2025

SMART WASTE SORTING SYSTEM

Sinethemba Nocuze

IFS 315 EXERCISE 3

5/17/2025

TABLE OF CONTENTS

Contents

TABLE OF CONTENTS 1
Abstract 2
METHODOLOGY 3
1.1 Data Collection 3
1.2 Data Pre-processing 3
1.3 Data Splitting 4
1.4 Model Development 4
1.5 Evaluation 5
1.6 Generalization Testing with Webcam Input 5
1.7 Performance Visualization 5
2. APPLICATIONS 6
3. LIMITATIONS AND ETHICAL CONSIDERATIONS 6
3.1 Limitations 6
3.2 Ethical Considerations 7
4. RESPONSIBLE USE STRATEGY 7
CONCLUSION 8
APPENDICES 9
REFERENCES 10

Abstract

This report presents the design, implementation, and evaluation of an intelligent vision-based classification system for five types of recyclable waste: plastic, paper, metal, glass, and cardboard. The core of the system is a convolutional neural network built using transfer learning from MobileNetV2, a lightweight yet powerful model optimized for image classification. The dataset was downloaded from public sources, including the **TrashNet dataset**, and was improved with user-generated images to capture diverse real-world conditions. Pre-processing steps, including resizing, normalization, and augmentation, were applied to ensure model robustness and mitigate overfitting.

The data was split using a stratified sampling strategy to preserve class balance across training, validation, and test sets. The model was evaluated using accuracy, precision, recall, and confusion matrix analysis. Additionally, real-time testing was conducted using webcam input to simulate real-world deployment and assess generalizability.

The report includes a detailed methodology, performance metrics, limitations, ethical considerations, and recommendations for responsible use. This work contributes to the broader goal of sustainable waste management through automation and intelligent decision-making systems.

METHODOLOGY

1.1 Data Collection

The dataset used in this project includes five types of recyclable **waste: plastic, paper, metal**, **glass, and cardboard.** The main source of this dataset is the open TrashNet dataset, which we found on the Kaggle platform. Kaggle is a popular site for sharing datasets and working on machine learning projects. TrashNet contains labelled images of different kinds of common recyclable items.

After downloading and checking the images, we noticed that some of them were damaged during the resizing process. These images had broken pixels or were visually unclear, so they couldn't be used for training. To keep the data clean and accurate, we manually deleted all the damaged images before splitting the data for training, validation, and testing.

Following the cleaning process, the remaining dataset included **2,354 images distributed across five classes**, with each class containing more than 50 images. The images were then randomly split into three subsets:

- **Training Set**: 70%
- **Validation Set**: 20%
- **Test Set**: 10%

The dataset was split into separate folders using **Python's os, shutil, and random modules**. A combination of for loops and list comprehensions was used to ensure simplicity, efficiency, and maintainability.

The class labels were derived from a predefined **class_names.txt** file, and care was taken to preserve class balance during the split. This stratified sampling ensured balanced representation of all classes across the subsets, supporting consistent model training and unbiased evaluation.

1.2 Data Pre-processing

Before training the model, all images were resized to 256x56 pixels to match the input size expected by MobileNetV2. This helped make sure that all images had the same shape, making training easier and faster. After resizing, pixel values were normalized by scaling them between 0 and 1. This step made the training process more stable and allowed the model to learn faster. These changes were important to prepare the data for deep learning. Without them, the model might struggle to learn properly.

To make the model more accurate and avoid overfitting, data augmentation was used on the training set. Augmentation means adding small changes to images like rotating them, flipping them, zooming in, or shifting their size. These changes helped the model learn to recognize images even if they look a bit different. TensorFlow's **ImageDataGenerator() Class** from the **Keras API** was used to apply these changes automatically. This made the training data more diverse without collecting more images. As a result, the model became better at handling real-world images.

1.3 Data Splitting

After the data was prepared, it was divided into three parts: training, validation, and testing. The training set had 70% of the data, the validation set had 20%, and the test set had 10%.

Stratified sampling was used to split the data, which means each part kept the same mix of classes. This helped the model learn from a balanced dataset. The training set was used to teach the model, while the validation set checked how well it was learning. If the model did poorly on the validation set, changes were made to improve it.

The test set was kept separate and only used after the model was trained. It helped measure how well the model could predict on new, unseen data. Since the test set was never shown to the model during training, it gave a fair view of the model's performance. Keeping the test set untouched is important to avoid bias. This splitting method made sure the model's results were reliable. It also helped avoid overestimating the model's true accuracy.

