# Image Processing Documentation

**Aim of The Task:**

To Learn the Basics of Image Processing using OpenCV in Python

**Things Learnt in the Task:**

1. Opening and Reading Images using OpenCV
2. Finding out Rows, Columns and Channels of a Image
3. Finding out the Contour and the area & perimeter of contours
4. Converting BGR Images to Grayscale and also to one color Images

**Software Used:**

1. Open CV
2. Python
3. Numpy Module

**Different Commands/Functions Used in The Task:**

1. **import cv as cv** – This command is used to import the OpenCV library(or module) into the project file so that we can use it to work with images
2. **import numpy** – This command is used to import the Numpy library to manipulate matrices. Since the images we use are stored as matrices, numpy makes it easier to work with images and manipulate them
3. **cv.imread(filename,[flags])**: cv.imread() method loads an image from the specified file in the form of a matrix. **Filename** is a string representing the path of the image to be read and the **flag** specifies the way in which the image should be read. It's default value is **cv.IMREAD_COLOR** which loads the image as color. To load image as grayscale cv.IMREAD_GRAYSCALE is used

4. **cv.waitKey(time):** Its argument is the time in milliseconds. The function waits for specified milliseconds for any keyboard event. If any key is pressed, the program continues. If 0 is passed, it waits indefinitely for a keystroke. It can also be set to detect specific key strokes.

5. **cv.destroyAllWindows():** This command asks python to close all the open windows.

6. **cv.VideoCapture(camera)**– This command tells python to set up an instance of a VideoCapture class and assign it to the variable capture. In other words, we need to tell python where we are getting our images from, in this case the number assigned to the camera we need to use.

7. **<capture>.imread** - This is the command we use to retrieve images from the source we have named as capture.

8. **<capture>.release()** – This command releases the camera that we had initialized when we used the cv.VideoCapture command.

9. **cv.cvtColor(image, cv.COLOR_BGR2GRAY)**: As first input, this function receives the original image. As the second input, it receives the color space conversion code. Since we want to convert our original image from the BGR color space to gray, we use the code COLOR_BGR2GRAY.

10. **cv.findContours()** - This function is used to finds contours in a binary image.The function retrieves contours from the binary image. The contours are a useful tool for shape analysis and object detection and recognition. The different parameters are:

    a. Image: source, an 8-bit single-channel image. Non-zero pixels are treated as 1's. Zero pixels remain 0's, so the image is treated as binary.
    b. Mode: detected contours
    c. Method: containing information about the image topology. It has as many elements as the number of contours.

11. **cv.drawContours()**: This is Used to draw Contours of an Image. The Parameters are:
    a. Image: Destination image
    b. Contours:All the input contours.
    c. Contour ID: Parameter indicating a contour to draw. If it is negative, all the contours are drawn
    d. Color:Color of the contours
    e. Thickness of Line: Thickness of lines the contours are drawn with. If it is negative, the contour interiors are drawn.

12. **cv.arcLength():** It is also called arc length.. Second argument specify whether shape is a closed contour (if passed True), or just a curve.

13. **cv.contourArea():** Contour area is given by the function.

**Bibliography:**


1. https://docs.opencv.org/master/d9/df8/tutorial_root.html

2. OpenCV Python Tutorial For Beginners (Channel: Programming Knowledge)

3. www.stackoverflow.com


**Project By:**


Pratik Kumar Panda