



**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**LAB MANUAL**

**Unit V – Deep Learning**

**Unit V – Deep Learning**

**Lab 3. To develop a CNN model to classify images of Organic or Recyclable**

**Objective**

* Collect and preprocess a dataset of labeled images categorized as "Organic" and "Recyclable."
* Design a CNN architecture tailored to extract features from waste images, including spatial patterns and textures.
* Train the CNN model on the preprocessed dataset using a training-validation split.
* Evaluate the effectiveness of the model in reducing manual waste segregation efforts.
* Demonstrate how the model can support recycling initiatives and improve environmental sustainability.

**Problem**

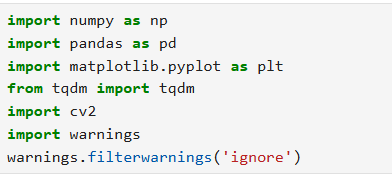
Develop a Convolutional Neural Network (CNN) model to classify images into two categories: 'Organic' and 'Recyclable.' The system will analyze and categorize waste images, aiding in efficient waste management and promoting sustainability practices. **Solution**

we'll go through the following steps:

1. Import required libraries
2. Load Dataset
3. Collections Count
4. Show data
5. Scale the data for better performance
6. Find number of classes in Data
7. Build the CNN Model
   1. Initialize the model
   2. Convolution, Activation and Pooling layers
   3. Flatten, Dropout and output layers
   4. Compile the model
8. Model Summary
9. Normalization
10. Train and Test data generation
11. Train the Model
12. Plot Training Accuracy
13. Plot Loss
14. Model Prediction

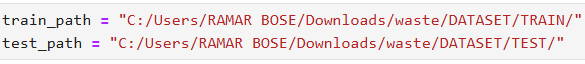
**Procedures**

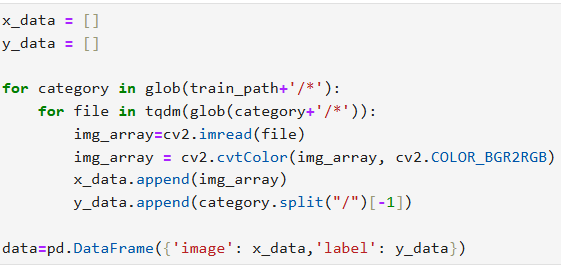
1. Import required libraries





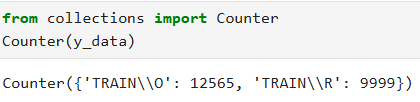
1. Load Dataset

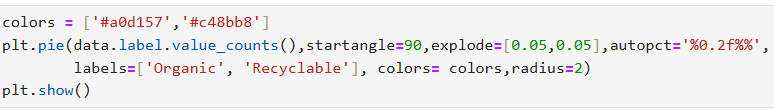


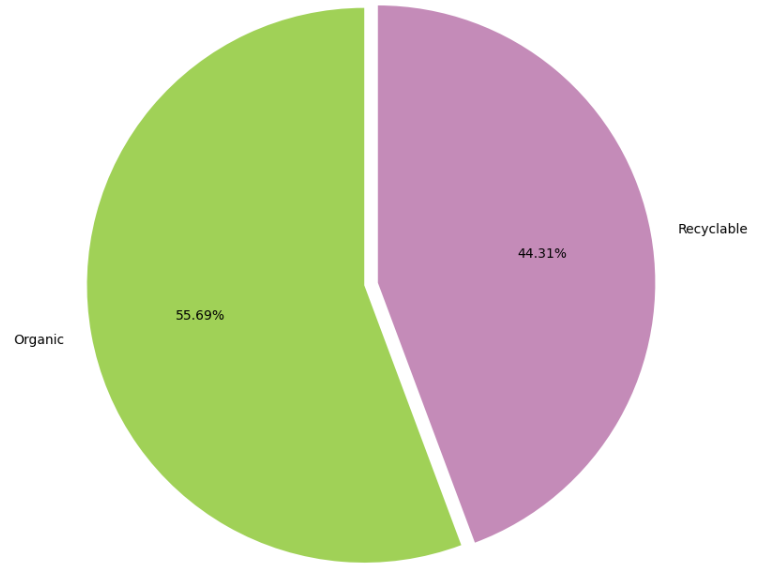




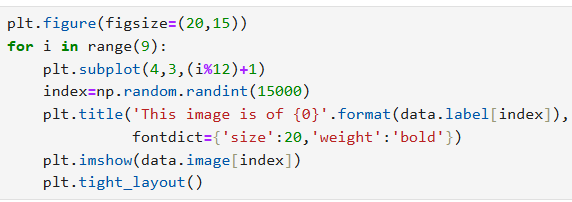
1. Collections Count

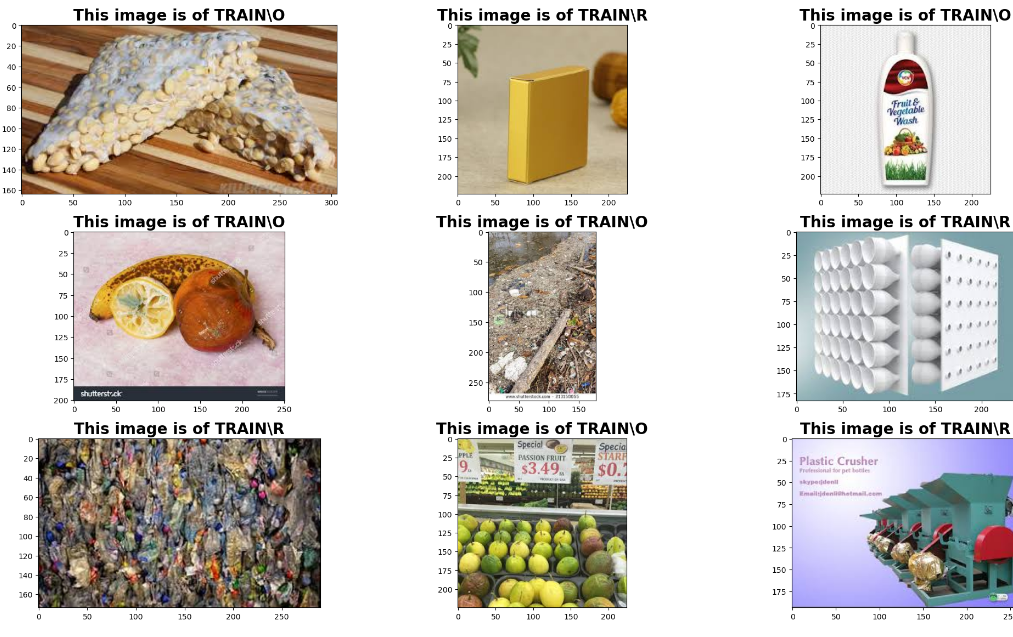




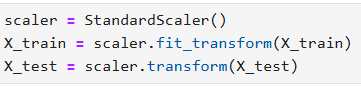


1. Show data

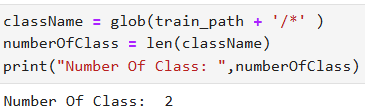




1. Scale the data for better performance



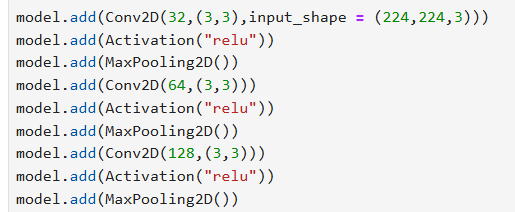
1. Find number of classes in Data



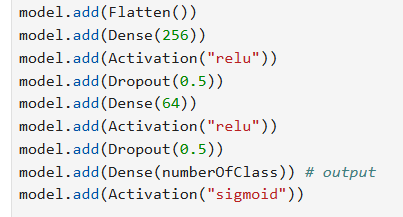
1. Build the CNN Model
   1. Initialize the model



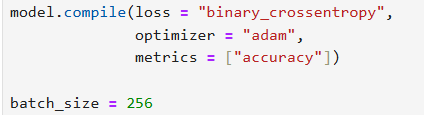
* 1. Convolution, Activation and Pooling layers



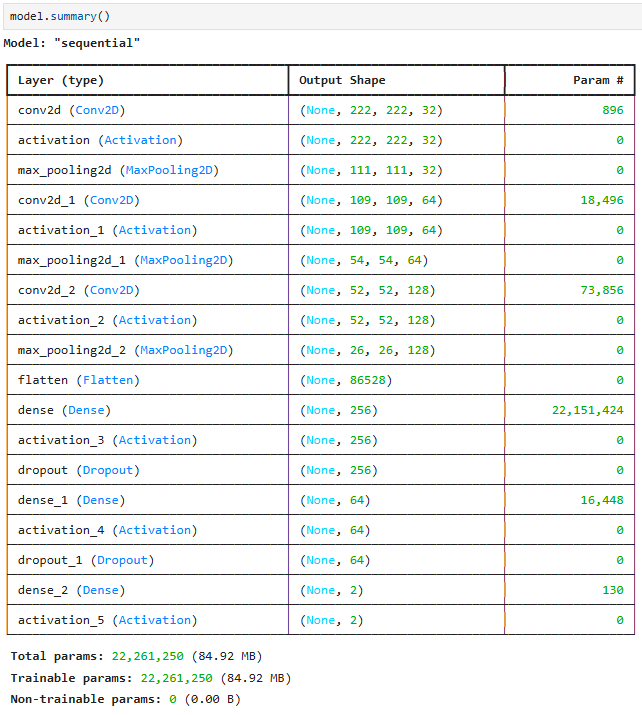
* 1. Flatten, Dropout and output layers



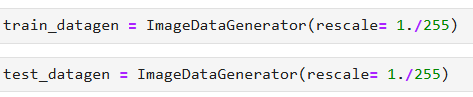
* 1. Compile the model



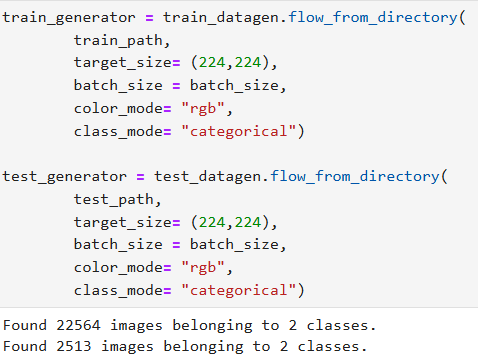
1. Model Summary



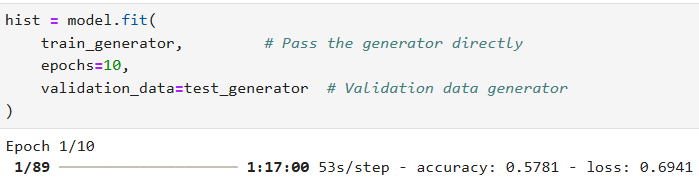
1. Normalization



1. Train and Test data generation

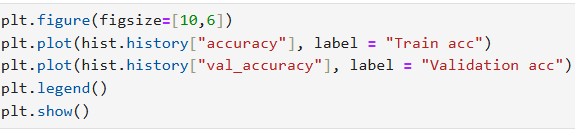


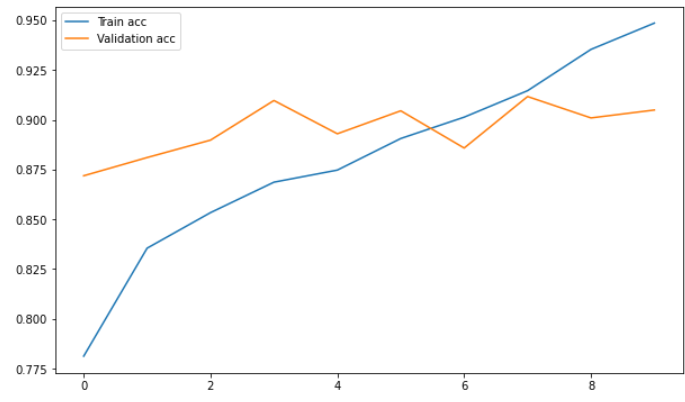
1. Train the Model



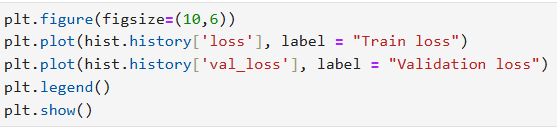
Running until 10th epoch.

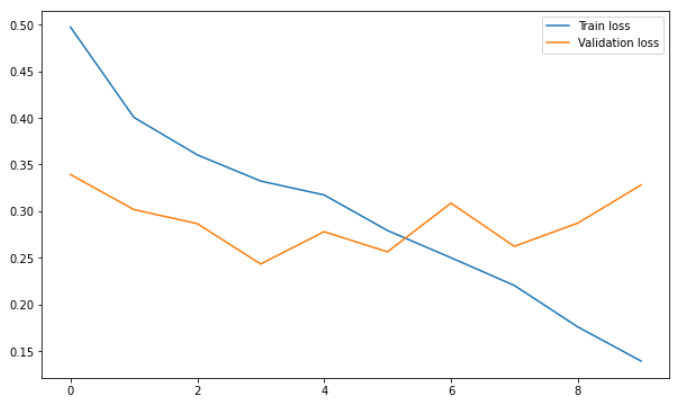
1. Plot Training Accuracy





1. Plot Loss





1. Model Prediction

