

# 深度学习

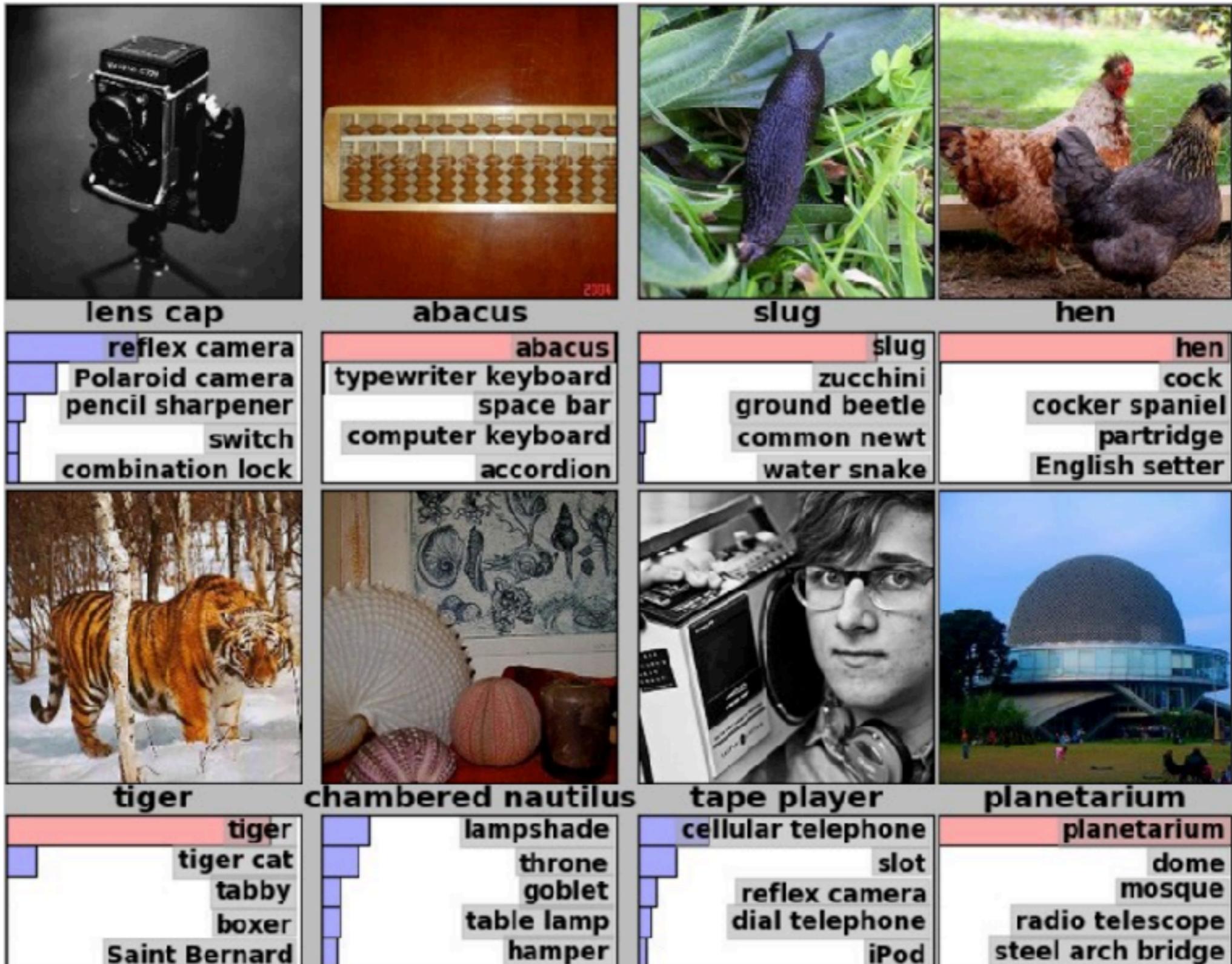
## 卷积神经网络 (1)

# 概览

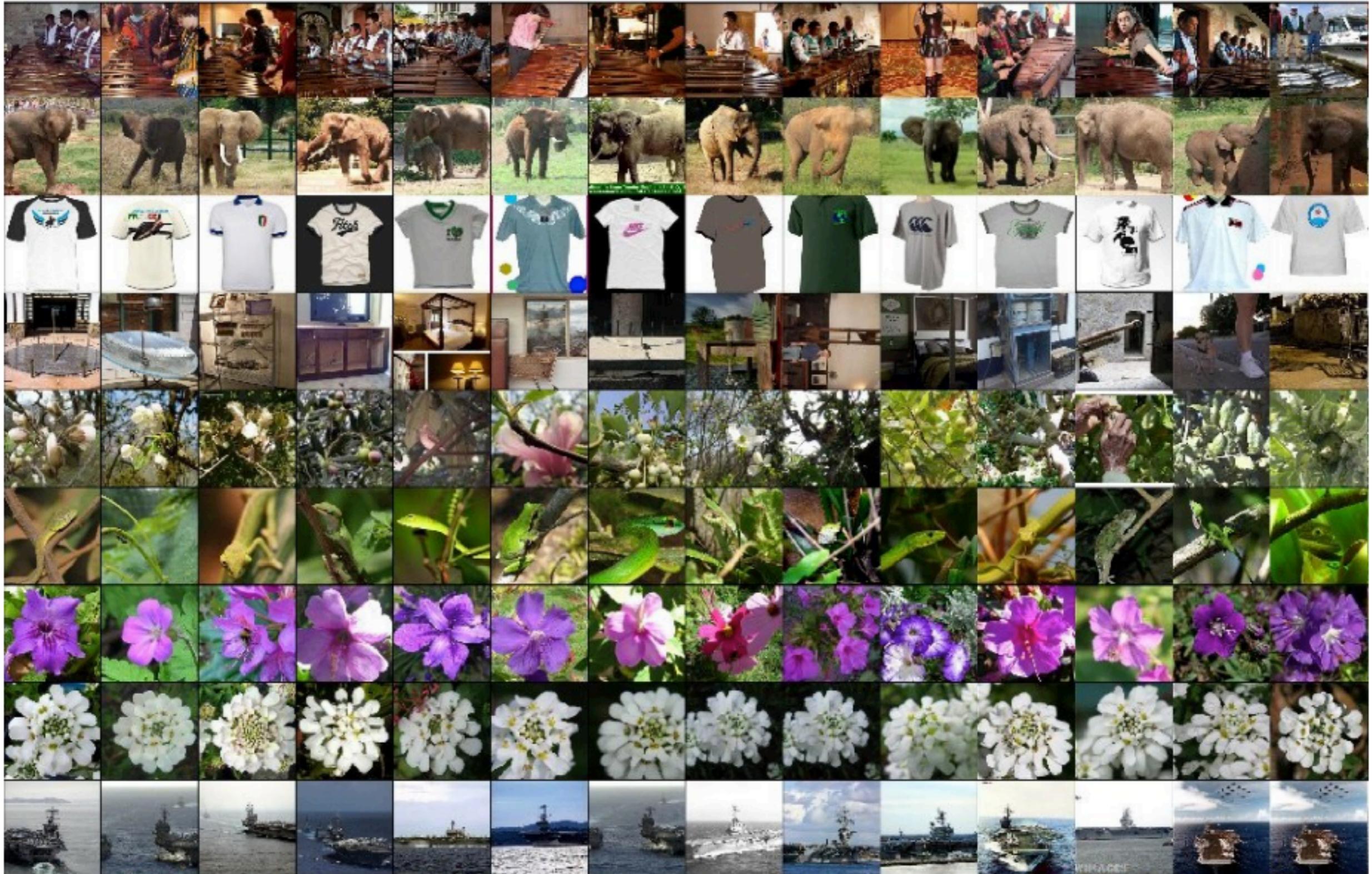
1. 无处不在的卷积神经网络。
2. 视觉是什么？
3. 直接使用普通ANN实现图片分类。
4. 卷积神经网络。
  - 4.1.认识感受器与感受野。
  - 4.2.使用“edge”特征描述图片。
  - 4.3.卷积。
  - 4.4.池化。
5. 卷积与池化的黄金组合。

# 1. 无处不在的卷积神经网络

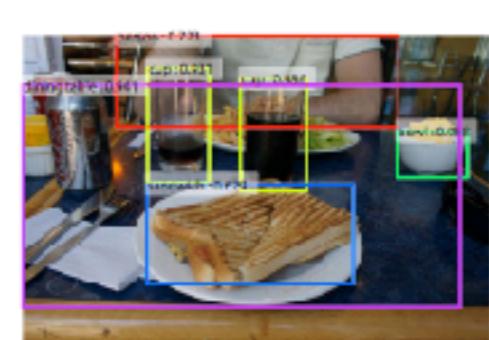
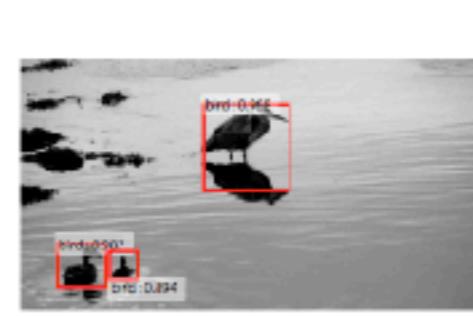
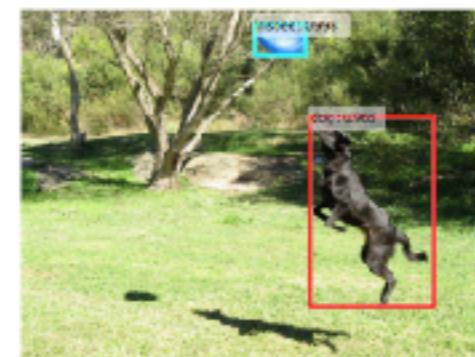
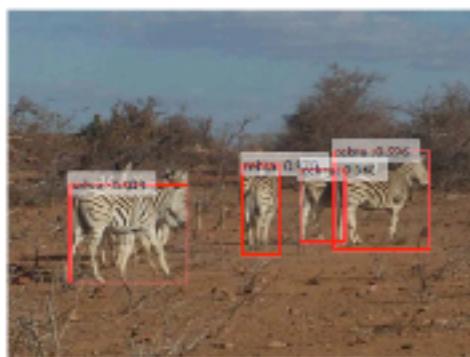
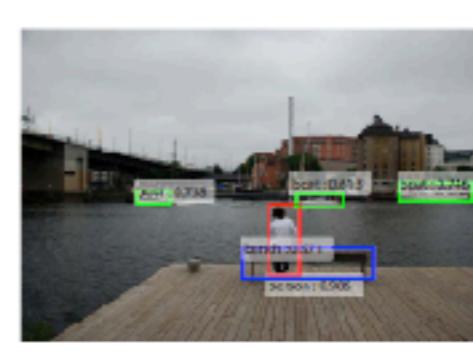
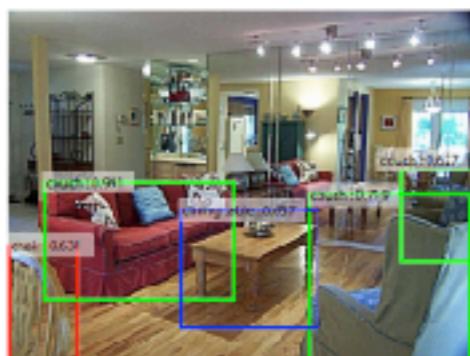
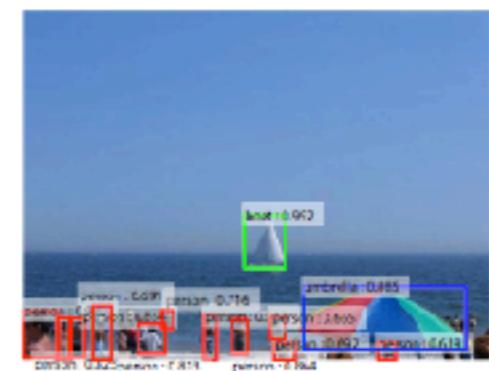
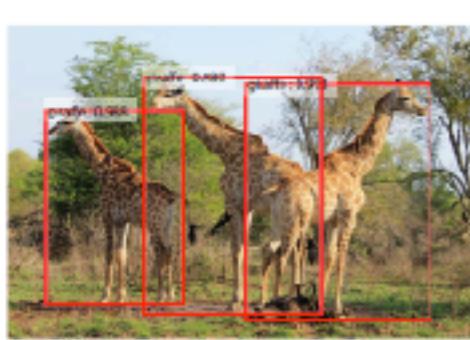
# 分类



# 检索



# 目标侦测



# 图片分割



来源：Learning Hierarchical Features for Scene Labeling

Author:WangQi  
Email:wangqikaixin@gmail.com

# 图片内容描述



a man holding a tennis racquet on a tennis court



a close up of a plate of food on a table



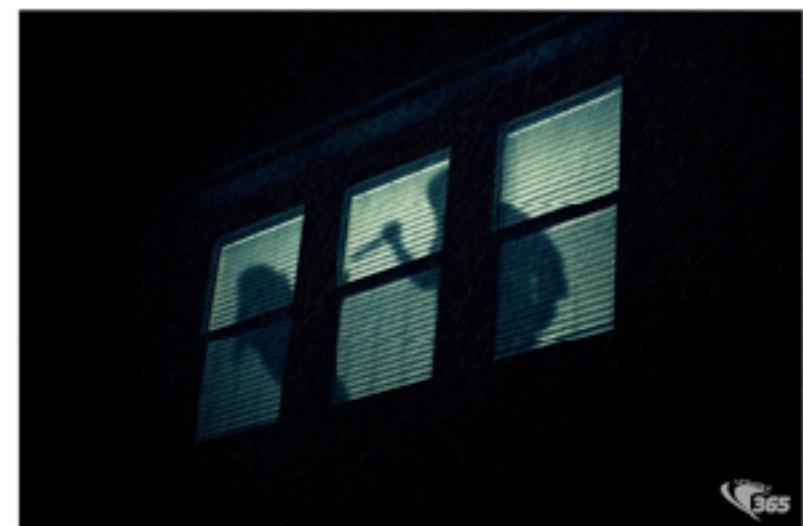
a desk with a laptop and a monitor



a group of people playing a game with nintendo wii controllers

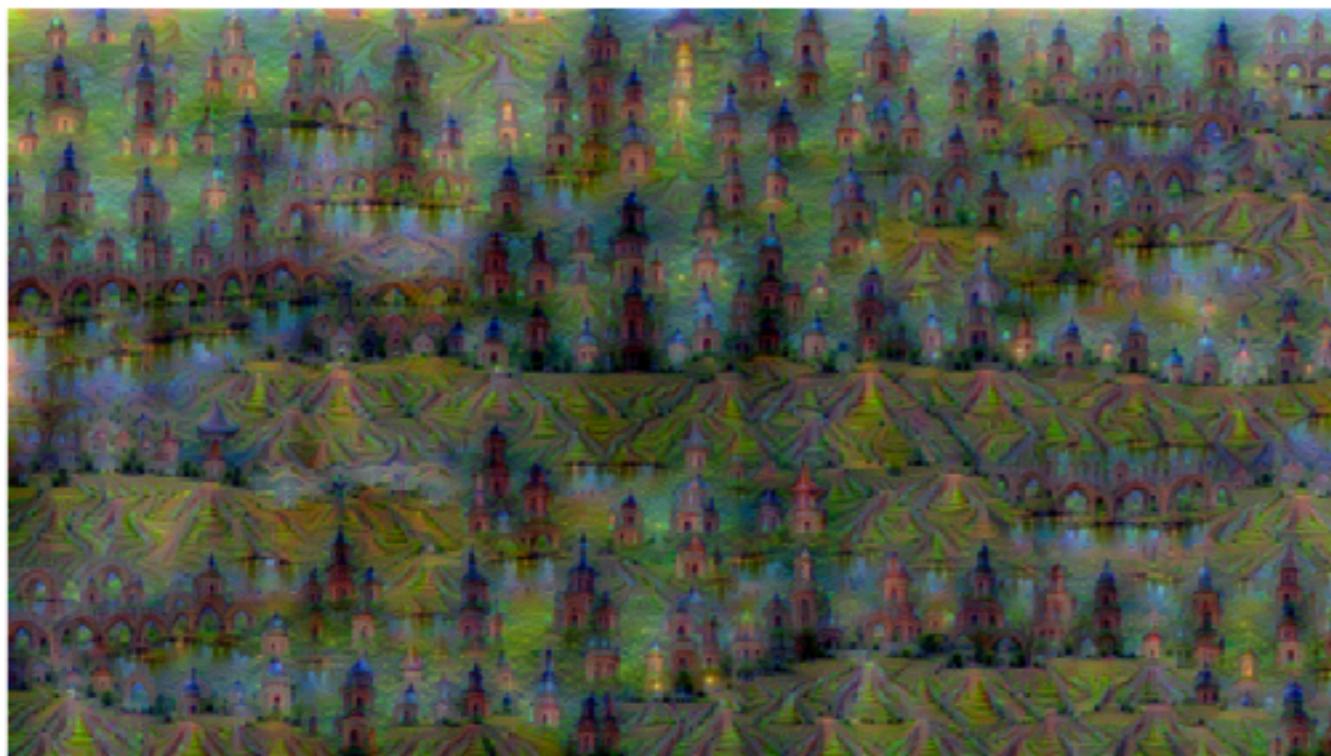
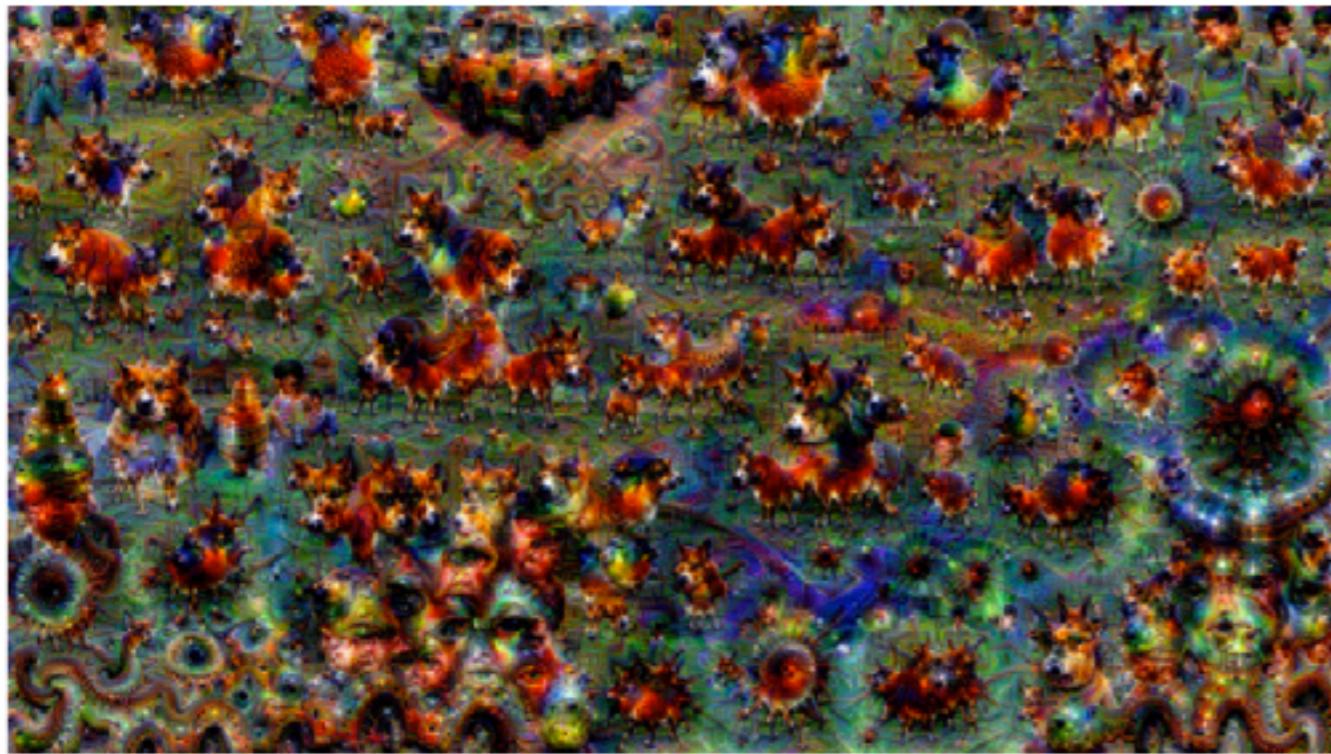


a pile of luggage sitting on the ground



a view of a building with a clock on the top of it

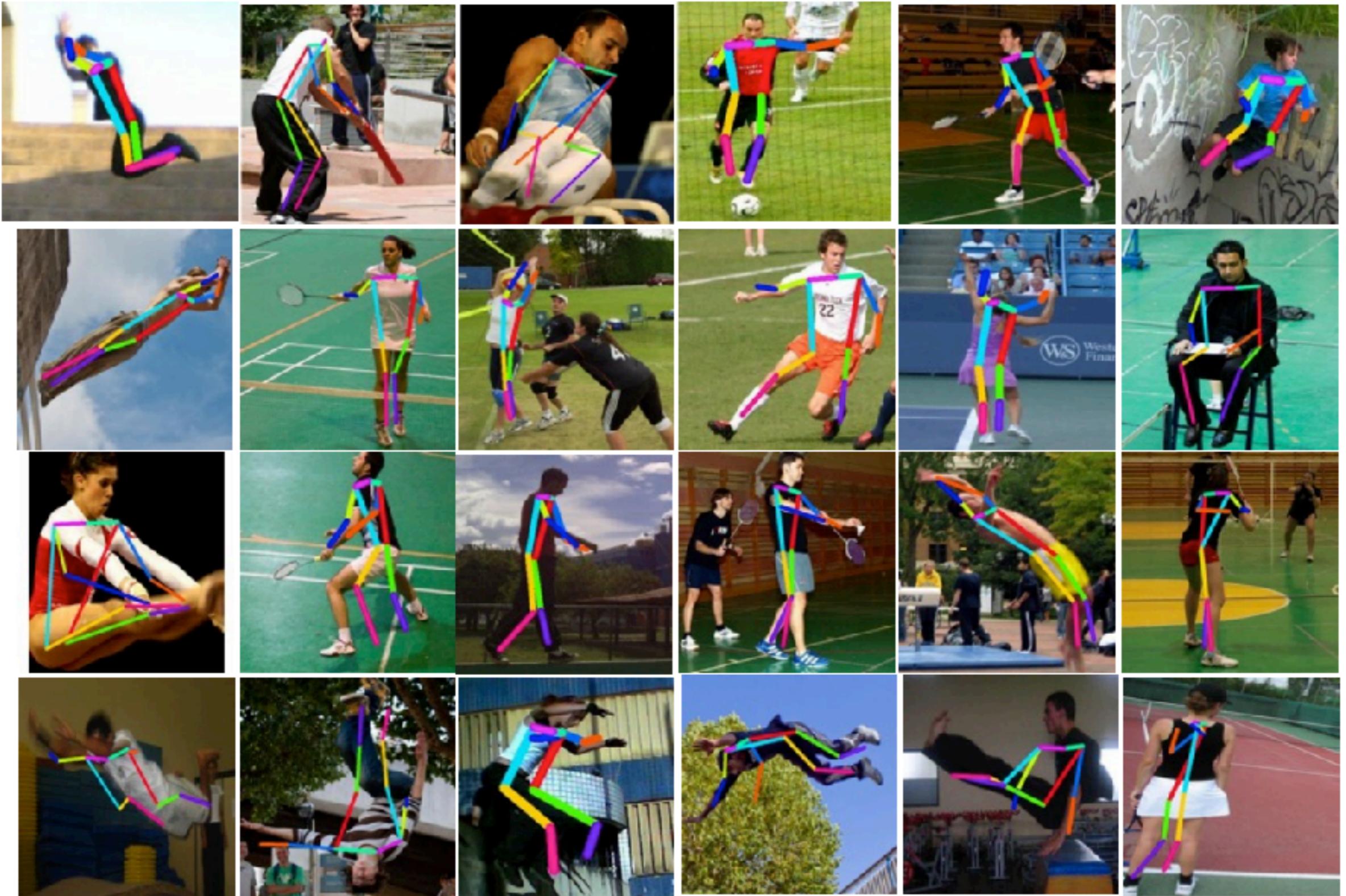
# 图片风格化



来源：[cs231n\\_2017\\_lecture5.pdf](https://cs231n.github.io/neural-networks-for-computer-vision/#style-transfer)

Author:WangQi  
Email:wangqikaixin@gmail.com

# 人体姿态估计



## 2. 视觉是什么？

# 谁看到了真实的世界？

HUMAN VISION



BIRD VISION

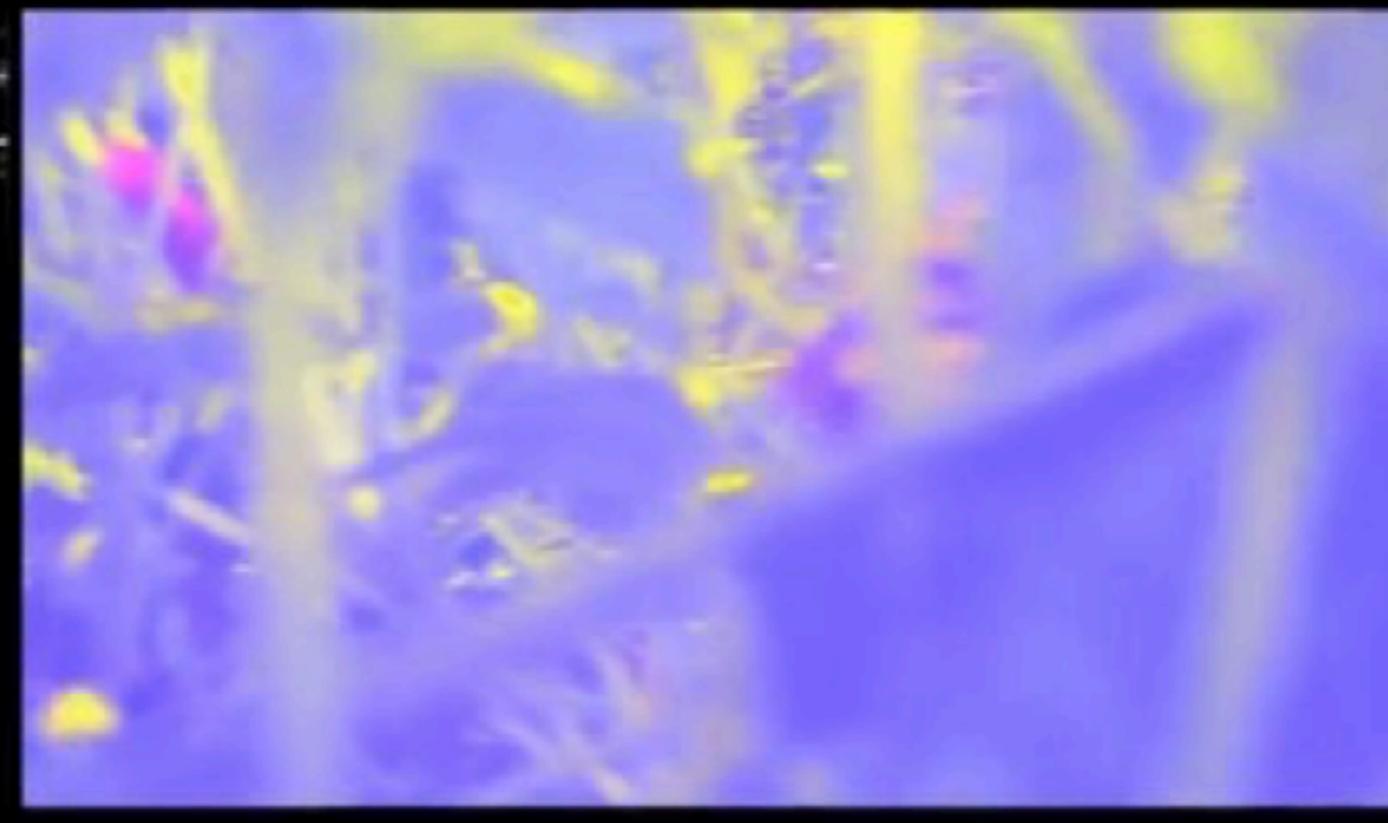


# 谁看到了真实的世界？

HUMAN VISION



SNAKE VISION

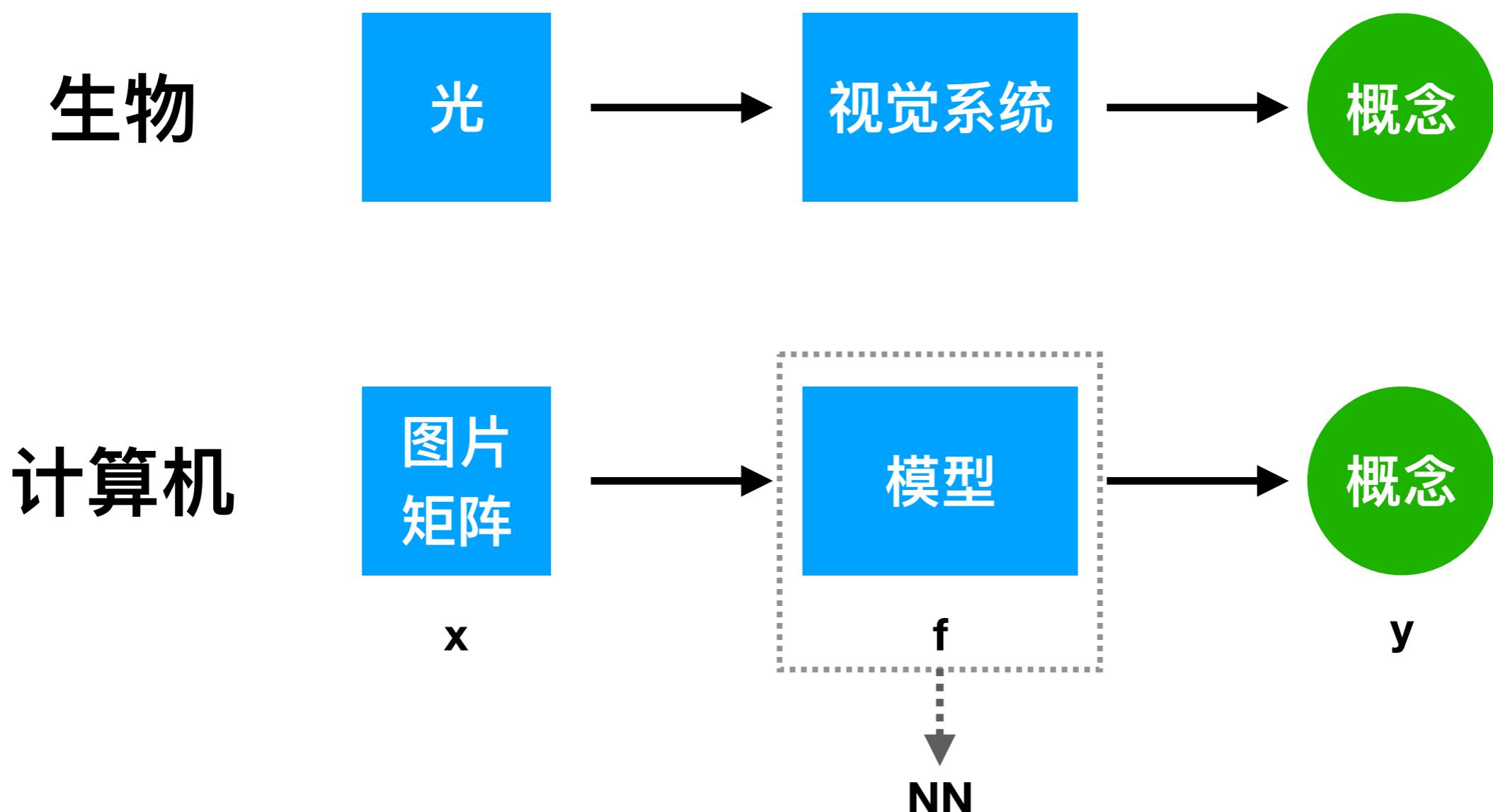


# 视觉的本质

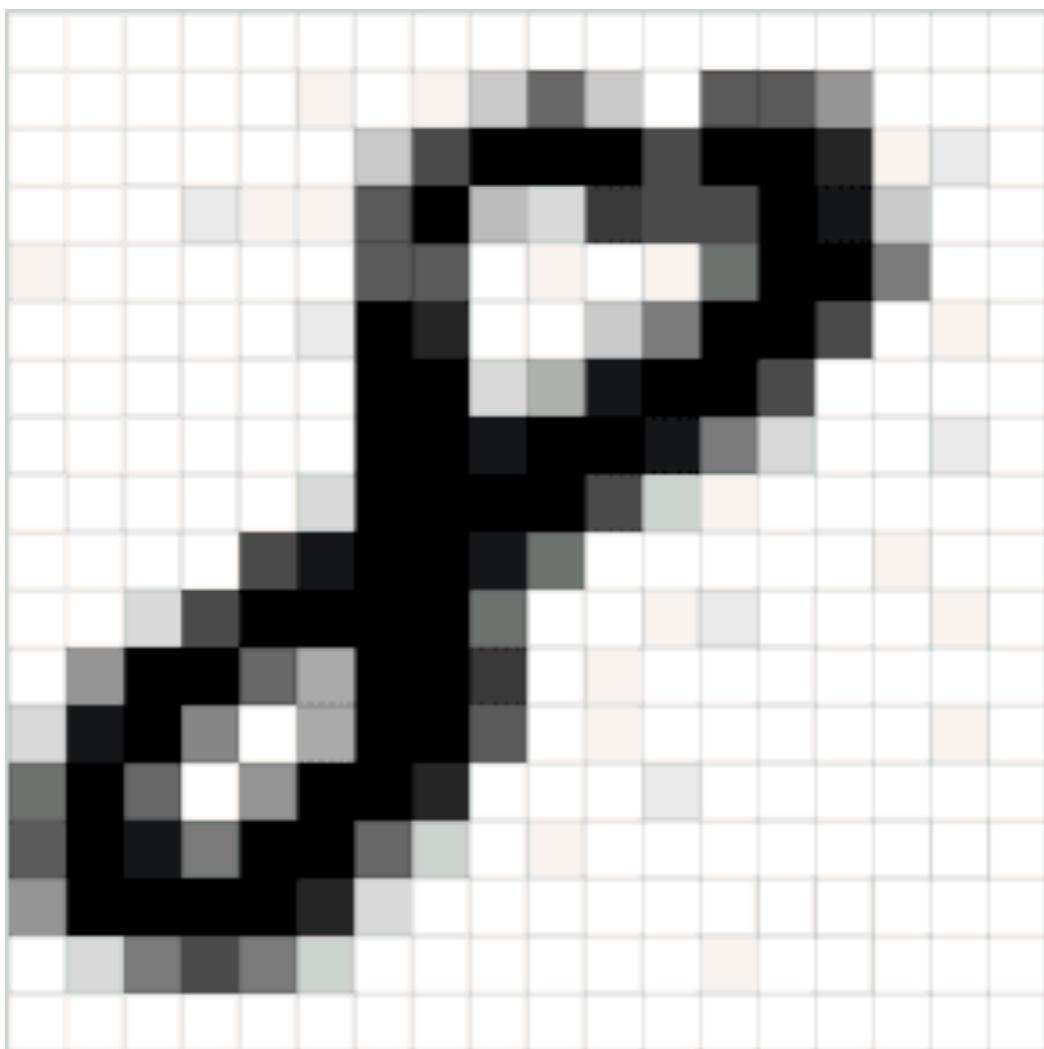


视觉本质上是对外界刺激抽象出一个概念的过程。  
我们所看到的世界是主观的世界，不是世界的本来面目。

# 计算机视觉



# 视觉与机器视觉



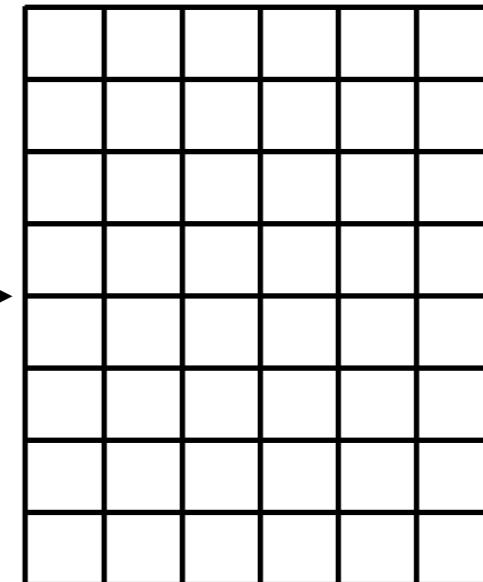
# 人的视觉

计算机

# 图片通道



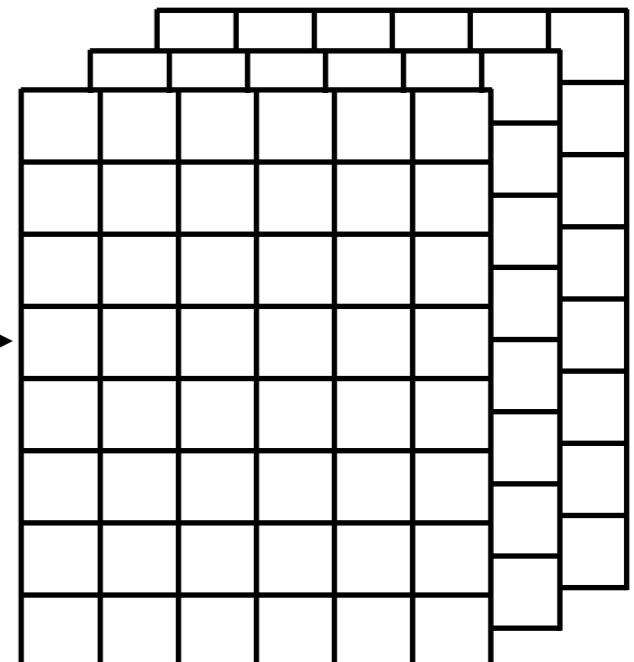
灰度图像



图片矩阵

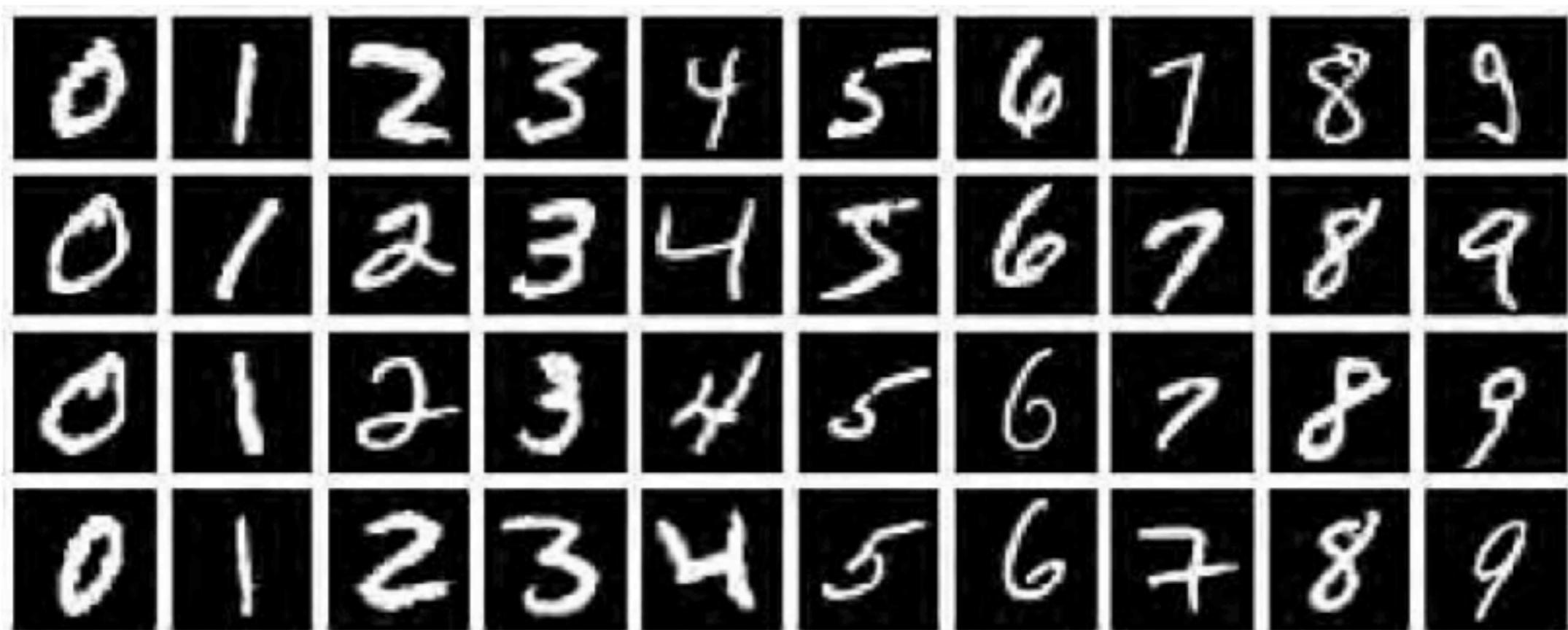


RGB图像



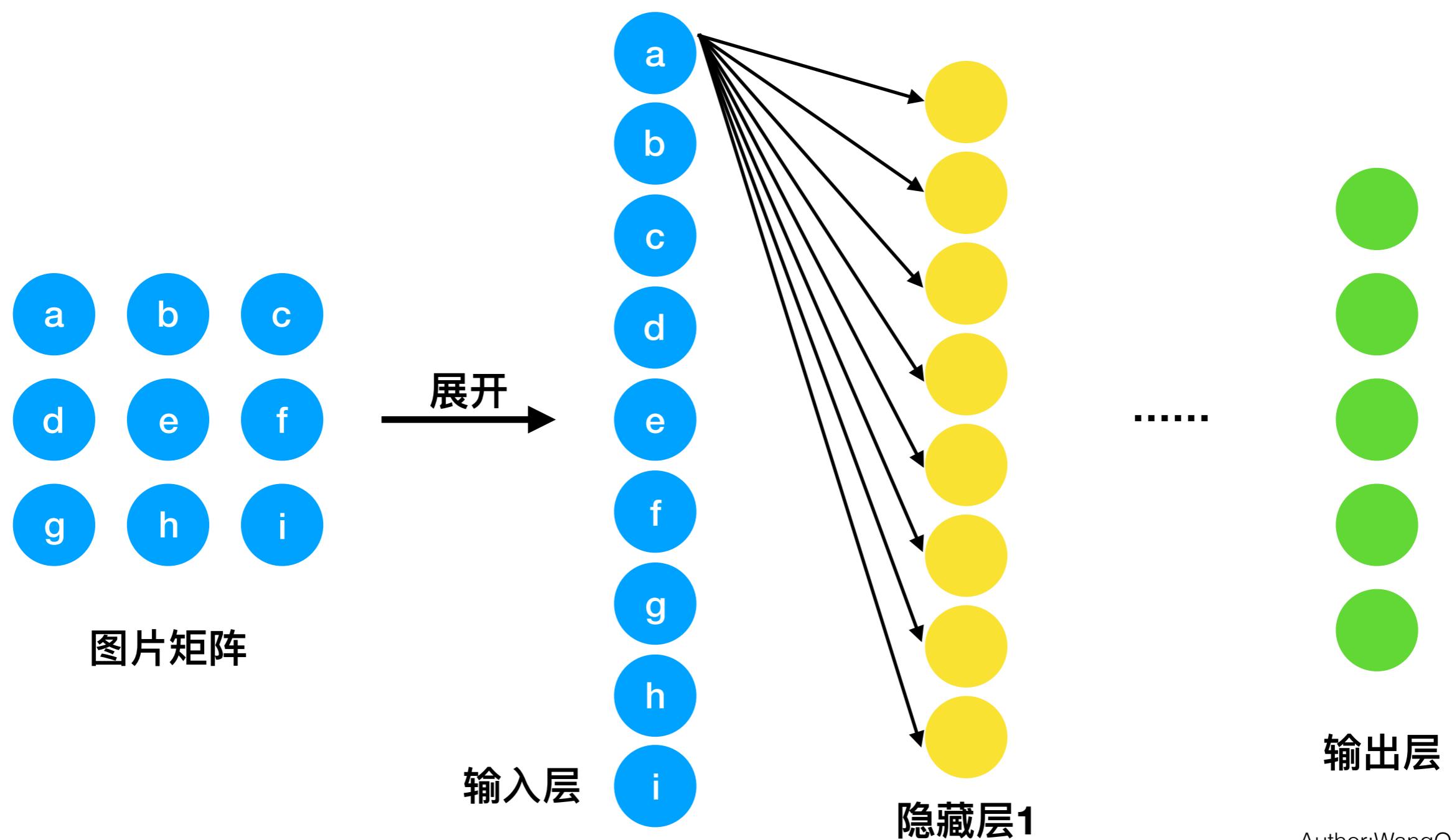
### 3. 使用ANN实现图片分类

# 使用普通ANN实现图片分类

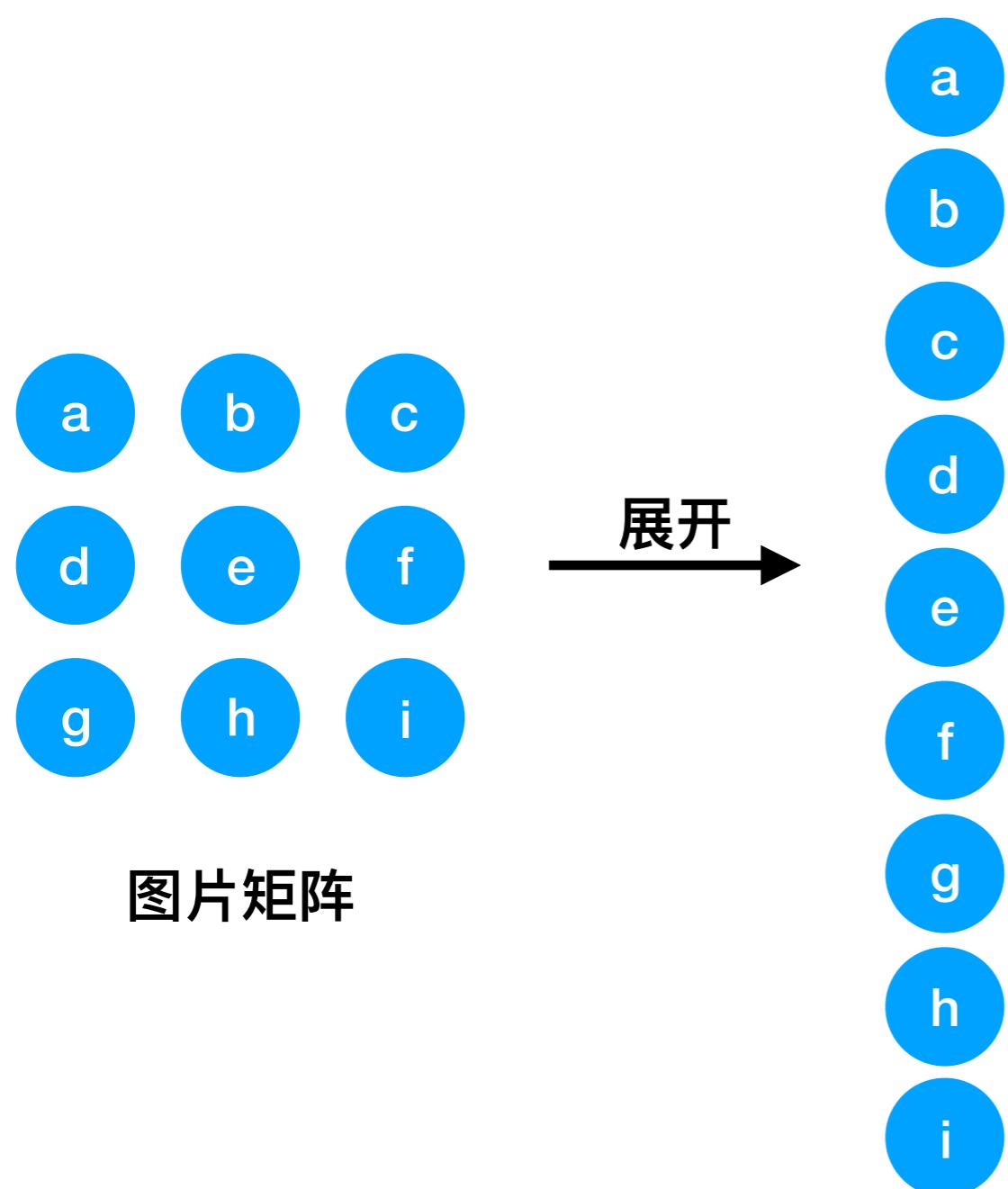


思考：如何将图片输入给ANN，并进行分类？

# 使用普通ANN实现图片分类



# 使用普通ANN实现图片分类



可以看到：

1. 二维矩阵展开丢失二维信息。
2. 输入层与隐藏层全连接带来了维度灾难问题。

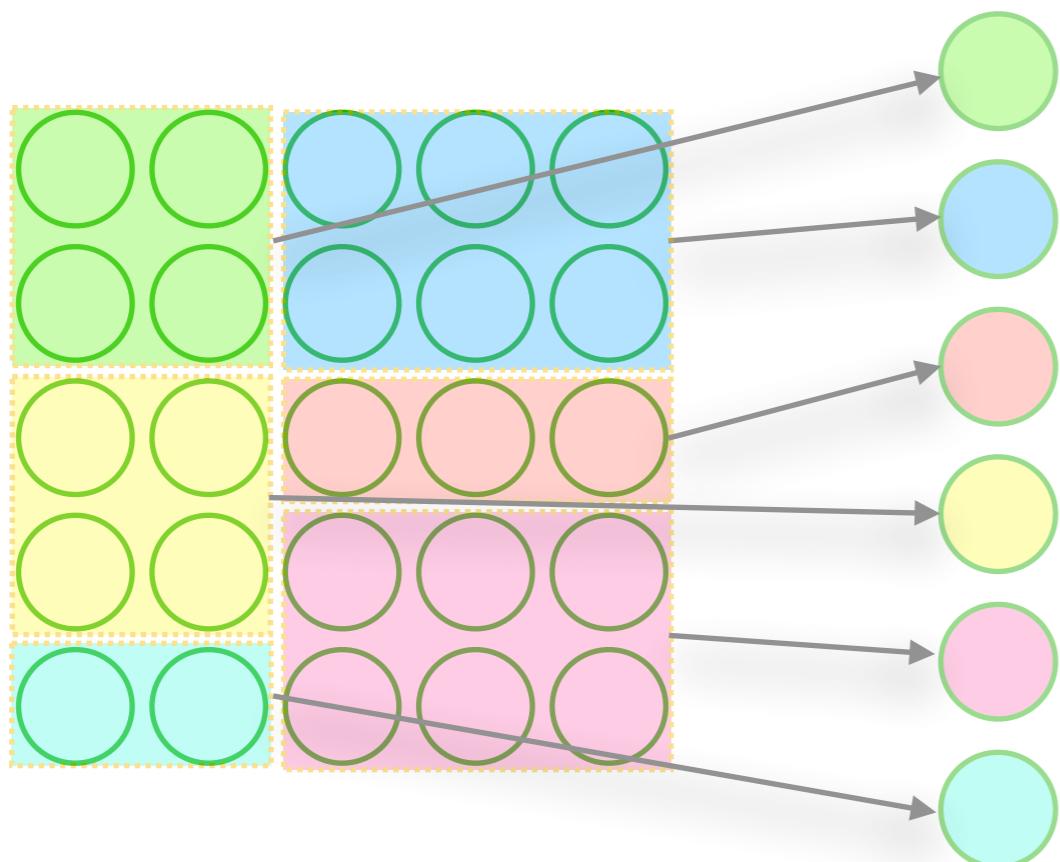
# 4. 卷积神经网络

# 卷积神经网络

**卷积神经网络(CConvolutional Neural Network, CNN)**是一种经典的前馈神经网络，主要受生物学中的感受野(receptive field)的概念提出。感受野在生物体中广泛存在，一个感受野连接多个感受器细胞，这些感受器细胞共同决定了感受野是否兴奋。通过感受野的机制，生物体传入的信号数量会大大降低，同时也能很好的对输入信号进行特征提取。卷积神经网络有两大工具来降低参数数目：卷积与池化。

# 4.1 认识感受器与感受野

# 感受器与感受野



5\*5图片

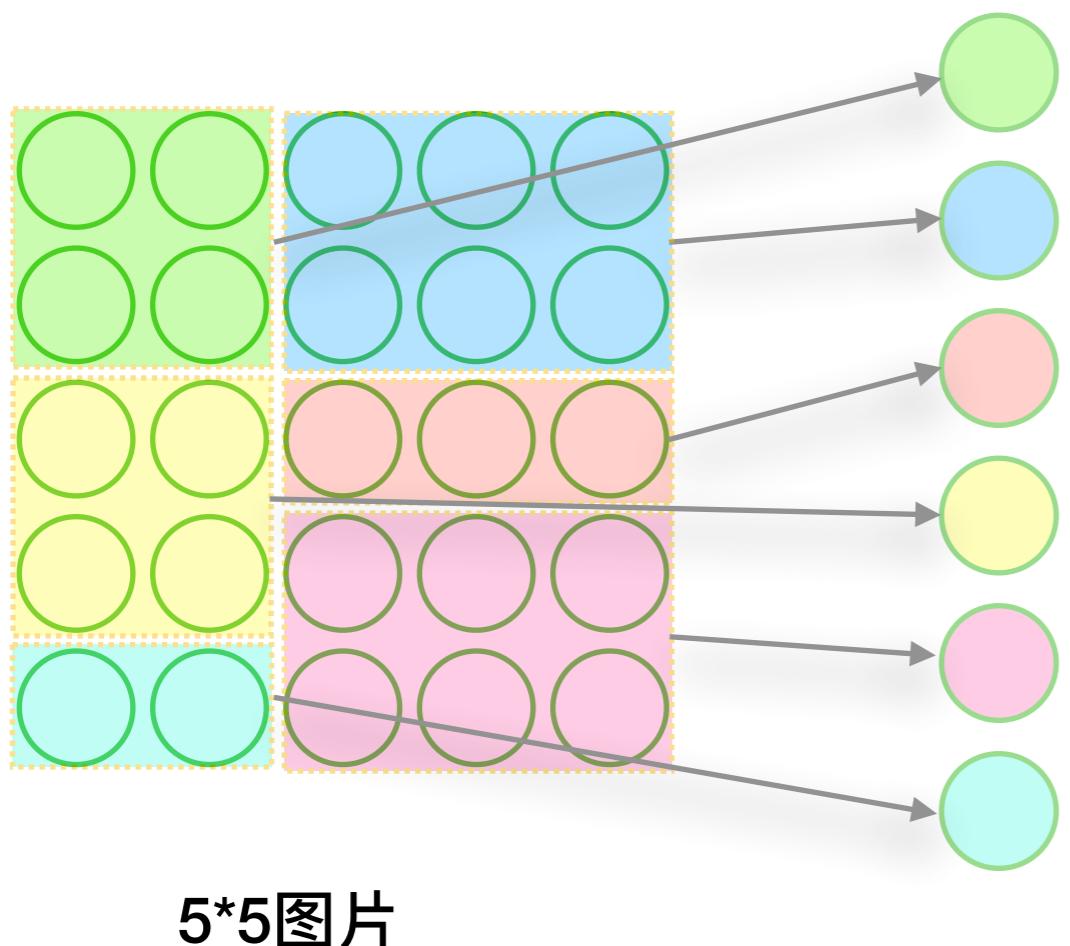
局部感知：图像的局部联系紧密，局部像素可以作为整体的一个特征。

局部连接：具有紧密联系的局部区域与一个神经元相连。没有紧密联系的部分无须相连。



利用了2维信息，降低了连接数量。

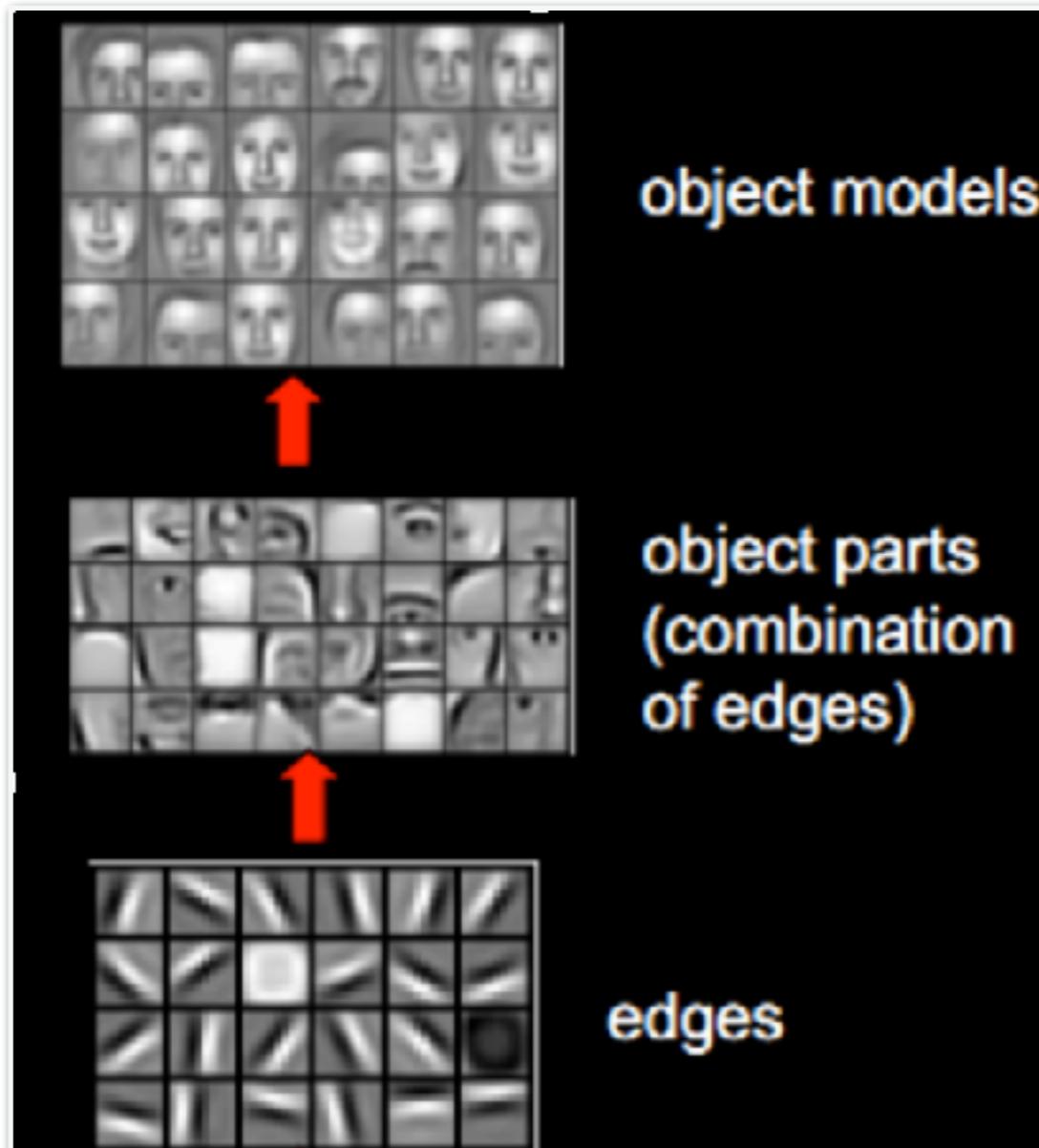
# 感受器与感受野



感受野感知到的信息是什么？

Hubel和Wiesel在1962年通过实验发现：大脑中的一些特别的神经细胞只会对特定方向的边缘“edge”做出反应。

# 感受器与感受野



“边”的特点：

1. 每种“边”都是不同的。即每种“边”可以看作一种特征。
2. 有限数量的边进行组合可以构成无限多的图片。

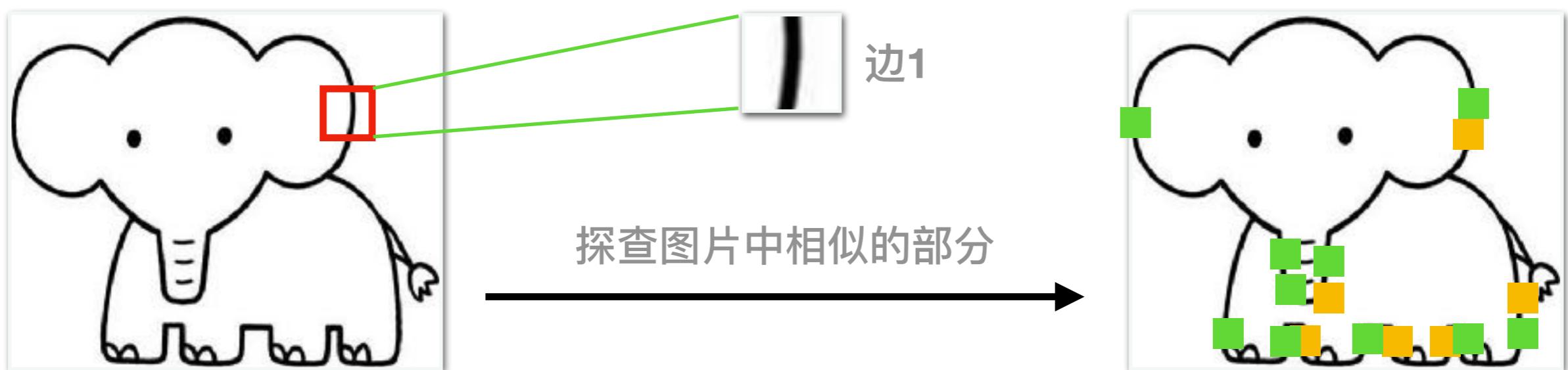
规律：所有图片均可以通过一定数量的“边”组合得到。

使用“边”构成各种人脸

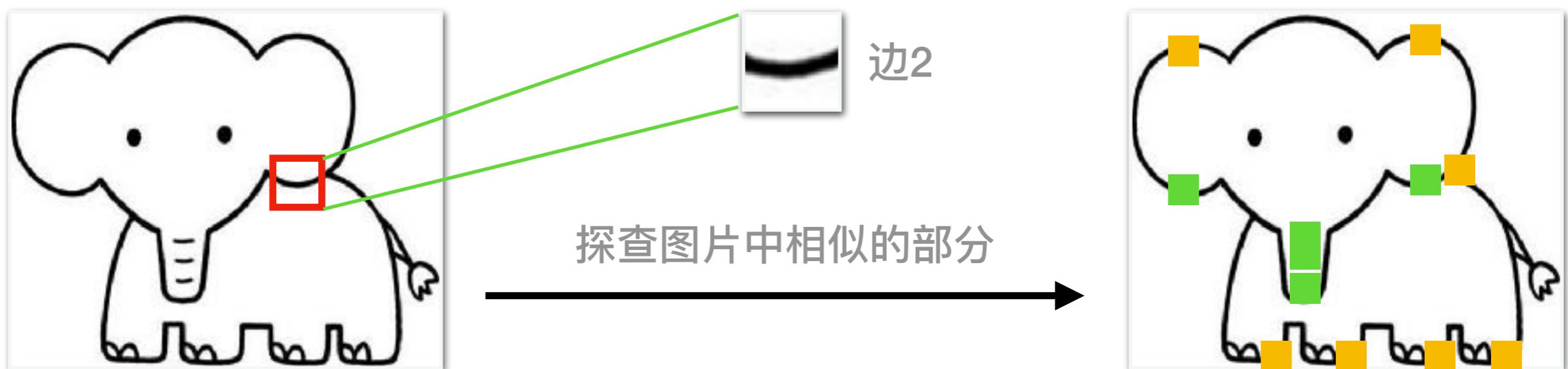
## 4.2 使用“edge”特征描述图片

**思考：假如构成图像的“边”已知，如何使用这些“边”描述现有图像？**

# 查找“边”在图像中的位置



# 查找“边”在图像中的位置



**思考：人通过观察可以找到与“边”相似的部分，计算机该如何查找呢？**

# “边”矩阵



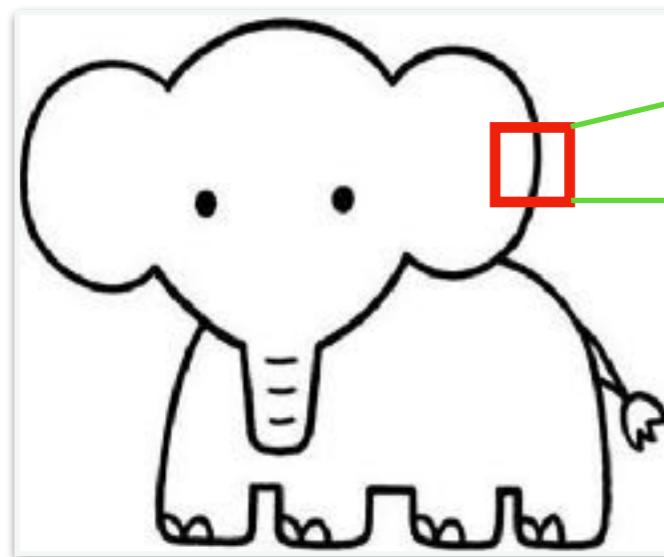
边1



0	0	100	0	0
0	0	100	0	0
0	0	100	0	0
0	0	90	0	0
0	20	70	0	0

边1对应的矩阵

# 子图1矩阵



图片

子图1



0	0	90	0	0
0	0	110	0	0
0	0	100	0	0
0	0	90	0	0
0	40	80	0	0

子图1对应的矩阵

# 求值

0	0	100	0	0
0	0	100	0	0
0	0	100	0	0
0	0	90	0	0
0	20	70	0	0

边1对应的矩阵

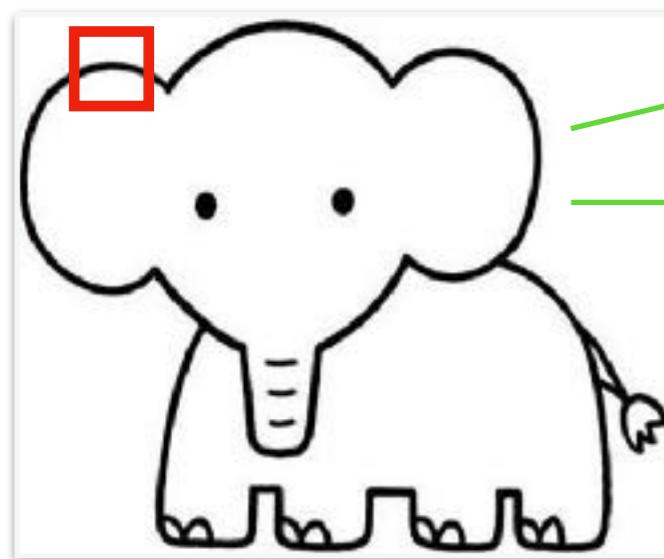
\*

0	0	90	0	0
0	0	110	0	0
0	0	100	0	0
0	0	90	0	0
0	40	80	0	0

子图1对应的矩阵

$$y_1 = 100 * 90 + 100 * 110 + 100 * 100 + 90 * 90 + 20 * 40 + 70 * 80 = 44500$$

# 子图2矩阵



图片



子图2

0	0	0	0	0
0	0	30	0	0
80	80	100	80	70
20	0	0	0	10
0	0	0	0	0

子图2对应的矩阵

# 求值

0	0	100	0	0
0	0	100	0	0
0	0	100	0	0
0	0	90	0	0
0	20	70	0	0

边1对应的矩阵

\*

0	0	0	0	0
0	0	30	0	0
80	80	100	80	70
20	0	0	0	10
0	0	0	0	0

子图2对应的矩阵

$$y_2 = 100 * 30 + 100 * 100 = 13000$$

# 两次求值对比

0	0	100	0	0
0	0	100	0	0
0	0	100	0	0
0	0	90	0	0
0	20	70	0	0



边1对应的矩阵

0	0	90	0	0
0	0	110	0	0
0	0	100	0	0
0	0	90	0	0
0	40	80	0	0



子图1对应的矩阵

0	0	0	0	0
0	0	30	0	0
80	80	100	80	70
20	0	0	0	10
0	0	0	0	0



子图2对应的矩阵

$$y_1 = 100 * 90 + 100 * 110 + 100 * 100 + 90 * 90 + 20 * 40 + 70 * 80 = 44500$$

$$y_2 = 100 * 30 + 100 * 100 = 13000$$

# 两次求值对比

0	0	100	0	0
0	0	100	0	0
0	0	100	0	0
0	0	90	0	0
0	20	70	0	0



边1对应的矩阵

0	0	90	0	0
0	0	110	0	0
0	0	100	0	0
0	0	90	0	0
0	40	80	0	0



子图1对应的矩阵

0	0	0	0	0
0	0	30	0	0
80	80	100	80	70
20	0	0	0	10
0	0	0	0	0



子图2对应的矩阵

与“边”接近的“子图”，对应元素乘积的和较大，反之较小。

思考：只有与“边”相近的子图， 对应  
元素乘积和才会较大吗？

## 4.3 卷积

# 卷积

1 <small><math>\times 1</math></small>	1 <small><math>\times 0</math></small>	1 <small><math>\times 1</math></small>	0	0
0 <small><math>\times 0</math></small>	1 <small><math>\times 1</math></small>	1 <small><math>\times 0</math></small>	1	0
0 <small><math>\times 1</math></small>	0 <small><math>\times 0</math></small>	1 <small><math>\times 1</math></small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

