卷积神经网络

0：引言

上节课的T型结构刚开了头 → CNN要在CV江湖中一统天下 →

天下武功出少林？

CNN能干啥？

这么厉害吗？

这江湖有啥BOSS呀？

还有啥门派呢？

CNN到底是啥？

CV江湖中
天下武功出卷积！

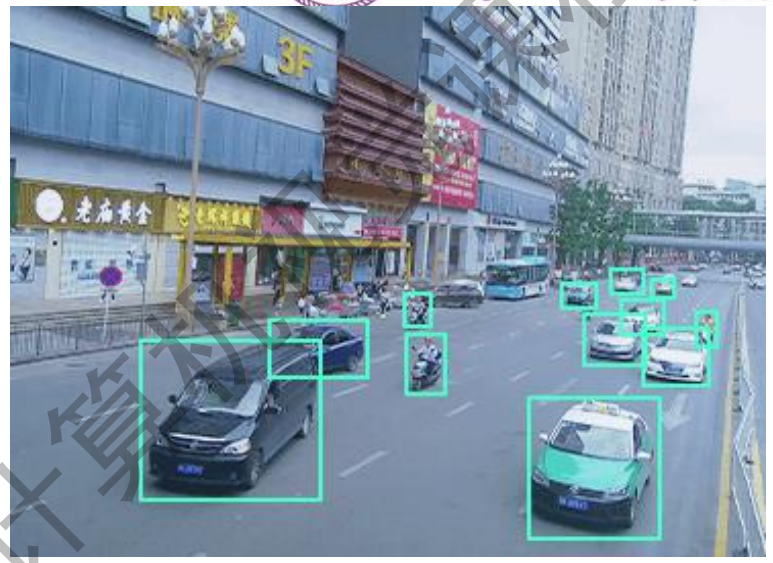
法律还能阻止他不？

还有啥CNN解决不了的吗？

CNN凭啥这么厉害？

1.1：CNN应用概览

分类
（classification）

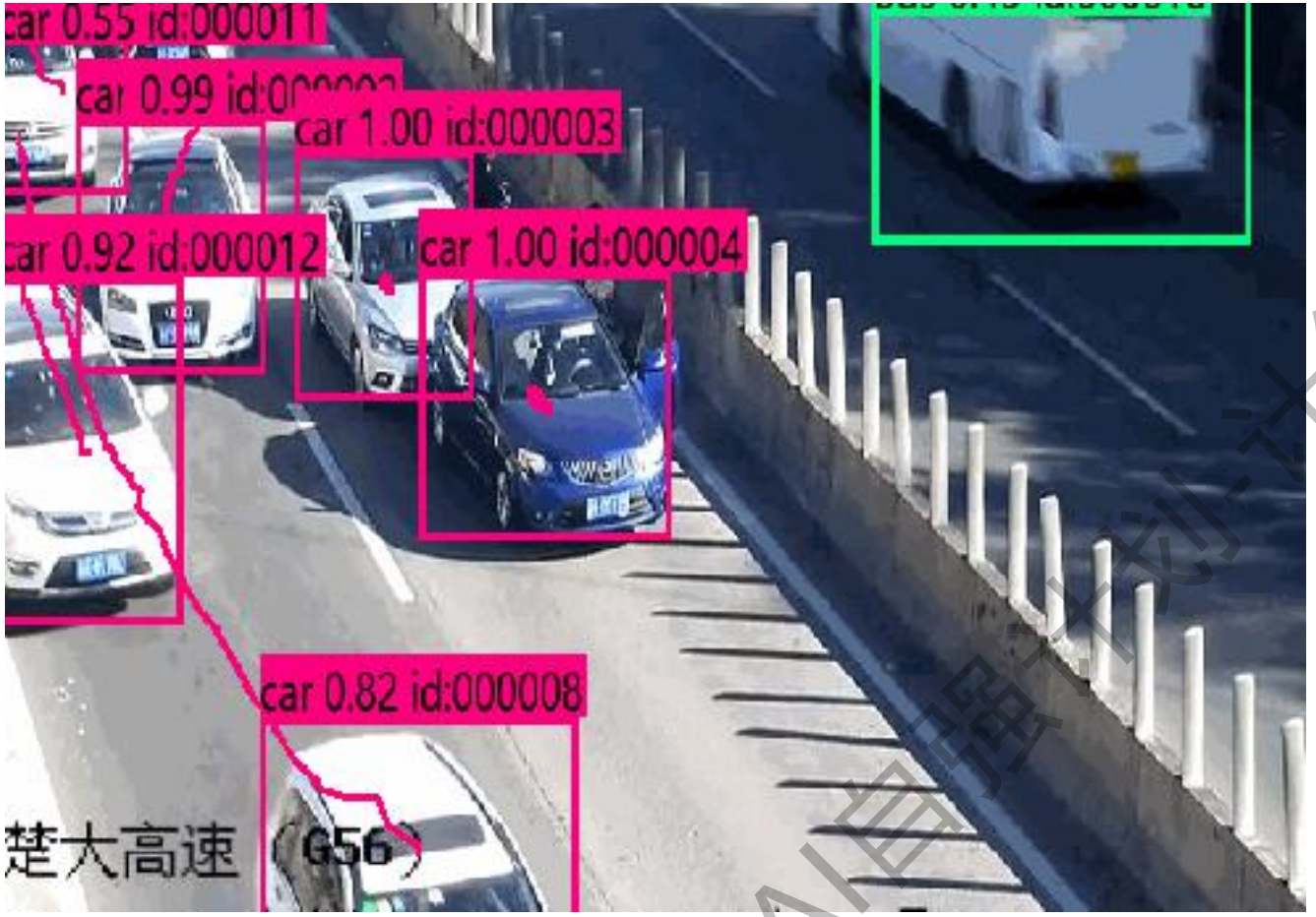


探测
（detection）

语义分割
（semantic segmentation）

实例分割
（Instance segmentation）

1.1： CNN应用概览



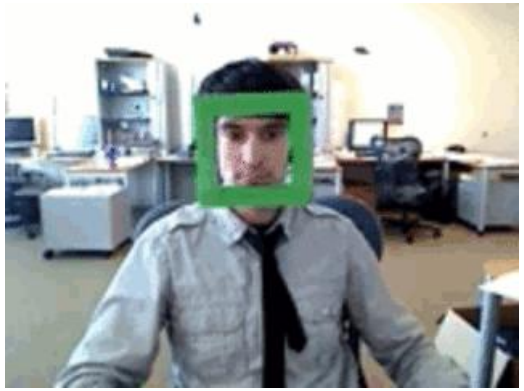
跟踪
（Tracking）

OCR
（字符识别）

1.1：CNN应用概览-什么落地最好？

CV四大独角兽都从“人脸”起家

SENSETIME 商汤科技



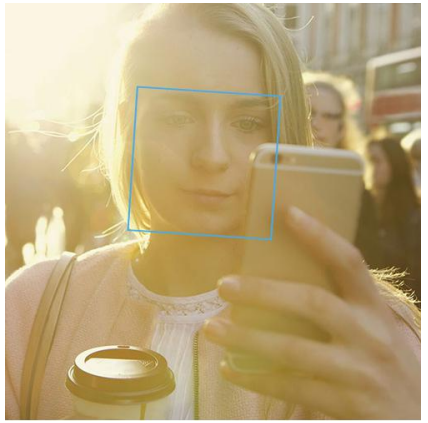
Face++ 旷视

依图 | YITU

云从科技 CLOUDWALK

Human Task

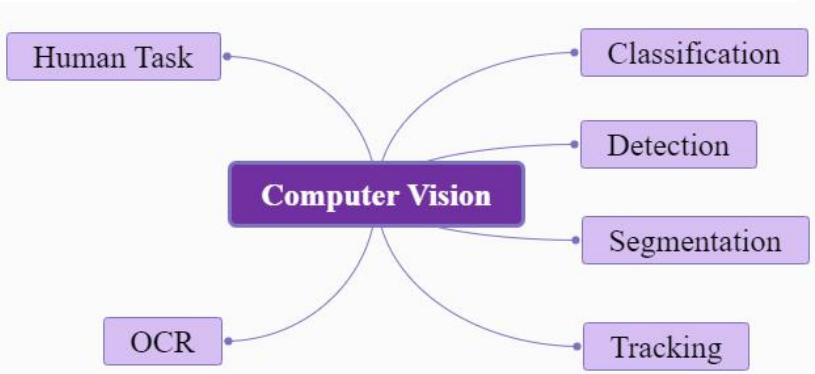
美颜



人脸解锁

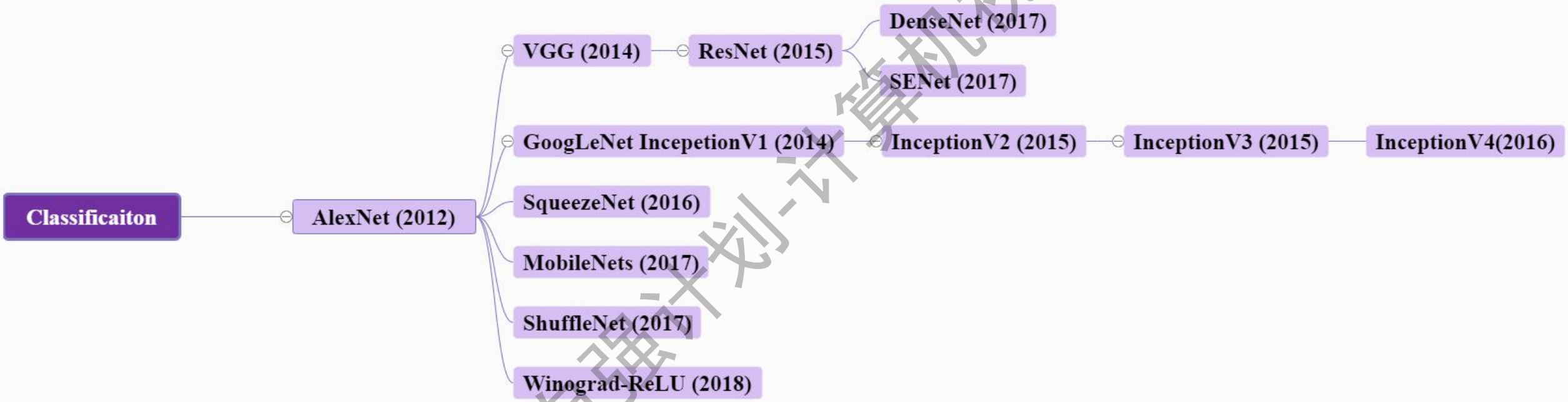
抖音-尬舞机
人体姿态识别

1.2：CNN技术概览

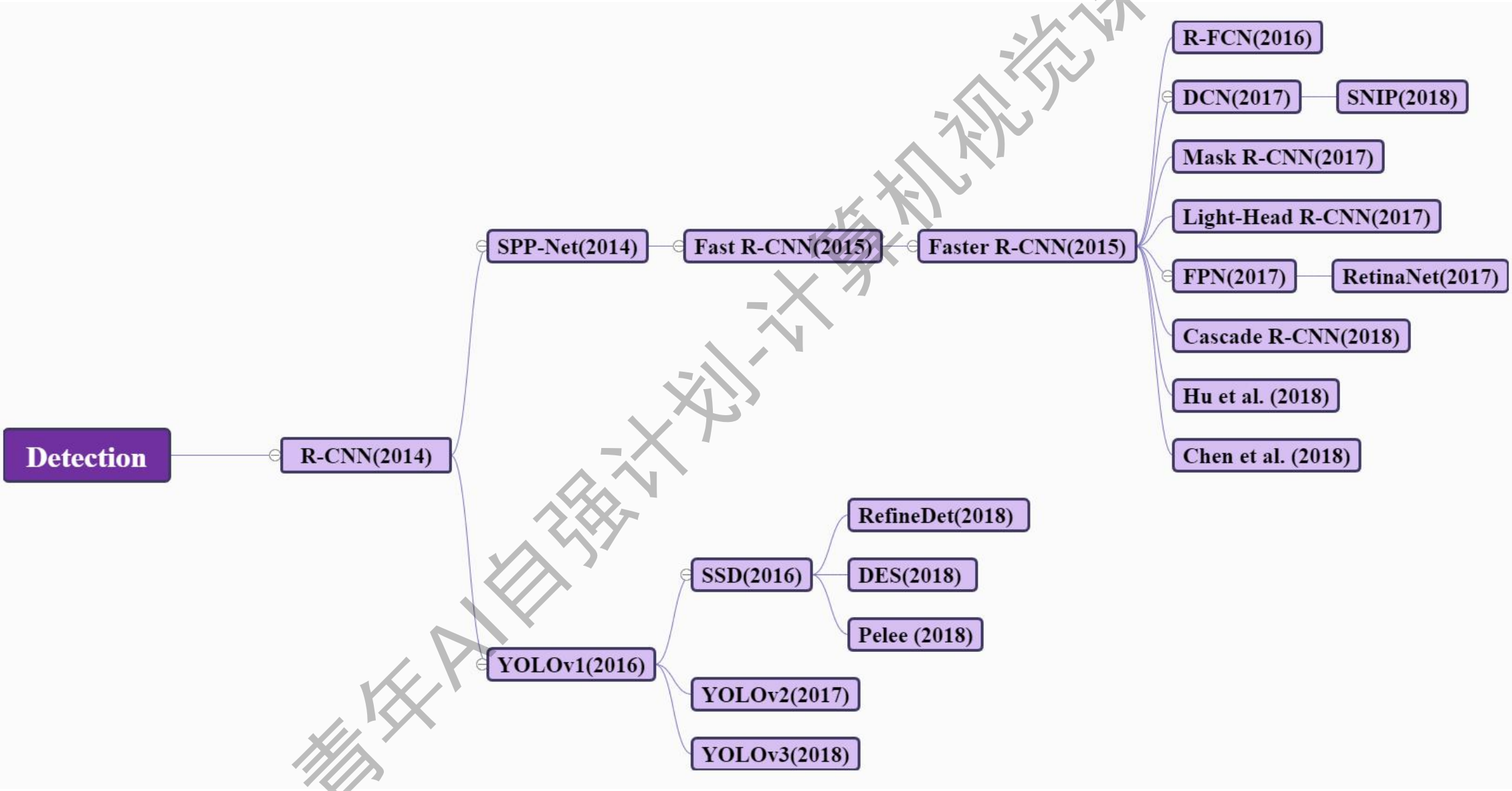


- Computer Vision
  - Classification
    - AlexNet (2012)
    - VGG (2014)
    - ResNet (2015)
      - DenseNet (2017)
      - SENet (2017)
    - GoogLeNet IncepetionV1 (2014)
    - InceptionV2 (2015)
    - InceptionV3 (2015)
    - InceptionV4(2016)
    - SqueezeNet (2016)
    - MobileNets (2017)
    - ShuffleNet (2017)
    - Winograd-ReLU (2018)
  - Detection
    - R-CNN(2014)
    - SPP-Net(2014)
    - Fast R-CNN(2015)
    - Faster R-CNN(2015)
      - R-FCN(2016)
      - DCN(2017)
      - SNIP(2018)
      - Mask R-CNN(2017)
      - Light-Head R-CNN(2017)
      - FPN(2017)
      - RetinaNet(2017)
      - Cascade R-CNN(2018)
      - Hu et al. (2018)
      - Chen et al. (2018)
    - YOLOv1(2016)
      - SSD(2016)
        - RefineDet(2018)
        - DES(2018)
        - Pelee (2018)
      - YOLOv2(2017)
      - YOLOv3(2018)
  - Segmentation
    - Semantic Segmentation
      - FCN (2014)
        - DeepLabV1 (2014)
        - DeepLabV2 (2016)
        - DeepLabV3 (2017)
        - DeepLabV3+ (2018)
        - SegNet (2015)
        - U-Net (2015)
