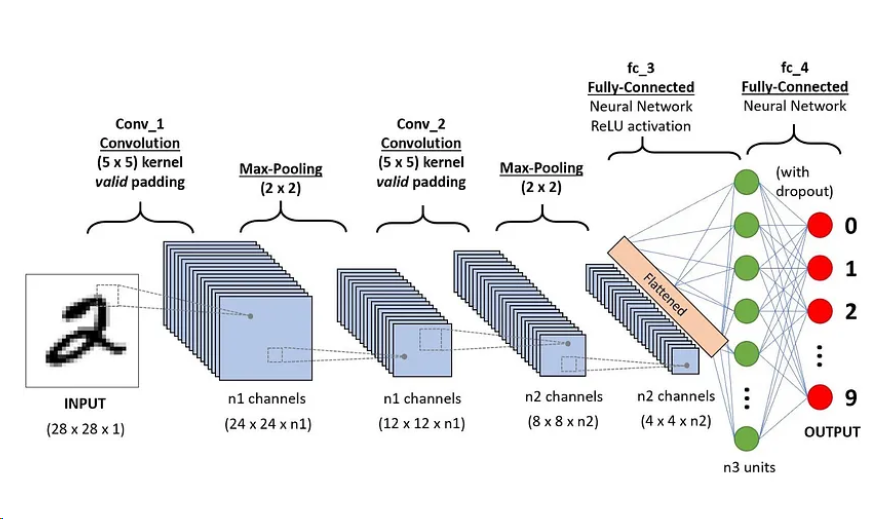
CNN



Slide 1: CNN Architecture Overview for Image Classification

Input Layer: Takes a grayscale image (28x28x1) as input.

Example: Handwritten digit (MNIST dataset).

Convolutional Layer 1 (Conv\_1): Applies a 5×5 kernel with valid padding.

Produces feature maps with n1 channels, transforming the input to size 24×24×n1.

Max-Pooling Layer 1: Reduces spatial dimensions by pooling with a 2×2 kernel.

Output size becomes 12×12×n1.

Slide 2: Deep Network Layers and Output

Convolutional Layer 2 (Conv\_2): Second 5×5 convolution with valid padding.

Output feature maps with n2 channels, resulting in size 8×8×n2.

Max-Pooling Layer 2: Another pooling layer with

2×2 kernel, reducing dimensions to 4×4×n2.

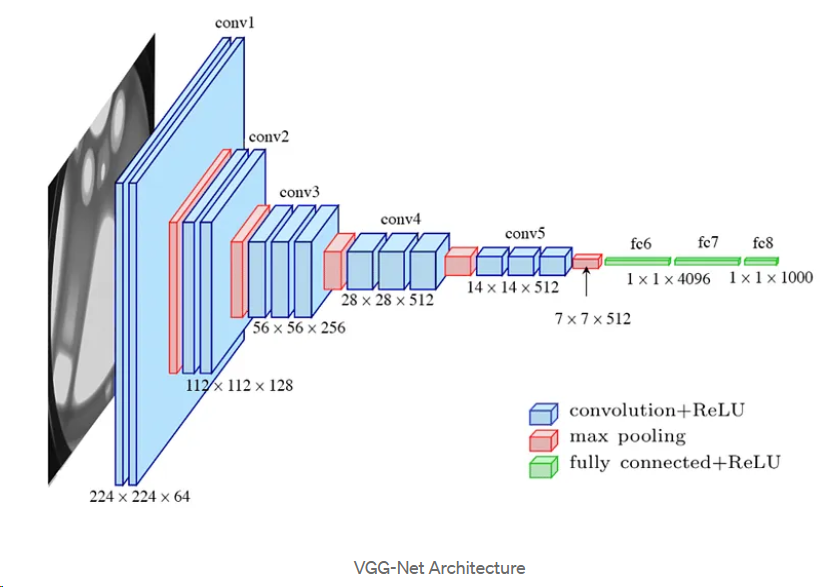
Fully Connected Layers (fc\_3 and fc\_4):

fc\_3: Neural network layer with ReLU activation, flattened to n3 units.

fc\_4: Output layer with dropout, classifying digits (0-9).

Output: Predicts digit class (0-9) based on learned features.

VGGNet



Slide 1: VGG-Net Architecture Overview

Input Layer: Processes input image of size 224×224×3.

Typical application: Image classification tasks.

Convolutional Layers: Uses small 3×3 convolution filters with ReLU activation.

Stack of convolutional layers for hierarchical feature extraction.

Increases depth as we go deeper: conv1 (64 filters), conv2 (128 filters), conv3 (256 filters), conv4 (512 filters), conv5 (512 filters).

Max Pooling Layers: Reduces spatial dimensions after each convolutional block.

Helps in down-sampling and retaining important features.

Slide 2: Fully Connected Layers and Output

Fully Connected Layers (fc6, fc7, fc8):

fc6 and fc7: Fully connected layers with 4096 units each and ReLU activation.

fc8: Final fully connected layer for classification with 1000 units (corresponding to 1000 classes in ImageNet).

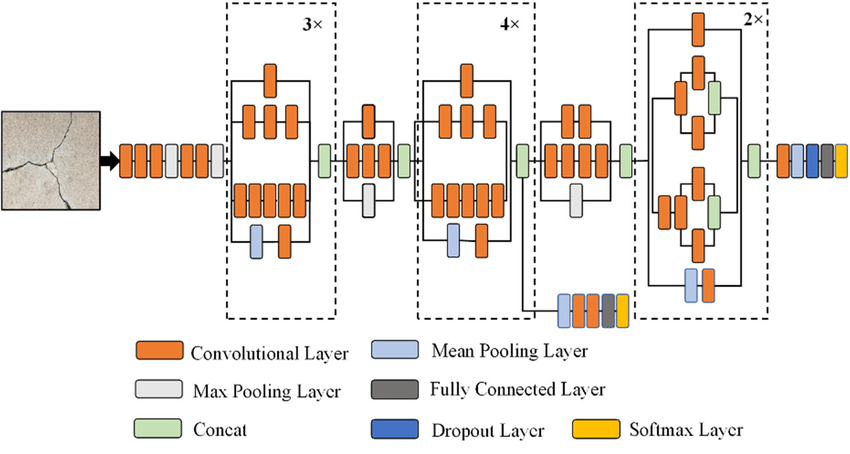
Feature Map Transition: Progressively reduces spatial dimensions from 224×224 to 7×7 while increasing feature depth.

Achieves a balance between spatial resolution and feature richness.

Applications of VGG-Net: Highly effective in image classification and transfer learning tasks.

Popular baseline model for object detection and segmentation.

Inception V3



Slide 1: Inception V3 Model Overview

Model Purpose:

Designed for image classification tasks, leveraging complex feature extraction.

Efficiently learns both local and global features.

Inception Module Architecture:

Uses multiple convolutional filters (e.g., 1×1, 3×3) in parallel within each module.

Each module has multiple pathways, capturing different feature scales simultaneously.

Pooling Layers:

Max Pooling and Mean Pooling layers are used for spatial dimension reduction.

Pooling layers help in down-sampling, retaining key features while reducing computational load.

Slide 2: Advanced Layers and Output

Concatenation Layers:

Combines outputs from different filters within each Inception module.

Allows the network to merge feature maps from various filter sizes.

Fully Connected and Dropout Layers:

Fully connected layers at the end for classification, enabling learning of high-level features.

Dropout layer helps in regularization to prevent overfitting.

Output Layer:

Softmax layer for multi-class classification (shown in yellow).

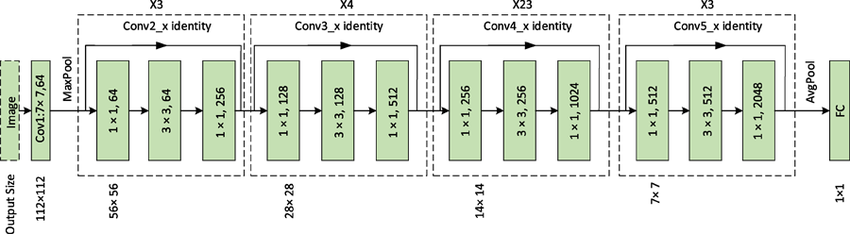
Final output represents probability distribution across possible classes.