1.4 Model Development

For the development of the waste classification model, I used the Keras API from Tensorflow along with the MobileNetV2 architecture. I chose MobileNetV2 because it provides a good balance between accuracy and processing speed. This model is designed to perform well even on devices that have limited computing power, which made it a perfect fit for my project. Since I do not have access to high-end or powerful hardware, it was important to use a lightweight and efficient model that could still deliver strong performance. MobileNetV2 allowed me to train the model effectively, without requiring long processing times or expensive equipment, and still achieved accurate results. This made the entire development process smoother and more practical for my setup.

Transfer Learning Implementation:

I used transfer learning by importing a version of MobileNetV2 that had been pre-trained on the ImageNet dataset. Then removed its top layer and added my own custom layers to make it suitable for the five-classes scenario.

The custom layers included a **GlobalAveragePooling2D layer**, followed by a dense layer with 128 units and ReLU activation. I then added a dropout layer with a rate of 0.5 to prevent overfitting. Finally, a **softmax output layer** was used to classify the objects into one of the five categories.

Training:

I saved the model in **HDF5 (h5) format** because it keeps everything in one file, the model structure, weights, and training settings — which makes it easy to load later without starting over.

Since I was working on a **multi-class classification** problem, I used **categorical crossentropy** as the loss function. I chose the **Adam optimizer** because it works fast and adjusts learning rates automatically, helping the model learn more efficiently. I trained the model with a **batch size of 32** and set the **number of epochs between 20 and 30**.

To improve performance and avoid **overfitting**, I used **EarlyStopping** to stop training when the model stopped getting better, and **ModelCheckpoint** to save the best version of the model during training.

1.5 Evaluation

Once the model finished training, I tested how well it performed using the test dataset, which contained new images the model hadn't seen before. The first metric I looked at was accuracy, which tells us how many images the model got right overall. But accuracy alone doesn't give the full picture.

So, I also calculated precision and recall for each of the five categories (plastic, paper, metal, cardboard, and glass). This helped me understand which materials the model identified well and where it made mistakes. For example, the confusion matrix as shown in **APPENDIX A** showed that the model sometimes mixed-up paper and cardboard, which look similar in some cases. Overall, though, the results showed that the model was able to classify each class fairly well and consistently.

1.6 Generalization Testing with Webcam Input

To test how the model would work outside of training and testing environments, I connected it to a **webcam** using the OpenCV library and gave it real-time input. This was an important step to see if the model could work in real-world scenarios, like someone holding up an object to sort it. The model did a good job when the lighting was clear, and the object was fully visible in front of the camera.

However, when I tested it in poor lighting, with background distractions, or when part of the object was hidden, the model's accuracy dropped. These results showed that while the model is promising, it still needs improvements to be reliable in every environment. Still, this webcam test proved that the model can be used in real-time systems with just a few adjustments.

1.7 Performance Visualization

To track how the model improved during training, I created **graphs** showing both accuracy and loss for each training epoch. In **APPENDIX B**, the **accuracy curve** showed how often the model predicted correctly, while the **loss curve** showed how far the predictions were from the correct labels.

By comparing these curves for the training and validation sets, I could see if the model was learning properly or just memorizing the training data (which is known as overfitting). In this case, the curves showed steady improvement and no major signs of overfitting.

I also used **early stopping**, which stopped the training automatically when the model was no longer improving. These visual tools helped us confirm that our training process was successful and that the model was well-optimized.

2. APPLICATIONS

The waste classification system I developed has several real-world applications that can help improve waste management and support sustainability efforts.

One of the key applications is the use of **smart bins**. These bins are automated and can sort waste at the source, meaning that as people throw away their trash, the system can immediately identify and separate recyclable materials from non-recyclable ones. This makes it easier for communities to manage waste and ensures that recyclable materials are properly processed without contamination.

Another important application is in **recycling plants**, where the system can be used to support conveyor-based sorting. By integrating my model with sorting systems at recycling plants, the model can automatically identify and separate different types of materials, such as plastics, papers, or metals, speeding up the recycling process and improving efficiency.

The system can also be used in **mobile assistants** to help educate people on what can and cannot be recycled. By using the camera on a mobile device, users can take pictures of items, and the system can instantly provide information on whether the item is recyclable, helping to raise awareness and encourage better recycling habits.

Lastly, the model can be used in **awareness campaigns** to promote sustainability. Interactive tools powered by my waste classification system can engage people in learning about recycling and waste management. These tools can help increase public understanding of environmental issues and encourage more responsible waste disposal practices.

