Trash Image Classification Using Machine Learning and Deep Learning Ensemble

Presented by:

Ahmed Awadallah - 221000354

Esraa Ahmed - 221000462

Alhussein Nagdy – 221002149

Instructed by: Dr. Sahar Selim Soussa

Abstract

With the increasing generation of waste worldwide, efficient classification of trash into categories (metal, organic, paper and plastic) can aid recycling and waste management systems. This project implements a hybrid machine learning pipeline combining Convolutional Neural Networks (CNNs) and classical models (Random Forest and Logistic Regression) in a soft weighted voting ensemble. A FastAPI backend exposes a REST endpoint for image classification, while a React frontend visualizes predictions. Accuracy on the dataset acquired reaches 78%.

1. Introduction

Waste management is a critical global challenge. Automated trash classification using computer vision techniques can improve recycling efficiency and reduce human labor. Traditional methods rely on manual sorting, which is slow and error-prone. Machine learning offers a scalable approach for automated classification.

This project explores the combination of deep learning and classical ML models to classify trash images into 4 categories, which are metal, organic, paper and plastic. It also provides a backend, an API and a user-friendly frontend interface for more practical use.

2. Related Work

Recent studies in waste classification highlight the efficacy of deep learning methods for automating the categorization of trash images. Shafek *et al.* employed a VGG16 CNN architecture to distinguish between organic and recyclable waste demonstrating CNNs' suitability in this domain [1]. Also, Rathod *et al.* used a DenseNet201-based network with transfer learning and augmentation to classify garbage across multiple categories [2]. Kruthika *et al.* compared several modern CNN architectures, such as MobileNet and NASNet, for multi-class classification which showed that advanced models can effectively handle diverse waste types [3]. Shi *et al.* proposed a hybrid CNN design with simplified structure, showing that tailored CNN architectures can maintain high accuracy with fewer parameters [4].

3. Methodology

3.1 Dataset & Preprocessing

- Dataset consists of images scraped from DuckDuckGo using the ddgs library.
- Images are organized into folders for each class: metal, organic, paper, plastic.
- Preprocessing uses Keras ImageDataGenerator for:
    - Rescaling pixel values to [0,1]
    - Data augmentation (rotation, width/height shift, horizontal flips)
    - Train/validation split (80/20)
    - Resizing images to (128,128)

3.2 CNN Model

- A sequential CNN with three convolutional layers, max pooling, fully connected layers, and dropout.
- Uses softmax output for 4-class classification.
- Early stopping on validation accuracy prevents overfitting.

3.3 Classical Models

- Logistic Regression and Random Forest are trained on CNN-extracted features, allowing them to leverage semantic features learned by the CNN.

3.4 Soft Voting Ensemble

Combines CNN, Logistic Regression, and Random Forest predictions using weighted soft voting:

- CNN: 0.25
- Logistic Regression: 0.25
- Random Forest: 0.5

Produces final class prediction and confidence scores.

3.5 Backend API

- Implemented using FastAPI.

- Accepts image uploads through POST /classify and returns a JSON with predicted class, confidence, and probabilities for all classes.

3.6 Frontend

- Offers a friendly interface built with React and Tailwind CSS, using ShadCN UI components.
- Displays predicted class, confidence, and probability distribution.

3.7 Model Persistence

CNN model saved with cnn_model.save(), while the Logistic Regression and Random Forest models are saved using joblib.dump(). This is important to allow for more compact utilization, as in the backend and the API for this project.