Advantages of Inception V3:

Optimized for both accuracy and computational efficiency.

Suitable for real-world applications, including object detection and medical imaging.

ResNet101



Slide 1: ResNet-101 Architecture Overview

Model Purpose:

ResNet-101 is a deep residual network, designed to solve the vanishing gradient problem in very deep networks.

Enables training of over 100 layers, improving feature learning and performance.

Residual Blocks with Identity Mapping:

Uses identity shortcuts to skip layers, allowing gradient flow and helping the model avoid degradation in accuracy.

Each block has a 3-layer bottleneck structure with 1×1, 3×3, and 1×1 convolution.

Initial Layers: Input image size starts at 224×224.

First layer: 7x7 convolution with 64 filters, followed by MaxPooling, reducing spatial dimensions.

Slide 2: Deep Layers and Output

Layer Stacks:

Contains 4 main stages with increasing filter sizes:

Conv2\_x: 256 filters, repeated 3 times.

Conv3\_x: 512 filters, repeated 4 times.

Conv4\_x: 1024 filters, repeated 23 times.

Conv5\_x: 2048 filters, repeated 3 times.

Deeper layers capture more complex and abstract features.

Pooling and Fully Connected Layers:

Average Pooling layer reduces the spatial dimensions to 1×1.

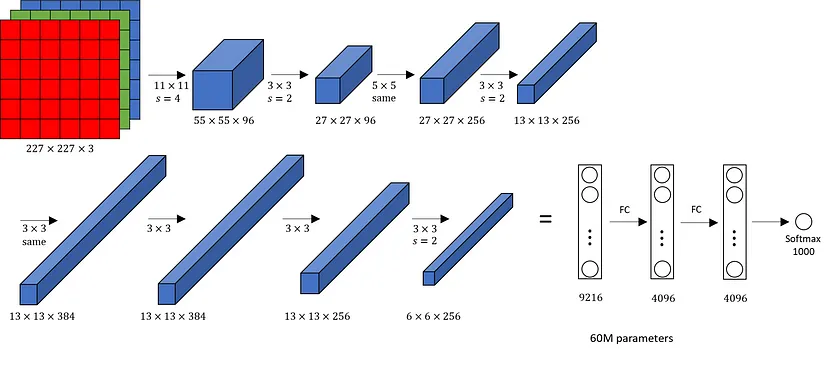
Fully Connected (FC) layer for final classification.

Key Advantages:

Skip connections help avoid vanishing gradients.

Improved accuracy with minimal computational cost increase.

AlexNet



Slide 1: AlexNet Architecture Overview

Introduction:

AlexNet is a pioneering deep convolutional neural network designed for large-scale image classification (ImageNet).

Achieved breakthrough performance and won the ImageNet competition in 2012.

Input and Initial Convolution Layers:

Input image size: 227×227×3 (RGB image).

First convolutional layer: 11×11 kernel, stride of 4, produces feature maps of size 55×55×96.

Subsequent convolutional and pooling layers reduce spatial dimensions and increase depth.

Layer Design:

Alternates between convolutional layers with ReLU activation and pooling layers for down-sampling.

Introduced overlapping pooling, which improves generalization.

Slide 2: Fully Connected Layers and Model Summary

Convolutional Layers and Feature Extraction:

Stacks additional convolutional layers with 3×3 kernels, building a rich feature representation.

Final convolutional layer outputs feature maps of size 6×6×256.

Fully Connected Layers:

Three fully connected layers:

fc1: 9216 units (flattened from previous layer).

fc2 and fc3: 4096 units each with ReLU activation.

Uses Dropout for regularization, helping prevent overfitting.

Output and Parameters:

Final output layer: Softmax with 1000 units for classification.

Total parameters: approximately 60 million.

AlexNet’s design balances depth with computational efficiency, suitable for large datasets.