**Roll No: 120A2036**

**Batch:A2**

**Date of performance: 29/03/23**

**EXPERIMENT NO. 7**

**TO PERFORM FEATURE EXTRACTION USING CO-OCCURRENCE MATRIX**

**EXPERIMENT NO. 7: Feature extraction**

**AIM: -** To perform feature extraction using co-occurrence matrix

**OBJECTIVES:**

1. To obtain co-occurrence matrix for the given image.
2. To extract some features from the co-occurrence matrix.

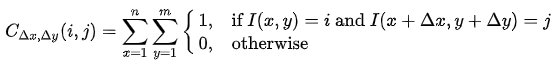
**EQUIPMENTS/SOFTWARE:** Python

**THEORY: -**

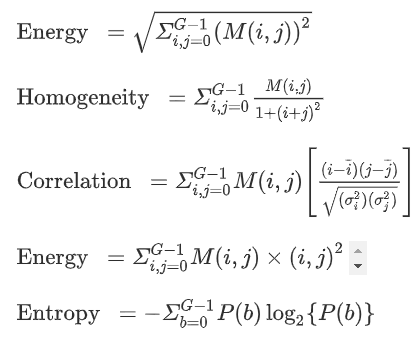
A **co-occurrence matrix** or **co-occurrence distribution** (also referred to as : *gray-level co-occurrence matrices* GLCMs) is a matrix that is defined over an image to be the distribution of co-occurring pixel values (grayscale values, or colors) at a given offset. It is used as an approach to texture analysis with various applications.

GLCM is a second-order statistical texture analysis method. It examines the spatial relationship among pixels and defines how frequently a combination of pixels are present in an image in a given direction *Θ* and distance *d*. Features are extracted. Each feature is normalized to range between 0 to 1.

For an image with {\displaystyle p}different ‘p’ pixel values, the p X p {\displaystyle p\times p}co-occurrence matrix **C** is defined over an n X m {\displaystyle n\times m}image**{\displaystyle I}**, parameterized by an offset {\displaystyle (\Delta x,\Delta y)}, as:



where: {\displaystyle i}i and {\displaystyle j}j are the pixel values; {\displaystyle x}x and {\displaystyle y}y are the spatial positions in the image **I**; the offsets {\displaystyle (\Delta x,\Delta y)} define the spatial relation for which this matrix is calculated; and {\displaystyle I(x,y)}I(x,y) indicates the pixel value at pixel {\displaystyle (x,y)}(x,y).



**Code:-**

import matplotlib.pyplot as plt  
import numpy as np  
from skimage.feature.texture import graycomatrix, graycoprops  
from skimage import data  
import cv2  
  
PATCH\_SIZE = 21  
# open the camera image  
  
image = data.camera()  
cv2.imshow("image",image)  
  
(h, w) = image.shape  
print("width={}, height={}".format(w, h))  
  
# select some patches from grassy areas of the image  
grass\_locations = [(280, 454), (342, 223), (444, 192), (455, 455)]  
grass\_patches = []  
for loc in grass\_locations:  
 grass\_patches.append(image[loc[0]:loc[0]+PATCH\_SIZE,  
 loc[1]:loc[1]+PATCH\_SIZE])  
 print(loc[0], loc[1])  
 cv2.imshow("Image grass patch",image[loc[0]:loc[0]+PATCH\_SIZE, loc[1]:loc[1]+PATCH\_SIZE])  
  
# select some patches from sky areas of the image  
sky\_locations = [(38, 34), (139, 28), (37, 437), (145, 379)]  
sky\_patches = []  
for loc in sky\_locations:  
 sky\_patches.append(image[loc[0]:loc[0] + PATCH\_SIZE,  
 loc[1]:loc[1] + PATCH\_SIZE])  
 print(loc[0], loc[1])  
 cv2.imshow("Image sky patch",(image[loc[0]:loc[0] + PATCH\_SIZE,  
 loc[1]:loc[1] + PATCH\_SIZE]))  
  
# compute some GLCM properties each patch  
xs = []  
ys = []  
for patch in (grass\_patches + sky\_patches):  
 glcm = graycomatrix(patch, distances=[5], angles=[0], levels=256,  
 symmetric=True, normed=True)  
 xs.append(graycoprops(glcm, 'dissimilarity')[0, 0])  
 ys.append(graycoprops(glcm, 'correlation')[0, 0])  
  
# create the figure  
fig = plt.figure(figsize=(8, 8))  
# display original image with locations of patches  
ax = fig.add\_subplot(3, 2, 1)  
ax.imshow(image, cmap=plt.cm.gray, vmin=0, vmax=255)  
for (y, x) in grass\_locations:  
 ax.plot(x + PATCH\_SIZE / 2, y + PATCH\_SIZE / 2, 'gs')  
 for (y, x) in sky\_locations:  
 ax.plot(x + PATCH\_SIZE / 2, y + PATCH\_SIZE / 2, 'bs')  
 ax.set\_xlabel('Original Image')  
 ax.set\_xticks([])  
 ax.set\_yticks([])  
 ax.axis('image')  
# for each patch, plot (dissimilarity, correlation)  
ax = fig.add\_subplot(3, 2, 2)  
ax.plot(xs[:len(grass\_patches)],  
ys[:len(grass\_patches)], 'go', label='Grass')  
ax.plot(xs[len(sky\_patches):], ys[len(sky\_patches):], 'bo',label='Sky')  
  
ax.set\_xlabel('GLCM Dissimilarity')  
ax.set\_ylabel('GLCM Correlation')  
ax.legend()  
# display the image patches  
  
for i, patch in enumerate(grass\_patches):  
 ax = fig.add\_subplot(3, len(grass\_patches), len(grass\_patches)\*1 + i + 1)  
 ax.imshow(patch, cmap=plt.cm.gray,  
 vmin=0, vmax=255)  
 ax.set\_xlabel('Grass %d' % (i + 1))  
 for i, patch in enumerate(sky\_patches):  
 ax = fig.add\_subplot(3, len(sky\_patches), len(sky\_patches)\*2 + i + 1)  
 ax.imshow(patch, cmap=plt.cm.gray, vmin=0, vmax=255)  
 ax.set\_xlabel('Sky %d' % (i + 1))  
  
# display the patches and plot  
fig.suptitle('Grey level co-occurrence matrix features', fontsize=14, y=1.05)  
plt.tight\_layout()  
plt.show()

**Output:-**





**Graphical user interface

Description automatically generated**

Text

Description automatically generated with low confidence

**CONCLUSION** :-

Feature extraction using co-occurrence matrix was performed successfully. We are able to obtain co-occurrence matrix for the given image.and extract some features from the co-occurrence matrix. We were make aware of the library like skimage. Using co-occurrence matrix we were to able to grass and sky patch of given image.