---

4. Results & Discussion

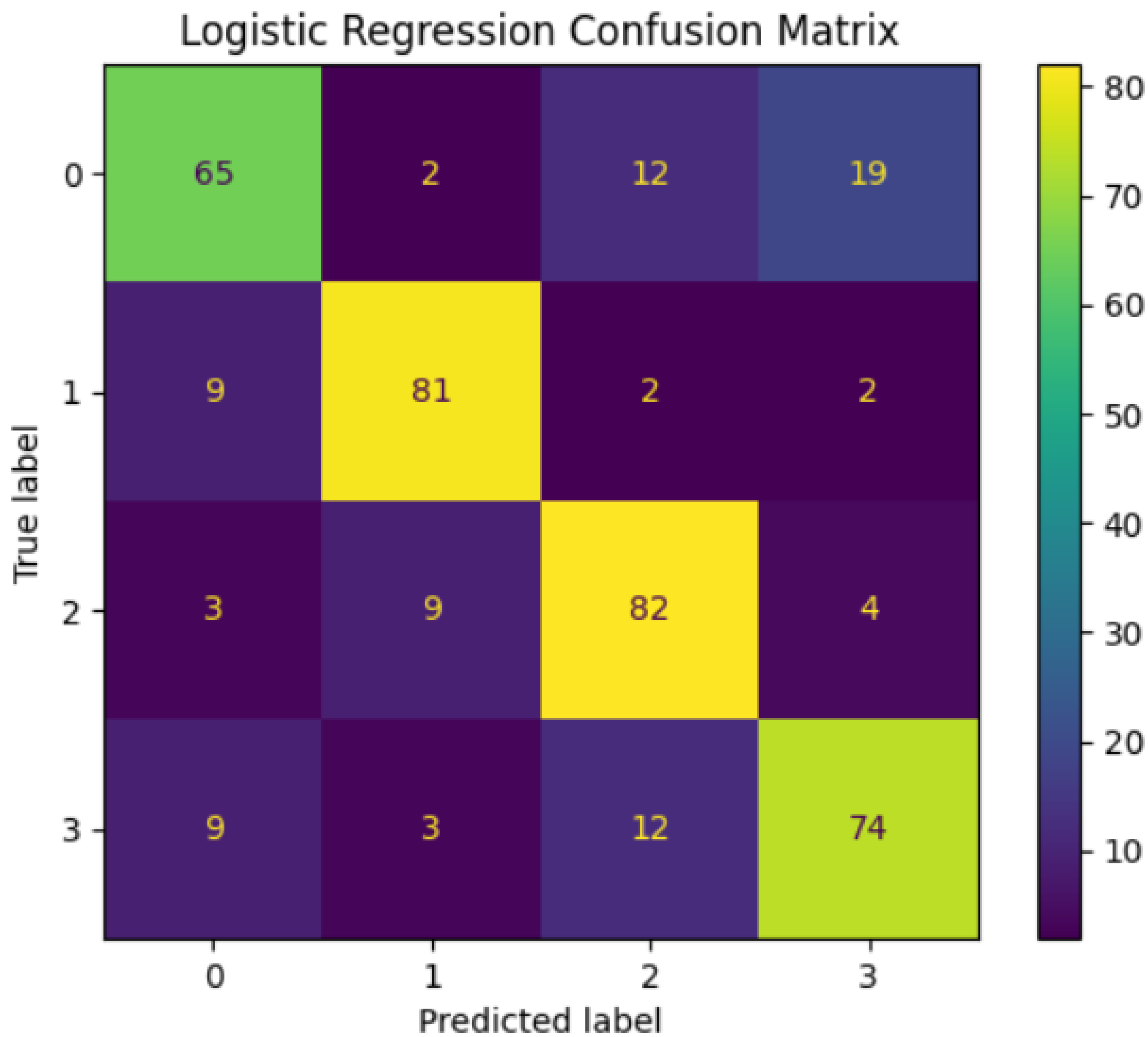
4.1 Model Performance

Model	Test Accuracy	Error Rate
CNN	0.74	0.25
Logistic Regression	0.77	0.22
Random Forest	0.77	0.22
Ensemble	0.78	0.21

- The ensemble slightly improves accuracy compared to individual models.
- All of the models alone perform well, but combining models captures complementary strengths.

4.2 Confusion Matrix (Figure 1)

Figure 1



- Confusion matrices show some misclassification between metal and plastic indicating visually similar features.
- Weighted soft voting helps resolve ambiguous cases.

5. Conclusion

This project demonstrates an effective hybrid approach for trash classification using both deep learning and classical machine learning. The soft voting ensemble improves overall accuracy and robustness.

The deployed FastAPI backend and React frontend enable practical usage for real-time application.

6. Tools & Libraries

Backend & API

- API Development: FastAPI, Pydantic, Uvicorn, python-multipart
- Machine Learning & Data Processing: TensorFlow/Keras, scikit-learn, numpy, pandas, Pillow
- Visualization: matplotlib
- Model Persistence: joblib

Frontend & Visualization

- React (TypeScript), ShadCN, Tailwind CSS

References:

[1] D. Shafek, H. W. Hilow, and M. Ahmed, "Classification of waste images using deep learning technique," *J. Appl. Res. Technol.*, vol. 23, no. 4, pp. 381–391, 2025.

📌 [2] K. Rathod et al., "Garbage Classification Based on Dense Network (GCDN) using Transfer Learning and Modified Hyperparameter," *Int'l J. Intell. Sys. Appl. Eng.*, 2025.

📌 [3] S. M. Kruthika et al., "Garbage Classification: A Deep Learning Perspective," *Int'l Res. J. Adv. Eng. Hub*, vol. 2, no. 12, pp. 2774–2780, Dec. 2024.

📌 [4] C. Shi, C. Tan, T. Wang, and L. Wang, "A Waste Classification Method Based on a Multilayer Hybrid Convolution Neural Network," *Appl. Sci.*, vol. 11, no. 18, p. 8572, 2021.