**Convolutional Neural Networks for Dummies**

[[Prathammodi](https://medium.com/@prathammodi001?source=post_page-----e68f464d608f--------------------------------)](https://medium.com/@prathammodi001?source=post_page-----e68f464d608f--------------------------------)

[Prathammodi](https://medium.com/@prathammodi001?source=post_page-----e68f464d608f--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Ff774a0701bc6&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40prathammodi001%2Fconvolutional-neural-networks-for-dummies-a-step-by-step-cnn-tutorial-e68f464d608f&user=Prathammodi&userId=f774a0701bc6&source=post_page-f774a0701bc6----e68f464d608f---------------------post_header-----------)

6 min read

·

Oct 14

55

So you want to learn about Convolutional Neural Networks, CNNs, huh? Well, you’ve come to the right place.

This step-by-step tutorial is going to break down CNNs in simple terms.

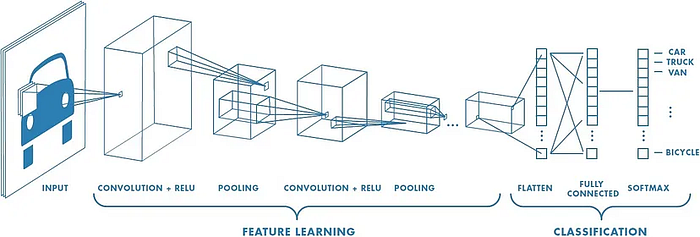
We’ll go through setting up your environment, building a CNN model, and training it to classify images. By the end, you’ll have built and trained your own CNN that can identify different breeds of dogs with pretty good accuracy.

Sound like fun? Let’s get started!

**What Are Convolutional Neural Networks (CNNs)?**

A convolutional neural network or CNN is a type of artificial neural network used in image recognition and processing that is inspired by the biological processes in the visual cortex of animals. They are made up of neurons that have learnable weights and biases.

CNNs use a technique called convolution instead of general matrix multiplication in at least one of their layers. Convolution is a specialized kind of linear operation.



**Fig 1: Neural network with many convolutional layers**

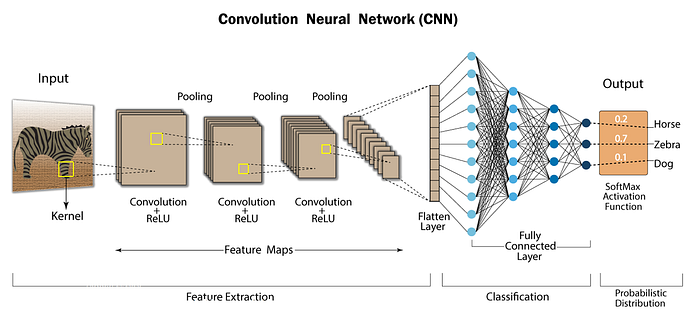
**Working of Convolutional Neural Network:**

CNNs apply filters (small rectangles) to an input image to detect features like edges or shapes. The filters slide over the width and height of the input image and compute dot products between the filter and the input to produce an activation map.

Activation maps are fed into pooling layers that downsample the maps to reduce the dimensionality. This makes the model more efficient and robust. The final layer is a fully connected layer that classifies the input image into categories like “dog” or “cat”.

Some popular CNN architectures are AlexNet, VGGNet, ResNet, and Inception. These have been used to solve complex problems like identifying thousands of objects or detecting diseases from medical scans.

To build a CNN, you define the architecture by selecting hyperparameters like number of filters, filter size, stride, and pooling size. Then you train the network on a large dataset and use backpropagation to update the weights and biases. With enough data and computing power, CNNs can achieve superhuman performance on many visual tasks. CNNs have revolutionized computer vision and are used by companies like Google, Facebook, and others to power image recognition in apps and services. They have become an indispensable tool for any machine learning practitioner.



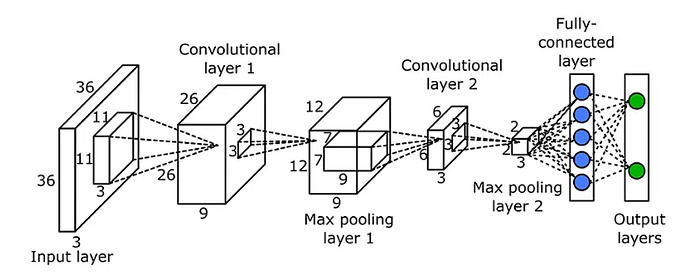
**Fig 2: Convolutional Neural Network — CNN architecture**

**Layers of Convolutional neural network:**

**1. Convolutional Layers**

Convolutional layers apply a convolution operation to the input, passing a filter over the entire image. This filter detects features like edges or curves in the image. Multiple filters can detect different features.

The convolution operation combines the input and filter to create a feature map. This shows the locations and strength of the features detected. By stacking multiple convolutional layers, the network can detect higher-level and more complex features.



**Fig 3: Convolutional Layer**

**2. Pooling Layers**

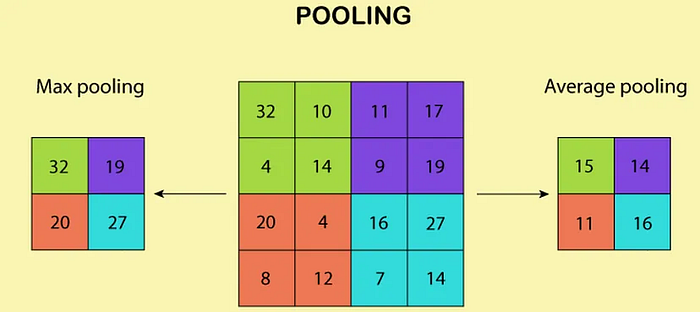
Pooling layers are inserted between convolutional layers. They downsample the feature maps to reduce the number of parameters, control overfitting and make the network invariant to small translations.

The most common types are max pooling, which takes the largest value in a kernel, and average pooling, which takes the average. Pooling layers subsample the feature map, keeping only the most important information.

By cleverly stacking multiple convolutional and pooling layers, a CNN can learn to detect complex features in images like faces, objects, scenes, etc. The output of the final convolutional layer is then flattened into a single vector and passed to a fully connected layer for classification.

Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

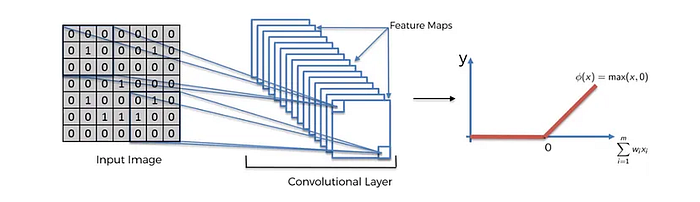
* Max Pooling
* Average Pooling
* Sum Pooling



**Fig 4: MaxPooling Layer**

**3. Activation Layer:**

The activation layer applies a non-linear activation function, such as the ReLU function, to the output of the pooling layer. This function helps to introduce non-linearity into the model, allowing it to learn more complex representations of the input data.

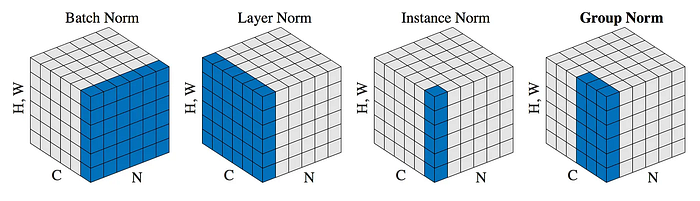


**Fig 5: Activation Layer**

With the basics of convolutional and pooling layers under your belt, you now understand the fundamentals of CNNs! In the next section, we’ll explore how to build a CNN in Keras.

