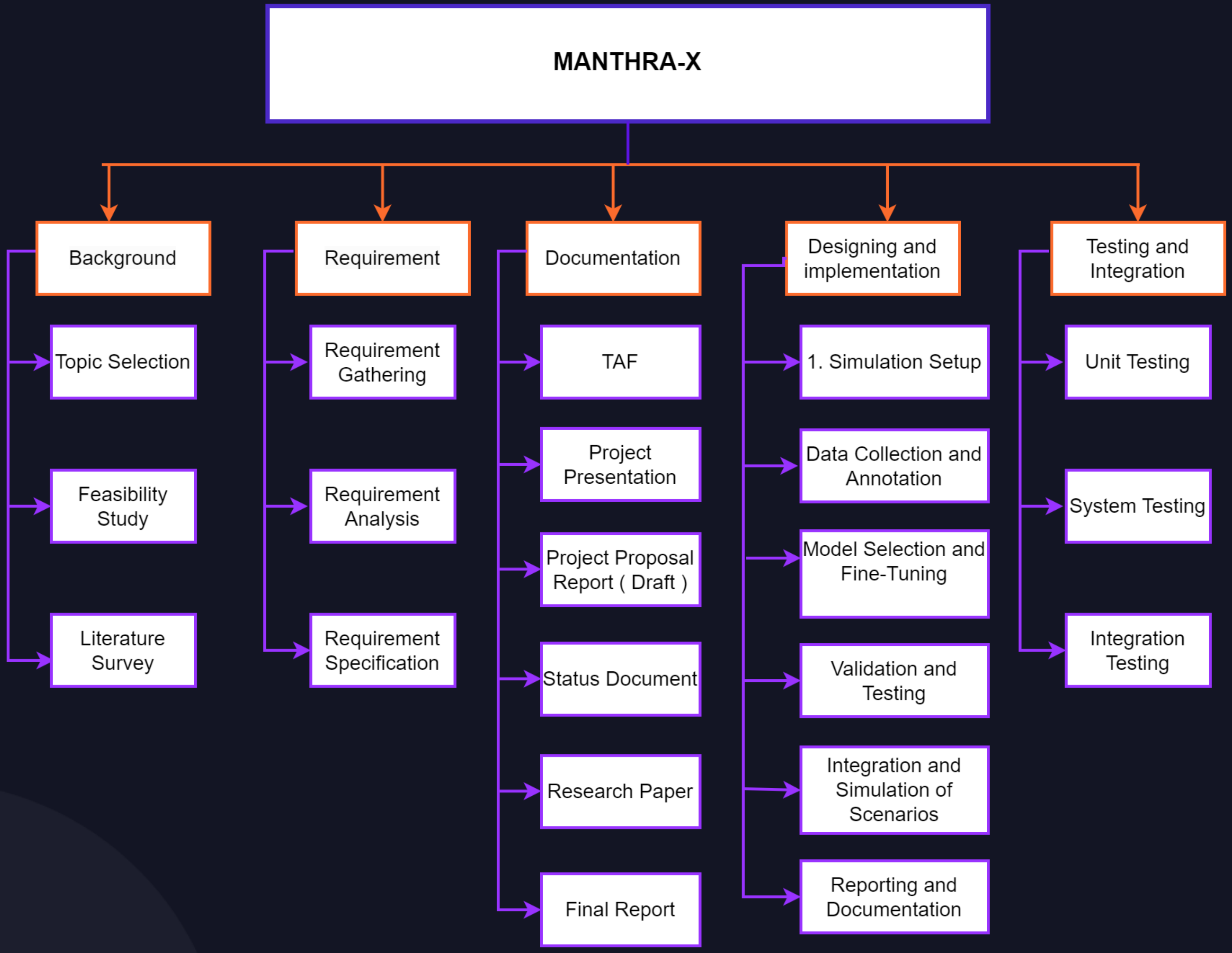
mongoDB

Databases:
NoSQL Databases:
MongoDB, Cassandra

Simulation Frameworks:
CARLA, AirSim, or Unity

Web Frameworks: Node.js,
Flask, Django

WBS



MANTHRA-X
Background
Topic Selection
Feasibility Study
Literature Survey
Requirement
Requirement Gathering
Requirement Analysis
Requirement Specification
Documentation
TAF
Project Presentation
Project Proposal Report (Draft)
Status Document
Research Paper
Final Report
Designing and implementation
1. Simulation Setup
Data Collection and Annotation
Model Selection and Fine-Tuning
Validation and Testing
Integration and Simulation of Scenarios
Reporting and Documentation
Testing and Integration
Unit Testing
System Testing
Integration Testing

IT21155048|GANEPOLA G.A.N.B.

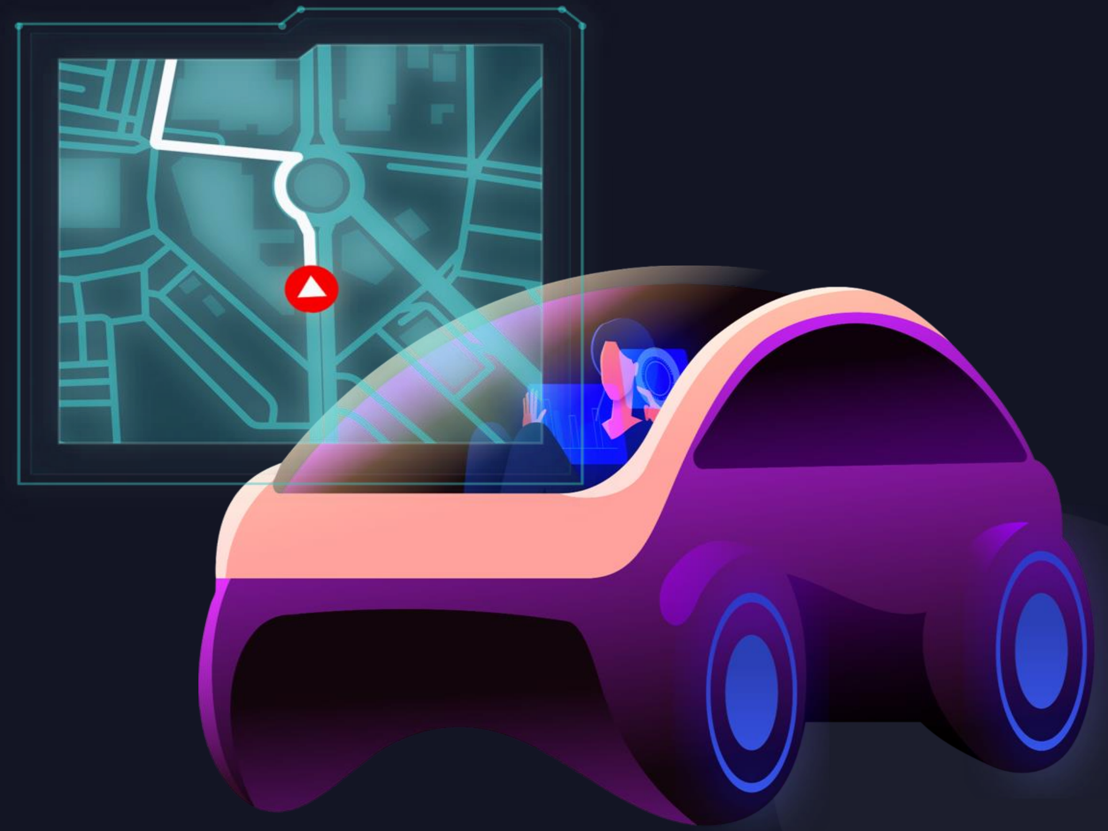
BSc (Hons) in Information Technology (specialization in Data Science)

Component 02

DECISION MAKING AND COLLISION AVOIDANCE

BACKGROUND

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



Why Advanced Decision Making is important?

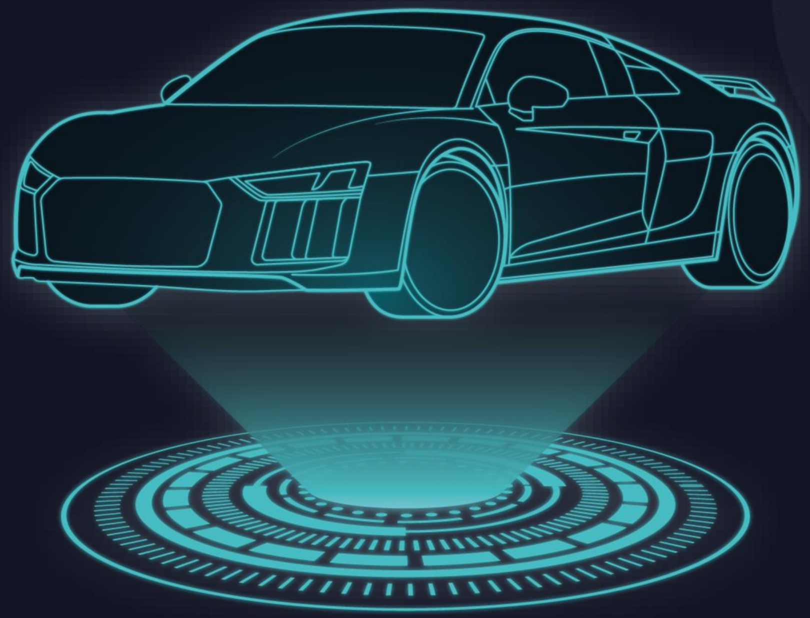
Complex Traffic Scenarios

Safety Enhancement

Human Like Decision Making

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.

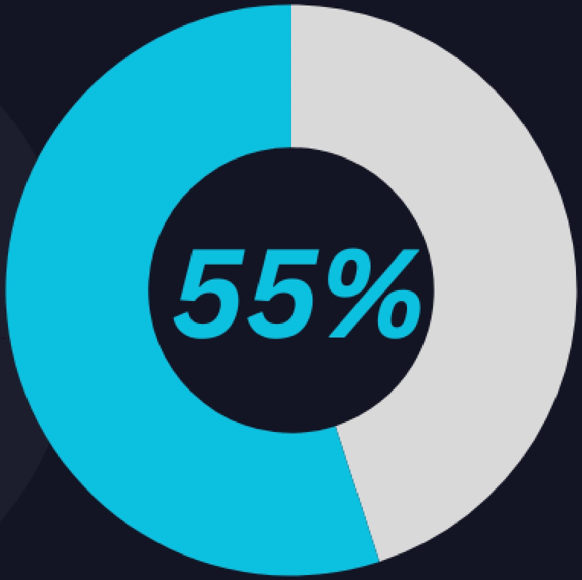
COLLISION AVOIDANCE MECHANISM

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



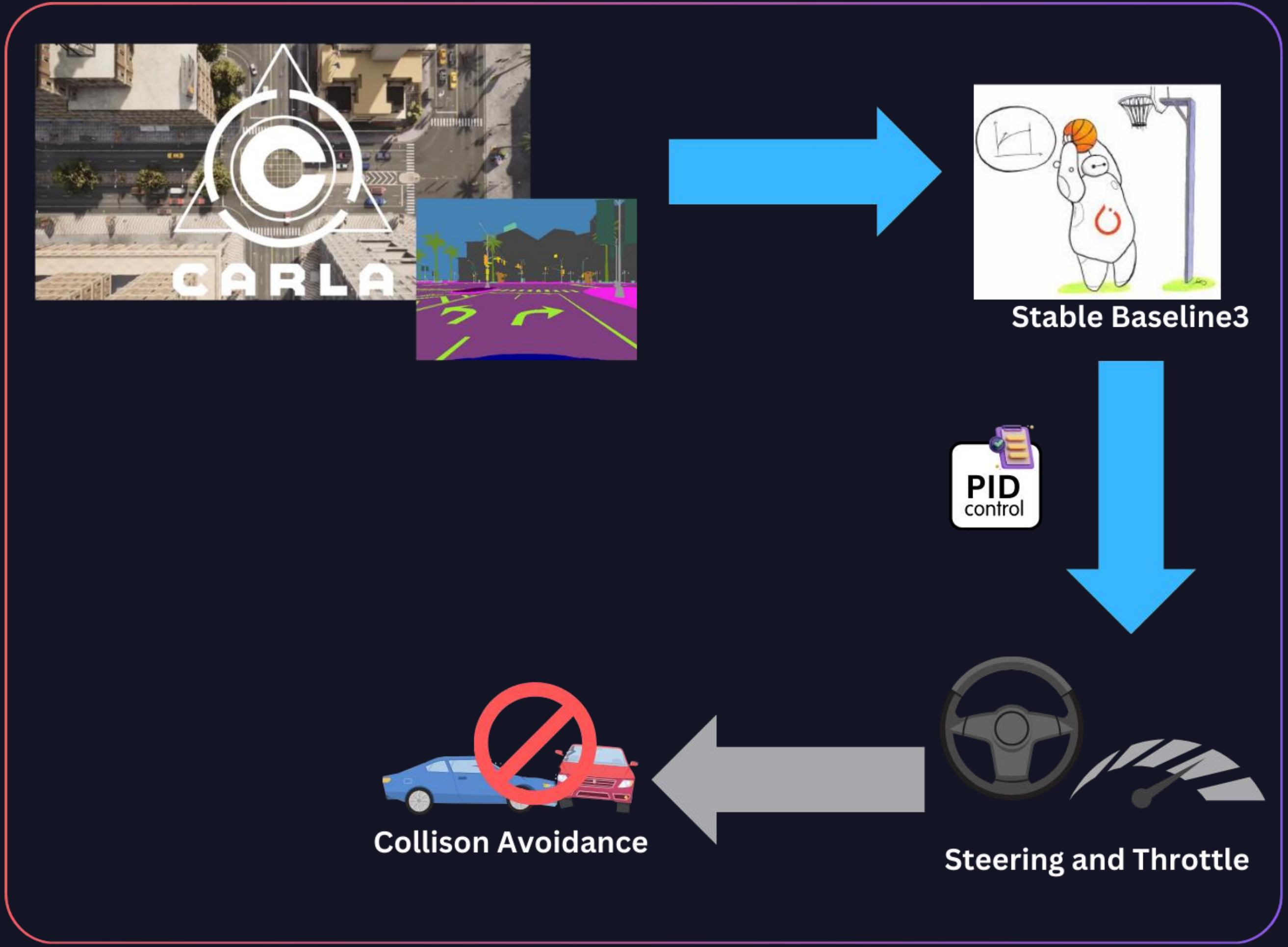
Completed Tasks

- CARLA Environment Setup
- Reinforcement Learning Model
- Path Planning

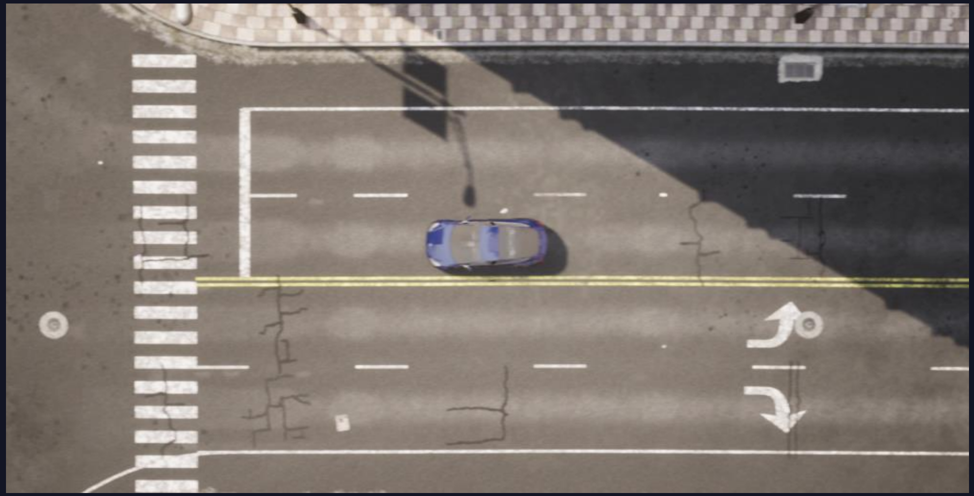
Upcoming Tasks

- Collison Avoidance Mechanism
- System Integration

Progress Diagram



Training Process



```
-----------------------------------
| rollout/           |             |
|    ep_len_mean     | 459         |
|    ep_rew_mean     | 1.12e+03    |
| time/              |             |
|    fps             | 26          |
|    iterations      | 1           |
|    time_elapsed    | 76          |
|    total_timesteps | 751616      |
-----------------------------------
```

```
Action taken: [6], Reward: 7.164316495856956, Done: False
Action taken: [6], Reward: 7.264937365217697, Done: False
Action taken: [6], Reward: 7.379717706279149, Done: False
Action taken: [6], Reward: 7.446442323399609, Done: False
Action taken: [6], Reward: 7.52512246800857, Done: False
Action taken: [6], Reward: 7.579921236753064, Done: False
Action taken: [6], Reward: 7.68344535321228, Done: False
Action taken: [6], Reward: 7.829452801046496, Done: False
Action taken: [6], Reward: 7.91307982605043, Done: False
```

- The Proximity Policy Optimization model from Stable Baselines 3 is used, which is an advanced policy optimization method for RL

Reward and Penalty

- Rewards List:
  - Waypoint Reached Reward: +300 for reaching the waypoint.
  - Proximity to Waypoint Reward: Up to +10 for staying closer to the waypoint.
  - Speed Maintenance Reward: Positive reward for maintaining speed close to the preferred value (within 5 units).

- Penalties List:
  - Collision Penalty : -300 for any collision.
  - Lane Invasion Penalty : -300 for crossing restricted lane markings (e.g., solid lines).
  - Angle Deviation Penalty: -300 if the angle difference exceeds 18 degrees.
  - Scaled penalty for smaller deviations.
  - Distance Penalty:-300 for distances greater than 4 meters from the desired path.
  - Scaled penalty for smaller deviations above 0.5 meters.
  - Overspeed Penalty: -300

PID Controller

- 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

TECHNOLOGIES

CARLA

Gymnasium

jupyter

REQUIREMENT SPECIFICATION

System Requirements

- **High-Performance GPU: For running CARLA and training RL models**
- **Multi-Core CPU: For managing simulation and computational tasks simultaneously.**
- **RAM: At least 16GB for efficient model training and simulation.**
- **Disk Space: Minimum 50GB for models and simulation files.**
- **Networking :** High speed internet connection for real time data transfer, communication and for downloading dependencies and updates.

REQUIREMENT SPECIFICATION

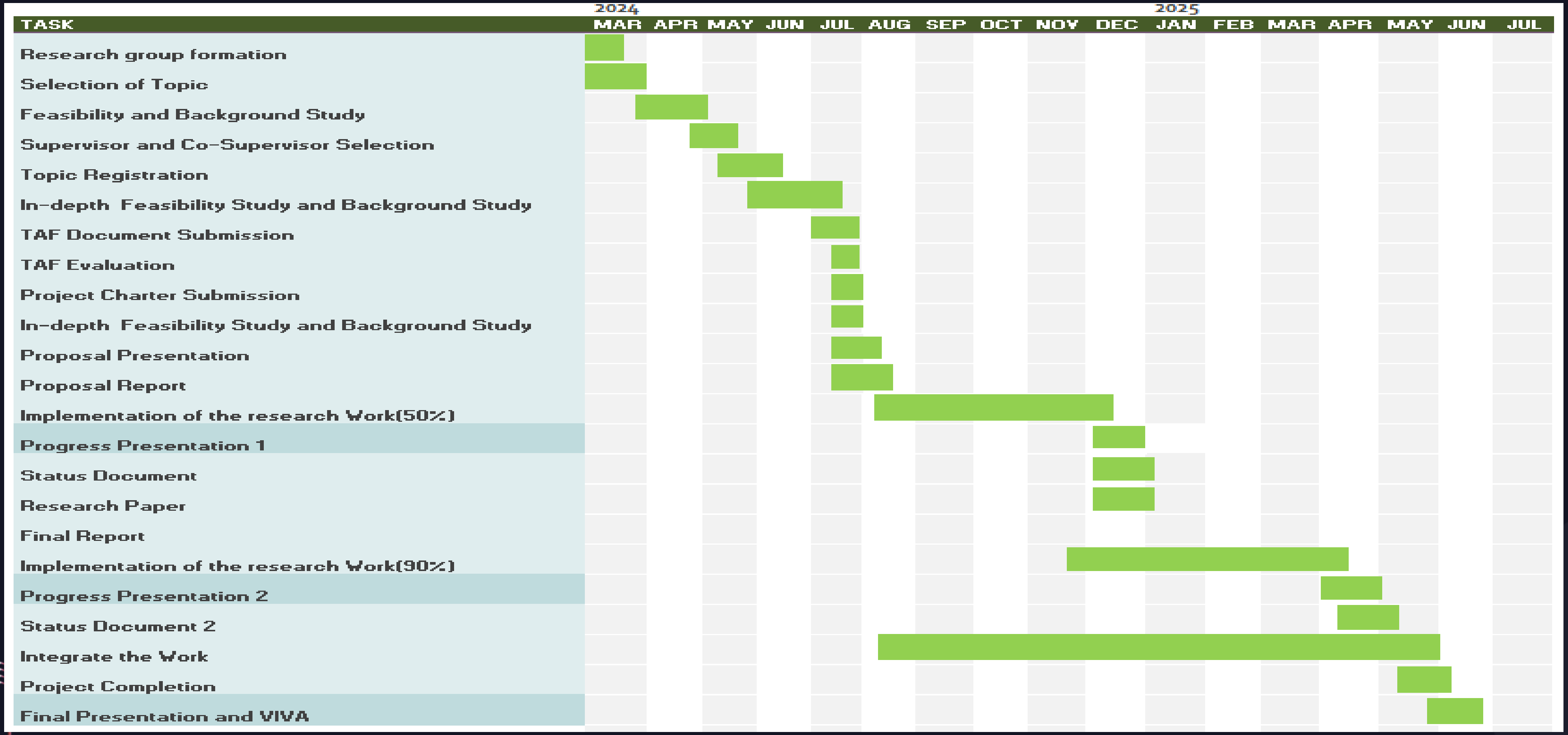
Personal Requirements

- Technical Proficiency
- Experience with Tools and Frameworks
- Problem Solving Abilities
- Teamwork and Communication
- Project Management
- Adaptability
- Time Management Skills

CHALLENGES ENCOUNTERED

- Running CARLA requires high computational power, which can slow down the simulation.
- Training RL models is computationally expensive and time-consuming.
- Compatibility issues with Python libraries and CARLA versions.

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] H. Wang et al., "Risk Assessment and Mitigation in Local Path Planning for Autonomous Vehicles With LSTM Based Predictive Model," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 4, pp. 2738-2749, Oct. 2022, doi: 10.1109/TASE.2021.3075773.

[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)

Component 03

Ethical Decision Making in Autonomous Vehicles

Enhance Ethical Decision Making by Integrating Ethical Frameworks, Driver Attention and Drowsiness Monitoring, Hazard Detection and Object Prioritization.

Safety

Fairness

Adaptivity in autonomous driving

BACKGROUND

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	✓
Multi-Criteria Decision-Making Approaches	No	No	No	Yes	Yes	✓

RESEARCH PROBLEM

How can ethical decision-making models be developed that accurately reflect individual driver preferences?

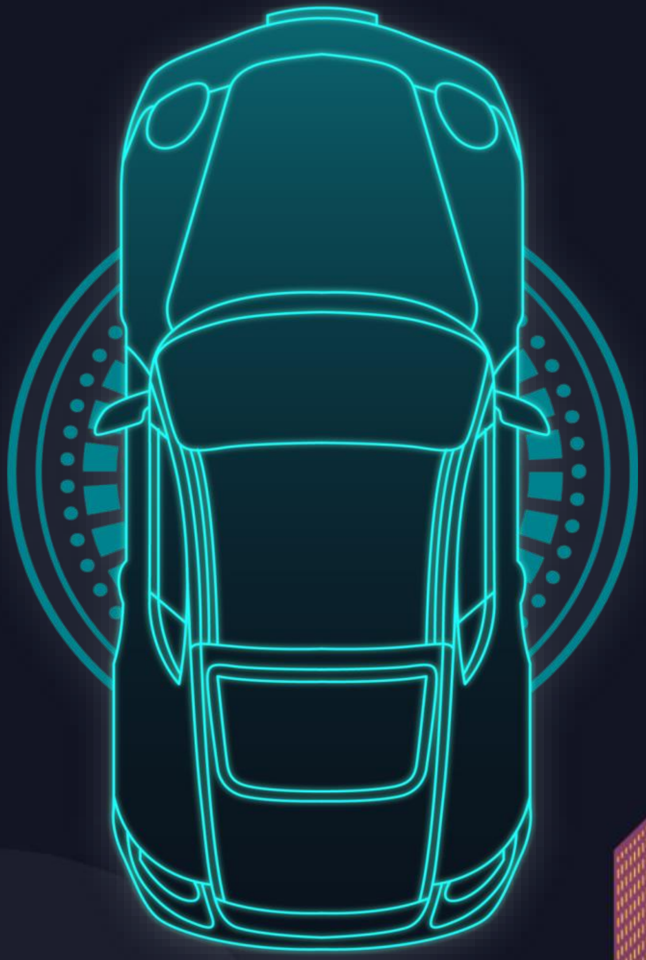
What factors should be considered when prioritizing objects in complex driving scenarios, and how can these be integrated?

