A Project Report
on

GEO WASTE CLASSIFICATION USING NEURAL NETWORKS

Submitted in partial fulfillment of the requirements for the award of degree of
BACHELOR OF TECHNOLOGY

in
Information Technology

by

D. Tejaswini (20WH1A1291)
B. Akshitha (20WH1A12A0)
Ch. Dharani (20WH1A12B3)

Under the esteemed guidance of
Ms.M.Sudha Rani
Associate Professor

Department of Information Technology

BVRIT HYDERABAD College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

May, 2024

BVRIT HYDERABAD
College of Engineering for Women
Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090
(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

CERTIFICATE

This is to certify that the Project report on **" Geo Waste Classification Using Neural Networks"** is a bonafide work carried out by **D. Tejaswini (20WH1A1291), B. Akshitha (20WH1A12A0)** and **Ch. Dharani (20WH1A12B3)** in the partial fulfillment for the award of B.Tech degree in **Information Technology , BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad** affiliated to Jawaharlal Nehru Technological University, Hyderabad, under my guidance and supervision. The results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide	**Head of the Department**
Ms. M. Sudha Rani	**Dr. Aruna Rao S.L**
Assistant Professor	**Professor & HoD**
Department of IT	**Department of IT**

External Examiner

DECLARATION

We hereby declare that the work presented in this project entitled **"Geo Waste Classification Using Neural Networks"** submitted towards completion in IV year I sem of B.Tech , IT at "BVRIT HYDERABAD College of Engineering for Women", Hyderabad is an authentic record of our original work carried out under the esteemed guidance of **Ms. M. Sudha Rani, Assistant Professor**, Department of Information Technology.

D. Tejaswini (20WH1A1291)

B. Akshitha (20WH1A12A0)

Ch. Dharani (20WH1A12B3)

ACKNOWLEDGMENTS

We would like to express our profound gratitude and thanks to **Dr K.V.N. Sunitha, Principal, BVRIT HYDERABAD College of Engineering for Women** for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. Aruna Rao S.L, Professor & Head, Department of IT, BVRIT HYDERABAD College of Engineering for Women** for all the timely support, constant guidance and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Ms. M. Sudha Rani, Assistant Professor, Department of IT, BVRIT HYDERABAD College of Engineering for Women** for her constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinators **Ms Dr. J.Kavitha , Associate Professor and Mr. Mukthar Sofi, Assistant Professor**, all the faculty and staff of Department of IT who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

D. Tejaswini (20WH1A1291)

B. Akshitha (20WH1A12A0)

Ch. Dharani (20WH1A12B3)

ABSTRACT

The increasing global waste generation has elevated waste management to critical concern. It is observed that globally solid waste has surged, hitting 2.01 billion tons annually in 2016, with predictions of 3.40 billion tons by 2050. Such vast amounts of waste can lead to severe environmental degradation, loss of biodiversity and generation of green house gases which have a long-lasting effect on planet. Existing methods, relying on manual sorting and implementation of IOT. All of the reviewed surveys focus on object detection and a few on waste detection and classification. However, none of them comprehensively surveyed the available benchmarked dataset and the deep learning models for single and multi-object detection on the waste detection and classification. This project proposes deep learning models, specifically convolutional neural networks (CNNs) and Mask regional convolutional neural networks (MRCNNs), to address these challenges and enhance waste management. The proposed method involves training deep learning models on a comprehensive dataset of waste materials, enabling automated and accurate sorting based on visual and textual characteristics. This project's significance lies in its potential to revolutionize waste management by automating classification and recycling processes, reducing human labor, enhancing recycling efficiency, and minimizing waste sent to landfills, thereby reducing environmental pollution and conserving resources. Therefore, it provides the way for smarter and more sustainable waste management practices, contributing to a cleaner and healthier planet.

Keywords: Convoultuional neural network (CNN),Support vector machine (SVM), MobileNetv2, YOLO Architecture, Mask regional convoulutional neural network (MRCNN)

Contents

List of Figures

Chapter 1

Introduction

The development of geo waste classification using deep neural networks is imperative in addressing the pressing need for effective waste management strategies to combat environmental degradation and promote sustainability. The exponential growth in population and urbanization has led to an unprecedented increase in waste generation, necessitating innovative solutions to mitigate the environmental impact of improper disposal. Traditional waste management methods often fall short in accurately categorizing and handling diverse waste types, leading to inefficiencies in recycling processes and environmental pollution. In this context, the implementation of deep neural networks for geo waste classification emerges as a transformative approach. The need arises from the urgency to enhance the accuracy and efficiency of waste classification, particularly on a geographical scale, where distinct waste patterns and compositions may exist. By leveraging the capabilities of deep learning models, the technology can autonomously recognize and categorize waste materials, facilitating optimal resource recovery and recycling. The importance of this endeavor lies in its potential to significantly reduce the environmental footprint of waste disposal, promote resource conservation through improved recycling rates, and create healthier ecosystems. Additionally, geo waste classification using deep neural networks holds promise in fostering community well-being by minimizing the presence of hazardous materials and contributing to the creation of safer living environments. As a technological innovation, it not only showcases the positive impact of artificial intelligence on real-world challenges but also provides educational and advocacy opportunities, raising awareness about responsible waste management practices. This pioneering initiative aligns with global sustainability goals outlined in agendas like the United Nations Sustainable Development Goals, emphasizing responsible consumption, climate action, and the preservation of life on land and below water. In summary, the development of geo waste classification using deep neural networks is not just a response to a crucial environmental need but a pathway towards a more sustainable and resilient future, where technological advancements play a pivotal role in shaping responsible waste management practices.

1.1 Motivation

The development and implementation of geo waste classification through deep neural networks offers a transformative opportunity to tackle environmental issues and foster sustainability. Utilizing cutting-edge technology for waste management on a geographic scale positions you at the forefront of positive change. The motivation encompasses several crucial aspects, including reducing environmental impact by accurately categorizing waste, enabling efficient recycling, and promoting ecosystem health. Proper waste classification contributes to resource conservation, enhancing recycling rates and supporting a more sustainable, circular economy. Addressing waste management challenges at the geographic level positively impacts community health by minimizing hazardous materials and creating safer living environments. The project not only showcases the potential of technology for societal good but also provides educational opportunities, raising awareness about responsible waste management. Accurate waste classification data supports evidence-based policymaking, aligns with global sustainability goals, and positions you as an innovator in waste management, contributing to scalable and efficient solutions for a more sustainable future. In summary, the motivation stems from the positive environmental impact, community well-being, and the chance to pioneer technological innovation in sustainable waste management practices.

1.2 Objective

The objectives of geo waste classification using deep learning are centered on revolutionizing waste management practices. This entails achieving superior accuracy and efficiency compared to traditional methods through the utilization of deep neural networks, thereby automating the classification process and reducing reliance on manual sorting. The integration of spatial and temporal features in geospatial data aims to provide a comprehensive understanding of waste distribution patterns and adapt to dynamic trends across diverse geographic regions. The exploration of multi-modal data fusion facilitates a holistic view of the waste landscape by integrating various data sources. Scalability and adaptability are prioritized through the design of a scalable deep learning architecture capable of handling diverse datasets and waste compositions. Transfer learning techniques are implemented to accelerate training and foster model generalization, ensuring effectiveness in different environmental contexts. The user-friendly interface and real-time monitoring capabilities contribute to actionable insights for decision-makers, while the system's ability to assess environmental impact aids targeted interventions for pollution prevention and resource conservation. The overarching goal is to leverage deep learning for sustainable and effective environmental conservation efforts in waste management.

1.3 Problem Definition

Existing waste classification methods, relying on manual sorting and conventional techniques, face significant drawbacks in modern waste management. Manual sorting is labor-intensive, time-consuming, and error-prone, leading to inaccuracies in categorization. Traditional methods lack scalability and struggle to adapt to dynamic waste patterns across diverse regions. These systems fail to comprehensively analyze spatial and temporal features in geospatial data, limiting real-time insights and hindering automation. The challenges intensify with large datasets and diverse waste compositions, resulting in suboptimal decision-making. The problem statement for geo waste classification using deep learning stems from these limitations, aiming to revolutionize waste management by leveraging deep neural networks. This advanced approach integrates geospatial data, addresses manual sorting issues, enables real-time monitoring, and provides a scalable solution for accurate waste classification, contributing to more sustainable environmental conservation efforts.

1.4 Aim

The Aim of geo waste classification using deep learning is to revolutionize waste management practices by automating and enhancing the categorization of waste based on its composition, and environmental impact. Leveraging advanced deep neural networks, the goal is to overcome the limitations of manual sorting and conventional methods and providing recycling methods of major components in waste. This approach seeks to enable real-time monitoring, adaptability to dynamic waste patterns, and comprehensive analysis of spatial and temporal features in geospatial data. The ultimate objective is to provide a scalable, efficient, and accurate solution for waste classification, contributing to more sustainable and effective environmental conservation efforts on both local and global scales..

Chapter 2

Literature Survey

2.1 Research Papers

2.1.1 Waste Detection and Classification Using Deep Learning

[1] The paper "A Survey on Waste Detection and Classification Using Deep Learning" reveals a growing interest in the application of machine learning and deep learning algorithms in waste management. The studies reviewed provide insights into the classification technology of domestic waste, deep learning-based object detection in challenging environments, and salient object detection in the context of deep learning. Additionally, the survey highlights the potential of deep learning technology to contribute to sustainable development and environmental conservation. Overall, the literature survey underscores the importance of continued research and development in this field to address the challenges of waste management and promote a cleaner, healthier environment.

2.1.2 Illegal Trash Thrower Detection Based on HOGSVM for a Real-Time Monitoring System

[2] The paper, titled Illegal Trash Thrower Detection Based on HOGSVM for a Real-Time Monitoring System" offers a comprehensive overview of intelligent surveillance systems (ISS) and related methodologies for various applications, including the detection of illegal trash littering persons. It discusses the limitations of current systems, such as issues with background subtraction, shadow removal, and the detection of multiple objects. The survey also highlights the use of techniques such as Gaussian mixture model (GMM), histogram of oriented gradients (HOG), and support vector machine (SVM) algorithms in surveillance-based systems. Furthermore, the proposed methodology for illegal trash throwing person

detection is based on GMM, HOG, and SVM algorithms, with a focus on foreground detection using GMM, feature extraction using HOG, and classification using SVM. The system overview includes stages such as region of interest selection, background modeling, foreground detection, and various image processing techniques including shadow removal, background subtraction, morphological operations, and labeling and filtering operations. The survey also references several relevant works in the field, including studies on sterile zone monitoring, unattended object identification, smoke and flame detection, human detection, and background subtraction techniques using GMM. Additionally, it highlights the use of SVM for human detection and the preparation of effective datasets for detecting and segmenting trash from images. Overall, the literature survey provides a comprehensive overview of existing ISS and related methodologies, highlighting their limitations and the proposed approach for illegal trash throwing person detection, while referencing relevant works in the field.

2.1.3 Deep learning applications in solid waste management

[3] The paper ”Deep learning applications in solid waste management” reflects a progressive shift towards automated waste management in response to the inefficiencies of manual waste sorting, this literature survey explores the proposed Multi-model Cascaded Convolutional Neural Network (MCCNN) as an innovative solution for domestic waste image detection and classification. The MCCNN amalgamates three cutting-edge subnetworks (DSSD, YOLOv4, and Faster-RCNN) to enhance detection precision. To address false-positive predictions, a cascaded classification model is incorporated. The survey highlights the creation of the Large-Scale Waste Image Dataset (LSWID), comprising 30,000 multi-labeled images across 52 categories, as the largest dataset in domestic waste image classification. The study further delves into the practical application of the proposed technology through the implementation of a smart trash can (STC) in a Shanghai community, showcasing its potential for scalable and real-world impact. Experimental results demonstrate a state-of the-art performance, with an average 10percent improvement in detection precision. This literature survey positions the MCCNN system as a transformative advancement in waste management, emphasizing its potential for global adoption and implications for sustainable practices.

2.1.4 Drowsiness Detection with OpenCV

[4] The paper titled ”Waste Management Using Machine Learning and Deep Learning Algorithms ” encompasses a comprehensive review of related works in the domain of waste management using machine learning and deep learning algorithms. The survey includes references to various studies that have employed different approaches, such as Support Vector Machine (SVM), Random Forest Classifier, Gaussian Naive Bayes, Multilayer Perceptron, Internet of Things (IoT) technologies, Raspberry Pi, Computer Vision implementation, and transfer learning with lightweight neural networks. These studies have explored diverse methodologies, including the use of IoT devices for waste segregation, the integration of machine learning models with IoT devices and sensors, and the implementation of computer vision techniques for waste classification. Additionally, the survey highlights the application of transfer learning to address overfitting issues and the use of correlation coefficients for feature analysis. The literature survey provides a rich overview of the existing research landscape and the various technological approaches employed in the field of waste management using machine learning and deep learning algorithms

2.1.5 Reach on waste classification and identification by transfer learning and lightweight neural network

[5] In this paper, the authors addresses the critical issue of waste recognition and classification using machine learning and deep learning techniques within the field of computer vision. The study highlights challenges such as incomplete garbage datasets and poor performance of complex models on smart devices, leading to suboptimal results in existing garbage classification methods. The authors propose an innovative waste classification and identification approach based on transfer learning and a lightweight neural network. By adapting and enhancing the MobileNetV2 model for feature extraction and integrating support vector machines (SVM) for classification, the method achieves a remarkable 98.4 percent accuracy on a dataset of 2527 labeled garbage items. The research demonstrates the effectiveness of the proposed method in improving classification accuracy, addressing challenges related to limited data, and overcoming overfitting issues in small datasets commonly encountered in deep learning. The study contributes valuable insights and methodologies to the evolving field of garbage recognition, emphasizing the importance of artificial intelligence in waste management.

2.1.6 A Non-intrusive Approach for Driver's Drowsiness Detection

[6] The paper provides valuable insights into the field of waste management and Internet-of-Things (IoT) technology. It discusses the potential of IoT in improving waste management systems , . Additionally, it references works such as "Waste Management in IoT-Enabled Services Smart Cities" [7] and "Garbage monitoring system using IoT" [8], which further explore the application of IoT in waste management. The survey also highlights the significance of machine learning, monitoring systems, and sensors in the context of waste management . Furthermore, it emphasizes the growing global concern regarding waste management due to the increasing population and the associated challenges, as evidenced by the impact of typhoon Ondoy in the Philippines . This literature survey provides a comprehensive overview of the current state of waste management and the potential for IoT technology to address these challenges.

2.1.7 Intelligent Waste Sorting Using Deep Neural Networks

[7] The paper "RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks" explores the application of deep learning techniques to waste management and recycling, emphasizing the need for intelligent systems over human labor in dump-yards for more efficient and safe recycling. The study evaluates various deep convolutional neural network architectures, including ResNet50, MobileNet, Inception-ResNetV2, DenseNet121, and Xception, experimenting with both training from scratch and transfer learning approaches. The authors introduce a novel architecture called RecycleNet, specifically optimized for the classification of recyclable objects. RecycleNet achieves an 81 percent test accuracy on a dataset of six common recyclable materials. The research compares different optimization methodologies, such as Adam and Adadelta, and explores the impact of data augmentation on model performance. The article also addresses the real-time implementation potential of the models, crucial for applications like smart bin systems. The study concludes that deep learning, particularly with the proposed RecycleNet model, demonstrates feasibility in waste sorting, showcasing promising results for ecological awareness and sustainable waste management. Future work is suggested to refine RecycleNet further and explore advancements in convolutional neural networks for deformations and adversarial examples.

2.1.8 Multilayer hybrid deep-learning method for waste classification

[8] The paper "Multilayer hybrid deep-learning method for waste classifica-

tion and recycling" explained a multilayer hybrid deep-learning system (MHS) to automatically sort waste disposed of by individuals in the urban public area. This system deploys a high-resolution camera to capture waste image and sensors to detect other useful feature information. The MHS uses a CNN-based algorithm to extract image features and a multilayer perceptrons (MLP) method to consolidate image features and other feature information to classify wastes as recyclable or the others. The MHS is trained and validated against the manually labelled items, achieving overall classification accuracy higher than 90 percent under two different testing scenarios, which significantly outperforms a reference CNN-based method relying on image-only inputs

2.1.9 Artificial intelligence in automated sorting in trash recycling

[9] The paper "Artificial intelligence in automated sorting in trash recycling" The related work section emphasizes the global problem of uncontrolled garbage disposal, leading to health risks and environmental impact. Various technologies, such as Radio Frequency Identification (RFID) and Sensor Networks (SN), have been explored to optimize waste management. Previous studies focused on identifying, tracking, and analyzing discarded garbage using RFID technology. Some proposed methods aimed to estimate household waste volume based on image analysis and associate each bin with the house's address using RFID tags. However, the text highlights that none of these studies specifically addressed assisting the population in the correct disposal of garbage. It then introduces works that aimed to help consumers properly discard waste, such as using Ontology Web Language (OWL) for smart waste sorting and classifying garbage into recycling categories using support vector machines (SVM) and convolutional neural networks (CNN). The proposed project extends these works by testing different neural network techniques for waste classification.

2.1.10 Friendly waste segregation using deep learning

[10] The paper encompasses a wide range of studies and resources relevant to waste management, deep learning, and agricultural technology. It includes works such as "An Adaptive Approach of Tamil Character Recognition Using Deep Learning with Big Data" by R. Jagadeesh Kanan and S. Subramanian, "Automatic Detection and Classification of buried objects in GPR images using Genetic algorithms and Support vector machines" by Edoardo Pasolli, Farid Melgani, Massimo Donelli, Redha Attoui, Merieete De vos, and "An introduction to Genetic Algorithms" by Melanie Mitchell. Furthermore, it references government docu-

ments such as ”e-Waste in India” by the Rajya Sabha Secretariat and ”The Gazette of India” regarding ”The prohibition of employment as Manual Scavengers and their rehabilitation Act, 2013” . These diverse sources provide a comprehensive overview of the existing research and policies in the field, offering valuable insights for the development of the proposed automatic waste segregation system.

Chapter 3

System Design

3.1 Proposed System

The proposed system, named ”Geo Waste Classification Using Neural Networks” integrates various algorithms and models for detecting waste and identifies the efficient model for detecting waste and provides recycling methods for the detected components in waste.

3.2 System Architecture

The building system architecture of the Geo Waste Classification project is designed to efficiently process and analyze waste data for effective recycling recommendations. The project begins with a diverse and comprehensive dataset containing Geo spatial information on waste types. The data pre-processing stage involves cleaning, normalization, and feature engineering to ensure the quality and relevance of the dataset. The core of the system is the machine learning model, trained to detect and classify waste based on its geographical context. This model utilizes advanced algorithms and neural networks to make accurate predictions. The detection and prediction component leverages spatial data to provide real-time insights into waste composition and distribution. Finally, the system includes a recommendation module that suggests optimal recycling methods based on the waste classification, promoting sustainable waste management practices. Overall, the architecture integrates data processing, machine learning, and recommendation systems to enhance waste classification and contribute to more effective recycling strategies.

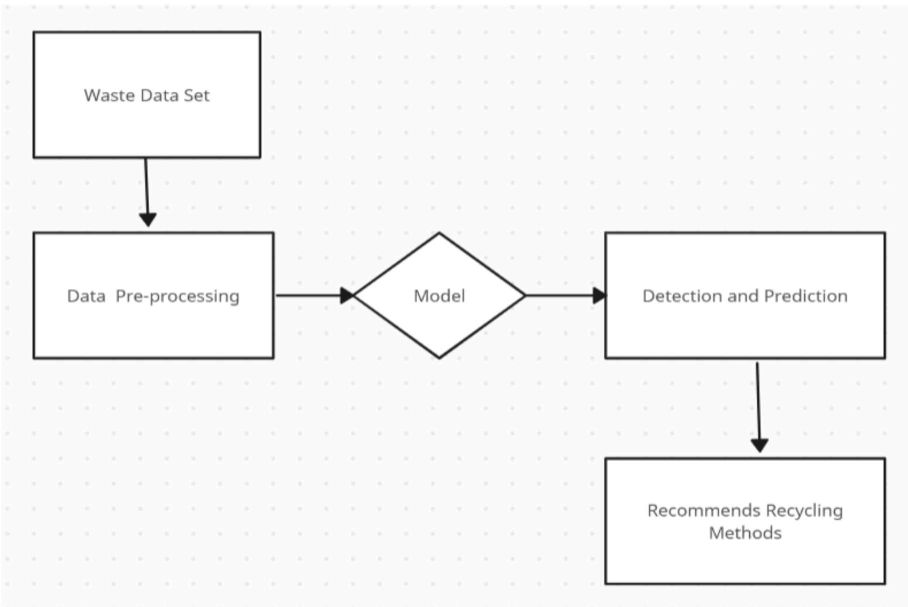

Figure 4.1.3: System Architecture

Chapter 4

Methodology

4.1 Algorithms used

4.1.1 SVM Algorithm

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. Its primary goal is to find the optimal hyperplane that best separates classes in a high-dimensional space. In classification, SVM aims to create a decision boundary (hyperplane) that maximizes the margin, which is the distance between the closest data points of different classes known as support vectors. These support vectors are the critical data points that influence the position and orientation of the hyperplane. SVM can handle linear and nonlinear data by using different kernel functions (like polynomial, radial basis function, etc.) that map the data into higher-dimensional spaces where classes are more separable. For regression tasks, SVM employs a similar principle by finding a hyperplane that best fits the data within a certain margin of tolerance. It aims to minimize errors while maximizing the margin, thus finding the optimal fit for the given data. SVM is effective in handling high-dimensional data, works well with a clear margin of separation between classes, and is robust against over fitting when the right parameters are chosen. However, it can be sensitive to noise and might become computationally expensive with large data sets.

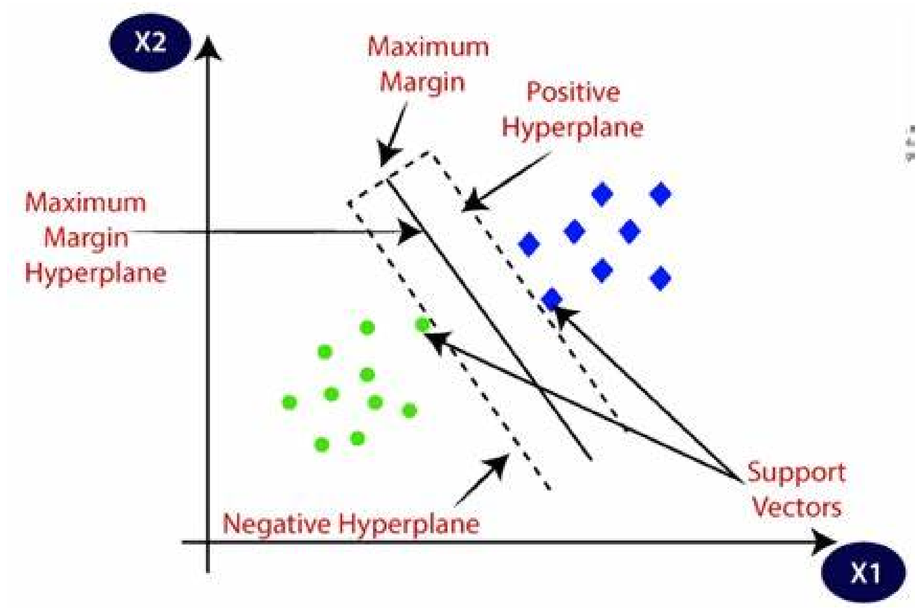

Figure 4.1: SVM Algorithm

4.1.2 CNN Algorithm

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily used for analyzing visual imagery in applications such as image recognition, object detection, and image classification. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from the input data. They consist of multiple layers including convolutional layers, pooling layers, and fully connected layers.

key components of cnn are:

- Input Layer
- Conventional layers (Conv2D)
- Batch normalization layers (Batch Normalization)
- Max Pooling
- Flatten Layer
- Output Layer

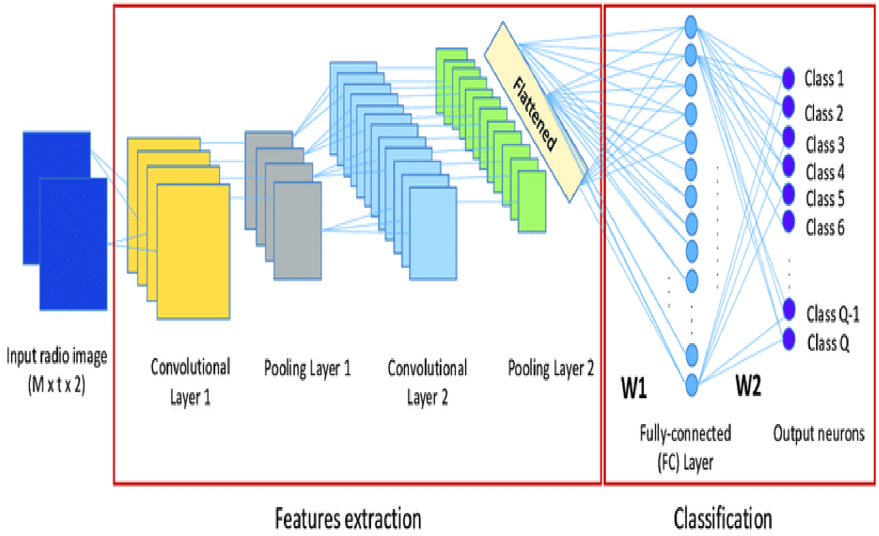

Figure 4.1.3: CNN Algorithm

Input layer: This layer takes in an image as input. In your case, the input shape is (224, 224, 3), which means the image is 224 pixels wide, 224 pixels tall, and has 3 channels (one for each color: red, green, and blue).

Convolutional layers (Conv2D): These layers extract features from the image by convolving it with a small filter. The first convolutional layer in your model has 32 filters, each of which is 3x3 pixels in size. This means that the layer will extract 32 different features from the image. The second convolutional layer has 64 filters, and so on.

Batch normalization layers (BatchNormalization): These layers help to stabilize the training process by normalizing the activations of the previous layer.

Pooling layers (MaxPooling2D and AveragePooling2D): These layers reduce the size of the feature maps by taking the maximum or average value of a small region of the input. This helps to reduce the number of parameters in the network and prevent overfitting.

Flatten layer: This layer converts the feature maps into a one-dimensional vector.

Fully-connected layers (Dense): These layers learn complex relationships between the features extracted by the convolutional layers. The first fully-connected layer in your model has 2056 neurons, the second layer has 512 neurons, and the third layer has 256 neurons.

Output layer: This layer outputs the predictions of the network. In your case, the output layer has two neurons, one for each class in your classification task.

4.1.3 MobileNet

MobileNet is a family of neural network architectures designed for efficient on-device image classification and computer vision tasks, particularly optimized for mobile and embedded devices with limited computational resources. Key features include the use of depthwise separable convolutions to reduce computational cost, compact model sizes for faster inference on mobile devices, and variants such as MobileNetV1, V2, and V3, each improving accuracy, efficiency, and speed. MobileNet models find widespread applications in tasks like image classification, object detection, and semantic segmentation, making them suitable for real-time applications on smartphones and IoT devices. Supported by popular deep learning frameworks like TensorFlow and PyTorch, MobileNet provides flexibility for a diverse range of developers, with the choice of a specific variant based on factors such as computational resources, accuracy requirements, and latency constraints.

4.2 YOLO Architecture

YOLO (You Only Look Once) is a real-time object detection system that predicts bounding boxes and class probabilities for multiple objects in an image. Introduced by Joseph Redmon and Santosh Divvala, YOLO divides the image into a grid, applying detection at each cell to predict bounding box coordinates, object confidence, and class probabilities. Its one-stage approach achieves high accuracy and real-time speeds, different from traditional two-stage detectors. YOLOv2 and YOLOv3 improve accuracy, speed, and small object handling. Utilizing a neural network backbone, often based on Darknet, and techniques like anchor boxes for localization, YOLO is widely used in applications such as surveillance, autonomous vehicles, and robotics for efficient real-time object detection.

4.2.1 Implementation of YOLO Architecture

Implementing the YOLO (You Only Look Once) architecture involves several steps, and it typically requires a deep learning framework like TensorFlow or PyTorch. Below, I provide a high-level overview of the steps involved in implementing YOLO using TensorFlow:

Set up Environment: Installing the required libraries, which includes TensorFlow, NumPy, and any other dependencies.

Dataset Preparation: Collecting a dataset for training. Annotated images with bounding box coordinates and class labels are necessary.

Model Architecture: Defining the YOLO model architecture. The architecture typically involves a convolutional neural network (CNN) backbone, detection head, and output layer for bounding box predictions and class probabilities.

Loss Function: Implementing a custom loss function that combines localization loss (for bounding box coordinates) and classification loss. YOLO uses a combination of mean squared error and binary cross-entropy.

Training: Training the YOLO model on the prepared dataset using stochastic gradient descent or an optimizer of choice. Transfer learning with pre-trained weights on a large dataset like ImageNet is common for improved performance.

Post Preprocessing: Implementing post-processing steps to filter out low-confidence detections and apply non-maximum suppression to remove redundant bounding boxes.

Evaluation: Evaluating the performance of the YOLO model on a separate validation set to assess accuracy, precision, recall, and other relevant metrics.

4.3 MRCNN

Mask R-CNN (Mask Region-based Convolutional Neural Network) is a sophisticated deep learning model designed for the task of instance segmentation. Building upon the Faster R-CNN framework, Mask R-CNN introduces an additional branch dedicated to predicting segmentation masks alongside the existing components for object detection and bounding box regression. This allows the model not only to accurately identify and locate objects within an image but also to provide pixel-level segmentation, outlining the precise boundaries of each detected object. Mask R-CNN's capability for detailed instance segmentation makes it a powerful tool in computer vision applications, such as image segmentation, object recognition, and scene understanding, offering high accuracy in delineating and understanding complex visual scenes.

Chapter 5

Implemention

5.1 SVM

```
import os
import numpy as np
from skimage import io, transform
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.feature_extraction import image
train_folder = '/content/drive/MyDrive/waste project/data/train'
test_folder = '/content/drive/MyDrive/waste project/data/test'
validate_folder = '/content/drive/MyDrive/waste project/data/valid'
def extract_features_from_folder(folder):
    features = []
    labels = []
    for label in os.listdir(folder):
        label_folder = os.path.join(folder, label)
        for filename in os.listdir(label_folder):
            img_path = os.path.join(label_folder, filename)
            img = io.imread(img_path)
            img = transform.resize(img, (100, 100))
            feature = np.ravel(img)
            features.append(feature)
            labels.append(label)
    return np.array(features), np.array(labels)
X_train, y_train = extract_features_from_folder(train_folder)
X_test, y_test = extract_features_from_folder(test_folder)
X_validate, y_validate = extract_features_from_folder(validate_folder)
```

Figure 5.1.1: importing libraries of svm

```
svm_model = make_pipeline(StandardScaler(), svm.SVC(kernel='linear',
C=1.0))
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on test set: {accuracy}")
param_grid = {'svc__C': [0.1, 1, 10], 'svc__kernel': ['linear', 'rbf']}
grid_search = GridSearchCV(svm_model, param_grid, cv=5)
grid_search.fit(X_train, y_train)
best_params = grid_search.best_params_
print("Best parameters:", best_params)
y_validate_pred = grid_search.predict(X_validate)
validate_accuracy = accuracy_score(y_validate, y_validate_pred)
print(f"Accuracy of SVM Model: {validate_accuracy* 100:.2f}%")
```

Figure 5.1.2: Traninig and prediction by svm

5.2 CNN

5.3 MobileNetv2

5.4 YOLO Architecture

5.5 MRCNN

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
img_width, img_height = 100, 100
batch_size = 32
train_datagen = ImageDataGenerator(
    rescale=1./255,
    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'
)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
    train_folder,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical')  # Change to 'categorical' for multiple
classes

test_generator = test_datagen.flow_from_directory(
    test_folder,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical')  # Change to 'categorical' for multiple
classes
```

Figure 5.2.1: Data preprocessing by CNN

```
validate_generator = test_datagen.flow_from_directory(
    validate_folder,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical')
output_units = 6
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(img_height, img_width, 3)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(output_units, activation='softmax')  # Output
layer for multi-class classification
])
model.compile(optimizer='adam',
```

Figure 5.2.2: Training the model by CNN

```
              loss='categorical_crossentropy',  # Use categorical
crossentropy for multi-class
              metrics=['accuracy'])
history = model.fit(train_generator,
                    steps_per_epoch=train_generator.samples //
batch_size,
                    epochs=30,
                    validation_data=validate_generator,
                    validation_steps=validate_generator.samples //
batch_size)
test_accuracy = model.evaluate(test_generator,
steps=test_generator.samples // batch_size)

print(f"Test Accuracy for CNN Model: {test_accuracy[1] * 100:.2f}%")
```

Figure 5.2.3: Evaluating the model by CNN

```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, classification_report
import numpy as np

 train_datagen = ImageDataGenerator(

rescale=1./255,

    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'
)
val_datagen = ImageDataGenerator(rescale=1./255)
train_data_dir = '/content/drive/MyDrive/waste project/data/train'
val_data_dir = '/content/drive/MyDrive/waste project/data/valid'
test_data_dir = '/content/drive/MyDrive/waste project/data/test'
batch_size = 32
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(224, 224),
    batch_size=batch_size,
    class_mode='categorical'
)
```

Figure 5.3.1: Loading and preprocessing data

```
validation_generator = val_datagen.flow_from_directory(
    val_data_dir,
    target_size=(224, 224),
    batch_size=batch_size,
    class_mode='categorical'
)
test_generator = val_datagen.flow_from_directory(
    test_data_dir,
    target_size=(224, 224),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)
base_model = tf.keras.applications.MobileNetV2(
    weights='imagenet',
    include_top=False,
    input_shape=(224, 224, 3)
)
base_model.trainable = False
model = tf.keras.Sequential([
```

Figure 5.3.2: Model building

```
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(6, activation='softmax']
])
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples
    epochs=30,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples
)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
model.save('/content/drive/MyDrive/waste
project/waste_classification_mobilenetv2.pb')
model = tf.keras.models.load_model('/content/drive/MyDrive/waste
project/waste_classification_mobilenetv2.h5')
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)
true_classes = test_generator.classes
conf_matrix = confusion_matrix(true_classes, predicted_classes)
```

Figure 5.3.3: Model Evaluation

```
conf_matrix = confusion_matrix(true_classes, predicted_classes)
class_names = ['Plastic', 'Glass', 'Metal', 'Paper', 'Cardboard',
'trash']
print("\nClassification Report:")
print(classification_report(true_classes, predicted_classes,
target_names=class_names))
import matplotlib.pyplot as plt
import seaborn as sns
class_names = ['Plastic', 'Glass', 'Metal', 'Paper', 'Cardboard',
'trash']
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='d',
xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
import cv2
from google.colab.patches import cv2_imshow
```

Figure 5.3.4: Plotting the graphs

```
def preprocess_image(image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (224, 224))
    img = img / 255.0
    img = np.expand_dims(img, axis=0)
    return img
def predict_and_visualize(image_path):
    img = preprocess_image(image_path)
    predictions = model.predict(img)
    class_labels = ['plastic', 'glass', 'metal', 'paper', 'cardboard',
'trash']
    predicted_class = np.argmax(predictions)
    confidence = np.max(predictions) * 100
    image = cv2.imread(image_path)
    image = cv2.resize(image, (224, 224))
    cv2.putText(image, f'{class_labels[predicted_class]} -
{confidence:.2f}%', (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0),
2)
cv2.rectangle(image, (0, 0), (image.shape[1], image.shape[0]), (0, 255,
0), 3)
    cv2_imshow(image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
image_path_to_test = '/content/drive/MyDrive/waste
project/data/valid/plastic/plastic121.jpg'
predict_and_visualize(image_path_to_test)
```

Figure 5.3.5: confusion matrix and classification report

```
!nvidia-smi
!pip install ultralytics
from ultralytics import YOLO
import os
from IPython.display import display, Image
from IPython import display
display.clear_output()
!yolo mode=checks
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="z8OtJ6cNQH7JbWYbNaBX")
project = rf.workspace("dark-mqa4m").project("waste-segregation-3ykjs")
dataset = project.version(1).download("yolov8")
!yolo task=detect mode=train model=yolov8m.pt data=/content/waste-
segregation-1/data.yaml epochs=20 imgsz=416
Image(filename=f'/content/runs/detect/train5/confusion_matrix.png',
width=600)
Image(filename=f'/content/runs/detect/train5/results.png', width=600)
!yolo task=detect mode=val
model=/content/runs/detect/train5/weights/best.pt
data={dataset.location}/data.yaml
!yolo task=detect mode=predict
model=/content/runs/detect/train5/weights/best.pt conf=0.5
source={dataset.location}/test/images save_txt=true save_conf=true
import glob
from IPython.display import Image, display

for image_path in glob.glob(f'/content/runs/detect/predict/*.jpg'):
      display(Image(filename=image_path, height=600))
      print("\n")
```

Figure 5.4.1: YOLO Architecture

```
!pip list | grep tensorflow
!git clone --depth 1 https://github.com/tensorflow/models
%%bash
apt-get install -y protobuf-compiler
%%bash
cd models/research
protoc object_detection/protos/*.proto --python_out=.
cp object_detection/packages/tf2/setup.py .
python -m pip install .
!pip list | grep tensorflow
!pip install tensorflow==2.10.0
import os
import pathlib
import cv2
import logging
logging.disable(logging.WARNING)
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from six import BytesIO
from PIL import Image
from six.moves.urllib.request import urlopen
import tensorflow as tf
import tensorflow_hub as hub
tf.get_logger().setLevel('ERROR')
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.utils import ops as utils_ops
%matplotlib inline
!wget
```

Figure 5.5.1: importing libraries

```
!wget
https://storage.googleapis.com/tf_model_garden/vision/waste_identificat
ion_ml/material_model.zip
!wget
https://storage.googleapis.com/tf_model_garden/vision/waste_identificat
ion_ml/material_form_model.zip
%%bash
mkdir material material_form
unzip material_model.zip -d material/
unzip material_form_model.zip -d material_form/
ALL_MODELS ={
'material_model' : 'material/saved_model/saved_model',
'material_form_model': 'material_form/saved_model/saved_model/',
}
IMAGES_FOR_TEST ={
'Image1':'models/official/projects/waste_identification_ml/pre_processi
ng/config/sample_images/image_2.png'
def normalize_image(image,
                    offset=(0.485, 0.456, 0.406),
```

Figure 5.5.2: Loading dataset

```
                    scale=(0.229, 0.224, 0.225)):
                    with tf.name_scope('normalize_image'):
                      image=tf.image.convert_image_dtype(image,
dtype=tf.float32)
                      offset = tf.constant(offset)
                      offset= tf.expand_dims(offset, axis=0)
                      offset= tf.expand_dims(offset, axis=0)
                      image-=offset
                      scale= tf.constant(scale)
                      scale=tf.expand_dims(scale, axis=0)
                      scale=tf.expand_dims(scale, axis=0)
                      image/=scale
                      return image
def load_image_into_numpy_array(path):
  image=None
  if(path.startswith('http')):
    response= urlopen(path)
    image_data =response.read()
    image_data=BytesIO(image_data)
    image =Image.open(image_data)
  else:
    image_data = tf.io.gfile.GFile(path, 'rb').read()
    image=Image.open(BytesIO(image_data))
  (im_width, im_height) =image.size
  return np.array(image.getdata()).reshape((1,  im_height, im_width,
3)).astype(np.uint8)
def build_inputs_for_segmentation(image):
  image= normalize_image(image)
  return image
model display name='material model'
```

Figure 5.5.3: Data preprocessing

```
conf_matrix = confusion_matrix(true_classes, predicted_classes)
class_names = ['Plastic', 'Glass', 'Metal', 'Paper', 'Cardboard',
'trash']
print("\nClassification Report:")
print(classification_report(true_classes, predicted_classes,
target_names=class_names))
import matplotlib.pyplot as plt
import seaborn as sns
class_names = ['Plastic', 'Glass', 'Metal', 'Paper', 'Cardboard',
'trash']
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='d',
xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
import cv2
from google.colab.patches import cv2_imshow
```

Figure 5.5.4: Data preprocessing

```
 3: {'id': 3,
  'name': 'Rubber and leather products'},
 4: {'id': 4,
  'name': 'Wood products '},
 5: {'id': 5,
  'name': ' food waste'},
 6: {'id': 6,
  'name': 'Plastic '},
 7: {'id': 7,
  'name': 'bio waste'},
 8: {'id': 8,
  'name': 'Cardboard products'},
 9: {'id': 9,
  'name': 'Glass products '},
 10: {'id': 10,
  'name': 'metal waste'}}
print(labels)

category_index.update(labels)

print(category_index)
import tensorflow_hub as hub
print('loading model...')
hub_model = hub.load(model_handle)
print('model loaded!')
selected_image = 'Image1'
image_path=IMAGES_FOR_TEST[selected_image]
image_np= load_image_into_numpy_array(image_path)
print('min:', np.min(image_np[0]), 'max:', np.max(image_np[0]))
plt.figure(figsize=(10,10))
```

Figure 5.5.5: Loading the model

```
plt.imshow(image_np[0])
plt.show()
hub_model_fn = hub_model.signatures["serving_default"]
height = hub_model_fn.structured_input_signature[1]['inputs'].shape[1]
width = hub_model_fn.structured_input_signature[1]['inputs'].shape[2]
input_size=(height,width)
print(input_size)
!pip install tensorflow-addons
image_np_cp = cv2.resize(image_np[0],input_size[::-
1],interpolation=cv2.INTER_AREA)
image_np=build_inputs_for_segmentation(image_np_cp)
image_np = tf.expand_dims(image_np,axis=0)
image_np.get_shape()
plt.figure(figsize=(10,10))
plt.imshow(image_np[0])
plt.show()
results = hub_model_fn(image_np)
result = {keys:value.numpy() for keys,value in results.items()}
print(result.keys())
```

Figure 5.5.6: Evaluation of model

```
label_id_offset =0
min_score_thresh =0.6
use_normalized_coordinates=True
if use_normalized_coordinates:
  result['detection_boxes'][0][:,[0,2]]/=height
  result['detection_boxes'][0][:,[1,3]]/=width
if 'detection_masks' in result:
  detection_masks = tf.convert_to_tensor(results['detection_masks
  detection_boxes = tf.convert_to_tensor(results['detection_boxes

  detection_masks_reframed =
utils_ops.reframe_box_masks_to_image_masks(
      detection_masks,detection_boxes,
      image_np.shape[1],image_np.shape[2])
  detection_masks_reframed=tf.cast(detection_masks_reframed>0.5,np
8)

