“GARBAGE CLASSIFICATION USING IBM CLOUD” CHAPTER 1

INTRODUCTION

With the increase in the number of industries in the urban area, the disposal of solid waste is becoming a big problem, and solid waste includes paper, wood, plastic, metal, glass, etc. The common way of managing waste is burning waste and this method can cause air pollution and some hazardous materials from the waste spread into the air which can cause cancer. Hence it is necessary to recycle waste to protect the environment and human beings’ health, and we need to separate the waste into different components which can be recycled using different ways. The present way of separating waste/garbage is the hand-picking method, whereby someone is employed to separate out the different objects/materials. The person who separates waste is prone to diseases due to the harmful substances in the garbage. This problem can be overcome by automating the garbage classification process. In this project, we will be building a deep-learning model that can detect and classify types of garbage. A web application is integrated with the model, from where the user can upload a garbage image like paper waste, plastic waste, etc., and see the analyzed results on User Interface.

1.1 SYSTEM ARCHITECTURE

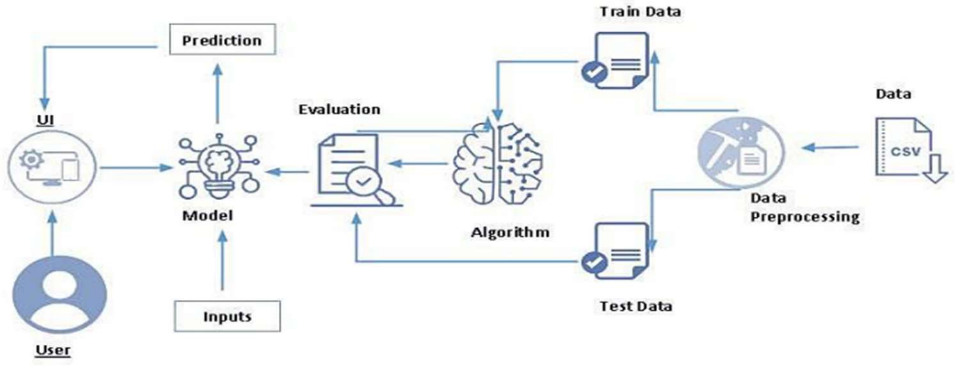


Fig 1.1 Architecture Diagram

1.2. OVERVIEW

By the end of this project we will:

- know fundamental concepts and techniques of Artificial Neural Networks and
- Gain a broad understanding of image data.
- Work with Sequential type of modeling.
- Work with Keras capabilities
- Work with image processing techniques
- know how to build a web application using the Flask framework.
- Convolution Neural netwoks

1.3. PURPOSE

Several times we have come across the news that people working in garbage yards face severe diseases which directly impact their health. The purpose of this project is to reduce human involvement in the process of Garbage classification and replace this with an Intelligent machine that does the Garbage classification the same as humans and this protects humans from the toxic and dangerous environment of garbage classification. “We can save people working in Garbage yards which in turn is a great achievement.”

1.4. EXISTING PROBLEM

Municipal solid waste (MSW) is hazardous to human health and the environment if not properly handled throughout all management processes, beginning with collection, separation, transfer, treatment, disposal, or recycling and reuse. The World Health Organization (WHO) has raised awareness of the dangers of improper solid waste disposal in terms of soil, water, and air pollution, as well as the health consequences for the people living in the surrounding areas. MSW production is predicted to reach 3.40 billion tonnes globally by 2050. Inadequate waste management is one of the causes of the rise of infectious diseases. Most viruses, bacteria, and parasites that cause illness are found in blood, bodily fluids, and bodily secretions, which are components of bio-medical waste. This spreads through several human contacts, each of whom is a possible "receiver" of the illness. The Human Immunodeficiency Virus (HIV) and hepatitis viruses are at the forefront of a long list of illnesses and disorders that have been linked to biomedical waste. Other prevalent diseases spread owing to poor waste management include tuberculosis, pneumonia, diarrhea, tetanus, whooping cough, and others. “Workers who work in dumping areas are badly exposed and are more probable of getting infected.”

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1.AIM

Aim is to design a intelligent human like machine which is capable of classifying garbage based on its previous training given by using multiple images of distinct garbage materials.

2.2 SCOPE OF THE PROJECT

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 to develop an automated system that can sort waste. and this system can take 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 Networks.

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

2.3 WORK FLOW

- User interacts with a user interface to upload the image.
- The uploaded image is analyzed by the model which is integrated.
- Once the model analyses the uploaded image, the prediction is showcased on the UI.
- Data collection:

  Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes.
- Data Preprocessing:

  Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process.
  - Import the ImageDataGenerator library.
  - Configure ImageDataGenerator class.
  - Apply ImageDataGenerator functionality to Trainset and Testset
- Model Building:
  - Import the model-building Libraries.
  - Initializing the model.
  - Adding Input Layer.
  - Adding Hidden Layer.
  - Adding Output Layer.
  - Configure the Learning Process.
  - Training and testing the model.
  - Optimize the Model.
  - Save the Model.
- Application Building.
  - Create an HTML file.
  - Build Python Code.

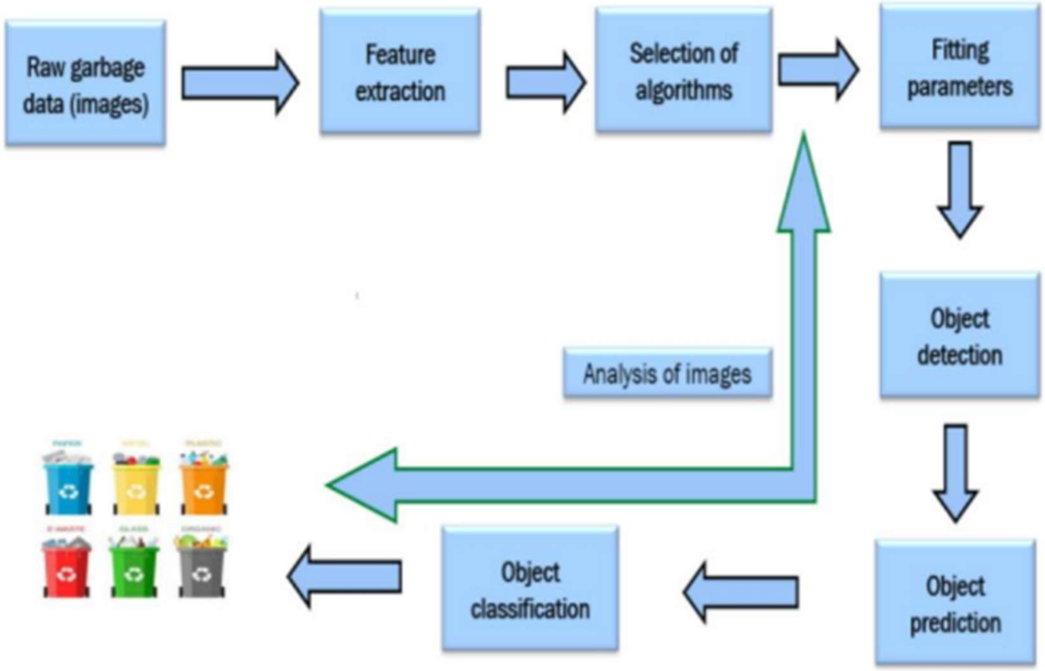


Fig 2.1 Block Diagram

2.4 WORK FLOW DESCRIPTION

The above picture depicts the block diagram of the process which gives clear insights into the entire process where raw garbage data i.e., images then feature extraction is done in which features of garbage are extracted and then stored which are further used in the classification process. After this selection of appropriate algorithms is done then parameters are fitted for the process of object detection and then object prediction and classification is the final step.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

3.1 HARDWARE / SOFTWARE DESIGNING

To complete this project you should have the following software and packages

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder. To build Deep learning models you must require the following packages

Tensor flow

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML-powered applications.

Keras

Keras leverages various optimization techniques to make high-level neural network API easier and more performant. It supports the following features: • Consistent, simple, and extensible API. • Minimal structure - easy to achieve the result without any frills. • It supports multiple platforms and backends. • It is a user-friendly framework that runs on both CPU and GPU. • Highly scalability of computation.

Flask

Web framework used for building Web applications

Python packages

- open anaconda prompt as administrator
- Type “pip install numpy” and click enter.
- Type “pip install pandas” and click enter.
- Type “pip install sci-kit-learn” and click enter.
- Type “pip install tensorflow==2.3.2” and click enter.
- Type “pip install keras==2.3.1” and click enter.
- Type “pip install Flask” and click enter.

3.2 METHODS AND ALGORITHMS USED

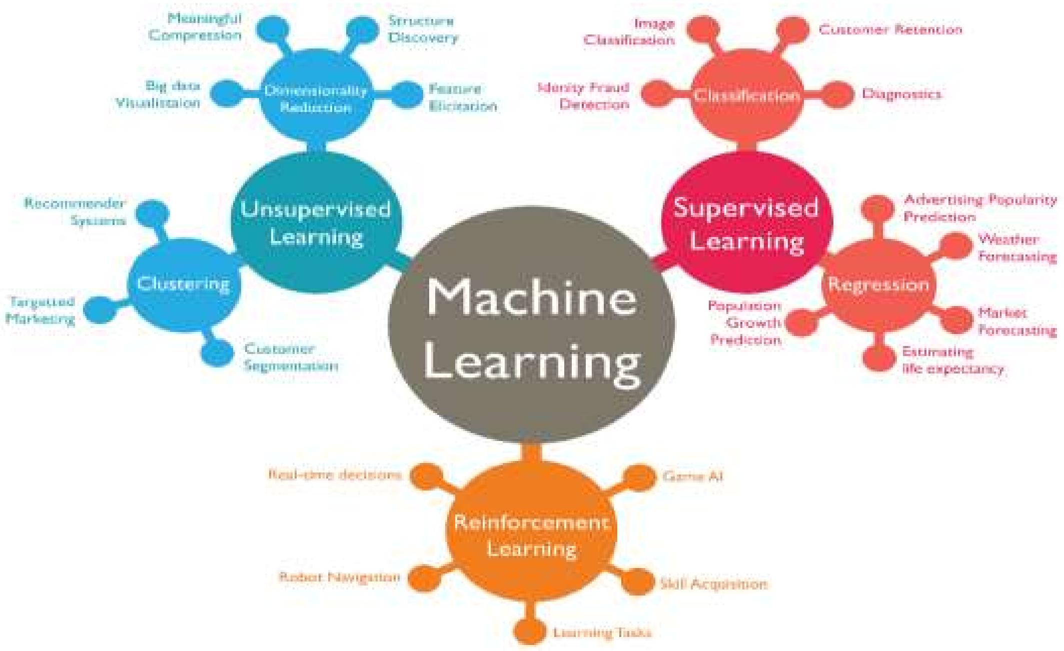


Fig 3.1 Types of Machine Learning Algorithms

Supervised Learning Algorithm

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process.

Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.

Supervised learning can be separated into two types of problems when data mining—classification and regression:

- Classification

  Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest, which are described in more detail below.

- Regression

  Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

UnSupervised Learning Algorithm

Unsupervised machine learning and supervised machine learning are frequently discussed together. Unlike supervised learning, unsupervised learning uses unlabeled data. From that data, it discovers patterns that help solve for clustering or association problems. This is particularly useful when subject matter experts are unsure of common properties within a data set. Common clustering algorithms are hierarchical, k-means, and Gaussian mixture models.

- Clustering

  Clustering is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

  Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

3.3 Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google’s search algorithm.

3.4 CONVULUTIONAL NEURAL NETWORKS

Convolutional neural networks now provide a more scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image. That said, they can be computationally demanding, requiring graphical processing units (GPUs) to train models.

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

S.NO	CNN Layers
1.	Convolutional layer
2.	Pooling Layer
3.	Fully connected Layer

Table 3.1 CNN layers used

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

Convolutional Layer

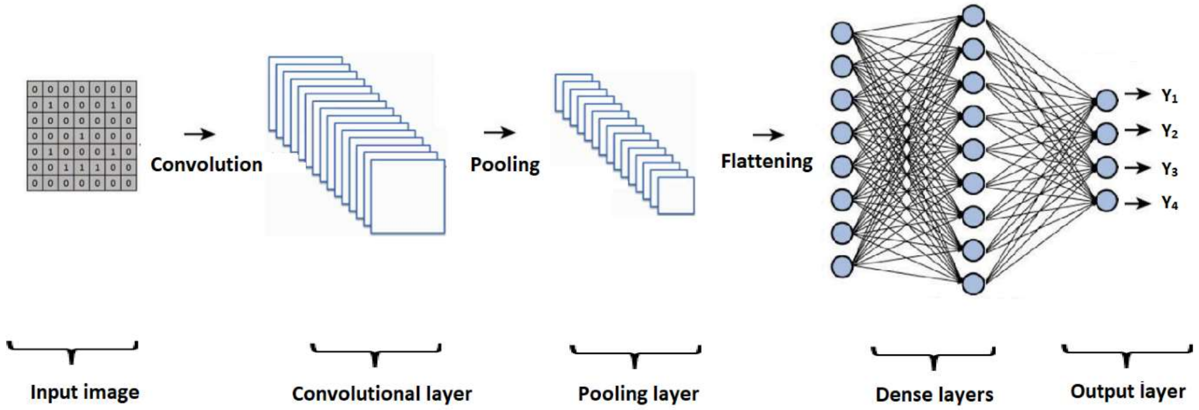
The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

Pooling Layer

Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Fig 3.2 Representation of the architecture of a convolutional neural network (CNN)

Fully-Connected Layer

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

3.5 FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[2] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

3.6 IBM CLOUD FOR GARBAGE CLASSIFICATION

IBM Cloud offers a range of tools and services that can be used for garbage classification. For example, the Watson Visual Recognition service can be trained to recognize different types of waste materials based on images.In addition, the Watson Natural Language Understanding service can be used to analyze text-based information about waste materials, such as product labels or safety data sheets.

There are several benefits to using IBM Cloud for garbage classification. Firstly, it allows for more accurate and efficient sorting of waste materials, which can save time and resources.Secondly, it can help to reduce the risk of human error in the classification process, ensuring that waste is disposed of safely and responsibly.

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULT

The current prototype is capable of picking up 100-200gms of garbage. A future version of the machine is being designed keeping in mind to pick up the garbage of up to 2-3Kgs in weight. Whereas, the number of garbage that can be detected by the machine can be easily increased either by training a new Convolutional Neural Network from scratch, or by using techniques like transfer learning. The machine can be further connected to the internet and using it as an IoT device.

4.2 PERFORMANCE OF THE MODEL

Following are the outputs which are presented in the screenshot format.

1.HOME PAGE

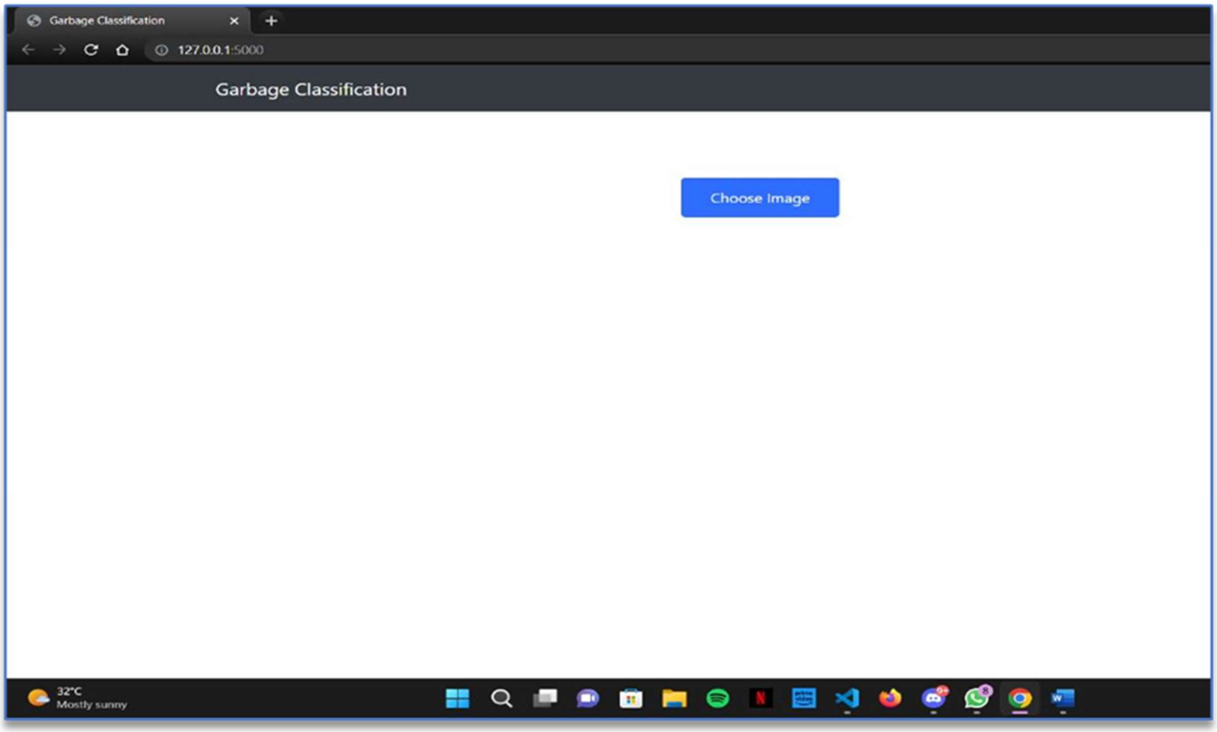


Fig 4.1 Home Page

2.CLICK ON CHOOSE IMAGE

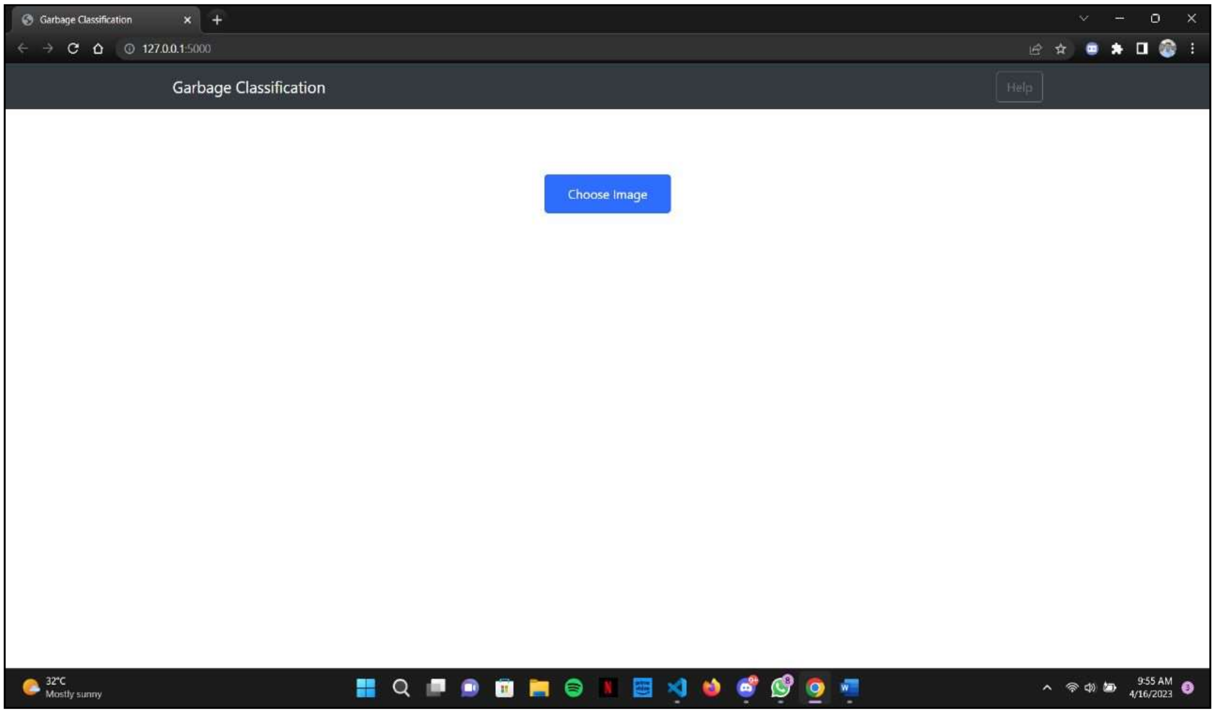


Fig 4.2 Choosing Image

3.SELECTING THE IMAGE

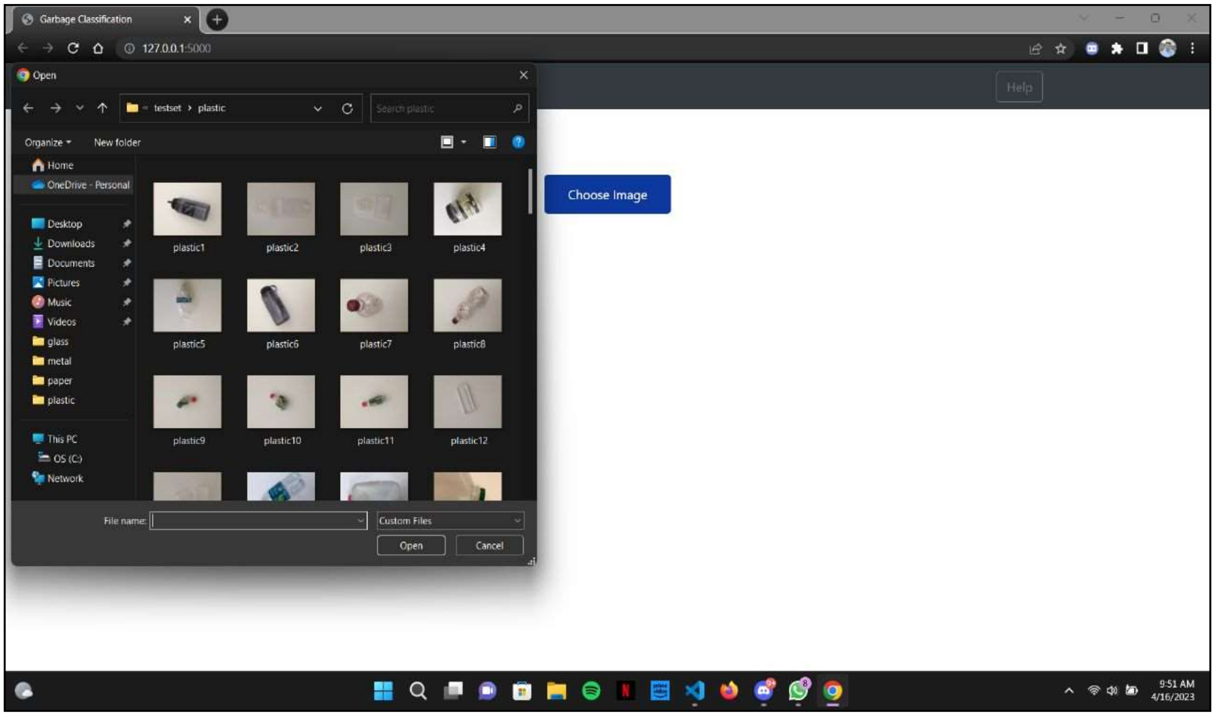
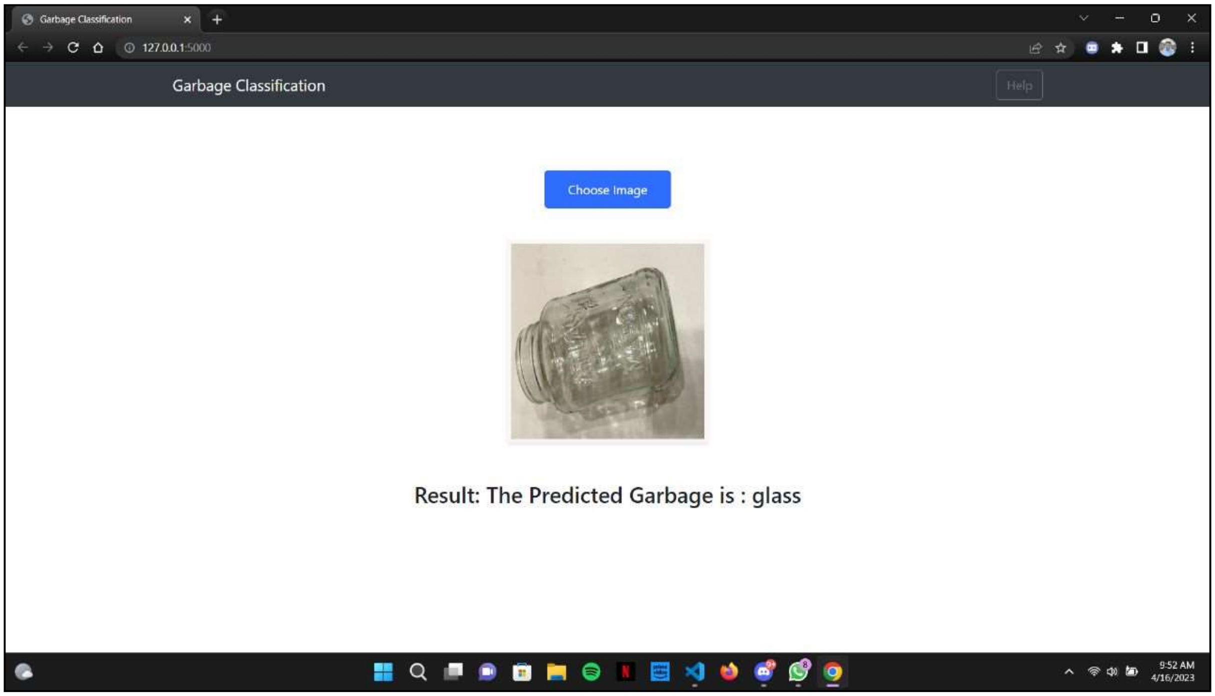


