

## 卷积神经网络 (1)



深度学习

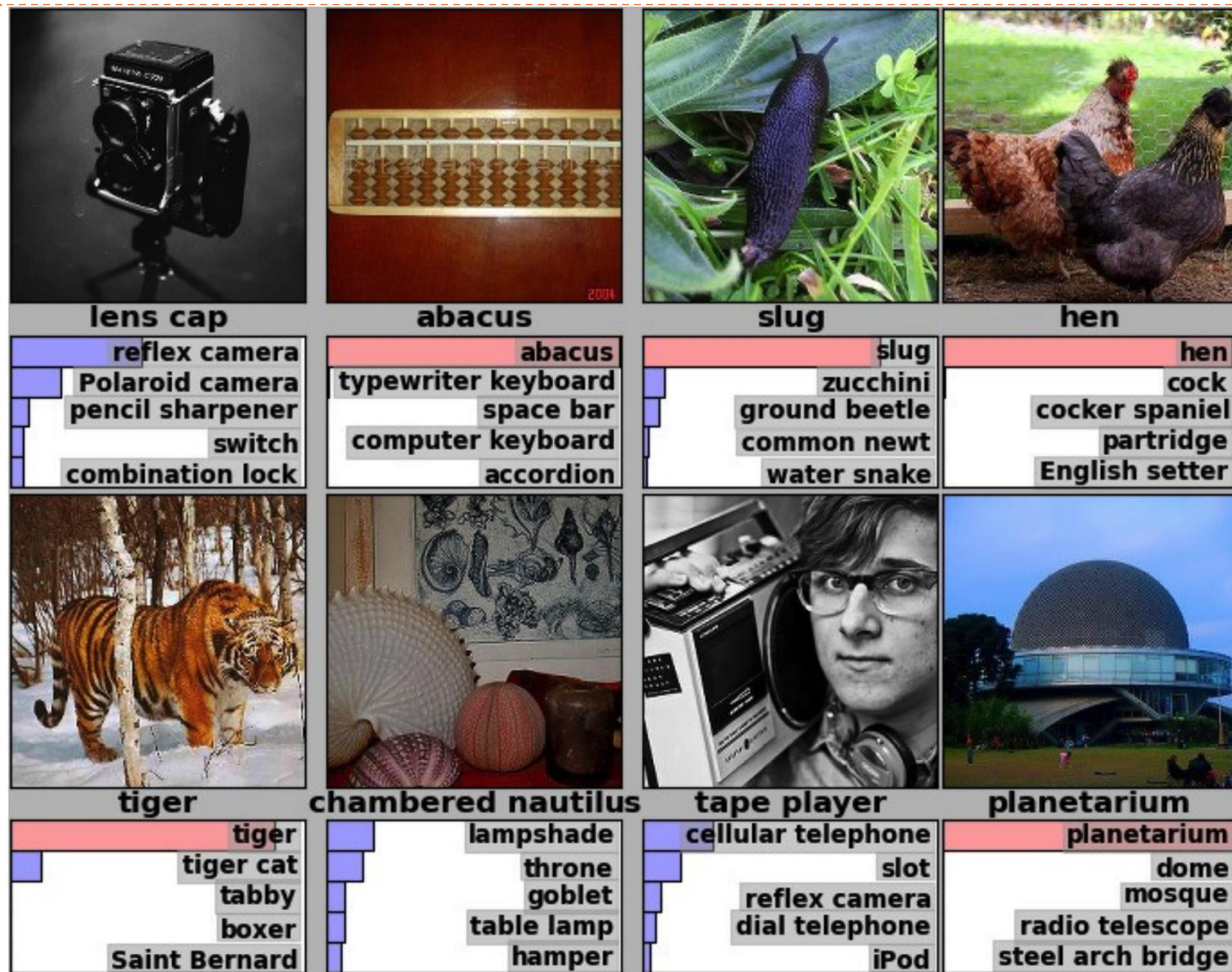
# 概览

---

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。
  1. 认识感受器与感受野。
  2. 使用“edge”特征描述图片。
  3. 卷积。
  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

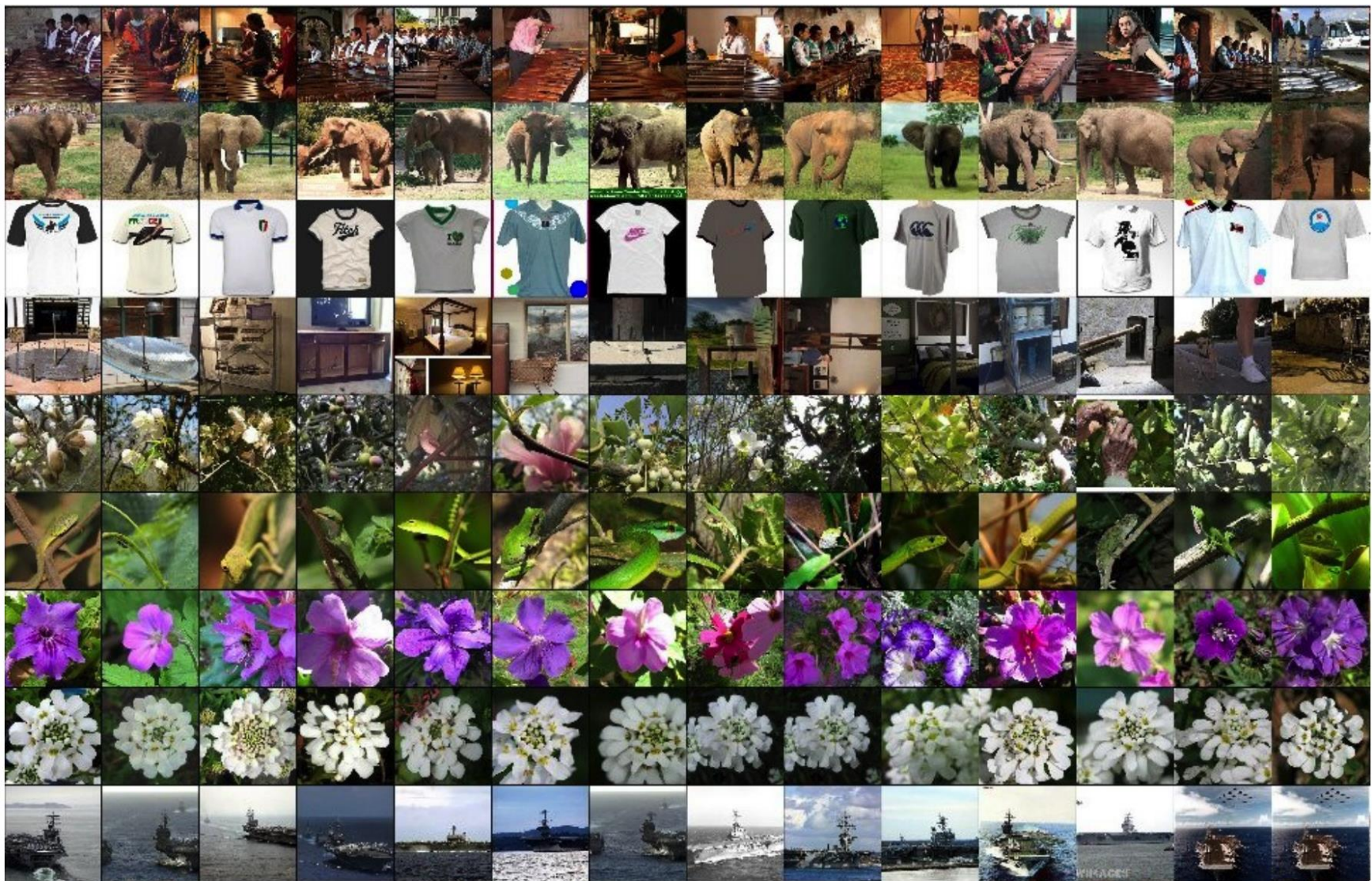
# 分类



来源: [ImageNet Classification with Deep Convolutional Neural Networks](#)  
《深度学习》



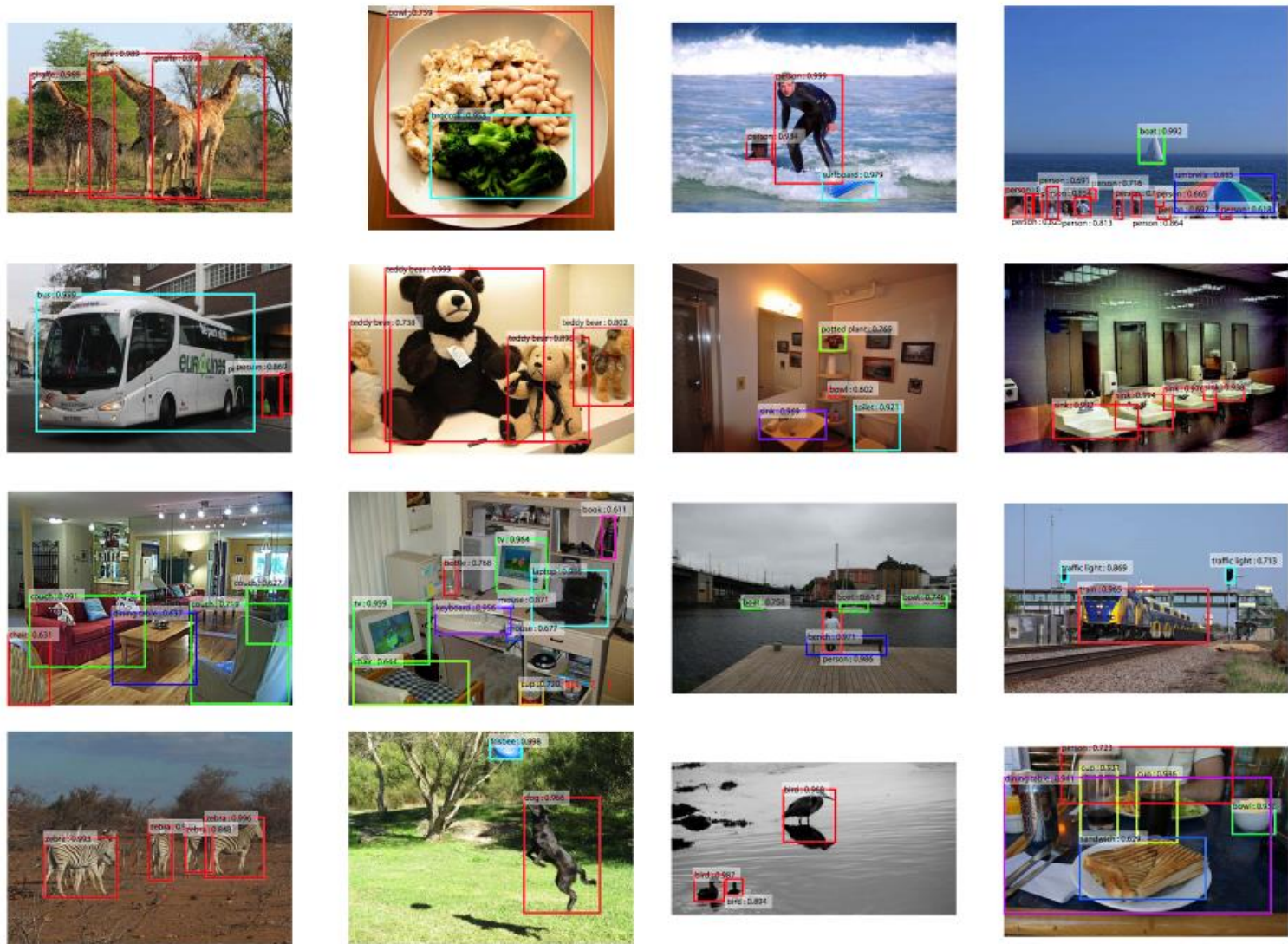
# 检索



来源: [ImageNet Classification with Deep Convolutional Neural Networks](#)  
《深度学习》



# 目标检测



来源: [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](#)  
《深度学习》

# 图片分割



来源: [Learning Hierarchical Features for Scene Labeling](#)  
《深度学习》



# 图片转文字



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



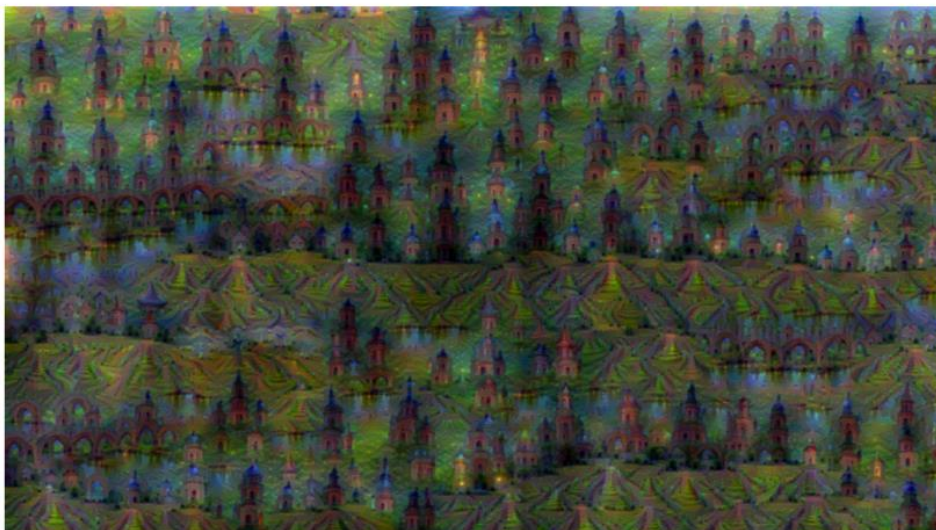
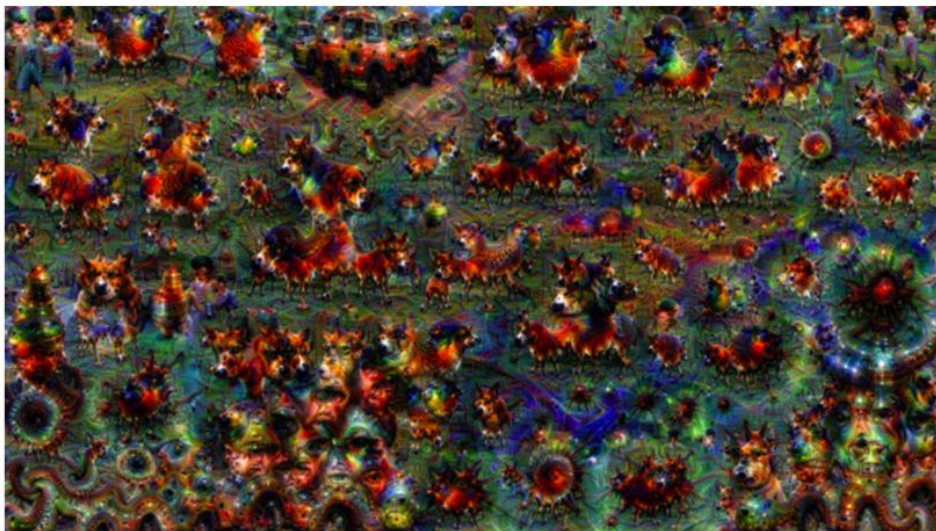
a pile of luggage sitting on the ground



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

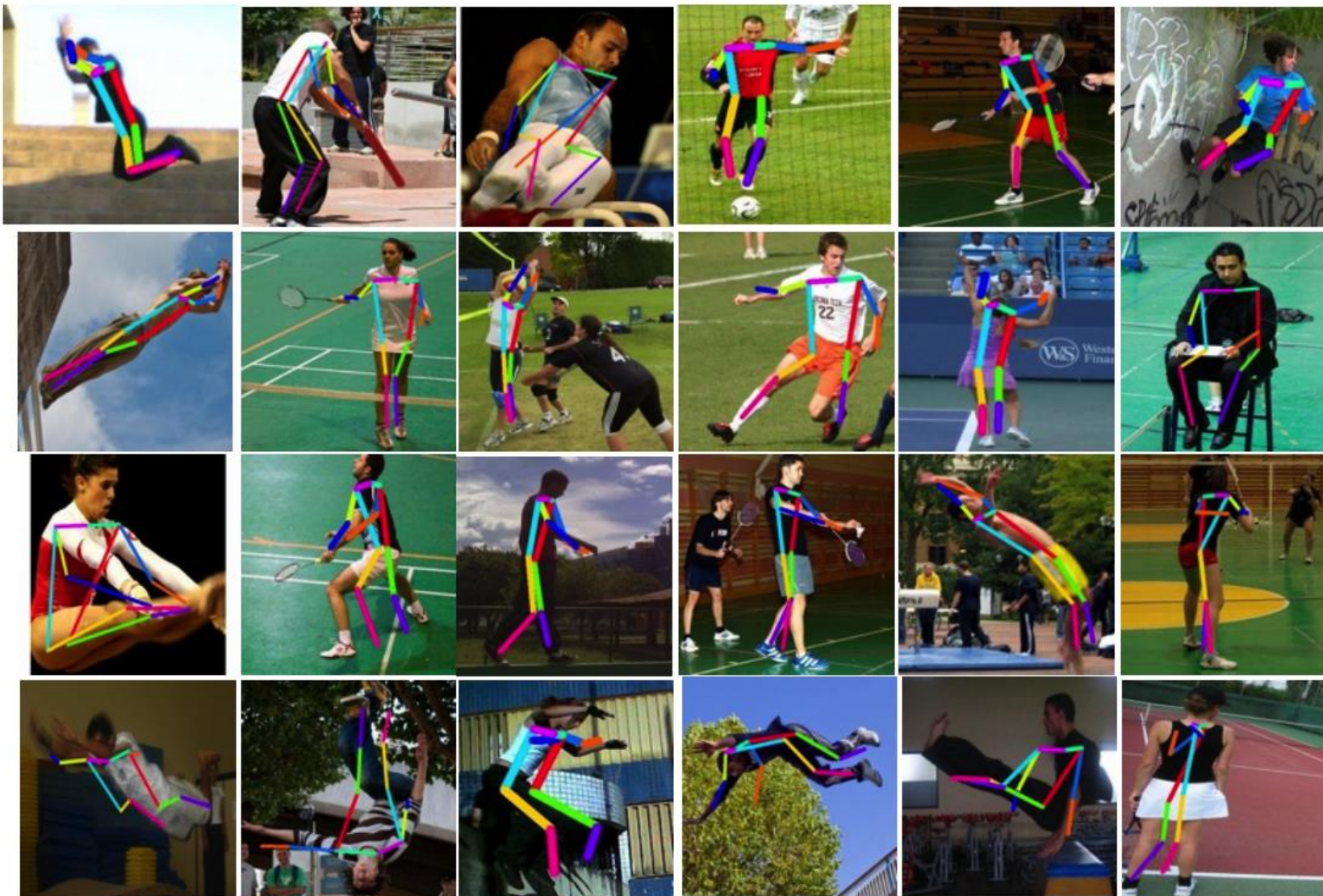


# 图片风格化





# 人体姿态估计



来源: [DeepPose: Human Pose Estimation via Deep Neural Networks](#)  
《深度学习》





## 2. 视觉是什么？

# 谁看到了真实的世界？

---





# 谁看到了真实的世界？

---



# 视觉的本质

---



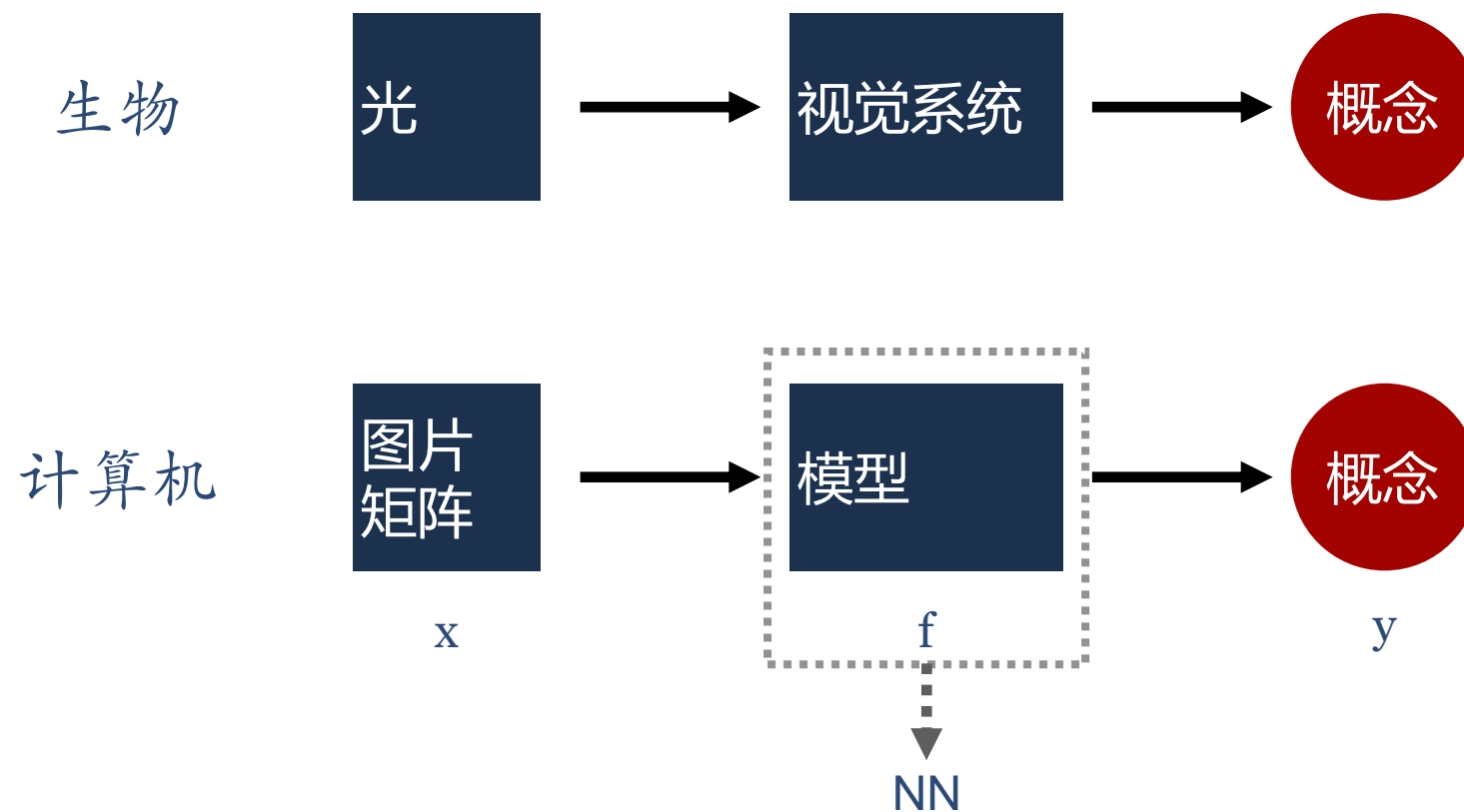
视觉系统本质上是对外界刺激进行抽象表示的系统。

我们所看到的世界是主观的世界，不是世界的本来面目。

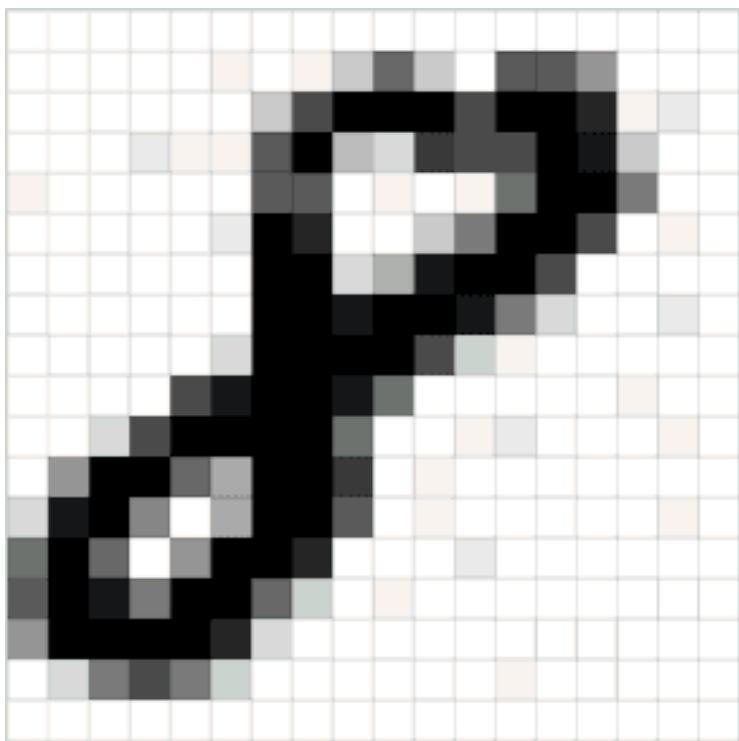


# 计算机视觉

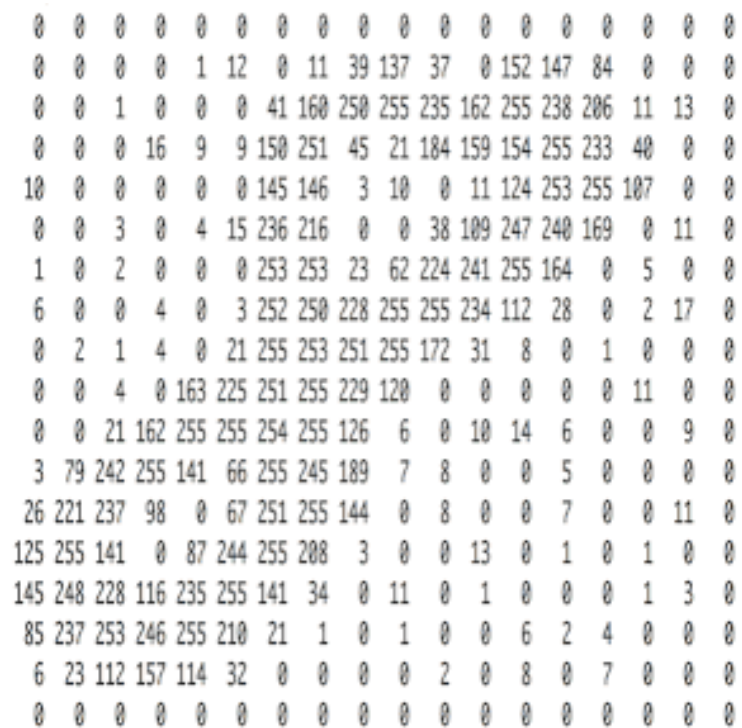
---



# 视觉与机器视觉

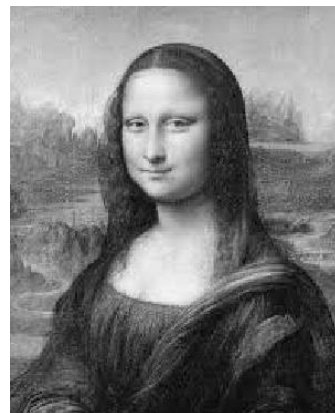


## 人的视觉

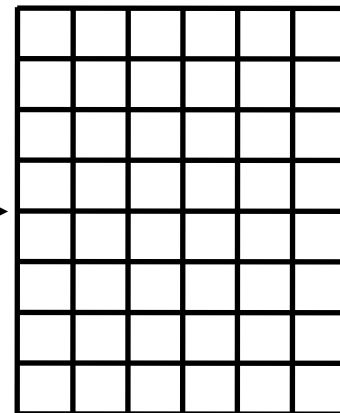


# 计算机

# 图片的通道



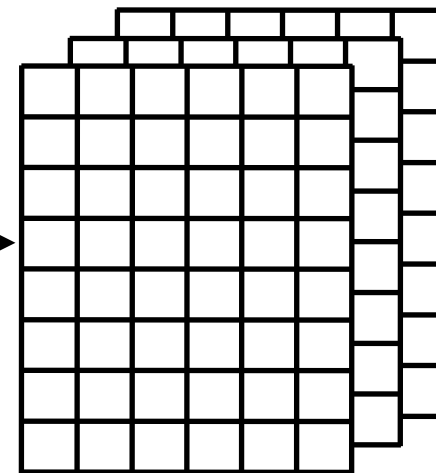
灰度图像



图片矩阵



RGB图像



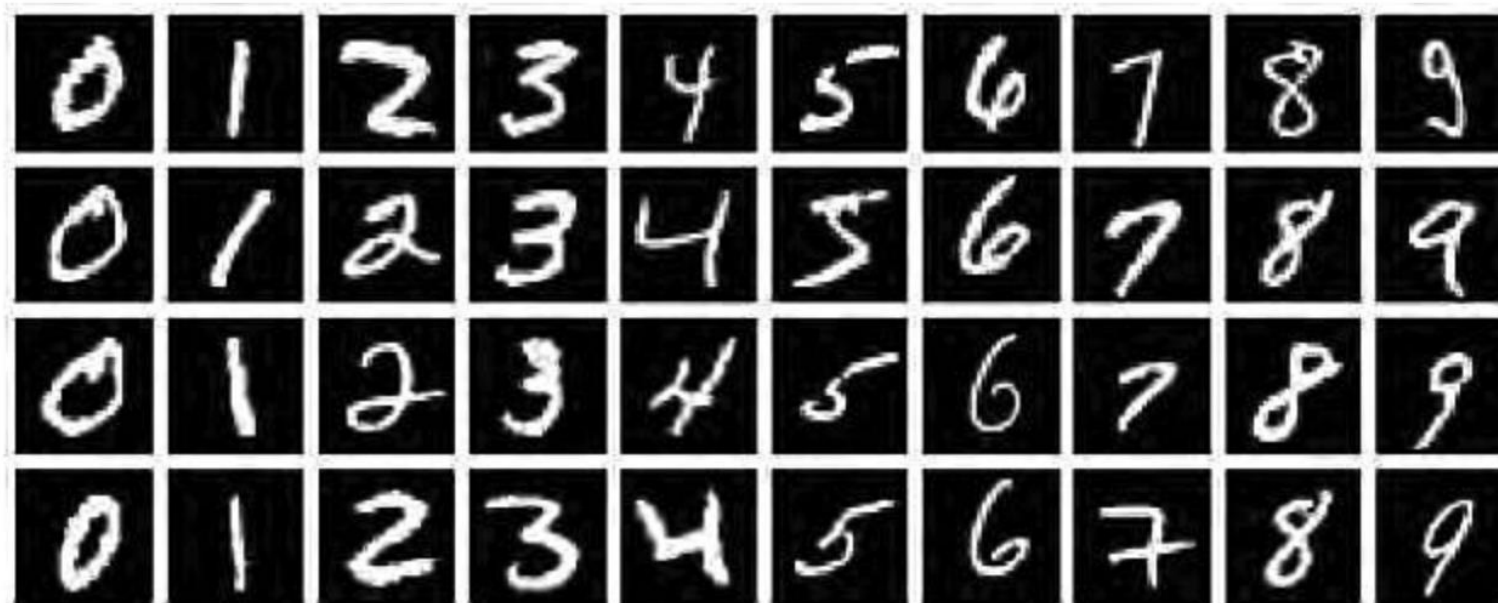
图片矩阵



### 3. 使用ANN实现 图片分类

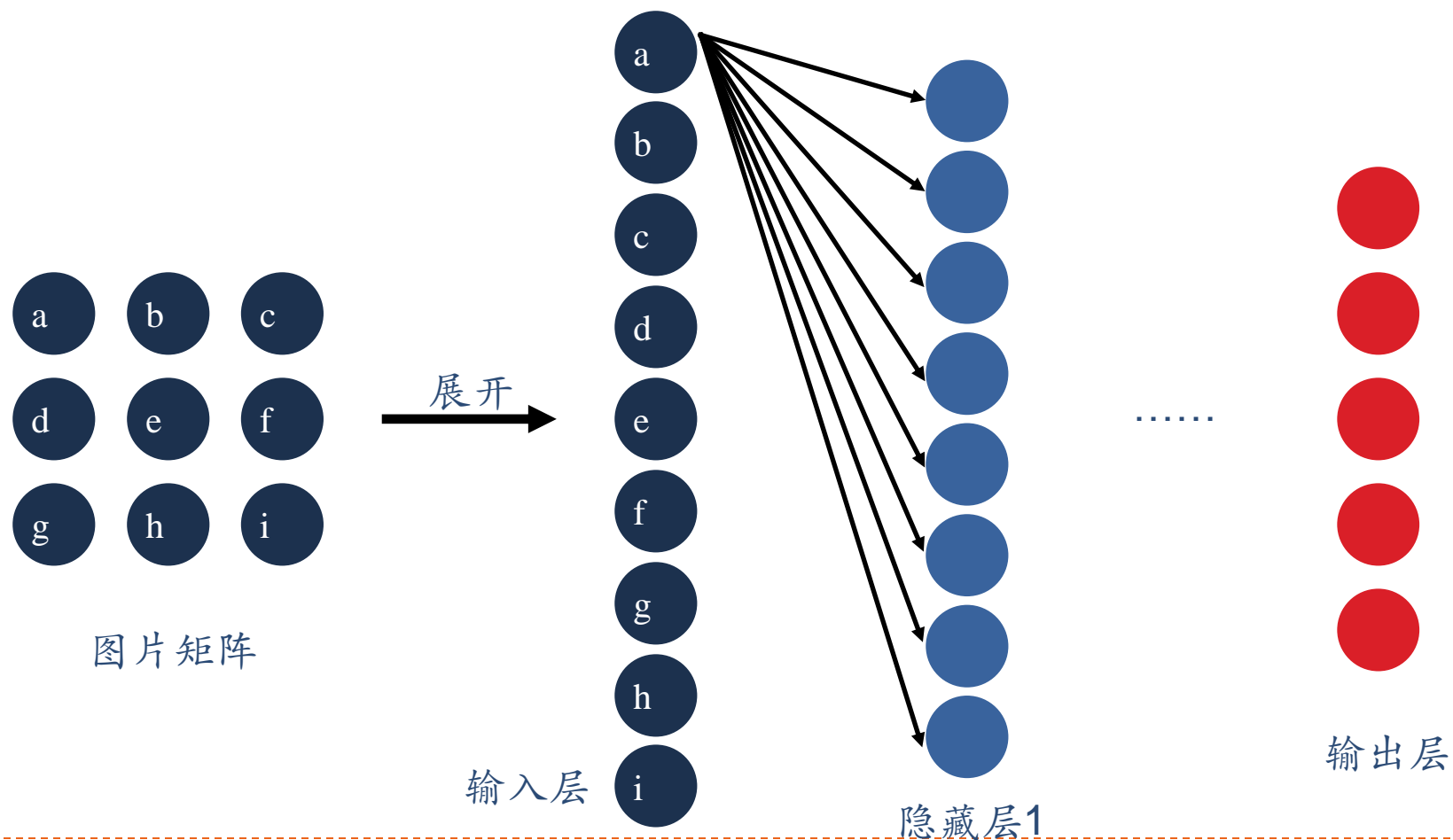
# 使用全连接NN模型实现图片分类

---



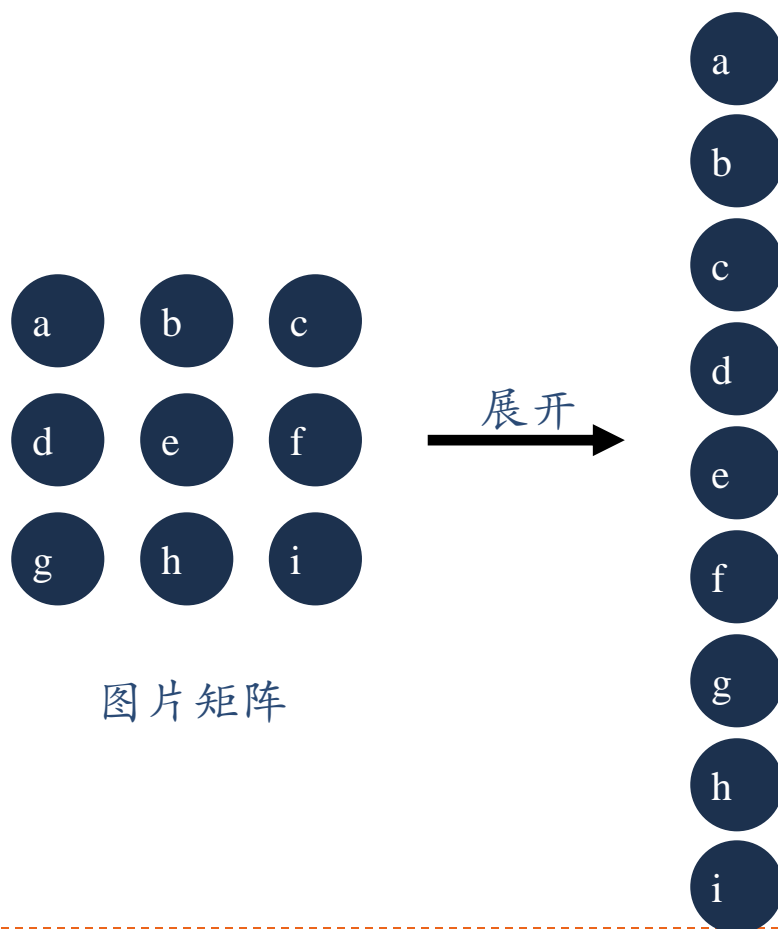
思考：如何将图片输入给神经网络模型，并进行分类？

# 使用全连接网络实现图片分类





# 使用全连接网络实现图片分类



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

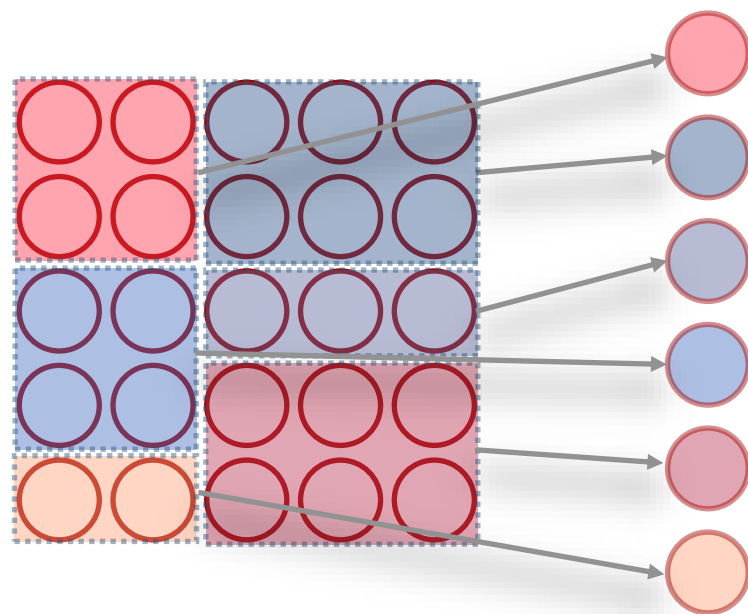
---

卷积神经网络(**Convolutional 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。



## 4.1 认识感受器与 感受野

# 感受器与感受野



图片

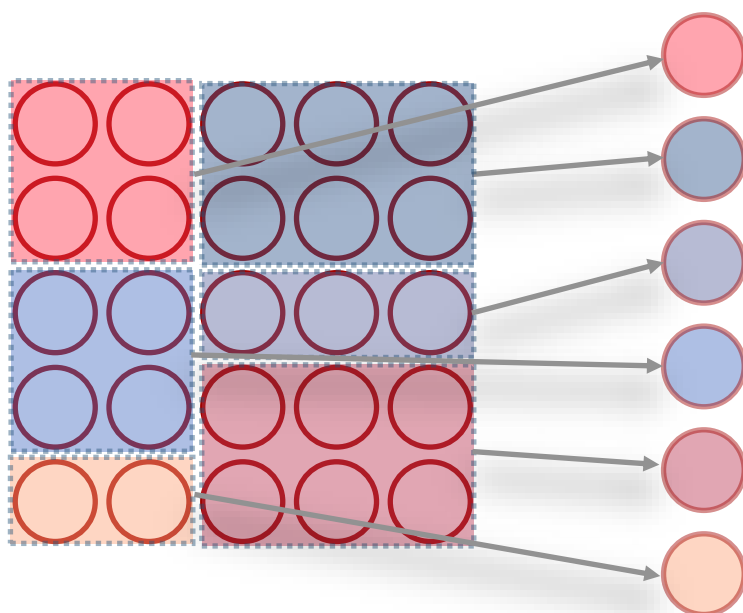
**局部感知：**图像的局部联系紧密，局部像素可以作为整体的一个特征。

**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



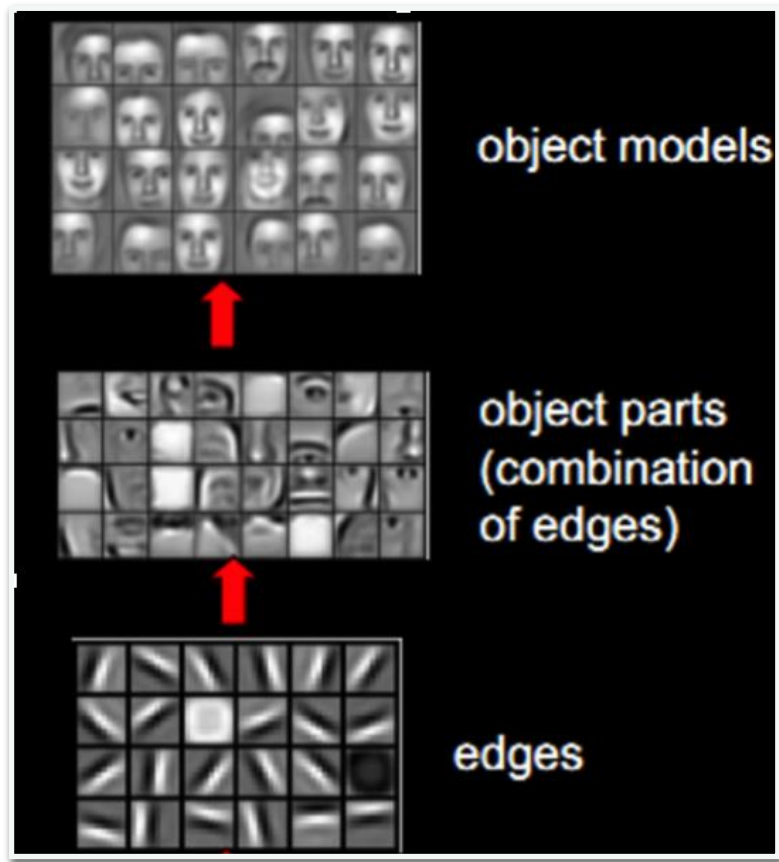
图片

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内容丰富的各种图片。
3. 通过统计发现大约需要20种左右的边即可描述各种各样的图片。

一个模型中构成人脸的“边”

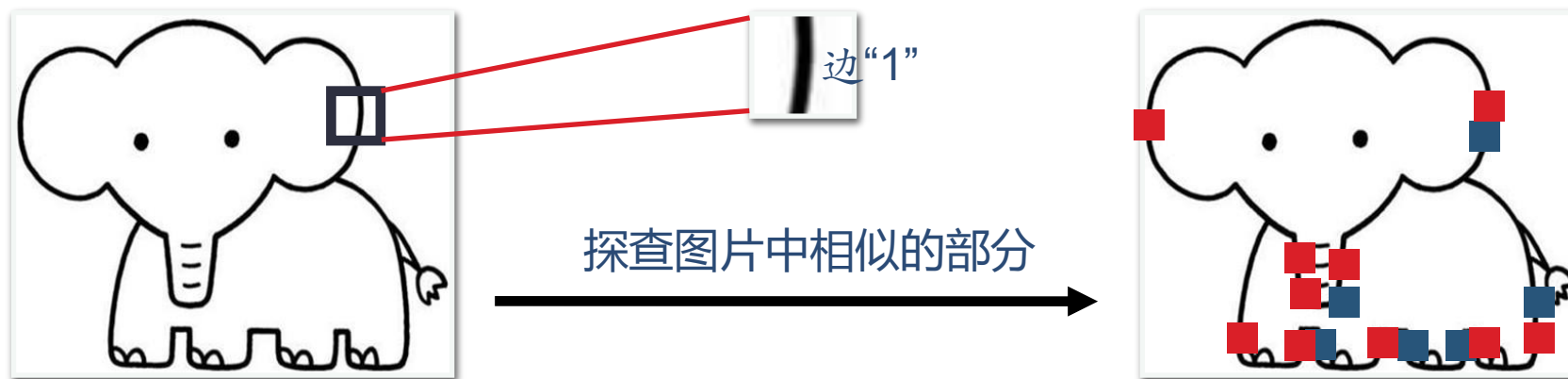
## 4.2 使用“边”特征 描述图片

---

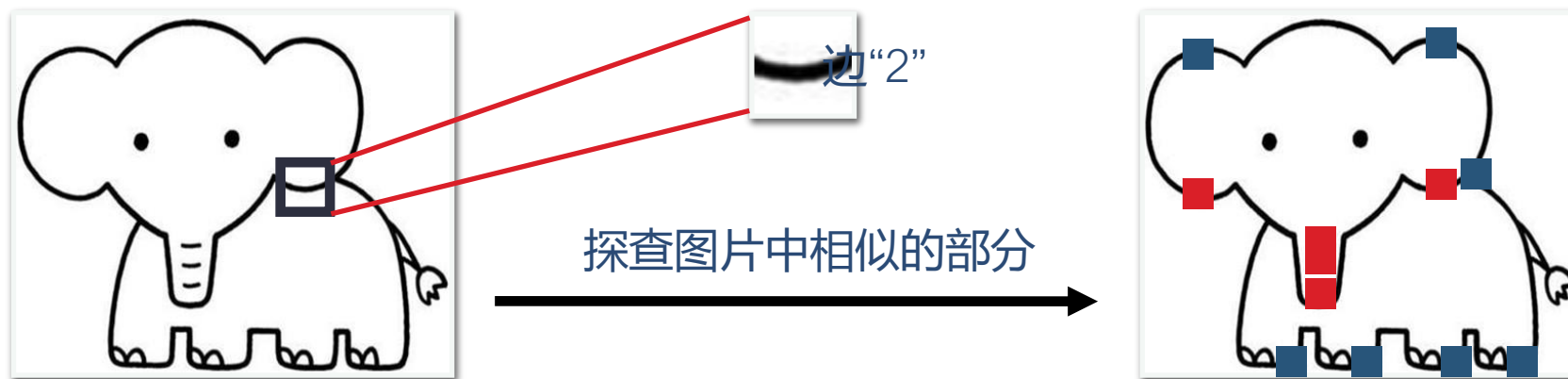
思考：统计发现构成数字图像的边都是近似的，假如构成图像的“边”已知，如何使用这些“边”描述现有图像？



# 查找“边”在图像中的位置



# 查找“边”在图像中的位置

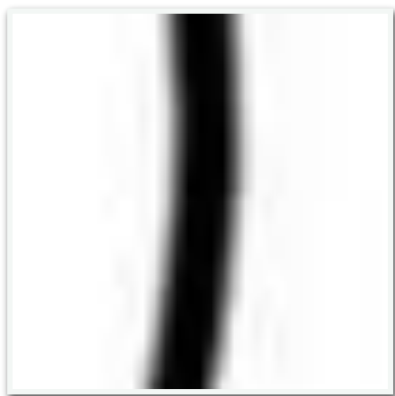


人通过观察可以找到与“边”相似的部分，计算机该如何查找呢？



# “边”矩阵

---



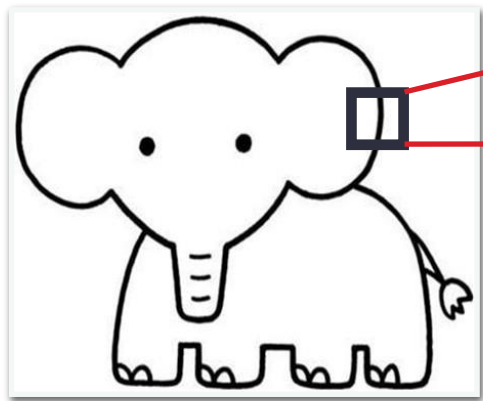
边“1”



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

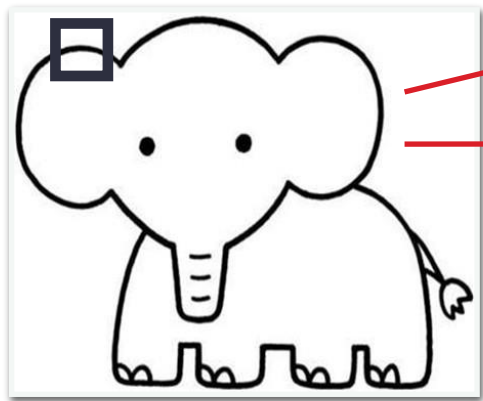
\*

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



边“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

\*

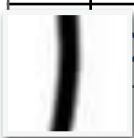
0	0	0	0	0
0	0	30	0	0
80	80	100	80	70
20	0	0	0	10
0	0	0	0	0

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



子图“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



子图“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

可以看到：一般的，与“边”接近的局部“子图”，对应元素乘积的和较大，反之较小。

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 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

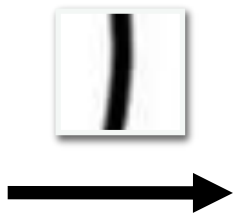
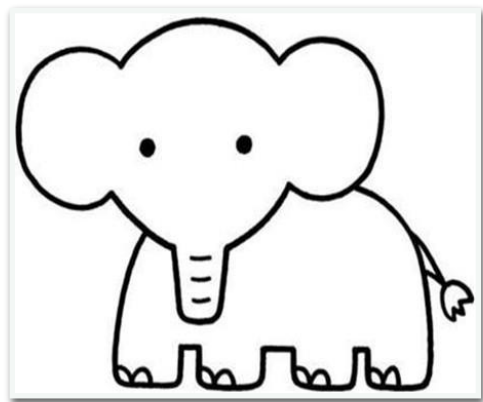
Convolved  
Feature

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\*3的“边”矩阵叫做**卷积核(Convolution 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。

