CNN

A CNN (Convolutional Neural Network) is a type of deep learning model mainly used for image processing, computer vision, and increasingly for audio, text, and spatial data. It is especially powerful at detecting patterns such as edges, textures, shapes, and objects from raw images.

LeNET

- Introduced by Yann LeCun in 1998 for handwritten digit recognition (MNIST).
- 7 layers with learnable parameters (excluding input), including 2 conv layers and 2 fully connected layers.
- Uses average pooling (subsampling) instead of max pooling.
- Total parameters ≈ 60,000 (specifically ~60k–62k depending on implementation).
- Activation: tanh (original version), not ReLU.
- Input size is 32×32 grayscale, giving the network slightly more border context for convolution.

AlexNet

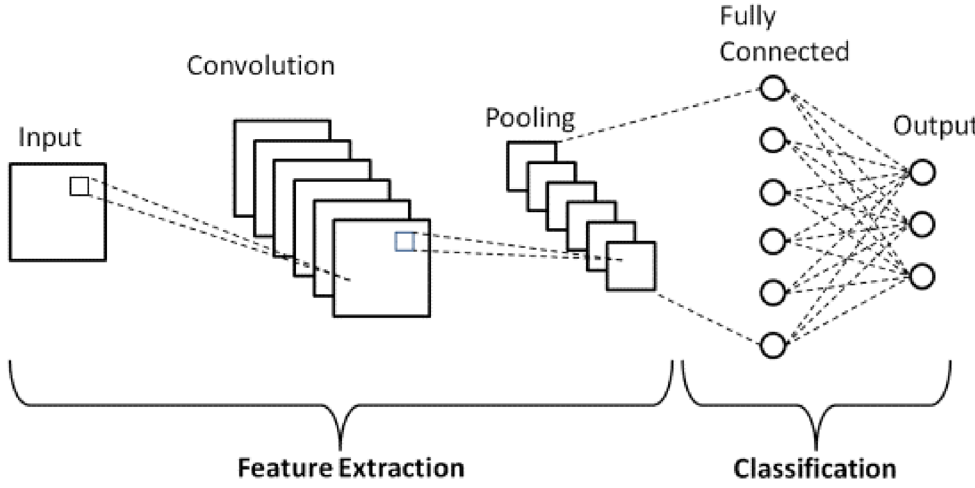
- Introduced in 2012 by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton — first CNN to win ImageNet with huge performance gap.
- Total parameters ≈ 60 million, significantly larger than previous CNNs like LeNet.
- 8 learnable layers: 5 convolutional + 3 fully connected layers.
- Used ReLU activation, which dramatically sped up training compared to tanh/sigmoid.
- Introduced dropout (0.5) in fully connected layers to reduce overfitting.
- Uses overlapping max-pooling and local response normalization (LRN) in early layers.

GoogleNet

- Introduced in 2014 (ImageNet winner) by Szegedy et al. using the Inception architecture.
- Very deep network — 22 layers (27 with pooling), but computationally efficient.
- Uses Inception modules, combining 1×1, 3×3, 5×5 convolutions + pooling in parallel.
- 1×1 convolutions used for dimensionality reduction, drastically lowering computation.
- Total parameters ≈ 6.8 million, far fewer than AlexNet (60M).
- Includes auxiliary classifiers to help gradient flow in deep layers.

ResNet

- Introduced in 2015 (He et al.) and won ImageNet 2015 by a large margin.
- Introduced Residual Learning using skip/shortcut connections to solve vanishing gradients.
- Extremely deep networks — ResNet-18, 34, 50, 101, 152 layers.
- Identity skip connections help gradients flow directly, enabling stable training of 100+ layers.
- Uses bottleneck blocks (1×1 → 3×3 → 1×1) in deeper versions like ResNet-50+.
- Parameters:
  - ResNet-18 → ~11.7M
  - ResNet-34 → ~21.8M
  - ResNet-50 → ~25.6M
  - ResNet-101 → ~44.5M



CNN=CONVOLUTION+POOLING

Feature Extraction

Dimension Reduction

CNN Architecture Analysis

LENET

- **Structure:** 2 conv layers + 2 average pooling + 2 FC + outStructure: 2 conv layers + 2 average pooling + 2 FC + output layer
- **Convolutions:** 5×5 filters
- **Parameters**: ~60K
- **FC layers**: 2 (120 → 84 → 10)
- **Distinctive Features**: Tanh activation, designed for MNIST digits, simple & small.

ALEXNET

- **Structure:** 5 conv layers + 3 FC layers
- **Convolutions:**
  Conv1: 11×11
  Conv2: 5×5
  Conv3–5: 3×3
- **Parameters:** ~60 **million**
- **FC Layers:**
  4096 → 4096 → 1000
- **Distinctive Features:**
  First large-scale CNN success
  ReLU activation
  Dropout to reduce overfitting
  Local Response Normalization (LRN)
  Overlapping max-pooling

GOOGLENET

- **Structure:** 22 layers deep; multiple Inception modules
- **Convolutions:** Parallel 1×1, 3×3, 5×5 filters
- **Parameters:** ~6.8 million (very efficient)
- **FC Layers:** 1 small FC + softmax
- **Distinctive Features:**

1. Inception module (multi-scale feature extraction)
2. 1×1 convolutions for dimensionality reduction
3. Two auxiliary classifiers inside the network
4. Very deep yet computationally efficient

-

RESNET

- **Structure:** 50 layers using residual blocks
- **Convolutions:**

1. Bottleneck: 1×1 → 3×3 → 1×1

- **Parameters:**

2. ResNet-18 → 11M
3. ResNet-34 → 21M
4. ResNet-50 → 25.6M

- **FC Layers:** 1 FC layer (2048 → 1000)
- **Distinctive Features:**

5. Skip connections fix vanishing gradient
6. Allows 100+ layer deep networks
7. Stable and easy to optimize
8. State-of-the-art backbone for many tasks
9.

CNN Additional Feature

Guided Back Propagation

- Visualization technique (2014) used to understand what features a CNN has learned by highlighting important pixels that contribute to a prediction.
- Modifies standard backpropagation by allowing only positive gradients to pass through ReLU, combining ideas of deconvnet and backprop.
- Produces high-resolution, fine-grained saliency maps, especially effective for visualizing textures and edges.
- Used mainly for model interpretability, helping analyze convolutional filters, neuron activations, and class decisions.

Deep Dream

- DeepDream (2015) is a visualization technique from Google that enhances and amplifies patterns learned by a CNN to reveal what the network “sees.”
- It works by maximizing activations of selected layers during gradient ascent, causing the image to evolve into dream-like patterns.
- Produces hallucinogenic, surreal images with exaggerated textures, shapes, and objects (e.g., dogs, eyes, buildings).
- Used for model interpretability and artistic image generation, helping visualize feature hierarchies in deep networks.

Deep Art

- DeepArt (2015) popularized neural style transfer, where a CNN (usually VGG-19) transfers the style of one image (painting) onto the content of another image.
- Uses Gram matrices to capture artistic style features (colors, textures, brush strokes) from deep CNN layers.
- Optimizes an output image by minimizing a content loss + style loss to blend both images harmoniously.
- Applied for artistic image generation, photo stylization, and creative artwork production using deep learning.