# **Basic Functions of CV**

## **Introduction**

This project demonstrates the use of **OpenCV** for basic image processing tasks. OpenCV is a popular library for computer vision, image processing, and numerical operations. This assignment focuses on performing fundamental operations on images such as reading, displaying, slicing, drawing shapes, and saving images.

## **Objective**

* To read and display an image using OpenCV.
* To retrieve image properties such as dimensions and channels.
* To perform image slicing.
* To draw various shapes on the image (rectangle, circle, line).
* To add text annotations.
* To save the modified image.

## **Tools & Libraries Used**

| **Tool / Library** | **Purpose** |
| --- | --- |
| OpenCV (cv2) | Image processing library |
| NumPy | Numerical operations |
| Matplotlib | Display images inside Jupyter Notebook (optional) |

## **Methodology**

The project follows these simple steps:

1. Load the image from a file.
2. Display the image using OpenCV functions.
3. Fetch image properties such as height, width, and number of channels.
4. Perform image slicing to extract a portion of the image.
5. Draw shapes (rectangle, circle, lines) on the image.
6. Add text annotations.
7. Save the resulting image to disk.

## **Implementation**

**Step 1: Import Required Libraries**

* import cv2
* import numpy as np
* from matplotlib import pyplot as plt

**Step 2: Read an Image**

* image = cv2.imread("img.jpg")
* cv2.imshow("Original Image", image)
* cv2.waitKey(0)
* cv2.destroyAllWindows()

**Step 3: Get Image Properties**

* print("Image Height:", image.shape[0])
* print("Image Width:", image.shape[1])
* print("Number of Channels:", image.shape[2])

**Step 4: Image Slicing**

* slice\_image = image[100:300, 200:400]
* cv2.imshow("Sliced Image", slice\_image)
* cv2.waitKey(0)
* cv2.destroyAllWindows()

**Step 5: Draw Rectangle**

* cv2.rectangle(image, (200,100), (400,300), (255,0,0), 2)
* cv2.imshow("Rectangle", image)
* cv2.waitKey(0)
* cv2.destroyAllWindows()

**Step 6: Draw Circle**

* cv2.circle(image, (300,200), 50, (0,255,0), 3)
* cv2.imshow("Circle", image)
* cv2.waitKey(0)
* cv2.destroyAllWindows()

**Step 7: Draw Line**

* cv2.line(image, (0,0), (500,500), (0,0,255), 3)
* cv2.imshow("Line", image)
* cv2.waitKey(0)
* cv2.destroyAllWindows()

**Step 8: Add Text**

* font = cv2.FONT\_HERSHEY\_SIMPLEX
* cv2.putText(image, "OpenCV Demo", (50,50), font, 1, (255,255,255), 2)
* cv2.imshow("Text", image)
* cv2.waitKey(0)
* cv2.destroyAllWindows()

**Step 9: Save Modified Image**

* cv2.imwrite("output.jpg", image)
* print("Image saved as output.jpg")

## **Results**

The following tasks were successfully implemented:

* Displaying the original image.
* Extracting a portion of the image using slicing.
* Drawing shapes like rectangles, circles, and lines.
* Adding text to the image.
* Saving the final image to the disk.

All operations were performed successfully and verified visually.

## **Conclusion**

This assignment demonstrated how to use OpenCV for basic image processing operations. We have learned how to:

* Read and display images.
* Access image properties.
* Slice images.
* Draw geometric shapes.
* Add text.
* Save images to disk.

This project serves as a foundation for more advanced image processing and computer vision tasks.