1. Introduction to computer vision

CV 模拟人眼系统的机制：①Attention 机制：人会关注自己感兴趣的区域，忽略嘈杂背景。应用：self-attention。②Hierarchical 分层机制：远近，先看大致的轮廓,找到目标后，再看具体的细节。应用：CNN 中多尺度信息融合、SIFT 和 HOG 中的多尺度算子、深度学习结构。

CV system 相关领域：①自动驾驶：感知、图像处理、定位、规划、人机交互。②收集：人脸识别、全景构建、表情检测。**挑战**：类内方差、尺度差异、运动、弱光、杂乱的背景、遮挡、模糊。

CV system 过程：视觉捕捉，数据预处理，特征提取，检测，分割，建模。

2. Image filtering

高斯核性质：①从图像中删除"高频"成分。②与自身的卷积是另一种高斯分布，所以可以用小 σ 核进行平滑，重复，并得到与更大的 σ 核卷积的效果。使用 σ 核进行两次卷积与$\sqrt{2}\sigma$卷积一次相同。③2D 高斯核可以分离为两个 1D 高斯核的外积。对于 $n\times n$的图像 with $m\times m$的核，分离后复杂度从$O(n^2m^2)$降为$O(n^2m)$。④如果卷积时在边界无 padding 则结果小于原图。padding 方法包括常量填充、镜像填充、复制边缘像素填充等。

噪声：①椒盐：随机出现黑白像素。②脉冲：随机出现白像素。③高斯：从高斯正态分布得出的强度变化。

高斯滤波：适合除高斯噪声，简单高效，但是会模糊边缘且失去特征。**中值滤波**：适合除椒盐噪声，可以保护边缘，新的值是从图像的真实值中得到的，但是非线性而且慢。

锐化：原图−模糊后图像=细节。原图+细节=锐化。锐化的原理是凸显周围与中心的差异，为此使用了 Laplacian 滤波器：

$$H=\begin{bmatrix}0&-1&0\\-1&5&-1\\0&-1&0\end{bmatrix}\ or\ H=\begin{bmatrix}-1&-1&-1\\-1&8&-1\\-1&-1&-1\end{bmatrix}$$

3. Edge detection

梯度：指图像强度变化最明显的方向。把图像视为离散的二元函数$I(x,y)$，那么图像边缘就是函数值突变即梯度较大的位置。由于像素的最小单位是 1，所以以差分近似表示梯度：

$$I_x=I(x+1,y)-I(x-1,y)$$
$$I_y=I(x,y+1)-I(x,y-1)$$

边缘滤波器：

Prewitt:

$$G_x=\begin{bmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{bmatrix}\quad G_y=\begin{bmatrix}1&1&1\\0&0&0\\-1&-1&-1\end{bmatrix}$$

Roberts:

$$G_x=\begin{bmatrix}0&1\\-1&0\end{bmatrix}\quad G_y=\begin{bmatrix}1&0\\0&-1\end{bmatrix}$$

Sobel:

$$G_x=\begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix}\quad G_y=\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix}$$

Canny 算法：i) 步骤：①灰度化。②高斯滤波降噪。③计算梯度的大小和方向（可以使用任意一阶算子）。④（非极大值抑制）NMS：过滤局部非最大值，让边缘变细：根据某点梯度G_x,G_y正负和大小确定该点梯度方向。然后用靠近梯度方向的周围两个像素的梯度与中心点梯度进行比较。若中心点梯度幅度最大则保留，否则置为 0。⑤滞后阈值剔除假边缘，连接破碎边缘：设置 minVal 和 maxVal，值小于 minVal 的像素点被判为非边缘，大于 maxVal 的被判为真边缘。处于之间的，如果与真边缘相邻，则判断为边缘；否则为假边缘。实现上使用 DFS 遍历周围像素点。

4. Local features – corner

角点：在任何方向上移动一个窗口都会引起强度较大的变化。对于窗口W，窗口函数$w(x,y)$（可以定义为在窗口内取 1 在外为 0（下取此定义），或高斯函数），当移动大小$[u,v]$时，定义灰度变化$E(u,v)$:

$$\begin{aligned}E(u,v)&=\sum_{(x,y)\in W}[I(x+u,y+v)-I(x,y)]^2\\&\approx\sum_{(x,y)\in W}\left[I(x,y)+I_xu+I_yv-I(x,y)\right]^2\\&=[u\quad v]\sum_{(x,y)\in W}\begin{bmatrix}I_x^2&I_xI_y\\I_xI_y&I_y^2\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}\\&=[u\quad v]M\begin{bmatrix}u\\v\end{bmatrix}\end{aligned}$$

上式约等于使用了泰勒公式。这里的M经过对角化得:

$$M\sim\begin{bmatrix}\lambda_1&0\\0&\lambda_2\end{bmatrix}$$

角点响应函数：$R=det(M)-\alpha\text{trace}\,(M)^2=\lambda_1\lambda_2-\alpha(\lambda_1+\lambda_2)^2$。如果$\lambda_i$（即$|R|$）都接近于零，则非角点；若其中一个大（$R<0$）则是边缘；若都很大（$R>0$）才是角点。

Harris 角点检测：①使用$w(x,y)$卷积，再计算每个像素的梯度I_x,I_y。②在每个像素窗口内计算二阶矩矩阵 M。③计算角点响应函数 R 选择④R大于某一阈值的点作为角点⑤NMS 找到响应函数的局部最大值。

不变性：算法或特征对某种变换不敏感的性质。

协变性：算法或特征在某种变换下能保持相同的变换性质。Harris 算法有光照、角度不变性，平移、旋转协变性，不具有缩放协变性。对仿射强度变化部分不变。

5. Local features – blob detection

LoG：高斯函数二阶导，是圆形对称斑点检测算子。在图像中强度急剧变化的地方，Laplacian 响应是一个波；两个波靠近时会出现局部最大值，即找到了斑点。Laplacian 响应会随σ增大而减小，因此为保证缩放不变性，需要对 LoG 乘σ^2，即

$$\nabla^2_{\text{norm}}g=\sigma^2\left(\frac{\partial^2g}{\partial x^2}+\frac{\partial^2g}{\partial y^2}\right)$$

为了得到最大响应，Laplacian 的零点必须与斑点中心对齐，即令$\nabla^2_{\text{norm}}g=0$，解得斑点半径$r=\sqrt{2}\sigma$。

尺度-空间斑点检测：使用 scale norm Laplacian 在不同尺度上卷积，当中心点是相邻尺度空间中$3\times3\times3$范围内最大值时才被判定为斑点。

DoG：为减少计算开销，使用 DoG 代替 LoG：$G(x,y,k\sigma)-G(x,y,\sigma)\approx(k-1)\sigma^2\nabla^2G$。

SIFT：①由灰度图像生成高斯金字塔$L_{s\times N}=G(x,y,\sigma)*I(x,y)$。金字塔共 S 组，每组 N 层。每层使用的$\sigma=\sigma_0 2^{s+\frac{n}{N-3}}$。第$s-1$组$n-3$层降采样后作为第$s$组第 0 层。 k=2的s分之一 ②算出 DoG 金字塔$D_s=L_s(n+1)-L_s(n)$，共$S\times(N-1)$层。③DoG 金字塔中，当中心点是相邻尺度空间中$3\times3\times3$范围内最大值时才被判为关键点。④亚像素插值以精准定位关键点，再消除边缘响应。⑤在关键点$d\times d$范围内对 36 个方向直方图统计，选出关键点主方向。⑥关键点附近取$r\times r$矩形并旋转至主方向，在该范围内对 8 个方向直方图统计梯度和方向，拼接成$r\times r\times8$维描述符，并归一化。**SIFT 特性**：尺度、旋转不变，光照、视角、遮挡部分不变，特征维度小。

HOG：①将$H\times W$图像划分为$h\times w$个$n\times n$的 cells。②以长度为180°/20°的向量表示个 cell：将各点梯度按比例分到对应角度，拼接并归一化。③$m\times m$个 cell 组成一个 block，拼接各 cell 的向量并归一化。

④在图像上以s个 cell 步长滑动计算各 block 向量，最后拼成$\left(\lfloor\frac{h-m}{s}\rfloor+1\right)\times\left(\lfloor\frac{w-m}{s}\rfloor+1\right)\times m^2\times$9维向量。**HOG 特征**：HOG 对图像几何和光学形变具有不变性、适用刚性物体特征提取，但无法处理遮挡。

纹理：能够反映材料和外观的属性。**纹理表示**尝试总结局部结构中的重复模式。**滤波器组**通过使用多个滤波器，能够考虑边缘、斑点、方向。

6. Fitting

RANSAC：①随机选择s个（能拟合模型的最少数量）样本点。②拟合模型。③计算所有点的误差，如果内点数大于上一个模型，则使用当前参数。重复上述步骤N次。至少有一个随机样本没有外点的概率p满足：$(1-(1-e)^s)^N=1-p$，迭代次数$N=\log(1-p)/\log(1-(1-e)^s)$。$e$是外点在数据中占比。**优点**：简单，适用于多种问题，通常表现好。**缺点**：需要调很多参数，不能总是根据最小样本数得到很好的模型初始化，低内点率时效果不佳。

