PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

The project **"CleanTech - Transforming Waste Management through Transfer Learning"** aims to develop a smart waste classification system that can automatically identify and categorize waste as **recyclable** or **non-recyclable** using deep learning techniques. The system leverages image data of waste items and uses **Transfer Learning** to train a high-performance classification model with limited data. The project utilizes pre-trained CNN architectures such as **ResNet50** and **EfficientNetB0**, which are fine-tuned for this specific task. The classification model is integrated into a user-friendly web interface where users can directly upload waste images and receive real-time classification results. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess the model's effectiveness. The project aims to support sustainable waste management practices by enabling smart segregation at the source.

1.2 Purpose

- To design an intelligent and automated system for waste classification using image recognition.
- To minimize human intervention and reduce errors in manual waste sorting processes.
- To apply **Transfer Learning** to train a robust classification model with limited dataset availability.
- To compare and analyze the performance of various CNN architectures like **ResNet50** and **EfficientNetB0**.
- To assist in proper waste segregation by distinguishing **recyclable** and **non-recyclable** items accurately.
- To promote environmentally friendly practices through smart waste management technologies.
- To build a scalable and reusable system that can be adapted for other classification tasks in the field of environmental tech.

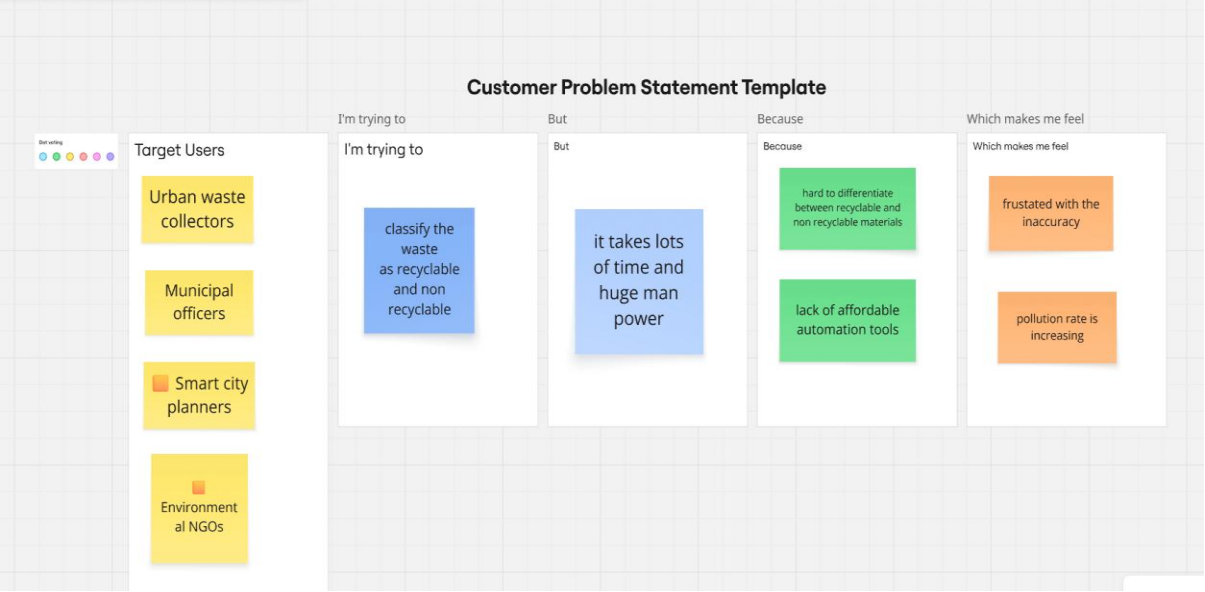
2. IDEATION PHASE

2.1 Problem Statement

Customer Problem Statement:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



Reference: https://miro.com/templates/customer-problem-statement/

2.2 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Reference: https://www.mural.co/templates/empathy-map-canvas

2.3 Brainstorming

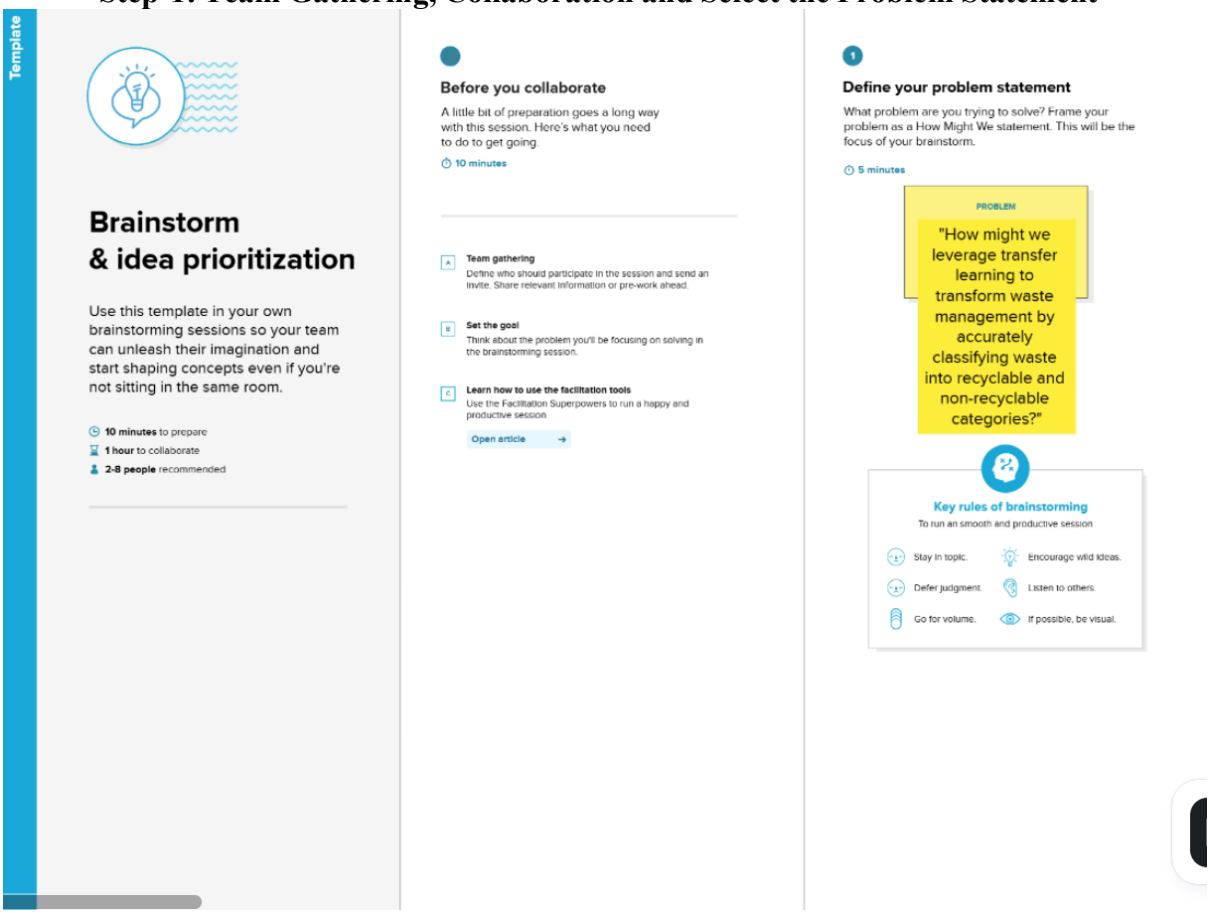
Brainstorm & Idea Prioritization:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

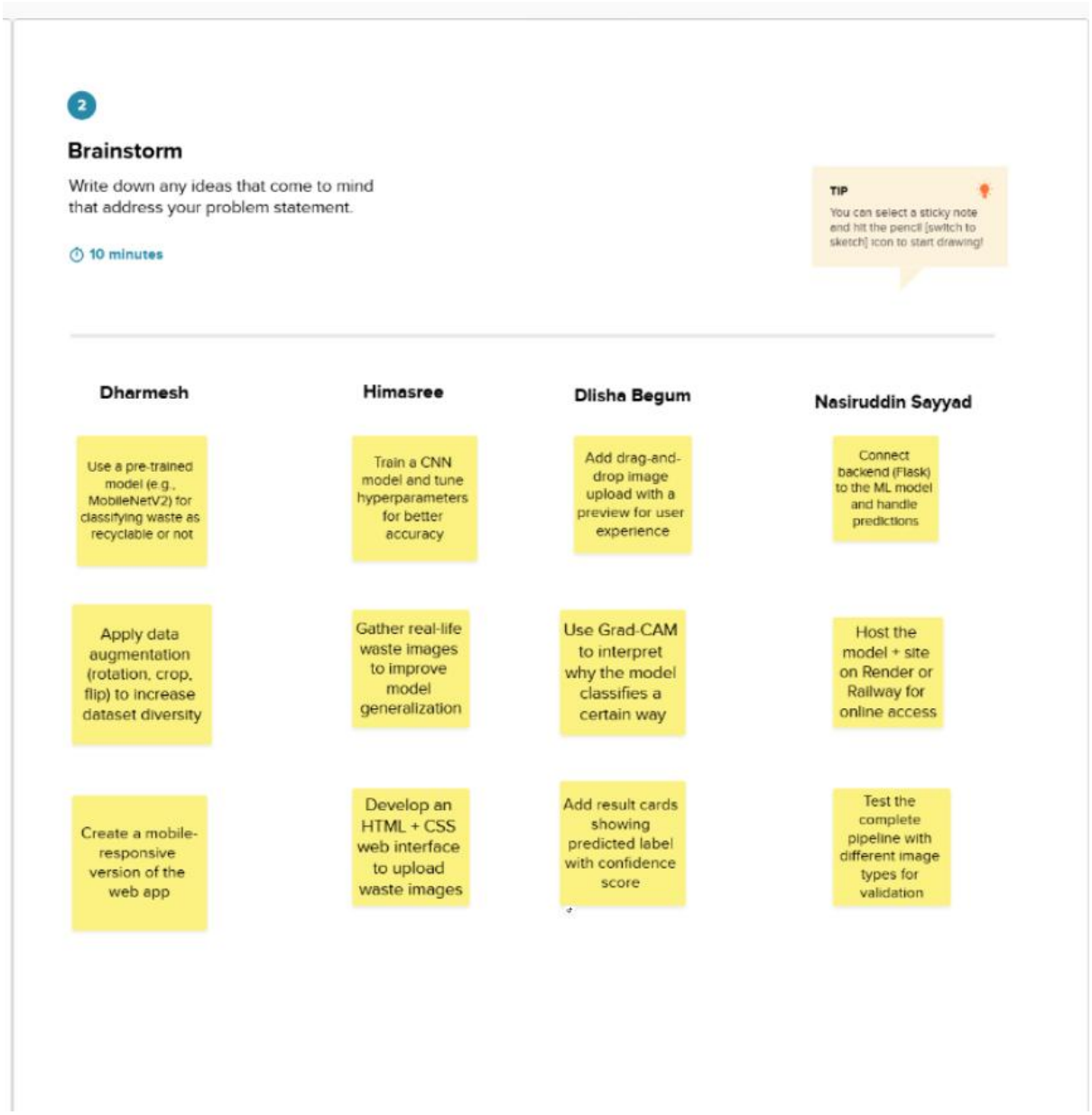
Use this

Reference: https://www.mural.co/templates/brainstorm-and-idea-prioritization

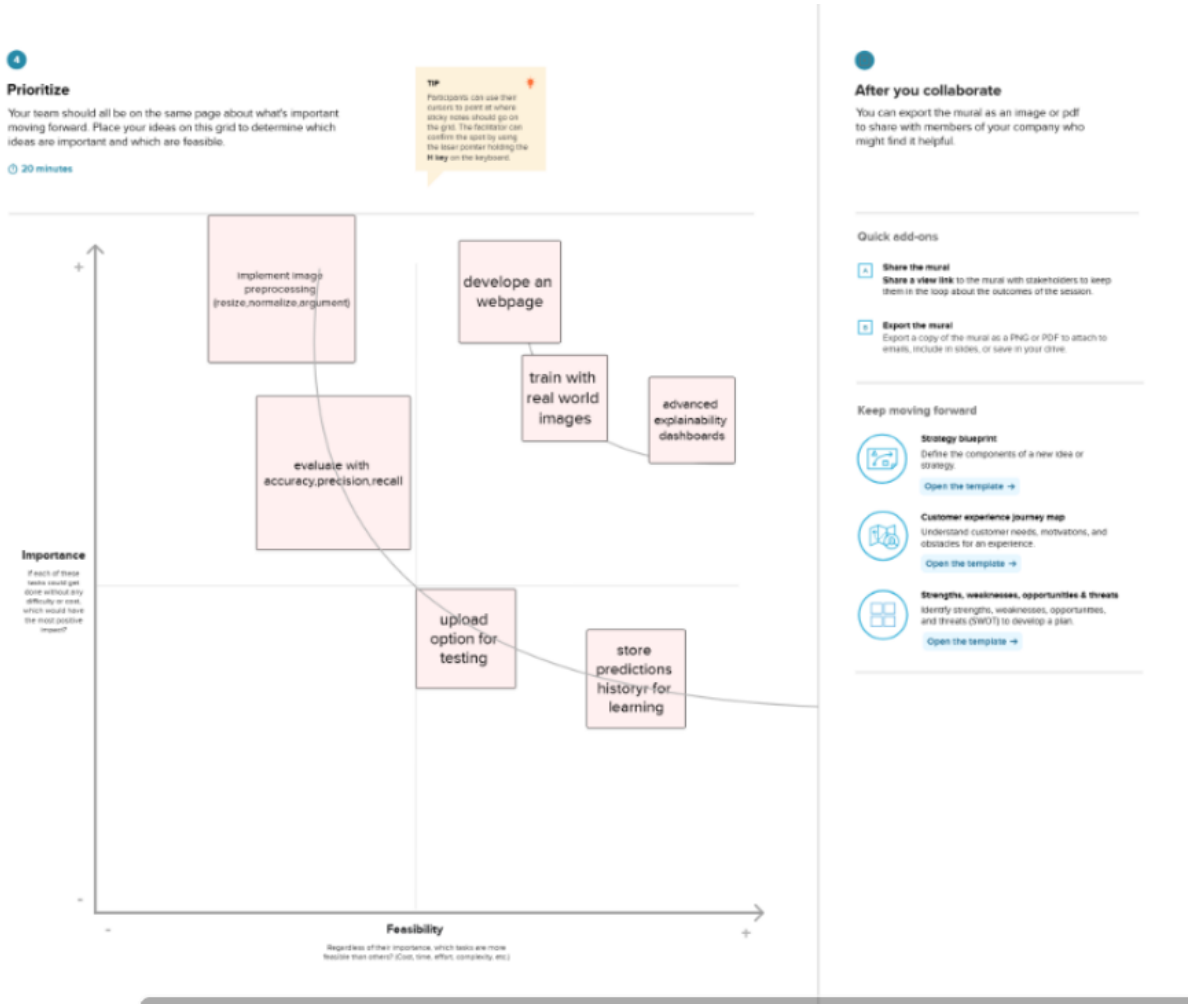
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3. REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
		Registration through Gmail
		Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email
FR-3	Waste Classification	Upload Waste Image
		Classify as Recyclable/Non-recyclable
		Show Confidence Score
FR-4	Dashboard & Reports	Visual Waste Statistics

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
		Downloadable Reports
		Trend Analysis Over Time
FR-5	User Feedback	Submit Rating for Classification
		Provide Comments
FR-6	Admin Module	View All Classifications
