



南京大學
NANJING UNIVERSITY

人工智能导论

卷积神经网络 (Convolutional Neural Network)

郭兰哲

南京大学 智能科学与技术学院

<https://www.lamda.nju.edu.cn/guolz/IntroAI/fall2025/index.html>

Email: guolz@nju.edu.cn

大纲

- 神经元模型到前馈神经网络
- 参数优化：BP算法
- 深度学习
- 计算机视觉与卷积神经网络
- 自然语言处理与循环神经网络
- 多模态学习

计算机视觉(Computer Vision)

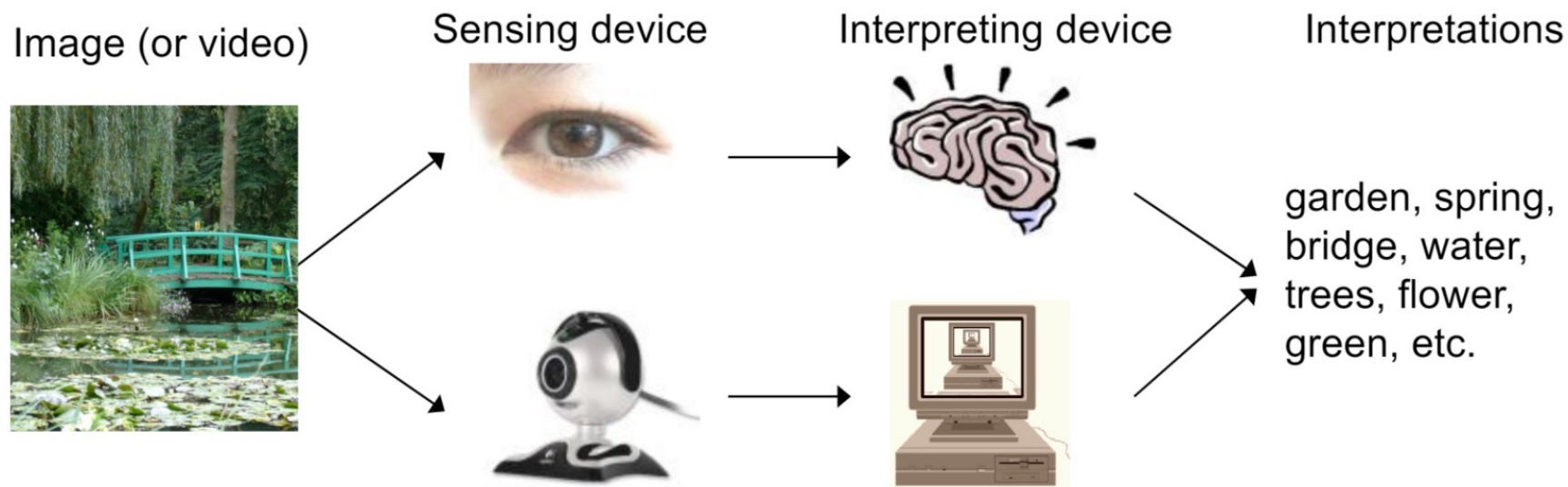
什么是计算机视觉

计算机视觉有哪些任务

如何用人工智能技术解决计算机视觉任务

计算机视觉(Computer Vision)

计算机视觉是一门研究如何对 **图像** 或者 **视频** 进行高层理解的交叉学科

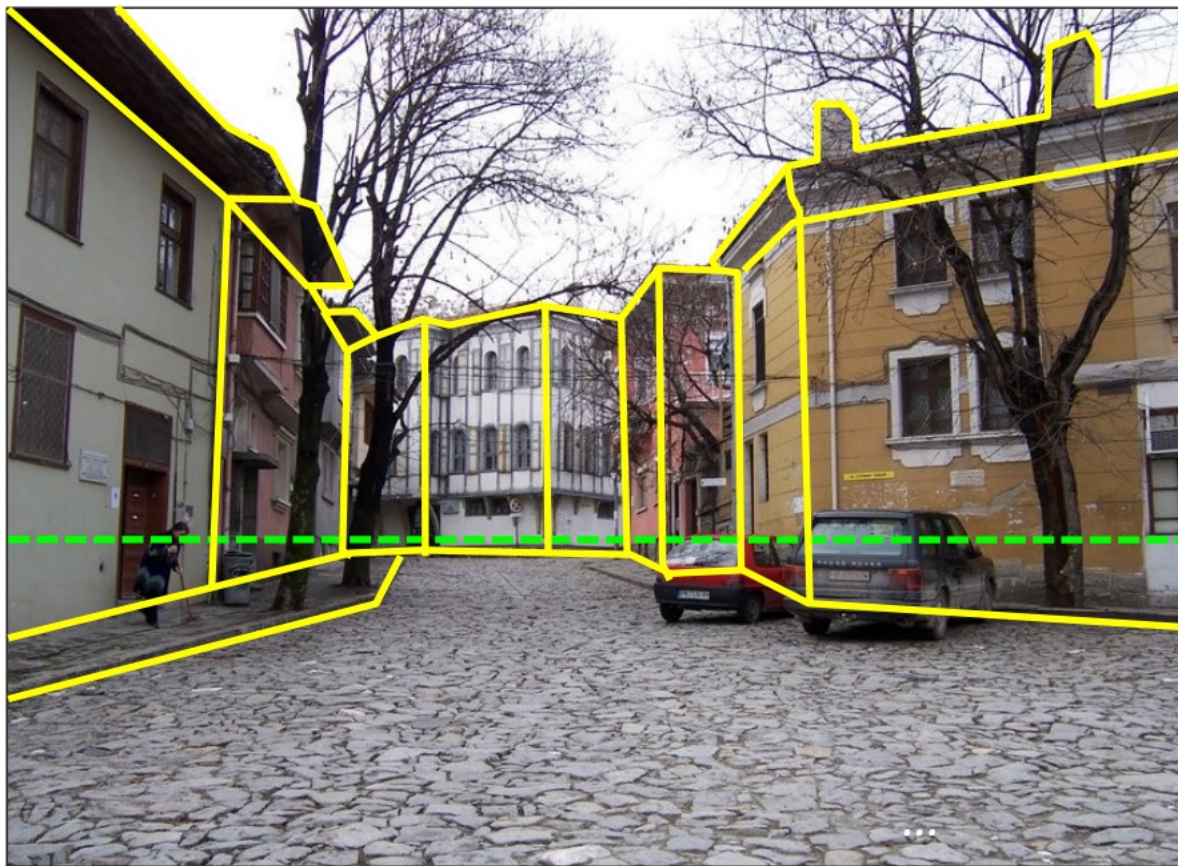


人工智能视角：赋予机器“看”的智能，即，用机器自动实现人类视觉系统的功能，图像或视频的获取、处理、分析和理解等诸多任务

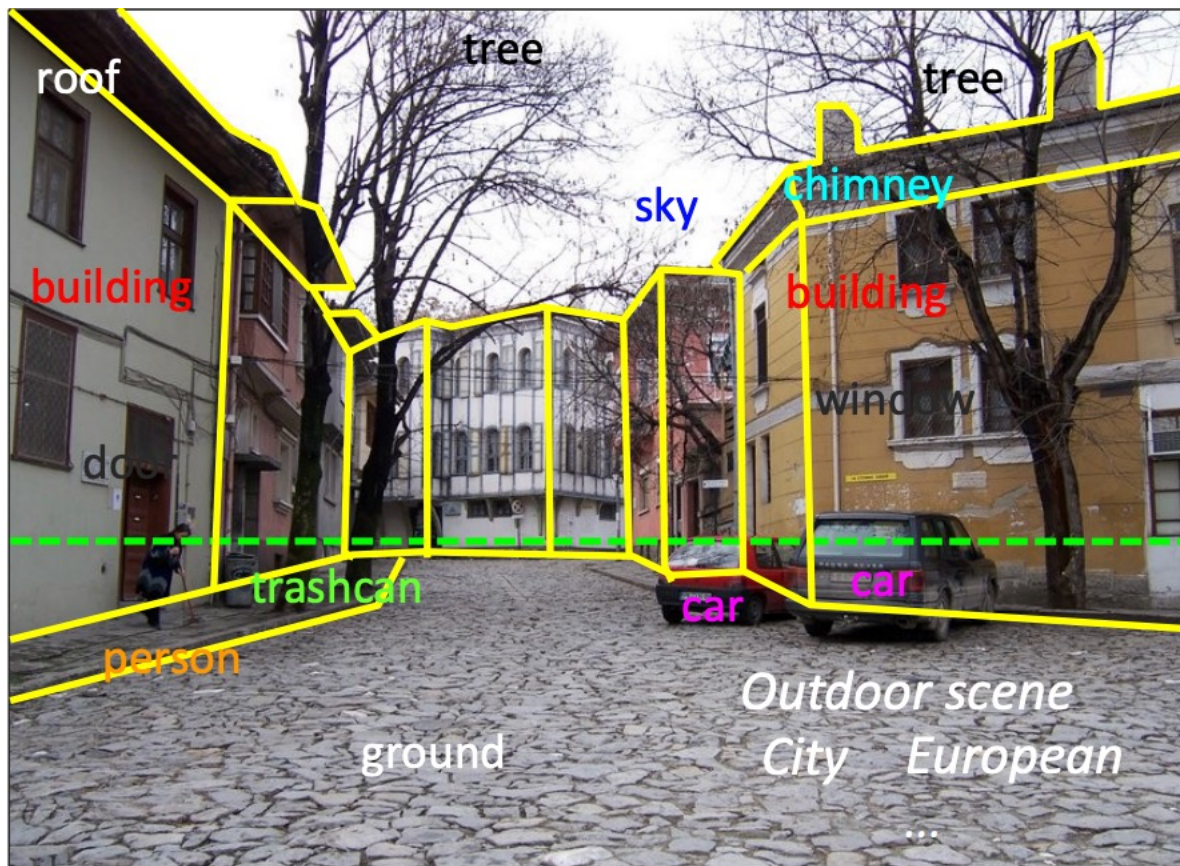
计算机视觉(Computer Vision)



计算机视觉(Computer Vision)

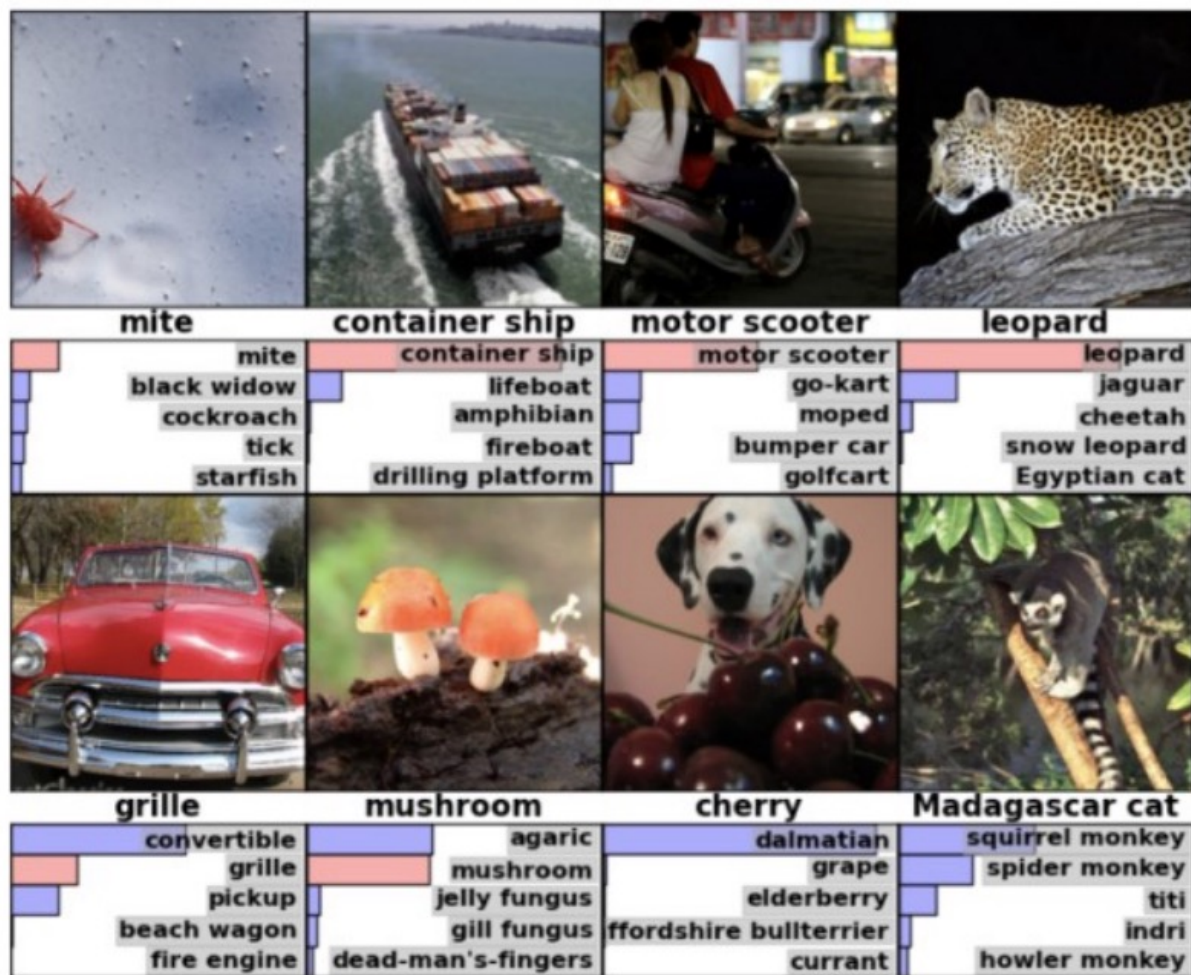


计算机视觉(Computer Vision)



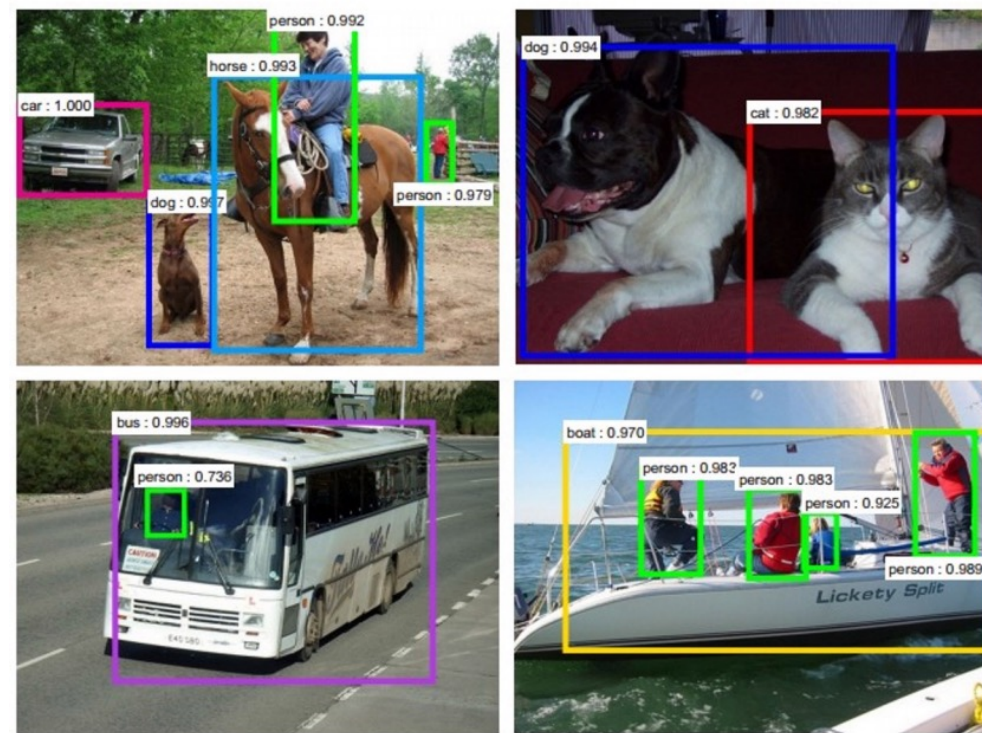
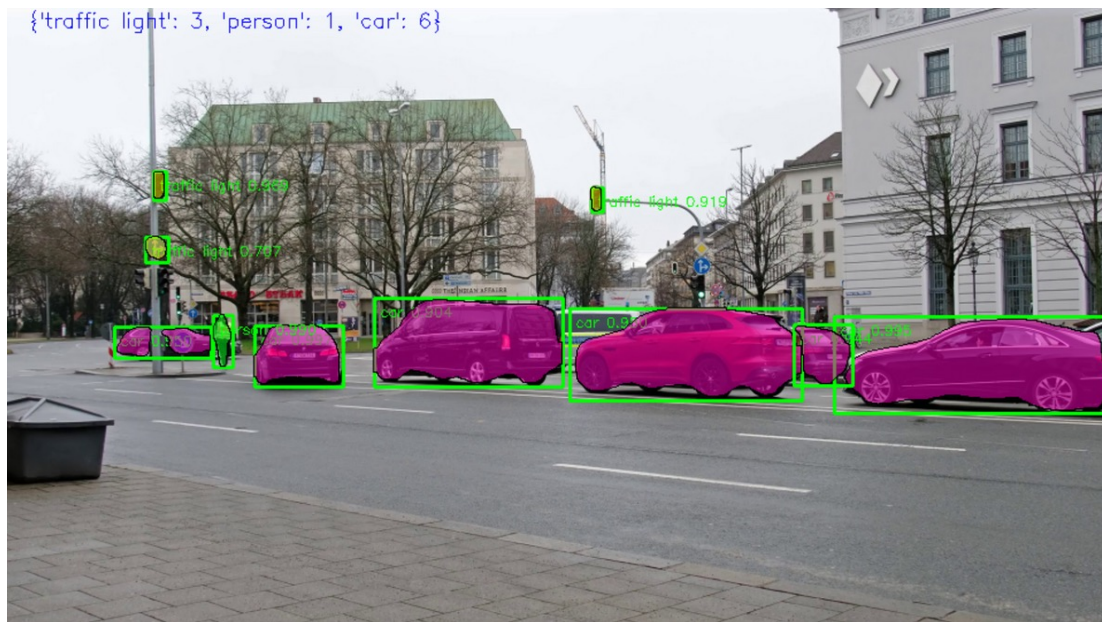
图像分类(Image Classification)

图像分类：将图片划分到某个特定的类别中



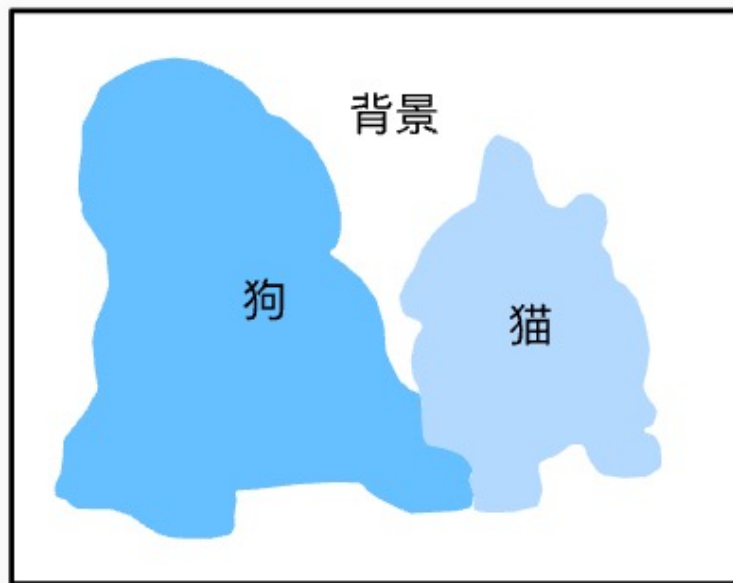
目标检测(Object Detection)

目标检测：检测图像中某个类别的物体的任务，一般来说，就是在图像中寻找我们感兴趣的目标，并给出其位置



语义分割(Semantic Segmentation)

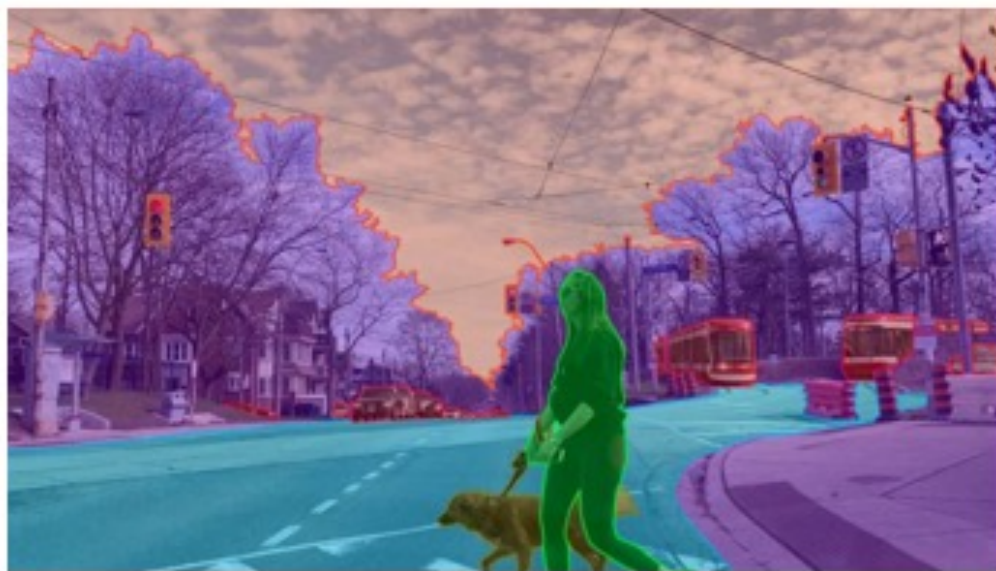
语义分割：关注如何将图像分割成不同类别的区域，与目标检测不同，语义分割可以识别并理解图像中每个像素的内容，其语义区域的标注和预测是像素级的



实例分割(Instance Segmentation)

实例分割：检测和划定出图像中出现的每个不同的感兴趣的对象，研究如何识别图像中各个目标实例的像素级区域。

与语义分割不同，实例分割**不仅需要区分语义，还要区分不同的目标实例**



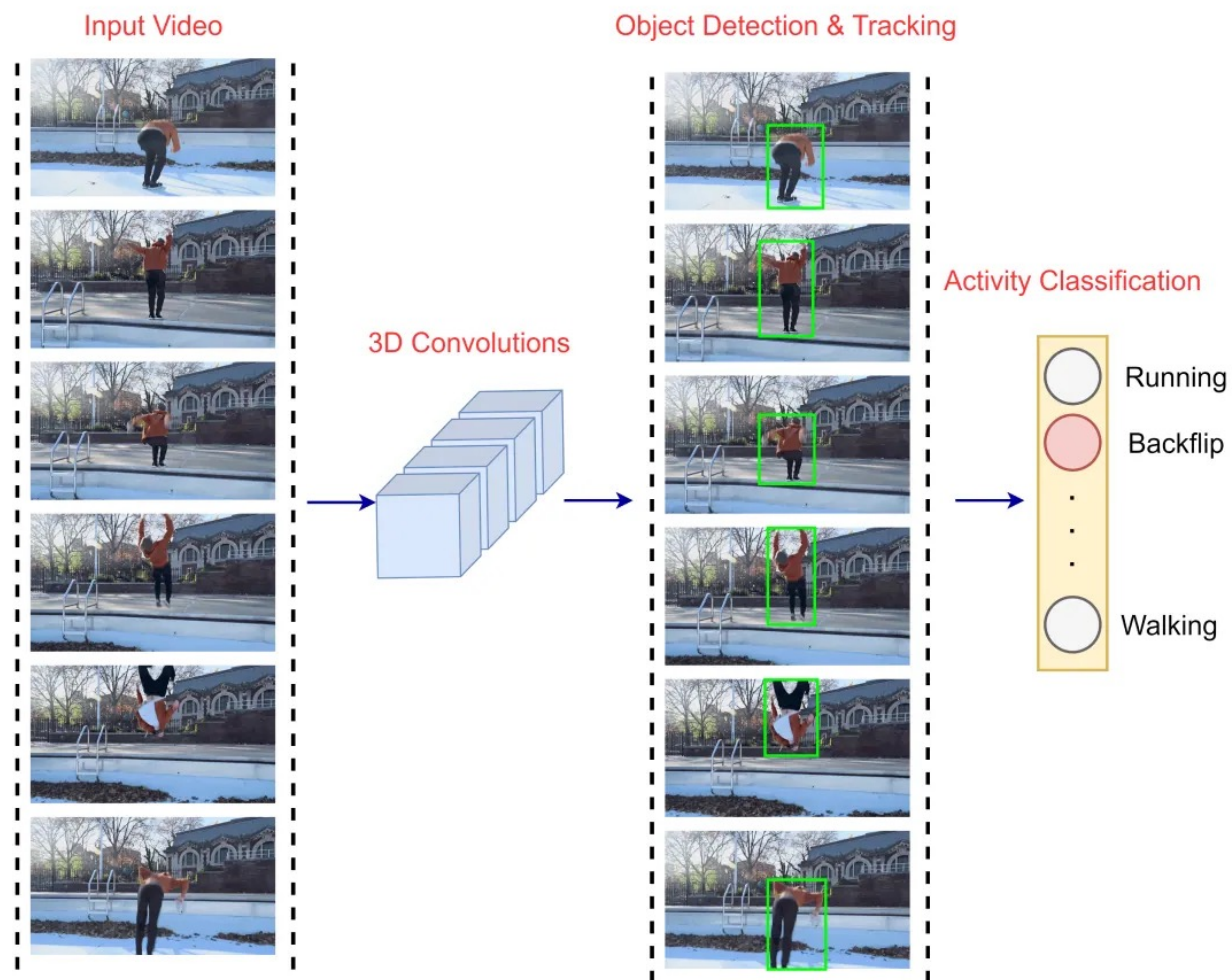
语义分割



实例分割

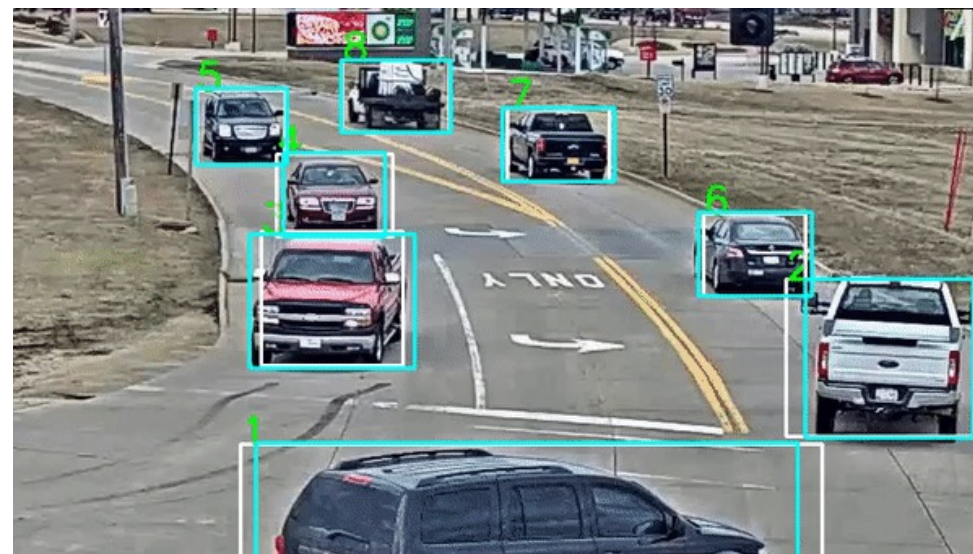
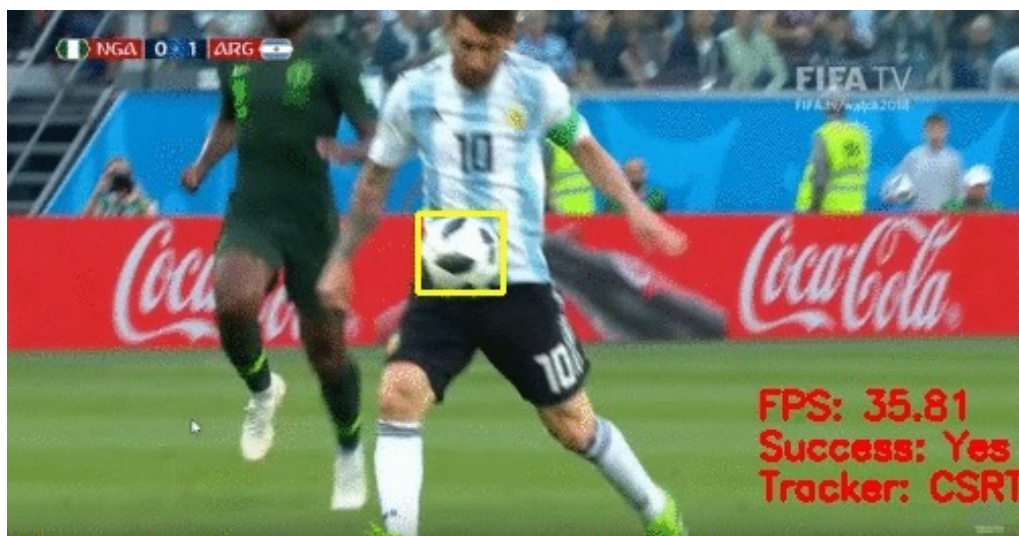
视频分类(Video Classification)

视频分类是给定一个视频，为其划分到指定的类别中



目标跟踪(Object Tracking)

目标跟踪：根据视频序列第一帧中目标的初始状态（中心位置和比例），自动获得该物体在后续视频帧中的状态



视觉任务的挑战



0	120	120	120	120	120	120	120	121	121	121	11	121	121	122	122	121	121	121	121	122	123	123
1	82	87	88	88	88	88	89	89	88	89	89	89	89	89	90	90	90	90	90	89	89	89
2	95	94	94	95	95	95	95	94	95	95	95	95	95	95	96	96	96	96	96	96	96	96
3	98	98	98	98	98	99	99	99	99	99	100	100	99	100	101	101	101	100	100	100	100	100
4	99	99	100	100	100	100	99	100	101	101	101	102	101	102	102	102	102	102	102	102	102	102
5	91	95	94	96	96	94	95	95	95	95	95	94	95	95	95	95	95	96	96	96	97	97
6	101	101	102	102	102	102	102	102	102	102	103	103	103	103	103	103	103	103	103	103	103	103
7	92	97	96	97	97	97	97	97	97	97	97	97	97	97	97	97	98	98	97	98	98	98
8	103	102	103	102	103	103	103	103	103	104	104	104	104	104	104	104	104	104	104	104	105	105
9	93	97	98	98	98	98	98	98	98	98	99	99	98	98	98	98	98	99	99	99	99	99
0	103	103	104	104	104	104	104	104	104	105	106	106	106	106	106	106	106	106	105	106	105	105
1	95	99	98	99	99	99	99	99	100	100	100	99	101	101	100	100	101	100	100	100	100	100
2	104	104	105	106	105	105	106	107	106	107	107	107	107	107	107	107	107	108	108	107	108	108
3	96	100	100	100	102	101	102	101	102	102	102	102	102	102	102	102	102	102	102	103	102	103
4	107	107	107	107	107	107	107	107	108	108	108	108	109	108	109	109	108	109	109	109	109	109
5	98	102	102	102	102	103	103	103	103	103	103	103	103	103	103	103	103	103	104	104	103	104
6	108	108	108	108	109	108	108	109	109	109	109	109	109	109	109	110	109	110	111	111	111	111
7	100	103	103	103	103	104	104	104	104	104	104	104	104	104	104	104	104	105	105	106	105	106
8	109	109	109	109	110	109	110	110	110	110	110	111	111	111	112	112	112	112	112	113	113	113
9	101	104	104	105	105	106	106	105	105	105	106	105	106	107	107	106	107	107	107	107	107	107
0	111	111	112	111	112	112	112	112	112	112	113	113	112	112	112	113	113	113	114	113	114	113
1	102	106	107	107	107	106	107	107	106	107	108	107	108	108	108	108	108	108	108	108	108	108
2	113	113	113	113	113	113	113	113	114	114	114	113	114	114	114	114	114	115	115	115	115	115
3	105	108	108	108	108	108	108	108	109	109	109	109	109	109	110	110	109	109	109	109	109	109
4	114	114	114	114	114	114	114	114	115	116	115	115	115	116	117	116	117	117	117	117	117	117
5	106	109	109	109	109	109	110	110	110	111	111	110	111	111	111	112	112	111	111	111	112	113
6	116	116	116	116	116	116	116	117	117	117	117	117	117	118	118	118	118	118	118	118	118	118
7	108	111	110	110	110	111	111	112	111	112	112	113	113	113	113	113	113	113	113	113	113	113
8	117	117	118	118	118	118	118	118	118	119	118	120	120	119	119	119	119	119	119	119	119	119
9	110	113	111	113	113	113	114	114	114	114	114	114	114	114	114	114	115	115	115	115	115	115
0	119	119	119	119	119	119	119	120	120	120	119	129	129	120	121	121	121	121	120	121	122	122
1	111	114	114	114	115	115	115	115	115	115	116	116	117	116	117	116	117	117	117	116	117	117
2	121	120	120	120	120	121	120	121	122	122	121	126	130	122	122	122	123	123	123	123	123	123
3	112	116	116	117	117	117	117	117	118	118	118	118	118	118	118	118	118	118	118	118	118	118
4	122	122	123	123	123	123	123	123	124	124	124	123	127	124	124	124	125	125	125	125	124	124
5	114	117	118	118	118	119	119	119	122	120	119	119	120	119	120	119	119	120	121	120	120	120
6	124	124	124	124	124	125	125	125	126	125	125	118	127	125	125	126	127	127	127	127	127	127
7	116	119	119	120	120	120	121	119	137	123	122	121	122	121	122	121	121	122	122	122	122	122
8	125	125	126	126	127	127	127	127	127	127	126	121	129	128	128	128	129	129	128	129	129	129
9	117	120	121	122	122	123	123	121	145	128	122	123	123	124	124	124	124	124	124	124	124	124

