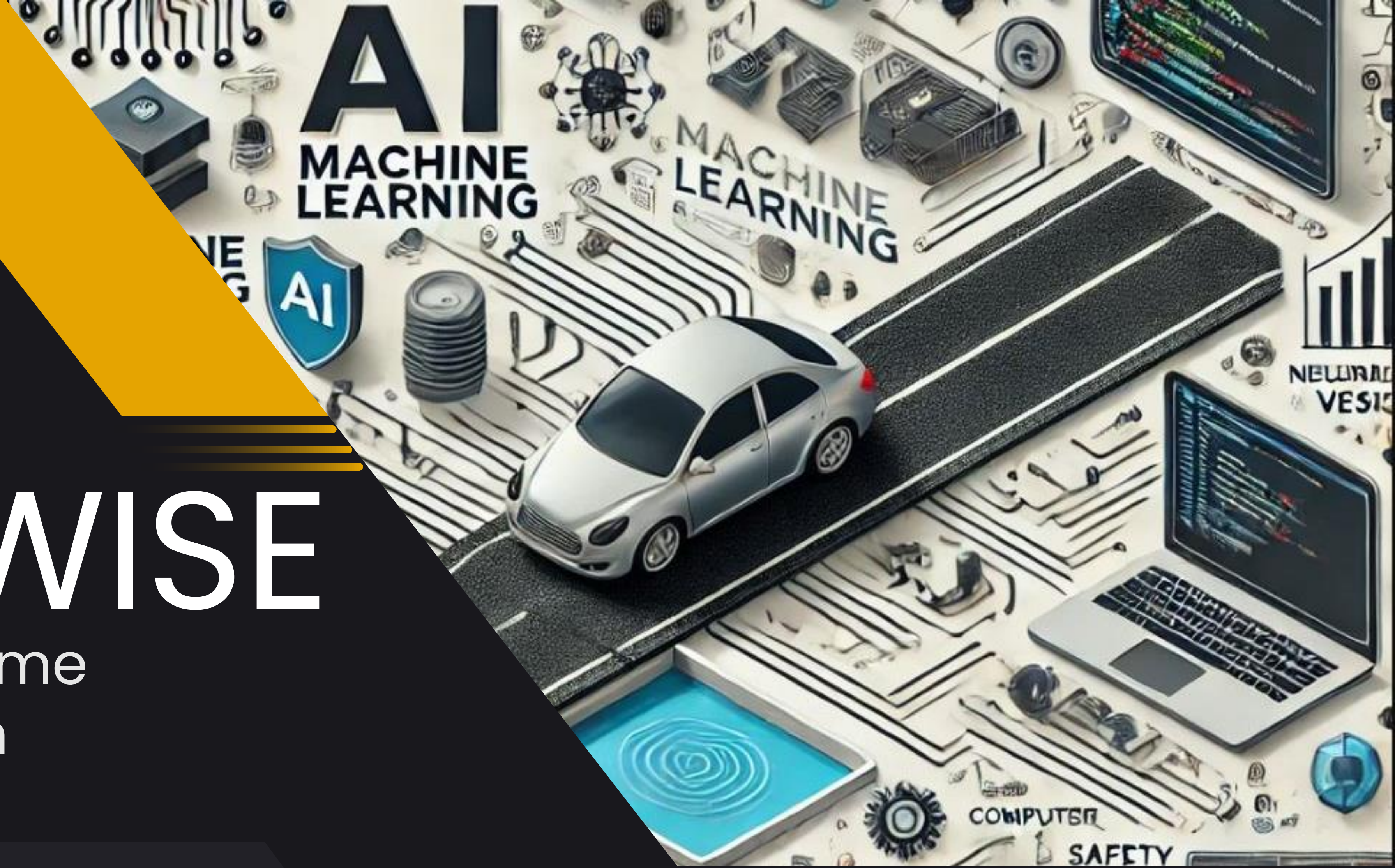
PSIT
Kanpur



ROADWISE

AI-Enabled Realtime Pothole Detection

A Group Project by:

- Shruti Singh
- Parth Awasthi
- Ashumendra Pratap Singh
- Shivansh Mishra
- Shivanshu Gupta

Team Id:
25_CS_IOT_4A_01

Under the mentorship of :
Mr. Hritik Mandal
(Assistant Professor)

TABLE OF CONTENTS

PSIT
Kanpur

INTRODUCTION

- In India, road safety is a critical issue due to complex traffic patterns, informal driving practices, and deteriorating road conditions, resulting in frequent accidents and risks for both drivers and pedestrians. To address this, we developed an AI-powered system that uses advanced machine learning (ML) algorithms, object detection, and adaptive learning to recognize and respond to real-time road hazards.

- By integrating Indian-specific driving behaviors, our system detects potholes, uneven surfaces, and unmarked road conditions. It prioritizes pedestrian safety through ML-based detection, crosswalk recognition, and alerts, optimizing lane usage and reducing congestion to enable safer, more efficient journeys.

Advanced AI & ML Integration

Enhanced Road Safety

Optimized Traffic Flow

PROBLEM STATEMENT

- India's roads face significant issues with maintenance, leading to frequent potholes and deteriorating road surfaces. These conditions create serious safety hazards, especially for high-density traffic areas
- Existing infrastructure and maintenance efforts are often insufficient to address these issues effectively, highlighting the need for a proactive, real-time solution.

PSIT Kanpur

PSIT
Kanpur

PROPOSED SOLUTION

Our solution is an ML-driven system designed for Indian road conditions, using advanced machine learning algorithms trained on diverse scenarios to detect hazards like potholes and uneven surfaces in real time. Adaptive learning allows the system to adjust its responses based on local driving patterns, ensuring adaptability in various environments.

The system includes ML-powered road condition recognition to detect surface anomalies and prioritizes pedestrian safety through features like pedestrian detection and crosswalk recognition. Leveraging cutting-edge ML, this solution enhances road safety and efficient navigation in India's challenging traffic landscape.

- Real-Time Hazard Detection
- **Multi-Modal Sensor Integration**
- **Traffic Flow Optimization**
- **User-Centric Feedback Loop**

PROJECT OBJECTIVES

Tailored for Indian road scenarios

Specifically designed to address the unique challenges present on Indian roads, including diverse traffic compositions, informal driving practices, and unpredictable road conditions.

Adaptive behavior learning

Utilizes reinforcement learning algorithms to adapt to local driving norms and dynamically adjust driving strategies in response to changing traffic conditions.

Pedestrian-centric design

Prioritizes pedestrian safety through advanced features like pedestrian detection, crosswalk recognition, and audible alerts, enhancing safety in busy urban areas.

Continuous improvement mechanism

Implements iterative learning and feedback mechanisms, utilizing simulation environments, real-world data to enhance performance and address emerging challenges.

- The objective of this project is to design and implement an integrated AI system to navigate the complex challenges of Indian roads, enhancing safety and efficiency. The system will employ advanced object detection algorithms, adaptive learning, and road condition recognition modules to address diverse traffic scenarios and road conditions.

- By achieving these objectives, we will enhance road safety and transportation efficiency across Indian road environments. This includes minimizing accidents and protecting pedestrians, cyclists, and drivers through pedestrian-centric features and precise recognition of road conditions. Optimizing traffic flow and reducing congestion will be possible by integrating real-time traffic management and adaptive driving strategies.

Feasibility Study

Technical Feasibility

This project is technically viable, using advanced AI algorithms and sensor technologies for real-time road data collection. Object detection systems can be adapted for Indian roads, while reinforcement learning allows dynamic strategy adjustments. Challenges include fine-tuning for diverse traffic and environmental conditions and ensuring compatibility with existing infrastructure. Long-term maintenance is critical.

Legal Feasibility

Compliance with legal standards such as data privacy laws, intellectual property rights, and local traffic regulations is essential to avoid conflicts. Partnerships with regulatory bodies may be required for approvals and to ensure adherence to safety standards. Potential legal issues could stem from ethical concerns around data usage or conflicts with current regulations, highlighting the importance of a thorough legal review to safeguard against liabilities.

Operational Feasibility

The system streamlines operations by automating hazard detection and providing real-time insights. User-friendly interfaces enhance usability, making the system accessible to various stakeholders. However, operational success depends on specialized skills in ML, data security, and system integration. Collaboration with local authorities may be necessary to upgrade infrastructure, and significant focus on user adoption and training is needed to ensure smooth implementation and acceptance by end-users.

Economic Feasibility

The project promises cost savings by enhancing traffic management efficiency and reducing accidents, which can lower long-term infrastructure costs. Initial investments in hardware, software, and AI development are substantial, and ongoing expenses will include maintenance, data storage, and system upgrades. A comprehensive cost-benefit analysis will determine the project’s financial viability, considering both direct savings and potential revenue from improved road utilization.

Methodology & Planning

Project Setup

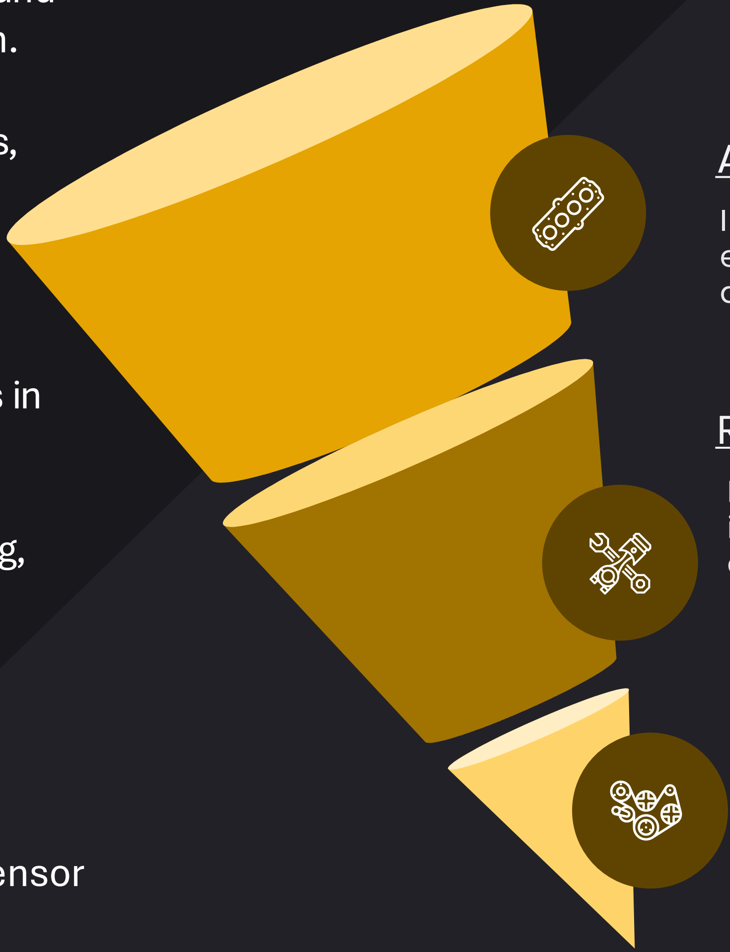
- Define Scope, Objectives & Requirements: Outline project goals and features like license plate recognition and dynamic slot allocation. Identify technical and user needs to guide development.
- Development Environment: Set up required tools and frameworks, including machine learning libraries, SDKs for payment, and appropriate platforms.

Data Acquisition and Preprocessing

- Data Collection: Gather a diverse dataset of license plate images in different lighting, angles, and vehicle types.
- Data Preprocessing: Cleanse and format the data through resizing, noise reduction, and conversion for optimal model performance.

System Development & Integration

- Front-End Development: Create an intuitive interface for user interactions, showing parking availability and payment options.
- Back-End Development: Build server logic for data processing, sensor communication, and payment integration.
- System Integration: Ensure seamless data flow between the ML model, front-end, and back-end components.



Advanced Model Training

Implement robust ML algorithms to enhance the accuracy of pothole detection and road quality analysis.