Hough 变换：笛卡尔坐标系中一个点对应 Hough 空间一条线。共线的点在 Hough 空间的线交于同一点。**投票方案**：①将 Hough 参数空间离散成网格 bins H。②图像中特征点在 Hough 空间的线每经过一个 bin 就投一票。③以票数最多的 bin 的坐标为所要拟合的直线参数。笛卡尔坐标系参数无界且$k=\infty$无法表示，故应使用极坐标系。对于特征点(x,y)，

$$\rho=xcos\theta+ysin\theta,(\theta\in[0,\pi])$$
$$H(\rho,\theta)=H(\rho,\theta)+1$$

设(ρ_0,θ_0)为$H(\rho,\theta)$局部最大值，则拟合出的直线为$\rho=xcos\theta_0+ysin\theta_0$。**网格划分**：太大导致大量不同线对应一个 bin，太小导致不完全共线的点为不同 bin 投票而错过了线。**噪点处理**：移除不相关特征（例如仅采用显著梯度大小的边缘点）、让附近 bin 也得票。**处理边缘点(x,y)**：θ取梯度方向。**优点**：可以处理遮挡、可以检测出一个模型的多个实例、对噪声具有一定鲁棒性：噪点不太可能对任何某个 bin 产生一致的投票。**缺点**：搜索时间的复杂性随着模型参数数量的增加而呈指数增长。非目标形状会在参数空间中产生虚假峰值。很难选择合适的网格大小

7. Segmentation

分割：将图像分离成连贯的物体，将相似的像素分组在一起以提高进一步处理的效率。

K-means：①随机初始化k个点作为聚类中心均。②将每个数据分配给距离最近的那个聚类中心。③更新聚类中心，每个聚类中心更新为该聚类包含的所有点的平均值。④重复 2~3 步直到没有点。**优点**：简单快速、收敛到局部最优解。**缺点**：内存密集型、需要指定k、对初始中心和外电敏感、不能检测非凸团簇、需要假设均值可计算。

Mean-shift：在特征空间中寻找模式或密度的局

部最大值。①寻找特征（颜色、梯度、纹理等）。②在每个特征点处初始化窗口。③对每个窗口计算从中心到各点的向量和，将中心移向质心。重复该过程直到收敛。④合并结束时在同一峰值或模式附近的窗口。**优点**：通用且独立于应用程序的工具、无模型，不假设数据集群上的任何先前形状（球形、椭圆形等）、只有一个参数（窗口大小 h）、h 具有物理意义（与 k-means 不同）、可查找可变数量的模式、对异常值具有鲁棒性。**缺点**：输出取决于窗口大小、计算（相对）昂贵、不适用于高维数据。**加速方法**：①将结束点附近半径范围内归为同一簇。②将 mean shift 过程中 r/c范围内所有点归于与结束点相同簇。

Graph-cut：①构建图：每个像素点为结点，彼此相连，边的权重是像素点间的相似度（基于颜色、亮度、纹理等）。②团簇 A 和 B 之间某些边的 cost $cut(A,B)=\sum_{p\in A,q\in B} w_{p,q}$，使$cut(A,B)$最小的切割方法就是 min-cut。min-cut 存在 bias，不易得到最佳边界，因此需要 normalized-cut 来归一化线段的大小：适用范围广 内存大 时间复杂度高

$$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)}$$

assoc(A, V)是接触 A 的所有边的权值之和。当我们得到两个具有许多高权重边且它们之间的低权重边很少的簇时，$Ncut$ 值较小。最小化 $Ncut$ 的近似解决方案：广义特征值问题。

8. Visual Recognition

能力：对图像或视频进行分类、检测并定位物体、估计语义和几何属性、对人类活动和事件进行分类。**挑战**：尺度视角光照变化、遮挡等。**路线**：训练样本→图像特征→结合训练标签训练→在测试样本中提取出的特征中用训练好的分类器分类→预测。

Bag of features：①特征提取：构建 blocks，提取外观或外观和位置。②用聚类算法学习常见的“视觉词汇”。③给定一个新图像，提取特征并构建直方图：使用视觉词汇量化特征。④ 通过“视觉词”的频率来表示图像。**优点**：是图像内容紧凑的总结、提供集合的向量表示、灵活的几何形状/变形/视角、实践效果非常好。**缺点**：基本模型忽略几何形状：必须事后验证，或通过特征进行编码、当 Bag 覆盖整个图像时背景和前景混合、最佳词汇形成仍不清楚。