- 数字图像由一个个像素(pixel)组成, 每个像素的亮度、颜色等属性在计算机内表示为一个或多个数字
- 如果是黑白图像(灰度图像), 每个像素由一个字节表示, 0-255的数值表示亮度, 如果是彩色图像, 通常用RGB三个字节表示

视觉任务的挑战



Illumination



Object pose



Clutter



Occlusions



**Intra-class
appearance**



Viewpoint

视觉任务的挑战:intra-class variation



机器学习解决视觉任务

120 120 120 120 120 120 120 120 121 121 121 121 121 122 122 121 121 121 122 123 1
82 87 88 88 88 88 89 89 88 89 89 89 89 90 90 90 90 90 89 89
93 94 94 95 95 95 95 95 95 95 95 95 96 96 96 96 96 96 96
89 92 92 92 92 92 92 93 93 93 93 93 93 93 93 93 93 93 93
98 98 98 98 98 99 99 99 99 100 100 99 100 101 101 101 100 100 101 101
90 93 94 94 94 94 93 93 94 93 94 94 94 94 94 94 95 95 96
99 99 100 100 100 100 99 100 101 101 101 102 101 102 102 102 102 102 102 102
91 93 94 96 96 94 95 95 95 95 95 94 95 95 95 95 96 96 96 97
101 101 102 102 101 102 102 102 102 102 103 103 103 103 103 103 103 103 103 103
92 97 96 97 97 97 97 97 97 97 97 97 97 97 97 97 98 98 97 98 98
103 103 103 103 103 103 103 103 104 104 104 104 104 104 104 104 104 104 104 105 1
93 97 98 98 98 98 98 98 98 99 99 98 98 98 98 99 99 99 99 99
103 103 104 104 104 104 104 104 104 105 106 106 106 106 106 105 106 105 105 105 1
95 99 98 99 99 99 99 99 100 100 100 99 101 101 100 100 100 100 100 100 100 1
104 104 105 106 105 105 105 106 107 106 107 107 107 107 107 107 108 108 107 108 1
96 100 100 100 102 101 102 101 102 102 102 102 102 102 102 102 103 102 103 1
107 107 107 107 107 107 107 107 108 108 108 109 108 109 109 108 109 109 109 109 109
98 102 102 102 102 103 103 103 103 103 103 103 103 103 103 103 103 104 104 103 1
108 108 108 108 109 108 108 109 109 109 109 109 109 109 110 109 110 111 111 111 111
100 103 103 103 103 104 104 104 104 104 104 104 104 104 104 104 105 105 106 105 1
109 109 109 109 110 109 110 110 110 110 111 111 111 112 112 112 112 113 113 113
101 104 104 105 105 106 106 105 105 106 105 106 107 107 106 107 107 107 107 107 107
111 111 112 111 112 112 112 112 112 112 112 112 112 113 113 113 113 114 113 114 1
102 106 107 107 107 106 107 107 106 107 108 108 108 108 108 108 108 108 108 108 108
113 113 113 113 113 113 113 114 114 114 114 114 114 114 114 115 115 115 115 115 115
105 108 108 108 108 108 108 109 109 109 109 109 109 110 109 109 109 109 109 109 1
114 114 114 114 114 114 114 114 114 115 116 115 115 116 117 116 117 117 117 117 117
106 109 109 109 109 109 110 110 110 110 111 110 111 111 112 112 111 111 111 112 1
116 116 116 116 116 116 116 117 117 117 117 117 118 118 118 118 118 118 118 118 118
108 111 110 110 110 111 111 112 111 112 112 113 113 113 113 113 113 113 113 113 113
117 117 118 118 118 118 118 118 119 119 119 120 119 119 119 119 119 119 119 119 119
110 113 113 113 113 113 114 114 114 114 114 114 114 114 114 114 115 115 115 114 115 115
119 119 119 119 119 119 119 120 120 120 119 119 120 120 121 121 121 120 120 121 121
111 114 114 114 115 115 115 115 115 115 116 116 117 116 117 117 117 117 117 116 117 117
121 120 120 120 121 120 121 122 122 121 126 130 122 122 122 123 123 123 123 123 123
112 116 116 117 117 117 117 118 118 118 118 118 118 118 118 118 118 118 118 118 118
122 122 123 123 123 123 123 123 123 124 124 123 127 124 124 124 125 125 125 125 124 124
114 117 118 118 118 119 119 120 119 120 119 120 119 120 119 120 121 120 120 120 120
124 124 124 124 125 125 125 126 125 125 118 127 125 125 126 127 127 127 127 127 127
116 119 119 120 120 120 121 119 127 123 122 121 122 121 122 121 122 122 122 122 122
125 125 126 126 127 127 127 127 127 128 128 128 128 128 128 128 128 128 128 128 128
117 120 121 122 122 123 123 121 125 128 127 123 123 124 124 124 124 124 124 124 124

X

$f_{\theta}(X)$

y

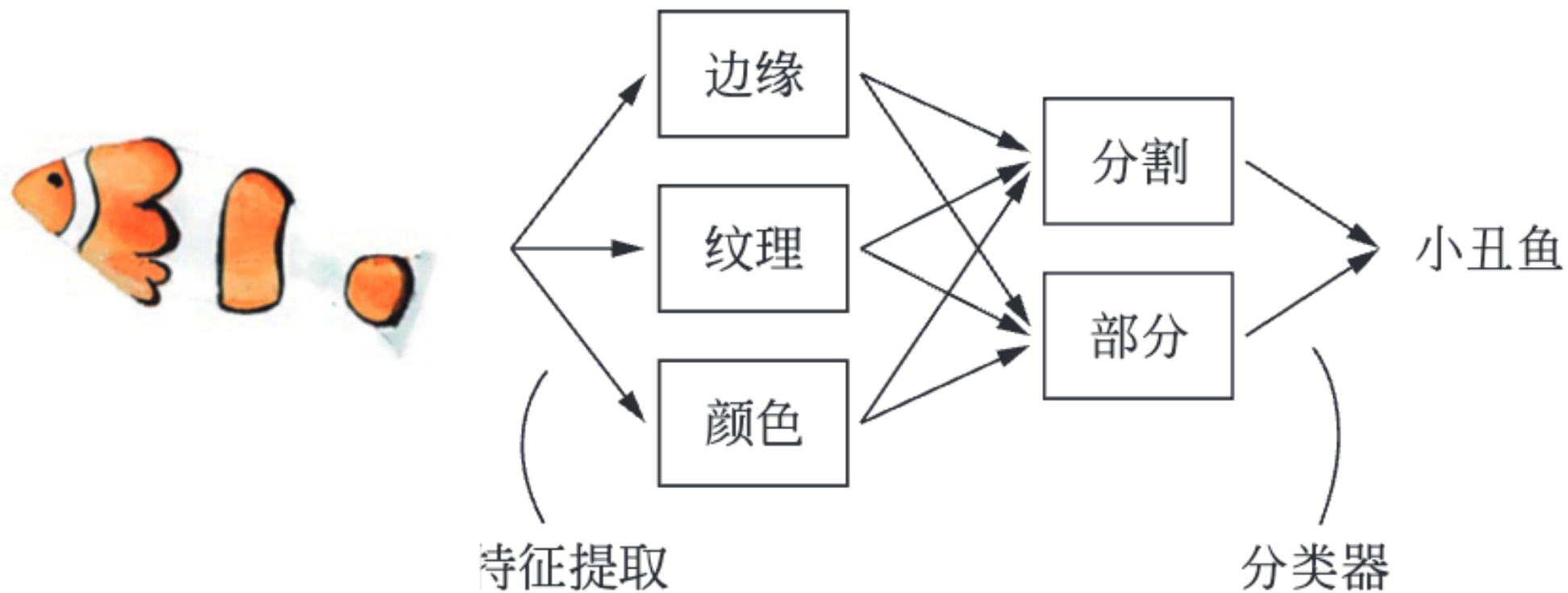
类别标签(如图像分类、物体识别等)

连续变量/向量/图像 (如目标检测、语义分割)

基于浅层模型的方法

- 步骤1：图像预处理：用于实现目标对齐、几何归一化、亮度或颜色校正等处理
- 步骤2：特征设计与提取：从预处理的图像中提取描述图像内容的特征
- 步骤3：特征汇聚或特征变换：对提取的特征进行统计或降维处理，从而得到更利于后续分类或回归任务的特征
- 步骤4：分类器或回归器的设计与训练：采用机器学习算法训练学习器，决策树、SVM、LR等

基于浅层模型的方法



图像分类的流程

特征设计与提取方法

- 人工设计特征本质上是一种**专家知识驱动**的方法，即研究者通过自身对所研究问题或目标的理解，设计某种流程来提取觉得“好”的特征
- 例如：人类识别早期，研究人员普遍认为应用**面部关键特征点的相对距离、角度或器官面积**等就可以区分人脸，但后来证明这些特征并不好
- 目前，多数人工设计的特征有两类，即**全局特征**和**局部特征**
 - 全局特征：建模图像中全部像素或多个不同区域像素蕴含的信息
 - 局部特征：从一个局部区域内的少量像素中提取信息

各种算子

Roberts算子

$$S_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, S_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Prewitt算子

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Sobel算子

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Canny算子

Laplacian算子

...

各种算子



原始图片



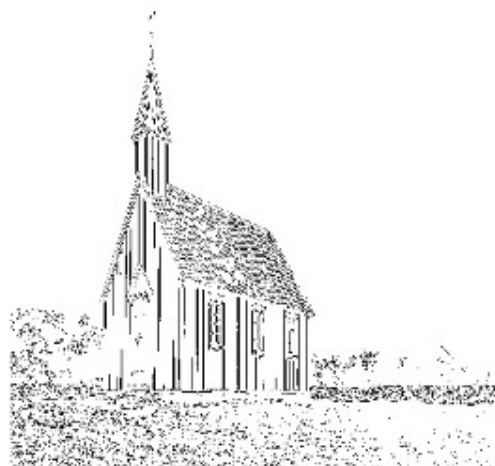
Roberts算子



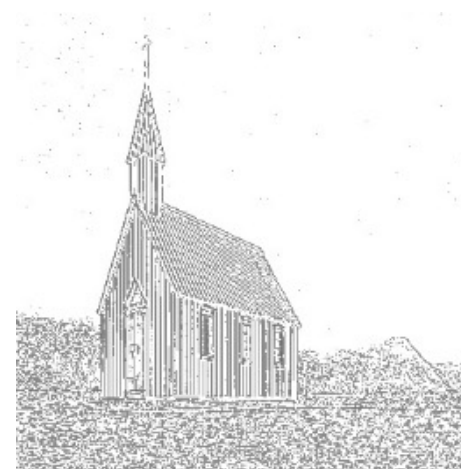
Prewitt算子



Sobel算子

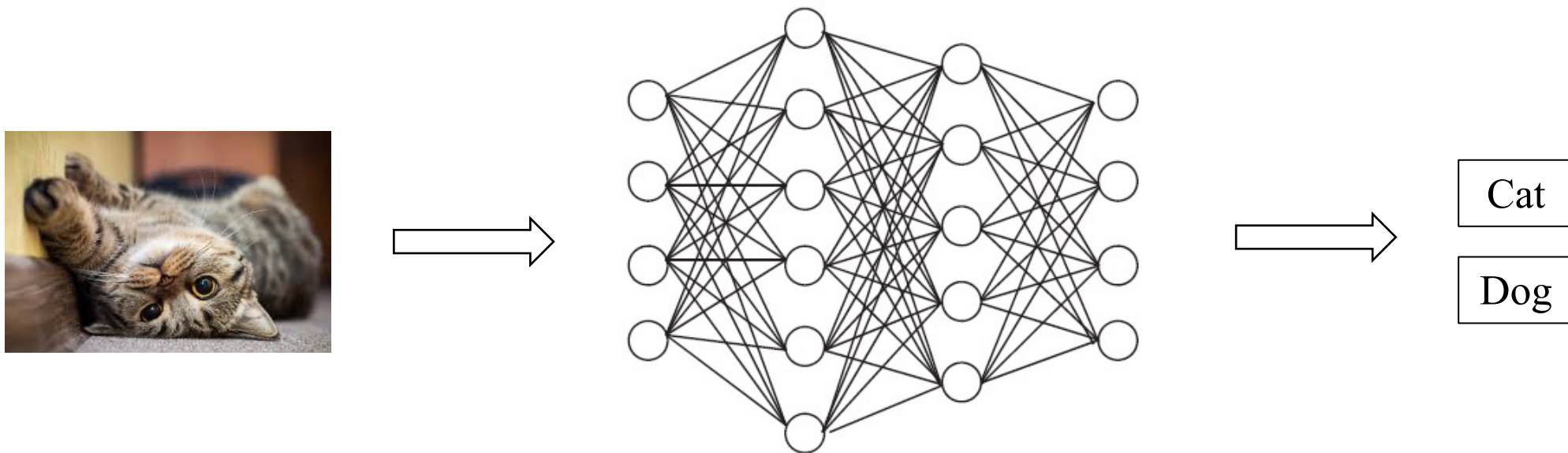


Canny算子



Laplacian算子

全连接前馈网络处理图像的问题

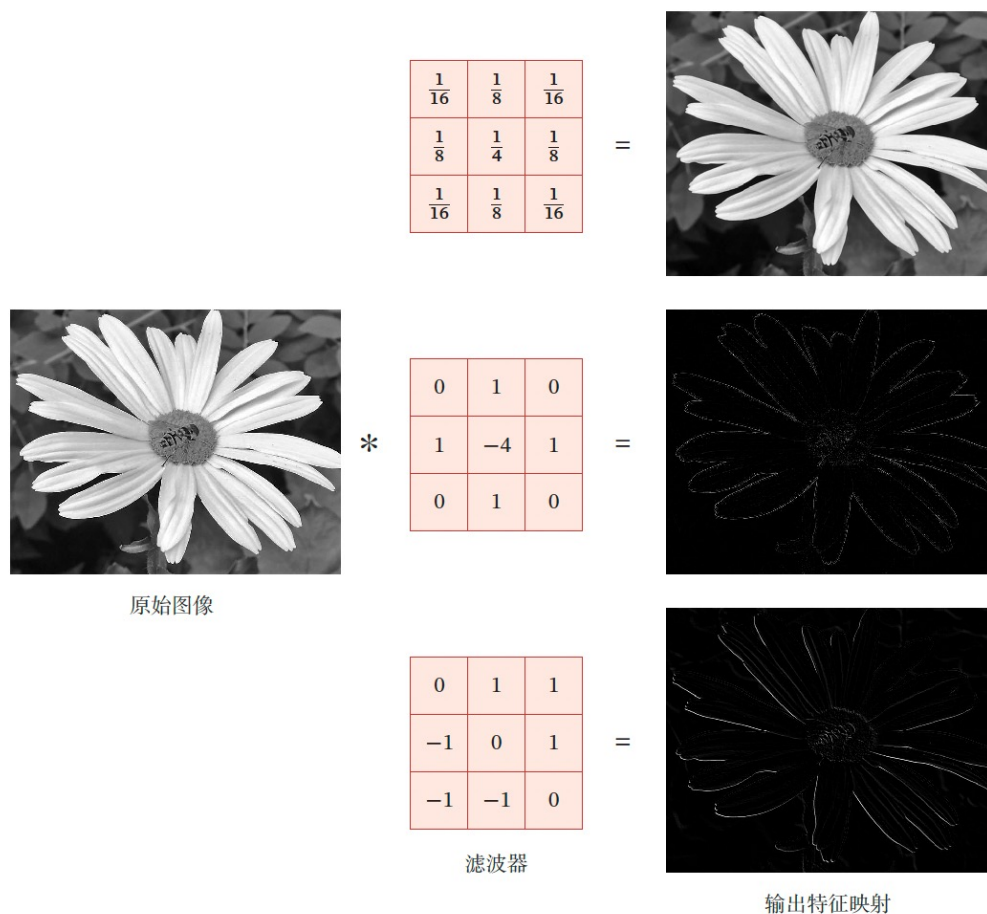


- 使用全连接前馈网络处理图像的问题：

- **参数太多**：如果输入数据为 1000×1000 的图像，需要转换为1,000,000维的向量，如果第一个隐藏层与输入维度一致，则输入层到第一个隐藏层的待训练参数为 10^{12} 个
- **局部不变性特征**：自然图像中的物体都具有局部不变形特征，比如尺度缩放、平移、旋转不影响语义，前馈网络很难提取局部不变性特征

卷积神经网络(Convolutional Neural Network, CNN)

- **卷积层**：每个卷积层包含多个特征映射, 每个特征映射通过一种**卷积滤波器**提取一种数据的特征 (**特征提取**)



卷积神经网络(Convolutional Neural Network, CNN)

- **卷积层**：每个卷积层包含多个特征映射，每个特征映射通过一种**卷积滤波器**提取一种数据的特征（**特征提取**）

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

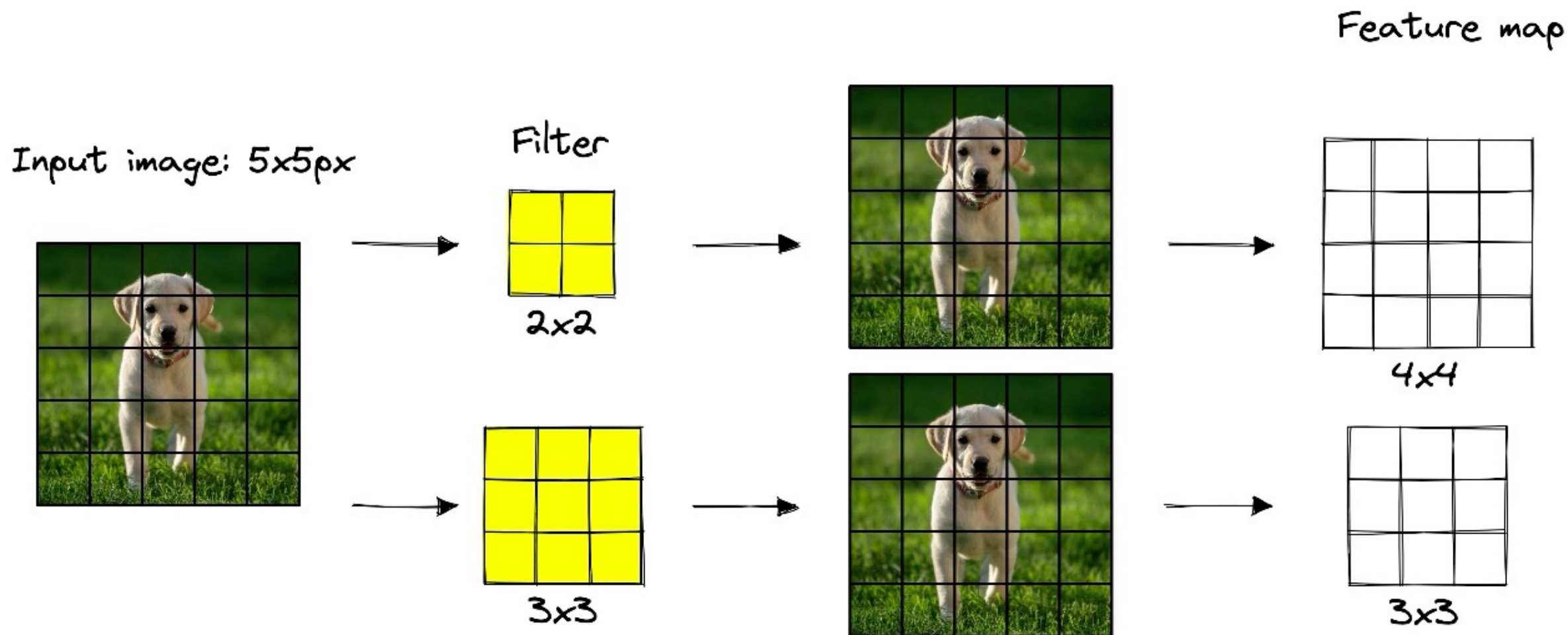
Image

4		

Convolved
Feature

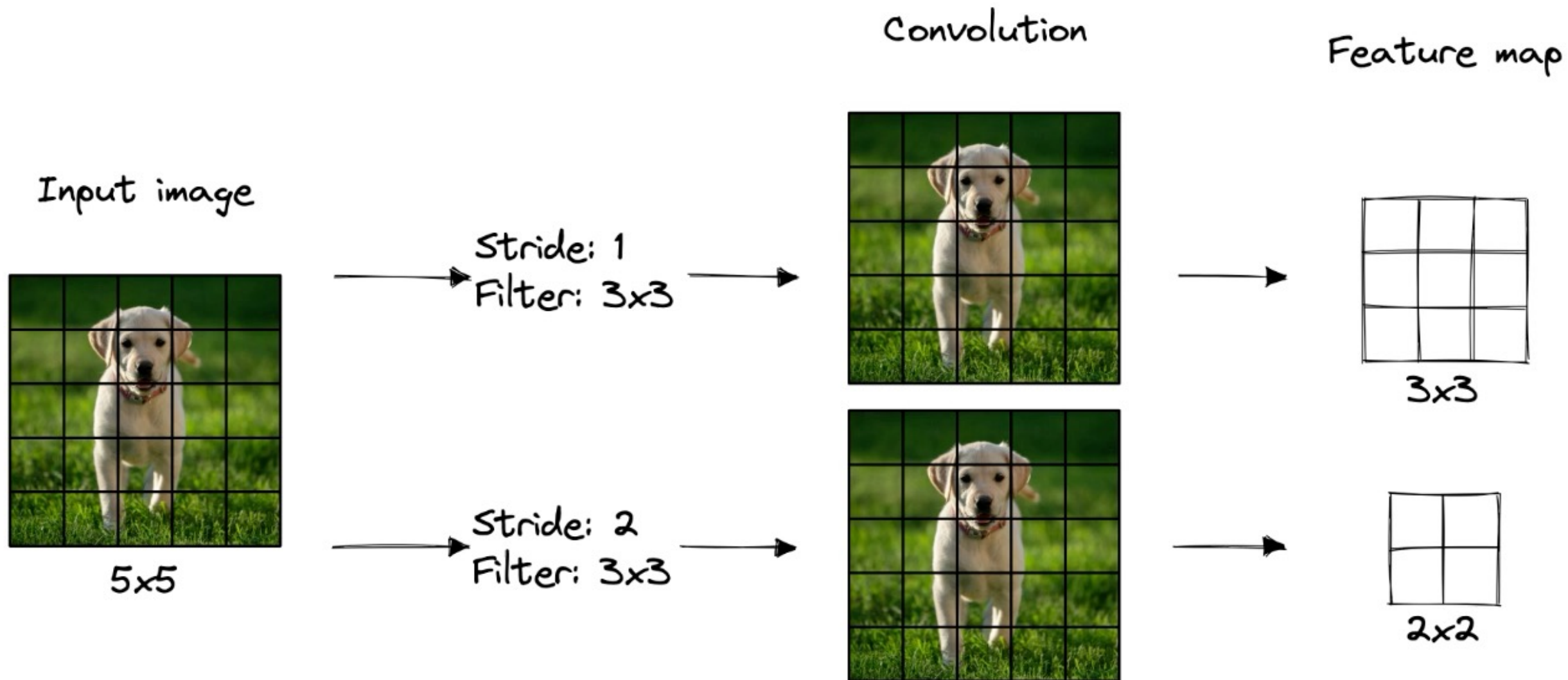
卷积神经网络(Convolutional Neural Network, CNN)

- **卷积层**：每个卷积层包含多个特征映射，每个特征映射通过一种卷积滤波器提取一种数据的特征（特征提取）



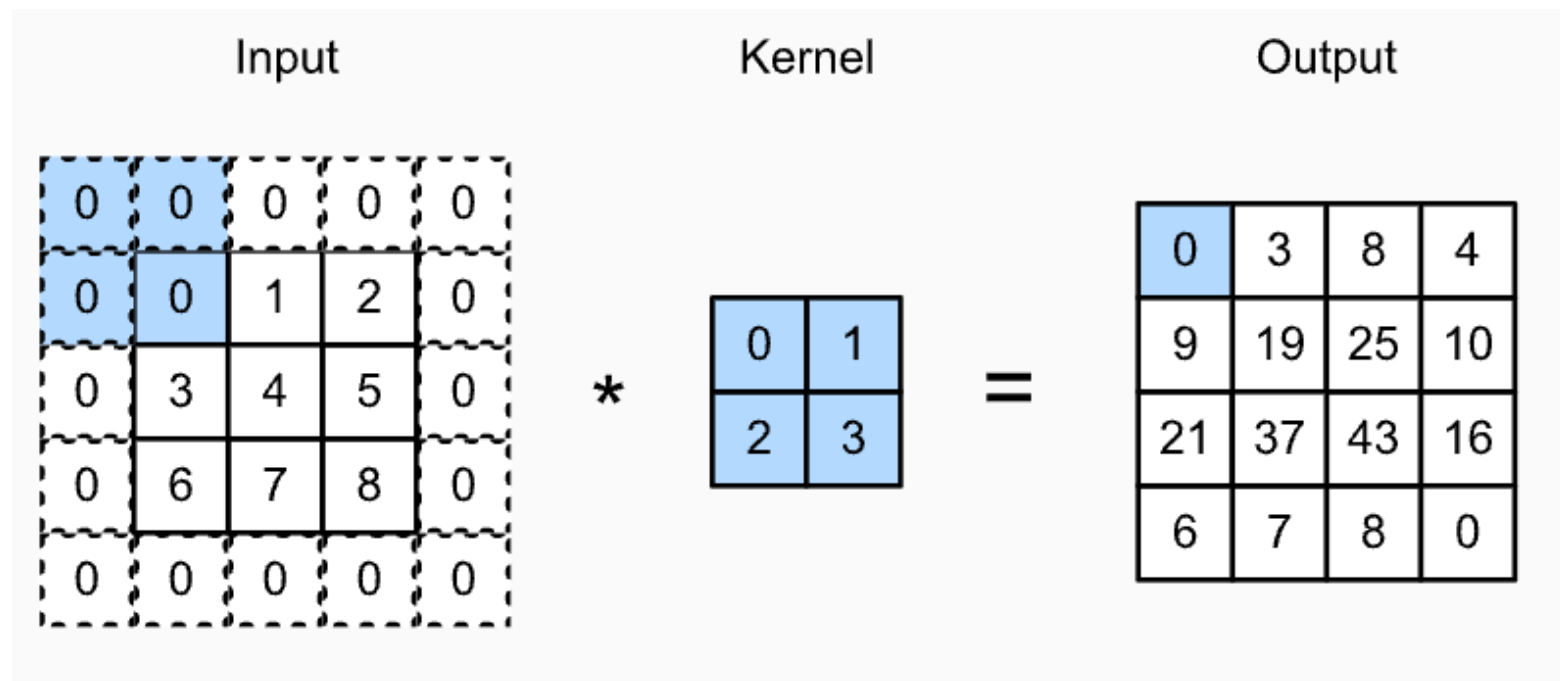
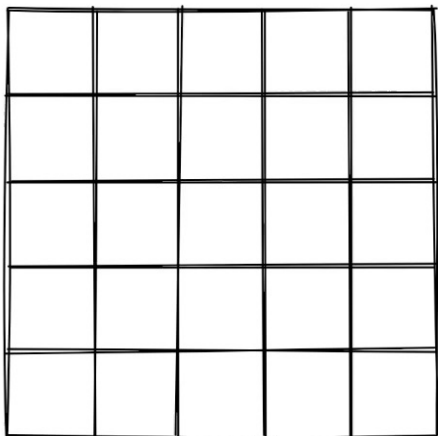
卷积神经网络(Convolutional Neural Network, CNN)

- 步长(Stride)



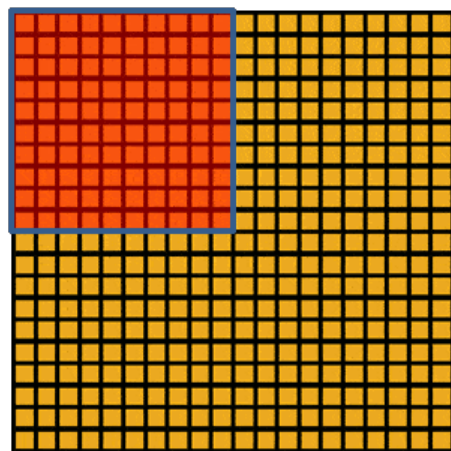
卷积神经网络(Convolutional Neural Network, CNN)

- 填充(Padding)

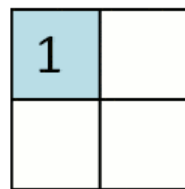


卷积神经网络(Convolutional Neural Network, CNN)

- **采样层**：亦称“池化(Pooling)层”，其作用是基于局部相关性原理进行亚采样，从而在减少数据量的同时保留有用信息（降低参数量级）



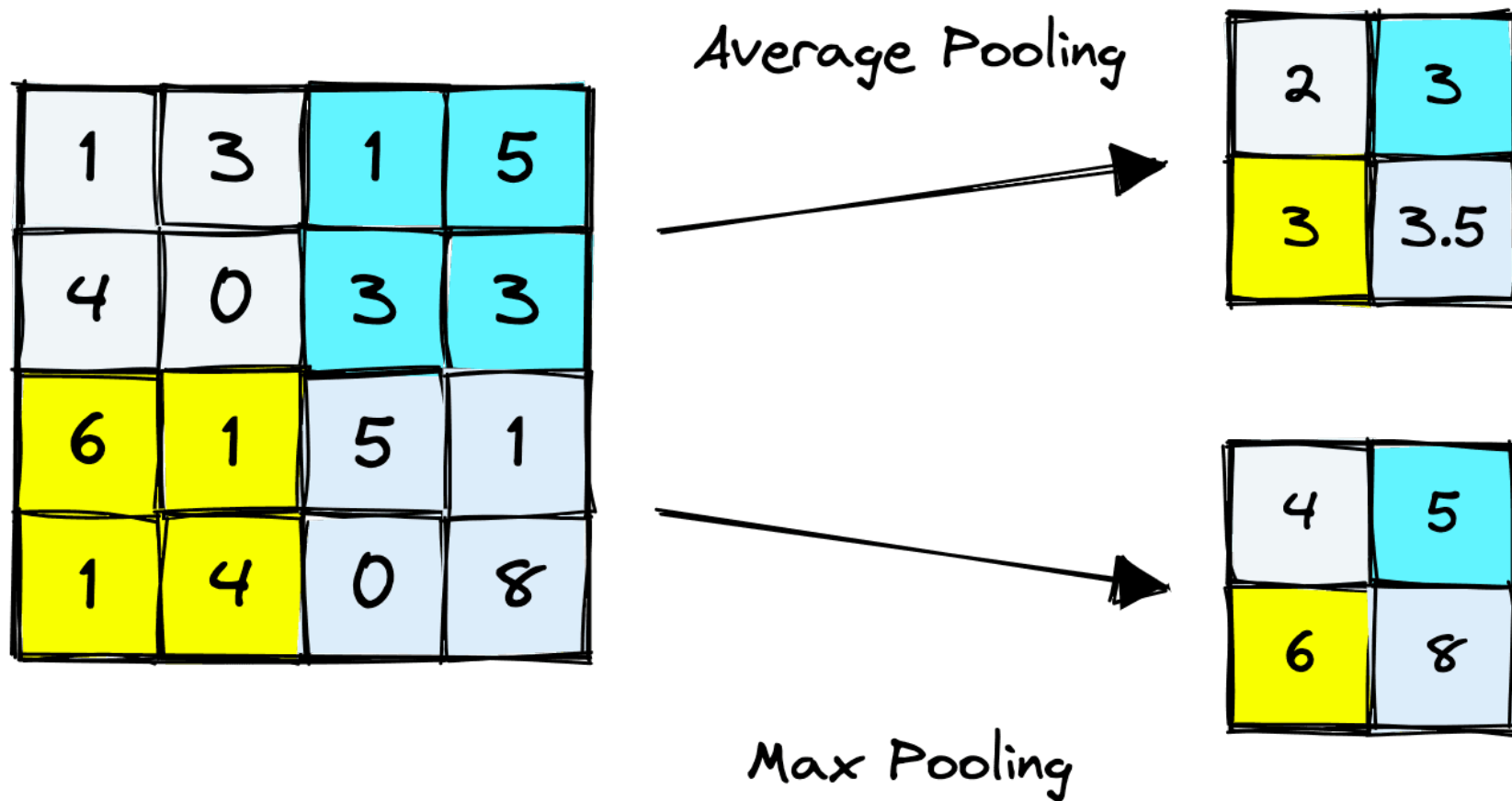
Convolved
feature



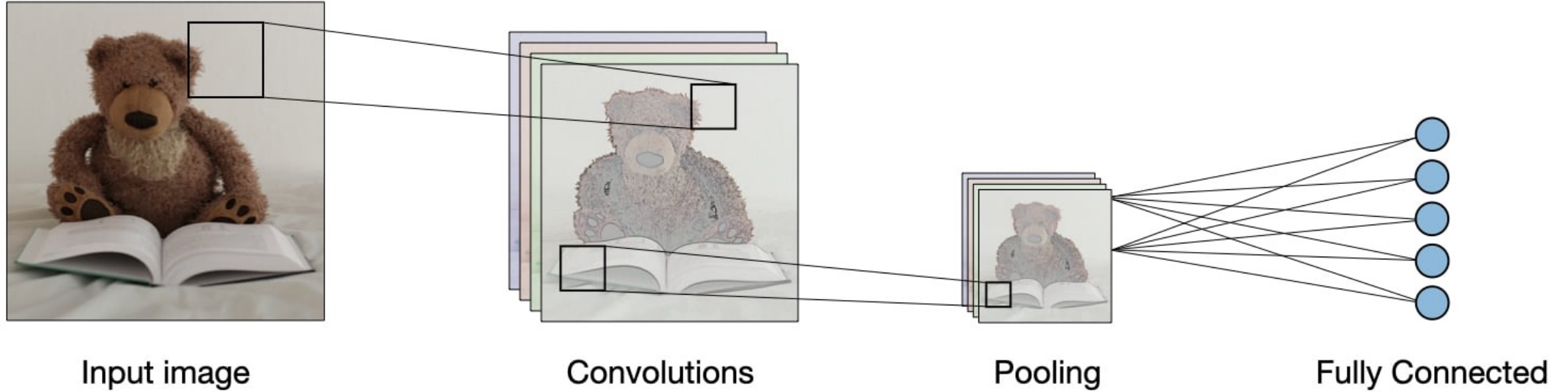
Pooled
feature

卷积神经网络(Convolutional Neural Network, CNN)

- **采样层**：亦称“池化层”，其作用是基于局部相关性原理进行亚采样，从而在减少数据量的同时保留有用信息（降低参数量级）

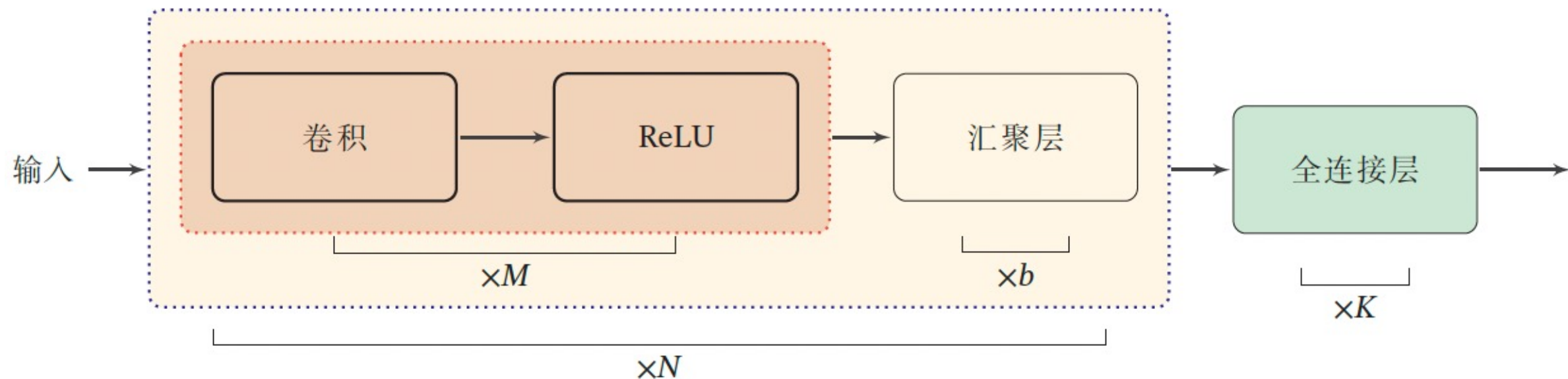


卷积神经网络(Convolutional Neural Network, CNN)



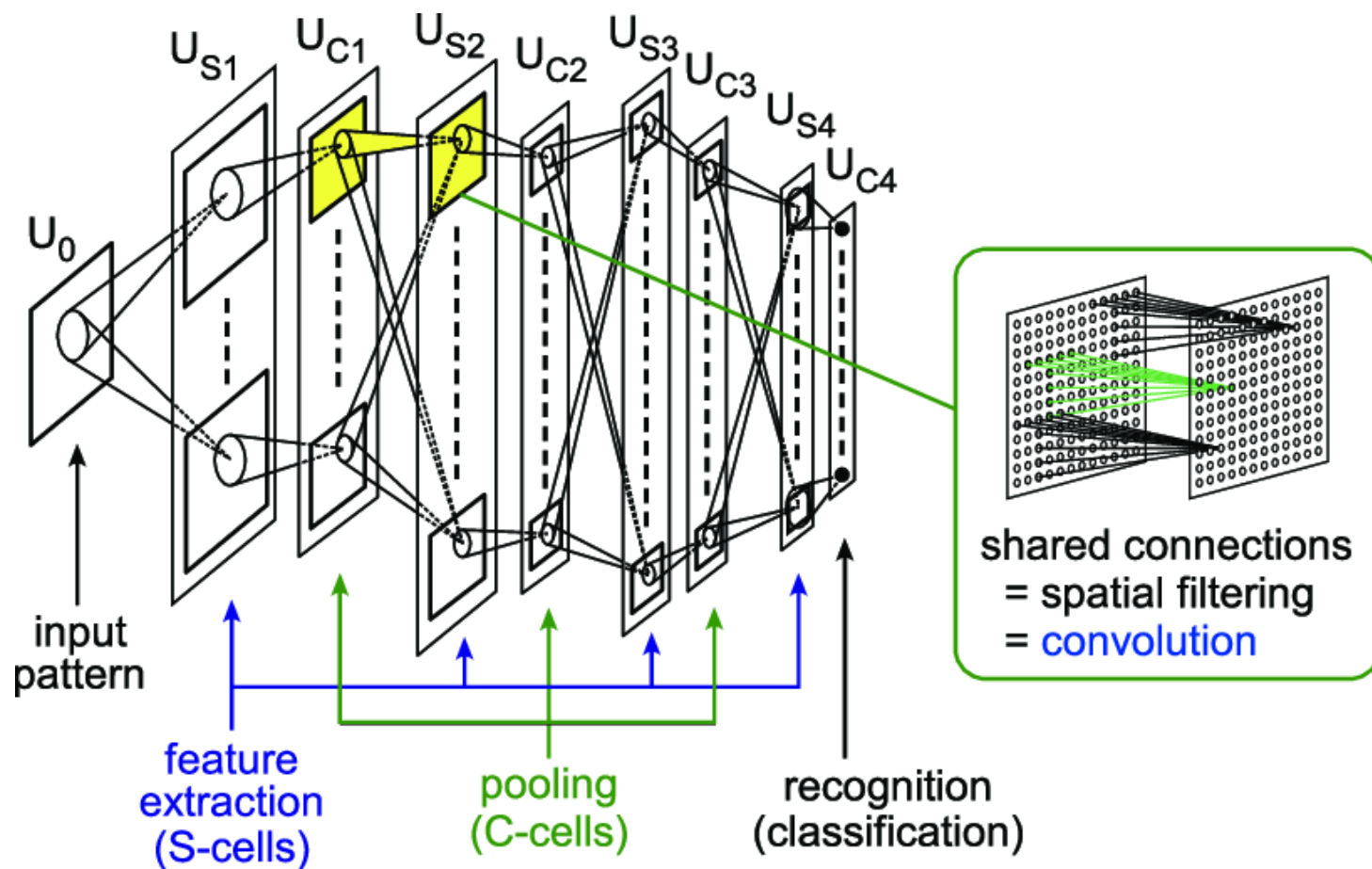
卷积神经网络(Convolutional Neural Network, CNN)

- 卷积神经网络：一般由卷积层、采样层和全连接层构成的前馈神经网络，具有局部连接、权值共享、汇聚等特性，使得卷积神经网络具有一定程度的平移、缩放和旋转不变性，同时与前馈神经网络相比，参数更少



卷积神经网络(1980-)

Neocognitron



Kenji Fukushima

卷积神经网络(1980-2010)

在20世纪80年代末和90年代初，Yann LeCun和他的团队进一步开发了CNN，引入了专门为手写数字识别设计的LeNet-5架构，应用于美国支票识别业务

LetNet-5 [LeCun and Bengio, 1995; LeCun et al. , 1998]

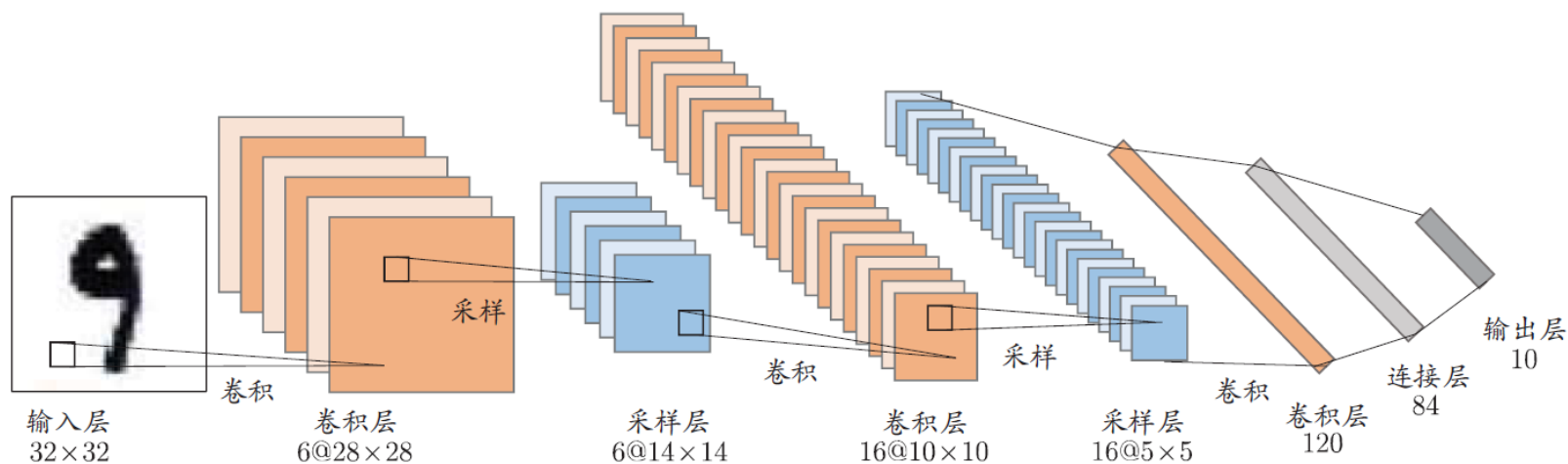


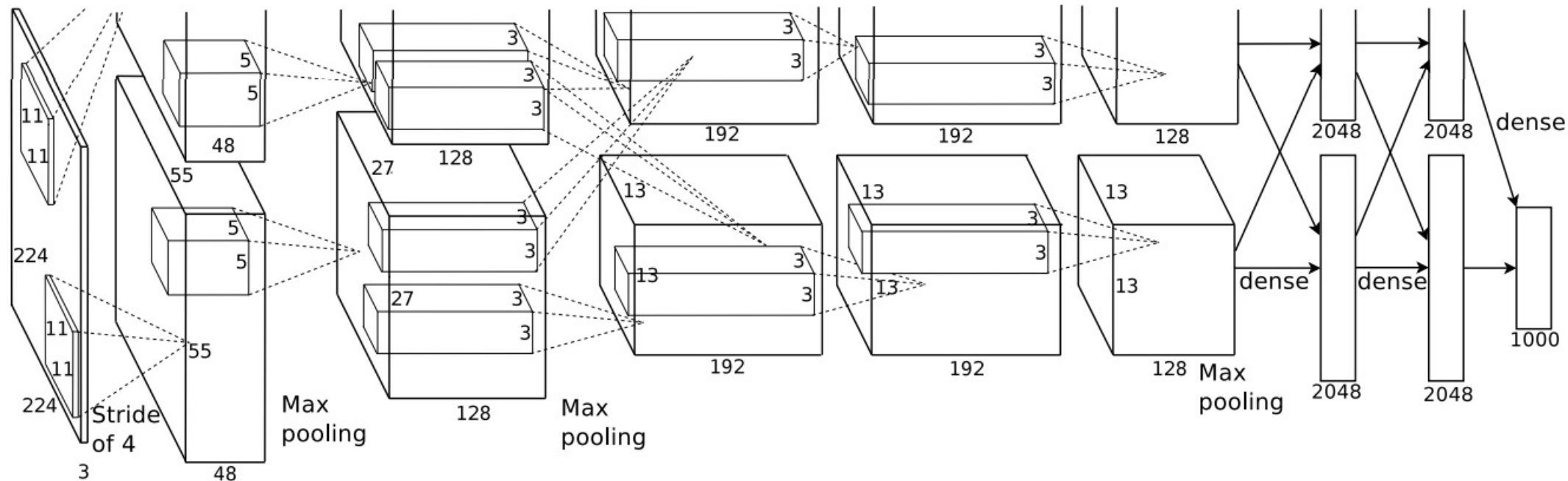
图 5.15 卷积神经网络用于手写数字识别 [LeCun et al., 1998]



Yann LeCun

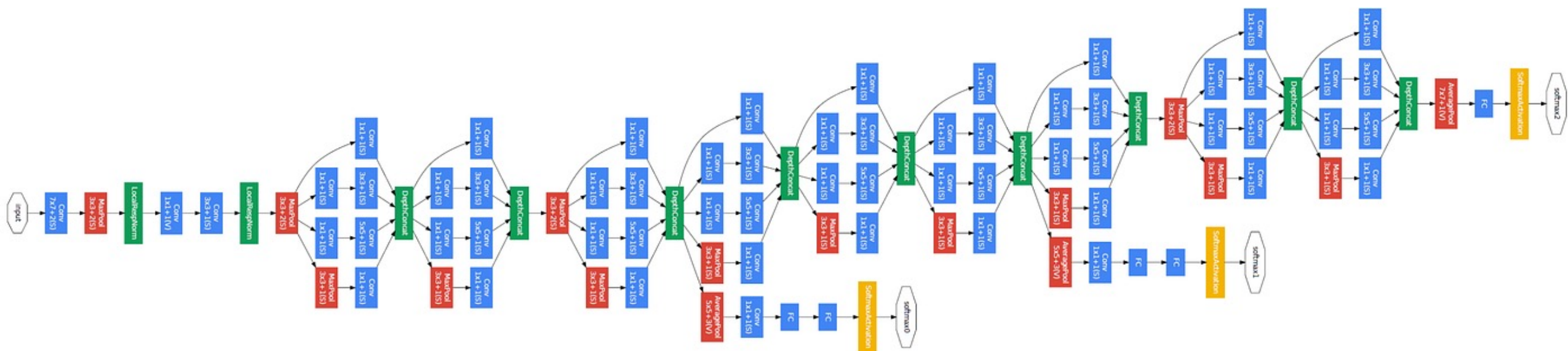
卷积神经网络-AlexNet

- 2012 ILSVRC winner (top 5 error of 16% compared to runner-up with 26% error)
 - 第一个现代深度卷积网络模型
 - 首次使用了很多现代深度卷积网络的一些技术方法
 - 使用GPU进行并行训练，采用了ReLU作为非线性激活函数
 - 使用Dropout防止过拟合，使用数据增强



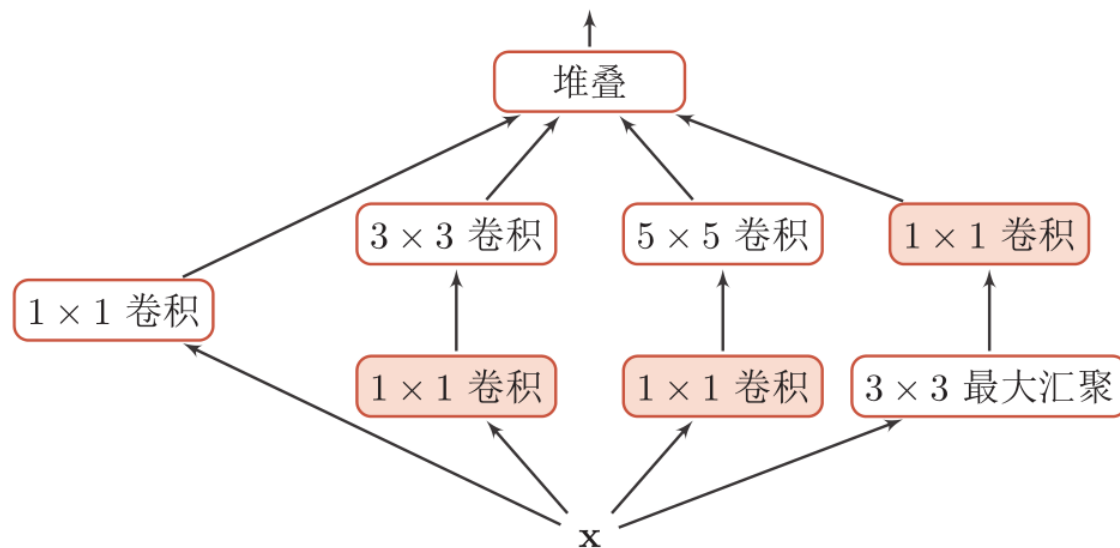
卷积神经网络-InceptionNet

- 2014 ILSVRC winner (22层)
 - 参数: GoogLeNet: 4M VS AlexNet: 60M
 - 错误率: 6.7%
 - Inception网络是由有多个inception模块和少量的汇聚层堆叠而成



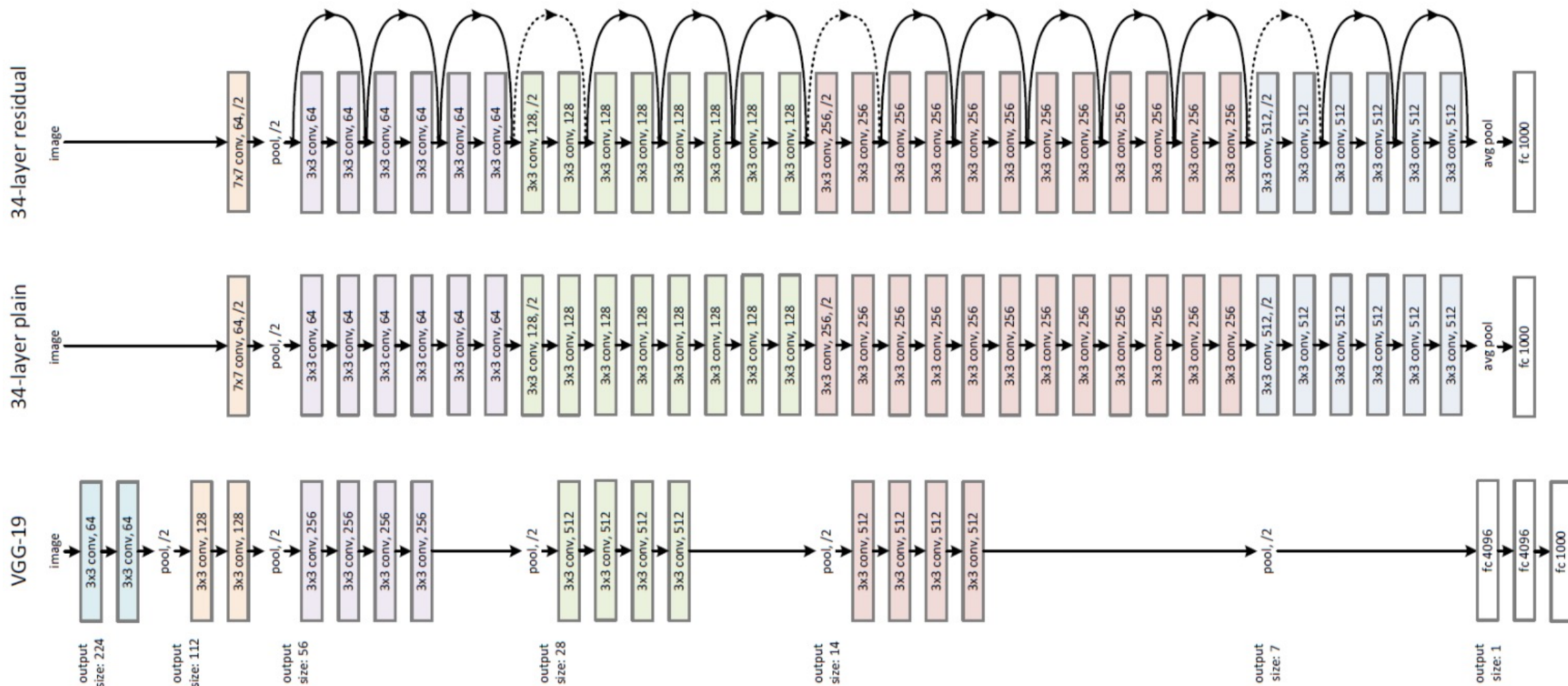
卷积神经网络-InceptionNet

- 在卷积网络中，如何设置卷积层的卷积核大小是一个十分关键的问题
- 在Inception网络中，一个卷积层包含多个不同大小的卷积操作，称为Inception模块
- Inception模块同时使用 1×1 、 3×3 、 5×5 等不同大小的卷积核，并将得到的特征映射在深度上拼接（堆叠）起来作为输出特征映射



卷积神经网络-ResNet

- 2015 ILSVRC winner (152层)
- 错误率: 3.57%



实例：CIFAR-10图像分类

<https://www.kaggle.com/c/cifar-10>



CIFAR-10 - Object Recognition in Images

Identify the subject of 60,000 labeled images

231 teams · 4 years ago

[Overview](#)

[Data](#)

[Discussion](#)

[Leaderboard](#)

[Rules](#)

Overview

Description

Evaluation

CIFAR-10 is an established computer-vision dataset used for object recognition. It is a subset of the [80 million tiny images dataset](#) and consists of 60,000 32x32 color images containing one of 10 object classes, with 6000 images per class. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

实例：ImageNet狗品种识别

<https://www.kaggle.com/c/dog-breed-identification>



扩展阅读

- 《动手学深度学习》：<https://zh.d2l.ai/>
- Fei-Fei Li's Lecture: <http://cs231n.stanford.edu/>
- 常用数据集：
- 图像分类：ImageNet、CIFAR、MNIST
- 目标检测：COCO数据集
- 语义分割：Pascal VOC2012
- 视频数据集：Kinetics-700
- ...