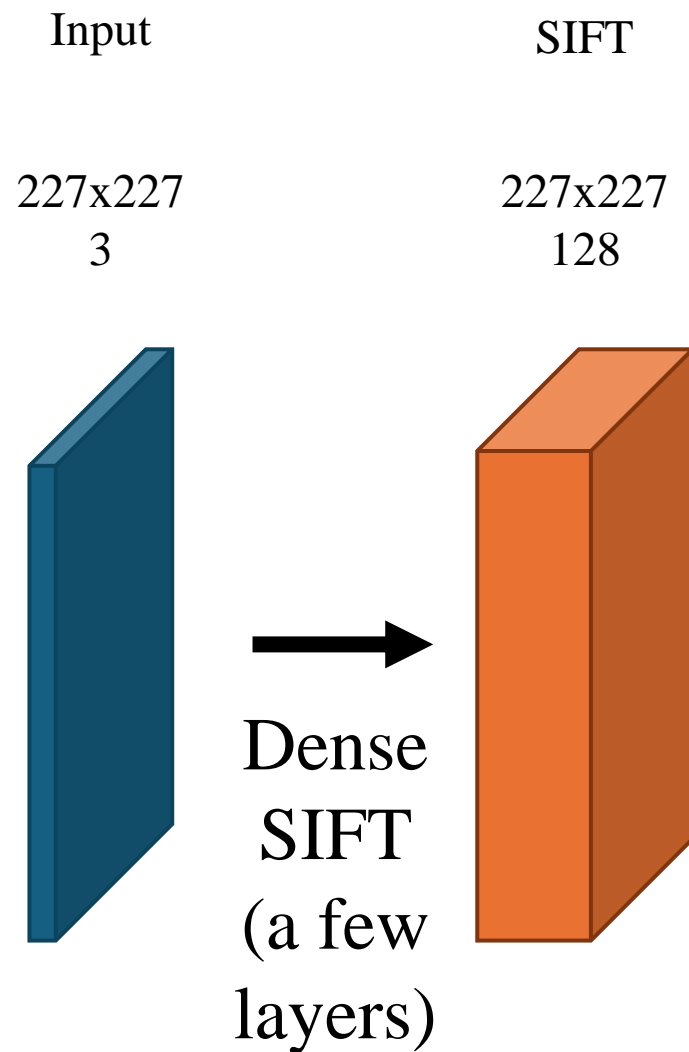
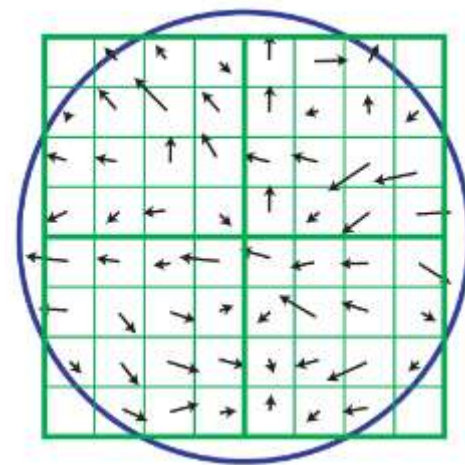


# 回顾：卷积神经网络

# 回顾：分类识别 传统模型 与 深度学习

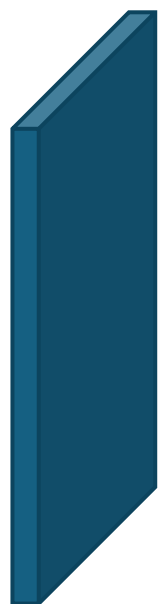


回顾：可以基于图像梯度直方图计算描述子。在每个像素处密集地执行。

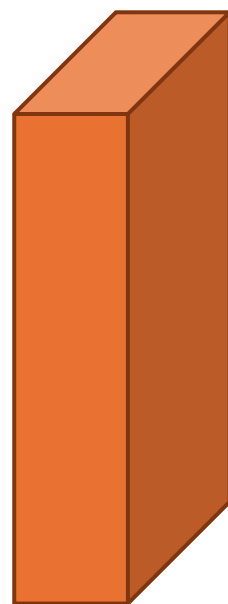


# 回顾：分类识别 传统模型 与 深度学习

Input	SIFT	Bag of Words
$227 \times 227$ 3	$227 \times 227$ 128	HxW #codewords

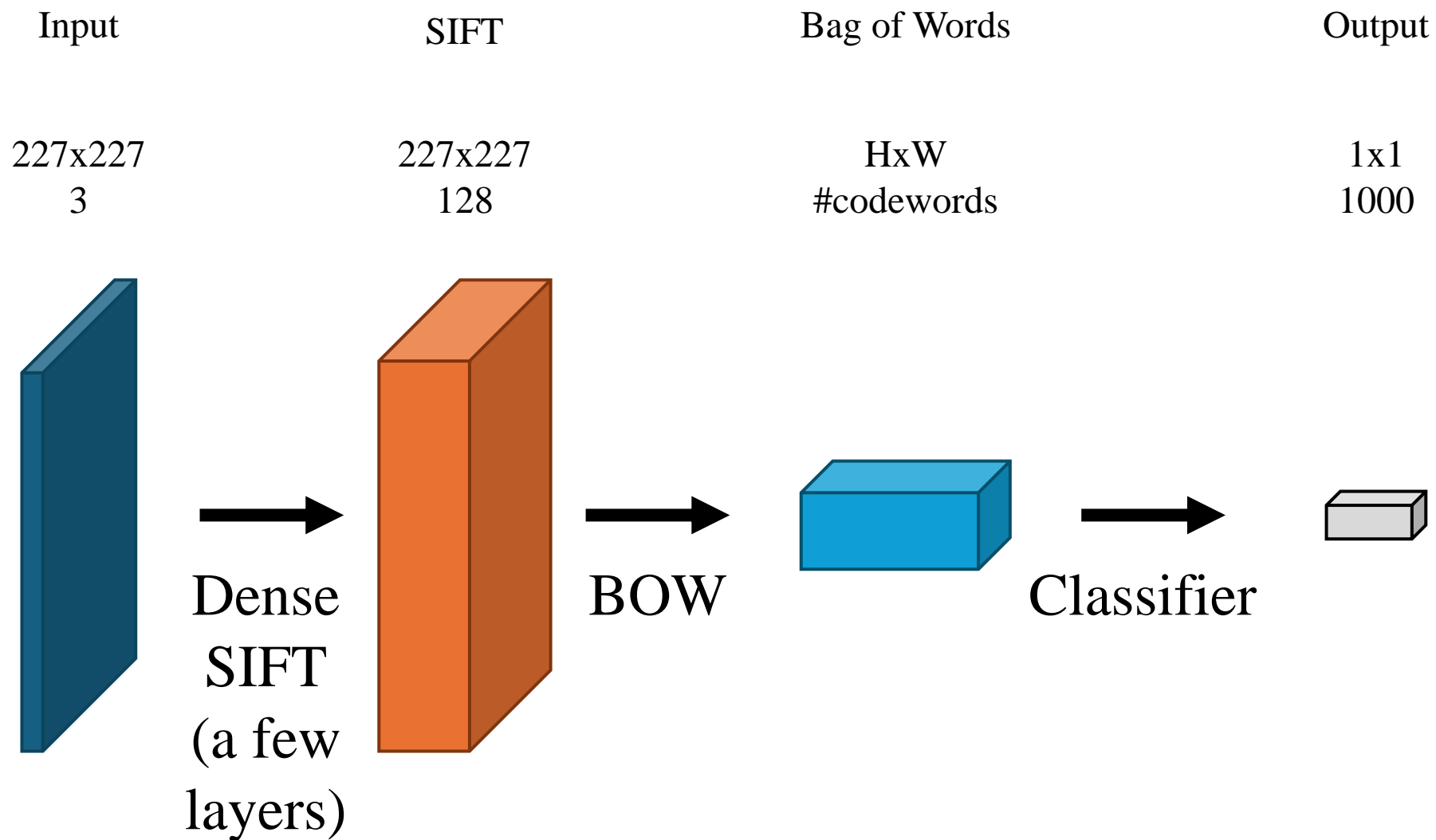


→  
Dense  
SIFT  
(a few  
layers)

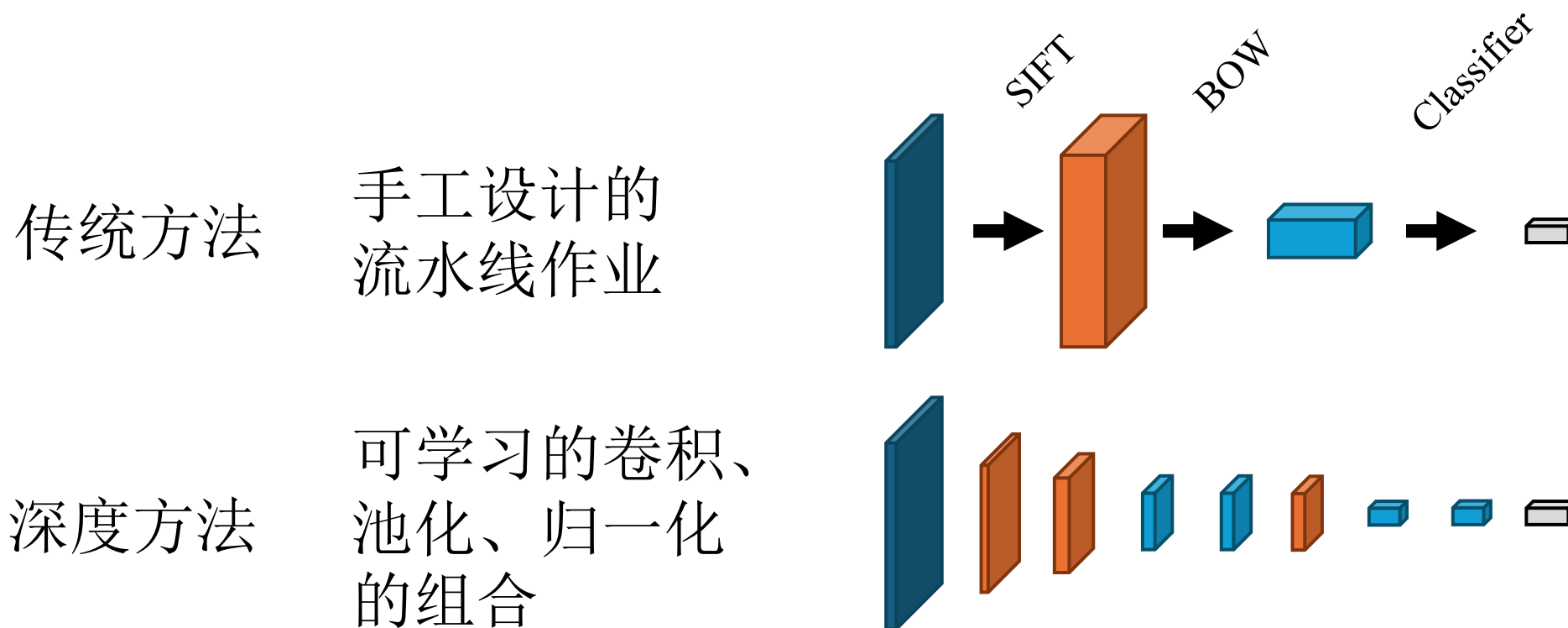


可以对SIFT执行类似词袋的技术(bag-of-words) 同时考虑到描述子在图像中的空间位置,对图像的局部区域进行编码, 并改进图像分类或检索的性能。

# 回顾：分类识别 传统模型 与 深度学习



# 回顾：分类识别 传统模型 与 深度学习



## 有什么区别？

传统步骤中的各个组件通常是独立设计和优化的，不会“相互交流”或从数据中学习大量参数。相比之下，深度学习方法中的层是通过学习数据自动优化参数的，各层之间会有信息的交流。

# 语义分割

# 回顾：计算机视觉任务

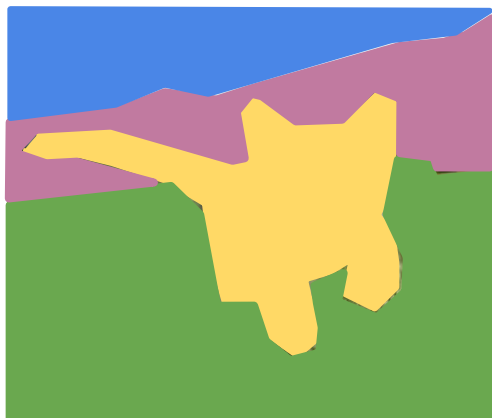
分类



**CAT**

无空间限定

语义分割



**GRASS, CAT,**  
**TREE, SKY**

像素级预测

物体检测



**DOG, DOG, CAT**

多物体

实例分割



**DOG, DOG, CAT**

# 语义分割

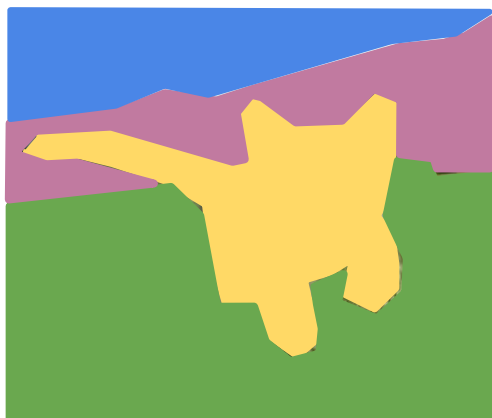
分类



CAT

无空间限定

语义分割



GRASS, CAT,  
TREE, SKY

像素级预测

物体检测



DOG, DOG, CAT

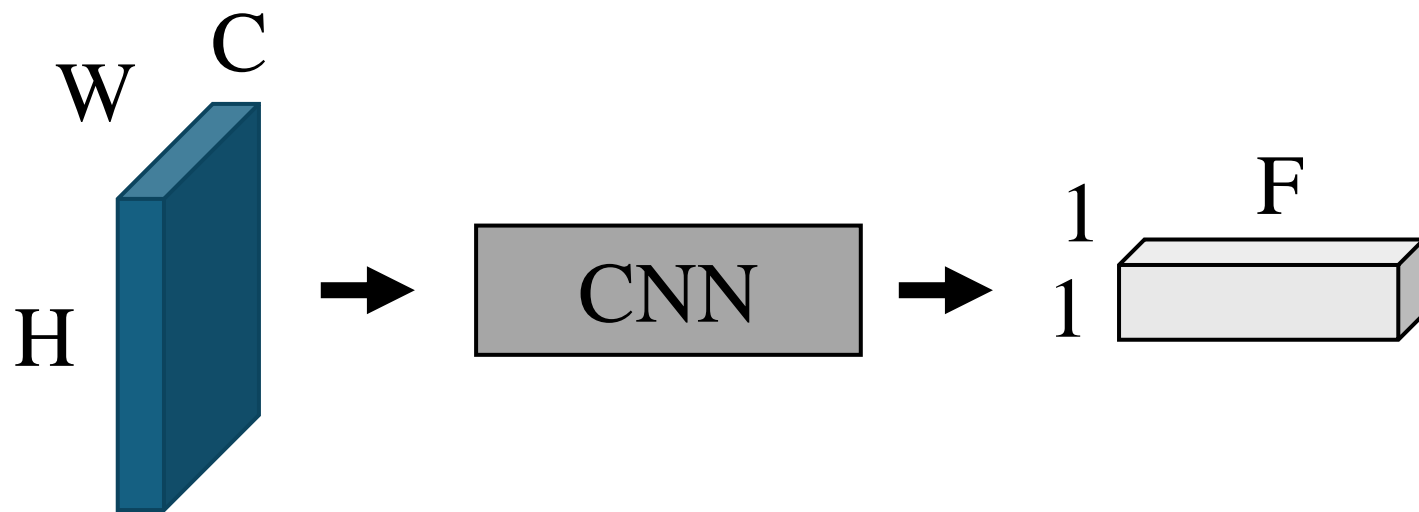
多物体

实例分割



DOG, DOG, CAT

## 回顾：之前的CNN



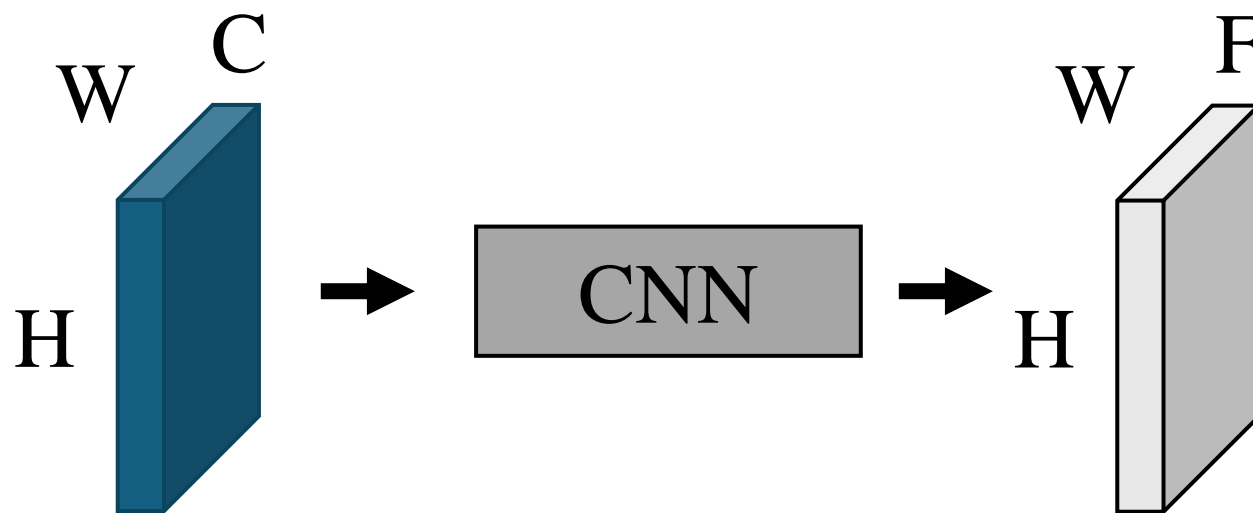
把  $H \times W$  大小的图像转换为  $F$ -维 向量

这是猫吗？

摄像机距离图像多远？

这是一张AI生成图像吗？

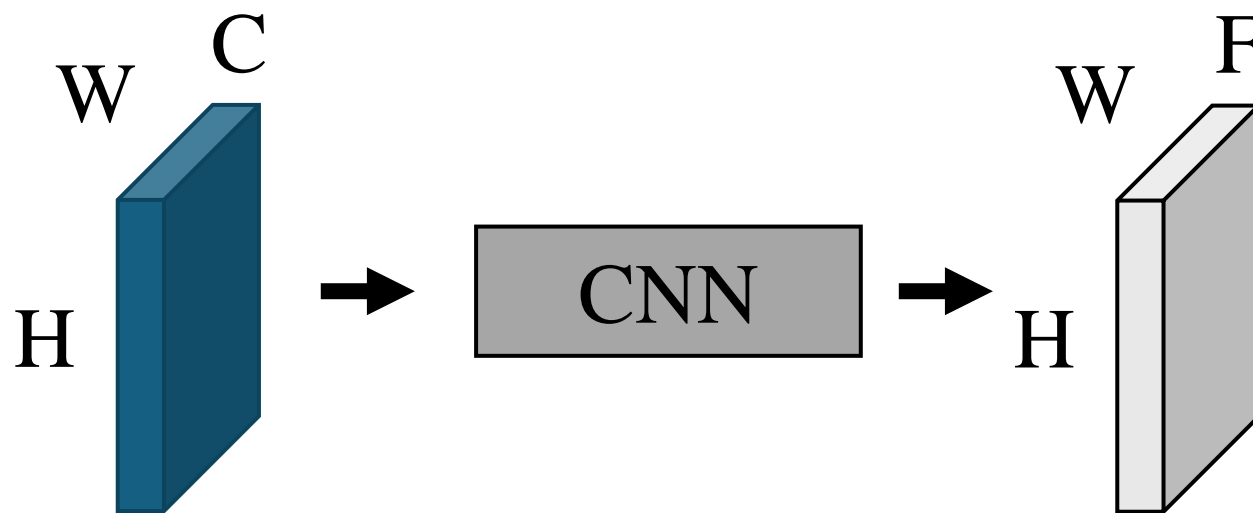
## 现在的任务



把  $H \times W$  大小的图像的每个像素转换为  $F$ -维 向量

哪些像素呈现一只猫?  
每个像素距离摄像机多远?  
哪些像素是伪造的?

# 语义分割



## 今天的主题

- 为每个像素提供一个 $F$ 维的向量
  - 用于预测 $F$ 个类别的概率
- 在 每个像素上计算损失，回传梯度

# 语义分割

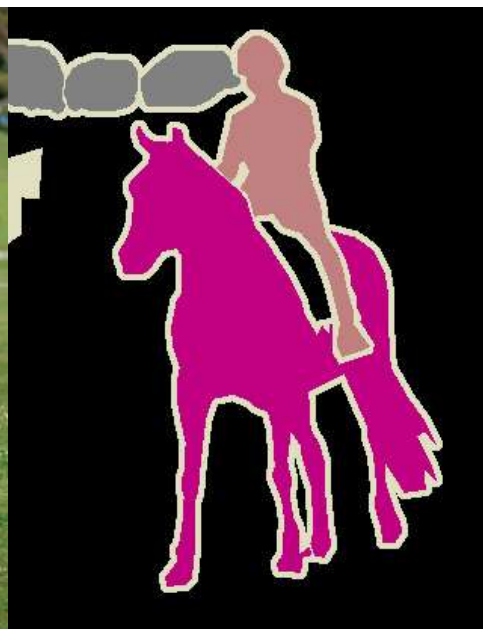
每个像素都有它的标签，包括 **背景**，和未知  
不同标签用不同颜色表示

不会区分相同类别下不同的实体（semantic vs. instance）

Input



Label



Input



Label



# 语义分割

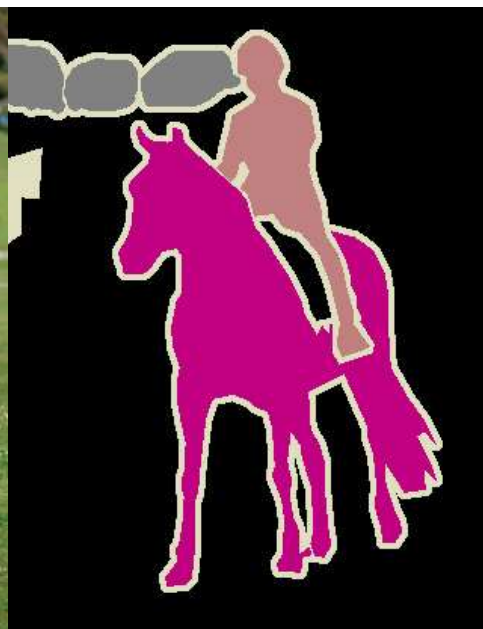
“Semantic” 语义: 不要太在意这个没有什么特殊含义的词  
lol.

区分不同种类的物体, 记得其和 instance 实体的区别.

