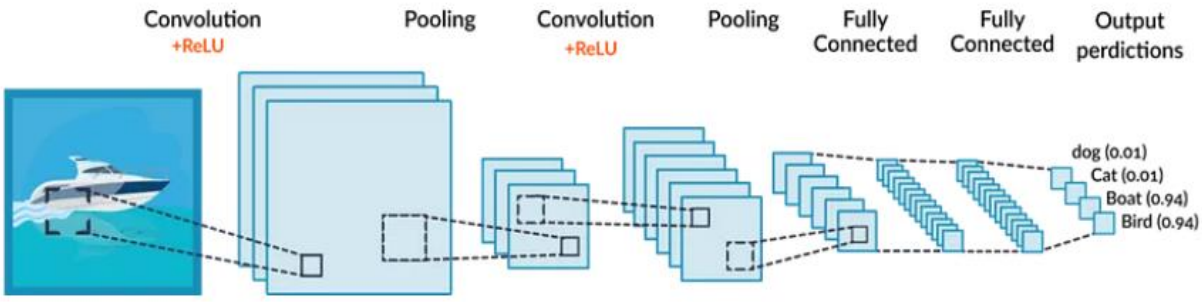
What is CNN?

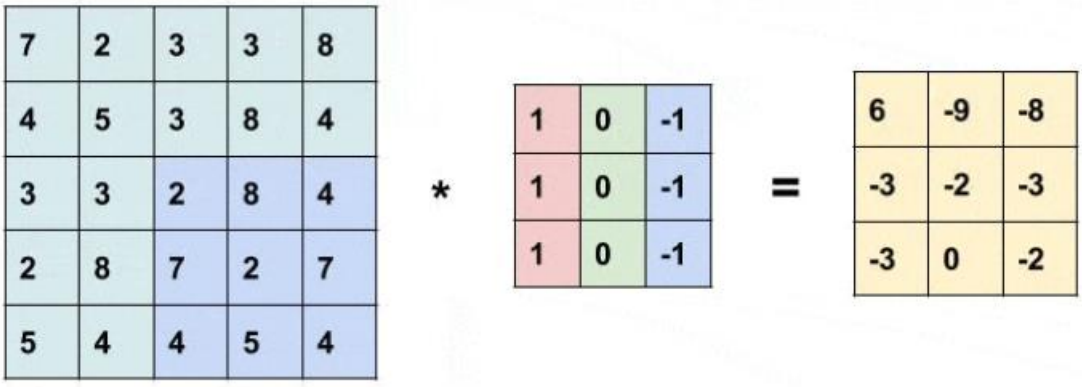
CNN is a deep learning algorithm that is often used in image processing and takes images as input. This algorithm, which captures features in images with different operations and classifies them, consists of different layers. The image that passes through these layers, namely Convolutional Layer, Pooling and Fully Connected, is subjected to different processes and reaches a consistency that can be entered into the deep learning model.



CNN (Convolutional Neural Networks)

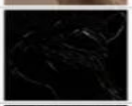
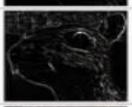
Convolutional Layer

Convolutional layer is the first layer that handles the image in CNN algorithms. As it is known, images are actually matrices consisting of pixels containing certain values. In the convolution layer, a filter smaller than the original image size hovers over the image and tries to capture certain features from these images.



As seen above, a 3×3 filter is hovered over a 5×5 image. The results are written on the feature map, which is our new matrix on the right side of the equation.

The parameters learned in CNN algorithms are the values in these filters. The model constantly updates these values and begins to detect features even better. Apart from this, images can be sharpened, blurred, and edge detected with some known filters. The filters used for these are as follows.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Stride

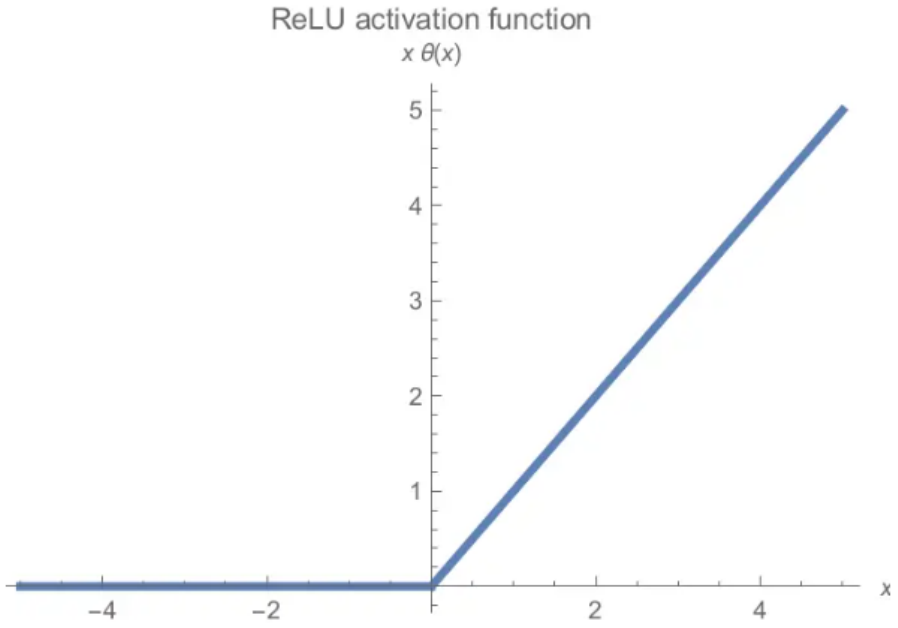
Stride value is a value that can be changed as a parameter in CNN models. This value determines how many pixels the filter will slide on the main image. For example, the stride value in the convolution operation above is one. So the filter does this by skipping just one pixel. If it were Stride 2, the resulting feature map would be smaller as the number of pixels it skips would increase.

