C O V E N T R Y
U N I V E R S I T Y

Faculty of Engineering, Environment and Computing

School of Computing, Electronics and Mathematics

MSc. Data Science

7150CEM

Data Science Research Project

Improving Convolutional Neural Network - Based Waste Classification Through Data Augmentation and Synthetic Data Generation.

Author: Blessing Oriaguaman

SID: 14196955

1st Supervisor: Dr. Peter Every
2nd Supervisor: Dr. Mahmoud Awad

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science

Academic Year: 2023/24

Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of Copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: Date: 30 July 2024

Please complete all fields

First Name:	**Blessing**
Last Name:	**Oriaguaman**
Student ID number	**14196955**
Ethics Application Number	**P177890**
1st Supervisor Name	**Dr. Peter Every**
2nd Supervisor Name	**Dr. Mahmoud Awad**

This form must be completed, scanned, and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking.

Abstract

The global increase in waste production has necessitated efficient and cost-effective waste management solutions. Traditional waste sorting methods, such as those employed by Wolverhampton City Council, are often inefficient and expensive. This project investigates the enhancement of Convolutional Neural Network (CNN)-based waste classification through data augmentation and synthetic data generation to address these challenges. By improving the accuracy and robustness of CNN models in identifying diverse waste types, this study aims to contribute to more sustainable waste management practices. This project Implemented various data augmentation techniques and employing modified lightweight Generative Adversarial Networks (GANs) to generate synthetic waste images, to enhance the training datasets and addresses the limitations of existing models. Experimental results demonstrate that the augmented and synthetic data significantly improve the models' performances. The VGG16 model's accuracy increased from 63.30% to 78.28%, the ResNet50 model's accuracy improved from 79.63% to 80.47%, the MobileNetV2 model's accuracy increased from 35.94% to 83.18%, and the InceptionV3 model's accuracy improved from 69.19% to 73.92%. Notably, the MobileNetV2 model, after augmentation, achieves the highest accuracy of 83.18%, showcasing its potential for real-world waste classification applications. This project contributes to the growing body of knowledge on AI-driven waste management solutions. Findings suggest that adopting CNN-based waste classification can reduce operational costs and enhance the overall efficiency of waste management processes, aligning with sustainability goals.

Keywords: Convolutional Neural Networks, Waste Classification, Data Augmentation, Synthetic Data Generation, Generative Adversarial Networks, Sustainability

Table of Contents

Table of Figures

Table of Tables

Acknowledgements

I thank God for granting me the strength to complete this dissertation. My deepest gratitude goes to my supervisor, Dr. Peter Every, for his invaluable guidance and constructive feedback. I also appreciate the entire course faculty for their teaching and assistance. Thank you all for your contributions.

1. INTRODUCTION

According to the British Broadcasting Commission (BBC), United Kingdom generates about 222.9 million tons of waste each year and England is responsible for 85% of the total waste generated. With the continuous increase in population at the municipals, the amount of waste generated will increase. The World Bank Data projected a 69% increase by 2050 (Ulutürk et al., 2023). There is therefore a need to ensure proper recycling of waste materials to reduce landfills and environmental pollution.

Wolverhampton City Council has made tremendous efforts in waste recycling using a combination of manual labour and several mechanical processes such as magnets, vibrating machines, eddy current and optical scanners to separate several types of waste (Wolverhampton Council, n.d). This technique seems efficient at the moment. However, the current method is costly, labour intensive and time-consuming. Recently researchers have made efforts in classifying waste with the use of deep learning especially Convolutional Neural Networks (CNNs) to enhance efficiency and reduce operational cost.

Deep learning methods, particularly CNNs, have shown potential in automating waste classification, thereby reducing the dependency on manual sorting and too many mechanical procedures as characterized by the current method. Studies have demonstrated that CNN-based models can achieve high accuracy in waste classification making it a viable alternative to current methods (Fan et al., 2023; Srivastan et al.,2021).

The application of CNNs in waste classification does not only address high cost but also improves the speed of sorting, thereby enhancing the overall recycling process. Furthermore, advanced techniques such as transfer learning and data augmentation have further boosted the performance of CNN models making them more robust and capable of handling diverse waste types (Ulutürk et al., 2023).

This project focused on exploring cost efficiency and effectiveness of using CNN-based deep learning approaches for waste classification using Wolverhampton City Council as a case study. By comparing operational costs and performance with the current method and the proposed methods, this project seeks to provide insights into the potential benefits of adopting AI driven solutions in the domain of waste management and to contribute to ongoing efforts to improve recycling processes and reduce environmental impacts of waste, which is in line with sustainability goals.

1.1 Problem Statement and Motivation

Waste classification is an important part of efficient waste management systems, contributing to recycling efforts and reducing environmental impact. Traditional methods of waste classification employed in Wolverhampton City Council rely on manual sorting and a combination of too many mechanical processes, although the council boast of achieving a high percentage of accuracy in waste sorting, the process is labour-intensive, time-consuming, and prone to human error (Wolverhampton Council, n.d). Recently, Convolutional Neural Networks (CNNs) have shown promising results in automating this process, achieving significant accuracy improvements (Nnamoko et al., 2022).

However, the performance of these models is often hindered by limited and unrepresentative datasets, which do not cover the full diversity of waste types encountered in real-world scenarios. Moreover, existing models tend to struggle with degraded waste samples, reducing their robustness and reliability (O'Gara & McGuinness, 2019). Therefore, there is a need to enhance the training datasets used for these models to improve their generalizability and robustness, while maintaining computational efficiency.

1.2 Project Aim and Objectives

The aim of this project is to evaluate the effectiveness of Convolutional Neural Networks (CNNs) for waste classification in comparison to traditional methods used by Wolverhampton Council. This research seeks to determine the cost efficiency, accuracy, performance, and robustness of CNN models enhanced through data augmentation and synthetic data generation. To achieve this aim, the following objectives are considered.

1. To investigate the current waste classification technologies used by Wolverhampton Council.
2. To examine the limitations of existing CNN models for waste classification, focusing on dataset deficiencies.
3. To implement various data augmentation techniques, including rotation, scaling, flipping, color adjustments and random erasing to enhance the dataset.
4. To develop a framework for synthetic data generation using Generative Adversarial Networks (GANs) to generate waste images.
5. To integrate the augmented and synthetic datasets into the CNN training process and validate the model's performance.

6. To compare the cost efficiency of CNN-based waste classification with traditional separation technologies used by Wolverhampton Council.
7. To analyze the impact of data augmentation and synthetic data generation on the accuracy, performance, and robustness of CNN models in waste classification.
8. To assess whether CNN-based waste classification can reduce operational and maintenance costs compared to previous methods used in Wolverhampton.
9. To provide recommendations for the best CNN-based model for waste classification in municipal waste management practices

Furthermore, this project will focus on the 3 research questions listed below:

1. How does the cost efficiency of CNN-based waste classification compare to the separation technologies utilized by Wolverhampton Council?
2. To what extent can data augmentation and synthetic data generation enhance the accuracy, performance, and robustness of CNN models in waste classification?
3. Can CNN-based waste classification reduce the operational and maintenance costs associated with the methods previously employed by Wolverhampton Council?

1.3 Overview of the Report

This report is structured as follows:

Section 1: This section shows the Introduction, problem statement, the research aims and objectives along with the research question.

Section 2: This is the Literature Review section. It details the outcomes of previous research on waste classification techniques, the cost implications of previous and proposed methods, and the most prevalent algorithms used, particularly focusing on CNN and GAN methodologies. It also discusses various data augmentation techniques to enhance image datasets.

Section 3: This is the Methodology section, and it covers the research methodology, describing the models and techniques used and implementation process.

Section 4: This section is for Results and Analysis. It presents the results obtained during the research experiments and provides an analysis of these results.

Section 5: In this section, the results are discussed, highlighting their significance and implications.

Section 6: This is the Project Management section, and it outlines the management aspects of the project.

Section 7: This section summarizes the study, highlighting achievements, limitations and directions for future research.

Section 8: This is the final section, and it contains the student’s reflection on the overall project.

2. LITERATURE REVIEW

The objective of this literature review is to provide a comprehensive examination of the current state of waste management and classification technologies, focusing on the integration of Convolutional Neural Networks (CNNs) and advanced data techniques such as data augmentation and synthetic data generation. This review aims to identify existing gaps and demonstrate the potential benefits of these advanced techniques in enhancing waste classification systems, especially for the Wolverhampton Council.

2.1 Waste Management and Classification Technologies

2.1.1 Overview of Waste Management

According to Demirbas (2011), Waste Management is the activity that involves handling, treating, disposing and recycling of waste materials. Effective waste management is great for protecting the environment and conserving resources. It ensures that waste materials are handled systematically from the moment of generation to the final disposal, mitigating adverse environmental impacts and preserving valuable resources.

Effective waste management is important in protecting the environment from pollution and degradation. Improper disposal of waste can lead to severe consequences to the environment, including water pollution, soil contamination and air pollution (Fang et al., 2023). Particularly, Landfills are significant sources of methane which is a potent greenhouse gas that contributes to climate change (Blair, 2024). By implementing comprehensive waste management practices such as recycling, harmful pollutants can be significantly reduced, thereby protecting the environment. According to BBC (2023), recycling paper, plastic, glass and metal reduces greenhouse gas emissions as well as conserving resources. Furthermore, composting food organics and vegetation waste can also produce valuable compost, thereby enhancing soil quality and reduces the over reliance on chemical fertilizers.

Waste management plays a pivotal role in conserving natural resources. Recycling and reusing of materials reduce the demand for raw materials extraction, thereby preserving finite natural resources such as minerals, timbers and fossil fuels. For instance, recycling paper and cardboard reduces the need for timber, thus preserving forests and biodiversity. Similarly, recycling metals like aluminum and steel can save a lot of energy and reduce the environmental footprint of mining activities (Wan et al., 2019).

The circular economy model, which promotes continuous resource use through recycling and reusing of waste materials (as shown in figure 1) is currently being widely adopted as a sustainable approach to managing resources. According to Butler (2023), the use of advanced sorting technologies improves the quality of recycled products produced, thereby supporting the principles of a circular economy.

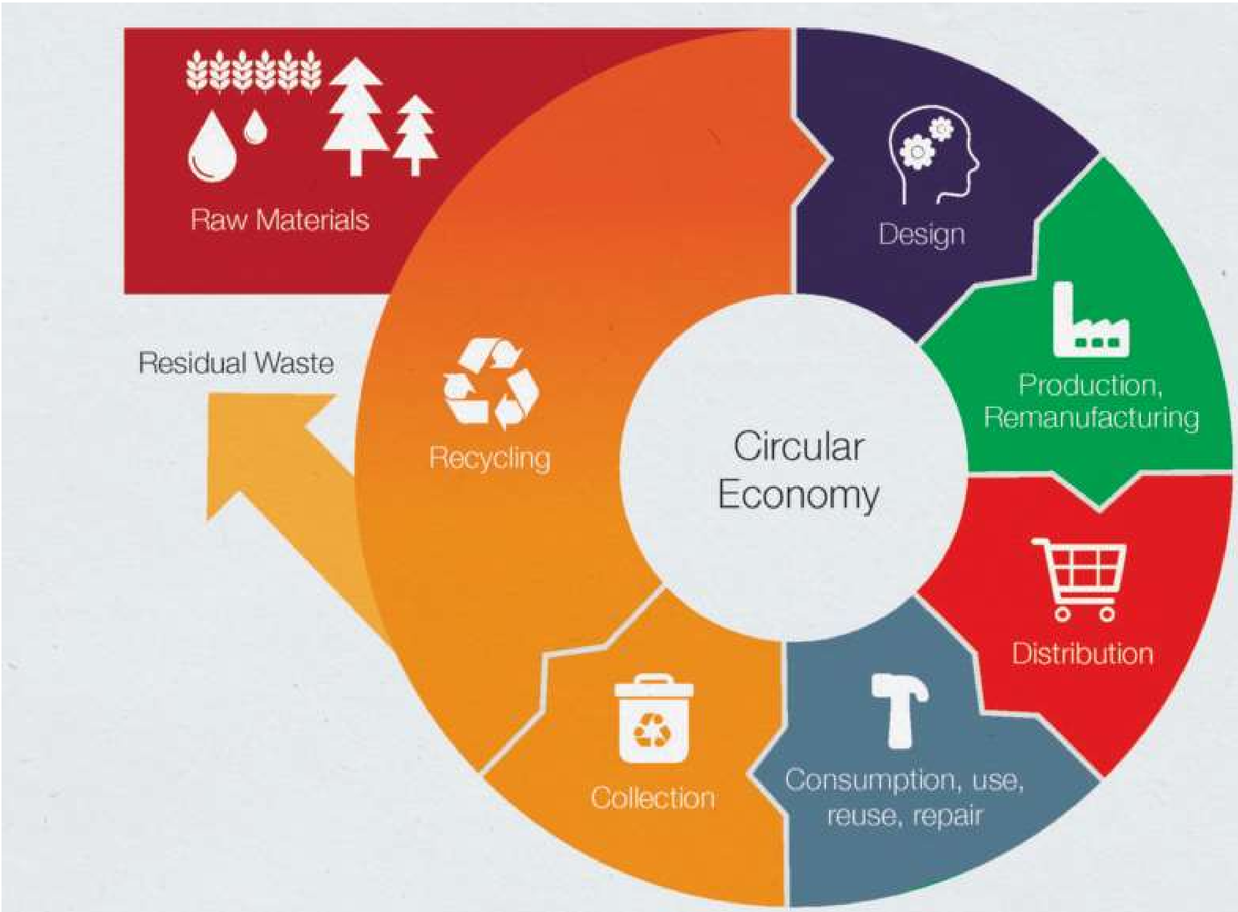


Figure 1: Diagram of a circular economy (Wolverhampton Council 2018).

2.1.2 Current Waste Management Practices in Wolverhampton Council

The City of Wolverhampton Council operates a waste management system designed to reduce landfill use and maximize recycling and reuse of materials. They are responsible for both the collection and disposal of all waste, ensuring a seamless and efficient waste management process.

In Wolverhampton Council, Different types of bins are used for waste management. The Black bins are for recycling waste, which include items such as aerosols, cardboard, drink cans, foil packaging, food tins, glass bottles and jars, household plastic bottles, newspapers, magazines, papers, food and drink cartons. Placing incorrect items in the recycle bins will result in your bin not being emptied until residents remove non-recyclable items in the bin before the next scheduled collection. General waste is placed in the brown or green bins which includes items like bread bags, CDs, DVDs, coat hangers, soft plastic wrappers, drinking glasses, food waste, nappies, pet waste, plastic bags and polystyrene. Garden wastes are placed in purple bins which includes grass cuttings, hedge clippings, leaves, small branches, tree pruning, twigs, weeds and wooden green waste, this service is offered as a paid-for service. Figure 2 shows the different bins used in the City of Wolverhampton for waste management. Additionally, the council provides special collections for bulky waste such as clinical waste, furniture and appliances (Wolverhampton Council, n.d).

The contents of the blue-lid bins (recyclables) are taken to the Material Recycling Facility (MRF). The current process of waste recycling at the material recycling facility is in the following stages as retrieved from Wolverhampton Council, (n.d).

1. Initial Sorting: At this stage the waste is loaded onto the conveyors, it is then sorted manually to remove any non-recyclable waste that was wrongly placed by the households.
2. Vibrating Machine: A vibrating machine separates cardboard and paper from the rest of the recyclable waste, then the separation between cardboard and papers is done by hand.
3. Magnetic Separation: Steels and tin cans are removed from the waste using magnets.
4. Optical Scanning: Plastic bottles, pots, tubs are separated with the use of optical scanners.
5. Eddy Current Separation: aluminum cans are separated using eddy current.
6. Glass Sorting: At this stage it is believed that the only item remaining will be glass. So, it is automatically bailed into a large container labelled as glass.

Figure 2: Bin types for waste management (*What Goes in My Bins? | City of Wolverhampton Council*, n.d.)

The City of Wolverhampton Council waste management system has made a commendable effort to reducing landfill use and maximizing the recycling and reuse of materials. The current collection and recycling of waste materials at the Material Recycling Facility (MRF) involves multiple steps ranging from initial sorting to glass sorting. The process seems thorough. However, it is labour-intensive and costly. Also, manual sorting at the initial stage to remove any non-recyclable items indicates a dependency on human skills, which can be highly prone to error and inefficiency. Advances in automated sorting technology like incorporating artificial intelligence, need to be explored to improve on this aspect and reduce overall long-term cost (Fang et al., 2023).

2.2 Convolutional Neural Networks (CNNs) in Waste Classification

2.2.1 Image Classification

Image Classification is a critical task in the field of computer vision. It involves the process of identifying the labels or probabilities of input images based on their content (Paneru & Jeelani, 2021). In the context of waste management, this involves categorizing images of different types of waste materials as depicted in figure 3. An image classification algorithm can differentiate between recyclable materials like cardboard, glass, metal, paper, plastic, food organics, textile trash and so on.

Object classification within waste images, segments them into discrete objects and classified them based on features like color, shape and texture. This approach mimic the human visual analysis and it is crucial to effectively managing waste sorting, (Paneru & Jeelani, 2021)

Early methods relied on statistical analysis of pixel data, which posed challenges due to variations of lighting, background clutter and the orientation or position of the object. However, with the adoption of Convolutional Neural Networks (CNNs), there has been an improvement in the classification accuracy significantly because, this neural network extracts meaningful features from the images (Wu et al., 2024).

Despite the advancements in classification with the use of Convolutional Neural Networks, there are still some challenges which include high variability in waste types, occlusions, environmental conditions during the image capturing, which can affect classification conditions (Wu et al., 2020).

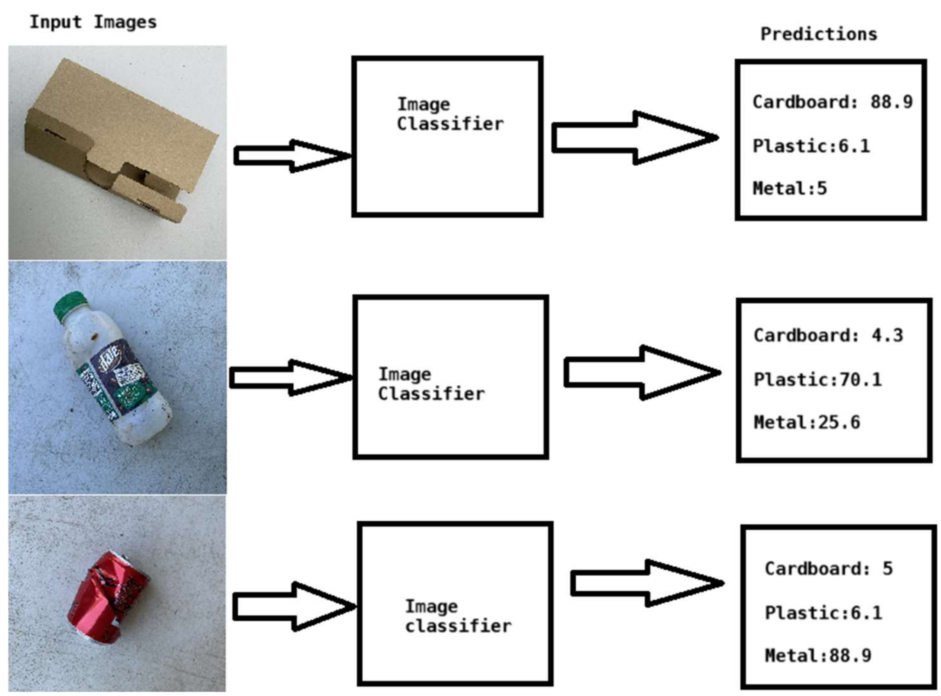


Figure 3: Illustration of techniques used for classifying household waste (Single et. al, 2023).

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specialized neural networks designed to process data with grid-like topology, such as time-series and image data. It was introduced by LeCun in 1989. The core operation performed in CNNs is convolution, this is a specific type of linear operation that is replacing general matrix multiplication in at least one layer of the network. During image processing, the convolution of an image I and a kernel K is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \quad (1)$$

(Goodfellow et al., 2016).

This operation generates feature maps that help in identifying features and patterns that are in the images. CNN also uses pooling operations, such as max pooling and average pooling to reduce the spatial dimensions of the feature maps, which further reduces computational complexity and control of overfitting problem (Goodfellow et al., 2016).

CNNs are specifically designed for processing and analyzing visual data, making it an ideal choice for classifying different types of waste. The convolutional layers can recognize edges, textures and any relevant characteristics in the waste images, which enables the neural network to learn and leverage these features for classifying waste accurately (Single et al., 2023).

Since a deep learning model called AlexNet won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) championship in 2012, CNNs have become the leading and most popular image recognition model among various other computer vision algorithms (Na et al., 2022).

CNNs are inspired by the biological mechanisms of visual cognition and typically consist of input layers, convolutional layers, pooling layers, connected layers, and output layers as shown in figure 4. CNNs have two notable features, Firstly, the neurons in the convolutional layers are not fully connected, and secondly, the weights of connections between certain neurons within the same layer are shared. This design, characterized by sparse connections and weight sharing, reduces the complexity of the network (Shijie et al., 2017).

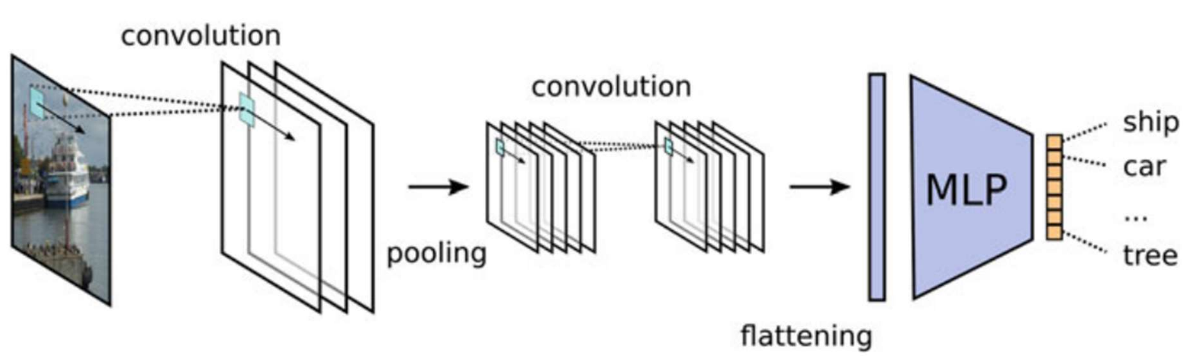


Figure 4: Topology of a Convolutional Neural Network (Toennies, 2024).

2.2.3 Applications of CNNs In Waste Classification (Related Works)

Recent advancements in convolutional Neural Networks have shown significant potential in automating waste classification processes, enhancing accuracy and further improving the overall waste management systems. This section reviews related works in this domain, highlighting various methodologies, results and limitations.

Adedeji and Wang (2019) conducted research to develop an intelligent waste material classification system that automates the separation of waste into different categories such as glass, metal, paper and plastic using a combination of Convolutional Neural Network and Support Vector Machine (SVM). The study addresses the need for an automated, accurate and efficient system for waste classification to enhance waste management processes. They utilize ResNet-80 CNN model as a feature extractor and SVM for classification. The test was performed on the image dataset developed by Gary Thung and Mindy Yang and it resulted in an accuracy of 87%. The system significantly reduces human involvement, thereby making the waste separation process more intelligent. Although the model performed well, the limitation was its non-robustness to totally new sets of waste images, because a small sized dataset was used for testing, which may not fully represent the variety of waste types encountered in the real-world.

To develop a highly accurate model for classifying waste into seven different categories, Srivatsan et al. (2021), utilized transfer learning with Convolutional Neural Networks (CNN) using the CompostNet dataset which is an extension of the TrashNet dataset. The experiment was conducted using pre-trained models such as, MobileNetV2, ResNet34 and Densenet121. It achieved a high accuracy of 96.42%, 96.27% and 96.273% respectively. The study concluded that these models can effectively classify waste materials and improve waste management process. One of the limitations is the reliance on a relatively small and specific dataset (CompostNet), which might not generalize well to other types of waste or dataset from different regions.

Ulutürk et al. (2023), carried out a study to develop and optimized several deep Convolutional Neural Network models for waste classification, focusing on hyperparameter optimization using grid search. The study addresses the challenges of achieving high classification accuracy in waste classification systems, particularly in datasets with limited data and high inter-class similarity. The key issue was overfitting and determining the optimal hyperparameter combination. The optimized models, particularly the DenseNet169 architecture achieved high accuracy of 96.42% and F1 score of 96%. The research highlights the potential for these models to be integrated into waste management systems for efficient and accurate waste classification. However, there was a challenge of excessive depth in the models, leading to gradient vanishing problems and the time taken for classification in real-time implementations took too long, which is a limiting factor for practical applications.

To evaluate the effectiveness of various CNNs for classifying real-life landfill waste, Single et al. (2023), used the newly developed RealWaste dataset, which includes real-world waste samples. This significantly improved classification accuracy, with the InceptionV3 model achieving an 89.19% classification accuracy, demonstrating a great potential of CNN for practical waste management applications. However, the model's accuracy depends heavily on the quality and representativeness of the training data. The researchers further suggested a future research focus on expanding the diversity of waste classes in the dataset to improve model training so it can handle varied and degraded states of real-world waste materials.

The reviewed works so far demonstrated the capabilities of CNN in waste classification, achieving high accuracy rates and showing great potential for integrating these models into practical environments of waste sorting and classification. However, common limitations among these studies include dataset size and diversity, model robustness and real-time performance challenges. Which can be addressed by incorporating advanced data augmentation and synthetic data generation to optimize the models for real-time applications, which will increase the accuracy, performance and practical applicability of CNN-based waste classification systems.

2.3 Data Augmentation and Its Techniques

2.3.1 Data Augmentation

One of the major challenges that continually limits the application of deep learning and computer vision is insufficient training data. This challenge usually leads to overfitting because the model has very little information to learn from (O'Gara & McGuinness, 2019). The more data any machine learning algorithm have, the more effective the model will be (Perez & Wang, 2017). Data Augmentation is the process of applying some amount of mutation in the original training data and synthetically creating new samples to increase the amount of the training data (Inoue, 2018).

There are several techniques in data augmentation, both basic and advanced techniques. Basic techniques include methods like flipping, rotating, cropping, and color jittering. This is illustrated in figure 5. Flipping an image horizontally or vertically is a simple yet effective technique to enhance model generalization. Rotating images at random orientations can help the model become invariant to the orientation of the objects within the images. Cropping involves cutting out a part of the image and resizing it to the original dimensions, which helps the model to focus on different parts of the image. Color jittering adjusts the brightness, contrast, and saturation of images, introducing variability that can help in making the model robust to different lighting conditions (Shijie et al., 2017).

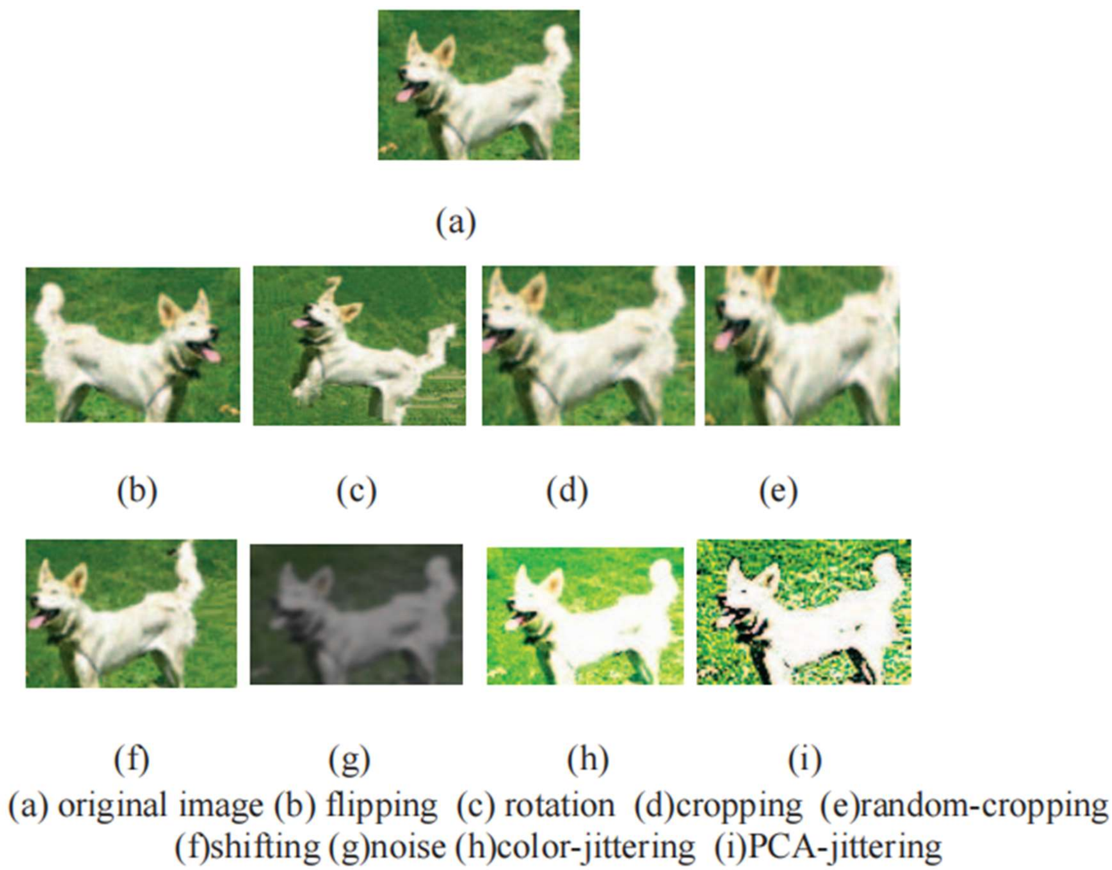


Figure 5: Original and augmented images (Shijie et al., 2017)

The advanced technique used in data augmentation involves more sophisticated approaches like, the use of Generative Adversarial Networks (GANs) to generate synthetic new data. GANs can produce highly realistic images that can enhance the training of the model (Perez & Wang, 2017).

2.3.2 Impact of Data Augmentation on Model Performance.

Data augmentation plays an important role in improving the generalization and accuracy of a Convolutional Neural Networks. It helps to mitigate overfitting and helps the model to perform optimally on new and unseen data.

Several studies have demonstrated the significant impact of data augmentation on model performance. For instance, Ulutürk et al. (2023) carried out research to develop a deep learning model that can achieve the best result in image classification task. The researchers employed simple augmentation techniques such as 30-degree rotation, 0.2% shift, 0.2 shear range, 0.2 zoom range and y-axis flip. Training and fine-tuning were done on DenseNet169 model which gave an accuracy of 96.42% and an F1 score of 96%.

In another study, Kumsetty et al. (2023) employed the use of data augmentation for waste classification using the TrashNet and TACO dataset with limited waste images. They performed data augmentation operations like flipping, rotations, shearing, center cropping, brightness adjustment and translation to create new images with different characteristics to improve the frequency of the dataset. The TrashNet dataset was trained on ResNet-34 model achieving an accuracy of 93.13%, and the TACO dataset was trained on ResNet-101 model achieving an accuracy of 91.84%. Performing considerably better than previous studies conducted without augmentation.

In a recent study by Gao et al. (2024) to classify food images using data augmentation, the researchers proposed a data augmentation method which they called "Augmentplus". This involves randomly selecting an image from the datasets (food-101 and Vireo Food-172) and performing techniques such as RandomResizedCrop, RandomHorizontalFlip, TrivialAugmentWide, GaussianBlur, Gray scale, Solarization, ColorJitter and RandomErasing. This is illustrated in figure 6. Which resulted in a 95% accuracy on food-101 dataset and 94.29% accuracy on Vireo Food-172 dataset.

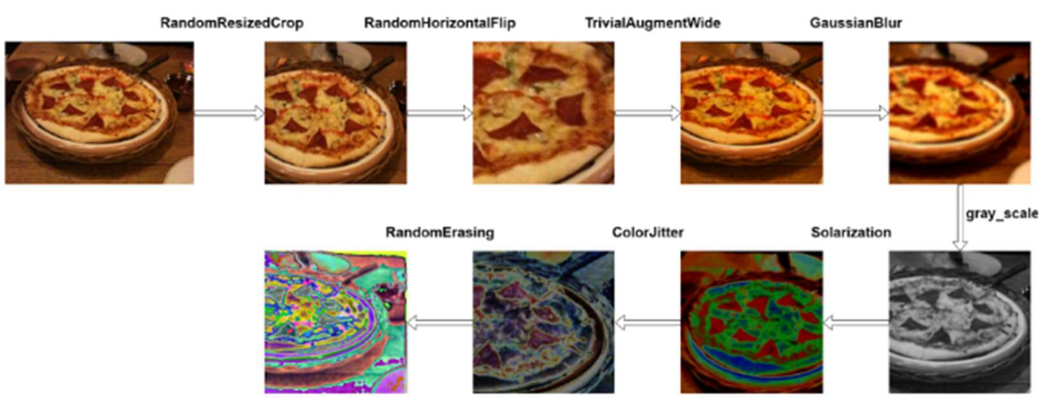


Figure 6: Illustration of Augmentplus (Gao et al., 2024).

Data augmentation techniques have been proven to significantly enhance the robustness and accuracy of CNN models, ultimately leading to better model generalization and performance. This project will incorporate both basic Data Augmentation in combination with synthetic image generation using Generative Adversarial Networks (GANs).

2.4 Synthetic Data Generation with Generative Adversarial Networks (GANs)

2.4.1 Overview of GANs

Generative Adversarial Networks (GAN) are a class of artificial intelligence algorithm that is used in unsupervised learning, which aims to generate new data samples that resemble a given distribution. It was introduced by Ian Goodfellow and his colleagues in June 2014. Since its introduction it has gained significant attention due to its ability to produce high quality synthetic data across various domains (Goodfellow et al., 2020).

GANs can generate videos, music and human faces that cannot be differentiated from a photo taken with a camera. A typical GAN consists of two primary components, The Generator G and The Discriminator D. The Generator model is responsible for creating new reasonable synthetic samples from the given samples. The Discriminator model is responsible for determining if the sample generated is real or fake by comparing it with the real data. The discriminator penalizes the generator every time it generates a fake image by giving it a low score (Kaddoura, 2023).

The generator G takes a random noise vector z from a latent space and maps it to the data space to create synthetic data samples G(z). The discriminator D takes the data sample as input and outputs a probability value indicating whether the sample is real (from the true data distribution p_r) or fake (from the generator). The objective of GANs can be formalized as a minimax game between G and D. This can be mathematically represented as:

$$L(D,G) = \min_{G} \max_{D} E_{x \sim p_r}\left[\log\left(D(x)\right)\right] + E_{z \sim p_z}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (2)$$

(Jenkins & Roy, 2024).

Through this adversarial process, the generator learns to produce increasingly realistic images, as it aims to deceive the discriminator, which makes the discriminator becomes better at detecting fakes as illustrated in figure 7 (Kaddoura, 2023).

Different variants of GANs have been developed to increase performance and address some specific challenges. Deep Convolutional GAN (DCGAN) uses convolutional layers to generate images of high resolution (Jenkins & Roy, 2024), also, Wasserstein GAN (WGAN) seeks to improve stability in training with the use of different loss function to effectively capture the distance between distribution (Li et al., 2024). Some other variants of note include the Conditional GAN (cGAN), which incorporates auxiliary information to generate data samples conditioned on the given inputs (Dong et al., 2022). CycleGAN enables image-to-image translation tasks without paired training example (Ni, 2023).

GANs have been very impactful in the field of data augmentation by increasing the training dataset, thereby enhancing model robustness and performance. It is a very powerful and versatile tool in the AI toolkit and has proven to be invaluable in scenarios with limited data. Their continued evolution promises to unlock further capabilities and applications in artificial intelligence and machine learning (Lu et al., 2022).

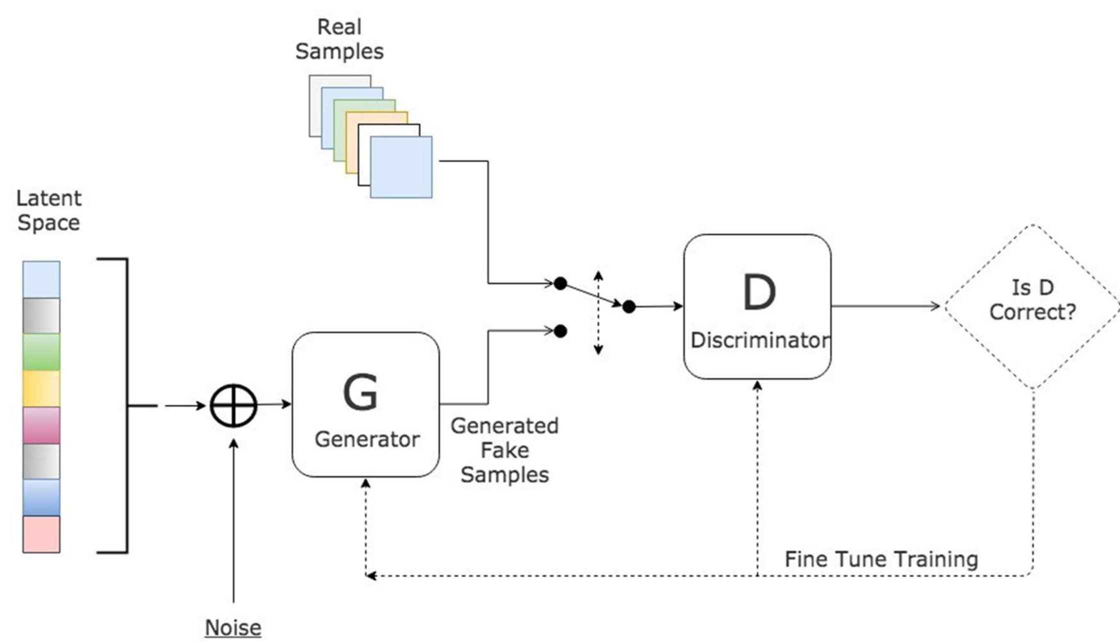


Figure 7: GAN Architecture (Gharakhanian, 2016).

2.4.2 Applications of GANs in Image Synthesis for Waste Classification (Related Works)

Research was carried out by Fan et al. (2023), seeking to enhance waste image recognition and detection using an improved DCGAN for multi-target waste data augmentation. Their research addresses the challenges of insufficient data samples in waste classification by employing the Wasserstein distance in DCGAN to generate realistic multi-target waste images. Their experiment demonstrated a significant improvement. There was a 74.05% reduction in model size, a 4.54% increase in Mean Average Precision to 96.35% and a frame rate of 24 FPS, outperforming other object detection models trained without the generated images. However, there were challenges in target recognition under complex backgrounds and overlapping objects, addressing these complexities is crucial for enhancing the practical applicability and accuracy of waste image recognition systems.

The application of GANs in image synthesis, particularly in waste classification, has been growing. Chatterjee et al. (2022) conducts research to classify plastic bottle waste. The research findings reveal a significant improvement in classification accuracy when they utilized a modified lightweight-GAN model to generate synthetic image of plastic bottle waste and integrated it with the original ones, achieving a remarkable accuracy of 99.06% surpassing traditional augmentation methods. However, there was potential overfitting issues which could be resolved through further validation across diverse dataset beyond plastic bottles.
To buttress the need for Synthetic Data Generation in image classification, Tsagarakis and Maniadakis (2024) conducted research on using Generative Adversarial Networks (GANs) for image synthesis to enhance waste classification. The research objective was to improve the accuracy of waste classification system. The authors employed a generator-discriminator approach to generating varied and realistic training datasets. This method involved using geometric deformations to diversify object appearances and training a neural network to evaluate how real the image is. The findings in the research revealed that incorporating synthetic images into the training dataset significantly improved model performance, particularly in detecting PETE bottles. However, challenges such as training instability were experienced in the research. This can be resolved by further optimizing the discriminator model and exploring additional deformation techniques to enhance the variety and realism of the synthetic dataset.

Qi et al. (2024) conducted research aiming to improve waste classification using GANs to synthesize images for data augmentation. The study addressed the issue of limited datasets in waste classification domain which usually affects the training of robust machine learning models. Their findings showed that the dataset generated by GANs significantly improved the accuracy of the model, achieving an accuracy increase from 75% to 85%. However, the research highlighted a limitation in the realism of generated images, which occasionally lack fine details critical for accurate classification.

Previous research carried out showed the potential of Generative Adversarial Networks (GANs) in enhancing waste classification system by addressing data scarcity through synthetic image generation. However, several limitations persist, including the problem in recognizing targets under complex backgrounds and overlapping objects (Fan et al., 2023), problem of potential overfitting (Chatterjee et al., 2022), training instability (Tsagarakis and Maniadakis, 2024) and lack of fine details in generated images (Qi et al., 2024). This project aims to resolve these issues by employing an advanced and optimized GAN model to generate more realistic and varied synthetic waste images. This approach will improve classification performance and reliability ensuring robust waste management applications across diverse complex environments.

2.5 Cost and Efficiency Considerations

2.5.1 Operational Costs of Current Methods of Waste Classification in Wolverhampton City Council

Wolverhampton City Council like most council in the United Kingdom employs a multi-stage process for recycling waste utilizing various technologies in addition with manual labour to ensure effective separation of recyclable waste materials.

The initial sorting stage involving manual removal of non-recyclable waste, is predominantly associated with labour wages, training and occupational health and safety measures. This stage of the process is time-consuming, contributing to higher operational cost due to the need for prolonged operation times and some potential inefficiencies that are usually involved with manual processing (Rahman et al., 2013).

The second phase involving the use of vibrating machine to separate cardboard and paper from the rest of the recyclable, requires an initial investment in machinery, and an ongoing maintenance and energy consumption are also significant (Wang et al., 2023). Moreover, the wear and tear on the machinery needs regular maintenance and part replacement, contributing to the overall operational cost (Yazdi, 2024).

The method used in the third phase involving the use of magnets to remove steel and tin cans from the waste stream, seems efficient and reduces the need for manual sorting. However, the installation and maintenance of magnetic separators and the energy consumption incur huge costs (Duncan, 2024).

Optical scanning used in the separation of plastic bottles, pots and tubs have significantly great accuracy and speed. However, the operational costs include regular calibration, maintenance and the need for several skilled technicians is high (Cujbescu et al., 2023).

Eddy current separation employed to extract aluminum cans, needs a considerable initial investment for installation. This method, although efficient in separating recyclable aluminum and reduces labour, it comes with high operational cost which includes but not limited to electricity and periodic maintenance (Romero-Arismendi et al., 2024).

Although the waste classification methods used at the Material Recycling Facility at Wolverhampton City Council have achieved a significant accuracy in waste separation, however, it involves a significant operational cost due to the combination of manual labour and lots of technologies. The presence of too many stages and machineries such as vibrating machines, magnetic separators, and optical scanners not only increases the complexity of the system but also the overall cost, because each stage requires regular calibration, energy consumption and skilled technicians for maintenance.

2.5.2 Cost Efficiency of Deep Learning Approaches

The application of deep learning, especially Convolutional Neural Networks (CNNs) in waste classification has been increasing in recent years, showing promise of cost efficiency compared to the traditional waste processes used by Wolverhampton City Council, which uses a combination of manual labour and various other machineries. These methods, although effective, are costly due to their complexity and labour-intensiveness.

Studies have shown that deep learning approaches, particularly CNNs, offer great cost-saving benefits. A study by Adedeji and Wang (2019) demonstrated that using CNN-SVM hybrid model for automated waste classification significantly reduced the need for manual labour. This therefore translates to lower operational cost.

Further research by Srivatsan et al. (2021) explored the use of transfer learning with pre-trained CNN models such as MobileNetV2, ResNet34 and DenseNet121 for waste classification. The study emphasized that the initial investment in deep learning infrastructure could be offset by the reduction in cost associated with manual sorting and an enhanced accuracy, which reduces the need for resorting of waste, thereby increasing the quality of recyclable materials produced.

Furthermore, Ulutürk et al. (2023) focused on optimizing deep CNN models through rigorous hyperparameter tuning, which further enhanced classification accuracy and robustness. The optimized model trained demonstrated high performance, thereby reducing the need for multiple stages of mechanical sorting. The optimization not only improves the cost efficiency, it also minimizes energy and maintenance cost associated with the machinery method.

Recent advancement in data augmentation and synthetic data generation techniques has further enhanced the cost efficiency of deep learning models. Ambika and Kannammal (2023) suggested that exploring data augmentation techniques, which include GANs, to generate synthetic samples and improve model performance, could reduce the reliance on large, labeled datasets, potentially lowering annotation costs and making deep learning-based waste classification more economically viable in the long run.

The shift towards deep learning methods, especially CNNs in waste classification will bring about a cost-efficient alternative to current methods used. The eradication of manual labour and decreased dependency in multiple machinery stages contribute to overall operational cost savings.

2.6 Gap Analysis

After thorough review of several related papers, it was discovered that there have been significant advancements in waste classification technologies. However, existing research has not fully leveraged the combined potential of Generative Adversarial Networks (GANs) and traditional data augmentation techniques. Studies such as those by Fan et al. (2023) and Chatterjee et al. (2022) focus on synthetic data generation using GANs, while others like Srivatsan et al. (2021) and Kim et al. (2021) employ basic data augmentation techniques independently. Nonetheless, none of these studies integrate both methods in a single framework. This leaves a notable gap in optimizing dataset diversity and robustness. This project aims to leverage the knowledge gained from these studies to create a more robust model, by combining GAN-generated synthetic data with traditional data augmentation to enhance the training dataset's comprehensiveness and robustness. Thus, this project fills this critical gap by demonstrating the synergistic benefits of integrating GANs with conventional augmentation techniques, providing a more robust solution for real-world waste classification challenges.

3. METHODOLOGY

The SEMMA methodology was be adopted in this research. SEMMA is an acronym that stands for Sample, Explore, Modify, Model, and Assess. It is a data mining process developed by SAS Institute to guide the implementation of data mining applications. The life cycle of the SEMMA methodology is depicted in Figure 8.

This methodology was chosen because it enables the understanding, organization, development, and maintenance of data mining projects, ultimately delivering solutions to business challenges and aligning with organizational goals (Shafique & Qaiser, 2014).

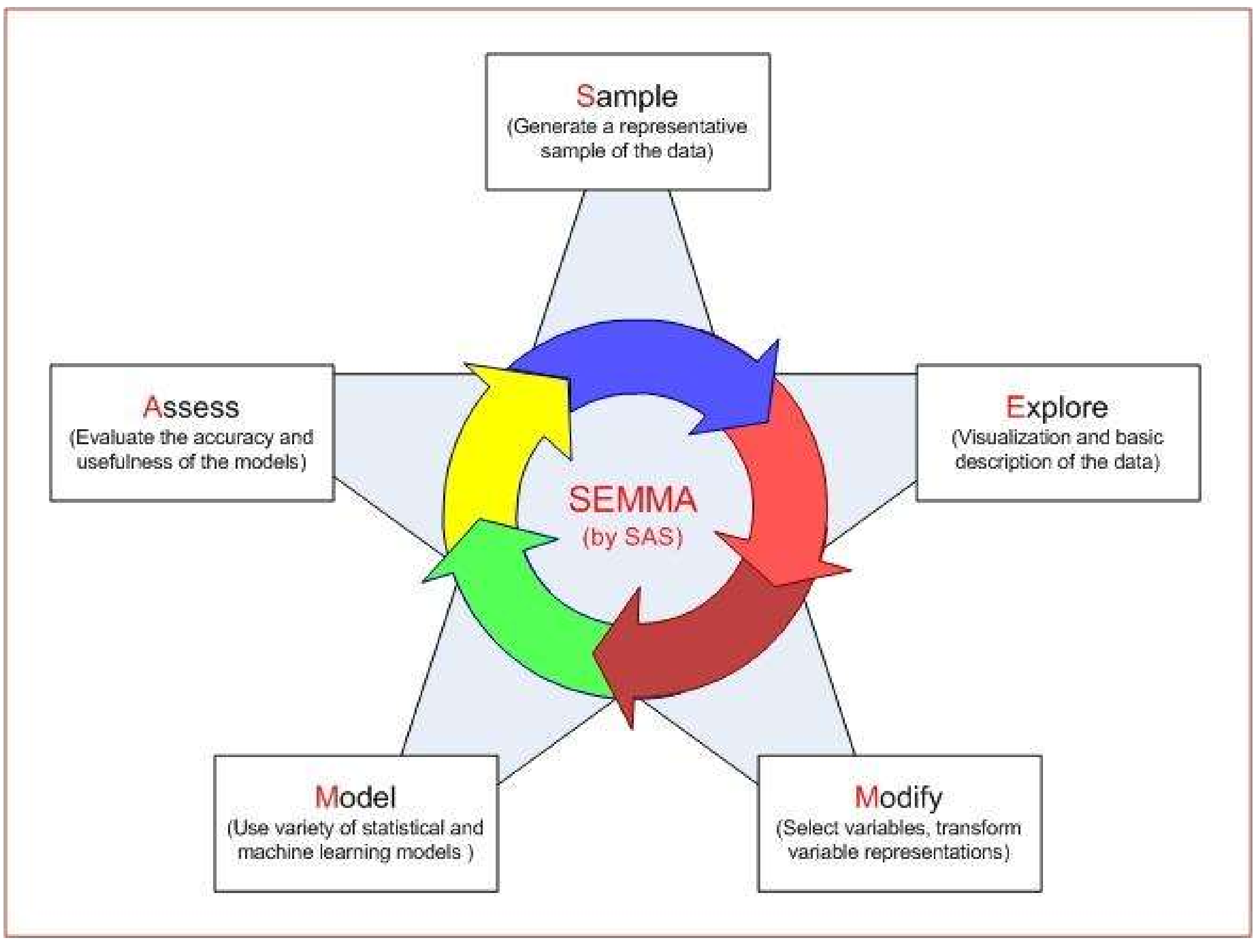


Figure 8: SEMMA Methodology (silsnetwork.com, 2019)

3.1 Datasets

Selecting a dataset that accurately represents household waste proved to be a challenge. However, after extensive research and evaluation of various options, the RealWaste dataset, sourced from the UCI Machine Learning Repository, was chosen for this project due to its comprehensive representation of real-world household waste materials. This dataset comprises of color images of waste items captured at the point of reception in a landfill environment, providing a realistic representation of waste materials encountered in real-world scenarios (Single et al., 2023). The images are available in 524x524 resolution.

The dataset includes nine classes of waste materials; Cardboard, Food Organics, Glass, Metal, Miscellaneous Trash, Paper, Plastic, Textile Trash, and Vegetation. It has a total of 4,752 samples. The distribution of images across these classes is shown in Table 1. The class assigned to each image represent the primary material type present in each waste item.

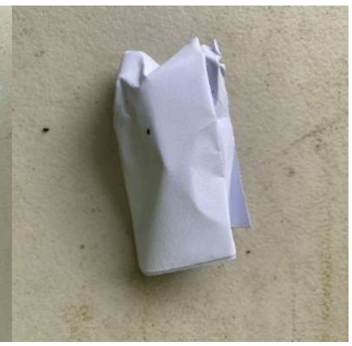
Table 1: Distribution of Images in each Class.

S/N	Class	Image Count
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

The RealWaste dataset offers lots of advantages for this project. Its real-world nature ensures that the trained CNN model will be exposed to the diversity and complexity of waste materials encountered in actual landfill settings. This is crucial for developing a model that can generalize well to real-world waste classification tasks. Table 2: shows some samples of the dataset in each category.

Table 2: Sample of Images in each category.

S/N	Class	Sample
1	Cardboard	 
2	Food Organics	 
3	Glass	 
4	Metal	 

5	Miscellaneous Trash	 
6	Paper	 
7	Plastic	 
8	Textile Trash	 

9	Vegetation	

3.2 Data Preprocessing

Before training the model, the RealWaste dataset undergone several preprocessing steps to ensure optimal performance and compatibility with the CNN architecture. These steps include:

1. Image Resizing: All images in the dataset were resized to a dimension of 80x80 pixels. This is a common practice in image classification tasks as it ensures consistency in input size for the CNN model.
2. Normalization: Pixel values of images are usually between 0 and 255. This number was normalized to a range of 0 to 1, by dividing each pixel by 255. This helps to stabilize and accelerate the training process of the CNN model.
3. Data Shuffling: The dataset was randomly shuffled to prevent any biases that might arise from the order of images during training.
4. Label Encoding: The categorical labels (waste types) were assigned numerical values. (cardboard: 0, food_organics:1, glass: 2, metal: 3, misc_Trash:4, paper: 5, plastic:6, textile:7, vegetation:8). This is necessary because machine learning and deep learning uses only numerical values to interpret and learn during model training.

These steps are essential for preparing the RealWaste dataset for model training because it makes the dataset suitable for input into the model and ensuring efficient and accurate learning.

3.3 Splitting Data for Training and Testing

To ensure the robust evaluation of the CNN model's performance, the RealWaste dataset was divided into two subsets (training and testing). The training set comprises 75% of the total dataset and was used to train the CNN model, enabling it to learn the features and patterns associated with different waste categories. The remaining 25% of the dataset was used as the testing set, it was used to assess the model's final performance on unseen data, providing an unbiased estimate of its generalization capabilities. Figure 9 shows the shape of the data after partitioning. This partitioning approach ensures that the model is trained on a substantial portion of the data while maintaining a significant set of data to evaluate the model, thereby providing a comprehensive measure of the model's effectiveness in classifying waste materials.

```
In [17]: X_train.shape
Out[17]: (3564, 80, 80, 3)

In [18]: X_test.shape
Out[18]: (1188, 80, 80, 3)
```

Figure 9: Output of the Test and Train Split

3.4 Class Imbalance

In many real-world datasets, class imbalance is a significant issue that can adversely affect the performance of machine learning models. Class imbalance occurs when some classes are under-represented compared to others. In this project certain categories of waste materials had fewer samples than others as shown in figure 10, such that if not addressed, will lead to biased model predictions. The data augmentation technique was used to balance the dataset by artificially increasing the size of the minority class and applying various transformations such as rotations, translations, and flips to the existing images. After augmentation, each category of the training set was having equal amount of 688 images.

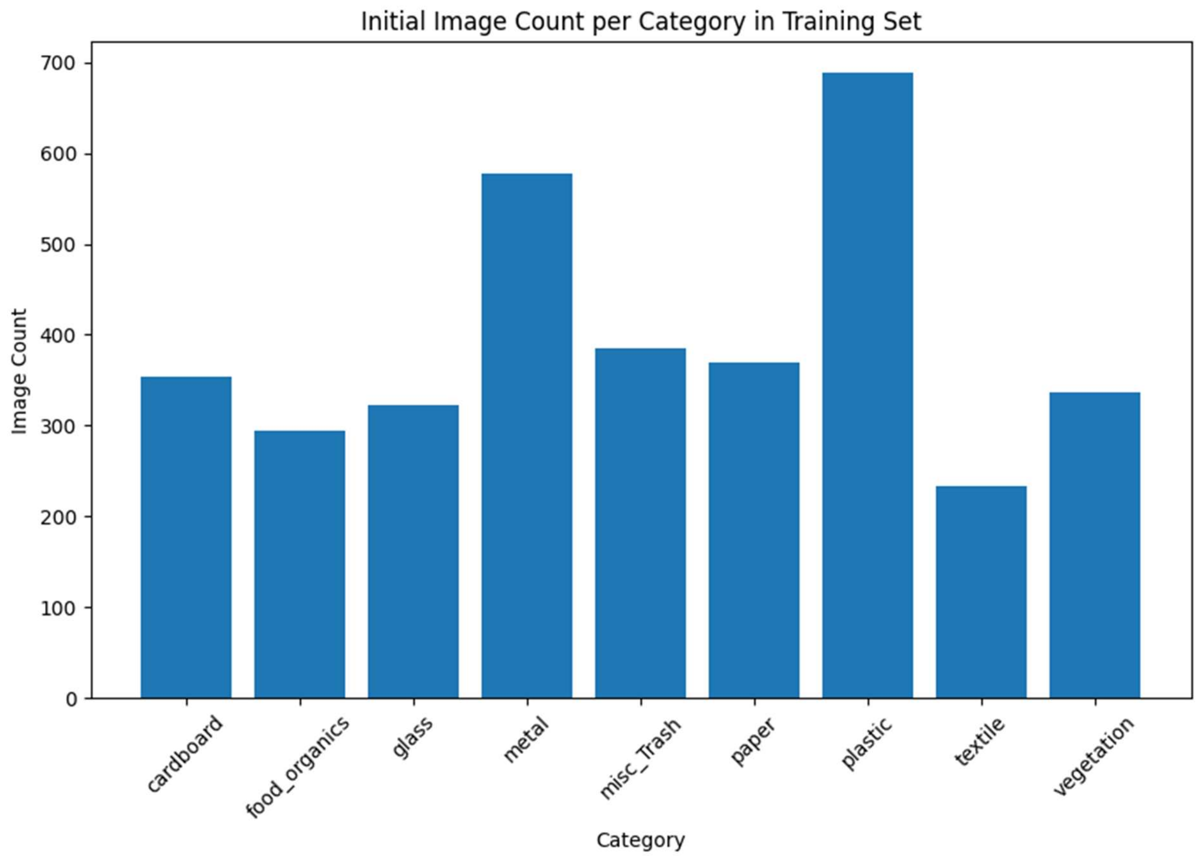


Figure 10: Class imbalance

3.5 Dataset Augmentation

To enhance the diversity of the samples, mitigate overfitting and class balancing, several augmentation techniques were employed to artificially expand the RealWaste dataset. This is a common challenge in deep learning, where models perform well on training data but poorly on new data.

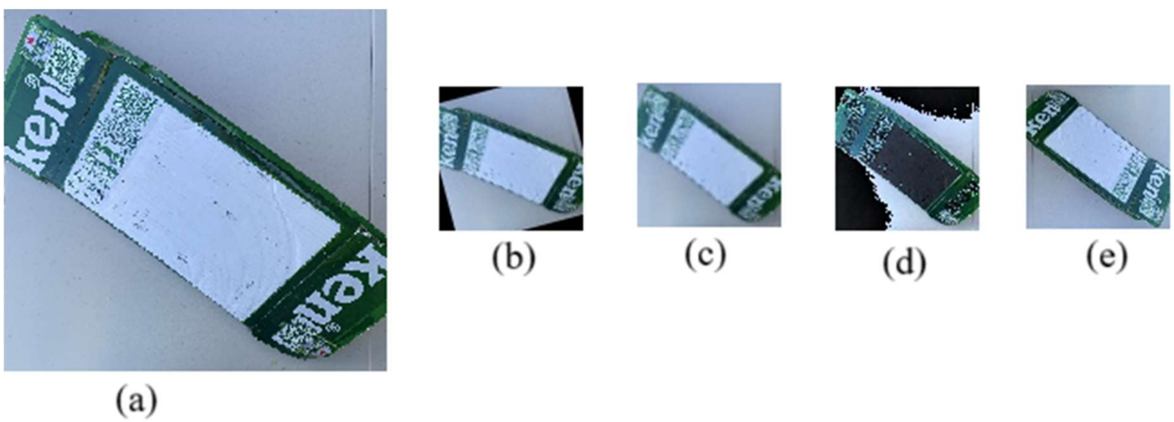
The following augmentation techniques was applied on the RealWaste images:

1. Rotation: Images was randomly rotated by varying angles, simulating different orientations of waste items.
2. Scaling: Images were randomly resized, introducing variations in the scale of waste items.
3. Color Adjustments: Brightness, contrast, saturation and hue was randomly adjusted to simulate different lighting conditions

4. Random Erasing: Random patches of the image was erased, forcing the model to focus on other features for proper classification

5. Flipping: Images were flipped horizontally and vertically, creating mirror images of different waste items.

These augmentation techniques were randomly applied to the images ensuring that the CNN model is exposed to a wide range of different waste images This will encourage the model to learn robust features that are invariant to transformations and variations in appearance, ultimately leading to improved classification on real world waste. A total of 1,962 samples was generated using this technique and samples of the real images and augmented images are shown in Figure 11. Figure 12 shows the block diagram of the augmentation process.



(a)

Original Waste Image (b) Rotation (c) Scaling (d) Random Erasing (e) Flipping

Figure 11: Samples Real Images and augmented images from the dataset.

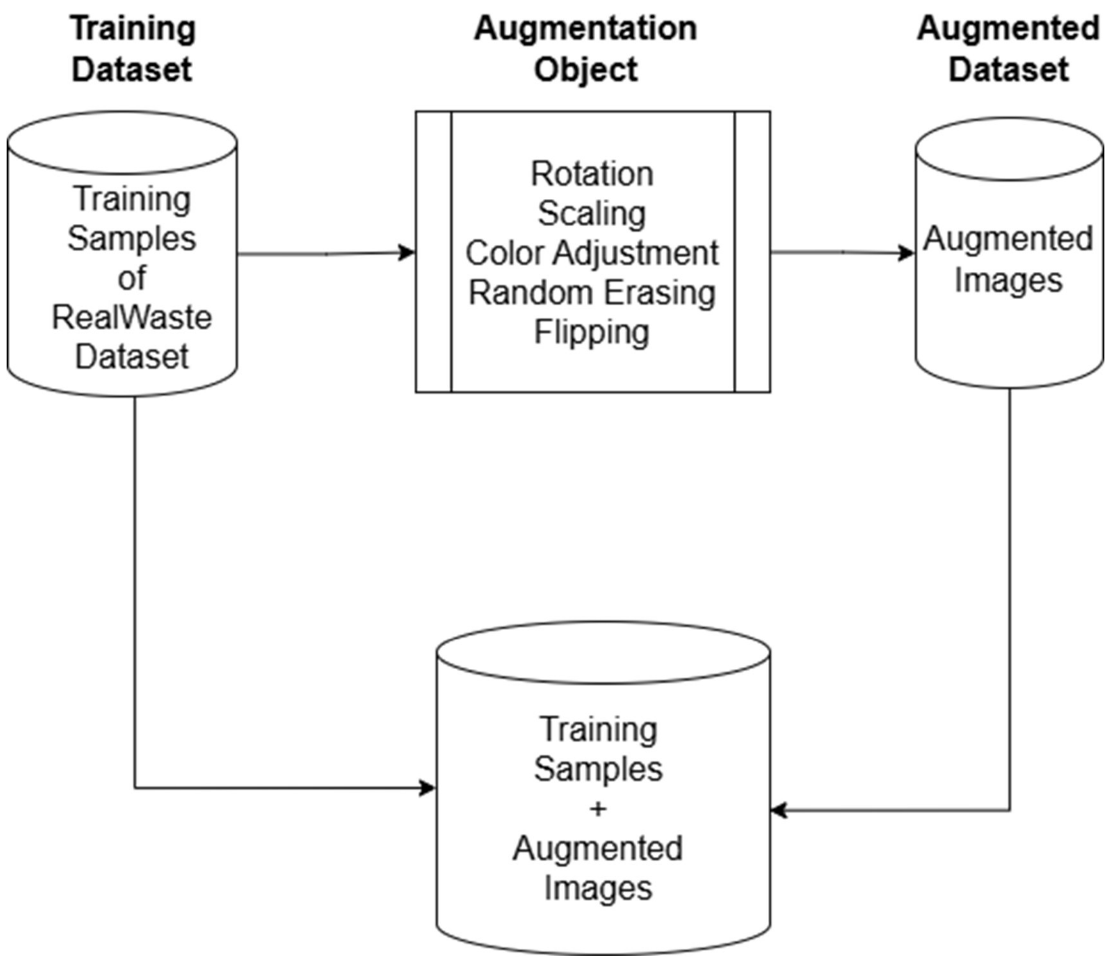


Figure 12: Block diagram illustrating the augmentation process.

3.6 Synthetic Data Generation Using GAN

To further augment the dataset and address the limitations of real-world data collection, synthetic waste images were generated using Generative Adversarial Networks (GANs). GANs are a class of deep learning models that consist of two neural networks, a generator and a discriminator, competing in a zero-sum game framework (Kaddoura, 2023).

In this project, a lightweight Generative Adversarial Network was employed due to its proven effectiveness in generating high-quality images. The generator took random noise as input and generated synthetic waste images, while the discriminator classified images as real or fake. The lightweight GAN was trained on the RealWaste dataset using 500 epochs, generating 16 images in each epoch. Figures 14, 15, 16, 17 and 18 show the images generated at every 100th epoch. Figure 19 shows a flow chart of the process.

