The Evolution of CNN Models

The Evolution of CNN Models

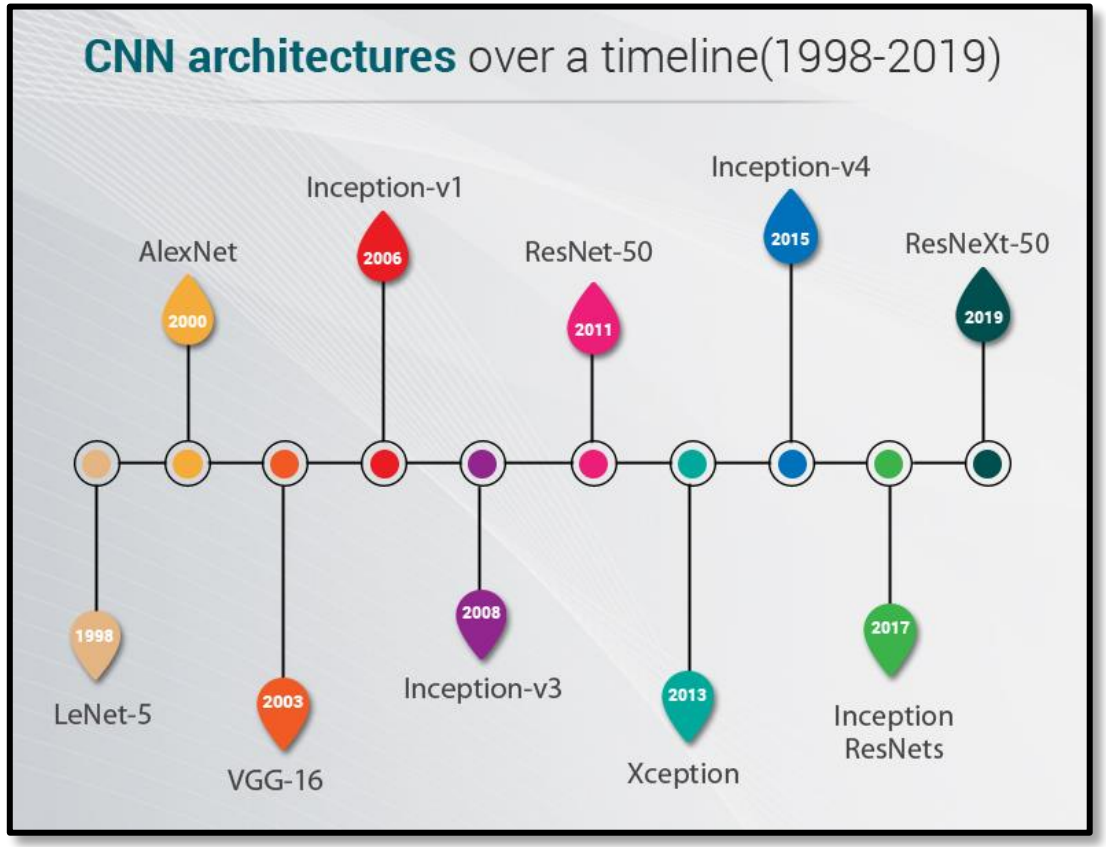
Why does the CNN Evolution Matter?

Studying the evolution of CNN models is crucial for anyone working in deep learning. Through this process, we learn how models have overcome various limitations, such as overfitting and vanishing gradients.
Additionally, understanding the advantages and limitations of each model helps us make informed decisions when selecting or designing models for specific applications. By studying the evolution of CNN models, we can develop more efficient and adaptable architectures.

The Evolution of CNN Models

Overview of CNN Evolution

CNN, like many other concepts in technology, has evolved drastically in the past few decades. The early models, such as LeNet-5, were relatively simple and could only analyze grayscale images. As we progress through time, we see increasingly complex models, such as ResNet50, which features many convolutional layers and performs exceptionally well on a wide variety of images, from medical to astronomical. In this presentation, we will examine the innovations and limitations of some of these models.

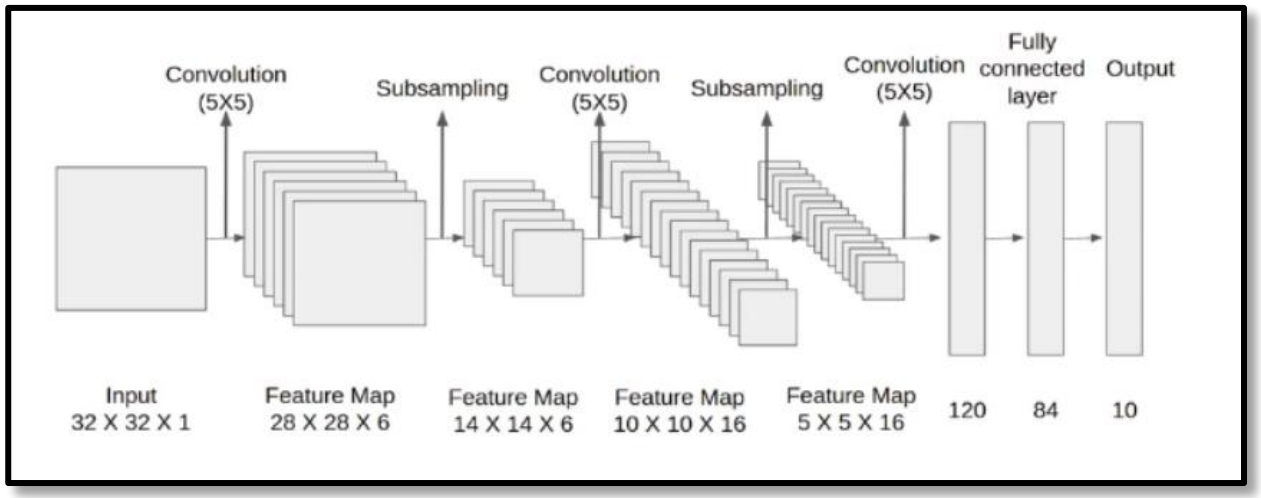


Source: www.aismartz.com

The Evolution of CNN Models

LeNet-5

Taking a look at the history of CNN models, the first popular model identified is LeNet-5. LeNet-5 is a relatively simple model, as it has three convolutional layers and two subsampling layers. Despite its simplicity, at the time, it was considered a revolution in image classification. It outperformed artificial neural network (ANN) models that were used to classify images. This model laid the foundation for future CNN architectures.

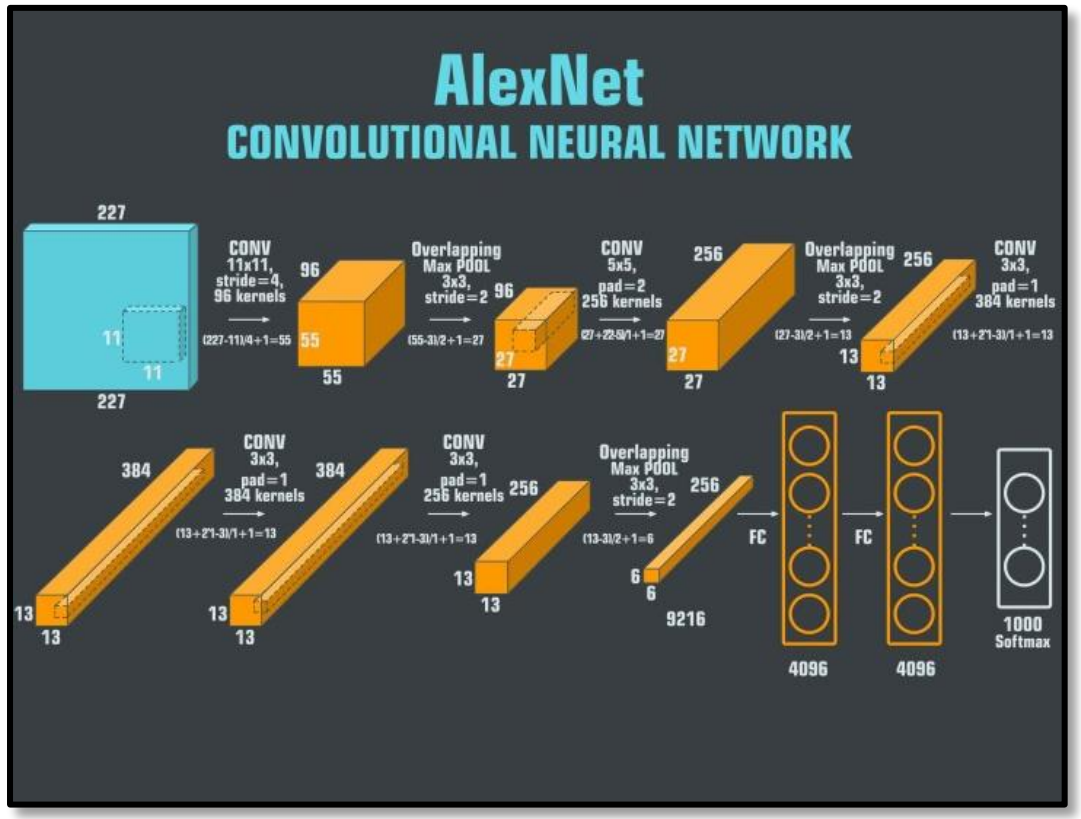


Source: www.analyticsvidhya.com

The Evolution of CNN Models

AlexNet

AlexNet, unlike LeNet, was able to classify color images. This model gained attention after winning the ImageNet Large Scale Visual Recognition Challenge in 2012.
A key innovation in AlexNet was the use of the ReLU activation function, which significantly speeds up training compared to traditional activation functions like sigmoid or tanh. This model also included dropout and normalization techniques to enhance performance and reduce overfitting.

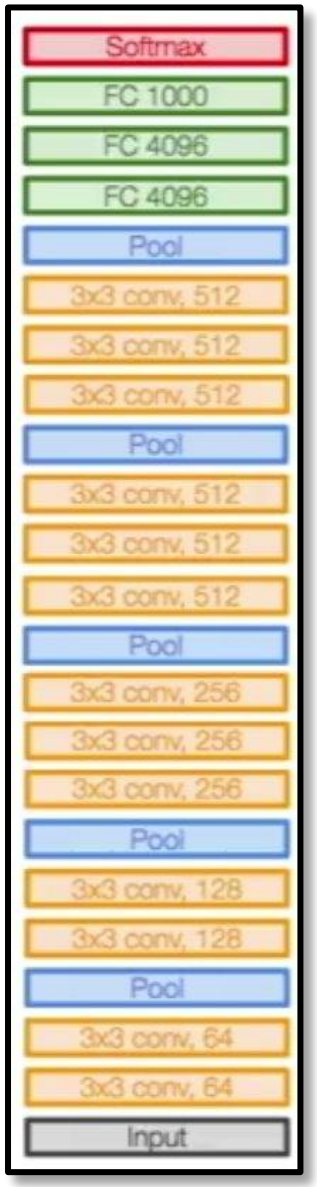


VGG16

VGG16, unlike previous models, uses small 3x3 filters. This model is among the first to contain many convolutional layers, featuring 13 convolutional layers, which is substantially more than the 5 convolutional layers in AlexNet.

This architecture enables the model to capture more complex features of images at the cost of increased trainable parameters. The increase in the number of parameters leads to higher memory consumption.

VGG16 outperformed AlexNet on the ImageNet dataset. The architecture of this model will be studied in a different lecture.

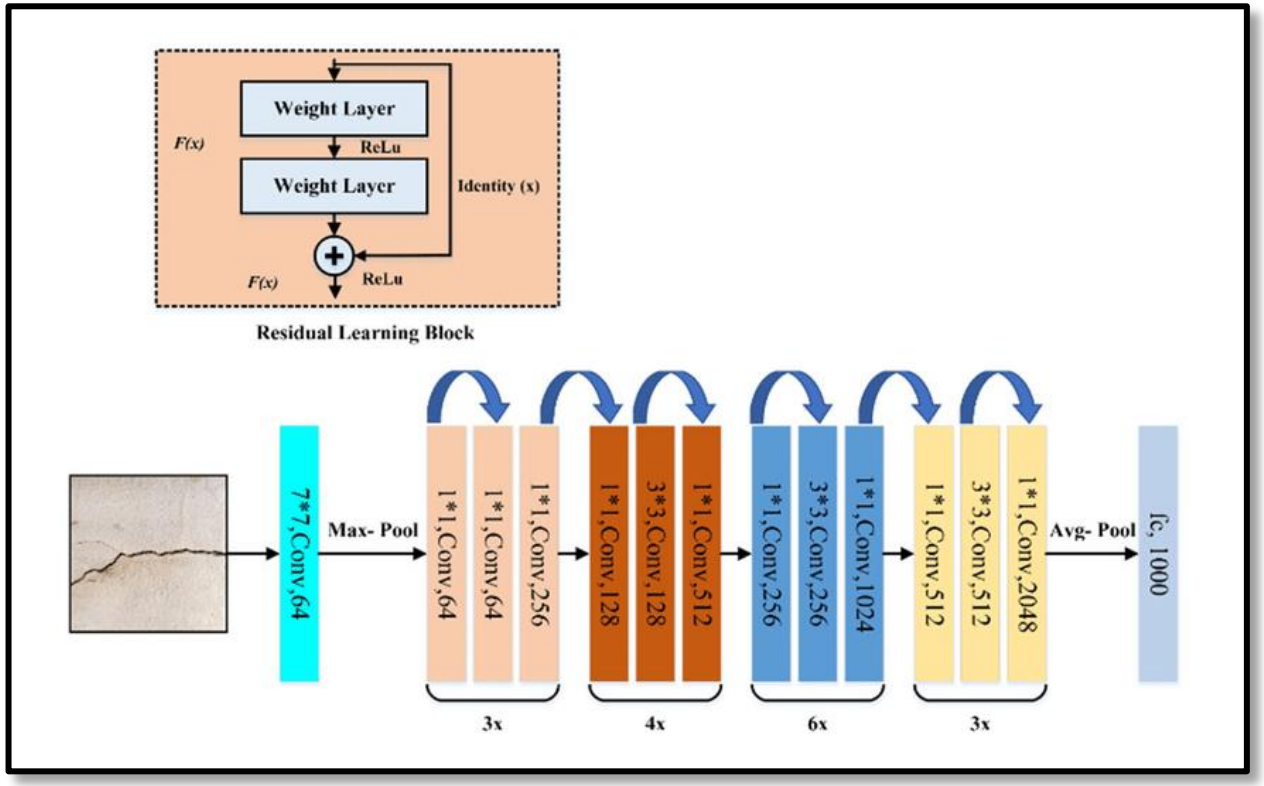


ResNet50

ResNet50 took the game to the next level by introducing an architecture with 50 convolutional layers. This achievement was made possible through the introduction of residual connections, which effectively combat the vanishing gradient problem.

ResNet50 outperformed the VGG16 model on the ImageNet dataset, achieving state-of-the-art performance with an accuracy of around 76%.

This model is very powerful; however, its biggest drawback is the complexity of understanding the residual connections.



Source: www.researchgate.net

Comparative Analysis of Models

Now that we are familiar with all four models, we can summarize what we have learned in a table. In this table, we can see the number of layers, the number of learnable parameters, activation functions, input size, the number of pooling layers, and the key innovations of each model.

We observe that the networks have become **progressively deeper**, which was made possible by the innovations they introduced. Another common trend was the **increase in the number of parameters** until **ResNet50** addressed this issue.

Feature	LeNet-5	AlexNet	VGG16	ResNet-50
Number of Layers	7 (5 convolutional, 2 fully connected)	8 (5 convolutional, 3 fully connected)	16 (13 convolutional, 3 fully connected)	50 (49 convolutional, 1 fully connected)
Parameters	~60,000	~62 million	~138 million	~26 million
Activation Function	Sigmoid	ReLU	ReLU	ReLU
Input Size	32x32 grayscale images	227x227 RGB images	224x224 RGB images	224x224 RGB images
Pooling Layers	2 average pooling layers	3 max pooling layers	5 max pooling layers	3 max pooling layers
Key Innovations	Early use of convolutional layers	ReLU activation, dropout, data augmentation	Small 3x3 filters, deep architecture	Residual connections to combat vanishing gradients

Model Complexity

We have observed that the models have progressively gotten larger, which may affect the training process. As the number of parameters increases, **more memory** is required during training.