Padding

When we apply the filter to an image, the output is smaller than the original image due to the dimensions. The method we can use to prevent this is padding. During the filling process, zeros are added to the image on all four sides, as if it were a frame. These zero added layers can be increased depending on the size of the filter.

ReLU

ReLU (Rectified Linear Unit) is a non-linear function that works as f(x) = max(0,x). For example, a ReLU function that takes the value -25 gives 0 as output, and a function that takes the value 25 gives 25. ReLU, whose main purpose is to get rid of negative values, has a very important position in CNNs.

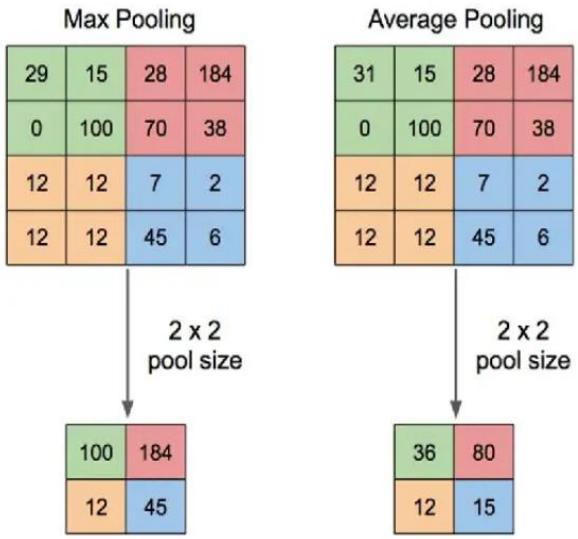


Non-linear functions such as ReLU, tanh and sigmoid are used to prevent our model from learning negative values or not being able to grasp some features due to these negative values.

Pooling

Like the convolutional layer, the pooling layer also aims to reduce dimensionality. In this way, the required processing power is reduced and the unnecessary features captured are ignored and more important features are focused on.

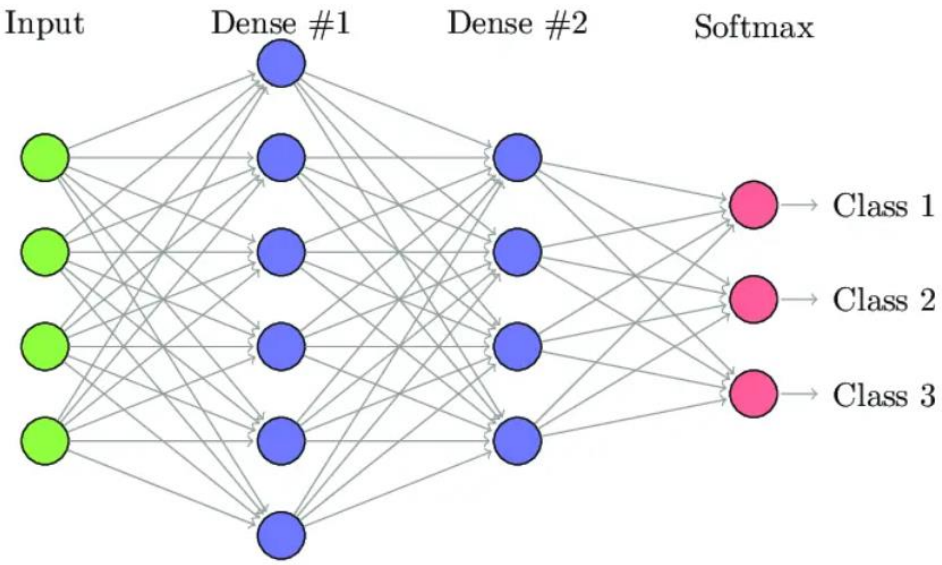
There are two different pooling techniques generally used in CNN models. One of these is Max (Maximum) and the other is Average (Average) pooling.



In the pooling layer, which has a kernel (filter) like the convolutional layer, this kernel also travels over the image. However, instead of convolutional operation, this time it applies the specified pooling technique. That is, if you are applying max pooling, it takes the largest value in the area covered by the filter, and if you are applying average pooling, it takes the average of the values in the filter. In this way, the size is reduced and we retain important features.

Fully Connected Layer

In the Fully Connected layer, our matrix image, which passes through the convolutional layer and pooling layer several times, is turned into a flat vector.



After we have brought our input image to a level where we can train it with neural networks, all that remains is the working logic in classical neural networks. Again, the features are kept in the nodes in the layers and the learning process is started by changing the weight and bias.

Continuous improvements are being made on the CNN algorithm, which basically has this kind of structure. If we ask what are some of the most well-known CNN architectures today, we can list them as follows:

- LeNet
- AlexNet
- VGGNet
- GoogLeNet
- ResNet
- ZFNet

1-)https://teknoloji.org/cnn-convolutional-neural-networks-nedir/
2-)https://bartubozkurt35.medium.com/cnn-convolutional-neural-networks-nedir-a5bafc4a82a1#:~:text=CNN%20genellikle%20g%C3%B6r%C3%BCnt%C3%BC%20i%C5%9Flemede%20kullan%C4%B1lan,bu%20algoritma%20farkl%C4%B1%20katmanlardan%20olu%C5%9Fmaktad%C4%B1r.