

## Problem Statement

### Objective

In this assignment, you will develop a waste material classification system using Convolutional Neural Networks (CNNs) to automate the sorting of waste into distinct categories. This project applies deep learning techniques to enhance recycling efficiency, reduce environmental pollution, and promote sustainable waste management practices. Through this assignment, you will:

- Train and fine-tune a CNN model to classify waste such as cardboard, glass, paper, and plastic.
- Evaluate model accuracy using performance metrics such as precision, recall, and F1-score.
- Gain insights into the role of AI in sustainable environmental solutions.

### Business Value

Improper waste disposal contributes to environmental degradation, increased landfill waste, and inefficient recycling processes. Manual sorting is labour-intensive, error-prone, and costly. An AI-powered waste classification system addresses these challenges by streamlining waste segregation, reducing operational costs, and improving recycling rates.

Key business benefits include:

- Automated waste sorting, cutting time and labour
- Increased recycling rated by precise classification of recyclables

### Use Cases:

- Smart Recycling Bins
- Automated Waste Sorting Facilities
- Waste Monitoring and Reporting

## Dataset Overview

### Context

The dataset consists of images of some common waste material categories. The images are classified into seven categories, namely: Food Waste, Metal, Paper, Plastic, Other, Cardboard, and Glass.

### Content

- The dataset contains seven folders, one containing images of each category of waste.
- There are around 7000 raw images divided among the categories.
  - The dataset consists of multiple folders, each representing a specific class, such as Cardboard, Food\_Waste, and Metal.
  - Within each folder, there are images of objects that belong to that category. However, these items are not further subcategorised.
  - For instance, the Food\_Waste folder may contain images of items like coffee grounds, teabags, and fruit peels, without explicitly stating that they are actually coffee grounds or teabags.

### Acknowledgements

This dataset is free and is publicly available on Kaggle.

## Scoring and Penalty

- **Total Marks: 50** (for code notebook)
- **Extension and Penalty:** As given in your learner handbooks

## Instructions

1. Each learner should attempt this assignment individually.
2. Programming Language: Python
3. You will be provided with the dataset and a starter notebook. You have to work in the starter notebook only.
4. It is very important that you do not change any headings, subheadings, questions or tasks in your notebook as it can cause problems with grading.
5. If you find any inconsistencies and outliers, please handle them as per your understanding and mention them in your solution.
6. You are encouraged to search the web and consult AI tools for conceptual understanding. However, using plagiarized or AI-generated code is strictly prohibited and strongly discouraged.
7. Submitting plagiarized and AI-generated code or reports will result in significant penalties to your scores.

## Submission Guidelines

1. To submit your solution, push your submission to GitHub and submit the GitHub link in the submission field.
2. You are required to upload your solution as a **zip file** titled "**CNN\_Waste\_Segregation\_<your\_name>.zip**" in a public GitHub repository.
3. The repository should be named after the assignment.
4. The zipped folder that you upload should contain an Interactive Python Notebook (.ipynb) that contains your code and your visualisations, analysis, results, insights, and outcomes.
5. Note that this file should only be generated from the starter files provided to you.
6. The Jupyter notebook should contain your name and the assignment title.
7. Mention all assumptions made.

## Results Expected from Learners

In the starter notebook, you will find headings, subheadings, and checkpoints stating the tasks you need to perform. The marks associated with each checkpoint will also be mentioned in the notebook. Keep in mind not to edit the cells with marking schemes and questions. You can find a brief description of the tasks below.

### 1. Load the data

Unzip and load the data

### 2. Data Preparation [25 marks]

#### 2.1 Load and Preprocess Images [8 Marks]

- i. Create a function to load the images
- ii. Load images and labels

#### 2.2 Data Visualisation [9 Marks]

- i. Create a bar plot to display the class distribution
- ii. Visualize some sample images
- iii. Based on the smallest and largest image dimensions, resize the images

#### 2.3 Encoding the classes [3 Marks]

- i. Encode the target class labels

#### 2.4 Data Splitting [5 Marks]

- i. Split the dataset into training and validation sets

### 3. Model Building and Evaluation [20 marks]

#### 3.1 Model building and training [15 marks]

- i. Build and compile the model. Try various configurations
- ii. Train the model

#### 3.2 Model Testing and Evaluation [5 marks]

- i. Evaluate the model on test dataset. Derive appropriate metrics.

### 4. Data Augmentation [optional]

#### 4.1 Create a Data Augmentation Pipeline

### 5. Conclusion & Key Takeaways [5 Marks]

#### 5.1 Conclude with outcomes and insights gained [5 Marks]

## Evaluation Rubrics

The following rubrics will be used while judging your solutions to the above tasks.

Table 1: Rubrics

Criteria	Meets expectations	Does not meet expectations
<b>Data Loading</b>	<ol style="list-style-type: none"> <li>1. Images and labels are properly loaded. Labels are accurately assigned to images.</li> <li>2. All annotated objects in the images are fetched.</li> <li>3. Images and labels are converted to appropriate formats as required for further steps.</li> </ol>	<ol style="list-style-type: none"> <li>1. Images and labels are not properly loaded or are assigned wrong.</li> <li>2. All annotated objects in the images are not fetched/incorrectly fetched.</li> <li>3. Images and labels are not converted to appropriate formats as required for further steps.</li> </ol>
<b>Data Understanding</b>	<ol style="list-style-type: none"> <li>1. All data quality issues are correctly identified and reported.</li> <li>2. Wherever required, the meanings of the variables are correctly interpreted and written either in the comments or text.</li> </ol>	<ol style="list-style-type: none"> <li>1. Data quality issues are overlooked or are not identified correctly such as missing values, outliers and other data quality issues.</li> <li>2. The variables are interpreted incorrectly or the meaning of variables is not mentioned.</li> </ol>
<b>Data Preparation and Dataset Creation</b>	<ol style="list-style-type: none"> <li>1. Images have been resized correctly.</li> <li>2. Proper class encoding, data splitting and batch creation steps are performed.</li> </ol>	<ol style="list-style-type: none"> <li>1. Images have not been resized correctly.</li> <li>2. Encoding, data splitting or batch creation steps are missing/incorrect.</li> </ol>

Continued on next page

Table 1: Rubrics (Continued)

Criteria	Meets expectations	Does not meet expectations
<b>Model Building and Training</b>	<ol style="list-style-type: none"> <li>Created a CNN model and tested out a few different configurations</li> <li>Defined appropriate optimiser and loss function for model training.</li> <li>Trained the model for at least 10 epochs.</li> <li>Explained the findings after the model fit with explanations for model performance. Grading will be done on how well the steps of model building are followed and how well the results are explained.</li> </ol>	<ol style="list-style-type: none"> <li>Insufficient model is created or different configurations are not tried out.</li> <li>Optimiser and loss function are not suitable for the model.</li> <li>Training is insufficient or causes overfitting.</li> <li>Results are not explained. Result analysis to get a better understanding of the model working and performance better is missing</li> </ol>
<b>Solution and Outcomes</b>	<ol style="list-style-type: none"> <li>The solution has a clear structure and is not unnecessarily complex.</li> <li>The training results, whether positive or negative, are explained clearly.</li> <li>If any assumptions are made, they are stated clearly.</li> </ol>	<ol style="list-style-type: none"> <li>The solution lacks structure, is too long or does not put emphasis on the important observations.</li> <li>Training results and outcomes are not analysed and explained.</li> <li>Contains unnecessary details or lacks important ones.</li> <li>Assumptions made, if any, are not stated clearly.</li> </ol>
<b>Conciseness and readability of the code</b>	<ol style="list-style-type: none"> <li>The code is concise and syntactically correct. Wherever appropriate, built-in functions and standard libraries are used instead of writing long code (if-else statements, loops, etc.).</li> <li>Custom functions are used to perform repetitive tasks.</li> <li>The code is readable with appropriately named variables and detailed comments are written wherever necessary.</li> </ol>	<ol style="list-style-type: none"> <li>Long and complex code is used instead of shorter built-in functions.</li> <li>Custom functions are not used to perform repetitive tasks resulting in the same piece of code being repeated multiple times.</li> <li>Code readability is poor because of vaguely named variables or lack of comments wherever necessary.</li> </ol>