|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Digital Image Processing** | **Semester:** | **VII** |
| **Date of Performance:** |  | **Batch No:** |  |
| **Faculty Name:** |  | **Roll No:** |  |
| **Faculty Sign & Date:** |  | **Grade/Marks:** |  |

**Experiment No: 1**

**Title: To study the basic process of reading and displaying the image and find aspect ratio.**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| To study the basic process of reading and displaying the image and find aspect ratio. |

|  |
| --- |
| **COs to be achieved:** |
| 1. **Understand fundamental theory and models of image processing** |

|  |
| --- |
| **Theory:** |
| A = imread(filename) reads the image from the file specified by filename, inferring the format of the file from its contents. If filename is a multi-image file, then imread reads the first image in the file.    PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The Image module provides a class with the same name which is used to represent a PIL image. The module also provides a number of factory functions, including functions to load images from files, and to create new images.  **PIL.Image.open()** Opens and identifies the given image file.  This is a lazy operation; this function identifies the file, but the file remains open and the actual image data is not read from the file until you try to process the data (or call the load() method).  ***Syntax:****PIL.Image.open(fp, mode=’r’)*  ***Parameters****:*  ***fp****– A filename (string), pathlib.Path object or a file object. The file object must implement read(), seek(), and tell() methods, and be opened in binary mode.* ***mode****– The mode. If given, this argument must be “r”.*  ***Returns type****: An image object.* ***Raises****: IOError – If the file cannot be found, or the image cannot be opened and identified.*  # Imports PIL module  from PIL import Image    # open method used to open different extension image file  im = Image.open(r"C:\Users\System-Pc\Desktop\ybear.jpg")    # This method will show image in any image viewer  im.show()    pip install opencv-python  pip install numpy  pip install matplotlib  The steps to read and display an image in OpenCV are:  1. Read an image using imread() function.  2. Create a GUI window and display image using imshow() function.  3. Use function waitkey(0) to hold the image window on the screen by the specified number of seconds, o means till the user closes it, it will hold GUI window on the screen.  4. Delete image window from the memory after displaying using destroyAllWindows() function.  Let’s start reading an image. using cv2.  To read the images cv2.imread() method is used. This method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.  ***Syntax:****cv2.imread(path, flag)*  ***Parameters:******path:****A string representing the path of the image to be read.* ***flag:****It specifies the way in which image should be read. It’s default value is****cv2.IMREAD\_COLOR***  ***Return Value:****This method returns an image that is loaded from the specified file.*  **Note:**   1. The image should be in the working directory or a full path of image should be given. 2. By default, OpenCV stores colored images in BGR(Blue Green and Red) format.   All three types of flags are described below:  ***cv2.IMREAD\_COLOR:****It specifies to load a color image. Any transparency of image will be neglected. It is the default flag. Alternatively, we can pass integer value****1****for this flag.* ***cv2.IMREAD\_GRAYSCALE:****It specifies to load an image in grayscale mode. Alternatively, we can pass integer value****0****for this flag.* ***cv2.IMREAD\_UNCHANGED:****It specifies to load an image as such including alpha channel. Alternatively, we can pass integer value****-1****for this flag.*  Below codes are implementations to read images and display images on the screen using OpenCV and matplotlib libraries functions.   **Example #1 (Using OpenCV) :**  Image used is:      # Python code to read image  import cv2    # To read image from disk, we use  # cv2.imread function, in below method,  img = cv2.imread("geeksforgeeks.png", cv2.IMREAD\_COLOR)    # Creating GUI window to display an image on screen  # first Parameter is windows title (should be in string format)  # Second Parameter is image array  cv2.imshow("image", img)    # To hold the window on screen, we use cv2.waitKey method  # Once it detected the close input, it will release the control  # To the next line  # First Parameter is for holding screen for specified milliseconds  # It should be positive integer. If 0 pass an parameter, then it will  # hold the screen until user close it.  cv2.waitKey(0)    # It is for removing/deleting created GUI window from screen  # and memory  cv2.destroyAllWindows() |

|  |
| --- |
| **Stepwise-Procedure:** |
| 1. Click selfie through your phone 2. Get image into the PC on which you are working using gmail 3. Read image using MATLAB online   A=imread(“Path of the image”)   1. Install Thonny IDE 2. Install required packages (PIL, Opencv-contrib-python) through Tools🡪 manage packages 3. Go to tools🡪Open system shell install PIP and upgrade |

|  |
| --- |
| **Output** |
| 1. Upload picture screenshots for all 3 approches   MATLAB  PIL  OPENCV  Answer the following questions:   1. What is the maximum resolution of your phone as per data sheet. 2. Calculate aspect ratio of your clicked images |

|  |
| --- |
| **Conclusions:** |
|  |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. What is aspect ratio in digital images 2. What are different camera sizes in MP are available for different commercial mobile phones |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |