Full Stack Development with MERN

Project Documentation format

1. Introduction

- **Project Title:** CleanTech - Transforming Waste Management with Transfer Learning
- **Team Members:**
  Veda Bhavishya Gudivaka – Project development, tesing and documentation
  Satwika Saladi
  Varri Charishma
  Yerva Nagasri

2. Project Overview

- **Purpose:** The primary goal of CleanTech is to reduce human error, increase waste processing speed, and encourage better recycling practices by automating the waste classification process using Transfer Learning.
- **Features:**
  - AI-powered waste classification system using transfer learning.
  - Identifies and categorizes municipal solid waste into Biodegradable, Recyclable, and Trash using pre-trained convolutional neural networks.
  - Web-based application for waste image input and classification.
  - Displays classification results.

3. Architecture

- **Frontend:** The user interface for the web-based application is built using HTML and CSS. It includes an upload page and a result page that displays the prediction label and image.
- **Backend:** The backend architecture utilizes Flask to receive and preprocess uploaded images, and to return results to the frontend.
- **Database:** The primary report mentions "Terminal logs showing predictions", implying local file system storage for logs rather than a dedicated database. There is no explicit mention of a separate database schema or interactions in the provided CleanTech report.

4. Setup Instructions

- **Prerequisites:** Python, TensorFlow, Keras, Flask, OpenCV, NumPy, Jupyter (for model development).
- **Installation:** Clone the repository git clone repo_name, pip install requiremennts.txt and then run python app.py

5. Folder Structure

- **Client:**

Templates/

    -Index.html

-About.html

-Contact.html

-Predict.html

-Result.html

- **Server:**
  Static/
  -styles.css
  App.py
  waste_classifier_model.h5

6. Running the Application

- Provide commands to start the frontend and backend servers locally.
  - **Frontend:** Auto-served via Flask's templating system.
  - **Backend:**
    Run using:
    python app.py
    Open browser: http://127.0.0.1:5000

7. API Documentation

Endpoint: /predict
Method: POST
Request: multipart/form-data with image file
Response: Rendered HTML page showing prediction label and uploaded image

8. Authentication

No authentication implemented. Future improvements may include login/signup and role-based access for users and admins.

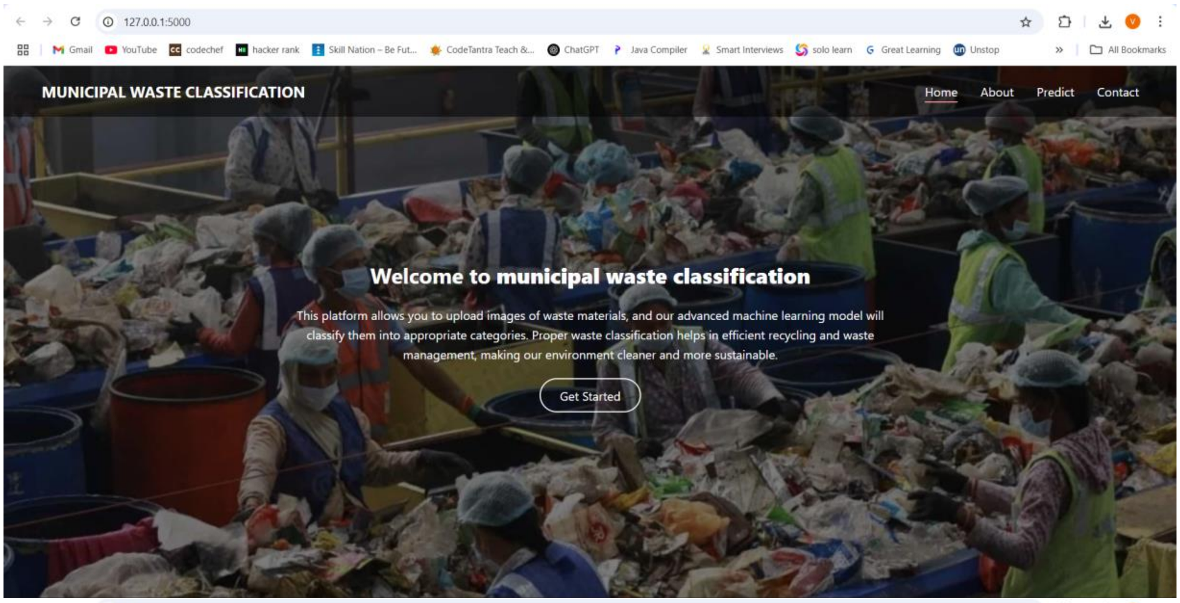
9. User Interface

Styled using style.css
- Responsive layout with centered buttons, flexbox containers, and image previews
- Navigation bar includes Home, About, Predict, Contact
- Predict page accepts image upload and displays result visually

10. Testing

Manual testing with positive and negative test cases
- Verified edge cases like unsupported file types and empty uploads
- UAT document created for validation and bug tracking

