CleanTech Report

1. INTRODUCTION

1.1 Project Overview

This project focuses on classifying municipal waste images into three categories: Biodegradable, Recyclable, and Trash. It uses a deep learning approach through the VGG16 model with transfer learning and is deployed via a Flask web application.

1.2 Purpose

The purpose of the project is to enable intelligent waste segregation at the source, thereby improving recycling rates and reducing environmental impact.

2. IDEATION PHASE

2.1 Problem Statement

Manual segregation of waste is time-consuming, prone to human error, and inefficient. Automating the classification using image-based deep learning models can solve this issue effectively.

2.2 Empathy Map Canvas

- **Users:** Waste collection agencies, municipalities, recycling plant workers
- **Needs:** A fast, simple, and accurate classification system
- **Pains:** Time lost in manual sorting, misclassification, and contamination of recyclables

  Gains: Improved efficiency and better segregation practices

2.3 Brainstorming

- Using image classification with CNNs
- Transfer learning to reduce training time
- Web application for real-time use
- Expandability to other waste categories

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User uploads waste image via UI
2. Flask backend sends it to the model

3. Model classifies the image
4. Prediction is shown back on the web interface

3.2 Solution Requirement

- Labeled waste image dataset
- Pre-trained VGG16 model
- Python (TensorFlow, Flask, Keras)
- Frontend in HTML

3.3 Data Flow Diagram

User → Web UI → Flask Backend → VGG16 Model → Prediction → UI

3.4 Technology Stack

- Python 3.x
- TensorFlow/Keras
- Flask
- HTML/CSS (Jinja2 for templates)

4. PROJECT DESIGN

4.1 Problem-Solution Fit

By applying deep learning with a lightweight deployment using Flask, users can classify waste images easily and quickly.

4.2 Proposed Solution

Train a CNN using transfer learning (VGG16), test for accuracy, and deploy using Flask for realtime use.

4.3 Solution Architecture

- **Frontend:** index.html and result.html
- **Backend:** app.py using Flask
- **Model:** vgg16.h5 (or .keras) trained using dataset

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week	Task
1	Dataset collection and preprocessing
2	Model design and training

3 Flask integration and UI creation
4 Testing and debugging
5 Final documentation and deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- **Accuracy:** Achieved ~90% accuracy on validation data
- **Tests:** Manual testing with 10 random images from each class
- **Response Time:** Less than 2 seconds per prediction

7. RESULTS

7.1 Output Screenshots

- Home page upload interface
- Prediction result displayed below uploaded image
- Backend terminal showing predicted output

(Insert screenshots here when finalizing the report)

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Simple UI and workflow
- Accurate classification using VGG16
- Lightweight backend using Flask **Disadvantages:**
- No database or login system
- Limited to 3 waste classes
- Requires consistent image lighting for better accuracy

9. CONCLUSION

The project demonstrates a working waste classification system using transfer learning and Flask deployment. It can serve as a base model for expanding to more classes and full-scale implementations.

10. FUTURE SCOPE

- ➓ Add more classes (e-waste, metals, glass)
- ➓ Integrate with smart bins or IoT devices
- ➓ Develop Android/iOS frontend
- ➓ Use cloud-based inference with databases

11. APPENDIX

Source Code: Available in `w_flask/` directory
Dataset: Dataset
GitHub Link: https://github.com/charansai2004/Waste-management-system.git

Demo Video: https://drive.google.com/file/d/1J2R11MnqxUWeHtB1OWZ0PeGD-cx99RrY/view?usp=drive_link