3. LIMITATIONS AND ETHICAL CONSIDERATIONS

3.1 Limitations

While the waste classification system shows great potential, there are a few limitations that should be considered. One of the main challenges is **visual overlap**. The model may struggle to correctly classify items that look very similar, like paper and cardboard. These materials have very similar textures and appearance, which can cause the model to mix them up, leading to misclassification.

Another limitation is the system's **dependence on lighting**. The model performs best under ideal conditions, but its accuracy drops when the lighting is poor or if parts of the object are hidden or obscured. This means that in real-world settings, where lighting is not always perfect, the system may not always perform well. Furthermore, the system's **scope** is also limited. It only focuses on certain types of waste, specifically common recyclables like paper, plastic, and metal. It does not classify more complex or hazardous waste, such as organic materials, composite products, or toxic waste, which require different handling and sorting methods.

3.2 Ethical Considerations

There are also important **ethical considerations** to keep in mind when using the waste classification system. One of the potential issues is **bias** in the training data. If the data used to train the model primarily comes from regions with clean, well-organized waste management systems, such as in Western countries, the model might not perform as well in areas with different waste characteristics. This can lead to inaccurate classifications for users in underrepresented regions.

Privacy is another concern, especially when the model is used with webcams. There is a possibility that while capturing images of waste items, the webcam might also unintentionally record identifiable personal information, such as people's faces or private locations, raising privacy issues. Lastly, **accessibility** is another ethical consideration. In less developed regions, where resources like high-quality cameras or stable internet connections may be limited, the model might not work as effectively, thus leaving certain communities at a disadvantage.

4. RESPONSIBLE USE STRATEGY

To promote ethical and responsible use:

First, **anonymization tools** can be used to protect user privacy. By applying face and object blurring in camera feeds, the system can prevent the accidental capture of personal or identifying information. This ensures that even though the system uses webcams for real-time classification, it will not infringe on anyone's privacy by recording unnecessary details.

Transparency is another key aspect of responsible deployment. Users should always be aware when cameras are in use. This could be done by displaying clear notifications or visual indicators when the system is recording or processing data. This way, users know when their data is being captured, promoting trust in the system. Additionally, **human oversight** is important. In cases where the model is unsure of an item's classification, it should allow a manual override. This ensures that a human can intervene to correct any low-confidence predictions, preventing incorrect classifications from going unchecked.

Another important strategy is **feedback integration**. The system should use incorrect or uncertain classifications to improve future training. By continuously learning from its mistakes, the model will become more accurate and reliable over time. Lastly, **inclusive data collection** should be a priority. The dataset used to train the model needs to be diverse and reflect global waste patterns. This includes different types of waste found in various regions and communities. By expanding the dataset to represent a wider variety of waste materials and environments, the system will perform better in a broader range of real-world contexts.

CONCLUSION

This project successfully built a computer vision system that can automatically classify recyclable waste using deep learning. I used the MobileNetV2 model, which is known for being both accurate and efficient. By applying transfer learning with pre-trained weights, we trained the model to recognize five different categories of recyclable materials. The model performed well on the test data and even showed promising results during live testing with a webcam, which means it can be used in real-time applications like smart bins or recycling stations.

Throughout the project, each phase was completed with care and attention, from collecting and preparing the images, to training the model, evaluating its performance, and testing it in real-world conditions. I didn't just focus on technical performance but also considered the ethical side.

For example, I made sure to think about user privacy when using a webcam, and I acknowledged that the model might not work equally well in all environments or communities. I also noticed some limitations, like the model's difficulty with objects that look very similar or when the lighting is not good. These challenges are common in computer vision and give me useful direction for future improvements.

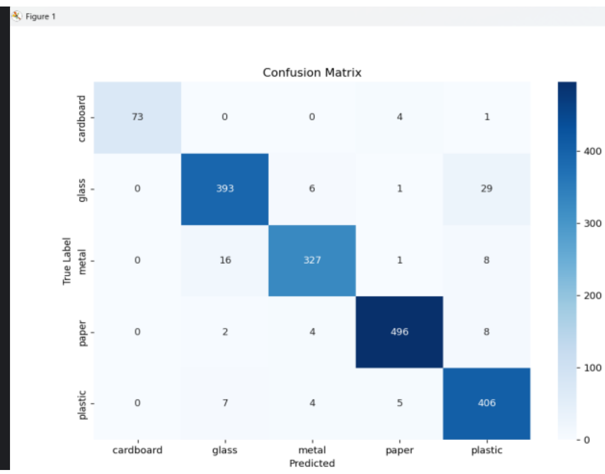
Despite those challenges, the system overall is a strong foundation for future development. With more diverse training data, especially from different regions of the world, and more feedback from real-world users, the model can be made even more accurate and inclusive.

I believe that this kind of technology can make recycling easier and smarter for everyday people. It can also be a helpful tool in larger waste management systems, leading to better sorting and reduced pollution. In the long term, this project shows how artificial intelligence can support global efforts to protect the planet and build a more sustainable future.

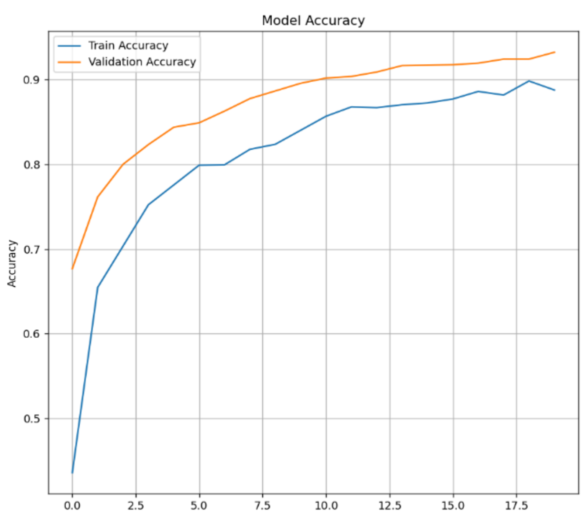
APPENDICES

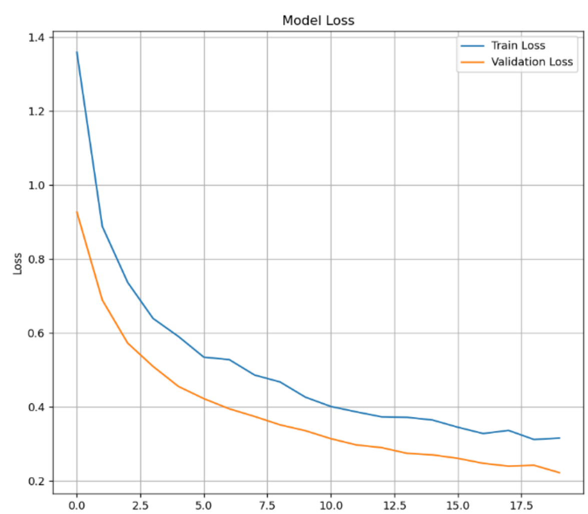
APPENDIX A

```
56/56 ━━━━━━━━━━━━━━━━━━━━ 47s 803ms/step

=== Classification Report ===
              precision    recall  f1-score   support

   cardboard       1.00      0.94      0.97        78
       glass       0.94      0.92      0.93       429
       metal       0.96      0.93      0.94       352
       paper       0.98      0.97      0.98       510
     plastic       0.90      0.96      0.93       422

    accuracy                           0.95      1791
   macro avg       0.96      0.94      0.95      1791
weighted avg       0.95      0.95      0.95      1791
```



APPENDIX B





REFERENCES

Awofeso, Z. (2024, October 11). *A summary of the "MobileNetV2: Inverted Residuals and Linear Bottlenecks" paper*. Medium. Available at: https://medium.com/codex/a-summary-of-the-mobilenetv2-inverted-residuals-and-linear-bottlenecks-paper-e19b187cb78a [Accessed 14 May 2025].

cchangcs. (n.d.). *Garbage classification dataset*. Kaggle. Available at: https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification [Accessed 12 May 2025].

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint. Available at: https://arxiv.org/abs/1704.04861 [Accessed 12 May 2025].

Keras. (n.d.). *ImageDataGenerator*. Keras. Available at: https://keras.io/api/preprocessing/image/ [Accessed 14 May 2025].

Renotte, N. (2022). *Build a Deep CNN Image Classifier with any Images* [YouTube video]. Available at: https://youtu.be/jztwpsIzEGc?si=i976BnG3IXak9-i5 [Accessed 12 May 2025].

Scikit-learn. (n.d.). *sklearn.metrics — Scikit-learn metrics module*. Available at: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics [Accessed 14 May 2025].

TensorFlow. (n.d.). *TensorFlow documentation*. Available at: https://www.tensorflow.org [Accessed 14 May 2025].

Thung, G. (n.d.). *TrashNet dataset*. GitHub. Available at: https://github.com/garythung/trashnet [Accessed 12 May 2025].

University of the Western Cape (UWC). (2025). *IFS 315 – Introduction to Machine Learning* [Course Material]. Author: Mr. Grant Hearn.

University of the Western Cape (UWC). (2025). *IFS 315 – Introduction to Machine Learning* [Course Material]. Author: Mr. Ruchen Wyngaard.