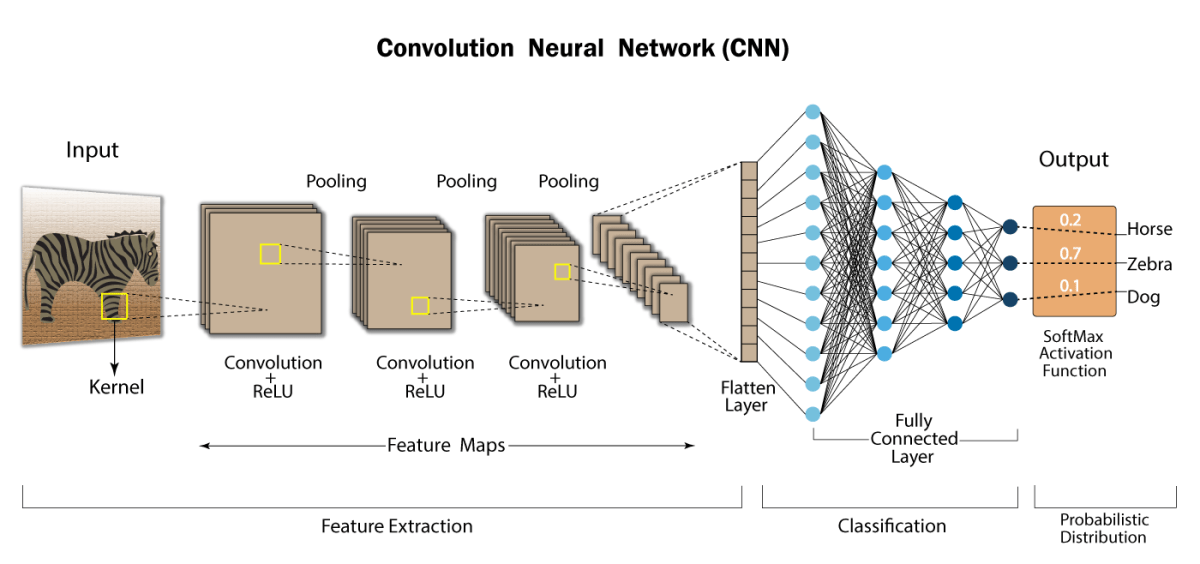
**What is Convolutional Neural Network?**

In today's world, we often hear about stuff like AI, ML, DL, etc. and, if we try to know about the technical products around us, you will find that they are very heavily dependent on concepts of AI, ML, DL, etc. Due to the advancements in new algorithms and technology, there is a huge industrial demand for these domains. One of several such fields is the domain of Computer Vision.

Computer Vision aims to enable machines to view the world as humans do. The algorithm recognizes the images/videos and uses the experience for several tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, etc. The progress in Computer Vision along with Deep Learning has been composed and improved with time on one particular algorithm commonly known as Convolutional Neural Network.



**Introduction**

A convolutional neural network (CNN) is a kind of Artificial Neural Network (ANN) used for image identification and processing. Whose primary purpose is to process pixel data.

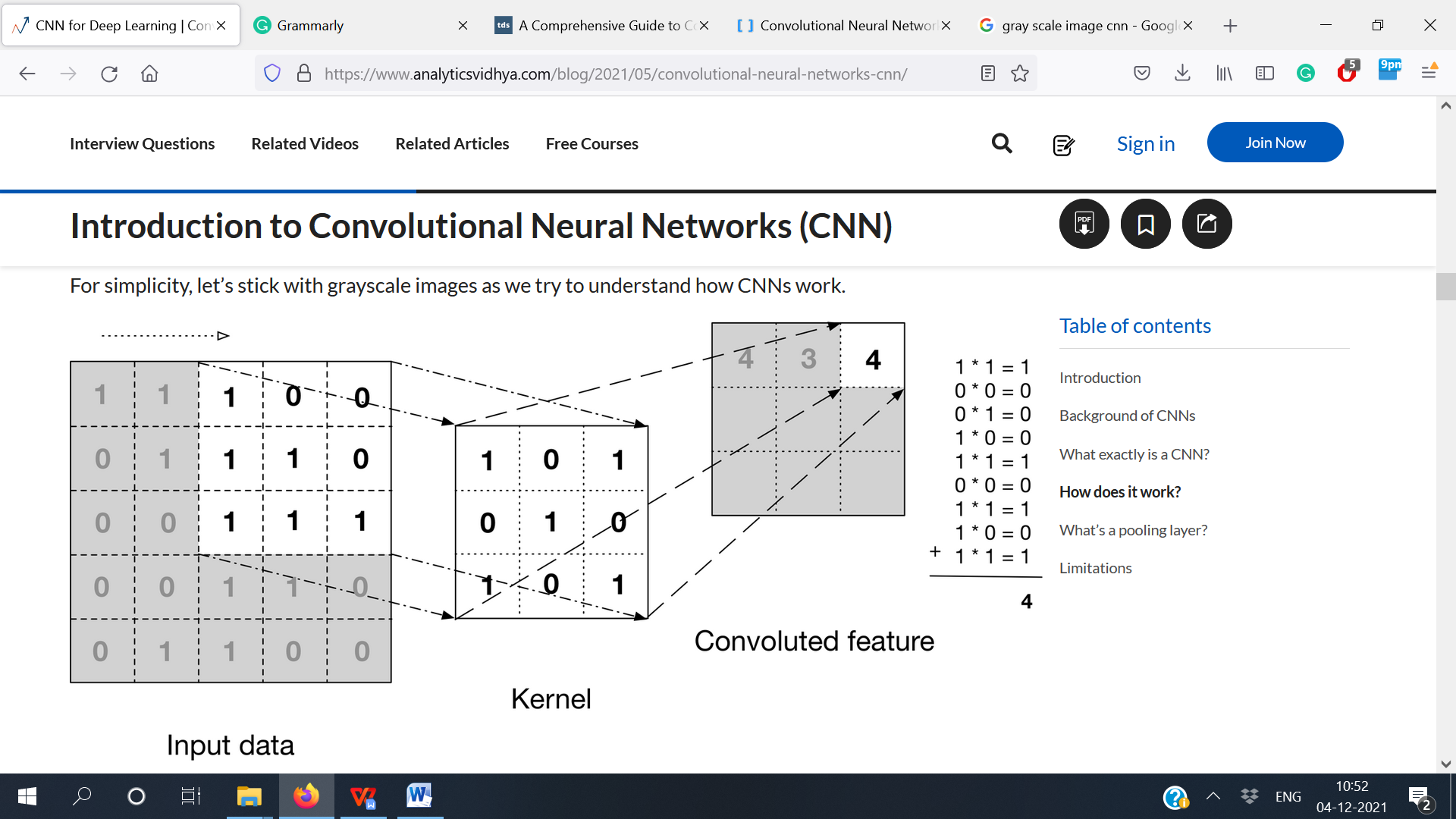
Convolutional neural networks are widely used in computer vision and have become the state-of-the-art for numerous visual applications such as image classification and have also gained progress in natural language processing for text classification. The design of a Conv-Net is similar to that of the connectivity pattern of Human Brain Neurons inspired by the structure of the Visual Cortex.

CNN's are very good at learning patterns in the input image, such as lines, gradients, circles, edges, etc. These properties make CNN so influential for computer vision. Unlike earlier computer vision algorithms, convolutional neural networks can operate directly on raw images. CNN's include numerous convolutional layers stacked one after the other, each capable of understanding complicated configurations.

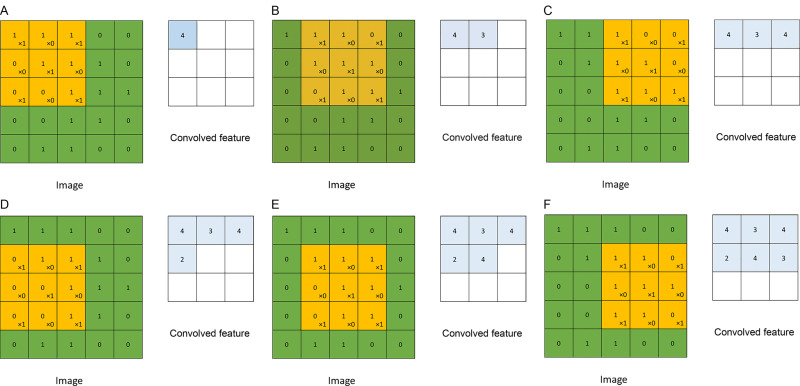
**How does it work?**

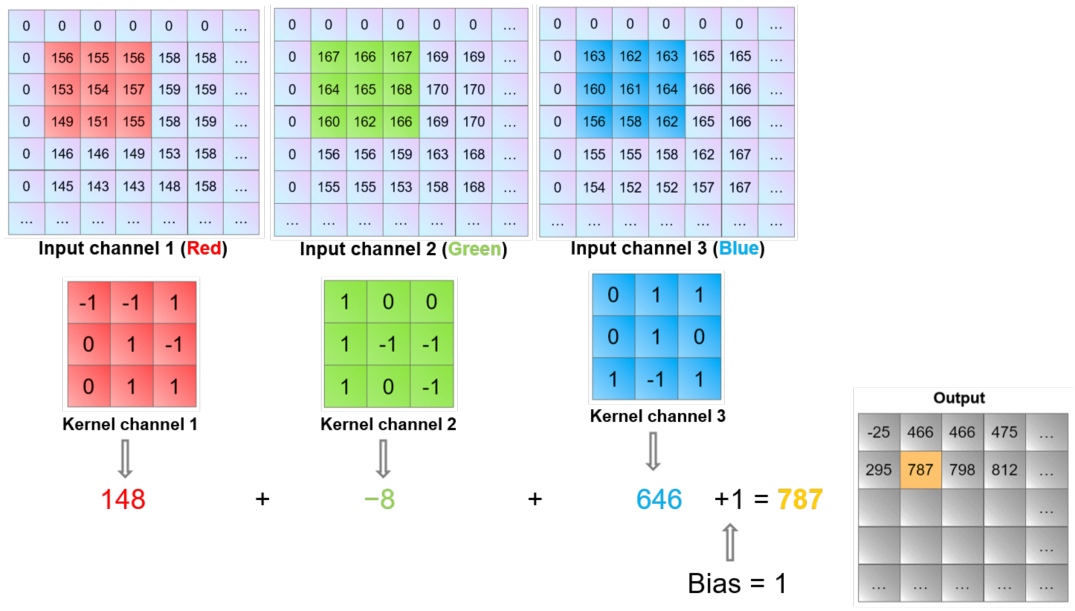
Before we go to the functioning of CNN, let’s cover the fundamentals of an image and how the images are represented. A colored image consists of 3 layers Red, Green, and Blue, represented by a matrix of pixel values. A grayscale image is a black and white image of 1 layer, represented by a matrix of pixel values. Take a look at this image to understand more.

Let’s first learn with grayscale images to understand how CNNs work.



The above image shows how a convolution is done. We take a 3x3 filter/kernel and move throughout the input image to obtain the convolved feature. After every convolution operation, the convolved feature is passed on to the next layer.





The above image represents how the convolution operation is performed on an RGB image. Unlike the grayscale image here, we take a 3X3 filter/kernel and move throughout each of the 3-layers.

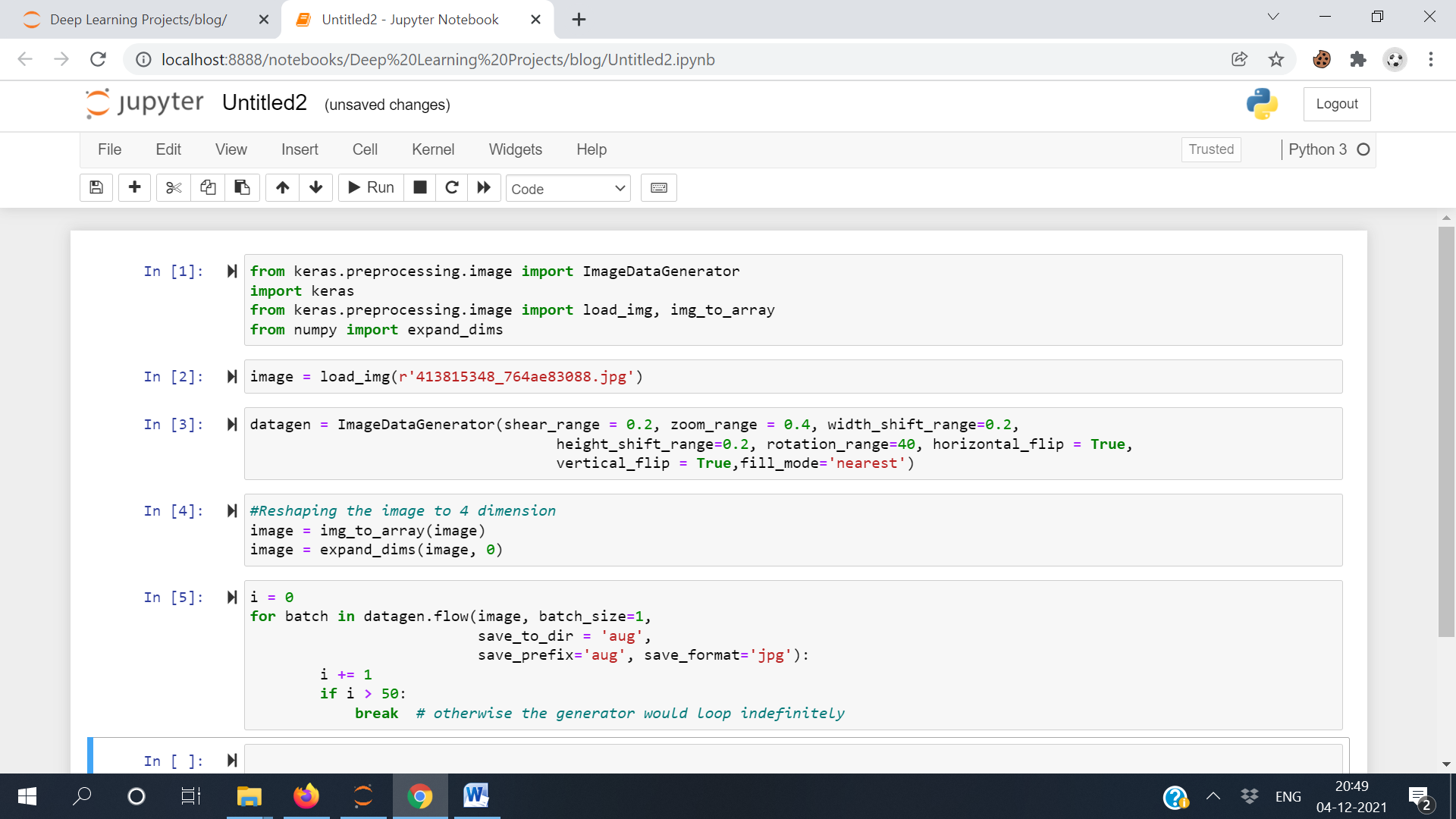
CNN's consist of various layers of artificial neurons. Artificial neurons are mathematical functions that compute the weighted sum of multiple inputs and outputs an activation value.

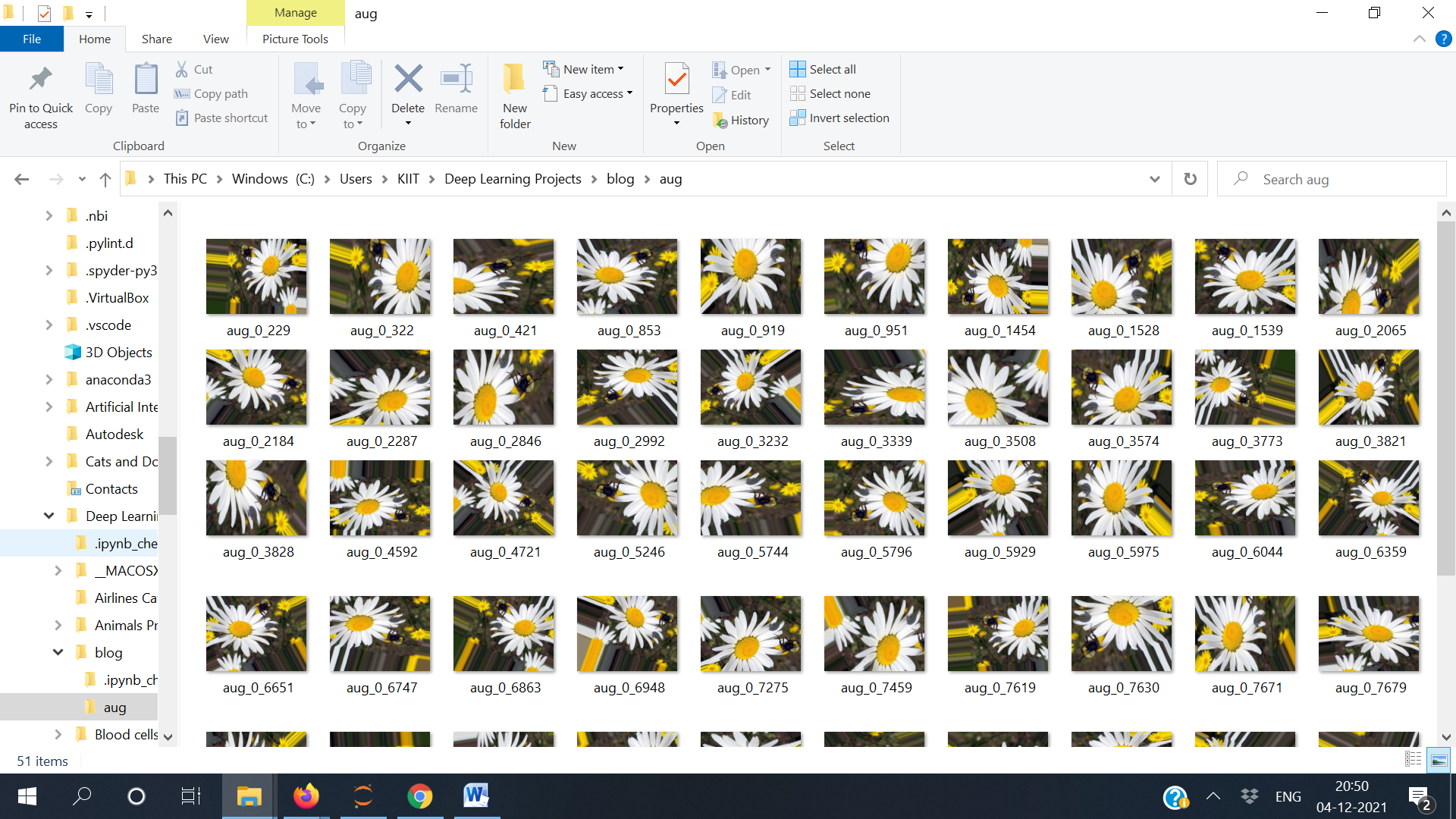
The first layer extracts primary features such as horizontal or vertical edges. This result is then passed on to the next layer that detects more complex features such as edges or combinational edges. As we move farther into the network, it can recognize even more complicated features such as eyes, ears, nose, etc.

**Reading and Generating New Images**

Working with image datasets, you will often find a situation when there is not much data to train your model. In this type of situation, you need the help of image augmentation. The image augmentation method is an excellent way to increase the size of your dataset. It comes up with new transformed images from the original dataset. Each new image is different from the other in the aspect of augmentation methods you apply like shifting, rotating, flipping, zoom, etc. Using variations on the original image does not alter its target class but simply gives a new perspective to the images.

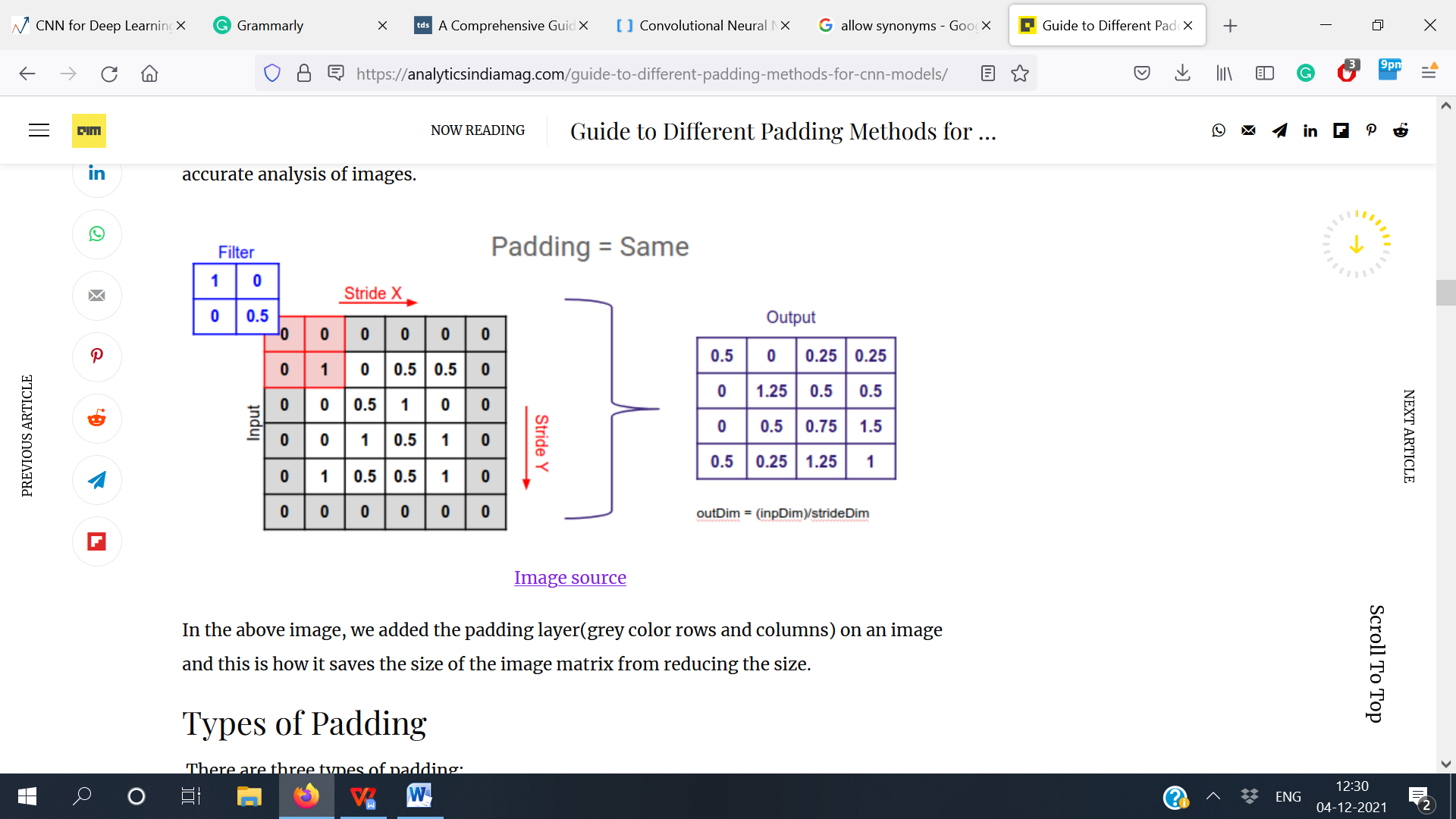
The Keras ImageDataGenerator allows you to augment the images. You can implement any arbitrary changes on any training image. It will make your model robust. Now let me assist you to dive into Keras ImageDataGenerator to see how it is implemented.





**Parameters of Convolutional Layers**

Padding increases the dimension of an image in which a convolutional processes. The kernel/filter which moves throughout the image scans each pixel and transforms the image into a smaller image. A padding layer adds zero to the outer frame of the image for more space for the filter to cover. Adding padding to an image treated by a CNN enables a more accurate analysis of images.

