Optimizing Waste Management Systems through Image Recognition and Feature Extraction Techniques Integrating DQN-Based Models

Siyuan Liang*

London College of Fashion, University of the Arts London, London, United Kingdom

Abstract. With increasing global attention on environmental protection and carbon neutrality goals, waste management has become a crucial component of achieving sustainability. Traditional waste disposal methods, such as manual sorting, identification, and crushing, often suffer from low efficiency and inconsistency. This study proposes a waste management system that integrates Artificial Intelligence (AI) to enhance the accuracy and efficiency of waste treatment through automation. The proposed system addresses waste classification, recognition, and fragmentation, utilizing advanced AI technologies like Graph Neural Networks (GNN), Convolutional Neural Networks (CNN), Visual Transformers (ViT), and Deep Reinforcement Learning (DRL). To optimize the waste management process, the study used the KTH-TIPS dataset, which contains 800 high-resolution images of various material types. Robust model training was ensured through a series of preprocessing and data augmentation techniques. The experimental results demonstrated that CNN and Transformer architectures, including MobileNet, ResNet, and ViT, achieved high accuracies of 100.00%, 100.00%, and 96.91%, respectively. While the GNN and VGG architectures had slightly lower accuracies of 82.88% and 84.57%, they still demonstrated competitive performance. The experimental results illustrate the variation in training and testing losses across different models over the training cycles, revealing the learning dynamics and efficiency of each model.

1 Introduction

The growing global emphasis on environmental protection and the urgent need to achieve carbon neutrality have placed waste management at the forefront of sustainability efforts. Traditional waste recycling methods, such as manual sorting, identification, and fragmentation, are often time-consuming and inconsistent. As a result, less reusable material is recovered, and more waste ends up in landfills. Artificial Intelligence (AI) offers the potential to make these processes more accurate and efficient. Waste management typically involves three stages: sorting, identification, and fragmentation. Sorting is usually done by type, either manually or with basic machinery. Identification uses sensors or optical scanners to assess the type and quality of materials. Finally, fragmentation reduces waste into smaller

* Corresponding author: s.liang0620211@arts.ac.uk

pieces for recycling, composting, or disposal. Each of these stages presents unique challenges, which can be mitigated through the application of advanced AI techniques.

Recent advancements in AI have introduced new tools like Convolutional Neural Networks (CNNs), which are highly effective in image classification by extracting features from images. Vision Transformers (ViTs) use self-attention mechanisms to analyze images, while Graph Neural Networks (GNNs), particularly Graph Convolutional Networks (GCNs), are useful for understanding and processing relationships between data points, such as in complex waste sorting systems.

Significant research has focused on applying machine learning (ML) and deep learning (DL) models to waste classification, particularly image-based waste identification. For instance, Guanhao Yang and colleagues used the YOLOv5 model to identify garbage images, achieving an average recognition accuracy of 94.5%. They also used a robotic arm for waste classification [1]. Similarly, Ziying Huang applied the CNN MobileNetV2 model to classify 14,051 waste images into four categories: recyclable, food, hazardous, and other waste, achieving 85.7% accuracy on the verification set [2]. Aidan Kurz, Ethan Adams, and others combined ViT and CNN architectures to process multiple Transformer blocks in parallel, achieving a 94.27% accuracy on their test set [3]. These studies illustrate the growing use of deep learning models for image recognition in waste sorting and identification. While most research has focused on improving the accuracy of waste classification, there has been less emphasis on simplifying the entire waste management process using data science techniques. Jagdeep Singh, in a comprehensive review, highlighted the lack of systematic, data-driven strategies for managing the entire waste lifecycle. Singh noted that while AI has been successfully applied to waste classification, its potential to streamline the full recycling process, including fragmentation, remains underexplored [4]. Researchers agree that integrating AI at all stages of waste management, including sorting, classification, fragmentation, and further processing, is crucial for improving efficiency. A key challenge is that many studies limit their scope to classification, neglecting downstream processes like fragmentation, which prepares waste for specialized recycling treatments.

To address these gaps, this paper proposes an AI-powered waste management system that integrates identification, sorting, and fragmentation. The system leverages AI techniques such as ViTs, CNNs, and GNNs to enhance waste material identification. Following feature extraction, Deep Reinforcement Learning (DRL) is used to simulate and optimize the waste cutting process, solving the "cutting box" problem to ensure maximum efficiency. This research offers a comprehensive approach to automating the waste management workflow, significantly improving recycling rates and reducing environmental impact.

2 Method

2.1 Dataset preparation

In this study, deep learning and reinforcement learning techniques are employed to optimize material cutting strategies. The dataset used in this experiment is the KTH-TIPS dataset [5], which consists of surface texture images from 10 different types of materials, such as sandpaper and aluminum foil. These images were captured under various lighting conditions, viewing angles, and scales to reflect real-world variations in texture. The dataset contains 800 high-resolution images, each of which was resized to 128×128 pixels and converted to grayscale to simplify the classification process while preserving critical texture information. There are 10 distinct categories based on material types shown in Fig. 1.

Fig. 1. The sample images in the collected dataset.

To ensure the robustness of the deep learning models and prevent overfitting, several preprocessing steps were applied to the dataset. All pixel values were normalized to a range between 0 and 1, which helps facilitate faster convergence of the model during training. Data augmentation techniques, including random rotations, flips, and zooms, were also employed to artificially increase the diversity of the training set and make the model more robust to real-world variations in texture. After augmentation, all images were resized to 128x128 pixels for compatibility with the GNN-based model architecture. This setup provides a balanced and preprocessed dataset for texture classification and cutting optimization tasks.

2.2 Prediction based on deep learning models

2.2.1 Framework of the model

The feature extraction process in this experiment is divided into two main phases shown in Fig. 2.

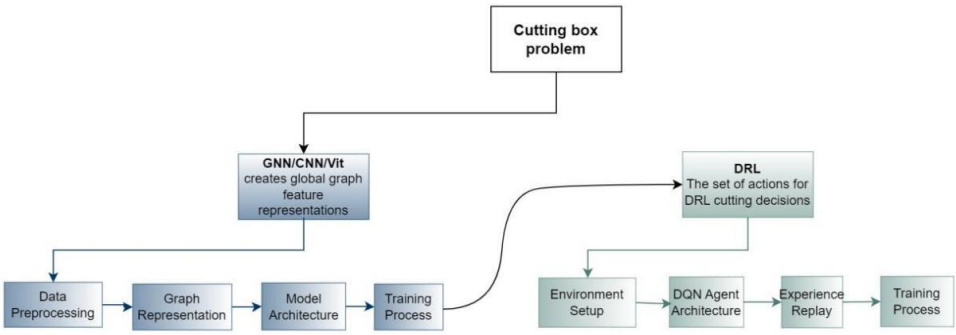

Fig. 2. Model Framework

The first phase of the experiment involves extracting texture features from images using various deep learning architectures, including Graph Neural Networks (GNN), Convolutional Neural Networks (CNN), and Vision Transformers (ViT). In the second phase, material cutting strategies are optimized using a Deep Q-Network (DQN). A DQN is an artificial agent that combines reinforcement learning with a deep neural network to learn successful policies directly from high-dimensional sensory inputs through end-to-end reinforcement learning. Deep learning models are employed for texture feature extraction due to their ability to represent image data in a graph format. In GNNs, each node corresponds to a section of the

image, and edges represent the relationships between these sections. The GNN model in this experiment utilizes a Graph Convolutional Network (GCN) with three layers to aggregate features at multiple levels. To capture both low-level and high-level texture information, a Jumping Knowledge Mechanism (JKM) is applied. This mechanism allows each node to flexibly leverage different neighborhood ranges, enabling better structure-aware representation [6]. The features from different layers are then combined, and the final feature representation is passed through a global mean pooling layer. A fully connected layer is used to predict the material class.

2.2.2 Feature extraction

Alternatively, Convolutional Neural Networks (CNNs) are used for feature extraction in other phases of the experiment, specifically employing the MobileNet, ResNet, and VGG16 architectures.

MobileNet is a lightweight CNN architecture designed to reduce the number of parameters and computational cost. It achieves this by using depthwise separable convolutions, which significantly lower model complexity while maintaining performance. In this experiment, MobileNet was used for tasks that required fast feature extraction, efficiently capturing material textures with lower computational costs.

ResNet, on the other hand, addresses the degradation problem often encountered in deep networks through the use of residual connections. In this study, ResNet18 is used, which consists of four main residual blocks. Each block includes multiple 3x3 convolutional layers, with the final output passing through a global average pooling layer, producing a 512-dimensional feature vector.

VGG16 employs five convolutional blocks, each followed by a MaxPooling layer. These blocks allow the model to capture increasingly abstract representations of material textures. The final output is a 4096-dimensional feature vector, which is then used for the material classification task.

More recently, the Vision Transformer (ViT) has gained popularity for image feature extraction tasks. ViT splits an image into fixed-size patches, linearly embeds each patch, adds positional embeddings, and feeds the resulting sequence of vectors into a standard Transformer encoder [7]. This architecture allows the model to capture long-range dependencies and global patterns in texture images, which are crucial for accurately identifying materials. ViT divides each image into 16x16 patches, flattens them, and projects them into an embedding space. A class token is prepended to the sequence of patches, summarizing the overall representation of the image. The Transformer encoder in ViT consists of multiple layers, each containing a self-attention mechanism and a feed-forward neural network. The self-attention mechanism is essential for modeling global relationships within the image, while layer normalization and residual connections help stabilize the training process.

2.2.3 Reinforcement learning stage

Once the feature extraction process is complete, the extracted features are passed to a Deep Q-Network (DQN), which is responsible for optimizing material cutting decisions. The DQN agent takes the feature vectors generated by the GNN, CNN, or ViT models as input and calculates Q-values for various cutting actions, such as rotating the material, adjusting the cutting position, and determining the cutting direction. The DQN agent interacts with a simulated environment that models the cutting process, receiving rewards based on the efficiency of each action. The reward function is designed to maximize material utilization and minimize waste. To improve learning efficiency, the system employs an Experience

Replay Mechanism, where past interactions are stored in a buffer and replayed during training. This mechanism allows the DQN agent to learn more effectively by using past experiences to optimize its decision-making process. The whole process can be found in Fig. 3.

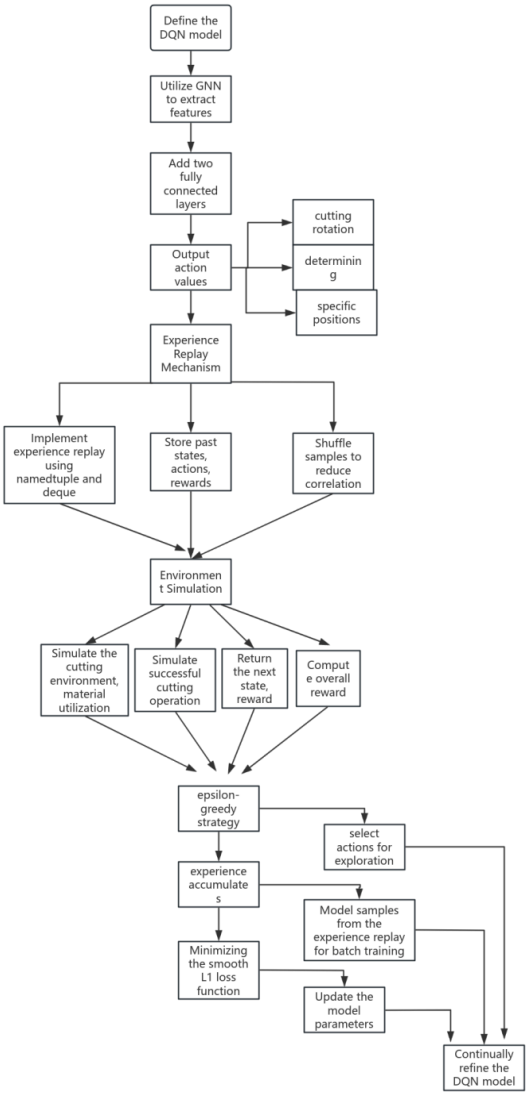

Fig. 3. Workflow of Cutting Environment Optimization Based on DQN Model

2.2.4 Implementation details

In this experiment's reinforcement learning process, the Deep Q-Network (DQN) is optimized using an epsilon-greedy exploration strategy. At the start of training, the agent explores various cutting actions randomly, but as training progresses, it gradually shifts towards exploiting the learned policy. The agent is trained over 1,000 episodes, during which it refines its cutting strategy based on feedback from the simulated environment. A discount factor of 0.99 is applied to future rewards, allowing the agent to balance immediate rewards with long-term goals. Throughout the experiment, the AdamW optimizer is used, as Zhuang

describes it as an approximation of a proximal gradient method that leverages the closed-form proximal mapping of the regularizer [8]. This optimizer is applied to update the parameters of the Vision Transformer (ViT), Graph Neural Network (GNN), Convolutional Neural Network (CNN), and DQN models. The learning rate is set at 0.0005 to ensure stable convergence, and a weight decay is applied to prevent overfitting. The DQN agent is trained using smooth L1 loss (Huber loss), which is less sensitive to outliers and provides a stable learning signal during reinforcement learning. A batch size of 32 is used to optimize computational resources while maintaining stable training dynamics.

3 Results and discussion

3.1 Experimental results

At the end of the training process, the performance of each deep learning vision model integrated with the DQN model was evaluated based on accuracy. In the first stage, the GCN model achieved an accuracy of 88.38%. In the second stage, after the model converged, the DQN model based on the MobileNet architecture achieved an accuracy of 82.88%. The loss graph for the GCN architecture in the first stage, as well as after integrating the DQN model, is shown in Fig. 4:

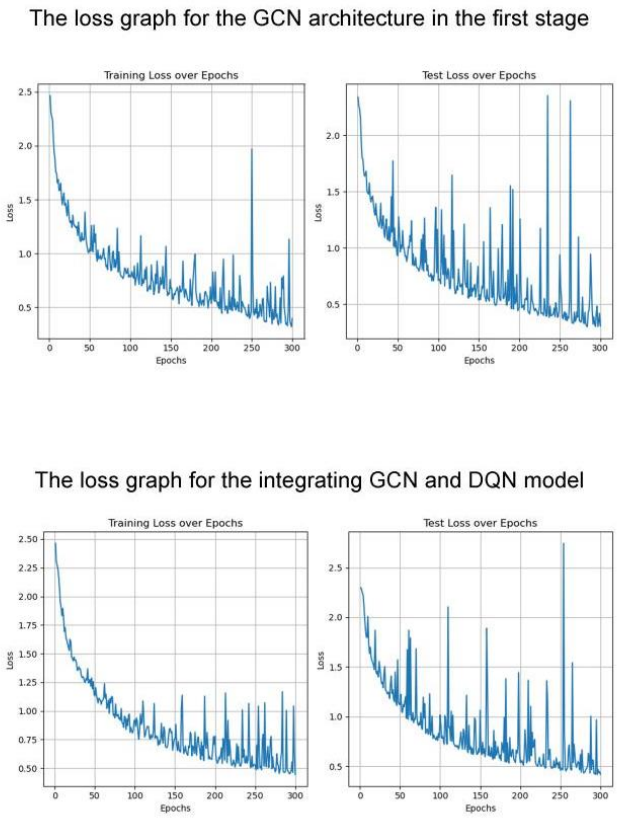

Fig. 4. Changes in Training and Testing Losses with Epochs of different models.

In the first stage, the MobileNet model achieved an accuracy of 99.38%. In the second stage of the experiment, after the model converged, the DQN model based on the MobileNet architecture achieved a perfect accuracy of 100.00%. Similarly, in the first stage, the ResNet model achieved an accuracy of 98.77%. After the model converged in the second stage, the DQN model based on the ResNet architecture also achieved an accuracy of 100.00%. For the VGG model, the first stage yielded an accuracy of 90.12%. However, in the second stage, after convergence, the DQN model based on the VGG architecture achieved an accuracy of 84.57%. This highlights the complexity of the VGG model in texture classification tasks and its limitations in feature extraction. Finally, the Vision Transformer model achieved an accuracy of 97.53% in the first stage. After convergence in the second stage, the DQN model based on the Vision Transformer architecture reached an accuracy of 96.91%. The loss graphs for the MobileNet, ResNet, VGG, and Vision Transformer architectures, both in the first stage and after integrating the DQN model, are shown below: ABCDEFGH. These results indicate that the MobileNet and ResNet architectures performed exceptionally well on this dataset, while the Vision Transformer also delivered competitive results.

Table 1. Comparison of Accuracy when model convergence.

Model	**Accuracy**
GNN+DRL	82.88%
MobileNet+DRL	100%
ResNet+DRL	100%
VGG+DRL	84.57%
ViT+DRL	96.91%

Table 1 shows that CNN architecture models perform more consistently in classification tasks, while GNN and ViT also demonstrate potential in handling more complex tasks with different architectural approaches.

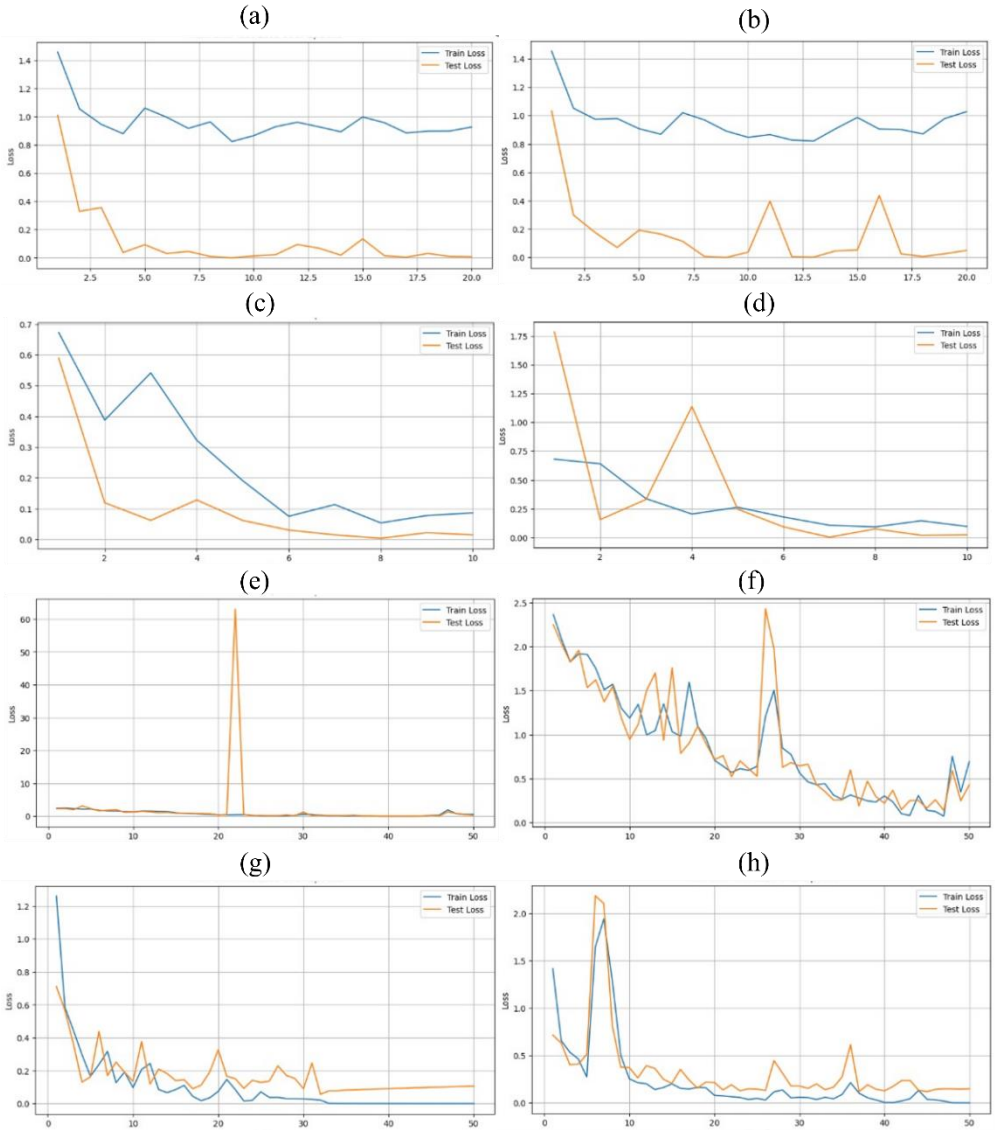

Fig. 5. Models Changes in Training and Testing Losses with Epochs for (a) MobileNet architecture (b) MobileNet architecture integrate DRL model (c) ResNet architecture (d) ResNet architecture integrate DRL model (e) VGG architecture (f) VGG architecture integrate DRL model (g) Vit architecture (h) Vit architecture integrate DRL model

As shown in Fig. 5, the MobileNet model exhibits a gradual decrease in loss from the early stages of training and stabilizes in the later stages. This indicates that the model quickly learns and adapts to the training data, ultimately achieving very low levels of loss, demonstrating its efficiency and superior generalization ability on this dataset. The ResNet model performs similarly to MobileNet, with a steady decrease in loss throughout training. This stable downward trend reflects how ResNet effectively mitigates the gradient vanishing problem through its residual connections, maintaining efficient learning in deeper network structures. The VGG model, though showing slight fluctuations in its loss curve during training, still converges well overall. These fluctuations may stem from the complexity of the

model in capturing more intricate features. However, the VGG model ultimately demonstrates effective learning. In contrast, the GNN model shows a slower reduction in loss compared to the CNN-based architectures. This reveals the challenges graph neural networks face in handling texture classification tasks, likely due to the complexity of processing graph-structured data. Nonetheless, the GNN model eventually reduces loss, reflecting its potential for adapting to graph-based data. The ViT model shows rapid loss reduction and stabilizes in the early stages of training. This suggests that ViT can effectively capture global image features, and its self-attention mechanism provides a significant advantage in image classification tasks. To better visualize the performance changes during model training, the training and testing losses of MobileNet, ResNet, VGG, GCN, and ViT models are plotted as a function of training epochs. The MobileNet and ResNet models exhibit rapid loss reduction and quick stabilization, while the VGG model, despite fluctuations, ultimately converges well. The GCN model demonstrates slower loss reduction, reflecting the task's complexity, while the ViT model shows a faster loss reduction rate and stabilizes early in training. The figure presents the training and testing loss curves for multiple models, providing an intuitive comparison of the performance of MobileNet, ResNet, VGG, GCN, and ViT during training. This comparison reveals the unique dynamics and efficiency of each model in processing complex data. The study further highlights that ViT models can perform better with smaller datasets. This is because the attention mechanism allows the model to gather more information from different image patches [9, 10].

3.2 Discussion on feature extraction and reinforcement learning

3.2.1 Feature extraction

Feature extraction plays a crucial role in the performance of deep learning models, particularly in image recognition and classification tasks. This study employed three different deep learning architectures to process waste image data: Convolutional Neural Network (CNN), Graph Neural Network (GNN), and Vision Transformer (ViT). Each architecture has its unique advantages and specific use cases. GNN is designed to process graph-structured data, extracting features by considering the relationships between nodes. Although GNN underperformed compared to CNN in this study's image classification tasks, its strength lies in handling structured data, which cannot be overlooked. The lower performance of GNN may be due to the complexity of image data, indicating that specific adjustments or optimizations are necessary for improving GNN’s effectiveness in image feature extraction tasks. CNN excels at capturing local features in images through its convolutional layers, which are essential for image recognition. In this study, both MobileNet and ResNet models successfully extracted key texture and pattern information from waste images using deep convolutional layers, which proved critical for accurate classification. CNN's ability to perform well in image classification tasks, especially when dealing with waste images that have complex textures and details, makes it a strong choice for such tasks. ViT is a novel architecture that segments images into small patches and captures global features through self-attention mechanisms. This approach allows ViT to handle long-distance dependencies in images, surpassing traditional CNNs in certain tasks. In this study, ViT demonstrated strong potential for processing waste images. Although its performance was slightly lower than that of CNN, its ability to extract global features presents a promising direction for future research.

3.2.2 Reinforcement learning

Reinforcement learning was used in this study to optimize the cutting strategy for waste materials. Through interaction with a simulated environment, the Deep Q-Network (DQN) model learned to maximize material utilization and minimize waste. The DQN model optimizes the cutting strategy by exploring different actions and learning from reward feedback. This reward-based learning mechanism enables the DQN to gradually improve its decisions and adapt to constantly changing environmental conditions. The performance of reinforcement learning is influenced by several factors, including the design of reward functions, the balance between exploration and exploitation, and the setting of learning rates. In this study, it can be found that appropriate exploration strategies and timely reward feedback were crucial for the successful learning of the DQN model.

4 Conclusion

This study successfully developed and implemented an integrated, artificial intelligence-driven waste management system that efficiently extracts features by combining CNN, GNN, and ViT, and optimizes material cutting strategies using DRL. The experimental results indicate that the proposed system has significant potential and effectiveness in automating waste classification and treatment. In terms of feature extraction, CNN architectures, particularly MobileNet and ResNet, excel due to their superior image processing capabilities. While ViT, though slightly less effective in some cases, offers promising directions for future research with its ability to capture global features. GNN shows natural advantages in processing graph-structured data, but further optimization is required to improve its performance in image classification tasks, as demonstrated in this study. For reinforcement learning, the DQN model effectively learned optimal cutting strategies by interacting with the simulated environment, demonstrating the practicality of reinforcement learning in automated decision support systems. Through a carefully designed reward mechanism, the model was able to learn how to maximize material utilization and minimize waste. Although this study achieved positive results in improving waste management efficiency, there are still limitations and challenges. Issues such as the scale and diversity of datasets, the computational cost of models, and the complexity of real-world application scenarios need further exploration and resolution. Future research should focus on expanding dataset scope, optimizing model architectures, and deploying the system across a wider range of practical applications.

References

1. G. Yang, et al., Garbage Classification System with YOLOV5 Based on Image Recognition, in 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP). Piscataway, New Jersey: IEEE, 17 – 17 (2021)
2. Z. Huang. Garbage classification based on convolutional neural network. In AIP Conference Proceedings (Vol. 3017, No. 1). AIP Publishing (2023)
3. A. Kurz, et al., WMC-ViT: Waste Multi-class Classification Using a Modified Vision Transformer, in 2022 IEEE MetroCon. Piscataway, New Jersey: IEEE, 1 – 3 (2022)
4. J. Singh, et al., Progress and challenges to the Global Waste Management System. Waste Manage. Res. 32, 800 – 812 (2014)

5. E. Hayman, et al., THE KTH-TIPS database, The KTH-TIPS and KTH-TIPS2 image databases. Available at: https://www.csc.kth.se/cvap/databases/kth-tips/index.html (Accessed: 11 September 2024)
6. K. Xu, et al. Representation learning on graphs with jumping knowledge networks. In International conference on machine learning (pp. 5453-5462). PMLR (2018)
7. A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
8. Z. Zhuang, M. Liu, A. Cutkosky, F. Orabona, Understanding AdamW through Proximal Methods and Scale-Freeness. arXiv preprint, arXiv:2202.00089 (2022).
9. J. Maurício, J et al. Comparing vision transformers and convolutional neural networks for image classification: A literature review. Applied Sciences, 13(9), 5521 (2023).
10. S. Khan, et al. Transformers in vision: A survey. ACM computing surveys (CSUR), 54(10s), 1-41 (2022).