



Introduction to Convolution Neural Network

Last Updated : 10 Oct, 2024

A **Convolutional Neural Network (CNN)** is a type of [Deep Learning neural network](#) architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.

When it comes to Machine Learning, [Artificial Neural Networks](#) perform really well. Neural Networks are used in various datasets like images, audio, and text. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use [Recurrent Neural Networks](#) more precisely an [LSTM](#), similarly for image classification we use Convolution Neural networks. In this blog, we are going to build a basic building block for CNN.

Table of Content

- [Neural Networks: Layers and Functionality](#)
- [Convolution Neural Network](#)
 - [CNN Architecture](#)
 - [How Convolutional Layers Works?](#)
 - [Mathematical Overview of Convolution](#)
 - [Layers Used to Build ConvNets](#)
- [Example: Applying CNN to an Image](#)
- [Advantages and Disadvantages of Convolutional Neural Networks \(CNNs\)](#)

Neural Networks: Layers and Functionality

In a regular Neural Network there are three types of layers:

1. **Input Layers:** It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

2. **Hidden Layer:** The input from the Input layer is then fed into the hidden layer. There can be many hidden layers depending on our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of the output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.
3. **Output Layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

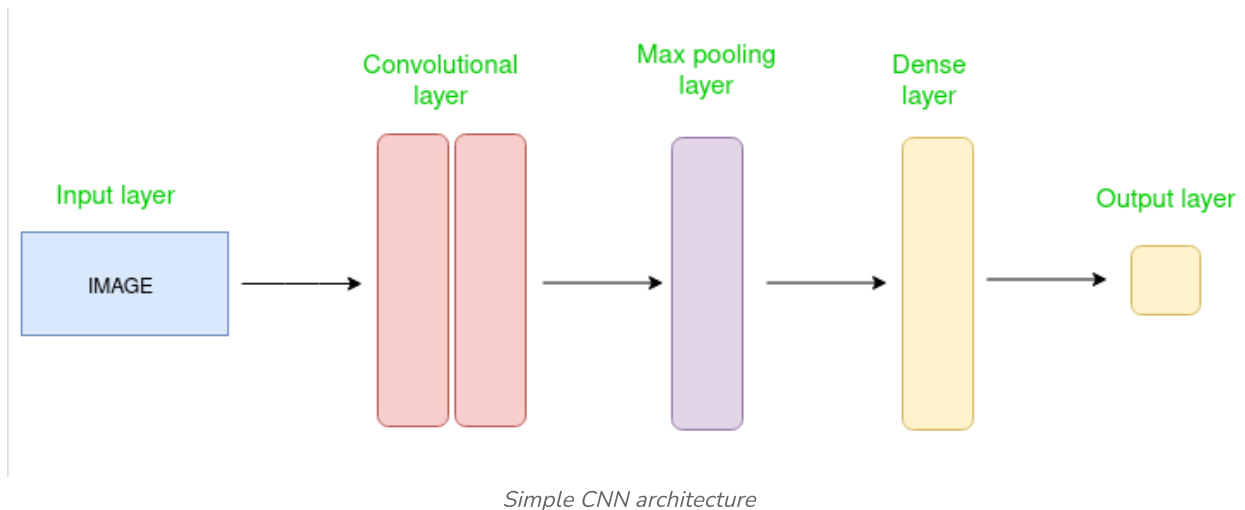
The data is fed into the model and output from each layer is obtained from the above step is called **feedforward**, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called **Backpropagation** which basically is used to minimize the loss.

Convolution Neural Network

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. For example visual datasets like images or videos where data patterns play an extensive role.

CNN Architecture

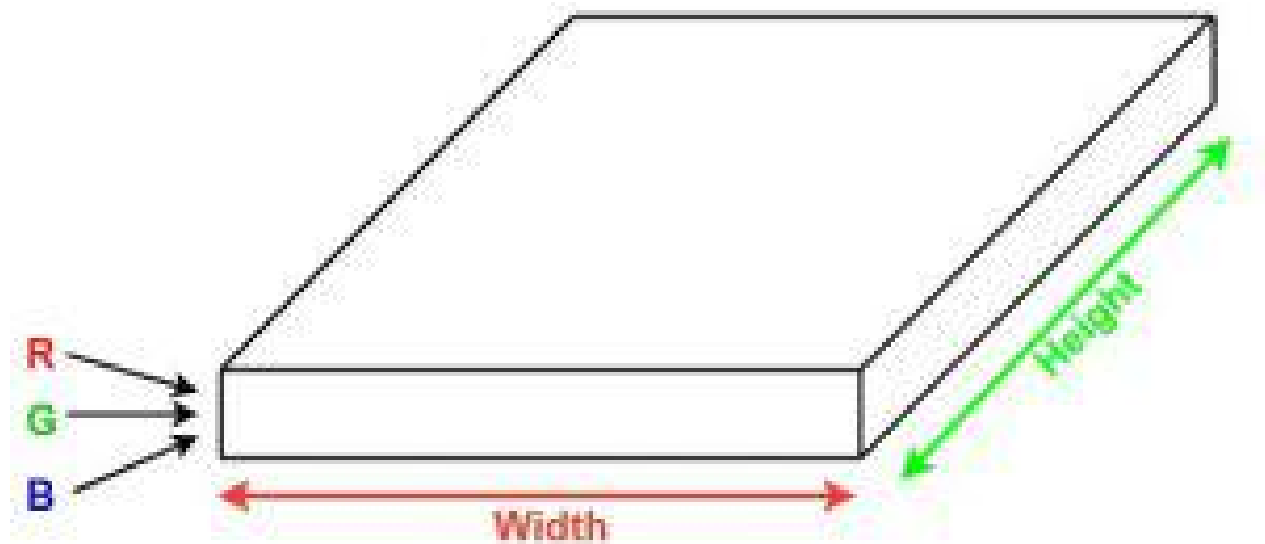
Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.



The Convolutional layer applies filters to the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

How Convolutional Layers Works?

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e the channel as images generally have red, green, and blue channels).



Now imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a

result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called **Convolution**. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.

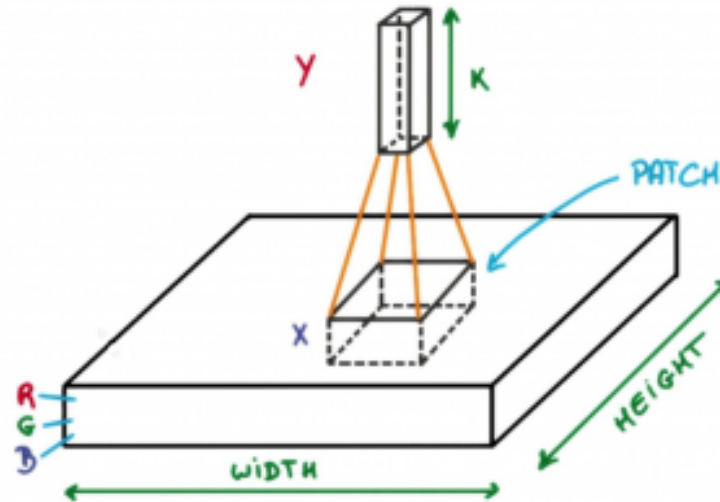


Image source: Deep Learning Udacity

Mathematical Overview of Convolution

Now let's talk about a bit of mathematics that is involved in the whole convolution process.

- Convolution layers consist of a set of learnable filters (or kernels) having small widths and heights and the same depth as that of input volume (3 if the input layer is image input).
- For example, if we have to run convolution on an image with dimensions $34 \times 34 \times 3$. The possible size of filters can be $a \times a \times 3$, where 'a' can be anything like 3, 5, or 7 but smaller as compared to the image dimension.
- During the forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have a value of 2, 3, or even 4 for high-dimensional images) and compute the dot product between the kernel weights and patch from input volume.
- As we slide our filters we'll get a 2-D output for each filter and we'll stack them together as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.

Layers Used to Build ConvNets

A complete Convolution Neural Networks architecture is also known as convnets. A convnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

Types of layers: datasets

Let's take an example by running a convnets on of image of dimension $32 \times 32 \times 3$.

- **Input Layers:** It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.
- **Convolutional Layers:** This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2×2 , 3×3 , or 5×5 shape. it slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred as feature maps. Suppose we use a total of 12 filters for this layer we'll get an output volume of dimension $32 \times 32 \times 12$.
- **Activation Layer:** By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are **RELU**: $\max(0, x)$, **Tanh**, **Leaky RELU**, etc. The volume remains unchanged hence output volume will have dimensions $32 \times 32 \times 12$.
- **Pooling layer:** This layer is periodically inserted in the convnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are **max pooling** and **average pooling**. If we use a max pool with 2×2 filters and stride 2, the resultant volume will be of dimension $16 \times 16 \times 12$.

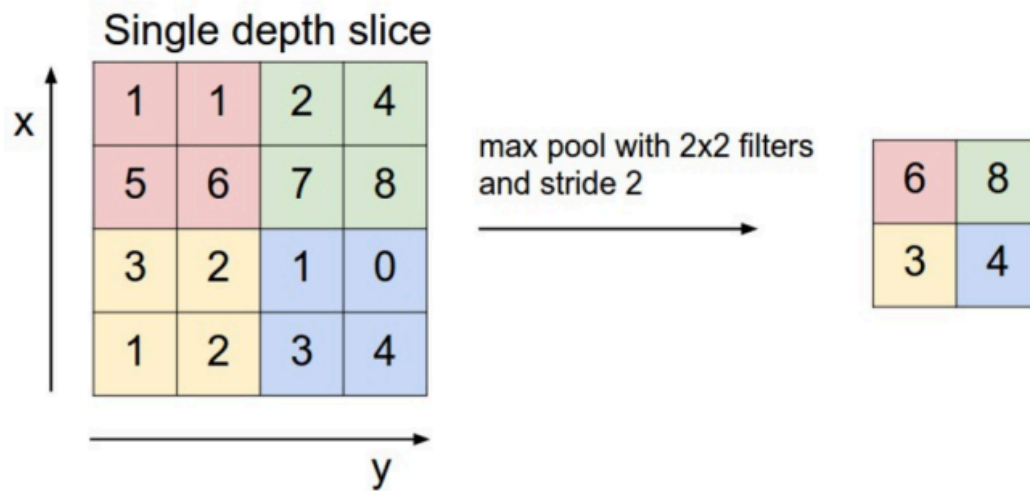


Image source: cs231n.stanford.edu

- **Flattening:** The resulting feature maps are flattened into a one-dimensional

Data Science Data Science Projects Data Analysis Data Visualization Machine Learning ML Projects Deep L

completely linked layer for categorization or regression.

- **Fully Connected Layers:** It takes the input from the previous layer and computes the final classification or regression task.

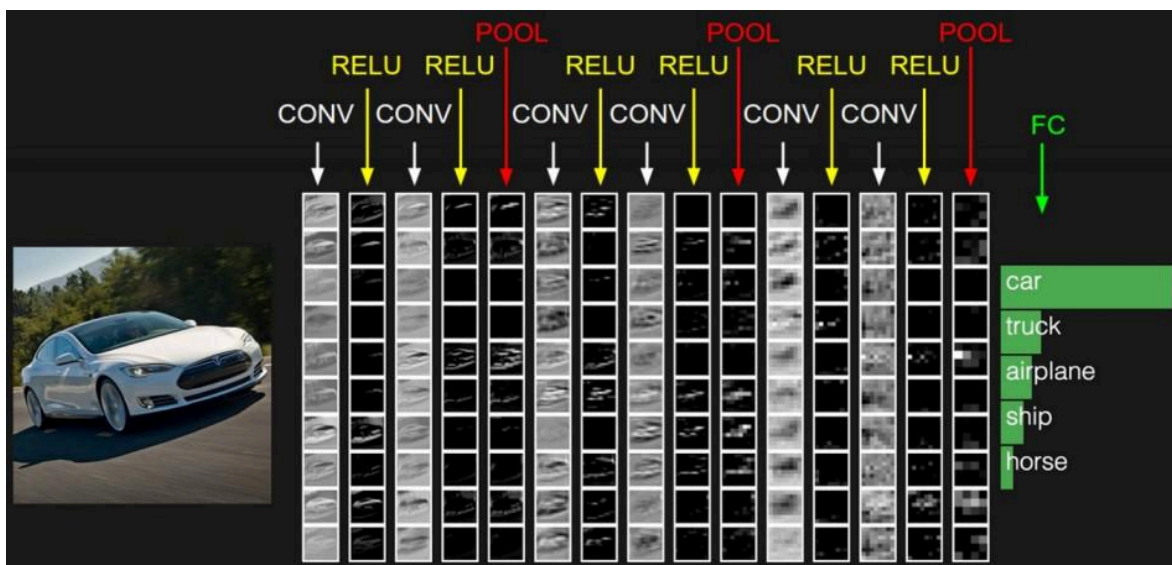


Image source: cs231n.stanford.edu

- **Output Layer:** The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

Example: Applying CNN to an Image

Let's consider an image and apply the convolution layer, activation layer, and pooling layer operation to extract the inside feature.

Input image:



Input image

Step:

- import the necessary libraries
- set the parameter
- define the kernel
- Load the image and plot it.
- Reformat the image
- Apply convolution layer operation and plot the output image.
- Apply activation layer operation and plot the output image.
- Apply pooling layer operation and plot the output image.

```
# import the necessary libraries
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from itertools import product

# set the param
plt.rc('figure', autolayout=True)
plt.rc('image', cmap='magma')

# define the kernel
kernel = tf.constant([[ -1, -1, -1],
                      [-1,  8, -1],
                      [-1, -1, -1],
                      ])
```



```
# Load the image
image = tf.io.read_file('Ganesh.jpg')
image = tf.io.decode_jpeg(image, channels=1)
image = tf.image.resize(image, size=[300, 300])

# plot the image
img = tf.squeeze(image).numpy()
plt.figure(figsize=(5, 5))
plt.imshow(img, cmap='gray')
plt.axis('off')
plt.title('Original Gray Scale image')
plt.show();

# Reformat
image = tf.image.convert_image_dtype(image, dtype=tf.float32)
image = tf.expand_dims(image, axis=0)
kernel = tf.reshape(kernel, [*kernel.shape, 1, 1])
kernel = tf.cast(kernel, dtype=tf.float32)

# convolution layer
conv_fn = tf.nn.conv2d

image_filter = conv_fn(
    input=image,
    filters=kernel,
    strides=1, # or (1, 1)
    padding='SAME',
)

plt.figure(figsize=(15, 5))

# Plot the convolved image
plt.subplot(1, 3, 1)

plt.imshow(
    tf.squeeze(image_filter)
)
plt.axis('off')
plt.title('Convolution')

# activation layer
relu_fn = tf.nn.relu
# Image detection
image_detect = relu_fn(image_filter)

plt.subplot(1, 3, 2)
plt.imshow(
    # Reformat for plotting
    tf.squeeze(image_detect)
)

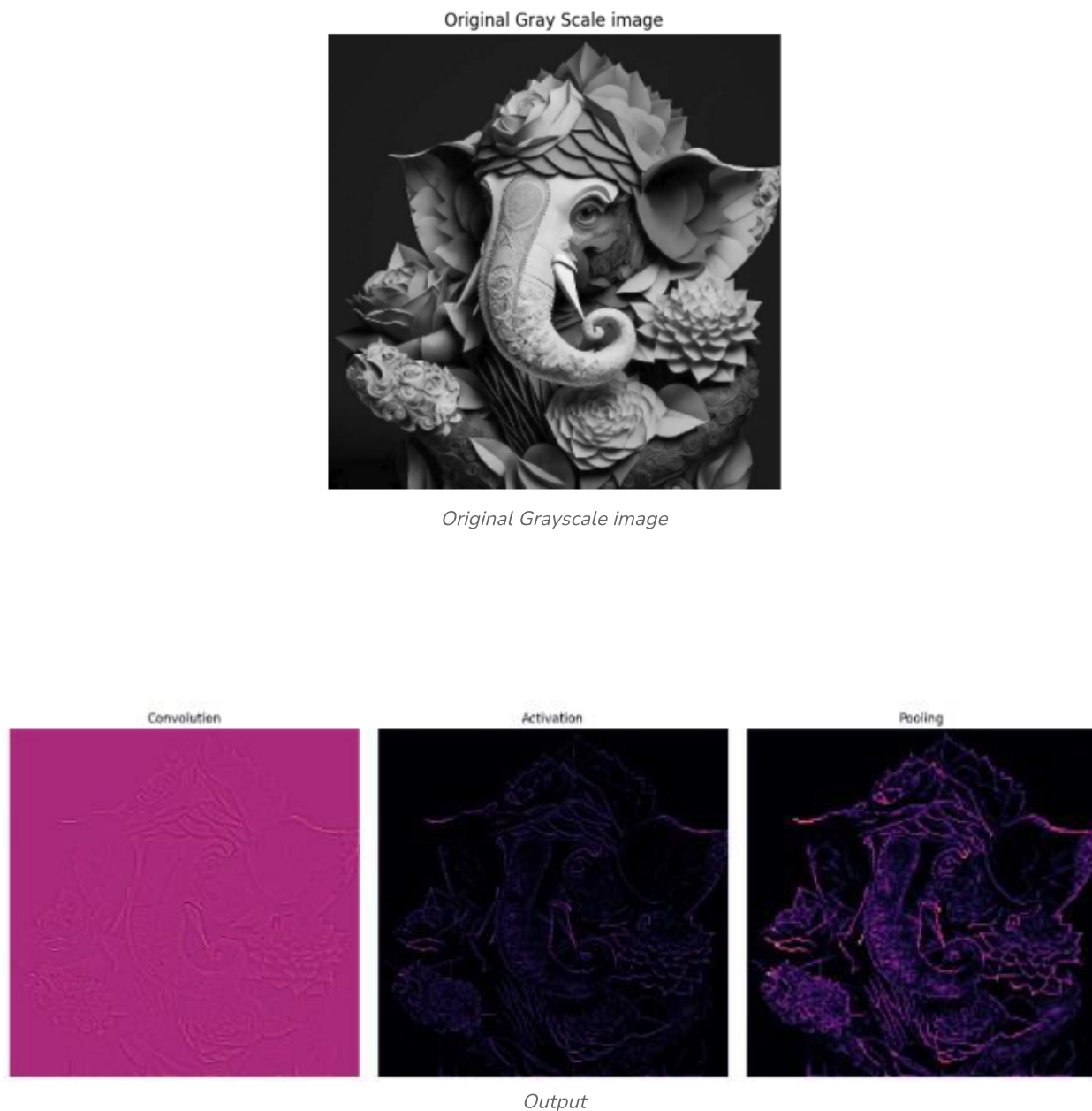
plt.axis('off')
plt.title('Activation')

# Pooling layer
pool = tf.nn.pool
image_condense = pool(input=image_detect,
                      window_shape=(2, 2),
```



```
pooling_type='MAX',  
strides=(2, 2),  
padding='SAME',  
)  
  
plt.subplot(1, 3, 3)  
plt.imshow(tf.squeeze(image_condense))  
plt.axis('off')  
plt.title('Pooling')  
plt.show()
```

Output:



Advantages and Disadvantages of Convolutional Neural Networks (CNNs)

Advantages of CNNs:

1. Good at detecting patterns and features in images, videos, and audio signals.
2. Robust to translation, rotation, and scaling invariance.
3. End-to-end training, no need for manual feature extraction.
4. Can handle large amounts of data and achieve high accuracy.

Disadvantages of CNNs:

1. Computationally expensive to train and require a lot of memory.
2. Can be prone to overfitting if not enough data or proper regularization is used.
3. Requires large amounts of labeled data.
4. Interpretability is limited, it's hard to understand what the network has learned.

Convolution Neural Network – FAQs

What is a Convolutional Neural Network (CNN)?

A Convolutional Neural Network (CNN) is a type of deep learning neural network that is well-suited for image and video analysis. CNNs use a series of convolution and pooling layers to extract features from images and videos, and then use these features to classify or detect objects or scenes.

How do CNNs work?

CNNs work by applying a series of convolution and pooling layers to an input image or video. Convolution layers extract features from the input by sliding a small filter, or kernel, over the image or video and computing the dot product between the filter and the input. Pooling layers then downsample the output of the convolution layers to reduce the dimensionality of the data and make it more computationally efficient.

What is the difference between CNN and convolution?

- *CNN (Convolutional Neural Network) is a type of deep learning neural network designed to process grid-like data, such as images, by using layers of convolutions to extract features.*
- *Convolution, on the other hand, is the specific mathematical operation within CNNs that applies filters (kernels) to the input data (like an image) to detect patterns such as edges or textures.*

What is the basic principle of CNN?

The basic principle of a Convolutional Neural Network (CNN) is to automatically learn and extract hierarchical features from input data, typically images, through the use of convolutional layers.

What is convolution and its types?

Convolution is a mathematical operation applied in Convolutional Neural Networks (CNNs) to extract features from input data, such as images. In the context of CNNs, convolution involves sliding a filter (kernel) over the input data, computing the dot product between the filter and a small patch of the input, and producing a feature map.

How many layers are in CNN?

There is no fixed number of layers in a CNN, as it varies depending on the architecture and task.

What is the purpose of using multiple convolution layers in a CNN?

Using multiple convolution layers in a CNN allows the network to learn increasingly complex features from the input image or video. The first convolution layers learn simple features, such as edges and corners. The deeper convolution layers learn more complex features, such as shapes and objects.

What is the difference between a convolution layer and a pooling layer?

A convolution layer extracts features from an input image or video, while a pooling layer downsamples the output of the convolution layers. Convolution layers use a series of filters to extract features, while pooling layers use a variety of techniques to downsample the data, such as max pooling and average pooling.

[Comment](#)[More info](#)[Advertise with us](#)

Next Article

[Digital Image Processing Basics](#)

Similar Reads

Deep Learning Tutorial

Deep Learning tutorial covers the basics and more advanced topics, making it perfect for beginners and those with experience. Whether you're just starting or looking to expand your knowledge, this guide makes it easy...

5 min read

Introduction to Deep Learning

Basic Neural Network

Activation Functions

Artificial Neural Network

Classification

Regression

Hyperparameter tuning

Introduction to Convolution Neural Network

Introduction to Convolution Neural Network

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to...

10 min read

Digital Image Processing Basics

Digital Image Processing means processing digital image by means of a digital computer. We can also say that it is a use of computer algorithms, in order to get enhanced image either to extract some useful...

7 min read

Difference between Image Processing and Computer Vision

Image processing and Computer Vision both are very exciting field of Computer Science. Computer Vision: In Computer Vision, computers or machines are made to gain high-level understanding from the input digital...

2 min read

CNN | Introduction to Pooling Layer

Pooling layer is used in CNNs to reduce the spatial dimensions (width and height) of the input feature maps while retaining the most important information. It involves sliding a two-dimensional filter over each chann...

5 min read

CIFAR-10 Image Classification in TensorFlow

Prerequisites:Image ClassificationConvolution Neural Networks including basic pooling, convolution layers with normalization in neural networks, and dropout.Data Augmentation.Neural Networks.Numpy arrays.In...

8 min read

Implementation of a CNN based Image Classifier using PyTorch

Introduction: Introduced in the 1980s by Yann LeCun, Convolution Neural Networks(also called CNNs or ConvNets) have come a long way. From being employed for simple digit classification tasks, CNN-based...

9 min read

Convolutional Neural Network (CNN) Architectures

Convolutional Neural Network(CNN) is a neural network architecture in Deep Learning, used to recognize the pattern from structured arrays. However, over many years, CNN architectures have evolved. Many variants o...

11 min read

Object Detection vs Object Recognition vs Image Segmentation

Object Recognition: Object recognition is the technique of identifying the object present in images and videos. It is one of the most important applications of machine learning and deep learning. The goal of this field is t...

5 min read

YOLO v2 - Object Detection

In terms of speed, YOLO is one of the best models in object recognition, able to recognize objects and process frames at the rate up to 150 FPS for small networks. However, In terms of accuracy mAP, YOLO was not th...

6 min read

Recurrent Neural Network



Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

Company

About Us
Legal
Privacy Policy
In Media
Contact Us

Languages

Python
Java
C++
PHP
GoLang

Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics

GeeksforGeeks Videos

DSA
Python

| | |
|-----------------|-----------------|
| Chemistry | Java |
| Biology | C++ |
| Social Science | Web Development |
| English Grammar | Data Science |
| Commerce | CS Subjects |
| World GK | |

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved