

Waste or Recycle?

By the AI Abusers:
Edward, Julian, Greg



Our Project



We are advocating for a better waste management system, as most waste ends up in landfills.

<u>Problems</u>	Inefficient waste management leads to excessive waste ending up in landfills.
<u>Increase in landfills</u>	Leads to environmental damage from pollution like methane and leachate, which can contaminate soil and water
<u>Increase in toxins</u>	leads to land, water, and air pollution
<u>Consumption of toxic waste by animals</u>	Without proper waste classification and management, animals consume toxic waste through various pathways, primarily by mistaking waste for food, the contamination of their environment

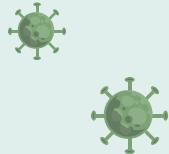


In a world facing growing waste management challenges, the ability to automatically identify and categorize waste is crucial to improving recycling efficiency and promoting environmental sustainability. Manual sorting is time consuming, costly, and prone to error, especially when waste streams are mixed.



SOTA: Smart Bin Systems

- These systems have gained traction recently because of their ability to optimize waste collection processes.
- Capabilities:
 - **Real-Time Monitoring** : They are equipped with sensors that monitor the fill levels of bins in real time. Continuous tracking prevents overflows.
 - **Route Optimization** : The collected data is then analyzed using AI algorithms, such as deep learning, to determine the optimal collection routes and schedules.
- Technologies:
 - **Advanced Sensors** : Integrating ultrasonic sensors and weight sensors provides more accurate data on fill levels.
 - **IoT Infrastructure** : Enables seamless communication between the bins and waste management authorities



Data Overview

Dataset: Waste Classification Data (Kaggle)

<https://www.kaggle.com/datasets/techsash/waste-classification-data>

Separated into a training and testing folder

Train:

The train folder contains 22564 images divided into Organic (O) and Recyclable (R)

Test:

The test folder contains 2513 images divided into Organic (O) and Recyclable (R)

Train: 80%

Validation: 20%

Test: Provided Kaggle test set



Exploratory Data Analysis/Preprocessing



**Image
Cleaning:**
Resized to 224x224
Scaled pixel values to [0,
1]

Random rotations
Shifts(horizontal/vertica
l)
Brightness adjustment



Model Architecture

CNN:

- We used a three-block architecture where each block increases the number of filters to learn more complex visual features.
- After flattening, we used a dense layer of 512 units with dropout (0.3) to prevent overfitting.
- Batch normalization keeps training stable and strides/pooling help reduce image size efficiently.

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)

Transfer Learning Model (MobileNetV2):

- We used MobileNetV2 as a feature extractor and added our own classification head.
- In Phase 1, we froze the base model and trained the new layers
- In Phase 2, we unfroze and fine-tuned the last 30 layers of MobileNetV2 so it could adapt to waste images without overfitting

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)

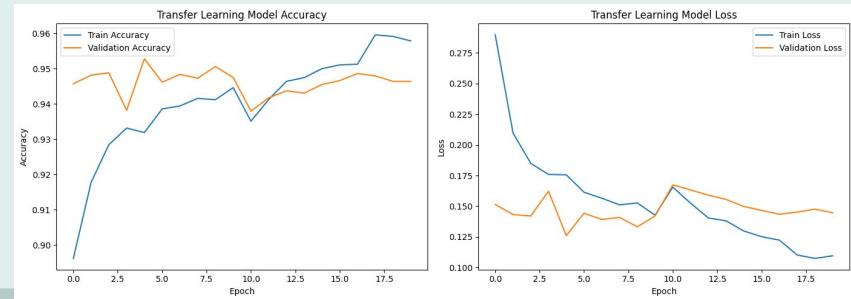
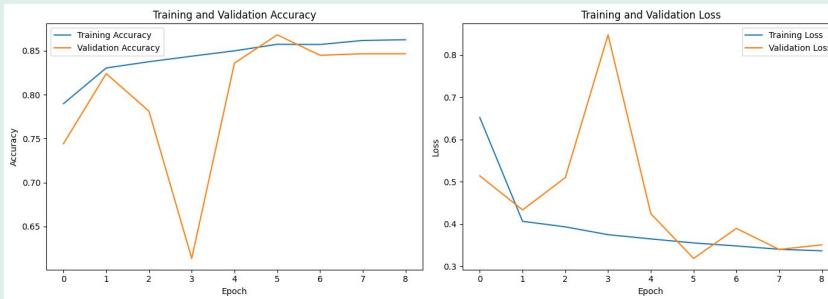
Classification Results

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

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



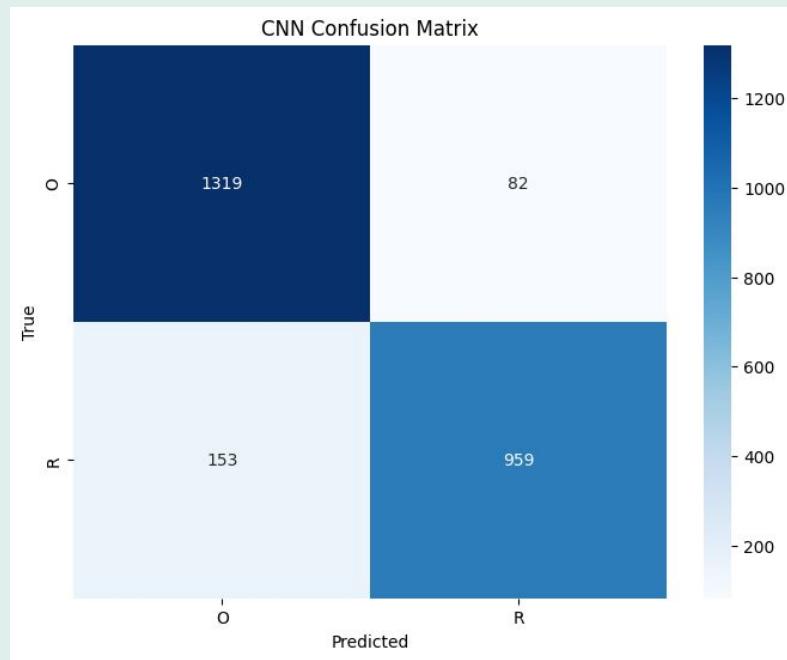
Error Analysis

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.



Prediction Results

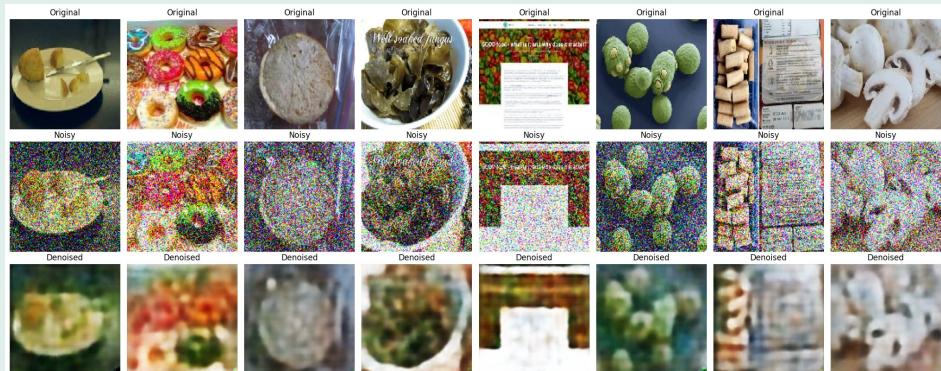
Correct



Incorrect



Autoencoder



Original vs Reconstructed Images



Autoencoder

Architecture

Encoder: The 224x224 images were compressed into a 512-dimensional latent vector.

Bottleneck: The 512-vector forces the model to prioritize the most significant features.

Decoder: Reconstructs the original image from the latent vector.

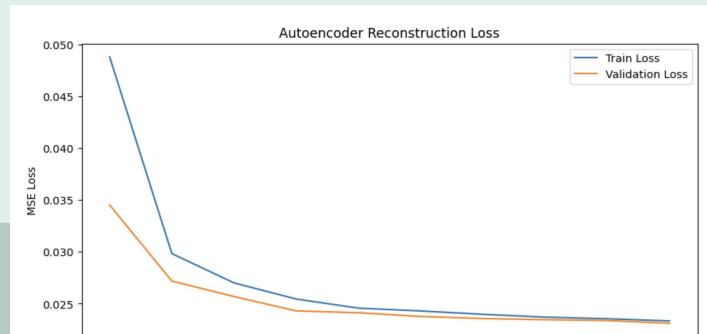
Original vs Reconstructed Images



Reconstruction Results

MSE Loss: 0.0200

The extremely low loss indicates that the 512-dimensional vector successfully captures ~98% of the visual information and suggests that the model "understands" the appearance of waste. The reconstructed images are blurry but still recognizable

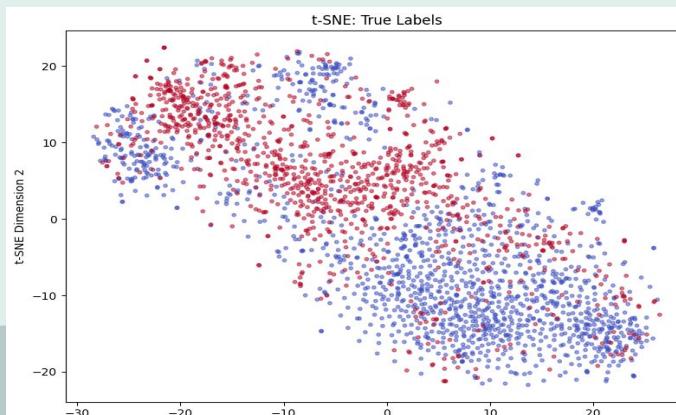


Clustering Visualizations

t-SNE (True Labels)

We see significant overlaps between the red and blue points, proving that the classes share significant visual similarities.

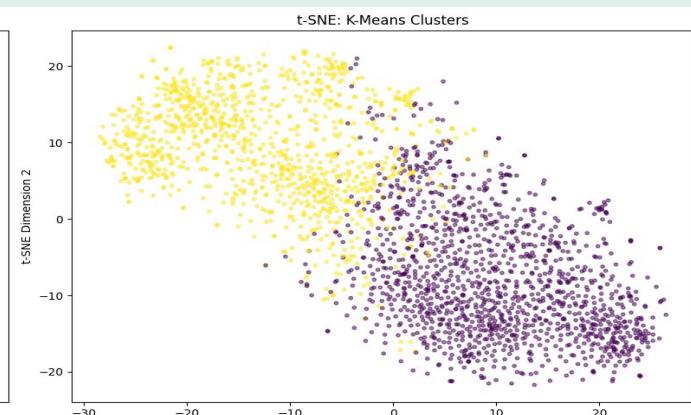
Indicates a semantic gap - without labels, the model groups items by how visually similar they are rather than by material type.



t-SNE (K-Means)

Clustering Metrics:
ARI: 0.1589 | NMI: 0.1147

We see a more defined separation between the clusters. K-Means grouped them by shape and color rather than category.



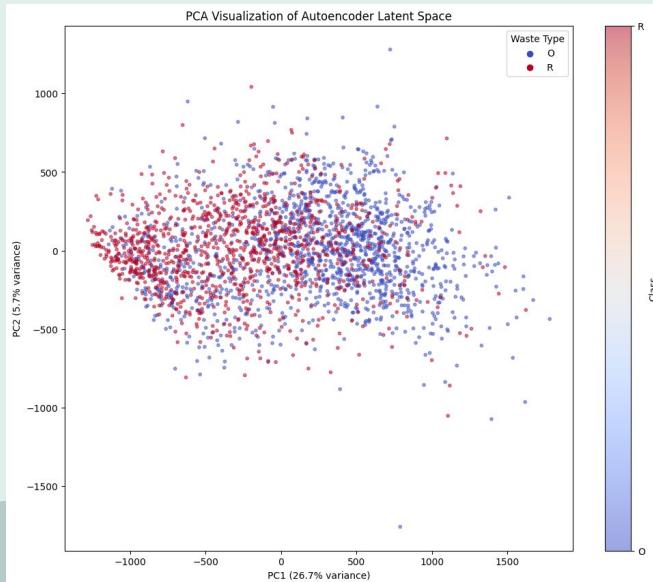
Latent Space Visualization

PCA Analysis:

(PC1: 26.67%, PC2: 5.69%)

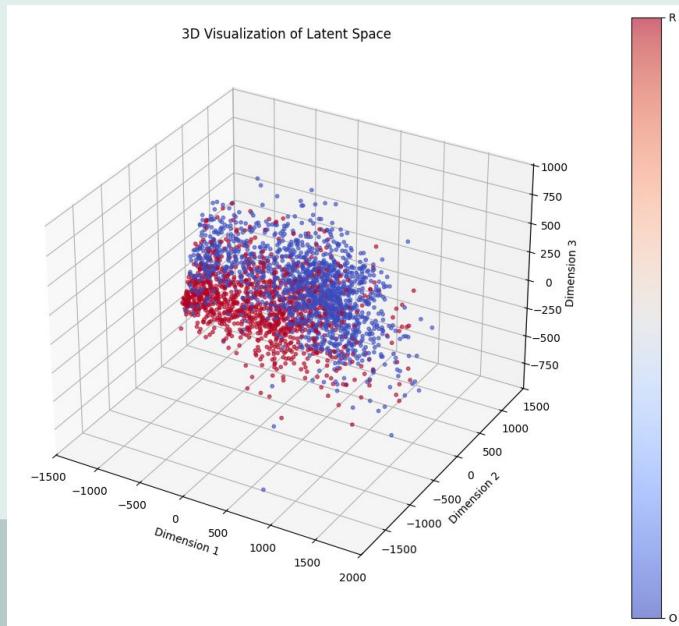
Total variance explained: 32.36%

- PCA captures ~ $\frac{1}{3}$ of the data complexity.
Clusters are visible but overlap, meaning this is a high-dimensional problem.



3D Visualization:

We test if the cluster overlap also appears in 3D. Here, the classes form “clouds” in 3D space, making the separation clearer than the 2 dimensional PCA plot.



Conclusion

1. Classification Models:
 - a. Our CNN achieved an accuracy of 90.65%, outperforming the Transfer Learning Model at 88.54%.
 - b. This result is meaningful because with >90% precision, our light architecture shows that automated waste sorting can be done without needing industrial hardware.
2. Unsupervised Learning Autoencoder:
 - a. The model was successful in reconstructing waste images given it's reconstruction loss of 0.02.
 - b. However, our clustering analysis and low ARI score of 0.1589 revealed that visual similarity does not equate to material type.
3. Conclusion:
 - a. We successfully built a CNN that classifies waste with high accuracy while providing interpretability into why models make the decisions they do.

Future Steps For Improvement

More Data Augmentation

Implement further transformations (e.g., color jitter, noise injection) to improve model robustness against real-world variability.



Multi Class Expansion

Instead of just organic and recycle waste, we can add other material types, such as glass or plastic.

Optimizing Transfer Learning

Conducting more research on different models and tuning the hyperparameters to address our overfitting issues.

References

- Olawade, D. B., Fapohunda, O., Wada, O. Z., Usman, S. O., Ige, A. O., Ajisafe, O., & Oladapo, B. I. (2024). *Smart waste management: A paradigm shift enabled by artificial intelligence*. Waste Management Bulletin, 2(2), 244-263. <https://doi.org/10.1016/j.wmb.2024.05.001>
- Bank, D., Koenigstein, N., & Giryes, R. (2020). *Autoencoders*. ArXiv. <https://arxiv.org/abs/2003.05991>
- Van der Maaten, L., & Hinton, G. (2008). *Visualizing Data using t-SNE*. Journal of Machine Learning Research, 9(11), 2579-2605. <https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>

Questions?