**Intelligent**

**Garbage Classification using Deep learning**

**A PROJECT REPORT**

Submitted to the

**LMS PORTAL**

**By**

**PAMIDI ESWAR (20KP1A1226)**

**SHAIK JANI BASHA (20KP1A1239)**

**P.MANI KUMAR (20KP1A4440)**

**G.RANAPRATAPRAM (20KP1A4410)**

**THANGELLA NAGA SRANANI (20KP1A1243)**



**NRI INSTITUTE OF TECHNOLOGY**

**VISADALA (M), GUNTUR(D), ANDHRA PRADESH**

**Project Report**

1. **INTRODUCTION**

1.1 Project Overview

1.2 Purpose

2. **IDEATION & PROPOSED SOLUTION**

2.1 Problem Statement Definition

2.2 Empathy Map Canvas

2.3 Ideation & Brainstorming

2.4 Proposed Solution

3. **REQUIREMENT ANALYSIS**

3.1 Functional requirement

3.2 Non-Functional requirements

**4. PROJECT DESIGN**

4.1 Data Flow Diagrams

4.2 Solution & Technical Architecture

4.3 User Stories

**5. CODING & SOLUTIONING**

5.1 Feature 1

5.2 Feature 2

**6. RESULTS**

6.1 Performance Metrics

**7. ADVANTAGES & LIMITATIONS**

**8.** **CONCLUSION**

**9. FUTURE SCOPE**

**10. APPENDIX**

10.1 Source Code

**1. Introduction:**

* 1. **Project Overview:**

**Objective:**

The project aims to develop an intelligent garbage classification system that utilizes deep learning techniques to accurately categorize different types of waste items.

**Importance:**

Effective garbage classification is crucial for sustainable waste management and recycling efforts. By automating the classification process using deep learning, we can improve efficiency and reduce human error.

By developing an intelligent garbage classification system using deep learning, this project aims to revolutionize waste management practices, contributing to a cleaner and more sustainable environment.

* 1. **Purpose:**

The purpose of Intelligent Garbage Classification using Deep Learning is to address the challenges and inefficiencies in traditional waste management systems by leveraging the power of artificial intelligence and deep learning techniques. The key purposes of this project are as follows:

1. Accurate Classification: The primary purpose is to develop a system that can accurately classify different types of garbage items. Deep learning algorithms excel at extracting intricate patterns and features from images, enabling more precise and reliable classification compared to manual sorting.

2. Automation and Efficiency: By automating the garbage classification process, the project aims to reduce the reliance on manual sorting, which can be time-consuming, subjective, and error-prone. Deep learning models can efficiently analyse vast amounts of data, leading to improved efficiency in waste management systems.

3. Sustainable Waste Management: The project aligns with the broader objective of sustainable waste management. By accurately categorizing garbage items, it becomes easier to implement appropriate recycling, composting, and disposal methods. This, in turn, helps reduce the environmental impact of waste and encourages a more circular economy.

4. Resource Optimization: Intelligent garbage classification allows for optimal resource allocation in waste management systems. By identifying recyclable materials accurately, valuable resources can be recovered efficiently, reducing waste and promoting resource conservation.

5. Improved Recycling Rates: Deep learning-based garbage classification systems have the potential to enhance recycling rates by ensuring that recyclable items are correctly identified and separated. This can lead to increased diversion of waste from landfills and a more efficient recycling process.

6. Education and Awareness: The project serves as an educational tool to raise awareness about proper waste classification and its impact on the environment. By showcasing the capabilities of deep learning in garbage classification, the project can help promote responsible waste management practices among individuals, communities, and organizations.

7. Scalability and Adaptability: An intelligent garbage classification system can be easily scaled and deployed in various settings, ranging from households and recycling centres to smart waste management infrastructure. Its adaptability allows for future improvements, including the ability to handle new waste types or changing patterns.

In summary, the purpose of Intelligent Garbage Classification using Deep Learning is to revolutionize waste management systems by automating and improving the accuracy of garbage classification, contributing to sustainable waste management practices, resource optimization, and increased recycling rates.

1. **IDEATION & PROPOSED SOLUTION:**
   1. **Problem Statement Definition:**

**Intelligent Garbage Classification using Deep learning:**

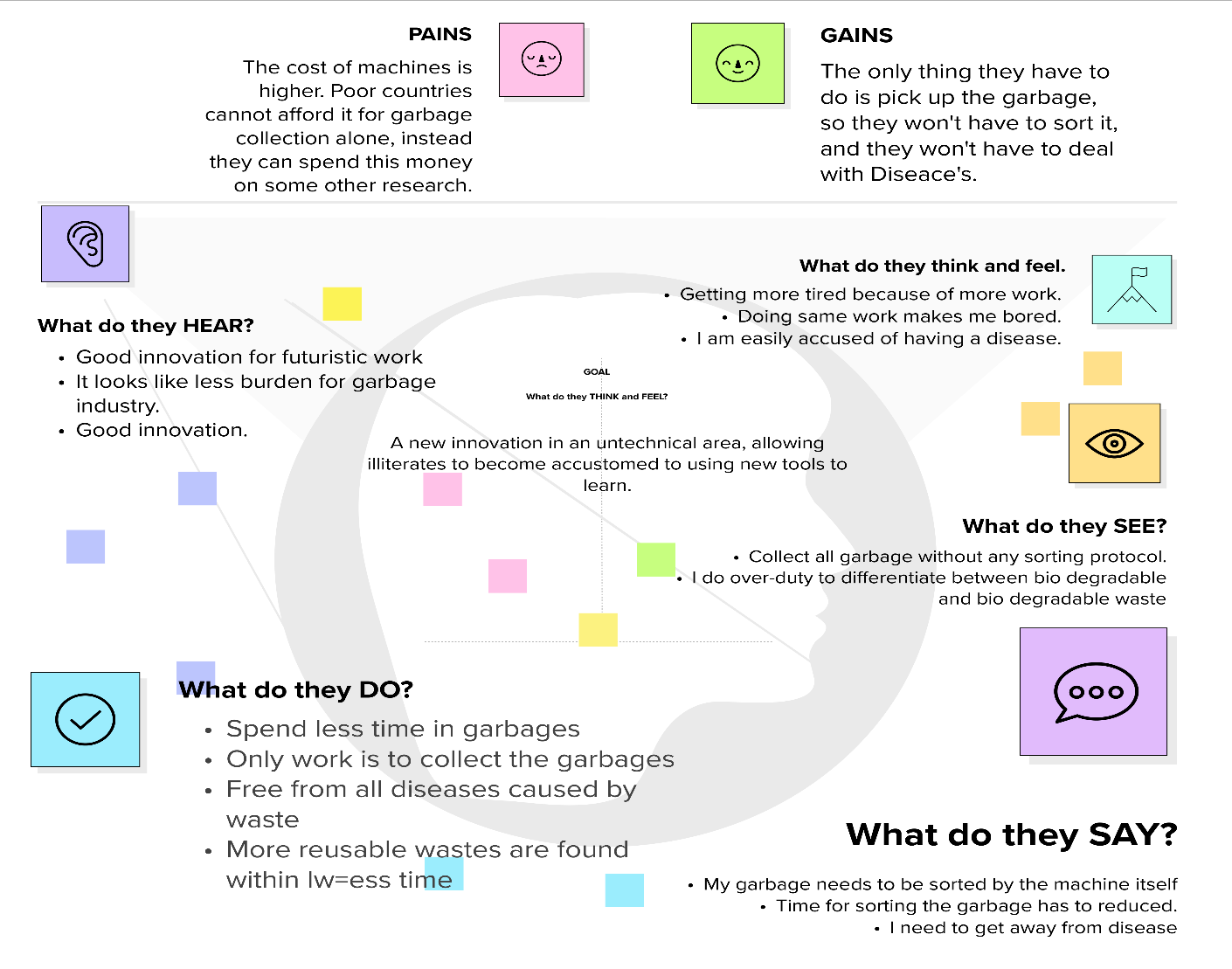


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Statement (PS)** | **I am (Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | Garbage Collector | Sort the garbage | It was done manually by me. | Manually sorting the garbage is the only option. | Hard to work. |
| PS-2 | Garbage Collector | Collect the garbage | I need to sort it by hands | There can’t be any machinery to be introduced in garbage system. | I am prone to more Diseases caused by collecting waste. |

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

* 1. **Empathy Map Canvas:**

**Intelligent Garbage Classification using Deep learning:**

 Reference:[**https://app.mural.co/t/venu2338/m/venu2338/1682783111176/848d0dd9a5e0db8db7031922f071beb6e9fa1353?sender=ud1888acbdfd6d3dbe9798783**](https://app.mural.co/t/venu2338/m/venu2338/1682783111176/848d0dd9a5e0db8db7031922f071beb6e9fa1353?sender=ud1888acbdfd6d3dbe9798783)

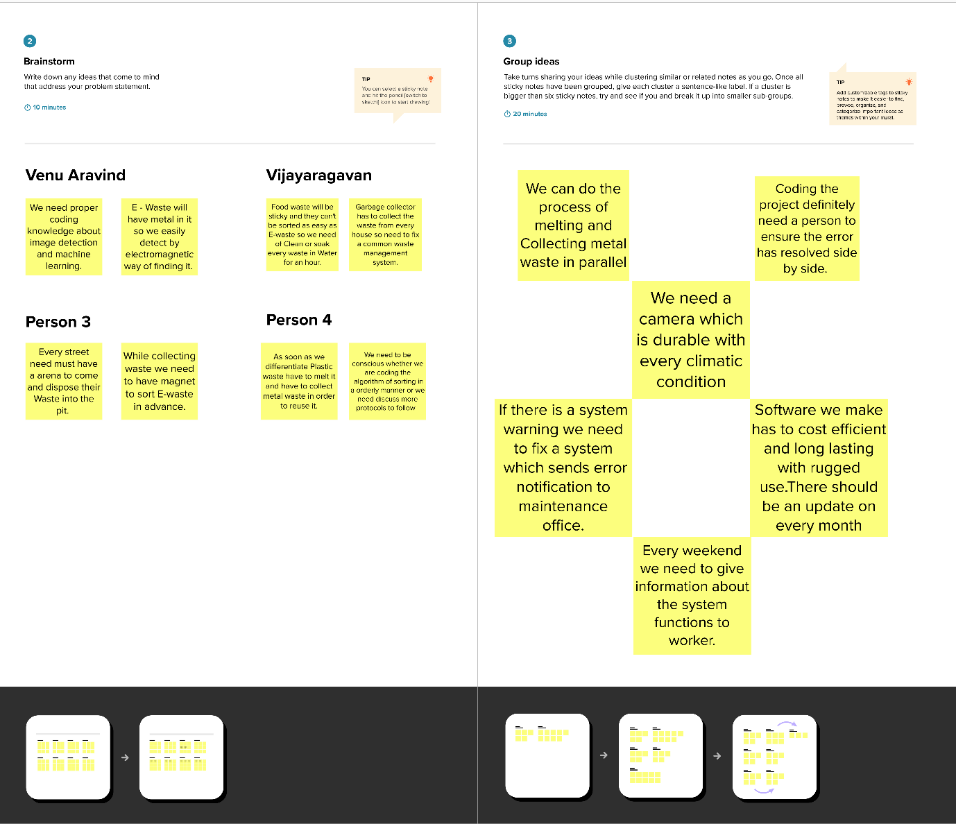
* 1. **Ideation & Brainstorming:**

Reference: <https://app.mural.co/t/venu2338/m/venu2338/1682837189672/91686130cef00aae5659e3cc146528d5f9c27362?sender=ud1888acbdfd6d3dbe9798783>

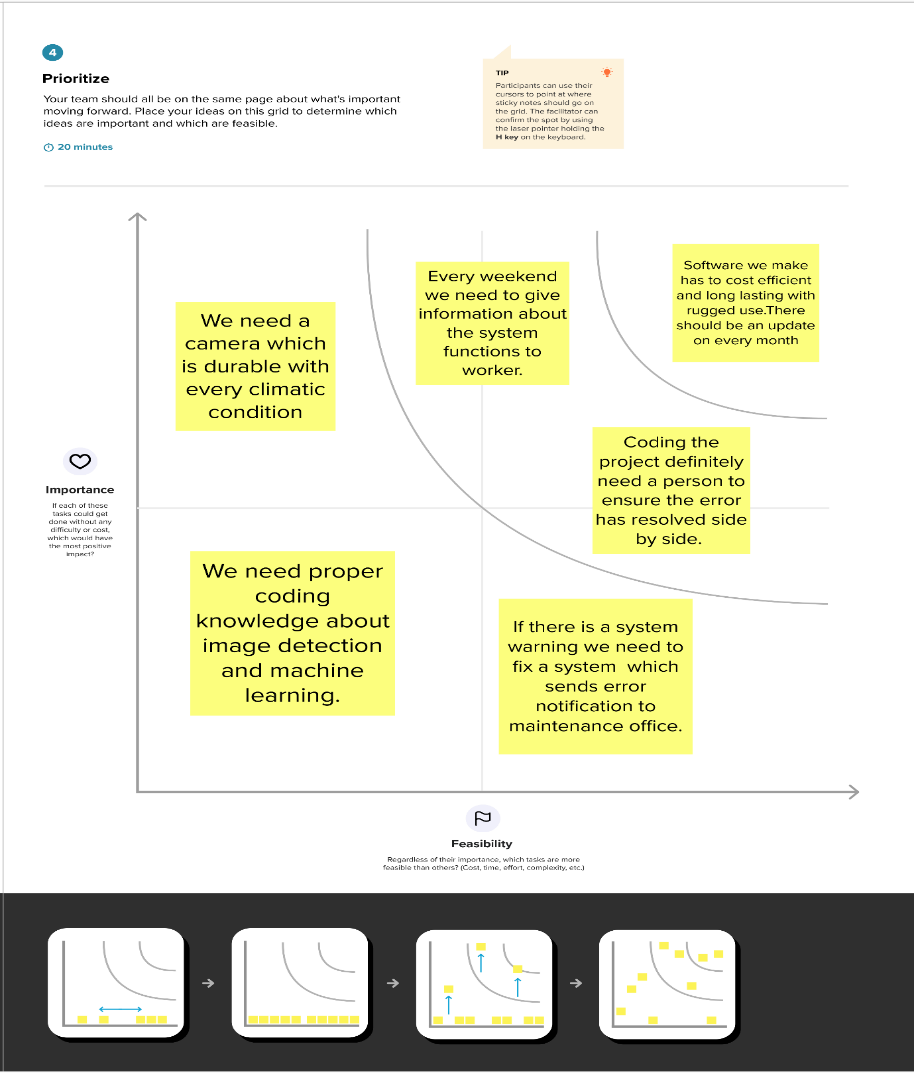
**Step-1: Team Gathering, Collaboration and Select the Problem**



