DEPLOYABLE WASTE CLASSIFICATION – PRACTICAL CONSIDERATIONS FOR CLASSIFICATION AND DEPLOYMENT WITH CNNS

NIKOLAJ HINDSBO*
MECHANICAL ENGINEERING
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA
NIKOLAJHINDSBO@GMAIL.COM

CHAD MERRILL*
MECHANICAL ENGINEERING
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA
CAMERRIL@ANDREW.CMU.EDU

NISHANTH MOHANKUMAR*
MECHANICAL ENGINEERING
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA
NMOHANKU@ANDREW.CMU.EDU

* EQUAL CONTRIBUTORS

ABSTRACT

This study explores the optimization of Convolutional Neural Networks (CNNs) for complex classification tasks, using the widely applied TrashNet dataset as a case study to address an engineering challenge in waste classification. Traditional approaches utilize standard CNN models like ResNet50, which, while effective, often face limitations when deployed on low-resource devices or when adapting to diverse and realistic datasets.

Our approach enhances the training and deployment of CNNs through a comprehensive methodology that includes data augmentation, model freezing strategies, and quantization. We implement a series of ablation studies to refine data transformation techniques and evaluate the impact of varying dataset sizes and model architectures, specifically comparing the performance of ResNet50 and EfficientNet B0 and B4 variants.

By fine-tuning these models on the TrashNet dataset and optimizing them for efficient operation on constrained microprocessor devices (e.g. Raspberry Pi), we achieved notable improvements in model accuracy and computational efficiency. Our results underscore the versatility of CNNs when appropriately adapted through advanced machine learning strategies.

Developing strategies for machine learning are ever-changing and often certain steps are overlooked or confusing. The contribution of our research is twofold: it not only guides the practical deployment of CNNs in real-world applications but also provides a framework for adapting these models to a variety of engineering challenges beyond waste classification. This work works to serve as a helping hand to gap the broader application of machine learning in solving diverse and practical problems, offering insights into the potential of CNNs in various industrial contexts for Engineers looking to polish practical deployment.

I. INTRODUCTION

The United States Environmental Protection Agency (EPA) defines Municipal Solid Waste (MSW) as a variety of items consumers discard after use, including bottles, corrugated boxes, food, yard trimmings, sofas, computers, tires, and refrigerators [1]. Notably, this definition does not encompass all materials that may be landfilled, such as construction and demolition (C&D) debris, municipal wastewater sludge, and certain non-hazardous industrial wastes, as they are handled through different waste streams. In the EPA's most recent study from 2018, it is reported that of the total 292.4 million short tons of MSW generated in the US in 2018, approximately 32.1% was recycled or composted, with an additional 11.8% combusted for energy recovery. These figures highlight a pressing issue within US waste management practices, as the EPA also clarifies that the other 50% of MSW was indeed landfilled, underlining the need for improved strategies to address this challenge.

Our research aims to address this challenge by leveraging two strategic approaches: enhancing the effectiveness of machine learning models and optimizing their deployment on low-resource devices. Building on existing research, we further contribute by conducting several ablation studies to identify optimal performance parameters. Moreover, we focus on optimizing our model for deployment on low-fidelity devices through techniques such as quantization, which reduces the model's memory footprint and computational requirements without significantly sacrificing accuracy. This enables efficient model operation even on computationally constrained platforms, making guided recycling technology more

accessible and implementable in diverse settings worldwide. Through these efforts, we strive to bridge a knowledge gap between cutting-edge machine learning techniques for practical, scalable solutions with an environmental sustainability solution in mind.

II. Related Work

A recent literature review on waste identification technologies reveals that Convolutional Neural Networks (CNNs) are predominantly used as the foundational architecture in the development of classification systems [2]. Among the various pre-trained CNN models, ResNet50 emerges as the most frequent choice because of its wide success in opening up deep learning from the use of skip connections to solve the vanishing gradient problem. This review highlights the Trashnet dataset as the most utilized dataset for fine-tuning these pre-trained CNNs. The Trashnet dataset is publicly available and encompasses six categories of waste: plastic, cardboard, paper, metal, glass, and trash. Furthermore, the review indicates that image-based classifiers, particularly those deployed on edge computing devices such as the Raspberry Pi, represent the most prevalent deployment approach for waste classification systems.

Transfer learning has emerged as a pivotal technique in the realm of CNN-based waste classification, as evidenced by recent studies that have explored its efficacy [3][4]. Notably, the comparative analysis of various pre-trained models including AlexNet, VGG-16, GoogleNet, and ResNet, underscored the versatility of transfer learning in adapting these architectures for the specific task of identifying recyclable materials using the TrashNet dataset with 6 classes. A critical component of this adaptation involves the introduction of new fully connected layers tailored to the classification task at hand, a method that was consistently applied across different studies. This process not only leverages the robust feature extraction capabilities developed through pre-training on extensive datasets like ImageNet but also allows for fine-tuning the models to enhance their precision for waste classification on chosen classes.

In addition to exploring the use of transfer learning from newer pre-trained architectures such as ResNeXt-50 32x4d, one study also explored the performance of different optimizers. Ultimately, this work determined that the Adam optimizer was the most suitable for their waste classification task, achieving a 98.02% accuracy on the TrashNet dataset with 6 classes [5][2]. Furthermore, this work also briefly explored the use of data augmentation to overcome the limitations of small training datasets like TrashNet. Utilizing just a small sample of augmentations to increase the size of the dataset, the results showed a modest increase of 3% in the accuracy of the model, suggesting that further investigation into data augmentation in the field of waste classification would prove valuable.

Building upon the insights gained from the related work, our study adopts several established strategies to enhance the effectiveness and applicability of our waste classification system. We select ResNet50 as our primary pre-trained model due to its frequent utilization and proven performance in similar tasks. For fine-tuning, we employ the TrashNet dataset, which is well-suited for our goal with its six distinct waste categories. In line with the findings on optimization techniques, we choose the Adam optimizer to train our model, capitalizing on its ability to achieve high accuracy rates effectively. Consistent with the prevalent trend of deploying image-based classifiers on edge devices, we plan to implement our system on a Raspberry Pi, aiming to leverage its accessibility and suitability for real-world applications in waste management.

Building on these foundational choices, our research explores the implications of freezing/unfreezing the underlying model, the impact of different data transformations during training, the effects of varying dataset sizes, and the potential benefits of employing different model architectures and quantization techniques for efficiency and accuracy.

III. Methods

A. Dataset

We utilize the TrashNet dataset, specifically curated for waste classification tasks, which consists of 2,527 hand-collected examples from diverse locations around Stanford University [6]. Reflecting the real-world condition of waste materials, the dataset features items that are worn, damaged, or partially degraded. To ensure the dataset captures the variability encountered in waste presentation, images include modifications such as random rotations, brightness adjustments, translations, scaling, and shearing. The dataset is divided into six categories: Trash, Plastic, Paper, Metal, Glass, and Cardboard, with a non-equal distribution across these classes. Table 1 provides a detailed breakdown of the dataset, and Figure 1 showcases representative examples from each category.

Label	Examples in Dataset	Training (70%)	Validation (15%)	Testing (15%)
Glass	501	350	75	75
Paper	594	415	89	89
Cardboard	403	282	60	60
Plastic	482	337	72	72
Metal	410	287	61	61
Trash	137	95	20	20
Total	2527	1768	379	379

Table 1: Breakdown of the Trashnet dataset and our training/validation/testing split using segmentation and stratification techniques.

Figure 1: Two example images from the Trashnet dataset. Each image is captured on a white background with altering lighting and damage to the product as shown [6].

B. Architectures

We leverage a pre-trained ResNet50 model from torchvision's collection of pretrained models due to its widespread use and proven effectiveness in waste classification tasks [2].

Using the convolution layers as our pre-trained base, we define two fully connected layers for our specific classification needs. The first layer takes the 2048 outputs from the ResNet50's final convolutional layer and reduces them to 512 nodes, utilizing a ReLU activation function for non-linear processing. The subsequent layer further processes these 512 inputs down to six to align with our classification scheme. We utilize torch's CrossEntropyLoss as the loss function for training, which inherently applies a Softmax function to the output, making an explicit Softmax layer unnecessary at the model's output for classification. This model serves as a comparable baseline to many of the other waste classification tasks in the field.

Since the release of Resnet-50, many organizations have spent considerate efforts to improve accuracy on the ImageNet dataset. In 2019, Google AI showed that balancing model depth, width, and resolution can lead to better performance with a series of models called "EfficientNet". EfficientNet represents current state-of-the art solutions to ImageNet, achieving a state-of-the-art accuracy of 84.3% top-1 accuracy for ImageNet while being 8.4x smaller and 6.1x faster on inference than the previous best existing ConvNet [7].

Therefore, we leverage EfficientNet as a high-end benchmark in our study. Particularly, we use the B0 and B4 variants of the EfficientNet models which have been shown to have better accuracy and computational efficiency in image classification tasks on datasets like ImageNet. The selection of EfficientNet-B0 was motivated by its minimal parameter count and enhanced speed while maintaining the same top-1 and top-5 accuracy compared to ResNet50. Meanwhile, EfficientNet-B4 was chosen for its parameter size, offering a closer comparison to ResNet-50, thus facilitating a more nuanced evaluation of model efficiency and performance. Aligned with our approach with ResNet50, we adapt the EfficientNet models with two fully connected layers, deviating only to account for the 1280 node output of EfficientNet's final convolutional layer rather than the 2048 for ResNet50. Likewise, we utilize torch's CrossEntropyLoss as the loss function for training, ensuring a uniform methodology in loss calculation and the application of Softmax across our architectural configurations.

C. Pre-Trained Weight Freezing/Unfreezing

When using pre-trained architectures, the question of freezing or leaving the convolutional layers unfrozen often comes up. We conduct an initial investigation of the impact of freezing versus unfreezing the underlying pretrained weights during fine-tuning for the ResNet50 model on the Trashnet dataset for our study.

Theory suggests that a large and diverse pre-trained models (such as ImageNet) are good at capturing basic image characteristics like edges, textures, and shapes. However, they may not be optimal for more specific or advanced features that are unique to the new dataset [8].

On a smaller dataset such as Trashnet, allowing backpropagation through all layers lets the model adjust even the initial, more generic features to better suit the specific dataset. This can lead to a better fit and higher accuracy on the training and validation set as the model can refine these features to better capture the nuances of the new dataset. However, this approach risks overfitting the training data, especially since the new dataset could not be large and diverse enough to represent the broader task domain [8].

We discuss our specific findings for this in the results section.

D. Data Augmentation Ablation Study

We investigate the impact of data augmentation on the robustness and generalization capabilities by conducting an ablation study with a range of image transformation techniques from the torchvision module. We perform this study using the unfrozen version of our ResNet50.

The chosen transforms included Sharpness Adjust, Color Jitter, Horizontal Flip, Random Rotation and Random Crop, which are widely used and can often have the highest impact on improving accuracy [9].

E. Dataset Size Ablation Study

We investigate the optimal number of images required to fine-tune a pre-trained model for a new task by conducting an ablation study analyzing model performance relative to the volume of training data employed. This is extremely important when considering the difference in architecture. A larger model often requires more images to minimize the risk of overfitting [10].

We allocate 80% of the dataset for training and the remaining 20% for validation. During the ablation study, we systematically vary the size of the training dataset, utilizing 25%, 50%, 75%, and 100% of the training data to observe the impact on model performance.

F. Quantized Architectures

We evaluate quantized versions of the three proposed architectures to further optimize our model for deployment on resource-constrained devices.

After fine-tuning was performed on each model, the best model weights were loaded and the quantization was performed with torch.quantization. The quantization was dynamically performed, and the layer's parameters were compressed to qint8. Compared to PyTorch's default float32, qint8 uses a quarter of the memory size, freeing up the number of operations that can be loaded into a processor at one time by 4x. In practice, a shift from float32 to qint8 has been shown to achieve 3x faster inference in forward propagation for large models [11].

IV. Results and Discussion

A. Weight Freezing Ablation Study

Our findings show a significant improvement in validation accuracy for the ResNet50 model when the entire model's weights are unfrozen and allowed to adjust during training (73% versus 89%). It's important to note that our findings are based on cross-validation conducted over 20 epochs with basic hyperparameters.

Based on the better performance of the unfrozen model, we chose to conduct utilize both frozen and unfrozen versions for the final model training. Although Trashnet is relatively small compared to ImageNet, the improvement in accuracy was much higher than expected from literature [8], and with no diverse dataset to compare against, we cannot confidently the degree to which overfitting either helps or hinders the final model in deployment. We attribute this enhancement to the model's ability to overwrite pre-learned, irrelevant features from the original training (e.g., features unique to dogs and other animals in the ImageNet dataset) with new, task-specific information. An important class that was often misunderstood with weight freezing was ImageNet's "water bottle" classification which we found often resulted in water bottle shaped items (such as tin cans or metal water bottles), despite their material composition, being misclassified as plastic.

B. Data Augmentation Ablation Study Results

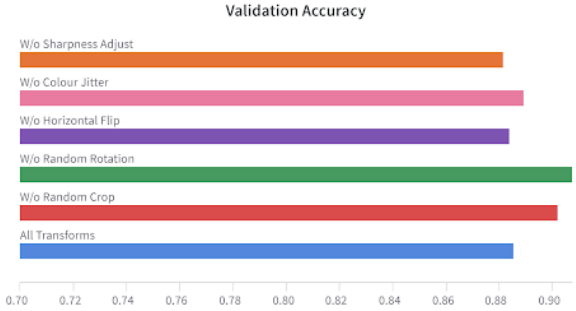

Figure 2: Data augmentation ablation study validation accuracy results.

Validation accuracy for each configuration of our ablation study is detailed in Figure 2, which outlines the results of our ablation study for these transformations.

Our findings suggest that excluding Random Rotation and Crop transformations yields improved model performance. Therefore, we train our final models with only Sharpness Adjustment, Random Horizontal Flip, and Color Jitter. Given that the TrashNet dataset primarily consists of centered images against uniform backgrounds, these transformations tend to zoom in on non-distinctive features such as the background, which do not contribute to class identification. Conversely, Sharpness Adjustment improves generalizability to blurry images, and Random Horizontal Flip aids in recognizing objects that have been rotated 180 degrees. Although Color Jitter offers minimal benefits, its inclusion may still be warranted, depending on its prevalence in the deployment environment.

C. Dataset Size Study

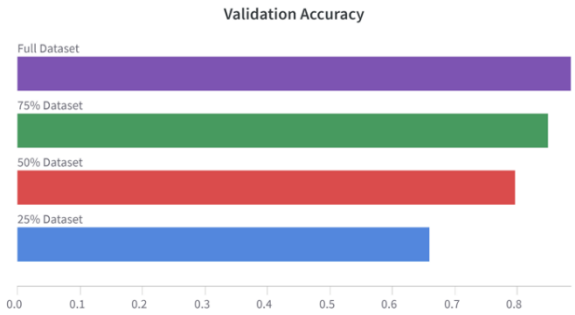

Figure 3: Dataset size ablation study validation accuracy results.

Our findings on dataset size indicate that model performance improves logarithmically as more data is added, with diminishing returns observed between the 75% and 100% data increments. This trend aligns with findings reported by Dawson et al. [10]. Regarding the behavior of convolutional architectures, these results suggest that fine-tuning a pre-trained model can be effectively achieved with a relatively small number of samples for the new underlying task.

Given the relatively small number of samples in the TrashNet dataset, we choose to use 100% of the dataset when training our final models as the difference in training time is negligible for our case. However, our results support that using a small sample size is valid for fine-tuning if the initial dataset were much larger than the size of the TrashNet dataset.

D. Final Trained Model Results

The final models incorporated insights from ablation studies on data augmentation, dataset size, and weight freezing. The transformation techniques used were resizing, random horizontal flipping, color jittering, sharpness adjustment, and normalization. We used the entire dataset, trained models in both frozen and unfrozen states, and additionally explored the impact of quantization on unfrozen models.

For computational efficiency analysis, we conducted a grid search training each model four times using consistent batch sizes and epoch counts (32 and 20, respectively). This approach ensured that comparisons of compute times, captured in Table 4, accurately reflected each model's architectural efficiency of floating point operations (FLOPS) during backpropagation and validation inference.

We further tuned hyperparameters by establishing a grid for learning rates, batch sizes, epochs, and learning rate scheduler step sizes. Initial testing guided adjustments primarily in learning rates and scheduler steps, maintaining consistent batch sizes due to hardware limitations. This tuning aimed at optimizing testing accuracies and ensuring fairness in comparisons, given the sensitivity of larger models to hyperparameters.

The models exhibited diverse responses to learning rates and step sizes. Typically, more complex backpropagation benefited from lower step sizes and higher initial learning rates. The employed learning rates were below 0.005, using a simple step scheduler that halved the rate every 4 to 8 epochs.

	Parameters	*Compute Time (Training 4 models, same # epochs/batch size)*	*Test Accuracy (on Trash Dataset)*
	(millions)	***(TFLOP-hours)***	***(%)***
Resnet50 (unfrozen)	25.6	4.26	96.58
Resnet50 (frozen)	25.6	0.922	83.41
Resnet50 (unfrozen, quantized)	25.6	4.36	96.54
EfficientNet-B0 (unfrozen)	**5.3**	**0.925**	**96.10**
EfficientNet-B0 (frozen)	5.3	0.799	83.82
EfficientNet-B0 (unfrozen, quantized)	**5.3**	**0.935**	**95.76**
EfficientNet-B4 (unfrozen)	19	38.32	94.68
EfficientNet-B4 (frozen)	19	4.02	**91.03**
EfficientNet-B4 (unfrozen, quantized)	19	38.42	94.68

Table 2: Final models compared on parameters, compute time, and test accuracy.

Table 2 summarizes the most important differences between ResNet50, EfficientNet-B0, and EfficientNet-B4 in training, comparing parameters, compute time, and testing accuracy on Trashnet. Our suggestion for a dataset like Trashnet is that if weight freezing is desirable, a larger model such as EfficientNet-B4 should be used for greater generalizability and accuracy compared to the other architectures. However, if the convolutional layers are unfrozen during fine-tuning, an efficient and smaller architecture such as EfficientNet-B0 showed great performance in both final testing accuracy and compute time to train. This decision should be based on compute availability, dataset size and heterogeneity, and how important your accuracy metric is compared to cost to implement.

The results of Table 2 show several interesting and conclusive results. First and foremost, the selection of architecture alone showed great significance. While ResNet50 and EfficientNet-B0 differed in parameters with a factor of 5, EfficientNet-B0 and ResNet50 achieved almost the same testing accuracies on Trashnet across the board.

Furthermore, EfficientNet-B4 which had approximately 20% less parameters than ResNet50 and similar FLOPS was able to improve testing accuracy almost 8% when frozen. In unfrozen testing, there was actually a decrease in testing accuracy. This is backed up by theory that large models are often perform better when frozen. Because they train on a much larger dataset, have great generalizability, and are extremely sensitive to training transformations, hyperparameters, loss functions, optimizers, and learning rate schedulers (CITE: Miao et al.) . We also conclude from our results that a large model such as EfficientNet-B4 would be better left unfrozen on a dataset like Trashnet for compute, generalizability, and testing set accuracy considerations.

Importantly, the difference in compute time is worth noting. As the number of parameters increased, the compute time increased as well. However, the scaling of compute time was not linear with number of parameters. While the frozen architectures all performed equally well in terms of compute time, there was a massive difference in the unfrozen models.

Lastly, the results justify quantization. There was nearly no difference in testing accuracy from Meanwhile, inference time stands to greatly improve by up to three times from q32 to q8 from the decrease in FLOPS [11]. This is especially important if considering deployment to a low-fidelity device such as a Raspberry Pi which can save money.

In conclusion, the findings from the ablation study, the selection of model architectures, and the meticulous hyperparameter tuning have demonstrated their substantial potential to enhance computational efficiency, improve testing accuracies, and increase generalizability to unseen data. These elements also significantly optimize inference times among other benefits in classification tasks. It is our aspiration that this document serves as a valuable resource for both students and professionals embarking on new projects, providing them with a deeper understanding and practical strategies to refine classification models tailored to their specific requirements.

D. BRIEF DISCUSSION

Our study using three different models—ResNet50, EfficientNet-B0, and EfficientNet-B4 explored various configurations including unfrozen, frozen, and quantized states to gauge their performance on the Trashnet dataset. Our findings suggest that not freezing the weights is preferable for applications such as Trashnet where the training conditions (specific table background and lighting) do not mimic real-world scenarios, like those in cafeterias or restaurant trash bins. The pre-trained models on ImageNet provide a robust foundation of generalized features beneficial for broad image recognition tasks. However, allowing complete weight adjustability might lead the model to overfit to the nuances and noise of the training dataset, thereby reducing its ability to generalize to different or unseen data.

This risk of over-specialization underscores the importance of selecting the right architecture that can maintain flexibility across diverse conditions without losing the ability to recognize core elements across varying backgrounds and types of trash. Although the models achieved high accuracy, with unfrozen weights closely matching the top performance metrics (only two misclassifications put us short of the 98% metric as an example), the results indicate that there is room for further exploration and discussion beyond accuracies. Specifically, further comparative studies on unseen, disparate data would illuminate which configurations best balance specificity with generalizability.

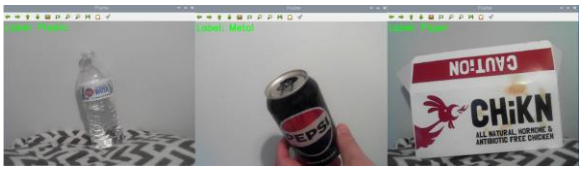

Figure 4: Raspberry Pi-5 camera live-time inference results.

Moreover, if time had permitted, it would have been beneficial to conduct additional

experiments using a Raspberry Pi to evaluate the inference times and adaptability of these architectures against varied backgrounds and trash types. A secondary, more diverse dataset could also enhance the robustness of our findings by providing a more challenging test environment. Such investigations could potentially validate or refine our current understanding, suggesting new paths for achieving even higher accuracy and efficiency in practical applications.

GitHub Repository

[1] "National Overview: Facts and Figures on Materials, Wastes and Recycling." EPA, Environmental Protection Agency, 22 Nov. 2023, www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/national-overview-facts-and-figures-materials.

[2] Arbeláez-Estrada, Juan C. et al. "A Systematic Literature Review of Waste Identification in Automatic Separation Systems." Recycling (2023): n. Pag.

[3] Ozkaya, Umut and Levent Seyfi. "Fine-Tuning Models Comparisons on Garbage Classification for Recyclability." ArXiv abs/1908.04393 (2019): n. Pag.

[4] Gyawali, Dipesh et al. "Comparative Analysis of Multiple Deep CNN Models for Waste Classification." *ArXiv* abs/2004.02168 (2020): n. Pag.

[5] Singh, S., et al. "Evaluation of Transfer Learning based Deep Learning Architectures for Waste Classification." *2021* IEEE Xplore, doi:10.1109/ISAECT53699.2021.9668454.

[6] Thung, Gary and Mingxiang Yang. "Classification of Trash for Recyclability Status." (2016).

[7] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." In Proceedings of the 36th International Conference on Machine Learning (ICML 2019), Long Beach, CA, USA, 2019.

[8] Miao, Zhengqing, and Meirong Zhao. "Weight Freezing: A Regularization Approach for Fully Connected Layers with an Application in EEG Classification." arXiv preprint arXiv:2306.05775 (2023).

[9] Osborne, Jason. (2002). Notes on the Use of Data Transformations. Practical Assessment, Research & Evaluation. 8.

[10] Harriet L. Dawson, Olivier Dubrule, Cédric M. John,Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification,Computers & Geosciences,Volume 171,2023.

[11] Grebennikov, Roman. "How to Compute LLM Embeddings 3x Faster with Model Quantization." *Medium*, Nixiesearch, 13 Nov. 2023. https://medium.com/nixiesearch/how-to-compute-llm-embeddings-3x-faster-with-model-quantization-25523d9b4ce5