Fig 4.3 Select the image

4.FOLLOWING IS A PREDICTED OUTPUT WHICH IS A CLASSIFIED RESULT



Result: The Predicted Garbage is : glass

Fig 4.4 Predicted Output

CHAPTER-5

SUMMARY AND CONCLUSIONS

5.1 CONCLUSION

By using the model that we developed in this project we can classify the garbage based on its attributes and parameters and make the process of classification more efficient and robust. A machine exposed to harmful garbage yards can be used efficiently as it is robust and has nothing to do with the harmful and toxic environment of dump yards. Whereas if humans are exposed to a harmful and toxic environment will encounter multiple severe diseases in the long run and the healthy life of a human cannot be retrieved at any cost. So we can conclude by saying that “Human life can’t be replaced but a machine can be replaced.”

In conclusion, we 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, we got an accuracy of 87%. The separation process of the waste will be faster and intelligent using our system without or reducing human involvement. If more image is added to the dataset, the system accuracy can be improved In the future, we will tend to improve our system to be able to categories more waste item, by turning some of the parameters used.

REFERENCES

[1] The datasets are collected from kaggle.com, data.gov, the UCI machine learning repository, etc.

[2] Image Data Generator Reference
https://keras.io/api/preprocessing/image/

[3] You can download the dataset used in this project using the GitHub link https://github.com/Guided-Projects/Garbage-Classification

[4] Load data from the directory reference
https://keras.io/api/preprocessing/image/ - imagedatasetfromdirectory-function

[5] Convolution Neural Networks reference
Convolutional Neural Network (CNN) | Convolutional Neural Networks With TensorFlow | Edureka - YouTube

[6] For information regarding CNN layers refer to this link https://victorzhou.com/blog/intro-to-cnns-part-1/

[7] Flask app reference
https://www.youtube.com/watch?v=lj4I_CvBnt0

[8] HTML reference
https://www.w3schools.com/html/

APPENDIX

SOURCE CODE

```
from _future_ import division, print_function
# coding=utf-8
import sys
import os
import glob
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout

from tensorflow.keras.applications.imagenet_utils import preprocess_input,
decode_predictions

from tensorflow.keras.models import load_model
from tensorflow.keras import backend
from tensorflow.keras import backend
from tensorflow import keras
import tensorflow as tf

# global graph
from skimage.transform import resize

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(_name_)

# Load your trained model
model = load_model(r'..\models\garbage1.h5')
    # Necessary
# print('Model loaded. Start serving...')
```

```
@app.route('/',methods=['POST','GET'])
def prediction(): # route which will take you to the prediction page
    return render_template('base.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':

        f = request.files['image']


        basepath = os.path.dirname(_file_)
        file_path = os.path.join(
            basepath, 'predictions',f.filename)
        f.save(file_path)
        img = image.load_img(file_path, target_size=(128, 128))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        a=np.argmax(model.predict(x),axis=1)

       # preds = model.predict_classes(x)
        index = ['cardboard','glass','metal','paper','plastic','trash']
        text = "The Predicted Garbage is : "+str(index[a[0]])

        # ImageNet
        return text

        if _name_ == '_main_':
        app.run(debug=False,threaded = False)
```