1.What exactly is a feature?

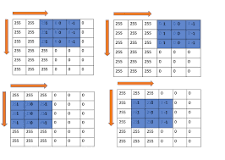
Key characteristics of the CSV file are: summary information for entire table provided at beginning of file. multiple header lines. comma delimited fields.  
...  
**Annotation or references may be applied to:**

* a group of tables.
* an entire table.
* a row.
* a coloumn.
* an individual cell.

2.For a top edge detector, write out the convolutional kernel matrix.

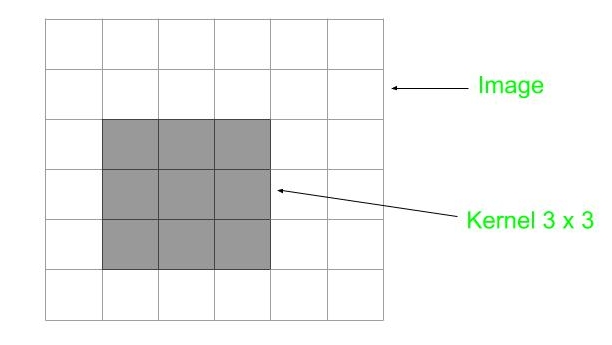
You calculate the convolution of each channel in the kernel with each corresponding channel in the image. Essentially, you need to **perform the 2D convolution operation three times over, and then you sum up the results to get the final kernel output.**

A convolution is a type of matrix operation, consisting of a kernel, a small matrix of weights, that slides over input data performing element-wise multiplication with the part of the input it is on, then summing the results into an output.



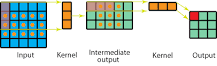
3. Describe the mathematical operation that a 3x3 kernel performs on a single pixel in an image.

Image is denoted as matrix inside computer. An image contains a lot of features like edge, contrast etc. In image processing features have to be extracted from the image for further study of image.  
Convolution is a fundamental operation on images in which a mathematical operation is applied to each pixel to get the desired result.  
For this purpose, another matrix called as **kernel** is used which is smaller in size of image. This is also called filter. This filter is applied on each pixel of the image and new value obtained is the value of that pixel. The image obtained is called **filtered image**.  
In kernel each cell contain some value, that kernel is kept above the pixel and corresponding values are multiplied and then summed up this value obtained is new the value of pixel.



4. What is the significance of a convolutional kernel added to a 3x3 matrix of zeroes?

Convolution is using a 'kernel' **to extract certain 'features' from an input image**. Let me explain. A kernel is a matrix, which is slid across the image and multiplied with the input such that the output is enhanced in a certain desirable manner.



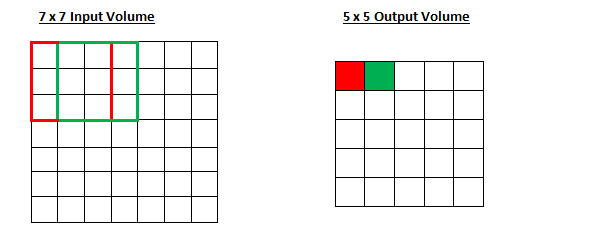
5. What exactly is padding?

Padding, also known as white space, is **the empty space between and around individual elements of a page layout**; these elements could be pieces of copy, images, cards, buttons, icons, etc. When used correctly, white space brings visual clarity and balance to a layout.

6. What is the concept of stride?

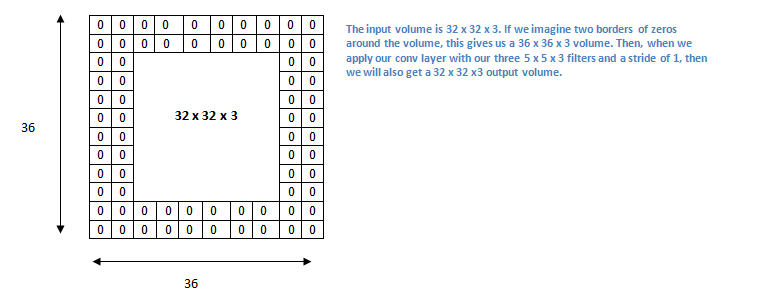
Stride is a component of [convolutional neural networks](https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network), or [neural networks](https://deepai.org/machine-learning-glossary-and-terms/neural-network) tuned for the compression of images and video data. Stride is a parameter 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.

**How does Stride work?**



[Source](https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/)

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](https://deepai.org/machine-learning-glossary-and-terms/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&#39;s 2D convolution&#39;s input and weight parameters?

8. What exactly is a channel?

[Color](https://en.wikipedia.org/wiki/Color) digital images are made of [pixels](https://en.wikipedia.org/wiki/Pixel), and pixels are made of combinations of [primary colors](https://en.wikipedia.org/wiki/Primary_color) represented by a series of code. A **channel** in this context is the [grayscale](https://en.wikipedia.org/wiki/Grayscale) image of the same size as a color image,made of just one of these primary colors. For instance, an image from a standard [digital camera](https://en.wikipedia.org/wiki/Digital_camera) will have a red, green and blue channel. A grayscale image has just one channel.

In [geographic information systems](https://en.wikipedia.org/wiki/Geographic_information_system), channels are often referred to as **raster bands**. Another closely related concept is **feature maps**, which are used in [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_network" \o "Convolutional neural network).

9.Explain relationship between matrix multiplication and a convolution?

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) but ONLY IN FREQUENCY DOMAIN**.

convolution is an operation (integration or summation, for continuous and discrete time, respectively) that relates the output of a linear and time-invariant (LTI) system to its input and its impulse response. I will consider discrete time only (since this is posted in DSP) and before defining some terms, you can think of a system as a black-box that takes inputs (functions) and provides outputs (other functions). We would like to have a way to compute the output of an LTI system for any input we can think.

Linearity and Time-invariance are two very useful and crucial properties for explaining how convolution comes up in systems. Linearity can be roughly thought as if you apply two different inputs to the system separately, ax1[n]ax1[n], bx2[n]bx2[n], and you get two outputs, ay1[n]ay1[n], by2[n]by2[n], then if you add the inputs together, ax1[n]+bx2[n]ax1[n]+bx2[n], and feed this sum as an input to the system, then the output will be simply the sum of the previous two separate outputs: ay1[n]+by2[n]ay1[n]+by2[n]. Time invariance can be briefly described as follows: if you apply an input x[n]x[n] to the system, you get an output y[n]y[n]. If you delay your input by, let’s say 3 samples, that is x[n−3]x[n−3], then the output is also a delayed version of the previous output, y[n−3]y[n−3]. A system that does \textit{not} behave like this is called a time \textit{variant} system.

Now, how can you tell what the system is doing on its inputs? You can tell by just knowing its so-called \textit{impulse response}, that is, how the system responds if you apply as input a very simple input signal: a signal that exists only in a single time instant, and then disappears. That is the so-called discrete-time delta function δ[n]δ[n], that is equal to 11 if n=0n=0, and is zero everywhere else. The output of such an input is called impulse response of the system and it is noted as h[n]. It’s easy to remember and it also makes sense: the response of the system to an impulse! :)

Why this impulse response is so important? Because all discrete time signals x[n]x[n] are linear combinations of scaled and shifted delta functions, that is

x[n]=∑k=−∞+∞x[k]δ[n−k]x[n]=∑k=−∞+∞x[k]δ[n−k]

We would like to know how an LTI system responds to that input - which is ANY input, in fact! By utilizing linearity and time invariance, we have the following input-output sequence:

δ[n]δ[n−k]x[k]δ[n−k]∑k=−∞+∞x[k]δ[n−k]x[n]⟶h[n]⟶h[n−k](by time invariance property)⟶x[k]h[n−k](by linearity)⟶∑k=−∞+∞x[k]h[n−k]⟶y[n]δ[n]⟶h[n]δ[n−k]⟶h[n−k](by time invariance property)x[k]δ[n−k]⟶x[k]h[n−k](by linearity)∑k=−∞+∞x[k]δ[n−k]⟶∑k=−∞+∞x[k]h[n−k]x[n]⟶y[n]

So we have just found out the output on an LTI system to ANY input! The output y[n]y[n] is defined as the \textbf{convolution} (noted by a star sign) of the input x[n]x[n] and the impulse response h[n]h[n], and once again

y[n]=x[n]∗h[n]=∑k=−∞+∞x[k]h[n−k]y[n]=x[n]∗h[n]=∑k=−∞+∞x[k]h[n−k]

It looks like a scary sum but it is quite simple, actually. All you have to do is time-reverse and shift the impulse response, and then multiply it with the input. Whenever their product is nonzero, you can compute the corresponding summation. Here’s an example of the operation of convolution.

