INTELLIGENT WASTE CLASSIFICATION SYSTEM USING DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK

Mini Project Report

Submitted by

NAINA VINCENT

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY-683577, ERNAKULAM(DIST)
FEBRUARY 2022

DECLARATION

I, **Naina vincent**, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date : 04-03-2022

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report titled **”INTELLIGENT WASTE CLASSIFICATION SYSTEM USING DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK”** submitted by **Naina Vincent** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by them during the year 2021.

Project Guide **Head of the Department**

Submitted for the viva-voice held on . . . . at . . . .

Examiner1 :

Examiner2 :

ACKNOWLEDGEMENT

I am extremely glad to present my mini project which i did as a part of our curriculum. I take this opportunity to express my sincere thanks to those who helped me in bringing out the report of my project.

We are deeply grateful to Dr. George Manoj George, Principal, FISAT, Angamaly and Dr. C. Sheela, Vice Principal, FISAT, Angamaly. Our sincere thanks to Dr.Deepa Mary Mathews, Head of the department of MCA, FISAT, who had been a source of inspiration. During the period of our project work, we have received generous help from Mrs. Anju L, my project guide, which i like to put on record here with deep gratitude and great pleasure.

I express my heartfelt thanks to all the faculty members in our department for their constant encouragement and never ending support throughout the project.

Finally i am grateful to all our friends who gave us a lot of sugges- tions for the successful completion of this project.

ABSTRACT

One of the most important steps of waste management is the separation of the waste into the different components and this process is normally done manually by hand picking. To simplify the process, I propose an intelligent waste material classification system, which will be developed using best generalized model from the customized versions of VGG-16, VGG19, Xception, and InceptionV3 architectures of deep convolutional neural network. Above 4 architectures will be customized and trained separately using waste images dataset to achieve better results. State of the art data augmentation techniques will be applied on dataset images, to make the deep learning model to learn in a better and efficient way. Comparison of these customized architectures in terms of classification accuracy will be shown and suitable generalized model with optimal performance will be proposed for the deployment. This model can classify waste with 9 different waste materials (LIGHT BULBS, PAPER, PLASTIC, ORGANIC, GLASS, BATTERIES, CLOTHES, METAL, E-WASTE) in an improved and efficient way than conventional CNN method and it will show the details of that waste materials.

Contents

Chapter 1

INTRODUCTION

Waste management is the precise name for the collection, transportation, disposal or recycling and monitoring of waste. This term is assigned to the material, waste material that is produced through human being activity. This material is managed to avoid its adverse effect over human health and environment. The world bank report showed that there are almost 4 billion tons of waste around the world every year and the urban alone contributes a lot to this number, the waste is predicted to increase by 70 percent in the year 2025. The most important reason for proper waste management is to protect the environment and for the health and safety of the population. Reduce the volume of the solid waste stream through the implementation of waste reduction and recycling programs.

Propose an intelligent waste material classification system, which will be developed using best generalized model from the customized versions of VGG-16, VGG-19, Xception, and InceptionV3 architectures of deep convolutional neural network. Above 4 architectures will be customized and trained separately using waste images dataset to achieve better results. State of the art data augmentation techniques will be applied on dataset images, to make the deep learning model to learn in a better and efficient way. Comparison of these customized architectures in terms of classification accuracy will be shown and suitable generalized model with optimal performance will be proposed for the deployment.

This model can classify waste with 9 different waste materials (LIGHT BULBS, PAPER, PLASTIC, ORGANIC, GLASS, BATTERIES, CLOTHES, METAL, E-WASTE) in an improved and efficient way than conventional CNN method and it will show the details of that waste materials. This will help to raise awareness for people to reduce and recycle waste.

Chapter 2

PROOF OF CONCEPT

Objectives

The recycling rate of waste in India is comparatively lower than other countries for different categories . If classification of waste is done at the earliest level most of the problem for recycling of waste can be solved easily. In this project, I aim to classify the images of recyclable waste into different categories of recyclable elements. The proposed system will be trained and tested on the trash image data set containing 8000+ above category images and expecting an accuracy of 87percentage plus on the data set. The separation process of the waste will be faster and intelligent using the proposed waste material classification system without or reducing human involvement.

Chapter 3

IMPLEMENTATION

In this project one of the main step is importing the data set. In this data set, different type of waste are used as the data.The data set is inclusive waste , different types of waste will be classified into 9 classes . After that the processes such as customization of model and training the model with the data are done.Once a model is trained I can get know the training and validation accuracy of the model and also calculate the training and validation loss etc.After building the models , I compare the models and choose the model from that which have the highest accuracy. That model will be fine tuned and saved. Design and develop the web user interface using the web framework Flask. The saved model will be integrated with the User Interface.

TOOLS OR PROGRAMMING LANGUAGE

- FRONT END :
  - Html
  - CSS
  - Bootstrap

- BACK END :
  - Python

MODULES

1. Data Preprocessing :

Data preprocessing is an iterative process for the transformation of the raw data into understandable and useable forms. Raw datasets are usually characterized by incompleteness, inconsistencies, lacking in behavior, and trends while containing errors. The preprocessing is essential to handle the missing values and address inconsistencies

2. Feature Extraction :

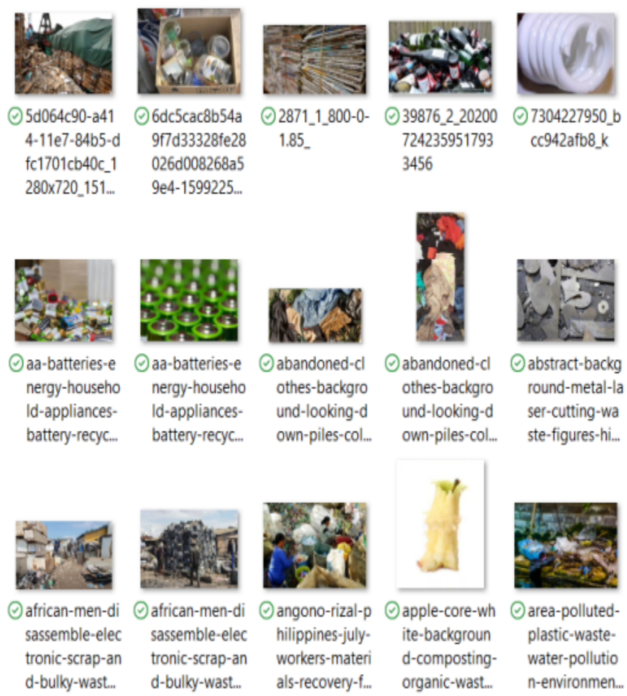
Feature extraction involves reducing the number of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

3. Training the Model :

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model

4. Evaluation :

Model evaluation techniques in machine learning are helping us to find a better model among all other models in machine learning. It is simply the selection of machine learning models or measuring the performance of machine learning models.



DATASET

The data set from Kaggle for building the model. This dataset and consist of 4546 images belonging to 9 classes, which is divided into four different classes LIGHT BULBS, PAPER, PLASTIC, ORGANIC, GLASS, BATTERIES, CLOTHES, METAL, E-WASTE

ALGORITHM TO BE USED

Convolutional Neural Network:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

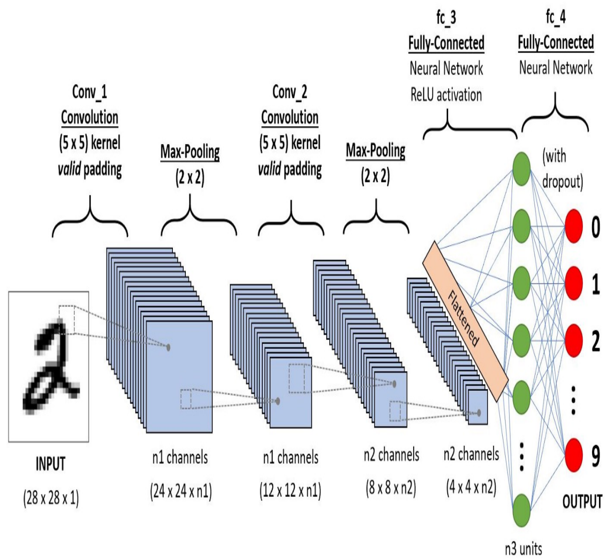


Figure 3.1: A CNN sequence to classify handwritten digits.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

Customized models:

InceptionV3:

Inception-v3 is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7 x 7 convolutions, and the use of an auxiliary classifer to propagate label information lower down the network (along with the use of batch normalization for layers in the sidehead).Inception v3 TPU training runs match accuracy curves produced by GPU jobs of similar configuration. The model has been successfully trained on v2-8, v2-128, and v2-512 configurations. The model has attained greater than 78.1 percentage accuracy in about 170 epochs on each of these.

Xception:

Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. It was developed by Google researchers Description. Xception is a convolutional neural network that is 71 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet databas

VGG16:

VGG16 (also called OxfordNet) is a convolutional neural network architecture named after the Visual Geometry Group from Oxford, who developed it. By only keeping the convolutional modules, our model can be adapted to arbitrary input sizes. The model loads a set of weights pre-trained on ImageNet.

VGG19:

Compared with VGG16, VGG19 is slightly better but requests more memory. VGG16 model is composed of convolutions layers, max pooling layers, and fully connected layers. The total is 16 layers with 5 blocks and each block with a

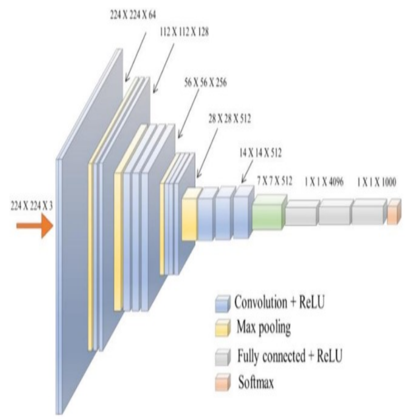


Figure 3.2: VGG19 Architecture

max pooling layer.This network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fullyconnected layers, each with 4,096 nodes are then followed by a softmax classifier. VGG-19 is a convolutional neural network that is 19 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224

Chapter 4

RESULT ANALYSIS

Accuracy is often the most used metric representing the percentage of correctly predicted observations, either true or false. To calculate the accuracy of a model performance, the following equation can be used: In most cases, high accuracy value represents a good model, but considering the fact that we are training a classification model in our case, an article that was predicted as true while it was actually false (false positive) can have negative consequences; similarly, if an article was predicted as false while it contained factual data, this can create trust issues.

Accuarcy = TP+TN/TP+TN+FP+FN

Confusion Matrix

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:

1. TP = True Positives

2. FP = False Positives

3. TN = True Negatives

4. FN = False Negatives

Chapter 5

CONCLUSION AND FUTURE SCOPE

Conclusion

Proposed a waste classification system that is able to separate different components of waste using the Machine learning tools. This system can be used to automatically classify waste and help in reducing human intervention and preventing infection and pollution. From the result, when tested against the trash dataset, i got an accuracy for customized VGG19 83.48 percentage. The separation process of the waste will be faster and intelligent using our system without or reducing human involvement.

Future Scope

This project can be further enhanced to provide greater flexibility and performance with certain modification whenever necessary. If more image is added to the dataset, the system accuracy can be improved In the future, I will tend to improve our system to be able to categories more waste item, by turning some of the parameters used.

Chapter 6

APPENDIX

```
from flask import Flask, request, render_template, redirect, jsonif
from flask_jsglue import JSGlue # this is use for url_for() working
javascript which is help us to navigate the url
import util
import os
from werkzeug.utils import secure_filename

application = Flask(__name__)

# JSGlue is use for url_for() working inside javascript which is he
navigate the url
jsglue = JSGlue() # create an object of JsGlue
jsglue.init_app(application) # and assign the app as a init app to
instance of JsGlue

util.load_artifacts()
#home page
@application.route("/")
def home():
    return render_template("home.html")

#classify waste
@application.route("/classifywaste", methods = ["POST"])
def classifywaste():
    image_data = request.files["file"]
    #save the image to uploads
    basepath = os.path.dirname(__file__)
    image_path = os.path.join(basepath, "uploads",
secure_filename(image_data.filename))
    image_data.save(image_path)

    predicted_value, details, video1, video2 =
util.classify_waste(image_path)
    os.remove(image_path)
    return jsonify(predicted_value=predicted_value, details=details
video1=video1, video2=video2)

# here is route of 404 means page not found error
@application.errorhandler(404)
def page_not_found(e):
    # here i created my own 404 page which will be redirect when 40
occured in this web app
    return render_template("404.html"), 404

if __name__ == "__main__":
    application.run()
```

```
import tensorflow as tf
import numpy as np

model = None
output_class = ["Batteries", "Clothes", "E-waste", "Glass", "Light
"Metal", "Organic", "Paper", "Plastic"]
data = {
"Batteries":

"Clothes":

"E-waste":

"Glass":

"Light Bulbs":

"Metal":

"Organic":

"Paper":

"Plastic":

}


def load_artifacts():
    global model
    model = tf.keras.models.load_model("classifyWaste.h5")

def classify_waste(image_path):
   global model, output_class
   test_image = tf.keras.preprocessing.image.load_img(image_path,
target_size=(224, 224))
   test_image = tf.keras.preprocessing.image.img_to_array(test_imag
   test_image = np.expand_dims(test_image, axis = 0)
   predicted_array = model.predict(test_image)
   predicted_value = output_class[np.argmax(predicted_array)]
   return predicted_value, data[predicted_value][0],
data[predicted_value][1], data[predicted_value][2]
```

VGG19.h5

```
#Install tensorflow with gpu
 ! pip install tensorflow-gpu
 # Dimensions of our images.
 img_width =224
 img_height = 224
 train_data_dir = '/content/drive/MyDrive/Datasets/waste/Train' |
validation_data_dir = '/content/drive/MyDrive/Datasets/waste/Val'
 # epochs - number times that the algorithm will work(iterate) through the entire training
 epochs = 70
 #batch_size -- the number of training examples utilized in one iteration.
#batch size of 16 is a good starting point, and you should also try with 32, 64, 128, and 2
batch_size = 16
from tensorflow.keras.applications.vgg19 import
 VGG19,preprocess_input
 basemdlvgg19=VGG19()
 basemdlvgg19.summary()
basemdlvgg19=VGG19(weights='imagenet',input_shape=(img_width,img_height,3),
include_top=False)
 # don't train existing weights
 for lyr in basemdlvgg19.layers:
 lyr.trainable=False
from tensorflow.keras import layers
 from tensorflow.keras.layers import Dense, Flatten
 from tensorflow.keras.models import Model
x=basemdlvgg19.output
 x=Dense(1024,activation='relu')(x)
 x=Dense(1024,activation='relu')(x)
 x=Flatten()(x)
 # Add a fully connected layer with 512 hidden units and ReLU activation
x=Dense(512,activation='relu')(x)
 x=Dense(256,activation='relu')(x)
```

```
# Add a dropout rate of 0.5 # to avoid overfitting
x = layers.Dropout(0.3)(x)
 # Add a final softmax layer for classification -- output
x=Dense(9,activation='softmax')(x)
 mymdl = Model(basemdlvgg19.input, x)
 mymdl.compile( loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'] )
 from tensorflow.keras.preprocessing.image import
 ImageDataGenerator train_datagen =
ImageDataGenerator(preprocessing_function=preprocess_input,rescale = 1/255, shear_range =
zoom_range = 0.2, brightness_range = (0.1, 0.5), horizontal_flip=True)
# this is the augmentation configuration we will use for testing:only rescaling
 val_datagen = ImageDataGenerator(rescale=1. / 255)
training_set = train_datagen.flow_from_directory(train_data_dir, target_size=(img_width,
img_height), batch_size = batch_size, class_mode = 'categorical')
 val_set = val_datagen.flow_from_directory(validation_data_dir, target_size = (img_width,
img_height), batch_size = batch_size, class_mode = 'categorical')
 history_vgg19 = model.fit( training_set, validation_data=val_set, epochs=epochs,
steps_per_epoch=len(training_set), validation_steps=len(val_set)) model.save("mymodel.h5")
 acc = history_vgg19.history['acc']
val_acc = history_vgg19.history['val_acc']
loss = history_vgg19.history['loss']
val_loss = history_vgg19.history['val_loss']
 epochs = range(len(acc))
plt.plot(epochs, acc, 'b', label='Training acc')
 plt.plot(epochs, val_acc, 'r', label='Validation acc')
 plt.title('Training and validation accuracy')
 plt.legend()
 plt.figure()
 plt.plot(epochs, loss, 'b', label='Training loss')
 plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
 plt.legend()
 plt.show()
```

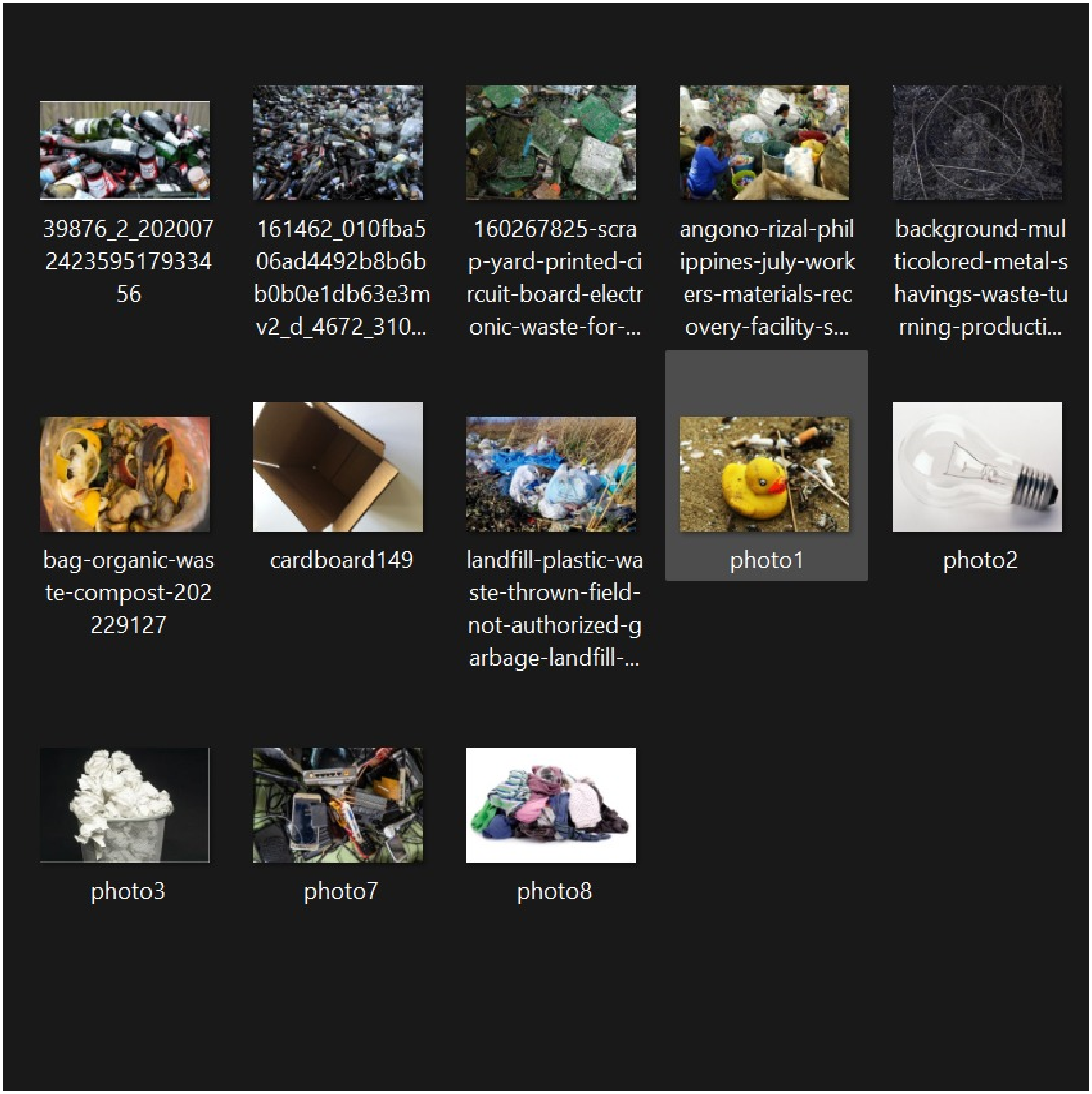


Figure 6.1: Sample Dataset.

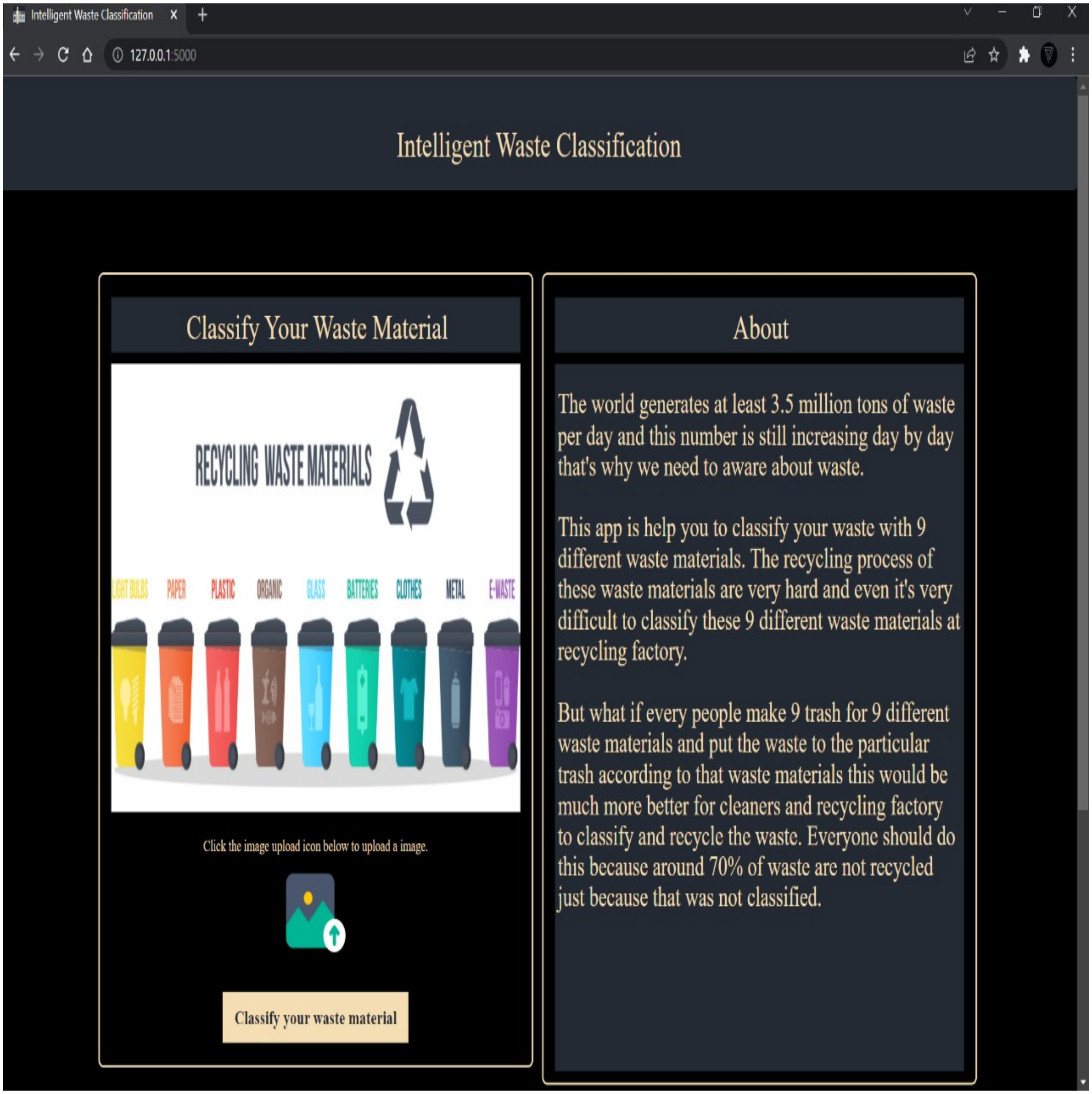
