

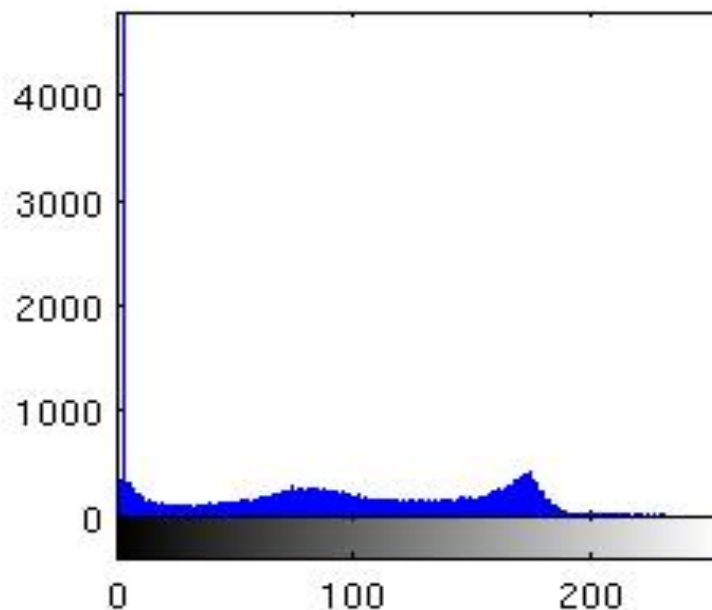
图像处理和模式识别

- ◆ 图像处理基础
- ◆ 人脸检测原理
- ◆ 机器视觉进展

4.1 直方图



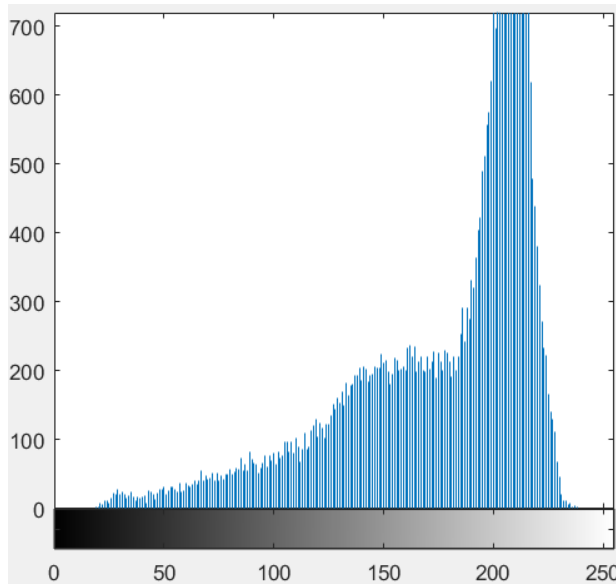
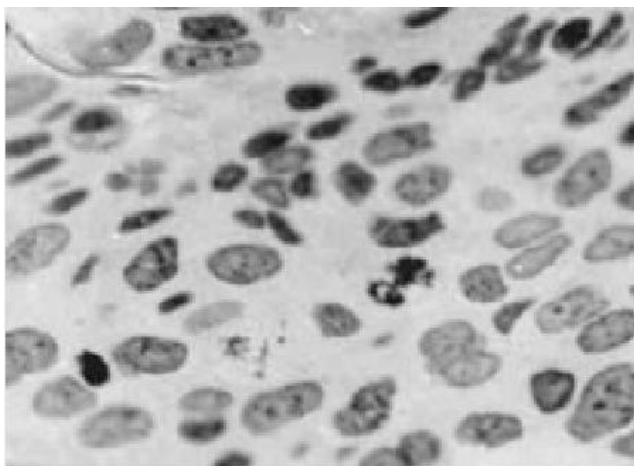
- 直方图是图像处理中非常有用的工具，它代表灰度的分布；
- 横坐标指示不同的灰度级，纵坐标指示某一灰度级的像素数。



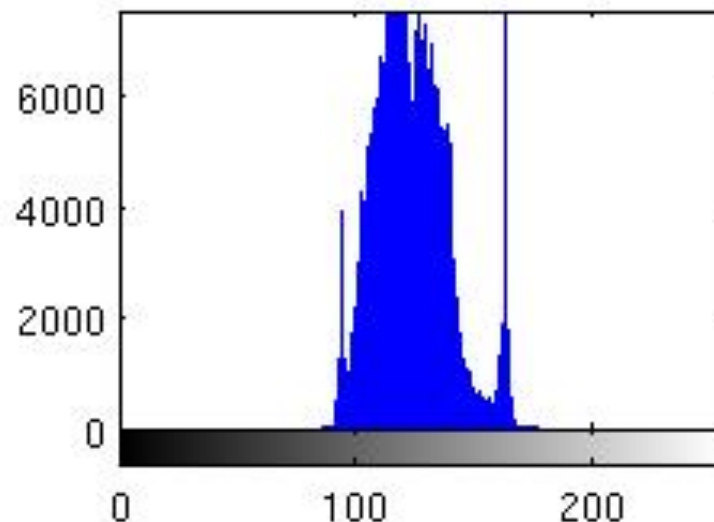
为了分割提取大脑图像，我们分离像素值小于100和大于100的图像来提取有效信息。



- 根据阈值对像素点分类，将灰度图像转为黑白二值图像；
- 自适应阈值选取：Otsu算法、Niblack算法等。



动态范围对图像来说非常重要，其直接关系到图像的质量。

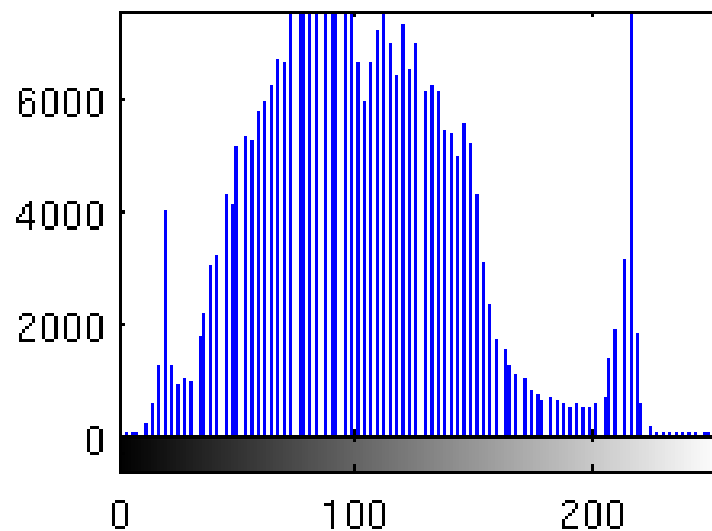
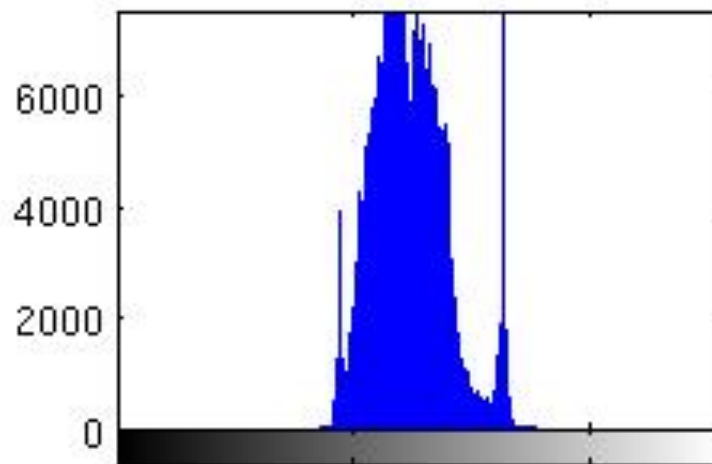


可以使用如下的方法调整该图像的动态范围：

$$S(n) = \text{round} \left(M \frac{(n - a)}{(b - a)} \right)$$

其中， n 是输入图像某一级灰度； a 和 b 是输入图像灰度级的下界和上界； M 是图像允许的最大灰度级（通常为255）。

动态范围对图像来说非常重要，其直接关系到图像的质量。



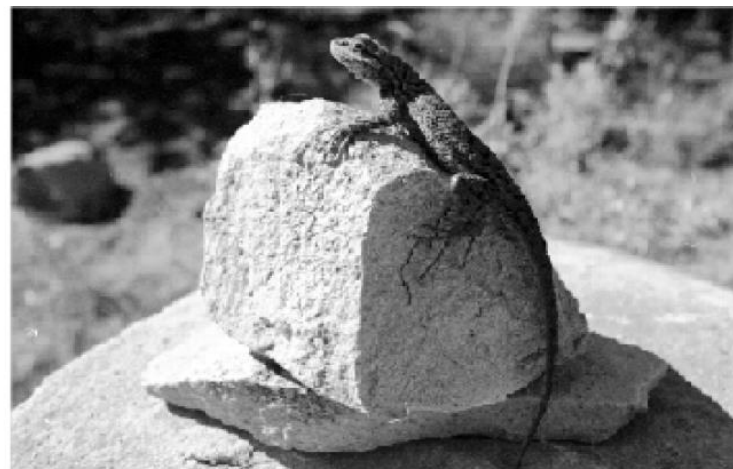
- 累积直方图的每个灰度级对应的值为小于或等于当前灰度级的所有像素数；
- 我们可以通过将该数字除以图像中的像素总数来对其进行归一化；
- 累积直方图纵轴的值将随灰度等级增长而增长，直到1。
- 累积直方图最著名的应用是直方图均衡；
- 为每个灰度级分配相同数量的像素，使直方图变得平坦；
- 我们通过以下公式获得的直方图均衡变换：

$$H_c(n) = \frac{1}{nb_pixels} \sum_{j=0}^n H(j) \quad \text{for } n = 0, \dots, 255$$

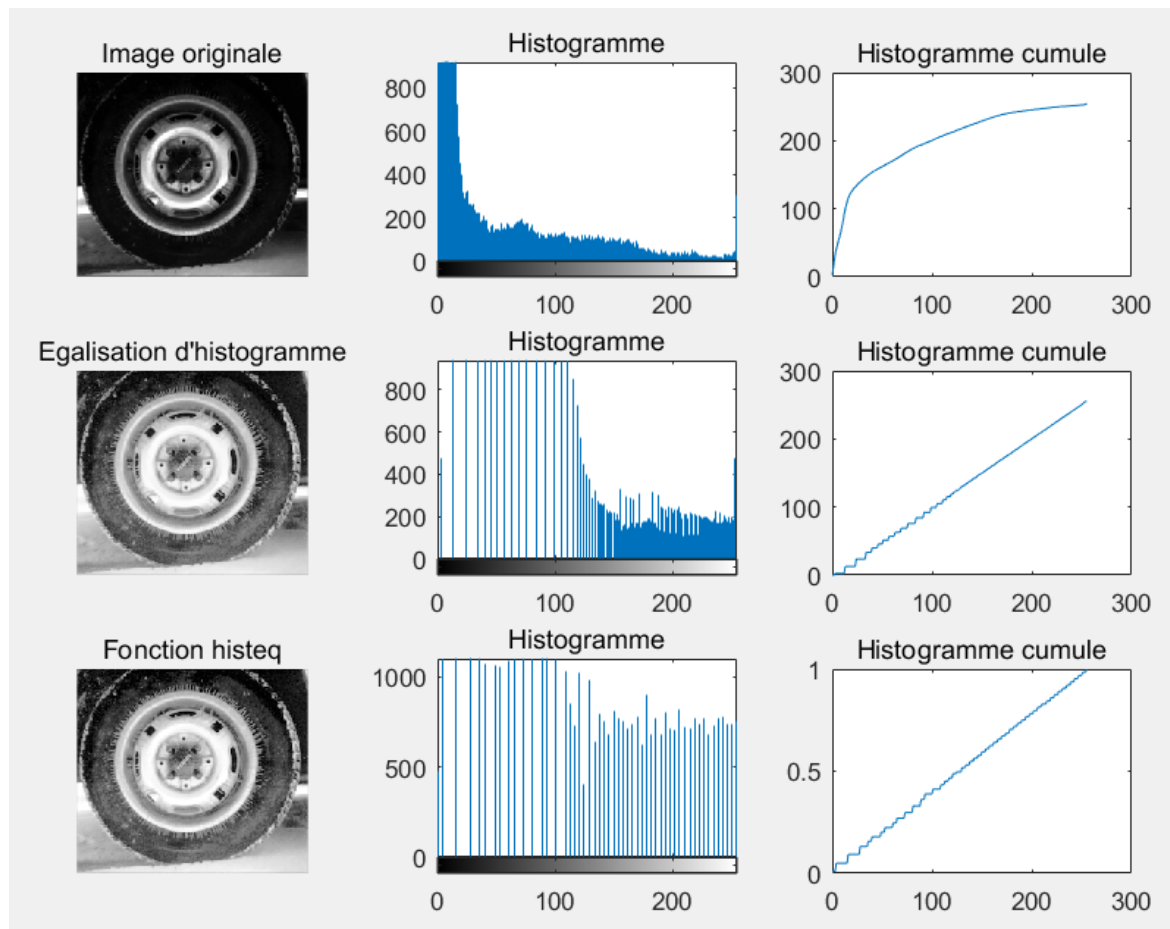
$$S(n) = \text{round} [255H_c(n)]$$

其中， n 是当前灰度级； $H(i)$ 代表累计直方图在灰度级 i 处的值。

- 累积直方图的每个灰度级对应的值为小于或等于当前灰度级的所有像素数；
- 我们可以通过将该数字除以图像中的像素总数来对其进行归一化；
- 累积直方图纵轴的值将随灰度等级增长而增长，直到1。
- 累积直方图最著名的应用是直方图均衡；
- 为每个灰度级分配相同数量的像素，使直方图变得平坦；
- 我们通过以下公式获得的直方图均衡变换：



- 累积直方图的每个灰度级对应的值为小于或等于当前灰度级的所有像素数；
- 我们可以通过将该数字除以图像中的像素总数来对其进行归一化；
- 累积直方图纵轴的值将随灰度等级增长而增长，直到1。

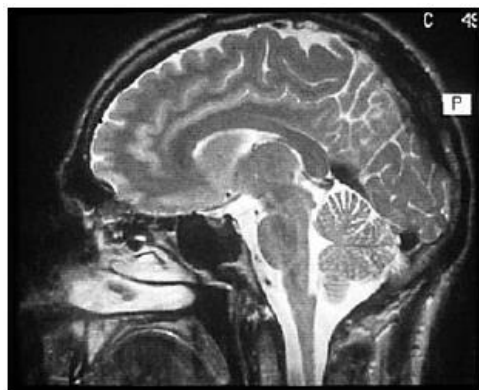


- 图像处理的典型技术是线性滤波器;
- 该滤波器是通过对图像进行卷积操作完成的;
- 因此滤波器也称为卷积核。

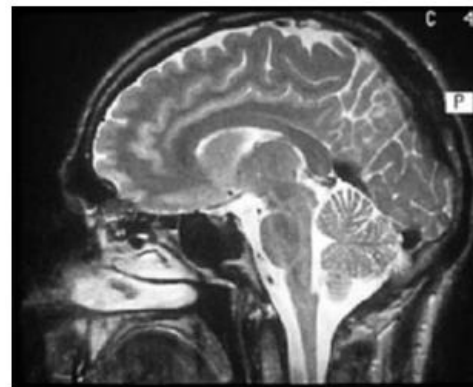
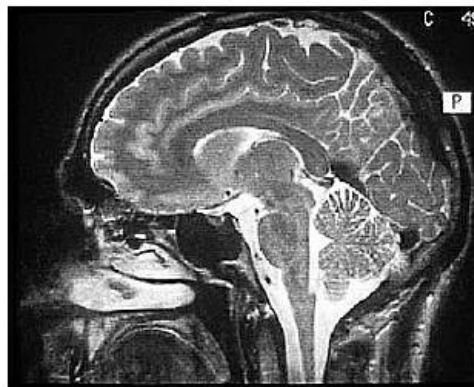
对于图像的卷积操作:

- 将卷积核 (以 3×3 矩阵为例) 的中心置于图像当前像素;
- 卷积核逆时针旋转180度和图像对应元素相乘;
- 将所有乘积相加, 替换当前像素的值, 然后卷积核滑动至下一像素处理。
- 卷积核各个元素的和不大于1, 以避免使得卷积后的图像过度饱和。

高通滤波



低通滤波



卷积滤波器也可以用来计算梯度。



$$\star \begin{pmatrix} -1 & 0 & 1 \\ -1 & \mathbf{0} & 1 \\ -1 & 0 & 1 \end{pmatrix} =$$



Approximation of $I_x = \frac{\partial I}{\partial x}$.

卷积滤波器也可以用来计算梯度。

- x 和 y 方向上二阶导的和 (Laplacian) : $\Delta I = I_{xx} + I_{yy}$



$$\star \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} =$$



有三个基本的顺序统计滤波器：

- 最小值滤波器返回邻域9个像素中的最小值，最小值滤波使图像变暗；
- 最大值滤波器返回邻域9个像素中的最大值，最大值滤波使图像变亮；

最小值滤波器



最大值滤波器



有三个基本的顺序统计滤波器：

- 最小值滤波器返回邻域9个像素中的最小值，最小值滤波使图像变暗；
- 最大值滤波器返回邻域9个像素中的最大值，最大值滤波使图像变亮；
- 均值滤波器返回邻域9个像素中的均值，均值滤波器可消除图像噪声。



- 图像膨胀、腐蚀是形态学中的两种对偶操作。
- 用滤波器扫描图像，将非零值覆盖区域的**最大像素值**作为中心点输出值；
- 图片中亮的区域“膨胀”。
- 用滤波器扫描图像，将非零值覆盖区域的**最小像素值**作为中心点输出值；
- 图片中亮的区域“腐蚀”。

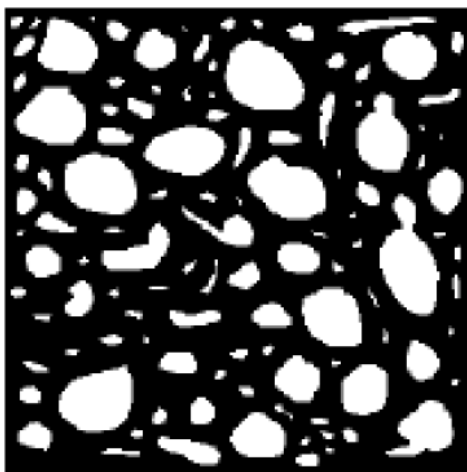
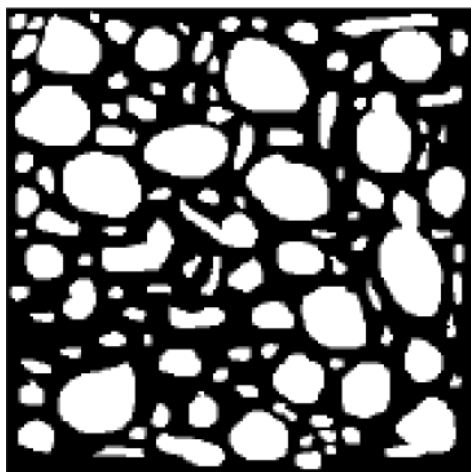
$$SE = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$SE = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

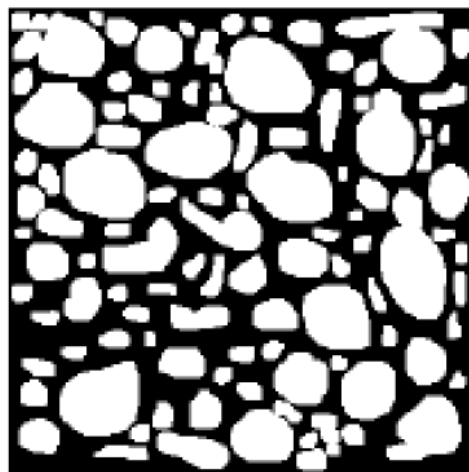
$$SE = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \dots$$

- 图像膨胀、腐蚀是形态学中的两种对偶操作。
- 用滤波器扫描图像，将非零值覆盖区域的**最大像素值**作为中心点输出值；
- 图片中亮的区域“膨胀”。
- 用滤波器扫描图像，将非零值覆盖区域的**最小像素值**作为中心点输出值；
- 图片中亮的区域“腐蚀”。

