Solution:

Part (a)

Since the inputs and outputs are fully connected by network parameters $w(.)$, the outputs y_{ijk} can be expressed as a linear combination of all inputs $x_{i'j'k'}$:

由于输入和输出通过网络参数完全连接 $w(.)$，输出 y_{ijk} 可以表示为所有输入的线性组合 $x_{i'j'k'}$：

$$y_{ijk} = \sum_{i'=1}^{I}\sum_{j'=1}^{J}\sum_{k'=1}^{K_1} w(i,j,k,i',j',k') \cdot x_{i'j'k'}$$

where $w(i,j,k,i',j',k')$ are the learnable parameters that connect input $x_{i'j'k'}$ to output y_{ijk}.在哪里 $w(i,j,k,i',j',k')$ 是连接输入的可学习参数 $x_{i'j'k'}$ 输出 y_{ijk} 。

Number of parameters: Each output y_{ijk} requires a unique weight for each input $x_{i'j'k'}$, so the number of parameters is the product of all possible combinations of i, j, k for the outputs and i', j', k' for the inputs:

参数数量：每个输出 y_{ijk} 每个输入都需要唯一的权重 $x_{i'j'k'}$ ，因此参数的数量是所有可能组合的乘积 i, j, k 对于输出和 i', j', k' 对于输入：

$$\text{Number of parameters} = I \times J \times K_2 \times I \times J \times K_1 = I^2 \times J^2 \times K_1 \times K_2$$

Number of multiplications: For each output y_{ijk}, we need to compute a weighted sum over all inputs $x_{i'j'k'}$. This requires:

乘法次数：对于每个输出 y_{ijk} ，我们需要计算所有输入的加权和 $x_{i'j'k'}$ 。这需要：

$$\text{Number of multiplications} = I \times J \times K_2 \times (I \times J \times K_1) = I^2 \times J^2 \times K_1 \times K_2$$

Part (b)

If the outputs are computed by applying K_2 spatial filters of size $3 \times 3 \times K_1$, denoted by $h(.)$, the output y_{ijk} can be expressed as:

如果通过应用计算输出 K_2 大小的空间滤波器 $3 \times 3 \times K_1$ ，表示为 $h(.)$ ，输出 y_{ijk} 可以表示为：

$$y_{ijk} = \sum_{u=-1}^{1}\sum_{v=-1}^{1}\sum_{k'=1}^{K_1} h(u,v,k,k') \cdot x_{(i+u)(j+v)k'}$$

where $h(u,v,k,k')$ represents the weight of the filter centered at (i,j) in the spatial dimensions for the k-th output channel and the k'-th input channel.

在哪里 $h(u,v,k,k')$ 表示以中心为中心的过滤器的重量 (i,j) 在空间维度上 k -th 输出通道和 k' -th 输入通道。

Number of parameters: Each filter for each output channel k has $3 \times 3 \times K_1$ parameters, and we have K_2 filters, so:

参数数量：每个输出通道的每个滤波器 k 有 $3 \times 3 \times K_1$ 参数，我们有 K_2 过滤器，所以：

$$\text{Number of parameters} = K_2 \times 3 \times 3 \times K_1 = 9 \times K_1 \times K_2$$

Number of multiplications: For each output y_{ijk}, the filter performs $3 \times 3 \times K_1$ multiplications. Since we have $I \times J \times K_2$ outputs:

乘法次数：对于每个输出 y_{ijk} ，滤波器执行 $3 \times 3 \times K_1$ 乘法。既然我们有 $I \times J \times K_2$ 输出：

$$\text{Number of multiplications} = I \times J \times K_2 \times (3 \times 3 \times K_1) = 9 \times I \times J \times K_1 \times K_2$$

Part (c)

Key Differences:主要区别：

1. **Parameter Efficiency and Training Complexity:参数效率和训练复杂度：**
   - In Part (a), the fully connected network has $I^2 \times J^2 \times K_1 \times K_2$ parameters, making it very large and computationally expensive to train, especially as I and J increase. This results in high memory requirements and a higher likelihood of overfitting.

     在(a)部分中，全连接网络有 $I^2 \times J^2 \times K_1 \times K_2$ 参数，使得训练非常大并且计算成本昂贵，特别是当 I 和 J 增加。这会导致高内存需求和更高的过度拟合可能性。
   - In Part (b), the convolutional network has only $9 \times K_1 \times K_2$ parameters, which is significantly fewer. This makes the network more efficient to train and less prone to overfitting.

     在(b)部分中，卷积网络只有 $9 \times K_1 \times K_2$ 参数，明显更少。这使得网络的训练效率更高，并且不太容易出现过度拟合。

2. **Spatial Invariance and Generalization:空间不变性和泛化：**
   - The fully connected structure in Part (a) does not take advantage of the spatial structure of the input, meaning it treats all input positions independently. This can result in poor generalization to images with slight spatial variations.

     (a) 部分中的全连接结构没有利用输入的空间结构，这意味着它独立处理所有输入位置。这可能会导致对具有轻微空间变化的图像的泛化能力较差。
   - In Part (b), the convolutional structure applies local filters across the image, allowing it to recognize patterns regardless of their position. This spatial invariance improves generalization and is more suitable for image processing tasks.

     在 (b) 部分中，卷积结构在图像上应用局部滤波器，使其能够识别图案，无论其位置如何。这种空间不变性提高了泛化能力，更适合图像处理任务。