Project Report Format

Topic:

Intelligent Garbage Classification Using Deep Learning

RAMJI T B - Team Leader

ARAVIND R P - Team Member 1

BHUVANESH BABU K R - Team Member 2

KISHORE LAL A R - Team Member 3

GUHAN M - Team Member 4

Faculty Mentor Name : Mr. L Vinoth Kumar

Team ID : NM2023TMID20200

Category : Artificial Intelligence

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. 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.

1.2 Purpose

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. 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 Convolutional Neural Network

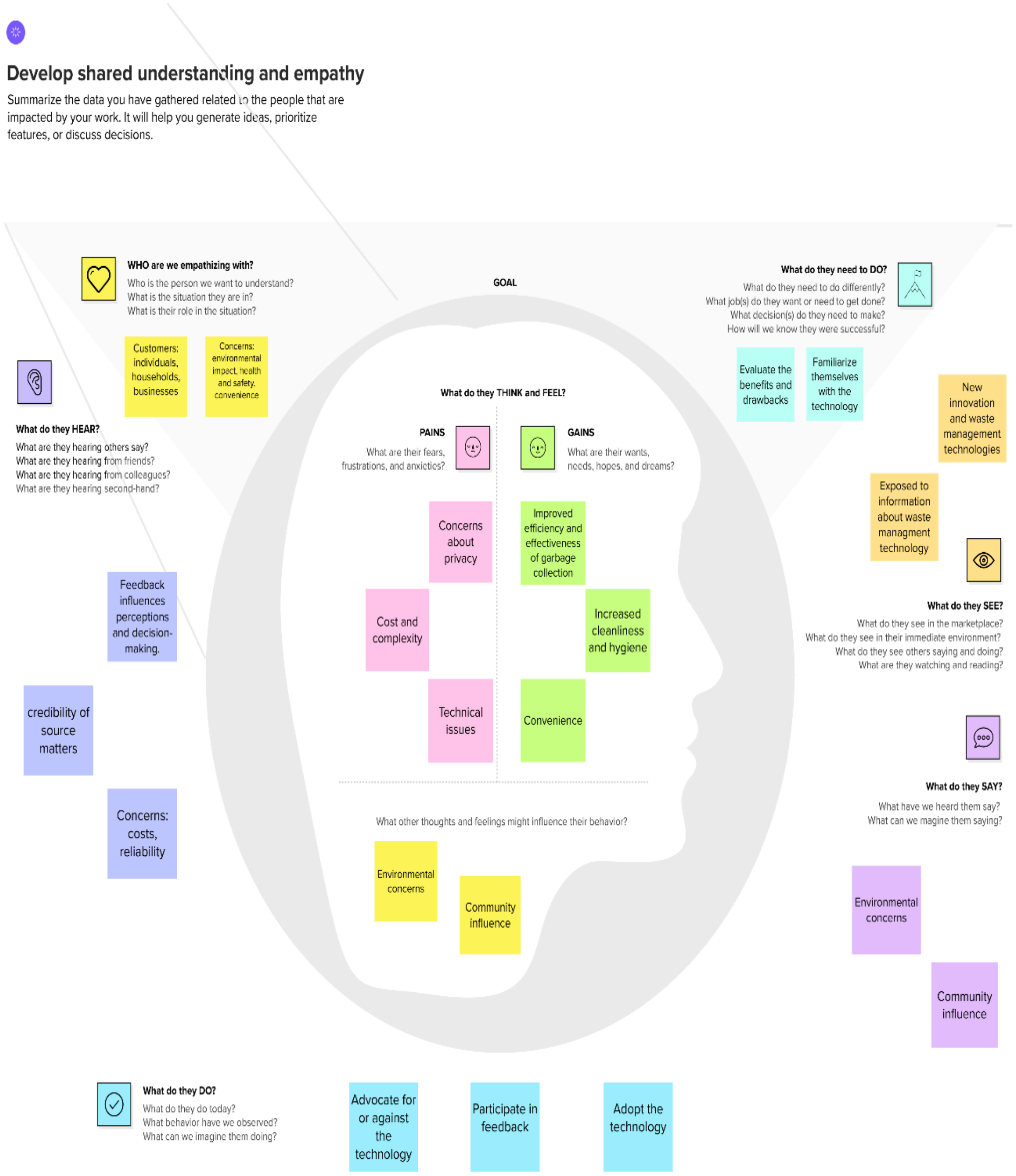
2. IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

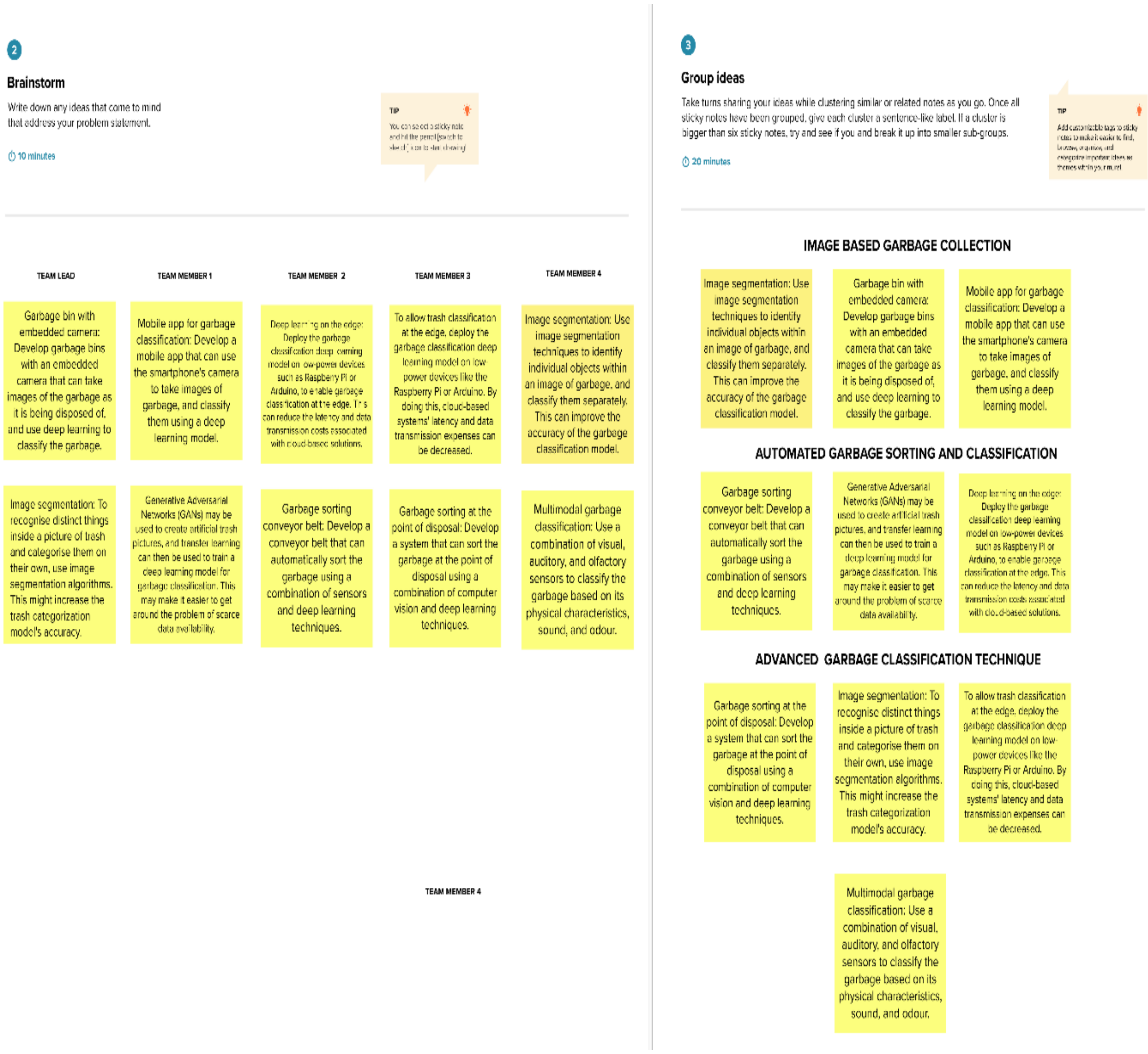
One of the most important tasks in waste management is precisely and efficiently sorting out garbage. Traditional garbage classification techniques can be laborious and prone to mistakes, which results in ineffective waste management techniques. To increase the precision and speed of trash classification, the idea of applying deep learning has been put out as a potential solution. The objective is to create a deep learning model that can correctly categorise various forms of trash

based on their visual characteristics, such as colour and texture. However, creating an efficient deep learning model for garbage classification necessitates overcoming a number of difficulties, such as creating a suitable neural network architecture, gathering and labelling a sizable and varied dataset of garbage images, and incorporating the model into actual waste management systems.

2.2 Empathy Map Canvas



2.3 Ideation & Brainstorming



2.4 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Intelligent Garbage Classification using Deep Learning
2.	Idea / Solution description	The problem is fairly clear; we are all increasingly accustomed to rubbish and other waste products that are detrimental to our society .When it comes to garbage production, the world generates roughly 5 million tonnes of trash each day, and that number is steadily increasing. We need to be mindful of waste as a result. You may learn more about each sort of waste using this model, which categorises six different forms of waste. People will be inspired to recycle and produce less rubbish as a result.
3.	Novelty / Uniqueness	Traditional garbage collection methods are inefficient and lead to excessive waste accumulation in public areas, resulting in health and environmental hazards. Intelligent garbage collection using deep learning can improve the efficiency of waste management systems by accurately detecting and classifying garbage items and optimizing garbage collection routes.
4.	Social Impact / Customer Satisfaction	✓ Accurate garbage detection and classification ✓ Efficient garbage collection routes ✓ Reduced waste accumulation in public areas ✓ Minimized environmental and health hazards ✓ User-friendly system interface ✓ Reliable system performance ✓ Positive feedback from users and stakeholders
5.	Business Model (Revenue Model)	✓ Cost structure: Research and development costs, hardware costs, marketing and sales costs, operational costs ✓ Customer segments: Municipalities and waste management companies, property management companies, industrial and manufacturing companies ✓ Customer relationships: Ongoing maintenance and support, regular communication and engagement with customers ✓ Revenue streams: Sale of hardware and software, subscription-based revenue model

6.	Scalability of the Solution	The need for continuous research and development to improve the system's accuracy, efficiency, and overall performance, and to stay up-to-date with the latest technologies and techniques.

3.**Requirement analysis:**

3.1 Functional requirements:

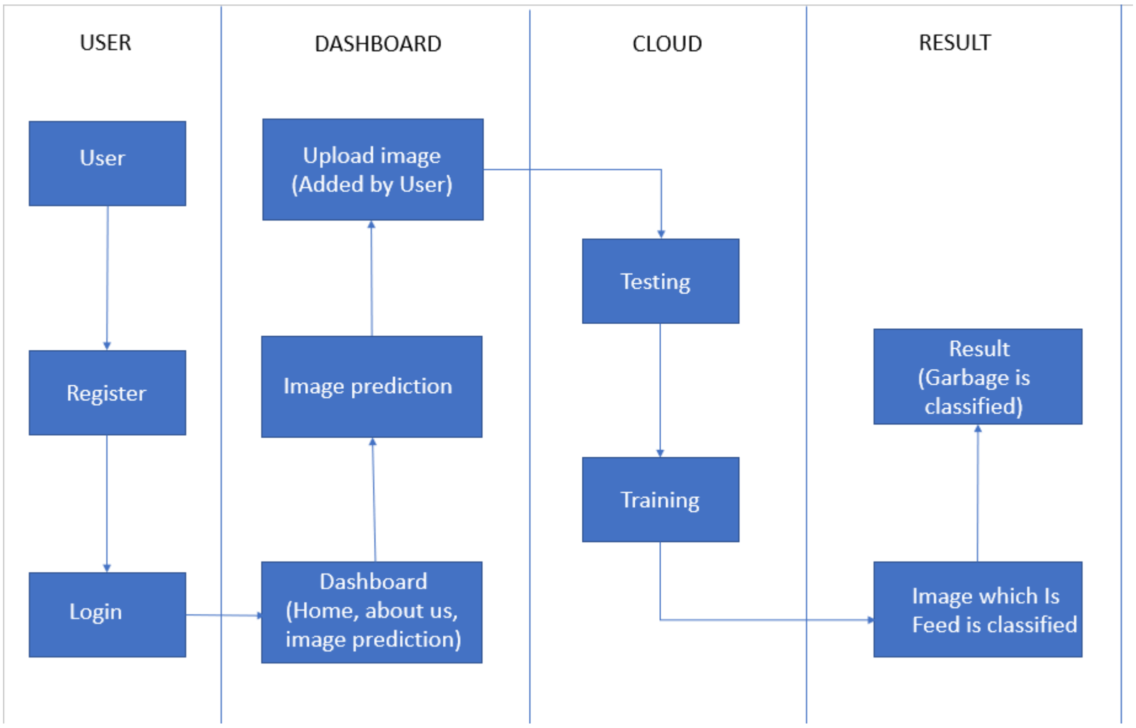
FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email or Mobile number Confirmation via OTP
FR-3	User Login	Enter the User id or User name Enter the password
FR-4	Home page	Exploring Dashboard and info about us
FR-5	About us	Contains the clear description of the project
FR-6	Image prediction	Upload the sample image for testing Ensure the predicted results

3.2 Non functional requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This project has high potential for improving waste management and promoting sustainable practices.
NFR-2	**Security**	Appropriate measures such as data encryption, secure access controls, and continuous monitoring can help ensure the security and privacy of user data.
NFR-3	**Reliability**	The accuracy and consistency of the model's predictions can also be affected by factors such as changes in lighting, camera angles, and variations in garbage items.
NFR-4	**Performance**	The time it takes for the system to classify a single piece of garbage, or a batch of garbage, can impact the overall efficiency and usability of the system.
NFR-5	**Availability**	The availability of high-quality and diverse training data, which is necessary to train accurate deep learning models, can impact the availability and performance of Intelligent Garbage Classification systems.
NFR-6	**Scalability**	Cloud-based infrastructure: Intelligent Garbage Classification systems can be deployed on cloud-based infrastructure such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). These cloud providers offer scalable computing resources, storage, and network services that can be scaled up or down based on demand.

4. PROJECT DESIGN

4.1 Data Flow Diagrams



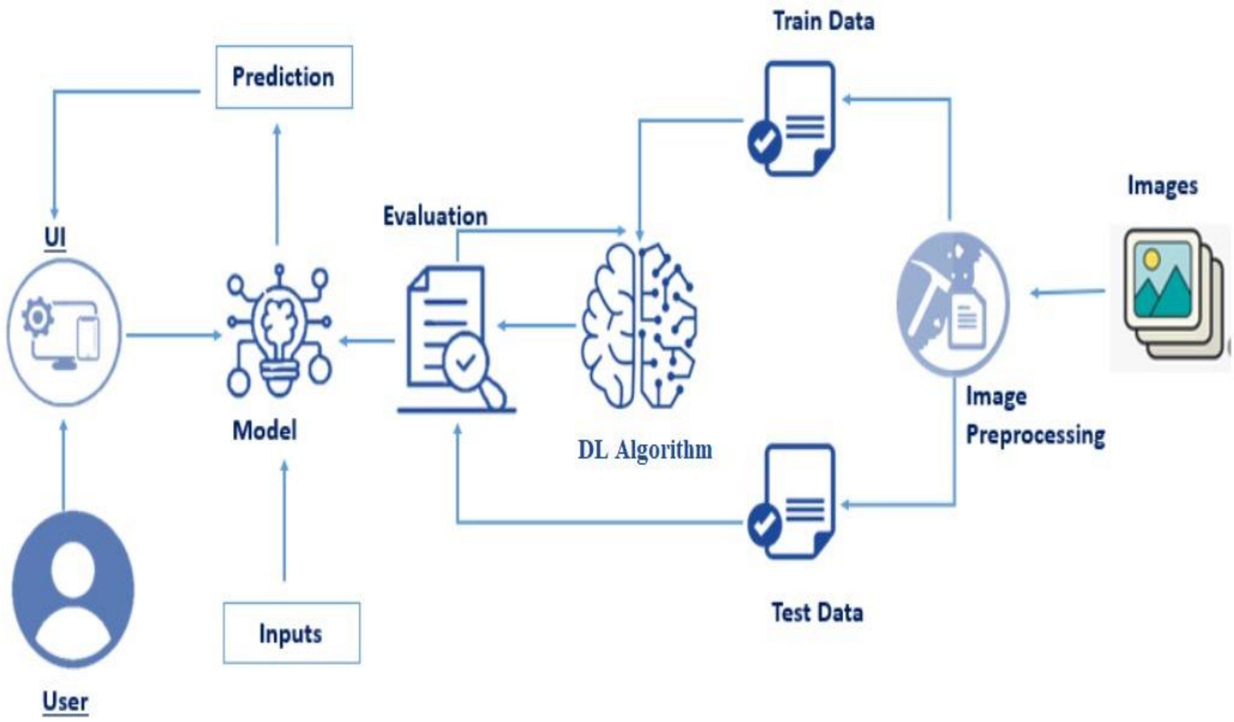
4.2 Solution & Technical Architecture

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

- ✓ Data Collection: The first step is to collect a large dataset of garbage images. This dataset should be diverse and representative of the different types of waste materials. The images can be collected using cameras installed in garbage bins or by manual labelling of images.
- ✓ Image Pre-processing: The collected dataset needs to be pre-processed before it can be used for training the deep learning model. This step includes image resizing, normalization, augmentation, and other techniques to ensure that the dataset is of high quality and is suitable for training the model.
- ✓ Model Selection: The next step is to select a deep learning model architecture that is appropriate for garbage classification. Common models for image

classification include convolutional neural networks (CNNs), which are capable of learning spatial features from images.

- ✓ Train Data: Once the model architecture is selected, the model needs to be trained on the Preprocessed dataset. The training process involves feeding the model with labeled images and adjusting the model weights until it learns to accurately classify different types of waste materials.
- ✓ Evaluation: After training, the model needs to be evaluated on a separate dataset to determine its accuracy and performance. The evaluation dataset should be representative of the real-world environment where the model will be deployed.
- ✓ Prediction: The final step is to deploy the trained model in the target environment. This can be done by integrating the model into a software application or by deploying it on a cloud-based platform. The model can then be used to classify garbage images in real-time.
- ✓ Continuous Improvement: Finally, the model should be continuously monitored and updated to ensure that it remains accurate and up-to-date. This can involve retraining the model on new data or fine-tuning its parameters to improve its 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 Mobile user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	Low	Ramji T B
		USN-2	As a Mobile user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Aravind R P
		USN-3	As a Mobile user, I can register for the application through Mobile number	I can register & access the dashboard with Mobile number and OTP	High	Bhuvanesh Babu K R
		USN-4	As a Mobile user, I can register for the application through Gmail	I can register & access the dashboard with Gmail id and verification code	Medium	Kishore Lal A R
	Login	USN-5	As a Mobile user, I can log into the application by entering User id or User name & password	User is directed to the dashboard	Medium	Guhan M
Customer (Web user)	Registration	USN-1	As a web user, I can register for the application by entering my Gmail and authenticating it with my password	I can access my account / dashboard	High	Kishore Lal A R
	Login	USN-2	As a web user, I can register for the application through Gmail	User is directed to the dashboard	High	Ramji T B
Customer (Mobile user) and Customer (Web user)	Dashboard	USN-1	As a user, I can have the home page, description of the project and Image prediction in the dashboard	I can access the web page based on my need	-	Bhuvanesh babu K R
Customer Care Executive	Server maintenance	USN-1	As a customer care executive, inorder to develop the web we train the modal with more number of image samples done by periodic maintenance break	I can develop the modal better by certain process	High	Ramji T B

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

5.1 Feature 1

```
<!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>
    <div class="social">
      <a href="#"_blank"><i class="fab fa-2x fa-twitter-square"></i></a>
       <a href="#" target="_blank">
        <i class="fab fa-2x  fa-linkedin"></i></a>
        <a href="#">
```

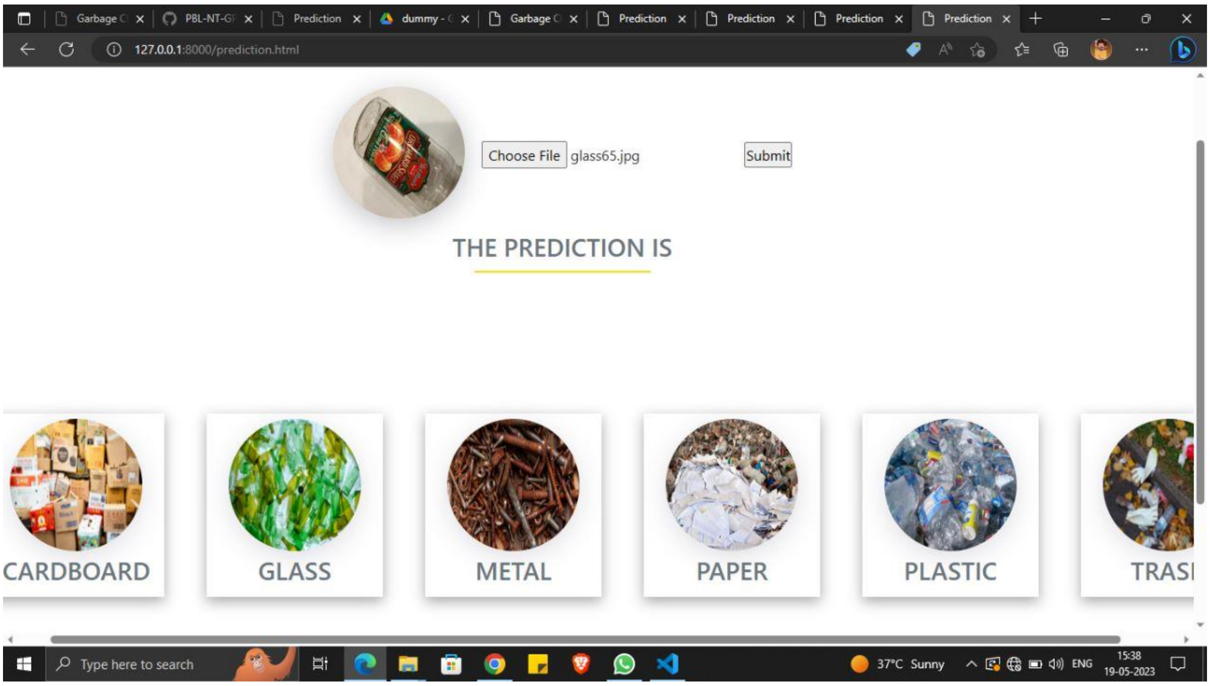
```
            <i class="fab fa-instagram-square fa-2x "></i>
          </a>
       </div>
    </section>
</body>

</html>
```

6. RESULTS

6.1-Performance Metrics

Training Accuracy – 0.8994

Validation Accuracy – 0.8488



5.2 Feature 2 (Back end)

```
import re

import numpy as np

import os

from flask import Flask, app,request,render_template

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(r"C:\Users\Kishore lal\OneDrive\Desktop\HTML\flask\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[int(prediction)])



        result
```

```
        return render_template('prediction.html',prediction=result)
```

```
""" Running our application """
if _name_ == "_main_":
    app.run(debug=False,port=8000)
```

7. ADVANTAGES

- Improved accuracy in garbage classification
- Automation and reduced reliance on manual labour
- Environmental benefits through efficient waste management and recycling
- Scalability to handle large amounts of data

DISADVANTAGES

- Challenges in acquiring diverse and high-quality labelled data
- High computational requirements and potential cost of advanced hardware
- Limitations in generalization to unseen or uncommon garbage types
- Lack of interpretability in the decision-making process of deep learning models
- Ethical considerations regarding data privacy and protection.

8. CONCLUSION

"Intelligent Garbage Classification using deep learning" offers significant advantages for waste management. By leveraging deep learning algorithms, it achieves improved accuracy in classifying different types of garbage. This leads to more efficient waste sorting and recycling, promoting environmental sustainability. The system's scalability enables it to handle large volumes of data, ensuring adaptability to growing waste quantities. Through automation, it reduces the reliance on manual labor, increasing productivity and cost-effectiveness. Overall, intelligent garbage classification using deep learning has the potential to revolutionize waste management practices, paving the way for more efficient, environmentally friendly, and sustainable waste disposal systems.

9. FUTURE SCOPE

1. Enhanced accuracy: Continued research and development can further improve the accuracy of garbage classification models. Fine-tuning deep learning algorithms and leveraging advanced architectures can help reduce misclassifications and increase the precision of identifying different types of garbage.

2. Integration with robotic systems: Intelligent garbage classification can be combined with robotic systems to automate the garbage sorting process. Robots equipped with deep learning models can identify and separate different types of garbage, leading to more efficient and precise waste management.

3. Real-time monitoring and feedback: Implementing real-time monitoring systems that continuously analyze and classify garbage can provide valuable insights into waste patterns. Deep learning models can be trained to recognize emerging trends, such as changes in waste composition, enabling proactive decision-making and targeted waste management strategies.

4. Mobile applications and IoT integration: Mobile applications can be developed to facilitate garbage classification using deep learning on personal devices. Integration with the Internet of Things (IoT) can enable smart garbage bins that automatically identify and sort garbage, providing convenience and promoting proper waste disposal practices.

5. Expansion to global contexts: Adapting intelligent garbage classification systems to different regions and countries can address the unique challenges and waste management practices specific to each location. This includes considering variations in waste types, cultural norms, and recycling infrastructure.

6. Collaboration with recycling industries: Deep learning-based garbage classification can be integrated with recycling industries to optimize resource recovery. By accurately identifying recyclable materials, the system can help streamline recycling processes, minimize waste, and promote circular economy practices.

The future of intelligent garbage classification using deep learning holds great potential for revolutionizing waste management systems, promoting sustainability, and contributing to a cleaner and greener environment.

10. APPENDIX

Source Code :

```
#import image datagenerator Library

from tensorflow.keras.preprocessing.image import ImageDataGenerator

#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)



#image data augmentation to the testing data.

val_datagen = ImageDataGenerator (rescale = 1./255)

from google.colab import drive

drive.mount('/content/drive')

train_transform = train_datagen.flow_from_directory (r"/content/drive/MyDrive/Garbage
classification/Garbage classification",

                                    target_size=(128,128),

                                    batch_size=64,

                                    class_mode='categorical')



test_transform = val_datagen.flow_from_directory (r"/content/drive/MyDrive/Garbage
classification/Garbage classification",

                                  target_size=(128,128),

                                  batch_size=64,

                                  class_mode='categorical')



#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 a convolution kernel import Convolution2D

from tensorflow.keras.layers import Convolution2D

# Adding Max pooling Layer

from tensorflow.keras.layers import MaxPooling2D

# Adding Flatten Layer

from tensorflow.keras.layers import Flatten

from tensorflow.keras.optimizers import Adam

# Initializing the model

model=Sequential()

# Initializing the model

model=Sequential()

#First Convolution layer and pooling

model.add(Convolution2D(32, (3,3), input_shape=(128,128,3), activation='relu'))

model.add(MaxPooling2D(2,2))

#Second Convolution layer and pooling

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

#input shape is going to be the pooled feature maps from the previous convolution.

model.add(MaxPooling2D(pool_size=2))

#Third Convolution layer and pooling

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

model.add(MaxPooling2D(2,2))

#Fourth Convolution layer and pooling

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

```
#input shape is going to be the pooled feature maps from the previous convolution.

model.add(MaxPooling2D(pool_size=2))

#Flattening the Layers

model.add(Flatten())

# Adding 1st hidden Layer

model.add(Dense (kernel_initializer='uniform', activation='relu', units=150))

# Adding 2nd hidden layer

model.add(Dense (kernel_initializer='uniform', activation='relu', units=68))

model.add(Dense(kernel_initializer='uniform', activation='softmax', units=6))

#Compiling the CNN Model

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])

model.summary()

res = model.fit_generator(train_transform, steps_per_epoch=2527//64,
validation_steps=782//64, epochs=10, validation_data=test_transform)

model.save("Garbage1.h5")

#import numpy Library

import numpy as np

#import load_model method to load our saved model

from tensorflow.keras.models import load_model

#import image from keras.preprocessing

from tensorflow.keras.preprocessing import image

#Loading our saved model file

model = load_model("Garbage1.h5")


img = image.load_img (r"/content/drive/MyDrive/Garbage
classification/test/glass/glass123.jpg", target_size=(128,128))
```

```
x=image.img_to_array(img) #converting in to array format

x=np.expand_dims (x,axis=0) #changing its dimensions as per our requirement

#img_data=preprocess_input(x)

#img_data.shape

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

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

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

result

train_transform.class_indices

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

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

result

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

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

result1
```

for the h5 file model.