Vidyavardhini's College of Engineering & Technology Department of Computer Engineering

Aim: To perform Handling Files, Cameras and GUIs

Objective: To perform Basic I/O Scripts, Reading/Writing an Image File, Converting Between an Image and raw bytes, Accessing image data with numpy.array,Reading /writing a video file, Capturing camera, Displaying images in a window ,Displaying camera frames in a window

Theory:

1. Basic I/O script:

Most CV applications need to get images as input. Most also produce images as output. An interactive CV application might require a camera as an input source and a window as an output destination. However, other possible sources and destinations include image files, video files, and raw bytes. For example, raw bytes might be transmitted via a network connection, or they might be generated by an algorithm if we incorporate procedural graphics into our application. Let's look at each of these possibilities

2. Reading/Writing an Image File:

For reading an image, use the imread() function in OpenCV.

syntax:

imread(filename, flags)

It takes two arguments:The first argument is the image name, which requires a fully qualified pathname to the file.The second argument is an optional flag that lets you specify how the image should be represented. OpenCV offers several options for this flag, but those that are most common include:

example:

imread() helps us read an image

img_grayscale = cv2.imread('test.jpg',0)

write/save an image into the file directory, using the imwrite() function.

syntax:imwrite(filename, image).

The first argument is the filename, which must include the filename extension (for example .png, .jpg etc). OpenCV uses this filename extension to specify the format of the file.

The second argument is the image you want to save. The function returns True if the image is saved successfully imwrite() writes an image into the file directory

example:cv2.imwrite('grayscale.jpg',img_grayscale)

3. Converting Between an Image and raw bytes:

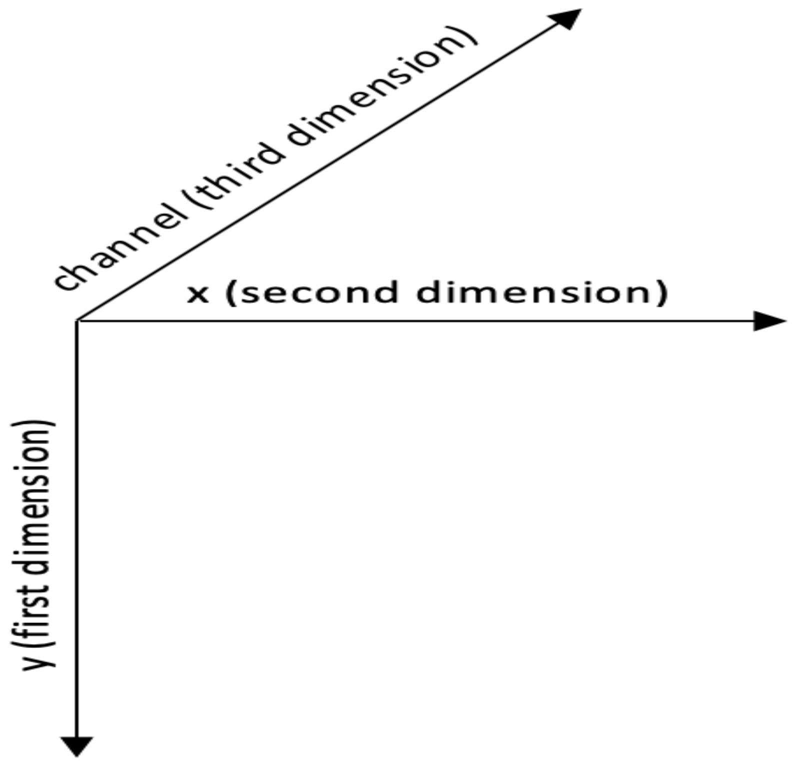
Sometimes, we may want an in-memory jpg or png image that is represented as binary data. But often, what we have got is image in OpenCV (Numpy ndarray) or PIL Image format.Conceptually, a byte is an integer ranging from 0 to 255. Throughout real-time graphic applications today, a pixel is typically represented by one byte per channel, though other representations are also possible.An OpenCV image is a 2D or 3D array of the numpy.array type. An 8-bit grayscale image is a 2D array containing byte values. A 24-bit BGR image is a 3D array.

a)using OpenCv:

```
import cv2

im = cv2.imread('test.jpg')

im_resize = cv2.resize(im, (500, 500))
```

b)Using PIL:

```
from PIL import Image

im = Image.open('test.jpg')

im_resize = im.resize((500, 500))
```



4. Accessing image data with numpy. Array:

Accessing image data with numpy Array: In Python, Pillow is the most popular and standard library when it comes to working with image data NumPy uses the asarray() class to convert PIL images into NumPy arrays. The np.array function also produce the same result. The type function displays the class of an image. The process can be reversed using the Image.fromarray() function. This function comes in handy when the manipulation is performed on numpy.ndarray image data, that we laterwant to save as a PNG or JPEG file.

Syntax:

data = numpy.asarray(image)

5.Reading/Writing a video file :

Reading and writing videos in OpenCV is very similar to reading and writing images. A video is nothing but a series of images that are often referred to as frames. So, all you need to do is loop over all the frames in a video sequence

cv2.VideoWriter – Saves the output video to a directorcv2.VideoCapture – Creates a video capture display the video.
cv2.VideoWriter – Saves the output video to a director

example:

VideoCapture(path,apiPreference)output=
cv2.VideoWriter('Resources/output_video_from_file.avi',
cv2.VideoWriter_fourcc('M','J','P','G'), 20, frame_size)

6.Capturing camera frames:

A frame is an image that forms a single instance of a video. A video consists of a lot of frames running per second (also known as frames per second). For example, a video streaming at 30 fps means that it displays 30 images in a second,using Python language for this purpose, but the OpenCV library

example:

```
vidcap=cv2.VideoCapture(0)
vidcap.isOpened():
ret, frame = vidcap.read()
```

7.Displaying images in a window:

To display image data, use the imshow() function.

imshow(window_name, image)

This function also takes two arguments:

The first argument is the window name that will be displayed on the window. The second argument is the image that you want to display.

example:

```
imshow("moon.tif");
```

Displaying camera frames in a window :

if the frame was successfully captured , we will display the captured frame by using cv2.imshow(windname, frame).

Conclusion:

In this experiment we explored reading , writing images , reading video , writing video's using OpenCv and also accessing images using numpy array and displaying images using imshow() function .