FINAL DISSERTATION

ScrapStrat: Smart Waste Management with Deep Learning

Course:	MSc in Data Analytics
Course ID:	DK_ICDAN_9
Year:	2024-25
Course Director:	Dr Abhishek Kaushik

Submitted By:	Himansh Arora
Student ID:	D00233455
Supervised By:	Dr Peadar Grant and Dr Siobhan Connolly Kernan

1. BACKGROUND

1.1 Abstract

This project aims to explore the use of Deep Learning for performing Smart Waste Classification to aid sustainable waste management. A dataset with the least ethical considerations and one that was most appropriate for the given problem definition was chosen. The dataset consists of 2,348 labeled images across 5 categories which are: Glass, Plastic, Metal, Bin, Other. This dataset was preprocessed, analyzed and augmented to address the issues of class imbalance and image diversity. The initial models include a basic Convolutional Neural Network (CNN) model and an advanced CNN model with extra features such as regularization and dropout. The evaluation of the impact of minority classes was performed which led to revealing that removing the minority 'Other' class and certain other preprocessing steps improved model performance.

Stratified K-Fold Cross-Validation and Hyperparameter Tuning was applied to make classification more efficient, yielding the best model with batch size 32, dropout 0.3, and 20 epochs. The methods of Transfer Learning and Fine-Tuning were then applied using models such as MobileNetV2, InceptionV3, EfficientNetB0, and ResNet50. Based on overall performance metrics, three best models were identified: the tuned CNN from scratch, MobileNetV2, and InceptionV3. These models were then tested on single images to examine prediction confidence. MobileNetV2 achieved the highest mAP score (0.76) with high precision and recall across categories, InceptionV3 performed perfect classification confidence for individual images with a 0.72 mAP score, and the CNN model from scratch performed comparably well but with a slightly lower mAP score (0.68).

The study highlights the importance of preprocessing, class weighting, cross-validation, and model selection to improve the accuracy of classification. Future work could explore hyperparameter optimization, testing additional models, real-time deployment of the model via a user-friendly interface, and categorizing waste as recyclable, non-recyclable, or compostable. These efforts aim towards accomplishing practical, ethical, and scalable AI solutions for sustainable waste management.

1.2 Introduction to the Problem

As per the waste management statistics in Ireland, the recycling rate of the country stands at 41% of all the municipal waste generated. Officials hope to increase this number to 55% by 2025, which is the European Union target. About 16% of the municipal waste goes to landfills and causes pollution. It is also noted that over 1,800,000 tonnes of household waste are improperly disposed of each year. This issue arises due to the misclassification of waste products by people in almost every household, due to a lack of waste management knowledge or negligence towards environmental sustainability. [1]

Often, a single non-recyclable waste material is put into the recycling bin, which renders the entire recyclable batch useless. Simpler materials like glass bottles, plastic bags, batteries, bubble wrap, and diapers, to mixed material wastes like used pizza boxes, wet paper towels, laminated paper, and so on, are sorted into recycling bins. Even a single contaminant leads to the entire batch being taken to landfills. This happens because there is a lack of proper education surrounding waste disposal, a surge in consumption and urbanisation, and an increase in immigration. Most immigrants follow a different waste disposal system in their countries and have no source for gaining knowledge about waste disposal in Ireland. Many locals find themselves confused while handling mixed-material waste. In a 2020 Repak survey, it was discovered that almost 60% of Irish citizens had difficulty sorting the waste into the proper bins. Many others simply give in to their negligent attitude towards recycling. [2]

This comes at a great cost to Ireland, as valuable recyclable materials are lost, causing a strain on the environment and exploiting resources. What increases the tension is the emission of greenhouse gases from landfill burning and landfill overflow. The economic burden of managing landfills is also not lost on the overall development of the country.

1.3 Problem Definition and Research Questions

The main research problem here is the incorrect household waste classification that leads to improper waste disposal. This research aims to achieve efficient and effective methods of creating an accessible waste classification technology for the public through the use of artificial intelligence. There is a need to create a smart, informed, and potentially automated waste classification technology using deep learning models and deploy the technology for efficient public use.

The research questions addressed in this report are as follows:

1. *How can we build deep learning models from scratch to classify waste materials into the correct categories?*

2. *How does performing hyperparameter tuning and stratified k-fold cross-validation on the deep learning models affect classification accuracy?*

3. *How does the model built for this project compare against transfer learning and fine-tuning using state-of-the-art models like ResNet50, MobileNetV2, InceptionV3, and EfficientNetB0 for waste classification by evaluating the accuracies, precision, recall, and mean average precision (mAP) scores?*

1.4 Project Relevance in the Field of Data Science

The project is a combination of novel machine learning technologies like computer vision, image preprocessing, and data analytics with environmental sustainability. The goals of the project align with the purpose of data analytics in analyzing raw data and extracting crucial information for creating technologies that address real-world challenges. The project details data collection, data preprocessing techniques, ethical considerations, building deep learning models from scratch, hyperparameter tuning, transfer learning techniques, state-of-the-art models like ResNet50, and MobileNet, exploring multiple techniques for model performance analyses, and deploying a user-friendly application to deliver the technology efficiently to the public. The code and clear documentation offer clarity into the training, validation, and testing of the models used in this project.

The technologies discussed and implemented offer a clear understanding of handling complex image data and focus on drawing a comparison between existing models and this project. This offers researchers a platform to study the drawbacks of existing models and drives their attention towards more efficient and effective methods of addressing the economic and environmental challenges of proper waste management. The project also facilitates data-driven decision-making and paves the way for automated artificial intelligence in waste disposal.

1.5 Core Technology, Architecture, and Research Processes Used

1. **Convolutional Neural Network:** For this project, the main technology used is computer vision. Computer vision heavily relies on convolutional neural networks for faster, efficient, and accurate image classification through its pattern recognition ability. It is a complex architecture that involves multiple layers, including convolutional layers, pooling, dropout, regularization, and fully-connected layers. The deep learning networks used in this project are CNN networks built from scratch, resnet50, and MobileNet. [3]

2. **Tensorflow:** It is a Python library that is used for the development and training of neural networks. [4]

3. **Python Image Library (PIL):** This is a Python library that allows users to perform image editing. This is important while loading and preprocessing image data. It is particularly useful in performing data augmentation to ensure variability and equal distribution in the dataset. [5]

4. **Data Pipeline:** Using a data pipeline for automating the preprocessing and training procedures. [6]

5. **Hyperparameter Tuning and Cross-Validation:** The project explores the effects of hyperparameter tuning and stratified k-fold cross-validation on the performance of the model in waste classification.

6. **Transfer Learning:** This research also performs transfer learning using models like MobileNetV2, EfficientNetB0, InceptionV3, and ResNet50 to make the study comprehensive and comparative.

7. **Fine-tuning:** While performing transfer learning, it is also identified that fine-tuning the models helps improve performance. This involves unfreezing the top layers and training them for the given dataset.

8. **Testing:** An important part of the project is testing the working of the best models on a given image for classification.

9. **Addressing Ethical Concerns:** The research process of taking ethical challenges into consideration is also at the core of this project, where all images have been acquired ethically, and the design of the model follows unbiased methods in classifying waste into correct categories.

10. **Literature Review:** A thorough literature review was conducted in order to explore the utilization of convolutional neural networks in image classification, following proper data preprocessing measures, and drawing accurate comparisons.

1.6 Hypothesis for a Solution

The proposed hypothesis for a solution is that a deep learning model can achieve more than 70% mean average precision score (mAP), 70% recall score, and 90% precision score in classifying waste products into correct categories through training, cross-validation, and hyperparameter tuning, and can provide an effective and efficient solution to the problem of improper waste disposal.

1.7 Structure of the Report

The report starts with a brief introduction to the topic, highlighting the statistics and the cause of the problem. It then delves into discussing problem definition and research questions addressed in this work. It also throws light on the relevance of this project in the field of Data Science. The core technologies and architecture of the report have been mentioned, followed by the hypothesis. It discusses five recent research papers in the field of using Deep Learning models for Image Classification in Proper Waste Disposal, discussing different approaches and methods, along with their limitations. The report then gets into details about the ethical considerations and approval methods undertaken, followed by data collection, data description, data preprocessing, and exploration methods. We draw a preliminary analysis based on the exploration and cleaning conducted, along with appropriate images obtained by running the code. Next is a flowchart of Artefact Development that shows all the procedures, from data cleaning to model fitting and testing. This section shows the design of the models built and the results obtained from them. The report goes on to show the fitting of a basic and an advanced deep learning model, along with a comparison and potential room for improvement. It also explores the impact of minority classes on model performance and overcomes these issues with hyperparameter tuning and stratified k-fold cross-validation for finding the best model. The next section covers transfer learning and fine-tuning methods used on MobileNetV2, InceptionV3, EfficientNetB0, and ResNet50. It also shows the testing of the best models. Further, we discuss the summary and conclusion, challenges faced, and conclude with future steps and proposed analysis.

2. LITERATURE REVIEW

Some of the research papers that align closely with the research questions of this project are listed below:

2.1 Towards Artificially Intelligent Recycling: Improving Image Processing for Waste Classification (2021) [7]

This 2021 research, conducted on waste classification using neural networks, studies the enhancement of IBM's Wastenet project. It compares the performance of transfer learning techniques prior to and post data augmentation. This study was able to showcase high test accuracies in both conditions. The baseline model shows a 91% test accuracy, and a 95% test accuracy post-augmentation. The dataset was acquired from TrashNet with 2557 images across 6 classes, and the models used for transfer learning included GoogleNet, VGG, and AlexNet. It also implements hyperparameter tuning by tuning the following parameters: learning rate schedulers, layer freezing, batch sizes, and loss functions. It compares the results of the models trained using a 10-fold cross-validation technique. It stands out by providing details of the implementation and results from this project.

Limitations: The study does not discuss the possibility of a user-friendly deployment system to make the technology readily available. This is crucial to ensure that the model created serves its purpose of environmental sustainability. Another drawback comes from the fact that the project uses only a limited variety of waste products and hence cannot be relied upon to classify complicated waste materials. It also induces bias in the study towards a particular type of waste, and leads to ethical challenges of preventing bias in the project. The background and lighting conditions are also very specific to a particular pattern and can pose a challenge in recognizing images with varied backgrounds. This is crucial to include real-world settings.

2.2 ConvoWaste: An Automatic Waste Segregation Machine Using Deep Learning (2023) [8]

This research, conducted in 2023, stands out for its automated waste segregation functionality through the use of a servo motor-based system. It also utilizes a GSM-based communication technology to notify users when the trash is full and ready to be emptied. This is done with the use of sensors. The dataset used consists of 12000 images across 6 classes, captured under different lighting and background conditions. This project employs Deep Convolutional Neural Networks for the classification of waste items into the right categories. The main aim of this project is to enhance the circular economy criteria. This study was able to achieve 98.30% accuracy in properly classifying waste in urban areas. It compares the research work with other pre-existing works and their ability to classify deep waste, intelligent waste, and recyclable waste, all of which have lower accuracies compared to the proposed DCNN model. The ConvoWaste model combines the Inception-resnet50 V2 model with several extra layers for feature extraction and classification. This is done by retaining the transfer learning parameters from preceding layers of Inception resnet50 V2, discarding the last layer, and adding a few extra layers. The user-friendly interface for communication and the automated waste segregation functionalities make this project efficient and accurate.

Limitations: Pre-trained models like the Inception-resnet50 V2 used in this study can be exposed to risks of overfitting due to fine-tuning on curated datasets. This project also failed to discuss potential user-friendly deployment strategies.

2.3 Deep Learning Approach to Recyclable Products Classification: Towards Sustainable Waste Management (2023) [9]

This study uses 5000 images of garbage and waste items, divided into five categories: paper, plastic, glass, metal, and others. It was created in 2021 and is available for free on Kaggle. The researchers delve deep into multiple deep learning models to draw comparisons and achieve the most accurate results for waste classification. It is a comprehensive study that highlights the properties of the dataset in detail and explains every model used in depth. The models used include a basic CNN model, DenseNet169, MobileNetV2, and resnet5050V2. Techniques like hyperparameter tuning and Randomized Search CV have been employed to improve model accuracy. The accuracy charts for all the models trained showed that resnet5050V2 achieved the highest accuracy of 98.95%.

Limitations: The project lacked real-time testing of the model to evaluate it for different kinds of backgrounds and materials. It also suffered from the bias factor due to a lack of categories in the dataset.

2.4 Intelligent Waste Classification Approach Based on Improved Multi-Layered Convolutional Neural Network (2024) [10]

This study uses a large dataset of 25,077 images to classify organic and recyclable waste. It trained a Deep Convolutional Neural Network (DCNN) with LeakyRelu activation unit and dropout for regularization. It also uses hyperparameter tuning of the following hyperparameters: learning rate, epochs, testing size, and Network layers. To study the performance of the model, accuracy, precision, recall, MDR, and FDR were studied. It compared the performance of the DCNN model with other state-of-the-art deep learning models like VGG16, VGG19, MobileNetV2, DenseNet121, and EfficientNetB0 after transfer learning. This study provides a detailed analysis of the model proposed and the pre-existing models used for transfer learning. It highlights the effects of hyperparameter tuning on model performance, like changing the number of epochs and altering the network design by introducing dropout layers. The proposed model was able to achieve an accuracy of 93.28% and performed well with MobileNet with an accuracy of 92%.

Limitations: This study again lacks a variety of waste categories and therefore fails to generalize to multiple and more complex waste items. The additional difficulty arises from the scalability concern regarding the hardware used in this project, which can be more difficult to deploy to a wider scale market at an efficient cost.

2.5 Real-Time Household Waste Detection and Classification for Sustainable Recycling: A Deep Learning Approach (2024) [11]

This work uses a custom waste image dataset of 3775 images containing 17 classes divided into 3 categories: recyclable, non-recyclable, and hazardous. It explains the data cleaning steps implemented using the Roboflow tool with emphasis on data augmentation, data annotation, and data preprocessing. The augmentation techniques applied were rotating, flipping, blurring, random cropping, and darkening. It explains the architecture of the YOLOV8 model used for training in detail, along with the placement and integration of SE and CBAM blocks. A real-time camera system was used to check the efficiency of the model. The performance measures utilized were precision, recall, and F1 score. The baseline model gave an accuracy of 85.3%, which was enhanced by data augmentation to 87.4%. The integration of SE and CBAM blocks improves model performance up to 87.1% and 88.2%, respectively.

Limitations: YOLO models depend on proper image annotation, which can be labor-intensive. It is also possible that these images are poorly annotated if labor is short, thereby compromising model performance. The dependence of real-time YOLO deployment on a GPU is also costly.

3. EXPLORATION OF DATA AND METHODS

3.1 Procedure of Ethical Approval Undertaken

During the building of this report, proper ethical guidelines were taken into consideration. An ethical thesis for the project was submitted to the respective supervisors of this module. The thesis detailed the topic of utilizing deep learning-based image recognition for sustainable waste management. It explored the potential ethical strengths and harms related to the data and the topic concerned. It went into detail with real-world examples about privacy and security, bias, transparency and fairness, and computational costs associated with the dataset. It also highlighted the ethical benefits of improving sustainability through AI, aiding recycling, reducing penalties, and deploying large-scale models for industrial use in this project. The numerous data challenges faced along the way that pose ethical regulations were discussed in detail with relevant examples and potential solutions to overcome these challenges. It highlighted the importance of laying down proper guidelines for users, gaining explicit consent, introducing flags and checks in the model to avoid bad actors and personal data, utilizing proper human annotations to ensure reliable accuracy, and making the model more robust and inclusive. It also performed the SWOT analysis of the project in terms of AI and ACL guidelines. The thesis was crucial in making the deployment and utilization of the machine learning model ethical and accurate for public use while adhering to proper ethical guidelines.

Furthermore, an ethical considerations form was filled and submitted to the supervisors, highlighting the core ethical considerations of the project for institutional and general approval.

3.2 Data Collection

For collecting the dataset, multiple online public datasets were explored. Finally, a dataset with the least ethical considerations and one that was most appropriate for the given problem definition was chosen. A zip file of pre-split train, test, and validation folders was downloaded with appropriate labels and images. The link for the dataset is given below:

Trash Sorting Dataset

3.3 Data Description

The dataset contains 2348 images of trash items belonging to 5 classes: Glass, Plastic, Metal, Bin, Other. The dataset has been divided into 3 folders: train, test, and valid, for training, validation, and testing of the model. Each folder contains .jpg images along with a CSV file of annotations corresponding to each image. The distribution of images to each folder is as follows: Train - 1482, Valid - 505, Test - 361. All the images are 640 x 640 pixels.

3.4 Evaluation Metrics

The following reference metrics were selected for studying and comparing the results. The dataset was taken from the platform mentioned below, which highlights evaluation metrics in its model section.

Reference Metrics

3.5 Data Cleaning, Exploration, and Visualizations

We printed a few pictures to visualize what the image dataset looks like. A thorough examination of class distribution was also conducted to study class imbalance in the dataset. The image sizes for all images in the train, test, and validation datasets were determined. The annotations present in the CSV file were extracted. [12]

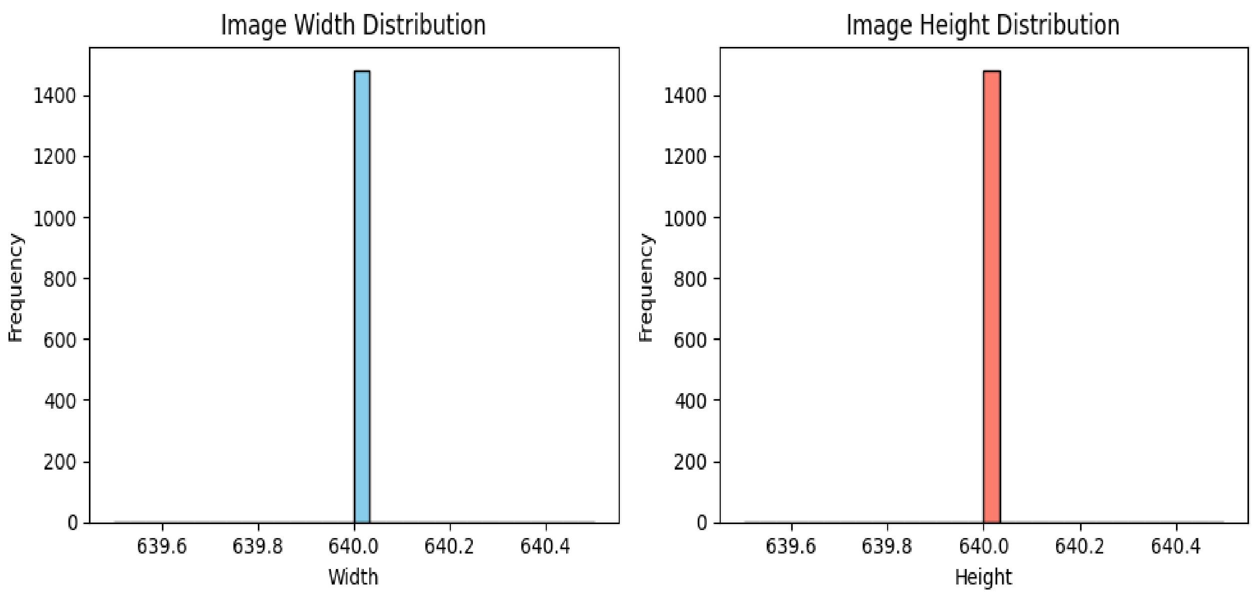
For data exploration and cleaning, the following steps were performed:

1. **Load the zip file:** The zip file containing train, test, and valid folders was uploaded to Google Colaboratory. The folder was unzipped, and all the images were acquired for analysis.

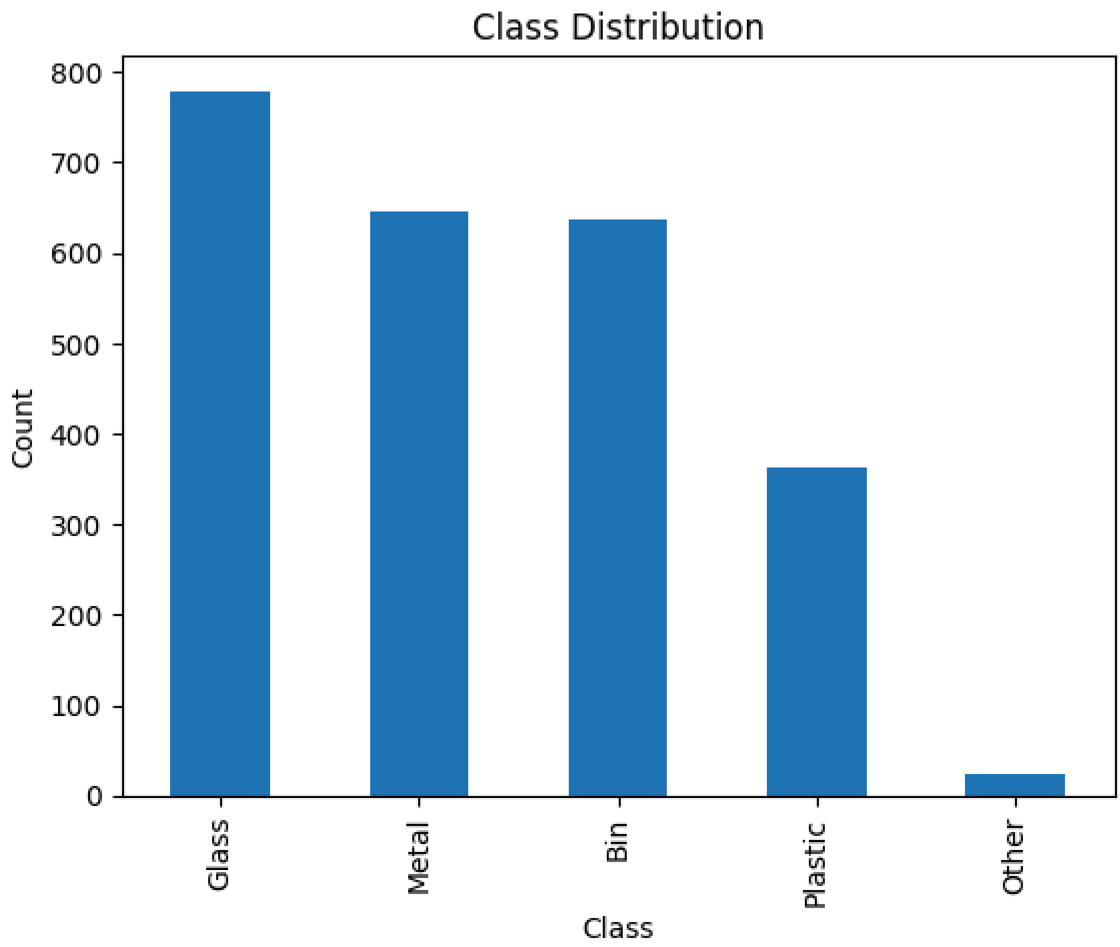
2. **Extract annotation:** The annotations CSV file was extracted from each folder.

3. **Display the dataset:** The first few images were displayed as follows

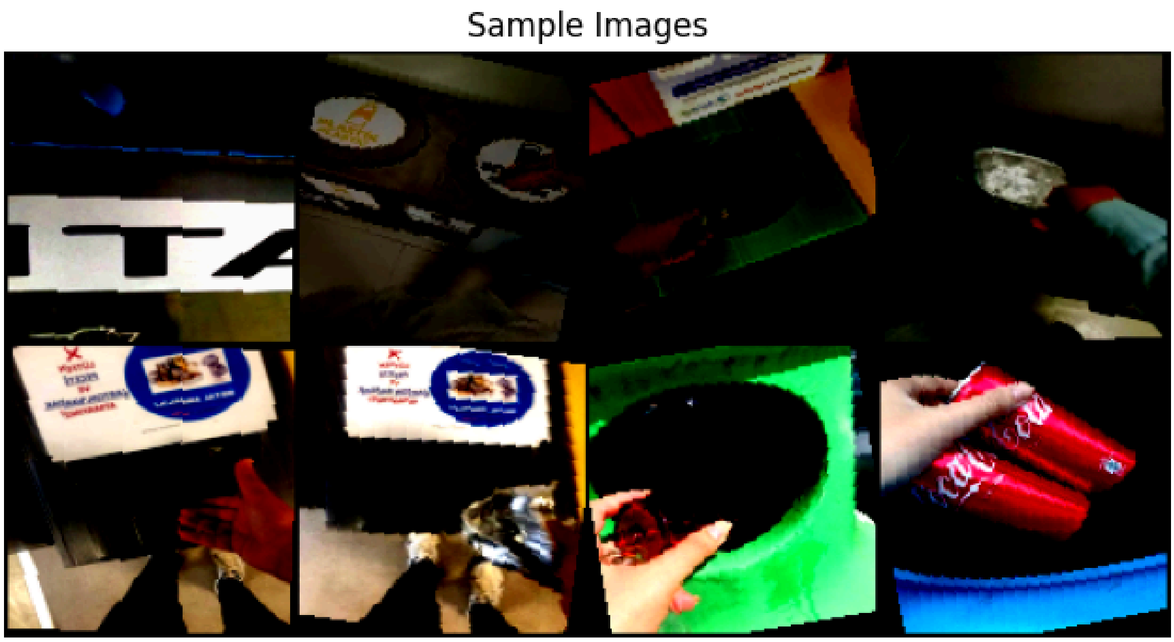
4. **Display image size distribution:** The image width and height distribution were displayed using a histogram. The chart below confirms the image width and height dimensions to be 640x640. The distribution is uniform for all images.



5. **Show class distribution:** The distribution of each class in the training dataset was visualized through a bar chart as follows:



6. **Perform advanced transformations:** The data was augmented using multiple transformations such as random flip, random rotation, color jitter, to tensor, normalization, and resizing. The resulting images were displayed as follows:



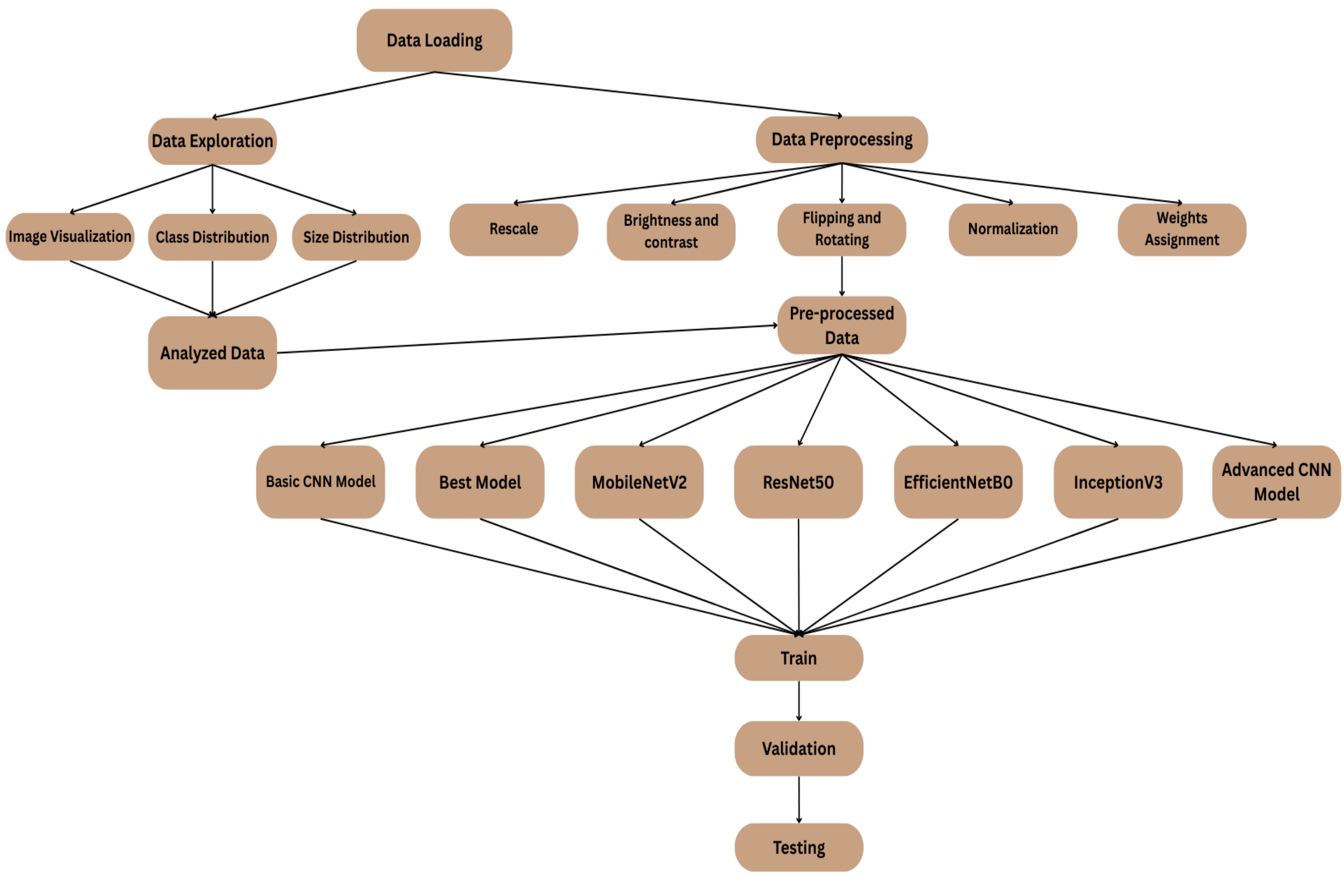
7. **Assign class weights:** Class weights were assigned to classes with lower frequencies to avoid bias in the dataset and the model.

3.6 Preliminary Analysis

Based on the explorations and visualizations conducted, a clear understanding of class distribution was determined, where it was discovered that the highest frequency of images belongs to "glass", closely followed by "metal", and "bin". "Plastic" has a low frequency in the dataset, and the "Other" waste items class is almost negligible as compared to the other classes. This shows a clear imbalance and a need for appropriate weight assignment to prevent that bias from affecting the training of the model.

It was also observed through image visualization that most of the waste products have a similar background, containing waste products in front of the appropriate bins. This can be challenging in real-world scenarios where there is more diversity in the background and lighting. An effort has been made to vary the contrast and brightness. The images have also been rescaled to convert them to grayscale. Rotation and flipping have been applied to augment the dataset.

4. ARTEFACE DEVELOPMENT



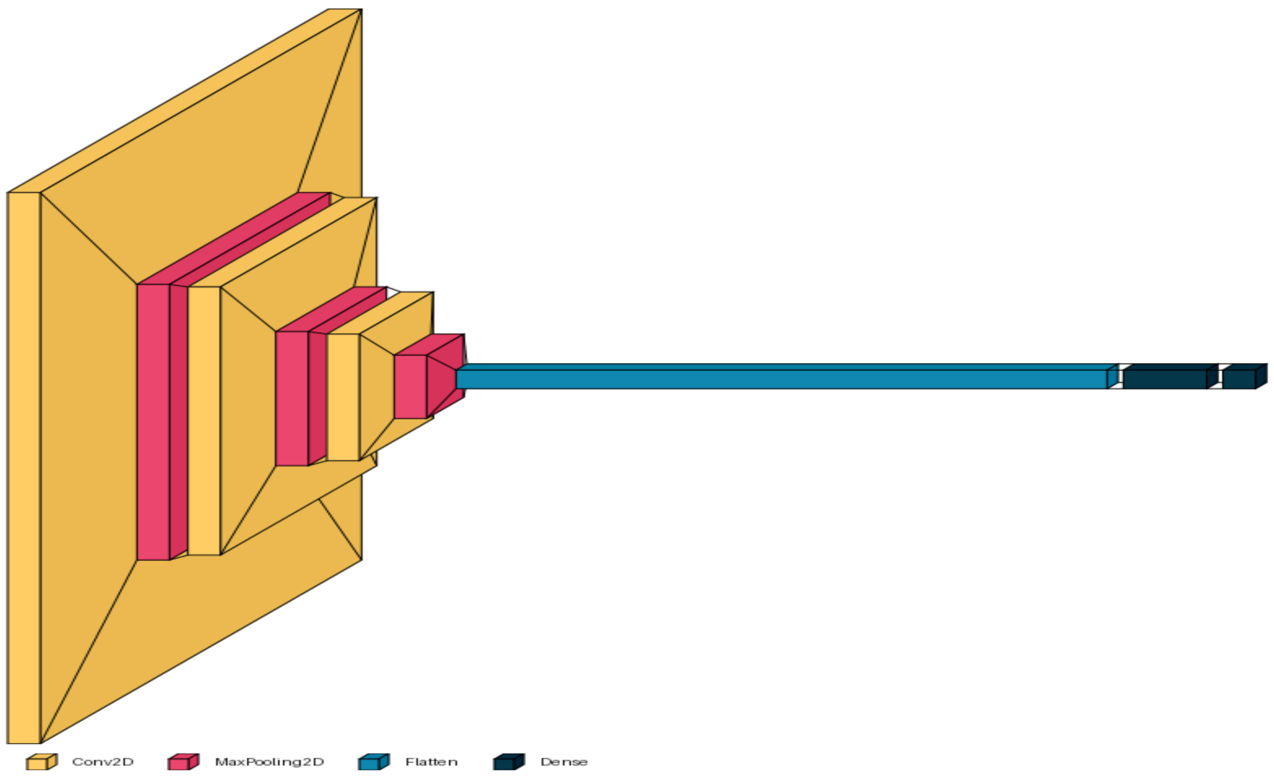
The next steps include building and training a basic deep learning model to see how well it performs on the given dataset. A workflow of the steps performed has been given below:

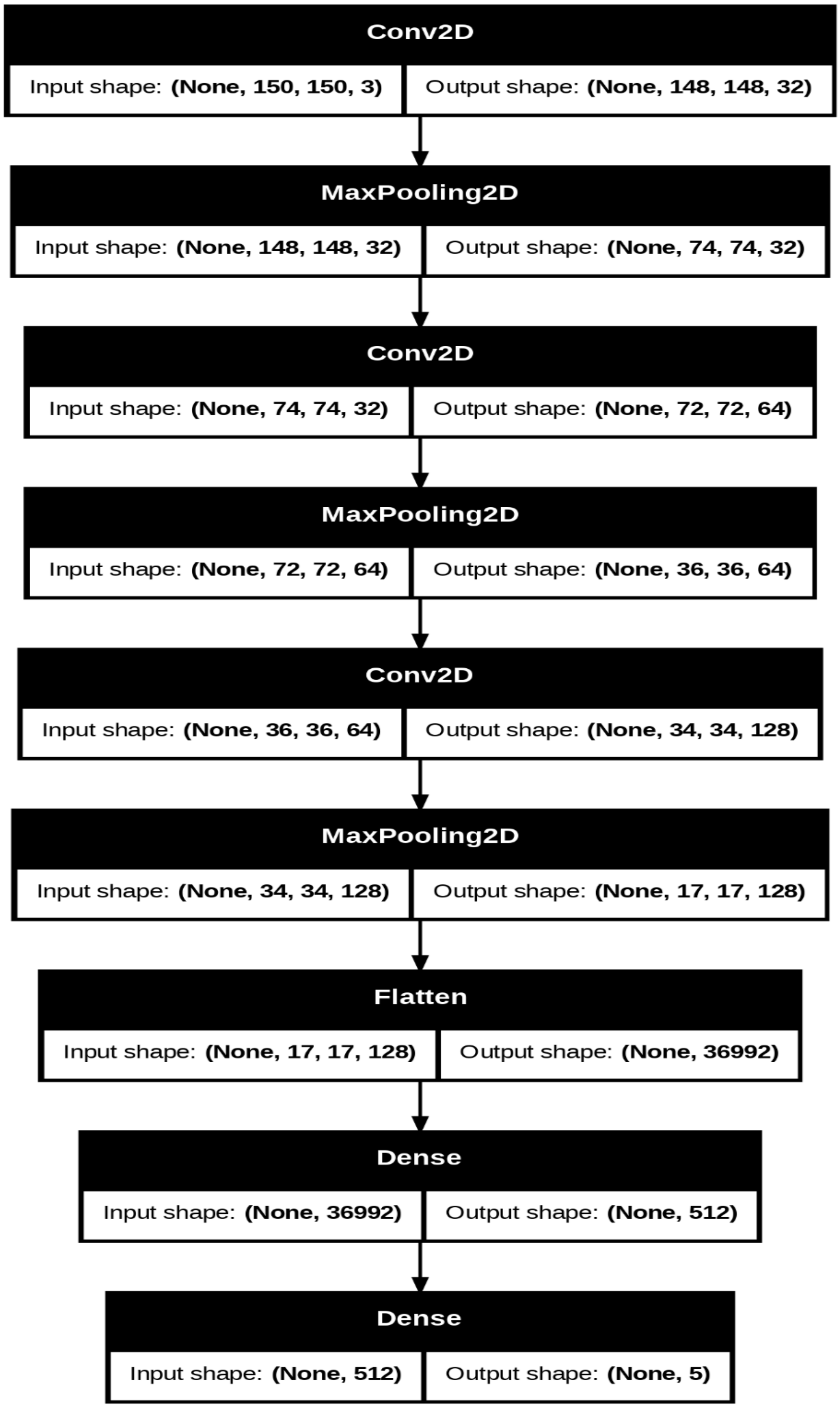
4.1 Basic Model

For the basic model, a simple Convolutional Neural Network (CNN) was trained containing the following layers: [13]

- **Conv2D:** This layer applies 32/64/128 filters of size 3x3 to the image. It also uses an activation layer, “relu”, to introduce non-linearity and learn complex patterns.
- **MaxPooling** Helps extract the most informative features and downsamples.
- **Dense:** This is the fully-connected layer used for classification purposes.
- **Flatten:** Helps with dimensionality reduction by converting the 3D map to a 1D vector.

The model design is shown as follows:



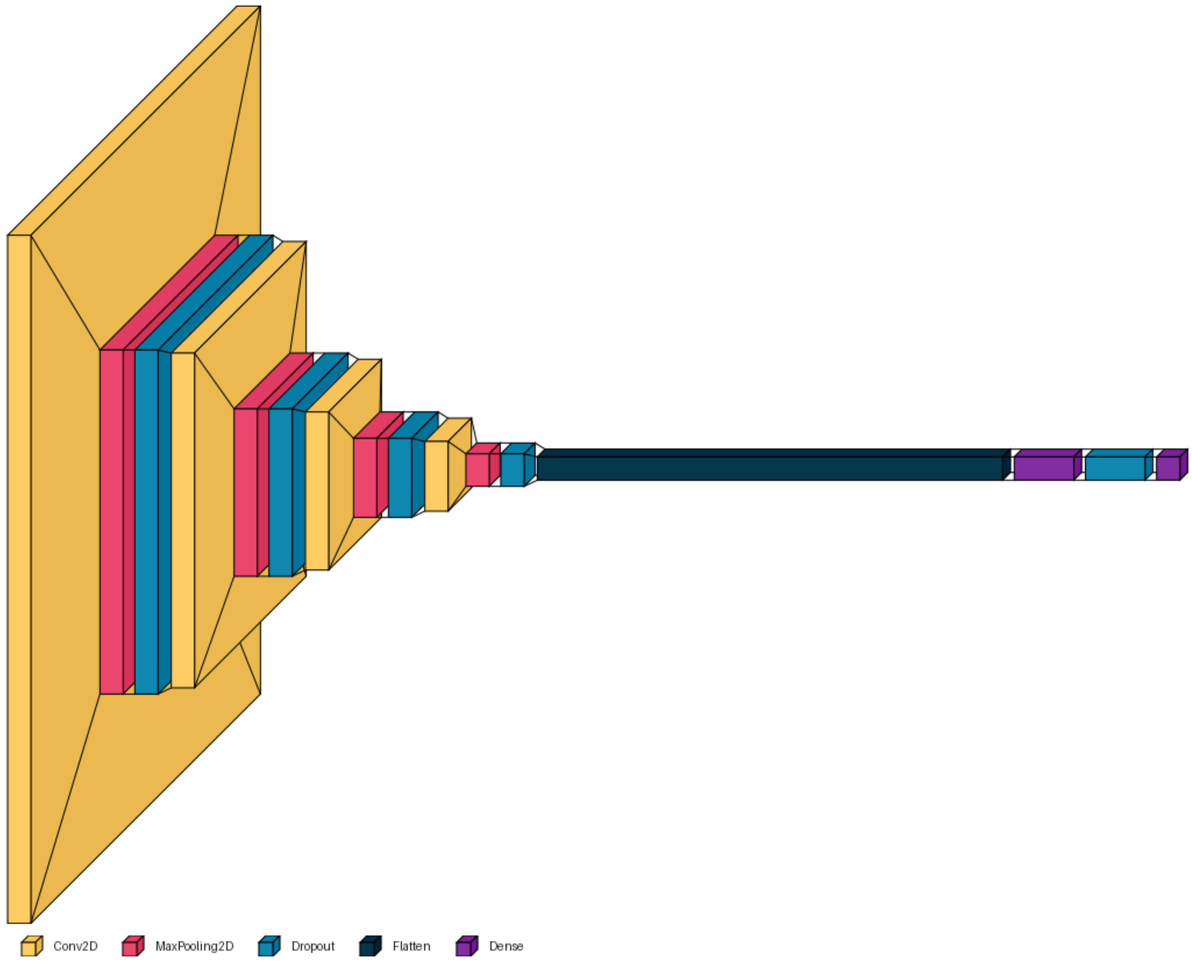


4.2 Advanced Model

The advanced model consists of 2 main additional features described as follows: [13]

- **Dropout:** Is a regularization technique that prevents overfitting.
- **Regularization:** Introduces L2 regularization to penalize large weights and reduce the risk of overfitting in the model.

The structure of the advanced CNN model is given below:



4.3 Model training

Both models are trained with the following parameters:

Basic Model

1. **Epochs: 10**
2. **Learning Rate: 0.001**
3. **Loss: Categorical Crossentropy**
4. **Metrics: Accuracy**

Advanced Model

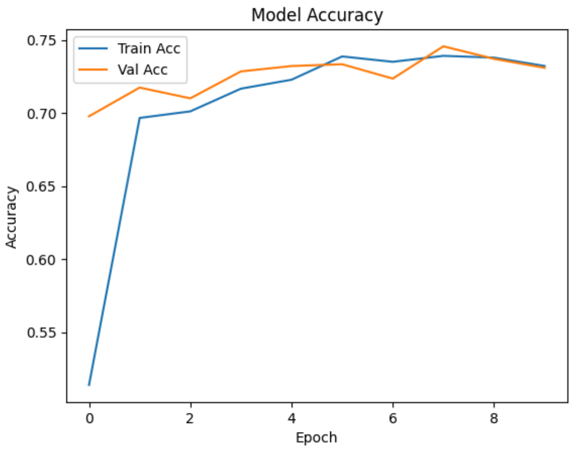
1. **Epochs: 10**
2. **Learning Rate: 0.005**
3. **Loss: Categorical Crossentropy**
4. **Metrics: Accuracy**

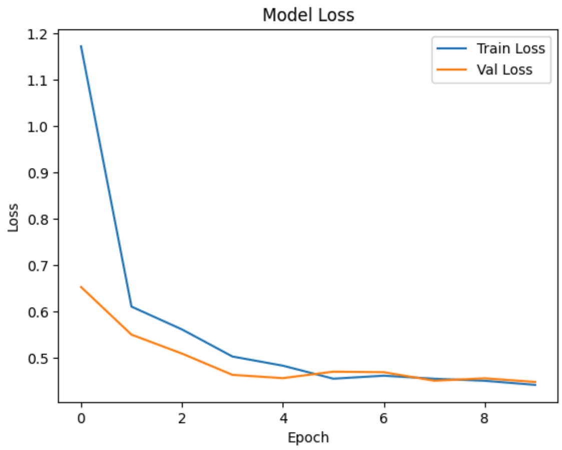
Training and Validation Accuracies

The training and validation accuracy metrics and categorical cross-entropy loss were recorded at each epoch, and a line chart was drawn to compare the accuracies and losses. The figures for both models are as follows:

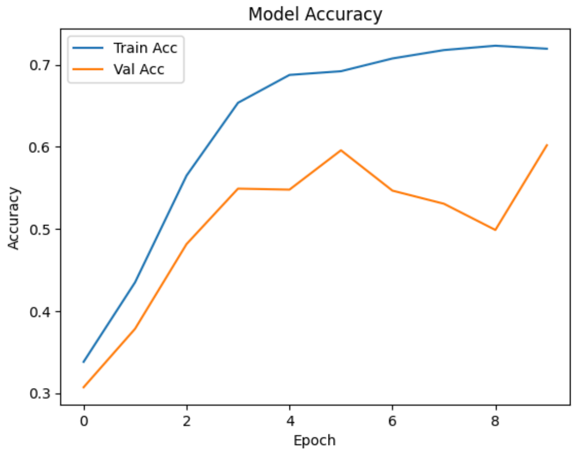
Basic Model

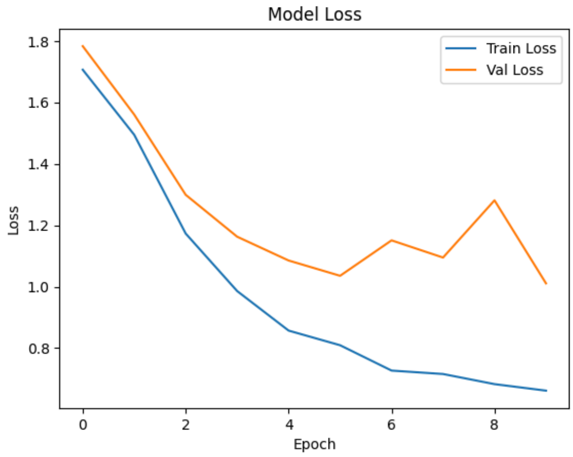
The basic model doesn't show a high accuracy, reaching only up to 75% for both training and validation data. Validation accuracy starts high, but tends to saturate between 70% - 75%. There is room for improvement in this model.





Advanced Model





The advanced model clearly shows signs of overfitting, where even though the training accuracy increases, the validation accuracy increases and then starts decreasing after a threshold. This requires further inspection of the model and calls for improvement. The accuracy for this model also seems to be lower than the accuracy of the basic model.

Testing Accuracies

The models were tested on the test dataset, and the following accuracies were recorded:

Basic Model: 72.80%
Advanced Model: 60.36%

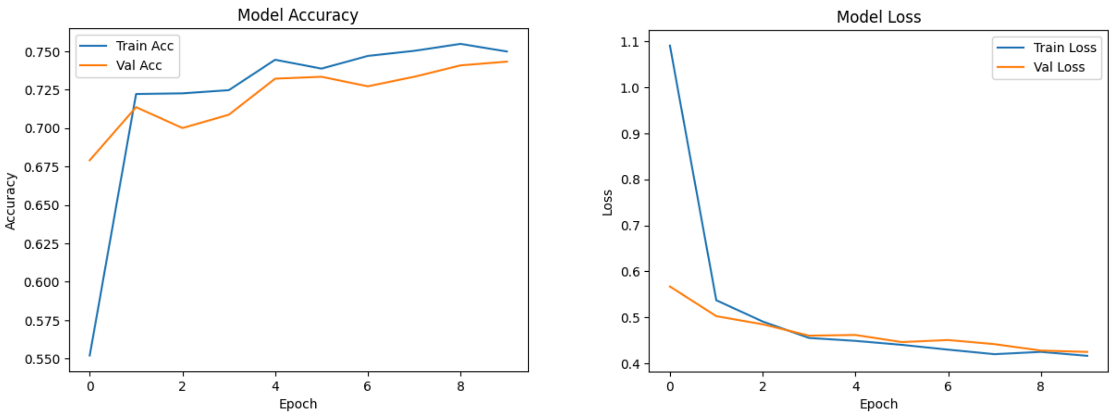
4.4 Checking the Minority Class effect on Model Performance

Sometimes, minority classes hinder the model's performance due to their under-representation in the dataset. This is because the model is not able to learn the features of the minority classes as well as those of the majority classes. In this section, we try to observe the effect of the minority class `Other` on the model's performance by training the model without the `Other` class. For this process, we fit the basic and advanced models again. The only difference is in the dataset, where the `Other` class has been temporarily removed.

Basic Model Without `Other`

1. **Epochs: 10**
2. **Learning Rate: 0.001**
3. **Loss: Categorical Crossentropy**
4. **Metrics: Accuracy**

The training accuracy goes up to 76.67% while the validation accuracy goes to 74.32%. The testing accuracy also increases to 73.94%. We can see some improvement in the model's performance.

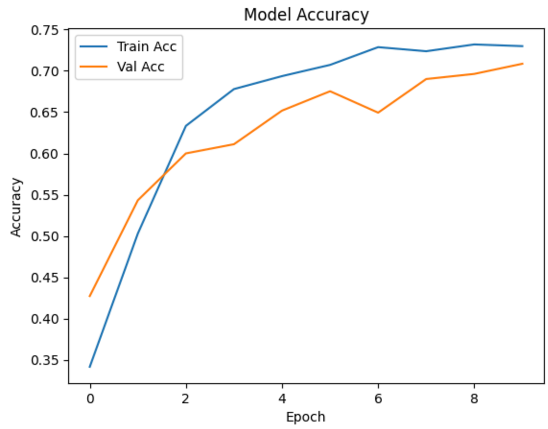


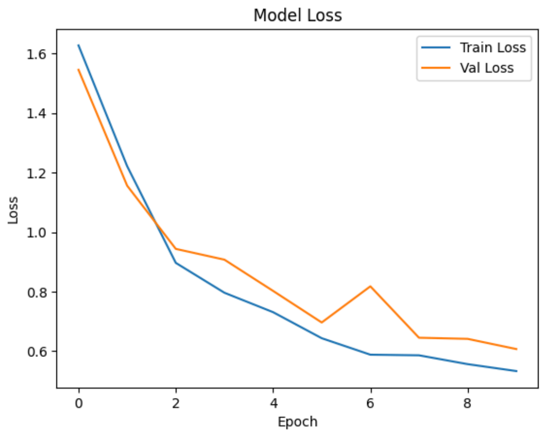
Advanced Model Without `Other`

1. **Epochs: 10**
2. **Learning Rate: 0.005**
3. **Loss: Categorical Crossentropy**
4. **Metrics: Accuracy**

The biggest improvement seen in this case is the removal of overfitting. After removing the `Other` class, the model stops overfitting and gives slightly better performance metrics as well. The training accuracy goes to 75.00% while the validation accuracy goes to 70.86%. The testing accuracy is also improved and goes to 68.90% from 60.00%.

The effect of the minority class `Other` indicates that there is a need to eliminate the class imbalance in the dataset. For this purpose, we explore stratified k-fold cross-validation along with hyperparameter tuning in the next section.





5. METHODS

The advanced model was taken and trained to achieve the best hyperparameters. Since the dataset used in this project is highly imbalanced, the method of stratified k-fold cross-validation was employed to overcome this problem.

5.1 Standard k-fold cross-validation

In standard k-fold cross-validation, the model is robustly trained k number of times separately. Each training iteration involves training on k-1 folds and validation on the remaining number of folds. This procedure ensures more reliability in the performance of the model by taking the average of each fold. It also prevents overfitting and underfitting and ensures generalizability of the model, so it performs well on unseen data. The problem, however, arises when the dataset is imbalanced. In such a scenario, it is crucial to ensure that every training fold has an equal number of all the classes. This is where stratified k-fold cross-validation comes in.[14]

5.2 Stratified k-fold cross-validation

Stratified k-fold cross-validation works in a similar way to k-fold cross-validation. However, it ensures that every training set has an equal class distribution as the entire dataset. This is beneficial to our model because of the high imbalance in classes present in the dataset used. To perform stratified k-fold cross-validation, the `StratifiedKFold` library from scikit-learn was implemented with n_splits = 3, ensuring 3 folds. The model was trained on 2 folds and validated on the third fold for every configuration of the hyperparameters. [15]

5.3 Hyperparameter Tuning

The Hyperparameters of a model are the parameters that need to be determined before training the machine learning model[16]. The most optimal hyperparameters are the ones that help improve the model's performance. In this case, the focus was on **batch size, dropout rate, and number of epochs** because these hyperparameters directly influence model generalization, training stability, and efficiency, which were the most critical factors considering the dataset size and computational constraints.

1. **Batch Size:** This refers to the number of samples used in a single training iteration. It directly affects the speed of training. In this case, the batch sizes experimented with were 32 and 64. Any batch size larger than that would need more GPU memory and may lead to poor generalization.
2. **Dropout Rate:** This hyperparameter helps prevent overfitting by dropping a random portion of neurons during training. In this project, the dropout rates used were 0.3 and 0.5. A lower rate (0.1-0.2) would not provide enough regularization and could increase the risk of overfitting, and a higher rate (0.5+) would drop too many neurons and could result in underfitting by slowing down convergence.
3. **Number of Epochs:** This hyperparameter controls the number of iterations for which the model is trained. In this case, 20 and 30 epochs were used. Training with less than 20 epochs, the model would not converge properly and training for a high number of epochs (50+) would cost a lot of computational time and risks overfitting as the dataset is not that large.

5.4 Transfer Learning

The transfer learning approach utilizes existing models that have been rigorously trained on large-scale datasets. The pre-trained neural network models are used to perform similar classification tasks in deep learning models. This is done by transferring the learned weights and general features from the pre-trained model to perform the new task. Transfer learning is preferred when the dataset is small and resembles features similar to those used for the pre-trained model. To understand this better, we will take a look at the layers:[17][18]

- **Early Layers:** These layers are responsible for learning the general transferable features from the model.
- **Middle Layers:** These layers take care of identifying parts of the objects. They are also general and transferable.
- **Final Layers:** These layers are used for performing a specific task and thus need to be customized for each task.

5.6 Fine Tuning

While in transfer learning, all convolutional layers are frozen, fine-tuning introduces the option of temporarily unfreezing some or all the convolutional layers and training them again. This ensures training the pre-trained model's layers for the problem at hand. It is more refined and offers better results. [17]

In this section, various pre-trained models have been explored to perform the task of classifying various waste materials properly. The performance of these pre-trained models after transfer learning and fine-tuning is studied and compared to each other. Below is a detailed explanation of each model used and its performance on this dataset:

1. **MobileNet**
2. **ResNet50**
3. **EfficientNetB0**
4. **InceptionV3**

6. MODEL TRAINING

6.1 Best Model after Hyperparameter Tuning and Cross-Validation

The stratified k-fold cross-validation combined with grid search hyperparameter tuning helps optimize the best model performance and gives better aggregated results. This was done by fitting the model for different combinations of hyperparameters over 3 folds and calculating the average accuracy of the model across the 3 folds. Finally, the optimal model parameter configuration and model weights were saved for later use.

The model that performed the best had the following hyperparameters:

Best Params -> Batch Size: 64, Dropout: 0.3, Epochs: 30
Best Validation Accuracy: 0.6492

6.2 MobileNetV2

MobileNetV2 is a lightweight convolutional neural network (CNN) model that has been trained on the ImageNet dataset for image classification problems. It was designed by Google researchers as an advancement to the original MobileNetV1 model for increasing accuracy and reducing complexity. The MobileNetV2 model uses the same feature of Depthwise Separable Convolution that was used in the MobileNetV1 model. The new features that enhance performance are the linear bottlenecks and the inverted residuals. The important features of MobileNetV2 are: [19] [20]

1. *Depthwise Separable Convolution:* This feature exists in MobileNetV1 and combines the Depthwise Convolution and Pointwise Convolution to decrease the number of parameters and convolution cost.

2. *Linear Bottlenecks:* This feature is responsible for linearly transforming the layers from the input domain into lower-dimensional layers while using ReLU as the activation function.

3. *Inverted Residuals:* These inverted residual blocks are similar to the bottleneck blocks. They first follow the process of reducing the dimensionality by incorporating information and then expanding it. This is done before the application of a linear transformation. The inverted residuals make the model more memory efficient.

Model Architecture

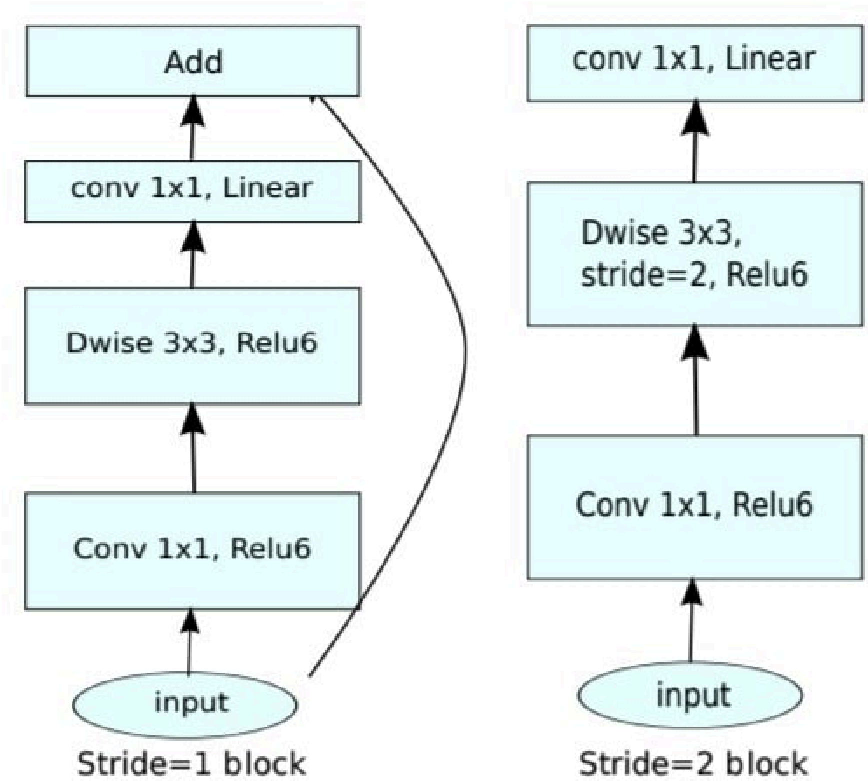


Image 1

In this project, the MobileNetV2 model is used to perform transfer learning and execute the waste classification task. Below are the steps executed and the results obtained, discussed in detail.

6.2.1 Without Fine-tuning

Process

We have used the pre-trained model MobileNetV2, trained on ImageNet dataset, to extract the generic transferable features and potentially increase the precision of the model's performance. In this regard, the MobileNetV2 model is considered a feature extractor.

Layers

- First, we froze the base model to extract features.
- Added trainable layers at the top to perform the waste classification task.
- **GlobalAveragePooling2D, Dense(512) + Dropout** were added.
- A final **Softmax layer** was added for classification.
- The model was evaluated on **precision, recall, accuracy, and Mean Average Precision (mAP).**

6.2.2 With Fine-Tuning

The next step in the process is to fine-tune the pre-trained model and compare the results obtained above.

Layers

- For fine-tuning, the last 30 layers were unfrozen. These are the deeper layers of the model that are not trained in normal transfer learning.
- The MobileNetV2 module was used from Keras.
- A low learning rate was chosen to keep the weights from the original model intact.
- Again, the model was trained and evaluated on the same metrics.

6.3 ResNet50

ResNet stands for Residual Networks. The ResNet50 model belongs to this class of deep convolutional neural networks and has 50 layers, which is considered a mid-size variant. The ResNet models are widely used for image classification tasks due to their high efficiency and depth. The depth offered by ResNet50 is efficient in extracting hierarchical features for image recognition tasks. It is also necessary to pass information to deeper layers using its shortcut connections. This helps remove the problem of vanishing gradient. The residual blocks in ResNet50 are responsible for solving the problem of degradation while training deep networks. The important components of the residual block are discussed below: [21][22]

1. *ReLU Activation:* It is important to introduce non-linearity in the model to capture intricate and complicated patterns.

2. *Bottleneck Convolutional Layers:* It consists of 3 convolutional layers with filter sizes of 1x1, 3x3, and then 1x1 followed by batch normalization layers. These convolutional layers are important in dimensionality reduction, feature extraction, and restoring to the original number of channels.

3. *Skip Connection:* The most important feature of the ResNet50 model is the skip connections, also known as the shortcut connections. These skip connections are responsible for providing a shortcut path for the gradient to pass through. They also

help the model learn an identity function to make sure that the higher layers perform as well as the lower layers.

Model Architecture

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Image 2

Fine-tune

For this model, we will directly perform transfer learning with fine-tuning and compare the results with the MobileNet model.

Layers

1. Again, the top few layers were unfrozen and trained. In this case, we took the top 140 layers because of the high model depth.
2. The ResNet50 module was used from Keras.
3. The learning rate was set at a low value to preserve the original weights of the base model.
4. The metrics used for evaluation included accuracy, precision, recall, and mAP.

6.4 EfficientNetB0

EfficientNetB0 belongs to the Efficient family. It is one of the smallest members and a baseline of this family. This model is pre-trained on the ImageNet dataset. While ResNet provides large model depth, EfficientNetB0 was designed by *Tan and Le* to provide accuracy along with computational efficiency and minimal model size. This is achieved by the compound scaling layers used in EfficientNetB0. The compound scaling ensures a uniform scaling of the depth, width, and resolution of the network. The initial layer of this compound scaling model is the Stem layer. This model is pre-trained and used via transfer learning for image classification tasks. It has been used for Medical Image Analysis tasks. It is particularly useful on devices where computational resources are limited. The layers and components that make up the architecture of EfficientNetB0 are discussed below in detail: [23]

1. *Stem Layer:* This is the initial layer of convolutional neural networks. In EfficientNetB0, this layer consists of a convolutional Conv2D layer, batch normalization, and a swish activation function. The swish activation function is a smooth, non-linear activation function. The smoothness property of swish offers better optimization.

2. *MBConv Blocks:* These blocks are the primary building blocks of the EfficientNetB0 model, similar to the ones used in MobileNetV2. Their main components consist of Depthwise Separable Convolutions for reducing the number of multiplications, stem connections to avoid gradient degradation, an expansion phase for creating more complex features, and a projection phase for controlling the size of the model.

Model Architecture

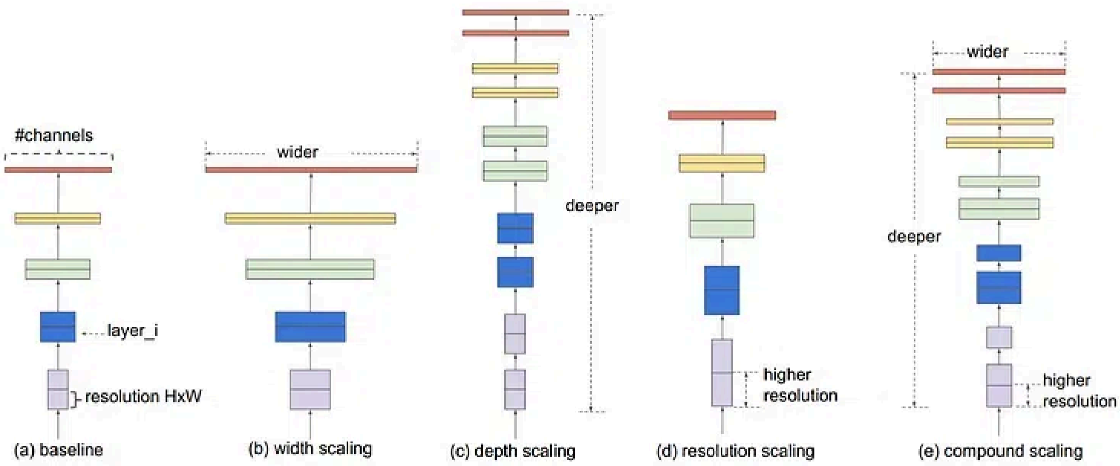


Image 3

Fine-Tune

Again, we perform fine-tuned transfer learning for the EfficientNetB0 model to perform the task of waste classification for the given dataset. We compare the results from this model with the previous results and identify the most accurate and efficient model.

Layers

1. EfficientNetB0 has 237 layers. For this process, a mid-point or mid-depth was chosen to unfreeze the layers.
2. The top 100 layers were unfrozen and trained on the dataset.
3. The EfficientNetB0 library was used from Keras.

4. The learning rate was chosen to be an infinitesimally small number to retain the original model weights.
5. The performance metrics were calculated, and accuracy, precision, recall, and mAP were compared with other models.

6.5 InceptionV3

InceptionV3 is a deep convolutional neural network belonging to the Inception family. This model was designed to perform deep learning tasks with computational efficiency. It offers a great depth, but is a manageable model that ensures no overfitting in the model. It provides high accuracy for large-scale models like ImageNet. It was designed by Google and has been pre-trained on the ImageNet dataset. The key components of the InceptionV3 model that make it extraordinary are inception modules, auxiliary classifiers, and factorized convolutions. The layers and components that make up the InceptionV3 model are discussed below: [24]

1. *Inception Module:* The inception module is responsible for feature extraction by employing filters of various sizes in a parallel convolution. This helps with multi-level feature capture. The inception module is the core building block of InceptionV3.

2. *Factorized Convolutions:* These convolutions are responsible for reducing computational cost and the number of parameters. This is done by factorizing large convolutions into smaller convolutions while retaining knowledge transfer.

3. *Auxiliary layers:* These are the classifiers using the softmax activation function attached to intermediate layers. This helps in the flow of information along deep networks without gradient degradation and overfitting. These classifiers act as regularizers and improve training.

Model Architecture

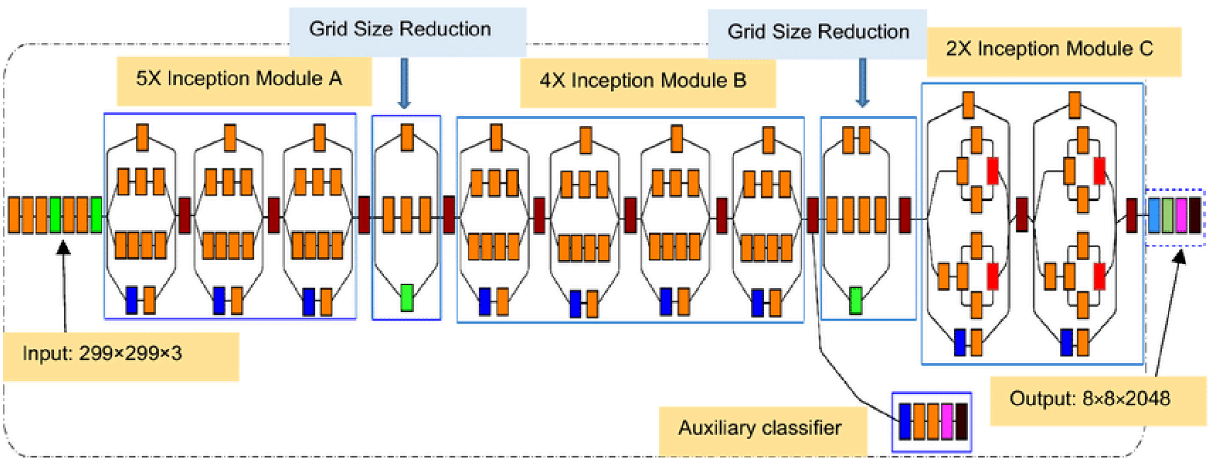


Image 4

Fine-Tune

Again, we perform fine-tuned transfer learning for the InceptionV3 model to perform the task of waste classification for the given dataset. We compare the results from this model with the previous results and identify the most accurate and efficient model.

Layers

1. The layers were frozen at 249 because these are the more task-specific layers in the InceptionV3 model. Meanwhile, the earlier frozen layers were used for preserving generic and transferable features.
2. The InceptionV3 module from Keras was used to implement this process.
3. The learning rate was chosen to be an infinitesimally small number to retain the original model weights.
4. The performance metrics were calculated, and accuracy, precision, recall, and mAP were compared with other models.

7. RESULTS

7.1 BEST MODEL

We took the advanced model for performing the stratified k-fold cross-validation and hyperparameter tuning. Below is the table for comparing model performance for all hyperparameter combinations across different folds:

Batch Size	Dropout Rate	Epochs	Fold 1 (%)	Fold 2 (%)	Fold 3 (%)	Average Accuracy (%)
32	0.3	20	61.89	65.15	67.73	**64.92**
32	0.3	30	64.83	62.70	66.99	**64.84**
32	0.5	20	66.87	69.08	33.99	57.31
32	0.5	30	51.10	41.84	66.38	53.11
64	0.3	20	26.72	64.54	31.90	41.05
64	0.3	30	67.28	62.09	65.40	**64.92**
64	0.5	20	44.00	53.50	56.20	51.23
64	0.5	30	31.86	46.75	31.90	36.84

The best model performance ties at an average accuracy score of **64.92%** at 2 hyperparameter combinations given as follows:

- **Batch Size 32, Dropout 0.3, Epochs 20**
- **Batch Size 64, Dropout 0.3, Epochs 30**

The first hyperparameter combination with batch size 32 and epochs 20 was chosen as the best model configuration. It provides not only better accuracy, but also faster execution.

Evaluating the Best Model

In order to test the model's performance, the original platform from which the dataset was acquired was taken as a reference. On this platform, the model was tested by studying the precision, recall, and Mean Average Precision (mAP). In order to compare the performances of the existing model and the best model obtained through cross-validation and hyperparameter tuning, these accuracy measures were included in the model.

The initial preprocessing steps were used, and the model was tested with the best hyperparameters. Initially, the model gave satisfactory results with all the preprocessing steps included; however, upon further inspection, it was determined that the model performed better without the brightness and contrast adjustment carried out during the preprocessing of the dataset. The results before and after the removal of brightness and contrast adjustment are recorded below:

BEST MODEL WITH BRIGHTNESS AND CONTRAST ADJUSTMENT

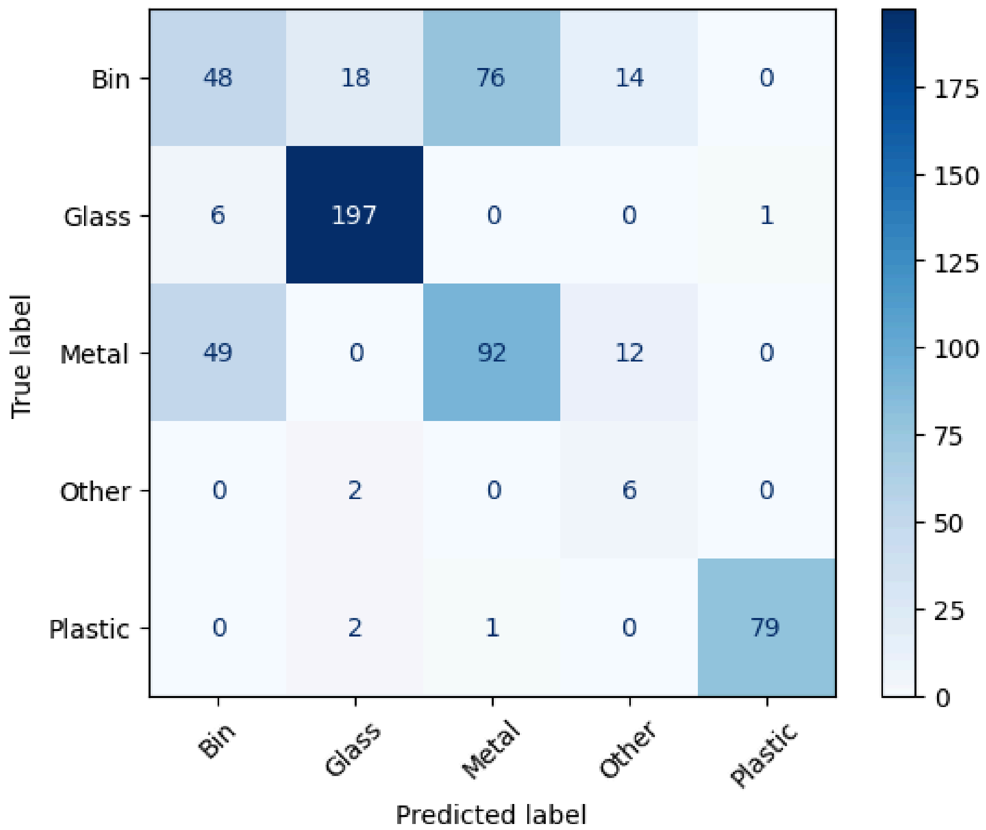
	precision	recall	F1 score	support
Bin	0.38	0.13	0.20	156
Glass	0.89	0.64	0.74	204
Metal	0.47	0.69	0.56	153
Other	0.07	0.88	0.13	8
Plastic	0.97	0.87	0.92	82
accuracy			0.56	603
Macro avg	0.55	0.64	0.51	603
Weighted avg	0.65	0.56	0.57	603

BEST MODEL WITHOUT BRIGHTNESS AND CONTRAST

	precision	recall	F1 score	support
Bin	0.47	0.31	0.37	156
Glass	0.90	0.97	0.93	204
Metal	0.54	0.60	0.57	153
Other	0.19	0.75	0.30	8
Plastic	0.99	0.96	0.98	82
accuracy			0.70	603
Macro avg	0.62	0.72	0.63	603
Weighted avg	0.70	0.70	0.69	603

Mean Average Precision (mAP): 0.6831

Below, we also have a visualization of the confusion matrix:



Looking at the output tables, it can be concluded that the model without brightness and contrast adjustment outperforms the one with these adjustments. This can be distinctly concluded through the accuracy, macro avg, and weighted avg, which show a better estimate when the adjustments are removed. **It can be explained as an outcome of the distortion caused in the image pixels, especially for low-quality images and overlapping objects like `Bin` and `Other`**.

While the model was able to achieve similar scores as the online reference, particularly Mean Average Precision mAP (70%), and Recall (70%), the model failed to achieve a high Precision score of 90.7%.

Confusion Matrix Explanation

The confusion matrix shows that `Glass` is the most distinguished and easily recognisable class among all classes. The `Plastic` class is also clearly identified, as can be verified from the confusion matrix. The clear confusion between `Bin` and `Metal' is most apparent, and seems like a clear cause for concern in this dataset. In many instances, `Bin` and `Metal` are misclassified as each other.

Bin → Metal (76)

Metal → Bin (49)

This area justifies improvement to enhance the model's performance and achieve better results.

7.2 MobileNetV2

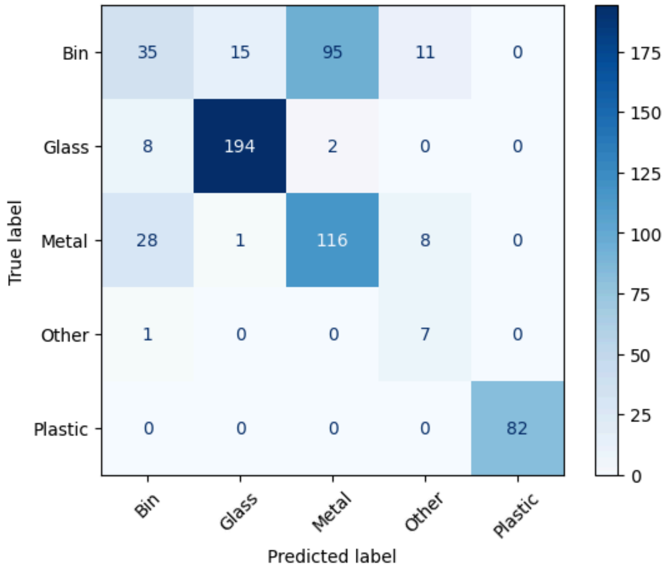
7.2.1 Without Fine-tuning

Outputs

	precision	recall	F1 score	support
Bin	0.49	0.22	0.31	156
Glass	0.92	0.95	0.94	204
Metal	0.54	0.76	0.63	153
Other	0.27	0.88	0.41	8
Plastic	1.00	1.00	1.00	82
accuracy			0.72	603

Macro avg	0.64	0.76	0.66	603
Weighted avg	0.72	0.72	0.70	603

mAP score: 0.73



Improvements

- The overall performance of the model has improved after performing transfer learning using MobileNetV2. This includes precision, accuracy, recall, and mAP.
- The model was able to abate the **`Metal`** vs **`Bin`** confusion, as the **recall** for `Metal` went from **0.57 to 0.76.**
- There was a notable improvement in the **`Other`** class as the **F1 score** increased from **0.30 to 0.41**. This is highly significant, especially given the low support of 8 for this sample.

Drawbacks

- The performance metric for `Bin` got slightly worse.

7.2.2 With Fine-Tuning

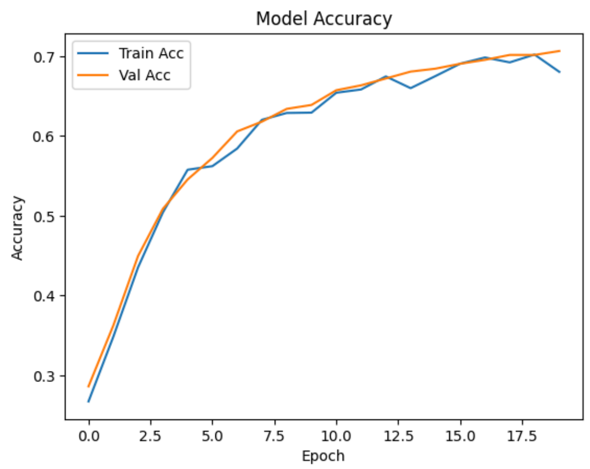
Outputs

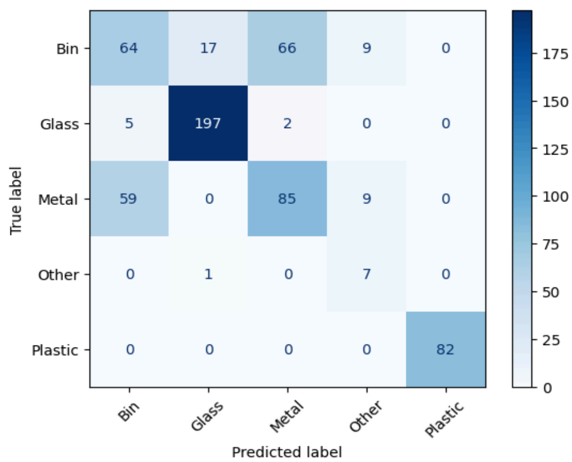
	precision	recall	F1 score	support
Bin	0.50	0.41	0.45	156

Glass	0.92	0.97	0.94	204
Metal	0.56	0.56	0.56	153
Other	0.28	0.88	0.44	8
Plastic	1.00	1.00	1.00	82
accuracy			0.72	603
Macro avg	0.65	0.76	0.67	603
Weighted avg	0.72	0.72	0.72	603

mAP score: 0.76

Below are the confusion matrix and accuracy curve for this fine-tune transfer learning approach using MobileNet.





Improvements

- There is a clear improvement in the precision and recall for `Bin`. It shows that `Bin` is better classified than before.
- `Plastic` and `Glass` show nearly 100% precision. This is comparable to the reference precision metrics on the website.
- The overall mAP increased. This value is significantly higher than the reference metrics given on the website.

Drawbacks

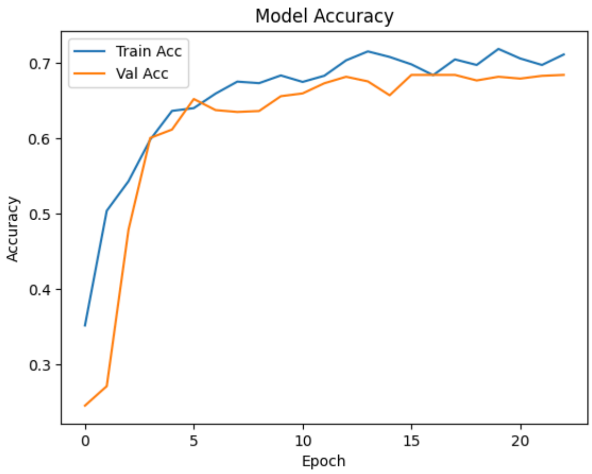
- The performance for `Metal` slightly worsened. However, given the overall model enhancement, it is acceptable.

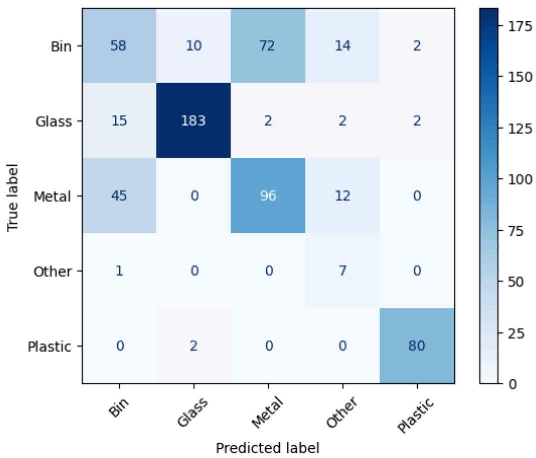
7.3 ResNet50

Outputs

	precision	recall	F1 score	support
Bin	0.49	0.37	0.42	156
Glass	0.94	0.90	0.92	204
Metal	0.56	0.63	0.59	153
Other	0.20	0.88	0.33	8
Plastic	0.95	0.98	0.96	82
accuracy			0.70	603
Macro avg	0.63	0.75	0.64	603
Weighted avg	0.72	0.70	0.71	603

mAP score: 0.68





The results obtained show that ResNet50 doesn't perform better than MobileNet. It can be clearly seen in the overall performance of the model by looking at the mAP score, which seems to have reduced for ResNet50. It also has lower F1 scores for `Bin`, `Other`, and `Plastic`. It only outperforms `Metal` by increasing recall, but it is not sufficient to overcome the losses. There is still a class confusion after applying the fine-tuned ResNet50. Below are the confusion matrix and accuracy charts to study the model performance. While the model itself has some satisfactory metrics, MobileNet outperforms it in almost every aspect. The confusion matrix shows the confusion between classes like `Bin` and `Metal`.

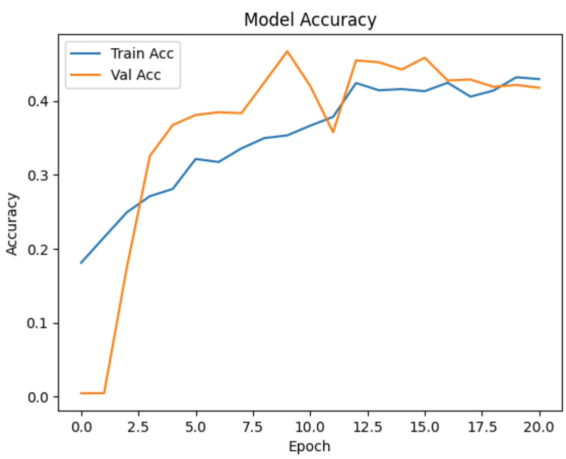
7.4 EfficientNetB0

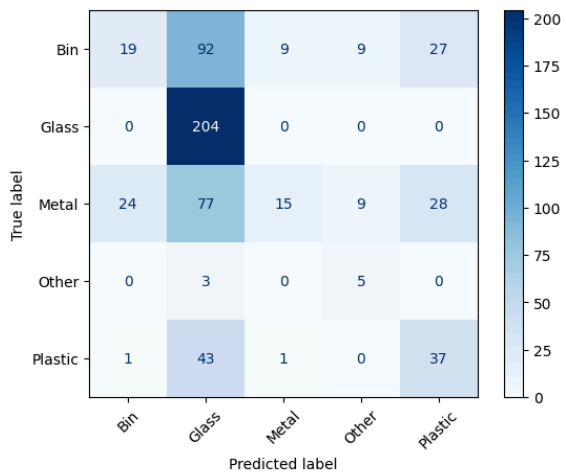
Outputs

	precision	recall	F1 score	support
Bin	0.43	0.12	0.19	156
Glass	0.49	1.00	0.65	204
Metal	0.60	0.10	0.17	153
Other	0.22	0.62	0.32	8
Plastic	0.40	0.45	0.43	82
accuracy			0.46	603
Macro avg	0.43	0.46	0.35	603
Weighted avg	0.49	0.46	0.38	603

mAP score: 0.57

Confusion matrix and accuracy chart:





The outputs clearly show that this model is worse than the other models in terms of all evaluation metrics. It can be seen from the mAP that the mean average precision of classification is lower than the rest of the models at a value of 0.57. The precision, recall, and F1 score are lower than MobileNetV2 and ResnNet50 by a significant margin for most of the samples.

Better fine-tuning and an EfficientNet pre-processing on the input data might help improve these results, and is worth exploring.

The confusion matrix and accuracy curve for this fine-tuned EfficientNetB0 model, which show the inefficient and limited performance explained above. It can be visualized from the accuracy chart that the accuracy saturates around 40%, which is the worst performance so far. The confusion matrix also shows a clear confusion between classes, with most of the classes misclassified as `Glass`.

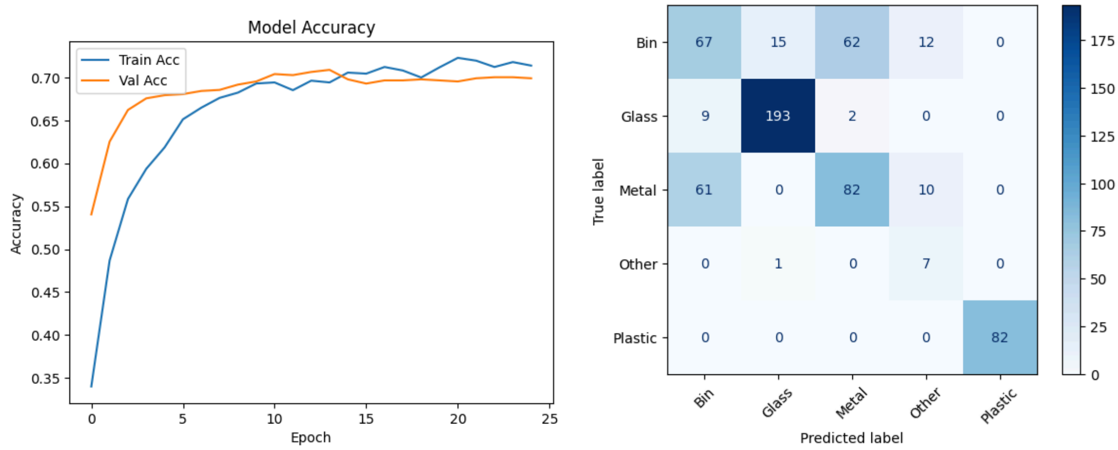
7.5 InceptionV3

Outputs

	precision	recall	F1 score	support
Bin	0.50	0.41	0.45	156
Glass	0.92	0.97	0.94	204
Metal	0.56	0.56	0.56	153
Other	0.28	0.88	0.44	8
Plastic	1.00	1.00	1.00	82
accuracy			0.72	603
Macro avg	0.65	0.76	0.67	603
Weighted avg	0.72	0.72	0.72	603

mAP score: 0.72

The confusion matrix and accuracy chart for this model are given below:



From the mAP calculated for this model, it can be concluded that this fine-tuned transfer learning model performs well for the waste classification task. It gives an overall accuracy and mAP close to the MobileNetV2 model. It identifies the `Plastic` and `Glass` classes with as much precision, recall, and F1 score as the MobileNetV2 model. However, the important classes like `Other` show lower precision and F1 scores of 0.24 and 0.38, as compared to 0.28 and 0.41. There is only a marginal difference in the confusion between `Metal` and `Bin`. However, MobileNetV2 was able to achieve slightly better results on a smaller number of epochs.

Although the model only marginally loses to MobileNetV2, it proves to be a significant model in this waste classification task, outperforming the fine-tuned transfer learning EfficientNet and ResNet models.

8. TESTING BEST MODEL

Once the most efficient and accurate models were identified, the next step was to save, load, and test them on a single image and check prediction confidence. The confidence score signifies the confidence with which the image was classified into a particular category. It is a value that lies between 0 to 1, with 1 being the highest confidence score. In this project, the 3 models that were found to be the best are listed below:

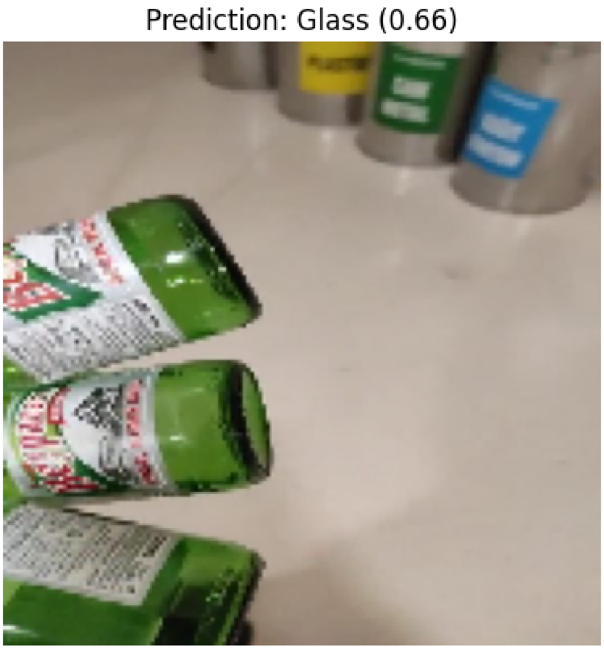
1. **Best model after Hyperparameter Tuning and Cross-Validation** (Model training time: 30-40 minutes; 18-20 hours for Hyperparameter Tuning and Cross-Validation)

2. **Transfer learning and fine-tuning the MobileNetV2 model** (Model training time: ~ 35 minutes)

3. **Transfer learning and fine-tuning the InceptionV3 model** (Model training time: ~ 45 minutes)

The image chosen for testing purposes consists of glass bottles. Each model was tested to classify the image into the right category. The outputs for the tested models are given below.

8.1 Best Model after Hyperparameter Tuning and Cross-Validation



The image above shows the classification into the `Glass` category with 66% confidence.

8.2 MobileNetV2 Model

It can be seen from the output above that the image was classified into the `Glass` category with a 97% confidence score. It is a high classification score for the correct category and shows that the model was able to correctly classify the image with high accuracy.

8.3 InceptionV3 Model

The image above shows that the InceptionV3 model was able to correctly classify the same image into the `Glass` category with a higher score of 100%.

A similar testing of the models can be performed on multiple images from different classes to check which model performs best and is most accurate in identifying the waste items.

9. DISCUSSIONS AND CONCLUSIONS

9.1 Build From Scratch

This project has focused on identifying waste items in the correct categories for users to dispose of their waste efficiently. The most crucial part in this study has been the cleaning and preprocessing of the dataset to implement model training. It is a comprehensive study of implementing a deep learning model from scratch to utilizing existing models and comparing the results obtained. It uses the online metrics available on the platform where the dataset was taken from as reference points of comparison and saves and tests the best models obtained. The preprocessing steps proved informative in gaining insights into this dataset. Image transformations involving flipping, rotating, and standardizing helped refine the image. Class distribution also helped provide insights into model development by distinguishing the minority classes. The removal of class `Other`, which was a minority class, helped understand the drawbacks and reasons for overfitting of the advanced model. The importance of weight assignment and stratified k-fold cross-validation was identified and implemented to obtain the best model. It was also discovered that the removal of certain preprocessing steps, like brightness and contrast adjustment, helped improve model performance, and built the best model from scratch.

9.2 Transfer Learning

Once the best model with an appropriate batch size, number of epochs, and dropout rate was obtained, the concept of transfer learning and fine-tuning was explored and executed for this dataset to improve performance and perform comparisons between multiple models. The models chosen for this purpose were MobileNetV2, InceptionV3, EfficientNetB0, and ResNet50. It was identified that MobileNetV2 and InceptionV3 gave the best performance in terms of precision, recall, accuracy, and mAP. They gave results slightly better than the best model built from scratch by removing the confusion between classes like `Metal` and `Bin`, and by improving the identification of minority classes like `Others`. It was further proven through testing that these models gave high confidence in correctly classifying waste items.

9.3 Results and Reference Metrics

The reference metrics chosen for this project were taken from the online development of waste classification models. This online metrics mentions 90% precision, 70% recall, and 70% mAP as the ideal performance standards. In addition to the online metrics, we also studied and compared accuracy of the models. It was interesting to note that while InceptionV3 gave the highest confidence in classification with a confidence score of 100%, the overall performance in terms of accuracy, precision, recall, mAP, and time-consumption for model training of MobileNetV2 was better than InceptionV3 and the best model built from scratch. MobileNetV3 was able to achieve high precision scores ranging between 92%-100% for `Plastic` and `Glass` and average precision score for classes with confusion like `Metal` and `Bin` with precision and recall scores ranging between 50-56%. The higher overall model performance of MobileNetV2 can be definitively seen from the higher mAP score of 0.76 for MobileNetV2, 0.72 for InceptionV3, and 0.68 for the best model from scratch. These results are also satisfactory to the reference point of online metrics. Amongst these 3 models, MobileNetV2 gives the highest precision and recall for individual categories.

10. SUMMARY AND PROPOSED FUTURE ANALYSIS

The work in this study explored deep learning models for waste classification. The main focus is on exploring the steps while building a classification model using multiple techniques. The processes involve data analysis, image preprocessing, deep-layer neural network models, and the use of appropriate classification metrics. The work consists of analyzing the imbalance in the dataset, checking the size of the images and standardizing them, checking that the annotations are properly done, and making advanced transformations like rotation, flipping, etc. Once the dataset is prepared and ready to feed to a deep learning model, we created a basic deep learning model and an advanced deep learning model for comparison. The study also showed the impact of removing minority classes on the overfitting problem. The problem was solved by performing stratified k-fold cross-validation and hyperparameter tuning. Further, it delves into transfer learning and fine-tuning MobileNetV2, InceptionV3, EfficientNetB0, and ResNet50.

The proposed future analysis contains the following procedures:

1. Performing hyperparameter tuning on more hyperparameters like learning rate to further improve the confusion between classes and improve precision score for all classes.
2. Exploring more models for transfer learning and fine-tuning.
3. Creating an application for testing image classification in real-time and checking the confidence score. A user interface is beneficial in deploying the model with ease and efficiency.
4. Another potential and useful procedure is to classify the identified object as recyclable, non-recyclable, or compostable.

These steps can be beneficial in expanding the work and deploying it with the required ethical considerations. They can pave the way for a business model for this project and solve the issue of improper waste disposal with efficiency.

11. REFERENCES

[1] Environmental Protection Agency (EPA) (2025). *Circular Economy and Waste Statistics*. Available at: https://www.epa.ie/our-services/monitoring--assessment/waste/national-waste-statistics/

[2] REPAK (2025). Available at: https://repak.ie/recycling/

[3] Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J., 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, *33*(12), pp.6999-7019.

[4] Developers, T., 2022. TensorFlow. *Zenodo*.

[5] Umesh, P., 2012. Image processing in python. *CSI Communications*, *23*(2), pp.23-24.

[6] Munappy, A.R., Bosch, J. and Olsson, H.H., 2020. Data pipeline management in practice: Challenges and opportunities. In *Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21* (pp. 168-184). Springer International Publishing.

[7] Yu, Y. and Grammenos, R., 2021. Towards artificially intelligent recycling Improving image processing for waste classification. *arXiv preprint arXiv:2108.06274*.

[8] Nafiz, M.S., Das, S.S., Morol, M.K., Al Juabir, A. and Nandi, D., 2023, January. Convowaste: An automatic waste segregation machine using deep learning. In *2023 3rd International conference on robotics, electrical and signal processing techniques (ICREST)* (pp. 181-186). IEEE.

[9] Ahmed, M.I.B., Alotaibi, R.B., Al-Qahtani, R.A., Al-Qahtani, R.S., Al-Hetela, S.S., Al-Matar, K.A., Al-Saqer, N.K., Rahman, A., Saraireh, L., Youldash, M. and Krishnasamy, G., 2023. Deep learning approach to recyclable products classification: Towards sustainable waste management. *Sustainability*, *15*(14), p.11138.

[10] Chhabra, M., Sharan, B., Elbarachi, M. and Kumar, M., 2024. Intelligent waste classification approach based on improved multi-layered convolutional neural network. *Multimedia Tools and Applications*, *83*(36), pp.84095-84120.

[11] Arishi, A., 2025. Real-Time Household Waste Detection and Classification for Sustainable Recycling: A Deep Learning Approach. *Sustainability*, *17*(5), p.1902.

[12] Tribuana, D. and Arda, A.L., 2024. Image preprocessing approaches toward better learning performance with cnn. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *8*(1), pp.1-9.

[13] Wang, Z.J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M. and Chau, D.H.P., 2020. CNN explainer: learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), pp.1396-1406.

[14] Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). Performance of machine learning algorithms with different K values in K-fold cross-validation. *International Journal of Information Technology and Computer Science*, *13*(6), 61-71.

[15] Prusty, S., Patnaik, S., & Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, *4*, 972421.

[16] Liao, L., Li, H., Shang, W., & Ma, L. (2022). An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *31*(3), 1-40.

[17] Ramdan, A., Heryana, A., Arisal, A., Kusumo, R. B. S., & Pardede, H. F. (2020, November). Transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets. In *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)* (pp. 206-211). IEEE.

[18] Vrbančič, G., & Podgorelec, V. (2020). Transfer learning with adaptive fine-tuning. *IEEE Access*, *8*, 196197-196211.

[19] Gulzar, Y. (2023). Fruit image classification model based on MobileNetV2 with deep transfer learning technique. *Sustainability*, *15*(3), 1906.

[20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[21] Koonce, B. (2021). ResNet 50. In *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization* (pp. 63-72). Berkeley, CA: Apress.

[22] Sharma, A. K., Nandal, A., Dhaka, A., Zhou, L., Alhudhaif, A., Alenezi, F., & Polat, K. (2023). Brain tumor classification using the modified ResNet50 model based on transfer learning. *Biomedical Signal Processing and Control*, *86*, 105299.

[23] Ashtagi, R., Kharat, P. V., Sarmalkar, V., Hosmani, S., Patil, A. R., Akkalkot, A. I., & Padthe, A. (2024). Enhancing melanoma skin cancer diagnosis through transfer learning: An EfficientNetb0 approach. *Acadlore Transactions on AI and Machine Learning*, *3*(1), 57-69.

[24] Lin, C., Li, L., Luo, W., Wang, K. C., & Guo, J. (2019). Transfer learning based traffic sign recognition using inception-v3 model. *Periodica Polytechnica Transportation Engineering*, *47*(3), 242-250.

Images

[Image 1] Awofeso, Zaynab. (2024). *MobileNetV1 and MobilenNetV2 architecture* [Photograph] https://medium.com/codex/a-summary-of-the-mobilenetv2-inverted-residuals-and-linear-bottlenecks-paper-e19b187cb78a

[Image 2] Kaushik, Aakash. *ResNet50 Architecture* [Table] https://iq.opengenus.org/resnet50-architecture/

[Image 3] (2025). *Different Scaling Methods vs Compound Scaling* [Photograph] https://www.geeksforgeeks.org/computer-vision/efficientnet-architecture/

[Image 4] Singh, T., & Vishwakarma, D. K. (2021). *Block-diagram of Inception-v3 improved deep architecture* [Figure]. In *A deeply coupled ConvNet for human activity recognition using dynamic and RGB images*. *Neural Computing and Applications*. ResearchGate. https://www.researchgate.net/figure/Block-diagram-of-Inception-v3-improved-deep-architecture_fig3_341563435