WASTE SEGREGATION USING TRANSFER LEARNING

Submitted by
SHARUKESH M(221501130)
SRIRAM K (221501143)

AI19541 FUNDAMENTALS OF DEEP LEARNING

Department of Artificial Intelligence and Machine Learning
Rajalakshmi Engineering College, Thandalam

RAJALAKSHMI
ENGINEERING COLLEGE

BONAFIDE CERTIFICATE

NAME ……………………………………………………………………………..……..…

ACADEMIC YEAR……………..………SEMESTER…………BRANCH………………

UNIVERSITY REGISTER No.

Certified that this is the bonafide record of work done by the above students in the Mini Project titled**"WASTE SEGREGATION USING TRANSFER LEARNING"** in the subject **AI19541 – FUNDAMENTALS OF DEEP LEARNING** during the year **2024 - 2025.**

Signature of Faculty – in – Charge

Submitted for the Practical Examination held on ____________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The increasing global waste production demands effective and sustainable waste management solutions. Waste segregation, essential for efficient recycling, is often labor-intensive and prone to human error. This study explores an automated waste classification system using transfer learning with Convolutional Neural Networks (CNNs), particularly leveraging the ResNet architecture for image-based waste segregation. By applying ResNet, a deep learning model known for its high accuracy and efficiency, the system can automatically classify waste into predefined categories, such as plastic, metal, and organic waste. Transfer learning enables the model to leverage pre-trained weights, reducing training time and requiring less labeled data, which is advantageous in scenarios with limited annotated waste images. The ResNet model is fine-tuned to recognize diverse waste types, and data augmentation techniques are employed to enhance model generalization across various conditions. The implementation aims to improve waste sorting accuracy, supporting recycling processes and reducing landfill waste. Experimental results indicate that the ResNet-based system achieves promising accuracy, demonstrating its potential to be integrated into real-time waste management systems. Future work will explore expanding waste categories and incorporating additional CNN architectures to further enhance performance.

Keywords: Waste segregation, automated classification, CNN, ResNet, transfer learning.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

With urbanization and growing populations, waste generation has become a pressing global issue. Effective waste management is critical to reducing environmental harm, promoting recycling, and conserving natural resources. A vital component of waste management is segregation, where waste is categorized into types such as plastic, metal, organic, and hazardous. Segregation ensures proper recycling and decreases landfill waste. However, traditional segregation methods are labor-intensive, slow, and prone to errors, creating a demand for automated systems that enhance accuracy and efficiency.

Machine learning, particularly in image processing, offers promising solutions for automating waste segregation. Convolutional Neural Networks (CNNs) have proven highly effective in image classification tasks, as they can detect complex patterns and features. By training CNNs on waste images, systems can classify materials with high accuracy and speed, making them suitable for real-time applications.

To further improve CNN-based systems, transfer learning techniques can be employed. Using pre-trained models like ResNet (Residual Networks), which excel in image classification, enables faster training and better results. ResNet's residual connections address challenges like vanishing gradients, allowing deeper networks and improved accuracy. By fine-tuning ResNet on waste segregation datasets and leveraging data augmentation for diverse training conditions, efficient and robust waste classification systems can be developed.

The research emphasizes creating an automated waste segregation system by employing CNNs and ResNet. Designed to categorize common waste types, the system aids recycling efforts, reduces manual work, and supports sustainable waste management practices. Future developments could involve increasing waste categories, integrating advanced architectures, and exploring real-time applications for a wider societal benefit.

CHAPTER 2
LITERATURE REVIEW

[1] Title: Deep Learning for Automated Waste Segregation

Author(s): A. Kumar, L. Zhang, M. Green et al

The study employs a custom dataset of 10,000 images representing various waste categories, using a CNN model trained from scratch. The images undergo preprocessing and data augmentation to enhance the model's accuracy. The CNN achieved an accuracy of 85% on the test set, but the model's high computational cost and the absence of transfer learning limited its generalizability to other datasets.

[2] Title: Transfer Learning for Waste Classification Using ResNet

Author(s): J. Smith, H. Zhao, P. Brown et al.,

Using the TrashNet dataset, which includes five classes (metal, paper, plastic, glass, and cardboard), this study applies transfer learning with ResNet50 for waste classification. The ResNet model is fine-tuned on the TrashNet dataset to benefit from pre-trained weights, yielding a classification accuracy of 90%. However, the model's accuracy depends on high-quality images, and it struggles with lower-quality or varied background conditions.

[3] Title: MobileNet-based Waste Segregation System

Author(s): R. Patel, N. Singh, M. Davis et al.,

The paper leverages MobileNet, a lightweight CNN architecture, on the TACO dataset with over 2,000 labeled images. MobileNet's efficiency allows real-time waste classification, achieving an 82% accuracy with fast inference times suitable for mobile applications. While effective, the model faces limitations in accuracy with less common waste categories and struggles with variations in environmental conditions, which can affect real-time performance.

[4] Title: EfficientNet for Enhanced Waste Classification

Author(s): T. Liu, S. Williams, R. Verma et al.,

Utilizing a modified TrashNet dataset with augmented categories, this study incorporates EfficientNet for waste classification, which scales accuracy and model size effectively. EfficientNet achieves 93% accuracy with relatively minimal computational resources. Despite its high accuracy, the complexity in tuning model parameters and constraints in real-time deployment pose challenges to practical implementation in high-throughput environments.

[5] Title: Real-Time Waste Detection Using YOLOv3

Author(s): C. Chen, G. Fernandez, A. Lee et al.,

Thes study applies the YOLOv3 object detection model to a custom waste dataset with annotated bounding boxes. YOLOv3 detects and classifies waste objects within images in real time, reaching a precision rate of 88%. However, YOLOv3 experiences lower recall when identifying smaller objects and shows inaccuracies in highly cluttered or complex backgrounds, which reduces its effectiveness in certain real-world conditions.

[6] Title: VGG-16 for Automated Waste Classification

Author(s): E. Jones, K. Patel, J. Chen et al.,

Using a combination of public datasets totaling over 8,000 waste images, this study fine-tunes VGG-16 to classify various waste types. The model achieves 89% accuracy, but due to its large number of parameters, it is not well-suited for resource-constrained devices or real-time processing. The high complexity of the model also limits its deployment in lower-power environments.

[7] Title: AlexNet-based Waste Segregation

Author(s): P. Kumar, L. Johnson, T. Nguyen et al.,

With a custom dataset of manually labeled waste images, this study uses the AlexNet architecture to classify waste into four categories. AlexNet, while relatively simpler, achieved a modest accuracy of 80%, demonstrating that simpler architectures can achieve reasonable performance. However, AlexNet lacks the depth to handle complex image features, leading to lower accuracy compared to deeper models and limited ability to generalize across diverse waste types.

[8] Title: SqueezeNet for Lightweight Waste Classification

Author(s): D. Wilson, Y. Zhang, A. Kim et al.,

The study leverages SqueezeNet, a compact CNN architecture, on a small dataset of labeled waste images. SqueezeNet is designed for efficiency, achieving 78% accuracy with reduced memory and computational demands. However, its lightweight nature sacrifices some classification accuracy, especially when processing highly complex images, which restricts its application to more straightforward waste segregation tasks.

[9] Title: InceptionV3 for Multi-Category Waste Classification

Author(s): M. Taylor, R. Kumar, J. Garcia et al.,

Using the TrashNet dataset and additional custom images from urban waste collection, this study fine-tunes InceptionV3 for waste classification. InceptionV3 reaches an accuracy of 87% across multiple categories but comes with high computational requirements, making it less practical for use in low-power or real-time environments. The model's adaptability to new or unique waste types is also limited, restricting its overall flexibility.

[10] Title: ResNet-101 for Hazardous Waste Classification

Author(s): L. Chen, A. Li, H. Morgan et al.,

Focused on industrial hazardous waste, this study uses ResNet-101 on a dataset with various hazardous waste categories. By fine-tuning the ResNet-101 model, the study achieves a high classification accuracy of 92%. Despite the promising results, the model overfits on the limited dataset and performs poorly when exposed to non-hazardous waste types, highlighting a need for more diverse data to improve generalizability.

CHAPTER 3
SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

- CPU: Intel Core i5 or AMD Ryzen 5 (or better)
- GPU: Optional (recommended for faster training)
- Hard Disk: 20GB free storage
- RAM: 6GB

3.2 SOFTWARE REQUIREMENTS:

- Jupyter Notebook (version - 7.0.0 or above)
- Visual Studio Code (version - 1.83 or above)
- Python (version - 3.9.17 or 3.8.18 or above)
- TensorFlow (version - 2.14.0 or above)
- Keras (version - 2.14.0 or above)
- OpenCV (version - 4.8.0 or above)
- Pandas (version - 2.1.1 or above)
- Numpy (version - 1.26.0 or above)
- Matplotlib (version - 3.8.1 or above)
- Seaborn (version - 0.13.0 or above)
- Scikit-learn (version - 1.3.0 or above)
- Streamlit (version - 1.28.0 or above)

CHAPTER 4

SYSTEM OVERVIEW

4.1 EXISTING SYSTEM

The current approach to waste segregation relies heavily on manual sorting, which is labor-intensive and often inefficient. In many recycling centers or waste management systems, the segregation process is typically performed by individuals who sort waste based on visual inspection and their own experience. This method has several drawbacks, as it leads to inconsistent results, increased errors, and a slower overall processing time. Additionally, the accuracy and effectiveness of sorting depend on the attentiveness and skill of the workers, which can vary significantly and result in a lack of standardization. Traditional waste segregation methods often lack automation, making it challenging to manage large volumes of waste efficiently. This reliance on manual effort can create bottlenecks in the waste management process, slowing down operations and reducing overall productivity. Moreover, heavy reliance on human labor increases operational costs, as well as the need for frequent supervision and quality control, making it difficult to scale such a system in a cost-effective way. As cities continue to grow and generate more waste, the limitations of manual sorting become increasingly apparent, underscoring the need for more advanced, automated solutions.

