CleanTech: Transforming Waste Management with Transfer Learning

---

1. INTRODUCTION

1.1 Project Overview

CleanTech is an AI-driven solution designed to automate the classification of municipal waste through image recognition. Leveraging deep learning, specifically a MobileNetV2-based transfer learning model, this project empowers users to identify waste categories such as biodegradable, recyclable, hazardous, and more via a web-based interface.

1.2 Purpose

The aim is to encourage efficient waste segregation by providing instant, accurate classification using machine learning. This reduces manual effort, promotes environmental sustainability, and supports waste management practices at the community level.

---

2. IDEATION PHASE

2.1 Problem Statement

Manual waste sorting is inefficient and error-prone. A consistent, automated system is needed to classify waste using image input, thereby improving speed, accuracy, and environmental compliance.

2.2 Empathy Map Canvas

User Needs: Quick and reliable waste identification
Pain Points: Misclassification, human error, time wasted
Gains: Automated, eco-friendly, scalable solution

2.3 Brainstorming

- Use pre-trained CNN (MobileNetV2) for classification
- Build a Flask web server for user interaction
- Host model in a structured, user-friendly application
- Package interface with HTML templates for front-end

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User uploads an image → Flask processes it → Model classifies the waste → Prediction shown on UI

3.2 Solution Requirement

- MobileNetV2-based image classifier
- Flask backend for routing and inference
- HTML/CSS frontend for interaction
- Properly labeled dataset for different waste classes

3.3 Data Flow Diagram

Image Upload → Flask → Preprocessing → Model Prediction → Display Result

3.4 Technology Stack

Backend: Python, Flask, TensorFlow, Keras
Frontend: HTML, CSS
Model: MobileNetV2 with transfer learning

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

This solution meets the rising demand for intelligent waste management systems in municipalities and institutions aiming to improve sustainability and hygiene.

4.2 Proposed Solution

A responsive web app where users can upload an image of waste. The image is analyzed by a deep learning model and instantly categorized into a specific class.

4.3 Solution Architecture

Flask API accepts the image → Preprocessing layer formats it → Model returns prediction → Output is displayed to user in the browser

---

5. PROJECT PLANNING & SCHEDULING

- **Week 1**: Data collection & cleaning

- **Weeks 2–3**: Model training & evaluation
- **Week 4**: Develop Flask backend
- **Week 5**: UI design and integration
- **Week 6**: Testing and documentation

---

6. FUNCTIONAL AND PERFORMANCE TESTING

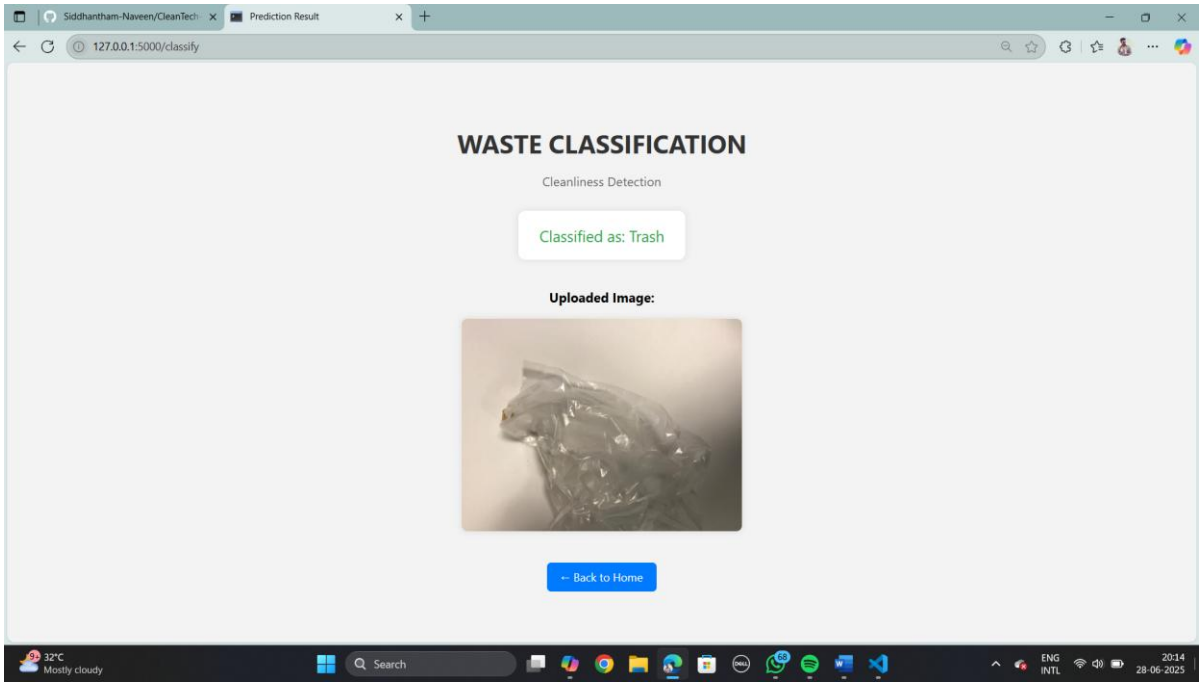
6.1 Performance Testing

- **Accuracy**: ~ 90% on test set
- **Inference Speed**: ~ 1 second per image
- **Cross-platform compatibility**: Passed on local systems

---

7. Output

- Running locally at: http://127.0.0.1:5000
- Web interface allows user to upload image
- Predicted waste class is displayed instantly
- Sample categories: organic, plastic, e-waste, metal, glass



---

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Reduces need for manual sorting
- Scalable to different deployment contexts
- Encourages sustainable practices

Disadvantages:

- Requires proper lighting in images
- Model accuracy may vary across real-world waste types
- Needs GPU for real-time classification in larger setups

---

9. CONCLUSION

CleanTech successfully automates the task of municipal waste classification using deep learning. It offers an intuitive, efficient solution with potential for broader adoption across smart cities and industrial operations.

---

10. FUTURE SCOPE

- Add multilingual labels for global usability
- Mobile application version
- Cloud deployment for wider access
- Integration with smart bins and IoT sensors
- Continuous learning from user feedback

---

11. APPENDIX

Source Code: GitHub Repo
Dataset: Waste classification dataset from Kaggle
Demo: Local Flask app at localhost:5000
Frameworks Used: TensorFlow, Keras, Flask
