Ex. No: 1 Date:	Image Processing Techniques from Matrix Representation to Manipulation

Aim:

To perform various image processing techniques from matrix representation to image manipulation.

Software Required:

- Anaconda

Task 1: Create a 2D matrix of size (8x8) with values 0's and 1's and display it as an image

Procedure:

1. Open PyCharm and create a new Python project.
2. Create a new Python file within your project.
3. Import necessary libraries: import numpy as np and import matplotlib.pyplot as plt.
4. Generate a 2D matrix of size (8x8) with random 0's and 1's using np.random.randint(0, 2, size=(8, 8)).
5. Display the matrix as an image using plt.imshow(matrix, cmap='gray').
6. Use plt.show() to display the image.

Code:

Task 2: Create a 2D matrix of size (8x8) with values 0 to 255 and display it as an image

Procedure:

1. Follow steps 1-3 from Task 1.
2. Generate a 2D matrix of size (8x8) with values 0 to 255 using np.random.randint(0, 256, size=(8, 8)).
3. Display the matrix as an image using plt.imshow(matrix, cmap='gray', vmin=0, vmax=255).
4. Use plt.show() to display the image.

Code:

Task 3: Read a RGB image, separate the channels, and display them separately

Procedure:

1. Follow steps 1-3 from Task 1.
2. Import OpenCV library: import cv2.
3. Read an RGB image using cv2.imread('path_to_your_image.jpg').
4. Separate the image into its blue, green, and red channels using b, g, r = cv2.split(image).
5. Display each channel separately using plt.imshow(channel, cmap='gray') for each channel (blue, green, red).
6. Use plt.show() to display the channels.

Code:

Task 4: Crop, resize, rotate, and flip an image

Procedure:

1. Follow steps 1-3 from Task 1.
2. Read an image using cv2.imread('path_to_your_image.jpg').
3. Perform cropping, resizing, rotation, and flipping using OpenCV functions (cv2.resize, cv2.rotate, cv2.flip).
4. Display the original and processed images using cv2.imshow() and cv2.waitKey(0) for each image.
5. Use cv2.destroyAllWindows() to close all windows when done.

Code:

Result:

Ex. No: 2 Date:	Implementation of advanced Image Manipulation Operations

Aim:

To implement advanced image manipulation operations like bitwise operations of two images, image interpolation, image blurring, and edge detection.

Software Required:

- Anaconda

Task 1: Perform Bitwise Operations on images – AND, OR and XOR

Procedure:

Code:

Task 2: Perform image blurring on images with the help of various low pass filter kernels:

- Gaussian Blur
- Median Blur
- Average Blur

Procedure:

Code:

Task 3: Perform image interpolation and apply it to image resizing (shrinking and zooming)

Procedure:

Code:

Task 4: Perform Edge Detection on images using Canny Operator.

Procedure:

Code:

Result:

Ex. No: 3 Date:	Implementation of Image Enhancement techniques like sharpening and thresholding

Aim:

To implement image enhancement techniques such as brightness and contrast adjustment, image sharpening and image thresholding.

Software Required:

- Anaconda

Task 1: To adjust the brightness and contrast of the images to enhance the visual appeal and effectiveness.

Procedure:

Code:

Task 2: To perform image sharpening to enhance the edges and fine details in an image.

Procedure:

Code:

Task 3: To perform image segmentation using simple, adaptive and Otsu thresholding methods.

Procedure:

Code:

Result:

Ex. No: 4 Date:	Implementation of Image Enhancement in Spatial Domain using Histogram Processing

Aim:

To implement image enhancement in spatial domain using histogram processing such as histogram equalization and histogram matching.

Software Required:

- Anaconda

Task 1: To verify the histograms of dark, light, low contrast and high contrast images.

Procedure:

Code:

Task 2: To perform histogram equalization to enhance the contrast of the image.

Procedure:

Code:

Task 3: To perform histogram matching to adjust the histogram of an image to match a specified target histogram.

Procedure:

Code:

Result:

Ex. No: 5 Date:	Implementation of Intensity Transformation Operations on Images

Aim:

To implement Intensity Transformation Operations on Images.

Software Required:

- Anaconda

Task 1: To perform commonly used intensity transformations on images:

- Image Negatives (Linear)
- Log Transformations
- Power-Law (Gamma) Transformations

Procedure:

Code:

Task 2: To perform Piecewise-Linear Transformation Functions on images.

- Contrast Stretching
- Intensity level slicing
- Bit-plane slicing

Procedure:

Code:

Result:

Ex. No: 6 Date:	Implementation of Intensity Transformation Operations on Images

Aim:

a) To apply Fourier Transform to an image and determine the magnitude spectrum.

b) To implement various image restoration techniques such as bandpass, bandreject, and Notch filters, Optimum Notch, Inverse and Wiener filtering.

Software Required:

- Anaconda

Task 1: To apply Fourier Transform to an image and determine the magnitude spectrum.

Procedure:

Code:

Task 2: To implement various image restoration techniques such as:

- Bandpass, bandreject, and Notch filters,
- Optimum Notch, Inverse and Wiener filtering.

Procedure:

Code:

Result:

Ex. No: 7 Date:	Implementation of Image Segmentation Techniques

Aim:

a) To perform basic morphological operations on images.

b) To implement region-based image segmentation techniques.

c) To implement Watershed Segmentation algorithm.

Software Required:

- Anaconda

Task 1: To implement basic Morphological operations on images such as erosion, dilation, opening and closing.

Procedure:

Code:

Task 2: To implement region-based image segmentation techniques such as Region Growing, and Region Splitting and Merging.

Procedure:

Code:

Task 3: To perform image segmentation using Watershed Segmentation algorithm.

Procedure:

Code:

Result:

Ex. No: 8 Date:	Implementation of 3D Vision Techniques

Aim:

a) To implement camera calibration with Chessboard and Circular grid.

b) To detect crack in the image using “shape from shading” technique.

c) To calculate disparity between two images (using stereoBM class)

Software Required:

- Anaconda

Task 1: To implement camera calibration with Chessboard and Circular grid.

Procedure:

Code:

Task 2: To detect crack in the image using “shape from shading” technique.

Procedure:

Code:

Task 3: To calculate disparity between two images (using stereoBM class).

Procedure:

Code:

Result:

Ex. No: 9 Date:	Implementation of 3D Vision – Structure from Motion Techniques

Aim:

To implement 3D vision's structure from motion techniques like bundle adjustment method, parametric motion and spline-based motion.

Software Required:

- Anaconda

Task 1: To implement 3D vision's structure using bundle adjustment method

Procedure:

Code:

Task 2: To implement 3D vision's structure using parametric motion method

Procedure:

Code:

Task 3: To implement 3D vision's structure using spline-based motion method

Procedure:

Code:

Result:

Ex. No: 10 Date:	Implementation of Real Time Applications – Document Image Analysis, Video Tracking

Aim:

To implement real time applications like Document Image Analysis to detect and process text and document boundaries, and Video Tracking to follow a specific object in a video feed.

Software Required:

- Anaconda

Task 1: To implement real-time Document Image Analysis

Procedure:

Code:

Task 2: To implement real-time video tracking

Procedure:

Code:

Result:

Ex. No: 11 Date:	Implementation of Real Time Applications - Object Recognition, Content-Based Image Retrieval

Aim:

To implement of Real Time Applications like Object Recognition, Content-Based Image Retrieval.

Software Required:

- Anaconda

Task 1: To implement real-time object recognition using a pre-trained deep learning model.

Procedure:

Code:

Task 2: To implement real-time Content-Based Image Retrieval using feature extraction and similarity measurement.

Procedure:

Code:

Result: