**Data Preparation, Feature Engineering, and Model Exploration Report**

**Title - Real-Time Recycling Sorting Using Deep Learning**

**Group no: Group 8**

**Group Members:**

1. Tekia Tekle

2. Firehiwet Zerihun

3. Biniyam Demissew

4. Daniel Alemu

5. Eyob Sisay

6. Haymanot Tamrat

# Data Preparation/Feature Engineering

## Overview

The effectiveness of a machine learning model heavily relies on the quality and preparation of the data. This project aims to classify waste images into six categories—cardboard, glass, metal, paper, plastic, and trash—by training three distinct models: a Convolutional Neural Network (CNN), a Vision Transformer (ViT), and YOLOv5. We conducted thorough data preprocessing, feature engineering, and exploratory data analysis (EDA) on the TrashNet dataset to improve model robustness, interpretability, and performance.

## Data Collection

* **Source**: The TrashNet dataset was collected from **Kaggle**.
* **Classes**: cardboard, glass, metal, paper, plastic, trash
* **Storage & Access**: The dataset was uploaded to **Google Drive** and accessed via **Google Colab**.
* **Input Format**: Images were organized in subfolders named by their class labels.
* **Image Sizes**:
  + CNN & ViT: resized to **224×224**
  + YOLOv5: resized to **640×640**

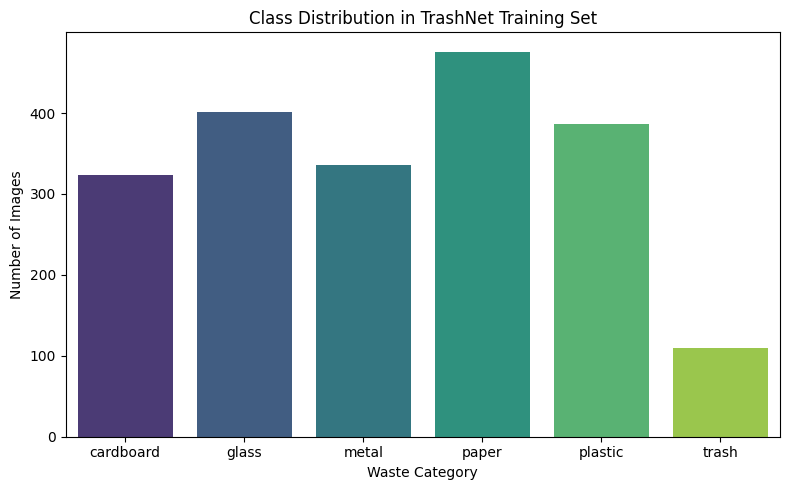
## ****Data Cleaning****

* The dataset was **inspected manually** and via code for:
  + Missing or corrupted files
  + Mis labelled or inconsistent folders
* No nulls or missing labels were found.
* **Normalization**: Pixel values were scaled to the [0, 1] range.
* **Duplicates**: Any visually identical or exact duplicate images were removed.
* Folder structures were verified to align with expected class indices.

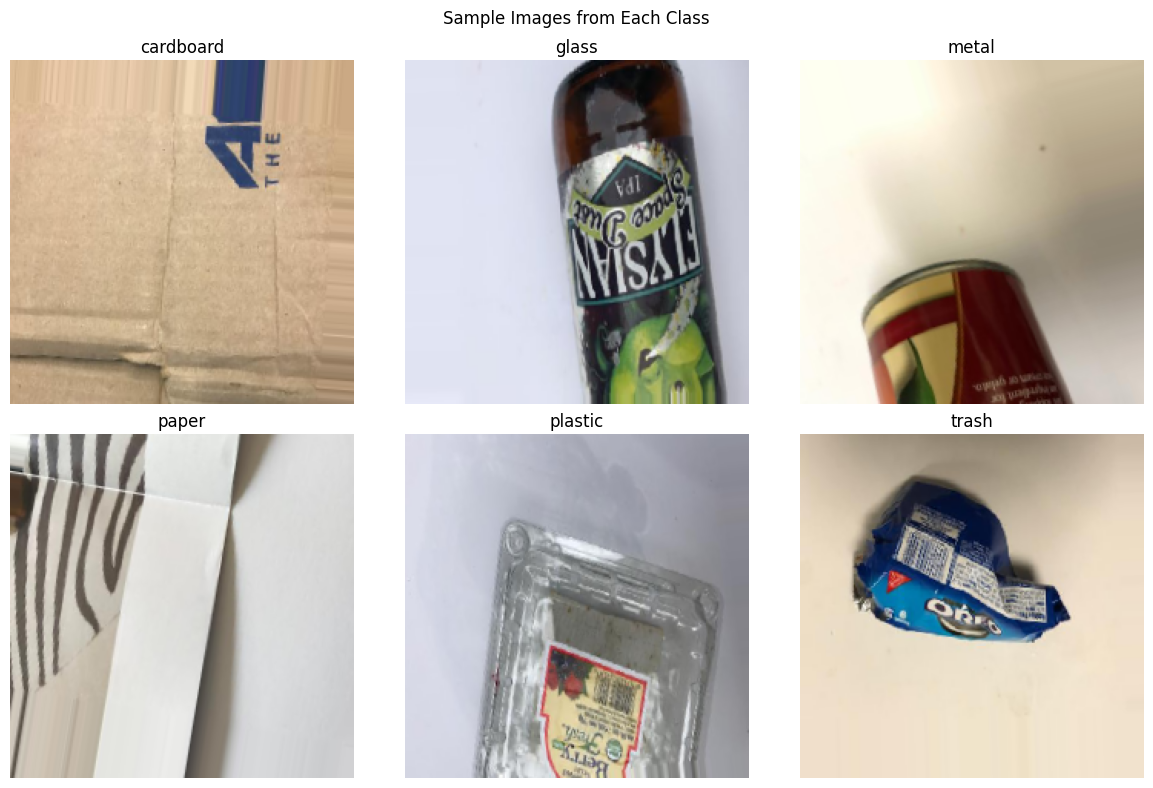
## ****Exploratory Data Analysis (EDA)****

### Class Distribution

* A **bar chart** was plotted to show image counts per class.
* Most classes were **well balanced**, with minor variations.
* This helped us justify the use of **data augmentation** to further balance training data.



### Sample Image Visualization

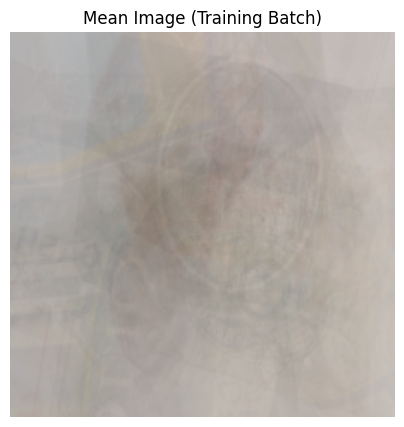
* Random sample images were displayed for each class.
* This confirmed:
  + Image clarity
  + Good class separability
  + Reasonable variability in lighting and background

### Image Shape and Pixel Range

* Sample image dimensions: **(224, 224, 3)**
* Pixel value range: **[0.0, 1.0]** after normalization

### Mean Image Visualization

* A “mean image” was calculated by averaging pixel values over a batch.
* This revealed consistent object centering and justified further augmentations like rotation and shifting.



## ****Feature Engineering****

To enhance the dataset and improve model robustness:

* **CNN/ViT augmentations**:
  + Random rotations (±20°)
  + Horizontal flips
  + Width/height shifts
  + Zoom and shear transformations
* **YOLOv5 augmentations**:
  + Built-in **mosaic augmentation**
  + Color distortions and auto-anchor updates
* **Label processing**:
  + CNN/ViT: labels were **one-hot encoded**
  + YOLO: bounding box labels were converted to YOLO format .txt files

## ****Data Transformation****

* **Rescaling**: All image pixels scaled to [0, 1] using rescale=1./255
* **Resizing**:
  + CNN/ViT → 224x224
  + YOLOv5 → 640x640

**Label encoding**:

* CNN/ViT → one-hot vectors
* YOLO → bounding box center x, y, w, h normalized

# Model Exploration

## ****Model Selection****

|  |  |  |
| --- | --- | --- |
| Model | Purpose | Strengths |
| CNN | Classification | Lightweight and fast |
| ViT | Classification | Strong for global spatial patterns |
| YOLOv5 | Detection | Real-time object localization |

## ****Model Training****

* **CNN**: Trained for 10 epochs with Adam optimizer and categorical\_crossentropy loss.
* **ViT**: Fine-tuned for 5 epochs with a small learning rate (3e-5) using TFViT For Image Classification .
* **YOLOv5**: Trained for 10 epochs using Ultralytics' train.py script with default hyperparameters and the yolov5s.pt backbone.

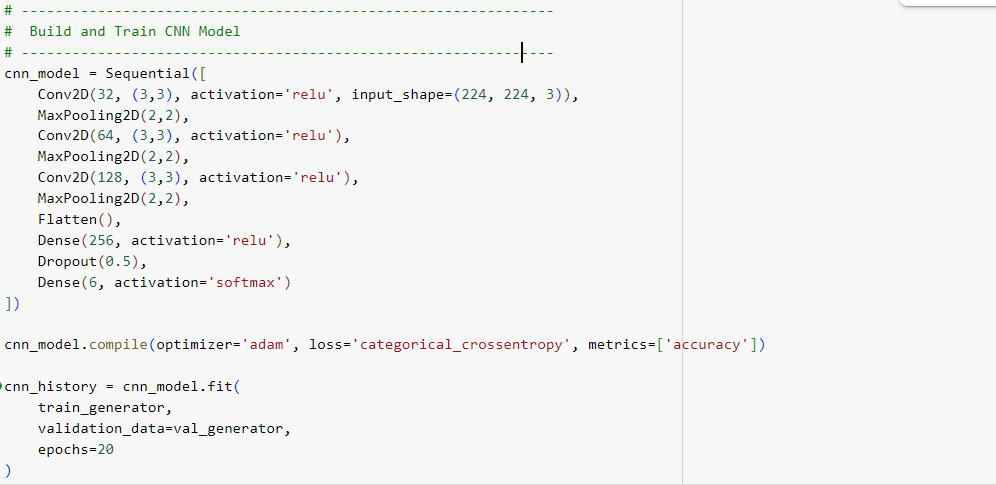
All models were trained in **Google Colab with GPU acceleration**.

## Model Evaluation

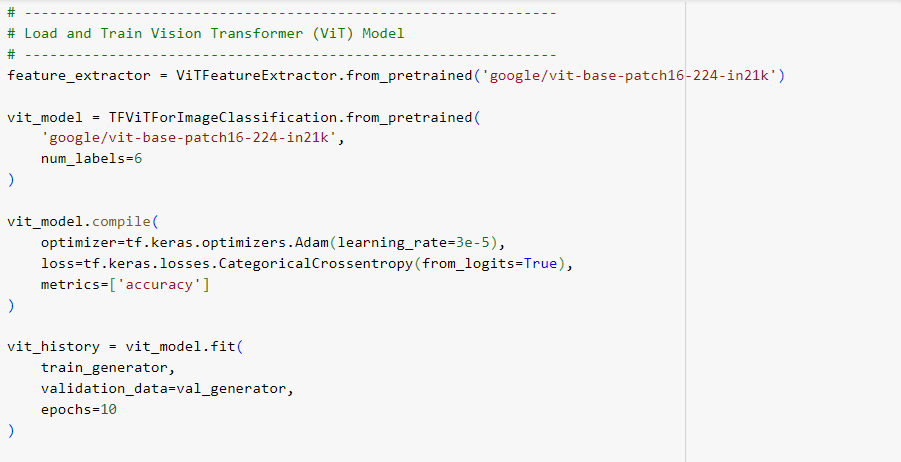
Evaluation metrics included validation accuracy, validation loss, confusion matrices (for CNN and ViT), and mAP (mean Average Precision) for YOLOv5.  
Confusion matrices for CNN and ViT showed most confusion between similar materials (e.g., plastic vs. glass), while YOLOv5 detected waste regions more precisely.

## Code Implementation

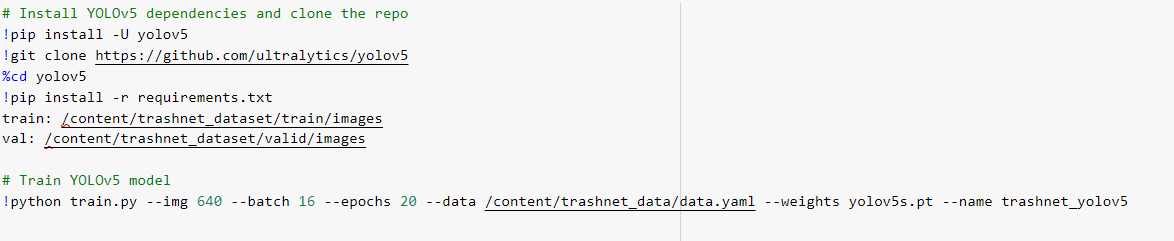
Here is a sample CNN code block:



Here is a sample VIT code block:



Here is a sample YOLO5 code block:



## ****Conclusion****

This project demonstrates the power and flexibility of modern deep learning techniques in automating waste classification, a task critical for smart city sustainability and environmental responsibility. Through careful data preparation, feature engineering, and model selection, we developed and evaluated three models: a traditional Convolutional Neural Network (CNN), a state-of-the-art Vision Transformer (ViT), and the real-time object detection model YOLOv5.

The **TrashNet dataset** was thoroughly analyzed and cleaned, revealing a balanced and diverse set of images suitable for robust training. Exploratory Data Analysis (EDA) provided crucial insights into class distribution, variability, and input quality, which directly informed our augmentation strategies. Data normalization and encoding steps ensured consistency across models.