  result['detection_mask_reframed']=detection_masks_reframed.numpy
viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_cp,
    result['detection_boxes'][0],
    (result['detection_classes'][0]+label_id_offset).astype(int),
    result['detection_scores'][0],
    category_index=category_index,
    use_normalized_coordinates=use_normalized_coordinates,
    max_boxes_to_draw=200,
    min_score_thresh = min_score_thresh,
    agnostic_mode =False,
    instance_masks = result.get('detection_masks_reframed',None),
    line_thickness=5
```

Figure 5.5.7: Prediction by mode

Chapter 6

Results and Discussions

Figure 6.1.2: prediction result by YOLO architecture

Figure 6.1.2: Input given to MRCNN

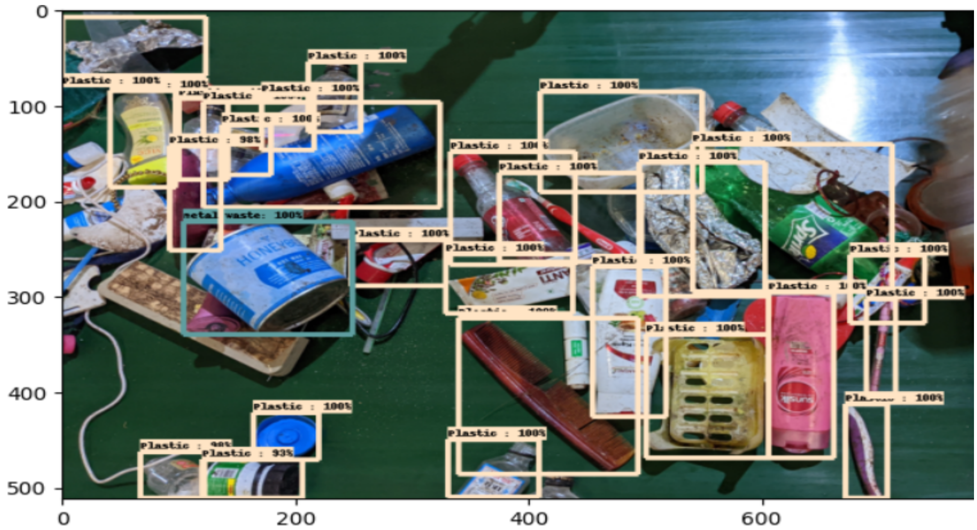

Figure 6.1.2: Prediction result by MRCNN

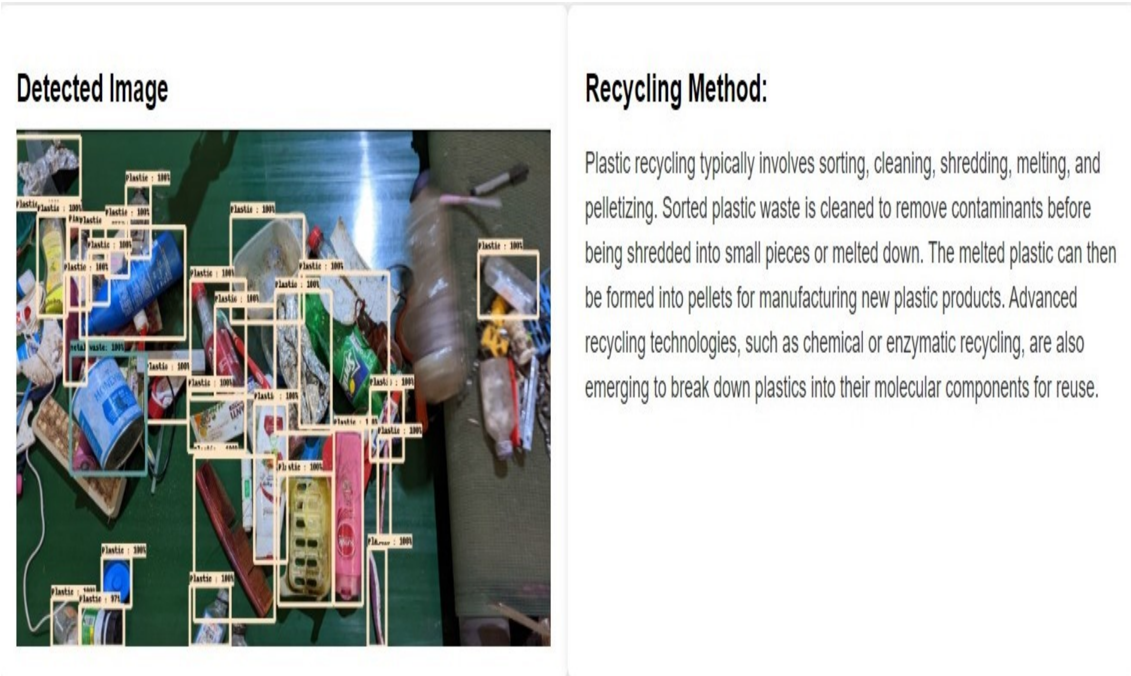

Figure 6.1.3: Graphical User Interface

Chapter 7

Conclusions and future works

7.1 Conclusion

In conclusion, our project represents a comprehensive exploration of waste detection algorithms, incorporating Convolutional Neural Networks (CNN), Support Vector Machines (SVM), Mask Region-based Convolutional Neural Network (MRCNN), YOLO, and MobileNetV2. We calculated accuracy and confidence levels for each algorithm. Notably, the utilization of MRCNN emerges as a promising choice, demonstrating effectiveness in capturing detailed information and spatial relationships within images. Overall, our project contributes valuable insights into advancing accurate waste detection, with MRCNN presenting itself as a noteworthy and promising algorithm in this context.

7.2 Future Scope

Usage of various techniques is done to identify and categorize different types of waste. As we look towards the future, our primary objective is to identify and employ the most effective algorithm that not only enhances the speed and accuracy of waste detection but also recommends improved recycling methods for the identified waste. This forward-looking approach is designed to align with eco-friendly practices and environmental conservation efforts. The goal of this comprehensive strategy is to optimize waste management processes, ultimately contributing to a greener and more efficient waste classification and recycling system.Tha main aim of this project is to establish a Waste Management System that is eco-friendly.

Bibliography

[1] H. Abdu and M. H. M. Noor, "A survey on waste detection and classification using deep learning," *IEEE Access*, vol. 10, pp. 128151–128165, 2022.

[2] N. Sarker, S. Chaki, A. Das, and M. S. A. Forhad, "Illegal trash thrower detection based on hogsvm for a real-time monitoring system," pp. 483–487, 2021.

[3] J. Li, J. Chen, B. Sheng, P. Li, P. Yang, D. D. Feng, and J. Qi, "Automatic detection and classification system of domestic waste via multimodel cascaded convolutional neural network," *IEEE transactions on industrial informatics*, vol. 18, no. 1, pp. 163–173, 2021.

[4] K. N. Sami, Z. M. A. Amin, and R. Hassan, "Waste management using machine learning and deep learning algorithms," *International Journal on Perceptive and Cognitive Computing*, vol. 6, no. 2, pp. 97–106, 2020.

[5] X. Xu, X. Qi, and X. Diao, "Reach on waste classification and identification by transfer learning and lightweight neural network," 2020.

[6] M. Pamintuan, S. M. Mantiquilla, H. Reyes, and M. J. Samonte, "i-bin: An intelligent trash bin for automatic waste segregation and monitoring system," pp. 1–5, 2019.

[7] C. Bircanoğlu, M. Atay, F. Beşer, Ö. Genç, and M. A. Kızrak, "Recyclenet: Intelligent waste sorting using deep neural networks," pp. 1–7, 2018.

[8] Y. Chu, C. Huang, X. Xie, B. Tan, S. Kamal, X. Xiong, *et al.*, "Multilayer hybrid deep-learning method for waste classification and recycling," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[9] B. S. Costa, A. C. Bernardes, J. V. Pereira, V. H. Zampa, V. A. Pereira, G. F. Matos, E. A. Soares, C. L. Soares, and A. F. Silva, "Artificial intelligence in automated sorting in trash recycling," pp. 198–205, 2018.

[10] S. Sudha, M. Vidhyalakshmi, K. Pavithra, K. Sangeetha, and V. Swaathi, “An automatic classification method for environment: Friendly waste segregation using deep learning,” pp. 65–70, 2016.