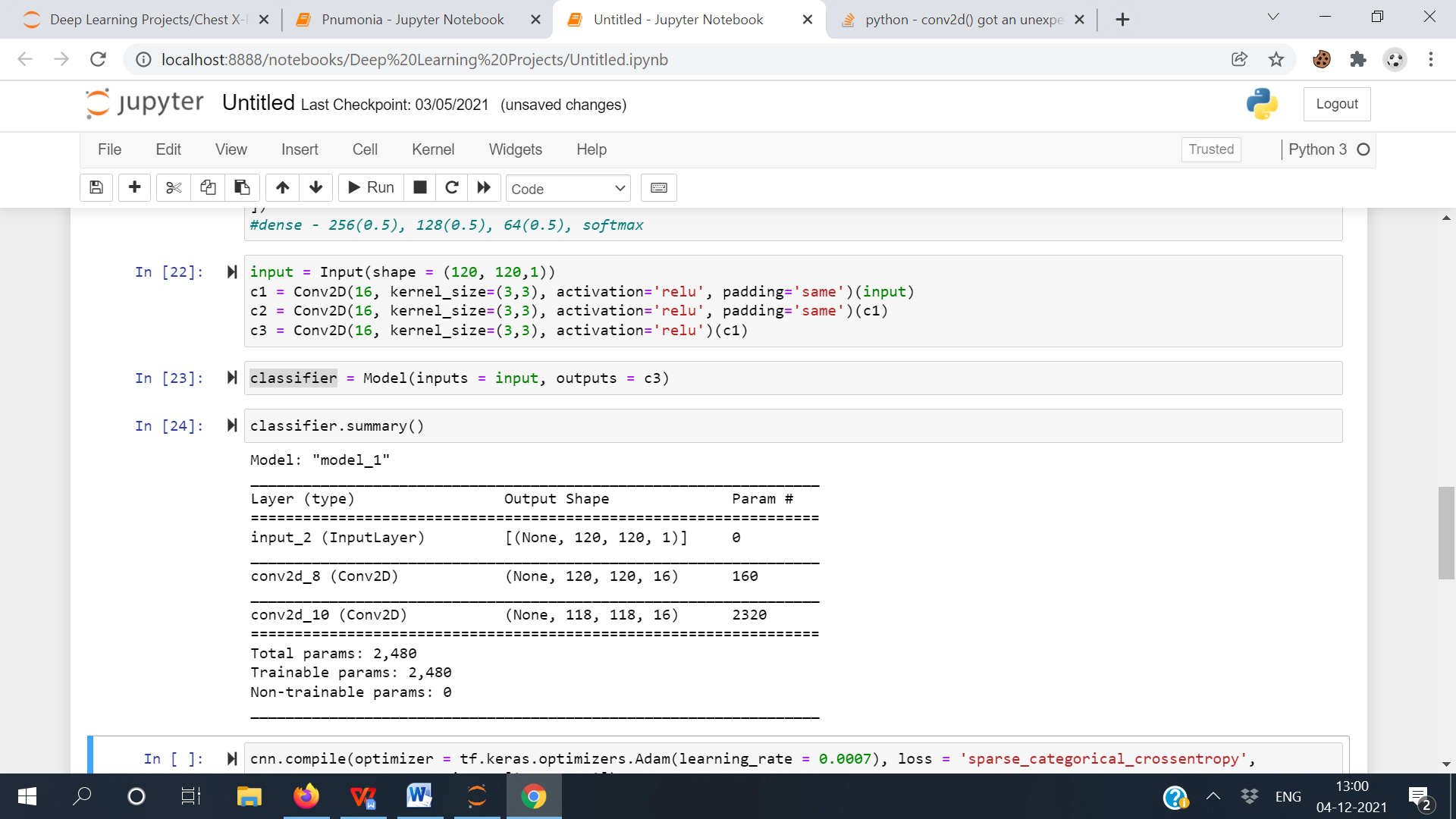


**Types of padding:**

* Same padding
* Causal padding
* Valid padding

### Same Padding

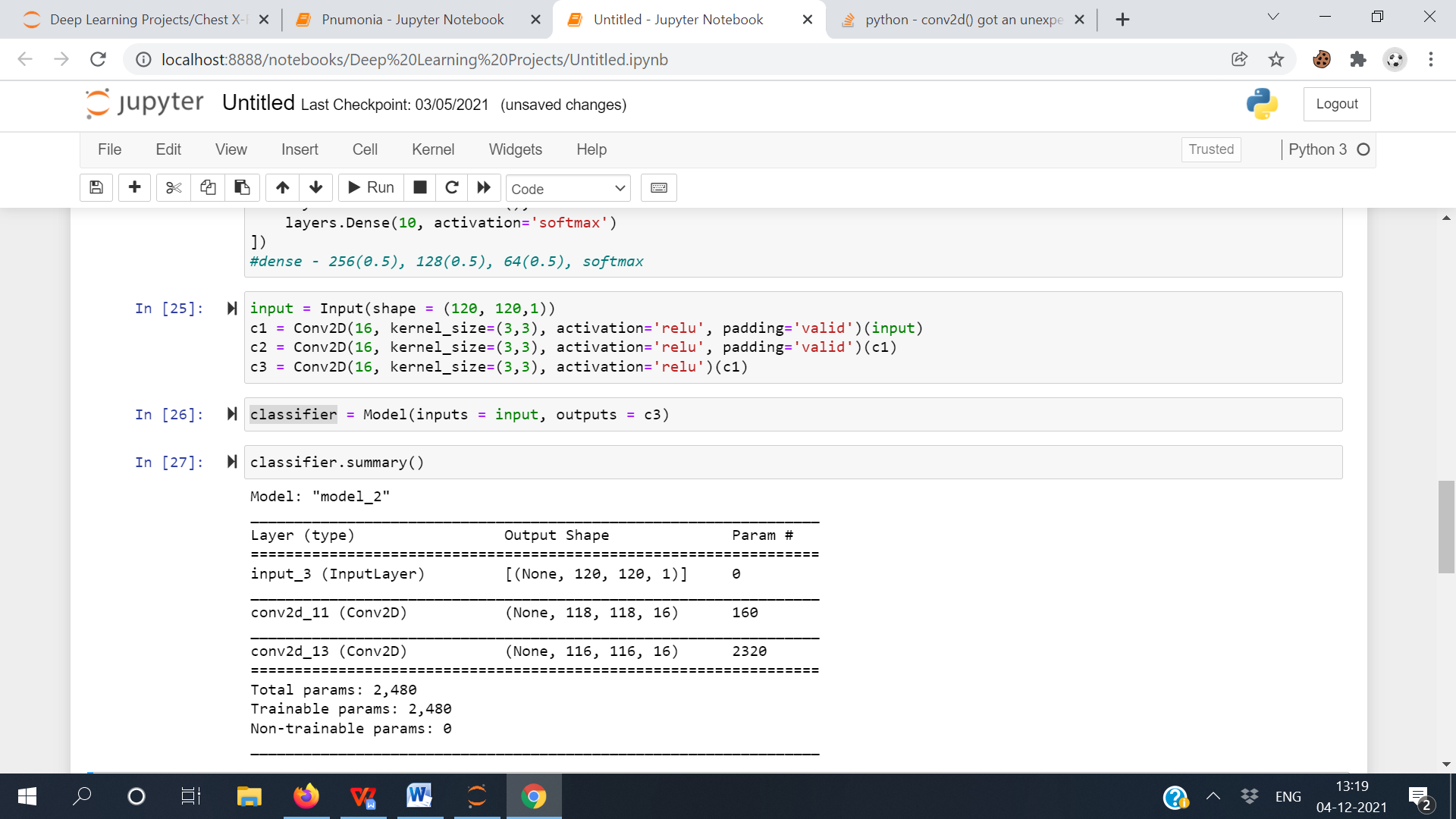
The same padding adds zero values in the outer frame of the images, so the filter we are applying can reach the edge of the matrix and make the result with them too. Below is an illustration of how we can build a model using the same padding.



From the summary of the model with the same padding and without padding, we can see how the size of the output dimensions change. In this way, we can prevent the loss of output size.

### ****Valid Padding****

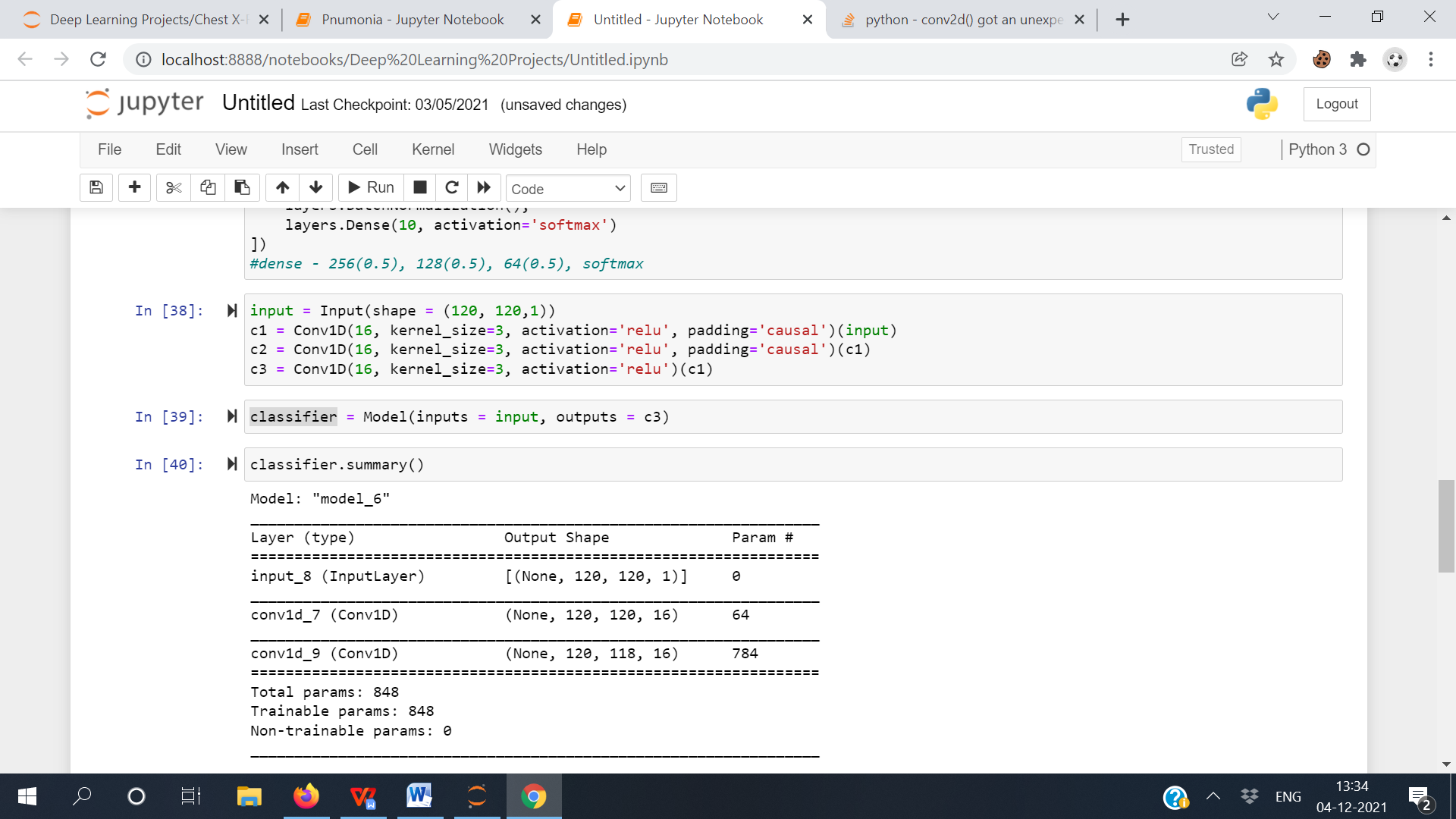
The valid padding is like no padding. Here, we don’t use padding, as we think that all pixel of the image is valid as each input can get thoroughly covered by the filter. Below is an illustration of how we can build a model using valid padding.



From the summary of the model with the same padding and without padding, we can see a similar change in the dimensions of the output image.

### ****Causal Padding****

The causal padding works with the 1-D convolutional layers. We use it mainly in time series analysis. Time-series formed from sequential data causal padding aids in adding zeros at the origin of the data. Some convolutional layers use this type of padding as each layer uses the learned weights of the previous layer. Below is an illustration of how we can build a model using causal padding.



From the summary of the model with the same padding and without padding, we can see the size of the output is not changing, which implies that it helps by adding zero values but in a 1-D convolutional layer.

**Activation Functions**

The activation function is a connection put at the end of or between two neural networks. They improve the decision-making of a neuron to fire or not. Some activation functions are as follows:-

* Binary step function
* Linear activation function
* Sigmoid
* Tanh
* Rectified Linear Unit
* Leaky ReLU
* Parametric ReLU
* Exponential Linear Unit, or ELU
* Softmax
* Swish
* Gaussian Error Linear Unit (GELU)
* Mish

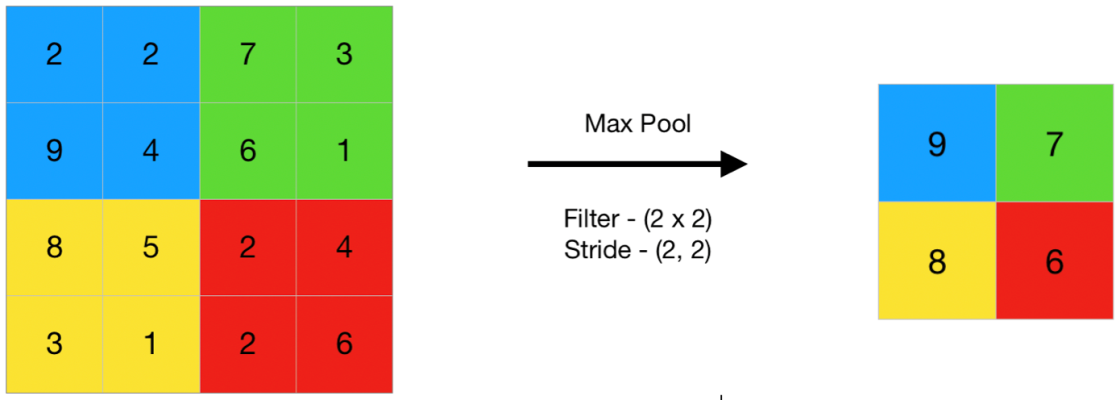
**Pooling Layers**

Pooling layers decrease the dimensions of the feature maps. Thus, it lessens the number of parameters to learn and the measure of computation conducted in the network. It summarises the features in a region of the feature map created by a convolution layer. The most common types of pooling operations are:-

* Max-Pooling
* Min-Pooling
* Avg-Pooling

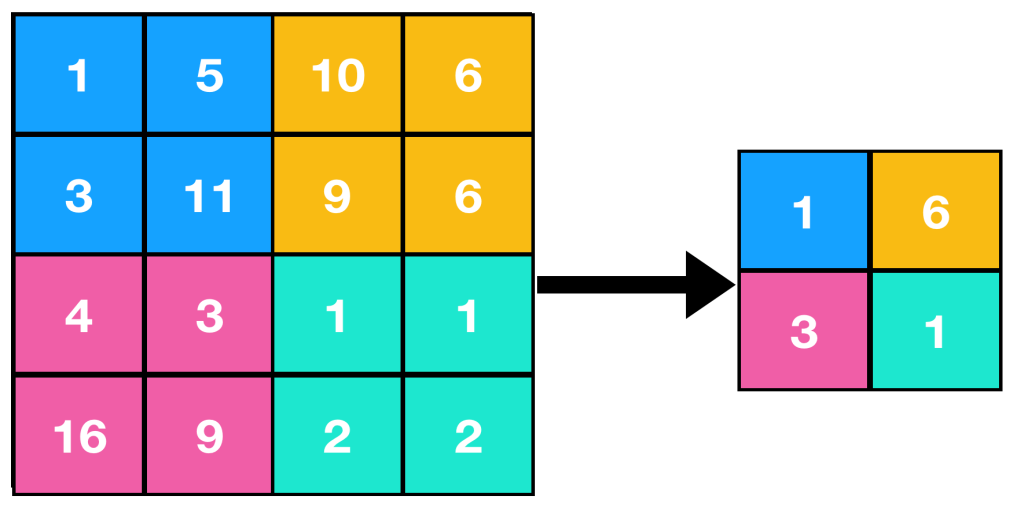
**Max-Pooling**

Max-Pooling, or Maximum pooling, is a pooling performance that determines the highest pixel value in each patch of the feature map. The pooled feature maps highlight most of the features present in the patch. The image shown below describes how the Max-Pooling layer performs.



**Min-Pooling**

Min-Pooling, or Minimum pooling, is a pooling performance that determines the lowest pixel value in each patch of the feature map. The pooled feature maps highlight most of the features present in the patch. The image shown below describes how the Min-Pooling layer performs.



**Avg-Pooling**

Avg-Pooling, or Average pooling, is a pooling performance that determines the average of the pixel value in each patch of the feature map. The pooled feature maps highlight most of the features present in the patch. The image shown below describes how the Avg-Pooling layer performs.

