

# 7.图像分割

分类



猫

不考虑空间位置

语义分割



草, 猫, 树, 天空

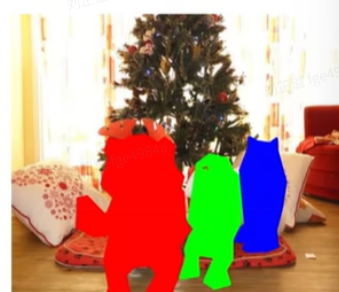
像素的类别

目标检测



狗, 狗, 猫

实例分割

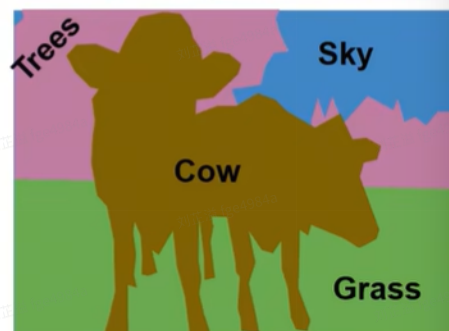
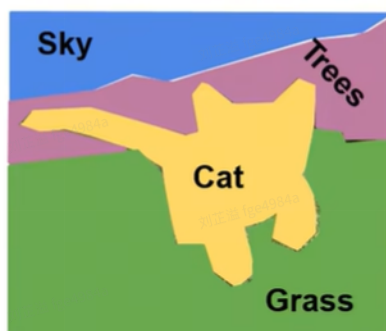


狗, 狗, 猫

多目标

## 语义分割

- 给每个像素分配类别标签；不区分实例，只考虑像素类别

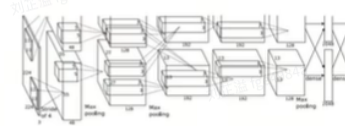
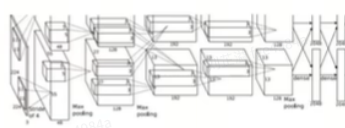
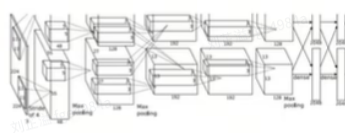
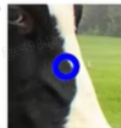
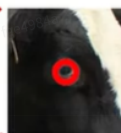


## 语义分割思路：滑动窗口

提取区域

利用CNN对中心点像素分类

Full image



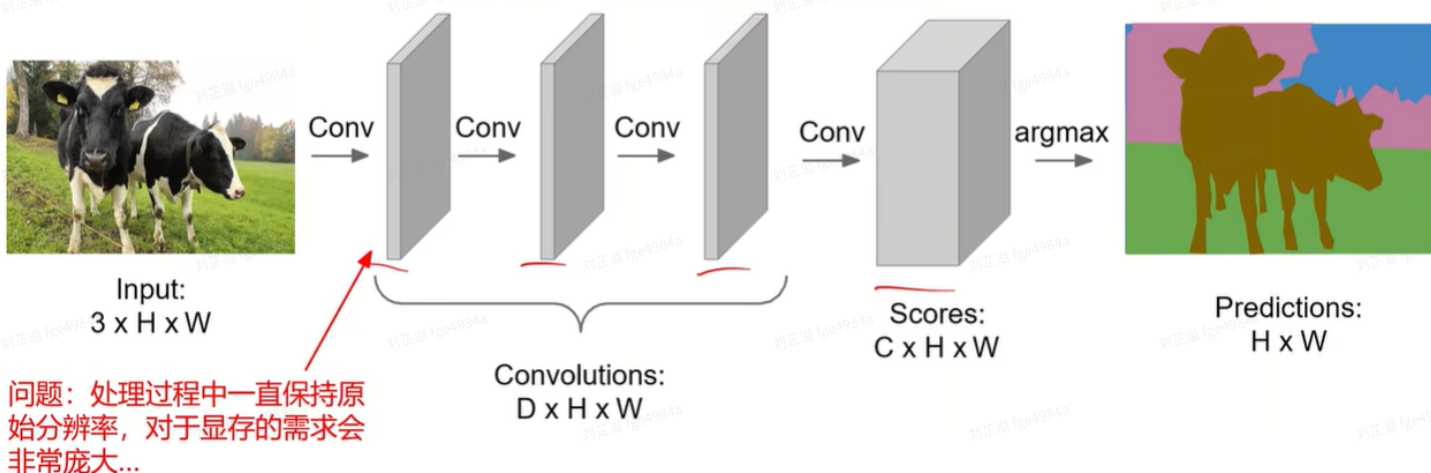
牛  
牛  
草

问题: 效率太低! 重叠区域的特征反复被计算

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013  
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

语义分割思路: 全卷积

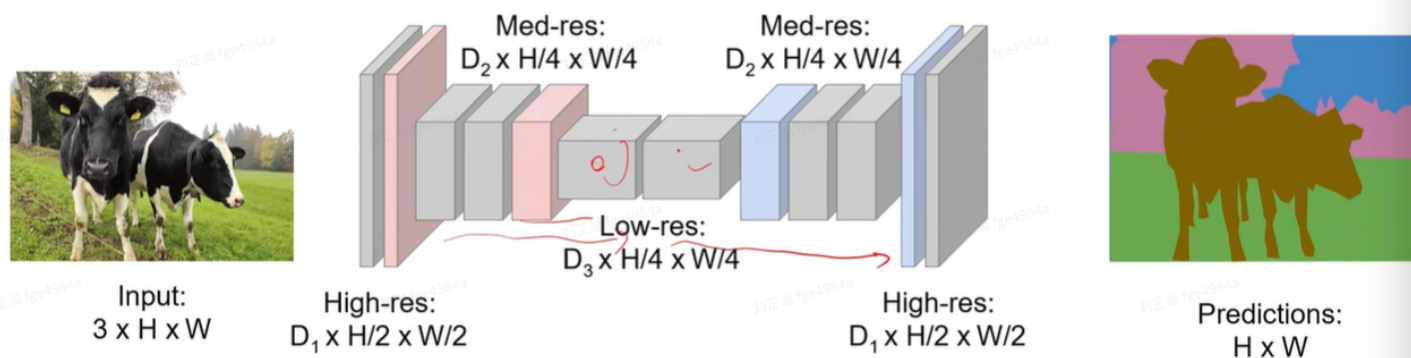
解决方案: 让整个网络只包含卷积层,  
一次性输出所有像素的类别预测。



问题: 处理过程中一直保持原始分辨率, 对于显存的需求会非常庞大...