How can eyeball tracking systems be designed and implemented to reliably monitor driver attention and contribute to ethical decision-making in autonomous vehicles?

How can machine learning techniques enhance the detection of critical hazards near the vehicle, ensuring informed and ethical responses to minimize risk?

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
- Hazard detection
- Integrate Ethical Principles and Frameworks to Develop Decision-Making Techniques for Object and Road User Prioritization.
- Enhance Transparency and User Trust

SUB OBJECTIVES

➢ 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
- 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

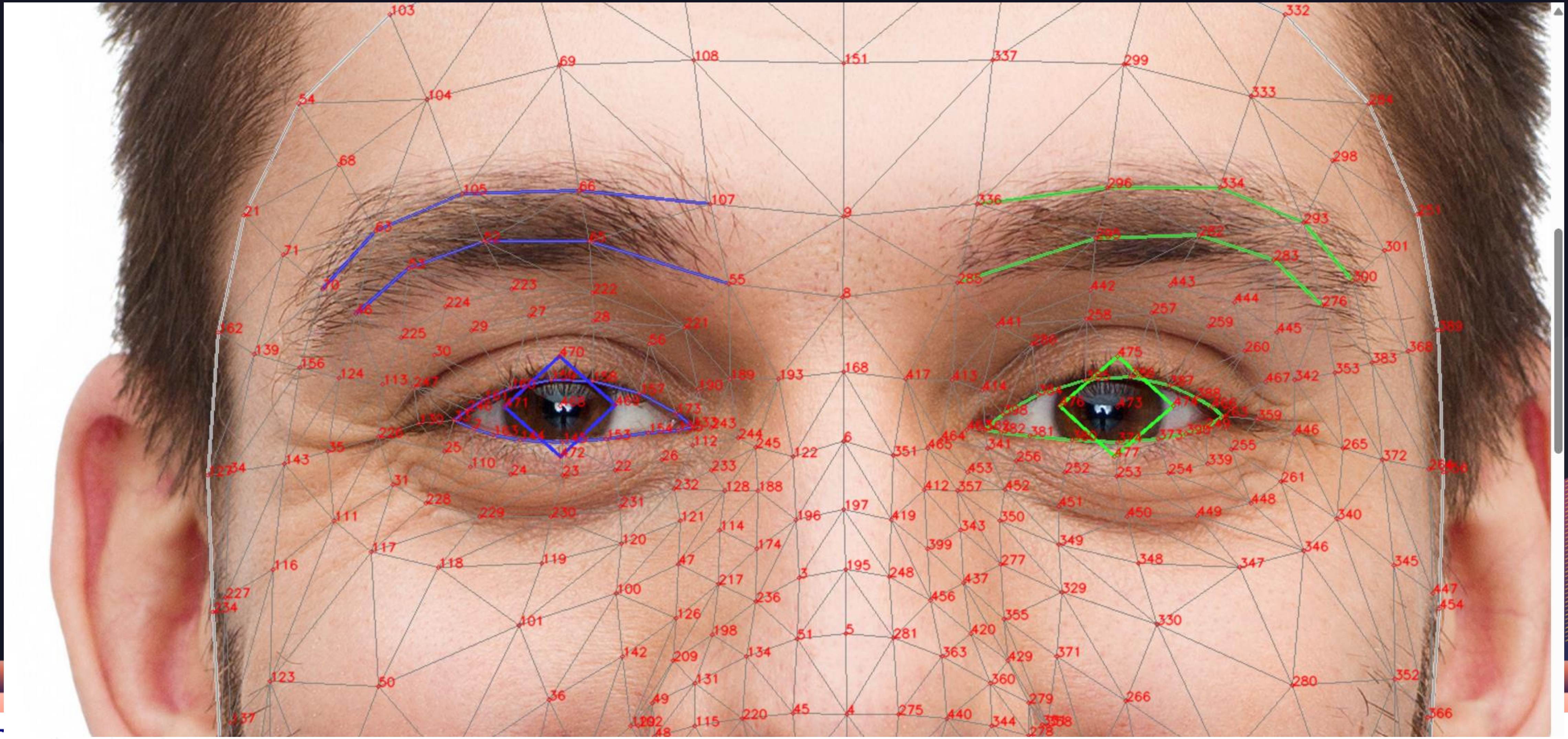
Eye Closure Detection

```
# 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
```

```
# 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

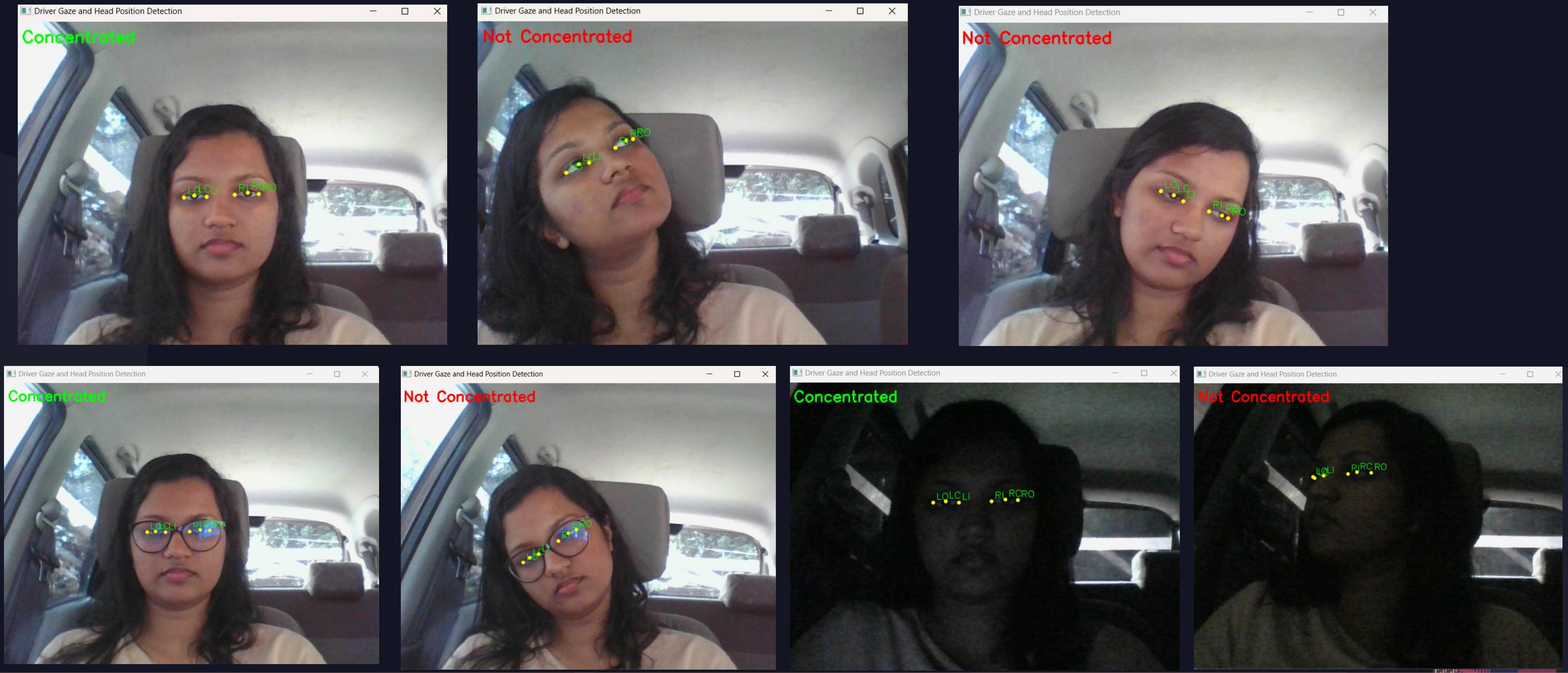
[Face landmark detection guide | Google AI Edge | Google AI for Developers]



Testing

SUB OBJECTIVES

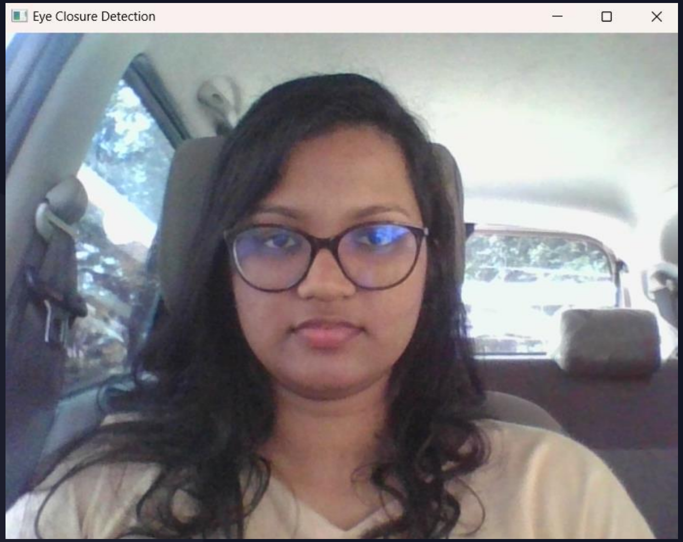
Driver Gaze and Head Position Detection

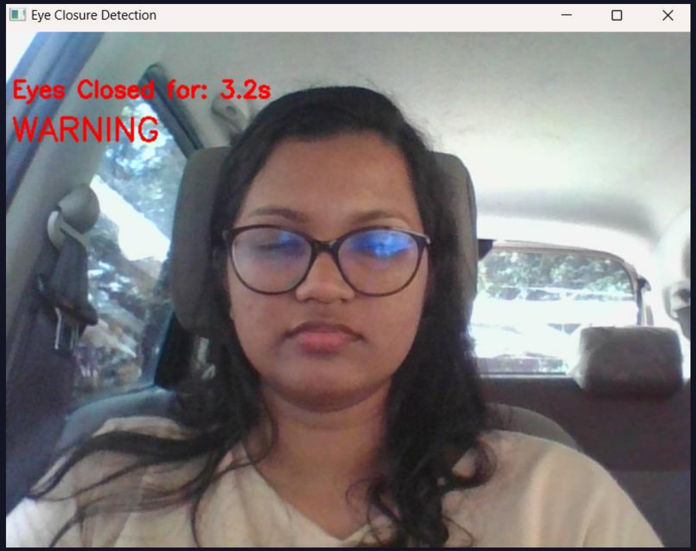


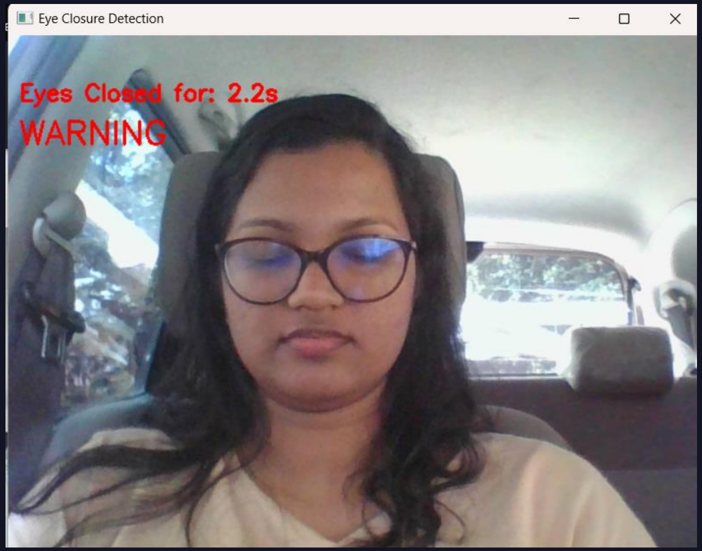
Testing

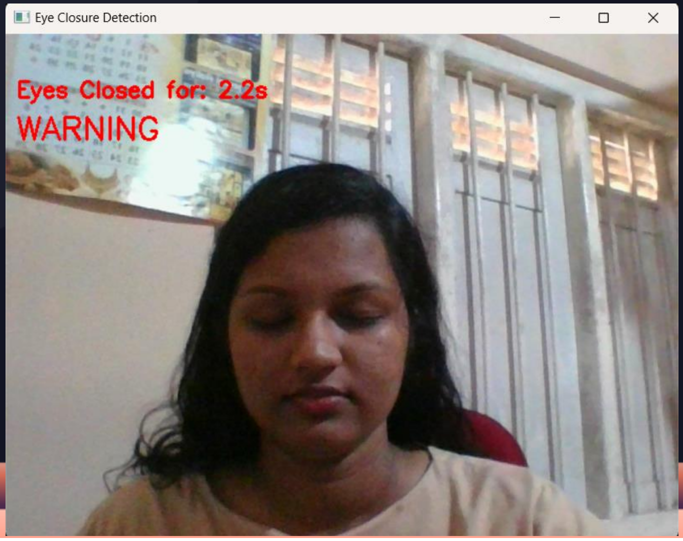
SUB OBJECTIVES

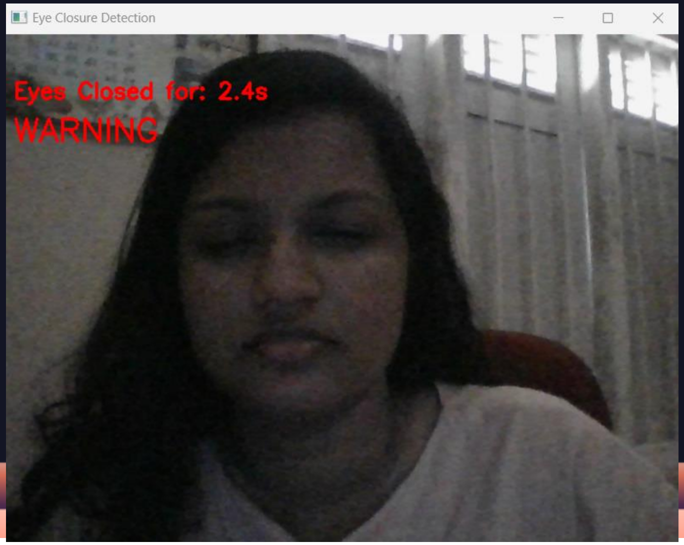
Eye Closure Detection











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

SUB OBJECTIVES

➢ Hazard detection

- Develop a hazard detection system that identifies potential hazards (e.g., persons, pets, potholes, cones) before the vehicle starts.
- Enhance driver and pedestrian safety by preemptively identifying obstacles in the vehicle's surroundings.

Key Features:

- Multiclass Detection
- Real-Time Processing
- Integration Flexibility

Methods and Technologies

- YOLOv8 Model Training
- Cameras (RGB/Monochrome)

SUB OBJECTIVES

Data Gathering and Preprocessing

- **Roboflow**

Roboflow offers high-quality, pre-labeled datasets optimized for machine learning tasks.

Datasets Used for Model Training:

- Adult/Child Dataset.
- Cone Dataset.
- Pothole Dataset.
- Cat/Dog Dataset

- Class Balancing Dataset: - COCO Dataset

Purpose: Used to balance classes and enrich training with diverse examples of hazards.

SUB OBJECTIVES

Data Gathering and Preprocessing

Steps Involved:

- XML to TXT Conversion
- Class Renaming and Unification
- Dataset Splitting
- Data Augmentation
- Class Balancing
- File Renaming for Consistency
- Test Set Creation

Aggregating the Final Dataset

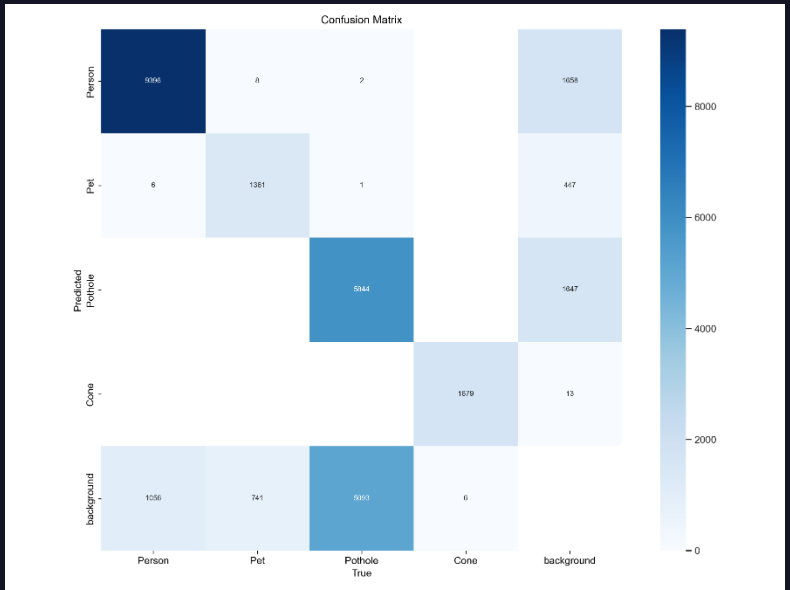
Combining Datasets from Various Sources

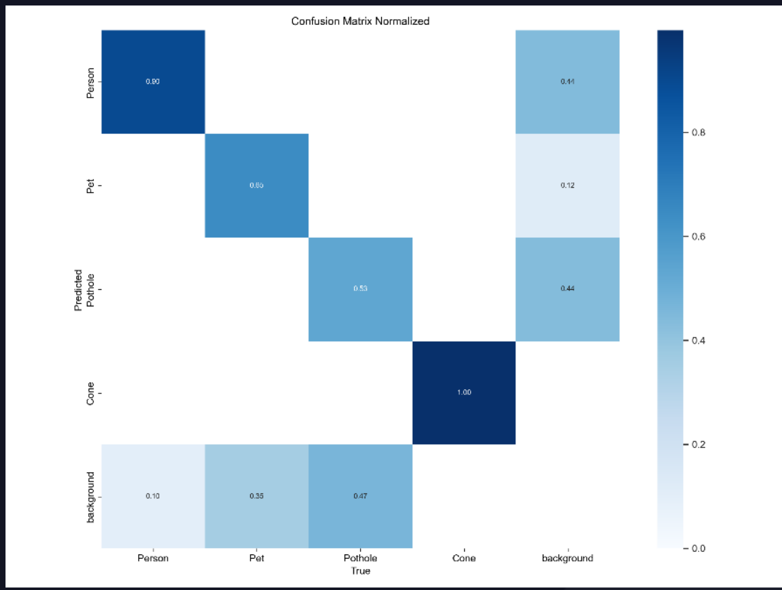
- To create a robust and diverse dataset, images were aggregated from multiple sources such as Roboflow, Coco, and other tested datasets.

SUB OBJECTIVES

Model Training

```python
from ultralytics import YOLO

model = YOLO('yolov8s.pt')

# Load the model with the trained weights
model = YOLO('runs/detect/yolov8s_sample/weights/best.pt')

# Training.
results = model.train(
    data='Full_Dataset/data.yaml',
    imgsz=320,
    epochs=10,
    batch=32,
    name='yolov8s_sample_continue'
)
```

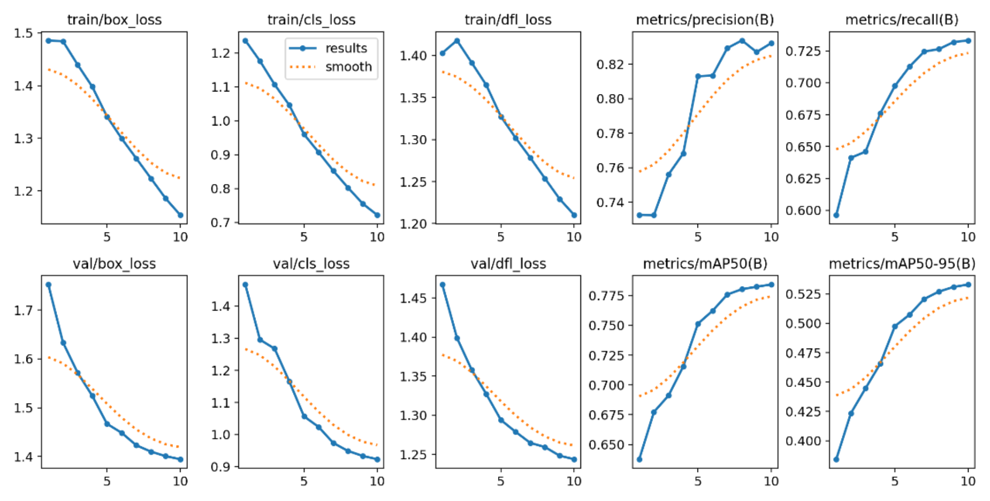


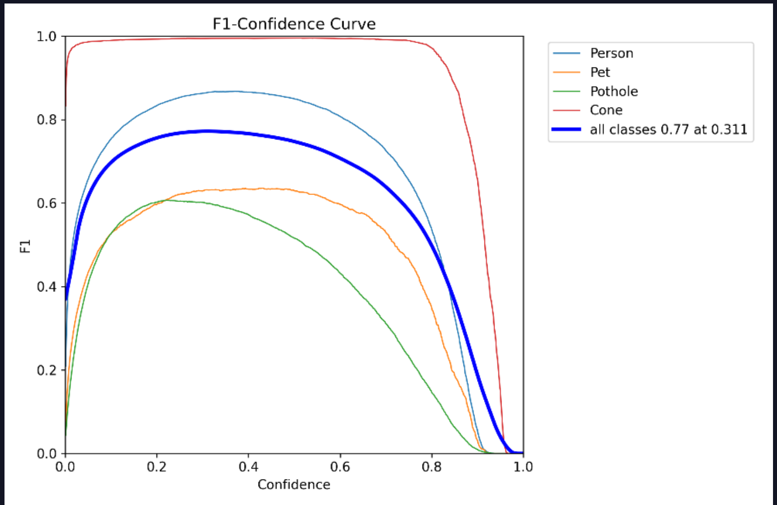


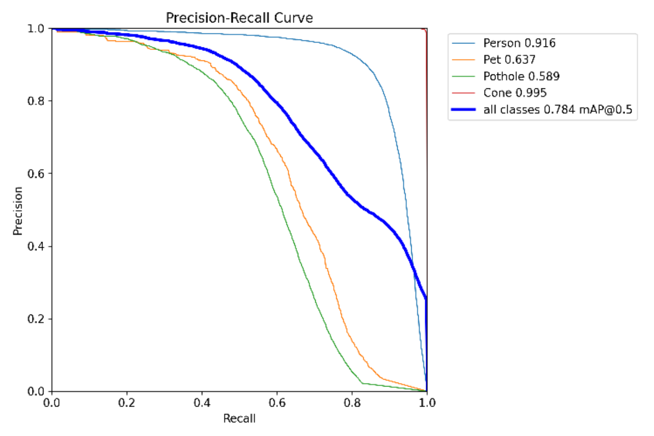