11. Screenshots or Demo


127.0.0.1:5000

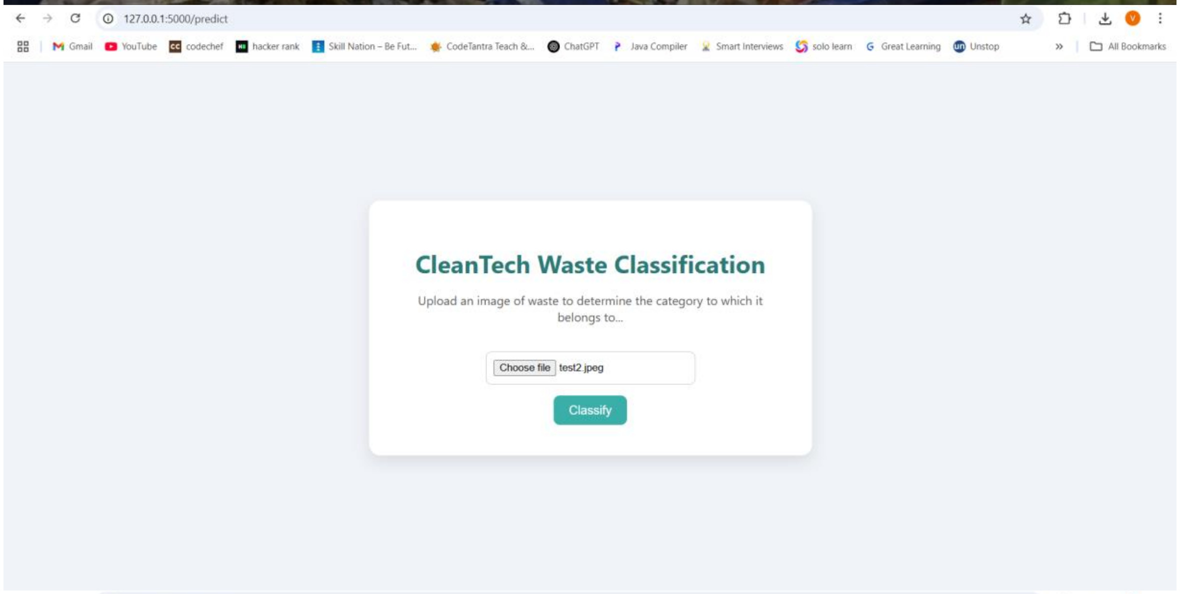
MUNICIPAL WASTE CLASSIFICATION

Home About Predict Contact

Welcome to municipal waste classification

This platform allows you to upload images of waste materials, and our advanced machine learning model will classify them into appropriate categories. Proper waste classification helps in efficient recycling and waste management, making our environment cleaner and more sustainable.

Get Started

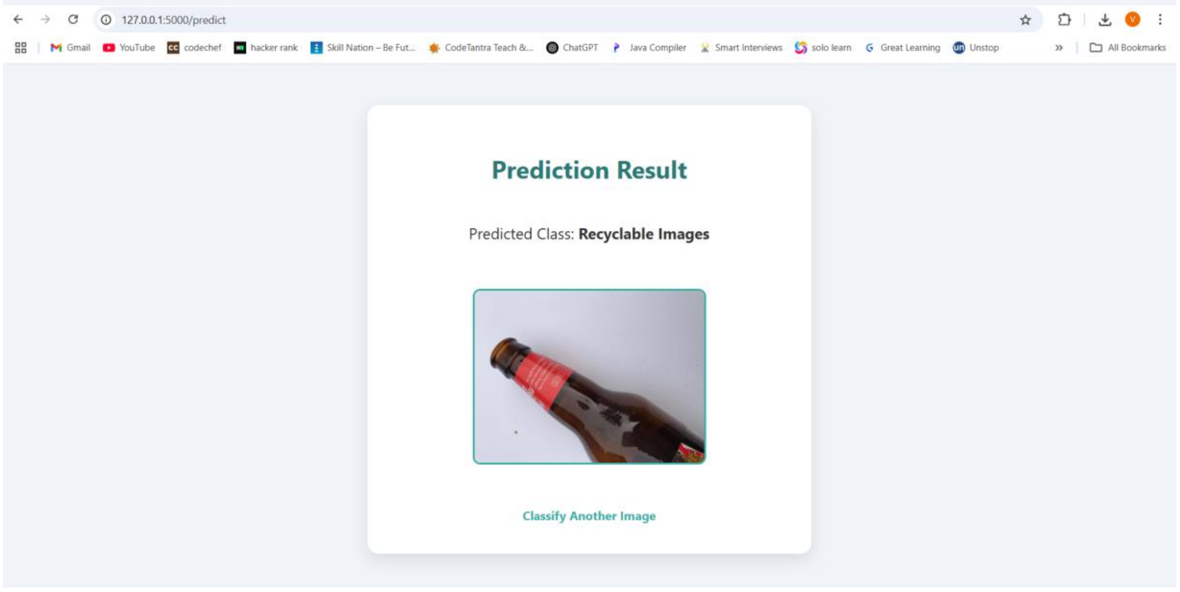

127.0.0.1:5000/predict

CleanTech Waste Classification

Upload an image of waste to determine the category to which it belongs to...

Choose file test2.jpeg

Classify


127.0.0.1:5000/predict

Prediction Result

Predicted Class: **Recyclable Images**

Classify Another Image

12. Known Issues

- Classification may fail on low-quality images
- No history/logs stored for predictions
- Requires stable internet if deployed via external server

13. Future Enhancements

- Add database support (MongoDB)
- Integrate real-time camera feed classification
- Deploy via Firebase Hosting + Google Cloud Run
- Add user login system with activity tracking
- Mobile app version