The leading algorithms in today's world which are used to solve the Computer vision problems

Convolutional Neural Networks: Why, what and How!

Why do we Need Convolutional Neural Networks (CNN) ?

Rajat Katyal Follow
Nov 5, 2019 · 9 min read

The leading algorithms in today's world which are used to solve the Computer vision problems are typically Convolutional Neural Networks. So what are the types of Computer vision problems or the applications of a CNN?

what are the types of Computer vision problems or the applications of a CNN?

Classification tasks

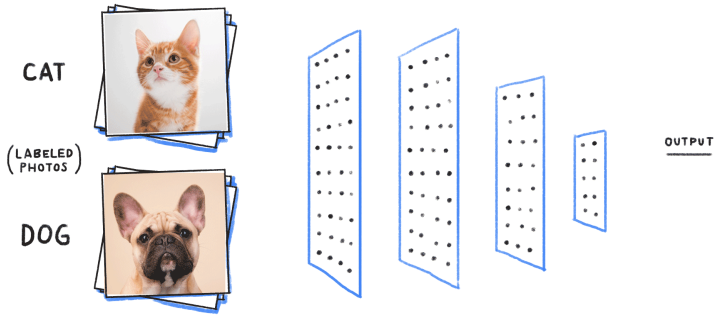
Recognition tasks

Object Detection

Pattern Detection

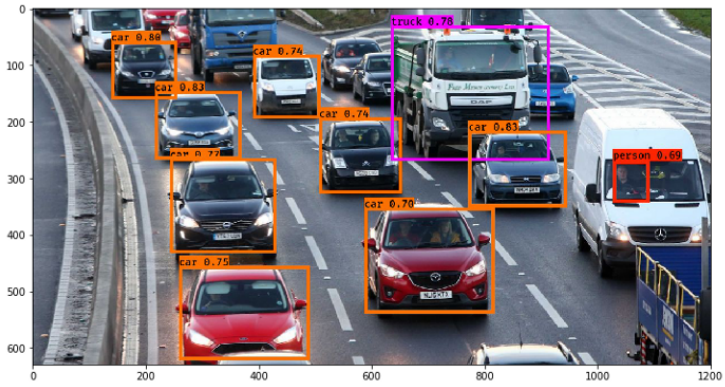
Natural Language Processing

1. **Classification tasks:** Telling the difference between a cat and a dog. Segregating 100,000s of images into specific categories.



2. **Recognition tasks:** Wonder how Facebook suggests the correct people to tag on the photos. How the smartphones use facial recognition to unlock your device.

3. **Object Detection:** Identifying objects of interest from a scene. Counting the number of cars on the highway.



4. **Pattern Detection:** Pattern detection has applications in Biology to detect anomalous or diseased cell structures. It has application in Manufacturing plants to detect faulty or damaged products.

5. **Natural Language Processing:** CNNs are used to read paper-Books or documents and classify them into Digital format.

So as you can see, the scope is pretty vast. Also given the vast amount of Video data that our society is generating, there would be a massive opportunity for Image/Video analytics based products and services in the future. Even the self-driving Teslas rely on neural-network training models.

What is a Convolutional Neural Network ?

A Convolutional Neural Network (CNN)

is a type of

artificial neural network

used in image recognition and processing

that is specifically designed to process large pixel data

Neural Networks mimic the way our nerve cells communicate with interconnected neurons and CNNs have a similar architecture

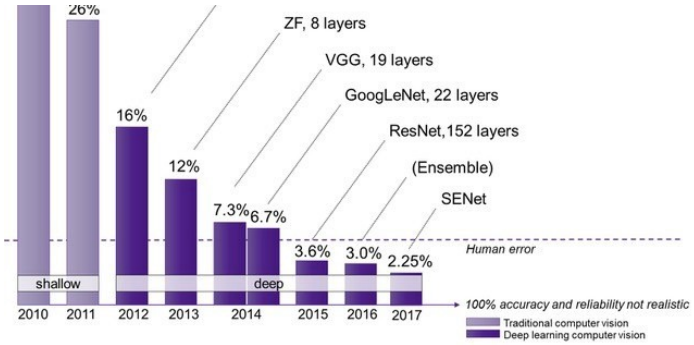
What makes CNN unique from other neural networks is

the convolutional operation that applies filters to every part of the previous input

in order to extract patterns and features maps.

CNN speed of computation of was exponentially faster than other algorithms.

LeNet-5 (by LeCunn and co. in 1997) was one of the first Convolutional Neural Networks(CNN) that was deployed in banks for reading cheques in real-time.



After this the popular events have been the development of Tensor FLow which is a free and open-source software library for dataflow developed by Google in 2015. Tensor flow is very popular in the Data Science community for having simplified the Deep Learning tasks.

There are certain core Layers in the CNN Architecture.

1. Input
2. Padding
3. Convolution + Activation/Relu
4. Pooling
5. Flatten/Dense
6. Fully Connected + Softmax

Layer 1 : Input Layer :

The input data is typically preprocessed to a multi-dimensional array format

Layer 2 : Padding Layer:

A padding layer is typically added to ensure that the outer boundaries of the input layer doesn't lose its features when the convolution operation is applied.

It is also done to adjust the size of the input. So in most cases a Zero Padding is applied, i.e. just adding a black space on the boundaries

Layer 3 : Convolution Layer:

This is an interesting operation that works by taking a 'Feature map' or say a filter and applying it on every part of the input image.

Here applying means doing an arithmetic operation explained below.

The result of the operation is stored as a value for the next layer.

This operation is repeated across the input image by Convolving the Filter

Why do we Convolve Input image with filter ?

Convolution helps in detecting features among the input image.
The 'Feature Map' (Filter) matrix with different values can detect different features

A Convolution layer can have a number of Feature Maps(or Filters) in each layer and so can produce as many outputs.

Activation function (Relu):

After applying the convolutional function, a non-linearity is added to the output.

Typically, this is done by the Rectified Linear Unit (Relu) Activation function

There are other forms of Activation functions as well like Sigmoid, Tanh and Softmax

Layer 4 : Pooling Layer or Down Sampling Layer :

Pooling is an operation which has two main impacts:

1. It reduces the dimensions of the feature maps, so lesser parameters are faster to compute in following layers.

Hence, it is also known as a down-sampling layer.

2. It highlights the importance of the features.

There are a few pooling operations which are popular:

Average pooling

Max pooling and

Sum Pooling.

Out of these max pooling is most widely used

Stride:

Stride is the way the convolution works in both convolution and pooling layers.

Notice that in the Convolution layer the filters moved Only 1 step to the right and 1 step to the bottom. That is because the stride was 1 in the convolution step.

In the Pooling step, the 2x2 filter moved 2 steps to the right and 2 steps to the bottom. This is because our stride was 2.

If in pooling were to apply the 2x2 filter on the 4x4 input with stride=1, then the Pooled feature map would have had a 3x3 dimension

output_with_stride = floor((input + 2*padding - filter) / stride) + 1

Layer 5 : Flatten/Dense:

The flatten layer basically takes the current pooling layer output

and it converts it into the format which is required for the Fully connected layer.

The fully connected layer is an artificial neural network in itself and requires a specific input.

Layer 6 : Fully Connected and Softmax (Activation):

The initial convolution layers help in detecting low level features like edges.

When we pass these again into convolutional layers, higher level features (For example: nose or ears) are detected.

The fully connected layer is the final piece of the puzzle.

It basically takes the high level feature maps(nose, ears, etc.) as the input and decides what the output category would be.

This is basically a multi-level Perceptron network that identifies which weights are more likely to contribute to the which outputs.

This is done when we train the model with a lot of images, it is able to decide which attributes associate more with which categories.

It is fully connected as every neuron in the previous layer is connected to every neuron in the next layer.

The fully connected layer has a Softmax Activation function at the end which ensures that the sum of output probabilities from the Fully Connected Layer is 1.

This is the final step of the classifier and decides the output.

Alright then!

You have learnt how a Convolutional Neural Network works! Of course this was just the basic overview of CNNs. If you are interested in knowing more about the actual implementation of a CNN, I could recomend a couple of resources:

1. CNN for Computer Vision by Stanford
2. DeepLearnign.AI

In terms of **Frameworks** to actually build a CNN, there are 3 very popular ones and you can use them in Python among other languages:

1. TensorFlow (by google)
2. Keras
3. Pytorch (by facebook)

Personal Work:

I recently had a computer vision project with about 100,00 images to be categorized into 15 categories. This was a classification task and I used a

CNN model to classify the images. Although I cannot share the Images, I have shared a link to the Code below if you would like to take a look.

References:

https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8

https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/

https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#7d8a

https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

https://medium.com/@purnasaigudikandula/a-beginner-intro-to-convolutional-neural-networks-684c5620c2ce

https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-4-full-connection

. . .

WRITTEN BY

Rajat Katyal

Follow

Noteworthy - The Journal Blog

The Official Journal Blog

Follow

More From Medium

Quick Start Guide to Product Strategy

Blaine Holt in Noteworthy - The Journal Blog

The End of White American Christianity

Sara Zarr in Noteworthy - The Journal Blog

RxJS Better Practice

kelly woo in Noteworthy - The Journal Blog

You Are Not Equal. I'm Sorry.

Dina Ley in Noteworthy - The Journal Blog

A Year of Writing and Profit on Medium

Logan Daley in Noteworthy - The Journal Blog

Dad Songs (Say So Much)

Joe Shetina in Noteworthy - The Journal Blog

Impeachment and the 25th Amendment Won't Work, But There Is Something That Will

Mitchell Plitnick in Noteworthy - The Journal Blog

White People Are Broken

Katherine Fugate in Noteworthy - The Journal Blog