        - RefineNet (2016)
        - PSPNet (2016)
        - ICNet (2017)
        - LinkNet (2017)
        - EncNet (2018)
        - PAN (2018)
        - DFN (2018)
        - FCAN (2018)
    - Instance Segmentation
      - Instance FCN (2016)
      - FCIS (2016)
      - Mask R-CNN (2017)
      - MaskLab (2018)
  - Tracking
    - MDNet(2016)
    - SiameseFC（2016)
    - GOTURN(2016)
    - ECO（2017)
    - CFNet(2017)
    - DaSiamRPN（2018)
  - OCR
    - Text detection
      - horizontal direction
        - CTPN (2016-CNN+RNN)
        - TextBoxes (2017-Based on SSD)
      - Multi-Oriented
        - RRPN (2017-Based on faster rcnn)
        - SegLink (2017-Based on SSD)
        - EAST (2017-Based on DenseBox)
        - DDRN (2017)
        - Wu et al. (2017-Based on FCN)
        - SSTD (2017-Based on SSD)
        - WordSup (2017)
        - TextBoxes++ (2018-Based on TextBoxes)
        - RRD (2018)
        - Lyu et al. (2018)
    - Text Recognition
      - RARE (2016)
      - DTRN (2016)
      - R2AM(2016)
      - FAN(2017)
      - CRNN (2017)
      - AON(2018)
    - Text Spotting
      - Li et al. (2017)
      - Deep TextSpotter (2017)
      - STN-OCR (2017)
      - SEE (2018)
      - FOTS (2018)
      - Textspotter(2018)
  - Human Task
    - Person Re-Identification
      - FPNN
      - HIPHOP
    - Human Detection
      - TA-CNN
      - SDN
    - Human pose estimation
      - DensePose
      - hourglass
    - Face Task
      - Face Detection
        - S3FD
        - MTCNN
      - Face Recognization
        - face++
        - DeepFace
        - DeepID
      - Face Alignment
        - 3DDFA
        - DenseReg
        - TCDCN)
      - Face AttributeComputing
        - LeanFace
        - AttNet
        - GTNN
      - head pose estimation
        - Hopenet
        - 3DMM
        - JFA

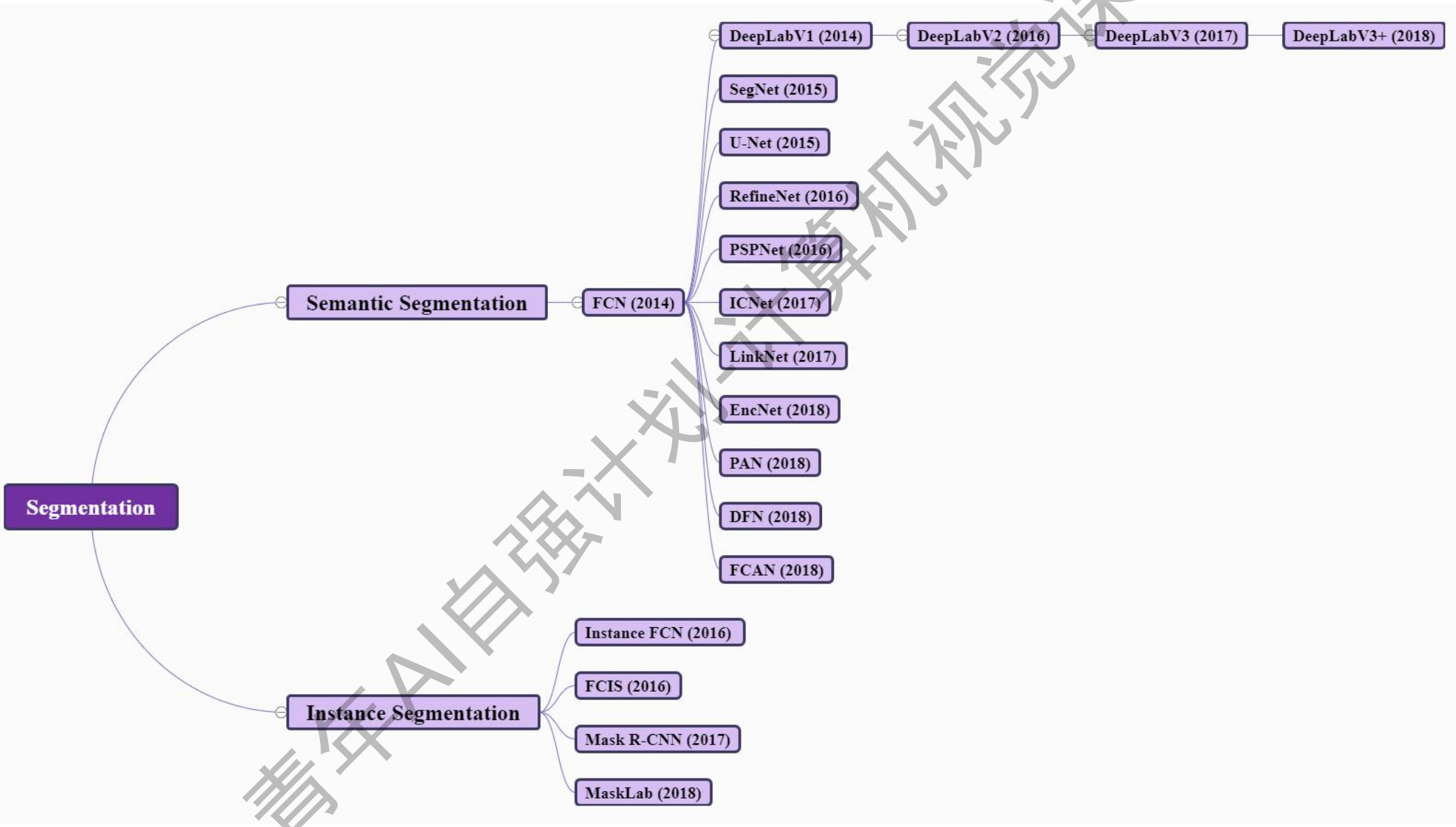
1.2：CNN技术概览



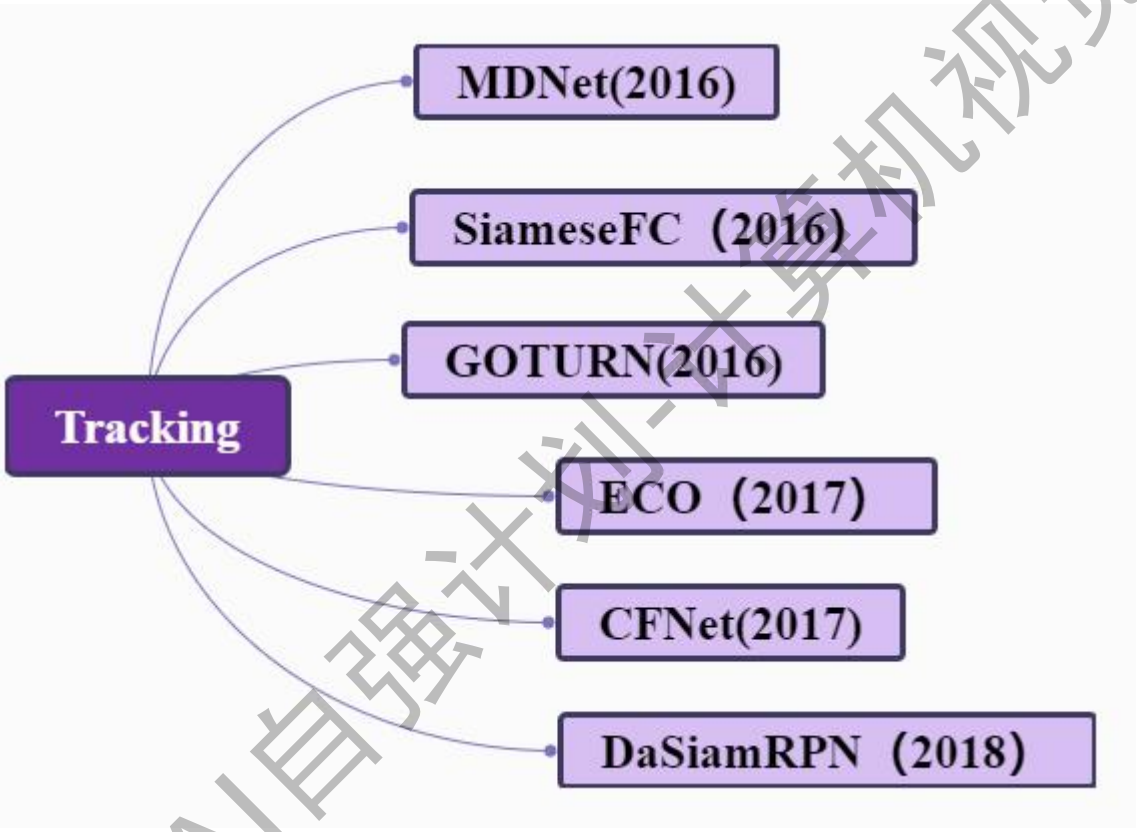
1.2：CNN技术概览



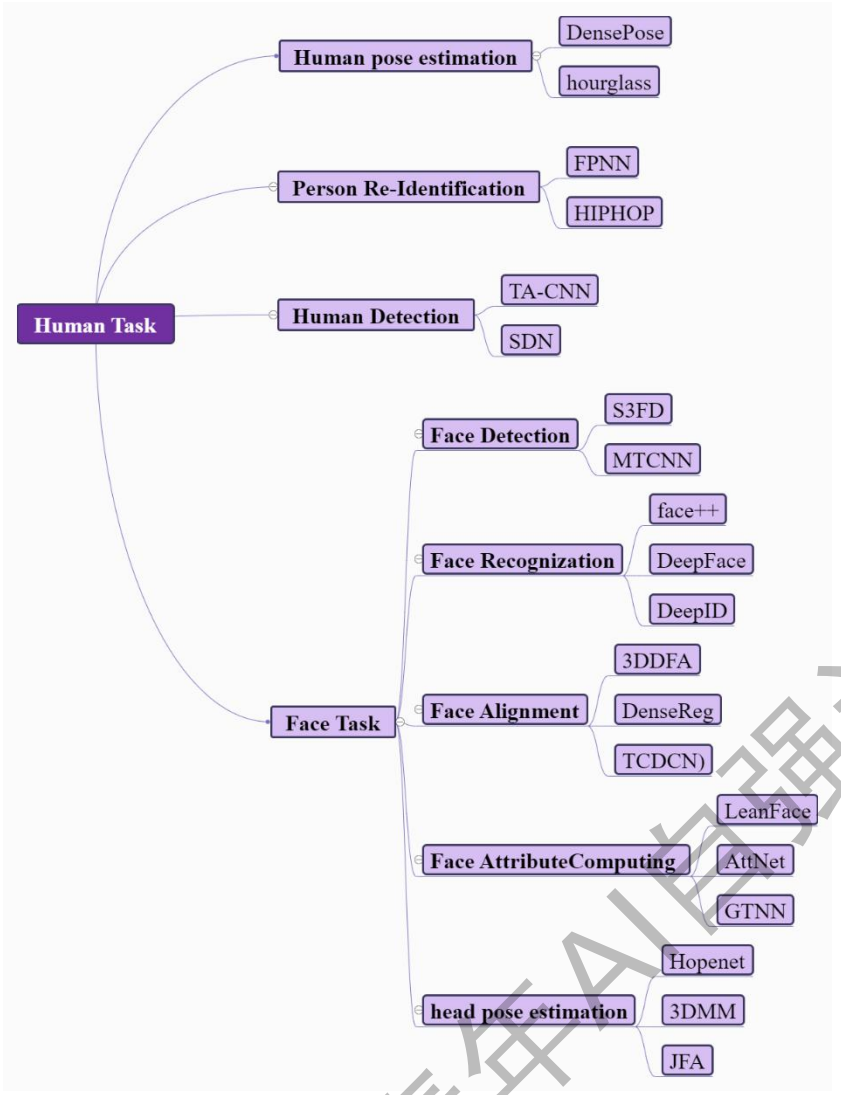
1.2：CNN技术概览

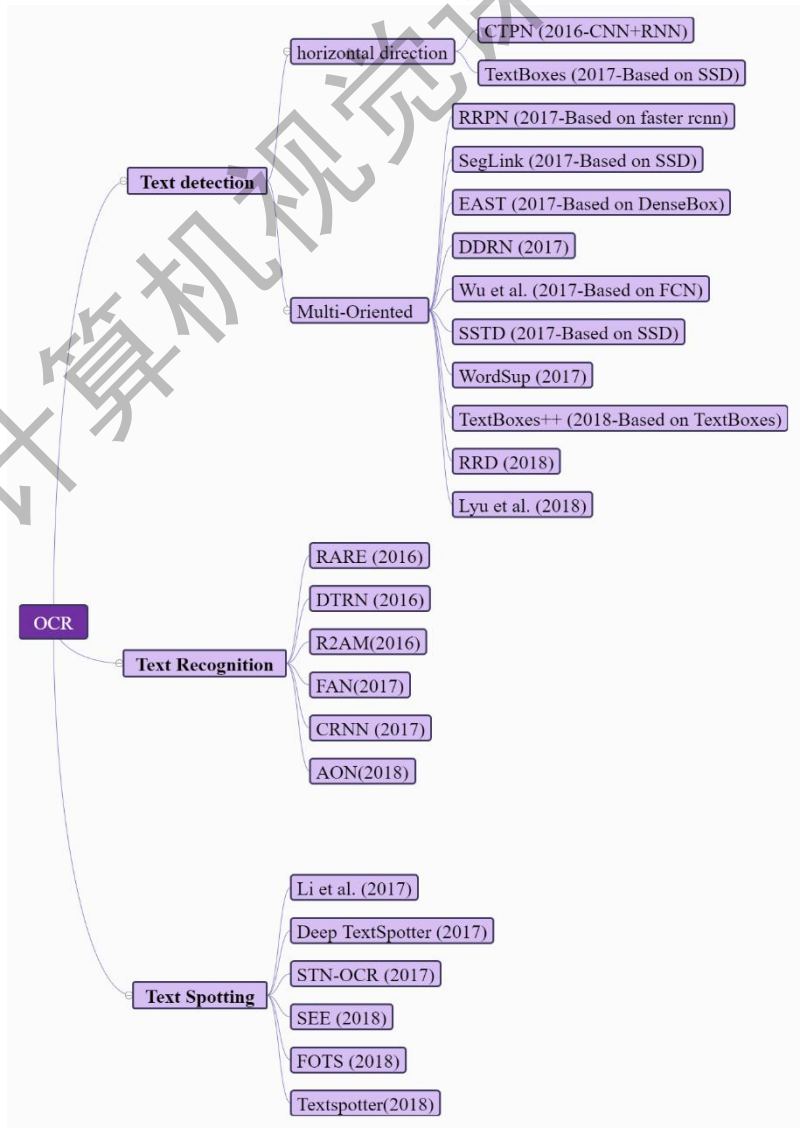


1.2：CNN技术概览



1.2：CNN技术概览





1.3：CNN未来的可能性

讲一个不太高兴的事让大家高兴一下

情绪的跌宕起伏

来到保安室后

1.3：CNN未来的可能性

真实监控图像

1、嫌犯可能出现在这70个小时中的任意时刻

2、如果按照1:1的速度浏览，显然耗时太久并不合算

3、最大加速比是16倍，加速之后的视频时间是4.3小时

4、回顾一下真实可能的肇事场景，肇事时间可能只有不到半分钟

5、在16倍加速的情况下，整个肇事过程只留给我不到2秒的时间观察，2秒是0.0005小时，只占视频总长的0.01%

6、经过试验，人会高度集中注意力到重复性极强的画面后，会开始出现幻觉[捂脸]

这根本不是给人干的活，
决定花300块钱解救自己

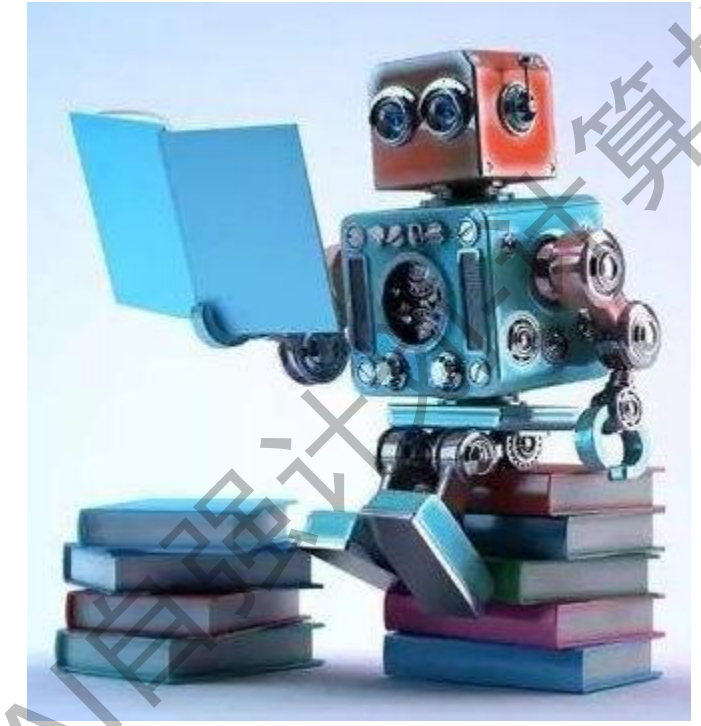
1.3：CNN未来的可能性

刚刚的问题
现在的算法无能为力

一张图片||
一段视频

现在的输入

理想中的输入



现在的输出

理想中的输出

标签labels

一个逻辑结果
及证明材料

一段逻辑：
谁 剐蹭 我车了？

把算法当作黑盒

1.3：CNN未来的可能性

生活中的通用CV需求

有谁借着出早操的名义
牵手谈恋爱的？

地中间的粑粑是谁拉的？

何时算法能够读懂
通过肢体语言表述的逻辑？

2.1：标准DNN的CV局限性-参数爆炸

是否要被上次的训练速度折磨疯了？
227*227灰度图，训练集约4.5万张

真实的图片要更加复杂
4160*2336彩色图，千万像素

```
A1 =    num_units=1024,
A2 =    num_units=2048,
A3 =    num_units=2048,
A4 =    num_units=1024,
```

结合上次作业的
模型结构算一笔账

上次简化训练情况

输入层约有
227×227 = 51,529个神经元

链接首个隐藏层的权值w个数为：
51,529×1,024 = 52,765,696个

用1个样本更新首层w，需要至少计算
105,531,392次，跑1个epoch需计算

105,531,392×45,000≈4兆7千5百亿次

CPU并行数约为核数，所以训练约需要3-5h

真实情况

输入层约有
4160×2336×3 = 29,153,280个神经元

链接首个隐藏层的权值个数为：
29,153,280 ×1,024 = 29,852,958,720个

用1个样本更新首层w，需要至少计算
59,705,917,440次，1个epoch需要计算

59,705,917,440 × 45,000 = 2,686,766,284,800,000 约为2700兆次，如果仍用CPU训练大概需要3-5个月

2.1：标准DNN的CV局限性-鲁棒性差

大家一起来找茬

大家眼中看起来没区别的两张图在计算机看来却又翻天覆地的变化

标准DNN提取特征的方式与人（平移不变性）有本质的差距

2.2：卷积（convolution）略解

卷　积：Conv
池　化：Pooling
全连接：FC

CNN的三种主要结构

好消息 →

FC结构
大家已经掌握

先说卷积是啥 →

卷积是一种运算

运算的三要素 →

1、发生关系的两个变量是啥
2、运算规则是啥
3、运算结果是啥

1、变量是啥？

What We See

What Computers See

简化 →

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

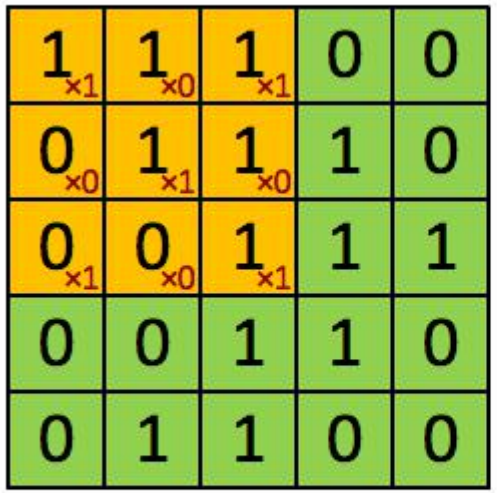
变量1：输入图像

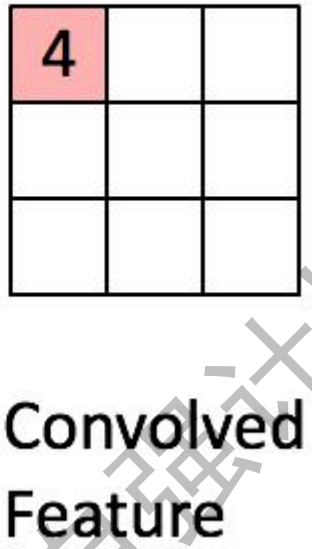
×1	×0	×1
×0	×1	×0
×1	×0	×1

变量2：卷积核

2.2：卷积（convolution）略解

2、运算规则是啥？





3、运算结果是啥？

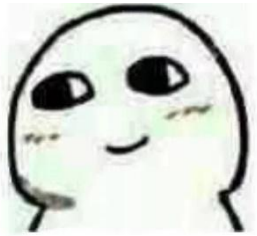
4	3	4
2	4	3
2	3	4

1、用卷积核从左到右，从上到下将需要处理的图像依次扫一遍；
2、每次被卷积核扫到的部分，与卷积核对应相乘再相加

一个新的矩阵
可以被称为Feature map

恭喜大家3分钟学会卷积运算

2.2：卷积（convolution）略解



微笑中透露着无奈

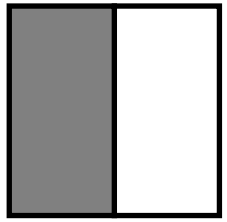
怎么感觉高兴不起来？

这玩意 → 分三个方面为大家解释 → *真的管用？*

1、卷积如何提取特征
2、与标准DNN相比的优点
3、与传统机器视觉算法相比的优点是啥

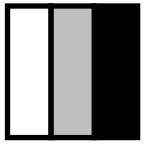
1、卷积如何提取图像中的特征

0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10



*

1	0	-1
1	0	-1
1	0	-1



=

0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0

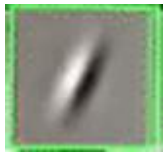
原画面中间的
边界特征被提取出来了

2.2：卷积（convolution）略解

一般的：不同取值的卷积核可以提取不同方向的边缘特征、不同种类的纹理特征颜色特征等

输入图像

卷积核1



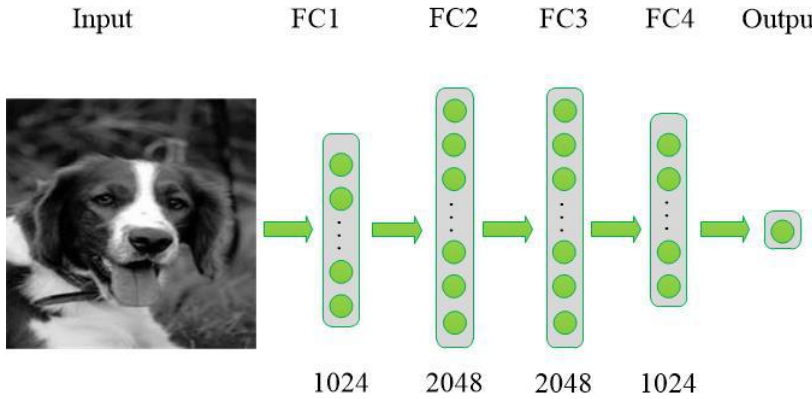
卷积核2

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}$	
Edge detection	$\begin{bmatrix}1&0&-1\\0&0&0\\-1&0&1\end{bmatrix}$	
	$\begin{bmatrix}0&1&0\\1&-4&1\\0&1&0\end{bmatrix}$	
	$\begin{bmatrix}-1&-1&-1\\-1&8&-1\\-1&-1&-1\end{bmatrix}$	
Sharpen	$\begin{bmatrix}0&-1&0\\-1&5&-1\\0&-1&0\end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$	

这与人类识别物体的原理是相仿的
我们人类也同样是通过识别物体的边缘来确定其形态进而对其分类的
比如同样出现在墙上，圆的一般是表，而方的一般是门窗

2.2：卷积（convolution）略解

2、卷积与标准DNN相比的优势：一个对比实验



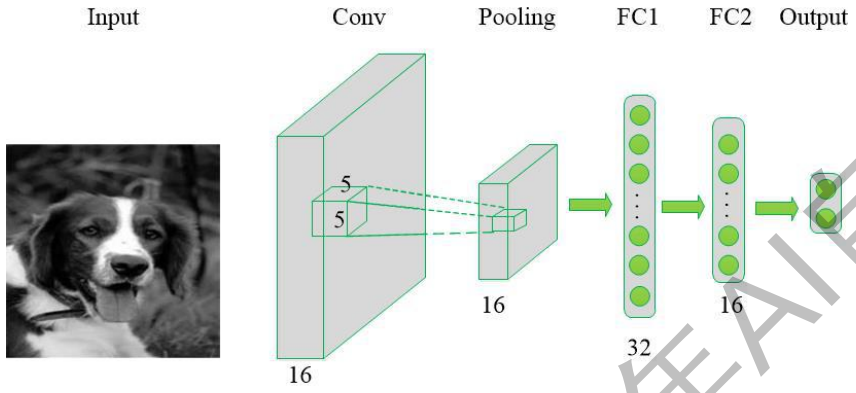
DNN的网络结构

精度相当，CNN比DNN的权值数量少了1.5个数量级

基本条件设置	对比结果	
	DNN	CNN
数据集：dot & not dog Batch-size：128 Lr：0.1（learning rate decay） Loss function：交叉熵 优化方法：Adam Batchnorm：使用 激活函数：relu	权值数量：61,161,473 检测精度：93%	权值数量：1,549,810 检测精度：94%

结果对比&结论

参数效率高：
随着输入图像尺寸的增大
a. 标准DNN参数指数型增加
b. CNN参数可以线性增加



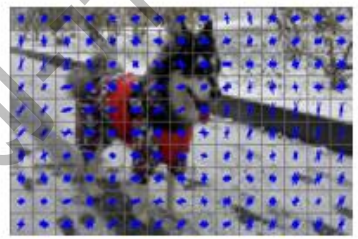
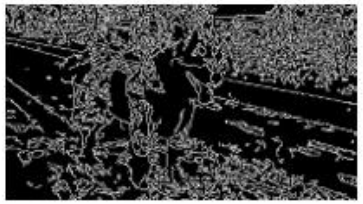
同等精度下简单CNN的网络结构

第一节课小实验，体会DNN和CNN的感受野

鲁棒性强：
a. 标准DNN首层1个参数感受1个像素点（1列像素，天差地别）
b. CNN一组参数在整个画幅感受一类特征（无垂直边缘改变，无差别）

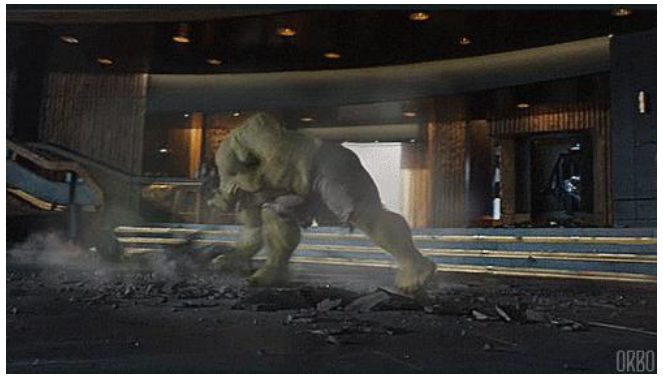
2.2：卷积（convolution）略解

“提取图边缘信息”来做特征表示，并不是CNN首创的，在他之前的传统机器视觉算法，大多符合这种指导思想。标准DNN曾经就一度败在他们手下

特征名称	功能描述	效果展示	
		特征提取图	原图
LBP	一种常用的纹理描述算子，用于度量和提取图像局部的纹理信息，对光照具有不变性。**LBP 结合 BP 神经网络**已经用于人脸识别等领域。		
HOG	通过计算和统计图像局部区域的梯度方向直方图来构成特征。Hog 特征结合 SVM 分类器已经被广泛应用于图像识别中，尤其在行人检测中获得了极大的成功。具有光照不变性，不具有尺寸和旋转不变性。**HOG+SVM 进行行人检测**的方法是法国研究人员 Dalal 在 2005 的 CVPR 上提出的。		
Haar	该算法是一种用于目标检测或识别的图像特征描述子，**Haar 特征通常和 AdaBoost 分类器组合使用**，且由于 Haar 特征提取的实时性以及 AdaBoost 分类的准确率，使其成为人脸检测以及识别领域较为经典的算法。		
Canny	该算法是目前最常用的边缘提取算法，通过计算梯度、非极大值抑制、双阈值和边界跟踪，可以完美的勾勒图像轮廓。		

2.2：卷积（convolution）略解

可还记得NLP
领域的学派大战



类似的

CNN是通过对大量经验数据的学习，
来对物体进行分类的“经验派”

传统机器视觉算法是希望通过规则，
描述物体的特征来对其进行分类的“规则派”

CNN的本质是神经网络，同样有FP和BP。卷积核中的权值都是“学”来的

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

之前例子中的卷积核

2012年的AlexNet开始，CV进入CNN时代

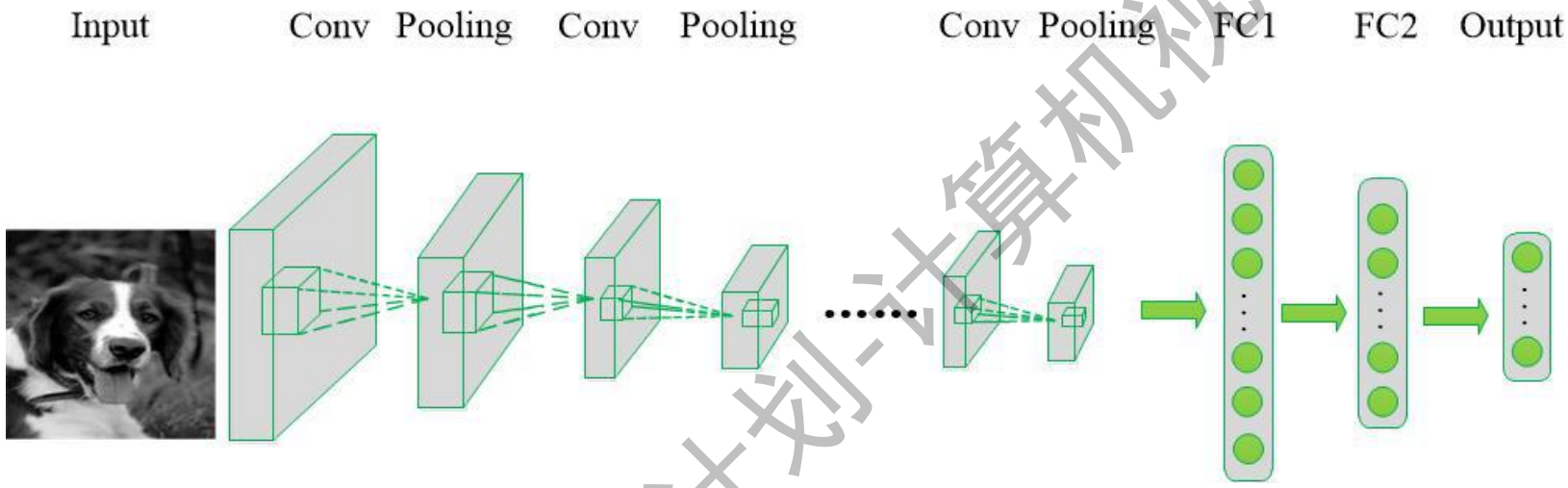
3、CNN与传统机器视觉算法的优势对比

优势一：源自NN的先天优势
CNN的特征提取时自动学习的，
每一个卷积核负责提取一类特征
无论在特征提取的全面性或者鲁棒性上都更优

2.2：卷积（convolution）略解

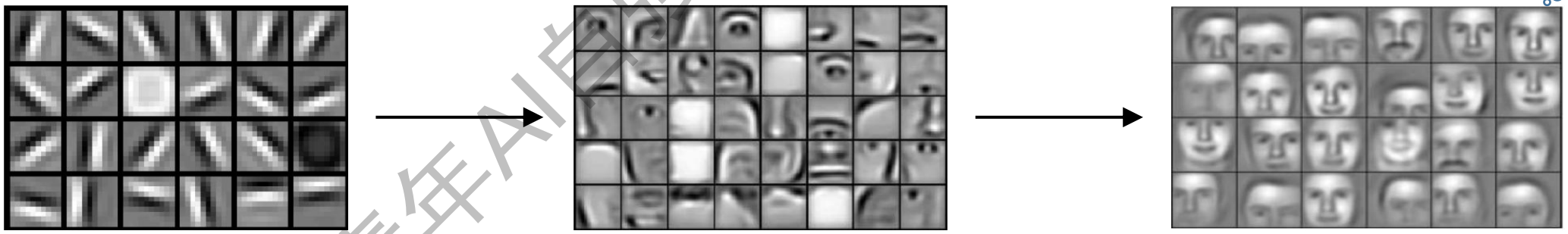
3、CNN与传统机器视觉算法的优势对比

优势二：源自CNN深度网络的先天优势
CNN的网络层数可以任意加深



以人脸识别为例，每层卷积负责提取不同抽象等级的特征

就是意思一下，
大家切勿较真



临近输入层　　*中间层*　　*临近输出层*

2.3：池化（pooling）略解

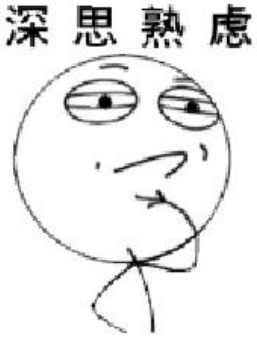
卷积还是挺厉害的呀

看看池化怎么样

同样分三个方面阐释

1、变量是啥
2、运算规则是啥
3、运算结果是啥

疑问



就这么扫一下能管用？

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
9	6	5	1	0
5	9	1	2	1

1.变量：输入的feature-map

2.运算规则：
扫一遍挑出最大的

Max pooling

没有参数需要更新，很满意吧？

1.变量：池化核

=

9	9	5
9	9	5
9	9	5

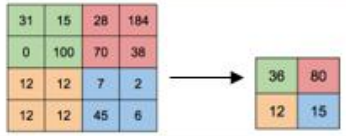
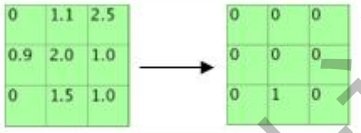
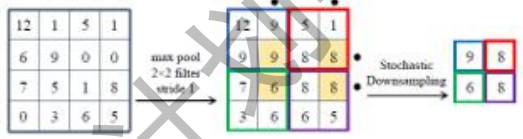
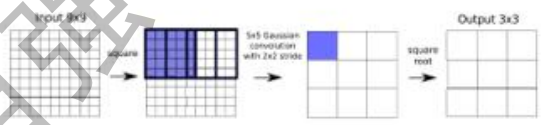
3.结果：输出新的feature-map

2.3：池化（pooling）略解

池化的作用1：减少参数总量
之前叫降采样（down sampling）

疑问

这么随便就把参数丢了吗？

Pooling 类型	描述	示意图
Average pooling	对邻域内的特征点求平均。	
Sum pooling	对邻域内的特征点求和（有三种不同版本）	
Stochastic pooling	对 feature map 邻域内的元素按照其概率值大小随机选择，即元素值大的被选中的概率也大。而不像 max-pooling 那样，永远只取那个最大值元素。	
S3Pool	首先采用最大池化得到激活值，然后随机行和列按照 grid 区域进行下采样。	 (c) S3Pool, pooling window $k = 2$, stride $s = 2$, grid size $g = 2$
Lp-Pooling	使用高斯核进行池化，具体操作如公式所示。O 表示池化输出，I 表示 feature map，G 表示高斯核。 $O = (\sum\sum I(i,j)^P \times G(i,j))^{1/P}$ P=1 表示简单的高斯平均。	 **Figure 2.** L2-pooling applied to a 9x9 feature map with a 3x3 Gaussian kernel and 2x2 stride

结果

是否记起了
OVA的例子？
简单≠low

除了一些特殊情况外
普遍采用的都是Max pooling

许多算法科学家与大家有同样的想法

2.3：池化（pooling）略解

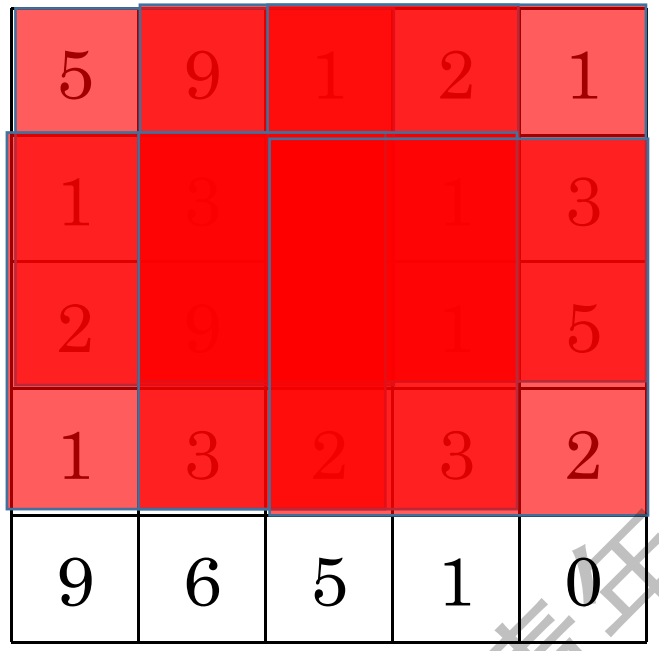
池化的作用2：平移不变性---小实验

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
9	6	5	1	0
5	9	1	2	1

Max pooling

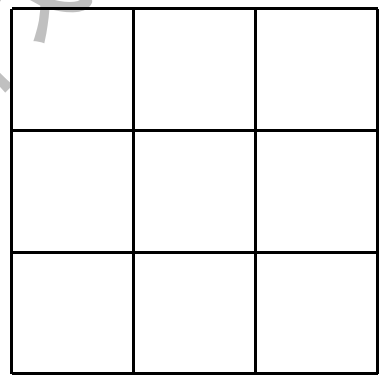
=

9	9	5
9	9	5
9	9	5

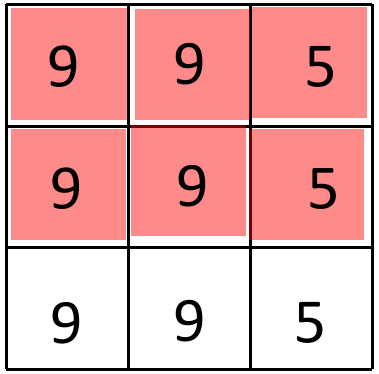


1.变量：向下平移了1行

Max pooling



=



3.结果：并没有改变

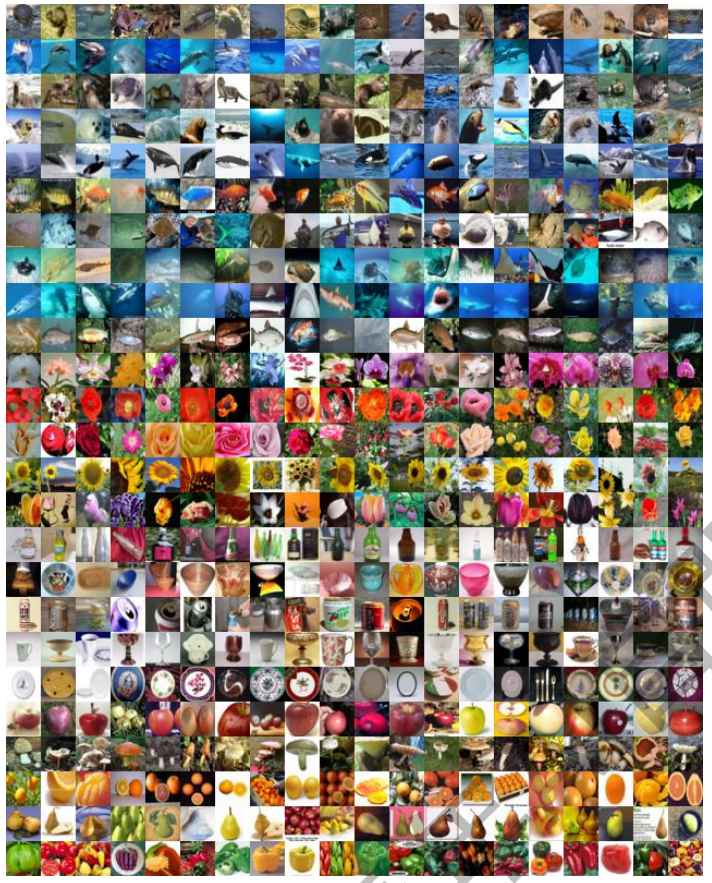
2.3：池化（pooling）略解

池化的作用2：平移不变性---结论

这跟人类处理图像的原理再一次不谋而合了，
当我们关注的目标物体在我们的视野中发生位移的时候，我们不会因此懵逼。
例如你看着一只羊在撒欢，他在你的视野中只是从右边跑到左边，你肯定不会因此而怀疑他是不是原来的那只羊
背后的原因是不管他在你的视野中平移与否，你对他的特征提取是完全一致的
maxpooling在一定程度上实现了这一点

3.1：数据集简介

本次作业数据集介绍



为了能让大家不浪费太多的时间在等待训练上，我们这次的作业还是选择一个入门级的数据集。

The CIFAR-10 dataset

共60000幅图像　图像尺寸：32 x 32 RGB
10类，每类6000幅图像。
训练集：50000幅图像
测试集：10000幅图像

3.2：Conv & pooling详解

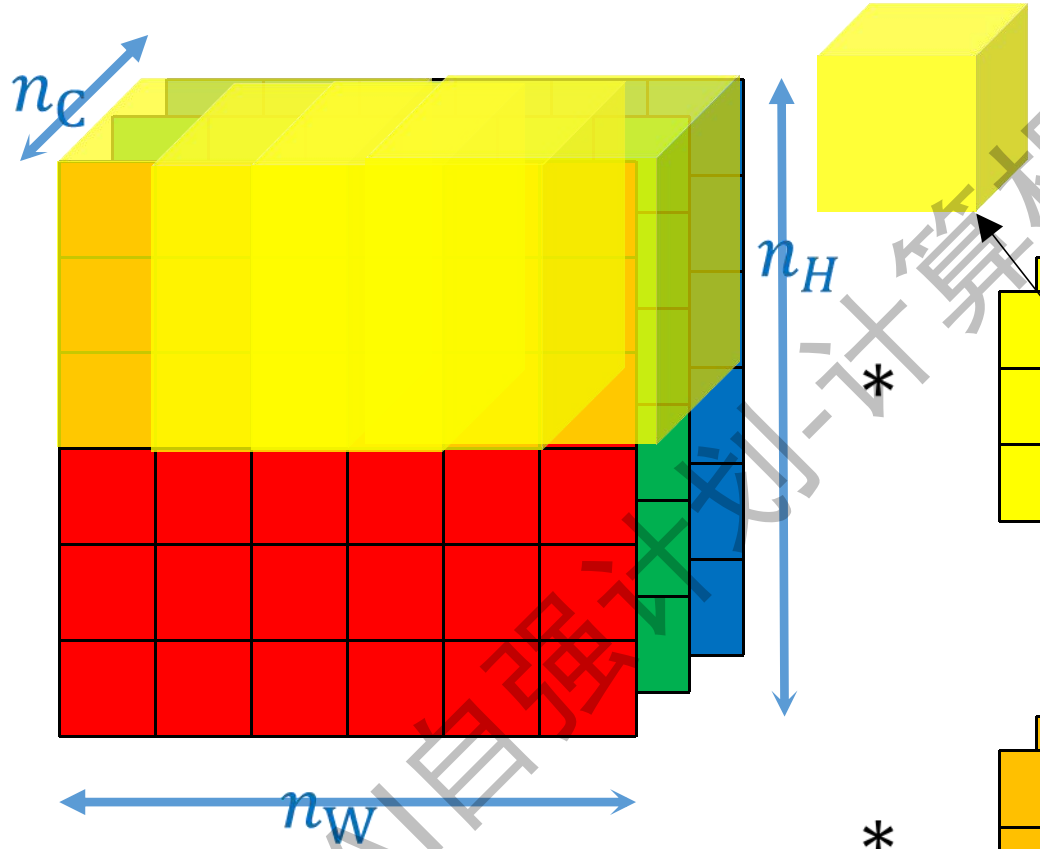
还有一些小问题

1. 输入图像是三维的（RGB），我们只学过平面卷积呀
2. 输入图像和卷积核的尺寸怎么标记呢？
3. 每扫一次必须只走一格吗？走两格行不行？
4. 你会发现这个“扫一遍”的过程对于角落里的元素不太公平，角落里的元素只被扫到了一次，但在画幅中心处的元素却被扫到了好几次。
5. 刚刚说到CNN会有好多隐藏层，那么不同隐藏层之间的卷积核怎么区分标记？

3.2：Conv & pooling详解

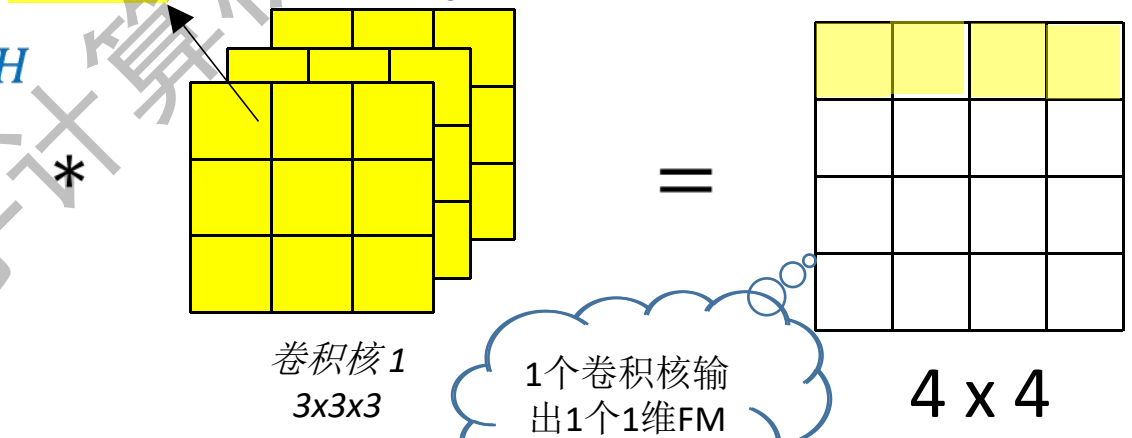
多维卷积&尺寸标记

图像是彩色的
有了三个颜色通道



$n_H \times n_W \times n_C = 6 \times 6 \times 3$

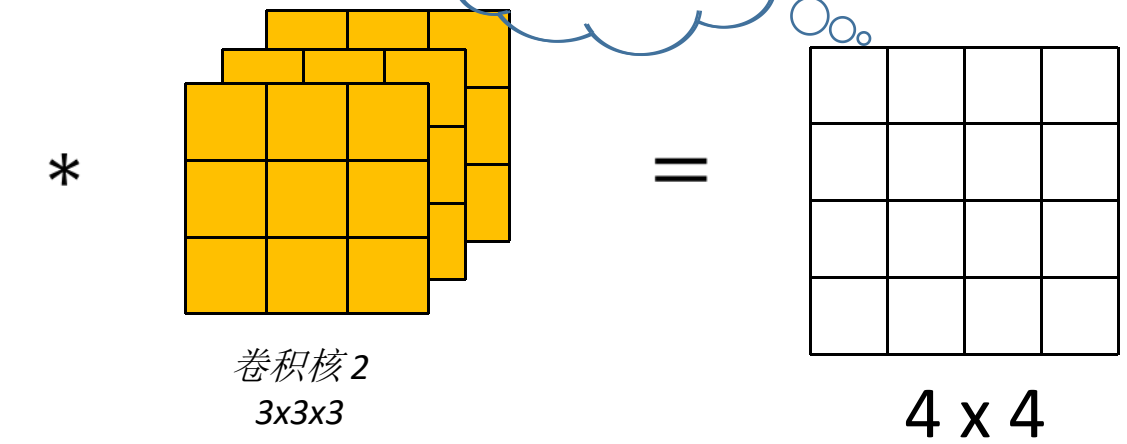
卷积核只有一个尺寸f
此处f=3，多维卷积中卷积核的
通道数要与输入图像/FM匹配



卷积核 1
3x3x3

1个卷积核输
出1个1维FM

4 x 4

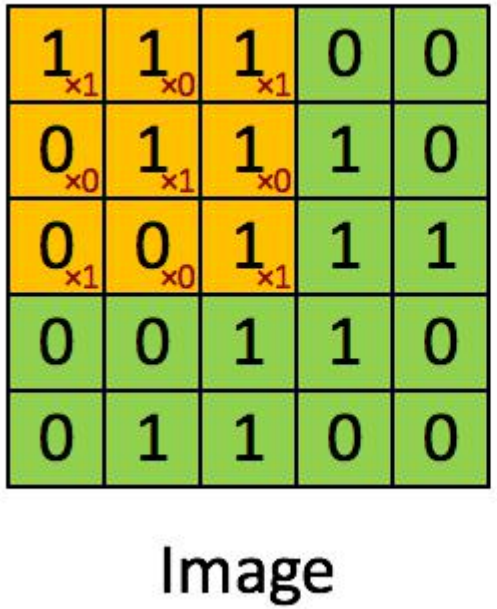


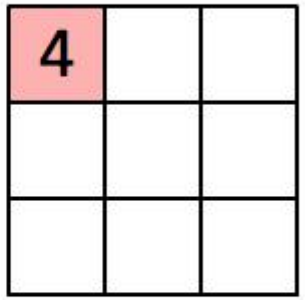
卷积核 2
3x3x3

4 x 4

3.2：Conv & pooling详解

stride





每扫一次走过的像素数量用s表示
S越大，运算后得到的FM越小，假设卷积核尺寸为f*f，输入图像尺寸为n*n
则得到FM的尺寸为：

$$\left[\frac{n-f}{s}+1\right]*\left[\frac{n-f}{s}+1\right]$$

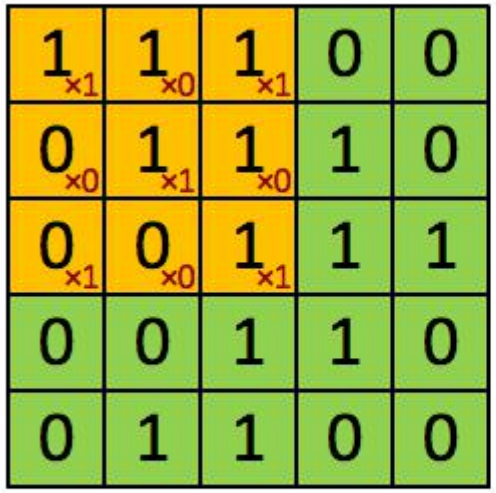
[ ]为向下取整符号

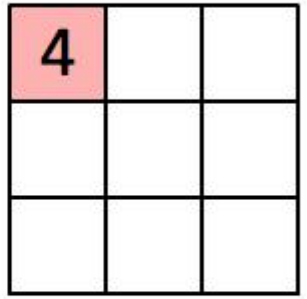
在做卷积/池化的时候，一次只能走一步吗？

3.2：Conv & pooling详解

padding

角落里的元素只被扫到一次不太公平？
越靠近边界，被扫到（特征表示）的几率越小





0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

在SAME padding且s=1时，$p=\frac{f-1}{2}$，所以f必为奇数

填补的主要策略有2种：
SAME：保持FM不缩小
VALID ： p=0。

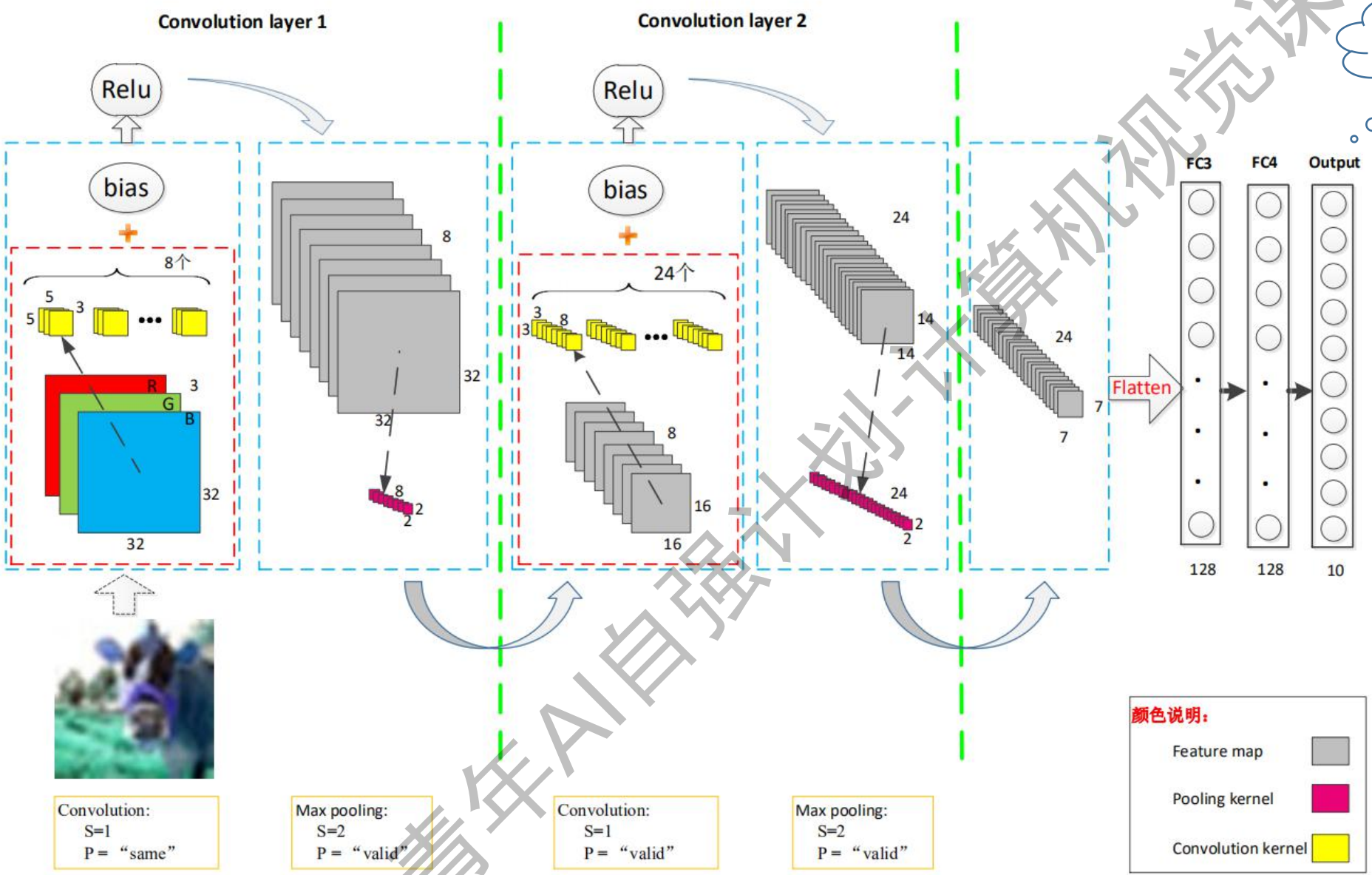
加入padding后，输出FM尺寸变为

$$\left[\frac{n+2p-f}{s}+1\right] * \left[\frac{n+2p-f}{s}+1\right]$$

在边界处填补（padding）一些像素块
边界向外拓展的像素个数用p表示，此处p=1

3.3：CNN前馈传播结构

良心示意图
重现江湖

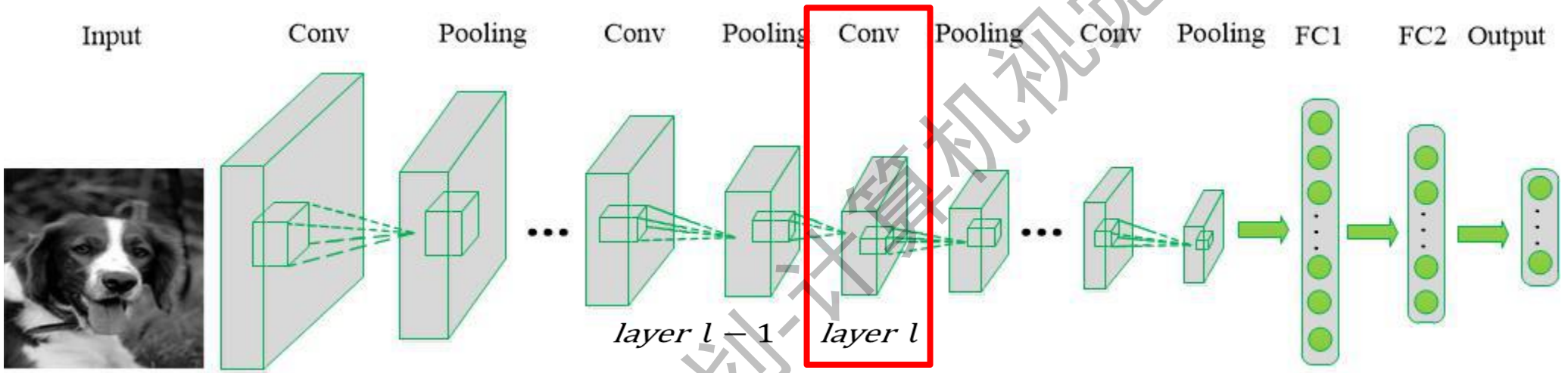


运算过程

1、输入图像
2、卷积计算
3、加入偏置
4、激活函数
5、池化计算
6、迭代直到FC
7、展开成1维向量
8、通过所有FC
9、通过softmax分类

3.3：CNN前馈传播结构

Notation汇总-以第l层卷积为例



Input FM size（*for layer* l）：$\mathrm{n}_H^{[l-1]} \times \mathrm{n}_W^{[l-1]} \times \mathrm{n}_C^{[l-1]}$

$f^{[l]}$ = 第l层卷积核尺寸值

$\mathrm{p}^{[l]}$ = 第l层的padding取值

$n_C^{[l-1]}$ = 第l层输入FM的通道数

同时也是第l层卷积核的通道数

第l层卷积核*权值总数*：$f^{[l]} \times f^{[l]} \times \mathrm{n}_C^{[l-1]} \times \mathrm{n}_C^{[l]}$

Output FM size（*for layer* l）：$\mathrm{n}_H^{[l]} \times \mathrm{n}_W^{[l]} \times \mathrm{n}_C^{[l]}$

Bias size $b^{[l]} = 1 \times 1 \times \mathrm{n}_C^{[l]}$

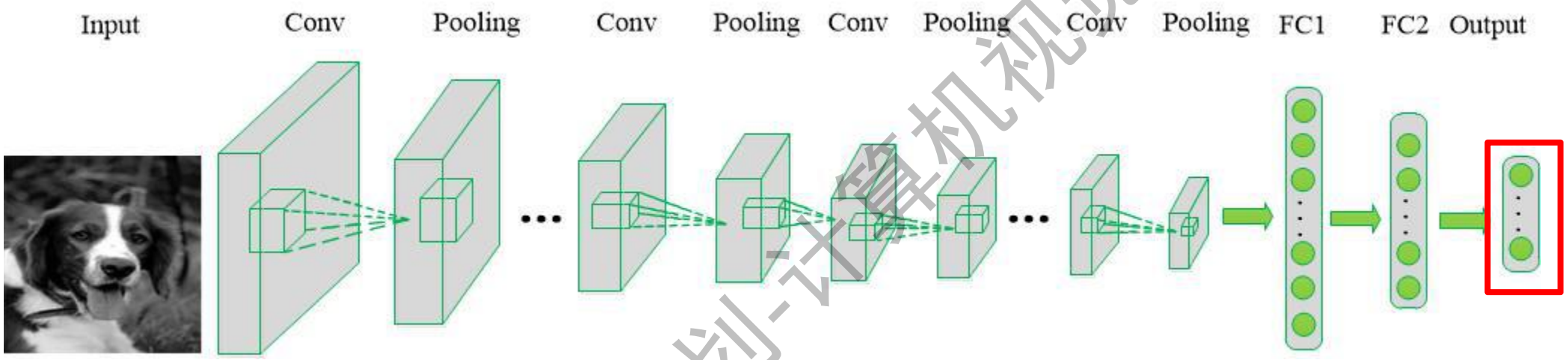
$\mathrm{a}^{[l]}$ *size* $= \mathrm{n}_H^{[l]} \times \mathrm{n}_W^{[l]} \times \mathrm{n}_C^{[l]}$

$z^{[l]}$经过激活函数后变为 $\mathrm{a}^{[l]}$

$$\mathrm{n}_H^{[l]} = \left[\frac{\mathrm{n}_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1\right]$$

3.4：CNN反向传播及softmax

多分类任务，如何做预测？



	sigmoid	softmax
函数	$f(z)=\frac{1}{1+\exp(-z)}$.	$S_i=\frac{e^{z_i}}{\sum_{j=1}^{N}e^{z_j}}$ 其中，z_i表示第 i 个输入值，S_i表示第 i 个输出值，N 为类别数
描述	sigmoid 将一个值**映射**到（0,1）的区间 这样可以用来做**二分类**。	softmax 把一个 k 维向量（a1,a2,a3,a4…）**映射**成另一个 K 维向量（b1,b2,b3,b4…），其中 bi 是一个 0~1 的常数，并且 b 向量的和为 1。然后可以根据 bi 的大小来**进行多分类**的任务。

3.4：CNN反向传播及softmax

取自真实例子
输入可以是任何数

所有输出的和为1
多分类与sigmiod相比的优势

Softmax输入	[-1.467, -0.309, 0.423, -0.884, 0.131, 0.942, 0.806, 1.097, -2.917, 0.589]
Softmax funtion	$S_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$ 其中，z_i表示第i个输入值，S_i表示第i个输出值，N为类别数，这里是10
Softmax输出	[0.017, 0.054, 0.111, 0.030, 0.083, 0.187, 0.163, 0.219, 0.004, 0.132]
Label	[0 1 0 0 0 0 0 0 0 0]
Loss function	$\text{loss} = -\sum_{i=1}^{N} label_i * \log(S_i)$ 其中，$label_i$表示标签中的第i个数，S_i表示softmax输出的第i个数 N表示类别数量，这里为10
loss	[2.918]

带入Label会发现
只剩一项了

要求大家定性掌握即可

BP比较复杂
不要求掌握

3.5：训练结果

基本条件设置	对比结果	
	DNN	CNN
数据集：CIFAR-10 数据增强：左右翻转 Batch-size：256 Lr：0.1（learning rate decay） Loss function：交叉熵 优化方法：Adam 训练次数：5000 模型层数：4 Batchnorm：使用 激活函数：relu	权值数量：11,550,730 检测精度：56.25%	权值数量：170,818 检测精度：70.18%

精度不是太高？

标准DNN的局限性 → 什么是卷积 → 卷积的优势 → 什么是池化 → 多维卷积与池化 → 实例练兵

3.6：尾声&作业说明

第一节课时的两个小实验

1、人类是通过对样本的学习来获取知识的
2、人类可以自动提取特征并进行分类
3、人类可以通过调整注意力来调整视角

CNN已经具备的

哪个是虎皮猫，哪个是小老虎呢?

1、人类可以通过小样本学习知识
2、人类学习之后的泛化能力极强
3、人类具备联想、抽象、举一反三的能力

CNN尚且缺少的

8.2：尾声&作业说明

作业说明：
本次只有1个作业
用简单的CNN拟合CIFAR100数据集
帮助大家彻底搞懂CNN的前馈传播结构

扫码加好友进群

关注直播间公告