Garbage Classification using Transfer Learning

Final Project Report - CS-GY-6923 Machine Learning

Chirag Mahajan
NET ID: cm6591

Abstract

Effective waste management is crucial for sustainable urban development. A key component of advanced waste management systems is the precise classification of waste materials. This project investigates the use of transfer learning for the classification of garbage, employing a model pre-trained on a substantial dataset to improve the accuracy of classifying multiple waste categories with limited data. The project utilized around 22,000 images, distributed across ten different waste types. The adaptation of the EfficientNetV2S model, initially trained on the ImageNet dataset, demonstrated significant improvements in the performance of the classification tasks, particularly in data-scarce environments. The process involved pre-cleaning the dataset to remove non-image files and resizing images for uniformity, followed by fine-tuning the pre-trained model using transfer learning techniques. The outcomes highlight a considerable increase in accuracy, demonstrating the potential of transfer learning in streamlining waste segregation processes. This project not only offers a scalable solution for garbage classification but also illustrates the applicability of machine learning techniques in optimizing resource management in environmental contexts.

Table of Contents

I. Introduction

Waste management is a critical issue for urban environments, influencing ecological sustainability and public health. Proper segregation of waste, a foundational step in waste management, ensures that recyclable materials are effectively processed and non-recyclable materials are appropriately disposed of. However, traditional methods of waste segregation are labor-intensive and often prone to human error, underscoring the need for automated systems that can reliably classify waste into its respective categories.

The advent of machine learning has opened new avenues for solving complex problems across various domains, including environmental management. Particularly, the field of image recognition has seen rapid advancements, thanks to deep learning algorithms. However, the application of these algorithms often requires large, well-labeled datasets and substantial computational resources, which may not be feasible in all scenarios, especially in under-resourced settings.

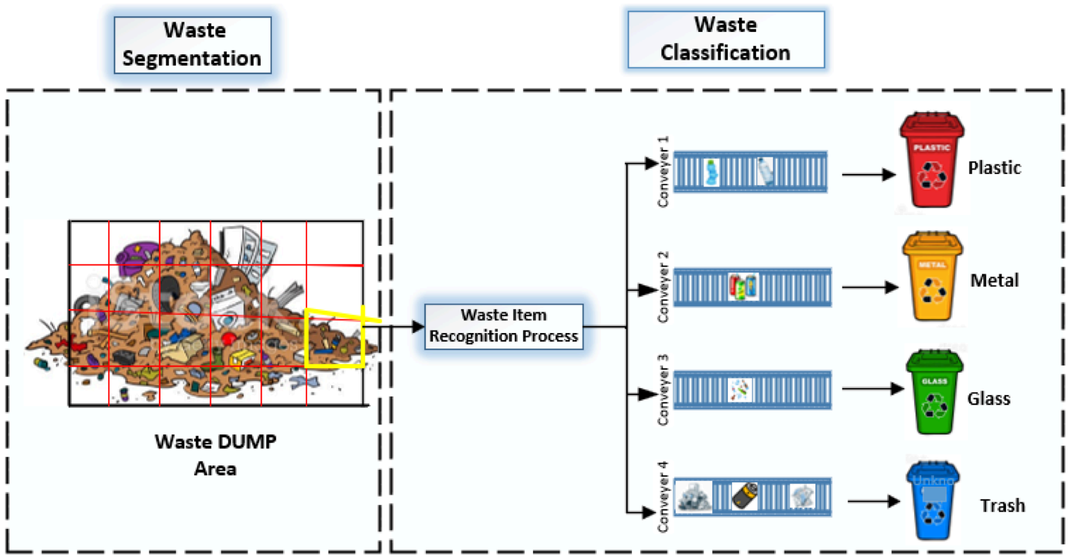

Figure 1.1 : Waste Segmentation and Classification Mechanism

Transfer learning has emerged as a potent solution to this problem. By utilizing a model pre-trained on a large dataset and fine-tuning it to specific tasks with smaller datasets, transfer learning can significantly reduce the need for extensive data and computational power. This project leverages the EfficientNetV2S model, a convolutional neural network pre-trained on the ImageNet dataset, to classify images of garbage into multiple categories. This approach not only enhances the accuracy of classifications but also minimizes the resources required for training the model.

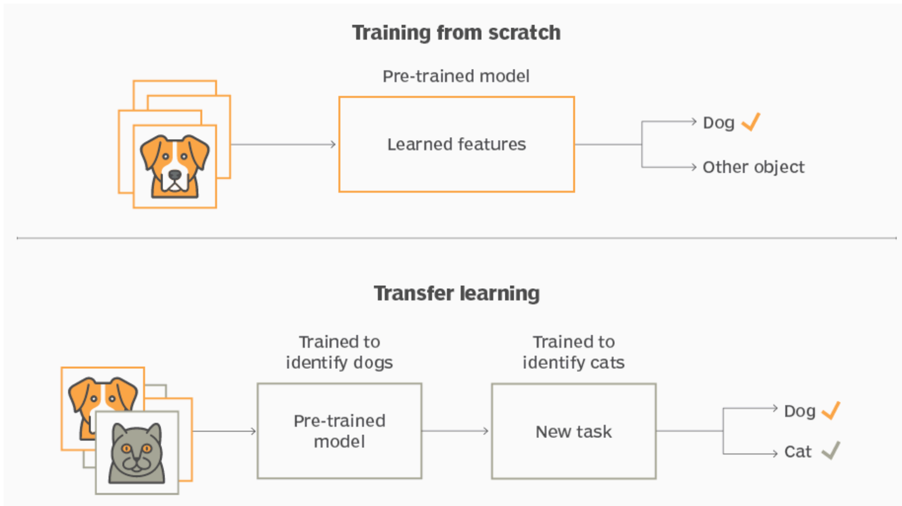

Figure 1.2 : How Transfer Learning works

This paper details the process of adapting a pre-trained model to classify waste types using a dataset of approximately 22,000 images categorized into ten classes, including metal, glass, and plastic, among others. We discuss the challenges of working with imbalanced and limited datasets and describe our methodologies for preprocessing the data, selecting the model, and fine-tuning the learning process to achieve high precision and recall in classification tasks.

Through this project, we aim to demonstrate the feasibility and effectiveness of applying advanced machine learning techniques, particularly transfer learning, to improve waste management systems, thus contributing to environmental sustainability efforts globally.

II. Dataset

The core of this project revolves around the application of transfer learning for the classification of waste using a publicly available dataset known as Trashnet. This dataset is essential for training the machine learning model to recognize and categorize different types of waste accurately.

The Trashnet dataset comprises a total of 18,859 images, which are divided across ten distinct classes representing various types of waste materials. These categories include commonly found waste items such as plastic, metal, cardboard, clothes, glass, biological, battery, paper, shoes and other trash that are routinely processed in recycling and waste management facilities. Each category is crucial for understanding the diversity of waste materials that must be accurately identified in a real-world waste management context.

The images in the dataset are of varying resolutions and are stored in common file formats conducive to processing via the TensorFlow framework. This diversity in image resolution and format presents a realistic challenge that any practical machine learning application in waste management would encounter, namely the variability of input data quality.

Figure 2.1 : Visualization of data

The dataset's primary role in this project was to train a baseline convolutional neural network (CNN) model and subsequently fine-tune a pre-trained EfficientNetV2S model through transfer learning. The varied and comprehensive nature of the dataset challenges the model to learn and generalize across a broad spectrum of waste types, which is critical for the success of automated waste classification systems.

III. Data Preprocessing

Data preprocessing is a crucial step in any machine learning workflow, particularly when working with real-world data like images. In this project, the Trashnet dataset underwent several preprocessing stages to ensure that it was optimally formatted for training the machine learning models.

1. Cleaning and Filtering

- Image File Validation: The first step involved validating that all files in the dataset were indeed images. This was achieved using Python's `imghdr` library, which identifies files based on their headers. Files that were not recognized as images were removed to prevent errors during the training process.

- Format Compatibility: TensorFlow, the framework used for building the models in this project, has specific requirements regarding image formats. Only images in `.png`, `.jpg`, `.jpeg`, `.bmp`, `.gif`, and a few other formats are compatible. Therefore, images in non-compatible formats were either converted to acceptable formats or excluded from the dataset.

2. Data Augmentation

- Augmentation Techniques: To enhance the robustness of the model against variations in new, unseen data, data augmentation techniques were applied. These included random rotations, translations, zooms, and flips. Such transformations help the model generalize better by simulating different ways in which images of waste might appear in real-world scenarios.

- Implementation: Augmentation was implemented using TensorFlow's `ImageDataGenerator` class, which allows for the specification of various transformations that are applied to the images as they are fed into the model during training. This not only diversifies the training data but also effectively increases the dataset size without the need for additional images.

3. Resizing and Normalization

- Uniform Image Size: All images were resized to a uniform size of 400x400 pixels. This standardization is necessary because the convolutional neural network (CNN) architecture that processes these images requires input dimensions to be consistent.

- Pixel Value Scaling: The pixel values of the images, originally ranging from 0 to 255, were scaled down to a range of 0 to 1. This normalization improves the convergence speed during training and leads to better overall model performance.

4. Splitting the Dataset

- Training, Validation, and Testing Sets: The dataset was split into three subsets: training (80%), validation (10%), and testing (10%). This split facilitates the different phases of the machine learning workflow:

- Training Set: Used to train the model, where the model learns to classify images based on the provided labels.

- Validation Set: Used to adjust hyperparameters and prevent overfitting. This set acts as a pseudo-test set to gauge the model's performance during training.

- Testing Set: Used only after the model has been trained and tuned, to evaluate its performance on completely unseen data, simulating how it would perform in real-world conditions.

5. Random Undersampling

- Addressing Class Imbalance: The initial dataset showed significant class imbalance, with some classes like 'Clothes' and 'Glass' having many more images than others such as 'Battery' or 'Trash'. To prevent the model from developing a bias towards the more represented classes, random undersampling was employed.

- Implementation: This process involved reducing the number of images in the over-represented classes to a set threshold. For instance, if a class had more than 1,000 images, random images were deleted from that class until only 1,000 remained. This was achieved programmatically using a script that iterated through each class directory, randomly selected excess images, and removed them from the dataset.

- Result: The result was a more balanced dataset where each class had either 1,000 images or the maximum number available if the original count was below 1,000. This balancing allows for more equitable learning where the model is less likely to be influenced by the predominance of any single class.

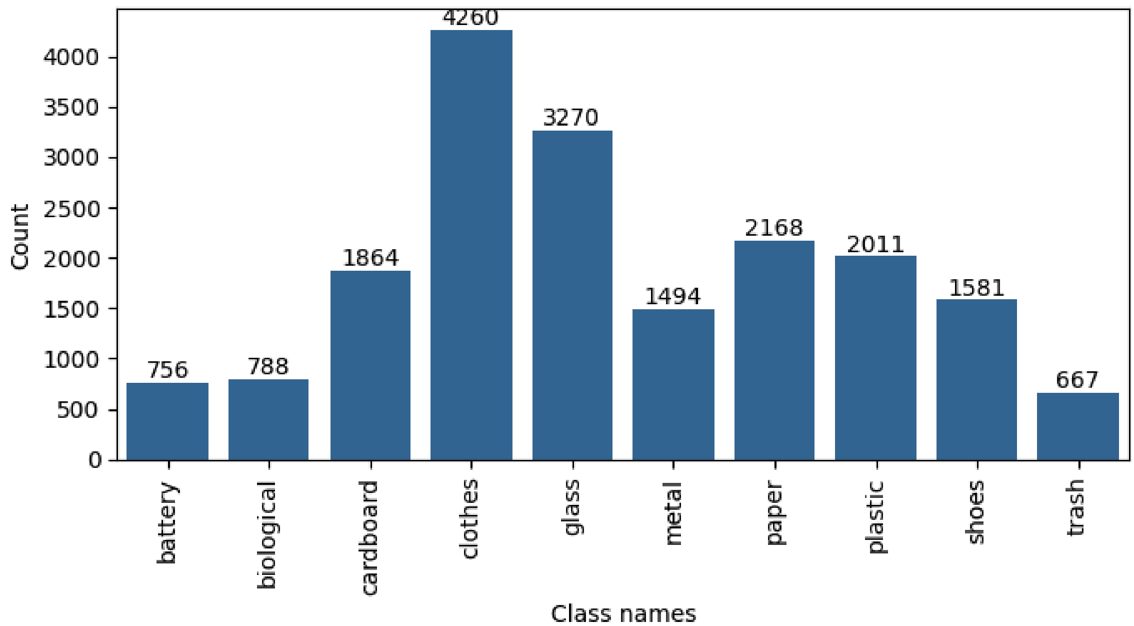

Figure 3.1 : Before Random Undersampling

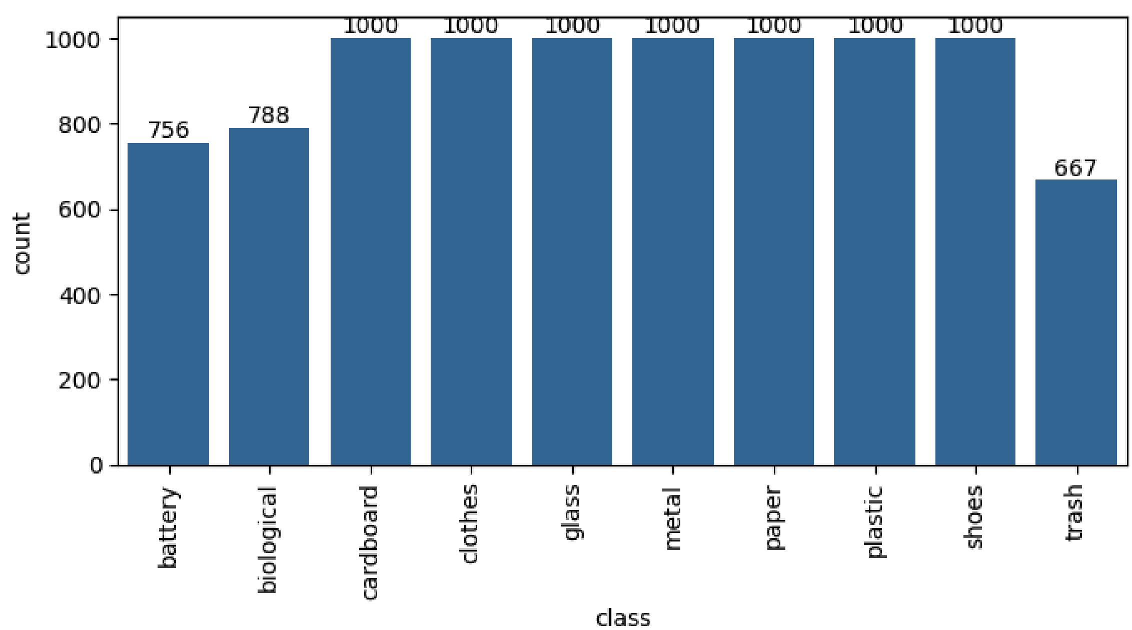

Figure 3.2 : After Random Undersampling

Outcome of Preprocessing

The preprocessing steps taken ensured that the data was clean, uniform, and augmented to foster effective learning. These steps are critical in setting up a strong foundation for building a robust machine learning model that performs well not just on the training data but also generalizes well to new, unseen data.

IV. Baseline Model

Selection and Justification:

The baseline model chosen for this project was a simple Convolutional Neural Network (CNN). This decision was driven by several factors:

- Simplicity and Speed: A straightforward CNN architecture is relatively easy to implement and quick to train. This allows for rapid iteration and experimentation, which is crucial in the early stages of a project.

- Effectiveness in Image Processing: CNNs are renowned for their performance in image classification tasks due to their ability to capture spatial hierarchies in images. Starting with a simple CNN helps in understanding how well the basic features can discriminate between different classes of waste.

- Establishing Performance Benchmarks: The primary purpose of the baseline model is to establish a performance benchmark for the project. It provides a metric of 'simplest case scenario' performance, against which all other models can be compared.

Architecture Details:

The baseline CNN model consisted of a few convolutional layers, each followed by a max pooling layer. These layers were designed to extract and down-sample the feature maps respectively, capturing the essential features of the images while reducing their dimensionality. The model concluded with a fully connected layer and a softmax activation function to classify the images into their respective waste categories.

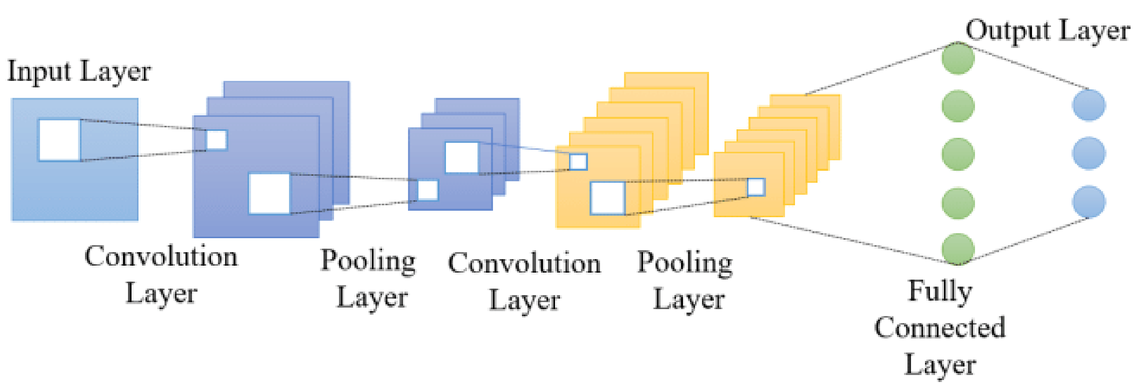

Figure 4.1 : Simple CNN Architecture

Training:

The model was trained on the preprocessed dataset, which had been augmented and balanced as previously described. The training process involved monitoring for signs of overfitting and adjusting parameters such as the learning rate and the number of epochs accordingly.

Results of Baseline Model:

The baseline CNN model achieved modest accuracy. It was effective in distinguishing between some classes that were significantly different, but struggled with classes that had similar visual features. The limitations in its performance were primarily due to its simple architecture, which was not capable of capturing more complex patterns and distinctions required for higher accuracy.

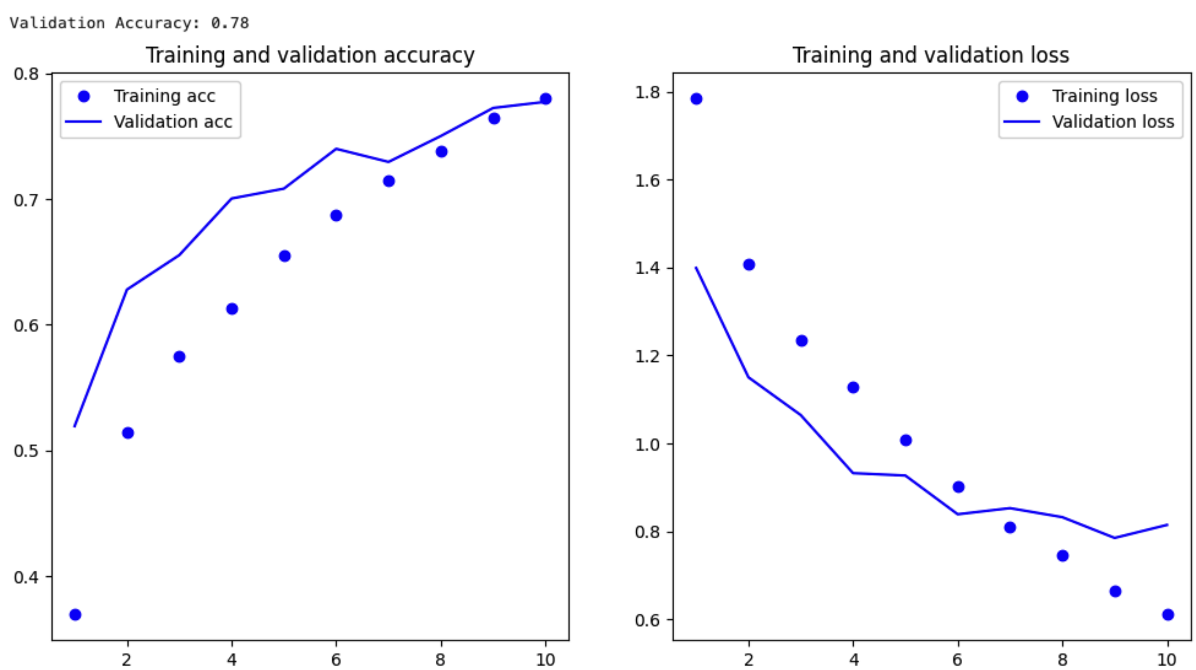

Figure 4.2 : Performance Metrics of the Baseline CNN Model

Analysis of Baseline Model Performance

Training and Validation Accuracy:

- The graph shows a steady increase in training accuracy over epochs, indicating that the model is learning effectively from the training data. Starting from about 50% accuracy, it climbs to approximately 78% by the tenth epoch.

- Validation accuracy, which is crucial for evaluating how well the model generalizes to new data, also shows an upward trend but with notable fluctuations. It starts around 55% and reaches a peak of 78%. The fluctuations suggest some variance in how the model performs on different subsets of the data, possibly due to the intrinsic differences within the validation set.

Training and Validation Loss:

- The training loss decreases significantly from around 1.8 to just below 0.8, which correlates with the increase in training accuracy. This decline in loss indicates that the model is increasingly successful in minimizing the error in its predictions as training progresses.

- Validation loss initially exhibits a steep decrease but shows fluctuations thereafter. This pattern aligns with the validation accuracy results and suggests that while the model is learning to reduce its prediction error, the variability in validation data impacts the consistency of loss reduction.

Interpretation and Implications for Model Improvement:

- Overfitting Concerns: The clear gap between training and validation accuracy, along with the less consistent reduction in validation loss, could suggest a beginning of overfitting. Although the model is improving overall, it may be learning specifics of the training data that do not generalize well to unseen data.

- Need for Model Tuning: The results underscore the potential need for further tuning of the model's hyperparameters or an increase in regularization to enhance its generalization capabilities. Techniques such as adding dropout layers or increasing the rate of existing ones, adjusting the learning rate, or employing more sophisticated data augmentation could help mitigate overfitting.

- Advanced Model Potential: Given the baseline model's performance, there is clearly room for improvement in accuracy and stability of learning. This sets the stage for experimenting with more complex architectures or methodologies, such as transfer learning with a pre-trained model like EfficientNetV2S, which could leverage deeper layers and more sophisticated features to potentially yield better and more stable results.

Need for a Better Model:

While the baseline model provided valuable insights into the basic separability of the classes in the dataset, its performance highlighted the need for a more sophisticated approach. The following factors necessitated the exploration of a better model:

- Complexity of Task: The task of classifying waste into various categories based on visual imagery is inherently complex due to the subtle differences between various types of waste materials.

- Improving Accuracy: To make the system practical and reliable for real-world applications, significantly higher accuracy was required than what the baseline model could provide.

- Handling Diverse and Complex Data: The dataset contained images with varied backgrounds, lighting conditions, and occlusions. A more advanced model with a greater capacity to learn from such a diverse dataset was needed.

Advancement to Transfer Learning:

Given these considerations, the project transitioned to using a more advanced model, EfficientNetV2S, through the application of transfer learning. This approach leveraged a pre-trained network known for its high efficiency and accuracy, which could be fine-tuned to the specific task of waste classification, potentially overcoming the limitations observed in the baseline model.

V. EfficientNetV2S Architecture

The architecture of the EfficientNetV2S model is a sophisticated deep learning framework designed for improved efficiency and effectiveness in handling diverse image processing tasks.

EfficientNetV2S is part of the EfficientNet family, which utilizes a compound scaling method that systematically scales up CNNs in a balanced way across depth (number of layers), width (number of channels), and resolution (input image size) based on a fixed set of scaling coefficients.

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	272	15
7	Conv1x1 & Pooling & FC	-	1792	1

Figure 5.1 : EfficientNetV2S Architecture

Detailed Architecture:

Stage 0:

- Operator: Conv3x3
- Stride: 2
- Channels: 24
- Layers: 1
- This is the initial convolutional layer with a 3x3 kernel size and a stride of 2. It serves as the entry point of the network where the input image is first subjected to convolution, effectively reducing its dimensionality while extracting initial features.

Stage 1:

- Operator: Fused-MBConv1, k3x3
- Stride: 1
- Channels: 24
- Layers: 2
- Uses Fused-MBConv blocks (a variant of the MBConv block that fuses convolution and depthwise convolution layers for efficiency). This stage uses 2 such layers, each with a 3x3 kernel, and operates without changing the spatial dimensions due to the stride of 1.

Stage 2 to 4:

- Operators: Range from Fused-MBConv4 to MBConv4 with different expansions and kernel sizes.
- Strides: Vary between 1 and 2 to either maintain or reduce spatial dimensions.
- Channels: Increase from 48 to 128 through these stages.
- Layers: Number of layers range from 4 to 6.
- These stages further refine features, gradually increasing the depth and capacity of the model. The use of Fused-MBConv and MBConv indicates a mix of fused layers for efficiency and regular MBConv layers for enhanced feature extraction.

Stage 5 to 6:

- Operator: MBConv6, k3x3, SEO 0.25
- Stride: Alternates between 1 and 2.
- Channels: Increase from 160 to 272.
- Layers: High number of layers (9 to 15) indicating significant complexity and depth for detailed feature extraction.
- These stages incorporate squeeze-and-excitation optimization (SEO), which helps the network focus on the most informative features by re-scaling the channel-wise feature responses.

Stage 7:

- Operator: Conv1x1 & Pooling & FC
- Stride: -
- Channels: 1792
- Layers: 1

- This final stage includes a 1x1 convolution that helps in mixing the features across channels followed by global average pooling and a fully connected (dense) layer which outputs the final predictions. This stage essentially consolidates the extracted features into predictions for the given classes.

EfficientNetV2S utilizes a complex arrangement of fused and MBConv layers across various stages, carefully increasing the network's depth and width while managing computational efficiency. The architecture is designed to handle a wide range of input resolutions effectively, making it versatile for applications requiring high accuracy and efficiency in image processing tasks such as classification.

VI. Model Training

Main Model Training with EfficientNetV2S

Model Selection and Adaptation:

EfficientNetV2S was chosen due to its optimal balance between efficiency and accuracy. The model architecture is designed to systematically scale width, depth, and resolution, which is ideal for handling the variability in the garbage classification dataset.

- Model Adaptation: The pre-trained EfficientNetV2S model, initially trained on the ImageNet dataset, was adapted for the specific task of garbage classification. This adaptation involved modifying the top layers of the network to tailor it to the ten specific waste classification categories. The top fully connected output layer of the model was replaced with a new layer that has 10 outputs, corresponding to the number of waste categories, with a softmax activation function to handle multi-class classification.

Training Process:

The training of EfficientNetV2S involved several steps designed to leverage transfer learning effectively:

1. Preprocessing and Data Augmentation: Similar to the baseline model, images were preprocessed which included resizing, normalization, and augmentation (e.g., rotation, zoom) to prepare them for efficient training. These steps help in reducing overfitting and improving the model's ability to generalize.

2. Freezing and Fine-tuning:

- Freezing Layers: Initially, most of the convolutional layers of the EfficientNetV2S were frozen, meaning their weights were not updated during training. This helps in preserving the generic features learned from the ImageNet dataset, which are useful for a wide range of image recognition tasks.

- Fine-tuning: After the initial training phase with the frozen layers, selective layers were unfrozen and the entire model underwent a fine-tuning process with a lower learning rate. This step helps the model to adapt its weights specifically to the features of the waste classification images.

3. Training and Validation: The model was trained using the training set while performance was validated using the validation set. This iterative process allowed for monitoring and tweaking parameters like learning rate and batch size to optimize performance. Special attention was paid to the validation loss and accuracy to ensure that the model was not overfitting to the training data.

4. Hyperparameter Optimization: Throughout the training process, hyperparameters were adjusted based on the performance metrics observed. This includes altering learning rates, adjusting dropout rates in dropout layers, and experimenting with different optimizers.

Evaluation:

Once the model training was complete, it was evaluated using the test set, which had not been used during the training or validation phases. The model's performance was assessed based on accuracy, precision, recall, and F1-score to understand its effectiveness in classifying various types of waste accurately.

VII. Results

Epoch	Loss (train)	Accuracy (train)	Loss (val)	Accuracy (val)	Learning Rate
1	0.4553	0.8608	0.2047	0.9358	0.001
2	0.2626	0.9153	0.1674	0.9482	0.001
3	0.2152	0.93	0.1476	0.9545	0.001
4	0.1989	0.9337	0.1409	0.9575	0.001
5	0.1745	0.9427	0.1418	0.9533	0.001
6	0.1535	0.9469	0.1399	0.9596	0.001
7	0.1486	0.9491	0.137	0.963	0.001
8	0.1329	0.9559	0.1321	0.9601	0.001
9	0.1286	0.9552	0.1288	0.9622	0.001
10	0.1205	0.9587	0.1331	0.9618	0.001
11	0.1132	0.9619	0.135	0.9618	0.0002
12	0.0905	0.969	0.125	0.966	0.0002
13	0.0867	0.9703	0.1247	0.966	0.0002
14	0.0837	0.9736	0.1252	0.966	0.0002
15	0.086	0.9711	0.123	0.9656	0.0002
16	0.0794	0.9733	0.1208	0.9664	0.0002
17	0.0764	0.9742	0.1249	0.9669	0.0002
18	0.0745	0.9751	0.1175	0.969	0.0002
19	0.0719	0.9782	0.1203	0.9673	0.0002
20	0.0674	0.9763	0.1202	0.9703	0.00004

Table 7.1 : Final Epochs Training and Validation Metrics

- Epoch 1-20 Performance:

- Accuracy and Loss Evolution: Training accuracy increased from 86.08% in Epoch 1 to 97.63% by Epoch 20, showing a steady enhancement. The corresponding validation accuracy also improved from 93.58% to 97.03% in the same interval.

- Loss Metrics: Training loss decreased from 0.4553 to 0.0674, reflecting the model's learning efficiency. Validation loss followed a downward trend from 0.2047 to 0.1202, indicative of the model's effectiveness in generalizing from training data to validation data.

- Learning Rate Adjustments: The learning rate was dynamically adjusted down to 0.00000018999890585, illustrating the use of techniques like ReduceLROnPlateau for optimizing training during later stages.

