



✦ 一、卷积 + 池化的回顾

了解通道之前必须搞懂卷积和池化到底在干啥。

◇ 卷积 (Convolution)

✓ 作用：

- 提取特征（边缘、纹理、形状等）
- 增加通道数（特征维度）

✓ 怎么做：

- 使用一个小卷积核（比如 3×3 ）在图像上滑动
- 每次滑动都对局部区域计算“加权和”
- 卷积核就像是一个“特征提取器”

🔗 举个例子：

原图像 ($640 \times 640 \times 3$)

↓ 经过一个 3×3 卷积 + BN + 激活函数

输出 ($320 \times 320 \times 64$)：尺寸减半，通道数从 3 → 64

👁 池化 (Pooling)

✓ 作用：

- 降低特征图尺寸（减小计算量）
- 保留主要特征（忽略细节）

✓ 两种常见方式：

类型	方式
MaxPool	每个区域取最大值（突出显著特征）

类型	方式
AvgPool	每个区域取平均值（平滑化）

🔗 举个例子：

输入特征图（320×320×64）
↓ 经过 2×2 MaxPooling，stride=2
输出变为（160×160×64）：**尺寸减半，通道数不变**

❖ 问题：卷积为什么会让图像尺寸变小？因为下采样设置！

关键影响尺寸的三个因素：

参数	含义
Kernel size	卷积核大小（例如 3×3）
Stride	步长，卷积核每次滑动的距离
Padding	边缘填充，是否在输入图像周围补零

❖ 卷积输出尺寸公式（2D 情况）：

对输入尺寸为 $H_{in} \times W_{in}$ ，输出尺寸为：

$$H_{out} = [(H_{in} + 2P - K) / S] + 1$$
$$W_{out} = [(W_{in} + 2P - K) / S] + 1$$

其中：

- K = 卷积核大小（如 3）

- S = 步长 (stride)
 - P = Padding (填充)
-

🔍 举例分析：

❓ 情况 1：Stride=1, Padding=0, Kernel=3

- 输入图像：5×5
- 卷积核大小：3×3
- 步长：1
- 无填充

输出尺寸：

$$= [(5 + 0 - 3) / 1] + 1 = 3$$

✓ 输出变为 3×3，每边少了 2 个像素（总共少 2，因卷积核要“贴住中心”）

❓ 情况 2：Stride=1, Padding=1, Kernel=3 (✓ 最常见)

- 输入图像：5×5
- 填充：每边补一圈0
- 输出尺寸：

$$= [(5 + 2 - 3) / 1] + 1 = 5$$

✓ 尺寸保持不变

所以只要 padding 设置得当 ($P = (K - 1)/2$)，stride=1 时尺寸是不会变化的！

❖ 情况 3: Stride=2, Padding=1, Kernel=3

- 输入图像: 6×6
- 输出尺寸:

$$= [(6 + 2 - 3) / 2] + 1 = (5/2) + 1 = 3.5 \rightarrow \text{向下取整} = 3$$

✓ 输出变为 3×3

也就是说:

Stride	Padding	尺寸变化
1	足够填充	尺寸不变
2	常规填充	尺寸减半!

✓ 回答你的问题:

“卷积为什么尺寸减半? stride=1 不应该只减一点点吗?”

答:

- 如果 stride=1 且 padding 合理, 尺寸不会减半
- 尺寸减半通常是因为用了 stride=2, 这是故意设计的, 为了“下采样”, 即:
通过步长=2的卷积替代池化 (MaxPool) 来降低分辨率 + 增加语义层次

👁👁 YOLOv7 是怎么设置的?

在 YOLOv7 的结构中, 通常卷积层配的是:

- kernel=3
- stride=2

- padding=1

➡ 这是 **常见的下采样卷积设置**，它会让图像尺寸直接减半！

✦ 小总结

场景	会不会改变尺寸？	说明
stride=1, padding=(k-1)/2	✗ 不变	常见做法，保持尺寸
stride=1, no padding	✓ 变小	每层减少一点
stride=2, padding=1	✓ 减半	下采样，特意设计

❖ 二、什么是通道数（Channel）？

在卷积神经网络中，**通道数**可以理解为**特征图的“深度”维度**。

类比理解：

- 一张彩色图像是 640×640×3：
 - 3 个通道对应 **RGB 三原色**
- 在神经网络中，经过卷积处理后图像变成了“多通道”形式：
 - 比如 160×160×128，意思是：
 - 高度：160
 - 宽度：160
 - 通道数：128（也可以理解为 128 张不同的特征图）

✦ 每个通道都学会检测不同的东西：

- 第一个通道可能“关注边缘”
- 第二个通道可能“检测纹理”
- 第十个通道可能“检测眼睛形状”
- ...

❖ 为什么通道数会随着层级增加？

这是深度 CNN 网络设计中的一个通用原则，背后逻辑非常重要：

层级	特征图大小	通道数	原因
浅层	大	少（如 32）	捕捉简单的细节（边缘、纹理）
中层	中	多（如 128）	抽象出形状、局部结构
深层	小	更多（如 512~1024）	高度抽象的语义（比如：人脸、动物）

通道数增加的 3 个核心原因：

1. 空间信息减少了，需要用“深度”补回来

- 卷积 + 池化会让特征图“变小”（空间分辨率下降）
- 为了不丢信息，我们让每个位置保存更多维度的表示 → 增加通道

2. 抽象语义更复杂，表示能力要增强

- 深层网络提取的是“人脸”、“汽车”、“猫的耳朵”等高级语义
- 需要更多通道来表示更多维度的语义信息

3. 现代网络的设计经验

- 很多经典网络（ResNet、VGG、YOLO）都遵循：

“分辨率越小 → 通道越多”

📌 YOLOv7 中通道变化举例

层级	特征图尺寸	通道数
输入图像	640×640	3

层级	特征图尺寸	通道数
第一层 CBS	320×320	64
第二层 ELAN	160×160	128
第三层 ELAN	80×80	256
第四层 ELAN	40×40	512
第五层 ELAN	20×20	1024

你会发现：特征图越“小”，通道越“多”。

✈ 总结一句话：

通道数 = 模型对“信息维度”的表达能力。

随着网络层数加深，图像的空间分辨率减小，但语义表达需求增加，所以我们用更多通道数来“丰富表示”。

而三、通道可视化示意

我们来做一个小图示，用于帮助你“看见”什么是通道数增加。

🔍 通道直观理解图（简化版）：

输入图像	尺寸：640×640×3 (R,G,B)
------	----------------------

↓ 卷积（提取边缘、颜色等）

特征图-1	尺寸：320×320×64
- 边缘1	
- 纹理1	
- 曲线检测	
- ... 共64种特征	

↓ 卷积 + 下采样

特征图-2	尺寸：160×160×128
- 更深层次的图形特征	

↓

特征图-3	尺寸：80×80×256
- 区域、局部形状、轮廓	

↓

特征图-4	尺寸：40×40×512
- 对象语义（例如“这像个眼睛”）	

↓

特征图-5	尺寸：20×20×1024
- 高层语义（例如“这是个猫”）	

🔍 每一层的“通道”就像是不同的滤镜：

- 通道1：检测水平边缘
- 通道2：检测垂直边缘
- 通道10：检测圆形区域

- 通道50：检测颜色块
- 通道100：检测类似“猫脸”的结构

💡 总结

概念	意义
卷积	提取局部特征，增强语义， 增加通道数
池化	缩小尺寸，保留主要信息， 保持通道数不变
通道	表示每个空间位置的特征维度， 越深表示能力越强

如果你想，我还可以做一张图展示“卷积 + 池化 + 通道可视化”的一整套流程图，像是一个从图像→特征→语义→检测结果的流水线。要不要？ 😊