**Step-2: Brainstorm, Idea Listing and Grouping**



**Step-3: Idea Prioritization**



* 1. **Proposed Solution:**

|  |  |  |
| --- | --- | --- |
| **S.no.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | Garbage classification is a critical aspect of waste management, which involves sorting and processing different types of waste materials to minimize their impact on the environment. However, manual garbage classification can be time-consuming, error-prone, and potentially hazardous for workers.  Therefore, there is a need for an automated garbage classification system that can accurately identify different types of waste materials and route them to the appropriate waste streams. |
|  | Idea / Solution description | In this project, we propose an intelligent garbage classification system that uses deep learning techniques to automatically classify garbage images into different categories, such as plastic, paper, metal, etc. The system will consist of a deep convolutional neural network (CNN) model that can learn to recognize the visual patterns and features of different types of garbage. The model will be trained on a large dataset of garbage images and validated using standard evaluation metrics. Once the model is trained, it will be deployed on a cloud infrastructure that can handle real-time image ingestion and classification.  The proposed system will offer several advantages over manual garbage classification, including faster processing times, improved accuracy, and reduced labour costs. It will also enable better waste management practices, as garbage can be sorted and processed more efficiently, leading to reduced environmental impact.  Overall, the proposed intelligent garbage classification system has the potential to revolutionize waste management practices and contribute to a more sustainable future. |
|  | Novelty / Uniqueness | Firstly, deep learning allows for more accurate and consistent classification, as the model can learn to recognize subtle visual patterns and features that may be difficult for humans to discern. This can lead to improved sorting accuracy and reduced contamination in waste streams.  Secondly, the use of deep learning enables faster processing times and scalability, as the model can handle large volumes of garbage images in real-time. This can lead to improved efficiency in waste management practices, reducing costs and increasing throughput.  Lastly, the proposed solution can be easily integrated into existing waste management systems, providing an automated and reliable way to sort and process different types of waste materials.  Overall, the use of deep learning for garbage classification is a novel and unique approach that has the potential to revolutionize waste management practices and contribute to a more sustainable future. |
|  | Social Impact / Customer Satisfaction | From a social impact perspective, the proposed solution can help to address a number of environmental and social issues related to waste management. For example, by reducing contamination in waste streams, it can help to improve recycling rates and reduce the amount of waste sent to landfills. This can have a positive impact on the environment, reducing pollution and greenhouse gas emissions, and helping to conserve natural resources.  In addition, the proposed solution can also improve worker safety and working conditions, as it reduces the need for manual sorting and processing of garbage. This can lead to a safer and healthier work environment, reducing the risk of accidents and injuries. |
|  | Business Model (Revenue Model) | * Subscription Model: In this model, waste management companies or municipalities can pay a monthly or annual subscription fee to access the Intelligent Garbage Classification system. This can include access to the deep learning model, cloud infrastructure, and ongoing technical support and maintenance. * Pay-per-Use Model: In this model, waste management companies or municipalities can pay for each image classification request processed by the system. This can be useful for companies with fluctuating volumes of garbage images, as they only pay for the classifications they need. * Licensing Model: In this model, the deep learning model and associated intellectual property can be licensed to waste management companies or municipalities for a fee. This can provide a source of ongoing revenue for the developers of the solution. * Value-Added Services Model: In this model, the Intelligent Garbage Classification system can be integrated with other waste management technologies, such as routing optimization or analytics platforms. This can provide additional value to customers and enable the developers to charge a premium for their solution. |
|  | Scalability of the Solution | Dataset Size: The size and diversity of the dataset used to train the deep learning model can impact scalability. A larger and more diverse dataset can lead to a more robust and accurate model that can handle a wider range of garbage images.  Model Architecture: The architecture and design of the deep learning model can impact scalability, as some architectures may be more computationally intensive than others. Choosing an architecture that balances accuracy with efficiency can improve scalability.  Computational Resources: The availability of computational resources, such as GPUs and cloud infrastructure, can impact scalability. Ensuring that sufficient resources are available to handle large volumes of garbage images can improve scalability.  Real-time Processing: The ability to process garbage images in real-time can impact scalability. Ensuring that the deep learning model is optimized for real-time processing and that the system architecture can handle large volumes of image requests can improve scalability.  Deployment Flexibility: The ability to deploy the Intelligent Garbage Classification system across a variety of platforms and environments can improve scalability. This can include deploying the system on-premise or in the cloud, and integrating with existing waste management systems. |

**3 REQUIREMENT ANALYSIS:**

**3.1 Functional requirement:**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Garbage Classification System by deep learning AI | * Develop an image capture feature in the system's user interface. * Implement deep learning AI algorithms to analyse the captured image. * Classify the garbage item accurately into recyclable, organic, or non-recyclable category. * Display the classification result to the user in real-time. |
| FR-2 | Garbage Classification System by deep learning AI | * Update the user interface to display the classification result immediately after image analysis. * Ensure the feedback is provided within a few seconds to maintain a smooth user experience. |
| FR-3 | Garbage Classification System by deep learning AI | * Train the deep learning AI model using a diverse dataset containing different types of garbage items. * Implement classification models for plastic, paper, glass, metal, food waste, electronic waste, and other relevant categories. * Ensure the system can accurately classify each type of garbage item. |
| FR-4 | Garbage Classification System by deep learning AI | * Develop a feedback mechanism for users to report misclassified items. * Collect and analyse user feedback data to identify patterns and improve the classification model. * Regularly update the deep learning AI model with new training data to enhance accuracy. |
| FR-5 | Garbage Classification System by deep learning AI | * Enhance the user interface to display relevant information alongside the classification result. * Provide instructions on proper disposal methods and recycling guidelines for each classified item. * Ensure the information is accurate, up-to-date, and helps users make informed waste management decisions. |
| FR-6 | Garbage Classification System by deep learning AI | * Develop an administrator dashboard with performance metrics, user feedback analysis, and database management functionalities. * Provide real-time monitoring of the system's performance, including accuracy rates and user activity. * Enable administrators to review and analyse user feedback, address reported issues, and manage the database of garbage items |

**3.2 Non-Functional requirements:**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | * The user interface should be intuitive, user-friendly, and easy to navigate. * Clear instructions and guidance should be provided to users on how to interact with the system. * The system should support multiple languages and be accessible to users with disabilities. |
| NFR-2 | **Security** | * The system should implement appropriate security measures to protect user data and ensure user privacy. * It should have secure authentication and authorization mechanisms to prevent unauthorized access. * Data transmission should be encrypted to safeguard against eavesdropping or data interception. |
| NFR-3 | **Reliability** | * The system should be available and operational 24/7, with minimal downtime for maintenance or upgrades. * It should be resilient to failures, including network disruptions or server crashes, and have mechanisms in place for fault tolerance and data recovery. |
| NFR-4 | **Performance** | * The system should provide real-time classification results within seconds of capturing the image. * The classification accuracy should meet or exceed a predetermined threshold. * The system should be able to handle a high volume of user requests simultaneously without significant performance degradation. |
| NFR-5 | **Availability** | * The system should be available and operational 24/7, with minimal downtime for maintenance or upgrades. * It should have a robust infrastructure and backup mechanisms to ensure high availability and minimize service disruptions. * The system should have a monitoring system in place to promptly identify and address any availability issues. * Planned maintenance activities should be scheduled during periods of low user activity to minimize impact on availability. * The system should have mechanisms for graceful degradation or failover in case of unexpected failures or outages. |
| NFR-6 | **Scalability** | * The system should be designed to accommodate a growing user base and increasing data volume. * It should support horizontal scalability, allowing for the addition of more computational resources to handle higher loads. |

**4 PROJECT DESIGN:**

**4.1 Data Flow Diagrams:**



**4.2 Solution & Technical Architecture:**



**4.3 User Stories:**

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Team Member** |
| --- | --- | --- | --- | --- | --- | --- |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-1 | As a user, I want to be able to take a picture of a garbage item and receive an accurate classification of whether it belongs to recyclable, organic, or non-recyclable category. | - The system should provide a user interface to capture images of garbage items.  - The system should use deep learning AI algorithms to analyse the images and classify the garbage item accurately.  - The classification result should be displayed to the user in real-time. | High | Venu Aravind R |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-2 | As a user, I want the Garbage Classification System to provide real-time feedback on the classification results. | - The system should display the classification result immediately after the user submits the image.  - The feedback should be provided within a few seconds to ensure a smooth user experience. | High | Vijayaragavan S |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-3 | As a user, I want the system to be user-friendly, allowing me to easily interact with it through a mobile application or a web interface. | - The system should have a well-designed and intuitive user interface.  - The interface should be responsive and accessible from both mobile devices and web browsers.  - The user should be able to navigate through the system easily and perform the required actions without confusion. | Medium | Chidambaram N |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-4 | As a user, I want the system to handle various types of garbage items, including but not limited to plastic, paper, glass, metal, food waste, and electronic waste. | - The system should be trained to recognize and classify a wide range of garbage items.  - The system should provide accurate classification for common types of garbage, such as plastic, paper, glass, metal, food waste, and electronic waste. | High | Gowthaman P |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-5 | As a user, I want the system to be able to handle both single-item classification and bulk classification of multiple items at once. | - The system should allow users to classify individual garbage items one at a time.  - The system should also provide an option for users to upload multiple images for bulk classification.  - The bulk classification should process all the images and provide classification results for each item. | Medium | Chidambaram N |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-6 | As a user, I want the Garbage Classification System to have a high accuracy rate in correctly classifying different types of garbage items. | - The system should achieve a high accuracy rate in classifying common types of garbage items.  - The accuracy rate should be measured and validated through testing and validation datasets. | High | Chidambaram N |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-7 | As a user, I want the system to provide additional information and guidelines on how to properly dispose of each classified item, including recycling instructions or suggestions for proper waste management. | - The system should display additional information alongside the classification result, providing instructions on proper disposal or recycling methods for each classified item.  - The information should be accurate and up-to-date, helping users make informed decisions about waste management. | Medium | Vijayaragavan S |
| User  (Worker) | Garbage Classification System by deep learning AI | USN-8 | As a user, I want the system to continuously learn and improve its classification accuracy over time by incorporating user feedback and data from a large database of garbage items. | - The system should have a feedback mechanism where users can provide feedback on misclassified items.  - The system should incorporate user feedback and update its training data to improve classification accuracy.  - The system should periodically update its deep learning models using a large database of garbage items to enhance accuracy. | High | Gowthaman P |
| Administrator | Garbage Classification System by deep learning AI | USN-9 | As an administrator, I want to have access to a dashboard or backend system where I can monitor the system's performance, review user feedback, and manage the database of garbage items. | - The system should provide an administrator dashboard with an overview of the system's performance metrics, including accuracy rates and user feedback statistics.  - The dashboard should allow administrators to review and analyse user feedback on misclassified items.  - The system should provide functionality to manage the database of garbage items, including adding new items, updating existing items, or removing obsolete items. | High | Venu Aravind R |
| Developer | Garbage Classification System by deep learning AI | USN-10 | As a developer, I want the system to be built on a scalable and flexible architecture, allowing for easy integration with different platforms and future enhancements. | - The system should follow modular and scalable architecture principles, allowing for easy integration with different platforms such as mobile applications, web interfaces, or API endpoints.  - The system should be designed with flexibility in mind, enabling future enhancements or updates to the deep learning models or classification algorithms.  - The code should be well-documented and maintainable, facilitating collaboration among developers. | High | Venu Aravind R |

**5 CODING & SOLUTIONING:**

**5.1 Feature (Model building with Testing and Training):**

Data Collection: Collect a diverse dataset of garbage images, including various types such as plastic, paper, glass, metal, organic waste, etc.

Data Pre-processing: Clean and pre-process the dataset by removing noise, resizing images, and ensuring balanced representation of different waste types.

Model Selection: Choose a deep learning architecture suitable for image classification tasks, such as convolutional neural networks (CNNs), which excel at extracting meaningful features from images.

Model Training: Split the dataset into training and validation sets. Use the training set to train the deep learning model, adjusting the model's parameters to minimize the classification error.

Model Evaluation: Evaluate the trained model on the validation set to measure its performance, such as accuracy, precision, recall, and F1-score.

Model Fine-tuning: Fine-tune the model by iteratively adjusting hyperparameters and architecture to achieve optimal performance.

Model Deployment: Deploy the trained model on a suitable platform or integrate it into a user-friendly interface for real-time garbage classification.

**5.2 Feature (Using Flask module to Render the HTML templates):**

**Importing Required Libraries:**The necessary libraries are imported, including re, sys, numpy, os, Flask, request, render\_template, url\_for, redirect, models, load\_model, and image from Keras.

**Loading the Pre-trained Model:**

The pre-trained deep learning model is loaded using load\_model from Keras. The model file (garbage\_new.h5) is assumed to be located at the specified path.

**Flask App Initialization:**

The Flask application is initialized using Flask(\_\_name\_\_).

**Route Definitions:**

Several routes are defined using the @app.route decorator to handle different URLs:

The default route (/) renders the index.html template.

The /prediction.html route renders the prediction.html template.

The /index.html route renders the index.html template.

The /result route is the endpoint for receiving the image file and performing the garbage classification.

**Route Functions:**

The route functions are defined to handle the corresponding routes:

The index() function renders the index.html template.

The prediction() function renders the prediction.html template.

The home() function renders the index.html template.

The res() function handles the POST request to /result and performs the garbage classification.

It retrieves the uploaded image file and saves it in the uploads folder.

The image is loaded and pre-processed using image.load\_img and image.img\_to\_array.

The image array is expanded to match the expected input shape of the model.

The model predicts the class label using model.predict.

The predicted class label is converted to a human-readable category using the index list.

The prediction.html template is rendered with the predicted result.

**Running the Application:**

The application is run using app.run() with the debug flag set to True to enable debug mode, and the port set to 5000.

Flask allows you to define routes and corresponding functions to handle different HTTP requests. In this case, the application has routes for the home page, prediction page, and the result endpoint where the image classification takes place. The templates (index.html and prediction.html) contain the HTML code and placeholders for displaying the results.

When a user uploads an image through the web interface, the Flask application receives the image, processes it, performs classification using the pre-trained model, and displays the predicted result on the prediction.html page. (Note: Code is given below at Source code page)

**6 Results:**

**6.1 Performance Metrics**

When evaluating the performance of a garbage classification model built using deep learning techniques, several performance metrics can be used to assess its effectiveness. Here are some commonly used performance metrics for garbage classification:

1. Accuracy: Accuracy measures the proportion of correctly classified garbage items out of the total number of items. It provides an overall measure of how well the model performs. However, accuracy alone may not be sufficient if the classes are imbalanced.

2. Precision: Precision represents the proportion of correctly classified positive instances

(e.g., correctly identified as a specific type of garbage) out of all instances predicted as positive. It measures the model's ability to avoid false positives.

3. Recall (Sensitivity): Recall measures the proportion of correctly classified positive instances out of all actual positive instances. It indicates the model's ability to identify all relevant instances correctly.

4. F1-score: The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between precision and recall. F1-score is useful when there is an imbalance between the classes.

5. Confusion Matrix: A confusion matrix provides a tabular representation of the model's performance, showing the number of correct and incorrect predictions for each class. It can help identify specific areas where the model may be struggling, such as misclassifications between similar types of garbage.

6. Classification Report: A classification report provides a comprehensive summary of various performance metrics, including precision, recall, and F1-score, for each class in the classification task. It helps assess the model's performance on individual classes.

7. Receiver Operating Characteristic (ROC) Curve: ROC curves plot the true positive rate (TPR) against the false positive rate (FPR) at various classification thresholds. It provides insights into the model's trade-off between sensitivity and specificity.

8. Area Under the ROC Curve (AUC-ROC): AUC-ROC summarizes the overall performance of the model across various classification thresholds. It provides a single value that represents the model's ability to distinguish between different classes.

These performance metrics collectively provide a comprehensive evaluation of a garbage classification model's effectiveness. It is essential to consider these metrics in combination to gain insights into the model's strengths and weaknesses and to make

informed decisions regarding model improvements or deployment.

**7 ADVANTAGES & LIMITATIONS:**

**ADVANTAGES:**

* Improved Accuracy: Deep learning algorithms can achieve high levels of accuracy in garbage classification tasks. They can learn complex patterns and features from garbage images, resulting in more precise and reliable classification compared to manual sorting.
* Automation and Efficiency: By automating the garbage classification process, deep learning systems can save time and effort compared to manual sorting methods. This automation leads to improved efficiency in waste management systems, enabling faster and more accurate sorting
* Scalability: Deep learning models can be easily scaled to handle large volumes of garbage items. They can process a vast amount of data in parallel, making them suitable for high-throughput garbage classification scenarios, such as recycling centres or waste management facilities.
* Adaptability: Deep learning models can be trained to recognize a wide range of garbage types and adapt to new waste categories. This adaptability allows the system to be flexible and robust in handling different garbage items, even as waste streams evolve over time.
* Resource Optimization: Intelligent garbage classification using deep learning facilitates optimal resource allocation in waste management systems. By accurately identifying recyclable materials, valuable resources can be efficiently recovered, reducing waste and promoting resource conservation.
* Environmental Impact: Effective garbage classification through deep learning contributes to environmental sustainability. It enhances waste management practices, reduces landfill waste, promotes recycling and composting, and minimizes the negative impact on the environment.

**LIMITATIONS:**

* Data Requirements: Deep learning models typically require a large amount of labelled data for training. Collecting and labelling a diverse and representative dataset for garbage classification can be challenging and time-consuming.
* Model Complexity and Training Time: Deep learning models can be computationally expensive to train, requiring substantial computational resources and time. Training large-scale models on limited resources may lead to longer training times or a compromise in model complexity.
* Overfitting: Deep learning models are prone to overfitting, where the model performs well on the training data but fails to generalize to unseen data. Addressing overfitting requires careful regularization techniques, cross-validation, and proper dataset partitioning
* Interpretability: Deep learning models are often considered as black boxes because they lack interpretability. Understanding the underlying reasons for the model's predictions can be challenging, making it difficult to explain or justify the classification decisions to stakeholders.
* Limited Generalization: While deep learning models can achieve high accuracy within the training domain, they may struggle with generalizing to unseen scenarios or variations in garbage items. This limitation necessitates careful validation and testing on diverse datasets to ensure real-world applicability.
* Ethical Considerations: Intelligent garbage classification systems raise ethical concerns regarding data privacy and potential biases. Care must be taken to handle and secure data appropriately and mitigate biases that could disproportionately impact certain waste types or communities.

**8 CONCLUSIONS:**

conclusion, Intelligent Garbage Classification using Deep Learning offers significant

potential for revolutionizing waste management systems. By leveraging the power of artificial intelligence and deep learning algorithms, this technology can provide accurate and automated garbage classification, leading to several benefits and advancements.

The advantages of Intelligent Garbage Classification using Deep Learning include improved accuracy in classifying garbage items, automation, and increased efficiency in waste management processes. The scalability and adaptability of deep learning models enable handling large volumes of garbage items and accommodating new waste categories as they emerge. Additionally, resource optimization and a reduced environmental impact are achieved by identifying and recovering valuable recyclable materials.

However, there are also certain limitations and considerations to keep in mind. Deep learning models require substantial amounts of labeled data for training, and training time and complexity can be high. Overfitting and limited generalization to unseen scenarios can also be challenges that need to be addressed. Interpretability of deep learning models and ethical considerations related to data privacy and biases should be carefully considered.

Despite these challenges, Intelligent Garbage Classification using Deep Learning holds great promise in promoting sustainable waste management practices. It has the potential to streamline waste sorting processes, enhance recycling rates, and minimize environmental impact. By integrating this technology into waste management systems, we can work towards a cleaner and more sustainable future.

Continued research, innovation, and collaboration among experts in the field will be crucial in further improving the performance, efficiency, and ethical aspects of intelligent garbage classification systems using deep learning. Through these efforts, we can unlock the full potential of this technology and drive positive change in waste management practices worldwide.

**8 FUTURE SCOPE:**

Transfer Learning: Explore the use of pre-trained models and transfer learning to leverage existing knowledge for garbage classification tasks.

Multi-modal Classification: Extend the system to classify garbage based on multiple modalities, such as textual information or sensor data.

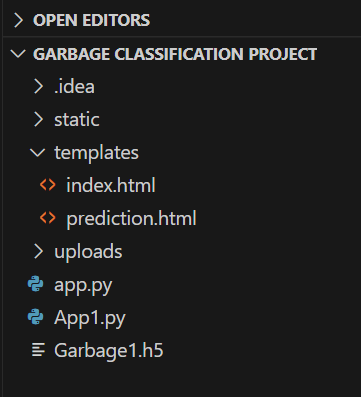
Continuous Learning: Implement mechanisms for continuous learning, allowing the system to adapt and improve over time as new waste types or patterns emerge.

Deployment in Smart Waste Management Systems: Integrate the intelligent garbage classification system with smart waste management infrastructure, such as automated sorting systems or waste collection robots.

