LAB 01# OpenCV Library Overview

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

In this section, we will learn some functions used in the OpenCV library in colab.

**Tasks:**

* Show Image
* Change RGB to Gray scale
* Write Image
* Read Gray Image
* Edit Image
* Read and Write Image
* Margin Two Image
* Create Black and White Picture
* Normalization Picture Pixel
* Image Flip
* Thresholding
* ROI (Region of Interest)
* Bilateral Filter
* Edge Filtering
* Sobel Filter
* Canny Edge Detection
* Gaussian Blur

**Import necessary libraries.**

import cv2 as cv

## import numpy as np

from google.colab.patches import cv2\_imshow

from matplotlib import pyplot as plt

**Showing an image.**

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

cv2\_imshow(img)

**Changing RGB to Grayscale**

gray\_img = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)

cv2\_imshow(gray\_img)

## **Write Image**

cv.imwrite("women\_face\_gray.png", gray\_img)

## **Edit Image**

img\_copy = np.copy(img)

img\_copy[100:200, 200:300, :] = 0

cv2\_imshow(img\_copy)

## **Read and Write Image**

h, w, ch = img.shape

for row in range(h):

  for col in range(w):

    b, g, r = img[row, col]

    b = 255 - b

    g = 255 - g

    r = 255 - r

    img[row, col] = [b, g, r]

cv2\_imshow(img)

## **Mergin Two Image**

img1 = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

img2 = cv.imread("/content/women\_face\_gray.png")

merge\_image = np.hstack((img1, img2))

cv2\_imshow(merge\_image)

## **Create Black and White Picture**

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg", cv.IMREAD\_GRAYSCALE)

means, stdDev = cv.meanStdDev(img)

img[np.where(img < means)] = 0

img[np.where(img > means)] = 255

cv2\_imshow(img)

## **Normalization Picture Pixel**

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

gray\_img = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)

gray\_img = np.float32(gray\_img)

print(gray\_img)

## MinMax Normalization

dst = np.zeros(gray\_img.shape, dtype=np.float32)

cv.normalize(gray\_img, dst=dst, alpha=0, beta=1.0, norm\_type=cv.NORM\_MINMAX)

cv2\_imshow(np.uint(dst\*255))

## Norm Inf Normalization

dst = np.zeros(gray\_img.shape, dtype=np.float32)

cv.normalize(gray\_img, dst=dst, alpha=1.0, beta=0, norm\_type=cv.NORM\_INF)

cv2\_imshow(np.uint(dst\*255))

## Norm L1 Normalization

dst = np.zeros(gray\_img.shape, dtype=np.float32)

cv.normalize(gray\_img, dst=dst, alpha=1.0, beta=0, norm\_type=cv.NORM\_L1)

cv2\_imshow(np.uint(dst\*10000000))

## Norm L2 Normalization

dst = np.zeros(gray\_img.shape, dtype=np.float32)

cv.normalize(gray\_img, dst=dst, alpha=1.0, beta=0, norm\_type=cv.NORM\_L2)

cv2\_imshow(np.uint(dst\*10000))

## **Image Flip**

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

# X-Flip 0

dst1 = cv.flip(img, 0)

cv2\_imshow(dst1)

# Y-Flip 1

dst1 = cv.flip(img, 1)

cv2\_imshow(dst1)

# XY-Flip -1

dst1 = cv.flip(img, -1)

cv2\_imshow(dst1)

## **Draw Geometry**

img = np.zeros((512, 512, 3),dtype=np.uint8)

cv.rectangle(img, (100,100),(300,300),(255,0,0),2,cv.LINE\_8,0)

cv.circle(img, (256,26),50,(0,0,255),2, cv.LINE\_8,0)

cv.ellipse(img, (256,256),(200,100),360,0,360,(0,255,0),2,cv.LINE\_8,0)

cv2\_imshow(img)

## **Thresholding**

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

th = 127

img\_gray = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)

for i in range(5):

  ret, binary = cv.threshold(img\_gray, th, 255, i)

  cv2\_imshow(binary)

# Geometric

 # Shifting

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

rows = img.shape[0]

cols = img.shape[1]

M =np.float32([[1,0,100], [0,1,90]])

shifted = cv.warpAffine(img, M, (cols, rows))

cv2\_imshow(shifted)

# rotation

rows = img.shape[0]

cols = img.shape[1]

M = cv.getRotationMatrix2D((cols/2, rows/2),90,1)

rotation = cv.warpAffine(img, M, (cols, rows))

cv2\_imshow(rotation)

# scaling

img = cv.resize(img, None, fx=0.2, fy=0.2, interpolation=cv.INTER\_CUBIC)

cv2\_imshow(img)

# ROI (Region of Interest)

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

h, w = img.shape[:2]

roi = img[100:200, 200:300, :]

cv2\_imshow(roi)

# Bilateral Filter

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

dst = cv.bilateralFilter(img, 0, 100, 30)

h, w = img.shape[:2]

result = np.zeros([h, w\*2, 3], dtype=img.dtype)

result[0:h, 0:w, :] = img

result[0:h, w:2\*w, :] = dst

cv2\_imshow(result)

# Edge Filtering

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

dst = cv.edgePreservingFilter(img, sigma\_s=50, sigma\_r=0.4, flags=cv.RECURS\_FILTER)

h, w = img.shape[:2]

result = np.zeros([h, w\*2, 3], dtype=img.dtype)

result[0:h, 0:w, :] = img

result[0:h, w:2\*w, :] = dst

cv2\_imshow(result)

# Sobel Filter

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

h, w = img.shape[:2]

x\_grade = cv.Sobel(img, cv.CV\_32F, 1, 0)

y\_grade = cv.Sobel(img, cv.CV\_32F, 0, 1)

x\_grade = cv.convertScaleAbs(x\_grade)

y\_grade = cv.convertScaleAbs(y\_grade)

dst =cv.add(x\_grade, y\_grade, dtype=cv.CV\_16S)

dst = cv.convertScaleAbs(dst)

cv2\_imshow(dst)

# Canny Edge Detection

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

dst = cv.Canny(img, 150, 250)

cv2\_imshow(dst)

# Gaussian Blur

img = cv.imread("/content/drive/MyDrive/Colab Notebooks/CV Lab/women-face.jpg")

dst = cv.GaussianBlur(img, (3,3), 0)

cv2\_imshow(dst)