NAAN MUDHALVAN

PHASE-2

ARTIFICIAL INTELLIGENCE & DATA SCIENCE
(2nd YEAR)

AUTONOMOUS VEHICLES AND ROBOTICS

AI-BASED ROUTE MEMORY AND SELF LEARNING

TEAM LEADER

Sri Ram R

TEAM MEMBERS

Nithin Deepak R

Santhosh S

Shanyu Starness P

Udhaya Kumar A

Phase 2: Innovation & Problem Solving

Title: AI-based Route Memory and Self-Learning Navigation System

Innovation in Problem Solving

The goal of this phase is to apply innovative AI techniques to tackle the real-world challenges of autonomous navigation, especially in unpredictable or unmapped environments. Inspired by the way humans and animals learn paths over time, this project uses reinforcement learning, memory-based algorithms, and environmental simulations to allow an agent to independently learn optimal paths. This system can later be deployed in assistive robotics, autonomous wheelchairs, or disaster rescue bots.

Core Problems to Solve

1. **Learning Without Maps**: The system must navigate in unfamiliar spaces without prior maps, relying on memory and learning from experience.
2. **Dynamic Obstacle Handling**: Environments may change dynamically. The system must adapt and update routes on the fly.
3. **Efficient Path Planning**: The algorithm must balance exploration and exploitation to find the shortest or safest path.
4. **Generalization Across Environments**: The model should be capable of learning in one environment and adapting the knowledge to others with similar structures.

Innovative Solutions Proposed

1. **Reinforcement Learning for Route Memory**

- **Solution Overview**: A reinforcement learning agent (e.g., using Q-learning) is trained in a simulated grid environment to navigate from a starting point to a goal.
- **Innovation**: Instead of relying on pre-programmed paths, the system learns from rewards and punishments, eventually developing memory-based optimal routing behavior.
- **Technical Aspects**:
    - State-action-reward-based learning using Q-tables or deep Q-networks.
    - Adaptation through episodic training and value updates.
    - Environmental simulation with variable layouts and obstacles.

2. **Experience-Based Adaptability**

- **Solution Overview**: The agent adapts to environmental changes by continuously updating its learned policies based on new inputs and experiences.

- **Innovation**: The system demonstrates lifelong learning capabilities, allowing it to "relearn" or modify routes without full retraining.
- **Technical Aspects**:
    - Dynamic reward shaping.
    - Real-time policy updating during runtime.
    - Temporal memory integration to weigh past successful routes.

3. **Interactive User Interface and Visualization**

- **Solution Overview**: A visual simulation showing the agent learning and navigating, offering users a clear view of decision-making processes.
- **Innovation**: Interactive controls and overlays explain why the agent chooses certain paths, increasing trust and transparency.
- **Technical Aspects**:
    - Visualization of agent state, path, and Q-table.
    - Manual override and real-time feedback injection.
    - Analytics on learning efficiency and episode success rate.

4. **Future Hardware Deployment**

- **Solution Overview**: Preparing the AI logic for integration with hardware robots for real-world tests.
- **Innovation**: The trained model will be exportable to real-world platforms using sensors like ultrasonic, IR, and wheel encoders for navigation.
- **Technical Aspects**:
    - Porting AI logic to microcontrollers (Arduino, Raspberry Pi).
    - Real-time sensor fusion.
    - Lightweight model optimization for edge devices.

Implementation Strategy

1. **Algorithm Development**
   The core reinforcement learning model is developed and tested in a 2D simulated environment. Different reward structures and obstacle types are used to enhance the model's robustness.
2. **Training Environment Setup**
   Grid-based environments with obstacles and varying goal locations are simulated using Python libraries such as Pygame or Gym.
3. **Real-World Mapping Strategy**
   Sensor data emulation is incorporated to simulate how the same AI could work with real-

world inputs. The plan includes porting the agent to a physical robot using minimal hardware components.

Challenges and Solutions

- **Slow Convergence**: In early stages, the agent may take long to find the optimal path. Solution: Tuned learning rate, added penalties for repeated mistakes, and reduced action randomness over time.
- **Environmental Complexity**: Realistic environments often include unpredictable elements. Solution: Introduced random obstacle generators to improve generalization.
- **Transfer to Physical Hardware**: Moving from simulation to real-world poses challenges like noise and delays. Solution: Added simulation delays and planned future testing with actual sensors.

Expected Outcomes

1. **Self-Learning Navigation**: The agent will be able to autonomously navigate unfamiliar areas using learned experiences.
2. **Scalable Intelligence**: The AI model can be trained once and deployed in multiple scenarios with minor adjustments.
3. **Assistive Robotics Potential**: This system can be applied to assistive robots for the elderly or disabled to move around safely and efficiently.
4. **Foundation for Autonomous Mobility**: Lays the groundwork for AI-based mobile robots that require no human intervention or predefined maps.

Next Steps

1. **Hardware Integration**: Begin porting the AI logic to a microcontroller-powered robot using IR sensors and basic motor control.
2. **User Feedback Loop**: Incorporate real-time performance feedback (path success, collision rate) to adapt navigation logic dynamically.
3. **Field Deployment**: Test the system in controlled indoor environments like hallways and rooms, then scale to semi-structured outdoor areas.