**Shri Ramdeobaba College of Engineering and Management, Nagpur-13.   
Department of Computer Science Engineering (AIML)  
CAP-309 - Fundamental of Digital Image & Video Processing  
Even Semester – 2023-24**

| **Name:** | Om Ghumre | | | | |
| --- | --- | --- | --- | --- | --- |
| **Batch / Roll No.** | E3-43 | | | | |
| **Semester/Section:** | VI / E | | | | |
| **Date of Performance:** | 1 Feb 24 | | | | |
| **Date of Submission:** |  | | | | |
| **Particulars** | **Experiment Performance** | **Result & Discipline** | **Viva** | **Journal** | **Total** |
| **Max. Marks** | **03** | **03** | **03** | **01** | **10** |
| **Marks Obtained** |  |  |  |  |  |
| **Name &**  **Signature of Faculty** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |

Practical No:- 01

**Aim:-** Lab1: To study and perform basic operations used in image processing.

i) Reading an image. ii) Displaying an image in color mode.

iii) Displaying an image in grayscale mode. iv) Saving an image

**Software used:-** Pycharm/Jupyter/Colab

**Theory:-**

i) Reading an Image:

Reading an image involves loading a digital image file into memory using functions like cv2.imread() in OpenCV. The image is then represented as a NumPy array, enabling subsequent processing and analysis.

ii) Displaying an Image in Color Mode:

Displaying an image in color mode is achieved with the cv2.imshow() function, creating a graphical window to showcase the image's RGB components. Paired with cv2.waitKey() and cv2.destroyAllWindows(), it offers an interactive way to visualize color images.

iii) Displaying an Image in Grayscale Mode:

Converting an image to grayscale with cv2.cvtColor() and then displaying it using cv2.imshow() provides a single-channel representation of intensity. This grayscale mode simplifies image analysis and visualization.

iv) Saving an Image:

Saving an image is crucial after processing. The cv2.imwrite() function allows you to store the modified image back to a file, ensuring that the processed result is preserved for future use.

**Code:-**

| import cv2  import numpy as np  image=cv2.imread("image.jpg")  gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  cv2.imwrite('grey.jpg',gray\_image)  cv2.imshow("Image",image1)  cv2.waitKey(0) |
| --- |

**Input image:-**

|  |
| --- |

**Output image:-**

|  |
| --- |

**Conclusion:-** In this practical i have learned to read a image, print the image, convert the image to grayscale and save the image.