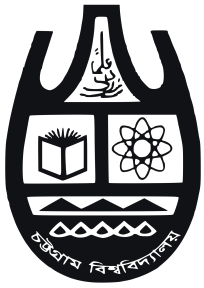
A study on Image Classification based on Deep Learning and PyTorch



Shahnewaz Khandaker

Student ID: 15701028

Session: 2014-2015

Supervisor: Prof. Dr. Rashed Mustafa

Department of Computer Science and Engineering

UNIVERSITY OF CHITTAGONG

This dissertation is submitted for the degree of

Bachelor of Science in Engineering (B.Sc. Engg.) in Computer Science and Engineering

CSECU August 2025

Report Code:

University of Chittagong

Department of Computer Science and Engineering

8-th Semester B.Sc. Engineering Examination 2023

Course No.: CSE 800

Title: A study on Image Classification based on Deep Learning and PyTorch

Report Code:

University of Chittagong

Department of Computer Science and Engineering

8-th Semester B.Sc. Engineering Examination 2023

Course No.: CSE 800

Student Name: Shahnewaz Khandaker
Student ID: 15701028
Session: 2014-2015
Hall: Shaheed Abdur Rab

Signature of Student:

Submission Date: 25 August 2025

Supervisor Approval Page

Title of the Thesis/Project: A study on Image Classification based on Deep Learning and PyTorch

Document Type: Bachelor of Science in Engineering Thesis/Project Plan

Degree Program: Bachelor of Science in Engineering in Computer Science and Engineering

Institution: Department of Computer Science and Engineering, University of Chittagong

Data of Submission: 25 August 2025

Table 1 Evaluation Criteria (to be filled by supervisor(s))

Evaluation Criteria	**Select an Option**		
Maintained Regular Communication ?	◯ *Yes*	◯ *No*	◯ *Partly*
Maintained Professionalism ?	◯ *Yes*	◯ *No*	◯ *Partly*
Addressed the given comments ?	◯ *Yes*	◯ *No*	◯ *Partly*
Checked by Plagiarism and AI content checker ?	◯ *Yes*	◯ *No*	◯ *Partly*
Report	◯ *Satisfactory*	◯ *Not Satisfactory*	◯ *Partly*
Approved	◯ *Yes*	◯ *No*	

This B.Sc.Engg. Thesis/Project Plan has been reviewed and (NOT) approved by __-
__ __-
considering the evaluation criteria outlined in Table 1

Signature
Prof. Dr. Rashed Mustafa
Department of Computer Science and Engineering
University of Chittagong

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Shahnewaz Khandaker
Student ID: 15701028
Session: 2014-2015
Department of Computer Science and Engineering
University of Chittagong
Chattogram-4331, Bangladesh.
Email: email@cu.ac.bd

I would like to dedicate this thesis to my parents and family, whose unwavering support and encouragement have been invaluable throughout my academic journey.

Acknowledgements

First and foremost, I would like to acknowledge I express my deepest gratitude to Almighty Allah for His guidance and blessings throughout my academic journey.

I am sincerely grateful to my supervisor, Dr. Rashed Mustafa, Professor, Department of Computer Science and Engineering, University of Chittagong, for his continuous support, excellent mentorship, and insightful critiques throughout my thesis work, which provided me the freedom to explore my research independently.

I would also like to thank Prof. Dr. Sanaullah Chowdhury, Chairman, Department of Computer Science and Engineering, University of Chittagong, for providing me the opportunity to pursue and complete my undergraduate degree.

My heartfelt appreciation goes to Dr. Abu Nowshed Chy, Assistant Professor, Department of Computer Science and Engineering, University of Chittagong, for his guidance and unwavering support during my academic journey.

Finally, I extend my thanks to all my teachers for their dedication and contributions, which have left a lasting impact on my academic development.

Abstract

Managing waste effectively is one of the biggest challenges for protecting our environment. Recycling and proper sorting of waste are essential for reducing landfill use and supporting sustainability, but current computer systems that automatically classify waste often struggle when applied in real-world conditions. Many of these systems are trained on clean and simple datasets, which do not reflect the messy and mixed nature of waste found in everyday landfill sites.

This thesis develops a new artificial intelligence model that uses an "attention mechanism" to focus on the most important parts of an image when identifying different types of waste. The model was tested on a dataset built from real landfill images, making it more realistic and practical. To make the system transparent, we also used visual tools that show which parts of each image the model is using to make its decisions.

Comparative experiments against established architectures such as VGG16, ResNet, DenseNet, Inception, and MobileNet demonstrate improved performance and robustness. The findings show that attention mechanisms significantly strengthen waste classification, contributing to more sustainable management systems by supporting recycling, reducing landfill dependency, and advancing environmental goals.

Table of contents

Chapter 1

Introduction

1.1 Background

Waste management has become a pressing global challenge, directly influencing environmental sustainability, public health, and economic development. With rapid urbanisation and increasing consumerism, the volume of waste generated worldwide has grown at an unprecedented rate. Traditional waste management strategies, often reliant on manual sorting or rule-based systems, are inefficient, labour-intensive, and prone to human error. As a result, misclassification of waste materials contributes to resource loss, higher landfill dependency, and increased environmental pollution.

Recent advances in artificial intelligence, particularly computer vision, have opened new possibilities for automating waste classification. Convolutional Neural Networks (CNNs) have demonstrated strong performance in image recognition tasks, making them a natural candidate for this domain. However, many existing automated systems are trained on synthetic or pristine datasets that fail to capture the complexity of real-world waste streams, where items are often contaminated, damaged, or visually ambiguous. This gap reduces the generalisability of such models when deployed in landfill or recycling environments, limiting their practical impact.

Another limitation of conventional CNN-based approaches is their difficulty in focusing on the most informative features of an image, especially when waste items share similar textures, shapes, or colours. This often leads to misclassification in mixed or overlapping categories, posing challenges for reliable real-world deployment. Furthermore, many models function as "black boxes," providing little interpretability for stakeholders who require transparent decision-making in critical environmental applications.

To address these challenges, this thesis investigates an enhanced waste classification framework built on CNNs augmented with an attention mechanism. The attention module is designed to guide the network towards the most discriminative regions of an image, improving classification accuracy on complex and noisy datasets. In addition, interpretability techniques such as Grad-CAM are employed to visualise the decision-making process, offering greater transparency and trust in model predictions. The proposed approach is evaluated using the RealWaste dataset, a large-scale collection of authentic landfill images, ensuring that results reflect real-world complexities rather than controlled conditions.

By bridging the gap between advanced neural network architectures and real-world waste management needs, this work contributes to the broader goal of environmental sustainability. Enhanced classification performance, combined with interpretability and adaptability, has the potential to support scalable, automated waste sorting systems that reduce landfill dependency, promote recycling, and optimise resource recovery.

1.2 Problem Statement

Effective waste management is essential for achieving environmental sustainability, yet current automated classification systems face major shortcomings. Many existing methods rely on pristine or synthetic datasets, which fail to

capture the complexity of real-world landfill waste streams. This gap results in models that resemble students trained only with textbook examples—they often perform poorly when confronted with messy, ambiguous, or mixed waste categories in practice. Conventional CNN-based approaches also struggle to capture fine-grained features needed to distinguish such categories, limiting their generalisability and scalability in real-world deployment.

To overcome these challenges, there is a need for enhanced model architectures that can focus more intelligently—like a human eye zooming in on the most relevant details of an object—while adapting to the irregularities of authentic waste environments. This motivates the exploration of attention-based neural networks, which offer the potential to bridge these gaps and contribute to more accurate, reliable, and sustainable waste management solutions.

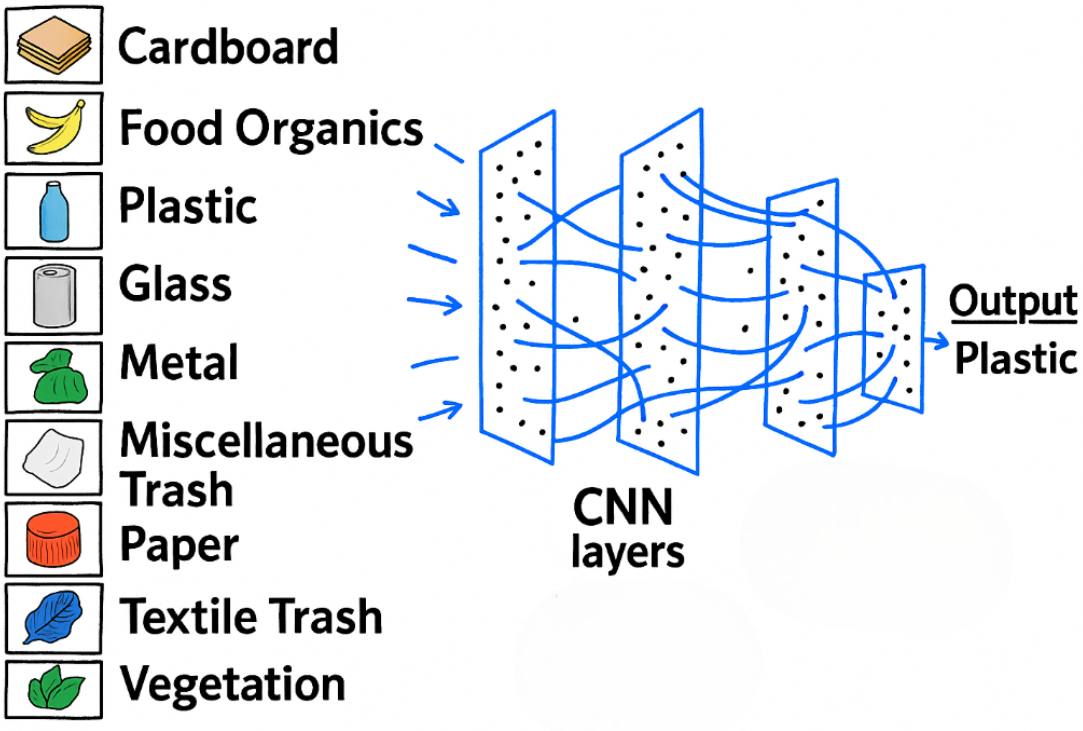


Fig. 1.1 High-level inference schematic: an input waste image is processed by CNN layers and produces class probabilities over the nine categories; the top class (here, *Plastic*) is selected[1].

1.3 Research Challenges

A number of practical constraints influenced both the design and evaluation of our models:

- **Class imbalance.** The RealWaste dataset contains 4,752 images unevenly distributed across nine classes. Plastic is heavily overrepresented (921 images), whereas classes like textile trash have fewer than 320 examples. Such imbalance biases learning toward majority categories and weakens performance on minority classes.
- **Hardware and CPU limitations.** Limited availability of GPU accelerators meant that many experiments had to run on CPUs. This significantly increased training times and restricted hyperparameter tuning. Accessing remote GPUs (e.g., via Google Colab) introduced additional complexity and time overhead.
- **Balancing model complexity and efficiency.** Deeper neural networks typically deliver higher accuracy but require substantial computational resources and memory. Given our limited hardware, we needed to select models that provided strong performance while remaining tractable on CPU-only systems. Striking this balance constrained architectural choices and motivated the use of attention modules to boost accuracy without excessive depth.
- **Limited computational capacity.** Even with CPUs, memory and processing constraints impacted batch size and input resolution, especially for high-resolution images. Some large models could not be trained effectively on available hardware, restricting the scope of experiments. Future work should address these limitations through memory-efficient architectures or access to more powerful hardware.

1.4 Key Contributions

This thesis makes the following key contributions toward advancing automated waste classification for environmental sustainability:

1. **Problem-driven Dataset Usage:** We highlight the limitations of existing pristine or synthetic datasets and adopt the *RealWaste* dataset, which better represents the complexity of real landfill environments. This ensures that the evaluation and results are grounded in realistic conditions rather than controlled laboratory scenarios.

2. **Enhanced Model Architecture with Attention Mechanism:** We design and implement an attention-based neural network that extends conventional CNN approaches. By allowing the model to focus selectively on the most informative regions of an image, the architecture improves the ability to distinguish between fine-grained and ambiguous waste categories.

3. **Comprehensive Performance Evaluation:** Beyond standard metrics such as accuracy, precision, recall, and F1-score, we employ Grad-CAM visualizations to provide interpretability. This dual perspective—quantitative and qualitative—demonstrates not only how well the model performs but also why it makes specific predictions.

4. **Comparison with Established Architectures:** We systematically compare the proposed attention-based model against widely used deep learning architectures (e.g., VGG16, ResNet, DenseNet, Inception, MobileNet) to establish benchmarks and highlight the performance gains achieved through attention integration.

5. **Contribution to Environmental Sustainability:** By improving the robustness and interpretability of waste classification, this work supports

the broader goal of sustainable waste management. More accurate classification directly contributes to better recycling, reduced landfill burden, and improved environmental outcomes.

1.5 Thesis Organization

This thesis is organised to guide the reader from motivation and background through methodology, experiments and conclusions. Chapter 1 (*Introduction*) outlines the environmental and societal challenges of improper waste management, motivates the need for automated waste sorting, and states the research objectives and scope. Chapter 2 (*Literature Review*) surveys related work on waste-classification and attention mechanisms in deep learning, identifying gaps in existing methods. Chapter 3 (*Methodology*) describes the RealWaste dataset and preprocessing, introduces the transfer-learning framework, details the backbone architectures and the channel-attention classifier, and outlines the training strategy. Chapter 4 (*Results and Analysis*) presents experimental findings: it reports quantitative metrics, discusses model performance, examines confusion matrices and Grad-CAM visualisations, and compares the various models without divulging specific numerical results here. Chapter 5 (*Conclusion and Future Work*) summarises the contributions, discusses implications of the findings, and proposes directions for future research and system improvement. By structuring the thesis this way, readers can progressively understand the context, methods, results, and implications of the study.

Chapter 2

Literature Review

In recent years, waste classification has emerged as an important research focus at the intersection of environmental science and machine learning, owing to its potential to advance sustainable waste management. A wide body of literature has explored both traditional and modern approaches, ranging from handcrafted feature extraction pipelines and classical machine learning to deep learning models capable of end-to-end training. With the advent of convolutional neural networks (CNNs), image classification for waste sorting has achieved significant improvements in accuracy and generalisation, while recent work has further integrated attention mechanisms to enhance recognition of small, complex, or occluded objects. Building on these developments, the following review examines key studies, methodological innovations, and technological contributions that inform the present research.

Early approaches to garbage classification relied on traditional machine learning techniques with handcrafted features. Meng and Chu [2] compared histogram of oriented gradients (HOG) combined with support vector machines (SVM) against convolutional neural networks (CNNs). Their findings demonstrated that deep models, particularly ResNet50 with data augmentation, significantly outperformed classical methods, achieving an accuracy of 95.35%. Similarly, Wang [3] employed the VGG16 architecture for a four-class classification task, integrating preprocessing, batch normalization,

and ReLU activations to stabilise training. While the system achieved 75.6% accuracy, it highlighted both the promise and limitations of early CNN-based methods when applied to practical waste sorting.

Lightweight CNN architectures soon emerged to enable deployment on resource-constrained devices. Rabano et al.[4] retrained MobileNet using transfer learning from ImageNet on the TrashNet dataset, achieving 87.2% accuracy. Importantly, their system was implemented as an Android application, demonstrating the feasibility of mobile-based real-time garbage classification. Expanding on this direction, Guo, Shi, and Wang [5] enhanced EfficientNet with group normalization and an attention mechanism to address small-batch training challenges. Their model, evaluated on 19,735 labelled images from the Huawei AI competition, achieved an average accuracy of 93.47% and a peak of 98.3%, underscoring its generalisation ability and practical relevance for intelligent bins and robotic systems.

More recent studies have focused on optimising network architectures for both efficiency and industrial applicability. Cao et al. [6] proposed a CNN framework combining residual modules with depthwise separable convolutions, significantly reducing parameter size without compromising accuracy. Qin et al. [7] approached the challenge from an Industry 4.0 perspective, presenting precision measurement strategies that informed deep learning-based classification under industrial standards. Zhang et al. [8] improved feature extraction through an attention-augmented DenseNet, reporting superior recognition accuracy while maintaining computational efficiency. Further, an improved MobileNetV3-Large has been introduced for intelligent garbage classification, balancing accuracy with low-latency inference to support deployment on mobile and embedded systems.Collectively, these studies illustrate the transition from traditional handcrafted methods to deep CNNs and, more recently, to lightweight and attention-enhanced models capable of real-time, scalable, and industrial-grade waste classification.

The introduction of Convolutional Neural Networks (CNNs) marked a paradigm shift toward end-to-end feature learning. Architectures such as AlexNet, VGG, ResNet, and GoogLeNet consistently outperformed traditional pipelines, demonstrating the effectiveness of hierarchical feature extraction [9, 10]. Benchmark datasets, including MNIST and ImageNet/ILSVRC [11, 12], provided standardized evaluation and facilitated progressive improvements in network design and training protocols. Large-scale datasets, coupled with GPU acceleration and data augmentation, were shown to be critical in achieving state-of-the-art accuracy. Transfer learning from pretrained CNNs further reduced training costs while often improving performance on downstream tasks [13, 14].

For deployment in resource-constrained environments, lightweight CNN architectures such as MobileNet and its variants were introduced [15]. These architectures employ factorized convolutions and adjustable width/resolution trade-offs to optimize the balance between accuracy and computational efficiency. More recently, Vision Transformers (ViTs) have emerged as competitive alternatives for image classification, offering data-efficient and lightweight designs that rival CNNs on multiple benchmarks [16, 17]. Comparative studies across deep-learning frameworks (Keras, PyTorch, MXNet) report non-trivial differences in accuracy, reproducibility, and training efficiency [18], supporting the adoption of PyTorch for controlled experimental setups.

Architecture-specific findings highlight critical design considerations. Basha et al. [19] reported that deeper networks can achieve high performance with fewer parameters in fully connected layers, whereas shallower networks require increased width to maintain comparable results. Image resolution is also a significant factor: Sabottke and Spieler [20] found that higher resolutions preserve fine details but necessitate smaller batch sizes due to memory constraints, while Tang et al. [21] demonstrated that overly large

or excessively small input images can degrade classification performance. Based on these insights, a resolution of 224×224 is widely adopted, balancing visual fidelity and computational efficiency.

Goyal et al. [22] propose a linear scaling rule for large mini-batch training, increasing the learning rate proportionally to batch size and using a short warm-up period to stabilize early training. This approach enables ResNet-50 to be trained on ImageNet with up to 8192-image batches without loss of accuracy, achieving high parallel scaling efficiency and demonstrating that careful batch-size and learning-rate scheduling can accelerate training while maintaining generalization.

Data preprocessing and augmentation are essential for improving model generalization, particularly in small or imbalanced datasets. Shijie et al. [23] explored various augmentation strategies on AlexNet using CIFAR-10 and ImageNet, while Lopez de la Rosa et al. [24] showed that geometric transformations can significantly boost the mean F1-score in defect detection tasks. Kumar et al. [25] further demonstrated that combining basic transformations—random flips, rotations, and resizing—yields consistent performance gains, highlighting the importance of augmentation for robust model training.

Normalization techniques such as Layer Normalization and AdaNorm stabilize gradient flow during backpropagation and mitigate overfitting [26]. Dropout has also been employed as a regularization method to improve generalization in fully connected layers [27]. In the context of class imbalance, Rezaei and Dastjerdehei [28] demonstrated that weighted cross-entropy effectively improves the learning of minority classes without requiring complex resampling, an approach that informs the use of class-weighted loss in this work.

CNNs trained on large-scale datasets such as ImageNet/ILSVRC have enabled deeper, more expressive networks [29, 12, 30–35]. Attention mechanisms, originally proposed in NLP Transformers [36], have been successfully

adapted to vision tasks. Channel attention modules, including Squeeze-and-Excitation (SE) blocks [37], allow networks to recalibrate feature maps by emphasizing informative channels, improving discriminative power and interpretability. Grad-CAM visualizations are frequently applied to highlight salient regions influencing network predictions, enhancing transparency in decision-making.

Kumar et al. [38] demonstrate that Grad-CAM produces class-specific heatmaps highlighting the key areas of an image that drive a CNN's prediction. These localized visual explanations link the network's internal activations to specific spatial regions, enabling interpretable insights into the model's decision process. By overlaying Grad-CAM maps on the input, one can verify whether the network attends to actual waste objects rather than irrelevant background. Such visual evidence helps identify misclassifications or biases when the model focuses on incorrect regions. In the proposed attention-based CNN for waste classification, Grad-CAM is similarly applied to validate that each predicted label corresponds to the highlighted regions, thereby improving model transparency and trust.

Application-specific studies have further motivated attention-based transfer learning. Abu et al. [1] demonstrated MobileNet's efficacy on flower classification with per-class accuracy gains, while Ahmed et al. [39] applied multiple CNN architectures on the TrashBox waste dataset, validating transfer learning and attention mechanisms for practical waste-image classification.

Early image classification relied on hand-crafted feature pipelines combined with algorithms such as Naive Bayes, K-Nearest Neighbor, and Support Vector Machines [40], but their dependence on manually engineered features constrained both accuracy and scalability. The advent of deep learning, particularly convolutional neural networks (CNNs), transformed the field by enabling end-to-end feature learning directly from raw pixel data and achieving state-of-the-art performance across standard benchmarks. CNNs

demonstrated the ability to capture hierarchical representations of visual information, significantly improving robustness compared to traditional methods. More recently, attention mechanisms have been introduced to complement CNNs by focusing computational resources on salient image regions, thereby enhancing the recognition of small, complex, or occluded objects. These advances have established the foundation for modern waste classification systems, motivating research into more accurate, efficient, and interpretable models—directions that this thesis seeks to extend.

Existing research in waste classification using deep learning reveals several limitations.Many studies rely on small datasets, exhibit limited generalization to diverse real-world scenarios, and provide insufficient analysis of model robustness and practical deployment. Furthermore, interpretability techniques, such as Grad-CAM, are often absent, restricting insights into the model's decision-making process. Common issues include overfitting, discrepancies between training and testing performance, and a lack of thorough evaluation in practical settings.Although some models achieve high accuracy, they frequently fail to generalize effectively. This research addresses these gaps by developing an attention-based neural network that leverages a medium and more diverse waste dataset, integrates advanced attention-guided transfer learning to improve robustness and accuracy, and employs interpretability techniques such as Grad-CAM to provide transparent, environmentally meaningful insights into model decisions.This approach aims to deliver a reliable and efficient waste-classification system capable of performing effectively in real-world applications.

Chapter 3

Methodology

3.1 Introduction

Image classification tasks can be approached using a variety of neural network architectures, including artificial, recurrent, and convolutional neural networks. These architectures differ in how they process data and the specific tasks they are best suited for.

In this study, we explore several models for waste classification. Our backbone set includes MobileNetV2 for its lightweight efficiency, ResNet-50 for its residual skip connections that ease optimization and provide a strong accuracy-compute trade-off, DenseNet-121 for its deep, densely connected layers, Inception–ResNet V2 for its hybrid residual–Inception design, Inception V3 for its multi-branch (Inception) modules, and VGG-16 as a comparatively shallow baseline. Additionally, we introduce a compact *custom CNN*, tailored specifically for the RealWaste dataset and trained from scratch with class-weighted loss and early stopping, as a non-pretrained baseline.

MobileNetV2:

- Lightweight architecture optimized for mobile and edge devices.
- Uses depthwise separable convolutions to reduce computational cost.

- Ideal for applications requiring low-latency inference.

ResNet-50 (Residual Networks):

- Incorporates residual (skip) connections to alleviate vanishing gradients.
- Enables deeper architectures with stable training dynamics.
- Widely used for feature extraction and image classification.

DenseNet121 (Dense Convolutional Networks):

- Connect each layer to every other layer, promoting feature reuse.
- Requires fewer parameters while achieving competitive performance.
- Effective for image recognition and transfer learning.

VGG16:

- VGG16 is a deep convolutional neural network architecture recognised for its straightforward design and strong performance.
- It is composed of 16 layers that utilise compact 3×3 convolutional filters.
- Performs well in transfer learning tasks with high-quality feature extraction.

Inception V3:

- Uses *Inception* (multi-branch) modules to capture features at multiple receptive-field scales within each block.
- Employs factorised convolutions to reduce compute, e.g., $5 \times 5 \rightarrow 2 \times (3 \times 3)$ and asymmetric $n \times 1$ / $1 \times n$ filters.
- Includes auxiliary classifiers and other regularisation tricks that stabilise training; a strong baseline for transfer learning.

Inception-ResNet V2:

- Hybrid architecture that combines Inception modules with *residual* (skip) connections for easier optimisation of very deep networks.
- Maintains multi-scale feature extraction while improving convergence and accuracy compared to pure Inception variants.
- Well-suited for fine-tuning on downstream tasks, often achieving competitive state-of-the-art performance among CNN backbones.

Custom CNN:

- Compact four–block convolutional backbone (32–256 channels) with ReLU and 2×2 max-pooling, tailored to our dataset and hardware budget.
- Uses adaptive average pooling to a fixed 7×7 grid and a lightweight classifier (512-unit fully connected layer with dropout), improving robustness and regularization.
- Trained from scratch on Real-Waste with class-weighted cross-entropy to handle class imbalance; integrates naturally with our augmentation pipeline.
- Remains interpretable via Grad-CAM, enabling qualitative analysis of salient regions and failure cases.

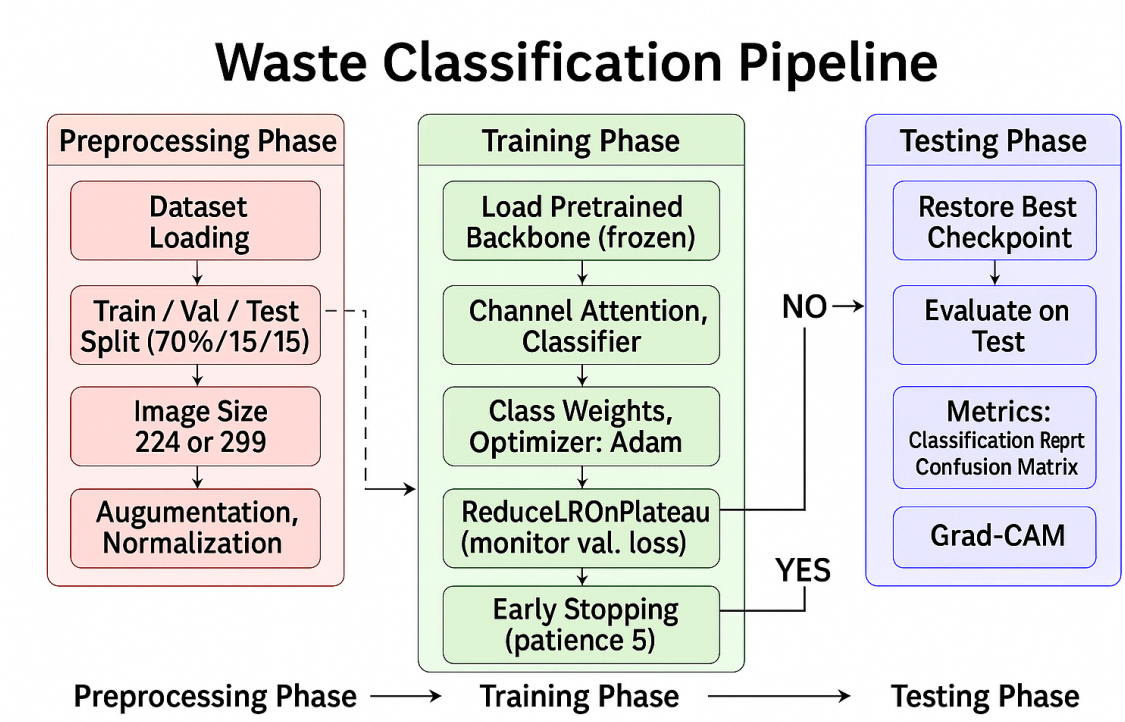


Fig. 3.1 Proposed Methodology.

As shown in Fig. 3.1, our pipeline consists of three phases—**Preprocessing**, **Training**, and **Testing**. We discuss each step in detail below.

3.2 Data Preparation

In this section, we describe the data preparation process for the *RealWaste* dataset. The dataset is structured as a directory containing subdirectories for each class, where each subdirectory holds images corresponding to that class. We use PyTorch for handling the dataset and data loaders to efficiently manage data loading and batching. The following workflow outlines the key steps involved in preparing the data for training and evaluation:

Table 3.1 Number of images according to waste type (RealWaste dataset).

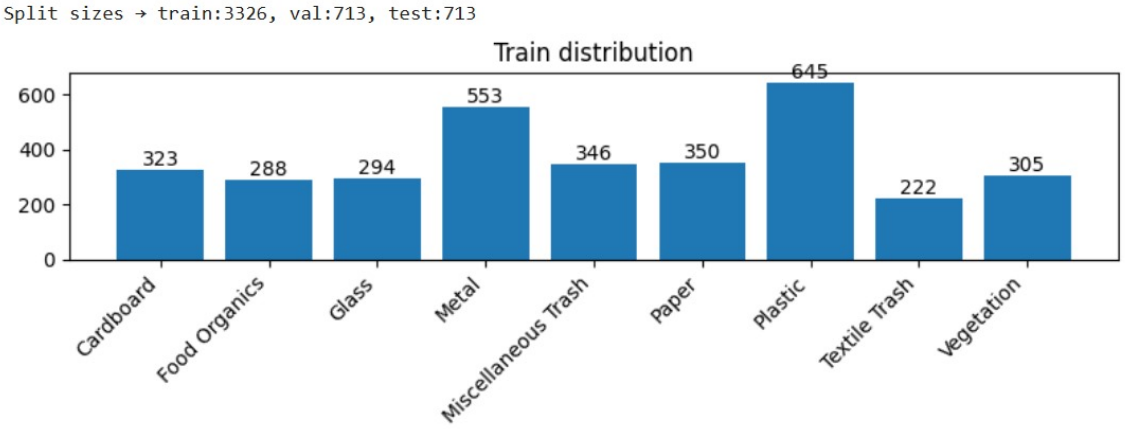
No.	Waste Type	No. of Images
1.	Cardboard	461
2.	Food Organics	411
3.	Glass	420
4.	Metal	790
5.	Miscellaneous Trash	495
6.	Paper	500
7.	Plastic	921
8.	Textile Trash	318
9.	Vegetation	436
	Total images	**4752**

3.2.1 Training Images

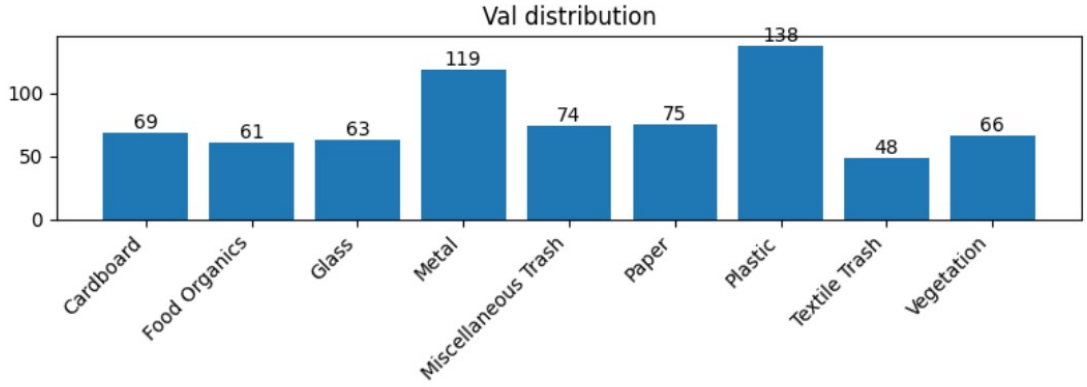
All experiments are conducted on the *RealWaste* dataset, which contains **4,752** RGB images across **nine** waste categories.The dataset is split into **70%/15%/15%** for training, validation, and testing while preserving class distribution (Fig. 3.2), and class imbalance is mitigated using **class-weighted cross-entropy** with per-class weights $w_c = \frac{N}{K \cdot n_c}$, where N is the number of training samples, K the number of classes, and n_c the sample count of class c.

3.2.2 Image size and preprocessing

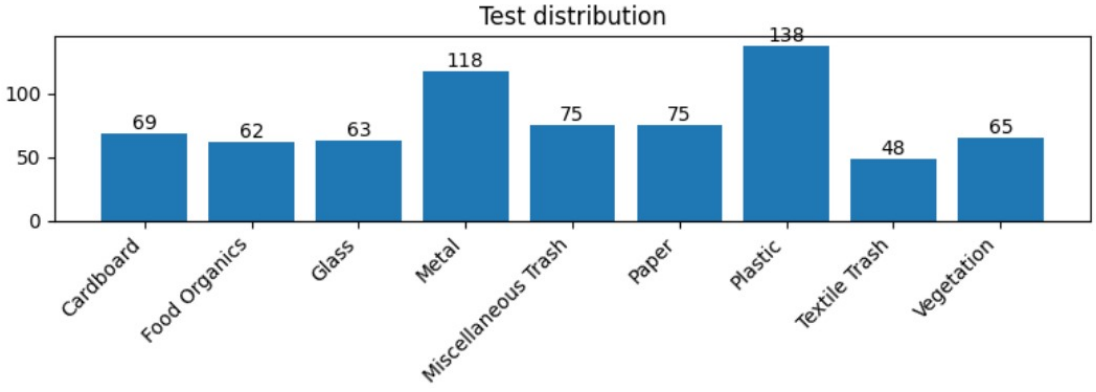
Images were resized to the canonical input size of each backbone: **224×224** for VGG16, ResNet-50, DenseNet-121, and MobileNetV2, and **299×299** for Inception V3 and Inception–ResNet V2. For training we applied `RandomResizedCrop` (224 or 299) and `RandomHorizontalFlip`. For validation/test we resized the short side to **256** (or **342** for 299-pixel models) and center-cropped to the target size. All inputs were tensorized and normalized using ImageNet statistics ($\mu = \{0.485, 0.456, 0.406\}$, $\sigma = \{0.229, 0.224, 0.225\}$).



(a) Train split (70%): per-class counts; total = 3326.



(b) Validation split (15%): per-class counts; total = 713.



(c) Test split (15%): per-class counts; total = 713.

Fig. 3.2 Stratified class distribution across the RealWaste dataset after a 70/15/15 split.

3.2.3 Data augmentation

To improve generalisation, we apply light, on-the-fly augmentation during training: `RandomResizedCrop`(224 *or* 299) and `RandomHorizontalFlip`. These transformations are applied only to the training split; validation and test use deterministic resizing and centre-cropping as described above. Additionally, for the non-pretrained custom CNN baseline we used mild `RandomRotation` ($\pm 15^{\circ}$) and `ColorJitter` to further regularise the model.

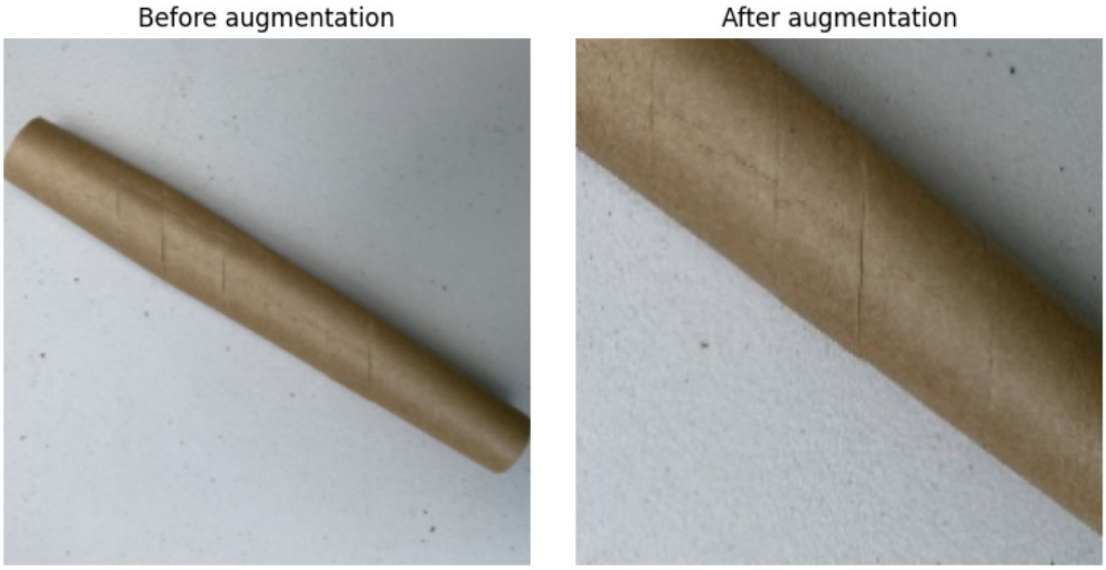


Fig. 3.3 Example of augmentation on the RealWaste dataset: the same sample **before** (left) and **after** (right) training-time transforms.

3.3 Training and Classifying the Waste Types Using Deep Neural Networks

3.3.1 Model architecture

Our model (Fig. 1.1) follows a standard CNN pipeline: an RGB input (224×224 for VGG16/ResNet50/DenseNet121/MobileNetV2 and 299×299 for InceptionV3/Inception–ResNetV2).

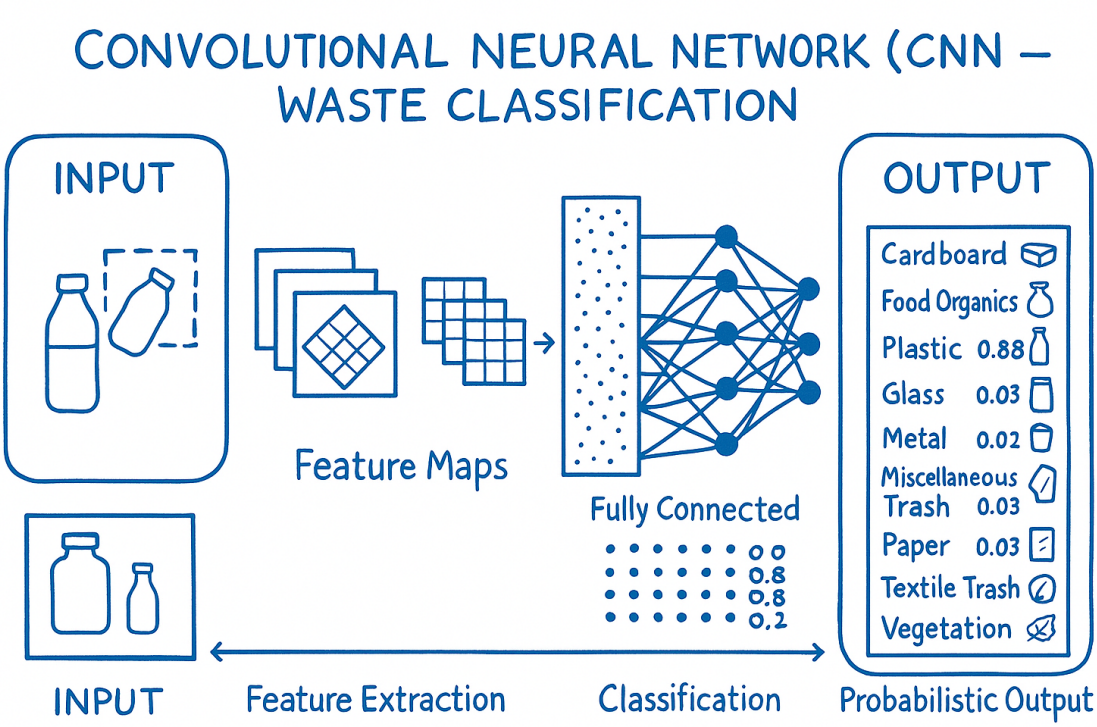


Fig. 3.4 CNN for waste classification: a cropped input patch is transformed into feature maps, passed through a fully connected layer, and yields class probabilities (Plastic highest)[41].

We evaluate six pretrained backbones—VGG16, MobileNetV2, ResNet-50, DenseNet-121, InceptionV3, and Inception–ResNetV2—under a feature-extraction regime; we also include a compact custom CNN with four Conv–ReLU–MaxPool blocks (32–64–128–256 channels), a channel-attention gate, adaptive average pooling, and a two-layer classifier.

All convolutional backbone layers are frozen, and the task head is trained from scratch on RealWaste. Concretely, we insert a lightweight *channel-attention* gate on the final feature map and replace the original classifier with a new fully connected layer mapping to C classes (here C=9). Only the attention module and the classifier parameters are updated during training; the backbone weights remain fixed. This setup standardises the input pipeline and optimisation protocol across models, enabling a fair comparison of archi-

tectures under identical data conditions. (For Inception-based models, inputs are 299×299; the auxiliary head is not used at evaluation.)

Table 3.2 Backbones and configuration used for transfer learning.

Model	Input size	Final feature channels	Trainable modules
VGG16	224×224	512	Attention + classifier
ResNet-50	224×224	2048	Attention + classifier
MobileNetV2	224×224	1280	Attention + classifier
DenseNet-121	224×224	1024	Attention + classifier
Inception v3	299×299	2048	Attention + classifier
Inception–ResNet v2 (timm)	299×299	1536	Attention + classifier

3.3.2 How the Attention Mechanism Works

Attention mechanisms, inspired by human cognition, improve image classification by highlighting the most informative regions of an image, which is particularly valuable for waste recognition. Within convolutional neural networks (CNNs), attention is applied to intermediate feature maps, where it assigns importance scores to different spatial regions. This is achieved by combining local features, which capture fine-grained details, with global features that provide contextual information across the entire image. The resulting attention map assigns higher weights to relevant areas (e.g., waste objects) while suppressing less useful regions such as the background. These re-weighted features are then passed to the classification layer, enabling the network to focus on critical details and achieve more accurate predictions.