The quality and diversity of the synthetic waste images was evaluated using the qualitative metrics which involve visually inspecting the images to assess how real they are and their resemblance to real waste items. A total of 990 synthetic images were generated by the lightweight GAN. The synthetic waste images generated were incorporated into the training dataset, along with the augmented images from the previous step. This combined dataset (6,192 training samples) as shown in figure 13, was used to train the CNN model for waste classification, with the expectation that the inclusion of synthetic data will improve the model's robustness, generalization ability, and overall performance. All the images generated in this project can be found in the Google drive link in Appendix A.

```
In [26]: X_train_augmented.shape
Out[26]: (6192, 80, 80, 3)

In [27]: y_train_augmented.shape
Out[27]: (6192,)
```

Figure 13: Shape of Dataset of data augmentation and GAN

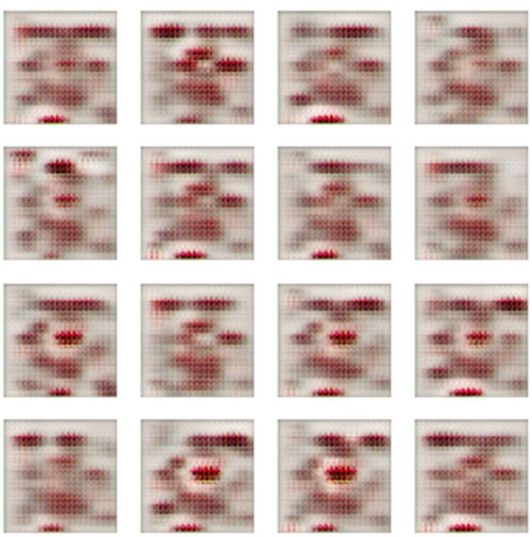


Figure 14: Images at 100 epochs

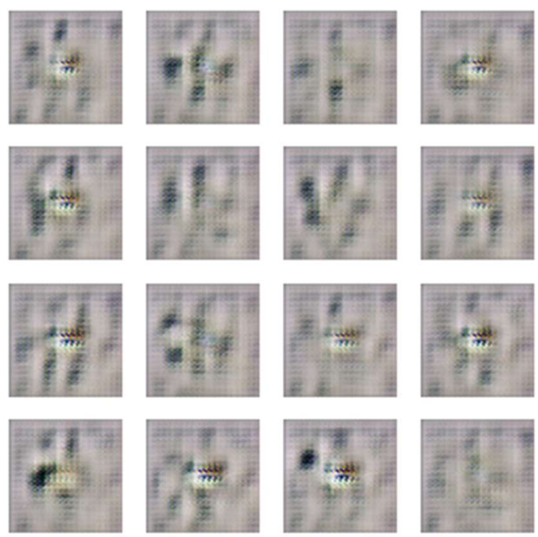


Figure 15: Images at 200 epochs

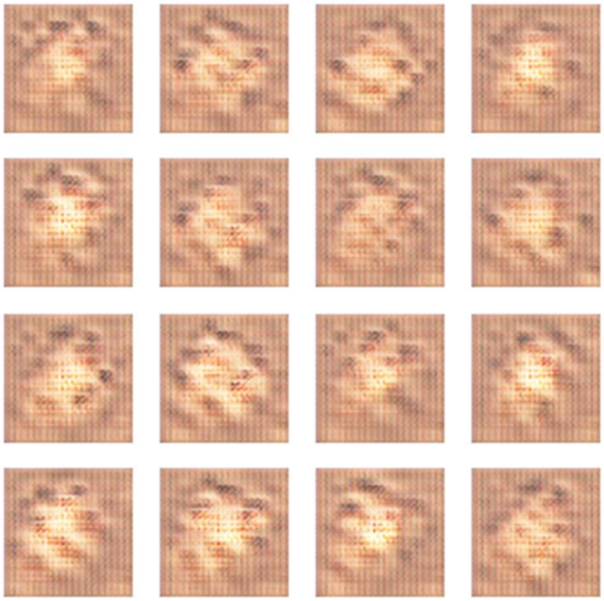


Figure 16: Images at 300 epochs

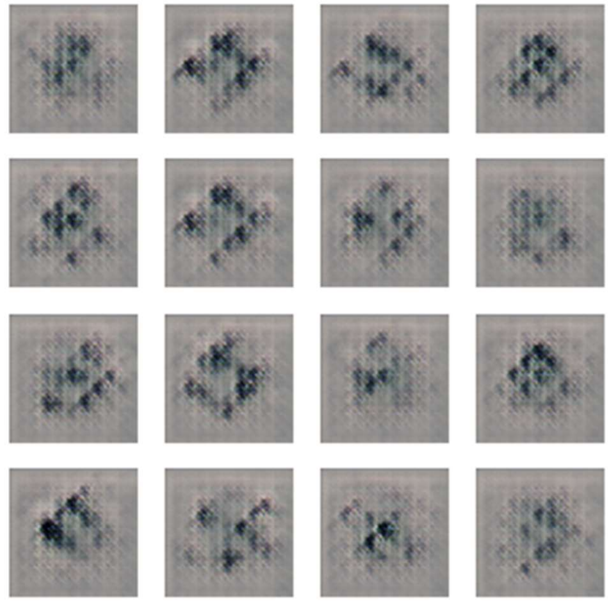


Figure 17: Images at 400 epochs

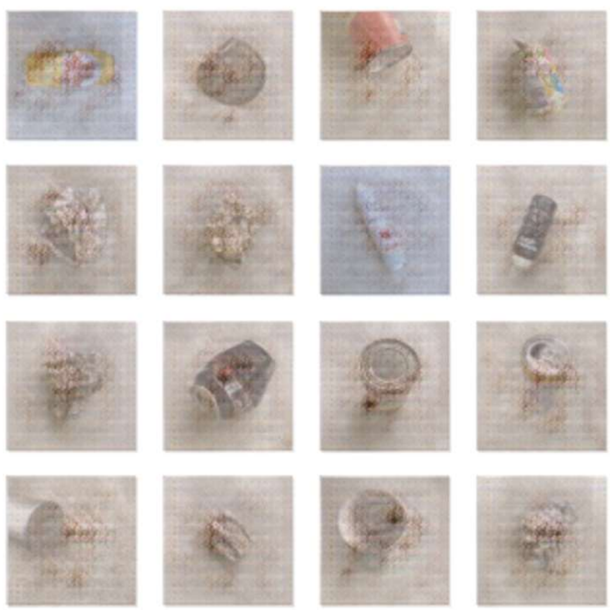


Figure 18: Images at 500 epochs

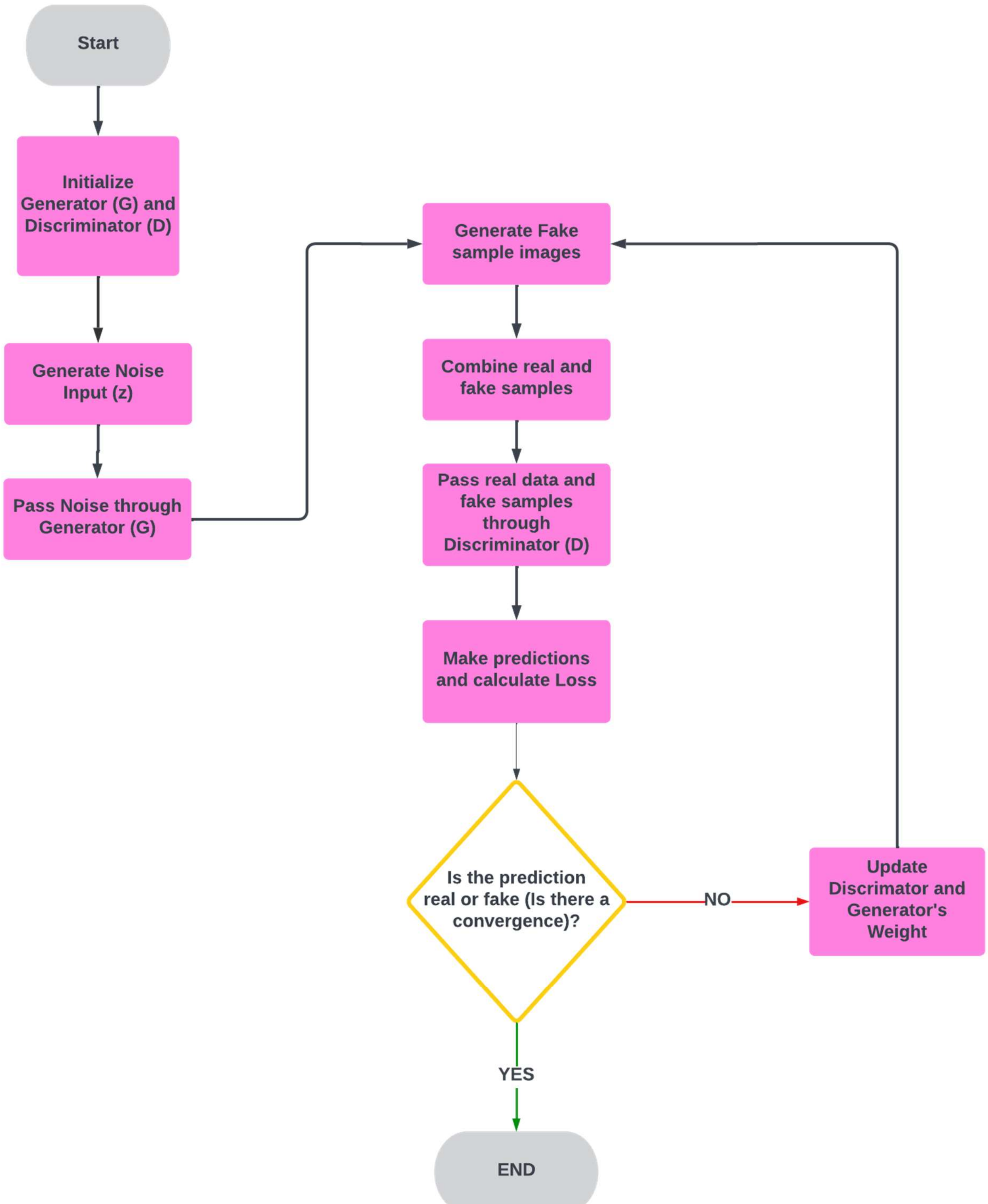


Figure 19: Flow chart of the Lightweight Generative Adversarial Network (GAN) Used.

3.7 Model Selection

In this project, the selection of an appropriate Convolutional Neural Network (CNN) model for waste classification is a critical step. After reviewing several literatures on the application of deep learning on image classification, four prominent CNN architectures, known for their effectiveness in image recognition tasks, were considered; VGG16, ResNet50, MobileNetV2 and InceptionV3. Figure 20 is a complete pictorial block diagram of the proposed system segregated into 5 blocks from the input to output block.

1. VGG16: A classic CNN architecture renowned for its simplicity and strong performance. It consists of multiple convolutional layers with small 3x3 filters, followed by max-pooling layers. This architecture enables the model to learn hierarchical representations of image features, leading to effective classification (Single et. al.,2023).
2. ResNet50: A deeper CNN architecture that utilizes residual connections to mitigate the vanishing gradient problem, allowing for the training of much deeper networks. ResNet50 has achieved state-of-the-art results in various image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Choudhary et al., 2024).
3. MobileNetV2: A lightweight CNN architecture designed for mobile and embedded vision applications. It utilizes depth-wise separable convolutions to reduce the number of parameters and computational cost significantly while maintaining competitive accuracy. This makes it an attractive option for real-time waste classification systems deployed on resource-constrained devices (Gulzar, 2023).
4. InceptionV3: This is an advanced CNN architecture that uses inception modules to capture multiscale features in images. It consists of convolutional layers with varying filter sizes in parallel, which allows the network to learn more complex and abstract features. It is known for its high accuracy and efficiency in image classification tasks, making it suitable for use in this project (Rao & Mahantesh, 2022).

The choice of the final model was determined based on a comprehensive evaluation of their performance on the combined dataset (real, synthetic and augmented images). The models were trained and validated using the same hyperparameters and optimization techniques to ensure a fair comparison. The model with optimal performance was selected as the final model for waste classification. Additionally, the computational efficiency of each model was considered, as it is a crucial factor for real-world deployment.

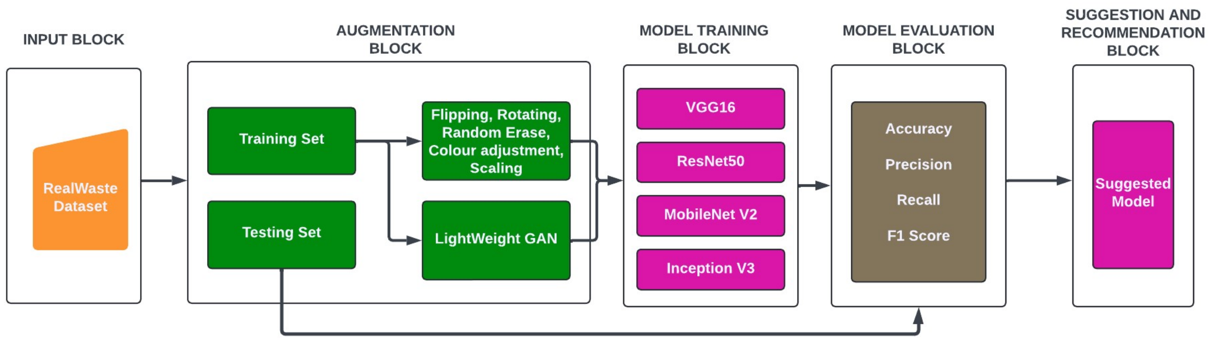


Figure 20: Block diagram of the proposed model for waste classification.

3.8 Model Training

The training process for each of the selected CNN models (VGG16, ResNet50, MobileNetV2, and InceptionV3) was conducted using the combined dataset, which includes both the original RealWaste images, augmented dataset and the synthetically generated images. The models were initialized with pre-trained weights on the ImageNet dataset. This approach leverages the knowledge learned from a vast and diverse dataset, enabling faster convergence and potentially better performance on the waste classification task.

The training process involved the following steps:

1. Model Initialization: Each model was initialized with pretrained weights from ImageNet.
2. Model Architecture: The last 6 layers for each pretrained model were unfrozen and allowed to be retrained for the purpose of fine-tuning. A global average pooling layer was used, and a dense layer with 256 units and ReLU activation was added, a dropout layer with a rate of 0.5 was included to prevent overfitting, an output layer with 9 units was also added which is same as the number of waste classes, and a SoftMax activation was added for multi-class classification. Formula (3) gotten from Cao and Xiang (2020), shows how softmax is calculated.

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j)} \qquad - \quad - \quad - \quad - \quad - \quad - \qquad (3)$$

3. Hyperparameter Tuning: The optimal hyperparameters for each model, including learning rate, batch size, and optimizer were selected based on initial experiments and fine-tuning. The Adam optimizer, known for its effectiveness in training deep learning models, was used as the baseline optimizer.
4. Loss Function: The sparse categorical cross-entropy loss function was employed to measure the discrepancy between the predicted probabilities and the true labels. This loss function is commonly used in multi-class classification tasks.
5. Training Epochs: The models were trained on 100 epochs. Early stopping was implemented to prevent overfitting. Early stopping monitors the validation loss and terminates training if the loss does not improve for a certain number of consecutive epochs.
6. Regularization: L2 regularization and dropout was used to mitigate overfitting and improve the model's generalization ability. Dropout randomly deactivates a fraction of neurons during training thereby promoting generalization.

The training process was conducted on the training set, while the test set was used to evaluate the model's performance. This performance was tracked on both the training and test sets to assess its learning progress and identify potential overfitting issues.

3.9 Model Evaluation Metric

The performance of the CNN models was evaluated using a comprehensive set of metrics commonly employed in multi-class classification tasks:

1. Accuracy: This metric measures the overall correctness of the model's predictions, calculated as the ratio of correctly classified waste images to the total number of images. While accuracy provides a general overview of performance, it may not be sufficient when dealing with imbalanced datasets, where some waste classes have significantly fewer samples than others (Ulutürk et al., 2023).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad - \quad - \quad - \quad - \quad - \quad - \qquad (4)$$

2. Precision: This metric focuses on the accuracy of positive predictions for each class, calculated as the ratio of True Positives (TP) to the sum of True Positives and False Positives (FP). High

precision indicates that the model is less likely to misclassify waste items into a particular category (Ulutürk et al., 2023).

$$\text{Precision} = \frac{TP}{TP + FP} \quad - \quad - \quad - \quad - \quad - \quad - \quad - \qquad (5)$$

3. Recall: It is also known as sensitivity. This metric measures the model's ability to identify all relevant instances of a particular class, calculated as the ratio of true positives to the sum of True Positives and False Negatives (FN). High recall indicates that the model is less likely to miss waste items belonging to a specific category (Ulutürk et al., 2023).

$$\text{Recall} = \frac{TP}{TP + FN} \quad - \quad - \quad - \quad - \quad - \quad - \quad - \qquad (6)$$

4. F1-Score: This metric provides a balanced measure of precision and recall, calculated as the harmonic mean of the two. The F1-score considers both the model's ability to correctly identify positive instances and its ability to avoid misclassifying negative instances (Ulutürk et al., 2023).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad - \; - \quad - \quad - \quad - \quad - \qquad (7)$$

By utilizing this combination of metrics, and the plotting of a Precision-Recall curve, a comprehensive evaluation of the CNN models' performance can be achieved, considering both overall accuracy and the specific performance for each waste class. This will enable a thorough assessment of the models' effectiveness in real-world waste classification scenarios, where the accurate identification of various waste types is crucial for efficient and sustainable waste management.

3.10 System Requirements

The development, training, and validation of deep learning models requires substantial computational power. The implementation of this project was carried out on a Dell G5 15 laptop equipped with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz processor, Nvidia GeForce RTX 2060 Graphics card, and 16GB RAM, ensuring sufficient computational capacity to handle the demanding requirements of the software development and testing processes.

3.11 Implementation Details

The implementation of this project was carried out on Linux operating system (Ubuntu). It used the TensorFlow framework with the Keras API, leveraging on the Jupyter Notebook environment for interactive development and analysis. TensorFlow is a widely used open-source machine learning library known for its flexibility, scalability, and comprehensive toolset for building and deploying deep learning models. Keras is a high-level API within TensorFlow that provides a user-friendly interface for defining and training neural networks, making it ideal for this project's scope.

The choice of TensorFlow and Keras is motivated by several factors. Firstly, TensorFlow's robust ecosystem offers a wide range of pre-trained models, including the VGG16, ResNet50, MobileNetV2 and InceptionV3 architectures to be evaluated in this project. Secondly, Keras simplifies the process of model definition, training, and evaluation, enabling efficient experimentation and rapid prototyping. Thirdly, TensorFlow's extensive documentation and active community support provide valuable resources for troubleshooting and optimization.

The implementation of this project involved defining the Convolutional Neural Network (CNN) models using the Keras Sequential API, compiling the models Adam optimizer and sparse categorical cross-entropy loss function, and training them on the combined dataset. This dataset included the original RealWaste images, augmented dataset, and synthetically generated images. Pre-trained models such as VGG16, ResNet50, MobileNetV2, and InceptionV3 were used with top layers unfrozen for fine-tuning. The code for this project was written in Python 3.9, utilizing several libraries such as NumPy for numerical operations, Matplotlib for visualization, TensorFlow and Keras for model implementation, training, and evaluation. Additionally, CUDA and cuDNN were used to leverage GPU acceleration for faster computation during model training. Table 3 shows the list of versions of the various dependencies used in the implementation of this project.

Table 3: Version of Dependencies used

Library	Version
TensorFlow	2.11.0
Keras	2.11.0
NumPy	1.24.0
Matplotlib	3.9.1
CUDA	10.1
cuDNN	8.9.7
Python	3.9.18

4. RESULTS AND ANALYSIS

This section presents the results of the various deep learning models discussed in Section 3. The results are described in detail to provide insights into their performance.

4.1 VGG16 Model

The VGG16 Model was evaluated to determine its performance in the classification of waste images into the various categories. The experiments were conducted both before and after augmentation to assess the impact of data augmentation on the model's performance.

The experiment done before data augmentation stopped at the 29th epoch out of 100epochs because early stopping criteria was specified during training to mitigate overfitting and this criterion has been met. While the Training done after data augmentation stopped at 44 epochs. The training result for both experiments is shown in Table 4.

Table 4: VGG16 training result for before and after data augmentation

Metrics	Before Data Augmentation	After Data Augmentation
Loss	0.1984	0.1827
Training Accuracy	0.9444	0.9837
Learning Rate	2.0000e-05	1.0000e-05

Upon evaluation of the test dataset, the VGG16 model achieved a test accuracy of 63.30% before data augmentation and 78.28% after data augmentation. The detailed classification report for both experiments as presented in Table 5, shows the model's performance across different classes with precision, recall and F1-score metrics.

Table 5: VGG16 Classification Report for both Before and After data augmentation

Class	**Precision**		**Recall**		**F1-Score**	
	Before	After	Before	After	Before	After
Cardboard	0.65	0.74	0.68	0.79	0.67	0.76
Food Organics	0.68	0.87	0.71	0.86	0.69	0.87
Glass	0.76	0.88	0.68	0.84	0.72	0.86
Metal	0.65	0.80	0.58	0.75	0.61	0.78
Misc Trash	0.40	0.62	0.42	0.57	0.41	0.60
Paper	0.65	0.72	0.58	0.81	0.61	0.76
Plastic	0.63	0.76	0.70	0.81	0.66	0.78
Textile	0.46	0.75	0.43	0.67	0.44	0.70
Vegetation	0.77	0.95	0.91	0.95	0.83	0.95

The precision-Recall (PR) curve for each class before and after augmentation is presented in Figures 21 and 22 respectively. Each curve shows the trade-off between precision and recall across different thresholds. The plot of training and validation accuracy over epochs, which is available in Figure 23 and 24, shows the model's performance trends throughout the training process before and after augmentation. The model demonstrated a robust performance in most classes, particularly in "Vegetation" and "Glass" classes, with precision of 0.77 and 0.76, respectively, and an F1-scores of 0.83 and 0.72 respectively before augmentation. After augmentation, the precision improved to 0.95 for the "Vegetation" class and 0.88 for the "Glass" class. However, there was a noticeable drop in performance for the "Miscellaneous Trash" class with a precision of 0.40 and an F1-score of 0.41 before augmentation, and a precision of 0.62 and F1-score of 0.60 after augmentation, indicating that there is a need for improvement in distinguishing this class from others.

Overall, the VGG16 model's performance after augmentation indicates a strong ability to classify waste images accurately, with an overall test accuracy of 78.28%. The significant improvement in accuracy and other metrics after data augmentation underscores the importance of this technique in enhancing the model's performance.

