Mansoura University
Faculty of Computers and Information
Department of Information Technology
First Semester- 2023-2024

- Image Processing.
  - ➢ color Models.
  - ➢ Resizing for images.

1. Grayscale :

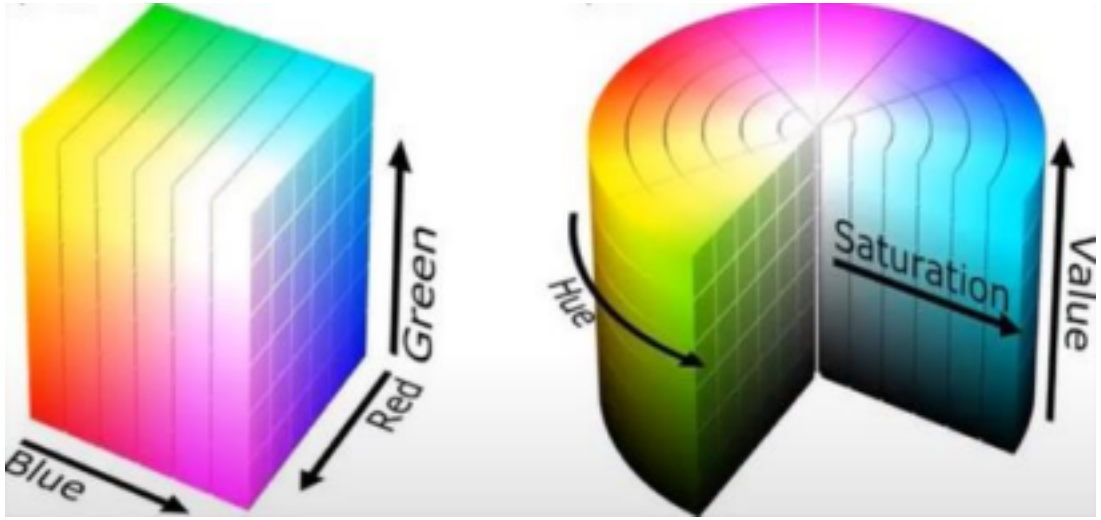
- which pixel in a grayscale image is represented by a single 8-bit value, ranging from 0 for black to 255 for white.

2. BGR :

- which each pixel has a triplet of values representing the blue, green, and red components.
- each pixel is represented by a triplet of 8-bit values, such as [0, 0, 0] for black, [255, 0, 0] for blue, [0, 255, 0] for green, [0, 0, 255] for red, and [255, 255, 255] for white.

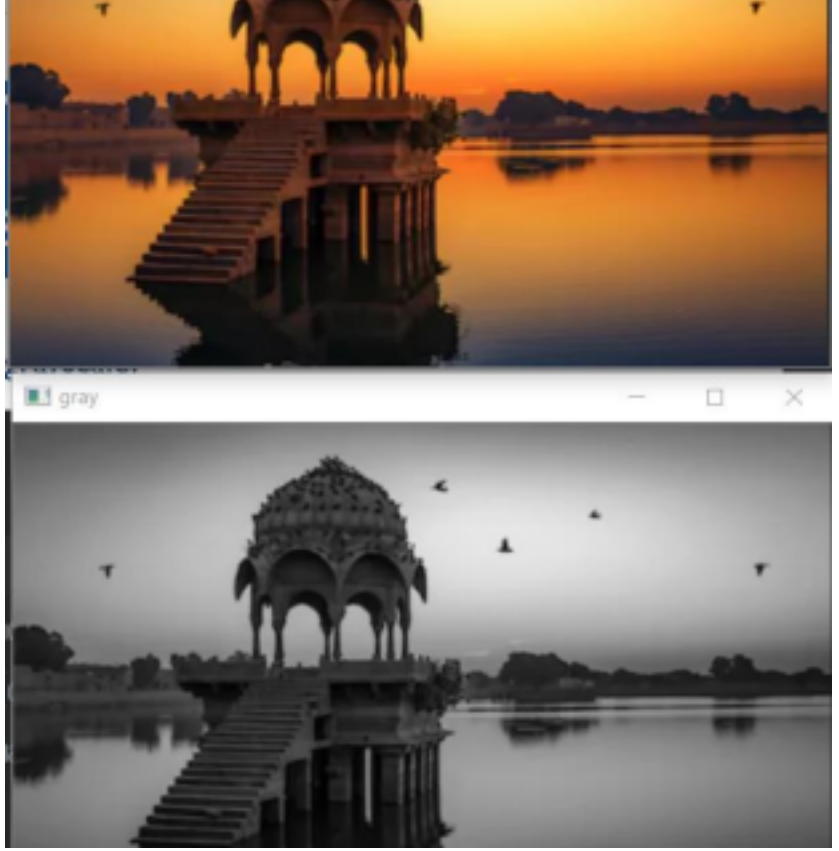
3. HSV :

- uses a different triplet of channels.
- Hue is the color's tone.
- saturation is its intensity.
- value represents its brightness.



```
import cv2

originalImage = cv2.imread( 'C:\\Users\\user\\Desktop\\img.png' )
grayImage = cv2.cvtColor(originalImage, cv2.COLOR_BGR2GRAY)
cv2.imshow("original", originalImage)
cv2.imshow("gray", grayImage)
cv2.waitKey()
cv2.destroyAllWindows()
```



```
import cv2

originalImage = cv2.imread( 'C:\\Users\\user\\Desktop\\img.png' )
#gray
grayImage = cv2.cvtColor(originalImage, cv2.COLOR_BGR2GRAY)
#binary
(thresh, blackAndWhiteImage) = cv2.threshold(grayImage, 127, 255,
cv2.THRESH_BINARY)

cv2.imshow("gray", grayImage)
cv2.imshow("binary", blackAndWhiteImage)
cv2.waitKey()
cv2.destroyAllWindows()
```

```
import cv2

originalImage = cv2.imread( 'C:\\Users\\user\\Desktop\\img.png' )
RGBImage = cv2.cvtColor(originalImage, cv2.COLOR_BGR2RGB)

cv2.imshow("original", originalImage)
cv2.imshow("rgb", RGBImage)
cv2.waitKey()
cv2.destroyAllWindows()
```

```
import cv2
```

```
originalImage = cv2.imread(
'C:\\Users\\user\\Desktop\\img.png' )
B, G, R = cv2.split(originalImage)

cv2.imshow("original", originalImage)
cv2.imshow("blue", B)
cv2.imshow("Green", G)
cv2.imshow("red", R)

m=cv2.merge((B, G, R))
cv2.imshow("merged", m)

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





```
import cv2

originalImage = cv2.imread( 'C:\\Users\\user\\Desktop\\img.png' )
hsvImage=cv2.cvtColor(originalImage, cv2.COLOR_BGR2HSV)
```

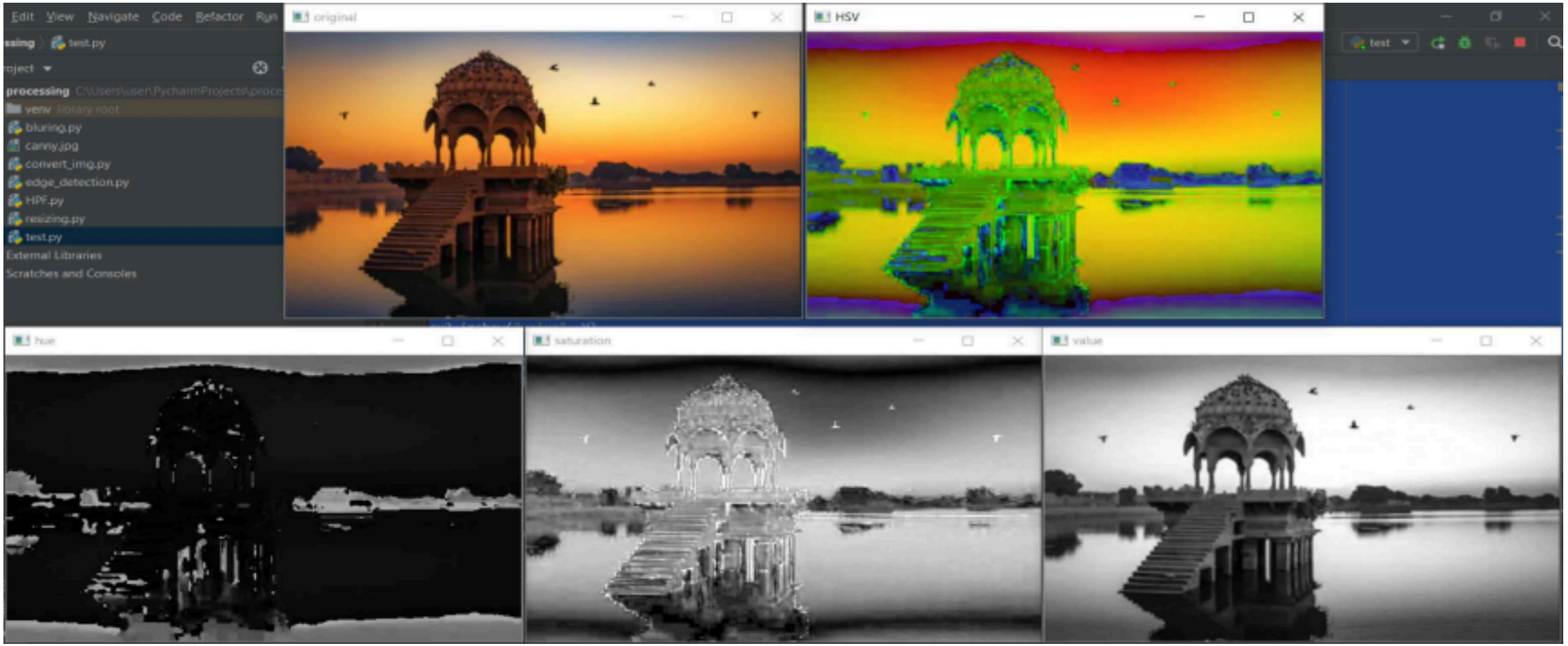
```
cv2.imshow("original", originalImage)
cv2.imshow("HSV", hsvImage)

H=hsvImage[:,:,0]
S=hsvImage[:,:,1]
V=hsvImage[:,:,2]

cv2.imshow("hue", H)
cv2.imshow("saturation", S)
cv2.imshow("value", V)

cv2.waitKey()

cv2.destroyAllWindows()
```


Edit View Navigate Code Refactor Run
test.py
processing C:\Users\user\PycharmProjects\proce
venv library root
bluring.py
canny.jpg
convert_img.py
edge_detection.py
HPF.py
resizing.py
test.py
External Libraries
Scratches and Consoles
original
HSV
test
hue
saturation
value

```
import cv2

image = cv2.imread( 'C:\\Users\\user\\Desktop\\img.png' )

half = cv2.resize(image, (0, 0), fx = 0.5, fy = 0.5)
bigger = cv2.resize(image, (1050, 1610))
stretch_near = cv2.resize(image, (780, 540),
interpolation = cv2.INTER_NEAREST)

cv2.imshow("image", image)
cv2.imshow("half", half)
cv2.imshow("big", bigger)
cv2.imshow("stretch", stretch_near)

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

Thank you