Open CV Assignment 4

**1. Explain how OpenCV is useful in real-world projects.**

A. There are lots of applications in real world which are solved using OpenCV, some of them are listed below :

* face recognition
* Automated inspection and surveillance
* number of people – count (foot traffic in a mall, etc)
* Vehicle counting on highways along with their speeds
* Interactive art installations
* Anamoly (defect) detection in the manufacturing process (the odd defective products)
* Street view image stitching
* Video/image search and retrieval
* Robot and driver-less car navigation and control
* object recognition
* Medical image analysis
* Movies – 3D structure from motion
* TV Channels advertisement recognition

**2. Explain how to use OpenCV's preprocessing methods.**

A.

1. **Reading of an image:** We can read an image using imread() method as below:

img = cv2.imread('Zebra.jpg')

1. **Writing of an image:** To display an image we need to use the cv2.imshow() function. This function helps to display the image in a separate window. We can write as below:

cv2.imshow('zebra',img)

1. **Display of an image:** We used cv2.imwrite() to save the image in the local system, we can write as below:

cv2.imwrite('zebra.png',img)

1. **Conversion of a colored image to gray-scale:** We can change the color of the image using cv2.cvtColor() as below:

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

1. **Histogram of an image:** The histogram shows the frequency of the pixels in the image. It is a plot between pixel values (x-axis) and the number of pixels (y-axis). A histogram shows how color density varies from black pixel (0) to white pixel (255). We can write as:

img = cv.imread('zebra.jpg',0)

hist = cv.calcHist([img],[0],None,[256],[0,256])

plt.hist(img.ravel(),256,[0,256])

1. **Edge Detection using Canny Algorithm:**

Canny edge detection is based on:

* gradient of the image i.e. difference between two adjacent pixels
* hysteresis filtering: It selects the lines using those pixels which are different from adjacent ones.

For the Canny edge detection algorithm, we need to provide 3 arguments in the cv2.Canny() function i.e. input image array, the minimum value of a pixel, the maximum value of the pixel.

We can write as:

edges = cv2.Canny(img,100,200)

plt.imshow(edges,cmap = 'gray')

plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

**3. Explain the many types of image filters available in OpenCV.**

A. Filters generally use many pixel for computing each new pixel value but point operations can use one pixel to perform an image processing. The filters can use for blurred or fuzzy the local intensity of image to make it smooth.

**Type of filters:** 2 types: 1. Linear Filtering 2. Non-linear Filtering

**1. Linear filter:** The convolution of matrix pixels and kernel matrix to reduce intensity of image, which is blurring the image.

There are 3 types of linear filter:

* Box filter
* Gauss filter
* Laplace filter or Mexican hat filter

**2. Non Linear Filter:** Using some non-linear function from the source pixel value. The idea is to replace the target pixel value with its neighbor pixels value from some ordering mechanism or function.

There are many types of Non Linear Filter but in this article I will show you just 3 of them:

* Minimum Filter
* Maximum Filter
* Median Filter

**4. Explain how to use OpenCV to implement feature extraction methods.**

A. There are various features that can potentially be extracted using different machine learning algorithms. Lowe et al. (2004) developed Scale Invariant Feature Transform (SIFT) aiming to solve intensity, viewpoint changes and image rotation in feature matching. SIFT allows estimation of scale-space extrema followed by keypoint localisation, orientation and subsequently computation of local image descriptor for each key point. Moreover, SIFT offers efficient recognition of objects in a given image, however, its implementation is computationally expensive and time-consuming.

SIFT has inspired the development of other variants in particular to overcome the complexity and the associated computational demand. One such variant is Speed up Robust Feature (SURF), which reportedly outperforms SIFT without a negative impact on the robustness of detected points and overall model accuracy. Another alternative to SIFT is Robust Independent Elementary Features (BRIEF) offering similar performance with significantly less complexity. Furthermore, Rublee et al., (2011) reported that Oriented FAST and Rotated BRIEF (ORB) provides more efficient performance.

Additionally, Histogram of Oriented Gradients (HOG) HOG is utilised in various tasks including object detection allowing the occurrence of edge orientations to be measured. In principle, this descriptor is representative of a local statistic of the orientations for the image gradients for a key point. Simply, each descriptor can be considered as a collection of histograms that are composed of pixel orientations assigned by their gradients.