Data Collection and Preprocessing Phase

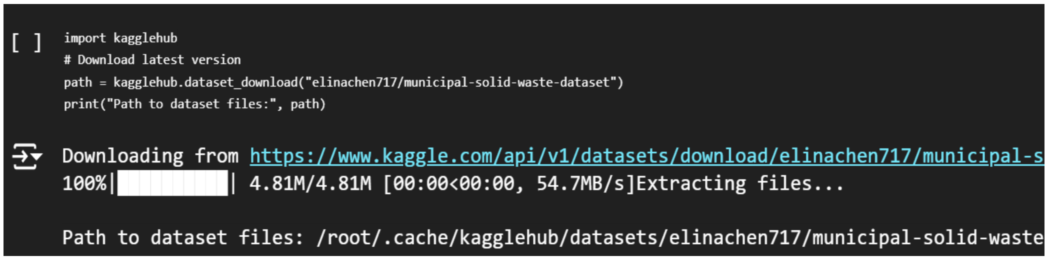
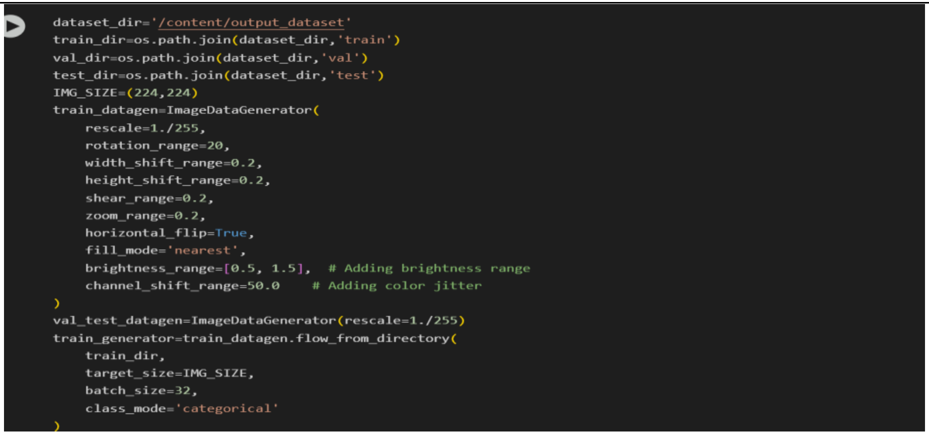
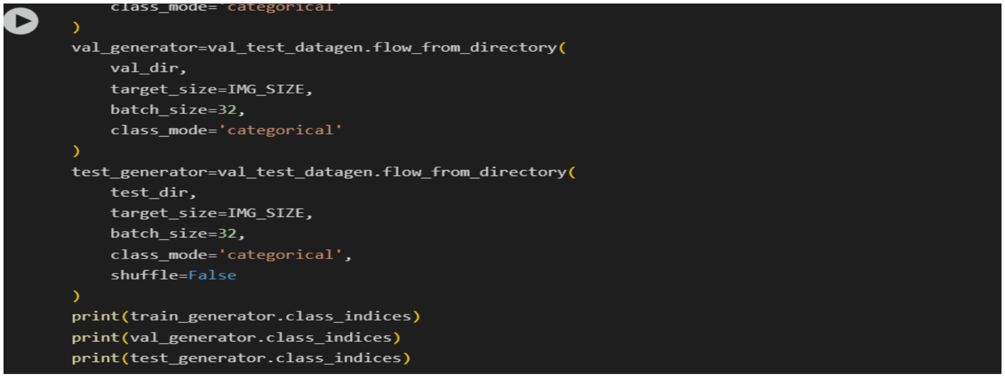
Date	15 March 2024
Team ID	739675
Project Title	Cleantech: Transforming Waste Management With Transfer Learning
Maximum Marks	6 Marks

Preprocessing

The proposed cleantech solution for transforming waste management leverages transfer learning to enhance waste classification accuracy and efficiency. By fine-tuning pre-trained convolutional neural networks (CNNs) such as ResNet50, DenseNet121, and MobileNetV2, the system can accurately categorize various waste types—including plastic, metal, paper, and organic materials—using limited labeled data .This approach significantly reduces the need for extensive datasets and computational resources, facilitating rapid deployment in diverse environments.

Section	**Description**
Data Overview	The dataset is sourced from **Kaggle**, consisting of waste management images including different wasteges
Resizing	Images are resized to a target size of **224x224 pixels**
Normalization	Pixel values are normalized to the range **[0, 1]** by dividing by **255**.
Data Augmentation	Augmentation techniques like **shearing**, **zooming**, and **horizontal flipping** are applied to enhance the dataset.

Data Preprocessing Code Screenshots

Loading Data	
Resizing , Normalisation, Augmentation	 

```
[ ]  import kagglehub
     # Download latest version
     path = kagglehub.dataset_download("elinachen717/municipal-solid-waste-dataset")
     print("Path to dataset files:", path)

     Downloading from https://www.kaggle.com/api/v1/datasets/download/elinachen717/municipal-s
     100%|██████████| 4.81M/4.81M [00:00<00:00, 54.7MB/s]Extracting files...

     Path to dataset files: /root/.cache/kagglehub/datasets/elinachen717/municipal-solid-waste
```

```
dataset_dir='/content/output_dataset'
train_dir=os.path.join(dataset_dir,'train')
val_dir=os.path.join(dataset_dir,'val')
test_dir=os.path.join(dataset_dir,'test')
IMG_SIZE=(224,224)
train_datagen=ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    brightness_range=[0.5, 1.5],  # Adding brightness range
    channel_shift_range=50.0    # Adding color jitter
)
val_test_datagen=ImageDataGenerator(rescale=1./255)
train_generator=train_datagen.flow_from_directory(
    train_dir,
    target_size=IMG_SIZE,
    batch_size=32,
    class_mode='categorical'
)
```

```
    class_mode='categorical'
)
val_generator=val_test_datagen.flow_from_directory(
    val_dir,
    target_size=IMG_SIZE,
    batch_size=32,
    class_mode='categorical'
)
test_generator=val_test_datagen.flow_from_directory(
    test_dir,
    target_size=IMG_SIZE,
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)
print(train_generator.class_indices)
print(val_generator.class_indices)
print(test_generator.class_indices)
```