Scilab case study project on

Facilitating citizen science initiatives for waste management via machine learning in Scilab

Naini Diwan
Indian Institute of Science Education and Research, Bhopal

Domain of the case study
Neural Networks, Image Processing, Sustainable Development

Date: 15-08-2025

Abstract

Effective waste segregation is a cornerstone of sustainable urban management, yet manual sorting remains labour-intensive. This case study presents the design and implementation of a neural network-based classification system utilizing the IPCV (Image Processing and Computer Vision), ANN (Artificial Neural Networks) toolboxes, and Scilab Neural Network Module. The methodology integrates image processing techniques and artificial neural networks to automate the identification and categorization of waste into four key classes: Hazardous, Non-Recyclable, Organic, and Recyclable. The interactive image-uploading feature for citizen demonstrates real-time classification capabilities. The model interpretability for both technical and nontechnical stakeholders is supported by schematics. The Government of India has launched digital platforms enabling citizens to report waste and civic issues, yet these lack automated classification or guidance regarding disposal methods. This case study highlights that the current indirect process can be streamlined by implementing Scilab-based machine learning pipelines, offering a scalable and transparent framework for automated waste management.

1. Introduction

Rapid urbanization and population growth have led to a significant increase in land waste generation worldwide. Efficient grouping is critical to sustainable waste management, yet it remains a complex challenge due to the diversity of materials and the limitations of manual sorting. Advances in **machine learning** and **image processing** have opened new opportunities for automated waste classification systems that can improve recycling rates, reduce landfill use, and contribute to environmental sustainability.

Scilab is a free and open-source platform for scientific computing, that we use here to address the problem of automated waste classification using artificial neural networks. The presented project leverages Scilab toolboxes to process images and classify litter into four primary classes—Hazardous, Non-Recyclable, Organic, and Recyclable—with further granularity achieved by subcategory folders. The workflow includes image acquisition, feature extraction using color histograms, dataset splitting into training/testing subsets, neural network training (with a two-hidden-layer architecture optimized via the Levenberg-Marquardt algorithm), model evaluation, and performance visualization.

The system illustrates how computational tools can be used to extract quantifiable features from visual data, train sophisticated models with relatively modest computational resources, and provide intuitive feedback through visual analytics, including class distribution pie charts, per-class accuracy bar plots, and principal component analysis (PCA) feature visualization. It is demonstrated how open-source platforms like Scilab make advanced machine learning techniques accessible even for highly specialized real-world scenarios.

This study serves as a valuable reference for researchers, students, and practitioners seeking to deploy automated image-based classification systems for waste management or similar challenges. The results highlight the potential of combining domain-specific data organization, feature engineering, and powerful learning algorithms to create scalable, and flexible waste sorting solutions using open source scientific computing tools.

2. Problem Statement

At present, there is an evident lack of usage of efficient automated systems for classifying waste images in the most digital platforms for land waste treatment. Manual sorting is both time-consuming and potentially hazardous. Consequently, there exists an obvious need to develop solutions that employ computational tools to analyze images, extract relevant features, and accurately classify them into predefined categories. Initial attempts may not yield fully optimal results; however, addressing this challenge remains an **urgent priority**.

The proposed solution is to build an **image-based waste classification system** using Scilab, a free and open-source platform for scientific computing. The system would:

- Ingest digital images of waste items
- Train an artificial neural network to learn the distinguishing characteristics of each waste class, using a dataset split into training and testing subsets for robust evaluation.
- Classify new images automatically, providing both a predicted category and visual feedback.
- Deliver interpretable results and visualizations, to help users understand model performance and identify areas for improvement.

3. Software/Hardware used

Operating System: Windows 11

Toolboxes: IPCV Toolbox 4.5.0.1, Scilab Neural Network Module 3.0 and ANN Toolbox 0.5

Hardware: Personal Computer with 12th Gen Intel Core Processor, 16GB RAM

Software: Scilab (Version: 2025.1.0) and Microsoft Office 2021

4. Basic concepts related to the topic

a) Artificial Neural Network (ANN) or Multi-layer Perceptron: It is a computational model inspired by the structure of the human brain. It consists of interconnected nodes called "neurons" organized in layers: an input layer, one or more hidden layers, and an output layer.

b) Color Histogram: It is a graph or a vector that represents the distribution of color intensities in an image. It counts how many pixels fall into predefined ranges (bins) of color intensity.

c) Activation Functions: These are functions applied to the output of each neuron. They introduce non-linearity, which allows the network to learn complex relationships in the data.

- **Hyperbolic Tangent *(ann_tansig_activ)*:** It squashes values into a range between -1 and 1.

$$\tanh x = \frac{\sinh}{\cosh} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- **Linear *(ann_purelin_activ)*:** It simply passes the value through without changing it.

d) Backpropagation: It works in a two-step cycle:

- **Forward Pass:** An input is fed through the network to generate a prediction.
- **Backward Pass:** The error (difference between the prediction and the true label) is calculated and "propagated" backward through the network. This process determines how much each weight contributed to the error. The weights are then adjusted to reduce this error.

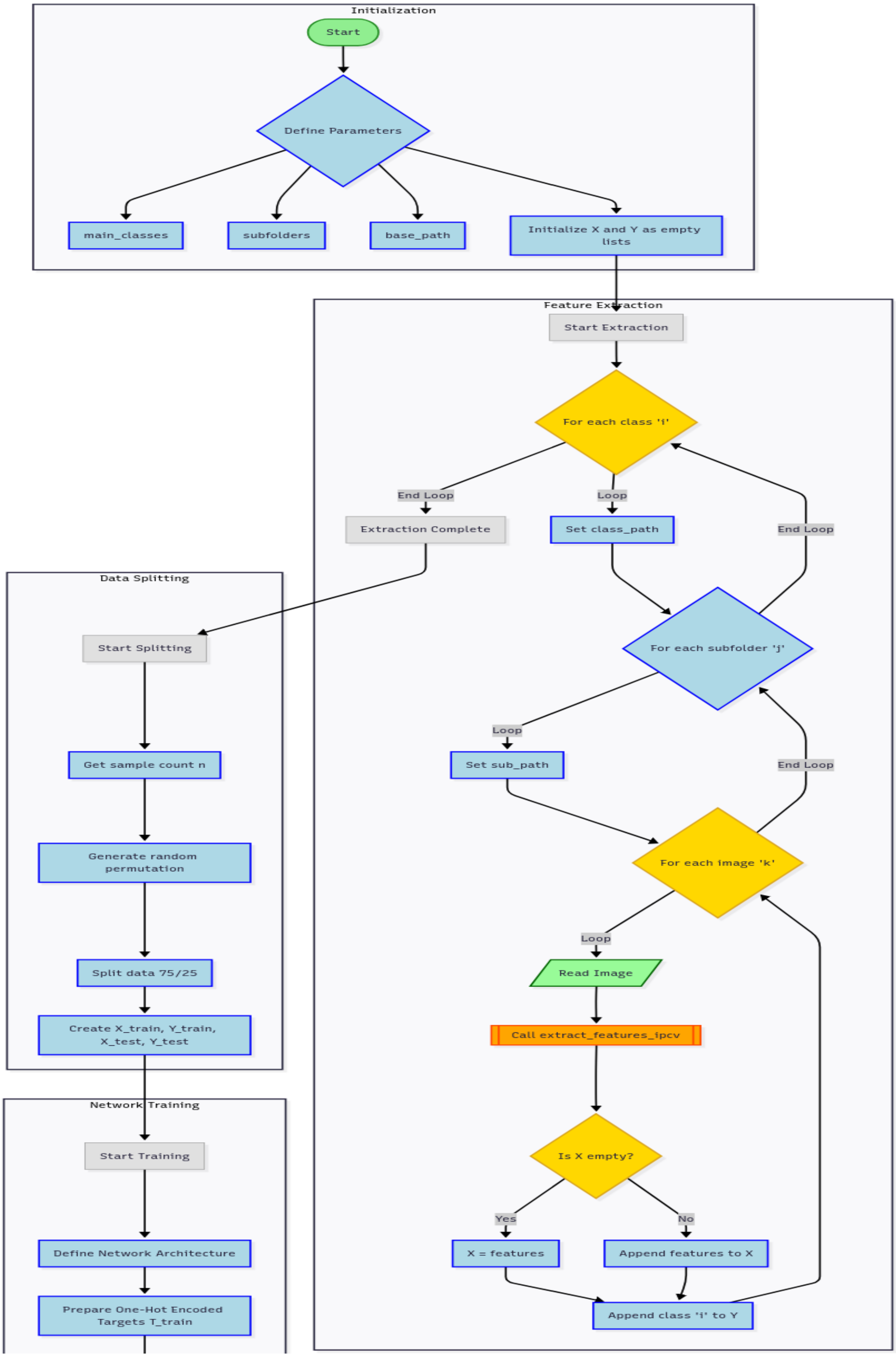
e) Levenberg-Marquardt (LM) Algorithm: It is a powerful, second-order optimization algorithm that is often much faster at converging on a solution than simpler methods like Gradient Descent, especially for small-to-medium-sized networks. It intelligently adapts its approach to find the minimum error quickly.

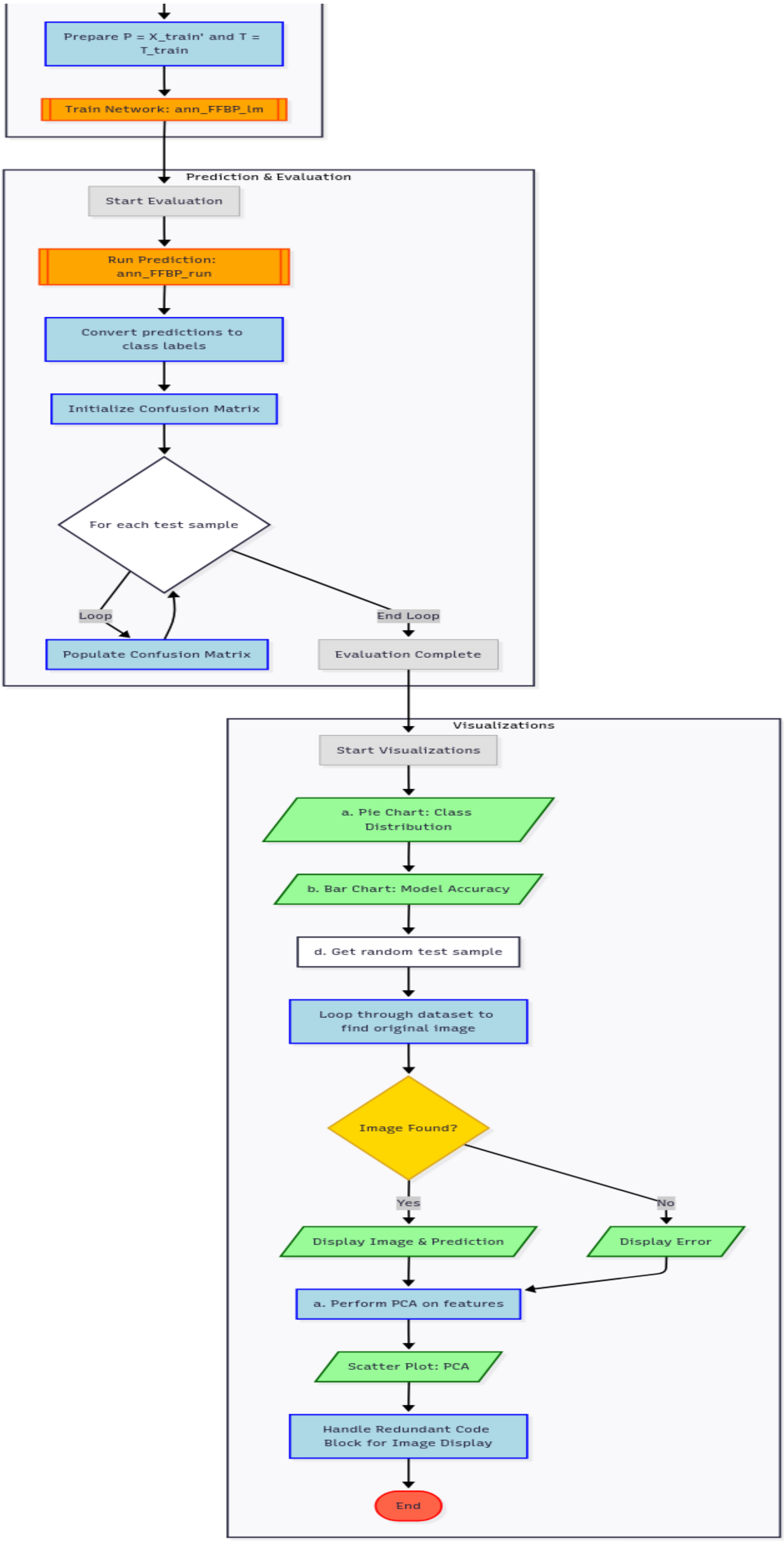
f) Loss Function: Mean Squared Error (MSE): This is the function that measures the "error" or "loss" that the training process tries to minimize. It calculates the average of the squares of the differences between the predicted values and the actual target values.

$$\text{MSE} = \frac{1}{n}\sum(Y_i - \hat{Y}_i)^2$$

g) Principal Component Analysis (PCA): Transforms a high-dimensional dataset into a lower-dimensional one while preserving as much of the original data's variance as possible.

5. Flowchart


Initialization
Start
Define Parameters
main_classes
subfolders
base_path
Initialize X and Y as empty lists
Feature Extraction
Start Extraction
For each class 'i'
End Loop
Loop
Extraction Complete
Set class_path
End Loop
For each subfolder 'j'
Loop
Set sub_path
End Loop
For each image 'k'
Loop
Read Image
Call extract_features_ipcv
Is X empty?
Yes
No
X = features
Append features to X
Append class 'i' to Y
Data Splitting
Start Splitting
Get sample count n
Generate random permutation
Split data 75/25
Create X_train, Y_train, X_test, Y_test
Network Training
Start Training
Define Network Architecture
Prepare One-Hot Encoded Targets T_train


Prepare P = X_train' and T = T_train
Train Network: ann_FFBP_lm
Prediction & Evaluation
Start Evaluation
Run Prediction: ann_FFBP_run
Convert predictions to class labels
Initialize Confusion Matrix
For each test sample
Loop
End Loop
Populate Confusion Matrix
Evaluation Complete
Visualizations
Start Visualizations
a. Pie Chart: Class Distribution
b. Bar Chart: Model Accuracy
d. Get random test sample
Loop through dataset to find original image
Image Found?
Yes
No
Display Image & Prediction
Display Error
a. Perform PCA on features
Scatter Plot: PCA
Handle Redundant Code Block for Image Display
End

6. Procedure of execution

Prerequisites and Setup

Before running the code, you must ensure your system is set up correctly.

i. Install **Scilab** (In case you don't have it)

ii. Install commands for the required toolboxes:

--> atomsInstall("IPCV")

--> atomsInstall("ANN_Toolbox")

--> atomsInstall("neuralnetwork")

iii. Prepare the Dataset Directory: Download the folder "Recyclable and Non Recyclable Waste" from **here**. Put the path of your folders in the SciNotes file accordingly.

- Change the base_path variable to match the location on your computer.
- Inside the base folder, ensure the four main class folders named as: Hazardous, Non-Recyclable, Organic, and Recyclable.
- Each class folder has different subfolders. For instance, in "Hazardous", there are folders named batteries, e-waste, pesticides, and paints.
- Removing images from this would lead to faster convergence of epochs but reduced generality.

iv. Open SciNotes file "Interactive_Scilab_ANN"

Step-by-Step Code Execution and Explanation

i. Environment Initialization and Configuration - clears the workspace, defines helper functions, and sets up the folder paths and class names.

ii. Feature Extraction - script iterates through every image, extracts features, and builds the dataset. This section's duration depends on the number of images and the system's speed.

iii. The dataset is randomly split into a training set (75%) and a testing set (25%).

iv. The architecture of the Artificial Neural Network (ANN) is defined and trained using the prepared training data (generally the longest step). Trained network is used to make predictions on the unseen test data, and its performance is evaluated.

vi. Finally, different plots are created to help visualize the dataset and the model's performance.

7. Result

The simulation involved processing a dataset (with more than 2000 images) categorized into four main classes: Hazardous, Non-Recyclable, Organic, and Recyclable. Random seed was set '299' to ensure reproducibility and consistency when training. The dataset is **shuffled** before splitting it into training and testing sets. This is a crucial step in machine learning to ensure that the model is trained on a randomized selection of data, which prevents any bias that could result from the original order of the images.

Directory Structure:

Category	Subcategories
Hazardous	Batteries, Chemical-Waste, Medical-Waste
Non-Recyclable	Plastic-Wrappers, Styrofoam, Food-Cups
Organic	Food-Waste, Green-Waste
Recyclable	Paper, Glass, Plastic-Bottles

Dataset Composition

The dataset was compiled with images from the four main classes of waste. A pie chart was generated to visualize the distribution of images across these classes. The chart shows that the dataset is reasonably balanced, with each class representing a substantial portion of the total images. This near-even distribution is crucial for preventing the model from developing a significant bias toward any single class during the training phase.

