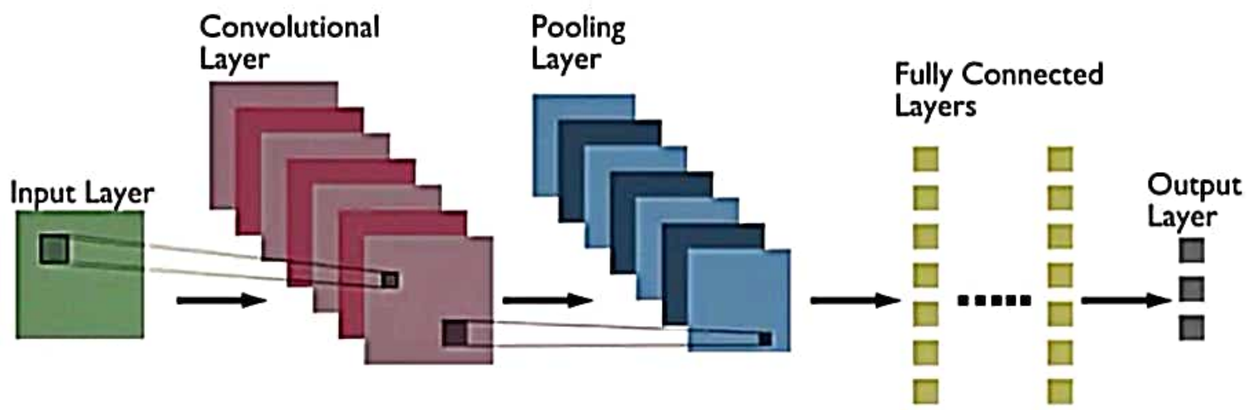
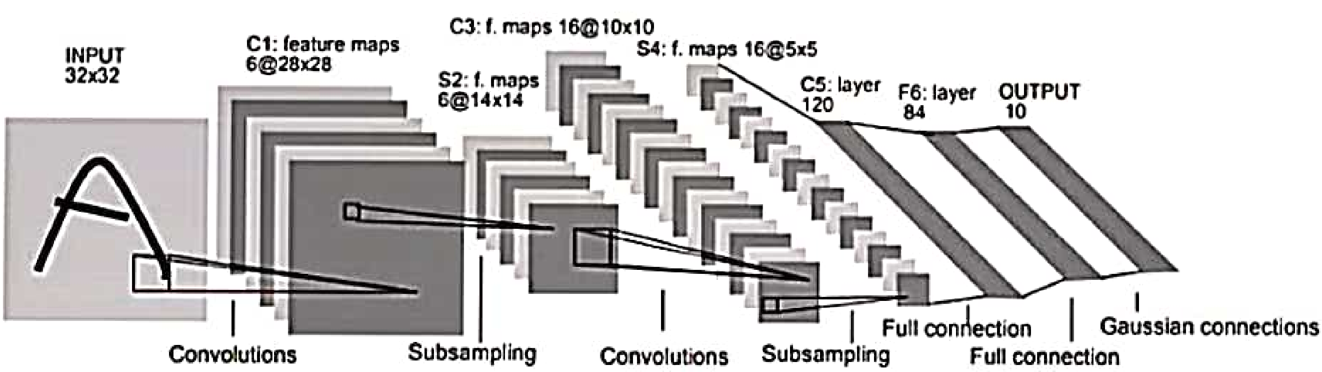
Convolutional Neural Networks (CNNs) have revolutionized the field of computer vision, powering advancements in image recognition, object detection, and various other visual tasks. The success of CNNs lies in their ability to automatically learn hierarchical representations from data. In this article, we'll explore the rich landscape of CNN architectures, each tailored to specific challenges and use cases.



CNN Architecture

1. LeNet-5: The Pioneer

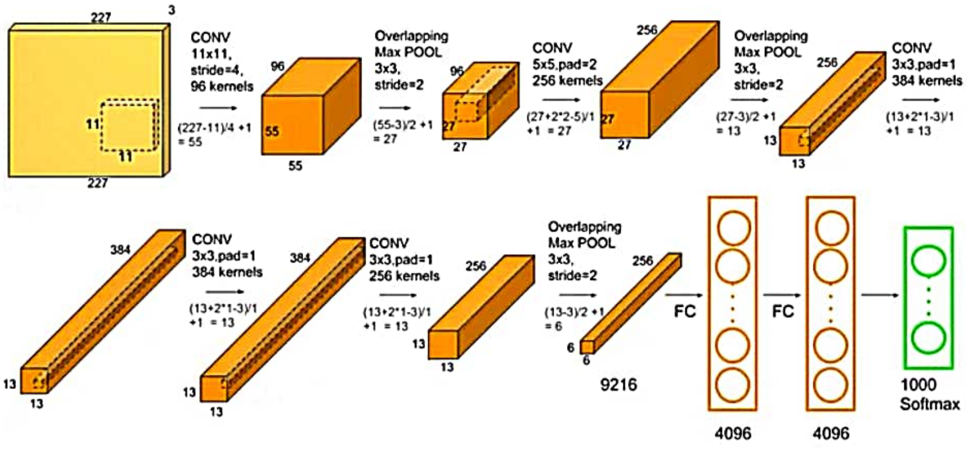
LeNet-5, introduced by Yann LeCun and his team in the 1990s, was one of the first successful CNN architectures. Designed for handwritten digit recognition, it laid the foundation for subsequent CNN developments. LeNet-5 features convolutional layers, subsampling layers, and fully connected layers, showcasing the core elements of modern CNNs.



LeNet CNN

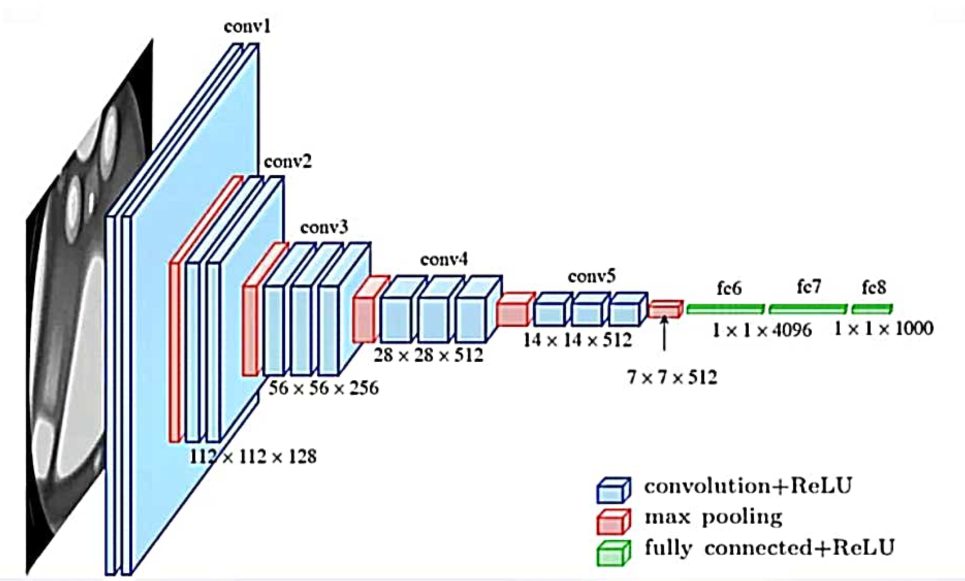
2. AlexNet: Igniting Deep Learning Resurgence

AlexNet, created by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, marked a turning point in deep learning. Introduced in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, AlexNet featured deep convolutional layers. The AlexNet architecture was designed to be used with large-scale image datasets and it achieved state-of-the-art results at the time of its publication. AlexNet is composed of 5 convolutional layers with a combination of max-pooling layers, 3 fully connected layers, and 2 dropout layers. The activation function used in all layers is Relu. The activation function used in the output layer is Softmax. The total number of parameters in this architecture is around 60 million.



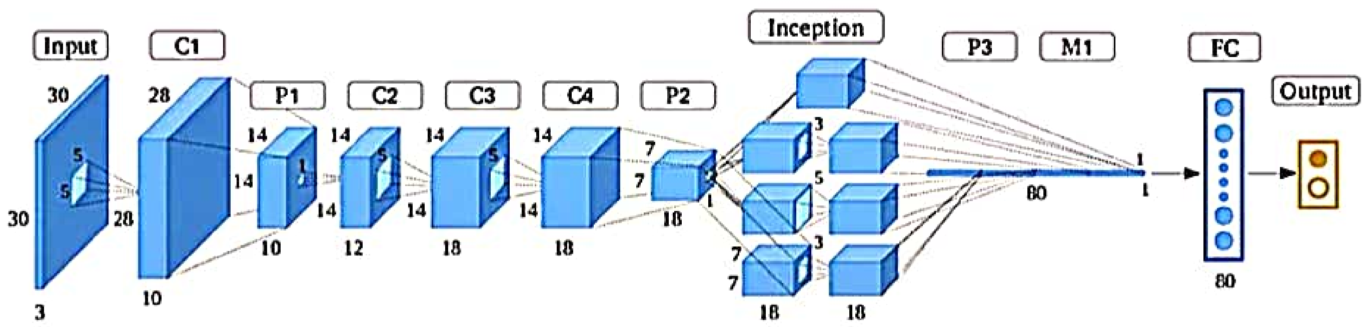
3. VGGNet: The Pursuit of Simplicity

The Visual Geometry Group (VGG) at Oxford University proposed the VGGNet architecture. VGGNet is known for its simplicity, featuring a uniform architecture with small receptive fields (3x3 convolutional kernels) and deep stacks of layers. Its straightforward design contributed to its popularity, and VGG models come in various depths (e.g., VGG16, VGG19).



4. GoogLeNet (Inception): Embracing Parallelism

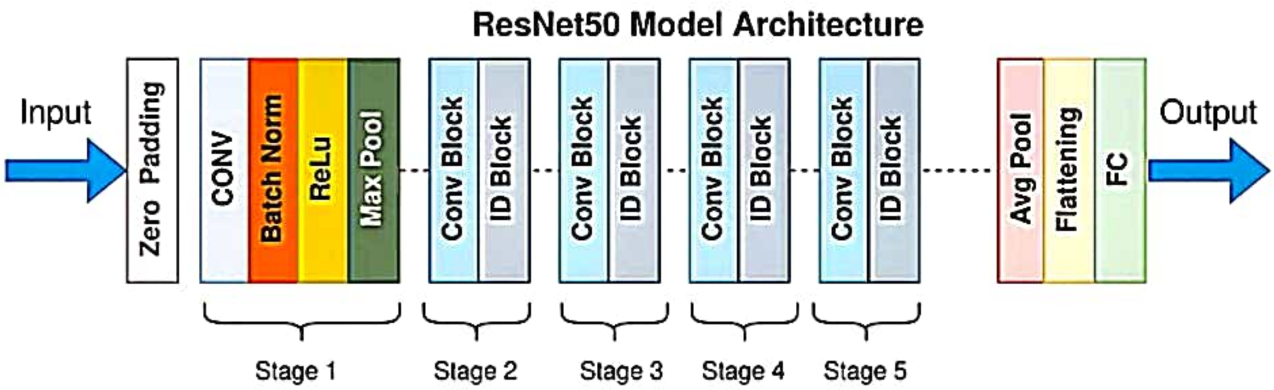
GoogLeNet, winner of ILSVRC 2014, introduced the Inception module, which employs parallel convolutional operations with different kernel sizes. This architecture efficiently captures features at multiple scales, promoting better generalization. GoogLeNet showcased the benefits of inception modules for improving performance.



Google Net Like Architecture

5. ResNet: Tackling Vanishing Gradients

Residual Networks, or ResNets, proposed by Kaiming He et al., addressed the challenge of training very deep networks. ResNets introduce shortcut connections that bypass one or more layers, allowing the gradient to flow more easily during backpropagation. This architectural innovation facilitated the training of extremely deep networks, reaching hundreds of layers.



6. MobileNet: Lightweight Efficiency

MobileNet, designed by Google, focuses on efficiency for mobile and edge devices. It employs depthwise separable convolutions, separating spatial and depthwise convolutions to reduce the number of parameters and computations. MobileNet strikes a balance between accuracy and computational efficiency, making it ideal for resource-constrained environments.

Data Augumentation

Purpose - Increase accuracy.

-> increase the size of the dataset and balance the classes by applying different transformation

Augumented data

-> driven from original data with some minor changes

-> In case of image

- flipping
- resizing
- cropping
- brightness

Synthetic data

Artificial

no original dataset

Audio Data Augumentation

1) Noise injection - add gaussian or random noise to the audio dataset to improve the model performance.

2) Shifting - shift audio left fast forward or right with random seconds

3) Changing the speed - Stretches the time series by a fixed rate.

4) Changing the pitch - randomly change the pitch of the audio.

Text Data Augmentation

1) Word or sentence Shuffling: randomly changing the position of a word or sentence

2) Word Replacement: replace words with synonyms.

3) Syntax-tree manipulation - Paraphrase the sentence using the same word

4) Random word insertion

5) Random word Deletion

Image Augmentation

1) Geometric transformations
- randomly flip
- Crop
- rotate
- Stretch and
- Zoom

multiple transformations

2) Color Space transformation

3) Kernel filters — randomly change the sharpness or blurring of the image

4) Random erasing
delete some part of the image

5) Mixing images
— blending and mixing multiple images.

Application

→ Healthcare
→ Self-Driving Cars
→ NLP
→ Automatic Speech Recognition