- 让整个网络只包含卷积层; 并在网络中嵌入下采样与上采样过程

解决方案: 让整个网络只包含卷积层,  
并在网络中嵌入下采样与上采样过程。

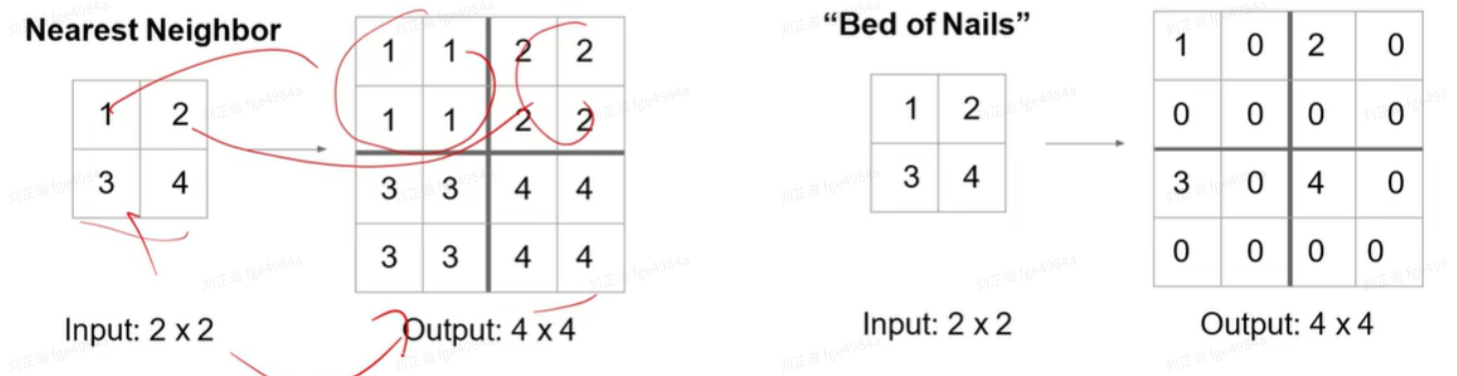


## 下采样

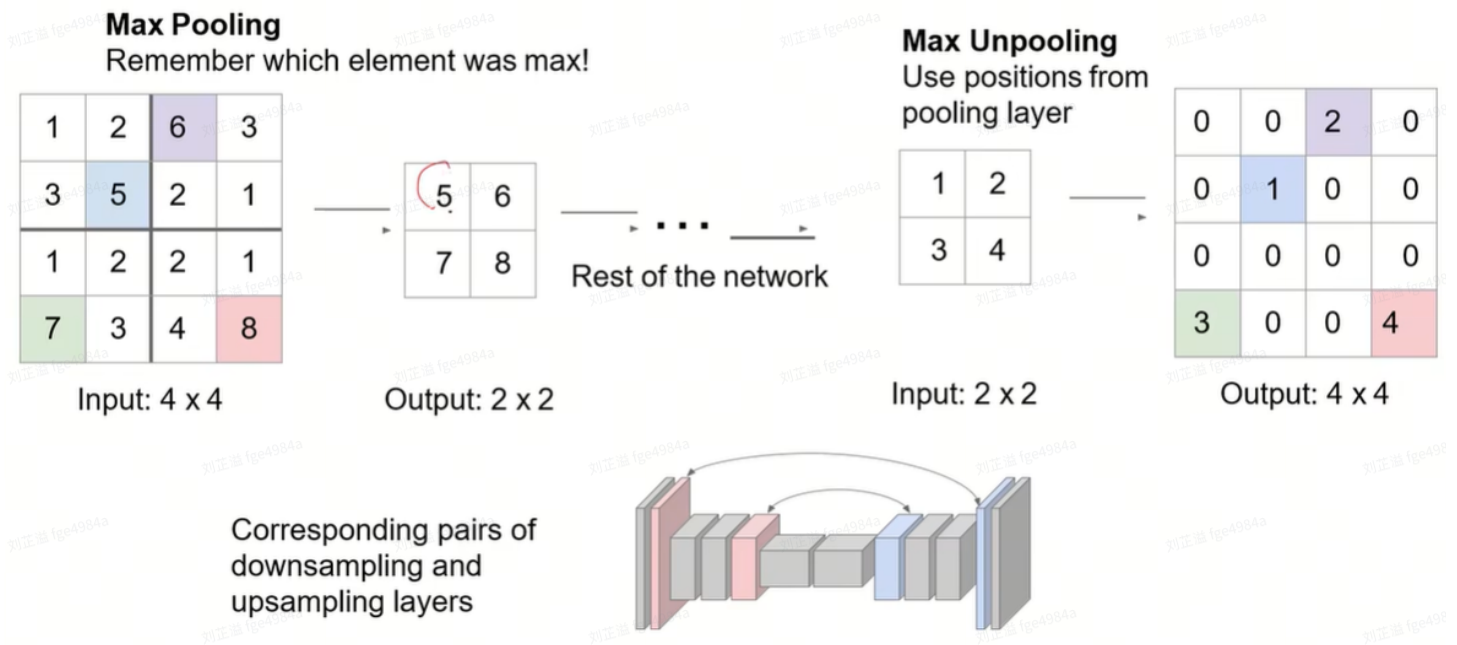
- Pooling, strided convolution

## 上采样

反池化操作: "Unpooling"