判别模型：对后验比值建模$P(\text{object}|\text{image})/P(\text{no object}|\text{image})$，寻找决策边界。**生成模型**：学习具有 object 的图像的概率分布对似然比建模$P(\text{image}|\text{object})/P(\text{image}|\text{no object})$。

9. Object Detection

基于窗口的目标识别：训练：①获取训练数据。②定义特征。③定义分类器。测试：①滑动窗口。②按分类器评分。**缺点**：必须匹配大量位置/尺度的组合、不好捕获可变形物体、对于稀疏图像计算效率低、上下文会损失

Viola-Jones face detector：①在感兴趣的窗口内用 Haar-like 特征表示局部纹理。②选择判别性特征作为弱分类器。③使用它们的增强组合作为最终分类器。④形成此类分类器的级联，快速拒绝明显的阴性结果。**实时目标检测的“范式”方法**：训练很慢，但检测非常快、用于快速特征评估的 integral image、增强特征选择、用于快速拒绝非面部窗口的注意力级联。

10. Deep Learning

权重初始化：典型方法是$w\sim N(0,\sigma^2)$以避免减少或增强层响应的方差（可能导致梯度消失或爆炸）。常用方法还有 Xavier、Kaiming、Pre-train+Fine-tune。bias 初始化为 0。**LR decay**：exp、cos 等，最常用是每隔几轮就以一个常数因比例降低。**Mini-batch SGD**：每轮每次用一个小 batch 更新参数，能引入随机性，缓解局部最优问题。小 mini-batch 有更小的内存开销，更大的梯度噪音；大 mini-batch 开更大、更少次更新参数、更少的梯度方差。**Batch normalization**：$y_i=\gamma\frac{x_i-\mu_B}{\sqrt{\sigma_B^2-\epsilon}}+\beta$，放在非线性层和激活函数之间。**优点**：避免梯度消失或爆炸、使大多数激活原理非线性饱和区、加速训练收敛。**缺点**：在小 mini-batch size 上效果不佳。**Dropout**：**Train:** 更新参数之前，将 p% 的神经元暂时隐藏（激活值设为 0），以新的网络进行训练。**Test**：所有权重乘 $1-p$。反卷积：$S_{rconv}=(S_{img}-1)*stride+S_{filter}-2pad$

CNN：卷积 output size= (img_sz + 2padding − ksize)/stride + 1。参数量= $n\times(\text{ksize}^2\times c+1)$。

向下取整

池化：放在非线性层后，对每个激活图进行操作、无可学习参数、可引入空间不变性。

Image Classification：**AlexNet**：用两块 GPU 加速训练、数据增强、ReLU、重叠最大池化、dropout。**VGG**：使用了更小的3×3核，这使得网络有更深的结构，捕捉更复杂的特征。**GoogLENet**：inception 模块：将1×1，3×3，5×5的卷积结果和3×3最大池化结果相拼接。这种结构允许网络在不同的尺度上捕获信息，而不会显著增加计算成本。**ResNet**：残差跳连、bottlenect 块通过1×1卷积先降维然后卷积再升维度，减小参数量。ResNet 学习输入和输出之间的残差，使得在深层网络中不会梯度消失。

Segmentation: 全卷积网络：FCN 通过替换传统 CNN 中的 FC 为 conv，可以接受任意大小的输入图像，并输出相应大小的分割图，使得模型能够在像素级别上输出分割结果。**上采样**：unpooling、deconv、dilated conv。上采样允许网络从深层的、空间分辨率较低的特征映射中恢复到输入图像的原始分辨率。

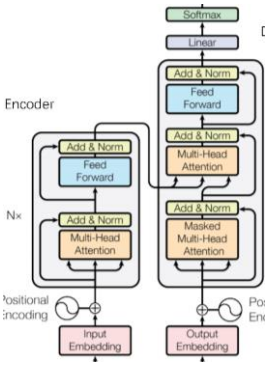
Object Detection：**R-CNN**：①selective search 选出 region proposal。②裁剪后用 CNN 提取特征。③用分别识别各个对象的 SVMs 分类。其动机是利用深度学习强大的特征提取能力来改进对象检测的准确性。**Fast R-CNN**：①在全图 CNN 提取特征。②selective search 在原图选出 region proposal。③在特征图中找到每一个 region proposal 对应的框，使用 ROI pooling 把每一个特征框划为统一大小。④使用 FC 分别进行分类和 bbox 回归。**Faster R-CNN**：把 Fast R-CNN 的 selective search 换成 RPN 以提高搜索速度。RPN 输出是判断是否包含 object 的分数和 bbox 的四个坐标。**YOLO**：分类和定位在同一步骤中完成。

Transformer：

$$\text{attn}(Q,K,V)=\frac{\text{softmax}(QK^\top)}{\sqrt{d_k}}V$$

Encoder self-attn 的输入是前一层的输出。Decoder self-attn 输入是 masked 的前一层输出。cross attn 的 Key Value 是 Encoder 的输出，Query 是上一层输出。

ViT：把图像分为 patches，线性映射到标准 transformer encoder。**CNN vs Transformer**：CNN



有平移不变性（跨空间位置共享内核）、有很好的局部模式感知能力、限于核大小，感受野局限。Transformer 有全局感受野，适合长距离依赖、无结构先验，在小规模数据中容易过拟合。

自监督学习：无监督学习的特例，意在解决难以获得人工标注数据的困境。直接从数据中学习的方法可以用于数据预测：上色、图像修复，缺陷检测。**自监督 vs 无监督：**无监督学习：任何类型的无标签学习、聚类和量化、降维、流形学习、密度估计。自监督学习：模型从数据中“组成”标签，然后解决监督任务。**自监督 vs 生成式：**两者都旨在从数据中学习，而无需手动标签注释。生成式方法旨在对数据分布进行建模，生成逼真的图像。自监督学习旨在通过 pretext task 学习高级语义特征。**用前置任务学习的方法**：前置任务有上下文预测、解决拼图问题、旋转预测。**对比方法**：从数据点的两个转换版本中提取表示，鼓励这些表示相似，即同时使用正负样本对。contrastive loss 如下：

$$-\log\frac{\exp(\cos(x,x^+)/\tau)}{\exp(\cos(x,x^+)/\tau)+\sum_{j=1}^{N-1}\exp(\cos(x,x_j^-)/\tau}$$

其中τ是温度系数。越大则结果分布越集中，越小则结果分布越平均。主要挑战：采样负样本。**非对比方法：**仅使用正样本训练。主要挑战：避免退化解决方案（所有表示都崩溃为恒定输出值）**迁移学习**：将知识从源模型转移到目标任务。**Domain shift**：训练和测试数据具有不同的分布。**微调方法**：通过源数据训练模型，然后通过目标数据微调模型。由于目标数据有限，要小心过拟合，这种情况下复制前几层网络参数，仅训练后几层网络。**Domain adaption**：Discrepancy-based：最小化特征空间中的 domain distance，工作重点是设计合理的距离。Adversarial-based：特征提取器最大化标签分类准确率+最小化 domain 分类（判断数据来自于原域还是目标域）准确率。标签预测其最大化标签分类准确率。domain 分类器最大化域标签准确率。Reconstruction-based：源或目标样本的数据重建是辅助任务。同时专注于在两个域之间创建共享表示并保持每个域的单独特征。

知识蒸馏：将较大深度神经网络中的知识提取到小型网络中。Response-based：利用教师模型最后输出层的神经响应进行迁移。直接模仿教师模型的最终预测。Feature-based：在中间层就进行迁移，利于训练深层网络。Relation-Based：目标是让学生模型尽可能学习并保留教师模型中的这些关系信息，而不仅仅是模仿单个样本的分类输出

GAN

- 生成器 G：输入噪声 $z\sim p_z(z)$，生成伪样本 $G(z)$。
- 判别器 D：判别样本是真实数据 p_{data} 还是伪数据 p_g。

目标函数：

$$\min_G\max_D V(D,G)=\mathbb{E}_{x\sim p_{data}}[\log D(x)]+\mathbb{E}_{z\sim p_z}[\log(1-D(G(z)))]$$

diffusion

通过**正向扩散**逐步添加噪声将数据转为高斯分布，并通过**反向扩散**从噪声逐步恢复数据。

公式：

正向扩散：$q(x_t|x_{t-1})=\mathcal{N}(x_t;\sqrt{1-\beta_t}x_{t-1},\beta_t\mathbf{I})$

反向扩散：$p_\theta(x_{t-1}|x_t)=\mathcal{N}(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta(x_t,t))$

损失函数：$L=\mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon-\epsilon_\theta(x_t,t)\|^2\right]$