RECYCLITRON: USING TRANSFER LEARNING AND CONTRASTIVE METHODS TO CREATE A RECYCLABLE WASTE CLASSIFIER

Aidan Banerjee **Max He** **Stephen Charkoudian**

Sharing permission statement: *Instructors can share this report to future students of similar classes*

1 INTRODUCTION

1.1 PREDICTION PROBLEM

We are developing a recyclable materials classifier designed to categorize 224x224x3 RGB images of various household waste items into six classes: cardboard, glass, paper, metal, plastic, and trash. The first five classes represent recyclable materials, while the last class, trash, represents non-recyclable waste. This classifier aims to assist in the automated sorting of household waste, streamlining recycling processes and reducing contamination in recycling streams.

1.2 LEARNING PROBLEM

The learning problem involves training machine learning models to classify waste into recyclable and non-recyclable categories with limited labeled data. While general labeled datasets scraped from web images are available for pre-training, they may not reflect the real-world variability of garbage and recycling images encountered in waste management facilities. We leverage models pre-trained on large data-sets either containing general recycling images with 30 classifications, recycling/garbage data-sets with 2 classifications, or utilizing Imagenet. Then we fine-tune the models on a smaller, specific datasets with the 6 classes mentioned above to optimize classification performance and model real-world training scenarios.

1.3 OUR GOAL

Our primary goal is to maximize classification accuracy, specifically, balanced accuracy. In the United States, recycling guidelines vary widely by municipality; although general guidelines exist, specific recycling capabilities differ across locations. Waste management facilities, therefore, require tailored models to classify the materials they can recycle, but often lack access to large, labeled datasets specific to their needs. Our project seeks to demonstrate the effectiveness of a model pre-trained on general datasets relevant to recyclable materials, which is then fine-tuned and tested on a target dataset representing the materials a single waste management facility can process. By providing a scalable, adaptable solution for enhancing recycling efficiency, this approach could enable practitioners to implement cost-effective classification systems that are customized to their facilities' capabilities. As a result, this project helps reduce contamination in the recycling stream, improves resource recovery rates, and supports the broader goal of sustainable waste management.

1.4 METHODS (OVERVIEW)

We are exploring three main methods to build an effective recyclable materials classifier. The first method involves using transfer learning, utilizing a pre-trained ConvNeXt model. Liu et al. (2022) After either using the given imagenet weights, or pretraining on one of two related tasks, we fine-tune a new classifier head on our target dataset. The fine-tuning consists of two stages: linear probing, where the feature extractor is frozen and only the classifier head is trained, and then fine tuning where we gradually unfreeze layers to adapt pretrained features to the the target task.

Our second method leverages SimCLR Silva (2021), a self-supervised contrastive learning framework. SimCLR learns representations by maximizing agreement between differently augmented views of the same image while pushing apart representations of different images, without requiring any label information. The network is trained to recognize any augmentations of the same image as positive pairs while treating all other images in the batch as negative examples. After pretraining a model using SimCLR on one of our datasets, we fine-tune it on our target task to evaluate how well the learned representations transfer to our target domain.

Our third method employs Supervised Contrastive Learning (SupCon), which combines SimCLR's contrastive learning method with a supervised component to optimize class separability. Khosla et al. (2020) SupCon uses all instances of the same class as positives, ensuring that similar classes, such as recyclable materials, are pushed closer together in the embedding space, while non-recyclables remain distinct. This approach allows for robust class clustering, particularly valuable for datasets with inherent class hierarchy. However, its reliance on large batch sizes to generate sufficient positive and negative pairs per instance presents a significant computational and memory cost, which may limit training efficiency on hardware with restricted resources.

1.5 Hypothesis

We hypothesize that our third method, SupCon, will outperform the other methods in classification accuracy for our recyclable materials dataset. This is because SupCon most directly leverages SimCLR's contrastive structure, which clusters similar classes closer together in the embedding space while using label information to maintain a clear distinction between recyclable materials and organic waste.

2 Methods

2.1 Transfer Learning

Our first method involves transfer learning, where we utilize ConvNeXt-Tiny architecture Liu et al. (2022), a state-of-the-art convolutional neural network designed to match the performance of vision transformers while maintaining the simplicity and efficiency of standard ConvNets. The architecture is constructed entirely from traditional convolutional operations, yet achieves 87.8% top-1 accuracy on ImageNet-1K classification and outperforms Swin Transformers on COCO detection tasks. This high performance and pure convolutional design make it an ideal backbone for transfer learning in our waste classification task.

For our experiments, we explore three possible sources for weights: ImageNet-1K pretrained weights imported directly from PyTorch, and two models we trained ourselves - one on TrashNet, a binary dataset divided between organic waste and recyclable materials, and another on a household waste classification dataset with 30 classes. The use of ImageNet weights allows us to leverage general visual features learned from a diverse dataset, while our domain-specific pre-training aims to capture features more relevant to waste classification.

For the two source models we trained ourselves, we conducted a grid search across learning rates (10^{-4}, 10^{-3}, 10^{-2}), L2 regularization penalties (0, 10^{-3}, 10^{-2}), 3 random seeds, with a fixed batch size of 64. Each source model was trained using the Adam optimizer for 70 epochs with early stopping (patience of 15 epochs), with a 70/30 train/val split.

We implemented progressive fine-tuning on our target dataset using PyTorch. Starting with linear probing (training only the classifier head), we then gradually unfroze the network in three stages: first the last 25% of layers, then 50%, and finally the entire network. For each stage, we decreased the learning rate. At the final stage, where the entire network was unfrozen, we performed a grid search over learning rates (10^{-4}, 10^{-3}, 10^{-2}), L2 regularization penalties (0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}), and batch sizes (64, 128), using the Adam optimizer with early stopping (patience of 15) for a maximum of 100 epochs.

Our transfer learning pipeline was implemented from scratch using PyTorch's standard modules. We used torchvision for the ConvNeXt-Tiny architecture and ImageNet pretrained weights, but built our own training infrastructure including progressive fine-tuning stages, early stopping, and hyperparameter search. The pipeline uses PyTorch's DataLoader for efficient data loading, and standard modules like nn.Linear and nn.Dropout for the classifier head. Data augmentation leveraged torchvision's aviailable transformation functions.

2.2 SimCLR

Our second method employs SimCLR, a contrastive learning framework where an image encoder learns representations by maximizing agreement between differently augmented views of the same image while pushing apart representations of different images. For each image in a batch, we create two augmented views using a consistent set of transformations (random cropping, random horizontal flipping, color jittering, and random grayscale conversion), resulting in twice as many samples as the original batch size. The similarity between two views i and j is measured using cosine similarity between their encoded representations $\mathbf{z}_i$ and $\mathbf{z}_j$. $\mathbb{1}_{[k\neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if and only if $k \neq i$ and τ The contrastive loss for a positive pair of examples (i, j) is defined as:

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \tag{1}$$

We conducted a grid search across learning rates (10^{-3}, 10^{-2}, 10^{-1}), temperature values (0.01, 0.07, 0.1), and two batch sizes (128, 256) which were effectively increased to 1024 and 2048 through gradient accumulation due to hardware constraints. Each configuration was tested across three random seeds using the ConvNeXt-Tiny architecture with a 128-dimensional projection head. Training ran for a maximum of 350 epochs with early stopping (patience of 15).

To evaluate the progress of the SimCLR training, we used a linear evaluation protocol at every epoch, where we trained a new classifier head while the learned features were frozen. The head was optimized using Adam for a maximum of 100 epochs with early stopping (patience of 15)

After pretraining reached completion (either through early stopping or maximum epochs), we performed full fine-tuning of the entire model. For this stage, we conducted a grid search over learning rates (10^{-4}, 10^{-3}, 10^{-2}), L2 regularization penalties (0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}), and batch sizes (64, 128). Each configuration was trained for a maximum of 100 epochs using the Adam optimizer with early stopping (patience of 15), maintaining the same random seed used during pretraining to ensure consistent evaluation.

Our implementation utilizes PyTorch's standard modules and training utilities. The training pipeline was built from scratch, using PyTorch's DataLoader for efficient data loading, nn.Linear and nn.LayerNorm for network components, and torchvision's transforms for data augmentation. For optimization, we used PyTorch's implementation of the Adam optimizer. The contrastive loss function was taken from a public respository Tian (2020) associated with a related paper. After pretraining, we used the same codebase as our transfer learning pipeline for fine-tuning, leveraging early stopping and learning rate scheduling through PyTorch's standard utilities.

2.3 Supervised Contrastive Learning

Our third method, Supervised Contrastive Learning (SupCon), extends SimCLR by leveraging label information to identify positive pairs. Rather than only considering differently augmented views of the same image as positives, SupCon treats all samples from the same class as positive pairs. For each image in a batch of size N, we still create two augmented views, but the loss function is modified to pull together all samples of the same class:

$$L_{out}^{sup} = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a\in A(i)} \exp(z_i \cdot z_a/\tau)} \tag{2}$$

where I represents all samples in the augmented batch, $P(i)$ is the set of all samples in the batch with the same class as anchor i (excluding i itself), $A(i)$ is the set of all samples in the batch except i, and τ is the temperature parameter. The embeddings z_i and z_p are outputs from the projection network. We maintain the same hyperparameter search space as SimCLR, and the quality of the learned representations is evaluated using the same linear evaluation protocol as SimCLR. Following pretraining, we employed the identical fine-tuning protocol as SimCLR, conducting a grid search over the same hyperparameter space while maintaining pretraining seeds. In the same vain, the implementation was done using the same files as our SimCLR implementation, using the same off the shelf packages from section 2.2, just using a different loss function available in Tian (2020)

3 Data and Experimental Plan

3.1 Datasets

Our project uses three datasets chosen to align with different stages of model training and testing, helping the model distinguish recyclable materials from non-recyclable waste. The pre-training datasets, Recyclable and Household Waste Classification and TrashNet, contain 15,000 and 22,500 images, respectively, with image dimensions of 256×256×3 for the former and 224×224×3 for the latter King (2024), Sekar (2019). The Recyclable and Household Waste Classification dataset features 30 distinct categories of waste, while TrashNet is organized into two broader categories: recyclable and organic waste. However, many images in these datasets appear to be either marketing photos or edited/cartoon-like images, making them suitable for pre-training as a related but less precise task. For our first method, transfer learning, both pre-training datasets were used to train separate models, leveraging the diverse information they provide. For methods two and three, SupCon and SimCLR with supervised fine-tuning, only TrashNet was used.

For fine-tuning and testing, we use the Garbage Classification dataset, which contains 2,467 real, unedited photos of common recyclable materials like cardboard, metal, glass, paper, plastic, and trash cchangcs (2018). The dataset has 6 classes in total, with size of 512x384x3 images. During training, we resize images down to 224x224x3 to ensure consistency. This Garbage Classification dataset aligns closely with our goal, enabling the model to refine its understanding on realistic examples and evaluate its performance on actual waste types.

3.2 Performance metric

We will use Balanced Accuracy as our primary performance metric since our test set is slightly unbalanced, especially for the trash class. This metric ensures each class is evaluated equally, focusing on true positive rates across classes to fairly assess performance regardless of class frequency. Additionally, we examine standard Accuracy to assess overall model performance, as well as macro-averaged Precision, Recall, and F1 scores to provide a comprehensive view of per-class performance that accounts for both false positives and false negatives. We also analyze confusion matrices to identify specific areas where the model may struggle, giving insights into class-specific misclassifications.

3.3 Experiment Plan

For transfer learning model development, we first pre-train our source model on either the TrashNet or Recyclable and Household Waste Classification datasets using a 70/30 train-validation split (our other methods, SimCLR and SupCon, will only use TrashNet for pre-training). These splits were not predefined but were constructed using stratified sampling with a fixed random seed to ensure reproducibility and fair comparison across methods.

For our target task evaluation, we employ a 20/20/60 split for train/validation/test respectively, on the Garbage Classification dataset, maintaining class proportions. This results in training and validation sets of 493 samples each, and a test set of 1,481 samples. Our dataset exhibits class imbalance, with most classes containing approximately 400 examples, while the 'trash' class has 127 samples. We maintain these proportions across splits to ensure our validation performance estimates reliably predict test performance. In order to combat this, we doubled the size of the trash in our test data set using random flip, crop, rotation, and color jitter. To verify data independence, we developed a Python program that calculates image similarity across datasets using a ResNet for feature extraction; after analysis, no images needed to be removed as none exceeded our similarity threshold, confirming that our evaluation is performed on genuinely distinct data.

Name	Purpose	Number of Classes	Total Images	Split Sizes (Train/Val/Test)
Garbage Classification	Hyperparameter selection, fine-tuning, and testing	6	2467	493/493/1481
TrashNet	Pretraining	2	22500	15750/6750/-
Recyclable and Household Waste Classification	Pretraining	30	15000	10500/4500/-

Table 1: Dataset details including split sizes. For Garbage Classification, class breakup cardboard (393), metal (400), glass (491), paper (584), plastic (472), trash (127). Pre-training datasets use only train/validation splits.

3.4 Training and Hyperparameter Selection

Models were trained and hyperparameters were selected according to Section 2. Seeds control random initialization of weights and the images in the train/validation split, but each model was evaluated on the same fixed test set. For SimCLR and SupCon, the same seed used in linear evaluation was maintained for full fine-tuning to ensure consistent representation of performance on held-out data. After hyperparameter selection, final models were retrained on the combined training and validation sets. For models that triggered early stopping during validation, training on the combined data continued until the previously determined optimal epoch.

4 Results

4.1 Table 2: Results

Method	Accuracy	Balanced Accuracy	Recall (Macro)	Precision (Macro)	F1 Score (Macro)
Transfer Learning (ImageNet)	92.04%	89.91%	89.91%	89.93%	89.88%
Transfer Learning (30 Class)	87.11%	81.53%	81.53%	86.31%	82.86%
Transfer Learning (Binary TrashNet)	86.13%	84.65%	84.65%	83.17%	83.65%
SimCLR	33%	31%	30.9%	27.9%	30.9%
SupCon	34%	32%	32.1%	26.8%	26.7%
Random Chance	17.42%	17.44%	17.44%	17.33%	16.78%
Most Common Class	23.54%	16.67%	16.67%	3.92%	6.35%

Table 2: Performance evaluation across different methods for recyclable materials classification. Transfer learning approaches significantly outperform contrastive learning methods, with ImageNet-pretrained transfer learning achieving the highest performance (89.91% Balanced accuracy). Both SimCLR (33%) and SupCon (34%) perform only marginally better than random chance (17.42%), contradicting our initial hypothesis that contrastive learning would excel at this task. We include random chance and most common class baselines for reference.

