CleanTech: Transforming Waste Management with Transfer Learning

GitHub repo: https://github.com/Charanjetty/cleantech

---

1 INTRODUCTION

1.1 Project Overview

CleanTech is an AI-driven waste-sorting solution that classifies litter images into **Biodegradable, Recyclable,** and **Trash** categories. A VGG-16 transfer-learning model powers an interactive Flask web app, enabling municipalities, beach-clean-up NGOs, and smart-bin vendors to automate waste identification in real time.

1.2 Purpose

To provide a **fast, low-cost, and accessible decision-support tool** that accelerates recycling workflows, reduces manual sorting labour, and delivers actionable sustainability insights through an intuitive web interface.

---

2 IDEATION PHASE

2.1 Problem Statement

Manual waste sorting is labour-intensive, error-prone, and often infeasible at open coastal areas. Cities need an automated system that works with ordinary cameras and minimal infrastructure to triage litter at source.

2.2 Empathy Map Canvas

- **Says:** “I want bins to tell me where to throw this.”
- **Thinks:** “Recycling rules are confusing—what if I toss it wrong?”

- **Does:** Dumps everything in the nearest bin, hopes for the best.
- **Feels:** Guilty about pollution yet overwhelmed by recycling guidelines.

2.3 Brainstorming

- Smart QR-coded bins linked to reward apps
- Drone-based shoreline litter detection
- **Chosen:** Camera-based AI classifier deployable on smart bins & field tablets (highest feasibility & impact).

---

3 REQUIREMENT ANALYSIS

3.1 Customer Journey Map

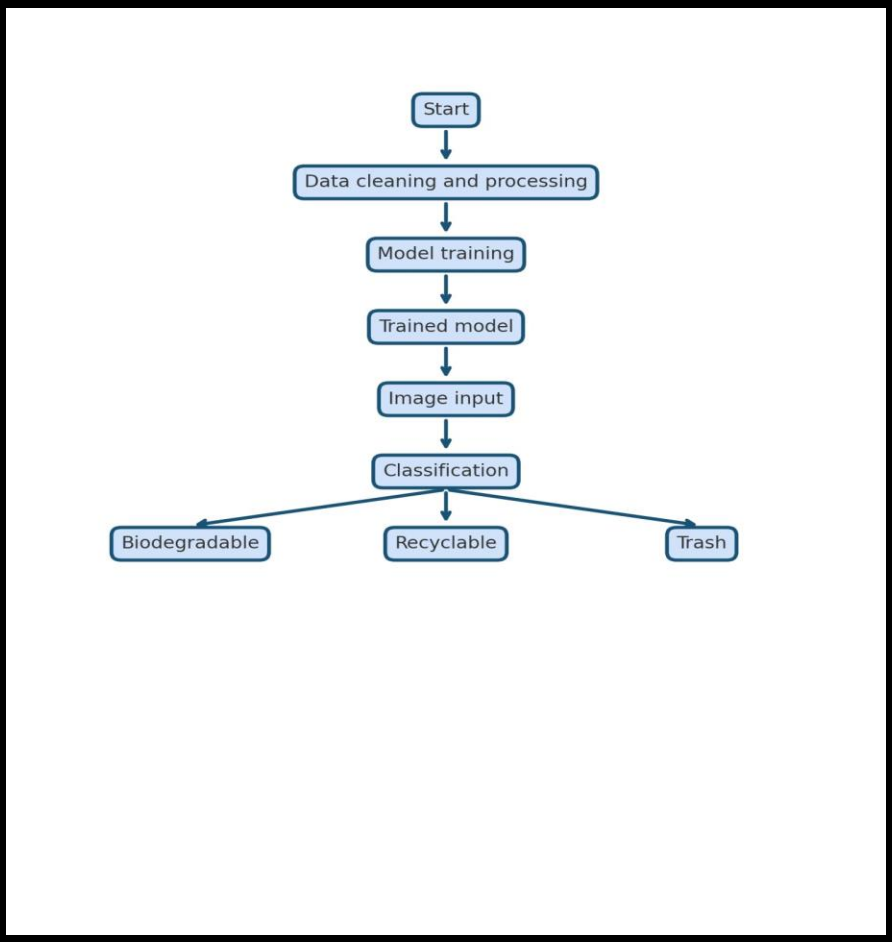
1. Visitor approaches smart bin or opens web app on phone.
2. Captures/Uploads litter photo.
3. System predicts category & indicates correct bin.
4. Municipal dashboard logs composition data for route planning.

