**Project Initialization and Planning Phase**

|  |  |
| --- | --- |
| Date | 20 June, 2025 |
| Team ID | SWTID1749906821 |
| Project Title | Neural Networks Ahoy: Cutting-edge Ship Classification for Maritime Mastery |
| Maximum Marks | 3 Marks |

**Project Proposal (Proposed Solution) template**

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

|  |  |
| --- | --- |
| **Project Overview** | |
| Objective | Develop an AI-powered ship classification system using neural networks to enhance maritime surveillance, security, and operational efficiency by accurately identifying vessel types in real-time. |
| Scope | - Train deep learning models to classify ships from radar/AIS data and imagery. - Integrate with maritime traffic systems for real-time monitoring. - Deploy scalable APIs for port authorities and coast guards. |
| **Problem Statement** | |
| Description | Manual ship identification is slow and error-prone, leading to security risks (e.g., unauthorized vessels), inefficient port management, and delayed emergency responses. Existing systems lack accuracy in dynamic weather conditions. |
| Impact | Solving this will: - Improve maritime security by 40% through real-time threat detection. - Reduce port congestion delays by 25% with automated traffic sorting. - Enhance search-and-rescue efficiency via rapid vessel identification. |
| **Proposed Solution** | |
| Approach | - Use convolutional neural networks (CNNs) for image-based classification (satellite/radar). - Apply recurrent neural networks (RNNs) to analyze AIS movement patterns. - Deploy models on edge devices (drones, buoys) for low-latency inference. |
| Key Features | - Multi-modal classification: Combines visual (RGB/IR) and trajectory data. - Adaptive learning: Retrains models with new ship designs/behavior. - API integration: RESTful APIs for coastal surveillance systems. |

**Resource Requirements**

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Description** | **Specification/Allocation** |
| **Hardware** | | |
| Computing Resources | GPU for model training | T4 GPU |
| Memory | RAM specifications | 8-16GB DDR4 |
| Storage | Disk space for data, models, and logs | 1 TB SSD |
| **Software** | | |
| Frameworks | Python frameworks | Flask |
| Libraries | Additional Libraries | Keras, pandas, numpy, matplotlib, tensorflow |
| Development Environment | Coding & collaboration | Jupyter Notebook, Git |
| **Data** | | |
| Data | Source, size, format | Kaggle dataset, 37.44 kB**,** csv |