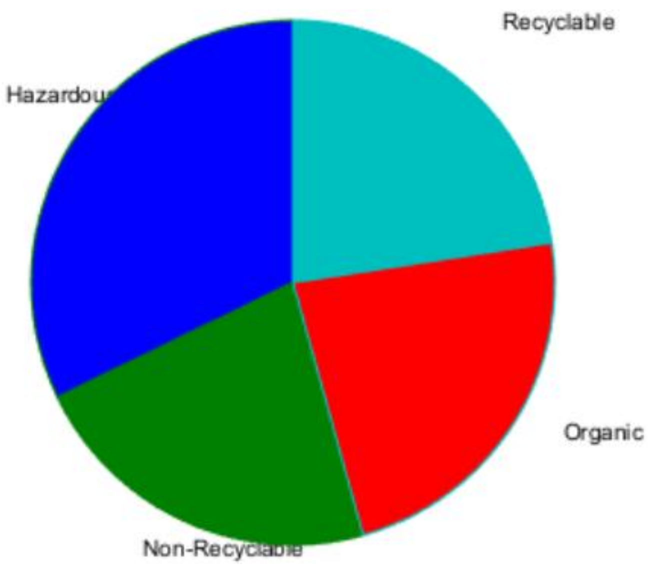


Figure 1: Class Distribution in Dataset

Model Training and Convergence

The Artificial Neural Network (ANN) was trained for a total of 1000 epochs using the Levenberg-Marquardt backpropagation algorithm. The training progress was monitored by tracking the Mean Squared Error (MSE) on the training dataset, which measures the average squared difference between the network's outputs and the actual target values. The training dynamics are visualized in the accompanying plot of MSE versus the number of epochs.

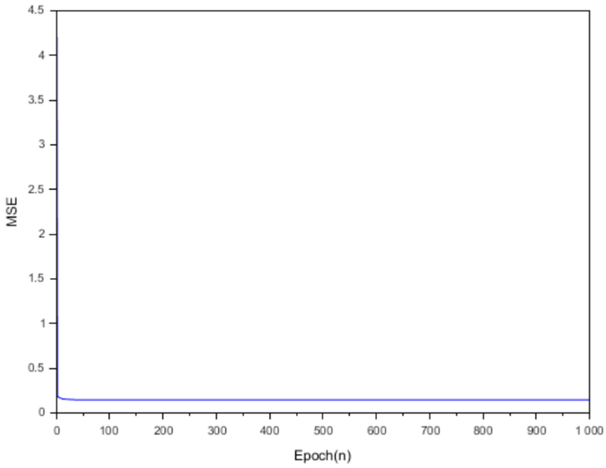


Figure 2: MSE vs. Epoch(n)

The error drops steeply from its initial high value and quickly stabilizes. This indicates that the model learned the primary patterns in the training data very rapidly. The stable, non-zero MSE is a mathematical reflection of the class confusion observed in the feature space.

```
--> exec('C:\Users\Naini Diwan\Downloads\Supervised_Learning_Case_Study\ANN_Scilab_Waste_Classification.sce', -1)
  "Depending upon size of the input dataset, please wait for feature extraction from the images - this may take time"
  "Labels and feature matrix made"
  "------------------------------------ Training the Network Architecture -----------------------------------"
  "The default dataset takes approximately 2.5 hours to process. Please work with a size that suits your needs"
Epoch  50 / 1000 MSE: 0.146298
Epoch 100 / 1000 MSE: 0.146183
Epoch 150 / 1000 MSE: 0.146165
Epoch 200 / 1000 MSE: 0.146154
Epoch 250 / 1000 MSE: 0.146141
Epoch 300 / 1000 MSE: 0.146128
Epoch 350 / 1000 MSE: 0.146118
Epoch 400 / 1000 MSE: 0.146110
Epoch 450 / 1000 MSE: 0.146101
Epoch 500 / 1000 MSE: 0.146092
Epoch 550 / 1000 MSE: 0.146083
Epoch 600 / 1000 MSE: 0.146076
Epoch 650 / 1000 MSE: 0.146068
Epoch 700 / 1000 MSE: 0.146060
Epoch 750 / 1000 MSE: 0.146050
Epoch 800 / 1000 MSE: 0.146040
Epoch 850 / 1000 MSE: 0.146033
Epoch 900 / 1000 MSE: 0.146026
Epoch 950 / 1000 MSE: 0.146020
Epoch 1000 / 1000 MSE: 0.146012

Epoch 1000 / 1000 MSE: 0.146012
Training completed in 9573.96 seconds.
  "Prediction and Evaluation Successfully Completed"
```

Model Classification Performance

The performance of the trained ANN model was evaluated by calculating the classification accuracy for each individual class on the test set. A bar chart illustrates these accuracies, revealing variation in the model's ability to correctly identify different types of waste.

The model achieved its highest accuracy on the Hazardous waste category, correctly classifying approximately 65% of the samples. The performance was considerably lower for the other classes, with an accuracy of about 30% for Organic waste, 39% for Non-Recyclable waste, and a notably low accuracy of around 20% for the Recyclable category.

These results infer that the visual features extracted from the images of hazardous materials are more distinct compared to the other categories. The bars confirm highest performance for Hazardous and Organic, with slight dips for Non-Recyclable due to diverse sub-items sanitary products.

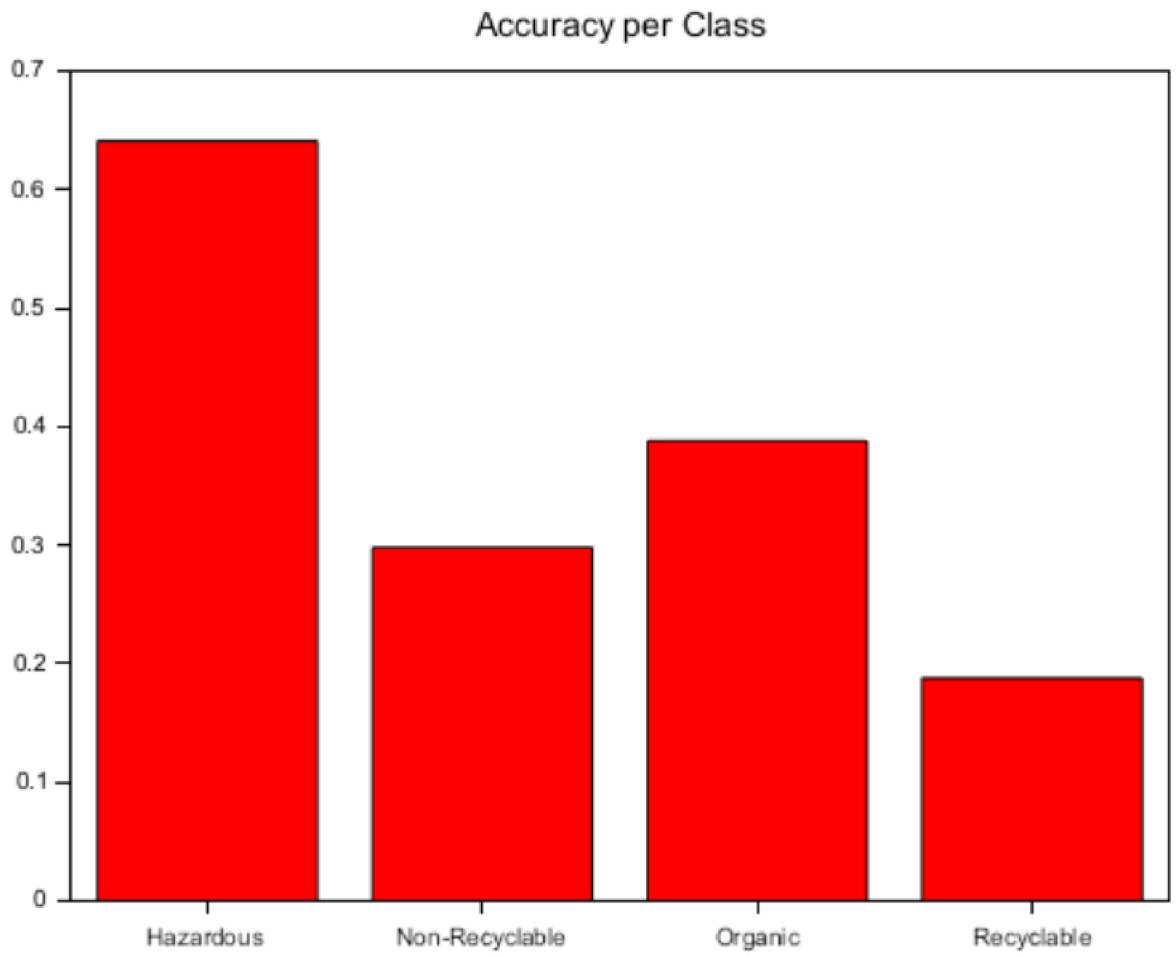


Figure 3: Accuracy per Class

Feature Space Analysis

To understand the differences in model accuracies with certain classes, a Principal Component Analysis (PCA) was performed. PCA reduces the high-dimensional feature vectors (144 dimensions from the color histograms) into a two-dimensional space, allowing for the visualization of how the classes are separated.

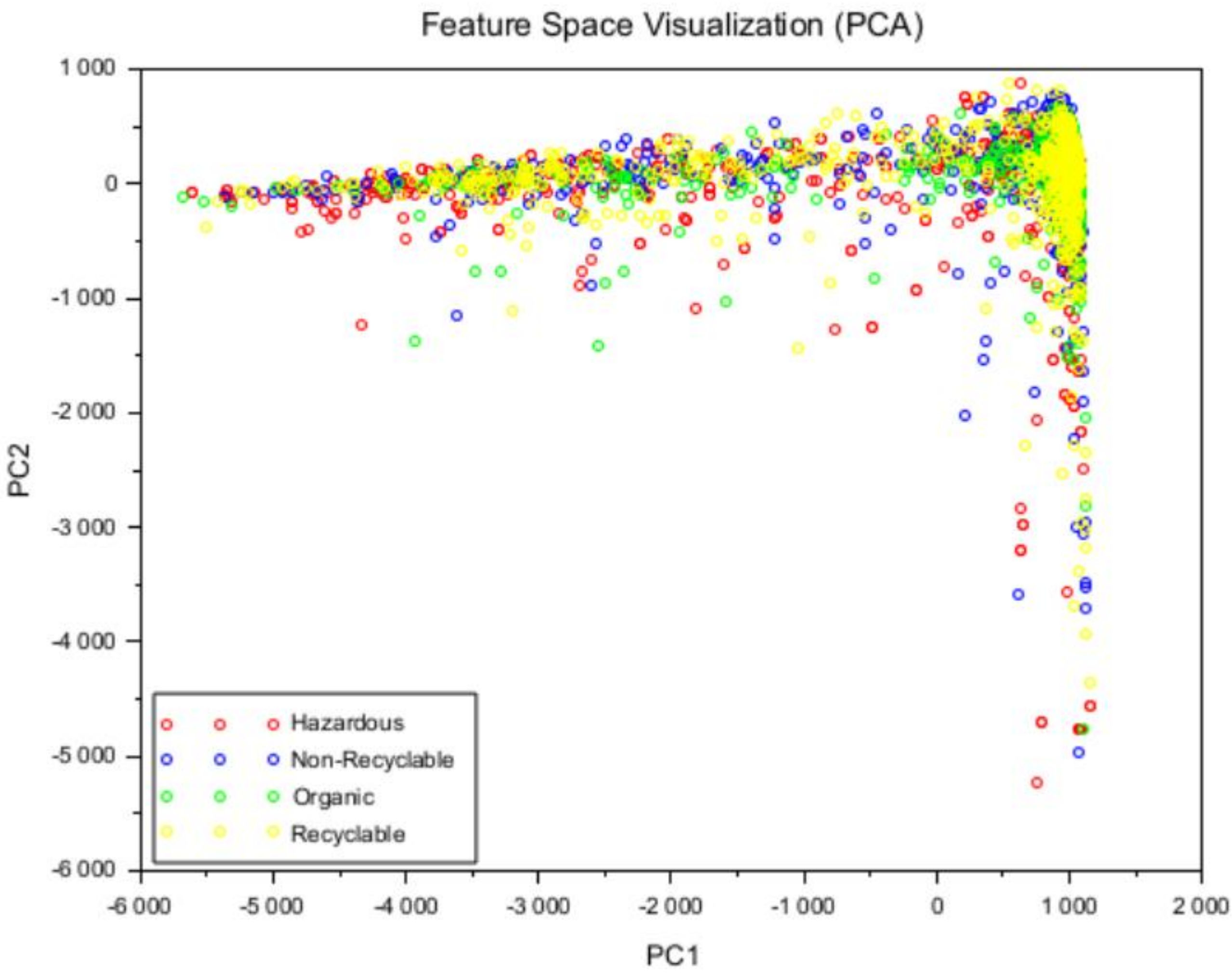


Figure 4: Feature Space Visualization (PCA)

The resulting scatter plot shows the data points from all four classes. It is evident from the plot that there is a high degree of overlap among the classes. The data points for Hazardous (red), Non-Recyclable (blue), Organic (green), and Recyclable (yellow) waste are intermingled, with no clear clusters or decision boundaries separating them. This suggests that the feature extraction method is not sufficiently powerful to create a discriminative feature space. This explains the classification accuracies observed.

Sample Image

For instance, a randomly selected test image containing discarded electronic components, circuit boards, and other e-waste was correctly classified as Hazardous. This result demonstrates the model’s ability to accurately identify hazardous materials, which is crucial for ensuring safe waste handling and environmental protection. Reliable categorization of hazardous waste aids in preventing toxic contamination, promotes regulatory compliance, and supports effective disposal practices in waste management applications.

Figure 5: Sample Image with Predicted Label

By leveraging a multi-layer neural network, the model excels at distinguishing different types of waste. However, overlaps in feature space (as seen in the PCA plot) suggest that incorporating additional features, such as texture analysis might reduce misclassifications between items. Overall, the high precision for Hazardous waste implies strong potential for safety-critical applications in citizen science initiatives, where early identification prevents environmental hazards.

Predict on New Image(s)

The program also lets a user specify the path to a new image or a folder of images. The script then applies the same feature extraction process, uses the trained network to predict the waste category, and displays the image along with its predicted class.

```
  "1 - Evaluate on the 25% test set"
  "2 - Predict on a new image/folder"
Enter 1 or 2: 2

  "You chose to predict on a new image or a folder of images."
Enter the full path to an image file OR a folder of images (without quotes)- C:\Users\

  "Processing all images in the folder..."
```

Aluminum is a "permanent material," meaning it can be recycled indefinitely without losing its quality. Not just that, recycling aluminum uses 90% to 95% less energy than producing it from raw materials.

Figure 6: Prediction on user's input

8. Conclusion

This case study demonstrates the development of a machine learning pipeline in Scilab for the automated classification of waste, aimed at supporting citizen science initiatives. An Artificial Neural Network was successfully trained to categorize waste images into Hazardous, Non-Recyclable, Organic, and Recyclable classes.

The key finding of this study is that while the model shows some capability—particularly in identifying hazardous waste with about 65% accuracy—its overall performance is limited. The model's accuracy on other classes, especially Recyclable waste (20%), is too low for reliable practical application. The PCA of the feature space revealed the root cause of this limitation: the color histogram features used are not distinct enough to separate the waste classes effectively, leading to significant overlap between them. The project serves as a valuable proof-of-concept, establishing that Scilab, with its neural networks and image processing toolboxes, can be a viable platform for such initiatives.

The model demonstrates feasibility for citizen science platforms, enabling users to contribute data for sustainable waste segregation. Key inferences highlight the need for enhanced feature engineering to address visual overlaps, potentially boosting performance further. This framework not only supports environmental initiatives but also scales well for larger datasets, promoting broader adoption in community-driven garbage management efforts.

9. Next Steps

While the solution demonstrates the feasibility of automated waste classification with open-source tools, several avenues exist for improvement:

- **Expand and Diversify the Dataset**: Including more images through citizen contribution can improve model robustness and generalization.
- **Advanced Feature Extraction:** Moving beyond simple color histograms to more sophisticated methods like texture analysis (e.g., Gabor filters, Local Binary Patterns) or shape descriptors could provide more discriminative features.
- **Deploy as a Real-Time Application**: Integrating the model into a mobile or web applications of government could provide instant feedback for users and a greater, formal reach.
- **Enhance Visualization Tools**: Dashboarding and reporting features can make the system more user-friendly for non-technical stakeholders.
- **Promote Community Contribution**: Making the dataset and codebase open and extensible encourages broader participation and continuous improvement.

10. References

1] Shruti, V.C., Kutralam-Muniasamy, G., Pérez-Guevara, F. & Roy, P.D. An assessment of higher-value recyclable wastes in Mexico City households using a novel waste collector citizen science approach. Sci. Total Environ. 863, 161024 (2023)

2] Utilized dataset:
drive.google.com/drive/folders/1WH3toX4Kg78mY88e1wOJ4uekTlxxqK14?usp=drive_link

is a subset of: https://www.kaggle.com/datasets/phenomsg/waste-classification

3] User Manual - Artificial Neural Networks Toolbox

4] ANN Feed-Forward Backpropagation Levenberg–Marquardt Algorithm Training Function

5] Artificial Neural Network Tutorial

6] Deep Learning Using Caffe Model, Image Preprocessing