**CNN MODELS**

**LeNet-5**

**1. Architecture of LeNet-5**

LeNet-5 consists of **seven layers** (excluding input) that include convolutional, pooling, and fully connected layers.

**Layer-wise Breakdown:**

| **Layer** | **Type** | **Number of Filters** | **Kernel Size** | **Activation** | **Output Size** |
| --- | --- | --- | --- | --- | --- |
| **Input** | Image (Grayscale) | - | - | - | 32 × 32 |
| **C1** | Convolutional | 6 | 5 × 5 | tanh | 28 × 28 × 6 |
| **S2** | Subsampling (Avg Pooling) | - | 2 × 2 | - | 14 × 14 × 6 |
| **C3** | Convolutional | 16 | 5 × 5 | tanh | 10 × 10 × 16 |
| **S4** | Subsampling (Avg Pooling) | - | 2 × 2 | - | 5 × 5 × 16 |
| **C5** | Fully Connected Conv | 120 | 5 × 5 | tanh | 1 × 1 × 120 |
| **F6** | Fully Connected | - | - | tanh | 84 |
| **Output** | Fully Connected | - | - | softmax | 10 (digits 0-9) |

**2. Key Features of LeNet-5**

✅ **Convolutional Layers** extract spatial features from input images.  
✅ **Pooling Layers** (subsampling) reduce dimensionality and improve translation invariance.  
✅ **Fully Connected Layers** classify the extracted features.  
✅ **Tanh Activation** (instead of ReLU, which became popular later).  
✅ **Softmax Layer** for digit classification (0–9).

**3. Advantages of LeNet-5**

* **Computationally efficient** due to small kernel sizes.
* **Less complex** compared to modern deep networks.
* **Pioneered CNN architecture** for modern deep learning model

**4. Applications of LeNet-5**

* **Handwritten digit recognition** (MNIST dataset)
* **Character recognition** in OCR systems
* **Early medical image classification tasks**
* **Fundamental CNN model for educational purposes**

**5. Limitations**

❌ **Not deep enough** for complex datasets.  
❌ **Uses tanh activation**, which is less efficient than ReLU.  
❌ **Not optimized** for large-scale image classification tasks like ImageNet.

**Conclusion**

LeNet-5 laid the foundation for modern CNN architectures like AlexNet, VGG, and ResNet. Despite its simplicity, it remains a **milestone in deep learning history** and is still widely used for learning and experimentation.

**GoogLeNet (Inception) CNN Model**

GoogLeNet (also known as **Inception v1**) is a deep CNN architecture developed by **Google** in **2014**. It won the **ILSVRC 2014** (ImageNet Large Scale Visual Recognition Challenge) by significantly reducing computational complexity while maintaining high accuracy.

**1. Key Features of GoogLeNet**

✅ **Inception modules**: Uses multiple kernel sizes (1×1, 3×3, 5×5) in parallel.  
✅ **Deep architecture (22 layers)**: But computationally efficient.  
✅ **1×1 Convolutions**: Used for dimensionality reduction before expensive convolutions.  
✅ **Auxiliary classifiers**: Additional classifiers in intermediate layers to improve gradient flow.  
✅ **No fully connected layers**: Uses global average pooling to reduce parameters.

**2. GoogLeNet Architecture**

The **GoogLeNet model has 22 layers** (excluding pooling layers) and consists of multiple **Inception modules**.

**Inception Module (Key Innovation)**

Each **Inception module** applies:

1. **1×1 Convolutions** (for dimension reduction and non-linearity)
2. **3×3 Convolutions**
3. **5×5 Convolutions**
4. **3×3 Max Pooling**
5. **Concatenation of all outputs**

This allows the model to **capture multi-scale features efficiently**.

**Overall Structure**

| **Layer** | **Type** | **Filter/Stride** | **Output Size** |
| --- | --- | --- | --- |
| **Input** | Image | 224×224×3 | 224×224×3 |
| **Conv1** | 7×7 Conv + Max Pool | 64 filters, stride 2 | 112×112×64 |
| **Conv2** | 3×3 Conv + Max Pool | 192 filters | 56×56×192 |
| **Inception 3a** | Inception Module | Various | 28×28×256 |
| **Inception 3b** | Inception Module | Various | 28×28×480 |
| **Max Pool** | Pooling | - | 14×14×480 |
| **Inception 4a** | Inception Module | Various | 14×14×512 |
| **Inception 4b, 4c, 4d, 4e** | Inception Modules | Various | 14×14×528 → 14×14×832 |
| **Max Pool** | Pooling | - | 7×7×832 |
| **Inception 5a** | Inception Module | Various | 7×7×832 |
| **Inception 5b** | Inception Module | Various | 7×7×1024 |
| **Global Average Pooling** | 1×1 | - | 1×1×1024 |
| **Fully Connected** | Softmax | 1000 classes | 1×1×1000 |

**3. Advantages of GoogLeNet**

✅ **Computationally Efficient**: Uses **1×1 convolutions** to reduce parameters.  
✅ **Deep but Optimized**: 22 layers, but **fewer parameters than VGG-16**.  
✅ **Better Feature Extraction**: Uses multi-scale convolutions in **Inception modules**.  
✅ **Improved Gradient Flow**: **Auxiliary classifiers** help with training deep networks.

**4. Applications of GoogLeNet**

* **Image Classification** (ImageNet, CIFAR-10)
* **Object Detection** (Faster R-CNN, YOLO)
* **Face Recognition** (FaceNet)
* **Medical Imaging** (CT/MRI image analysis)

**5. Limitations of GoogLeNet**

❌ **Complex Architecture**: Hard to implement manually.  
❌ **Vanishing Gradient Problem**: Improved later with **Batch Normalization (Inception v2/v3)**.  
❌ **Lacks Fully Connected Layers**: Might not work well for some tasks.

**Conclusion**

GoogLeNet introduced the **Inception module**, which became the foundation for later deep learning models. It was **lighter and more efficient** than VGG while maintaining **high accuracy**. Modern CNNs like **Inception v3, v4, Xception, and EfficientNet** evolved from this architecture.