Automated Waste Management using Deep Learning Models

CPSC-393

Edward Yu, Julian Ting, Greg Lee

1. Introduction

The global waste management crisis is escalating at an alarming rate. Due to rising consumption and inadequate infrastructures, the daily volume of waste generated exceeds the processing capacity of traditional waste disposal systems. A significant issue with current management methods lie within the segregation process, where waste is sorted at processing facilities. Improper waste disposal results in recyclable materials contaminating landfills, which causes many environmental problems. Some consequences include contaminating soil and water, the production of toxic leachates, and greenhouse gas emissions.

In a world facing growing waste management challenges, the ability to automatically identify and categorize waste is crucial to improving recycling efficiency and promoting environmental sustainability. Manual sorting is an inefficient process that is time consuming, costly, and prone to errors, especially when waste streams are mixed.

In this work, we are primarily concerned with developing an automated waste classification system through deep learning. Our objective is to have the model distinguish between organic and recyclable waste materials using image classification, and we have two main goals. First, we want to evaluate whether a CNN built from scratch or a generalized transfer learning model offers superior performance for waste identification. Second, we want to analyze the unsupervised latent space for a better understanding of the intrinsic structure of the waste data. We trained an autoencoder to be able to create clusters on the embeddings/latent representations to showcase the different types of items in the dataset.

2. Analysis

We use the Waste Classification dataset by *techsash* found on Kaggle. It consists of 25,000 total images, split into a training set of 22,564 images and a test set with 2,513 images. Sample images from the dataset are shown below in Figure 1.

Figure 1: Sample Training Images

2.1. Preprocessing and Data Augmentation

We found that the images in the dataset have high visual similarity, making it more challenging for feature extraction. Each waste image is assigned a class label of either “Organic” or “Recyclable”. In preprocessing, we implemented a pipeline using ImageDataGenerator from Keras for rescaling and data augmentation. We first normalized pixel values to the range of [0, 1] by rescaling the pixel values by 1/255. The training data was further split into a 80-20 train-validation set, leaving the model with 18,052 images for training and 4,512 images for validation.

We applied data augmentation to the training images to expose the model to more realistic scenarios. We added rotation shifts of 20 degrees to teach the model to recognize images regardless of a tilt or not. Additionally, we also added horizontal and vertical shifts to simulate different image positions. Furthermore, we used shear and zoom transformations along with horizontal flips for recognizing images in different directions and distances. These augmentations may sacrifice some accuracy, but they simulate real-world scenarios where objects aren't always perfectly captured, leading to better generalization.

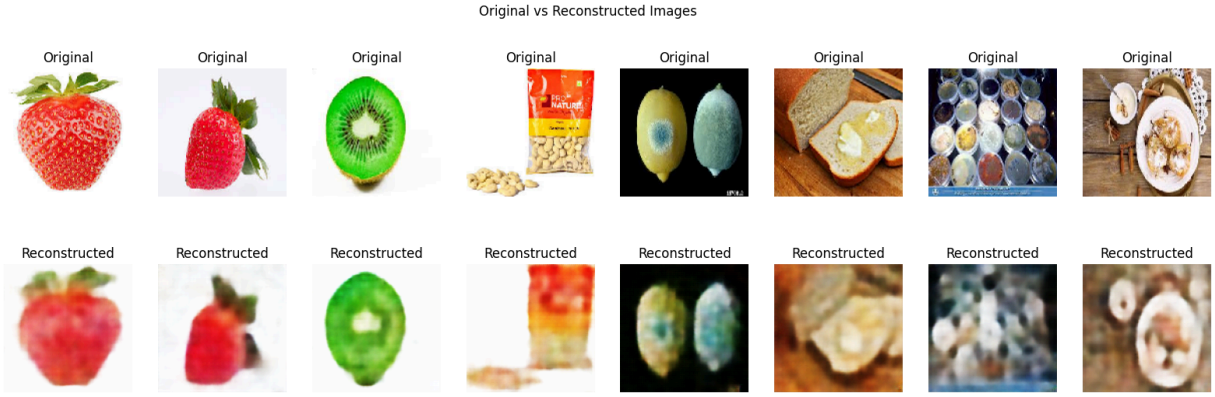


Figure 2: Autoencoder Reconstructed Images

In addition to supervised classification methods, we used an autoencoder as an unsupervised method to better understand the structure of the waste image data. The autoencoder learns to compress images into a lower dimensional latent representation that captures the most important visual features of each image. These representations help reveal what visual patterns distinguish recyclable from organic waste beyond simple labels. In Figure 2 above, we can visualize waste and recycling photos where images like the strawberry and the kiwi have a more patterned texture while the image of the smooth bag of nuts has a smooth not really rough texture and we can visualize that better from a lower dimensional latent representation.

3. Methodology

For our models, we implemented three architectures: CNN, transfer learning, and convolutional autoencoder.

3.1. Convolutional Neural Network:

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 111, 111, 32)	896
max_pooling2d (MaxPooling2D)	(None, 55, 55, 32)	0
batch_normalization (BatchNormalization)	(None, 55, 55, 32)	128
conv2d_1 (Conv2D)	(None, 27, 27, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 13, 13, 64)	256

conv2d_2 (Conv2D)	(None, 6, 6, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 5, 5, 128)	512
flatten (Flatten)	(None, 3200)	0
dense (Dense)	(None, 512)	1,638,912
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1,026

Total params: 1,734,082 (6.61 MB)

Trainable params: 1,733,634 (6.61 MB)

Non-trainable params: 448 (1.75 KB)

We employ a multi-layer convolutional neural network (CNN) for binary classification. Our CNN is designed to efficiently extract features from images while maintaining computational efficiency.

The model consists of three convolutional blocks, each responsible for learning increasingly complex visual features. The depth in each progressive block increases from 32 to 64 to 128 filters. Early layers (Block 1) allow the network to learn low-level features such as edges, textures, and color gradients. Middle layers (Block 2) capture more structured patterns

like shapes and object parts. Deeper layers (Block 3) learn high-level, abstract features useful for final classification.

Each convolutional block utilizes a 3x3 kernel and includes strided convolutions, MaxPooling layers, and Batch Normalization. Strided convolutions reduce the spatial dimensions of feature maps early in the network. This lowers computational cost and accelerates training while preserving important spatial information. Each convolutional layer is followed by a MaxPooling layer, which further downsample feature maps and makes our model more robust to small translations. To ensure training stability, we used Batch Normalization after each pooling layer. This normalizes the activations, allowing for faster convergence and the use of higher learning rates.

As the network goes deeper, the number of filters increases while the spatial resolution decreases. This reflects a common CNN design pattern: trading spatial detail for richer feature representations. After feature extraction, the Flatten layer converts the final feature maps into a 1D vector and passes them through a Dense layer of 512 neurons with ReLU activation. To reduce overfitting, we added a Dropout layer with a rate of 0.3. The final Dense output layer with 2 units produces class probabilities for binary classification.

3.2. Transfer-Learning-Model (MobileNetV2):

```
Model: "sequential_1"
┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
```

mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
batch_normalization_3 (BatchNormalization)	(None, 1280)	5,120
dropout_1 (Dropout)	(None, 1280)	0
dense_2 (Dense)	(None, 256)	327,936
batch_normalization_4 (BatchNormalization)	(None, 256)	1,024
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 2)	514

Total params: 2,592,578 (9.89 MB)

Trainable params: 331,522 (1.26 MB)

Non-trainable params: 2,261,056 (8.63 MB)

To evaluate our custom CNN's performance, we decided to use a transfer learning approach. We selected MobileNetV2 because the model was pretrained on the ImageNet dataset, and also for its light architecture and high computational efficiency. The training process followed two phases.