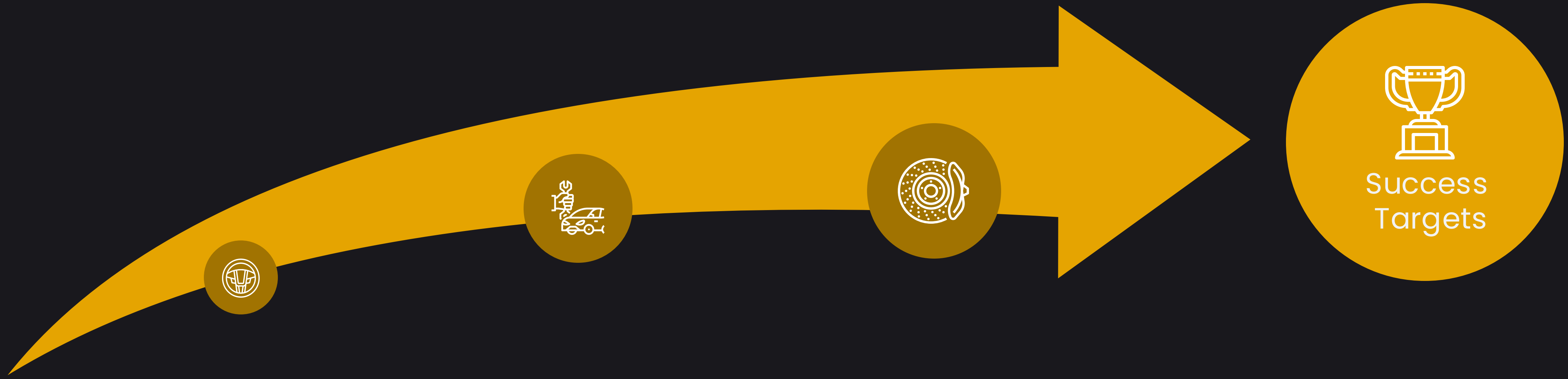
Real-Time Data Processing

Enable swift data handling for real-time insights, critical for on-the-go driving applications.

Scalable Infrastructure for ML Models

Build a scalable framework that supports continuous improvements and updates to driving-related ML models.

Planning



System Development and Integration

- **Front-End Development :** Create an intuitive interface for user interactions, showing parking availability and payment options.
- **Back-End Development :** Build server logic for data processing, sensor communication, and payment integration.
- **System Integration :** Ensure seamless data flow between the ML model, front-end, and back-end components.

Testing and Deployment

- **Unit Testing :** Test each component individually, covering model performance and API integration.
- **Integration Testing :** Evaluate the entire system in simulated environments for smooth operation.
- **Deployment :** Deploy on a suitable platform with necessary security configurations.

Monitoring and Maintenance

- **Monitoring** : Track system performance, including ML accuracy, resource usage, and error logs.
- **Maintenance :** Address issues proactively with model retraining, software updates, or security patches.
- **Response Time Analysis** : Regularly assess response times under varying loads to ensure smooth operation and scalability.

Technology Stack

Python

Python serves as the primary programming language for our project due to its simplicity, versatility, and strong support for machine learning and data processing. Its extensive libraries and frameworks make it ideal for developing AI-driven applications. Python's flexibility enables seamless integration with various data processing, computer vision, and machine learning tools, making it a popular choice in the AI and ML community.

NumPy

NumPy is a fundamental library in Python for numerical and scientific computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these data structures. In this project, NumPy facilitates efficient handling and manipulation of image data, which is crucial for processing large datasets. Its ability to perform complex calculations on arrays helps optimize data preprocessing and enhances model training efficiency.

OpenCV

OpenCV (Open Source Computer Vision Library) is a powerful library specifically designed for computer vision tasks. It offers tools for image processing, object detection, and machine learning, which are integral to our project's goals. OpenCV enables real-time image recognition, making it perfect for detecting road conditions and identifying objects, such as license plates or pedestrians, in video feeds. The library's versatility and support for various formats and camera integrations make it ideal for our system's vision-based functionalities.

Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow. It allows for rapid experimentation and model building with minimal code, making it a suitable tool for our machine learning requirements. In this project, Keras helps in building and training convolutional neural networks (CNNs) for tasks such as license plate recognition and pedestrian detection. Keras provides a user-friendly interface while also offering extensive customizability for complex ML models, contributing to high model accuracy and efficiency.

Data Collection And Preprocessing

Data Collection:

- **Lane Detection:**
  - Collected video and image data from public datasets and custom recordings.
  - Focused on diverse weather conditions, road types, and lighting to enhance model generalization.
- **Pothole Detection:**
  - Leveraged datasets from **Kaggle** and other open sources containing labeled images of road surfaces.
  - Ensured a mix of various pothole shapes, sizes, and environmental conditions (day/night, dry/wet).

  Examples of datasets:

  - **Pothole Dataset**
  - **Road Damage Detection**

Data Preprocessing:

- **Normalization:**
  Rescaled pixel values to a range of [0, 1] or [-1, 1] for faster convergence.
- **Resizing:**
  Standardized image dimensions (e.g., 224x224 or 512x512) to maintain uniformity across the dataset.
- **Grayscale Conversion:**
  Applied to certain models for computational efficiency in pothole detection.
- **Noise Removal:**
  Employed filters like Gaussian blur to smooth out irrelevant features.
- **Edge Detection:**
  Used techniques like **Canny** filters to enhance lane boundaries for better model interpretation.

Algorithm Selection and Model Training

Algorithm Selection:

Lane Detection: Convolutional Neural Networks (CNNs)

CNNs are highly effective for image-based tasks, especially where spatial features like edges and lines (e.g., road lanes) are critical.

Pothole Detection: YOLO (You Only Look Once)

- YOLO is an efficient object detection algorithm that performs real-time detection with high accuracy.
- Its single-shot detection framework allows simultaneous classification and localization of potholes in road images.

Model Training Process:

- **Lane Detection (CNN):**
  - **Loss Function:**
    - **Binary Cross-Entropy (BCE)** for binary lane mask segmentation.
    - **IoU Loss** for improving overlap accuracy.
  - **Optimizer:** Adam optimizer with an initial learning rate of 0.001.
  - **Evaluation Metric:** Dice Coefficient and Intersection over Union (IoU).

- **Pothole Detection (YOLO):**
  - **Loss Function:**
    - **Multi-part Loss:** Combines classification, objectness, and bounding box regression losses.
  - **Optimizer:** SGD with momentum for better gradient descent.
  - **Evaluation Metric:** mAP (mean Average Precision) and IoU.

Challenges & Limitations

Data Diversity and Quality: Gathering a comprehensive dataset to represent diverse road conditions, lighting, weather, and traffic patterns in India is challenging. Accurate and diverse data is essential to train models effectively and avoid bias.

Real-Time Processing Requirements: Achieving real-time data processing for quick hazard detection and adaptive responses is essential, particularly in high-traffic areas. Maintaining low latency is critical to ensure timely alerts and smooth operation.

Integration with Existing Infrastructure: Integrating the system with current road and traffic infrastructure can be costly and complex, requiring modifications or upgrades. Working efficiently within varied road networks and traffic management systems presents a significant challenge.

Environmental and Technical Constraints: Conditions such as rain, fog, and low light affect sensor performance, impacting the accuracy of computer vision and data capture systems. Developing a system robust enough to handle these conditions consistently is challenging.

Hardware and Maintenance Costs: Specialized hardware, such as sensors and cameras, increases initial and ongoing costs. Additionally, maintaining reliable performance and minimizing downtime across various environments is financially and logistically demanding.

User Acceptance and Trust: Achieving real-time data processing for quick hazard detection and adaptive responses is essential, particularly in high-traffic areas. Maintaining low latency is critical to ensure timely alerts and smooth operation.

Scalability and Adaptability: Scaling the system across diverse regions and adapting to evolving technology over time poses limitations. The system must be flexible and upgradeable to remain effective in changing environments and technological landscapes.

Data Storage and Management: Managing and storing vast amounts of road and traffic data requires significant storage capacity and efficient data management practices. Ensuring data security, especially when handling sensitive information, adds complexity and costs to the system.

Conclusion

Key Takeaways:

01. Problem Solved

Real-time pothole detection system enhancing road safety and reducing vehicle damage.

02. Technological Impact:

- YOLO-based model ensures fast, accurate, and scalable detection.

Future Scope:

01. Deployment

Integration with smart city infrastructure and autonomous vehicles.

02. Enhancements:

Improved accuracy with larger datasets and advanced models (e.g., YOLOv5).

03. Additional Features:

Severity classification for prioritizing repairs.

Final Thoughts:

01. **Social Impact:**

Enhances safety, prevents accidents, and lowers maintenance costs.

02. Scalability

Suitable for diverse geographies and road conditions

THANKS