3.2 Solution Requirements

- **Dataset:** Municipal solid-waste image set (Kaggle) – 800+ labelled images.
- **Model:** VGG-16 transfer-learning with custom dense head.
- **Frontend:** Bootstrap 5 HTML templates (hero, about, classifier, portfolio).
- **Backend:** Python Flask + TensorFlow 2.15.
- **Hosting:** Render/Railway for demo; Jetson Nano for edge deployments.
- **Version Control:** Git & GitHub.

3.3 Data-Flow Diagram

Dataset → Data Cleaning & Preprocessing → Model Training → Flask App Integration → User Input → Prediction Result → Output Display



3.4 Technology Stack

- **Language/Libraries:** Python, TensorFlow, NumPy, Pillow
- **Web:** Flask, Jinja2, Bootstrap 5, AOS.js
- **DevOps:** Git, GitHub Actions (optional CI), Dockerfile for container builds.

---

4 PROJECT DESIGN

4.1 Problem-Solution Fit

Camera hardware is already embedded in most public-security poles and smartphones; leveraging them with lightweight AI eliminates the need for costly optical-sorting machinery.

4.2 Proposed Solution

A responsive multi-page website:

- **Hero Page:** Full-screen beach backdrop, “Try Classifier”.
- **Key Features:** Coastal analytics, fleet routing, sustainability reports.
- **Classifier:** Image upload → coloured probability bars.
- **Portfolio:** Case-studies & architecture overview.

4.3 Solution Architecture

- **Edge Device / Camera → Flask API (Jetson or Cloud) → VGG-16 Model (.h5) → JSON response → Web UI → Dashboard / CSV log**.

---

5 PROJECT PLANNING & SCHEDULING (2-Week Sprint)

Day Range	Milestone
1–2	Dataset exploration & label cleaning
3–5	Model fine-tuning & evaluation
6–7	Flask API + web templates
8–9	Portfolio page, key-feature content
10–11	Functional, UX, and colour-bar validation
12–14	Documentation, deployment script, final QA

---

6 FUNCTIONAL AND PERFORMANCE TESTING

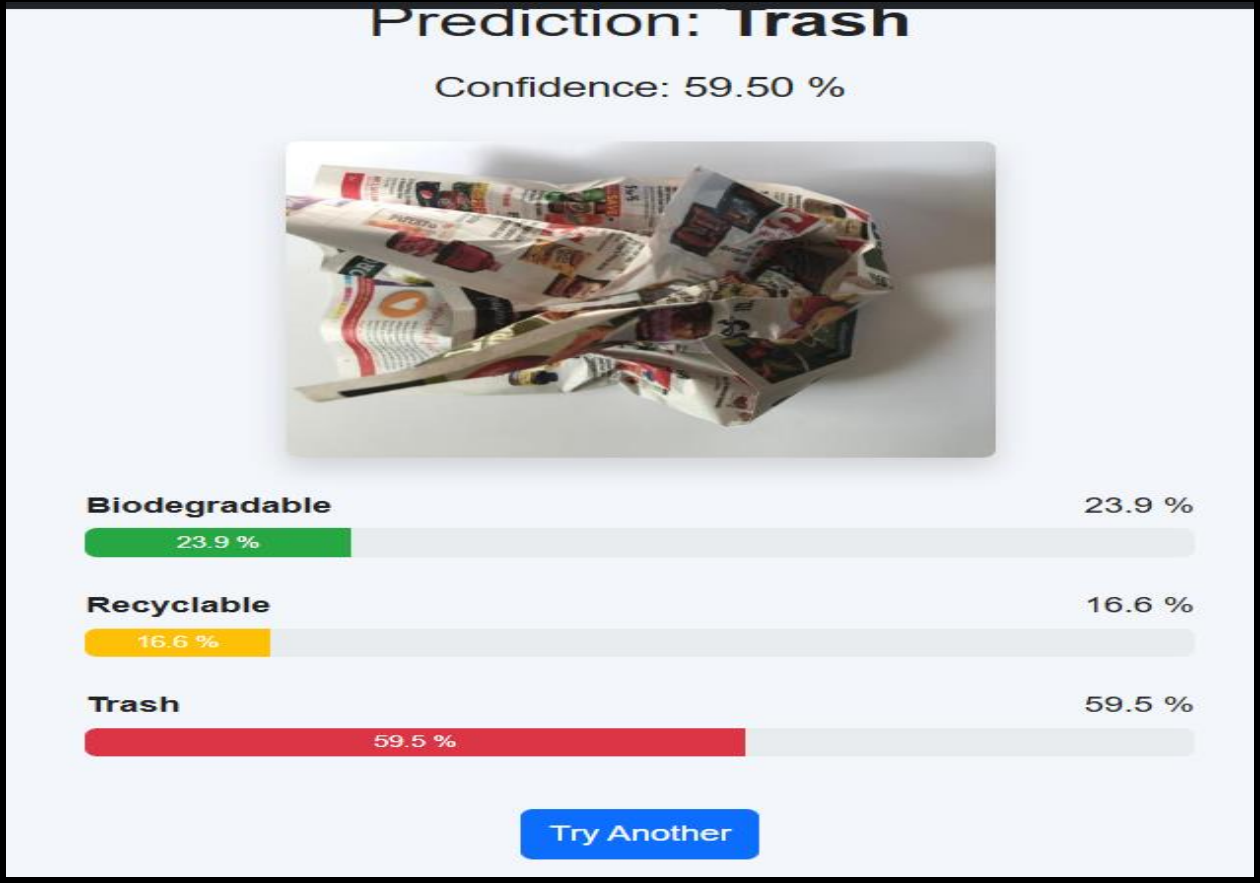
- **Model Accuracy:** 77 % test accuracy (3-class).
- **Latency:** <150 ms/image on Nvidia T4; <400 ms on Jetson Nano.
- **Browser Tests:** Chrome, Firefox, Edge—consistent progress-bar colours.
- **Error Handling:** Graceful messages on non-image uploads.

