

Ex No: 1

3-1-2026

Experimenting OpenCV installation and working with Python

AIM:-

To install OpenCV and perform basic image operations such as reading, displaying and saving an image using Python

ALGORITHM:-

1. Start Python program
2. import CV2 module
3. Load an image using cv2.imread()
4. Display the image using cv2.imshow()
5. convert the loaded image to grayscale using cv2.cvtColor()
6. Display the grayscale image
7. Save the grayscale image using cv2.imwrite()
8. Close all image windows using cv2.destroyAllWindows()

Program:-

for i : image:

import CV2

img = CV2.imread("image.jpg")

CV2.imshow("Original Image", img)

gray = CV2.cvtColor(img, CV2.COLOR_BGR2GRAY)

CV2.imshow("Grayscale Image", gray)

CV2.imwrite("gray-output.jpg", gray)

CV2.waitKey(0)

CV2.destroyAllWindows()

for 3 images..

```
import cv2
```

```
images = ["P1.jpg", "P2.jpg", "P3.jpg"]
```

for i, img_name in enumerate(images):

```
    img = cv2.imread(img_name)
```

If img is None:

```
    print(f"Could not read {img_name}")
```

Continue

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
cv2.imshow(f"original Image {i+1}", img)
```

```
cv2.imshow(f"Grayscale Image {i+1}", gray)
```

cv2.waitKey(0)

~~cv2.destroyAllWindows()~~

D
3/1/26

Result:

The OpenCV library was successfully installed in Python. An image was read, displayed, converted into grayscale.

Ex No:2
24. 01. 26

Basic Image Processing - loading Images, cropping, Resizing, Thresholding, contour analysis, Blob detection

Aim:-

To Study and perform basic Image processing operations such as loading an image, cropping, resizing, thresholding, contour detection and blob detection using opencv in Python

Pre requisite commands:-

Installation commands (Run in cmd / powershell)

Pip install opencv-python

pip install numpy

Basic concepts Required

- Pixel matrix representation
 - Image Indexing (Slicing)
- Thresholding Fundamentals
- contours
- blob properties

Operation	Command
Load image	cv2.imread()
Display image	cv2.imshow()
Cropping	image[y1:y2, x1:x2]
Resize	cv2.resize
Thresholding	cv2.threshold()
Find contours	cv2.findContours()

Algorithm:

1. Loading our Image:

- import OpenCV
- load an image using CV2.imread()
- Display using cv2.imshow()

2. Cropping

- Select 4 coordinates for cropping
- crop using slicing → cropped = img[y1:y2, x1:x2]

3. Resizing

- use CV2.resize()
- provide new width and height

4. Thresholding

- Convert image to grayscale
- Apply binary thresholding
- Get thresholded image

5. Contour detection

- Convert image to grayscale
- Apply threshold
- use CV2.findContours()
- Draw contours on original image

6. Blob Detection:

- Set blob parameters
- Create blob detector
- Detect blobs
- Draw detected blobs on image

Program:

- Import CV2

Import numpy as np

img = CV2.imread("C:/Users/Admin/Desktop/lab/PP.jpg")

- CV2.imshow("Original Image", img)

crop = img[100:300, 150:400]

CV2.imshow("Dropped Image", crop)

resize = CV2.resize(img, (400, 400))

CV2.imshow("Resized Image", resize)

gray = CV2.cvtColor(img, CV2.COLOR_BGR2GRAY)

ret, thresh = CV2.threshold(gray, 120, 255, CV2.THRESH_BINARY)

CV2.imshow("Thresholded Image", thresh)

contours, hierarchy = CV2.findContours(thresh, CV2.RETR_EXTERNAL)

CV2.drawContours(contour_img, contours, -1, (0, 255, 0), 3)

contour_img = img.copy()

CV2.drawContours(contour_img, contours, -1, (0, 255, 0), 3)

CV2.imshow("contours", contour_img)

Params = CV2.SimpleBlobDetector_Params()

Params.filterByArea = true

Params.minArea = 50

detector = CV2.SimpleBlobDetector_create(Params)

keypoints = detector.detect(gray)

blobs_img = CV2.drawKeypoints(img, keypoints, np.array([]), (0, 0, 255), CV2.DRAW)

```
cv2.imread("Blob Detection", blob-ing)
cv2.imwrite("cropped.jpg", crop)
cv2.imwrite("resized.jpg", resize)
cv2.imwrite("threshold.jpg", thresh)
cv2.imwrite("contours.jpg", contouring)
cv2.imwrite("blobs.jpg", blob-ing)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Result:

✓ 24/11/26

Basic image processing operations such as loading, cropping, resizing, thresholding, contour detection and blob detection were successfully implemented using python Open CV

Ex No: 3
2H .01.26

Construction of Image Annotation -

Drawing line, Text & circle, Rectangle,
Ellipse on Images

Dim.

To perform image annotation using open cv by
drawing line, rectangle, circles, Ellipses and
adding text on an image

One requisites:-

1. Open CV library installed
2. A Sample image available in the working directory

Algorithm:-

1. Import the CV library
2. Load the image using `CV2.imread()`
3. Check if the image is loaded correctly. If not, terminate the program
4. Draw the following shapes on the image:
 - Line using `CV2.line()`
 - Rectangle using `CV2.rectangle()`
 - Circle using `CV2.circle()`
 - Ellipse using `CV2.ellipse()`
5. Add text on the image using `CV2.putText()`
6. Display the annotated image

Image

Output
of "Annotation" below

Program:

import cv2

img = cv2.imread(r'C:\Users\Aelmin\ppp.jpg')

cv2.line(img, (50, 50), (300, 50), (0, 0, 255), 3)

cv2.rectangle(img, (50, 100), (300, 150), (0, 255, 0), 3)

cv2.circle(img, (200, 350), 60, (255, 0, 0), 4)

cv2.ellipse(img, (200, 500), (100, 50), 0, 0, 360,
(255, 255, 0), 3)

cv2.putText(img, "OpenCV Annotation", (50, 600),

cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), 3)

cv2.imshow("Annotated Image", img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Result:

The program loads the image dam.jpg and displays it with the following annotations

1. A red line at the top

2. A green rectangle below the line

3. A blue circle in the middle of the image

4. A cyan ellipse at the bottom.

5. Yellow text "OpenCV Annotation" between