



SUPERIOR UNIVERSITY

NAME :Muhammad ilyas

Roll number:033

Section:AI(4A)

SUBJECT:PAI-Lab

SUBMITTED TO :Rasikh Ali

Lab task 5:

Image Processing and Computer Vision using OpenCV

Introduction:

This project focuses on various image processing and computer vision techniques using OpenCV, a popular Python library. The objective of this assignment is to explore different image operations such as basic image handling, arithmetic operations, transformations, filtering, feature detection, and video processing.

1. Image Handling

The first step in the project was loading and displaying images. OpenCV's `cv2.imread()` function was used to read an image, and `cv2.imshow()` was used to display it. Additionally, images were saved using `cv2.imwrite()`.

2. Image Arithmetic Operations

- **Addition and Subtraction:** Two images were combined using `cv2.add()`, and differences were computed using `cv2.subtract()`.
- **Bitwise Operations:** Bitwise AND, OR, XOR, and NOT operations were applied to images to analyze pixel-wise manipulation.

3. Image Transformations

- **Resizing:** Images were resized to different scales using `cv2.resize()`.
- **Erosion:** Morphological transformation using `cv2.erode()` was applied to remove noise.
- **Blurring:** Various blurring techniques like Gaussian Blur, Median Blur, and Bilateral Filter were implemented to smooth images.

4. Image Analysis

- **Histogram Analysis:** A histogram of pixel intensity distribution was plotted using `cv2.calcHist()`.
- **Thresholding:** Different thresholding techniques were applied, including binary and inverse binary thresholding, to segment images.

5. Color Space Conversion and Splitting

- Images were converted from BGR to grayscale and HSV using `cv2.cvtColor()`.
- The image channels were split using `cv2.split()` to extract individual color components.

6. Feature Detection

- **Circle Detection:** The Hough Circle Transform method was used to detect circular objects in images.

- **Corner Detection:** The Harris Corner Detection algorithm was applied to find sharp corners.
- **Blob Detection:** Circular blobs were detected using `cv2.SimpleBlobDetector_Params()`.

7. Object Detection

- **Face and Smile Detection:** Haar cascade classifiers were used to detect faces and smiles in real-time from a webcam feed.

8. Drawing on Images Different shapes such as rectangles, circles, ellipses, and lines were drawn on images using OpenCV functions like `cv2.rectangle()`, `cv2.circle()`, `cv2.ellipse()`, and `cv2.arrowedLine()`. Text was also added using `cv2.putText()`.

9. Video Processing

- A video was read and displayed frame by frame using `cv2.VideoCapture()`.
- A video was created using multiple images and saved in AVI format.

Conclusion: This project provided hands-on experience with OpenCV and helped in understanding various image processing techniques. The application of these techniques is vast, ranging from object detection to real-time video analysis. OpenCV proves to be a powerful tool for computer vision applications.