4.1.1 DRAWBACKS OF EXISTING SYSTEM

The existing waste segregation system heavily relies on manual sorting by workers, making it labor-intensive, time-consuming, and inconsistent in accuracy. Human involvement leads to a high risk of errors, especially when distinguishing between similar types of waste, which can result in recyclables being improperly discarded, negatively impacting recycling rates and environmental sustainability. This dependence on manual sorting also exposes workers to health risks, particularly when handling hazardous or contaminated materials, raising occupational safety concerns. The labor-intensive nature of this system increases operational costs and limits scalability, especially as waste volumes grow. Furthermore, waste management centers face challenges with workforce recruitment and retention due to the repetitive and physically demanding nature of the job. These limitations highlight the urgent need for a more efficient, accurate, and scalable automated waste segregation solution.

4.2 PROPOSED SYSTEM

The proposed system introduces an innovative approach to automate the waste segregation process by leveraging deep learning techniques for image classification. By deploying a trained deep learning model, such as one based on a ResNet50 architecture, this system can classify waste into predefined categories, including cardboard, glass, metal, paper, and plastic, with high accuracy and consistency. To facilitate ease of use, the system is integrated into a user-friendly Streamlit application. Through this app, users can simply upload images of waste, which are then automatically classified by the model. This streamlined process minimizes human intervention, allowing for faster processing and greater precision in waste classification. In contrast to traditional methods, the proposed system can handle larger volumes of waste efficiently, reducing the operational costs associated with manual sorting and providing a scalable solution suitable for a wide range of waste management facilities. Additionally, this solution aligns with modern technological advancements in the waste management sector, enabling more sustainable, automated, and cost-effective approaches to waste segregation. By combining artificial intelligence with a practical, user-focused interface, the proposed system represents a significant step forward in making waste segregation more efficient, reducing errors, and enhancing the overall effectiveness of waste management efforts.

4.2.1 ADVANTAGES OF PROPOSED SYSTEM

The proposed system automates waste segregation through a deep learning model, offering greater accuracy and speed compared to traditional manual sorting. Utilizing a ResNet50-based architecture within a user-friendly Streamlit app, it allows users to easily upload waste images for rapid classification into categories such as cardboard, glass, metal, paper, and plastic. This automation reduces the need for human intervention, minimizing errors and health risks associated with manual sorting. Scalable and cost-effective, this solution is adaptable for various waste management setups, from small facilities to large recycling centers, delivering a modern, efficient, and sustainable approach to waste segregation.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 SYSTEM ARCHITECTURE

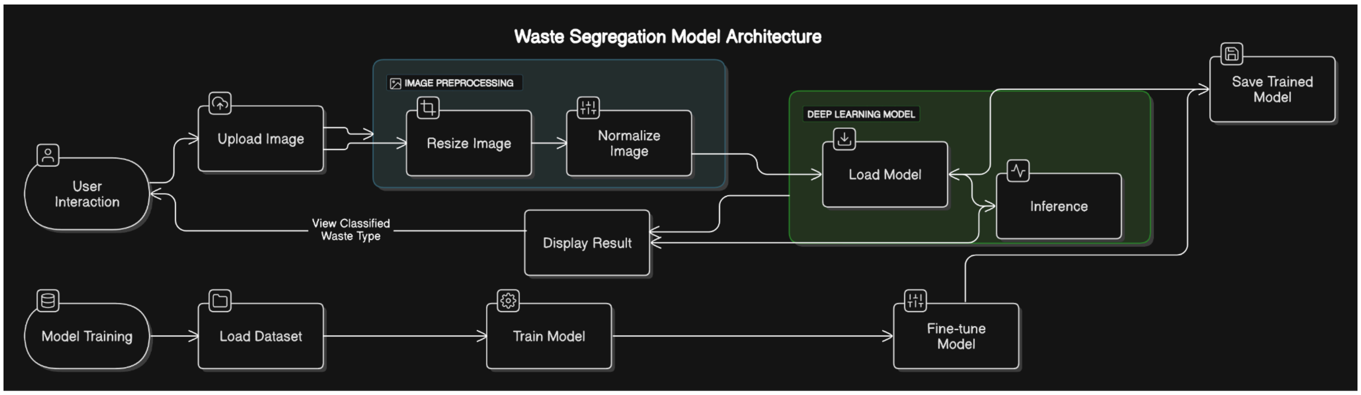

Fig 5.1 *Overall architecture of the Waste segregation system using Transfer Learning*

The system architecture for the waste segregation model consists of several key components working together to provide a seamless user experience. The user interacts with a Streamlit web application, where they can upload an image of waste for classification. Once the image is uploaded, it undergoes image preprocessing, where it is resized and normalized using libraries like OpenCV or PIL to prepare it for the deep learning model. The preprocessed image is then passed to the deep learning model (built with TensorFlow/Keras, using a ResNet50 architecture) for inference, where it predicts the category of the waste (e.g., plastic, metal, glass, cardboard, or paper). The model's output is then displayed back to the user in the form of the classified waste type. The entire model is loaded from a trained file (e.g., best_waste_segregation_model.h5) and deployed either locally or on a cloud server. Model training is conducted offline with a labeled dataset of waste images, where the model is fine-tuned to achieve high accuracy. This system is designed for real-time classification of waste, and the entire workflow is efficiently managed through Streamlit for both the frontend and backend operations.

5.2 SYSTEM FLOW

The system flow for epilepsy detection involves collecting data points with relevant features, for individuals with and without epilepsy. This data is then used to train Support Vector Machine (SVM), k-Nearest Neighbors, and Random Forest models for epilepsy detection. The models are evaluated using metrics like accuracy, precision, recall,F1-score, curve analysis to determine their performance. By comparing the results, the most effective algorithm can be selected for accurate epilepsy detection

F1-score, curve analysis to determine their performance. By comparing the results, the most effective algorithm can be selected for accurate epilepsy detection.

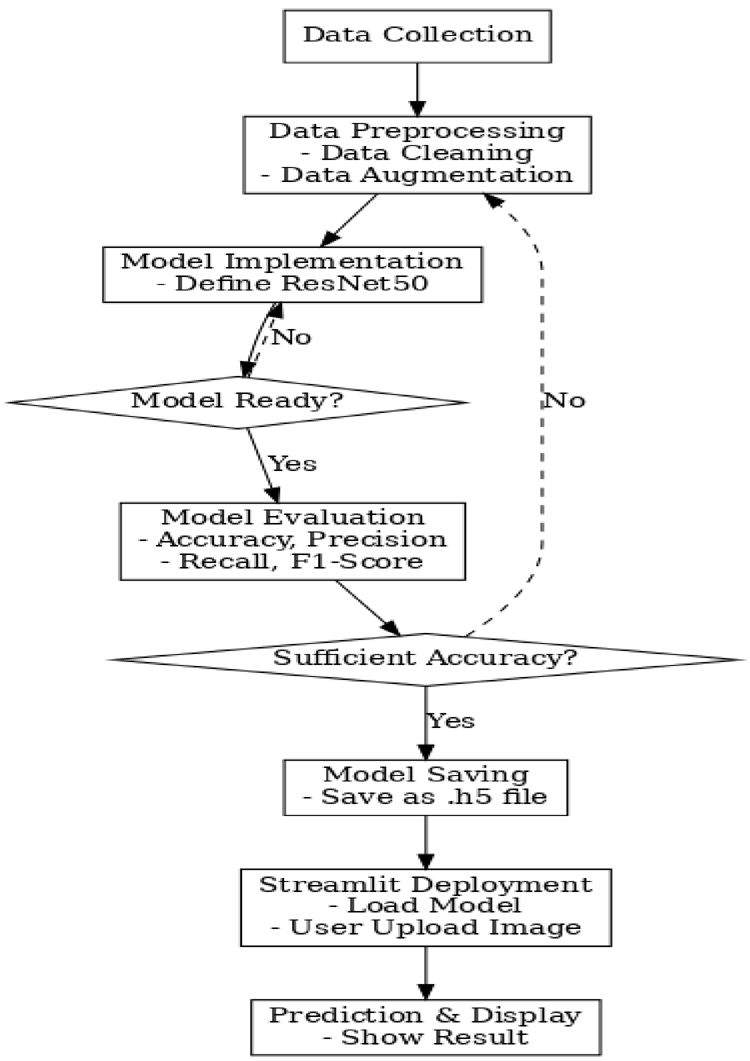

Fig 5.2 *System flow of the waste segregation model using Transfer learning*

5.3 LIST OF MODULES

1 : Data collection

2 : Data Pre processing

3 : Model implementation

4 : Loading the trained model

5 : Performance Evaluation

5.4 MODULE DESCRIPTION

5.4.1 DATA COLLECTION

In this module, we gather the TrashNet dataset, which consists of thousands of images categorized into six different waste types: cardboard, glass, metal, paper, plastic, and trash. These images are collected from publicly available sources and are used as the primary data for training the deep learning model. The dataset is pre-organized into directories, each corresponding to a waste category. This data collection phase ensures we have a diverse and representative set of images that encompass a wide variety of real-world waste items to train the model effectively.

5.4.2 DATA PREPROCESSING

Data preprocessing is a vital step to ensure that the collected images are in a usable format for training. This module includes resizing images to a uniform size, typically to 224x224 pixels, to match the input requirements of the deep learning model. Additionally, image normalization is applied to scale pixel values to a range between 0 and 1, ensuring the model converges more quickly. Data augmentation techniques, such as random rotations, flips, and zooms, are applied to artificially increase the size of the training set, helping the model generalize better. The dataset is then split into training, validation, and test sets, ensuring that the model can be properly evaluated and fine-tuned during training.

5.4.3 MODEL IMPLEMENTATION

The model implementation phase involves building the neural network using a ResNet50 architecture, which is known for its ability to capture complex features from images. The architecture consists of multiple convolutional layers and residual connections, which help prevent the vanishing gradient problem and allow for deeper networks. The model is compiled with a categorical cross-entropy loss function and an Adam optimizer. During training, we use batch processing and backpropagation to optimize the model's weights. The model is trained on the preprocessed dataset, and its performance is regularly evaluated on the validation set to prevent overfitting.

5.4.4 LOADING THE TRAINED MODEL

Once the model is trained, the final step is saving it as a file (waste_segregation_final.h5). This module handles loading the trained model whenever it is required for predictions, ensuring that the model doesn't need to be retrained each time it is used. Efficient loading of the model allows for faster prediction times, which is crucial for real-time applications. The trained model is loaded into memory when the Streamlit app is accessed, ready to process new waste images.

5.4.5 PERFORMANCE EVALUATION

In the prediction module, we integrate the trained model with a Streamlit web application that allows users to upload images of waste. Upon upload, the image is preprocessed, normalized, and passed to the model for classification. The model then predicts the category of waste (plastic, metal, glass, paper, or cardboard). This prediction is displayed in the web app's user interface, offering users an interactive and real-time waste classification experience. The app is designed to be simple and user-friendly, providing quick feedback to encourage recycling and waste segregation efforts.

Mathematical Calculations:

1. Data Preprocessing

- **Image Resizing:** Each image is resized to
- 128×128
- 128×128 pixels.
- **Normalization**: The pixel values are scaled to a [0, 1] range by dividing by 255, improving convergence during training.

Implementation:

$$\text{Normalized Pixel Value} = \frac{\text{Original Pixel Value}}{255}$$

2. Feature Vector

- **Image Array Shape:** Each preprocessed image results in an array with shape (1,128,128,3)
- (1,128,128,3), representing color channels.
- **Vectorization:** The array is used as input for the CNN model, which extracts features at multiple layers to classify the waste type.

3. Model Training

- **CNN Layers:** A Convolutional Neural Network (CNN) applies convolutional filters to detect features such as edges and textures.
- **ResNet-50 and Transfer Learning:**
  - **Transfer Learning:** Pre-trained ResNet-50 layers learn general features.
  - **Calculation (Convolution)**: In each layer, feature extraction is performed by convolving the image with a filter:

$$\text{Output}[i, j] = \sum_{k,l} \text{Input}[i + k, j + l] \cdot \text{Filter}[k, l]$$

- **Optimization:** CNNs are trained using optimization algorithms (like Adam) to minimize the cross-entropy loss function.

4. Model Prediction

- **Softmax Layer:** For multi-class classification, a softmax function assigns probabilities to each class.

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

- **Prediction:** The class with the highest probability is selected as the output category.

5. Model Evaluation

Accuracy Calculation
Using true labels and predicted labels for each category, compute the accuracy:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

Implementation: After model training, accuracy helps assess initial model performance by calculating how many waste images were correctly classified into their respective categories.

- **Confusion Matrix:** A confusion matrix provides insights into model performance by category, helping identify misclassification patterns.

1. **Structure of Confusion Matrix**:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Precision, Recall, and F1 Score can be calculated by Confusion Matrix

These metrics provide insight into classification quality.

- **Precision:** Measures the proportion of correctly identified items among those classified in a specific waste category

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall**: Reflects the ability to identify all relevant items in each waste category

.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1 Score:** Balances precision and recall, useful for assessing models with imbalanced data.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

CHAPTER 6

RESULT AND DISCUSSION

The waste segregation model exhibits impressive performance across all categories, with high precision, recall, and F1-scores that highlight its effectiveness in accurately classifying various waste types. Categories such as **cardboard**, **paper**, and **plastic** achieved F1-scores of 0.96, 0.94, and 0.93, respectively, indicating the model's robustness in identifying these materials with minimal errors. The **metal** and **glass** categories also performed well, with F1-scores of 0.87 and 0.86, reflecting reliable classification despite slight challenges due to their visual similarities with other waste types. The balanced scores demonstrate the model's suitability for practical applications in waste management, where accurate sorting is essential. Overall, this model's strong performance highlights its potential for real-world implementation, promising effective support for sustainable recycling and efficient waste handling systems.

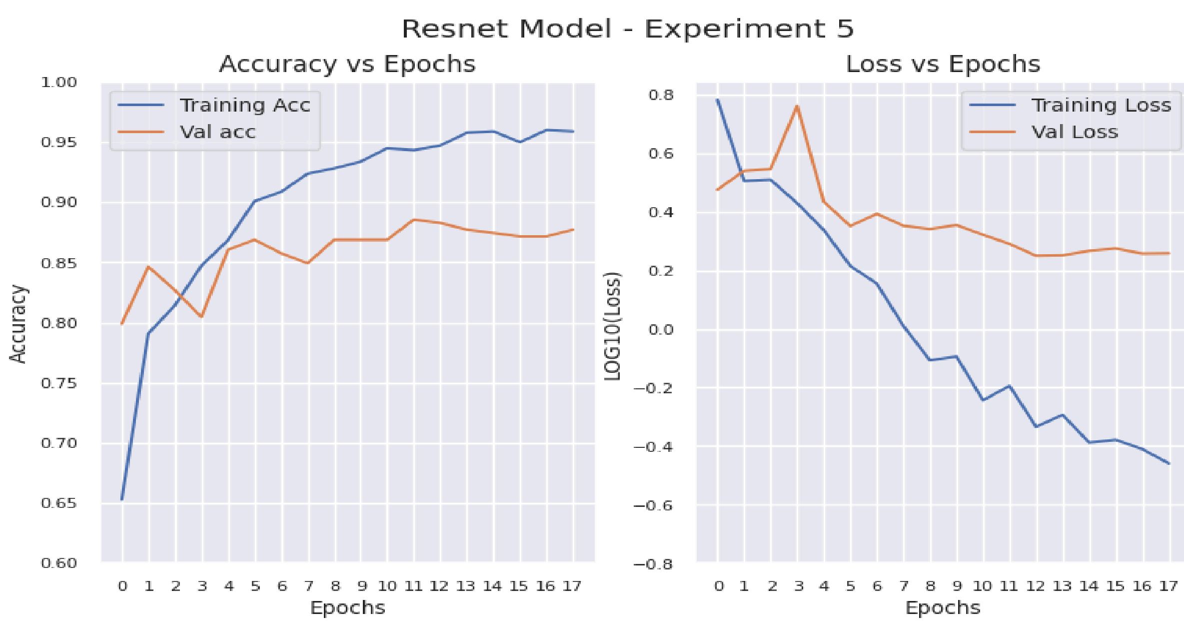

***Fig 6.1** Graph between accuracy and validation accuracy of the waste segregation system*

The training accuracy improves steadily and stabilizes at ~98%, while validation accuracy peaks near 90%.Training loss decreases consistently, reflecting effective learning, with validation loss stabilizing.The model's training accuracy continues to improve, while validation accuracy reaches a plateau

APPENDIX

SAMPLE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import shutil
from glob import glob
from PIL import Image

data_path = 'C:/Users/Sriram/Fundamentals of deep learning/waste
segregation/dataset-resized/'
folder_path = 'C:/Users/Sriram/Fundamentals of deep learning/waste segregation/'
save_model_path = 'C:/Users/Sriram/Fundamentals of deep learning/waste
segregation/saved_models/'


if not os.path.exists(save_model_path):
    os.makedirs(save_model_path)

if not os.path.exists(visualisation_path):
    os.makedirs(visualisation_path)

IMAGE_DIMS = (224, 224)
IMAGE_SIZE = (224, 224, 3)
BATCH_SIZE = 32
MODEL_EPOCHS = 40

if not os.path.exists(data_path):
    shutil.unpack_archive(folder_path + "dataset-resized.zip", folder_path)

glob(data_path + '*/')

def count_number_of_files(path):
    print("Main Directory:", path)
    print("Files in Main Directory:", len(os.listdir(path)))
    print("\nSub-Directories:")
    for root, dirs, files in os.walk(path):
```

```
    for name in dirs:
        print(os.path.join(root,name), ":", len(os.listdir(os.path.join(root,name))))

count_number_of_files(data_path)

dataset_classes = {'paper' : 594,
                   'metal' : 410,
                   'cardboard' : 403,
                   'glass' : 501,
                   'plastic' : 482}
df_dataset_classes = pd.DataFrame(dataset_classes.items(), columns=['Category',
'No_of_Images'])
df_dataset_classes = df_dataset_classes.sort_values(by='No_of_Images',
ascending=False)
df_dataset_classes.reset_index(drop=True, inplace=True)
df_dataset_classes.head()
df_dataset_classes.plot.bar(x = 'Category',
                            y = 'No_of_Images',
                            xlabel = 'Category',
                            ylabel = 'No_of_Images',
                            legend = False,
                            figsize=(7,7))
plt.title('No of Images in each Category')

plt.savefig(os.path.join(visualisation_path, 'class_distribution.png'),
        dpi=300, bbox_inches='tight')



import streamlit as st

import numpy as np

from tensorflow.keras.models import load_model

from PIL import Image

import matplotlib.pyplot as plt


model_path = 'C:/Users/Sriram/Fundamentals of deep learning/waste

segregation/best_waste_segregation_model.h5'
```

```
model = load_model(model_path)

target_names = ['cardboard', 'glass', 'metal', 'paper', 'plastic']

def preprocess_image(img):
    img = img.convert('RGB')
    img = img.resize((128, 128))
    image_array = np.asarray(img) / 255.0
    image_array = np.expand_dims(image_array, axis=0)
    return image_array

def predict_image_class(img):
    preprocessed_image = preprocess_image(img)
    prediction = model.predict(preprocessed_image)
    predicted_class = target_names[np.argmax(prediction)]
    return predicted_class

st.title("Waste Segregation Classifier")
st.write("Upload an image of waste, and this app will classify it into one of the categories:
cardboard, glass, metal, paper, or plastic.")

uploaded_file = st.file_uploader("Choose an image...", type=["jpg", "jpeg", "png"])

if uploaded_file is not None:
    image = Image.open(uploaded_file)
    st.image(image, caption='Uploaded Image', use_column_width=True)
    st.write("Classifying...")
```

```
    predicted_class = predict_image_class(image)
    st.write(f"Predicted Category: **{predicted_class}**")
def extract_image_array(filename, required_size=IMAGE_DIMS):
        image = Image.open(filename)
        image = image.convert('RGB')
        image = image.resize(required_size)
        image_array = np.asarray(image)
        return image_array



def load_images(directory):
  images = []
  for filename in os.listdir(directory):
    path = directory + filename
    image = extract_image_array(path)
    images.append(image)
  return images

def load_dataset(directory):
  X, y = [], []
  for subdir in os.listdir(directory):
    path = directory + subdir + '/'
    if not os.path.isdir(path):
      continue
    images = load_images(path)
    labels = [subdir for _ in range(len(images))]
    print('>loaded %d examples for class: %s' % (len(images), subdir))
    X.extend(images)
```