```
Model summary (fused): 168 layers, 11,127,132 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 147/147 [01:09<00:00,  2.10it/s]
                   all       9366      25193      0.832      0.733      0.784      0.533
                Person       2747      10458       0.87       0.86      0.916      0.586
                   Pet       1943       2110      0.671      0.596      0.637      0.339
               Pothole       3438      10940      0.794      0.481      0.589        0.3
                  Cone       1119       1685      0.994      0.995      0.995      0.907
Speed: 0.1ms preprocess, 0.9ms inference, 0.0ms loss, 1.4ms postprocess per image
```

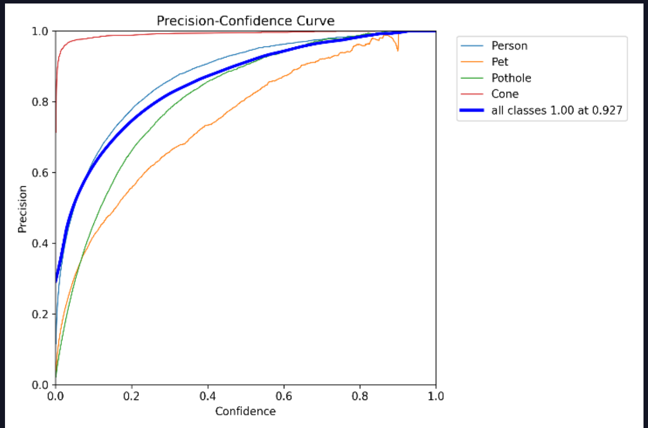
SUB OBJECTIVES

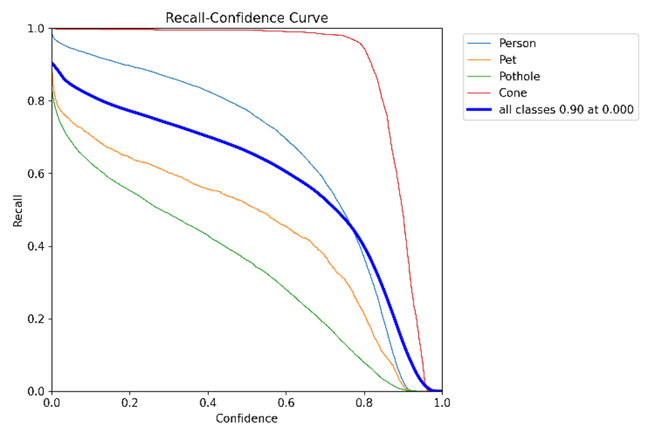
Model Training





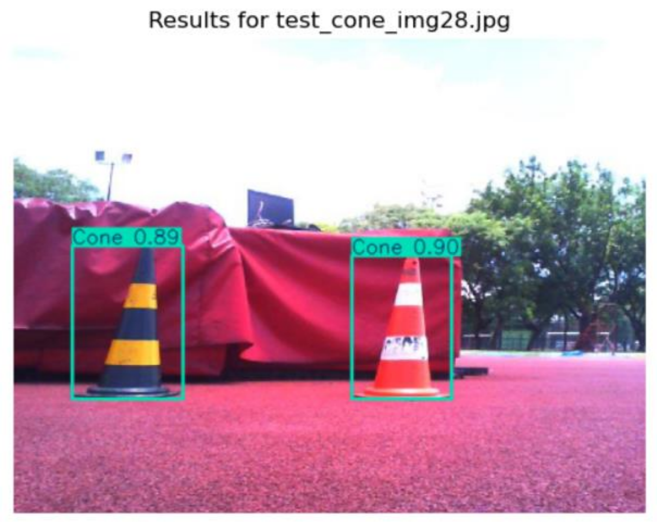






SUB OBJECTIVES

Testing



Why Choose ?

YOLOv8 (You Only Look Once)

- superior speed and accuracy.
- Real-time detection capabilities for dynamic hazard detection.
- Ease of training and deployment.
- Proven performance in similar real-world scenarios.
- Scalability and flexibility

YOLOv8s is a fast, lightweight, and efficient

Remaining Tasks

- Integrate Ethical Principles and Frameworks to Develop Decision-Making Techniques.
- Real-World Hazard Detection Using IoT Devices.
- Mobile Application-Based Alerting System.
- Testing.

