

Data Collection and Preprocessing Phase

Date	1 December 2024
Team ID	739673
Project Title	Garbage Classification Using Deep Learning
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	Overview of the dataset used for garbage classification, including categories number of samples, and source of data.
Resizing	Resize images to a standard size (e.g., 224x224 pixels) to ensure uniform input for the model.
Normalization	Normalize pixel values to a range of 0 to 1 or -1 to 1 to improve model performance.
Data Augmentation	. Apply techniques such as flipping, rotation, zooming, and shifting to increase the diversity of the dataset and prevent overfitting.
Batch Normalization	Batch normalization is a technique used to improve the training of deep neural networks by normalizing the inputs of each layer.
Data Preprocessing Code Screenshots	

Loading Data	<pre># Define paths data_dir = r"C:\Users\soumy\OneDrive\Desktop\Garbage Classification\Dataset"</pre>
Resizing	<pre># Training data generator train_generator = datagen.flow_from_directory(data_dir, target_size=(128, 128), batch_size=32, class_mode='categorical', subset='training') # Validation data generator validation_generator = datagen.flow_from_directory(data_dir, target_size=(128, 128), batch_size=32, class_mode='categorical', subset='validation')</pre>
Normalization	<pre># Initialize ImageDataGenerator with split for training and validation datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2) # 80%</pre>
Data Augmentation	<pre># Data Augmentation (Optional - you can enable it if needed) augment_datagen = ImageDataGenerator(rescale=1./255, rotation_range=20, # Random rotations width_shift_range=0.2, # Horizontal shifts height_shift_range=0.2, # Vertical shifts shear_range=0.2, # Shearing zoom_range=0.2, # Zooming horizontal_flip=True, # Horizontal flipping fill_mode='nearest', # Filling in missing pixels validation_split=0.2)</pre>

Batch Normalization

```
# Applying Batch Normalization
x = Dense(256, activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x) # Batch normalization after dense layer
x = Dropout(0.5)(x) # Dropout to prevent overfitting

x = Dense(128, activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x) # Batch normalization again
x = Dropout(0.5)(x)
```