First, the base model was frozen and trained using only the newly added top layers. This allowed the classifier to learn meaningful decision boundaries using stable pretrained features. We removed the original classification head and kept the convolutional backbone, which outputs a $7 \times 7 \times 1280$ feature map. To adapt MobileNetV2 to our waste classification task, we added a custom classification head, consisting of Global Average Pooling, Batch Normalization, Dropout, and a Dense layer with 256 units. This allowed the classifier to learn meaningful decision boundaries using stable pretrained features.

Second, we unfroze the base model and fine-tuned the last 30 layers of MobileNetV2, retraining the network with a lower learning rate of 1e-5. This enables the model to adapt and slightly adjust higher-level features to better match the visual characteristics of waste images. Fine-tuning only the deeper layers strikes a balance between task adaptation and overfitting control, while keeping most pretrained weights fixed. Overall, the transfer learning model provides better feature quality, faster training, and lower computational cost compared to a fully custom CNN, while achieving strong performance by leveraging pretrained knowledge.

3.3. Convolutional Autoencoder

We aimed to answer whether waste types cluster naturally through our autoencoder. It consists of an encoder and decoder. The encoder compresses the 224x224 input image into a compact latent vector of size 512 using convolutional layers with strided downsampling (filters 32, 64, 128). The decoder uses Conv2DTranspose layers to upsample the latent vector back to its

original image dimensions. The model was trained to minimize the MSE between the original input and reconstructed images. After training, we extracted the 512-dimensional latent vectors for the test set and applied K-Means clustering with k = 2. The Adjusted Rand Index (ARI) was employed to quantify how well these unsupervised clusters aligned with the true semantic labels of Organic and Recyclable.

4. Results

CNN	Precision	Recall	F1-Score	Support
Organic	0.8961	0.9415	0.9182	1401
Recyclable	0.9212	0.8624	0.8908	1112
Accuracy (Weighted Avg)	0.9072	**0.9065**	0.9061	2513

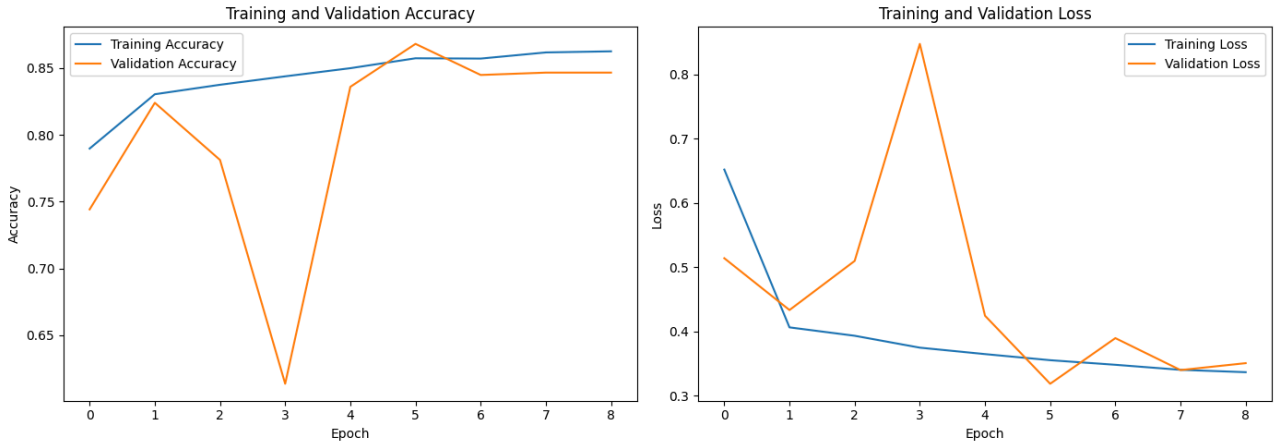


Figure 3: CNN Accuracy and Loss Plots

TLM	Precision	Recall	F1-Score	Support
Organic	0.8400	0.9814	0.9052	1401
Recyclable	0.9703	0.7644	0.8551	1112
Accuracy (Weighted Avg)	0.8976	**0.8854**	0.8830	2513

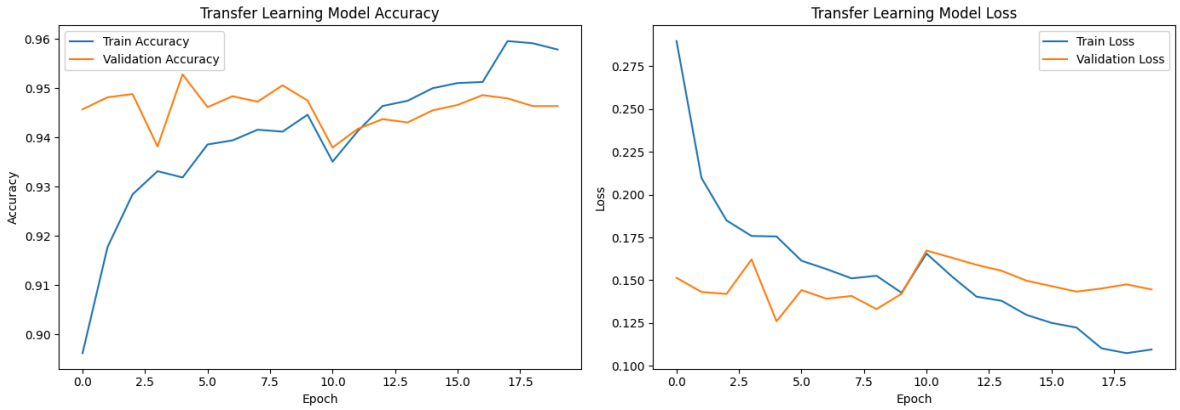


Figure 4: Transfer Learning Accuracy and Loss Plots

4.1. Supervised Model Performance

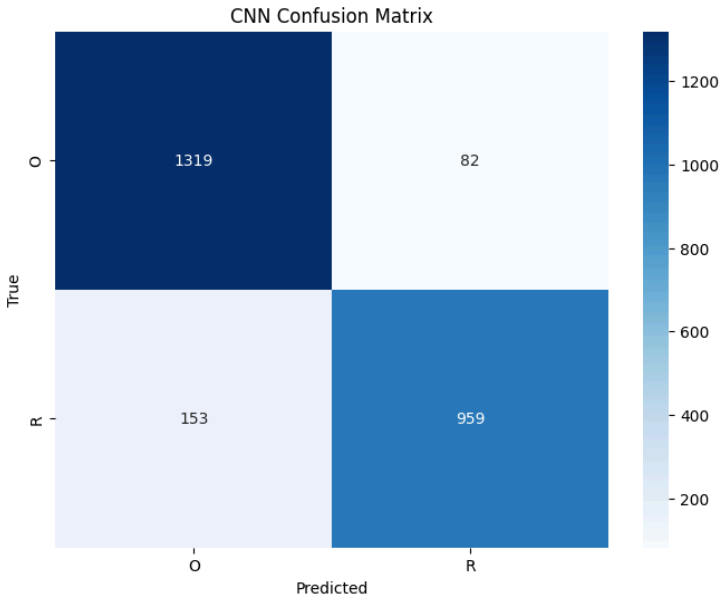
Contrary to what we expected, our CNN actually outperformed the transfer-learning MobileNetV2 model overall, achieving 90.65% overall accuracy compared to 88.54%. The classification report for the CNN showed a more balanced performance between organic and recyclable classes. This suggests our model was successful in learning specific features for waste material.

On the other hand, the transfer-learning model showed very high precision for organic waste but struggled more with recall on recyclable items. It exhibited heavy bias towards the Organic class, with a recall of 98.14% but dropping significantly to 76.44% for recyclables. This implies that the pre-trained model defaulted to classifying ambiguous items as Organic. It often mislabeled recyclable waste as organic, leading to more false negatives in that category.

However, when we look at the accuracy and validation loss of both models, we see that the CNN validation accuracy and loss has jumps in the graph indicating that there is possible overfitting leading to the model learning the training data too well, including its noise and specific characteristics. In our transfer learning model we see a consistent increase in our training

accuracy and validation accuracy and a consistent loss in our training and validation loss which is really good indicating that our model is not over or underfitting.

Error Analysis:



True Positives: 1319 organic waste correctly classified.

True Negatives: 959 recyclable waste correctly classified.

False Negatives: 153 errors where the model predicted organic when it was recyclable.

False Positives: 82 errors where the model predicted recyclable when it was organic.

Figure 5: CNN Confusion Matrix

In Figure 5, we can see that our model misclassified Recyclable images as organic twice as much as organic images as recyclable.

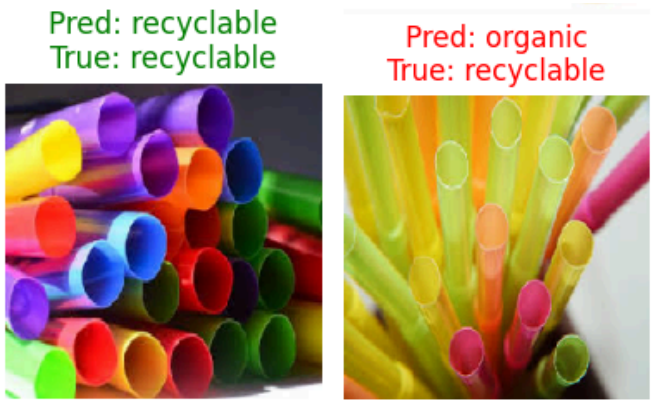


Figure 6: Correct vs Incorrect Predictions

Our CNN misclassifies recyclable items as organic more often because recyclable items have much more visual diversity and inconsistent backgrounds, illustrated above in Figure 6. Organic waste has more uniform textures and colors, so the model forms a stronger representation for “organic.” Thin plastic items like straws can resemble twigs or sticks depending on lighting and background, which explains why one straw was classified correctly while another was misclassified.

4.2. Latent Space Visualizations and Clustering

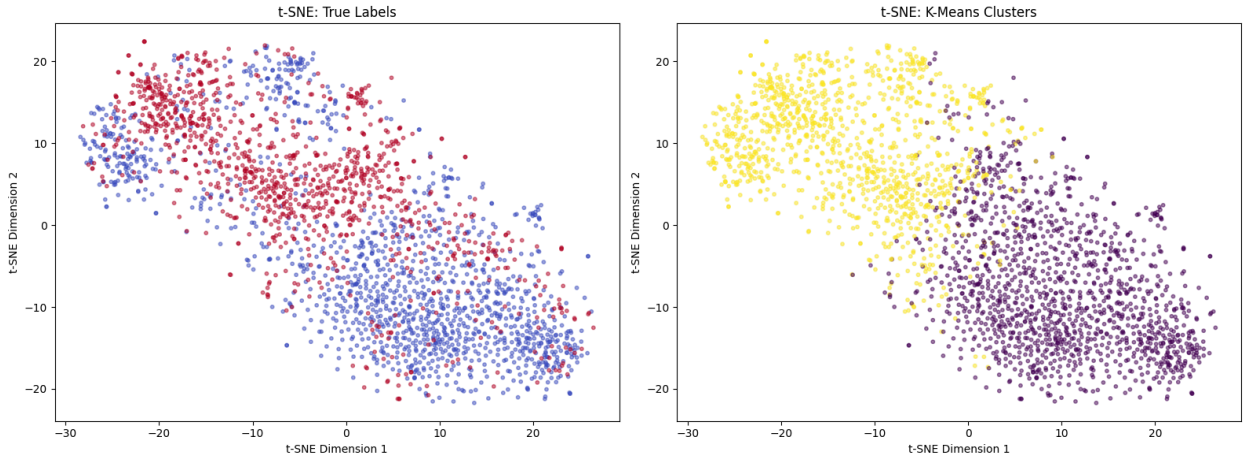


Figure 7: t-SNE Visualizations

Our autoencoder provided us with critical insights into the data’s structure. The model had a final validation loss of approximately 0.02, indicating that it was successful in compressing and reconstructing the images (See Figure 2). However, the K-Means clustering of the latent space resulted in an ARI score of 0.1589. Since an ARI of 0 indicates random labeling, this low score demonstrates that the visual clusters found do not correlate with material category in this domain, and visual appearance is a poor predictor for the waste material types.

We first employed t-SNE to project the 512-dimensional latent vectors into 2D space. The True Labels plot (Figure 7) reveals that the actual classes have significant overlaps and are

intermixed throughout the latent space, proving that the classes share many significant visual similarities. This indicates a semantic gap where without labels, the model groups items by their visual similarity rather than by material type. Conversely, the K-Means Clusters plot (Figure 7) shows a more defined separation between the clusters. K-Means grouped images by low-level visual shape and color rather than by actual category.

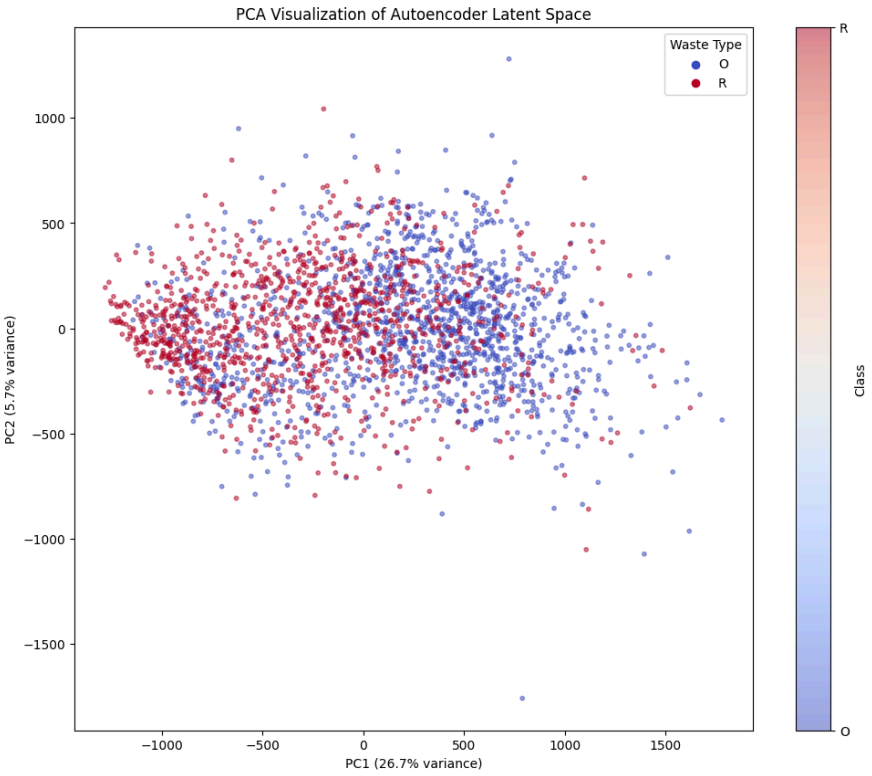


Figure 8: PCA Visualization

We also utilized a PCA visualization (Figure 8 above) to examine the variance of the embeddings. The scatter plot shows a dense, overlapping distribution between the classes with no defined linear decision boundary, confirming our t-SNE findings. PCA reduces the 512 dimensions to just 2, and our PCA analysis shows that the first two dimensions explain only 32.36% of the total variance (PC1: 26.67%, PC2: 5.69%). This result indicates that the

information necessary to distinguish these images is spread across many dimensions and is not easily reducible to simple linear factors. The lack of separation in both the PCA and t-SNE projections show us that an unsupervised system based solely on reconstruction loss is insufficient for waste classification, as it fails to distinguish textural differences that separates recyclables from organic matter.

When we project this into a 3D visualization of the latent space (see below in Figure 9), we notice that the classes form clouds in the 3D space. The separation becomes much clearer here than in 2D, indicating that the data is separable but the boundary is complex. This visualization justifies our use of a Deep Learning model because a neural network can learn to navigate this complex, high-dimensional manifold to find the right separation. The goal of these autoencoders is image reconstruction - not image classification. This means that even a deep autoencoder may not fully perform image classification simply because it has learned to compress the image into a latent dimension vector.

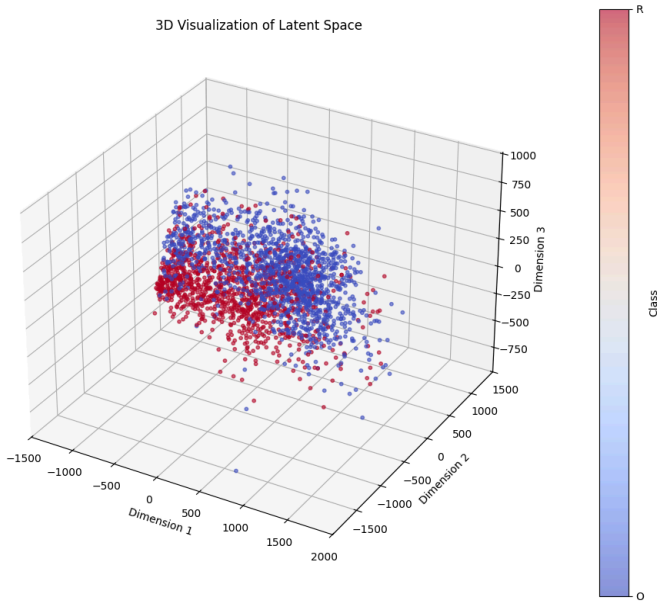


Figure 9: 3D Latent Space Visualization

5. Reflection

This project demonstrated that waste classification can be automated through the use of deep learning models, but requires architectural considerations on our end. The superior performance our CNN model demonstrated over the transfer learning model indicates that for domains where texture and material properties matter more, a smaller and specialized network can outperform a generalist model. This finding highlights a significant insight: recyclable and organic waste items often lack the universal learned shapes and features that models like MobileNetV2 are trained on. In our case, the pre-trained features were heavily biased and favored organic matter, causing it to misclassify recyclable waste.

One of our goals was to analyze the unsupervised latent space for a better understanding of the intrinsic structure of the waste data. While we achieved high classification accuracy with the CNN (90.65%), the autoencoder revealed limitations with our analysis. Notably, the autoencoder proved that visual similarity does not equate to semantic class labels. Our low ARI score of 0.1589 and overlapping t-SNE visualizations highlight this. Furthermore, it indicates that without supervised labels, a model based on reconstruction loss will group items by low-level features such as color or shape rather than the actual material composition.

Further improvements to our model include changing the task from binary image classification to multi-class architecture. Instead of just organic and recyclable waste, we can add different material types such as glass or plastic to the data. Additionally, we propose changing the supervised task to object detection rather than classification. Using an architecture such as YOLO would allow the system to identify overlapping images simultaneously. To resolve our failed unsupervised clustering, we would use a different framework instead of an autoencoder.

Contrastive learning methods, such as SimCLR, would force the model to learn visual representations that are distinct between classes.