

**Project Initialization and Planning Phase**

|  |  |
| --- | --- |
| Date | 15 March 2024 |
| Team ID | 739675 |
| Project Title | Cleantech: Transforming Waste Management With Transfer Learning |
| Maximum Marks | 3 Marks |

**Project Proposal (Proposed Solution)**

|  |  |
| --- | --- |
| **Project Overview** |  |
| Objective | To design and implement a transfer learning-based automated waste classification system that enhances the efficiency and accuracy of waste sorting processes, thereby contributing to sustainable waste management practices and supporting the development of smart, environmentally responsible communities |
| Scope | The scope of this cleantech project involves developing an intelligent waste classification system using transfer learning techniques. By fine-tuning pre-trained convolutional neural networks such as ResNet50 or DenseNet121, the system aims to accurately categorize waste into types like plastic, metal, paper, and organic materials, even with limited training data |
| **Problem Statement** |  |
| Description | The problem lies in the inefficiency and inaccuracy of traditional waste sorting methods, which heavily rely on manual labor and are prone to errors. |
| Impact | Solving this problem for Applying transfer learning in waste management enhances classification accuracy, achieving up to 97.8% F1 scores, thereby improving recycling efficiency and reducing environmental pollution. |
| **Proposed Solution** |  |
| Approach | The proposed solution involves fine-tuning pre-trained convolutional neural networks, such as EfficientNet-B0 and ResNet50, to accurately classify waste types using transfer learning techniques. |
| Key Features | * Enhanced Classification Accuracy: Utilizing pre-trained models like ResNet50 and VGG16 * Real-Time Processing * Scalability and Adaptability * User Engagement and Education |



**Resource Requirements**

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Description** | **Specification/Allocation** |
| **Hardware** |  |  |
| Computing Resources | CPU/GPU specifications, number of cores | e.g., 2 x NVIDIA V100 GPUs |
| Memory | RAM specifications | e.g., 8 GB |
| Storage | Disk space for data, models, and logs | e.g., 1 TB SSD |
| **Software** |  |  |
| Frameworks | Python frameworks | e.g., Flask, TensorFlow |
| Libraries | Additional libraries | e.g., NumPy, OS |
| Development Environment | IDE, version control | e.g., Google Colab, VS code |
| **Data** |  |  |
| Data | Source, size, format | e.g., Kaggle dataset, 10,000 images |