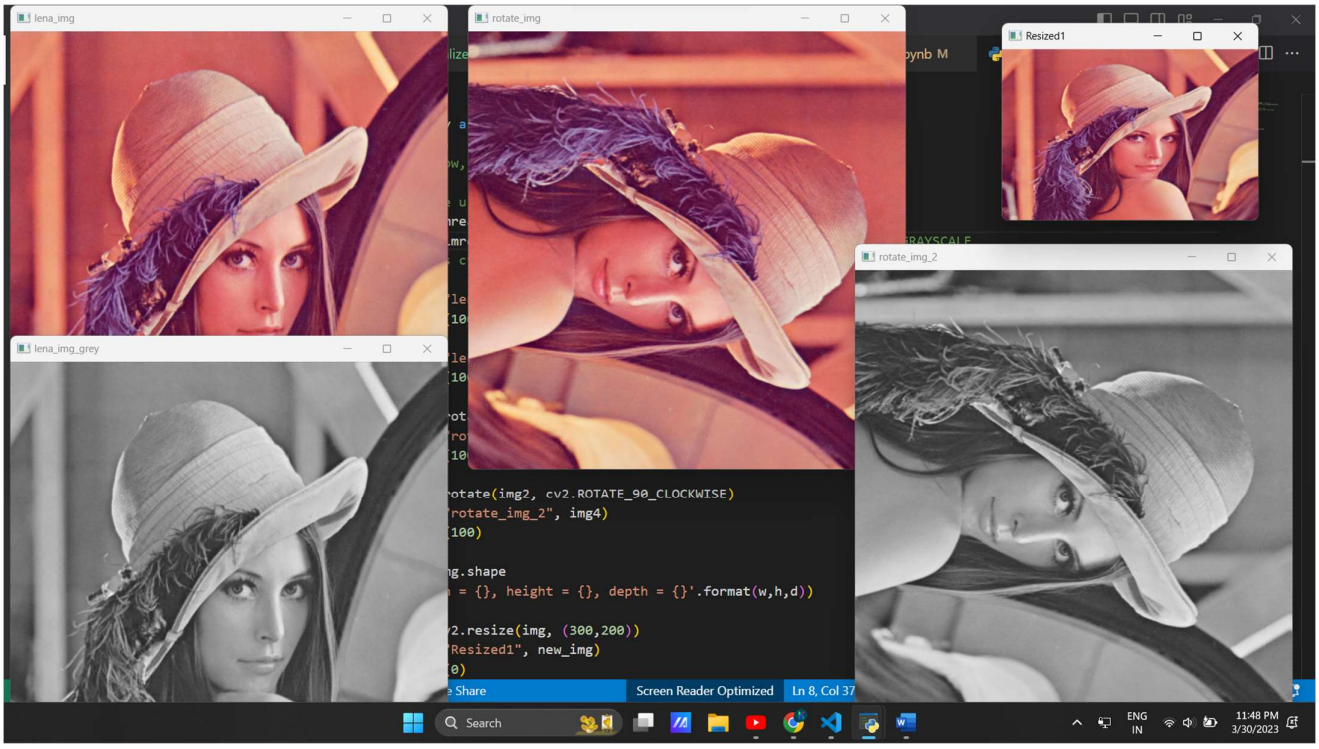
Practical 1:-

```
import cv2
import numpy as np
## Read, Show, Greyscale, Rotate, Shape, Resize
# load image using imread
img = cv2.imread(r"assets\Lenna.png", cv2.IMREAD_COLOR)  # Flag = 1
img2 = cv2.imread(r"assets\Lenna.png", 0)   # Flag = 0 is cv2.IMREAD_GRAYSCALE
# Flag -1 is cv2.IMREAD_UNCHANGED
cv2.imshow("lena_img", img)   # lena_img is the window name
cv2.waitKey(100)     # Waits for a key to be pressed to close the window
cv2.imshow("lena_img_grey", img2)
cv2.waitKey(100)
img3 = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
cv2.imshow("rotate_img", img3)
cv2.waitKey(100)
img4 = cv2.rotate(img2, cv2.ROTATE_90_CLOCKWISE)
cv2.imshow("rotate_img_2", img4)
cv2.waitKey(100)
(h,w,d) = img.shape
print('width = {}, height = {}, depth = {}'.format(w,h,d))
new_img = cv2.resize(img, (300,200))
cv2.imshow("Resized1", new_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

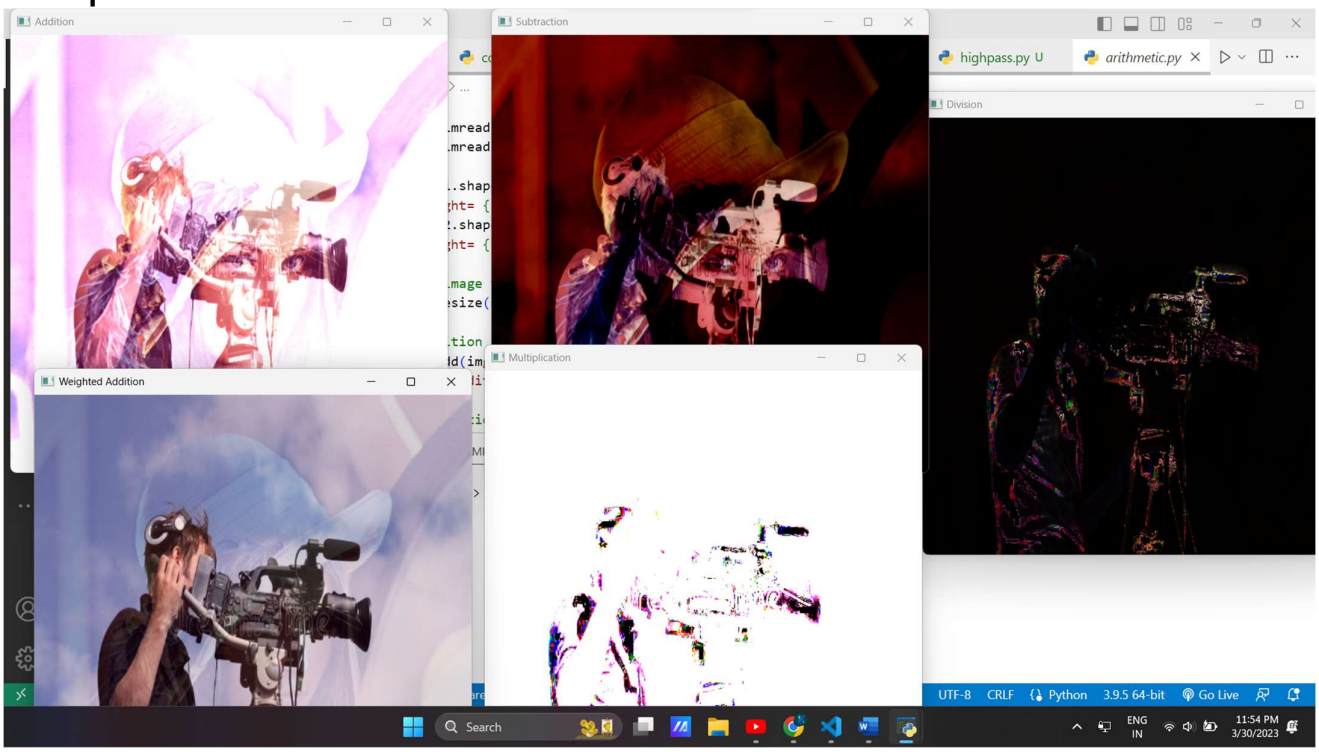
OutPut:-



Practical 2:-

```
import cv2
img1 = cv2.imread('./assets/Lenna.png', 1)
img2 = cv2.imread('./assets/camera_man.jpg', 1)
(h,w,d)=img1.shape
print(f"Height= {h}\nWidth= {w}\nDepth= {d}")
(h,w,d)=img2.shape
print(f"Height= {h}\nWidth= {w}\nDepth= {d}")
#resize of image to add them
new = cv2.resize(img2, (512, 512))
#normal addition
add = cv2.add(img1,new)
cv2.imshow('Addition', add)
#Weighted addition
addw = cv2.addWeighted(img1, 0.2, new, 0.8, 0)
cv2.imshow('Weighted Addition', addw)
#normal subtraction
sub = cv2.subtract(img1,new)
cv2.imshow('Subtraction',sub)
#multiplication
mul = cv2.multiply(img1, new)
cv2.imshow('Multiplication',mul)
#division
div = cv2.divide(img1, new)
cv2.imshow('Division',div)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:-

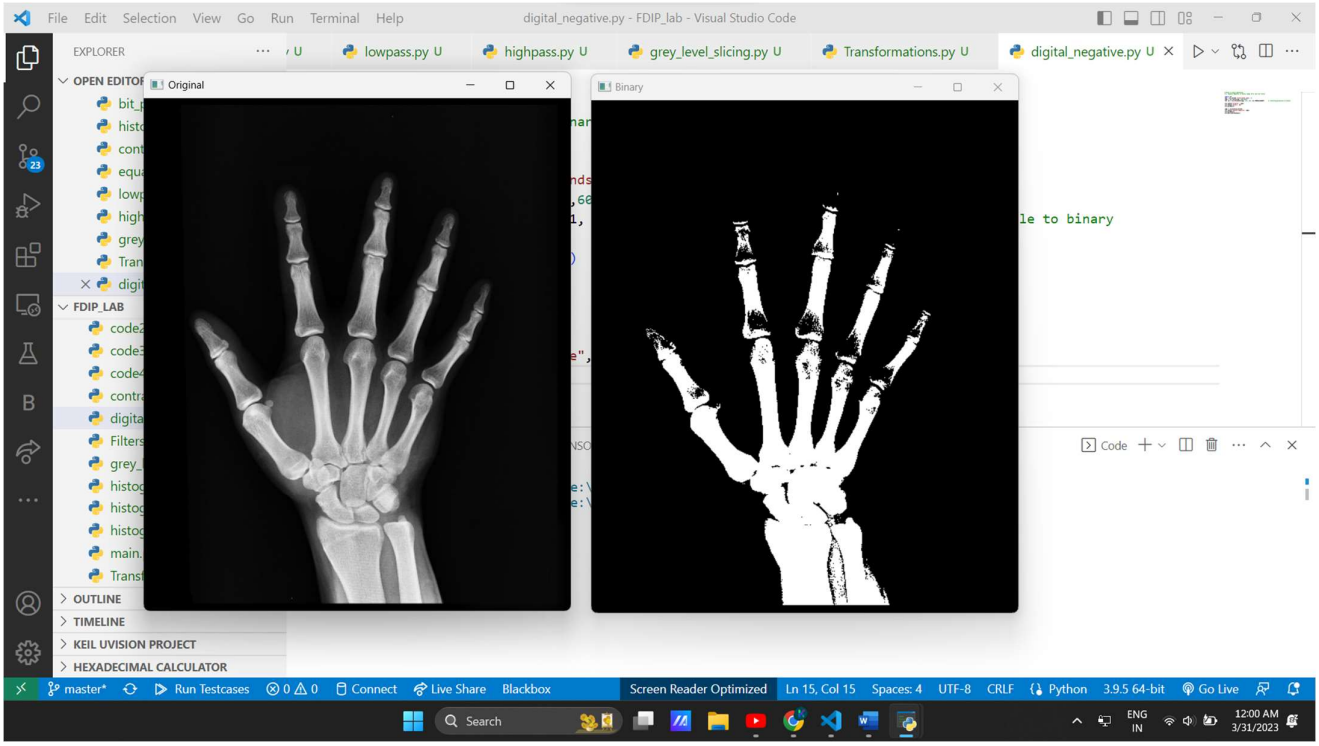


Practical 3:-

Digital Negative:-

```
# Point to point operations
# 1. Digital Negative of binary image (0->1 and vice versa)

import cv2
img = cv2.imread('assets\hands.jpeg', 1)
img1 = cv2.resize(img, (500,600))
ret, b1 = cv2.threshold(img1, 127, 255, cv2.THRESH_BINARY)      # converting greyscale to
binary

cv2.imshow('Original', img1)
cv2.imshow('Binary', b1)
cv2.waitKey(0)

img2 = cv2.bitwise_not(b1)
cv2.imshow("Digital Negative", img2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

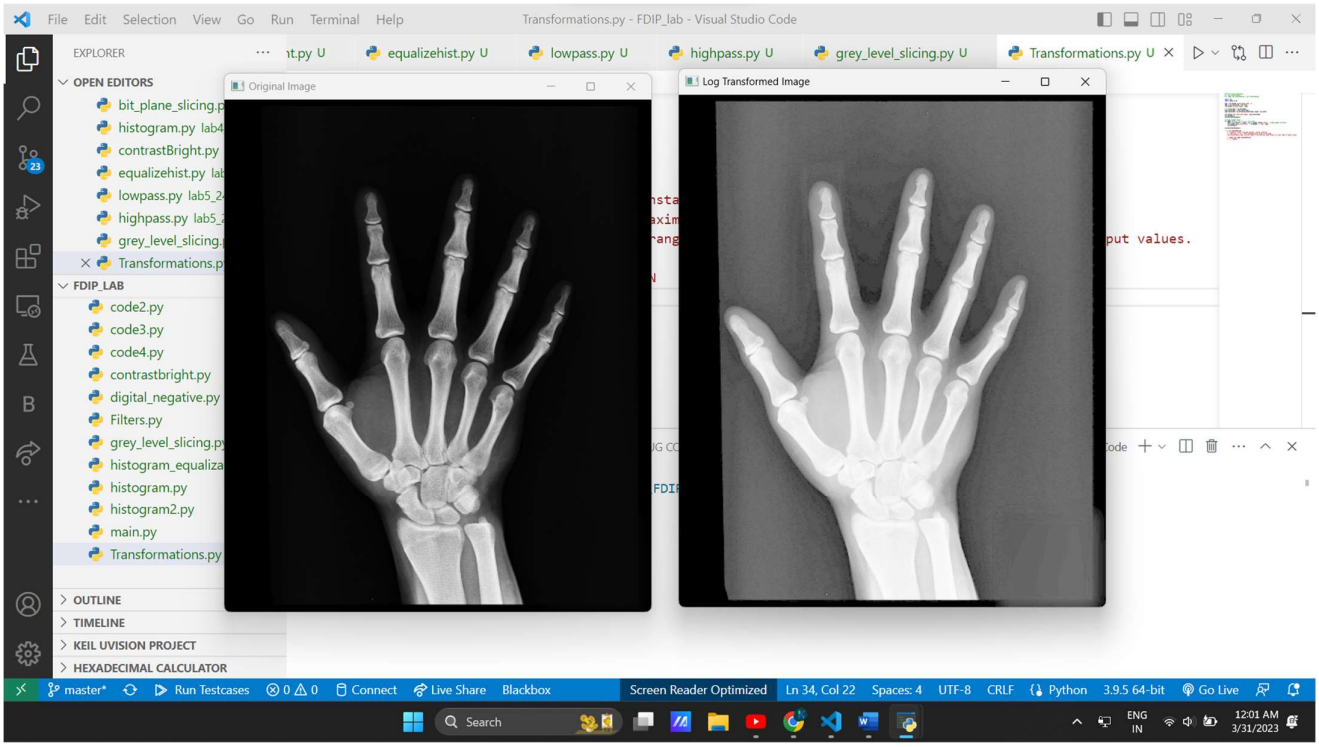
Output:-



Transformation:-

```
# Point to point operations
# 2. Log Transformation
# 3. Power Law Transformation / Law Transformation

import cv2
import numpy as np

img = cv2.imread('assets\hands.jpeg', 1)
img1 = cv2.resize(img, (500,600))
cv2.imshow("Original Image", img1)

c = 255/np.log(1 + np.max(img1))
log_transformed = c*(np.log(1+img1))
log_transformed = np.array(log_transformed, dtype = np.uint8)

cv2.imshow("Log Transformed Image", log_transformed)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Trying 4 gamma values.
for gamma in [0.1, 0.5, 1.5, 2.5, 3.5, 5.5]:
    img2 = np.array(255 * (img1 / 255) ** gamma, dtype='uint8')   # Apply gamma
correction.
    cv2.imshow('gamma_transformed ' + str(gamma) + '.jpg', img2)
    cv2.waitKey(0)

cv2.destroyAllWindows()
```

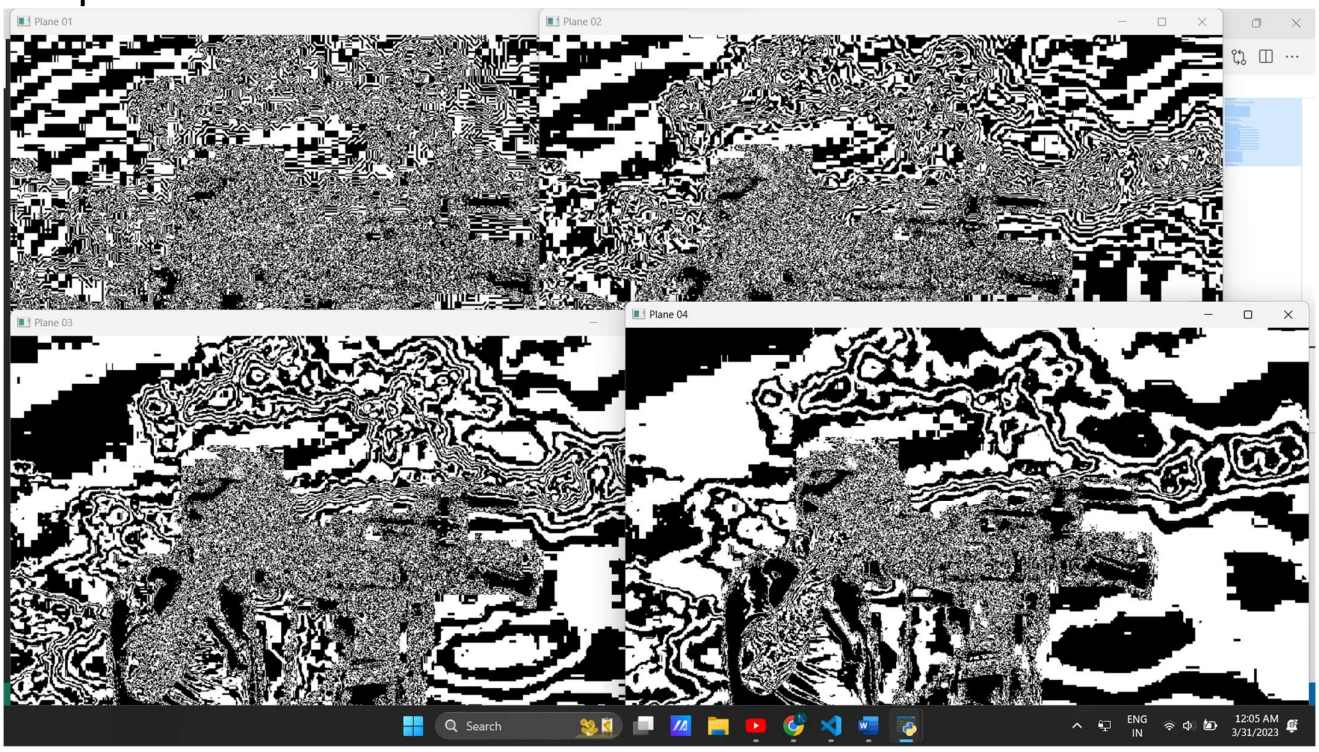
Output:-

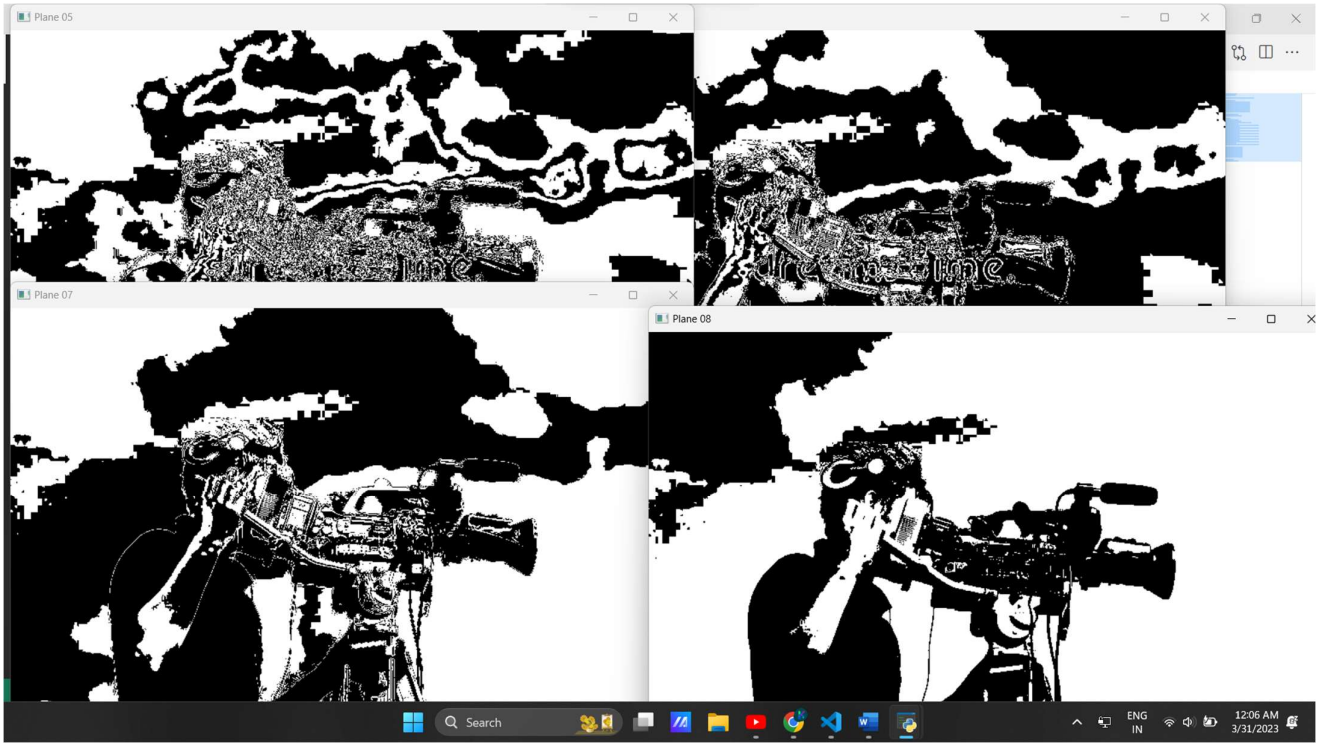


Bit Plane Slicing:-

```
import numpy as np
import cv2 as cv

img=cv.imread("./assets/camera_man1.png",0)

[w,h]=img.shape
img1=np.zeros([w,h,3],dtype=np.uint8)
img2=np.zeros([w,h,3],dtype=np.uint8)
img3=np.zeros([w,h,3],dtype=np.uint8)
img4=np.zeros([w,h,3],dtype=np.uint8)
img5=np.zeros([w,h,3],dtype=np.uint8)
img6=np.zeros([w,h,3],dtype=np.uint8)
img7=np.zeros([w,h,3],dtype=np.uint8)
img8=np.zeros([w,h,3],dtype=np.uint8)

def bitget(nbr,pos):
    return (nbr>>pos) & 1

planes=8
for i in range(0,w):
    for j in range(0,h):
        for p in range(planes-1,-1,-1):
            if p==7:
                img1[i][j]=255*bitget(img[i][j],p)
            elif p==6:
                img2[i][j]=255*bitget(img[i][j],p)
            elif p==5:
                img3[i][j]=255*bitget(img[i][j],p)
            elif p==4:
                img4[i][j]=255*bitget(img[i][j],p)
            elif p==3:
                img5[i][j]=255*bitget(img[i][j],p)
            elif p==2:
                img6[i][j]=255*bitget(img[i][j],p)
            elif p==1:
                img7[i][j]=255*bitget(img[i][j],p)
            else:
                img8[i][j]=255*bitget(img[i][j],p)

cv.imshow('Plane 08',img1)
cv.imshow('Plane 07',img2)
cv.imshow('Plane 06',img3)
cv.imshow('Plane 05',img4)
cv.imshow('Plane 04',img5)
cv.imshow('Plane 03',img6)
cv.imshow('Plane 02',img7)
cv.imshow('Plane 01',img8)

cv.waitKey(0)
cv.destroyAllWindows
```

Output:-

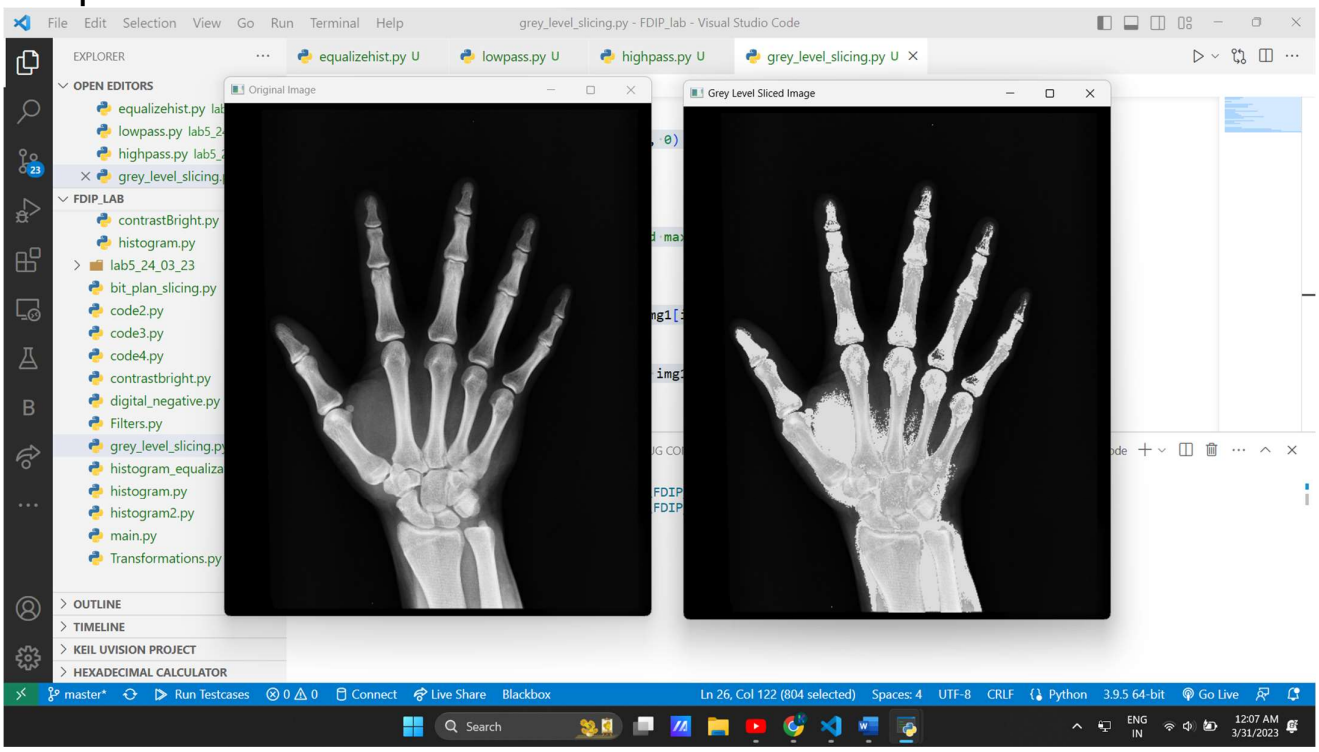

Plane 01
Plane 02
Plane 03
Plane 04
Search
ENG
IN
12:05 AM
3/31/2023


Plane 05
Plane 07
Plane 08
Search
ENG
IN
12:06 AM
3/31/2023

Grey Level Slicing:-

```
# Point to point operations
# 4. Grey Level Slicing

import cv2

img = cv2.imread(r"assets\hands.jpeg", 0)
img1 = cv2.resize(img, (500, 600))
cv2.imshow('Original Image', img1)

w,h = img1.shape
min_range = 80   # Specify the min and max range
max_range = 160
for i in range(0,w):
    for j in range(0,h):
        if img1[i][j]>min_range and img1[i][j]<max_range:
            img1[i][j] = 220

cv2.imshow('Grey Level Sliced Image', img1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:-

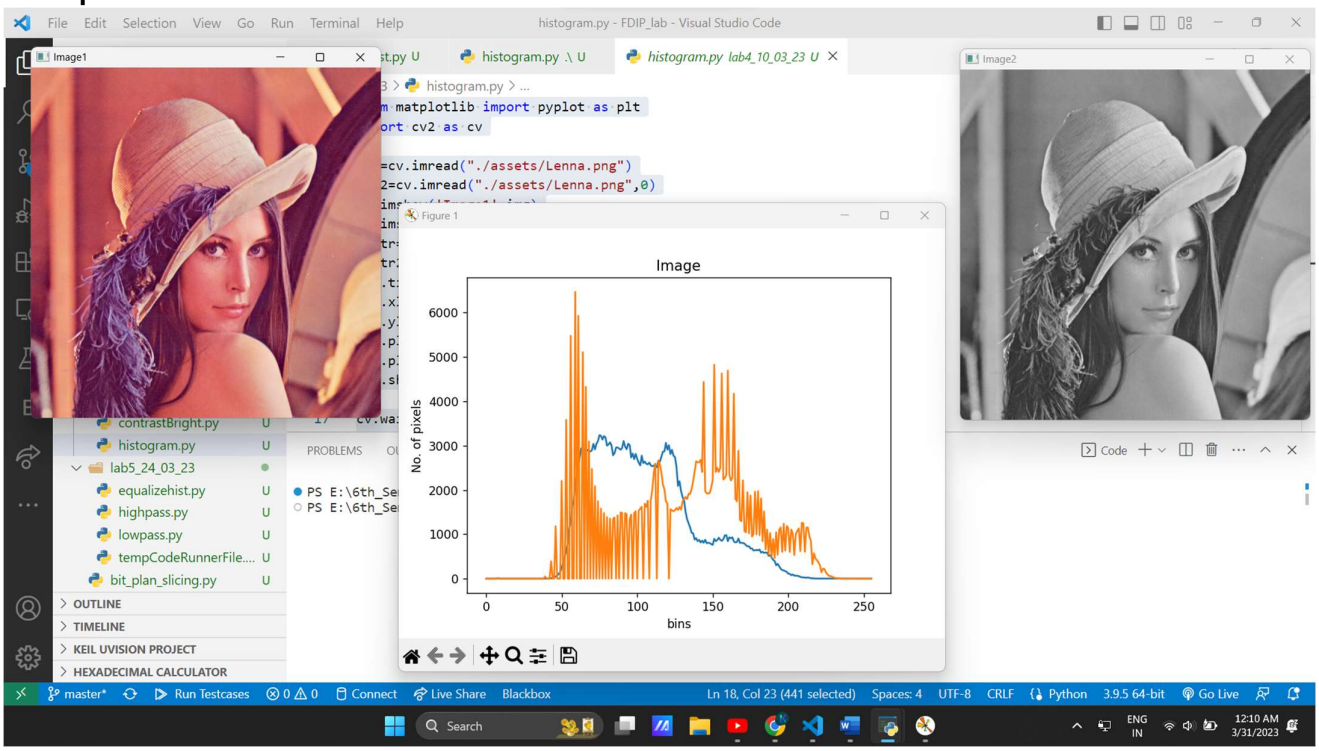


Practical 4:-

Histogram:-

```
from matplotlib import pyplot as plt
import cv2 as cv

img=cv.imread("./assets/Lenna.png")
img2=cv.imread("./assets/Lenna.png",0)
cv.imshow('Image1',img)
cv.imshow('Image2',img2)
histr=cv.calcHist([img],[0],None,[256],[0,256])
histr2=cv.calcHist([img2],[0],None,[256],[0,256])
plt.title("Image")
plt.xlabel('bins')
plt.ylabel('No. of pixels')
plt.plot(histr)
plt.plot(histr2)
plt.show()

cv.waitKey(0)
cv.destroyAllWindows()
```

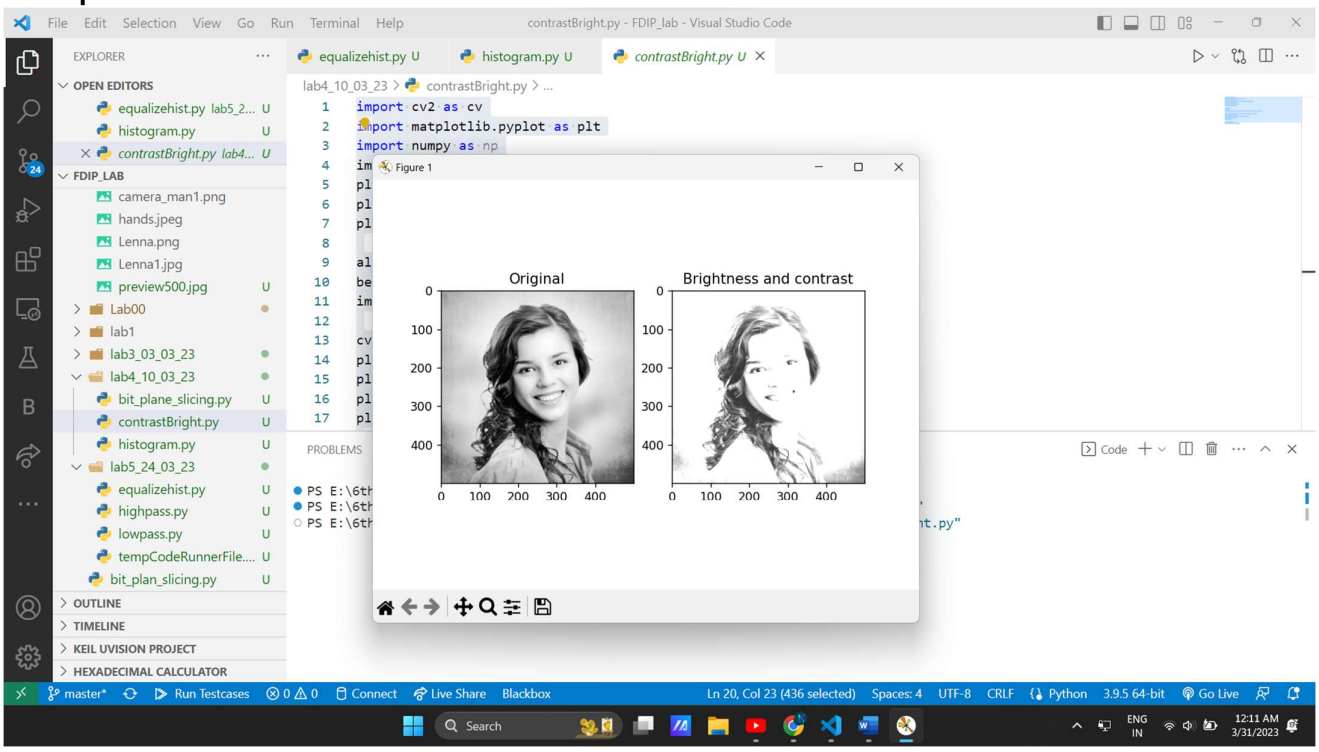
Output:-



Contrast Bright:-

```
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
image=cv.imread("./assets/preview500.jpg")
plt.subplot(1,2,1)
plt.title("Original")
plt.imshow(image)

alpha=1.5
beta=50
image2=cv.convertScaleAbs(image,alpha=alpha,beta=beta)

cv.imwrite("Brightness and contrast.jpg",image2)
plt.subplot(1,2,2)
plt.title("Brightness and contrast")
plt.imshow(image2)
plt.show()

cv.waitKey(0)
cv.destroyAllWindows()
```

Output:-



Histogram Eqalization:-

```
# import Opencv
import cv2

# import Numpy
import numpy as np

# read a image using imread
# read image
path = "assets\camera_man.jpg"
flag = 0
img = cv2.imread(path, flag)

# creating a Histograms Equalization
# of a image using cv2.equalizeHist()
equ = cv2.equalizeHist(img)

# stacking images side-by-side
res = np.hstack((img, equ))

# show image input vs output
cv2.imshow('cameraman', res)

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

Output:-