**10 APPENDIXES:**

**10.1 Source code:**

**Project Structure:**

****

**Model Building:**

**#!/usr/bin/env python**

**# coding: utf-8**

**# In[1]:**

**from tensorflow.keras.preprocessing.image import ImageDataGenerator**

**# In[2]:**

**#setting parameter for image data augmentation to the training data.**

**train\_datagen = ImageDataGenerator(rescale=1./255,**

**shear\_range=0.1,**

**zoom\_range=0.1,**

**horizontal\_flip=True)**

**# In[3]:**

**#image data augmentation to the testing data.**

**val\_datagen = ImageDataGenerator(rescale = 1./255)**

**# In[8]:**

**train\_transform = train\_datagen.flow\_from\_directory(r"C:/Garbage Classification /Project/Train",**

**target\_size=(128,128),**

**batch\_size=64,**

**class\_mode='categorical')**

**# In[9]:**

**test\_transform = val\_datagen.flow\_from\_directory(r"** **C:/Garbage Classification Project/Test",**

**target\_size=(128,128),**

**batch\_size=64,**

**class\_mode='categorical')**

**# In[10]:**

**#to define linear initializations import Sequential**

**from tensorflow.keras.models import Sequential**

**#To add layers import Dense**

**from tensorflow.keras.layers import Dense**

**# to create o convolution kernel import Convolution2D**

**from tensorflow.keras.layers import Convolution2D**

**# Adding tdax pooling Layer**

**from tensorflow.keras.layers import MaxPooling2D**

**# Adding Flatten Layer**

**from tensorflow.keras.layers import Flatten**

**from tensorflow.keras.optimizers import Adam**

**# In[11]:**

**model=Sequential()**

**# In[13]:**

**#1st**

**model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu'))**

**model.add(MaxPooling2D(2,2))**

**# In[16]:**

**#2nd**

**model.add(Convolution2D(64,(3,3),padding='same',activation='relu'))**

**model.add(MaxPooling2D(pool\_size=2))**

**# In[17]:**

**#3rd**

**model.add(Convolution2D(32,(3,3),activation='relu'))**

**model.add(MaxPooling2D(2,2))**

**# In[18]:**

**#4rth**

**model.add(Convolution2D(32,(3,3),padding='same',activation='relu'))**

**model.add(MaxPooling2D(pool\_size=2))**

**# In[19]:**

**model.add(Flatten())**

**# In[20]:**

**# Adding 1st hidden layer**

**model.add(Dense(kernel\_initializer='uniform',activation='relu',units=150))**

**# In[21]:**

**# Adding 2nd hidden layer**

**model.add(Dense(kernel\_initializer='uniform',activation='relu',units=68))**

**# In[23]:**

**# Adding 3rd hidden layer**

**model.add(Dense(kernel\_initializer='uniform',activation='relu',units=6))**

**# In[24]:**

**model.summary()**

**# In[29]:**

**#Compile**

**model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['acc'])**

**# In[30]:**

**res = model.fit\_generator(train\_transform,steps\_per\_epoch=2022//64,**

**validation\_steps=505//64,**

**epochs=30,**

**validation\_data=test\_transform)**

**# In[32]:**

**model.save('Garbage1.h5')**

**# In[33]:**

**import numpy as np**

**from tensorflow.keras.models import load\_model**

**from tensorflow.keras.preprocessing import image**

**model = load\_model("Garbage1.h5")**

**# In[34]:**

**img = image.load\_img(r"** **C:/Garbage Classification /Project/Garbage classification/glass/glass33.jpg",**

**target\_size=(128,128))**

**x = image.img\_to\_array(img)**

**x = np.expand\_dims(x,axis=0)**

**# In[39]:**

**a = np.argmax(model.predict(x), axis=1)**

**# In[40]:**

**index=['0','1','2','3','4','5']**

**result = str(index[a[0]])**

**result**

**# In[41]:**

**train\_transform.class\_indices**

**# In[42]:**

**index1=['cardboard','glass','metal','paper','plastic','trash']**

**result1 = str(index1[a[0]])**

**result1**

**Python Code (app.py):**

import re

import sys

import numpy as np

import os

from flask import Flask, app, request, render\_template, url\_for, redirect

from tensorflow.keras import models

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

from tensorflow.python.ops.gen\_array\_ops import concat

# Loading the model

model = load\_model("C:/Users/91902/Downloads/garbage\_new.h5")

# model = load\_model("Garbage1.h5")

app = Flask(\_\_name\_\_)

# default home page or route

@app.route("/")

def index():

    return render\_template('index.html')

@app.route("/prediction.html")

def prediction():

    return render\_template('prediction.html')

@app.route("/index.html")

def home():

    return render\_template('index.html')

@app.route("/result", methods=["GET", "POST"])

def res():

    if request.method == "POST":

        f = request.files['image']

        basepath = os.path.dirname(\_\_file\_\_)

# getting the current path i.e where app.py is present

        # print("current path",basepath)

        filepath = os.path.join(basepath,'uploads',f.filename)  # from anywhere in the system we can give image but we want that image later  to process so we are saving it to uploads folder for reusing

        # print("upload folder is",filepath)

        f.save(filepath)

        img = image.load\_img(filepath, target\_size=(128, 128))

        x = image.img\_to\_array(img)  # img to array

        x = np.expand\_dims(x,axis=0)  # used for adding one more dimension

        # print(x)

        prediction = np.argmax(model.predict(x),axis=1)

# Instead of predict\_classes(x) we can use predict(X) ---->predict\_classes(x) gave error

        # print("prediction is ",prediction)

        index = ["cardboard","glass","metal","paper","plastic","trash"]

        result = str(index[prediction[0]])

        result

        return render\_template('prediction.html', prediction=result)

""" Running our application """

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, port=5000)

**Index Code (index.html):**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!--Bootstrap -->

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

    <script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

    <link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="../static/style.css">

    <!-- <script defer src="../static/js/main.js"></script> -->

    <title>Garbage Classification</title>

</head>

<body>

    <header id="head" class="header">

   <section  id="navbar">

           <h1 class="nav-heading"><i class="fas fa-recycle m2"></i>Garbage Classification</h1>

       <div class="nav--items">

           <ul>

               <li><a href="#about">About</a></li>

               <li><a href="#services">Services</a></li>

               <li><a href="#contact">Contact</a></li>

               <li><a href="prediction.html">Prediction</a></li>

           </ul>

       </div>

   </section>

   <section id="slider">

    <div id="carouselExampleIndicators" class="carousel" data-ride="carousel">

        <ol class="carousel-indicators ">

            <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active "></li>

            <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>

            <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>

            <li data-target="#carouselExampleIndicators" data-slide-to="3"></li>

            <li data-target="#carouselExampleIndicators" data-slide-to="4"></li>

        </ol>

        <div class="carousel-inner">

            <div class="carousel-caption d-none d-md-block">

                <h2 class="font">We Help You To Classify Garbage</h2>

                <p class="text-light">Reuse the past, Recycle the present, Save the future.</p>

            </div>

            <div class="carousel-item active">

                <img class="d-block w-100" src="../static/img/first.jpg" alt="First slide">

            </div>

            <div class="carousel-item">

                <img class="d-block w-100" src="../static/img/second.jpg" alt="Second slide">

            </div>

            <div class="carousel-item">

                <img class="d-block w-100" src="../static/img/third.jpg" alt="Third slide">

            </div>

            <div class="carousel-item">

                <img class="d-block w-100" src="../static/img/fourth.jpg" alt="fourth slide">

            </div>

            <div class="carousel-item">

                <img class="d-block w-100" src="../static/img/fifth.jpg" alt="fifth slide">

            </div>

        </div>

        <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">

            <span class="carousel-control-prev-icon" aria-hidden="true"></span>

            <span class="sr-only">Previous</span>

        </a>

        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">

            <span class="carousel-control-next-icon" aria-hidden="true"></span>

            <span class="sr-only">Next</span>

        </a>

    </div>

   </section>

</header>

<section id="about">

    <div class="top">

        <h3 class="title text-muted">

            ABOUT PROJECT

        </h3>

        <div class="line"></div>

    </div>

<div class="body">

<div class="left">

    <h2>Problem:</h2>

    <p>

        With the increase in the number of industries in the urban area, the disposal of solid waste is really becoming a big problem, and solid waste includes paper, wood, plastic, metal, glass, etc. The common way of managing waste is burning waste and this method can cause air pollution and some hazardous materials from the waste spread into the air which can cause cancer. Hence it is necessary to recycle the waste to protect the environment and human beings’ health, and we need to separate the waste into different components which can be recycled using different ways. The present way of separating waste/garbage is the hand-picking method, whereby someone is employed to separate out the different objects/materials. The person who separates waste is prone to diseases due to the harmful substances in the garbage. This problem can be overcome by automating the garbage classification process.

    </p>

</div>

<div class="right">

    <h2>Solution:</h2>

    <p>

        In this project, we will be building a deep learning model that can detect and classify types of garbage. A web application is integrated with the model, from where the user can upload a garbage image like paper waste, plastic waste, etc., and see the analyzed results on UserInterface.

    </p>

</div>

</div>

</section>

<section id="services">

<h3 class="title text-muted">WE CLASSIFY</h3>

<div class="line"></div>

<div class="testimonials">

    <div class="card" style="width: 25rem;">

        <img src="../static/img/testimonials/cardboard.jpg" class="card-img-top" alt="cardboard">

        <div class="card-body">

          <h5 class="card-title text-muted">CardBoard</h5>

          <p class="card-text">Cardboard, also referred to as corrugated cardboard, is a recyclable material that is recycled by small and large scale businesses to save money on waste disposal costs.</p>

        </div>

      </div>

    <div class="card" style="width: 25rem;">

        <img src="../static/img/testimonials/bottle.png" class="card-img-top" alt="bottle">

        <div class="card-body">

          <h5 class="card-title text-muted">Glass</h5>

          <p class="card-text">Glass is found in municipal solid waste (MSW), primarily in the form of containers such as beer and soft drink bottles; wine and liquor bottles; and bottles and jars for food, cosmetics and other products.</p>

        </div>

      </div>

    <div class="card" style="width: 25rem;">

        <img src="../static/img/testimonials/metal.jpg" class="card-img-top" alt="metal">

        <div class="card-body text-muted">

          <h5 class="card-title">Metal</h5>

          <p class="card-text">Metal waste/scrap waste can be subjected to the recycling process over and over again without changing its properties. Steel, for example, is one of the most recycled metals on the planet. Lorem ipsum dolor sit amet.</p>

        </div>

      </div>

    <div class="card" style="width: 25rem;">

        <img src="../static/img/testimonials/page.jpg" class="card-img-top" alt="page">

        <div class="card-body text-muted">

          <h5 class="card-title">Paper</h5>

          <p class="card-text">Paper Waste is a severe problem in many industries and offices. Because of printing mistakes, junk mails, billings, and packaging. Lorem ipsum dolor sit amet consectetur, adipisicing elit.</p>

        </div>

      </div>

    <div class="card" style="width: 25rem;">

        <img src="../static/img/testimonials/plastic.jpg" class="card-img-top" alt="plastic">

        <div class="card-body text-muted">

          <h5 class="card-title">Plastic</h5>

          <p class="card-text">Plastic waste, or plastic pollution, is the accumulation of plastic objects  in the Earth's environment that adversely affects wildlife, wildlife habitat, and humans.</p>

        </div>

      </div>

    <div class="card" style="width: 25rem;">

        <img src="../static/img/testimonials/trash.jpg" class="card-img-top" alt="trash">

        <div class="card-body text-muted">

          <h5 class="card-title">Trash</h5>

          <p class="card-text">Trash, rubbish, or refuse is waste material that is discarded by humans, usually due to a perceived lack of utility. Lorem ipsum dolor sit amet.</p>

        </div>

      </div>

</div>

</section>

<!-- Contact  -->

<section id="contact">

    <h3 class=" text-muted title">CONTACT US</h3>

    <div class="line"></div>

    <div class="contact-container">

    <div class="conatct-left">

        <div class="items">

            <i class="fas fa-map-pin fa-2x"></i>

            <h3 class=" text-muted">

                 Address

            </h3>

            <p>Nri institute of technology </p>

        </div>

        <div class="items">

            <i class="fas fa-envelope fa-2x"></i>

            <h3 class="text-muted">

                 Enquiries

            </h3>

            <p>pamidieswar8@gmail.com </p>

        </div>

        <div class="items">

            <i class="fas fa-phone fa-2x"></i>

            <h3 class=" text-muted">

                 Call Us

            </h3>

            <p>+91 8897721907</p>

        </div>

    </div>

    <div class="contact-right">

        <div class="container noselect">

            <div class="canvas">

              <div class="tracker tr-1"></div>

              <div class="tracker tr-2"></div>

              <div class="tracker tr-3"></div>

              <div class="tracker tr-4"></div>

              <div class="tracker tr-5"></div>

              <div class="tracker tr-6"></div>

              <div class="tracker tr-7"></div>

              <div class="tracker tr-8"></div>

              <div class="tracker tr-9"></div>

              <div class="tracker tr-10"></div>

              <div class="tracker tr-11"></div>

              <div class="tracker tr-12"></div>

              <div class="tracker tr-13"></div>

              <div class="tracker tr-14"></div>

              <div class="tracker tr-15"></div>

              <div class="tracker tr-16"></div>

              <div class="tracker tr-17"></div>

              <div class="tracker tr-18"></div>

              <div class="tracker tr-19"></div>

              <div class="tracker tr-20"></div>

              <div class="tracker tr-21"></div>

              <div class="tracker tr-22"></div>

              <div class="tracker tr-23"></div>

              <div class="tracker tr-24"></div>

              <div class="tracker tr-25"></div>

              <div id="card">

              <p id="prompt">Reuse the past,<br> Recycle the present,<br> Save the future</p>

                <div class="title">A Project Done by :<br>eswar</div>

                <div class="subtitle">

                  Smart bridge

                </div>

              </div>

            </div>

          </div>

          <style>

            /\*works janky on mobile :<\*/

.container {

  position: relative;

  width: 190px;

  height: 254px;

  transition: 200ms;

}

.container:active {

  width: 180px;

  height: 245px;

}

#card {

  position: absolute;

  inset: 0;

  z-index: 0;

  display: flex;

  justify-content: center;

  align-items: center;

  border-radius: 20px;

  transition: 700ms;

  background: linear-gradient(43deg, rgb(65, 88, 208) 0%, rgb(200, 80, 192) 46%, rgb(255, 204, 112) 100%);

}

.subtitle {

  transform: translateY(160px);

  color: rgb(134, 110, 221);

  text-align: center;

  width: 100%;

}

.title {

  opacity: 0;

  transition-duration: 300ms;

  transition-timing-function: ease-in-out-out;

  transition-delay: 100ms;

  position: absolute;

  font-size: x-large;

  font-weight: bold;

}

.tracker:hover ~ #card .title {

  opacity: 1;

}

#prompt {

  bottom: 8px;

  left: 12px;

  z-index: 20;

  font-size: 20px;

  font-weight: bold;

  transition: 300ms ease-in-out-out;

  position: absolute;

  max-width: 110px;

  color: rgb(255, 255, 255);

}

.tracker {

  position: absolute;

  z-index: 200;

  width: 100%;

  height: 100%;

}

.tracker:hover {

  cursor: pointer;

}

.tracker:hover ~ #card #prompt {

  opacity: 0;

}

.tracker:hover ~ #card {

  transition: 300ms;

  filter: brightness(1.1);

}

.container:hover #card::before {

  transition: 200ms;

  content: '';

  opacity: 80%;

}

.canvas {

  perspective: 800px;

  inset: 0;

  z-index: 200;

  position: absolute;

  display: grid;

  grid-template-columns: 1fr 1fr 1fr 1fr 1fr;

  grid-template-rows: 1fr 1fr 1fr 1fr 1fr;

  gap: 0px 0px;

  grid-template-areas: "tr-1 tr-2 tr-3 tr-4 tr-5"

    "tr-6 tr-7 tr-8 tr-9 tr-10"

    "tr-11 tr-12 tr-13 tr-14 tr-15"

    "tr-16 tr-17 tr-18 tr-19 tr-20"

    "tr-21 tr-22 tr-23 tr-24 tr-25";

}

#card::before {

  content: '';

  background: linear-gradient(43deg, rgb(65, 88, 208) 0%, rgb(200, 80, 192) 46%, rgb(255, 204, 112) 100%);

  filter: blur(2rem);

  opacity: 30%;

  width: 100%;

  height: 100%;

  position: absolute;

  z-index: -1;

  transition: 200ms;

}

.tr-1 {

  grid-area: tr-1;

}

.tr-2 {

  grid-area: tr-2;

}

.tr-3 {

  grid-area: tr-3;

}

.tr-4 {

  grid-area: tr-4;

}

.tr-5 {

  grid-area: tr-5;

}

.tr-6 {

  grid-area: tr-6;

}

.tr-7 {

  grid-area: tr-7;

}

.tr-8 {

  grid-area: tr-8;

}

.tr-9 {

  grid-area: tr-9;

}

.tr-10 {

  grid-area: tr-10;

}

.tr-11 {

  grid-area: tr-11;

}

.tr-12 {

  grid-area: tr-12;

}

.tr-13 {

  grid-area: tr-13;

}

.tr-14 {

  grid-area: tr-14;

}

.tr-15 {

  grid-area: tr-15;

}

.tr-16 {

  grid-area: tr-16;

}

.tr-17 {

  grid-area: tr-17;

}

.tr-18 {

  grid-area: tr-18;

}

.tr-19 {

  grid-area: tr-19;

}

.tr-20 {

  grid-area: tr-20;

}

.tr-21 {

  grid-area: tr-21;

}

.tr-22 {

  grid-area: tr-22;

}

.tr-23 {

  grid-area: tr-23;

}

.tr-24 {

  grid-area: tr-24;

}

.tr-25 {

  grid-area: tr-25;

}

.tr-1:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(20deg) rotateY(-10deg) rotateZ(0deg);

}

.tr-2:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(20deg) rotateY(-5deg) rotateZ(0deg);

}

.tr-3:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(20deg) rotateY(0deg) rotateZ(0deg);

}

.tr-4:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(20deg) rotateY(5deg) rotateZ(0deg);

}

.tr-5:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(20deg) rotateY(10deg) rotateZ(0deg);

}

.tr-6:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(10deg) rotateY(-10deg) rotateZ(0deg);

}

.tr-7:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(10deg) rotateY(-5deg) rotateZ(0deg);

}

.tr-8:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(10deg) rotateY(0deg) rotateZ(0deg);

}

.tr-9:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(10deg) rotateY(5deg) rotateZ(0deg);

}

.tr-10:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(10deg) rotateY(10deg) rotateZ(0deg);

}

.tr-11:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(0deg) rotateY(-10deg) rotateZ(0deg);

}

.tr-12:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(0deg) rotateY(-5deg) rotateZ(0deg);

}

.tr-13:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(0deg) rotateY(0deg) rotateZ(0deg);

}

.tr-14:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(0deg) rotateY(5deg) rotateZ(0deg);

}

.tr-15:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(0deg) rotateY(10deg) rotateZ(0deg);

}

.tr-16:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-10deg) rotateY(-10deg) rotateZ(0deg);

}

.tr-17:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-10deg) rotateY(-5deg) rotateZ(0deg);

}

.tr-18:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-10deg) rotateY(0deg) rotateZ(0deg);

}

.tr-19:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-10deg) rotateY(5deg) rotateZ(0deg);

}

.tr-20:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-10deg) rotateY(10deg) rotateZ(0deg);

}

.tr-21:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-20deg) rotateY(-10deg) rotateZ(0deg);

}

.tr-22:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-20deg) rotateY(-5deg) rotateZ(0deg);

}

.tr-23:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-20deg) rotateY(0deg) rotateZ(0deg);

}

.tr-24:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-20deg) rotateY(5deg) rotateZ(0deg);

}

.tr-25:hover ~ #card {

  transition: 125ms ease-in-out;

  transform: rotateX(-20deg) rotateY(10deg) rotateZ(0deg);

}

.noselect {

  -webkit-touch-callout: none;

   /\* iOS Safari \*/

  -webkit-user-select: none;

   /\* Safari \*/

   /\* Konqueror HTML \*/

  -moz-user-select: none;

   /\* Old versions of Firefox \*/

  -ms-user-select: none;

   /\* Internet Explorer/Edge \*/

  user-select: none;

   /\* Non-prefixed version, currently

                  supported by Chrome, Edge, Opera and Firefox \*/

}

          </style>

    </div>

</div>

</section>

<section id="footer">

    <p>Copyright © 2021. All Rights Reserved</p>

</section>

</body>

</html>

**Prediction Code (prediction.html):**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!--Bootstrap -->

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

    <script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

    <link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="../static/style.css">

    <script defer src="../static/js/JScript.js"></script>

    <title>Prediction</title>

</head>

<body>

    <header id="head" class="header">

        <section  id="navbar">

                <h1 class="nav-heading"><i class="fas fa-recycle m2"></i>Garbage Classification</h1>

            <div class="nav--items">

                <ul>

                    <li><a href="index.html#about">About</a></li>

                    <li><a href="index.html#services">Services</a></li>

                    <li><a href="index.html#contact">Contact</a></li>

                    <li><a href="prediction.html">Prediction</a></li>

                </ul>

            </div>

        </section>

    </header>

    <!-- dataset/Training/metal/metal326.jpg -->

    <section id="prediction">

        <div class="prediction-input">

        <div class="circle">

            <img src="../static/img/dummy/metal326.jpg" alt="Demo" id="demo" class="circle">

        </div>

                <form id="form" action="/result" method="post"  enctype="multipart/form-data">

                    <input type="file" id="imageupload"  name="image" accept="image/\*" class="input-image">

                    <input type="submit" class="submitbtn">

                  </form>

            </div>

            <h3 class="title text-muted">

            THE PREDICTION IS

          </h3>

          <div class="line"></div>

              <div class="output-container">

                  <div data-type="cardboard" class="output img1">

                    <img src="../static/img/testimonials/cardboard.jpg" alt="" class="circle">

                    <h3 class="text-muted">CARDBOARD</h3>

                </div>

                  <div data-type="glass" class="output img2">

                    <img src="../static/img/testimonials/bottle.png" alt="" class="circle">

                    <h3 class="text-muted">GLASS</h3>

                </div>

                  <div data-type="metal" class="output img3">

                    <img src="../static/img/testimonials/metal.jpg" alt="" class="circle">

                    <h3 class="text-muted">METAL</h3>

                </div>

                  <div data-type="paper" class="output img4">

                    <img src="../static/img/testimonials/page.jpg" alt="" class="circle">

                    <h3 class="text-muted">PAPER</h3>

                </div>

                  <div data-type="plastic" class="output img5">

                    <img src="../static/img/testimonials/plastic.jpg" alt="" class="circle">

                    <h3 class="text-muted">PLASTIC</h3>

                </div>

                  <div data-type="trash" class="output img6">

                    <img src="../static/img/testimonials/trash.jpg" alt="" class="circle">

                    <h3 class="text-muted">TRASH</h3>

                </div>

              </div>

              <div class="hide" id="result">

                {{prediction}}

              </div>

    </section>

    <section id="footer">

        <p>Copyright © 2021. All Rights Reserved</p>

    </section>

</body>

</html>