



INDIAN INSTITUTE OF INFORMATION
TECHNOLOGY, NAGPUR

Digital Image Processing Lab Report

Submitted By :

Pranav Bharadwaj (BT17ECE044)

Semester 6

Electronics and Communication Engineering Dept.

Submitted To :

Dr. Tapan Kumar Jain

Assistant Professor

Electronics and Communication Engineering Dept.

Codes(Python)

Bit Slicing

{**Aim:-Compressing an image by Removing LSB**

```
import cv2
import os
import numpy as np
import matplotlib.pyplot as plt

def dec2bin(pixel):
    bin_value =
    if(pixel > 1) :
        bin_value = dec2bin(pixel//2)
    bin_value.append(pixel)
    return bin_value

path = 'C:/Users/Aditya/Pictures/'
image_path = 'Lenna(test_image).png'
original_image = cv2.imread(os.path.join(path, image_path))
gray_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2GRAY)
cv2.imshow('grayscaleimage', gray_image)
X, Y = gray_image.shape[0], gray_image.shape[1]

one_image = np.zeros([X, Y])
two_image = np.zeros([X, Y])
three_image = np.zeros([X, Y])
four_image = np.zeros([X, Y])
five_image = np.zeros([X, Y])
six_image = np.zeros([X, Y])
seven_image = np.zeros([X, Y])
eight_image = np.zeros([X, Y])
```

```
arr = np.zeros([8])
for x in range(X):
    for y in range(Y):
        result = dec2bin(grayiimage[x, y])
        arr = np.zeros([8])
        for i in range(len(result)) :
            arr[7 - i] = result[len(result) - 1 - i]
        oneiimage[x, y] = arr[0]
        twoiimage[x, y] = arr[1]
        threeiimage[x, y] = arr[2]
        fouriimage[x, y] = arr[3]
        fiveiimage[x, y] = arr[4]
        sixiimage[x, y] = arr[5]
        seveniimage[x, y] = arr[6]
        eightiimage[x, y] = arr[7]

fig = plt.figure()
cv2.imshow('first image', oneiimage)
cv2.imshow('twoimage', twoiimage)
cv2.imshow('thirdimage', threeiimage)
cv2.imshow('fourthimage', fouriimage)
cv2.imshow('fifthimage', fiveiimage)
cv2.imshow('sixthimage', sixiimage)
cv2.imshow('seventhimage', seveniimage)
cv2.imshow('eighthimage', eightiimage)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Histogram Equilization

Aim:To do Histogram Equilization Of pixels

```
from PIL import Image
import numpy as np
import cv2
import matplotlib.pyplot as plt
```

function to count the pixel values

```
def Count(countlist) :
    count = dict()
    for item in countlist :
        if item in count :
            count[item] = count[item] + 1
        else :
            count[item] = 1
    return count
```

function to calculate CDF function

```
def CDF(dict1) :
    sum1 = 0
    cdf = dict()
    for i, j in count.items() :
        sum1 = sum1 + count[i]
    cdf[i] = sum1
    return cdf
```

function to calculate histogram from CDF function

```
def calculatehistogram(dict1) :
    cdfm = min(dict1.values())
    L = 255
    hist = dict()
    for i, j in cdf.items() :
```

```
hist[i] = int(((dict1[i] - cdfm)/(699392 - cdfm)) * (L - 1))
```

```
return hist
```

```
originaliimage = cv2.imread(r'C :H awkesBayNZ.jpg')
grayscaleiimage = cv2.cvtColor(originaliimage, cv2.COLOR_BGR2GRAY)
cv2.imshow('grayscaleimage', grayscaleiimage)
cv2.imwrite(r'C :i mage.jpg', grayscaleiimage)
cv2.waitKey(0)
np.array(img)
print(img2arr)
print(img2arr.shape)
imageshape = grayscaleiimage.shape
```

```
list1 = []
list2 = []
for i in range(imagesshape[0]) :
    for j in range(imagesshape[1]) :
        list1.append(grayscaleiimage[i][j])
        list2.append(grayscaleiimage[i][j])
```

```
list1.sort()
```

```
count = Count(list1)
cdf = CDF(count)
```

```
hist = calculatehistogram(cdf)
```

```
list3 = []
for x in list2 :
```

```
list3.append(hist[x])
```

```
arr2img = np.array(list3, dtype = np.uint8)
arr2img = arr2img.reshape(imageshape[0], imageshape[1])
histequiimage = cv2.imshow('histequiimage', arr2img)
cv2.imwrite(r'C :_equiimage.png', arr2img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

RGB

Aim:- Splitting Different color of Image

```
import os
import numpy as np
import cv2
```

```
path = 'C:/Users/Aditya/Pictures/'
imagepath = 'images.jpg'
originaliimage = cv2.imread(r'C :_testiimage.png')
originaliimage = cv2.imread(os.path.join(path, imagepath))
cv2.imshow(imagepath, originaliimage)
print(originaliimage.shape)
X, Y, Z = originaliimage.shape[0], originaliimage.shape[1], originaliimage.shape[2]

blueiimage = np.zeros([X, Y, Z], dtype = np.uint8)
greeniimage = np.zeros([X, Y, Z], dtype = np.uint8)
rediimage = np.zeros([X, Y, Z], dtype = np.uint8)

seperateiimages = [blueiimage, greeniimage, rediimage]
```

```
for  $x, y$  in zip(seperateiimages, range( $Z$ )) :  
 $x[:, :, y] = original_{i}image[:, :, y]$   
 $attach = str(y) + '.jpg'$   
cv2.imwrite(os.path.join(path, attach),  $x$ )
```

```
cv2.imshow('Blue Image', blueiimage)  
cv2.imshow('GreenImage', greeniimage)  
cv2.imshow('RedImage', rediimage)
```

```
cv2.waitKey(0)  
cv2.destroyAllWindows()
```