		Manage Users
		Manage Feedback

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Interface should be simple, intuitive, and accessible for all age groups
NFR-2	Security	User data should be encrypted; secure login and authentication required
NFR-3	Reliability	System should deliver consistent classification results under various loads
NFR-4	Performance	Image classification should return results within 3–5 seconds
NFR-5	Availability	System should be available 99.9% of the time (24/7 access)
NFR-6	Scalability	Should support scaling for large user base and data volume in future phases

4. PROJECT DESIGN

4.1 Problem Solution Fit

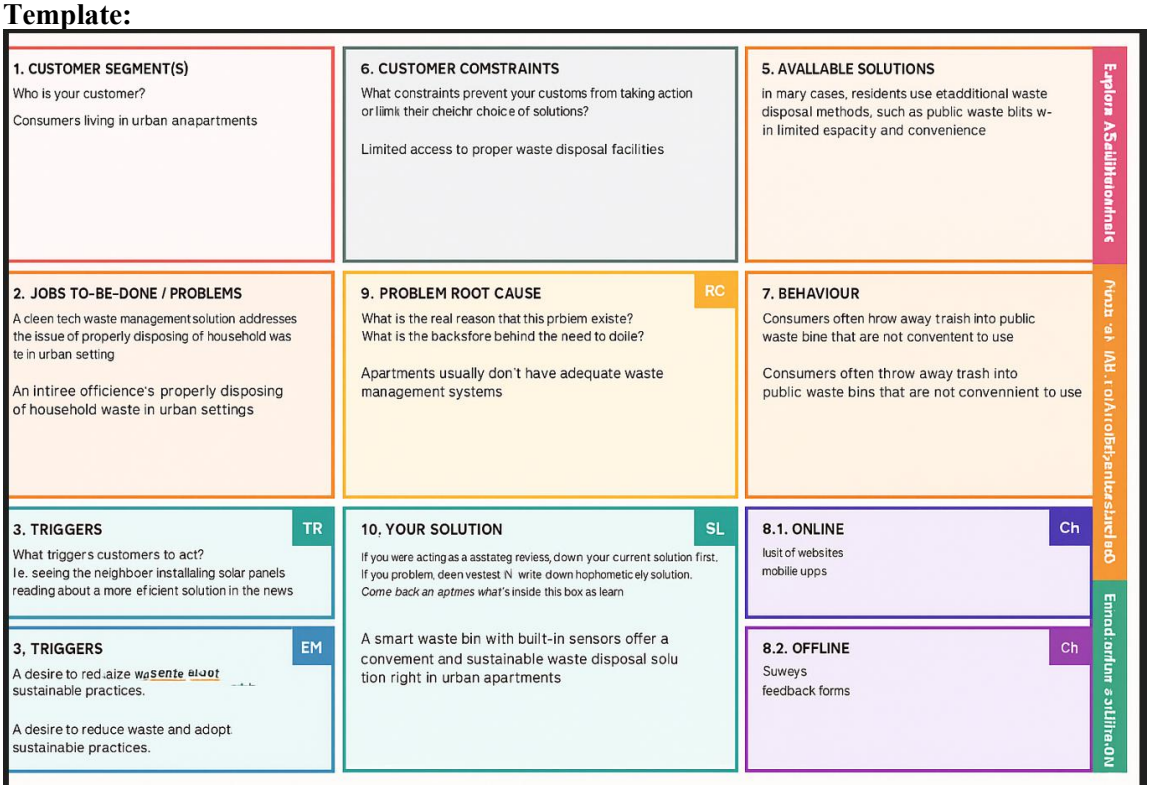
Problem – Solution Fit Overview:

Problem – Solution Fit Template:
The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

- ❑ Solve complex problems in a way that fits the state of your customers.
- ❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- ❑ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ❑ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- ❑ **Understand the existing situation in order to improve it for your target group.**

Template:



References:

1. https://www.ideahackers.network/problem-solution-fit-canvas/
2.https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe

4.2 Proposed Solution

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Inefficient waste segregation leading to environmental damage, lack of awareness, and improper recycling due to human error and manual sorting.

2.	Idea / Solution description	| A smart AI-based waste classification system using transfer learning that identifies and separates recyclable and non-recyclable waste automatically using image recognition.
3.	Novelty / Uniqueness	Utilizes pre-trained models to reduce data requirements and training time; deployable via mobile or edge devices for real-time classification in households, schools, and public areas.
4.	Social Impact / Customer Satisfaction	Promotes sustainability, reduces landfill waste, and increases recycling efficiency. Enhances awareness of proper waste management practices among the public.
5.	Business Model (Revenue Model)	B2B licensing to municipalities, NGOs, waste collection agencies; subscription-based SaaS for institutions; pay-per-use model for public recycling kiosks.
6.	Scalability of the Solution	Easily scalable across cities and regions. The AI model can be fine-tuned with local data. Can integrate with existing waste bins or IoT systems.

4.3 Solution Architecture

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.

- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:

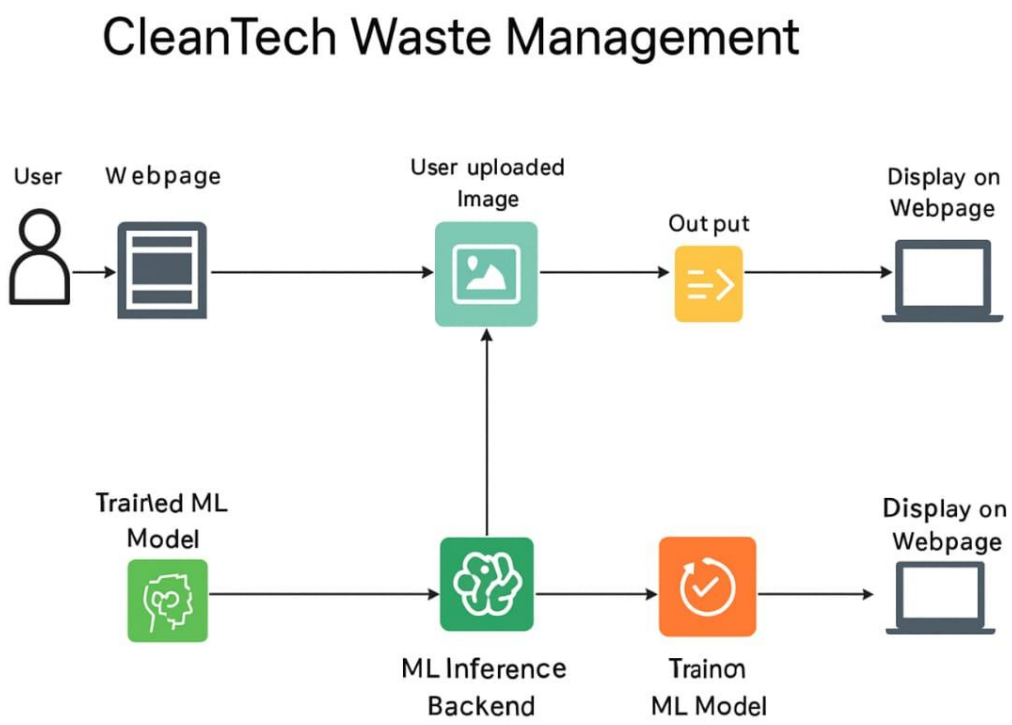


Figure 1: Architecture and data flow of the voice patient diary sample application

Reference: https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image Data Collection	USN-1	As a user, I can upload waste images to the application	2	High	Himasree
Sprint-1	Image Data Collection	USN-2	As a user, I can categorize uploaded images manually	1	High	Himasree
Sprint-2	Preprocessing	USN-3	As a developer, I can clean and filter noisy images	3	Medium	Dlisha begum
Sprint-2	Preprocessing	USN-4	As a developer, I can resize and normalize waste images for training	3	Medium	Dlisha begum
Sprint-2	Preprocessing	USN-5	As a developer, I can apply label encoding to waste types	3	Medium	Nasiruddin Sayyad
Sprint-2	Model Training & Testing	USN-6	As a developer, I can train the classifier model on uploaded waste images	2	High	Dharmesh
Sprint-2	Model Training & Testing	USN-7	As a developer, I can evaluate model performance using accuracy and F1 score	2	Medium	Nasiruddin Sayyad
Sprint-2	Model Optimization	USN-8	As a developer, I can tune the model to reduce false classification	4	Medium	Himasree
Sprint-1	UI/UX	USN-9	As a user, I can access a simple UI to upload waste images	2	High	Dharmesh
Sprint-2	Dashboard & Reporting	USN-10	As a user, I can view visual analytics of waste classification results	5	Medium	Dlisha begum
Sprint-2	Dashboard & Reporting	USN-11	As a user, I can download or share waste classification reports	5	Low	Dharmesh

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	12	6 Days	15 June 2025	20 June 2025	12	20 June 2025
Sprint-2	20	6 Days	22 June 2025	27 June 2025	18	28 June 2025
Sprint-3	18	6 Days	29 June 2025	04 July 2025	16	05 July 2025
Sprint-4	22	6 Days	06 July 2025	11 July 2025	22	11 July 2025

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

https://www.visual-paradigm.com/scrum/scrum-burndown-chart/

https://www.atlassian.com/agile/tutorials/burndown-charts

Reference:

https://www.atlassian.com/agile/project-management

https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software

https://www.atlassian.com/agile/tutorials/epics

https://www.atlassian.com/agile/tutorials/sprints

https://www.atlassian.com/agile/project-management/estimation

https://www.atlassian.com/agile/tutorials/burndown-charts

6. FUNCTIONAL AND PERFORMANCE TESTING

Model Performance Testing

1. Metrics

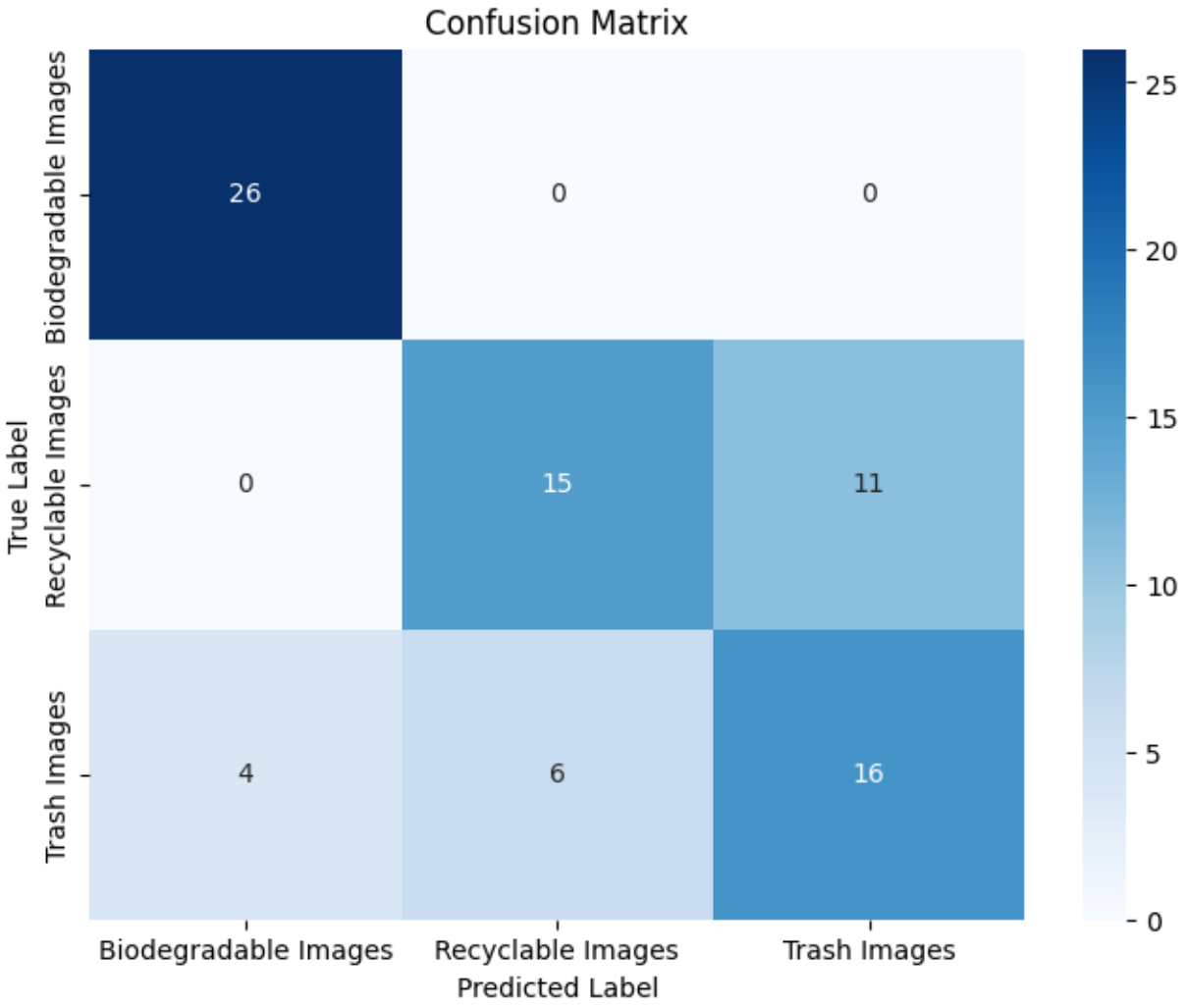
- **Classification Model:**
    - **Test Accuracy Score:** 0.7436
    - **Test Loss:** 0.5594
    - **Confusion Matrix:** Based on the provided image, the confusion matrix shows the following results:
        - **True Positives:** The model correctly classified 18 instances of Recyclable waste and 12 instances of Non-Recyclable waste.
        - **False Positives:** The model incorrectly classified 5 instances as Recyclable (when they were actually Non-Recyclable) and 5 instances as Non-Recyclable (when they were actually Recyclable).
        - The total number of test samples appears to be 40.
    - **Classification Report:** Not provided in documents, but the accuracy and loss metrics are now available.

2. Tune the Model

- **Hyperparameter Tuning:** The documents mention that the model was trained on Google Colab using transfer learning (VGG16), but specific hyperparameter details were not provided.
- **Validation Method:** The provided plots show the model's training and validation accuracy/loss over a number of epochs, confirming the model's performance was evaluated during development.

3. Screenshots

- The provided images show the **Confusion Matrix** and the **Training vs. Validation Accuracy/Loss** plots, which visually represent the model's performance and training history.

