

Experiment-10

Texture Analysis based on GLCM

Name : N U Praneeth Reddy

Reg.No: 21BAI1500

Aim: To perform GLCM (Gray Level Co-Occurrence Matrix) on a mountain image, with the Mountain as the foreground and the sky behind it as the background.

Resources Used: Anaconda Python Environment

Google Collab Jupyter Notebook

Theory :

OpenCV stands as an open-source library designed for computer vision and machine learning applications. Its primary goal is to offer a unified foundation for computer vision projects and to facilitate the integration of machine perception into various commercial products.

On the other hand, NumPy serves as a Python library, enabling support for large, multi-dimensional arrays and matrices, accompanied by an extensive array of high-level mathematical functions for manipulating these arrays.

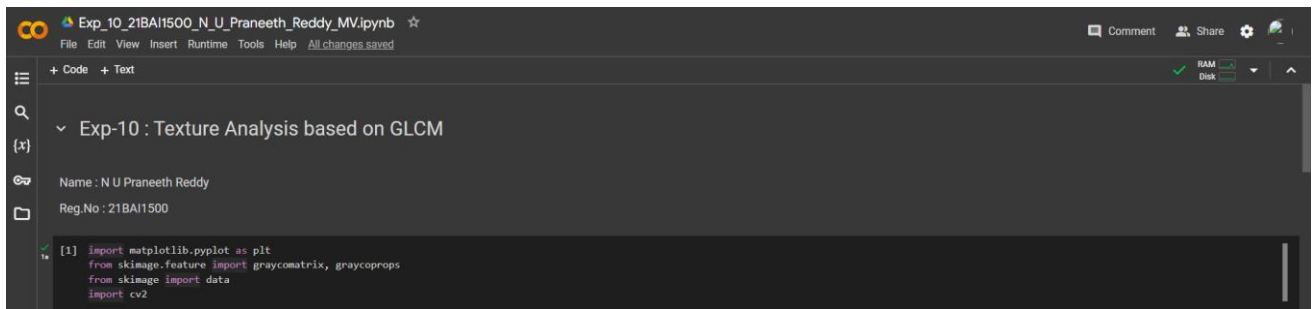
Additionally, Matplotlib functions as a Python plotting library, directly connected to the numerical mathematics capabilities of NumPy. It delivers an object-oriented API for seamlessly embedding plots within applications.

Tasks:

- 1) Obtain Region of Interest (ROI) of the given image namely few points for the foreground and few points on the background.
- 2) Compute the GLCM for the selected ROIs and display the min and max value of the selected ROIs
- 3) Provide a scatter plot considering the features contrast and Correlation and provide your inference

Procedure :

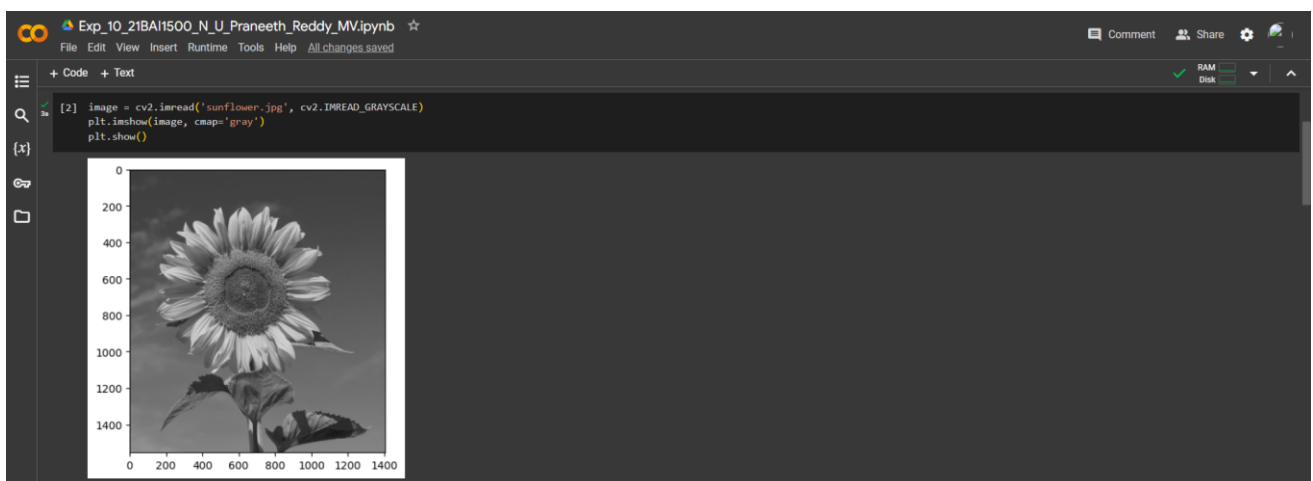
- Open Google Collab and create a new Jupyter Notebook.
- Import important libraries namely OpenCV, Numpy and Matplotlib.



```
[1] import matplotlib.pyplot as plt
from skimage.feature import graycomatrix, graycoprops
from skimage import data
import cv2
```

