Jay Siegfried, Byung Woong Ko
ECE 629
5/9/2025

Classifying Recyclable items using Machine Learning

Abstract

Proper waste classification is extremely crucial for sustainable waste management and reducing landfill use. However, many countries, including the United States, lack the infrastructure or public participation required for effective recycling. This project proposes a machine learning-based image classification system deployed on an embedded device to automatically identify recyclable materials at the point of disposal. By using transfer learning with lightweight convolutional neural network (CNN) architectures, the system achieves accurate real-time classification of four common recyclable categories: plastic, paper, glass, and metal cans. The final model is deployed on a Jetson Nano Development Board, demonstrating the viability of low-power, low-cost intelligent recycling systems as a real world application.

Introduction and Background

Waste management remains an important challenge worldwide. In the United States, the majority of trash is either sent to landfills or incinerated, while countries such as South Korea and Singapore have implemented advanced classification infrastructure to streamline recycling. This categorization is done by the citizens, that dispose of recycling/waste into many different bins. However, replicating such infrastructure in the U.S. is costly and would require significant changes in public behavior. Therefore, a low-cost, behavior-independent solution is necessary. This project presents a machine learning-powered IoT solution capable of classifying recyclable items without requiring users to change how they dispose of waste.

Convolutional Neural Networks are the most common type of ML model that is used for image classification, such as this task. These models use multiple convolutional layers that extract features from the image using a “convolutional kernel” that gives weights to neurons. These completed feature maps emphasize certain characteristics, and the network learns which features/filters are most important to identify more complex patterns. We utilize two different pre-trained CNN's in this project that were trained on the "ImageNet" dataset to identify particular objects in images: MobilenetV2 and EfficientNetB0. MobilenetV2 is designed for embedded systems and is more lightweight, while EfficientNetB0 is more robust and has more parameters but is still designed to be minimal.

Proposed System and Setup

The proposed system employs transfer learning using MobileNetV2 and EfficientNetB0 models to classify four categories of recyclable waste: can, glass, paper, and plastic. The model is trained on a combined dataset drawn from the TrashNet Dataset[1] and the Recycle Classification Dataset[2]. The Recycle Classification Dataset has around 9,400 images, and the TrashNet dataset has around 2,500 images, but we only used 1,500 that correlated to our classes of paper, plastic, cans, and glass. To enhance model robustness and generalization, data augmentation techniques such as horizontal flipping, rotation, zooming, and translation were applied to the combined dataset. Before being used by the models, the images are set to be 128x128 pixels, and each pixel value is normalized to a value between 0-1. The performance of both models will be analyzed, and the better model is deployed on an NVIDIA Jetson Nano, enabling real-time inference in low-resource environments.

Model Development and Optimization

The model training process involved two main phases. In the feature extraction phase, the pre-trained base (MobileNetV2 and EfficientNetB0) was frozen, and a custom classification head was added consisting of a global average pooling layer, a dense ReLU layer, dropout for regularization (50%), and a final softmax layer to get the scores of each class for an image. In the fine-tuning phase, the base model was unfrozen to allow deeper adaptation to the recycling dataset. A reduced learning rate was used to protect the pretrained weights. Model performance was evaluated using a confusion matrix, per-class accuracy, and detailed classification reports.

Experimental Results

Both the models using MobileNetV2 and EfficientNetB0 were tested on the same dataset, and the model using MobileNet had a 84% accuracy score, and the model using EfficientNet had a 73% accuracy score, and the MobileNet model also had better scores for each individual class accuracy as well.

	precision	recall	f1-score	support
can	0.8634	0.8879	0.8755	776.0000
glass	0.7268	0.8504	0.7838	341.0000
paper	0.9750	0.8069	0.8830	725.0000
plastic	0.7531	0.8185	0.7845	518.0000
accuracy	0.8424	0.8424	0.8424	0.8424
macro avg	0.8296	0.8409	0.8317	2360.0000
weighted avg	0.8537	0.8424	0.8446	2360.0000

	precision	recall	f1-score	support
can	0.7834	0.7642	0.7736	776.0000
glass	0.5315	0.7918	0.6360	341.0000
paper	0.8422	0.7655	0.8020	725.0000
plastic	0.7110	0.5985	0.6499	518.0000
accuracy	0.7322	0.7322	0.7322	0.7322
macro avg	0.7170	0.7300	0.7154	2360.0000
weighted avg	0.7492	0.7322	0.7353	2360.0000

Figure 1: MobilenetV2 (Left) vs EfficientNetB0 (right) Performance

In addition to having a higher accuracy score, the model using MobilenetV2 also had fewer parameters than the one using EfficientNet, as can be seen in Figure 2 below. Fewer parameters means that the model itself is faster and is smaller overall, making it better suited for mobile and embedded applications. As a result, due to both accuracy and parameter count, the model using MobileNetV2 was chosen to be saved and deployed for our real time system.

