

# **RECYCLABLE WASTE CLASSIFICATION USING DEEP LEARNING**



**A DESIGN PROJECT REPORT**

**Submitted by**

**AGNES MARY LAVANYA A**

**BRINDHA G**

**SAHANA SRI D**

**VINODHA R**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**in**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

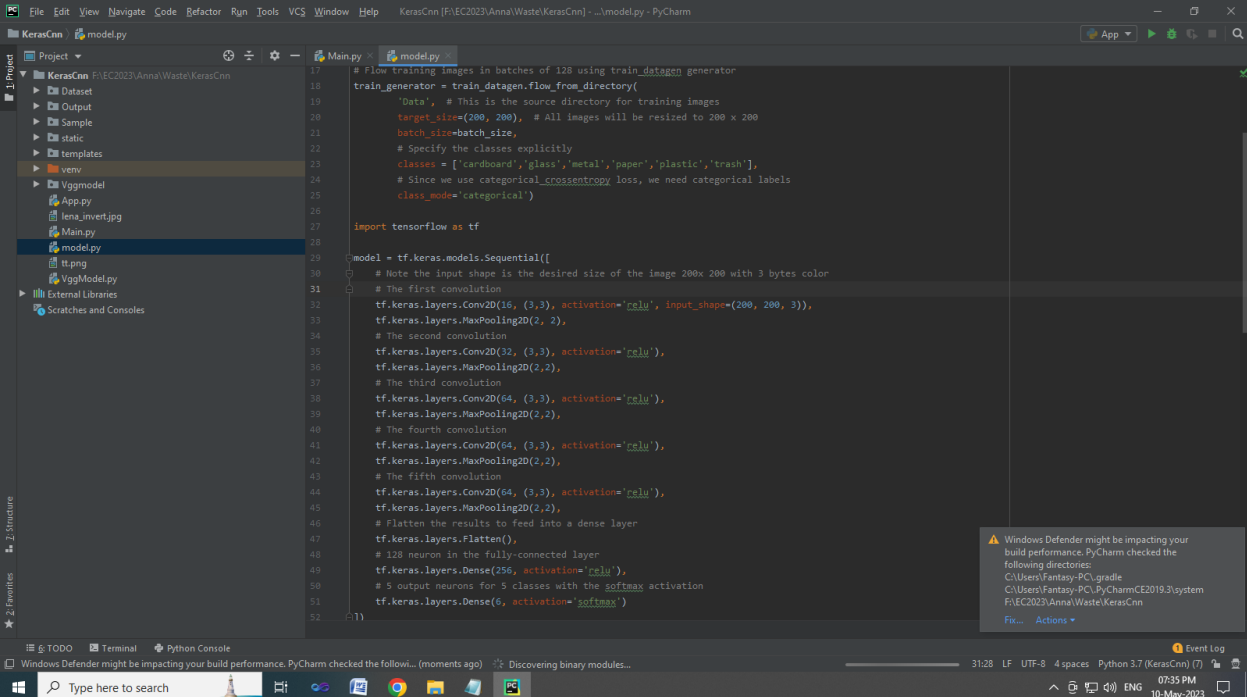
**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM-621112**

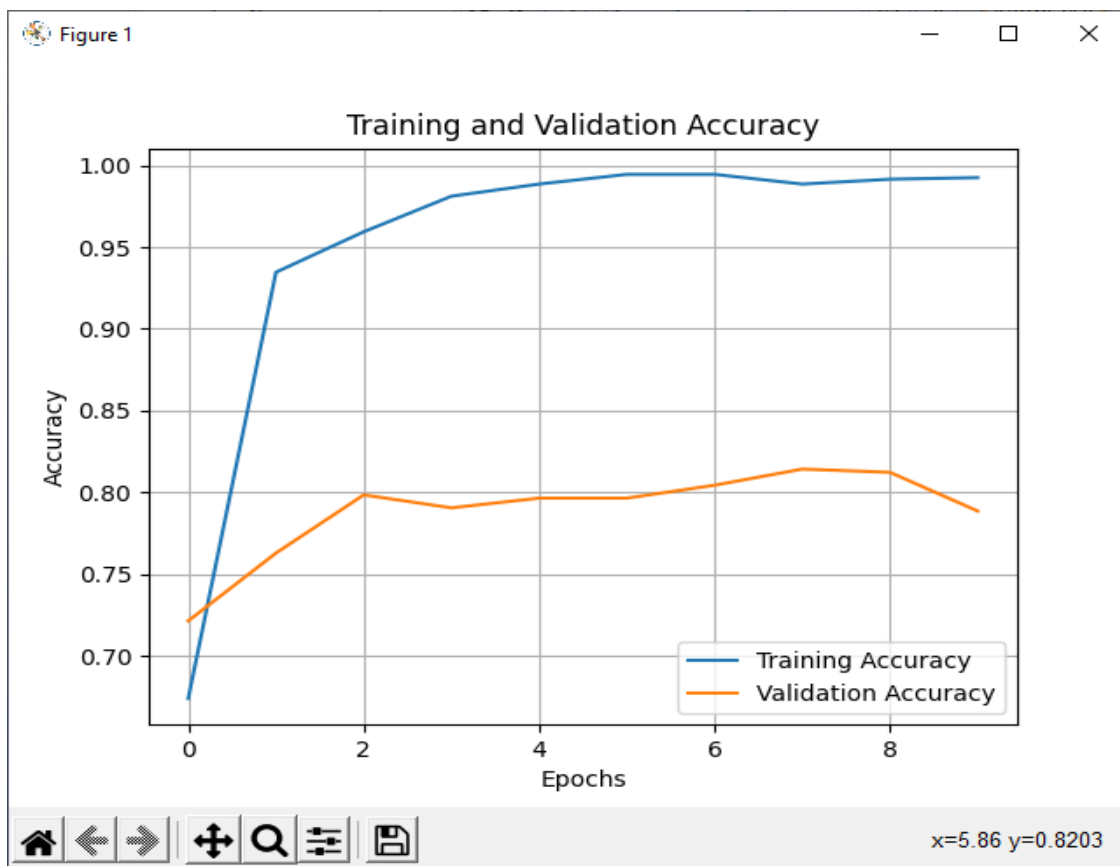
**NOVEMBER 2024**

## OUTPUT SCREENSHOT:

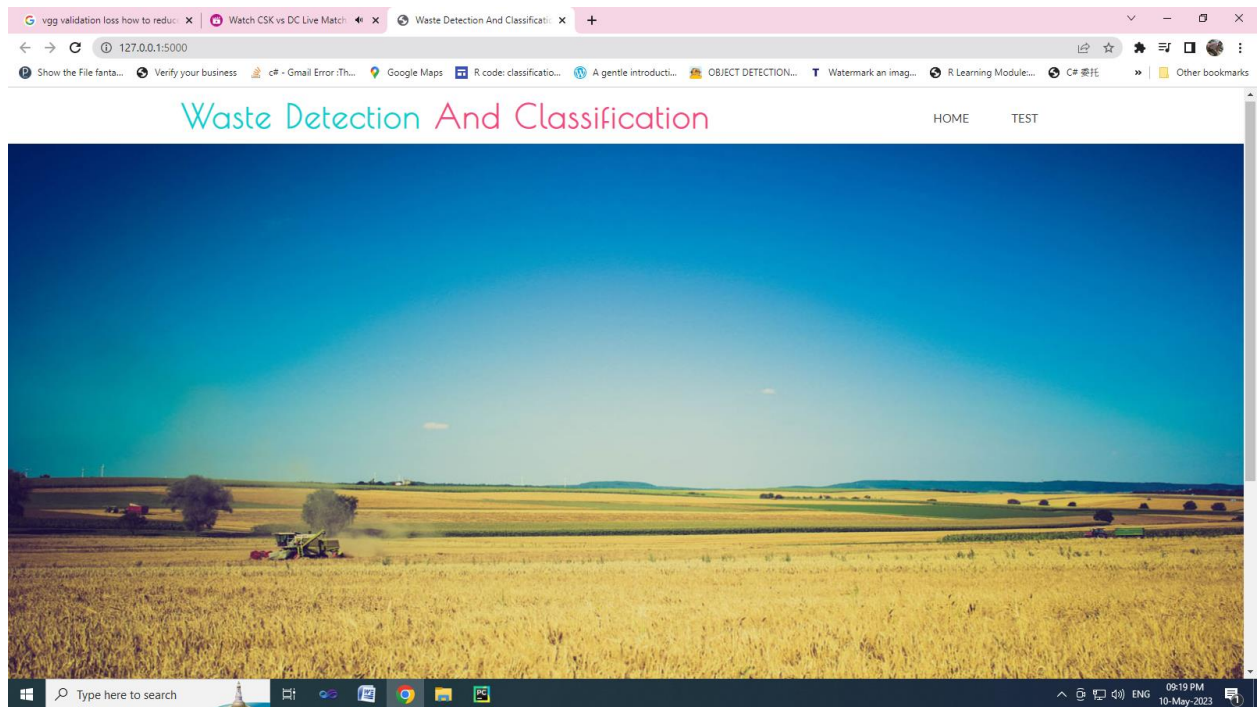


```
17 # Flow training images in batches of 128 using train_datagen generator
18 train_generator = train_datagen.flow_from_directory(
19     'Data', # This is the source directory for training images
20     target_size=(200, 200), # All images will be resized to 200 x 200
21     batch_size=batch_size,
22     # Specify the classes explicitly
23     classes = ['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash'],
24     # Since we use categorical_crossentropy loss, we need categorical labels
25     class_mode='categorical')
26
27 import tensorflow as tf
28
29 model = tf.keras.models.Sequential([
30     # Note the input shape is the desired size of the image 200x 200 with 3 bytes color
31     # The first convolution
32     tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200, 200, 3)),
33     tf.keras.layers.MaxPooling2D(2, 2),
34     # The second convolution
35     tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
36     tf.keras.layers.MaxPooling2D(2, 2),
37     # The third convolution
38     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
39     tf.keras.layers.MaxPooling2D(2, 2),
40     # The fourth convolution
41     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
42     tf.keras.layers.MaxPooling2D(2, 2),
43     # The fifth convolution
44     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
45     tf.keras.layers.MaxPooling2D(2, 2),
46     # Flatten the results to feed into a dense layer
47     tf.keras.layers.Flatten(),
48     # 128 neuron in the fully-connected layer
49     tf.keras.layers.Dense(128, activation='relu'),
50     # 5 output neurons for 5 classes with the softmax activation
51     tf.keras.layers.Dense(5, activation='softmax')
52 ])
```

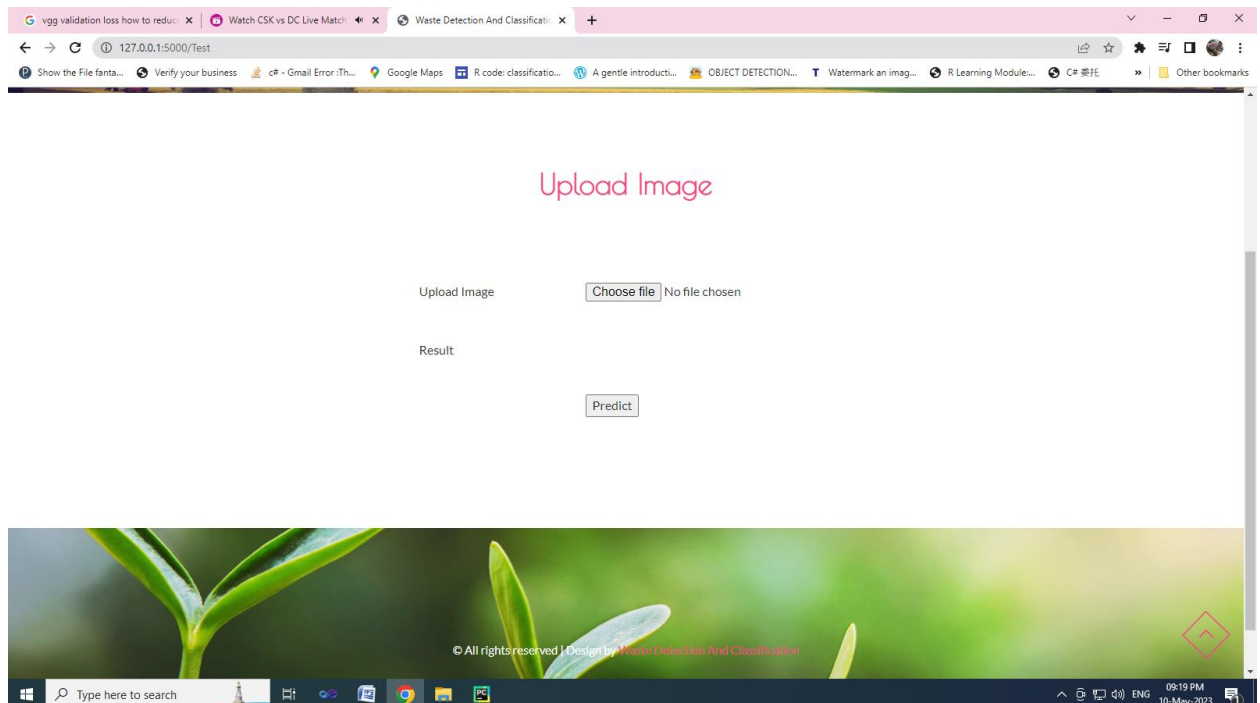
## TRAINING ACCURACY:



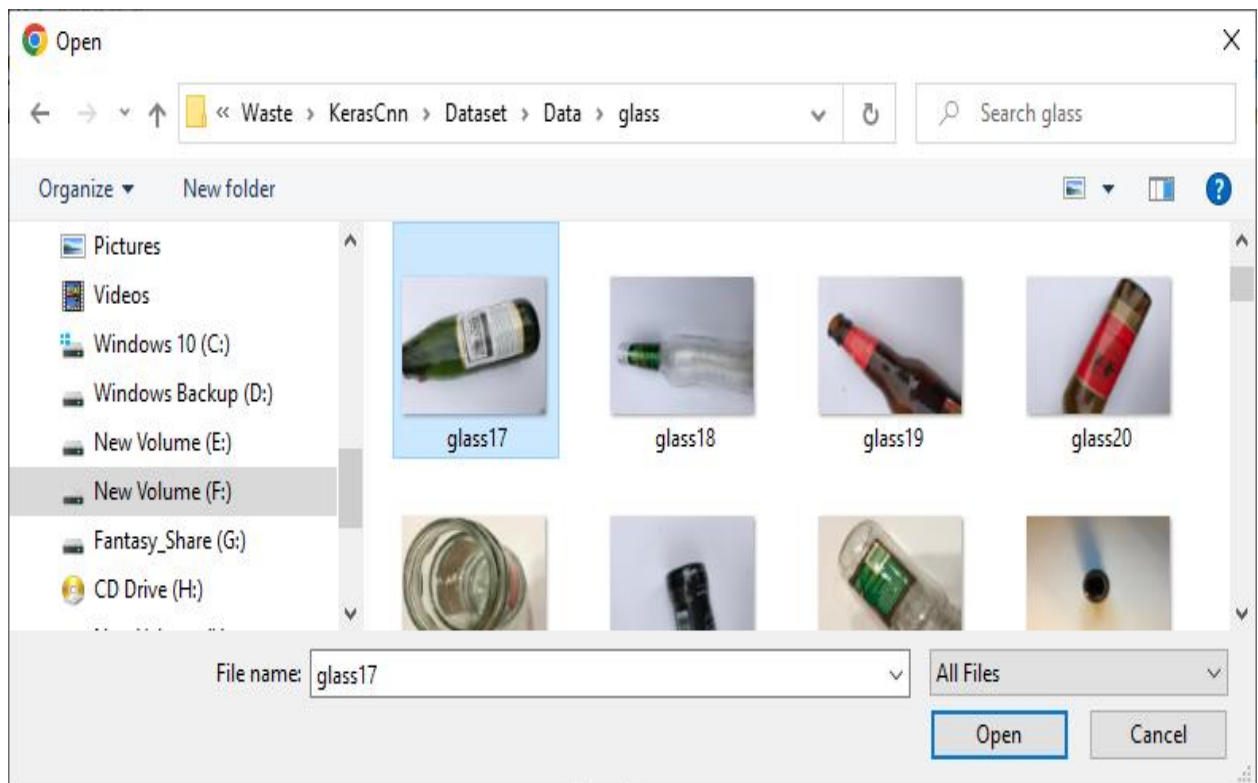
## HOME PAGE:



## IMAGE UPLOAD:



## IMAGE SELECTION:



## CLASSIFICATION RESULTS:

