Name : Aum M. Dhabalia **Date :** 21/07/2024

Roll No. : 21BEC027

Experiment…1 - Perform basic Image Handling and processing operations on the image.

Objective : The objective of this lab is to introduce the student to OpenCV, especially for image processing.

Import necessary libraries…

```
'''
Created on 21 July 2024 Mon 2:30:33 pm...
'''
import numpy as np
import math
import cv2
import matplotlib.pyplot as plt
```

Task 1. Reading, displaying, and writing an image using OpenCV.

```
#Task 1...
print("Read an Image")
I = cv2.imread(r"D:\Nirma Files\Ghost.jpg");
cv2.imshow('Original',I)
cv2.waitKey(0)
```

Output

Image in OpenCV

Task 2. Convert the image to another format using OpenCV.

```
#Task 2...
print("Image format conversion")
Irgb = cv2.cvtColor(I,cv2.COLOR_BGR2RGB)
plt.imshow(I)
plt.title("Original Image")
print(I.shape)
#hold the window
plt.waitforbuttonpress()
plt.close('all')
```

Output

```
Image format conversion
(1686, 800, 3)
```

Task 3. Perform image resizing using OpenCV.

```
print("Image Resizing")
resize = cv2.resize(I,(0,0),fx=0.6,fy=0.6)
cv2.imshow('Resized Image',resize)
cv2.waitKey(0)
```

Output

Resized Image

Task 4. Convert colored image into grayscale image using OpenCV.

```
print("Conversion from RGB to Grayscale")
Ig = cv2.cvtColor(I,cv2.COLOR_RGB2GRAY)
print(Ig.shape)
cv2.imshow('Grayscale Image',Ig)
cv2.imwrite(‘Grayscale Image.jpg’,Ig)
cv2.waitKey(0)
```

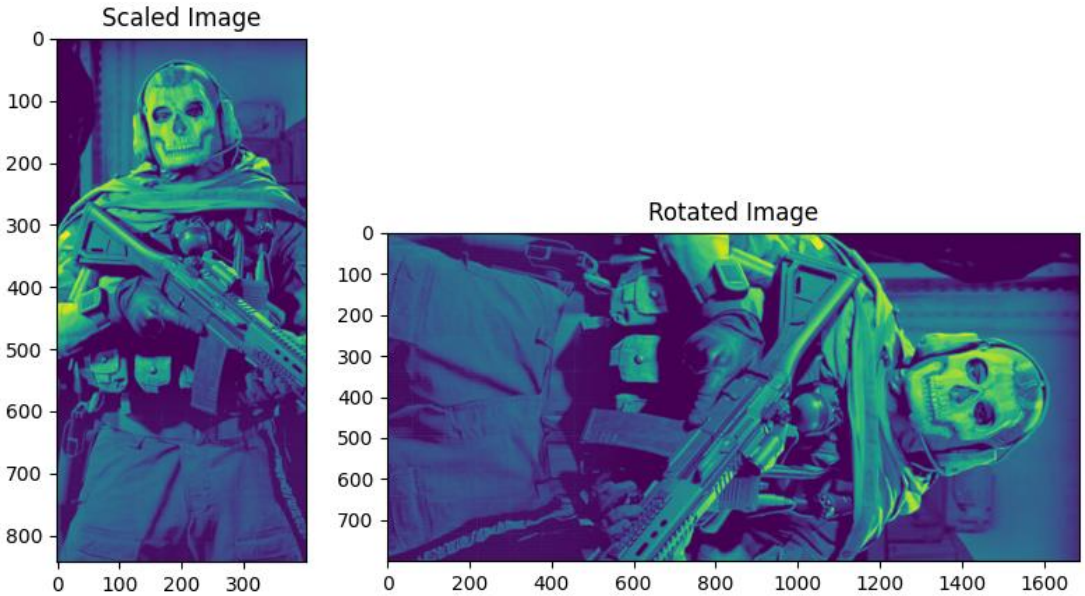
Output

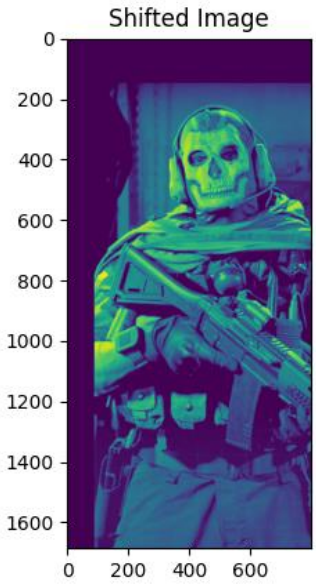
Grayscale Image

Task 5. Scaling, rotation and shifting operation on image using OpenCV.

```
print("Image Scaling\n")
Iscale = cv2.resize(Ig,(0,0),fx = 0.5, fy = 0.5,interpolation = cv2.INTER_AREA)
plt.imshow(Iscale)
plt.title("Scaled Image")
plt.waitforbuttonpress()
print("Image Rotating\n")
Irotate = cv2.rotate(Ig,cv2.ROTATE_90_CLOCKWISE)
plt.imshow(Irotate)
plt.title("Rotated Image")
plt.waitforbuttonpress()
print("Image Shifting")
T = np.float32([[1,0,10],[0,1,50]]);
Ishift = cv2.warpAffine(Ig,T,(Ig.shape[1],Ig.shape[0]));
plt.imshow(Ishift)
plt.title("Shifted Image");
plt.waitforbuttonpress()
```

Output





Task 6. Play video using OpenCV

```
vid = cv2.VideoCapture(r"D:\Nirma Files\Computer Vision\Believer.mkv");
if(vid.isOpened()==False):
    print("Error")
# Read the entire file until it is completed
while(vid.isOpened()):
  # Capture each frame
    ret,frame = vid.read()
    if(ret == True):
      # Display the resulting frame
      cv2.imshow('Believer', frame)
    if(cv2.waitKey(25) & 0xFF == ord('q')):
        break
vid.release()
cv2.destroyAllWindows()
```

Task 7. Extract images from video using OpenCV

```
vid = cv2.VideoCapture(r"D:\Nirma Files\Computer Vision\Believer.mkv");
frameno = 0
for i in range(0,10,1):
   ret,frame = vid.read()
```

```
  if ret:
      name = str(frameno) + '.jpg'
      print ('new frame captured...' + name)
      cv2.imwrite(name,frame)
      frameno += 1
  else:
     break
vid.release()
cv2.destroyAllWindows()
```

Output

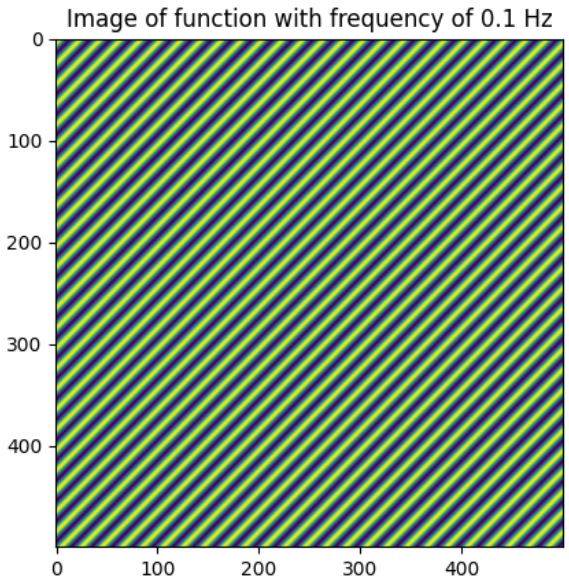
Task 8. Use following formula $f(i,j) = \sin(2\,\pi\,f\,(i+j))$ where i and j are indices of pixel, draw an image with different frequencies.

```
print("Display function in form of image")
x = 500
y = 500
t = 100;
fr = 1/t;
f = np.zeros([x,y],dtype = float);
for i in range(1,x,1):
   for j in range(1,y,1):
      f[i,j] = math.sin(math.pi*fr*(i + j))
for i in range(1,x,1):
   for j in range(1,y,1):
      f[i,j] = (((f[i,j] + 1.0)/2.0)*255).astype(np.uint8)
plt.imshow(f)
```

plt.title('Image of function with frequency of 0.1Hz')

plt.waitforbuttonpress()

Output



From the image, it is observed that the function f(i,j) produced an image of parallel straight lines at the angle of 45 degrees which are actually 2D representation of sinusoidal waveform with frrquency of 0.1Hz.

Conclusion:-

As the experiment performed,

- the concept of image as a 2D matrix was familiarized as well as reading and displaying of an image using python script. The concept of relation between image and video was familiarized in form of frames per second.
- basics concepts of image processing like format conversion, resizing, scaling, rotation, etc. were performed using OpenCV library as well as using matplot library.
- Using OpenCV, a video was read and displayed as well as frames from the video was extracted.
- image was generated from a mathematical function of rows and columns.

Various libraries and functions were used like matplotlib, OpenCV, cv2.imshow(), math library, VideoCapture(), vid.read(), cv2.rotate, cv2.warpAffine(), cv2.resize, etc.