将“边”与所有可能的子图运算，就是卷积的过程。

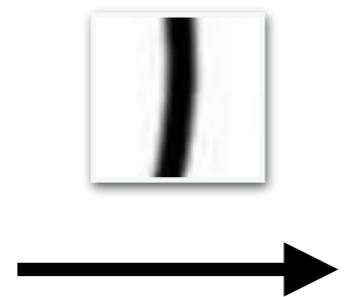
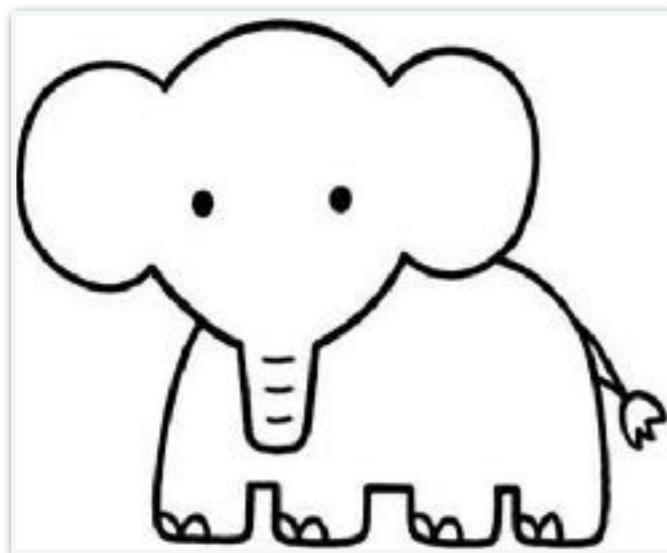
注意：上面，为了便于理解，我们使用了统计得到的卷积核，实际中我们通过训练获得卷积核。

卷积隐含的原则是：输入（此处是图像）的一部分的统计特性与其他部分是一样的。

我们把 $3 \times 3$ 的“边”矩阵叫做卷积核(Convolusion kernel)。右边的图是与卷积核运算的结果，我们把它叫做特征分布图，简称特征图(Feature map)。特征图反映了某个特征在某个输入（此处是图片）上的激活值。

思考：卷积一定是逐步滑动吗？

# 特征图

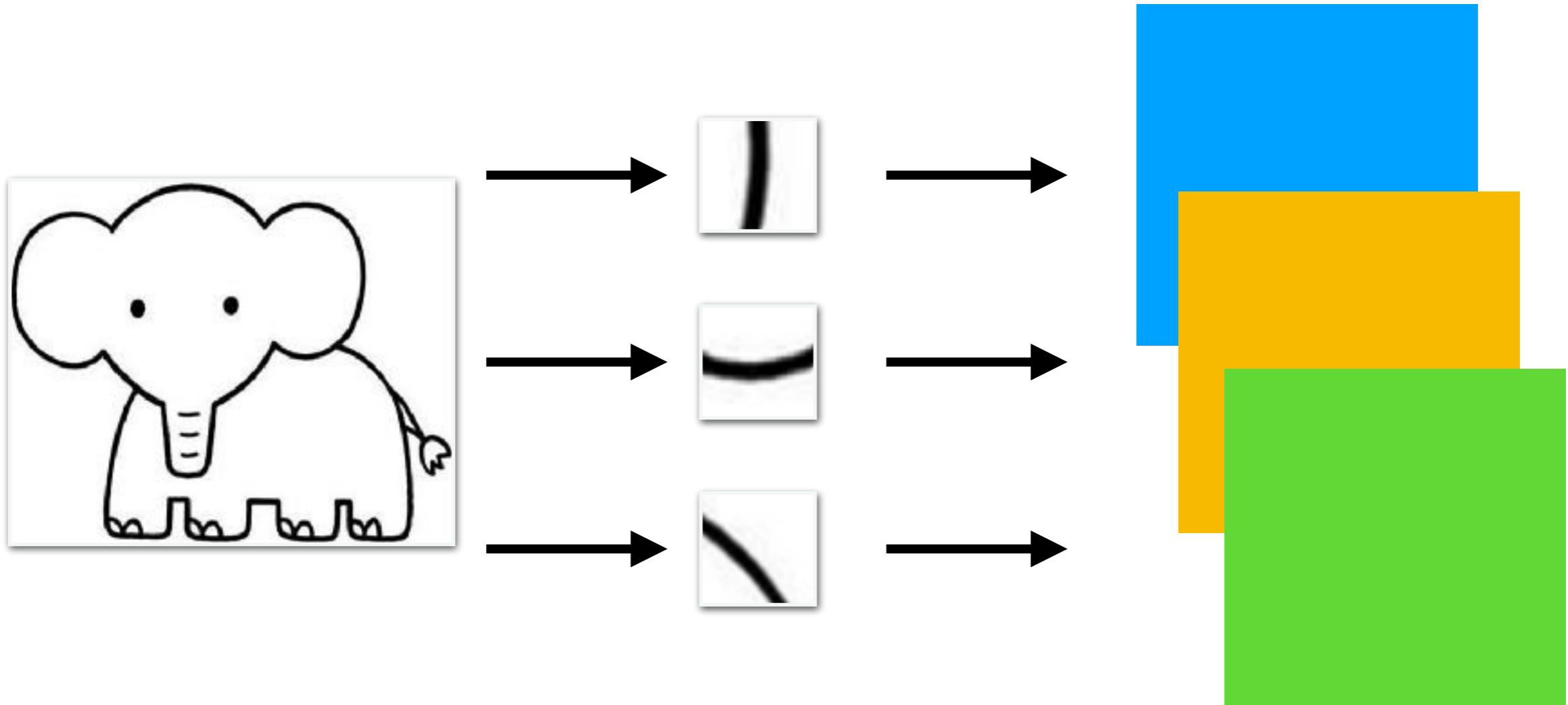


0	10	5	5	7	10	10	5	5	0	0	0
0	0	0	0	0	0	10	0	69	0	0	0
70	0	0	1	0	2	0	0	90	0	0	0
80	0	0	0	0	0	0	0	100	0	0	0
20	40	0	0	0	0	40	0	0	0	0	0
0	0	10	80	80	10	0	30	5	10	0	0
0	60	0	80	80	0	0	0	0	0	60	0
0	60	10	10	10	0	2	1	2	5	60	50
70		70	80	0	80	0	0	70	80	0	80
80	0	80	80	0	80	0	0	80	80		80

n个“边”通过卷积可以得到n个特征图。n个特征图就是原图的另一种表达形式。

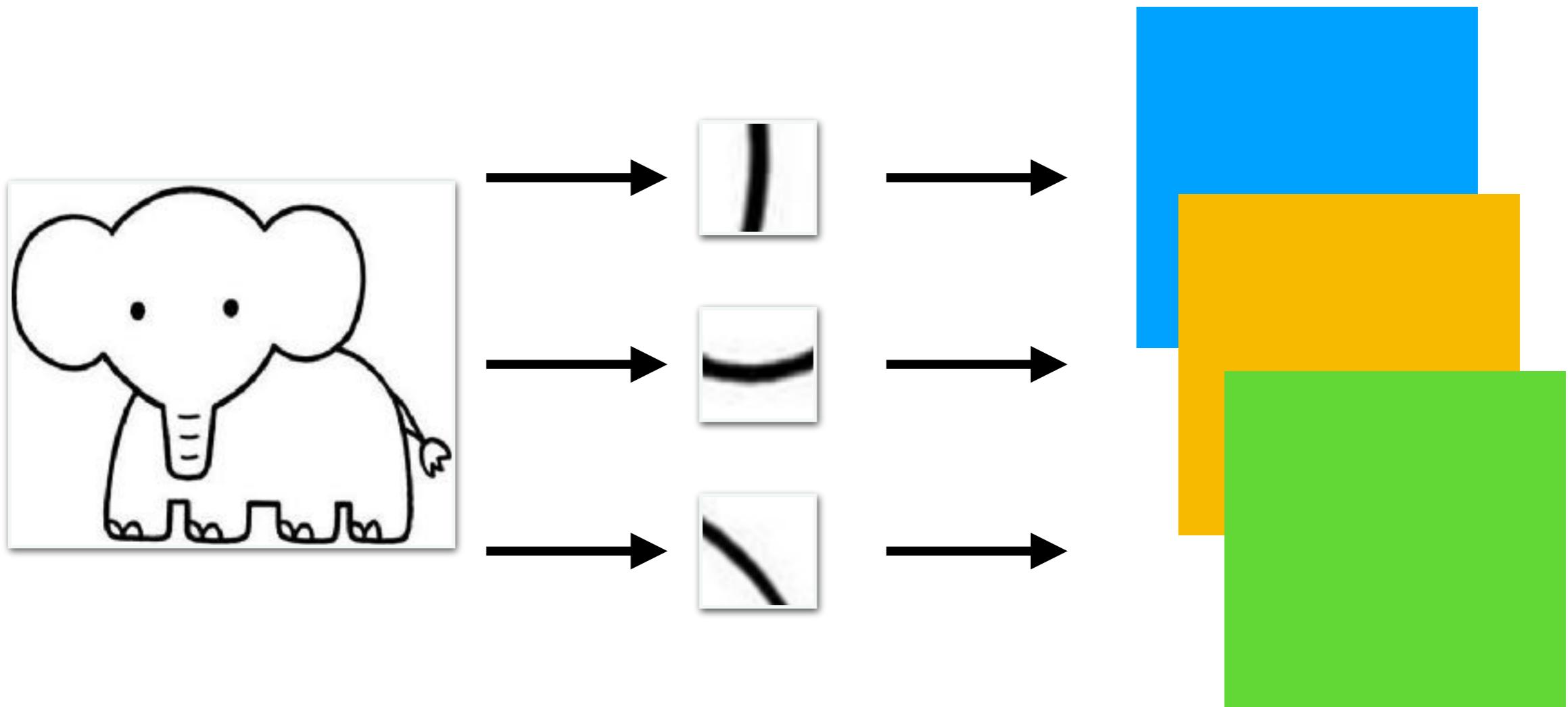
思考：1个“边”的特征分布图能  
表示图片的特征分布吗？

# 多核卷积



使用不同的卷积核（即“边”）可以得到不同的特征图。  
不同的特征图代表了不同特征（“边”）在原图中的分布情况。

# 特征图通道



n个卷积核卷积之后可以得到n个特征图，也被称为n个通道的特征图。

使用“通道”代替图片与特征图：1个通道卷积生成3个通道。

# 多核卷积实例

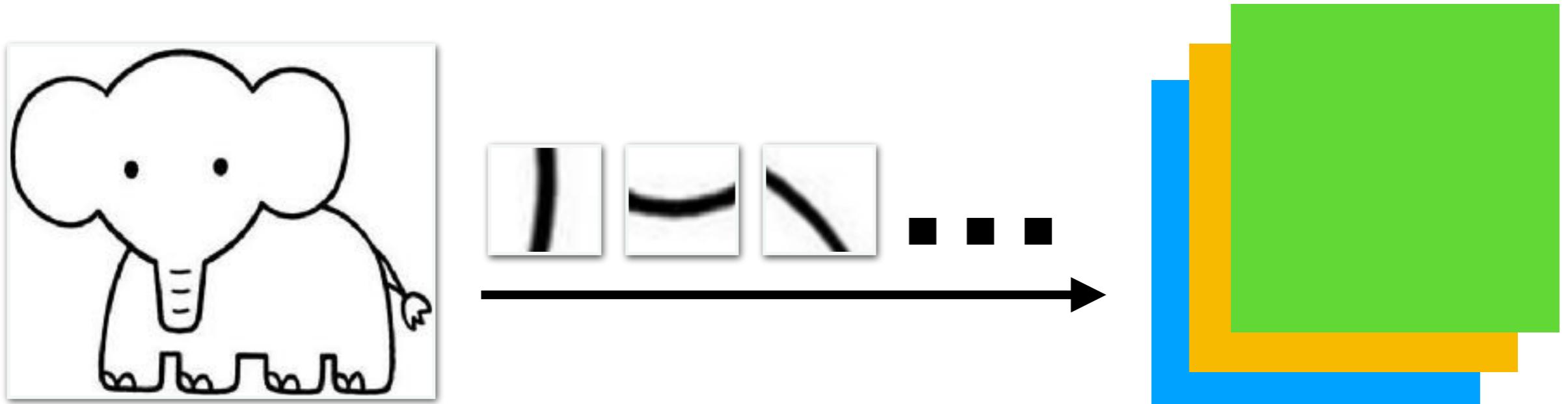


一个手写数字“2”，使用20个卷积核卷积后的特征图。

**思考：特征图保留了原图的那些信息？**

**特征图与原图的关系是什么？**

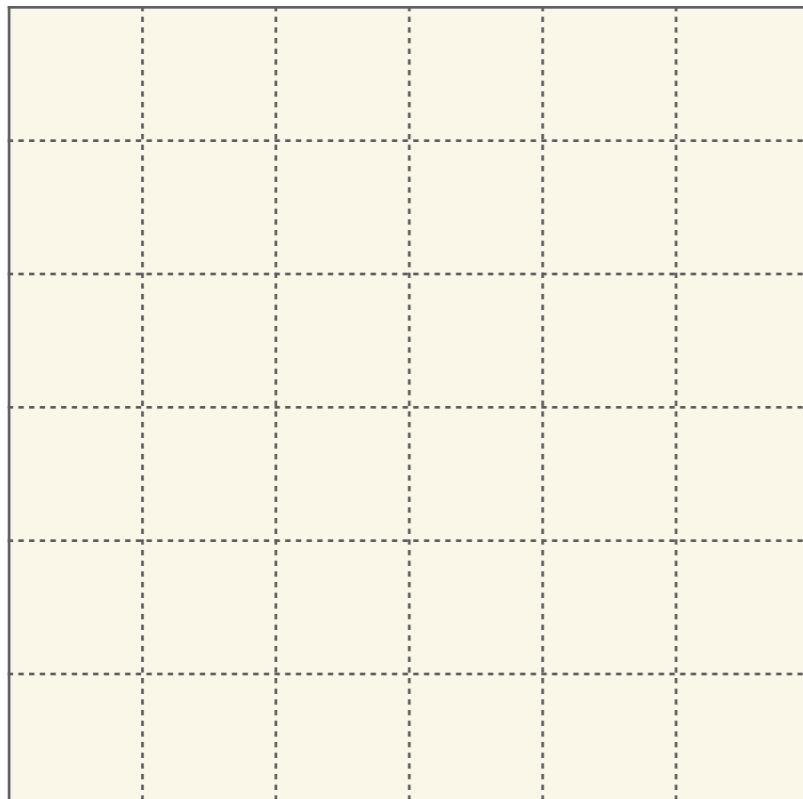
# 特征图与原图的关系



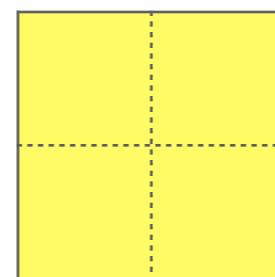
卷积核保留了原图的局部特征；特征图保留了局部特征的强弱和位置信息。

卷积核与特征图可以看做原图的另一种表达形式。

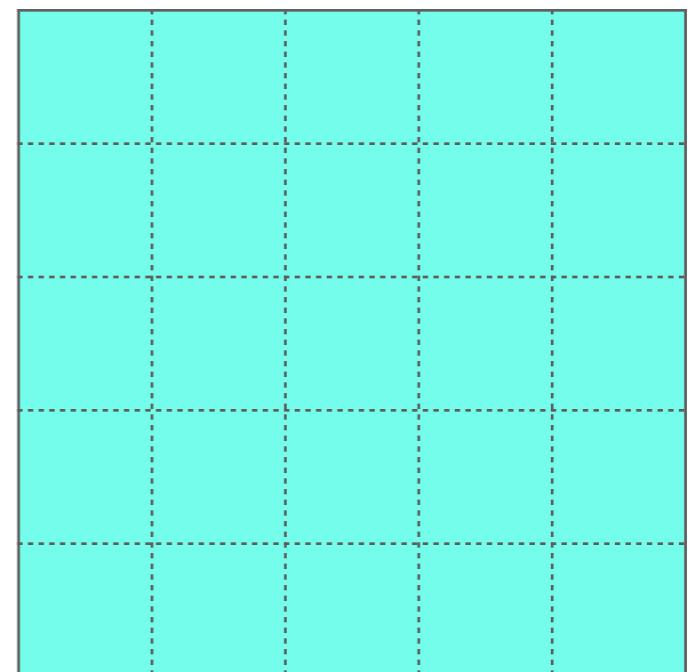
# 特征图与原图的大小关系



原图  $6 \times 6$



卷积核  $2 \times 2$



特征图  $5 \times 5$

原图的大小大于特征图的大小

# 小练习

1. 对 $10\text{px} \times 10\text{px}$ 的图片使用 $3\text{px} \times 3\text{px}$ 的卷积核做卷积得到的特征图大小是多少？
2. 对 $10\text{px} \times 10\text{px}$ 的图片使用 $2\text{px} \times 3\text{px}$ 的卷积核做卷积得到的特征图大小是多少？

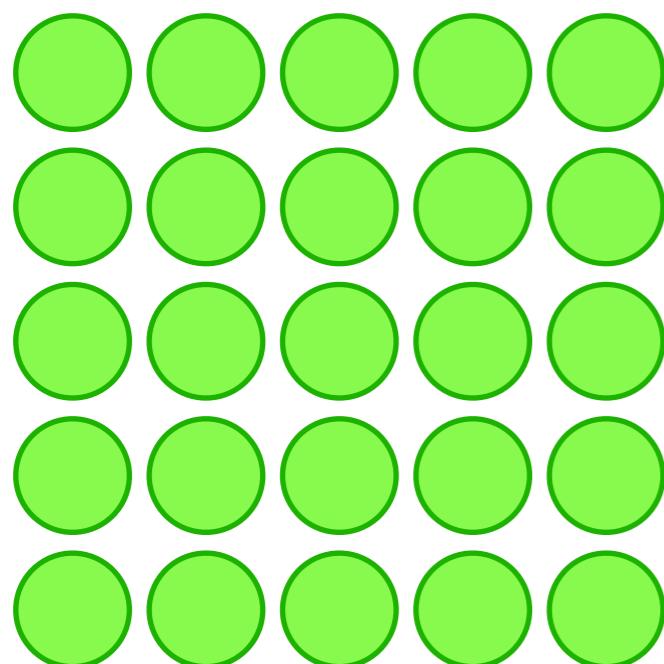
# 思考：卷积的本质是什么？

局部特征提取。

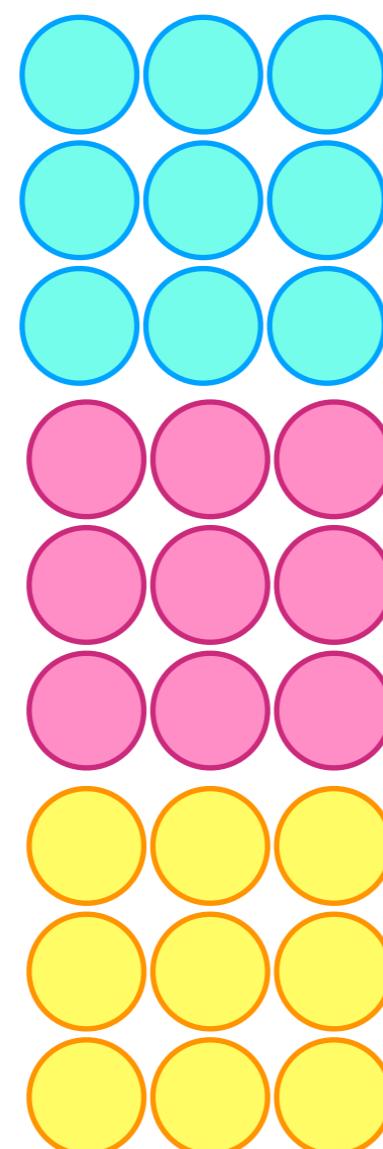
## 4.4 池化

思考：可否使用特征图代替原图输入  
到神经网络？这样做有什么问题？

# 特征图作为输入



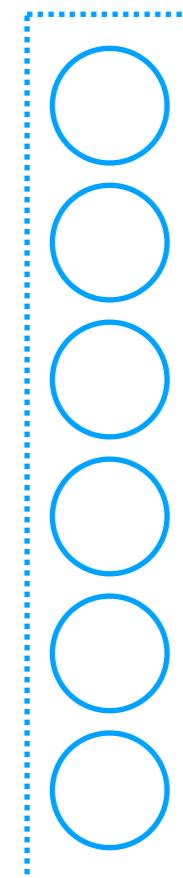
5px\*5px的图片



3次卷积得到的特征图

特征图 A  
特征图 B  
特征图 C

特征图展开  
后与下一层  
全连接。

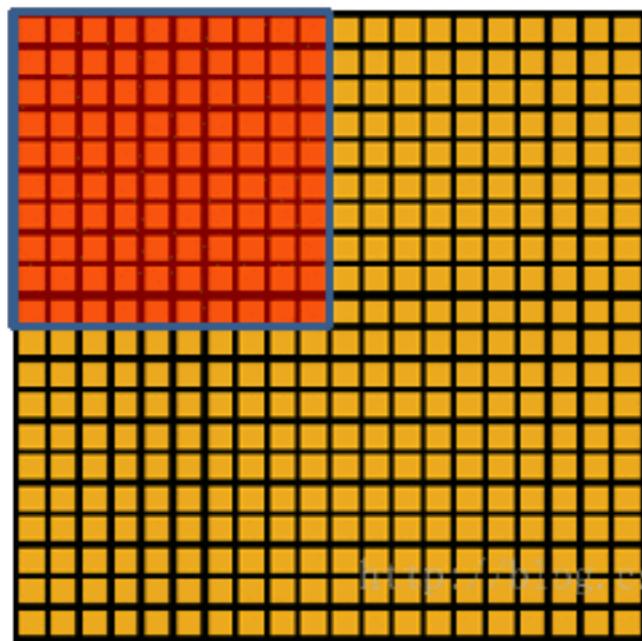


全连接层

# 特征图直接作为输入的优缺点

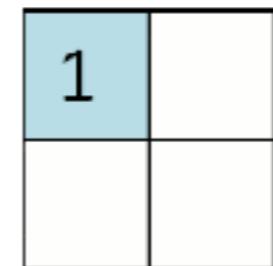
- 提取到了局部特征作为输出。
- 卷积使得图像具有了一定的平移不变性。
- 特征图展开破坏了高维的位置（全局）信息。
- 直接使用特征图输入全连接神经网络时参数规模较大。

# 特征图下采样——池化



Convolved  
feature

Pooled  
feature



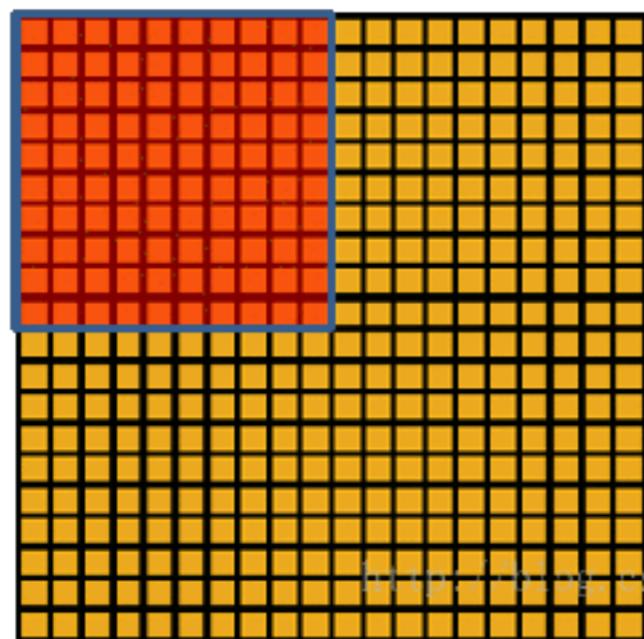
滑动窗大小：相对较大。

滑动步幅：通常等于滑窗大小。

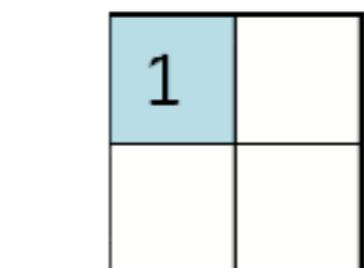
最大池化：取一个区域的最大值。

平均池化：取一个区域的平均值。

# 特征图下采样——池化



Convolved  
feature



Pooled  
feature

池化：在允许损失少量信息的情况下，对特征图进行下采样。

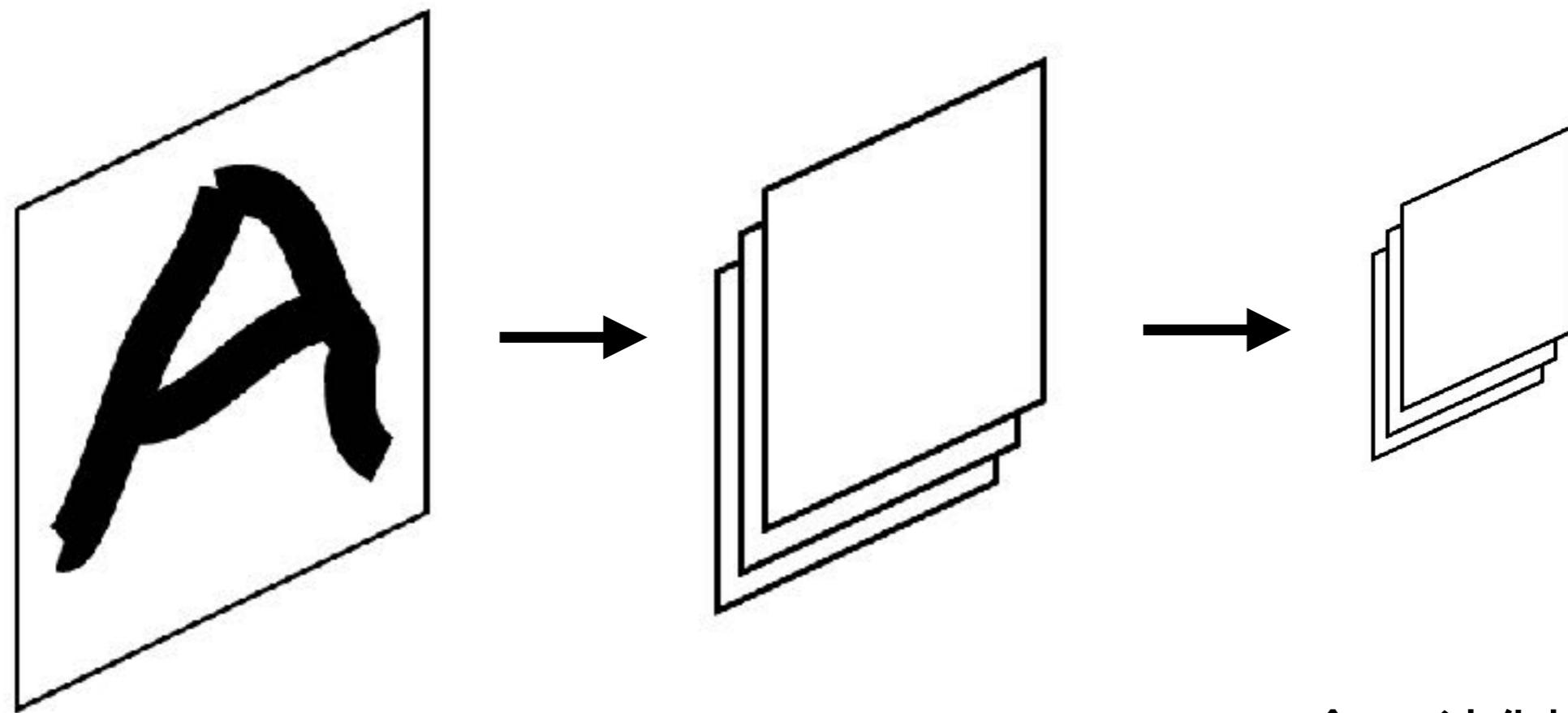
优点：使得图像获得了一定的平移不变性；使得特征图的规模减小。

# 小练习

1. 对 $6 \times 6$ 的特征图使用 $3px \times 3px$ 的滑动窗做池化得到的新特征图大小是多少？
2. 对 $10px \times 10px$ 的图片使用5个 $3 \times 3$ 的卷积核做卷积得到的特征图，再以 $4 \times 4$ 的滑动窗做池化，得到了几张特征图？每张特征图的大小是多少？

# 5. 卷积与池化的黄金组合

# 卷积与池化组合



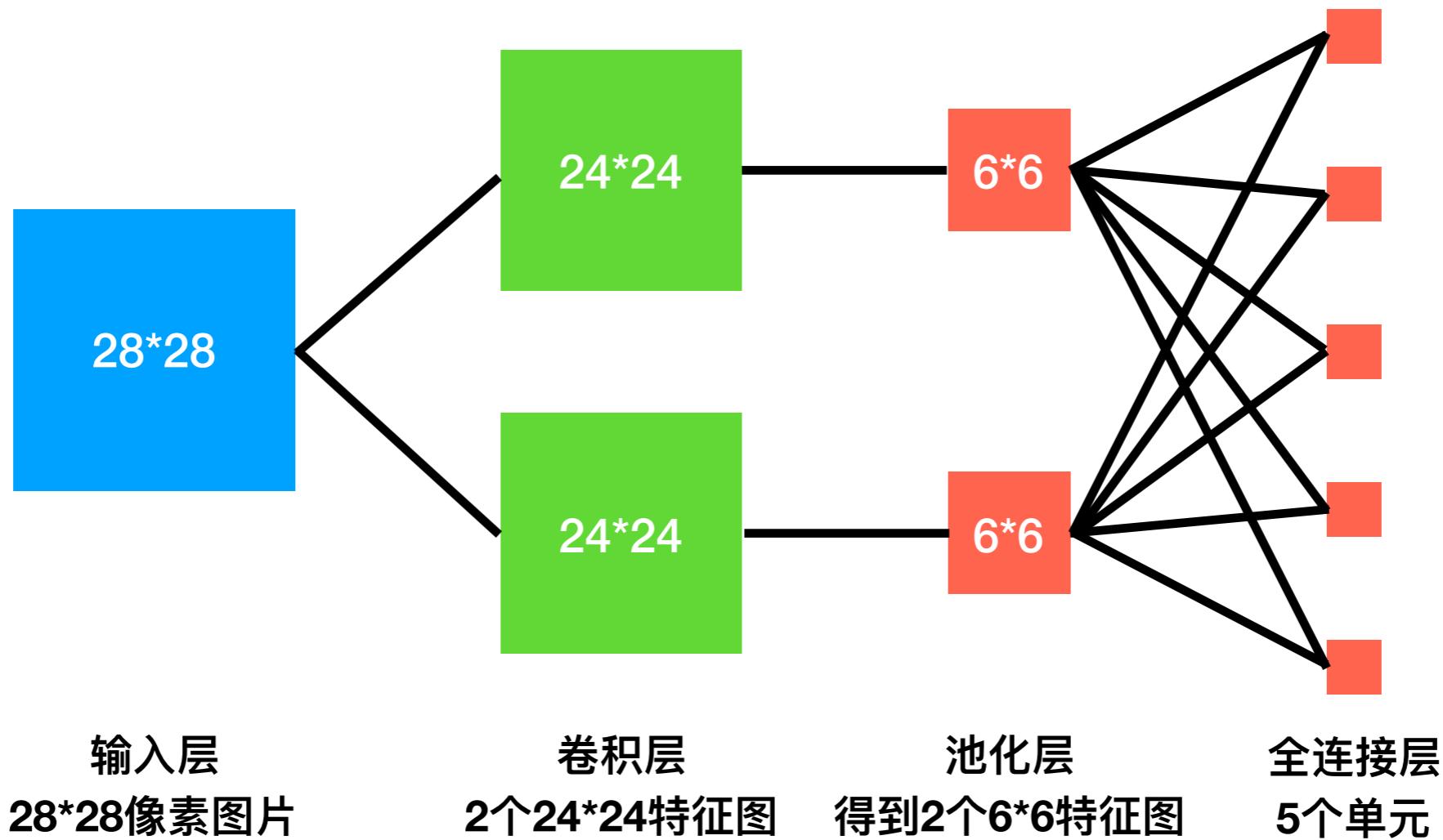
28\*28像素的图像

3个由5\*5卷积核得到  
的特征图：3\*24\*24

3个6\*6池化核池化后  
的特征图：3\*4\*4

通过卷积与池化，可以有效的提取数据中的特征，并达到降低输入纬度的目的。

# 简单的卷积神经网络结构图



# 卷积神经网络例子: LeNet

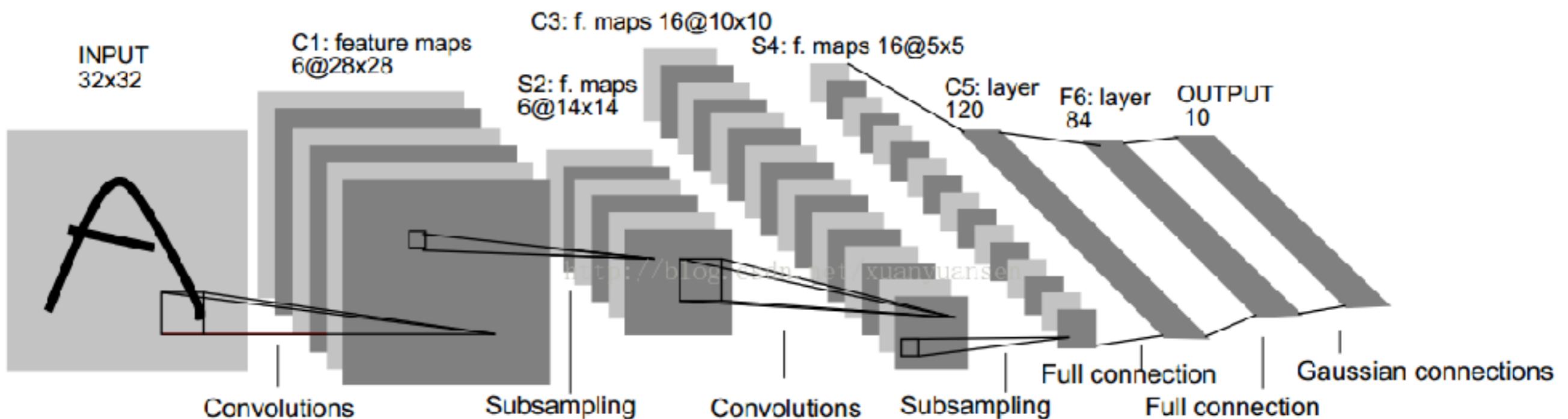
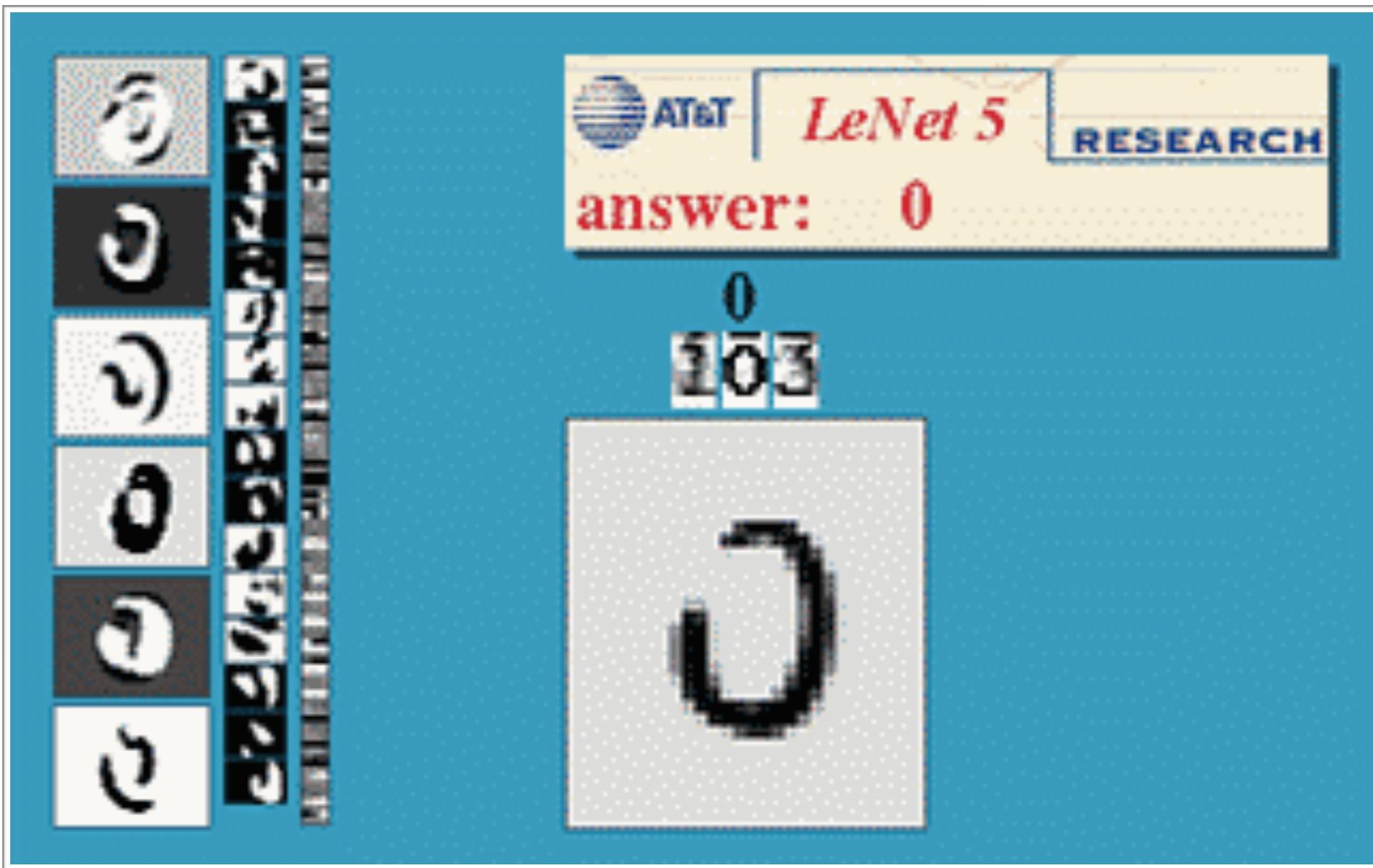


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

论文: [《Gradient-Based Learning Applied to Document Recognition》](#)

# 卷积神经网络:LeNet



LeNet曾是美国各大银行等大机构最常用的识别手写体的算法。

# 小结

- 卷积神经网络通常包含多个卷积层、池化层以及全连接层。
- 卷积层利用了数据的局部相关性，可以进行局部特征提取。
- 卷积核即为图像的局部特征——“edge”，卷积核与图像操作产生的结果是特征图，代表了当前“edge”在原图上的分布。
- 一个卷积核与原图作用可以得到一个特征图，通常需要多个卷积核对图片进行特征提取，才能较为完整的提取到图片的所有特征。
- 通过对卷积得到的特征图下采样（池化）可以降低特征图的维度。
- 卷积与池化使得图像具有了平移不变性。
- 卷积与池化组合使用，既提取到了特征又降低了神经网络规模。

# 下节内容

- 多通道卷积。
- 卷积相关的计算。
- 卷积与池化的灵活应用。

# THANKS