Open CV Assignment 2

**1. Explain how to read and write an image using OpenCV.**

A. Reading: For reading an image, we use the imread() function in OpenCV.

The syntax: imread(filename, flags)

It takes two arguments:

1. The first argument is the image name, which requires a fully qualified pathname to the file.
2. The second argument is an optional flag that lets you specify how the image should be represented. OpenCV offers several options for this flag, but those that are most common include:

* cv2.IMREAD\_UNCHANGED  or -1
* cv2.IMREAD\_GRAYSCALE  or 0
* cv2.IMREAD\_COLOR  or 1

Writing: To write/save an image into the file directory, using the imwrite() function.

The syntax: imwrite(filename, image).

It takes two arguments:

1. The first argument is the filename, which must include the filename extension (for example .png, .jpg etc). OpenCV uses this filename extension to specify the format of the file.
2. The second argument is the image you want to save. The function returns True if the image is saved successfully.

**2. Explain how you can build an Object detection project with using OpenCV**

A. Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones in their paper [Rapid Object Detection using a Boosted Cascade of Simple Features](https://www.researchgate.net/publication/3940582_Rapid_Object_Detection_using_a_Boosted_Cascade_of_Simple_Features). Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier. 

* **Positive images –** These images contain the images which we want our classifier to identify.
* **Negative Images –** Images of everything else, which do not contain the object we want to detect.

**3. Explain BoxFilter.**

A. The Box Filter operation is similar to the averaging blur operation; it applies a bilateral image to a filter. Here, we can choose whether the box should be normalized or not.

We can do this using **boxFilter()** method of the **imgproc** class.

The syntax is:

boxFilter(src, dst, ddepth, ksize, anchor, normalize, borderType)

This method accepts the following parameters −

* **src** − A **Mat** object representing the source (input image) for this operation.
* **dst** − A **Mat** object representing the destination (output image) for this operation.
* **ddepth** − A variable of the type integer representing the depth of the output image.
* **ksize** − A **Size** object representing the size of the blurring kernel.
* **anchor** − A variable of the type integer representing the anchor point.
* **Normalize** − A variable of the type boolean specifying weather the kernel should be normalized.
* **borderType** − An integer object representing the type of the border used.

**4. Explain Thresholding mechanism in OpenCV**

A. Thresholding is one of the most common (and basic) segmentation techniques in computer vision and it allows us to separate the *foreground* (i.e., the objects that we are interested in) from the *background* of the image.

Thresholding comes in three forms:

1. We have **simple thresholding** where we manually supply parameters to segment the image — this works extremely well in controlled lighting conditions where we can ensure high contrast between the foreground and background of the image.
2. We also have methods such as **Otsu’s thresholding** that attempt to be more dynamic and *automatically* compute the optimal threshold value based on the input image.
3. And finally we have **adaptive thresholding** which, instead of trying to threshold an image *globally* using a single value, instead breaks the image down into smaller pieces, and thresholds each of these pieces *separately* and *individually*.

**5. What is the drawback of OpenCV ?**

A. Some of the drawbacks of OpenCV are:

1. Takes more time to display image compare to matplotlib
2. Needs improvement for more effective face recognition systems