Report on Computer Vision

Computer Vision is a field of study which give computers the ability to understand images and come up with useful conclusions after analyzing the images. Some examples of application of computer vision are –

- Defect identification and classification in manufacturing
- CV can also be used to improve security at airports by analyzing the X-Rays of luggage and identifying whether it contains any weapons / dangerous equipment
- In medicine, CV can be used to identify cancerous tumors
- A popular application of CV is self-driving cars
- IBM has helped its client ADNOC to analyze rocks using CV , as analyzing rocks is a crucial step in extruding oil, and usage of CV has made this task much more efficient

Next, I learnt that an image is represented as a grid of pixels in a computer. Usually, 8 bits are assigned to each pixel and the pixel can take intensity values from 0 to 255. Also, a color image has three channels, red, blue and green. A grayscale image has only one channel. A video is a sequence of images. I also learnt about two popular libraries for image processing, **PIL** and **OpenCV**. I learnt about reading and writing images in PIL as well as OpenCV. Quantization of an image refers to the number of intensities a pixel can take. An important difference between PIL and OpenCV is that the order of channels when an image is read in PIL is RGB while in case of OpenCV it is BGR. Some operations that can be performed on images are – copying, flipping, cropping, changing the intensity values at some pixels, drawing some shapes on images, adding text on images, converting a color image to a grayscale image etc. Cropping an image can be easily done by slicing the image matrix. In case of copying an image, we cannot directly assign the image to another variable as those two variables will then have the same memory locations. Instead, using the .copy() method would prevent this. In pixel transformations, I learnt about getting no. of occurrences of each pixel intensity value, intensity transformation and image thresholding and segmentation. We can apply pixel transformations independently to each pixel. Consider an example of transformation –

$$g(i,j) = \alpha f(i,j) + \beta$$

Here, f represents original intensity value while g represents intensity value after transformation at pixel (i, j). Here the parameter α controls the contrast of the image, while the parameter β controls the brightness of the image. I also learnt about image thresholding.

Image negatives can be obtained by setting $\alpha = -1$ and $\beta = N - 1$, where N is the quantization of the image

There are various algorithms which help us determine the correct value of contrast. One such algorithm is the histogram equalization algorithm.

We can also perform geometric operations on images, those are – scaling, translation and rotation.

Scaling and translation can in general be represented by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

If we assume the axis of rotation at the center of the image, then we have direct functions in PIL and OpenCV which take theta as input and return the rotated image.

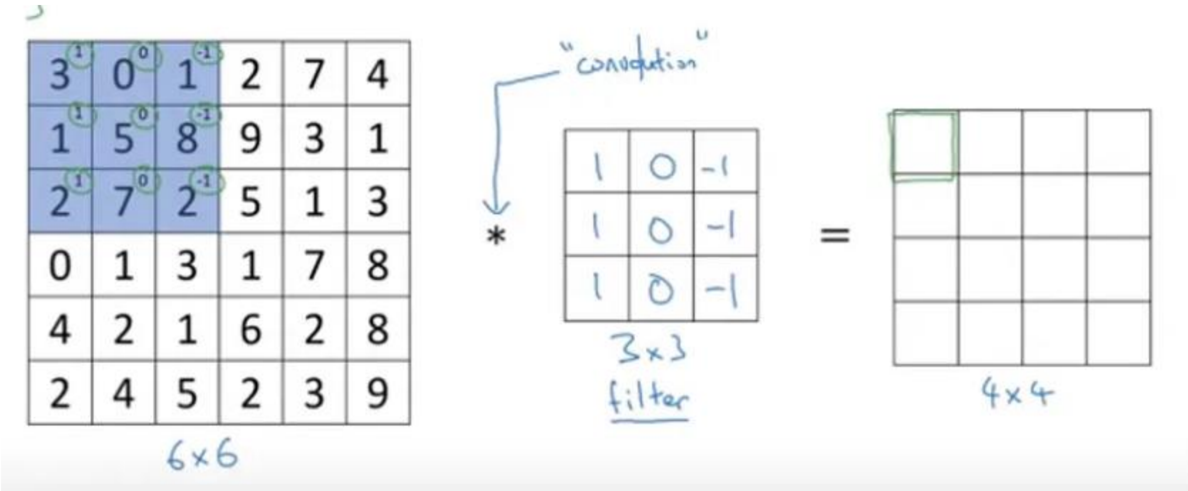
An important point related to scaling-

When we perform for e.g. horizontal scaling, there are some pixel columns in the scaled image which do not have a mapping with any column of the original image, we use KNN algorithm to determine the pixel intensity values in these columns.

Spatial Operations – Spatial operations include convolution/linear filtering, edge detection and median filters.

Convolution is defined as – $Z = W * X$

Z is the matrix obtained after convolution, X is the image matrix, W is called kernel of filter



(Source - https://vincmazet.github.io/bip/filtering/convolution.html)

Edge detection involves detecting edges (places where the intensity values change abruptly.). We do this approximating the partial derivatives of pixel intensities along x and y direction. If $I(i, j)$ represents intensity at a pixel located in ith row and jth column, then partial derivative of I w.r.t. j is $I(i, j + 1) - I(i, j)$. By applying such approximations along Y direction as well and combining the partial derivatives obtained in X and Y directions, we get the gradients. These approximations of derivatives can be obtained by convolving the image matrix with a kernel. There kernels are called Sobel operators.

Median filtering is a method in which in which we first decide the size of neighborhood around a particular pixel, and then we compute the median of those values and then assign those values to all the selected pixels. In case of boundary elements, we use zero padding.

Next, I learnt briefly about the KNN algorithm. Here are some details – KNN is used for classification. It is a supervised learning algorithm. Firstly, the data is divided into training and testing datasets. For a given example, it computes the 'K' nearest examples to it based on distance, and then the example belongs to the class to which the majority of the neighbor examples belong. That value of K is chosen which gives maximum accuracy. I also learnt about Linear classifiers, logistic function and gradient descent. In case of linear classifiers, we compute

$$z = \vec{w} \cdot \vec{x} + b$$

Here **w** represents weight vector and **x** represent the feature vector. b represents the bias.

Choosing the appropriate value of **w** and **b** gives us the decision plane. For binary classification, we use thresholding, i.e. if z is greater than 0, then y_hat = 1, else y_hat = 0.

However, the sigmoid function f provides better results than thresholding as the output of the sigmoid function can be interpreted as probability whereas the threshold function only produces a binary output.

$$f(z) = \frac{1}{1 + e^{-z}}$$

For ensuring whether the model is working as expected or not, we define the loss and the cost functions. Examples of loss functions are classification error and Crossentropy loss function. The latter is preferred because it is differentiable and we can apply the gradient descent algorithm to optimize the cost function. Cost function is the average of loss function over all the training examples. Gradient descent is an optimization technique used to optimize the cost function

$$\vec{w}_i = \vec{w}_{i-1} - \eta \nabla C(\vec{w}_{i-1})$$

In case of multi-class classification, we first define multiple decision planes and then use the soft-max function to convert the output zi into probability. Then the example belongs to the class which has the maximum probability.