**Detailed Documentation on CNNs (AlexNet, VGGNet, and GoogleNet)**

This detailed documentation covers the three popular Convolutional Neural Networks (CNNs) that significantly influenced the field of computer vision:

* **AlexNet (2012)**
* **VGGNet (2014)**
* **GoogleNet (Inception) (2014)**

## ****1. AlexNet****

### 📌 **Introduction**

* **Developed by:** Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton
* **Published in:** NIPS 2012
* **Competition:** ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2012
* **Achievement:** Won 1st place, reducing the error rate from 26% to 15%

AlexNet was a groundbreaking architecture that popularized the use of deep convolutional neural networks. It marked a significant leap in image classification accuracy and brought AI to the forefront.

### 📦 **Architecture Overview**

* **8 Layers** (5 Convolutional and 3 Fully Connected)
* **ReLU Activation** for faster training
* **Dropout** for regularization
* **Max-Pooling** for feature reduction
* **Softmax Classifier** for multi-class classification

#### 📊 **Detailed Layer-by-Layer Breakdown**

| **Layer** | **Type** | **Number of Filters** | **Kernel Size** | **Stride** | **Activation** | **Additional Features** |
| --- | --- | --- | --- | --- | --- | --- |
| Input | Image Input | - | 227x227x3 | - | - | RGB Image |
| Conv1 | Convolutional Layer | 96 | 11x11 | 4 | ReLU | Overlapping MaxPooling |
| MaxPool1 | MaxPooling | - | 3x3 | 2 | - | - |
| Conv2 | Convolutional Layer | 256 | 5x5 | 1 | ReLU | - |
| MaxPool2 | MaxPooling | - | 3x3 | 2 | - | - |
| Conv3 | Convolutional Layer | 384 | 3x3 | 1 | ReLU | - |
| Conv4 | Convolutional Layer | 384 | 3x3 | 1 | ReLU | - |
| Conv5 | Convolutional Layer | 256 | 3x3 | 1 | ReLU | - |
| MaxPool3 | MaxPooling | - | 3x3 | 2 | - | - |
| FC1 | Fully Connected Layer | 4096 | - | - | ReLU | Dropout (50%) |
| FC2 | Fully Connected Layer | 4096 | - | - | ReLU | Dropout (50%) |
| FC3 | Fully Connected Layer | 1000 | - | - | Softmax | - |

### 🧠 **Key Features of AlexNet**

1. **ReLU Activation:**
   * Faster convergence than sigmoid or tanh.
   * Helps prevent vanishing gradient problems.
2. **GPU Utilization:**
   * Implemented on NVIDIA GPUs for accelerated training.
3. **Data Augmentation:**
   * Applied techniques like random cropping, flipping, and color shifting to increase dataset size.
4. **Dropout:**
   * Reduced overfitting by randomly dropping neurons during training.

### ⚙️ **Limitations of AlexNet**

* Large memory consumption (~60 million parameters).
* High computational complexity.
* Overlapping max pooling increased redundancy.

## ****2. VGGNet****

### 📌 **Introduction**

* **Developed by:** Visual Geometry Group (VGG), University of Oxford
* **Published in:** ILSVRC 2014
* **Achievement:** Achieved second place with 92.7% top-5 accuracy

VGGNet introduced a simple and uniform architecture using small convolutional filters to achieve high accuracy. Its deep architecture makes it ideal for transfer learning tasks.

### 📦 **Architecture Overview**

* **16 or 19 Layers** (VGG16 or VGG19)
* **Convolutional Layers with 3x3 Kernels**
* **Max-Pooling with 2x2 Kernels**
* **Fully Connected Layers**
* **Softmax Classifier**

#### 📊 **VGG16 Layer Breakdown**

| **Layer** | **Type** | **Number of Filters** | **Kernel Size** | **Stride** | **Activation** | **Additional Features** |
| --- | --- | --- | --- | --- | --- | --- |
| Input | Image Input | - | 224x224x3 | - | - | RGB Image |
| Conv1\_1 | Convolutional Layer | 64 | 3x3 | 1 | ReLU | - |
| Conv1\_2 | Convolutional Layer | 64 | 3x3 | 1 | ReLU | - |
| MaxPool1 | MaxPooling | - | 2x2 | 2 | - | - |
| Conv2\_1 | Convolutional Layer | 128 | 3x3 | 1 | ReLU | - |
| Conv2\_2 | Convolutional Layer | 128 | 3x3 | 1 | ReLU | - |
| MaxPool2 | MaxPooling | - | 2x2 | 2 | - | - |
| Conv3\_1 | Convolutional Layer | 256 | 3x3 | 1 | ReLU | - |
| Conv3\_2 | Convolutional Layer | 256 | 3x3 | 1 | ReLU | - |
| MaxPool3 | MaxPooling | - | 2x2 | 2 | - | - |
| FC1 | Fully Connected Layer | 4096 | - | - | ReLU | Dropout (50%) |
| FC2 | Fully Connected Layer | 4096 | - | - | ReLU | Dropout (50%) |
| FC3 | Fully Connected Layer | 1000 | - | - | Softmax | - |

### 🧠 **Key Features of VGGNet**

1. **Uniform Architecture:**
   * Consistent use of 3x3 convolutional filters.
2. **Depth:**
   * Deeper network with up to 19 layers.
3. **Pretrained Models:**
   * VGGNet is widely used for feature extraction in transfer learning tasks.

### ⚙️ **Limitations of VGGNet**

* **High Computational Cost:** Requires more GPUs for training.
* **Large Number of Parameters:** Over 138 million parameters.
* **Increased Memory Usage:** Due to fully connected layers.

## ✅ ****3. GoogleNet (Inception)****

### 📌 **Introduction**

* **Developed by:** Google
* **Published in:** ILSVRC 2014
* **Achievement:** Won 1st Place with a 93.3% top-5 accuracy

GoogleNet introduced the **Inception module** to reduce computational complexity by combining multiple convolution operations.

### 📦 **Architecture Overview**

* **22 Layers** (Including Inception Modules)
* **Inception Modules with 1x1, 3x3, and 5x5 Convolutions**
* **Auxiliary Classifiers for Regularization**
* **Global Average Pooling** Instead of Fully Connected Layers

### 🧠 **Key Features of GoogleNet**

1. **Inception Modules:**
   * Efficient multi-scale feature extraction using parallel convolutions.
2. **1x1 Convolutions:**
   * Reduced dimensionality and computational cost.
3. **Global Average Pooling:**
   * Minimized overfitting by using fewer parameters.

### ⚙️ **Limitations of GoogleNet**

* Complex design, making it harder to implement and debug.
* Computationally intensive for edge devices.