FUNCTION DESCRIPTION

1. Cv2.cvtColor:
   It converts an image from one color space to another . A BGR image can be converted to HSV, LAB or RGB but an HSV image cannot be directly converted to LAB or RGB. It has to be first converted into a BGR image as BGR is the channel in which openCV works.

2. Cv2.GaussianBlur:
   It applies Gaussian blur to an image. It uses a kernel to reduce the noise and smooths the image. Greater the kernel size , more the noise will be reduced and the image will be smoothen.

3. Cv2.calcHist:
   It calculates the histogram, of an image. We can compute the histogram of color as well as gray scale images. Plots the number of pixels with the same intensity.

4. Cv2.circle:
   It draws a circle on the image. Also use to mask as image. Takes src, centre of circle, radius, colour, and thickness of the image as parameters. If thickness is set to -1 then the circle will be filled.

5. Cv2.putText:
   Put text on the image . Takes src img, text that has to be printed, starting point of text, font style , font size , color, and thickness as parameters. It writes the text with specific font, size and color.

6. Image cropping:
   It crops a section of image. Here we make use of numpy slicing.

7. Cv2.rectangle:
   It draws a rectangle on the figure. It takes src img, 2 points that are diagonally opposite, color, thickness as parameters. If thickness is set to -2 then the rectangle will be filled.

8. Cv2.flip:
   it fips the image. It takes src image and a code which represents the way we want to flip it.
   1 – horizontal flip
   0 – vertical flip
   -1 – flip both horizontal as well as vertical

9. Cv2. Threshold:
   There are 2 types of thresholding:-
   1. Simple
   2. Adaptive

In this we make use of simple threshold which takes src image, a threshold value and the another value and the threshold method. So basically what happens is that If the intensity of pixel is greater than the threshold pixel then it is set to the set value otherwise it is set to zero.

10. Rotation Function:
    It rotates the image by the angle about a rotation point. For this what we do is we make a rotation matrix by using the getRotationMatrix2D and use the .warpAffine function which takes the src image , the rotation matrix and the dimensions of the image as input and gives the rotated image.

11. Cv2.Sobel:
    It applies the Sobel operator on an image to find edges. We can get the edges in the x direction by using sobelx and in y direction by using sobely.

12. Cv2.bitwise_or:
    It performs the or operation on two images. Using the concept of sets, we get the union of two sets, similarly we get the union of two images

.