4.2 FIGURE 3: CONFUSION MATRIX OF THE BEST METHOD

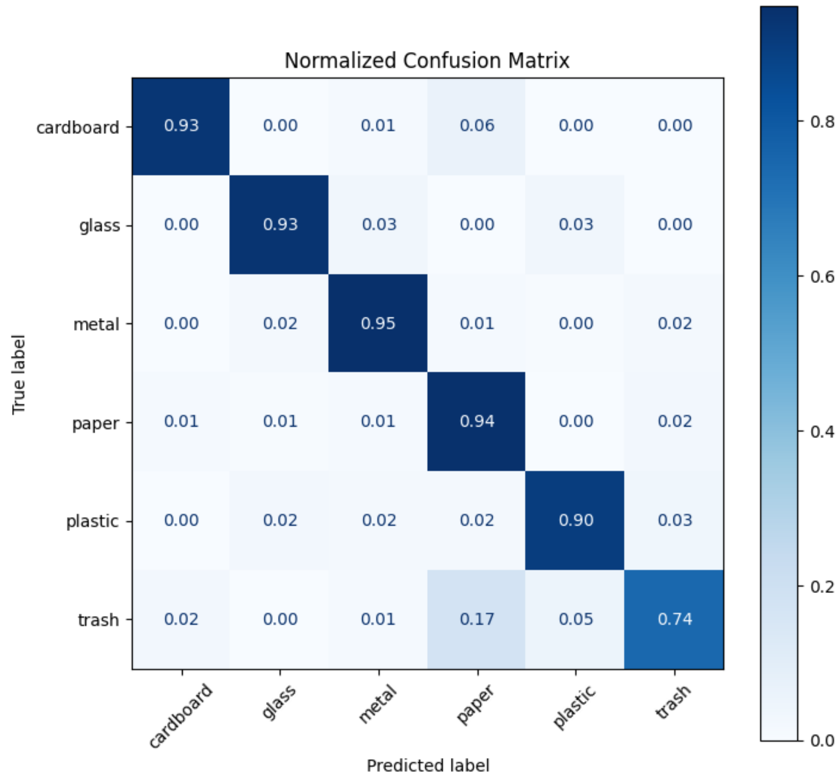

Figure 1: Normalized confusion matrix for our best performing model (Transfer Learning with Image-Net Weights) evaluated on the test set. The model demonstrates strong performance on recyclable materials, achieving over 90% accuracy for all five recyclable categories (cardboard: 93%, glass: 93%, metal: 95%, paper: 94%, plastic: 90%). However, it shows relatively weaker performance on the non-recyclable trash class (74%), with most misclassifications occurring as paper (17%) and plastic (5%)

5 RESULTS

Our experimental results contradict our initial hypothesis. As shown in Table 2, both SimCLR (33% accuracy) and SupCon (34% accuracy) performed only marginally better than random chance (17.42%), despite our hypothesis that supervised contrastive learning would excel at this task. This poor performance suggests that contrastive learning methods may have failed to learn sufficiently discriminative features from our limited domain-specific data, despite the theoretical advantages of class-aware contrastive losses.

In contrast, transfer learning approaches demonstrated strong performance, with ImageNet-pretrained transfer learning achieving the highest performance (92.04% accuracy, 89.91% balanced accuracy) followed by 30-class transfer learning (87.11% accuracy, 81.53% balanced accuracy). As shown in Figure 1, our best model achieves over 90% accuracy on all recyclable categories. However, it struggles significantly with the trash class (74% accuracy), commonly misclassifying it as paper (17%) or plastic (5%). This systematic error pattern suggests that while transfer learning from ImageNet provides robust general visual features, distinguishing non-recyclable waste remains a key challenge. This may be due to the the small size of the trash class available for fine tuning, or due to the wide range of items which could fall into this category. If its not any of the other ones, its trash.

6 REFLECTION AND OUTLOOK

Our experimental results yield several important insights about transfer learning and representation learning in the context of recyclable materials classification. The significant under-performance of SupCon and SimCLR compared to traditional transfer learning approaches suggests that the feature spaces learned through contrastive learning methods may have insufficient overlap with the target domain, despite their theoretical advantages. This highlights a crucial lesson for the field: domain relevance and general feature knowledge from large-scale pre-training can be more valuable than sophisticated learning mechanisms, particularly in specialized classification tasks like waste sorting.

Looking forward, a critical next step would be to extend this work from pure classification to object detection, where multiple items could be identified and classified simultaneously within a single image. This would better reflect real-world recycling scenarios, where waste materials often appear mixed together rather than in isolation. Such a system could leverage our current classification architecture as a backbone while adding region proposal networks to detect and localize individual items. Moreover, this system could be evaluated in conditions that approximate deployment scenarios, such as with materials moving through automated sorting systems on conveyor belts. Future work could also explore how temporal information from conveyor belt sequences might enhance detection accuracy, potentially providing more robust feature representations for complex, multi-object scenes.

References

cchangcs. Garbage classification, 2018. https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification/data.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *CoRR*, abs/2004.11362, 2020. https://arxiv.org/abs/2004.11362.

Alistair King. Recyclable and household waste classification, 2024. https://www.kaggle.com/datasets/alistairking/recyclable-and-household-waste-classification/data.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CoRR*, abs/2201.03545, 2022. https://arxiv.org/abs/2201.03545.

Sashaank Sekar. Trashnet: Waste classification data, 2019. https://www.kaggle.com/datasets/techsash/waste-classification-data/data.

Thalles Silva. Simclr: A simple framework for contrastive learning of visual representations, 2021. https://github.com/sthalles/SimCLR. GitHub repository.

Yonglong Tian. Supcontrast: Supervised contrastive learning. https://github.com/HobbitLong/SupContrast, 2020. Accessed: 2024-06-17.