50%
Completed

METHODOLOGY

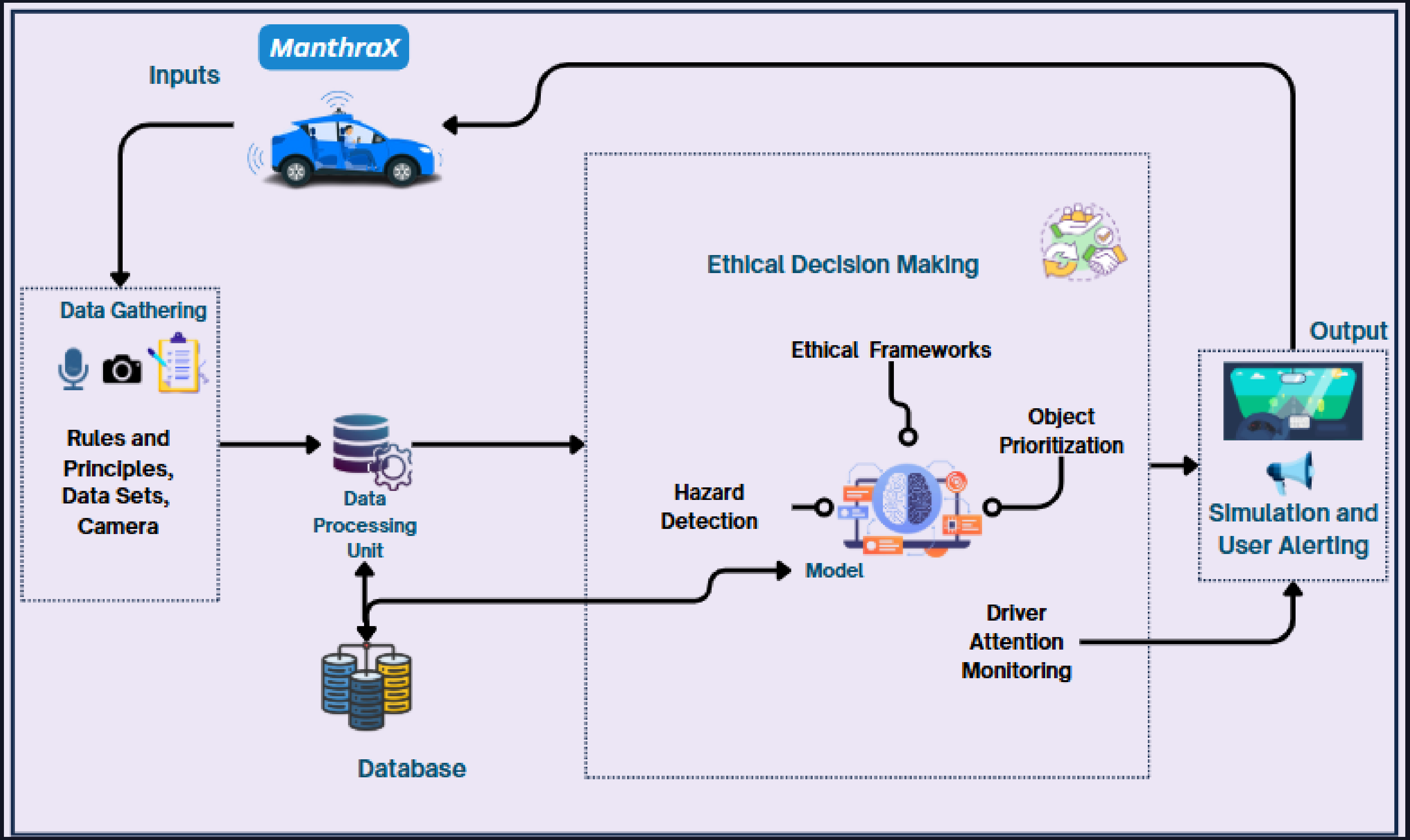
SYSTEM DIAGRAM

REQUIREMENT SPECIFICATION

- System Requirements
- Software Requirements
- Personal Requirements

TECHNOLOGIES TO BE USED

SYSTEM DIAGRAM



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

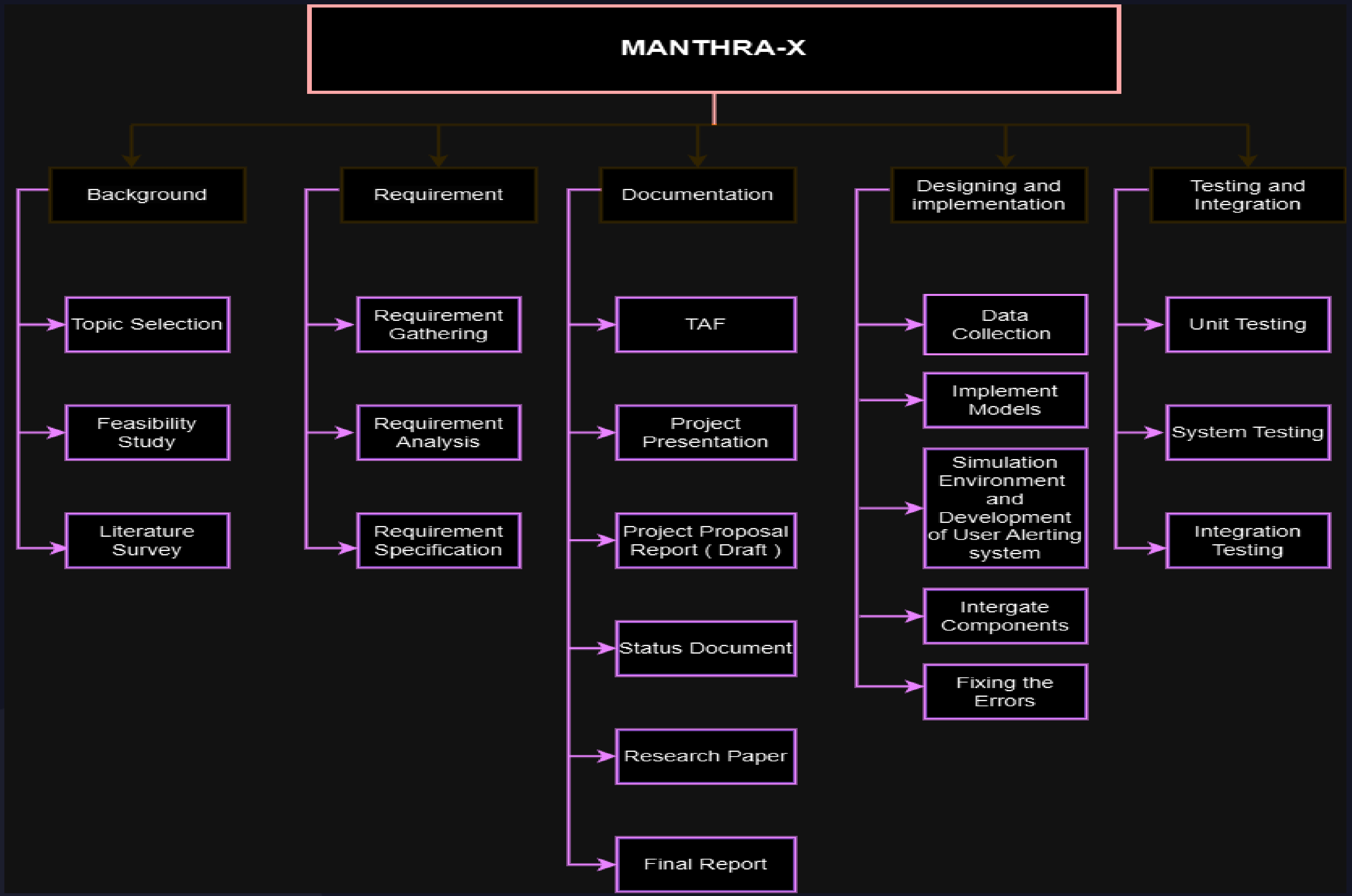
AI/ML Freameworks:
Tensorflow, Pytorch, Keras

Development Tools:
IDEs: VSCode, PyCharm,
Jupyter Notebook

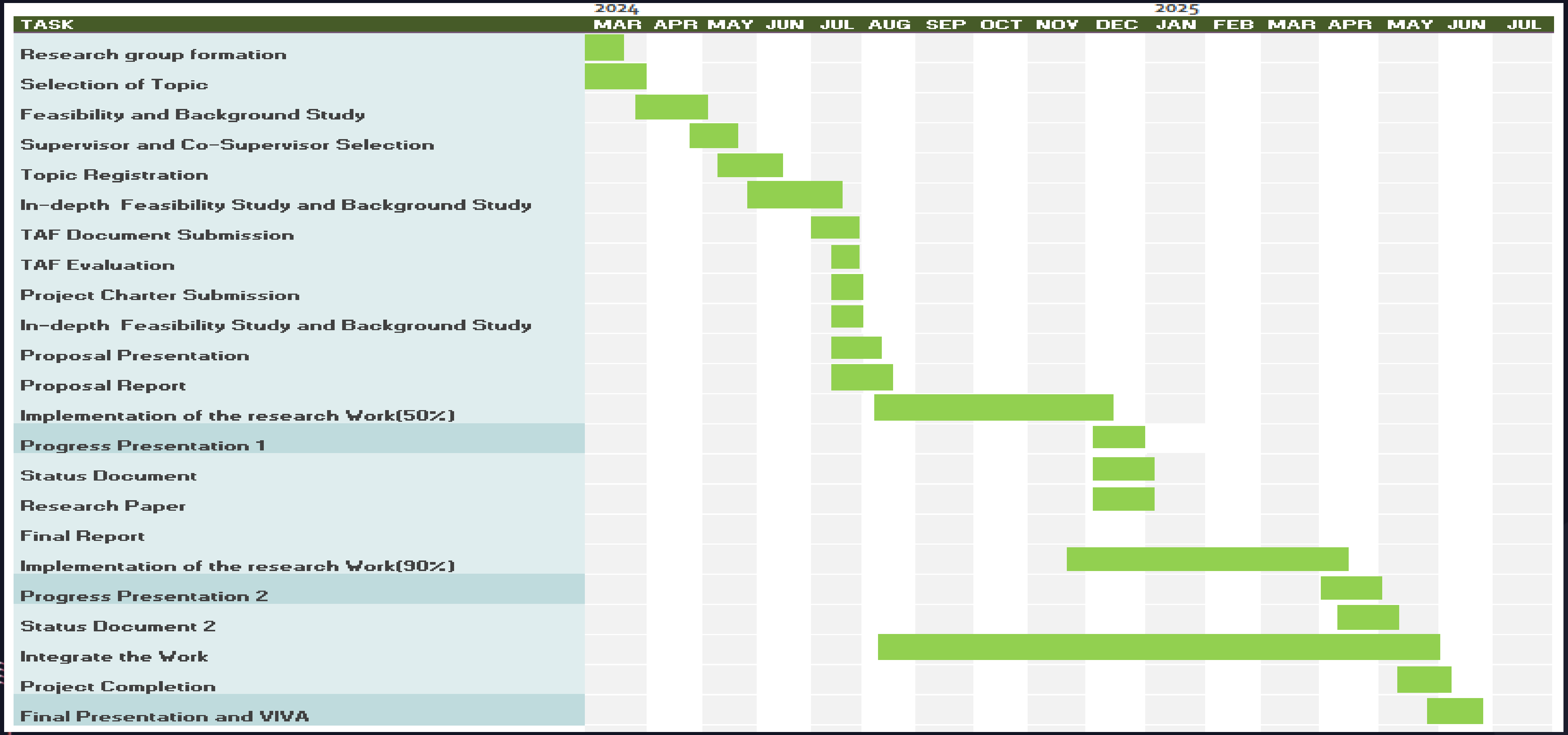
Simulation Frameworks:
CARLA, AirSim, or Unity

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

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

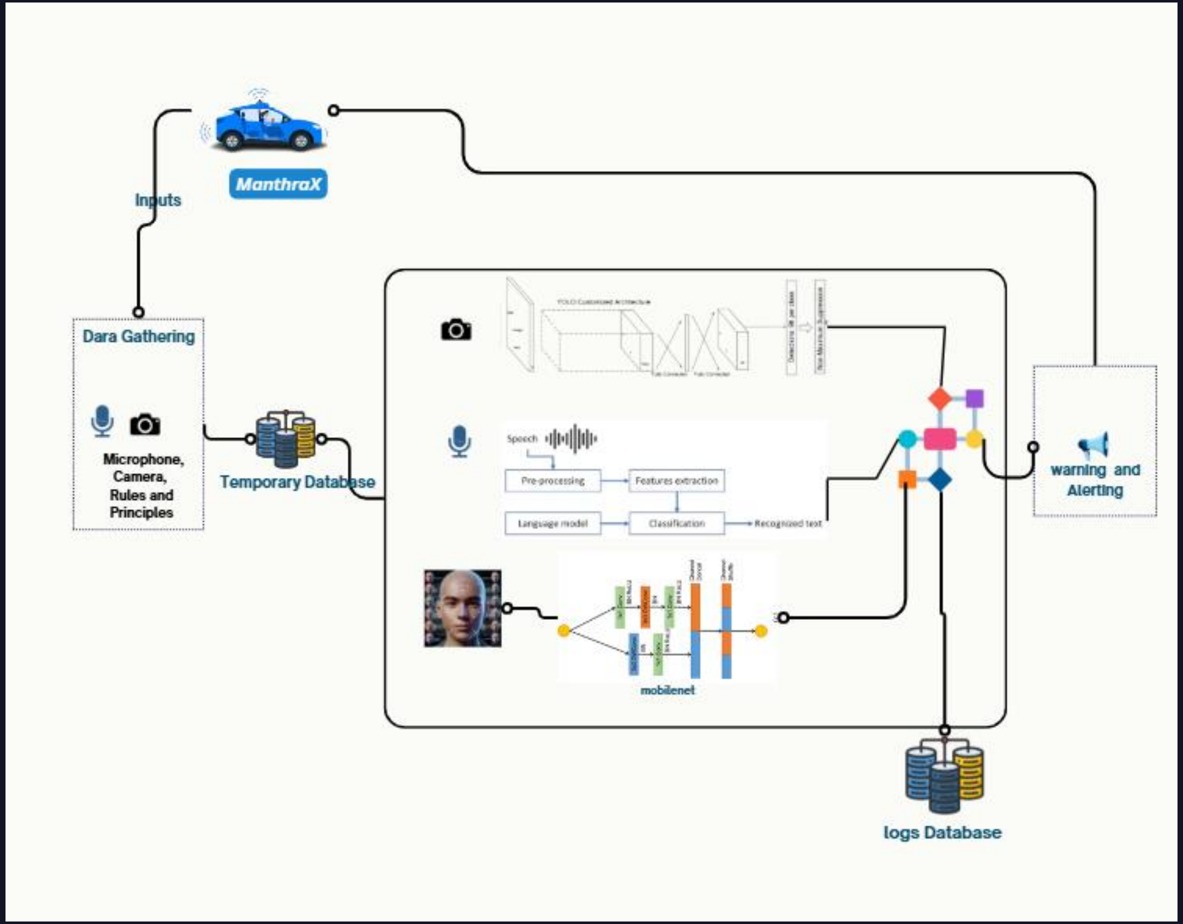
Identifying unauthorized voices or keywords in an environment with multiple overlapping voices.

Synchronizing object and voice detection to accurately identify unauthorized activities.

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

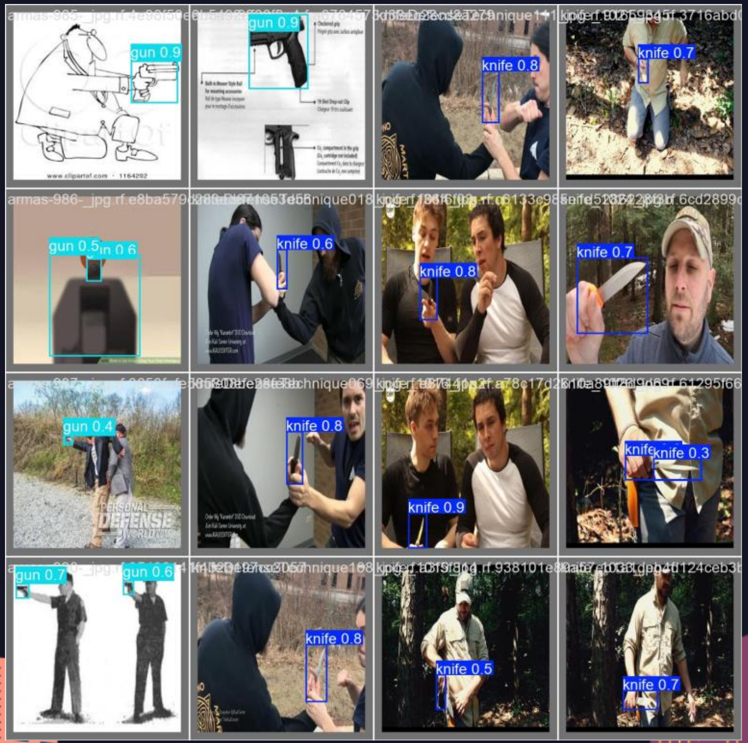
- **Resizing**: Uniform image resizing to 640x640 for consistent input.
- **Augmentation**: Enabled with transformations like flips, scaling, rotations, and cropping.
- **Normalization**: Scales pixel values to a 0–1 range for model compatibility.
- **Batch Loading**: Processes images in batches of 16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

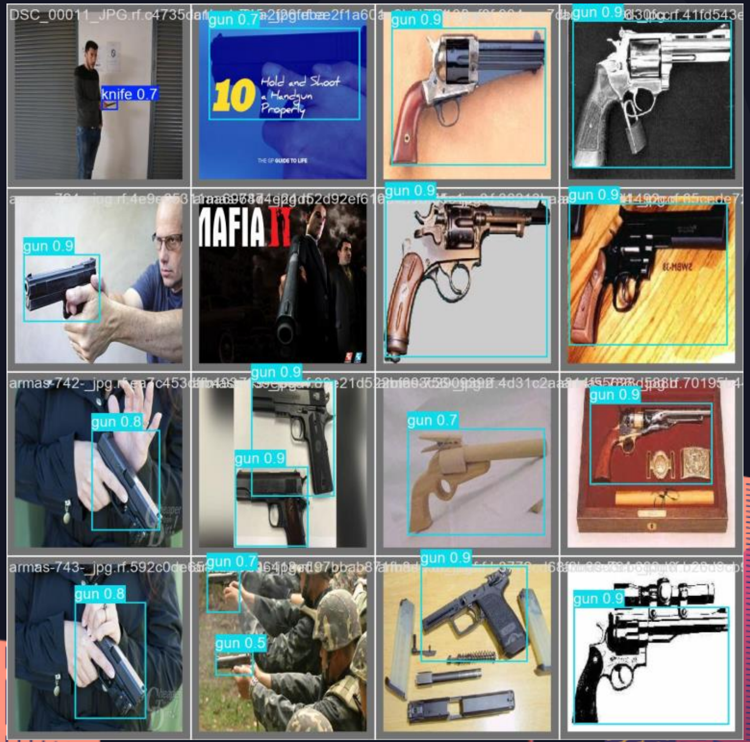
EVIDENCE OF COMPLETION

```
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%**.

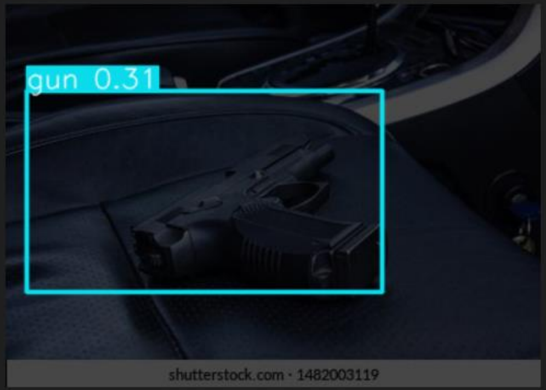
The model demonstrates high accuracy in detecting classes with strong overall performance.





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):
```

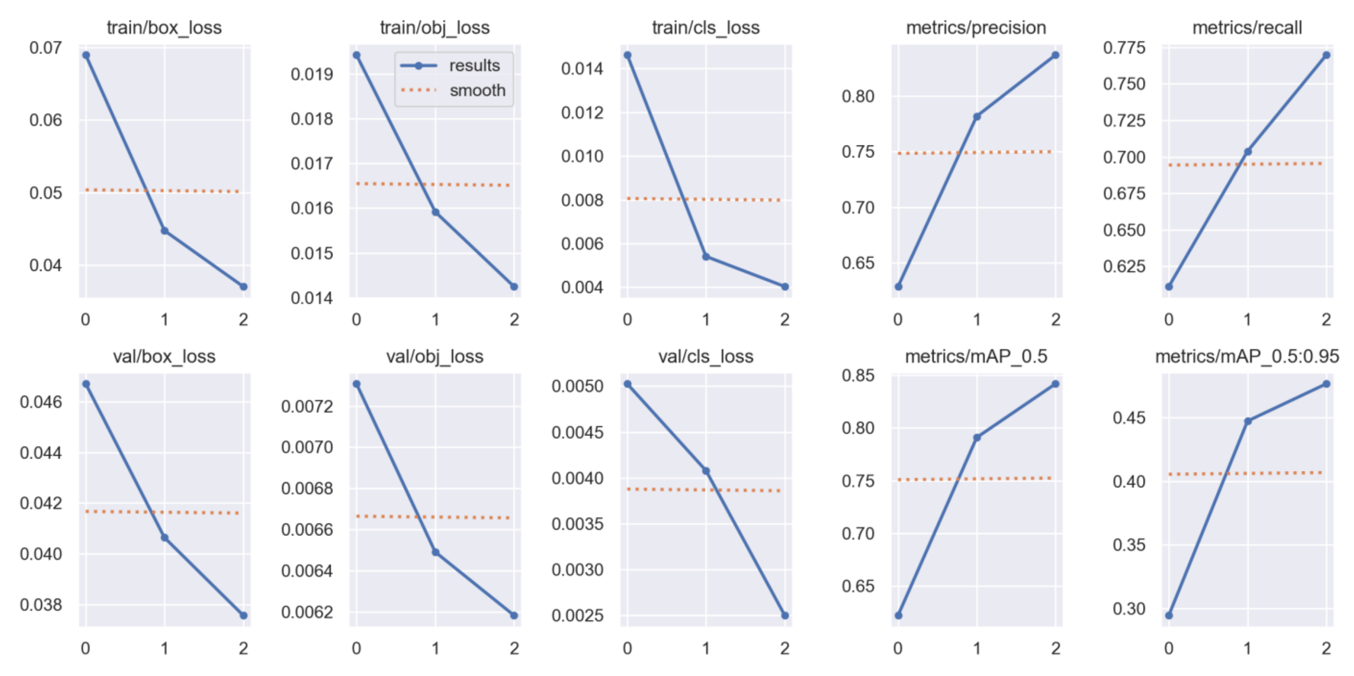


EVIDENCE OF COMPLETION

- **Training Losses**: Decreasing box, objectness, and classification losses indicate effective learning.

- **Validation Metrics**: Reduced losses and improved precision, recall, and mAP show strong generalization.

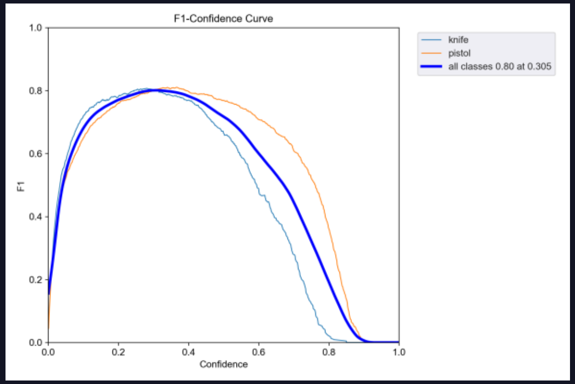
- **Performance Trend**: Continuous improvement across epochs highlights successful optimization.

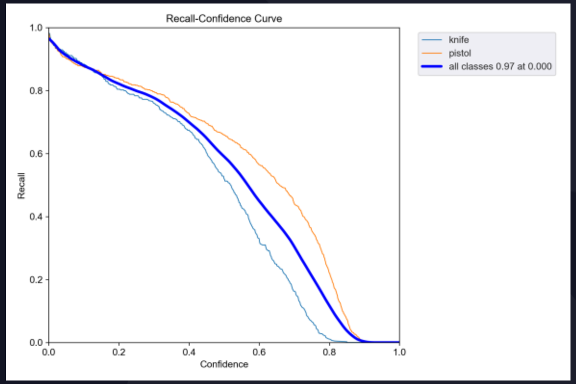


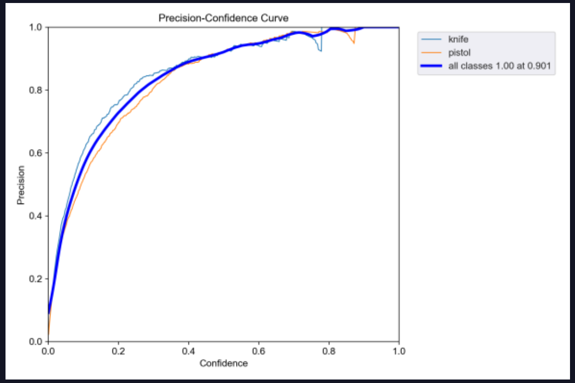
EVIDENCE OF COMPLETION

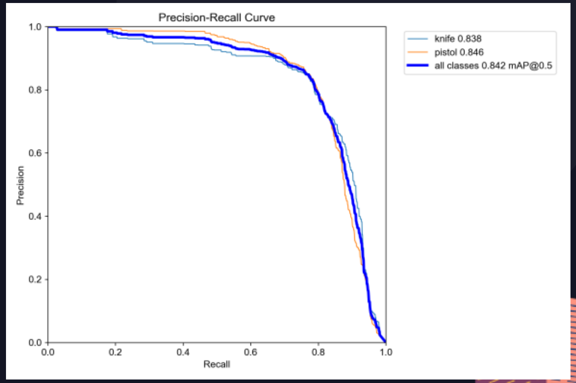
- Precision improves as confidence increases, while recall decreases.

- 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.

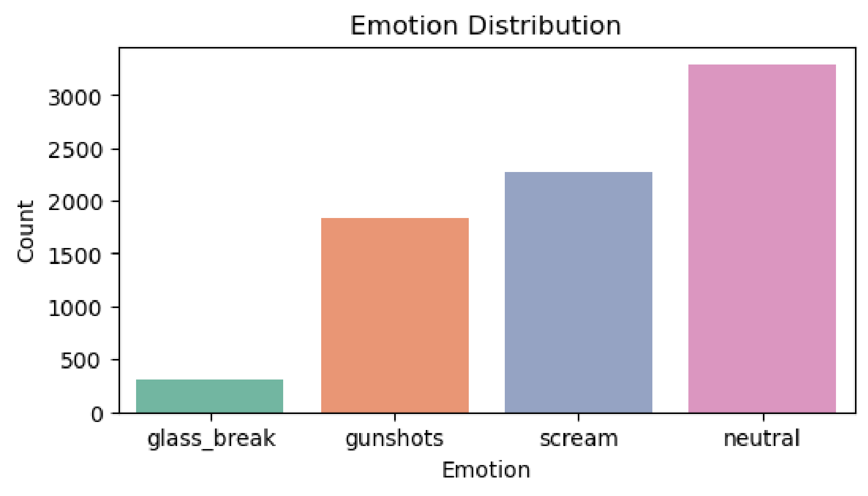
EVIDENCE OF COMPLETION - Data Augmentation

1.Noise Injection

2.Time Stretching

3.Time Shifting

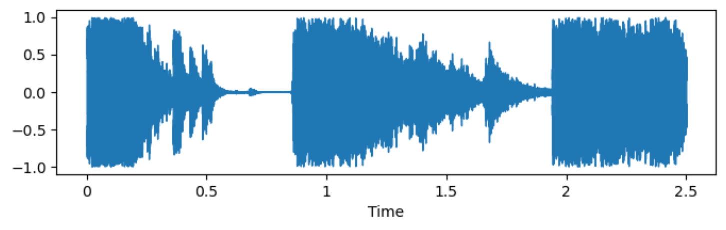
4.Pitch Shifting



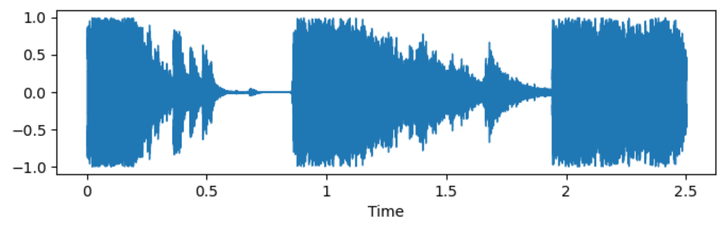
These techniques introduce variability into the training data, helping the model generalize better by mimicking real-world variations in audio data.

EVIDENCE OF COMPLETION

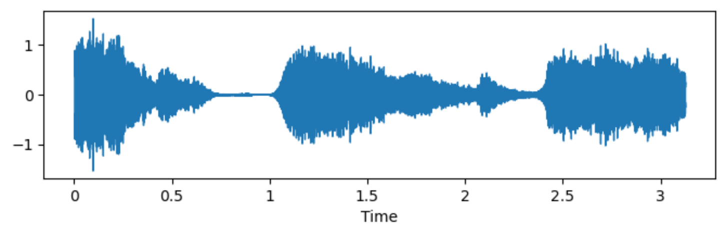
1. Original audio



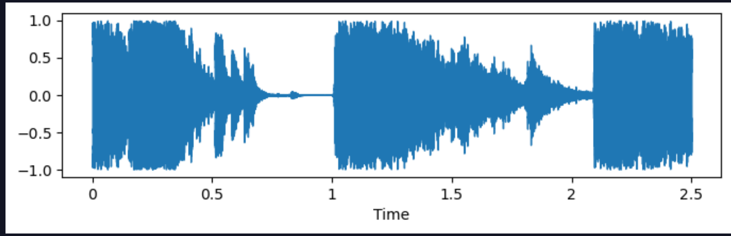
2. Noise injection



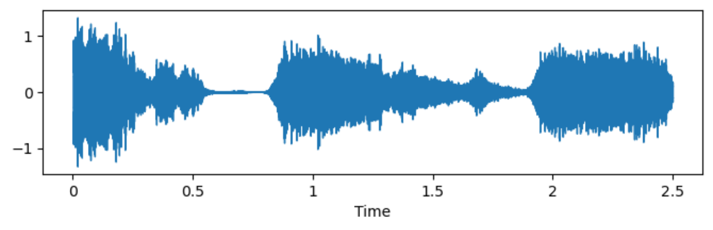
3. stretching



4. shifting



5. Pitch

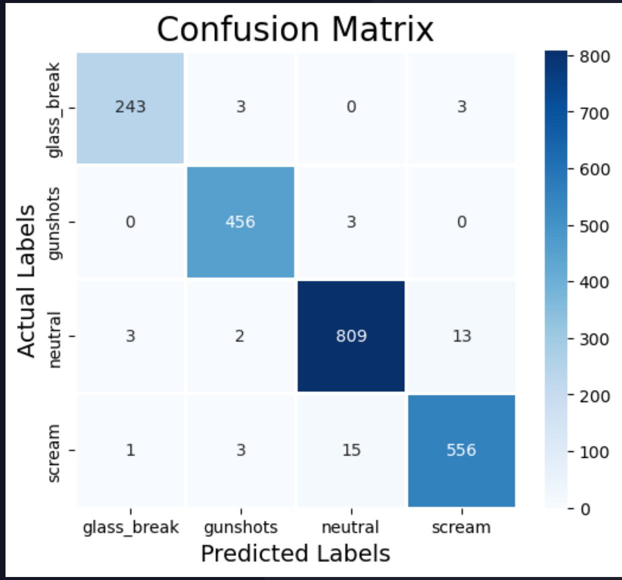


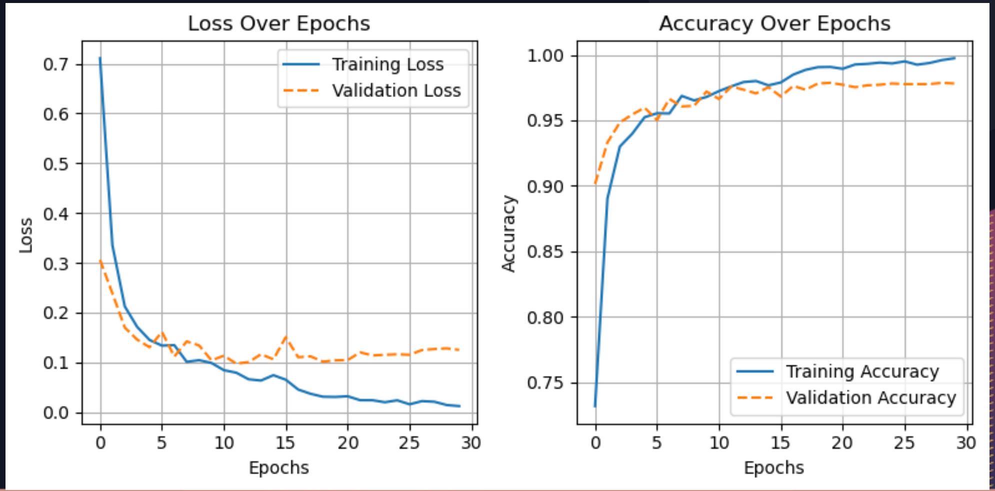
EVIDENCE OF COMPLETION

- 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.

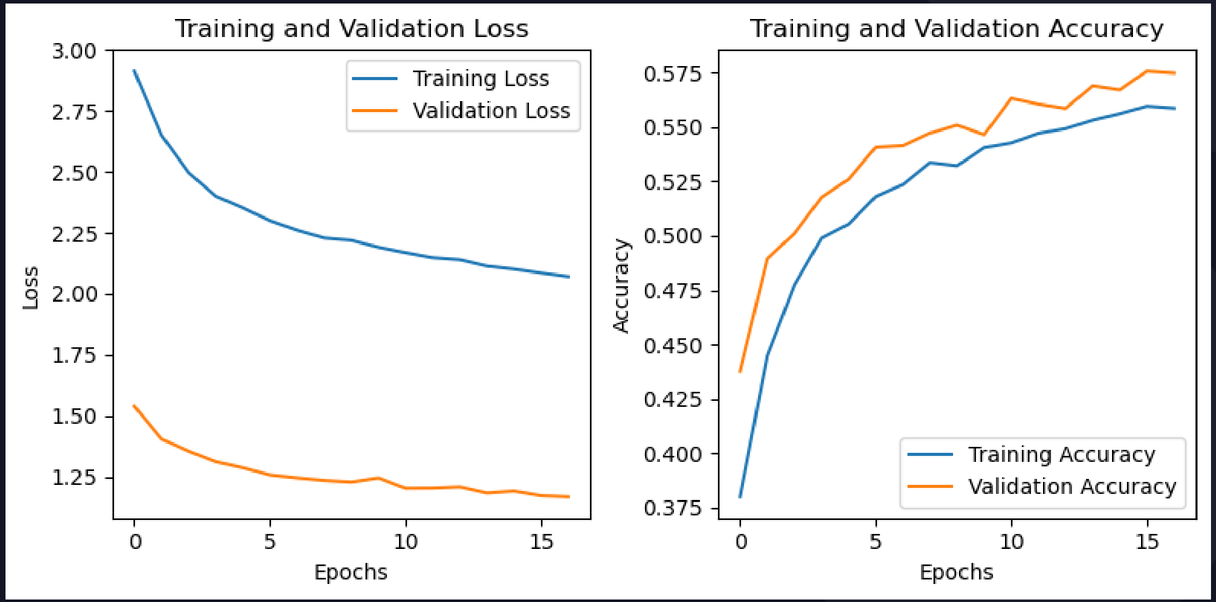
EVIDENCE OF COMPLETION

```
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...



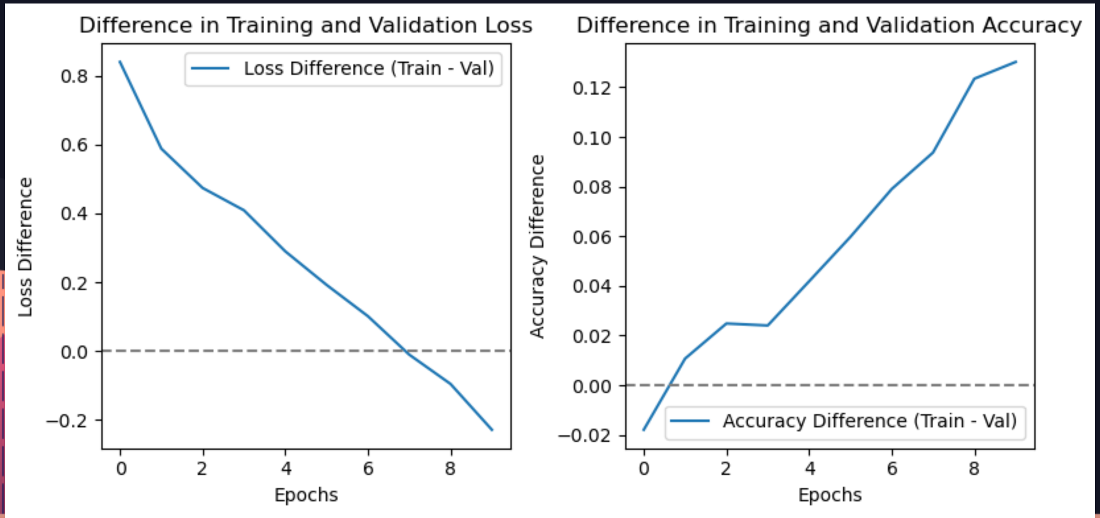
- **Training Loss:** Decreasing steadily (final: 0.7487)
- **Training Accuracy:** Improved to 83.23%
- **Validation Loss:** Fluctuating but manageable (final: 0.9770)
- **Validation Accuracy:** Improved from 63.88% to 70.21%

EVIDENCE OF COMPLETION

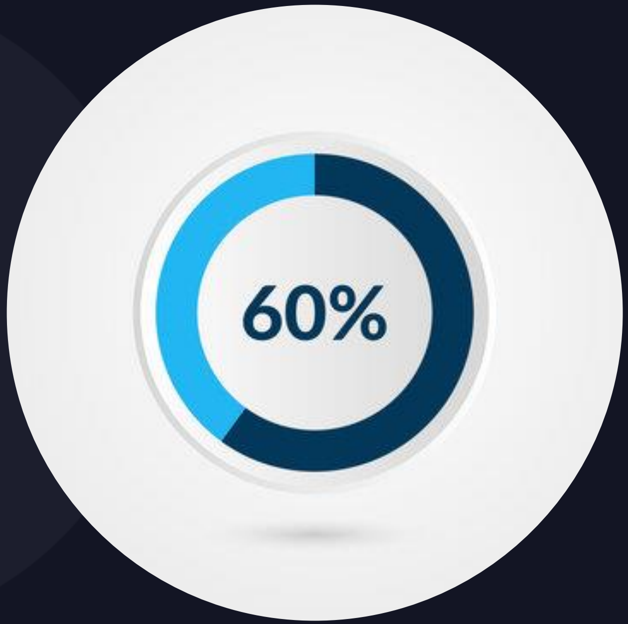
- **Loss Difference:** Training and validation losses converge, indicating a balanced learning process.

- **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.



Progress



Completed Tasks

- Weapon and harmful objects detection
- Facial emotion detection
- Harmful status detection by audio

Upcoming Tasks

- alert and notification system
- System Integration

Technologies to be Used

AI/ML Frameworks:
TensorFlow, PyTorch, Keras.

Development Tools:
IDEs: VSCode, PyCharm,
Jupyter Notebook

Simulation Frameworks:
CARLA, AirSim, or Unity

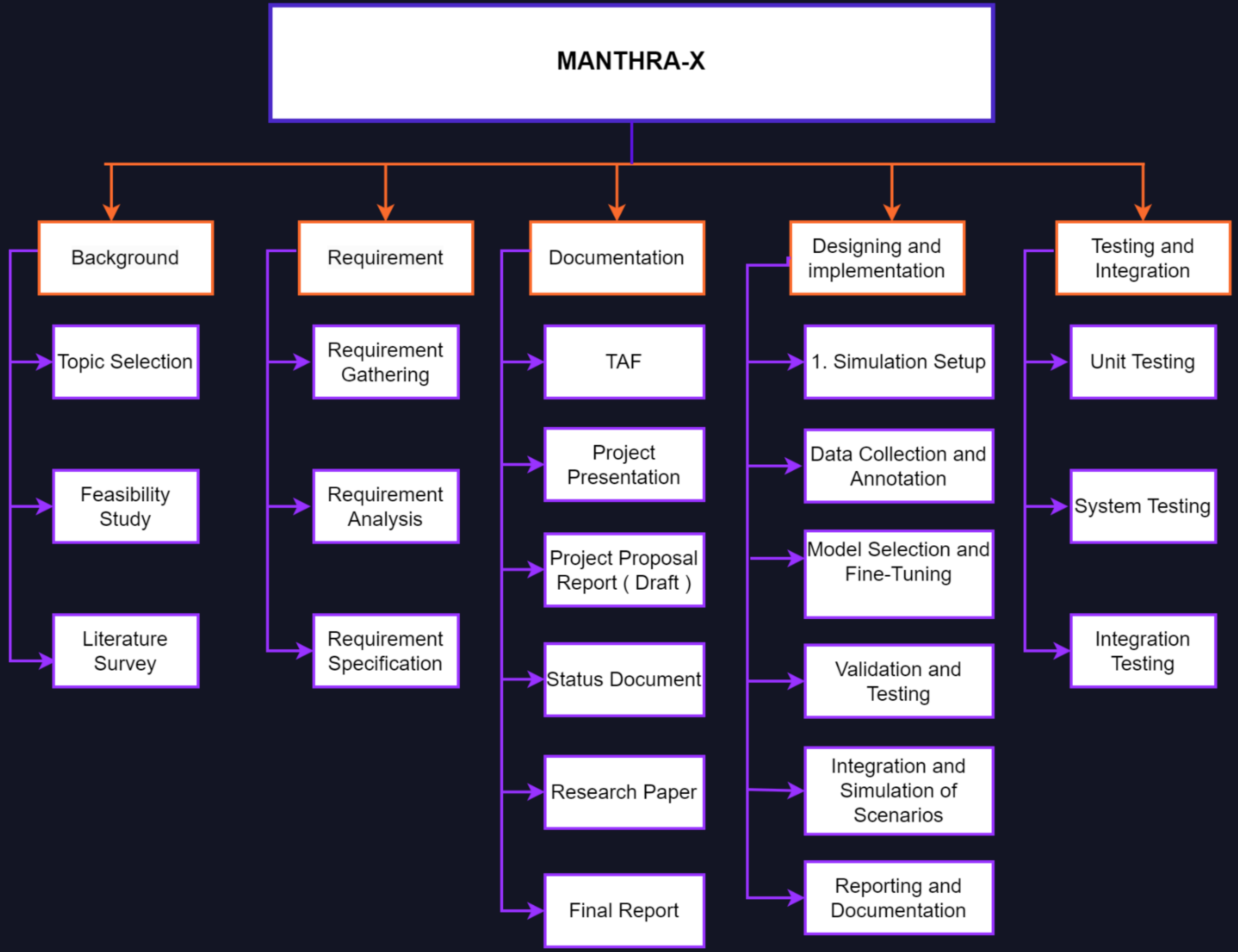
Security: Encryption.

REQUIREMENT SPECIFICATION

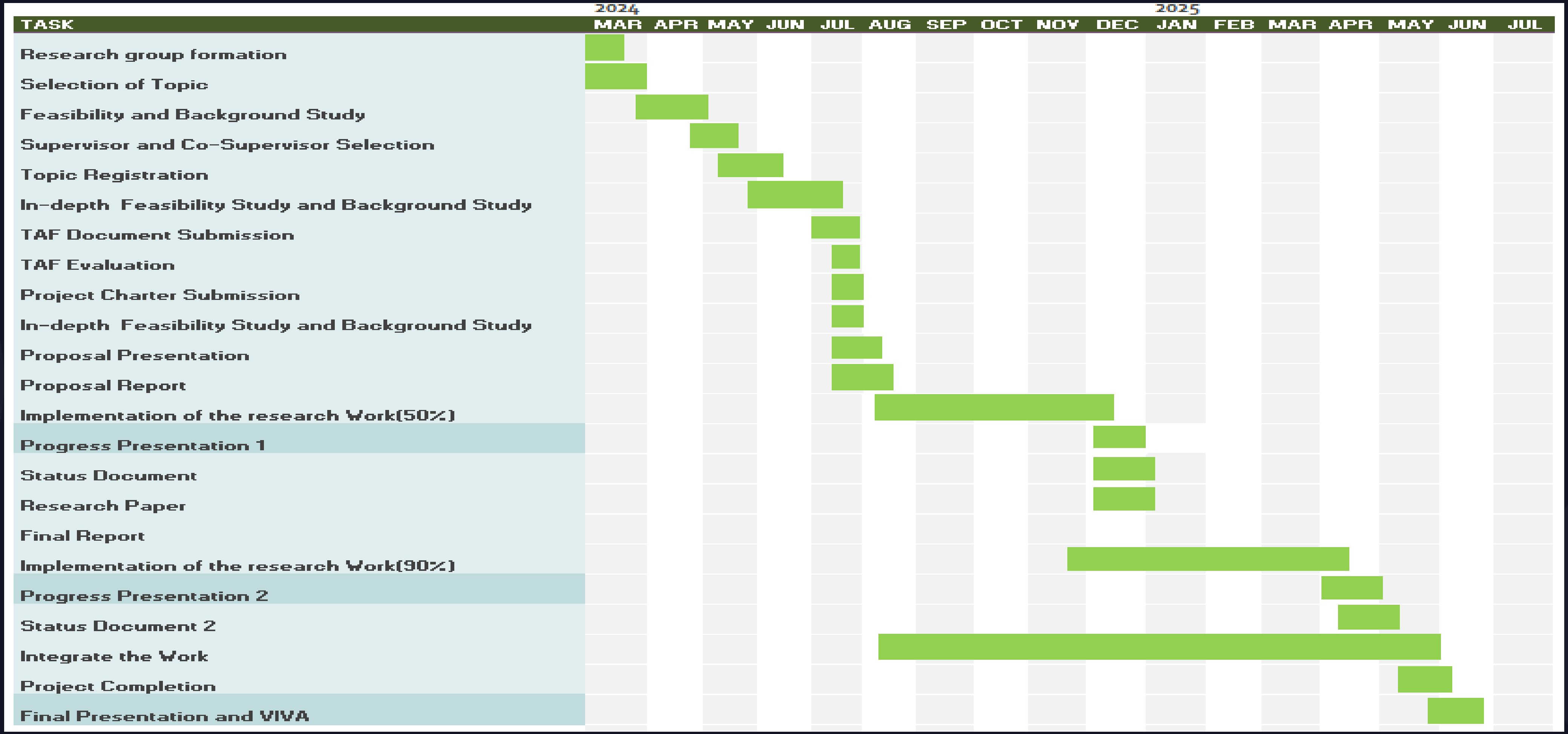
System Requirements

- **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-

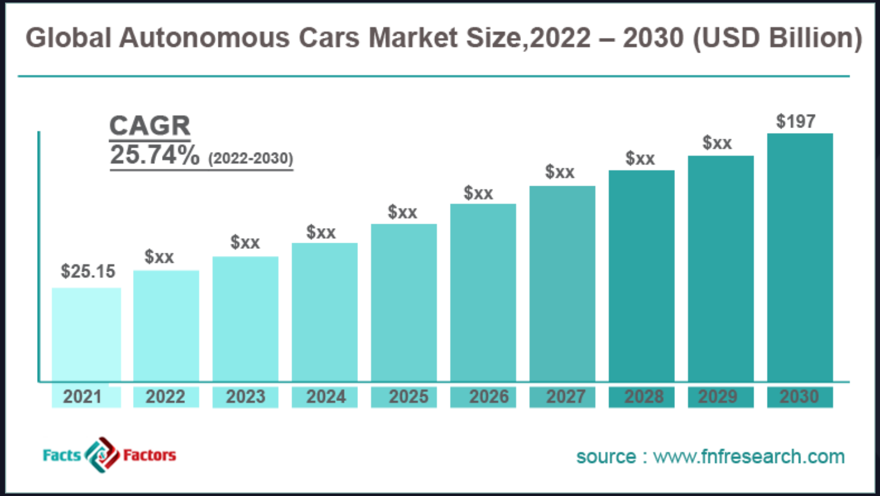
Create features that provide clear explanations of the system's decision-making processes, improving transparency and fostering trust among users regarding the ethical decisions made by the vehicle.

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