Image-Based Waste Classification with Visual Explanations

Maya Kusumakar[1] Shifanaaz Fazalur[1]
[1]College of Engineering, University of California, Davis, CA, USA
{Mkusumakar, sfazalur}@ucdavis.edu

Abstract

Recycling contamination is a major barrier that causes waste. It is often driven by public uncertainty about how to sort everyday items. To address this issue, we developed an image based decision system that classifies objects, which is given as RGB images. We classify it as either recyclable or non-recyclable (organic). We used a waste classification dataset from Kaggle which has provided these labeled images in two categories (recyclable and organic). We will also be using Kaggle's provided code to train (1) a baseline of convolutional neural network (CNN) and (2) a transfer-learning model based on MobileNetV2 for binary classification. We will expand our project with a computer-vision pipeline. The data are divided into a training split (70%) and a held-out evaluation split (30%). For the baseline CNN, the model divided the evaluation split into validation and test subsets. Model performance is evaluated using accuracy, macro-F1 score, confusion matrices, ROC-AUC, and precision-recall curves.

We extended the classifier to provide visual explanations with computer vision. This was done by converting the image into grayscale, applying Gaussian blurring to reduce the noise, and then detecting edges. The model will detect the object most likely responsible for the prediction and draw a bounding box around the object. This is a visual cue. We also performed a small test on additional Google images that were never seen during training. Our model aims to support communities by providing an interpretable and lightweight tool. It will help individuals correctly identify recyclable materials in order to reduce waste.

1. Introduction

In the recent years, we have seen increasing use of computer vision and machine learning for environmental challenges. One important area in the recycling aspect of the environment is where individuals are often unsure how to dispose of everyday items. This can lead to contamination of recycling streams, and a loss of recyclable material.

Our goal is to design a relatively lightweight system that is interpretable. We used a baseline convolutional neural network (CNN) implemented in TensorFlow/Keras, and MobileNetV2-based CNN in PyTorch using transfer learning. Both models use RGB images as input and classify items as recyclable or non-recyclable (organic).

We extended the classifier with a visual explanation. The model explains it's decision with a bounding box drawn around the object. This shows the region that is associated with the model's decision.

We train and evaluate our models on the Kaggle "Waste Classification Data" dataset. This dataset contained thousands of labeled images in organic and recyclable categories.

2. Dataset

We use the Waste Classification Data: dataset available on Kaggle. The dataset consists of RGB images of items labeled as either organic or recycle. We treated these labels as a binary classification task with organic waste as label 0 and recyclable items as label 1.

2.1. Metadata

- **Classes:** 2 (organic and recyclable)
- **Color mode:** RGB
- **File type:**.jpg

2.2. Train/Validation/Test Split

To train and evaluate our models, we first split the Kaggle dataset into a training split (approximately 70% of the images) and a held-out evaluation split (approximately 30%).

For the baseline CNN, the held-out evaluation split is further divided into validation and test subsets of equal size.

3. Proposed Methodology

We had two parts to this project.

1. Two CNN classifiers compared for binary waste classification
2. Object highlight bounding box as a visual explanation.

3.1. Preprocessing and Data Augmentation

For both models, we load the images in three-channel RGB format. The model applies resizing and normalization.

Baseline CNN (Keras). For the baseline CNN, we use Keras ImageDataGenerator with:
- resizing to 65×65 pixels,
- rescaling pixel values to $[0, 1]$,
- data augmentation: random rotations, width/height shifts, shear, zoom, and horizontal flips.

MobileNetV2 (PyTorch). For MobileNetV2, the model uses torchvision.transforms with:
- resizing to 224×224 pixels,
- random rotation and horizontal flip for training,
- conversion to tensors,
- normalization using mean and standard deviation.

3.2. Baseline CNN (TensorFlow/Keras)

The first model is a baseline convolutional neural network implemented in TensorFlow and Keras. It operates on $65 \times 65 \times 3$ inputs. It construct a sequence of convolutional blocks consisting of a 2D convolution with 3×3 filters, batch normalization, ReLU activation, and max pooling. The number of filters increases across blocks 32, 64, 128, and 256.

The feature maps are flattened and passed through a fully connected layer with 4096 units and ReLU, dropout with rate 0.5, a second fully connected layer with 4096 units and ReLU, a final output neuron with sigmoid activation for binary classification.

The model uses Adam optimizer and binary cross-entropy loss. The metrics tracked are accuracy, precision, recall, true/false positives and negatives.

3.3. MobileNetV2 Transfer Learning (PyTorch)

The model we chose is transfer learning with MobileNetV2. The model is initialized with ImageNet-pretrained weights.

The model freezes pretrained layers and replace the final classifier layer with a new linear layer to output for the two classes. The model uses weighted cross-entropy loss with class weights. The loss is optimized with Adam with learning rate of 10^{-4}. The training loss and accuracy, validation loss and accuracy, precision, recall, macro-F1, ROC–AUC is tracked per epoch.

The best model is saved with the lowest validation loss.

3.4. Evaluation Metrics

We evaluate classifier performance using:
- **Accuracy** (overall fraction of correctly classified images),

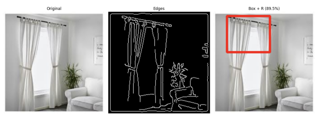

Figure 1. Example of the object-highlighting pipeline: original image, edge map, and final bounding-box overlay with predicted label and confidence.

- **Macro-F1 score** (average F1 across organic and recyclable classes),
- **Confusion matrix** (counts of true/false positives and true/false negatives),
- **ROC–AUC** (area under the ROC curve),
- **Precision–recall curve** and average precision.

4. Computer Vision Extension: Object Highlighting

We go beyond classification. This is done by providing visual explanations of the model's decisions with object detection box.

We do this by the following pipeline:
1. Convert the image to grayscale.
2. Apply Gaussian blur to reduce noise and smooth small details.
3. Run the Canny edge detector to obtain an edge map.
4. Extract contours using OpenCV.
5. Remove contours that are too small (below an area threshold)
6. For filtered contours that remain, take the union of their bounding boxes. Draw the resulting bounding box on the original image.

Figure 1 depicts the example of the object-highlighting pipeline.

5. Experiments and Results

The model train both the baseline CNN and the MobileNetV2 transfer-learning model on training and evaluation set.

5.1. Training Dynamics

For both models, training loss decreases and training accuracy increases over epochs. The baseline CNN shows noisy validation curves at the beginning (spikes in validation loss and accuracy). It stables in high 0.80 showing sensitiveness to validation. MobileNetV2 has smooth training and validation curves, with accuracy stabilizing around 0.93, showing better generalization.

5.2. Results and Classification Performance

We compare the baseline CNN and the MobileNetV2 model using accuracy, macro-F1, and ROC-AUC. MobileNetV2

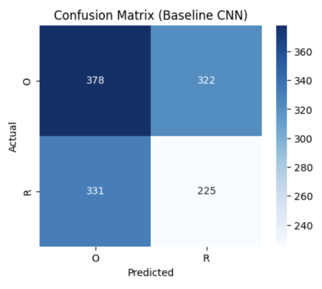

Figure 2. Baseline CNN Confusion Matrix

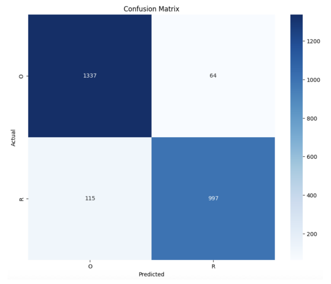

Figure 3. Confusion matrix for the MobileNetV2 transfer-learning model on the test set

generally achieves slightly higher accuracy of 92.9 percent and macro-F1 of 0.9286 than the baseline CNN accuracy of 92.88 percent and macro - F1 of 0.4080. This shows that the transfer learning is better model due to the ImageNet pretraining.

The Figure 2 and 3 show the confusion matrix for each model. For the Baseline CNN, the number of recycle items are mixed up with organic waste. The MobileNetV2 show that most images are correcly identified with some mistakes.

ROC and precision - recall curves show the trade-off between true-positive and false-positive rates. The baseline CNN has ROC-AUC of 0.48 and a flat precision-recall curve. This shows that it has a medium ability of classifying. The MobileNetV2 has ROC-AUC of 0.98 and has high precision-recall curve. This highlights how it can classify images into classes with more reliability.

Using MobileNetV2, we did a small test. We also ran on a few Google images that were never seen during training. The model still produced reasonable labels and bounding boxes. This shows that it generalizes beyond the Kaggle dataset.

6. Conclusion and Future Work

We developed an image-based system for classifying waste items as recyclable or non-recyclable (organic). Two models (baseline CNN in TensorFlow/Keras and a MobileNetV2 transfer-learning model in PyTorch) were trained. Both operate on RGB images. They are evaluated using accuracy, macro-F1 score, ROC-AUC, precision-recall curves, and confusion matrices on validation and test sets.

We extended the project with a computer vision pipeline. This pipeline highlights the main object in each image. This pipeline converts the image to grayscale, applies Gaussian blur Canny edge detection, and contour analysis. It draws a bounding box around the object. This result is shown in Figure 1 with the classifier's predicted label and confidence. The purpose of this is to give users a visual cue about why the model made its decision.

For our future work, we plan to deploy our model as a web based or mobile app. We also plan to collect and test on more images.

7. Project Roadmap

Timeline Overview:

Week	Milestones
1-3	Team formation and role assignment. Dataset selection. Review of sample images and drafting of project goals (binary classification + object highlighting).
4	We Finalized modeling scope. Decided to use a baseline CNN and a MobileNetV2 transfer-learning model. Outlined the computer-vision pipeline for bounding-box highlighting.
5-6	Implemented image preprocessing pipelines in TensorFlow/Keras and PyTorch.
7-8	Trained the baseline CNN baseline. Trained the MobileNetV2 transfer-learning model using the validation set. Logged loss/accuracy curves and computed accuracy, macro-F1, ROC-AUC, and confusion matrices.
9	Implemented and refined the object-highlighting pipeline (grayscale conversion, Gaussian blur, edge detection, contour-based bounding boxes).Integrated visualizations showing bounding boxes with model predictions and confidence scores.
10	Finalized and integrated all report sections. Cleaned and organized code, prepared figures and visualizations, incorporated feedback from TAs/Professors, and submitted all deliverables including the video presentation.

Team Contributions:

- **Shifanaaz Fazalur and Maya Kusumakar** together trained, tested the models (baseline CNN and MobileNetV2 transfer learning). We generated metrics (accuracy, macro-F1, ROC-AUC, confusion matrices), designed and coded the object-highlight pipeline. We produced visuals, worked on the report and presentation materials.

8. References:

Kaggle Dataset: https://www.kaggle.com/datasets/techsash/waste-classification-data

MobileNetV2 Model Code: https://www.kaggle.com/code/aakansh8/waste-management-cnn

Baseline CNN Code: https://www.kaggle.com/code/nicholaslee26/waste-classification-using-neural-network