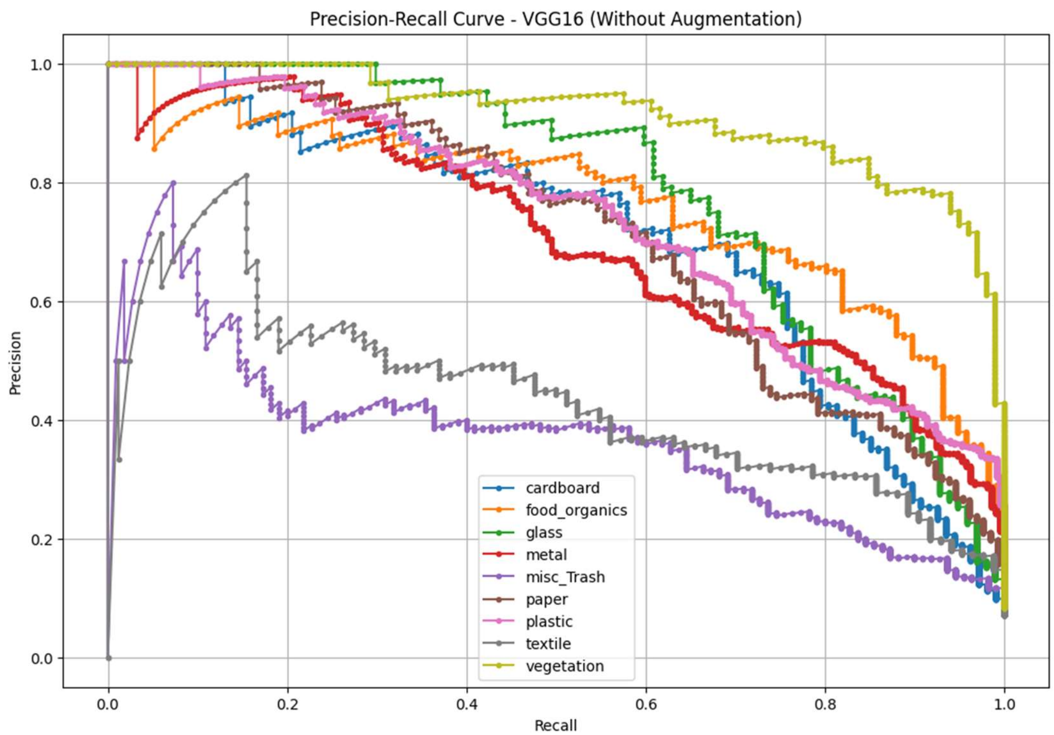


Figure 21: VGG16 Precision-Recall (PR) curve before data augmentation

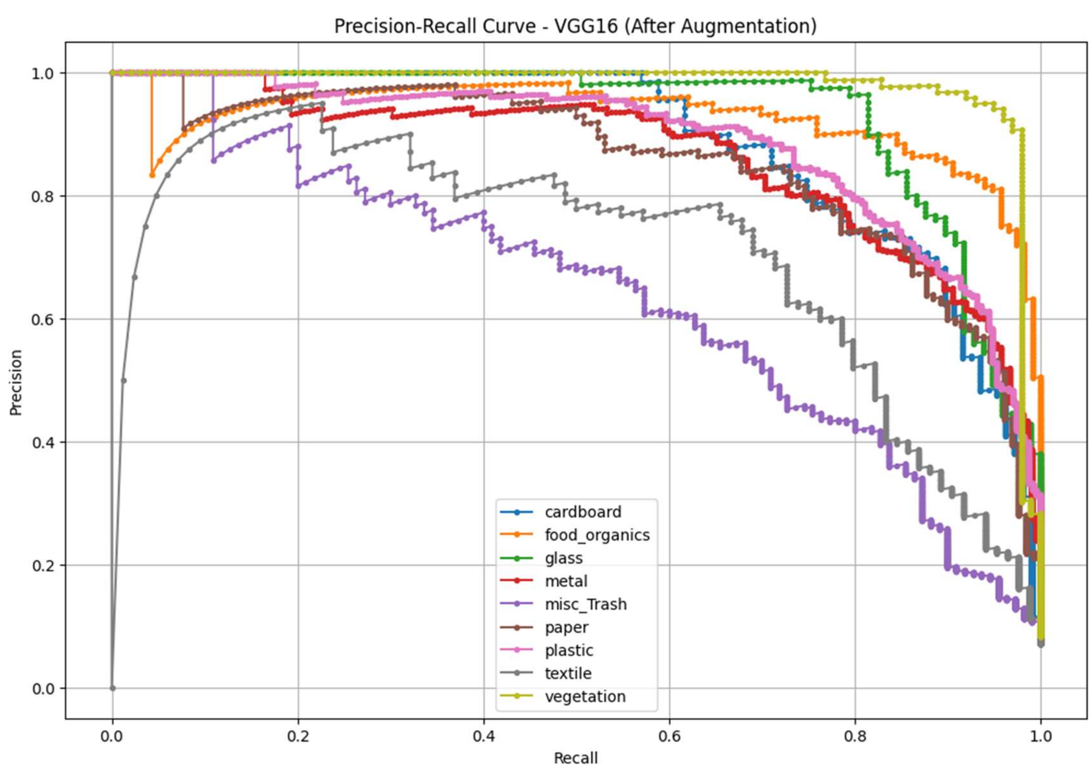


Figure 22: VGG16 Precision-Recall (PR) curve after data augmentation

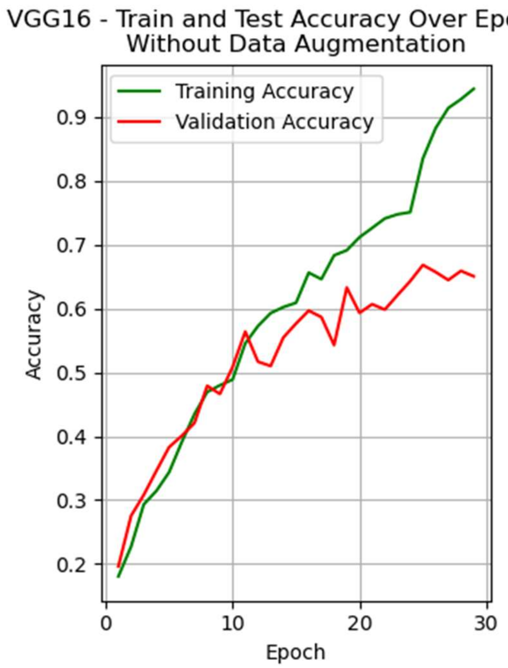


Figure 23: VGG16 Training and Testing Accuracy over Epochs (Without data Augmentation)

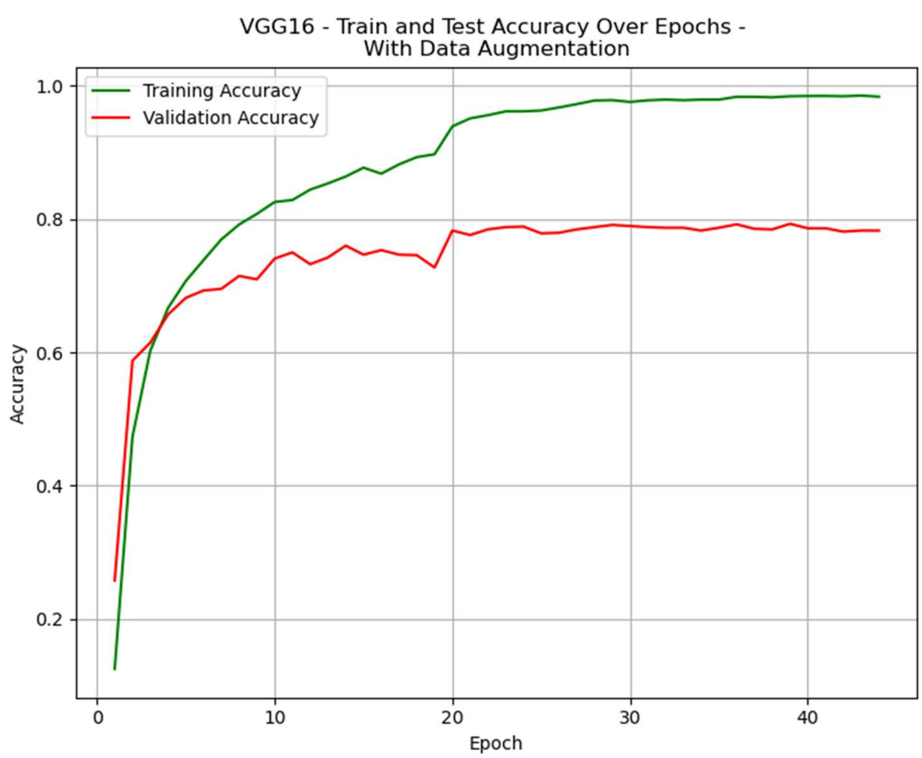


Figure 24: VGG16 Training and Testing Accuracy over Epochs (With data Augmentation)

4.2 ResNet50 Model

The training on the ResNet50 Model was also conducted for both before and after augmentation to assess the impact of data augmentation on the model's performance. The experiment conducted before data augmentation stopped at the 30th epoch out of 100 epochs due to the early stopping criteria specified during training to mitigate overfitting. Training after data augmentation halted at the 35th epoch for similar reasons. The training results for both experiments are shown in Table 6.

Table 6: ResNet50 Training Results for before and after data augmentation

Metrics	Before Data Augmentation	After Data Augmentation
Loss	0.0177	0.0089
Training Accuracy	0.9997	0.9998
Learning Rate	4.0000e-05	4.0000e-05

Upon evaluation, the ResNet50 model's performance on the test dataset improved with data augmentation, increasing from 79.63% accuracy to 80.47%. Table 7 provides a comprehensive breakdown of the model's performance across various classes, including precision, recall, and F1-score metrics for both scenarios (before and after data augmentation).

Table 7: ResNet50 classification Report for both before and after data augmentation.

Class	Precision		Recall		F1-Score	
	Before	After	Before	After	Before	After
Cardboard	0.75	0.74	0.82	0.84	0.79	0.79
Food Organics	0.90	0.90	0.79	0.79	0.84	0.84
Glass	0.93	0.89	0.80	0.87	0.86	0.88
Metal	0.82	0.79	0.81	0.84	0.81	0.81
Misc Trash	0.65	0.66	0.57	0.63	0.61	0.64
Paper	0.85	0.86	0.82	0.78	0.83	0.82
Plastic	0.76	0.81	0.85	0.83	0.80	0.82
Textile	0.67	0.71	0.65	0.65	0.66	0.68
Vegetation	0.85	0.88	0.96	0.95	0.90	0.91

Figures 25 and 26 illustrate the precision-recall (PR) curves for each class before and after data augmentation, respectively. These curves visually depict the trade-offs between precision and recall at various probability thresholds. The plot of training and validation accuracy over epochs, which is available in Figure 27 and 28, shows the model's performance trends throughout the training process before and after augmentation. In general, the model exhibited strong performance across most waste categories. Notably, the "Vegetation" and "Glass" classes also stood out as in the case of VGG16 model, achieving precision rates of 0.85 and 0.93, respectively, alongside F1-scores of 0.90 and 0.86 before augmentation. The inclusion of data augmentation further enhanced precision in these classes, reaching 0.88 for "Vegetation" and 0.89 for "Glass." On the other hand, the "Miscellaneous Trash" class presented a challenge for the model. It exhibited lower precision (0.65) and an F1-score of 0.61 before augmentation. While augmentation slightly improved the precision to 0.66, the F1-score remained relatively low at 0.64.

The overall test accuracy of 80.47% reflects the model's robust performance in the classification task. The PR curves and detailed metrics indicate strong classification abilities in several categories while also identifying specific areas that need further refinement.

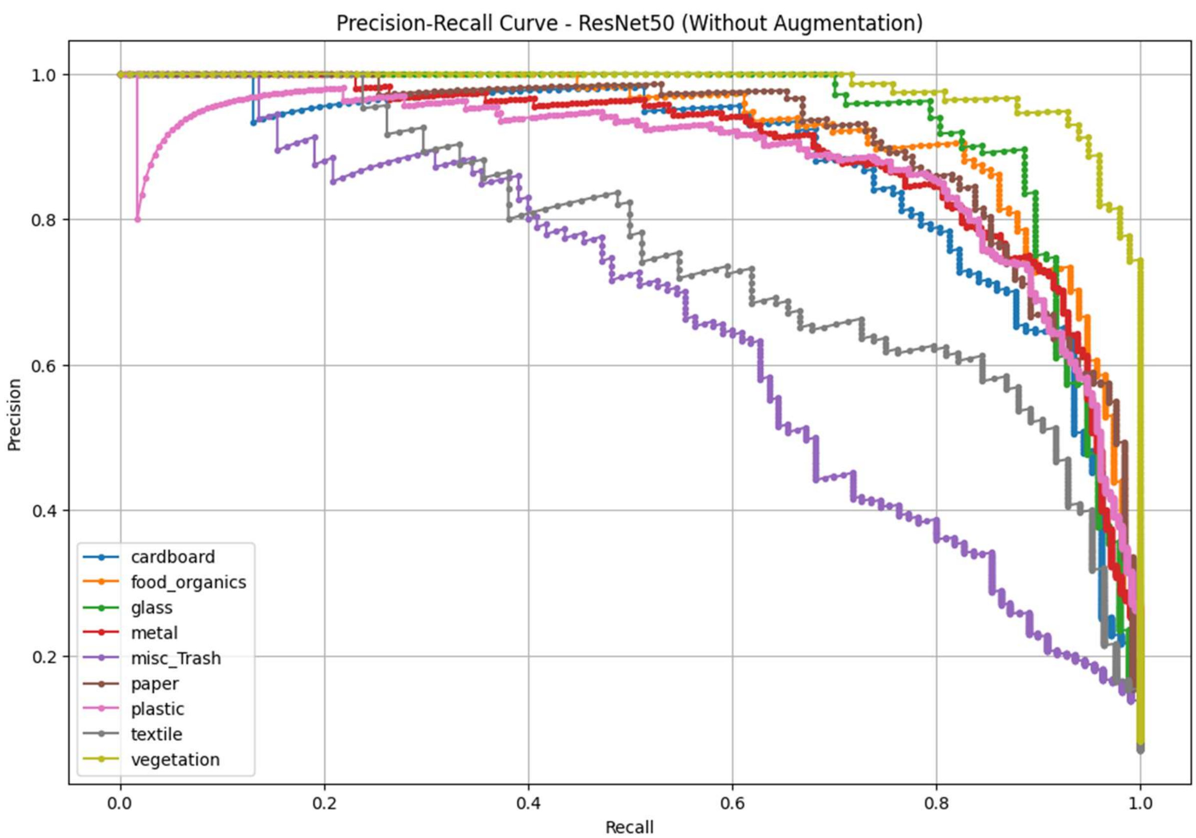


Figure 25: Precision-Recall (PR) curve of the ResNet50 model before data augmentation.

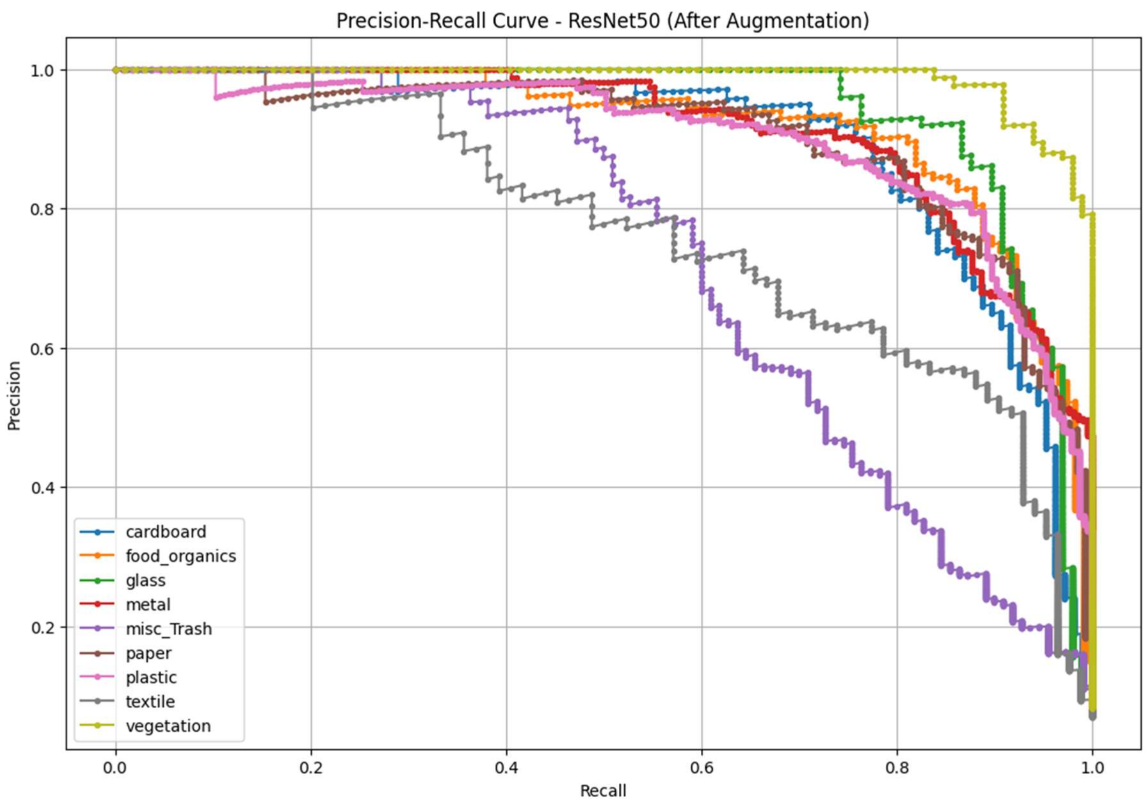


Figure 26: Precision-Recall (PR) curve of the ResNet50 model after data augmentation.

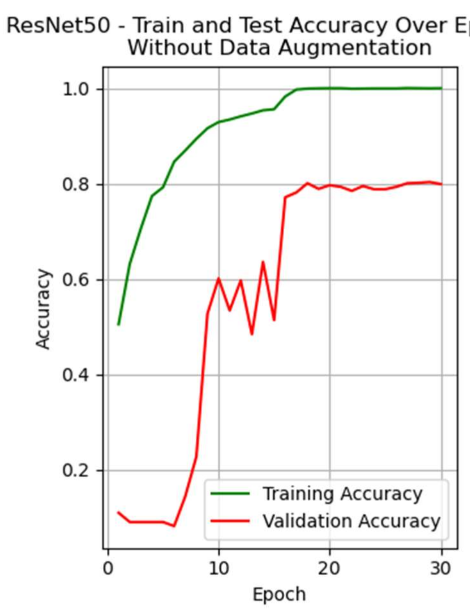


Figure 27: ResNet50 Training and Testing Accuracy over Epochs (Without data Augmentation)

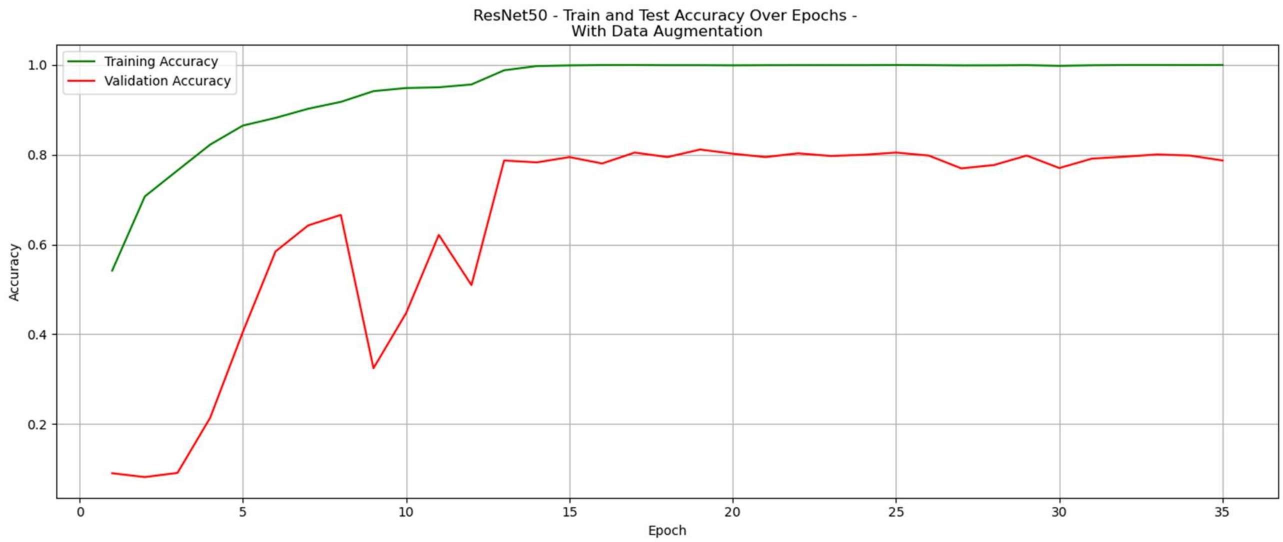


Figure 28: ResNet50 Training and Testing Accuracy over Epochs (With data Augmentation)

4.3 MobileNetV2 Model

The experiments done on MobileNetV2 model training were also conducted for both before and after augmentation. Early stopping criteria were applied to prevent overfitting. The model trained for 16 epochs with the experiment done before data augmentation and 55 epochs for the experiments after data augmentation. Details of the training result are shown in Table 8.

Table 8: Training result of the MobileNetV2 Model for before and after data augmentation

Metrics	Before Data Augmentation	After Data Augmentation
Loss	0.1369	0.0145
Training Accuracy	0.9871	0.9997
Learning Rate	2.0000e-04	1.0000e-05

Evaluation was done on the test dataset to know how well the model will perform on unseen data. The model achieved a test accuracy of 35.94% before data augmentation and 83.18% after data augmentation. The detailed classification report for both experiments is presented in Table 9.

Table 9: Classification Report for MobileNetV2 Model for both before and after data augmentation.

Class	**Precision**		**Recall**		**F1-Score**	
	Before	After	Before	After	Before	After
Cardboard	0.22	0.79	0.76	0.89	0.34	0.84
Food Organics	0.63	0.83	0.32	0.83	0.42	0.83
Glass	0.00	0.88	0.00	0.86	0.00	0.87
Metal	0.39	0.83	0.77	0.84	0.52	0.83
Misc Trash	0.21	0.71	0.35	0.70	0.27	0.71
Paper	0.55	0.86	0.14	0.86	0.22	0.86
Plastic	0.64	0.86	0.04	0.80	0.07	0.83
Textile	0.00	0.79	0.00	0.77	0.00	0.78
Vegetation	0.68	0.90	0.80	0.96	0.73	0.93

The precision-recall (PR) curve for each class before and after augmentation is presented in Figures 29 and 30, respectively. The plot of training and validation accuracy over epochs, is available in Figure 31 and 32 and it shows the model's performance trends throughout the training process before and after augmentation. The test accuracy increased significantly from 35.94% before augmentation to 83.18% after augmentation. There were also substantial improvements in precision, recall, and F1-scores for almost all classes. Notably, the "Vegetation" class had a precision of 0.68 and an F1-score of 0.73 before augmentation, which improved to 0.90 for both metrics after augmentation. The "Glass" class had very poor performance before augmentation, with precision and F1-scores both at 0.00, indicating it could not be correctly classified at all. However, after augmentation, the precision and F1-score for the "Glass" class improved dramatically to 0.88 and 0.87, respectively. The substantial improvement in accuracy and other metrics after data augmentation proves the importance of this technique in enhancing the model's performance.

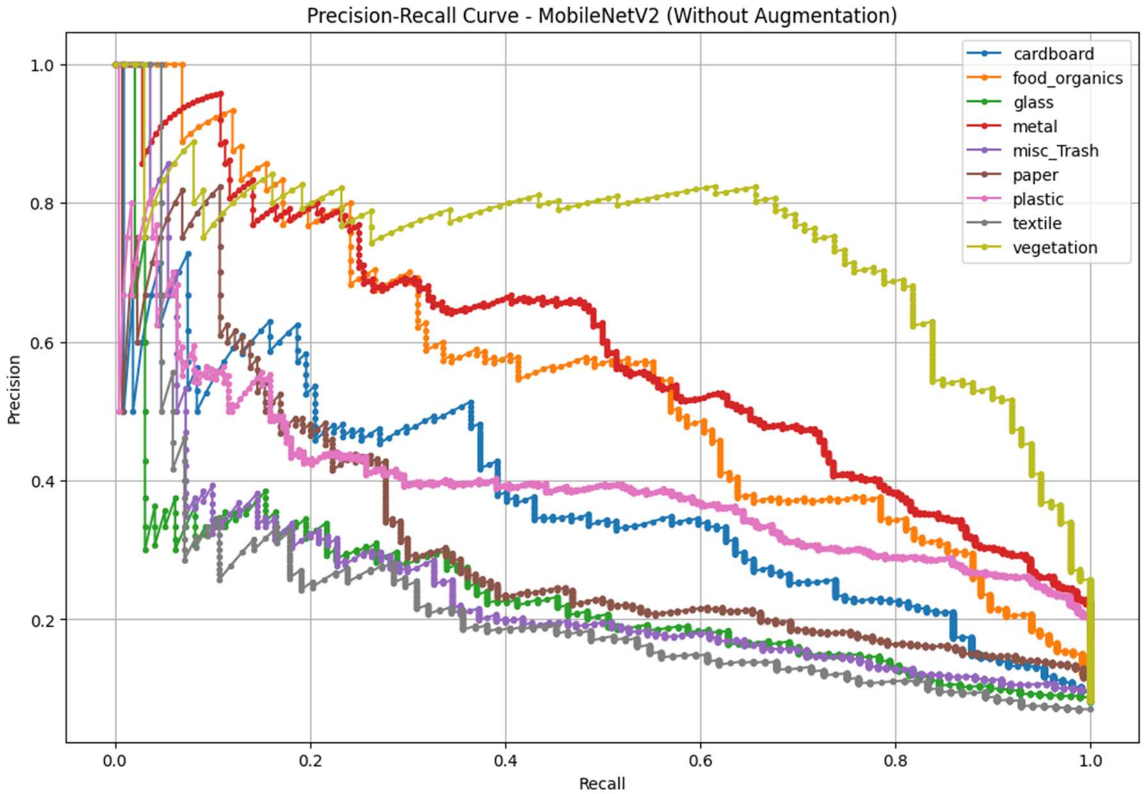


Figure 29: Precision-Recall (PR) curve of the MobileNetV2 model before data augmentation.

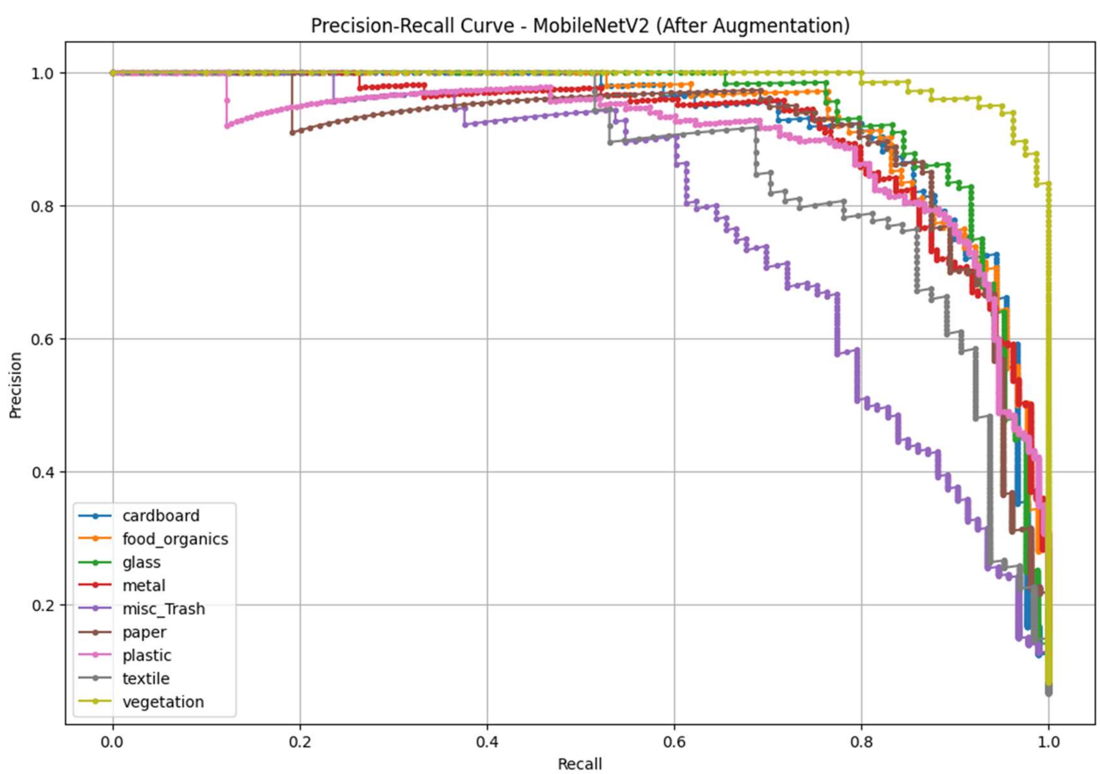


Figure 30: Precision-Recall (PR) curve of the MobileNetV2 model after data augmentation.

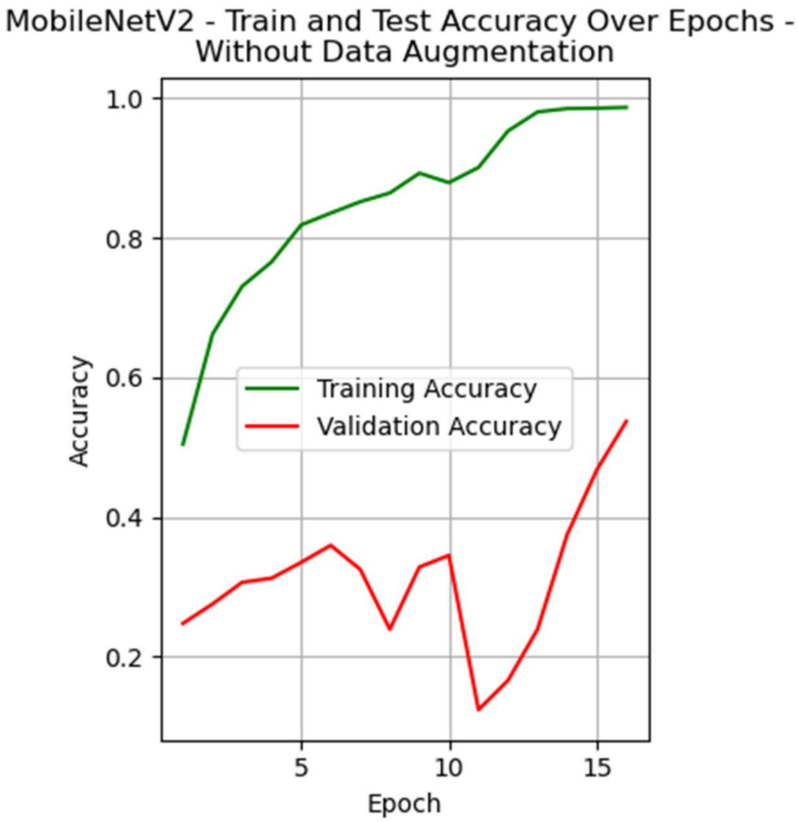


Figure 31: MobileNetV2 Training and Testing Accuracy over Epochs (Without data Augmentation)

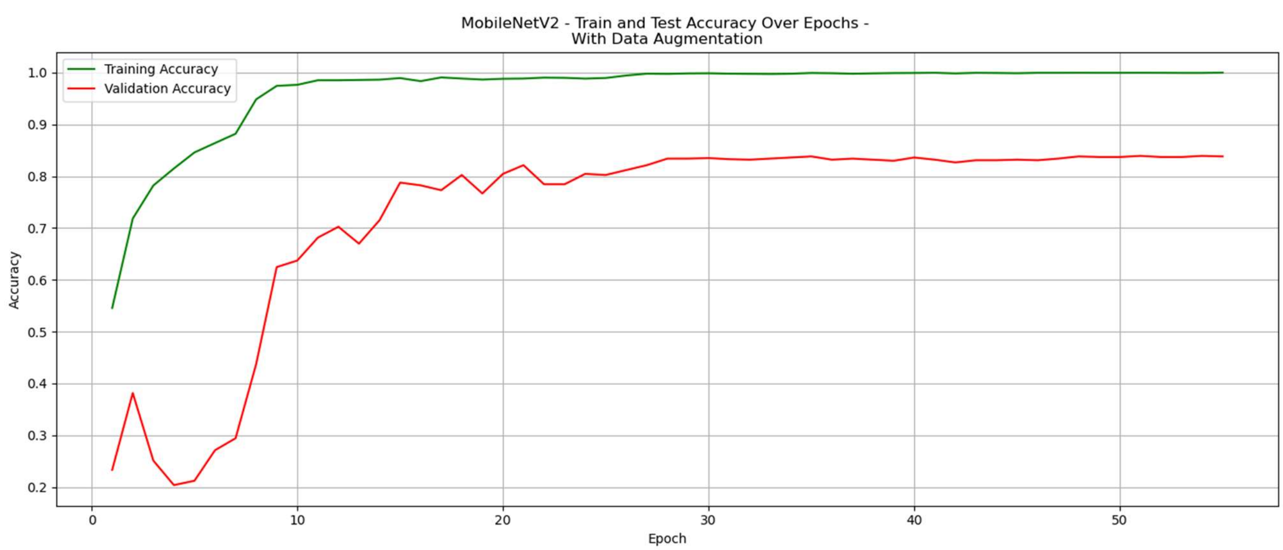


Figure 32: MobileNetV2 Training and Testing Accuracy over Epochs (With data Augmentation)

4.4 InceptionV3 Model

The InceptionV3 model was also evaluated to determine its performance in classifying waste images. Experiments were also performed on data before augmentation and after augmentation in order to effectively assess the impact of augmentation on the model's performance. Before data augmentation, the model training halted at the 35th epoch out of 100 epochs due to early stopping criteria. After data augmentation, the training stopped at the 55th epoch. The training results for both experiments is summarized in Table 10.

Table 10: InceptionV3 model Training results for before and after data augmentation

Metrics	Before Data Augmentation	After Data Augmentation
Loss	0.4112	0.3029
Training Accuracy	0.8681	0.9097
Learning Rate	4.0000e-05	1.0000e-05

Upon evaluating the test dataset, the Inception V3 model achieved a test accuracy of 69.19% before data augmentation and 73.92% after data augmentation. The detailed classification report for both experiments is presented in Table 11, showing the model's performance across different classes with precision, recall, and F1-score metrics.

Table 11: Classification Report for InceptionV3 Model for both before and after data augmentation

Class	Precision		Recall		F1-Score	
	Before	After	Before	After	Before	After
Cardboard	0.66	0.69	0.72	0.80	0.69	0.74
Food Organics	0.83	0.84	0.76	0.85	0.76	0.85
Glass	0.76	0.88	0.77	0.79	0.77	0.83
Metal	0.76	0.73	0.66	0.69	0.71	0.71
Misc Trash	0.50	0.50	0.52	0.63	0.51	0.56
Paper	0.64	0.70	0.62	0.73	0.63	0.72
Plastic	0.63	0.86	0.72	0.68	0.67	0.76
Textile	0.63	0.55	0.55	0.69	0.59	0.61
Vegetation	0.88	0.94	0.92	0.93	0.90	0.93

The Precision-Recall (PR) curves for each class before and after data augmentation are presented in Figures 33 and 34, respectively. The plot of training and validation accuracy over several epochs is available in Figure 35 and 36 and it shows the model's performance trends throughout the training process before and after augmentation. There were significant improvements in precision, recall and F1-scores for almost all classes. For instance, the "Vegetation" class had a precision of 0.88 and an F1-score of 0.90 before augmentation, which improved to 0.94 and 0.93, respectively, after augmentation. The "Glass" class also showed notable improvement, with precision increasing from 0.76 to 0.88 and the F1-score from 0.77 to 0.83.

Overall, the Inception V3 model's performance after augmentation indicates a strong ability to classify waste images accurately. The substantial improvement in accuracy and other metrics after data augmentation highlights the importance of this technique in enhancing the model's performance.

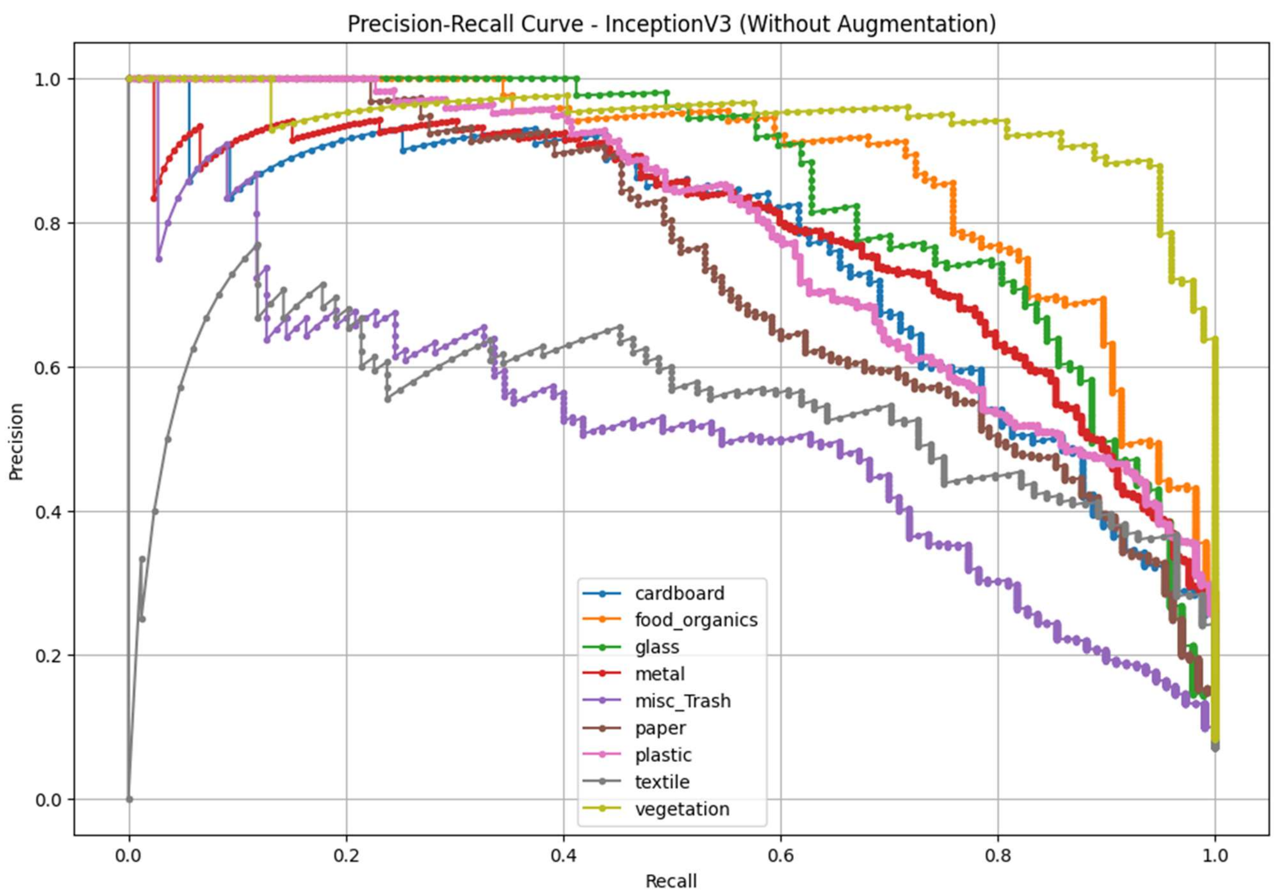


Figure 33: Precision-Recall (PR) curve of the InceptionV3 model before data augmentation

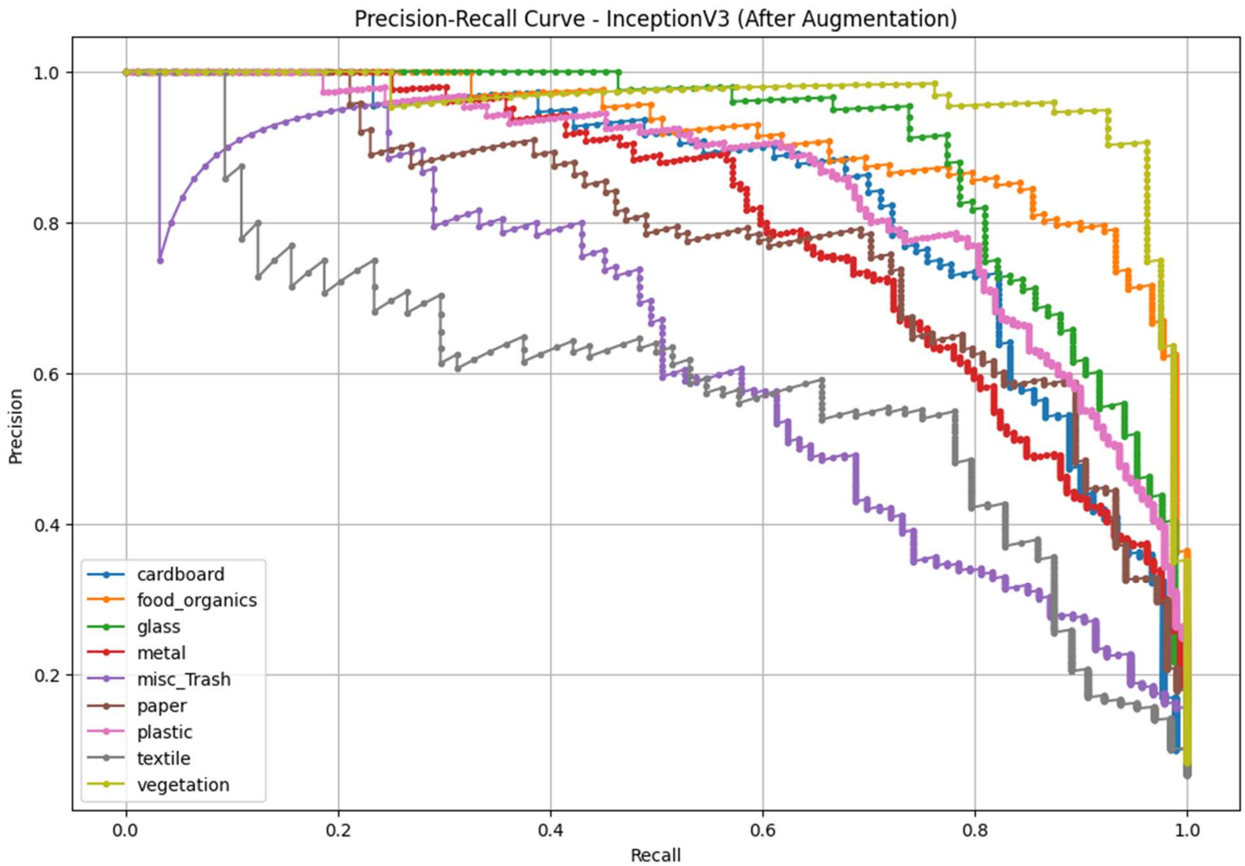


Figure 34: Precision-Recall (PR) curve of the InceptionV3 model after data augmentation

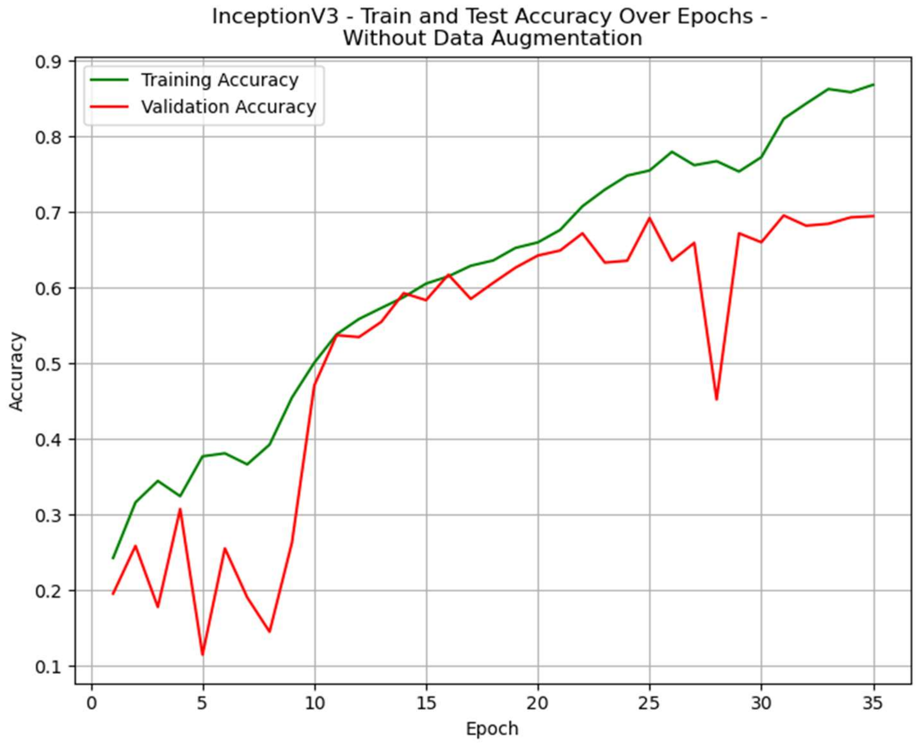


Figure 35: InceptionV3 Training and Testing Accuracy over Epochs (Without data Augmentation)

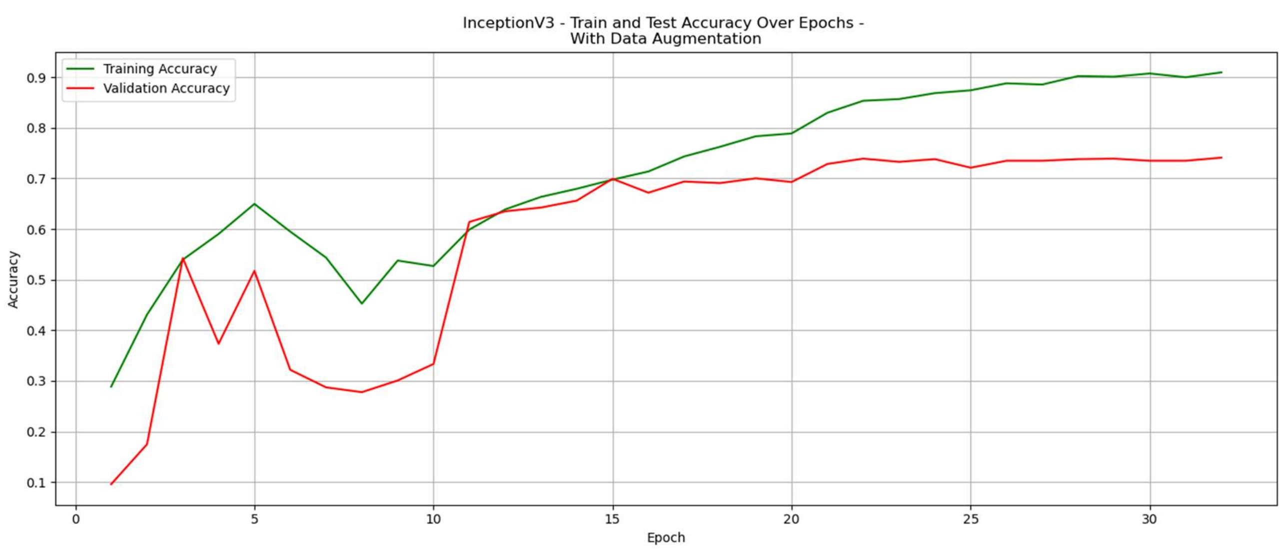


Figure 36: InceptionV3 Training and Testing Accuracy over Epochs (With data Augmentation)

5 DISCUSSIONS

The experimental results and analysis presented in Section 4 provide valuable insights into the performance of various deep learning models (VGG16, ResNet50, MobileNetV2, and InceptionV3) for waste image classification.

Across all models, data augmentation consistently demonstrated a significant positive impact on classification accuracy, highlighting its crucial role in mitigating overfitting and improving generalization. A summary of results and classification report for all models is presented in Tables 12 and 13 respectively. The VGG16 model, despite being the least complex model, achieved a notable accuracy of 78.28% after augmentation, showcasing the effectiveness of data augmentation even for simpler architectures. Whereas ResNet50, which is known for its residual connections, showcased a high baseline accuracy of 79.63%, which further improved to 80.47% with augmentation. This model consistently performed well across most waste categories, demonstrating its robustness and generalization capability.

The MobileNetV2 model, designed mostly for mobile devices, exhibited the most significant improvement with data augmentation, jumping from a low 35.94% to an impressive 83.18% accuracy. On the other hand, InceptionV3, with its complex inception modules, consistently performed well, achieving 69.19% accuracy before augmentation and 73.92% after augmentation. Although, the improvement with augmentation was less drastic compared to MobileNetV2, it still proves the benefit of data augmentation for complex models and in scenarios with limited dataset.

Notably, the results attained for InceptionV3 model were a bit different with the findings of Single et al. (2023), who utilized the same RealWaste dataset to evaluate CNNs for real-life landfill waste classification. In their study, the InceptionV3 model achieved 89.19% in classification accuracy. However, it's important to note that this project benefited from a more elaborate dataset. The incorporation of data augmentation and synthetic image generation techniques have contributed to better model generalization, robustness and performance compared to the result by Single et al. (2023).

In conclusion, MobileNetV2 and ResNet50 model trained with augmented data outperformed other models, with MobileNetV2 achieving the highest test accuracy of 83.18% and demonstrating superior waste image classification capabilities. Its design for mobile devices ensures computational efficiency, reducing hardware and energy costs. The substantial performance improvement after augmentation showcases its ability to generalize well from the augmented data, which is a crucial factor in real-world waste classification scenarios. Additionally, its lightweight architecture allows for easy deployment and scalability for future expansion.

This study successfully demonstrated the effectiveness of various deep learning models for waste image classification. Data augmentation consistently improved performance, underscoring its importance in enhancing model generalization. The findings pave the way for the development of robust and accurate waste classification systems, contributing to sustainable waste management practices.

Table 12: Summary of Results

Model	**VGG16**	**ResNet50**	**MobileNetV2**	**InceptionV3**
Accuracy before Augmentation	63.30%	79.63%	35.94%	69.19%
Accuracy After Augmentation	78.28%	80.47%	83.18%	73.92%

Table 13: Summary of classification report for all Models before and after data Augmentation

Model	**Metric**	**Cardboard**	**Food Organics**	**Glass**	**Metal**	**Misc Trash**	**Paper**	**Plastic**	**Textile**	**Vegetation**
VGG16 Before Augmentation	Precision	0.65	0.68	0.76	0.65	0.40	0.65	0.63	0.46	0.77
	Recall	0.68	0.71	0.68	0.58	0.42	0.58	0.70	0.43	0.91
	F1-Score	0.67	0.69	0.72	0.61	0.41	0.61	0.66	0.44	0.83
VGG16 After Augmentation	Precision	0.74	0.87	0.88	0.80	0.62	0.72	0.76	0.75	0.95
	Recall	0.79	0.86	0.84	0.75	0.57	0.81	0.81	0.67	0.95
	F1-Score	0.76	0.87	0.86	0.78	0.60	0.76	0.78	0.70	0.95
ResNet50 Before Augmentation	Precision	0.75	0.90	0.93	0.82	0.65	0.85	0.76	0.67	0.85
	Recall	0.82	0.79	0.80	0.81	0.57	0.82	0.85	0.65	0.96
	F1-Score	0.79	0.84	0.86	0.81	0.61	0.83	0.80	0.66	0.90
ResNet50 After Augmentation	Precision	0.74	0.90	0.89	0.79	0.66	0.86	0.81	0.71	0.88
	Recall	0.84	0.79	0.87	0.84	0.63	0.78	0.83	0.65	0.95
	F1-Score	0.79	0.84	0.88	0.81	0.64	0.82	0.82	0.68	0.91
MobileNetV2 Before Augmentation	Precision	0.22	0.63	0.00	0.39	0.21	0.55	0.64	0.00	0.68
	Recall	0.76	0.32	0.00	0.77	0.35	0.14	0.04	0.00	0.80
	F1-Score	0.34	0.42	0.00	0.52	0.27	0.22	0.07	0.00	0.73
MobileNetV2 After Augmentation	Precision	0.79	0.83	0.88	0.83	0.71	0.86	0.86	0.79	0.90
	Recall	0.89	0.83	0.86	0.84	0.70	0.86	0.80	0.77	0.96
	F1-Score	0.84	0.83	0.87	0.83	0.71	0.86	0.83	0.78	0.93
Inception V3 Before Augmentation	Precision	0.66	0.83	0.76	0.76	0.50	0.64	0.63	0.63	0.88
	Recall	0.72	0.76	0.77	0.66	0.52	0.62	0.72	0.55	0.92
	F1-Score	0.69	0.76	0.77	0.71	0.51	0.63	0.67	0.59	0.90
Inception V3 After Augmentation	Precision	0.69	0.84	0.88	0.73	0.50	0.70	0.86	0.55	0.94
	Recall	0.80	0.85	0.79	0.69	0.63	0.73	0.68	0.69	0.93
	F1-Score	0.74	0.85	0.83	0.71	0.56	0.72	0.76	0.61	0.93

6. PROJECT MANAGEMENT

6.1 Project Schedule

For the successful completion of any project, a plan or project schedule is essential. In the early stages of this project, a Gantt chart was created to outline the schedule, spanning from the ethical application and approval phase to the project completion phase. The initially designed Gantt chart for this project is shown in Figure 37, with start and end dates from May 16th to July 25th, 2024.

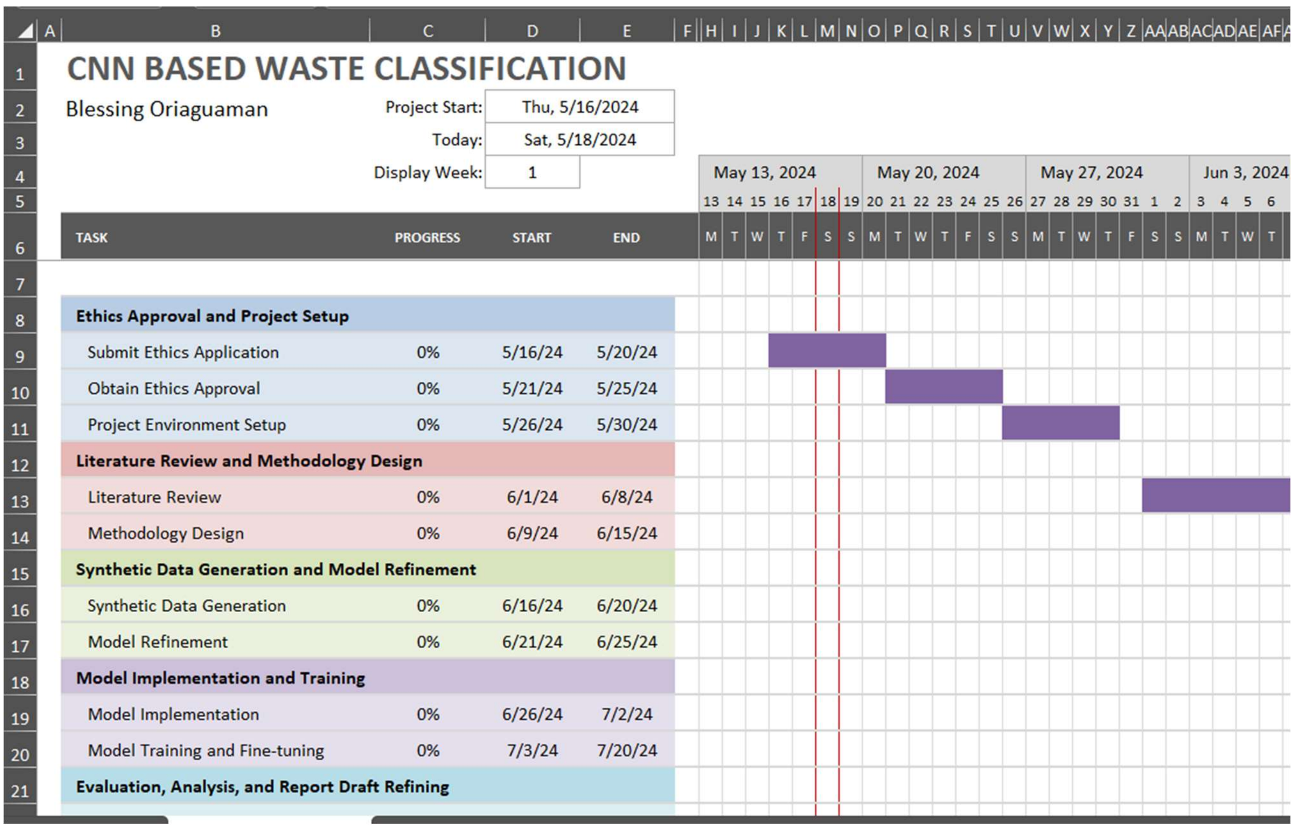


Figure 37: Initial Gantt Chart design for project management

The Gantt chart illustrates that every project has deadlines that need to be met. However, the chart had to be rescheduled due to unforeseen circumstances. For instance, the allocation of the project supervisor was done at the end of the second week of the semester. Additionally, obtaining ethical approval and training the model took longer than anticipated. The actual project plan followed is shown in the Gantt chart in Figure 38, which details the project breakdown and timelines. All necessary adjustments were made to the project plan to accommodate the lost time. There were also meeting records (Appendix B) to keep track of what was discussed, and the advice given by the supervisor.

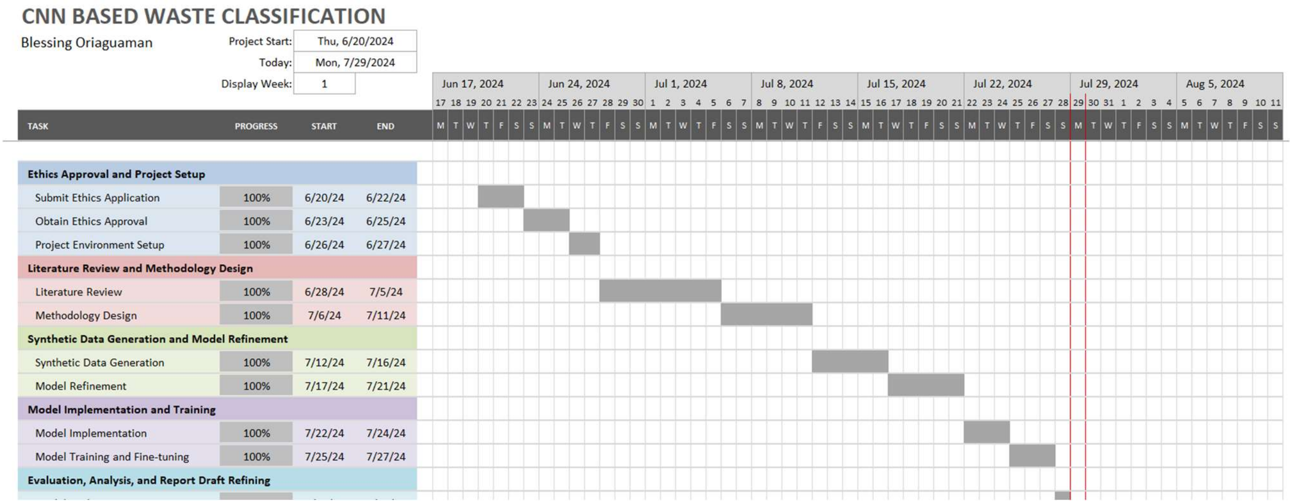


Figure 38: Actual project plan followed for this research.

6.2 Risk Management

Effective risk management is crucial for the successful completion of any project. It involves identifying potential risks that could negatively impact the project, assessing their likelihood and potential consequences, and developing strategies to mitigate these risks. This proactive approach helps in minimizing disruptions and ensures that the project stays on track. Table 14 outlines the possible risks identified for this project, their potential impact, and the corresponding mitigation strategies.

Table 14: Possible risk and their mitigation strategies

S/N	Risk	Description	Impact	Likelihood	Mitigation Strategy
1	Technical Issues with Software	Software bugs or incompatibility issues can disrupt project progress	High	Medium	Conduct thorough testing in early phases, maintain regular backups, and have technical support on standby.
2	Loss of Data	Data loss due to accidental deletion, corruption, or cyberattacks	High	Medium	Implement regular backups, use reliable storage solutions, and employ data recovery tools.
3	Faulty/Loss of System	Hardware failures or loss of critical systems can halt project activities	High	Low	Maintain hardware redundancy, perform regular maintenance checks, and have contingency plans for system recovery.

From the anticipated risks, the one listed as S/N 1(Technical issues with software) occurred. Its impact on the project plan was low. There was an issue connecting my GPU to the TensorFlow environment. I had to seek technical support from an expert to resolve the problem.

6.3 Quality Management

Ensuring the highest quality throughout a project lifecycle is crucial for achieving the desired outcomes and meeting stakeholders' expectations. To ensure quality was not compromised in the delivery of this project, the project's framework, flowcharts, and blocks were created as shown in the methodology section of this report. This plan was followed from design to implementation to ensure it met the required standards. Additionally, feedback and suggestions were provided by the supervisor to ensure it met the expected quality. Appendix B shows the meeting reports with my supervisor.

6.4 Social, Legal, Ethical and Professional Considerations

In any machine learning project, addressing social, legal, and ethical considerations is essential, especially when dealing with datasets that may contain sensitive or personal information. For this project, which focuses on waste classification using Convolutional Neural Networks (CNNs), these concerns are minimal as the dataset used does not include sensitive information.

The dataset utilized for this project was sourced from publicly available online repositories (UCI) and is freely accessible for study and research purposes. It comprises images of various types of waste and does not contain any personally identifiable information or sensitive data that could impact individuals or raise ethical concerns.

Furthermore, the project adhered to ethical standards by obtaining formal ethical approval. The Ethical Approval certificate, provided in Appendix C, confirms that the project's use of data complies with relevant ethical guidelines and legal requirements. Throughout the research, all sources of information were appropriately cited and credited, ensuring transparency and academic integrity.

By following these protocols, the project upholds the highest standards of social responsibility, legal compliance, and ethical practice, ensuring that the research is conducted professionally and respectfully.

7. CONCLUSIONS

7.1 Achievements

This research aimed to evaluate the effectiveness of Convolutional Neural Networks (CNNs) for waste classification in comparison to traditional methods used by Wolverhampton Council. One major achievement of note was that other research either used only GANs or traditional augmentation process to increase dataset size, but this project combined both processes in a single framework. Also, by addressing the research questions and objectives outlined at the beginning of the project, several significant achievements were realized.

1. **Comparison with Traditional Methods**

   Objective: To compare the cost efficiency of CNN-based waste classification with traditional separation technologies used by Wolverhampton Council.

   Achievement: The research demonstrated that CNN-based waste classification could potentially offer improved cost efficiency over traditional methods. Through comprehensive cost analysis, it was shown that while initial implementation costs for CNN systems might be higher, long-term savings in operational and maintenance costs could make CNN-based solutions more economically viable in the long run. This finding supports the feasibility of adopting CNNs for waste classification within municipal settings.

2. **Addressing Dataset Limitations**

   Objective: To examine the limitations of existing CNN models for waste classification, focusing on dataset deficiencies.

   Achievement: The study identified specific limitations in existing CNN models related to dataset deficiencies, such as class imbalance and insufficient image diversity. By implementing data augmentation techniques and synthetic data generation, these limitations were effectively addressed, leading to significant improvements in model robustness.

3. **Model Integration and Performance Validation**

   Objective: To integrate the augmented and synthetic datasets into the CNN training process and validate the model's performance.

Achievement: The integration of both augmented and synthetic datasets into the CNN training process led to a marked improvement in model performance. The validation results showed that the CNN models achieved high accuracy and robustness, demonstrating the effectiveness of the enhanced training datasets in overcoming real-world challenges in waste classification.

4. **Impact Analysis of Data Augmentation and Synthetic Data**

Objective: To analyze the impact of data augmentation and synthetic data generation on the accuracy, performance, and robustness of CNN models in waste classification.

Achievement: The analysis confirmed that data augmentation and synthetic data generation had a substantial positive impact on CNN model performance. Metrics such as accuracy, precision, recall, and F1-score improved significantly, highlighting the effectiveness of these techniques in enhancing the CNN models' ability to classify waste materials accurately and robustly.

The achievements of this research demonstrate the significant potential of CNN-based waste classification systems to enhance accuracy, performance, and cost efficiency in waste management. The successful application of data augmentation and synthetic data generation techniques has addressed key challenges, validating the effectiveness of CNNs in real-world scenarios.

7.2 Challenges and Future Work

Despite the overall improvements, certain classes such as “Miscellaneous Trash” and “Textile” consistently showed lower performance across all models. This suggests inherent challenges in these categories, possibly due to greater variability in appearance and composition. Due to time constraint the project concludes here. However, there is a significant potential for future research to build upon these findings.

For future work, more sophisticated augmentation techniques specifically tailored to these challenging classes should be explored. Additionally, advanced model architectures like transformer-based networks, which have demonstrated promise in handling complex image classification tasks, could be investigated. Incorporating more diverse and representative datasets would likely enhance the robustness of the models. Furthermore, leveraging transfer learning from models pre-trained on extensive datasets could provide a substantial performance boost.

8. STUDENT REFLECTIONS

Reflecting on the journey through this project, several insights were gained, and personal learnings emerged. The experience has been both challenging and rewarding, providing a deep understanding of both technical and practical aspects of applying Convolutional Neural Networks (CNNs) to waste classification.

Throughout this project, I gained a profound understanding of how Convolutional Neural Networks work and their application in image classification. This is my first project working with CNN. Initially the concept was abstract and complex, but as I delved into this project it became clearer and has given me hands-on experience working with several CNN models.

One of the critical lessons learned was the significance of data preparation in Deep learning. The project showed how essential it is to address dataset deficiencies such as class imbalances and limited image diversity to achieve accurate and robust model performance. Another piece of knowledge I was happy to have learnt was Data Augmentation and Synthetic data generation using GANs and how it helps to overcome data limitations.

Beyond technical skills, the project has fostered personal growth in several areas. I have developed stronger problem-solving abilities, learned to approach complex challenges methodically, and improved my skills in project management and communication. The iterative nature of the project, from initial planning to final implementation, has honed my ability to manage and execute a multifaceted research project.

In summary, this project has been an invaluable learning experience, offering deep insights into CNNs, data augmentation, and the practical application of machine learning technologies. The challenges faced and lessons learned have significantly contributed to my development in the field of artificial intelligence and machine learning. I look forward to applying these insights and skills to future projects and continuing to explore the transformative potential of advanced technologies in solving real-world problems.

BIBLIOGRAPHY AND REFERENCES

Adedeji, O., & Wang, Z. (2019). Intelligent Waste Classification System using Deep Learning convolutional neural network. *Procedia Manufacturing*, *35*, 607–612. https://doi.org/10.1016/j.promfg.2019.05.086

Ambika, M., & Kannammal, K. E. (2023). Deep Learning Approaches for Waste Classification: A Comprehensive Review and analysis. In IJCRT, *International Journal of Creative Research Thoughts* (Vol. 11, Issue 6) [Journal-article]. https://ijcrt.org/papers/IJCRT2306322.pdf

BBC. (2023, March 18). *How recycling can help the climate and other facts*. https://www.bbc.com/future/article/20230317-how-recycling-can-help-the-climate-and-other-facts

Blair, A. (2024, March 28). *Study finds landfill point source emissions have an outsized impact and opportunity to tackle U.S. waste methane*. Carbon Mapper. https://carbonmapper.org/study-finds-landfill-point-source-emissions-have-an-outsized-impact-and-opportunity-to-tackle-u-s-waste-methane/

Butler, J. (2023). Circular Innovation: Advancing recycling technologies for a Greener World. In University of Pittsburgh, *Advances in Recycling & Waste Management* (pp. 2–2) [Journal-article]. https://doi.org/10.37421/2475-7675.2023.8.295

Cao, L., & Xiang, W. (2020). Application of convolutional neural network based on transfer learning for garbage classification. 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). https://doi.org/10.1109/itoec49072.2020.9141699

Chatterjee, S., Hazra, D., Byun, Y., & Kim, Y. (2022). Enhancement of image classification using transfer learning and GAN-Based synthetic data augmentation. *Mathematics*, *10*(9), 1541. https://doi.org/10.3390/math10091541

Choudhary, N., Sharma, A., Rathore, V. S., & Tiwari, N. (2024). Performance comparison of ReSNet50V2 and VGG16 models for feature extraction in deep learning. In Lecture notes in networks and systems (pp. 223–229). https://doi.org/10.1007/978-981-99-8031-4_21

City of Wolverhampton Council. (2018). *Waste Strategy 2018 - 2028*. https://wolverhampton.moderngov.co.uk/documents/s75571/Appendix%201%20for%20Waste%20Strategy%202018-2028.pdf

Cujbescu, D., Nenciu, F., Persu, C., Găgeanu, I., Gabriel, G., Vlăduț, N., Matache, M., Voicea, I., Pruteanu, A., Bularda, M., Paraschiv, G., & Boruz, S. P. (2023). Evaluation of an optical sorter effectiveness in separating maize seeds intended for sowing. *Applied Sciences*, *13*(15), 8892. https://doi.org/10.3390/app13158892

Demirbas, A. (2011). Waste management, waste resource facilities and waste conversion processes. *Energy Conversion and Management*, *52*(2), 1280–1287. https://doi.org/10.1016/j.enconman.2010.09.025

Dong, Y., Xiao, H., & Dong, Y. (2022). SA-CGAN: An oversampling method based on single attribute guided conditional GAN for multi-class imbalanced learning. *Neurocomputing*, *472*, 326–337. https://doi.org/10.1016/j.neucom.2021.04.135

Duncan, A. (2024, February 6). Navigating Magnetic Separators: Key considerations for your next magnet. *TENNANT SPECS*. https://www.tennantspecs.com/post/navigating-magnetic-separators-key-considerations-for-your-next-magnet

Fan, J., Cui, L., & Fei, S. (2023). Waste Detection System based on data augmentation and YOLO_EC. *Sensors*, *23*(7), 3646. https://doi.org/10.3390/s23073646

Fang, B., Yu, J., Chen, Z., Osman, A. I., Farghali, M., Ihara, I., Hamza, E. H., Rooney, D. W., & Yap, P. (2023). Artificial intelligence for waste management in smart cities: a review. *Environmental Chemistry Letters*, *21*(4), 1959–1989. https://doi.org/10.1007/s10311-023-01604-3

Gao, X., Xiao, Z., & Deng, Z. (2024). High accuracy food image classification via vision transformer with data augmentation and feature augmentation. *Journal of Food Engineering*, *365*, 111833. https://doi.org/10.1016/j.jfoodeng.2023.111833

Gharakhanian, A. (2016). *GANs: One of the hottest topics in machine learning.* Linkedin. Retrieved July 24, 2024, from https://www.linkedin.com/pulse/gans-one-hottest-topics-machine-learning-al-gharakhanian/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. In *MIT Press eBooks*. https://dl.acm.org/citation.cfm?id=3086952

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gulzar, Y. (2023). Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. Sustainability, 15(3), 1906. https://doi.org/10.3390/su15031906

Inoue, H. (2018). Data augmentation by pairing samples for images classification. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1801.02929

Jenkins, J., & Roy, K. (2024). Exploring deep convolutional generative adversarial networks (DCGAN) in biometric systems: a survey study. *Discover Artificial Intelligence*, *4*(1). https://doi.org/10.1007/s44163-024-00138-z

Kaddoura, S. (2023). A primer on generative adversarial networks. In *SpringerBriefs in computer science*. https://doi.org/10.1007/978-3-031-32661-5

Kumsetty, N. V., Nekkare, A. B., Kamath, S. S., & Kumar, M. A. (2023). An approach for waste classification using data augmentation and transfer learning models. In *Lecture notes in electrical engineering* (pp. 357–368). https://doi.org/10.1007/978-981-99-0189-0_27

Li, C. T., Zhang, J., & Farnia, F. (2024, April 18). *On Convergence in Wasserstein Distance and f-divergence Minimization Problems*. PMLR. https://proceedings.mlr.press/v238/ting-li24a.html

Lu, Y., Chen, D., Olaniyi, E., & Huang, Y. (2022). Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review. *Computers and Electronics in Agriculture*, *200*, 107208. https://doi.org/10.1016/j.compag.2022.107208

Na, S., Heo, S., Han, S., Shin, Y., & Lee, M. (2022). Development of an artificial intelligence model to recognise construction waste by applying image data augmentation and transfer learning. *Buildings*, *12*(2), 175. https://doi.org/10.3390/buildings12020175

Ni, C. (2023). Image-to-Image Translation Based on CycleGAN: From CT to MRI. *E 1st International Conference on Data Analysis and Machine Learning (DAML 2023)*, 229–233. https://doi.org/10.5220/0012799400003885

Nnamoko, N., Barrowclough, J., & Procter, J. (2022). Solid waste image classification using deep convolutional neural network. *Infrastructures*, *7*(4), 47. https://doi.org/10.3390/infrastructures7040047

O'Gara, S., & McGuinness, K. (2019). Comparing Data augmentation Strategies for deep image Classification. *IMVIP 2019: Irish Machine Vision & Image Processing, Technological University Dublin, Dublin, Ireland*. https://doi.org/10.21427/148b-ar75

Paneru, S., & Jeelani, I. (2021). Computer vision applications in construction: Current state, opportunities & challenges. *Automation in Construction*, *132*, 103940. https://doi.org/10.1016/j.autcon.2021.103940

Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1712.04621

Qi, J., Nguyen, M., & Yan, W. Q. (2024). NUNI-Waste: novel semi-supervised semantic segmentation waste classification with non-uniform data augmentation. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-024-18265-1

Rahman, M. O., Hussain, A., & Basri, H. (2013). A critical review on waste paper sorting techniques. *International Journal of Environmental Science and Technology*, *11*(2), 551–564. https://doi.org/10.1007/s13762-013-0222-3

Rao, A. S., & Mahantesh, K. (2022). Image classification based on Inception-V3 and a mixture of handcrafted features. In Lecture notes in electrical engineering (pp. 527–537). https://doi.org/10.1007/978-981-19-2281-7_49

Romero-Arismendi, N. O., Olivares-Galvan, J. C., Hernandez-Avila, J. L., Escarela-Perez, R., Jimenez-Mondragon, V. M., & Gonzalez-Montañez, F. (2024). Past, present, and future of new applications in utilization of Eddy currents. *Technologies*, *12*(4), 50. https://doi.org/10.3390/technologies12040050

Shafique, U., & Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). In Innovative Space of Scientific Research Journals, *International Journal of Innovation and Scientific Research* (Vol. 12, Issue 1, pp. 217–222).

Shijie, J., Ping, W., Peiyi, J., & Siping, H. (2017). Research on data augmentation for image classification based on convolution neural networks. *2017 Chinese Automation Congress (CAC), Jinan, China*, 4165–4170. https://doi.org/10.1109/cac.2017.8243510

silsnetwork.com. (2019, July 9). Data Mining - SILS Technology. SILS Technology. https://www.silsnetwork.com/data-mining/

Single, S., Iranmanesh, S., & Raad, R. (2023). RealWaste: A novel Real-Life data set for landfill waste classification using deep learning. *Information*, *14*(12), 633. https://doi.org/10.3390/info14120633

Srivatsan, K., Dhiman, S., & Jain, A. (2021). Waste Classification using Transfer Learning with Convolutional Neural Networks. *IOP Conference Series. Earth and Environmental Science*, *775*(1), 012010. https://doi.org/10.1088/1755-1315/775/1/012010

Toennies, K. D. (2024). Feature Extraction by Convolutional Neural Network. In *An Introduction to Image Classification* (pp. 169–186). https://doi.org/10.1007/978-981-99-7882-3_8

Tsagarakis, N., & Maniadakis, M. (2024). On the generation and assessment of synthetic waste images. In *IEEE Conference on Artificial Intelligence (CAI)* (pp. 1019–1019). https://ieeecai.org/2024/wp-content/pdfs/540900b019/540900b019.pdf

Ulutürk, S., Kaya, M., Kaya, Y. Ç., Altintaş, O., & Turan, B. (2023). Optimization of several deep CNN models for waste classification. *Sakarya University Journal of Computer and Information Sciences*, *6*(2), 91–104. https://doi.org/10.35377/saucis. . .1257100

Wan, C., Shen, G. Q., & Choi, S. (2019). Waste Management Strategies for Sustainable development. In *Springer eBooks* (pp. 1–9). https://doi.org/10.1007/978-3-319-63951-2_194-1

Wang, Y., Li, D., Nie, C., Gong, P., Yang, J., Hu, Z., Li, B., & Ma, M. (2023). Research progress on the wear resistance of key components in agricultural machinery. *Materials*, *16*(24), 7646. https://doi.org/10.3390/ma16247646

Wolverhampton Council. (n.d.). *Waste and Recycling | What happens to my recycling that's collected?* Retrieved June 26, 2024, from https://www.wolverhampton.gov.uk/waste-and-recycling

Wu, M., Zhou, J., Peng, Y., Wang, S., & Zhang, Y. (2024). Deep Learning for image Classification: A review. *Lecture Notes in Electrical Engineering*, 352–362. https://doi.org/10.1007/978-981-97-1335-6_31

Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, *396*, 39–64. https://doi.org/10.1016/j.neucom.2020.01.085

Yazdi, M. (2024). Maintenance strategies and optimization techniques. In *Springer series in reliability engineering* (pp. 43–58). https://doi.org/10.1007/978-3-031-53514-7_3

APPENDIX A – PROGRAMMING CODE

Most of the code used in this project were adopted from Codebasics YouTube tutorial series (https://youtube.com/playlist?list=PLeo1K3hjS3uu7CxAacxVndI4bE_o3BDtO&si=tsHbeo5xSW1G3OX6) and StackOverflow. The codes were adapted to meet the specific requirement of my project. For code not sourced from Codebasics, references have been included as comments within the respective code blocks.

Below are code blocks for Sythetic Image generation with GAN, data augmentation and the model training used in this project. The complete code and dataset for this project is available in the following Google Drive folder: https://drive.google.com/drive/folders/1HRPHO_DCXf7nG3b-fAckeBEdE5MT7sZ3?usp=sharing

Lightweight GAN Generator

```
# code sourced from https://www.youtube.com/watch?v=AALBGpLbj6Q&t=603s
# Configuration
image_size = (80, 80, 3)
latent_dim = 100
batch_size = 16
epochs = 500

# Lightweight Generator
def build_lightweight_generator():
    model = Sequential(name="Lightweight_Generator")
    model.add(layers.Dense(10 * 10 * 128, use_bias=False, input_shape=(latent_dim,)))  # Adjust for new image size
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())
    model.add(layers.Reshape((10, 10, 128)))  # Adjust for new image size

    model.add(layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same', use_bias=False))
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())

    model.add(layers.Conv2DTranspose(32, (5, 5), strides=(2, 2), padding='same', use_bias=False))
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())

    model.add(layers.Conv2DTranspose(3, (5, 5), strides=(2, 2), padding='same', use_bias=False, activation='tanh'))
    return model
```

Lightweight GAN Discriminator

```
# code sourced from https://www.youtube.com/watch?v=AALBGpLbj6Q&t=603s
# Lightweight Discriminator
def build_lightweight_discriminator():
    input_img = layers.Input(shape=image_size)

    x = layers.Conv2D(32, (5, 5), strides=(2, 2), padding='same')(input_img)
    x = layers.LeakyReLU()(x)
    x = layers.Dropout(0.3)(x)

    x = layers.Conv2D(64, (5, 5), strides=(2, 2), padding='same')(x)
    x = layers.LeakyReLU()(x)
    x = layers.Dropout(0.3)(x)

    x = layers.Conv2D(128, (5, 5), strides=(2, 2), padding='same')(x)
    x = layers.LeakyReLU()(x)
    x = layers.Dropout(0.3)(x)

    validity = layers.Flatten()(x)
    validity = layers.Dense(1)(validity)

    return Model(input_img, validity, name="Lightweight_Discriminator")
```

Lightweight GAN Loss function and optimizer specification

```
# code sourced from https://www.youtube.com/watch?v=AALBGpLbj6Q&t=603s
# Loss Functions
def generator_loss(fake_output):
    return -tf.reduce_mean(fake_output)

def discriminator_loss(real_output, fake_output):
    return tf.reduce_mean(fake_output) - tf.reduce_mean(real_output)

# Optimizers
generator_optimizer = Adam(1e-3)
discriminator_optimizer = Adam(1e-3)

# Build and compile models
generator = build_lightweight_generator()
discriminator = build_lightweight_discriminator()
```

Training Loop for the Lightweight GAN

```
# code sourced from https://www.youtube.com/watch?v=AALBGpLbj6Q&t=603s
# Training Loop
@tf.function
def train_step(images):
    noise = tf.random.normal([batch_size, latent_dim])

    with tf.GradientTape() as gen_tape, tf.GradientTape() as disc_tape:
        generated_images = generator(noise, training=True)

        real_output = discriminator(images, training=True)
        fake_output = discriminator(generated_images, training=True)

        gen_loss = generator_loss(fake_output)
        disc_loss = discriminator_loss(real_output, fake_output)

    gradients_of_generator = gen_tape.gradient(gen_loss, generator.trainable_variables)
    gradients_of_discriminator = disc_tape.gradient(disc_loss, discriminator.trainable_variables)

    generator_optimizer.apply_gradients(zip(gradients_of_generator, generator.trainable_variables))
    discriminator_optimizer.apply_gradients(zip(gradients_of_discriminator, discriminator.trainable_variables))
```

Specifying directory and labels

```
data_dirs = pathlib.Path('./dataset/original')
waste_images_dictss = {
    'cardboard': list(data_dir.glob('Cardboard/*')),
    'food_organics': list(data_dir.glob('FoodOrganics/*')),
    'glass': list(data_dir.glob('Glass/*')),
    'metal': list(data_dir.glob('Metal/*')),
    'misc_Trash': list(data_dir.glob('MiscellaneousTrash/*')),
    'paper': list(data_dir.glob('Paper/*')),
    'plastic': list(data_dir.glob('Plastic/*')),
    'textile': list(data_dir.glob('TextileTrash/*')),
    'vegetation': list(data_dir.glob('Vegetation/*')),
}

waste_image_labelss = {
    'cardboard': 0,
    'food_organics': 1,
    'glass': 2,
    'metal': 3,
    'misc_Trash': 4,
    'paper': 5,
    'plastic': 6,
    'textile': 7,
    'vegetation': 8,
}
```

Resizing each image in the directory to size 80x80

```
X, y = [], []
for waste_name, images in waste_images_dictss.items():
    for image in images:
        img = cv2.imread(str(image))
        resized_img = cv2.resize(img, (80, 80))
        X.append(resized_img)
        y.append(waste_image_labelss[waste_name])

X = np.array(X)
y = np.array(y)

# Normalize images to [-1, 1]
X = (X - 127.5) / 127.5
```

Function to save Generated Images

```
# code sourced from https://www.youtube.com/watch?v=AALBGpLbj6Q&t=603s
# Create a directory to save generated images if it doesn't exist
output_dir = 'generated_images'
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

# Training Loop
def generate_and_save_images(model, epoch, test_input):
    predictions = model(test_input, training=False)

    fig = plt.figure(figsize=(4, 4))

    for i in range(predictions.shape[0]):
        plt.subplot(4, 4, i+1)
        plt.imshow((predictions[i].numpy() * 127.5 + 127.5).astype(np.uint8))  # Convert to numpy before astype
        plt.axis('off')

    image_path = os.path.join(output_dir, f'image_at_epoch_{epoch:04d}.png')
    plt.savefig(image_path)
    plt.show()
```

Generating the synthetic images

```
# code sourced from https://www.youtube.com/watch?v=AALBGpLbj6Q&t=603s
# Creating a fixed noise vector for visualization of the generator's progress
num_examples_to_generate = 16
seed = tf.random.normal([num_examples_to_generate, latent_dim])

# Training loop
for epoch in range(epochs):
    print(f'Starting epoch {epoch+1}/{epochs}')

    for batch_start in range(0, len(X), batch_size):
        image_batch = X[batch_start:batch_start + batch_size]

        # If the batch is smaller than the batch size, skip it
        if image_batch.shape[0] != batch_size:
            continue

        train_step(image_batch)

    # Generate and save images every 10 epochs
    if (epoch + 1) % 10 == 0:
        generate_and_save_images(generator, epoch + 1, seed)
        generator.save(f'generator_epoch_{epoch+1}.h5')
        discriminator.save(f'discriminator_epoch_{epoch+1}.h5')

    print(f'Epoch {epoch+1} completed')

print('Training finished.')
```

Specifying directory and creating class specific directory in preparation for Data Augmentation

```
# Create a directory to save augmented images
augmented_image_dir = './augmented_images_nnn'
os.makedirs(augmented_image_dir, exist_ok=True)
```

```
# Create class-specific directories within the augmented images directory
class_names = {v: k for k, v in waste_image_labels.items()}
for class_id, class_name in class_names.items():
    class_dir = os.path.join(augmented_image_dir, class_name)
    os.makedirs(class_dir, exist_ok=True)
```

Creating Data Augmentation function

```
#sourced from codebasics https://youtube.com/playlist?list=PLeo1K3hjS3uu7CxAacxVndI4bE_o3BDtO&si=tsHbeo5xSW1G3OX6
# Define the target size
target_size = (80, 80)

# Augmentation function
def augment_image(image):
    augmented_images = []
    h, w = image.shape[:2]

    # Random rotation
    angle = random.uniform(-20, 20)
    M = cv2.getRotationMatrix2D((w / 2, h / 2), angle, 1)
    aug_img = cv2.warpAffine(image, M, (w, h))
    aug_img = cv2.resize(aug_img, target_size)
    augmented_images.append(aug_img)

    # Random zoom
    zoom_factor = random.uniform(0.8, 1.2)
    aug_img = cv2.resize(image, None, fx=zoom_factor, fy=zoom_factor)
    h_zoom, w_zoom = aug_img.shape[:2]
    if h_zoom > target_size[1] and w_zoom > target_size[0]:
        start_x = (w_zoom - target_size[0]) // 2
        start_y = (h_zoom - target_size[1]) // 2
        aug_img = aug_img[start_y:start_y + target_size[1], start_x:start_x + target_size[0]]
    aug_img = cv2.resize(aug_img, target_size)
    augmented_images.append(aug_img)

    # Random brightness adjustment
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    value = random.uniform(0.5, 1.5)
    hsv[:, :, 2] = hsv[:, :, 2] * value
    aug_img = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
    aug_img = cv2.resize(aug_img, target_size)
    augmented_images.append(aug_img)

    # Random flip
    flip_code = random.choice([-1, 0, 1])
    aug_img = cv2.flip(image, flip_code)
    aug_img = cv2.resize(aug_img, target_size)
    augmented_images.append(aug_img)

    return augmented_images
```

Generating images based on class size

```
#sourced from codebasics https://youtube.com/playlist?list=PLeo1K3hjS3uu7CxAacxVndI4bE_o3BDtO&si=tsHbeo5xSW1G3OX6
# Count the number of images in each category
train_counts = {class_name: sum(y_train == class_id) for class_id, class_name in class_names.items()}
max_images_per_category = max(train_counts.values())

X_train_augmented = []
y_train_augmented = []

# Add the original data to augmented data
X_train_augmented.extend(X_train)
y_train_augmented.extend(y_train)

# Dictionary to keep count of images per category
augmented_count = {class_name: train_counts[class_name] for class_name in class_names.values()}

print("Augmenting data...")
for class_id, class_name in class_names.items():
    class_indices = [i for i, label in enumerate(y_train) if label == class_id]
    while augmented_count[class_name] < max_images_per_category:
        for i in class_indices:
            x = X_train[i]
            augmented_imgs = augment_image(x)
            for augmented_img in augmented_imgs:
                if augmented_count[class_name] >= max_images_per_category:
                    break
                X_train_augmented.append(augmented_img)
                y_train_augmented.append(class_id)
                augmented_count[class_name] += 1

                # Save the augmented image to the class-specific directory
                augmented_img_path = os.path.join(augmented_image_dir, class_name,
                                                  f'augmented_{class_id}_{augmented_count[class_name]}.jpg')
                success = cv2.imwrite(augmented_img_path, augmented_img)
                if not success:
                    print(f"Failed to save image at {augmented_img_path}")
```

VGG16 Model

```
# code sourced from https://towardsdatascience.com/transfer-learning-with-vgg16-and-keras-50ea161580b4
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

# Define the model using VGG16 with some trainable layers
input_shape = (80, 80, 3)
base_model = VGG16(weights='imagenet', include_top=False, input_shape=input_shape)

# Unfreeze the top layers of VGG16 for fine-tuning
for layer in base_model.layers[-10:]:
    layer.trainable = False

model = Sequential([
    base_model,
    layers.Flatten(),
    layers.Dense(256, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.1)),
    layers.Dropout(0.5),
    layers.Dense(len(waste_image_labels), activation='softmax')
])
```

ResNet50 Model

```
# code sourced from https://stackoverflow.com/questions/67701192/transfer-learning-with-resnet50-for-image-classification
from tensorflow.keras.applications import ResNet50


input_shape = (80, 80, 3)
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)

# Unfreeze the top layers of ResNet50 for fine-tuning
for layer in base_model.layers[-6:]:
    layer.trainable = True

model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(256, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    layers.Dropout(0.5),
    layers.Dense(len(waste_image_labels), activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

MobileNetV2 Model

```
#code sourced from https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV2
from tensorflow.keras.applications import MobileNetV2

# Define the model using MobileNetV2 with some trainable layers
input_shape = (80, 80, 3)
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=input_shape)

# Unfreeze the top layers of MobileNetV2 for fine-tuning
for layer in base_model.layers[-6:]:
    layer.trainable = True

model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(256, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    layers.Dropout(0.5),
    layers.Dense(len(waste_image_labels), activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

InceptionV3 Model

```
# code sourced from
# https://medium.com/@armielynobinguar/simple-implementation-of-inceptionv3-for-image-classification-using-tensorflow-and-keras-6

from tensorflow.keras.applications import InceptionV3

# Define the model using InceptionV3 with some trainable layers
input_shape = (80, 80, 3)
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=input_shape)

# Unfreeze the top layers of InceptionV3 for fine-tuning
for layer in base_model.layers[-6:]:
    layer.trainable = True

model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(256, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    layers.Dropout(0.5),
    layers.Dense(len(waste_image_labels), activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

APPENDIX B – MEETING RECORDS

Day	Topic Discussed	Date and Time
1	***Discussion of Project Topics and Ethics Application***	June 6, 2024 (1:00 PM - 2:00 PM)
	- Discussed about my project topic	
	- Reviewed ethics application process.	
	- Scheduled specific times for weekly meetings.	
2	***Refining Project Ideas and Utilizing University Resources***	June 13, 2024 (1:00 PM - 2:00 PM)
	- Worked on narrowing the project idea for feasibility.	
	- Guided on using the university library to search for resources.	
	- Discussed how to narrow search results to about 1,000 outputs.	
	- Advised to aim for a small incremental change in the project.	
	- Encouraged rethinking the project outcome from different perspectives.	
	Submission of 500 word outlining the problems, methodology and ideal outcome of the proposed project.	
3	***Feedback and Further Research Guidance***	June 21, 2024 (1:00 PM - 2:00 PM)
	- Received feedback on the 500-word outline submitted.	
	- Advised on researching deployment locations and scales.	
	- Discussed existing waste management methods and measurement techniques.	
	- Discussed about current challenges in waste classification.	
	- Suggested ways to source relevant data.	
	- Provided tips for writing a comprehensive literature review.	
4	***One-on-One Support Session***	June 28, 2024 (1:30 PM - 2:00 PM)
	- Discussed challenges faced in the literature review.	
	- Provided specific advice on overcoming these challenges.	
5	***Progress Review and Accuracy Discussion***	July 12, 2024 (2:00 PM - 2:30 PM)
	- Reviewed project progress and addressed any challenges.	
	- Submitted draft literature review for feedback.	
	- Discussed accuracy of project results compared to Wolverhampton Council's separation accuracy.	
	- Advised to consider cost implications and efficiency as part of the system evaluation.	
6	***Report Structure and Literature Review Feedback***	July 26, 2024 (2:00 PM - 2:30 PM)
	- Discussed the structure of the final report.	
	- Outlined compulsory and optional sections of the report.	
	- Provided feedback on the submitted literature review.	

APPENDIX C – CERTIFICATE OF ETHICAL APPROVAL

Coventry University

Certificate of Ethical Approval

Applicant:	Blessing Oriaguaman
Project Title:	Improving Convolutional Neural Network - Based Waste Classification Through Data Augmentation and Synthetic Data Generation.

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:	27 Jun 2024
Project Reference Number:	P177890