On the other hand, as the models become deeper, it generally takes **longer** to complete both the forward and backward passes through the network. This increase in complexity can slow down the training process of the model.

It is crucial to be aware of memory usage and training time when building large CNN models.

CNN	Forward Pass	Backward Pass	Total Time
AlexNet	0.052 s	0.061 s	1140.15 s
GoogLeNet	0.013 s	0.019 s	332.228 s
VGG16	0.047 s	0.014 s	1960.2 s
VGG19	0.167 s	0.371 s	5411.31 s
ResNet-50	0.096 s	0.114 s	2101.19 s

Source: www.researchgate.net

The Evolution of CNN Models

Performance Metrics

One of the competitions and benchmarks for CNN models and image classifiers is ImageNet. This dataset contains large images of various objects we encounter in our everyday lives and includes 1,000 classes.
The accuracy of each model is measured in two different ways: top-1 accuracy and top-5 accuracy.
Top-1 accuracy refers to the percentage of times the model's highest predicted class matches the true label. In contrast, top-5 accuracy measures the percentage of times the true label is among the model's five highest predicted classes.

Architecture	Top-1 Accuracy	Top-5 Accuracy	Year
Alexnet	57.1	80.2	2012
Inception-V1	69.8	89.3	2013
VGG	70.5	91.2	2013
Resnet-50	75.2	93	2015
InceptionV3	78.8	94.4	2016

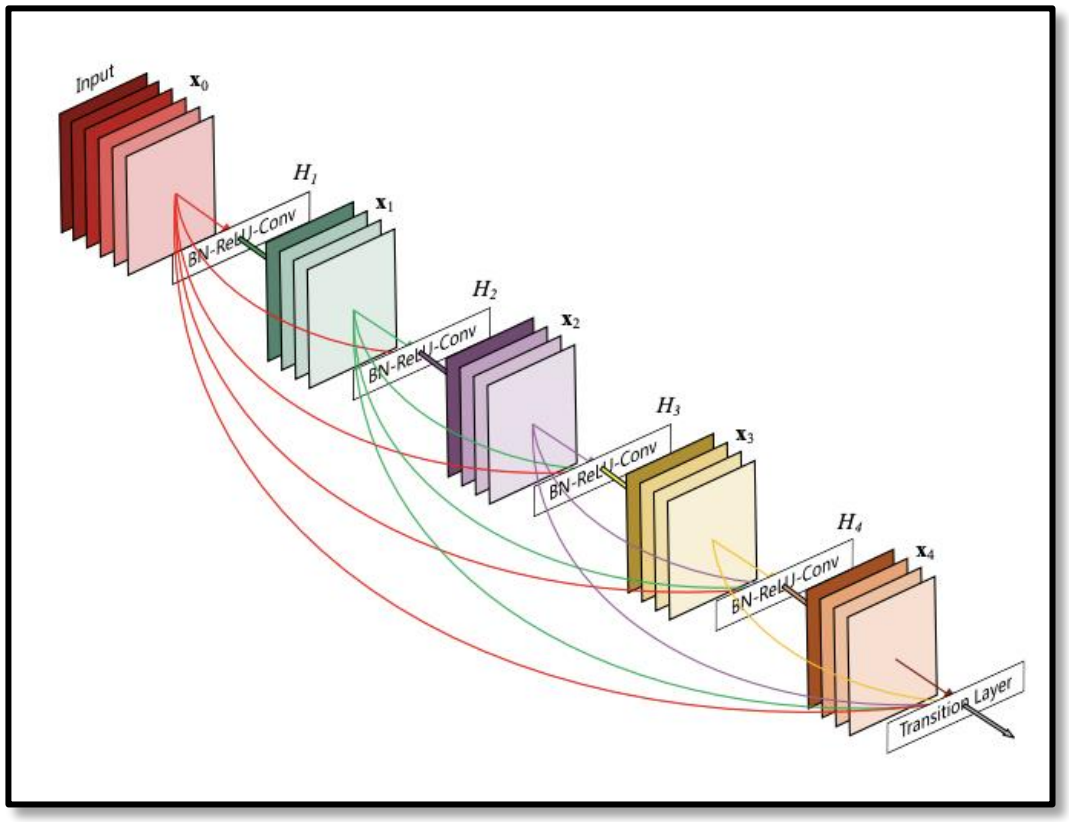
Source: cv-tricks.com

Innovations Beyond ResNet50

The ResNet's skip connections inspired more efficient methods for building and training very deep networks. For example, DenseNet establishes dense connections between all previous and subsequent layers.