腐蚀



膨胀



- 可以对腐蚀和膨胀进行连续操。
- **开运算**就是先腐蚀后膨胀，他的主要作用是消除一些小的亮点：

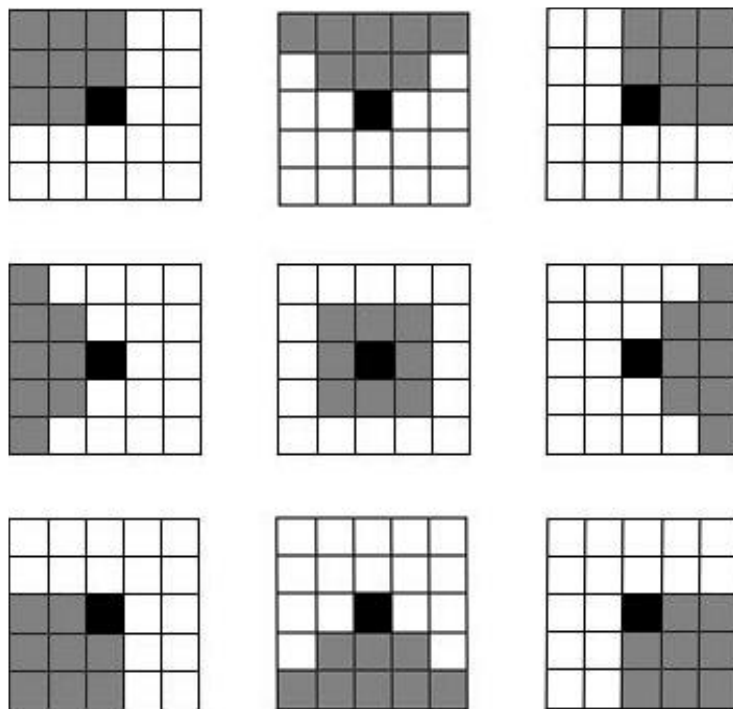


- **闭运算**就是先膨胀后腐蚀，他常被用来填充前景物体中的小洞或缝隙：



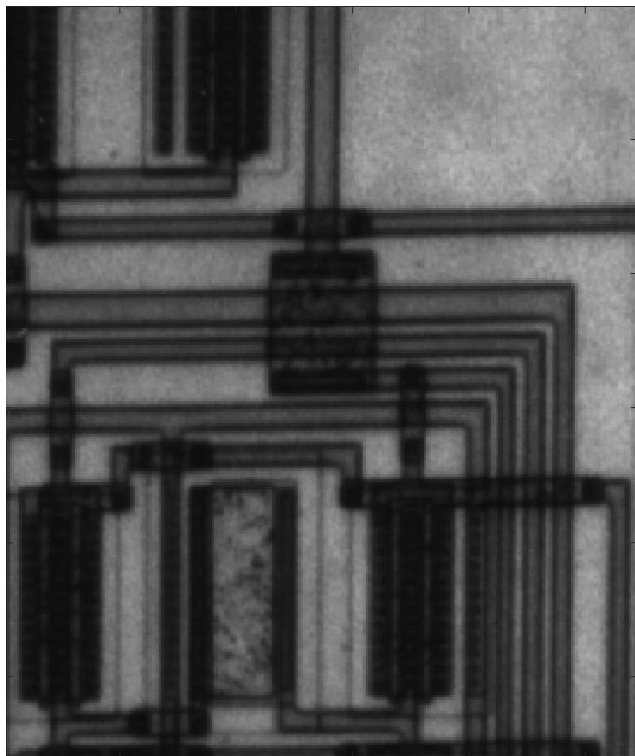
Nagao滤波器:

- 对于每个窗口我们计算非白色像素的方差值;
- 然后保留最小方差的窗口, 计算像素均值, 代替原有中心像素;
- 这种过滤可以使图像平滑、消除强噪点、保留轮廓, 多次迭代增强对比度。

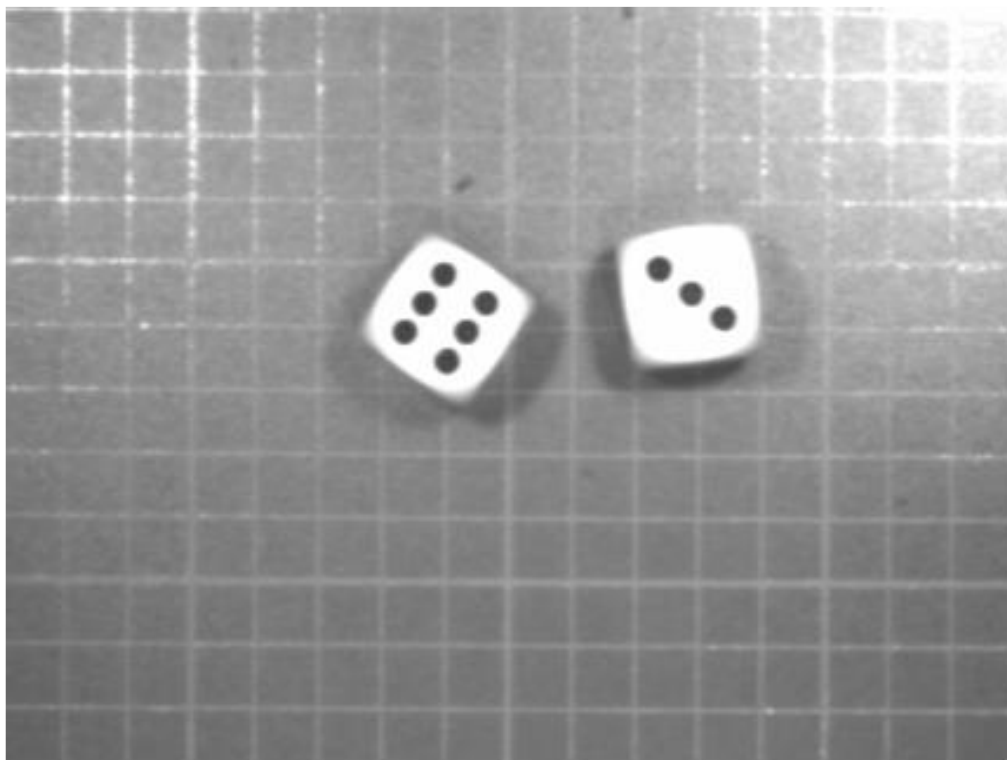


Nagao滤波器:

- 对于每个窗口我们计算非白色像素的方差值;
- 然后保留最小方差的窗口, 计算像素均值, 代替原有中心像素;
- 这种过滤可以使图像平滑、消除强噪点、保留轮廓, 多次迭代增强对比度。



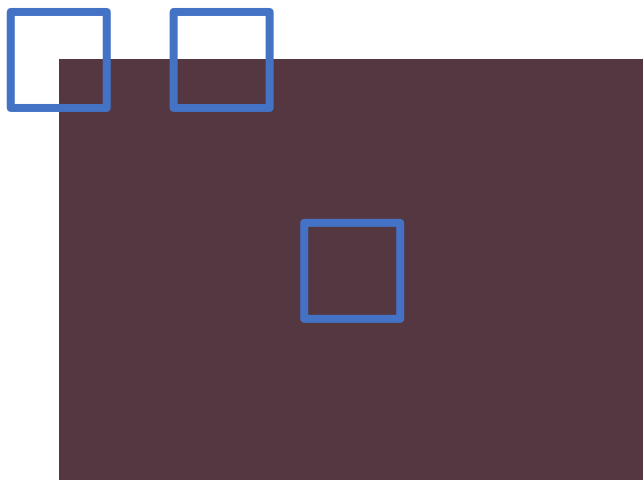
应用所学的图像处理技术，如何计算图片中色子的数目？



4.2 图像特征



关键点检测



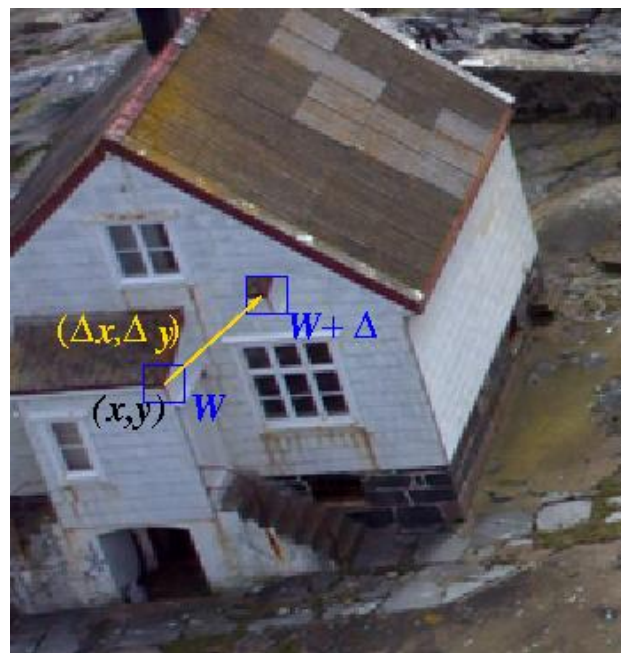
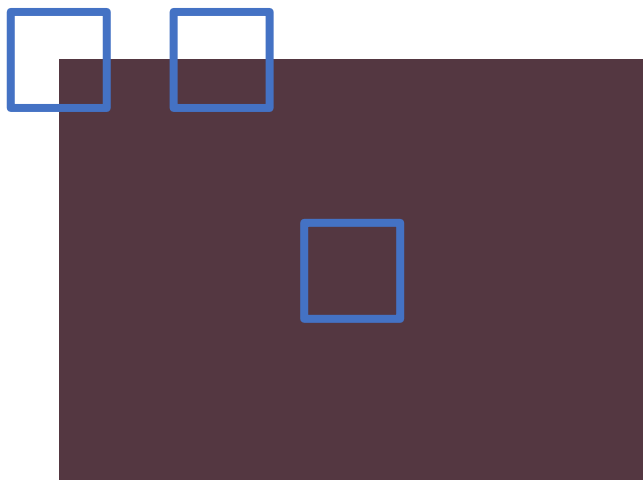
4.2 图像特征



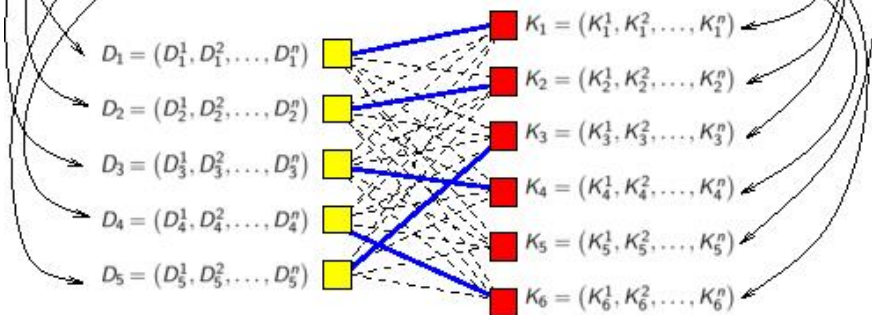
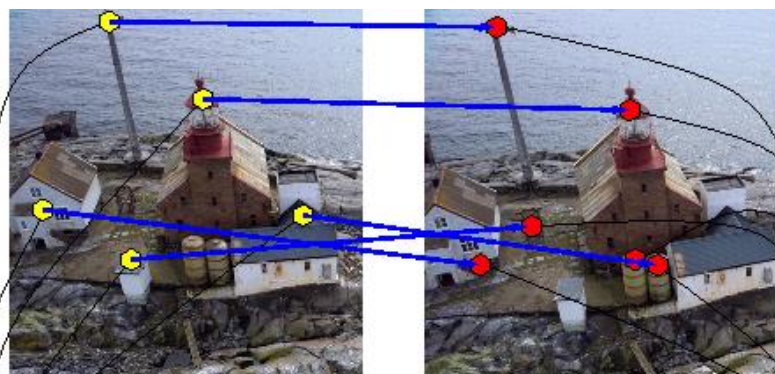
关键点检测

- 角点：图像中不同方向亮度剧烈变化的部分。

- 计算公式：
$$\chi(x, y) = \sum_{(i, j) \in W} (I(i, j) - I(i + \Delta x, j + \Delta y))^2$$



4.2 图像特征



图像特征：

- 用来表示图像块信息的一个向量

一个好的特征描述子应该：

- 低维（高效性）
- 易辨别（有效性）
- 适应图像变化（对旋转、光线稳健）

举例：

临近图像块的直方图、亮度变化等



4.2 视觉目标跟踪

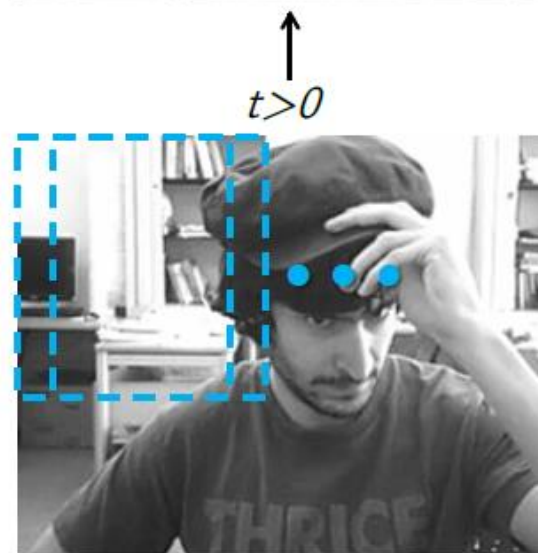
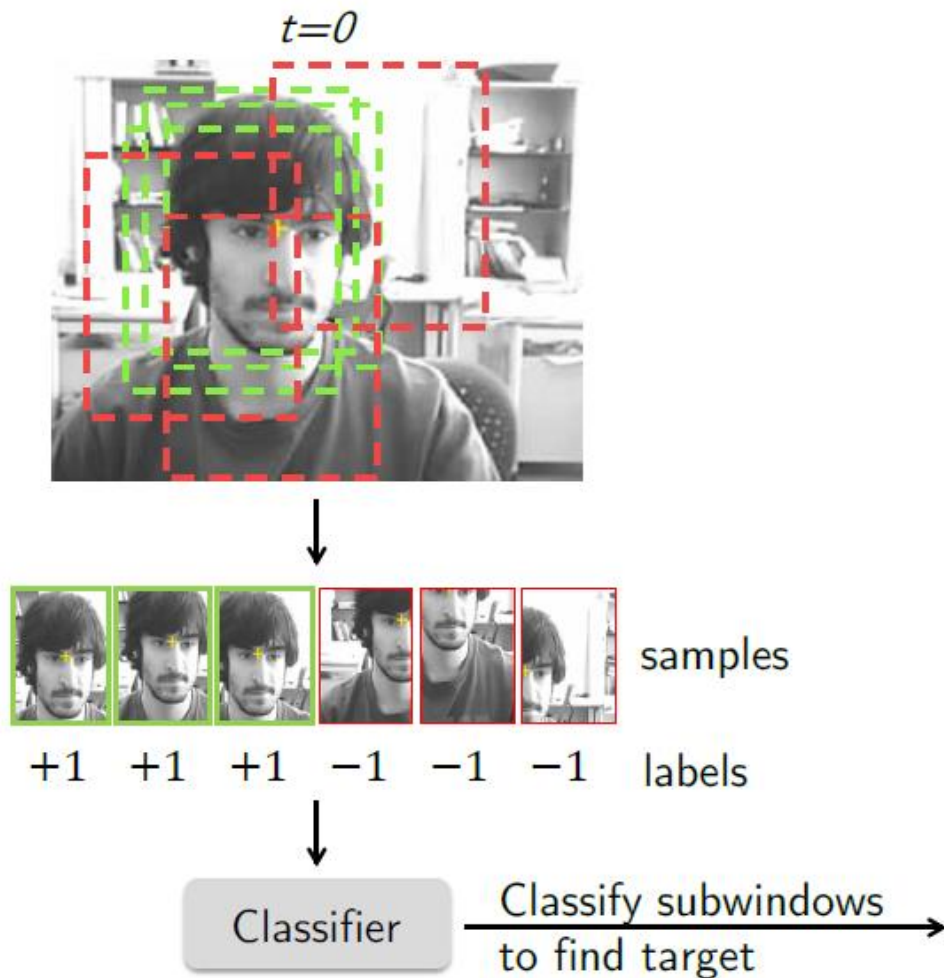


Image source:
J. Matas, ICVSS 2017

智能无人系统综合设计



4.2 视觉目标跟踪

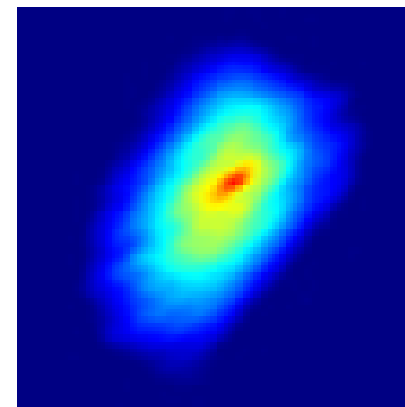


生成跟踪响应图：

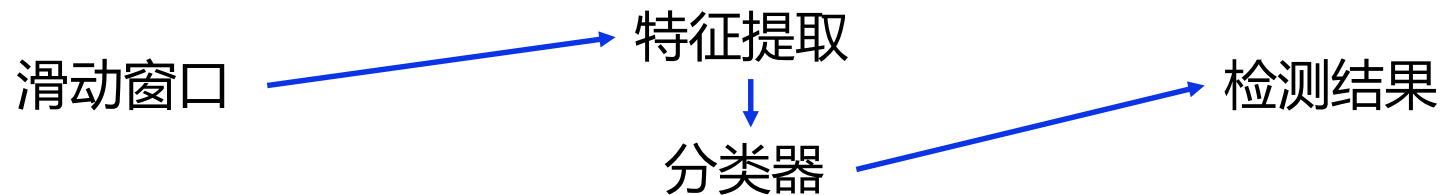


$$y_i = \mathbf{w}^T \mathbf{x}_i$$

$$\mathbf{y} = \mathbf{x} \circledast \mathbf{w}$$



4.2 人脸检测



4.2 Haar-like 特征



A



B



C



D



E

- Haar特征分为三类：边缘特征、线性特征、中心特征和对角线特征。
- Haar特征值反映了图像的灰度变化情况。
- 对于矩形特征 E，计算时需要对黑色区域乘以2，以使两种像素数目一致。
- Haar特征需要进行归一化处理：

$$sum = \sum i(x, y)$$

$$mean = \frac{sum}{w \times h}$$

$$norm_value = \frac{feature_value}{\sqrt{mean_sq - mean^2}}$$

$$sum_sq = \sum i^2(x, y)$$

$$mean_sq = \frac{sum_sq}{w \times h}$$



4.2 积分图



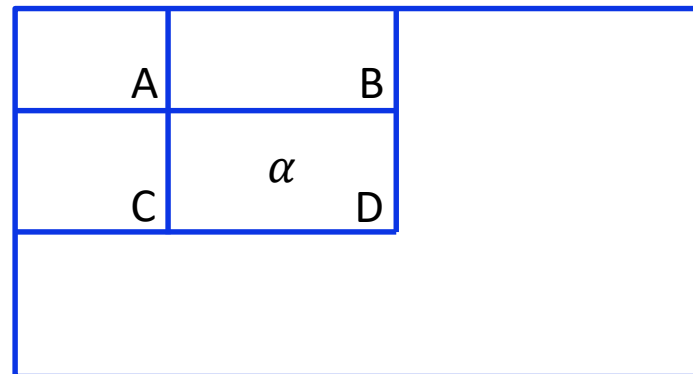
- 积分图：计算像素左上方所有像素之和。
- 积分图的构建：
逐行扫描图像，计算像素 (i, j) 行方向的累加和 $s(i, j)$ 和积分图像 $ii(i, j)$ 的值。

$$s(i, j) = s(i, j - 1) + f(i, j)$$

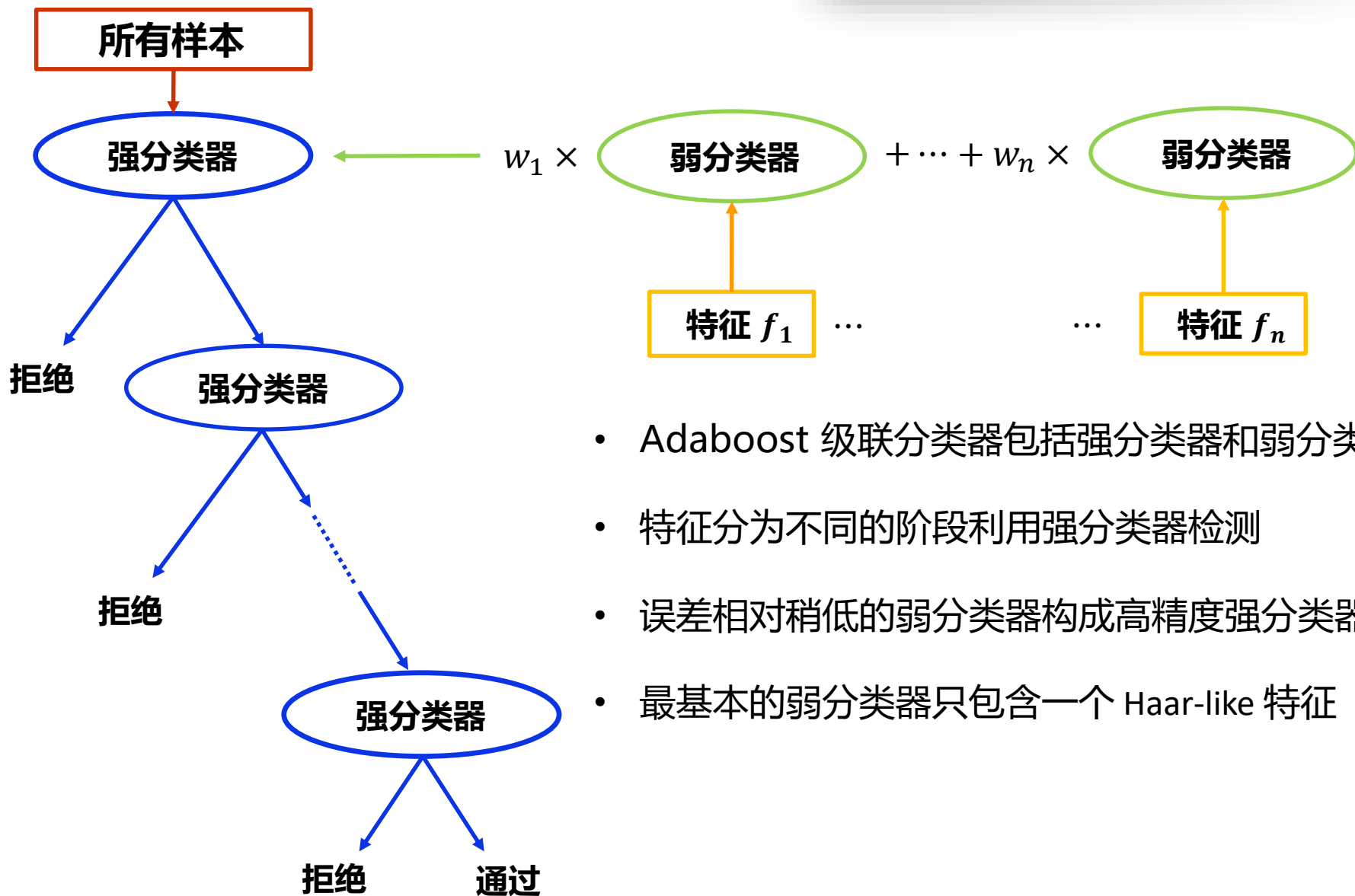
$$ii(i, j) = ii(i, -1, j) + s(i, j)$$

- 积分图的使用：

$$\alpha_{sum} = ii(D) - ii(C) - ii(B) + ii(A)$$



4.2 Adaboost 级联分类器



- Adaboost 级联分类器包括强分类器和弱分类器
- 特征分为不同的阶段利用强分类器检测
- 误差相对稍低的弱分类器构成高精度强分类器
- 最基本的弱分类器只包含一个 Haar-like 特征



- **训练弱分类器** $h_i(x) = \begin{cases} 1, & p_i f_i < p_i \theta_i \\ -1, & \text{otherwise} \end{cases}$ ，即寻找分类器阈值 θ_i ，使该分类器对所有样本的分类误差最低。
- 具体步骤如下：
 1. 对于某一特征 f_1 ，计算所有样本的特征值，并排序。
 2. 排序完成后，选取每个元素和其之前元素之间的值作为阈值，计算：
 - 全部正样本的权重和 t_1 ；
 - 全部负样本的权重和 t_0 ；
 - 此阈值之前正样本的权重和 s_1 ；
 - 此阈值之前负样本的权重和 s_0 。
 3. 该阈值的分类误差 $\varepsilon = \min (s_1 + (t_0 - s_0), s_0 + (t_1 - s_1))$ 。
 4. 选取最小 ε 对应的阈值。

训练样本的权重是如何确定的？

- **训练强分类器** $H(x) = \text{sign}(\sum_{i=1}^T \alpha_i h_i(x))$, 即将弱分类器分类结果加权求和, 判断最终结果。
- 具体步骤如下:
 1. 训练集样本数为 N , 样本权重 w_i^j 为 $\frac{1}{N}$;
 2. 归一化权重, 计算每个特征对应弱分类器分类误差;
 3. 选取误差最小弱分类器 $h_i(x)$, 其权重 $\alpha_i = \frac{1}{2} \log(\frac{1-\varepsilon_i}{\varepsilon_i})$;
 4. 重新计算各样本权重: $w_{i+1}^j = \begin{cases} \frac{w_i^j}{2(1-\varepsilon_i)}, & y_i = h_i(x^j) \\ \frac{w_i^j}{2\varepsilon_i}, & y_i \neq h_i(x^j) \end{cases}$;
 5. 循环步骤2、3、4迭代 T 轮, 得到 T 个弱分类器。

分类识别



语义分割



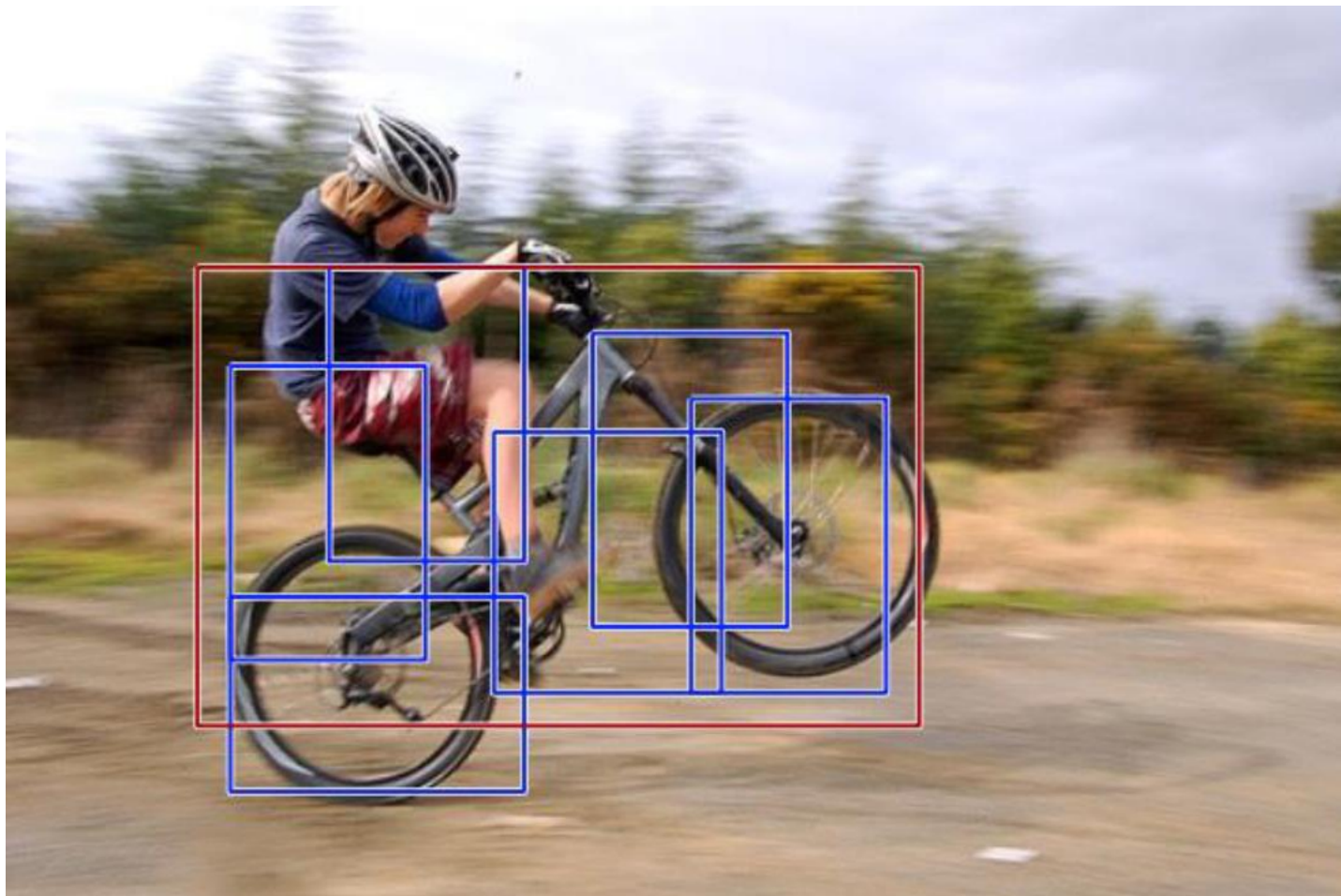
目标检测



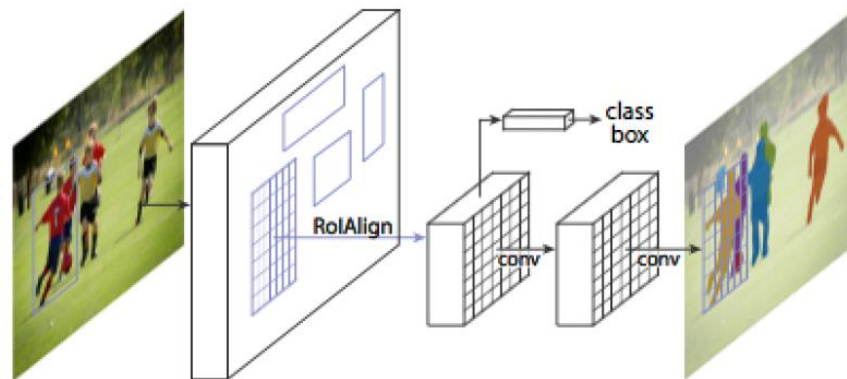
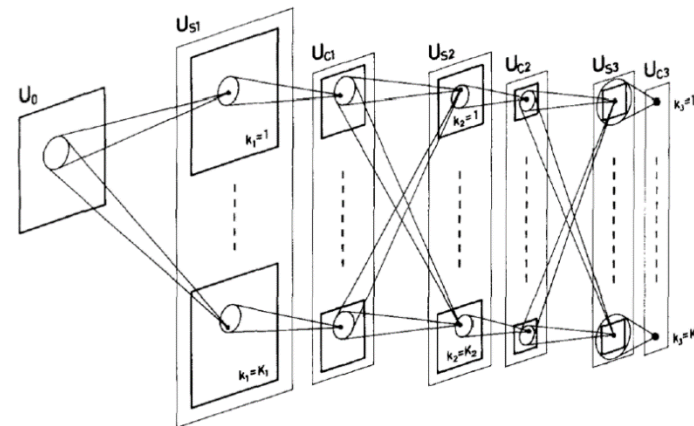
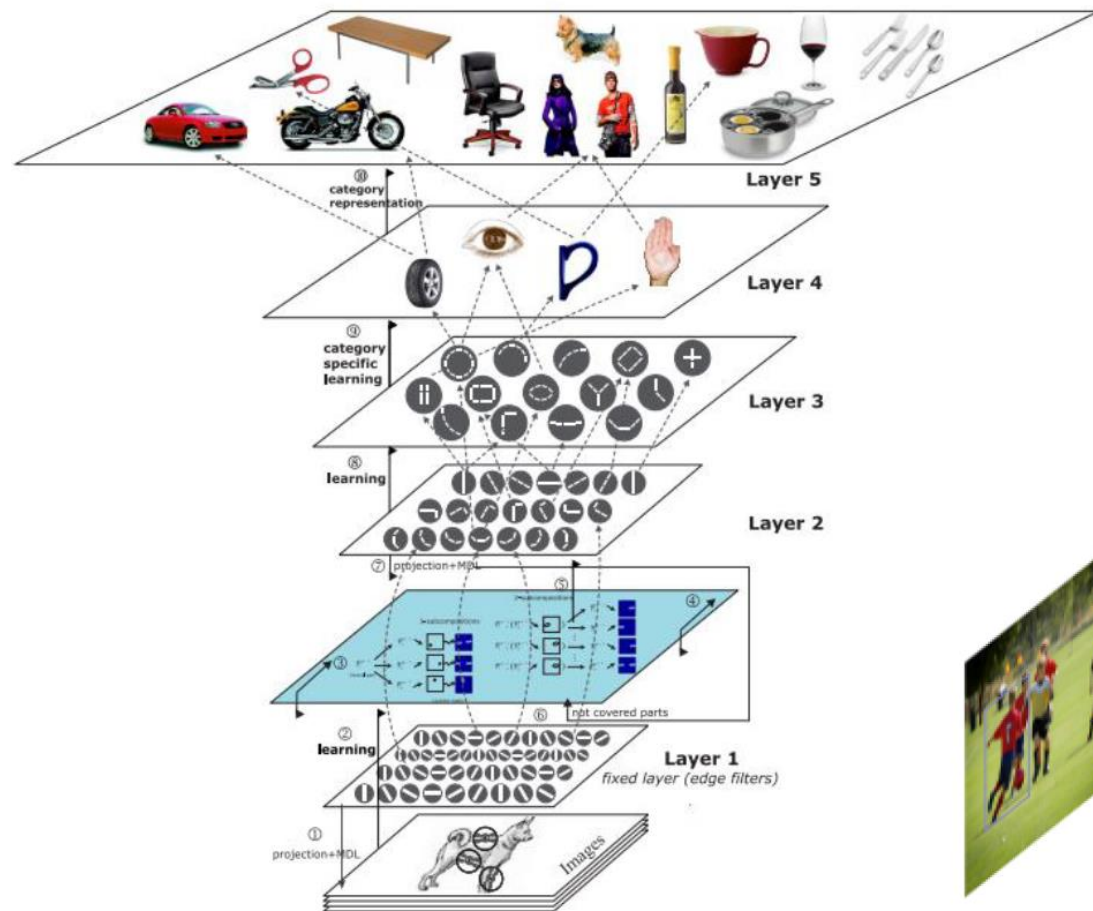
实例分割



回到2009年...

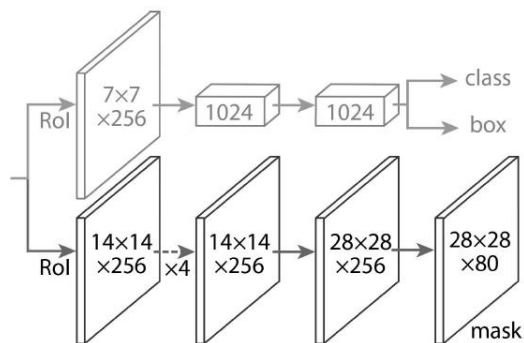


Deformable Part Models

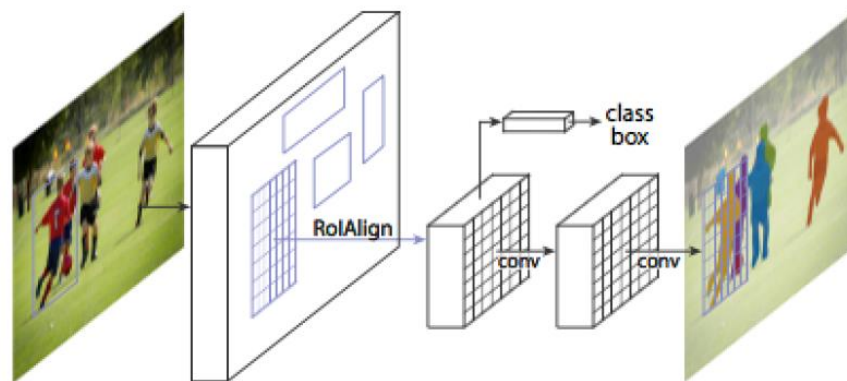
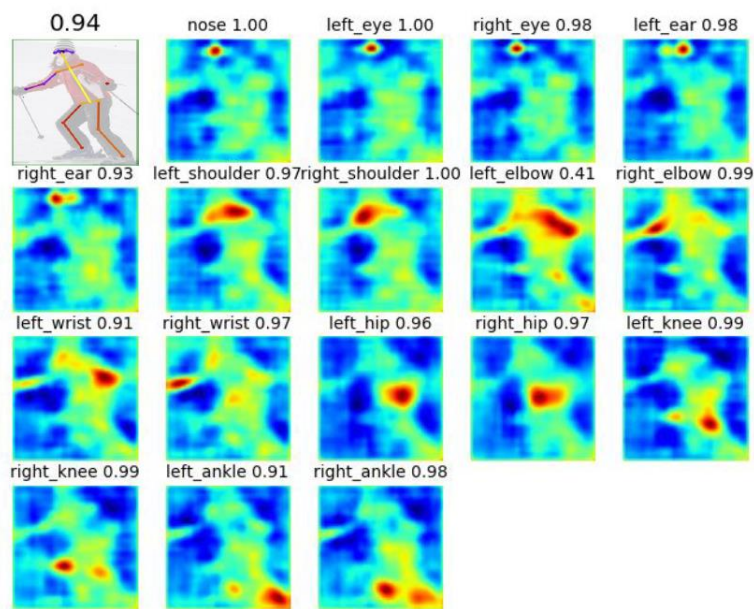


Mask R-CNN. He et al. 2017.





- 增加关键点检测分支 (28x28x17)
- 为每个关键点预测一个热图



Mask R-CNN. He et al. 2017.





课程结束，欢迎提问

THANK YOU FOR WATCHING



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

智能无人系统综合设计