Computer Vision for Waste Classification

Russell Tan Scott Susanto Pablo Tayun-Mazariegos
Link to Github: https://github.com/Russellkusuma/TrashRecogntion/tree/main

1. Introduction

Recycling is a well known solution to saving landfill space, however, many people do not know how or often make mistakes in sorting trash. Sorting recyclables before reaching the recycling facility is crucial for effective recycling as it keeps costs down by preventing clogged machinery and the need of manual sorting in the facilities. If contaminants were to pass, the final product would be deemed unsatisfactory and thrown into the landfill rather than being reused. This experiment's purpose is to help improve models designed to classify six different forms of waste: glass, cardboard, metal, paper, plastic and trash.

The classifier, which consists of some form of CNN, will take an image input containing a single piece of waste on a white background (or any solid color background). The model should then classify the object into one of six possible waste categories mentioned prior.

2. Related Work

"Classification of Trash for Recyclability Status" is a report written by Gary Thung and Mindy Yang (Yang & Thung, 2016). It has good research on waste classification using machine vision. Their model classifies up to 6 categories of waste. One of their methods was to use the support vector machine which serves as the experiment's baseline due its consistency and simpler implementation compared to its alternative methods. The second method was to use a convolutional neural network that was inspired by the architecture of "AlexNet" but at 3/4ths the scale in terms of layers. The SVM was able to achieve a superior score of 63% accuracy compared the 22% accuracy from the CNN. The students suspect that suboptimal hyperparameters and a lack of datasets stunted the potential of CNN's from getting a higher accuracy.

3. Dataset and Evaluation

Most of the experiments in this report used images and their labels provided by the "Garbage Classification" dataset (CCHANG, 2018). There are 2527 images from the dataset, containing 403 cardboard images, 501 glass images, 410 metal images, 594 paper images, 482 plastic images, and 137 trash images. The experiment required randomly split data such that 1769 are assigned to training set, 253 to dev set, and 505 to test set. Using a 70/10/20 data ratio. Each image will then be resized to 224 x 224 pixels.

We have chosen 3 key evaluation metrics: accuracy, latency/throughput, and size of model. Firstly, model accuracy is important because it determines how well the model can classify images and sort trash correctly. Next, the latency and throughput is important because it determines how quickly the model can be trained, and how quickly it makes predictions, which is essential when dealing with extremely large amounts of trash. Finally, the amount of storage required to store it, which is required to determine if additional are costs needed for memory solutions in the IoT devices responsible for trash classification. By measuring these metrics, we can observe the corresponding tradeoffs each model makes when considering them.

4. Methods

4.1. Method Introduction

In the midterm report, we considered only the number of hidden layers in the model, hence the comparison between the baseline of AlexNet and the lighter MobileNet. This time, we're considering the different machine breakthroughs that have occurred over the past decade in machine learning. This includes the deep CNN layers and residual connections with ResNet, shallow depthwise convolutional operations with MobileNetV3, and the "squeeze and excitation" 1x1 convolution approach with SqueezeNet.

Furthermore in each class of models, we're experimenting with the number of layers, for instance, ResNet18, ResNet50, and ResNet101 has 18, 50, and 101 layers respectively. MobileNetV3 small and large differ in the amount of layers and learn-able parameters it has with MobileNetV3L having more.The subsections below outlines our experiment in greater detail.

4.2. Step 1: Implementing Python Functions

First, we defined the training algorithm, implementing the Python definitions for accuracy, training_step, validation_step. The training and validation step utilizes cross entropy to calculate the difference between predicted and actual output. Binary cross-entropy is special case of cross-entropy which can be used in multi-class classification to determine if each label is either 0 or 1. In a neural network, this type of prediction can be achieved by sigmoid activation, which was in fact used at the end of all model layers in this experiment. Unlike the softmax returning a probability vector, sigmoid function helps return a ”one hot” vector showing a 1 to the predicted label.

4.3. Step 2: Choosing the Models

Next, we initialize well-known models from Pytorch. Here is the list of the models used: AlexNet, ResNet18, ResNet50, ResNet101, MobileNet_V3S, MobileNet_V3L, SqeezeNet_V1p1. These models were loaded onto a pytorch model object where they are all slightly altered to have the classification layer output 6 to function in the context of garbage classification. These models are pre-trained by the pytorch library, meaning they have been trained with a large dataset called ”ImageNet” to develop useful features and weights. The pretrained image classifiers can then be further fine-tuned to handle downstream tasks like classifying trash through the process of transfer learning.

4.4. Step 3: Fine-tuning

After selecting each model, we fine-tune the model with our own dataset. Since our dataset is much smaller compared to the dataset used to pre-train the models, fine-tuning works by optionally freezing the earlier feature layers of the CNNs while training the later classification layers to better fit our downstream trash classification job. Most models are further trained with the garbage dataset using the Pytorch Adam optimization algorithm for 20 epochs, with a learning rate of 0.00005. However, if learning rates are too high, learning rate schedules are used during training, which reduce the learning rate during epochs with worsening accuracy.

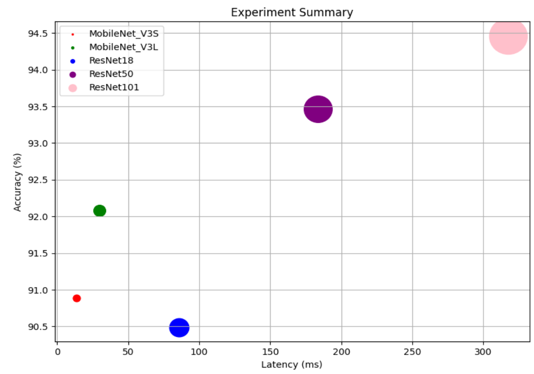
4.5. Step 4: Testing

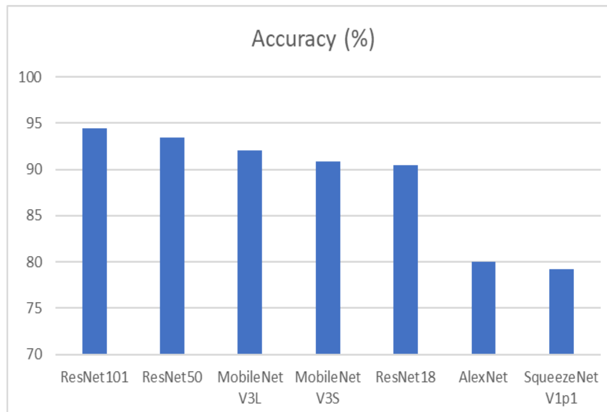
After fine-tuning it was time to test the model. As we mentioned before, we are evaluating based on accuracy, latency, and storage size. To capture accuracy we calculated the ratio of correct to total predictions, for latency we measured the average execution time after predicting through the entire test set, and for the storage size we analyzed the total storage space taken up by each model's pretrained weights stored in Google Colab.

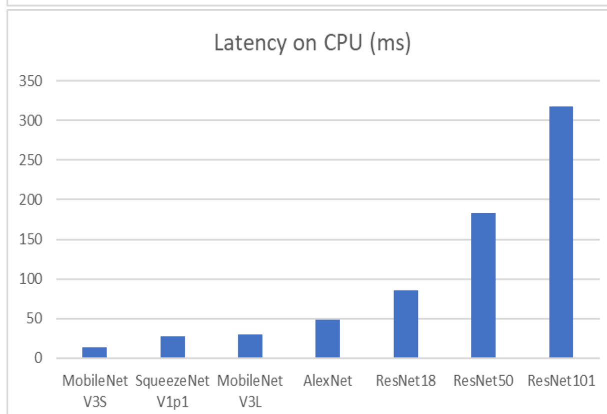
5. Experiments

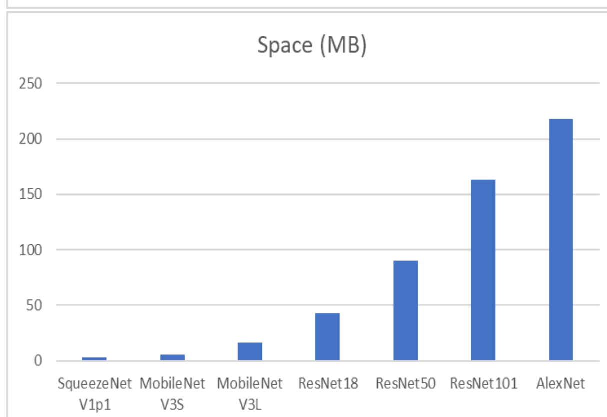
5.1. Accuracy vs Latency vs Space Complexity

The table below summarizes our test set results.







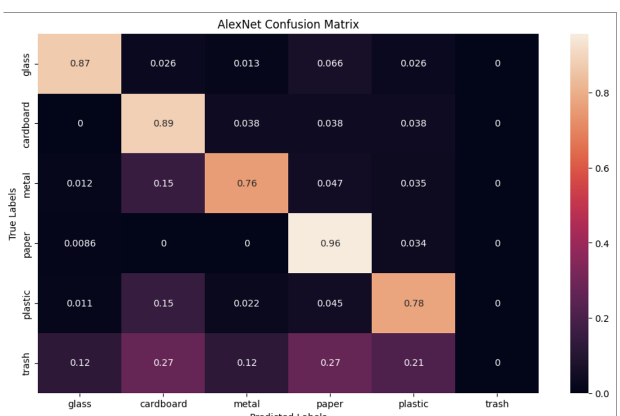


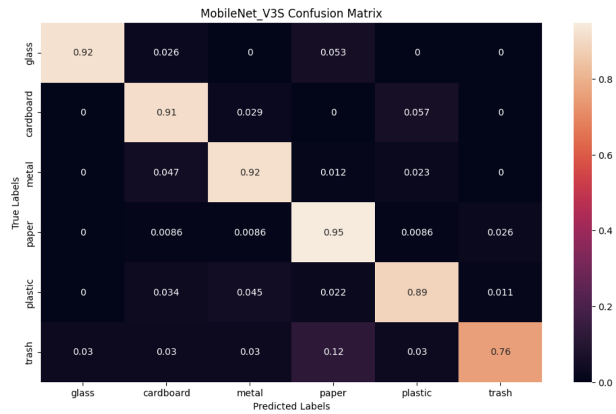
The results of our experiment as per each model are as follows:

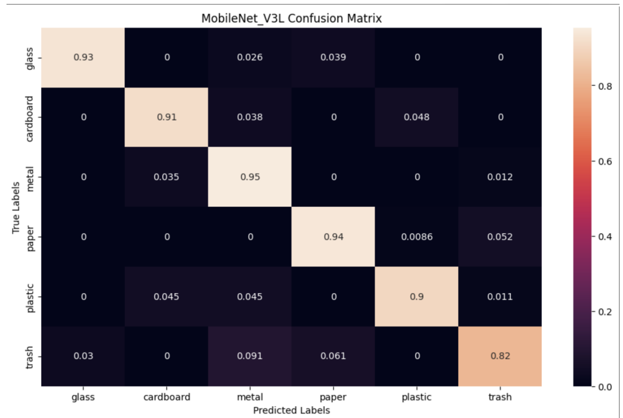
- AlexNet
  - Accuracy: 80%
  - CPU Latency: 48.6ms
  - Space: 217.6MB
- MobileNet_V3S
  - Accuracy: 90.89%
  - CPU Latency: 13.4ms
  - Space: 6.0MB
- MobileNet_V3L
  - Accuracy: 92.08%
  - CPU Latency: 29.8ms
  - Space: 16.3MB
- ResNet18
  - Accuracy: 90.49%
  - CPU Latency: 85.7ms
  - Space: 42.7MB
- ResNet50
  - Accuracy: 93.47%
  - CPU Latency: 183.6ms
  - Space: 90.1MB
- ResNet101
  - Accuracy: 94.46%
  - CPU Latency: 317.4ms
  - Space: 162.8MB
- SqueezeNet_V1p1
  - Accuracy: 79.21%
  - CPU Latency: 27.1ms
  - Space: 2.8MB

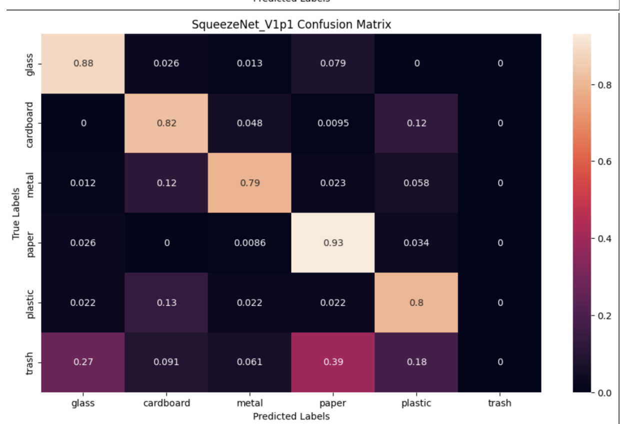
5.2. Confusion Matrices for all models

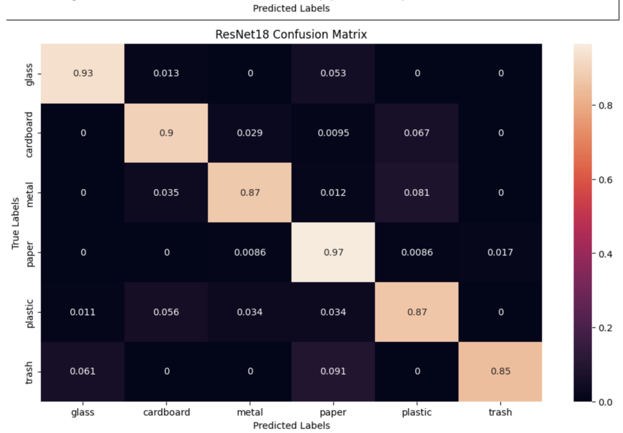
Below are the confusion matrices for each model.

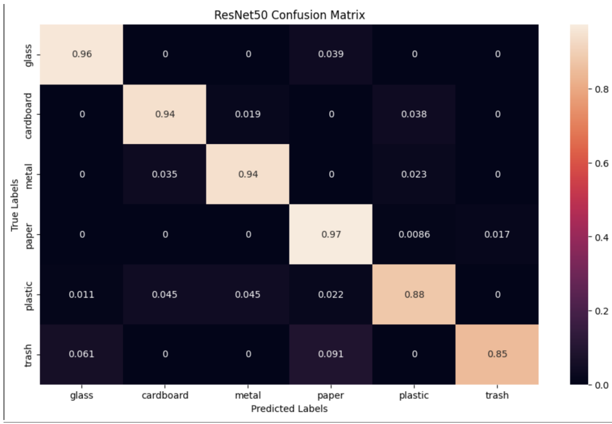


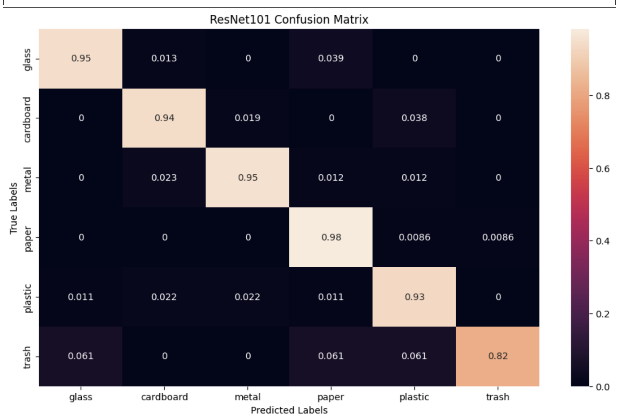












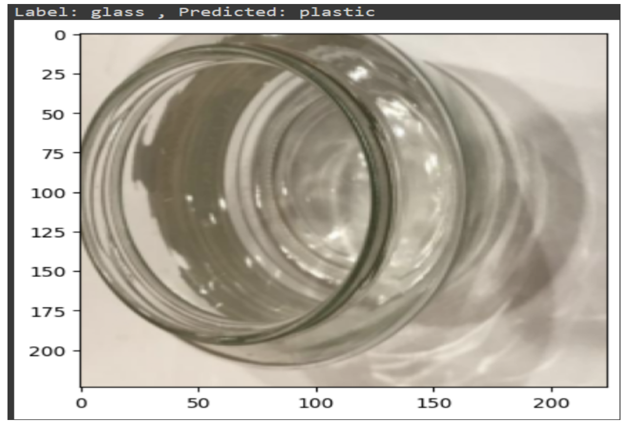
6. Discussion

6.1. Choosing a Model

Balancing all 3 evaluation metric, we chose the MobileNet_V3L as the superior machine learning model of choice for trash evaluation due to it being in the top 3 in all of the measured metrics. It had a significantly high accuracy of 92.08%, a very small latency of 29.8ms, and a space complexity of only 16.3MB, which is likely to fit in most IoT devices.

6.2. Manual Error Analysis

When performing manual error analysis, we realized that some errors made by the model can be attributed to some strong features which are present in more than one category. For example when comparing a piece of glass, the model might attribute the object's transparency to classify it as glass. However, if a piece of plastic is sufficiently clean and transparent, then the model might mistakenly classify that piece as glass as well. This is exactly what happened for our model. Other examples includes our model wrongly classifying pieces of cardboard as plastic because its surrounding sticker wrappers, and wrongly classifying pieces of cardboard as paper when the paper has been stained to a brownish color. It also appears that the models are struggling to classify the "trash" category, which is understandable since the garbage dataset had much less images of trash compared to other recyclable objects. Most of the models performed well, except for AlexNet and SqueezeNet1.1 which had problems with the trash category. These could be due to some issues in the layer regarding ReLU.



An image example where glass may be misrepresented as plastic

7. Conclusion

In our experiments, MobileNet_V3L was the clear winner for the model of choice in trash classification. It's dominance in space complexity is unsurprising, given that MobileNet was designed particularly for mobile and embedded devices. However, a more surprising result is it's high accuracy and low latency which even outperformed the heavier CNN architectures with orders of magnitudes more learnable parameters, which goes to show how the number of parameters yields decreasing returns to accuracy in a rate that we initially did not expect.

With recent advancements in GPT 4.0, this finding might be extended to other fields in machine learning like natural language processing, where we can explore if effective fine-tuning on a lightweight, general purpose model could potentially perform just as well as models with a trillion parameters.

References

CCHANG. Garbage classification, 2018. URL https://www.kaggle.com/ds/81794.

Yang, M. and Thung, G. Classification of trash for recyclability status. Technical report, Stanford University, 2016.