In [7]:

```
# This is the code for the extraction of features of Raveling images, I extracte
# Then both csv files are merged containing features of both named as 'images_fo
# CODE FOR EXTRACTING ALL 34 FEATURES OF TRAIN(RAVELING & NON-RAVELING) FOLDER o
# CURRENT CODE FOR TRAIN (RAVELLING IMAGES)
# NAME = KRISHAN KUMAR RAI
# ROLL NO. = 231030029

# importing required libraries for extracting 2 types of features color-based an
import os
import cv2
import mahotas
import numpy as np
from scipy.stats import skew, kurtosis, entropy
import pandas as pd

def calculate_entropy(channel):
    """Calculate entropy of an image channel using histogram."""
    hist, _ = np.histogram(channel, bins=256, range=(0, 256), density=True)
    return entropy(hist, base=2)

def extract_color_features(image):
    features = {}
    blue, green, red = cv2.split(image)


    for channel_name, channel in zip(['red', 'green', 'blue'], [red, green, blue
        features[f'mean_{channel_name}'] = np.mean(channel)
        features[f'std_{channel_name}'] = np.std(channel)
        features[f'skewness_{channel_name}'] = skew(channel.flatten())
        features[f'kurtosis_{channel_name}'] = kurtosis(channel.flatten())
        features[f'entropy_{channel_name}'] = calculate_entropy(channel)
        features[f'range_{channel_name}'] = np.ptp(channel)

    return features


def extract_glcm_features(image):
    features = {}

    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)


    glcm = mahotas.features.haralick(gray_image)

    angles = ['0_deg', '45_deg', '90_deg', '135_deg']
    for i, angle in enumerate(angles):
        features[f'contrast_{angle}'] = glcm[i, 1]
        features[f'correlation_{angle}'] = glcm[i, 2]
        features[f'asm_{angle}'] = glcm[i, 0]
        features[f'entropy_{angle}'] = glcm[i, 8]

    return features

def extract_all_features(image):
    features = {}
    features.update(extract_color_features(image))
    features.update(extract_glcm_features(image))
    return features
```

```
image_folder_path = 'C:/Users/krishan/Desktop/Raveling'

image_files = [f for f in os.listdir(image_folder_path) if f.endswith('.jpg')]

features_list = []

for image_file in image_files:
    image_path = os.path.join(image_folder_path, image_file)
    image = cv2.imread(image_path)

    if image is not None:

        features = extract_all_features(image)
        features['image_name'] = image_file
        features_list.append(features)

features_df = pd.DataFrame(features_list)

csv_file_path = 'krishan_features.csv'
features_df.to_csv(csv_file_path, index=False)

print(f"Feature extraction complete. CSV saved at {csv_file_path}")
```

```
Feature extraction complete. CSV saved at krishan_features.csv
```

In [8]:

```
# CODE FOR EXTRACTING ALL 34 FEATURES OF TEST(RAVELING & NON-RAVELING) FOLDER OF
# This is the code for the extraction of features of Raveling images, I extracte
# Then both csv files are merged containing features of both named as 'images_fo
# NAME = KRISHAN KUMAR RAI
# ROLL NO. = 231030029

import os
import cv2
import mahotas
import numpy as np
from scipy.stats import skew, kurtosis, entropy
import pandas as pd

def calculate_entropy(channel):
    """Calculate entropy of an image channel using histogram."""
    hist, _ = np.histogram(channel, bins=256, range=(0, 256), density=True)
    return entropy(hist, base=2)

def extract_color_features(image):
    features = {}
    blue, green, red = cv2.split(image)

    for channel_name, channel in zip(['red', 'green', 'blue'], [red, green, blue
        features[f'mean_{channel_name}'] = np.mean(channel)
        features[f'std_{channel_name}'] = np.std(channel)
        features[f'skewness_{channel_name}'] = skew(channel.flatten())
        features[f'kurtosis_{channel_name}'] = kurtosis(channel.flatten())
        features[f'entropy_{channel_name}'] = calculate_entropy(channel)
        features[f'range_{channel_name}'] = np.ptp(channel)

    return features
def extract_glcm_features(image):
    features = {}
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    glcm = mahotas.features.haralick(gray_image)
    angles = ['0_deg', '45_deg', '90_deg', '135_deg']
    for i, angle in enumerate(angles):
        features[f'contrast_{angle}'] = glcm[i, 1]
        features[f'correlation_{angle}'] = glcm[i, 2]
        features[f'asm_{angle}'] = glcm[i, 0]
        features[f'entropy_{angle}'] = glcm[i, 8]

    return features
def extract_all_features(image):
    features = {}
    features.update(extract_color_features(image))
    features.update(extract_glcm_features(image))
    return features
image_folder_path = 'C:/Users/krishan/Desktop/test'
image_files = [f for f in os.listdir(image_folder_path) if f.endswith('.jpg')]
features_list = []
for image_file in image_files:
    image_path = os.path.join(image_folder_path, image_file)
    image = cv2.imread(image_path)

    if image is not None:
        features = extract_all_features(image)
        features['image_name'] = image_file
        features_list.append(features)
features_df = pd.DataFrame(features_list)
csv_file_path = 'krishan_test.csv'
features_df.to_csv(csv_file_path, index=False)

print(f"Feature extraction complete. CSV saved at {csv_file_path}")
```

```
Feature extraction complete. CSV saved at krishan_test.csv
```

In [ ]: