**1.Develop a program to display grayscale image using read and write operation.**

**------------------------------------------------------------------------------------**

**import cv2**

**import numpy as np**

**image = cv2.imread('wolf1.jpg')**

**image = cv2.resize(image, (0, 0), None, .25, .25)**

**gray = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)**

**gray\_3\_channel = cv2.cvtColor(grey, cv2.COLOR\_GRAY2BGR)**

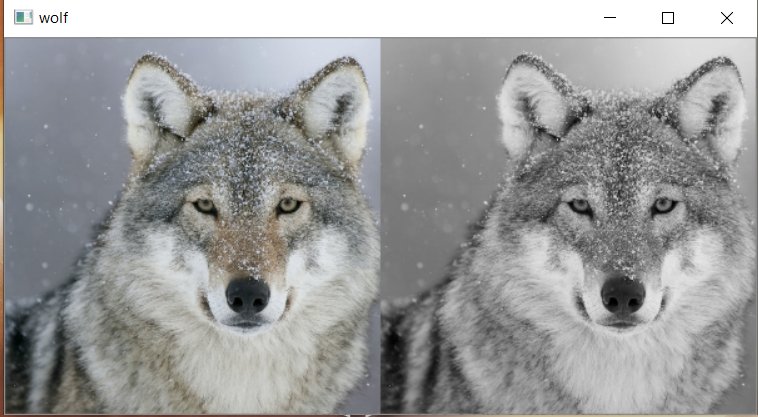
**numpy\_horizontal = np.hstack((image, gray\_3\_channel))**

**numpy\_horizontal\_concat = np.concatenate((image, gray\_3\_channel), axis=1)**

**cv2.imshow('wolf', numpy\_horizontal\_concat)**

**cv2.waitKey()**

**OUTPUT**



**2.Develop a program to perform linear transformation on image.**

**------------------------------------------------------------------------------------------------**

**import cv2**

**import numpy as np**

**FILE\_NAME = 'wolf1.jpg'**

**try:**

**img = cv2.imread(FILE\_NAME)**

**(height, width) = img.shape[:2]**

**res = cv2.resize(img, (int(width / 2), int(height / 2)), interpolation = cv2.INTER\_CUBIC)**

**cv2.imwrite('result.jpg', res)**

**cv2.imshow('image',img)**

**cv2.imshow('result',res)**

**cv2.waitKey(0)**

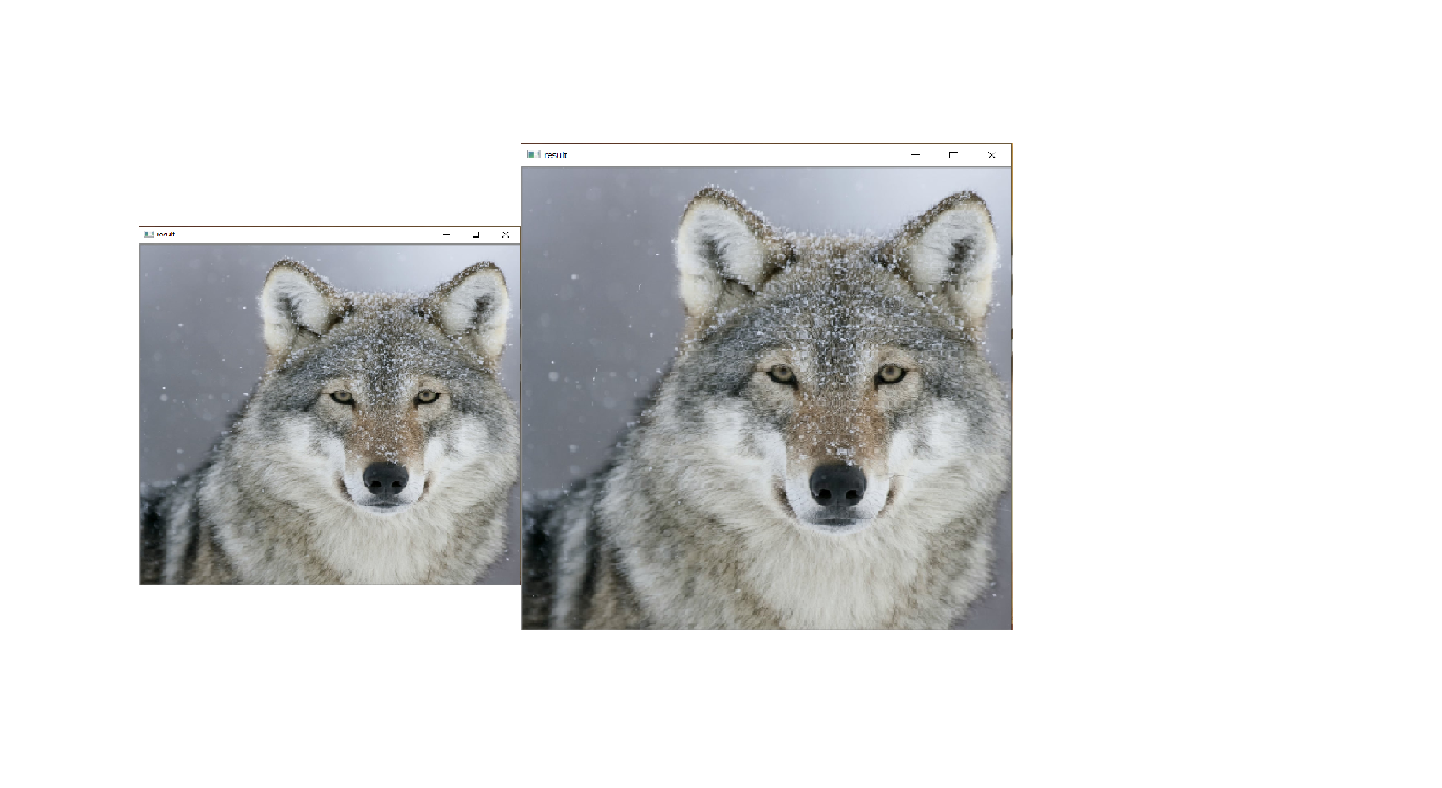
**except IOError:**

**print ('Error while reading files !!!')**

**cv2.waitKey(0)**

**cv2.destroyAllWindows(0)**

**Output:**



**import cv2**

**import numpy as np**

**FILE\_NAME = 'wolf1.jpg'**

**img = cv2.imread(FILE\_NAME)**

**(rows, cols) = img.shape[:2]**

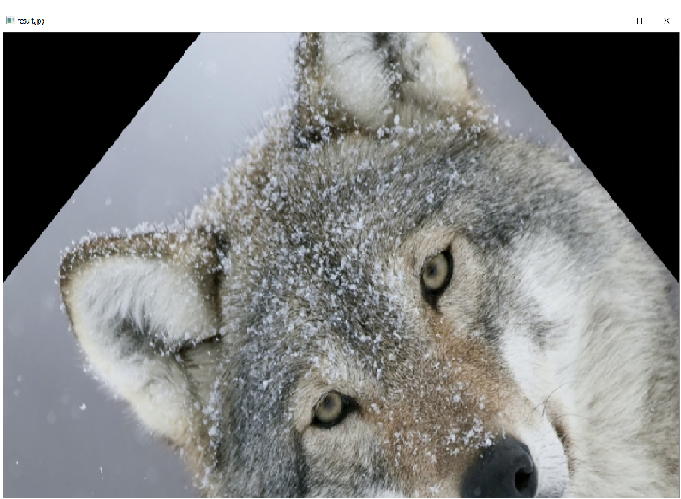
**M = cv2.getRotationMatrix2D((cols / 2, rows / 2), 45, 1)**

**res = cv2.warpAffine(img, M, (cols, rows))**

**cv2.imshow('result.jpg', res)**

**cv2.waitKey(0)**

**Output:**



**3.Develop a program to find sum and mean of a set of images.**

**Create n number of images and read the directory and perform operation.**

**------------------------------------------------------------------------------------------------**

**import cv2**

**import os**

**path = "E:\IP\Resize"**

**imgs=[]**

**dirs=os.listdir(path)**

**for file in dirs:**

**fpat=path+"\\"+file**

**imgs.append(cv2.imread(fpat))**

**i=0**

**for im in imgs:**

**cv2.imshow(dirs[i],imgs[i])**

**i=i+1**

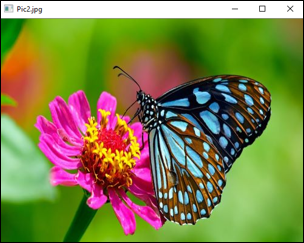
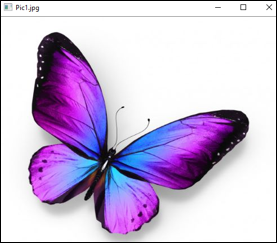
**print(i)**

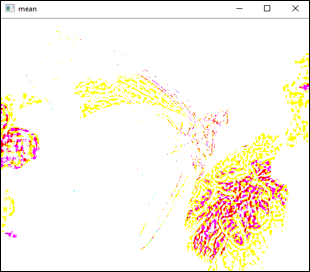
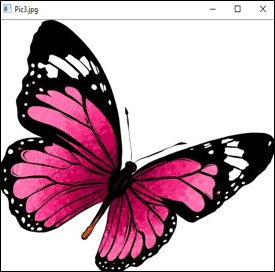
**cv2.imshow('sum',len(im))**

**cv2.imshow('mean',len(im)/im)**

**cv2.waitKey(0)**

**Output:**





**4.Write a program to convert color image into gray scale and binary image.**

**------------------------------------------------------------------------------------------------**

**import cv2**

**img = cv2.imread("wolf2.jpg")**

**gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)**

**cv2.imshow("Binary Image",gray)**

**cv2.waitKey(0)**

**cv2.destroyAllWindows()**

**ret, bw\_img = cv2.threshold(img,127,255,cv2.THRESH\_BINARY)**

**cv2.imshow("Binary Image",bw\_img)**

**cv2.waitKey(0)**

**cv2.destroyAllWindows()**

**Output:**



**5.Write a program to convert color image into different color space.**

**------------------------------------------------------------------------------------------------**

**import cv2**

**img = cv2.imread("dog.jpg")**

**gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)**

**hsv=cv2.cvtColor(img,cv2.COLOR\_BGR2HSV)**

**lab=cv2.cvtColor(img,cv2.COLOR\_BGR2LAB)**

**hls=cv2.cvtColor(img,cv2.COLOR\_BGR2HLS)**

**cv2.imshow("GRAY image",gray)**

**cv2.waitKey(0)**

**cv2.imshow("HSV image",hsv)**

**cv2.waitKey(0)**

**cv2.imshow("LAB image",lab)**

**cv2.waitKey(0)**

**cv2.imshow("HLS image",hls)**

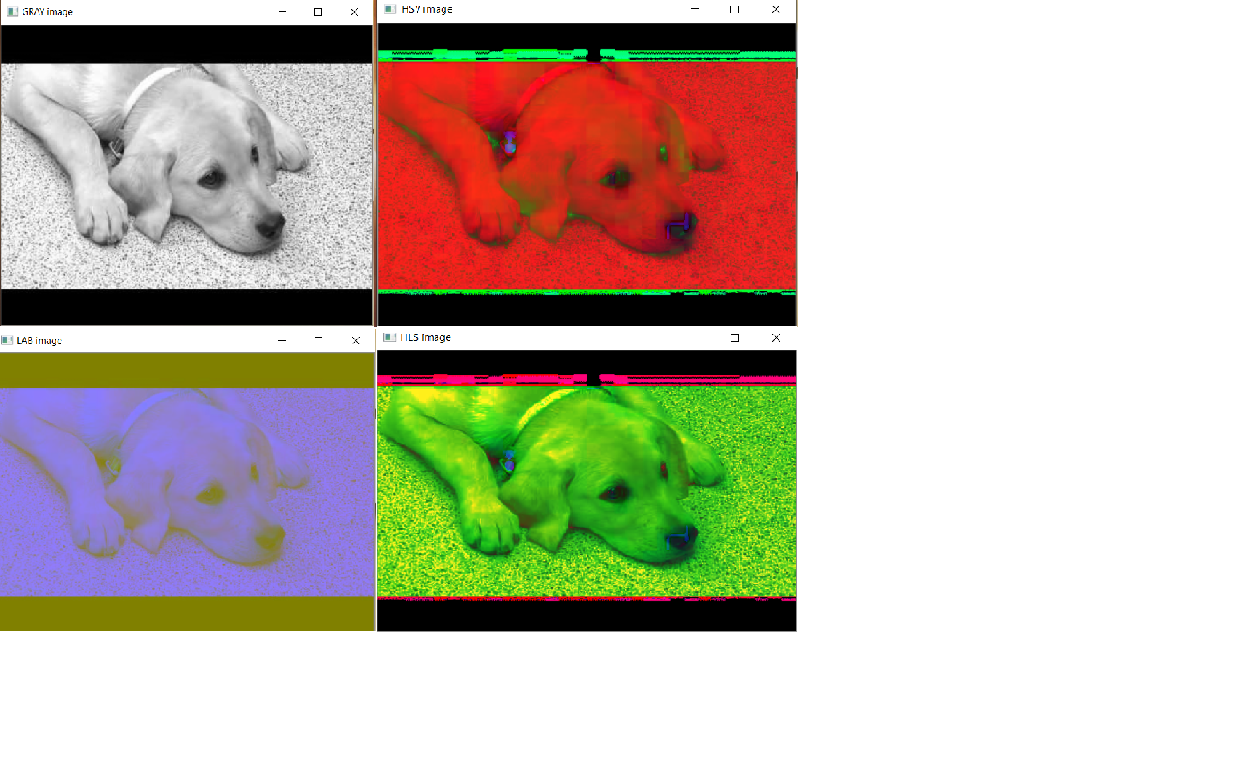
**cv2.waitKey(0)**

**cv2.destroyAllWindows()**

**ret, bw\_img = cv2.threshold(img,127,255,cv2.THRESH\_BINARY)**

**cv2.destroyAllWindows()**

**Output:**



**6.Develop a program to create an image from 2D array.**

**import numpy as np**

**from PIL import Image**

**import cv2 as c**

**array = np.zeros([100, 200, 3], dtype=np.uint8)**

**array[:,:100] = [150, 128, 0] #Orange left side**

**array[:,100:] = [0, 0, 255] #Blue right side**

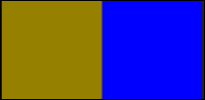
**img = Image.fromarray(array)**

**img.save('Panda.jpg')**

**img.show()**

**c.waitKey(0)**

**Output:**



**7.Find the sum of all neighborhood values of the matrix.**

**import numpy as np**

**M = [[1, 2, 3],**

**[4, 5, 6],**

**[7, 8, 9]]**

**M = np.asarray(M)**

**N = np.zeros(M.shape)**

**def sumNeighbors(M,x,y):**

**l = []**

**for i in range(max(0,x-1),x+2):**

**for j in range(max(0,y-1),y+2):**

**try:**

**t = M[i][j]**

**l.append(t)**

**except IndexError: # if entry doesn&#39;t exist**

**pass**

**return sum(l)-M[x][y] # exclude the entry itself**

**for i in range(M.shape[0]):**

**for j in range(M.shape[1]):**

**N[i][j] = sumNeighbors(M, i, j)**

**print ("Original matrix:\n", M)**

**print ("Summed neighbors matrix:\n", N)**

**Output:**

**Original matrix:**

**[[1 2 3]**

**[4 5 6]**

**[7 8 9]]**

**Summed neighbors matrix:**

**[[11. 19. 13.]**

**[23. 40. 27.]**

**[17. 31. 19.]]**