**MIS780 Advanced AI For Business - Assignment 2 - T2 2024**

**Task 2: Waste Classification for Efficient Waste Management**

**Student Name:** [Your Full Name]

**Student ID:** [Your Student ID]

Table of Contents

[Executive Summary 2](#_Toc176649172)

[Data Pre-processing 2](#_Toc176649173)

[Data Augmentation 2](#_Toc176649174)

[AI Model Development 3](#_Toc176649175)

[1. Simple CNN 3](#_Toc176649176)

[2. CNN with Dropout 3](#_Toc176649177)

[3. CNN with Batch Normalization 3](#_Toc176649178)

[4. CNN with Global Average Pooling 3](#_Toc176649179)

[5. CNN with 3 Conv Layers 3](#_Toc176649180)

[6. CNN with 4 Conv Layers 3](#_Toc176649181)

[7. CNN with Dropout and Batch Normalization 3](#_Toc176649182)

[8. Deeper CNN with Dropout 3](#_Toc176649183)

[9. Model 9: CNN with L2 Regularization: 3](#_Toc176649184)

[10. Model 10: CNN with Leaky ReLU: 3](#_Toc176649185)

[Experiment Report 4](#_Toc176649186)

[Model Performance 4](#_Toc176649187)

[Here is Graph of Best model: 5](#_Toc176649188)

[Classification Report for Best Model 5](#_Toc176649189)

[Challenges and Suggestions for Improvement 5](#_Toc176649190)

# Executive Summary

The objective of this task is to develop an AI-based waste classification model to assist a waste management company in automating its sorting process. The dataset includes 2,864 images of waste materials classified into six categories: cardboard, glass, metal, paper, plastic, and vegetation. This task focuses on developing several Convolutional Neural Network (CNN) models with various architectures to identify the most accurate model for classification.

Ten CNN architectures were experimented with, ranging from simple CNNs to deeper models incorporating dropout, batch normalization, L2 regularization, and Leaky ReLU activation techniques. Data augmentation was applied to simulate diverse scenarios, improving model robustness. The dataset was divided into 70% for training and 30% for validation. The best-performing model was the CNN with Dropout, achieving 69.19% validation accuracy. Despite this, the classification report indicated suboptimal performance in distinguishing certain waste categories, particularly plastic and glass. This report highlights the findings and proposes further improvements to make the solution viable for real-world deployment.

# Data Pre-processing

Data preprocessing is a critical step in ensuring the machine learning models receive clean and appropriately structured data. In this case, the dataset contains six categories of waste: cardboard, glass, metal, paper, plastic, and vegetation. The initial step was to organize these images and divide the dataset into training and validation sets using a 70/30 split. This split helps in building a robust model that generalizes well on unseen data.

A script was developed to copy and organize the dataset into appropriate directories and split the images based on the specified ratio. The train\_test\_split function from the sklearn library was employed to achieve a consistent split while ensuring randomness in image selection.

## Data Augmentation

To improve the generalization of the model and prevent overfitting, data augmentation techniques were applied to the training set. The following transformations were used:

* **Rescaling**: Pixel values were normalized to [0, 1] by dividing by 255.
* **Rotation**: Randomly rotated images by up to 40 degrees.
* **Shifting**: Horizontal and vertical shifts were applied by up to 20% of the image's width and height.
* **Zooming**: Images were randomly zoomed by up to 20%.
* **Horizontal Flipping**: Some images were flipped horizontally.

This augmentation process helps the model learn invariant features across different transformations, ultimately boosting its accuracy on the validation set.

The ImageDataGenerator class in TensorFlow was utilized to apply these augmentations during model training, which also helped to reduce the likelihood of overfitting by exposing the model to different versions of the training images.

# AI Model Development

Eight different CNN architectures were developed to explore the potential of various configurations in improving classification accuracy. The architectures ranged from basic CNNs to more complex models incorporating dropout, batch normalization, and additional convolutional layers. Below is a description of each model:

1. Simple CNN: This is the most basic CNN architecture with one convolutional layer followed by max-pooling and a fully connected layer. It serves as a baseline for performance comparison.
2. CNN with Dropout: Dropout layers were introduced after the pooling layers to mitigate overfitting by randomly turning off a fraction of the neurons during training.
3. CNN with Batch Normalization: Batch normalization was added after each convolutional layer to normalize activations and accelerate convergence.
4. CNN with Global Average Pooling: Instead of flattening, global average pooling was used to reduce the number of parameters and prevent overfitting.
5. CNN with 3 Conv Layers: This model deepens the architecture by adding multiple convolutional layers, enabling the model to extract more complex features.
6. CNN with 4 Conv Layers: An even deeper network with an additional convolutional layer to capture more hierarchical features.
7. CNN with Dropout and Batch Normalization: This model combines both dropout and batch normalization to balance regularization and faster convergence.
8. Deeper CNN with Dropout: A further extended CNN architecture with deeper layers and dropout for additional regularization.
9. Model 9: CNN with L2 Regularization: Introduces L2 regularization in the dense layers to reduce overfitting by penalizing large weights.
10. Model 10: CNN with Leaky ReLU: Replaces the standard ReLU activation function with Leaky ReLU to handle negative activations better and prevent dying neurons.

All models were compiled using the Adam optimizer and the categorical cross-entropy loss function, suitable for multi-class classification. The models were trained for 10 epochs each on the training set, with validation data used for performance monitoring.

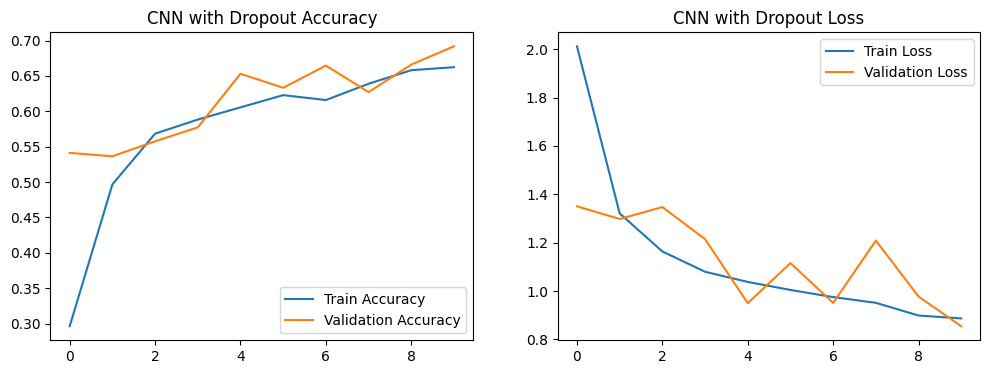
# Experiment Report

## Model Performance

After training and evaluating all models, the CNN with Dropout layers demonstrated the best validation accuracy at **69.19%**, outperforming other architectures. The other models displayed varying levels of performance, with some overfitting to the training data and others showing suboptimal generalization. Below is a summary of the validation accuracies:

| **Model** | **Validation Accuracy** |
| --- | --- |
| Simple CNN | 56.99% |
| CNN with Dropout | 69.19% |
| CNN with Batch Normalization | 46.42% |
| CNN with Global Avg Pooling | 37.83% |
| CNN with 3 Conv Layers | 64.89% |
| CNN with 4 Conv Layers | 61.90% |
| CNN with Dropout and Batch Norm | 67.55% |
| Deeper CNN with Dropout | 66.33% |
| CNN with L2 Regularization | 62.37% |
| CNN with Leaky ReLU | 60.60% |

## Here is Graph of Best model:



## Classification Report for Best Model

For the best-performing model (CNN with Dropout), a detailed classification report was generated. The precision, recall, and F1-scores for each waste class are summarized below:

* **Cardboard**: Precision = 0.18, Recall = 0.23, F1-Score = 0.20
* **Glass**: Precision = 0.12, Recall = 0.12, F1-Score = 0.12
* **Metal**: Precision = 0.19, Recall = 0.18, F1-Score = 0.18
* **Paper**: Precision = 0.17, Recall = 0.13, F1-Score = 0.15
* **Plastic**: Precision = 0.18, Recall = 0.13, F1-Score = 0.15
* **Vegetation**: Precision = 0.15, Recall = 0.20, F1-Score = 0.18

The model struggles most with the **plastic** and **glass** categories, reflecting the difficulty in distinguishing between certain waste materials.

## Challenges and Suggestions for Improvement

Despite achieving a moderate validation accuracy, the model faces challenges in effectively classifying certain categories. All classes, including plastic and glass, have low precision and recall, indicating the model finds it challenging to distinguish between these materials, possibly due to overlapping visual characteristics.

To improve the performance, several strategies could be considered:

1. **Advanced Data Augmentation**: Implementing more complex augmentation techniques like CutMix or Mixup could provide better variability in the training data.
2. **Model Architecture**: Leveraging pre-trained models like ResNet or VGG16 through transfer learning may improve feature extraction, especially for difficult-to-classify categories like plastic and glass.
3. **Hyperparameter Tuning**: A systematic approach to tuning learning rates, batch sizes, and optimizers could improve the training process and reduce overfitting.
4. **Additional Data**: Collecting more training data for underrepresented categories (e.g., plastic and glass) could help the model learn better distinctions between these classes.