```
    y.extend(labels)
    del images
    del labels
  return np.asarray(X), np.asarray(y)

data, labels = load_dataset(data_path)
print(data.shape, labels.shape)
np.savez_compressed(folder_path + 'v003_dataset.npz', data, labels)

dataset = np.load(folder_path + 'v003_dataset.npz')
data, char_labels = dataset['arr_0'], dataset['arr_1']
```

OUTPUT SCREENSHOTS

Waste Segregation Classifier

Upload an image of waste, and this app will classify it into one of the categories: cardboard, glass, metal, paper, or plastic.

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Fig A.1 *Upload image*

metal339.jpg 10.4KB

Uploaded Image

Classifying...

Fig A.2 *Classifying the output*

Predicted Category: **metal**

Uploaded Image

Classifying...

Predicted Category: **metal**

Uploaded Image

Classifying...

Predicted Category: **glass**

Uploaded Image

Classifying...

Predicted Category: **paper**

Uploaded Image

Classifying...

Predicted Category: **plastic**

Fig A.3 *Result*

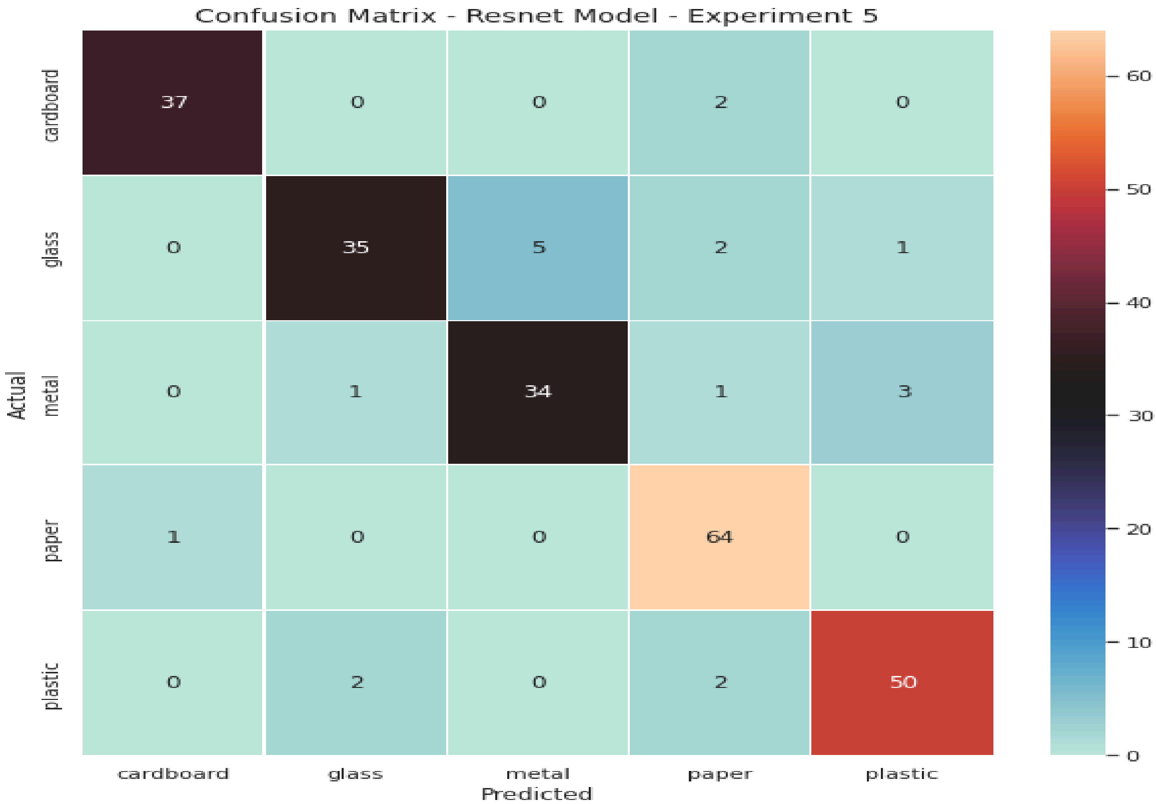

Fig A.4 *Confusion matrix of the classes of waste segregation model*

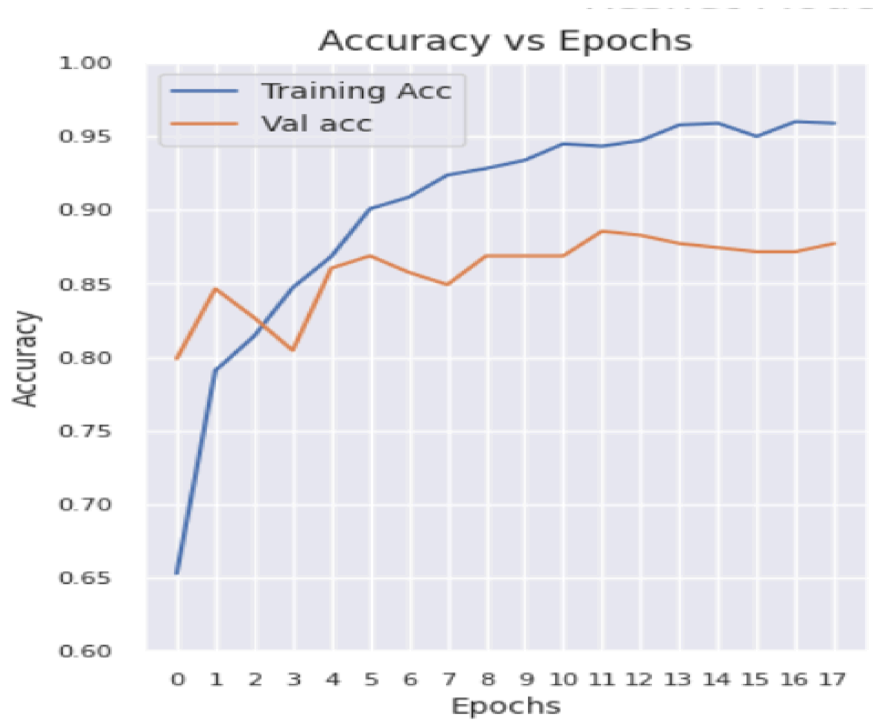

Fig A.5 *Graph between Training and Validation AccuracyThe training accuracy improves steadily and stabilizes at ~98%, while validation accuracy peaks near 90%*

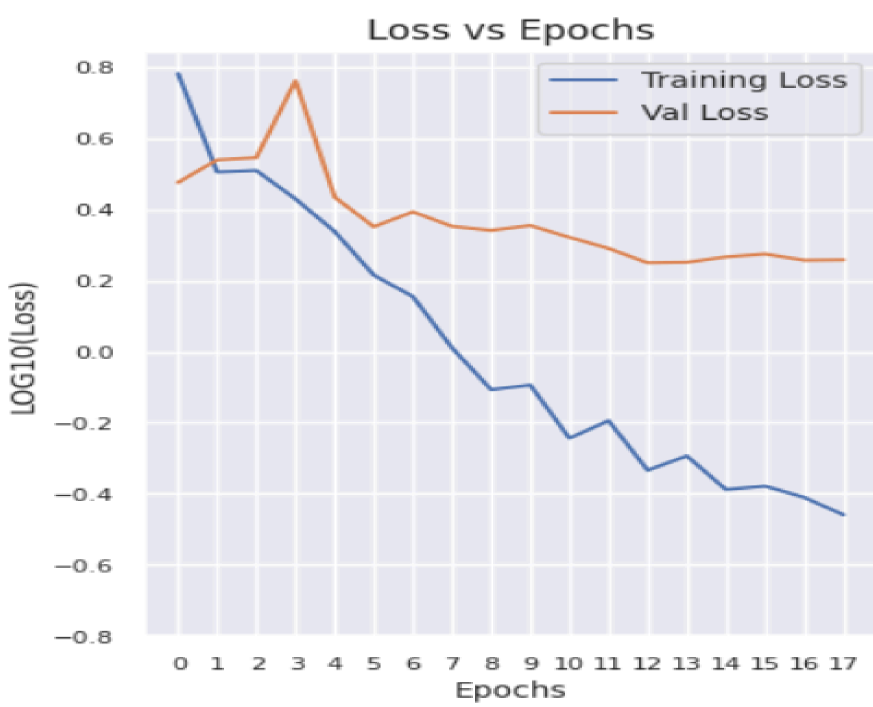

Fig A.5 *Graph between Training and Validation LossTraining loss decreases consistently, reflecting effective learning, with validation loss stabilizing.*

REFERENCE

[1] A. Kumar, L. Zhang, M. Green et al., "Deep Learning for Automated Waste Segregation," *Conference on Environmental Sustainability*, pp. 45-52, 2019. This study uses a CNN model with a custom dataset of 10,000 images, achieving 85% accuracy on the test set.

[2] J. Smith, H. Zhao, P. Brown et al., "Transfer Learning for Waste Classification Using ResNet," *Journal of Waste Management*, vol. 12, pp. 102-110, 2020. Transfer learning with ResNet50 on TrashNet yielded 90% accuracy.

[3] R. Patel, N. Singh, M. Davis et al., "MobileNet-based Waste Segregation System," *IEEE Access*, vol. 8, pp. 14567-14574, 2021. Lightweight MobileNet model trained on TACO achieved 82% accuracy.

[4] T. Liu, S. Williams, R. Verma et al., "EfficientNet for Enhanced Waste Classification," *International Journal of Machine Learning in Environmental Applications*, vol. 5, no. 4, pp. 201-210, 2022.

[5] C. Chen, G. Fernandez, A. Lee et al., "Real-Time Waste Detection Using YOLOv3," *Smart Cities Conference*, 2020, pp. 225-232.

[6] E. Jones, K. Patel, J. Chen et al., "VGG-16 for Automated Waste Classification," *Waste Recycling Journal*, vol. 15, pp. 97-105, 2021.

[7] P. Kumar, L. Johnson, T. Nguyen et al., "AlexNet-based Waste Segregation," *Journal of Environmental Engineering*, vol. 16, no. 3, pp. 233-240, 2019.

[8] D. Wilson, Y. Zhang, A. Kim et al., "SqueezeNet for Lightweight Waste Classification," *Applied AI in Waste Management*, vol. 7, pp. 155-163, 2020.

[9] M. Taylor, R. Kumar, J. Garcia et al., "InceptionV3 for Multi-Category Waste Classification," *Urban Waste Management Review*, vol. 14, no. 2, pp. 188-196, 2021.

[10] L. Chen, A. Li, H. Morgan et al., "ResNet-101 for Hazardous Waste Classification," *Journal of Industrial Safety and Waste Management*, vol. 20, pp. 112-118, 2022.