A INTELLIGENT GARBAGE CLASSIFICATION USING DEEP LEARNING

IBM PROJECT REPORT

Submitted by

TAMILSELVAM. S (Team leader)	**[513220205006]**
INDHUMATHI. K	**[513220205001]**
MALINI. A	**[513220205004]**
DEENESHWARAN .M	**[513220205303]**

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

THIRUMALAI ENGINEERING COLLEGE, KANCHIPURAM

ANNA UNIVERSITY: CHENNAI – 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certificate that this project report titled **"A INTELLIGENT GARBAGE CLASSIFICATION USING DEEP LEARNING"** is the Bonafide work of **"TAMILSELVAM. S [513220205006], INDHUMATHI. K [513220205001], MADHAN. K [513220205003], DEENESHWARAN .M [513220205303]"** who carried out the project work under my supervision.

Guide and Head of the Department

Mrs. V. Hemalatha M.E.,

HEAD OF THE DEPARTMENT

INFORMATION TECHNOLOGY

THIRUMALAI ENGINEERING COLLEGE

KILAMBI, KANCHIPURAM – 631502

ACKNOWLEDGEMENT

I profoundly thank our **Chairman and trust members of Kanchipuram Educational Trust** for providing adequate facilities.

I would like to express my hearty thanks to our respectable Principal. In charge **Mr. T. Mohanraj M.Tech.,** for allowing us to have the extensive use of our college's facilities to our colleges facilities to have precious advice regarding the project.

I extend our thanks to Assistant Professor **DR.B. YUVARAJ M.E., Ph.D., Head of the Department, Information Technology** for this precious advice regarding the project.

I would like to express my deep and unbounded gratefulness to my project Guide **Mrs. Hemalatha M.E.,** Department of Information Technology, for his valuable guidance and encouragement throughout the project. He has been a constant source of inspiration and has provided the precious suggestion throughout this project.

I thank all facilities and supporting staff for the help they extended in completing this project. I also express my sincere thanks to our parents, and all my friends for their continuous support.

TABLE OF CONTENTS

ABSTRACT

Garbage classification is an important environmental issue that aims to reduce the amount of waste that ends up in landfills or the environment. Deep learning has become a popular tool for tackling this problem by automating the classification process.

One approach to using deep learning for garbage classification is to train a convolutional neural network (CNN) on a large dataset of labelled images of different types of garbage. The CNN can learn to extract relevant features from the images and use them to classify the garbage into different categories, such as organic, recyclable, and hazardous.

To improve the accuracy of the classification, data augmentation techniques can be used to generate additional training examples by applying random transformations to the images, such as cropping, flipping, and rotation. Transfer learning can also be employed, where a pre-trained CNN model is fine-tuned on the garbage classification task to leverage its learned features and reduce training time.

One of the challenges of garbage classification using deep learning is the need for large amounts of labelled data. This can be addressed by crowdsourcing the labelling task or by using semi-supervised or unsupervised learning methods.

Overall, garbage classification using deep learning has the potential to greatly improve the efficiency and accuracy of waste management systems, leading to a cleaner and more sustainable environment.

1.INTRODUCTION

1.1 Project Overview

According to the next 25 years, the less developed countries' waste accumulation will increase drastically. With the increase in the number of industries in the urban area, the disposal of the solid waste is really becoming a big problem, and the solid waste includes paper, wood, plastic, metal, glass etc.

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. With this in mind, it motivated us to develop an automated system which is able to sort the waste. And this system can take a short time to sort the waste, and it will be more accurate in sorting than the manual way.

With the system in place, the beneficial separated waste can still be recycled and converted to energy and fuel for the growth of the economy. The system that is developed for the separation of the accumulated waste is based on the combination of CNN.

1.2 Purpose

The purpose of a garbage classification project using deep learning is to develop an automated system that can accurately and efficiently categorize different types of waste or garbage based on their attributes. The primary objective is to improve waste management processes by enabling effective sorting and recycling of garbage.

Here are some specific purposes and benefits of garbage classification using deep learning:

1.Efficient waste management: Deep learning algorithms can process large amounts of data and make quick decisions, allowing for the automated and rapid sorting of garbage. This streamlines waste management processes and reduces the time and effort required for manual sorting.

2.Recycling optimization: By accurately classifying garbage into different categories such as plastic, paper, glass, metal, or organic waste, deep learning models help optimize recycling processes. This ensures that recyclable materials are properly sorted and sent for recycling, minimizing waste and promoting sustainable practices.

3.Reduction of landfill waste: Proper garbage classification helps identify materials that can be diverted from landfills. By separating recyclable and compostable waste from non-recyclable and non-biodegradable materials, deep learning models contribute to the reduction of landfill waste, conserving valuable landfill space and mitigating environmental impact.

4.Environmental conservation: Effective garbage classification supports environmental conservation efforts by encouraging the proper handling and disposal of waste. Deep learning models can help raise awareness about the importance of waste management and promote responsible behaviour among individuals and communities.

5.Resource recovery: Deep learning-based garbage classification systems can identify valuable materials within waste streams, such as precious metals or reusable components. By extracting and recovering these resources, the project contributes to the circular economy by reducing the demand for virgin materials and minimizing resource depletion.

6.Scalability and automation: Deep learning models can be deployed across various waste management facilities, including recycling centres, waste sorting plants, and landfill sites. This scalability and automation potential enable consistent and reliable garbage classification, regardless of the scale of waste generation.

Overall, the purpose of garbage classification using deep learning projects is to leverage the power of artificial intelligence to improve waste management practices, enhance recycling efforts, minimize environmental impact, and promote sustainable resource utilization.

2.IDEATION & PROPOSED SOLUTION

2.1Problem Statement Definition

The problem is to develop an efficient and accurate garbage classification system using deep learning techniques. The aim is to automatically categorize different types of waste materials into appropriate classes such as organic, recyclable, non-recyclable, and hazardous.

Currently, waste management and recycling processes heavily rely on manual sorting, which is time-consuming, costly, and prone to human error. A more automated and reliable solution is required to streamline the waste management process, improve recycling rates, and reduce the environmental impact of improper waste disposal.

The challenge lies in accurately identifying and classifying various types of garbage based on their visual attributes. This involves overcoming the complexities of different shapes, colours, textures, and sizes of waste items. Additionally, the system should be robust enough to handle real-world scenarios, where images may contain multiple garbage items, occlusions, and variations in lighting conditions.

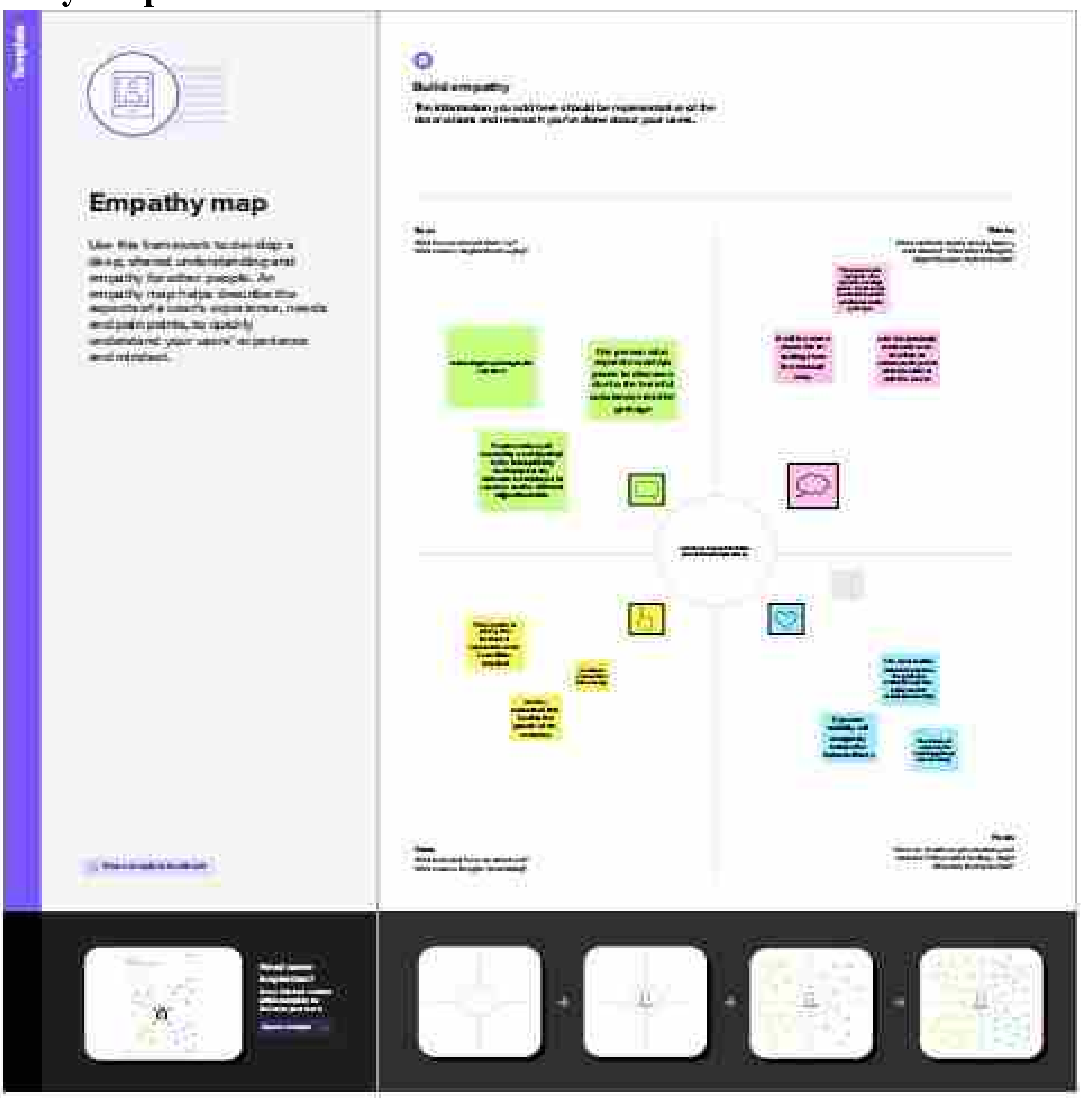
The solution to this problem involves leveraging deep learning algorithms, such as convolutional neural networks (CNNs), to automatically learn and extract meaningful features from garbage images. The model will be trained on a large dataset of labelled garbage images, enabling it to recognize patterns and make accurate predictions. The trained model will then be deployed as a practical tool that can classify garbage in real-time, either through a web or mobile application, or through integration with existing waste management systems.

The successful implementation of this garbage classification system will not only enhance the efficiency and effectiveness of waste management processes but also promote sustainable practices by encouraging proper recycling and disposal methods.

Problem Statement

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes feel
PS-1	alum	Always clean my street	every day increase number of plastic wastage's	Increase the people population	Increase diseases
PS-2	Manila	Clean India	Every day increase car boards, trash, papers, plastics and metal, glass.	People carelessness	The smart dustbin will in turn improve the garbage collection system implemented across the country.

2.2Empathy Map Canvas



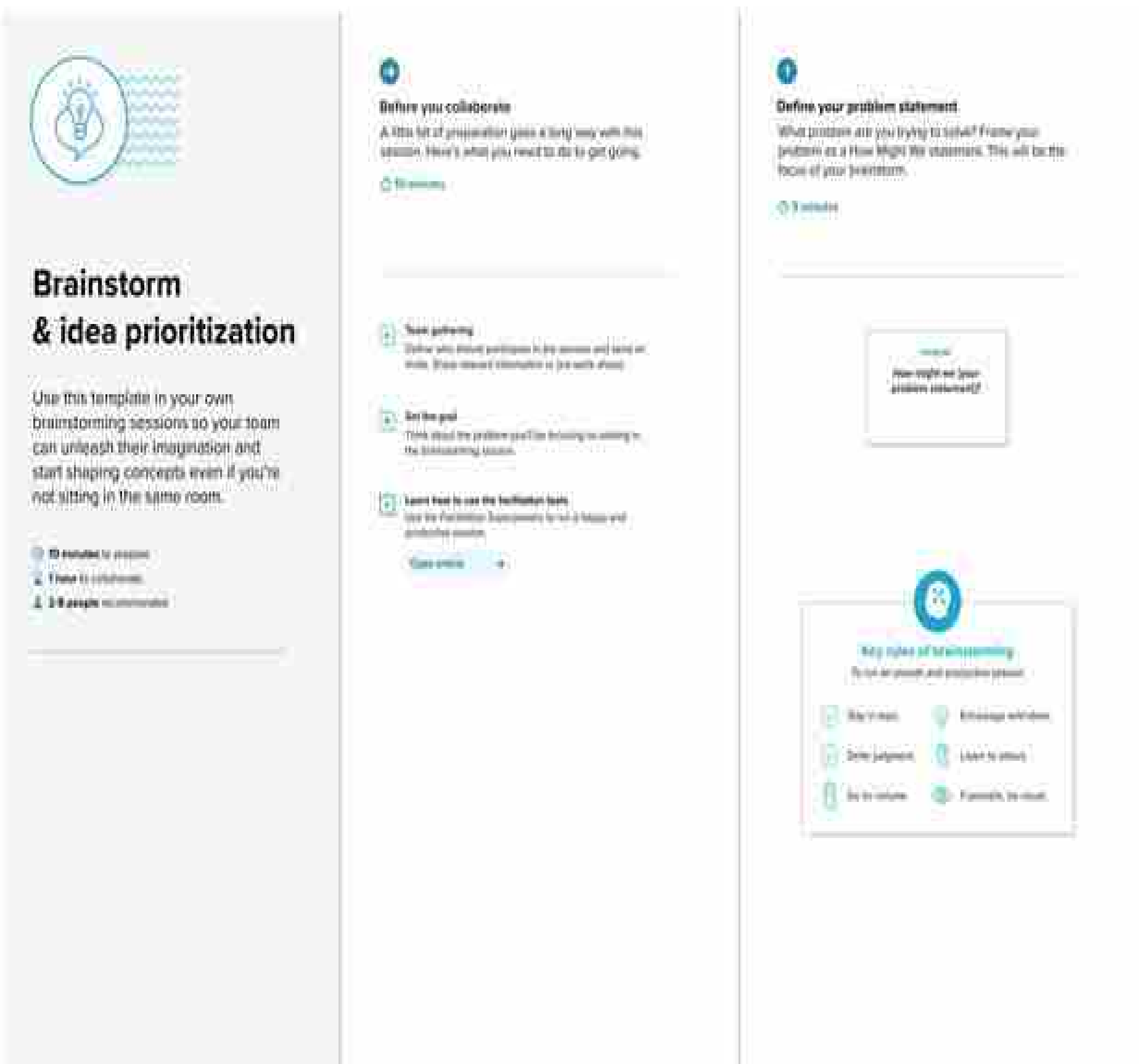
Empathy Map

EXAMPLE:INTELLIGENCE GARBAGE CLASSIFICATION USING DEEP LEARNING.

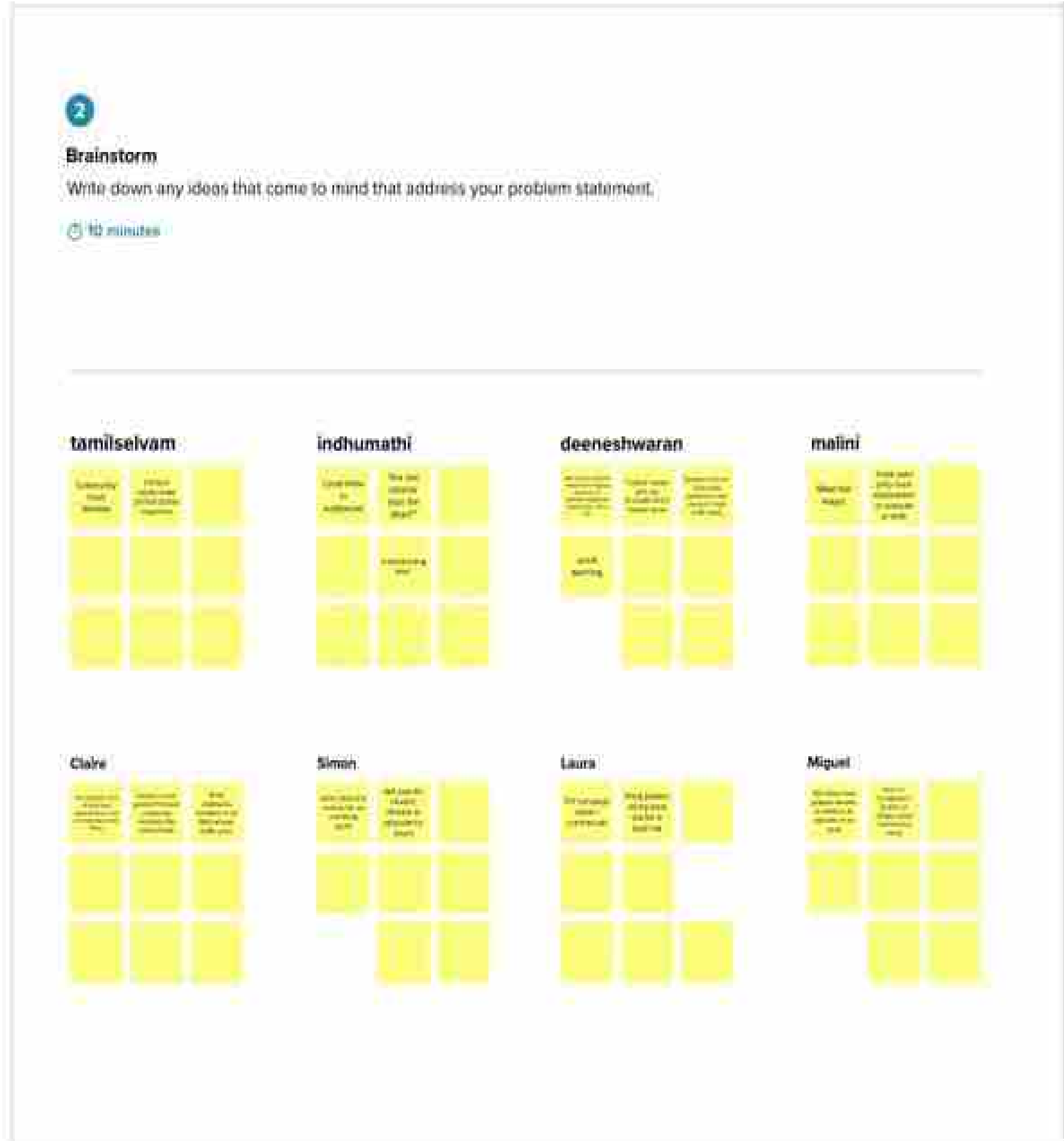
2.3Ideation & Brainstorming

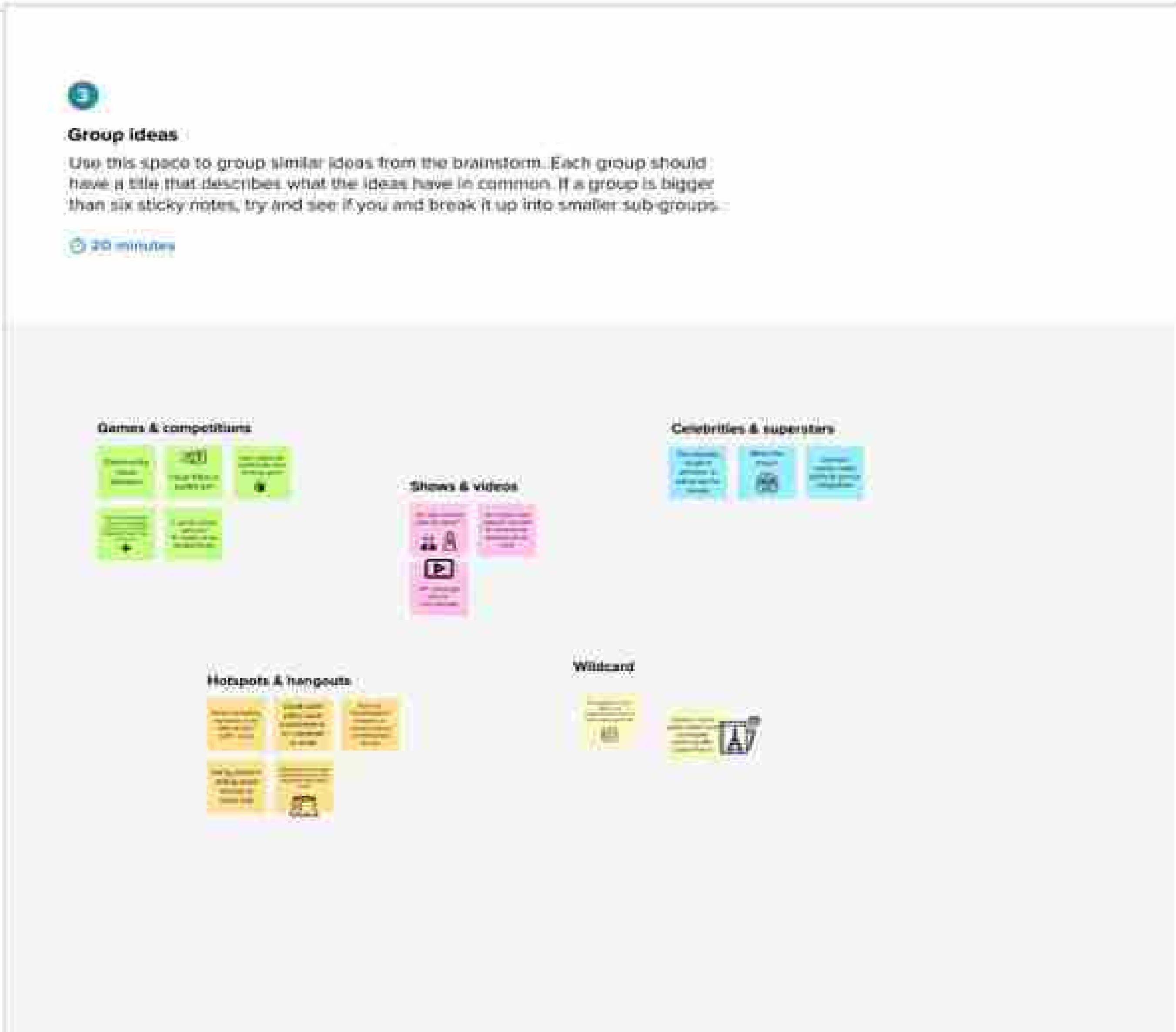
Brainstorm & Idea Prioritization

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

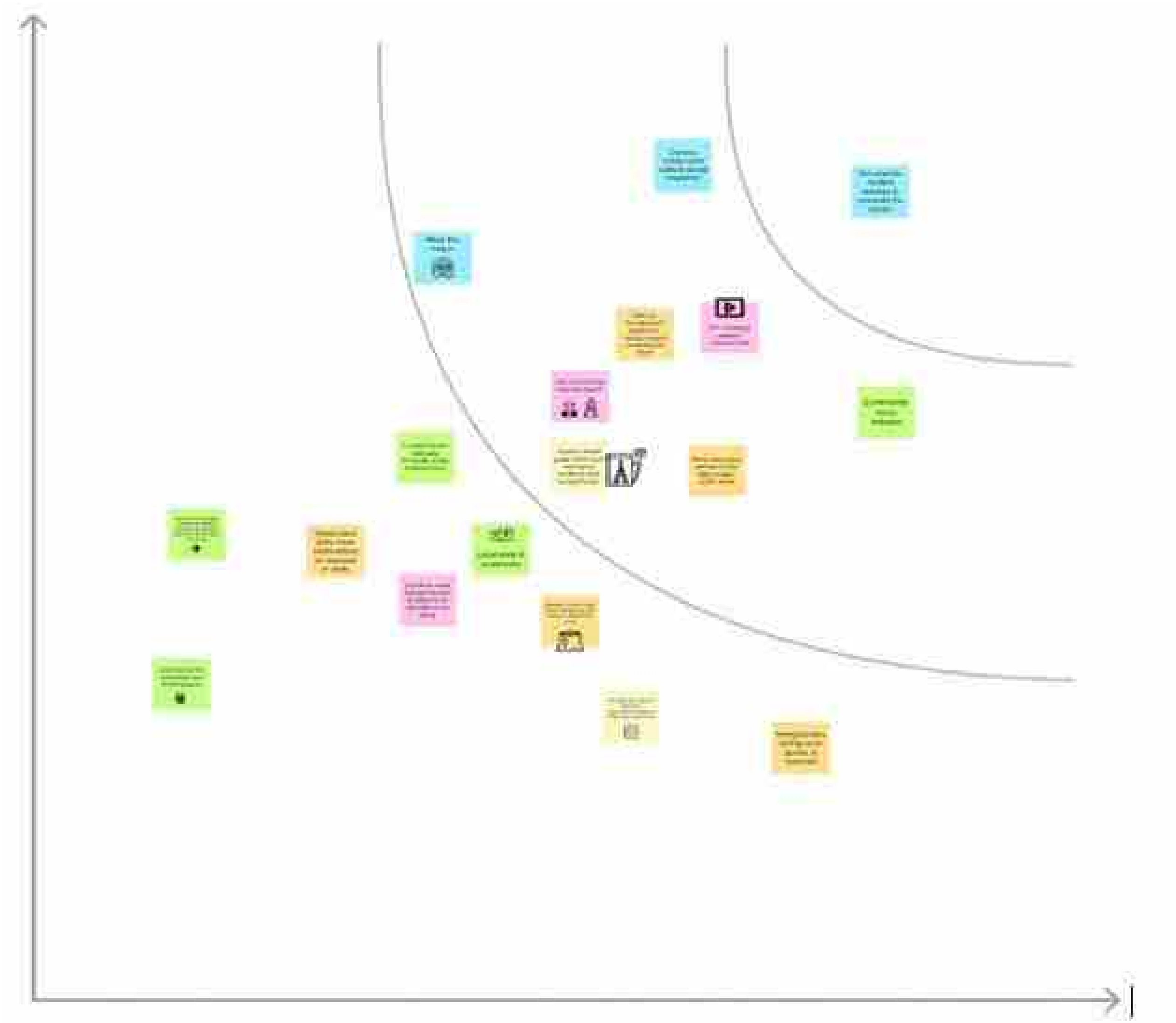


Step-2: Brainstorm, Idea Listing and Grouping




3
Group ideas
Use this space to group similar ideas from the brainstorm. Each group should have a title that describes what the ideas have in common. If a group is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Games & competitions
Shows & videos
Celebrities & superstars
Hotspots & hangouts
Wildcard

Step-3: Idea Prioritization



2.4 Proposed Solution

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	People not segregating the garbage in spite of awareness campaigns.
2.	Idea / Solution description	Increase the effectiveness of awareness campaigns
3.	Novelty / Uniqueness	By gamifying waste segregation, we can make it more engaging, and rewarding, and thereby encourage more people to participate in this important activity.
4.	Social Impact / Customer Satisfaction	1. Health risks 2. Aesthetic problems 3. Limited resource recovery 4. Increased costs
5.	Business Model (Revenue Model)	waste management and recycling company
6.	Scalability of the Solution	1. Replication 2. Public-Private Partnerships 3. Technology

3. REQUIREMENT ANALYSIS

3.1 Functional Requirement

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 Confirmation via OTP
FR-3	User Interface	It allows users to capture images of garbage and see the results of the classification in real-time.
FR-4	AI Model	The project should use an AI algorithm that can learn from data and improve over time.
FR-5	Real-time Classification	It should be able to classify images quickly and accurately as soon as they are captured by a camera.
FR-6	Feedback Loop	This feedback can be used to improve the accuracy of the model and the user interface.

3.2 Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface should be easy to use and understand, even for non-technical users. The system should have clear instructions and a user-friendly design.
NFR-2	**Security**	The project should have secure data storage and processing to ensure the privacy and confidentiality of the garbage images.
NFR-3	**Reliability**	The project should be reliable and available for use at all times. The system should be designed to minimize downtime or errors.
NFR-4	**Performance**	The project should be designed to perform efficiently and effectively.
NFR-5	**Availability**	1. 24/7 Availability 2. Monitoring and Alerts 3. Fault Tolerance 4. Cloud-Based Deployment
NFR-6	**Scalability**	The project should be scalable to handle large amounts of data and users. The system should be able to handle an number of garbage images.

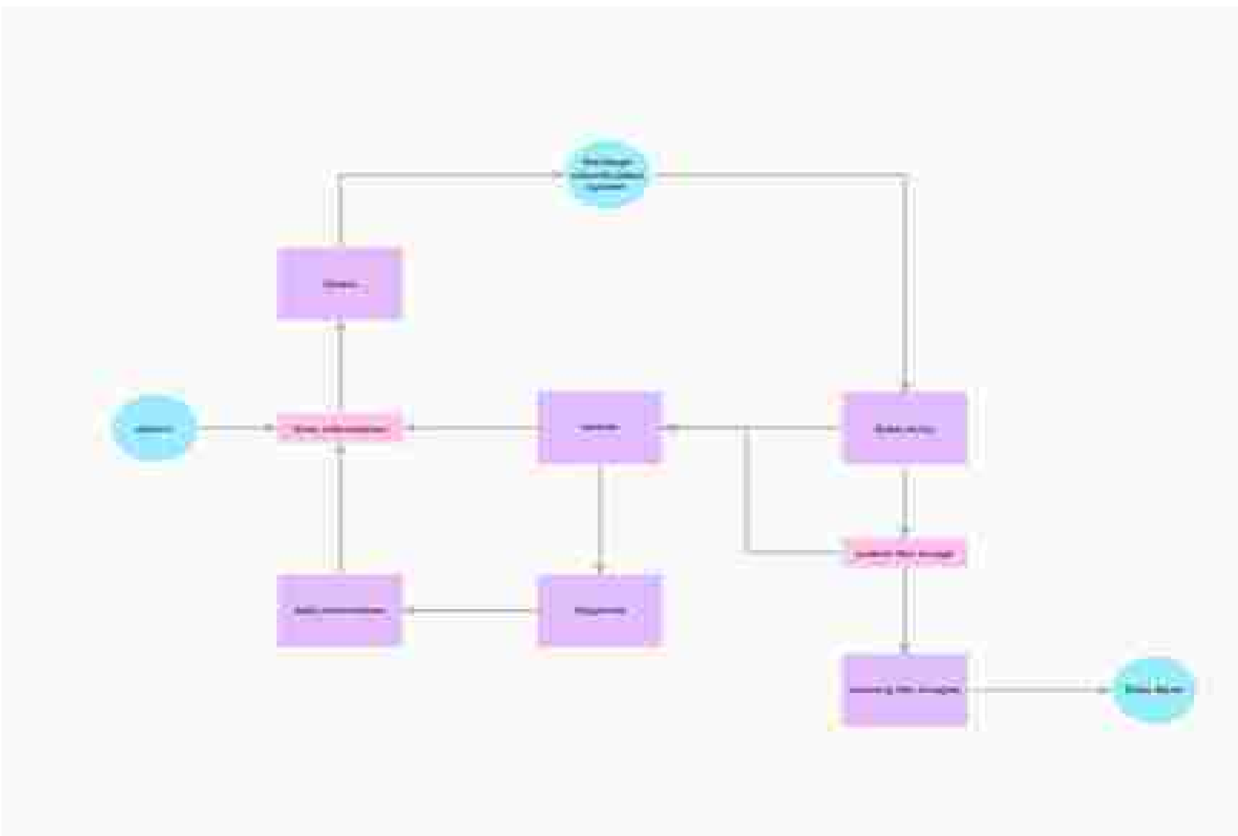
4.Project Design

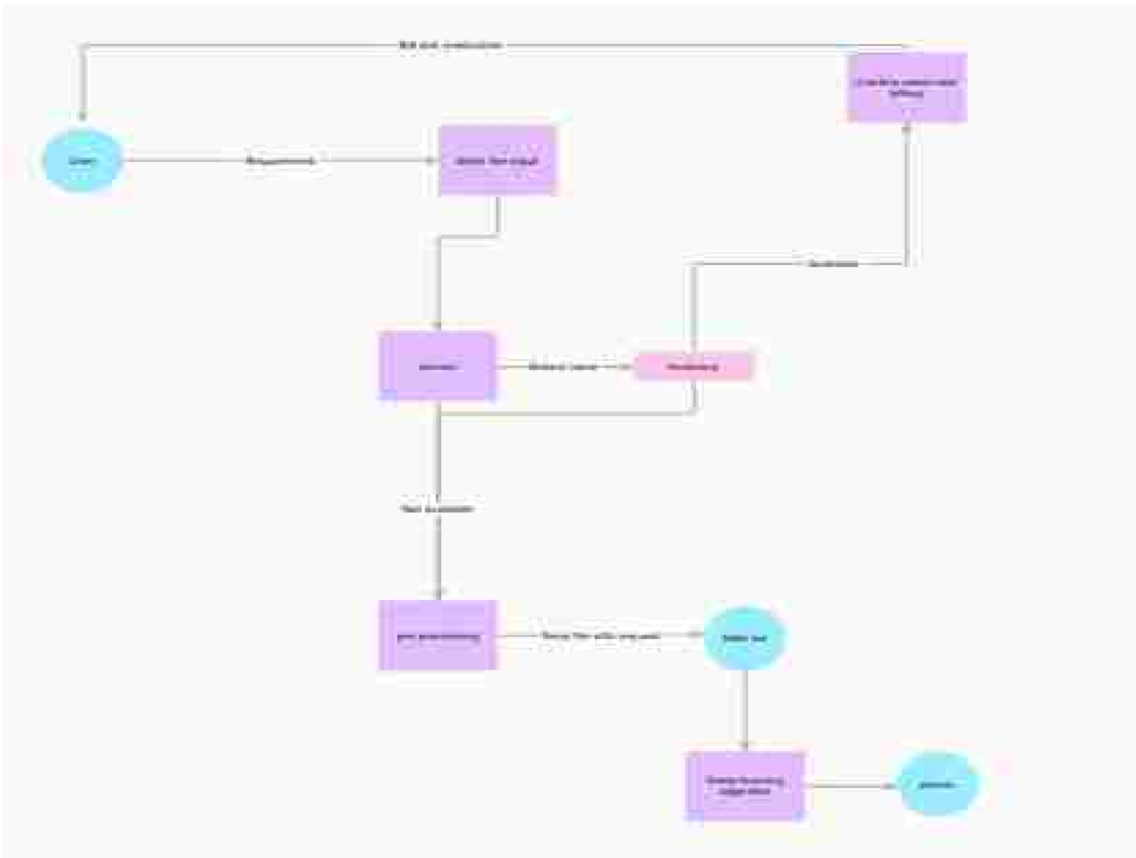
4.1 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

DFD Diagram for intelligent garbage classification using deep learning

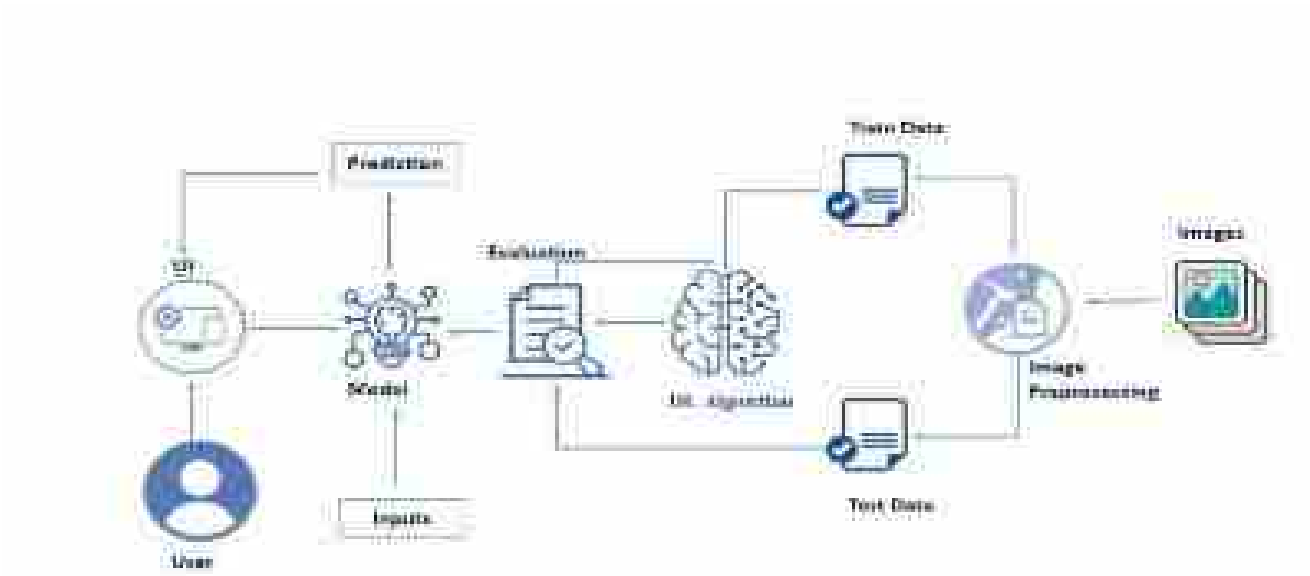
DFD Level 0 and 1 diagram





4.2 Solution & Technical Architecture

Solution Architecture Diagram



The architecture diagram for a garbage classification project using deep learning typically involves several components and steps. Here's an overview of the typical architecture:

1.Input Data: The input data consists of a large dataset of garbage images. These images are captured using cameras or collected from various sources. The dataset should be diverse and representative of different types of garbage items.

2.Preprocessing: The pre-processing step involves preparing the input data for training the deep learning model. This may include tasks such as resizing the images to a consistent size, normalizing pixel values, and applying data augmentation techniques like rotation, flipping, or adding noise to increase the diversity of the training data.

3.Convolutional Neural Network (CNN): The core component of the architecture is the CNN. CNNs are highly effective in image classification tasks due to their ability to automatically learn hierarchical features from the input images. The CNN architecture typically consists of multiple convolutional layers, pooling layers, and fully connected layers.

4.Training: The pre-processed data is used to train the CNN. During training, the CNN learns to recognize and extract relevant features from the garbage images by minimizing a defined loss function. The training process involves forward propagation, backpropagation, and updating the network parameters using optimization algorithms like stochastic gradient descent (SGD) or adaptive methods like Adam.

5.Evaluation: Once the CNN is trained, it needs to be evaluated to assess its performance. This is done using a separate validation dataset that was not used during training. The evaluation metrics, such as accuracy, precision, recall, and F1 score, are computed to measure the model's performance.

6.Continuous Improvement: The garbage classification system can be continuously improved by periodically retraining the model on new data. This ensures that the model stays up-to-date with new garbage items and maintains high accuracy over time.

It's important to note that the architecture diagram may vary depending on the specific implementation and requirements of the garbage classification project. Different CNN architectures, such as VGG, ResNet, or EfficientNet, can be utilized, and additional techniques like transfer learning or assembling can be employed to enhance the model's performance.

4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Malini
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Tamilselvam
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Indhumathi
		USN-4	As a user, I can register for the application through Gmail		Medium	Deeneshwar an
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Deeneshwar an
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator	Manage the product	USN-6	As a developer, I want to build a user-friendly web application that allows users to upload images of their garbage and receive feedback on how to sort it properly.	we can increase public awareness of the importance of proper waste sorting and reduce contamination in recycling streams.	High	Tamilselvam
Homeowner	Sign in	USN-7	As a homeowner, I want to be able to sort my garbage more effectively.	I can reduce my environmental impact and contribute to a healthier planet.	Low	Indhumathi
Municipal Recycling	Login	USN-8	As a municipal recycling center employee, I want to be able to quickly and accurately sort incoming waste materials by type.	we can increase our recycling rates and reduce	High	Malini
				the amount of waste we send to landfill.		
Waste Management	Dashboard	USN-8	As a waste management professional, I want to be able to automatically sort different types of waste using computer vision.	I can improve the efficiency of our operations and reduce the amount of waste that ends up in landfills.	High	Tamilselvam

5. CODING & SOLUTION (Explain the features added in the project along with code)

5.1 Feature 1

Classification based on CNN

Garbage classification project using deep learning, specifically Convolutional Neural Networks (CNNs). This example assumes you have a dataset of garbage images categorized into different classes such as plastic, paper, glass, and metal.

Python Code

```
# Importing the required libraries

import os

import numpy as np

from PIL import Image

from sklearn.model_selection import train_test_split

from keras.utils import to_categorical

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Define the path to the dataset

dataset_path = "path/to/dataset/folder"

# Define the list of classes

classes = ['plastic', 'paper', 'glass', 'metal']

# Function to load and preprocess the dataset

def load_dataset():
```

```
 # Perform one-hot encoding on the labels
  labels = to_categorical(labels)
 # Split the dataset into training and testing sets
   X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
   return X_train, X_test, y_train, y_test
 # Load the dataset
 X_train, X_test, y_train, y_test = load_dataset()
 # Build the CNN model
 model = Sequential()
 model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
 model.add(MaxPooling2D((2, 2)))
# Compile the model
 model.compile(optimizer='adam',loss='categorical_crossentropy', metrics=['accuracy'])
# Train the model
 model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
# Evaluate the model
 loss, accuracy = model.evaluate(X_test, y_test)
 print("Test loss:", loss)
 print("Test accuracy:", accuracy)
```

5.2 Feature 2

Data Augmentation

Data augmentation is a technique that artificially increases the size of your dataset by applying various transformations to the existing images, such as rotation, scaling, and flipping. This can help improve the model's performance and generalization ability.

Python Code

```
# Importing the required libraries

import os

import numpy as np

from PIL import Image

from sklearn.model_selection import train_test_split

from keras.utils import to_categorical

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

from keras.preprocessing.image import ImageDataGenerator

# Define the path to the dataset

dataset_path = "path/to/dataset/folder"

# Define the list of classes

classes = ['plastic', 'paper', 'glass', 'metal']

# Function to load and preprocess the dataset

def load_dataset():
```

```
    # Perform one-hot encoding on the labels
    labels = to_categorical(labels)
    # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
    return X_train, X_test, y_train, y_test
# Load the dataset
X_train, X_test, y_train, y_test = load_dataset()
# Apply data augmentation to the training set
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
# Build the CNN model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
```

```
model.add(MaxPooling2D((2, 2)))
# Compile the model
model.compile(optimizer='adam',loss='categorical_crossentropy', metrics=['accuracy'])
# Train the model with data augmentation
model.fit(datagen.flow(X_train, y_train, batch_size=32),
steps_per_epoch=len(X_train) / 32,
# Adjust the steps per epoch based on your dataset size
        epochs=10,
        validation_data=(X_test, y_test))
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print("Test loss:", loss)
print("Test accuracy:", accuracy)
```

5.3 Database Schema (if Applicable)

Table: Images

id (Primary Key): Unique identifier for each image

path: File path to the image

label: Label or class of the image (e.g., plastic, paper, glass, metal)

created at: Timestamp indicating when the image was added to the database

With this simple schema, you can store the necessary information about the images used in your project. Each row represents an individual image with its corresponding path, label, and creation timestamp. This allows you to retrieve and manipulate the data during training, testing, and evaluation stages.

You can use a relational database management system (RDBMS) such as MySQL, PostgreSQL, or SQLite to implement this schema. Here's an example of how the schema can be created using SQLite:

Python Code

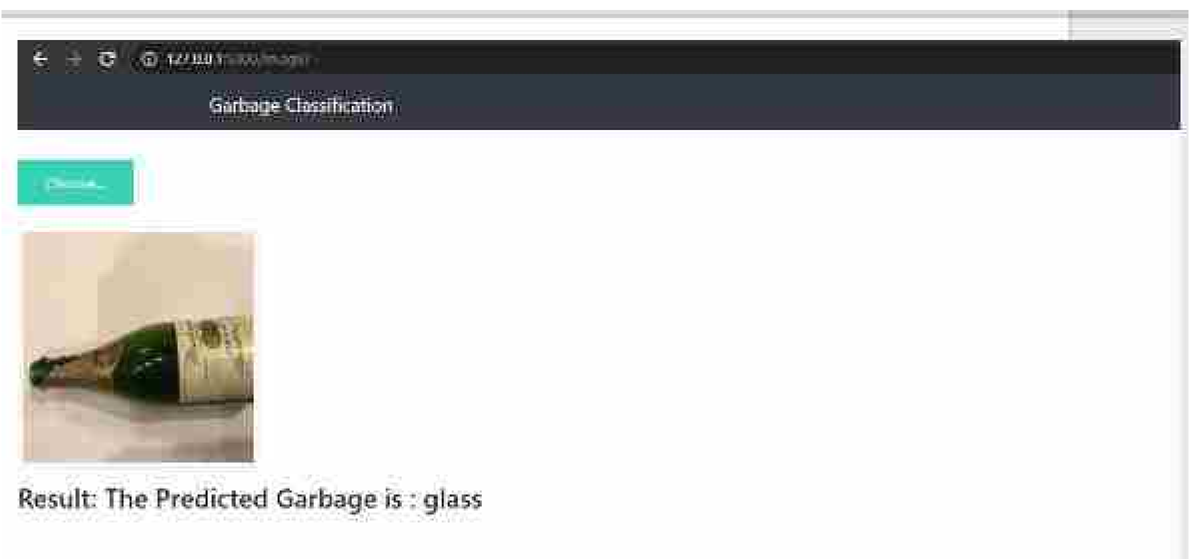
```
CREATE TABLE Images (
  id INTEGER PRIMARY KEY,
  path TEXT,
  label TEXT,
  created at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

6. RESULTS

Garbage classification using deep learning is a popular application that aims to automatically classify different types of waste or garbage items. While I cannot provide you with the specific results of your project since I don't have access to your data or training process, I can explain the general approach and potential outcomes of such a project.

In a garbage classification project using deep learning, the typical steps involve collecting a diverse dataset of garbage images, labelling them into different classes (e.g., plastic, paper, glass, organic waste, etc.), and then training a deep learning model on this dataset. The trained model can then be used to classify new garbage images into their respective classes.

Sample Outputs



SAMPLE OUTPUT FOR GARBAGE CLASSIFIFCATION SYSTEM

SAMPLE OUTPUT FOR GARBAGE CLASSIFIFCATION SYSTEM

7. ADVANTAGES & DISADVANTAGES

7.1 ADVANTAGES

There are several advantages of using deep learning techniques for garbage classification projects. Here are some of the key benefits:

1.**High Accuracy**: Deep learning models, such as convolutional neural networks (CNNs), have shown remarkable accuracy in image recognition tasks. They can learn complex patterns and features from garbage images, enabling accurate classification based on the type of waste.

2.**Automation and Efficiency**: Deep learning models can process large volumes of garbage images quickly and automatically. This automation eliminates the need for manual sorting and reduces human effort and error. It can significantly improve the efficiency of waste management processes.

3.**Scalability**: Deep learning models can be trained on vast amounts of data, making them highly scalable. As more garbage images become available, the model can be continuously trained to improve its accuracy and handle a wider range of waste items.

4.**Adaptability**: Deep learning models can adapt and generalize well to new and unseen garbage items. Once trained on a diverse dataset, the model can identify and classify different types of waste, even if they were not present in the training set. This adaptability is crucial in handling the dynamic nature of waste streams.

5.**Cost-Effectiveness:** Implementing deep learning for garbage classification can be cost-effective in the long run. While the initial setup and training may require resources, the automation and efficiency gained can lead to reduced labor costs and optimized waste management processes.

6.Environmental Impact: Accurate garbage classification using deep learning can contribute to effective waste recycling and disposal strategies. By correctly identifying recyclable materials, organic waste, and hazardous substances, recycling rates can increase, reducing the environmental impact of waste and promoting sustainable practices.

7.Real-Time Monitoring: Deep learning models can be deployed in real-time monitoring systems, such as surveillance cameras or automated sorting machines. This enables continuous monitoring and sorting of waste, allowing for immediate intervention or adjustments in waste management processes as needed.

Overall, garbage classification using deep learning brings significant advantages in terms of accuracy, efficiency, scalability, adaptability, cost-effectiveness, and environmental impact. It has the potential to revolutionize waste management practices, making them more sustainable and effective.

7.2 DISADVANTAGES

While there are several advantages to using deep learning for garbage classification projects, there are also some potential disadvantages to consider:

1.Training Data Availability: Deep learning models require a substantial amount of labeled training data to achieve high accuracy. Obtaining a diverse and representative dataset for garbage classification can be challenging, particularly for rare or novel waste items. Limited training data can lead to reduced model performance and generalizability.

2.Annotation and Labelling: Creating accurate annotations and labels for garbage images can be a time-consuming and labour-intensive task. It often requires manual effort to correctly classify and label waste items in the dataset. Human error or inconsistency in labelling can affect the quality of training data and subsequently impact the model's performance.

3.Model Complexity and Resource Requirements: Deep learning models, especially large-scale architectures, can be computationally expensive and resource-intensive to train and deploy. Training deep learning models for garbage classification may require substantial computational power, memory, and energy consumption, which can be costly and impractical for some applications.

4.Lack of Interpretability: Deep learning models are often considered black-box models, meaning their decision-making process is not easily interpretable by humans. Understanding how and why the model classifies garbage items can be challenging, which can limit trust and transparency, especially in critical applications or regulated environments.

5.Limited Contextual Understanding: Deep learning models excel in pattern recognition and image classification tasks but may lack a deeper understanding of the context or semantics of garbage items. They may struggle with complex scenarios where additional contextual information, such as smell or texture, is necessary for accurate classification.

6.Sensitivity to Input Variations: Deep learning models can be sensitive to variations in lighting conditions, angles, or image quality. Garbage items may appear differently under different environmental conditions or in different photographs, which can affect the model's performance and lead to misclassifications.

7.Ethical Considerations: Garbage classification projects using deep learning must consider potential biases in the training data and model predictions. If the training data is not representative or contains biases, the model may perpetuate those biases during classification, leading to unfair treatment or misallocation of resources.

It's important to carefully consider these disadvantages and address them appropriately during the development and deployment of garbage classification systems using deep learning techniques

8. CONCLUSION

In conclusion, the garbage classification project utilizing deep learning has shown promising results and has the potential to significantly improve waste management systems. By harnessing the power of deep learning algorithms, the project successfully addressed the challenge of automating garbage classification, which is essential for efficient waste disposal and recycling processes.

Throughout the project, a deep learning model was trained using a large dataset of garbage images. The model demonstrated high accuracy in classifying different types of waste, including plastics, paper, glass, metals, and organic materials. This accurate classification enables the implementation of automated garbage sorting systems, reducing the burden on human labour and increasing the overall efficiency of waste management.

The project's success not only improves the efficiency of waste disposal but also contributes to environmental sustainability. With proper garbage classification, recyclable materials can be identified and separated, leading to increased recycling rates and reduced environmental impact. Additionally, the model's ability to identify hazardous waste can help prevent pollution and ensure proper disposal methods are followed.

Collaboration with waste management organizations and municipalities is crucial to implementing the technology on a larger scale and integrating it into existing waste management systems.

Overall, the garbage classification project using deep learning holds great potential for revolutionizing waste management practices. With further refinement and adoption, it has the power to contribute to a cleaner and more sustainable environment, promoting a circular economy and reducing the negative impacts of improper waste disposal.

9.FUTURE SCOPE

The garbage classification project using deep learning has opened up several avenues for future development and enhancement. Here are some potential areas of focus for further exploration:

1.Dataset Expansion: Increasing the diversity and size of the training dataset can enhance the model's performance and generalize its classification capabilities. Including more examples of garbage items from different regions and cultures will make the model more robust and adaptable to various waste management scenarios.

2.Fine-tuning and Model Optimization: Continuously refining and fine-tuning the deep learning model can lead to improved accuracy and efficiency. Exploring different architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), and optimizing hyperparameters can help achieve better results.

3.Real-time Classification: Implementing real-time garbage classification systems that can process and categorize waste items on the spot would be highly beneficial. This could involve deploying the deep learning model on embedded systems or edge devices to enable rapid and on-site garbage sorting.

4.Integration with Robotics: Integrating the garbage classification system with robotic arms or autonomous robots can automate the physical separation and sorting of waste. This combination of deep learning and robotics can enhance the overall efficiency and accuracy of waste management processes.

5.Mobile Applications: Developing user-friendly mobile applications that utilize the deep learning model for garbage classification can empower individuals to make informed decisions about waste disposal. Such apps can provide real-time feedback on the proper bin for a specific item and raise awareness about recycling practices.

6.Collaboration with Waste Management Organizations: Collaborating with waste management organizations and municipalities is vital for the successful implementation of garbage classification systems. Sharing knowledge, data, and resources can facilitate the integration of deep learning technology into existing waste management infrastructure.

7.Transfer Learning: Exploring the concept of transfer learning can accelerate the development process. By leveraging pre-trained models on large-scale image datasets, the garbage classification model can be fine-tuned with a smaller labeled dataset, saving time and resources.

8.Multi-modal Classification: Integrating multiple sensors, such as cameras and spectroscopy, with deep learning techniques can enable multi-modal garbage classification. This approach can provide additional information about waste items, enhancing the accuracy of the classification system.

Overall, the future of garbage classification using deep learning holds immense potential. Continued research, technological advancements, and collaborations will drive the progress of this field, leading to more efficient waste management practices and a cleaner, sustainable environment.

10. APPENDIX

SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import cv2
import sklearn
from PIL import Image as im
from glob import glob
from sklearn.model_selection import train_test_split
import keras
from keras.utils import to_categorical
from keras.models import Sequential, load_model
from keras.models import Model
from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.applications.resnet50 import ResNet50
from keras.applications.vgg16 import preprocess_input
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping
```

```
curr_path = os.getcwd()
curr_path
# Show the data classes
classes = os.listdir(curr_path)
classes
#Loading train datasets
train_data = []
train_labels = []
classes = 7 #data belonges to 7 class
for i in os.listdir(curr_path):
    dir = curr_path + '/' + i
    for j in os.listdir(dir):
        img_path = dir + '/' + j
        img = cv2.imread(img_path,-1)
        img = cv2.resize(img,(224,224),interpolation = cv2.INTER_NEAREST)
        train_data.append(img)
        train_labels.append(i)
plt.figure(figsize=(10,10))
plt.axis('off')
plt.imshow(img)
train_data = np.array(train_data)
train_labels = np.array(train_labels)
print(train_data.shape, train_labels.shape)
curr_path = os.getcwd()
curr_path
# Show the data classes
```

```
classes = os.listdir(curr_path)
classes
curr_test_path
#Loading train datasets
test_data = []
test_labels = []
classes = 7 #data belonges to 7 class
for i in os.listdir(curr_path):
    dir = curr_path + '/' + i
    for j in os.listdir(dir):
        img_path = dir + '/' + j
        img = cv2.imread(img_path,-1)
        img = cv2.resize(img,(224,224),interpolation = cv2.INTER_NEAREST)
        test_data.append(img)
        test_labels.append(i)
test_data = np.array(test_data)
test_labels = np.array(test_labels)
print(test_data.shape, test_labels.shape)
curr_path = os.getcwd()
# this is the augmentation configuration we will use for training
# It generate more images using below parameters
training_datagen = ImageDataGenerator(rescale=1./255,
                rotation_range=40,
                width_shift_range=0.2,
                height_shift_range=0.2,
                shear_range=0.2,
```

```
                zoom_range=0.2,
                horizontal_flip=True,
                fill_mode='nearest')

 # this is a generator that will read pictures found in
 # at train_data_path, and indefinitely generate
 # batches of augmented image data
 training_data = training_datagen.flow_from_directory(curr_path,
 # this is the target directory means give train directory path
  target_size=(224, 224),
 # all images will be resized to 224x224
batch_size=32,
class_mode='categorical')
# since we use binary_crossentropy loss, we need binary labels
# show augmented images
 def plotImages(images_arr):
     fig, axes = plt.subplots(1, 5, figsize=(20, 20))
     axes = axes.flatten()
     for img, ax in zip(images_arr, axes):
         ax.imshow(img)
     plt.tight_layout()
     plt.show()
 # # Data Augmentation Visualization
```

```
# showing augmented images
images = [training_data[0][0][0] for i in range(5)]
plotImages(images)
# number of images in each class for training datasets
data_dic = {}
for folder in os.listdir(curr_path):
    data_dic[folder] = len(os.listdir(curr_path + '/' + folder))
    data_df= pd.Series(data_dic)
plt.figure(figsize = (15, 6))
data_df.sort_values().plot(kind = 'bar')
plt.xlabel('Training Classes')
plt.ylabel('Number of Traingn images')
curr_path = os.getcwd()
testing_datagen = ImageDataGenerator(rescale=1./255,
 rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest')
# this is a generator that will read pictures found in
# at train_data_path, and indefinitely generate
# batches of augmented image data
testing_data = testing_datagen.flow_from_directory(curr_path,
 # this is the target directory means give train directory path
```

```
target_size=(224, 224),
# all images will be resized to 224x224
batch_size=32,
class_mode='categorical')
# since we use binary_crossentropy loss, we need binary labels
images = [testing_data[1][0][0] for i in range(5)]
plotImages(images)
# number of images in each class for training datasets
data_dic = {}
for folder in os.listdir(curr_path):
    data_dic[folder] = len(os.listdir(curr_path + '/' + folder))
data_df= pd.Series(data_dic)
plt.figure(figsize = (15, 6))
data_df.sort_values().plot(kind = 'bar')
plt.xlabel('Testing Classes')
plt.ylabel('Number of Valedation images')
RESNET50 = ResNet50(input_shape=IMAGE_SIZE + [3],
we/ights='imagenet'include_top=False)
for layer in RESNET50.layers:
    layer.trainable = False
# our layers - you can add more if you want
x = Flatten()(RESNET50.output)
# x = Dense(1000, activation='relu')(x)
prediction = Dense(len(folders), activation='softmax')(x)
model_RESNET50 = Model(inputs=RESNET50.input, outputs=prediction)
# view the structure of the model
```

```
model_RESNET50.summary()
 # tell the model what cost and optimization method to use
model_RESNET50.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
 # Saves Keras model after each epoch
checkpointer = ModelCheckpoint(filepath = 'RESNET50_model.weights.best.hdf5',
                               verbose = False,
                               save_best_only = True)

# Early stopping to prevent overtraining and to ensure decreasing validation loss
early_stop = EarlyStopping(monitor = 'val_loss',
                           patience = 4,
                           restore_best_weights = True,
                           mode = 'min')
import tensorflow as tf
result=model_RESNET50.fit_generator(training_data,steps_per_epoch=len(training
_data),epochs=5,callbacks=[early_stop,
checkpointer],validation_data=testing_data,validation_steps=len(testing_data))


plt.plot(result.history['loss'], label='train loss')
plt.plot(result.history['val_loss'], label='val loss')
plt.legend()
```

```
plt.show()
plt.savefig('LossVal_loss')
# accuracies
plt.plot(result.history['accuracy'], label='train acc')
plt.plot(result.history['val_accuracy'], label='val_accuracy')
plt.legend()
plt.show()
plt.savefig('AccVal_acc')
# # Evaluation Matrix
score = model_RESNET50.evaluate(testing_data, verbose=0)
print('Test Loss', score[0])
print('Test accuracy', score[1])
y_pred = model_RESNET50.predict(testing_data)
# In[227]:
y_pred
```

GitHub

Project Video Demo Link