Numerical Problem on Calculating CNN Parameters

A Convolutional Neural Network (CNN) is designed with the following architecture:

Layer 1: Convolutional Layer

- Input Image Size: **128 × 128 × 3**
- Number of Filters: **16**
- Kernel Size: **5 × 5**
- Stride: **1**
- Padding: **same**

Layer 2: Max Pooling Layer

- Pool Size: **2 × 2**
- Stride: **2**

Layer 3: Convolutional Layer

- Number of Filters: **32**
- Kernel Size: **3 × 3**
- Stride: **1**
- Padding: **valid**

Layer 4: Fully Connected Dense Layer

- Number of Neurons: **128**

Layer 5: Output Layer

- Number of Neurons: **10** (for classification into 10 classes)

Questions

1. Calculate the **output size** of each layer.
2. Compute the **number of trainable parameters** in each layer.
3. Find the **total number of trainable parameters** in the network.

Step 1: First Convolutional Layer

- **Input Size**: $128 \times 128 \times 3$
- **Filter Size**: 5×5
- **Number of Filters**: 16
- **Stride**: 1
- **Padding**: `"same"` (output size remains the same)

1.1 Output Shape Calculation

Since **padding is "same"**, the formula for output size is:

$$\text{Output Size} = \frac{\text{Input Size} - \text{Kernel Size} + 2 \times \text{Padding}}{\text{Stride}} + 1$$

$$= \frac{128 - 5 + 2(2)}{1} + 1 = 128$$

So, **Output Shape**: $128 \times 128 \times 16$

1.2 Parameter Calculation

Each **filter** has:

$$\text{Weights} = (\text{Kernel Width} \times \text{Kernel Height} \times \text{Input Channels}) + 1\ (\text{bias})$$

$$= (5 \times 5 \times 3) + 1 = 76$$

Since we have **16 filters**, total trainable parameters:

$$16 \times 76 = 1,216$$

Step 2: First Max Pooling Layer

- **Pool Size**: 2×2
- **Stride**: 2
- **Padding**: `"valid"` (no padding)

2.1 Output Shape Calculation

Pooling reduces the size as:

$$\text{Output Size} = \frac{\text{Input Size} - \text{Pool Size}}{\text{Stride}} + 1$$

$$= \frac{128 - 2}{2} + 1 = 64$$

So, **Output Shape**: $64 \times 64 \times 16$

📌 **No trainable parameters** in pooling layers.

Step 3: Second Convolutional Layer

- **Input Shape**: $64 \times 64 \times 16$
- **Filter Size**: 3×3
- **Number of Filters**: 32
- **Stride**: 1
- **Padding**: `"valid"` (no padding)

3.1 Output Shape Calculation

Since **padding is "valid"**, the output size is reduced:

$$\text{Output Size} = \frac{\text{Input Size} - \text{Kernel Size}}{\text{Stride}} + 1$$

$$= \frac{64 - 3}{1} + 1 = 62$$

So, **Output Shape**: $62 \times 62 \times 32$

3.2 Parameter Calculation

Each **filter** has:

$$\text{Weights} = (3 \times 3 \times 16) + 1 = 145$$

Since we have **32 filters**, total trainable parameters:

$$32 \times 145 = 4,640$$

Step 4: Flattening

- **Input Shape**: $62 \times 62 \times 32$

Since the **flatten layer** converts 3D tensors into 1D:

$$\text{Output Shape} = 62 \times 62 \times 32 = 123,008$$

📌 **No trainable parameters in flattening.**

Step 5: Fully Connected Dense Layer

- **Input Neurons**: $123,008$
- **Output Neurons**: 128

5.1 Parameter Calculation

$$\text{Weights} = 123,008 \times 128 = 15,745,024$$

$$\text{Bias} = 128$$

$$\text{Total Parameters} = 15,745,152$$

Step 6: Output Layer

- **Input Neurons**: 128

- **Output Neurons**: 10 (for 10-class classification)

6.1 Parameter Calculation

$$\text{Weights} = 128 \times 10 = 1,280$$

$$\text{Bias} = 10$$

$$\text{Total Parameters} = 1,290$$

Final Table: Summary of Parameters

Layer	Output Shape	Trainable Parameters
Conv2D (16 filters)	(128, 128, 16)	**1,216**
MaxPooling2D	(64, 64, 16)	0
Conv2D (32 filters)	(62, 62, 32)	**4,640**
Flatten	(123,008)	0
Dense (128 neurons)	(128)	**15,745,152**
Dense (Output, 10 neurons)	(10)	**1,290**
Total Parameters	-	**15,752,298**

Final Answer

Total Trainable Parameters = 15,752,298

This CNN model learns **over 15.75 million parameters** through **backpropagation and gradient descent**.
