**Name: Muhammad Ahmad**

**Roll NO. 21L-5617 BS(DS) 7A**

**Deep Learning Assignment\_2**

**Task 1: Research and Analyze the Following Architectures**

**1.VGG16**

**Types of Convolutions:**

* **3x3 Convolutions:** These are the main filter size used in the network (Section 2.2: "We use very small 3 × 3 receptive fields throughout the whole network").
* **1x1 Convolutions:** Used sometimes to add non-linearity (Section 2.2: "1 × 1 convolutions... to increase the non-linearity of the decision function").

**Layer Dimensions:**

* The network has **11 to 19 layers**, including convolutional and fully connected layers (Section 2.3: "We evaluated the performance of networks of increasing depth...").
* Each convolutional layer has a **stride of 1** and uses **padding** to keep the same size (Section 2.2: "The convolution stride is fixed to 1 pixel").
* **Max-pooling layers** (2x2 with a stride of 2) reduce the size by half (Section 2.2: "Max-pooling is performed over a 2 × 2 pixel window, with stride 2").

**Number of Parameters:**

* The largest model has about **138 million parameters** (Section 3.3: "The largest model... has about 144M parameters").

**2. AlexNet**

### First Convolutional Layer

* **Input Dimensions:** 224 × 224 × 3 (Width × Height × Channels)
* **Kernel Size:** 11 × 11 × 3
* **Number of Kernels:** 96
* **Stride:** 4

Output Height=⌊Stride Input Height−Kernel Height+2×Padding​⌋+1

**Output Dimensions:** 54 × 54 × 96

Parameters=(Kernel Height×Kernel Width×Input Channels+1)×Number of Kernels

Parameters=(11×11×3+1)×96=(363+1)×96=364×96=34,944

### Second Convolutional Layer

* **Input Dimensions:** Output of First Layer (54 × 54 × 96)
* **Kernel Size:** 5 × 5 × 48
* **Number of Kernels:** 256

**Output Dimensions:** 50 × 50 × 256

Parameters=(5×5×96+1)×256=(1,200+1)×256=1,201×256=307,456

**Third Convolutional Layer**

* **Input Dimensions:** Output of Second Layer (50 × 50 × 256)
* **Kernel Size:** 3 × 3 × 256
* **Number of Kernels:** 384

**Output Dimensions:** 48 × 48 × 384

Parameters=(3×3×256+1)×384=(2,304+1)×384=2,305×384=884,640

### Fourth Convolutional Layer

* **Input Dimensions:** Output of Third Layer (48 × 48 × 384)
* **Kernel Size:** 3 × 3 × 192
* **Number of Kernels:** 384

**Output Dimensions:** 46 × 46 × 384

Parameters=(3×3×384+1)×384=(3,456+1)×384=3,457×384=1,327,488

### Fifth Convolutional Layer

* **Input Dimensions:** Output of Fourth Layer (46 × 46 × 384)
* **Kernel Size:** 3 × 3 × 192
* **Number of Kernels:** 256

**Output Dimensions:** 44 × 44 × 256

Parameters=(3×3×384+1)×256=(3,456+1)×256=3,457×256=884,672

### Total Parameters

To calculate the total number of parameters in the convolutional layers:

Total Parameters=34,944+307,456+884,640+1,327,488+884,672=2,439,200

**3.MobileNet**

MobileNet uses depthwise separable convolutions to enhance efficiency and maintain accuracy by reducing parameters and computational complexity.

### Depthwise Separable Convolutions

* **Depthwise Convolution**: Each input channel is convolved separately with a 3×33 \times 33×3 filter, significantly reducing computations.
* **Pointwise Convolution**: A 1×11 \times 11×1 convolution combines the outputs from the depthwise layer across all channels, allowing for complex feature interactions.

### Layer Sizes and Parameter Counts

MobileNet consists of depthwise convolution layers with 3×3 filters and pointwise convolution layers with 1×1 filters. It has fewer than **4.2 million parameters**, significantly fewer than traditional architectures like VGG16 and GoogleNet, while achieving comparable accuracy.

**4.GoogleNet (Inception v1)**

In the GoogLeNet architecture, the Inception module includes the following layers with their sizes and parameters:

1. **1×1 Convolution**:
   * **Layer Size**: 1×1
   * **Filters**: 128
   * **Purpose**: Dimensionality reduction.
2. **3×3 Convolution**:
   * **Layer Size**: 3×3
   * **Filters**: 192
   * **Purpose**: Capture spatial features.
3. **5×5 Convolution**:
   * **Layer Size**: 5×5
   * **Filters**: 96
   * **Purpose**: Enhanced feature extraction.
4. **Max Pooling Layer**:
   * **Kernel Size**: 3×3
   * **Followed by**: 1×1 Convolution
   * **Filters**: 32
   * **Purpose**: Downsample feature maps.

**5.ResNet**

**Layer Dimensions:**

* The architecture consists of a stack of layers organized by feature map sizes:
  + **Feature Map Sizes:** 32×32, 16×16, and 8×8.
  + **Number of Layers:**
    - 1+2n1 + 2n1+2n layers for 32×32 feature maps,
    - 2n2n2n layers for 16×16 feature maps,
    - 2n2n2n layers for 8×8 feature maps.
  + **Filters:**
    - 16 filters for 32×32 maps,
    - 32 filters for 16×16 maps,
    - 64 filters for 8×8 maps.

**Parameter Calculation:**

* The number of parameters for each configuration is:
  + ResNet-20: 0.27M parameters,
  + ResNet-32: 0.46M parameters,
  + ResNet-44: 0.66M parameters,
  + ResNet-56: 0.85M parameters,
  + ResNet-110: 1.7M parameters,
  + ResNet-1202: 19.4M parameters.

**Task 2: Analyze the Following for Each Architecture**

**1.Types of Convolutions Used**

### 1. VGG16

* **Type:** Standard 3×33 \times 33×3 convolutions.
* **Benefit:** Deep network for better feature extraction.

### 2. AlexNet

* **Type:** Standard convolutions (11×11, 5×5,3×3).
* **Benefit:** Captures large and small features effectively.

### 3. MobileNet

* **Type:** Depthwise separable convolutions.
* **Benefit:** Lowers computation for mobile devices.

### 4. GoogleNet

* **Type:** Inception modules with various filter sizes.
* **Benefit:** Extracts features at multiple scales efficiently.

**2.Number of Parameters:**

### 1. VGG16

* **Parameters:** ~138 million
* **Impact:** High cost for training and slow inference.

### 2. AlexNet

* **Parameters:** ~60 million
* **Impact:** Moderate training time; faster than VGG16 but still resource-intensive.

### 3. MobileNet

* **Parameters:** ~4.2 million (varies by version)
* **Impact:** Fast training and inference; efficient for mobile devices.

### 4. GoogleNet

* **Parameters:** ~5 million
* **Impact:** Quick training and efficient inference with good accuracy.

**3.Layer Dimensions:**

### 1. VGG16

* **Input:** 224×224×3
* **Key Layers:**
  + Conv1: 224×224×64
  + Max Pooling: 112×112×128
  + Fully Connected: 4096

### 2. AlexNet

* **Input:** 227×227×3
* **Key Layers:**
  + Conv1: 55×55×96
  + Max Pooling: 13×13x256
  + Fully Connected: 4096

### 3. MobileNet (V1)

* **Input:** 224×224×3
* **Key Layers:**
  + Conv1: 224×224×32
  + Depthwise Conv: 224×224×32224 \times 224 \times 32224×224×32
  + Global Average Pooling: 1×1×1024

### 4. GoogleNet (Inception v1)

* **Input:** 224×224×3
* **Key Layers:**
  + Conv1: 224×224×64
  + Inception Module: 112×112×192
  + Global Average Pooling: 1×1×1024

**4.Problem Solved:**

### 1. VGG16

* **Problem Solved:** Image classification.
* **Performance on ImageNet:** Achieved a top-5 accuracy of **92.7%**.

### 2. AlexNet

* **Problem Solved:** Image classification.
* **Performance on ImageNet:** Achieved a top-5 accuracy of **84.6%**; significantly improved on previous models.

### 3. MobileNet

* **Problem Solved:** Image classification, particularly for mobile and embedded devices.
* **Performance on ImageNet:** Achieved a top-1 accuracy of **70.6%** and a top-5 accuracy of **89.5%** with significantly fewer parameters.

### 4. GoogleNet (Inception v1)

* **Problem Solved:** Image classification and multi-scale feature extraction.
* **Performance on ImageNet:** Achieved a top-5 accuracy of **89.8%**.