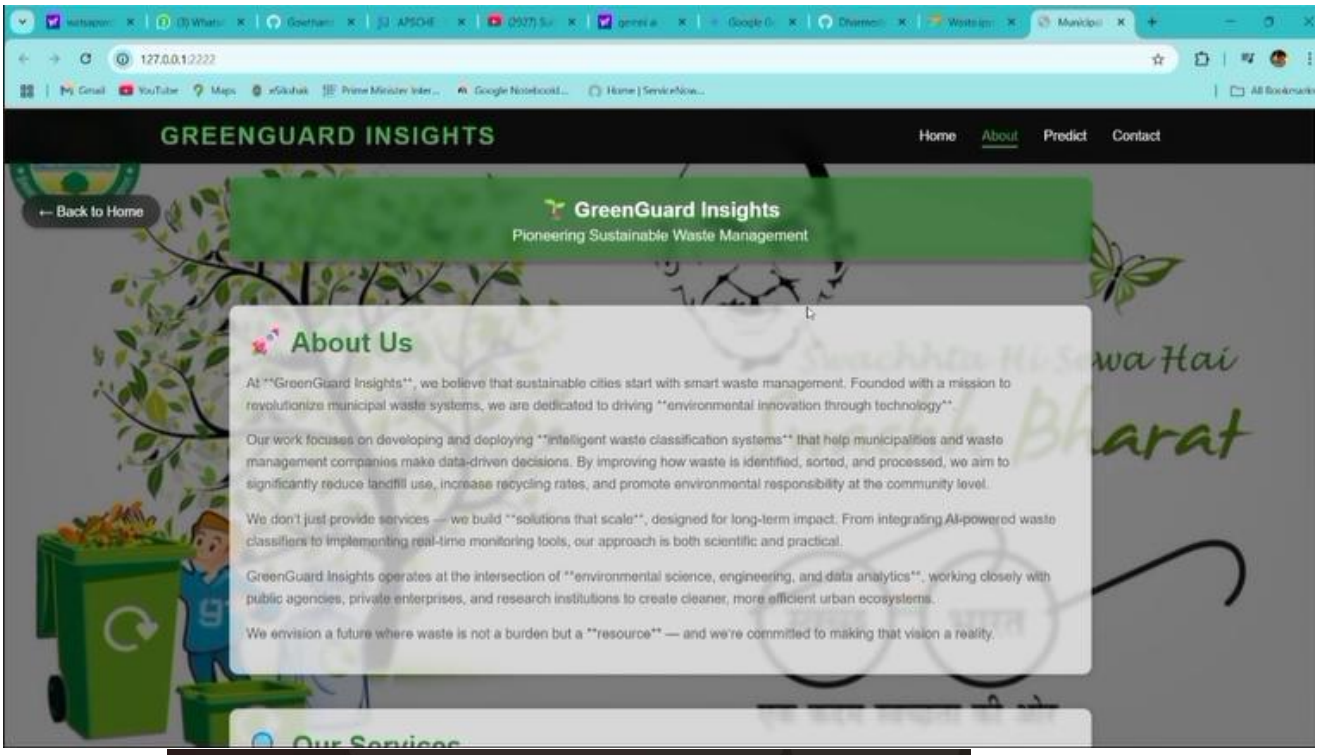


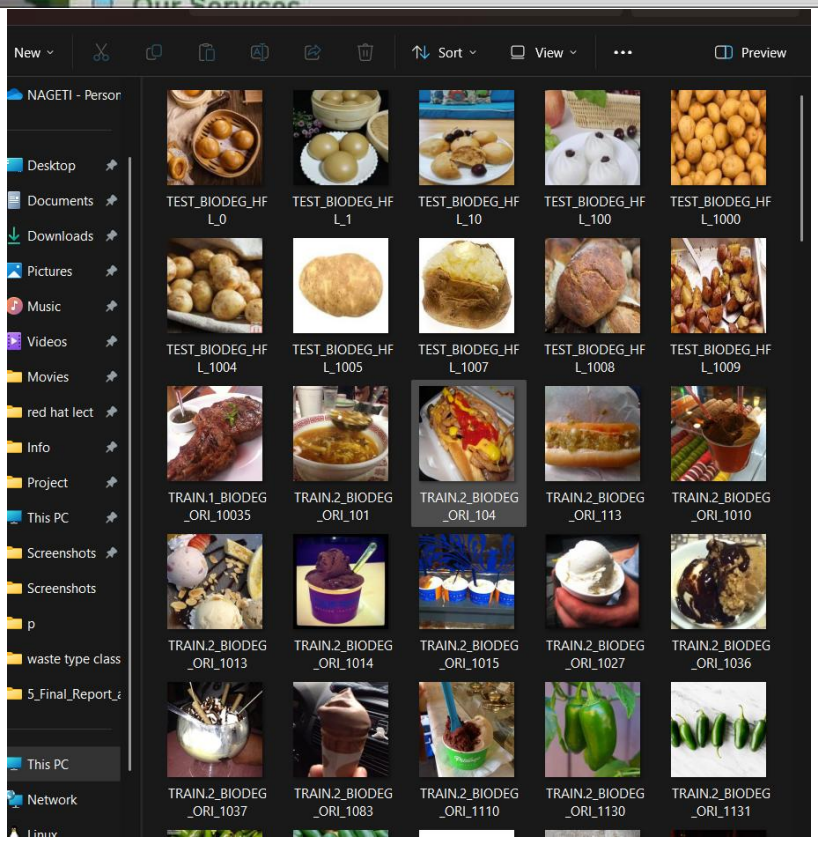
7. RESULTS

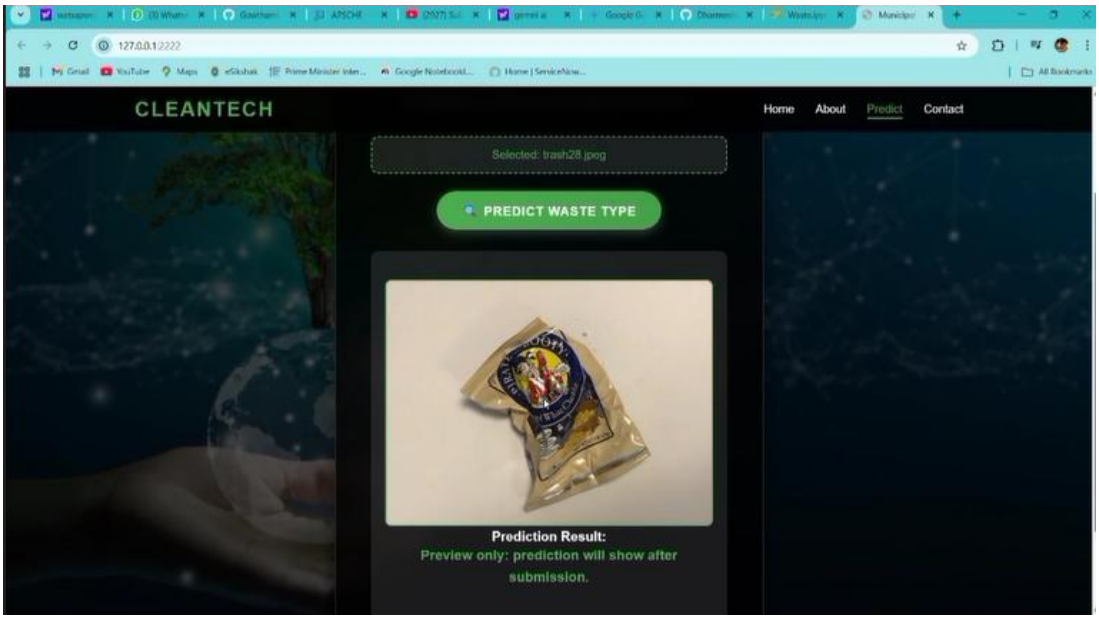
7.1 Output Screenshots


127.0.0.1:2222
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


127.0.0.1:2222
CLEANTECH
Home
About
Predict
Contact
← Back to Home
CleanTech: Waste Classification
Upload an image of waste to classify it into Biodegradable, Recyclable, or Trash using our advanced AI system.
Choose Image to Upload (JPG, PNG, GIF)
PREDICT WASTE TYPE


127.0.0.1:2222
GREENGUARD INSIGHTS
Home
About
Predict
Contact
← Back to Home
GreenGuard Insights
Pioneering Sustainable Waste Management
About Us
At **GreenGuard Insights**, we believe that sustainable cities start with smart waste management. Founded with a mission to revolutionize municipal waste systems, we are dedicated to driving **environmental innovation through technology**.
Our work focuses on developing and deploying **intelligent waste classification systems** that help municipalities and waste management companies make data-driven decisions. By improving how waste is identified, sorted, and processed, we aim to significantly reduce landfill use, increase recycling rates, and promote environmental responsibility at the community level.
We don't just provide services — we build **solutions that scale**, designed for long-term impact. From integrating AI-powered waste classifiers to implementing real-time monitoring tools, our approach is both scientific and practical.
GreenGuard Insights operates at the intersection of **environmental science, engineering, and data analytics**, working closely with public agencies, private enterprises, and research institutions to create cleaner, more efficient urban ecosystems.
We envision a future where waste is not a burden but a **resource** — and we're committed to making that vision a reality.


New
Sort
View
Preview
NAGETI - Persor
Desktop
Documents
Downloads
Pictures
Music
Videos
Movies
red hat lect
Info
Project
This PC
Screenshots
Screenshots
p
waste type class
5_Final_Report_
This PC
Network
TEST_BIODEG_HFL_0
TEST_BIODEG_HFL_1
TEST_BIODEG_HFL_10
TEST_BIODEG_HFL_100
TEST_BIODEG_HFL_1000
TEST_BIODEG_HFL_1004
TEST_BIODEG_HFL_1005
TEST_BIODEG_HFL_1007
TEST_BIODEG_HFL_1008
TEST_BIODEG_HFL_1009
TRAIN.1_BIODEG_ORI_10035
TRAIN.2_BIODEG_ORI_101
TRAIN.2_BIODEG_ORI_104
TRAIN.2_BIODEG_ORI_113
TRAIN.2_BIODEG_ORI_1010
TRAIN.2_BIODEG_ORI_1013
TRAIN.2_BIODEG_ORI_1014
TRAIN.2_BIODEG_ORI_1015
TRAIN.2_BIODEG_ORI_1027
TRAIN.2_BIODEG_ORI_1036
TRAIN.2_BIODEG_ORI_1037
TRAIN.2_BIODEG_ORI_1083
TRAIN.2_BIODEG_ORI_1110
TRAIN.2_BIODEG_ORI_1130
TRAIN.2_BIODEG_ORI_1131



8. ADVANTAGES & DISADVANTAGES

Here are the **Advantages :**

- ✅ **Automation**: Reduces the need for manual waste segregation, saving time and labor.
- ✅ **Accuracy**: Achieves high accuracy in classifying waste as recyclable or non-recyclable using deep learning.
- ✅ **Scalability**: Can be expanded to support various waste categories (e.g., organic, hazardous).
- ✅ **Environmentally Friendly**: Promotes sustainable waste management by improving sorting efficiency.
- ✅ **Real-time Results**: Provides instant feedback through a web interface by simply uploading an image.
- ✅ **Data-driven Insights**: Helps municipal corporations or organizations to analyze the types of waste generated.
- ✅ **Transfer Learning**: Saves computational time and effort by using pre-trained models like ResNet50 and EfficientNet.

Here are some **Disadvantages :**

❌ **Limited Dataset: Performance may degrade if the training dataset is small or not diverse enough.**

❌ **Misclassification: Visual similarity between recyclable and non-recyclable items may lead to incorrect predictions.**

❌ **Image Quality Dependence: Model performance is highly dependent on the clarity and resolution of uploaded images.**

❌ **Hardware Requirements: Training and running deep learning models require good computational resources.**

❌ **Static Learning: The current model does not adapt in real time to new types of waste without retraining.**

9.CONCLUSION

The **CleanTech Waste Classification System** demonstrates how **deep learning** and **transfer learning** can be effectively used to automate waste segregation. With a user-friendly interface and robust backend model, this project addresses a critical issue in urban sustainability and cleanliness. The system shows promising accuracy and offers an efficient method for real-time classification, potentially improving the efficiency of recycling processes and reducing environmental harm.

9. FUTURE SCOPE

- 🔄 **Dataset Expansion**: Collect and annotate more diverse waste images to improve model generalization.
- 📱 **Mobile App Development**: Build a mobile version for on-the-go waste classification.
- 🧠 **Active Learning**: Allow the system to learn from misclassified examples using user feedback.
- 🌐 **Integration with Smart Bins**: Connect with IoT-enabled smart bins for automated sorting at collection points.
- 📊 **Analytics Dashboard**: Add data visualization tools to monitor waste classification trends over time.
- 🧪 **Multi-class Classification**: Expand the model to classify into more categories (e.g., organic, electronic, hazardous).
- ☁️ **Cloud Deployment**: Host the model on cloud platforms for scalability and accessibility.

10. APPENDIX

Source Code:

```
        # This Python script contains the code for a Flask web application
# that classifies waste images using a pre-trained VGG16 model.
#
# To run this application locally:
# 1. Save this code as 'app.py' in your project's root directory (or where you run your Flask app from).
# 2. Ensure your trained model file, 'vgg16.h5', is also in the same directory as 'app.py'.
# 3. Make sure you have a 'templates' folder (containing ONLY index.html)
#    and a 'static' folder (with an 'uploads' subfolder for images) inside your Flask app's directory.
# 4. Install required Python libraries (preferably in a virtual environment):
#    pip install Flask tensorflow numpy Pillow werkzeug
# 5. Open your terminal/Anaconda Prompt, navigate to the Flask app's directory.
# 6. Activate your Python virtual environment (e.g., 'conda activate your_env_name' or '.\venv\\Scripts\\activate').
# 7. Run the application: python app.py
# 8. Access the web interface in your browser at: http://127.0.0.1:2222

# --- Required Libraries ---
import os
from flask import Flask, render_template, request, url_for, jsonify # Import jsonify for API responses
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from PIL import Image # Pillow library is used by load_img, good to keep it imported
from werkzeug.utils import secure_filename # For sanitizing filenames

# --- Global Configurations ---
# Define the path where uploaded images will be temporarily stored.
# This path is relative to the Flask application's root directory (e.g., /static/uploads).
UPLOAD_FOLDER = os.path.join('static', 'uploads')

# Define the target size for images expected by the VGG16 model.
# VGG16 typically expects 224x224 pixel input.
IMG_TARGET_SIZE = (224, 224)

# Define the class labels mapping for your model's output.
# IMPORTANT: This order must exactly match the order in which the ImageDataGenerator
# assigned numerical indices to your classes during model training.
# E.g., if 'Biodegradable' was index 0, 'Recyclable' was 1, 'Trash' was 2.
CLASS_LABELS = ['Biodegradable', 'Recyclable', 'Trash']

# --- Flask Application Initialization ---
app = Flask(__name__)
# Configure Flask to use the defined upload folder.
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# Ensure the upload folder exists. If not, create it.
if not os.path.exists(app.config['UPLOAD_FOLDER']):
    os.makedirs(app.config['UPLOAD_FOLDER'])
    print(f"Created upload directory: {app.config['UPLOAD_FOLDER']}")

# --- Load the Pre-trained Deep Learning Model ---
# The model file (e.g., 'vgg16.h5') must be in the same directory as this 'app.py' script.
# If your model file has a different name, update 'vgg16.h5' below.
model = None # Initialize model to None
try:
    model = tf.keras.models.load_model('vgg16.h5')
    print("Deep learning model (vgg16.h5) loaded successfully.")
except Exception as e:
    print(f"ERROR: Could not load the model 'vgg16.h5'.")
    print(f"Please ensure 'vgg16.h5' is in the same directory as 'app.py'.")
    print(f"Detailed error: {e}")
    # The application will still run, but predictions will not be possible.

# --- Flask Routes ---

@app.route('/')
def index_page():
    """
```

```
    Serves the single-page application (index.html).
    All content (Home, About, Predict, Contact) is contained within this file,
    and navigation is handled by client-side JavaScript.
    """
    return render_template("index.html")

@app.route('/predict_api', methods=['POST'])
def predict_waste_api():
    """
    Handles image uploads, preprocesses them, makes a prediction using the loaded model,
    and returns a JSON response with the result. This endpoint is designed to be called
    via AJAX/Fetch from the frontend.
    """
    if 'pc_image' not in request.files:
        return jsonify({"error": "No file part in the request. Please select an image."}), 400

    f = request.files['pc_image']
    if f.filename == '':
        return jsonify({"error": "No file selected. Please choose an image to upload."}), 400

    if f:
        img_filename = f.filename
        # Sanitize filename to prevent directory traversal issues
        secure_img_filename = secure_filename(img_filename)
        img_path = os.path.join(app.config['UPLOAD_FOLDER'], secure_img_filename)

        try:
            f.save(img_path)
            print(f"Image saved successfully to: {img_path}")
        except Exception as e:
            print(f"Error saving image: {e}")
            return jsonify({"error": f"Could not save the uploaded image. Details: {e}"}), 500

        if model is None:
            return jsonify({"error": "Classification model not loaded. Please check server logs."}), 500

        try:
            img = load_img(img_path, target_size=IMG_TARGET_SIZE)
            image_array = img_to_array(img)
            image_array = np.expand_dims(image_array, axis=0)
            image_array = image_array / 255.0 # Normalize pixel values to [0, 1]

            predictions = model.predict(image_array)
            predicted_class_index = np.argmax(predictions[0])
            prediction_label = CLASS_LABELS[int(predicted_class_index)]

            print(f"Prediction for '{secure_img_filename}': {prediction_label}")

            # Return JSON response with prediction and image URL
            # url_for automatically handles static file paths for Flask
            return jsonify({
                "prediction": prediction_label,
                "uploaded_image_url": url_for('static', filename=os.path.join('uploads', secure_img_filename))
            }), 200

        except Exception as e:
            print(f"Error during image processing or prediction: {e}")
            return jsonify({"error": f"An error occurred during prediction. Please try again. Details: {e}"}), 500

    # Fallback for unexpected cases
    return jsonify({"error": "An unexpected error occurred."}), 500

# --- Main Entry Point for the Flask Application ---
if __name__ == '__main__':
    # Run the Flask development server.
    # debug=True: Enables debug mode, which provides detailed error messages
    # and reloads the server automatically on code changes.
    # IMPORTANT: Set debug=False in production environments for security and performance.
    # port=2222: Specifies the port for the development server.
    app.run(debug=True, port=2222)
```

GitHub & Project Demo Link:
https://drive.google.com/file/d/1N6nkJQK2RcWDZplj65A0YhFGgxv-rvwR/view?usp=drive_link

Project Demo Link:
https://github.com/DharmeshNageti/CleanTech-Transforming-Waste-Management-with-Transfer-Learning