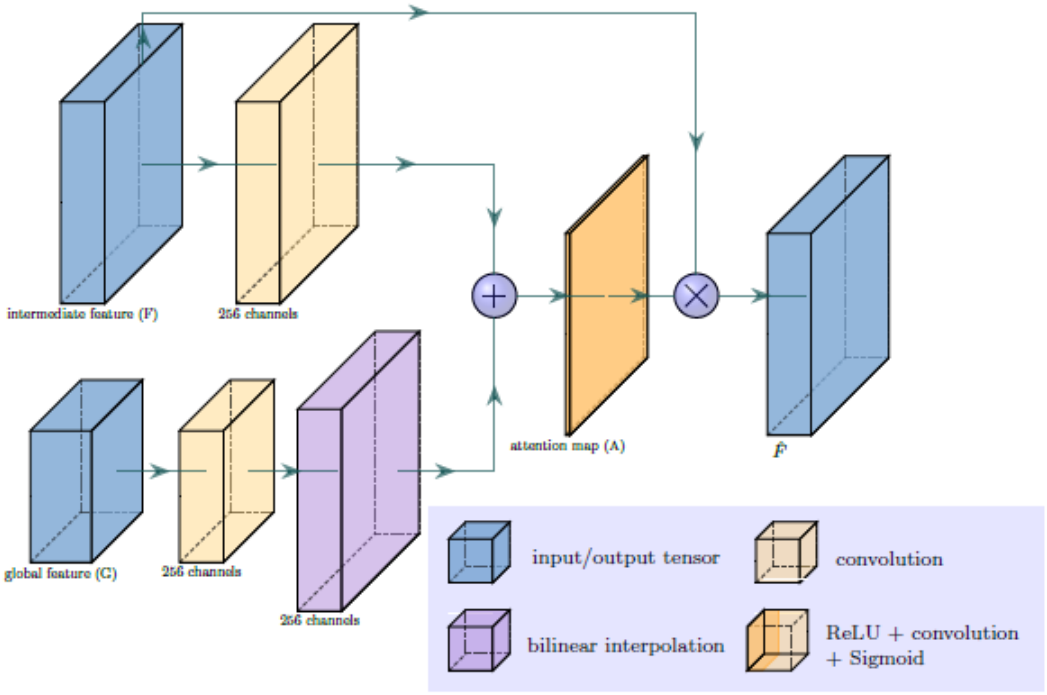


Fig. 3.5 Inner architecture of the attention module [42].

In this work, attention modules are integrated into multiple pretrained CNN architectures—ResNet-50, DenseNet-121, MobileNetV2, InceptionV3, VGG-16, and Inception-ResNetV2—as well as a custom CNN, to classify waste categories including plastic, glass, paper, and cardboard. This demonstrates the effectiveness of attention mechanisms in both improving performance and enhancing interpretability across diverse architectures.

3.3.3 Implementation & Reproducibility

All experiments in this project were implemented in **Python** using the **PyTorch** framework (`torchvision`; `timm` for Inception–ResNet V2). The model training and evaluation were carried out in the **Google Colab** environment utilizing a single NVIDIA GPU (CUDA). Checkpoints of the best-performing model were saved to Google Drive for easy restoration during testing.

The following Python packages were used for various utilities throughout the implementation:

`scikit-learn` for generating classification metrics and confusion matrix,

`NumPy` for numerical operations,

`Matplotlib` for data visualization (such as training curves and confusion matrix plots),

`tqdm` for progress bars in loops.

To ensure reproducibility, random seeds were fixed for Python and PyTorch. Class-weighted cross-entropy loss was used to handle class imbalance, with the `ReduceLROnPlateau` scheduler dynamically adjusting the learning rate based on validation loss. Early stopping with a patience of five epochs prevented overfitting. The best model was selected according to peak validation accuracy, with its weights restored for final evaluation on the test set. Training was performed with a batch size of 32 and an initial learning rate of 1×10^{-3}.

Below is a breakdown of the main components used in the code:

- **Dataset**: We utilised the RealWaste dataset, publicly available on Kaggle, comprising 4,752 RGB images of everyday waste items collected under realistic disposal conditions.A notable challenge of this dataset lies in its class imbalance: for instance, the plastic category contains 921 images, while textile trash includes only 318. Such disparities make it a demanding benchmark, underscoring the necessity of techniques like class-weighted loss to ensure fair model learning across all categories.
- **Models**: We trained several pretrained models, including:
  - **MobileNetV2**
  - **VGG16**
  - **Inception V3**
  - **DenseNet121**

  - **Inception-ResNetV2**
  - **ResNet50**

  Each pretrained architecture was fine-tuned for the waste classification task by replacing the final fully connected layer with one outputting nine classes, while keeping the convolutional backbone frozen to preserve its pretrained feature representations.

  In addition, we developed a **Custom CNN** designed with a lightweight architecture for efficient computation in resource-constrained settings, such as edge devices. This network comprises four convolutional blocks, each followed by max-pooling to progressively reduce spatial dimensions, and concludes with a fully connected layer for final classification.

- **Training with and without Attention**: We performed training twice: once without the attention mechanism and once with an attention mechanism added to the model. The attention mechanism helps the model focus on more relevant features, which may improve classification performance, especially in complex tasks like waste classification where visual cues can be subtle. The implementation with attention utilized a **Channel Attention** module that was inserted after the feature extraction layers of the pretrained models.

- **Training Strategy**: Models were trained using the Adam optimizer with `ReduceLROnPlateau` scheduling, early stopping, and checkpointing to retain the best weights.

  For the training process, we employed the following hyperparameters:

  Batch size: 32,

  Maximum number of epochs: 50,

  Initial learning rate: 1×10^{-3},

  Patience for early stopping: 5.

The training history (loss and accuracy per epoch) was tracked and visualized using `Matplotlib`, showing the improvement of the model over time.

Reproducibility was ensured by fixing random seeds and saving all model checkpoints to Google Drive. All required code, datasets, and dependencies have been documented to facilitate future work and ensure the results can be reproduced under the same conditions.

3.3.4 Custom CNN for Waste Classification

Dataset & Preprocessing

We use the **RealWaste** image dataset arranged in class-specific directories. Images are resized to $\mathbf{224 \times 224}$ and normalized with ImageNet statistics ($\mu = \{0.485, 0.456, 0.406\}$, $\sigma = \{0.229, 0.224, 0.225\}$). The dataset is split **70%/15%/15%** into training/validation/test, preserving class distribution via folder structure. To improve generalization, we apply online augmentation on the training split:

- `RandomResizedCrop(224)`, `RandomHorizontalFlip`
- `RandomRotation(`$\pm 15^{\circ}$`)`
- `ColorJitter` (brightness/contrast/saturation/hue)

Network Architecture

The proposed model is a lightweight **CNN with channel attention**. The feature extractor comprises four convolutional blocks followed by a squeeze-and-excite–style attention module, and a fully connected head:

- **Block 1:** Conv($3 \to 32$, 3×3, pad 1) $\to$ ReLU $\to$ MaxPool(2×2)
- **Block 2:** Conv($32 \to 64$, 3×3, pad 1) $\to$ ReLU $\to$ MaxPool(2×2)

- **Block 3:** Conv($64 \to 128$, 3×3, pad 1) $\to$ ReLU $\to$ MaxPool(2×2)
- **Block 4:** Conv($128 \to 256$, 3×3, pad 1) $\to$ ReLU $\to$ MaxPool(2×2)

Channel Attention: given the final feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ with C=256, we compute both global average and max pooling, concatenate them, and pass through a two-layer MLP with sigmoid gating to obtain channel weights $\mathbf{a} \in \mathbb{R}^{C}$. The reweighted features are $\tilde{\mathbf{X}} = \mathbf{X} \odot \mathbf{a}$ (broadcast along spatial dimensions).
Classifier Head: AdaptiveAvgPool(7×7) $\to$ Flatten $\to$ FC($256\cdot7\cdot7 \to 512$) $\to$ ReLU $\to$ Dropout(0.5) $\to$ FC($512 \to C$ classes).

Training Procedure

We optimize the cross-entropy loss with **class weights** computed from the training-set label frequencies to mitigate imbalance. Unless stated otherwise, we use the settings in Table 3.3. A ReduceLROnPlateau scheduler monitors validation loss and decreases the learning rate by a factor of 0.1 on plateau. We employ **early stopping** with patience of 5 epochs based on validation loss. The **best checkpoint** is selected by the highest validation accuracy.

Table 3.3 Training hyperparameters for the Custom CNN with attention.

Optimizer	Adam
Initial learning rate	1×10^{-3}
Batch size	32
Max epochs	50
Scheduler	ReduceLROnPlateau (monitor: val loss, factor 0.1, patience 2)
Early stopping	Patience 5 (monitor: val loss)
Checkpointing	Best validation accuracy (state dict saved)
Loss	Cross-entropy with class weights
Input resolution	224×224
Normalization	ImageNet mean/std

3.3.5 Evaluation & Explainability

In the evaluation phase, model performance was assessed on the test set using accuracy, precision, recall, and F1-scores, with a confusion matrix generated via `scikit-learn`. To enhance interpretability, **Grad-CAM** heatmaps were produced, highlighting the regions most influential in the model's predictions.The evaluation measures are discussed in detail in Section 4.2.

Chapter 4

Experiments and Evaluation

This chapter reports the performance of the evaluated models on the RealWaste dataset, highlighting the impact of attention mechanisms through key metrics, confusion matrices, and Grad-CAM visualisations.

4.1 Dataset Description

The dataset used for this project, referred to as *RealWaste*, is an image classification dataset specifically designed for waste classification. It consists of color images of waste items, collected from an authentic landfill environment. The dataset contains a diverse range of waste materials, making it suitable for training convolutional neural networks (CNNs) to classify real-world waste materials.

4.1.1 Overview

RealWaste is designed to represent real-world waste classification scenarios. The images are taken from a landfill, where various waste materials are present in their natural, unprocessed state. This allows the dataset to

closely mimic real-life conditions in waste sorting facilities, making it highly applicable for waste management and recycling operations.

4.1.2 Dataset Information

Purpose of the Dataset: *RealWaste* was developed as part of an honours thesis to examine the performance of convolutional neural networks (CNNs) on real-world waste materials. Its main aim is to assess model accuracy on authentic waste images, in contrast to pristine or synthetic datasets, thereby evaluating CNNs' ability to handle the complexities of real landfill conditions.

Instances in the Dataset: The dataset consists of colour images of waste items collected at landfill reception points. Each image is provided at a resolution of 524×524 pixels, as specified in the accompanying research paper. Access to higher-resolution images can be requested from the dataset's corresponding author.

4.1.3 Classes and Labels

The dataset includes nine major material types, each representing a waste category found in the landfill. The material types are:

- **Cardboard**
- **Food Organics**
- **Glass**
- **Metal**
- **Miscellaneous Trash**
- **Paper**
- **Plastic**

- **Textile Trash**
- **Vegetation**

Each image in the dataset is labeled with one of these categories, and the labels represent the material type present in the image. These labels are crucial for training the machine learning model to classify new images based on the waste type.

4.1.4 Dataset Statistics

The dataset contains the following number of images per class:

- Cardboard: 461 images
- Food Organics: 411 images
- Glass: 420 images
- Metal: 790 images
- Miscellaneous Trash: 495 images
- Paper: 500 images
- Plastic: 921 images
- Textile Trash: 318 images
- Vegetation: 436 images

In total, the dataset contains 4,752 images, each belonging to one of the nine categories listed above. These images are diverse in terms of waste types and their appearance in a landfill environment, providing a comprehensive sample for training a robust waste classification model.The dataset is well-suited for training deep learning models due to its diversity and realistic representation of waste materials in a landfill environment.

4.1.5 Missing Values

There are no missing values in this dataset, and each image is correctly labeled with its corresponding class.

(a) Food Organics

(b) Cardboard

(c) Glass

(d) Metal

(e) Miscellaneous Trash

(f) Paper

(g) Plastic

(h) Textile Trash

(i) Vegetation

Fig. 4.1 Dataset's sample images[43].

4.1.6 Dataset Source

This dataset was collected from Kaggle, a popular platform for datasets and machine learning competitions. The dataset has been made publicly available for research purposes, and it provides a valuable resource for studying waste classification using deep learning.

4.1.7 Conclusion

The *RealWaste* dataset serves as an excellent resource for training and evaluating image classification models, especially in the context of waste management. By using this dataset, we aim to develop a model capable of accurately classifying various waste materials in a real-world setting, which could be directly applied to improve waste sorting processes in industries like recycling centers and l

4.2 Evaluation Measure

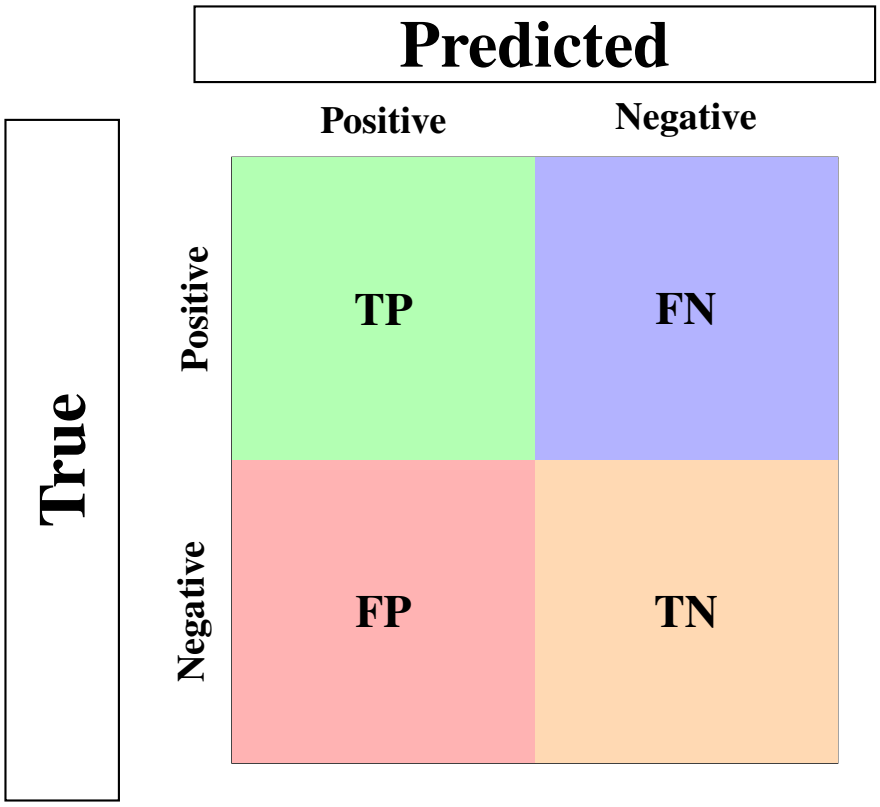
The model was evaluated on the test set using accuracy, precision, recall, and F1-score, with Grad-CAM heatmaps providing interpretability by highlighting key regions influencing predictions.

4.2.1 Model Evaluation

After the model has been trained and the best-performing weights have been saved (based on validation accuracy), we evaluate the model on the held-out test set. The evaluation process includes calculating several key performance metrics to understand how well the model generalizes to unseen data.

- **Overall Accuracy**: The test accuracy is calculated as the ratio of correctly classified images to the total number of images in the test set. This provides a high-level view of the model's performance.
  $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$
- **Classification Report**: In addition to overall accuracy, we also compute the classification report for each class. The report includes:
  * **Precision**: The proportion of true positive predictions out of all predicted positives for each class.
    $Precision = \frac{TP}{TP+FP}$
  * **Recall**: The proportion of true positive predictions out of all actual positives for each class.
    $Recall = \frac{TP}{TP+FN}$
  * **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of the model's performance across all classes.
    $F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$

  The classification report helps identify how well the model performs for each waste category, revealing whether there are any classes with particularly poor performance.
- **Confusion Matrix**: We also generate a confusion matrix, which visualizes the performance of the classifier by showing how often predictions of one class are mistakenly classified as another. The matrix is a valuable tool for identifying classes that are often confused with others, helping to improve the model in future iterations.



These metrics are computed using the true labels and predicted labels from the model's output, and are presented in a classification report, followed by a confusion matrix for a more detailed evaluation.

4.2.2 Grad-CAM for Model Explainability

While metrics such as accuracy, precision, and F1-score indicate how well a model performs, they do not explain why certain predictions are made. To address this, we employ Grad-CAM (Gradient-weighted Class Activation Mapping), a widely used technique that generates heatmaps to visualise the image regions most influential for the model's decisions. These heatmaps are derived from the final feature maps of the model (post-attention), enabling us to understand where the model focuses during classification.

- **Grad-CAM Procedure**: A test image is passed through the model, and the gradients of the predicted class score with respect to the last convolutional feature maps are calculated. These gradients are averaged to obtain weights, which are then combined with the feature maps to produce a heatmap that highlights the most influential regions.
- **Heatmap Overlay**: The generated heatmap is superimposed on the original image, providing a clear visualisation of the regions that guided the classification decision. This allows us to assess whether the model attends to relevant features, such as waste objects, or is distracted by irrelevant background areas.
- **Class Activation Visualisation**: For each sample, the Grad-CAM heatmap is displayed alongside the original image. This comparison helps interpret whether the model's focus aligns with the predicted class label, offering deeper insight into the reasoning process behind each prediction.

Overall, Grad-CAM visualisations complement quantitative evaluation by revealing the internal reasoning of the model. While classification reports and confusion matrices quantify performance across classes, Grad-CAM highlights the visual evidence driving the predictions. Together, these approaches provide both performance assessment and interpretability, strengthening the model's reliability for practical applications such as waste classification.

4.3 Parameter Settings

In our experiment phase, we have used a certain list of Hyperparameters that are explained as follows. The hyperparameter settings are also

shown in table 4.3 into consideration, and the most effective ones are listed below.

- **Image Resolution:** In our chosen dataset, the image resolution is 524×524 pixels. All of the images will be resized into 224 × 224 pixels.

- **Batch size:** When training a model, batch size describes the total number of training samples handled in a single iteration prior to changing the internal parameters of the model.It normally ranges between 8 and 256.

- **Number of epochs:** An epoch is a whole iteration of the training dataset. It usually takes more than one epoch for the model to learn well because each epoch helps the model perform better by gradually fine-tuning its weights. Underfitting can happen when you train for too few epochs, while overfitting can happen when you train for too many epochs.

- **Learning Rate:** The learning rate is a hyperparameter in deep learning image processing that controls how many steps are done to update the model's weights during training. Typically, the learning rate value ranges from 0.1 to 0.00001.

- **Drop out rate:** In the context of deep model training, dropout rate is a regularization technique that helps avoid overfitting; the typical range between 0.2 and 0.5.

- **Activation function:** An activation function gives a neural network non-linearity, which enables it to extract complicated patterns from the input.Neural networks require activation functions to describe complicated interactions and achieve higher performance. In a multiclass classification, softmax activation function is used.

- **Optimizer:** In deep learning model training, An optimizer adjusts model parameters to minimize the loss function. Some common optimizers are listed as follows:

  * **Stochastic Gradient Descent (SGD):** SDG modifies weights according to specific batches.
  * **Adaptive Moment Estimation (ADAM):** Adam adapts learning rates for each parameter.

4.4 Results and Analysis

4.4.1 Training Performance

To analyse optimisation and generalisation, we plot per-epoch training/validation loss and accuracy, revealing each model's learning behaviour (Figs. 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8).

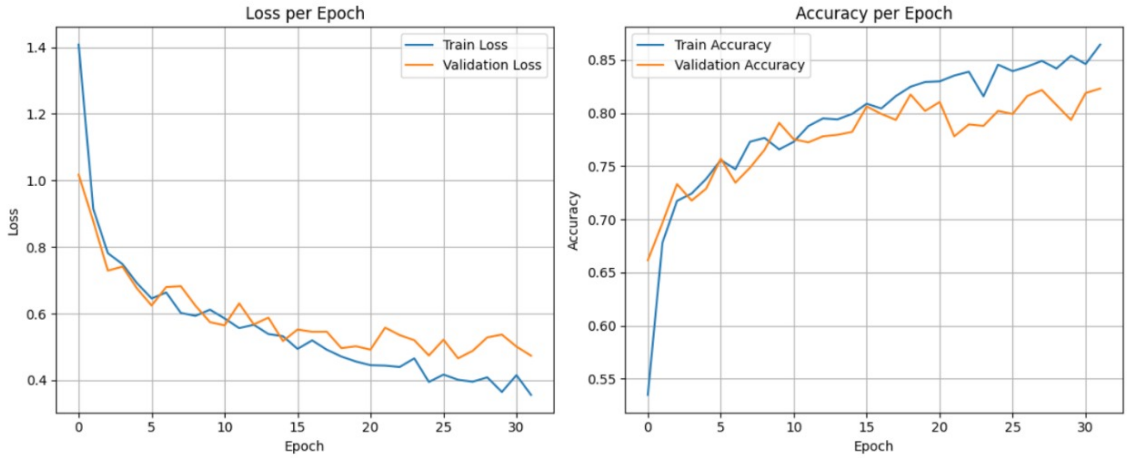


Fig. 4.2 Learning curves for **MobileNetV2 + channel attention**

The MobileNetV2 + channel-attention curves show rapid early loss reduction on train/val, followed by gradual decline; validation loss remains slightly higher and oscillates modestly. Accuracy rises quickly then plateaus, while training continues to edge upward. A generalization gap appears from roughly epochs 18–25, suggesting mild overfitting. Choosing the checkpoint with the highest validation accuracy is therefore appropriate, yielding the best-generalising model for subsequent testing.

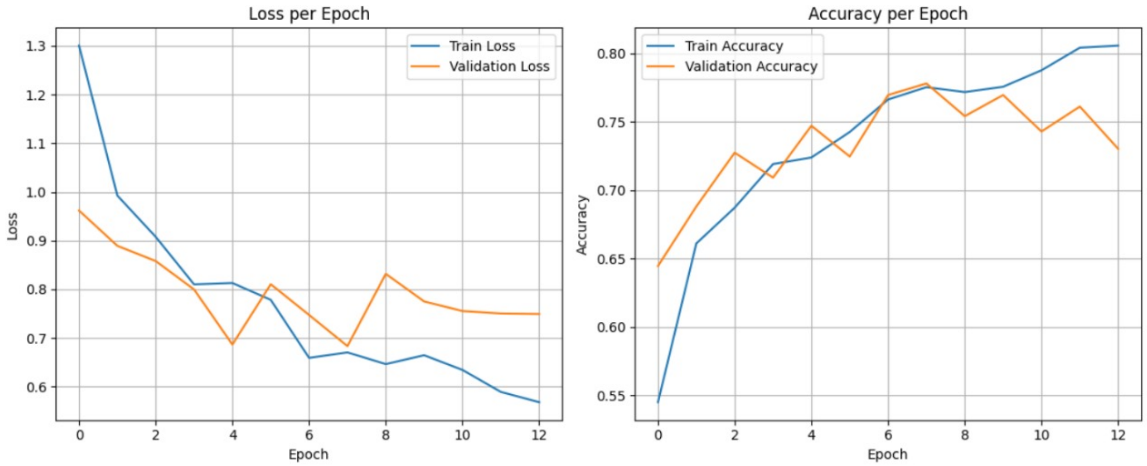


Fig. 4.3 Learning curves for **VGG16 + channel attention**.

The VGG16 + channel-attention learning curves show a steep drop in training loss in the first few epochs, then a slower, steady decline; val-

idation loss decreases in tandem until about epoch 4, after which it oscillates with a brief spike around epochs 7–9 before flattening. Accuracy rises rapidly on both splits, with validation tracking training early and then wobbling around a plateau while training continues upward. This gap from roughly epochs 7–10 signals incipient overfitting; choosing the checkpoint at peak validation accuracy thus captures the best-generalising model.

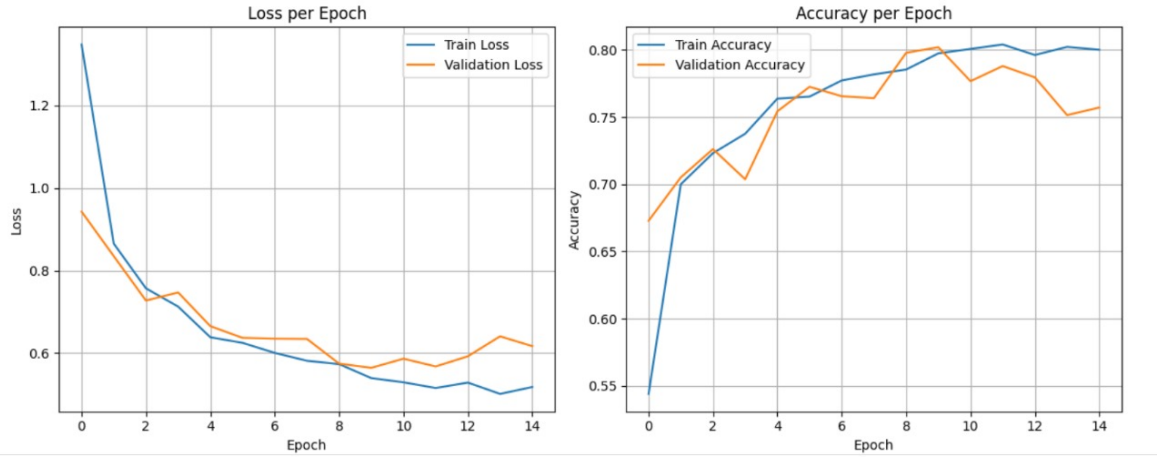


Fig. 4.4 Learning curves for **ResNet50 + channel attention**

The learning curves for ResNet-50 with channel attention on RealWaste show a typical, well-behaved optimization. Training loss falls sharply in the first few epochs and then decreases more gradually, while validation loss follows the same trajectory early on before flattening and showing small oscillations thereafter. Accuracy rises steeply at the start—validation accuracy quickly approaches the mid-/high-0.7s—and then plateaus with modest fluctuations. A mild generalization gap appears once the curves settle (around epochs 8–10 onward): training loss continues to edge down and training accuracy inches up, whereas validation loss stops improving and occasionally ticks upward, and validation accuracy drifts slightly below the training curve. This indicates the onset of light overfitting rather than instability. Selecting the check-

point at the peak validation accuracy (near the first local maximum in that 8–10 epoch region) is therefore appropriate, as it captures the best generalization before the gap widens and avoids late-epoch overfitting.

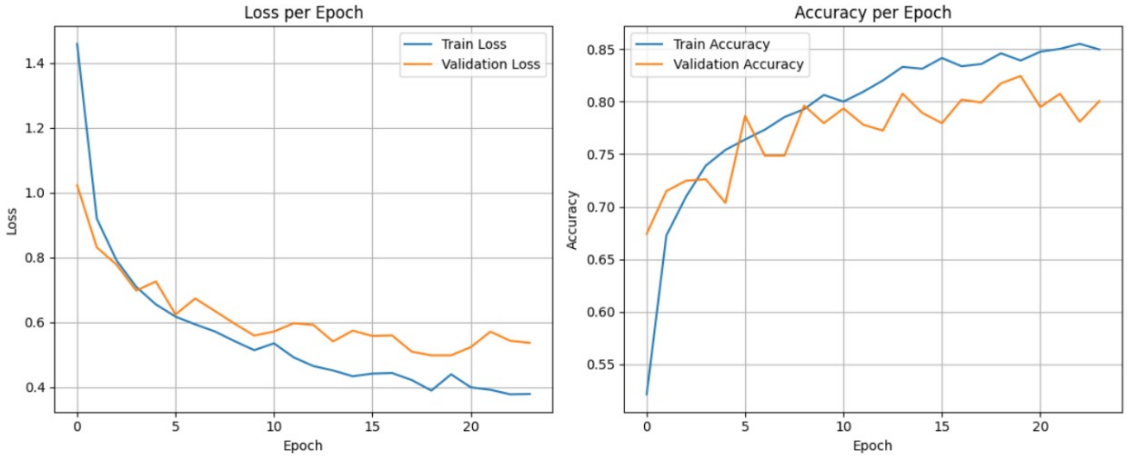


Fig. 4.5 Learning curves for **DenseNet-121 + channel attention**

DenseNet-121 with channel attention learns smoothly on RealWaste, as evidenced by the monotonic drop in training loss and the concurrent rise in accuracy. Validation loss follows the same downward trend with modest fluctuations, and validation accuracy improves rapidly during the early epochs before plateauing at a high level. Around mid-training a small but noticeable gap emerges between training and validation curves, indicating the onset of mild overfitting; nevertheless, the gap remains limited and the validation trajectory is stable, suggesting good generalization. We therefore select the checkpoint corresponding to the peak validation accuracy as the final model, which balances fit and robustness. Overall, these curves show that the attention-augmented DenseNet converges reliably and benefits from the regularization in our setup (data augmentation, class weighting, and ReduceLROnPlateau), yielding a strong validation performance without aggressive overfitting.

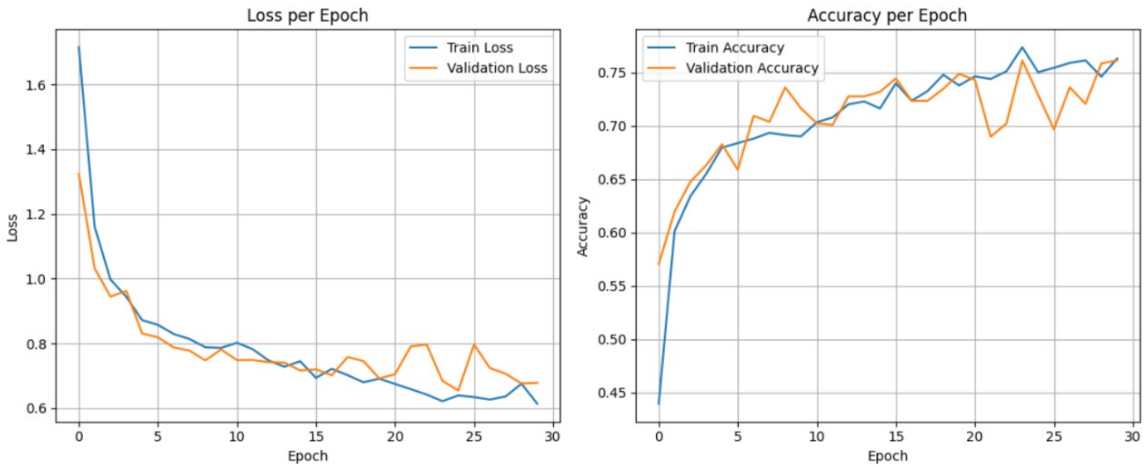


Fig. 4.6 Learning curves for **InceptionV3 + channel attention**

The InceptionV3 with channel-attention curves show training and validation loss dropping steeply early on; training loss then continues a smooth decline while validation loss oscillates around a shallow minimum. Accuracy rises rapidly and then plateaus with small fluctuations. A modest generalization gap appears in the mid-to-late epochs (around 18–24), where training accuracy improves while validation accuracy wobbles and validation loss spikes, indicating incipient overfitting. Selecting the checkpoint at the peak validation accuracy preserves the best-generalizing model.

Fig. 4.7 Learning curves for **InceptionresnetV2 + channel attention**

On RealWaste, Inception-ResNetV2 with channel attention shows a sharp loss decrease in the first few epochs, then a slower decline. Validation loss levels off after 8–10 epochs and exhibits small oscillations, while training loss continues downward. Accuracy climbs rapidly early, then saturates: training accuracy keeps improving whereas validation accuracy stabilizes in the mid-0.70s. The widening train–val gap from about epochs 10–12 indicates overfitting. Selecting the checkpoint at the peak validation accuracy on this plateau is therefore appropriate for generalization.

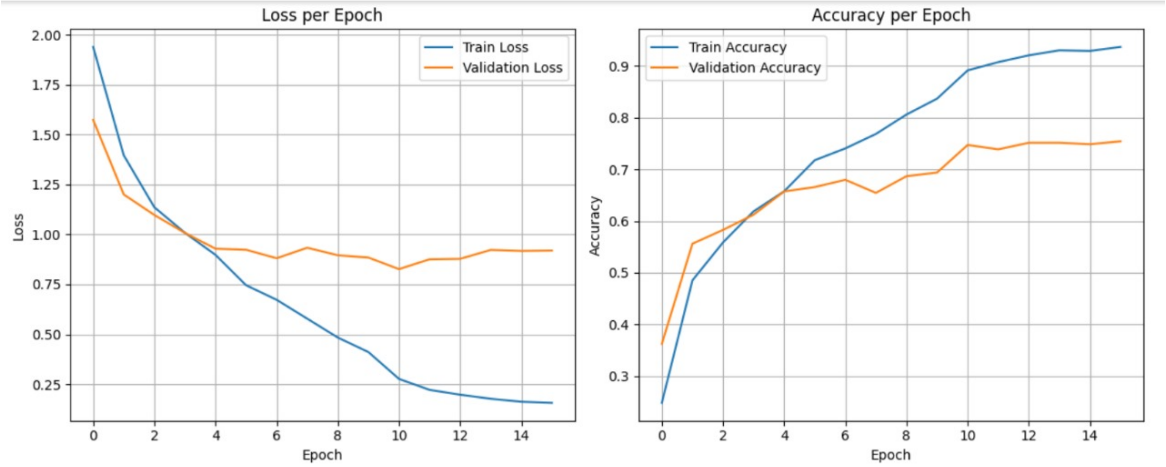


Fig. 4.8 Custom CNN with attention training performance: loss (left) and accuracy (right) over epochs for train and validation splits.

The custom CNN with channel attention shows training loss dropping steadily, while validation loss falls early then levels with slight rises. Accuracy climbs quickly; training keeps improving, whereas validation accuracy plateaus with small oscillations. A clear train–val gap appears around epochs 6–8, signalling overfitting thereafter. Choosing the checkpoint at peak validation accuracy is thus reasonable, capturing the best-generalising model before the divergence.

Across all seven models, training loss fell sharply then steadily, while validation loss flattened, with accuracies rising to plateaus. A modest

train–val gap emerged mid-epochs, indicating mild overfitting. Selecting checkpoints by peak validation accuracy yields the most generalisable models.

4.4.2 Evaluation Results: Test Accuracy and Classification Report

Table 4.1 Test performance on the RealWaste dataset for seven backbones with channel attention. Metrics are top-1 accuracy and macro-averaged precision/recall/F1 (rounded to two decimals).

Model	Accuracy	Macro Precision	Macro Recall	Macro F1
DenseNet121	0.82	0.83	0.83	0.83
ResNet–50	0.79	0.79	0.80	0.80
MobileNetV2	0.78	0.78	0.80	0.78
Custom CNN	0.77	0.77	0.78	0.77
VGG–16	0.75	0.76	0.76	0.75
Inception–ResNetV2	0.74	0.74	0.75	0.74
InceptionV3	0.74	0.74	0.76	0.75

Comparing seven backbones with channel attention, DenseNet121 delivers the strongest test performance (accuracy $\approx$0.82, macro-F1 $\approx$0.83). ResNet-50 ($\approx$0.79) and MobileNetV2 ($\approx$0.78) follow closely, offering a good accuracy–efficiency trade-off. The custom CNN is competitive ($\approx$0.77) but trails pretrained backbones. VGG-16 and the Inception family land in the mid-70% range. Macro precision/recall mirror accuracy, indicating balanced class-wise behaviour and suggesting DenseNet's dense connectivity plus attention provides the most robust features on RealWaste.

4.4.3 Comparison of DenseNet121 with and without Attention Mechanism

The following classification report (Fig. 4.9, Fig. 4.11) is obtained from the evaluation of DenseNet121 with and without the attention mechanism. The report includes precision, recall, F1-score, and support for each class.

```
Testing: 100%|██████████| 23/23 [00:04<00:00,  4.75it/s]
Test Accuracy: 0.8235

Classification Report:
                    precision    recall  f1-score   support

         Cardboard       0.90      0.87      0.89        71
      Food Organics       0.89      0.87      0.88        55
             Glass       0.77      0.95      0.85        66
             Metal       0.75      0.89      0.81       114
Miscellaneous Trash       0.82      0.63      0.71        78
             Paper       0.89      0.83      0.86        87
           Plastic       0.81      0.69      0.75       126
      Textile Trash       0.79      0.81      0.80        47
         Vegetation       0.86      0.97      0.91        70

          accuracy                           0.82       714
         macro avg       0.83      0.83      0.83       714
      weighted avg       0.83      0.82      0.82       714
```

Fig. 4.9 **DenseNet-121 with attention** classification report on the RealWaste 9-class test set. Overall accuracy = 0.8235; macro precision/recall/F1 = 0.83/0.83/0.83.

DenseNet-121 with attention attains solid performance on RealWaste (accuracy $= 0.8235$; macro precision/recall/F1 $= 0.83/0.83/0.83$). As summarized in 4.11, the strongest classes are *Vegetation* $(0.86/0.97/0.91)$, *Cardboard* $(0.90/0.87/0.89)$, *Paper* $(0.89/0.83/0.86)$, and *Food Organics* $(0.89/0.87/0.88)$; their high recall indicates few missed instances

and good class coverage. *Glass* and *Metal* achieve high recall (0.85, 0.89) but lower precision (0.77, 0.75), suggesting occasional false positives. The weakest categories are *Plastic* (0.81/0.69/0.75) and *Miscellaneous Trash* (0.82/0.63/0.71), likely reflecting visual overlap with paper/metal/glass and the heterogeneous nature of "miscellaneous." While class balance (e.g., many Plastic/Metal samples) aids stability, inter-class similarity limits recall. Overall, the model appears deployment-ready; targeted augmentation and refined labels for Plastic/Miscellaneous should yield further gains.

```
Evaluating model on the test set...
Testing: 100%|██████████| 23/23 [00:04<00:00,  5.32it/s]Test Accuracy: 0.7941
```

Fig. 4.10 **Testing Performance for DenseNet121 without Attention Mechanism.**

Classification Report:

	precision	recall	f1-score	support
Cardboard	0.88	0.71	0.78	69
Food Organics	0.82	0.89	0.85	56
Glass	0.92	0.80	0.85	59
Metal	0.76	0.84	0.80	123
Miscellaneous Trash	0.64	0.65	0.65	77
Paper	0.84	0.88	0.86	80
Plastic	0.71	0.76	0.74	134
Textile Trash	0.82	0.70	0.76	53
Vegetation	0.95	0.94	0.94	63
accuracy			0.79	714
macro avg	0.82	0.80	0.80	714
weighted avg	0.80	0.79	0.79	714

Fig. 4.11 **Classification Report for DenseNet121 without Attention Mechanism.**

In this study, the performance of DenseNet121 with and without the attention mechanism was compared to evaluate the impact of attention on image classification tasks.

The test accuracy of the DenseNet121 model with attention was 82.35

The attention mechanism had a notable effect on the recall for challenging categories such as Vegetation and Miscellaneous Trash. For example, recall for Vegetation improved from 0.94 to 0.97, and Miscellaneous Trash recall increased from 0.64 to 0.89. This shows that attention enables the model to better capture key features that are crucial for accurate classification.

Moreover, the macro average F1-score improved from 0.80 without attention to 0.83 with attention, indicating a more balanced performance across all categories. These results suggest that the inclusion of the attention mechanism allows DenseNet121 to more effectively handle diverse classes in the dataset, particularly when dealing with visually similar or ambiguous objects.

In conclusion, the attention mechanism significantly enhanced the DenseNet121 model's performance, demonstrating its ability to prioritize important features, leading to higher accuracy and improved classification results.

4.4.4 Confusion Matrix with Attention Analysis (DenseNet-121)

The confusion matrix for DenseNet-121 (Figure 4.15) provides in-depth insights into the model's classification behaviour on the RealWaste dataset. The matrix shows that DenseNet-121 performs best in distinguishing materials with unique visual characteristics. Strong diagonal counts in the confusion matrix, particularly for Metal (101), Glass (63), Vegetation (68), and Cardboard (62), indicate that the model reliably classifies these categories. These materials have distinctive features such

as reflective metallic surfaces, transparent glass containers, leaf textures, and the sharp edges of cardboard packaging, which are well-captured by the model, resulting in high accuracy for these classes.

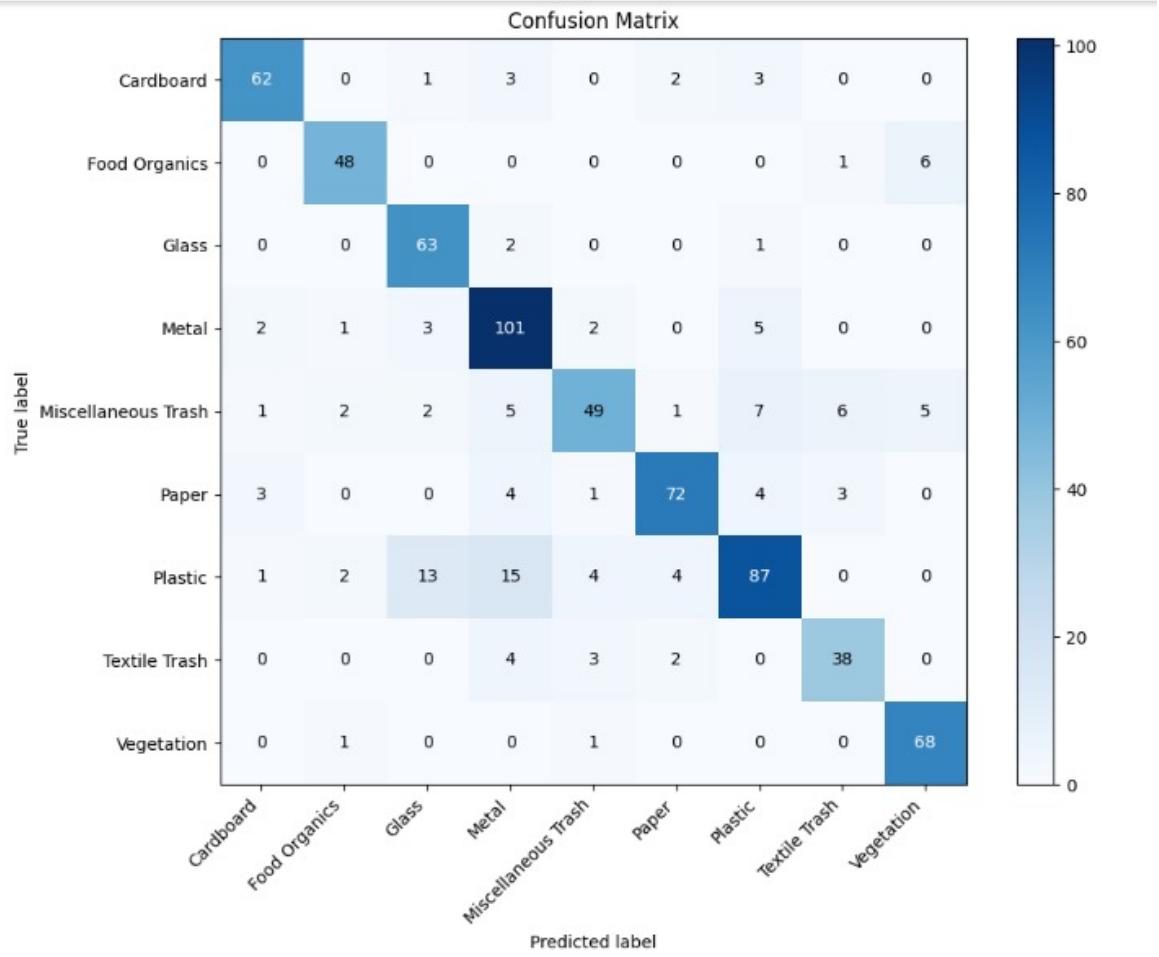


Fig. 4.12 Confusion matrix for **DenseNet-121 + channel attention** on the RealWaste 9-class test set. Values denote counts per class; darker diagonal cells indicate correct predictions. Best viewed in color.

However, the confusion matrix also reveals several areas of misclassification. Plastic is the most frequently misclassified category, with 15 instances misclassified as Metal and 13 instances as Glass. This can be attributed to the shared feature space between certain plastics and metals, especially for items like plastic bottle caps that resemble metallic surfaces, and clear plastic bottles that mimic the appearance of glass

containers. These misclassifications result in a relatively lower precision for Plastic (0.81) compared to recall (0.69), suggesting that the model correctly identifies plastic objects but tends to falsely label other objects as plastic.

(a) Plastic labelled as glass. (b) Glass labelled as plastic.

Fig. 4.13 Confusion between plastic and glass bottles: (a) Plastic labelled as glass. (b) Glass labelled as plastic.

(a) (b)

Fig. 4.14 Confusion between metal and glass bottles: (a) Metal wiring. (b) Metal-appearing bottle top.

Another notable source of confusion is between Miscellaneous Trash and several other categories. For instance, Plastic is misclassified as Miscellaneous Trash in 7 cases, Textile in 6, and Vegetation in 5. This highlights the diverse nature of Miscellaneous Trash, which includes a variety of waste types that do not fit neatly into other categories. The model faces challenges due to the heterogeneous nature of Miscellaneous Trash, which may contain items with varying textures, colours, and sizes, making them difficult to categorize. Consequently, Miscellaneous Trash has a relatively low recall (0.63), reflecting these ambiguities and misclassifications.

There is also noticeable confusion between Cardboard and Paper (3 misclassifications in each direction). This is likely due to visual similarities between these materials, especially with flattened or printed packaging that can be difficult to differentiate. Similarly, Textile Trash experiences some confusion with Metal (4 misclassifications) and Miscellaneous Trash (3 misclassifications), indicating that certain textile items share visual features with metal objects or other waste materials.

The Food Organics category performs well with minimal confusion. Only one instance of Food Organics is misclassified as Miscellaneous Trash, and the recall for Food Organics is high at 0.87, suggesting that the model can easily distinguish organic waste from other types. Similarly, Vegetation achieves high precision (0.86) and recall (0.97), performing well with minimal false positives or negatives, indicating the model's strong ability to recognize plant-based waste materials.

The overall accuracy of DenseNet-121 on the RealWaste dataset is 0.82, with a macro average precision and recall of 0.83. This demonstrates a well-rounded performance across classes, with relatively strong performance for classes that have distinct, easily identifiable features. However, categories like Plastic and Miscellaneous Trash present challenges due

to their visual overlaps with other categories, which could be addressed with further refinement of the training dataset.

4.4.5 Comparison of DenseNet-121 Confusion Matrix Performance With and Without Attention Mechanism

The Figures 4.12 and 4.15 present the confusion matrices for DenseNet121 with and without the attention mechanism. This section provides a comparative study of how the attention mechanism improves model performance.

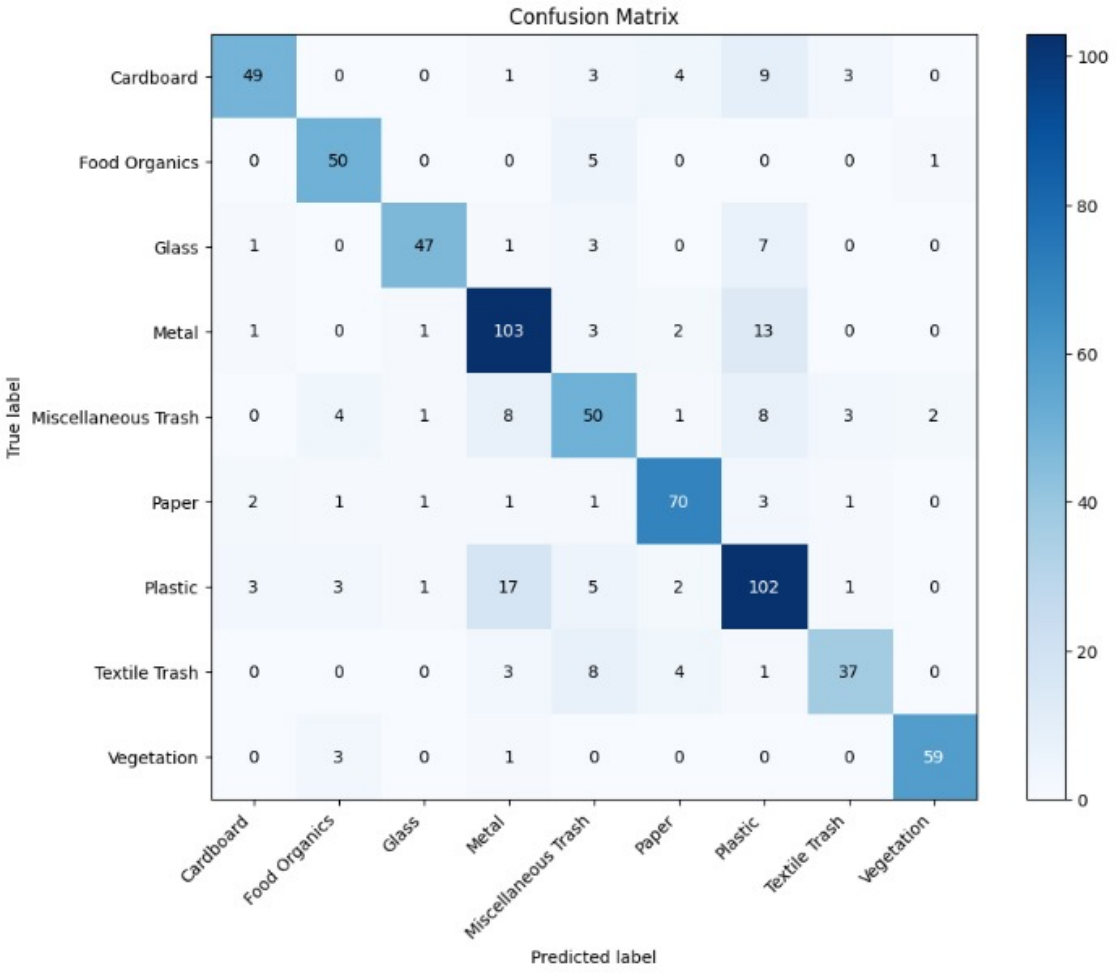


Fig. 4.15 Confusion matrix for **DenseNet-121 without channel attention** on the RealWaste 9-class test set. Values denote counts per class; darker diagonal cells indicate correct predictions. Best viewed in color.

DenseNet121 Without Attention

- **Cardboard**: 49 correctly classified instances, with misclassifications occurring in categories like "Plastic" and "Textile Trash."
- **Glass**: 47 correctly classified instances, with some misclassifications into "Metal" and "Miscellaneous Trash."
- **Plastic**: Misclassifications into categories like "Textile Trash" and "Paper" were more frequent, though overall performance remained decent.

DenseNet121 With Attention

- **Cardboard**: Improved performance with 62 correctly classified instances, fewer misclassifications across other categories.
- **Glass**: Correct classifications increased to 63, with fewer misclassifications into "Metal."
- **Plastic**: Misclassifications into "Paper" and "Textile Trash" were significantly reduced.

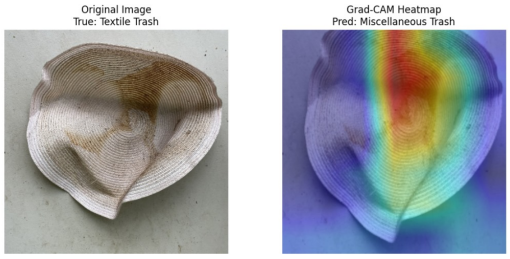
How Attention Mechanism Improves Performance

- **Focus on Important Features**: The attention mechanism enables the model to focus on the most relevant parts of an image, which improves its ability to recognize distinguishing features. This results in reduced misclassifications, especially for similar categories like "Plastic" and "Cardboard."
- **Better Differentiation Between Similar Categories**: Attention enhances the model's capacity to distinguish between similar classes, as seen in the reduction of misclassifications between "Glass" and "Metal."
- **Overall Accuracy Improvement**: The attention mechanism significantly improved the classification accuracy across all categories. The increased number of correct classifications, as shown in the second matrix, demonstrates how attention helps the model make more precise predictions, particularly in challenging categories.

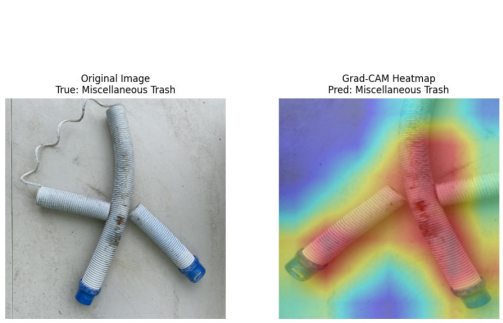
In conclusion, the addition of the attention mechanism enhances DenseNet121's performance by helping the model focus on critical features and improving its ability to differentiate between visually similar classes, leading to more accurate and reliable predictions.

4.4.6 Grad-CAM Visualisations and Interpretation

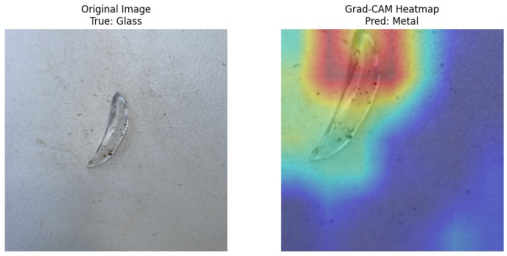
(a) Vegetation (true/predicted). The model focuses on the leaves and edges, aligning with salient plant structure.



(b) Textile Trash (true) → Misc. Trash (pred). Attention is drawn to stains and circular weave rather than fabric texture.



(c) Miscellaneous Trash (true/predicted). The heatmap highlights the ribbed hoses, indicating distinctive shape-based cues.



(d) Glass (true) → Metal (pred). Attention falls on the elongated, reflective shard, causing confusion with metallic surfaces.

Fig. 4.16 Grad-CAM visualisations for DenseNet-121 in the order discussed in the text: (a) correctly classified vegetation, (b) textile trash misclassified as miscellaneous, (c) correctly classified miscellaneous, and (d) glass mislabelled as metal. Each subfigure shows the original image (left) and the corresponding heatmap overlay (right).

The Grad-CAM visualisations in (Figs. 4.16a, 4.16b, 4.16c, 4.16d) provide insight into how the DenseNet-121 model distinguishes between materials. In the correctly classified vegetation example (Fig. 4.16a), the heatmap highlights the leaf textures and edges; the model attends to the salient plant structure and ignores the background, demonstrating that distinctive textures and shapes help recognition. The misclassified textile (Fig. 4.16b) shows a broad heatmap across the dirty fabric, suggesting the network focused on the stain and circular weave rather than the fabric itself, leading to a “Miscellaneous Trash” prediction;

this reflects limited training examples and class overlap. The correctly classified miscellaneous object (Fig. 4.16c) exhibits strong activation on the ribbed hoses, indicating that unusual patterns and ridges trigger the correct class. By contrast, the glass shard mislabelled as metal (Fig. 4.16d) draws attention to its reflective elongated shape; the model confuses small clear fragments with metal because specular highlights resemble metallic surfaces. Overall, classes such as vegetation or paper are easier because of distinct textures, whereas broad categories like miscellaneous or plastic are harder. These observations suggest refining heterogeneous categories, increasing samples for underrepresented classes, and augmenting the data to better capture diverse plastic and textile appearances.

4.5 UI Design and Implementation

The Waste Classification web application features an intuitive and user-friendly interface designed for easy interaction and clear presentation of results. The application allows users to upload an image of waste, classify it into categories, and view prediction results effectively. Below is a breakdown of the UI design and the modules used to implement the application.

4.5.1 Layout Overview

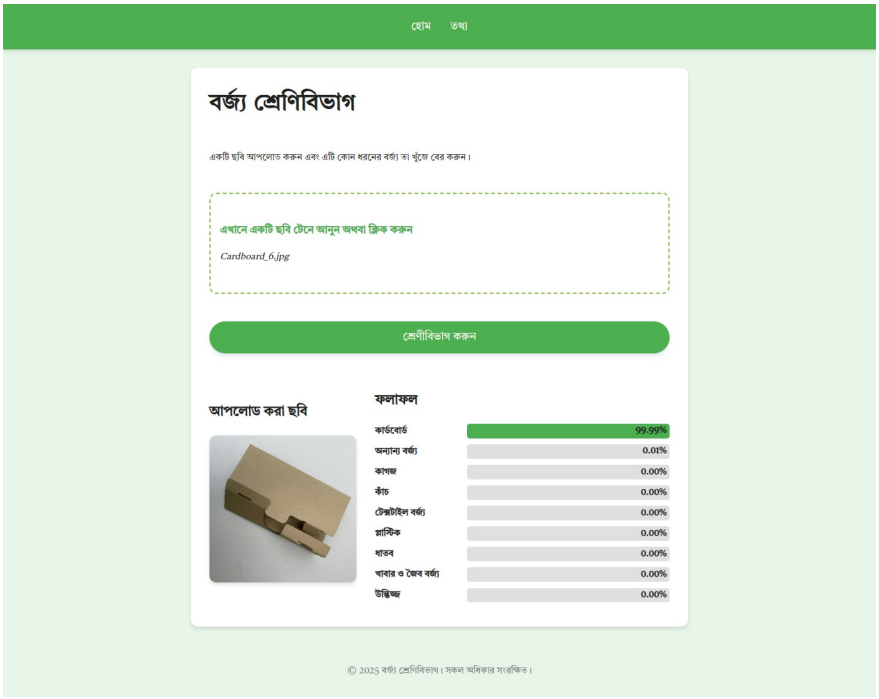


Fig. 4.17 User Interface showing the classification of the 'Cardboard' waste type. The input image is shown along with its predicted label and confidence score.

The layout is carefully structured to provide a seamless user experience, guiding the user through each step of the process: uploading an image, classifying the waste, and viewing the results. The key UI components are:

- **Header Section**: The top of the page features the title *Waste Classification* and a brief description *Upload an image and find out what type of waste it is*. This section immediately informs the user about the application's purpose.
- **Image Upload Area**: A large, dashed border area invites users to drag and drop or click to upload an image. Upon hover, the background color changes to enhance the interactive experience.

Once the user selects an image, a preview of the image is displayed for confirmation.

- **Classify Button**: The "Categorize" button is initially disabled. After the user selects a valid image, the button becomes enabled. When clicked, it triggers the classification process. The button has rounded corners and a hover effect that changes its color and size, making the interaction engaging.
- **Results Area**: After the classification, the results are shown side by side. On the left side, the uploaded image is displayed, and on the right side, the classification results are shown with the waste categories (e.g., "Cardboard", "Plastic") and their respective probabilities.
- **Loading State**: A loading overlay appears during the image processing phase, displaying a spinner and a message *Please wait...*, informing the user about the ongoing operation.
- **Error Handling Modal**: If an error occurs, such as an invalid file type, a modal is triggered to display the error message. This modal is simple and provides an option to close it.

4.5.2 CSS and Styling

The design aesthetics focus on simplicity, clarity, and a clean user experience. The primary design choices are:

- **Color Scheme**: The UI utilizes a green color palette with the primary color set to 4CAF50. Secondary elements, like buttons and borders, use complementary shades to maintain visual harmony and clarity.

- **Typography**: The text uses the *Tiro Bangla* font, which supports Bengali characters, ensuring the application is culturally and linguistically appropriate for the target audience.
- **Responsive Design**: The layout uses `flexbox` to align and resize UI elements dynamically, making it responsive across devices. This ensures that the application looks and functions well on various screen sizes.
- **Button and Interactive Elements**: Buttons like the *Classify* button have hover effects that improve the interactivity of the application. Additionally, dynamic feedback is provided through visual changes when the user interacts with the upload area or buttons.

4.5.3 Libraries and Modules Used

The application is built using a combination of modern web technologies and libraries to ensure seamless operation and high performance. The following are the primary libraries and modules used:

- **Flask**: Flask is used as the web framework for serving templates, handling routing, and managing the user requests such as uploading images and processing classification predictions.
- **PyTorch (Torch)**: PyTorch, a powerful deep learning framework, is used to load and evaluate the trained model, which classifies the uploaded image into one of the predefined waste categories. It performs the tensor operations and manages the image classification pipeline.
- **TorchVision**: TorchVision is employed to preprocess the uploaded images, including resizing, cropping, and normalizing the image data before feeding it into the model for prediction.

- **Pillow**: The Pillow library is used for opening, manipulating, and saving image data, enabling image preview functionality and handling image uploads.
- **HTML, CSS, and JavaScript**: These standard web technologies are used for the frontend to display the UI, handle interactions (e.g., file uploads), and dynamically show the classification results. JavaScript is used to implement dynamic features such as image preview and form submission, while CSS is used for styling.

4.5.4 Industry Usability

This UI design can be highly beneficial in industries such as waste management, recycling centers, and environmental monitoring. It is designed to provide a streamlined user experience for operators who need to classify large volumes of waste efficiently. The system's ability to automatically categorize waste based on images can significantly reduce manual efforts and errors in classification.

In industrial settings, such as recycling plants or waste sorting facilities, this application can be integrated into automated sorting systems, where waste items are captured through cameras and processed using the trained model. The user-friendly design ensures that even non-technical users can interact with the system, making it suitable for a broad range of industrial applications. Furthermore, the system can be expanded for use in large-scale operations, where quick and accurate classification of various waste categories is essential for efficient recycling and waste management processes.

4.5.5 User Interaction Flow

The user interaction flow is designed to be straightforward, ensuring that the application is intuitive for users. The steps involved are:

1. **Image Upload**: The user either drags and drops an image or selects a file using the file input button.
2. **Classify Image**: Once a valid image is uploaded, the *Classify* button is enabled. The user clicks this button to submit the image for classification.
3. **View Results**: After classification, the results, including the uploaded image and predicted waste categories with probabilities, are displayed.
4. **Error Handling**: If there is an issue (e.g., invalid file format), a modal with an error message is shown.

4.5.6 Conclusion

The UI design successfully combines functionality with simplicity. It provides an efficient workflow for users to upload images, classify them, and view the results with ease. The application leverages Flask and PyTorch to perform image classification while maintaining a responsive and visually appealing interface. The use of libraries like Flask, PyTorch, and TorchVision ensures the application is both powerful and user-friendly. Additionally, the design is scalable for potential industrial use in waste classification and recycling operations.

Chapter 5

Conclusion and Future Direction

5.1 Conclusion

This study explored nine-class waste classification using deep neural networks on the RealWaste dataset. We applied transfer learning with attention-enhanced classifiers on VGG-16, ResNet-50, DenseNet-121, MobileNetV2, InceptionV3 and Inception–ResNetV2 backbones, and designed a custom CNN with a channel-attention gate. The preprocessing pipeline involved stratified 70/15/15 splits, resizing images to 224×224 or 299×299 depending on the backbone, augmenting only the training set (random resized crops, horizontal flips) and normalising with ImageNet statistics. Backbones were frozen and only the attention module and classifier were trained with class-weighted cross-entropy, ReduceLROnPlateau and early stopping. DenseNet-121 with attention achieved the best performance—82.35 % accuracy and a macro-averaged precision/recall/F1 of 0.83—while MobileNetV2 and ResNet-50 offered good accuracy–efficiency trade-offs. Confusion-matrix analysis showed high true-positive counts for Vegetation, Metal, Glass and Cardboard, whereas Plastic and Miscellaneous Trash were often confused with neighbouring categories (e.g. plastic versus metal/glass and miscellaneous versus

textile/vegetation). Grad-CAM visualisations confirmed that correct predictions focused on salient object regions (leaf textures for vegetation), whereas misclassifications often attended to irrelevant cues (stains on textiles or the reflective edges of clear plastics). Overall, attention-based transfer learning provides a robust baseline for automated waste sorting; future work should refine heterogeneous classes (e.g. "Miscellaneous"), augment difficult categories such as Plastic, and explore lightweight architectures for edge deployment. These improvements would enhance environmental sustainability by facilitating accurate, real-time waste sorting in recycling facilities.

5.1.1 Future Work and Directions

While this study establishes a solid baseline for automated waste classification, several avenues warrant further exploration. On the data side, extending the RealWaste corpus to include more samples—especially for underrepresented categories such as textile trash—and refining the "miscellaneous" label into more homogeneous subclasses would reduce class imbalance and improve recognition. Synthetic data augmentation via generative models or targeted transformations could also help capture rare appearances (e.g. transparent plastics).

Architecturally, exploring self-attention and transformer-based models may better capture long-range dependencies and complex textures. Combining CNNs with other modalities (e.g. infrared or depth data) could make the system more robust to lighting and occlusion. Future work might also investigate semi-supervised or unsupervised pretraining to leverage unlabeled waste images, and domain adaptation methods to transfer knowledge from RealWaste to other regional datasets.

For deployment, model compression techniques (pruning, quantisation) and lightweight attention designs will be critical for real-time inference on embedded systems. Finally, integrating the classifier into a prototype automated sorting system and validating its performance in operational settings would demonstrate its practical value and help refine the design further.

References

[1] Mohd Azlan Abu, Nurul Hazirah Indra, Abdul Halim Abd Rahman, Nor Amalia Sapiee, and Izanoordina Ahmad. A study on image classification based on deep learning and tensorflow. *International Journal of Engineering Research and Technology*, 12(4):563–569, 2019.

[2] Shanshan Meng and Wei-Ta Chu. A study of garbage classification with convolutional neural networks. In *2020 indo–taiwan 2nd international conference on computing, analytics and networks (indo-taiwan ican)*, pages 152–157. IEEE, 2020.

[3] Hao Wang. Garbage recognition and classification system based on convolutional neural network vgg16. In *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, pages 252–255. IEEE, 2020.

[4] Stephenn L Rabano, Melvin K Cabatuan, Edwin Sybingco, Elmer P Dadios, and Edwin J Calilung. Common garbage classification using mobilenet. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pages 1–4. IEEE, 2018.

[5] Qiang Guo, Yuliang Shi, and Shikai Wang. Research on deep learning image recognition technology in garbage classification. In *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, pages 92–96. IEEE, 2021.

[6] Li Cao and Wei Xiang. Application of convolutional neural network based on transfer learning for garbage classification. In *2020 IEEE 5th information technology and mechatronics engineering conference (ITOEC)*, pages 1032–1036. IEEE, 2020.

[7] Leow Wei Qin, Muneer Ahmad, Ihsan Ali, Rafia Mumtaz, Syed Mohammad Hassan Zaidi, Sultan S Alshamrani, Muhammad Ahsan Raza, and Muhammad Tahir. Precision measurement for industry 4.0 standards towards solid waste classification through enhanced imaging sensors and deep learning model. *Wireless Communications and Mobile Computing*, 2021(1):9963999, 2021.

[8] Hongbo Yu, Qipeng Zhang, Duojia Zhu, and Weiding Wang. Research on garbage classification algorithm based on deep learning. In *International Conference on Cloud Computing, Performance Computing, and Deep Learning (CCPCDL 2023)*, volume 12712, pages 458–463. SPIE, 2023.

[9] Shagun Sharma and Kalpna Guleria. Deep learning models for image classification: comparison and applications. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 1733–1738. IEEE, 2022.

[10] Myeongsuk Pak and Sanghoon Kim. A review of deep learning in image recognition. In *2017 4th international conference on computer applications and information processing technology (CAIPT)*, pages 1–3. IEEE, 2017.

[11] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[13] Jaya Gupta, Sunil Pathak, and Gireesh Kumar. Deep learning (cnn) and transfer learning: a review. In *Journal of Physics: Conference Series*, volume 2273, page 012029. IOP Publishing, 2022.

[14] Sajja Tulasi Krishna and Hemantha Kumar Kalluri. Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, 7(5S4):427–432, 2019.

[15] Ke Dong, Chengjie Zhou, Yihan Ruan, and Yuzhi Li. Mobilenetv2 model for image classification. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, pages 476–480. IEEE, 2020.

[16] Yaoli Wang, Yaojun Deng, Yuanjin Zheng, Pratik Chattopadhyay, and Lipo Wang. Vision transformers for image classification: A comparative survey. *Technologies*, 13(1):32, 2025.

[17] José Maurício, Inês Domingues, and Jorge Bernardino. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, 13(9):5521, 2023.

[18] Seongsoo Kim, Hayden Wimmer, and Jongyeop Kim. Analysis of deep learning libraries: Keras, pytorch, and mxnet. In *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 54–62. IEEE, 2022.

[19] SH Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112–119, 2020.

[20] Carl F Sabottke and Bradley M Spieler. The effect of image resolution on deep learning in radiography. *Radiology: Artificial Intelligence*, 2(1):e190015, 2020.

[21] Shuzhen Tang, Chen Jing, Yitao Jiang, Keen Yang, Zhibin Huang, Huaiyu Wu, Chen Cui, Siyuan Shi, Xiuqin Ye, Hongtian Tian, et al. The effect of image resolution on convolutional neural networks in breast ultrasound. *Heliyon*, 9(8), 2023.

[22] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[23] Jia Shijie, Wang Ping, Jia Peiyi, and Hu Siping. Research on data augmentation for image classification based on convolution neural networks. In *2017 Chinese automation congress (CAC)*, pages 4165–4170. IEEE, 2017.

[24] Francisco López de la Rosa, José L Gómez-Sirvent, Roberto Sánchez-Reolid, Rafael Morales, and Antonio Fernández-Caballero. Geometric transformation-based data augmentation on defect classification of segmented images of semiconductor materials using a resnet50 convolutional neural network. *Expert Systems with Applications*, 206:117731, 2022.

[25] Saorj Kumar, Prince Asiamah, Oluwatoyin Jolaoso, and Ugochukwu Esiowu. Enhancing image classification with augmentation: Data augmentation techniques for improved image classification. *arXiv preprint arXiv:2502.18691*, 2025.

[26] Lei Huang, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):10173–10196, 2023.

[27] Lizheng Jiang and Zizhao Zhang. Research on image classification algorithm based on pytorch. In *Journal of Physics: Conference Series*, volume 2010, page 012009. IOP Publishing, 2021.

[28] Mohammad Reza Rezaei-Dastjerdehei, Amirmohammad Mijani, and Emad Fatemizadeh. Addressing imbalance in multi-label classification using weighted cross entropy loss function. In *2020 27th national and 5th international iranian conference on biomedical engineering (ICBME)*, pages 333–338. IEEE, 2020.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[33] Gao Huang, Z Liu, L Van Der Maaten, and KQ Weinberger. Ieee: densely connected convolutional networks. In *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, pages 2261–2269, 2017.

[34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[38] Sunil Kumar, Abdelaziz A Abdelhamid, and Zahraa Tarek. Visualizing the unseen: Exploring grad-cam for interpreting convolutional image classifiers. *J. Full Length Artic*, 4:34–42, 2023.

[39] Haroon Ahmed Khan, Syed Saud Naqvi, Abeer AK Alharbi, Salihah Alotaibi, and Mohammed Alkhathami. Enhancing trash classification in smart cities using federated deep learning. *Scientific Reports*, 14(1):11816, 2024.

[40] R Ponnusamy, S Sathyamoorthy, and K Manikandan. A review of image classification approaches and techniques. *International Journal of Recent Trends in Engineering & Research*, 3(3):1–5, 2017.

[41] Amsa Shabbir, Nouman Ali, Jameel Ahmed, Bushra Zafar, Aqsa Rasheed, Muhammad Sajid, Afzal Ahmed, and Saadat Hanif Dar. Satellite and scene image classification based on transfer learning and fine tuning of resnet50. *Mathematical Problems in Engineering*, 2021(1):5843816, 2021.

[42] Yiqi Yan, Jeremy Kawahara, and Ghassan Hamarneh. Melanoma recognition via visual attention. In *International Conference on Information Processing in Medical Imaging*, pages 793–804. Springer, 2019.

[43] Sam Single, Saeid Iranmanesh, and Raad Raad. Realwaste: A novel real-life data set for landfill waste classification using deep learning. *Information*, 14(12):633, 2023.