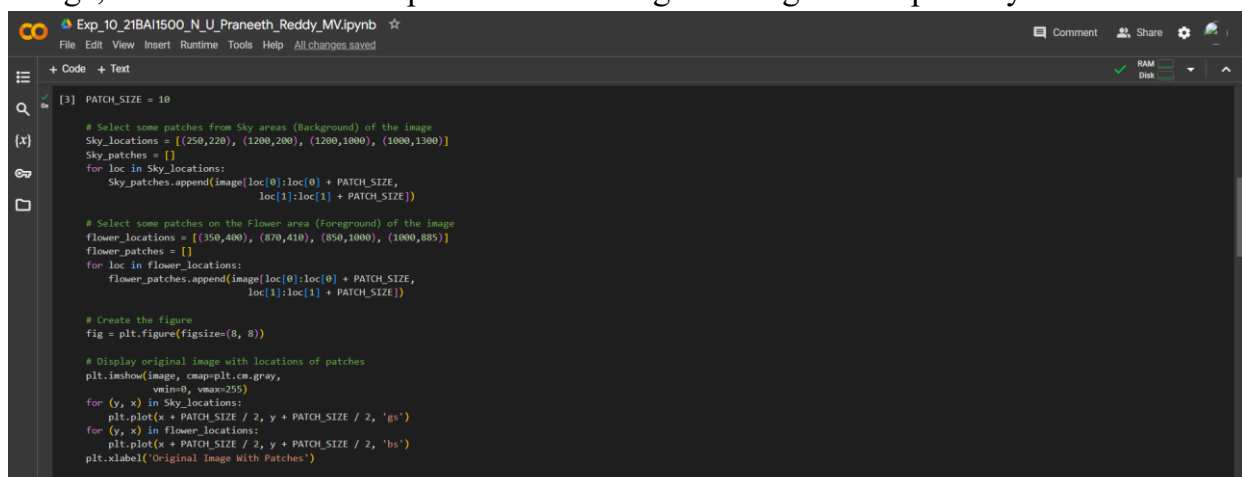
Task 1: Obtain Region of Interest (ROI) of the given image namely few points for the foreground and few points on the background.

- Read the image in grayscale and observe it for points in the foreground and background.



```
[2] image = cv2.imread('sunflower.jpg', cv2.IMREAD_GRAYSCALE)
plt.imshow(image, cmap='gray')
plt.show()
```

- Selects patches from the sky (background) and flower (foreground) areas of the grayscale image, then visualizes these patches on the original image and separately.



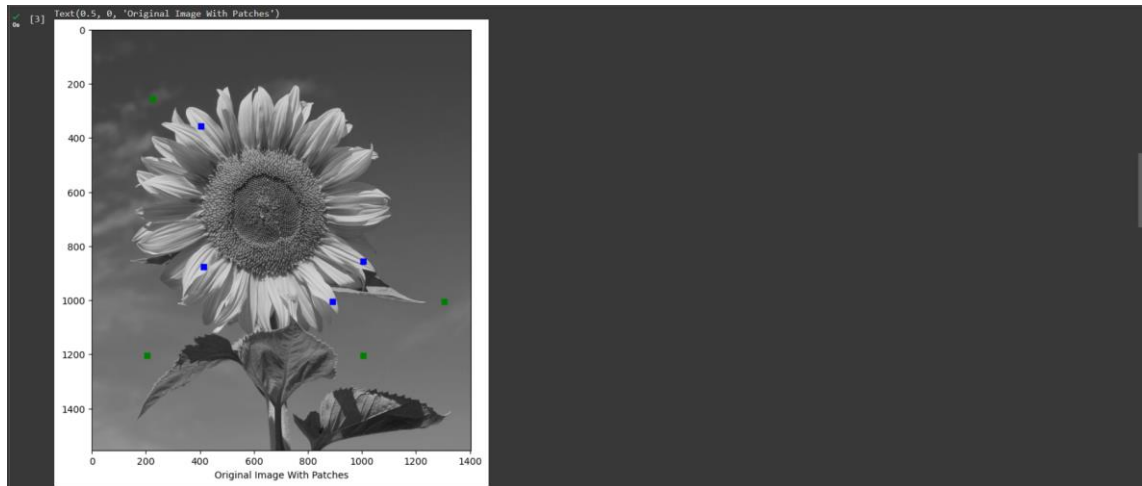
```
[3] PATCH_SIZE = 10

# Select some patches from Sky areas (Background) of the image
Sky_locations = [(250,220), (1200,200), (1200,1000), (1000,1300)]
Sky_patches = []
for loc in Sky_locations:
    Sky_patches.append(image[loc[0]:loc[0] + PATCH_SIZE,
                           loc[1]:loc[1] + PATCH_SIZE])

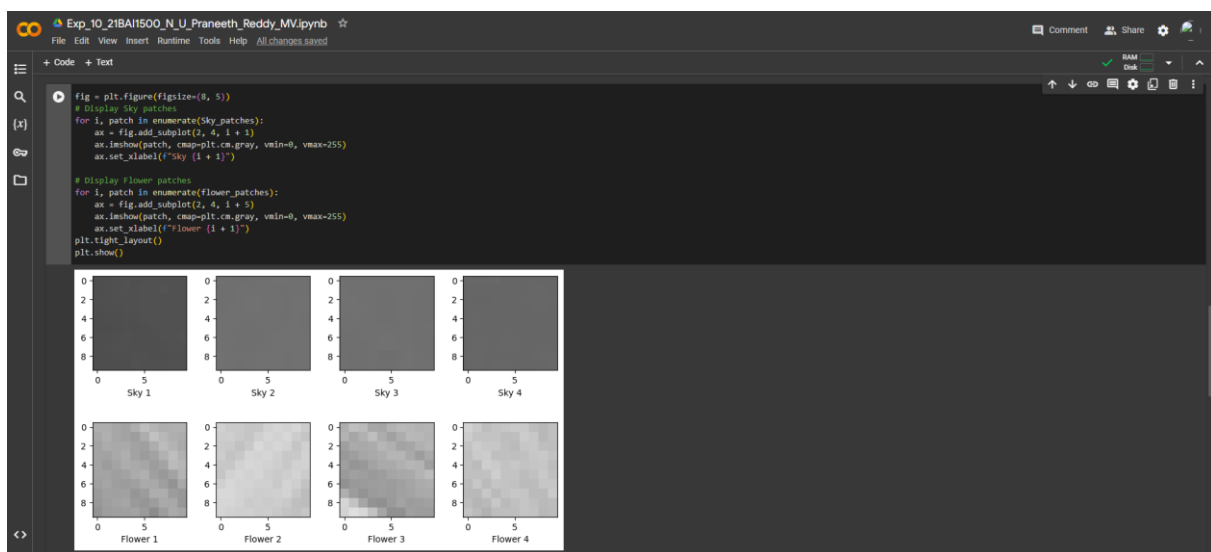
# Select some patches on the flower area (Foreground) of the image
flower_locations = [(350,400), (870,410), (850,1000), (1000,885)]
flower_patches = []
for loc in flower_locations:
    flower_patches.append(image[loc[0]:loc[0] + PATCH_SIZE,
                              loc[1]:loc[1] + PATCH_SIZE])

# Create the figure
fig = plt.figure(figsize=(8, 8))

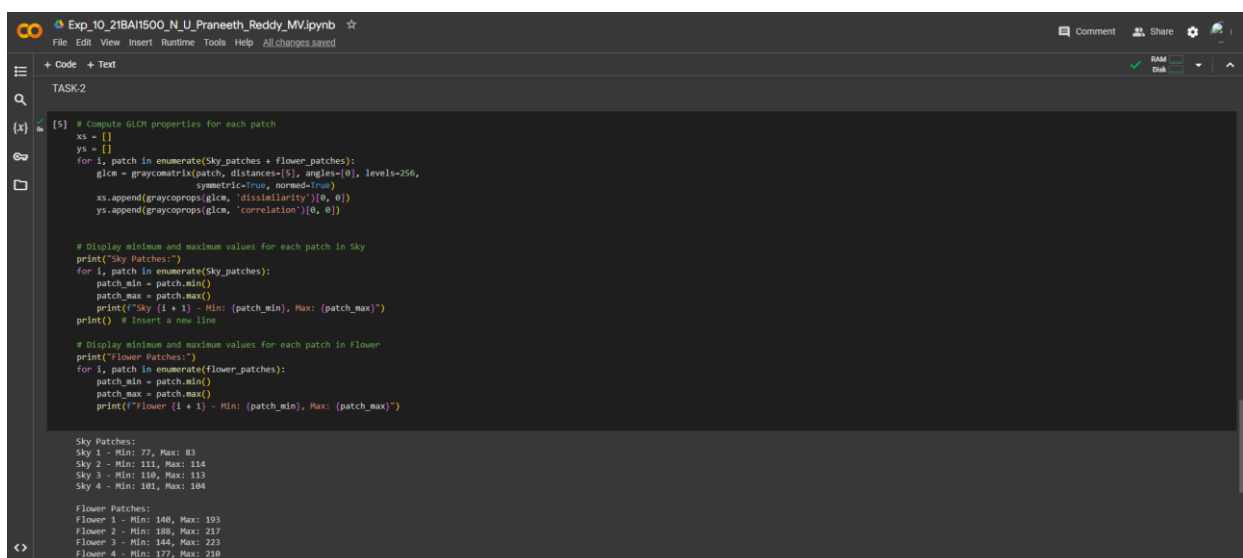
# Display original image with locations of patches
plt.imshow(image, cmap=plt.cm.gray,
           vmin=0, vmax=255)
for (y, x) in Sky_locations:
    plt.plot(x + PATCH_SIZE / 2, y + PATCH_SIZE / 2, 'gs')
for (y, x) in flower_locations:
    plt.plot(x + PATCH_SIZE / 2, y + PATCH_SIZE / 2, 'bs')
plt.xlabel('Original Image With Patches')
```



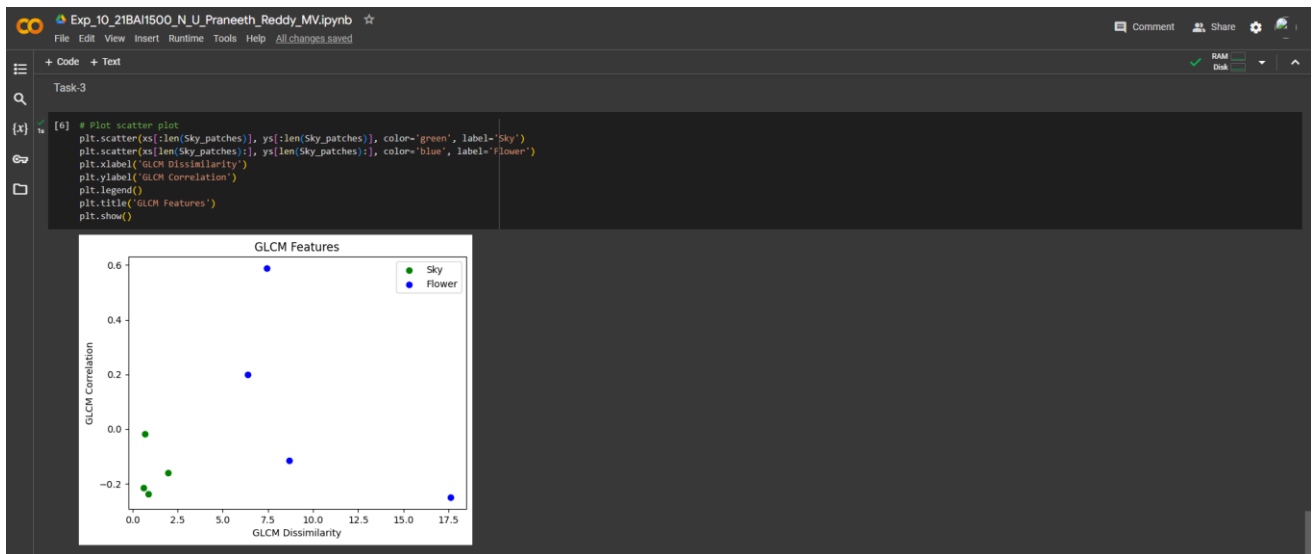
- Display sky and flower patches from the selected regions of interest (ROI)



Task-2: Computes GLCM properties (dissimilarity and correlation) for each patch and prints the minimum and maximum pixel values for both sky and flower patches.



Task-3: Plots a scatter plot showing the relationship between GLCM dissimilarity and correlation for the patches, with sky patches in green and flower patches in blue.



Inference: The scatter plot reveals sky patches with low contrast and weak correlation, indicating uniform texture. In contrast, flower patches exhibit higher contrast and a mix of positive and negative correlations, suggesting varied and intricate textures. GLCM features effectively differentiate texture characteristics, aiding in image analysis and classification tasks.

Results: The given tasks have been done using programs in Python using OpenCV, Skimage and Matplotlib libraries.

Conclusion: Python program have been created to Obtain Regions of Interest (a few points in the background - sky and a few points in the foreground - Flower) for the Flower image, GLCM for the selected ROI's has been computed, the min and the max values of the selected ROI's have been displayed and a scatter plot of features contrast vs correlation has been generated

Google Collab Link :

<https://colab.research.google.com/drive/1fRMI6FcKMLbRcZNgUkF1zp-ojudUardh?usp=sharing>