

# Lecture 8



# Introduction: 自动瞄准系统构成

Part 1: 识别目标

Part 2: 运动状态估计

Part 3: 击打规划

# 数字图像处理

# 图像的数字表示

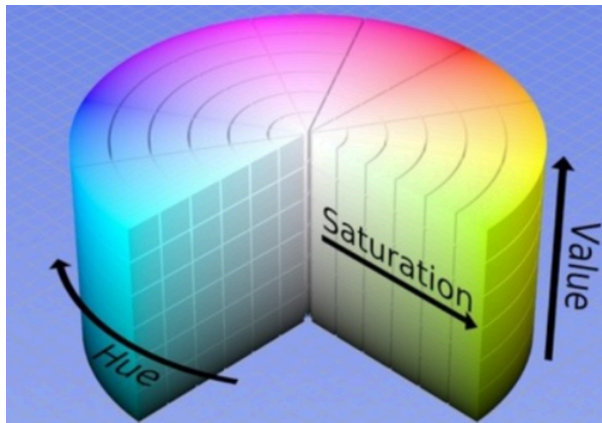
分辨率：图像拥有的像素数量。一般表示方法为 $w * h$ 。

- 一张横向有1920像素，纵向有1080个像素的图像，分辨率为 $1920 * 1080$

颜色空间：每个像素用三个8-bit数(0-255)表示, 表示此像素点的颜色信息。

- 不同的颜色空间，像素的值有不同含义。opencv提供将图像在不同颜色空间中进行转换的函数

- RGB颜色空间：Red, Green, Blue
- HSV颜色空间：色相，饱和度与亮度



HSV颜色空间

# 灰度图 & 二值图

## 灰度图

每个像素点只有一个值，表示灰度值。0表示黑色，255表示白色。

- 灰度图的计算方法：

$$Gray = 0.299 * R + 0.587 * G + 0.114 * B$$

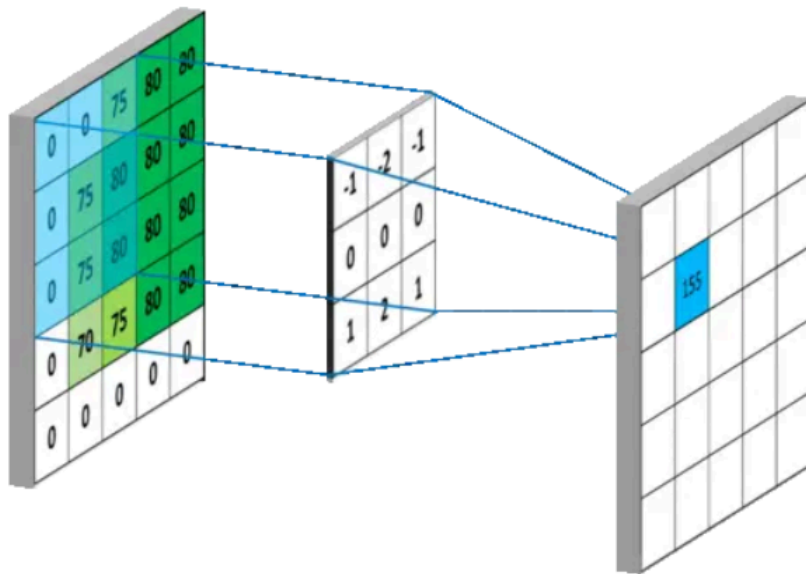
## 二值图

特殊的灰度图，像素点的灰度只有0和255两种情况

稍后会讨论如何将灰度图转换为二值图



# 卷积



## Demo: 均值滤波

|   |    |    |    |    |
|---|----|----|----|----|
| 0 | 0  | 75 | 80 | 80 |
| 0 | 75 | 80 | 80 | 80 |
| 0 | 75 | 80 | 80 | 80 |
| 0 | 70 | 75 | 80 | 80 |
| 0 | 0  | 0  | 0  | 0  |

\*

|     |     |     |
|-----|-----|-----|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

=

|  |    |  |  |  |
|--|----|--|--|--|
|  |    |  |  |  |
|  | 43 |  |  |  |
|  |    |  |  |  |
|  |    |  |  |  |
|  |    |  |  |  |



# 形态学运算

- **膨胀:** 用一个结构元素在图像上滑动，如果结构元素与图像重叠的部分有一个像素是非零的，那么图像上对应的像素就是1，否则为0。
- **腐蚀:** 用一个结构元素在图像上滑动，如果结构元素与图像重叠的部分全是非零像素，那么图像上对应的像素就是1，否则为0。
- **开运算:** 先腐蚀后膨胀
- **闭运算:** 先膨胀后腐蚀

# Introduction to Deep Learning

# 多层感知器: 神经元

可用于拟合一些

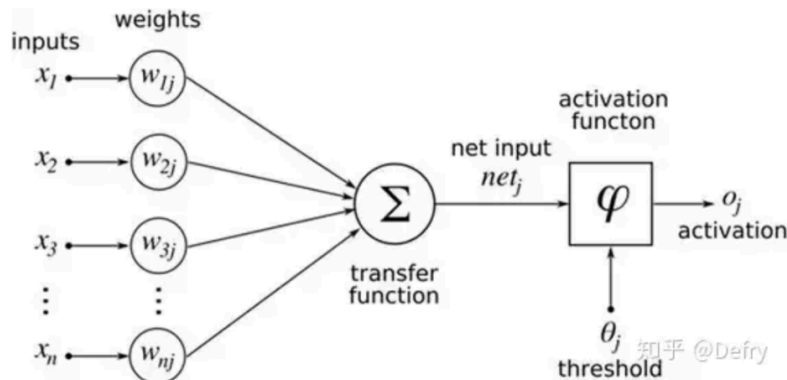
- 输入层：接受输入信号
- 权重：每个输入信号的权重
- 激活函数：将输入信号的加权和转换为输出信号

正向传播：

$$output = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4$$

反向传播：

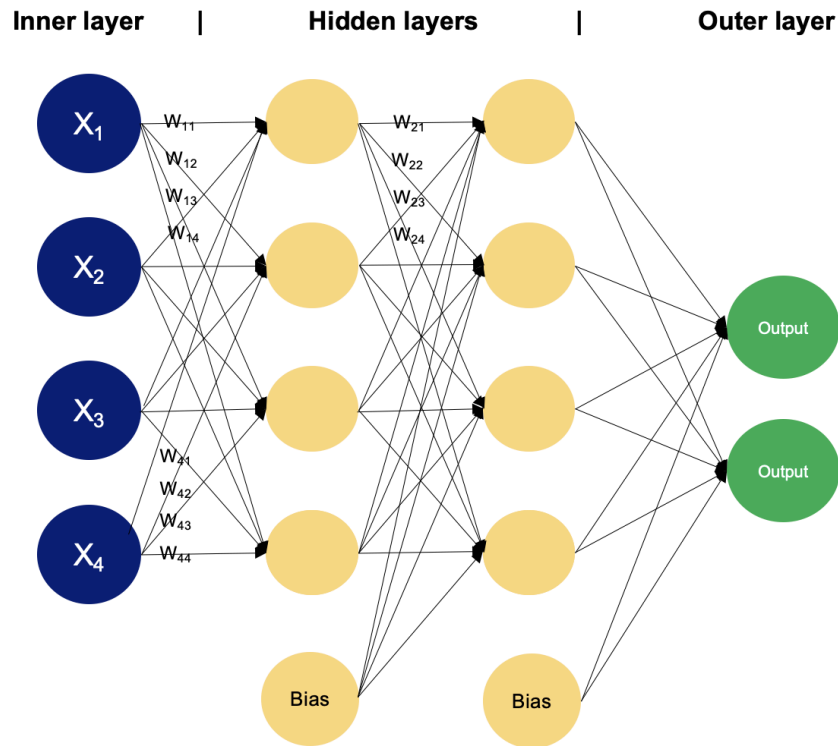
$$\frac{\partial E}{\partial w_{1,j}} = \frac{\partial E}{\partial output} * \frac{\partial output}{\partial w_{1,j}}$$



# 多层感知器

由多层神经元组成的结构

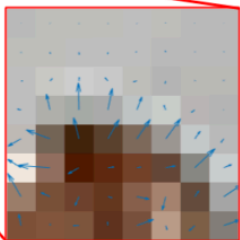
- 拟合函数
- 激活函数引入非线性项



# 图像的特征&卷积神经网络

**特征:** 描述图像某个区域的某种属性。直接对图像处理计算量过大时，通常会使用一些特征提取算法提取出特征，来减少运算量，去除不必要的数。

Example: HOG(Histogram of oriented gradients) 定向梯度直方图

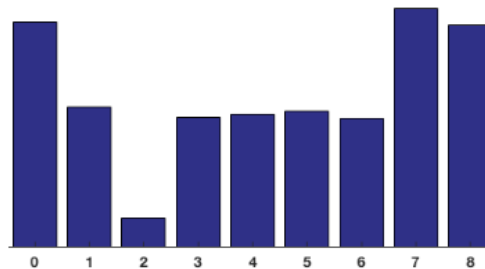


|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 2   | 3   | 4   | 4   | 3   | 4   | 2   | 2   |
| 5   | 11  | 17  | 13  | 7   | 9   | 3   | 4   |
| 11  | 21  | 23  | 27  | 22  | 17  | 4   | 6   |
| 23  | 99  | 165 | 135 | 85  | 32  | 26  | 2   |
| 91  | 155 | 133 | 136 | 144 | 152 | 57  | 28  |
| 98  | 196 | 76  | 38  | 26  | 60  | 170 | 51  |
| 165 | 60  | 60  | 27  | 77  | 85  | 43  | 136 |
| 71  | 13  | 34  | 23  | 108 | 27  | 48  | 110 |

Gradient Magnitude

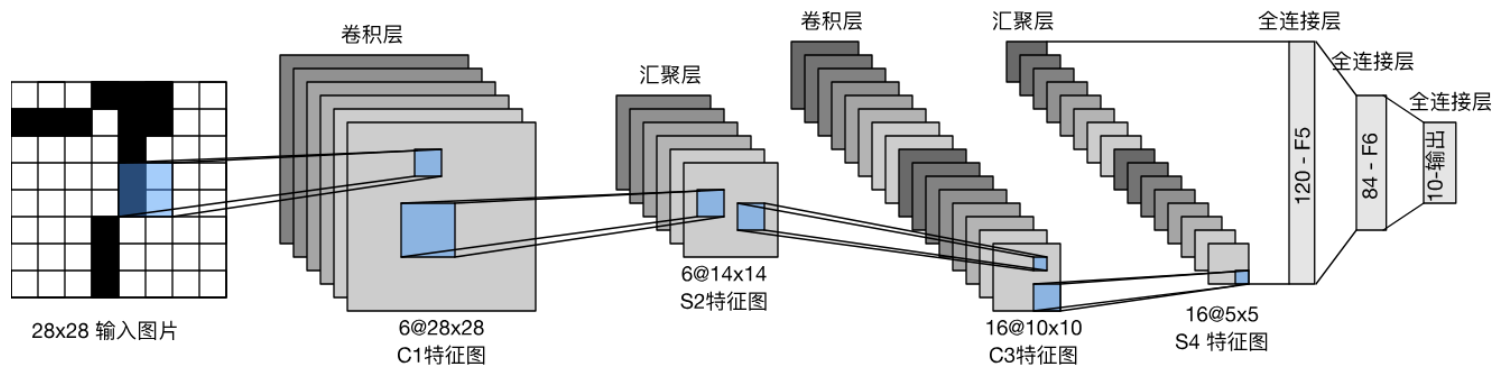
|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 80  | 36  | 5   | 10  | 0   | 64  | 90  | 73  |
| 37  | 9   | 9   | 179 | 78  | 27  | 169 | 166 |
| 87  | 136 | 173 | 39  | 102 | 163 | 152 | 176 |
| 76  | 13  | 1   | 168 | 159 | 22  | 125 | 143 |
| 120 | 70  | 14  | 150 | 145 | 144 | 145 | 143 |
| 58  | 86  | 119 | 98  | 100 | 101 | 133 | 113 |
| 30  | 65  | 157 | 75  | 78  | 165 | 145 | 124 |
| 11  | 170 | 91  | 4   | 110 | 17  | 133 | 110 |

Gradient Direction



# 卷积神经网络

卷积核为可学习的参数



# YOLO(You Only Look Once)

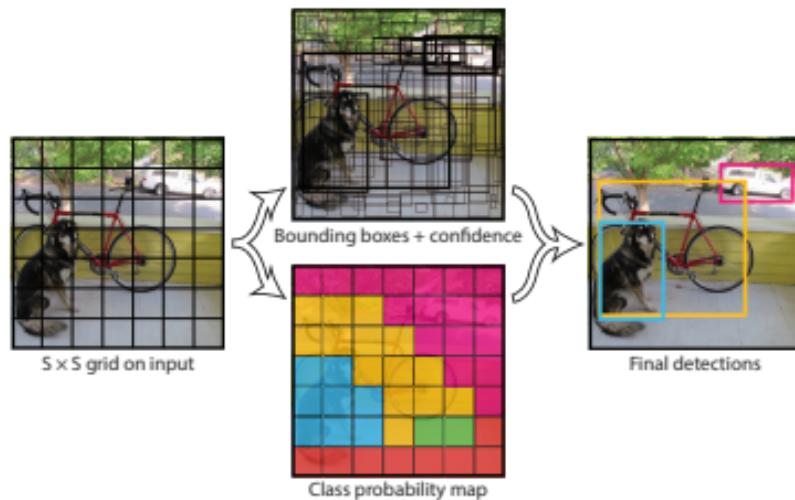
YOLO是一种单阶段、快速的深度学习目标检测模型。在十余年的发展中，YOLO进化出了许多版本。

这里以YOLOv1举例。

队内目前使用的装甲板识别模型基于yolov5更改得来。

1. 使用CNN对图像进行处理，将原图像转化为 $7 * 7 * 1024$ 的特征图
2. 将此特征图放入mlp，运算得到 $7 * 7 * 30$ 的结果向量, 包含2组物体的信息(中心点位置，宽高，类别概率)
3. 损失函数为结果向量的最后一个维度与对应图像区域标签之差

一个可行的理解是，YOLO将图像特征提取后，用一个 $1024 - 30$ 的mlp拟合了 **特征向量-物体信息** 的 函数。



# Project Preview