Input



Label



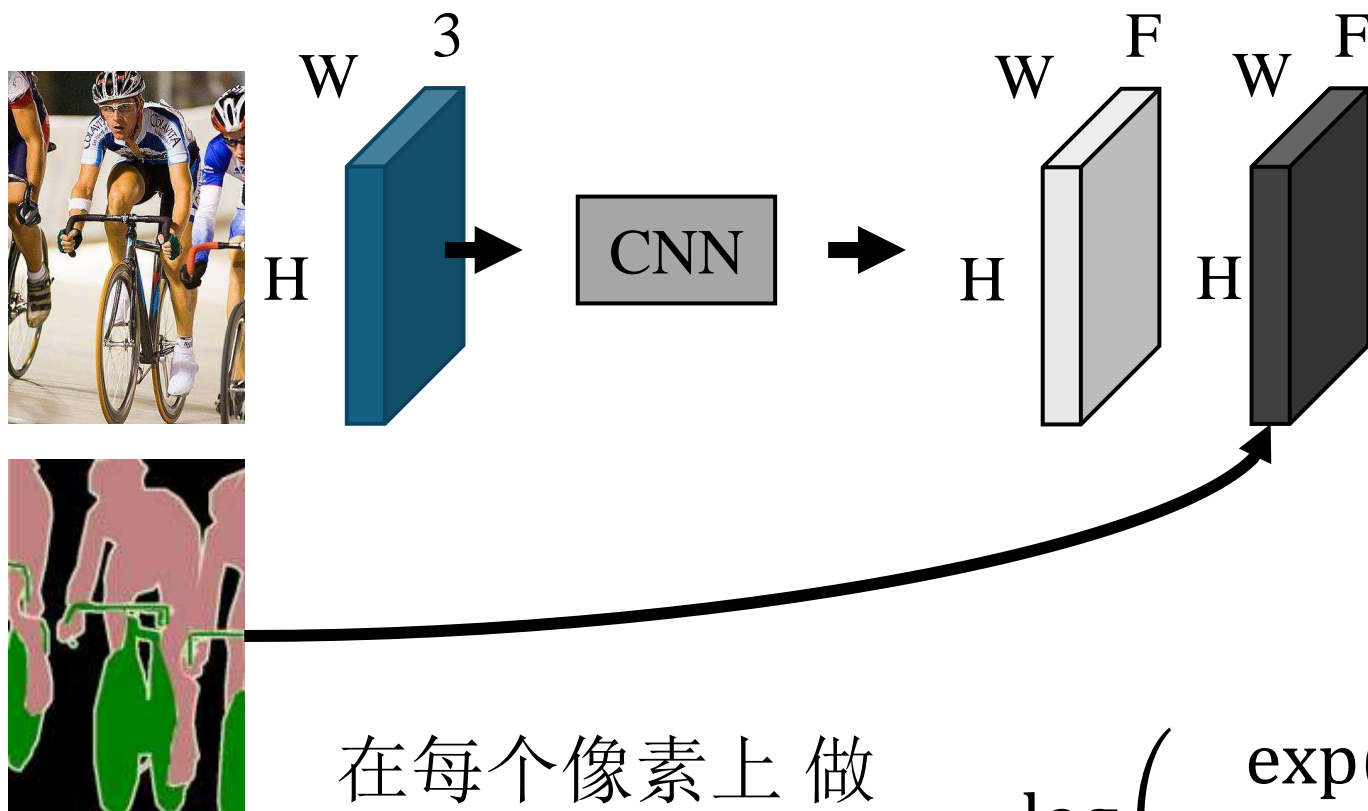
Input



Label



# 语义分割

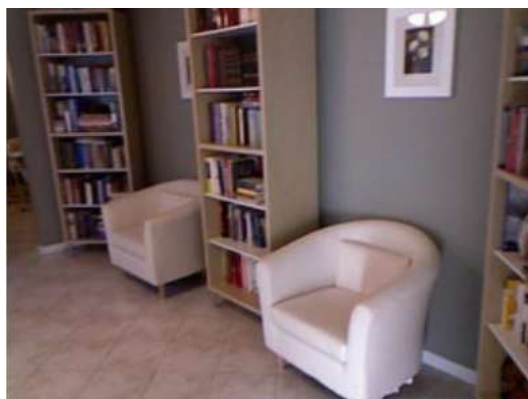


在每个像素上做  
F-分类交叉熵损失：
$$-\log\left(\frac{\exp((Wx)_{y_i})}{\sum_k \exp((Wx)_k)}\right)$$

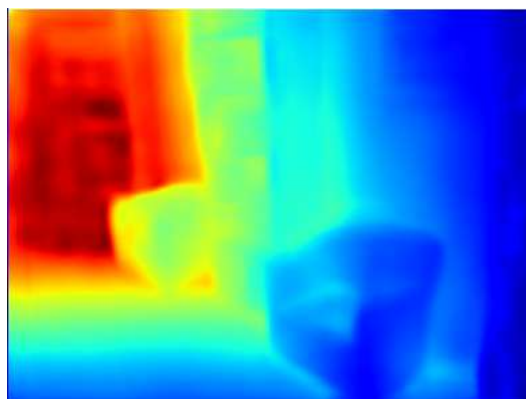
## 其他任务——深度预测

给定深度图标注，训练回归网络 优化  $\|z_i - \hat{z}_i\|$ ，对于像素 $i$ ， $z_i$  为真实深度  $\hat{z}_i$  为预测深度).

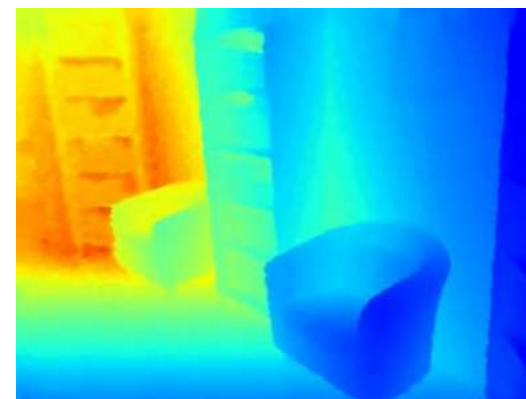
Input  $H \times W \times 3$   
RGB Image



Output  $H \times W \times 1$   
Depth Image



True  $H \times W \times 1$   
Depth Image

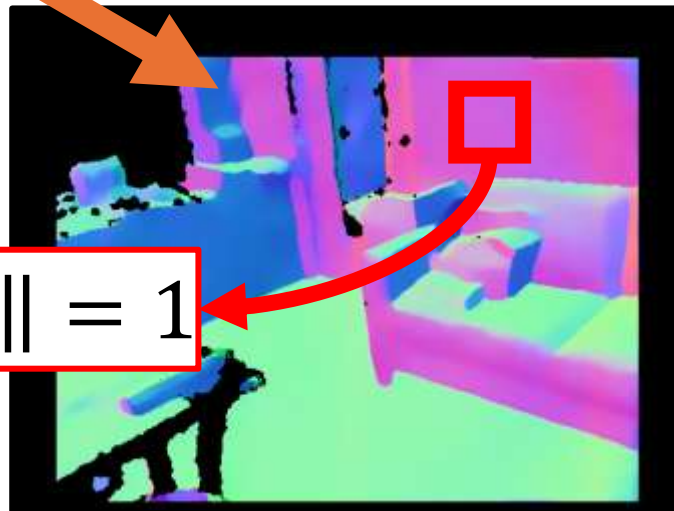


# 其他任务——Surface Normals 表面预测



$$\mathbf{n} = [n_x, n_y, n_z], \|\mathbf{n}\| = 1$$

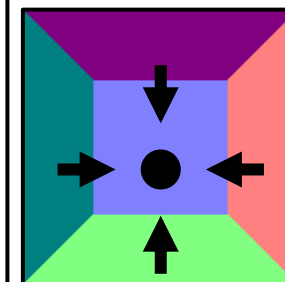
Color Image



Normals



Room



Legend

# 表面预测

训练网络最小化  $\|n_i - \hat{n}_i\|$  ,  
 $n_i$  是真实法向量,  $\hat{n}_i$  是对像素  $i$  的预测法向量.

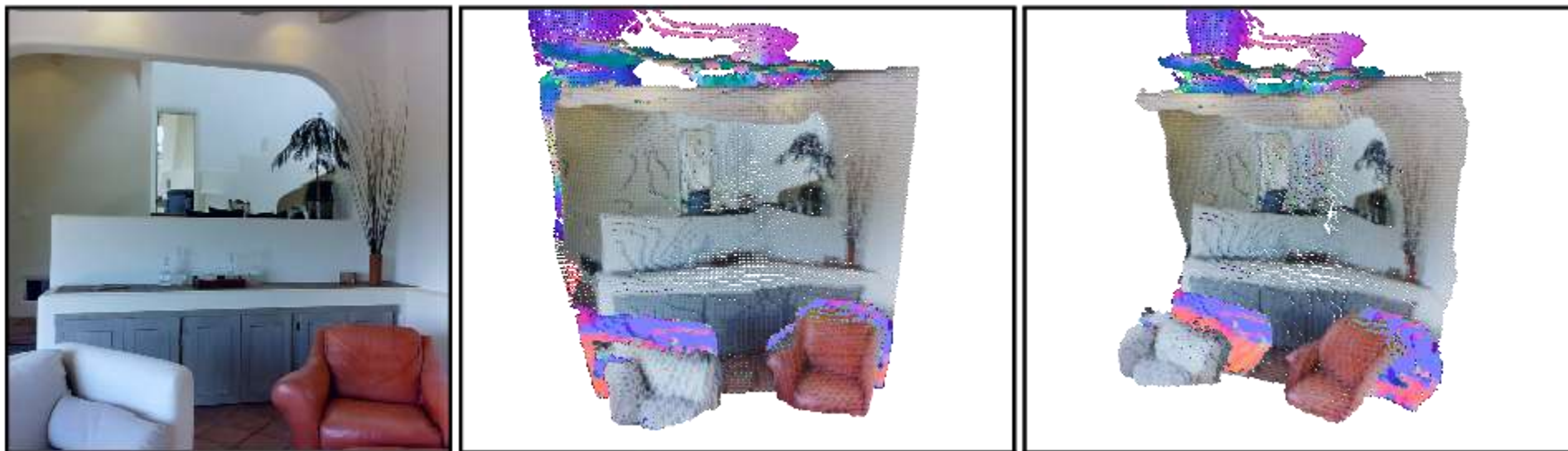
Input: HxWx3  
RGB Image



Output: HxWx3  
Normals



# 3D 重建



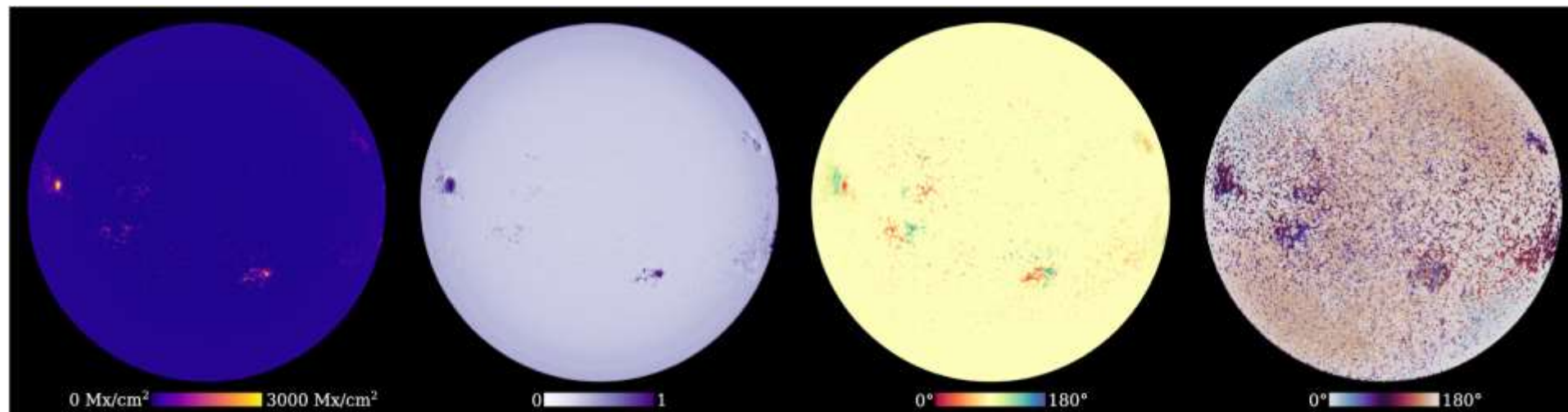
Result credit: N. Kulkarni, J. Johnson, D.F. Fouhey, *What's Behind The Couch: Directed Ray Distance Functions for 3D Reconstruction*. ???, 2022.

# 人体姿态估计

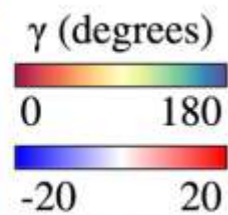
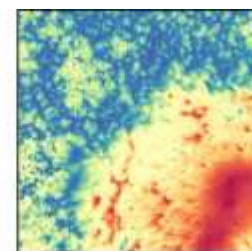
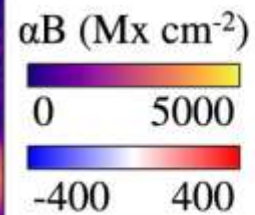
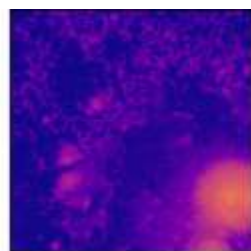
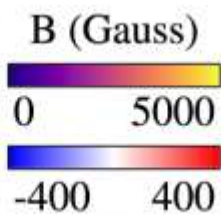
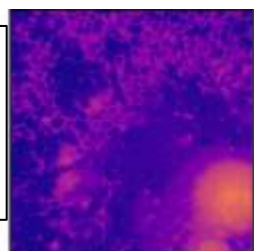


Result credit: Z. Cao et al. *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. CVPR 2017.

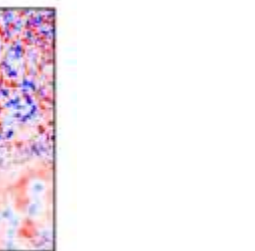
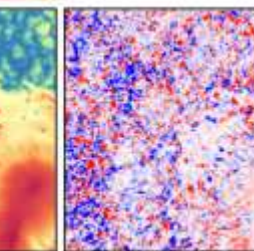
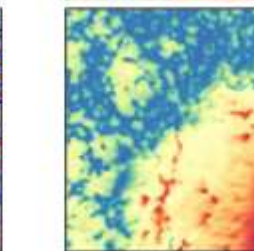
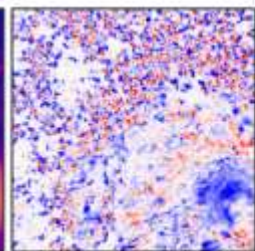
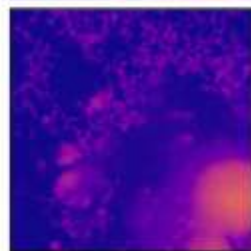
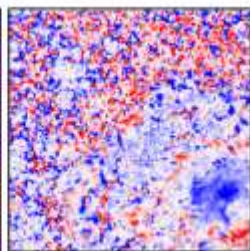
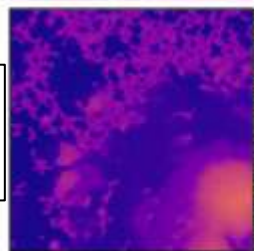
# 太阳



Normal  
Physics  
Model



Deep  
Learning

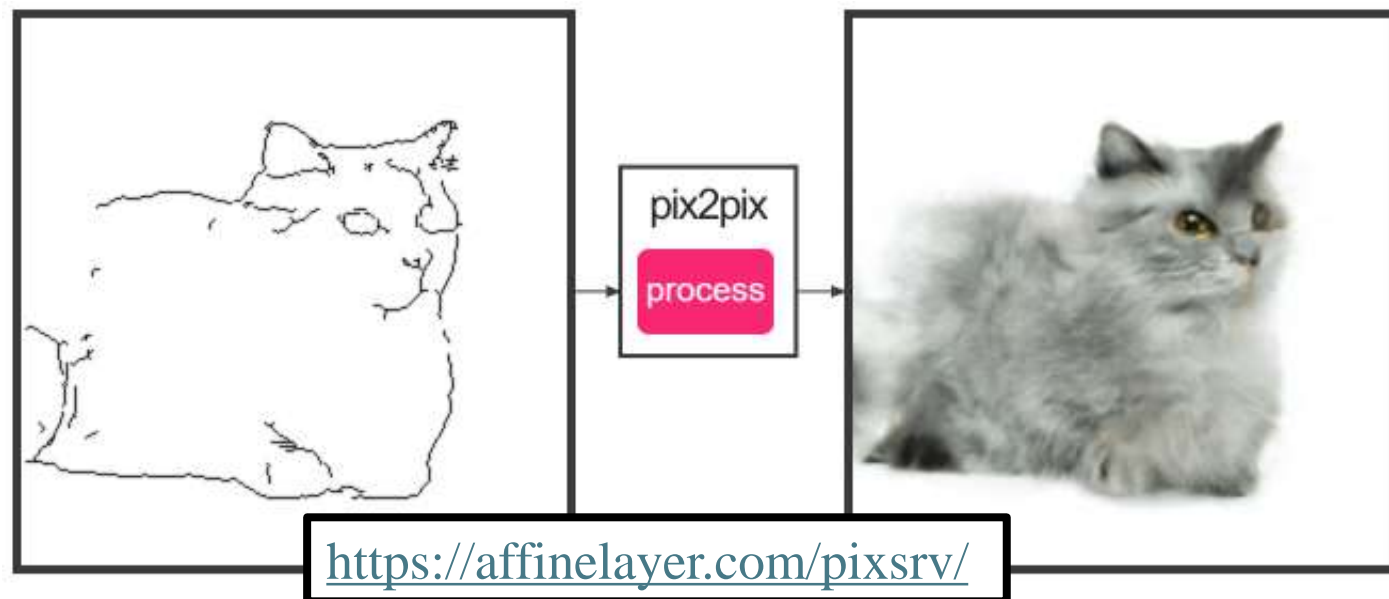


# 素描生成

训练网络最小化  $\|I_j - \hat{I}_j\|$   
 $I_j$  是真实图  $\hat{I}_j$  是在像素  $j$  的预测值.

Input: HxWx1  
Sketch Image

Output: HxWx3  
Image



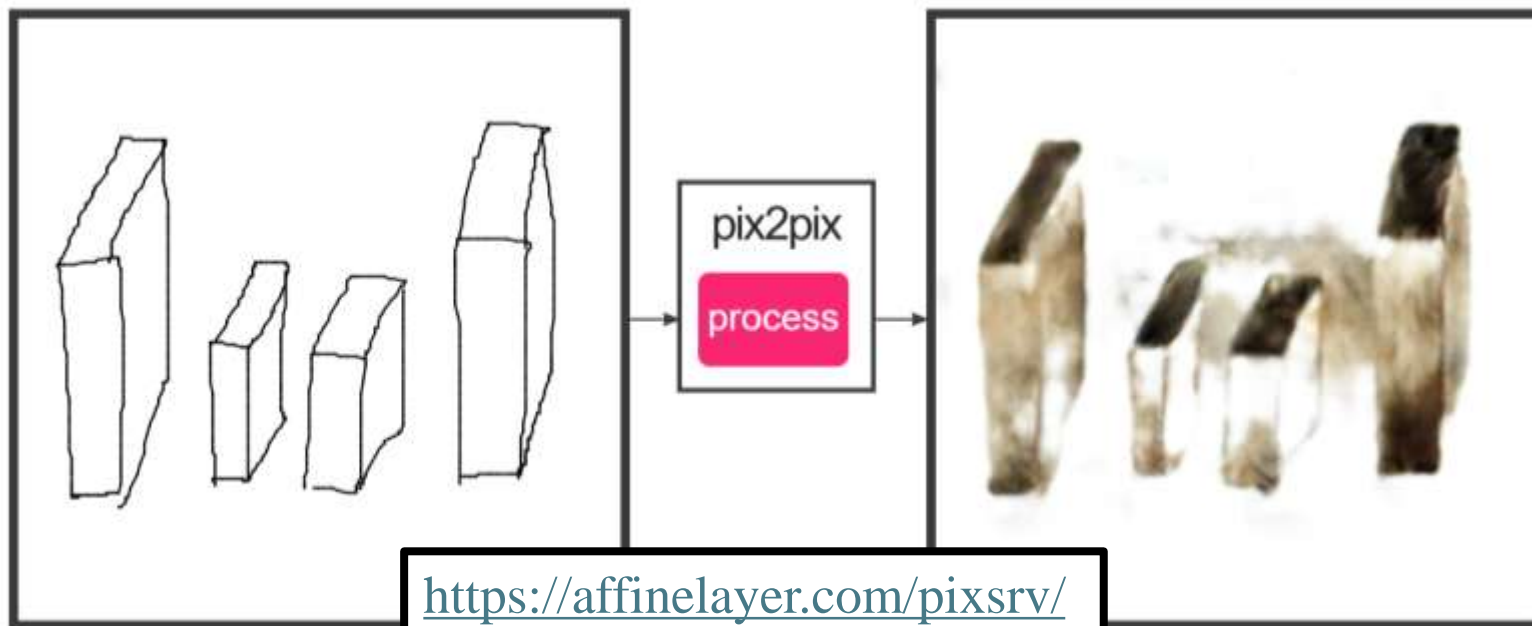
# 素描生成

训练网络最小化  $\|I_j - \hat{I}_j\|$

$I_j$  是真实图  $\hat{I}_j$  是在像素  $j$  的预测值.

Input: HxWx1  
Sketch Image

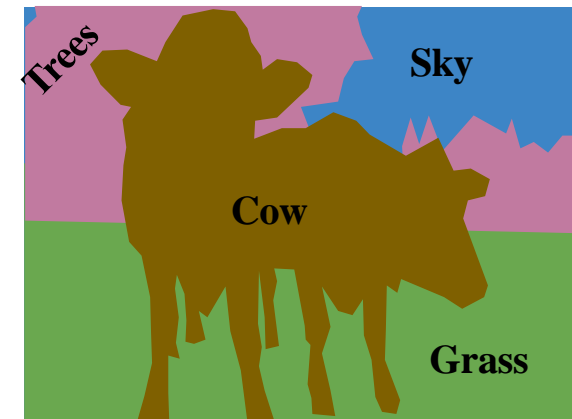
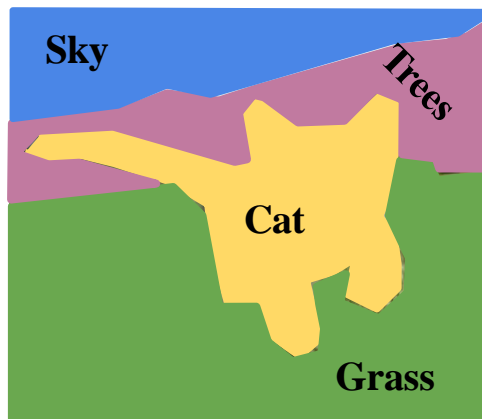
Output: HxWx3  
Image



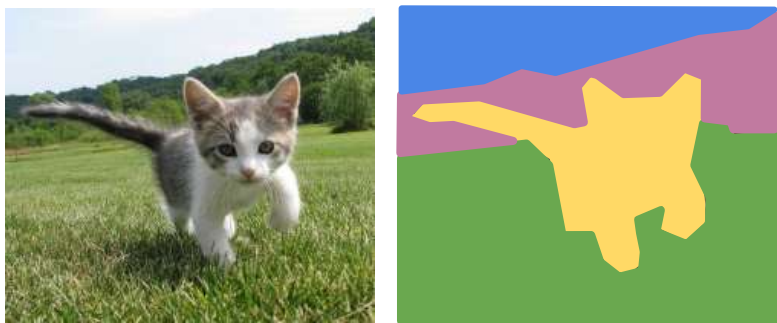
# 语义分割: 问题定义

每个像素点都有一个标签

不关注物体，  
只关注每个像素点

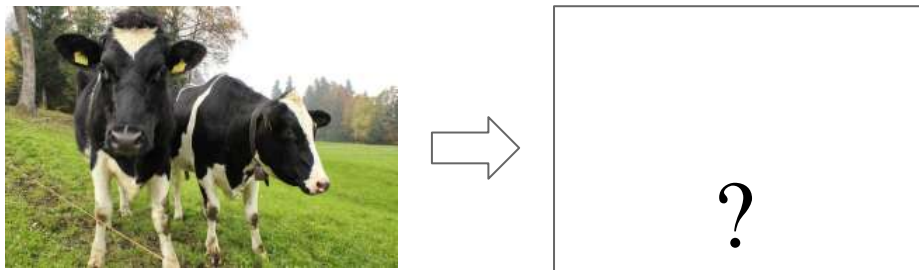


# 语义分割: 问题定义



GRASS, CAT,  
TREE, SKY, ...

成对的训练数据：  
包括每个像素点的  
标注



测试时预测  
每个像素的类别

# 为什么这个任务很难？

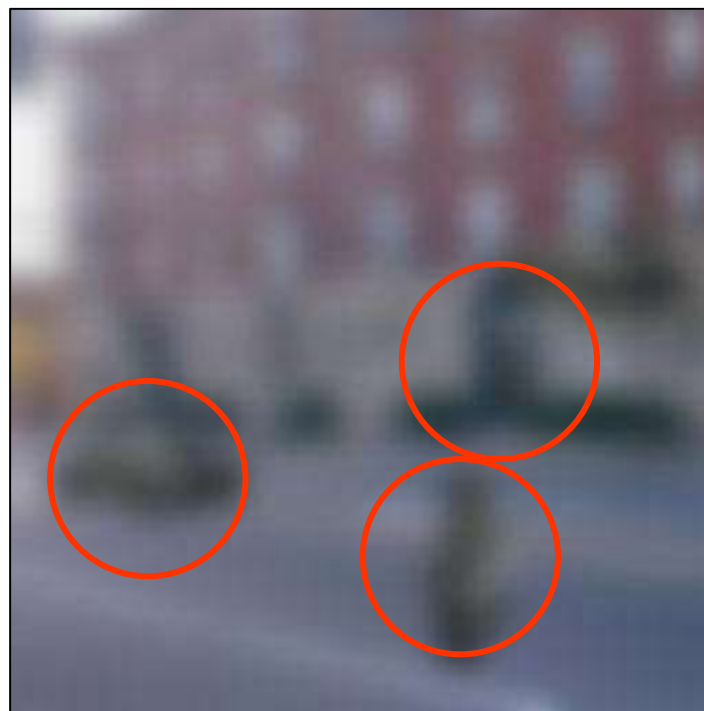
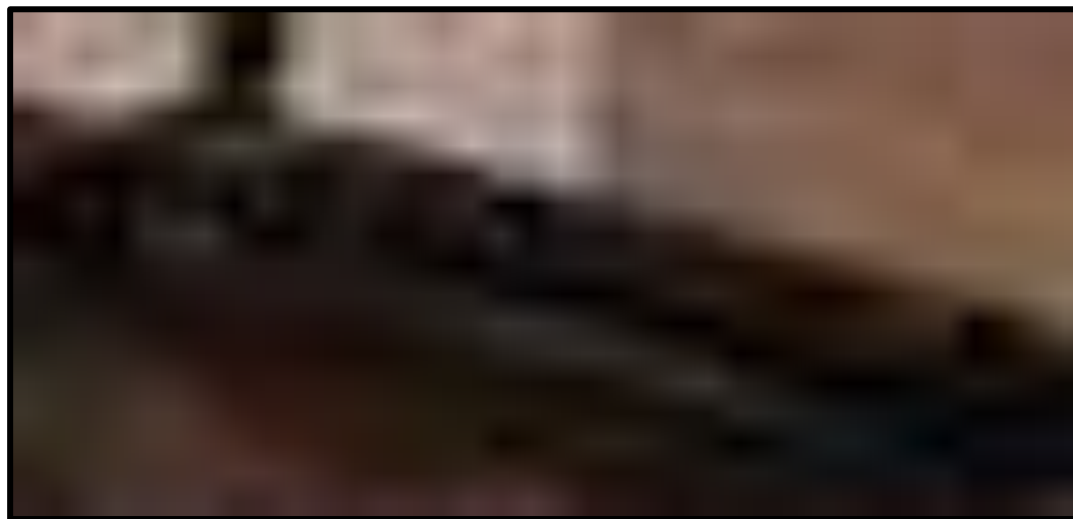


Image credit: A. Torralba

# 为什么这个任务很难？

## 这是啥？



(1) Keyboard?

(3) Old cell phone?

(2) Hammer?

(4) Xbox controller?

# 为什么这个任务很难？



# 来自伟大的人工智能的回答

J You



我把局部放大了，现在请你再来看看这是什么？

ChatGPT

放大之后的图片显示，红框内的物体是一只远程控制器。可以辨认出它的典型长条形状和可能是按钮的细节，这些特征与常见的电视或音响系统的遥控器相符合。

J You



图像红框里是什么物体？

ChatGPT

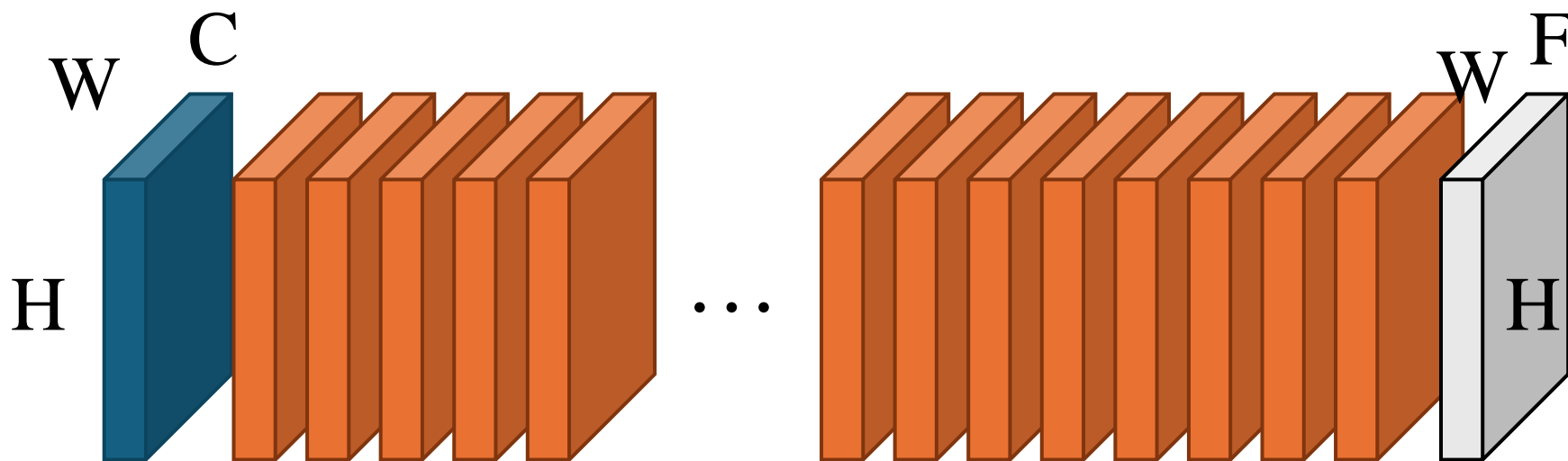
红框里的物体是一只猫。它似乎是在沙发边的地板上蜷缩着，与环境的其他色彩相比，它的颜色较为淡，可能导致它不那么显眼。猫的姿态表明它可能在休息或打盹。



# 为什么这个任务很难？



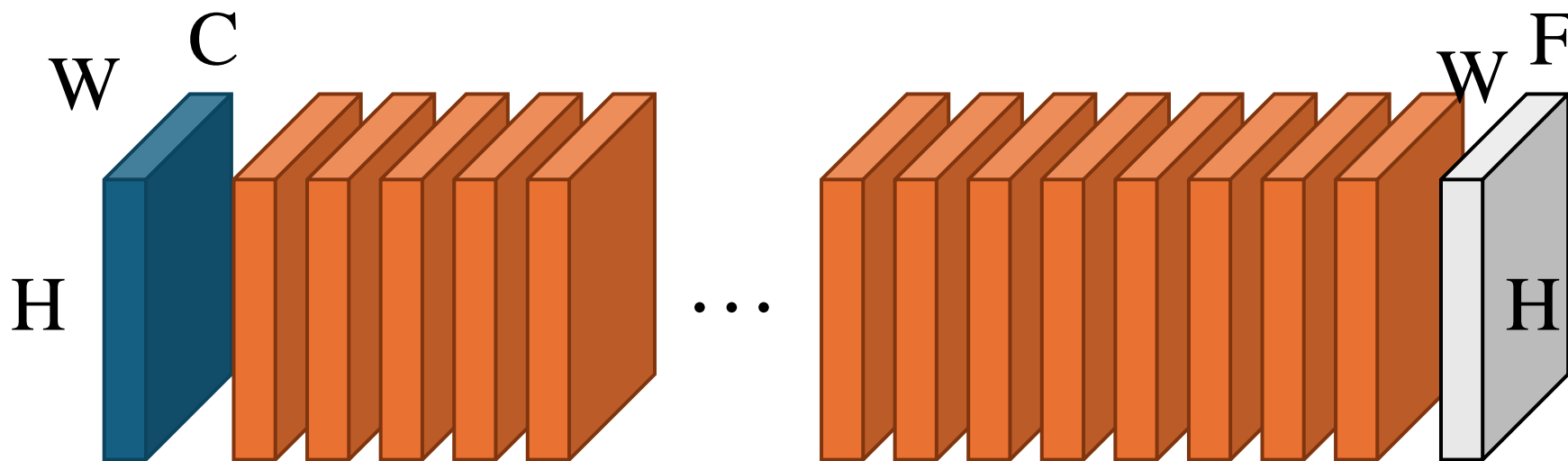
怎么做？堆叠卷积？



$N$  个  $3 \times 3$  卷积的感受野是  $2n+1$  大小的方形区域  
如果想让感受野  $\geq 200$  需要多少卷积？

**100**

## 堆叠卷积? 似乎不行



如果有 200 个 3x3 卷积通道,  $H=W=400$

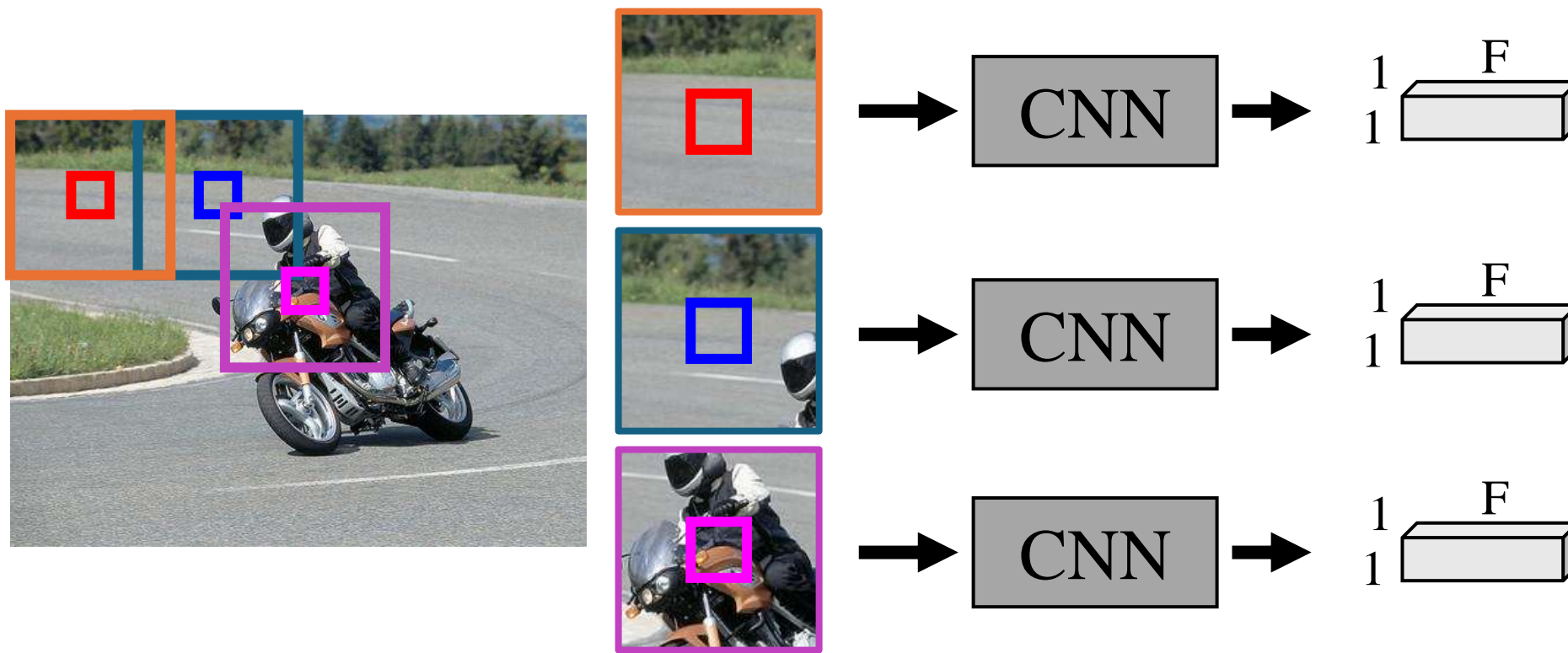
每层的存储开销:  $200 * 400 * 400 * 4 \text{ bytes} = 122\text{MB}$

# 内存溢出!\*

\*100 层, batchsize 20 = 238GB 内存!

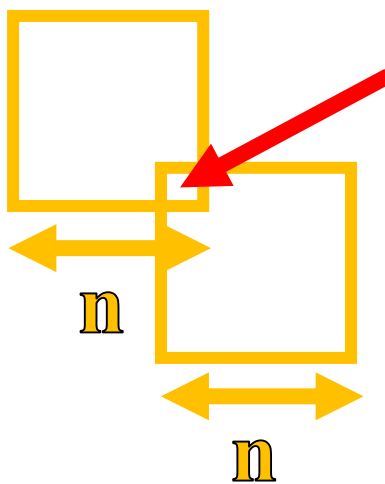
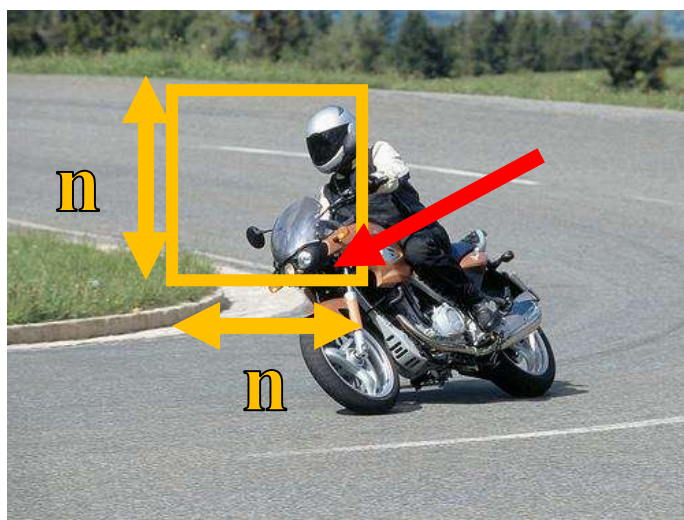
## 想法2: 滑动窗口

把每个窗口的局部拿出来过CNN,  
预测窗口中心的像素标签



# 这样做有什么问题? ...

如果我们单独对每  $N \times N$  patch用CNN提取特征.  
红色的像素点会被CNN计算几次?



Answer:  
 $N * N$

# 核心挑战

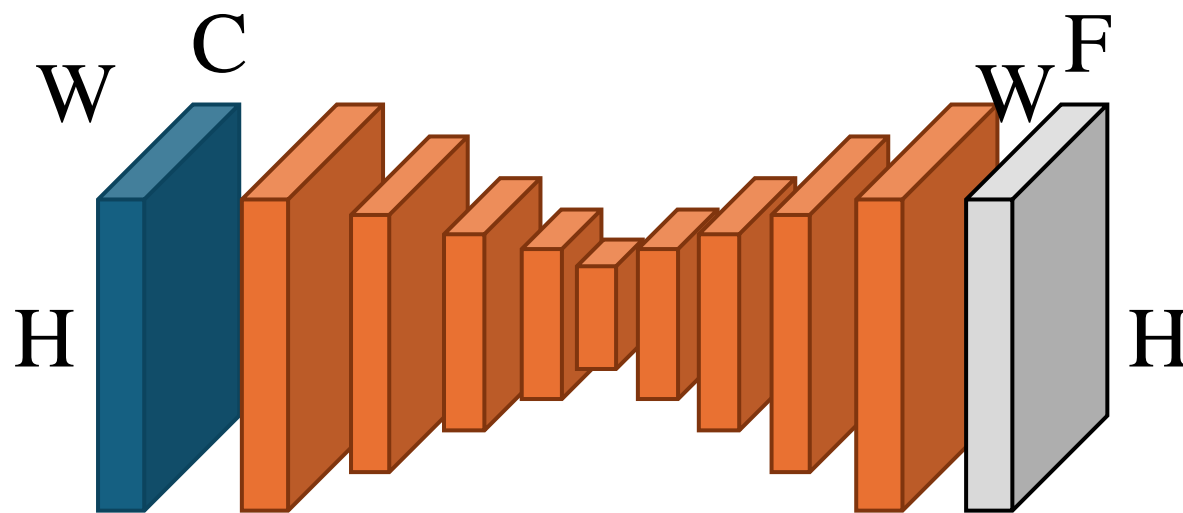
我们需要:

1. 很大的感受野来判断每个像素点周围的情况
2. 不能在满足1的情况下有太大的内存开销

如何解决这对矛盾的问题?

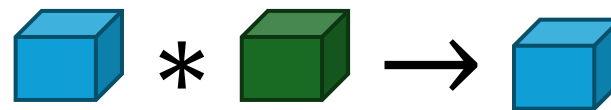
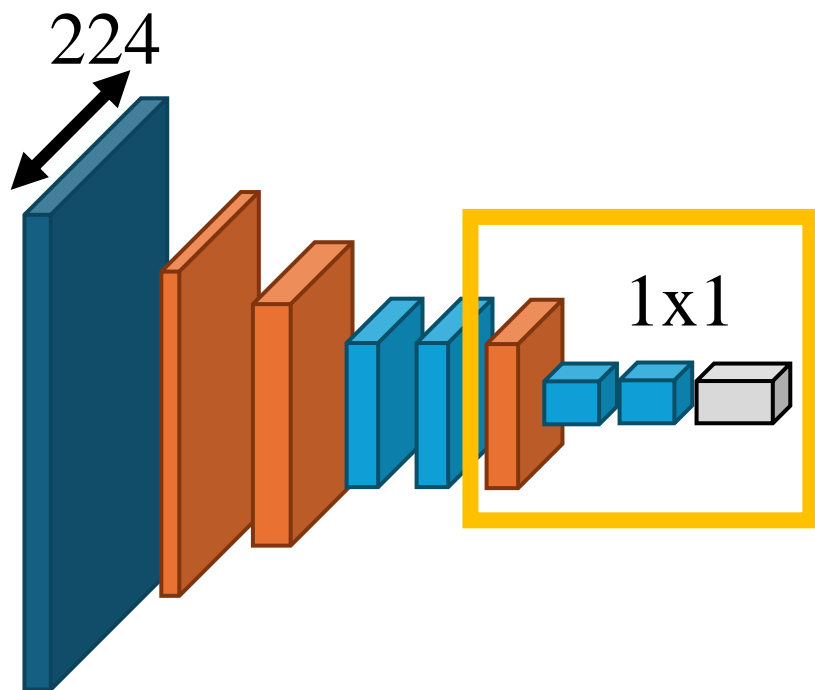
# 编码器-解码器

核心思想: 先 **downsample** 下采样  
再 **upsample** 上采样.  
怎么做下采样?  
卷积, 池化



# 如何设计这样的网络?

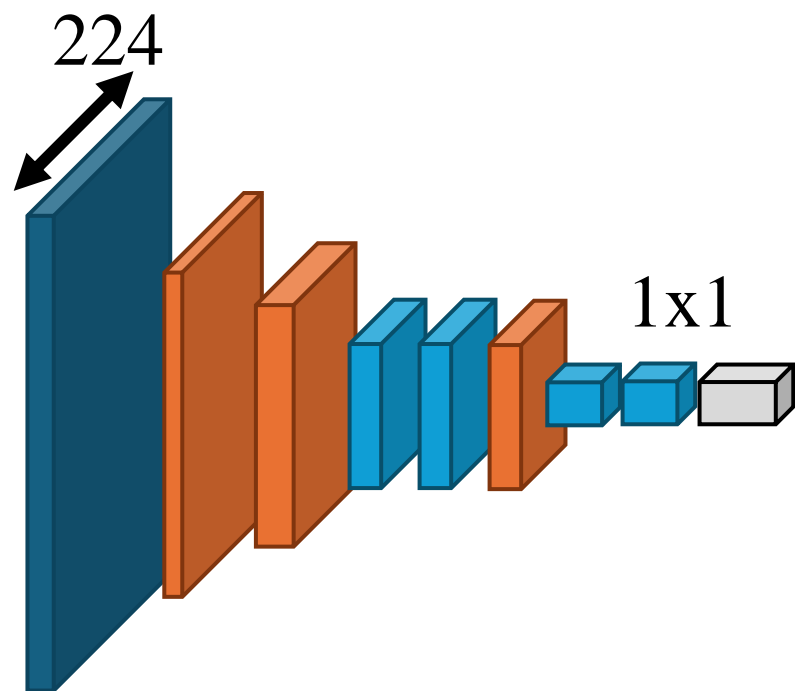
之前的网络都是1维的  
最终输出



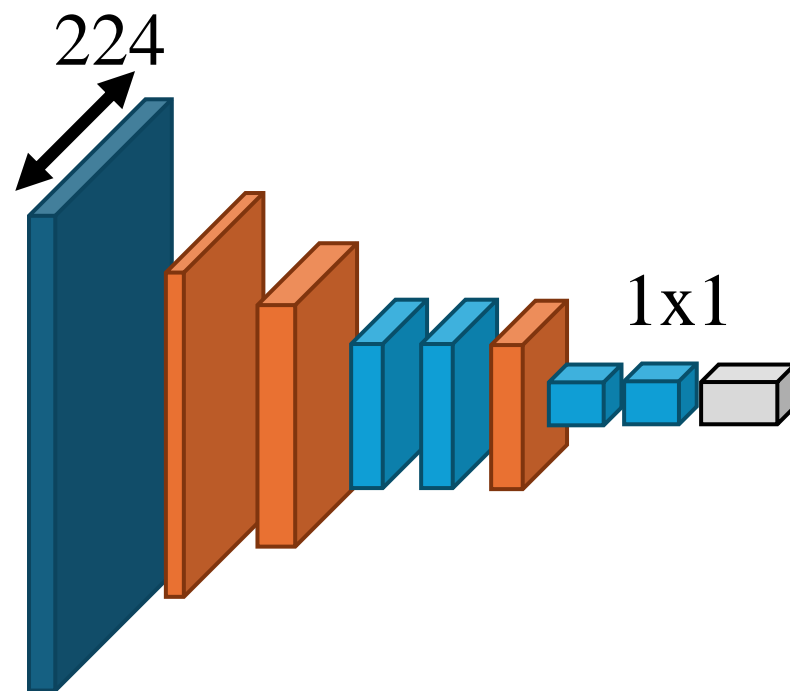
最后一层全连接层我们  
可以用1\*1的卷积来替代

# 如何设计这样的网络?

把图像映射到向量的  
CNN



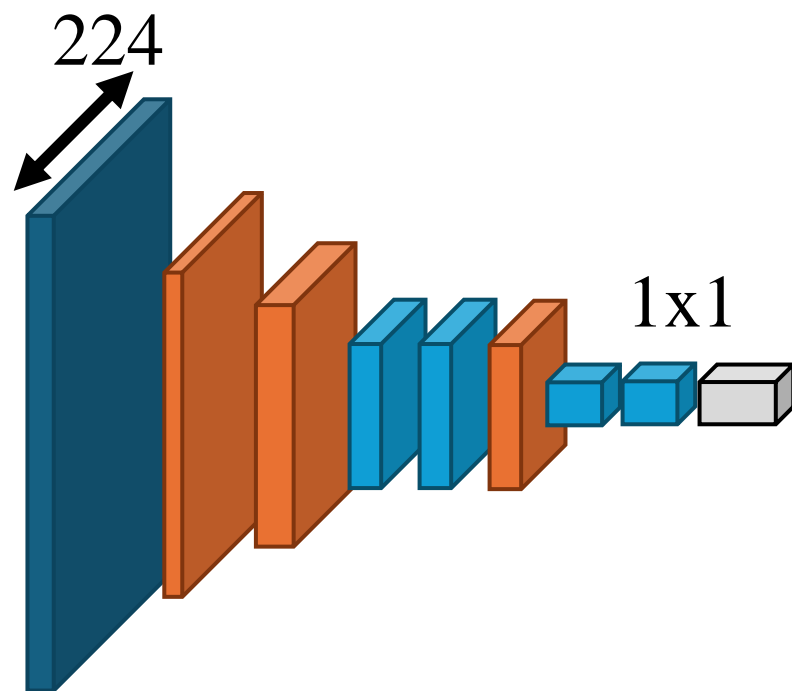
把图像映射到图像的  
CNN



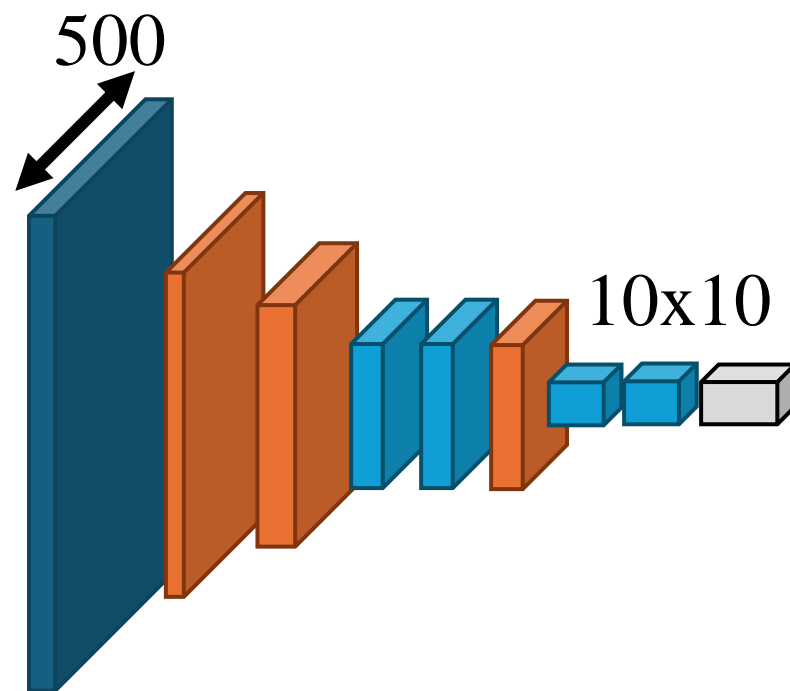
输入尺寸变大会怎样?

# 如何设计这样的网络？

把图像映射到向量的  
CNN

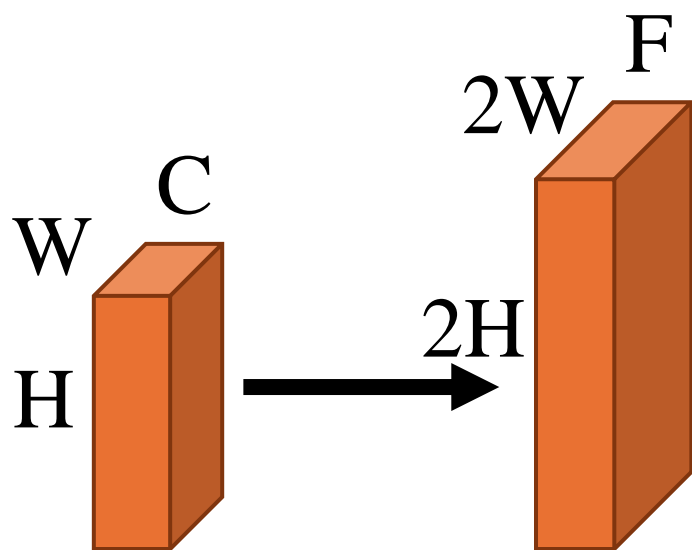


把图像映射到图像的  
CNN



输出有空间尺度了！ 我们可以利用它！

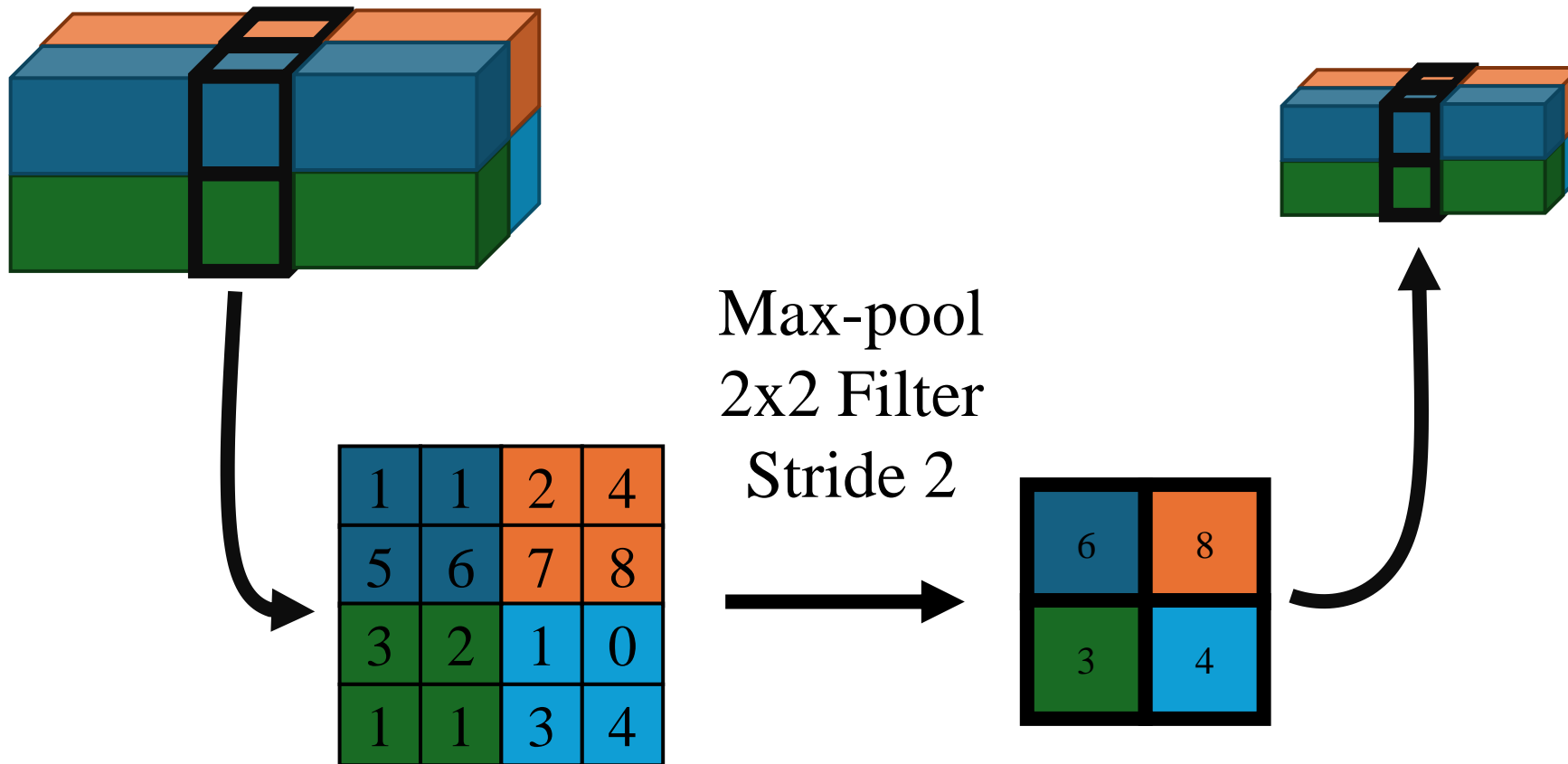
# 怎么做上采样?



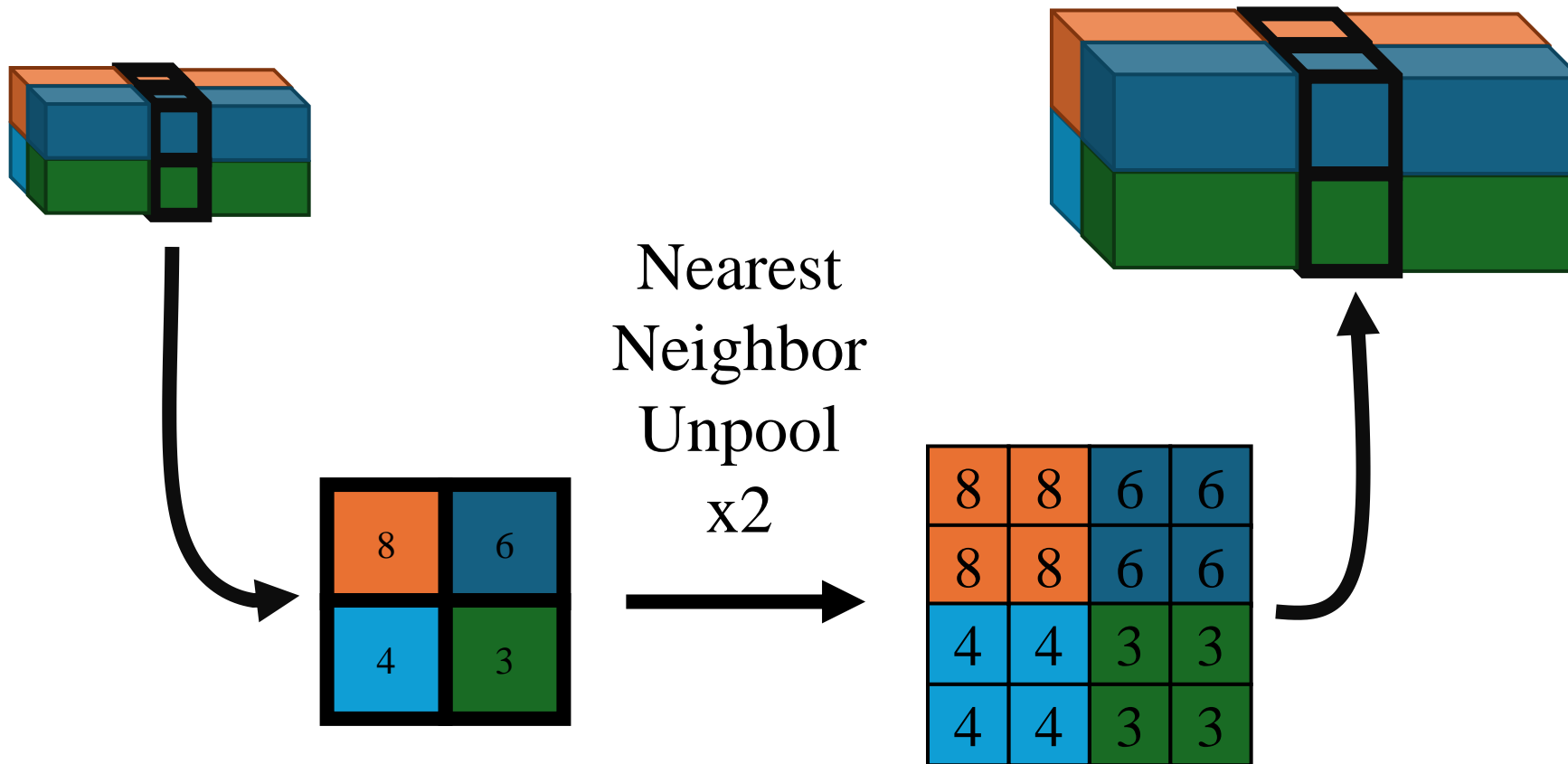
做下采样的反向操作:

1. Pooling  $\rightarrow$  “Unpooling”
2. Convolution  $\rightarrow$  “Transpose Convolution”

# 回顾：池化

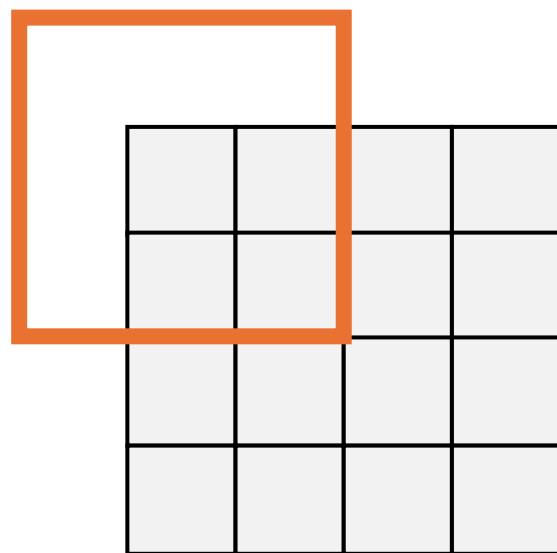


# Unpooling——反池化



# 回顾：卷积

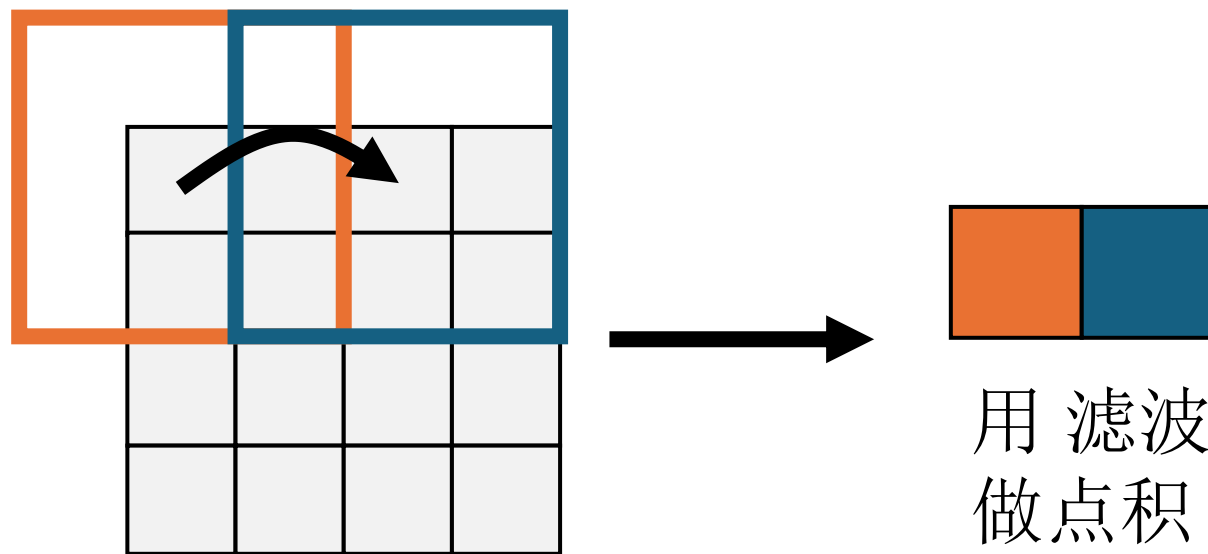
3x3卷积, Stride 2, Pad 1



用滤波  $f$  与输入  
做点积

# 回顾：卷积

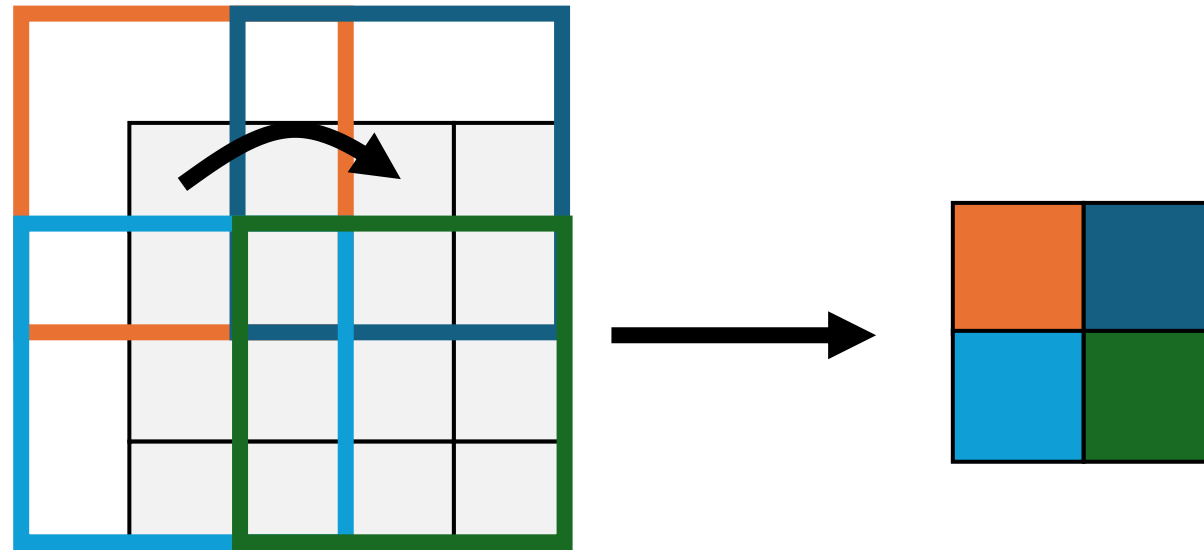
3x3卷积, Stride 2, Pad 1



用滤波  $f$  与输入  
做点积

# 回顾：卷积

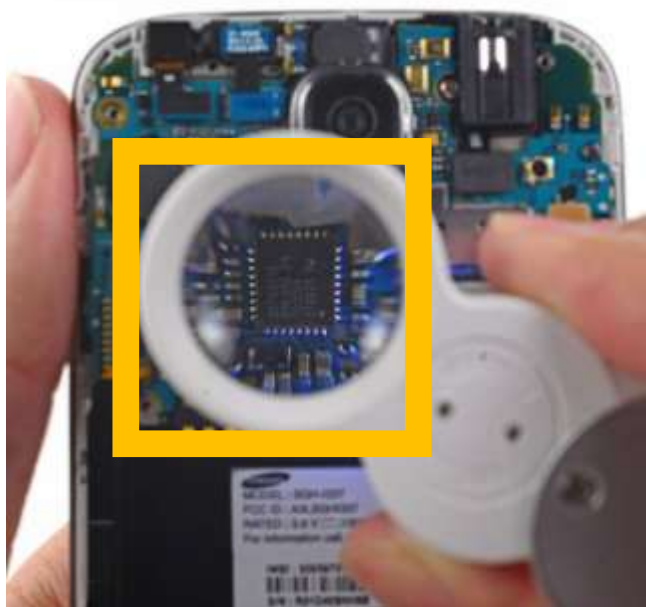
3x3卷积, Stride 2, Pad 1



# 反卷积

## 卷积

滤波的作用：关注局部区域的内容



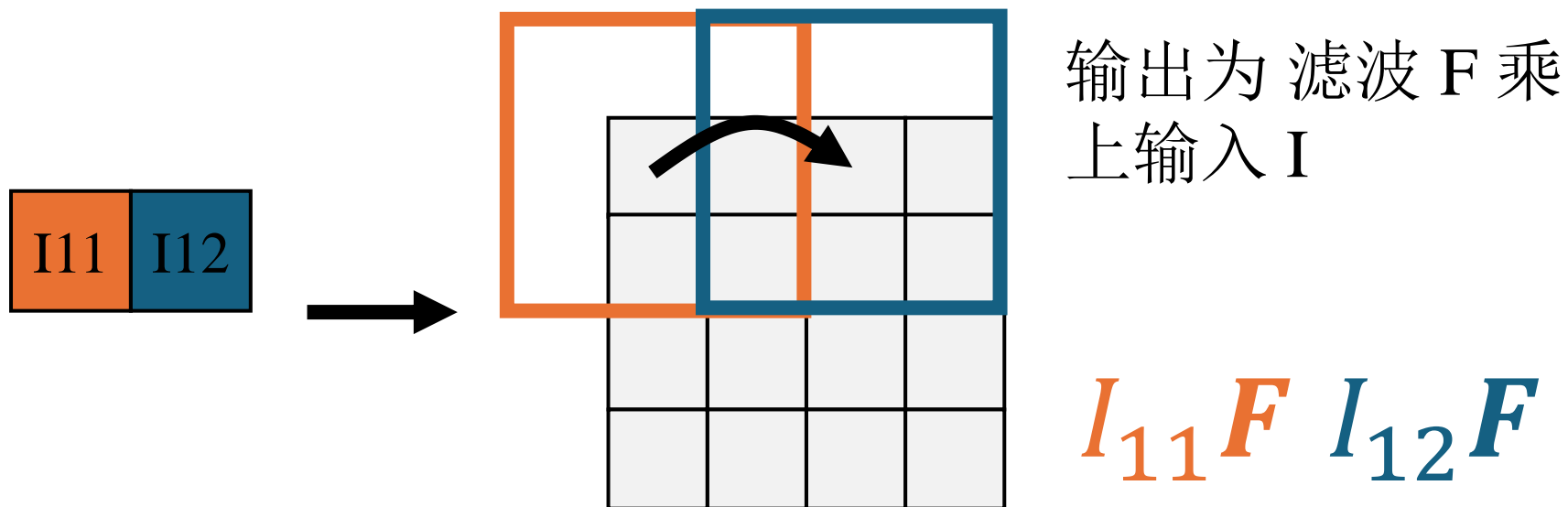
## 反卷积

滤波的作用：像贴瓷砖一样把滤波“铺上去”



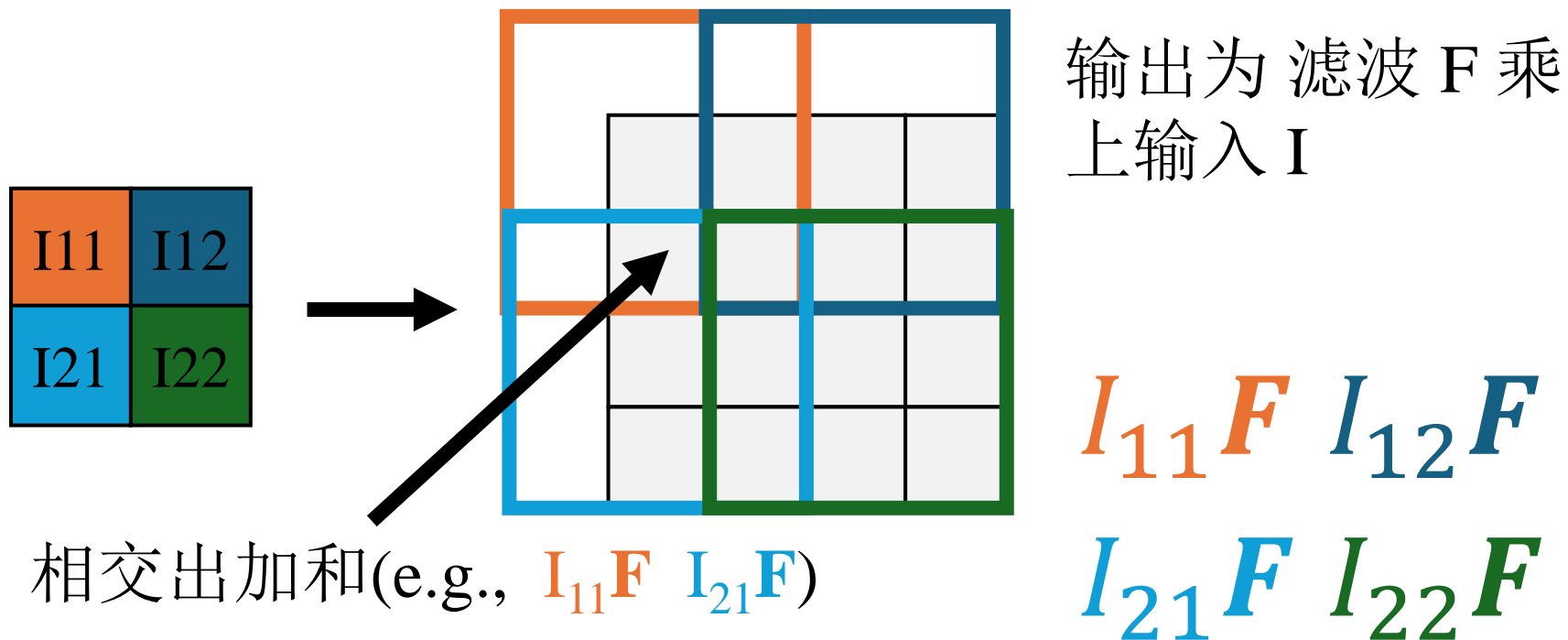
# Transpose Convolution、DeConvolution——反卷积

3x3 Transpose Convolution, Stride 2, Pad 1



# Transpose Convolution、DeConvolution——反卷积

3x3 Transpose Convolution, Stride 2, Pad 1



# 为什么叫“Transpose Convolution”？

我们可以用矩阵乘法的形式实现卷积

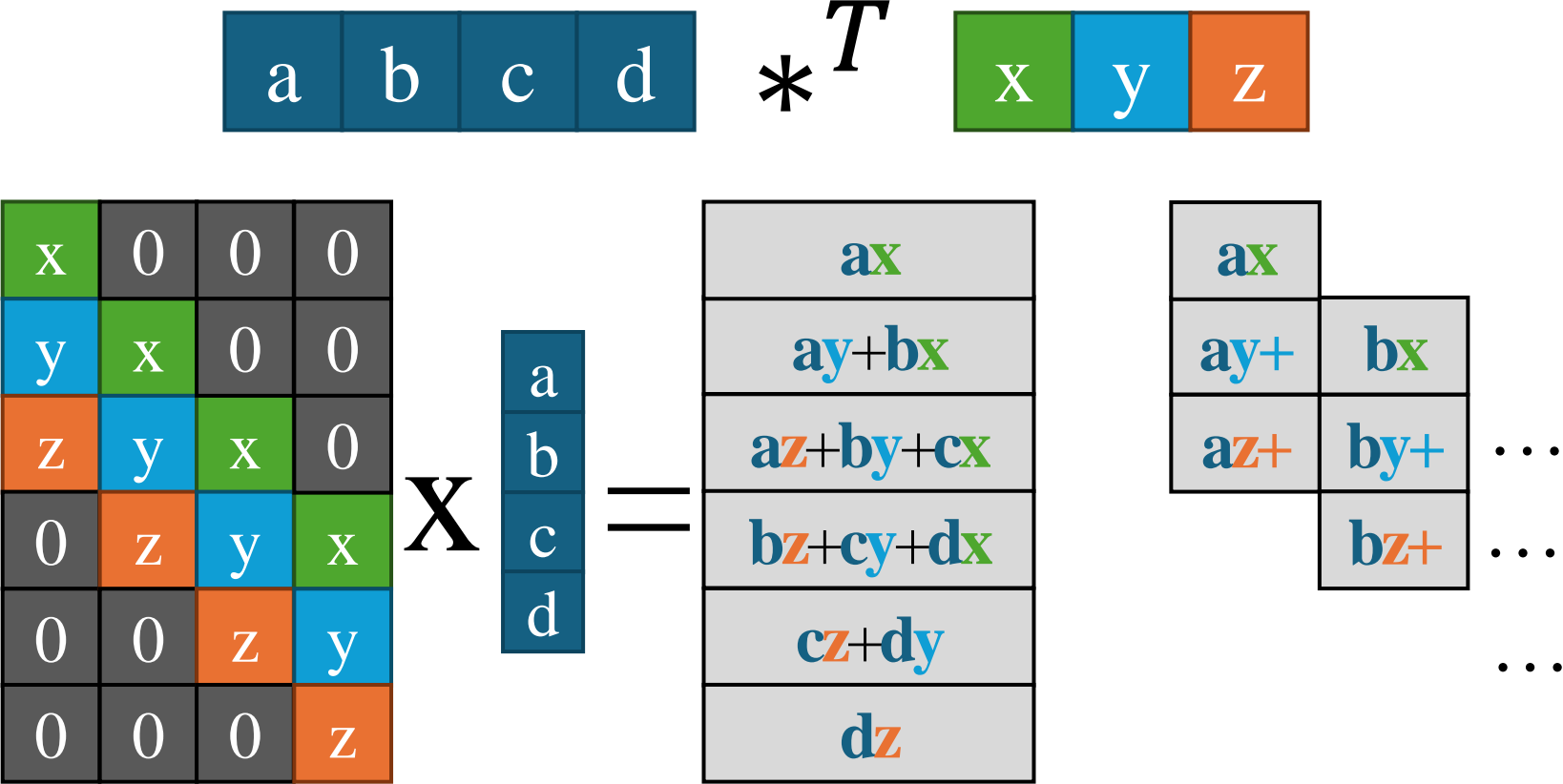
Input: 4, Filter: 3, Stride: 1, Pad: 1

$$\begin{bmatrix} a & b & c & d \end{bmatrix} * \begin{bmatrix} x & y & z \end{bmatrix}$$

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

# 为什么叫“Transpose Convolution”？

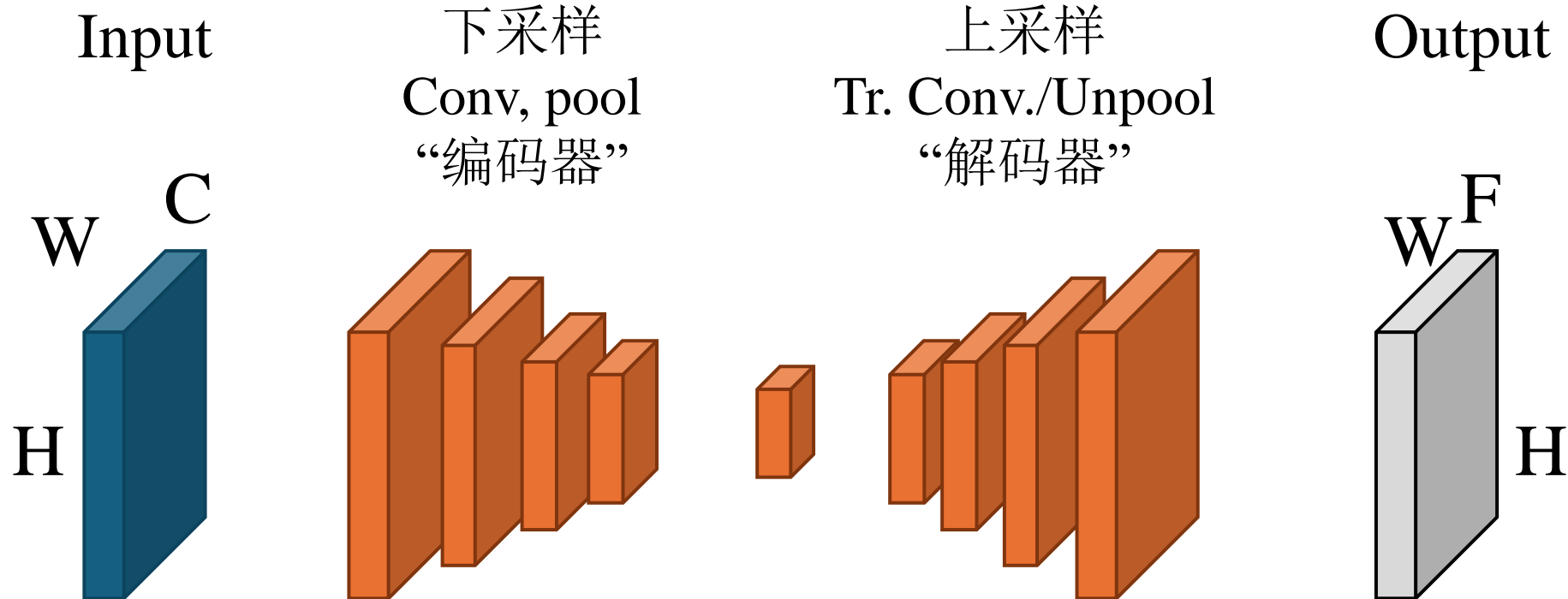
反卷积与卷积矩阵乘法形式为转置操作



Example Credit: L. Fei-Fei, J. Johnson, S. Yeung

# 汇总起来

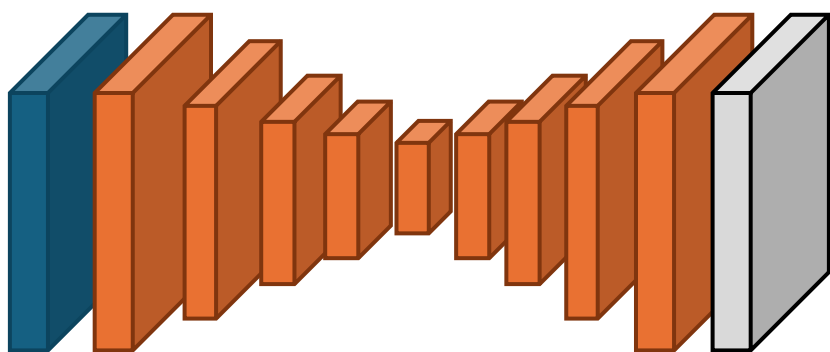
卷积 + 池化 来做 下采样/压缩/编码  
反卷积/反池化 来做 上采样/解压缩/解码



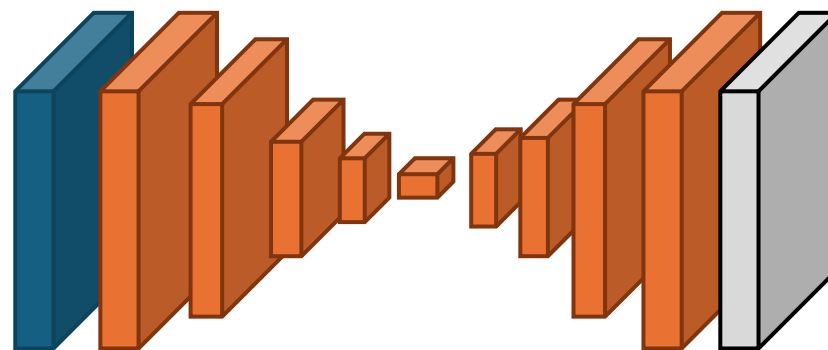
# 那么怎么设计这样的网络呢？

- 最好一直保留一些空间尺度

编码到较小尺度, 然后解码.

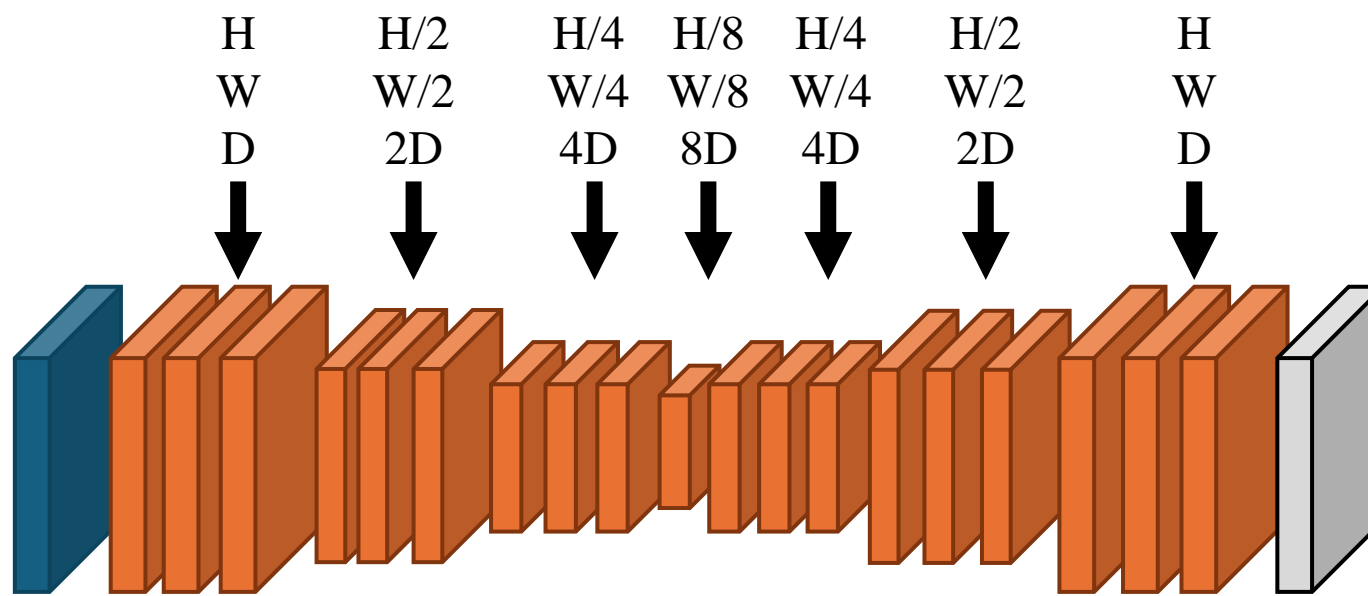


编码到1维向量, 然后解码



# 设计经验

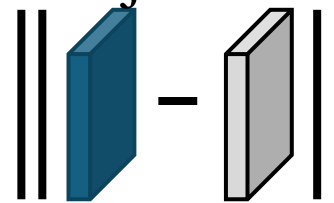
- 在每个分辨率上堆叠几个普通卷积层.
- 经常设计成隔几层输出空间尺寸减半，通道数量翻倍



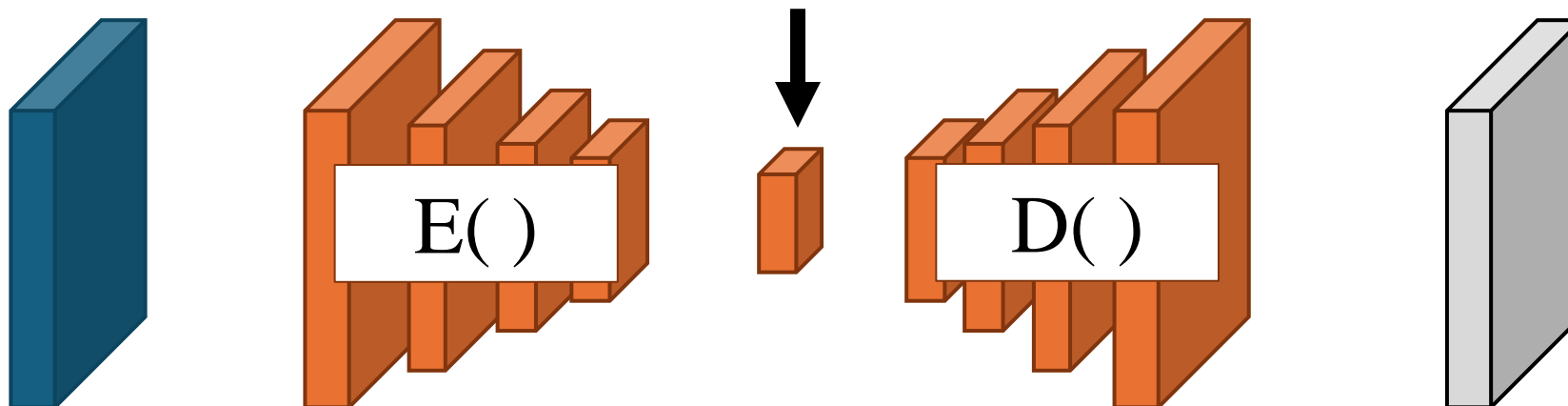
# Autoencoders 自编码器

把输入编码压缩到“bottleneck”，然后  
解码到输入原尺寸

Objective:


$$\|D(E(X)) - X\|$$

Bottleneck 瓶颈/  
Latent Space 隐变量空间/  
Latent Code 隐变量编码



# Latent Space\* 隐变量空间

## Interpolation in Latent Space



\*In the interest of honesty in advertising: not an autoencoder, but a similar method with the same goal of learning a latent space

Result from Wu et al. *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*. NIPS 2016

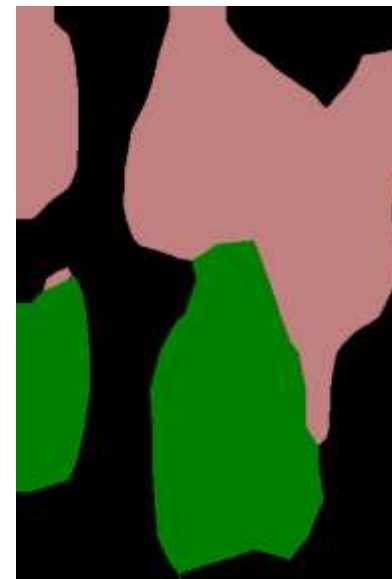
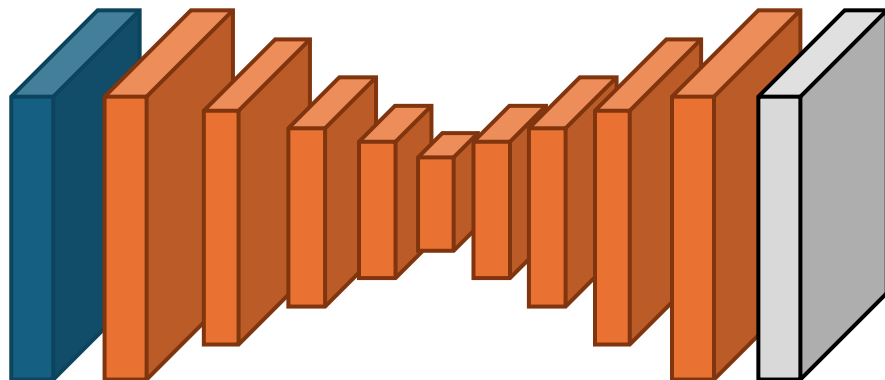
# 缺失的细节内容

如果输出尺寸时  $H \times W$ , 直接上采样一个经过下采样的图像, 我们经常会得到一个缺失细节的结果.

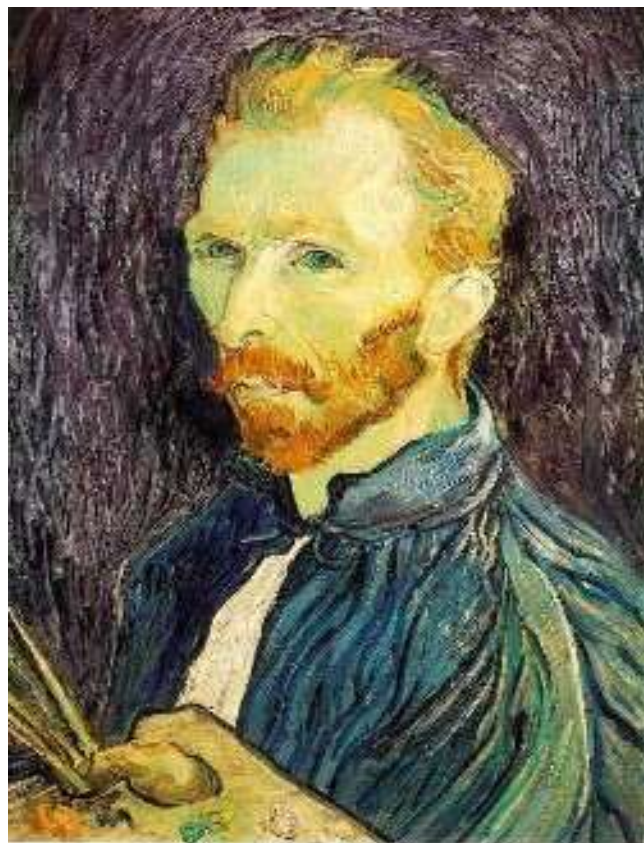
为什么?



下采样时信息会丢失!



# 回顾：下采样



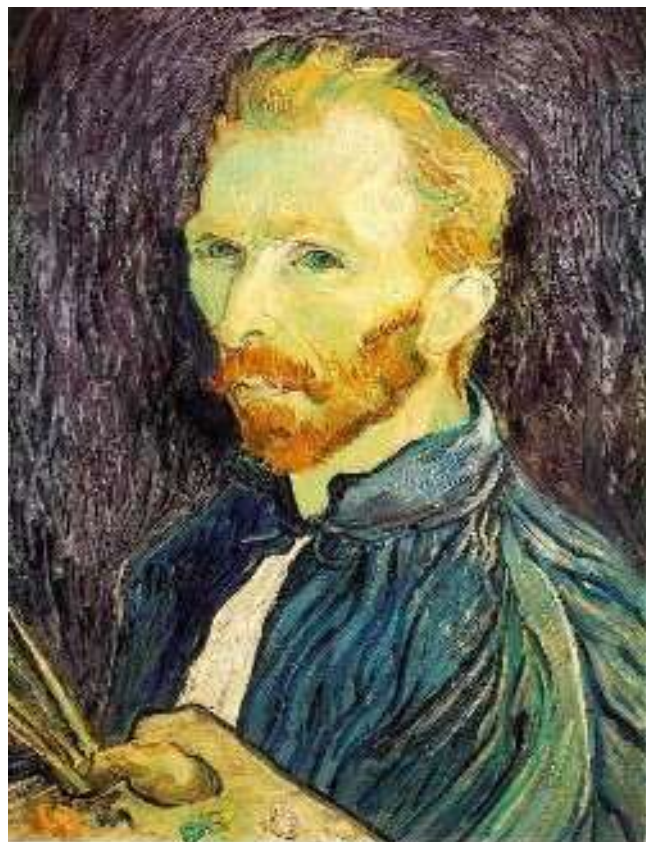
1/4



1/8

隔一行隔一列扔掉一半的图像  
- 图像下采样 *image sub-sampling*

# 回顾：下采样



1/2



1/4 (2x zoom)



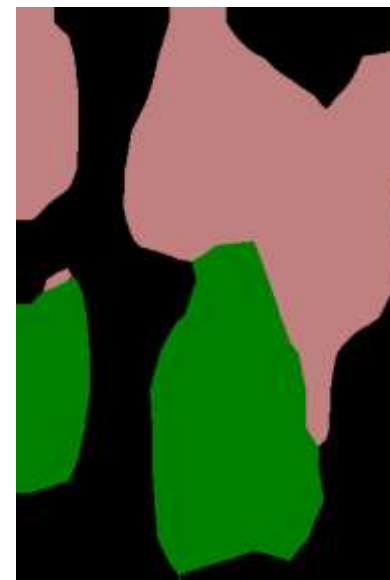
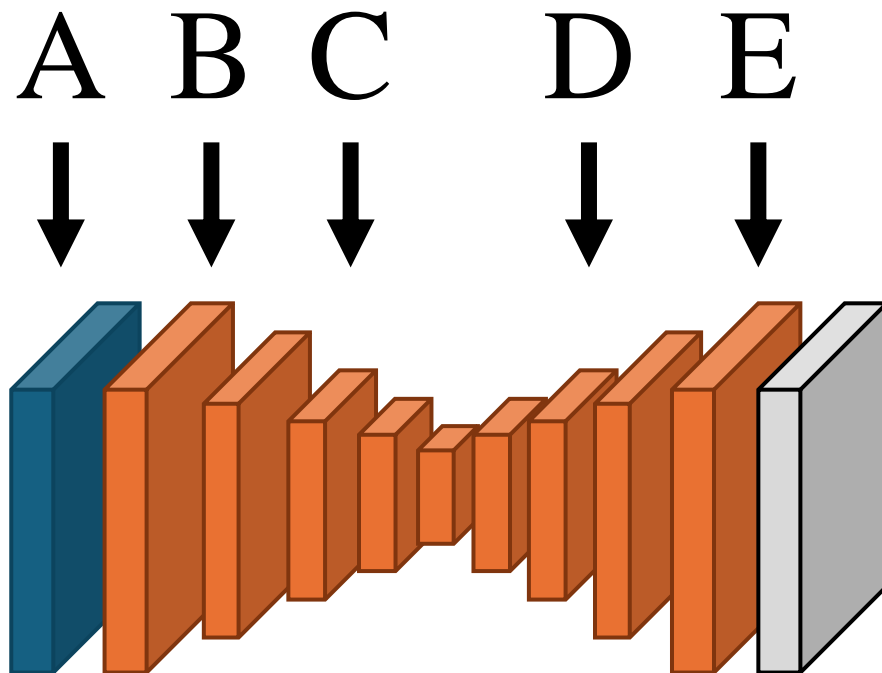
1/8 (4x zoom)

如果再上采样回来，为什么看起来这么粗糙？

Source: S. Seitz

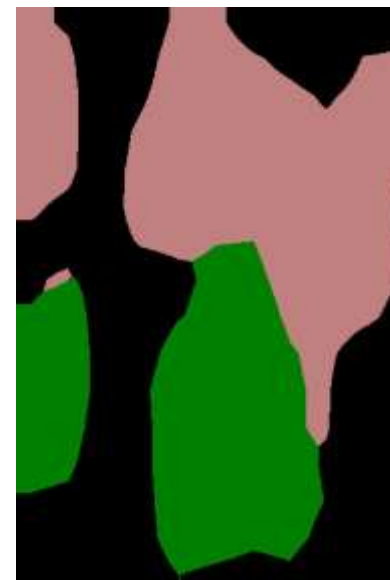
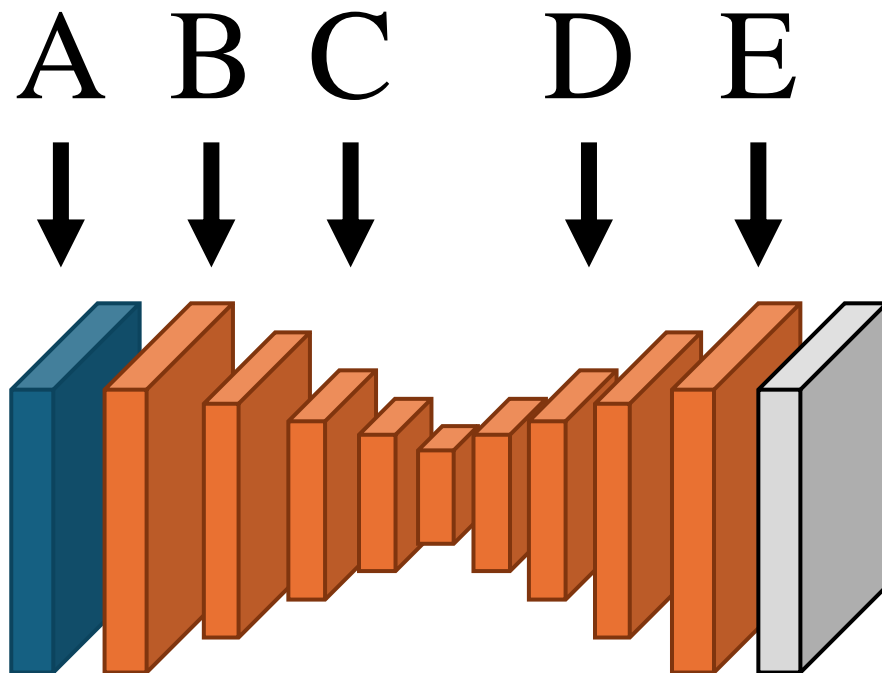
# 缺失的细节内容

哪里包含的图像高频信息最多？



# 缺失的细节内容

哪里缺失的图像高频信息最多？

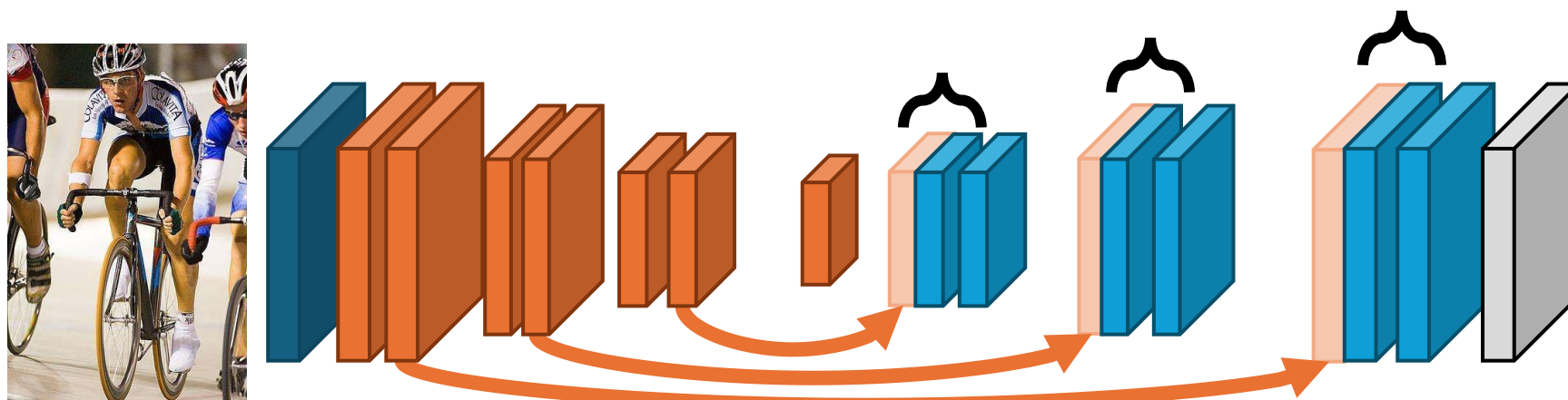


# 缺失的细节内容

怎么把缺失的细节补回来？

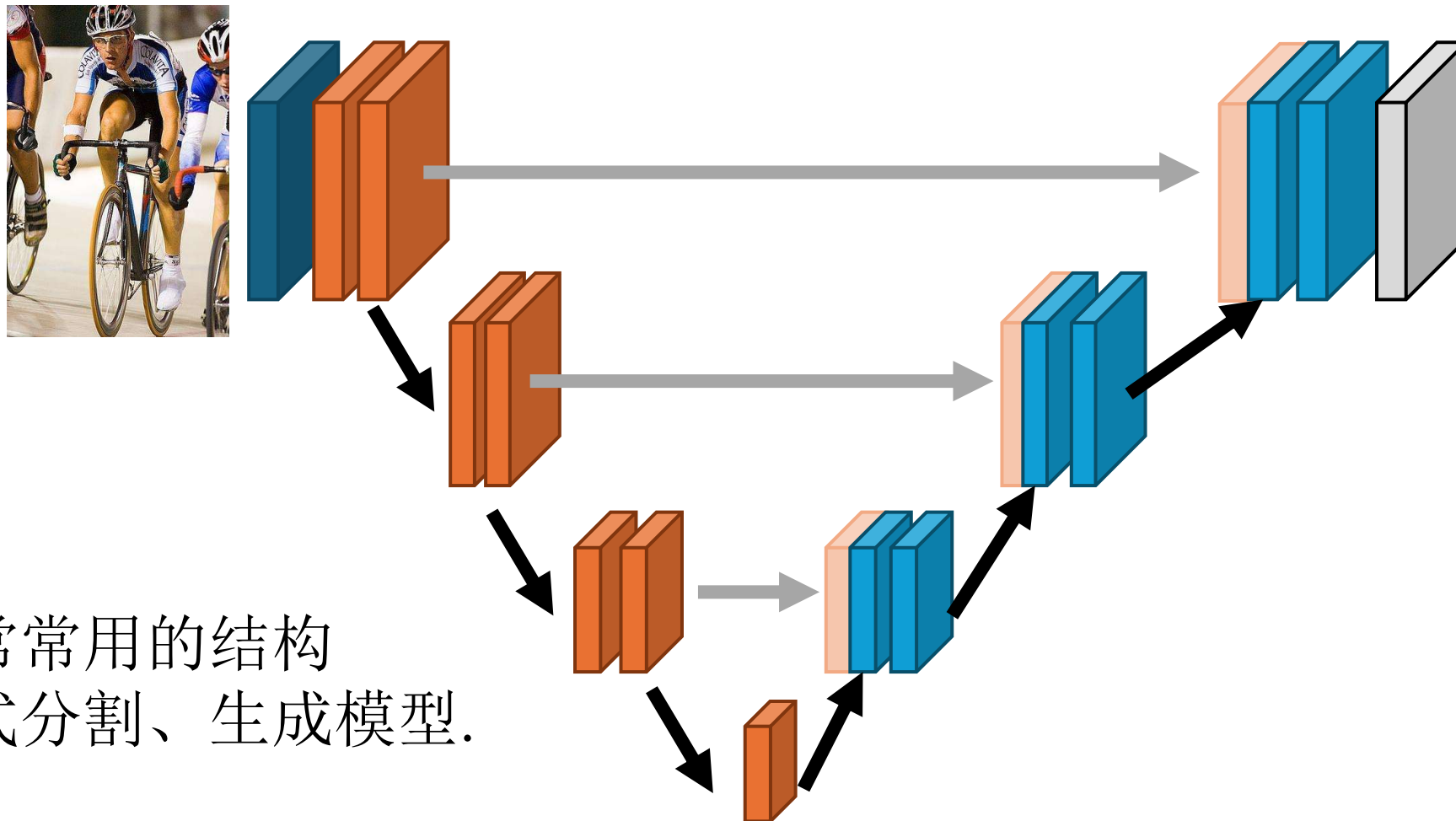
复制这些层的输出。

把他们拷贝到相同分辨率的、缺失细节的地方去。



Copy

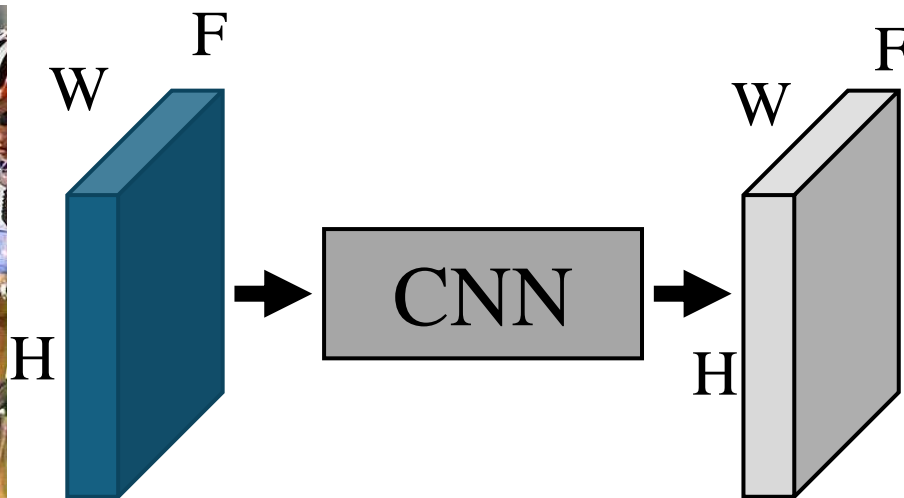
# U-Net



非常非常常用的结构  
用于各式分割、生成模型.

# 评估像素级别的预测

输入图像



Predicted  
Classes



怎么得到 $H \times W \times F$ 的分类预测?

在评分输出上取argmax

# 评估语义分割

我们有了预测值，怎么做评估？

Input



Prediction ( $\hat{y}$ )



Ground-Truth ( $y$ )

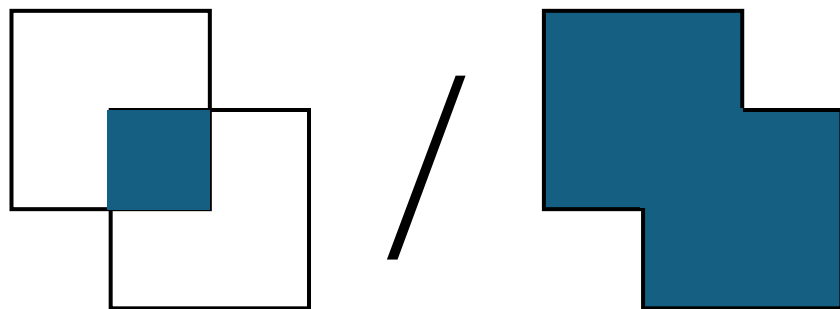


# 评估语义分割

每个像素都被分为F类中的某一个。

准确率:  $\text{mean}(\hat{y} = y)$

交并比 IoU (Intersection over union), 预测类别和真实类别的重合区域/合并区域。所有类别取平均



Prediction  
( $\hat{y}$ )



Ground-Truth  
( $y$ )



# 回顾： 来自伟大的人工智能的回答

J You



我把局部放大了，现在请你再来看看这是什么？

ChatGPT

放大之后的图片显示，红框内的物体是一只远程控制器。可以辨认出它的典型长条形状和可能是按钮的细节，这些特征与常见的电视或音响系统的遥控器相符合。

J You



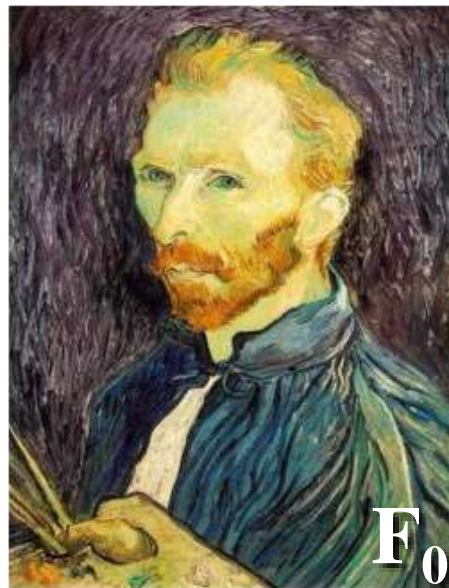
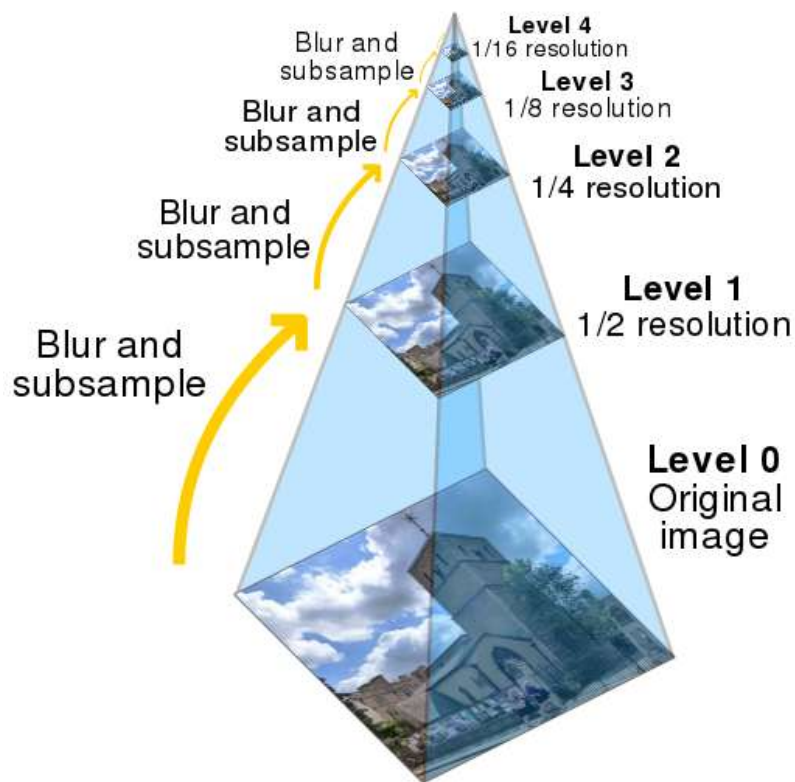
图像红框里是什么物体？

ChatGPT

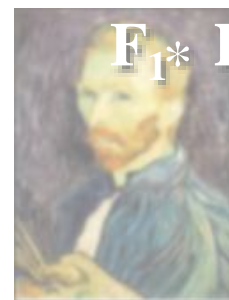
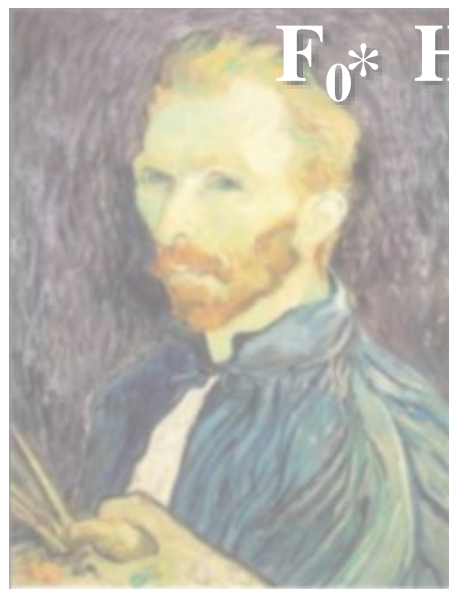
红框里的物体是一只猫。它似乎是在沙发边的地板上蜷缩着，与环境的其他色彩相比，它的颜色较为淡，可能导致它不那么显眼。猫的姿态表明它可能在休息或打盹。



*Gaussian pyramid*  
高斯金字塔



•••



# 语义分割变种: DeepLab

沿用FCN/UNET网络结构

添加 Spatial pyramid pooling:

融合不同尺度的信息, 以增强分割效果

提升尺度不变性, 性能较好

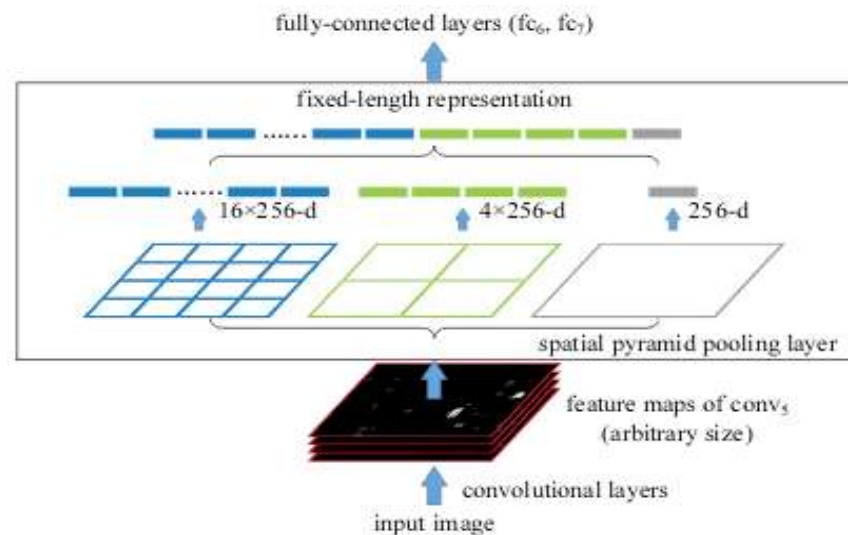


Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the conv<sub>5</sub> layer, and conv<sub>5</sub> is the last convolutional layer.

# 语义分割变种：HRNet

维护每个分辨率的特征

多分辨率并行多次特征融合

继续提升尺度不变性，性能更好

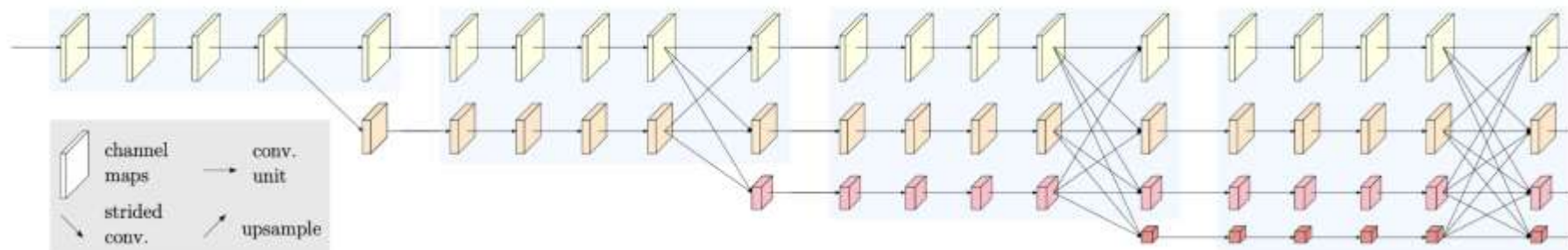


Fig. 2. An example of a high-resolution network. Only the main body is illustrated, and the stem (two stride-2  $3 \times 3$  convolutions) is not included. There are four stages. The 1st stage consists of high-resolution convolutions. The 2nd (3rd, 4th) stage repeats two-resolution (three-resolution, four-resolution) blocks. The detail is given in Section 3.

# 实例分割与物体检测

分类



**CAT**

无空间限定

语义分割



**GRASS, CAT, TREE, SKY**

像素级预测

物体检测



**DOG, DOG, CAT**

实例分割



**DOG, DOG, CAT**

多物体

## 回顾

“语义分割”：标注每个像素点其对应的类别。

Input

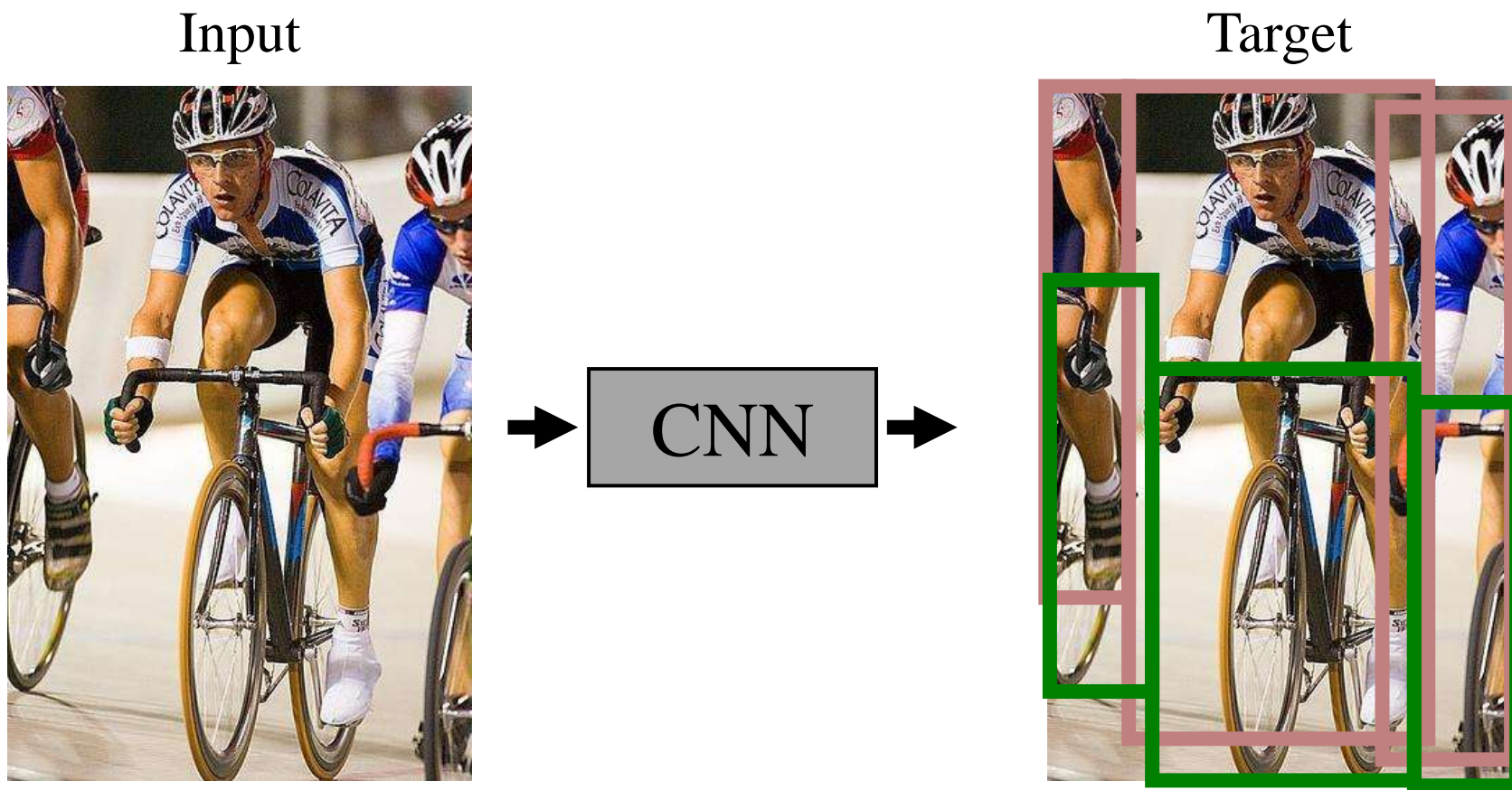


Target



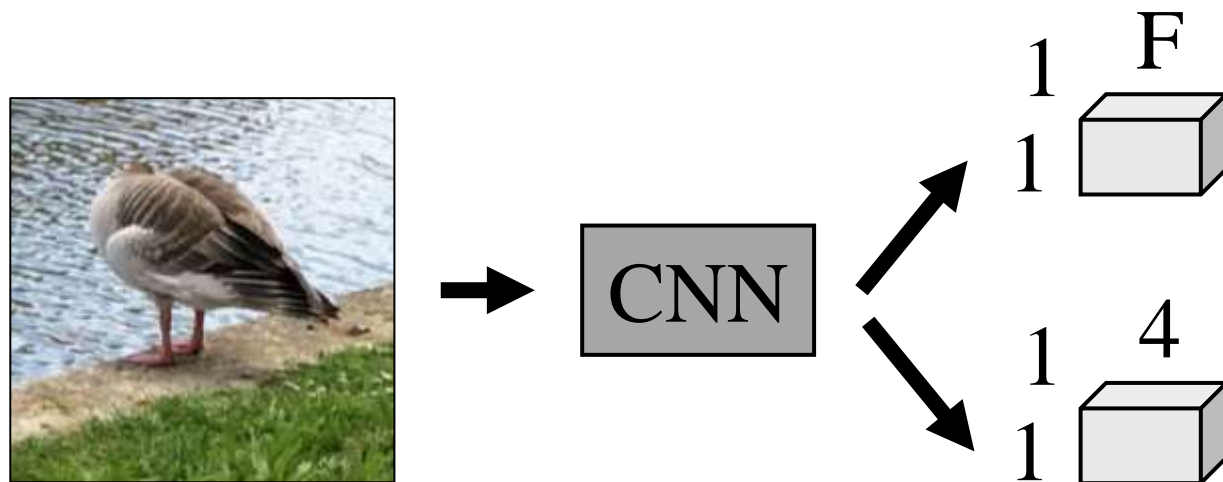
## Object Detection: 物体检测

“Object Detection”: 对某个类别的所有实体画一个边界框





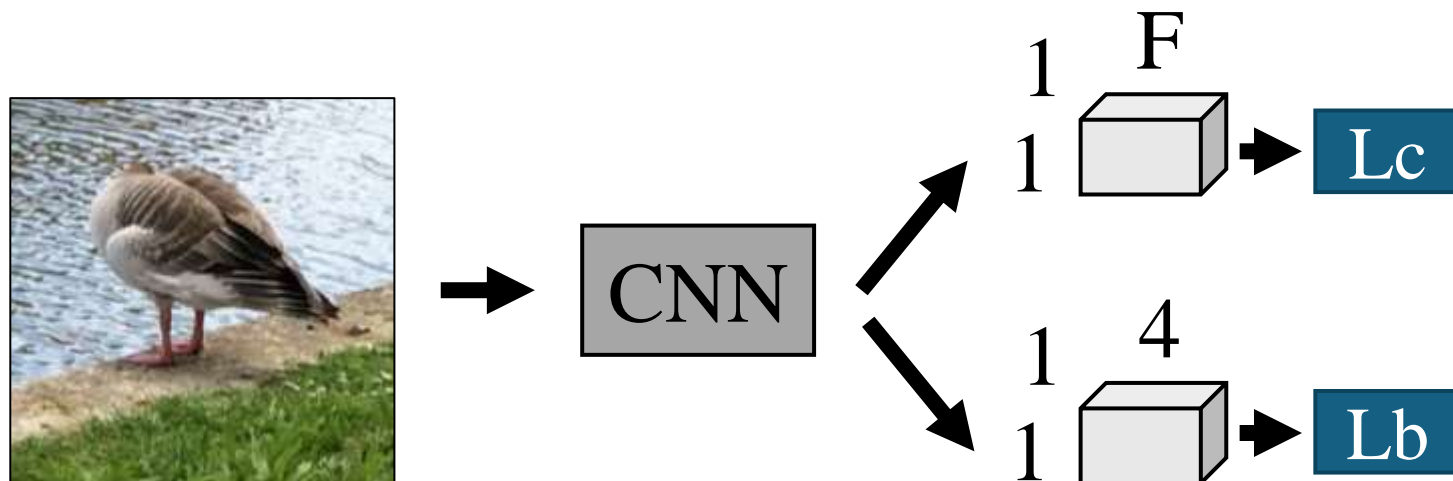
# 如何错误地做物体检测



加一个额外的输出：  
预测物体所在的位置

$[x,y,width,height]$  or  $[\min X,\min Y,\max X,\max Y]$

# 如何错误地做物体检测



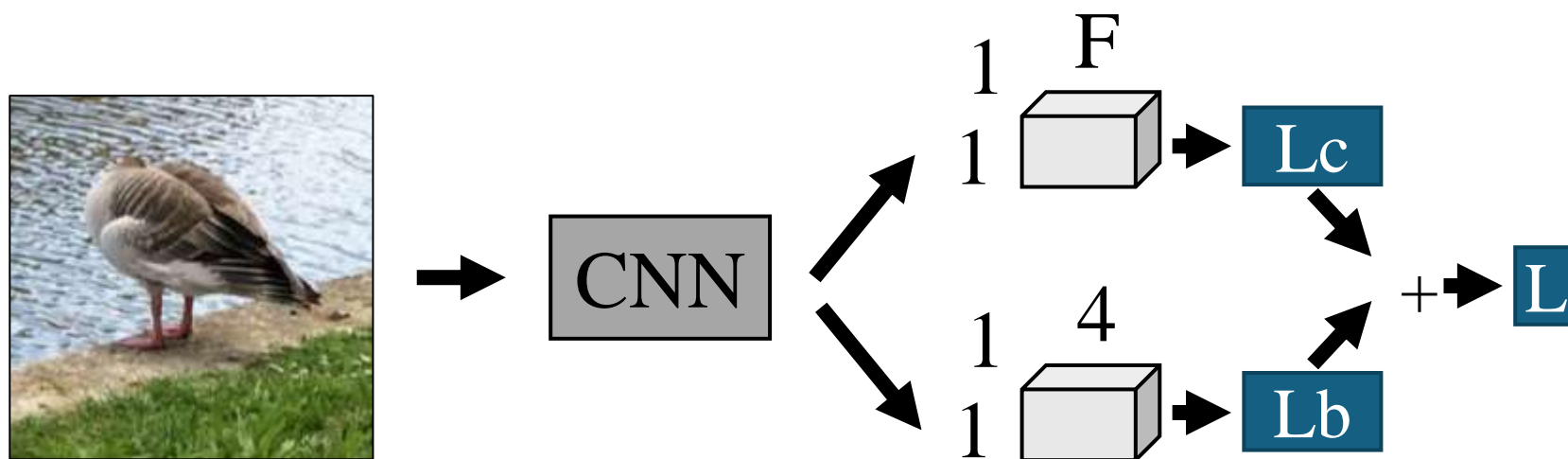
加上损失函数:

惩罚分类错误/定位错误

$L_c = \text{negative log-likelihood}$

$L_b = \text{L2 loss}$

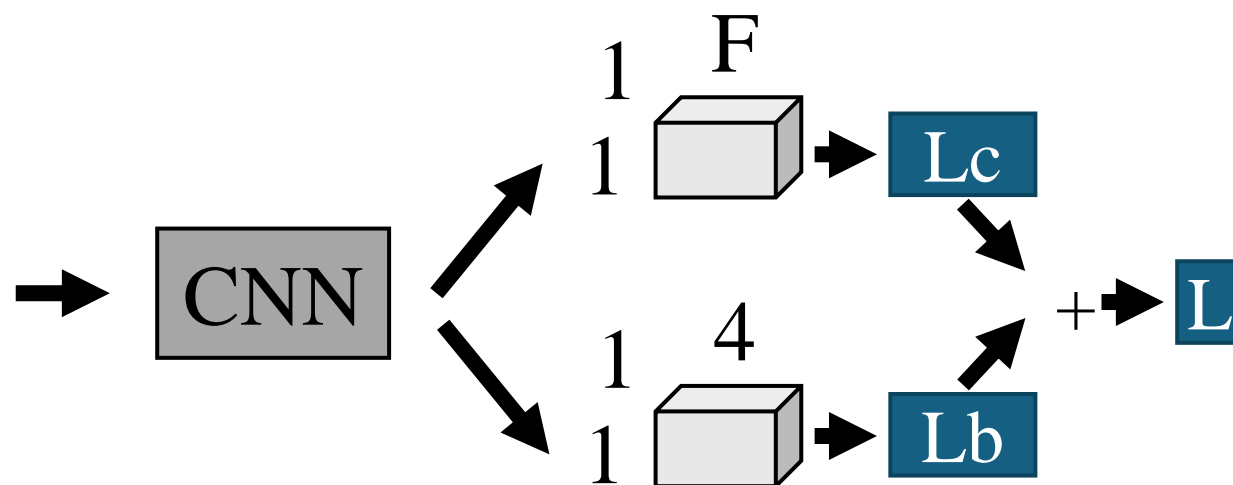
# 如何错误地做物体检测



计算误差，梯度反传  
最终的误差:  $L = Lc + \lambda Lb$

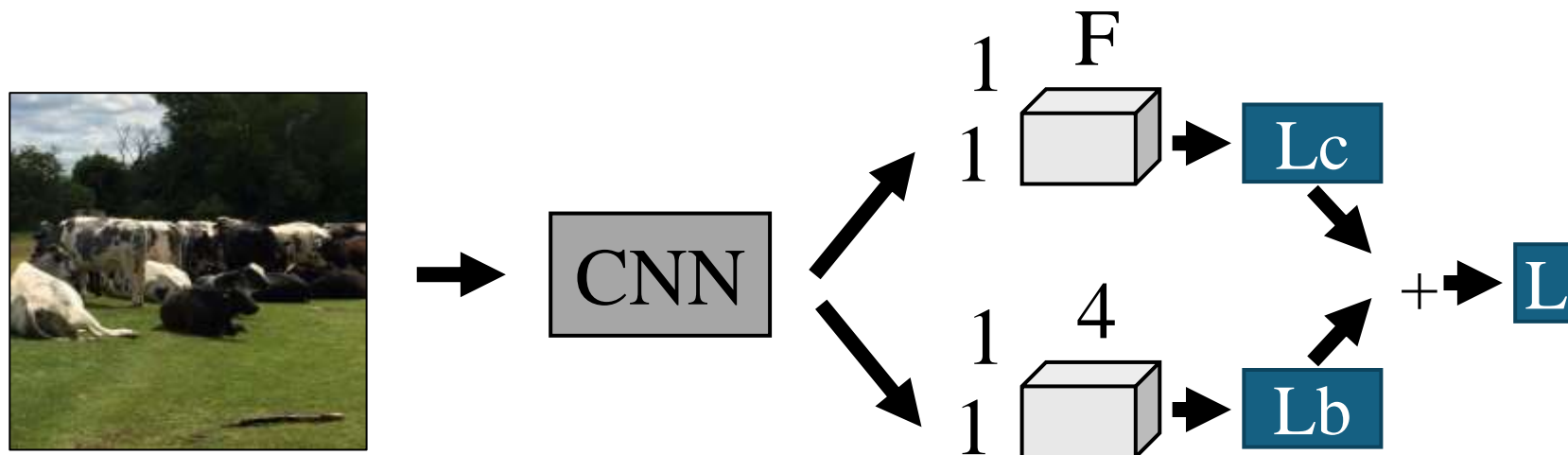
为什么会有 $\lambda$ ?

# 如何错误地做物体检测



现在游过两只鸭了。  
我们需要多少输出？  
 $F, 4, F, 4 = 2*(F+4)$

# 如何错误地做物体检测

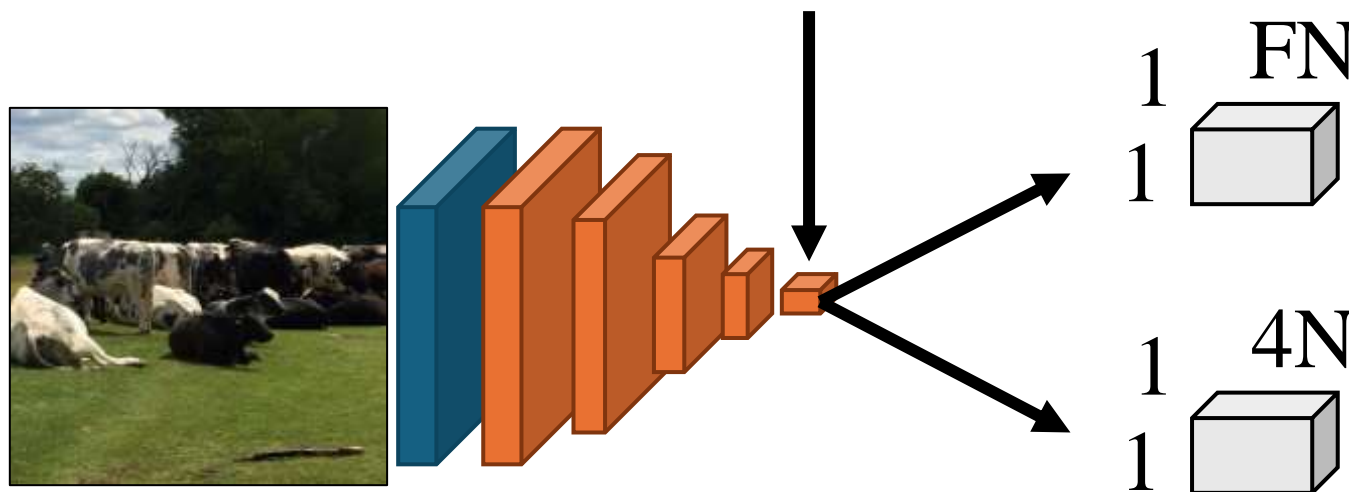


现在牛群来了。  
我们需要更多的输出  
(甚至根本不知道到底需要多少输出)。

# 总结问题——物体检测

- 我们不知道输出有多少个 因为它经常是在变化的
  - 就算我们可以，我们怎么让网络来预测呢.

关键在于怎么让网络知道这N个物体  
在哪



## 换个思路

检查所有窗口，看看这个窗口是否“紧紧包含”  
某一个物体



Yes



No?

记住这个窗口!



No

# 滑动窗口检测

假设我们在固定尺寸的窗口寻找行人



# 滑动窗口检测

在每个窗口上寻找...



# 滑动窗口检测

在每个尺度每个窗口上寻找...



Note – 固定尺寸的窗口

# 所有的窗口



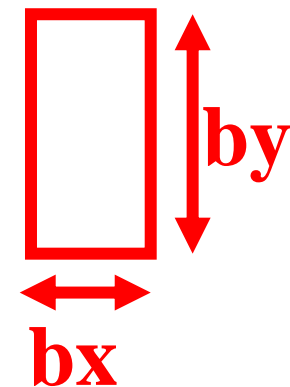
Slide credit: J. Hays

一共有多少窗口？

给定大小  $H \times W$  的图像和一个“template 模板”大小为  $b_y, b_x$ .

**Q.** 有多少框大小为  $(b_y, b_x)$ ?

**A.**  $(H - b_y) * (W - b_x)$



这是在考虑以下问题以前的答案:

- *scales* ( $b_y * s, b_x * s$ )
- *aspect ratios* ( $b_y * s_y, b_x * s_x$ )

# 物体检测的挑战

- 成千上万的框
- 却没几个有用的，包括物体的框极少

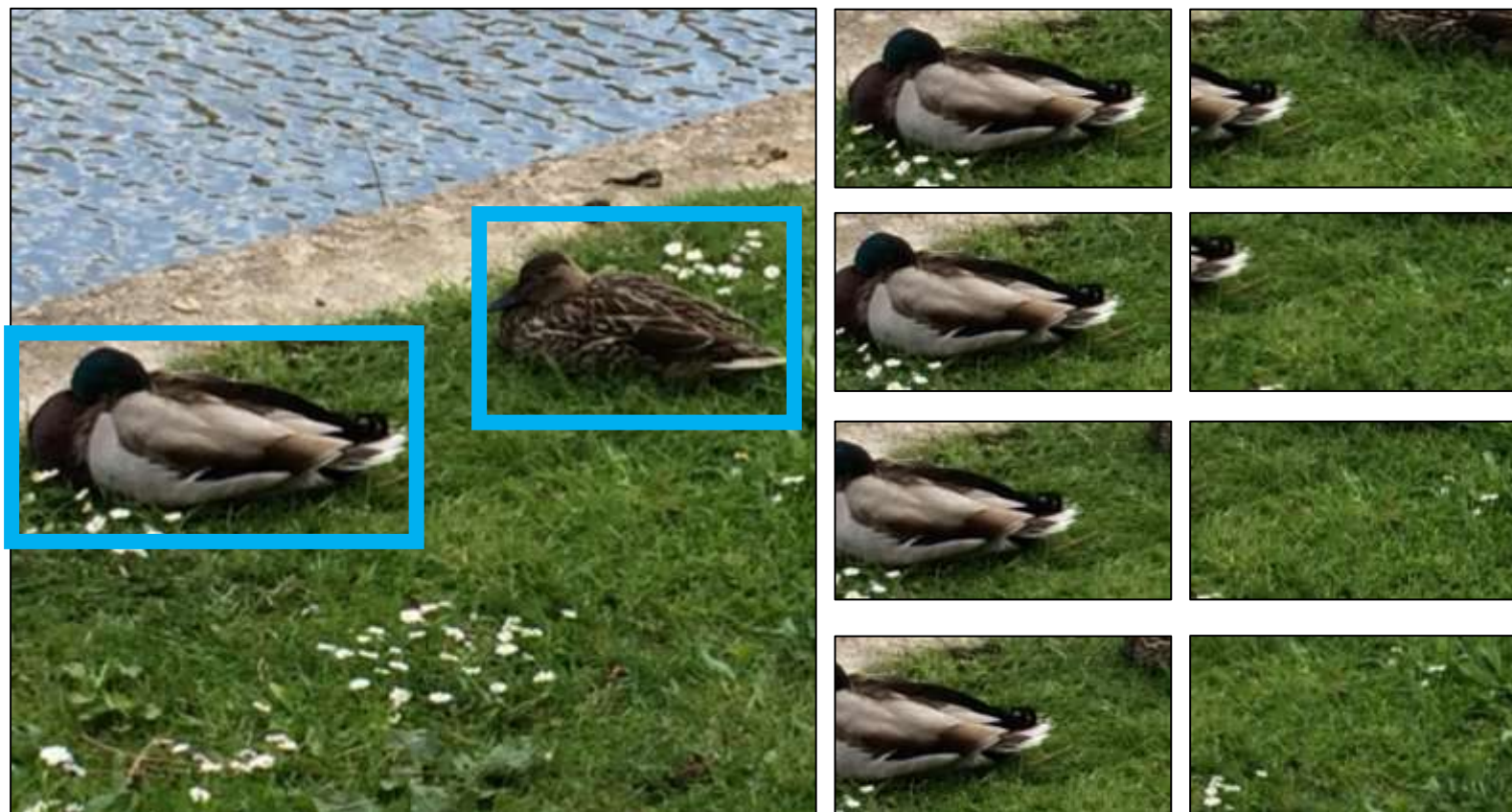


怎么整一个错误的框？

1. Wrong left x
2. Wrong right x
3. Wrong top y
4. Wrong bottom y

# 评估——边界框

告诉我什么时候你觉得检测不再正确了。



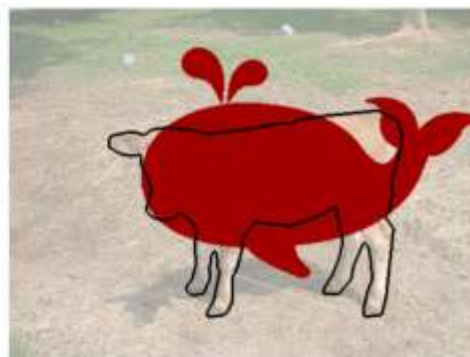
# 评估——边界框

标准的评价方法:

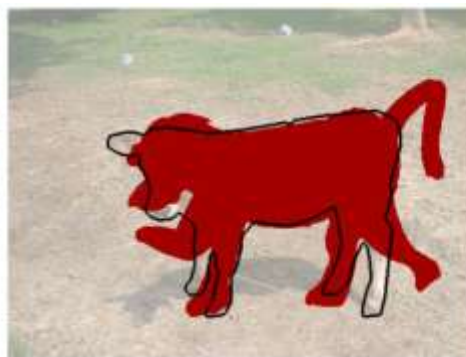
Intersection over union/IoU/Jaccard coefficient



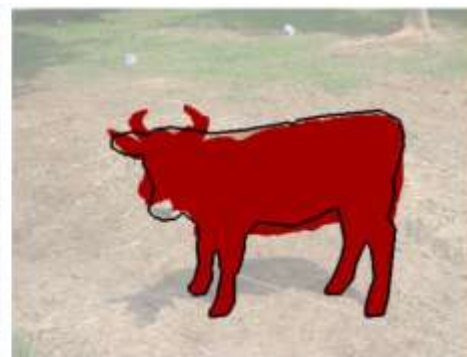
(a) Ground truth



(b)  $\mathcal{J} = 0.554$



(c)  $\mathcal{J} = 0.703$



(d)  $\mathcal{J} = 0.910$

# 知识问答



你比五年级小学生聪明吗？

成年人跟五年级小学生比小学知识竞赛。

成年人不见得能赢，因为你学的知识没有用！

# CV 知识问答



你的模型比随机数生成器厉害吗?

训练后的模型与随机模型相比.

如果你的评价指标不够真实, 其实并不一定哪个模型谁更好lol.

# 你真的比随机数生成器更智能？

- 求概率：1000类分类正确？
  - $1/1,000$
- 求概率：猜边界框的四个角，且误差小于10% 图像大小？
  - $(1/10)*(1/10)*(1/10)*(1/10)=1/10,000$
- 两者概率叠加:  $1/10,000,000$
- 不能用最常出现的标签（没有物体）来直接预测

# 性能评估

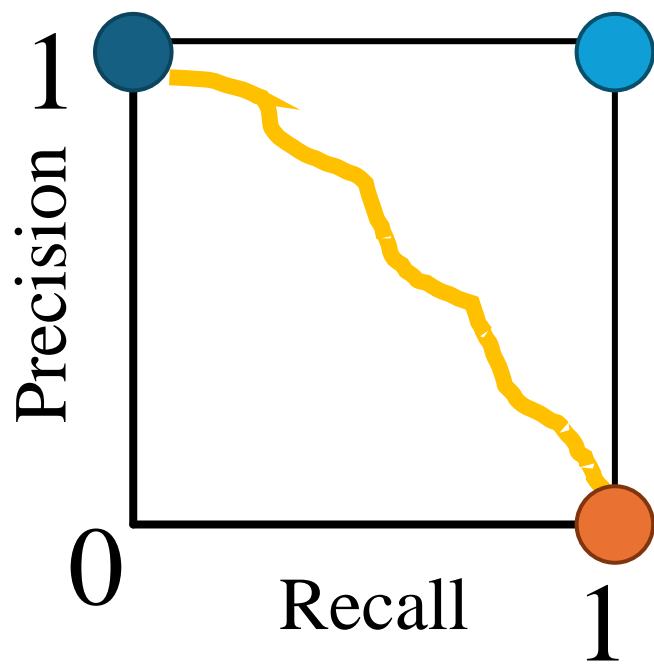
- 准确率 = 预测正确的概率
- 如果我声称我的行人检测99%正确.
- **这代表我的检测做的很好吗?**
- 大街上没有多少人!



# 性能评估

- True detection 真实检测 (true positive) : 高 IoU
- Precision 精确度:  $\frac{\text{\#真实检测}}{\text{\#检测总数}}$
- Recall 召回率:  $\frac{\text{\#真实检测}}{\text{\#真实的物体数量}}$

如果拒绝一切，那么就不会有错误

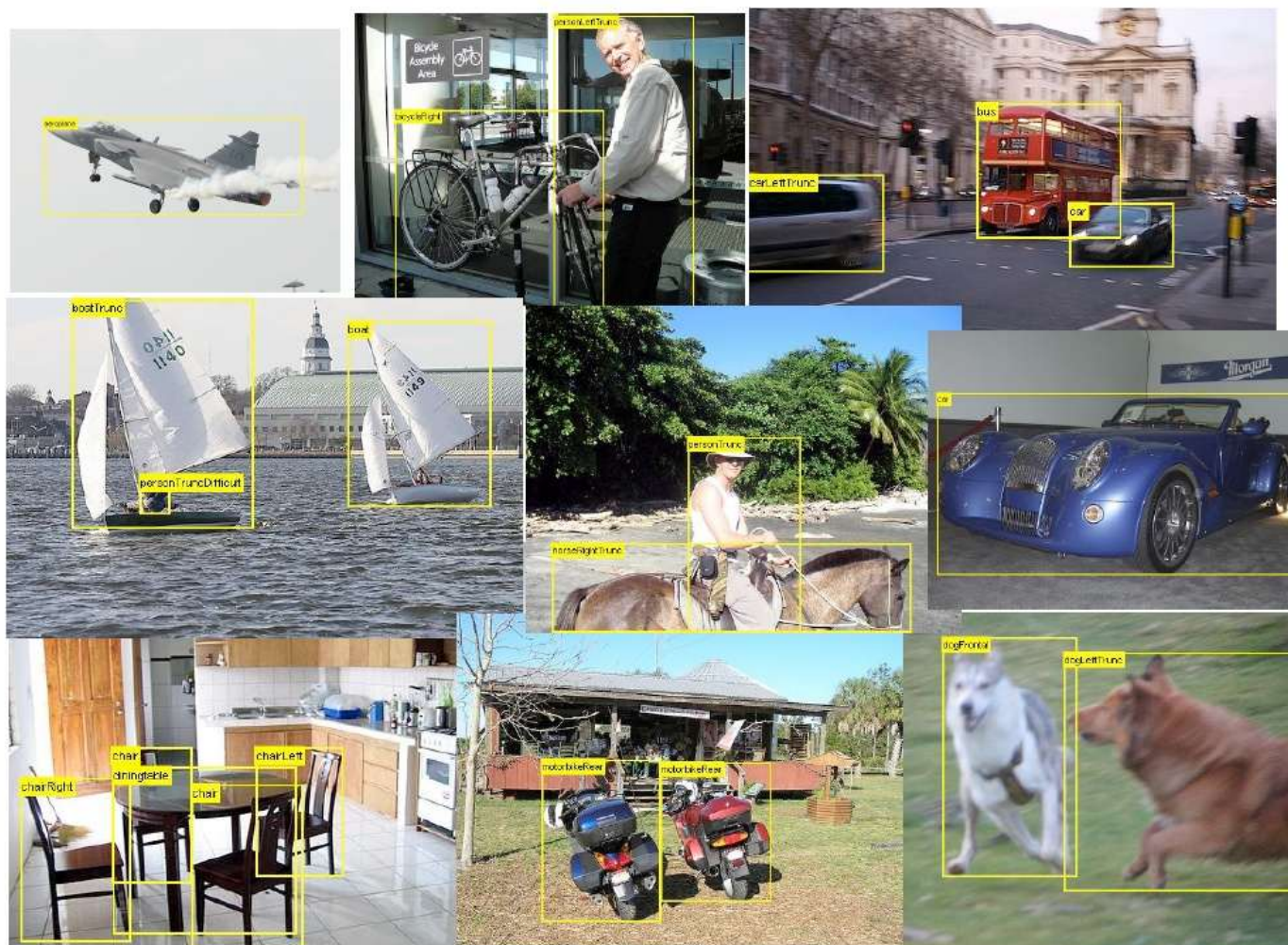


理想情况是精确度和召回率都很高!

**AUC area under curve (avg. precision)**

如果接受一切，那么就不会错过任何东西

# 各式物体检测



# 传统方法: Histograms of oriented gradients (HOG)

将图像分割成多个块, 并在每个块中计算梯度方向的直方图

$H \times W \times 3$  Image



$H' \times W' \times C'$  Image

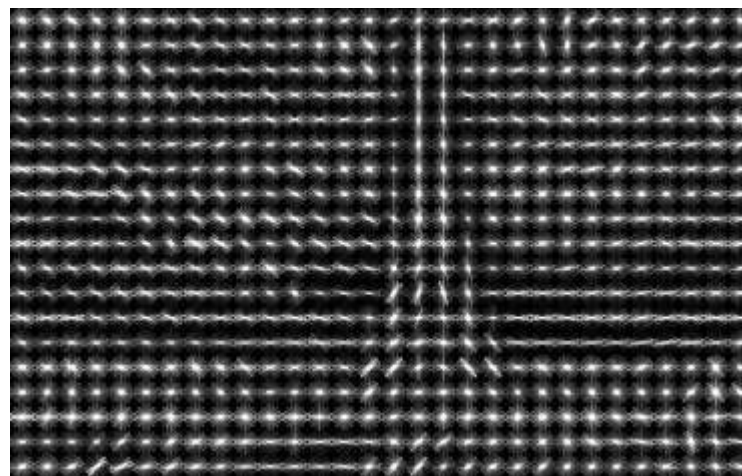


Image credit: N. Snavely

N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](#), CVPR

Slide Credit: S. Lazebnik

2005

# 使用HOG进行行人检测

- 使用线性支持向量机（SVM）来训练一个行人模板

正训练样本：包含行人



负训练样本：没有行人

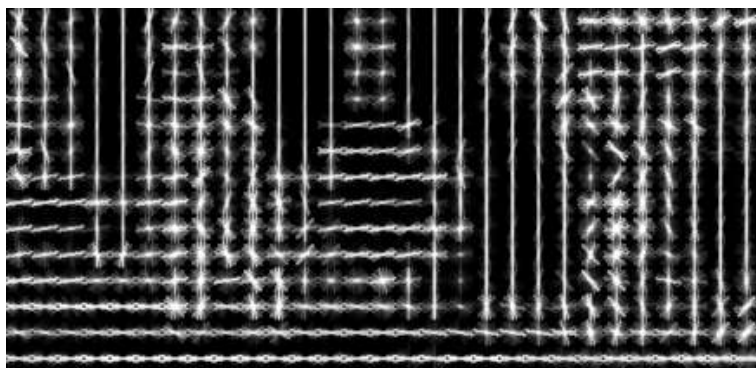


N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](#), CVPR

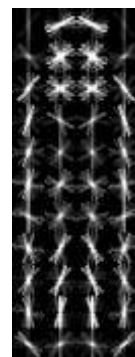
# 使用HOG进行行人检测

- 用线性SVM训练行人“模板”
- 测试时对HOG特征图用模板做卷积
- 找到局部最大响应值
- 不要忘了应对多尺度： HOG *pyramid* 金字塔

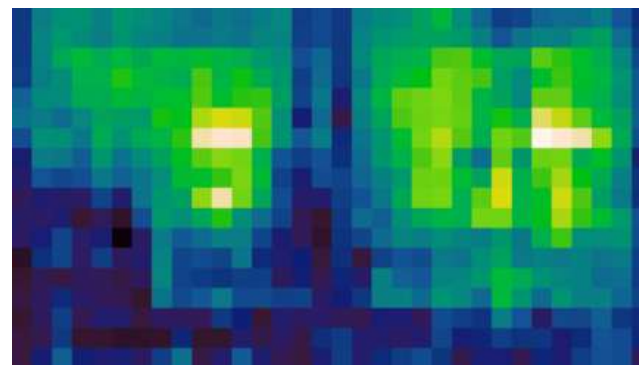
HOG 特征图



模板



检测器响应图



# 检测效果



[Dalal and Triggs, CVPR 2005]

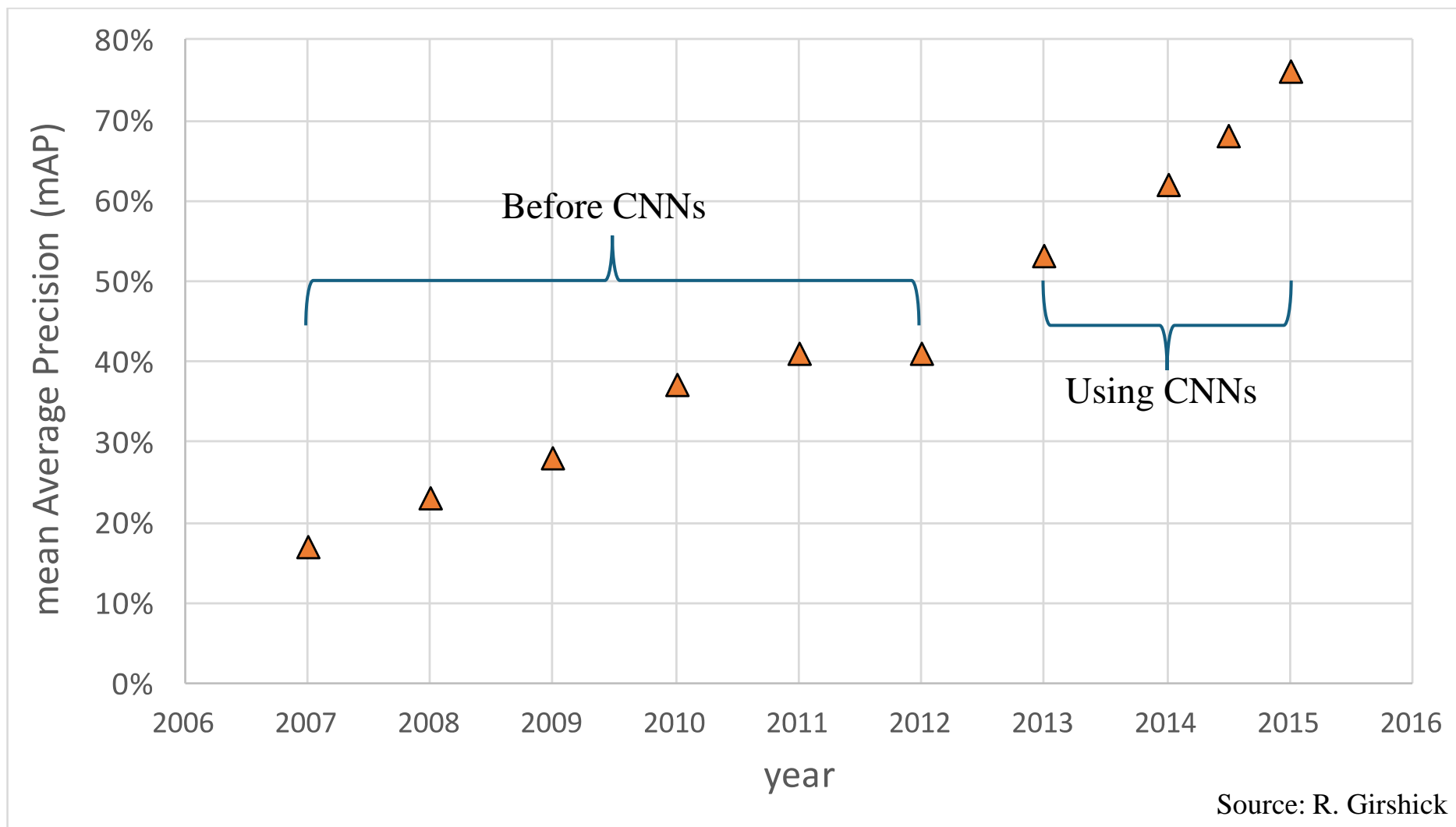
# PASCAL VOC 数据集 (2005-2012)



- 20 个类别:
- *Person*
- *Animals*: bird, cat, cow, dog, horse, sheep
- *Vehicles*: aeroplane, bicycle, boat, bus, car, motorbike, train
- *Indoor*: bottle, chair, dining table, potted plant, sofa, tv/monitor
- 数据集大小 (2012年): 11.5K 训练/校验图像, 27K 边界框, 7K 分割图

# 物体检测的性能发展

PASCAL VOC



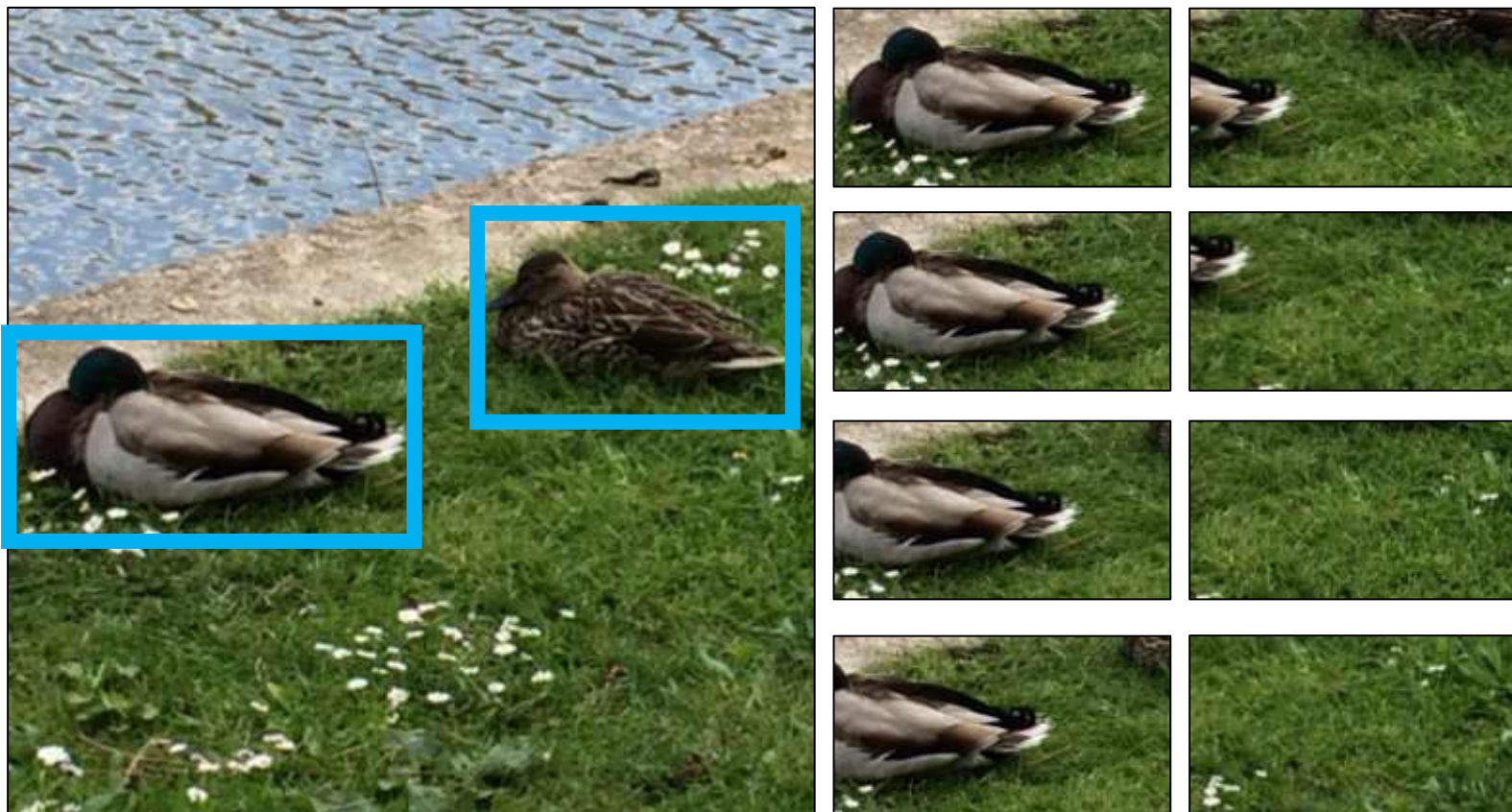
回顾：所有的窗口——只有少量窗口有目标物体



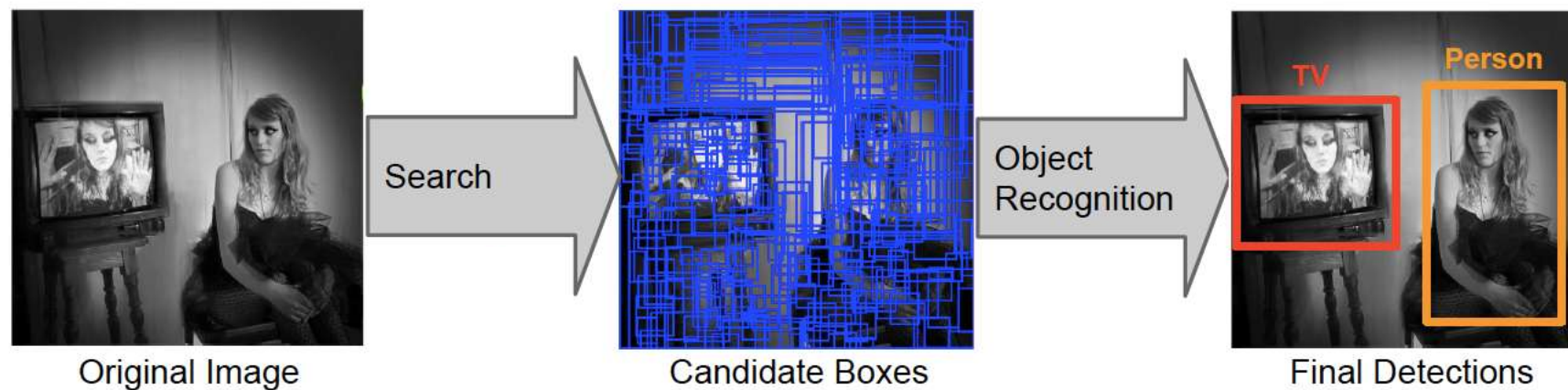
Slide credit: J. Hays

# Region Proposals 区域提议

我们需要花时间对草地的所有框进行筛选吗？

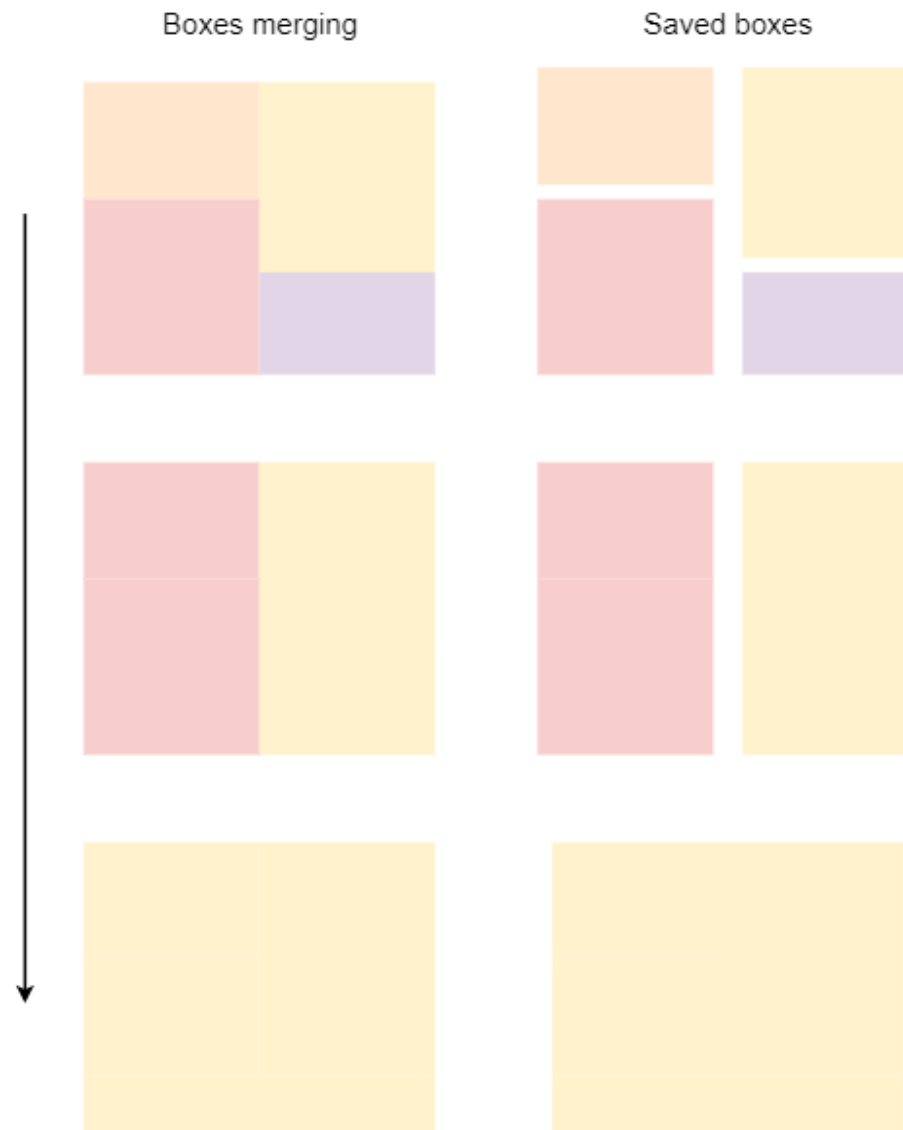


# Region Proposals 区域提议



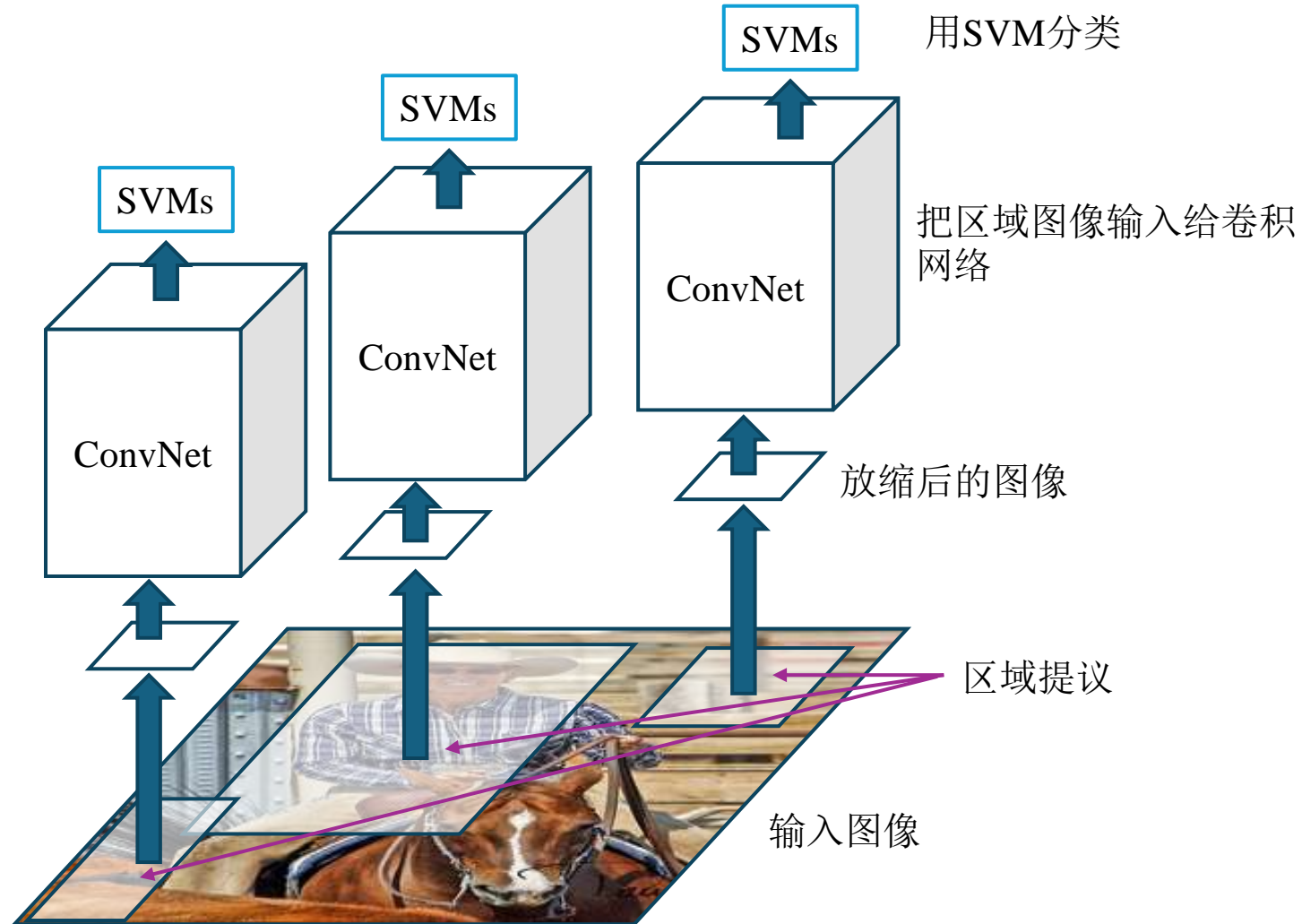
- 替代滑动窗口，只选出部分候选框作为 *region proposals* 区域提议
  - 谨记：特征提取和分类器速度较慢，区域提议可以减少候选区域，提升运行效率
  - 一般候选区域类别无关
  - 这个机制可以被训练（后面会讲）

# Region Proposals: Selective Search



# R-CNN: Region proposals + CNN features

Source: R. Girshick



## 第十周周三（5.6）课程大作业开题汇报

- 每组汇报总时长为 **5–8 分钟**，包含展示、提问与切换时间；**不因组员人数增加而额外延长**。
- 每组汇报结束后，由**助教进行简短提问**；同学们也可就大作业中的问题进行交流。
- 在本次汇报前，各组需**基本确定选题、数据集和组内分工**。
- 选择**自主选题**的组，需**提前与老师或助教沟通确认**，并在本次汇报中说明选题内容、研究目标、数据来源和实施计划。
- 本次汇报为**阶段性汇报**，重点考察**选题思路、方案设计与可行性**，**不要求已经完成全部实验结果**。