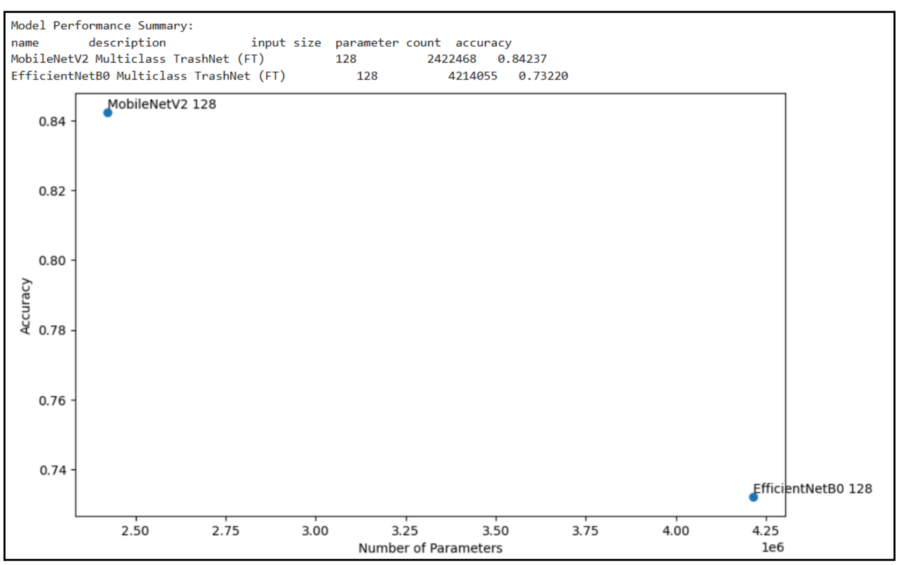

Figure 2: Model Accuracy vs. Parameter Count

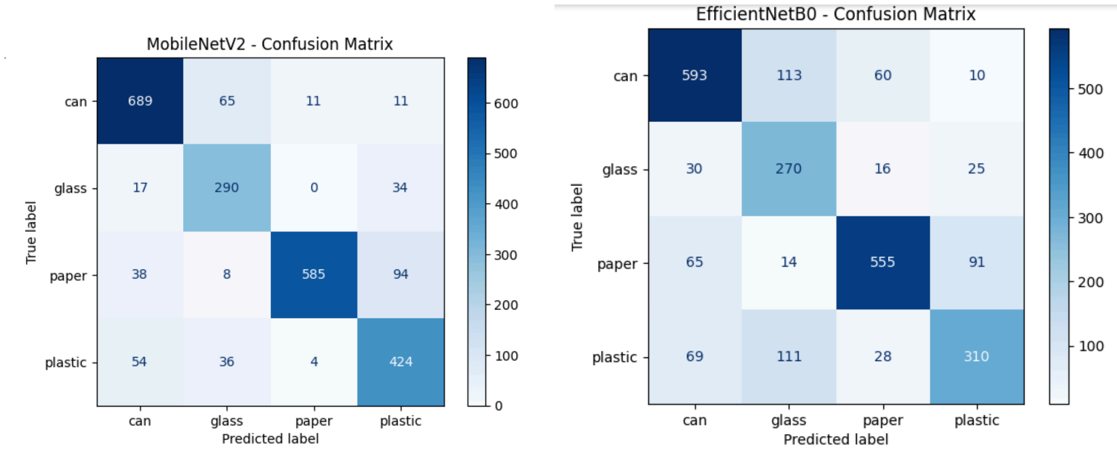

Figure 3: Confusion Matrices of Both Models

Another key takeaway from testing both of these models is the models' performance in categorizing glass compared to the other classes. As can be seen in both Figure 3 and Figure 1, glass has the lowest class accuracy with both of the models, being only around 73% for the chosen final model.

<u>Embedded Deployment</u>

The model was initially saved in ".h5" format and then converted to ".onnx" format for deployment. ONNX (Open Neural Network Exchange) offers a lightweight and widely supported format optimized for performance, especially on embedded systems like the Jetson Nano. The model was transferred via USB and integrated with a live camera feed using the CSI-Camera GitHub repository[3]. The model overlays live predictions onto the video feed, simulating a real-world recycling classification system, as can be seen in Figure 4 below.

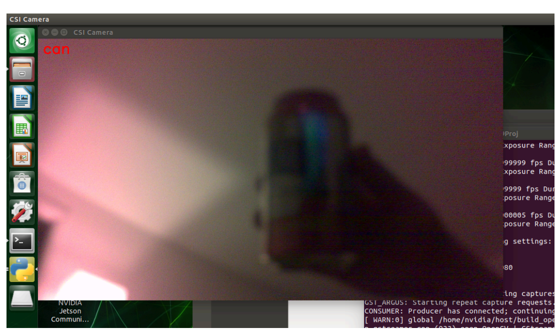

Figure 4: Live Classification on the Jetson Nano

During testing, the model successfully classified objects in real time, although some issues occurred. One issue was the model defaulting to "plastic" when no objects were present in the camera frame, and only changing when an item was clearly presented in the middle of the frame. Additionally, the model showed difficulty in recognizing glass from plastic, likely as a result of the camera quality not being as clear as the training images in the dataset. Camera latency and a red image tint were also issues that could be improved upon.

Conclusions and Future Work

This project demonstrates the feasibility of deploying machine learning models for recycling classification on embedded systems. MobileNetV2 provided an optimal trade-off between model size and accuracy, making it suitable for real-time inference on devices like the Jetson Nano. Future enhancements include introducing a two-stage classifier to first detect whether an item is trash or recyclable and then determine the recyclable category. Additionally, implementing federated learning in a real time scenario could allow for distributed model updates from multiple devices and recycling plants. Improving the image quality and camera integration is also a critical next step for improving accuracy.

Works Cited

[1] Ozkefe, Feyza. "Trashnet." Kaggle, 2021.
https://www.kaggle.com/datasets/feyzazkefe/trashnet

[2] Wang, Jinyeong. "Recycle_Classification_Dataset." Kaggle, 2020.
https://www.kaggle.com/datasets/jinfree/recycle-classification-dataset

[3] JetsonHacksNano. "Jetsonhacksnano/CSI-Camera." GitHub. Accessed 7 May 2025.