Analysis of Training and Validation Loss and Accuracy:

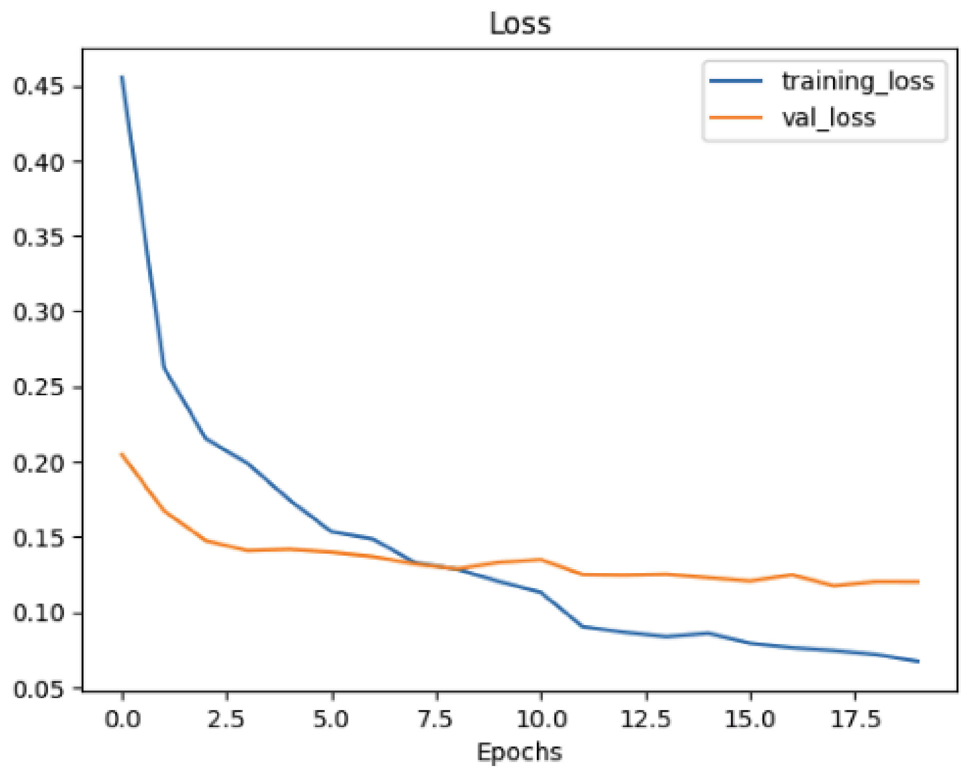

Figure 7.1 : Loss Graph

- Loss Graph Analysis:

- Training Loss: The training loss shows a significant and rapid decrease from the first epoch and continues to decline steadily as the epochs progress. Starting from above 0.4, the training loss drops to below 0.1 by the end of the observed epochs. This steep decline demonstrates the model's effective learning and optimization in minimizing the error between the predicted outputs and the actual labels.

- Validation Loss: The validation loss initially mirrors the sharp drop seen in the training loss but begins to level off after around 5 epochs. It stabilizes at approximately 0.15, showing a slight divergence from the training loss towards later epochs. This pattern suggests a good fit, although the slight elevation compared to the training loss indicates minor overfitting or variance in the validation dataset not entirely captured by the model.

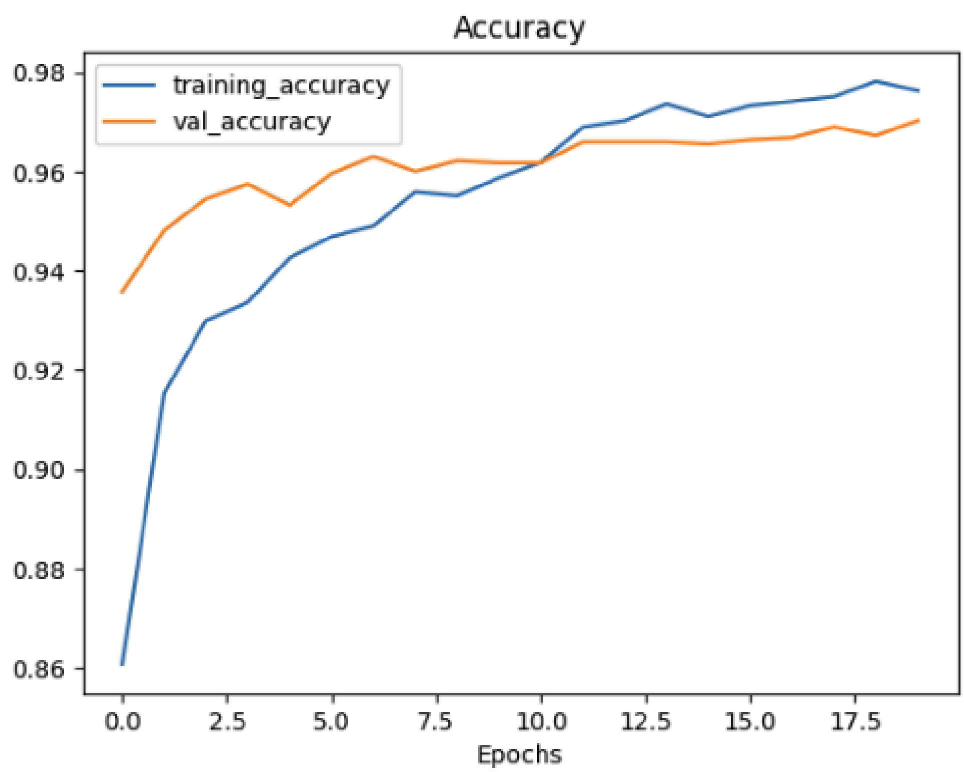

Figure 7.2 : Accuracy Graph

- Accuracy Graph Analysis:

- Training Accuracy: The training accuracy starts at around 92% and shows a gradual increase throughout the training process. It peaks slightly above 97% by the end of the training epochs. The steady ascent in training accuracy reflects the model's increasing proficiency in classifying the training data correctly.

- Validation Accuracy: The validation accuracy also begins at a high level, around 96%, and slightly fluctuates as the epochs continue. It generally follows an upward trend, albeit with less consistency compared to the training accuracy, and stabilizes just under 98%. The high validation accuracy throughout the training process indicates that the model generalizes well to new data, despite some minor fluctuations which might reflect responses to specific features or noise in the validation dataset.

Classification Report Metrics:

Classification Report

	precision	recall	f1-score	support
battery	0.99	0.94	0.96	95
biological	0.99	0.99	0.99	99
cardboard	0.98	0.97	0.98	234
clothes	1.00	0.99	0.99	533
glass	0.97	0.96	0.97	410
metal	0.93	0.99	0.96	188
paper	0.98	0.99	0.98	271
plastic	0.95	0.95	0.95	252
shoes	0.98	0.98	0.98	199
trash	0.98	0.96	0.97	84
accuracy			0.98	2365
macro avg	0.97	0.97	0.97	2365
weighted avg	0.98	0.98	0.98	2365

Figure 7.3 : Classification Report

- Precision, Recall, and F1-Score:

- High Precision: Precision scores were exceptionally high across all categories, with the highest being 1.00 for clothes and the lowest being 0.93 for metal, indicating very few false positives in these predictions.

- Recall Effectiveness: Recall scores, which measure the model's ability to find all relevant instances of a class, were also high, with the majority of categories scoring above 0.95. This denotes that the model successfully identified the correct categories almost all the time.

- Balanced F1-Scores: F1-scores, which balance precision and recall, were consistently strong, peaking at 0.99 for biological and several other categories, suggesting excellent overall classifier performance.

Confusion Matrix Analysis:

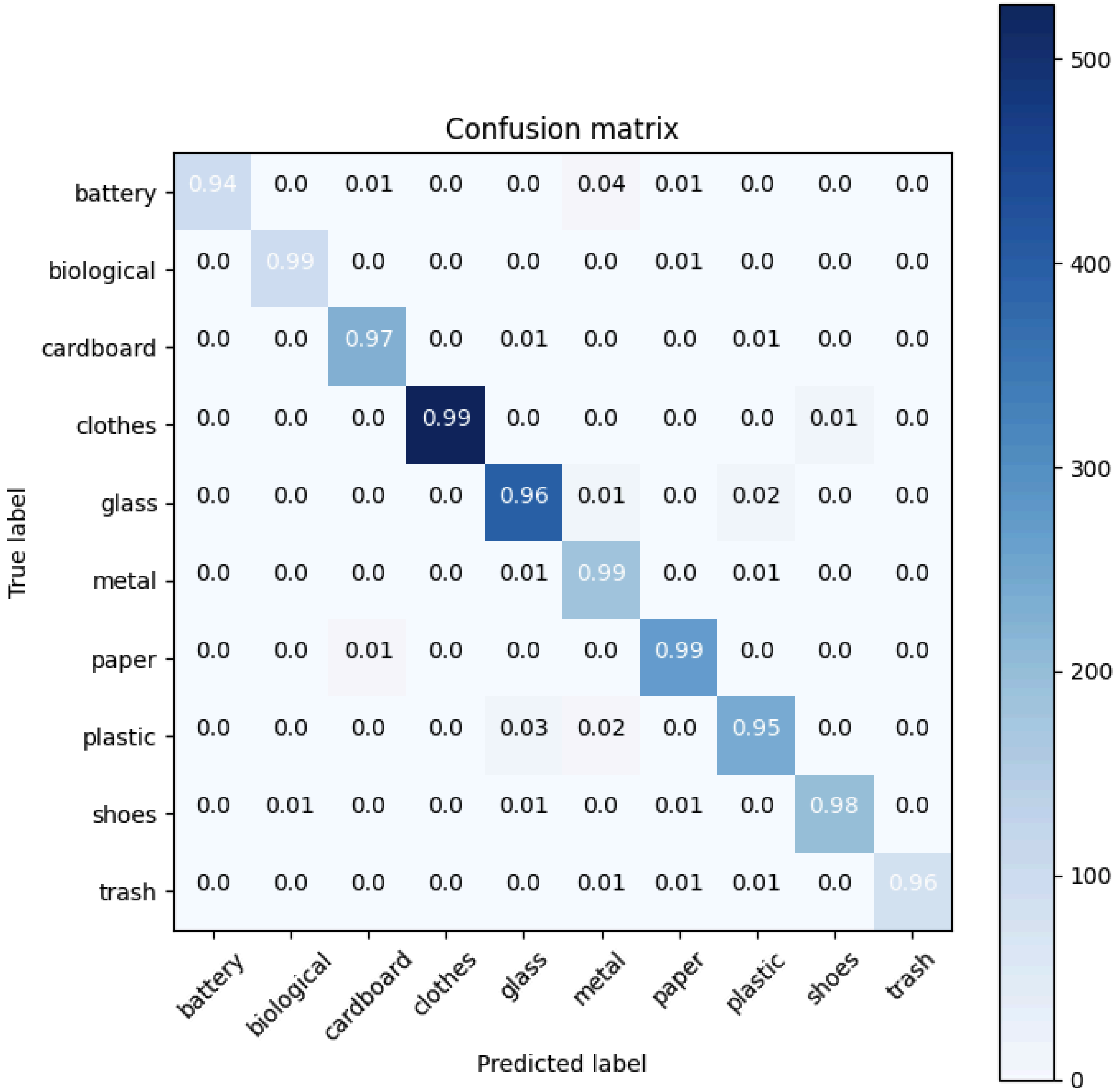

Figure 7.4 : Confusion Matrix

- The confusion matrix displayed a strong diagonal, indicating correct classifications with minimal confusion between categories. Notable observations include:

  - Biological and Paper: Nearly perfect classification with no misclassifications noted.

  - Minor Misclassifications: Some minor misclassifications were seen, such as between glass and plastic, where a few instances were confused, likely due to visual similarities in certain cases.

Practical Applications and Predictive Accuracy:

Figure 7.5 : Sample Predictions by the EfficientNetV2S Model for Various Waste Types

It is essential to demonstrate not only the model's overall accuracy and loss metrics but also its practical effectiveness in accurately classifying real-world objects.

Significance of These Predictions:

These examples serve as tangible evidence of the model's precision and reliability across different waste categories. By successfully classifying diverse items with high probabilities, the model demonstrates its potential to be integrated into automated waste management systems, where it can contribute significantly to enhancing the sorting process. The ability to identify and categorize waste with such high accuracy ensures that materials are correctly processed for recycling or appropriate disposal, optimizing both environmental and operational efficiencies.

Overall Model Performance Conclusion:

The EfficientNetV2S model showcased outstanding performance across various metrics during the last epochs of training. The high scores in precision, recall, and F1 across all categories confirm the model's ability to classify garbage types accurately and reliably. The minimal misclassification as observed in the confusion matrix further attests to the model’s robustness in dealing with diverse waste categories.

Implications for Deployment:

Given these results, the EfficientNetV2S model is well-suited for deployment in automated waste sorting systems where high accuracy and reliability are critical. The model's demonstrated ability to handle variations and subtle distinctions among waste types can significantly enhance the efficiency and effectiveness of recycling and waste management processes.

VIII. Future Work

The results from the EfficientNetV2S model showcase significant advancements in garbage classification accuracy and efficiency. However, several areas can be explored further to enhance the model's capabilities and extend its application:

1. Improving Class Discrimination: While the model performs exceptionally well across many categories, further improvements can be made in distinguishing between visually similar categories. Advanced techniques such as multi-spectral imaging or adding sensory inputs like smell or weight could enhance the model's accuracy.

2. Expanding the Dataset: Increasing the diversity and volume of the dataset to include a broader range of garbage items, particularly from different geographical locations or from varied socioeconomic backgrounds, could help in improving the robustness and adaptability of the model.

3. Real-Time Processing: Developing capabilities for real-time waste classification could significantly impact automated waste sorting systems. Integrating the model into IoT devices or embedding it within camera systems at waste processing facilities could facilitate on-the-spot sorting and recycling.

4. Deployment in Diverse Environments: Testing and deploying the model in various environments, such as residential areas, commercial complexes, and industrial sites, will help understand its performance dynamics across different settings and identify any additional adaptations needed.

5. Model Optimization for Low-Resource Devices: Optimizing the model to run efficiently on devices with lower computational power could broaden its applicability, allowing its deployment in regions with limited technological infrastructure.

IX. Conclusion

This project successfully implemented the EfficientNetV2S model for the classification of waste into various categories, achieving high accuracy and demonstrating the model’s potential in enhancing waste management systems. The model's ability to learn from a diverse set of image data and generalize to new, unseen images was clearly evidenced by its performance metrics, which consistently showed high precision, recall, and F1-scores.

The practical applications of this model are vast, extending from automated waste sorting in recycling facilities to integration into smart city waste management systems, potentially leading to more efficient recycling processes and better environmental conservation practices.

In conclusion, while the current achievements are significant, the potential for future enhancements and broader applications of the model promises further advancements in sustainable waste management. By continuing to refine and adapt this technology, it is possible to make substantial impacts on environmental health and resource conservation.

References

1. Mridha, K., Shaw, R.N., Ghosh, A. (2021). Intelligent-based waste management awareness developed by transfer learning. 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), 1–5. Available at: https://doi.org/10.1109/GUCON50781.2021.9573586

2. Iqbal, S., Naz, T., Naseem, M. (2021). Challenges and opportunities linked with waste management under a global perspective: a mini review. Journal of Quality Assurance in Agricultural Sciences, 1(9–13). Available at: https://doi.org/10.52862/jqaas.2021.1.1.2

3. Lin, K., Zhao, Y., Gao, X., et al. (2022). Applying a deep residual network coupling with transfer learning for recyclable waste sorting. Environmental Science and Pollution Research, 29(91081–91095). Available at: https://doi.org/10.1007/s11356-022-22167-w

4. Liu, W., Ouyang, H., Liu, Q., et al. (2022). Image recognition for garbage classification based on transfer learning and model fusion. Mathematical Problems in Engineering, 2022. Available at: https://doi.org/10.1155/2022/4793555

5. Rashida, J., Hamzah, R., Samah, K.A.F.A., et al. (2022). Implementation of faster region-based convolutional neural networks for waste type classification. 2022 International Conference on Computer and Drone Applications (IConDA), 125–130. Available at: https://doi.org/10.1109/ICONDA56696.2022.10000369

6. Nafiz, M.S., Das, S.S., Morol, M.K., et al. (2023). Convowaste: An automatic waste segregation machine using deep learning. 2023 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 181–186. Available at: https://doi.org/10.1109/ICREST57604.2023.10070078

7. Cheema, S.M., Hannan, A., Pires, I.M. (2022). Smart waste management and classification systems using cutting-edge approaches. Sustainability, 14(10226). Available at: https://doi.org/10.3390/su141610226

8. Shreya, M., Katyal, J., Ramesh, R., et al. (2022). Technical solutions for waste classification and management: A mini-review. Waste Management Research, 41(801). Available at: https://doi.org/10.1177/0734242X221135262

9. Narayan, Y. (2021). DeepWaste: Applying deep learning to waste classification for a sustainable planet. arXiv Preprint. Available at: https://arxiv.org/abs/2102.07307

10. Feng, J., Tang, X. (2020). Office garbage intelligent classification based on inception-v3 transfer learning model. Journal of Physics: Conference Series, 1487(012008). Available at: https://doi.org/10.1088/1742-6596/1487/1/012008

11. Yong, L., Ma, L., Sun, D., et al. (2023). Application of MobileNetV2 to waste classification. PLoS One, 18(e0282336). Available at: https://doi.org/10.1371/journal.pone.0282336

12. Girsang, A.S., Yunanda, R., Syahputra, M.E., et al. (2022). Convolutional neural network using res-net for organic and inorganic waste classification. 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), 1–6. Available at: https://doi.org/10.1109/ICOSNIKOM56551.2022.10034869

13. Nurahmadan, I.F., Arjuna, R.M., Prasetyo, H.D., et al. (2021). A mobile-based waste classification using mobilenets-v1 architecture. 2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), 279–284. Available at: https://doi.org/10.1109/ICIMCIS53775.2021.9699161

14. Rismiyati, Endah, S.N., Khadijah, et al. (2020). Xception architecture transfer learning for garbage classification. 2020 4th International Conference on Informatics and Computational Sciences (ICICoS), 1–4. Available at: https://doi.org/10.1109/ICICoS51170.2020.9299017

15. Yang, M., Thung, G. (2017). Trashnet. GitHub. Available from: https://github.com/garythung/trashnet

16. Zhou, Y., Wang, X., Zhang, M., et al. (2019). Mpce: a maximum probability-based cross entropy loss function for neural network classification. IEEE Access, 7: 146331–146341. Available at: https://doi.org/10.1109/ACCESS.2019.2946264

17. Li, J., Gao, M., D'Agostino, R. (2019). Evaluating classification accuracy for modern learning approaches. Statistical Medicine, 38: 2477–2503. Available at: https://doi.org/10.1002/sim.8103