**4. Normalization Layer:**

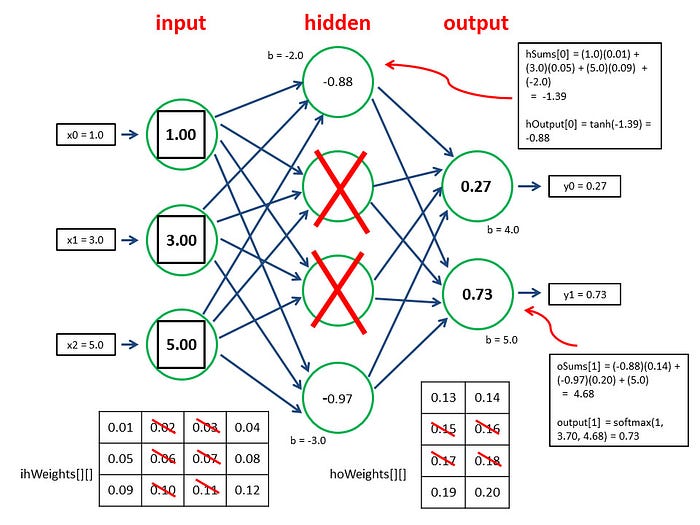
The normalization layer performs normalization operations, such as batch normalization or layer normalization, to ensure that the activations of each layer are well-conditioned and prevent overfitting.



**FIg 6: Different Types of Normalization**

**5. Dropout Layer:**

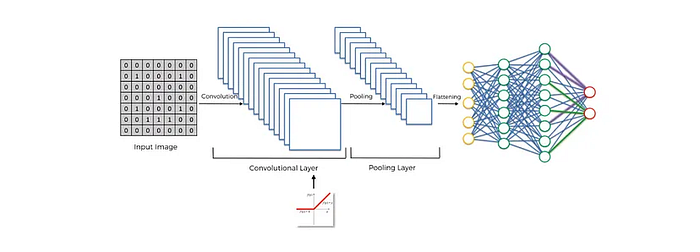
The dropout layer is used to prevent overfitting by randomly dropping out neurons during training. This helps to ensure that the model does not memorize the training data but instead generalizes to new, unseen data.



**Fig 7: How does Dropout Layer Work?**

**6. Dense Layer:**

After the convolutional and pooling layers have extracted features from the input image, the dense layer can then be used to combine those features and make a final prediction. In a CNN, the dense layer is usually the final layer and is used to produce the output predictions. The activations from the previous layers are flattened and passed as inputs to the dense layer, which performs a weighted sum of the inputs and applies an activation function to produce the final output.



**Fig 8: Dense Layer**

**Benefits of Convolutional Neural Network:**

1. **Feature extraction:** CNNs are capable of automatically extracting relevant features from an input image, reducing the need for manual feature engineering.
2. **Spatial invariance:** CNNs can recognize objects in an image regardless of their location, size, or orientation, making them well-suited to object recognition tasks.
3. **Robust to noise:** CNNs can often handle noisy or cluttered images, making them useful for real-world applications where image quality may be variable.
4. **Transfer learning:** CNNs can leverage pre-trained models, reducing the amount of data and computational resources required to train a new model.
5. **Performance:** CNNs have demonstrated state-of-the-art performance on a range of computer vision tasks, including image classification, object detection, and semantic segmentation.

**Limitations of Convolutional Neural Network:**

1. **Computational cost:** Training a deep CNN can be computationally expensive, requiring significant amounts of data and computational resources.
2. **Overfitting:** Deep CNNs are prone to overfitting, especially when trained on small datasets, where the model may memorize the training data rather than generalize to new, unseen data.
3. **Lack of interpretability:** CNNs are considered to be a “black box” model, making it difficult to understand why a particular prediction was made.
4. **Limited to grid-like structures:** CNNs are limited to grid-like structures and cannot handle irregular shapes or non-grid-like data structures.

**Conclusion:**

In conclusion, Convolutional Neural Networks (CNNs) is a powerful deep learning architecture well-suited to image classification and object recognition tasks. With its ability to automatically extract relevant features, handle noisy images, and leverage pre-trained models, CNNs have demonstrated state-of-the-art performance on a range of computer vision tasks. However, they also have their limitations, including a high computational cost, overfitting, a lack of interpretability, and a limited ability to handle irregular shapes. Nevertheless, CNNs remain a popular choice for many computer vision tasks and are likely to continue to be a key area of research and development in the coming years.

If you’re more interested in coding rather than theory, do check out the in-depth tutorial in using CNN to build a Multi-Class Dog Classifier (a shameless plugin)