



How do convolutional neural networks (CNNs) work?

Last Updated : 06 Aug, 2025

Convolutional Neural Networks (CNNs) have transformed computer vision by allowing machines to achieve unprecedented accuracy in tasks like image classification, object detection, and segmentation. CNNs, which originated with Yann LeCun's work in the late 1980s, are inspired by the human visual system and process visual data using a hierarchical structure. This article delves into the workings of CNNs, specifically their layers and convolutional operations.

Overview of CNN

[Convolutional Neural Networks](#) (CNNs) are deep neural networks designed to handle grid-like data, such as images. CNNs, unlike traditional neural networks, use convolutional layers to learn spatial feature hierarchies automatically. They are composed of multiple layers, each of which serves a specific function in processing and transforming input data to extract meaningful patterns.

Key Characteristics of CNN

- **Local receptive fields:** CNNs employ small, [machine-learning](#) filters that move across the input image, focusing on specific regions to detect local features such as edges, textures, and patterns.
- **Weight sharing:** The same filter is applied to various regions of the input image, decreasing the number of parameters and computational complexity while allowing the network to recognize objects regardless of position.

- **Pooling:** Pooling layers minimize the spatial dimensions of feature maps, making the network more efficient and adaptable to changes. Max pooling, for example, takes the highest value from each patch, aggregating feature presence while decreasing parameters.

CNNs' structure, which combines local receptive fields, [weight sharing](#), and pooling, makes them extremely efficient and reliable for image processing tasks. This enables CNNs to perform tasks such as image classification, object detection, and segmentation, making them indispensable in computer vision.

Layers in CNN

The convolutional layer is the primary component of a CNN. It works as a convolution operation by sliding a filter (which is also referred to as a kernel) across the input image and calculating the dot product of the filter and the input's receptive field. This operation aids in detecting local characteristics such as edges, textures, and trends. The layers in CNN are:

- **Convolutional Layers:** Apply filters to the input image to extract local features such as edges and textures.
- **Pooling Layers:** Reduce the spatial dimensions of feature maps, decreasing the computational load and enhancing feature robustness.
- **Fully Connected layers:** Connect every neuron in one layer to every neuron in the next, integrating extracted features for final predictions.
- **Dropout:** Randomly sets a fraction of neurons to zero during training to prevent overfitting and improve model generalization.
- **Activation Functions:** Introduce non-linearity into the network, enabling the learning of complex patterns; common examples include ReLU and Sigmoid.

Convolutional Operations and Working

The convolution operation entails sliding a filter across the input image and calculating the dot product between the filter and the local region of interest. This operation generates a feature map that highlights the detected features.

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n)$$

Where I is the input image, K is the kernel, and (i, j) are the coordinates of the output feature maps.

Example of a Convolution Operation

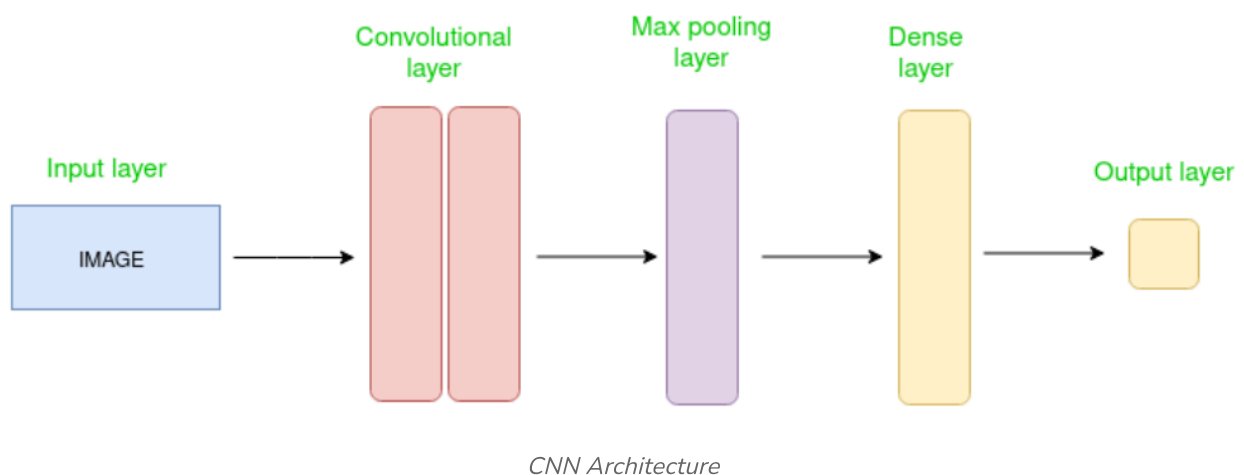
Consider a 3x3 filter used on a 5x5 input image. The filter moves across the image, computing the dot product at each position, yielding a smaller feature map.

Pooling Operation

Pooling reduces the spatial dimensions of the feature map by aggregating the presence of features in different patches of the map.

Workflow of CNN

Convolutional Neural Networks (CNNs) are designed to process and analyze visual data by learning spatial feature hierarchies automatically and adaptively. Here's a thorough explanation of how CNNs operate:



1. Input Layer

The input layer of a CNN receives the image's raw pixel values. A color image typically has three channels (RGB), whereas a grayscale image only has one. For example, a color image measuring 32x32 pixels would have an input dimension of 32x32x3.

2. Convolutional Layers

The main operation in [convolutional layers](#) is convolution, which involves applying filters ([kernels](#)) to the input data. A filter is a small matrix (e.g., 3x3) that moves across an input image and performs element-wise multiplication and summation to yield a single output value. Convolution is the process that produces a feature map or an activation map.

For example, if a 3x3 filter is applied to a 5x5 input image, the filter slides over it, calculating the dot product between the filter and the input. The output feature map captures specific features such as edges, corners, or textures, depending on the filter's learned values.

3. Stride and Padding

- **Stride:** The [stride](#) determines how the filter traverses the input image. A stride of 1 indicates that the filter moves one pixel at a time, whereas a stride of 2 moves two pixels at a time. Higher strides produce smaller output feature maps.
- **Padding:** [Padding](#) is used to control the spatial dimensions of the final feature map. "Same" padding adds zeros around the input's border, resulting in an output feature map with the same dimensions as the input. "Valid" padding does not include any padding, resulting in a smaller output feature map.

4. Activation Functions

[Activation functions](#) are important in neural networks because they introduce nonlinearity, which allows the network to learn complex

patterns and depictions. Without non-linear activation functions, a [neural network](#) behaves like a linear model, regardless of depth.

ReLU(Rectified Linear Unit)

The [ReLU](#) activation function is widely used in CNNs due to its simplicity and effectiveness. It is defined as:

$$ReLU(x) = \max(0, x)$$

This means that if the input value is positive, ReLU outputs it directly; otherwise, it returns zero. ReLU helps to mitigate the vanishing gradient problem, allowing for faster and more effective deep network training by keeping the gradient flow active and non-zero for positive inputs.

Sigmoid

The [sigmoid](#) activation function is frequently used in binary classification tasks' output layers. It converts any input value into a value between 0 and 1, which can be used to calculate probability. Sigmoid function is defined as:

$$Sigmoid(x) = \frac{1}{1+e^{-x}}$$

While the sigmoid function is useful in some applications, it can in some applications, it can suffer from vanishing gradients and slower convergence than ReLU.

5.Pooling Layers

[Pooling layers](#) are used to reduce the spatial dimensions of feature maps, lowering the computational load and the network's parameter count. Pooling is typically used after the convolution and activation layers. Max pooling is the process of extracting the maximum value from each receptive field (for example, 2x2) of the feature map. Average pooling computes the average value for each receptive field.

6. Stacking Layers

CNNs are made up of stacked convolutional and pooling layers. Early layers detect basic features such as edges and textures, while later layers detect more complex structures and objects. This hierarchical feature extraction is critical to the success of CNNs in visual recognition.

7. Fully Connected Layers

Following several convolutional and pooling layers, the network's high-level reasoning is carried out via fully connected layers ([dense layers](#)). Each neuron in these layers is connected to every neuron in the layer before it. The features extracted by the convolutional layers are combined in the fully connected layers to make final predictions.

8. Dropout

[Dropout layers](#) are commonly used in CNNs to prevent overfitting. During training, dropout randomly assigns a fraction of input units to zero at each update cycle. This helps to make the model more generalizable by preventing it from relying too heavily on individual neurons.

Conclusion

Finally, Convolutional Neural Networks (CNNs) efficiently process and analyze visual data using layers of convolution, activation, and pooling. CNNs perform well in tasks like image classification, object detection, and segmentation because they automatically learn hierarchical features. Their ability to capture spatial hierarchies while reducing computational complexity renders them indispensable in modern computer vision applications.

[Comment](#)**V** verma... [+ Follow](#)

1

Article Tags :

[Computer Vision](#)[AI-ML-DS](#)[Interview-Questions](#)

Explore

Introduction to Computer Vision

Image Processing & Transformation

Feature Extraction and Description

Deep Learning for Computer Vision

Object Detection and Recognition

Image Segmentation

3D Reconstruction

40+ Top Computer Vision Projects [2025 Updated]

4 min read



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



Company

[About Us](#)

[Legal](#)

Explore

[POTD](#)

[Job-A-Thon](#)

[Privacy Policy](#)

[Blogs](#)

[Contact Us](#)

[Nation Skill Up](#)

[Advertise with us](#)

[GFG Corporate Solution](#)

[Campus Training Program](#)

Tutorials

[Programming Languages](#)

[DSA](#)

[Web Technology](#)

[AI, ML & Data Science](#)

[DevOps](#)

[CS Core Subjects](#)

[Interview Preparation](#)

[Software and Tools](#)

Courses

[IBM Certification](#)

[DSA and Placements](#)

[Web Development](#)

[Programming Languages](#)

[DevOps & Cloud](#)

[GATE](#)

[Trending Technologies](#)

Videos

[DSA](#)

[Python](#)

[Java](#)

[C++](#)

[Web Development](#)

[Data Science](#)

[CS Subjects](#)

Preparation Corner

[Interview Corner](#)

[Aptitude](#)

[Puzzles](#)

[GfG 160](#)

[System Design](#)

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

[Do Not Sell or Share My Personal Information](#)