**Experiment – 2**

**Basic Concepts in OpenCV Image and Video Handling**

**Aim:**

1. To code the following tasks in OpenCV
   1. To read and display an image by using OpenCV
   2. To resize an image using OpenCV
2. To capture video from Camera, play a video from a file and save a video file.

**Software/ Package Used:**

1. Pycharm IDE
2. Libraries used:
3. NumPy
4. opencv-python
5. matplotlib
6. scipy

**Programs:**

1. **Read and Display image using OpenCV:**
   1. **Read and display**

# import all the libraries

import cv2 as cv  
import sys

#load an image using imread function   
img=cv.imread(r" C:/Users/21r228\Downloads\download.jpg ")  
print(img)

#image not found   
if img is None:  
 sys.exit("Could not read the image")

#display the image   
cv.imshow("Final",img)  
k=cv.waitKey(0)  
l=cv.destroyAllWindows()

**Input : Color image**



**Output : Color image**



**Image matrix in form of pixels**

[[[255 255 255]

[255 255 255]

[255 255 255]

* 1. **Colour to Grayscale image**

#import all libraries  
import cv2 as cv  
import sys

#load an image  
img=cv.imread(r" C:/Users/21r228\Downloads\download.jpg ",0) # 0 indicates grayscale   
print(img)  
if img is None:  
 sys.exit("Could not read the image") #image location not found

#display the image in grayscale  
cv.imshow("Final",img)  
k=cv.waitKey(0)  
l=cv.destroyAllWindows()

**Input: Color image**



**Output: Grayscale image**



**Image matrix**

[[ 0 0 0 ... 252 254 254]

[ 0 0 0 ... 252 254 254]

[ 0 0 0 ... 252 254 254] ...

[ 49 45 42 ... 246 247 250]

[ 61 56 52 ... 246 247 250]

[ 76 70 65 ... 246 247 250]]

1. **Write an image**

import cv2 as cv  
import sys  
# Load an image  
image =cv.imread(r" C:/Users/21r228\Downloads\download.jpg ",0) # image path  
# Check if the image was successfully loaded  
if image is not None:  
 # Display the image in a window  
 cv.imshow("Dark", image) # name of the output window  
else:  
 print("Image not found") # image not found in target location  
k = cv.waitKey(0)  
if k == ord('s'):  
 cv.imwrite(r" C:/Users/21r228\Downloads\download.jpg ", image)  
 cv.destroyAllWindows() # Close all OpenCV windows

**Input : Colour image**



**Output : Saved image**

1. **Resize an Image Using OpenCV:**

# import all the libraries  
import cv2 as cv  
import sys

# load an image  
img=cv.imread(r" C:/Users/21r228\Downloads\download.jpg ")  
# resize the image using scaling factor  
img2=cv.resize(img,(0,0),fx=0.5,fy=0.5)  
# resize the image by varying the width and height of the image  
img3=cv.resize(img,(300,500))  
if img is None:  
 sys.exit("Could not read the image") # image location not found

# display the output  
cv.imshow("Original",img)  
cv.imshow("Scaling Factor",img2)  
cv.imshow("Bigger",img3)  
k=cv.waitKey(0)

# display the height, width and channels of the images  
dim= img.shape  
print("Dimensions of original image :",dim)  
print("Height : ",img.shape[0])  
print("Width : ",img.shape[1])  
print("Channels : ",img.shape[2],"\n")  
dim1= img2.shape  
print("Dimensions of scaled factor image:",dim1)  
print("Height : ",img2.shape[0])  
print("Width : ",img2.shape[1])  
print("Channels : ",img2.shape[2],"\n")  
dim2= img3.shape  
print("Dimensions of resized image:",dim2)  
print("Height : ",img3.shape[0])  
print("Width : ",img3.shape[1])  
print("Channels : ",img3.shape[2])

**Input:**



**Output:**

1. **Original image**



1. **Scaled image**



1. **Varying the parameters**



**Shape of the images**

Dimensions of original image : (250, 150, 3)

Height : 250

Width : 150

Channels : 3

Dimensions of scaled factor image: (178, 100, 3)

Height : 178

Width : 100

Channels : 3

Dimensions of resized image: (500, 300, 3)

Height : 500

Width : 300

Channels : 3

1. **Video Handling**
2. Read, display, Gray scaling, resizing of the video

#import all libraries

import cv2 as cv  
import sys

# connection between the camera and the system  
cap=cv.VideoCapture(0)  
while cap.isOpened():  
 ret,frame = cap.read()  
 if not ret:  
 print("cannot receive frame(stream end?).Exiting..")  
 break

#display the video  
 cv.imshow("camera",frame)  
 if cv.waitKey(1)==ord("q"):  
 break

#convert colour video to grayscale video  
 gray=cv.cvtColor(frame,cv.COLOR\_BGR2GRAY)  
 cv.imshow("gray",gray)

# resize the video  
 res=cv.resize(frame,(1050,640)) # by varying the parameters  
 res1=cv.resize(frame,None,fx=2,fy=1) # using scaling factor

#display the resized video  
 cv.imshow("bigger",res)  
 cv.imshow("scaled",res1)  
cap.release()  
cv.destroyAllWindows()

**Input :**

From the USB camera

**Output:**



**Output: Grayscale video**



**Output: Resized video by varying the parameters**



**Output: resized video by using scaling factors**



1. **Writing the video**

import cv2

#import the video

video= cv2.VideoCapture(C:/Users/21r228\Downloads/tech [MConverter.eu].mp4')

#read the video

success,image = video.read()

count = 0

success = True

while success:

success,image = video.read()

frame\_width = int(video.get(3))

frame\_height = int(video.get(4))

size = (frame\_width, frame\_height)

result = cv2.VideoWriter('filename.avi',

cv2.VideoWriter\_fourcc(\*'MPEG'),10, size)

cv2.imwrite("frame%d.jpg" % count, image) # save frame as JPEG file

cv2.imshow('Frame',image)

if cv2.waitKey(10) == 27: # exit if Escape is hit

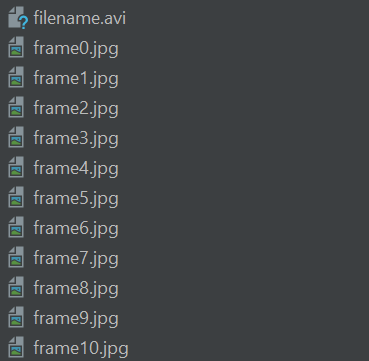
break

count += 1

**Input:**



**Output: Saved in file**









1. **Split and Merge Image**

import cv2  
image = cv2.imread("C:/Users/21r228\Downloads\download.jpg")  
(b\_channel, g\_channel, r\_channel) = cv2.split(image)

#display the images  
cv2.imshow('blue channel',b\_channel)  
cv2.imshow('green channel',g\_channel)  
cv2.imshow('red channel',r\_channel)  
# merge the image

image\_merged = cv2.merge((b\_channel,g\_channel,r\_channel))  
cv2.imshow('merged image',image\_merged)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

**Input Image :**

****

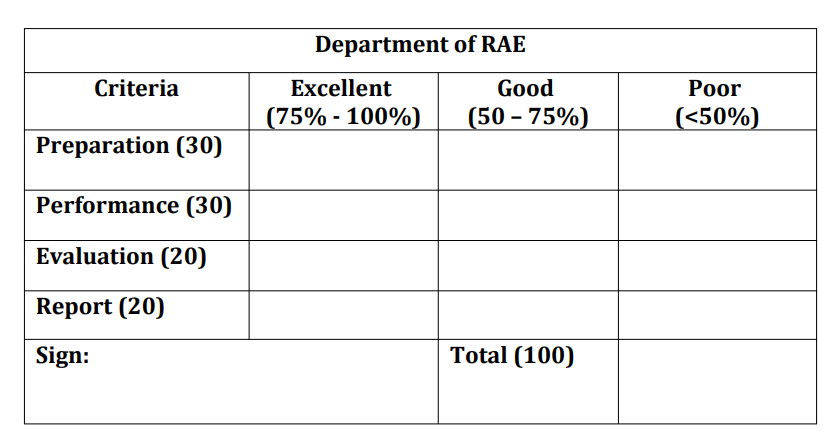
**Output Image :**

**Split Image :**

RED GREEN BLUE



**MERGED IMAGE :**



**Result:**

The basics of OpenCV Image and Video Handling were learnt using OpenCV- python in Pycharm IDE.