CLASS : T.E. E &TE **SUBJECT: DIP**

EXPT. NO. : 1 **DATE:**

TITLE: IMAGE PROCESSING TOOLBOX - OPENCV

CO 1:	Apply the fundamentals of digital image processing to perform various image-enhancement and image segmentation operations on gray scale image.

AIM:

To implement: Various OpenCV commands using Python.

SOFTWARES REQUIRED: Jupyter Notebook or Google Colaboratory.

THEORY:

OpenCV was started at Intel in 1999 by Gary Bradsky, and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.

OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language. OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

1.1 Installing OpenCV

- Download latest OpenCV release from GitHub or Source Forge site and double-click to extract it.
- Then goto opencv/build/python/ (the corresponding folder).
- Copy cv2.pyd to C:/Python27/lib/site-packages.
- Open Python IDLE and type following codes in Python terminal: -

```
>>> import cv2 as cv
>>> print (cv. version__)
```

- If the results are printed correctly then, the it is installed correctly

1.2 OpenCV Commands

a. **Accessing an image**: Function cv2.imread() is used to read an image using OpenCv.

Syntax: cv.imread(filename[, flags])

Flag vales can be any of the following: -

1 : for reading image in the form of 3 channel RGB.

0 : for reading image in greyscale.

b. **Display an image:** Function cv2.imshow() is used to display an image on the screen.

Syntax: cv2.imshow("img_variable_name")

c. Resizing image: Function cv2.resize() is used to resize an image.

Syntax: cv2.resize(s, size,fx,fy,interpolation)

s – input image (required).
size – desired size for the output image after resizing (required)
fx – Scale factor along the horizontal axis. (optional)
fy – Scale factor along the vertical axis.
Interpolation(optional)- This flag uses following methods:
INTER_NEAREST – a nearest-neighbor interpolation
INTER_LINEAR – a bilinear interpolation (used by default)
INTER_AREA – resampling using pixel area relation. It may be a preferred method for image decimation, as it gives moire'-free results. But when the image is zoomed, it is similar to the INTER_NEAREST method.

d. Rotating an image: Function cv2.rotate() is used to rotate the image (2D array) in multiples of 90 degrees.

Syntax: cv2.rotate(src, rotateCode[, dst])

src: It is the image to be rotated.
rotateCode: It is an enum to specify how to rotate the array. Here are some of the possible values:
cv2.cv2.ROTATE_90_CLOCKWISE
cv2.ROTATE_180
cv2.ROTATE_90_COUNTERCLOCKWISE

e. **Addition of image:** Function cv2.add() is used to add images. This function adds the image directly. Before adding the image, it is necessary to resize the image.

Syntax: cv2.add(img1, img2)

f. **Histogram function:** Function cv2.calcHist() is used to find the histogram of an image. For 3 channel image (RGB image), the hist function needs to be calculated for each channel.

Syntax: cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])

images: it is the source image of type uint8 or float32 represented as "[img]".
channels: it is the index of channel for which we calculate histogram. For grayscale image, its value is [0] and
color image, you can pass [0], [1] or [2] to calculate histogram of blue, green or red channel respectively.
mask: mask image. To find histogram of full image, it is given as "None".
histSize: this represents our BIN count. For full scale, we pass [256].
ranges: this is our RANGE. Normally, it is [0,256].

g. **Color conversions:** Function cv2.cvtColor() is used to convert an image from one color space to another. There are more than 150 color-space conversion methods available in OpenCV.
For Example: - cv2.COLOR_BGR2GRAY is used to convert Color image to greyscale.

Syntax: cv2.cvtColor(src, code[, dst[, dstCn]])

src: It is the image whose color space is to be changed.
code: It is the color space conversion code.
dst: It is the output image of the same size and depth as src image. It is an optional parameter.

dstCn(optional): It is the number of channels in the destination image. If the parameter is 0 then the number of the channels is derived automatically from src and code.

h. **Simple Thresholding**: Function cv2.threshold() is used for thresholding.
For every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to a certain value(usually zero) , otherwise, it is set to another value(usually maximum value).

Syntax:
cv2.threshold(source, thresholdValue, maxVal, thresholdingTechnique**)**

source: Input Image array (must be in Grayscale).
thresholdValue: Value of Threshold below and above which pixel values will change accordingly.
maxVal: Maximum value that can be assigned to a pixel.
thresholdingTechnique: The type of thresholding to be applied. There are various types of thresholding that are used.
For Example: - cv2.THRESH_BINARY: If the pixel intensity is greater than the threshold, the pixel value is set to 255(white), else it is set to 0 (black).

i. **Adding Noise to the image**: Function cv2.randn fills an array with normally-distributed random numbers with the specified mean and standard deviation1. It can be used to add noise to images or generate random images.

Syntax: cv2.randn(dst, mean, stddev)

dst: the output array
mean: the mean value of noise array.
stddev: the standard deviation of noise array.

1.2 Conclusion:

_____

_____

_____

_____

_____

_____

_____

_____

_____

1.3 References:

i. Gonzalez R, Woods R, "Digital image processing", Pearson Prentice Hall, 2008.

ii. Gonzalez R, Woods R, Steven E, "Digital Image Processing Using MATLAB®", McGraw Hill Education,2010.

iii. Jayaraman S, Esakkirajan S and Veerakumar T,"Digital Image Processing" Tata McGraw Hill, 2010

iv. Joshi, Madhuri A. "Digital Image Processing: an algorithm approach", PHI Learning Pvt. Ltd., 2006.

v. Pictures taken from: http://www.imageprocessingplace.com/root_files_V3/image_databases.html

_____

(Course Teacher)