将“边”与所有可能的子图运算，就是卷积的过程。

# 特征图

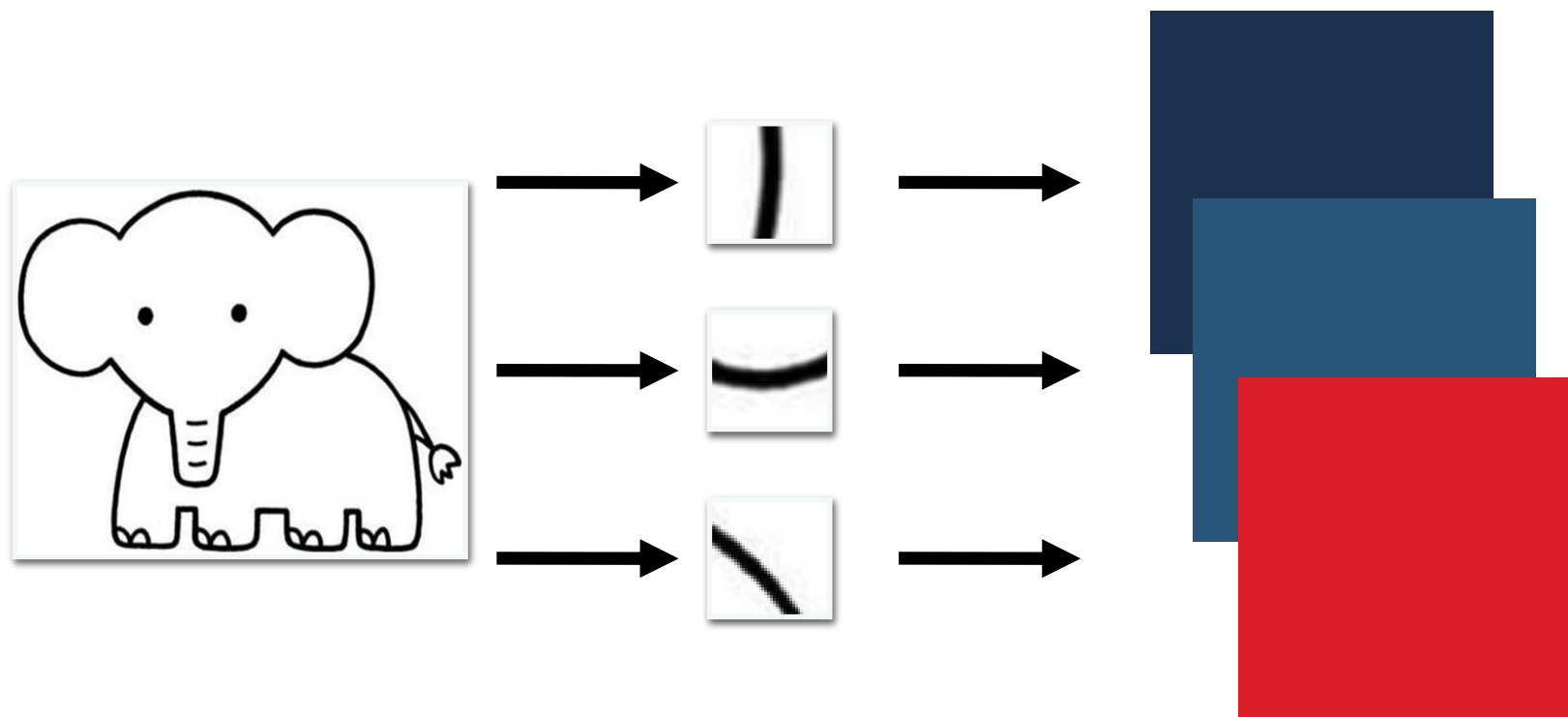


0	10	5	5	7	10	10	5	5	0	0	0
0	0	0	0	0	0	10	0	69	0	0	0
70	0	0	1	0	2	0	0	90	0	0	0
80	0	0	0	0	0	0	0	100	0	0	0
20	40	0	0	0	0	40	0	0	0	0	0
0	0	10	80	80	10	0	30	5	10	0	0
0	60	0	80	80	0	0	0	0	0	60	0
0	60	10	10	10	0	2	1	2	5	60	50
70		70	80	0	80	0	0	70	80	0	80
80	0	80	80	0	80	0	0	80	80		80

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？

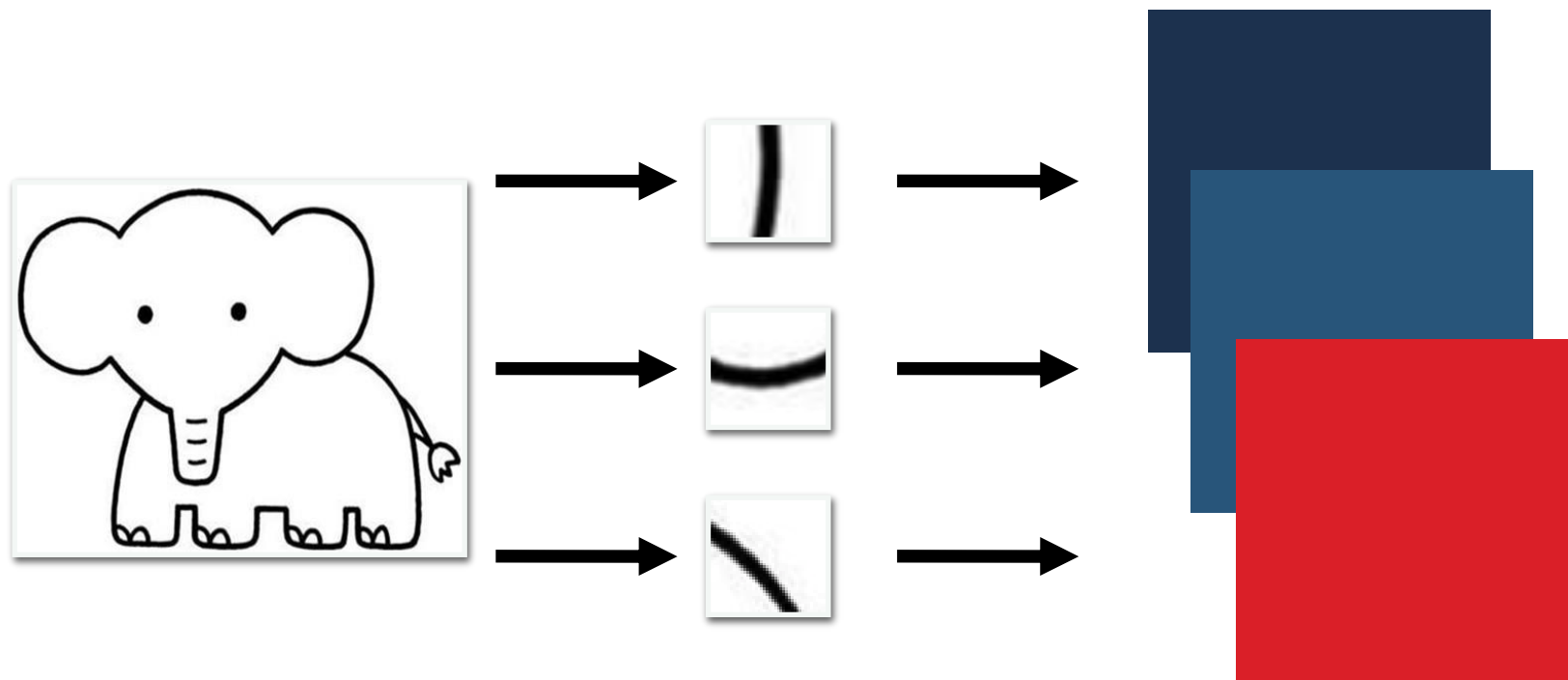
# 多核卷积



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。

为了表述方便，使用“通道”代替图片与特征图，例如：

1个通道通过3次卷积生成3个通道。



# 多核卷积实例

---



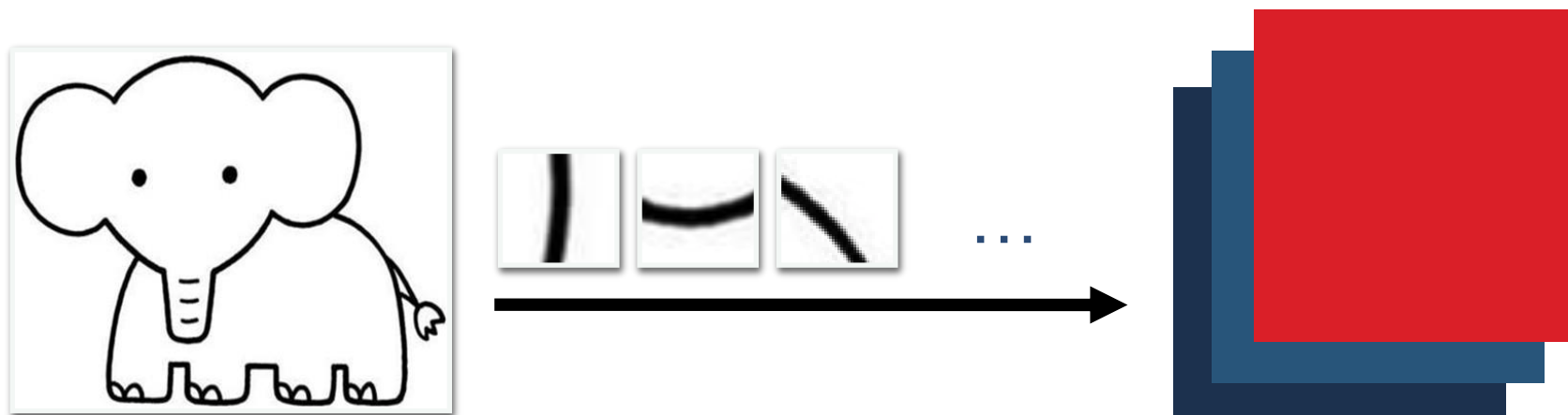
一个手写数字“2”，使用20个卷积核卷积后的特征图。

思考：

1. 特征图保留了原图的哪些信息？
2. 特征图与原图的关系是什么？

# 特征图与原图的关系

---

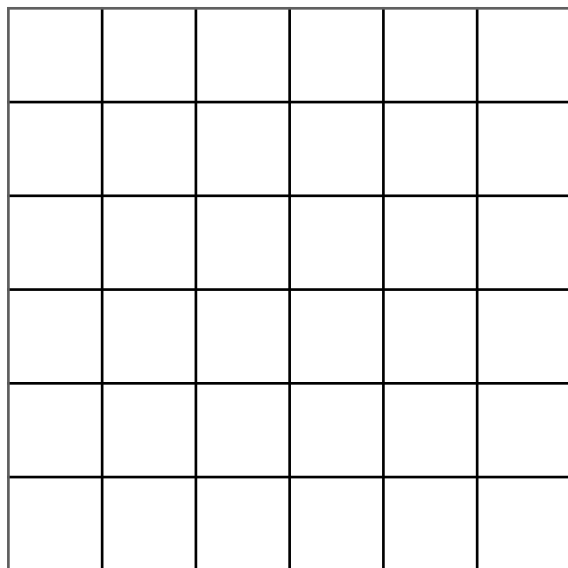


卷积核保留了原图的局部特征信息。  
特征图保留了局部特征的强弱和位置信息。  
卷积核与特征图可以看做原图的另一种表达形式。

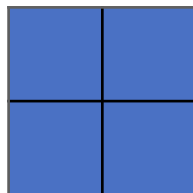
# 特征图与原图的大小关系

---

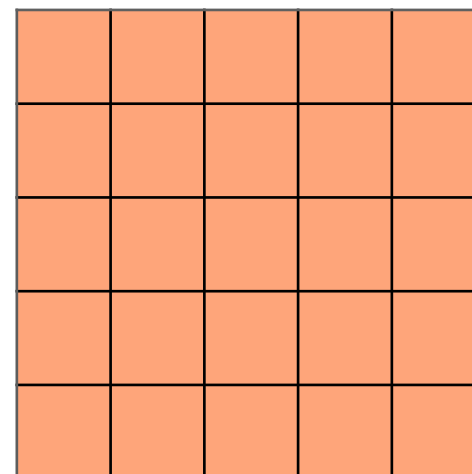
原图 6\*6



卷积核 2\*2



特征图 5\*5



一般的，原图大小大于等于卷积得到特征图的大小

---

## 小练习

1. 对“10px\*10px”大小的图片使用“3\*3”大小的卷积核做卷积得到的特征图大小是多少？
2. 对“10px\*10px”大小的图片使用“2\*3”大小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 在干什么？

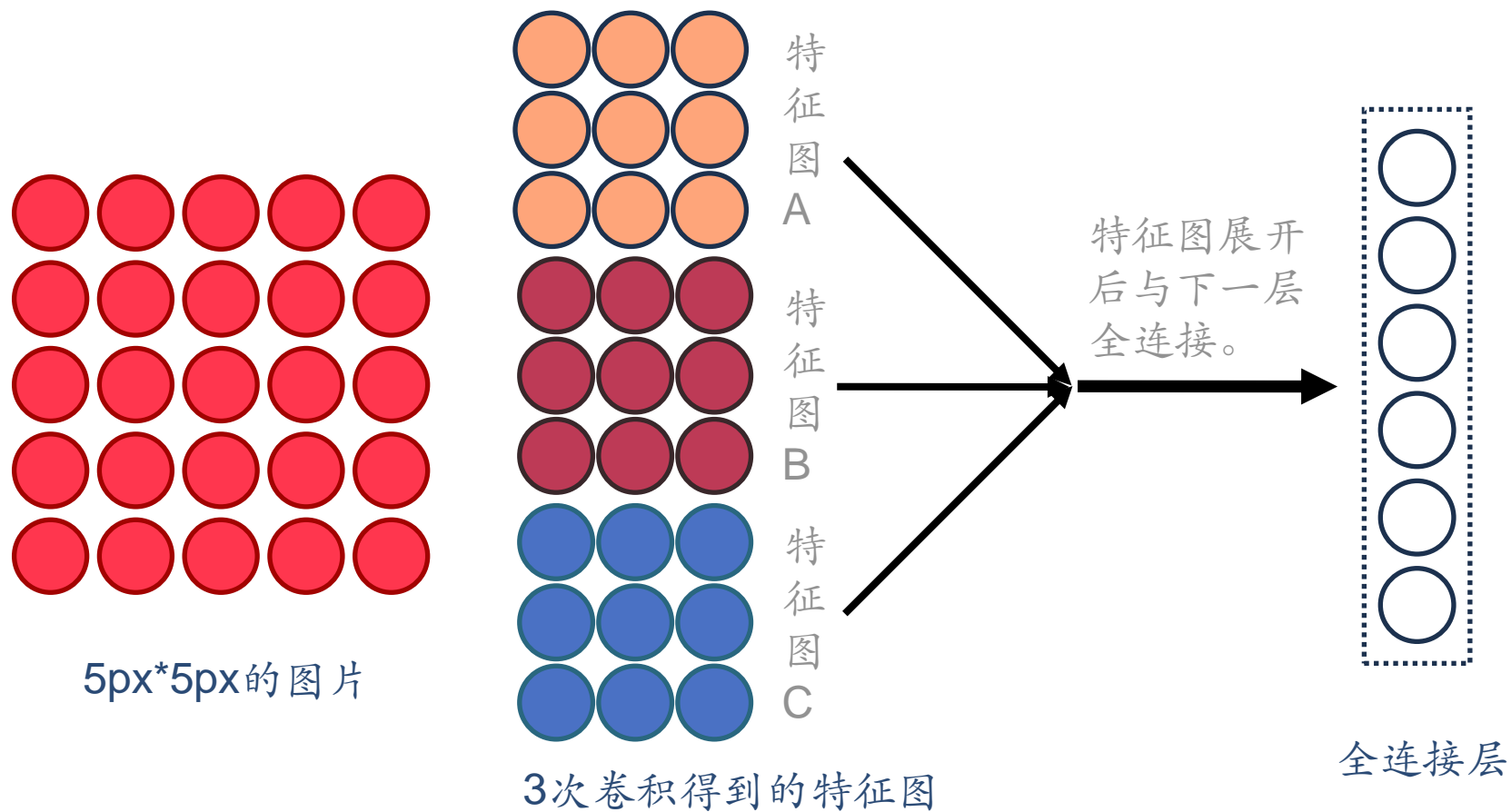
卷积就是利用特征（卷积核）进行局部特征提取，并得到特征分布图（特征图）的过程。



## 4.4 池化





- 
- ▶ 根据我们对全连接神经网络做数字图像处理相关任务的分析，思考：
  - ▶ 1. 使用卷积提取到的特征图是否有助于处理数字图像相关任务？
  - ▶ 2. 如果将特征图数据送入全连接神经网络是否可行？

# 特征图作为输入

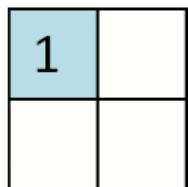
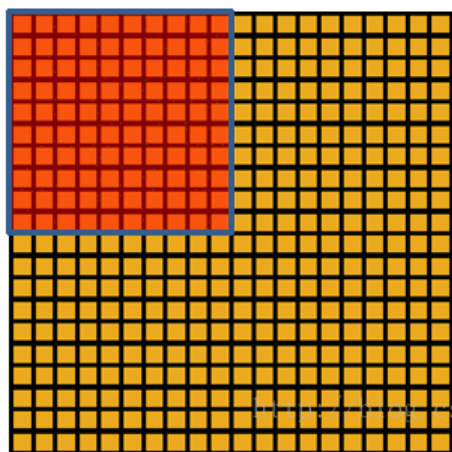


# 特征图作为模型输入的优缺点

---

- ▶  提取到了局部特征作为输出。
- ▶  卷积使得图像特征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



<https://www.zybang.com/question/8f11e0e0211>

Convolved  
feature

Pooled  
feature

滑动窗大小：一般取 $2 \times 2$ 。

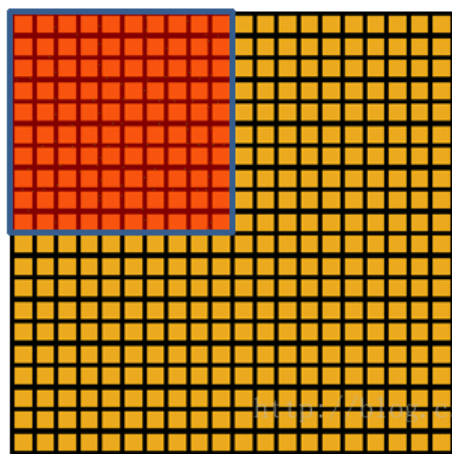
滑动步幅：通常等于滑动窗大小。

最大池化：取一个区域的最大值。

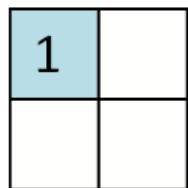
平均池化：取一个区域的平均值。



# 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：使得图像特征获得了一定的平移不变性；使得特征图的规模减小。

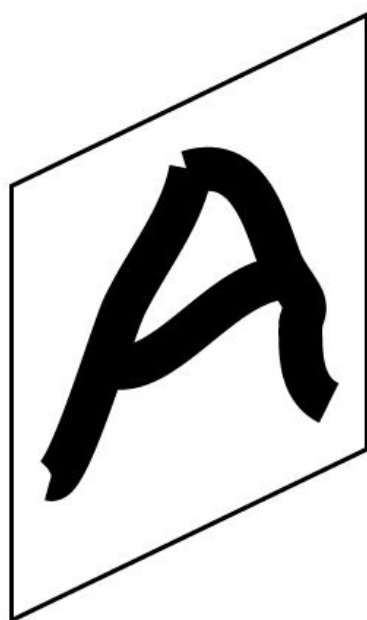
---

## 小练习

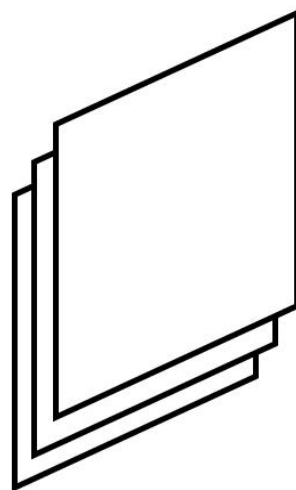
1. 对  $6 \times 6$  大小的特征图使用  $3 \times 3$  大小的滑动窗做池化得到的新特征图大小是多少？
2. 对  $10\text{px} \times 10\text{px}$  大小的图片使用 5 个  $3 \times 3$  大小的卷积核做卷积得到的特征图，再以  $4 \times 4$  的滑动窗做池化，得到了几张特征图？每张特征图的大小是多少？

## 5. 卷积与池化的 组合

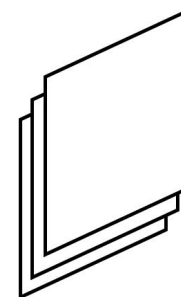
# 卷积与池化组合



28\*28像素的图像



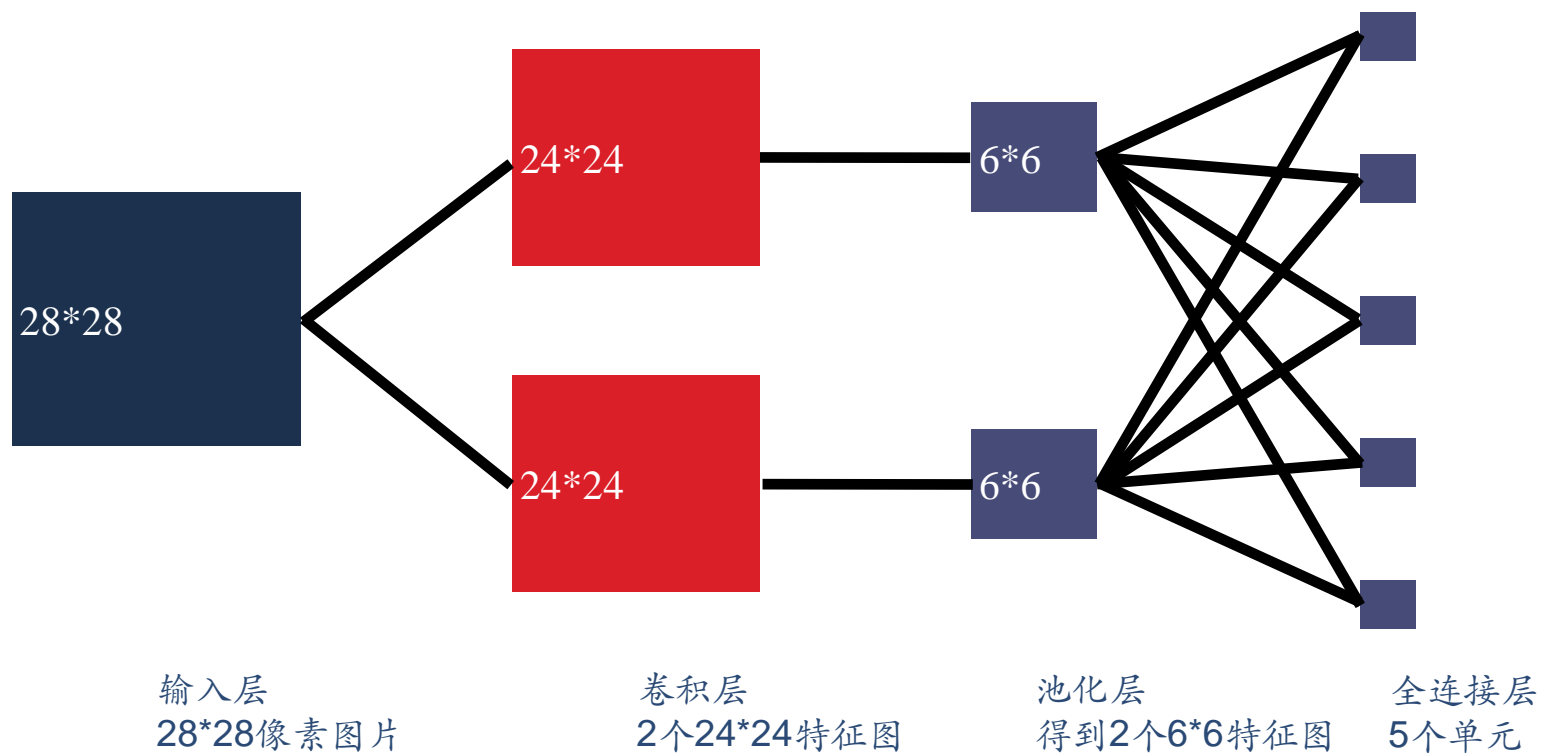
3个由5\*5卷积核得到的  
特征图：3\*24\*24



3个2\*2池化后的特征  
图：3\*12\*12

通过卷积与池化，可以有效的提取数据中的特征，并降低特征图维度。

# 简单的卷积神经网络结构图



# 卷积神经网络例子: LeNet

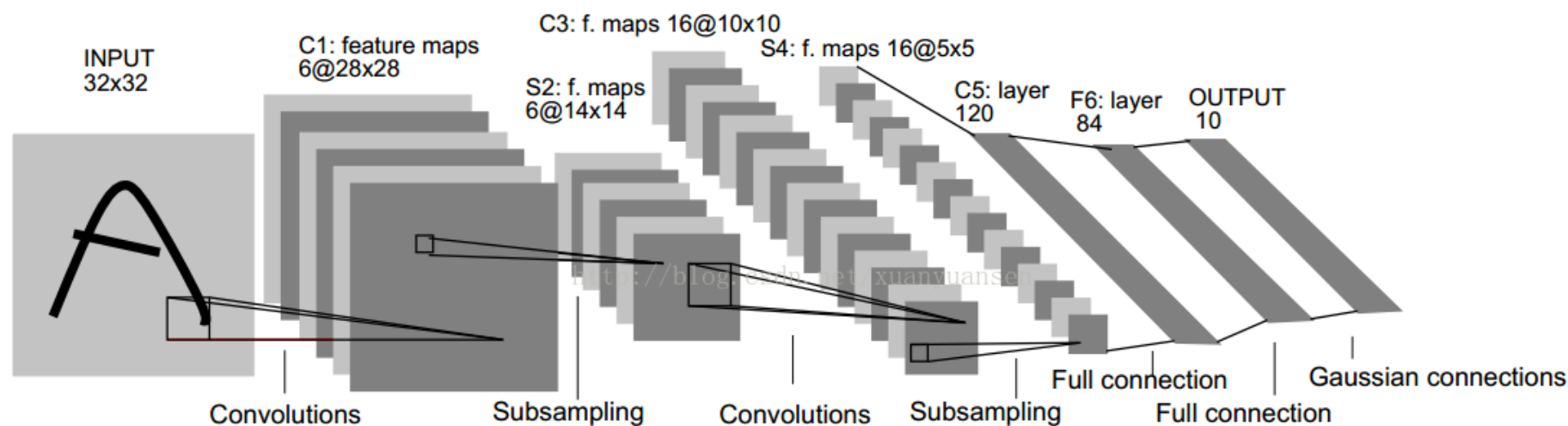
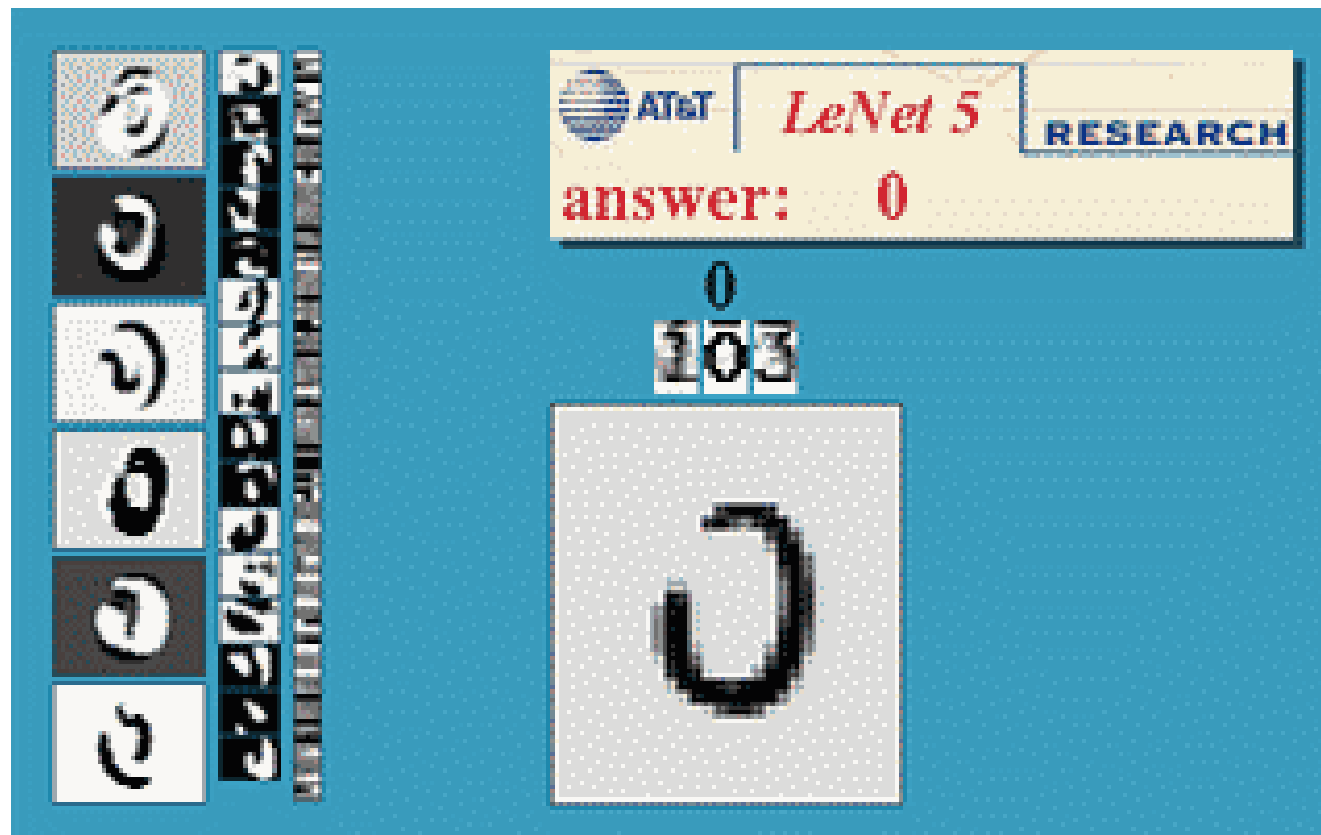


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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



# 卷积神经网络:LeNet



LeNet曾是上世界90年代中后期美国各大银行、证券公司等最常用的识别票据手写体的算法。

# 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