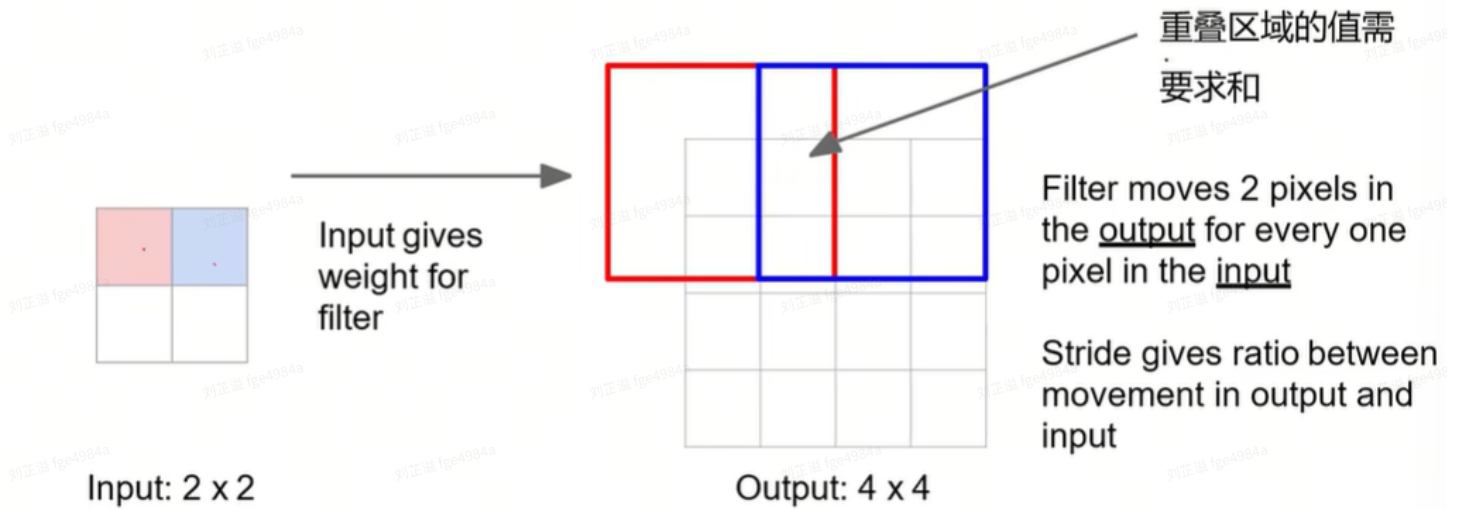


反池化操作: "Max Unpooling"



可学习的上采样: 转置卷积(Transpose Convolution)

## 3 x 3 转置卷积 (transpose convolution) , stride 2 pad 1



### 卷积与矩阵相乘

- 转置卷积的参数都是神经网络自己学习的

## 卷积与矩阵相乘 (一维例子)

将卷积写为矩阵乘法

$$\vec{x} * \vec{a} = X \vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & x & y & x & 0 & 0 \\ 0 & 0 & x & y & x & 0 \\ 0 & 0 & 0 & x & y & x \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$$

*Handwritten note: 6x1*

例子: 1D 卷积, 卷积核尺寸=3, 步长=1, 零填充=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ cz + dy \\ dz \end{bmatrix}$$

*Handwritten notes: 6x4, 4x1, 6x1*



## 将卷积写为矩阵乘法

$$\vec{x} * \vec{a} = X \vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & 0 & x & y & x & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

例子: 1D 卷积, 卷积核尺寸=3, 步长=2, 零填充=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$