It also uses bottleneck layers, which are 1x1 convolutional layers employed to reduce the size of the feature maps.

This architecture enables DenseNet to achieve better performance than ResNet with fewer parameters and less computational cost.

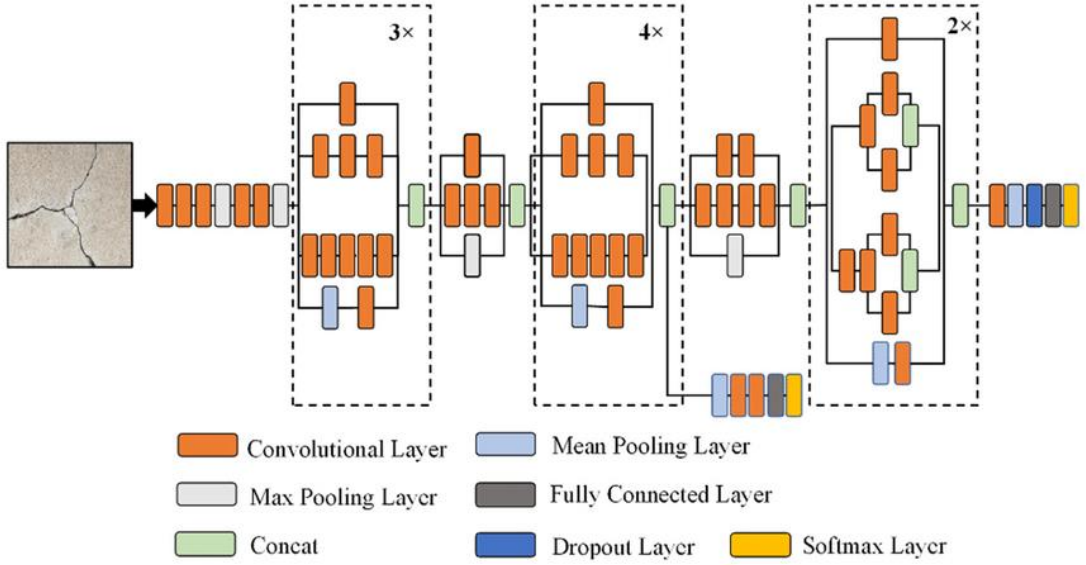


Source: www.pytorch.org

Innovations Beyond ResNet50

Another architecture inspired by ResNet is Inception. This architecture introduced the Inception module, which concatenates features from convolutions of different kernel sizes (1x1, 3x3, 5x5) to capture multi-scale features.

This design allows for increasing the network's depth and width without a significant increase in computational complexity.



Source: www.researchgate.net

Limitations of Current Models

Despite the exponential advances in CNN architecture, these models face numerous limitations. Current models, like DenseNet and VGG16, require substantial computational resources and memory due to their large number of parameters. Training these models is very time-consuming, often necessitating powerful GPUs and extensive time to achieve optimal performance. The complexity of these models increases the risk of overfitting, especially when trained on smaller datasets.

Additionally, the performance of these models can be highly sensitive to hyperparameters, such as learning rate and batch size, which complicates the training process. Understanding how these models make decisions can also be challenging, which is a significant concern in fields like healthcare, where model transparency is crucial. Furthermore, the complexity of these models can lead to latency issues in real-time applications, making them unsuitable for tasks requiring immediate responses, such as autonomous driving or live video analysis.

Future Directions

Many researchers are conducting studies to address the current limitations of CNN architectures. Some efforts focus on data preparation techniques to reduce overfitting, while others are exploring self-supervised learning, which enables models to leverage unlabeled data for pre-training.

Additionally, advancements in interpretability are crucial for the future of CNNs. The development of Explainable AI (XAI) methods will enhance our understanding of the decision-making processes of CNNs, making these models more transparent and trustworthy.