**UIT2521 – INFORMATION THEORY AND APPLICATIONS**

**ASSIGNMENT – 2**

**NAME :VASANTHRAJ**

**CLASS :** IT-B

**ROLL NUMBER** : 3122215002309

**DATE :** 25.11.2023

**1. Use the test1.rar file. Upon unzipping you’ll find 10 different classes of images. There are totally 100 images.**

**(i) Give the entropy estimate of each class. You’ll need to write a code.**

**(ii) You use the entropy for the classification. Keep some figures for testing and give the testing accuracy.**

**PROGRAM:**

import os

import cv2

import numpy as np

from collections import Counter

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

def calculate\_entropy(labels):

    n = len(labels)

    counter = Counter(labels)

    entropy = 0.0

    for count in counter.values():

        probability = count / n

        entropy -= probability \* np.log2(probability) if probability > 0 else 0

    return entropy

# Replace 'path/to/your/folder' with the actual path to your image folder

folder\_path = r'images'

# List all files in the folder

file\_list = os.listdir(folder\_path)

# Assuming your images are in a common image format like JPEG or PNG

image\_extensions = ['.jpg', '.jpeg', '.png']

# Filter the list to include only image files

image\_files = [file for file in file\_list if any(file.lower().endswith(ext) for ext in image\_extensions)]

# Load and process each image

images = []

labels = []

for image\_file in image\_files:

    # Construct the full path to the image file

    image\_path = os.path.join(folder\_path, image\_file)

    # Use OpenCV to read the image

    image = cv2.imread(image\_path)

    # Process the image as needed

    # (You can add your image processing code here)

    # For example, let's calculate the average pixel value as a simple feature

    average\_pixel\_value = np.mean(image)

    # Append the processed image to the list

    images.append([average\_pixel\_value])  # Add more features as needed

    # Assign a label to the image

    # Extract the numeric part from the filename without the file extension

    label = int(os.path.splitext(image\_file)[0])

    labels.append(label)

# Calculate entropy for the entire dataset

class\_entropy = calculate\_entropy(labels)

print(f"Entropy for the entire dataset: {class\_entropy}")

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(images, labels, test\_size=0.2)

# Train a decision tree classifier

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

# Make predictions on the testing set

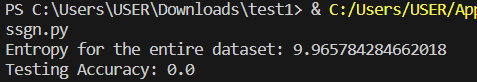
y\_pred = clf.predict(X\_test)

# Calculate accuracy

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Testing Accuracy: {accuracy}")

**OUTPUT:**

****