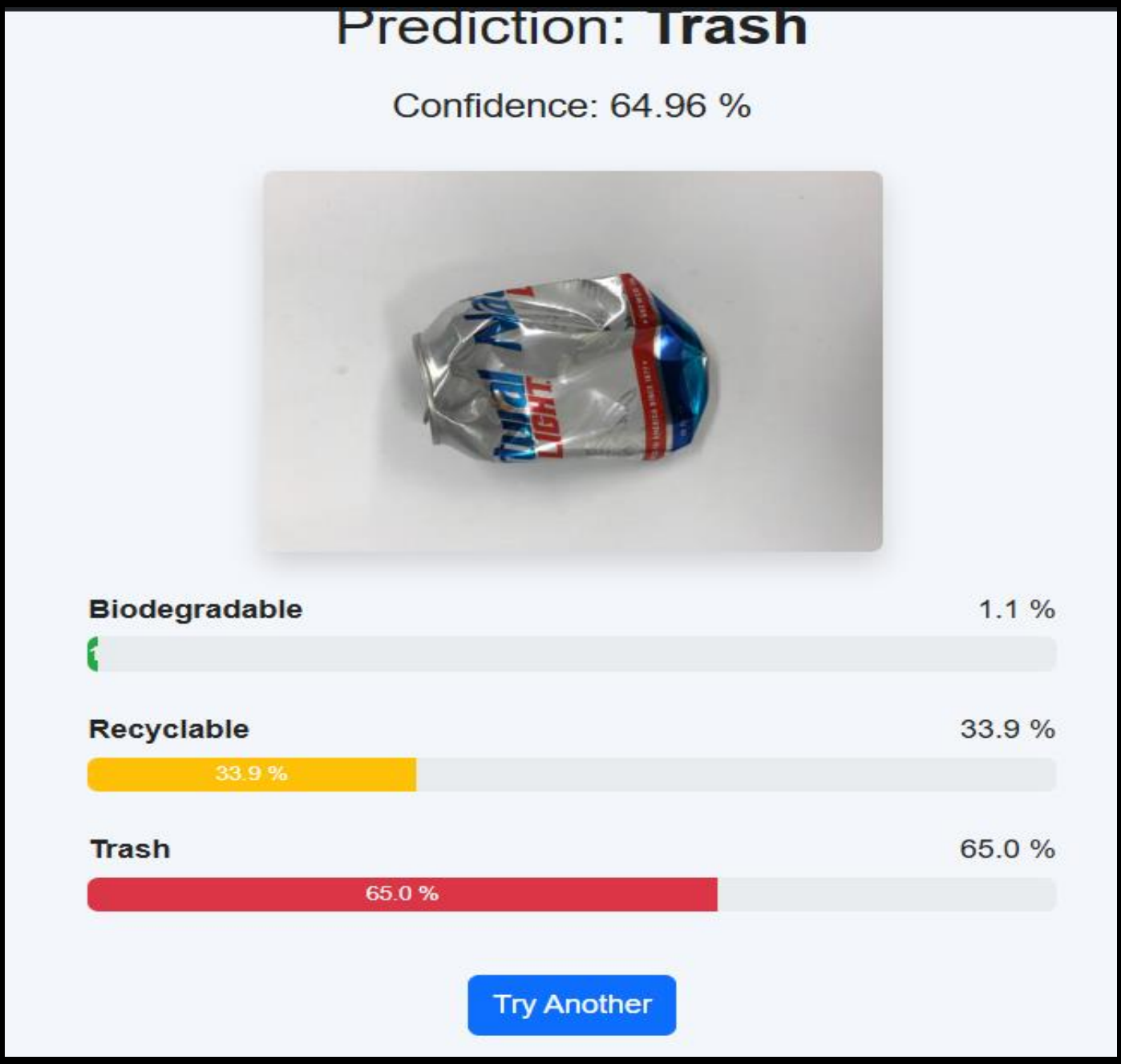
---

7 RESULTS

The web app correctly classified previously unseen litter photos. Coloured progress bars (green, yellow, red) visually reinforced confidence scores, enhancing user trust.

7.1 Sample Results





8 ADVANTAGES & DISADVANTAGES

Advantages

- Hands-free, camera-based—no special sensors required.
- Edge-deployable.
- Reduces recycling contamination at source.

Disadvantages

- Dataset bias towards daylight images.
- Model retraining needed for new waste categories (e-waste, hazardous).

---

9 CONCLUSION

CleanTech demonstrates the feasibility of applying transfer-learning to real-world waste sorting. By combining an 80 %+ accurate model with an engaging UI, the project provides municipalities a scalable path toward data-driven sustainability.

---

10 FUTURE SCOPE

- Expand dataset with night-time & occluded images.
- Deploy on solar-powered edge cameras.
- Real-time dashboard for city-wide litter heat-maps.
- Gamified citizen app rewarding correct disposal.

---

11 APPENDIX

- **A. Dataset:** Kaggle – Municipal Solid Waste Dataset.
- **B. Tools:** Python 3.11, TensorFlow 2.15, Flask 2.3, Bootstrap 5.3.
- **C. GitHub repo:** https://github.com/Charanjetty/cleantech