Deep Learning with Keras and TensorFlow

Convolutional Neural Networks

Learning Objectives

By the end of this lesson, you will be able to:

- Interpret the structure and functionality of various convolutional neural network (CNN) architectures
- Implement convolutional and pooling layers to extract significant features from images in a neural network
- Apply advanced CNN architectures, such as ResNet, to solve complex image recognition problems
- Utilize TensorBoard to monitor, analyze, and optimize the performance of convolutional neural networks throughout the training process

Business Scenario

A startup is working on an image recognition system designed to aid in medical diagnoses through medical imaging.

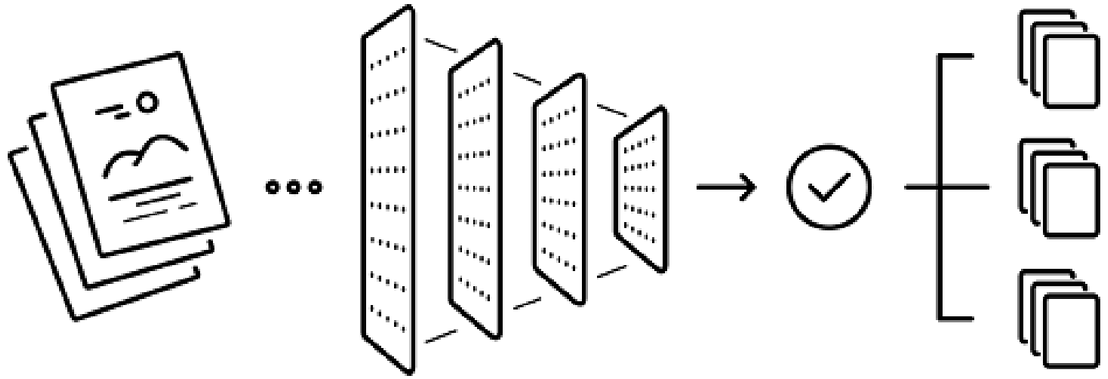
The company explores the application of convolutional neural network (CNN) algorithms, training their system to identify various medical conditions from X-rays and CT scans. They use TensorBoard to visualize the performance of these models, adjusting as needed. In their pursuit of obtaining optimal results, they experiment with various CNN filters, including horizontal and vertical Sobel filters, blur filters, and outline filters, to identify the most effective ones for their specific medical imaging use case.

In a bid to enhance system accuracy, they're contemplating the integration of a residual neural network (ResNet) architecture.

Introduction to CNN

Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a deep learning model specifically designed for visual data analysis, extracting features through convolutional and pooling layers to achieve high-level pattern recognition and classification.



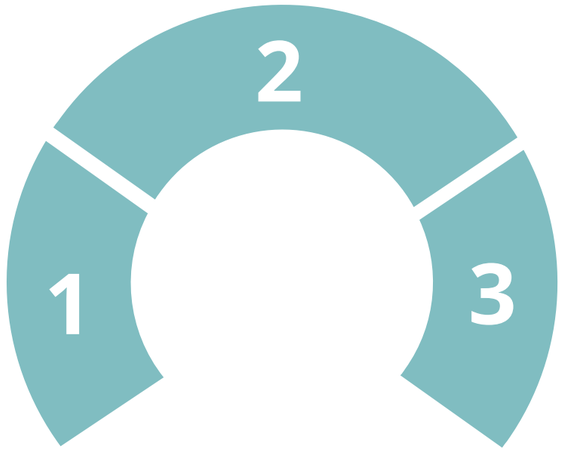
Its architecture is modeled after the visual cortex and has transformed computer vision tasks.

Advantages of CNN

Some of the advantages of CNN are:

2. They automatically detect essential features without any human supervision.

1. Compared to feed-forward neural networks, CNNs have higher accuracy in resolving image recognition problems.



3. The results are more accurate than generic machine-learning techniques.

Disadvantages of CNN

Some of the disadvantages of CNN are:

2. It requires more computing power and huge amounts of training data.

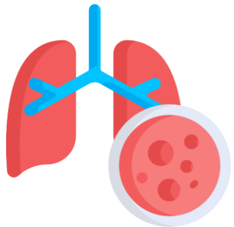
1. It does not encode the position and orientation of the object.

3. It has high computational requirements.

CNN Applications

CNN is largely used in computer vision applications such as:

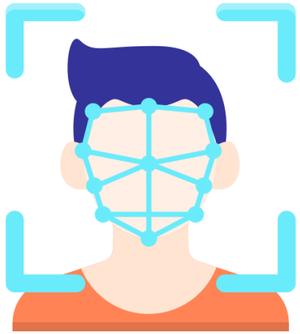
Detection of tools in a factory, where a CNN model can be trained to detect misplaced tools by factory workers



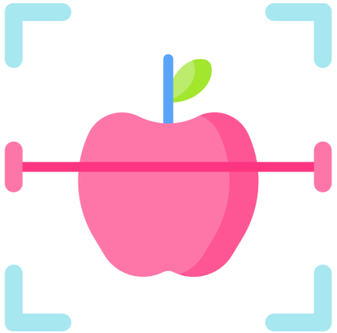
Detection of pulmonary fibrosis, where a CNN model can be fed a large dataset of a patient's lung images to identify any scarring in lung tissues

CNN Applications

CNN is a popular algorithm used widely in the field of computer vision for the following applications:



Face recognition



Object detection

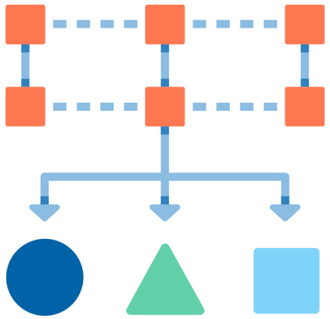


Image classification tasks

It uses a set of filters to extract features from the image.

Uses of CNN

Consider an example of a dataset containing 60,000 images, each with dimensions (28,28,3) representing height, width, and color channels, respectively

If these images are to be processed by a feed-forward neural network (FFN), each image must first be flattened.

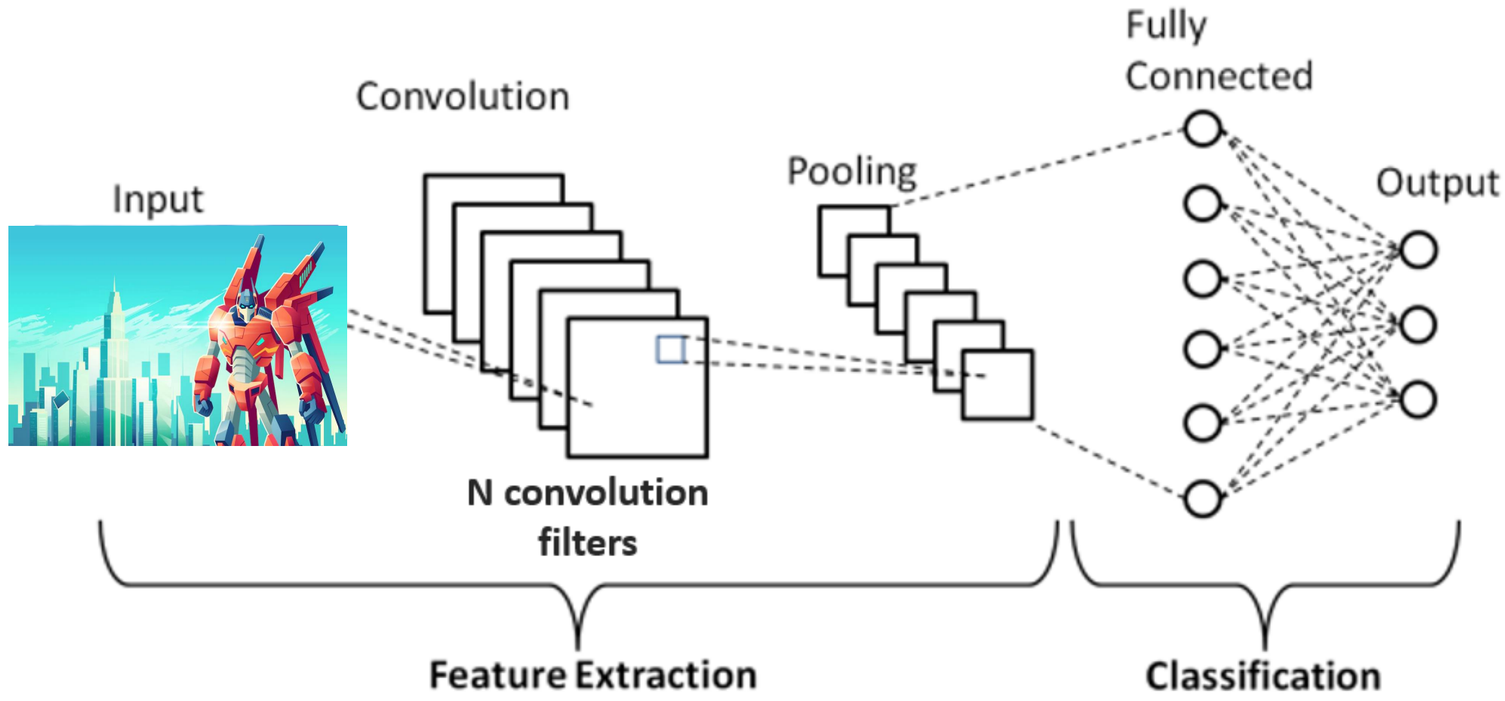
The shape of each image after flattening will be 2352.

When images have dimensions (1000, 1000, 3) representing height, width, and color channels (RGB), a feed-forward neural network (FFN) requires additional processing power for computations.

CNN was introduced to avoid such issues.

Uses of CNN

It extracts features from the images and converts them into lower dimensions without losing their characteristics.



Uses of CNN

Consider the image:

The initial image size is (400, 400, 3), and without convolution,

400 x 400 x 3 = 4,80,000 neurons will be needed in the input layer.

After applying convolution, the input tensor dimension is reduced to 1 x 1 x 3. Therefore, only three neurons are needed in the first layer of the FFN.

Getting Started with Image Data

Image Data

An image consists of three dimensions: height, width, and channel

For example:

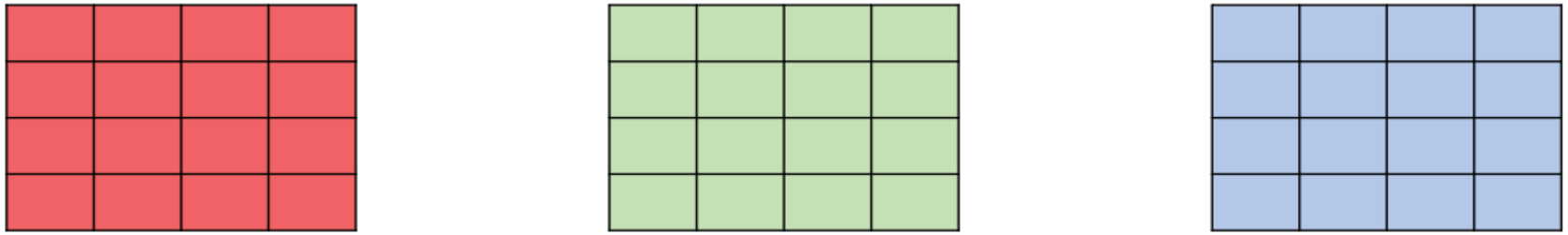
(400, 400, 3) denotes the shape of the image.

(400, 400) denotes the height and width of the image.

3 denotes the three channels in the image.

Image Data

The three channels, R, G, and B, represent the color of the image in a combination of red, green, and blue as shown below:



R G B

For each channel, the values range from 0 to 255, where 0 represents the absence of that color and 255 represents the maximum intensity of that color.

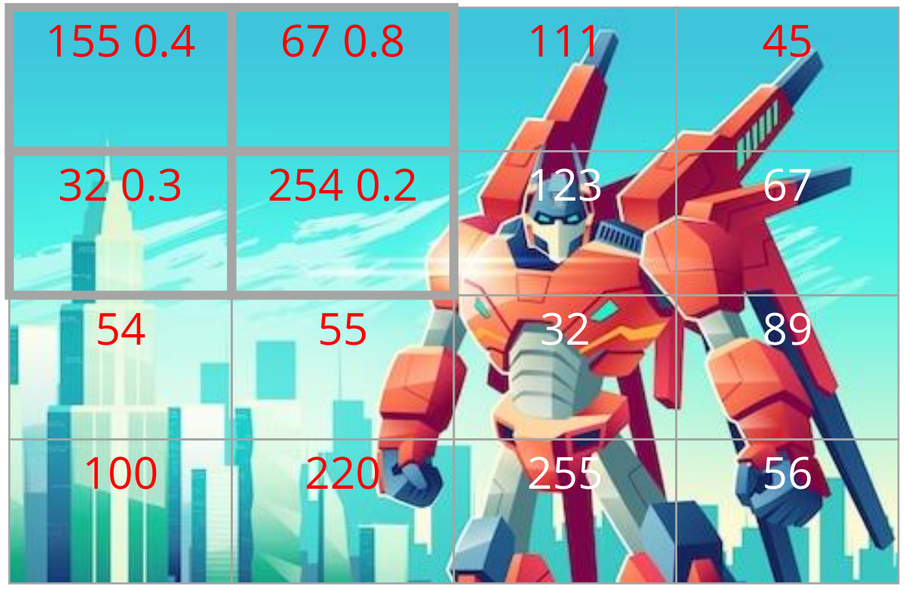
CNN with Image Data

In a CNN operation on the image data, convolution performs robust feature extraction, identifying and isolating various characteristics like edges, textures, and shapes within the image.

CNN extracts all the necessary information that is helpful to train models.

Convolution Operation

Convolution is a mathematical operation used in CNNs to extract features from input images. It involves sliding a filter (or kernel) over the image and computing the dot product of the filter with the local regions of the image.



A filter (or kernel) is a small matrix used to apply effects such as blurring, sharpening, and edge detection. The filter contains weights that are learned during the training process.

Convolution Operation

The following image shows a re-estimated value of pixels after convolution operation:

There is an image with shape (4,4) and a filter with shape (2,2).

The filter values are convoluted with the image pixel values, and the filter is moved across the image by one step.

After convolution, one gets the information about specific features that the filter is designed to detect.

176	216.4	113.2
144	112	167
156	132	118

Assisted Practice

Let’s understand the concept of CNN with image data using Jupyter Notebooks.

- 8.03_Working with Image Data

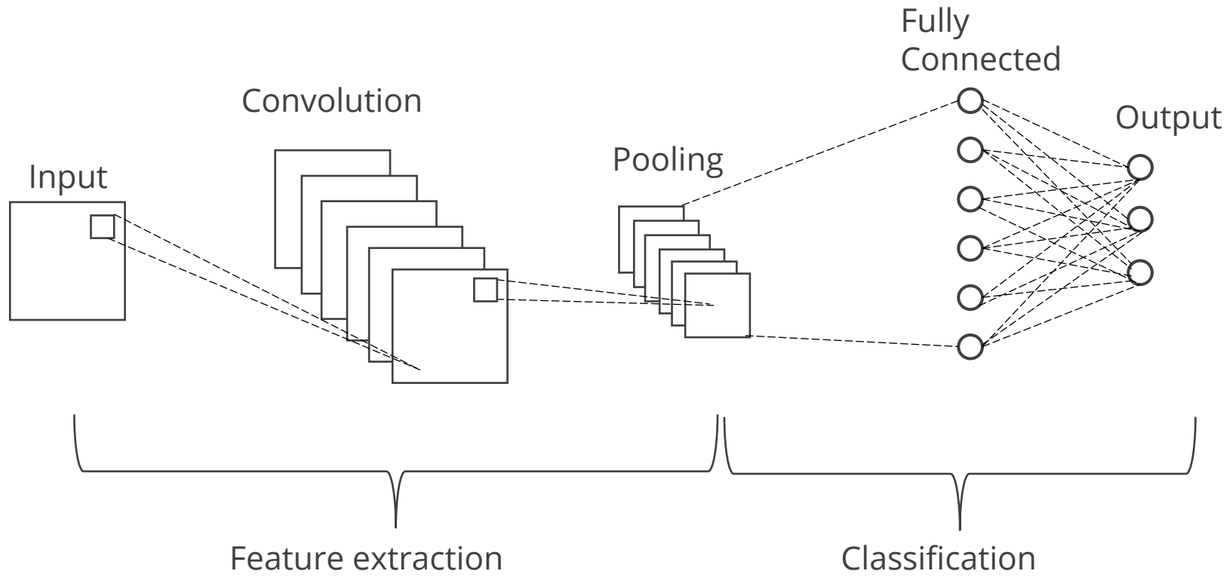
Note: Please refer to the Reference Material section to download the notebook files corresponding to each mentioned topic

ASSISTED PRACTICE

CNN Architecture

CNN Architecture

A CNN combines a backpropagation algorithm with multiple layers, including convolution, pooling, and fully connected layers.



It aims to automatically and adaptively learn spatial hierarchies of input data.

CNN Architecture

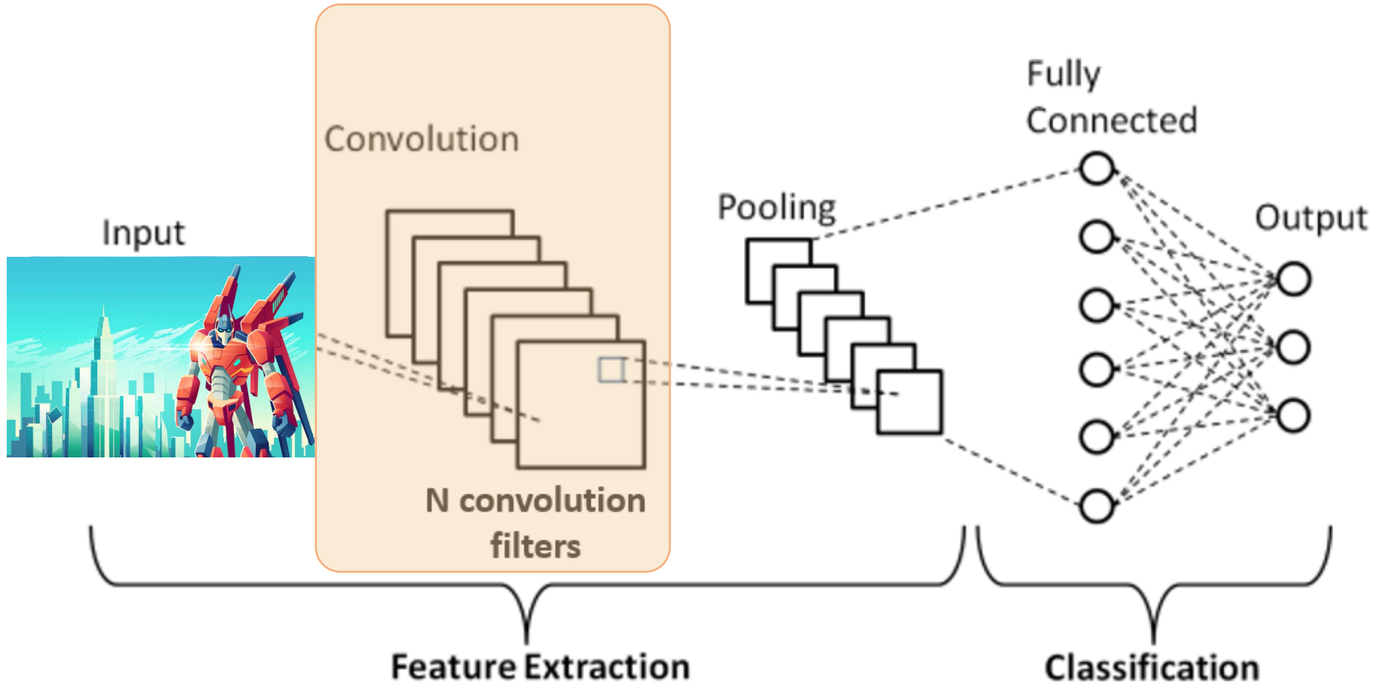
The basic architecture layers in a typical convolutional neural network are:

- Convolution layer
- Pooling layer
- Fully connected layer
- Activation function
- Output layer

The parameters used are padding and strides.

Convolution Layer

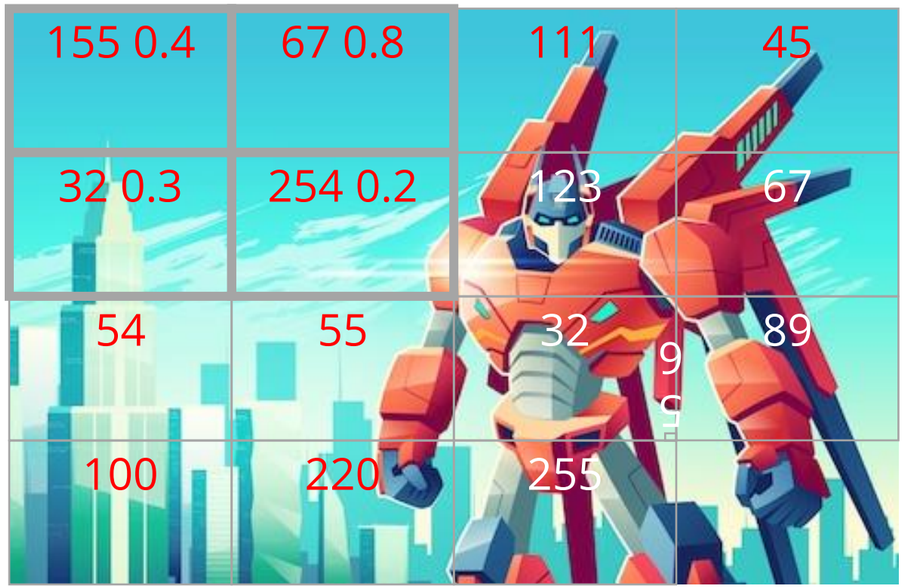
This layer is responsible for performing convolution operations on the given image and filter.



The convolution operation is the sum of the product of the filter values with pixel values.

Convolution Layer

The convolution operation is performed on each patch of the image using a filter, and a feature matrix (feature map) is created.



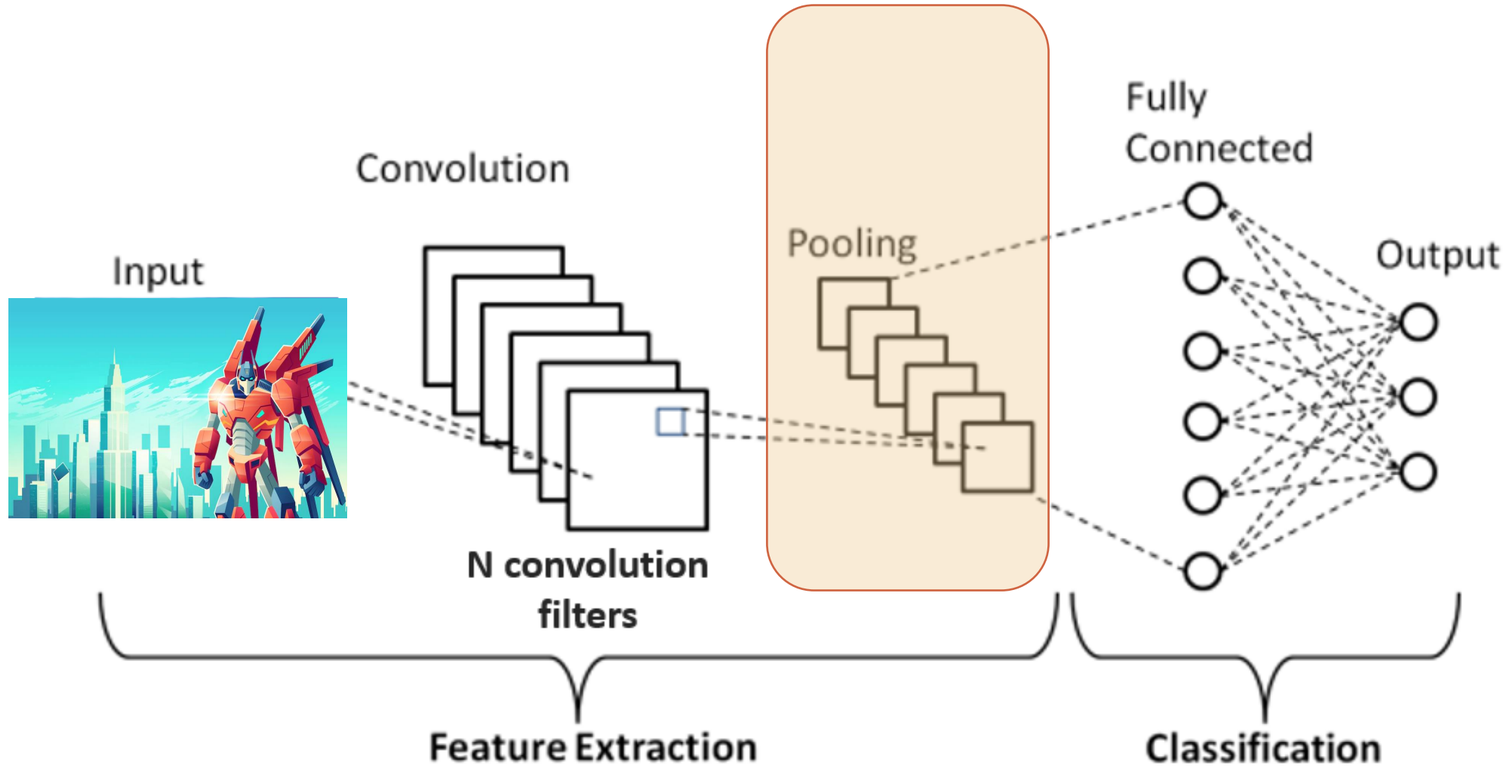
Convolution operation
on the input image

176	216.4	113.2
144	112	167
156	132	118

Convolution output:
Feature map

Pooling Layer

Pooling layer, also called as downsampling layer, reduces the size of feature maps, which in turn makes computation faster.



Pooling Layer

Some commonly used pooling operations are:

Maximum pooling

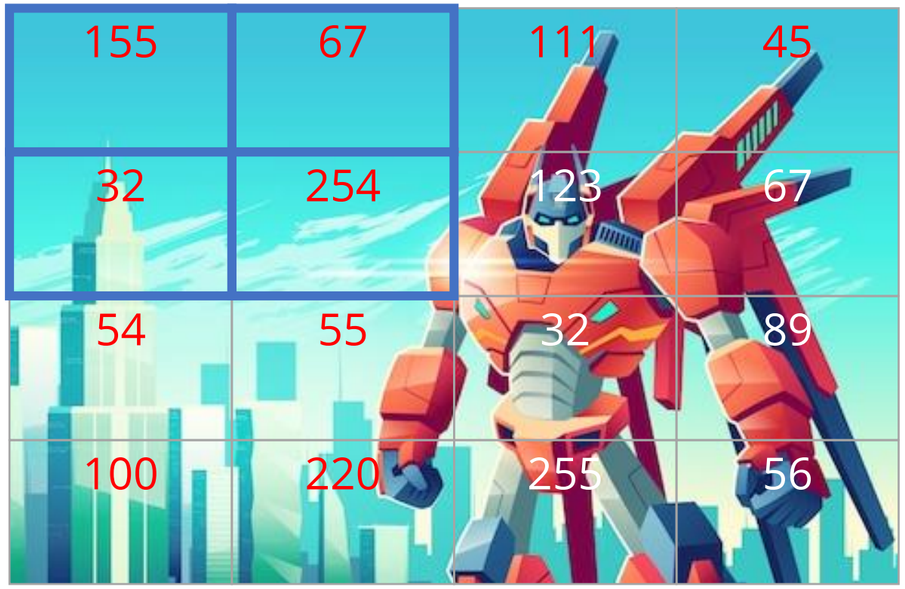
Calculates the maximum value for each patch of the feature map

Average pooling

Calculates the average value for each patch of the feature map

Maximum Pooling

In the following image, a max-pooling window of shape (2,2) is used to calculate the maximum value in each (2,2) patch of the image.



Max-pool operation on the input image

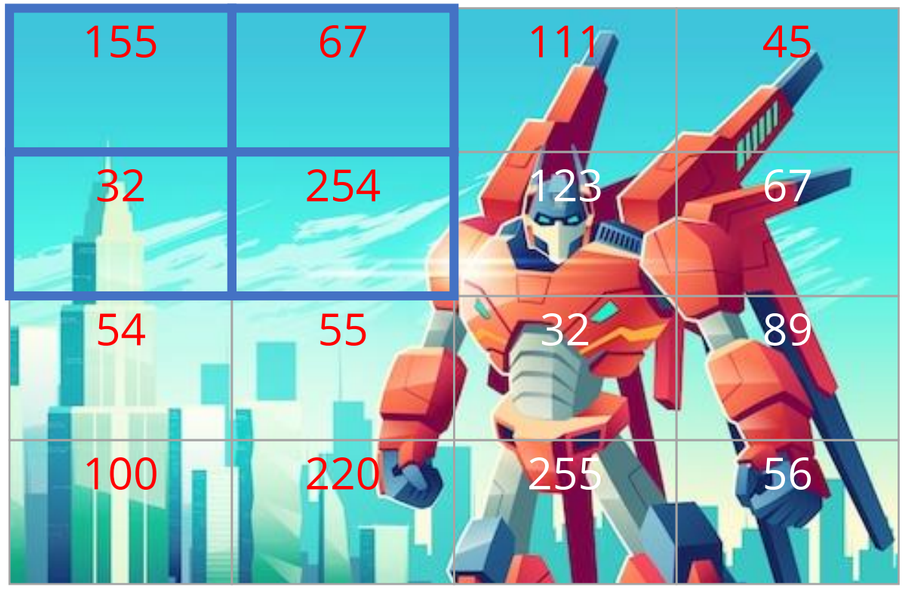
155	254	123
254	254	123
220	255	255

Max-pool output

Max pooling is predominantly used for images with dark backgrounds, as it tends to select the brighter pixels.

Average Pooling

In the following image, an average-pooling window of shape (2,2) is used to calculate the average value in each (2,2) patch of the image.



Max-pool operation on the input image

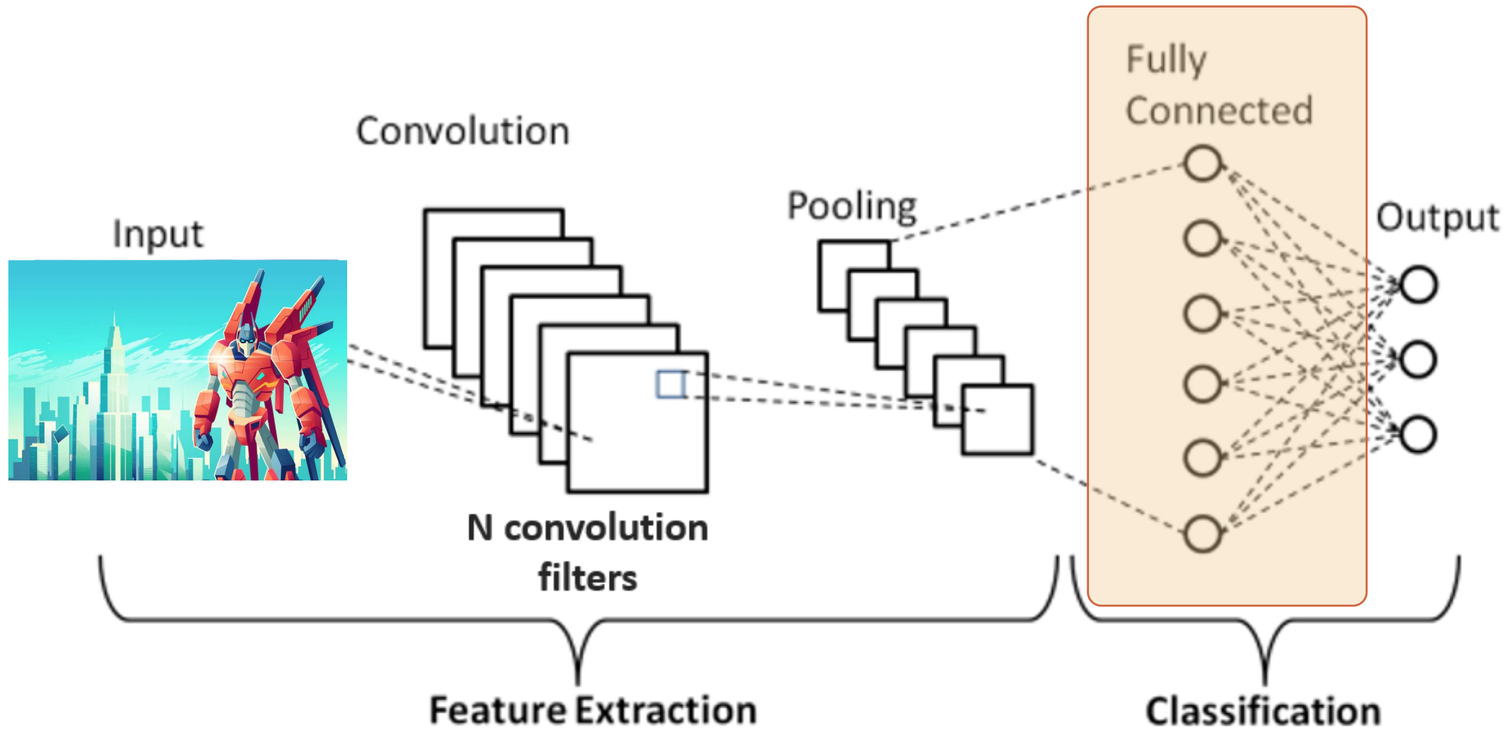
127	138.75	86.5
98.75	114.25	67.75
107.25	138.75	147.75

Average-pool output

Average pooling smooths the harsh edges of a picture and is used when such edges are not important.

Fully Connected Layer

It is a feed-forward neural network and forms the last few layers in the network.

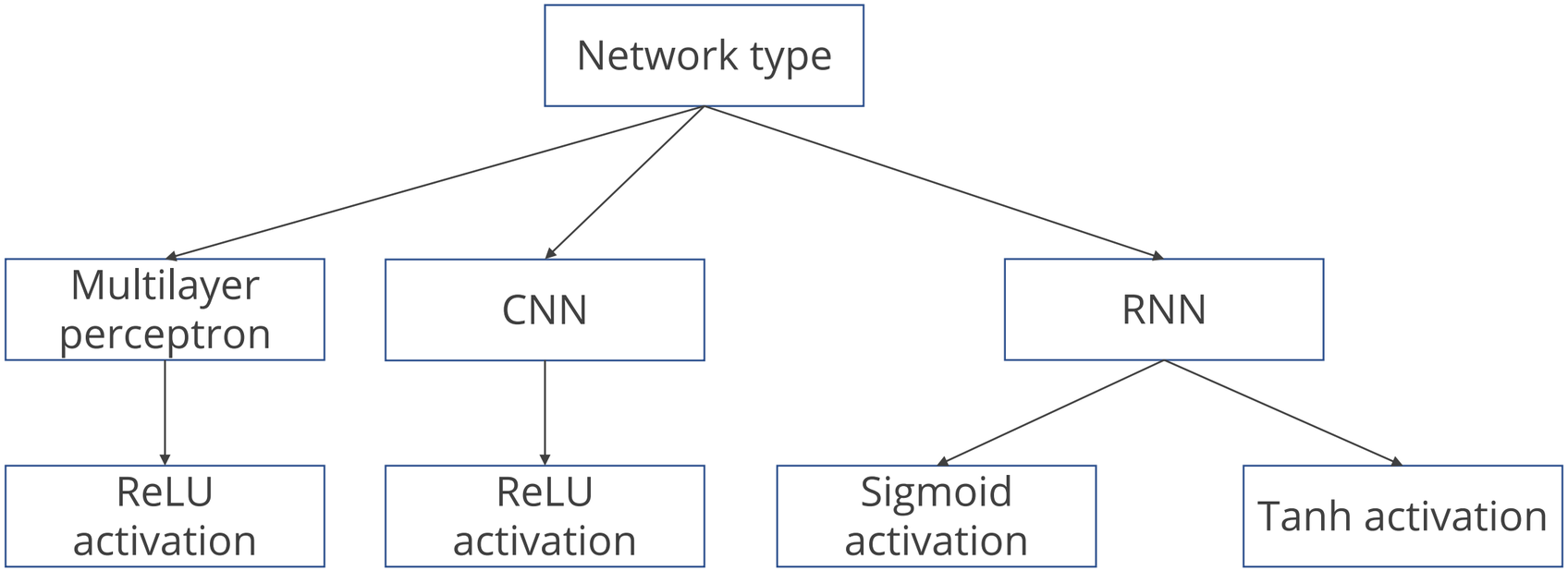


Output from the final pooling or convolutional layer is flattened and then fed into the fully connected layer

Activation Function

The activation function calculates the weighted sum and adds bias to decide if a neuron should be activated.

There are several commonly used activation functions, such as:



It introduces nonlinearity into the neuron's output to perform more complex tasks.

Output Layer

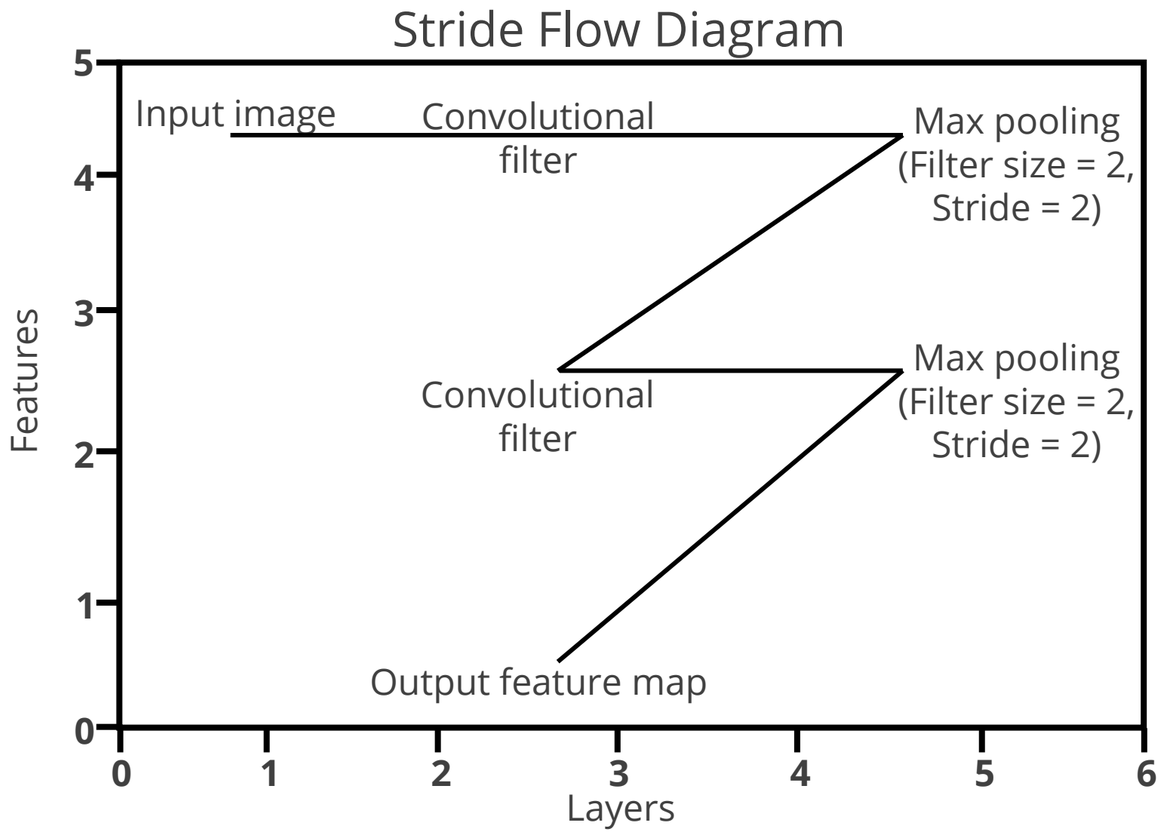
The output layer in a CNN is responsible for producing the final predictions or classifications based on the extracted features from the previous layers.

It consists of one or more fully connected layers, followed by an activation function such as softmax for classification tasks.

The output layer's weights are learned through backpropagation during training to minimize loss and improve prediction accuracy.

CNN Architecture Parameters: Strides

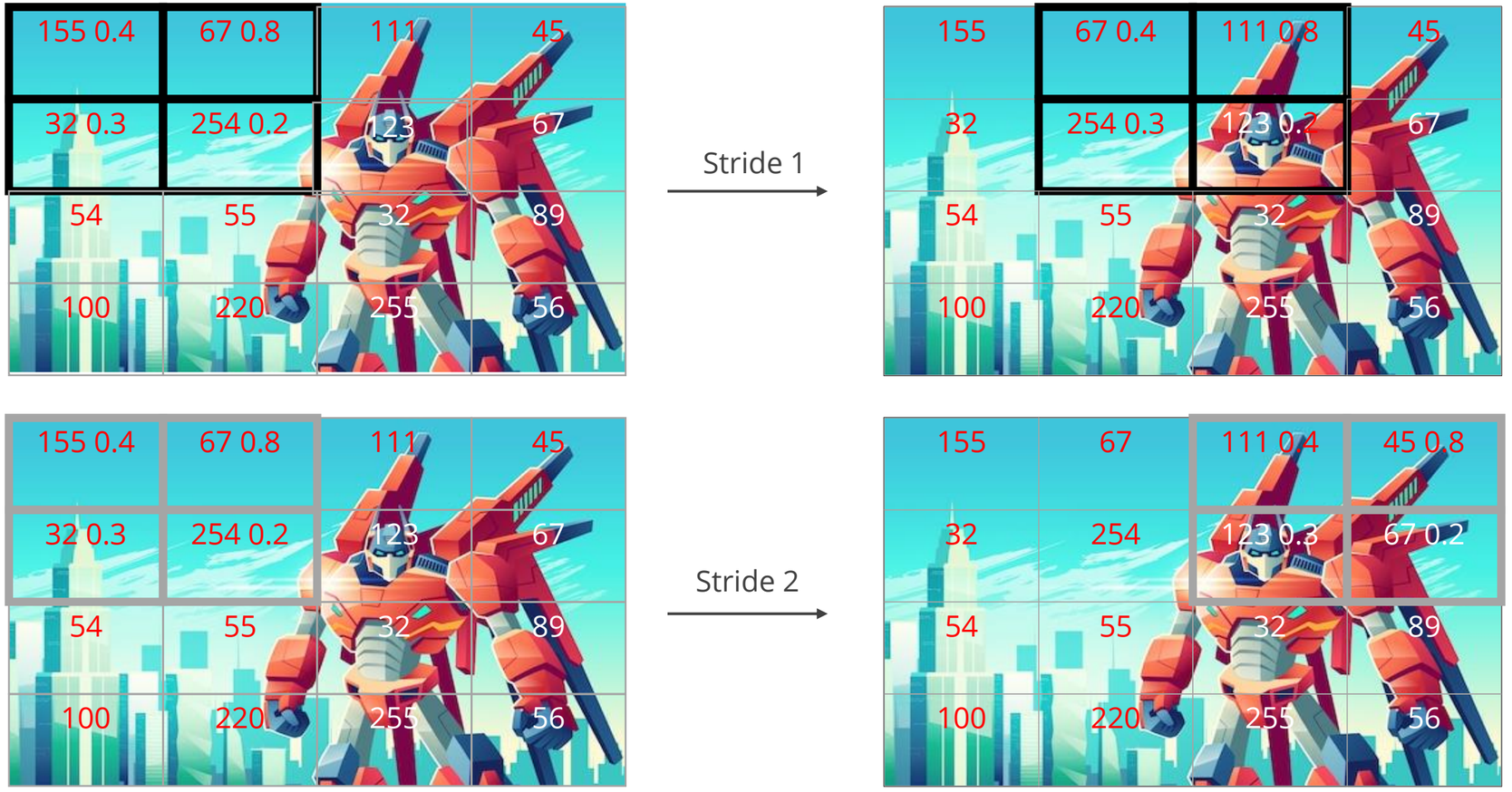
The movement of the processing window is controlled by the stride in CNN operations like convolution and max-pooling.



It denotes the number of pixel shifts across the input matrix. This affects the output size and may have an impact on the computational effectiveness and amount of detail in the generated feature maps.

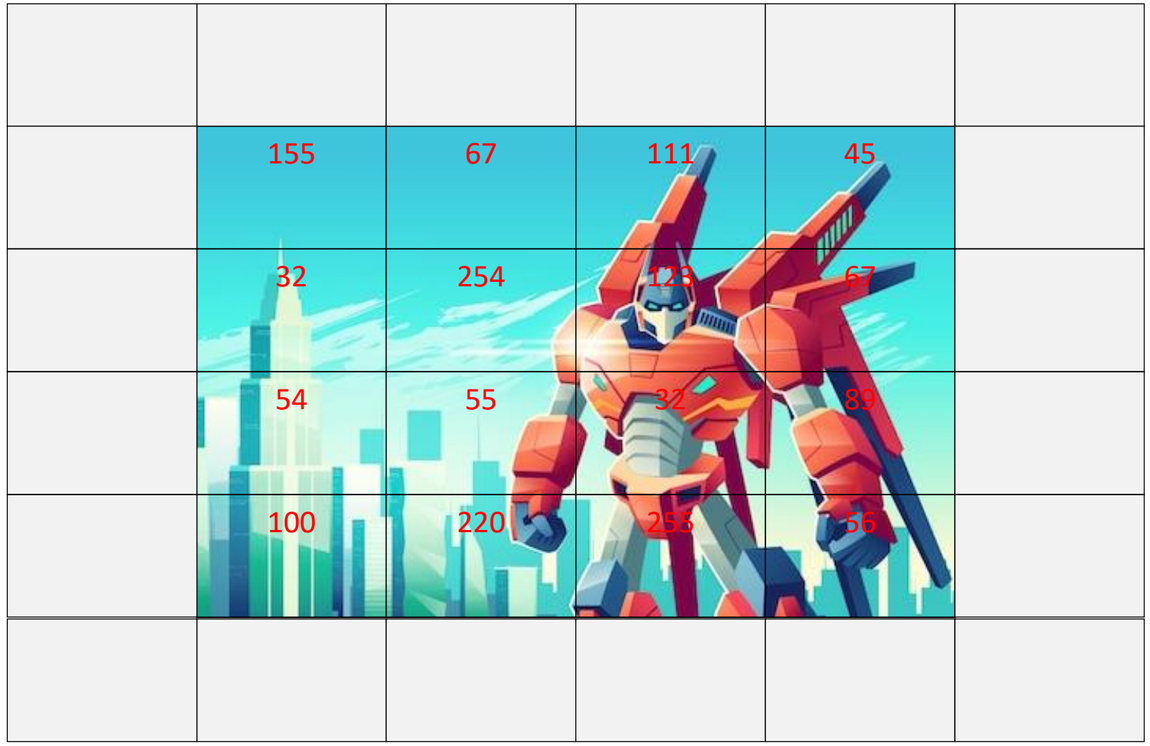
CNN Architecture Parameters: Strides

For example, if the stride is set to 1, then the kernel moves horizontally and vertically by one pixel.



CNN Architecture Parameters: Padding

It adds zeros to the input matrix symmetrically to make the shape of the output matrix the same as the input.



The gray area denotes the values padded with zero.

It reduces image shrinkage and increases image analysis accuracy.

ResNet

ResNet

Residual neural network (ResNet) is a convolutional neural network architecture widely used in computer vision tasks.

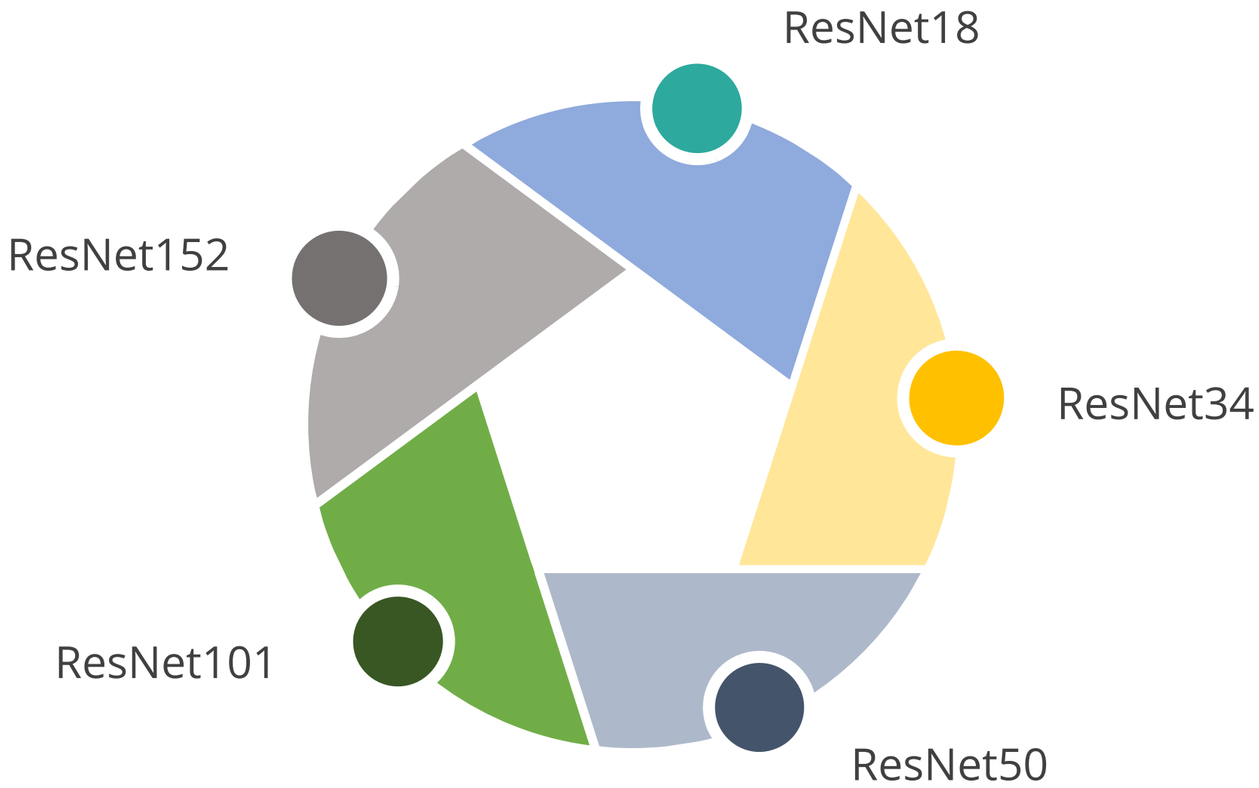
Residual connections, or skip connections, in ResNet facilitate direct pathways for gradient flow during training, helping to preserve the gradient through many layers and combat the vanishing gradient problem.

Residual connections copy the learned representations from a shallower model and add additional layers to establish identity mapping.

It supports the construction of neural networks with thousands of convolutional layers.

ResNet

ResNet has many variants such as:



The numbers represent the total number of layers in the neural network.

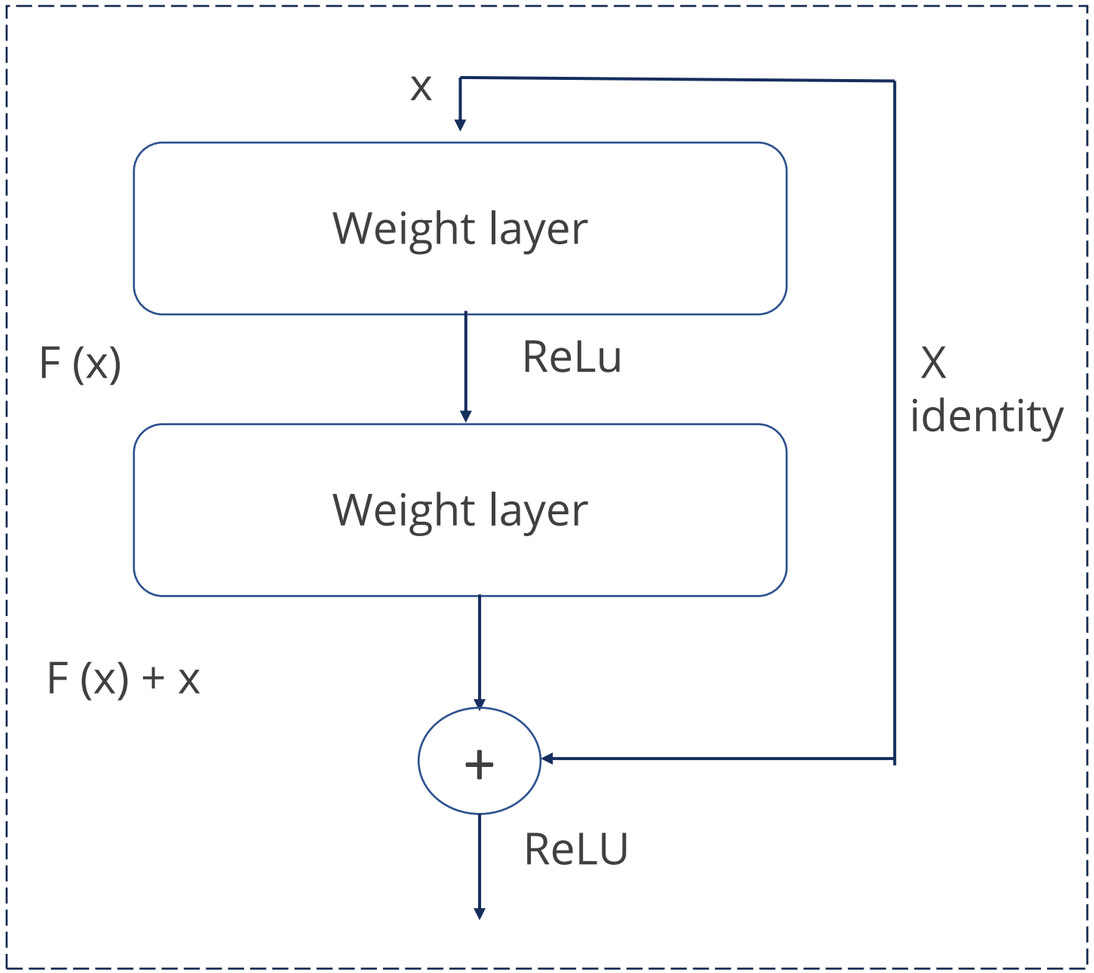
ResNet50

It has 50 layers, which include 48 convolutional layers, one max-pool layer, and one average pool layer.

It won the ILSVRC image classification challenge in 2015.

ResNet50

The layers are divided into 5 blocks, each containing a set of residual blocks. These blocks allow for the preservation of information from earlier layers, which helps the network to learn better representations of the input data.



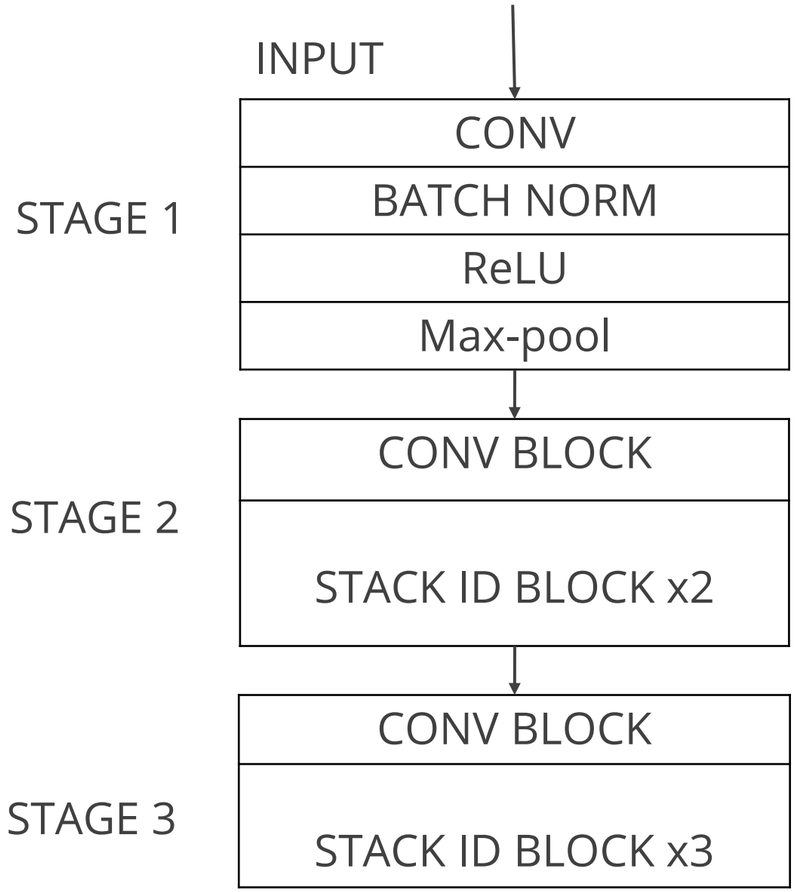
Residual Learning Block

Skip connection, or identity connection, makes it easier for the model to learn complicated patterns and solve the vanishing gradient problem in deep networks.

Skip connections are a fundamental component of residual networks.

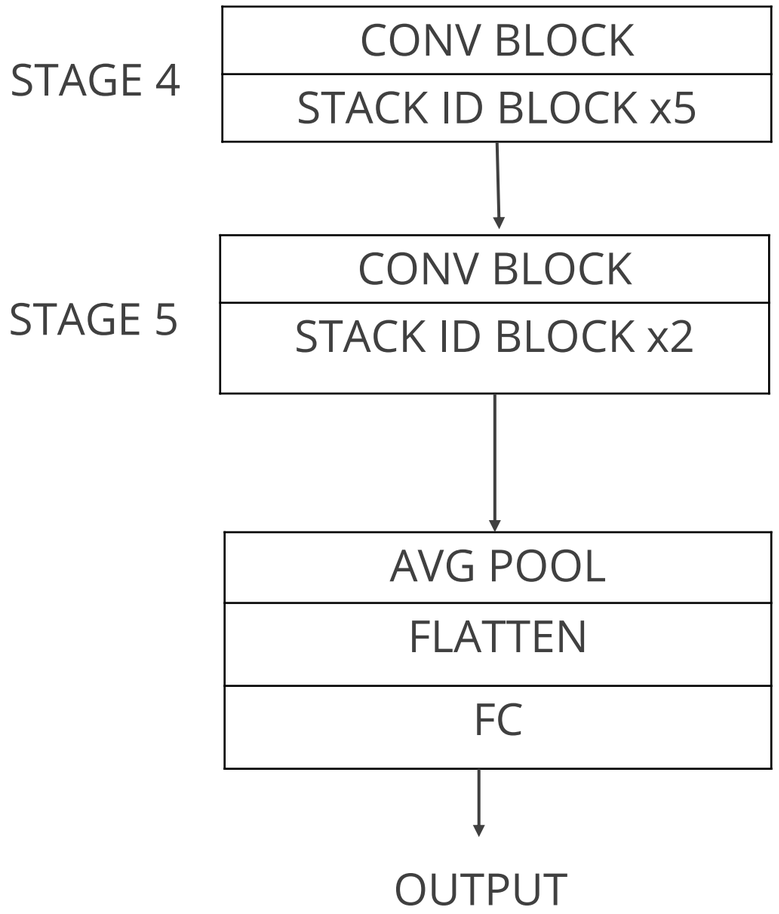
ResNet50 Architecture

In the ResNet50 architecture, the residual blocks are stacked to improve representation power in further layers.



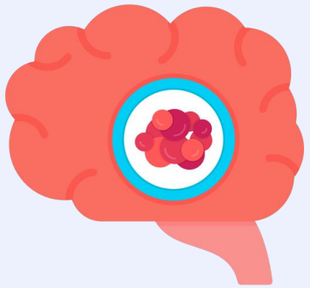
ResNet50 Architecture

Every residual block has three 3x3 convolution layers.

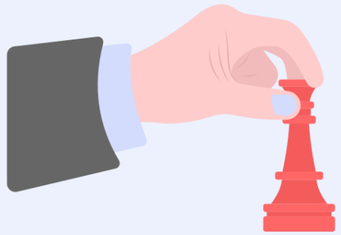


Use Cases of ResNet

Due to its large number of layers and residual connections, ResNet solves almost any computer vision problem with ease.



Detects brain tumors based on patients' brain MRI scan images

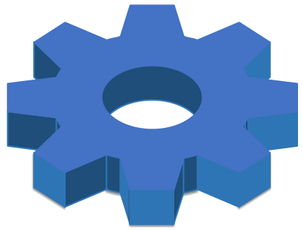


Recognizes player activities in games to produce equally challenging bots

Recognizes human emotions to understand their behaviour

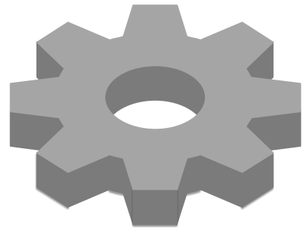
Commonly Used CNN Architectures

Other well-known architectures of convolutional neural networks (CNNs) are as follows:



VGG16

Alex Net



GoogLeNet or Inception

Filters in CNN

Filters in CNN

Filters (kernels) detect spatial patterns or features in an image, such as edges, arches, and diagonals, by detecting changes in the intensity values of an image.

The convolution operation is the sum of the products of the filter matrix and the pixel values of the image.

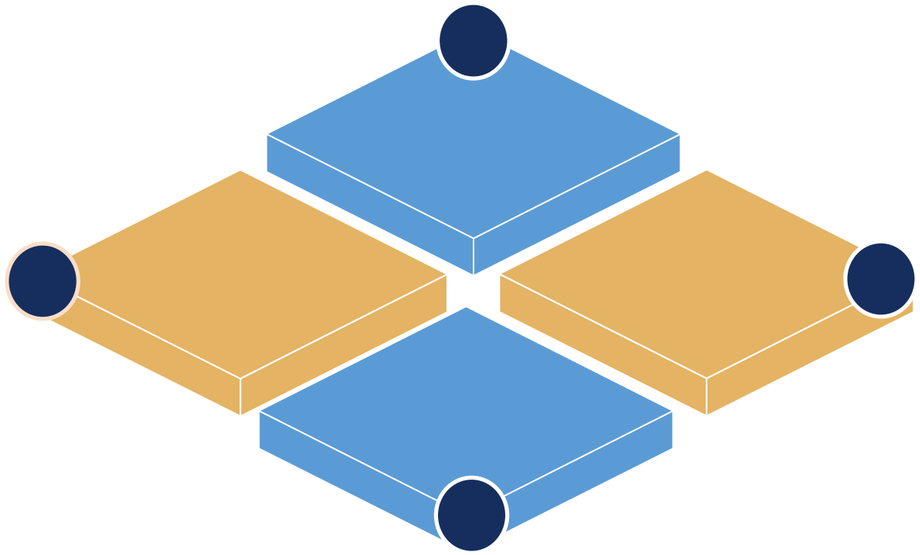
The type of filter used affects the output produced after convolution.

Filters in CNN

The different types of CNN filters are:

Horizontal Sobel filter

Outline filter



Vertical Sobel filter

Blur filter

Horizontal Sobel Filter

This filter is primarily used to detect horizontal edges in images. It emphasizes horizontal lines by detecting changes in pixel intensity in the horizontal direction.

Typical representation of horizontal Sobel filter matrix is:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

This matrix is designed to detect horizontal edges in an image by emphasizing horizontal gradients.

Horizontal Sobel Filter

Consider the image and the output shown to understand its working

Horizontal Sobel edge detection →

Vertical Sobel Filter

This filter is used to detect vertical edges in images by highlighting vertical gradients.

Typical representation of vertical Sobel filter matrix is:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Vertical Sobel Filter

Consider the image and the output shown to understand its working

Vertical Sobel edge detection

Blur Filter

It is responsible in smoothening the image, reducing noise and detail, which can be particularly useful in preprocessing steps to reduce model sensitivity to minor variations.

Typical representation of blur filter matrix is:

$$\begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix}$$

Blur Filter

The image shows a blurred output:

Image softening

Outline Filter

It is responsible for detecting the outline of objects in an image.

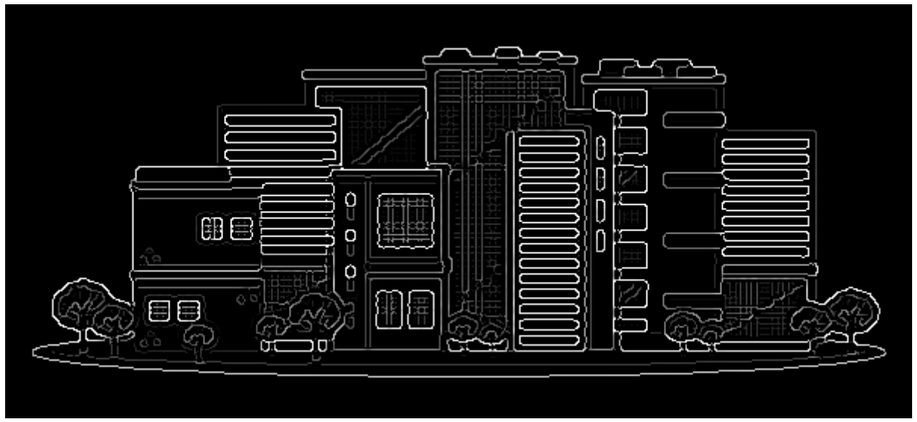
Typical representation of outline filter matrix is:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Outline Filter

The image shows the output with an outline filter.

Boundary extraction



Working of CNNs

Working of CNNs

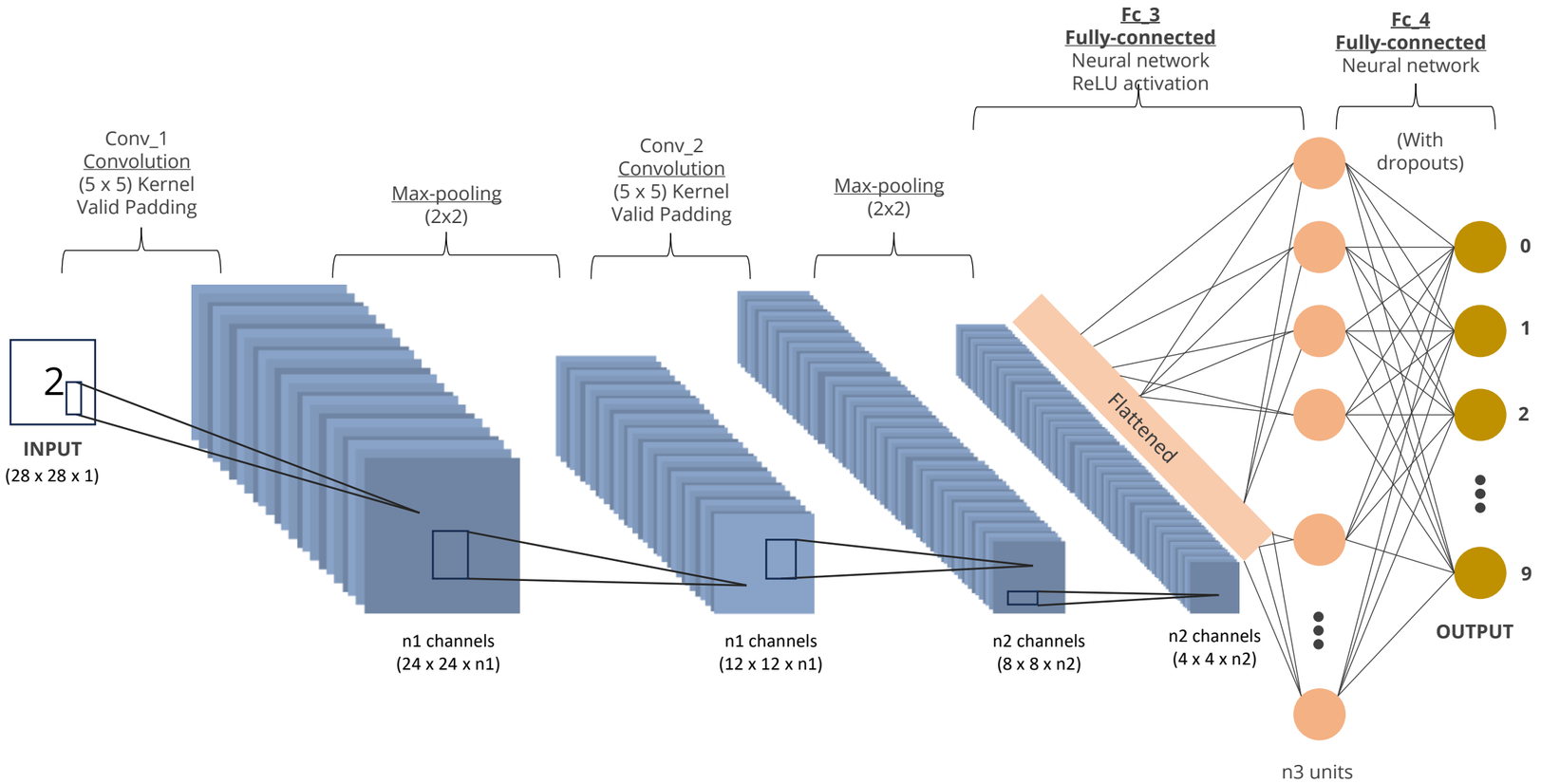
CNNs can provide more abstract and comprehensive information by stacking convolutional layers atop each other.

They perform hierarchical feature learning, like the human brain’s image recognition.

The problems to be solved and the features to be learned determine the convolutions to be applied.

Working of CNNs

In a CNN, each filter's optimum value is acquired during training and not explicitly defined.



Humans can recognize and extract meaning from images by utilizing learned filters in the visual cortex.

2D Convolution layer

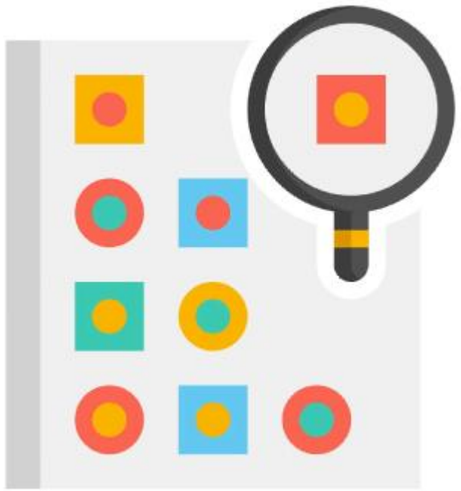
A **2D Convolution Layer**, often referred to as Conv2D, is a fundamental building block in convolutional neural networks (CNNs). It is primarily used for processing visual data, such as images or videos.

It uses filter or kernel which has a height and a width, often smaller than the input image, sliding over the entire image.

The receptive field is the area of the image where the filter is applied, determining the region of the input image that each neuron in the layer looks at.

2D Convolution layer

The conv2D filters extend across the three-color channels (RGB), with different filters for each channel.



- The individual channel convolutions are combined to produce the final image.
- The filters undergo random initialization and distribution to ensure they learn differently.
- Eventually, the filters learn to detect various aspects of the image.

2D Convolution layer

Multiple conv2D filters are used in a single layer to recognize distinct features, as each filter learns a different feature.

Each filter serves as an input to the neural network's next layer. The output from each filter is known as a feature map. When multiple filters are used, each filter produces its own feature map.

Constraints

Though highly accurate, the conv2D layer has some drawbacks.

It is computationally expensive.

A large conv2D filter compilation is time-consuming, and stacking multiple filters in layers increases the number of calculations.

Constraints

To overcome the constraints, the filter size can be reduced and the strides can be increased.

However, the filter's effective receptive field and the quantity of data it can capture are reduced.

Assisted Practice

Let’s understand the concept of CNN for image classification using Jupyter Notebooks.

- 8.08_Image classification using CNN

Note: Please refer to the Reference Material section to download the notebook files corresponding to each mentioned topic

ASSISTED PRACTICE

Pooling in CNN

Pooling in CNN

The pooling operation involves sliding a two-dimensional filter over each spatial dimension of the feature map, reducing its spatial size while retaining significant features.

It summarizes the features that lie within the region covered by the filter.

Pooling in CNN

For a feature map with dimensions n_h x n_w x n_c, the output dimensions obtained after pooling are:

$$\left(\left\lfloor \frac{n_h - f}{s} + 1 \right\rfloor , \left\lfloor \frac{n_w - f}{s} + 1 \right\rfloor , n_c \right)$$

Where,

n_h : Height of the feature map
n_w : Width of the feature map
n_c: Number of channels in the feature map
f: Size of the filter
s: Stride length

A common CNN model architecture contains multiple stacked convolution and pooling layers.

Use of Pooling in CNN

It reduces the number of parameters to learn and the amount of computation in the network.

It summarizes the features present in a region of the feature map generated by a convolution layer.

It performs further operations on the summarized features instead of precisely positioned features.

It exhibits increased resilience to variations in the positions of features within the input image.

Types of Pooling Layers

There are three types of pooling layers:

- Max-pooling
- Average pooling
- Global pooling

Max-pooling

It selects the maximum element from the region of the feature map covered by the filter.

2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

Max-pool

Filter: (2x2)
Stride: (2, 2)

9	7
8	6

The output of this layer is a feature map with the most prominent features of the previous feature map.

Average Pooling

It computes the average of the elements present in the region of the feature map.

2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

Average pool

Filter: (2x2)
Stride: (2, 2)

4.25	4.25
4.25	3.5

It gives the average of the features present in a patch.

Global Pooling

It reduces each channel in the feature map to a single value.

A nh x nw x nc feature map is reduced to a 1 x 1 x nc feature map, equivalent to using a filter of dimensions nh x nw.

It can either be global max-pooling or global average pooling.

Introduction to TensorBoard

TensorBoard

It offers a web-based interface enabling visualization of diverse aspects related to model performance, data exploration, and real-time monitoring of training progress.

It can display image, text, and audio data and aims to decrease the complexity of neural networks.

TensorBoard

TensorBoard offers a wide range of visualization tools, such as graphs and histograms to help understand and interpret machine learning models. These tools are used to interpret the results of:

- Loss
- Accuracies
- Other metrics from the model

TensorBoard

To understand the concept better, consider the following images with two types of concrete:

Plain

Marred

Image Source: https://data.mendeley.com/datasets/5y9wdsg2zt/2

TensorBoard

Construct a classifier to detect the two types of surfaces

The classifier classifies surfaces on construction sites to understand the withstanding capacity of buildings.

It can determine whether surfaces have been damaged because of earthquakes or natural disasters.

TensorBoard

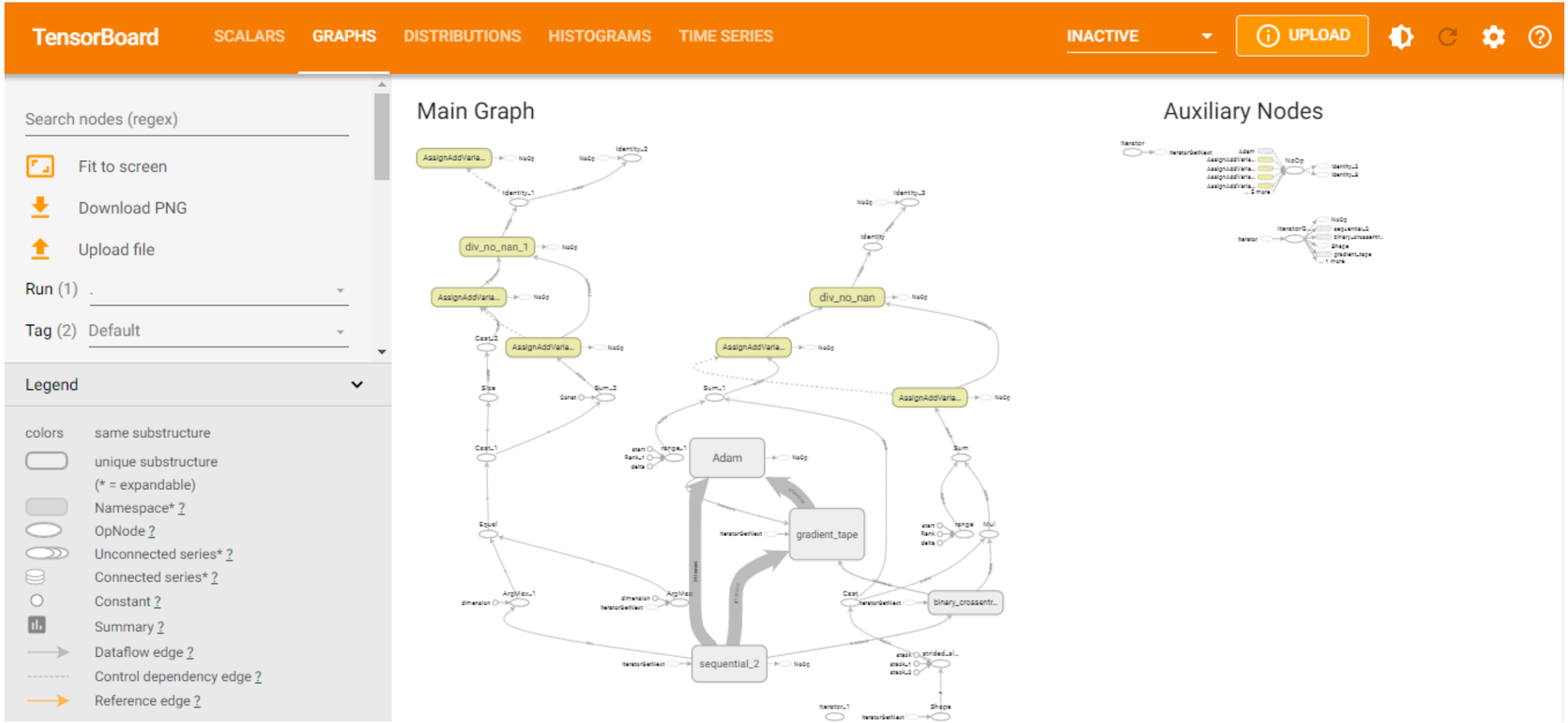
It can be initialized using the following command:

```
tensorboard --logdir path_to_logdir
```

The --logdir parameter specifies the directory where the logs of your model are saved during training.

TensorBoard

Consider the following image:



TensorBoard

The image has five sections:

Scalars	The graphs of metrics, such as training loss and accuracy, are saved.
Graphs	The model architecture is clearly portrayed.
Distributions	The weight distribution along each layer is shown.
Histograms	The histogram plots with a frequency of 1 along each layer are displayed.
Time series	The distribution along each layer is checked, along with the time.

These details help one understand and debug the machine learning model under study thoroughly.

Assisted Practice

Let’s understand the concept of introduction to TensorBoard using Jupyter Notebooks.

- 8.11_Introduction to TensorBoard

Note: Please refer to the Reference Material section to download the notebook files corresponding to each mentioned topic

Key Takeaways

- CNN is a popular algorithm that is used widely in the field of computer vision.
- The convolution operation is the sum of the product of filter values to the pixel values.
- The three essential layers in a CNN are the convolution, pooling, and fully connected layers.
- Residual neural network (ResNet) is a convolutional neural network architecture widely used in computer vision tasks.

Key Takeaways

- The different types of CNN filters are horizontal and vertical Sobel, blur, and outline filters.
- The Conv2D filters extend across the three-color channels (RGB), and the individual channel convolutions are combined to produce the concluding image.
- TensorBoard is an interface used to visualize, understand, and debug machine learning models.

Knowledge Check

Knowledge Check

1

What does the shape of an image data represent?

A. Height and depth of the image

B. Width, depth, and brightness of the image

C. Height, width, and brightness of the image

D. Height, width, and channels of the image

Knowledge Check 1

What does the shape of an image data represent?

A. Height and depth of the image

B. Width, depth, and brightness of the image

C. Height, width, and brightness of the image

D. Height, width, and channels of the image

The correct answer is **D**

The shape of an image data represents the height, width, and channels of the image, denoted as (height, width, channels).

Knowledge Check

2

What does the convolution operation do in a CNN?

A. Flattens the image data

B. Extracts all the necessary information from the image data

C. Performs an addition operation on image data

D. Subtracts the filter values from the pixel values in the image data

Knowledge Check 2

What does the convolution operation do in a CNN?

A. Flattens the image data

B. Extracts all the necessary information from the image data

C. Performs an addition operation on image data

D. Subtracts the filter values from the pixel values in the image data

The correct answer is **B**

In a CNN, the convolution operation extracts all the necessary information from the image data, which is helpful in training models.

Knowledge Check

3

What is the purpose of pooling layers in CNNs?

A. To increase the number of parameters to learn and the amount of computation in the network

B. To reduce the feature map dimensions and amount of computation in the network

C. To increase the feature map dimensions and amount of computation in the network

D. To reduce the feature map dimensions and number of layers in the network

Knowledge Check

3

What is the purpose of pooling layers in CNNs?

A. To increase the number of parameters to learn and the amount of computation in the network

B. To reduce the feature map dimensions and amount of computation in the network

C. To increase the feature map dimensions and amount of computation in the network

D. To reduce the feature map dimensions and number of layers in the network

The correct answer is **B**

Pooling layers reduce the feature map dimensions, which in turn helps reduce the number of parameters to learn and the amount of computation in the network.

Lesson-End Project: Image Classifier with CIFAR10

Problem statement: Build a deep learning convolutional neural network to recognize characters using the Chars74k dataset

Objective: Build a neural network-based classification model to recognize characters using the following metrics:
Use four convolution layers with a 3*3 kernel and activation function as ReLU.
Add maximum pooling layers after every other convolution layer and two hidden layers with dropout.

Access: Click on the **Lab** tab on the left side of the LMS panel. Copy the generated username and password. Click on the **Launch Lab** button.
On the new page, enter the username and password you copied earlier into the respective fields. Click **Login** to start your lab session. A full-fledged Jupyter lab opens, which you can use for your hands-on practice and projects.

LESSON-END PROJECT

Thank You!