DEPARTMENT OF PHYSICS
UNIVERSITY OF COLOMBO

PH 3034- Digital Image Processing – I
2021

Images Analysis

Name: Umeshika Dissanayaka

Index Number: s14320

Date: 11/10/2021

ABSTRACT

The main purpose of this practical was to analyze the images with using Python 03 and this practical was mainly based on the basic level of image processing therefore, the main objective of this practical was to gain brief knowledge about the mechanism of storing images in computers, enhancing images in the intensity domain and extracting the spectral information out of RGB images. In addition, this report consists of study and implementation of the representing the image RGB image, Grayscale image, Binary image, converting and separating image from RGB to Gray scale, geometric operations such as rotate and flip to images, graphical representation of basic intensity analysis on images, analyzing brightness and contrast adjustments to images, Calibrating dimensions of images to real-world values, and Segmenting images using thresholding. The Samsung galaxy s10+ phone camera was used to capture the photographs and the graphical representations were mainly obtained by using pixel intensity histograms. Furthermore, this experiment can be improved by using a high-quality DSLR camera with capturing photographs. In conclusion, Image analysis is given the extricate significant data which can be obtained using python and pixel intensity histogram.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In general idea of an image is a two dimensional intensity function of f(x,y) and which is an antique that portrays visual insight such as two-dimensional screen display, a photograph. And also analyzing image which examines the extraction of significant data. Therefore, it is called the image processing. Currently one of fastest growing technologies are Image processing techniques which is used vital areas such as knowledge-based image analysis, image morphology, image data compression, image recognition, neural networks, edge identifiers and full-color image processing. In current world many fields of image processing are used digital image processing techniques and there are fundamental concepts of image processing are included with graphics applications. Therefore, it can be categorized as two method of principal application areas which are the scene data processing for autonomous machine perception and pictorial information improvement for human interpretation.

The fundamental steps in image processing are Image Acquisition, Image Enhancement, Image Restoration, Color Image Processing, Wavelets, Compression, Morphological Processing, Image Segmentation, Representation and Description, Recognition and Interpretation, and Knowledge Base. According to Image processing to computer vision has three levels that can be categorized up into basic level, intermediate level and high-level processes.

In this report mainly based on the basic level of image processing using algorithms. And also the main objective of this practical was to gain brief knowledge about the mechanism of storing images in computers, enhancing images in the intensity domain and extracting the spectral information out of RGB images. Therefore, the main purpose of this practical was to process, analyze, Implement and visualize the images and intensity histograms by using the open software of high-level programming language of Python 03.

Python 03 is an object-oriented scripting language, and it has been easy to understand therefore the prime benefits of learning python that has allowed debugging of snippets of code and interactive testing. Currently, python 03 is used in a lot of fields of image processing because everyone can download freely and various fundamental standard image processing libraries and packages are available in python which can import and get the output easily such as Scikit-image, Opencv, NumPy and matplotlib. Not only that it is it is utilized familiar English syntaxes which are easy to understand and learn.

2 THEORY

2.1 Pixel

The minimum unit of an image is called pixel and it usually has square shape or dots. Therefore, the digital images are construct by using this minimum unit. The high-quality image has large amount of very small size pixel and low-quality image has normal amount of pixel with normal size. Furthermore, millions of different colors can be visualized using red, blue, and green lighting pixel. And also the horizontal and vertical measurements of an image is represented using pixel dimensions.

2.2 RGB Image

The RGB image is created with the three-color combination of red, green, blue pixels and 8-bit numbers of red, green, blue color value include the in an RGB image of pixel. It can be categorized as the three channels of red, green, blue and these colors are used in computer vision as human eye receptors.

2.3 Splitting Layers

Evidentially, the red, green, blue colors of three integers represent in the pixel of the image. Therefore, the idea of Splitting the Layers is separated to the three RGB channel layers of image with the image array.

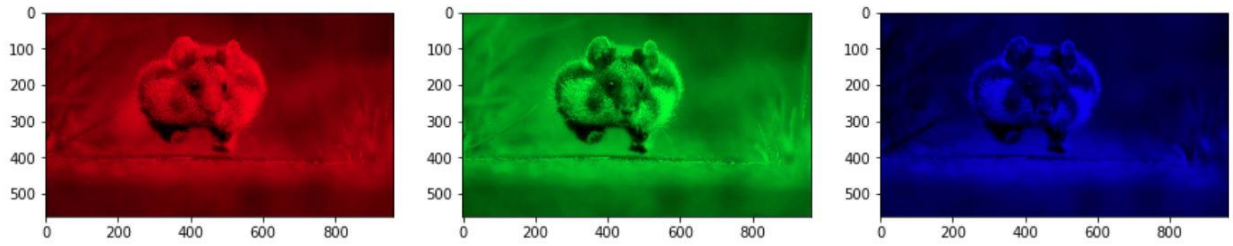


Figure 2.3-1 The image of Splitting the Layers that is separated to the three RGB channel layers of image (Resource:https://www.codementor.io/@innat_2k14/image-data-analysis-using-numpy-opencv-part-1-kfadbafx6)

2.4 Grey images

Using 2-Dimentional arrays, it can be stored the Black and white images which are mainly divided to the two types of Black and White images. They are Greyscale image that includes with Ranges of shades of grey 0 to 255 bit and Binary image which Pixel are either black or white (0 or 255 bits). Therefore the main idea of Grey scaling image is converting from a full color to shades of grey.

Figure 2.4-1 The grey scale image (Resource: https://www.codementor.io/@innat_2k14/image-data-analysis-using-numpy-opencv-part-1-kfadbafx6)

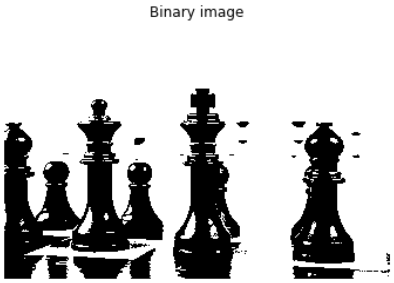


Figure 2.4-2 The binary image (Resource:https://colab.research.google.com/github/goodboychan/chans_jupyter/blob/main/_notebooks/2020-07-31-01-Introducing-Image-Processing-and-scikit-image.ipynb#scrollTo=FWA8e2BN-YeU)

2.5 Histogram

The histogram is a statistical data graphical representation with the frequency of the dataset. Therefore, the distribution of pixels of an image can be visualized using histograms. Hence the histogram for an image always shows how frequently various color values distribute. This statistical method very useful for image processing such as Thresholding, Contrast, Exposure, Saturation and Dynamic Range.

n an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values. Histograms can also be taken of color images --- either individual histograms of red, green and blue channels can be taken, or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count. The exact output from the operation depends upon the implementation --- it may simply be a picture of the required histogram in a suitable image format, or it may be a data file of some sort representing the histogram statistics.

2.5.1 RGB histrgram and grey color histogram

The image can be separated as three layers for red, green and blue therefore, the histograms can be obtained for each color channel. This histogram is visualized with the pixel intensity count and pixel intensity value. And also 8 bit grayscale image has 256 different intensities. The grey color histogram is obtained using 8 bit grayscale image.

2.6 Exposure triangle

The exposure triangle has three sides which are Aperture, shutter speed, and ISO. The good combination of these three elements are given the properly exposed photograph.

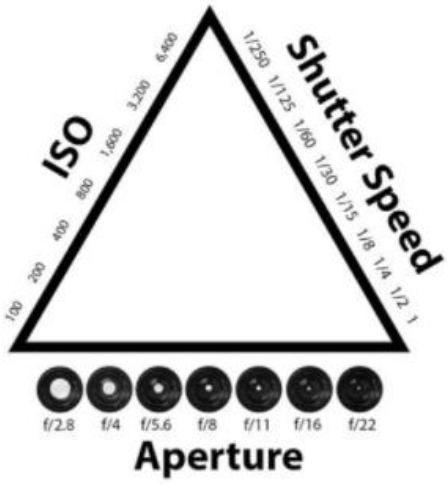


Figure 2.6-1 The exposure triangle (referernce:https://photographylife.com/what-is-exposure-triangle)

2.6.1 ISO

ISO is the sensitivity of the digital sensor and high values of ISO of digital sensor does not need to lighter for quality exposure and also low value ISO digital sensor does need to more light for quality exposure.

2.6.2 shutter speed

The time period for hitting the sensor is call shutter speed. There are various shutter speeds are available in smart phone cameras that are 1/16000 s to 30s which are depend on the phone camera quality.

2.7 white balance

Adjusting the color according to the color of the light source white object visualized white and using proper camera to balance the color temperature of light source is called white balance that has relative warmth and coolness.

2.8 Raileygh Scattering

When particles that have a radius less than 1/10 the wavelength of the radiation that would be dispersion of electromagnetic radiation is called Rayleigh scattering.

2.9 Global and Local thresholding

The Global thresholding is the degree of intensity separation with two intensity values in the image and local thresholding consider the intensity values for every pixel in the image withgray scale.

3 METHODOLOGY

3.1 Exercise 1:

- A photograph of the face was taken using the mobile phone and copied it to the computer. After that image is loaded into the computer using the Scikit Image library in python and showed the loaded image using matplotlib then changed the title to filename.
- The new function was defined which named “imshow” to plot and show the grayscale image without axis

$$\text{def}\, imshow(image, title = \text{'image title'}, \text{cmap\_type} = \text{'gray'}):$$

- The grayscale image was showed using “imshow” function and converted again that image into a grayscale image. After that the grayscale image was saved the computer as a “png” in a different folder
- Then the memory of the IDE was cleared using “gc.collect()” command and loaded both RGB and grayscale images two variables I1 and I2. The dimensions of them were checked using “.shape” command.
- 02.585338 s was taken to execute the code and following code was used to obtained the results.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct  6 21:55:47 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
from skimage.io import imread
import matplotlib.pyplot as plt
from skimage import  color
import gc
#The variable is create to get the starting time when the code is run
starttime=datetime.now()
img = imread('face.jpg') #load the image

#%%define the function of imshow1
def imshow1(image, title1, cmap_type='gray'):
          imggray=color.rgb2gray(image)#convert data to gray color
          plt.imshow(imggray, cmap=cmap_type)#take the gray scale image
          plt.title(title1)#label the title
          plt.axis('off')#off the axis
```

```
23          plt.figure(1) #plot figure
24          fig = plt.figure(1)
25          plt.show()
26          return fig
27 #gray color image is taken using imshow1 function
28 imshow1(img,"face",cmap_type='gray')
29 #save the gray color image different location
30 fig=imshow1(img,"face",cmap_type='gray')
31 fig.savefig('E:/Third Year/Second Semester
32             /PH 3034 Digital Image Processing 1/week 03/faceimage.png')
33 #load the saved image
34 img2=imread('E:/Third Year/Second Semester
35             /PH 3034 Digital Image Processing 1/week 03/faceimage.png')
36 #The saved grayscale image is Converted into a grayscale
37 image2=color.rgb2gray(img2)
38 plt.imshow(image2, cmap='gray')#take the gray scale image
39 plt.title("gray color image 2")#label the title
40 plt.axis('off')#off the axis
41 plt.figure(2)#plot figure
42 plt.show()#show image
43
44 gc.collect()#Clear memory of the IDE
45 #Load both RGB and grayscale images two variables I1 and I2
46 I1=imread('face.jpg')
47 I2=imread('E:/Third Year/Second Semester/
48          PH 3034 Digital Image Processing 1/week 03/faceimage.png')
49 #take the dimensions of them
50 print(I1.shape)
51 print(I2.shape)
52
53 """The code executing time is calculated using
54  difference of the start time and end time"""
55 print (datetime.now()-starttime)
56
57
```

3.2 Exercise 2:

- The sample image of “ouc_pg_convocation.jpg” was loaded in the sample image folder and separated three (RBG) layers of images into three variables of r,g, and b. After that “r”,”g”, and “b” variables were showed in a separate figure.
- The original image was rotated using “rot90 function” in “numpy”. And image was shown after rotated by 90°, 180°, and 270° angles in separate figures. After that image was flipped horizontally and vertically using “fliplr” command and “flipud’ command in numpy functions.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Oct  8 10:10:09 2021
```

```
 4
 5 @author: umeshika
 6 """
 7 #import the libraries
 8 from datetime import datetime
 9 import numpy as np #import numpy library
10 from skimage.io import imread
11 import matplotlib.pyplot as plt
12
13 #The variable is create to get the starting time when the code is run
14 starttime=datetime.now()
15
16 img = imread('ouc_pg_convocation.jpg') #load the image
17 #define function image to plot figures
18 def image(data,title1, grid_size, grid="on"):
19  d=data
20  plt.figure()
21  plt.title(title1)
22  plt.imshow(d)
23 #define variables for Separate red,green,blue layers
24 r=img[:,:,0]#R layer
25 g=img[:,:,1]#G layer
26 b=img[:,:,2]#B layer
27 image(r,"R channel",5) #plot R channel of image
28 image(g,"G channel", 5) #plot G channel of image
29 image(b,"B channel", 5) #plot B channel of image
30 #%%
31 f_r = np.rot90(img)#rotate image 90'
32 f_r1 = np.rot90(img, 2)#rotate image 180'
33 f_r2= np.rot90(img, 3)#rotate image 270'
34 image(f_r,"The image of 90' rotation",5)#plot 90'rotation of image
35 image(f_r1,"The image of 180' rotation",5)#plot 180' rotation of image
36 image(f_r2,"The image of 270' rotation",5)#plot 270' rotation of image
37 #%%
38
39 #the image vertically flips
40 img_vertical_flip = np.flipud(img)
41
42 #the image horizontally flips
43 img_horizontal_flip = np.fliplr(img)
44
45 # plot the flip images
46 image(img_horizontal_flip, 'The horizontal flip image',5)
47 image(img_vertical_flip, 'The vertical flip image',5)
48
49 """The code executing time is calculated using
50  difference of the start time and end time"""
51 print (datetime.now()-starttime)
```

- Secondly, the 'moon.jpg' and 'dark_sky.jpg' images were loaded to two variables. the intensity histogram of them were plot using hist function available with matplotlib library and 255 bins was used when using hist function.

```
# -*- coding: utf-8 -*-
"""
Created on Fri Oct  8 13:23:22 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
import numpy as np #import numpy library
from skimage.io import imread
import matplotlib.pyplot as plt
#The variable is create to get the starting time when the code is run
starttime=datetime.now()
#load the images
img = imread('moon.jpg')
img1 = imread('dark_sky.jpg')

#define hist function for ploting histogram
def histogram(image,title1,ylim):
            plt.hist(image,bins=np.arange(255))
            plt.ylim([0,ylim])
            plt.title(title1)
            plt.xlabel("Pixels Intesity")
            plt.ylabel("Pixels Intesity Count")
            plt.show()
#plot histogram
histogram (img,'HIstogram for moon Image',8)
histogram(img1,'HIstogram for dark sky Image',420)
"""The code executing time is calculated using
 difference of the start time and end time"""
print (datetime.now()-starttime)
```

- The histograms for RBG channels of ouc_pg_convocation.jpg file were taken using histogram function.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct  6 22:19:10 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
import numpy as np
from skimage.io import imread
import matplotlib.pyplot as plt
```

```
12
13 #The variable is create to get the starting time when the code is run
14 starttime=datetime.now()
15 img = imread('ouc_pg_convocation.jpg') #load the image
16 #define variables for Separate three RGB layers
17 r=img[:,:,0]#R layer
18 g=img[:,:,1]#G layer
19 b=img[:,:,2]#B layer
20 #%%define hist function for ploting histogram
21 def histogram(image,title1,ylim,colour):
22             plt.hist(image.ravel(),bins=np.arange(255), color=colour)
23             plt.ylim([0,ylim])
24             plt.xlim([-2,260])
25             plt.title(title1)
26             plt.xlabel("Pixels Intesity")
27             plt.ylabel("Pixels Intesity Count")
28             plt.show()
29 #plot histogram
30 histogram(r,'HIstogram for R channel Convocation Image',16000,'red')
31 histogram(g,'HIstogram for G channel Convocation Image',40000,'green')
32 histogram(b,'HIstogram for B channel Convocation Image',7000,'blue')
33
34
35 """The code executing time is calculated using
36  difference of the start time and end time"""
37 print (datetime.now()-starttime)
```

3.3 Exercise 3:

- The parameters of ISO, exposure, shutter speed, white balance and HDR of Samsung galaxy s10+ phone camera was learnt and arranged the suitable indoor place to do photographing where have artificial lighting.
- The Samsung galaxy s10+ phone camera was turned on and set it to manual mode, switched of auto white balance and got the HDR.
- The same photograph with 10 different ISO levels, 10 different shutter speeds, and 10 different exposures were taken and the histograms for each setting were plot

Table 3-1 The table of 10 different ISO level

Image number	ISO level
01	50
02	80
03	100
04	160
05	250
06	400
07	640

08	800
09	1600
10	3200

Table 3-2 The table of 10 different shutter speed level

Image number	Shutter Speed (s)
01	$\frac{1}{16000}$
02	$\frac{1}{12000}$
03	$\frac{1}{2000}$
04	$\frac{1}{1000}$
05	$\frac{1}{500}$
06	$\frac{1}{250}$
07	$\frac{1}{125}$
08	$\frac{1}{60}$
09	$\frac{1}{45}$
10	$\frac{1}{20}$

Table 3-3 The table of 10 different exposures

Different exposures	ISO	Shutter Speed (s)
01	3200	$\frac{1}{12000}$
02	1600	$\frac{1}{4000}$
03	800	$\frac{1}{2000}$
04	500	$\frac{1}{1500}$
05	400	$\frac{1}{350}$
06	400	$\frac{1}{125}$

07	320	$\frac{1}{45}$
08	250	$\frac{1}{20}$
09	100	$\frac{1}{6}$
10	50	30

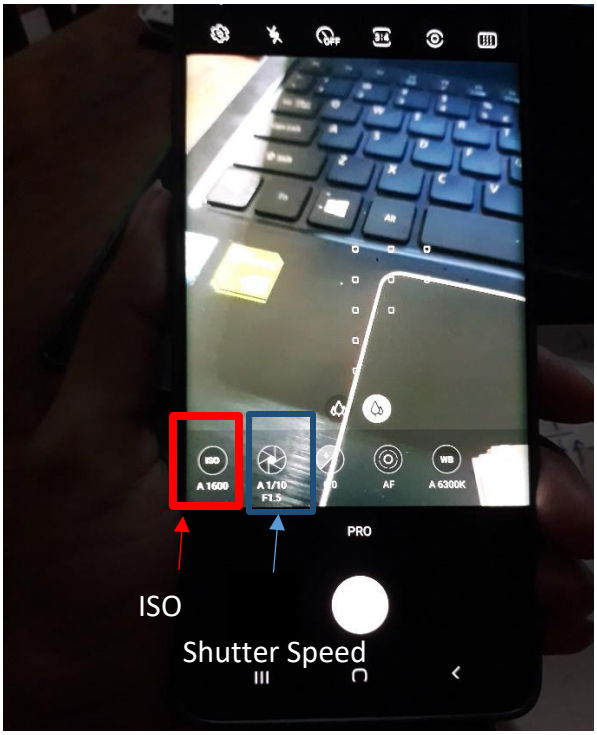


Figure 3.3-1 The image of Samsung galaxy s10+ phone camera ISO, exposure and shutter speed setting

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct  7 19:19:15 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
import numpy as np
from skimage.io import imread
import matplotlib.pyplot as plt

#The variable is create to get the starting time when the code is run
starttime=datetime.now()
#load the images
img = imread('50.jpg')
```

```
img1 = imread('80.jpg')
img2 = imread('100.jpg')
img3 = imread('160.jpg')
img4 = imread('250.jpg')
img5 = imread('400.jpg')
img6 = imread('640.jpg')
img7 = imread('800.jpg')
img8 = imread('1600.jpg')
img9 = imread('3200.jpg')


#define function to plot histogram
def histogram(imagehis,title1,ylim,colour):
            plt.hist(imagehis.ravel(),bins=np.arange(255), color=colour)
            plt.ylim([0,ylim])
            plt.xlim([-2,260])
            plt.title(title1)
            plt.xlabel("Pixels Intesity")
            plt.ylabel("Pixels Intesity Count")
            plt.show()

#plot histogram for diffrent ISO levels
histogram(img,'HIstogram for ISO level 50 Image',850000,'purple')
histogram(img1,'HIstogram for ISO level 80 Image',850000,'purple')
histogram(img2,'HIstogram for ISO level 100 Image',850000,'purple')
histogram(img3,'HIstogram for ISO level 160 Image',850000,'purple')
histogram(img4,'HIstogram for ISO level 250 Image',850000,'purple')
histogram(img5,'HIstogram for ISO level 400 Image',850000,'purple')
histogram(img6,'HIstogram for ISO level 640 Image',850000,'purple')
histogram(img7,'HIstogram for ISO level 800 Image',850000,'purple')
histogram(img8,'HIstogram for ISO level 1600 Image',850000,'purple')
histogram(img9,'HIstogram for ISO level 3200 Image',850000,'purple')

"""The code executing time is calculated using
 difference of the start time and end time"""
print (datetime.now()-starttime)
```

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct  7 19:19:15 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
import numpy as np
from skimage.io import imread
import matplotlib.pyplot as plt

#The variable is create to get the starting time when the code is run
starttime=datetime.now()
```

```
#load the images
imgs = imread('1_16000.jpg')
imgs1 = imread('2_12000.jpg')
imgs2 = imread('3_2000.jpg')
imgs3 = imread('4_1000.jpg')
imgs4 = imread('5_500.jpg')
imgs5 = imread('6_250.jpg')
imgs6 = imread('7_125.jpg')
imgs7 = imread('8_60.jpg')
imgs8 = imread('9_45.jpg')
imgs9 = imread('10_10.jpg')


#define function to plot histogram
def histogram(imagehis,title1,ylim,colour):
            plt.hist(imagehis.ravel(),bins=np.arange(255), color=colour)
            plt.ylim([0,ylim])
            plt.xlim([-2,260])
            plt.title(title1)
            plt.xlabel("Pixels Intesity")
            plt.ylabel("Pixels Intesity Count")
            plt.show()
#plot histogram for diffrent ISO levels
histogram(imgs,'HIstogram for 1/16000 s Shutter Speed of Image',10000000,'deepskyblue')
histogram(imgs1,'HIstogram for 1/12000 s Shutter Speed of Image',10000000,'deepskyblue')
histogram(imgs2,'HIstogram for 1/2000 s Shutter Speed of Image',1500000,'deepskyblue')
histogram(imgs3,'HIstogram for 1/1000 s Shutter Speed of Image',1500000,'deepskyblue')
histogram(imgs4,'HIstogram for 1/500 s Shutter Speed of Image',850000,'deepskyblue')
histogram(imgs5,'HIstogram for 1/250 s Shutter Speed ofImage',850000,'deepskyblue')
histogram(imgs6,'HIstogram for 1/125 s Shutter Speed of Image',850000,'deepskyblue')
histogram(imgs7,'HIstogram for 1/60 s Shutter Speed of Image',850000,'deepskyblue')
histogram(imgs8,'HIstogram for 1/45 s Shutter Speed of Image',1000000,'deepskyblue')
histogram(imgs9,'HIstogram for 1/20 s Shutter Speed of Image',1000000,'deepskyblue')
"""The code executing time is calculated using
 difference of the start time and end time"""
print (datetime.now()-starttime)
```

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct  7 19:19:15 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
import numpy as np
from skimage.io import imread
import matplotlib.pyplot as plt

#The variable is create to get the starting time when the code is run
starttime=datetime.now()
```

```
#load the images
img = imread('1.jpg')
img1 = imread('2.jpg')
img2 = imread('3.jpg')
img3 = imread('4.jpg')
img4 = imread('5.jpg')
img5 = imread('6.jpg')
img6 = imread('7.jpg')
img7 = imread('8.jpg')
img8 = imread('9.jpg')
img9 = imread('10.jpg')


#define function to plot histogram
def histogram(imagehis,title1,ylim,colour):
            plt.hist(imagehis.ravel(),bins=np.arange(255), color=colour)
            plt.ylim([0,ylim])
            plt.xlim([-2,260])
            plt.title(title1)
            plt.xlabel("Pixels Intesity")
            plt.ylabel("Pixels Intesity Count")
            plt.show()

#plot histogram for diffrent ISO levels
histogram(img,'HIstogram for exposure Image 1',850000,'Yellow')
histogram(img1,'HIstogram for exposure Image 2',850000,'Yellow')
histogram(img2,'HIstogram for exposure Image 3',850000,'Yellow')
histogram(img3,'HIstogram for exposure Image 4',850000,'Yellow')
histogram(img4,'HIstogram for exposure Image 5',850000,'Yellow')
histogram(img5,'HIstogram for exposure Image 6',850000,'Yellow')
histogram(img6,'HIstogram for exposure Image 7',850000,'Yellow')
histogram(img7,'HIstogram for exposure Image 8',850000,'Yellow')
histogram(img8,'HIstogram for exposure Image 9',850000,'Yellow')
histogram(img9,'HIstogram for exposure Image 10',850000,'Yellow')

"""The code executing time is calculated using
 difference of the start time and end time"""
print (datetime.now()-starttime)
```

- The image ‘village.bmp’ was loaded and obtained the intensity histogram of it.
- The issue of the ‘village.bmp’ image was fix with using of contrast stretching techniques for maximum utilizing the dynamic range.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct  7 19:19:15 2021

@author: umeshika
"""
#import the libraries
from datetime import datetime
```

```
import numpy as np
import cv2
from skimage.io import imread
import matplotlib.pyplot as plt

#The variable is create to get the starting time when the code is run
starttime=datetime.now()
img = imread('village.bmp') #load the image
def histogram(imagehis,title1,ylim,colour):
            plt.hist(imagehis.ravel(),bins=np.arange(255), color=colour)
            plt.ylim([0,ylim])
            plt.xlim([-2,260])
            plt.title(title1)
            plt.xlabel("Pixels Intesity")
            plt.ylabel("Pixels Intesity Count")
            plt.show()
#plot histogram
histogram(img,'HIstogram for Village Image',25000,'purple')
#%%

# Read the image
img2 = cv2.imread('village.bmp',0)

def image(imageplot,title1,grid="on"):
 plt.figure()
 plt.title(title1)
 plt.imshow(imageplot)

# stretched image Creates with zeros array to store
r_min_max= np.zeros((img2.shape[0],img2.shape[1]),dtype = 'uint8')

# apply Min-Max formulae using Loop over the image
for i in range(img2.shape[0]):
    for j in range(img2.shape[1]):
        r_min_max[i,j] = 255*(img2[i,j]-np.min(img2))/(np.max(img2)-
np.min(img2))

# plot the Contrast stretched image
image(r_min_max,'The Contrast Stretched Image')
cv2.imshow('The Contrast Stretched Image',r_min_max)
#save the image
cv2.imwrite('The Contrast Stretched Image.png', r_min_max)

cv2.waitKey(0)
"""The code executing time is calculated using
 difference of the start time and end time"""
  print (datetime.now()-starttime)
```

- The photograph of the famous Niger river was loaded and obtaibed the intensity histogram of it.

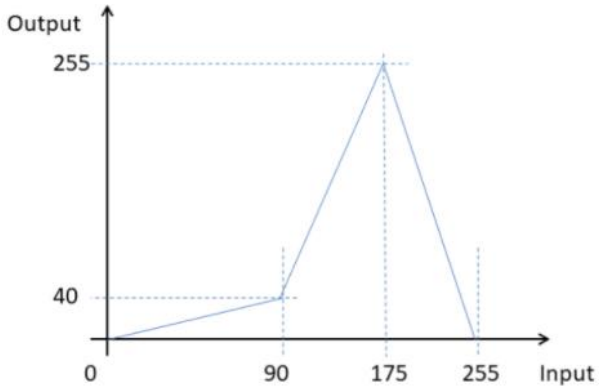
```python
# -*- coding: utf-8 -*-
"""
Created on Sat Oct  9 05:44:53 2021

@author: Umeshika
"""
#import the libraries
from datetime import datetime
import numpy as np
import cv2
from skimage.io import imread
import matplotlib.pyplot as plt

#The variable is create to get the starting time when the code is run
starttime=datetime.now()
img = imread('niger.bmp') #load the image
def histogram(imagehis,title1,ylim):
            plt.hist(imagehis.ravel(),bins=np.arange(255))
            plt.ylim([0,ylim])
            plt.xlim([-2,260])
            plt.title(title1)
            plt.xlabel("Pixels Intesity")
            plt.ylabel("Pixels Intesity Count")
            plt.show()
#plot histogram
histogram(img,'HIstogram for Niger River Image',12000)
#%%
# Read the image
img2 = cv2.imread('niger.bmp',0)
#define function to plot figures
def image(imageplot,title1,grid="on"):
 plt.figure()
 plt.title(title1)
 plt.imshow(imageplot)

# stretched image Creates with zeros array to store
r_min_max= np.zeros((img2.shape[0],img2.shape[1]),dtype = 'uint8')

# apply Min-Max formulae using Loop over the image
for i in range(img2.shape[0]):
    for j in range(img2.shape[1]):
        r_min_max[i,j] = 255*(img2[i,j]-np.min(img2))/(np.max(img2)-np.min(img2))

# plot the Contrast stretched image
image(r_min_max,'The Contrast Stretched Niger River Image')
cv2.imshow('The Contrast Stretched Niger River Image',r_min_max)
#save the image
cv2.imwrite('The Contrast Stretched Niger River Image.png', r_min_max)
cv2.waitKey(0)
#%%
img3 = imread('The Contrast Stretched Niger River Image.png') #load the image
histogram(img3,'HIstogram for Niger River Image',12000)
```

```
53  """The code executing time is calculated using
54   difference of the start time and end time"""
55  print (datetime.now()-starttime)
```

- A far object was captured a using Samsung galaxy s10+ phone without using HDR and automatic white balance and intensity histograms for all three colour channels were obtained.

```
 1 # -*- coding: utf-8 -*-
 2 """
 3 Created on Sun Oct 10 07:43:30 2021
 4
 5 @author: umeshika
 6 """
 7 #import the libraries
 8 from datetime import datetime
 9 import numpy as np
10 from skimage.io import imread
11 import matplotlib.pyplot as plt
12
13 #The variable is create to get the starting time when the code is run
14 starttime=datetime.now()
15 img = imread('far.jpg') #load the image
16 #define variables for Separate three RGB layers
17 r=img[:,:,0]#R layer
18 g=img[:,:,1]#G layer
19 b=img[:,:,2]#B layer
20 #%%define hist function for ploting histogram
21 def histogram(image,title1,ylim,colour):
22             plt.hist(image.ravel(),bins=np.arange(255), color=colour)
23             plt.ylim([0,ylim])
24             plt.xlim([-2,260])
25             plt.title(title1)
26             plt.xlabel("Pixels Intesity")
27             plt.ylabel("Pixels Intesity Count")
28             plt.show()
29 #plot histogram
30 histogram(r,'HIstogram for R channel far object image',800000,'red')
31 histogram(g,'HIstogram for G channel far object image',800000,'green')
32 histogram(b,'HIstogram for B channel far object image',800000,'blue')
33
34
35 """The code executing time is calculated using
36  difference of the start time and end time"""
37 print (datetime.now()-starttime)
```

- The ‘niger.bmp’ image was loaded and modified intensities of each pixel by using the following function and showed original and modified images along with the intensity histograms.



3.4 Exercise 4:

- The photograph of object with a measuring ruler were taken in the field of view.
- The pixel distance of two points in ruler was taken and mapped that to the distance in real-world distance (in cm or mm) by using function of ginput in matplotlib.
- The certain length of the object was estimated in the image in pixels and converted those lengths into real-world values and estimated the uncertainty of the measurements.
- The microscopic image of pollen of Arabis flower diameter was taken with the uncertainty.

3.5 Exercise 5:

- The image of ‘toys.jpg’ was loaded to variable img1 and converted that into a grayscale image after that it was saved in variable img_gray then plot the histogram of the img_gray.
- A binary image can be obtained using thresholding the grayscale image where the threshold value used here is 0.5 and threshold grayscale image to get a binary image consider as img_bin =img_gray>0.5
- After that running this threshold value in an array, it was found that a suitable threshold value with discriminates toys from the background.
- Alternatively, the threshold value with discriminates toys from the background could be taken using different method availane in skimage from skimage.filters.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Oct  9 13:59:50 2021
4
5 @author: Umeshika
6 """
7 #import the libraries
8 from datetime import datetime
9 from skimage.io import imread
```

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import  color
from skimage.filters import threshold_otsu,try_all_threshold

#The variable is create to get the starting time when the code is run
starttime=datetime.now()
img1 = imread('toys1.jpg') #load the image

#%%define the function of imshow1
def imshow1(image, title1, cmap_type='gray'):
          imggray=color.rgb2gray(image)#convert data to gray color
          plt.imshow(imggray, cmap=cmap_type)#take the gray scale image
          plt.title(title1)#label the title
          plt.axis('off')#off the axis
          plt.figure(1) #plot figure
          fig = plt.figure(1)
          plt.show()
          return fig
#gray color image is taken using imshow1 function
#imshow1(img1,"The Toys Image",cmap_type='gray')
#save the gray color image different location
fig=imshow1(img1,"The Toys Gray Image",cmap_type='gray')
fig.savefig('img_gray1.png')

img2 = imread('img_gray1.png') #load the image
plt.hist(img2.ravel(),bins=np.arange(255))
plt.ylim([0,12000])
plt.xlim([-2,260])
plt.title('The histogram for grey scale toy image')
plt.xlabel("Pixels Intesity")
plt.ylabel("Pixels Intesity Count")
plt.show()
#%%
#the grayscale iamge is taken using rgb2gray
toyimage_gray = color.rgb2gray(img1)

# take threshold value with otsu
threshimage = threshold_otsu(toyimage_gray)

#the thresholding image is taken
binary =toyimage_gray > 0.5
#plot the binary image
plt.imshow(binary,cmap='gray')
plt.title("The Toys Binary Image")
plt.axis('off')
plt.show()
print(threshimage)
#%% using different method availane in skimage to get binary image
fig1, ax = try_all_threshold(toyimage_gray, verbose=False)
plt.show()

```

```
62
63 """The code executing time is calculated using
64  difference of the start time and end time"""
65 print (datetime.now()-starttime)
```

- Above mentioned steps were repeated for book_page1.png and local thresholding was used to fix book_page1.png.

```
 1 # -*- coding: utf-8 -*-
 2 """
 3 Created on Sat Oct  9 16:50:22 2021
 4
 5 @author: umeshika
 6 """
 7 #import the libraries
 8 from datetime import datetime
 9 from skimage.io import imread
10 import matplotlib.pyplot as plt
11 from skimage import  color
12 from skimage.filters import
13 threshold_otsu,try_all_threshold,threshold_local
14
15 #The variable is create to get the starting time when the code is run
16 starttime=datetime.now()
17 img = imread('book_page1.png') #load the image
18 #the grayscale iamge is taken using rgb2gray
19 image_gray = color.rgb2gray(img)
20
21
22 #the thresholding image is taken
23 binary =image_gray >0.5
24 #plot the binary image
25 plt.imshow(binary,cmap='gray')
26 plt.title("The Book Page Binary Image")
27 plt.axis('off')
28 plt.show()
29
30 #%% using different method availane in skimage to get binary image
31 fig1, ax = try_all_threshold(image_gray, verbose=False)
32 plt.show()
33 #%%
34 # take the value of optimal otsu global thresh
35 global_thresh = threshold_otsu(image_gray)
36
37 # take binary image using global thresholding
38 binary_global = image_gray > global_thresh
39
40 #plot the binary image
41 plt.imshow(binary_global,cmap='gray')
42 plt.title("The Book Page Binary Image with global thresholding")
```

```
plt.axis('off')
plt.show()



# Set the block size to 35
block_size = 35
# take the value of optimal local thresholding
local_thresh = threshold_local(image_gray, block_size, offset=0.1)

#  take binary image using local thresholding
binary_local = image_gray > local_thresh

#plot the binary image
plt.imshow(binary_local,'Local thresholding')
plt.title("The Book Page Binary Image with lobal thresholding")
plt.axis('off')
plt.show()

print(global_thresh)
print(local_thresh)
"""The code executing time is calculated using
 difference of the start time and end time"""
  print (datetime.now()-starttime)
```

4 RESULTS AND ANALYSIS

4.1 Exercise 01

Figure 4.1-1 The grey image of face

Figure 4.1-2 The second grey image of face which was taken using above grey image

Using ".shape" function, the number of dimensions in the current array was given as tuple of numbers to represent because RGB channel values that are 8 bit and between 0 to 255. Which is three-dimensional matrix. The RGB image has dimension of (795, 643, 3) and grey image has

dimension of (288, 432, 4). The RGB figure dimension has given in 795x643 pixel and it was same as ".shape" function output. And grey image dimension values are less than the RGB RGB image has dimension.

<Figure size 432x288 with 0 Axes>

(795, 643, 3)- RGB image of face

(288, 432, 4)- Grey Image

Code execution time= 02.585338 s

4.2 Exercise 02

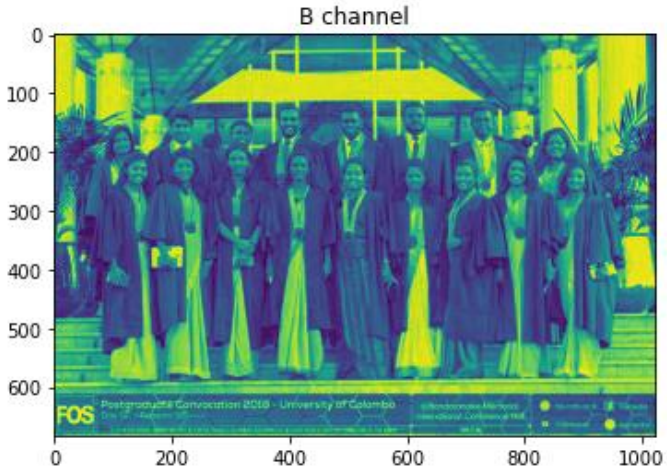


Figure 4.2-1 The image of blue channel for convocation image

This figure shows the blue channel of the convocation image and light colors represent more yellowish color and black color represent dark blue. Therefore, this blue channel figure has not show other colors it totally has yellowish blue color combination.

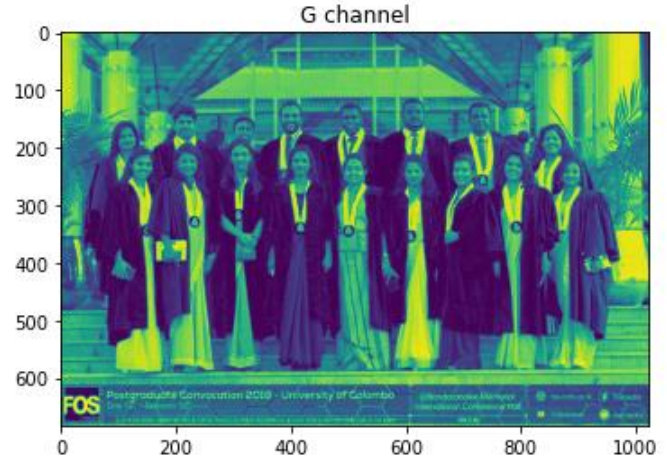


Figure 4.2-2 The image of green channel for convocation image

This figure shows the green channel of the convocation image and light colors represent more green color and black color represent without changing it. Therefore, this green channel figure has not show other colors it totally has yellowish green and black color combination.

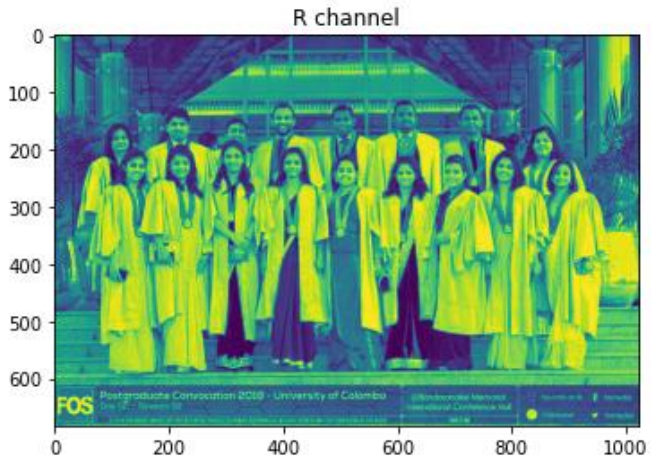


Figure 4.2-3 The image of red channel for convocation image

This figure shows the red channel of the convocation image and light colors represents more green color , red color represent as dark blue mixed black color and black color represents the yellowish green. Therefore, this red channel figure has not show other colors it totally has yellowish green and red color combination.

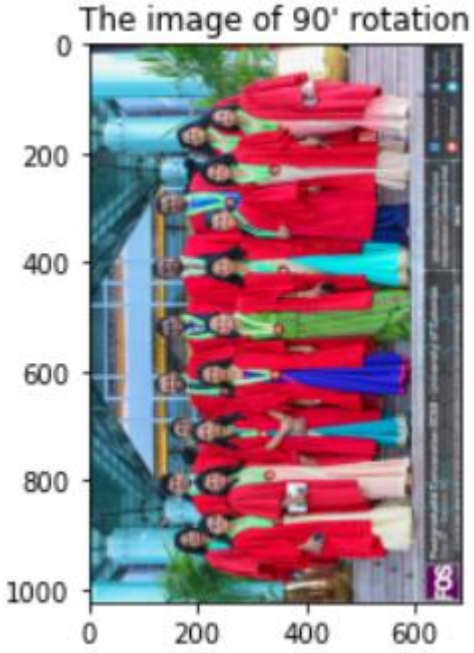


Figure 4.2-4 The image with 90’ rotation for convocation image

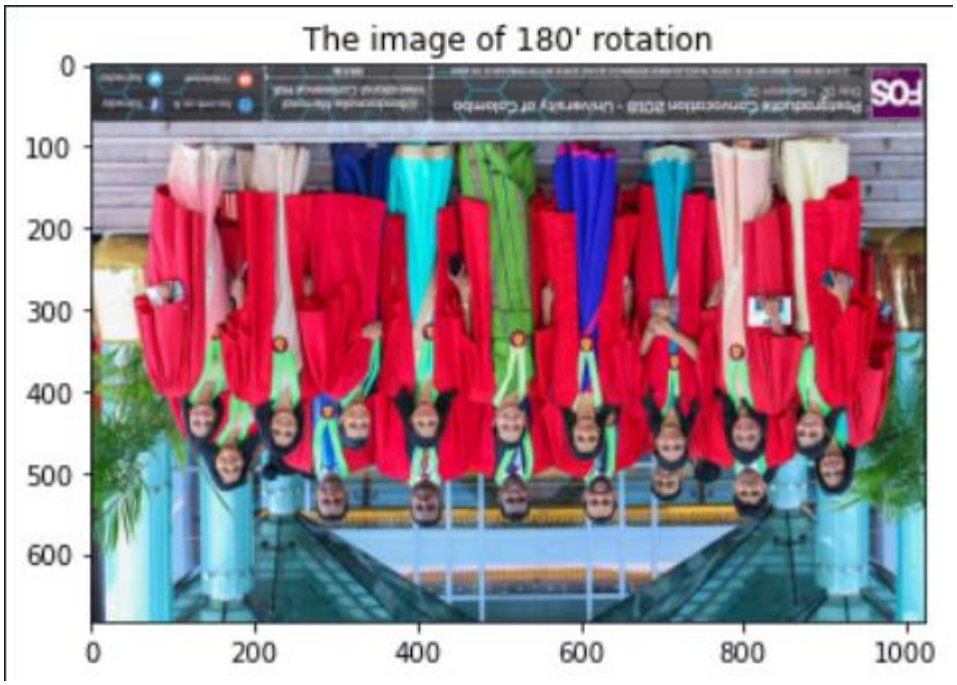


Figure 4.2-5 The image with 180’ rotation for convocation image

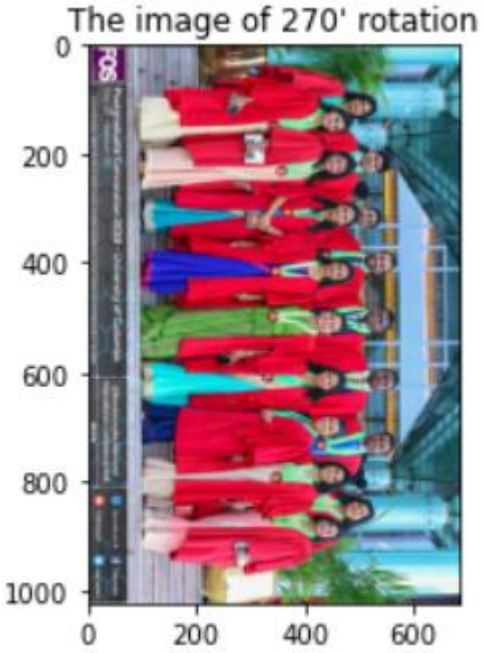


Figure 4.2-6 The image with 270’ rotation for convocation image

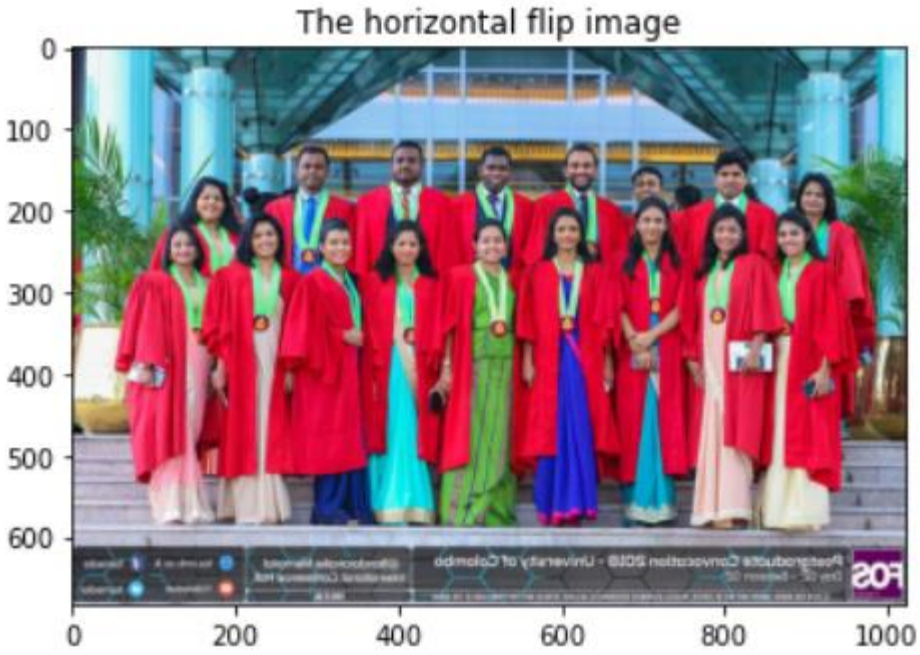


Figure 4.2-7 The image with the horizontal flip for convocation image

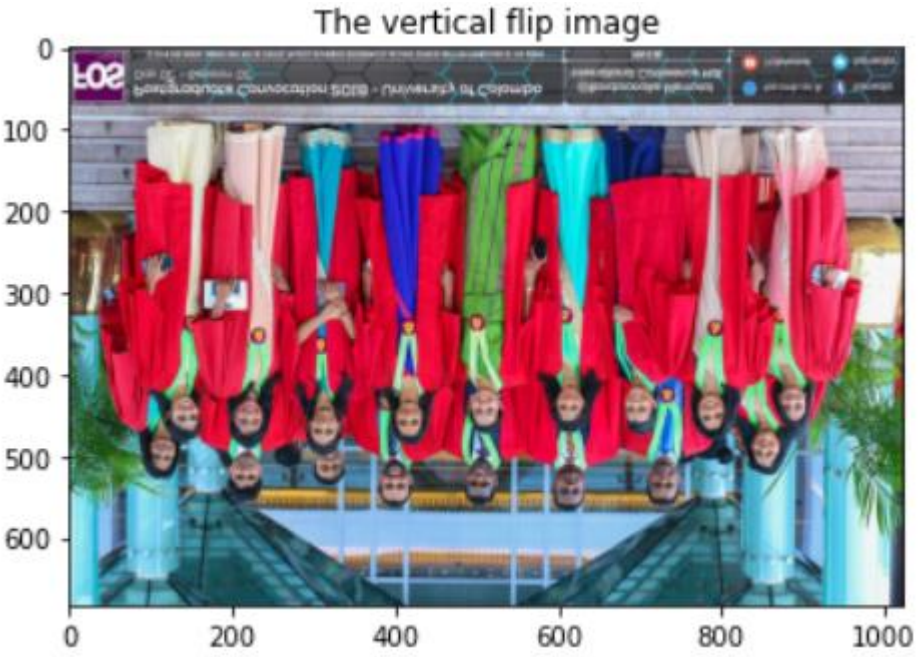


Figure 4.2-8 The image with the vertical flip for convocation image

Code execution time= 0.180477 s

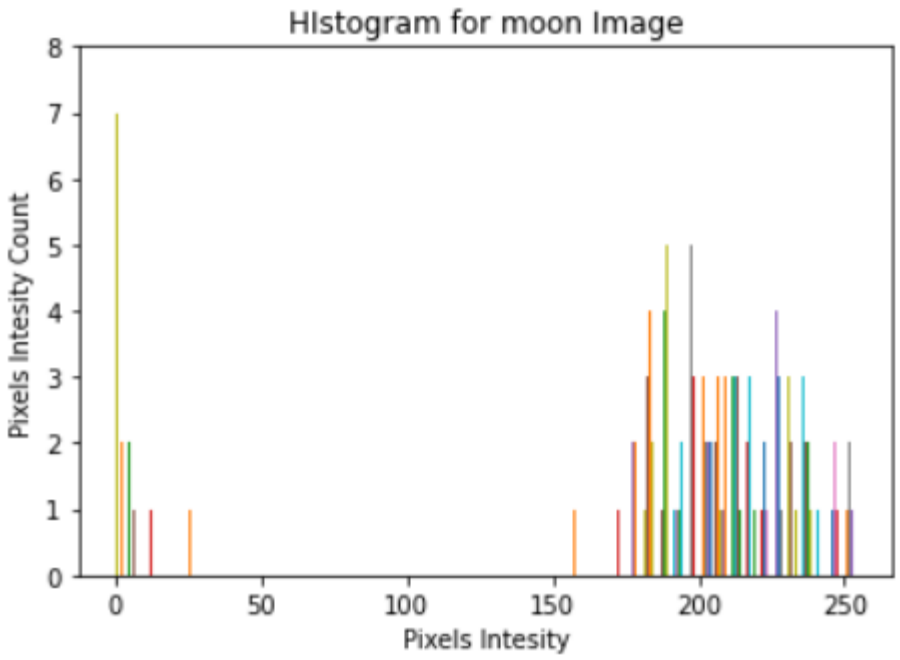


Figure 4.2-9 The histogram for moon image

This histogram is looked the grayscale image because high concentration of dark shades peaks on the left side that means large number of pixels are in lower pixel intensity range which is color black. And the moon image is moving more grey scale ranged image because the concentrations of white pixels increase in the right-hand side.

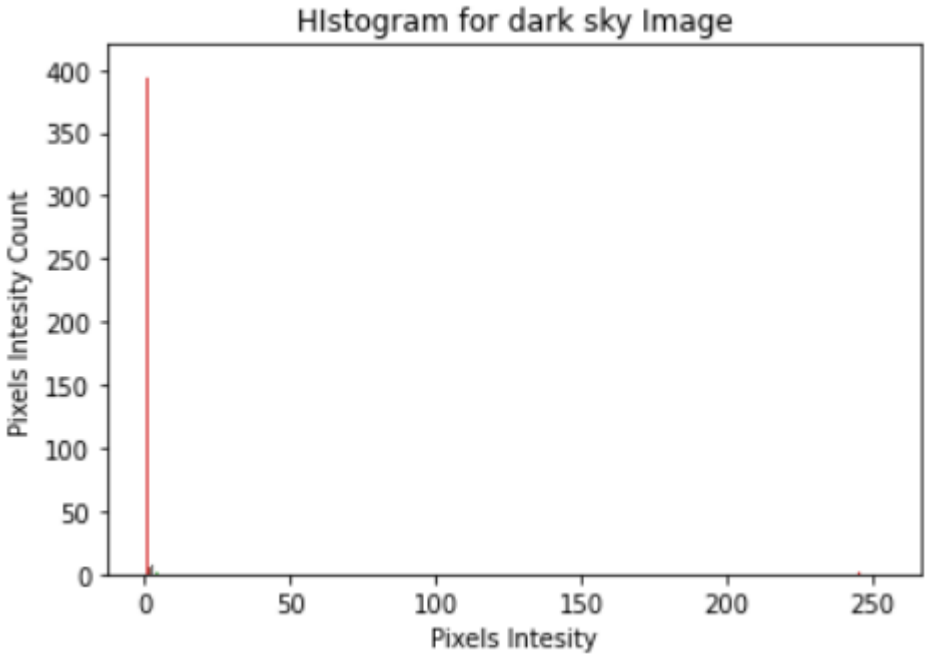


Figure 4.2-10 The histogram for dark sky image

This histogram is looked the binary image because only two peaks are shown in here and high concentration of dark black peaks on the left side that means large number of pixels are in lower pixel intensity range which is color black. And the dark sky image is not moving more grey scale ranged image because the concentrations of white pixels peak is very small in the right-hand side.

Code execution time= 03 minutes 23.226764 seconds

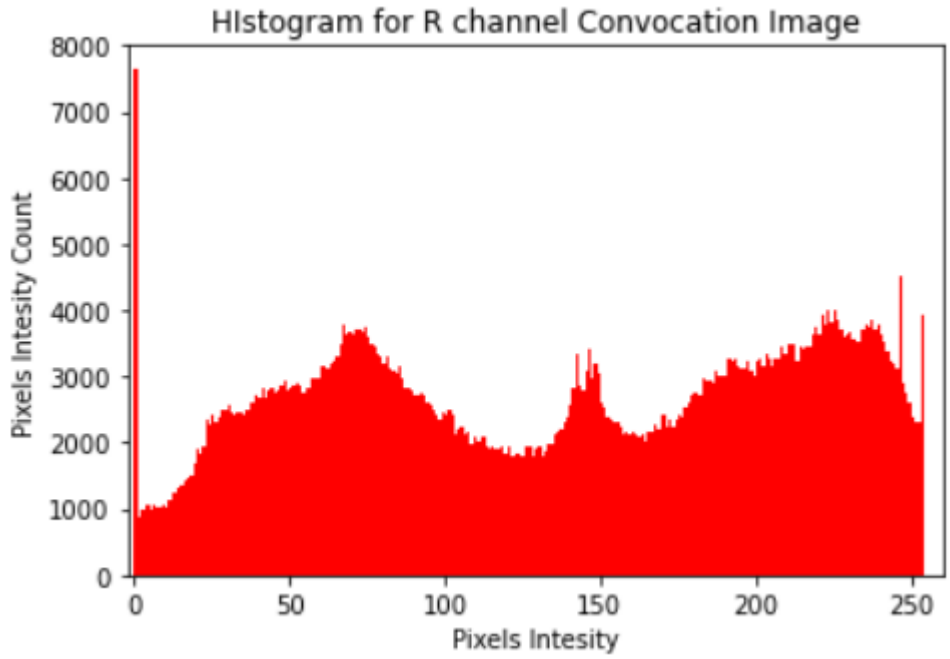


Figure 4.2-11 The histogram of R channel for convocation image

This histogram is looked like the equally distributing pixel intensity in the R channel of convocation image because almost equal peaks range are spread in range of 0 to 255 and high concentration of 0 value peaks on the left side that means large number of pixels are in lower pixel intensity range which is color black. And the convocation of R channel is not moving more grey scale ranged image because the concentrations of white pixels peak also very similar to the other ranges peak.

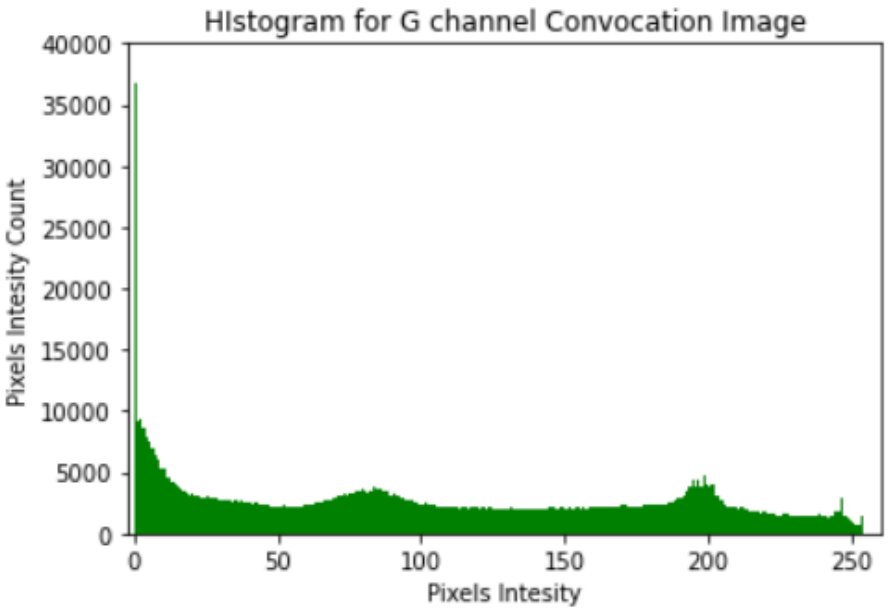


Figure 4.2-12 The histogram of G channel for convocation image

This histogram is looked like the equally distributing pixel intensity in the G channel of convocation image because almost equal peaks range are spread in range of 0 to 255 and high concentration of 0 value peaks on the left side that means large number of pixels are in lower pixel intensity range which is color black. And the convocation of G channel is not moving more grey scale ranged image because the concentrations of white pixels peak also very similar to the other ranges peak.

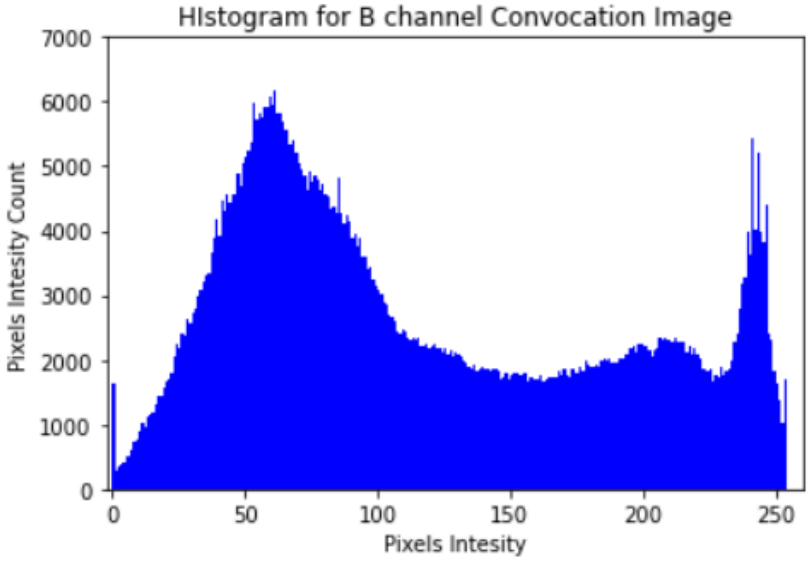
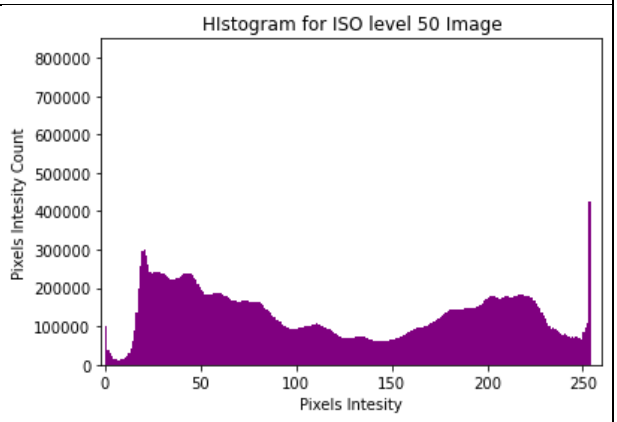
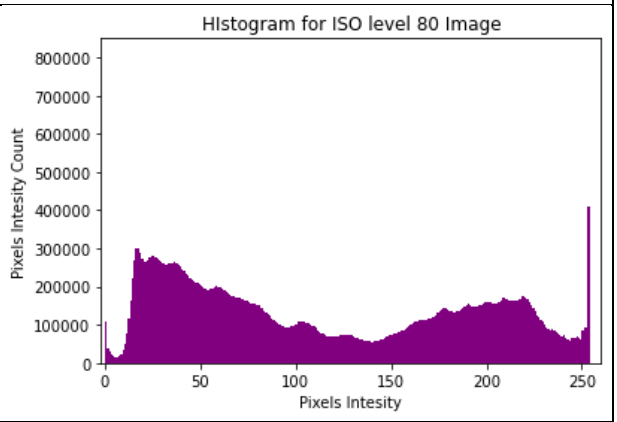
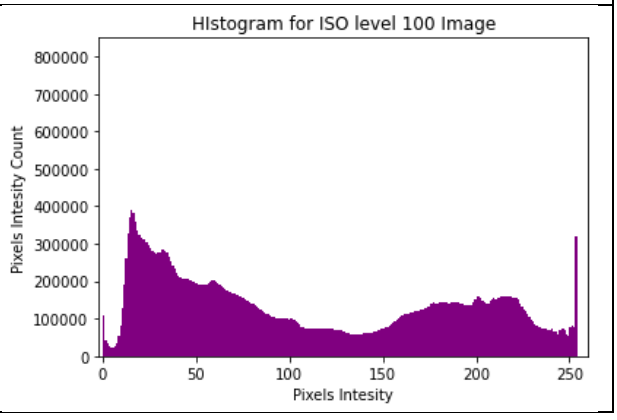


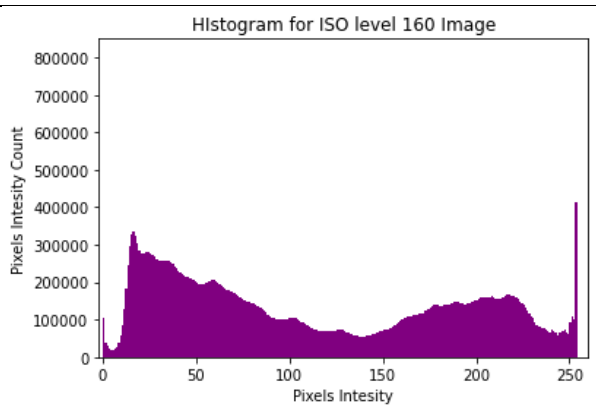
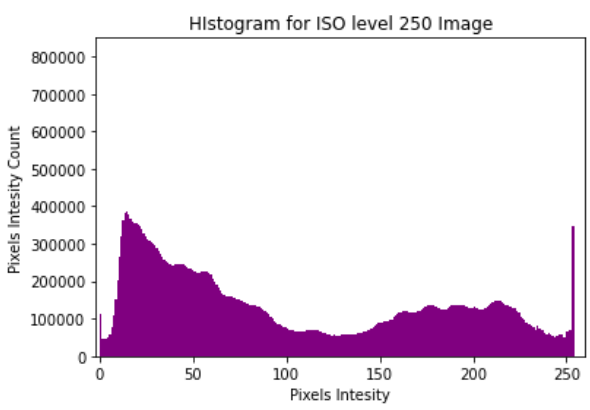
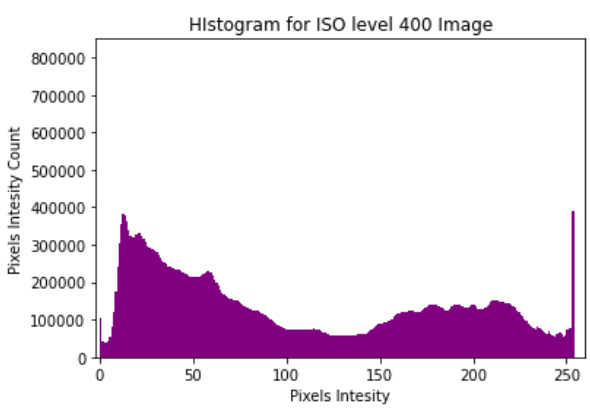
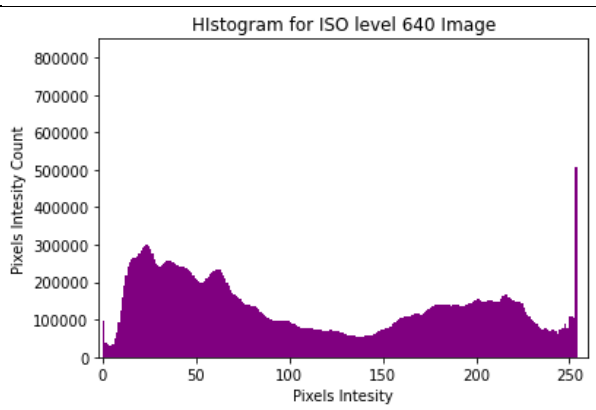
Figure 4.2-13 The histogram of B channel for convocation image

This histogram is looked like the distributing pixel intensity in the B channel of convocation image because the equal peaks range are spread in range of 100 to 200 and high concentration and 0 value peaks on the left side has small value that means minimum number of pixels are in lower pixel intensity range which is color black. And the convocation of B channel is not moving more grey scale ranged image because the concentrations of white pixels peak also almost similar to the 50-100 pixel intensity ranges peak.

Code execution time= 01.323512 s

4.3 Exercise 03

ISO level	Photograph	Histogram
50		
80		
100		

160		
250		
400		
640		

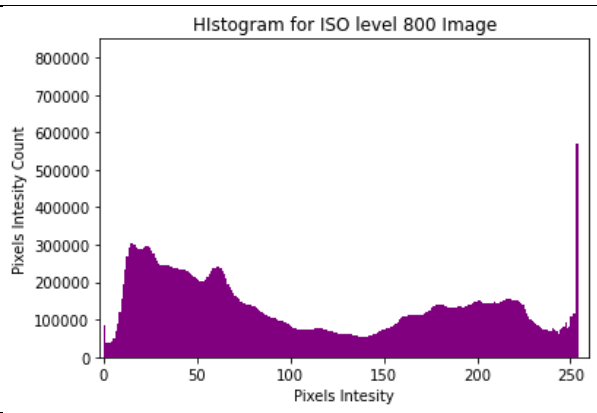
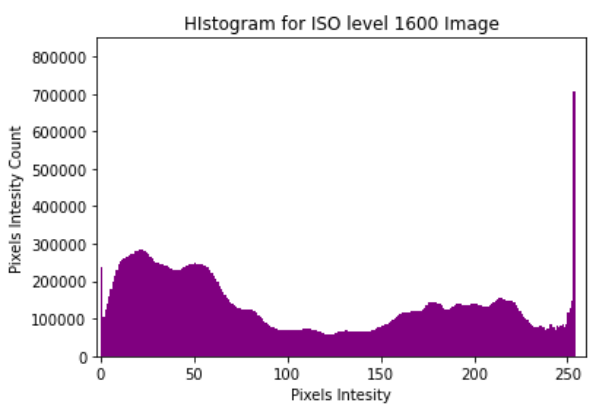
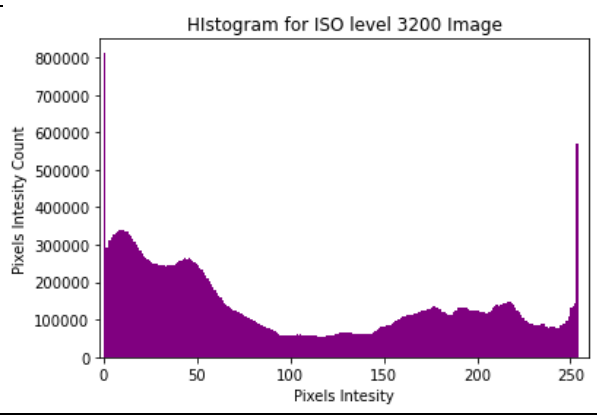
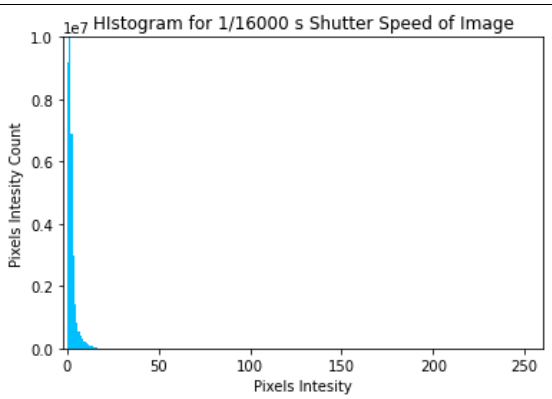
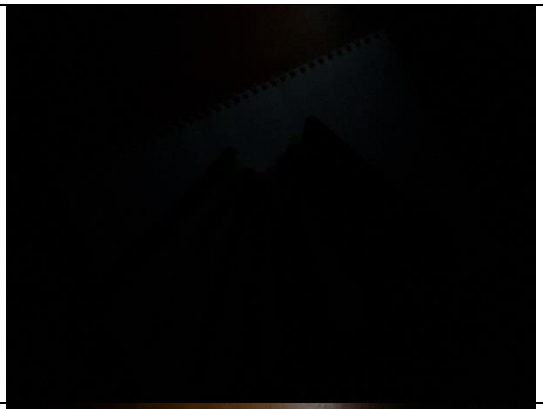
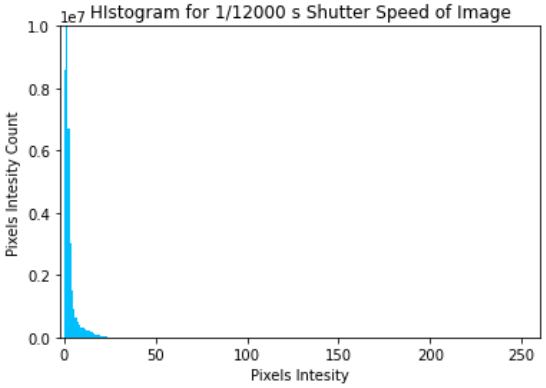
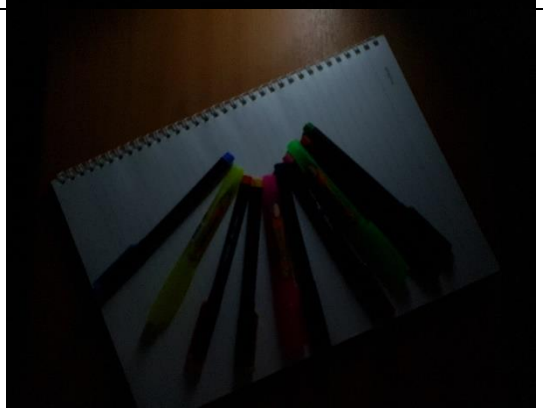
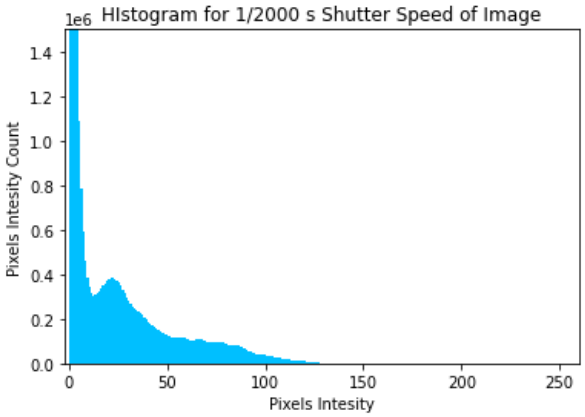
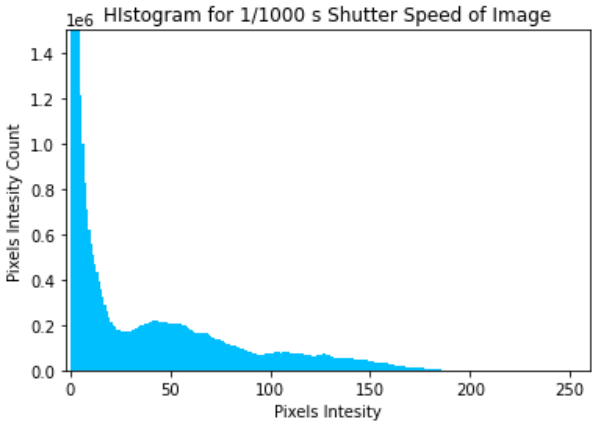
800		
1600		
3200		

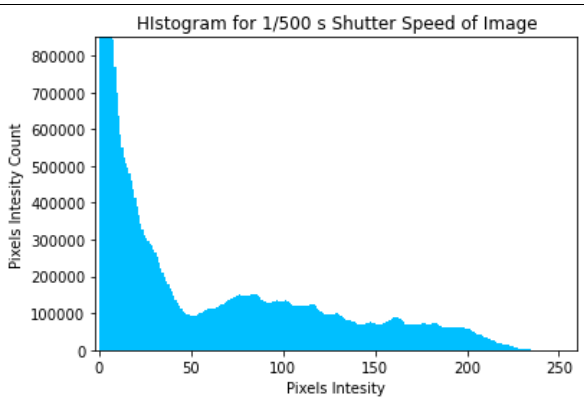
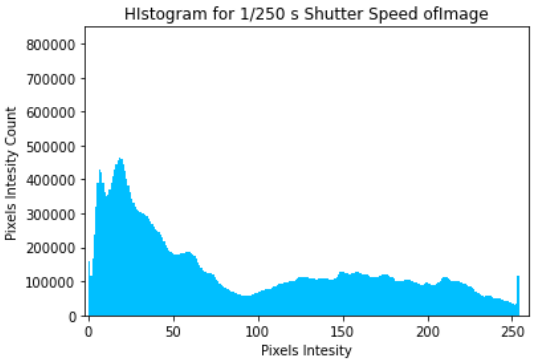
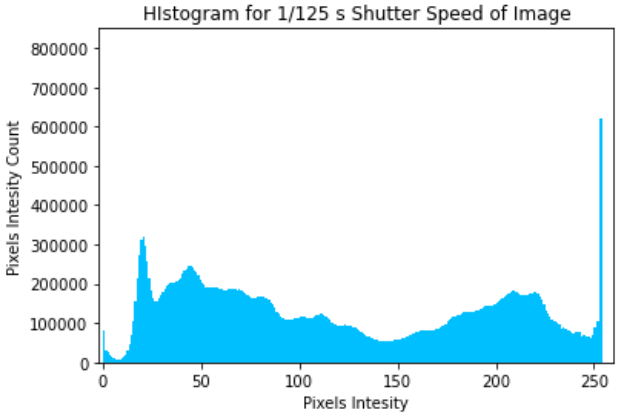
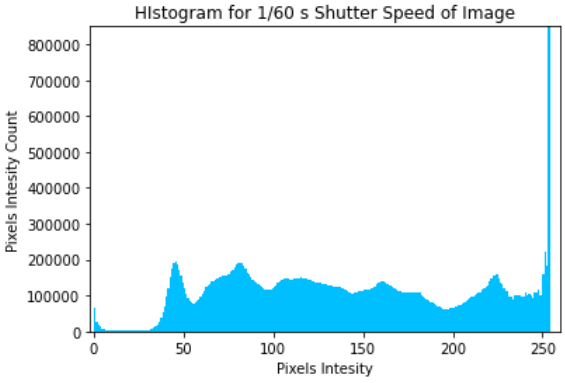
Figure 4.3-1 The figure panel for photograph with 10 different ISO levels and its histogram

This figure panel shows that the distributing pixel intensity in the 10 different ISO levels images and its histogram therefore the low ISO level of images' histogram represents almost equal peaks range are spread in range of 0 to 255.

When ISO level is increasing high concentration of pixel value peaks on the right side which is 250 that means large number of pixels are in large pixel intensity range which is color white. And the 10 different ISO levels is increasing and it would be more grey scale ranged image.

Shutter Speed (s)	Photograph	Histogram

$\frac{1}{16000}$		
$\frac{1}{12000}$		
$\frac{1}{2000}$		
$\frac{1}{1000}$		

$\frac{1}{500}$		
$\frac{1}{250}$		
$\frac{1}{125}$		
$\frac{1}{60}$		

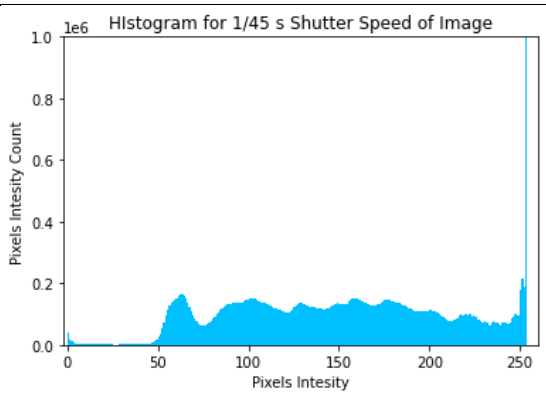
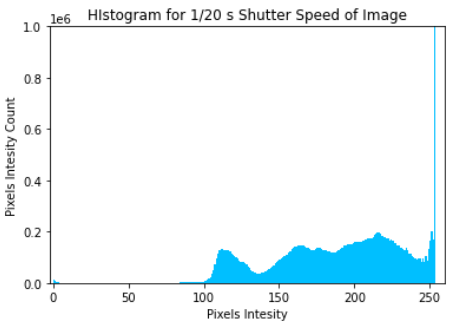
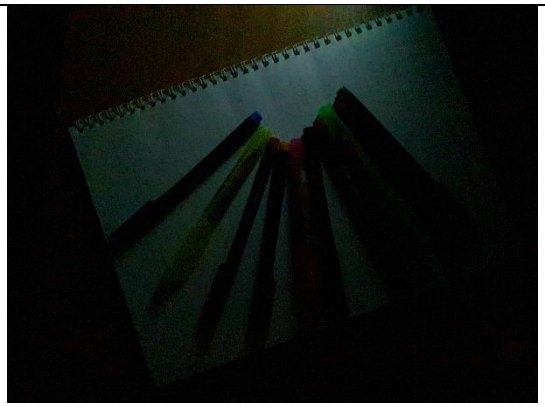
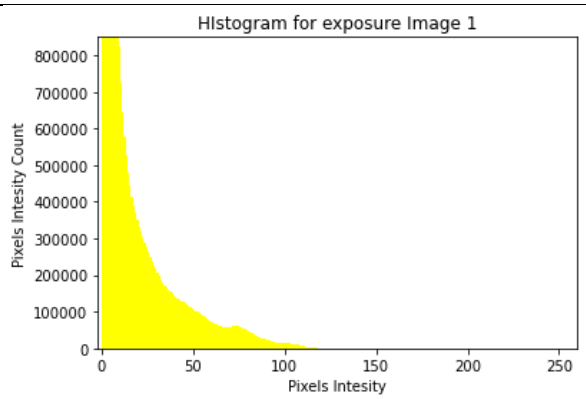
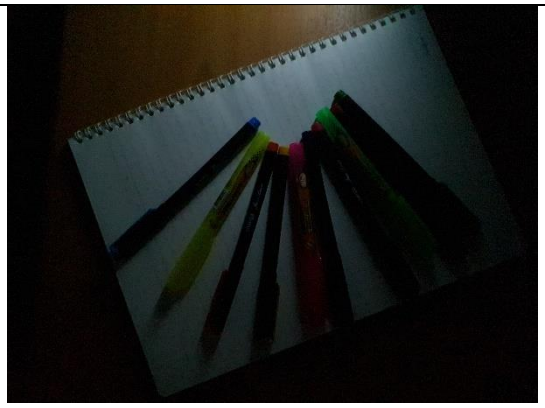
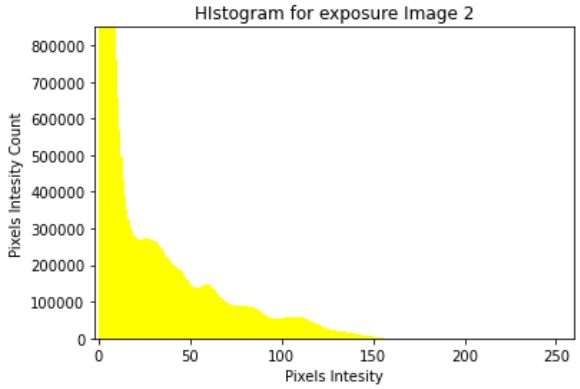
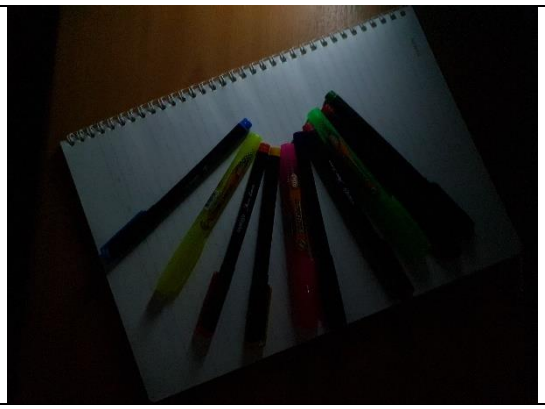
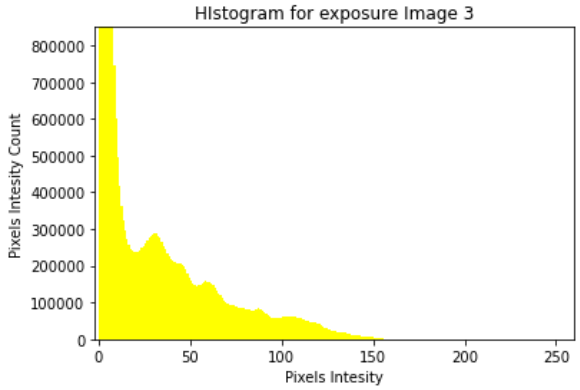
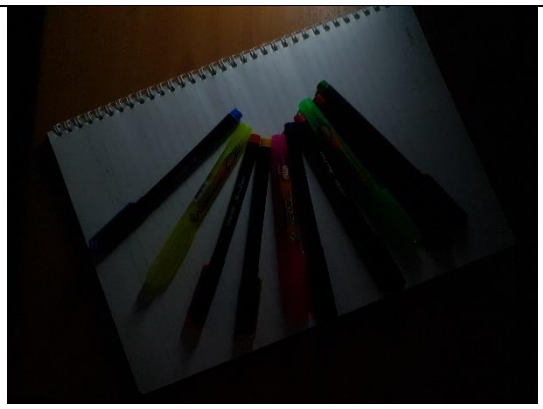
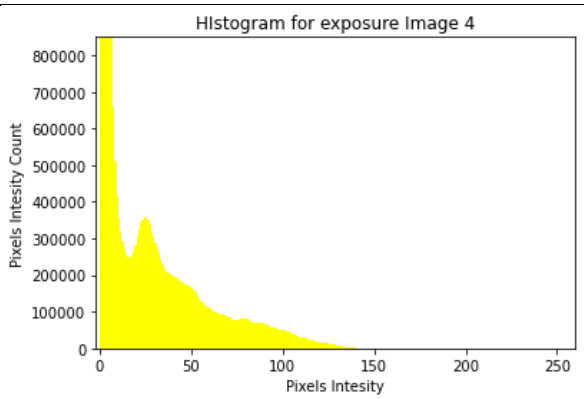
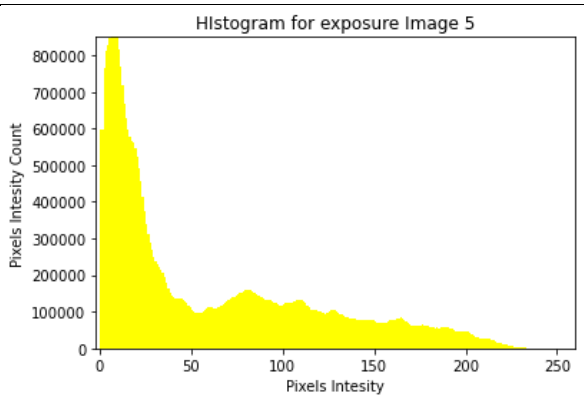
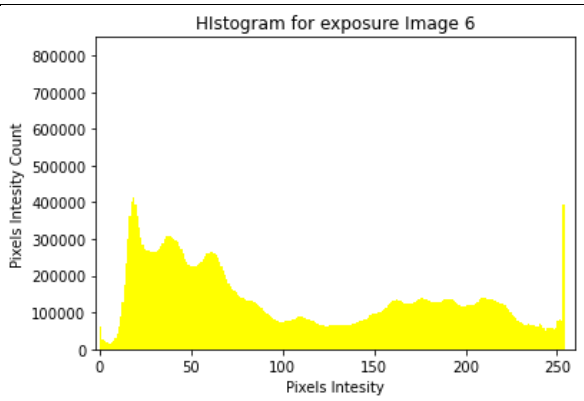
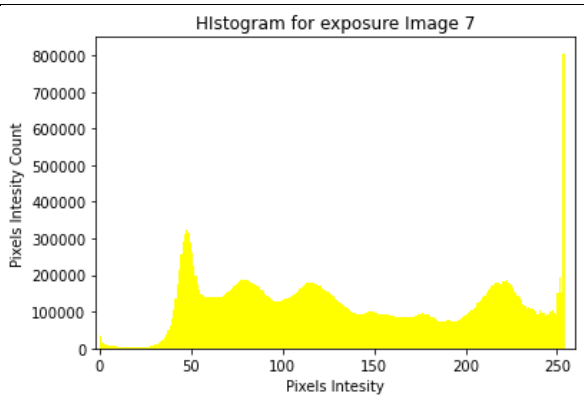
$\frac{1}{45}$		
$\frac{1}{20}$		

Figure 4.3-2 The figure panel for photograph with 10 different shutter speeds and its histogram

This figure panel shows that the distributing pixel intensity in the 10 different shutter speeds images and its histogram therefore the low shutter speeds of images' histogram represents almost the peaks are lied the 0 pixel intensity value which is black

When shutter speeds level is increasing high concentration of pixel value peaks on the right side which is 250 that means large number of pixels are in large pixel intensity range which is color white. And the 10 different shutter speeds is increasing and it would be more white scale ranged image.

Shutter Speed (s) And ISO level	Photograph	Histogram
$\frac{1}{12000}$ And 3200		
$\frac{1}{4000}$ And 1600		
$\frac{1}{2000}$ And 800		

$\frac{1}{1500}$ And 500		
$\frac{1}{350}$ And 400		
$\frac{1}{125}$ 400		
$\frac{1}{45}$ and 320		

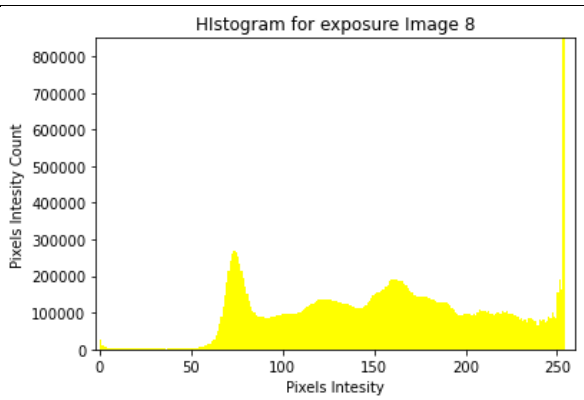
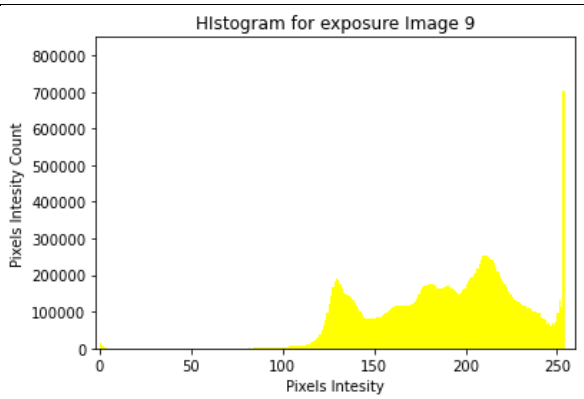
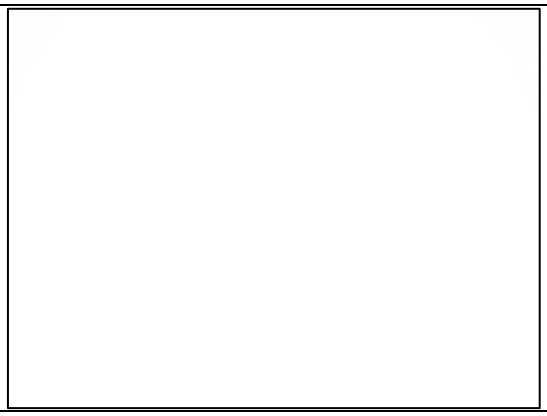
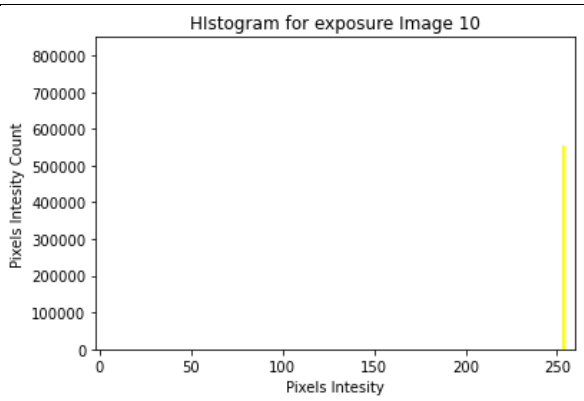
$\frac{1}{20}$ and 250		
$\frac{1}{6}$ and 100		
30 and 50		

Figure 4.3-3 The figure panel for photograph with 10 different exposure and its histogram

This figure panel shows that the distributing pixel intensity in the 10 different exposure images and its histogram therefore the low exposure of images' histogram represents almost the peaks are lied the 0 pixel intensity value which is black

When exposure level is increasing high concentration of pixel value peaks move to the right side and finally which is in 250 that means the highest exposure image histogram has large number of pixels are in large pixel intensity range which is color white. And the 10 different exposure is increasing and it would be more white scale ranged image.

Code execution time for 10 different ISO level and its histogram = 15.573325 s

Code execution time for 10 different Shutter Speed and its histogram = 17.440733 s

Code execution time for 10 different exposure and its histogram = 15.472530 s

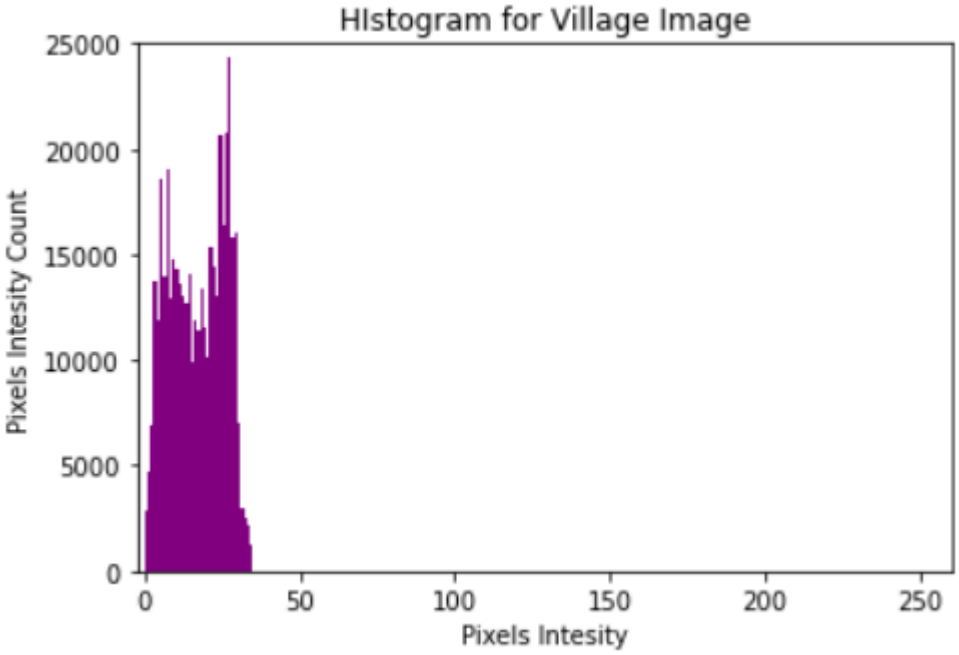


Figure 4.3-4 The histogram for village image

This histogram shows that the distributing pixel intensity of the village images it represents the peaks are lied the 0 to 40 pixel intensity value which is black and dark color range. This image is an old image which was taken by black and white camera therefore it only have more drak colors.

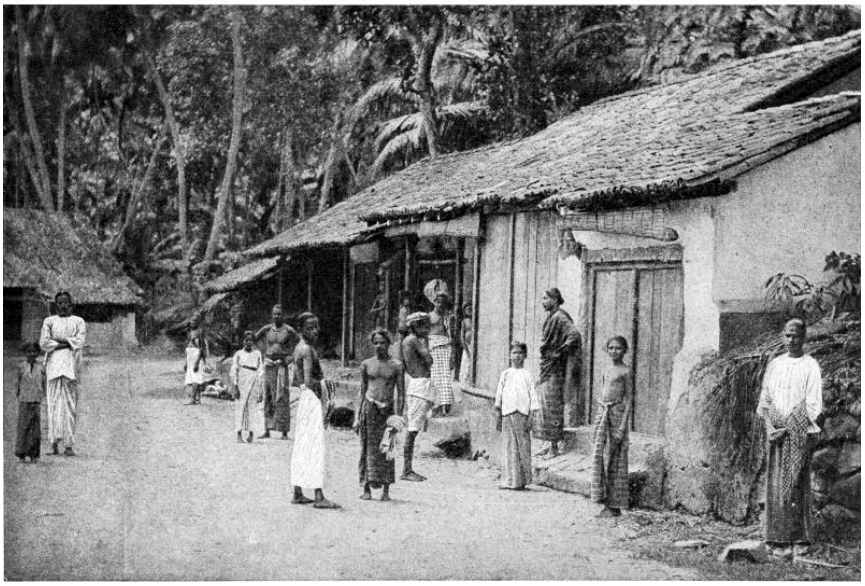


Figure 4.3-5 The Contrast Stretched village image

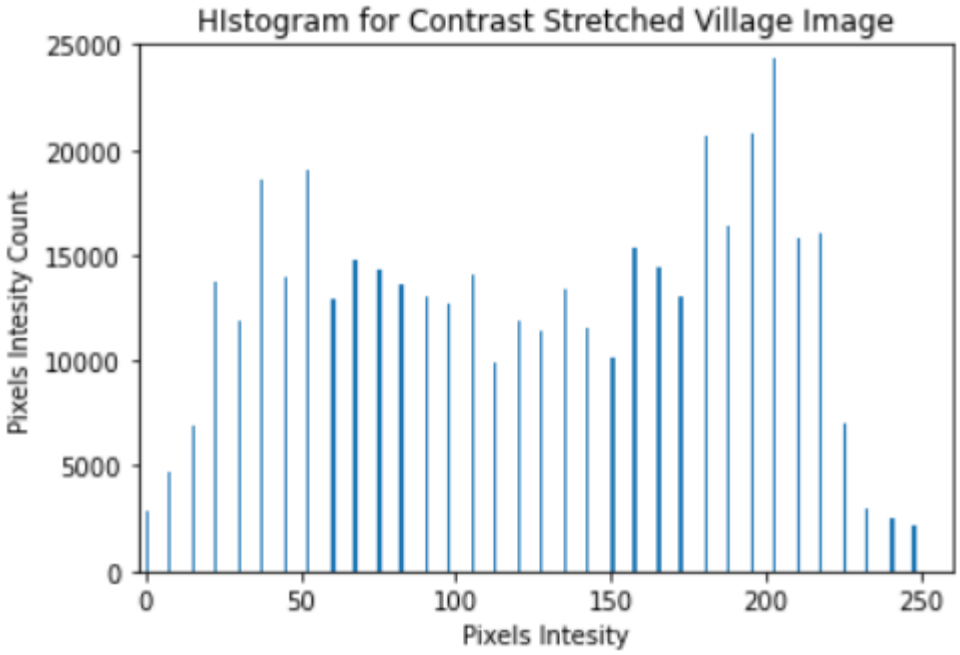


Figure 4.3-6 The histogram for Contrast Stretched village image

Code execution time= 55.323512 s

Figure 4.3-7 The Contrast Stretched Niger River image

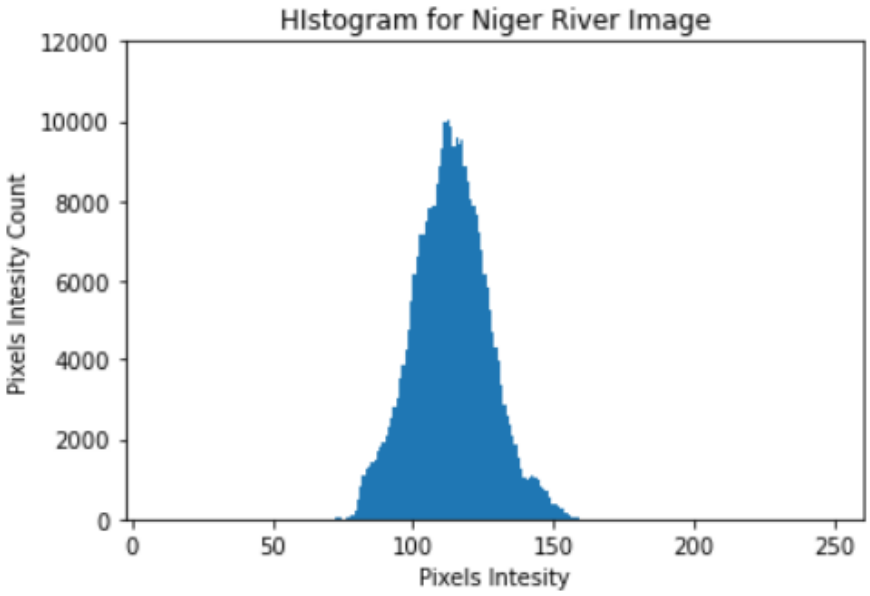


Figure 4.3-8 The histogram for Niger River Image

This histogram shows that the distributing pixel intensity of the niger river images it represents the peaks are lied the 90 to 150 pixel intensity value which middle color range in between black and white. And the image is not moving more grey scale ranged because the concentrations of white pixels has not in the right-hand side.

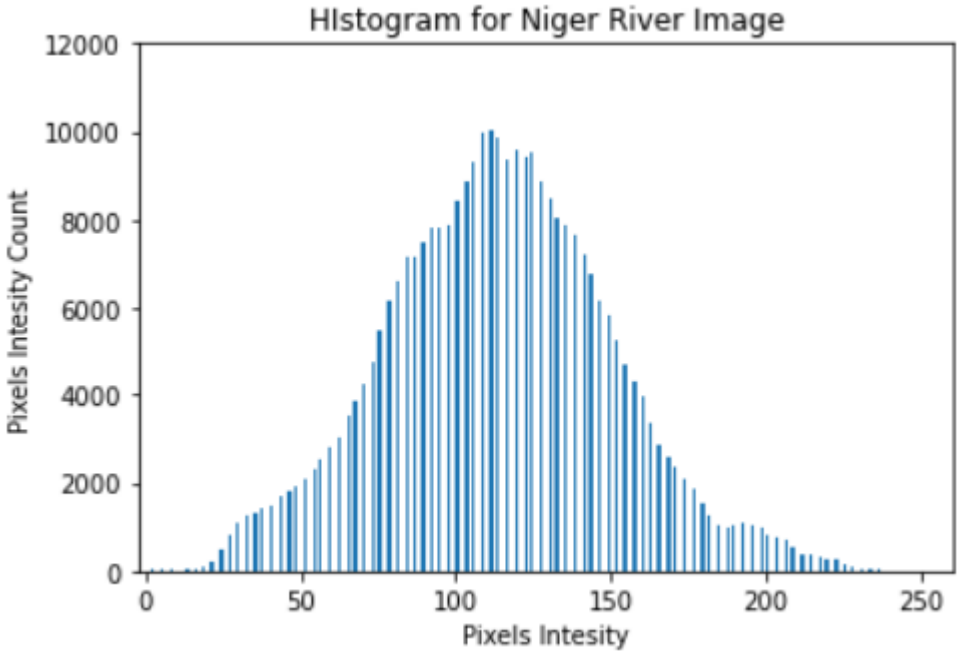


Figure 4.3-9 The histogram for Contrast Stretched Niger River image

Code execution time= 49.383582 s

Figure 4.3-10 The photograph for far object

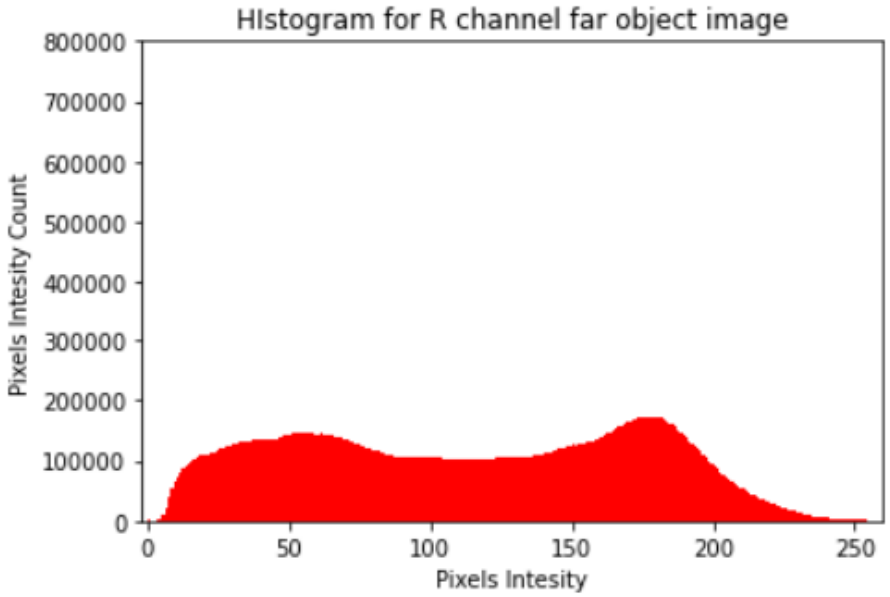


Figure 4.3-11 The histogram for R channel of far object image

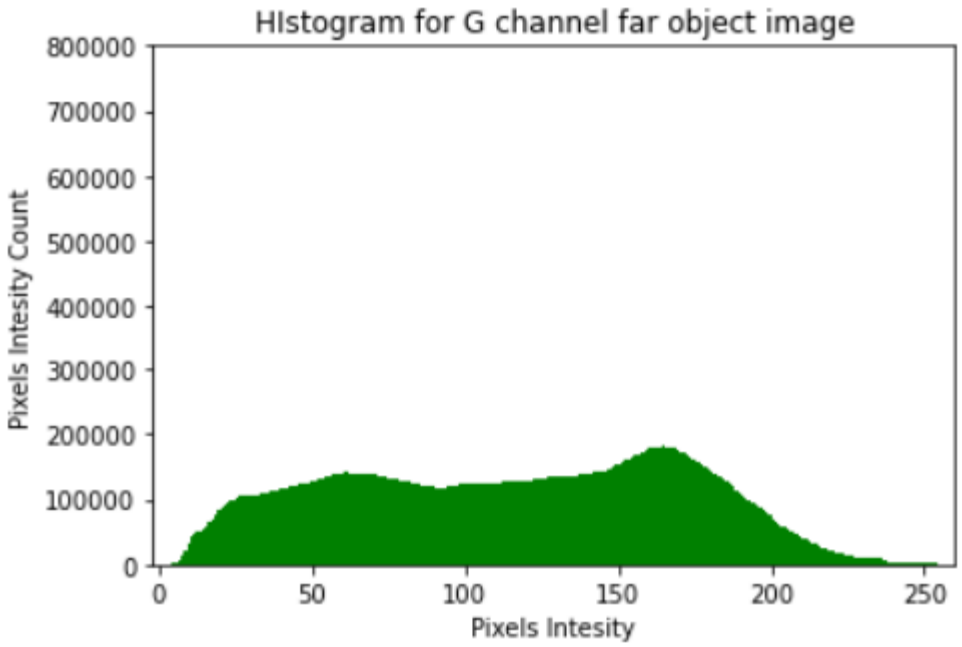


Figure 4.3-12 The histogram for G channel of far object image

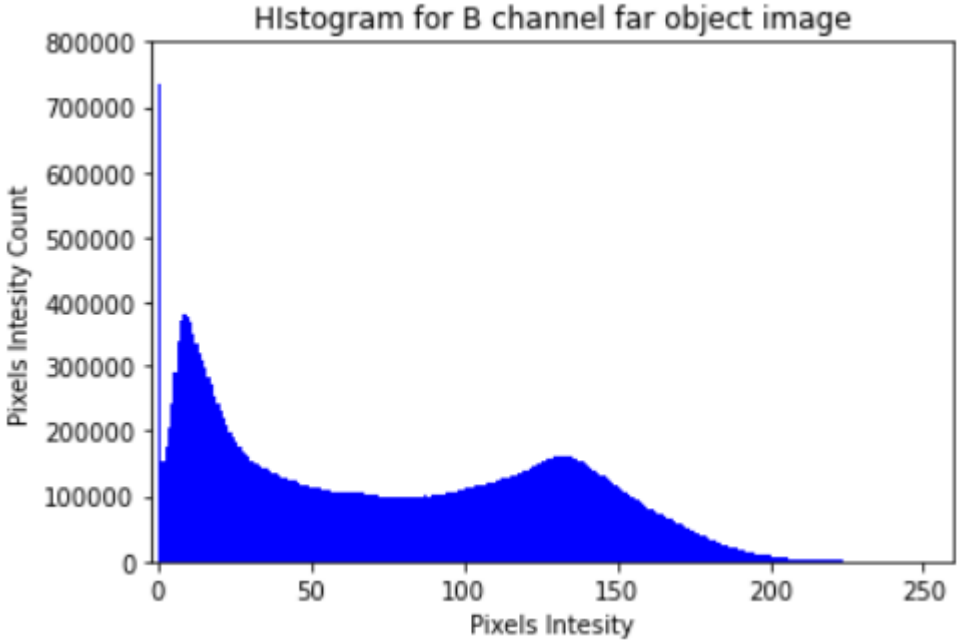


Figure 4.3-13 The histogram for B channel of far object image

Those three channel histograms have equally distributing data because of day light that is sunlight in Earth's atmosphere. According to the Rayleigh scattering, this far object image appears the yellowish to reddish light of the low Sun which is represent above histograms.

Code execution time= 0.393131 s

4.4 Exercise 04

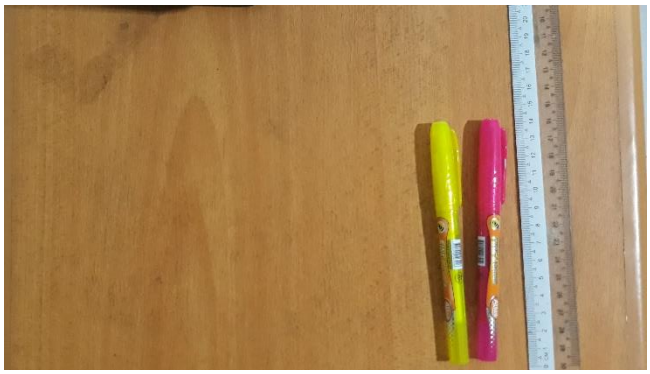


Figure 4.4-1 The photograph of object with a measuring ruler in the field of view

4.5 Exercise 05

Figure 4.5-1 The grey scale toy image

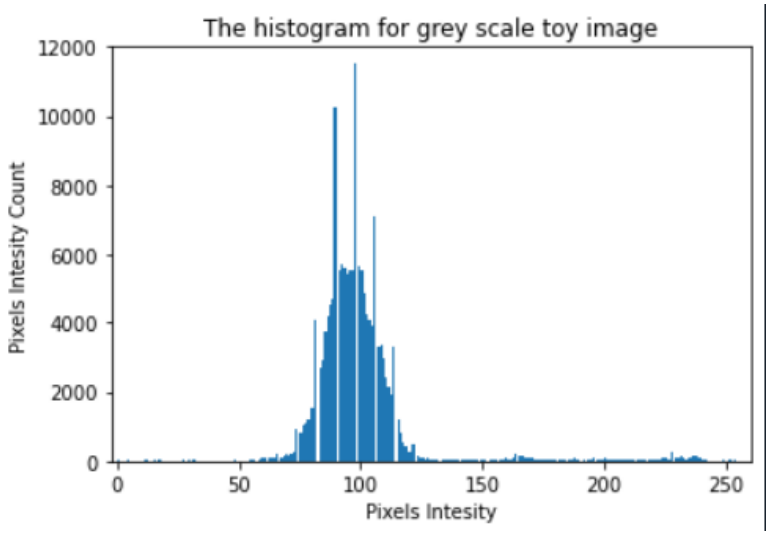


Figure 4.5-2 The histogram for grey scale toy image

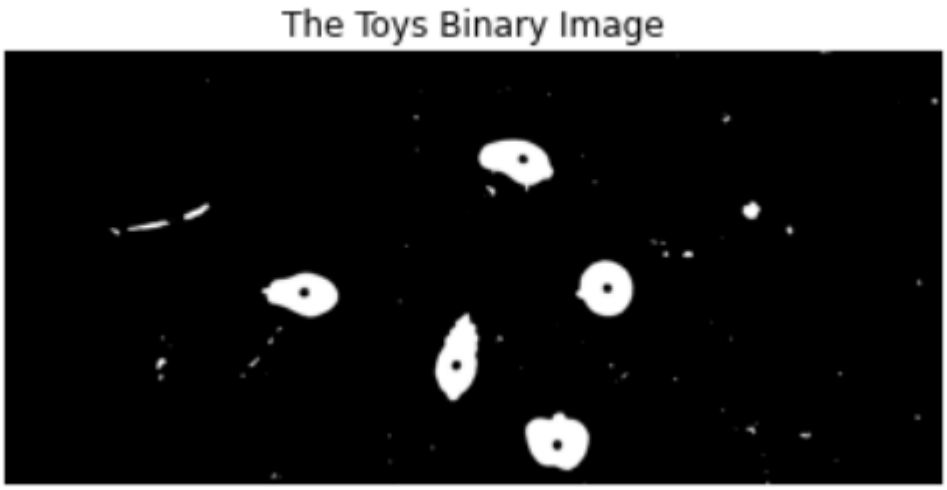


Figure 4.5-3 The binary image for toy image

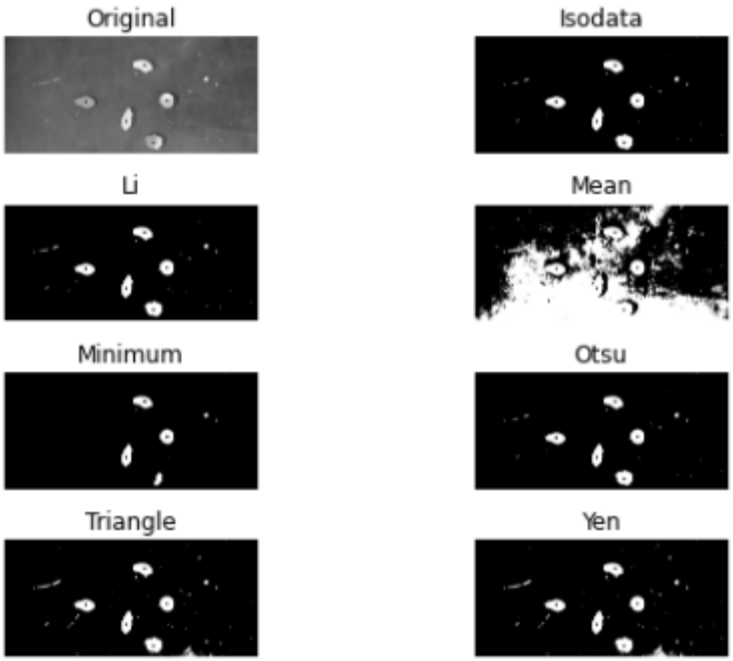


Figure 4.5-4 The toy images of different threshold filters by using skimage.filters

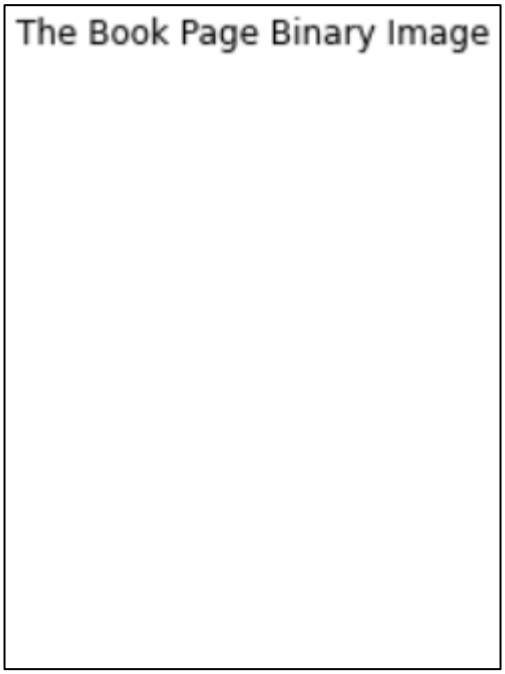


Figure 4.5-5 The binary image for book page

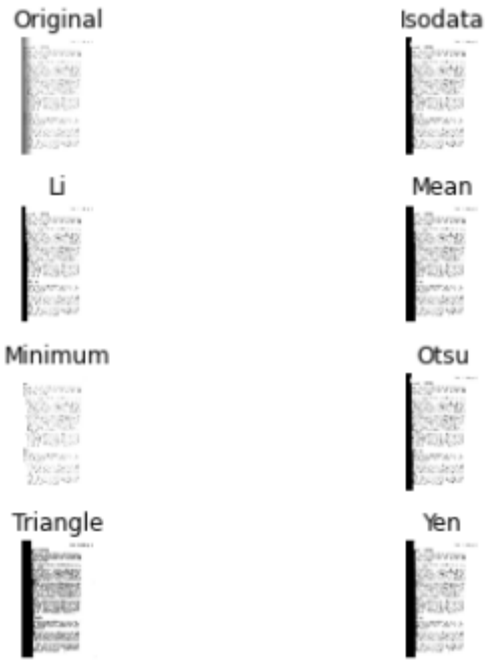


Figure 4.5-6 The book page images of different threshold filters by using skimage.filters

Figure 4.5-7 The binary image for book page with global thresholding

When comparing the binary image for book page and global thresholding book page, The binary image for book page with global thresholding is more visualized.

Code execution time= 0.43519 s

5 DISCUSSION AND CONCLUSION

In the exercise1 imshow function was used to take the results which could be obtain the grey scale figures without axis. The RGB image has 3 dimension and RGB channel values that are 8 bit and between 0 to 255. And grey image dimension has (288, 432, 4) value. The 8 bit image color intensities are always lied on 0 to 255 range. RGB image can be separated three layers.

In the second exercise was defined the histogram function to plot the histograms of RGB image separated three layers which could be observed that yellow intensity was absorbed blue light and light green and while the red intensity was ingested green light and somewhat blue and red, were passing to be reflected back. These results were determined that the essential shades of the best added substance shading framework are just red, green, and blue (RGB). Furthermore, flipping images with different directions, it was observed the different results.

In the third exercise results were observed that histogram peaks were moved to the black color intensity range to white color intensity when the ISO level, shutter speed and exposures were increased because camera could be open wider to the light.

Furthermore when Contrast Stretching, the range of pixel intensities has been spread all over the range which was effectively normal distributed the pixels in the image and therefore the brightness in the image has been increased the brightness.

And also the image is comprised of small spots of minimal light or all the more explicitly the pixels and what more intriguing is to see that those minuscule dots of minimal light are really numerous colors of minimal light of various tones which are only Red, Green, Blue channels.

In conclusion Vector designs are to some degree diverse strategy for putting away pictures that plans to keep away from pixel related issues. However, even vector pictures, eventually, are shown as a mosaic of pixels. The word pixel implies an image component. A straightforward way of depicting every pixel is utilizing a blend of three tones, in particular Red, Green, Blue. This is the thing that we call a RGB picture. Finally this report was concluded that image data could be processing using Pixel Intensity Histogram in Python Programming Langauge.

6 REFERENCES

[1] *GreatLearning Blog: Free Resources what Matters to shape your Career!. 2021. Introduction to Image Processing | What is Image Processing? - Great Learning. [online] Available at: <https://www.mygreatlearning.com/blog/introduction-to-image-processing-what-is-image-processing/> [Accessed 10 October 2021].*

[2] *How to Write an Essay. 2021. How to Write an Image Analysis Essay in 6 Easy Steps - How to Write an Essay. [online] Available at: <https://www.tutorphil.com/blog/how-to-write-analysis/ > [Accessed 10 October 2021].*

[3] *Medium. 2021. The History Of Image Processing Information Technology Essay. [online] Available at: <https://medium.com/@danilloleite82/the-history-of-image-processing-information-technology-essay-a30690c352f4 > [Accessed 10 October 2021].*

[4] *Codementor.io. 2021. Basic Image Processing In Python - Part 1 | Codementor. [online] Available at: <https://www.codementor.io/@innat_2k14/image-data-analysis-using-numpy-opencv-part-1-kfadbafx6 > [Accessed 10 October 2021].*

[5] *Datacarpentry.org. 2021. Creating Histograms – Image Processing with Python. [online] Available at: <https://datacarpentry.org/image-processing/05-creating-histograms/ > [Accessed 11 October 2021].*

[6] *Homepages.inf.ed.ac.uk. 2021. Image Analysis - Intensity Histogram. [online] Available at: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/histgram.htm > [Accessed 11 October 2021].*

[7] *Matplotlib.org. 2021. Image tutorial — Matplotlib 3.1.2 documentation. [online] Available at: < https://matplotlib.org/3.1.1/tutorials/introductory/images.html#sphx-glr-tutorials-introductory-images-py> [Accessed 11 October 2021].*

[8] *Encyclopedia Britannica. 2021. Rayleigh scattering | physics. [online] Available at: <https://www.britannica.com/science/Rayleigh-scattering > [Accessed 11 October 2021].*

[9] *Camera Jabber. 2021. How to use the Samsung S10 Pro camera mode. [online] Available at: <https://camerajabber.com/how-to-use-the-samsung-s10-pro-mode/ > [Accessed 11 October 2021]*