1)what exactly is features ?

Ans : In computer vision and image processing, a feature is a piece of information about the content of an image; typically about whether a certain region of the image has certain properties. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image. Other examples of features are related to motion in image sequences, or to shapes defined in terms of curves or boundaries between different image regions.

Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is repeatability: whether or not the same feature will be detected in two or more different images of the same scene.Feature detection is a low-level image processing operation. That is, it is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features. As a built-in pre-requisite to feature detection, the input image is usually smoothed by a Gaussian kernel in a scale-space representation and one or several feature images are computed, often expressed in terms of local image derivative operations.

There are many computer vision algorithms that use feature detection as the initial step, so as a result, a very large number of feature detectors have been developed. These vary widely in the kinds of feature detected, the computational complexity and the repeatability.

2)for a top edges detector write out the convolutional kernel matrix .

Ans : Edges represents the object boundaries. So edge detection is a very important preprocessing step for any object detection or recognition process. Simple edge detection kernels are based on approximation of gradient images.

Prewitt operator

For Ix(x,y)

⎡⎣⎢−1−1−1 000 111⎤⎦⎥

For Iy(x,y)

⎡⎣⎢−101 −101 −101⎤⎦⎥

Sobel operator

For Ix(x,y)

⎡⎣⎢−1−2−1 000 121⎤⎦⎥

For Iy(x,y)

⎡⎣⎢−101 −202 −101⎤⎦⎥

Laplacian

It is an approximation of second order derivative that defines zeros crossing. For Example 3x3 laplacian is :

⎡⎣⎢010 1−41 010⎤⎦⎥

Laplacian usually is applied after gaussian smoothing. So LOG refers to laplacian of gaussian

3)describe the mathematical operation that 3x3 kernel performs on a single pixel in an image .

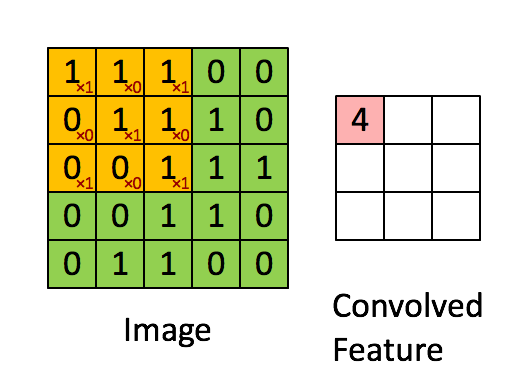
Ans : In image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image.

In this article, here are some conventions that we are following —

We are specifically referring to 2D convolutions that are usually applied on 2 matrix objects such as images. These concepts also apply for 1D and 3D convolutions, but may not correlate directly.

While applying 2D convolutions like 3X3 convolutions on images, a 3X3 convolution filter, in general will always have a third dimension in size. This filter depends on (and is equal to) the number of channels of the input image. So, we apply a 3X3X1 convolution filter on gray-scale images (the number of channels = 1) whereas, we apply a 3X3X3 convolution filter on a colored image (the number of channels = 3).

We will refer to all the convolutions by their first two dimensions, irrespective of the channels. (We are observing the assumption of zero padding).



A convolution filter passes over all the pixels of the image in such a manner that, at a given time, we take ‘dot product’ of the convolution filter and the image pixels to get one final value output. We do this hoping that the weights (or values) in the convolution filter, when multiplied with corresponding image pixels, gives us a value that best represents those image pixels. We can think of each convolution filter as extracting some kind of feature from the image.

Therefore, convolutions are done usually keeping these two things in mind -

Most of the features in an image are usually local. Therefore, it makes sense to take few local pixels at once and apply convolutions.

Most of the features may be found in more than one place in an image. This means that it makes sense to use a single kernel all over the image, hoping to extract that feature in different parts of the image.

Now that we have convolution filter sizes as one of the hyper-parameters to choose from. The choice is can be made between smaller or larger filter size.

4)what is the significance of a convolutional kernel added to 3x3 matrix of zeros.?

Ans : image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image

5)what exactly is padding ?

Ans : Padding is a term relevant to convolutional neural networks as it refers to the amount of pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero. If, however, the zero padding is set to one, there will be a one pixel border added to the image with a pixel value of zero.

6)what is the concept of stride ?

Ans : Stride is a component of convolutional neural networks, or neural networks tuned for the compression of images and video data. Stride is a parameter of the neural network's filter that modifies the amount of movement over the image or video. For example, if a neural network's stride is set to 1, the filter will move one pixel, or unit, at a time. The size of the filter affects the encoded output volume, so stride is often set to a whole integer, rather than a fraction or decimal.

Imagine a convolutional neural network is taking an image and analyzing the content. If the filter size is 3x3 pixels, the contained nine pixels will be converted down to 1 pixel in the output layer. Naturally, as the stride, or movement, is increased, the resulting output will be smaller. Stride is a parameter that works in conjunction with padding, the feature that adds blank, or empty pixels to the frame of the image to allow for a minimized reduction of size in the output layer. Roughly, it is a way of increasing the size of an image, to counteract the fact that stride reduces the size. Padding and stride are the foundational parameters of any convolutional neural network.

7)what are the shapes of pytorch's 2D convolution inputs and weight parameters ?

Ans : The PyTorch nn conv2d is defined as a Two-dimensional convolution that is applied over an input that is specified by the user and the particular shape of the input is given in the form of channels, length, and width, and output is in the form of convoluted manner.

input: Input is defined as an input tensor of shape(minibatch, in\_channels).

weight: Weight is defined as a filter of shape(out\_channels).this are the shapes of input & weight parameters.

8)what exactly is channel?

Ans : channel is a there can be any number of conventional primary colors making up an image; a channel in this case is extended to be the grayscale image based on any such conventional primary color. By extension, a channel is any grayscale image of the same dimension as and associated with the original image.Channel is a conventional term used to refer to a certain component of an image. In reality, any image format can use any algorithm internally to store images. GIF images actually refer to the color in each pixel by an index number, which refers to a table where three color components are stored. However of how a specific format stores the images, discrete color channels can always be determined, as long as a final color image can be rendered.An RGB image has three channels: red, green, and blue. RGB channels roughly follow the color receptors in the human eye, and are used in computer displays and image scanners.

9)explain the relationship between matrix multiplication and a convolution?

Ans : convolution, in time domain, you had to flip and then slide(slide means that each and every element of variable needs to be slided) which is computationally very expensive. Instead, in frequency domain, the same result as that of convolution is achieved just by multiplication and yes when you signal is in freq. domain, you simply multiply the signal(X)-which is matrix with Signal(Y), which is also a matrix.

So, now you will be able to understand that, Yes convolution is same as matrix multiplication(where matrix X and Y matrix of signal)