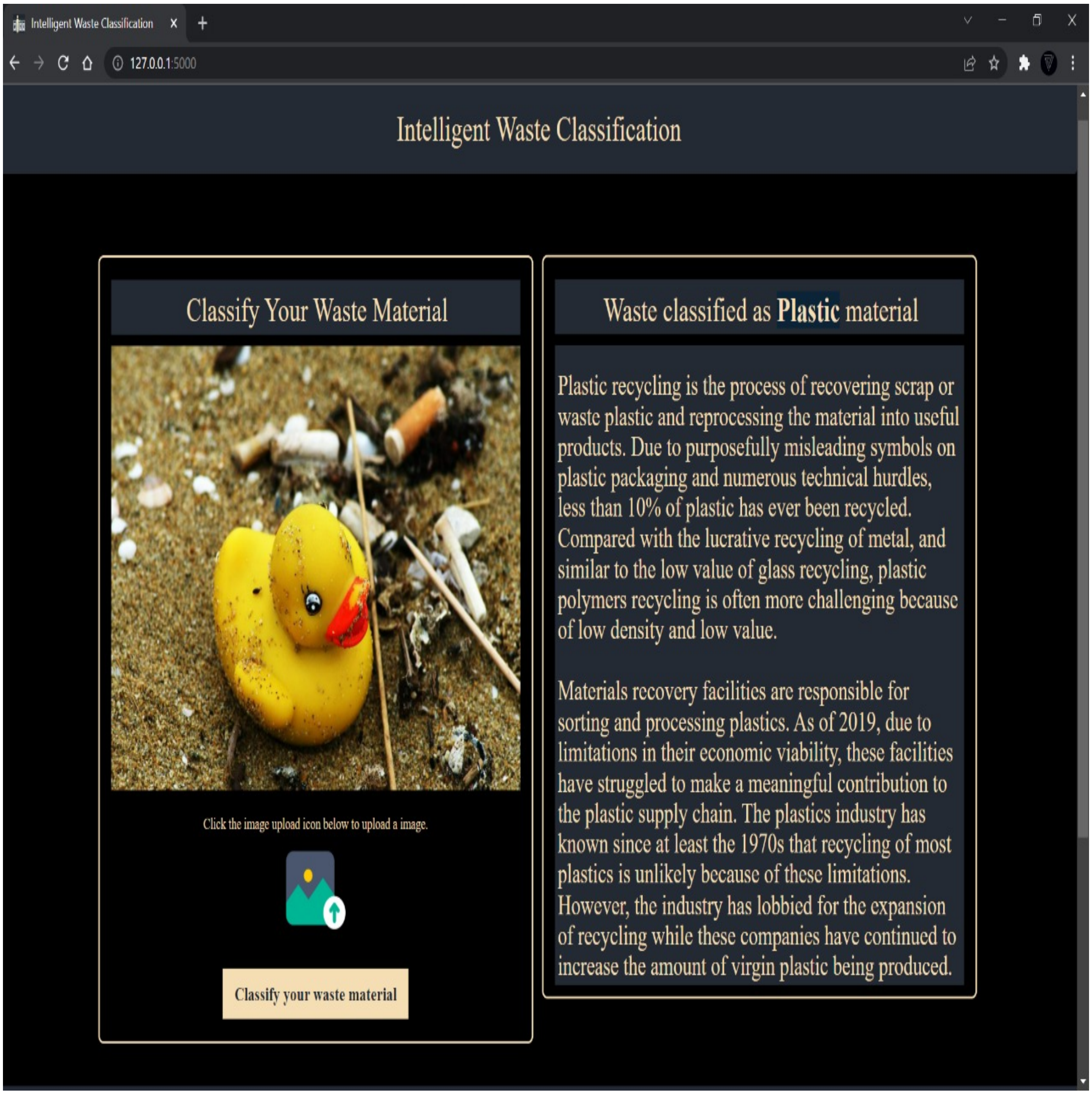


Figure 6.2: UI Design


Intelligent Waste Classification
127.0.0.1:5000
Intelligent Waste Classification
Classify Your Waste Material
Click the image upload icon below to upload a image.
Classify your waste material
Waste classified as Plastic material
Plastic recycling is the process of recovering scrap or waste plastic and reprocessing the material into useful products. Due to purposefully misleading symbols on plastic packaging and numerous technical hurdles, less than 10% of plastic has ever been recycled. Compared with the lucrative recycling of metal, and similar to the low value of glass recycling, plastic polymers recycling is often more challenging because of low density and low value.
Materials recovery facilities are responsible for sorting and processing plastics. As of 2019, due to limitations in their economic viability, these facilities have struggled to make a meaningful contribution to the plastic supply chain. The plastics industry has known since at least the 1970s that recycling of most plastics is unlikely because of these limitations. However, the industry has lobbied for the expansion of recycling while these companies have continued to increase the amount of virgin plastic being produced.

Chapter 7

REFERENCES

[1] D. Hoornweg and P. Bhada-Tata, “A Global Review of Solid Waste Management,” (2012) 1-116.

[2] Niehs.nih.gov, “Cancer and the Environment,” 46, 2018. [Online]. Available: https://www.niehs.nih.gov/health/materials/cancerandtheenvironment508.pdf.

[3] N. Johansson and H. Corvellec, “Waste policies gone soft: An analysis of European and Swedish waste prevention plans,” Waste Manag., vol. 77, pp. 322–332, 77 (2018) 322-332.

[4] K. He, X. Zhang, S. Ren and J. Sun, ”Deep Residual Learning for Image Recognition,” 2015.

[5] D. T. Vijayakumar and M. R. Vinothkanna, ”Mellowness detection in dragon fruit using deep learning strategy,” Journal of Innovative Image Processing (JIIP) 2, pp. 35-43, 2020.

[6] R. C. O’Reilly, D. Wyatte, S. Herd, B. Mingus, and D. J. Jilk, “Recurrent processing during object recognition,” Front. Psychol.,4 (2013) 1-12.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with

Deep Convolutional Neural Networks," ImageNet Classif. with Deep Convolutional Neural Networks, (2012) 1097–1105.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal loss for Dense Object Detection.," in IEEE International Conference on Computer Vision (ICCV), 2017.

[9] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013," Comput. Vision–ECCV 2014, 8689 (2014) 818-833.

[10] M. Lin, Q. Chen, and S. Yan, "Network In Network," arXiv Prepr., , (2013) 10-20

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, 86 (1998) 2278-2323.

[12] S. Zhang and E. Forssberg, "Intellige–nt Liberation and classification of electronic scrap," in Powder Technology, (1999) 295-301