

# ASSIGNMENT

- Name : Achal Rajesh Mate
- Roll No : 2203541
- Enroll No : MITU20BTCSD001
- Branch : CSE
- Class : TY CSE Is - 3
- Guided By : Prof Nagesh Jadhav Sir

**Title:-** Perform basic image processing using opencv

**Objectives:-**

1. To learn about images processing
2. To perform various operations on images

## OpenCV

- OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.
- Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel).
- The library is cross-platform and free for use under the open-source BSD license.
- OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers.
- It promotes OpenVisionCapsules, which is a portable format, compatible with all other formats.

## Image processing operations

- Digital image processing is the use of a digital computer to process digital images through an algorithm.
- As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing.
- It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and distortion during processing.

- Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.
- The generation and development of digital image processing are mainly affected by three factors: first, the development of computers; second, the development of mathematics (especially the creation and improvement of discrete mathematics theory); third, the demand for a wide range of applications in environment, agriculture, military, industry and medical science has increased.
- Digital image processing allows the use of much more complex algorithms, and hence, can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analogue means.
- In particular, digital image processing is a concrete application of, and a practical technology based on:
  - Classification
  - Feature extraction
  - Multi-scale signal analysis
  - Pattern recognition
  - Projection
- Some techniques which are used in digital image processing include:
  - Anisotropic diffusion
  - Hidden Markov models
  - Image editing
  - Image restoration
  - Independent component analysis
  - Linear filtering
  - Neural networks
  - Partial differential equations
  - Pixelation
  - Point feature matching
  - Principal components analysis
  - Self-organizing maps
  - Wavelets

## Import All Necessary libray

```
In [2]: import cv2 as cv
```

```
import numpy as np
from google.colab.patches import cv2_imshow
```

## Read The Image

```
In [9]: image=cv.imread("download.jfif")
# gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
```

## Display the image

```
In [10]: cv2_imshow(image)
cv.waitKey(0)
```



```
Out[10]: -1
```

## View Dimension of Image

```
In [11]: image.shape
```

```
Out[11]: (194, 259, 3)
```

```
In [12]: corner=image[0:100,0:100]
```

```
In [13]:
```

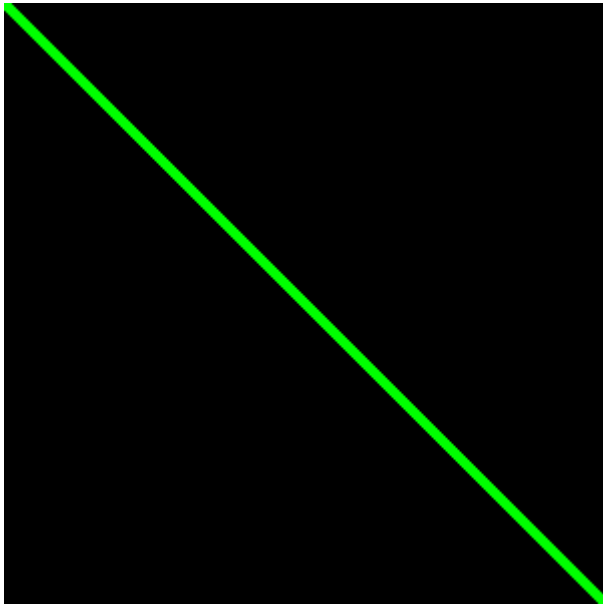
```
cv2.imshow('corner')  
cv.waitKey(0)
```



Out[13]: -1

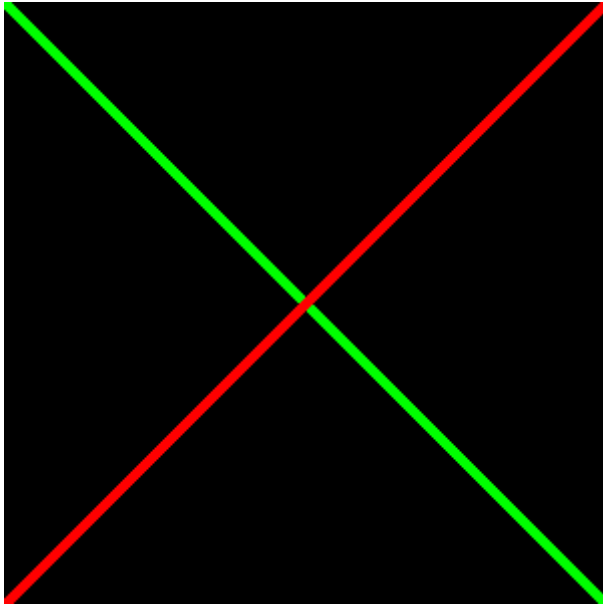
Drawing a stright **line**

```
In [14]: canvas = np.zeros((300, 300, 3), dtype = "uint8")  
         #RGB(255,255,255)  
         green = (0, 255, 0)  
         cv.line(canvas, (0, 0), (300, 300), green,4)  
         cv2.imshow(canvas)  
         cv.waitKey(0)
```



Out[14]: -1

```
In [15]: red = (0, 0, 255)
cv.line(canvas, (300, 0), (0, 300), red, 3)
cv2.imshow(canvas)
cv.waitKey(0)
```



Out[15]: -1

## Image Processing

```
In [16]: # Image Translation
#The first row of the matrix is [1, 0, tx], where tx is the number of pixels we will shift the image left or right.
#Negative values of tx will shift the image to the left and positive values will shift the image to the right
M = np.float32([[1, 0, 25], [0, 1, 50]])
print(image.shape[0])
print(image.shape[1])
shifted = cv.warpAffine(image, M, (image.shape[1], image.shape[0]))
cv2.imshow(shifted)
cv.waitKey(0)
```

194  
259



Out[16]: -1

## Rotate the image

```
In [17]: # Rotation
cv2_imshow(image)
(h, w) = image.shape[:2]
#h=image.shape[0]
#w=image.shape[1]

center = (w // 2, h // 2)
M = cv.getRotationMatrix2D(center, 45, 1.0)
rotated = cv.warpAffine(image, M, (w, h))
print("Rotated by 45 Degrees")
cv2_imshow(rotated)
cv.waitKey(0)
```



Rotated by 45 Degrees



Out[17]: -1

## Change Dimensions of Image

```
In [18]: cv2_imshow(image)
print(image.shape)
r = 150.0 / image.shape[1]
dim = (150, int(image.shape[0] * r))
resized = cv.resize(image, dim, interpolation = cv.INTER_AREA)
print(resized.shape)
cv2_imshow(resized)
cv.waitKey(0)
```



```
(194, 259, 3)
(112, 150, 3)
```



Out[18]: -1

## Flip the image

In [19]:

```
cv2_imshow(image)
flipped = cv.flip(image, 0)
cv2_imshow(flipped)
cv.waitKey(0)
```



Out[19]: -1



## Display grey Image

```
In [20]: i_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY) #BGR to gray conversion  
cv2.imshow(i_gray)  
cv.waitKey(0)
```



Out[20]: -1

## Display RBG image

```
In [21]: i_rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB) #BGR to RGB (useful for matplotlib)  
cv2.imshow(i_rgb)  
cv.waitKey(0)
```



Out[21]: -1

## Display Normalize Image

```
In [22]: i = cv.normalize(image, None, 0, 255, cv.NORM_MINMAX, cv.CV_8U) #Normalizes I between 0 and 255  
cv2.imshow(i)  
cv.waitKey(0)
```



Out[22]: -1

## Conclusion

Thus I have Successfully learn about images processing and perform various operations on images using OpenCV

In [ ]: